Launch Pad Tech Talks

•••

Talk about cool stuff you know!

What is this?

- Speak for 2-10 minutes after standup
- **Anyone** can do a tech talk about **anything**
- Sign up by slacking Bruno/Sherry or filling out the form

Why are we doing this?

- Safe and fun environment to practice public speaking
- Share knowledge between people and teams
- Everyone knows something interesting that we don’t know

Signup Form

goo.gl/forms/YZmju10ehRjxNJkp2

(I'll post on Slack too)

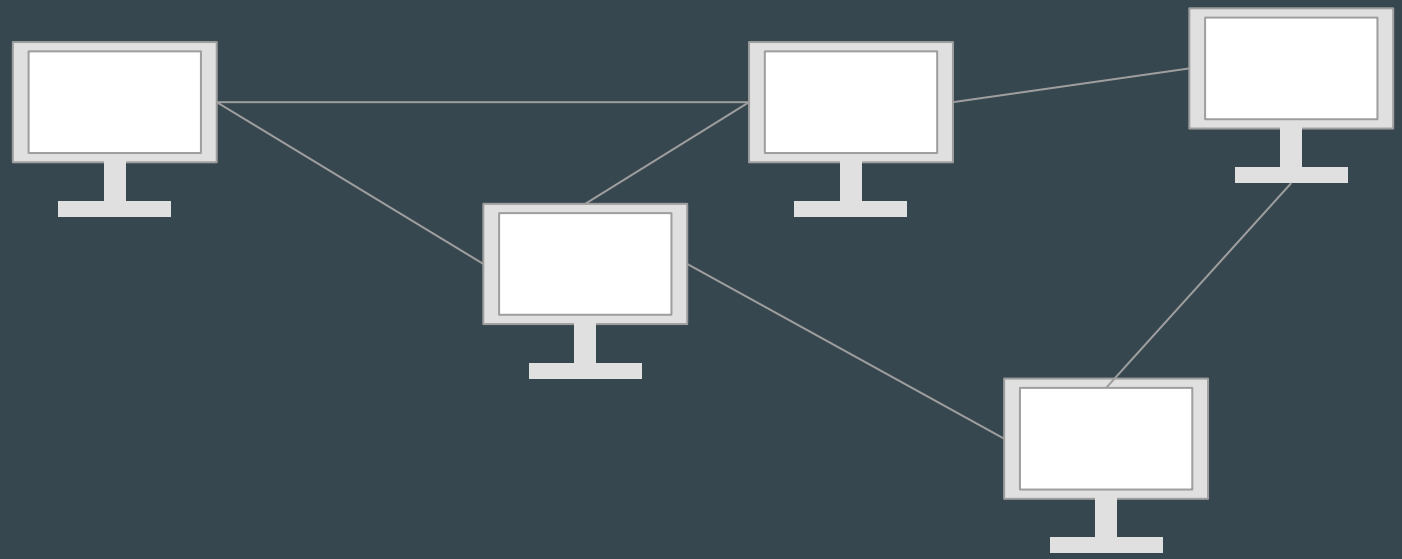
Blockchain

•••

A Technical Introduction

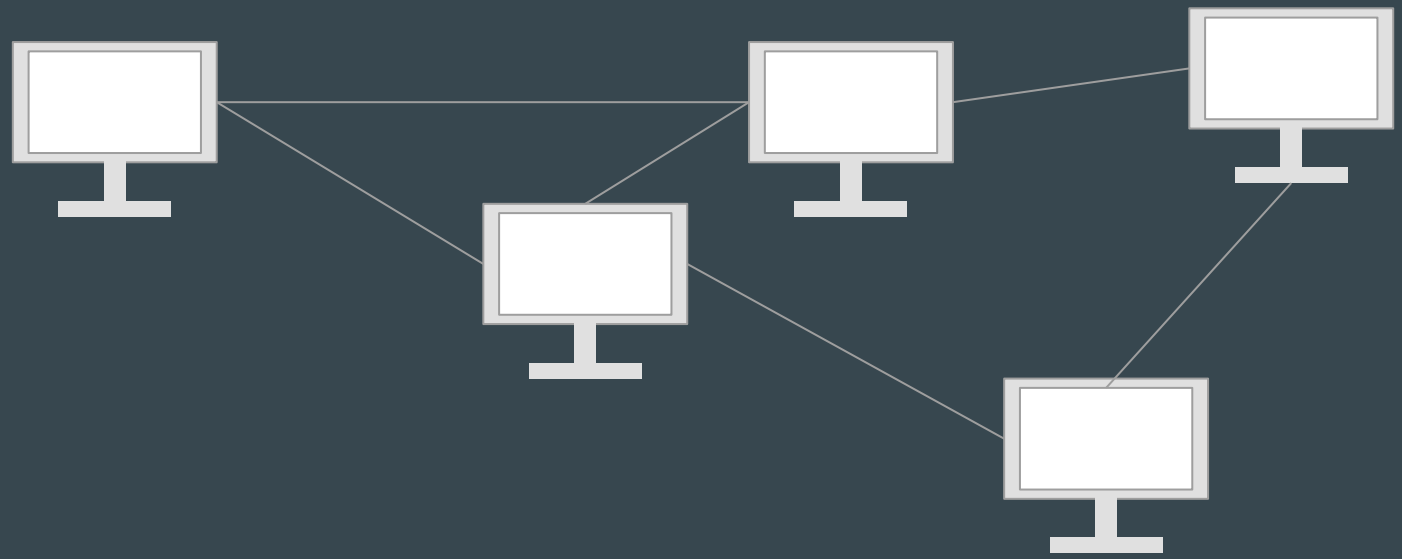
[1] Physical Layer

An ad-hoc network of computers sending messages to each other



[2] State Layer

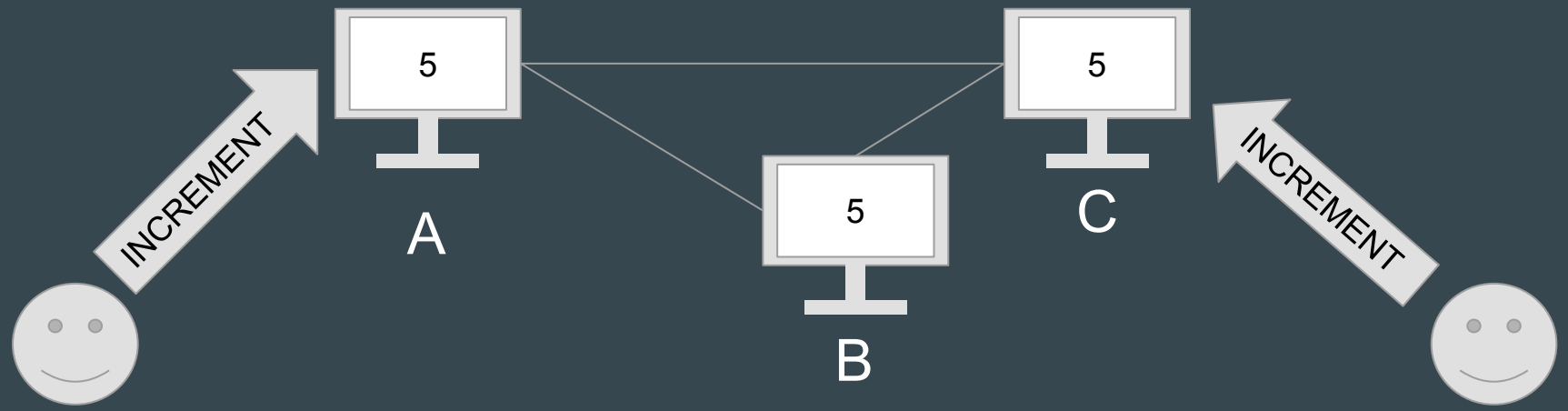
This network is working to maintain a shared state



[2] State Layer

Example: Maintaining the state of a counter

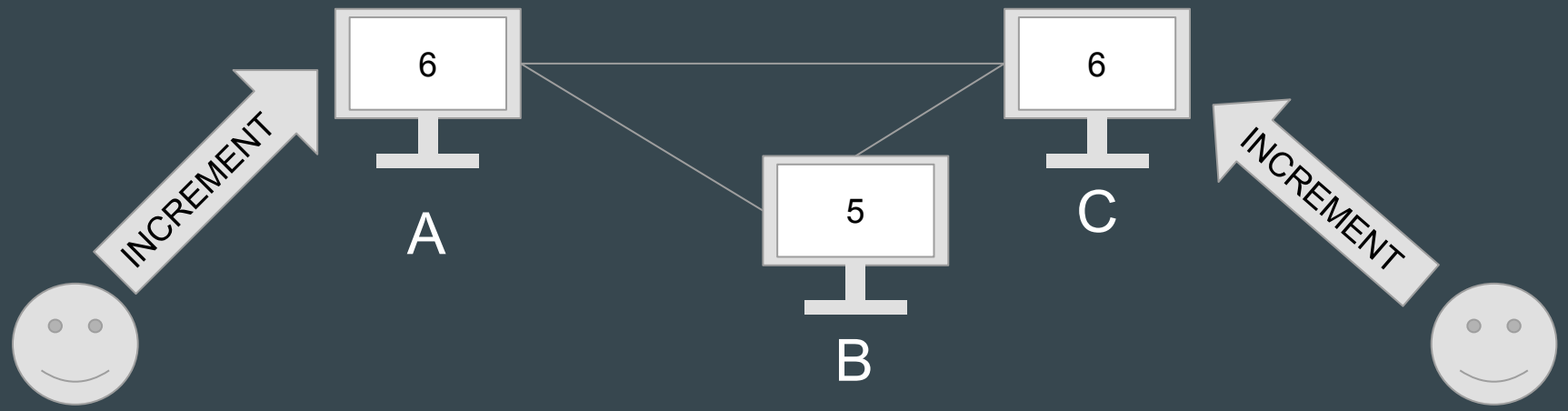
State: 5



[2] State Layer

Example: Maintaining the state of a counter

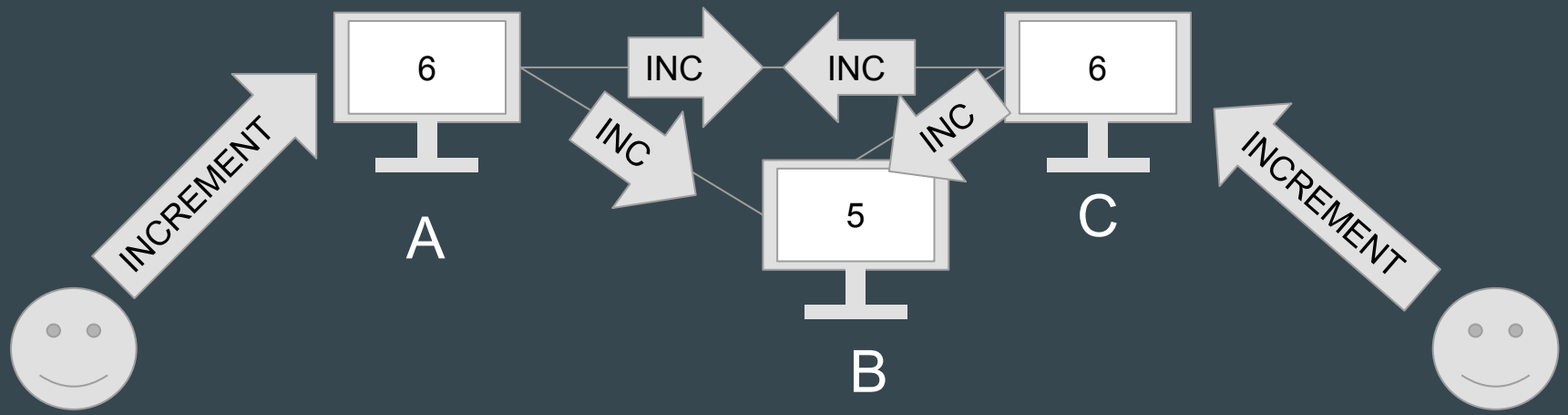
State: 5/6/7?



[2] State Layer

Example: Maintaining the state of a counter

State: 5/6/7?



[2] State Layer

Example: Maintaining the state of a counter

Resending transactions works for addition, because it's **associative**

$$(+) \quad : \quad a + (b + c) = (a + b) + c$$

[2] State Layer

Example: Maintaining the state of a counter

Resending transactions works for addition, but doesn't work in general!

$(+) \quad : \quad a + (b + c) = (a + b) + c$ ✓

$(+, \times) : \quad 2 \times (3 + 4) \neq (2 \times 3) + 4$ ✗

$14 \neq 10$

[2] State Layer

Example: Maintaining the state of a counter

What if we send ALL the transactions every time.
The state is what you get after applying all the transactions IN ORDER

1. ADD 1
2. MUL 2
3. ADD 5
4. MUL 2
5. ADD 1

...

[2] State Layer

Example: Maintaining the state of a counter

What if we send ALL the transactions every time.
The state is what you get after applying all the transactions IN ORDER

1. ADD 1
2. MUL 2
3. ADD 5
4. MUL 3
5. ADD 1

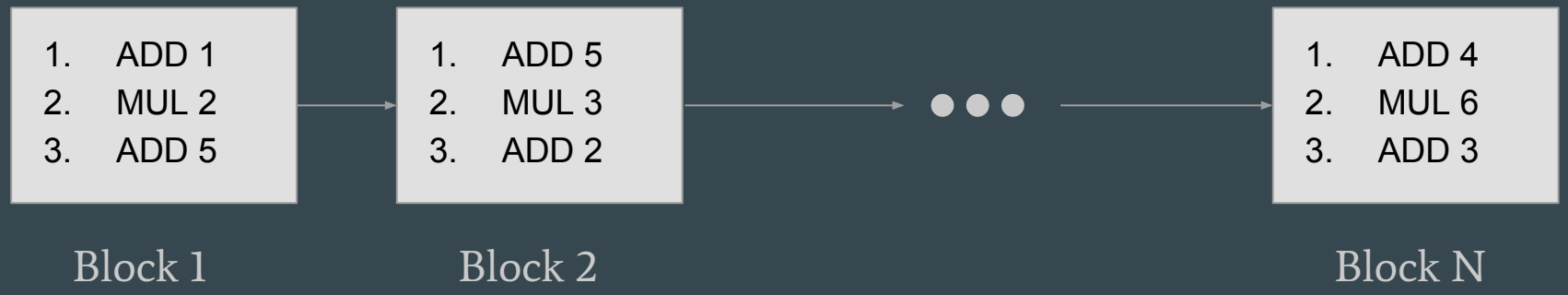
...

((((0 + 1) x 2) + 5) x 3) + 1) ...

[2] State Layer

Example: Maintaining the state of a counter

In reality there are a lot of transactions, so we split them into blocks.
Each block has a reference to the last block, so ordering is preserved.



[2] State Layer

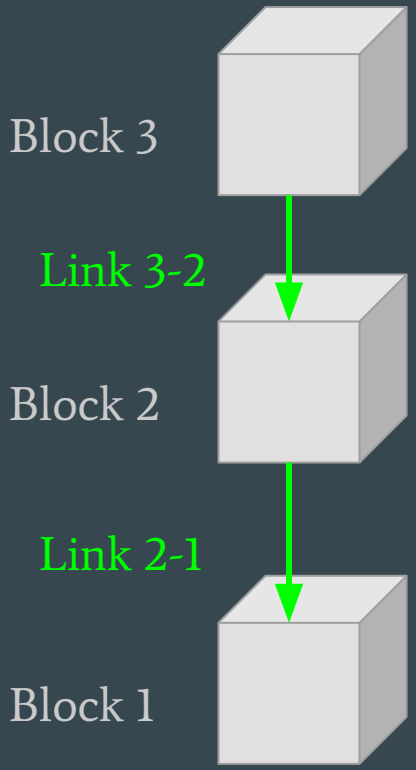
Recap

- Transactions are stored in groups called blocks
- The state is what you get by executing every transaction in order

[3] Trust Layer

Why do we trust things stored in the blockchain?

Each block contains a fingerprint of the last block

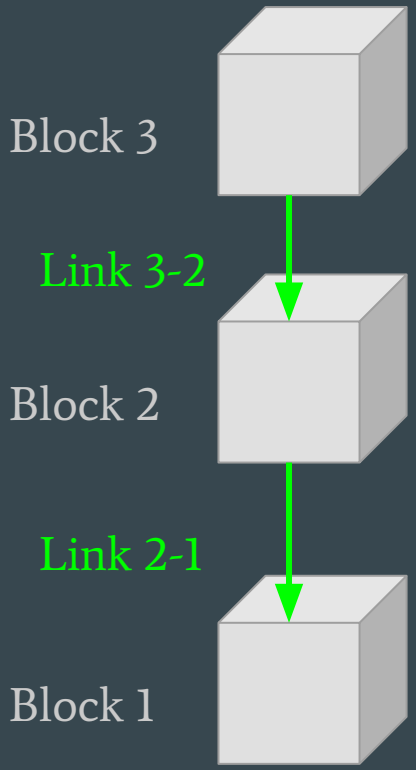


[3] Trust Layer

Why do we trust things stored in the blockchain?

Each block contains a fingerprint of the last block

If any previous block is changed, the fingerprint won't match

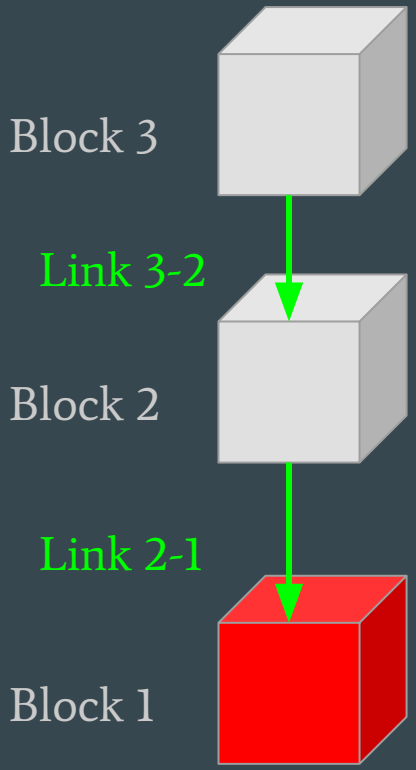


[3] Trust Layer

Why do we trust things stored in the blockchain?

Each block contains a fingerprint of the last block

If any previous block is changed, the fingerprint won't match

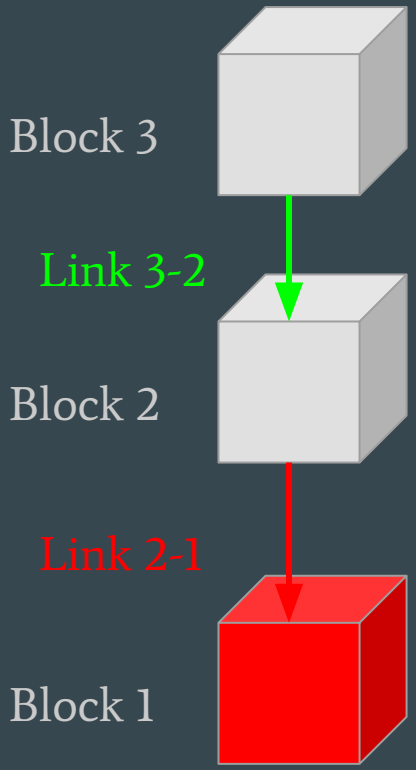


[3] Trust Layer

Why do we trust things stored in the blockchain?

Each block contains a fingerprint of the last block

If any previous block is changed, the fingerprint won't match

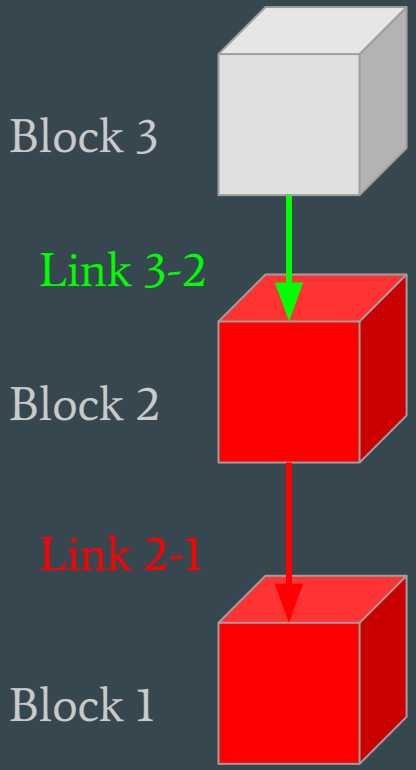


[3] Trust Layer

Why do we trust things stored in the blockchain?

Each block contains a fingerprint of the last block

If any previous block is changed, the fingerprint won't match

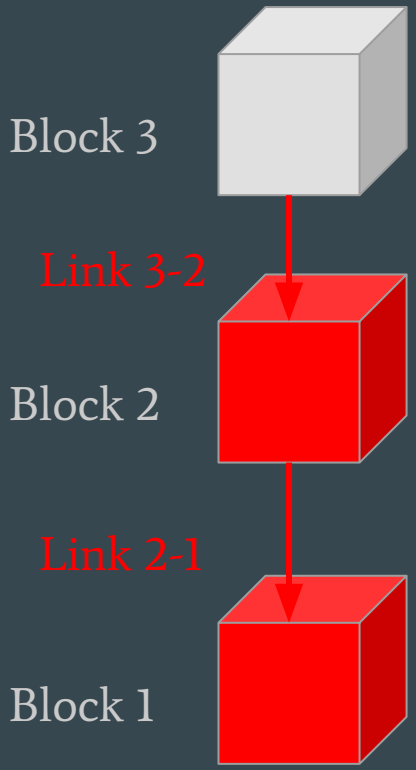


[3] Trust Layer

Why do we trust things stored in the blockchain?

Each block contains a fingerprint of the last block

If any previous block is changed, the fingerprint won't match



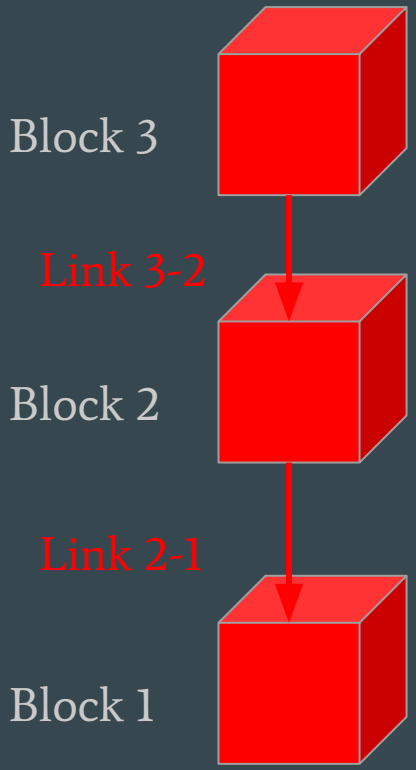
[3] Trust Layer

Why do we trust things stored in the blockchain?

Each block contains a fingerprint of the last block

If any previous block is changed, the fingerprint won't match

This mismatch propagates up the chain, making the whole thing invalid!



[3] Trust Layer

Proof of Work

- Each block contains a “proof” that a large amount of computational work was expended in producing the block

[3] Trust Layer

Proof of Work

- Each block contains a “proof” that a large amount of computational work was expended in producing the block
- Nodes that produce valid blocks are rewarded

[3] Trust Layer

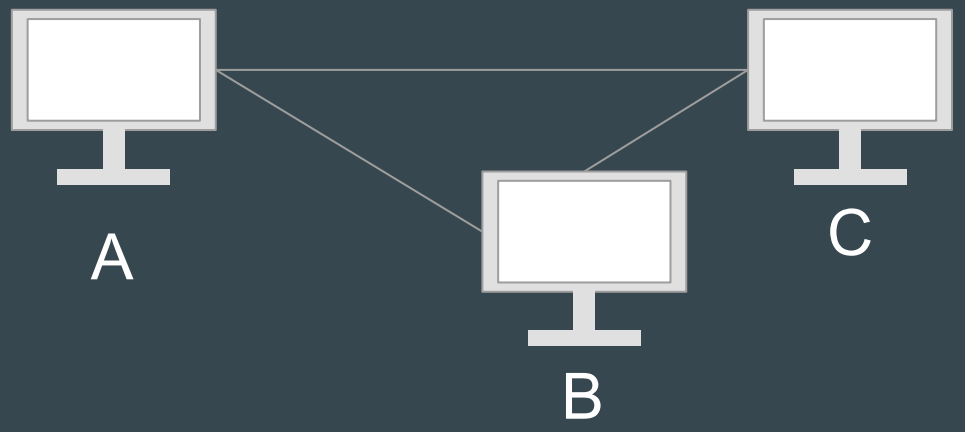
Proof of Work

- Each block contains a “proof” that a large amount of computational work was expended in producing the block
- Nodes that produce valid blocks are rewarded
- It is expensive and difficult to produce fake blocks

[4] Consensus Layer

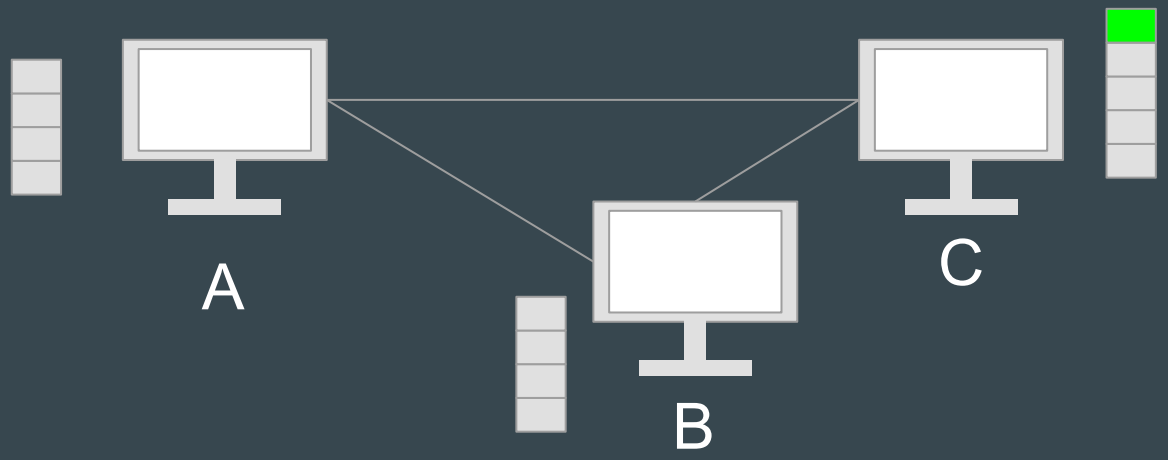
How does the network agree on the ordering of transactions?

The longest chain rule



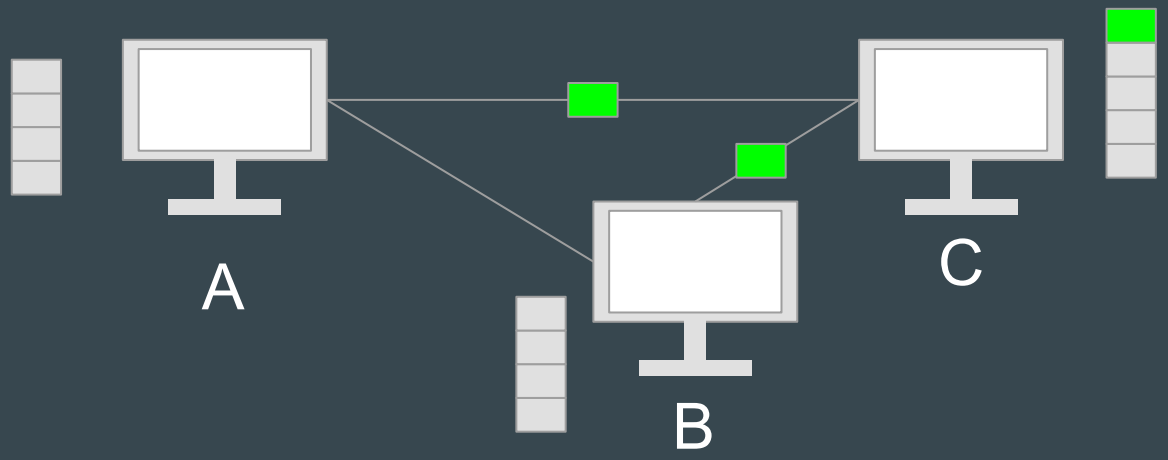
[4] Consensus Layer

How does the network agree on the ordering of transactions?



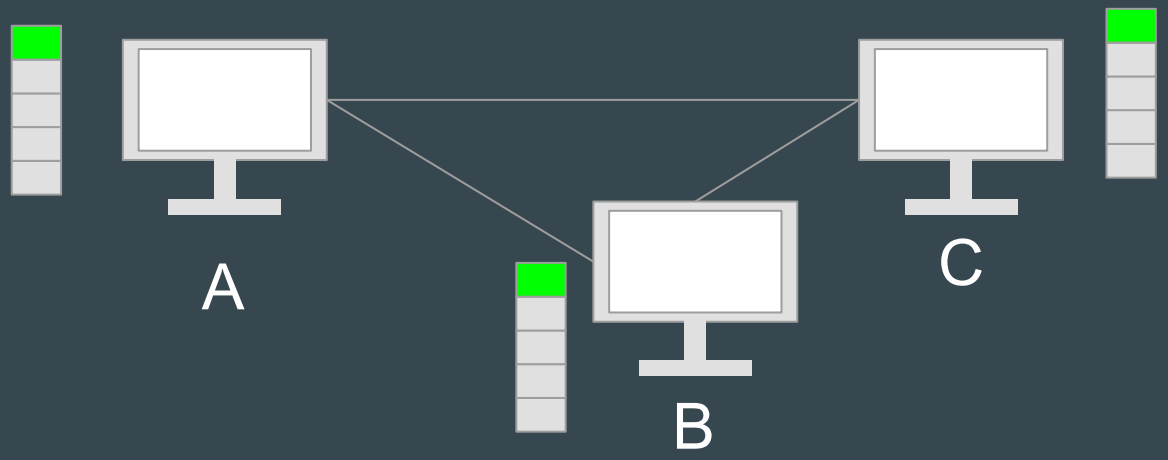
[4] Consensus Layer

How does the network agree on the ordering of transactions?



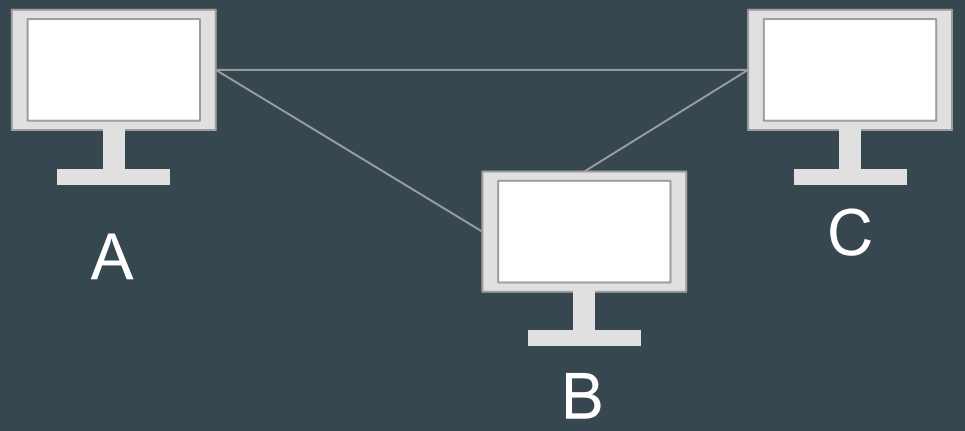
[4] Consensus Layer

How does the network agree on the ordering of transactions?



[5] Identity Layer

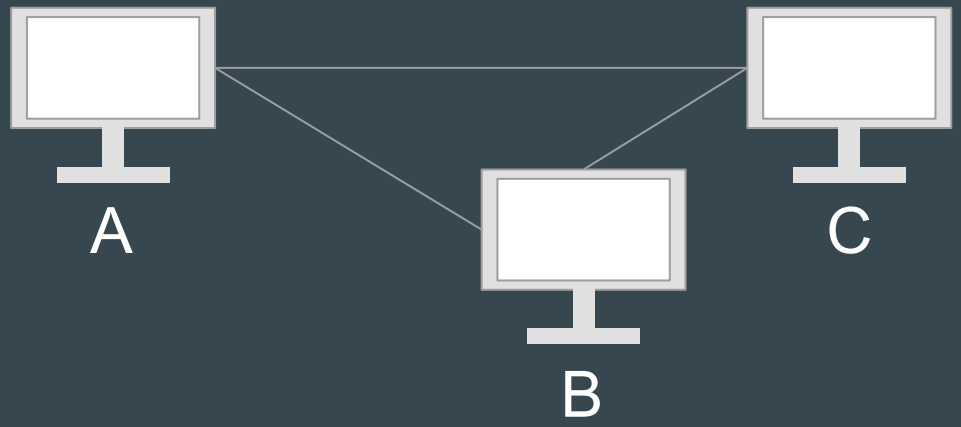
Who creates all these transactions?



[5] Identity Layer

Who creates all these transactions?

Users can create identities in the system at any time by generating an “address”.



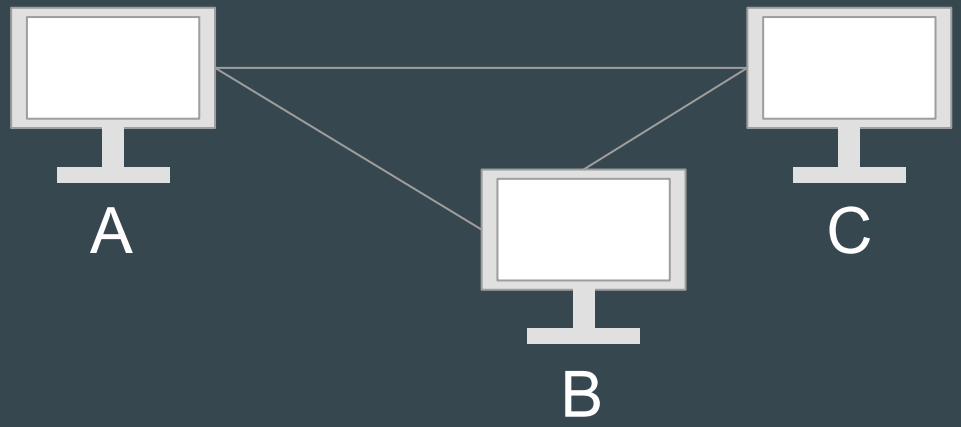
Address:
0x06012c8cf97bead5deae237070f9587f8e7a266d

[5] Identity Layer

Who creates all these transactions?

Every address has a corresponding secret key. If you know the secret key, you can prove you own the address.

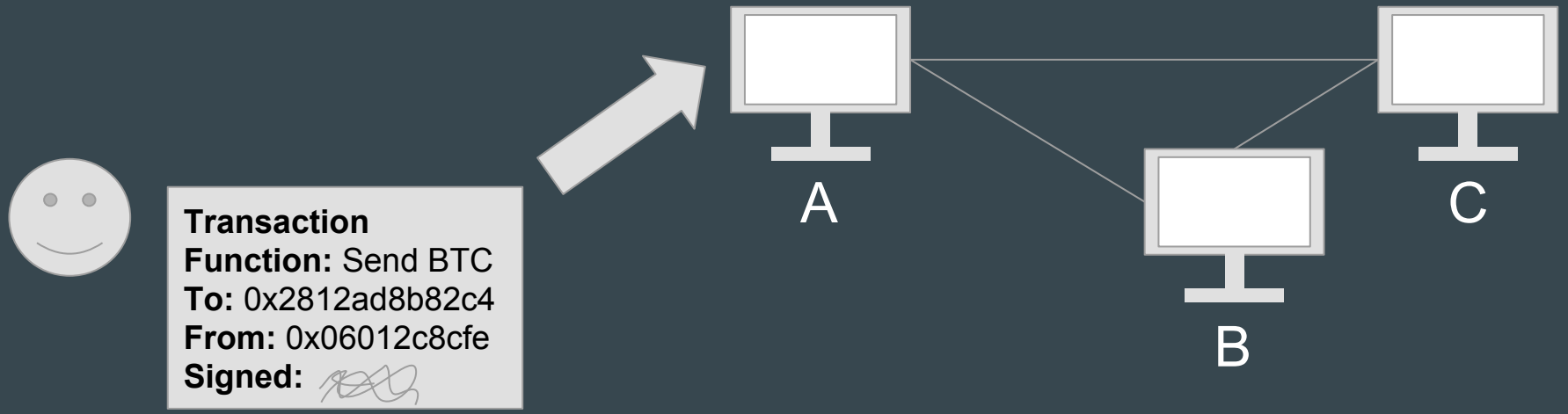
Secret:
password1234



[5] Identity Layer

Who creates all these transactions?

You prove you know the secret using a “digital signature”.



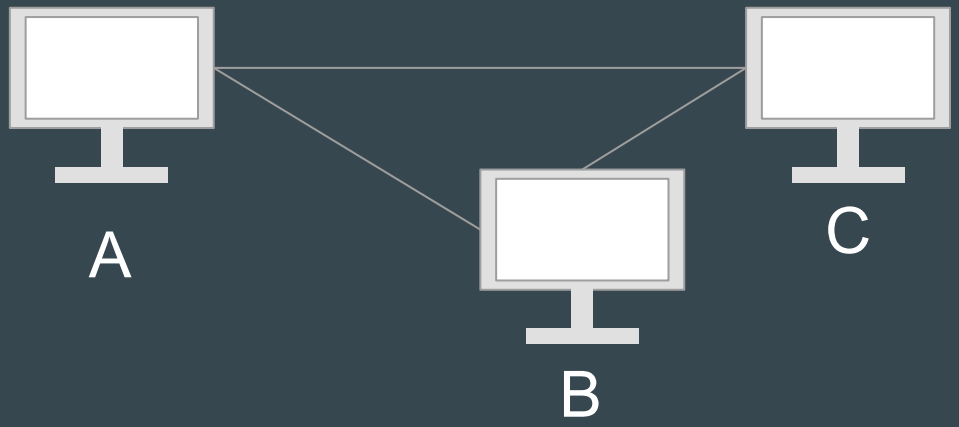
[5] Identity Layer

Recap

- Anyone can generate an identity at any time
- Each identity consists of:
  - an address (how others send you stuff)
  - a secret key (how you access your stuff)

[6] Application Layer

What can you build with it?



[6] Application Layer

What can you build with it?

Smart contracts

- Application state
- Set of functions that modify state

[6] Application Layer

What can you build with it?

Smart contracts

- Application state
- Set of functions that modify state

"Special Numbers" Smart Contract

State:

```
mapping (address -> int) specialNumbers
```

Functions:

```
// Set your special number. "caller" refers
// to the address of the caller
setNum(int i) {
    specialNumbers[caller] = i
}

// Get anyone's special number
getNum(address addr) {
  return specialNumbers[addr]
}
```

[6] Application Layer

What can you build with it?

Transaction
Function: setNum
Parameters:
- 42

To: “Special Numbers”
From: 0x06012c8cfe
Signed:

“Special Numbers” Smart Contract

State:
mapping (address -> int) specialNumbers

Functions:

```
// Set your special number. “caller” refers
// to the address of the caller
setNum(int i) {
    specialNumbers[caller] = i
}

// Get anyone’s special number
getNum(address addr) {
  return specialNumbers[addr]
}
```

Q&A