RSAConference2019

San Francisco | March 4 – 8 | Moscone Center

BETTER.

SESSION ID: LAB3-R09

Blockchain, Cryptocurrency, Smart Contracts and Initial Coin Offerings: A Technical Perspective

Tom Plunkett

Consulting Solutions Director
Oracle

Captain Brittany Snelgrove

United States Marine Corps

Captain Brandan Schofield

United States Marine Corps

Agenda

Blockchain and Cryptocurrency Overview
Cryptography: Hashes, Digital Signatures, PKI
Bitcoin and Blockchain
Ethereum and Solidity
Identity and Access Management with Blockchain
Bitcoin Lab Demo
Ethereum Lab Demo

Blockchain and Cryptocurrency

- Over 6000 Cryptocurrencies exist, and over 1000 new ones being created every year.
- VCs invested over $3 billion in 2018.
- Initial Coin Offerings over $15 billion in 2018.
- Over 3000 blockchain patent applications filed.
- Over 30 Presentations at RSA Conference about Blockchain

Hash functions:

takes any string as input

fixed-size output (example 256 bits)

efficiently computable

Security properties:

collision-free (Nobody can find x and y such that x != y and H(x)=H(y))

hiding (Given H(x), infeasible to find x)

puzzle-friendly (best search strategy is to just try random values of x)

Hash as message digest

If we know $H(x) = H(y)$, it's safe to assume that $x = y$.

To recognize a file that we saw before, just remember its hash.

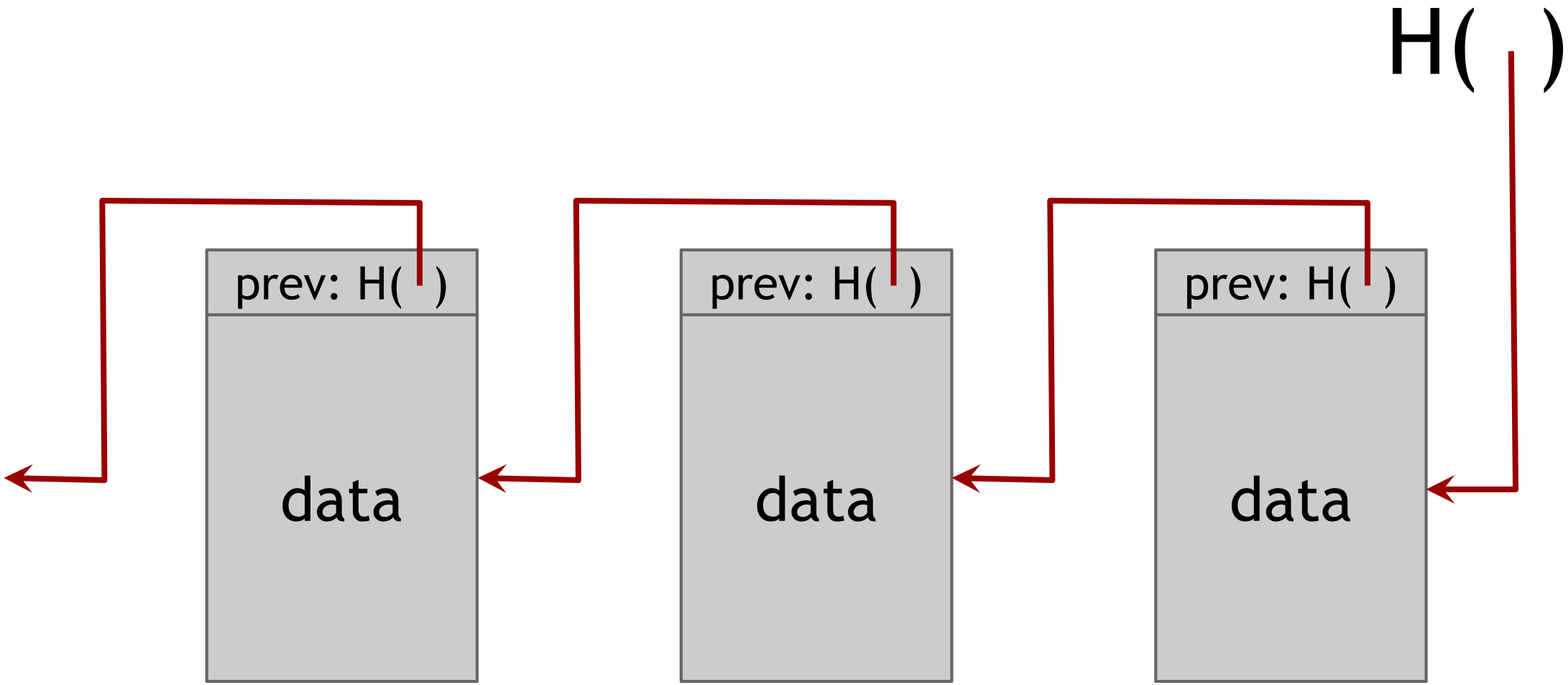
Useful because the hash is small.

Hash as a Commitment

Want to “seal a value in an envelope”, and “open the envelope” later.

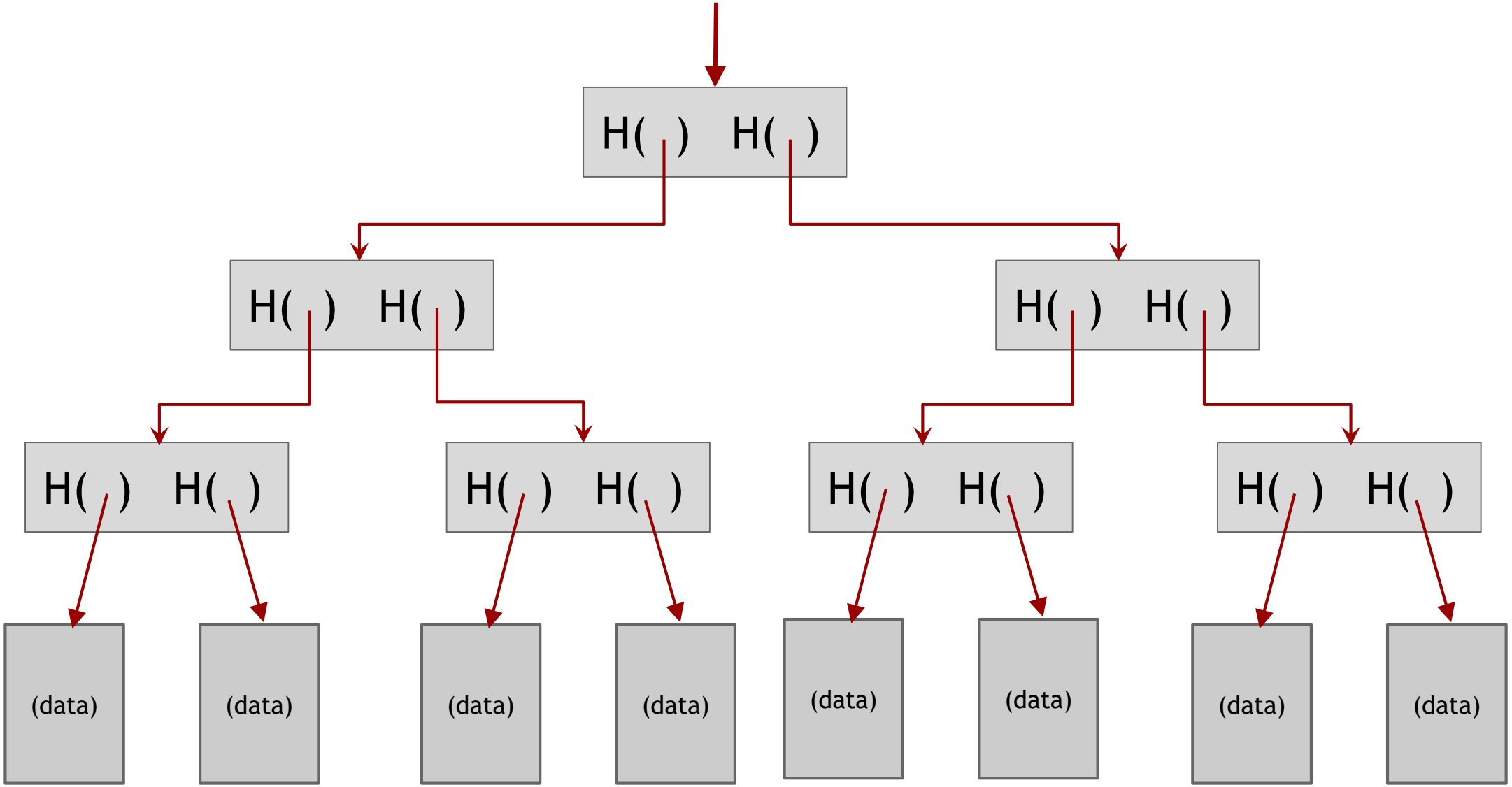
Commit to a value, reveal it later.

linked list with hash pointers = “block chain”



use case: tamper-evident log

binary tree with hash pointers = “Merkle tree”



Digital Signatures, Public/Secret Keys

(sk, pk) := generateKeys(keysize)

sk: secret signing key

pk: public verification key

sig := sign(sk, message)

isValid := verify(pk, message, sig)

Digital Signatures

"valid signatures verify"

verify(pk, message, sign(sk, message)) == true

"can't forge signatures"

adversary who:

knows pk

gets to see signatures on messages of his choice

can't produce a verifiable signature on another message

Aspects of decentralization in Bitcoin

1. Who maintains the ledger?
2. Who has authority over which transactions are valid?
3. Who creates new bitcoins?
4. Who determines how the rules of the system change?
5. How do bitcoins acquire exchange value?

Beyond the protocol:
exchanges, wallet software, service providers...

Aspects of decentralization in Bitcoin

Peer-to-peer network:

open to anyone, low barrier to entry

Mining:

open to anyone, but inevitable concentration of power

often seen as undesirable

Updates to software:

core developers trusted by community, have great power

Distributed consensus

Key technical challenge of decentralized electronic cash: distributed consensus

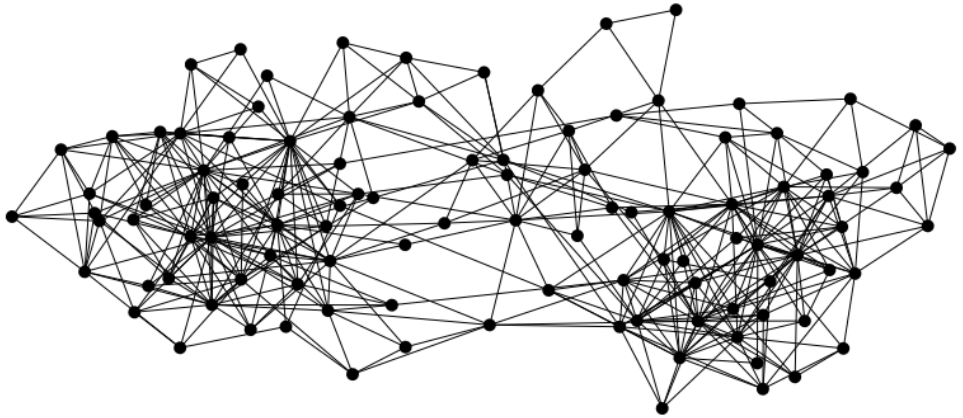
Definition: The protocol terminates and all correct nodes decide on the same value. This value must have been proposed by some correct node.

Bitcoin is a peer-to-peer system

When Alice wants to pay Bob:
she broadcasts the transaction to all Bitcoin nodes

signed by Alice

Pay to pk_{Bob} : H( )



Note: Bob's computer is not in the picture

Why consensus is hard

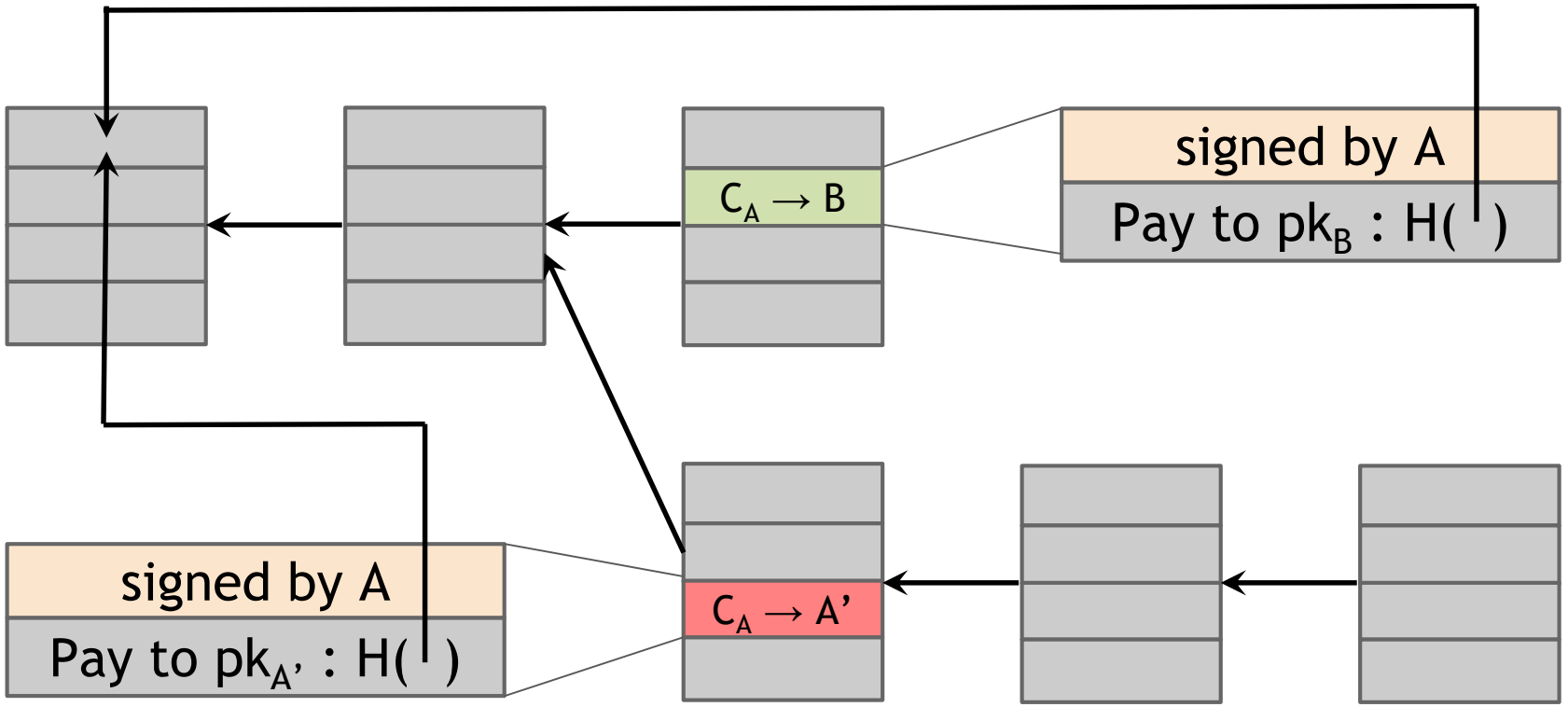
Nodes may crash
Nodes may be malicious

Network is imperfect

- Not all pairs of nodes connected
- Faults in network
- Latency

No notion of global time

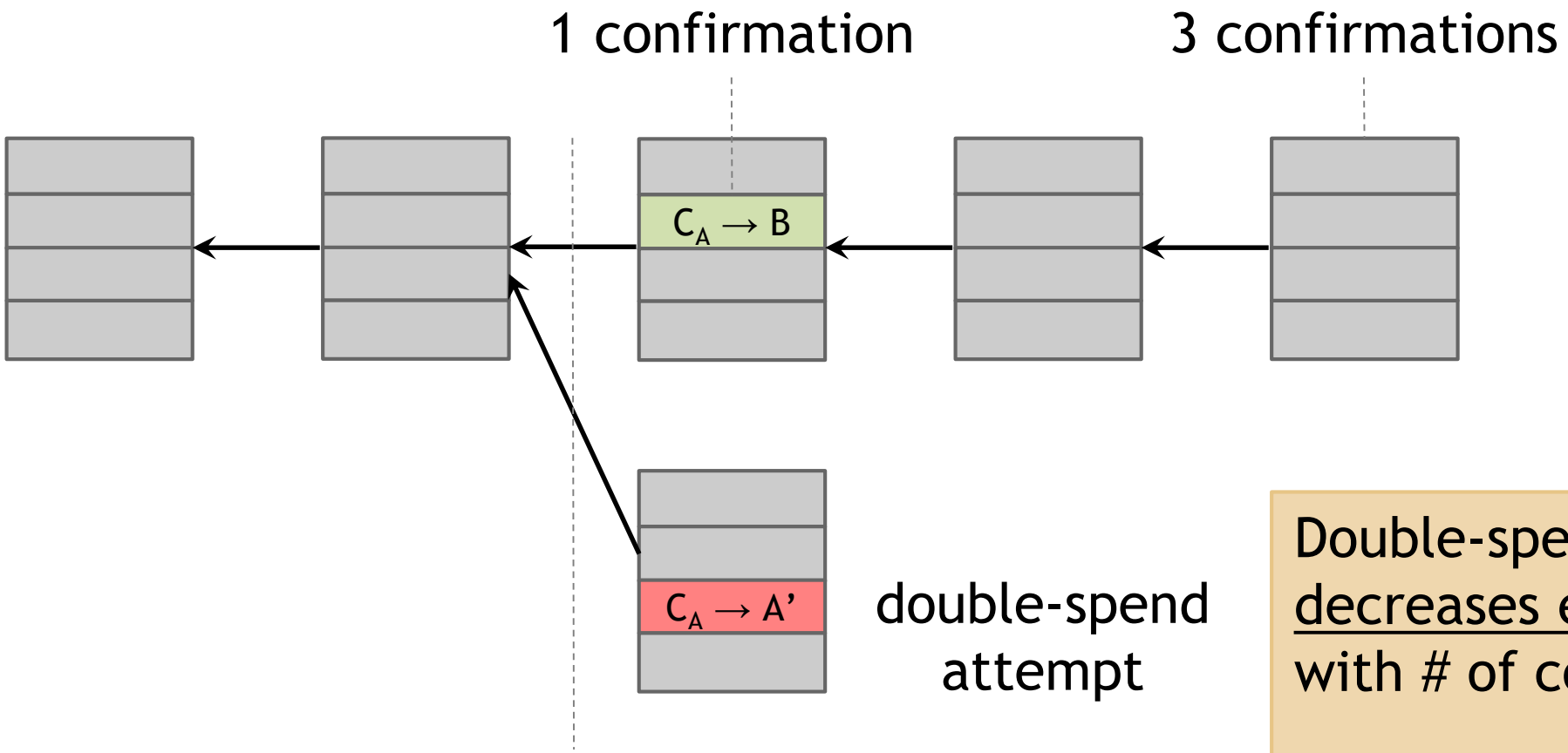
What can a malicious node do?



Double-spending attack

Honest nodes will extend the <u>longest valid branch</u>

From Bob the merchant's point of view



Double-spend probability <u>decreases exponentially</u> with # of confirmations

Most common heuristic: 6 confirmations

Incentives: block rewards and mining fees

Creator of block gets to

- include special coin-creation transaction in the block
- choose recipient address of this transaction

Block creator gets to “collect” the block reward only if the block ends up on long-term consensus branch!

Transaction Fees: Creator of transaction can choose to make output value less than input value. Remainder is a transaction fee and goes to block creator

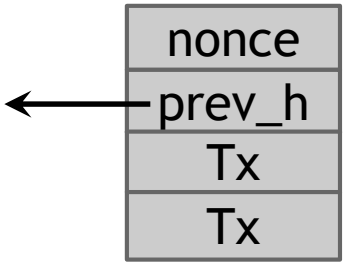
Proof of work

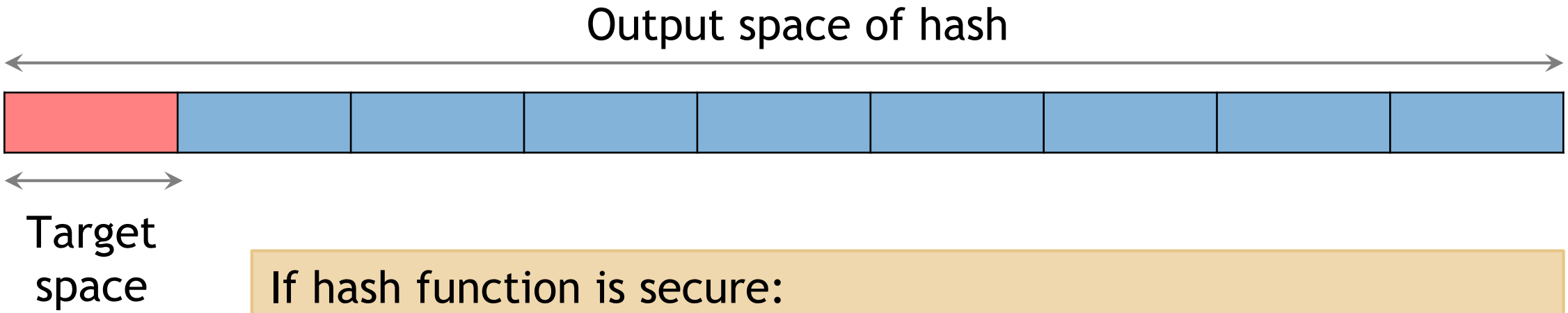
To approximate selecting a random node:
select nodes in proportion to a resource
that no one can monopolize (we hope)

- In proportion to computing power: proof-of-work
- In proportion to ownership: proof-of-stake

Hash puzzles

To create block, find nonce s.t.
H(nonce ‖ prev_hash ‖ tx ‖ … ‖ tx) is very small





If hash function is secure:
only way to succeed is to try enough nonces until you get lucky

Mining economics

If mining reward (block reward + Tx fees)	>	hardware + electricity cost	→	Profit

Complications:

- fixed vs. variable costs
- reward depends on global hash rate

A transaction-based ledger (Bitcoin)

time

1

Inputs: Ø

Outputs: 25.0→Alice

2

Inputs: 1[0]

Outputs: 17.0→Bob, 8.0→Alice

SIGNED(Alice)

change address

3

Inputs: 2[0]

Outputs: 8.0→Carol, 7.0→Bob

SIGNED(Bob)

4

Inputs: 2[1]

Outputs: 6.0→David, 2.0→Alice

SIGNED(Alice)

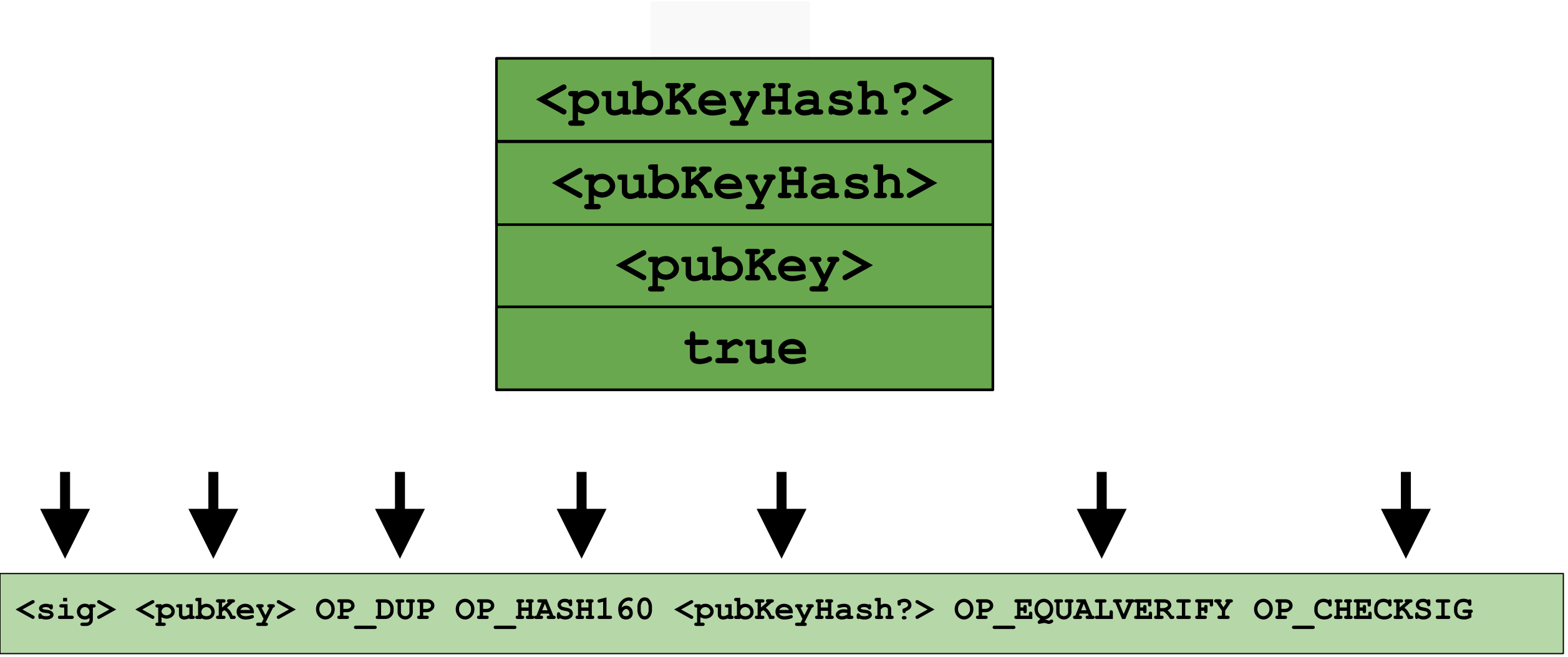
we implement this with hash pointers

finite scan to check for validity

is this valid?

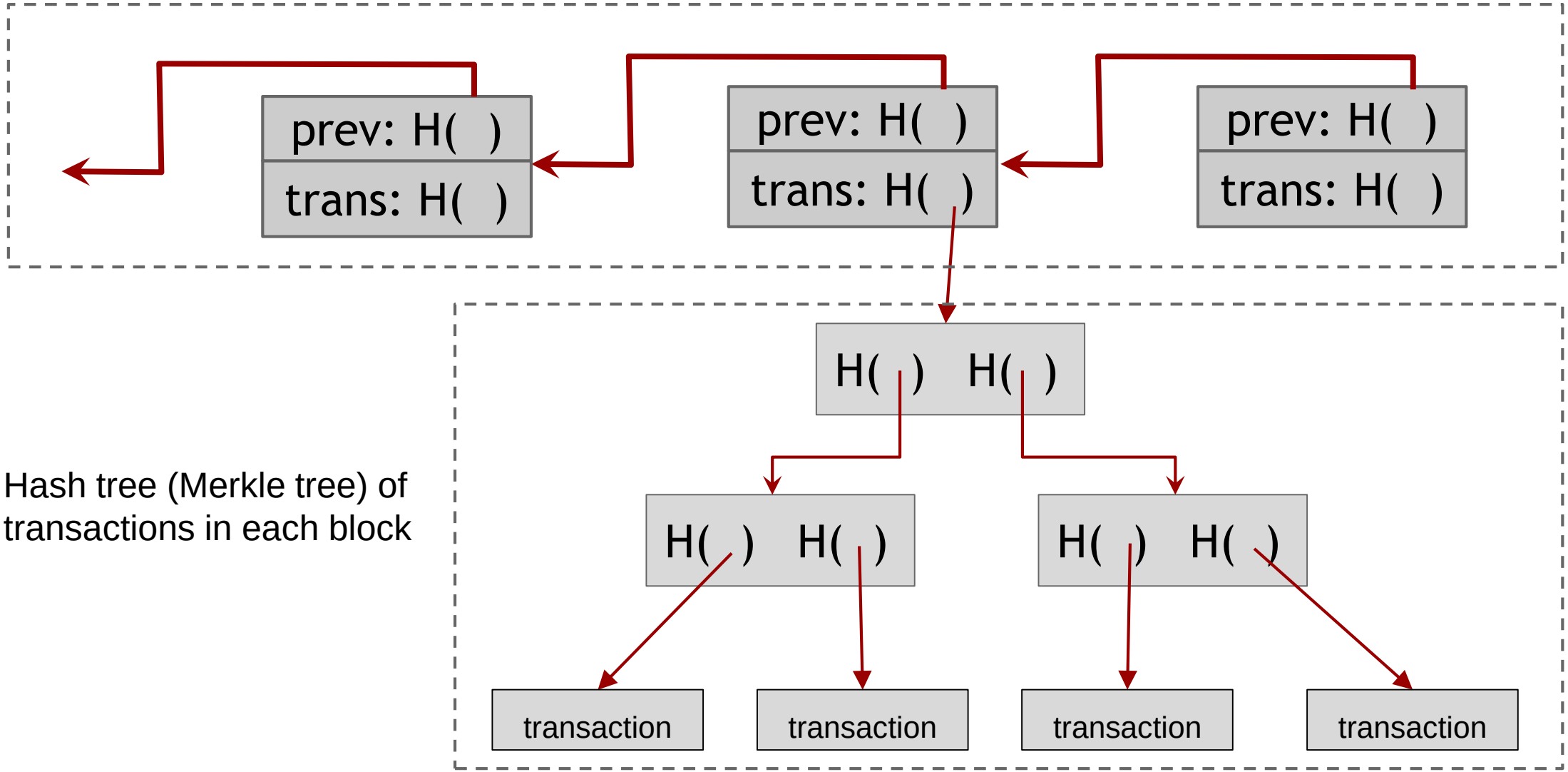
SIMPLIFICATION: only one transaction per block

Bitcoin script execution example



Bitcoin block structure

Hash chain of blocks



Hash tree (Merkle tree) of transactions in each block

<u>Storing Private Keys</u>: store key in a file,
on your computer or phone

Very convenient.
As available as your device.
device lost/wiped ⇒ key lost ⇒ coins

As secure as your device.
device compromised ⇒ key leaked ⇒ coins stolen


東京で MT.GOX のデモ
へ参加してください。
東京都渋谷区渋谷
２丁目１１−５
MTGOX
WHERE IS
OUR MONEY

Evolution of Bitcoin mining

CPU

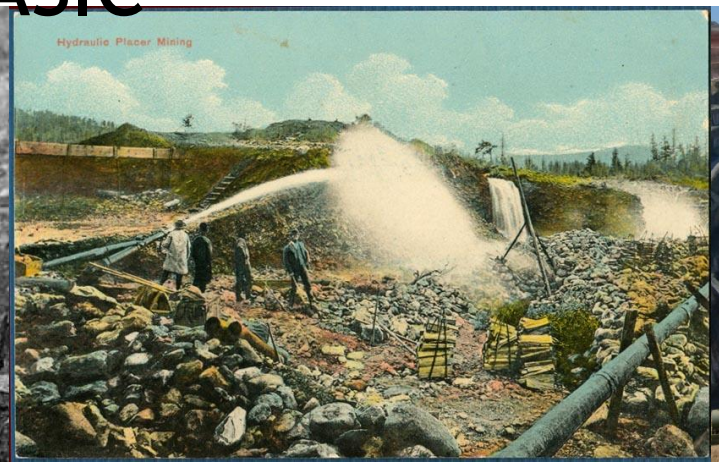
GPU

ASIC

FPGA

gold pan

sluice box



placer mining

pit mining

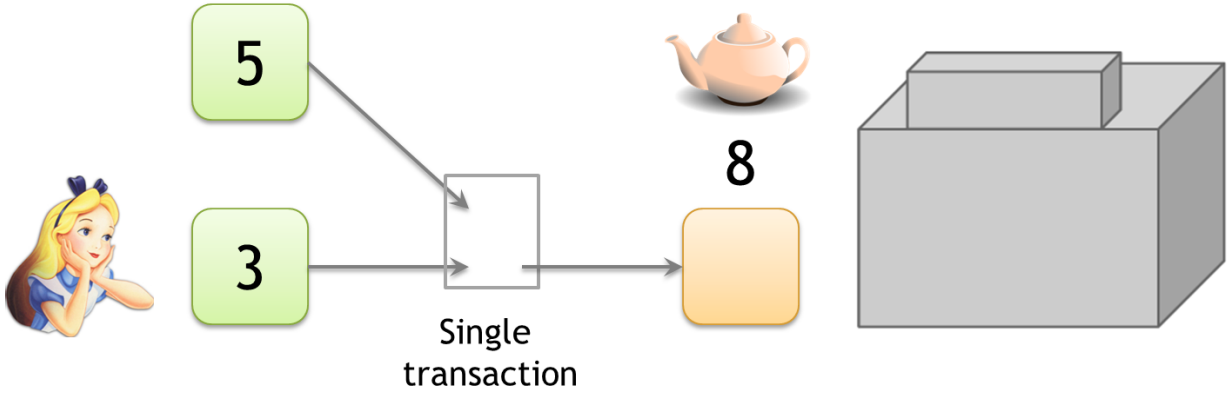
Professional mining centers

Needs:
- cheap power
- good network
- cool climate

BitFury mining center, Republic of Georgia

Identifying Addresses By Spending

Shared spending is evidence of joint control



Addresses can be linked transitively

Bitcoin links

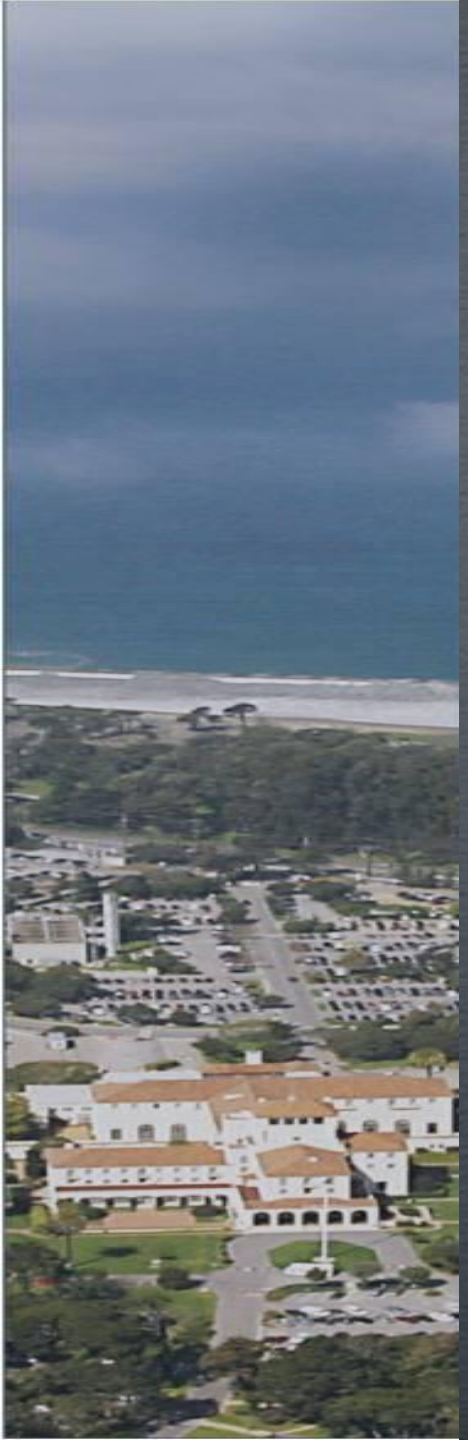
- https://bitcoin.org/bitcoin.pdf
- https://github.com/bitcoinbook/bitcoinbook/blob/develop/book.asciidoc
- https://p2sh.info/dashboard/db/home-dashboard?orgId=1
- https://github.com/petertodd/python-bitcoinlib
- https://en.bitcoin.it/wiki/Script
- https://bitinfocharts.com/top-100-richest-bitcoin-addresses.html
- http://cs251crypto.stanford.edu/18au-cs251/
- http://bitcoinbook.cs.princeton.edu/

Ethereum

- Solidity programming language: similar to Java/Javascript, with cryptocurrency functionality built in
- Smart Contracts are the Solidity equivalent of java classes that run on a blockchain in an Ethereum virtual machine
- Transaction costs in Solidity are called gas costs. Everything that executes on the blockchain has a gas cost associated with it.
- Security is very important. Solidity has greater capabilities than Bitcoin Script, and far greater security vulnerabilities.

Ethereum and Solidity links

- http://bit.do/cs251solidity
- https://remix.ethereum.org/
- https://coursetro.com/posts/code/97/Ethereum-Smart-Contracts:-Variables-and-Types-Tutorial



Identity and Access Management with Blockchain on GCSS-MC

Capt Brandan Schofield
Capt Brittany Snelgrove

Evolve the MAGTF, operate with resilience, and enhance the Marine Corps' maneuverability

- Marine Corps Operating Concept

WDKBAZ405026 Unclassified

Oracle Applications Home Page - Microsoft Internet Explorer provided by NMCI

https://gcssmc-ebs.csd.disa.mil/OA_HTML/OA.jsp?OAFunc=OAHOMEPAGE&akRegionApplicationId=0&navRespId=503998&navRespAppId=513&navSecGrpId=0&transactionid=695117465&oapc=2&oas=Z

File Edit View Favorites Tools Help

Oracle Applications Home Page

GCSS-MC GLOBAL LOGISTICS AT THE SPEED OF BATTLE ENTERPRISE NIPRNET

*** For Official Use Only ***

Logout Help

Logged In As JOSEPH.SCHNEIDER

Worklist

From	Subject
	Standard Document Number: M2154010870001 Failed Internal Funds Reservation

TIP Vacation Rules - Redirect or auto-respond to notifications.

TIP Worklist Access - Specify which users can view and act upon your notifications.

Favorites

Edit Favorites

You have not selected any favorites. Please use the "Edit Favorites" button to set up your favorites.

Navigator

- GCSS-MC DBI Customer Support Dashboard User
- GCSS-MC DBI Depot Repair Dashboard User
- GCSS-MC DBI Field Service Dashboard User
- GCSS-MC Discoverer Reports User
- GCSS-MC Discoverer Reports Writer
- GCSS-MC Financial Inquirer
- GCSS-MC iSupport Requestor
- GCSS-MC Maint Shipping & Receiving SNCO
- GCSS-MC Maintenance Chief
- GCSS-MC Maintenance Management Officer / Chief
- GCSS-MC Maintenance Quality Control Chief
- GCSS-MC Maintenance Shipping & Receiving NCO
- GCSS-MC Mechanic / Technician
- GCSS-MC Mobile Field Service Administrator
- GCSS-MC Order Manager
- GCSS-MC Resource Group Setup
- GCSS-MC User Management
- Oracle Installed Base User
- Preferences SSWA

GCSS-MC Resource Group

- Calendar Setup
- Resource Addresses and

Maintain Resources

- Import Resources
- Resources
- Groups

Territory Management

- Territory Administration

Loading

Logout | Help

Global Combat Support System - US Marines

Privacy Statement

Done Internet 100%

start Microsoft PowerPoint ... GCSS-MC Portal - Mic... Oracle Applications H... 13:57

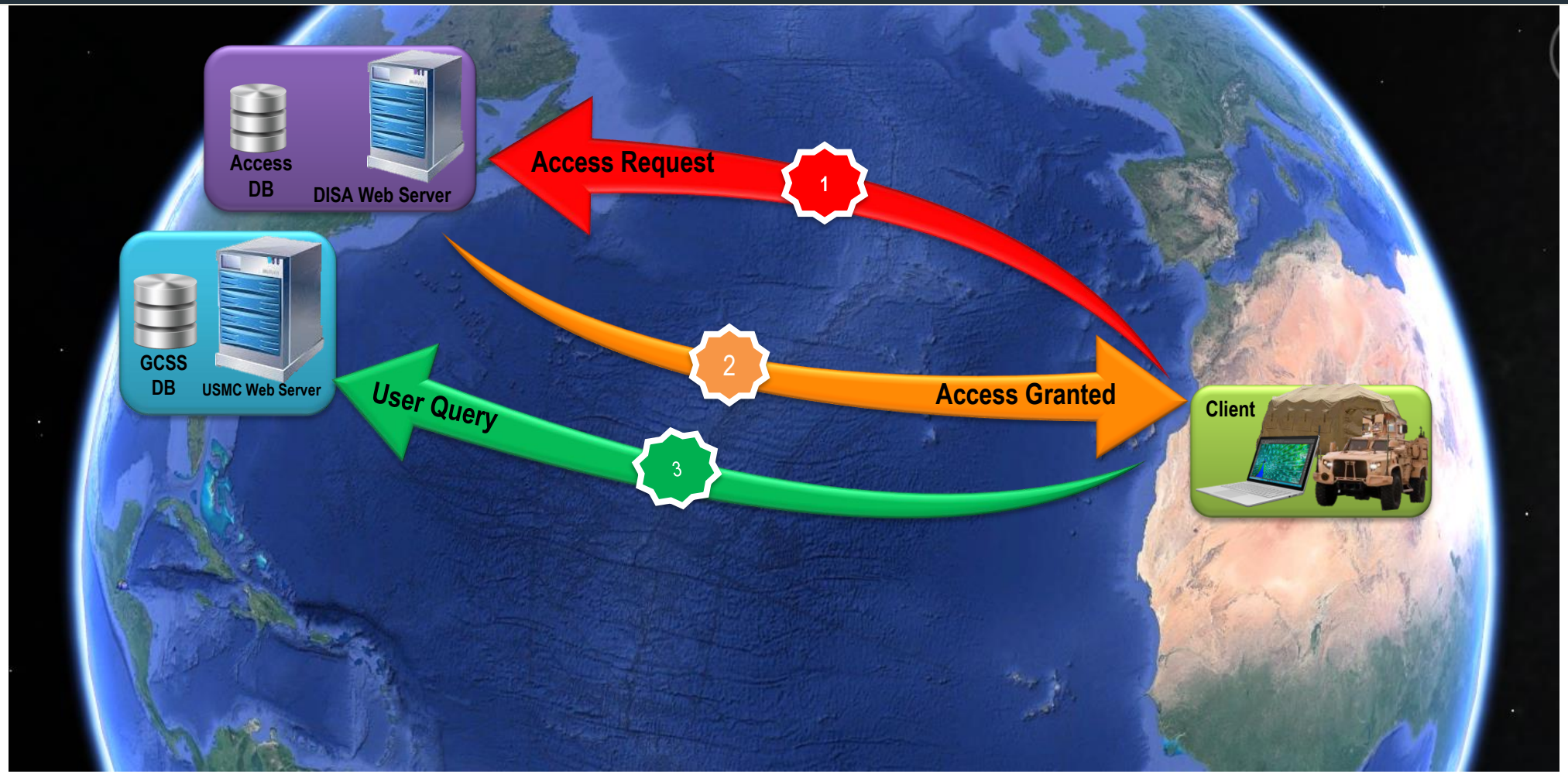
Agenda

- What is GCSS-MC?
- Current architecture
- Research Questions
- What is blockchain?
- Proposed architecture
- Benefits
- Timeline
- Questions

What is GCSS-MC?

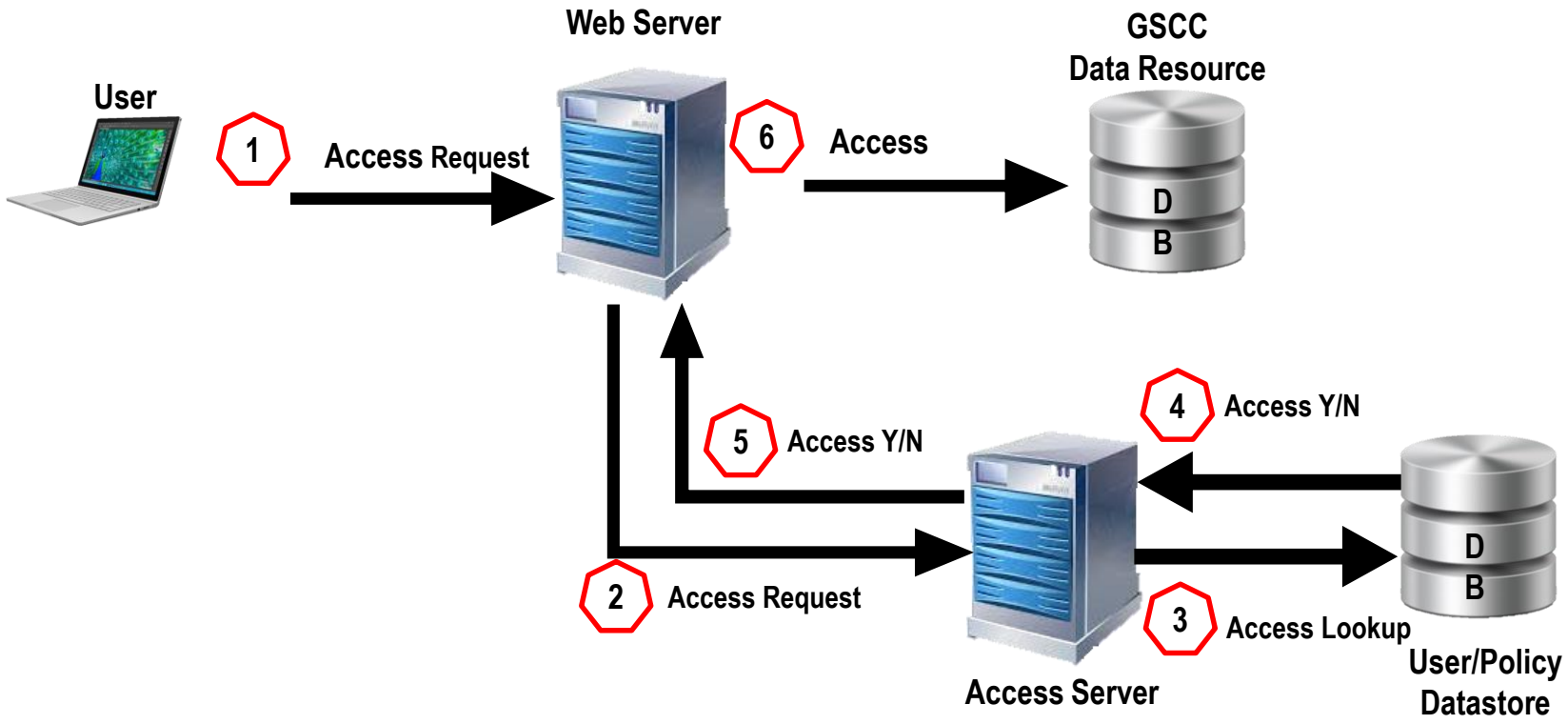
- USMC's Supply and Maintenance Management Web-enabled Data-resource
- Used to manage, control, identify and distribute ground supplies and coordinate maintenance actions for all ground Marine units
- Software – Oracle E-Business Suite (EBS) version R12
- 3 Tier System
  - Database Tier
  - Application Tier
  - Client Tier
- Requires Internet Connection to function
- Access Management
  - Oracle Access Management (OAM) using Online Certificate Status Protocol (OCSP) part of PKI terminating at CONUS based DISA Servers

Current Architecture



<u>Problem</u>: 67% of user transaction time devoted to DISA check
13.5% of network overhead (Mbps)

Current Physical Architecture

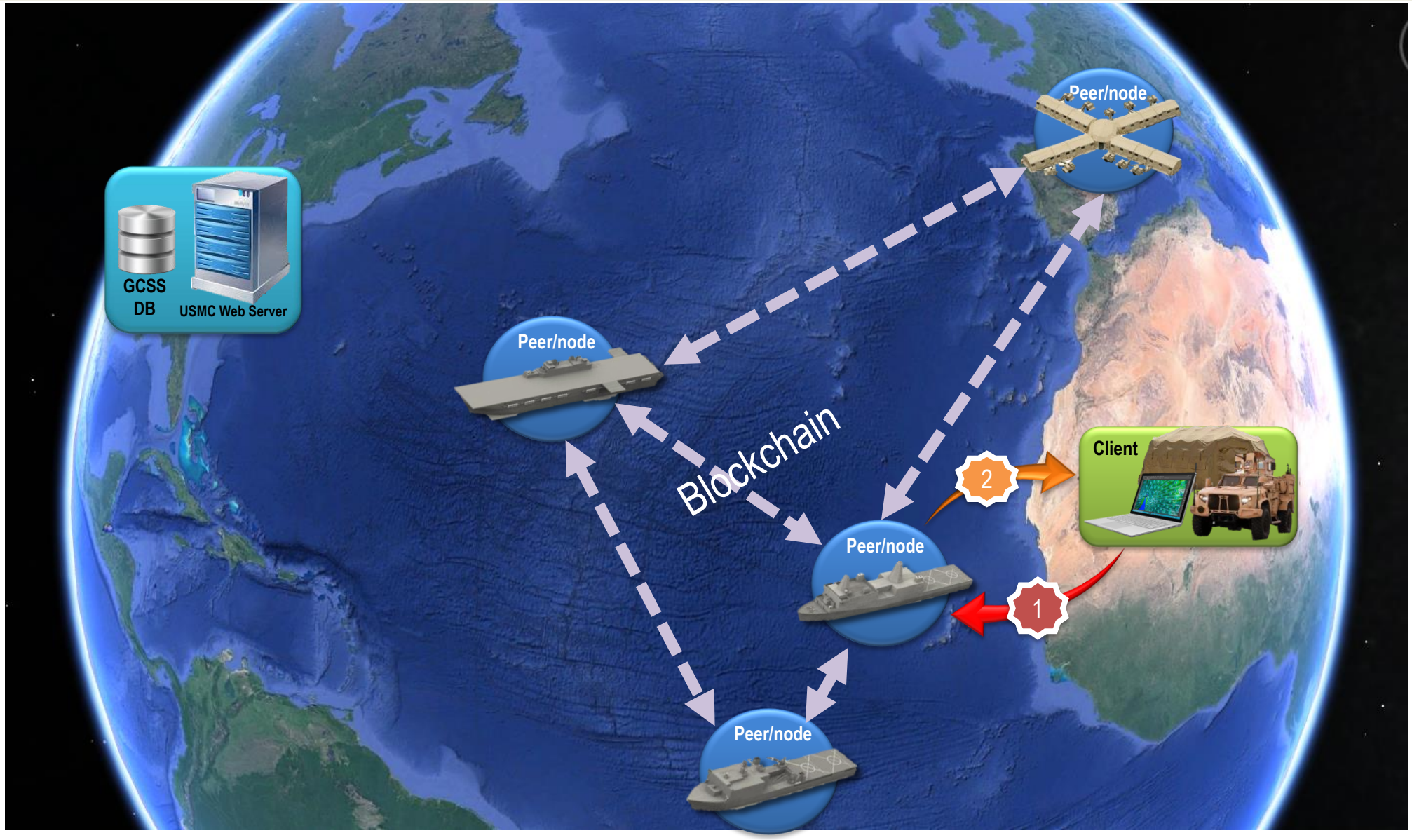


Typical Oracle Access Management

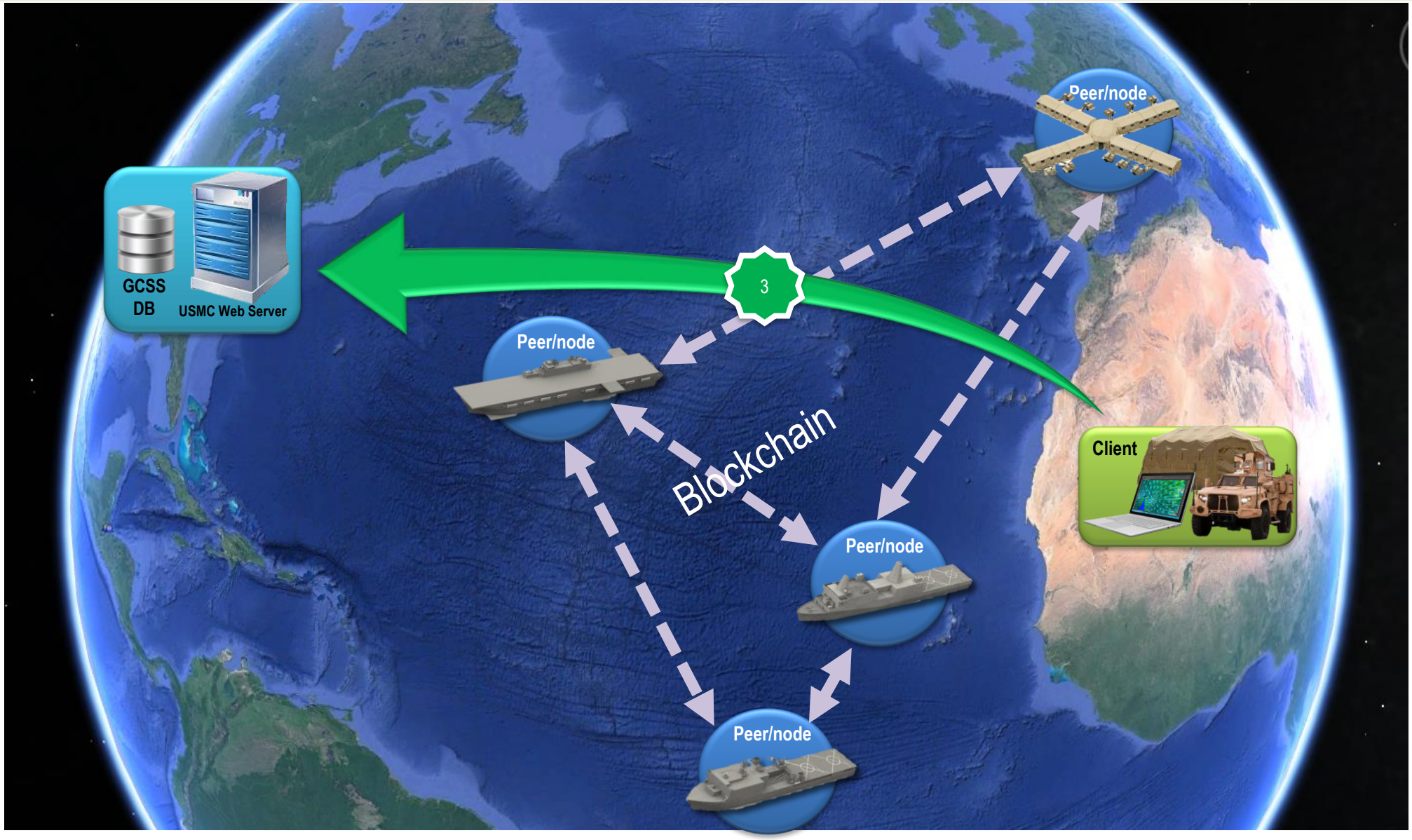
Research Questions

1. How can a blockchain database be used to authenticate clients on the GCSS-MC web-enabled data resource? (Experimentation)
2. How can a blockchain database be feasibly acquisitioned and integrated into the current GCSS-MC architecture? (Qualitative)

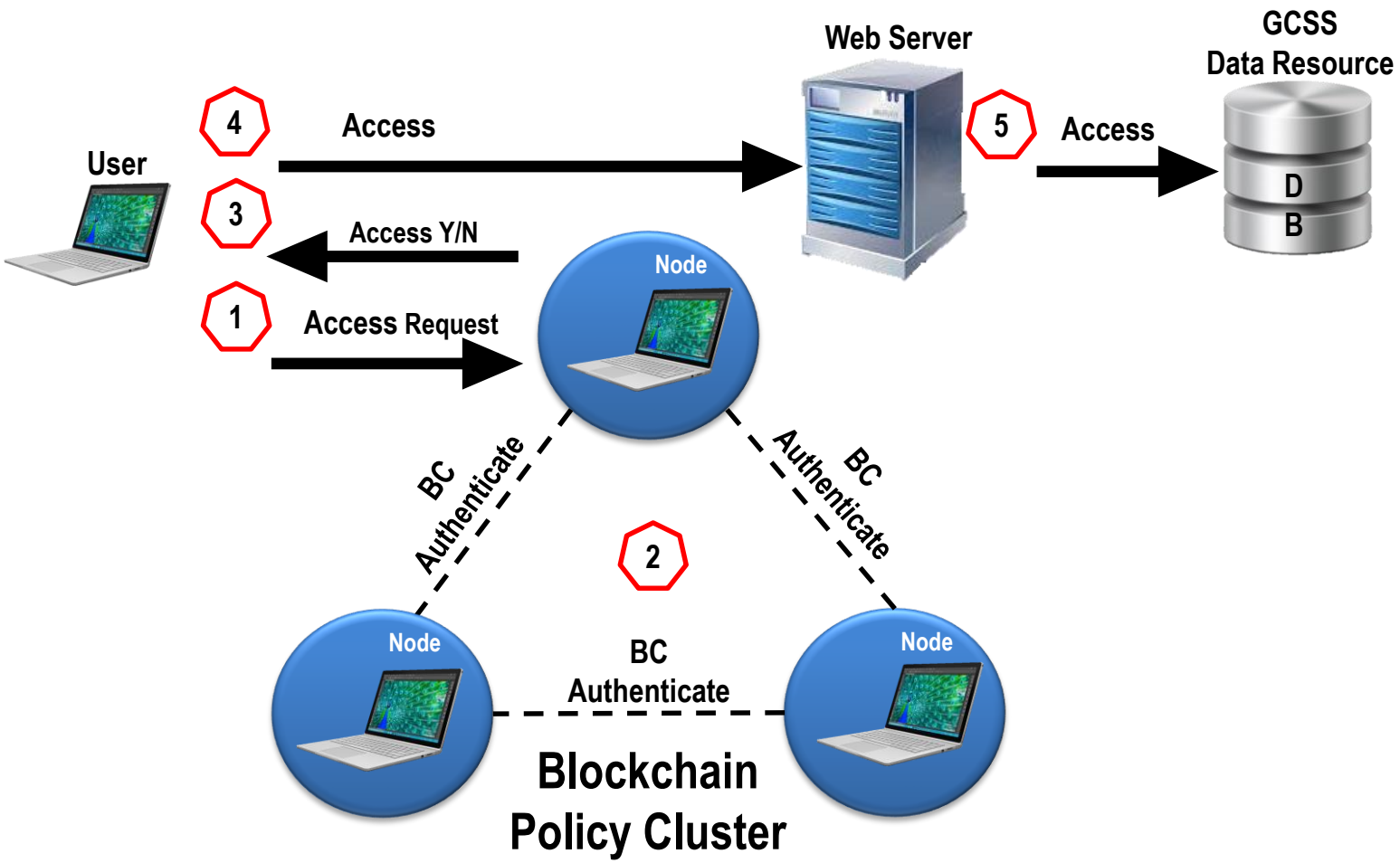
Proposed Architecture



Proposed Architecture



Proposed Logical Architecture



Blockchain Access Management

Benefits

- Decentralized user authentication
- Network overhead potentially reduced
- No DMZ required (trustless system)
- No expensive centralized web-servers and data-stores
- Potential increase in availability for remote users
- Policy enforcement through algorithm
  - *“Trust through algorithm”*

Questions


Communicate
Maneuver
Identity
Shoot
Logistics
Network
Access
Maneuver
Blockchain
Move
Bandwidth
Target
Target
Access and Identity
Sustainment
Sustainment
Security
Management
Information
Identity
Cyber
Communicate
Target
Move
Security
Shoot
Cyber
Logistics
Move
Communicate

Bitcoin Lab Demo #1

Ethereum Lab Demo #1