# Introduction to Blockchains, Cryptocurrencies, and Smart Contracts

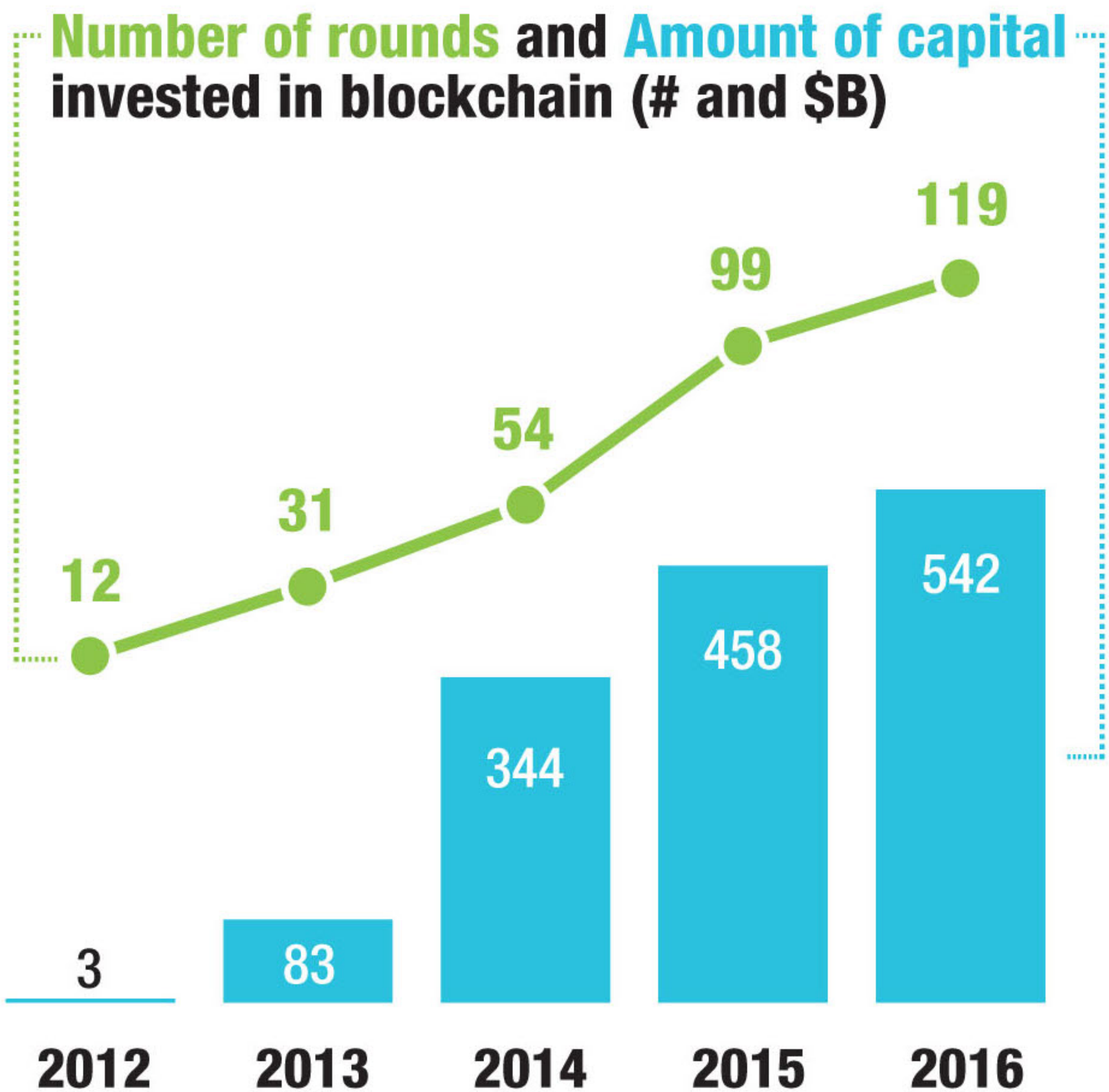
Bryan Ford
Decentralized/Distributed Systems (DEDIS)

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

# The Call of the Blockchain


(credit: Tony Arcieri)

# Broad Promise & Global Interest



Source: Money of the Future, Life.SREDA

There is a **decreasing tendency towards launching new blockchain companies:**

| | |
|---|---|
| 2016 | **169** |
| 2015 | **221** |
| 2014 | **233** |
| | new companies launched |

There is an **increase in investment rounds:**

| | |
|---|---|
| 2016 | **119** |
| 2015 | **99** |
| 2014 | **54** |
| | rounds |

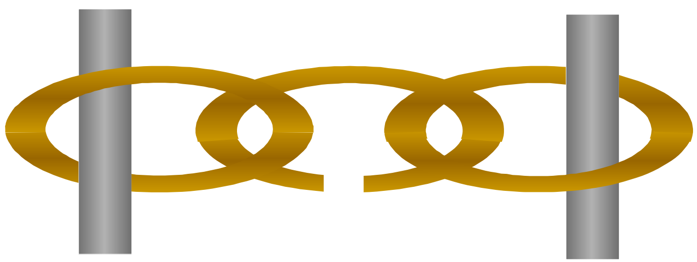
# Outline

- **Introduction: the distributed trust principle**
- Bitcoin blockchain: principles and operation
  - Decentralization via public ledgers
  - Privacy via cryptographic identities
  - Scarcity via proof-of-work
- Bitcoin limitations and design alternatives
  - Mining vs Permissioned, Proof-of-Stake, etc.
  - Scalability challenges and potential solutions
- Smart contract systems: Ethereum
- Privacy approaches: Zcash, Chain-Secrets

# The Fundamental Problem

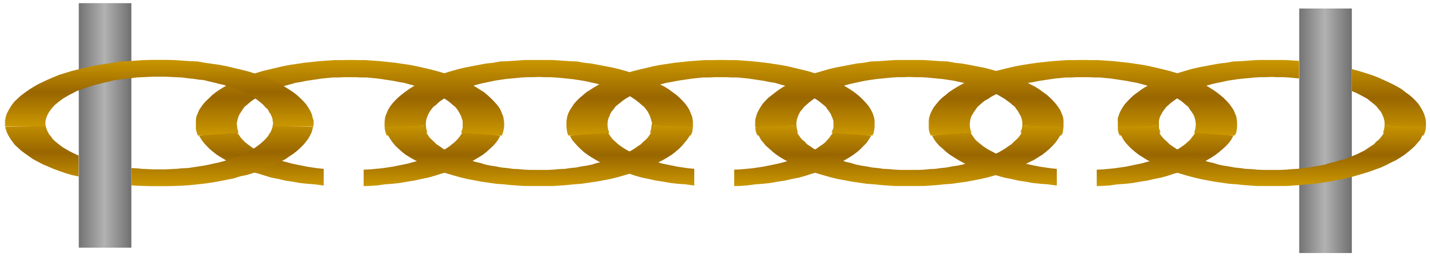
In today's IT systems, security is an afterthought

- Designs embody "weakest-link" security

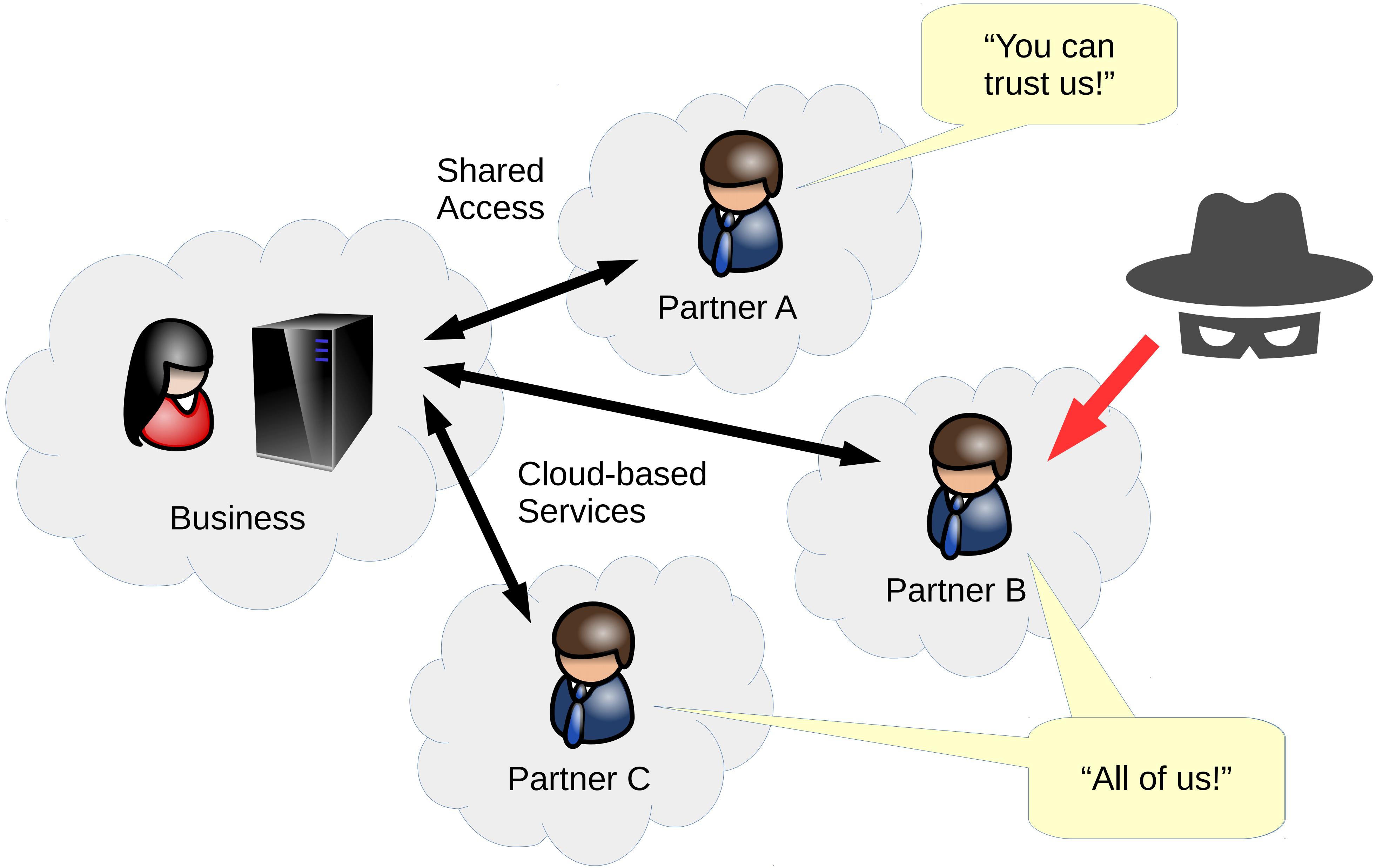


Scaling to bigger systems → weaker security

- Greater chance of any "weak link" breaking



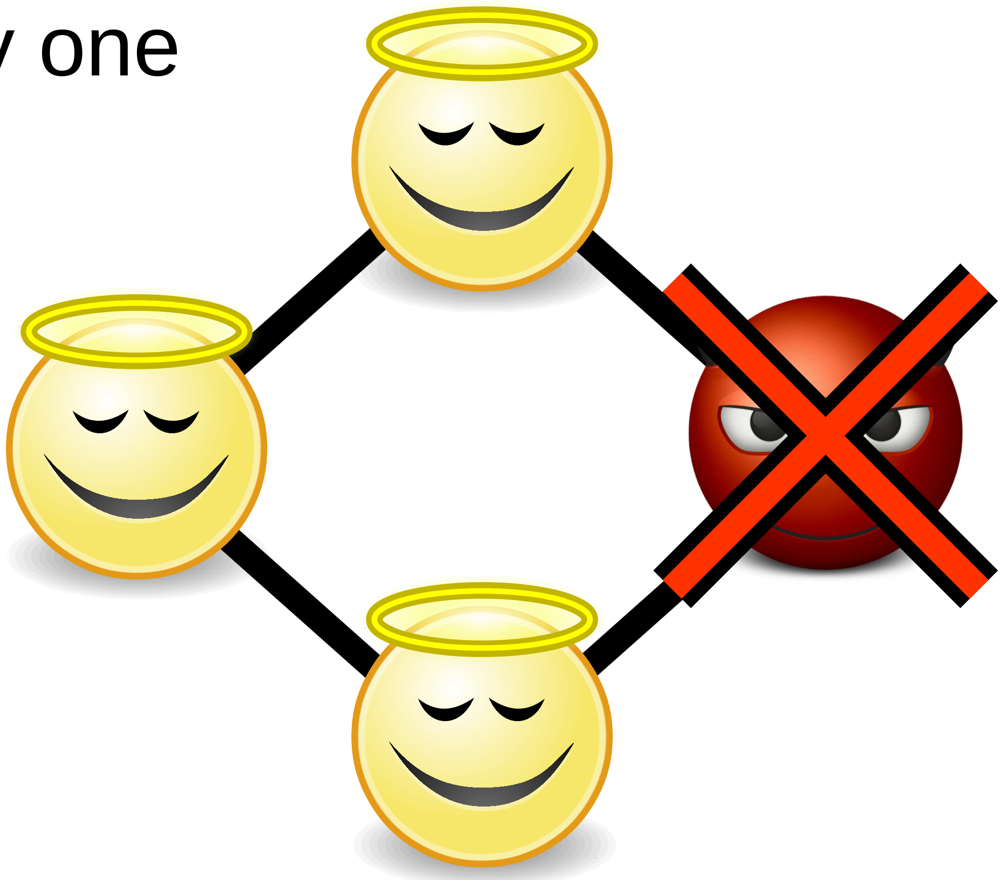
# More interconnection, more devices, more data sharing → greater risk



# The Distributed Trust Principle

Certain 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, …)

# The Distributed Trust Principle

Certain 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, …)

# Dimensions of Information Security

We usually want *three* orthogonal properties:

1. **Integrity:** the system computes honestly, remembers and results correctly
2. **Availability:** it’s there when you need it, provides answers in reasonable amount of time
3. **Privacy:** it doesn’t leak confidential information to anyone who isn’t supposed to have it

Challenge: Information sharing & replication helps integrity & availability but *works against* privacy

# Outline

- Introduction: the distributed trust principle
- **Bitcoin blockchain: principles and operation**
  - Decentralization via public ledgers
  - Privacy via cryptographic identities
  - Scarcity via proof-of-work
- Bitcoin limitations and design alternatives
  - Mining vs Permissioned, Proof-of-Stake, etc.
  - Scalability challenges and potential solutions
- Smart contract systems: Ethereum
- Privacy approaches: Zcash, Chain-Secrets

# Bitcoin (2008)

First successful decentralized cryptocurrency

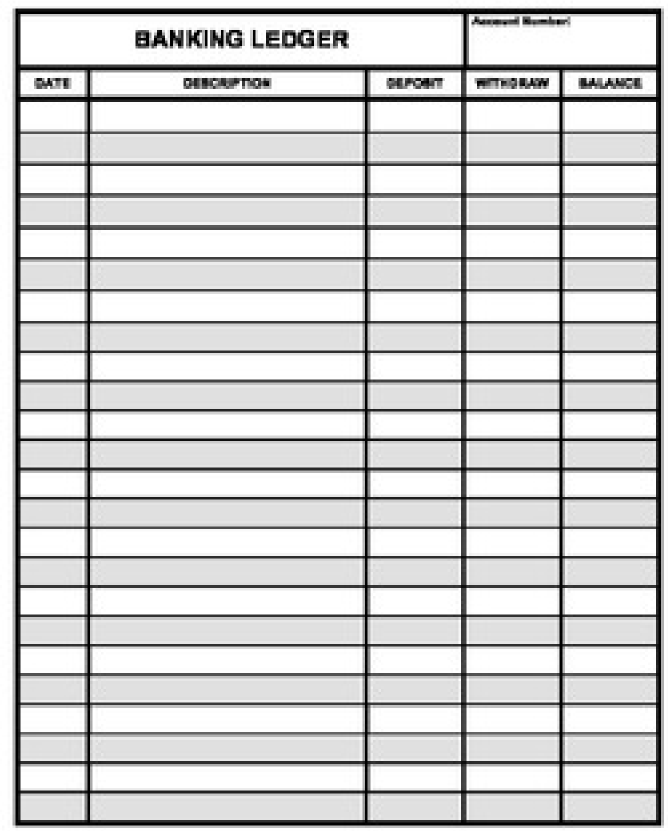
# Two Basic Foundations for Money

**Things**

- Gold, beads, cash...


**Ledgers**

- Who owns what?


BANKING LEDGER

Account Number:

| DATE | DESCRIPTION | DEPOSIT | WITHDRAW | BALANCE |
|---|---|---|---|---|

# 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)*

# Expectations of Things vs Ledgers

**Things:**

- **Decentralized:** anyone can trade
- **Untraceable:** can trade in private
- **Physical scarcity:** value in the object

**Ledgers:**

- **Centralized:** needs authority, e.g., bank
- **Traceable:** bank sees all transactions
- **Fiat scarcity:** value defined by bank

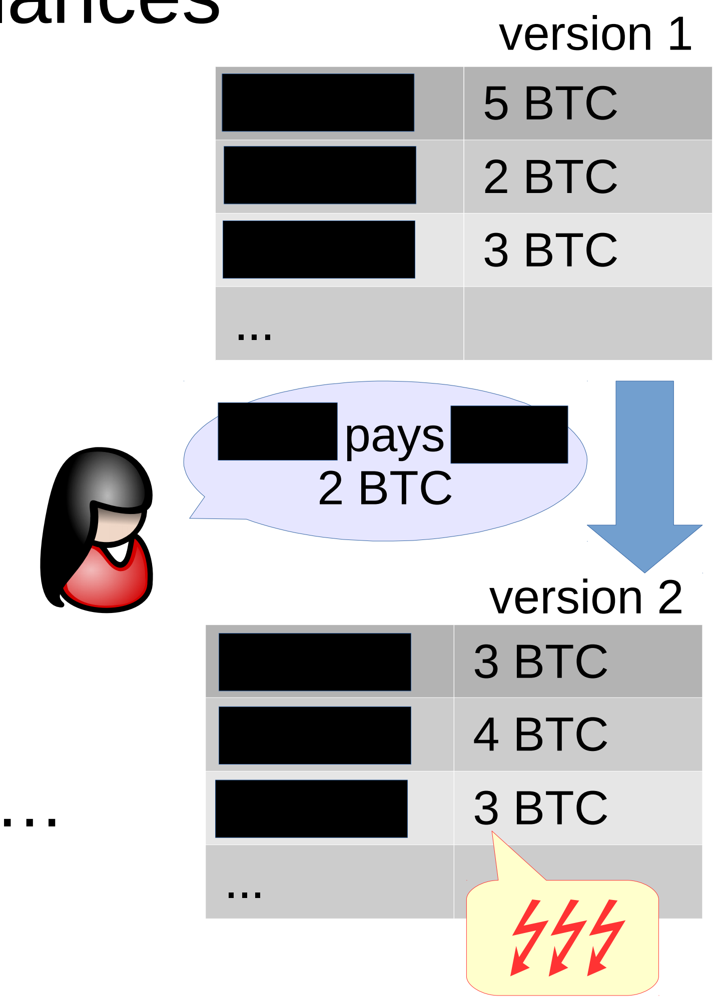
**First key concept:**

Bitcoin is actually a **ledger** that cleverly mimics the expected **properties** of physical currencies.

# Introducing Strawman Bitcoin

Assume there is a “Bitcoin Authority” in the sky

- Keeps a ledger of everyone's balances
  - Perfectly honest, trustworthy
- Allows anyone to trade, any time
  - Simply by calling out to authority
- Keeps all identities secret
  - Like a Swiss bank
- Will trade BTC for **energy**
  - Electricity, gas, whatever you want…
  - Always delivers, like Santa Claus



# From Strawman to Real Bitcoin

One step at a time:

1) From **centralized** ledger managed by authority
   to **decentralized** ledger managed by many

2) From **public** account names/identities
   to **private** cryptographic identities

3) From **authority-backed** scarcity
   to **physical scarcity** via proof-of-work

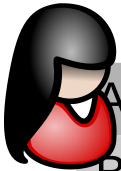
# Decentralized Public Ledgers

**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**

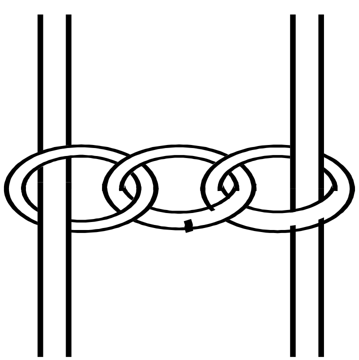
| | |
|---|---|
| Alice | 5 BTC |
| Bob | 2 BTC |
| Charlie | 3 BTC |
| ... | |

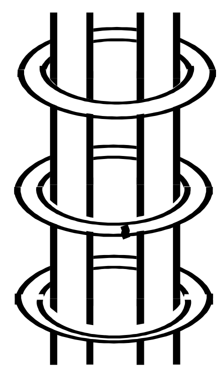
# Properly-Designed Blockchains Eliminate Single Points of Compromise

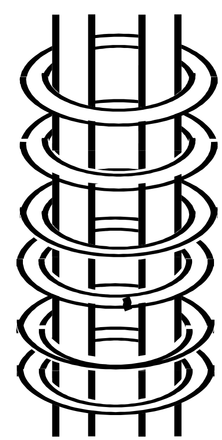
Weakest-link Security:
T = 1

Strongest-link Security:
T = 2-10

Collective Security:
T = 100s,1000s







T: threshold of compromised parties to break security

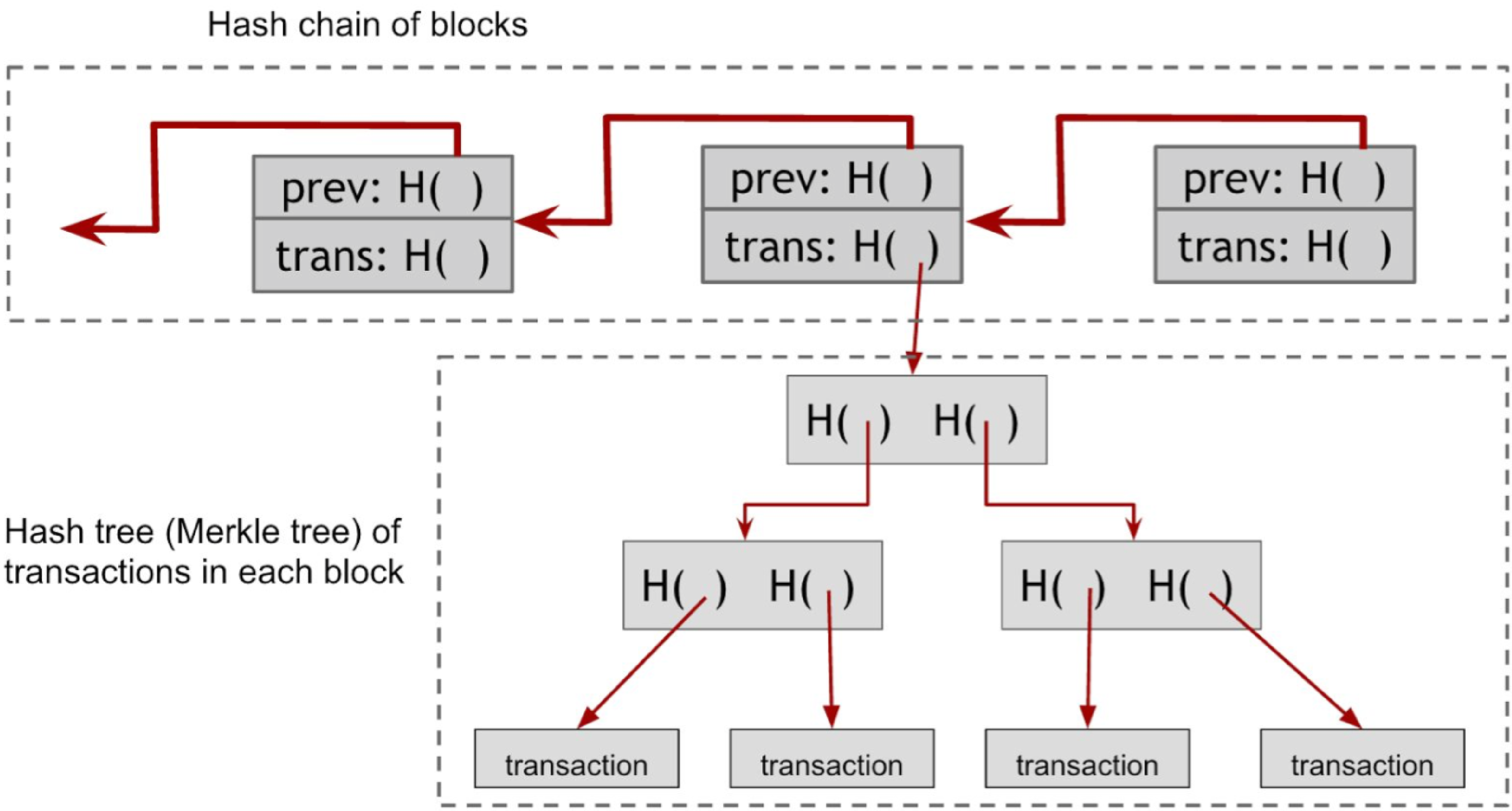
# The Bitcoin Blockchain

How to ensure structural consistency of ledger?

**Tamper-evident logging:**

- Everyone starts with **same ledger state**
- Everyone keeps **complete record of changes**
  - **Blockchain:** log of **all** transactions ever processed
- Everyone independently **checks all changes**
  - **Block:** batch of new transactions to apply to ledger
- Everyone arrives at **same account balances** after any valid set of transactions (blocks)

# The Bitcoin Blockchain



# Decentralized Consensus

Who decides **what changes to make** to log?

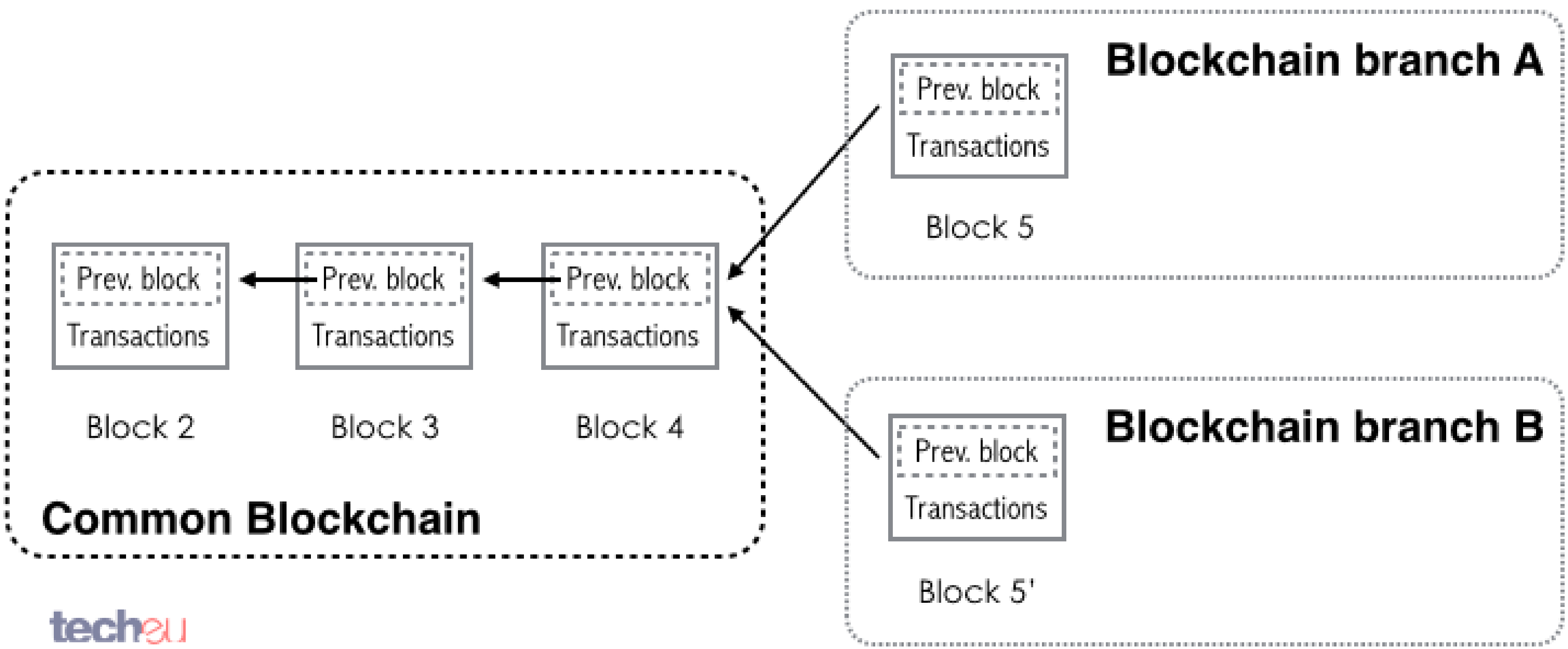
**Bitcoin miners** who process transactions


- Each miner "plays a lottery" constantly, gets series of randomly-numbered "tickets"
- Each ticket has small probability of "winning"
- Each successive lottery winner earns the right to append **one new block** to the blockchain
  - One block may contain **many transactions**
- By design, one lottery winner every ~10mins

# Bitcoin consensus is probabilistic

What if two miners win at **about the same time**?

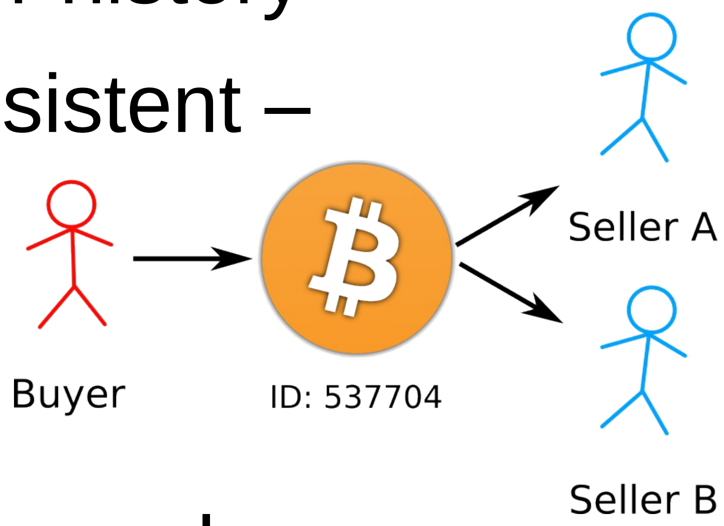
The blockchain **forks**:



# Forks and Double-Spending

Forks would be **disastrous** if they persisted

- Example: Bob has 2 BTC in his account
- Pays 2 BTC to Alice in one branch of history
- Pays 2 BTC to Carol in the other history
- Both histories are *internally* consistent – but which one represents "the truth"??




Bitcoin's answer: **it doesn't matter**, as long as forks are resolved **soon** one way or the other

# Resolving Temporary Forks

How are forks resolved?

Probabilistically, via the **longest chain rule**:

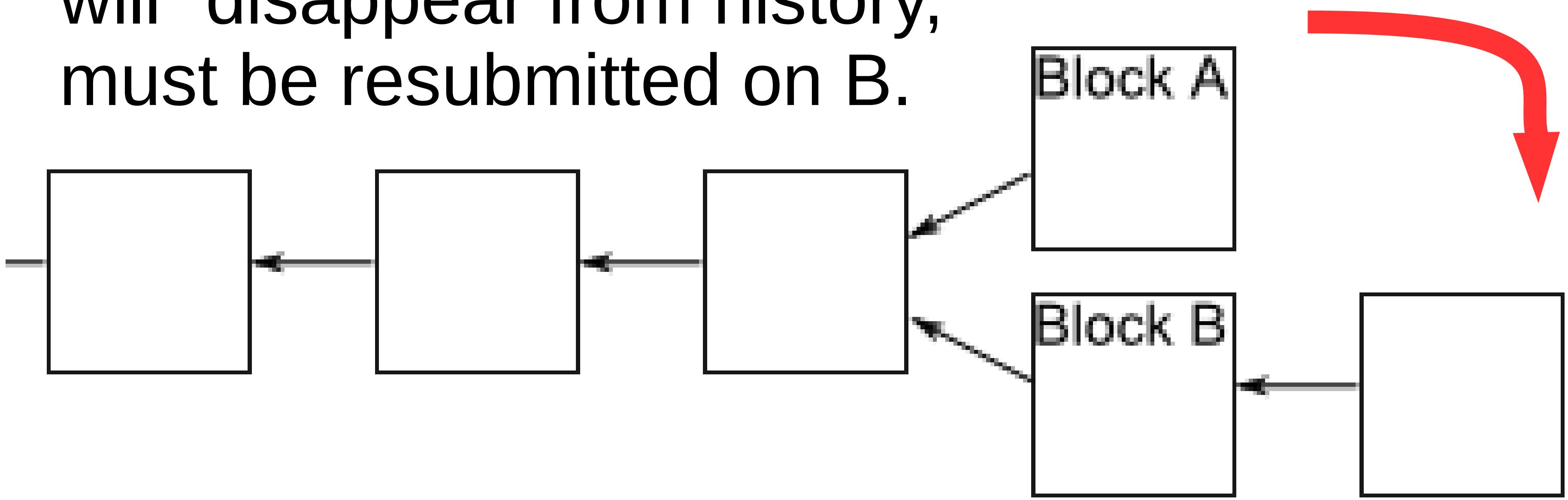
- All miners try to extend longest blockchain, **regardless** of "where it came from" or "how"
- While two equal-length blockchains exist, neither of them "wins" immediately
  - Some miners work on one, others work on other
- But then *some* miner wins, extend one fork
  - Other miners immediately "jump" to winning fork

# Resolving Temporary Forks

**Example:**

As soon as a miner “wins” a ticket to extend B, miners working on A “jump ship” to B's history.

- Any transactions only appearing in block A will “disappear from history,” must be resubmitted on B.



# Implication of Temporary Forks

If someone pays you some Bitcoin,
how do you **know** you've received it?

- **You don't**, at least not immediately!
  - They could have **also** paid the same coin to someone else on a different history
  - If the other history eventually wins, you lose
- **Solution 1:** you wait, and keep checking.
  - Every 10 minutes you wait, it becomes exponentially less likely you could lose
- **Solution 2:** you trust a Bitcoin exchange site
  - Who “promises” to serve you, wait on your behalf

# How Secure is Bitcoin Consensus?

Depends on **who has the mining power**

- **51% attack:** any party, or colluding group, with >50% hash power could “double spend”
  - “Spend like a sailor” on the main history
  - Build an alternate history with coins unspent
  - Eventually reveal the longer, alternate history
- **Selfish mining:** smaller pools of mining power (e.g., ~33%) could use more complex attacks

Mining pools (e.g., Ghash.io) have exceeded 50%

- But apparently didn't “choose to” attack

# From Strawman to Real Bitcoin

One step at a time:

1) From **centralized** ledger managed by authority
   to **decentralized** ledger managed by many

2) From **public** account names/identities
   to **private** cryptographic identities

3) From **authority-backed** scarcity
   to **physical scarcity** via proof-of-work

# How Bitcoin Identifies Users

Not with “names” but with **public signing keys**

- Each user creates a **public/private key-pair**
- Just large random numbers, not “identifying”
  - Any user “could have” picked **any** key-pair
  - It's just **highly improbable** anyone else picks yours

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

| | |
|---|---|
| NZEXOoBrAJhNRpfaLbv7kmsesuUnA1 | 5 BTC |
| gsx7rA8dNoxbxfQrKO11uQTfprLV/m | 2 BTC |
| BvkrSa2Bp52SS6Ee53hre0jycSB6jY | 3 BTC |
| ... | |

- **Public key** identifies account, **receives funds**
- **Private key** confers ability to **spend funds**

# Bitcoin “Privacy”

Does Bitcoin offer **privacy**?

**Kinda:**

- You don't have to tell anyone your “real name” or anything but your **public key**

**On the other hand:**

- The **complete history** of every coin is fully traceable in the public ledger
  - If a criminal is busted and their public key revealed, anyone can tell if you're spending “dirty Bitcoin”
- The **transactions** you perform can reveal a lot of information about your identity

# From Strawman to Real Bitcoin

One step at a time:

1) From **centralized** ledger managed by authority
   to **decentralized** ledger managed by many

2) From **public** account names/identities
   to **private** cryptographic identities

3) From **authority-backed** scarcity
   to **physical scarcity** via proof-of-work

# Bitcoin's Proof-of-Work Lottery

Recall: miners get to **extend the blockchain** by winning a “lottery ticket”


Who “prints” lottery tickets? Miners!

- Create a **candidate block** containing some transactions and a counter: 1, 2, 3, …
- Compute a **cryptographic hash** of the block: → a non-reversible, random-looking “ticket”
- If the ***k*** first bits of the hash are all zero, **win!**
  - **0000000000**d13d8a5aa62ce042f2f714
  - Happens with probability $\mathbf{1/2^{k}}$

# How Bitcoins are Created

The initial **genesis block** contained no bitcoins!

But each time a miner extends the blockchain by winning a lottery, he also creates new Bitcoin

- Normal transactions must be “conservative”: ***N*** input coins → no more than ***N*** output coins
- But one non-conservative transaction per block: **no input coins** → fixed “reward” ***R*** output coins

Globally agreed-upon rules determine reward ***R***

- Now 25 bitcoins, **halves** every 210,000 blocks
- At most about 21 million Bitcoins can ever exist

# Then what, will miners still mine?

Miners can also charge **transaction fees**

- Most users don't **directly** mine Bitcoin
  - Expensive, no longer cost-effective without ASICs
- Instead, users **submit transactions** to miners
  - Choose some amount to “tip” the winning miner
- Miners can **prioritize transactions** to serve
  - Transactions that “tip well” get included first

Rewards from **transaction fees** gradually replace rewards from **creating new Bitcoin.**

# Applications of Blockchains

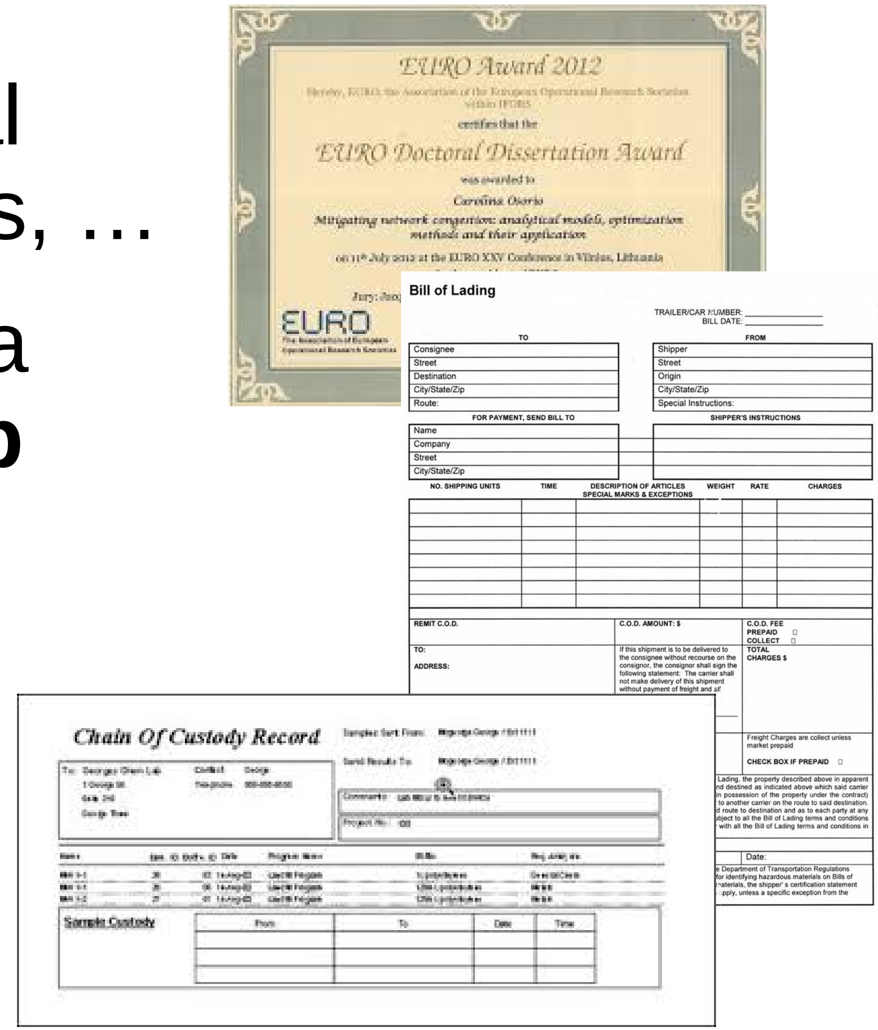
Can represent a distributed electronic record of:

- 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)

# Secure Digital Documents?

Significant interest in digital degrees, awards, land titles, …

- Blockchain can provide a hard-to-forge **timestamp**




But how do you *verify* a digital document?

- Current blockchains: *you must be online*
- Doesn’t work if network down, too slow, costly

# Outline

- Introduction: the distributed trust principle
- Bitcoin blockchain: principles and operation
  - Decentralization via public ledgers
  - Privacy via cryptographic identities
  - Scarcity via proof-of-work
- **Bitcoin limitations and design alternatives**
  - Mining vs Permissioned, Proof-of-Stake, etc.
  - Scalability challenges and potential solutions
- Smart contract systems: Ethereum
- Privacy approaches: Zcash, Chain-Secrets

# Today’s Blockchains Suck

Public/permissionless (e.g., Bitcoin, Ethereum)

- Mining is huge energy waste, and re-centralized
- Unscalable: low throughput, long commit delays
- Limited privacy protection, can’t hold secrets
- Clients must be online and well-connected

Private/permissioned (e.g., HyperLedger, R3, …)

- Weak security – single points of compromise

# How Bitcoin Creates Scarcity

Bitcoin is an **energy-backed currency**:

- Strawman Bitcoin Authority made BTC scarce by promising to **trade it for energy**
- Real Bitcoin makes BTC scarce by making miners **prove they wasted energy**

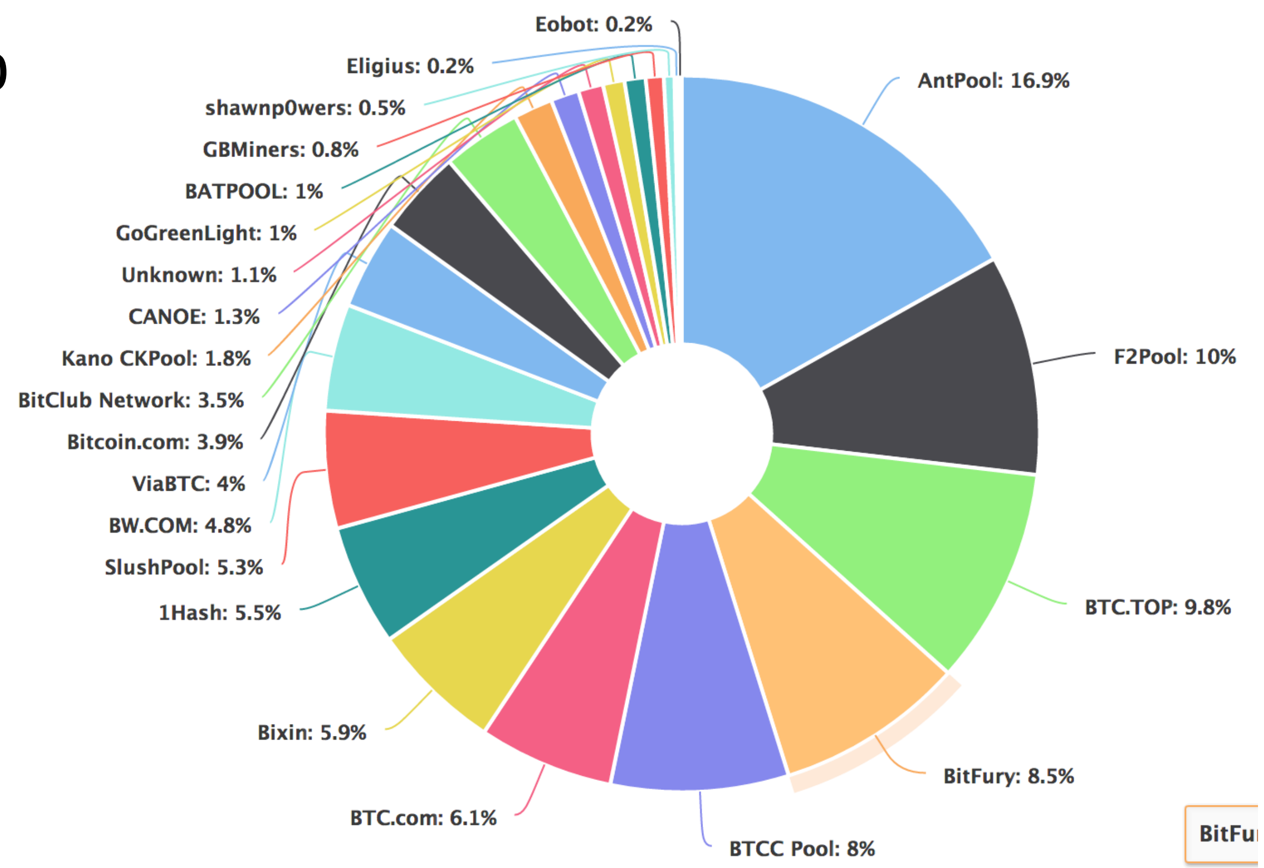
By **proof of work:** solve "computational puzzle"

- Takes lots of CPU cycles (energy) to **create**
- But trivial, cheap for anyone to **verify**

# Bitcoin's Re-Centralization

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.



# “Altcoins” and Blockchain Security

Hundreds of Bitcoin variants created,
mostly with small design tweaks

- **Litecoin:** improves the proof-of-work
- **Ethereum:** adds smart contracts
- **Dogecoin:** just because
- …


Smaller ones can be **attacked**
by anyone with deep pockets
& anti-competitive attitude

# Alternative: 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

# 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…

# Great, but what of it?

Proof-of-stake isn't going to "fix the world."

- Just a more-automated embodiment of the already-ubiquitous **joint-stock corporation**.
- Same vulnerability to takeover by "bigger fish": same as standard hostile takeover issue.
- May put more white-collar workers out of work, help bring on the "automation apocalypse"…

So what else?

# Democratic Blockchains?

**Proof-of-Personhood:** “one person one vote”

- e.g., via Pseudonym Parties [SocialNets ‘08]
- Participants mint new currency at equal rate
  - Decentralized permissionless “basic income”?

# Outline

- Introduction: the distributed trust principle
- Bitcoin blockchain: principles and operation
  - Decentralization via public ledgers
  - Privacy via cryptographic identities
  - Scarcity via proof-of-work
- Bitcoin limitations and design alternatives
  - Mining vs Permissioned, Proof-of-Stake, etc.
  - **Scalability challenges and potential solutions**
- Smart contract systems: Ethereum
- Privacy approaches: Zcash, Chain-Secrets

# Scalability & Efficiency Challenges

- **Speed:** ~10+ mins to “confirm” transaction
  - Decreasing “much” would compromise security
- **Cost:** mining has become highly specialized
  - Only those with ASICs, cheap energy can play
- **Scalability:** max rate ~10 transactions/sec
  - Larger blocks make mining even more expensive
- **Storage:** full nodes keep all transactions ever
  - Participants forget history “at own risk”

# Scaling Blockchains is Not Easy

# Blockchain Scaling Approaches

**Avoid the problem:**

- Move more work off-blockchain (Bitcoin)
  - Shifts burdens onto users, “trusted” intermediaries
- Tweak tuning parameters (Ethereum)
  - Limited headroom, reduced security margins
- Small, semi-closed groups (Ripple, Stellar)
  - Lose openness, public transparency benefits

**Solve the problem:**

- Rethink architecture (Bitcoin-NG, DEDIS ByzCoin)
  - Technically hard but best long-term solution

# Better Performance & Scalability

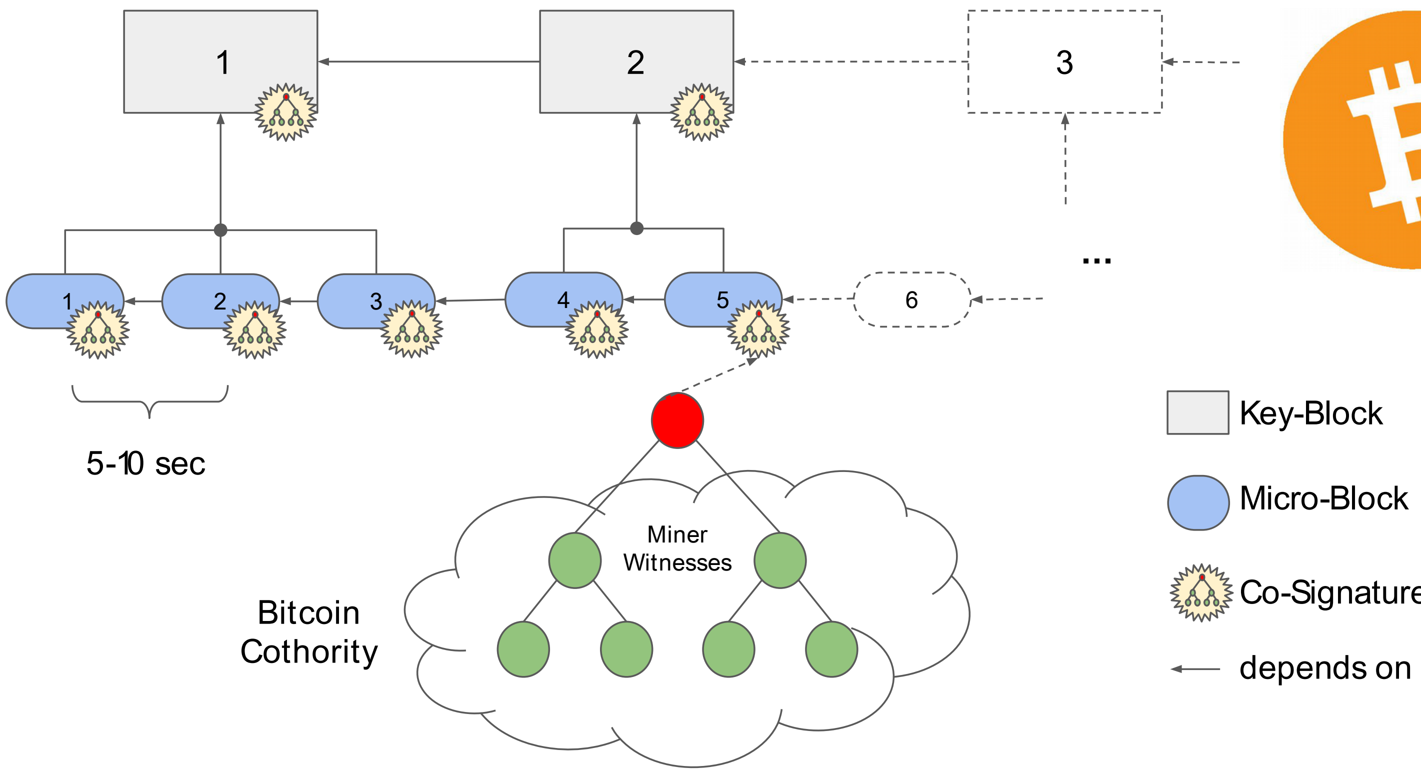
Faster processing, more transactions/second:

- **Bitcoin-NG** (Cornell): miners can commit many “micro-blocks” between mined “macro-blocks”
- **Micropayments** (ETHZ): efficient support for many small “off-blockchain” transactions
- **Cothorities** (EPFL): uses “collective signing” to validate large blocks quickly/efficiently
  - <5 sec for thousands of nodes to collectively sign
  - Users just check collective signature, quick & easy
  - Strong consistency: no forks, even temporarily
  - Preliminary experiments suggest 2,500+ TPS

# ByzCoin: Fast, Scalable Blockchains

DEDIS lab project presented in [USENIX Security ‘16]

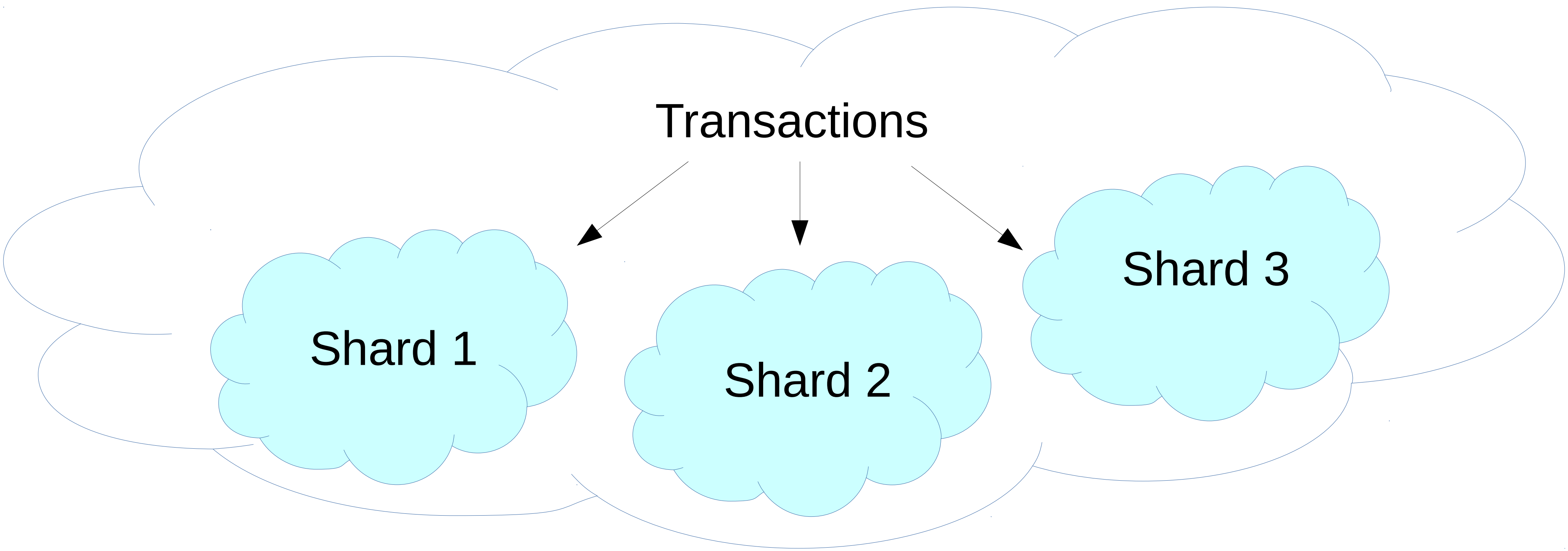
- **Permanent** transaction commitment in **seconds**
- **700+ TPS** demonstrated (100x Bitcoin, ~PayPal)
- **Low-power** verification on light mobile devices



# Application: Blockchain Sharding

**OmniLedger: A Secure Scale-Out Ledger** [preprint]

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



# Bitcoin's Security/Usability Problem

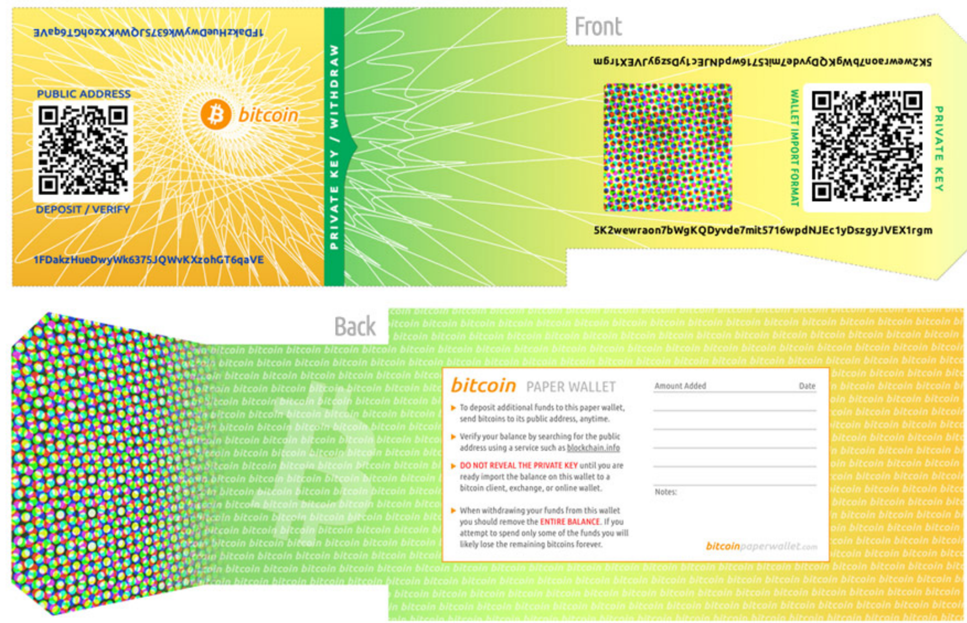
Even if the blockchain is secure, your money isn't!

- The most convenient/feature-rich ways to use Bitcoin are via less-secure Web “exchanges”
  - Ask you to “trust them” but frequently compromised


# “Secure Wallet” Approaches

## Specialized Devices

## Paper Wallets




Potential opportunity for hybrid currencies, “smart banknotes”?

# Outline

- Introduction: the distributed trust principle
- Bitcoin blockchain: principles and operation
  - Decentralization via public ledgers
  - Privacy via cryptographic identities
  - Scarcity via proof-of-work
- Bitcoin limitations and design alternatives
  - Mining vs Permissioned, Proof-of-Stake, etc.
  - Scalability challenges and potential solutions
- **Smart contract systems: Ethereum**
- Privacy approaches: Zcash, Chain-Secrets

# Basic Smart Contracts

Bitcoin already supported limited “scripting”

- Transactions may contain **executable program**
- Determines **conditions** required to spend coin
- Miners **run the script** to validate a transaction that spends the scripted coin

Example:

- Place coins in an “account” whose balance can only be spent if 2 of 3 managers sign-off

# Ethereum: General Smart Contracts

Makes coin-spending scripts **Turing complete**

- Can attach **any computation** to a coin
- May be costly → must **fund** script execution
- Scripts can interact in complex ways

Examples:

- “If stock exceeds **X**, pay to **A**, else pay to **B**”
- “If candidate **X** wins next election…”
- “If newspaper reports **X** assassinated…” !?

# Example: Token Trading Contract

```
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;
    }
}
```

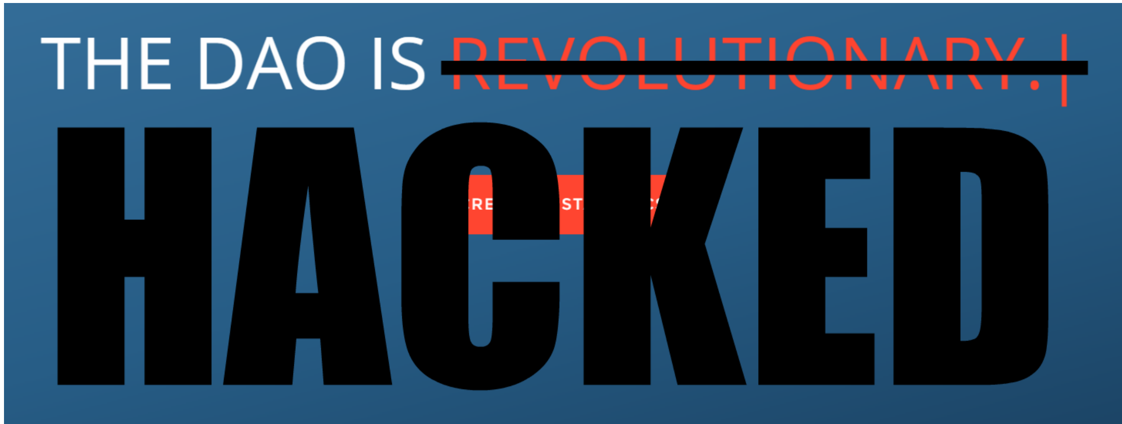
# Smart Contracts (e.g., Ethereum)

Insert arbitrary *software* into a blockchain

- Can programmatically supervise cryptocurrency
  - e.g., automatically settle an insurance payment (see AXA “fizzy” flight delay insurance)

Extremely powerful (and interesting), but risky

- One software bug → spectacular hacks
  - DAO: $70M USD of $150M USD contract stolen in hours (June 2016)



# The “Universal Bug Bounty”

First successful hacker can steal a *lot* of money

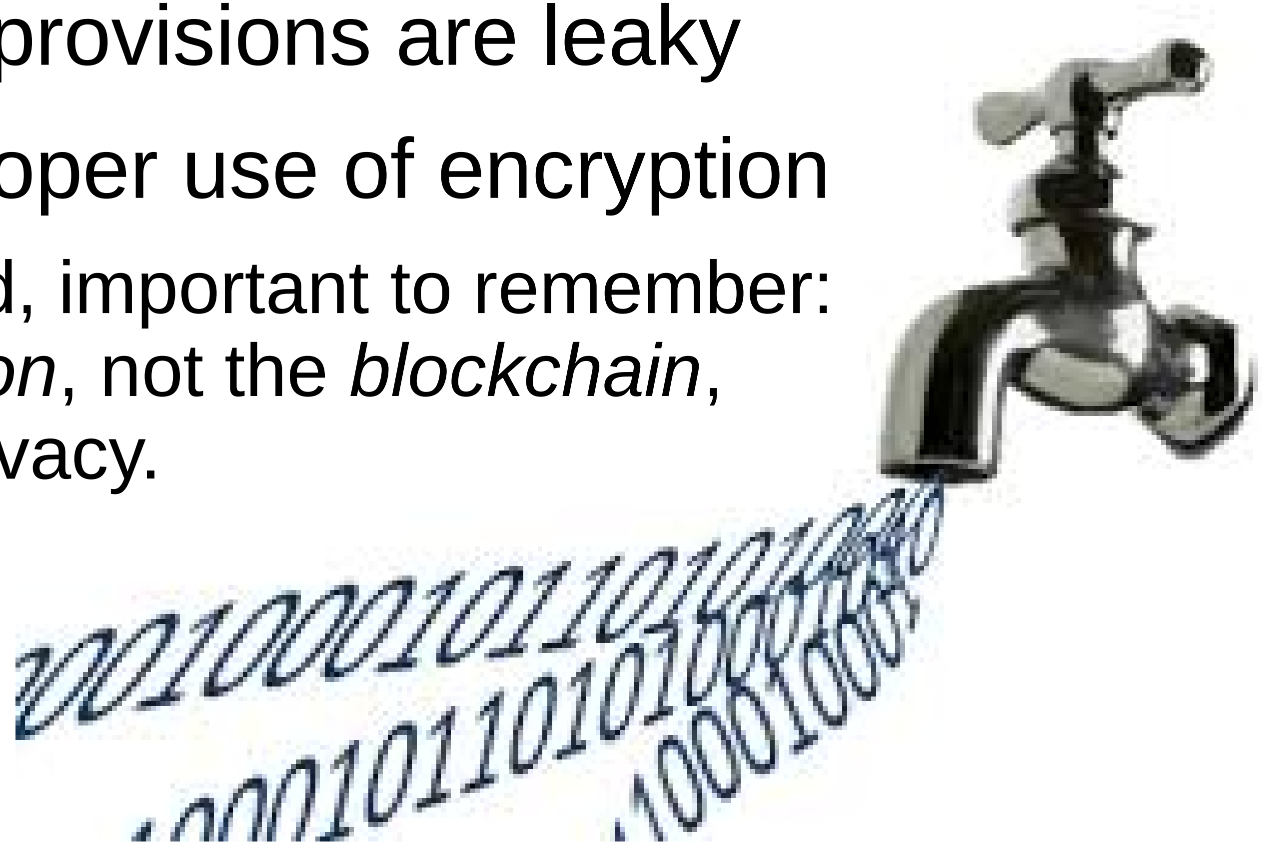
# Outline

- Introduction: the distributed trust principle
- Bitcoin blockchain: principles and operation
  - Decentralization via public ledgers
  - Privacy via cryptographic identities
  - Scarcity via proof-of-work
- Bitcoin limitations and design alternatives
  - Mining vs Permissioned, Proof-of-Stake, etc.
  - Scalability challenges and potential solutions
- Smart contract systems: Ethereum
- **Privacy approaches: Zcash, Chain-Secrets**

# 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.



# Stronger Privacy Protection

Approaches to improve Bitcoin's privacy:

- **ZeroCash:** uses sophisticated cryptographic zero-knowledge proofs to “blind” transactions
  - Anyone can validate the ledger without actually knowing **anything** about transactions processed!
- **CoinShuffle:** simpler, lightweight approach: simply “shuffles” many transactions together
  - No one can tell which inputs/outputs associated with which transactions in a bundle

But these work against accountability (KYC etc)

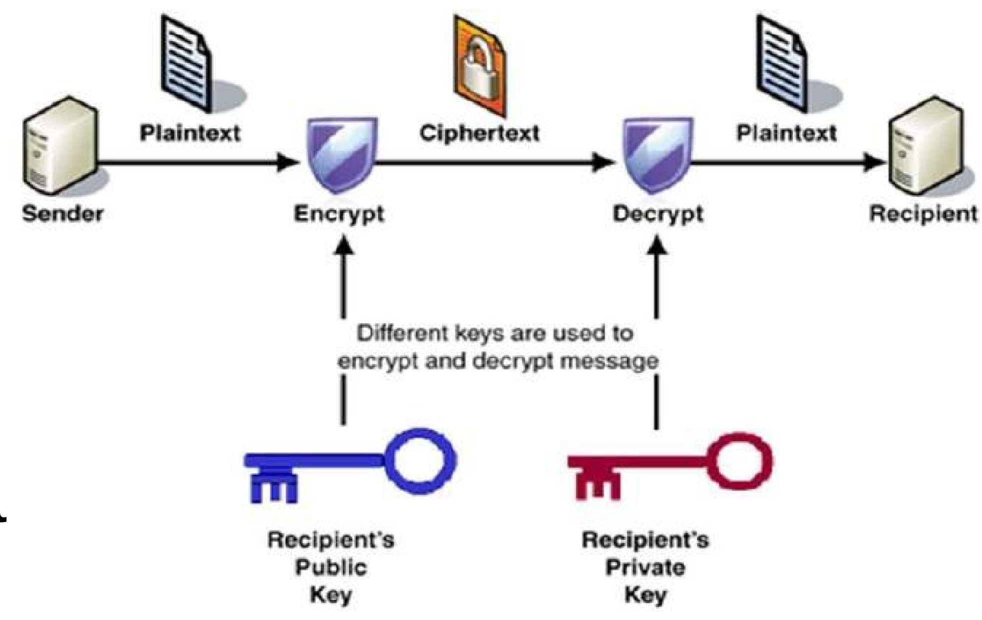
# Blockchains Can’t Keep Secrets

In current blockchains, secrets (keys, passwords) must be held “off-chain” by private parties

- Just a hash on-chain → document might be lost
- Encrypted on-chain → encrypted to whom?
  - Decided at encryption, *cannot be changed/revoked*

Current blockchains can’t manage secrets, because they would leak to *all* participants

- Weakest-link security a



# Blockchains *Want To* Keep Secrets

Example: secure document management

- Doc encrypted to access list,
  then previous access-holder revoked/fired
  - If user's private key leaked/misused
    → document exposed undetectably, forever!
  - Can't tell whether user ever accessed, may access
- Can't enforce data retention policies


Current blockchains can *hold*
but can't *manage* secrets!

# DEDIS "Chain-Managed Secrets"

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

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

- On-chain policies can determine how and when secrets used, who should have access when
  - Any access change immediately, atomically applied
- Can *re-encrypt* secrets to authorized parties
  - Tamper-proof log of all uses or attempted uses
- Can enforce document retention/deletion policies

# Conclusion

Bitcoin established first “practical” cryptocurrency

- Makes a **ledger** mimic the properties of **cash**
- **Decentralization** by replication, consensus
- **Privacy** via public keys to identify accounts
- **Scarcity** via proof of work, i.e., wasted energy

Many limitations, but many improvement activities

Might (or might not) decentralize, revolutionize finance, monetary policy, contracts…