



Blockchain: Your Questions. Our Answers.

Eliezer Kanal & Gabriel Somlo

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15213



Copyright 2018 Carnegie Mellon University. All Rights Reserved.

This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

The view, opinions, and/or findings contained in this material are those of the author(s) and should not be construed as an official Government position, policy, or decision, unless designated by other documentation.

References herein to any specific commercial product, process, or service by trade name, trade mark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by Carnegie Mellon University or its Software Engineering Institute.

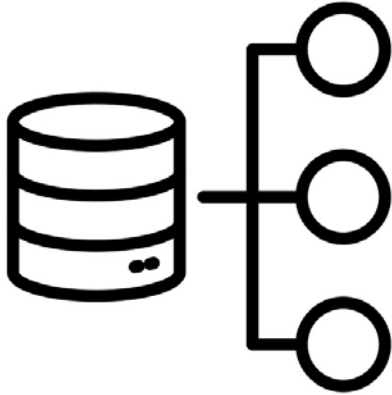
NO WARRANTY. THIS CARNEGIE MELLON UNIVERSITY AND SOFTWARE ENGINEERING INSTITUTE MATERIAL IS FURNISHED ON AN "AS-IS" BASIS. CARNEGIE MELLON UNIVERSITY MAKES NO WARRANTIES OF ANY KIND, EITHER EXPRESSED OR IMPLIED, AS TO ANY MATTER INCLUDING, BUT NOT LIMITED TO, WARRANTY OF FITNESS FOR PURPOSE OR MERCHANTABILITY, EXCLUSIVITY, OR RESULTS OBTAINED FROM USE OF THE MATERIAL. CARNEGIE MELLON UNIVERSITY DOES NOT MAKE ANY WARRANTY OF ANY KIND WITH RESPECT TO FREEDOM FROM PATENT, TRADEMARK, OR COPYRIGHT INFRINGEMENT.

[DISTRIBUTION STATEMENT A] This material has been approved for public release and unlimited distribution. Please see Copyright notice for non-US Government use and distribution.

This material may be reproduced in its entirety, without modification, and freely distributed in written or electronic form without requesting formal permission. Permission is required for any other use. Requests for permission should be directed to the Software Engineering Institute at permission@sei.cmu.edu.

Carnegie Mellon® and CERT® are registered in the U.S. Patent and Trademark Office by Carnegie Mellon University. DM18-0583

Previous models of computing

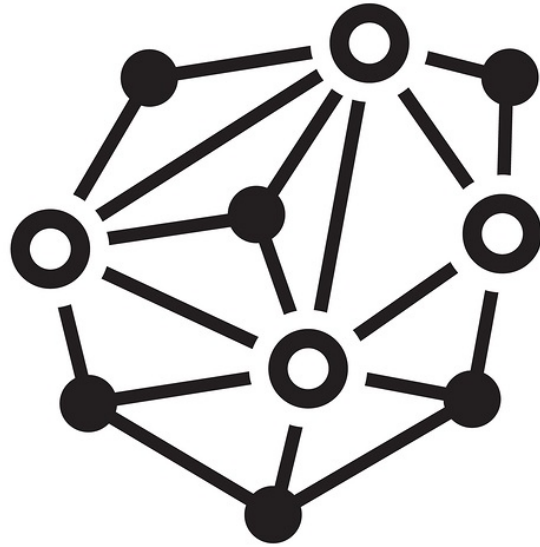


Data Storage:
Database



Program Execution:
Local

Blockchain



Data Storage:
Blockchain

Program Execution:
Blockchain

Blockchain Properties

Data on the chain cannot be removed

Identity fundamentally linked to activity

Easily auditable

Mediates untrusted party interactions





Classic Currency: Store of Value

- A \$100 bill “stores” a \$100 value
- My checking account “stores” a \$148.23 balance
- If I pay Adam \$48.23, there’s an atomic transaction:



begin atomic

Gabriel.Checking -= \$48.23;

Adam.Checking += \$48.23;

end atomic

FIRST BANK OF WIKI
1425 JAMES ST. P.O. BOX 4000
VICTORIA BC V8K 3K4 1-800-555-5555

CHEQUING ACCOUNT STATEMENT
Page: 1 of 1

JOHN JONES
1613 DUNDAS ST W ART ST
TORONTO, ON M6K 1Y2

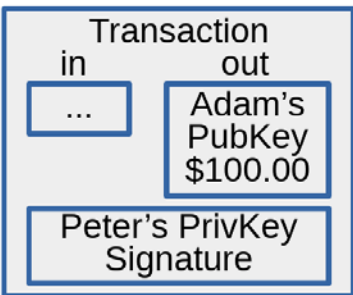
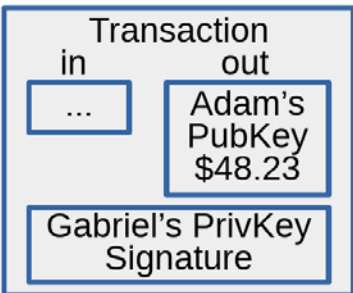
Statement period: 2003-10-09 to 2003-11-08
Account No.: 69335-123-459-7

| Date | Description | Ref. | Withdrawals | Deposits | Balance |
|----------------|-----------------------------------|------|-------------|----------|---------|
| 2003-10-08 | Previous balance | | | | 0.55 |
| 2003-10-14 | Payroll Deposit - HOTEL | | | 694.81 | 695.36 |
| 2003-10-14 | Web Bill Payment - MASTERCARD | 0085 | 200.00 | | 495.36 |
| 2003-10-16 | ATM Withdrawal - INTERAC | 3000 | 21.25 | | 474.11 |
| 2003-10-16 | Fees - Interac | | 1.50 | | 472.61 |
| 2003-10-20 | Interac Purchase - ELECTRONICS | 1075 | 2.99 | | 469.62 |
| 2003-10-21 | Web Bill Payment - ADME X | 3314 | 300.00 | | 169.62 |
| 2003-10-22 | ATM Withdrawal - FIRST BANK | 3054 | 100.00 | | 69.62 |
| 2003-10-23 | Interac Purchase - SUPERMARKET | 1559 | 29.08 | | 40.54 |
| 2003-10-24 | Interac Refund - ELECTRONICS | 1075 | | 2.99 | 43.53 |
| 2003-10-27 | Telephone Bill Payment - VISA | 2475 | 6.77 | | 36.76 |
| 2003-10-28 | Payroll Deposit - HOTEL | | | 694.81 | 721.57 |
| 2003-10-30 | Web Funds Transfer - From SAVINGS | 2020 | | 50.00 | 771.57 |
| 2003-11-03 | Pro-Subj. Payment - INSURANCE | | 23.55 | | 748.02 |
| 2003-11-03 | Cheque No. - 409 | | 100.00 | | 648.02 |
| 2003-11-05 | Mortgage Payment | | 710.49 | | -62.47 |
| 2003-11-07 | Fees - Chevrolet | | 5.00 | | -67.47 |
| 2003-11-08 | Fees - Monthly | | 5.00 | | -72.47 |
| *** Totals *** | | | 1,515.63 | 1,442.91 | |

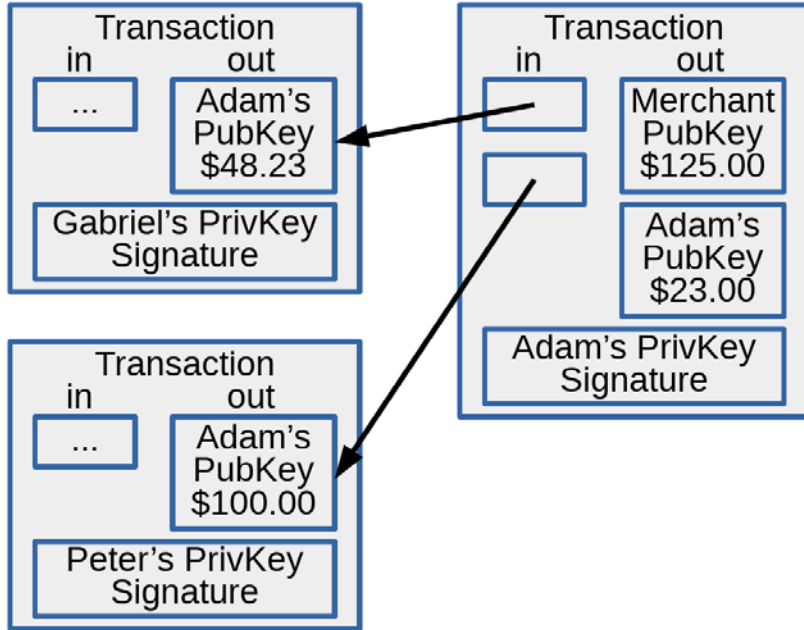
Cryptocoins: IOUs

- Gabriel owes Adam \$48.23
- Peter owes Adam \$100.00
 - Therefore, Adam “has” \$148.23
 - Assuming IOUs collected instantly, on demand!
- To pay for something, Adam must:
 - Collect (some of) his IOUs
 - Issue a fresh IOU to the payee/merchant
- IOUs (a.k.a. *Transactions*) passed around by nodes of a distributed, P2P network

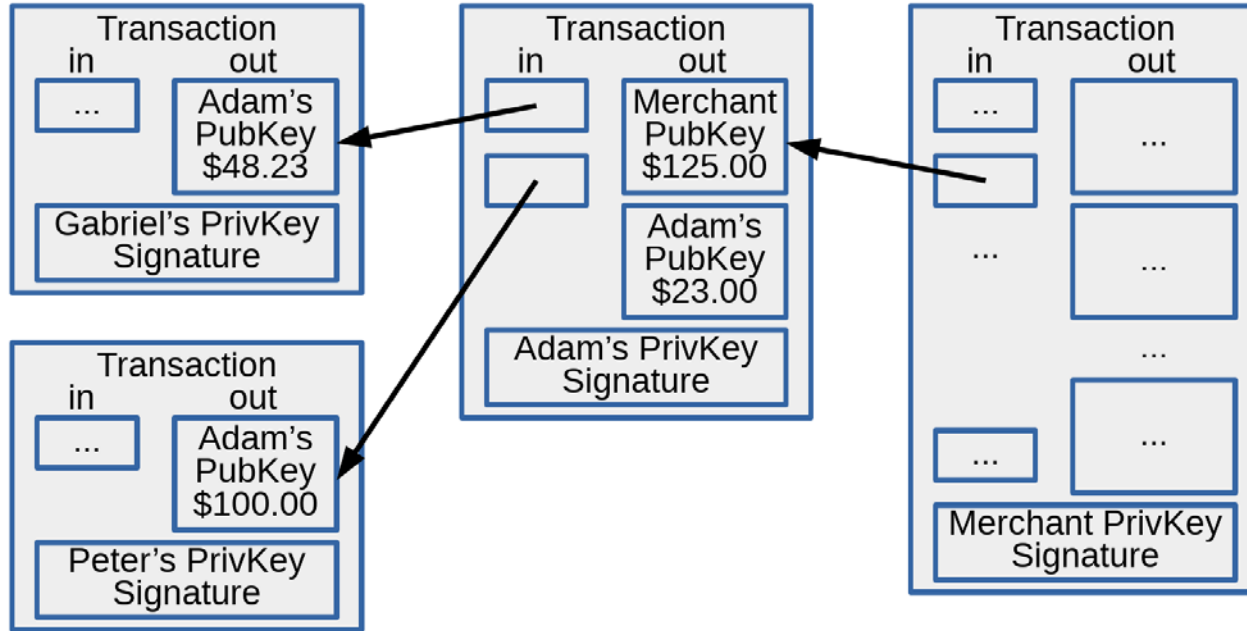
Transactions



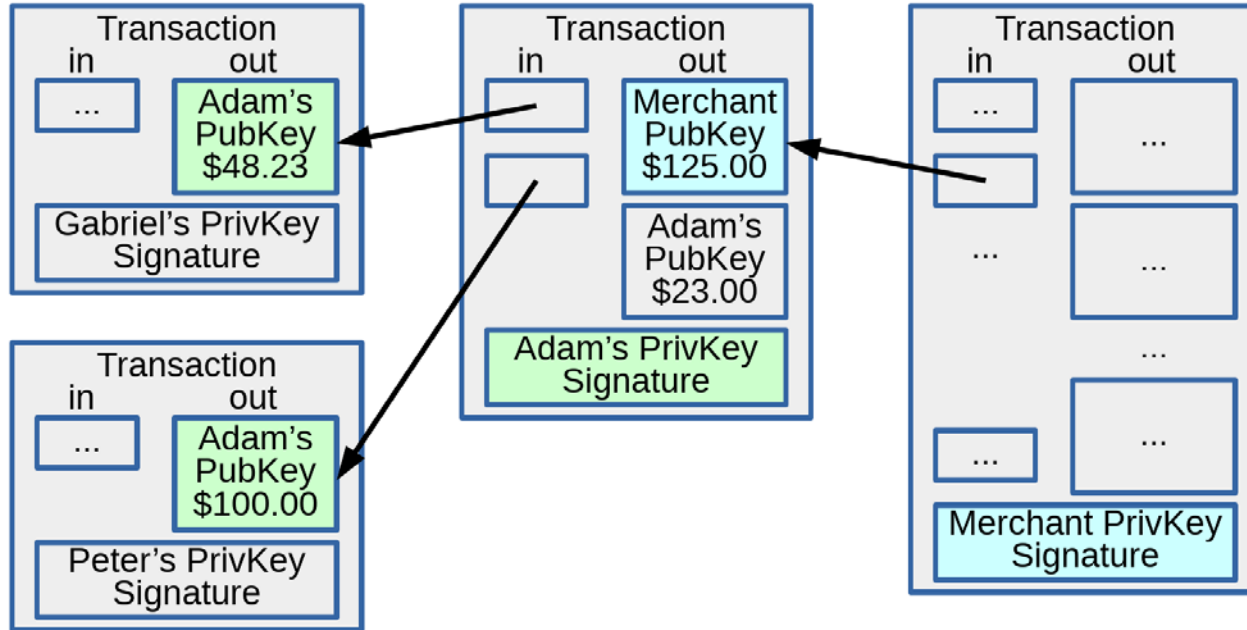
Transactions



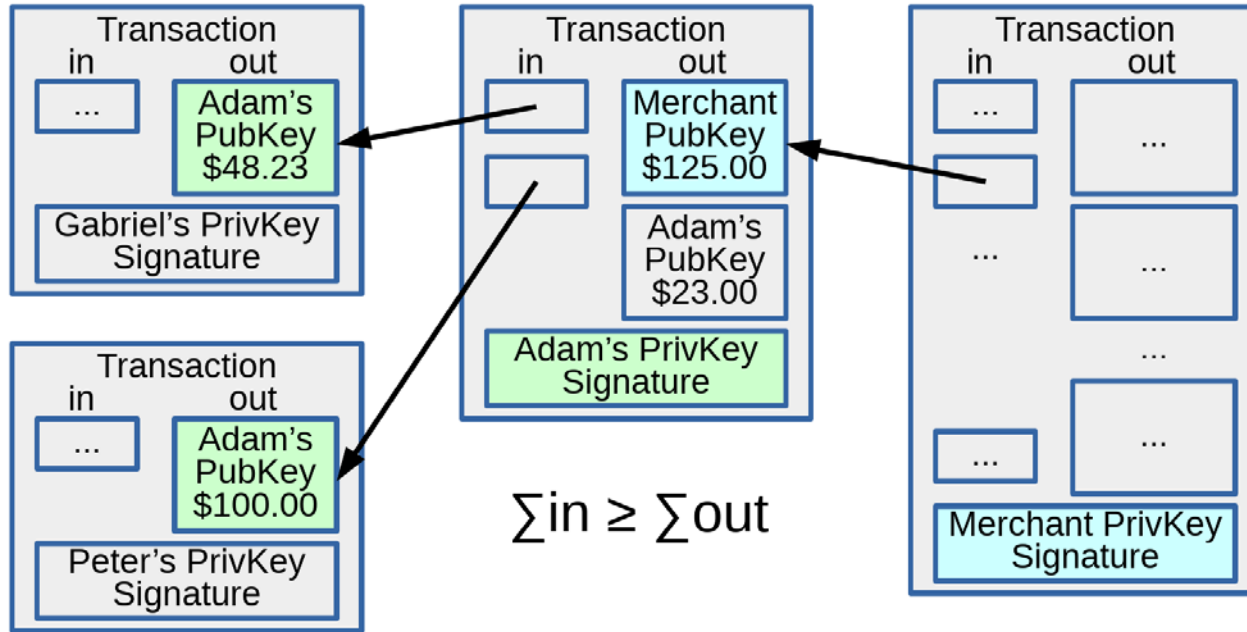
Transactions



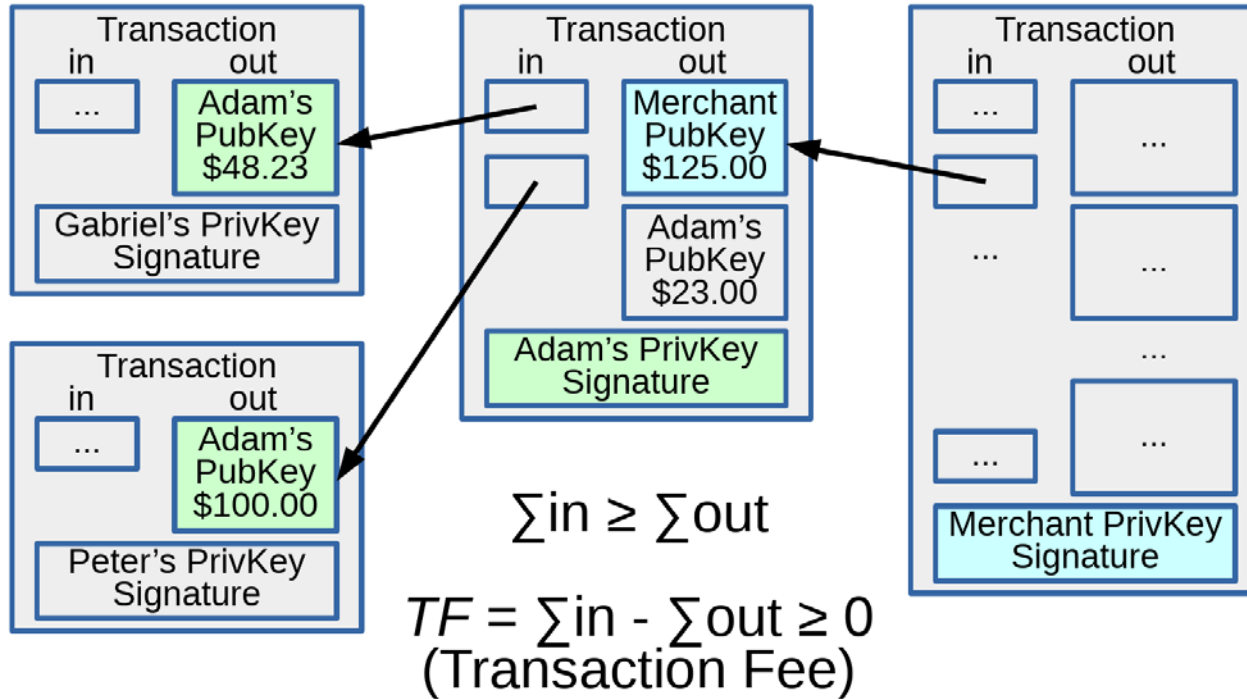
Transactions: Identity of Parties



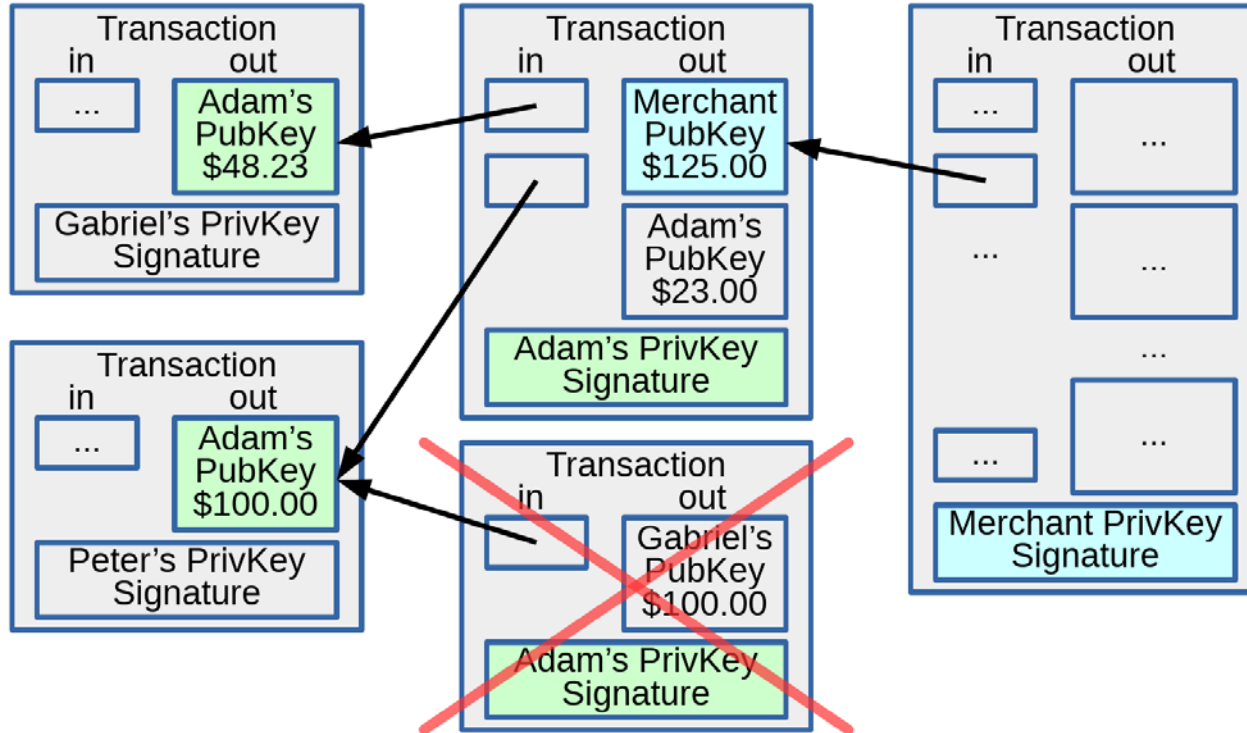
Transactions: No Overspending!



Transactions: No Overspending!

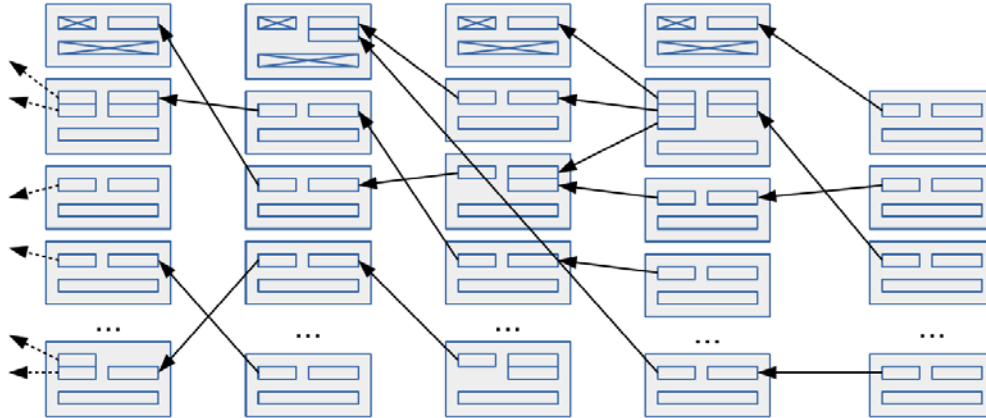


Transactions: No Double-Spending!



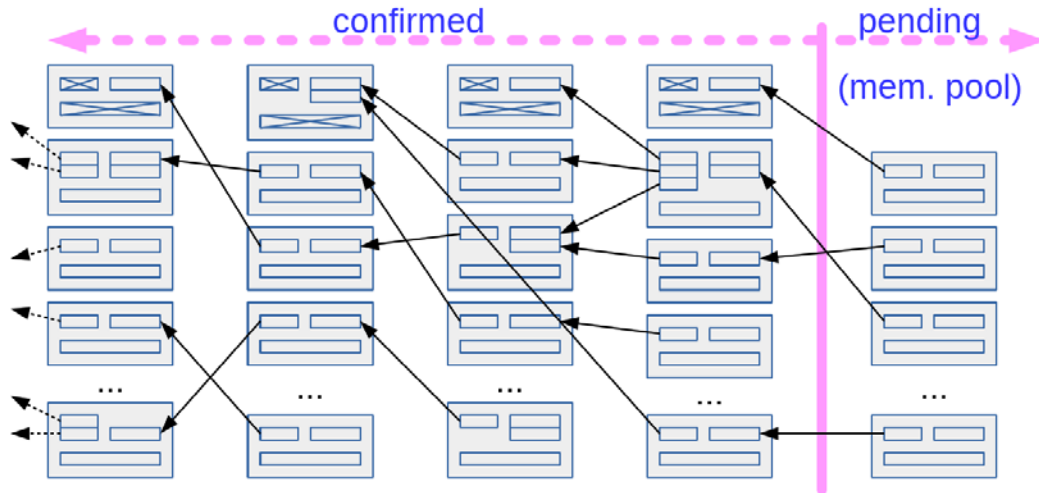
Ledger

- DAG of *all* transactions *ever* issued
 - Append-only data structure
- Every peer node maintains a copy



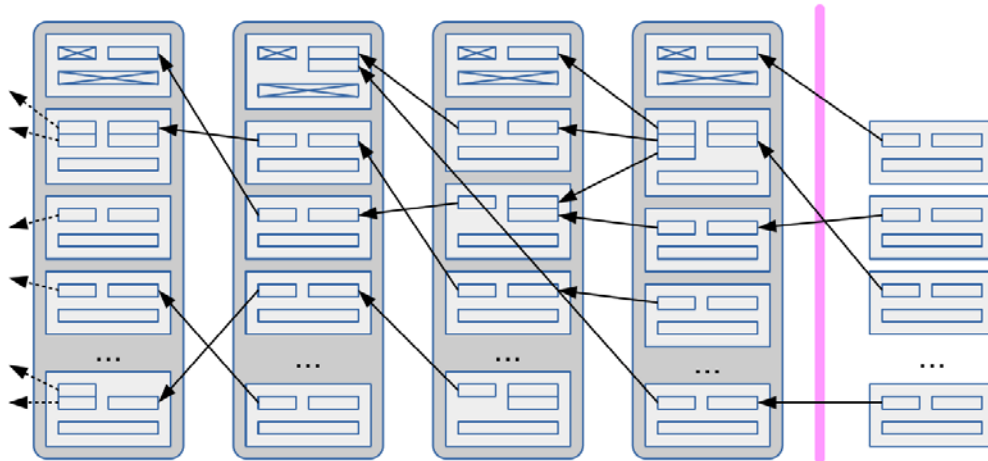
Ledger

- Existing (*confirmed*) transactions on HDD
- New (*pending*) transactions in Memory Pool
 - Must be valid w.r.t. existing state to be confirmed



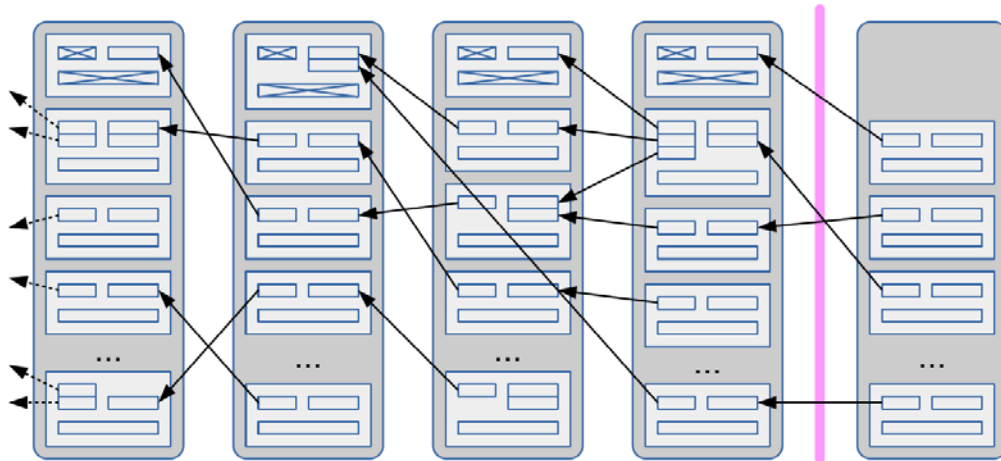
Transaction Blocks

- Confirmed transactions grouped in *blocks*



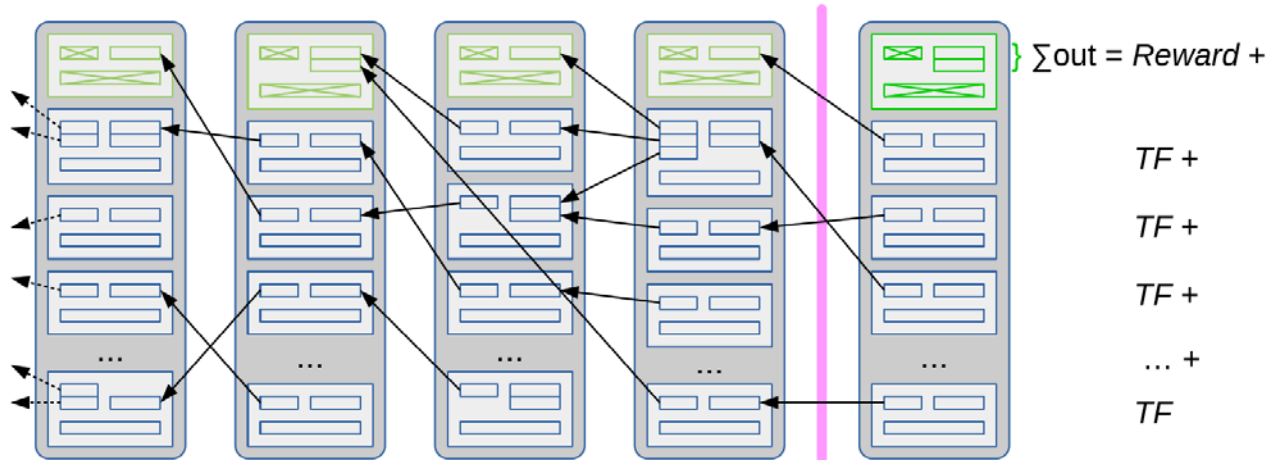
Transaction Blocks

- Confirmed transactions grouped in *blocks*
- Peers (*miners*) **compete** to create newest block
 - Containing newly validated (confirmed) transactions

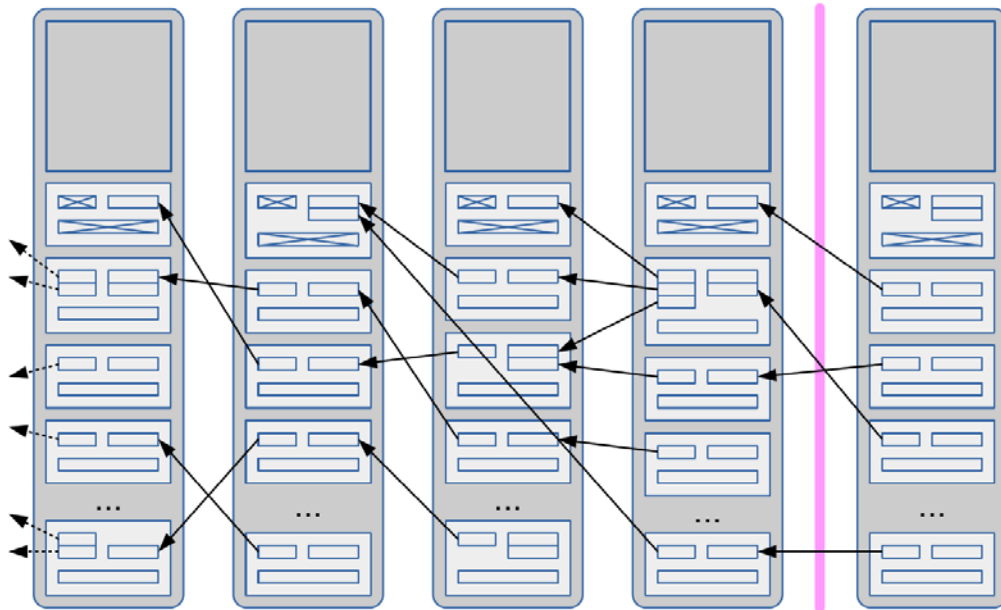


Coinbase Transactions

- Compensate miners for “community service” work
 - i.e., confirming users’ pending transactions
- *Reward* (freshly “minted” money)
 - Also *transaction fees* from each confirmed transaction

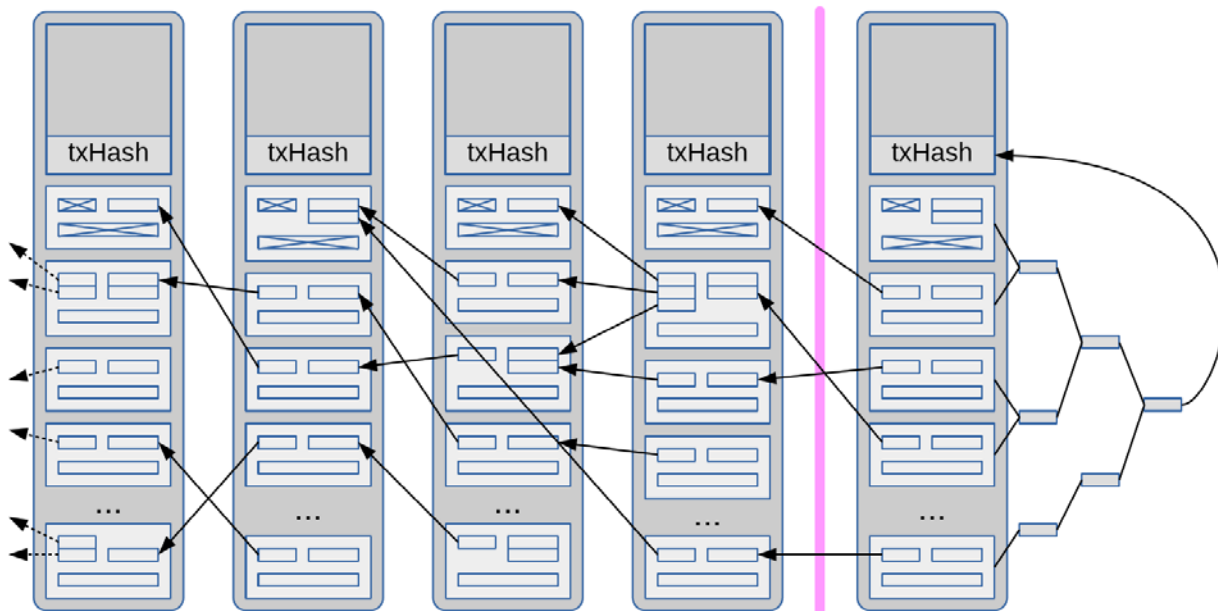


Block Header



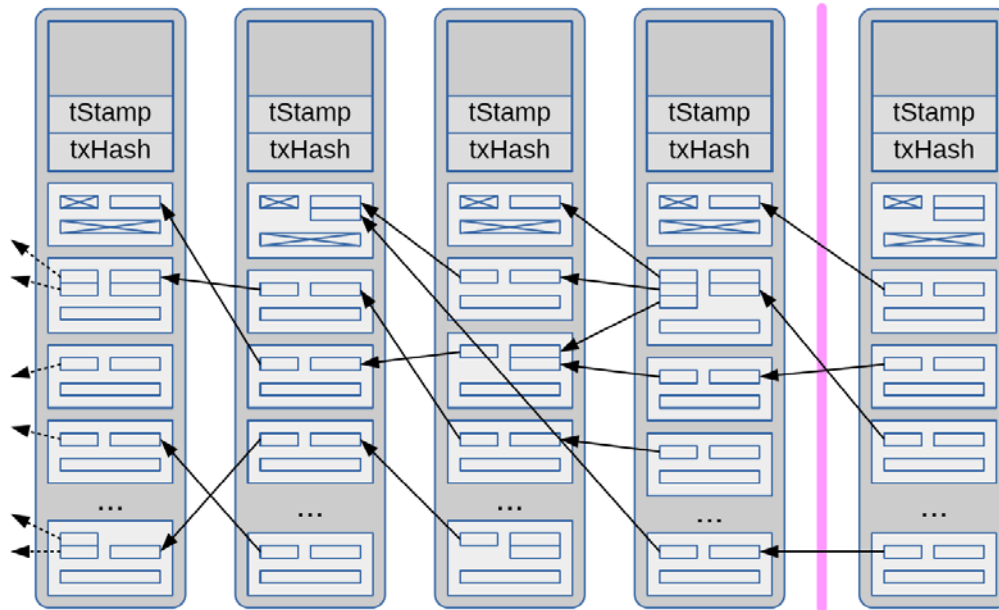
Block Header

- **Merkle Tree** root of transaction hashes
 - Uniquely identify transactions included in block



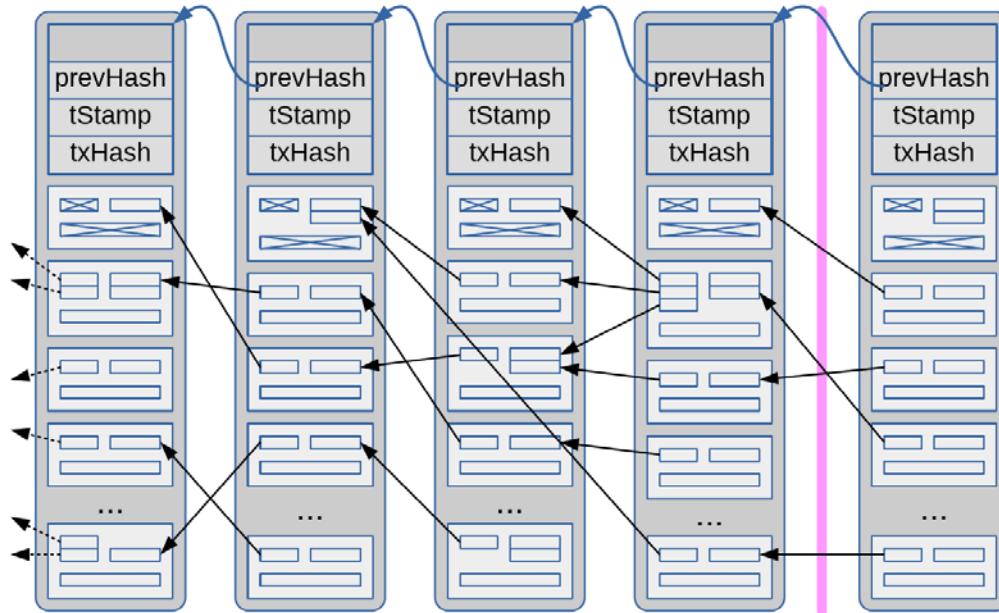
Block Header

- Timestamp of block creation
 - Monotonically increasing



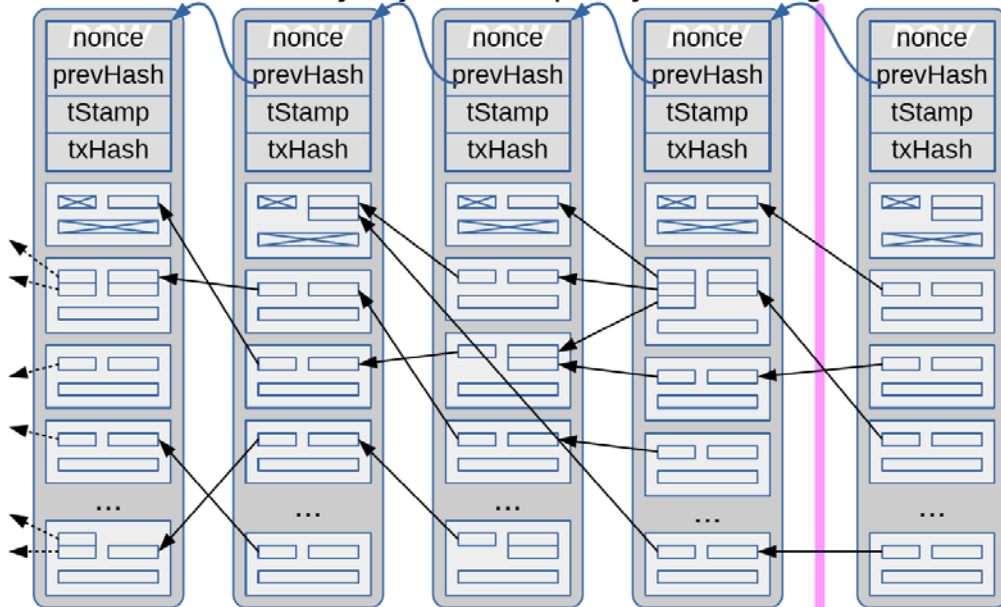
Block Header

- Hash of previous block header
 - Linked list → *block chain*



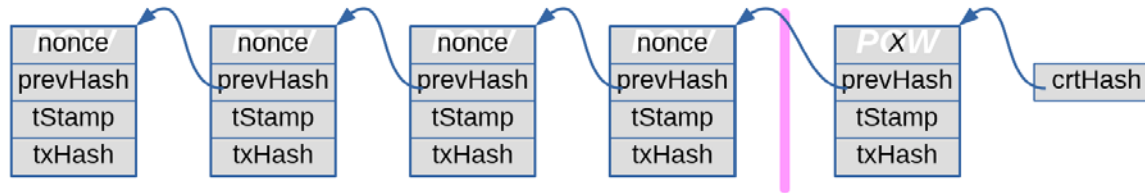
Block Header

- PoW nonce: limit block creation rate to 1 / 10min.
 - Give peers time to double-check block & transaction validity
 - Difficulty adjusted adaptively toward target block creation rate



PoW, a.k.a. “Difficult Math Puzzle”

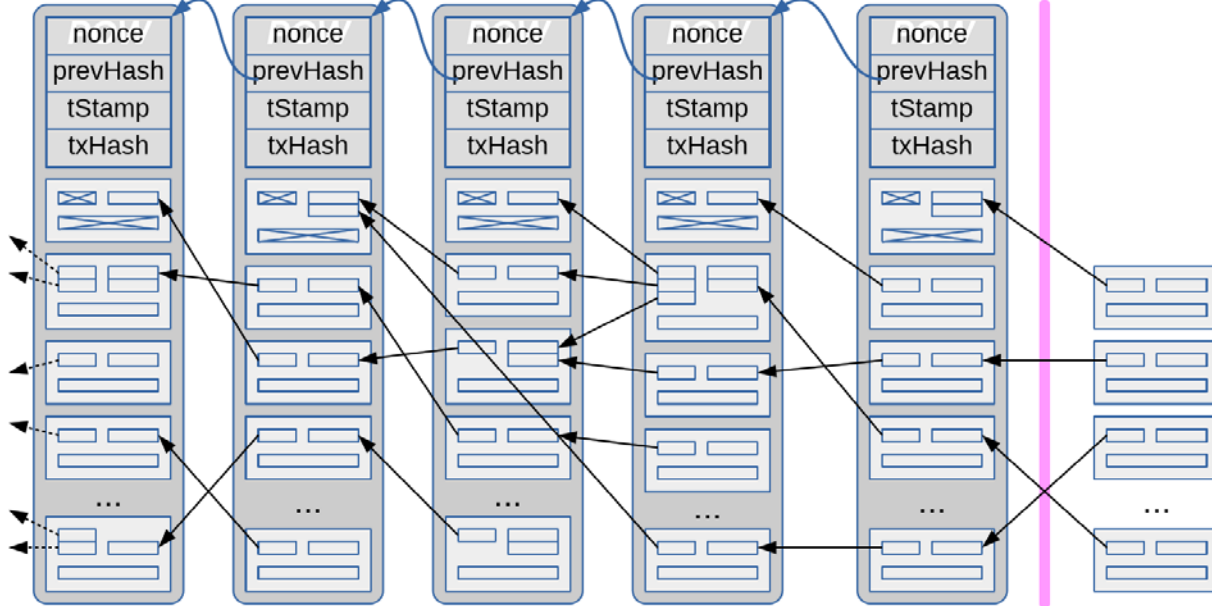
$$H(x, \text{prevHash}, \text{tStamp}, \text{txHash}) \leq 0x\underbrace{00\dots0}_{n \text{ 0-bits}}FF\dots F$$



- Hash function H output unpredictable (by design)
 - No formula to solve for x : Try all x until solution found!
 - Statistically, difficulty (expected # of attempts) is $2^{(n-1)}$, where n is the # of leading 0-bits at output of H func.
- Goals:
 - Control block creation rate (every 10 minutes for BTC)
 - Prohibit changes in previously settled (confirmed) blocks

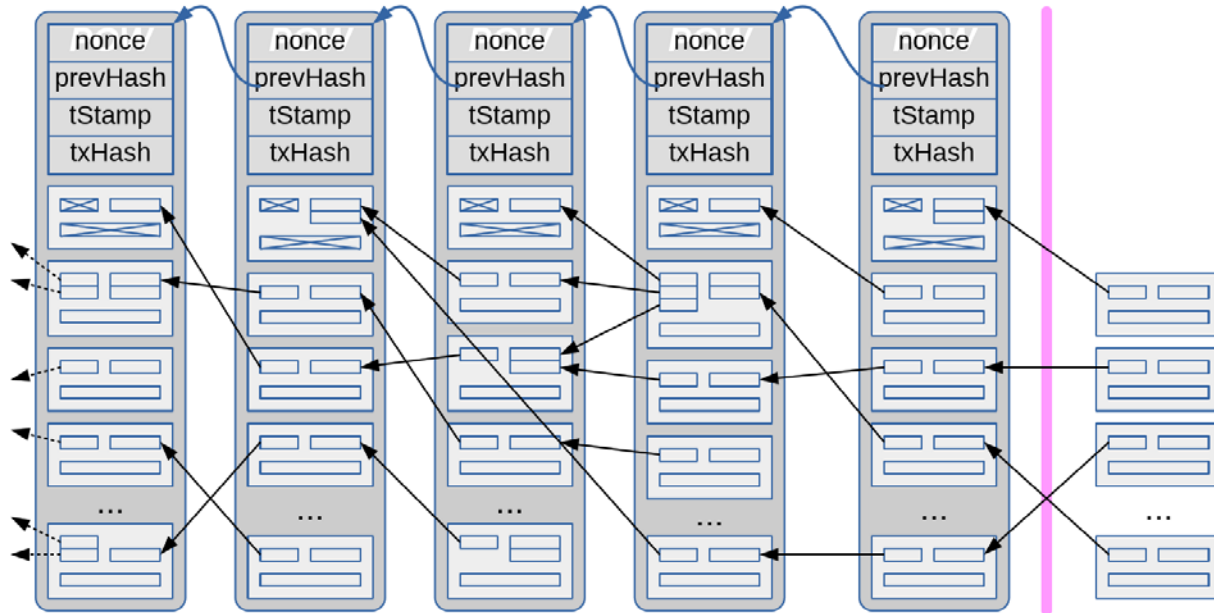
Blockchain

- Non-repudiable ledger of confirmed-transactions
 - Peers *always* prefer longest known blockchain (per protocol)
 - PoW makes it unfeasible to recompute, catch up to peers



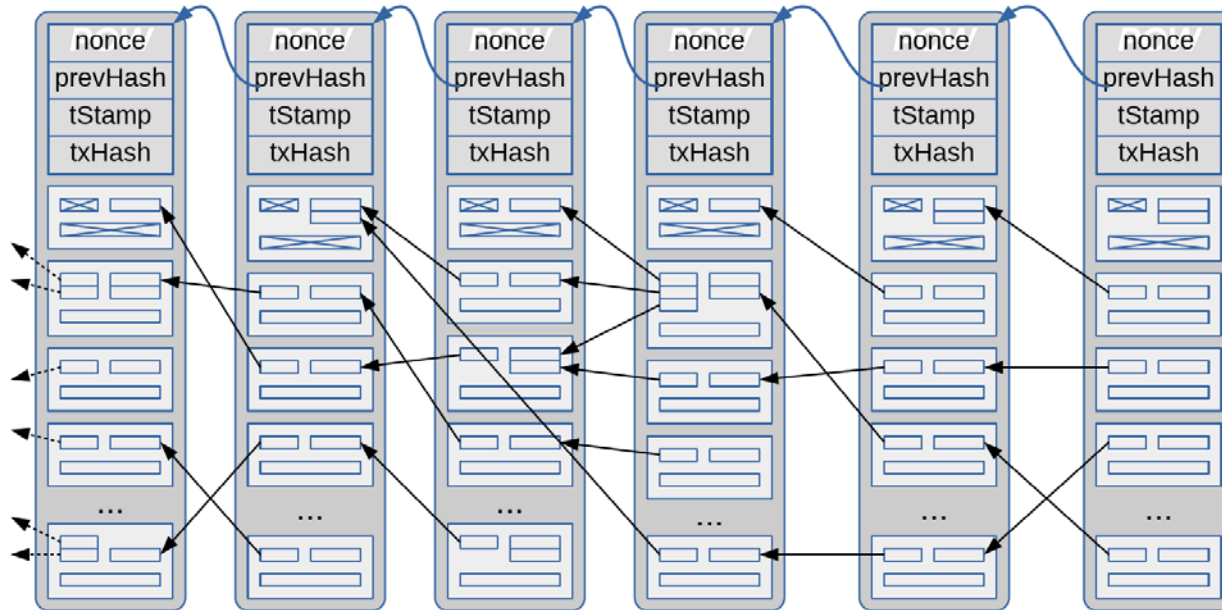
Depth, Height, Confirmations

Depth, #confirmations: ... 3 2 1 0
Height: ... N-2 N-1 N N+1



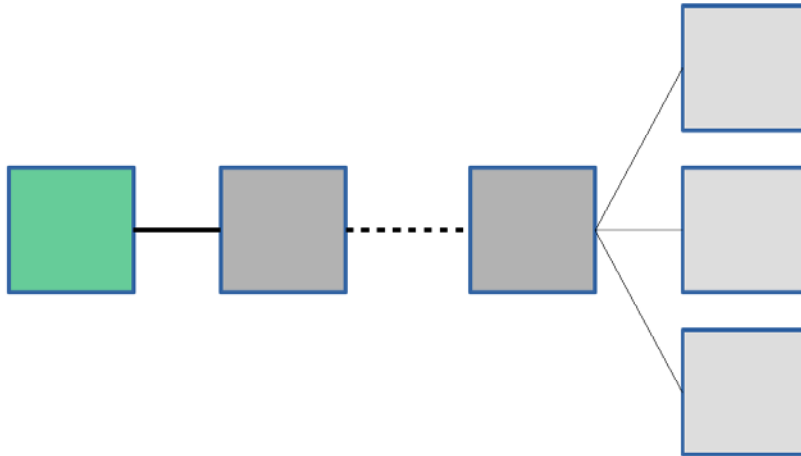
Depth, Height, Confirmations

Depth, #confirmations: ... 4 3 2 1
Height: ... N-2 N-1 N N+1



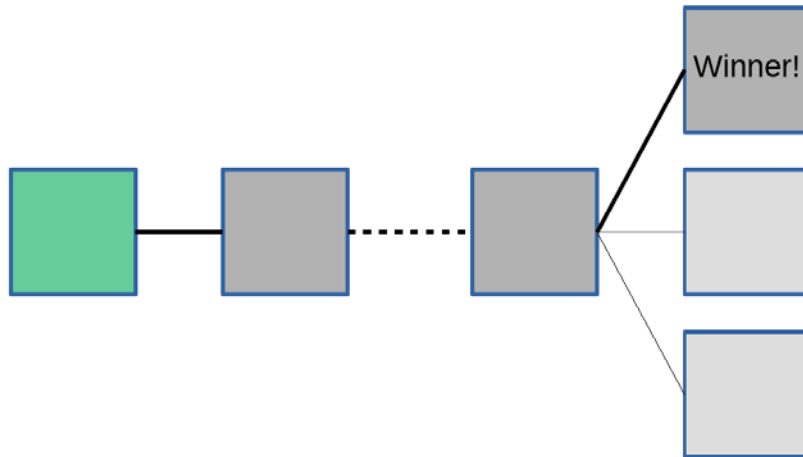
Stale Blocks

- Multiple miners race to create next block



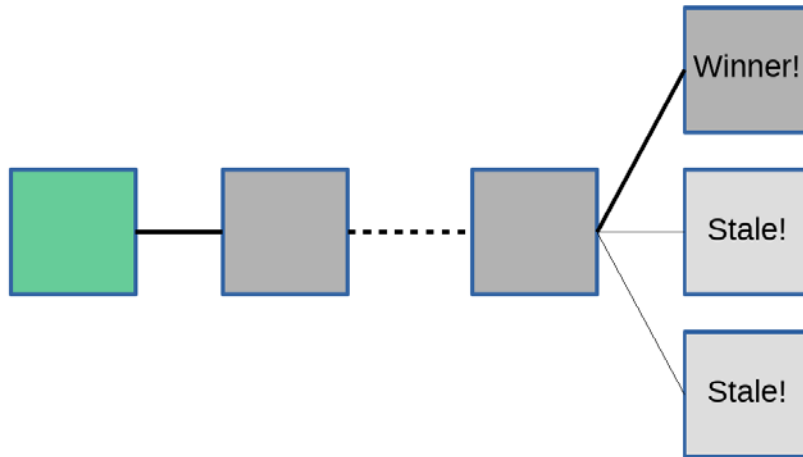
Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers



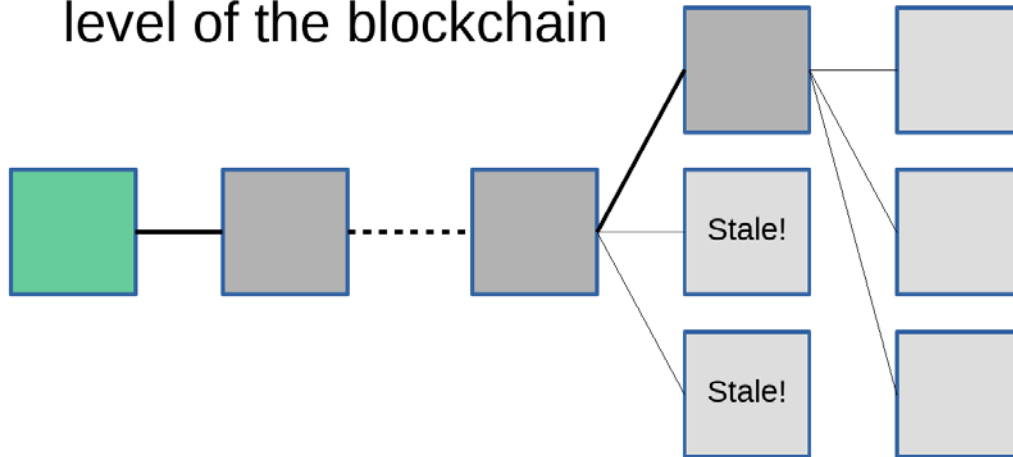
Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers
- Losers' work-in-progress becomes *stale*



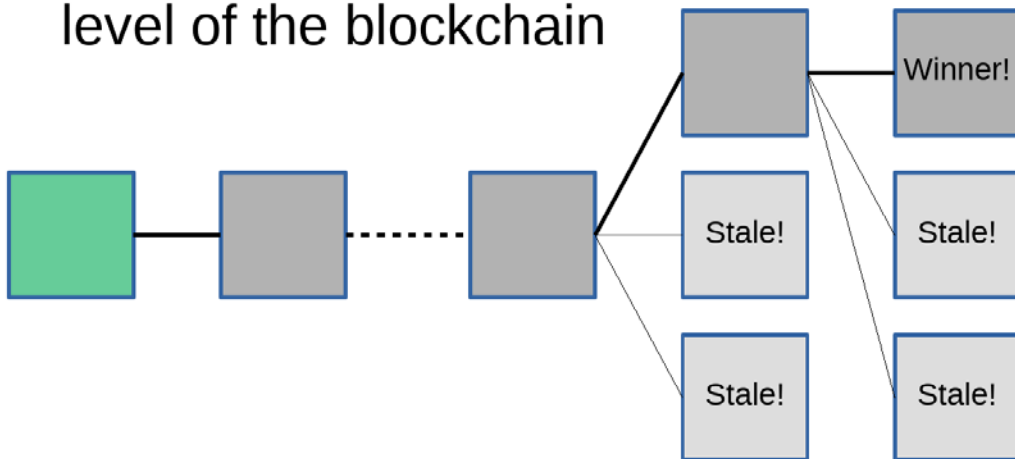
Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers
- Losers' work-in-progress becomes *stale*
- Race restarts at next level of the blockchain



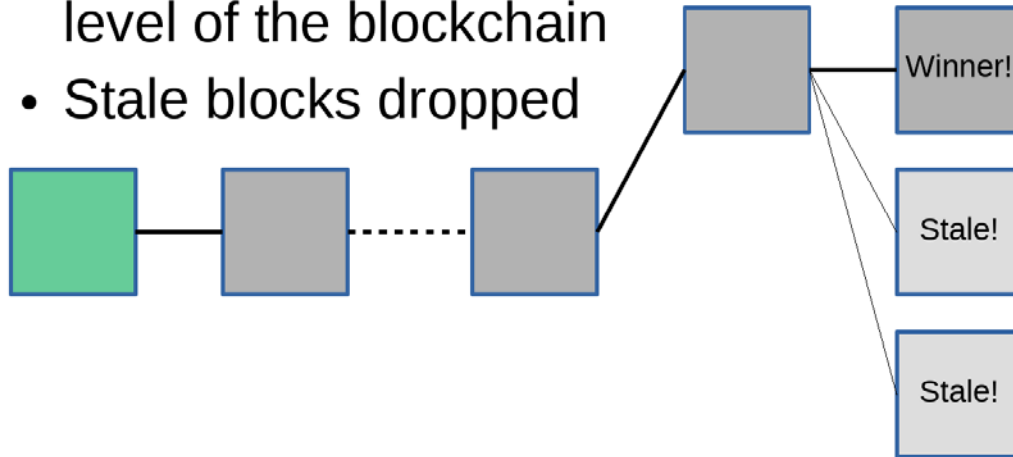
Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers
- Losers' work-in-progress becomes *stale*
- Race restarts at next level of the blockchain



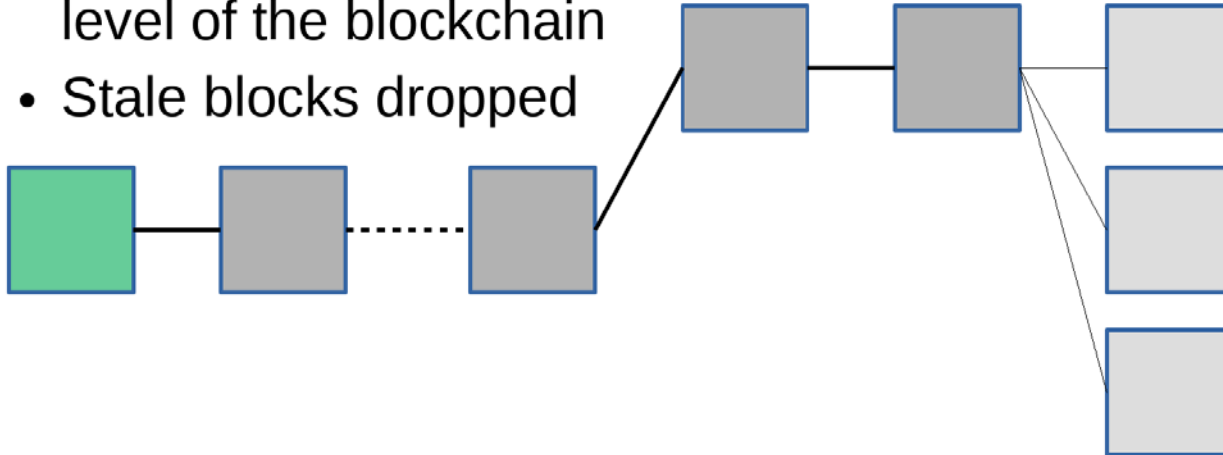
Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers
- Losers' work-in-progress becomes *stale*
- Race restarts at next level of the blockchain
- Stale blocks dropped

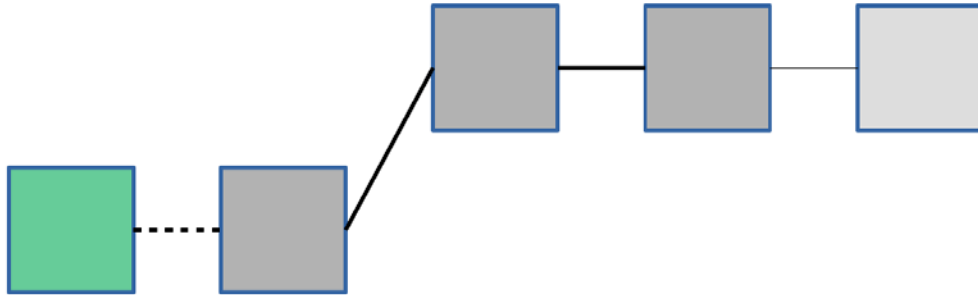


Stale Blocks

- Multiple miners race to create next block
- Winner broadcasts their block to all peers
- Losers' work-in-progress becomes *stale*
- Race restarts at next level of the blockchain
- Stale blocks dropped

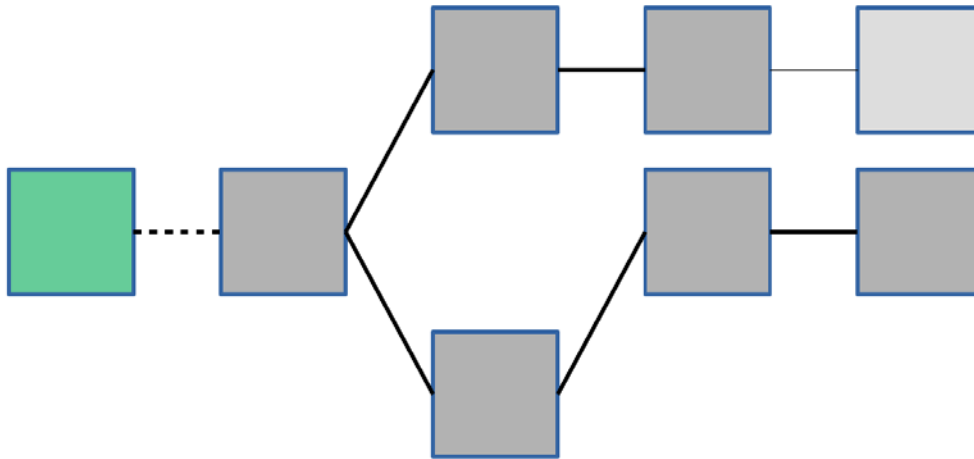


Orphan Blocks



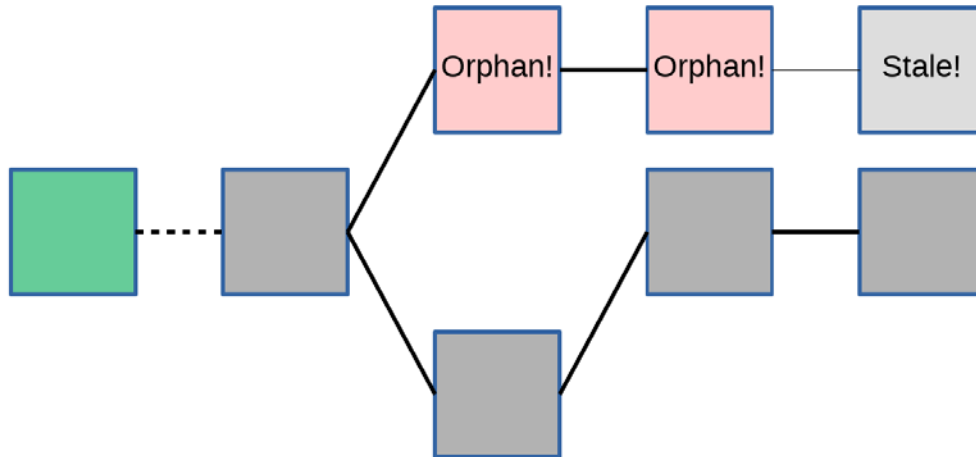
Orphan Blocks

- Suddenly a valid, *longer* chain is announced
 - Presumably, after network delay or temp. partition



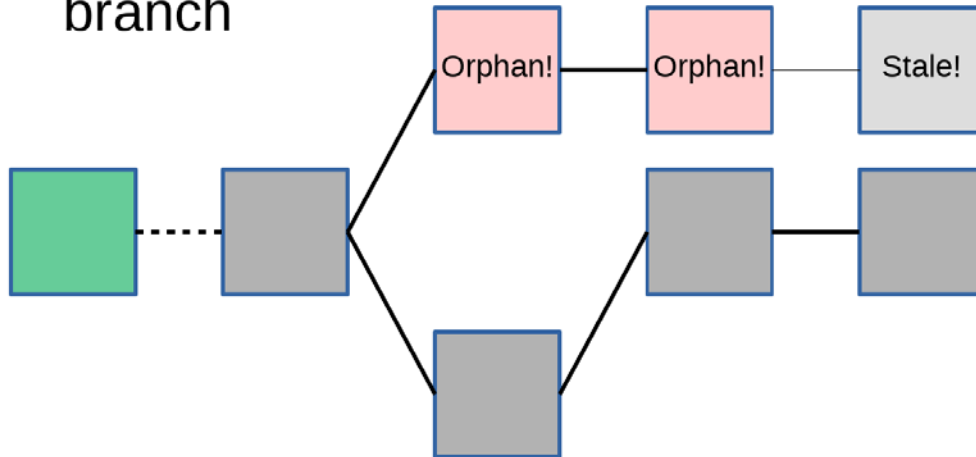
Orphan Blocks

- Suddenly, a valid longer chain is announced
 - Presumably, after network delay or temp. partition
- Previous branch becomes orphaned



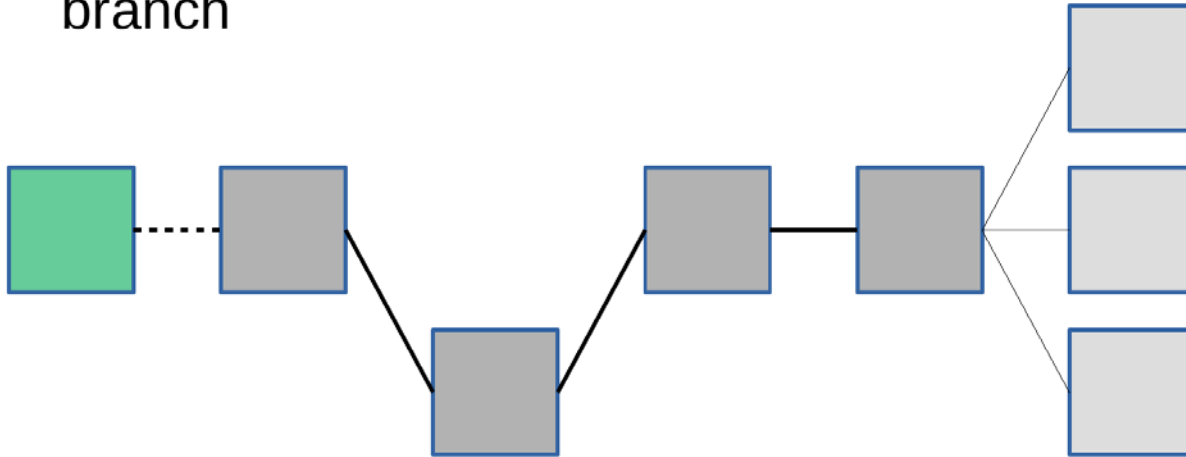
Orphan Blocks

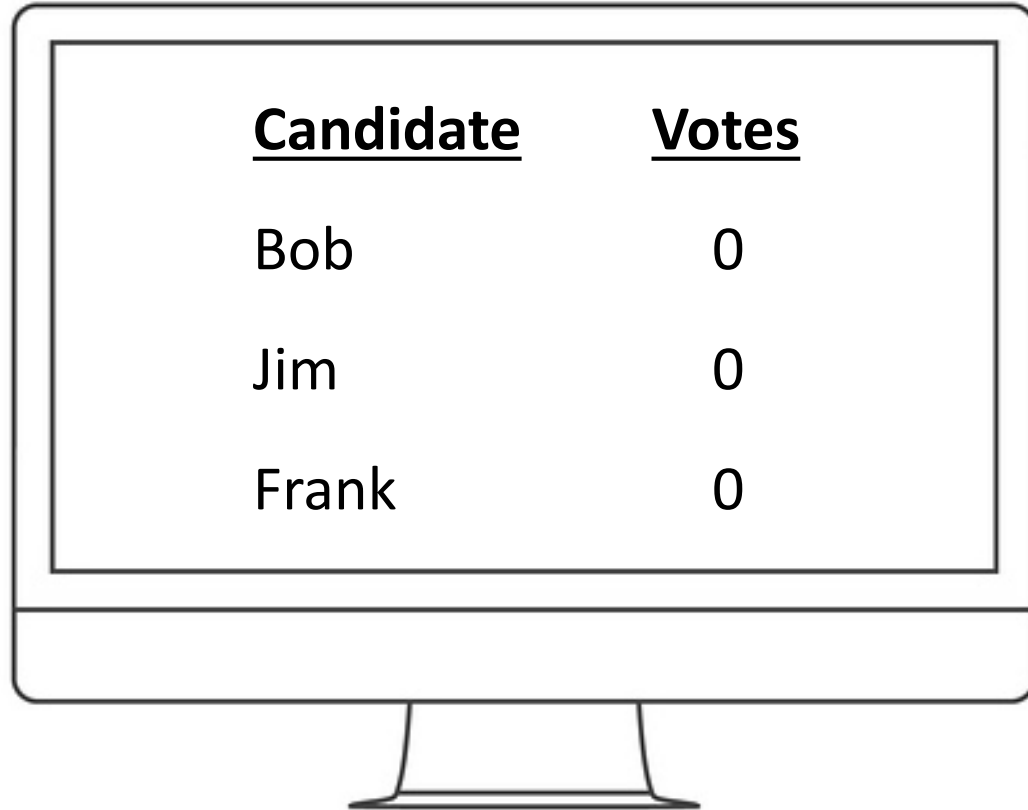
- Suddenly, a valid longer chain is announced
 - Presumably, after network delay or temp. partition
- Previous branch becomes orphaned
- Miners begin working on longer, preferred branch



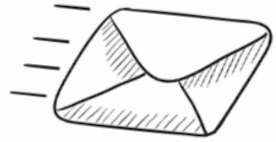
Orphan Blocks

- Suddenly, a valid longer chain is announced
 - Presumably, after network delay or temp. partition
- Previous branch becomes orphaned
- Miners begin working on longer, preferred branch

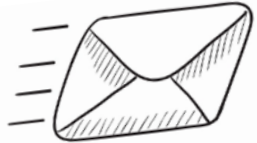




| <u>Candidate</u> | <u>Votes</u> |
|------------------|--------------|
| Bob | 0 |
| Jim | 0 |
| Frank | 0 |

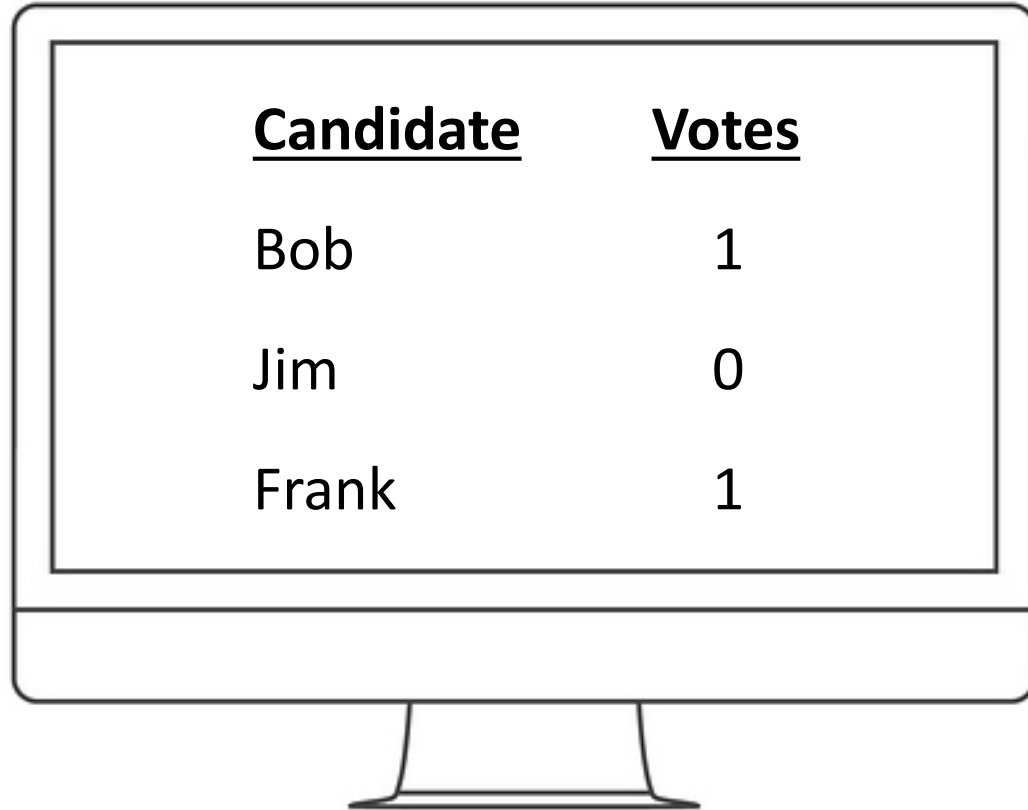


Bob: 1 vote



Frank: 1 vote

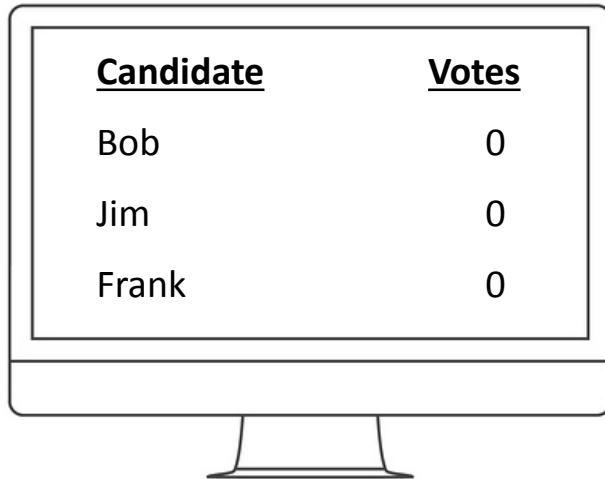
| <u>Candidate</u> | <u>Votes</u> |
|------------------|--------------|
| Bob | 0 |
| Jim | 0 |
| Frank | 0 |



A computer monitor is shown with a table of candidate votes on its screen. The table has two columns: 'Candidate' and 'Votes'. The data is as follows:

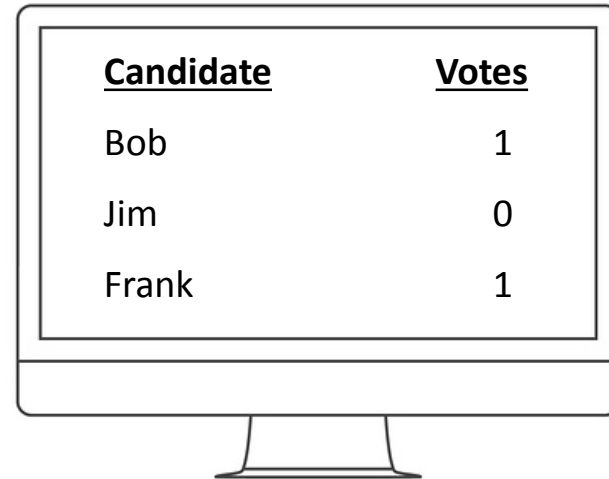
| <u>Candidate</u> | <u>Votes</u> |
|------------------|--------------|
| Bob | 1 |
| Jim | 0 |
| Frank | 1 |

State: 1



| <u>Candidate</u> | <u>Votes</u> |
|------------------|--------------|
| Bob | 0 |
| Jim | 0 |
| Frank | 0 |

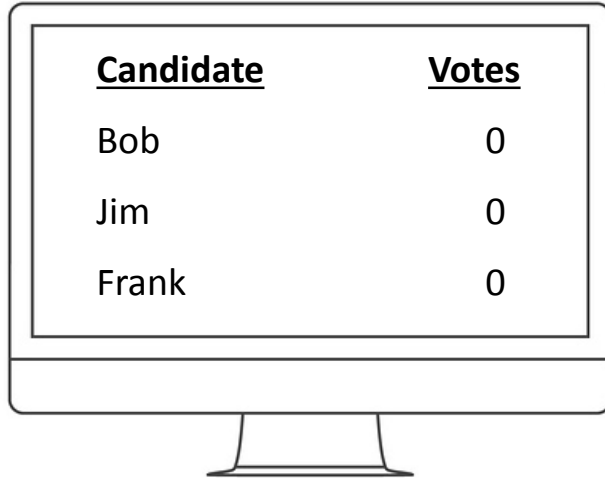
State: 2



| <u>Candidate</u> | <u>Votes</u> |
|------------------|--------------|
| Bob | 1 |
| Jim | 0 |
| Frank | 1 |

Equivalent to:

State: 1



| <u>Candidate</u> | <u>Votes</u> |
|------------------|--------------|
| Bob | 0 |
| Jim | 0 |
| Frank | 0 |

State: 2

State 1 plus...



Bob: 1 vote



Frank: 1 vote

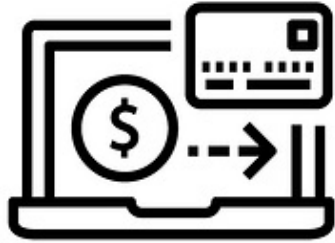
General purpose blockchains

Messages are... anything!

Each block is the system state at that time

$$\textit{Current State} = \textit{Original state} + \textit{All Changes}$$

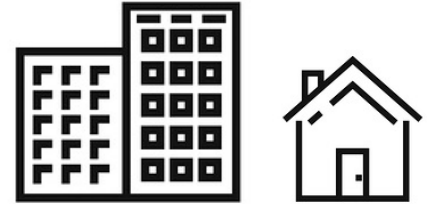
Use cases around



Payment System



**Health Care
Records**



**Real Estate
Records**

Addendum

High Energy Use

- As of Apr. 2018, the overall Bitcoin P2P network used cca. 930 kWh per *transaction* (not block)!
 - <https://digiconomist.net/bitcoin-energy-consumption>
 - Only slightly more than the *monthly* use of the [average US home](#) (900 kWh as of 2016)
- Increase in perceived BTC value → more competing miners → harder PoW difficulty (to maintain 10-minute block creation interval)
 - Non-linear increase in per-transaction electricity use
- Turns out, decentralization is highly expensive!

Blockchain: Executive Summary

Pros:

Authentication built-in

Easy to audit history

Easy to detect data manipulation

Very difficult to disrupt

Cons:

Proof-of-work very inefficient

- Alternatives exist!

State updates are slow

Best for simple computations

Bitcoin

Block #509169

| Summary | |
|------------------------------|----------------------|
| Number Of Transactions | 1915 |
| Output Total | 10,289.28130284 BTC |
| Estimated Transaction Volume | 1,818.68925455 BTC |
| Transaction Fees | 0.4893378 BTC |
| Height | 509169 (Main Chain) |
| Timestamp | 2018-02-14 15:16:59 |
| Received Time | 2018-02-14 15:16:59 |
| Relayed By | 58COIN |
| Difficulty | 2,874,674,234,415.94 |
| Bits | 392292856 |
| Size | 1132.416 kB |
| Weight | 3992.574 kWU |
| Version | 0x20000000 |
| Nonce | 1858980081 |
| Block Reward | 12.5 BTC |

| Hashes | |
|----------------|--|
| Hash | 000000000000000002c4b94355945eea353bc720c58a73c2b8593f489550cb3 |
| Previous Block | 00000000000000001d620a2e3ad126ec5038bf42343c419eb6fcdf7240a471 |
| Next Block(s) | |
| Merkle Root | 3ad680735c45cc62b1ea6b7efeb34f82a2660c5e8280354c45f7ffa03c9137e2 |

Transactions

| | | |
|--|---|--|
| ab0da64ea834fd2acb81eb081d8103c9e31fd14a7d055f2ce2718c59dd4fa5df | | 2018-02-14 15:16:59 |
| No Inputs (Newly Generated Coins) | → 14DJTuAUhB7cwRsbU1z6W8hZY6FnExpfLS Unable to decode output address | 12.9893378 BTC 0 BTC |
| | | 12.9893378 BTC |
| 4feb8981da942b10a2a384003fba1c1d78c8f192cd2747e43ae552ed237f267d | | 2018-02-14 15:16:59 |
| 1H6ZZpRmMnrw8ytepV3BYwMjYnEKWDqVP | → 12PaHfRJBrmvJYmTpZ32Pswf8eYbKcAE131 1GpqR4vsdvEfgtNyiUrDrDLTBjvnsentX 1H6ZZpRmMnrw8ytepV3BYwMjYnEKWDqVP | 0.4983 BTC 0.1495 BTC 5.01651602 BTC |
| | | 5.66431602 BTC |

Ethereum

Block 5089469

[Previous](#)[Next](#)

| | |
|-------------------|---|
| Hash: | 0x4b7ced1ac95fa07a06fbb0352468797bd038e8c1fb0f6d4de2838f5712469c27 |
| Difficulty: | 2,863,007,803,096,150 |
| Miner: | ✓ miningpoolhub1 (0xb293...) (Mined in 19s) |
| Reward: | 3.13573 ETH <u>\$2,777.29</u> (Block Reward: 3 ETH + Fee Reward: 0.13573 ETH + Uncle Inclusion Reward: 0 ETH) |
| Tx Fees: | 0.13573 ETH <u>\$120.22</u> (4.33% of the total block reward) |
| Tx / Uncles: | 202 Transactions and 0 Uncles |
| Gas Limit: | 8,000,029 |
| Gas Usage: | 83.8 % (6,701,815 of 8,000,029) |
| Lowest Gas Price: | 1 GWei |
| Time: | 02/14/2018 10:31:12 AM (a minute ago) |
| Size: | 28,742 bytes |
| Extra | t3 (Raw: 0x7433) |

[202 Transactions](#)[0 Uncles](#)[Details](#)

| Hash | Type | From | To | Value | Fee | Gas Price |
|---------------------------------|--------|------------------------------------|------------------------------------|-------------|-------------|-----------|
| 0x00622dc883... | Tx | 0x5BaEac0a0417a... | 0x342DB8C17dF30... | 0.03175 ETH | 0.0021 ETH | 100 GWei |
| 0x2f21a28b88... | Tx | 0x96b7DA642FAA7... | 0xee4d84B1E8C78... | 0.01 ETH | 0.00208 ETH | 99 GWei |
| 0x4a6a150361... | Tx | ✓ Bittrex (0xfbb1...) | 0x419D0d8BdD9aF... | 0 ETH | 0.00531 ETH | 90 GWei |
| | ↪ Call | 0x419D0d8BdD9aF... | 0x267808e5246D1... | 0 ETH | 0.0028 ETH | 90 GWei |
| | ↪ Call | 0x267808e5246D1... | 0xe6a51Bd48f93A... | 0 ETH | 0.00247 ETH | 90 GWei |
| 0x7359bb70de... | Tx | 0x45a0ba49c5244... | 0xAA1A6e3e6EF20... | 4.97698 ETH | 0.00233 ETH | 70 GWei |

Existing blockchain programs are vulnerable

- Over **\$40M** were stolen from TheDAO due to a bug in the implementation (June 2016)
- **\$32M** were stolen due to a bug in a commonly used contract (June 2017)
- Bugs in smart contracts cannot be fixed after deployment

We want to build correct software, but current approaches have been shown to have security vulnerabilities

Obsidian: a new programming language

Goals

- Make certain vulnerabilities impossible
- Make it easier to write correct programs
- Show effectiveness and correctness

Components

1. Typestate-oriented programming
 - Shown to be helpful in documentation, but no studies of writing code
2. Resource types
 - Integration into an OO-style language is novel

Contact Information

Presenters

Elli Kanal

Technical Manager

Email: ekanal@cert.org

Gabe Somlo

Cybersecurity Researcher

Email: glsomlo@cert.org

