BLOCKCHAIN UNDER THE HOOD

TECHNICAL MASTERCLASS

PRESENTED BY

neuroware

NEUROWARE - MEET THE FOUNDERS

MSC MALAYSIA Status Company

Mark Smalley - CEO

Living in Malaysia for the past 20 years

Building Fintech Solutions for 15+ years

Spent 10 years building tech communities

Building blockchain apps for 5+ years

Ruben Tan - CTO

More than 10+ years of software engineering exp

Active community evangelist & technology speaker

Early developer in MyTeksi, OnApp, Bookya, etc

Studied distributed consensus as a hobby

NEUROWARE - MEET THE FOUNDERS

MSC MALAYSIA Status Company

THIS IS ME!

Mark Smalley - CEO

Living in Malaysia for the past 20 years

Building Fintech Solutions for 15+ years

Spent 10 years building tech communities

Building blockchain apps for 5+ years

Ruben Tan - CTO

More than 10+ years of software engineering exp

Active community evangelist & technology speaker

Early developer in MyTeksi, OnApp, Bookya, etc

Studied distributed consensus as a hobby

OVERVIEW

- **BLOCKCHAIN REVIEW**
  - General traits of blockchains
  - Consensus algorithm overview
  - Q&A + Break
- **BITCOIN REVIEW**
  - Bitcoin deep dive
  - Satoshi consensus
  - Q&A + Break
- **SMART CONTRACTS**
  - Payment contracts overview
  - General contracts overview
  - Q&A + Break
- **STORING & SECURITY CRYPTOCURRENCIES**
  - End-user/Exchange security
  - Methods of storing wallets
  - Q&A + Break

1 hr

per segment
(hopefully)

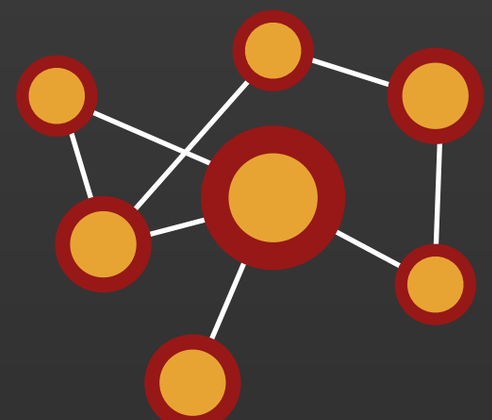
BLOCKCHAIN REVIEW

Definition and general traits

GENERAL TRAITS OF BLOCKCHAINS

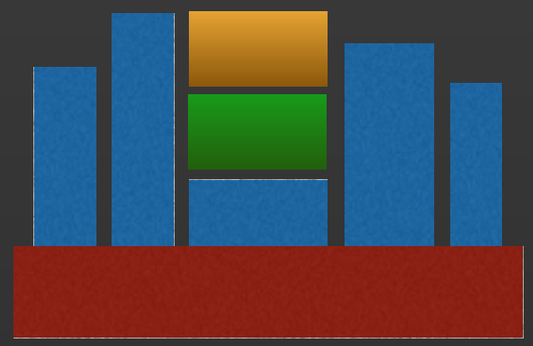
Blockchain stores data

- Ledgers, DNS records, etc
- Immutable once recorded
- Everybody has a copy



Blockchain is a network

- Fully distributed
- Peer to peer connection
- Has a consensus algorithm



Blockchain is infrastructure

- Enables trust-less interaction
- Enables high automation
- Creates new business models

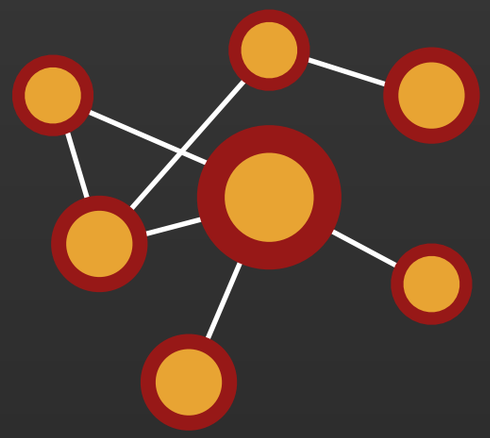
BLOCKCHAIN AND STORAGE

- **Ledgers, DNS records, etc**
  - Most blockchains are highly specialised
  - Not as generalised as normal databases
  - Algorithms and consensus algorithm depends on the kind of data stored and how it is accessed
  - Usually have a very simple query mechanism
- **Immutable once recorded**
  - Immutability is built-in, not opt-in
  - Uses deterministic algorithms to ensure consistency of data
  - Not suitable for transient/noise/trivial data
- **Everybody has a copy**
  - Distributed nature means it is meant to be propagated
  - Each person has a full copy of the entire blockchain*
  - Consensus algorithm ensures all copies stay constant

BLOCKCHAIN AND STORAGE

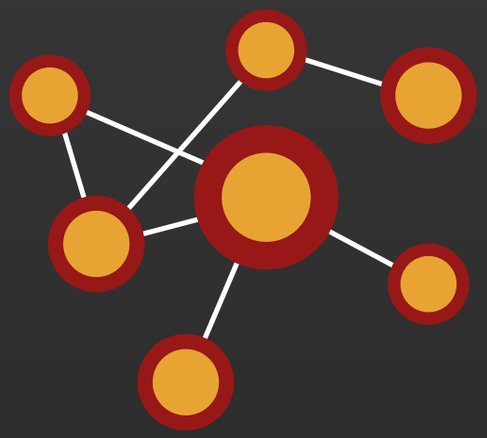
- **Real-world analogy: the phone book**
  - **Immutability** - Contains public data that is mostly immutable
  - **Distributed** - Distributed to every person, no central control
  - **Consensus** - Updated every now and then

BLOCKCHAIN AND NETWORK



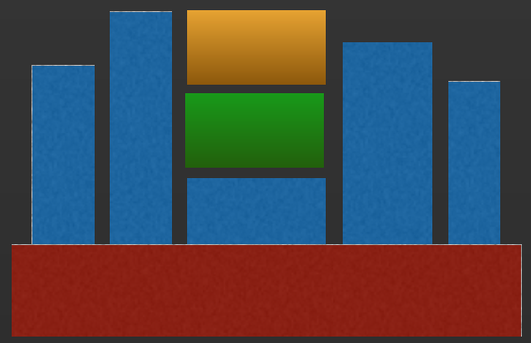
- **Fully distributed**
  - Everybody has a copy, every copy is *consistent*
  - No central authority to control all nodes
  - Changes/updates are made through a quorum or consensus
  - Highly resilient but not real-time (tradeoff)
- **Peer to peer connection**
  - Each node connects to peers, not a central authority
  - Updates/changes is propagated through a gossip protocol
  - Peers have a discovery mechanism to find each other
- **Has a consensus algorithm**
  - Consensus algorithm is used to ensure consistency
  - Nodes that are not consistent are rejected from the network

BLOCKCHAIN AND NETWORK



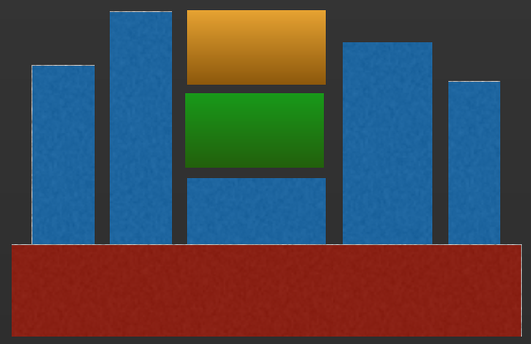
- **Real-world analogy: firechat**
  - **Distributed** - nodes are not connected to a central server
  - **P2P Connection** - nodes can talk directly to another node
  - **Consensus** - depends on the need of the mesh networks

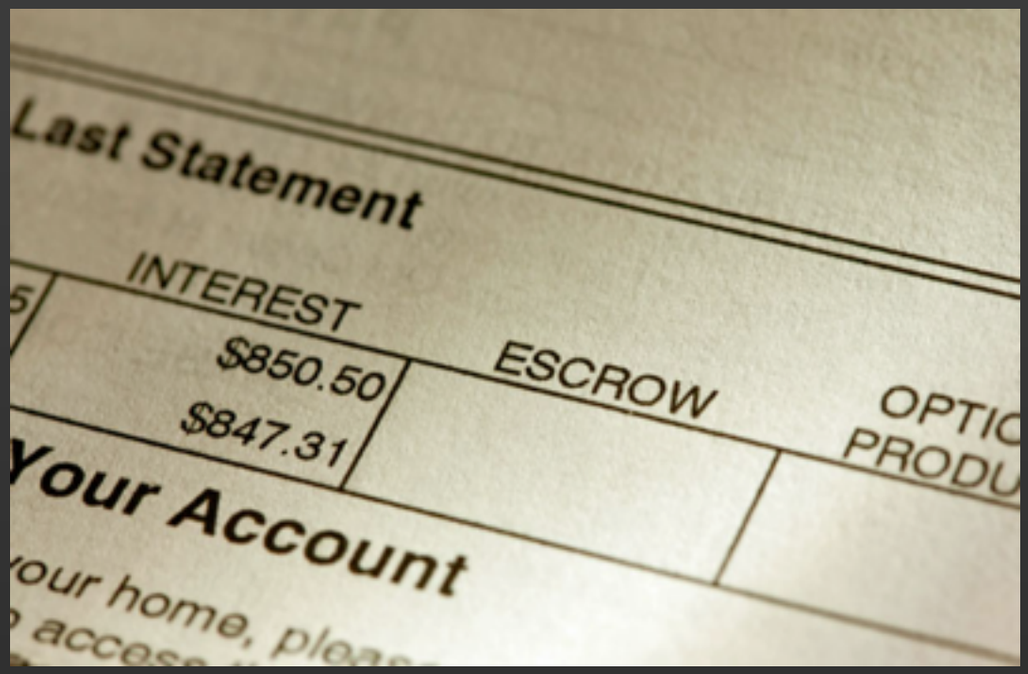
BLOCKCHAIN AND INFRASTRUCTURE



- **Enable trust-less interaction**
  - Removes humans from the equation
  - Trust the algorithms and network, not people and institutions
  - Corporates have a neutral ground to conduct business on
  - Infrastructure is not owned by anybody
- **Enables high automation**
  - Data is always “clean” because everything is automated
  - Data is specialised, allowing easier automation
- **Creates new business models**
  - Removes the bar of entry for traditional API models
  - Allows third-party to easily build solutions on top of it
  - Allows owner of the infrastructure to monetise and control it

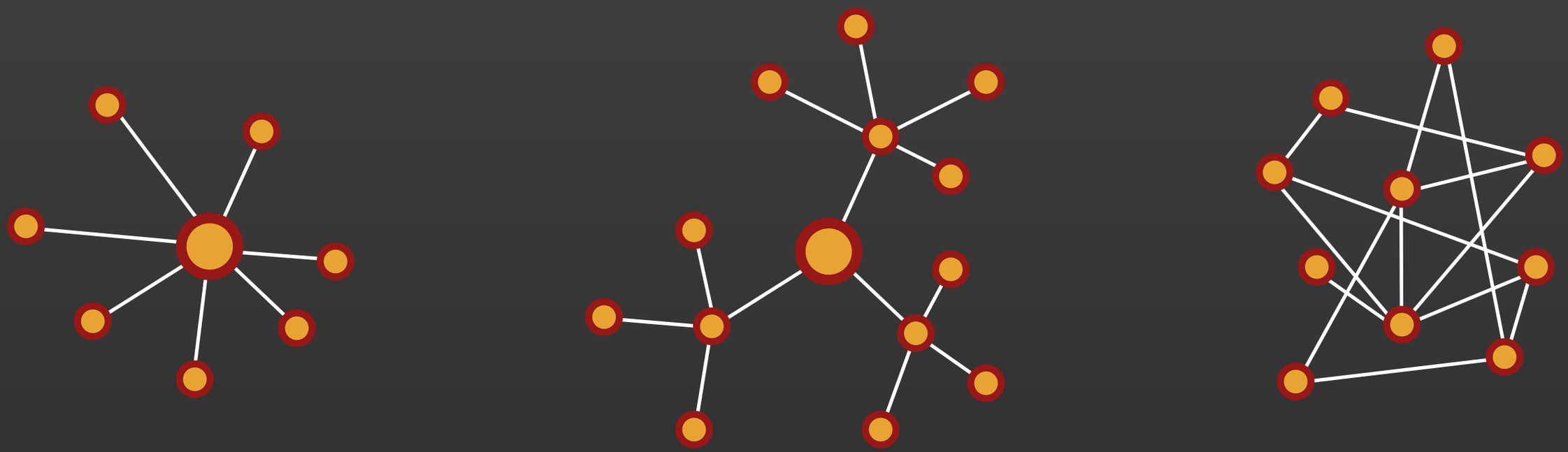
BLOCKCHAIN AND INFRASTRUCTURE





- **Real-world analogy: distributed escrow system**
  - **Trustless** - escrow company a human-less, neutral entity
  - **Automation** - notifications, settlements, audit, etc
  - **Business** - allows research companies to create reports

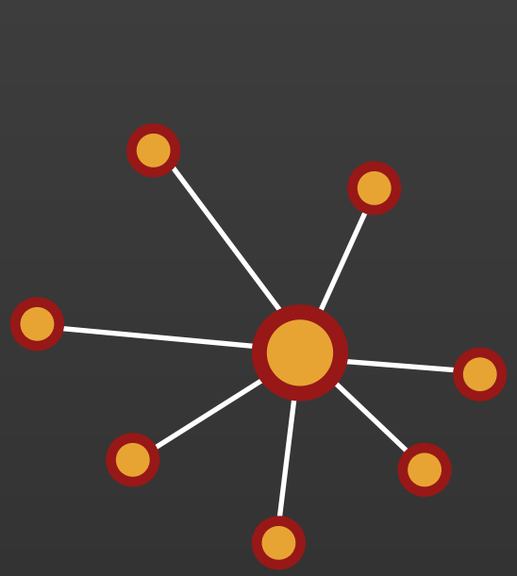
EVOLUTION OF TOPOLOGIES/BUSINESS MODELS

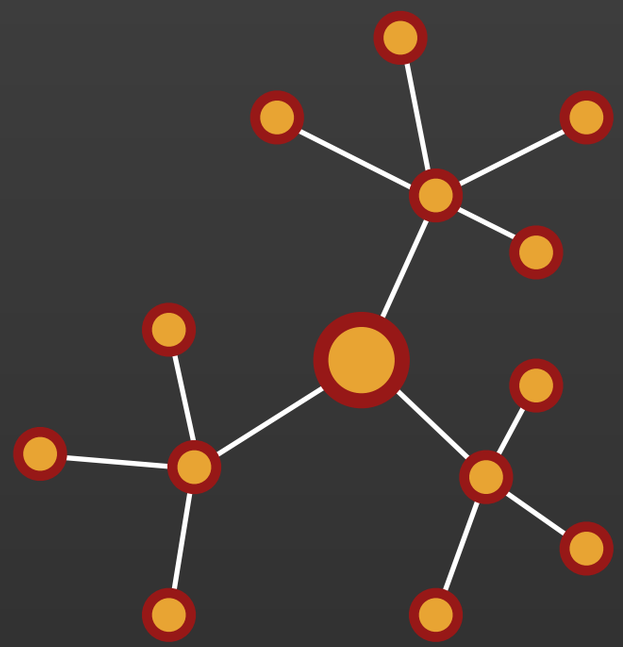


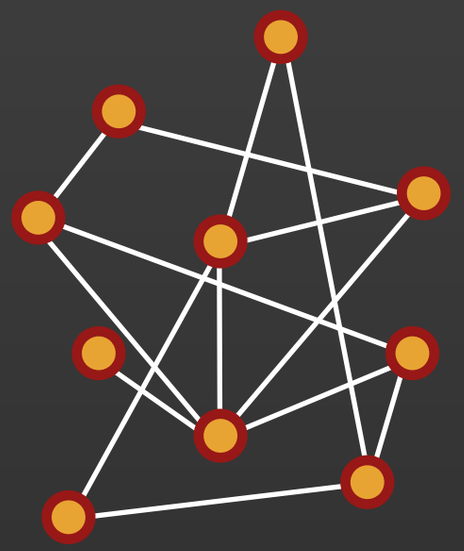
Centralised	Cloud	Distributed
• Single point of failure • Expensive to scale	• Basically outsourced infrastructure • Mitigates failure • Requires trust	• More secure as more nodes exist/join • Highly resilient to data loss and attacks

Evolution →

EVOLUTION OF BUSINESS MODELS



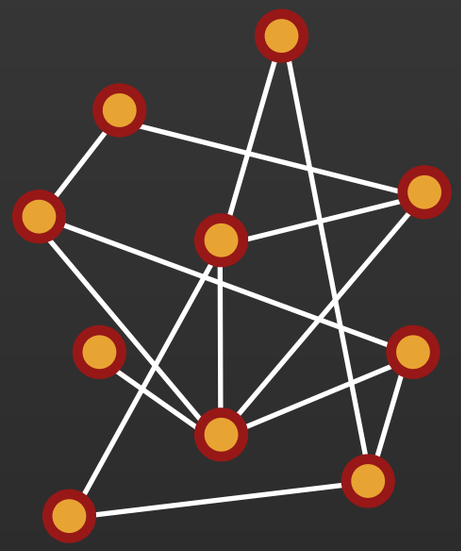




Centralised	Cloud	Distributed
Telephone Service	Whatsapp	Firechat
Walmart	Amazon	Open Bazaar
Taxi Services	Uber	Lazooz

Evolution →

DISTRIBUTED TECHNOLOGIES



- **Pros**
  - The more participating nodes, the more durable the network is
  - Harder to achieve 100% data loss due to massive redundancy
  - Can still function under partial availability or split network
  - Can distribute computing power to save costs
  - Usually less infrastructure to manage and control
- **Cons**
  - Lack of central authority means harder to make big changes
  - Requires a consensus of some kind to maintain order
  - Does not have a good reputation (bittorrent, bitcoin, etc)

THE PHONE BOOK EXERCISE

Understanding the difficulties of a distributed system

CONSENSUS IS EVERYTHING

- Any distributed system needs a way to synchronise each node

- In computer science, this can be illustrated by the **CAP theorem**
  - **Consistency** - each read returns the most-recent data
  - **Availability** - each read returns a non-error response
  - **Partition Tolerance** - system continues to operate despite network failure

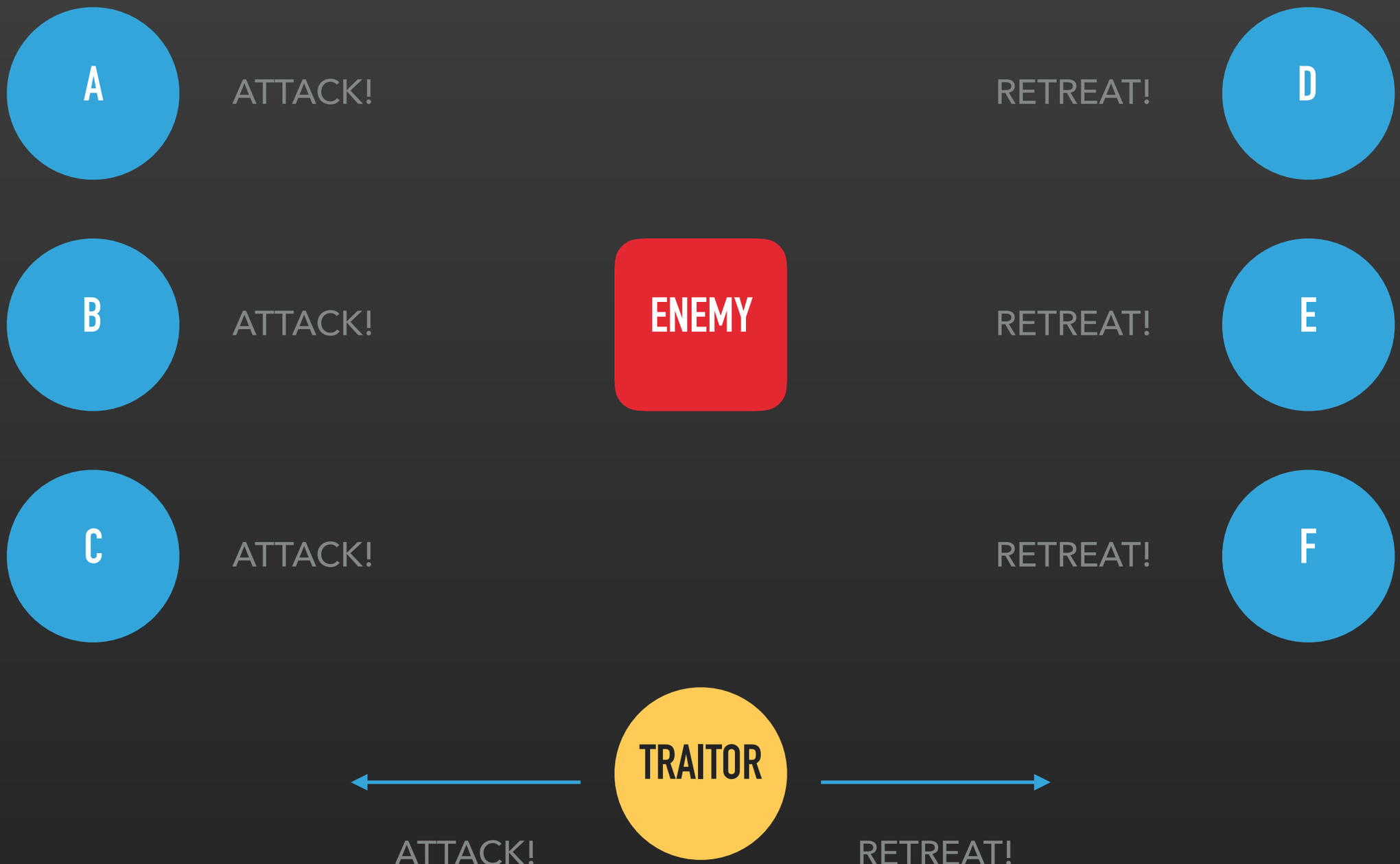
- Choose C+P or A+P
  - **Consistency + Partition tolerance** - mission critical database systems
  - **Availability + Partition tolerance** - high volume concurrent systems

- Blockchains are strictly C+P
  - Transactions are always most-recent
  - An outdated node is unable to propagate changes to the network

BYZANTINE GENERAL'S PROBLEM

- Consensus algorithms ensure the **consistency** of a distributed system
- Failure modes
  - **Fail-stop** - a node dies and does not recover
  - **Fail-recover** - a node dies and recovers some time later
  - **Byzantine** - a node misbehaves or displays unexpected behavior
- Byzantine general's problem
  - One of the first impossibility proof in computer communications
  - Impossibility proof = impossible to solve in a perfect manner
  - Originated from the **Two General's Problem** (1975)
  - Explored in detail in the **Byzantine General Problem** (1982)
  - Remains as the benchmark in evaluating distributed consensus algorithms today
- Common terms
  - **Byzantine fault** - any fault that presents different symptoms to different observers
  - **Byzantine failure** - loss of system service reliant on consensus due to byzantine fault
  - **Byzantine fault tolerance** - a system that is resilient against byzantine faults

BYZANTINE GENERAL'S PROBLEM

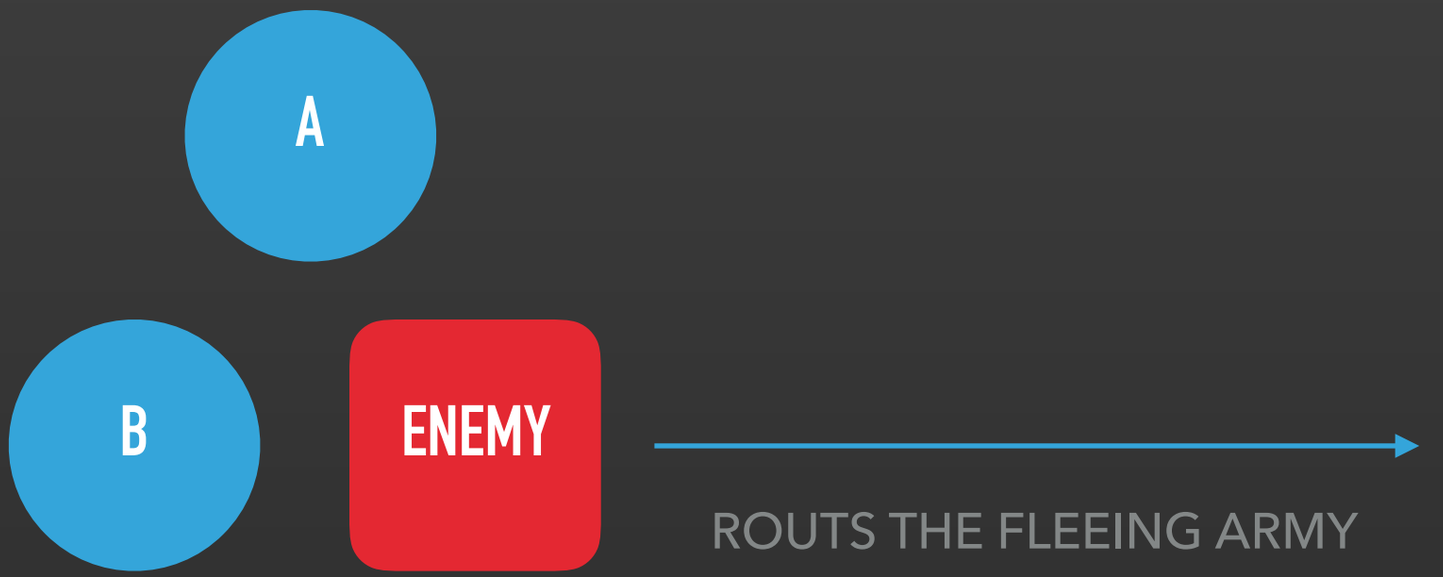


BYZANTINE GENERAL'S PROBLEM

ATTACKERS HAVE

INSUFFICIENT FORCE

AND ARE DESTROYED



MUAHAHA, NO CONSENSUS!

TWO GENERAL PROBLEM

2 Volunteers Needed!

DISTRIBUTED SYSTEMS ARE HARD

- Distributed computing is inherently **unreliable**
  - Peter Deutsch, Bill Joy, Tom Lyon, and James Gosling - **The Eight Fallacies of Distributed Computing** (1994-1997)

- Today many software engineers still believe in some or all of the fallacies:
  - The network is reliable
  - Latency is zero
  - Bandwidth is infinite
  - The network is secure
  - Topology doesn't change
  - There is only one administrator
  - Transport cost is zero
  - The network is homogeneous

DISTRIBUTED SYSTEMS ARE HARD

When you believe in any of the eight fallacies...

IMPOSSIBILITY PROOF

- Distributed systems achieve **consensus** when all nodes are acting as **one entity**

- Michael J. Fisher, Nancy A. Lynch, and Michael S. Patterson - **Impossibility of Distributed Consensus with One Faulty Process** (1985, Dijkstra Award winner, 2001)
  - Also known as the **FLP Impossibility Proof**
  - Consensus is **mathematically impossible** - tradeoffs instead of solutions

IMPOSSIBILITY PROOF

- Distributed systems achieve **consensus** when all nodes are acting as **one entity**

- Michael J. Fisher, Nancy A. Lynch, and Michael S. Patterson - **Impossibility of Distributed Consensus with One Faulty Process** (1985, Dijkstra Award winner, 2001)
  - Also known as the **FLP Impossibility Proof**
  - Consensus is **mathematically impossible** - tradeoffs instead of solutions

CONSENSUS ALGORITHMS

- Algorithm is a “formula” to allow a system to operate without human interference to solve problems
- **Consensus algorithm** allows a distributed system to agree on a value as the truth
- A good algorithm must have:
  - **Termination** - all nodes eventually decide on a value
  - **Agreement** - all nodes must agree on the same value
  - **Integrity** - values must be proposed by a node and not a default value
  - **Validity** - if a value is proposed and accepted, all nodes must accept the same value
- Common consensus algorithms:
  - **Two phase commit** - we will do a demo in a moment
  - **Three phase commit**
  - **Paxos**
  - **Proof-of-work** - we will be going into detail in the next segment

TWO PHASE COMMIT

Volunteers Needed!

SEGMENT I - BREAK TIME

QUESTION & ANSWER SESSION

15 mins

BITCOIN REVIEW

Understanding how it works

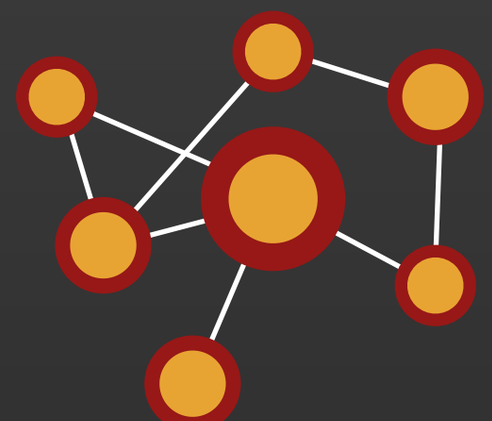
OVERVIEW

- Bitcoin Traits
- Block format
- Transaction format
- Inputs & Outputs
- Mining
- Proof-of-work
- Data Immutability

BITCOIN TRAITS

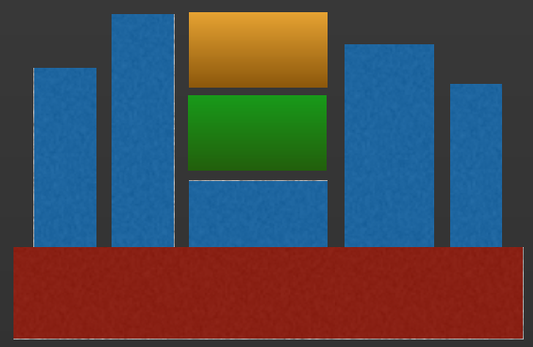
Bitcoin stores a ledger

- Each node stores a complete copy of all transactions
- Miner nodes work to secure the ledger from attacks and also mint new coins



Bitcoin is a network

- Fully distributed, no central authority controls it
- Open participation where anybody can join, or even start mining coins

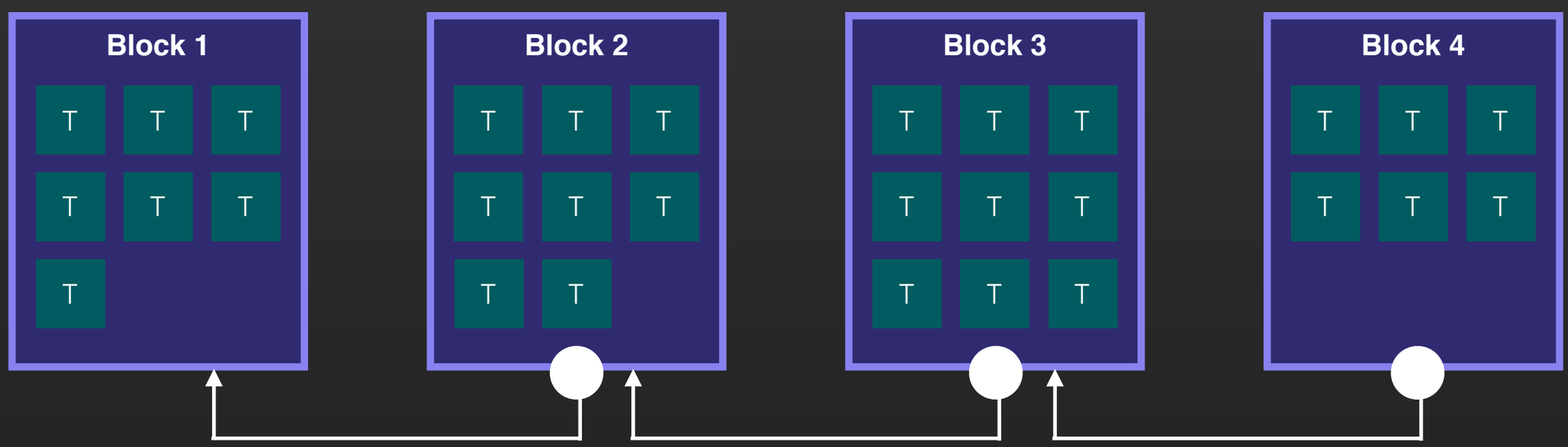


Bitcoin is p2p digital cash

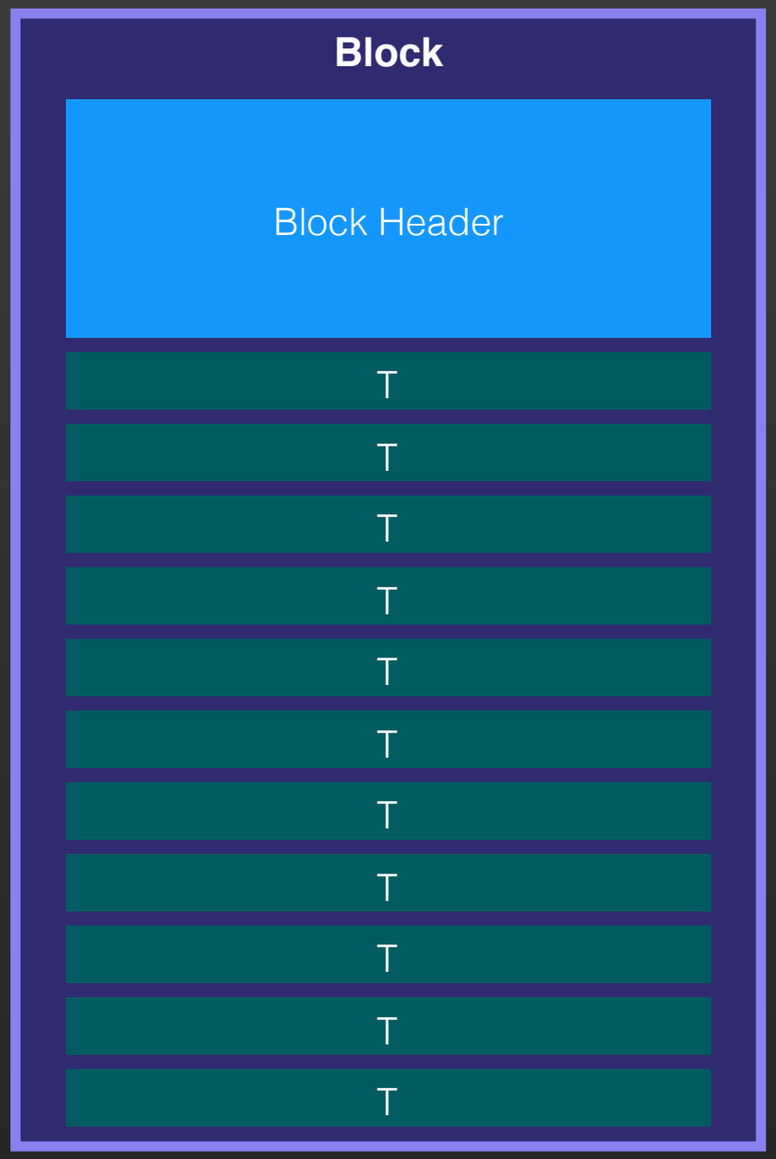
- Enables almost instant payments to anybody anywhere in the world
- Has escrow and multi-signature features, thus making it possible to automate settlements

DIVING DEEPER INTO BITCOIN

- A bitcoin node stores a ledger of all transactions that has ever happened
- The ledger is made out of blocks
- Each block contains a series of transactions at a specific point of time
- Each block contains condensed information about the previous block
  - This creates a chain of blocks
  - Chain cannot be broken easily
  - Unbreakable chain leads to immutability of data



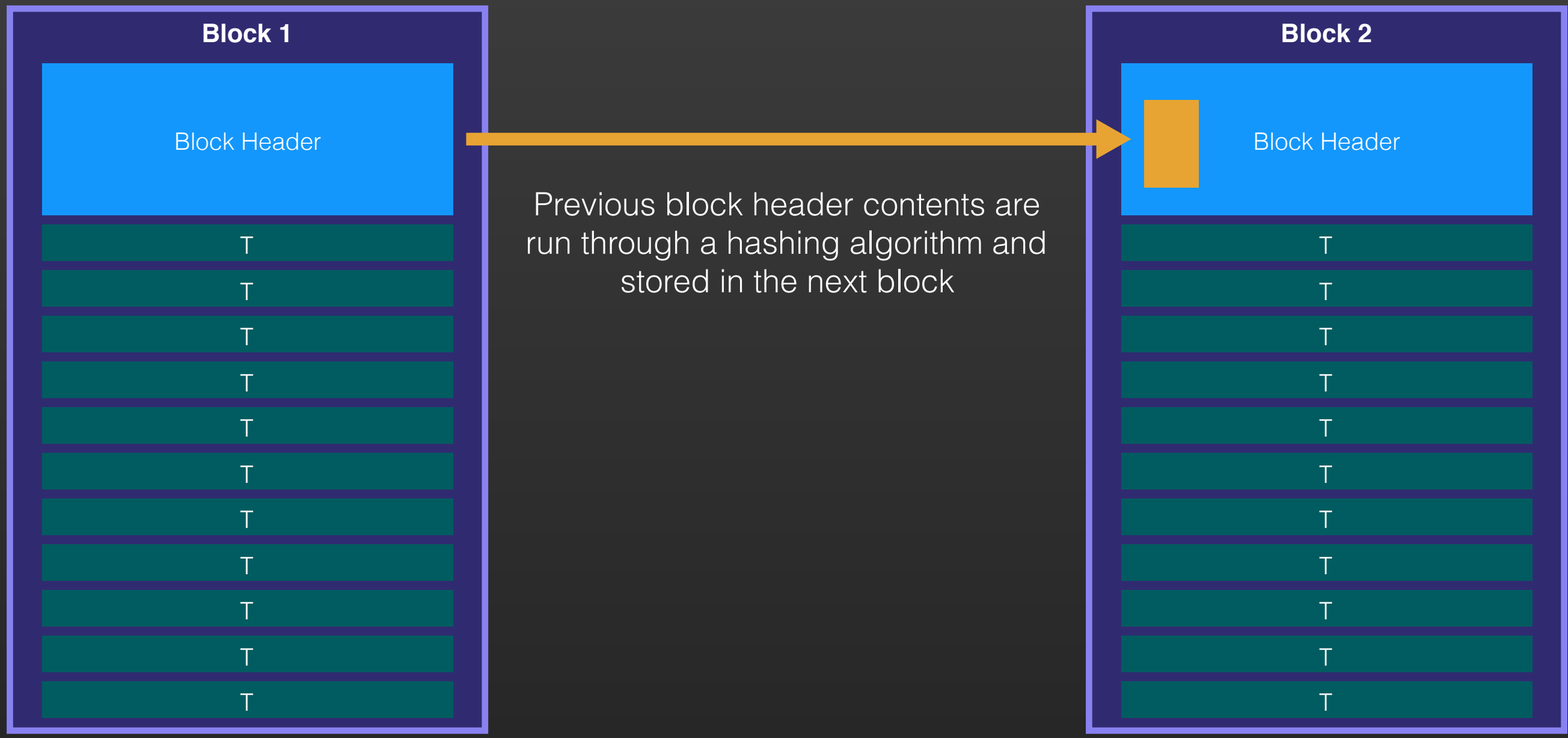
BLOCK FORMAT



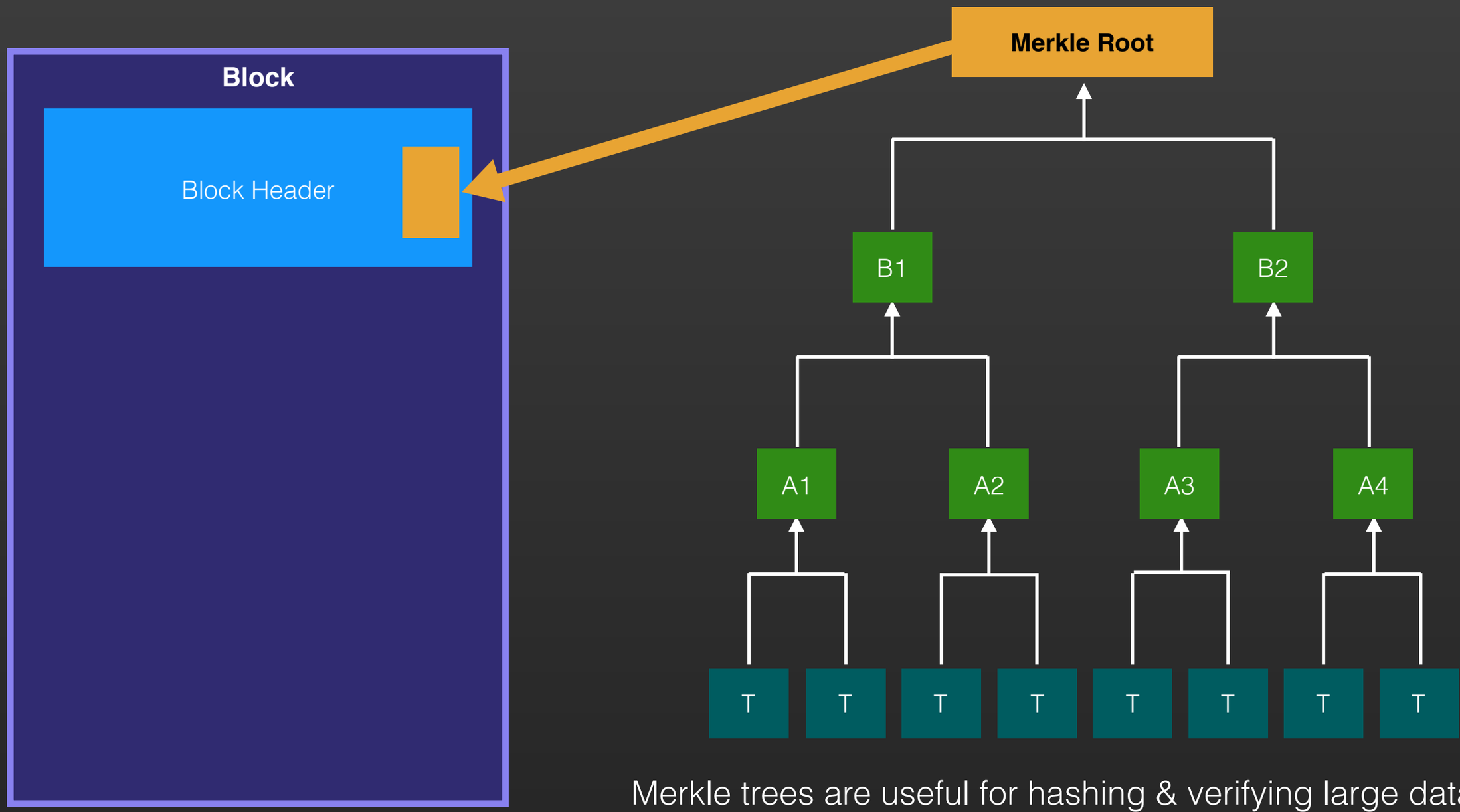
Block Format

- Blocks are the most basic primitive in bitcoin
- Each block is composed of a block header + a list of transactions
- Block header contains:
  - Version
  - **Hash of previous block header**
  - **Merkle root hash**
  - Time
  - Block Difficulty
  - **nonce**

BLOCK FORMAT

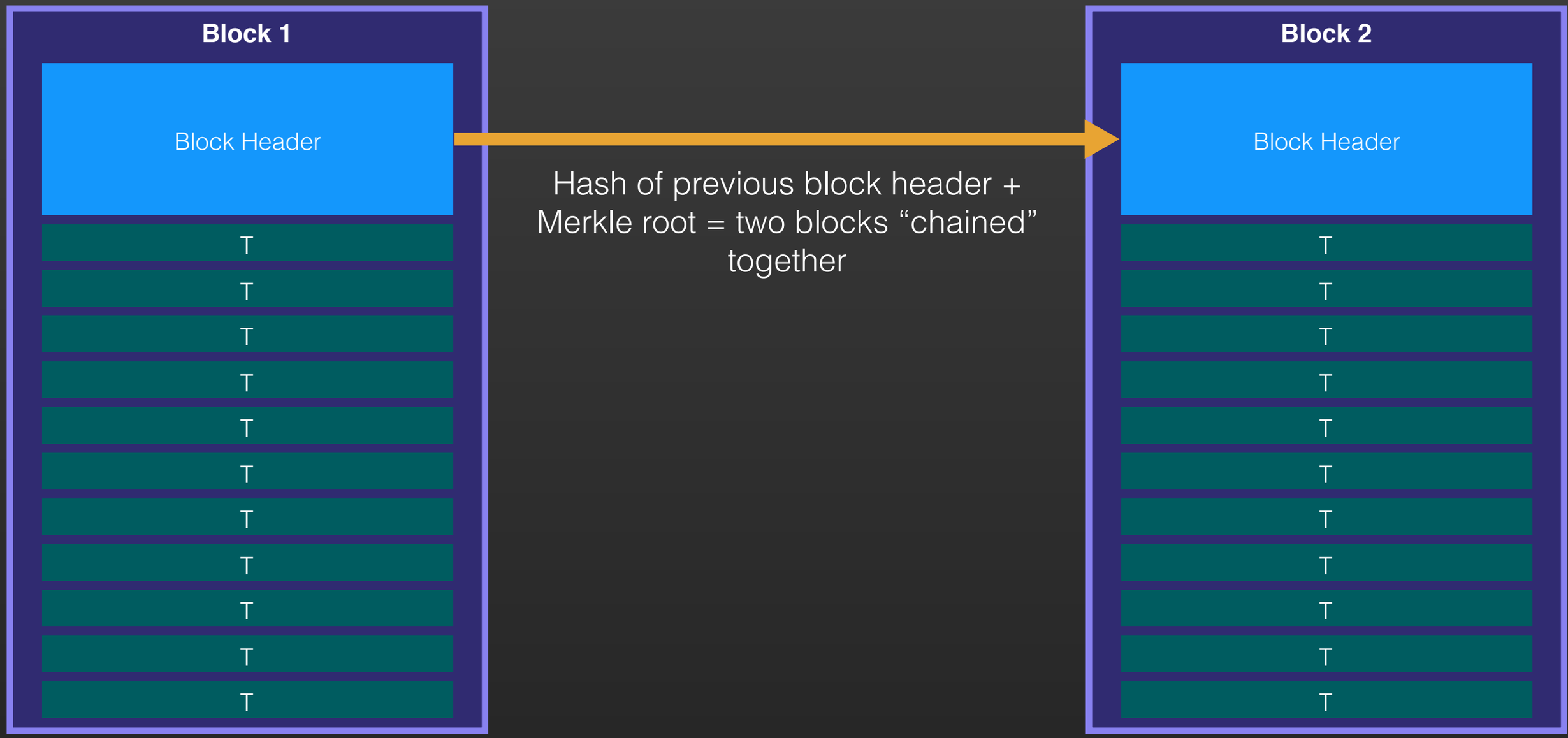


BLOCK FORMAT

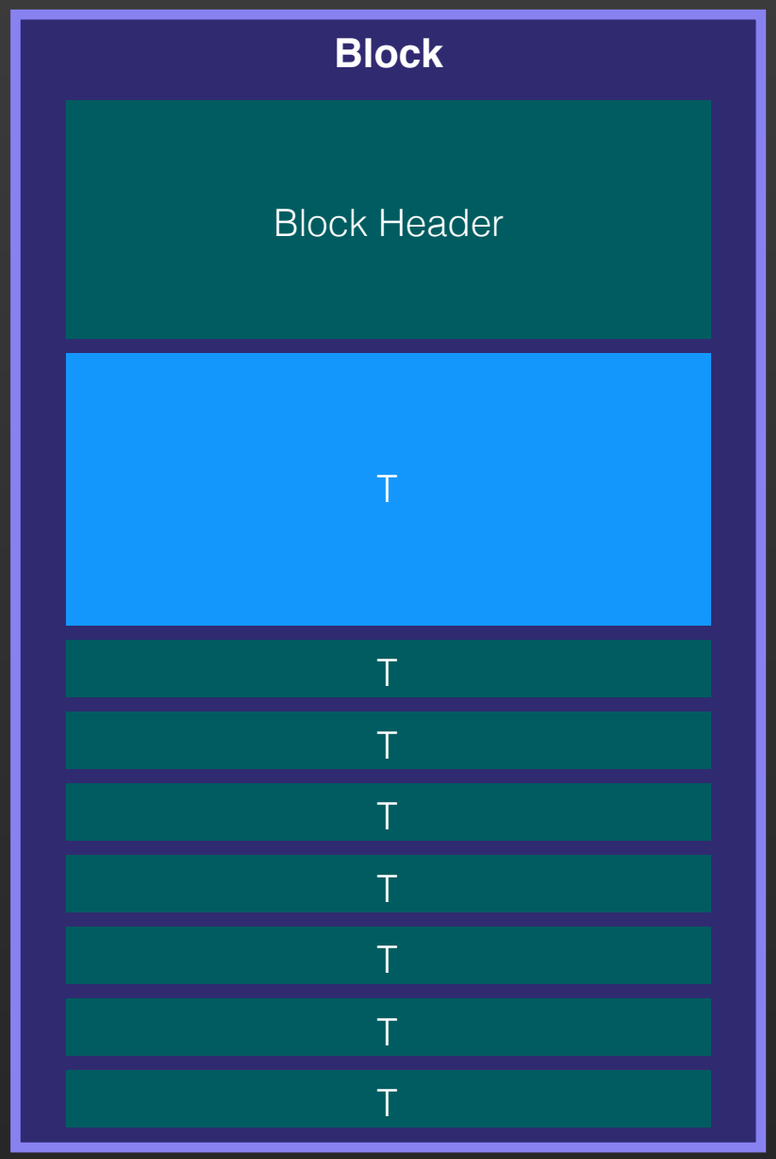


Merkle trees are useful for hashing & verifying large datasets

BLOCK FORMAT

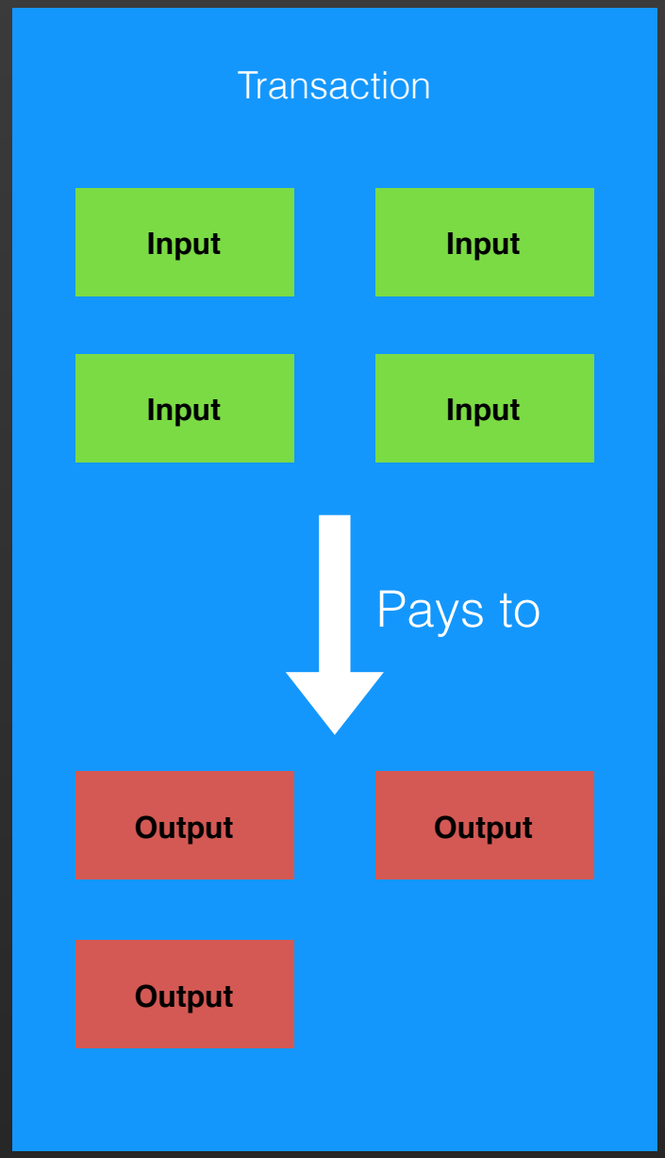


TRANSACTION FORMAT

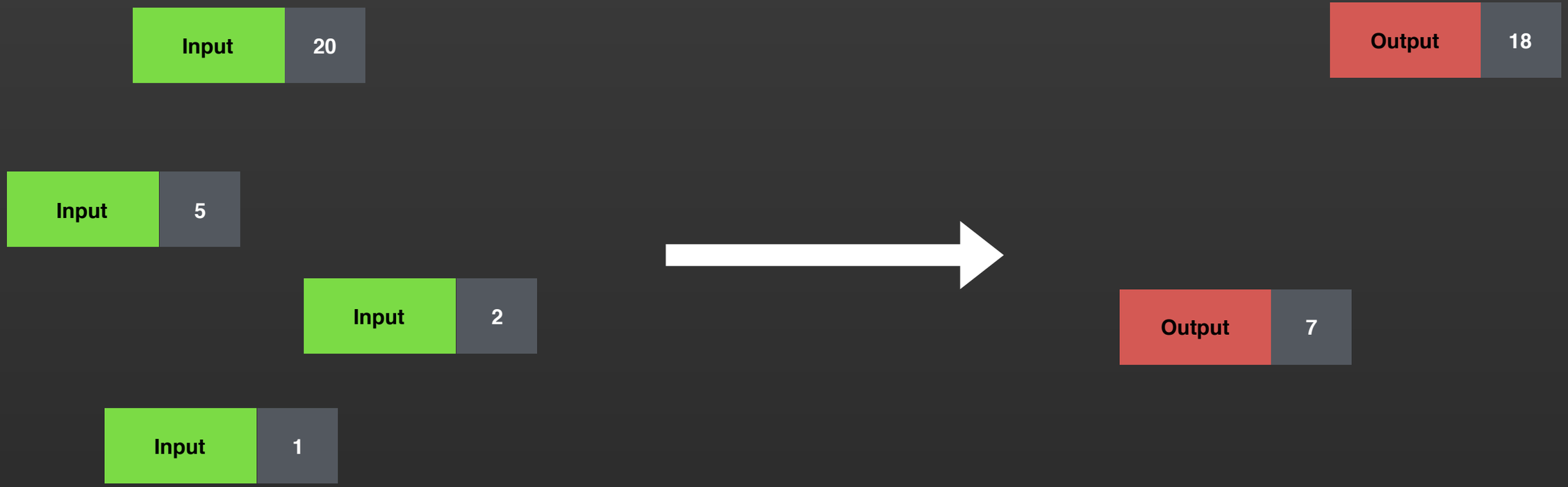


Transaction Format

- Version
- Total number of inputs
- List of inputs
- Total number of outputs
- List of outputs
- Lock time

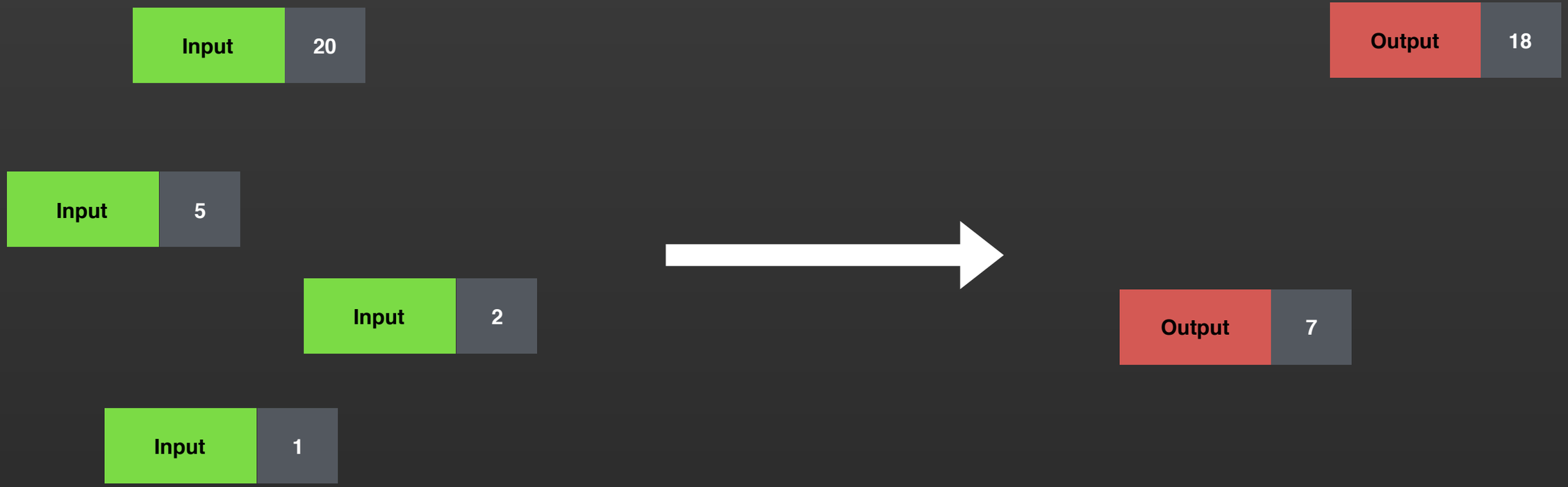


TRANSACTION - INPUTS AND OUTPUTS



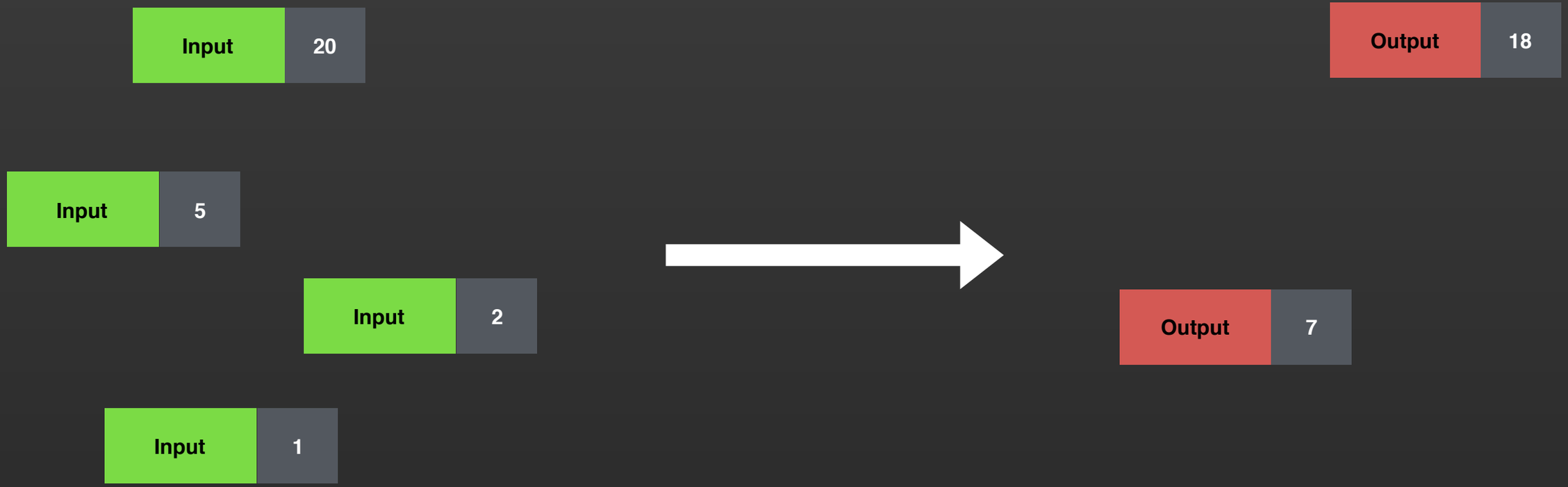
Total input ?

TRANSACTION - INPUTS AND OUTPUTS



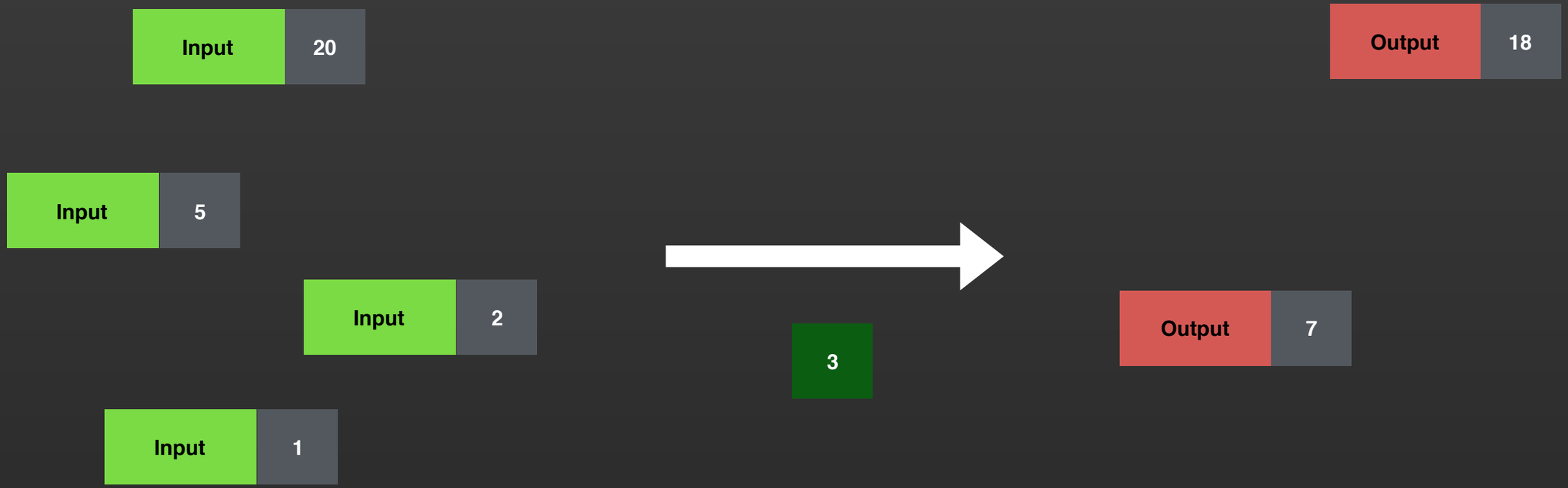
Total output ?

TRANSACTION - INPUTS AND OUTPUTS



Total fees ?

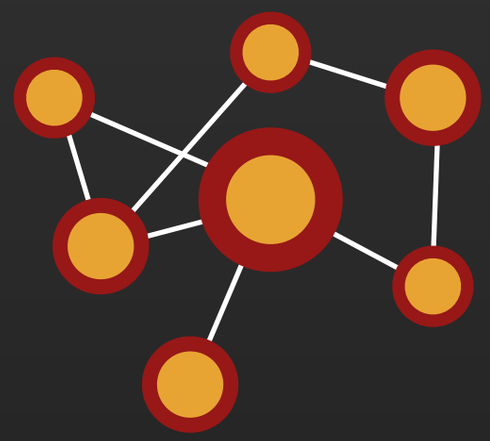
TRANSACTION - INPUTS AND OUTPUTS



Fees = Total input - Total output
= 28 - 25
= 3

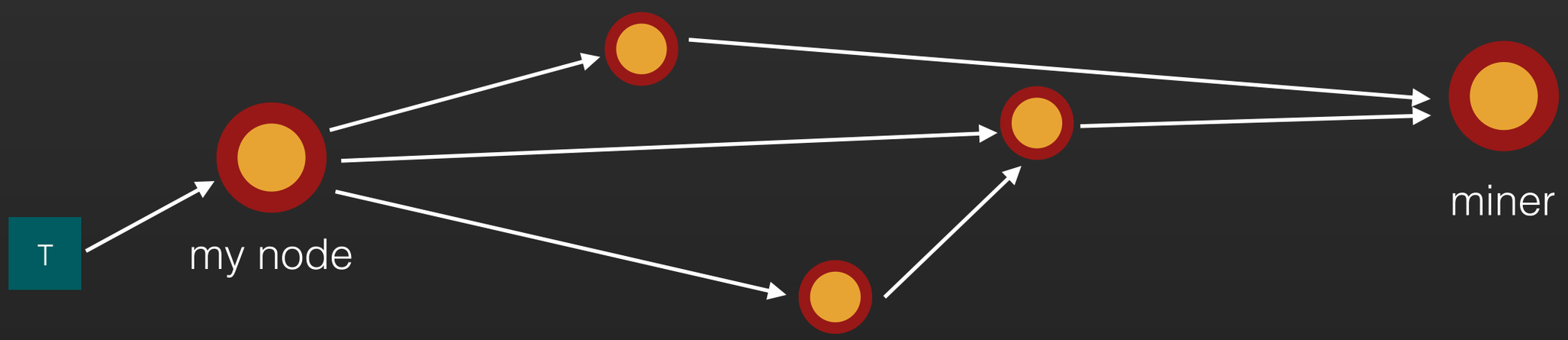
BITCOIN NETWORK

- Each node is connected to one another, maintaining a list of peers
- Nodes stay consistent with each other through a consensus algorithm called proof of work
- New transactions can be created on any node, and once created is relayed to other nodes
- Transactions are packed into new blocks on the blockchain through a process called mining



PEER TO PEER GOSSIP

- New transactions can be created on any node, and once created is relayed to other nodes
  - Any node can create a transaction
  - New transactions are relayed to other nodes using a gossip protocol
  - New transactions reside in a **pool**
  - Miner nodes eventually will receive a copy of the transaction

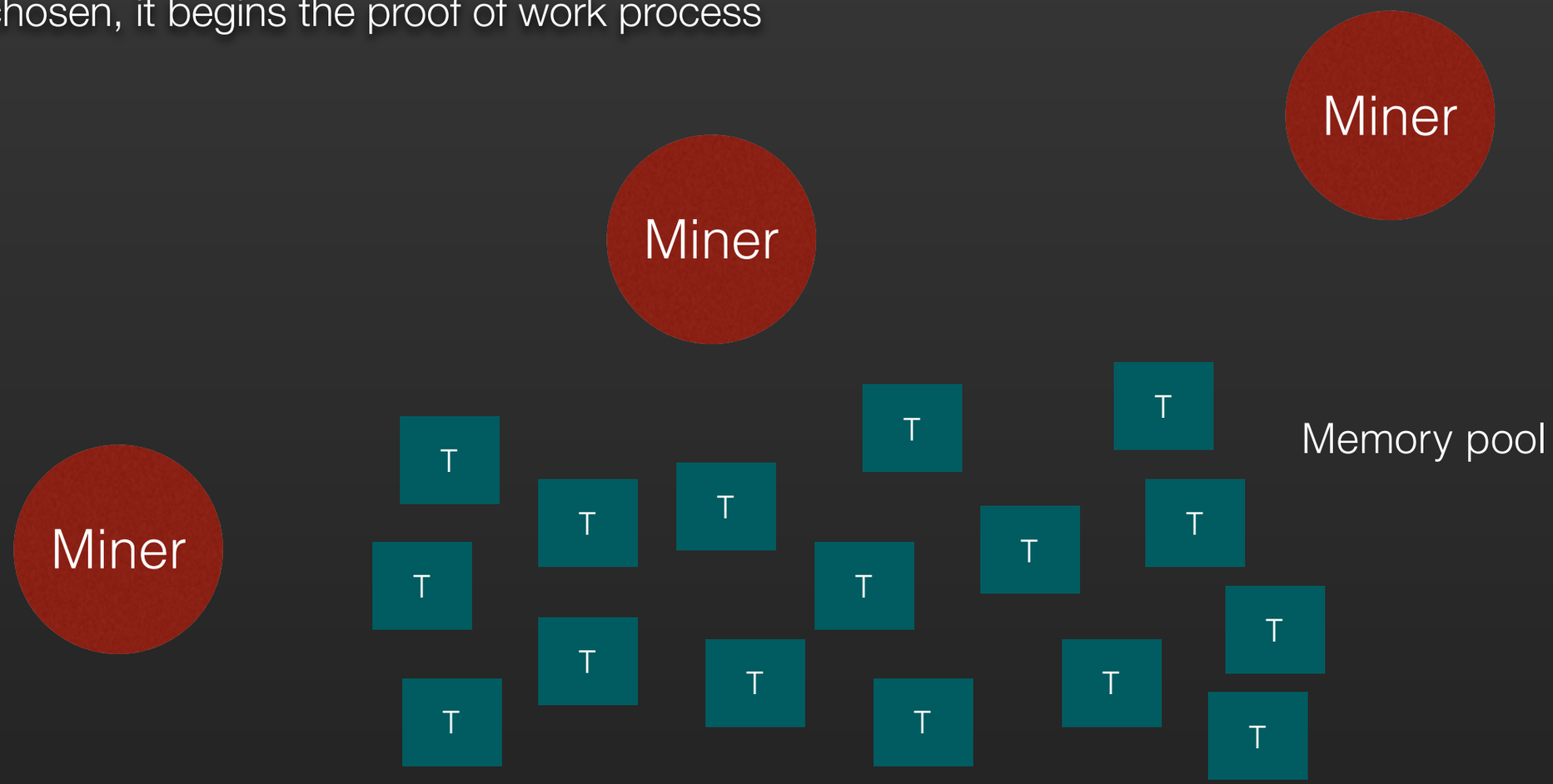


MINING

- Miners are special nodes with the capability to **add blocks to the chain**
- Miners ensure that everybody's data is always consistent
- Miners are also a source of new coins
    - Every time a new block is added to the chain, the miner that added it is rewarded with newly created bitcoins
- Miners must perform a **proof-of-work** to gain the right to add a block to the chain

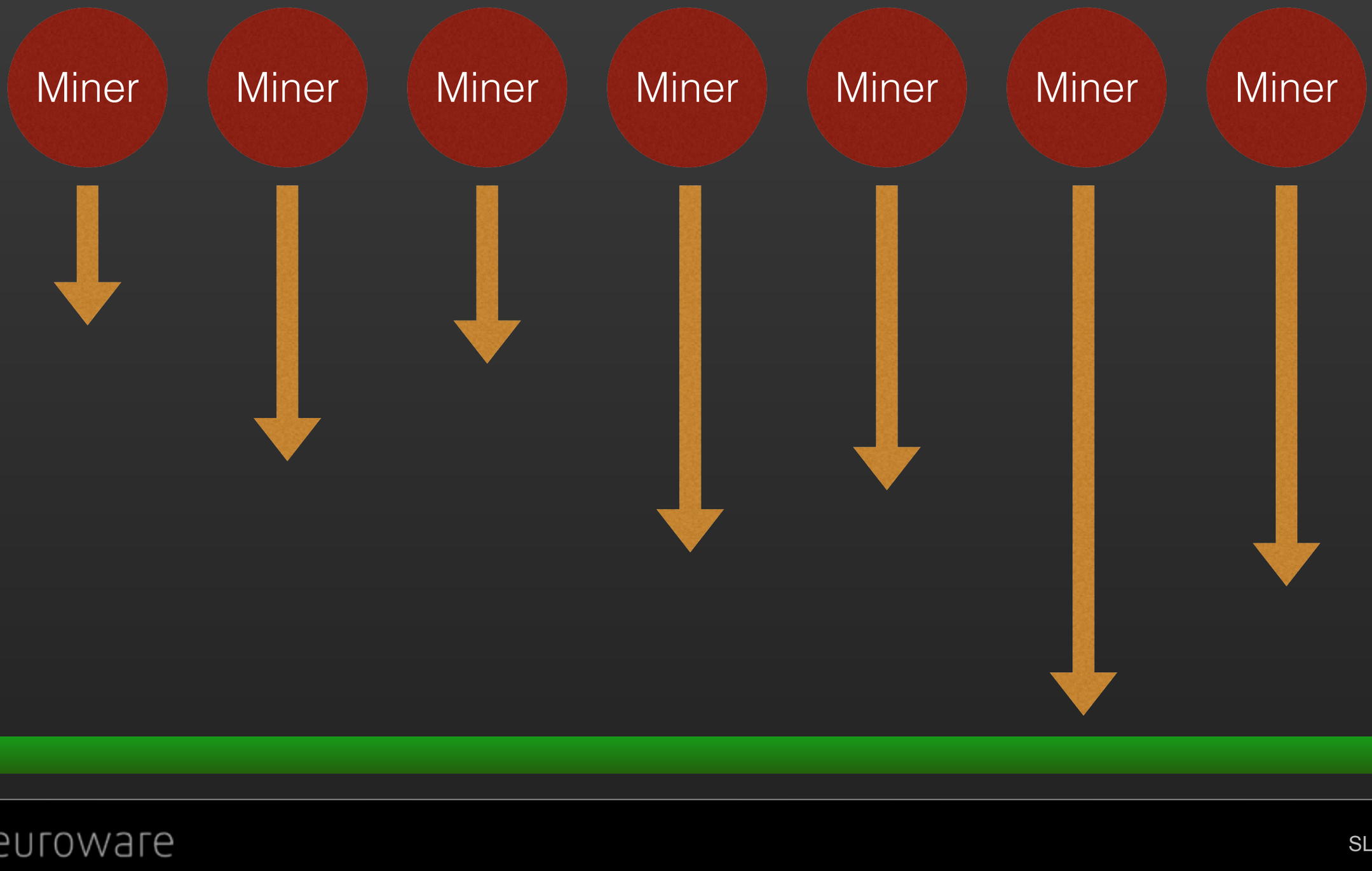
MINING

- All unconfirmed transactions goes into a memory pool
- Each miner must pick a list of transactions to create a block
- Different miner have different criteria of which transactions to pick
- Once a transaction is chosen, it begins the proof of work process



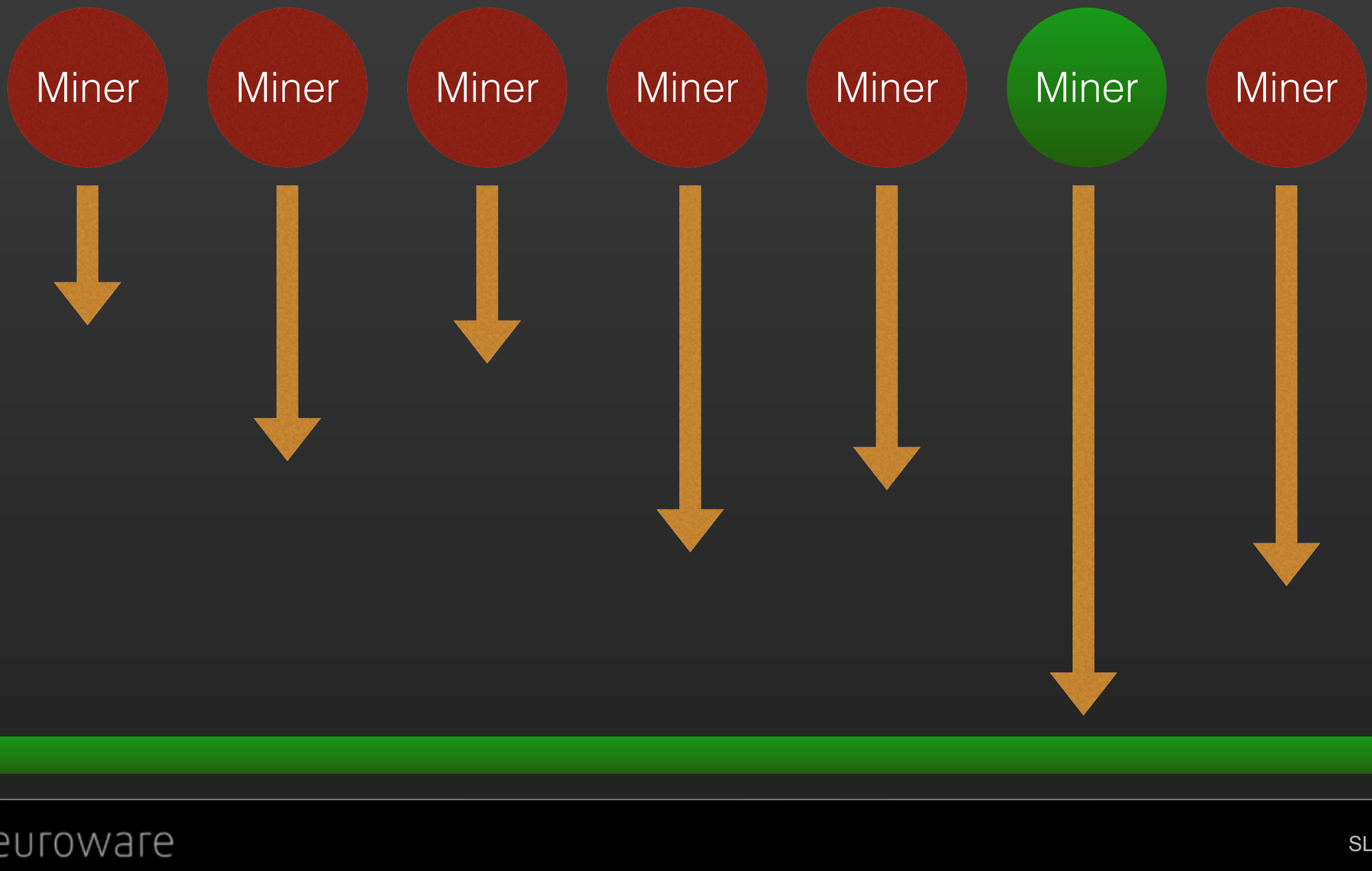
MINING PROCESS

Thousands of miners race to solve a math puzzle



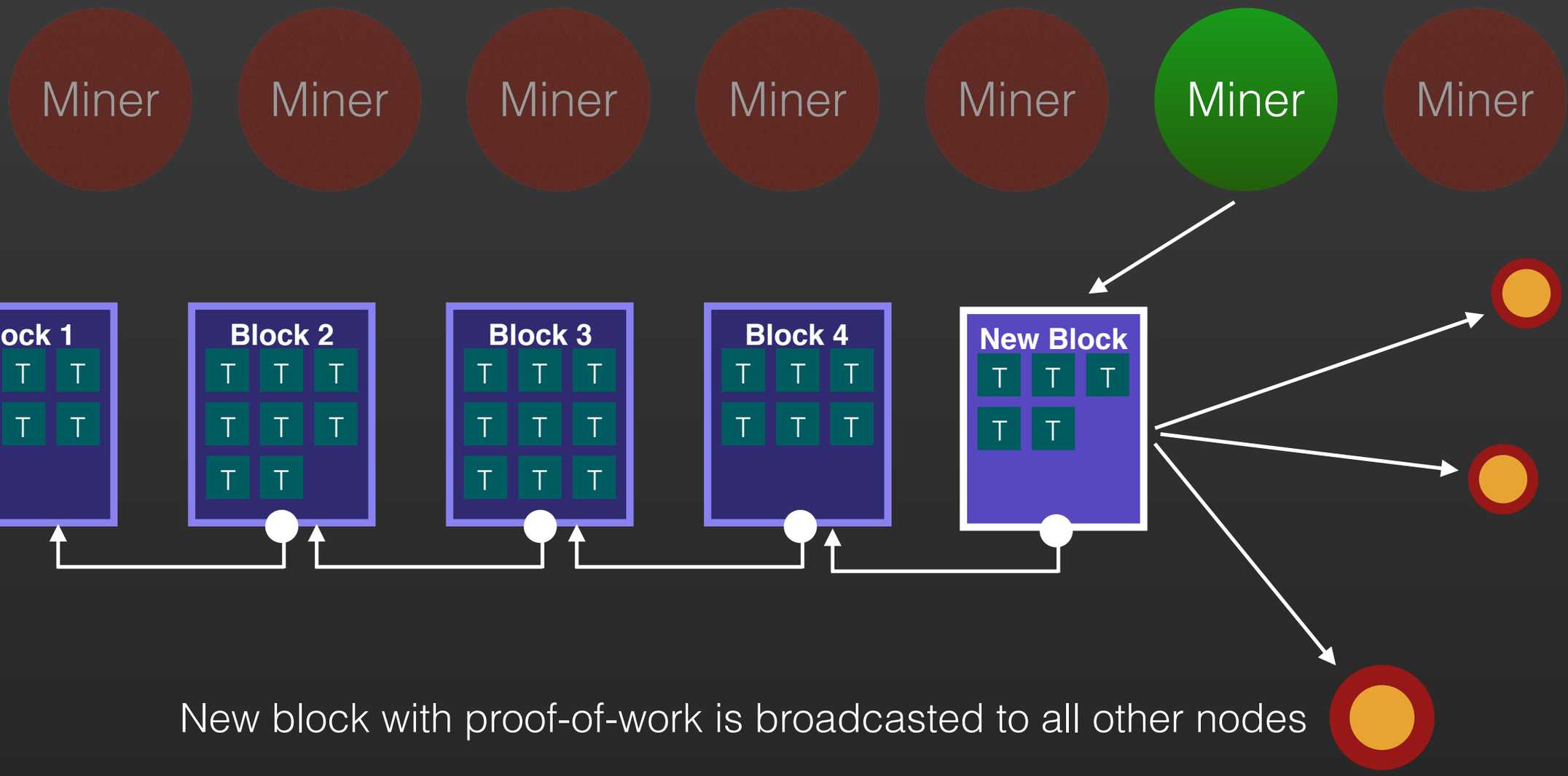
MINING PROCESS

First to solve gets the right to **add a new block**



MINING PROCESS

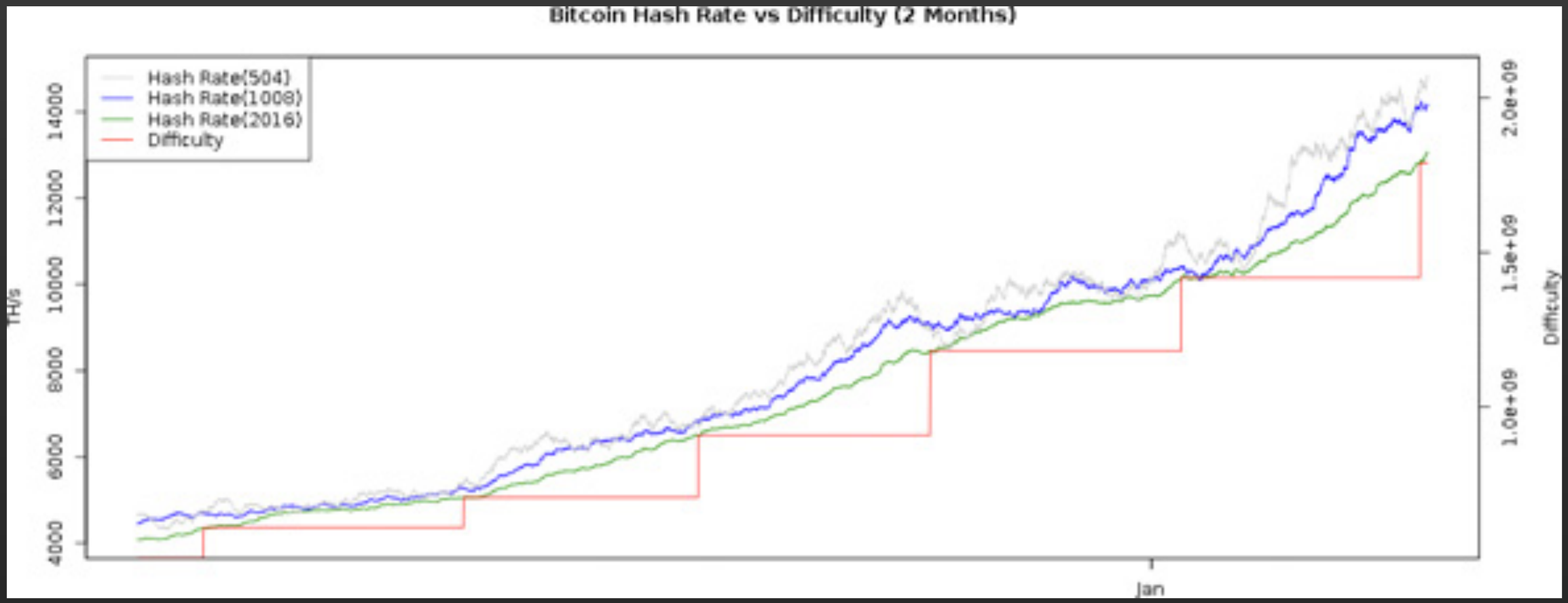
Miner is rewarded with **new coins** and **transaction fees**



New block with proof-of-work is broadcasted to all other nodes

AUTO-ADJUSTING DIFFICULTY

Auto-adjusting difficulty acts as **traffic control**



EARLY MINING RIGS

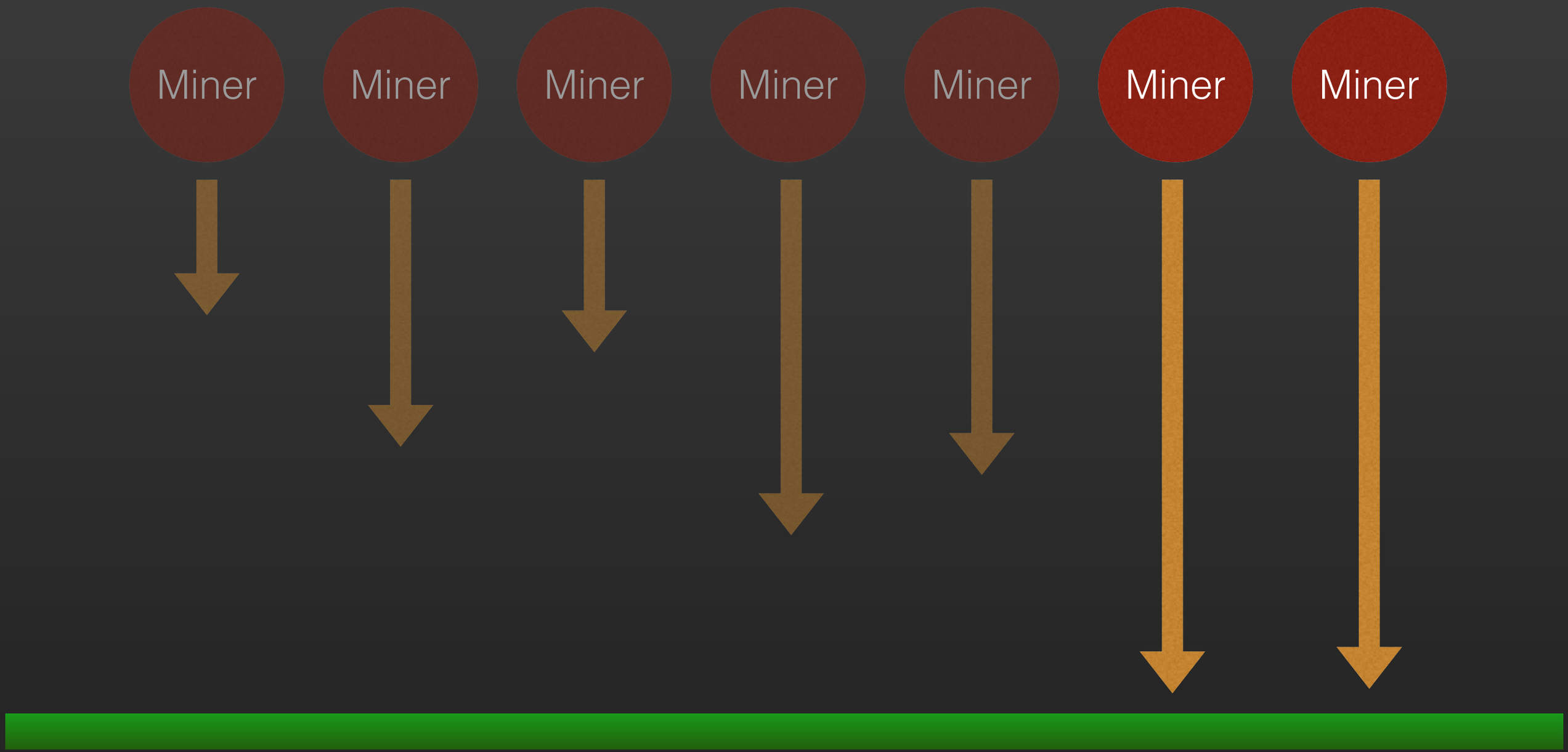
MODERN MINING RIGS

PROOF OF WORK

2 Volunteers Needed!

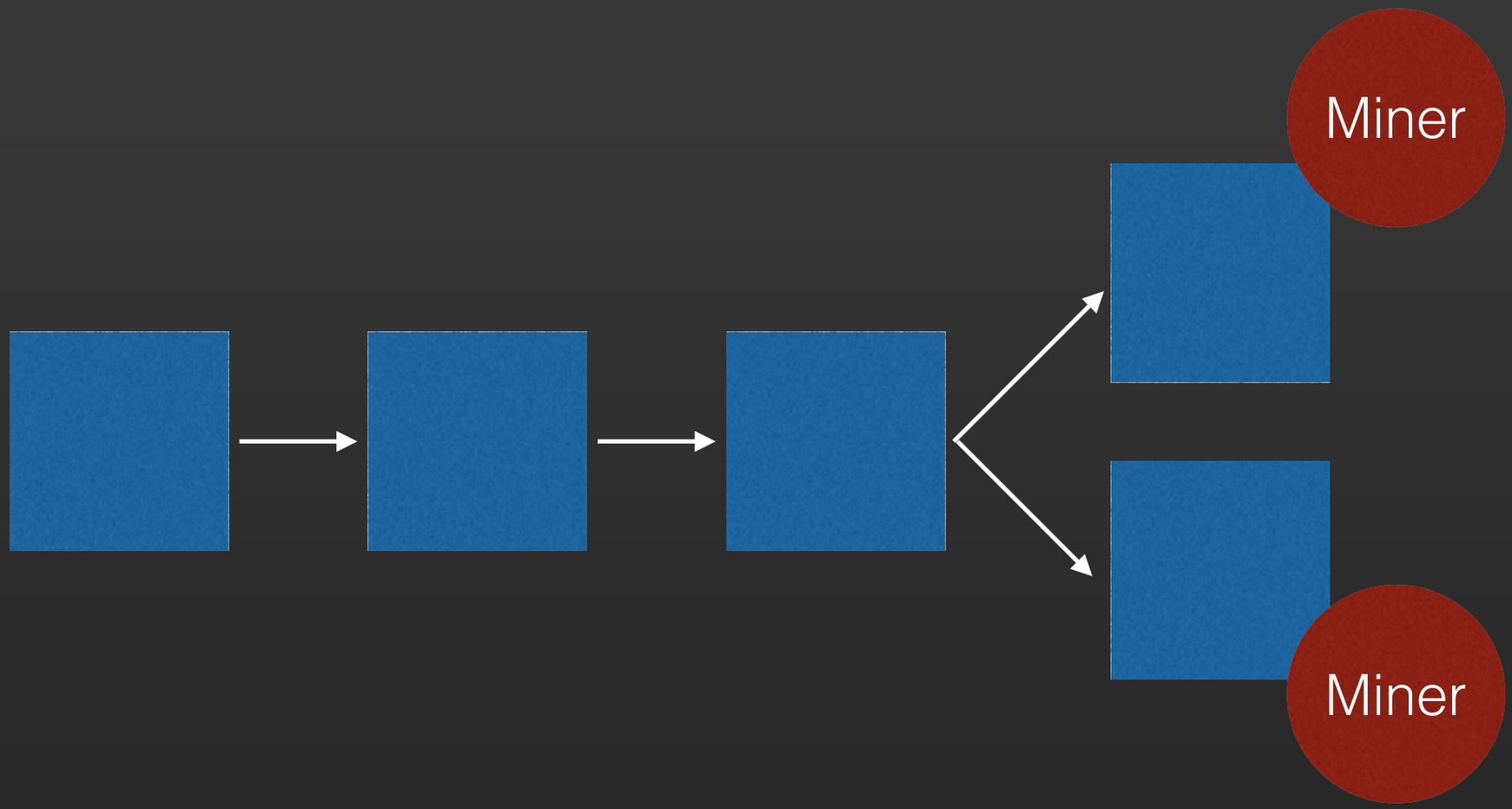
FORKING

What happens when 2 miners find the solution simultaneously?



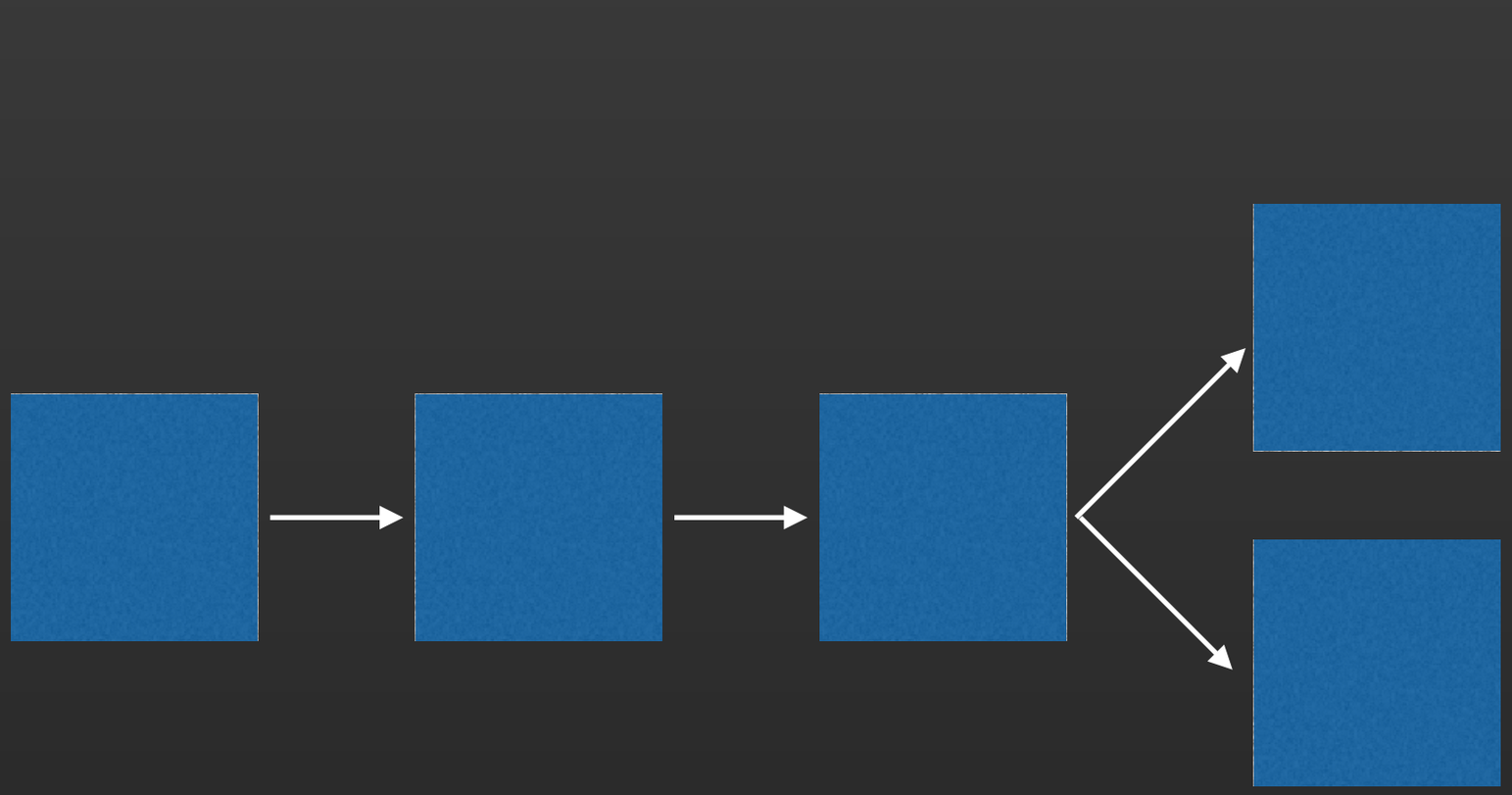
FORKING

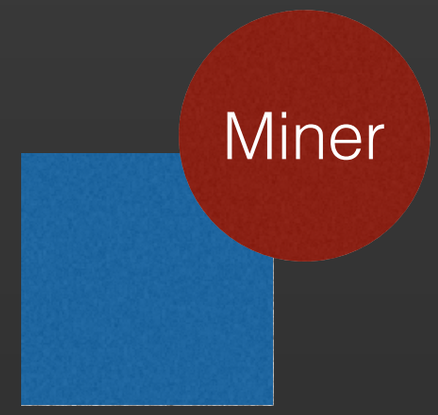
A fork happens!



FORKING

Longest-chain protocol comes into effect

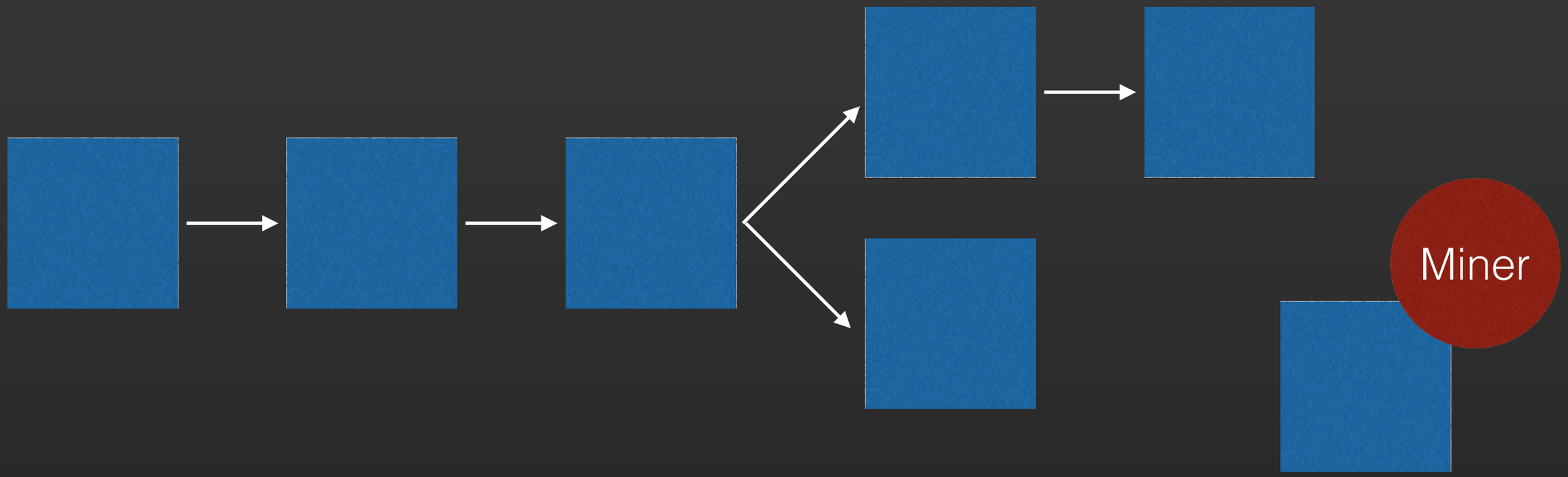




Randomly picks one chain if both forks are the same length

FORKING

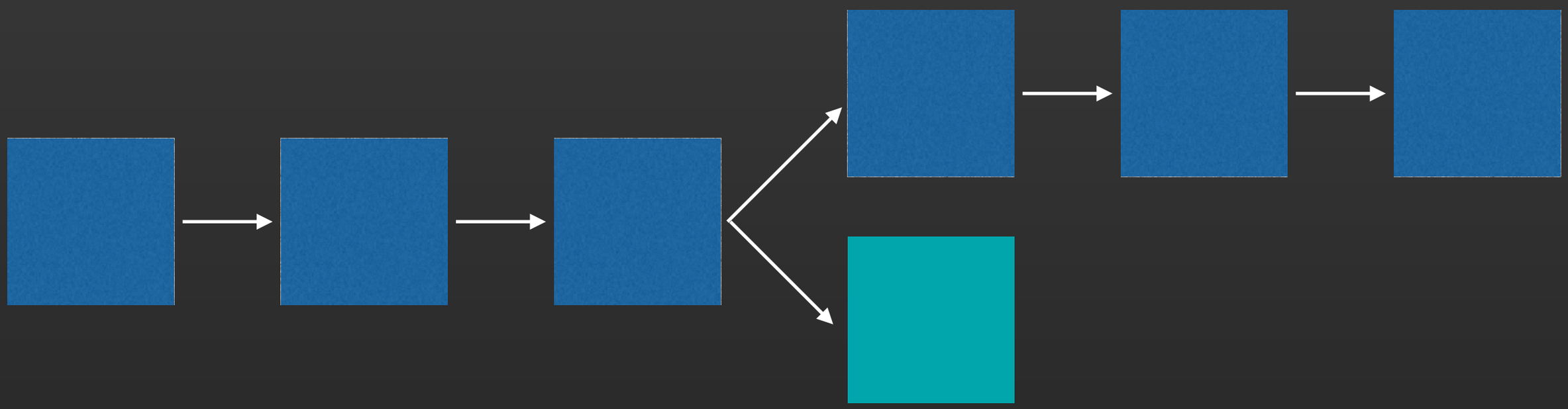
Longest-chain protocol comes into effect



If one fork is longer than the other, pick the longest fork

FORKING

Longest-chain protocol comes into effect



This block is now orphaned! After a while, all transactions within are returned to the memory pool

DATA IMMUTABILITY

- 3 factors gives bitcoin **byzantine fault tolerance:**
  - Chain of blocks ensures that once data is stored, it cannot be changed, or the chain will break
  - Network of miners ensure that new blocks are valid in order to reap the rewards in bitcoin
  - Auto-adjusting difficulty ensures that block intervals grow in parallel to hashing power

- Attack vectors:
  - **Sybil attack** - many malicious nodes in order to perform localised malicious operations
  - **Denial-of-service** - transaction spam, protocol attacks, resource starvation
  - **Quantum computing** - probably not a problem in the short-term
  - **51% attack** - achieve quorum power, needs a lot of money ($461,684,264.00 hardware)
  - **Double spending** - consensus protocol ensures that double spends do not happen

SEGMENT 2 - BREAK TIME

QUESTION & ANSWER SESSION

15 mins

LUNCH BREAK

Be back in 1 hour

SMART CONTRACTS

What are they and how do they work

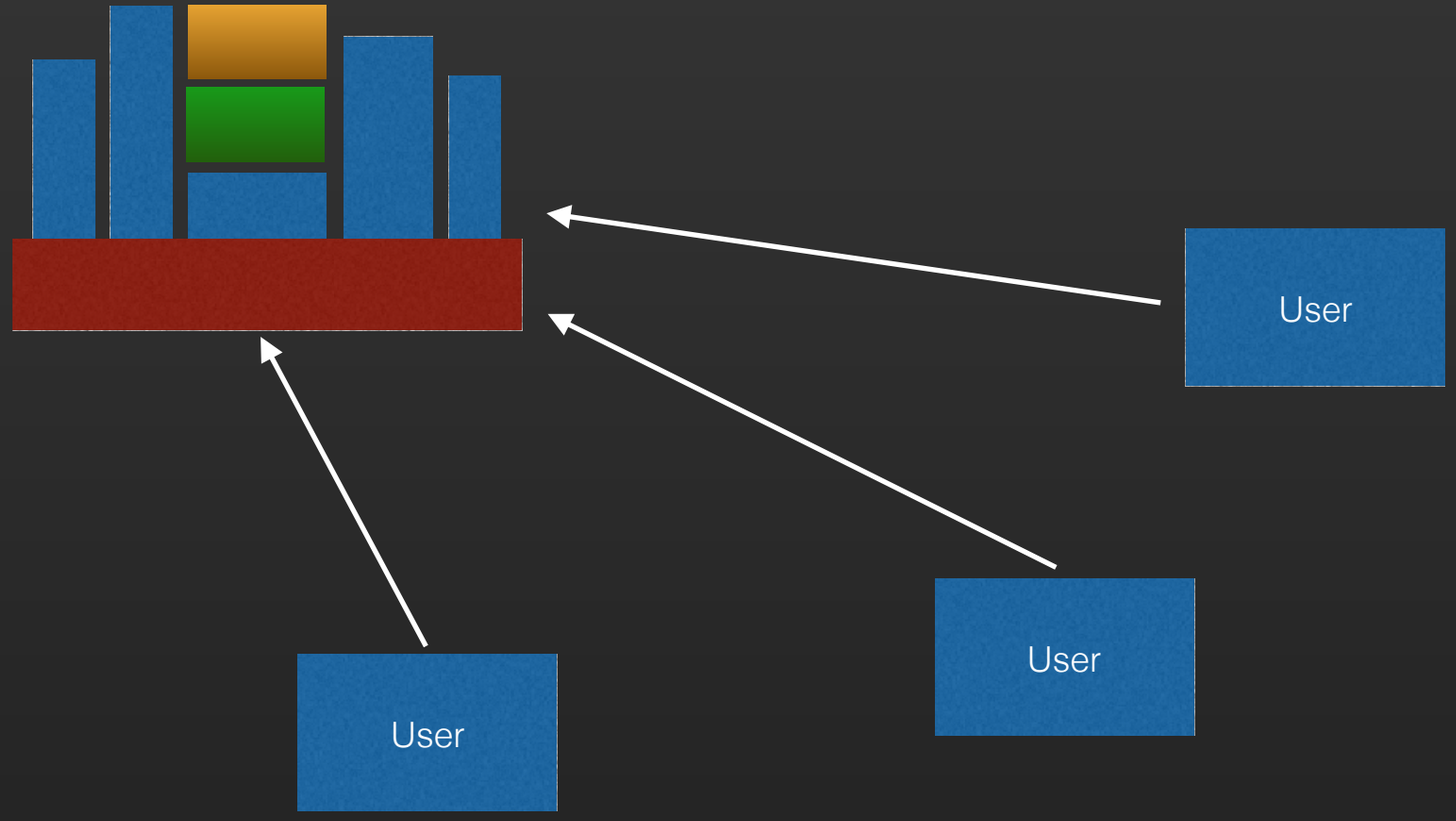
OVERVIEW

- Conventional contracts
- Payment contracts
- Smart contracts

CONVENTIONAL CONTRACTS

Conventional service deployment

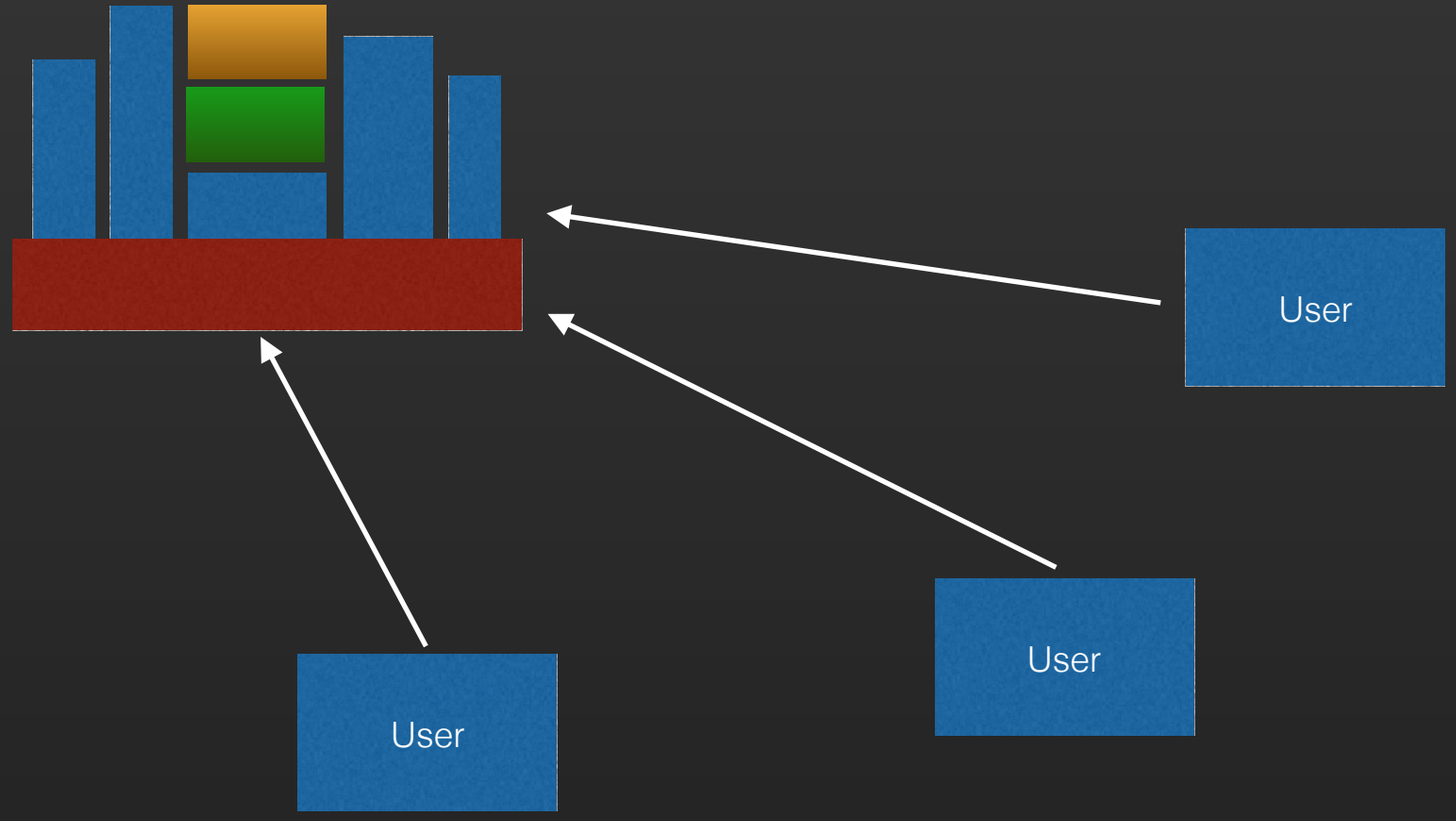
- Requires centralised infrastructure
- Requires one to trust the owner of the infrastructure
- Owner of the infrastructure needs to worry about security
- Infrastructure is a single-point-of-failure



CONVENTIONAL CONTRACTS

Technology is supplemented with human interaction

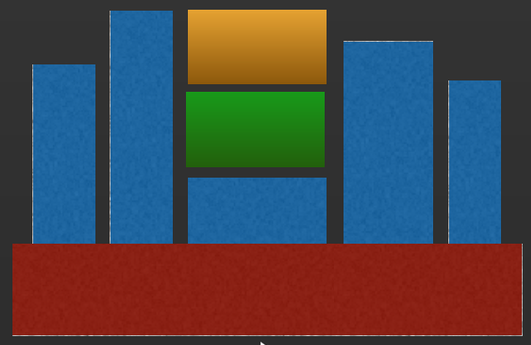
- 100% automation is not possible
- The more mission critical, the more human involvement
- Difficult to guarantee availability under scale



CONVENTIONAL CONTRACTS

User can be compromised

Server can be misconfigured or attacked



Man-in-the-middle attacks

User

Transport layer can be flooded

User

User

Avenues of Attack

SMART CONTRACTS

- Code is hosted in a **distributed** computing network
  - Immutability guarantee prevents one from tampering the code

- Execution is **deterministic** and predictable
  - Immutability guarantee allows code to execute as expected

- Can be **expensive** to launch an attack
  - Network hashing power discourages malicious attacks and spam

- Interaction programmed into **protocol**
  - Limited instruction set leads to simpler system, harder to make mistakes

BITCOIN “SMART” CONTRACT

- Bitcoin has a programmable language:

```
OP_DUP
OP_HASH160
88b028348642ad1bbaa8fcc054273070eda045fe
OP_EQUALVERIFY
OP_CHECKSIG
```

- Can be used to program payment instructions
  - Escrows
  - Payment channels
  - Automated executions

- Multi-signature technology allows the involvement of trustees
  - x-of-y signatures to authorise payment
  - Automated refund after x time

MULTI-SIGNATURE TECHNOLOGY IN ACTION

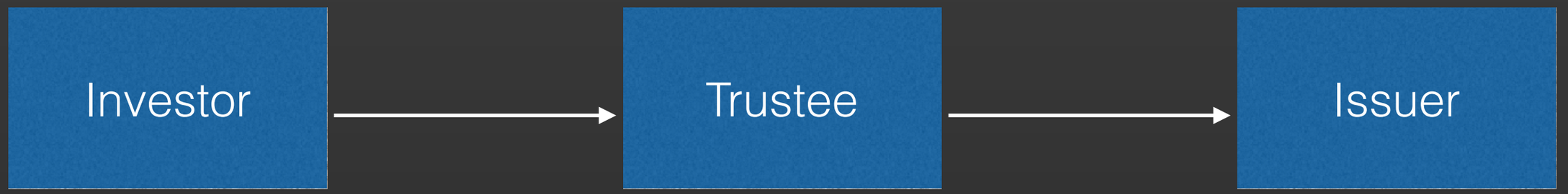
Multi-signature technology can be used to automate business processes.
Here we show a 2-of-3 multi-signature process in a crowd funding model.

Investor

Trustee

Issuer

MULTI-SIGNATURE TECHNOLOGY IN ACTION



Investor contacts trustee
to commit funds to an issuer

Operator sets up multi-signature
process with both parties

Funds are “locked” in a special payment contract that can be “unlocked” with 2-of-3 signatures

MULTI-SIGNATURE TECHNOLOGY IN ACTION

Funding success! Time to release funds from investor to issuer

Investor

Trustee

Issuer

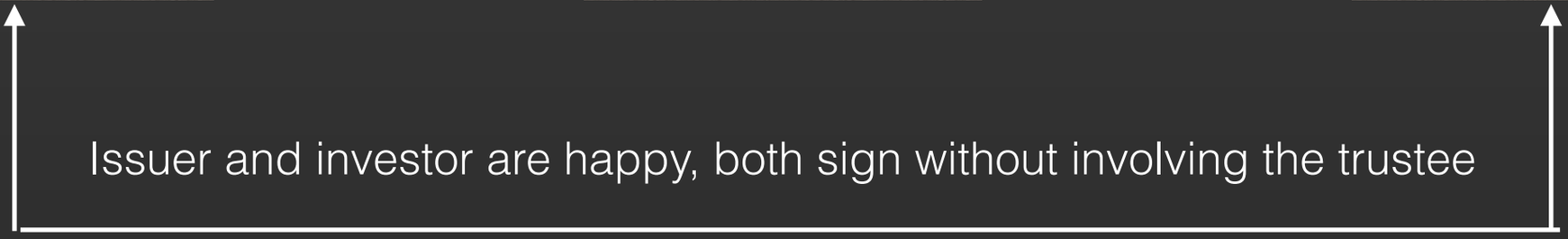
MULTI-SIGNATURE TECHNOLOGY IN ACTION

Funding success! Time to release funds from investor to issuer

Investor

Trustee

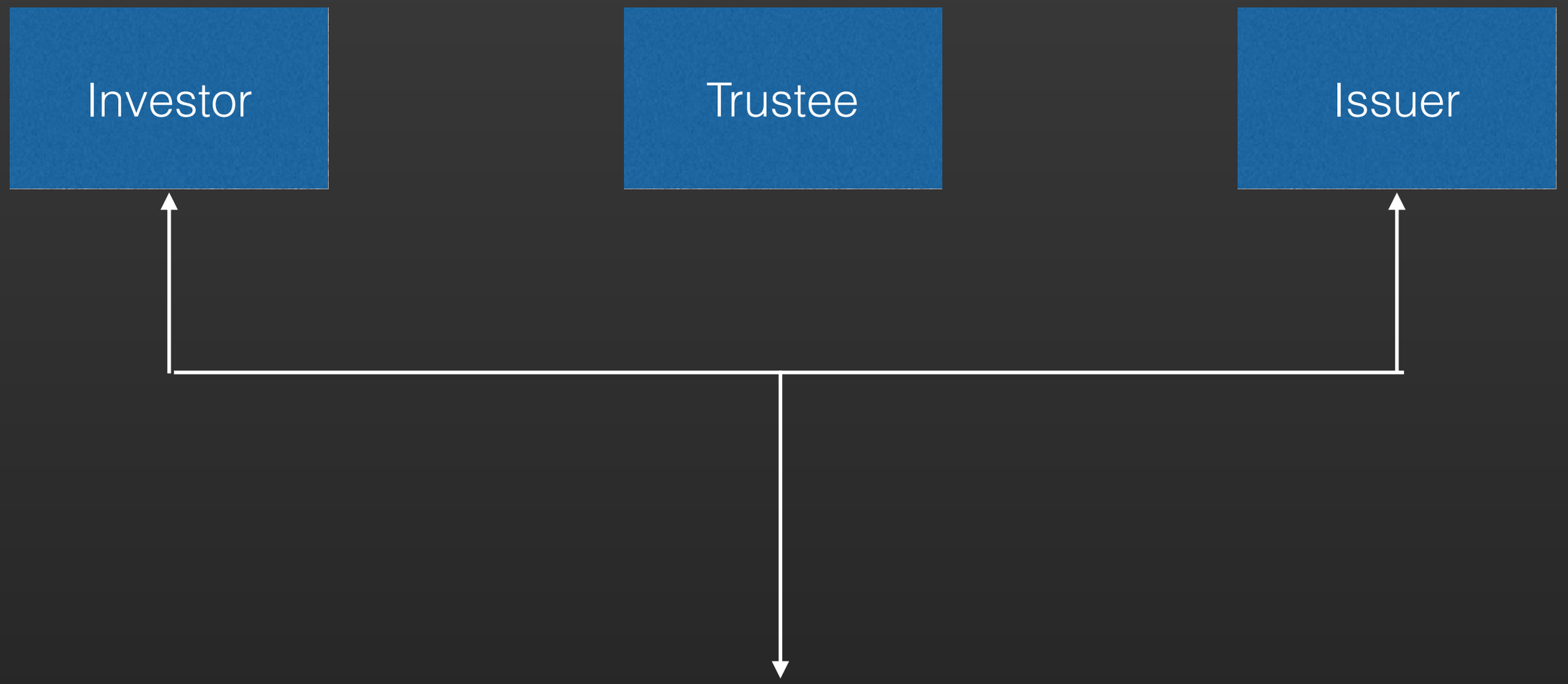
Issuer



Issuer and investor are happy, both sign without involving the trustee

MULTI-SIGNATURE TECHNOLOGY IN ACTION

Funding success! Time to release funds from investor to issuer



MULTI-SIGNATURE TECHNOLOGY IN ACTION

Funding failed! Time to return the funds to the investor

Investor

Trustee

Issuer

MULTI-SIGNATURE TECHNOLOGY IN ACTION

Funding failed! Time to return the funds to the investor

Investor

Trustee

Issuer

OH NO, Issuer refuse to sign because they suspect foul play!

MULTI-SIGNATURE TECHNOLOGY IN ACTION

Funding failed! Time to return the funds to the investor

Investor

Trustee

Issuer

nooooooooooooooooo

Trustee and investor signs anyways, following protocol and enforcing the contract.

USES OF MULTI-SIGNATURE SIGNING

Use cases

- Corporate procurement protocols (2-of-3 signature)
  - Purchase has a key
  - CEO has a key
  - Finance has a key
  - Purchaser + Finance sign for most procurements
  - Purchaser + CEO sign for big procurements

- Voting systems for AGM (50% signature)
  - Special smart contract
  - 6-of-11 signatures required
  - Once at least 6 signatures are signed, transaction is committed

OTHER "SMART" CONTRACTS

Bitcoin is developing lightning networks that allows micro-transactions or 0-confirmation purchases to happen.

BLOCKCHAIN SMART CONTRACT

ETHEREUM

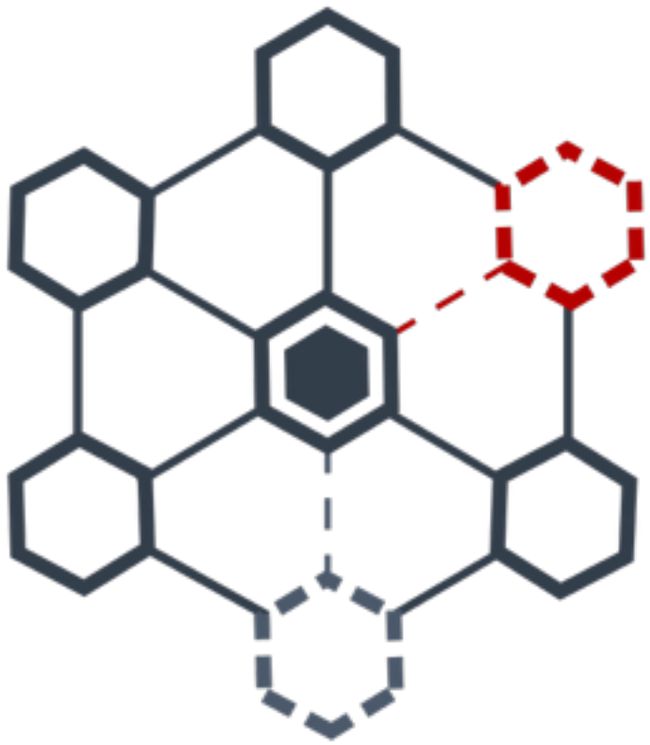
- Ethereum is a turing-complete distributed computing system, based on blockchain fundamentals

- Treats the blockchain network as a distributed computing platform
  - Allows the execution of code in public nodes in a deterministic manner
  - Allows anybody to put “contracts” on ethereum as long as they fulfil the protocol

- Creates an overlay network on top of a blockchain to give it extra capabilities
  - Contracts are “mined” into the ethereum blockchain
  - Once on the ethereum, the contract can be executed any time
  - Execution of contract consumes an internal resource called ether

- Ethereum can be used to create new digital tokens
  - Most popular (and technically sound) way to create custom digital tokens
  - Used in many distributed loyalty programmes
  - One of the many things you can do in Ethereum

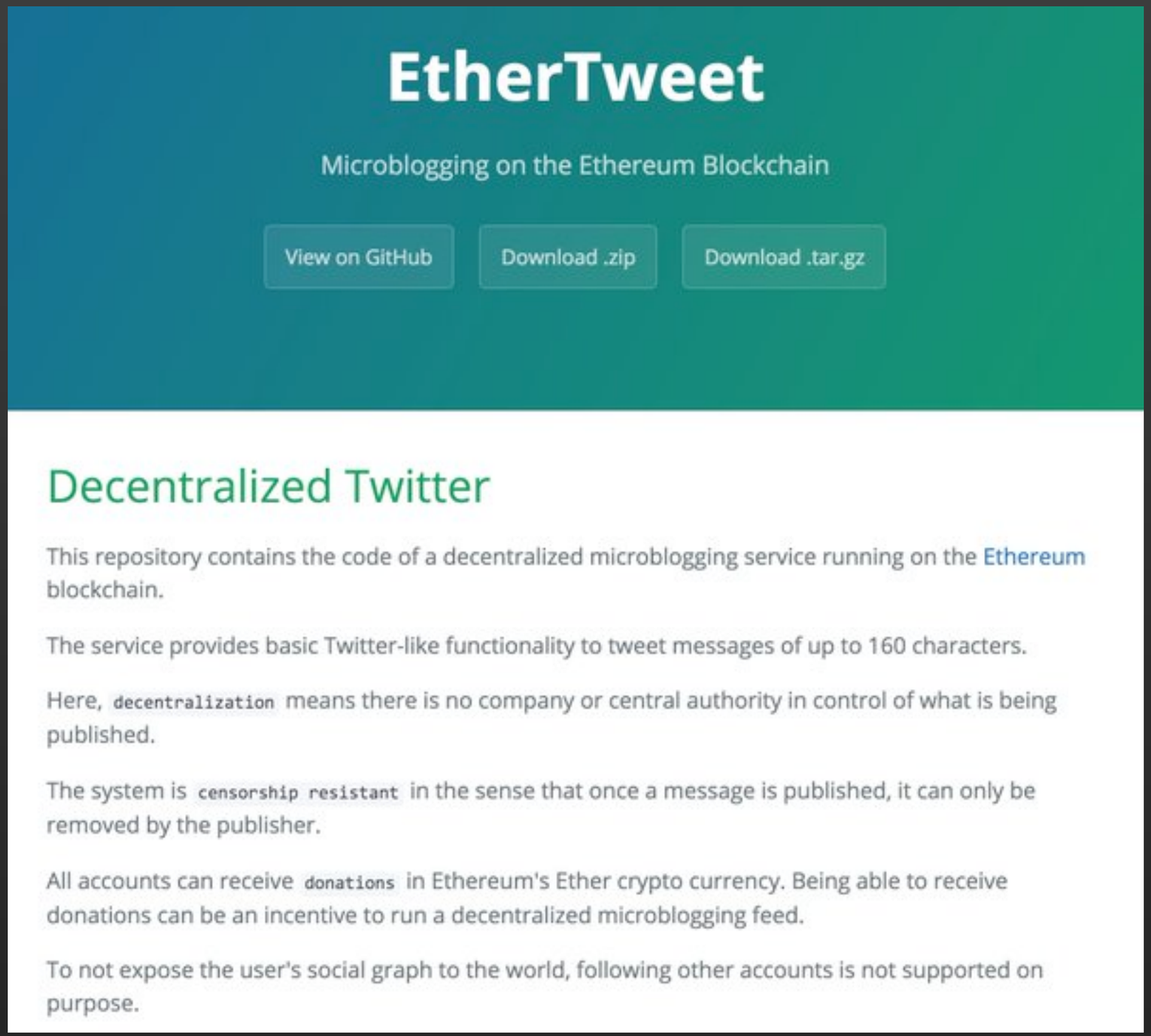
EXAMPLES OF DISTRIBUTED APPS



KYC-CHAIN

EXAMPLES OF DISTRIBUTED APPS

EXAMPLES OF DISTRIBUTED APPS



SEGMENT 3 - BREAK TIME

QUESTION & ANSWER SESSION

15 mins

USING CRYPTOCURRENCIES

Blockchains from the user's eyes

OVERVIEW

- Wallets
- Sending/Receiving Tokens
- Crypto-exchanges
- Key Management

WALLETS

- All blockchains can be interacted in two ways:
  - Programmatically (API calls)
  - Transaction (Wallet)

- Wallet is a generic term used to refer to the pool of private keys that only you own
  - You are 100% responsible for your private keys
  - Lose your private keys = lose your “money”

- Special terms
  - Accounts
  - Private Keys
  - Public Keys
  - Addresses

SETUP YOUR FIRST WALLET

Let’s get paid!

SENDING/RECEIVING TOKENS

- Participants within the same blockchain using the same cryptocurrency can send each other tokens
  - No central authority needed
  - All wallet clients must conform to protocol

- Transaction and Settlement have very little interval
  - Transaction is made from the user's own node
  - Transaction is settled when enough confirmations are achieved

- Transfers can be programmed and automated
  - Multi-signature/payment contracts/smart contracts can be used
  - Some wallets allow programming of smart contracts

SEND TOKENS TO ONE ANOTHER

Get the economy working!

TYPES OF WALLETS

Light Wallet

Full Wallet

Hosted Wallet

TYPES OF WALLETS

Light Wallet

- Only stores a limited snapshot of data relevant to the user

- Can easily be embedded on limited devices

- Does not contribute to the ecosystem's integrity

Full Wallet

Hosted Wallet

TYPES OF WALLETS

Light Wallet

Full Wallet

- Contains the full node of the blockchain, all transactions are stored
- Helps provide redundancy to the ecosystem
- Requires a lot of space and hardware resources down the line

Hosted Wallet

TYPES OF WALLETS

Light Wallet

Full Wallet

Hosted Wallet

- Wallet with the node hosted by a third-party
- Requires user to trust the host of the node to play nice
- Insecure, and risky at times, but can often be the only feasible solution

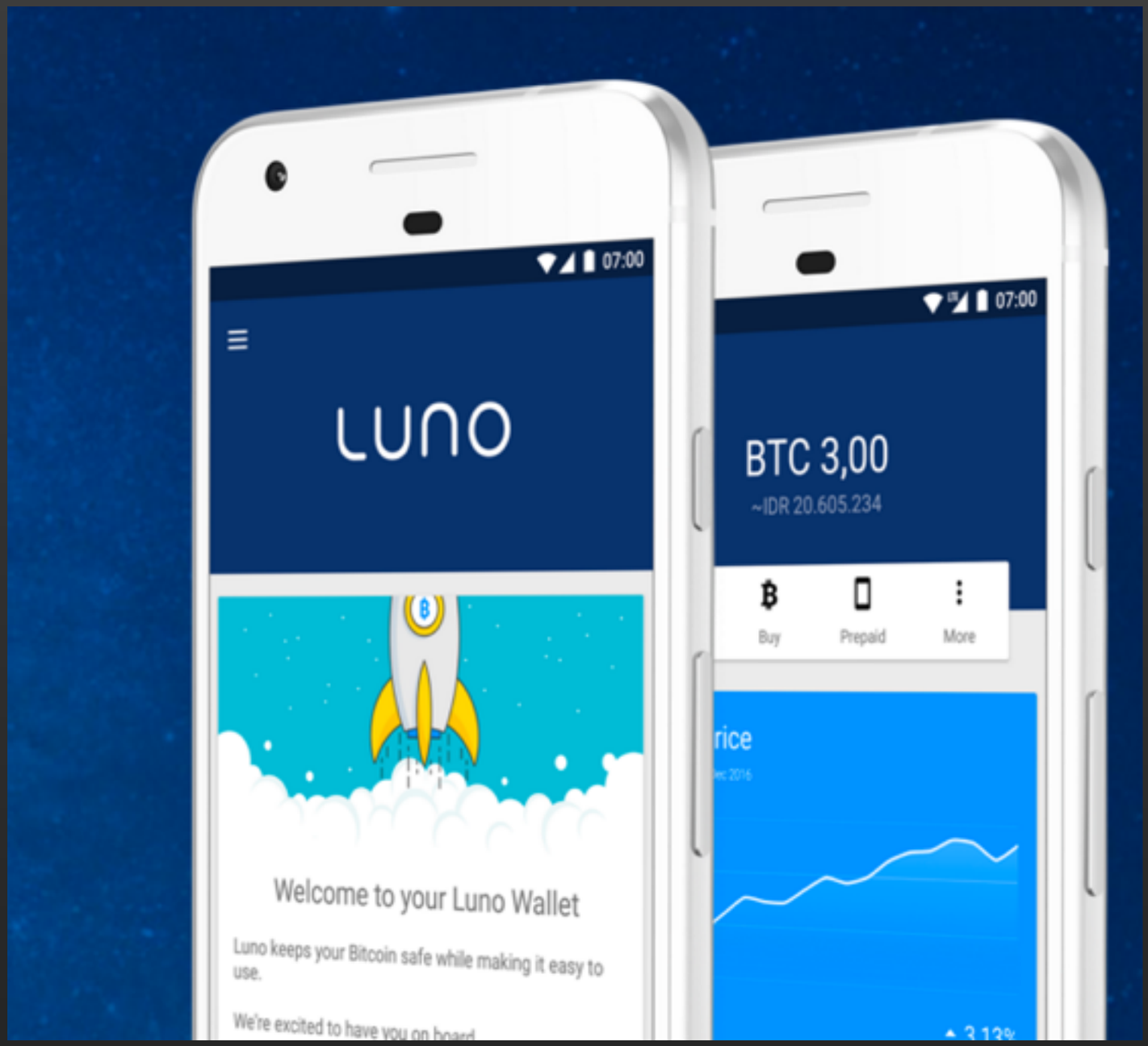
CRYPTO-EXCHANGES

- Only way to gain participation into a blockchain is through crypto-exchanges

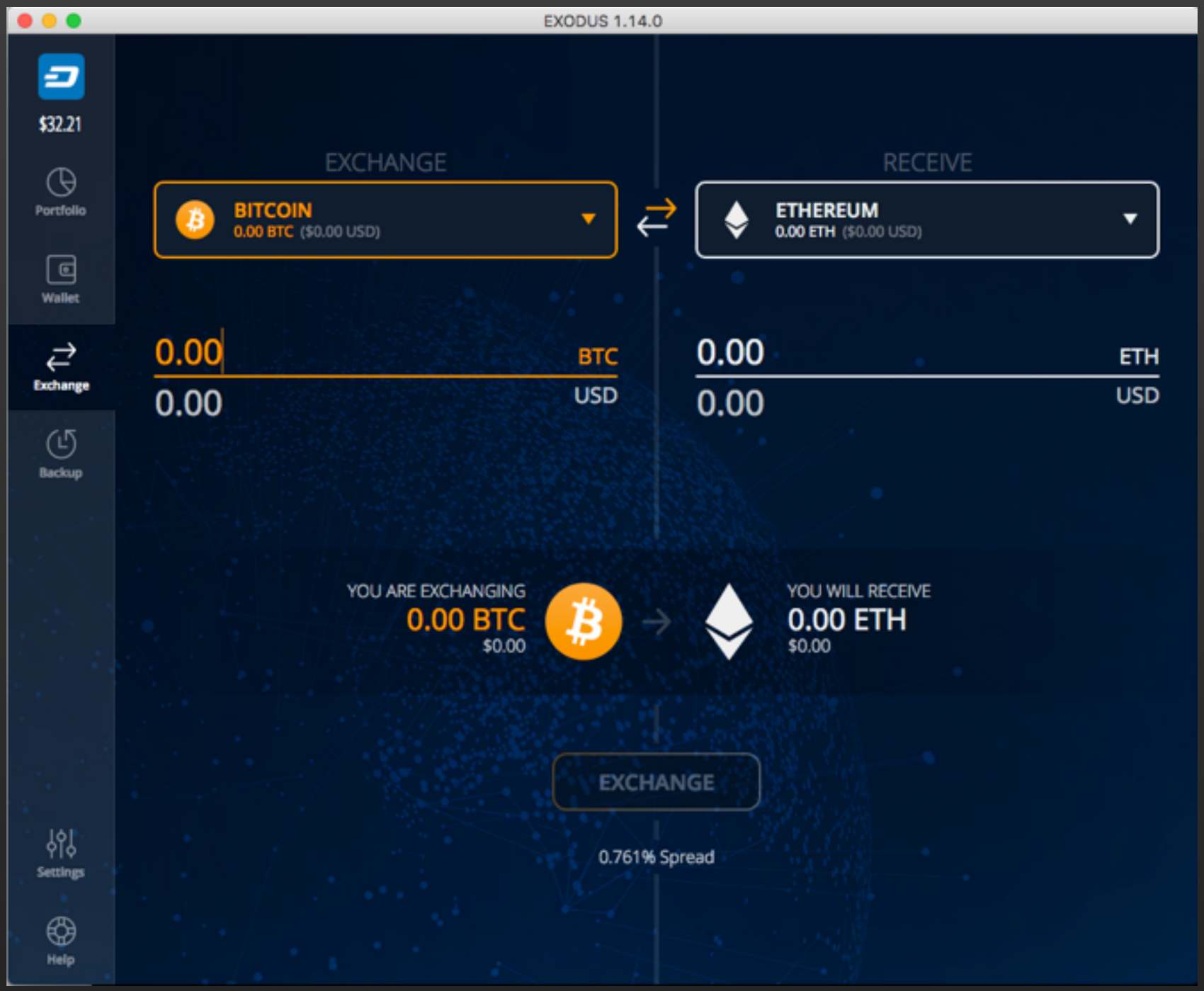
- Steps:
  - Login to an exchange
  - Select amount of cryptocurrency to purchase
  - Pay exchange via credit card/debit card/real money
  - Trust exchange to send you cryptocurrency
  - …
  - Profit?

- Crypto-exchanges are notorious for being vulnerable to hacks
  - Functions the same as banks/stock exchanges
  - Run without regulations

CRYPTO-EXCHANGES



CRYPTO-EXCHANGES



KEY MANAGEMENT

- Most crypto-currencies operate using private/public key pairs
  - Private key —> Public key —> Address

- Since there are no central authorities, if you lose your private key, your “money” is gone
  - Your money, your responsibility

- Ways to manage keys:
  - Memorise/store the private key with backups
  - Entrust the private key to a trustee
  - Use a hardware wallet
  - Use a deterministic wallet
    - Type 1 - Deterministic Wallet
    - Type 2 - Hierarchical Deterministic (HD) Wallet
  - **All of the above**

POPULAR KEY MANAGERS/WALLETS

ELECTRUM

POPULAR KEY MANAGERS/WALLETS

POPULAR KEY MANAGERS/WALLETS

SEGMENT 4 - BREAK TIME

QUESTION & ANSWER SESSION

15 mins

THANK YOU

LEARN MORE ABOUT

neuroware

http://neuroware.io