CAS Module 1: Days 1+2, Zürich, Fridays, September 4 and 11, 2020

Blockchains

Prof. Dr. Burkhard Stiller[1], Bruno Rodrigues[1], Christian Killer[1], Dr. Thomas Bocek[2]

With many thanks addressed to M. Franco, G. Parangi, S. Rafati, E. Scheid, and Dr. E. Schiller – all @ CSG[1]

[1]Communication Systems Group CSG, Department of Informatics IfI
University of Zürich UZH
[2]Axelra AG und Fachhochschule OST

Universität Zürich[UZH]

[stiller¦rodrigues¦scheid]@ifi.uzh.ch
thomas.bocek@axelra.com

CSG Communication Systems Group

ifi

Overview – CAS in Blockchain

Module 1: "Blockchains"

- **Fri 4.9.202 1.0 Introduction to Blockchains, Prof. Dr. Burkhard Stiller**
- **Fri 11.9.2020 1.0 Blockchain Platforms and Architectures, Prof. Dr. Burkhard Stiller, Dr. Thomas Bocek**
- Fri 18.9.2020 1.0 Smart Contracts, Prof. Abraham Bernstein, Ph.D., Dr. Daniele Dell'Aglio
- Thu 24.9.2020 Written Exam (on-line)

Module 2: "Blockchain Business and Economics"

- Sat 26.9.2020 1.0 Enterprise Blockchains, Prof. Dr. Gerhard Schwabe
- Sat 3.10.2020 1.0 Cryptocurrencies, Prof. Dr. Thorsten Hens
- Fri 9.10.2020 1.0 Applications of Public Blockchains, Prof. Dr. Helmut Dietl
- Sat 10.10.2020 1.0 Blockchain Cryptoeconomics and Analytics, Prof. Dr. Claudio J. Tessone
- Fri 16.10.2020 Hand-out Written Assignment

Module 3: "Blockchain Regulation and Law"

- Fri 16.10.2020 1.0 Token Regulations and ICOs, Prof. Dr. Rolf H. Weber
- Sat 17.10.2020 1.0 Cryptocurrencies and Smart Contracts, Prof. Dr. Rolf H. Weber, Prof. Dr. Peter Picht
- Sat 24.10.2020 1.0 Data Protection and Property Law, Prof. Dr. Florent Thouvenin, Dr. Alfred Früh
- Thu 29.10.2020 Written Exam
- Thu 10.12.2020 Hand-in Written Assignment
- Thu 17.12.2020 Graduation Party and Handover of Certificates

Outline Day 1 – Introduction to Blockchains

Part 0: Administrative Overview

Part I: Introduction and Background

1. Distributed Systems
2. Overlay Networks and Peer-to-peer Systems
3. Security and Trust

Part II: Blockchain Basics (1)

4. Principles of Blockchains
5. Blockchain Operations

Lunch Break

Part III: Blockchain Basics (2)

6. Blockchain Eras and Application Domains
7. Consensus Mechanisms
8. Blockchains and Quantum Security

Part IV: Cryptocurrencies (Technical Perspective)

9. Electronic Payment Systems
10. Bitcoins

Outline Day 2 – Blockchain Platforms and Architectures

Part V: Ethereum

11. Bascis
12. Mechanism and Protocol
13. Smart Contracts

Part VI: The Blockchain World

14. Other Blockchains
15. Blockchain Interoperability

Lunch Break

Part VII: Practical Exercises

16. Ethereum Wallet
17. Smart Contract

Part VIII: Blockchain Evaluations (Technical View, mainly)

18. To Blockchain or not to Blockchain
19. Comparisons and Assessment
20. Challenges and Risks
21. Expected Impacts and Conclusions

Day 1: Introduction to Blockchains

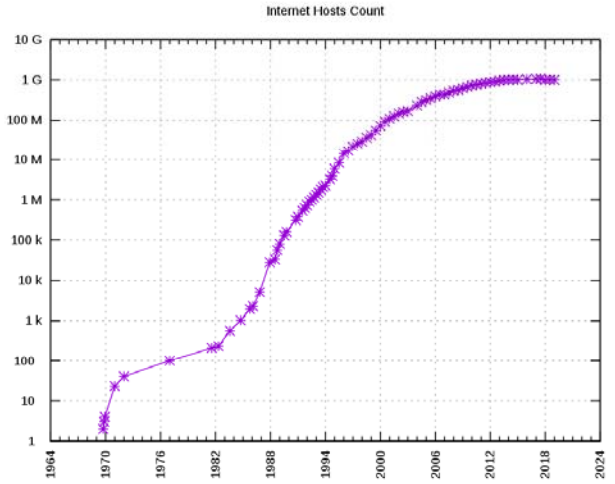
Part I: Introduction and Background

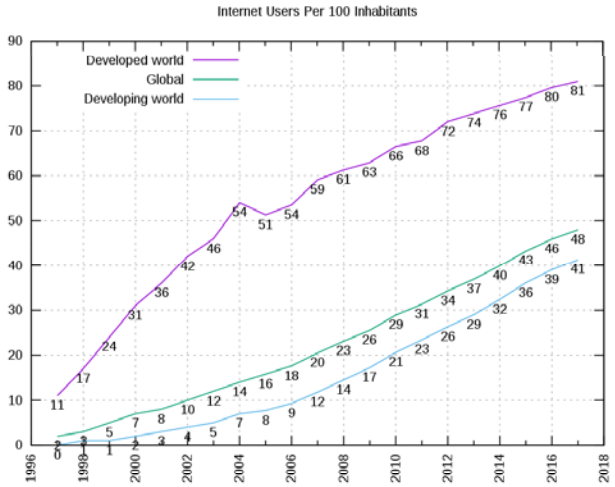
1. Distributed Systems

Today's Systems Scale

- Increasing number of computers
  - Adding devices as for the Internet-of-Things (IoT) ➔ 20+ Billion
- Increasing number of Internet users
- Increasing number of distributed applications
  - Industry 4.0

- All embracing needs are increasingly
  - Complex
  - Larger scale
  - Application-specific





https://en.wikipedia.org/wiki/Global_Internet_usage#Internet_hosts
https://en.wikipedia.org/wiki/Global_Internet_usage

“The” or “a” Definition

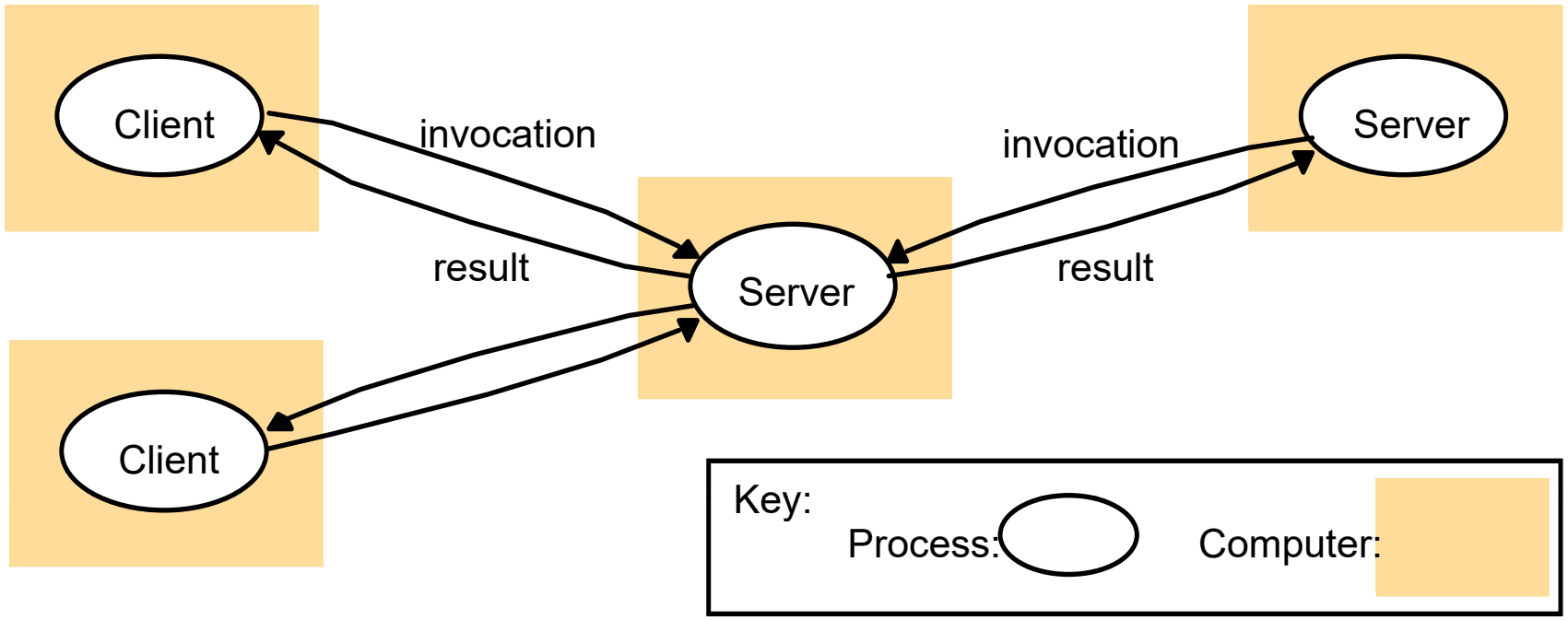
- Distributed Systems come in many flavors!
  - Computers connected by a network and
  - Computers spatially separated by any distance
- Def.: A collection of independent computers that appears to its users as a single coherent system
  - Hardware: All machines are fully autonomous
  - Software: Users think they deal with a single system
- Selected key consequences
  - Concurrency
  - No global clock
  - Independent failures

Distributed Systems – Examples

- Telephony
- Client Server Communications
- Internet
- Automation Networks
- Personal Networks
- Time Stamping Systems
- Overlay Systems/Networks, *e.g.*, Peer-to-Peer Systems
- Cloud Computing
- Blockchains

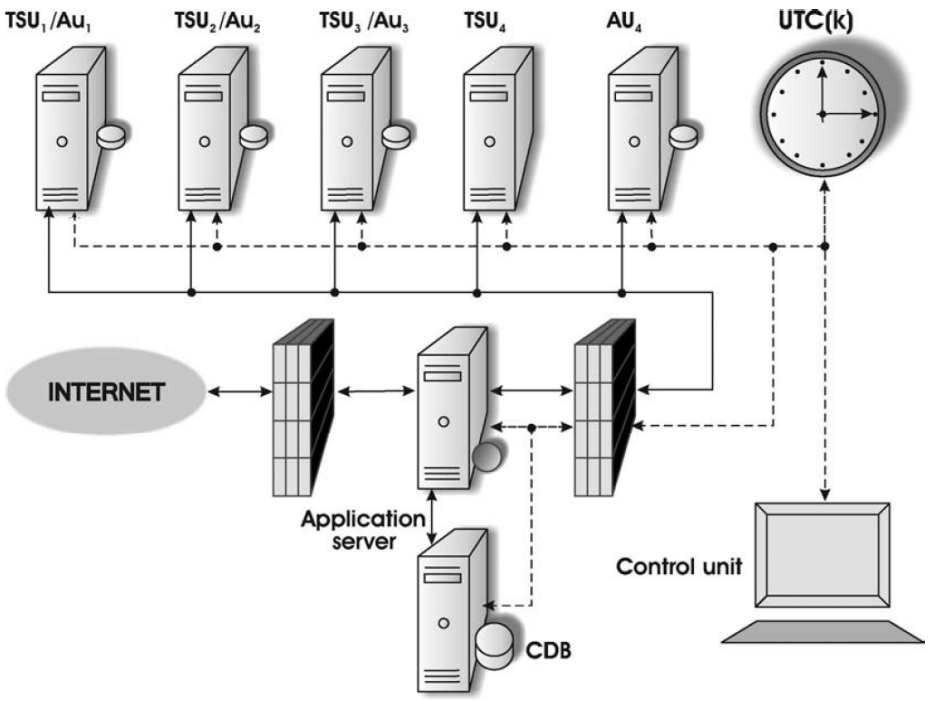
… whatever the future will bring …

Client Server Communications



Time Stamping Systems

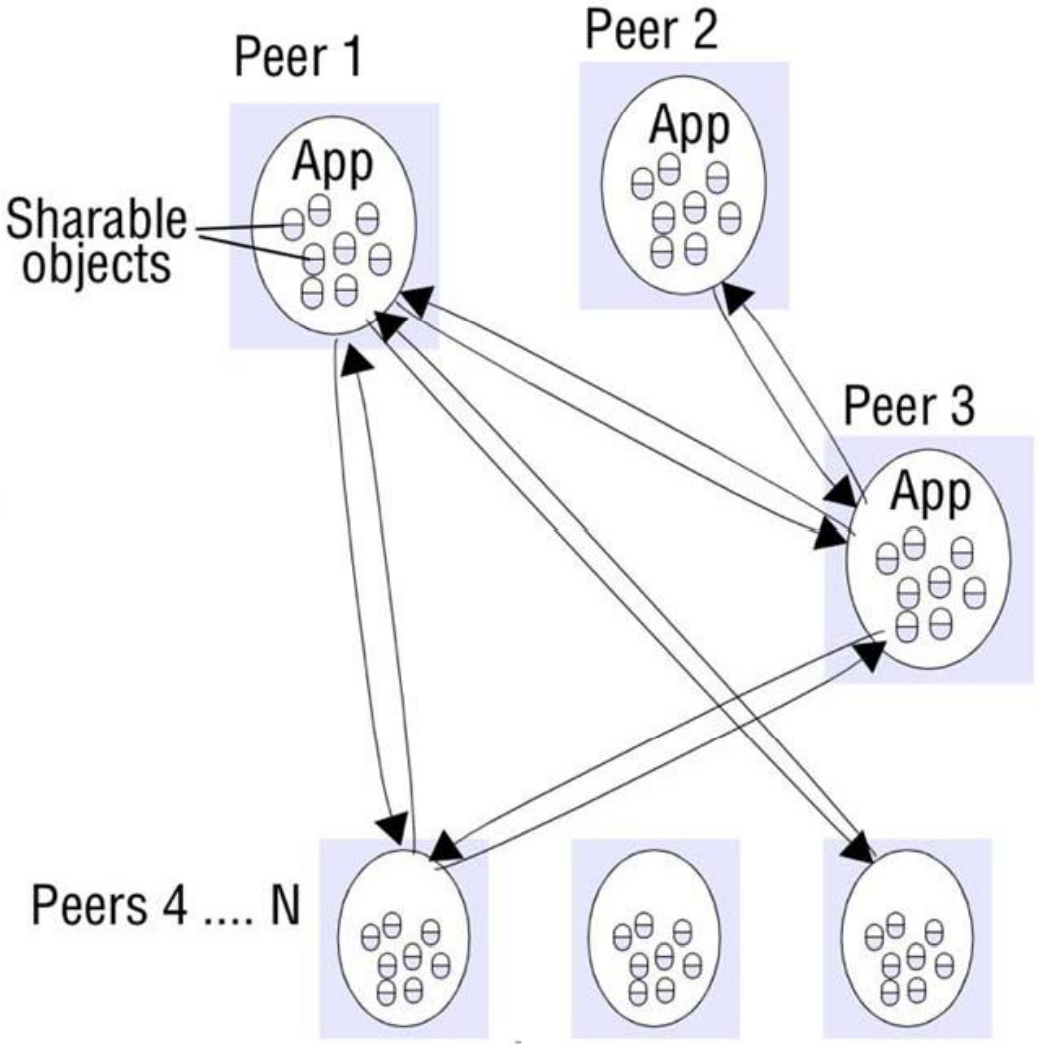
- Timestamp: sequence of characters or encoded information identifying when a certain event occurred
  - Usually with date and time of day, optionally accurate to a small fraction of a sec
- Trusted generation, distribution, and general verifiability required



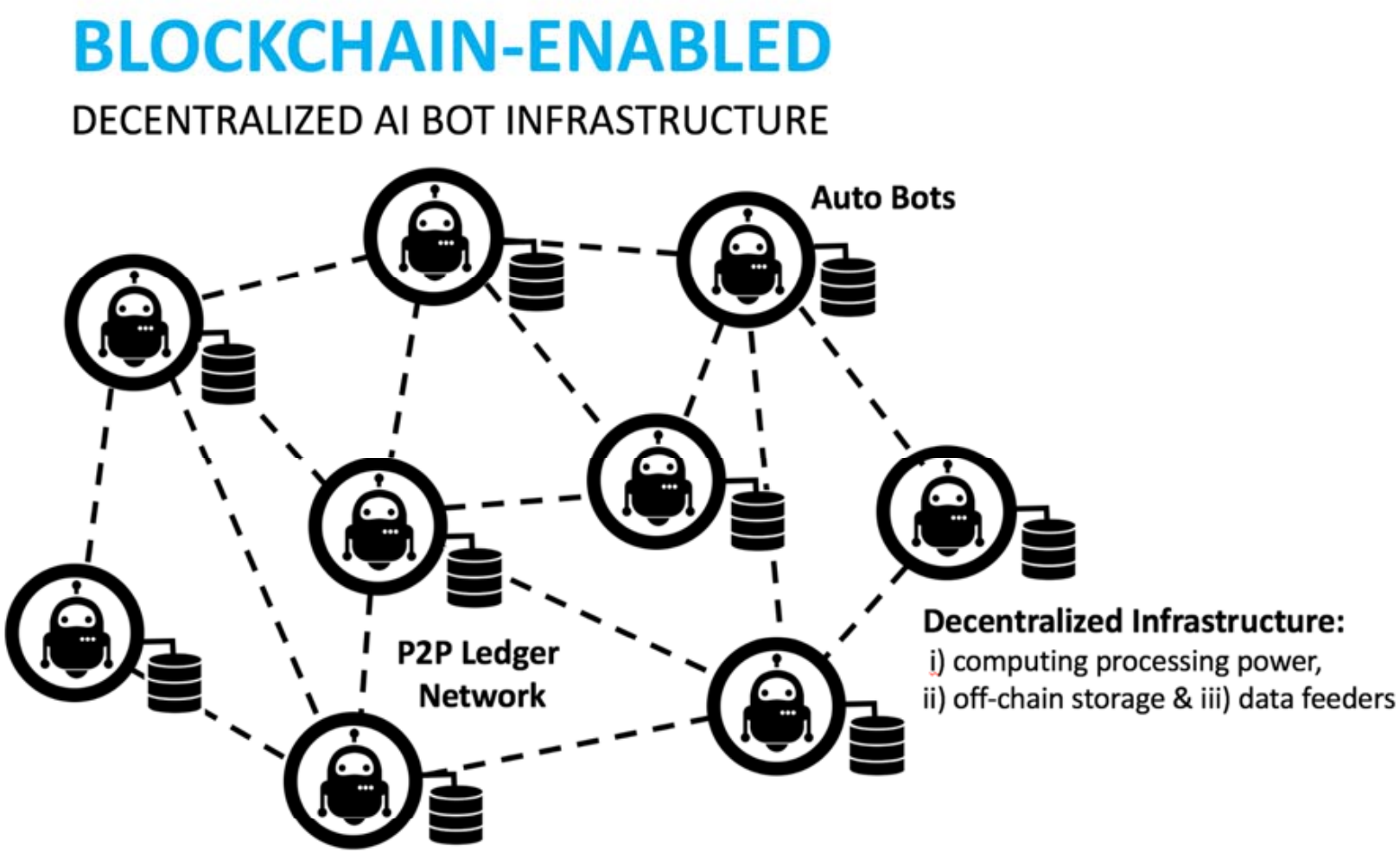
Rimantas Miskinis et al.: Digital Time Stamping System based on Open Source Technologies, 2010

Peer-to-Peer Systems

Distributed resource sharing based on interests



Blockchains



https://medium.com/@GimmerBot/decentralized-artificial-intelligence-and-autonomous-bots-auto-bots-in-distributed-8a85e0d07d14

Characteristics of Distributed Systems

1. Transparency
   - Single view of the system
   - Hide numerous details
2. Heterogeneity
   - Networks
   - Computers (HW)
   - Operating systems (SW)
   - Programming languages
   - Developers
3. Failure Handling
   - Detecting
   - Masking
   - Tolerating
   - Redundancy
   - Recovery
4. Openness
   - Extensibility
   - Publication of interfaces
5. Scalability
   - Controlling the cost of resources
   - Controlling the performance
   - Preventing resources from running out
   - Avoiding performance bottlenecks
6. Security
   - Secrecy, privacy, integrity
   - Confidentiality
   - Authentication, authorization
   - Non-repudiation

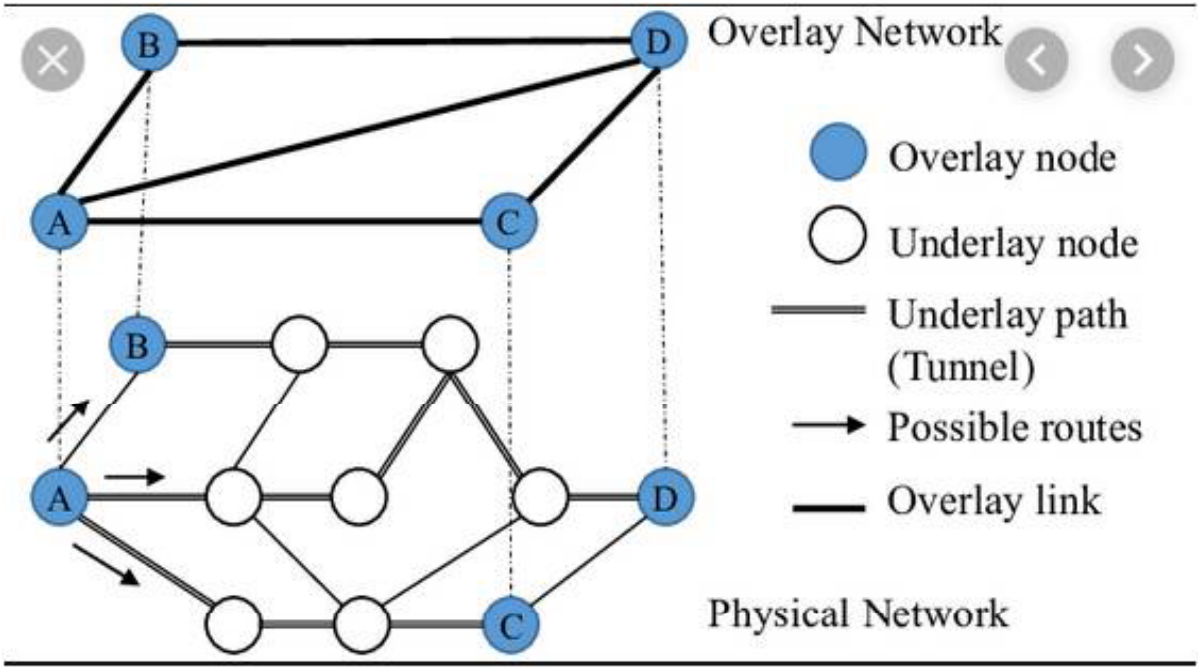
Distributed Systems to Remember

- Distributed systems face very large numbers of
  - Devices, users, applications, …
  - which determine scalability challenges
- Any "participant" – hardware, software, individual – is operated or operates autonomously and transparently
  - There is no central control, no single, real-time, unique state
- Failures happen in a heterogeneous environment and need to be handled (application- or system-specific) such that systems runs securely

2. Overlay Networks and Peer-to-peer Systems

Overlay Networks

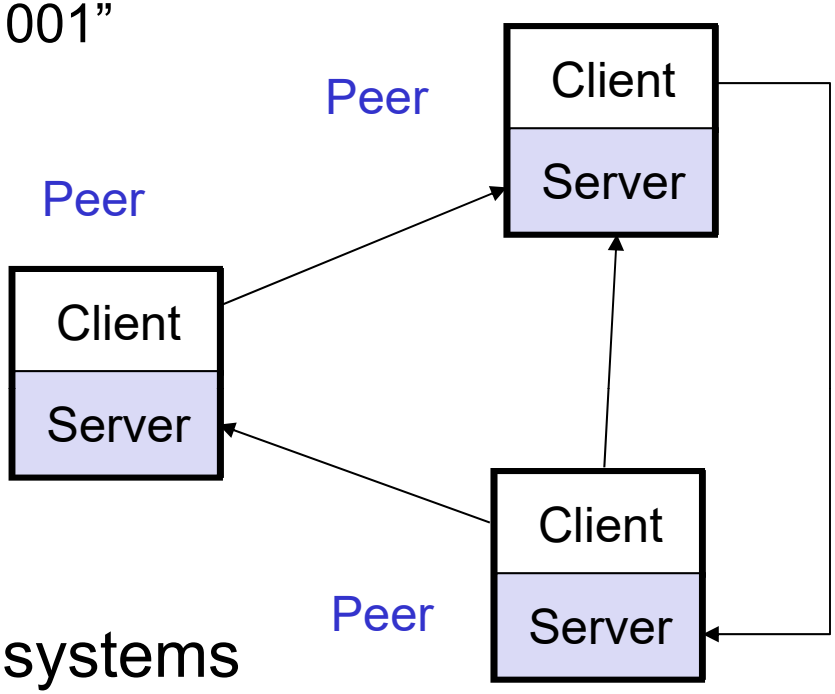
- Overlay Network (ON) definition
  - An ON is a computer network built on the top of any other network, today, typically on top of the Internet (Internet Protocol-, IP-based)
  - Nodes in the ON are connected by virtual or logical links
    - Such links correspond to a path, possibly through many physical links, in the underlay



https://www.researchgate.net/publication/311715170_A_Distributed_Algorithm_for_Throughput_Optimal_Routing_in_Overlay_Networks/figures?lo=1

P2P Systems and Peers

- P2P systems
  - Composed out of direct connections between peers
  - Addressing scheme different than of the underlay network
    - Traditional IP address: “10.2.5.76”
    - P2P “address”: “Steam Engine BR 001”
- Peers
  - Have all the same capabilities
    - Ability to act in any role
  - Can act as ”clients“ and “servers“ at the same time
  - Clear difference to client-server systems



Peer-to-peer (P2P) System Definition

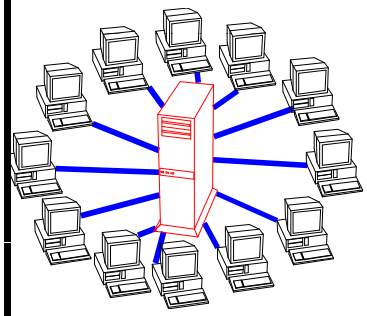
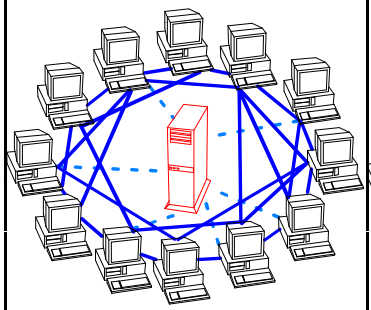
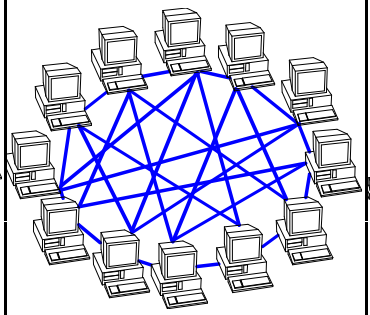
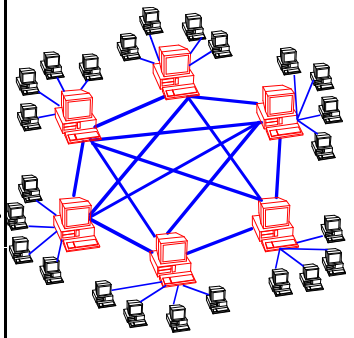
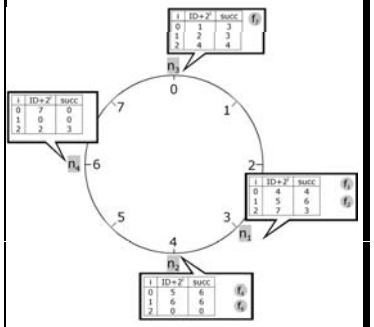
A. Oram: Peer-to-Peer : Harnessing the Power of Disruptive Technologies

1. A Peer-to-Peer (P2P) system is „*a self-organizing system of equal, autonomous entities (peers)* [which] *aims for the shared usage of distributed resources in a networked environment avoiding central services.*“
2. *Peer-to-peer; denoting a network or data communications in which no dedicated server is involved*.

http://dictionary.com

– Derived key characteristics of a P2P system:
  - Equality: All peers are equal (peer = gleichgestellt)
  - Decentralization: No centralized services
  - Self-organization: No coordination from outside
  - Shared resources: Peers use resources provided by other peers
  - Direct interaction: Peers communicate directly with other peers

P2P System Generations

Client-Server	Peer-to-Peer			
1. Server is the central entity and only provider of service and content. → Network managed by the Server 2. Server as the higher performance system. 3. Clients as the lower performance system Example: WWW	1. Resources are shared between the peers 2. Resources can be accessed directly from other peers 3. Peer is provider and requestor (Servent concept)			
	Unstructured P2P			Structured P2P
	Centralized P2P	*Pure P2P*	*Hybrid P2P*	*DHT-Based*
	1. All features of Peer-to-Peer included 2. Central entity is necessary to provide the service 3. Central entity is some kind of index/group database Example: Napster	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities Examples: Gnutella 0.4, Freenet	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → dynamic central entities Example: Gnutella 0.6, JXTA	1. All features of Peer-to-Peer included 2. Any terminal entity can be removed without loss of functionality 3. → No central entities 4. Connections in the overlay are "fixed" Examples: Chord, CAN
				

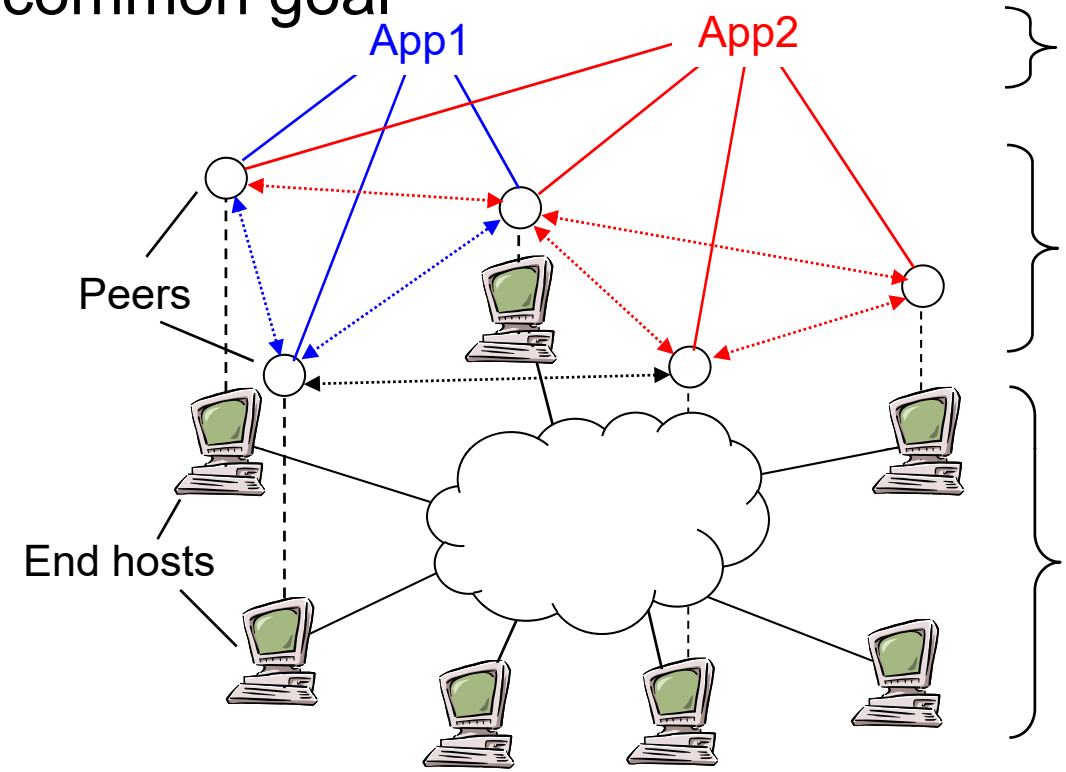
1st Gen. 2nd Gen. Current Gen.

Desirable Properties of an ON/P2P

- High fault-tolerance
  - Churn
  - Connection problems
  - Malicious behavior
- Wide heterogeneity
    - Mobile devices vs. powerful devices, low vs. high bandwidth, battery vs. connected to power source
- Acceptable fairness
  - Each peer should have a similar workload (exceptions!)
- Good scalability
  - Workload proportional to number of peers
  - Ideally, no limit on the number of peers

Overlay Applications

- Overlay Application on top of ON for a specific purpose
  - Drawing on the cooperation of peers in the form of services these peers provide to each other in order to achieve a common goal



P2P Application
E.g., KaZaA, eMule, BitTorrent, Blockchain

P2P Overlay Network
E.g., random, structured, hybrid, centralized

Underlay Network
E.g., Internet

Overlay Networks to Remember

- ON operates on top of underlay
  - Typically the underlay is the Internet (IP-based)
  - ON is determined by informal “rules and regulations”
- P2P system instantiates a dedicated overlay
  - Using its own addressing scheme
  - Applying its own high-level communication protocol for peers
  - “Limited” to participating peers following the overlay rules
- Overlay application
  - Runs within a given P2P system exploiting its functionality

3. Security and Trust

Internet Privacy

- When do people care about Internet privacy?

- What about credentials?
  - 81% of all hacking activities are targeted to obtain credentials (credential harvesting)

https://www.securityweek.com/foundation-cyber-attacks-credential-harvesting

Vulnerability, Threat, and Risk

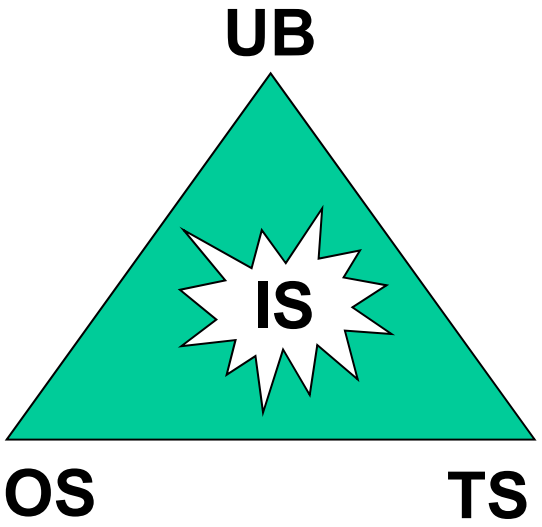
- Vulnerability (Verletzlichkeit):
  - A quality or characteristic of a system that provides an opportunity for misuse.

- Threat (Bedrohung):
  - Any potentially malicious or otherwise occurrence that can have an undesirable effect on assets and resources of an IT system.
- Risk (Risiko):
  - = Threat X Vulnerabilities ***OR*** Likelihood X Impact

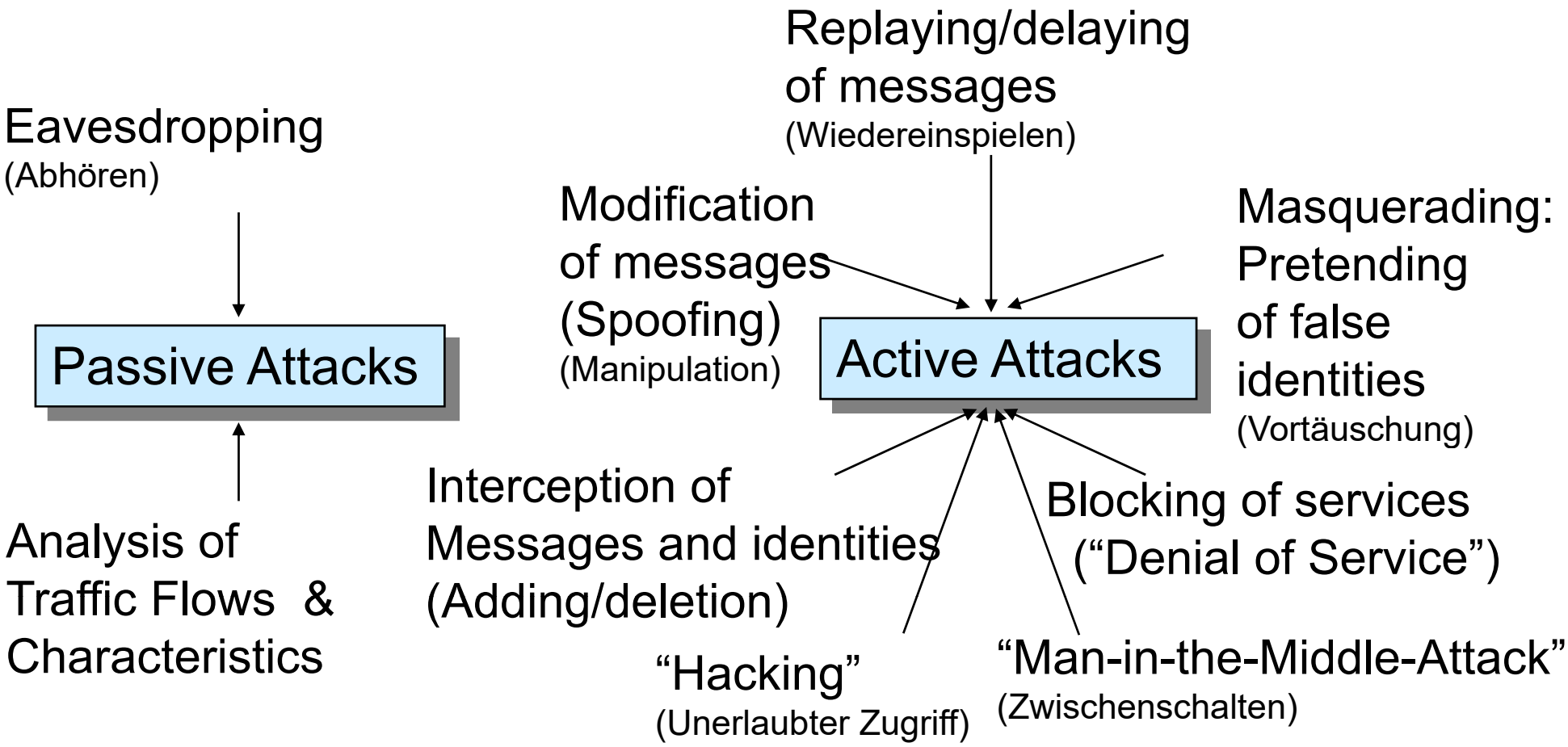
Security defines a process of risk management, supported by a set of suitable (technical, economic, behavioral) measures!

Security Areas

- Organizational Security (OS):
  - Trusted Third Party (TTP)
  - Certification Authority (CA)
    - Access rights (who will be enabled to do what)
    - Key management (distribution of keys)
- Technical Security (TS):
  - Security services, mechanisms, algorithms, …
- User Behavior (UB):
  - Passwords, internal- and external attacks, …
- Information Security/Information System Security (IS):
  - Effect on content, procedures, or the system



Threats and Attacks



⇨ High risk!

Major 7 Security Pillars (1)

- Authentication (Authentifizierung/Authentifikation):
  - Authentication ensures that partners involved in communications can prove that the peer is that it claims to be.
- Authorization (Autorisierung):
  - Authorization ensures that a partner with a known ID is enabled to utilize a service.
- Integrity (Unversehrtheit, Fälschungssicherheit):
  - Integrity provides protection against the modification of a message along a transmission path.
- Privacy (Privatheit):
  - Privacy defines the degree of publication of personal information and data.

Major 7 Security Pillars (2)

- Confidentiality (Vertraulichkeit):
  - Confidentiality protects transmitted data against eavesdroppers in a communication channel ensuring that only an authorized receiver can interpret the message received.
- Non-repudiation (Nicht-Zurückweisbarkeit/Nicht-Abstreitbarkeit):
  - Non-repudiation provides that neither the sender nor the receiver can deny that a communication has taken place.
- Anti-replay protection (Schutz gegen Wiedereinspielung):
  - Anti-replay protection protects a receiver from the duplicated reception of a previously obtained and already authenticated message.

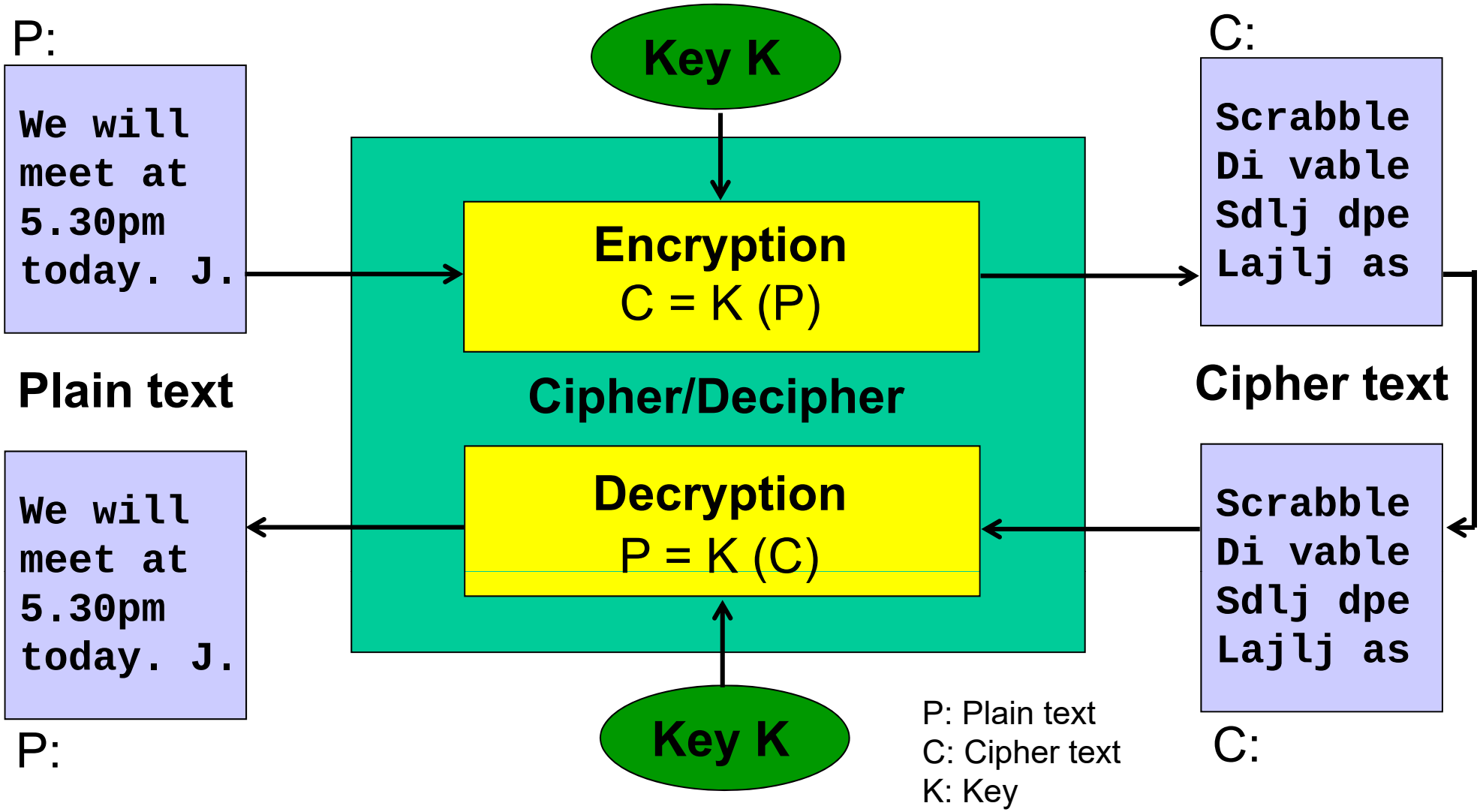
Additional Security Aspects (1)

- System security (Systemsicherheit):
  - To obtain a full-fledged security degree, the entire system, going beyond the communication security, has to be protected by means of security mechanisms and protocols.
- Anonymity (Anonymität):
  - Anonymity defines a condition in which a person's true identity is not known.
- Pseudonymity (Pseudonymität):
  - Pseudonymity defines a condition in which a person has taken on an assumed identity.
- Auditing (Diensteprüfung):
  - A. defines the process to collect unforged events and facts.

Additional Security Aspects (2)

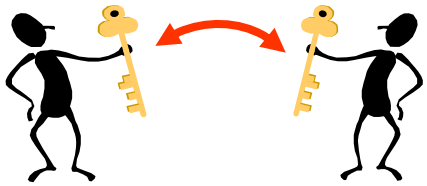
- Identity (Identität):
  - The collective aspect of the set of characteristics by which an object/human is definitively recognizable or known.
- Identity Management (Verwaltung der elektronischen Identität):
  - Encompasses different mechanisms, *a.o.*, password management, user provisioning, and access management. Enables/maintains user access to resources, which includes creation of the user entity, authorization, and permissions, and a single point of administration for de-/provisioning accounts.
- Trust and Trusted Third Party (TTP) (Vertrauenswürdiger Dritter):
  - Besides two arbitrary parties a third one (entity, instance, administration), whom those two parties fully trust.

Cryptography



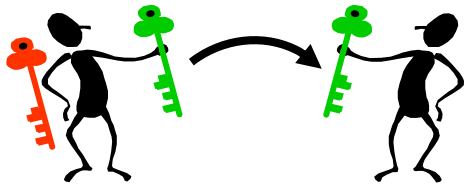
Cryptographic Variants

- Symmetric cryptography:
  - Entities own a shared, secret key



- Advantages:
  - Small overhead/calculation
  - Short keys
- Drawbacks:
  - Key exchange complicated
  - No commitment

- Asymmetric cryptography (public key cryptography):
  - Key pair of private/public parts



- Advantages:
  - Public keys easy to publish
  - Commitment possible (CA)
- Drawbacks:
  - Longer keys
  - Larger overhead/calculation

RSA (1)

- Encryption defines a function f, mapping plain text to cipher text; decryption defines the inverse of f.

RSA: Rivest-Shamir-Adelman

- f requires 5 properties:
  1. f is one-to-one (uniquely invertible)
  2. f is easy to compute (encryption easy)
  3. f^{-1} is difficult to compute (decryption difficult for senders)
  4. f has a domain that is easy to sample from (Bob easily generates a key)
  5. Existence of an easy-to-compute function d of the input of f making computing f^{-1} easy (Bob decrypts easily)

f is one-way

f is a trapdoor function

RSA (2)

- Example trapdoor function: a mod b: remainder after dividing a by b
  - $\boldsymbol{f}(x,e,p,q) = (x^e \bmod pq, pq, e)$,
  - p and q are prime numbers, x (encoded) plaintext, $y=x^e \bmod pq$ is the cipher text
  - $\boldsymbol{d}(x,e,p,q)=e^{-1} \bmod (p-1)(q-1)$ (property 5.) that is $e\boldsymbol{d} \bmod (p-1)(q-1) = 1$
- f is the basis for the RSA Cryptosystem:
  - (e, pq) public encryption key (“public key”)
  - (p, q, d) secret decryption key (“secret key”)
  - Alice generates a private/public key pair once (re-use later)
  - Alice encrypts/signs message with her private key
  - Bob decrypts this message with Alice’s public key

Example (Simplified) – RSA (1)

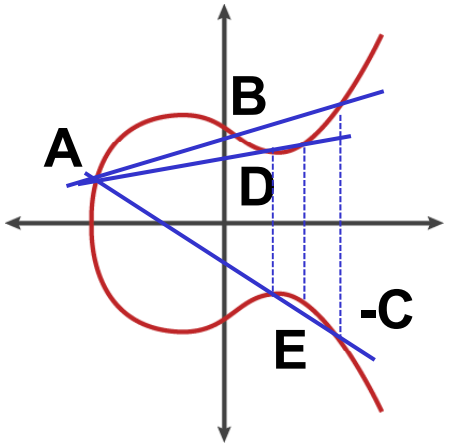
- 2 (large, very large) prime numbers p,q are selected
  - p=11, q=13
- The product N is calculated as N = p * q = 143
- Value e ist calculated to: 1 < e < (p-1) (q-1)
  - e = 23
- Value d ist calculated to: d*e mod (p-1)(q-1) = 1
  - d = 47
- Thus, the number pair (e, N) defines the public key:
  - K_{pub} = (23, 143)
- Thus, the number triple (p, q, d) defines the private key:
  - K_{priv} = (11, 13, 47)

Example (Simplified) – RSA (2)

- Text (plain text) to be decrpyted: P = 75
- Calculation of the cipher text y as: $y = P^e \bmod (p * q)$
  - $y = 75^{23} \bmod 143$
  - $y = 69$
- Cipher text C = 69 will be transmitted as the message
- Calculation of the plain text x as: $x = C^d \bmod N$
  - $x = 69^{47} \bmod 143$
  - $x = 75 \rightarrow P$

Basic Elliptic Curve Cryptography (ECC)

- ECC generates a public-private key pair
  - A 256 bit key in ECC is at about the same security as a 3072 bit key using RSA
  - ECC trapdoor function is similar to a mathematical “game of pool”



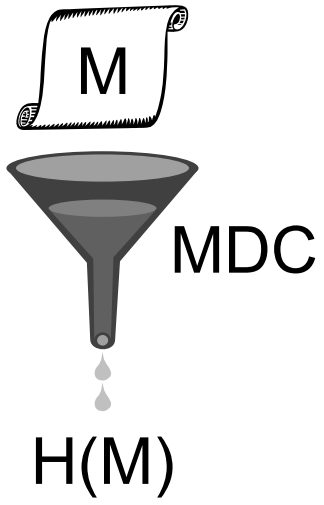
  - Knowing where starting point (A) is and how many hops are required to get to the ending point (E), it is “easy” to find the ending point

    But by knowing where the starting point/ending points are, it is nearly impossible to find how many hops it took to get there
- Public Key: Starting Point A, Ending Point E
- Private Key: Number of hops from A to E

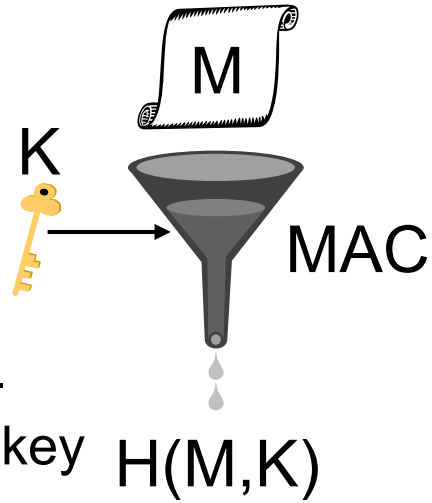
https://blog.goodaudience.com/very-basic-elliptic-curve-cryptography-16c4f6c349ed

Hash Functions and Hashes

- Message Digest Code, MDC
  - Message M (arbitrarily long) → Hash H(M) (hash with a minimum of 128 bit length)
  - Note: "One-way" feature of hash function:
    - Efficient generation
    - Very low collision possibility: M, M' with H(M)=H(M')
    - Examples: (MD5), RIPEMD-160, SHA-256



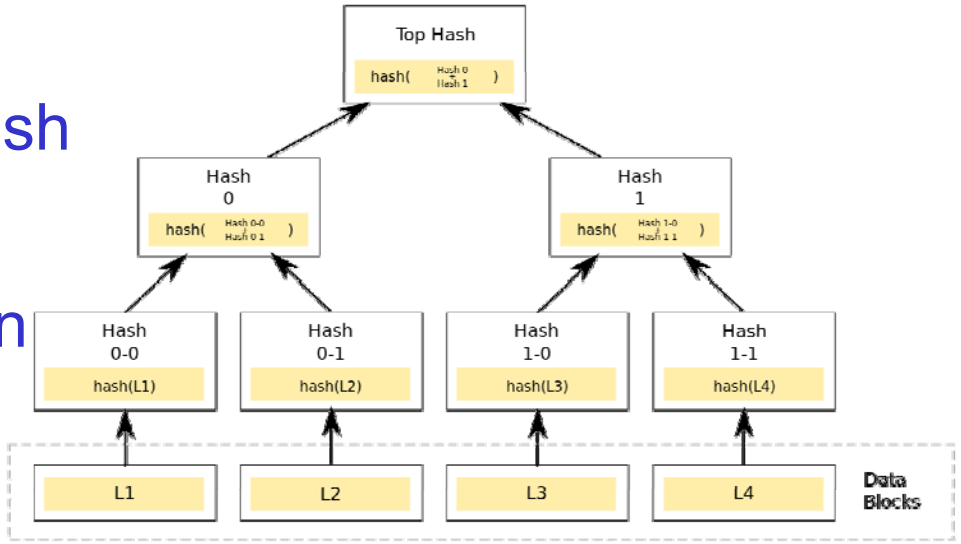
- Message Authentication Code, MAC
  - Message M, key K → Hash H(M, K)
  - May be constructed out of MDC
  - HMAC (RFC 2104), *e.g.*, HMAC-SH-256
  - Cryptographic strength of underlying hash function depending on size of its hash output, size, and quality of key



HMAC: Hash-based MAC, RIPEMD: RACE Integrity Primitives Evaluation Message Digest, SHA: Secure Hash Algorithm

Merkle Trees

- Merkle (or hash) tree
  - Every leaf node is labeled with the hash of a data block and
  - Every non-leaf node is labeled with the cryptographic hash of the labels of its child nodes.
  - Top of a tree there is a root hash
- Characteristics
  - Efficient and secure verification content of large data
  - For binary hash trees: computing hashes is proportional to the logarithm of the number of leaf nodes of the tree
  - Usually cryptographic hash functions used (*e.g.*, SHA-256)



https://en.wikipedia.org/wiki/Merkle_tree#/media/File:Hash_Tree.svg

Zero Knowledge Proofs (ZKP)

- A ZKP is a method by which one party (the prover) can prove to another party (the verifier) that they know a value x, without conveying any information apart from the fact that they know the value x (as of 2013).
  - Any third party reviewing the transcript created cannot be convinced that either the prover or the verifier knows the secret.
    - Ideal cryptographic hash function required
  - Originates from authentication systems, where one party wants to prove its identity to a second party via a secret
  - New: to guarantee transaction validity, despite the fact that information about the sender, the recipient, and other transaction details remain hidden

https://en.wikipedia.org/wiki/Zero-knowledge_proof

Nonce

- Nonce
  - Arbitrary number that can be used just once in a cryptographic communication
    - Often a random or pseudo-random number issued in an authentication protocol to ensure that old communications cannot be reused in replay attacks
  - A nonce may also include a timestamp to ensure exact timeliness
    - Prerequisite: clock synchronization between organizations
  - Example:
    - Salt: random data that is used as an additional input to a one-way function hashing data (password or passphrase)

Trust

- General: Trust is the firm belief in the reliability (character, strength), truth, or ability of someone or something
- In Information Security: Computational trust is the generation of trusted authorities, trusted protocols, or user trust through cryptography
- Key trust facets
  - Initial level of trust needed as an incentive to participate
  - Theoretical aspects of trust and its quantification
  - Relationship between security and trust
  - Significance of trust in distributed network security

https://pdfs.semanticscholar.org/0567/12cf5bc0fba369eb60ebbc23b327d23b1ab8.pdf

Security Issues to Remember

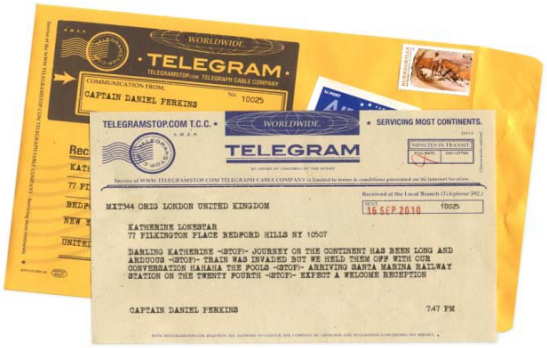
- Security mechanisms are inevitable for modern communication networks and distributed systems
  - Operational and technical security and user behavior
- Security is only as strong as its weakest component
- Security is costly and does affect the ease-of-use concept of a system and its interfaces
- Trust increases/decreases and evolves over time as a consequence of security mechanisms applied
  - All security aspects (mechanisms and use) in combination only can deliver the basis to generate trust between users and systems; trust exists between users/systems

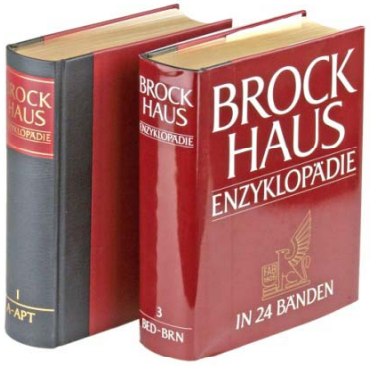
Part II: Blockchain Basics (1)

4. Principles of Blockchains

The Digitization of Society

Physical Objects





Telegram

Encyclopedia

Money

Ended Dec 29, 2017 in Belgium

Digitized Representations

Since mid 70's, RFC 524

Since 2001

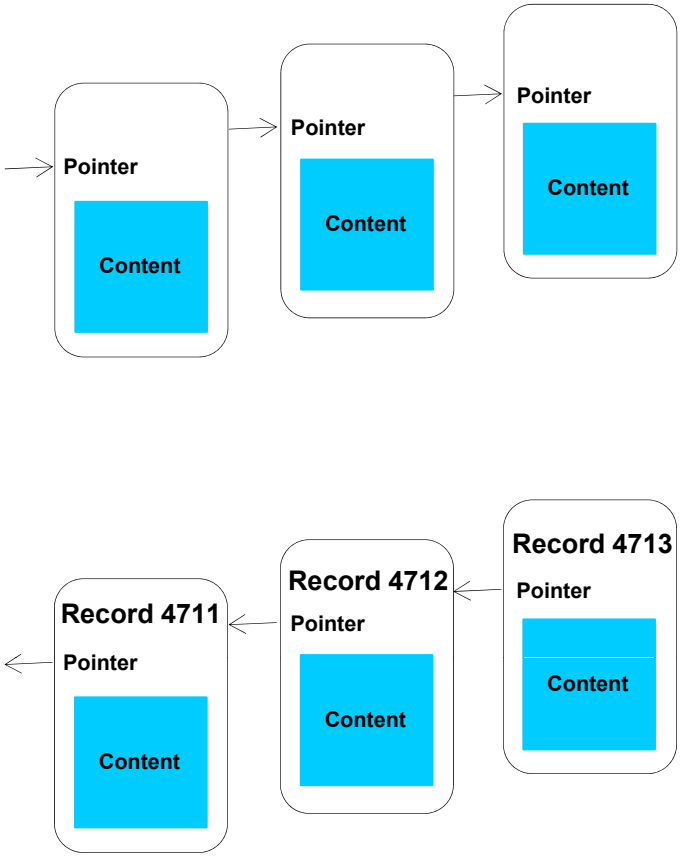
Since 2009

Bitcoins

Systems operated as decentralized and distributed systems!

Excursus 1 – Data Type “Linked List”

- Linear collection of data elements (records)
  - Linear order of records is given by pointers to the next record
  - Data structure as a group of nodes represents an implicit sequence
  - Example: backward linked list
  - Data structure as a group of nodes represents an explicit sequence due to record identifiers added



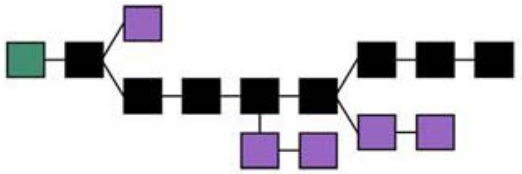
Excursus 2 – Databases

- Databases (DB)
  - Access control for users and the “SysAdmin” (trusted root)
    - Centralized, physical, and trusted servers are maintained
  - Potentially central point of failure, loss, or misuse
  - Exchange of records between clients/database via commands
    - SQL statement s include: INSERT, SELECT, UPDATE, DELETE, …
  - Databases typically maintain the ACID principle on transactions
    - Atomicity, Consistency, Isolation, Durability (ACID)
    - However, records updated are updated and records deleted are deleted → No storage of any history (standard case)

- Distributed DBs
  - Every copy runs on a trusted node

Distributed Ledgers

- Can data be stored fully decentralized and handled reliably between non-trusted stakeholders?
  - Unstructured/structured data stored across the world by anyone
  - Access control by “all” w/o a central root
  - No central point, redundant copies, non-trusted participants, and detectable misuse → Distributed (Shared) Ledger (DL)

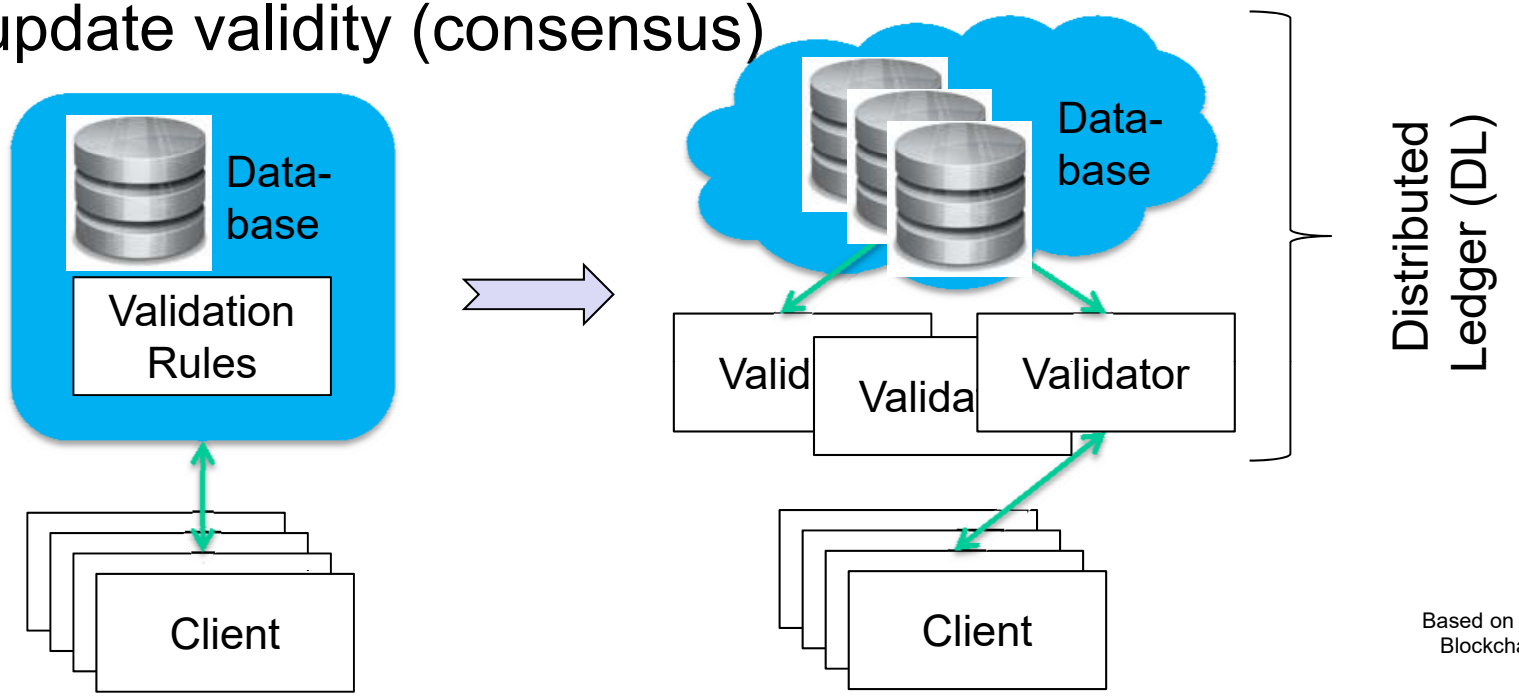


Block

  - DL defines a “consensus” on replicated, shared, and synchronized digital data (blocks), geographically spread across many sites on earth
    - Every node participating may hold a copy of the full DL, thus, no copy is trusted differently and no formal main “no. 1” instance exists

Key Idea: “Replacing” (Central) Databases

- Distributed Ledgers replace clients’ access-protected writes to an authoritative database via validation rules by a distributed consensus of decentralized validators
  - where the database’s state depends on majority agreement of update validity (consensus)



Based on Terence Spies:
Blockchain Mechanics

Definition(s)

- [Distributed Ledgers (DL) or] Blockchains (BC)
  - Digital records of who-owns-what w/o a central storage

  - Consensus Mechanism (CM) ensures that each node's copy of the ledger is identical to every other node's copy

  - Write access to BCs by miners or validators (with data from any asset owner) for transactions via CM and cryptographic signatures, read access at no "costs"

- Key advantages of (public) BCs
  - Immutable, traceable, and preventing "double spending"

"Who-owns-what" Records

- A digital asset = an electronic representation, *e.g.*, file
  - Inherently bears the exclusive right of use of this file
- A token (digital token) = digital asset
  - Issued by a stakeholder, giving right to participate within that network of stakeholders
  - It may allow for "payments" inside that network
- A coin (digital coin) = electronic representation of value
  - Specifically designed to represent digital "money" within a network of stakeholders, typically the BC, and beyond
  - Counterfeiting and double-spending prevented by cryptography

⇨ One can buy a token with a coin, but generally not a coin with a token.

Token Types

- Utility tokens provide access digitally to an application or service by means of a blockchain
- Asset tokens represent assets such as (a) a debt or (b) an equity claim on the issuer
  - *E.g.*, they promise a share in future company earnings or future capital flows – analog to equities, bonds, or derivatives
- Payment tokens are used (a) as a means of payment for acquiring goods or services or (b) as a means of money or value transfer
  - Synonymous to "cryptocurrencies" (thus, a dedicated "coin")
    - Note: Cryptocurrencies give rise to no claims on their issuer

Finma Guidelines for ICOs, February 2018

DL/BC Types and Terminology – Simplified

Private permissioned

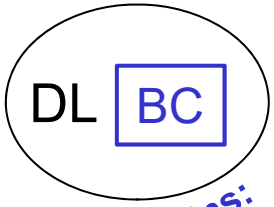
- Read/write/consensus restricted to authorized nodes (pre-defined stakeholders)

– "Enterprise-grade" DL

Public permissioned

- Write/consensus restricted to authorized nodes (pre-defined stakeholders)
- Read open to everyone

– "Controlled collaborative" DL



No real blockchains: "restricted" stakeholders capabilities!

Private permissionless

- Write/consensus restricted to authorized nodes (pre-defined stakeholders)
- Read partially open

– "Consortium-grade" DL

Public permissionless

- Read/write/consensus open to everyone
- No restrictions and full transparency

– "Public" BC, **the** BC

The real and only blockchain (BC)!

5. Blockchain Operations

Blockchain Ingredients (1)

- Public key cryptography and hashes
  - Asymmetric approach for arbitrary users
    - Ensures validation and authentication (in turn authorization)

- Internet
  - Networked infrastructure for everyone
  - Distributed system with arbitrary users and devices (nodes)
    - Peer-to-peer (overlay network) communication paradigms: protocol
    - Storage capabilities for “any”-sized data volumes

- Incentives
  - Supporting rewards for participants’ tasks performed within an overlay network by a “protocol” enabling communications
    - Ensures participation of anyone (potentially non-trusted stakeholders)

Blockchain Ingredients (2)

- Current cryptography determines the basis for BCs
- Hashes

Based on Terence Spies: Blockchain Mechanics

  - A hash function (like SHA-256) takes a block of data and produces effectively a random fixed size integer, *e.g.*, 256 bit
  - Any change to the input randomizes the output
    - "Simple" input changes, *e.g.*, one char, lead to "dramatic" result changes

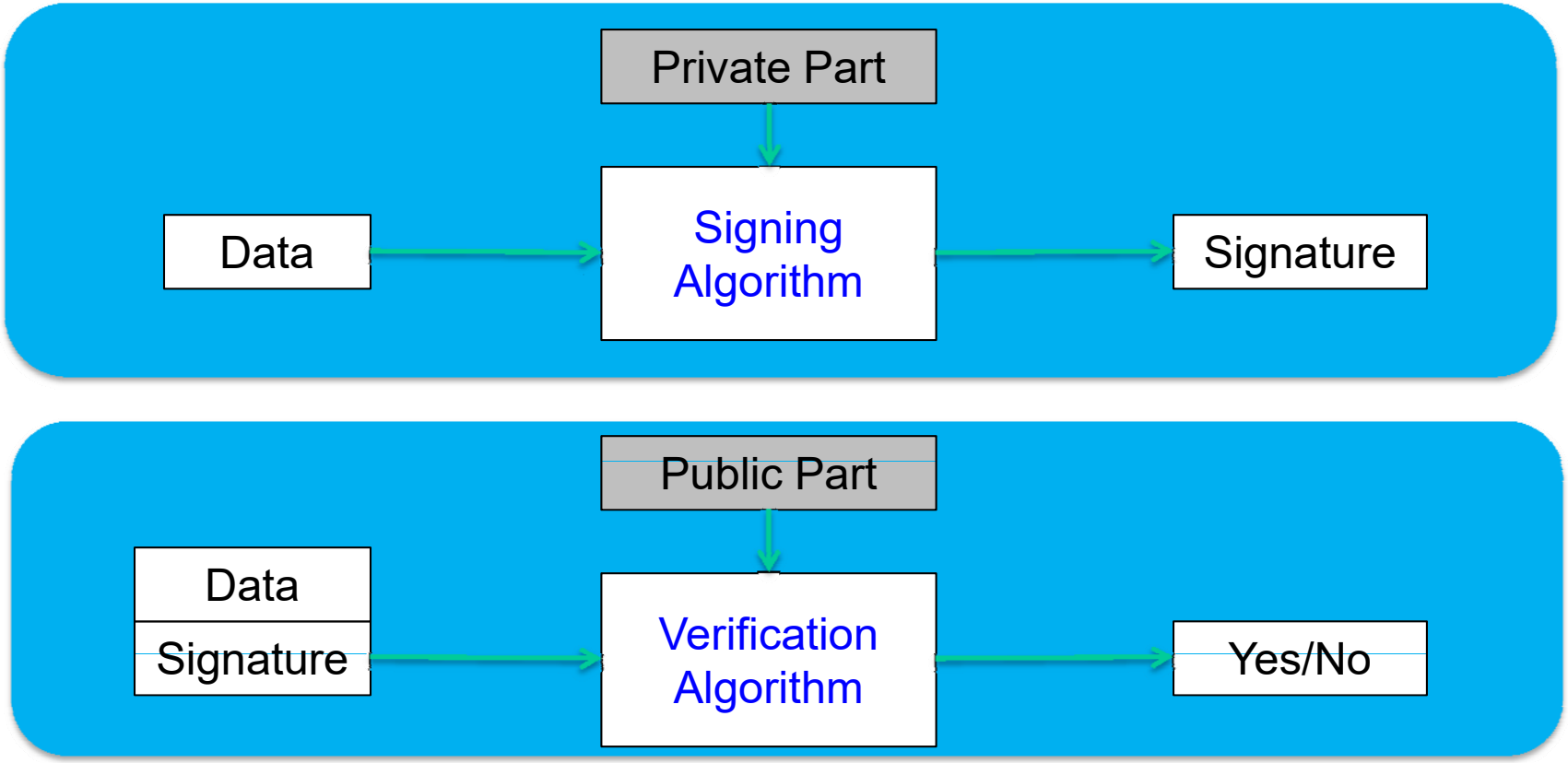
  - Merkle tree "links" pairs of hashes hierachically up to root
    - Any leaf change changes all hashes based on that (incl. the root)

Blockchain Ingredients (3)

- Signatures
  - *E.g.*, RSA

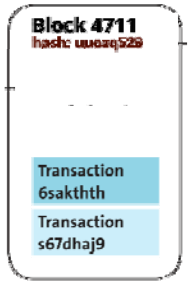
Signing Key	
Public Part	454F4D3E..
Private Part	56F23F2D..

Based on Terence Spies: Blockchain Mechanics



Blockchain Operations

- Transactions (content) collected in blocks
  - New blocks created regularly
- A block contains a hash of and a pointer to the previous block …



Blockchain

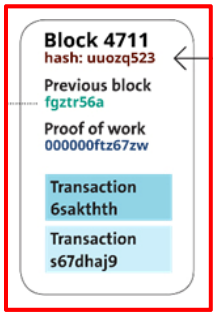
- Consensus mechanism required to determine the block to be integrated into this blockchain

  - Public blocks contain, *e.g.*, solved crypto puzzles (PoW)
    - *E.g.*, a form of partial hash collisions (SHA-256)
- Creation of valid blocks performed by anyone (incentive)
  - Solving crypto puzzles ≡ confirmation of blocks ≡ Mining
  - Computational expensive → Avoids double spending

reward

Blockchain Data Structure in Detail (1)

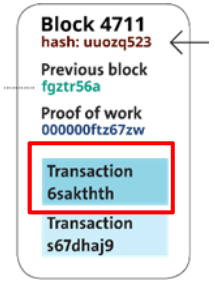
- BCs are a backward-ordered, linear list of blocks
  - Chain starts with genesis block to which others are back-linked
- Blocks contain (at least)
  - Transaction (tx) data (content, payload)
  - Pointer to and a hash of the previous block
  - Cryptographically hashed value of crypto puzzle (result of PoW)
  - Time stamp



- BC's structural and technical characteristics
  - Chain may show side chains, but only one valid branch finally
    - Chronological order guaranteed by previous block's hashes
  - A BC network is organized as a peer-to-peer network
    - Overlay topology may change, replicas of BC are hold on multiple nodes, exchanges of new blocks performed within that overlay, anyone to join

Blockchain Data Structure in Detail (2)

- Transactions (tx/Tx)
  - Data structure encoding the transfer of “value”
    - From a source (“input”) to a destination (“output”)
  - Structures are typically not related to accounts or identities
    - Chunk of “value” locked with a specific secret known by the owner
  - Example tx content



UTXO or account

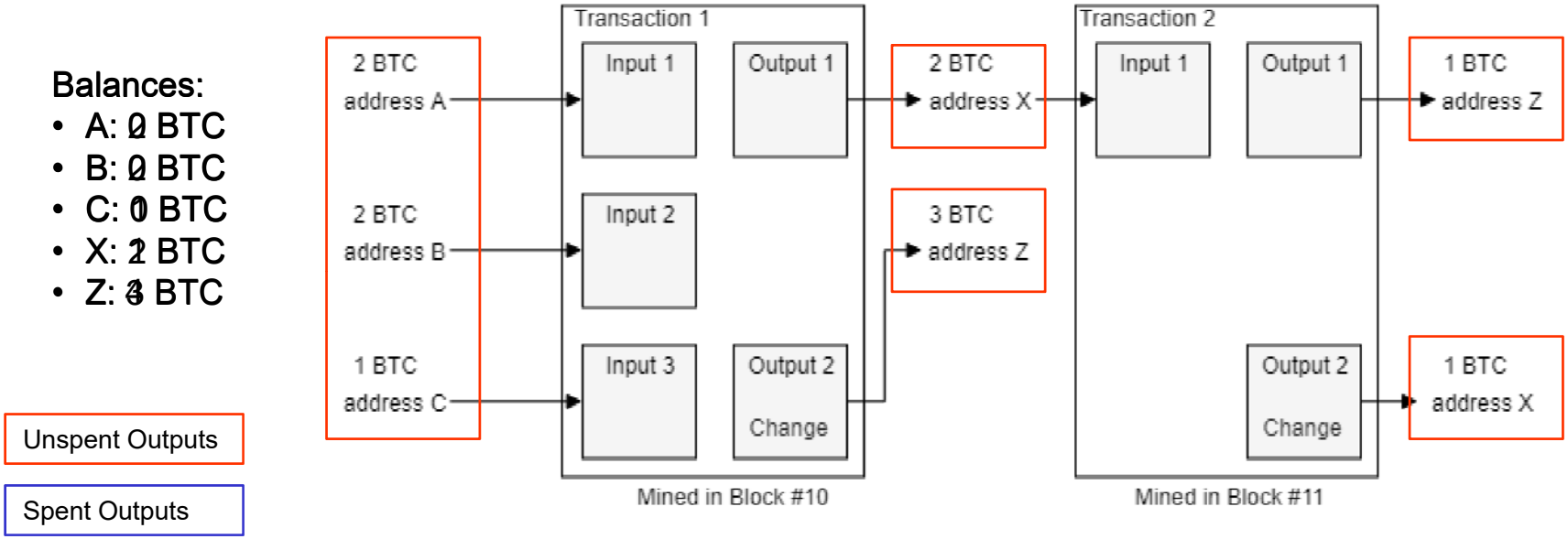
    - Version of tx
    - Input counter – The number of separate input values
    - Inputs – The values
    - Output counter – The number of separate input values
    - Outputs – The values
    - Locktime – Earliest time at which tx will be valid (to be relayed on the BC network or added to the BC), typically a Unix timestamp

Unspent Transaction Output (UTXO)

- Concept based on physical money, *e.g.*, bills

  - Balance is calculated by total of unspent outputs
  - Each transaction has inputs (unspent outputs) and outputs
    - Possible to combine one or more outputs as inputs to form a transaction

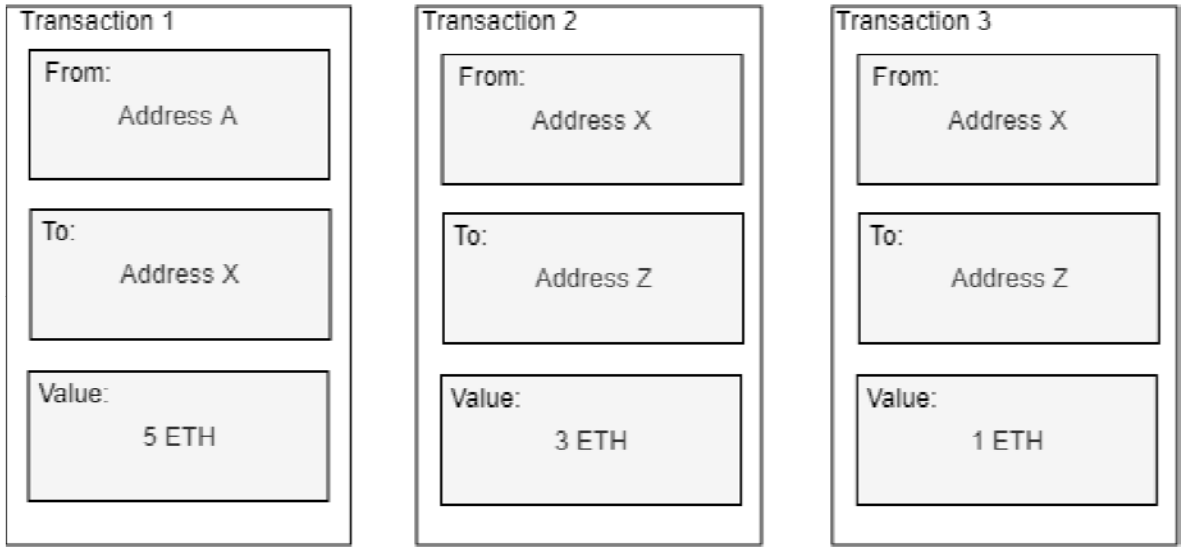
Balances:
- A: 0 BTC
- B: 0 BTC
- C: 0 BTC
- X: 2 BTC
- Z: 3 BTC



Account-based Model

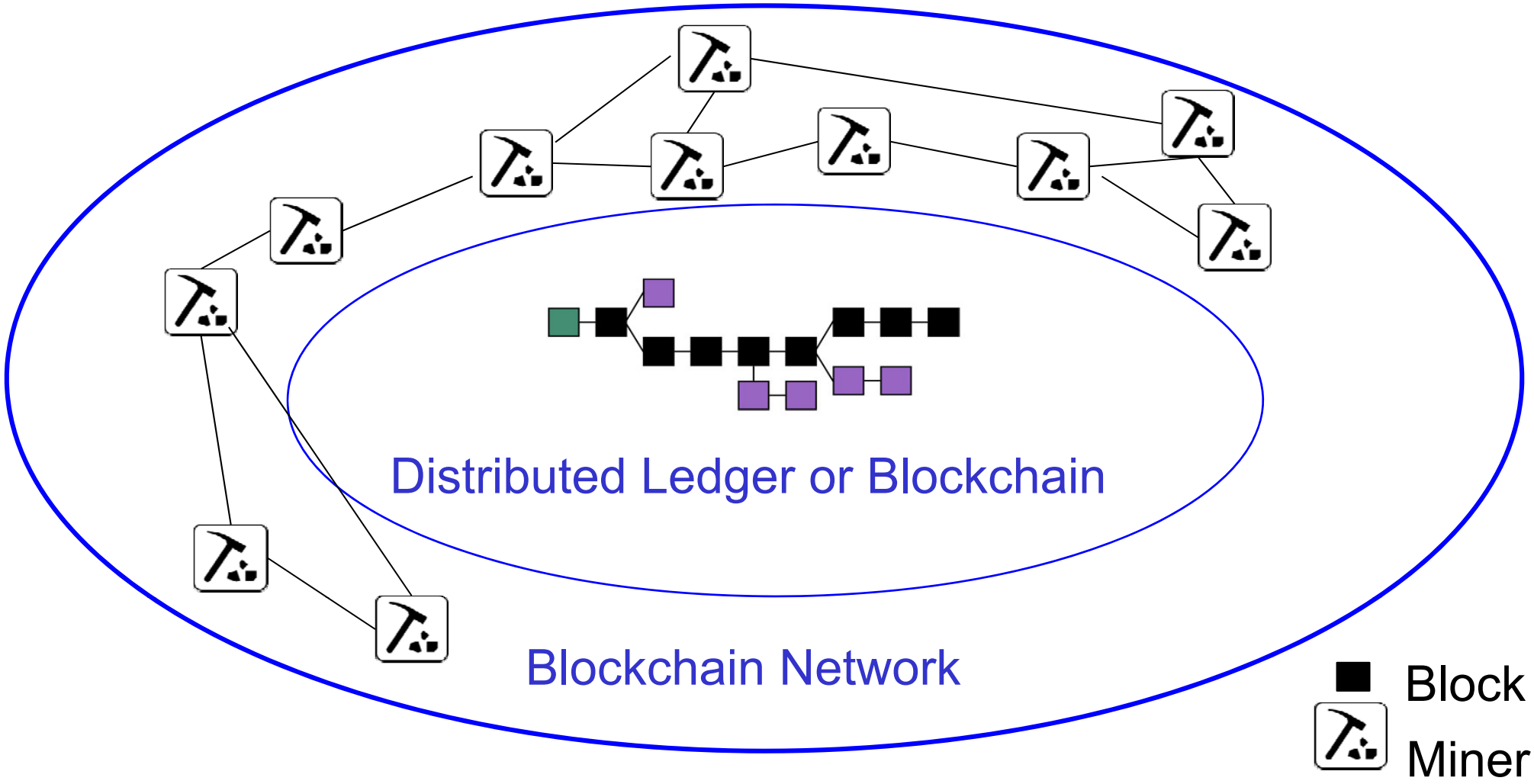
- Account maintained by a global state
- Similar concept to a debit card
  - Each transaction modifies the state of accounts
    - Balance should be larger or equal than spending amount

Balances:
- **Account A: 0 ETH**
- **Account X: 3 ETH**
- **Account Z: 3 ETH**



Account A's initial balance: 5 ETH

Blockchain Terminology (1)



Chain: Backward-linked list of cryptography-based (hash-secured) pointers to previous blocks

Blockchain Operations in Detail

For illustrative reasons, the Bitcoin BC (BTC) is used here.

- Overall operations consists out of 4 steps (in principle)
  1. User A wants to pay user B the amount of 1 BTC (bitcoin)
     - This intent is broadcast as a transaction via the P2P protocol into the BC network
  2. Any miner interested, adds this transaction to his/her respective current block
  3. Any of these minors wins a lottery and “mines” the block
     - That miner solved the crypto puzzle at first
     - Such a block may contain multiple transactions and is mined in well-defined periods of time
  4. The “definitive block “– including the transactions and the new hash – is broadcast to the BC network again as well and added to everyone’s BC copy

Hash Computation and Checks (1)

- Once a set of transactions becomes available, a block is created by utilizing the following data
  - Transaction(s)
  - Hash of previous block
  - Nonce (arbitrary number, used only once)
  - Other information (depending on the BC)
- The hash of new block is calculated meeting the "target"
  - Target hash is a number (absolute value) that a hashed block header must be less than or equal to in order for a new block to be awarded
    - Value used to determine the "difficulty of the input"
    - Value adjusted to ensure that blocks are processed "in time"
- Once the hash was computed, the hash is broadcast to the BC network and checks are performed

Hash Computation and Checks (2)

- Accepted blocks require the miners to solve a crypto puzzle (PoW): hash
  - Difficult to produce (time-consuming, energy), but easy to verify
  - Validity is ensured by checking if a block's hash value is less than "current target"
    - Hash is above the target value → Another miner may have found a suitable hash (not known yet to others), block attached to local BC only, but miner lost the distributed lottery, nonce will be incremented and a retry starts
    - Hash is below the target value → This miner won the distributed lottery and the new block's hash determines the crypto puzzles' result (PoW result ≡ hash)
- Since each block contains the preceding block's hash, a sequence of those determines a larger amount of work
  - Changing any block would require the regeneration of all successors and redoing the work on the data they contain
  - Thus, at least 6 successors are required to consider a block being valid
- The PoW (hash, result of the crypto puzzle) is validated by other members of the BC network by confirming the new hash before adding the block to their local copy, at 51% of the network hashing power' ok, the block is "definitive"

Blockchain Terminology (2)

- Mining (Process)
  - The process of BC members trying to solve the crypto puzzle and adding the respective new block onto the BC
- Miners
  - Those BC members, who run machines to solve crypto puzzles
  - Their reward in case of a successful inclusion are tokens of BC
    - In case of the bitcoin BC the reward are BTC (incentives to participate)
- Checks
  - Verification of hashes broadcast to the BC network

Target Value and Adaptation Algorithm

- Mining is rewarded → Likely more miners join
  - Higher processing capacity increases likelihood on finding hash earlier
- Block creation rate in-, average mining time decreases
  - To maintain the ideal goal of 10 min mining per block: change difficulty, which effects the "target value"
- BC network decreases target value to increase difficulty
  - Decreasing target values increases difficulty to find the hash
- Block creation rate decreases and average mining time increases again
  - System stabilizes itself again, and continue as of above

– Reverse principle of cycle applies, if block creation rate decreases

Consensus: Hash-based Proof-of-Work (PoW) (1)

Based on Terence Spies: Blockchain Mechanics

- Key: One cannot compute an input from an output
  - To find a hash with *N* zeros at input start, requires 2**N* computations, which proves computational work performed
  - Hashing an incrementing "nonce" as hash input, leads to zeros

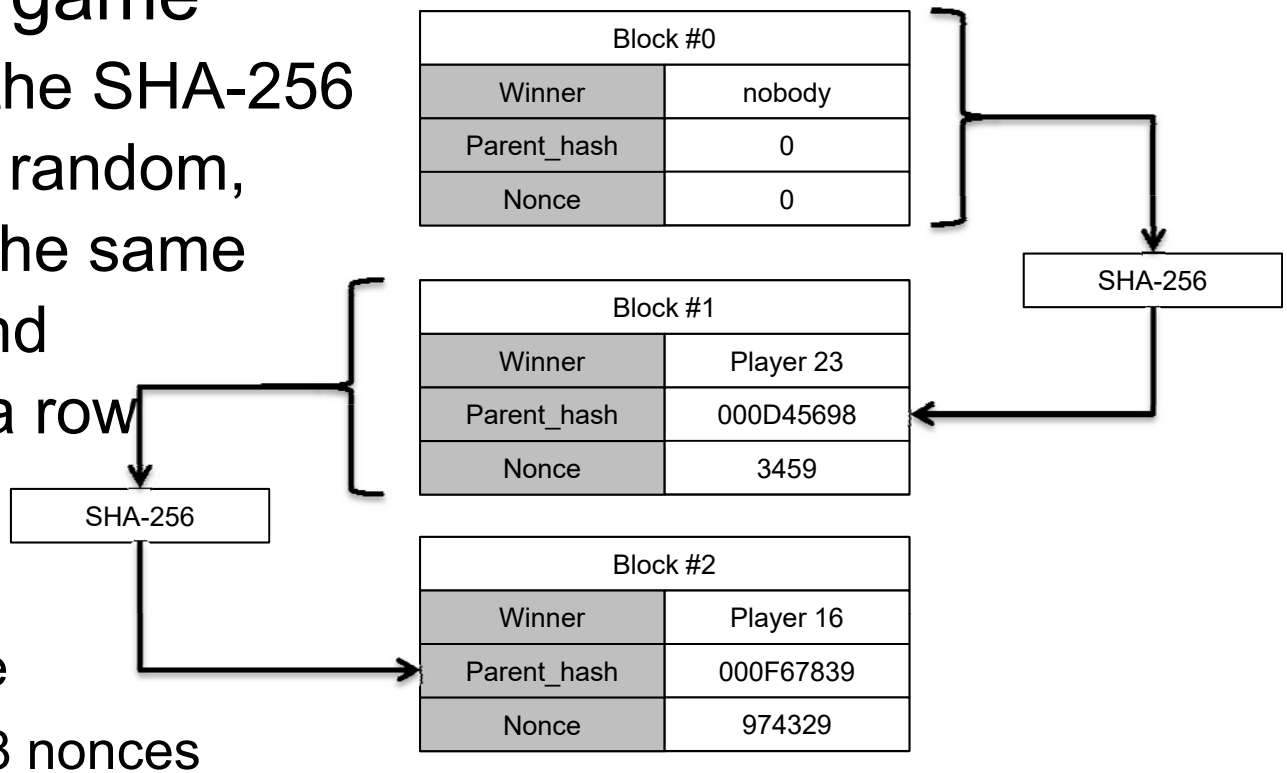
```
in 3e-05 seconds, nonce = 0 yielded 0 zeros.  value = 4c8f1205f49e70248939df9c7b704ace62c2245aba9e81641edf…
in 0.000138 seconds, nonce = 12 yielded 1 zeros.  value = 05017256be77ad2985b36e75e486af325a620a9f29c54…
in 0.000482 seconds, nonce = 112 yielded 2 zeros.  value = 00ae7e0956382f55567d0ed9311cfd41dd2cf5f0a7137…
in 0.014505 seconds, nonce = 3728 yielded 3 zeros.  value = 000b5a6cfc0f076cd81ed3a60682063887cf055e47b…
in 0.595024 seconds, nonce = 181747 yielded 4 zeros.  value = 0000af058b74703b55e27437b89b1ebcc46f45ce55d6….
in 3.491151 seconds, nonce = 1037701 yielded 5 zeros.  value = 00000e55bd0d2027f3024c378e0cc511548c94fbeed0e….
in 32.006105 seconds, nonce = 9913520 yielded 6 zeros.  value = 0000007 7a77854ee39dc0dc996dea72dad8852afbde6….
in 590.89462 seconds, nonce = 186867248 yielded 7 zeros.  value = 0000000225060b16117b23dbea9ce6be86ac439d….
in 4686.171007 seconds, nonce = 1424462909 yielded 8 zeros.  value = 000000002dd743724609a9f57260e2492908d….
```

- The distributed game sets the difficulty *N* of the game
  - Players get a list of blocks and their content (see slide before)
- Players accumulate "points" by creating blocks
  - Hashing the previous block, finding a hash of the new block with enough zeros, and transmitting this block to everyone

Consensus: Hash-based Proof-of-Work (PoW) (2)

Based on Terence Spies: Blockchain Mechanics

- The "chain race" (probability theory)
- The difficulty of the game
  - For N zeros, since the SHA-256 output is effectively random, getting zero bits is the same as flipping a coin and getting N heads in a row
  - For N zeros, have to try 2*N/2 nonces
    - N=1 …. Try 1 nonce
    - N = 16 … Try 32768 nonces
    - N = 32 … Try 2 billion nonces
  - Winning a block proves the player did work!



Blockchain Terminology (3)

- Difficulty
  - Measure of how difficult it is to find a new block compared to the easiest it can ever be
  - Recalculated every 2016 blocks in the BTC BC such that the last 2016 blocks would have been calculated within two weeks
    - On average, one block is mined every 10 min
- Consensus
  - State reached where the majority of members of the same P2P network agrees on the same mining output (51% agree)
  - This state of the consensus is secure and tamper-resistant, immutable with respect to the blocks
  - Respective block data is persisted to the BC network's nodes

Communications – The Flood Protocol

- New transactions and mined blocks are broadcast to the BC network
  - Minimum of 8 neighbors maintained in Bitcoin P2P network
- Temporary BC splits possible
  - 2 miners arrive at 2 different, but valid hashes at the same time
  - P2P network resolves that split in short time to 1 valid branch
  - Clients only accept “longest chain of blocks” as valid (pruning)
    - Length is determined by the most combined difficulty, not #blocks
- As soon as block gets “solved” as valid, every miner in the BC network must stop the current mining process on that block and can start with the next one

Segregated Witness (SegWit)

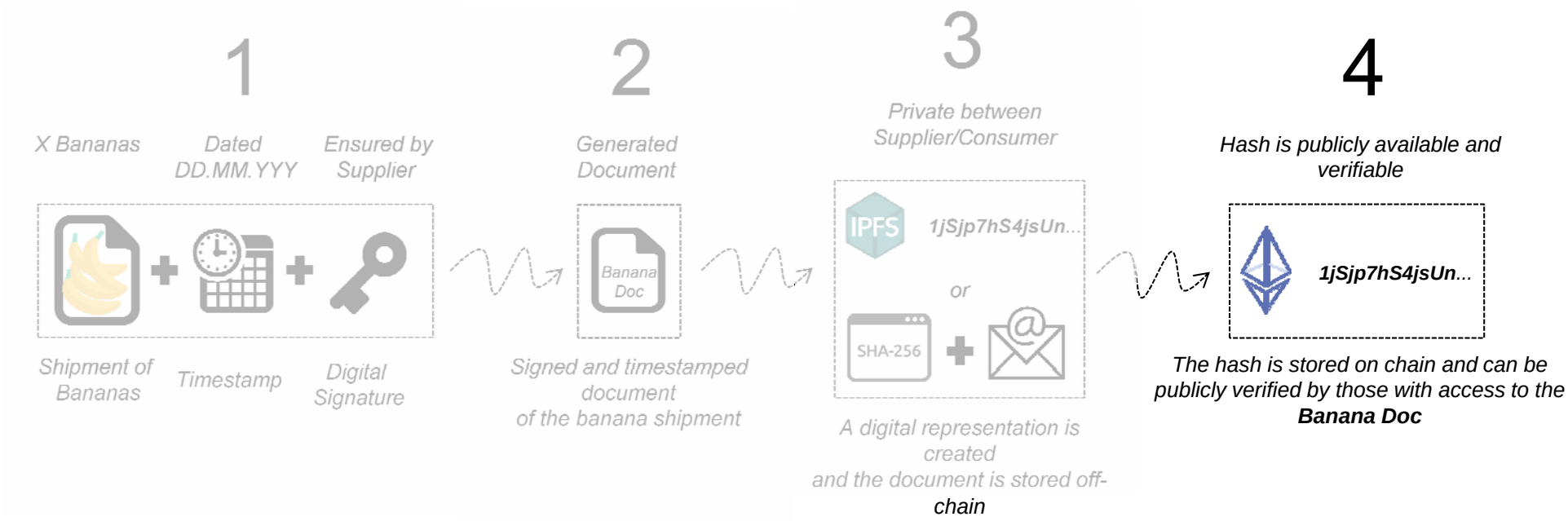
- BC protocol “upgrade” to provide interception protection on transaction malleability and increase block capacity

  Transaction malleability is an attack that lets a person change a Bitcoin transaction's unique ID before confirmation on the Bitcoin network. This change makes it possible for the person to pretend that a transaction didn't happen.

  - SegWit separates *witness* from the list of inputs
  - Witness contains data required to check a transaction validity, but is not required to determine tx effects https://en.bitcoin.it/wiki/Segregated_Witness
  - *weight* parameter defined
  - Blocks are allowed to have at most 4 million weight units (WU)
    - Non-witness and pre-segwit witness bytes weigh 4 WU, but each byte of Segwit witness data only weighs 1 WU, allowing blocks that are larger than 1 MB without a hardforking change
- SegWit was the protocol change needed to make the Lightning Network safe to deploy on the Bitcoin network

Blockchain‘s Immutability Exploited

- Time stamping proves existence, integrity, and creation/exchange of digital assets
  - Keeping track of the creation and modification time



Supplier can use a blockchain-based time stamping solution to ensure the quality of his products

Transaction (tx/Tx) Fees

- Mining and consensus finding require resources
  - Specifics depend on the dedicated approach used
- Generally, tx fees are part of the tx
  - Compensation for miner to mine
    - Tx fees are collected by that miner
    - Tx fees are set by the BTC's market forces
  - Tx fees incentivize a miner to include a tx into a block
    - High fees: early selection; low fees: delayed selection
      - No requirement for tx fees, but mining may be delayed "forever"
  - Tx fees are countermeasures against "spam" as providing no tx fees inside a tx may not lead to its selection
  - Tx fees are typically calculated by the size of the tx (in Byte)
    - A calculation of the tx fees based on the tx values is not applied

Blockchain Transaction Types

- On-chain tx
  - Available on the blockchain visible to all nodes on the BC
  - Tx valid when a communicated that tx across the network
    - Tx times may vary depending on the network load or tx queued
- Off-chain tx
  - Value resides outside of the blockchain
  - Tx executed instantly
- On-chain tx with off-chain storage of data
  - The same is valid as for on-chain tx
  - Additionally, the tx does not contain the full data, but only a hash of the data, which is stored off-chain
    - Reduces storage size of BC for many tx or large volume

Blockchain Addresses (1)

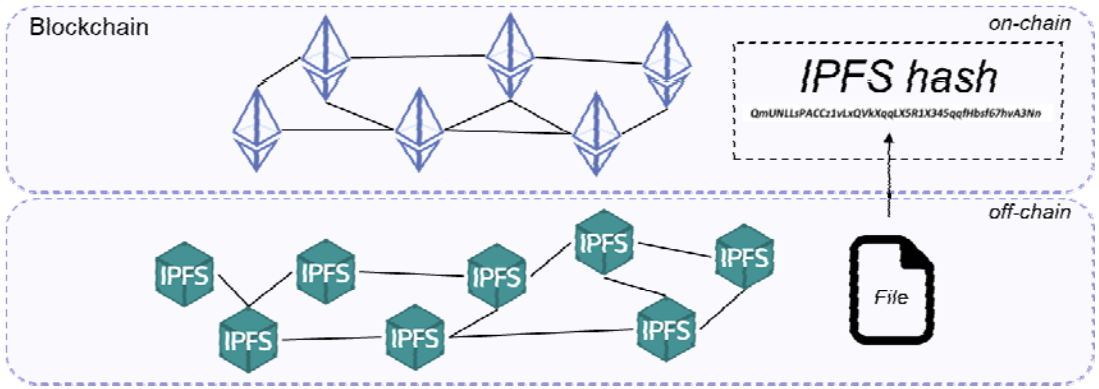
- Transactions on a blockchain require the knowledge of the blockchain address of the sender and recipient
  - Address as 27-34 digit code consists of letters and numbers
    - Generated by the wallet, where public-private key pairs are stored, from public key via cryptographic hash function (“finger print”)
    - Usually currency-specific, addresses belong to a certain blockchain
  - Example Bitcoin
    - A = RIPEMED160(SHA256(K))
      with A Bitcoin address and K public key
    - Results in a 160 bit (20 Byte) number
  - Note: A public key **IS NOT** the same as a Bitcoin address, but this is derived from a key with a one-way hash function!

Blockchain Addresses (2)

- Address representations
  - Base58 encoding (subset of Base64)
    - Text-based binary-encoding format, using upper- and lowercase letters, but omitting “0” and “O”, “I” and “l”, and “\”, “/”, and “+”
    - Balancing compact representation, readability, and error detection
  - Base58Check encoding
    - Adds 4 Byte checksum (error-checking over encoded data) to the add
      - Used for BTC, prevents transcription and typing errors
    - Example:
      - Add a *prefix* to *data* (version byte), $0x00_h$ in the Bitcoin address case $0x80_h$ in case of a private key to be encoded
      - Compute double SHA checksum (*CS*):
        CS = SHA256(SHA256(*prefix*+*data*)), with “+” concatenation, and use of 4 first bytes of this 32 Byte hash

Off-chain Signaling of Addresses

- Problem: BCs typically show limited storage capacity
- Solution
  - Digital representation of a file ("hash reference" != "PoW result") is stored on-chain and the file itself is stored off-chain
- Practice
  - Peer-to-peer network stores and shares hypermedia (*e.g.*, graphics, audio, video, plain text) in a distributed file system



IPFS: Inter-Planetary File System
https://ipfs.io/

Wallets

- Container for private keys
  - Structured files, databases
  - Wallets contain keys (as keychains of private/public keypairs) not coins (which are stored on the BC as tx outputs), thus, keys prove the ownership of a coin!
- Wallet types
  - Non-deterministic (random): just a collection (hard to maintain)
  - Deterministic (seeded): keys derived from a common seed (random number plus index) via hash-functions
    - Hierarchical deterministic (HD) – BIP0032/BIP0044 standard – with keys in a tree structure: parent ⇨ children ⇨ grandchildren
    - Structure organizes inputs, branches with categories (*e.g.*, departments)
  - Paper: character strings, 2D codes, BIP0038 encrypted print

Smart Contracts (1)

- A Smart Contract (SC) may reside inside transactions
  - Executed & validated on every node upon persisting that block
    - *E.g.*, for Bitcoins (blockchain-based cryptocurrency) SCs specify how to withdraw, escrow, refund, or transfer BTC from A to B
- SCs first mentioned in 1996

"Active" database!

> A smart contract is a computerized transaction protocol that executes the terms of a contract. The general objectives of [a] smart contract['s] design are to satisfy common contractual conditions (such as payment terms, liens, confidentiality, and even enforcement), minimize exceptions both malicious and accidental, and minimize the need for trusted intermediaries. Related economic goals include lowering fraud loss, arbitrations and enforcement costs, and other transaction costs.
>
> N. Szabo

- Smart contracts alone are not "smart"
  - They need an infrastructure ("technology")
  - A blockchain forms ***the*** ideal, distributed basis for SCs
- The legal relevance of "coded", more general contracts?

Smart Contracts (2)

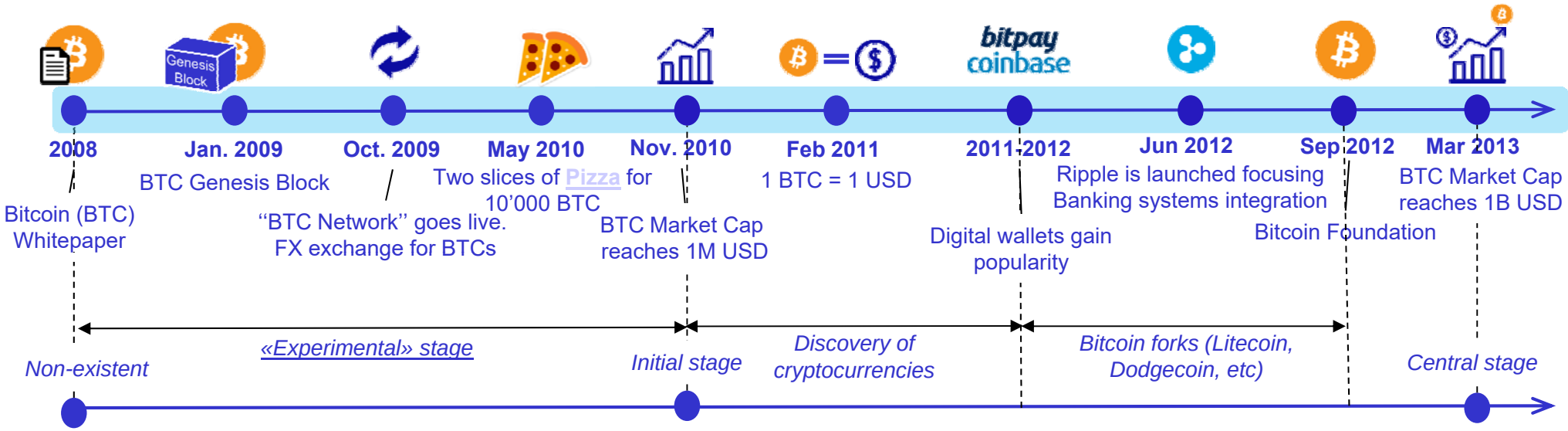
- SCs can be exchanged (not only transactions)
  - This is the code or program to be executed
  - Programs and APIs on the BC
  - Thus, SC are programs that encode conditions and outcomes
- *E.g.*, SCs can define secure escrow services
  - Ready to use program working on pre-determined conditions between any two users (e.g., a provider and a customer)
  - "Real-time" operations
  - Near zero marginal cost
- Application domains grow!
  - Well beyond crypto currencies and Fintech
  - *E.g.*, multi-signature account services for asset protection, estate planning, dispute resolution, governance

Part III: Blockchain Basics (2)

6. Blockchain Eras and Application Domains

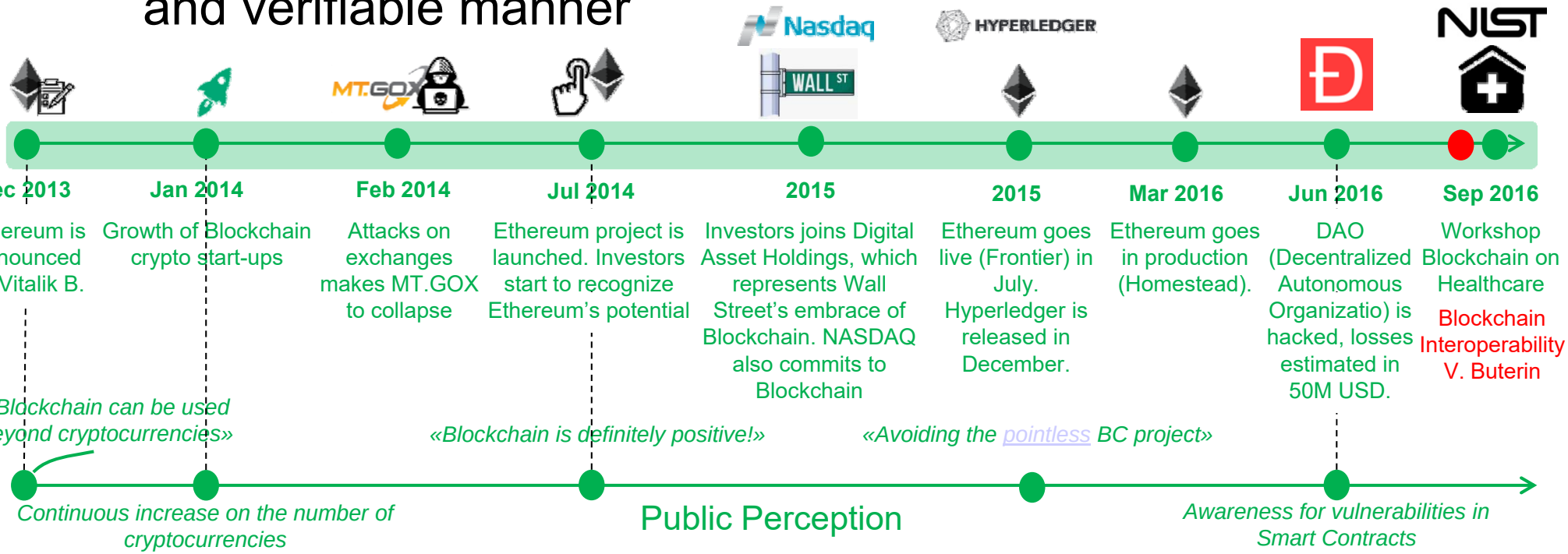
Blockchain 1.0

- Digital Currency or Cryptocurrencies
  - Decentralized payment system
  - Bitcoin as the father of digital currencies
    - Still, not much awareness of (other) blockchain capabilities
  - Proof-of-Work (PoW)



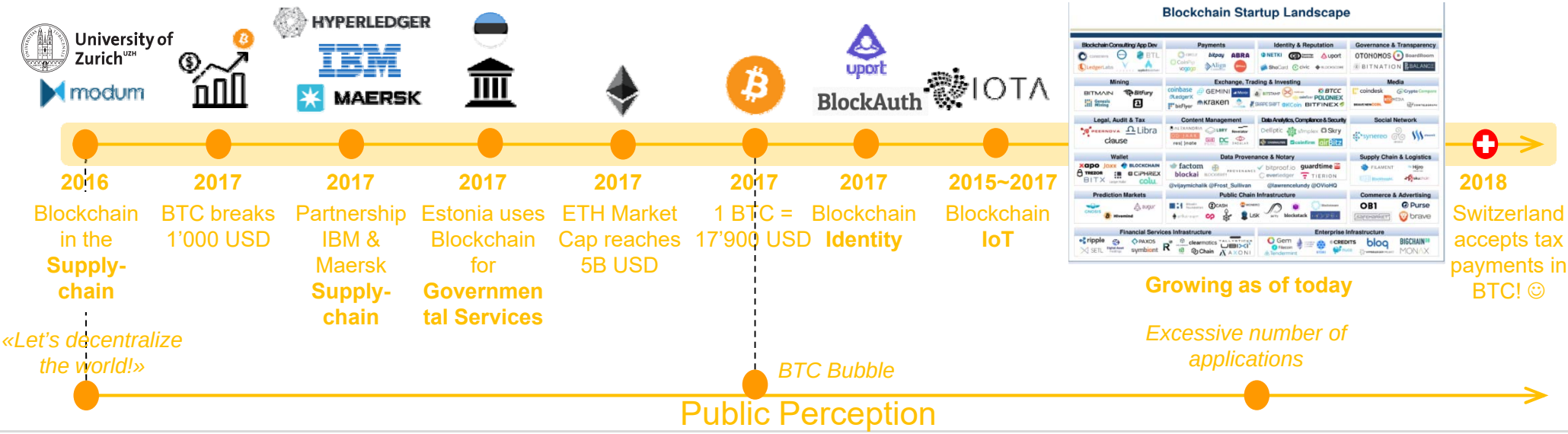
Blockchain 2.0

- Smart Contracts
  - Ethereum unlocks the blockchain potential beyond cryptocurrencies
  - Blockchain is able to run computer programs in a transparent and verifiable manner



Blockchain 3.0

- Decentralized Applications (DApps)
  - Production stage:
    - Large number of applications
  - Scalability/Performance issues:
    - Need for performance → new consensus protocols
    - Need for storage → off-chain storage tools



Blockchain 4.0

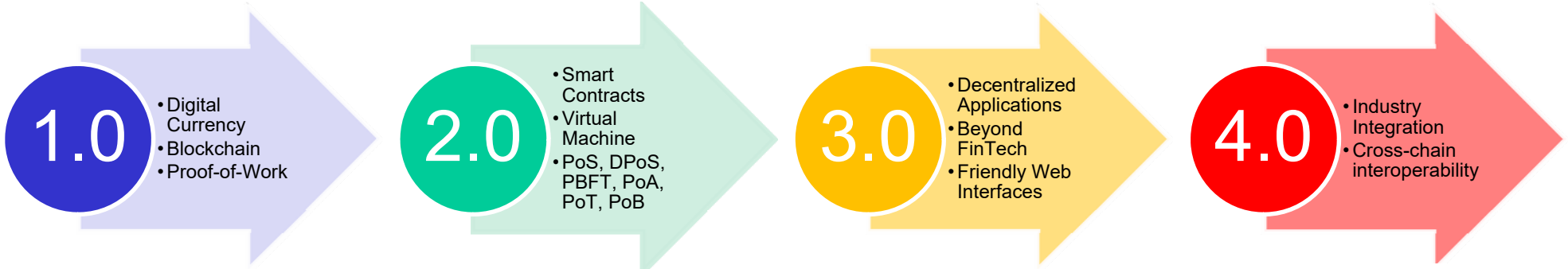
- Ecosystem and Industry Integration
  - Making blockchain effective in industry
  - Decentralized and disconnected blockchain networks
    - Vendor-specific blockchain technology, interoperable chains
  - Need for standardization

As of today

origintrail
wanchain
COSMOS INTERNET OF BLOCKCHAINS
ARK.io
Ambrosus
BLOCKONOMI
aelf.
HERDIUS
Seele
blockstack
AION The Third Generation Blockchain Network
INTERLEDGER Payments across payment networks
MULTIVERSUM HERE TO STAY
Polkadot.
CROWD MACHINE

Blockchain Eras and Evolution

- 4 different BC eras are running in parallel today



  - 1.0 – December 08/January 09: Bitcoins
    - More than 6500+ cryptocurrencies available today

  - 2.0 – 2012-14: Ethereum, Smart Contracts, Solidity, …
  - 3.0 – April 2012: Decentralized Apps (dApps) – "Satoshi Dice"

https://hackernoon.com/dapp-and-things-you-need-to-know-4f50853a4cb7

    - Running on peer-to-peer network, all data transparent and tamper-proof
  - 4.0 – App. 2015: BC ecosystems and industrial integration
    - Countless Blockchain projects in many fields
      - FinTech, supply-chain, governmental, identity, …

Number of BC Application Domains

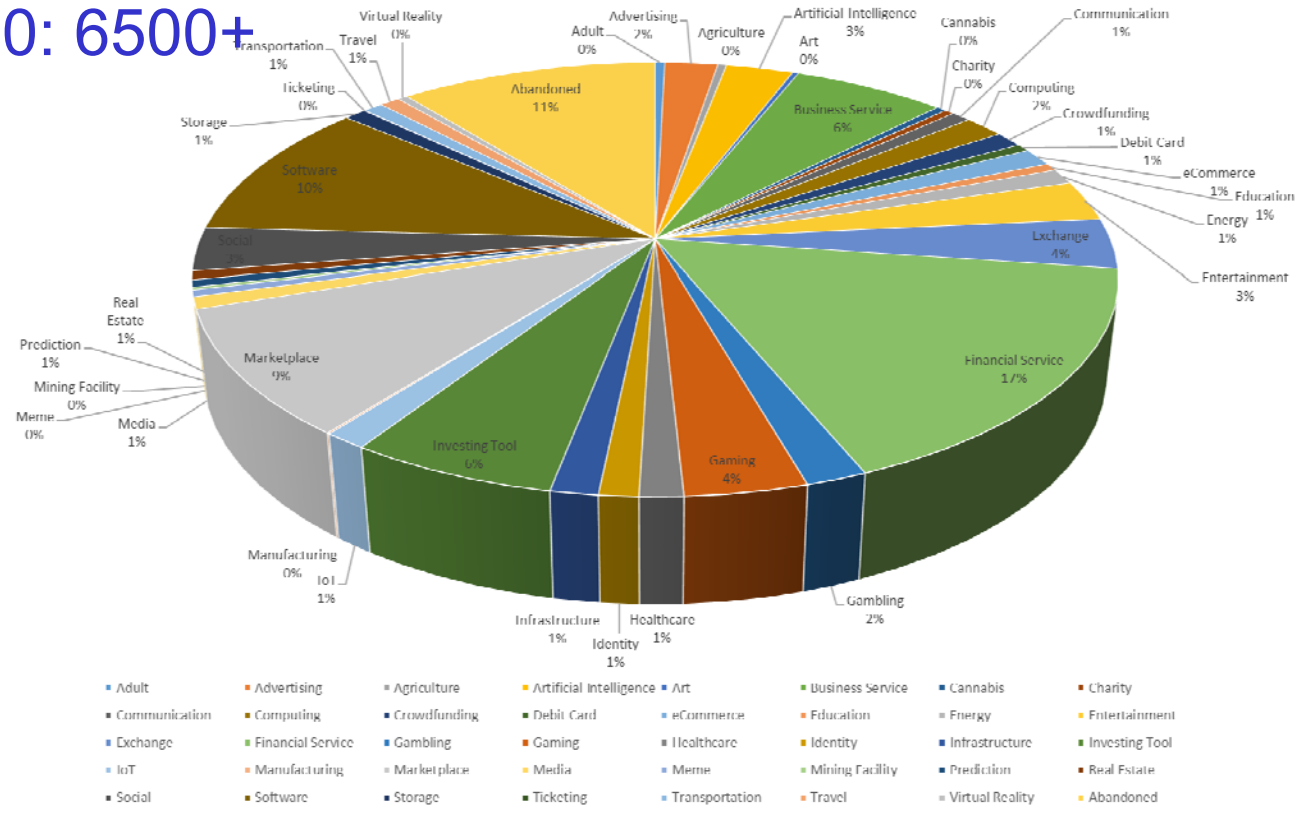
- Based on app. 2500+ cryptocurrencies as of Aug 2019
  - Not all cryptocurrencies are implemented on its own Blockchain

New: August 2020: 6500+

Cryptocurrencies: 6.578 · Markets: 26.897

CoinMarketCap

https://coinmarketcap.com/

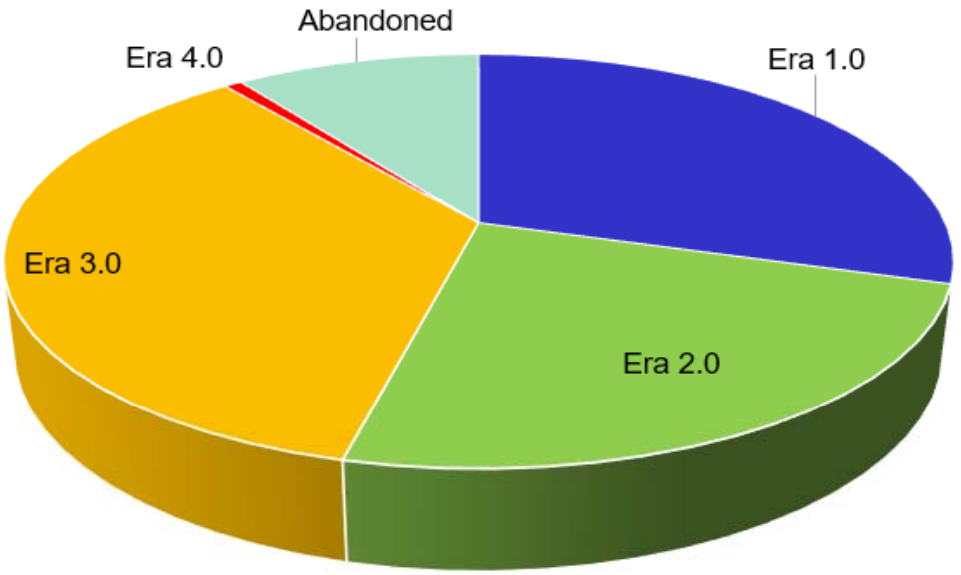


- Finance
  - 17%
- Abandoned
  - 11%
- Software
  - 10%
- Marketplaces
  - 9%

Application Domains Grouped by BC Eras

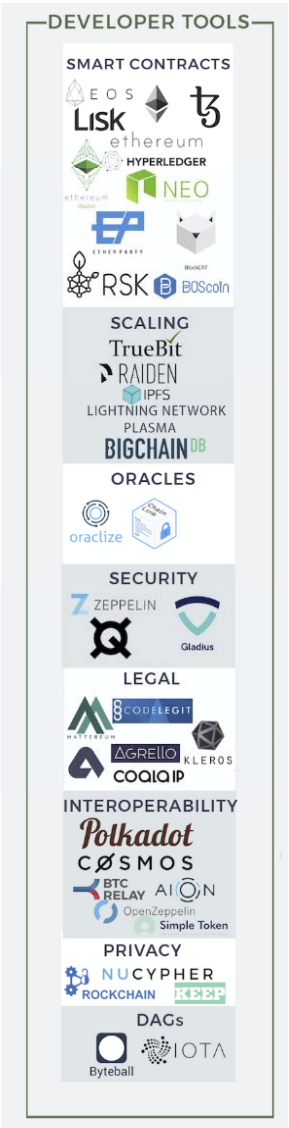
- Similarly, BC projects/domains grouped in Eras
  - Cryptocurrencies and digital finance sector is still dominant
  - However, dApps represents the major number of projects

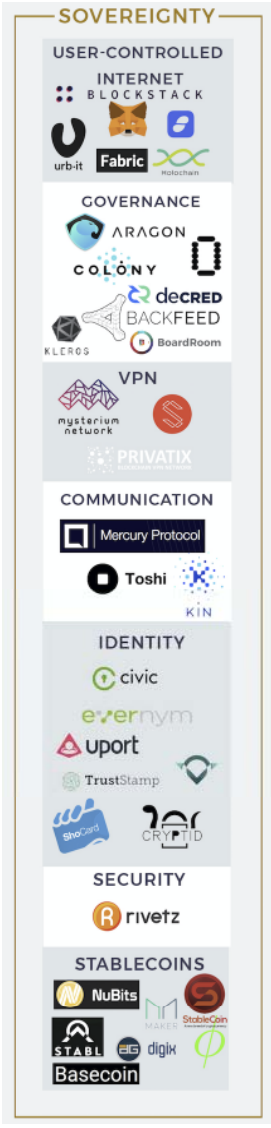
- Era 1.0 (Finance)
  - 738 projects, 30%
- Era 2.0 (Smart Contracts)
  - 602 projects, 24%
- Era 3.0 (dApps)
  - 884 projects, 35%
- Era 4.0 (Integration)
  - 18 projects, 1%
- Abandoned
  - 248 projects, **10%**



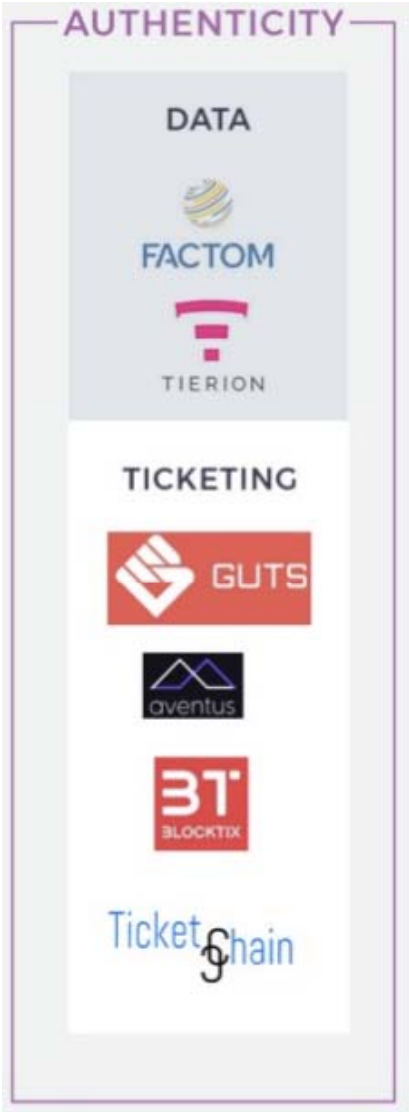
Missing more integration!!

Current Application Domains (1)





Current Application Domains (2)



Application Domain Characteristics

Application Domain	Blockchain (BC) Era	BC or Distributed Ledger (DL)	Read (R) and Write (W) Permissions	Mining Permissions
Currencies	1.0	BC	Public R/W	All nodes
Developer Tools	2.0	BC	Public R/W	All nodes
FinTech	3.0	DL	Private R/W	Selected nodes
Sovereignty	3.0	DL	Private R/W	Selected nodes
Value Exchange	3.0	DL	Public R, Private W	Selected nodes
Shared Data	3.0	DL	Public R, Private W	Selected nodes
Authenticity	3.0	DL	Public R, Private W	Selected nodes
Networking	3.0	BC	Private R/W	Selected nodes
Interoperability	4.0	BC	Public R/W	Selected nodes

*Based on a general observation, characteristics of specific projects within each application domain may vary

Effects of Blockchain Eras

- Different BC Eras present different requirements
  - Performance: transactions per second, latency
  - Reliability: number of nodes
  - Security: trade-offs between confidentiality and transparency

The consensus mechanism is the main responsible to ensure these aspects!

- As a consequence, different consensus mechanisms were necessary with the BC and DL development
  - Since traditional Byzantine Fault Tolerance (BFT) could not deliver all demands, election-based ones (PoW, PoS) appeared

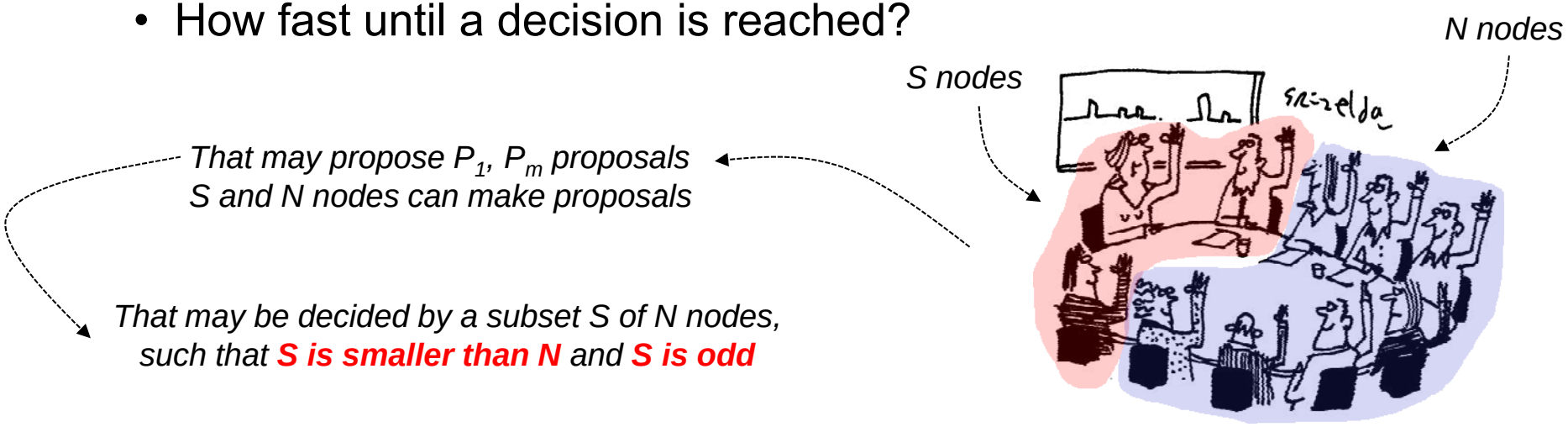
7. Consensus Mechanisms

Mechanisms for Distributed Agreement

- Also called "Distributed Consensus" algorithms
- The 4 key characteristics https://pradeeploganathan.com/blockchain/consensus/
  - Uniform agreement: No two nodes decide differently
  - Integrity: No node decides twice
  - Validity: If a node decides on value *v*, then *v* was proposed by some node
  - Termination: Every node that does not crash eventually decides on some value
- Given a cluster of *N* nodes and a set of proposals P_1 to P_m, every non-failing node will eventually decide on a single proposal P_x without the possibility to revoke that decision. All non-failing nodes will decide on the same P_x.

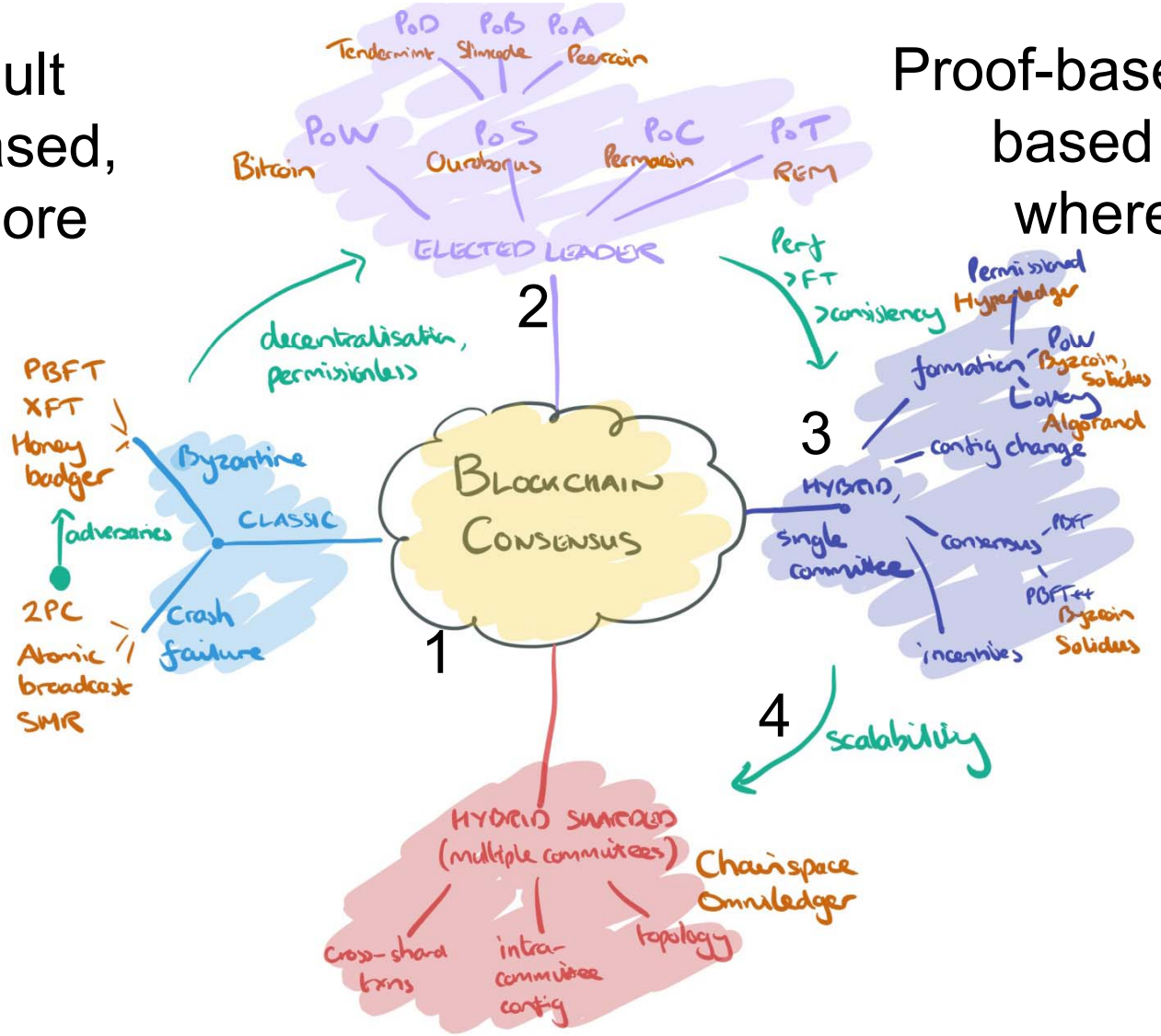
Mechanisms for Distributed Agreement

- How to decide on a single consensus proposal P_x?
- There are many aspects a consensus mechanism has to address:
  - How many of the non-failing nodes are entitled to vote?
    - How to decide who is entitled to vote?
  - How to ensure a decision in a deterministic manner?
    - How fast until a decision is reached?



Evolution of Consensus Mechanisms

Byzantine fault tolerance-based, which is a more traditional approach based on rounds of votes.



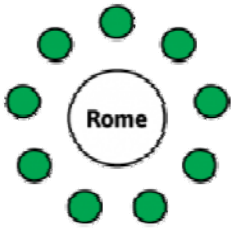
Proof-based or leader-based consensus, whereby a leader is elected and proposes a final value

Figure

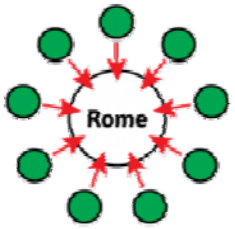
Consensus Mechanism Types (1)

- Classical Consensus Models
  - Crash failure models → honest nodes failing
  - Byzantine Failure Tolerance (BFT)
    - State machine replication
  - BFT General's Problem

HyperLedger (SOLO, Kafka mechanisms), Stellar

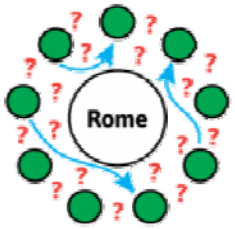


Imagine Rome being besieged by **nine armies**, each commanded by a (Byzantine) general.

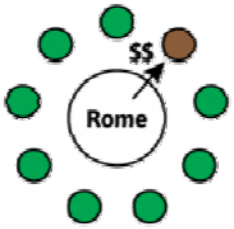


In order to launch a succesful **attack** or retreat, all armies have to **do the same**, otherwise they will be decimated by Rome's armies.

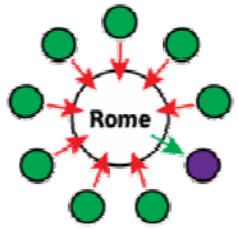
The decision to either **attack** or retreat is put up to a **vote**. Whichever option receives **more than 50%** of the votes, that's what the Generals will do (**retreat** in the example above).



Problem 1
The generals communicate by using **couriers**, who have to cross unknown areas controlled by the Romans, risking capture or their message becoming corrupt.



Problem 2
Each of the generals could be bribed by the Romans: **Traitorous Generals**.



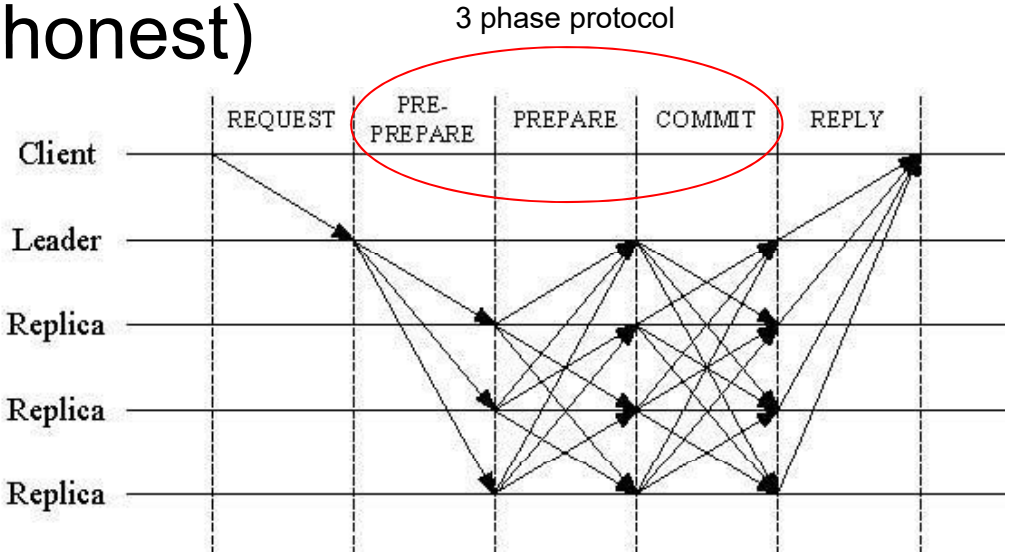
Problem 3
Any of the Generals can make the wrong decision, regardless of the vote: **Improperly Functioning Generals**.

CryptoGraphics.info

Byzantine Fault Tolerance (BFT)

- Described as the capacity of a system to handle or survive unreliable situations and (all kinds of) failures
- Practical BFT (PBFT): assume a small fraction of nodes as Byzantines (dishonest)

1. A client sends a request to invoke a service
2. The primary leader multicasts the requests to the replicas
3. Replicas execute the request and send a reply to the client
4. The client wats for F+1 replies from different replicas with the same result



- Other examples: XFT, HoneyBadger

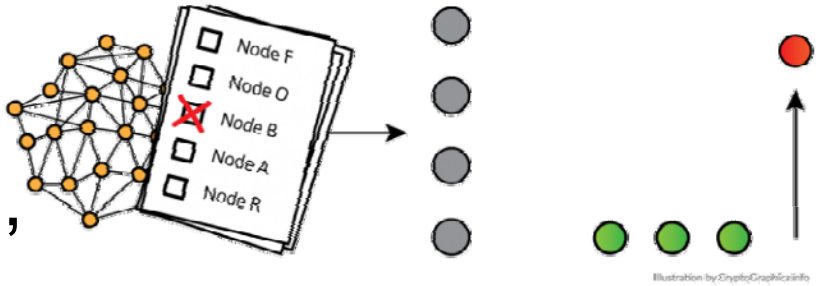
n = Total # of nodes in network

$f = \frac{n-1}{3}$ (Max # of faulty nodes)

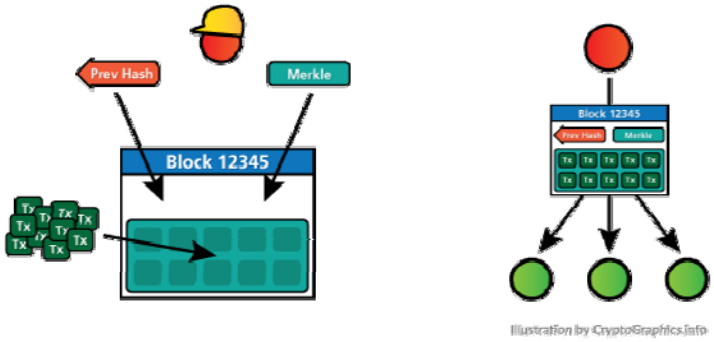
PBFT property

delegated BFT (dBFT)

- Nodes in the network elect a group of consensus nodes (*e.g.*, CoZ)
- Leader/speaker randomly chosen from consensus nodes, remainders are delegates

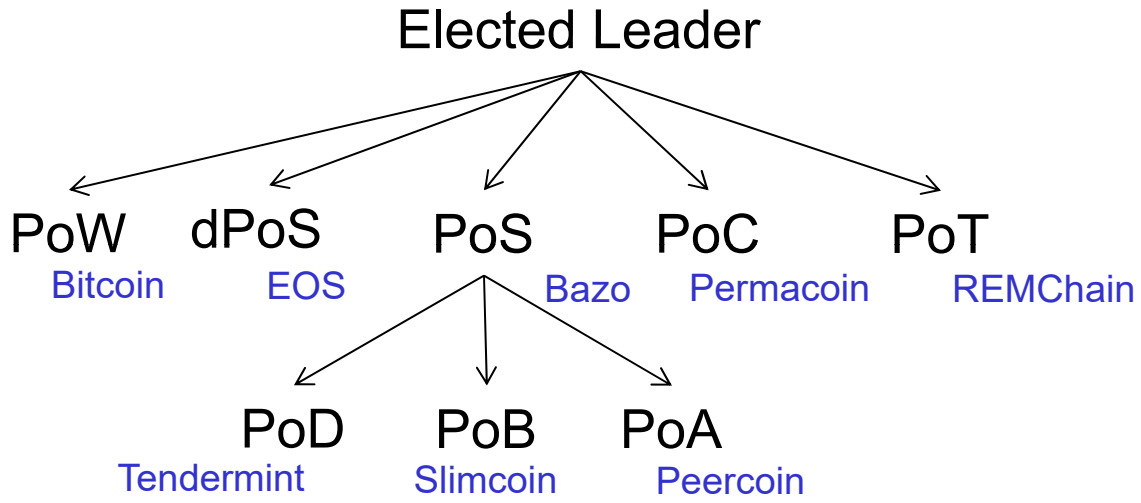


- Leader/speaker creates new block, needs to be positively checked by 2/3 of all delegates
- If 2/3 agree, block is added to the chain
- Countermeasures for dishonest leader/speakers or delegates



Consensus Mechanism Types (2)

- Elected Leader Models
  - Probabilistic elected leader in a
    - Lottery-like
    - Competition, or
    - Probabilistic algorithm



PoX: Proof-of-X, where *X*=

A: Age
B: Burn
C: Capacity (storage)
D: Deposit
S: Stake
T: Trust
W: Work
d: delegated

Proof-of-Work (PoW)

- Set of transactions becomes available, block is created, by utilizing the following data
  - Transaction(s), hash of previous block
  - Nonce (arbitrary number, can only be used once)
  - Other information (depending on BC)
- Hash of new block is calculated
- Checking performed once hash was computed
  - Hash is above the target value → Another miner may have found a suitable hash, block attached to local BC, but miner lost the lottery, otherwise nonce will be incremented, retry
  - Hash is below the target value → This miner won the lottery and the new block's hash determines the PoW result

Partial Hash Collision (PoW)

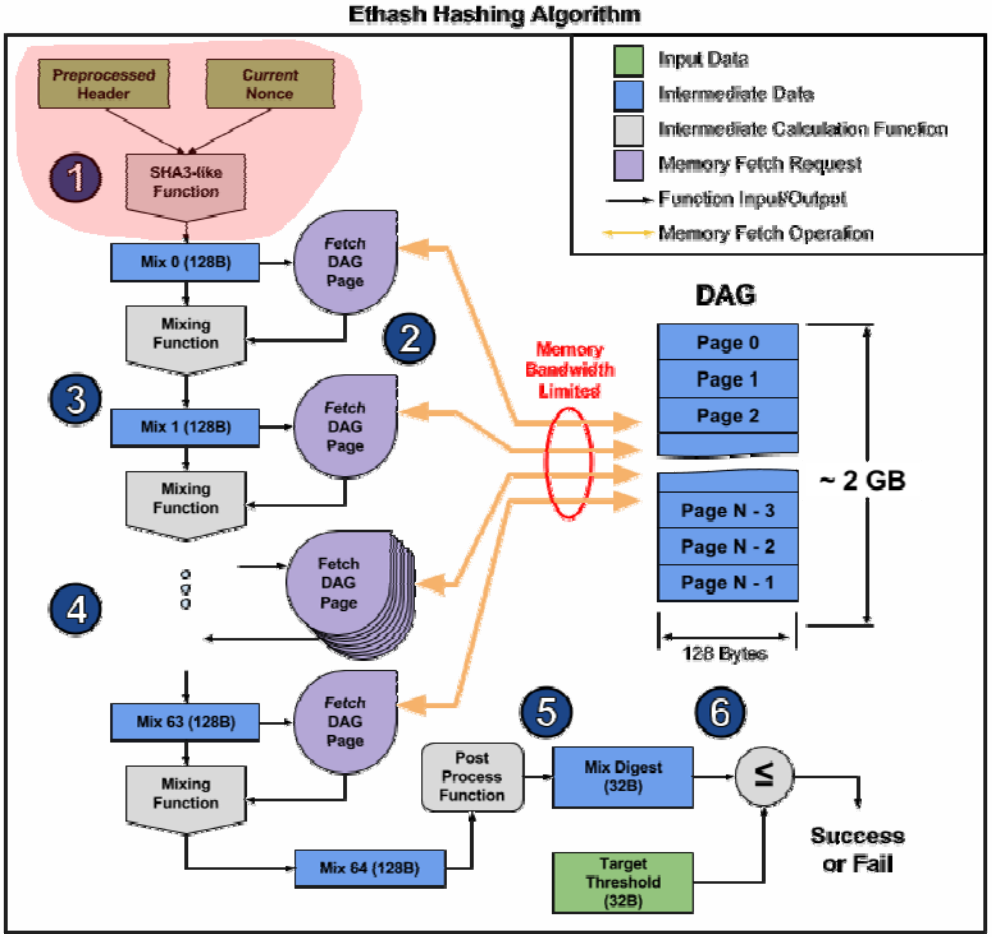
- Key: One cannot compute an input from an output
  - To find a hash with *N* zeros at input start, requires 2**N* computations, which proves computational work performed
  - Hashing an incrementing "nonce" as hash input, leads to zeros

in 3e-05 seconds, nonce = 0 yielded 0 zeros. value = 4c8f1205f49e70248939df9c7b704ace62c2245aba9e81641edf…
in 0.000138 seconds, nonce = 12 yielded 1 zeros. value = **0**5017256be77ad2985b36e75e486af325a620a9f29c54…
in 0.000482 seconds, nonce = 112 yielded 2 zeros. value = **00**ae7e0956382f55567d0ed9311cfd41dd2cf5f0a7137…
in 0.014505 seconds, nonce = 3728 yielded 3 zeros. value = **000**b5a6cfc0f076cd81ed3a60682063887cf055e47b…
in 0.595024 seconds, nonce = 181747 yielded 4 zeros. value = **0000**af058b74703b55e27437b89b1ebcc46f45ce55d6….
in 3.491151 seconds, nonce = 1037701 yielded 5 zeros. value = **00000**e55bd0d2027f3024c378e0cc511548c94fbeed0e….
in 32.006105 seconds, nonce = 9913520 yielded 6 zeros. value = **00000**77a77854ee39dc0dc996dea72dad8852afbde6….
in 590.89462 seconds, nonce = 186867248 yielded 7 zeros. value = **0000000**225060b16117b23dbea9ce6be86ac439d….
in 4686.171007 seconds, nonce = 1424462909 yielded 8 zeros. value = **00000000**2dd743724609a9f57260e2492908d….

- Distributed game sets the difficulty *N* of the game
- Players accumulate points by creating blocks
  - Hashing the previous block, finding a hash of the new block with enough zeros, and transmitting this block to everyone

Ethereum PoW – Ethash

- Also based on partial hash collision (target hash)
  - Works as a memory-bound cryptographic sponge
  - Requires heavy fetching of data on the DAG (RAM intensive process)



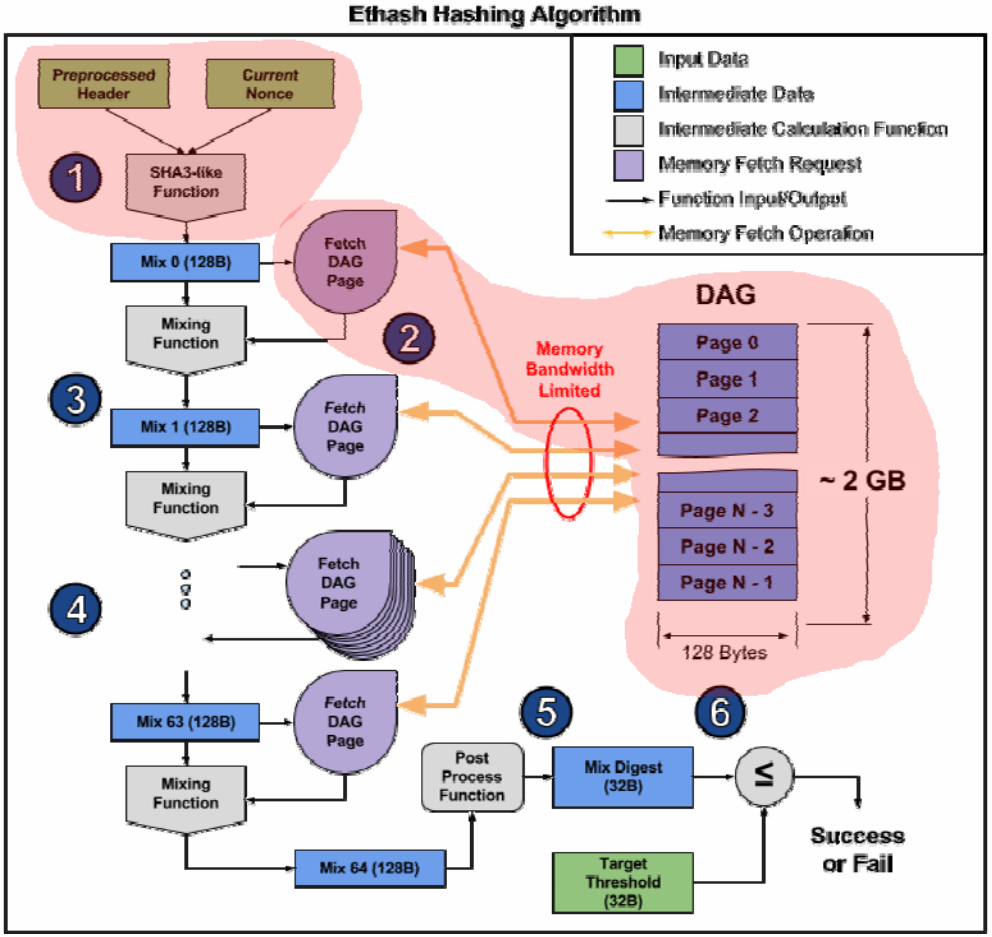
- DAG (Directed Acyclic Graph) - randomly generated dataset
- Designed to be ASIC-resistant as it only require large amounts of memory, which favors smaller miners

1

The **Pre-processed Header** (derived from the latest block) and the **Current Nonce** (the current guess), are combined using a SHA3-like algorithm to create our initial 128 byte mix, called **Mix 0** here.

Ethereum PoW – Ethash

- Also based on partial hash collision (target hash)
  - Works as a memory-bound cryptographic sponge
  - Requires heavy fetching of data on the DAG (RAM intensive process)



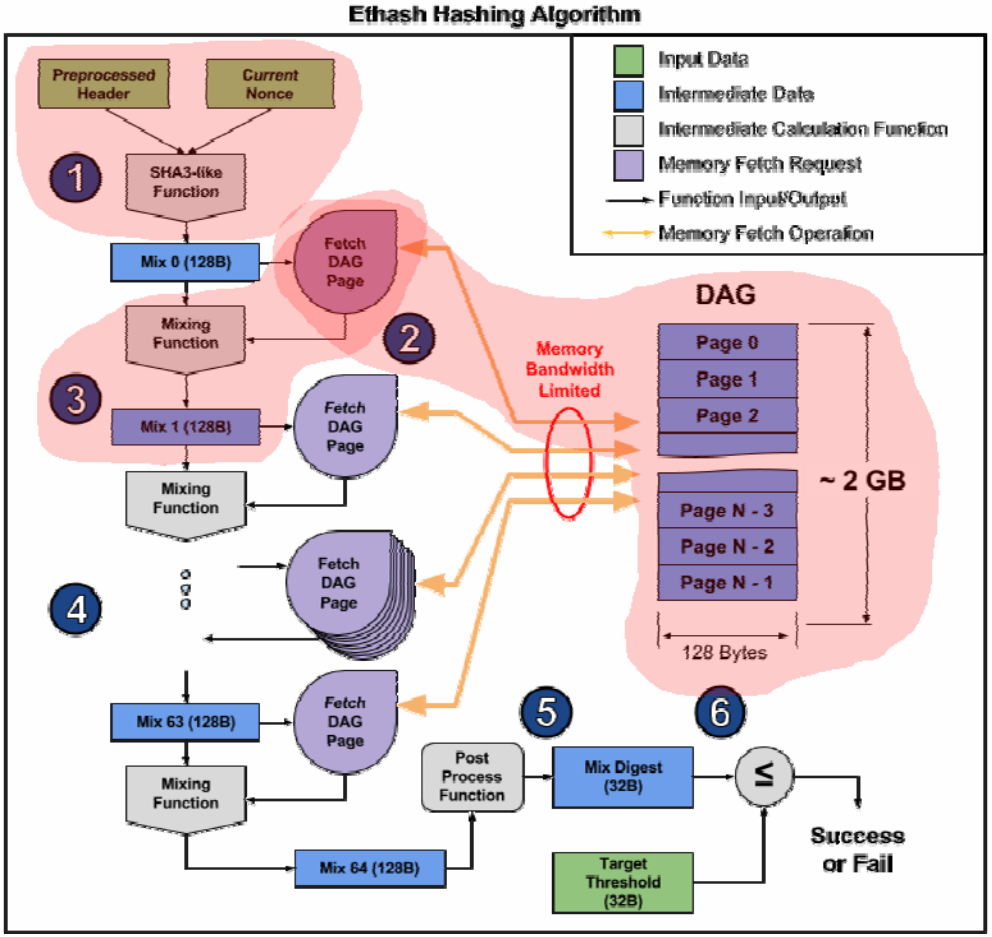
1 2

The **Mix** is combined with the retrieved DAG page. This is done using a ethereum-specific mixing function to generate the next mix, called **Mix 1** here.

- DAG (Directed Acyclic Graph) - randomly generated dataset
- Designed to be ASIC-resistant as it only require large amounts of memory, which favors smaller miners

Ethereum PoW – Ethash

- Also based on partial hash collision (target hash)
  - Works as a memory-bound cryptographic sponge
  - Requires heavy fetching of data on the DAG (RAM intensive process)



1 2 3

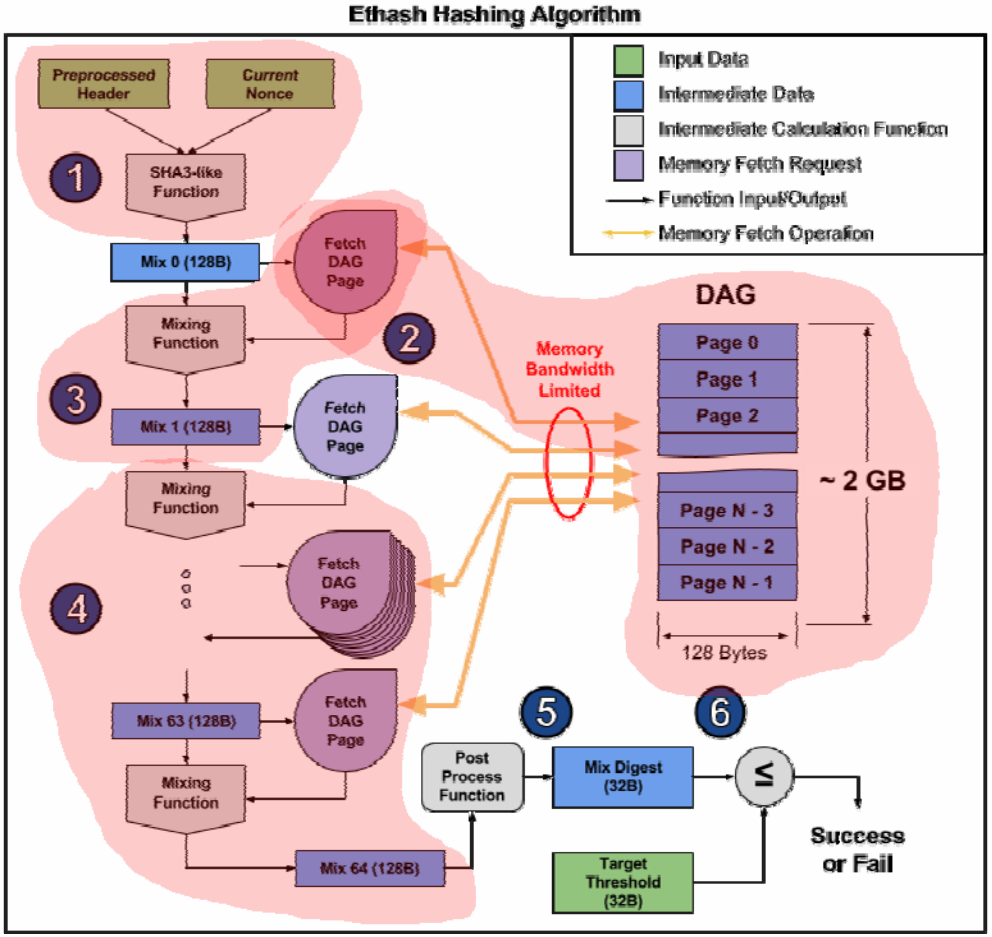
The **Mix** is used to compute which 128 byte page from the DAG to retrieve, represented by the **Get DAG Page** block.

- DAG (Directed Acyclic Graph) - randomly generated dataset
- Designed to be ASIC-resistant as it only require large amounts of memory, which favors smaller miners

Ethereum PoW – Ethash

- Also based on partial hash collision (target hash)
  - Works as a memory-bound cryptographic sponge
  - Requires heavy fetching of data on the DAG (RAM intensive process)

Steps 2 & 3 are repeated 64 times, finally yielding **Mix 64**.

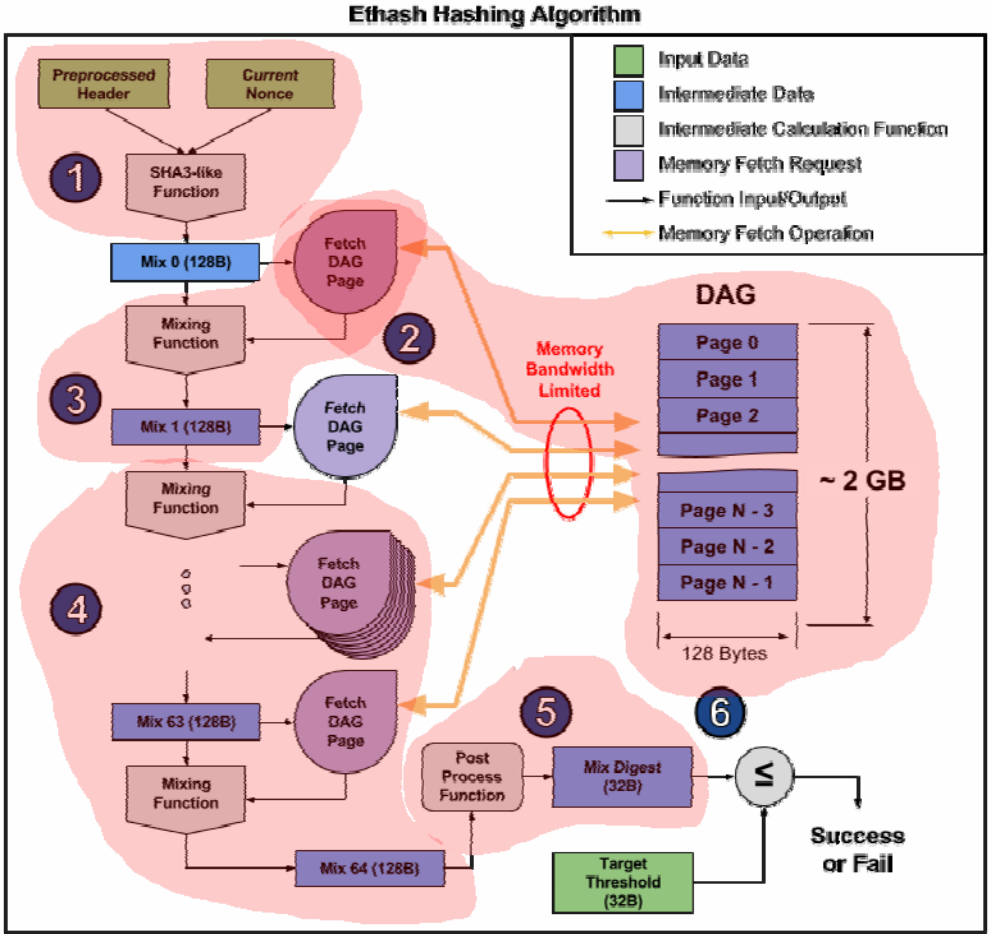


- DAG (Directed Acyclic Graph) - randomly generated dataset
- Designed to be ASIC-resistant as it only require large amounts of memory, which favors smaller miners

Ethereum PoW – Ethash

- Also based on partial hash collision (target hash)
  - Works as a memory-bound cryptographic sponge
  - Requires heavy fetching of data on the DAG (RAM intensive process)

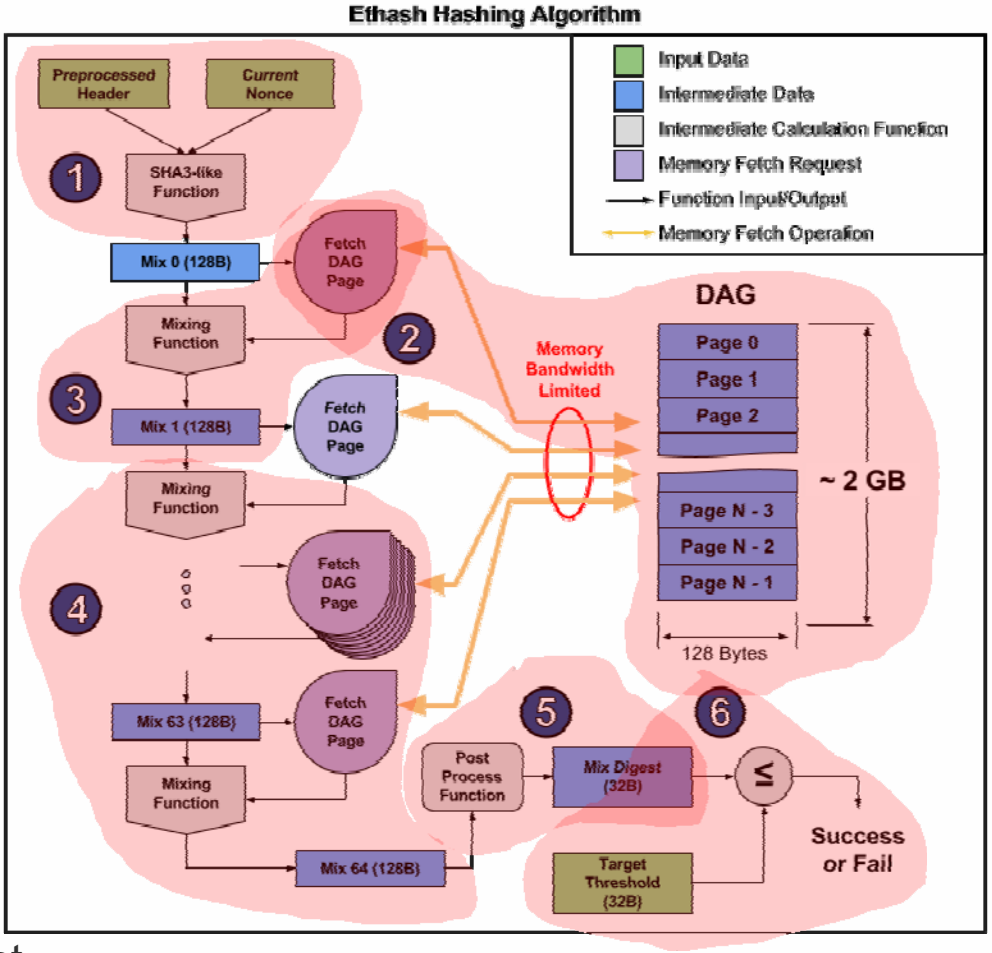
Mix 64 is post processed, yielding a shorter, 32 byte **Mix Digest**.



- DAG (Directed Acyclic Graph) - randomly generated dataset
- Designed to be ASIC-resistant as it only require large amounts of memory, which favors smaller miners

Ethereum PoW – Ethash

- Also based on partial hash collision (target hash)
  - Works as a memory-bound cryptographic sponge
  - Requires heavy fetching of data on the DAG (RAM intensive process)



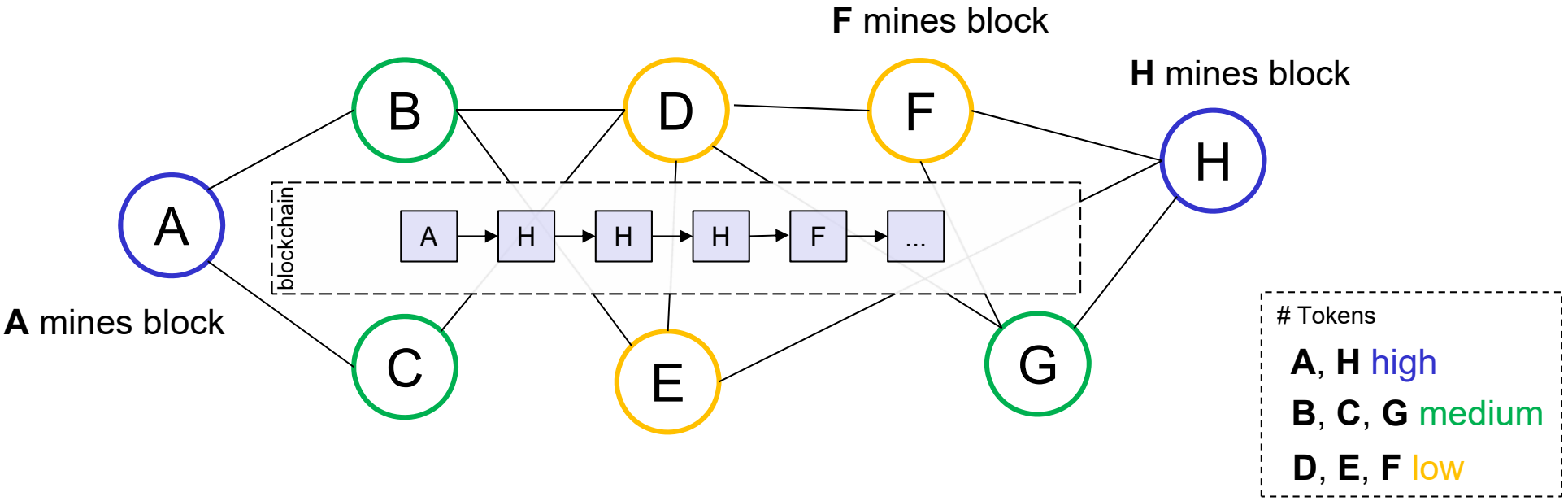
1 2 3 4 5 6

Mix Digest is compared against the predefined 32 byte **Target Threshold**. If **Mix Digest** is less than or equal to **Target Threshold**, then the **Current Nonce** is considered successful, and will be broadcast to the ethereum network. Otherwise, **Current Nonce** is considered invalid, and the algorithm is rerun with a different nonce

- DAG (Directed Acyclic Graph) - randomly generated dataset
- Designed to be ASIC-resistant as it only require large amounts of memory, which favors smaller miners

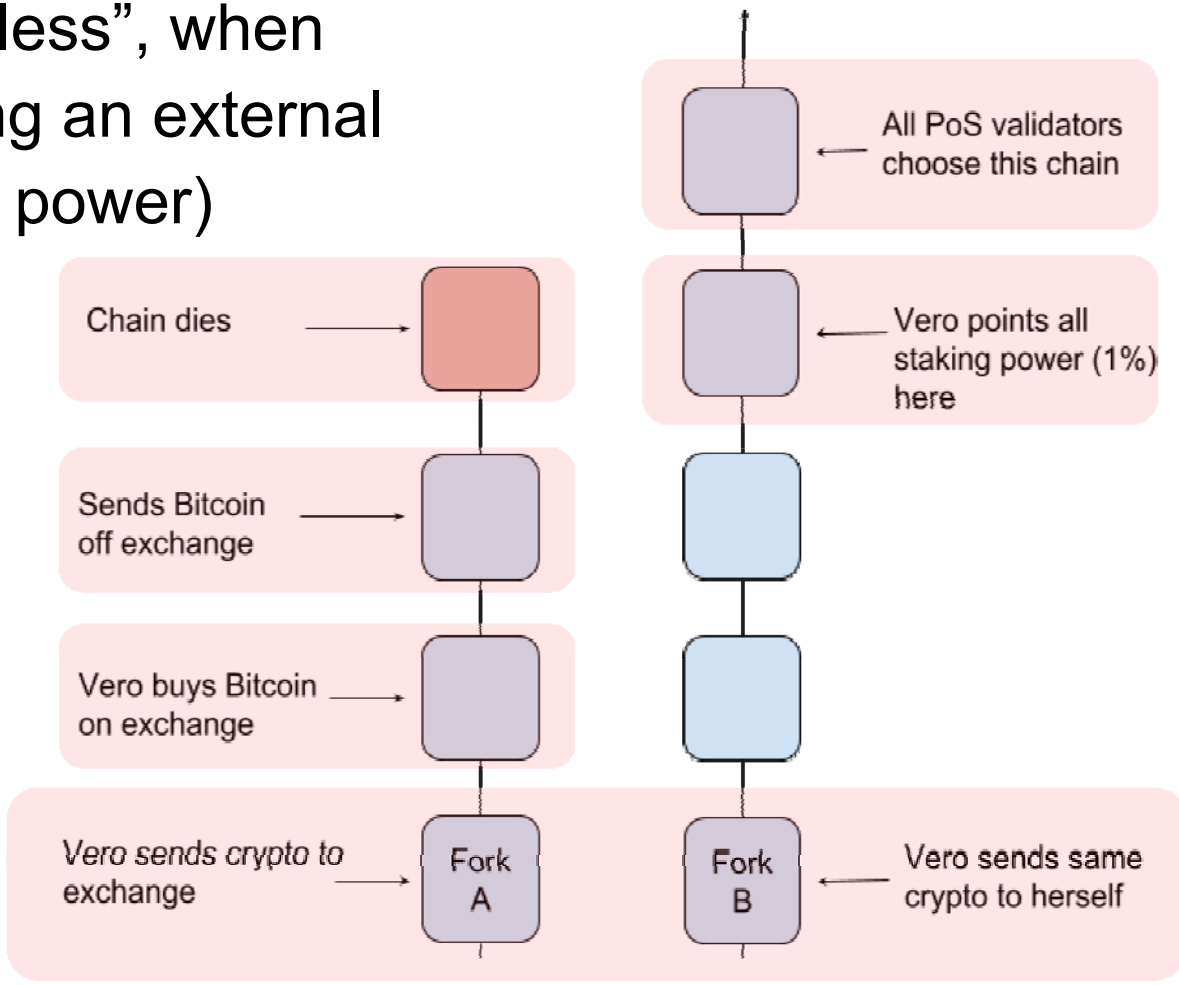
Proof-of-Stake – PoS (1)

- Blocks are “mined” according to the amount of “tokens” he or she holds (stake)
  - The higher is the number of tokens (coins) at stake, the higher is the “mining power” (linear relation)
  - Nodes receive block reward as inherent incentive



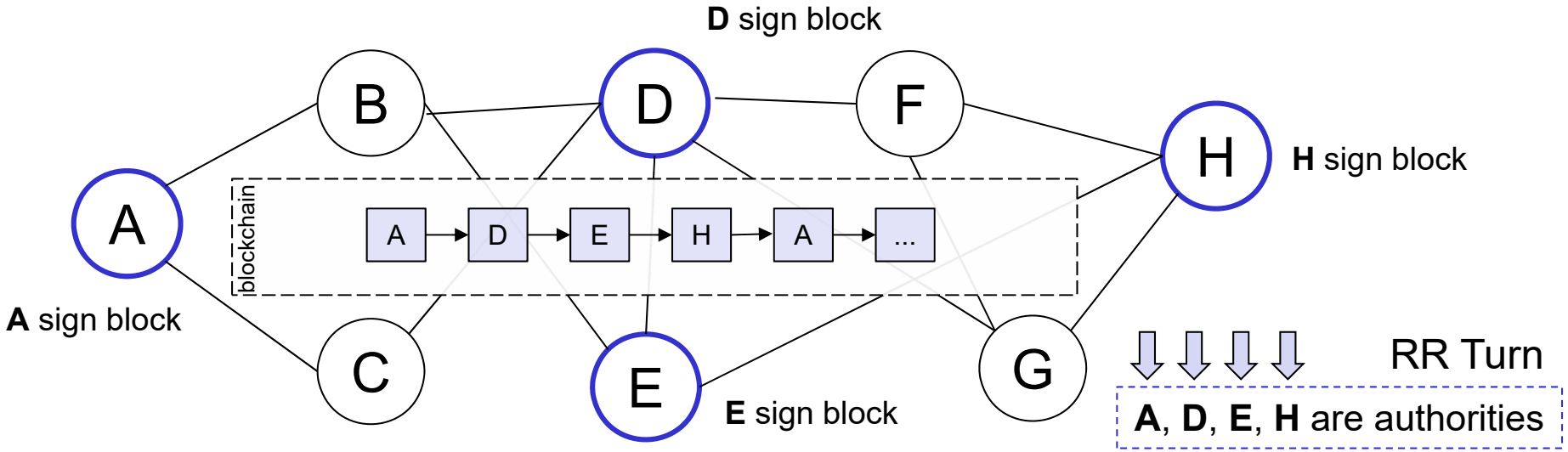
Proof-of-Stake – PoS (2)

- Nothing-at-stake problem
  - Creating forks is “costless”, when someone is not burning an external resource (*e.g.*, mining power)
    - PoS alone is “unworkable” and leads to misuse
- PoS variant:
  - DPoS: nodes can delegate their stakes to a list of nodes



Proof-of-Authority (PoA)

- PoA is a modified form of PoS, where instead of stake a validator's identity performs the role of stake
- Authorities (nodes) are allowed to create news blocks
  - Clique (practical implementation) of PoA
    - Requires N/2+1 (more than 50%) of signers to be honest
    - Authorities sign new blocks in a Round-robin (RR) fashion



Alternatives

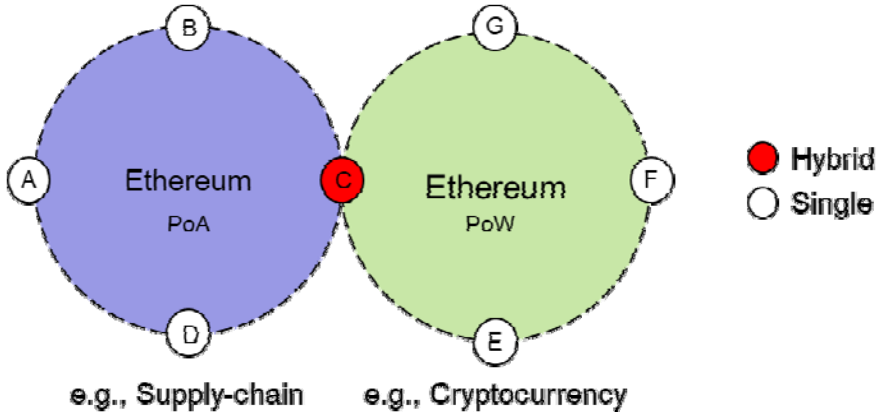
- Proof-of-Burn (PoB)
  - Mechanisms should proof that participants burned “coins”
    - Sending them to a verifiably unspendable address
    - This is expensive from each individual’s point of view
    - But it consumes no resources other than burned asset
    - Ultimate source of scarcity remains PoW-based cryptocurrency

- Proof-of-Capacity (PoC)
  - Also known as Proof-of-Space

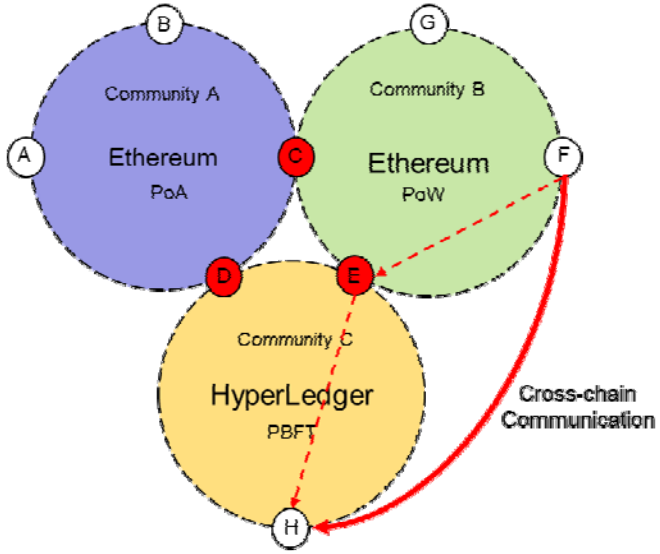
    - Miner has to proof a (free) storage capacity to be entitled to create blocks
    - Miners has to stake their disk (hard-drive or SSD) capacity to create blocks
      - Similar to PoS in this sense

Hybrid Consensus

- Hybrid consensus models
  - Using a single consensus results in limitations
    - Combination of different consensus mechanisms



- Hybrid sharding
  - System can be organized into shards (communities)
    - Cross-chain communications
    - Applied by CSG's Bazo BC (*cf.* later)



Applications of Consensus Mechanisms within CSG Projects

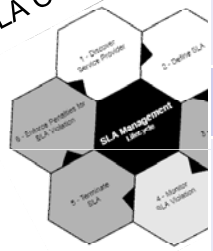
Pollution Monitoring

Trading Contracts

Project	BC Technology	Application Domain	Era	Consensus
IoT-based Pollution Monitoring	Ethereum	Environment	3.0	**PoW**
Blockchain-based Trading Contracts	Ethereum	Marketplaces	3.0	**PoW**
Blockchains for Cold-chains (BC4CC)	Multiple	Interoperability	4.0	**Multiple**
BLW: Foodchain	Ethereum	Food Supply-chain	3.0	**PoW**
Cryptocurrency from Scratch: BAZO	BAZO	Cryptocurrency	1.0	**PoS**
Blockchain Signaling System (BloSS)	Ethereum	Network Security	3.0	**PoA**
Blockchain eVoting	Dedicated BC	eVote	3.0	**PoA**
Automated SLA Compensation	Ethereum	Business Support	3.0	**PoW**

BloSS

SLA Compensation



Comparison of Consensus Mechanisms

Incomplete view!

Mechanism	Security Level	Depending on	Scalability	Remarks
BFT	“Reasonable”: Leader pre-elected 51% failure	-	Medium	Trust in pre-election
dBFT	“Reasonable”: set of leaders pre-elected	-	Medium	Trust in set of leaders
PoW	High: 51% attack	Hashes	Controversial	Energy consumption high, needed to ensure high security level (by design)
PoS	Unknown: “Nothing-at-Stake”	PoW-based “stake”	Under discussion	“Costless” forking, thus, measurable assets needed
PoA	Identity-based	PoS, PoW	Under discussion	Authorities required
Shards	Unknown	Any Po“X”	Unknown	Communities, interoperability

8. Blockchains and Quantum Security

Blockchain Security

- Security is based on existing (a) asymmetric cryptography protocols and (b) hashing schemes

The Principle is to create computationally expensive problems, i.e., NP-hard problems that can be solved in exponential time:

- Discrete logarithm problem: ECC, ECDSA, ECDH, ElGamal
- Factorization of large prime numbers: RSA, Benaloh

- In practice, Bitcoin and Ethereum uses
  a) ECDSA to generate public keys
  b) SHA-256 (Bitcoin), Keccak (Ethereum)

ECDSA: Elliptic Curve Digital Signature Algorithm

Traditional vs. Quantum Computing

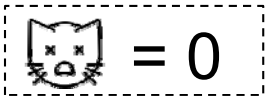
Traditional

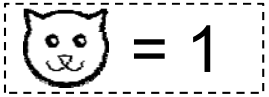
- Based on Bits: stream of electrical/optical pulses represented as 0's or 1's
  - A Bit is either 0 or 1
  - Complex structures are translated in binary

Quantum

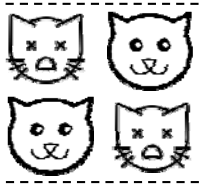
- Based on Qubits: use Quantum Mechanics Properties (QMP) superposition and entanglement to store combinations of 0's or 1's

Schrödinger's cat is either dead or alive but not both

Bit  = 0

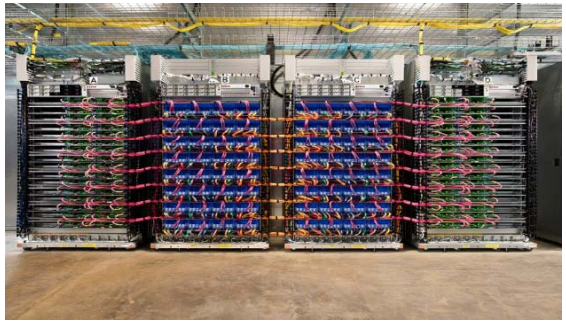
Bit  = 1

Schrödinger's cat can be in any combination of dead and alive at the same time

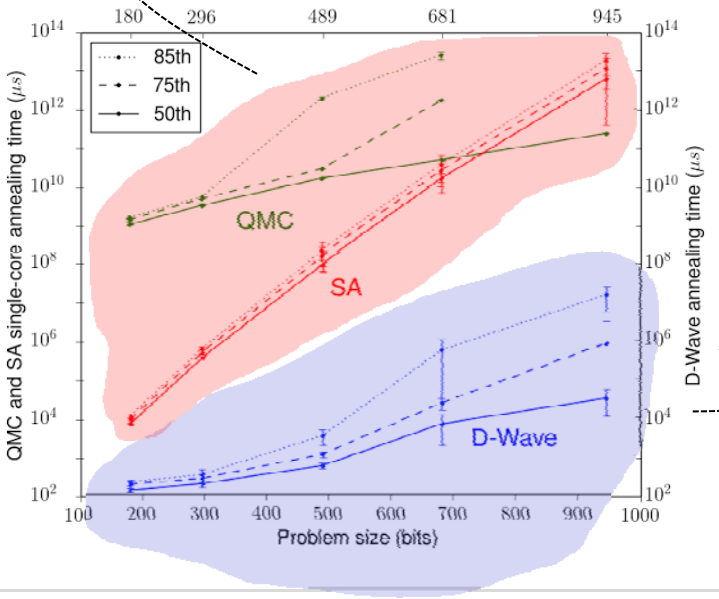
Single Qubit  = 01

= 10

Contrasting the Performance Difference

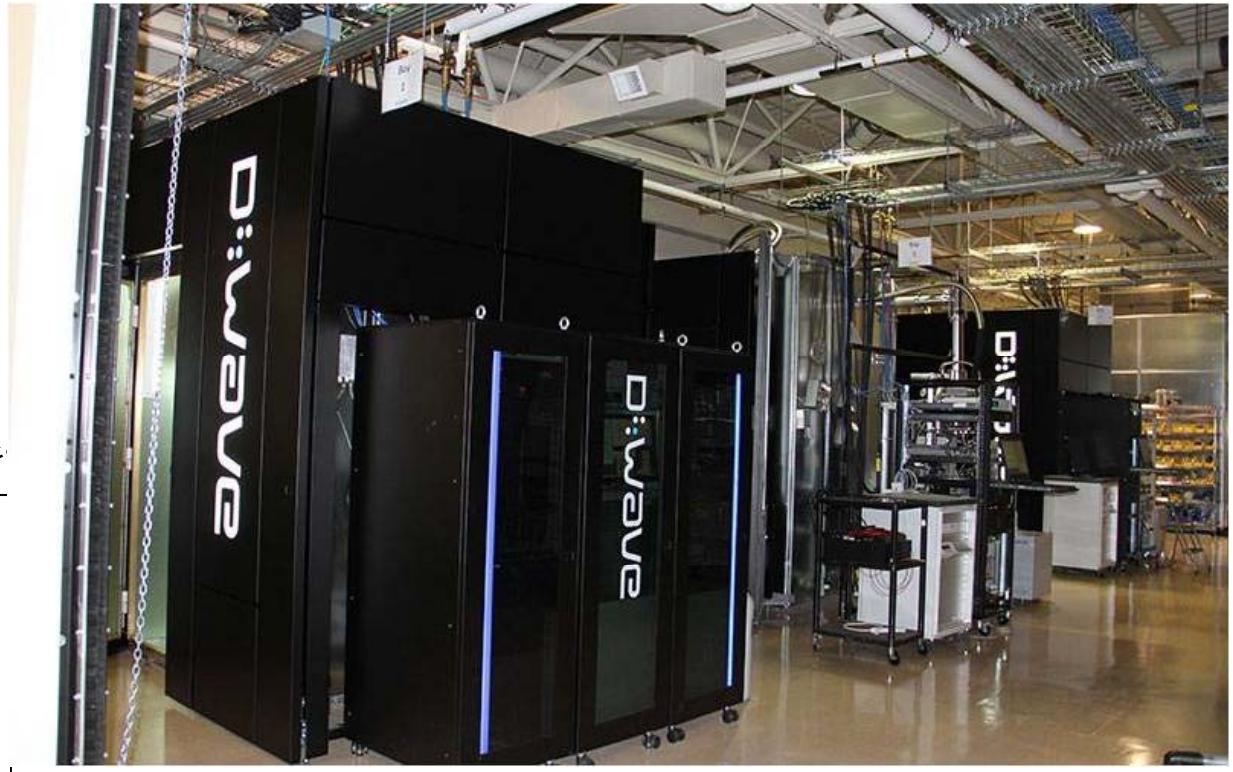


Simulated methods (QMC and SA) running on a traditional Google supercomputer



Lower = F

Google's D-Wave Quantum Computer

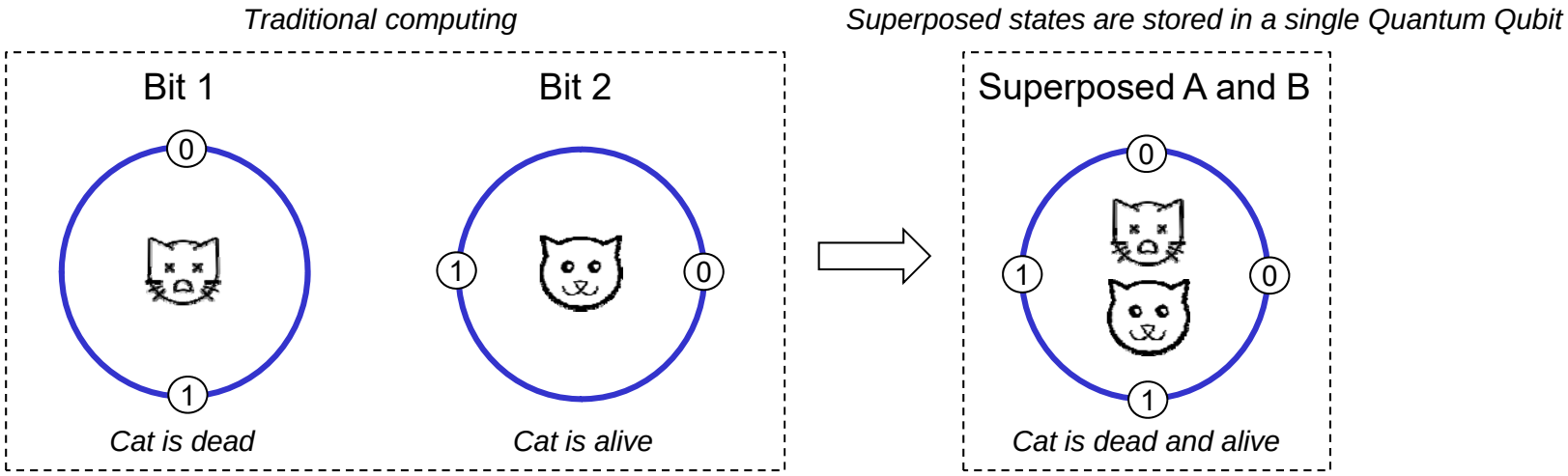


More than **108 times** faster than classical counterparts to solve a combinatorial problem than SA (Simulated Annealing) and QMC (Quantum Monte Carlo)

https://ai.googleblog.com/2015/12/when-can-quantum-annealing-win.html

QMP – Superposition

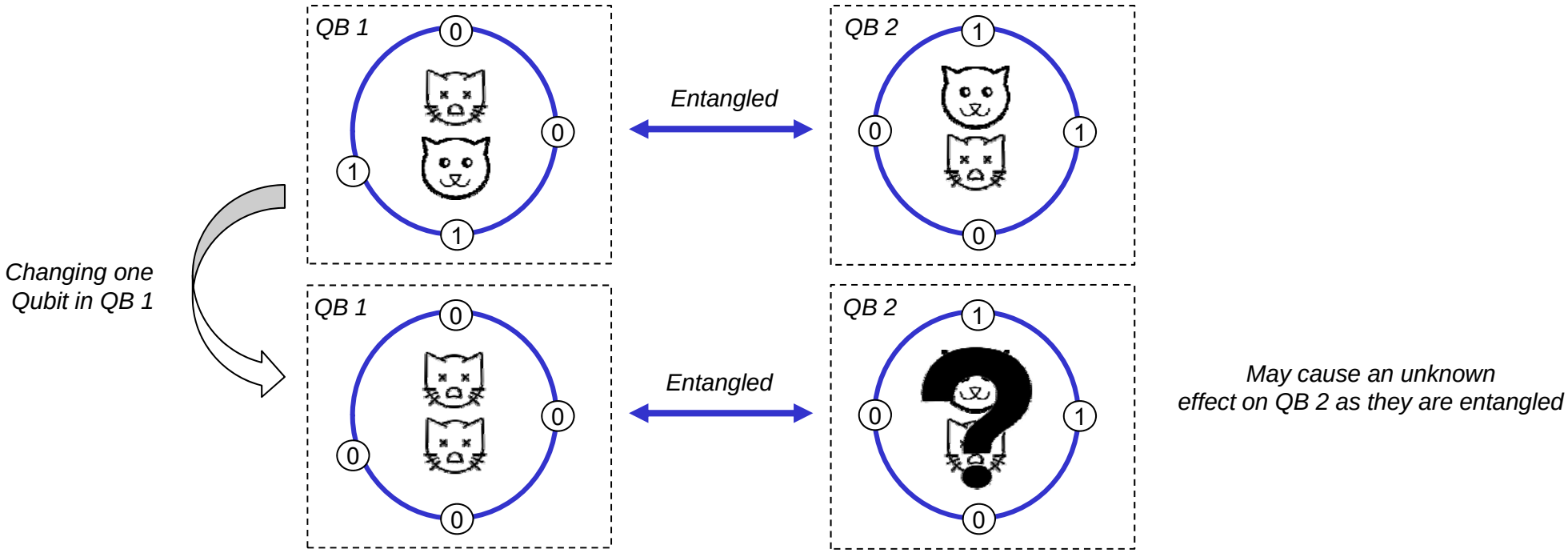
- Two or more states can be combined (superposed) resulting in another valid quantum state



  - Whereas, a bit in traditional computer would only store 0 or 1, so a Cat would be *dead* or *alive*

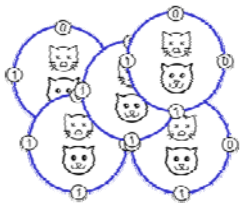
QMP – Entanglement

- Pairs of Qubits (QB) can exist in a way that their quantum state cannot be described independently
  - If the Schrödinger's cat is dead-alive, then the cat is alive-dead.



Impacts on Asymmetric Cryptography

- In theory, existing asymetric protocols would be broken
  - A quantum computer would solve existing exponential problems in polynomial time based on the *Shor's algorithm (1994)*:
    - A theoretical proof that problems involving discrete logarithms (ECDSA) and factorization of prime numbers (RSA) would change from exponential to polynomial time using a quantum computer
  - In theory, this would make it easier to steal BTC or ETH by deriving private keys from public keys



- In practice, a quantum computer is infeasible to achieve a sufficient scale of Qubits as of today

Impacts on Hash Functions

- In theory, selected existing protocols can survive quantum computers, *i.e.,* they are quantum-resistant
  - The *Groover's algorithm (1996)* can be used to find hash collisions
    - Quantum algorithm to find the probability of an input to produce a particular output
  - A Quantum computer would always win the mining race based on existing output sizes of SHA-256 and Keccak
- In practice, SHA-256 (SHA2) or Keccak (SHA3) would would resist by
  - Increasing the output hash size
  - Raising the mining difficulty

If and only if a quantum computer with sufficient Qubits scale is available

Advantages and Disadvantages

- Main quantum computing advantage
  - Greater Performance

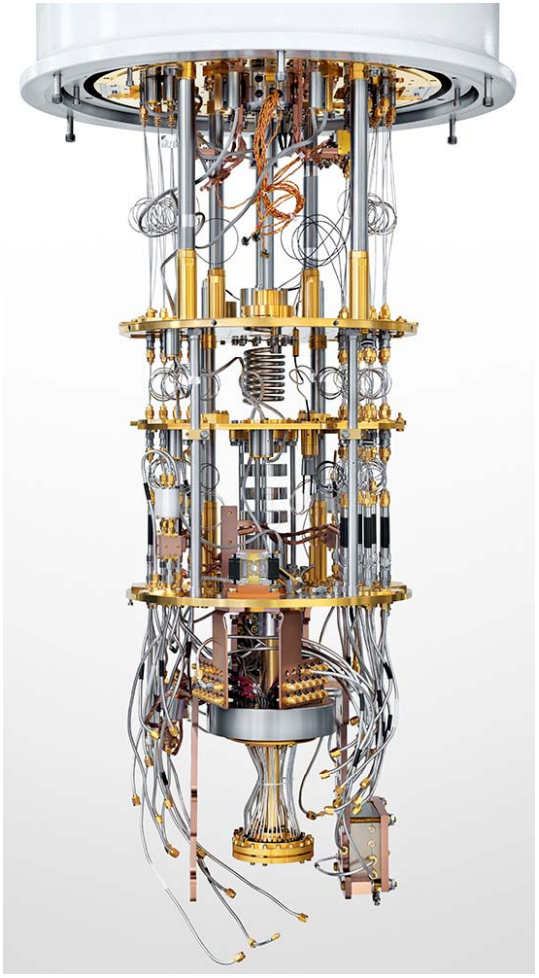
- Superposition: Qubits store more values of 0's and 1's simultaneosly

- Entanglement: Scale exponentially as more Qubits are added (entangled)

- Result: quantum computers are exponentially faster than traditional computers

- Main disadvantage
  - Lack of reliability

- Decoherence: Qubits are extremely sensible to environment conditions (vibrations or temperature changes), thus, they disturb superpositions



Most powerful Quantum computer as of 2018
Rigetti Computing - 128 Qubits

General Quantum Mechanics Impacts

- Quantum mechanics as of today are a theoretical threat
  - Existing asymmetric cryptography protocols are, in theory, broken
  - However, the entire Internet is based on AES, RSA, ECC 
    - Unsafe Blockchains would **not be the largest problem** at all

... and in reality

  - Large scale Quantum computers are still not reliable 

- Future steps and concerns (BC-related and general IT)
  - Theory is also advancing on post-quantum cryptography
    - But these will not work on traditional computers
  - Transition from traditional to quantum computing not smooth

Part IV: Cryptocurrencies (Technical Perspective)

9. Electronic Payment Systems

Electronic Payments (or Systems)

- Traditional payment approaches
  - Cash, check
- Gap in the past: no transfer of money via the Internet
- Electronic payment approaches
  - Means of running payments electronically over a network, such as the Internet or dedicated networks (IP-based)
  - Example (Systems): Electronic Fund Transfer (EFT), debit/ credit card, ecash, software wallets, smart cards, crypto-currencies, Financial Electronic Data Interchange), interbank clearing (SWIFT)
- Micropayment systems
  - Pre-pay, post-pay, pay-as-you-go (once per transaction)

All with "measurable" centralized elements

Electronic Payments and Cryptocurrencies

- Electronic money created solely by IT means
- Formalized process (centralized or distributed) and software exists, which
  - controls the cryptocurrency's creation,
  - enables and protects transactions, and
  - may hide identities of its users
- Secure, anonymous, and electronic on-line payments determine a set of requirements for any, but at the same time trustful handling approach
  - Similar requirements are posed for any centralized approach

- Advances in cryptography, compute performance, and BCs as an underlying distributed ledger enable fully decentralized electronic payment solutions

Cryptocurrencies (1)

- A form of digital cash with general characteristics
  - Faster, cheaper, reliable than “paper/coin” issued currency
- Main two alternatives for creation of coins
  - Issued by government(s) – centralized approach
    - Involving Central Bank, banks, brokers, end-users
  - Issued in a fully distributed manner – modern cryptocurrency
    - Users transact directly with each other (no involvement of banks)
    - Users store money themselves
- General problems of cryptocurrencies
  - Countermeasures against distributed fraud and manipulation
    - Lack of trust
    - Malicious transactions
    - Double spending

Cryptocurrencies (2)

- Automated prevention mechanisms against fraud
  - Issuing of currency to be traceable & deterministic (algorithm)
  - All transactions can be recorded in a decentralized manner
  - All transactions can be verified, own and everyone's else transactions
    - Cryptographic mechanisms essential
- Blockchains, distributed ledgers or public records, provide the key basis for modern cryptocurrencies
  - They do not require a Trusted Third Party (TTP) nor trust as such, neither banks or users → "trustless"
  - Everyone can trace the money being sent, received, verified, and recorded

Blockchains and Cryptocurrencies

- Cryptocurrencies can either be
  - Mined or
  - Pre-created/burned

Mining Crypto Currency (CRY-M)	Pre-creating Crypto Currency with Distribution (CRY-P)	No Use of Crypto Currency (NCRY)
The result of mining is a block with a reward in the form of crypto currency. Bitcoin and Ethereum reward with bitcoins and ethers, respectively. Some blockchains allow to define various other crypto currencies or assets besides its native crypto currency.	Instead of mining crypto currency, the currency can be pre-created and distributed in an Initial Coin Offering (ICO). The incentive to mine a block is to collect transaction fees. Other variations include "Proof-of-Burn" (PoB) or "Proof-of-Possession" (PoP) using another crypto currency.	Some blockchains do not need any kind of native crypto currency, but allow for overlay assets. Especially private blockchains do not use a native currency.

Cryptocurrency Overview (Never Complete)

~ 2500 cryptocurrencies today:
https://coinmarketcap.com/

Approach	Accessibility	Consensus	Crypto Currency
Bitcoin	Public	PoW/ASIC	CRY-M/Bitcoins
Ethereum	Public	PoW/MEM-HARD	CRY-M/Ethers
Ethereum Casper/Serenity	Public	PoS/PoA	CRY-M/Ethers
Litecoin	Public	PoW/MEM-HARD	CRY-M/Litecoin
Monera	Public	PoW/MEM-HARD	CRY-M/XMR
Lisk	Public	PoS/Del	CRY-M/Lisk
R3 Corda	Private	TE	NRCY
Openchain	Private/sidechain	TE	NRCY/various
IOTA	Public	PoW	CRY-P/IOTA tokens
Eris:DB	Private	PoS	CRY-M/Ethers
Chain Core	Private	TE	NRCY/various
Hyper Ledger	Private	TE/PoW/PoS	NRCY
Nxt	Public	PoS	CRY-P/various
Stratis	Private/sidechain	PoW (PoS in future)	CRY-M/STRAT token
Multichain	Private	PoW	NRCY/various
BigchainDB	Private	TE	NRCY/various
Rootstock	Public/sidechain	PoW (Bitcoin)	CRY-M/Bitcoin-Rootcoin
Counterparty	Public	PoW (embedded Bitcoin consensus)/PoB	CRY-P/various
Ripple	Public	TE	CRY-P/Ripple/various
Stellar	Public	TE/PoP	CRY-P/Lumen/various

Cryptocurrency ≠ Blockchain

Many Coins with Similar Mechanisms

- All electronic, digital, backed by a "scarce" resource
  - Thus, avoiding double spending is the key!
  - Bitcoin: SHA-256 partial hash collision: time, CPU, electricity
  - Ethereum: variant of Dagger-Hashimoto, time, CPU, memory, electricity, miner store dataset: 1 GB, verification needs 16 MB
  - Litecoin: script partial hash collision: time, CPU, memory, electricity
  - Ripple XRP: Unique node list (trusted validators, 1000): Web of trust
  - PPCoin/Peercoin: Proof-of-Stake (PoS)/PoW:
    - Holding 1% will generate 1% of coins
    - 1% inflation per year
    - Energy-efficient PoS

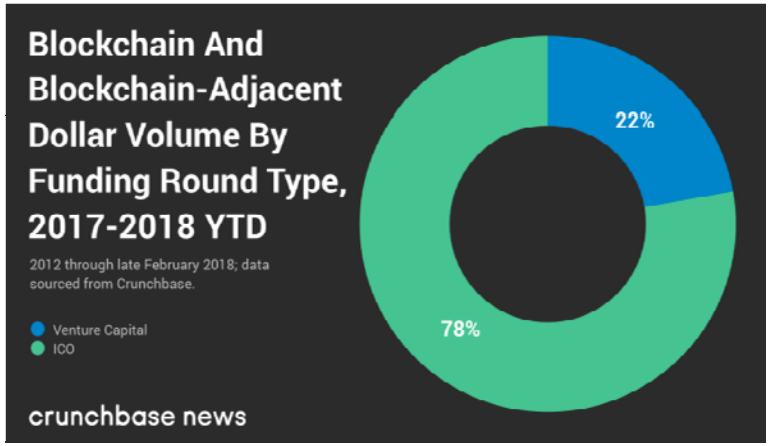
https://coinmarketcap.com/

Initial Coin Offering (ICO)

- Initial Coin Offering (ICO) is a means of crowd funding
  - Token sale: release of a new cryptocurrency, e.g., Ethereum
- It can be done with today's technologies! However, …
  - Micro payments with today's banking system not really feasible
    - Solution: Use central service collection funds until threshold
  - Banking transactions are expensive (sending to Africa/India?)
    - Solution: Increase threshold (*e.g.*, low for SEPA, higher otherwise)
  - IBAN can change – if you change your bank
    - Solution: keep IBANs in sync with users where possible
- Alternatively, use of a blockchain-based approach
  - Publicly, world-wide accessible via cryptocurrency
  - ICOs offer a dedicated "payment system" alternative

ICO Economic Background

- ICO Funding https://www.icodata.io/stats/2018
  - 2019: US$ 346,089,025 (until July 31)
  - 2018: US$ 7,812,150,041
  - 2017: US$ 6,226,689,449
  - 2016: US$ 90,250,273
- ICOs delivered at least 3.5x more capital to blockchain startups than VC (Venture Capital) in 2017+2018



Wall Street Journal:
20 percent of all ICOs are fraudulent
SATIS Group Report:
78% of ICOs are Scams

ICO Pros/Cons and Technical Steps

Fundraising Advantages	Drawbacks	Investor Advantages	Drawbacks
Everyone can participate	Likelihood of failure, 2017 52% of ICOs failed	There is a “token”!	Token may show no value or worse, will be scam
Global audience	Time-consuming technical and admin. set-up	Easy participation for tech-savvy people	Due diligence time-consuming, even is tech-savvy people are involved
	Regulations country-specific		

- Create public/private key pair for pay-ins
  - Store these keys in a bank vault, copy public keys
- Create an ICO backend/frontend
  - Backend with minimal endpoints: tiers, register, address, white list
  - ICOnator: easy, secure, configurable, scalable open source ICO engine

https://iconator.io/

- Minting – aka – create coins out of thin air ☺
- Consolidation: x investors, x accounts → 1 account (KYC!)

KYC: Know Your Customer

10. Bitcoins

WIRED SUBSCRIBE » SECTIONS » BLOGS » REVIEWS » VIDEO » HOW

Sign In | RSS Feeds

FEATURES
For *John Carter*, Director Andrew Stanton Leaps From Animation to Live-Action Sci-Fi

START
MIT's Sebastian Seung Wants Computers to Map the Brain

PLAY
The Five-Year Engagement Takes Director Nick Stoller Off the Grid

/ MAGAZINE

FEATURES 19.12

The Rise and Fall of Bitcoin

759 348 123

By Benjamin Wallace | November 23, 2011 | 2:52 pm | Categories: Wired December 2011

Tweet +1 Share

The Economist

Log in | Register | Subscribe

World politics | Business & finance | Economics | Science & technology | Culture | Blogs

Babbage
Science and technology

Previous | Next | Latest Babbage | Latest from

Virtual currency

Bits and bob

Jun 13th 2011, 20:30 by J.P. | LONDON AND G.T. | MELBOURNE

Like 1.8k Tw

de fr it

Ihr Ort: Zürich 19° Mi 20° Do 26° Übersicht Schweiz

Registrieren Login

20 minuten ONLINE

Video TV Infografik Games E-Prospekte RS

Inhalt A-Z Suchen

Schweiz | Ausland | Panorama | Wirtschaft | Sport | Eishockey-WM | People | Entertainment | Digital | Mehr

News SMI Alle Indices Ratgeber Geld E-Trading PROMOTION Anlageprodukte PROMOTION Finanzrechner Dossiers

From 2011

Ihre Story, Ihre Informationen, Ihr Hinweis? feedback@20minuten.ch

BITCOIN, DIE DEVISE IM WEB

07. Juni 2011 07:15; *Akt: 07.06.2011 09:11*

Der gefährliche Cyber-Dollar

von Gérard Moinat - Die Online-Währung Bitcoin wirft hohe Wellen. Es sei das «gefährlichste Open-Source-Projekt aller Zeiten» und «gefährde

Bitcoins in the News

- March 31, 2017 (NZZ)
  „Die Digitalisierung revolutioniert den Rohstoffhandel“
- March 25, 2017 (Handelsblatt)
  „Wann kommt der Durchbruch? “
- March 28, 2016 (FAZ)
  „Der nächste große Umbruch
  in der Finanzwelt rückt näher “
- April 12, 2017 (WSJ)
  „People Love Talking About Bitcoin More Than Using It“

As of 2017

KOMMENTAR

Bitcoin über 8000 Dollar: Die Spekulationslust ist zurück

Der Bitcoin ist wachgeküsst. Innerhalb von Wochen hat sich die Notierung der Kryptowährung verdoppelt.

Werner Grundlehner
14.5.2019, 16:21 Uhr

In 2018+19

Nach einer langen Agonie über fast das ganze Jahr 2018 und einem ebensolchen Jahresbeginn 2019 ist der Bitcoin seit Anfang April von 4000 auf über 8000 $ geklettert.

MEISTGEL

Immer stärk
kalte Enteig

Michael Rasch, F

Bitcoin

Übersicht News Hintergrund Meinung Service

News

Weshalb der Bitcoin ein Comeback feiert

Der Einstieg eines US-Vermögensverwalters in das Geschäft mit Bitcoin ist nur ein Treiber des aktuellen Höhenflugs. Experten warnen, dass mit starken Preisschwankungen zu rechnen ist. Mehr...

Bernhard Kislig. 15.05.2019

Bitcoin durchbricht auch Marke von 8'000 Dollar

Die Kryptowährung setzt ihren Kursanstieg der vergangenen Wochen fort. Mehr...

14.05.2019

Viel Startkapital – ungewisse Zukunft

Stiftungspräsident Ryan Jesperson ist optimistisch, dass das mit einer halben Milliarde Dollar ausgestattete Start-up Tezos seine Probleme überwunden hat. Mehr...

Bernhard Kislig. 04.12.2018

Ist der Bitcoin am Ende?

Fast 60 Milliarden Dollar weniger Wert als noch vor knapp einer Woche: Der Sinkflug vieler Digitalwährungen hat sich auch am Dienstag fortgesetzt. Mehr...

20.11.2018

Bitcoin: Zwei Chancen für neue Kursrekorde

Dass der Bitcoin bald wieder seine Rekordwerte von knapp 20000 $ erreicht viele Beobachter für unwahrscheinlich. Zwei Entscheidungen in den USA könnten aber genau dafür sorgen.

Patrick Herger
22.5.2019, 11:00 Uhr

ERKLÄRT

Bitcoin Hausse vorerst auf Eis gelegt – und jetzt?

Angefangen hat alles vor mehr als zehn Jahren mit einem abstrakten Aufsatz, verfasst unter dem Pseudonym Satoshi Nakamoto. Heute sorgt die Digitalwährung Bitcoin selbst bei normalen Anlegern oft für Bluthochdruck. Die wichtigsten Fakten zum Hype.

Werner Grundlehner / Thomas Schürpf / David Bauer
Letzte Aktualisierung am 15.08.2019

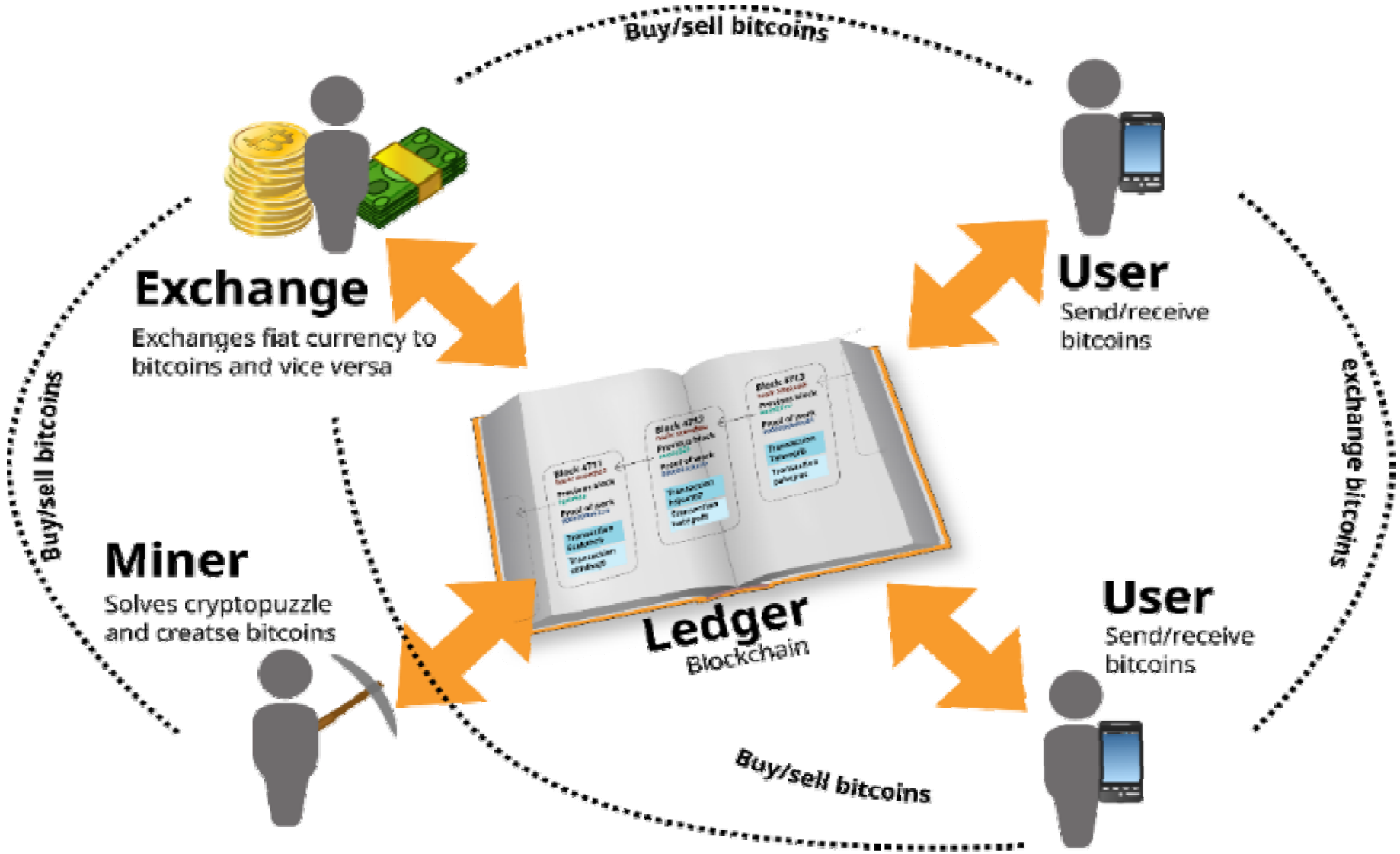
Bitcoin Characteristics (1)

- Bitcoins are an ***experimental*** cryptocurrency (digital)
  - Bitcoin is fully peer-2-peer (no central entity)
  - 1st Bitcoin issued on January 3, 2009
  - Smallest unit: 0.00000001 BTC = 1 satoshi
- Key characteristics
  - Maximum of 21 million BTC
  - Every transaction broadcast to all peers
    - Every peers knows all transactions (~200 GByte as of today)
  - Validation by proof-of-work (partial hash collision SHA-256)
    - Difficult to fake Proof-of-Work (PoW)
    - No double-spending
- The Bitcoin initiator/inventor is unknown so far

Bitcoin Characteristics (2)

- Not relying on trust, but on strong cryptography
- Weak anonymity (pseudonymity)
  - All peers know all transactions
  - Clustering: *e.g.*, if a transaction has multiple input addresses, assume those addresses belong to the same wallet
- Bitcoin not controlled by a single entity
  - Only one development community, but not a central bank – Bitcoin Core's SegWit vs. BTU (Bitcoin Unlimited)
- Creation of Bitcoins performed via the reward for validation of payments (partial hash collisions)
- Bitcoins can be exchanged for real currencies
  - Several companies allow to exchange BTC for Dollar, Euro

The Big Picture – Bitcoin (BTC)



Bitcoin Addresses

- Bitcoin address is an identifier of 26-35 alphanumeric characters, beginning with the number 1 , 3, or bc1 that represents a possible destination for a Bitcoin payment
  - Addresses can be generated at no cost by any user of Bitcoin
- Currently 3 formats
  1. P2PKH which begin with the number `1` , eg: `1BvBMSEYstWetqTFn5Au4m4GFg7xJaNVN2` .
  2. P2SH type starting with the number `3` , eg: `3J98t1WpEZ73CNmQviecrnyiWrnqRhWNLy` .
  3. Bech32 type starting with `bc1` , eg: `bc1qar0srrr7xfkvy5l643lydnw9re59gtzzwf5mdq` .
- Bitcoin transfer: Bitcoin sent to a person by sending bitcoins to one of their addresses
  - Created on demand (offline, too), may be used multiple times
  - Multi-signature addresses: combination of multiple private keys

P2PKH: Pay-to-Publickey-Hash, P2SH: Pay-to-Script-Hash, Bech32: Segregated Witness Address https://en.bitcoin.it/wiki/Address

Bitcoin Mechanisms

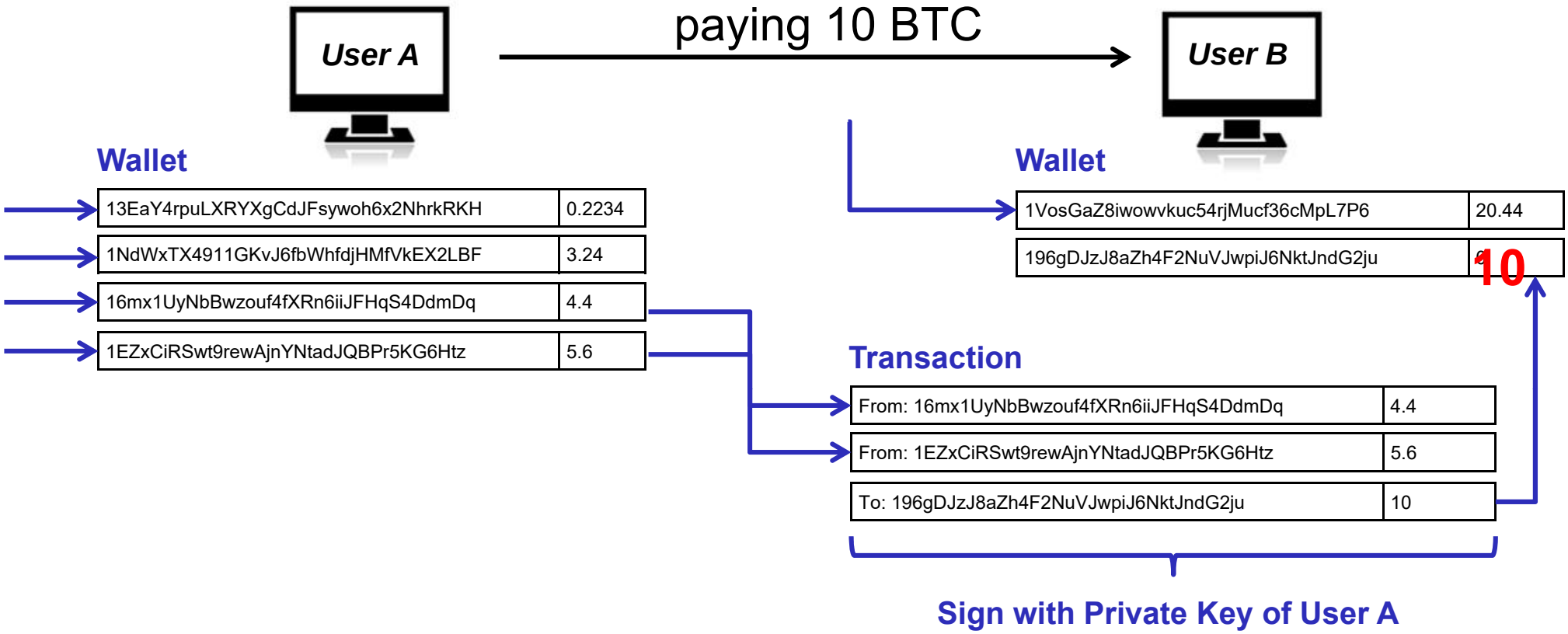
- UTXO (Unspent Transaction Output)
  - Represents a chain of ownership as a chain of digital signatures where the owner signs a message transferring ownership of his UTXO to the receiver's public key
- A wallet holds public-private keys (wallet.dat)
  - Public key, ECDSA 256 bit → Bitcoin address (receives BTC)
  - Private key used for signing transactions

ECDSA: Elliptic Curve Digital Signature Algorithm

- Transaction
  - Peer A to send BTC to peer B → Creates transaction message
  - Transaction contains input/output (BTC from/to)
  - Peer A broadcasts transaction to all peers in the network
  - Transaction stored in blocks → Block is created, verified ~1 min

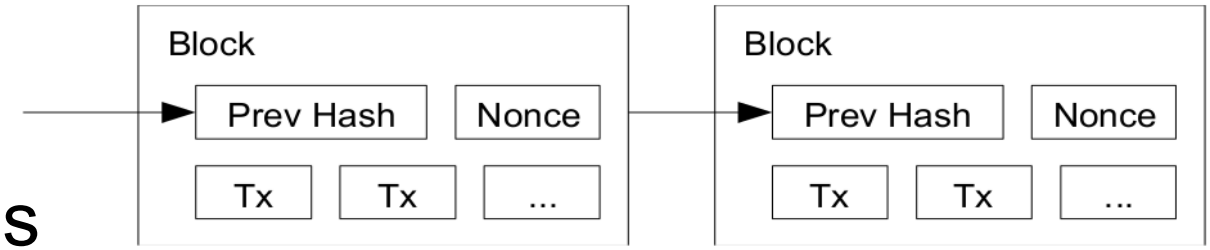
Main Bitcoin Operations

- Private key authorizes the transaction ("access")
  - If keys are stolen, thief may use "your" coins
  - If keys are lost, coins are lost



Double Spending and Coin Generation

- The Bitcoin blockchain avoids double spending by
  - Ensuing that transactions in blocks are confirmed
  - Guessing value that results in number of zero bits is “hard”
    - 00000000000001805ff174586b6acf100f733aaf634e92f9580b4fac9272ed97
  - Chained PoW



- Generation of coins
  - Mining/creating blocks
    - Every time 210,000 blocks are added to the blockchain, mining reward is halved to ensure a steady supply of bitcoins
      → Miner gets currently a 12.5 BTC reward per creation
      – Adjustable difficulty always for 6 blocks/h (on long-term average)
    - May 17, 2020 (app. at 11.14 UTC) the reward will be at 6.25 BTC

Bitcoin Protocol

- Tx in detail

version		01 00 00 00
input count		01
input	previous output hash (reversed)	48 4d 40 d4 5b 9e a0 d6 52 fc a8 25 8a b7 ca a4 25 41 eb 52 97 58 57 f9 6f b5 0c d7 32 c8 b4 81
	previous output index	00 00 00 00
	script length	8a
	scriptSig	47 30 44 02 20 2c b2 65 bf 10 70 7b f4 93 46 c3 51 5d d3 d1 6f c4 54 61 8c 58 ec 0a 0f f4 48 a6 76 c5 4f f7 13 02 20 6c 66 24 d7 62 a1 fc ef 46 18 28 4e ad 8f 08 67 8a c0 5b 13 c8 42 35 f1 65 4e 6a d1 68 23 3e 82 01 41 04 14 e3 01 b2 32 8f 17 44 2c 0b 83 10 d7 87 bf 3d 8a 40 4c fb d0 70 4f 13 5b 6a d4 b2 d3 ee 75 13 10 f9 81 92 6e 53 a6 e8 c3 9b d7 d3 fe fd 57 6c 54 3c ce 49 3c ba c0 63 88 f2 65 1d 1a ac bf cd
	sequence	ff ff ff ff
output count		01
output	value	62 64 01 00 00 00 00 00
	script length	19
	scriptPubKey	76 a9 14 c8 e9 09 96 c7 c6 08 0e e0 62 84 60 0c 68 4e d9 04 d1 4c 5c 88 ac
block lock time		00 00 00 00

http://www.righto.com/2014/02/bitcoins-hard-way-using-raw-bitcoin.html

Bitcoin Script ("Smart Contract")

- Non-Turing complete (*e.g.*, no loops)
  - Very "limited" Smart Contract language
- With scripts
  - Multisig, n-of-m, escrow, and dispute mediation
  - Micropayment channel, refund tx in future
- Opcodes
  - Data operations: OP_PUSHDATA1, OP_PUSHDATA4,...
  - Flow control: OP_IF, OP_ELSE, ...
  - Stack: OP_DUP, OP_SWAP, ...
  - Arithmetic: OP_ADD, OP_ABS, ...
  - Crypto: OP_SHA256, OP_CHECKSIGVERIFY

Bitcoin Scripting Language (Examples)

- ScriptSig
  - PUSHDATA
  - signature data and SIGHASH_ALL
  - PUSHDATA
  - public key data
- ScriptPubKey
  - OP_DUP
  - OP_HASH160
  - PUSHDATA
  - Bitcoin address (public key hash)
  - OP_EQUALVERIFY
  - OP_CHECKSIG

Bitcoin Wallets – Off-line/Paper

- For long-term storage of bitcoin (or giving as gifts) it is not safe to store bitcoin in an exchange or online wallet.
  - Generate new address(es)
    - Private key and QR code
  - Printing own tamper-resistant bitcoin wallets
    - Optional: tamper-evident serialized hologram stickers
  - Example: Bitcoin Cash

  - No crypto-code is run on the Web server
  - No addresses are transmitted over the Internet

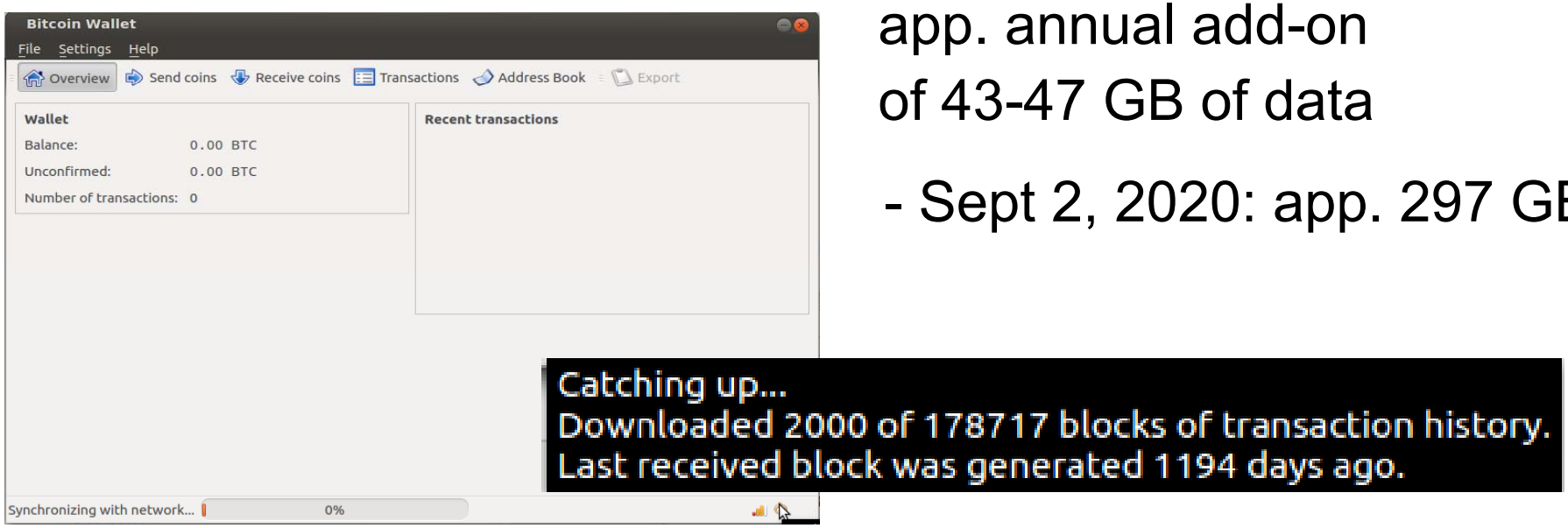
https://bitcoinpaperwallet.com/

Anonymity and Pseudonymity

- Reasonable anonymity with Bitcoin is complicated and perfect anonymity may be impossible
  - Sending and receiving Bitcoins is pseudonymous!
- Wallet deficits – as an example concern
  - Weak privacy aspects due to the pseudonym “Bitcoin Address”
  - Nodes can know almost exactly which addresses constitute accounts
    - Bitcoin users use a new address for each transaction to avoid the transactions being linked to a common owner
    - Best practice, but not sufficient due to multi-input transactions
- Committed bloom filters and heuristics for improved wallet performance and Simplified Payment Verification (SPV) security

Bitcoin Software

- Bitcoin is also the name of the software
  - 2012: app. 2 hours and 1.8 G less disk space later, full Bitcoin BC downloaded
  - 2013: 8 GB, 2014: 19 GB, 2015: 36 GB disk space
  - 2016: 71 GB, 2017: 120 GB, early 2019: 220 GB disk space

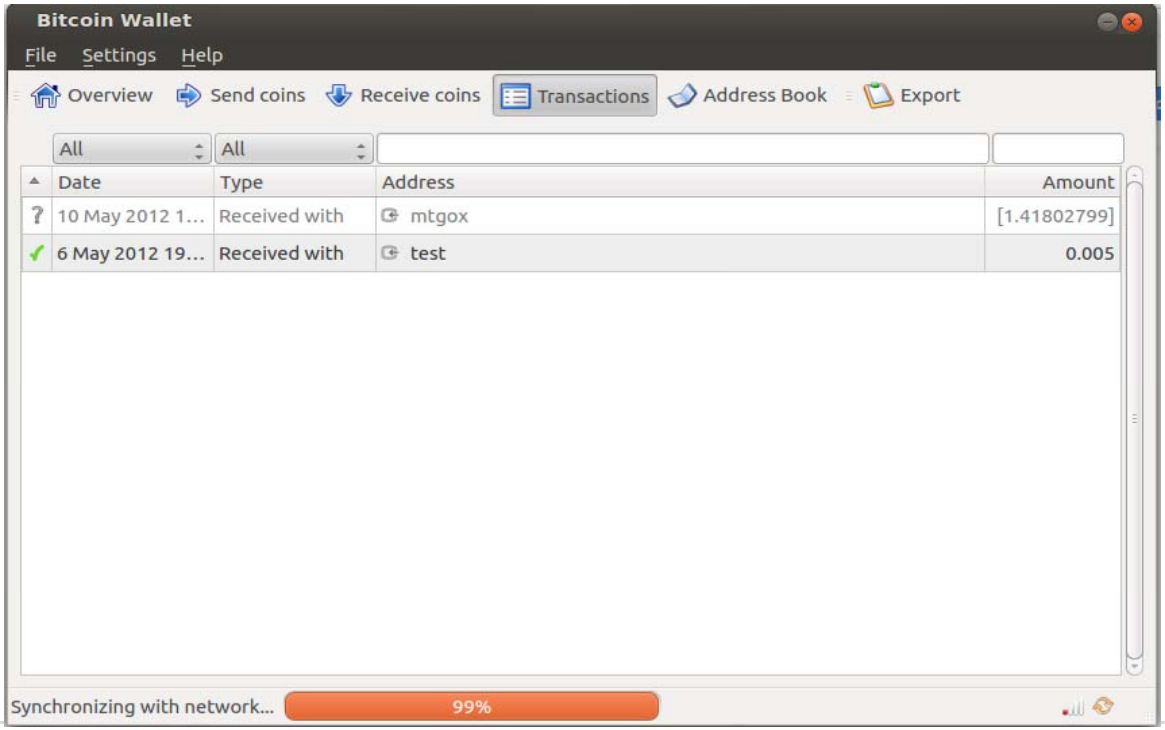


app. annual add-on of 43-47 GB of data

- Sept 2, 2020: app. 297 GB

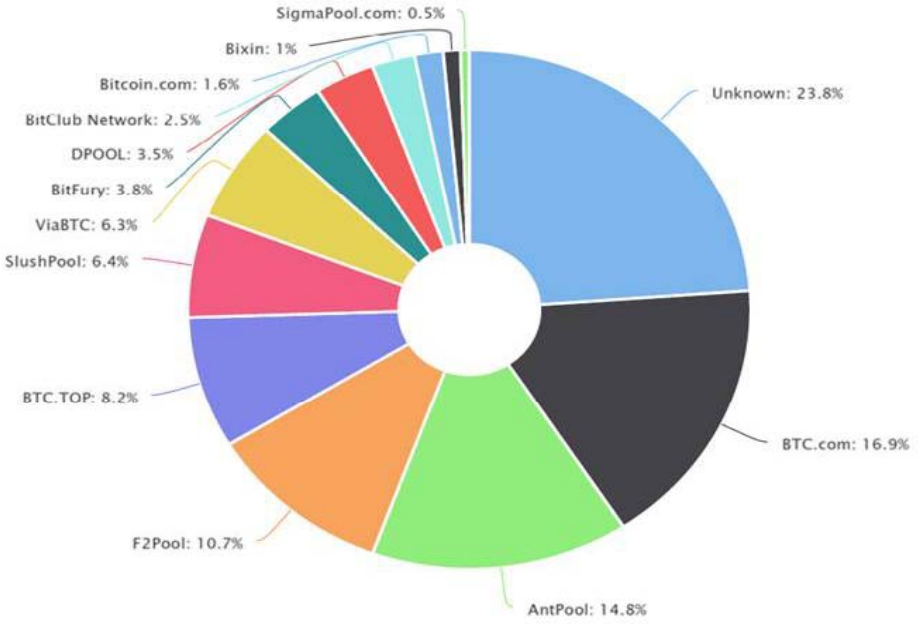
Bitcoin Clients

- 2012: Not easy to add funds … (credit cards or paypal)
- 2018: More market places available
  - http://coinbase.com, http://bitstamp.net, https://www.kraken.com/
  - Not operating: http://mtgox.com, http://bitcoin-24.com, http://bitfloor.com, http://bitcoin-central.net



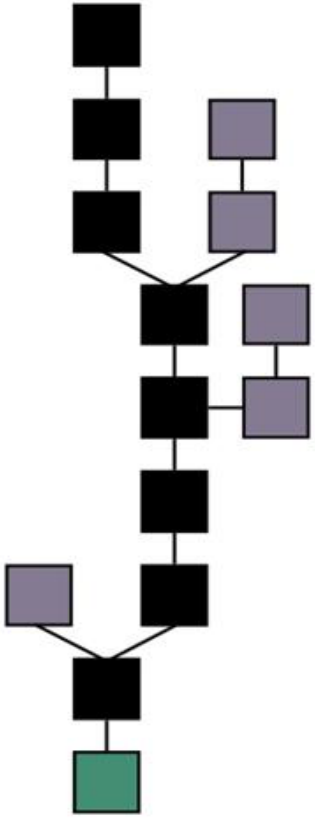
Bitcoin Mining (1)

- Couple of big miners
  - Miners specialized, AMD GPUs, FPGA, ASIC (Application-specific Integrated Circuit) [1][2][3]



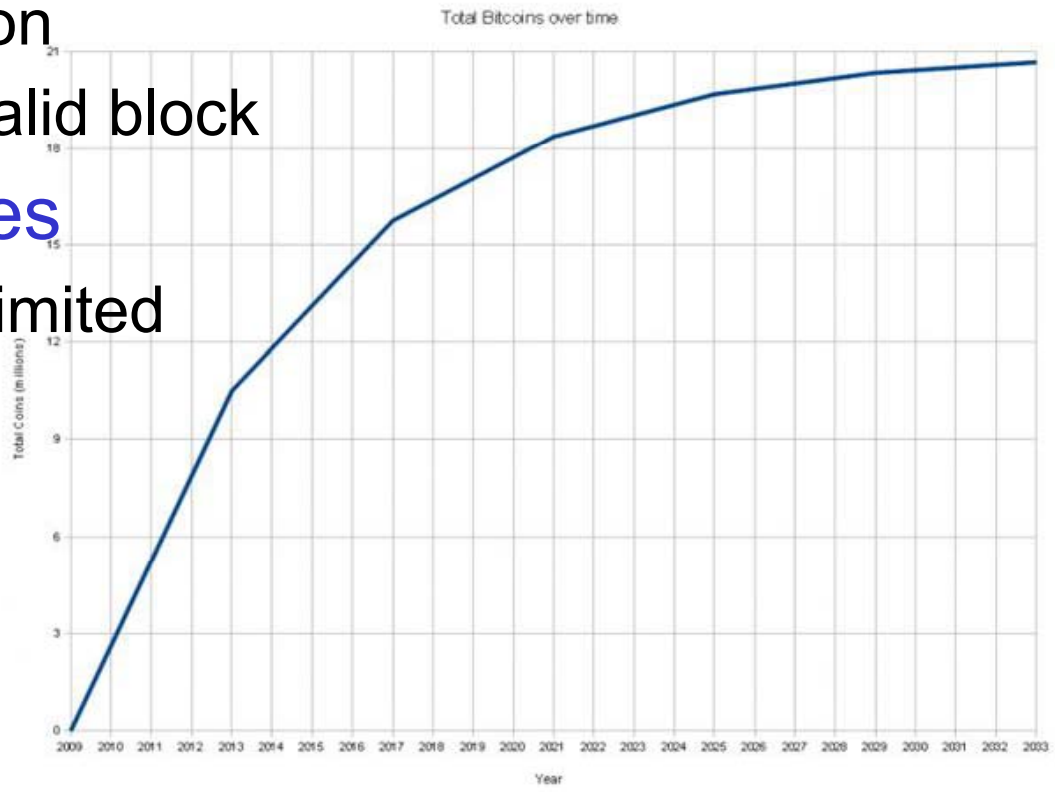
http://blockchain.info/pools

- Mining = creating valid blocks
- Blocks are linked to previous blocks
  - Longest block survive (most difficult)
- Different level of confirmations
  - 6 block conf. is considered secure



Bitcoin Mining (2)

- Mining reward
  - Halves every 210.000 blocks → max. 21 million BTC
  - 12.5 BTC per block creation
  - Competition for creating valid block
- Miner receives also tx fees
  - Required, since BTC are limited
- Dangerous, if someone has more than 51% computing power
  - Can exclude and modify ordering of transactions



Mining Evolution (1)

- CPU (Central Processing Unit) ≈ **100 M hashes/s**

https://99bitcoins.com/20-insane-bitcoin-mining-rigs/

Mining Evolution (2)

- GPU (Graphics Processing Unit) ≈ **1 G hashes/s**

https://bitcointalk.org/index.php?topic=7216.560

Mining Evolution (3)

- FPGA (Field Programmable Gate Arrays)
  - **≈ 10 G hashes/s**

http://www.openmobilefree.net/?p=1308

Mining Evolution (4)

- Application-specific Integrated Circuit (ASIC) Mining Forms ≈ **1-10 T hashes/s**
- "Life Inside a Secret Chinese Bitcoin Mine"
  - 4,000 BTC per month

https://www.youtube.com/watch?v=K8kua5B5K3I

- KnCMiner
  - Operator of the giant mining facility in Boden, Sweden
  - Maker of mining computers
  - Bankruptcy

https://www.coindesk.com/kncminer-declares-bankruptcy-cites-upcoming-bitcoin-subsidy-halving

Mining Evolution (5)

- Scenario: “New” ASIC miner
  - Example: ANTMINER S
    - ≈ 13 TH/s, 1300 W

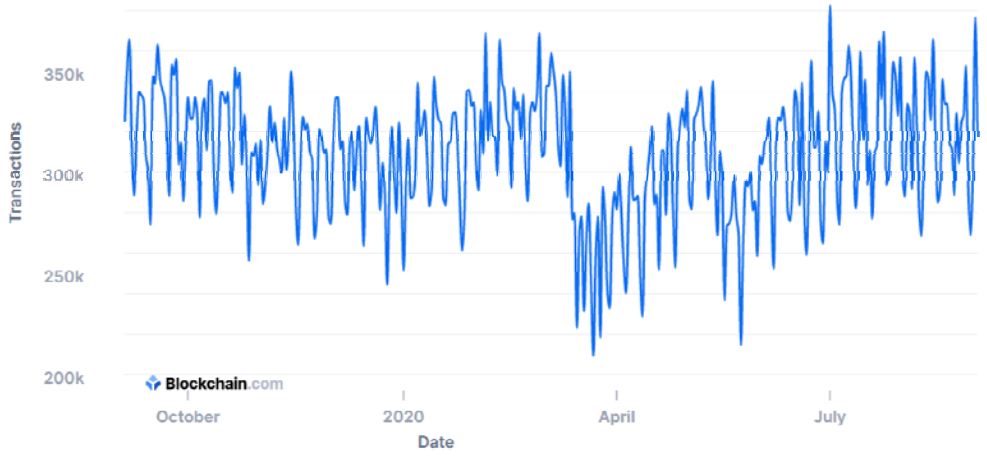
https://shop.bitmain.com/promote/antminer_s9i_asic_bitcoin_miner/overview

  - Difficulty increment: 0.5% or 5.5%?

ANTMINER S7	Very low difficulty increment
Difficulty	520808749422
Rate BTC/USD	1300
Electricity	0.09 / 0.18 CHF/kWh
HW Break Even	264 days

Bitcoin Transactions

- ~ 370,000 transactions per day
  - 3 transactions per second



https://www.blockchain.com/charts/n-transactions

- 300,000 BTC traded per day
  - ~ 6 Billion US$

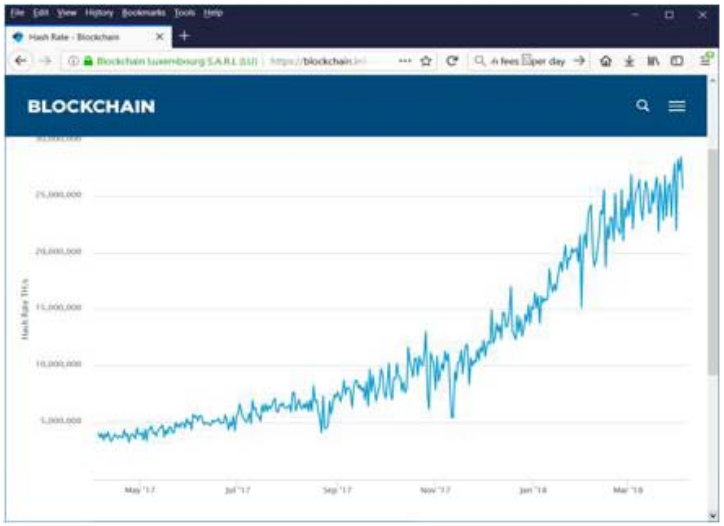
- Transaction fees per day
  - ~ 127 BTC

https://btc.com/stats/fee

Bitcoin Hashrate

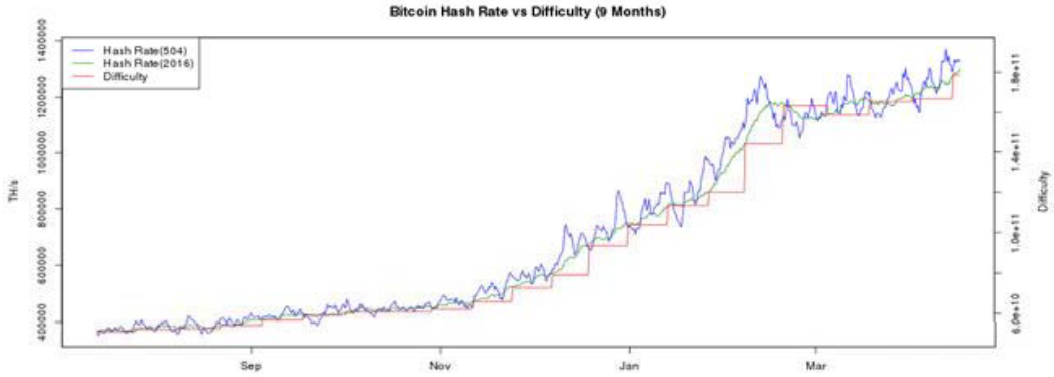
- Network Hashrate (http://bitcoincharts.com)
  - ~ 138.02 EH/s in September 2020

    Exa = “quintillion” (1 Million Tera)
  - ~ 16,500,000 in 2016
  - ~ 4,400,000 in 2015
  - ~ 713,756 in 2014
  - ~ 900 in 2013
  - ~ 155 in 2012



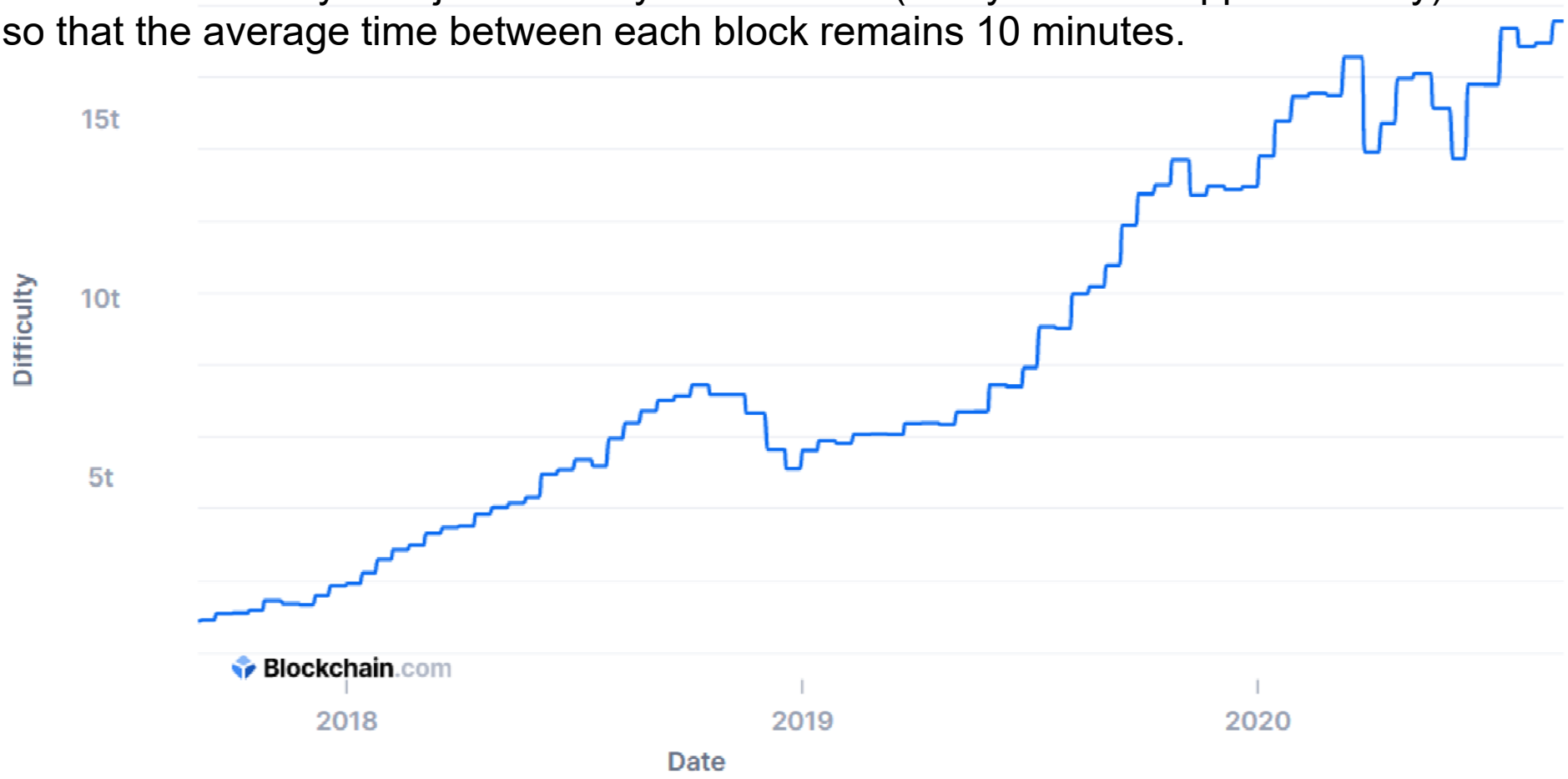
- Fastest supercomputer (top500.org) Sunway TaihuLight
  - – 93 PetaFLOPS (max),

    all 500 machines at ~1291 PetaFLOPS



Bitcoin Difficulty (Target Value)

The difficulty is a measure of how difficult it is to find a hash below a given target (mine).
Note: The difficulty is adjusted every 2016 blocks (every 2 weeks approximately) so that the average time between each block remains 10 minutes.

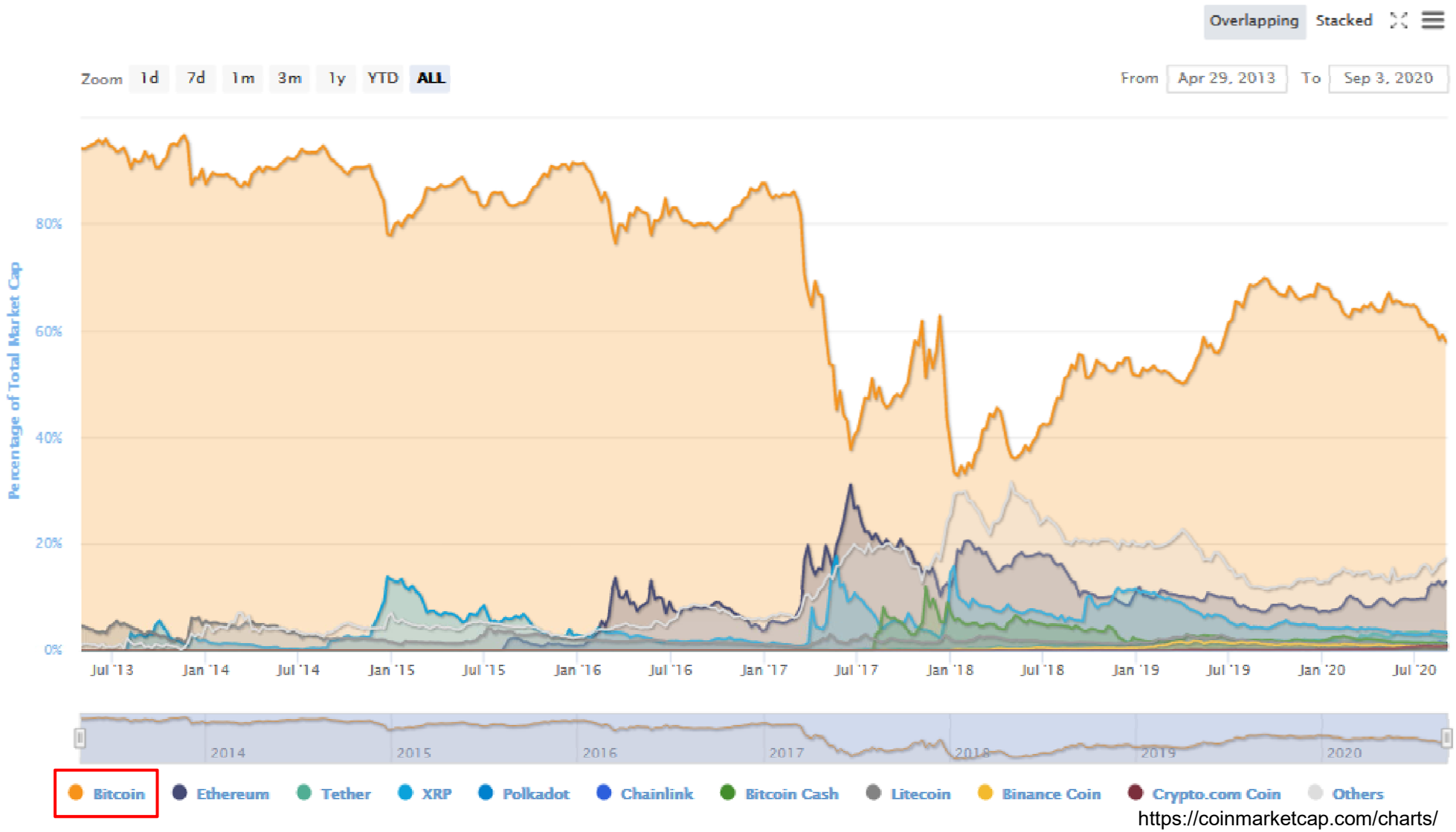


https://www.blockchain.com/charts/difficulty?timespan=all&showDataPoints=false&daysAverageString=1&show_header=true&scale=0&address=

Total Market Capitalization in US$

https://coinmarketcap.com/charts/

Percentage of Totel Market Capitalization (Dominance: BTC 56.3%)



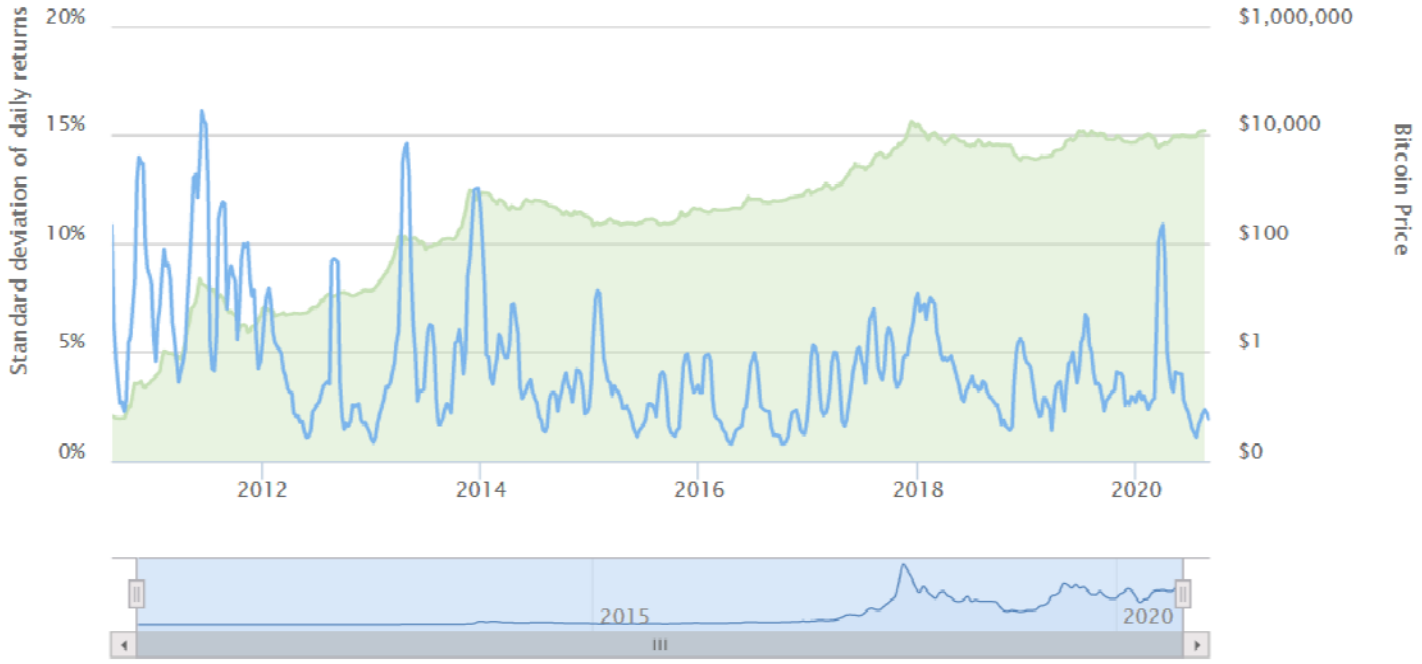
https://coinmarketcap.com/charts/

Bitcoin (BTC) Value and Coins Mined

- 1 BTC ≈ 6,816.79 US$ (April 5, 18), 8,127.52 US$ (April 13, 2018), 11,365.28 US$ (Aug 26, 2020)
- Total of ≈ 18,478,000 BTC mined (September 2, 2020)

https://blockchain.info/charts/total-bitcoins

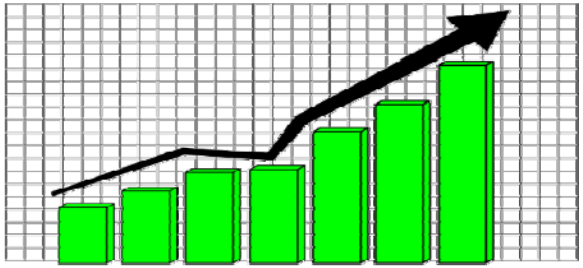
Volatility!



https://www.buybitcoinworldwide.com/volatility-index/

Bitcoin Advantages and Drawbacks (1)

- Advantages
  - Low (fixed) tx fees
    - 10-30 satoshi per Byte
  - Scalable
    - Hardware/storage gets faster

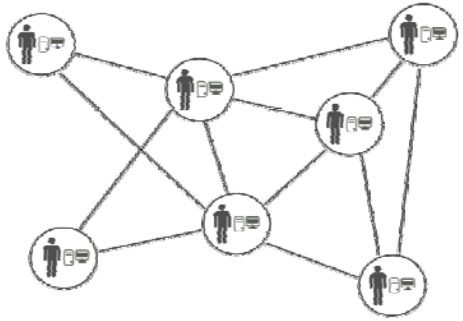


  - Anonymity
    - No privacy concerns/ datamining difficult

- Disadvantages
  - Power consumption
    - ~ 250 - 500 MW (KKW Beznau ~730 MW)
  - Not scalable
    - Bitcoin with 7 tps vs. VISA 57,000 tps (December) [tps: transactions per sec]
  - Anonymity
    - Can be used for illegal activities

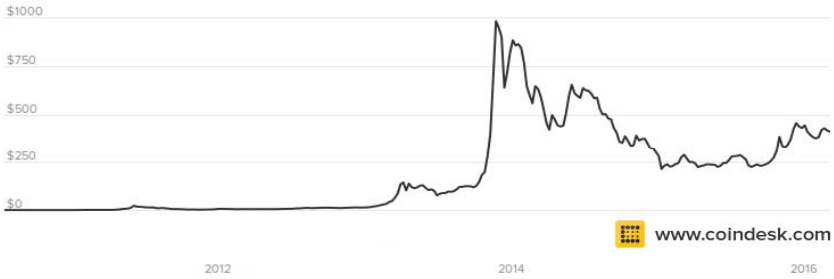
Bitcoin Advantages and Drawbacks (2)

- Advantages
  - No major “crashes”
    - Mt. Gox was an exchange site!
  - Decentralized
    - Open protocol

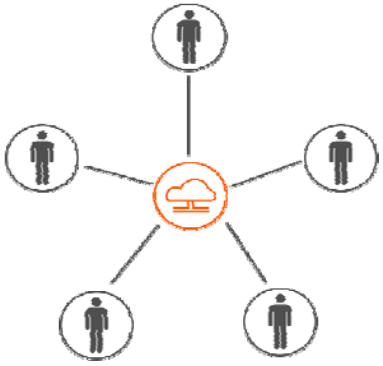


  - Other blockchain usage
    - Smart contracts, many application areas

- Disadvantages
  - Volatile exchange rate



  - Central elements
    - 5 core developers



Thank you for your attention!

End of Day 1:
Introduction to Blockchains