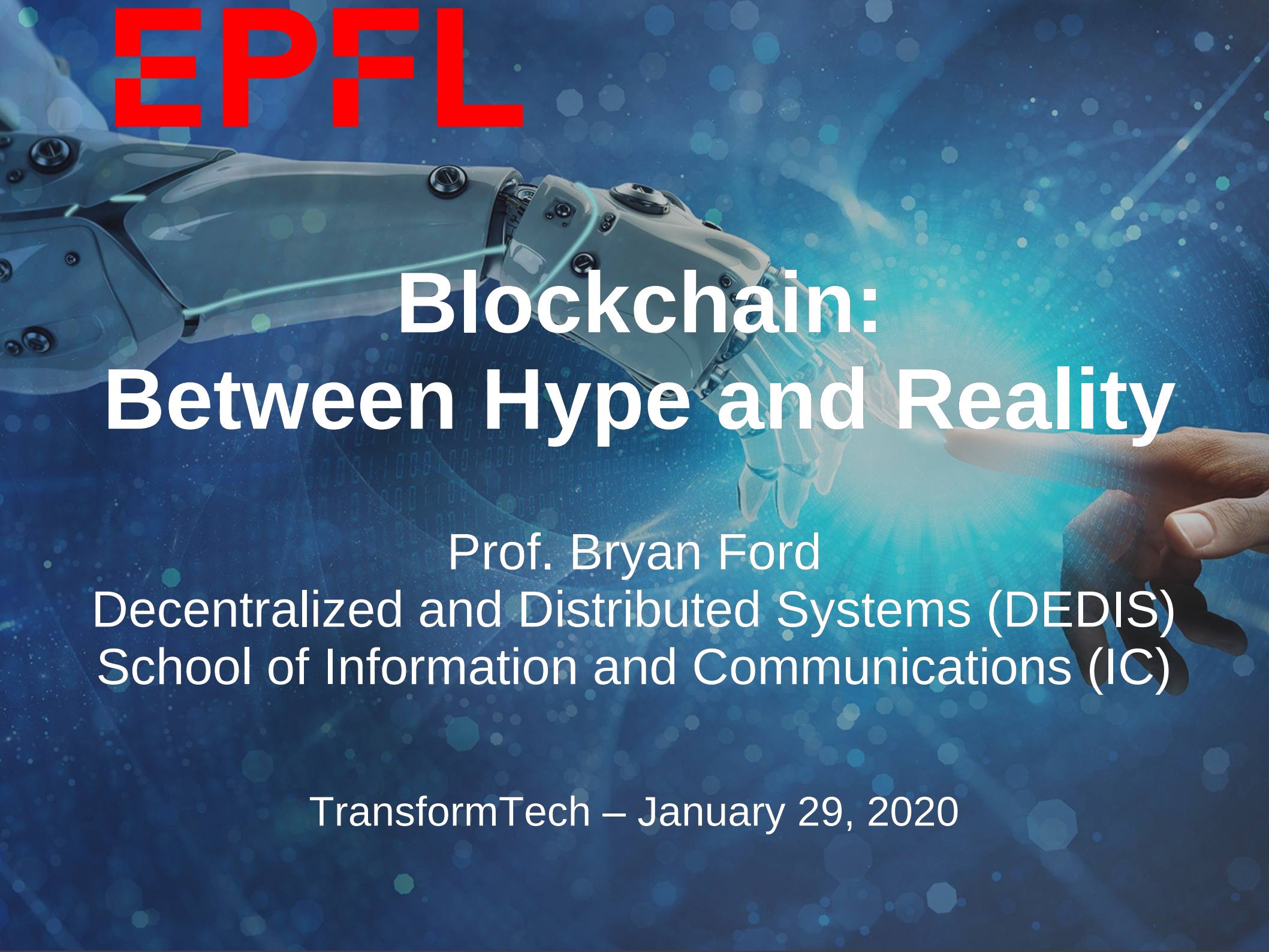


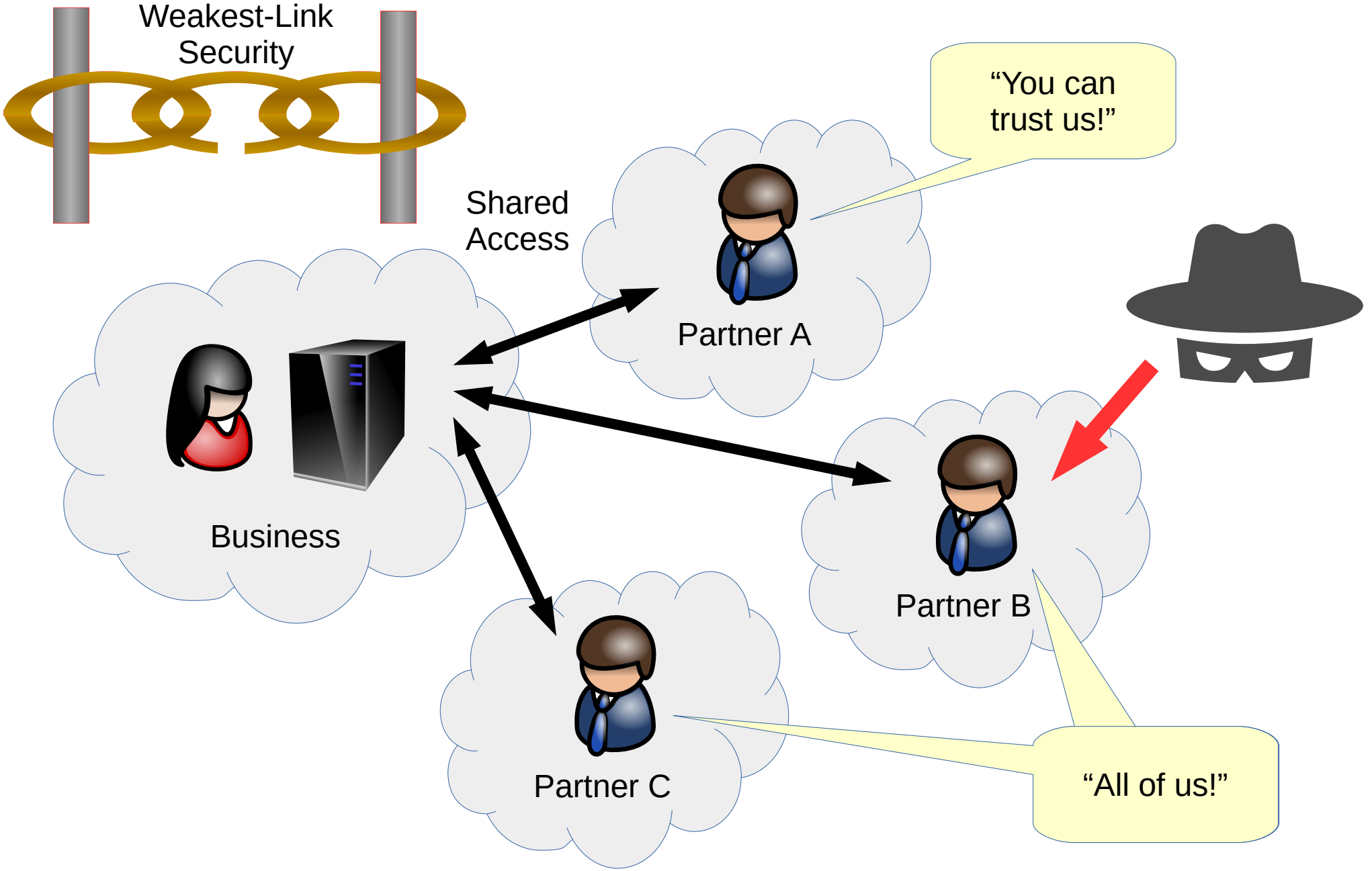
Blockchain: Between Hype and Reality

Prof. Bryan Ford
Decentralized and Distributed Systems (DEDIS)
School of Information and Communications (IC)

TransformTech – January 29, 2020

Where there's data, there's risk...

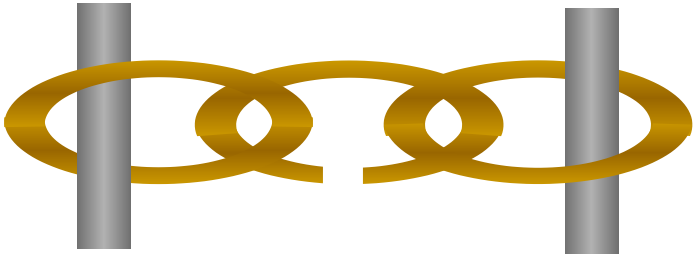
Access, sharing compounds risk



A Fundamental Challenge

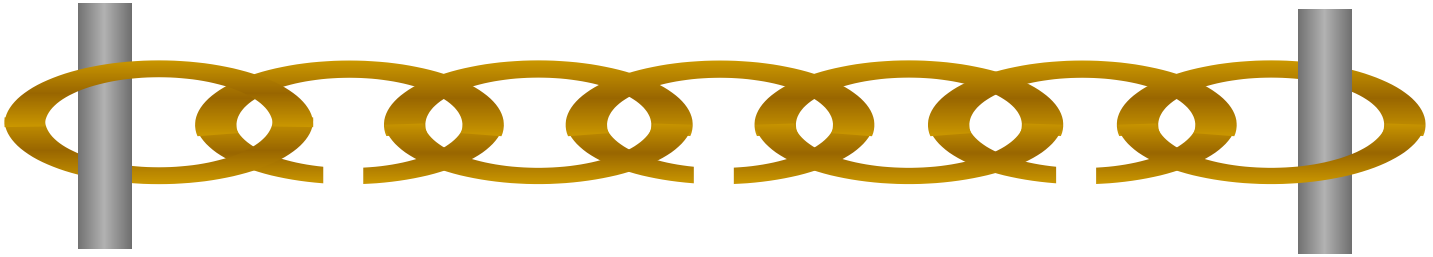
In today's IT systems, security is an afterthought

- Designs embody "weakest-link" security

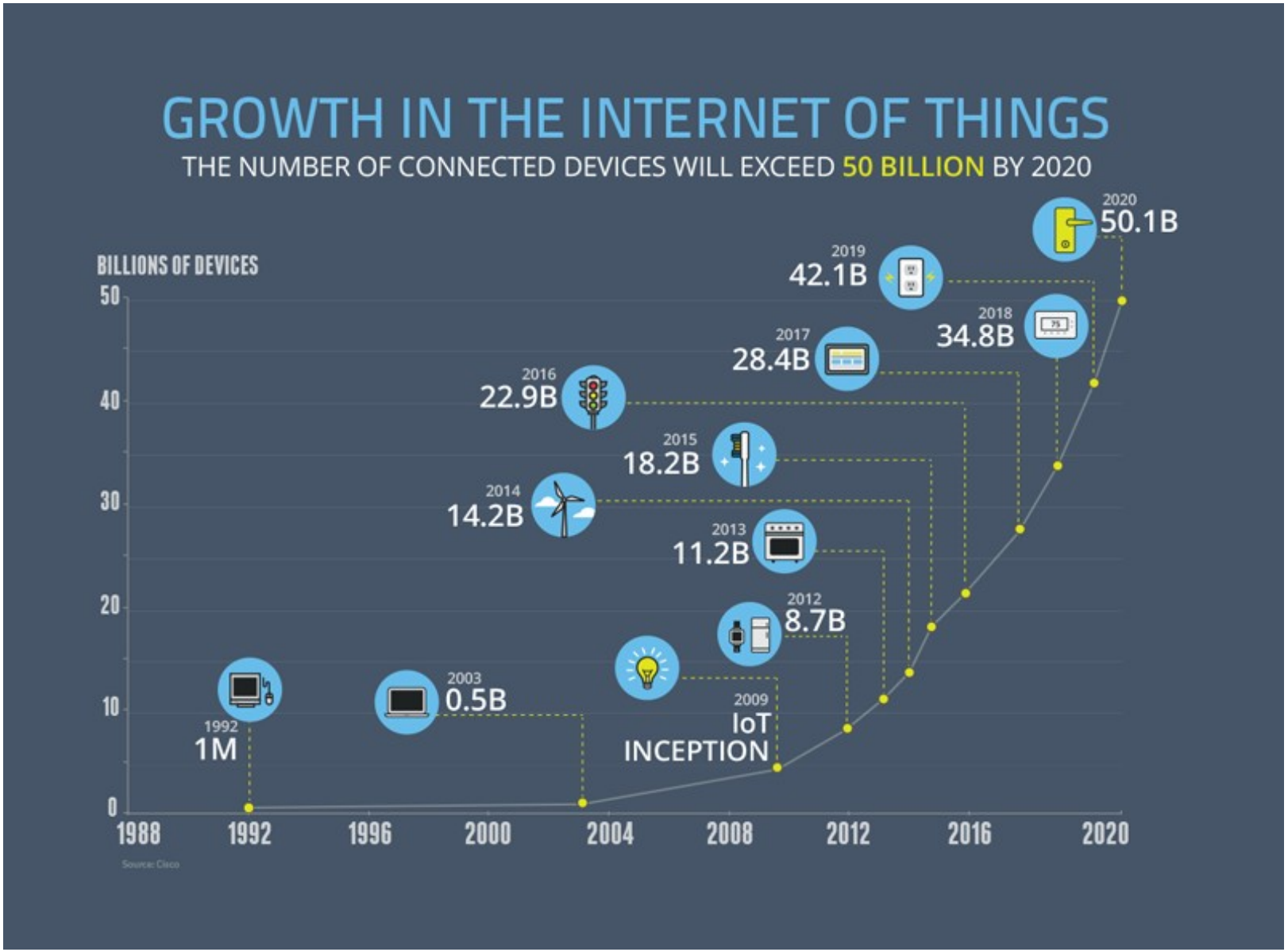


Scaling to bigger systems → weaker security

- Greater chance of any "weak link" breaking

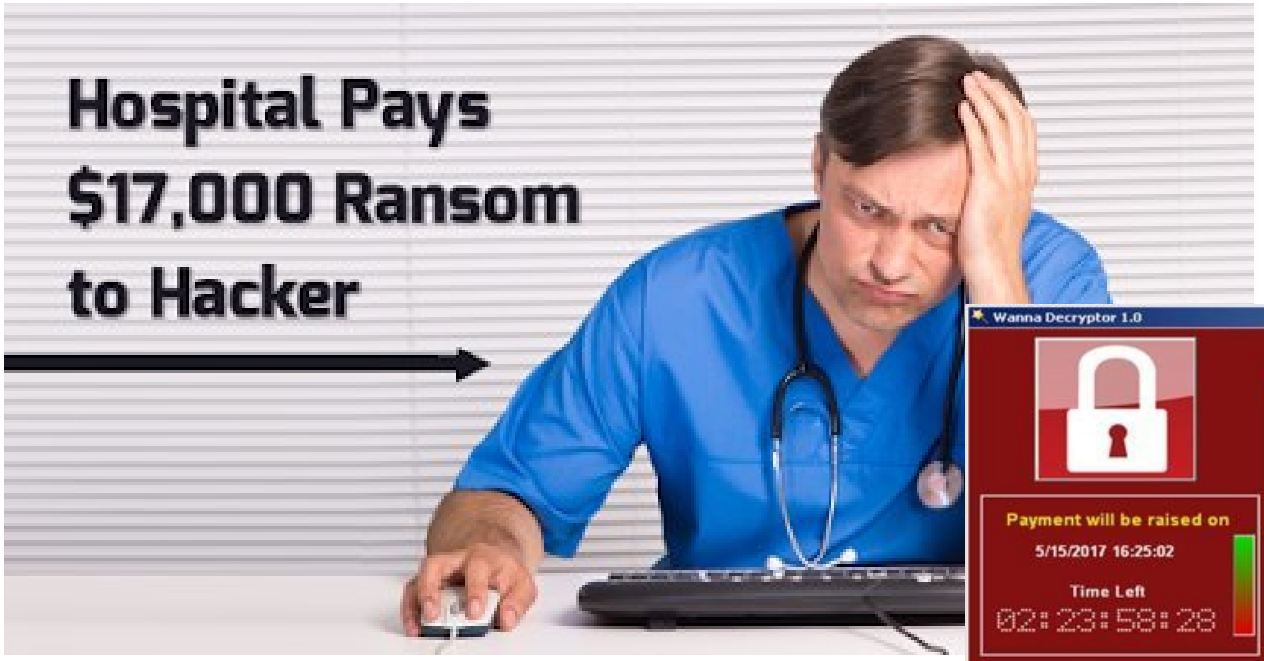


More Devices → More Weak Links



Critical Devices = Attractive Targets

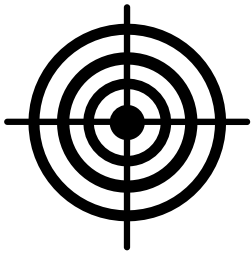
Repeated hospital ransomware attacks



[Nextgov, 23-June-2015]

[WannaCry ransomware, May 2017]

Central Databases = Attractive Targets



One of three credit rating agencies in the US

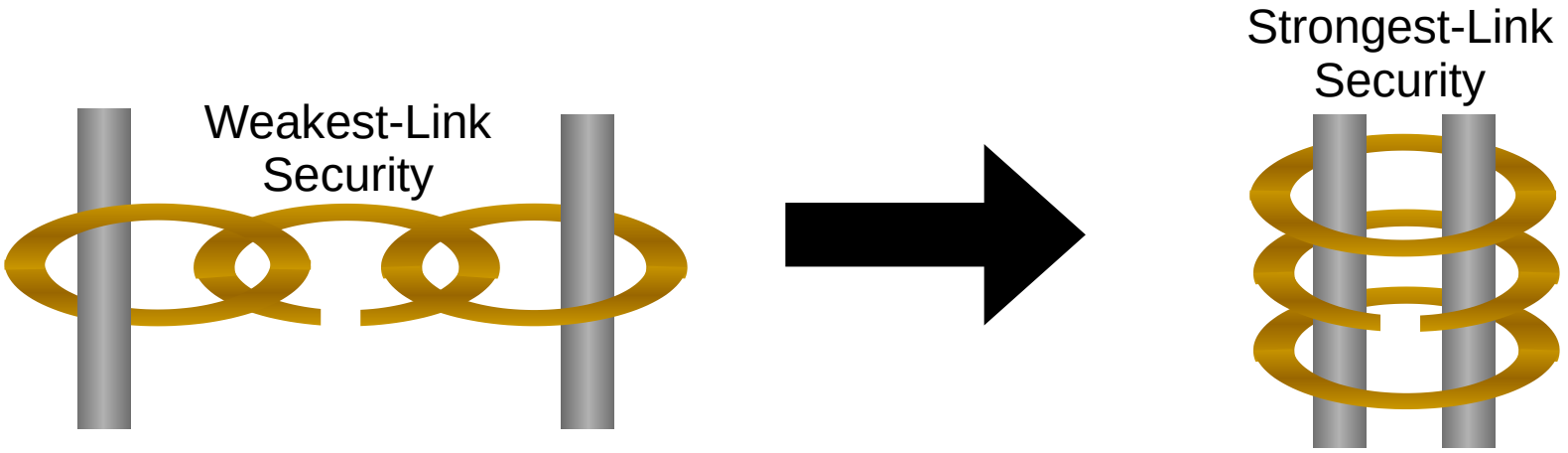
- Exposed sensitive personal information about 143 million people (44% of US population)

The DEDIS lab at EPFL: Mission

Design, build, and deploy secure privacy-preserving
Decentralized and Distributed Systems (DEDIS)

- **Distributed:** spread widely across the Internet & world
- **Decentralized:** independent participants, no central authority, *no single points of failure or compromise*

Overarching theme: building decentralized systems
that **distribute trust** widely with **strongest-link security**



Turning Around the Security Game

Design IT systems so that making them bigger makes their security *increase* instead of *decrease*

Weakest-link security

Strongest-link security

Scalable Strongest-link security

DEDIS Laboratory Members

Bryan Ford
Associate Professor

Philipp Jovanovic
Postdoctoral Scholar

Lefteris Kokoris-Kogias
Postdoctoral Scholar

Henry Corrigan-Gibbs
Postdoctoral Scholar

Kirill Nikitin
Ph.D. Student

Cristina Basescu
Ph.D. Student

Enis Ceyhun Alp
Ph.D. Student

Jeff R. Allen
Software Engineer

Gaylor Bosson
Software Engineer

Noémien Kocher
Software Engineer

Gaurav Narula
Software Engineer

Today's Hot Decentralized Technology

(credit: Tony Arcieri)

Lecture Outline

- Introduction: What is a Blockchain?
- Applications: What are Blockchains Good For?
- Smart Contracts: Can Blockchains Compute?
- Consensus: How do Blockchains Coordinate?
- Privacy: Can Blockchains Keep Secrets?
- Wrap-up: Promise and Challenges

Lecture Outline

- **Introduction: What is a Blockchain?**
- Applications: What are Blockchains Good For?
- Smart Contracts: Can Blockchains Compute?
- Consensus: How do Blockchains Coordinate?
- Privacy: Can Blockchains Keep Secrets?
- Wrap-up: Promise and Challenges

Structurally,
a blockchain is just a **log** or **ledger**
recording events that happen over time.

BANKING LEDGER

Account Number:

DATE	DESCRIPTION	DEPOSIT	WITHDRAW	BALANCE

VEHICLE REGISTRATION CONTROL LOG

1			2	3	4	5	6	7	8	9	10	11	12	13
REPORT-OF-SALE NUMBER	NO. STOCK	DEAL NO.	CUSTOMER NAME	DATE OF SALE	DATE DEALER NOTICE SENT DMV	DATE WORK SENT DMV	$ AMT OF LICENSE FEES CHARGED CUSTOMER	$ AMT. OF DMV RATING OF FEES	$ AMT DIFFERENCE (plus or minus) (Cols. 6-7)	DATE AND NO. OF REFUND CHECK AND NOTATIONS	DATE OF RDF	DATE RESUB-MITTED	LIST NUMBER	REASON FOR RDF

Like a log book,
a blockchain can potentially record
any type of event.

ALIEN ABDUCTION LEDGER

DATE	DESCRIPTION

But the *motivation* for recording these events often has to do with **security**.

VISITOR LOG

DATE	TIME IN	TIME OUT	NAME	AFFILIATION

Why We Record Things in Logs

1. **Transparency:** more people have a better chance at noticing something wrong earlier.

2. **Accountability:** so if something goes wrong, there's a way to figure out whose head rolls.

But is this documentation **Trustworthy**?

Key Security Challenge

Single points of **failure** or **compromise**.

What if the records get **burned in a fire** -
either accidentally or “accidentally”?

If thief, criminal, fraudster, spy, etc.,
can compromise the **master copy of the log**,
then he can just **change the log**
to cover his tracks

Solution: Redundant Records

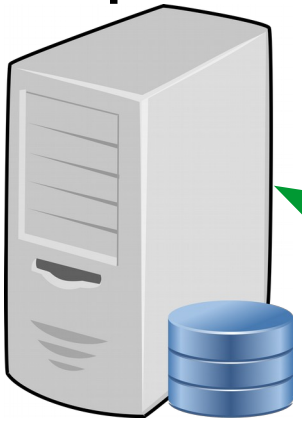
Keep **several independent copies** of records.

Keep them all **synchronized** (consistent).

Hope it’s hard for attacker to get **all of them**.

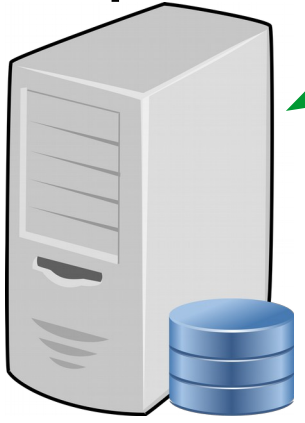
Replication for Security

Remote Replica



Cloud Replica

Local Replica



Local Replica

Replication alone isn’t enough

What if there’s one **central security office** from where **all the copies** can be controlled, and attacker gets access to that office?

Replication alone isn’t enough

The **human weak link** problem:
what if one administrator can control **all** copies,
and administrator [account] is compromised?

The **Real (But Difficult)** Solution

Keep **several independent copies** of the log.

Keep them all **synchronized** (consistent).

Make sure they are as **independent** as possible.

Replication
+
Synchronization
+
Independence

=

Decentralized Trust,
the principle underlying **Blockchain**

Bitcoin (2008)

First successful decentralized cryptocurrency…

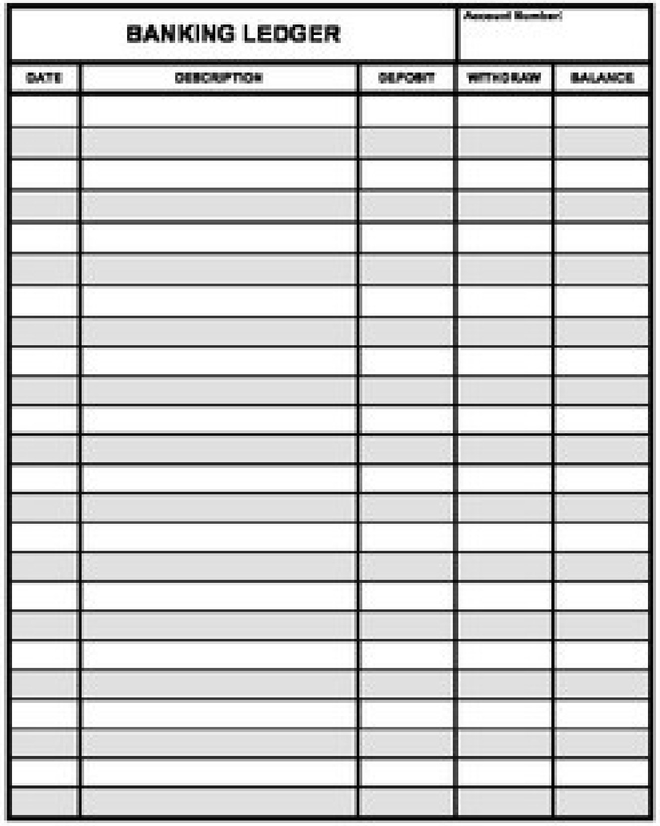
How to track wealth (or anything)?

Things

- Gold, beads, cash...

Ledgers

- Who owns what?



Precedent: the Rai Stones of Yap

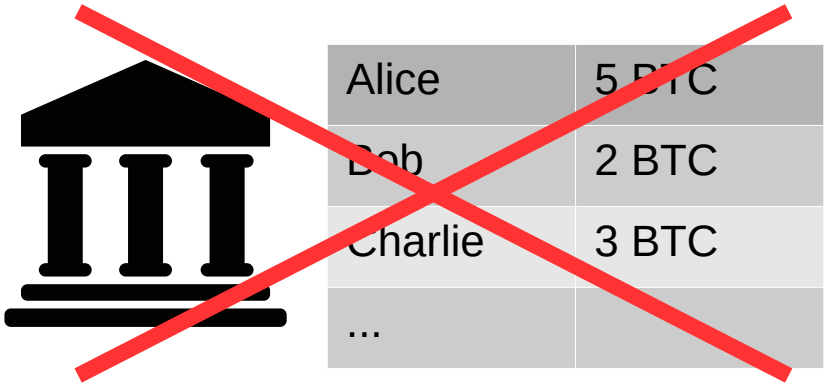
Stone “coins” weighing thousands of kilograms

- Left in place once created (“mined”)
- Ownership transfer by *public proclamation*

(this comparison shamelessly borrowed from Gün Sirer and others)

Distributed Ledgers

Problem: we don't want to trust any designated, centralized authority to maintain the ledger



Alice	5 BTC
Bob	2 BTC
Charlie	3 BTC
...	

Solution: "everyone" keeps a copy of the ledger!

- Everyone checks everyone else's changes to it

Alice's copy

Alice	5 BTC
Bob	2 BTC
Charlie	3 BTC
...	

Bob's copy

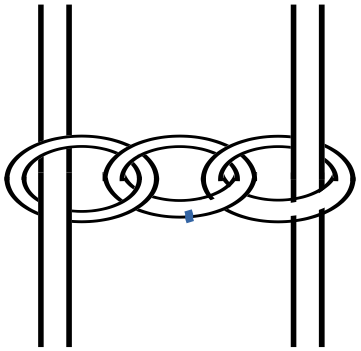
Alice	5 BTC
Bob	2 BTC
Charlie	3 BTC
...	

Charlie's copy

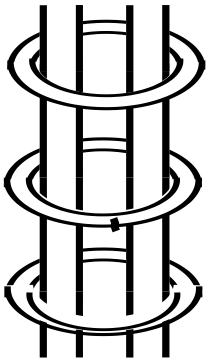
Alice	5 BTC
Bob	2 BTC
Charlie	3 BTC
...	

Properly-Designed Blockchains Eliminate Single Points of Compromise

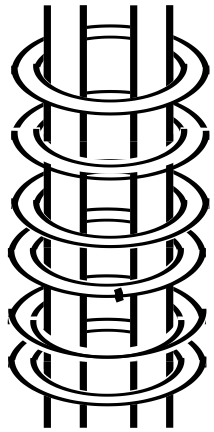
Weakest-link Security:
T = 1



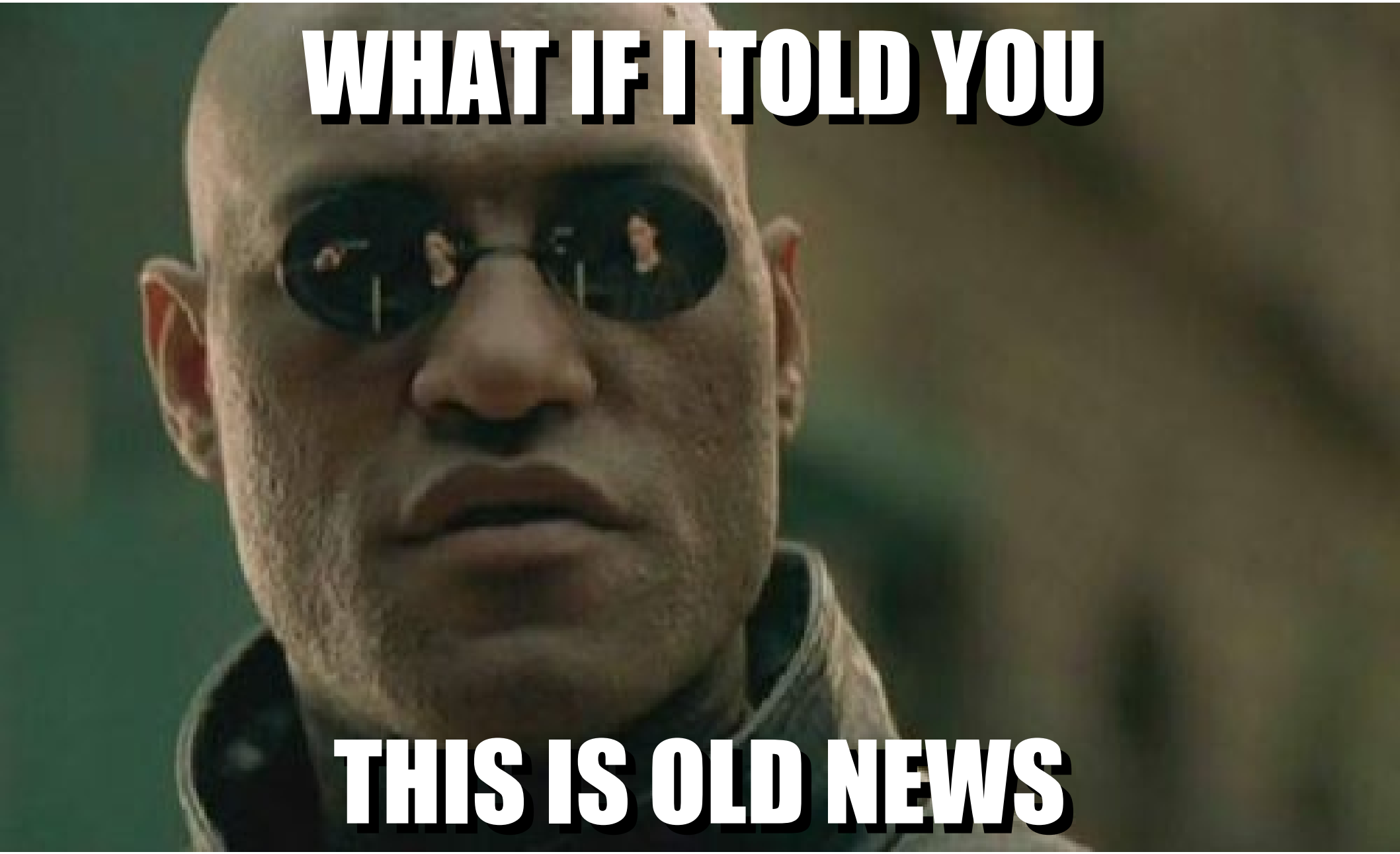
Strongest-link Security:
T = 2-10



Collective Security:
T = 100s,1000s



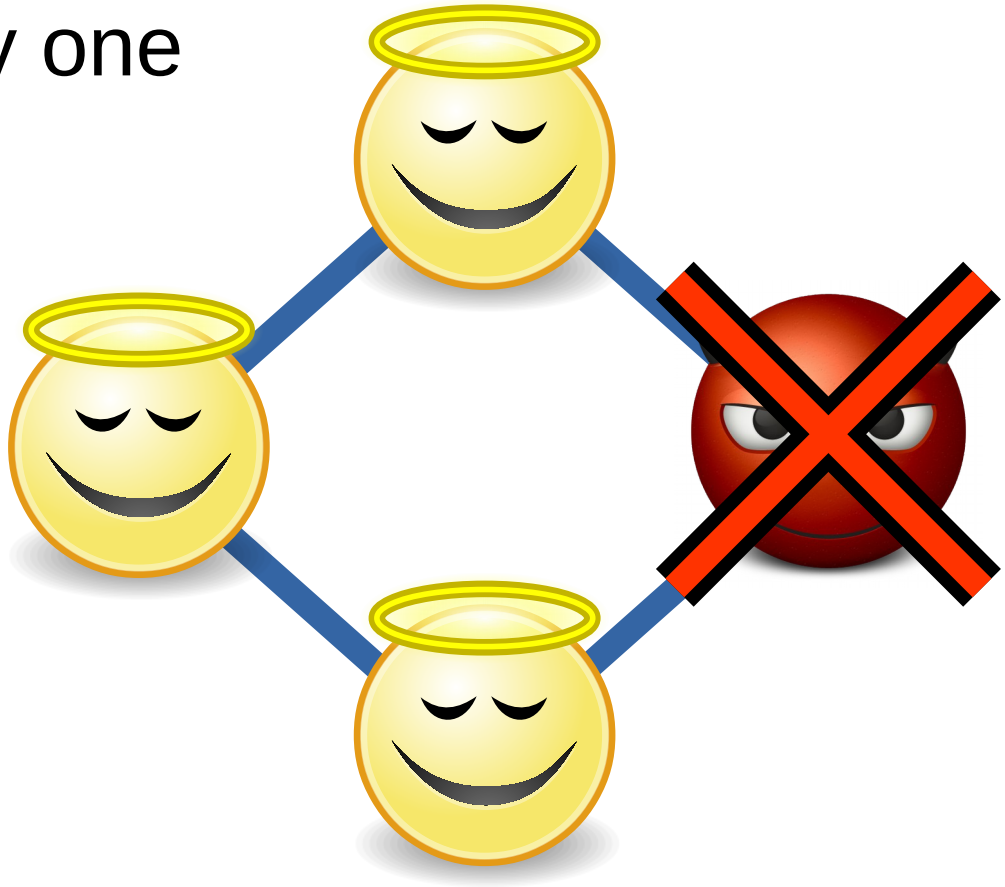
T: threshold of compromised parties to break security


WHAT IF I TOLD YOU
THIS IS OLD NEWS

Distributed Trust is Old News

Many algorithms allow us to **distribute trust** among multiple (preferably independent) parties

Work correctly despite any one (or several) participants being compromised, maliciously colluding



Example algorithms:

- Byzantine consensus
- Threshold cryptography (signing, encryption, …)

Distributed Trust is Old News

Many algorithms allow us to **distribute trust** among multiple (preferably independent) parties

Work correctly despite any one (or several) participants being compromised, maliciously colluding

Example algorithms:

- Byzantine consensus
- Threshold cryptography (signing, encryption, …)

So What's New?

They're not just
obscure computer science algorithms
anymore.

The rest of the world is (finally) interested.

But Beware the Lemon Market

George A. Akerlof won Nobel Prize in economics for observing:

> *If buyers have less information than sellers about product quality, incentives lead to reduced quality*

The cybersecurity market is a lemon market…

Schneier on Security

Blog | Newsletter | Books | Essays | News | Talks | Academic | About Me

Blog >

A Security Market for Lemons

More than a year ago, I wrote about the increasing risks of data loss because more and more data fits in smaller and smaller packages. Today I use a 4-GB USB memory stick for backup while I am traveling. I like the convenience, but if I lose the tiny thing I risk all my data.

The Blockchain Lemon Market

Today's blockchain market is too. ☹

Economically-leading "first-to-market" designs completely compromise decentralized security

- One-click "Blockchain-as-a-Service" on cloud
- Non-Byzantine consensus in deployment
- Centralized PKI in permissioned blockchains

Lecture Outline

- Introduction: What is a Blockchain?
- **Applications: What are Blockchains Good For?**
- Smart Contracts: Can Blockchains Compute?
- Consensus: How do Blockchains Coordinate?
- Privacy: Can Blockchains Keep Secrets?
- Wrap-up: Promise and Challenges

When Can You Use a Blockchain?

Could be used anywhere a database is useful!

- Whenever you need to keep records, *i.e., practically any process of modern civilization*
- A highly general technology in principle, hence the hype

"Everything looks like a nail."

When Do You **Need** a Blockchain?

Only when *several participants* have interests and don’t want to trust *any one* completely

- To avoid single points of failure or compromise

If you *do* have a single partner everyone trusts, a standard database server is faster & cheaper!

- Amazon has plenty if you don’t want your own

Applications of Distributed Ledgers

Can represent a distributed electronic record of:

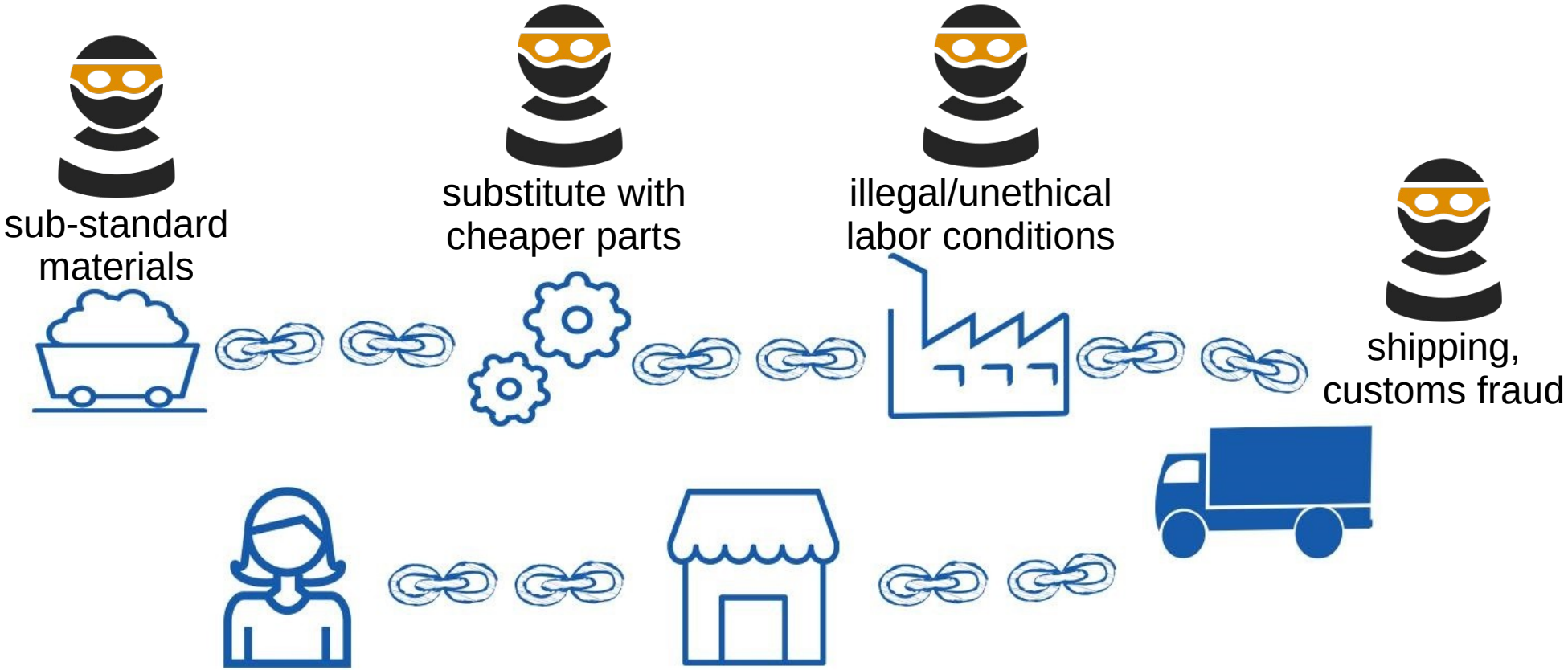
- Who owns how much **currency**? (Bitcoin) 
- Who owns a name or a digital work of art? 
- What are the terms of a **contract**? (Ethereum) 
- When was a **document** written? (notaries) 

- What is the **provenance** of a part? (supply chain)
- Who **are** you? (self-sovereign identity)
- Who used **data** for what purpose? (access logs)
- …

Application Example: Supply Chain

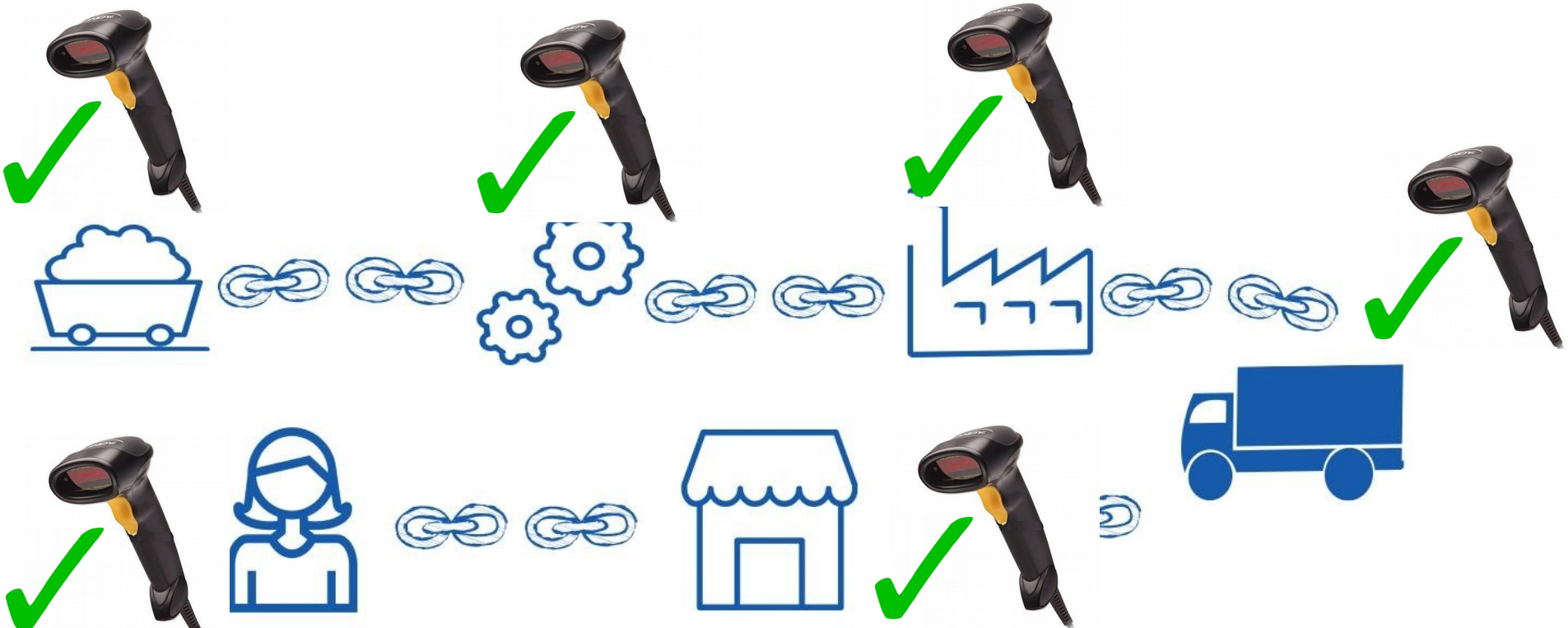
Consumers, manufacturers depend on complex supply chains of materials, parts, labor, shipping

- All links in chain are vulnerable to compromise



The [Potential] Promise

With the right automated & semi-automated scanning & tracking processes throughout, feeding a **common distributed log** or **ledger**
→ make substitution, fraud, etc., much harder



Supply Chain Grand Challenge

Can we make complex supply chains
transparent, **manageable**, and **accountable**
“end to end” from raw materials to consumer…

while still enabling companies
to maintain the **confidentiality** they need
to compete effectively?

Maybe, with the right distributed trust architecture

Lecture Outline

- Introduction: What is a Blockchain?
- Applications: What are Blockchains Good For?
- **Smart Contracts: Can Blockchains Compute?**
- Consensus: How do Blockchains Coordinate?
- Privacy: Can Blockchains Keep Secrets?
- Wrap-up: Promise and Challenges

Smart Contracts: Blockchain Code

Data (tokens, etc) stored on blockchain with **Code** (smart contract) controlling its use

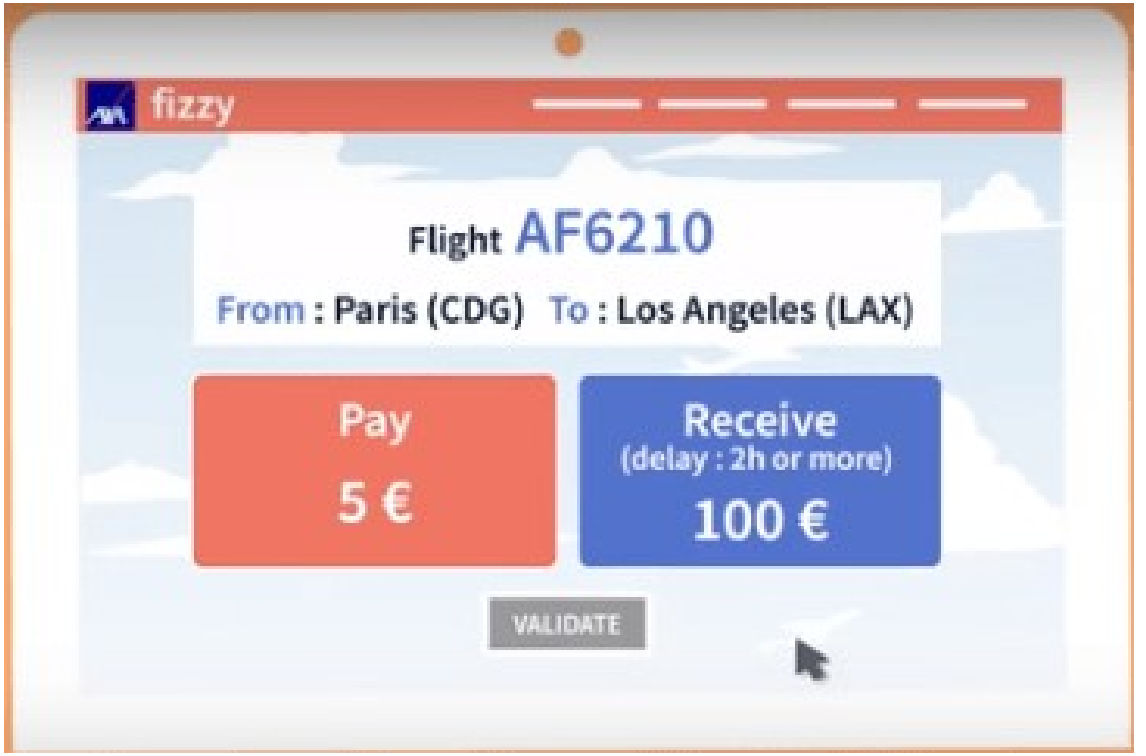
```
contract token {
    mapping (address => uint) public coinBalanceOf;
    event CoinTransfer(address sender, address receiver, uint amount);

  /* Initializes contract with initial supply tokens to the creator of the contract */
  function token(uint supply) {
        if (supply == 0) supply = 10000;
        coinBalanceOf[msg.sender] = supply;
    }

  /* Very simple trade function */
    function sendCoin(address receiver, uint amount) returns(bool sufficient) {
        if (coinBalanceOf[msg.sender] < amount) return false;
        coinBalanceOf[msg.sender] -= amount;
        coinBalanceOf[receiver] += amount;
        CoinTransfer(msg.sender, receiver, amount);
        return true;
    }
}
```

Application Example: Insurance

Idea: encode an insurance policy in smart contract that “lives on the blockchain”

- Users can buy in by depositing into the contract
- Terms of contract are “transparent”: defined by code
- Contract pays out automatically if conditions met

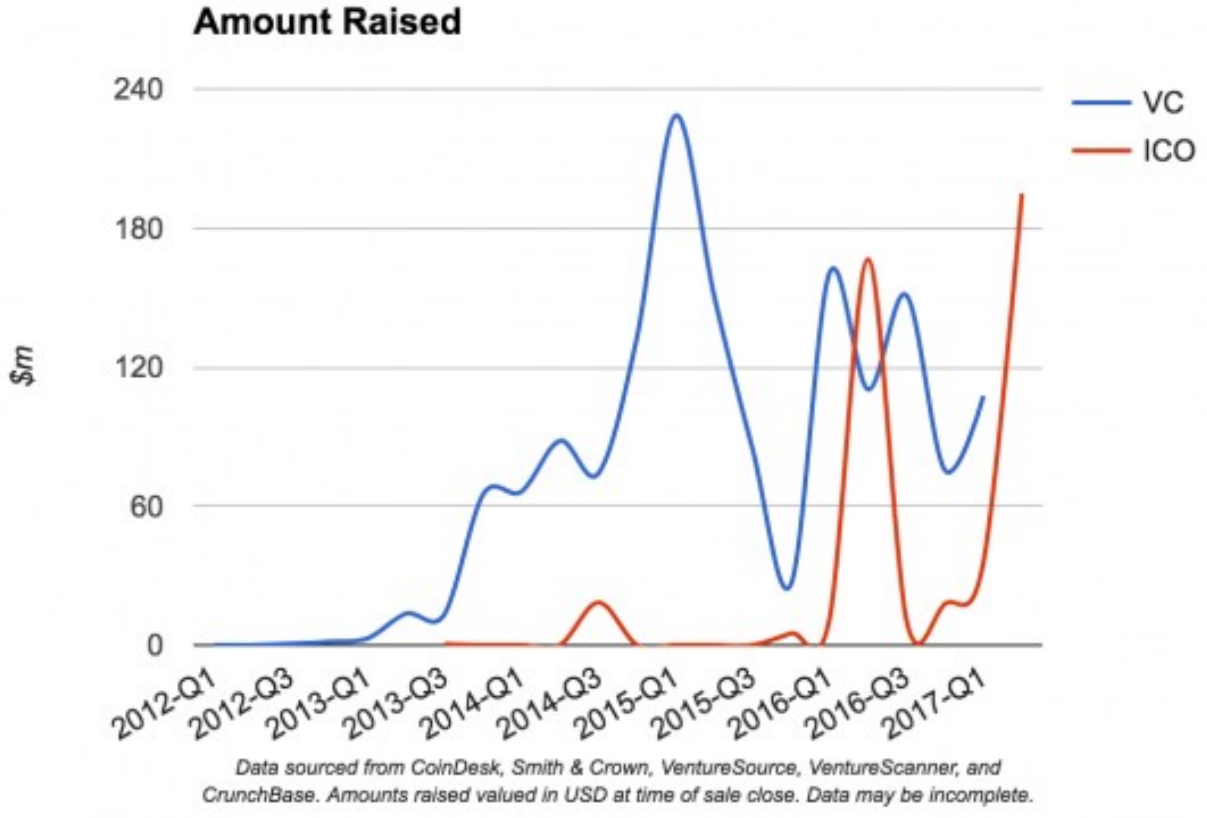
Ex: AXA “Fizzy”



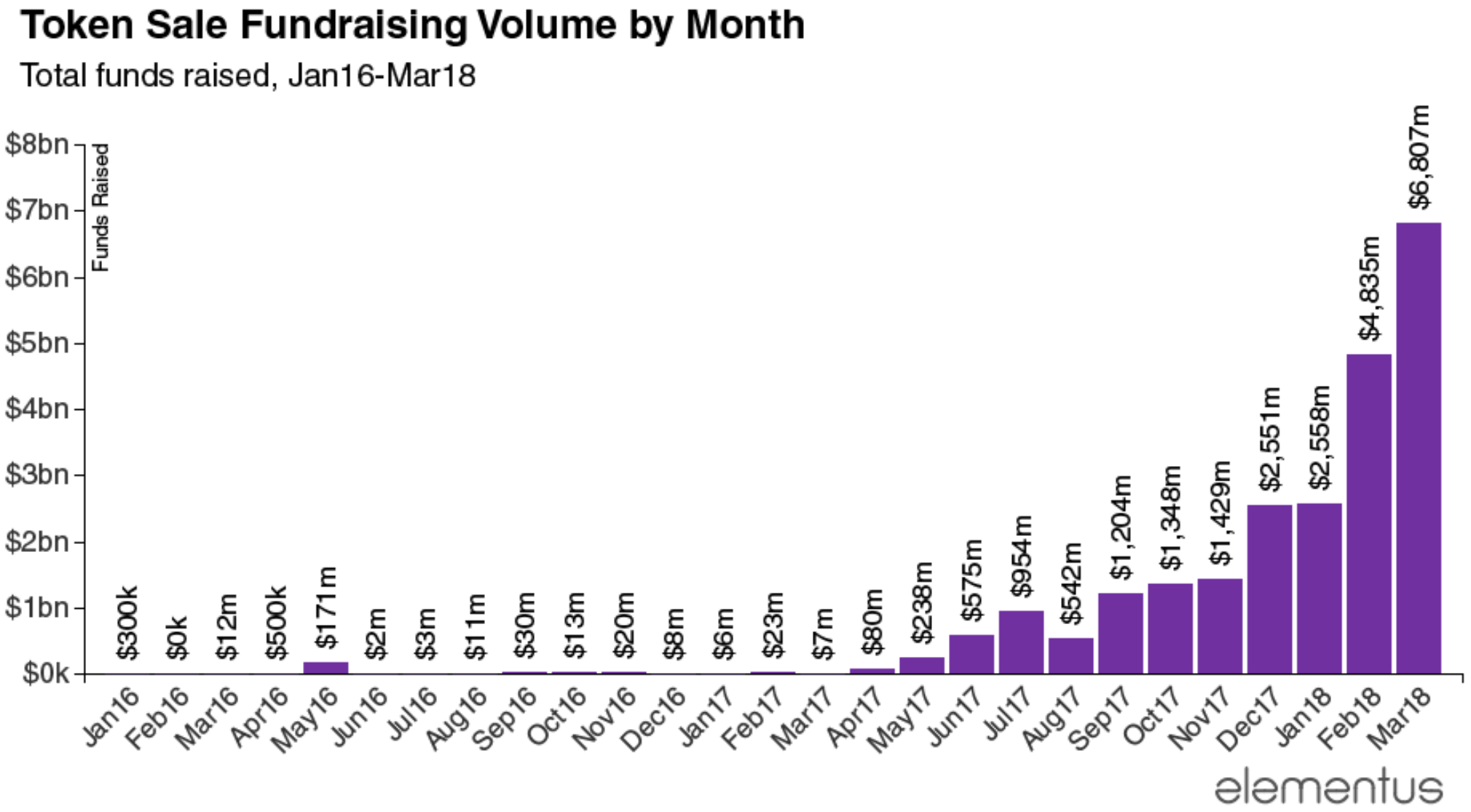
Enabled “new” form of investment...

ICOs: “Initial Coin Offerings”

- Digital tokens representing digital goods and services *yet to be created…*



Enabled “new” form of investment...



The Problem of Software Bugs

When everyone is running the **exact same code** (on or off the blockchain), one bug can be fatal

- Example: Ethereum DAO hack:
  attacker exploited one bug to steal $70M+

A “Universal Bug Bounty”

First successful hacker can cause a *lot* of damage

Lecture Outline

- Introduction: What is a Blockchain?
- Applications: What are Blockchains Good For?
- Smart Contracts: Can Blockchains Compute?
- **Consensus: How do Blockchains Coordinate?**
- Privacy: Can Blockchains Keep Secrets?
- Wrap-up: Promise and Challenges

Stake, Influence, and Consensus

Any organization – *or blockchain* – must determine:

- Who holds a *stake* in decision-making
- How much *influence* each stakeholder wields
- How decisions are actually agreed on: *consensus*

Without secure stake, consensus foundations → fail

Blockchain Consensus Now & Future

Many foundations for stake & consensus possible

We'll look at a few examples:

- Proof-of-Work Mining (common now)
- Private/Permissioned Ledgers ("now")
- Proof-of-Stake Ledgers (emerging)
- Proof-of-Personhood (research stage)

Blockchain Consensus Now & Future

Many foundations for stake & consensus possible

We'll look at a few examples:

- **Proof-of-Work Mining (common now)**
- Private/Permissioned Ledgers ("now")
- Proof-of-Stake Ledgers (emerging)
- Proof-of-Personhood (research stage)

Nakamoto Consensus

Public blockchains such as Bitcoin, Ethereum use consensus by crypto-lottery

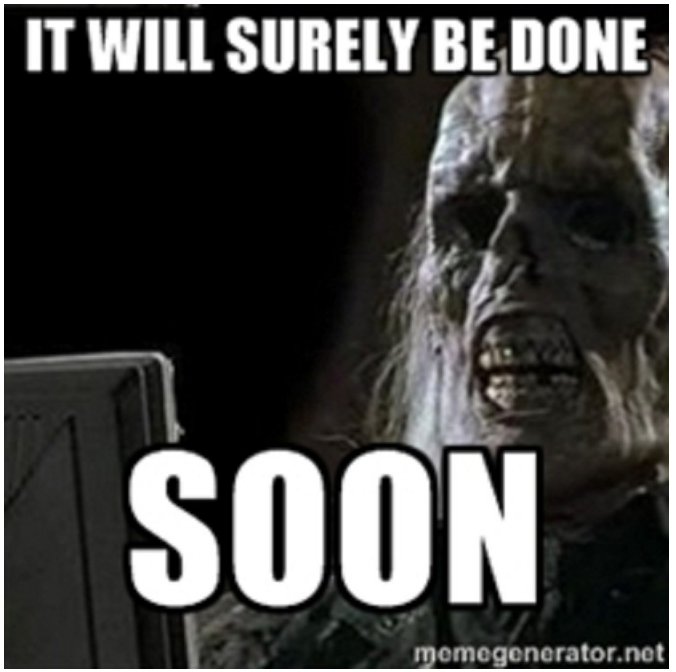
1) **Miners** print their own "lottery tickets" by solving crypto-puzzle (**proof-of-work**)

2) Winner gets to add one **block** to blockchain; typically gets **reward**: e.g., print new money

3) All miners gravitate to **longest chain.** Repeat.

Drawbacks of Nakamoto Consensus

- **Transaction delay**
  - Any transaction takes ~10 mins *minimum* in Bitcoin
- **Weak consistency:**
  - You're not *really* certain your transaction is committed until you wait ~1 hour or more
- **Low throughput:**
  - Bitcoin: ~7 transactions/second
- **Environmental costs:**
  - Miners waste energy *just to prove they did*



Public Blockchain Cost, Availability

Public blockchains (Bitcoin, Ethereum) also present *cost* and *availability* risks…

- Because *anyone* can impose transaction loads
- Or cause the price of transactions to jump

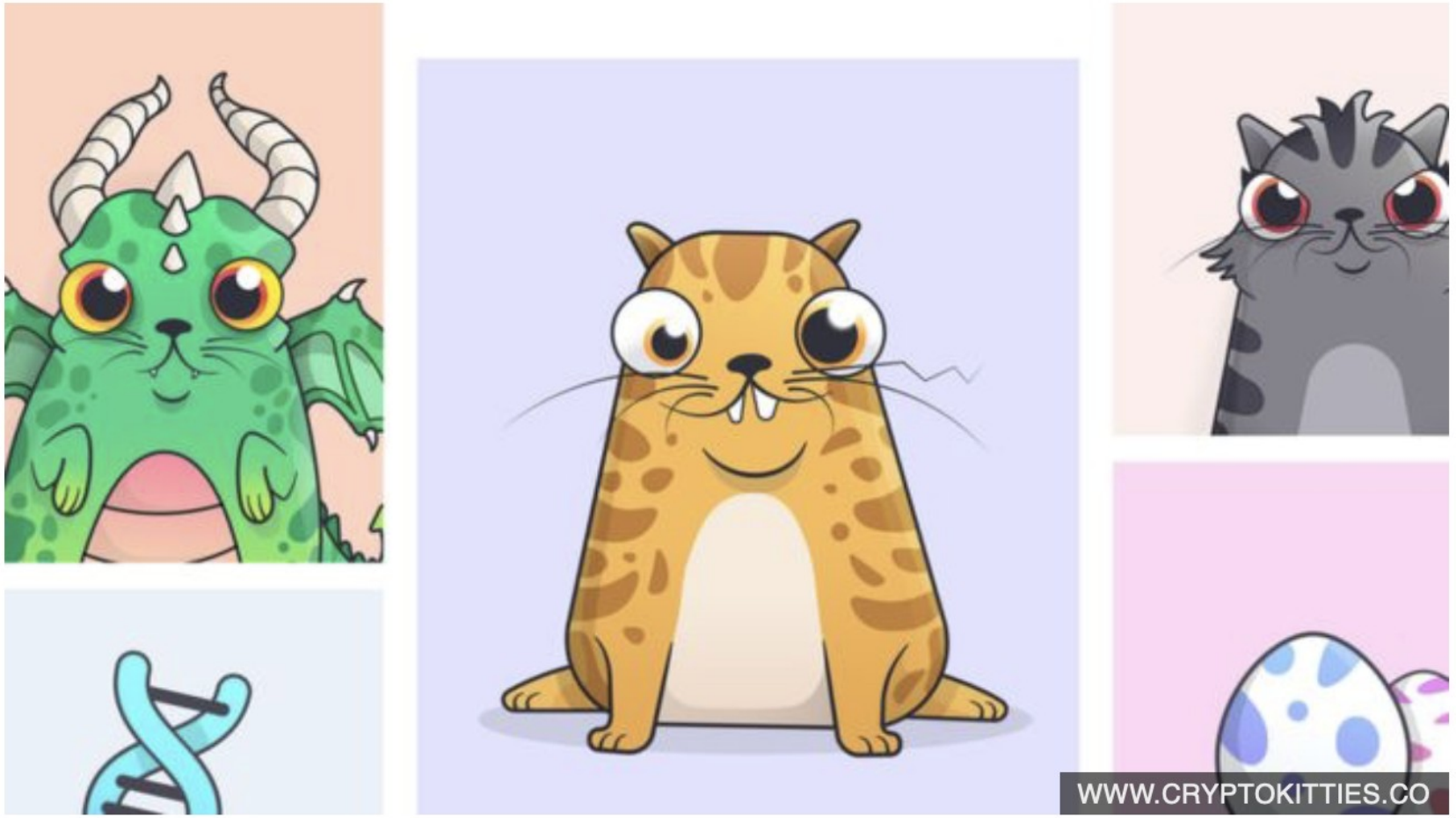
Limited scalability and transaction capacity can lead to outages due to uncontrollable events.

- Examples: Crypto-Kitties, Fomo3D, …
- Overload from trading: everything else stops

CryptoKitties craze slows down transactions on Ethereum

5 December 2017

Share



A new craze for virtual kittens is slowing down trade in one of the largest crypto-currencies.

How the winner got Fomo3D prize — A Detailed Explanation

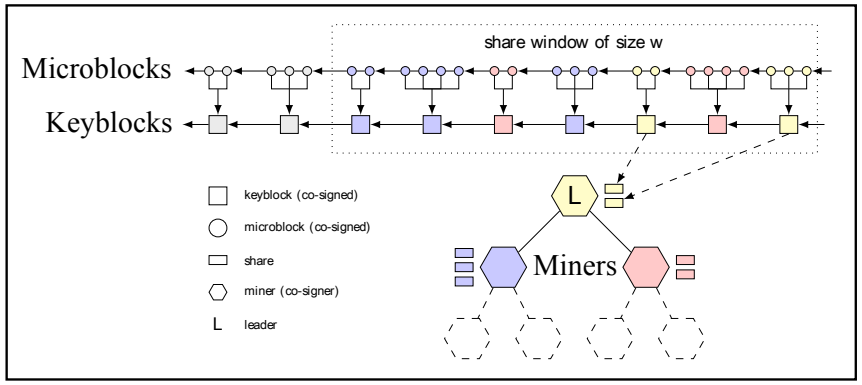
Solving these challenges is not easy

Scaling & Performance Goals

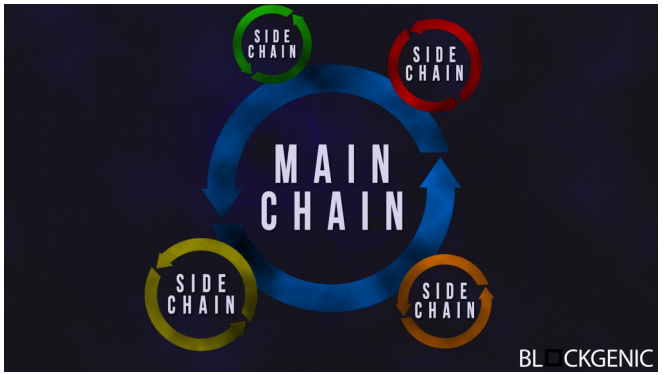
- Increased transaction processing **throughput**
  - From 4.7 TPS to VISA (1000s of TPS) and beyond
- Reduced transaction processing **latency**
  - From 10s of minutes to seconds to milliseconds…
- Reduced cost of on-chain data **storage**
  - Don’t make *everyone* store *everything, forever*
- Reduced cost of on-chain **computation**
  - Preferably not millions of times slower than native

On- versus Off-Chain Scaling

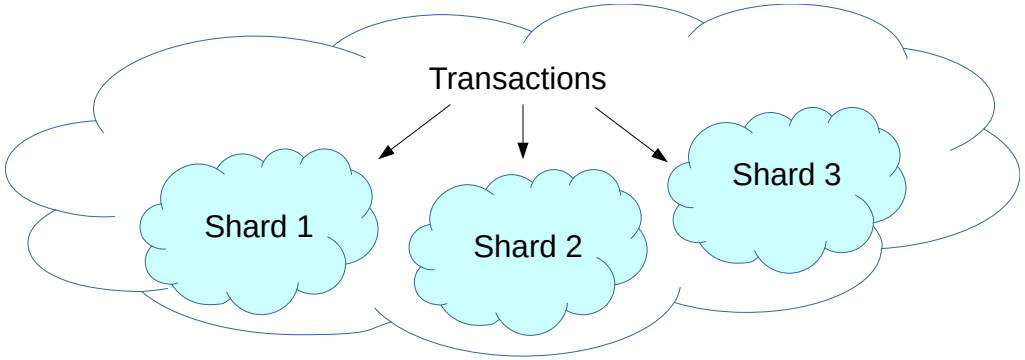
Scalable BFT



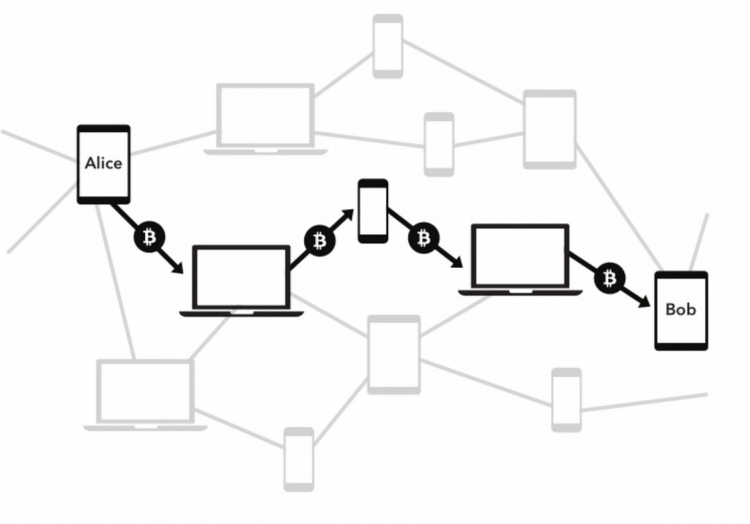
Sidechains



Horizontal Sharding



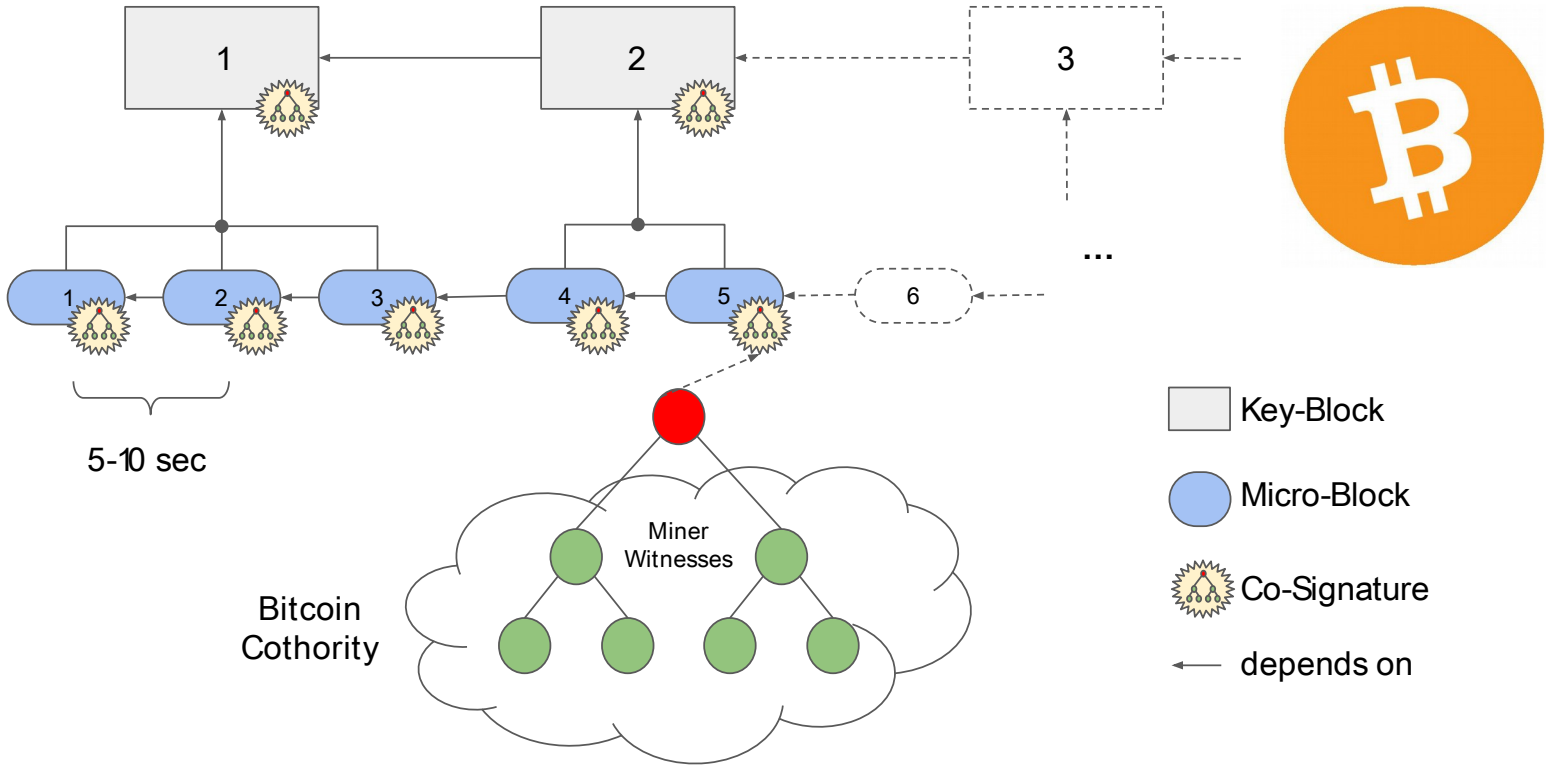
Payment Networks



ByzCoin: Fast, Scalable Blockchains

Scalable PBFT blockchain consensus [USENIX Security '16]

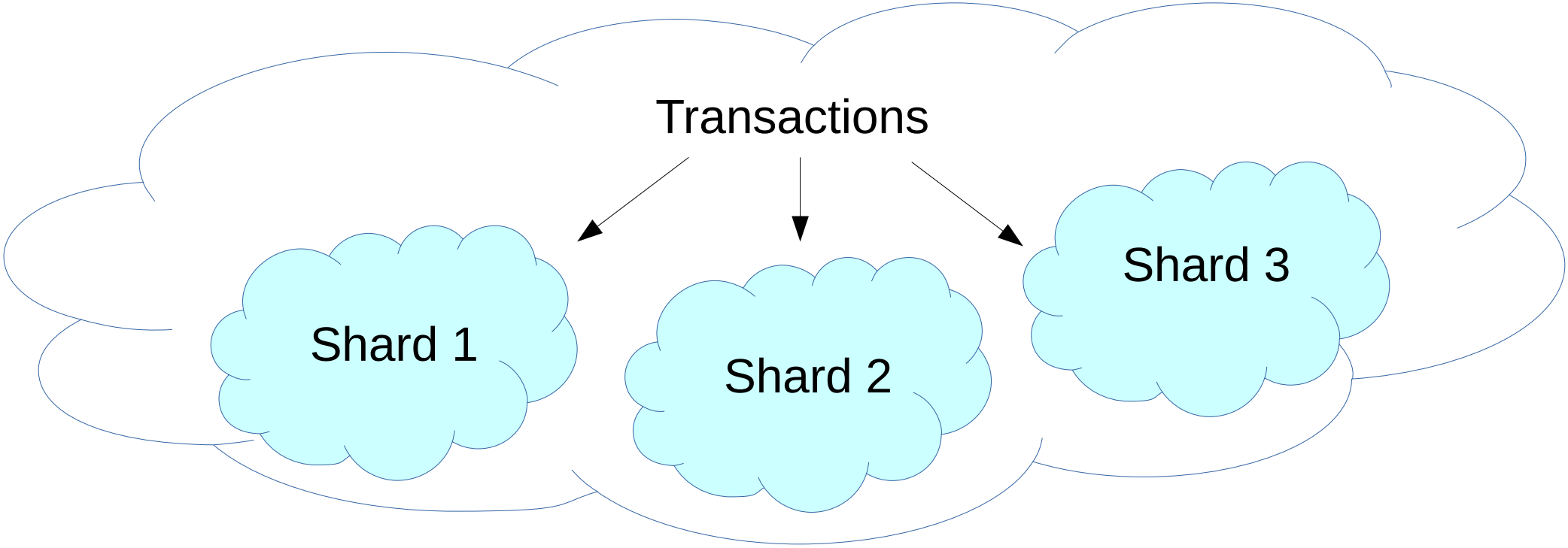
- Permanent transaction commitment in seconds
- 700+ TPS demonstrated (100x Bitcoin, ~PayPal)



Scaling Blockchains via Sharding

OmniLedger: A Secure Scale-Out Ledger [S&P 18]

- Break large collective into smaller subgroups
- Builds on scalable bias-resistant randomness protocol (IEEE S&P 2017)
- 6000 transactions/second: competitive with VISA



Proof-of-Work as a Basis for Stake

Proof-of-Work requires miners to *expend energy* surmounting an *artificial barrier to entry*,
just in order to prove they did that.

Important point: Proof-of-Work servers *no purpose* other than to erect an artificial barrier to entry and create competition for mining rewards!

Have we seen human practices like this before?

PoW: Membership by Hazing Ritual

Anything that not everyone will do on a whim: entire purpose is to *create a barrier to entry*

May be uncomfortable and/or embarrassing…

PoW: Membership by Hazing Ritual

Or just plain weird…

- MIT ‘58: using Oliver Smoot to measure bridge

PoW: Membership by Hazing Ritual

Or difficult, requiring energy and cooperation

- Yap: chisel a giant circular “coin” out of stone available only on another, distant island

PoW: Bitcoin’s Hazing Ritual

Digitally flip coins.

Many coins.

Billions of them.

By forming new “blocks” and feeding them into a *cryptographic hash*

- Converts any information to pseudorandom number

Repeat endlessly.


JUST...ONE...
...MORE...BITCOIN

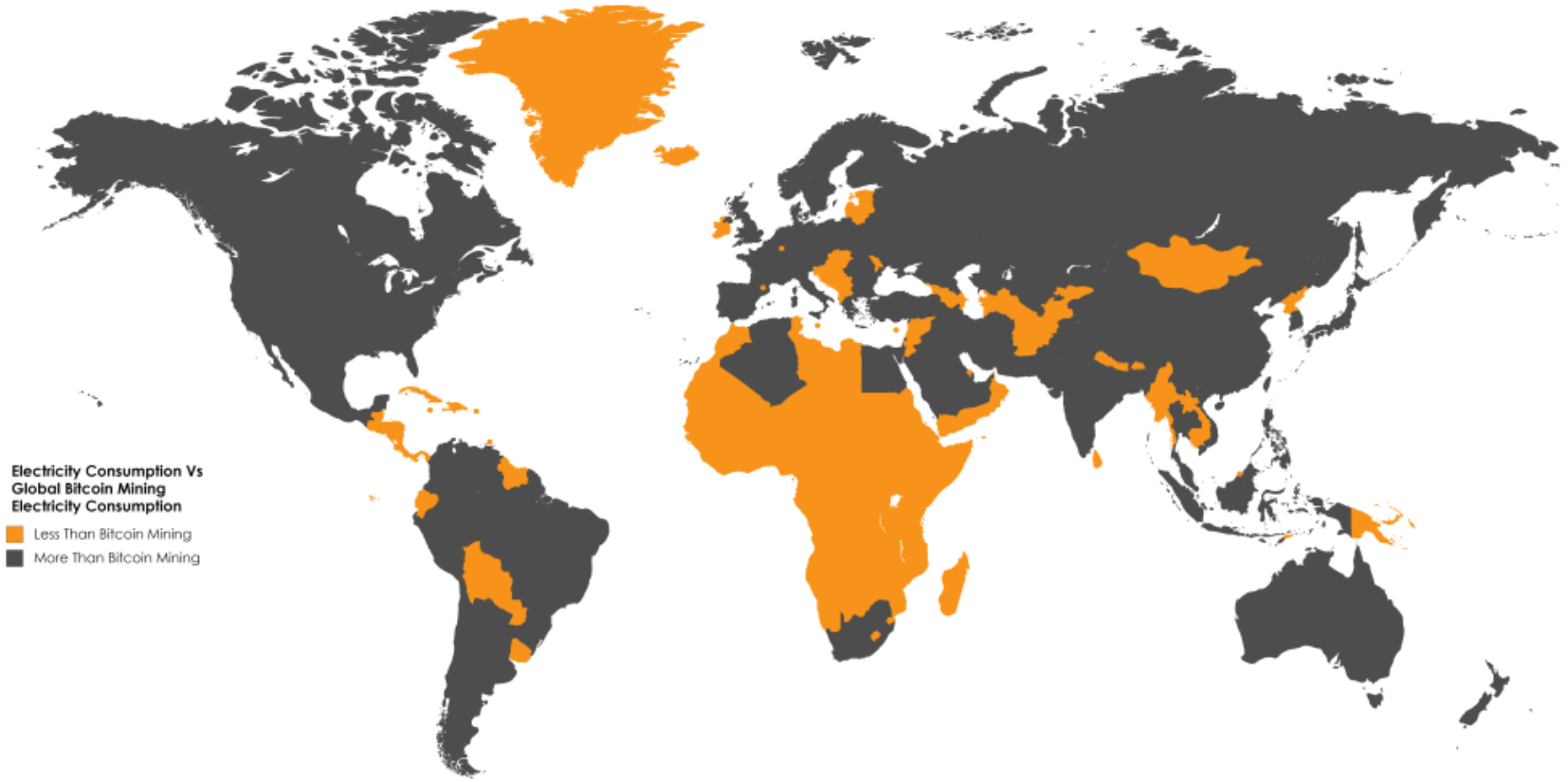
Environmental Costs

Proof-of-work = “scorched-earth” blockchains

- Bitcoin makes BTC scarce by making miners prove they **wasted energy**
- Serves **no purpose** except to prove they did it
-

Bitcoin Energy Consumption Index

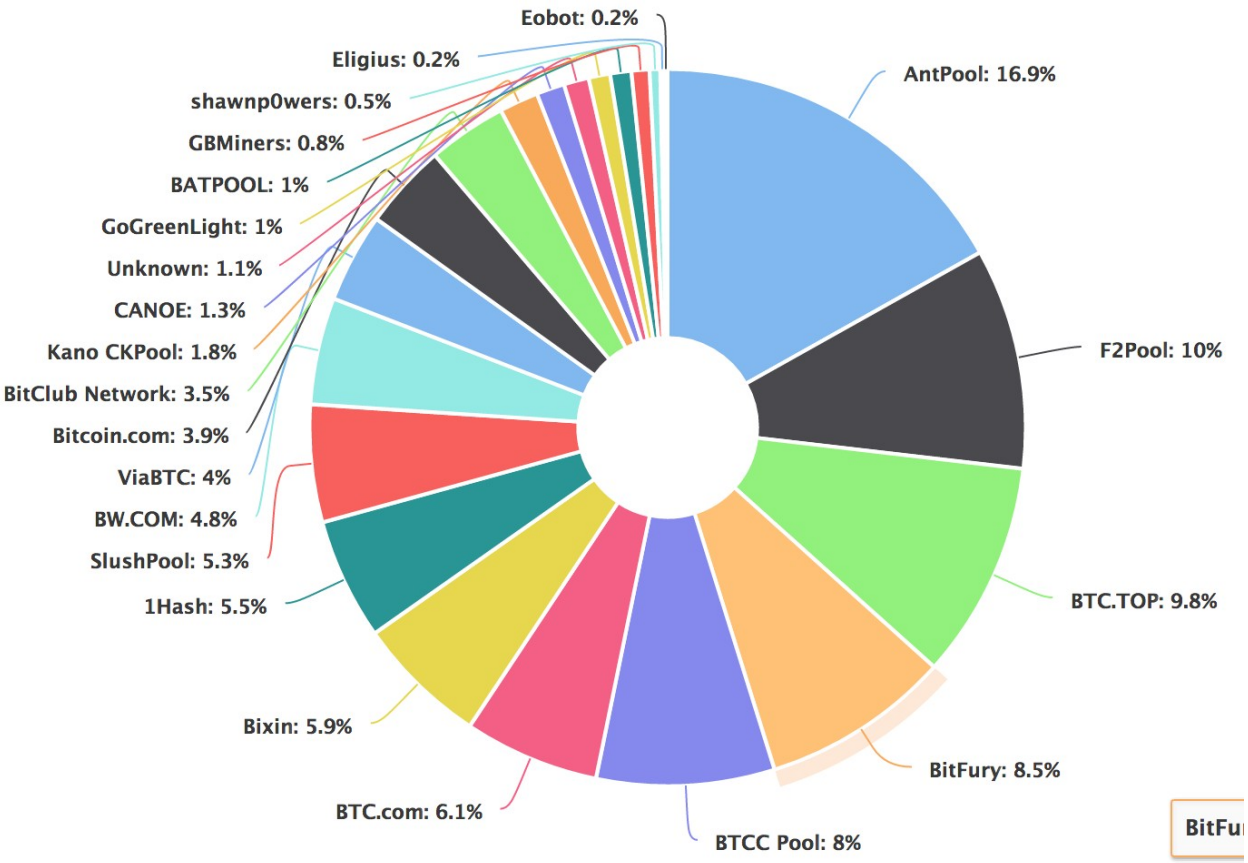
Bitcoin now *wastes* more energy than
159 countries use for their people to live on!



Not Even Decentralized Anymore

Market incentives drive consolidation of hashrate or “voting power” to a few powerful mining pools

- Over 60% currently in one country (China)
- Any faction >51% can control or veto decisions, censor, etc.



A Problem Not Unique to Bitcoin

Most cryptocurrencies aren't that decentralized

are we decentralized yet?

Name	Symbol	Consensus	Miners/voters Incentivized?	# of entities in control of >50% of voting/mining power
Bitcoin	BTC	PoW	Y	3
Ethereum	ETH	PoW	Y	3
Ripple	XRP	RPCA (voting system)	N	1
Bitcoin Cash	BCH	PoW	Y	3
Litecoin	LTC	PoW	Y	2
Cardano	ADA	PoS	N	1
Stellar	XLM	FBA	N	1
Neo	NEO	DBFT	N	1

Blockchain Consensus Now & Future

Many foundations for stake & consensus possible

We'll look at a few examples:

- Proof-of-Work Mining (available now)
- **Private/Permissioned Ledgers ("now")**
- Proof-of-Stake Ledgers (emerging)
- Proof-of-Personhood (research stage)

Permissioned Ledgers

Just decide **administratively** who participates;
Fixed or manually-changed group of “miners”

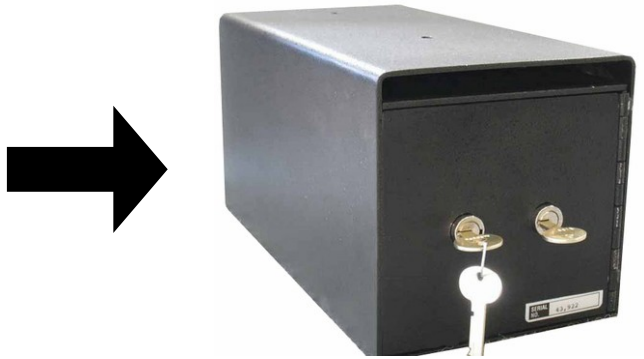
- ☺ No proof-of-work needed → low energy cost
- ☺ More mature consensus protocols applicable
- ☹ Higher human organizational costs
- ☹ No longer open for “anyone” to participate

The Weakness of Limited Scale

Public/permissionless designs in principle have the advantage of *security scaling with size*

- As more participants arrive, security increases

Weakest-link security



Strongest-link security

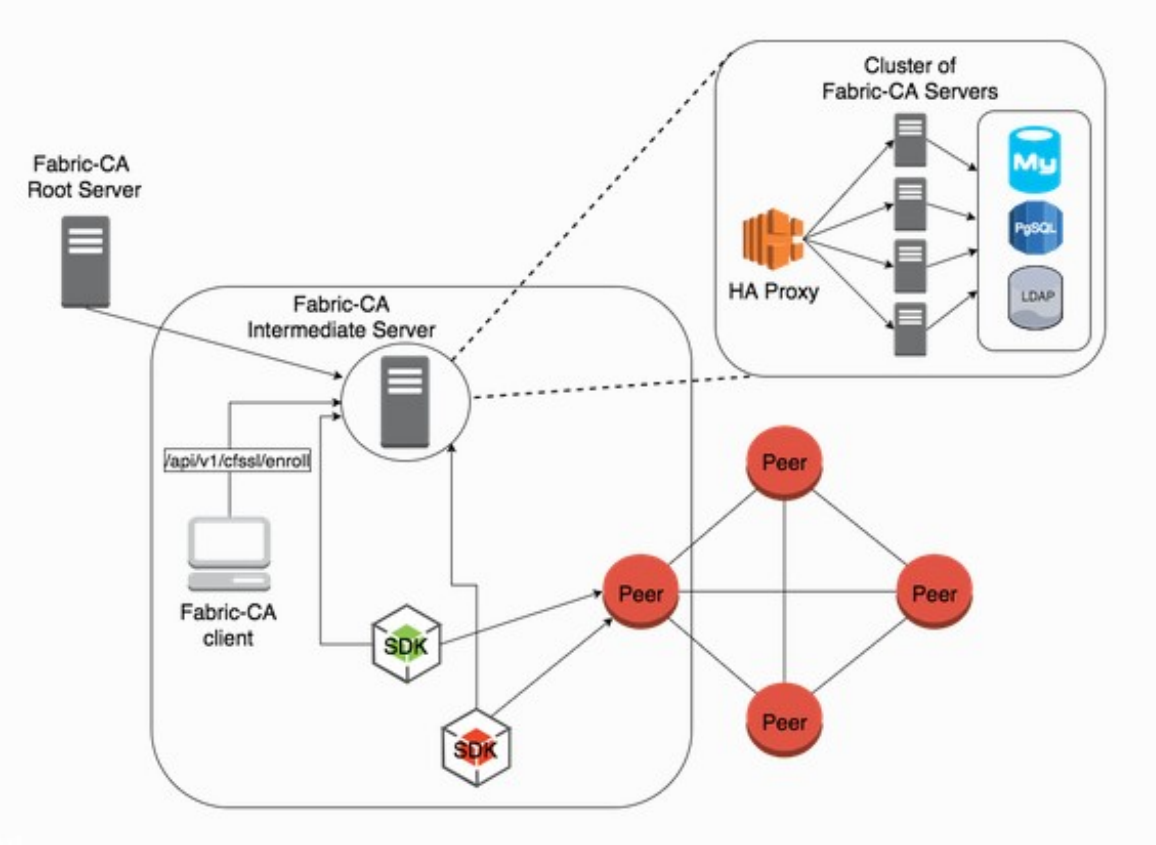
Scalable Strongest-link security

Closed participation designs limit security scaling!

Beware the Lemon Market (Again)

Many (most?) permissioned blockchains currently on the market introduce *single points of failure* in their approaches to permissioning!

- Whole network depends on one *certificate authority* server
- If compromised, attacker can impersonate *whole network*



Blockchain Consensus Now & Future

Many foundations for stake & consensus possible

We'll look at a few examples:

- Proof-of-Work Mining (common now)
- Private/Permissioned Ledgers ("now")
- **Proof-of-Stake Ledgers (emerging)**
- Proof-of-Personhood (research stage)

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
- Securing proof-of-stake is a nontrivial, interesting, but mostly-solved problem
  - e.g., Orobouros, Algorand
  - Also implementable with CoSi + SkipChains + OmniLedger + RandHound

Blockchain Consensus Now & Future

Many foundations for stake & consensus possible

We'll look at a few examples:

- Proof-of-Work Mining (common now)
- Private/Permissioned Ledgers ("now")
- Proof-of-Stake Ledgers (emerging)
- **Proof-of-Personhood (research stage)**

Toward People-Centric Blockchains

Can we build decentralized technology that will

- Securely *remain decentralized* over time?
- Offer a fairness metric *meaningful to people*?
- Be accountable to *all human stakeholders?*

> “We must act to ensure that technology is designed and developed to serve humankind, and not the other way around”
>
> - Tim Cook, Oct 24, 2018

One Person One Vote?

Proof-of-Personhood [IEEE S&B ‘17]

- Proof-of-Stake but *one stake unit per person*

Proof-of-Personhood: Approaches

- Legacy Identities (e.g., government-issued)
  - Require costly ID-checking, not that hard to fake
- Global Biometric Databases (India, UNHCR)
  - Huge privacy issues, false positives+negatives
- Trust Networks (PGP “Web of Trust” model)
  - Unusable in practice, doesn’t address Sybil attacks
- Pseudonym Parties [SocialNets ‘08]
  - Requires *in-person* participation, physical security
  - Low-cost: verifies *only* personhood, not ID or trust

Blockchain Consensus Summary

Any decentralized system must define who its stakeholders are, how much influence each get

- **Proof-of-Work:** simple but an energy disaster
- **Permissioned:** efficient but closed, often weak
- **Proof-of-Stake**: permissionless, low-energy, but still can have high concentrations of stake
- **Proof-of-Personhood:** attempts to distribute stake more widely among *human* stakeholders

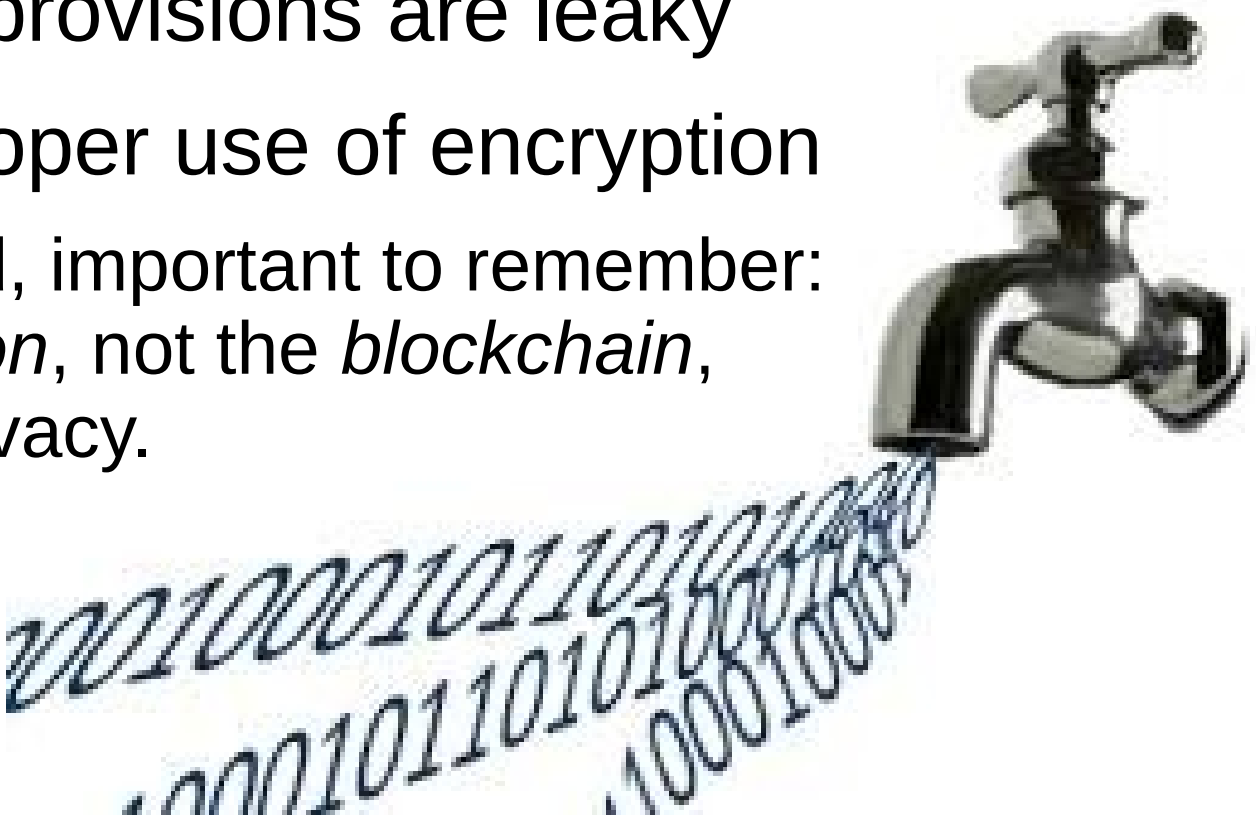
Lecture Outline

- Introduction: What is a Blockchain?
- Applications: What are Blockchains Good For?
- Smart Contracts: Can Blockchains Compute?
- Consensus: How do Blockchains Coordinate?
- **Privacy: Can Blockchains Keep Secrets?**
- Wrap-up: Promise and Challenges

The Blockchain Privacy Challenge

Blockchains protect the **integrity** of data by *giving everyone a copy* for independent checking

- This works *against* **privacy** & confidentiality
- Current privacy provisions are leaky
- Solvable with proper use of encryption
  - When combined, important to remember: it's the *encryption*, not the *blockchain*, that protects privacy.



Blockchain and Data Privacy

Is data “on a blockchain” privacy-protected?

- By default, **no**: blockchain makes privacy *worse*
  - Gives copies to *many* parties; *any one* can leak!

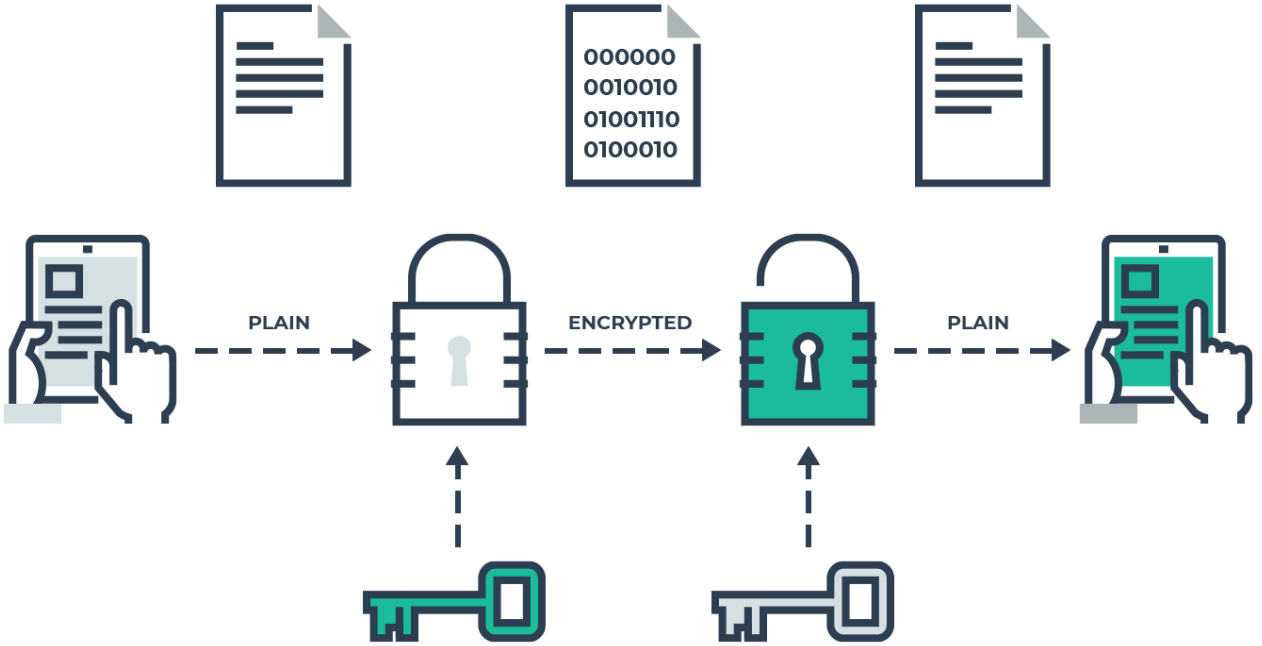
Why it’s critical to separate *integrity* from *privacy*

- Consensus and replication is *good* for integrity
- Consensus and replication is *bad* for privacy

So how can we *actually* get data privacy?

So How Do We Get Privacy?

Encryption, of course!



Encrypt data before storing, decrypt on use…

But Who Holds the Keys?

Any encrypted data is secured with a *private key*

- A private key is *just information* (a number)!
- If the *key* leaks, anyone can decrypt the data
  - Regardless of where it's stored: cloud, blockchain…

If the private key is held by a *single party*,
then that party is a *single point of compromise*

- If key-holder hacked, attacker gets *everything*
- Regardless of whether it's "on a blockchain"!

Beware the Lemon Market (Again)

Many blockchain-based data protection/sharing designs just write *access logs* onto a blockchain

- If the access logs are on the blockchain, must be secure and tamperproof, right? *Wrong!*
- Blockchains only protect the *integrity* of logs if they were **correct** and **complete** *when written!*

Typical designs still entrust a *single party* to store private keys, check access rights, and log access

- Key-holder compromised: *logs don't get written!*

The Data Availability Problem

Will non-public data be there when you need it?

Many blockchains aren't scalable enough to hold large amounts of data *on the blockchain itself*

- Store only a *hash* of the data on the blockchain
- Must store the data in a server or "cloud"…

But that's another **single point of compromise**: if storage server or cloud is down or unreachable, you won't be able to access the data!

How to Get Privacy, Accountability?

Blockchains don't protect privacy & accountability without single points of compromise; how can we?

With another technology: **secret sharing**.

- Known for decades, but new in blockchain tech

Essential idea: after encrypting data,
"deal" the secret key to a *threshold* ***t*** of ***n*** parties

- At least ***t*** parties must *work together* to recover
- If just one (or fewer than ***t***) compromised, attacker can't recover the key - or *any* data!

Secret Sharing: Illustration

Suppose you're a pirate & bury your treasure…

Keeping the Location Secret

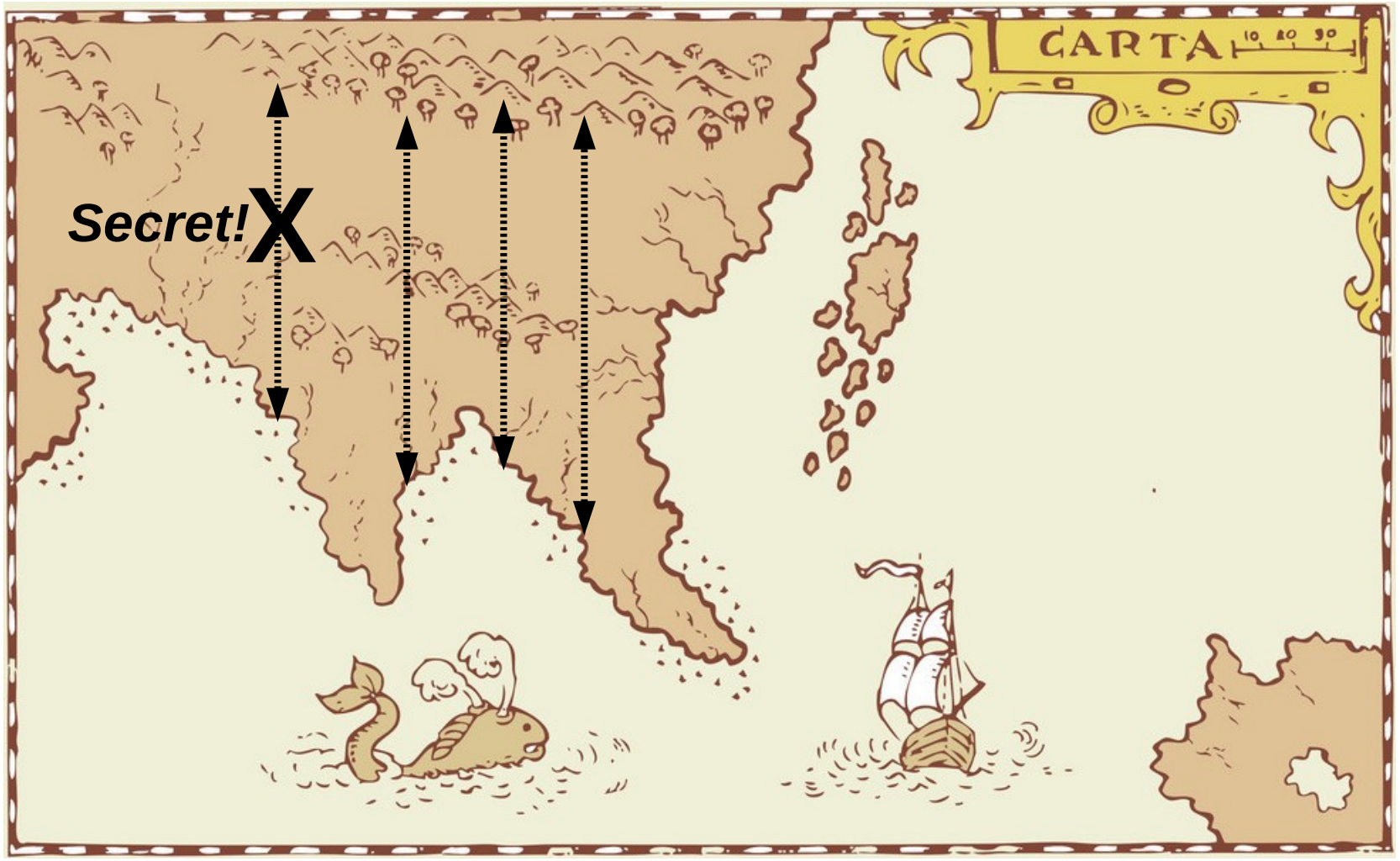
You have 3 henchmen who you want to send back for it later, but you don’t trust *any one* completely

Secret Sharing: Illustration

You mark the spot between two reference points

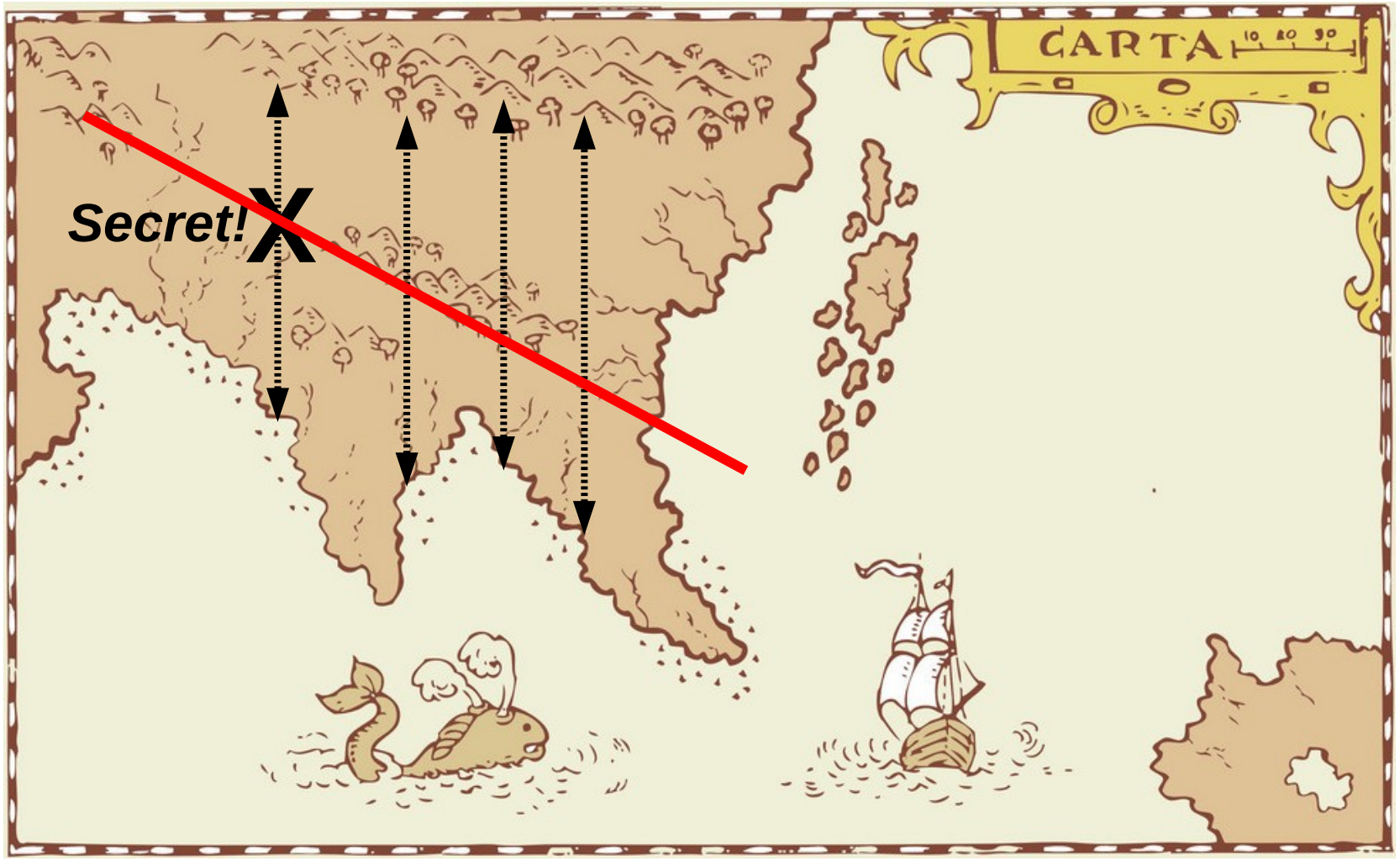
Secret Sharing: Illustration

Then draw three parallel reference lines…



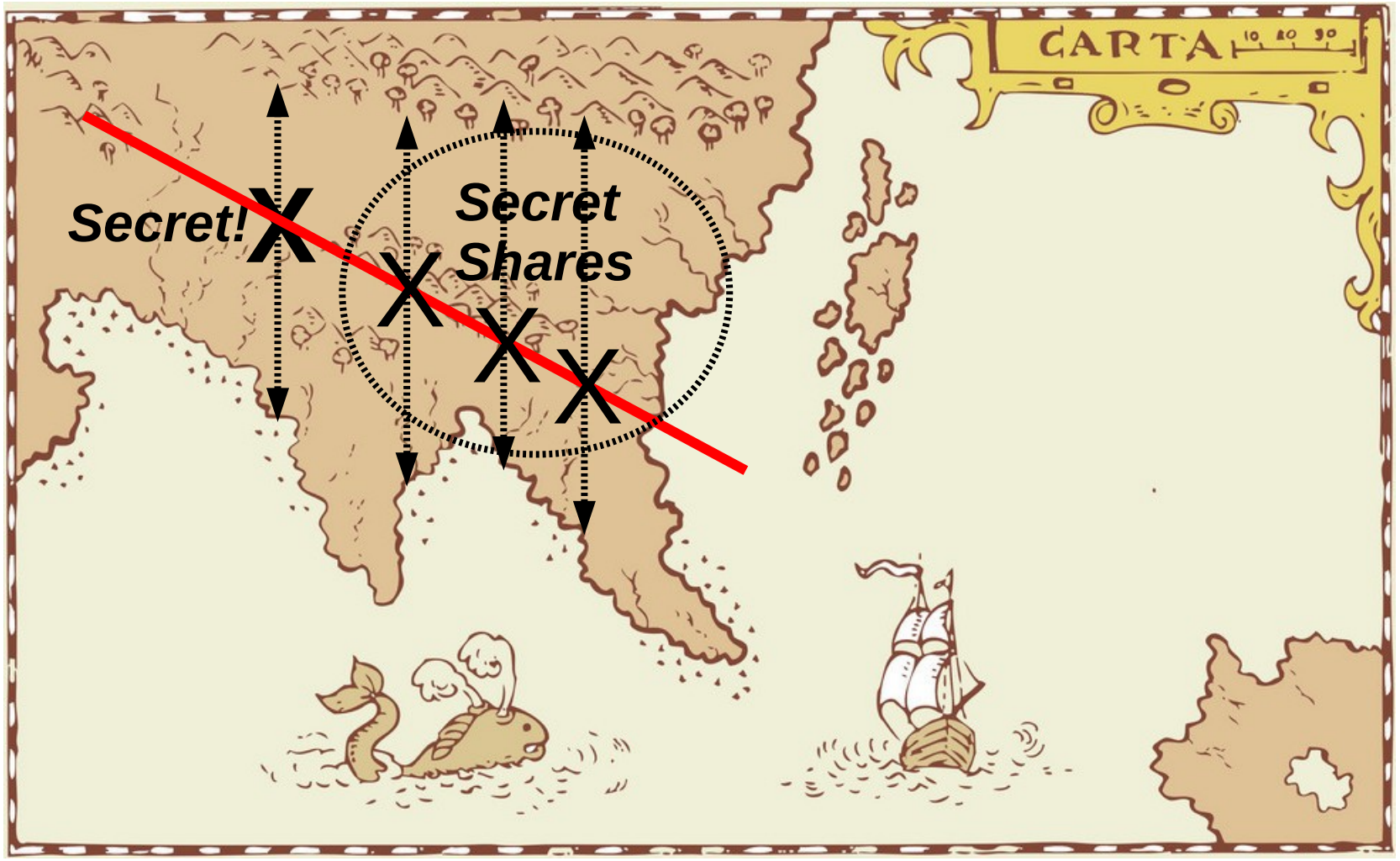
Secret Sharing: Illustration

…and another line intersecting all four…



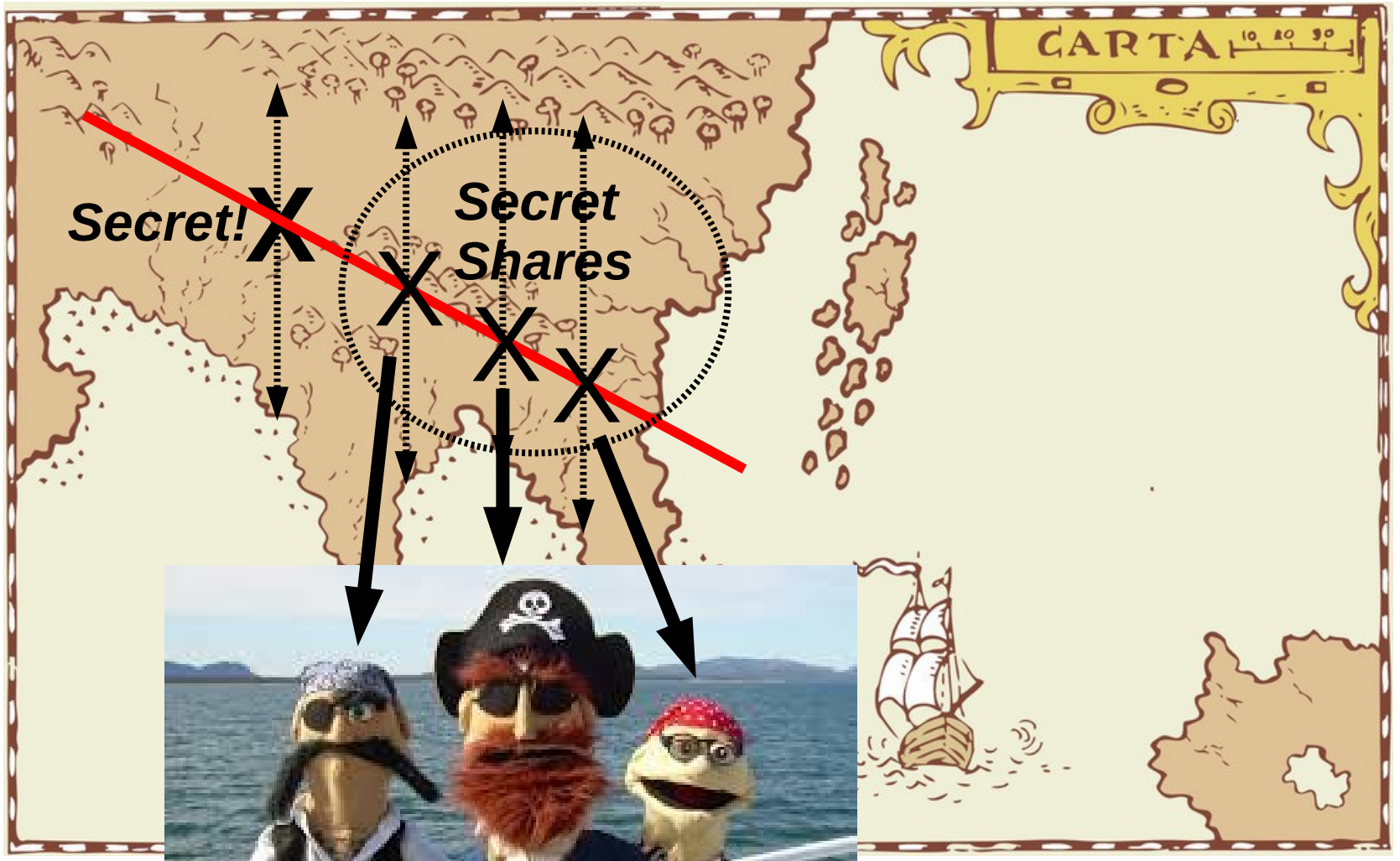
Secret Sharing: Illustration

The intersection points are the *secret shares...*



Secret Sharing: Illustration

You give *one* of these shares to *each* henchman



Threshold Secret Sharing

Now suppose your henchmen come back later to recover the treasure…

- Any **one** henchman won't know how to find it
- Any **two** henchmen together will be able to!

You get both **threshold privacy** of the secret…

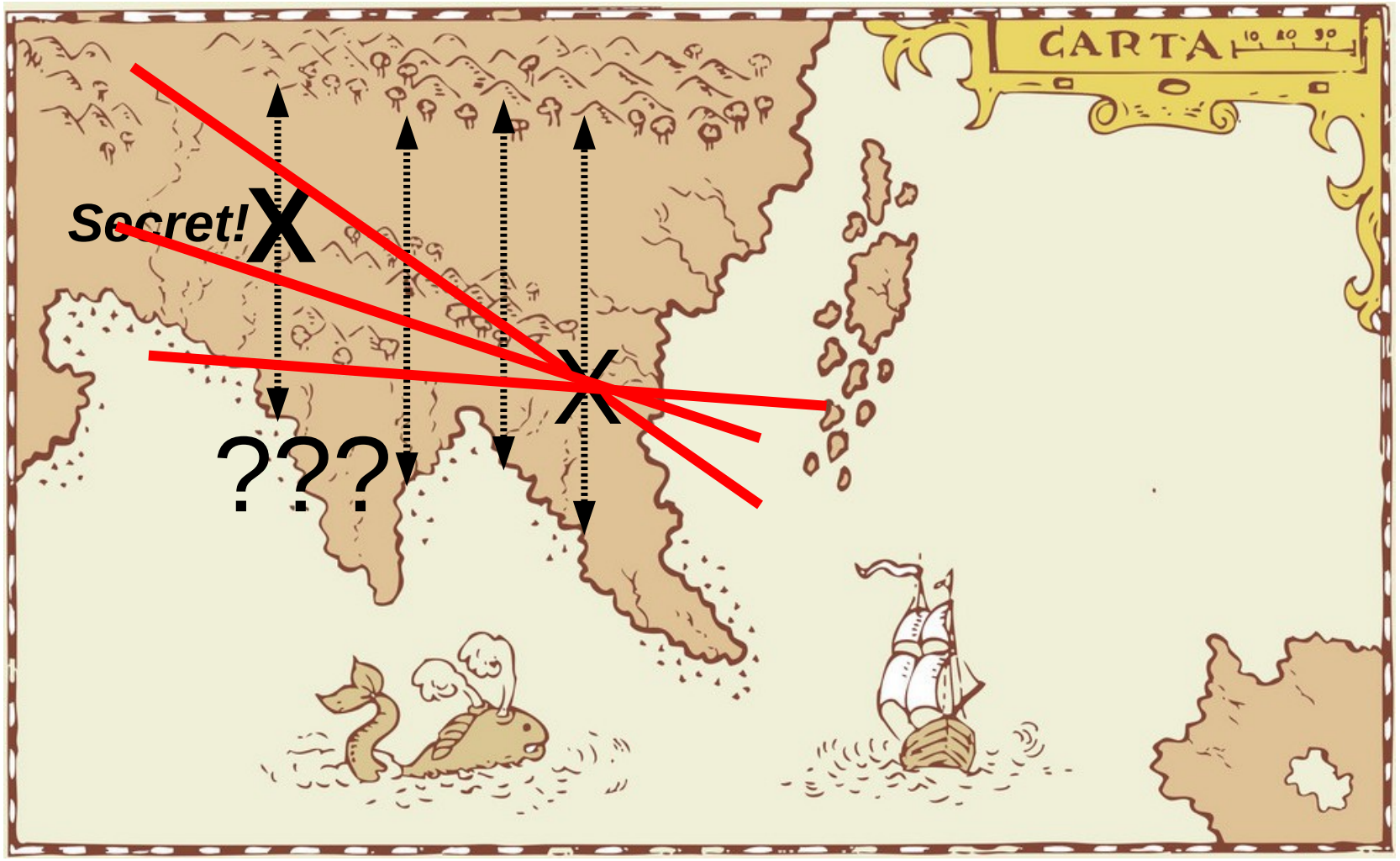
- No single compromised party can recover it

You also get **threshold availability** of the secret

- Can still recover if one henchman goes missing

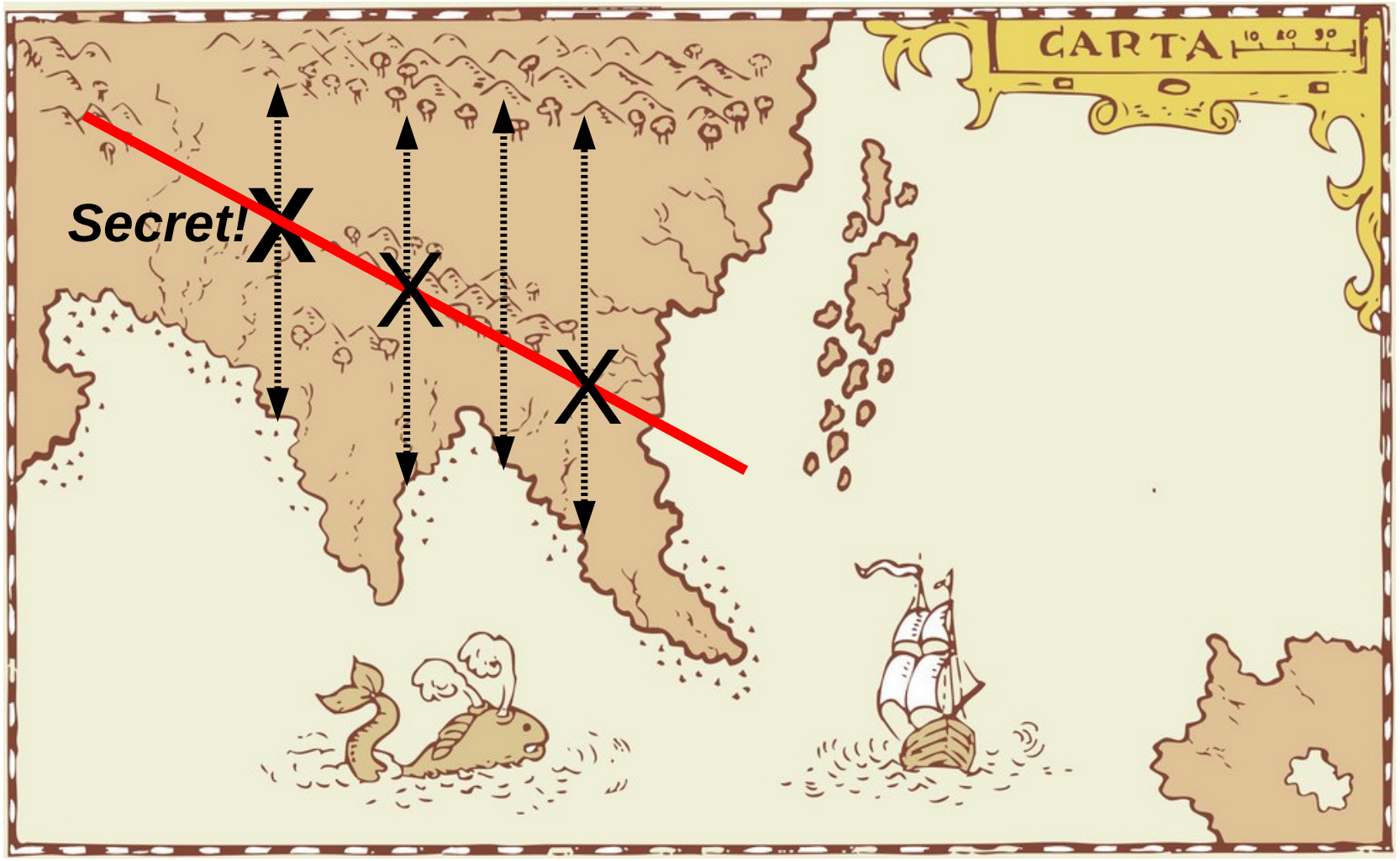
Secret Sharing: Illustration

One henchman alone can’t recover secret



Secret Sharing: Illustration

…but *any two* working together can!



DEDIS On-Chain Secrets

“CALYPSO: Auditable Sharing of Private Data”

Allow blockchain to hold and *manage secrets*
via verifiable, transparent, dynamic access policies

- Example: decryption keys, access lists for documents
- Example: login credentials for access to services

Ensures that attackers cannot:

- (a) Modify or delete existing access records
- (b) Access sensitive data without access being logged
- (c) Prevent data from being revealed as policy dictates

Existing blockchains can ensure only (a), not (b) or (c)

Scaling Privacy & Integrity with Size

Ensures architecturally that not just the *integrity* but also the *privacy* of data entrusted to blockchain scales as participation increases

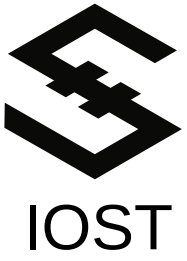
Weakest-link security

Strongest-link security

Scalable Strongest-link security

Ongoing Industry Adoption

Supporting partners collaborating with DEDIS



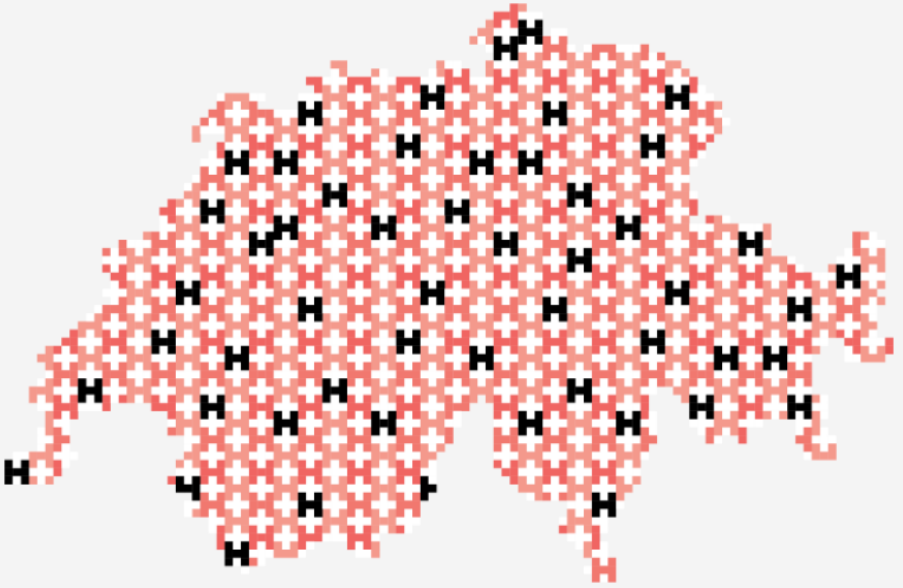
Other companies building on DEDIS research

Application: Personalized Health

DPPH

Data Protection
in Personalized Health

Personalized Medicine, Personalized Health Research Project funded by the Strategic Focus Area Personalized Health and Related Technologies (PHRT) of the ETH Board.



Application: Data Science

SWISS DATA SCIENCE CENTER

A COMPLEX JOURNEY MADE SIMPLE

We accompany the academic community and the industrial sector in their data science journey, putting to work AI and ML and facilitating the multidisciplinary exchange of data and knowledge

ETH zürich

EPFL ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

Screenshot

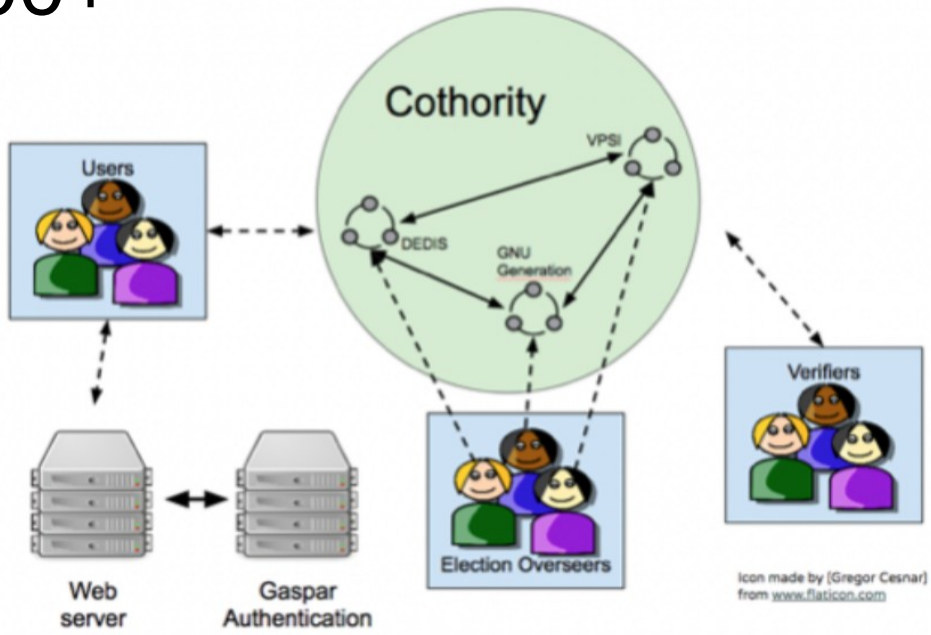
Application: Blockchain E-voting

Prototyped blockchain-based e-voting system

- State-of-the-art cryptographic security/privacy
- Validated, approved for deployment within EPFL community of 10,000+

Exploring next-generation e-voting technologies

- In contact with Geneva, Swiss Post e-voting efforts



Summary: Blockchain Data Privacy

Enabling blockchains to *manage confidential data* is an unsolved (but solvable) research challenge.

Beware common easy answers such as:

“Just encrypt the data”
→ important, but *who holds the keys*?

“Leave private data off the blockchain”
→ blockchain can’t ensure *data availability*

“Don’t worry, it’s a private blockchain”
→ reintroduces *single points of compromise*

Lecture Outline

- Introduction: What is a Blockchain?
- Applications: What are Blockchains Good For?
- Smart Contracts: Can Blockchains Compute?
- Consensus: How do Blockchains Coordinate?
- Privacy: Can Blockchains Keep Secrets?
- **Wrap-up: Promise and Challenges**

Key Takeaway Points

Blockchains can keep **any type of records**

- *Usable* in any application that wants a database
- *Needed* only if you want *no single trusted party*

An important and promising technology space

- Well worth exploration and early investment, with awareness of limitations and immaturity!
- Challenges: Scalability, Energy use, Limited Decentralization, Data Privacy, Availability…