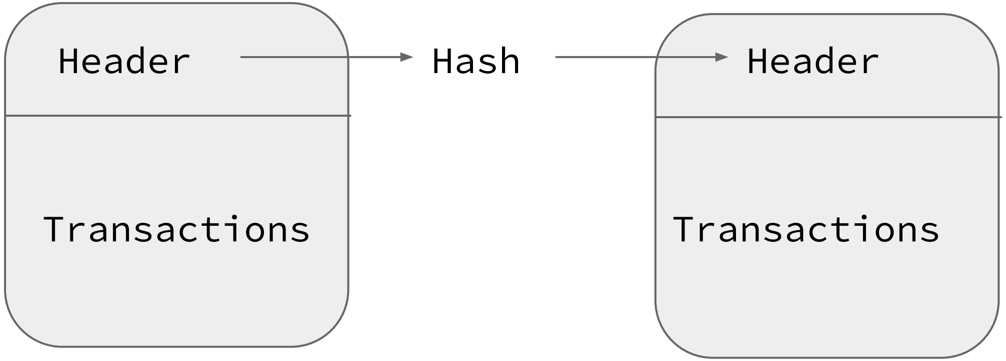
Blockchain

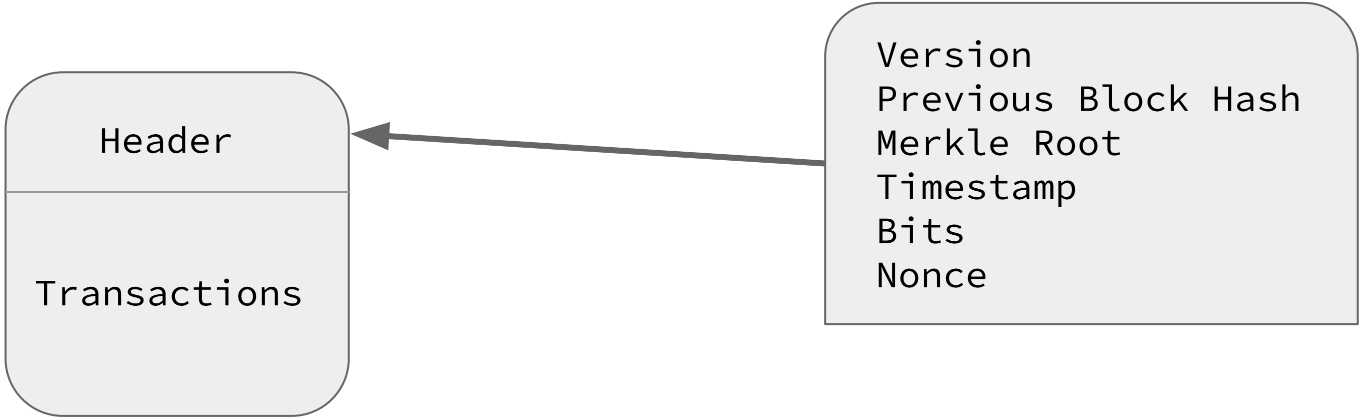
Introduction

WHAT IS A BLOCKCHAIN?

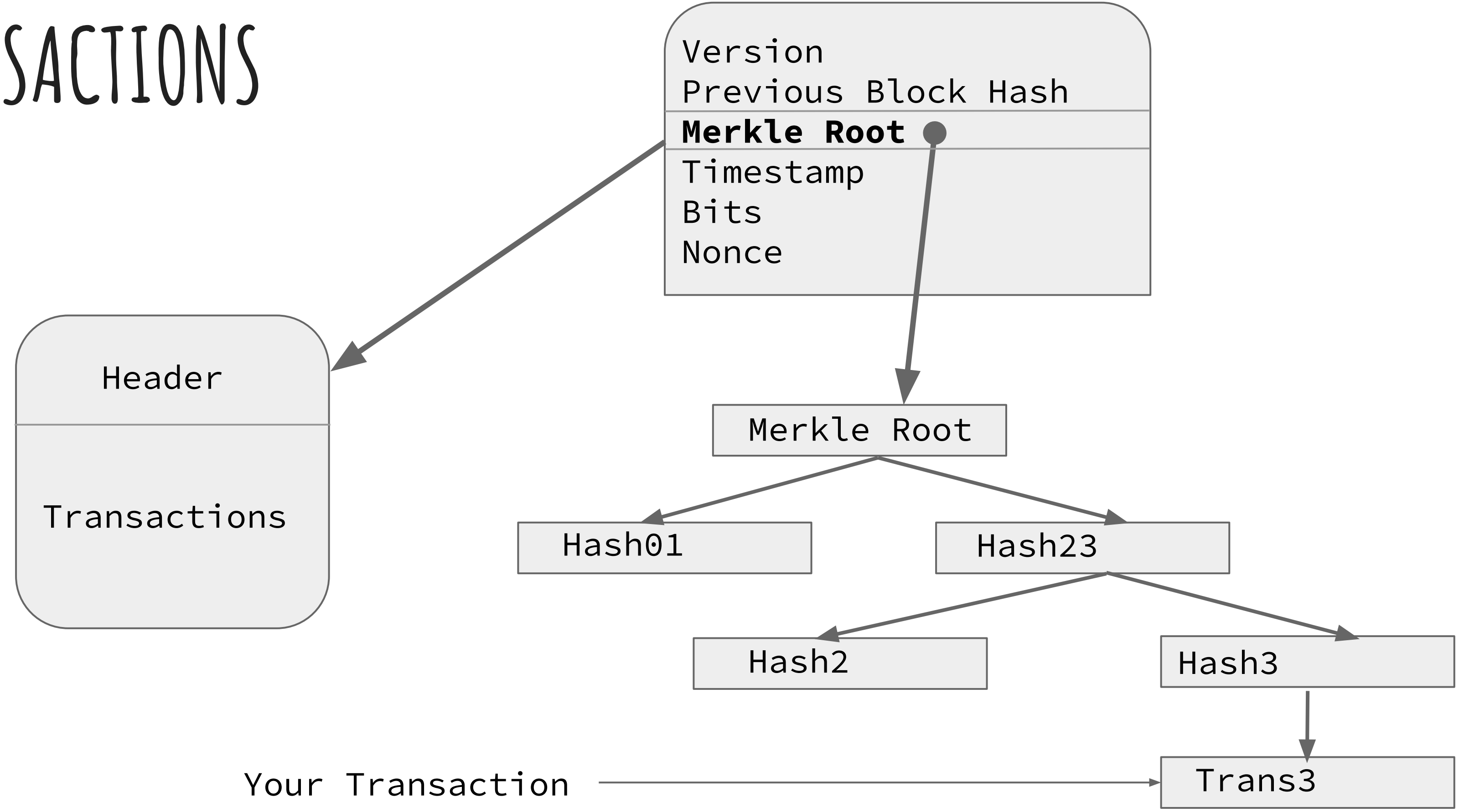
- A blockchain is made up of blocks



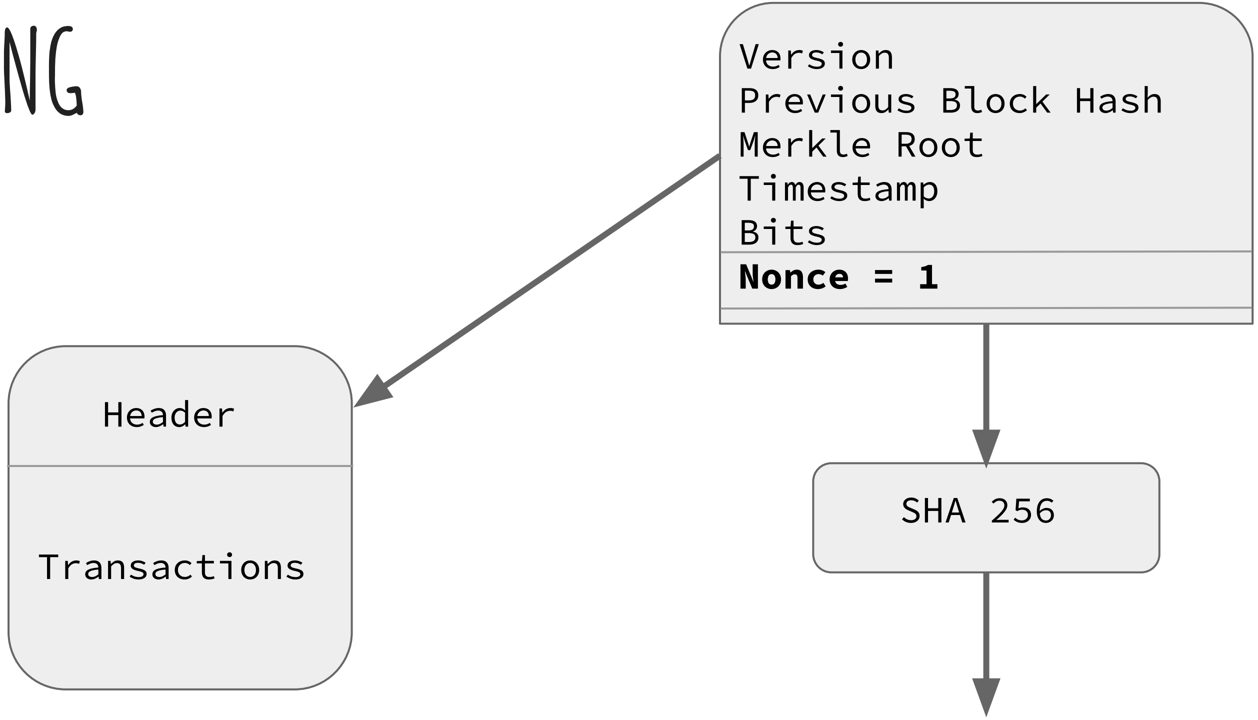
BLOCK HEADERS



TRANSACTIONS



MINING



Version
Previous Block Hash
Merkle Root
Timestamp
Bits

Nonce = 1

Header

Transactions

SHA 256

0000000w359gfy8wneab143b0d668b9407accc32429f5336bf42e5a1d303c8

- This Proof of Work would be rejected
- It doesn’t hash to a small enough number
- Increase the nonce and try again!

Proof of Work

- Dwork and Naor (1992)
- a means of transmitting a value signal over the internet
- Initially a spam deterrence mechanism
- Solve a cryptographic puzzle to send a message
- A *strong economic signal* could be sent over a digital channel
- Eliminate the need to rely on trust

BITCOIN

- The first blockchain (live 2009)
- Use proof of work for a digital payment system
- A decentralized, peer-to-peer network
- Mining power has consolidated
- Bitcoin uses more electricity than the country of Switzerland

Ethereum

- Second most popular public blockchain
- Functionality for smart contracts and decentralized applications
- People were trying to build these things on top of Bitcoin
- There were 2 main changes made by Buterin:
    - 1. Turing completeness: He allowed Ethereum to run loops which was something that Bitcoin did not allow for security reasons.
    - 2. Account Model: Bitcoin used a UTXO model. Transactions had to be fully spent and this prevented incremental changes to states which was something that was needed by developers. Contracts have to be re-created constantly in Bitcoin.

Ethereum virtual machine

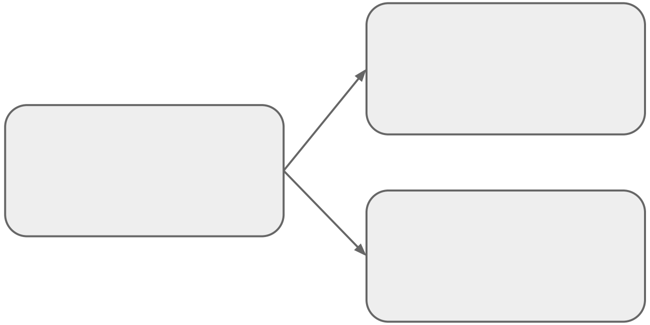
- Is a harvard stack machine
- Programs in EVM run on every client
- Called the world computer
- Storage and execution length is limited by the GAS charge of every instruction
- Ex. Store operations are more expensive than add or subtract
- Has 6 address spaces: Code, Memory, Storage, Arguments, Return Arguments, Execution Stack
- Every transaction contains values for the GAS LIMIT and GAS PRICE.
- This prevents infinite loops.

Criticisms of bitcoin and ethereum

- Both are Proof-of-work (POW) based blockchains
- 2 main criticisms:
  - 1. Energy use: Bitcoin uses more electricity than Switzerland. Ethereum = Bolivia.
  - 2. Processing speed: Bitcoin can process about 7 tps, Ethereum does better at 20 tps thanks to the GHOST protocol. For mainstream adoption both would have to scale well beyond 1000 tps.
- Alternative consensus mechanisms exist that solve both issues like Proof-of-stake (POS) and Federated Byzantine Agreement (FBA).
- With other consensus models, both security and decentralization potentially suffer.

FORKING

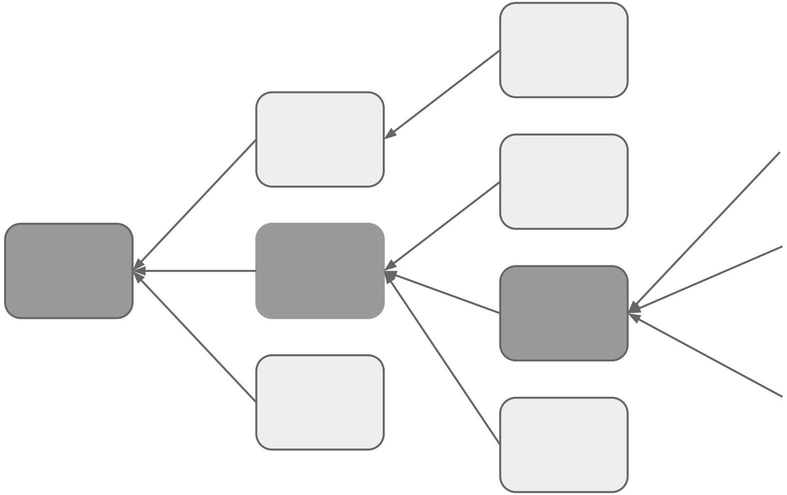
- Proof-of-Work blockchains are slow for a reason
- If 2 miners both solve a cryptographic puzzle in the same time frame - blocks have to be broadcast through a huge peer-to-peer network - then the blockchain will fork



- Mining difficulty is set to limit forking
- Forking still happens
- The network is kept slow on purpose
- Speeding up the network would cause too much forking and there would soon be many Bitcoin blockchains because nodes would not have time to agree on one particular chain

DAG BASED BLOCKCHAINS

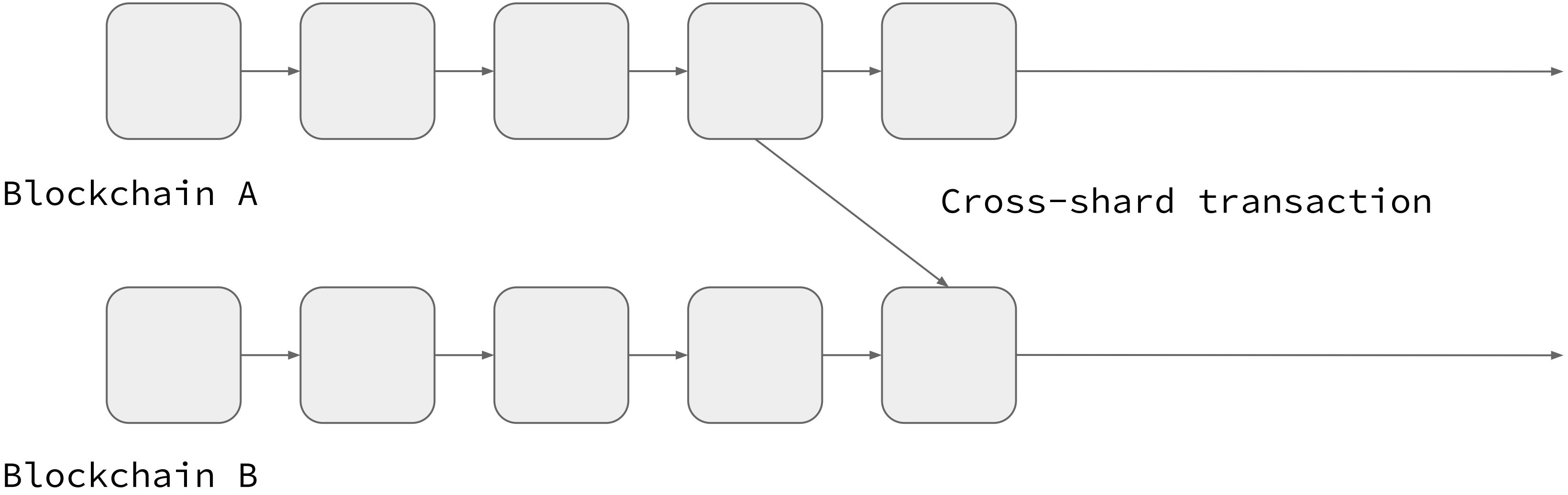
- With excessive forking, a blockchain is just a directed acyclic graph
- Design graph based algorithms to help achieve consensus



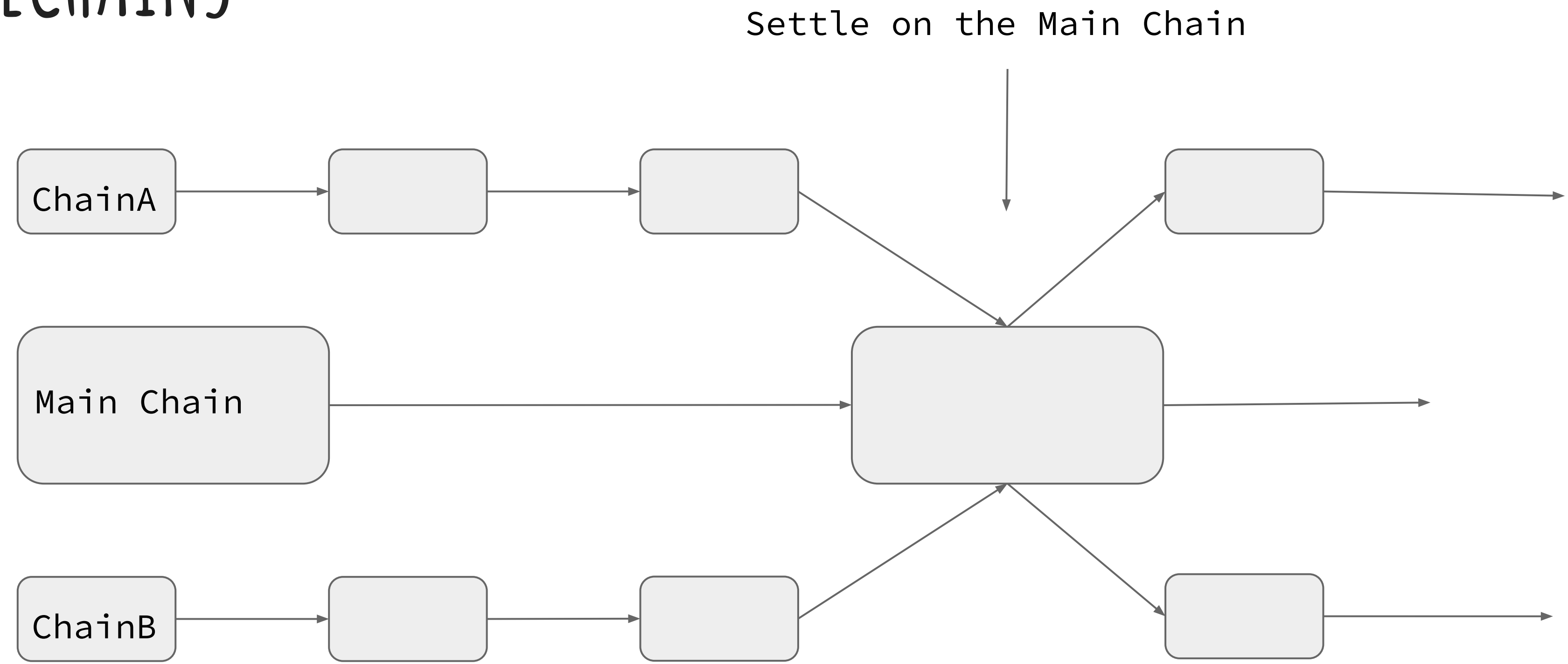
- The darker chain would be the agreed upon chain because it is referenced by the most blocks
- This is a simplified example
- Many ideas exist with DAG based blockchains
- Some protocols encourage forking others actually prevent it

CONCURRENT BLOCKCHAINS

- Allow blockchains to run concurrently
- This can be done with Sharding which is what Ethereum is attempting to implement



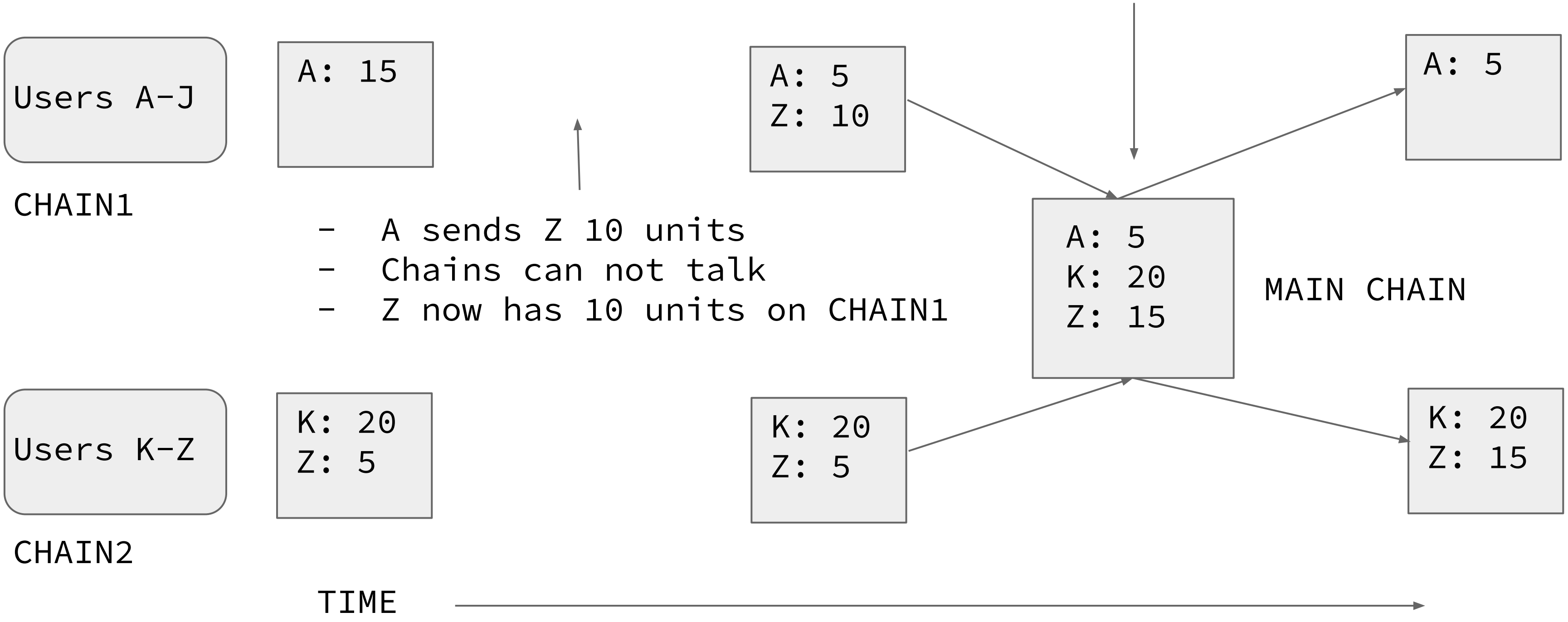
SIDECHAINS



Implementing sidechains

- Settle on Main chain
- Just add tokens for each account from each side chain then fork again
- Simple by design
- Much more complicated in real life
- **Critical section**

NAIVE SOLUTION



- A sends Z 10 units
- Chains can not talk
- Z now has 10 units on CHAIN1

Concurrency in blockchains

- I did a final project in CSC464: Concurrency, on this topic
- It’s now part of my thesis work for my Master’s degree
- I coded a sidechain implementation in GO
- I also wrote a report mostly talking about different DAG-based blockchain implementations
- Ultimately, I would like to code a working public p2p blockchain network (I would probably need help - big job)
- My favorite courses have been CSC464: Concurrency, CSC466: Peer to Peer Networking, and CSC462: Distributed Systems (462 is being offered this summer!)
- If you’re interested in any of this, don’t hesitate to contact me   jonathan.d.healy@gmail.com