

Blockchains

Introduction to Blockchains

Christian Grothoff

Bern University of Applied Sciences

11.04.2025

2025-01-06

Blockchains

Blockchains
Introduction to Blockchains

Christian Grothoff
Bern University of Applied Sciences
11.04.2025

1. In this lecture, we will give you a first brief introduction to Blockchains.
2. Blockchains, like AI, are currently a hype topic in computing.
3. It is thus important to understand what they can or cannot do.

Learning Objectives

What is a Blockchain?

What properties are Blockchains claimed to have?

How does Proof-of-Work solve the Byzantine consensus problem?

Bitcoin and Payments: A good match?

What are other applications for Blockchains?

Bonus: Depolymerization [5]

2025-01-06

Blockchains

Learning Objectives

1. One difficulty with Blockchains is that the term is not well defined. So we will begin with a high-level illustration of the concept, omitting details that do not universally apply.
2. Then we will look at what Blockchain proponents claim as the properties of Blockchains.
3. The consensus problem is central to all blockchains, so we will explore this in more detail.
4. Payments are seen as one big application domain for Blockchains, so we will see how suitable Blockchains are for this key application.
5. Finally, we will see what other applications may benefit from Blockchains.

What is a Blockchain?
What properties are Blockchains claimed to have?
How does Proof-of-Work solve the Byzantine consensus problem?
Bitcoin and Payments: A good match?
What are other applications for Blockchains?
Bonus: Depolymerization [5]

Blockchain¹



¹Illustrations by Alexandra Dirksen, IAS, TUBS [4]

1. We will begin our exploration with a simple transaction as a starting point: Bob wants to buy a phone from Alice.
2. We'll assume Bob already has some "money" in the system, let's not yet worry where it came from.

Blockchain



2025-01-06

Blockchains

└─ What is a Blockchain?

└─ Blockchain

1. To make the transaction “real”, Alice and Bob take a snapshot of their transaction data.
2. They then make it public by posting it on a public bulletin board.
3. Everyone in the world can then see that Alice sold her phone to Bob.



1. Next, let's assume Peter wants to buy a car from Charlie.
2. Now, in this case, it probably doesn't matter that this happens after Alice sold her phone to Bob.
3. But, sometimes the order of transactions matters.
4. Just imagine Alice buying the car from Peter with the money from Bob.



1. Next, let's assume Peter wants to buy a car from Charlie.
2. Now, in this case, it probably doesn't matter that this happens after Alice sold her phone to Bob.
3. But, sometimes the order of transactions matters.
4. Just imagine Alice buying the car from Peter with the money from Bob.

Blockchain



2025-01-06

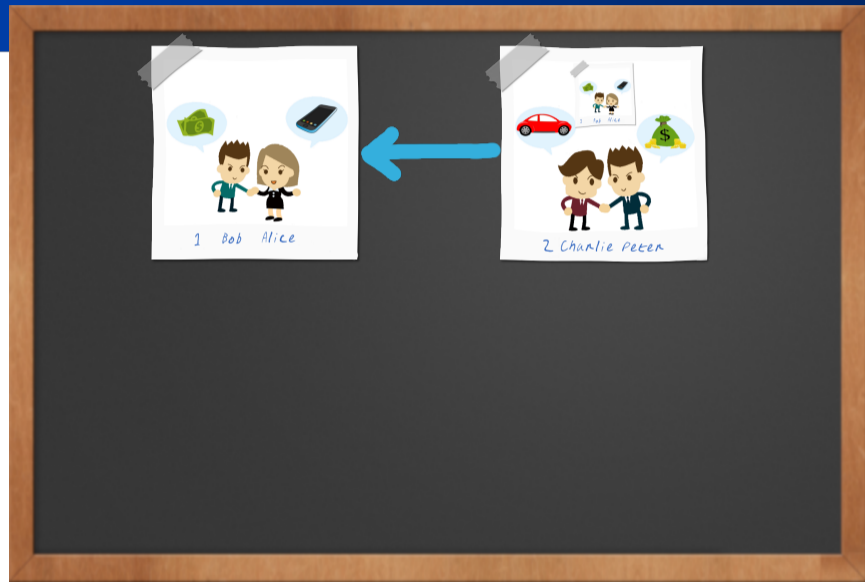
Blockchains

└─ What is a Blockchain?

└─ Blockchain

1. Charlie and Peter can show that Charlie sells the car after Alice sold her phone by putting the snapshot of Alice and Bob's transaction into their background when producing evidence of their own transaction.
2. This both affirms Alice and Bob's transaction and establishes a transaction order.
3. Cryptographically, it is of course enough to put the hash of the original transaction into the new snapshot.

Blockchain



2025-01-06

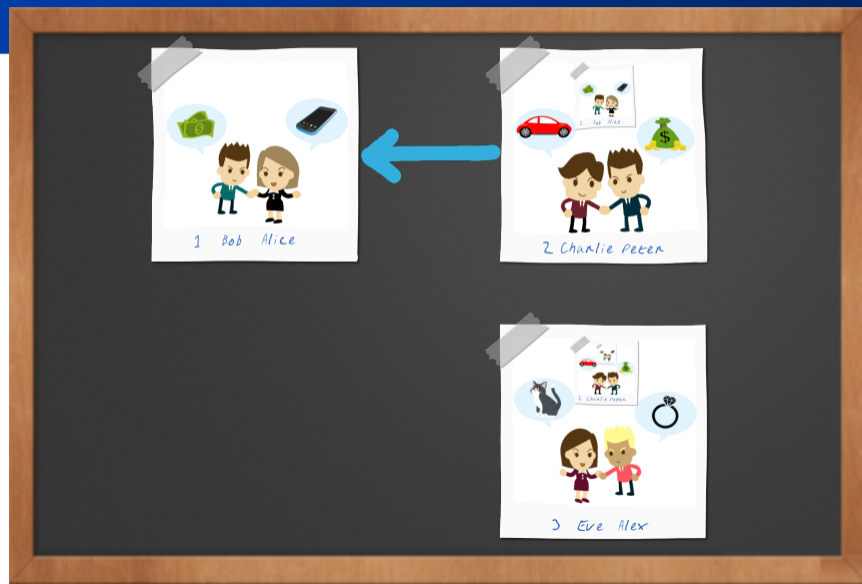
Blockchains

└─ What is a Blockchain?

└─ Blockchain

1. As before, the snapshot of the new transaction is put up on the public bulletin board.

Blockchain



2025-01-06

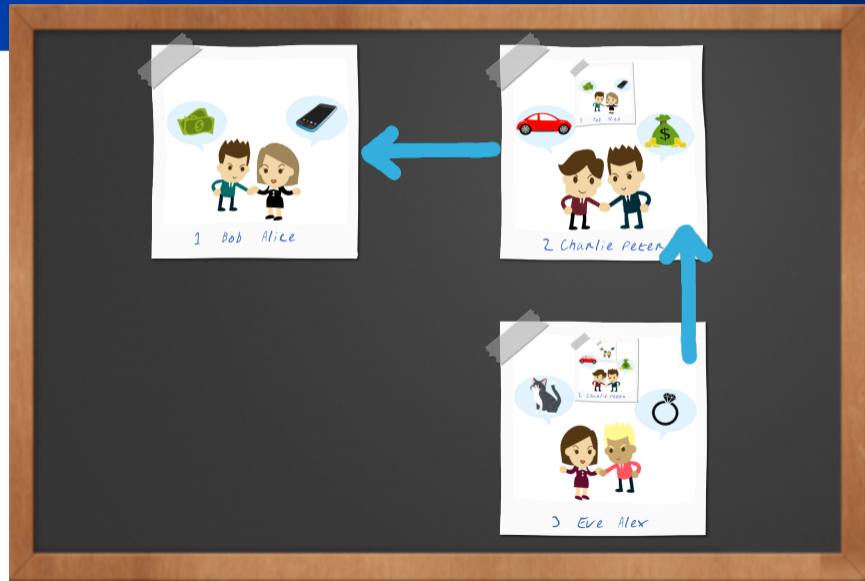
Blockchains

└ What is a Blockchain?

└ Blockchain

1. The hash of the previous transaction in the background “chains” the two transactions.
2. If somebody wanted to now alter the transaction between Alice and Bob, they would also have to alter the snapshot posted by Charlie and Peter to ensure consistency of the chain.

Blockchain



2025-01-06

Blockchains

└─ What is a Blockchain?

└─ Blockchain

1. Further transactions follow the same pattern.

Blockchain

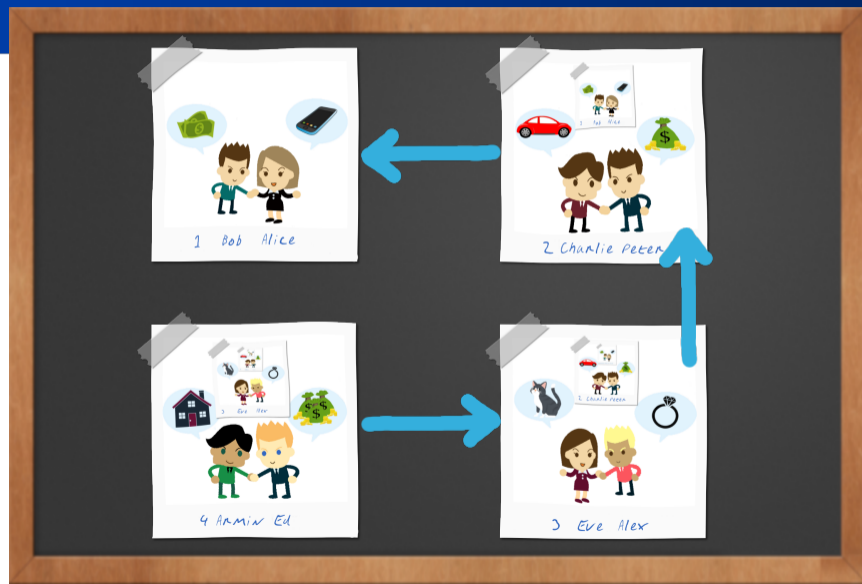


2025-01-06

Blockchains
└─ What is a Blockchain?
└─ Blockchain

1. Note that each new snapshot does not only affirm its immediate predecessor, but transitively all previous transactions.

Blockchain



2025-01-06

Blockchains

└─ What is a Blockchain?

└─ Blockchain

1. Due to the use of a hash, the actual size of each block (snapshot) is pretty constant: the reference to the previous block always has the same size, regardless of how long the chain has gotten.
2. However, to fully understand the balances of people involved, we do need the full chain, and not just the last snapshot.
3. So a Blockchain grows linearly as more transactions are added, so space consumption is a key concern.

Advertised Blockchain “properties”



2025-01-06

Blockchains

└ What properties are Blockchains claimed to have?

└ Advertised Blockchain “properties”

1. Now, let's talk about the properties Blockchains are frequently proclaimed to have.
2. I'm saying “proclaimed” here, as each of these properties kind-of holds.
3. And the kind-of is critical as the limitations are serious sources of problems.

Immutability



2025-01-06

Blockchains

└ What properties are Blockchains claimed to have?

└ Immutability

1. Immutability means that once a transaction has been posted, it is final and cannot be changed anymore.
2. The claim does not arise from transactions being posted in public, as there could be conflicting pictures posted in public.
3. Instead, the idea is that *old* transactions cannot be modified because one would need to update all of the newer transaction records as well.
4. So this is a cost-based argument: to modify the transaction between Charlie and Peter, we would need to re-do the work for subsequent transactions between Eve and Alex and Armin and Ed.
5. But, at least in principle, it is of course always possible to re-do that work.

Transparency



2025-01-06

Blockchains

└ What properties are Blockchains claimed to have?

└ Transparency

1. The transparency property is that everyone sees everything that is going on with the Blockchain.
2. This is because all transactions are public on a bulletin board.
3. Sure, honest participants in the peer-to-peer network will share their view of the Blockchain.
4. However, the Internet doesn't exactly have a bulletin board, and downloading large Blockchains (some have grown to many Terrabytes) can be prohibitive for most users.
5. Also, network outages can still prevent timely visibility.
6. Finally, malicious participants can hide new blocks.
7. There are attacks where hiding new blocks from competitors can yield economic benefits.

Decentralisation



2025-01-06

Blockchains

└ What properties are Blockchains claimed to have?

└ Decentralisation

1. The decentralization claim is that anyone can participate in the system on an equal footing.
2. There is no operator, the system operates as a *permissionless* peer-to-peer network.
3. In reality, few users can afford to download the entire Blockchain, and to effectively participate requires specialized resources.
4. Thus, very few entities end up dominating the process.
5. Finally, not all “Blockchains” are open decentralized peer-to-peer networks. Sometimes closed proprietary systems with a well-defined restricted set of participants are still called “Blockchains”. These are often called *permissioned* or *private* Blockchains.

Autonomy



2025-01-06

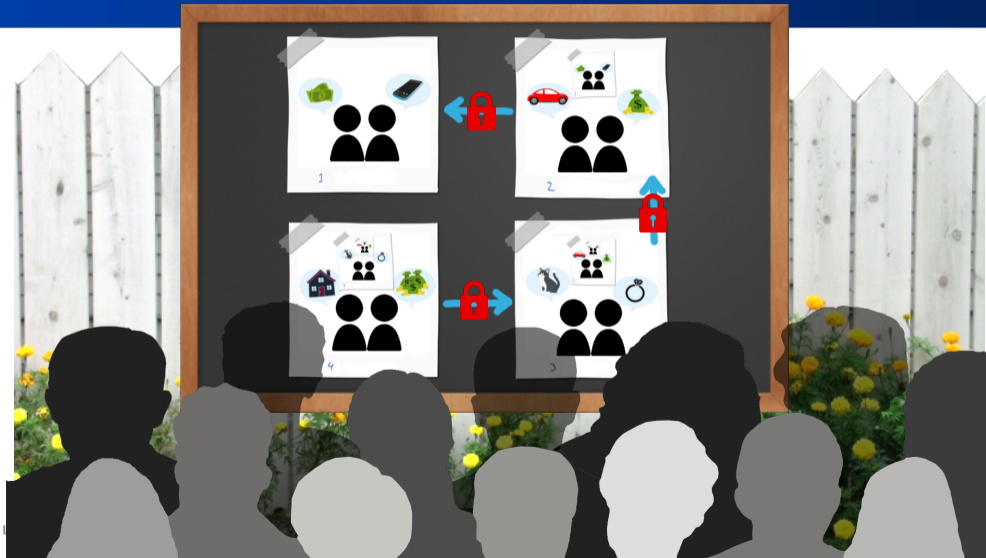
Blockchains

└ What properties are Blockchains claimed to have?

└ Autonomy

1. Autonomy is about the Blockchain continuing to work even if some participants drop out.
2. This ignores the issue that the participants that dropped out may have the resources to create a competing fork of the Blockchain, thus threatening immutability.
3. The possibility that anyone *could* find the resources to run a peer at any time makes the system very hard to shut down legally.

Anonymity



2025-01-06

Blockchains

└ What properties are Blockchains claimed to have?

└ Anonymity

1. Anonymity technically is about ensuring that a transaction cannot be linked to the individuals involved or other transactions (by the same individuals).
2. On a typical Blockchain, transactions are not actually between individuals but between cryptographic keys.
3. Anonymity is claimed as the individual in control of a private key may not be known.
4. However, multiple transactions by the same key are linkable, so often Blockchains at best achieve pseudonymity which is a weaker form of anonymity where transactions can still be linked.
5. Pseudonymity can be difficult to maintain, as one transaction that exposes the link to one's identity may expose many other transactions linked to the same key.
6. Furthermore, few Blockchains include anonymization at the IP layer, so merely accessing the overlay network to add a new transaction has the potential to break anonymity.

Summary: Blockchain “properties”



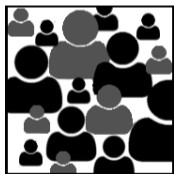
Im-
mutability



Trans-
parency



Anonymity



Decentralisation



Autonomy



Irreversibil-
ity

2025-01-06

Blockchains

└ What properties are Blockchains claimed to have?

└ Summary: Blockchain “properties”

1. Irreversibility or finality are a variant of immutability, which stresses that transactions also cannot later be deleted. The same caveats as with immutability apply.
2. Again, all of these **only** hold with **significant caveats!**
3. Immutability, autonomy, decentralization and anonymity are the key reasons why Blockchains can be seen as “censorship-resistant”.
4. For example, some CSAM posted on some Blockchains cannot be effectively removed.

Who gets to append the next block?

1. The most critical operation of any Blockchain is adding new transactions.
2. In practice, the process is not done for an individual transaction, but for a set of transactions.
3. A *block* is thus simply a new set of transactions (with a hash chaining it to its predecessor) that is being appended to the Blockchain.
4. Alice can use her private key to sign two conflicting transactions: one to send all her money to Bob, and another to send all her money to Carol.
5. All transaction systems need to ensure consistency: agreement about who owns what.
6. Thus, it is critical to determine which block is valid to decide: Should the “money” go to Bob or to Carol?

Proof of Work



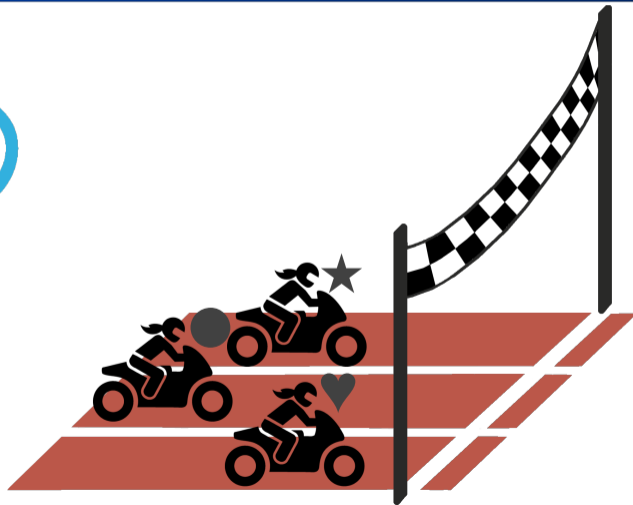
2025-01-06

Blockchains

- └ How does Proof-of-Work solve the Byzantine consensus problem?
 - └ Proof of Work

1. Proof of work is an attempt to solve the Byzantine consensus problem.
2. Here, we have a cloud of conflicting possible future realities.
3. Each of these futures has a stakeholder (miner) that would primarily benefit from this future.
4. These miners are *competitors*, each competing for their version of reality to become consensus.

Proof of Work



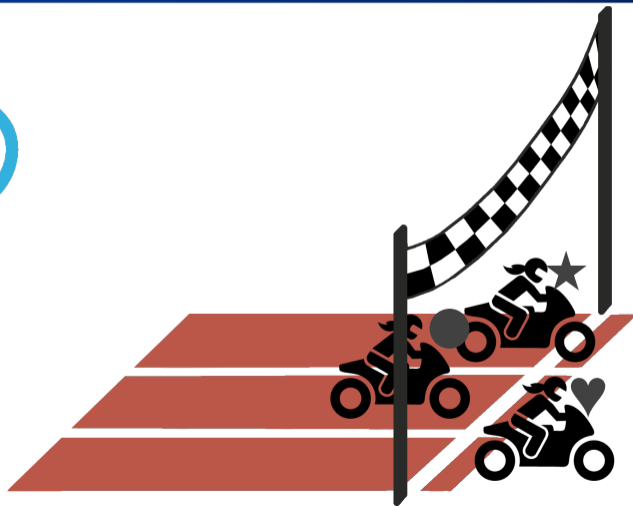
2025-01-06

Blockchains

- └ How does Proof-of-Work solve the Byzantine consensus problem?
 - └ Proof of Work

1. Proof of Work can be seen as a race.
2. The miners are challenged to solve a computational puzzle.
3. A typical puzzle involves finding an input that results in a (partial) hash collision.
4. For example, given a future reality represented by block B , find an I such that $H(B, I) \bmod N \leq M$ for a given value of N and M .
5. Given a cryptographic hash function, the chance of winning is $\frac{M}{N}$ per random input I .

Proof of Work



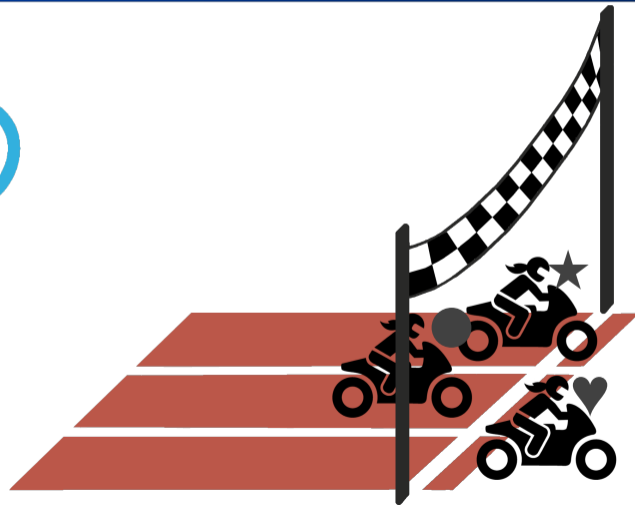
2025-01-06

Blockchains

- └ How does Proof-of-Work solve the Byzantine consensus problem?
 - └ Proof of Work

1. The block proposed by the first miner to solve the puzzle wins.
2. Miners can improve their chances to finish the computational puzzle first by putting in more computational power.
3. Usually, a monetary award is made to the miner who wins.
4. The choice of puzzle may provide advantages to miners using general-purpose CPUs, GPUs or specialized ASICs.
5. Some Blockchains use puzzles that are not about computational power but about storage space.

Proof of Work



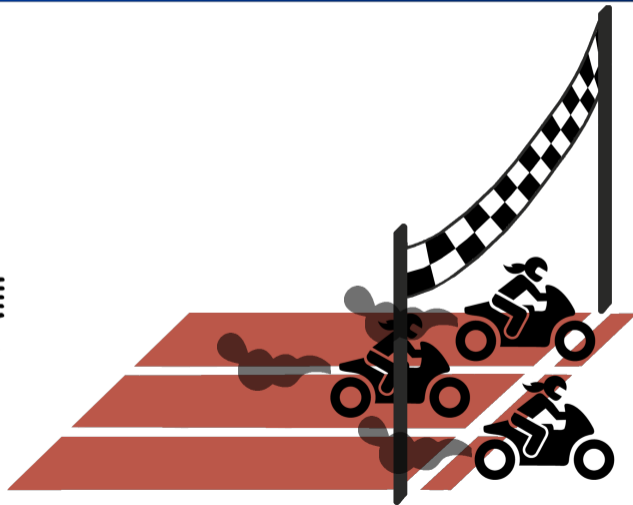
2025-01-06

Blockchains

- └ How does Proof-of-Work solve the Byzantine consensus problem?
 - └ Proof of Work

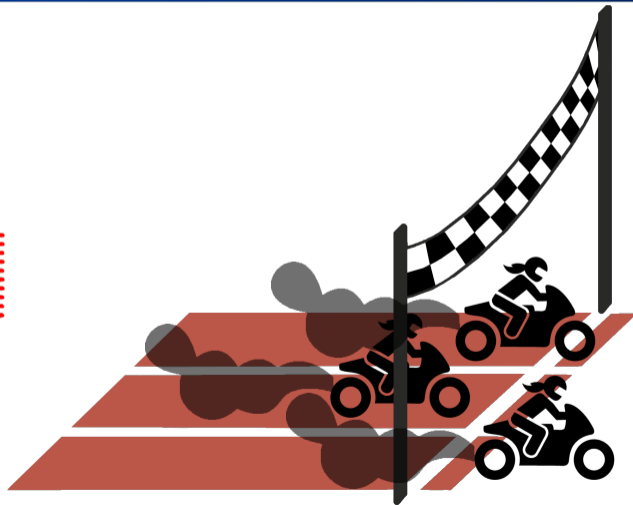
1. Usually, a monetary award is made to the miner who wins.
2. For example, in Bitcoin a special transaction that increases the money supply is made to the miner's account.
3. Outside of Blockchains, monetary policy issues are usually decided by experts at central banks behind closed doors.
4. Money supply policies programmed into Blockchains are detached from economic realities and used as a primary differentiator when advertising new shitcoins to unsavvy investors.
5. Algorithms designed to only ever create a finite amount of cryptocurrency create artificial scarcity and thus embody the cryptocurrency with a sense of value.

Proof of Work



1. After the race is before the race.
2. Once a new Block was mined, all miners are supposed to start with solving the next puzzle.
3. The general rule is that mining should be done on the longest chain.
4. The longest chain, that had the most puzzles solved, is considered "valid".

Proof of Work



└ How does Proof-of-Work solve the Byzantine consensus problem?

└ Proof of Work

1. Solving proof of work is a major contributor to global energy consumption. Bitcoin alone clocks **137 TWh/year**. (<https://www.statista.com/statistics/881472/worldwide-bitcoin-energy-consumption/>)
2. Globally, we produce $\approx 29,000$ TWh/year (<https://www.statista.com/statistics/270281/electricity-generation-worldwide/>). So Bitcoin uses 0.5%.

Bitcoin for Payments

Bitcoin claims to be a *payment system* using a Blockchain:

- ▶ Public keys identify accounts, private keys used to send money from the account into other accounts.
- ▶ Set of internally consistent transactions form each block
- ▶ Each block includes a transaction creating fresh coins and transferring applicable fees to block creator
- ▶ Computational difficulty adjusts to mining power. A new block is mined in ≈ 10 minutes
- ▶ Amount of bitcoin money supply created per block is exponentially decreasing

2025-01-06

Blockchains

└ Bitcoin and Payments: A good match?

└ Bitcoin for Payments

Bitcoin claims to be a payment system using a Blockchain:

- ▶ Public keys identify accounts, private keys used to send money from the account into other accounts.
- ▶ Set of internally consistent transactions form each block
- ▶ Each block includes a transaction creating fresh coins and transferring applicable fees to block creator
- ▶ Computational difficulty adjusts to mining power. A new block is mined in ≈ 10 minutes
- ▶ Amount of bitcoin money supply created per block is exponentially decreasing

1. Miners make two types of profits from each block they mine.
2. One source of income are the new bitcoins created, effectively inflating the money supply.
3. The second source are fees paid by users wanting their transactions to be included in the block.
4. Transactions with higher fees are thus more likely to be included in a block by miners.

Rational Forking

Imagine:

- ▶ The previous block had a transaction from X to Y over 100 BTC with a fee of 0.001 BTC, a block reward of 7.5 BTC and total transaction fees of 5 BTC.
- ▶ The next consistent blocks can be assumed to again have block rewards of 7.5 BTC and transaction fees of 5 BTC.
- ▶ The issuer X of the 100 BTC transaction now signs a conflicting transaction where 50 BTC go to Z with a 25 BTC transaction fee.

What is the **rational** behavior for a miner M ?

2025-01-06

Blockchains

└ Bitcoin and Payments: A good match?

└ Rational Forking

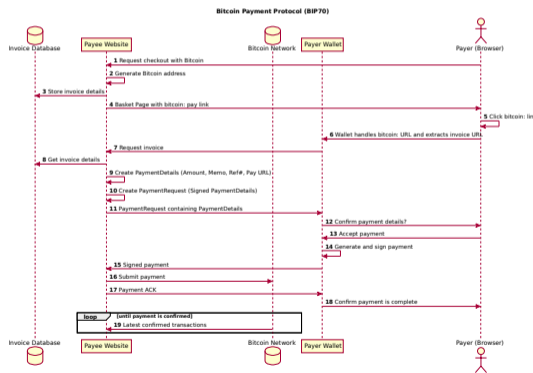
Imagine:

- ▶ The previous block had a transaction from X to Y over 100 BTC with a fee of 0.001 BTC, a block reward of 7.5 BTC and total transaction fees of 5 BTC.
- ▶ The next consistent blocks can be assumed to again have block rewards of 7.5 BTC and transaction fees of 5 BTC.
- ▶ The issuer X of the 100 BTC transaction now signs a conflicting transaction where 50 BTC go to Z with a 25 BTC transaction fee.

What is the **rational** behavior for a miner M ?

1. Re-mine the previous block P to produce a fork is clearly financially more beneficial than mining the next block.
2. But, you want the other miners to then also move to your fork.
3. Thus, M should post another dummy transaction from M with say a 5 BTC transaction fee that is only valid if M 's fork becomes the new longest chain, effectively rewarding other miners to switch.
4. In reality, the game theory involved gets quite complex. Note that especially the immutability and durability properties of the Blockchain depend on the game-theoretic results of the case.

Bitcoin Payment flow (by W3C Payment Interest Group)



2025-01-06

Blockchains

└ Bitcoin and Payments: A good match?

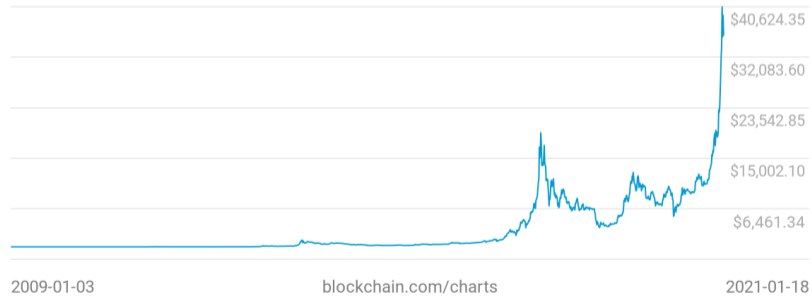
└ Bitcoin Payment flow (by W3C Payment Interest Group)



1. Here is an example for a W3C proposal for using Bitcoin for online payments.
2. Most of the details do not matter too much for us here.
3. But the critical step is the **loop** at the bottom in step **19**.
4. Here, after the payer was given a payment confirmation, the receiver is expected to loop **until payment is confirmed**.
5. But what does that mean? Satoshi's original paper suggests to wait for 6 blocks after the block containing the transaction in question to avoid issues with forks. At 10 minutes per block, this would take **one hour**.

The Value of Bitcoin

Market Price (USD)
\$35,793.01



2025-01-06

Blockchains

└ Bitcoin and Payments: A good match?

└ The Value of Bitcoin



1. The market value of Bitcoin has been extremely volatile.
2. Reasons include hype-driven speculation and limited liquidity.
3. While this type of fluctuation inherently benefits some and bankrupts others, we need to remember the golden rule of pure Pyramid schemes: This is a zero-sum game, so what one person gains, another must lose.
4. Central bankers define “money” as an asset with 3 properties: it can be used to **purchase** goods and services by entities other than the issuer, serves as a **store of value**, and as a **unit of account**. [3]
5. To serve as a **unit of account** requires reasonable stability. Bitcoin is not useful in that respect.

Mining requires:

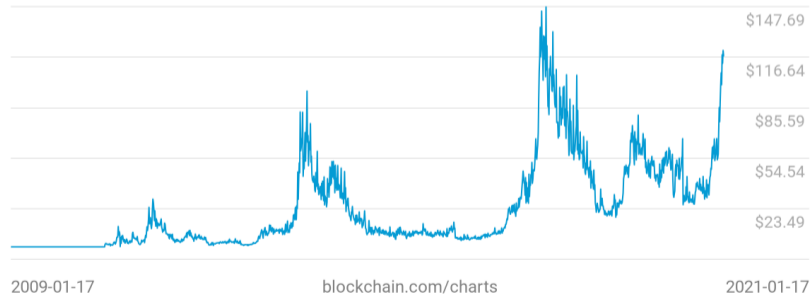
- ▶ Learning pending transactions from peers
- ▶ Selecting a subset of transactions which is valid (no double spending) by computing current account balances against the entire history
- ▶ Finding a hash collision (with adaptive difficulty)
- ▶ Propagating the new block to other miners

Usually specialized systems are used for finding hash collisions.

1. Checking that transactions are valid requires checking a cryptographic signature and an account balance.
2. To know all account balances, one must compute it from the (entire) history of the Blockchain.
3. A single wallet can use many keys (= accounts) to hold its assets. Using more keys may improve privacy.
4. Thus, there can be many more accounts than there are actual users on a Blockchain!

Mining cost

Cost per Transaction
\$117.47



Current average transaction value: \approx 1000 USD

2025-01-06

Blockchains

└ Bitcoin and Payments: A good match?

└ Mining cost



1. This chart is based on the mining rewards from computing each block, that is both the new money created as well as the transaction fees.
2. This total is then divided by the number of transactions in the block and converted from BTC to USD using the current exchange rate.
3. It does thus not reflect the actual fee paid, but includes the cost all Bitcoin owners are indirectly bearing via inflation of the money supply.
4. This is the **rational** limit of resources (hardware, electricity, bandwidth) a miner would spend to on proof-of-work per transaction in a perfect market.
5. In reality, miners of course also compete by going for the cheapest available electricity supply (= poor countries where governments provide subsidies to make electricity affordable). Miners have of course also directly been stealing electric power.
(<https://www.bbc.com/news/uk-england-birmingham-57280115>)
6. At this cost, using Bitcoin to transact is primarily rational for very large and/or illegal transactions.

Bitcoin performance

- ▶ Privacy: all transactions happen in the clear in public view
- ▶ Latency: transactions take 1h to kind-of be confirmed
- ▶ Storage: grows linearly forever, no garbage collection
- ▶ Power: Bitcoin mining consumes more than the Netherlands today
- ▶ Rate: Network handles at most about 7 transactions per second
- ▶ Accountability: use of public keys as addresses enables criminal use

⇒ Bitcoin fever lasting for years. Why?

2025-01-06

Blockchains

└ Bitcoin and Payments: A good match?

└ Bitcoin performance

- ▶ Privacy: all transactions happen in the clear in public view
- ▶ Latency: transactions take 1h to kind-of be confirmed
- ▶ Storage: grows linearly forever, no garbage collection
- ▶ Power: Bitcoin mining consumes more than the Netherlands today
- ▶ Rate: Network handles at most about 7 transactions per second
- ▶ Accountability: use of public keys as addresses enables criminal use

→ Bitcoin fever lasting for years. Why?

1. Privacy: So as a user, you have no good privacy assurances as you have to assume that someone might be able to link the public key of your account(s) to you eventually.
2. Latency: The EU now requires instant payments to be done in about 10 seconds between European banks. So 1 hour is not exactly fast.
3. Storage: This is a forever-cost. Even just 10 year retention that is common in banking, storage costs can be a major IT cost driver.
4. Power consumption by country:
<https://www.statista.com/statistics/1260553/eu-power-demand-country/>
5. Rate: This is a theoretical limit, in practice Bitcoin has barely crossed over 4 TPS, despite having always over 100k transactions in the pool waiting to be included in the block chain.
6. Accountability: Blockchains with stronger privacy assurances of course do even worse here.
7. Why? Investors have an interest in pumping their investment!

- ▶ Dogecoin: same as Bitcoin, just named after a dog meme (an idea that is obviously worth billions!)
- ▶ Zcash: uses ZKSNARKs³ to hide transactions (criminal activity on Bitcoin was too low)
- ▶ Ethereum: run Turing-complete virtual machine logic in the blockchain to enable “smart” contracts and arbitrary applications, not just payments (is “Accelerando” an utopia or dystopia?)
- ▶ Polkadot: use side-chains to improve scalability

³ ≈ 1-15 minutes CPU time to create new transaction needed!

- ▶ Dogecoin: same as Bitcoin, just named after a dog meme (an idea that is obviously worth billions!)
- ▶ Zcash: uses ZKSNARKs³ to hide transactions (criminal activity on Bitcoin was too low)
- ▶ Ethereum: run Turing-complete virtual machine logic in the blockchain to enable “smart” contracts and arbitrary applications, not just payments (is “Accelerando” an utopia or dystopia?)
- ▶ Polkadot: use side-chains to improve scalability

³ ≈ 1-15 minutes CPU time to create new transaction needed!

1. Pumping and dumping is easier in illiquid markets. Thus, cryptocurrencies proliferate.
2. Tons of possible design variations: different puzzles, privacy, programmability, scalability
3. Proof-of-work is rather egalitarian; proof-of-stake more efficiently distributes wealth to those already wealthy by giving those with the biggest amount of coins the power to mine.
4. The elimination of private money issued by private banks and the centralization of credit at a national (central) bank was one of the core demands from the Communist Manifest (1848).

Blockchain Trilemma

Blockchains claim to achieve three properties:

- ▶ Decentralization: there are many participants, and each participant only needs to have a small amount of resources, say $O(c)$
- ▶ Scalability: the system scales to $O(n) > O(c)$ transactions
- ▶ Security: the system is secure against attackers with $O(n)$ resources

The Blockchain trilemma is that one can only have two of the three.

2025-01-06

Blockchains

└ Bitcoin and Payments: A good match?

└ Blockchain Trilemma

Blockchains claim to achieve three properties:

- ▶ Decentralization: there are many participants, and each participant only needs to have a small amount of resources, say $O(c)$
- ▶ Scalability: the system scales to $O(n) > O(c)$ transactions
- ▶ Security: the system is secure against attackers with $O(n)$ resources

The Blockchain trilemma is that one can only have two of the three.

1. Naturally, anyone trying to sell you some shitcoin is going to claim that their shitcoin satisfies all three.
2. In practice, they will have made some trade-off which could be more-or-less interesting.
3. Polkadot [1] has an interesting approach where they achieve scalability by offloading transactions onto side-chains (sharding) and security when the side-chains are merged back into the main chain.

James Mickens on Blockchains

James W. Mickens is an American computer scientist and the Gordon McKay Professor of Computer Science at Harvard John A. Paulson School of Engineering and Applied Sciences at Harvard University. His research focuses on distributed systems, such as large-scale services and ways to make them more secure.

At the Digital Initiative's Future Assembly on April 6, 2018, he presented "Blockchains Are a Bad Idea: More Specifically, Blockchains Are a Very Bad Idea."

2025-01-06

Blockchains

└─What are other applications for Blockchains?

└─James Mickens on Blockchains

James W. Mickens is an American computer scientist and the Gordon McKay Professor of Computer Science at Harvard John A. Paulson School of Engineering and Applied Sciences at Harvard University. His research focuses on distributed systems, such as large-scale services and ways to make them more secure.

At the Digital Initiative's Future Assembly on April 6, 2018, he presented "Blockchains Are a Bad Idea: More Specifically, Blockchains Are a Very Bad Idea."

1. So maybe payments are not actually a strength of Blockchains. But maybe there are other applications for Blockchains?
2. For this, we will now watch an opinionated talk by James Mickens.
3. He is probably qualified to speak on the subject, after all he explained at USENIX Security that a core tenet of technological manifest destiny is that history is uninteresting — which is key given that Blockchain proponents seem to have “forgotten” the historical issues that led to the creation of central banks. So what does he think of the Blockchain religion?

2025-01-06

Blockchains

└─ What are other applications for Blockchains?

<https://www.youtube.com/watch?v=15RTC22Z2xI> (2018)

<https://www.youtube.com/watch?v=15RTC22Z2xI> (2018)



Break

2025-01-06

Blockchains

└─ What are other applications for Blockchains?

Break

Security Goals for Time Stamping Services

- ▶ Document must have existed at the timestamp
- ▶ Modifications must be detected
- ▶ Document must have been created after the timestamp
- ▶ Validation of timestamp proof possible forever
- ▶ Non-repudiation
- ▶ No trusted third party (see [2, 6] for protocols with trusted third party)
- ▶ Availability

2025-01-06

Blockchains

└─ What are other applications for Blockchains?

└─ Security Goals for Time Stamping Services

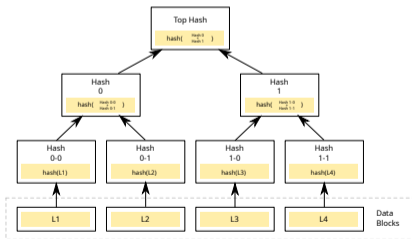
- ▶ Document must have existed at the timestamp
- ▶ Modifications must be detected
- ▶ Document must have been created after the timestamp
- ▶ Validation of timestamp proof possible forever
- ▶ Non-repudiation
- ▶ No trusted third party (see [2, 6] for protocols with trusted third party)
- ▶ Availability

1. Applications for time stamping include establishing ownership, for example of prior art in patent or copyright disputes.
2. Timestamping services using a trusted third party have existed online and offline for a very long time.
3. But as always, we do not like trusted third parties in information security.

Blockchain-based Time Stamping Services

- ▶ <https://originastamp.com/>: Bitcoin&Ethereum, 100 timestamps \$10
- ▶ <https://blockchainsign.io/>: Ethereum, 1 timestamp \$5
- ▶ <https://guardtime.com/>: private KSI Blockchain (!?)

Key idea:



2025-01-06

Blockchains

└─ What are other applications for Blockchains?

└─ Blockchain-based Time Stamping Services

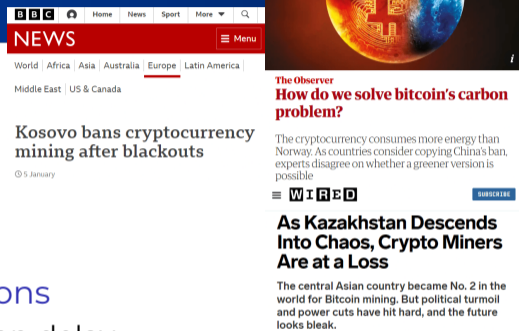
- ▶ <https://originastamp.com/>: Bitcoin&Ethereum, 100 timestamps \$10
- ▶ <https://blockchainsign.io/>: Ethereum, 1 timestamp \$5
- ▶ <https://guardtime.com/>: private KSI Blockchain (!?)



1. There are various commercial providers offering timestamping solutions using Blockchains.
2. They differ in pricing and Blockchains used. I do not see a tangible benefit of using a private Blockchain over just using some trusted third parties.
3. As putting data onto a Blockchain is expensive, the various documents to be timestamped are hashed, and the hashes combined with each other in a Merkle tree.
4. The root of the Merkle tree is then included in a block on the Blockchain.

Bonus: Depolymerization

Blockchain based cryptocurrencies



Biggest cryptocurrencies

- ▶ **BTC** Bitcoin
- ▶ **ETH** Ethereum

Common blockchain limitations

- ▶ **Delay** block and confirmation delay
- ▶ **Cost** transaction fees
- ▶ **Scalability** limited amount of transaction per second
- ▶ **Ecological impact** computation redundancy
- ▶ **Privacy & regulatory compliance**

2025-01-06

Blockchains

- └ Bonus: Depolymerization [5]

- └ Blockchain based cryptocurrencies

1. The blockchain trilemma is not the only dilemma for cryptocurrencies.
2. We will now look at some practical approaches for addressing or lessening some of these issues.



Related work

Centralization - Coinbase off-chain sending

- + Fast and cheap: off chain transaction
- Trust in Coinbase: privacy, security & transparency

Layering - Lightning Network

- + Fast and cheap: off-chain transactions
- Requires setting up bidirectional payment channels
- Fraud attempts are mitigated via a complex penalty system

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Related work

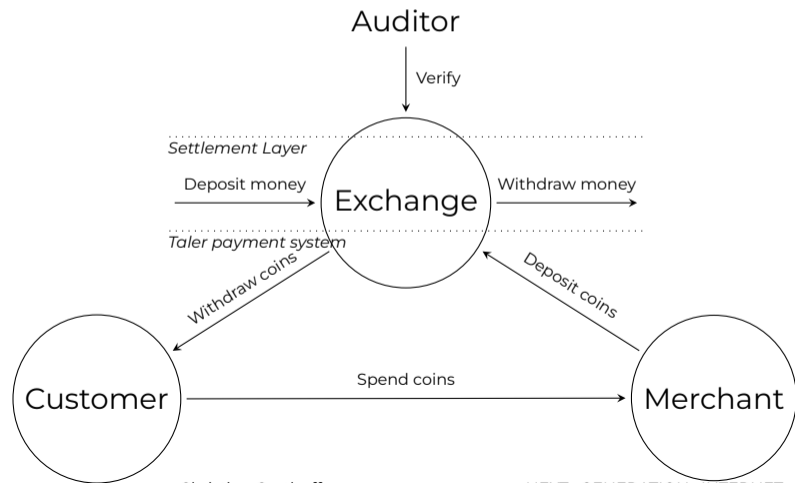
Centralization - Coinbase off-chain sending

- + Fast and cheap: off chain transaction
- Trust in Coinbase: privacy, security & transparency

Layering - Lightning Network

- + Fast and cheap: off-chain transactions
- Requires setting up bidirectional payment channels
- Fraud attempts are mitigated via a complex penalty system

1. One solution is to trade crypto-currencies off-chain. This involves creating an account at a “trusted” intermediary such as FTX and hoping that they do not embezzle the funds. Providers pick jurisdictions favorable to them.
2. In light of the blockchain trilemma, this is the solution that basically **only** offers performance and no cryptographic security or decentralization.
3. The Lightning network creates a layer-2 network over Bitcoin where pairs of nodes can perform fast off-chain transactions over payment channels. Opening a payment channel requires locking up funds, and payments are limited to the amount locked up by both parties. When the channel is closed, the final delta between the accounts is transferred on-chain.
4. The main issue is the requirement to lock up funds (limiting the availability of effective payment channels), and as a result lighting is fast and cheap if it works, but for some payments it may simply not work at all because no route with adequate capacity exists between payer and payee!



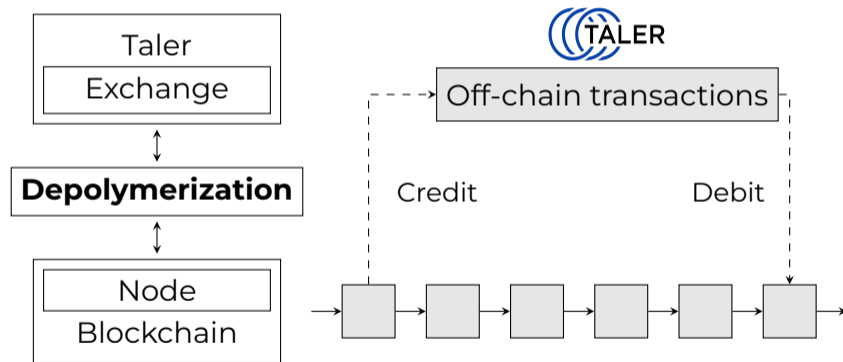
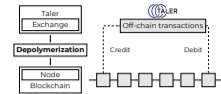
1. This is a review of the Taler architecture.
2. Key for Taler is the **existence** of a settlement layer. Usually, this is some existing wholesale payment system, such as SEPA, UPC or SWIFT. This is the “core banking system” of the respective fiat currency used by banks to make transactions.
3. But, Taler is not limited to traditional core banking systems and fiat currencies!

Project Depolymerization

Taler with blockchain settlement layer

2025-01-06

Blockchains
└ Bonus: Depolymerization [5]
└ Project Depolymerization



1. A polymer, commonly called plastic, is a long-chained molecule created by the process of polymerization. For example, Poly-Ethylen (PE) is created from many Ethylen molecules.
2. The process of **depolymerization** is when we recycle plastic trash into monomers, usually by applying heat.
3. Project depolymerization turns long chains of blocks into “unlinked” digital coins useful for high-speed transactions. The digital coins can eventually be put back onto the blockchain.
4. As always, GNU Taler assumes that the operator of the Taler exchange is trustworthy because they are easily identified and must be regulated and independently audited.

Depolymerization [5]

Architecture



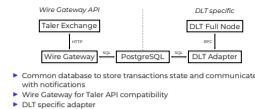
- ▶ Common database to store transactions state and communicate with notifications
- ▶ Wire Gateway for Taler API compatibility
- ▶ DLT specific adapter

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

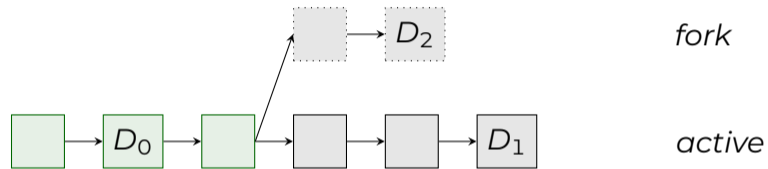
└ Depolymerization [5]



1. This is a high-level overview of the Depolymerization software architecture.
2. We use a generic wire gateway to implement the GNU Taler wire gateway REST interface, a generic database to store transaction data, and then a blockchain-specific adapter that talks to a full node over some blockchain-specific RPC API.
3. This is about 12k LOC in Rust for both Bitcoin and Ethereum, with the blockchain-specific code being around 2k LOC for each blockchain.
4. We will now take a brief look at key challenges involved in implementing this.

CAP & Bitcoin

Chain reorganization



Bitcoin is inconsistent:

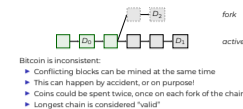
- ▶ Conflicting blocks can be mined at the same time
- ▶ This can happen by accident, or on purpose!
- ▶ Coins could be spent twice, once on each fork of the chain!
- ▶ Longest chain is considered "valid"

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ CAP & Bitcoin

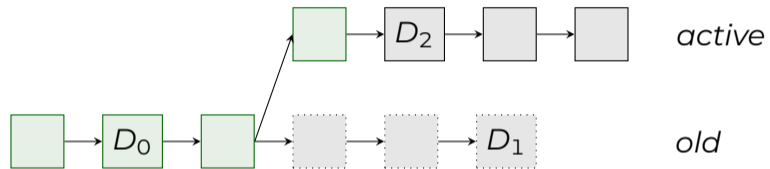


Bitcoin is inconsistent:

- ▶ Conflicting blocks can be mined at the same time
- ▶ This can happen by accident, or on purpose!
- ▶ Coins could be spent twice, once on each fork of the chain!
- ▶ Longest chain is considered "valid"

1. A fork is when concurrent blockchain states coexist. Nodes will follow the longest chain, replacing recent blocks if necessary during a blockchain reorganization.
2. Basically, this raises the question as to when Depolymerizer can be sure that an inbound transaction is actually final. Original Bitcoin paper suggests to consider transaction confirmed only after at least 6 blocks past the transaction, but competitively long alternative chains could void durability even after 6 blocks!
3. Once Depolymerizer issues blindly signed anonymous digital coins there is no good way to "undo" this, so we need to be sure that the money did arrive in our escrow account: If a deposit transaction disappears from the blockchain, coins created from **final** Taler withdraw transactions might no longer be backed by credit!

Handling blockchain reorganization



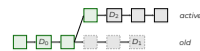
- ▶ As small reorganizations are common, Satoshi already recommended to apply a confirmation delay to handle most disturbances and attacks.
- ▶ If a reorganization longer than the confirmation delay happens, but it did not remove credits, Depolymerizer is safe and keeps running.

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

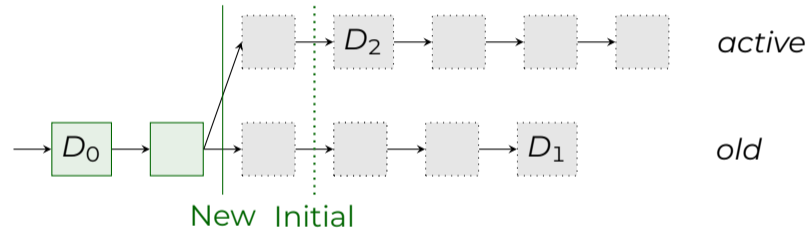
└ Handling blockchain reorganization



- ▶ As small reorganizations are common, Satoshi already recommended to apply a confirmation delay to handle most disturbances and attacks.
- ▶ If a reorganization longer than the confirmation delay happens, but it did not remove credits, Depolymerizer is safe and keeps running.

1. Thus, first of all, Depolymerizer also waits for ≥ 6 blocks before considering an inbound Bitcoin transaction to be “final”.
2. Second, if a fork is detected **despite** waiting 6 blocks, we check if this actually affected our balance. Not all forks will be **relevant**.

Adaptive confirmation



2025-01-06

Blockchains

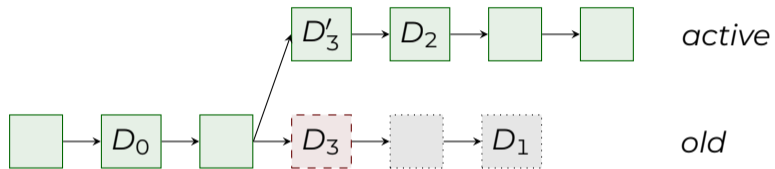
└ Bonus: Depolymerization [5]

└ Adaptive confirmation



1. Nevertheless, if we experience any such reorganization once, its dangerously likely for another one of a similar scope to happen again. Depolymerizer learns from reorganizations by increasing its confirmation delay.
2. In the previous slides, we used a confirmation delay of 3 blocks, but after experiencing a successful fork after 4 blocks, the confirmation delay would be increased to 5 blocks.
3. Of course in practice the minimum would be ≥ 6 blocks.

Handling blockchain reorganization



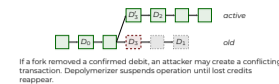
If a fork removed a confirmed debit, an attacker may create a conflicting transaction. Depolymerizer suspends operation until lost credits reappear.

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Handling blockchain reorganization



1. Now, imagine we had a confirmation delay of only two blocks, and originally accepted D_3 when the "old" chain was the longest chain. When the "fork" became the new "active" (longest) chain an incoming transfer to the depolymerizer might disappear.
2. If Depolymerizer issued coins for a blockchain transaction that then "disappeared", Depolymerizer stop all processing until either the administrator manually intervenes or the transaction appears again on-chain (as it *should* still be in the pool of transactions yet to be mined, unless there is now a hard conflict).

Challenges

Taler Metadata

- ▶ Metadata are required to link a wallet to credits and allow merchant to link deposits to debits
- ▶ Putting metadata in blockchain transactions can be tricky

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Challenges

Taler Metadata

- ▶ Metadata are required to link a wallet to credits and allow merchant to link deposits to debits
- ▶ Putting metadata in blockchain transactions can be tricky

1. To determine the wallet eligible to withdraw funds, GNU Taler requires a wallet-specific **reserve public key** to be encoded in the wire transfer subject.
2. However, most blockchains lack the ability to encode such meta-data nicely on-chain.

Storing metadata

Bitcoin

Bitcoin - Credit

- ▶ Transactions from code
- ▶ Only 32B + URI
- ▶ **OP_RETURN**

Bitcoin - Debit

- ▶ Transactions from common wallet software
- ▶ Only 32B
- ▶ **Fake Segwit Addresses**

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Storing metadata

1. When doing outgoing wire transfers, Depolymerizer can encode meta-data using **OP-RETURN**.
2. However, users cannot generate such transactions with contemporary (Blockchain) wallets.
3. Thus, for **incoming** transactions, we use fake Segwit addresses. Basically, the transaction must contain three outputs, the amount to be withdrawn must be sent to the Deploymerizer's Bitcoin wallet address, and two additional outputs (with nominal amounts) must go two "fake" Bitcoin wallet addresses which actually encode the reserve public key. The (few) Satoshis involved in the transfers to the fake addresses are burned.

Bitcoin - Credit

- ▶ Transactions from code
- ▶ Only 32B + URI
- ▶ **OP_RETURN**

Bitcoin - Debit

- ▶ Transactions from common wallet software
- ▶ Only 32B
- ▶ **Fake Segwit Addresses**

Storing metadata

Ethereum

Smart contract?

- ▶ Logs in smart contract is the recommend way (ethereum.org)
- ▶ Expensive (additional storage and execution fees)
- ▶ Avoidable attack surface (error prone)

Custom input format

Use input data in transactions, usually used to call smart contract, to store our metadata.

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Storing metadata

1. Ethereum recommends encoding meta-data in smart contracts. But, this is actually more expensive as the result is large and comes with execution (Gas) fees. The complexity also makes it fragile.
2. Depolymerizer instead uses a custom input format.

Smart contract?

- ▶ Logs in smart contract is the recommend way
- ▶ Expensive (additional storage and execution fees)
- ▶ Avoidable attack surface (error prone)

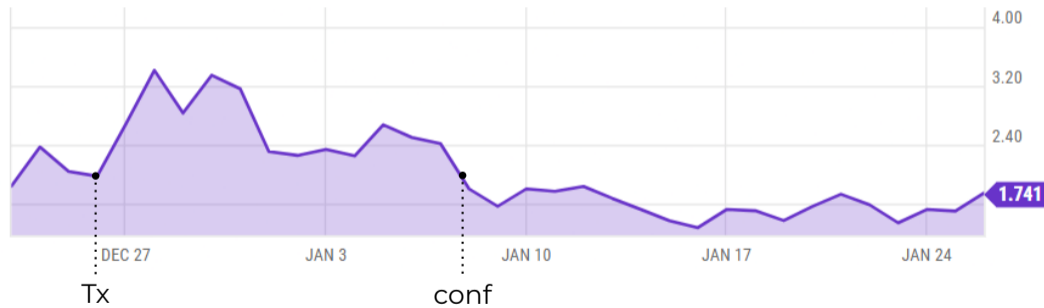
Custom input format

Use input data in transactions, usually used to call smart contract, to store our metadata.

Blockchain challenges

Transactions stuck in mempool

We want confirmed debits within a limited time frame.



2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Blockchain challenges

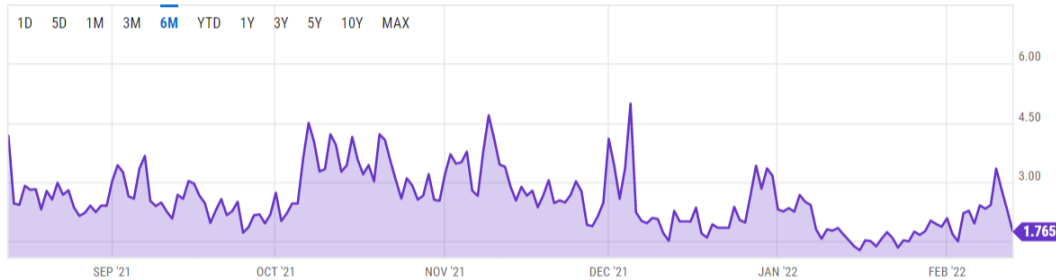
We want confirmed debits within a limited time frame.



1. When we trigger a debit with a fee too small, it may not be confirmed in a timely fashion.
2. If we picked the current transaction cost at time Tx and the minimum cost for inclusion in a block goes up afterwards, it would take until time "conf" for the transaction to be actually included in a block and thus confirmed. And that is assuming it even stays in the mempool for this long.

Blockchain challenges

Transactions stuck in mempool



Bitcoin average transaction fee over 6 months (ychart)

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Blockchain challenges



1. However, transaction fees are unpredictable as shown in the historical chart.
2. Offering high fees to miners would significantly increase transaction costs, while using low fees risks transactions not being mined in a reasonable amount of time.
3. Depolymerizer thus **monitors** stuck transactions, and if necessary increases the transaction fees paid to miners if transactions are stuck for too long.

Future work

- ▶ Support other blockchains
- ▶ Universal auditability, using sharded transactions history
- ▶ Multisig by multiple operators for transactions validation

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Future work

- ▶ Support other blockchains
- ▶ Universal auditability, using sharded transactions history
- ▶ Multisig by multiple operators for transactions validation

1. It might be interesting to see what challenges and solutions apply to other blockchains.
2. Right now, a Taler auditor always requires a full copy of an exchange database, which would not scale if “everybody” wanted to participate in an audit. Universal auditability isKwould about allowing everyone to check that an exchange is operating correctly. With public blockchains, at least the underlying settlement layer data is already public!
3. Still, this would only detect problems **after** an incident. Requiring multiple signatures for on-chain transactions from independent operators sharing access to the escrow account would eliminate the single point of failure.

Conclusion

Blockchains can be used as a settlement layer for GNU Taler with Depolymerizer.

- Trust exchange operator or auditors
- + Fast and cheap
- + Realtime, ms latency
- + Linear scalability
- + Ecological
- + Privacy when it can, transparency when it must (avoid tax evasion and money laundering)

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Conclusion

Blockchains can be used as a settlement layer for GNU Taler with Depolymerizer.

- Trust exchange operator or auditors
- + Fast and cheap
- + Realtime, ms latency
- + Linear scalability
- + Ecological
- + Privacy when it can, transparency when it must (avoid tax evasion and money laundering)

1. Note that this approach does not address the ecological footprint of the underlying blockchain, it only addresses the problem for off-chain transactions.
2. It also does not prevent money laundering and criminal abuse on the underlying blockchain, only the off-chain part **could be made compliant**.
3. Making the off-chain part compliant is still a huge challenge, as the operator would also have to validate the legality of the **source of funds** for all incoming transactions on the blockchain, and it is not easy to do that well.

Security Goals for Name Systems

- ▶ Query origin anonymity
- ▶ Data origin authentication and integrity protection
- ▶ Zone confidentiality
- ▶ Query and response privacy
- ▶ Censorship resistance
- ▶ Traffic amplification resistance
- ▶ Availability

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Security Goals for Name Systems

- ▶ Query origin anonymity
- ▶ Data origin authentication and integrity protection
- ▶ Zone confidentiality
- ▶ Query and response privacy
- ▶ Censorship resistance
- ▶ Traffic amplification resistance
- ▶ Availability

1. QOA: We do not know who asked for name resolution. In traditional DNS, the origin is only exposed to the ISP running the recursive resolver.
2. DOA: We want to make sure that the answer is correct.
3. ZC: The zone publisher does not want all records to be public, only given the label one should be able to determine the record set. Not all labels are public.
4. QRP: The name resolved (question) and the record set returned (answer) are themselves private and not disclosed to the infrastructure.
5. CR: Authorities cannot selectively block access to some record sets.
6. TAR: The system cannot be abused as a traffic multiplier for DDoS attacks.
7. A: New zones can be added, and new record sets published and resolved.

Approaches Adding Cryptography to DNS

- ▶ DNSSEC
- ▶ DNSCurve
- ▶ DNS-over-TLS (DoT)
- ▶ DNS-over-HTTPS (DoH)
- ▶ RAINS
- ▶ GNU Name System (GNS)

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Approaches Adding Cryptography to DNS

- ▶ DNSSEC
- ▶ DNSCurve
- ▶ DNS-over-TLS (DoT)
- ▶ DNS-over-HTTPS (DoH)
- ▶ RAINS
- ▶ GNU Name System (GNS)

1. These are all mostly cryptographic proposals making different political trade-offs.
2. Except for GNS, they all assume some ICANN-like authority to manage the root zone.
3. Even GNS *supports* (and would benefit from) trustworthy registration authorities.
4. But can we do without a trusted third party like ICANN?

Namecoin

No need for a trusted third party: put the records into the Blockchain!

Or rather, put the public key of the owner and signed updates into it.

Plus, expiration rules.

2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Namecoin

No need for a trusted third party: put the records into the Blockchain!

Or rather, put the public key of the owner and signed updates into it.

Plus, expiration rules.

1. And of course use our new **utility token** to pay for registration.
2. Big issues: trademark infringement, use by malware, etc.

Ethereum Name System⁴

Let's have a smart contract in the Blockchain manage naming!

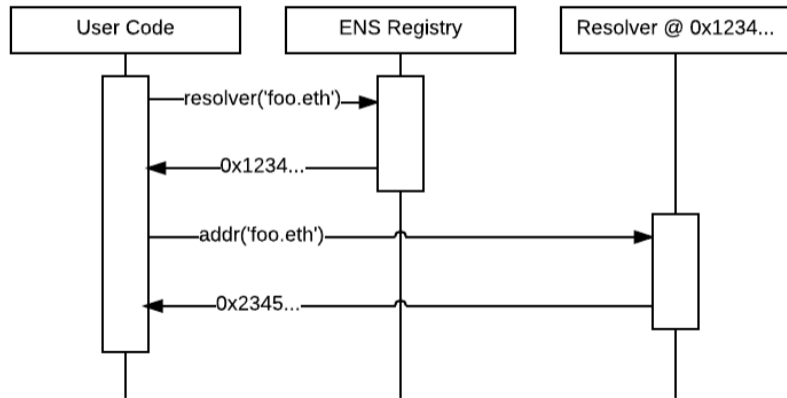
Blockchain contains smart contract and data who controls which name.

Contract allocates names under .eth using auctions.

⁴<https://ens.domains/>

1. Why write another Blockchain? Ethereum is programmable!

Ethereum Name System⁶

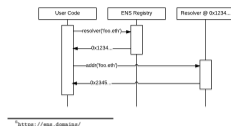


2025-01-06

Blockchains

└ Bonus: Depolymerization [5]

└ Ethereum Name System^a



1. Actual lookup is indirect: first lookup resolver code
2. Then ask resolver code for actual resolution result(s)
3. Issue: Ethereum chain is **huge**

Handshake Name System⁸

Incremental improvements over Namecoin and ENS:

- ▶ New blockchain with “HNS” utility tokens
- ▶ Compact proofs: resolvers do not need the full chain
- ▶ Pre-reserved names (ICANN TLDs, top-100k Alexa domains)
- ▶ Air-drop to “stakeholders” to boost adoption



⁸<https://handshake.org/>

1. Cryptographic improvement: secure name resolution without full chain
2. Social consideration: do not allow anybody to register any name immediately, instead give existing DNS-owners grace period to register “\$TRADEMARK.hns”.
3. Presumes trademark owners care enough about potential success of Handshake to spend money to buy HNS and register trademarks in yet another (unofficial) TLD.

References I

-  Hanaa Abbas, Maurantonio Caprolu, and Roberto Di Pietro. Analysis of polkadot: Architecture, internals, and contradictions. In *2022 IEEE International Conference on Blockchain (Blockchain)*, pages 61–70, 2022.
-  C. Adams, P. Cain, D. Pinkas, and R. Zuccherato. Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP). RFC 3161 (Proposed Standard), August 2001. Updated by RFC 5816.

References II

-  David Chaum, Christian Grothoff, and Thomas Moser.
How to issue a central bank digital currency.
In *SNB Working Papers*, number 2021-3. Swiss National Bank,
February 2021.
-  Alexandra Dirksen.
A blockchain picture book.
https://media.ccc.de/v/35c3-9573-a_blockchain_picture_book, 12
2018.

2025-01-06

Blockchains
└─References

└─References

 David Chaum, Christian Grothoff, and Thomas Moser.
How to issue a central bank digital currency.
In *SNB Working Papers*, number 2021-3. Swiss National Bank,
February 2021.

 Alexandra Dirksen.
A blockchain picture book.
https://media.ccc.de/v/35c3-9573-a_blockchain_picture_book, 12
2018.

References III

-  Antoine d'Aligny, Emmanuel Benoist, and Christian Grothoff.
Project depolymerization: Tokenization of blockchains.
In 4th Conference on Blockchain Research and Applications for Innovative Networks and Services, September 2022.
-  D. Pinkas, N. Pope, and J. Ross.
Policy Requirements for Time-Stamping Authorities (TSAs).
RFC 3628 (Informational), November 2003.

2025-01-06

Blockchains
└─References

└─References

 Antoine d'Aligny, Emmanuel Benoist, and Christian Grothoff.
Project depolymerization: Tokenization of blockchains.
In 4th Conference on Blockchain Research and Applications for Innovative Networks and Services, September 2022.

 D. Pinkas, N. Pope, and J. Ross.
Policy Requirements for Time-Stamping Authorities (TSAs).
RFC 3628 (Informational), November 2003.

Acknowledgements

Co-funded by the European Union (Project 101135475).




Co-funded by
the European Union

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union.

Neither the European Union nor the granting authority can be held responsible for them.

Co-funded by SERI (HEU-Projekt 101135475-TALER).

Project funded by

 Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Swiss Confederation

Federal Department of Economic Affairs,
Education and Research EAER
**State Secretariat for Education,
Research and Innovation SERI**

2025-01-06

Blockchains
References

Acknowledgements

Co-funded by the European Union (Project 101135475)



Co-funded by
the European Union

Co-funded by SERI (HEU-Projekt 101135475-TALER)

Project funded by

 Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

Federal Department of Economic Affairs,
Education and Research EAER
**State Secretariat for Education,
Research and Innovation SERI**

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union.
Neither the European Union nor the granting authority can be held responsible for them.