COM-402: Information Security and Privacy

Blockchain
3/3

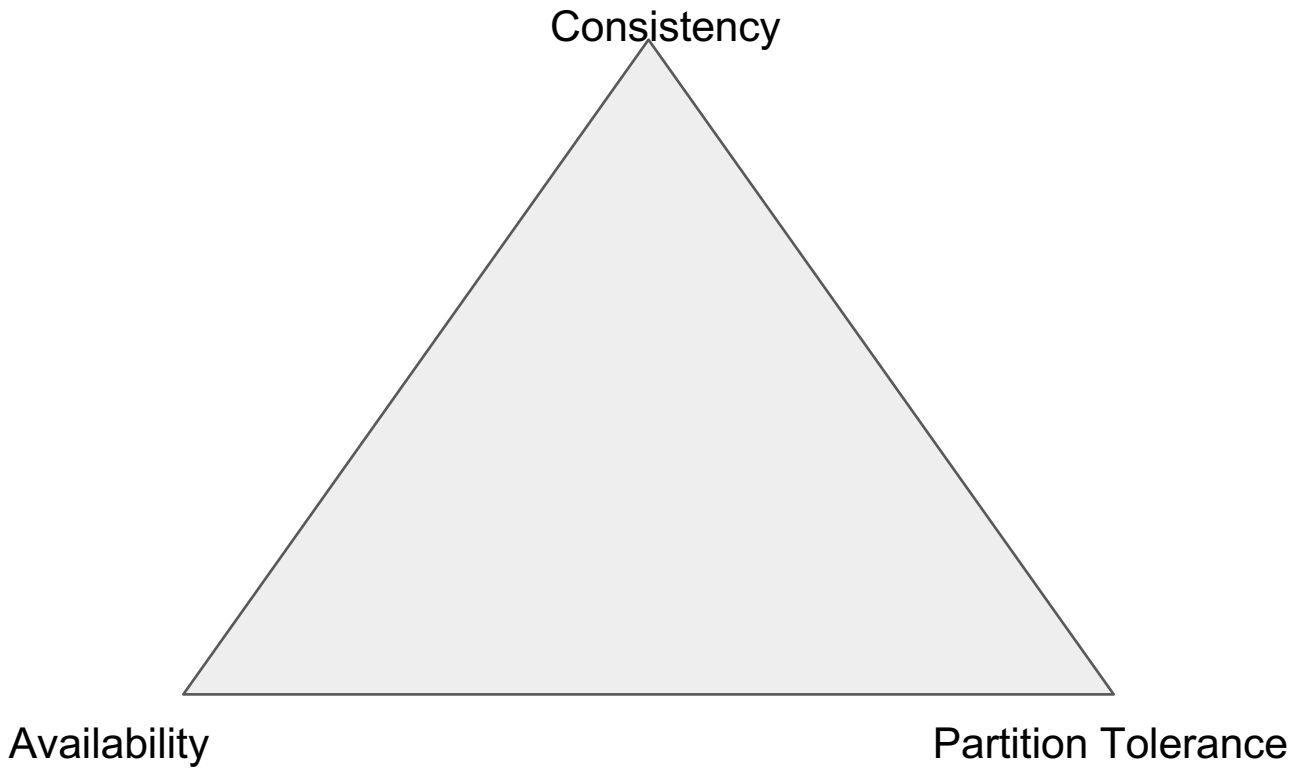
Slide credits: Bryan Ford, Lefteris Kokoris-Kogias, David Tse

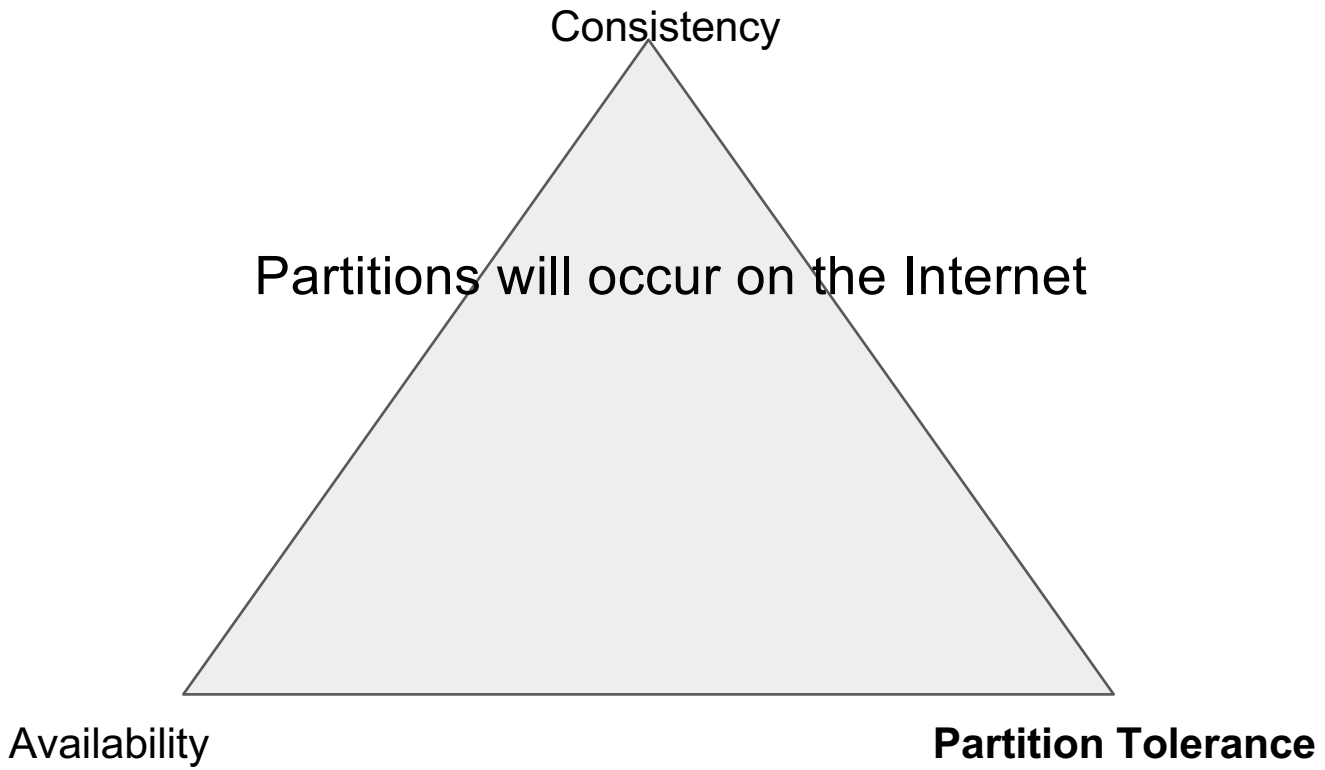
Some of the slides are also inspired by CS-522 POCS EPFL, Highly Available Transactions VLDB 2014, ECE-598 AM UIUC

Acknowledgments

Some of the slides are partly inspired by:

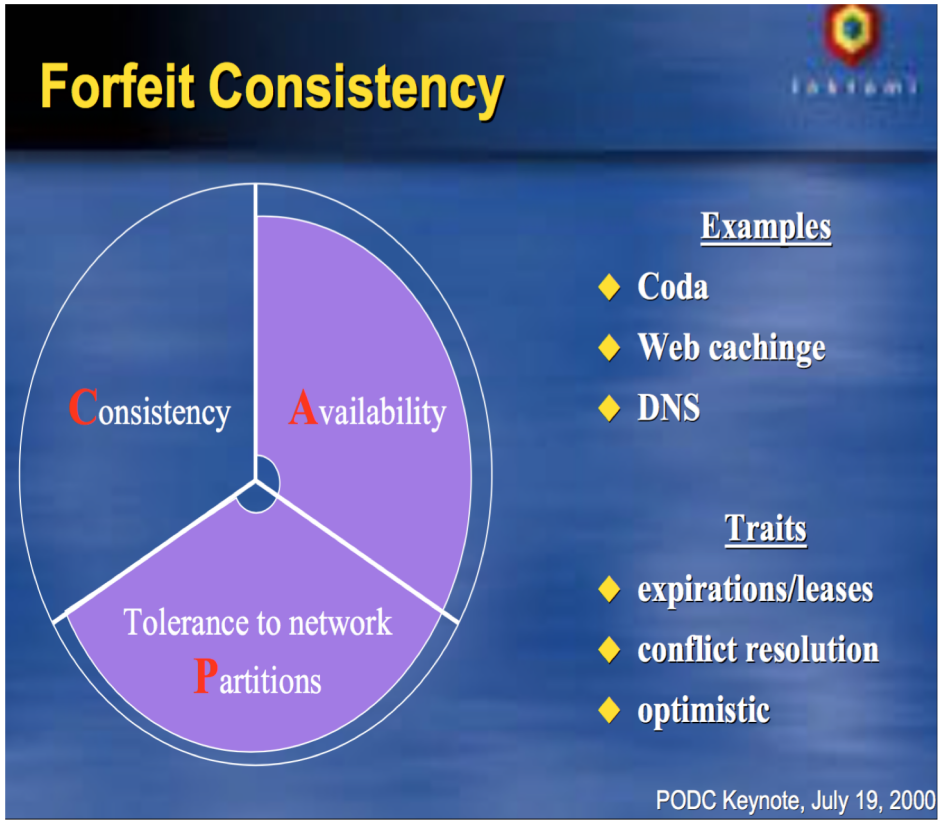
- CS-522 POCS EPFL
- Highly Available Transactions VLDB 2014
- ECE-598 AM UIUC


Consistency
Availability
Partition Tolerance


Consistency
Partitions will occur on the Internet
Availability
Partition Tolerance

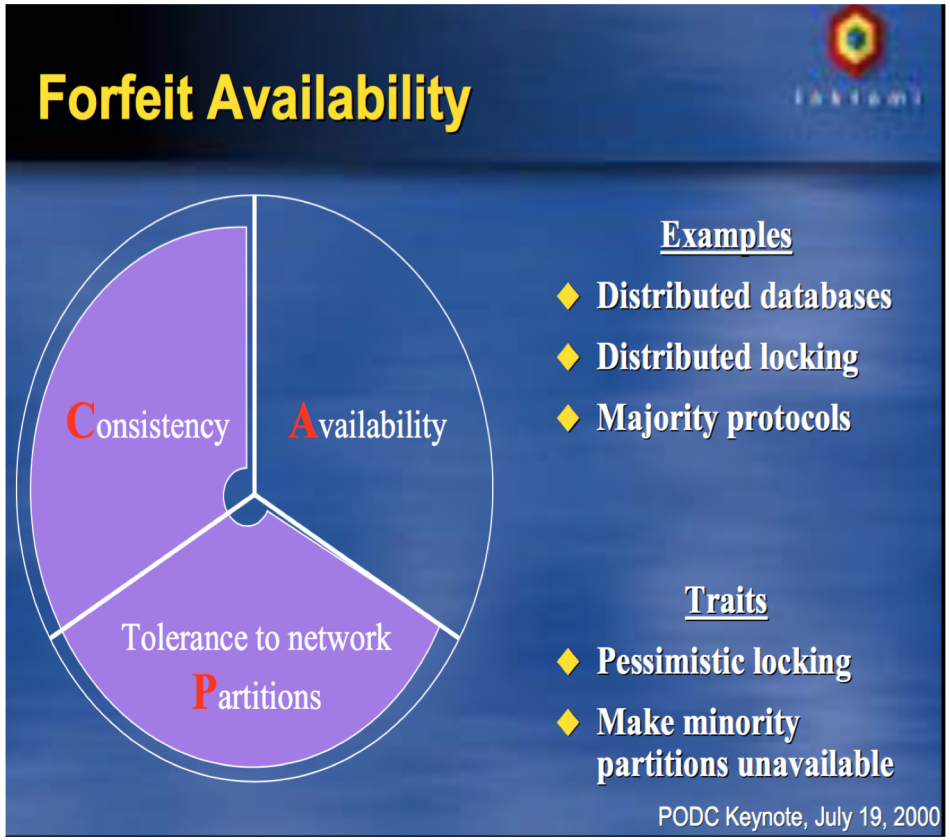
The AP Choice

- Strong Consistency is not possible
  - The system can reply with stale data
- Many applications do not care
  - DNS
  - Shopping carts
  - NoSQL Databases
- Benefits of weak consistency
  - Highly Available systems
  - Low-Latency
  - No Coordination



The CP Choise

- Strong Consistency
  - Safety first
  - System halts on partitions
- Needs Coordination
  - Consensus Protocols
- Benefits
  - Writes are atomic
  - Any data read are the freshest possible

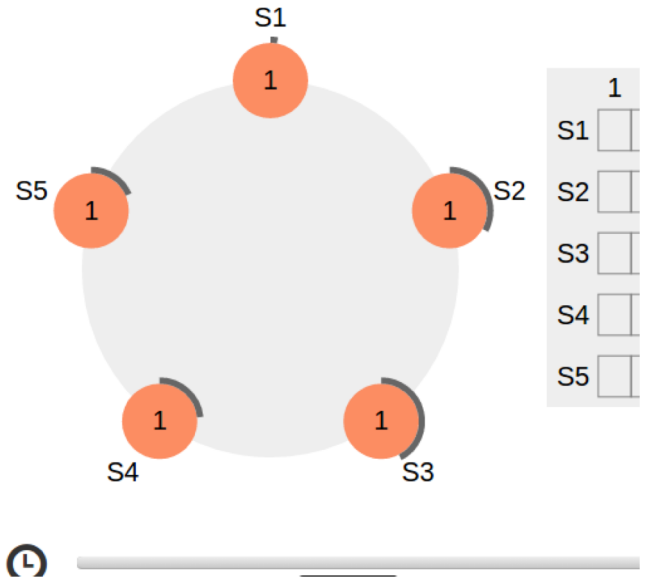


Consensus

- In the consensus problem, the processes propose values and have to agree on one among these values
- Properties
  - Validity: Any value decided is a value proposed
  - Agreement: No two correct processes decide differently
  - Termination: Every correct process eventually decides
  - Integrity: No process decides twice

Consensus in a Data Center

- Not a central focus of this course
  - CS-451
  - Paxos Made Simple L.Lamport
  - Raft ->In Search of an Understandable Consensus Algorithm D.Ongaro et al, ATC 14’



https://raft.github.io/

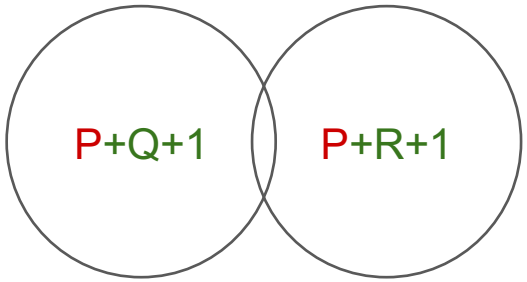
Byzantine Failures

- Assume some nodes and the network may be actively malicious
  - They might not reply at all (direct DoS attack)
  - They might be able to prevent honest nodes from communicating (indirect DoS attack)
  - They might send different messages to different nodes (equivocation
- Fundamentally need N=3f+1 for consensus in the general case
  - f out of N might not reply => Need to proceed with N-f or 2f+1
  - f out of the N-f might be malicious => Need majority
    - N-2f > f => N>3f or N=3f+1
- Can be relaxed to N=2f+1 under various stronger assumptions
  - Trusted hardware components to prevent equivocation
  - Assumptions that honest nodes can communicate within a finite time (synchronicity)

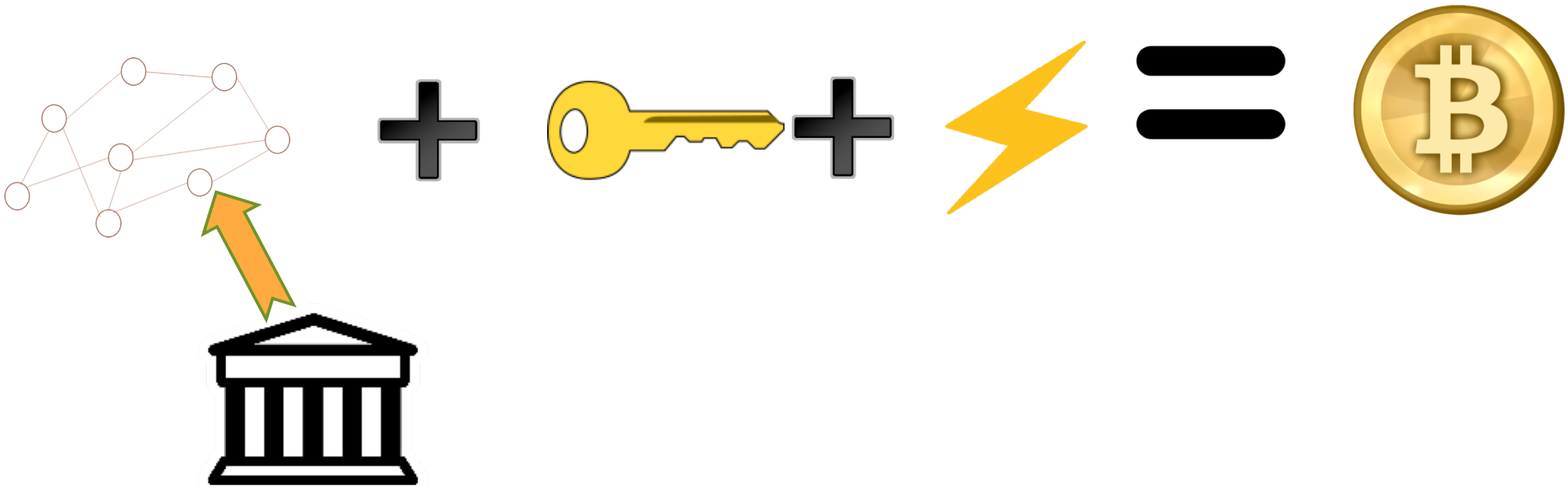
Impossibility results

No Byzantine Consensus f >= N/3,

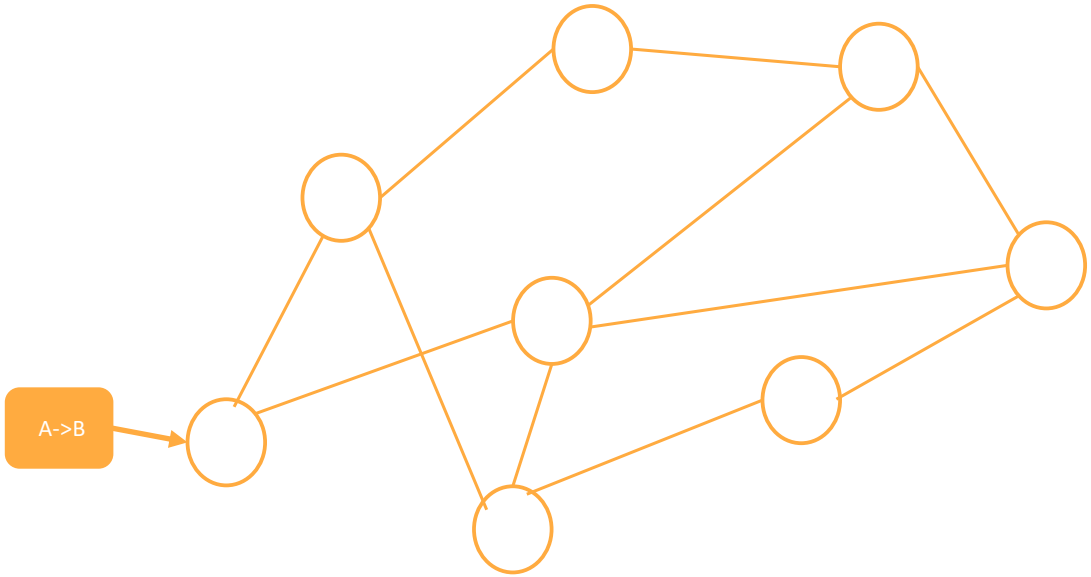
- Counter example: divide into 3 equal groups, P Q and R.
  - P is corrupted and contains the sender
  - Temporarily partition Q and R.
  - P behaves as though the Sender says “0” and interacts with Q.
  - P behaves as though the Sender says “1” and interacts with R.
- (P and Q) must behave the same as if R has crashed (pick “0”)
- (P and R) must behave the same as if Q had crashed (pick “1”)



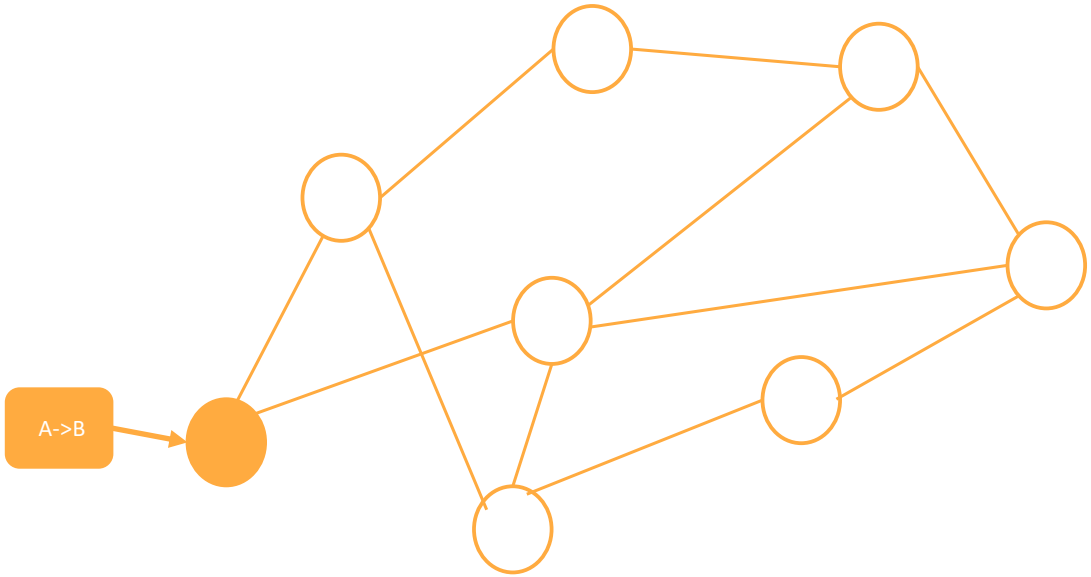
Bitcoin



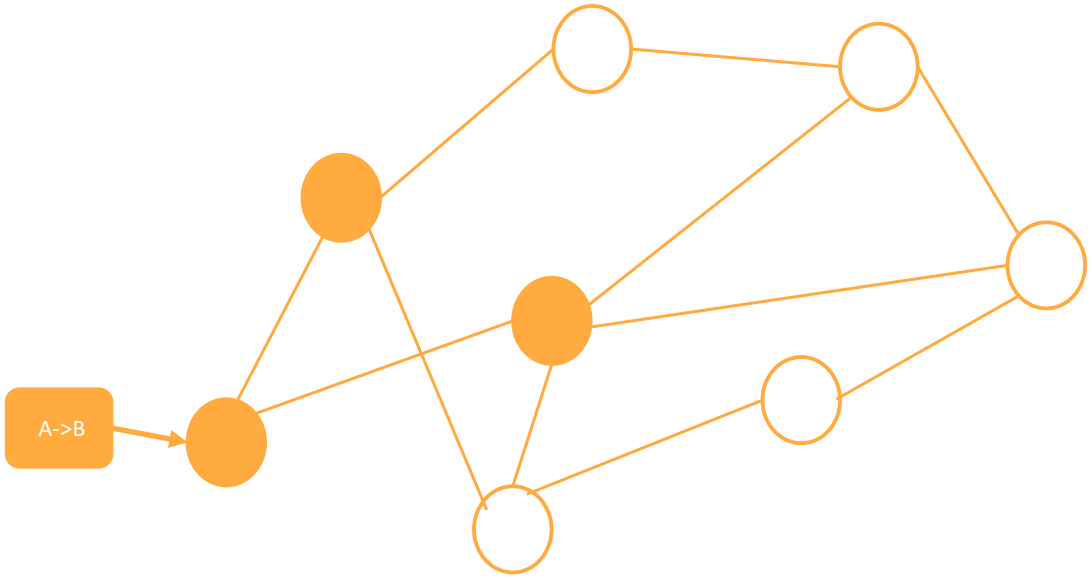
Transaction Verification in Bitcoin



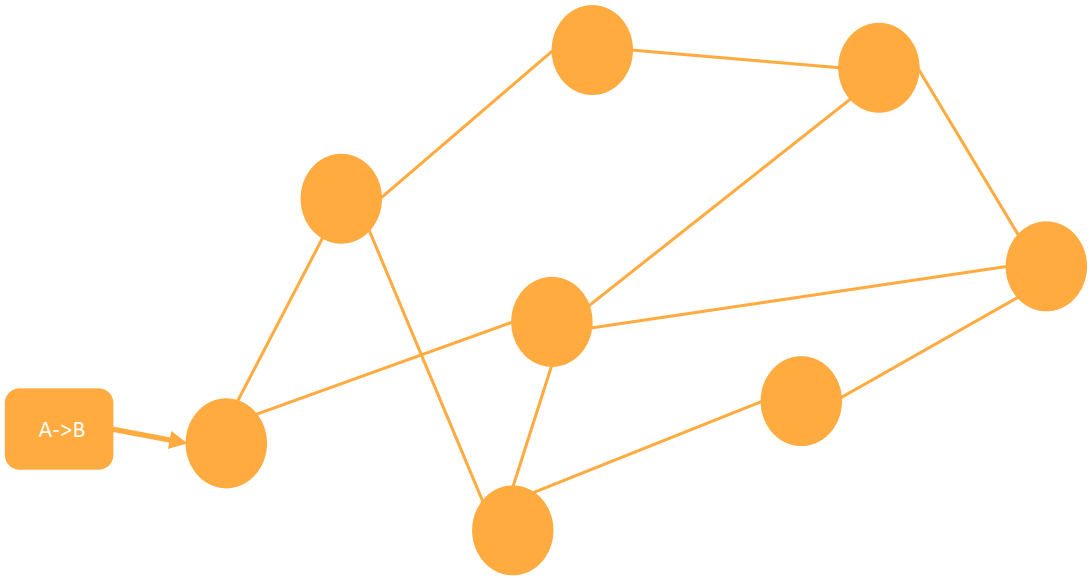
Transaction Verification in Bitcoin



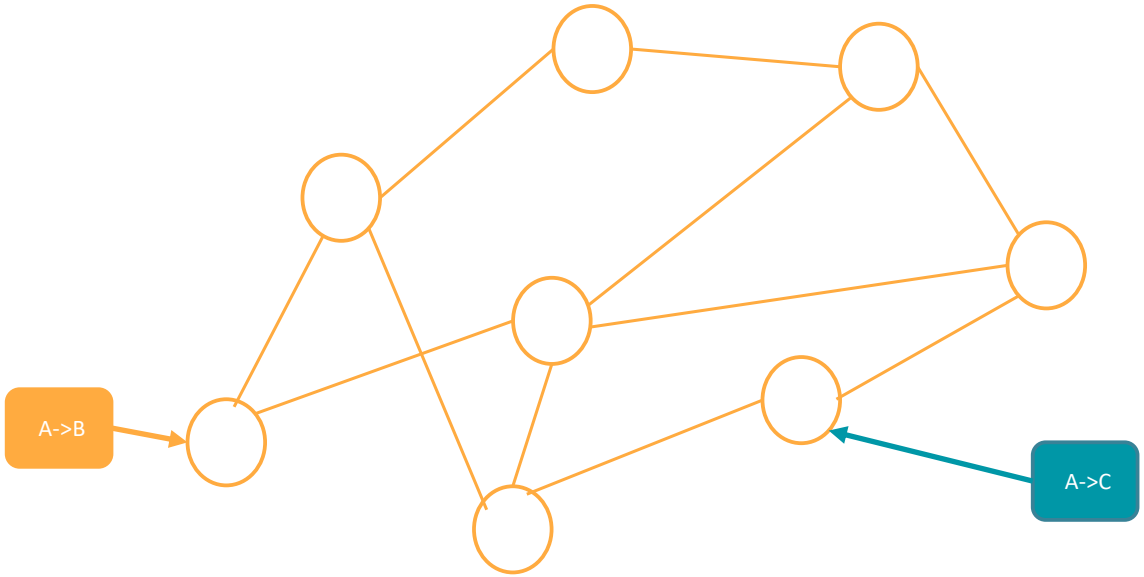
Transaction Verification in Bitcoin



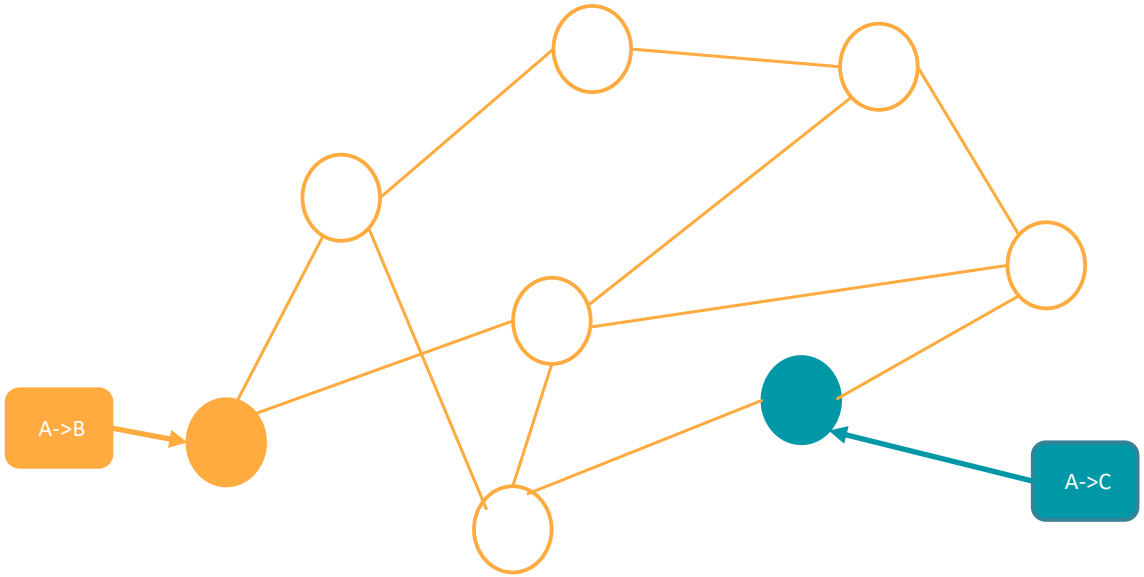
Transaction Verification in Bitcoin



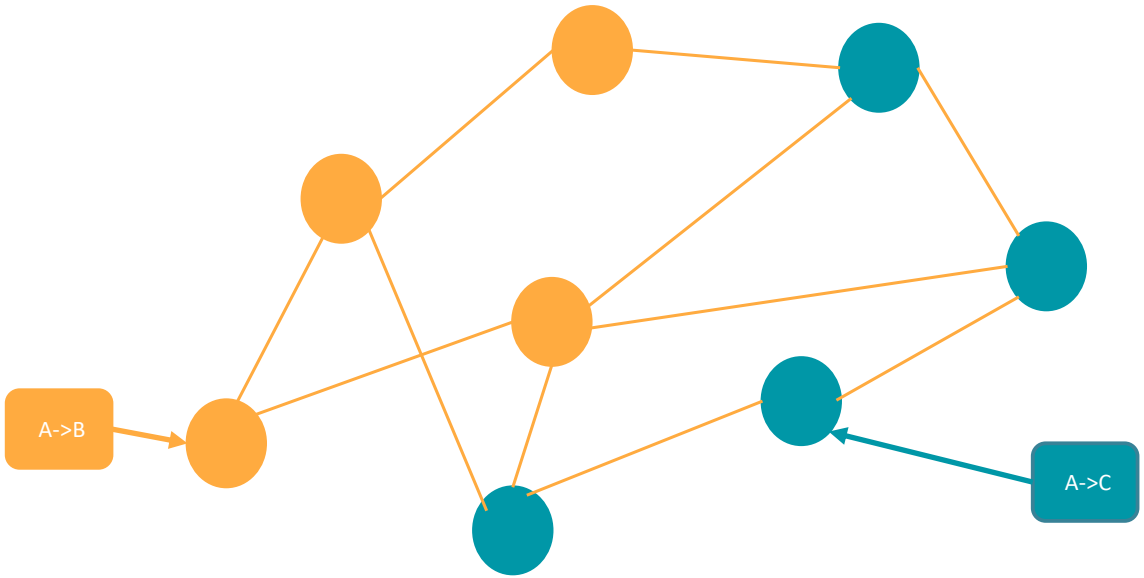
Conflict Resolution



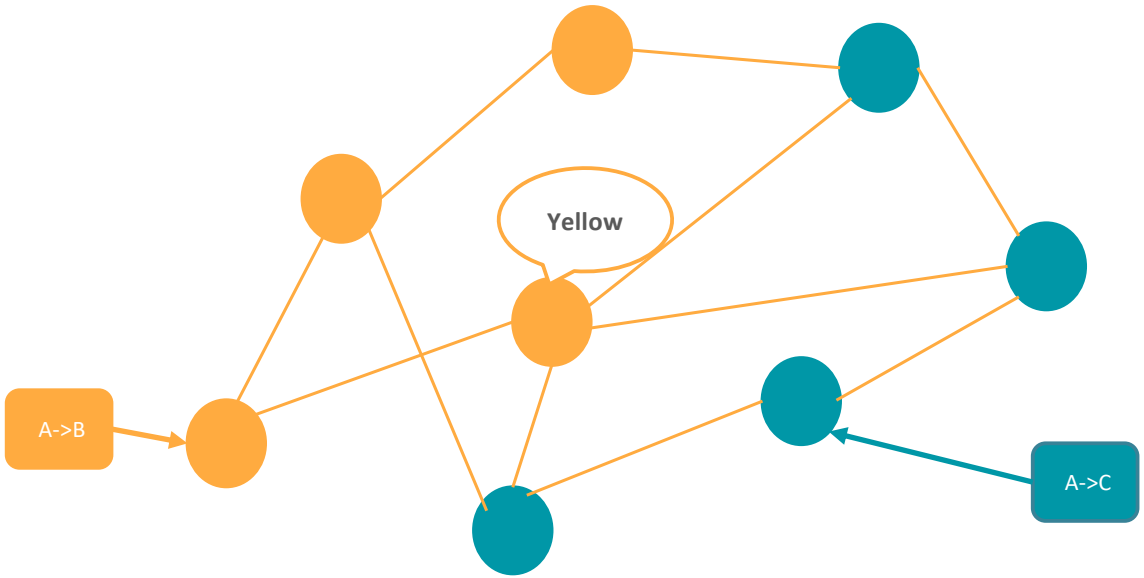
Conflict Resolution



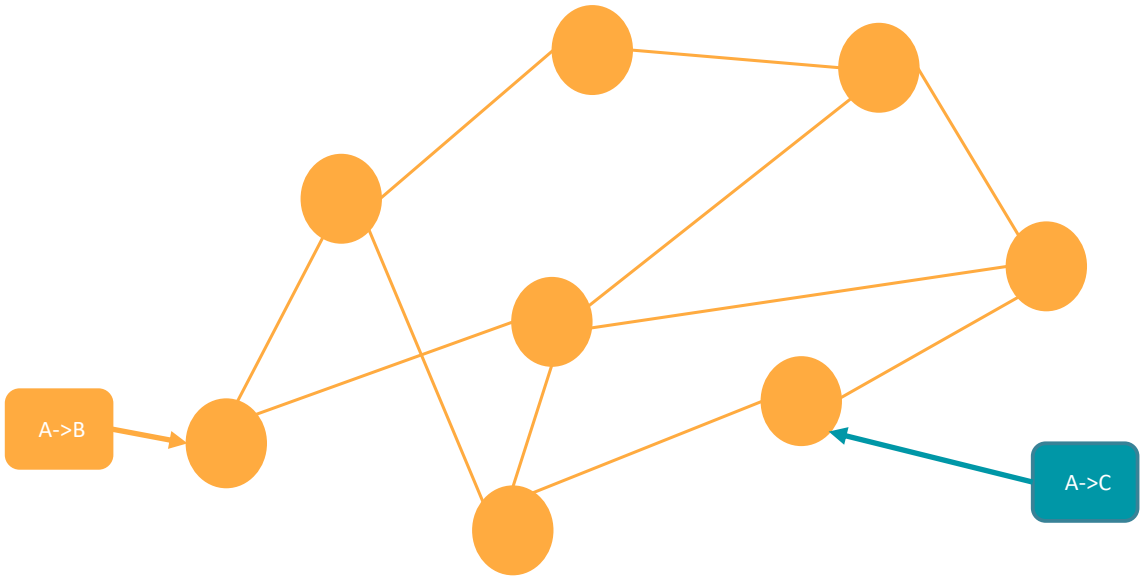
Conflict Resolution



Conflict Resolution

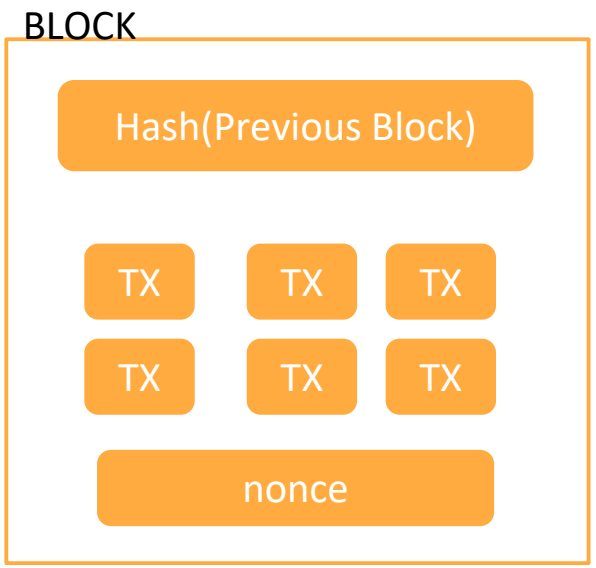


Conflict Resolution



Leader Election

Proof-of-Work



H(Block, nonce=0) =abc3426fe31233

H(Block, nonce=1) =fe541200abc229

H(Block, nonce=2) =0bc3429831233

.

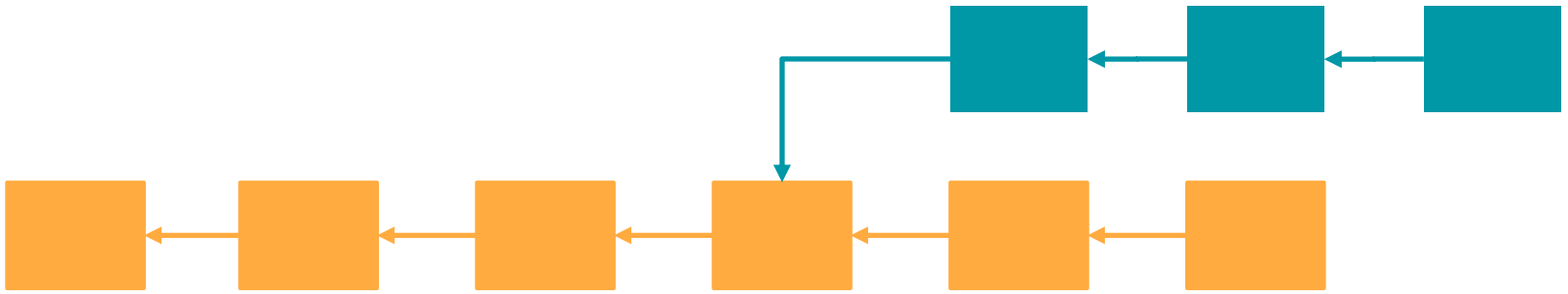
.

.

.

H(Block, nonce=f23) =0000fed98312

Unstable Consensus (Forks)



Question?

What happens if there is a network partition

a)The protocol halts preserving safety
b) Now we have 2 versions of bitcoin that will never merge back
c) The clients do not realize it and can be attacked
d) Free money for everyone

Risk or Wait

In order for a transaction to be valid it needs to be confirmed by the blocks.

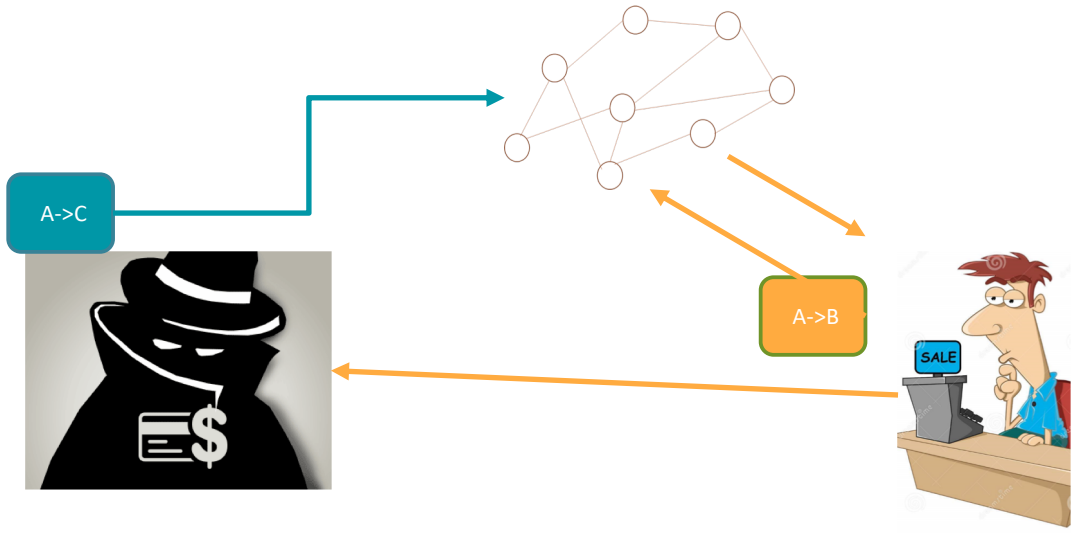
- Each confirmation takes **10 minutes**
- Wait **one hour** to spend your money
- Real time transactions are risky, **double-spending** them is not a hard thing to do.

What’s new about Bitcoin?

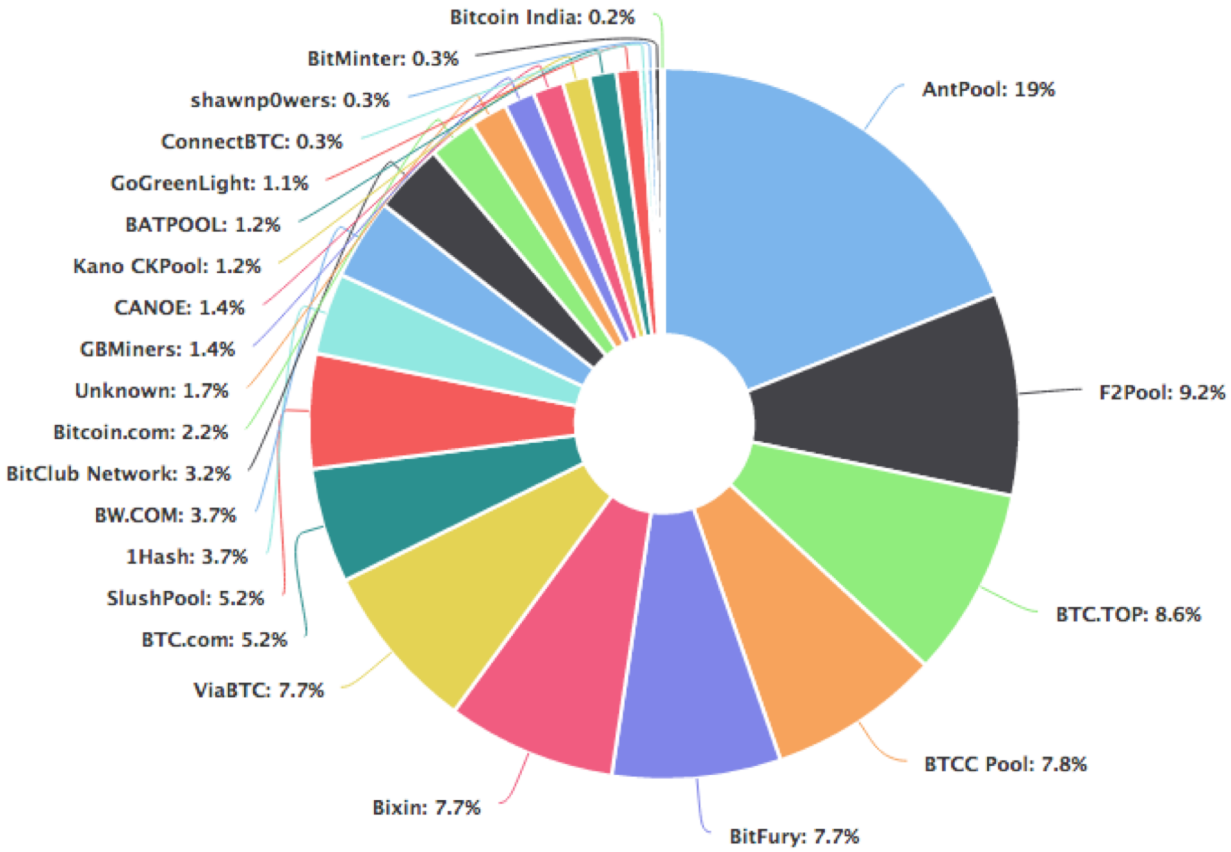
- We do not assume that we know all of the node IDs ahead of time!
  - This undercuts ~30 years of work.
- “Honest majority” measured as a fraction of “hashpower”
- Incentives for following the protocol
- Nodes do not need to output a final decision (aka “stabilizing consensus”)

Double Spending Attack

1) Give transaction to seller
2) Take the product
3) Send a 2nd transaction and create a longer chain



Is Bitcoin Decentralized?

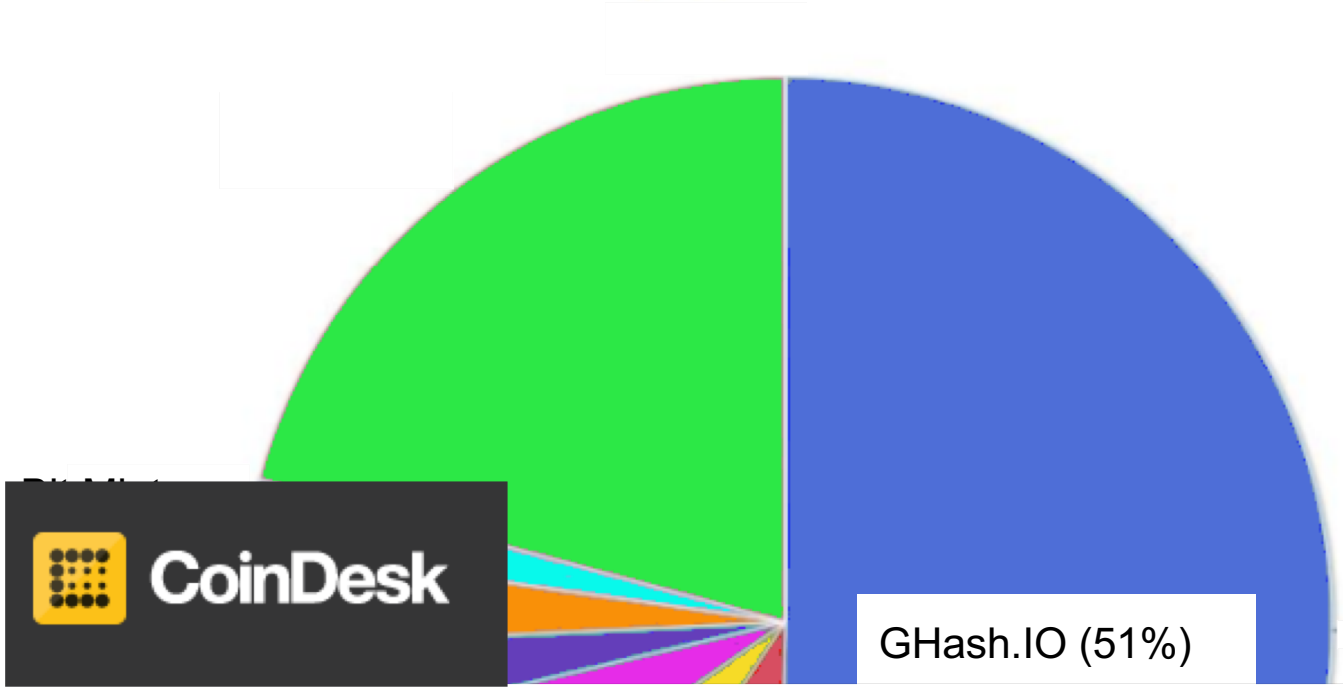


5 Mining pools can collectively attack the system.


Bitcoin Mining Calculator
Hash Rate (GH/s):
800.00
Power (Watts):
900.00
Power Cost ($/kWh):
0.10
Pool Fees %:
0.00
Difficulty:
46684376316.860300(
Block Reward:
25.00000000
Exchange Rate (USD):
243.12000000
Hardware Costs (USD):
0.00
7.948 years
Convert: 2,900.87 days
Days to generate one block mining solo: 2900.87 Day(s)
(can vary greatly depending on your luck)

Explanation of previous slide

It’s well known by now that in practice, miners form coalitions called mining pools, where they essentially follow the instructions of the pool administrator, get a share of the rewards.
The motivation to join pools is straightforward. On average, a block is found every 10 minutes.
Here’s a reasonably priced mining rig, $150. Power consumption like a toaster left on.
Mining solo, it would expect to take 8 years before it finds a block (could happen much earlier or much later, depending on luck)
Miners would prefer to get smaller amounts of revenue in a shorter time frame.



June 12, 2014
GHash.IO large mining pool crisis

MINING • NEWS

GHash Commits to 40% Hashrate Cap at Bitcoin Mining Summit

Stan Higgins | Published on July 16, 2014 at 18:40 GMT

Bitcoin Wallets

- Hot wallet for a few dozen CHF → Mobile


Copay


Airbitz


breadwallet


Bither

- Cold wallet for < 1k CHF → Multi-Sig Desktop


Armory


Electrum

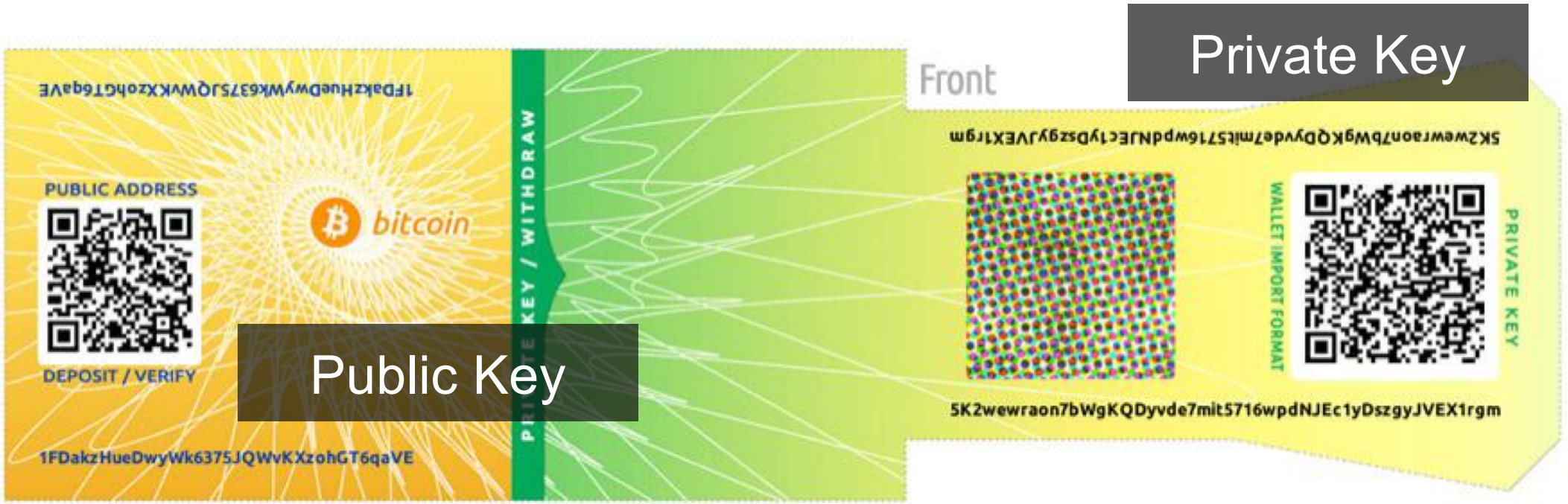

mSIGNA

- Cold wallet for > 1k CHF → Hardware

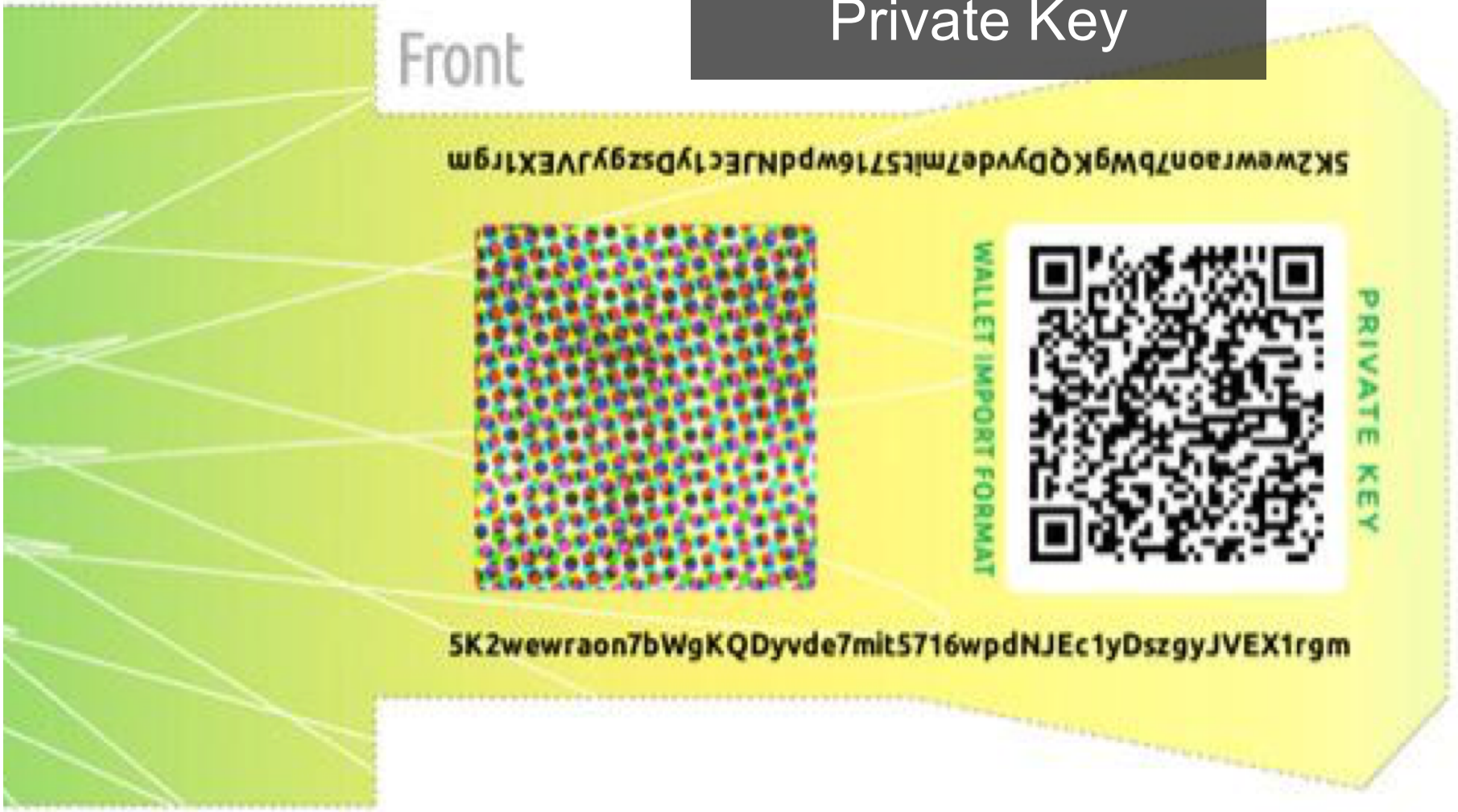
KeepKey


Trezor

Bitcoin Paper Wallet



Private Key


Front
5K2wewraon7bWgKQDyvde7mit5716wpdNJEc1yDszgyJVEX1rgm
WALLET IMPORT FORMAT
PRIVATE KEY
5K2wewraon7bWgKQDyvde7mit5716wpdNJEc1yDszgyJVEX1rgm

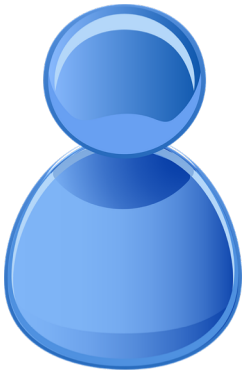

PUBLIC ADDRESS
bitcoin
DEPOSIT / VERIFY
1FDakzHueDwyWk6375JQWvKXzohGT6qaVE

Public Key

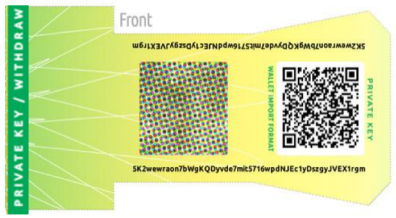
Alice and Bob are only identified by public keys

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Transfer 10 Bitcoins from me to Bob.



Alice



Signed with Alice's private key

Bitcoin Network

1BTC

3BTC

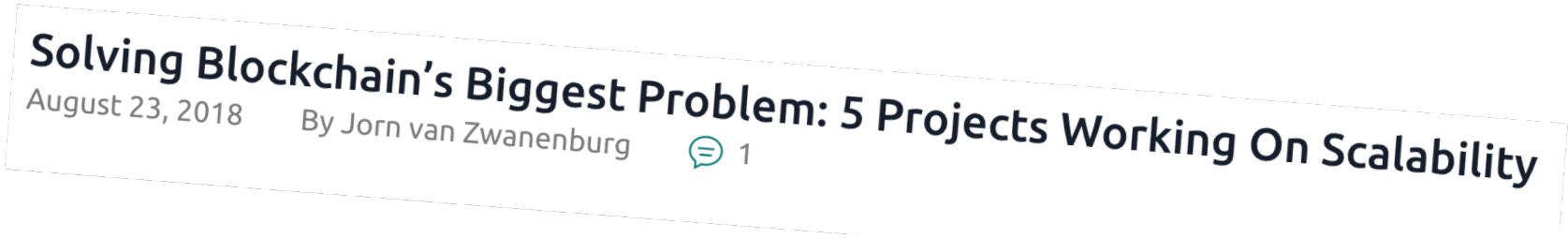
1BTC

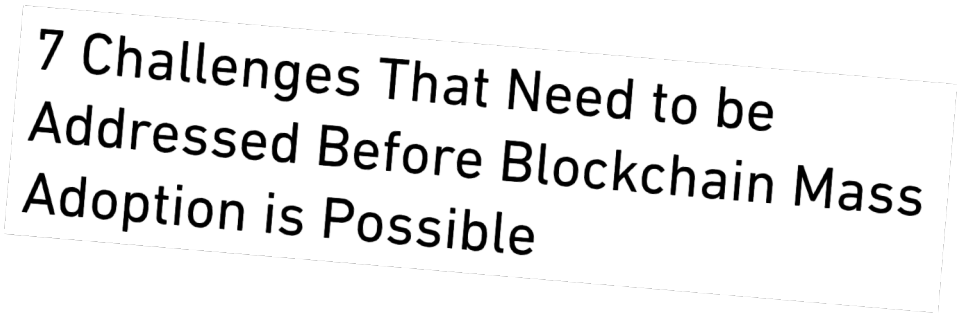
5BTC

Bitcoin performance

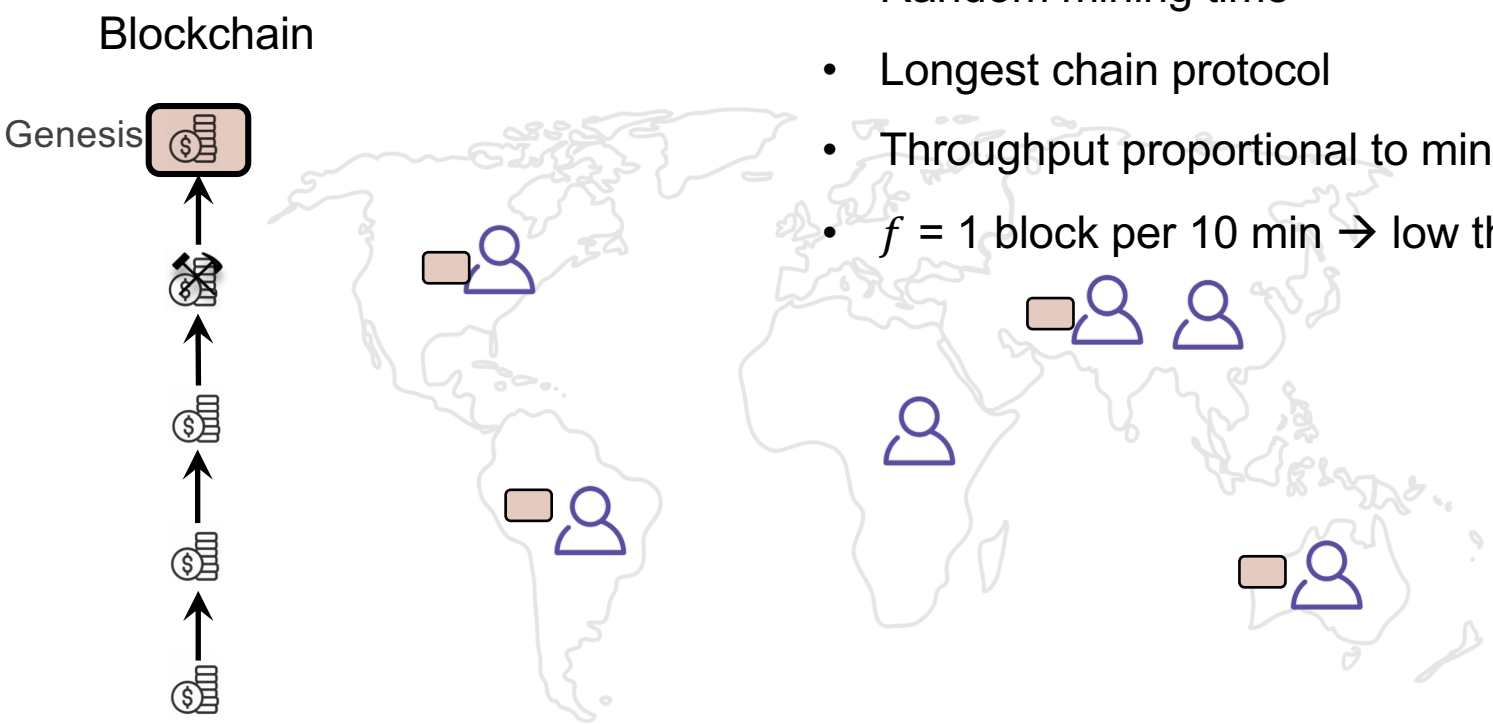
Security	Transaction throughput	Confirmation Latency
✓	✗	✗
50% adversary	5 transactions/s	hours

Principal challenge: Scalability



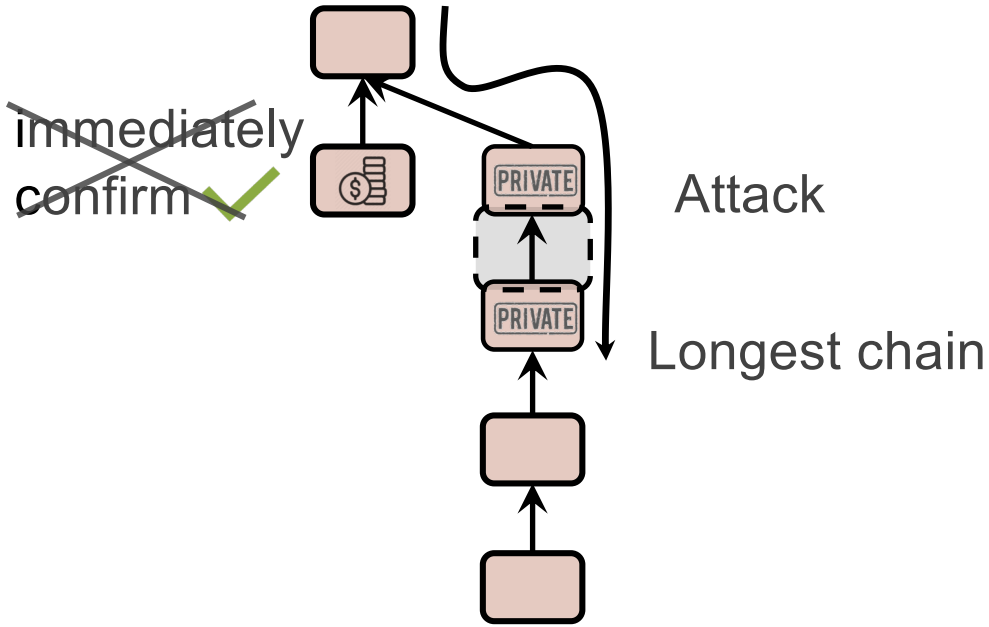


Bitcoin: A Distributed ledger



- Random mining time
- Longest chain protocol
- Throughput proportional to mining rate f
- f = 1 block per 10 min → low throughput

Transaction confirmation

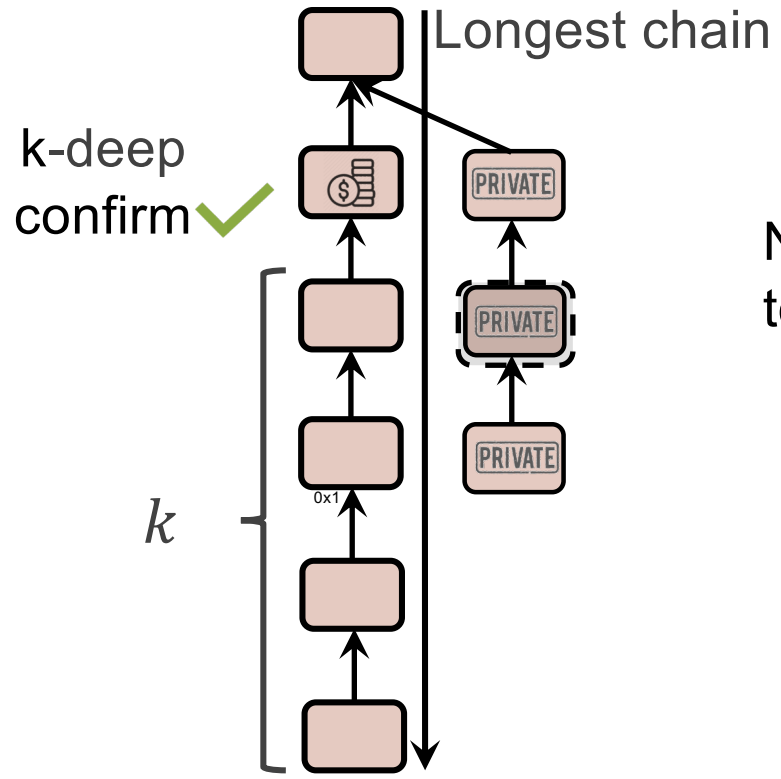


Transaction confirmation

30% adversary power

k=0 $\varepsilon = 1.0000000$
k=5 $\varepsilon = 0.1773523$
k=10 $\varepsilon = 0.0416605$
k=15 $\varepsilon = 0.0101008$
k=20 $\varepsilon = 0.0024804$
k=25 $\varepsilon = 0.0006132$
k=30 $\varepsilon = 0.0001522$
k=35 $\varepsilon = 0.0000379$
k=40 $\varepsilon = 0.0000095$
k=45 $\varepsilon = 0.0000024$
k=50 $\varepsilon = 0.0000006$

Nakamoto's table

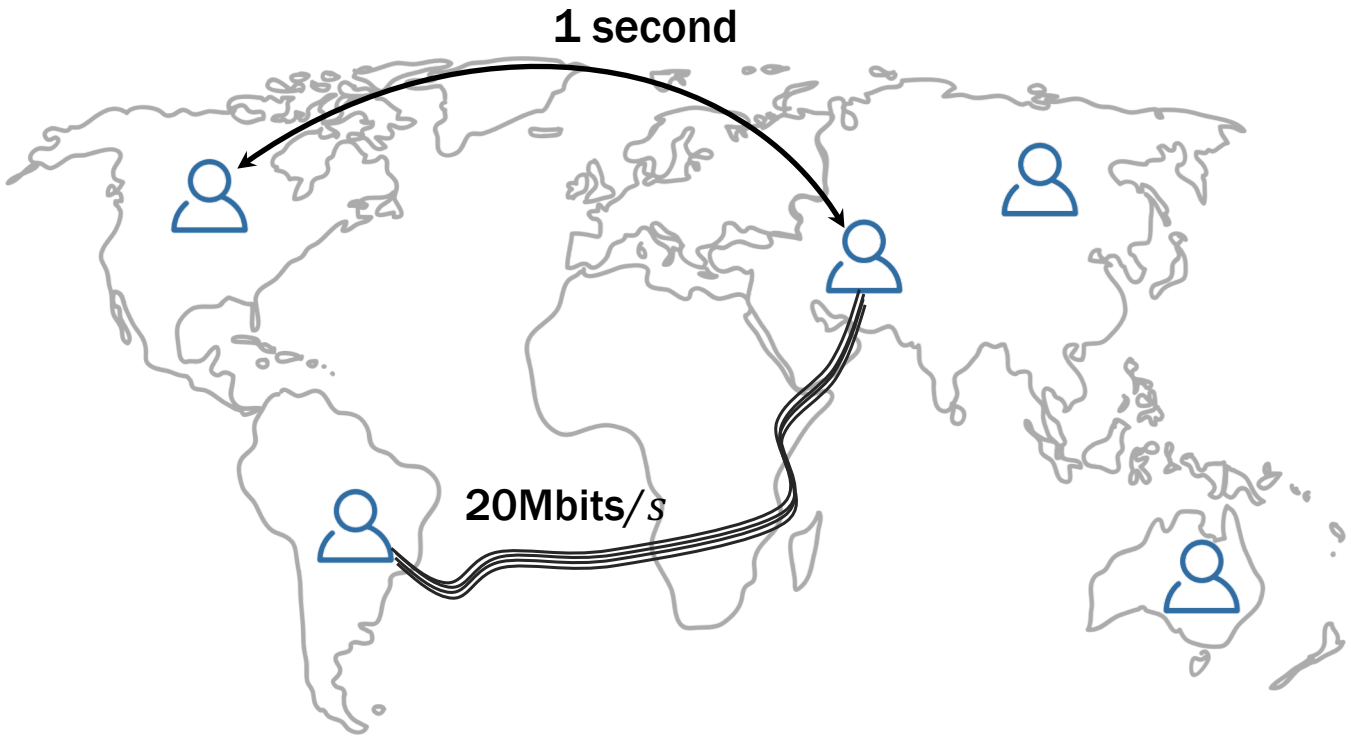


Needs to wait for many votes to confirm.

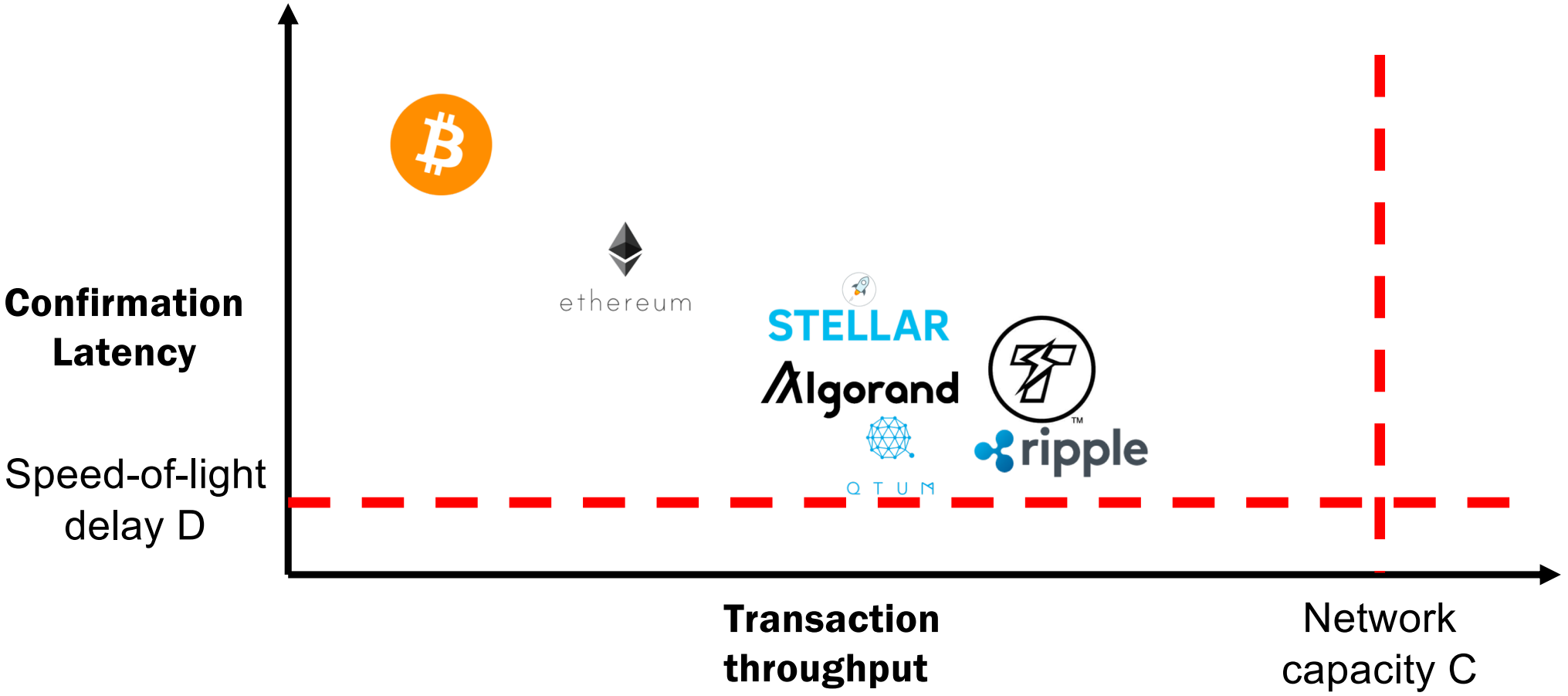
Physical limits

Network capacity C

Speed-of-light propagation delay D



Physical limits with Bitcoin security?

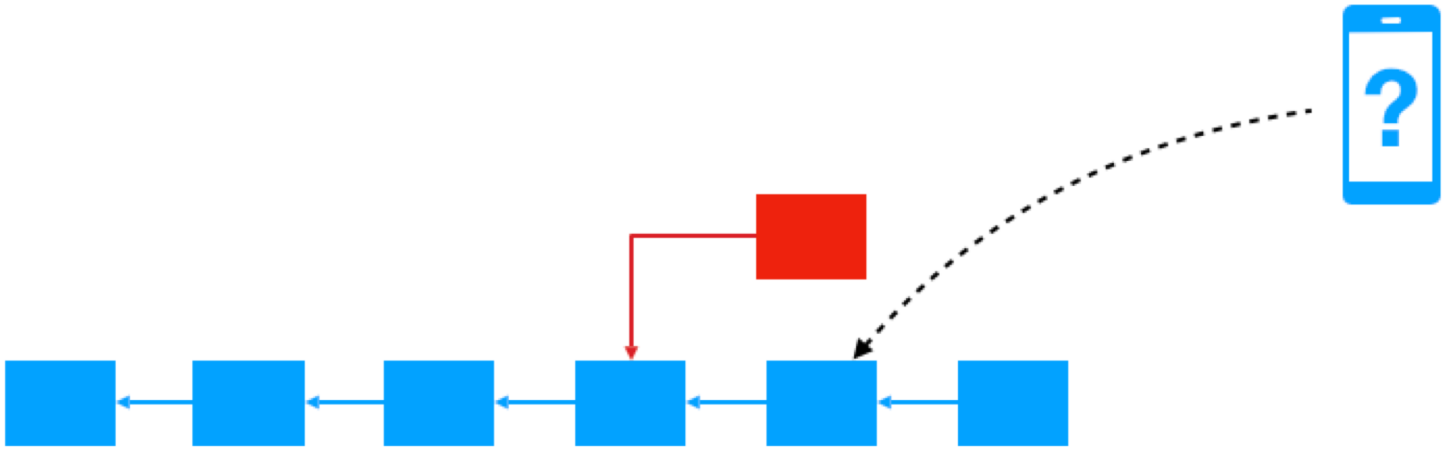


Optimization - Skipchains

- Thesis of Eleftherios Kokoris, EPFL 2019
- Supervised by Bryan Ford

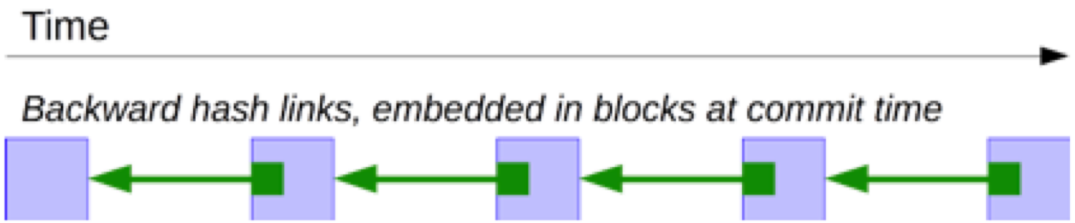
Problem: Efficient Verification

- How does a “light” (low-power, mobile) client securely confirm a recent (or old) transaction?
  - Especially after being offline for months, years?
  - Without “just trusting” central party (exchange)?

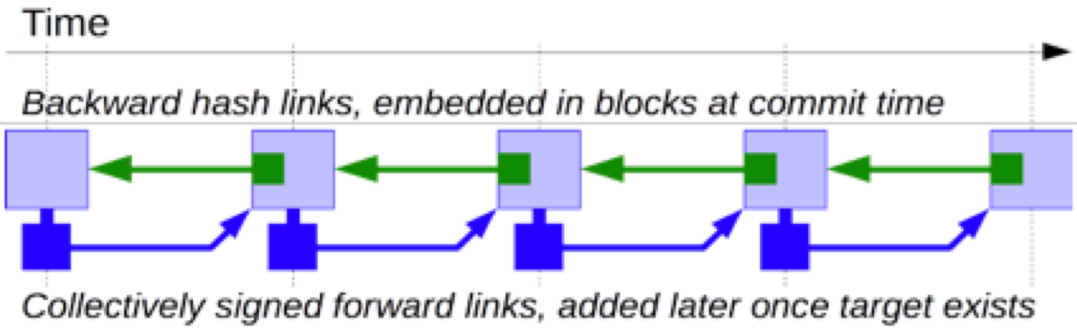


Backward and Forward Verifiability

- Standard blockchains traversable only **backward**
  - Via hash back-links from current head

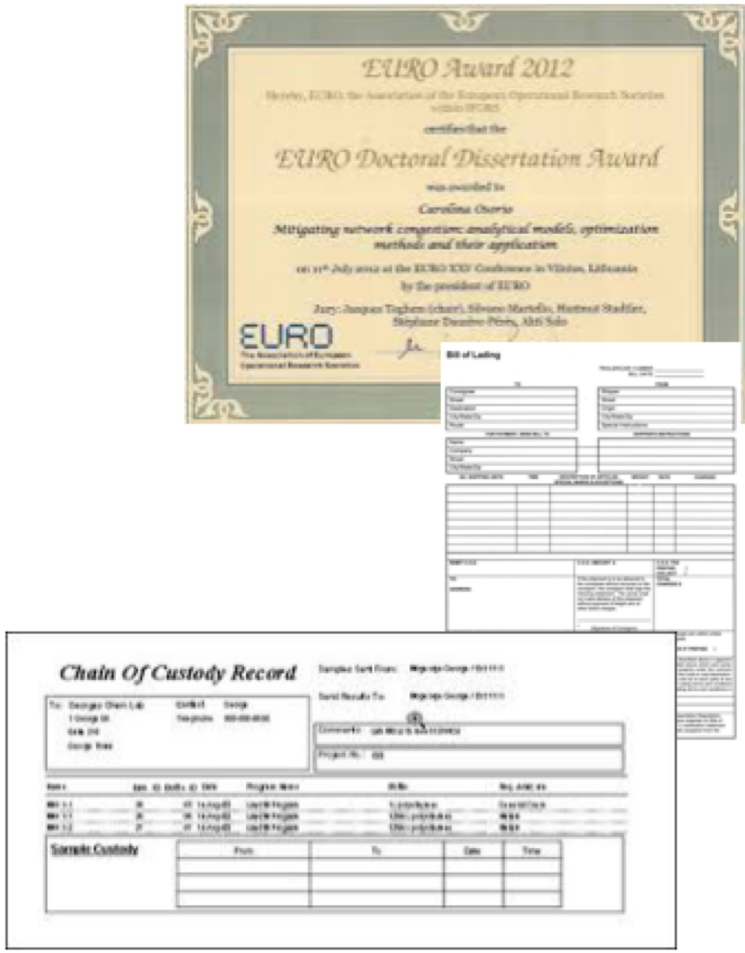


- We add traversability **forward in time***
  - Collective signature by prior consensus group



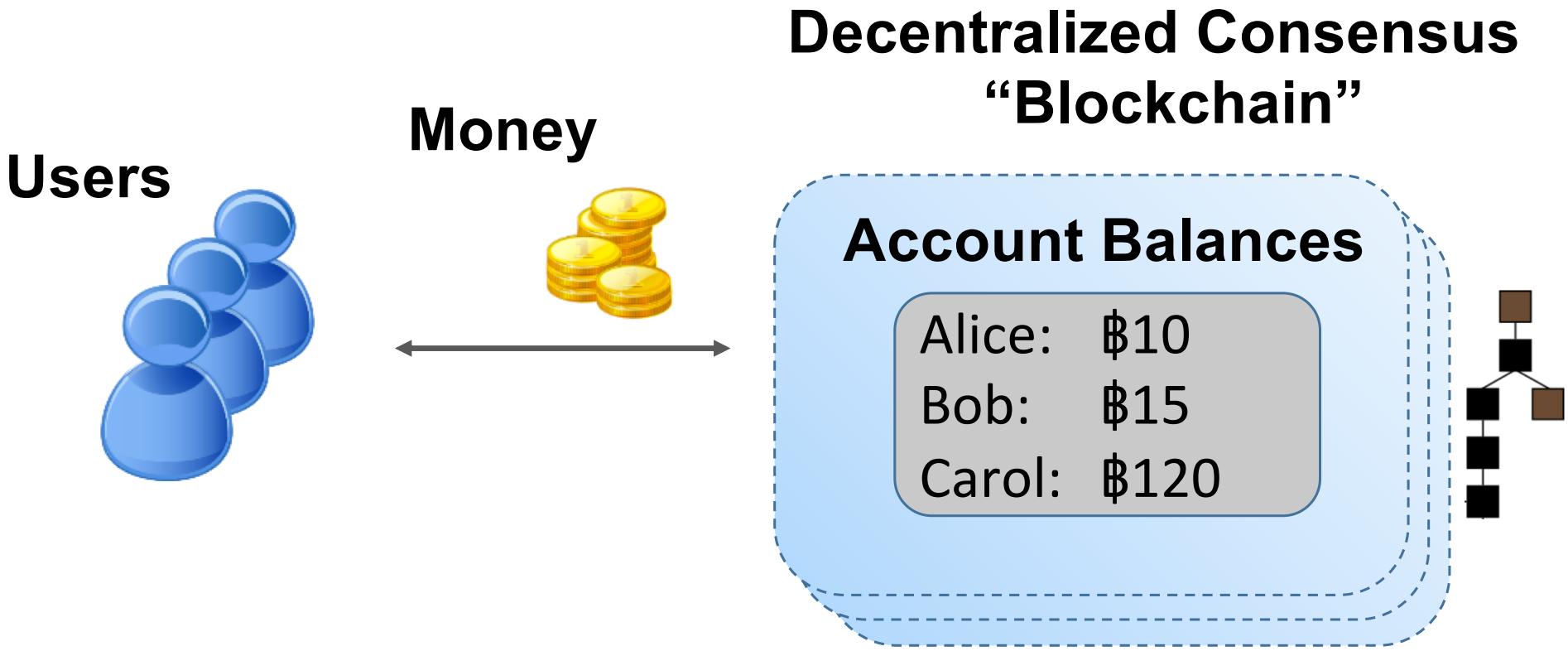
Applications of SkipChains

- Enable Offline/P2P verification
  - Works even if Internet is unavailable, slow, costly
- Broad applications
  - Software/key updates
  - Blockchain-Attested Degrees, Awards, …
  - Chain-of-Custody, Bills of Lading, …

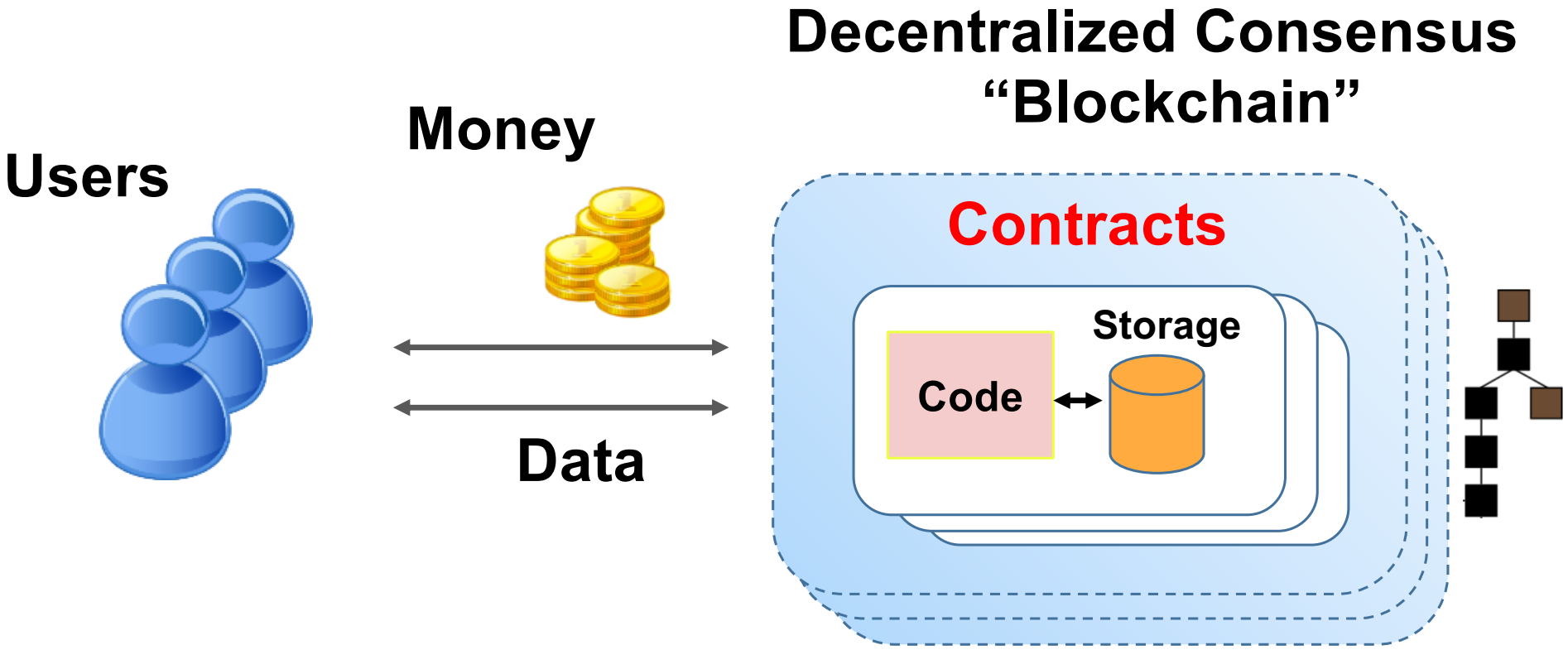


Smart Contracts

Digital currency is just one application on top of a blockchain



Smart Contracts: user-defined programs running on top of a blockchain



About Ethereum

Crowdfunded ~$20M in ~ a month

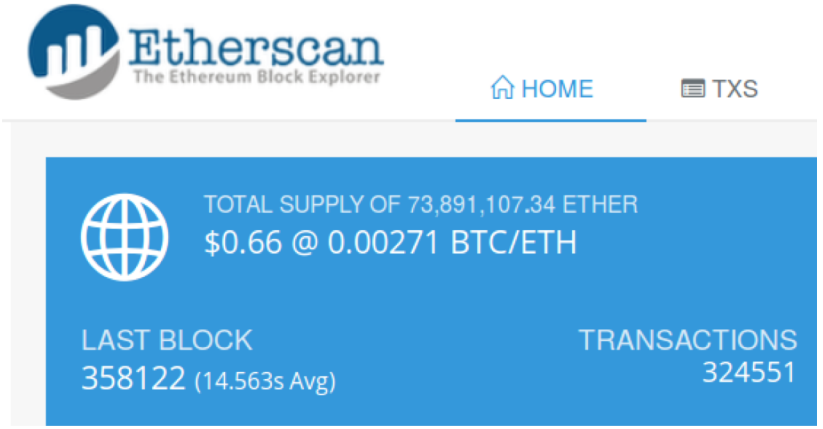
Popularized a grand vision of

“generalized” cryptocurrency

Vitalik Buterin (born 1994, founded Ethereum in 2013)

Flexible scripting language

“pyethereum” simulator, 2014



Key challenges in smart contract design:

- Smart Contracts in Ethereum can be trusted for correctness and availability, but not **privacy**
- Blockchain resources are expensive
- Race conditions and temporary forks

Examples

- “Namecoin”: a DNS replacement
  - Initially, all names are unregistered.
  - Anyone can claim an unregistered name.
  - Once it’s registered, no one can change it.

Examples: Namecoin

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

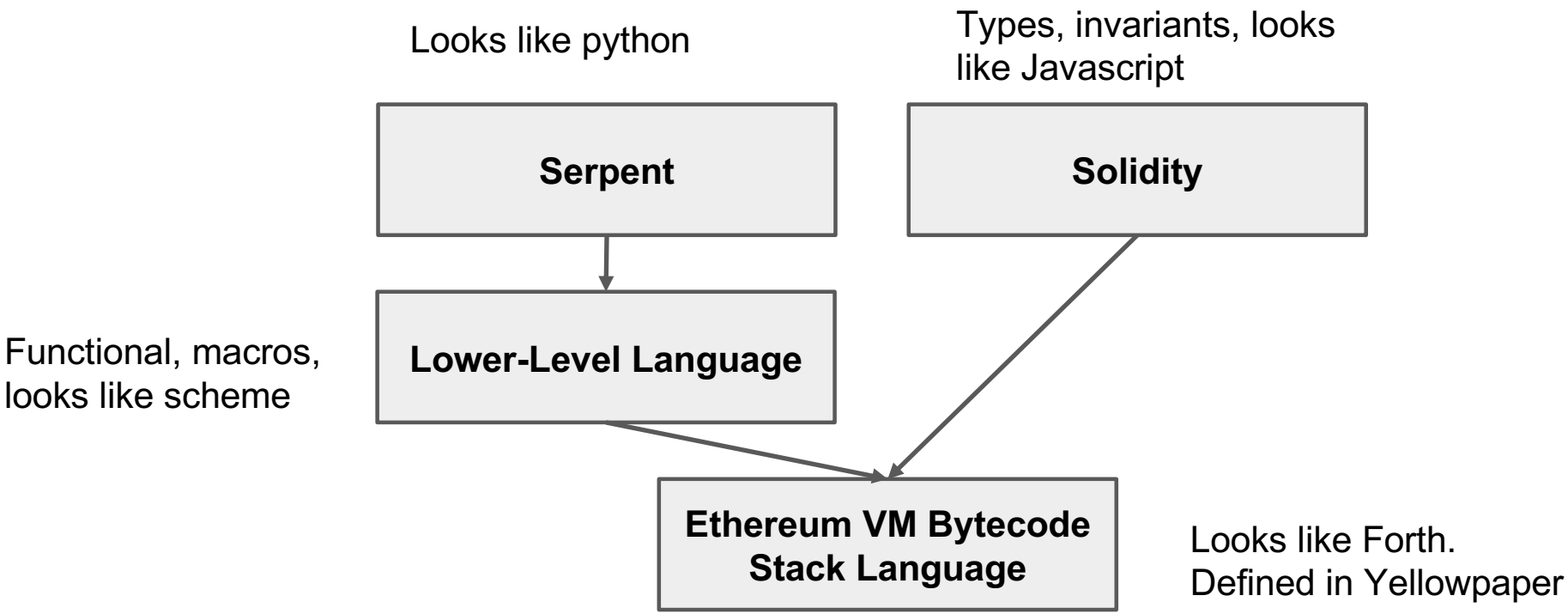
```
def register(k, v):

    if !self.storage[k]: # Is the key not yet taken?
        # Then take it!

        self.storage[k] = v

        return(1)

    else:

        return(0) // Otherwise do nothing
```

Reasons to be excited about this

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

- Programmable money is an excellent idea
  - Doesn’t strictly require a cryptocurrency, but it’s here first
- Exists outside mainstream business culture / jurisdictions
  - Could avoid government overreach and monopolies
- We might agree on a standard business database format

Ethereum Languages



Basics

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

- Submit a transaction to the blockchain in order to create a new contract
  - transaction contains the *code* as data
  - contracts have an “account balance” denominated in Ether
  - contracts have a persistent “storage” file
- Submit transactions to the blockchain to interact with the contract
  - transactions can contain monetary “value,” added to the account
  - procedure calls

Gas

Every **Tx** defines:

recipient, from, data, amount, ***gasPrice, gasLimit***

Validity: amount + gas*price <= accounts[from].balance

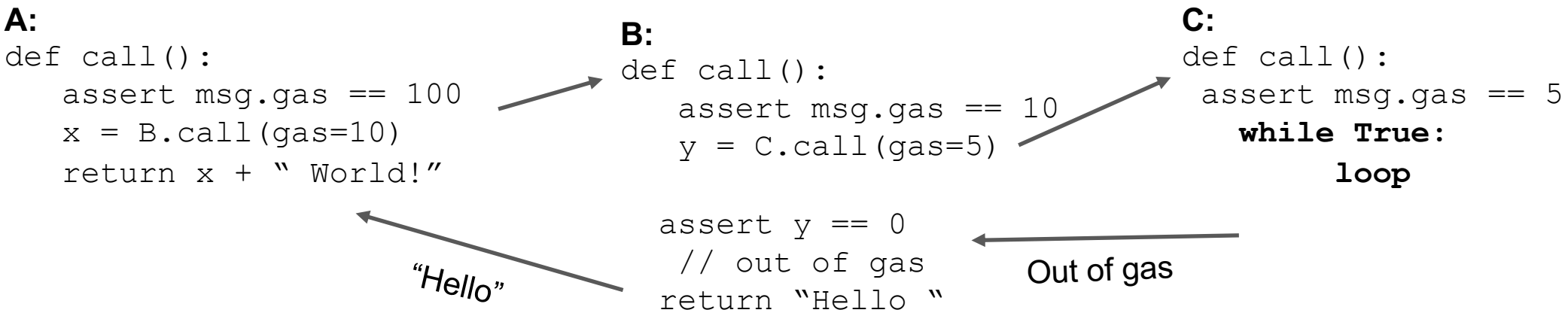
Update: recipient.balance += amount

from.balance -= amount **+ gas*price**

execute(code, amount, balance, mem, **gas**)

from.balance += unusedGas * price

Contract Call Stack



Returns “Hello World!”

What's the deal with The DAO?

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

- Crowdfunding instrument that raised $150+ million dollars of ETH
  - Initially, the goal was $10,000 to fund a Bike Lock company
- TheDAO contract is ambitious! Lets users pool their investments and vote
- A subtle bug in TheDAO led to the loss of $50 million worth of tokens
- The Ethereum community developed a "Hard Fork" to cancel the theft
- A faction of dissenters are maintaining "Eth Classic", also traded on exchanges

Alternative: Proof-of-Stake (PoS)

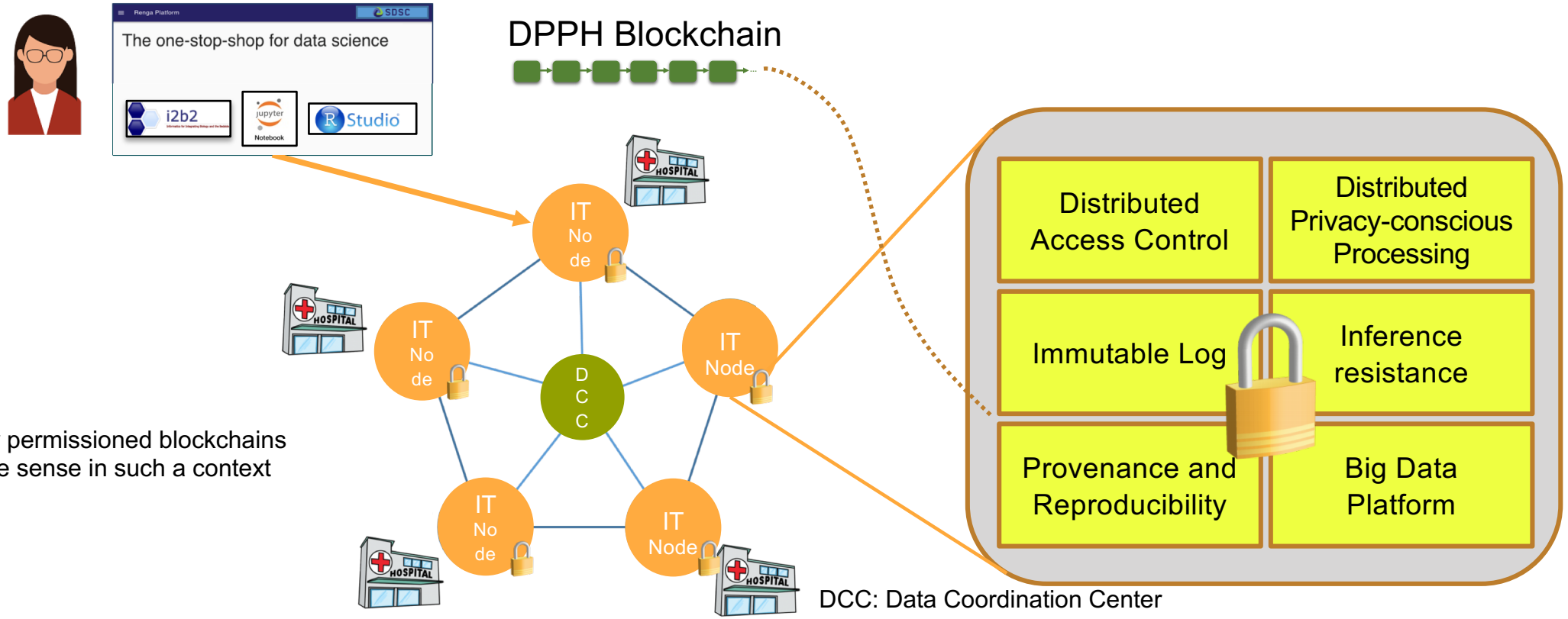
- **Proof-of-Stake:** assigns consensus shares in proportion to prior capital investment
  - Could address energy waste problem
  - Major unsolved security & incentive problems
- But PoS requires secure public randomness…

Alternative: Permissioned Ledgers

- Also called “permissioned blockchains” or “private blockchains”
- Just decide **administratively** who participates; Fixed or manually-changed group of trustees
- **Liability clearly defined**
- No proof-of-work needed → low energy cost
- More mature consensus protocols applicable
- Higher human organizational costs
- No longer open for “anyone” to participate

Tremendous potential for regulated sectors such as finance and health

Blockchain in Healthcare: Logging of Data Access

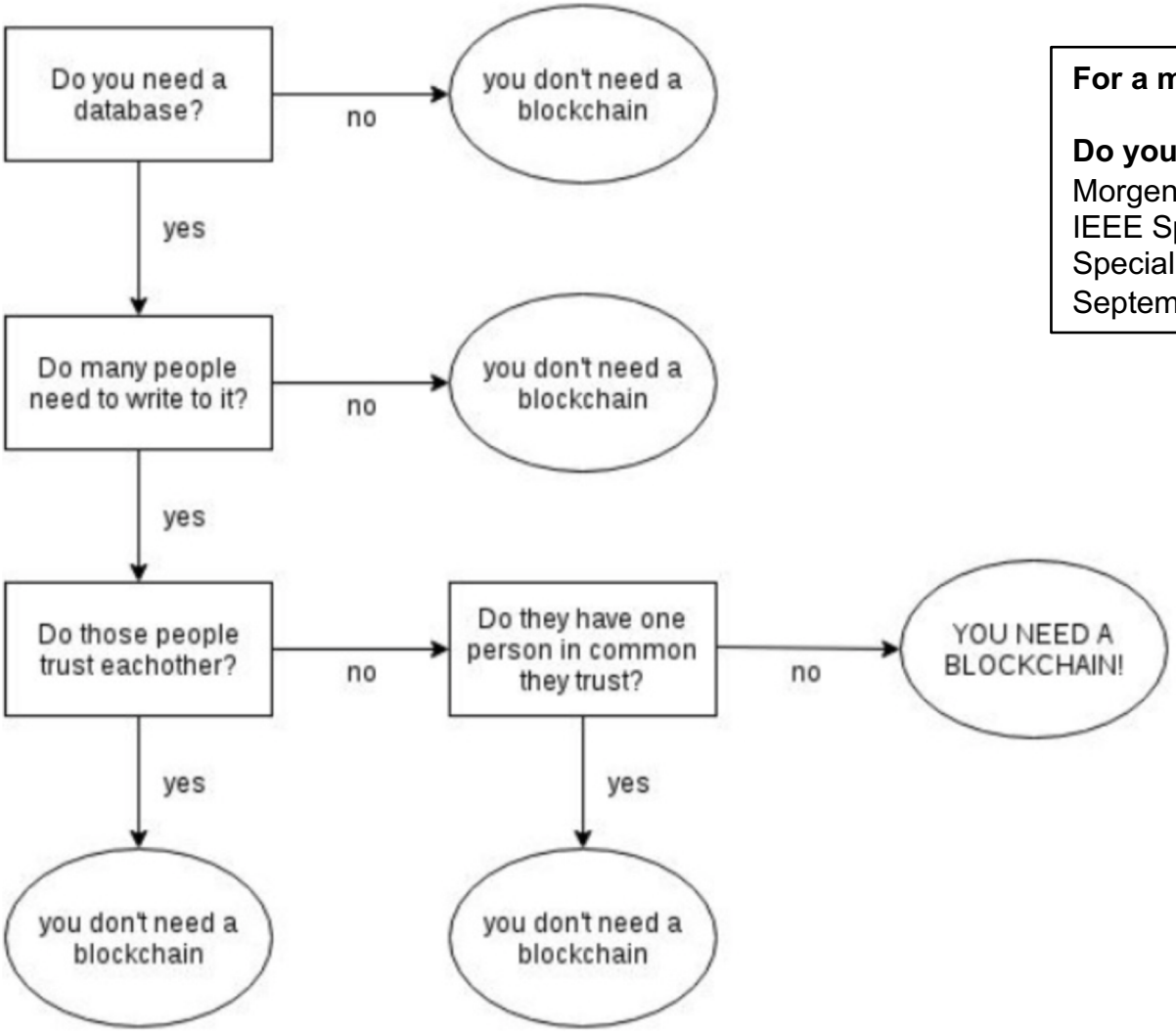


Only permissioned blockchains
Make sense in such a context

DCC: Data Coordination Center

Other application: smart contracts to support patient consent

Do You Need a Blockchain?



For a more complete version:

Do you Need a Blockchain?
Morgen E. Peck
IEEE Spectrum
Special Issue on Blockchains
September 2017

Additional resources

Lectures by Andrew Miller:

http://soc1024.ece.illinois.edu/teaching/ece598am/fall2016/

Online Coursera Course:

https://www.coursera.org/learn/cryptocurrency

Bitcoin and cryptocurrencies book, with many videos – Narayanan, Bonneau, Felten, Miller (2016):

- http://bitcoinbook.cs.princeton.edu