Security of data-intensive applications

Distributed Ledgers

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT INFORMATION SYSTEMS

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Agenda

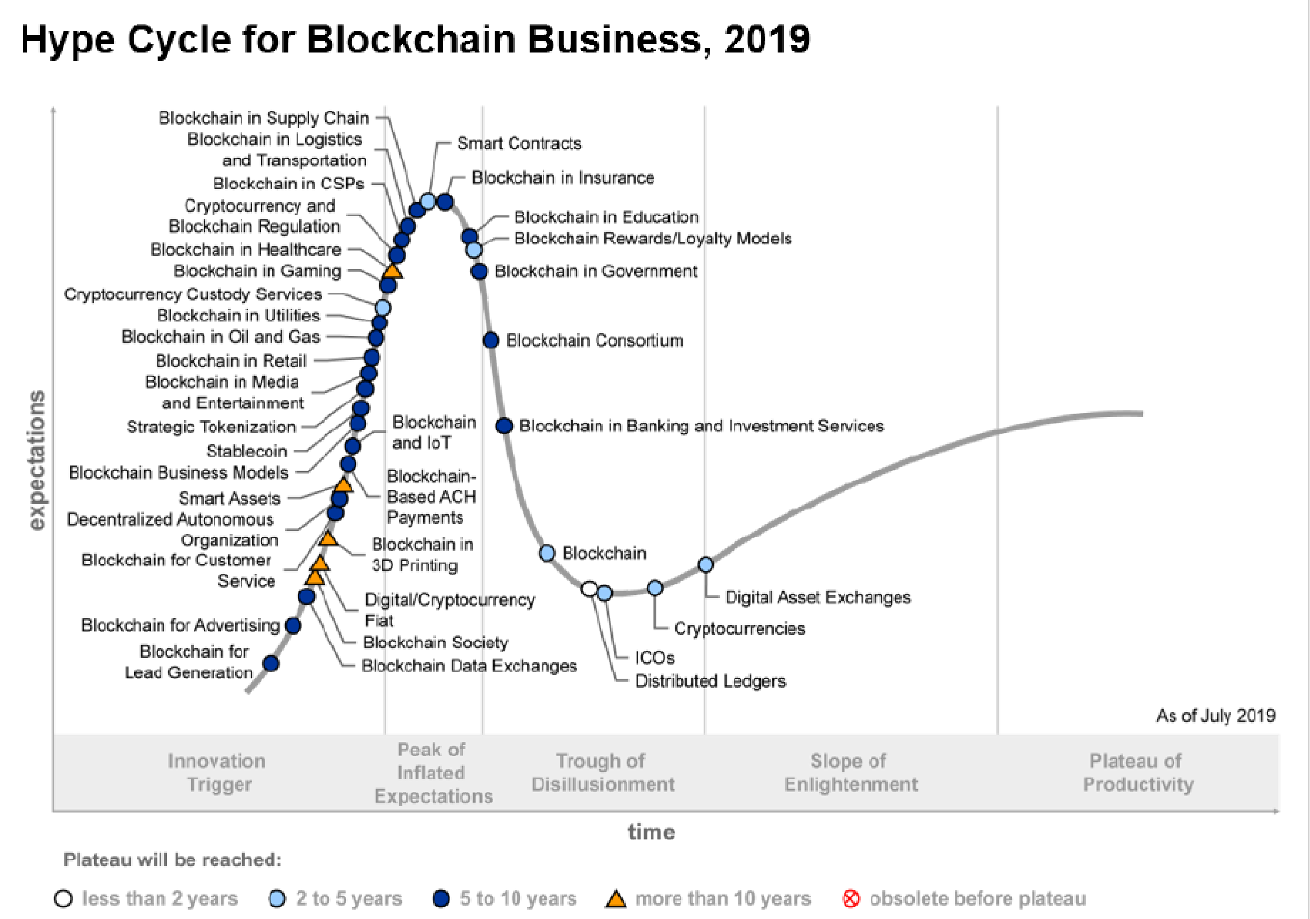
- Motivation
- Distributed Ledger Concepts
- Distributed Ledger Architecture
- Distributed Ledger Use Cases and Limitations

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Motivation: Cryptocurrencies

Motivation: Gartner Blockchain Hype Cycle 2019



Source: Gartner
ID: 390391

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

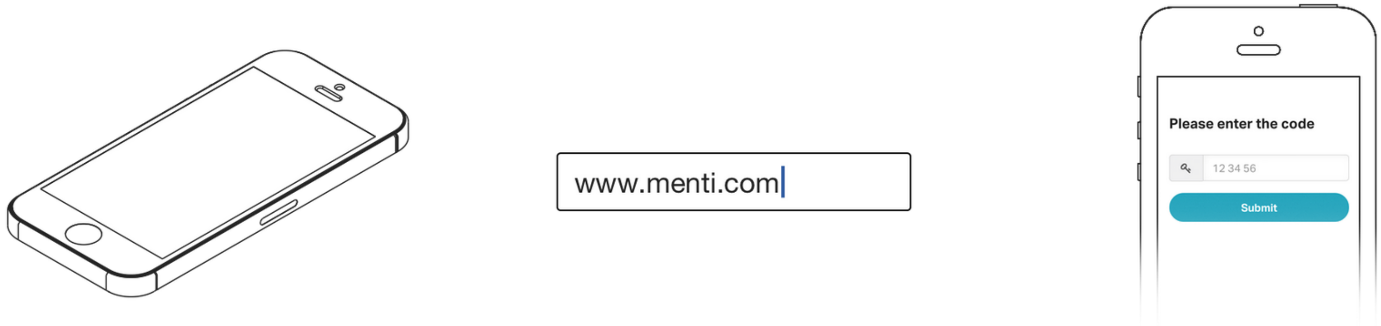
Motivation: Blockchain in Business

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

What do you know about distributed ledgers?

Go to **www.menti.com** and use the code **56 02 12**



1 Grab your phone

2 Go to www.menti.com

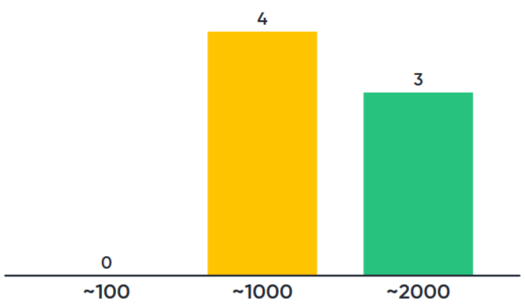
3 Enter the code 80 02 40 and vote!

Mentimeter

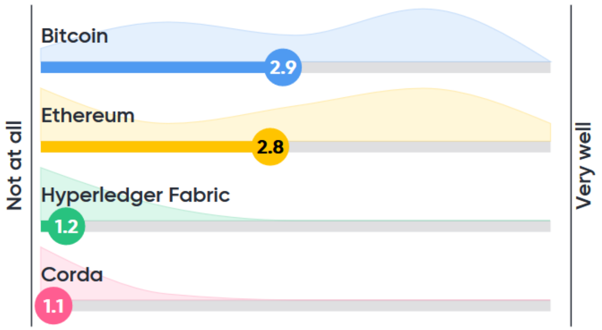
Universität Regensburg
Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Results from last year

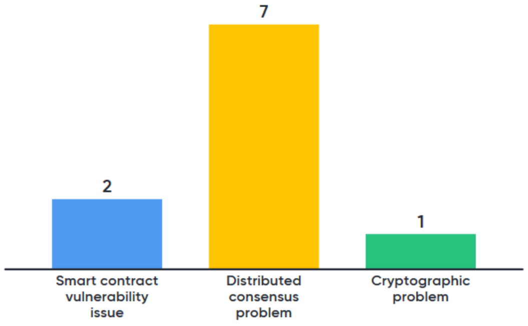
How many cryptocurrencies exist currently?



How well do you know the following DL frameworks?



What is the Byzantine Generals Problem?



Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

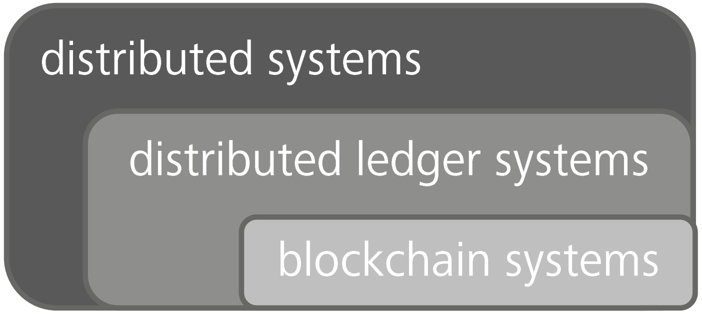
Agenda

- Motivation
- **Distributed Ledger Concepts**
- Distributed Ledger Architecture
- Distributed Ledger Use Cases and Limitations

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

What is a distributed ledger?

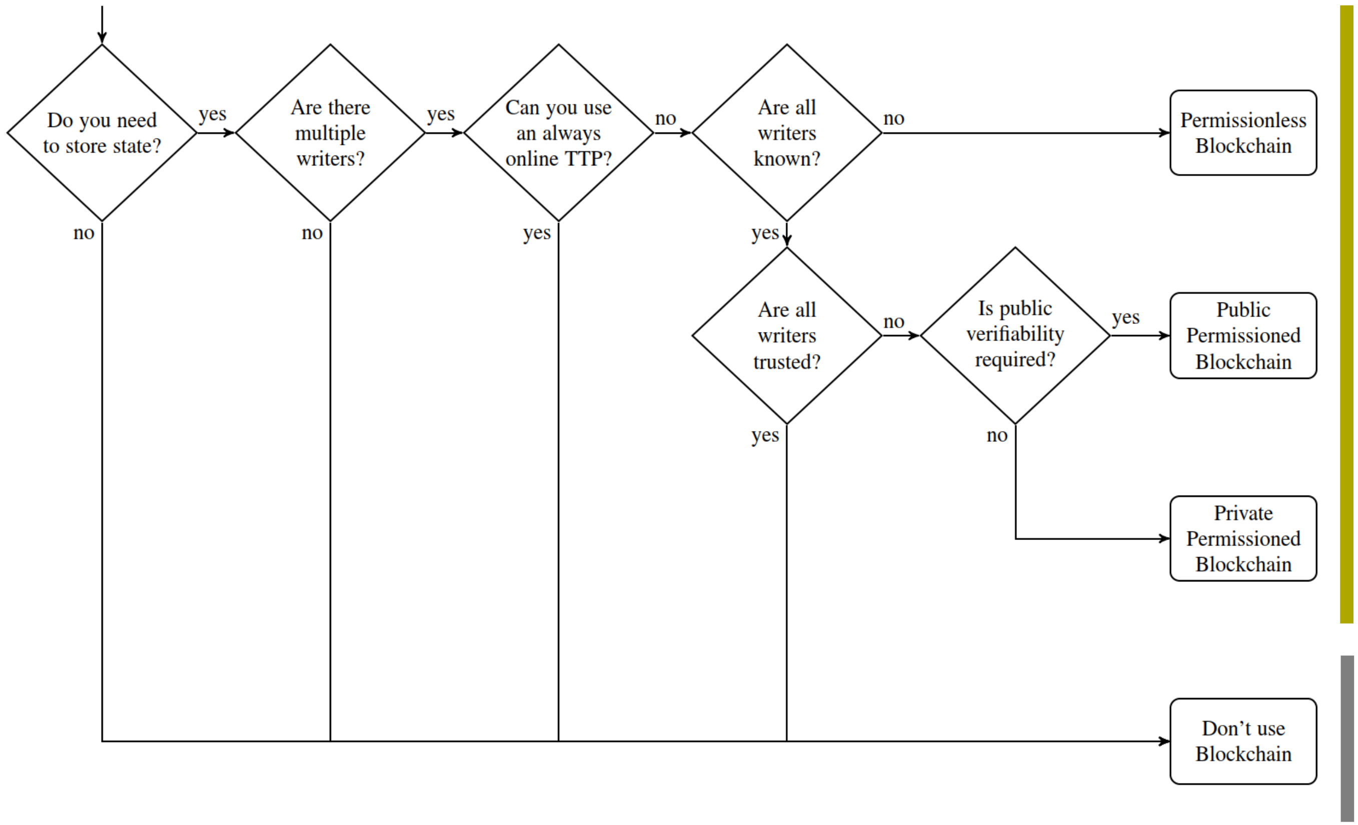
- **(Geo-)replicated, consensually maintained log of transactions**
- Primary purpose: distributed transaction validation and application execution without a central authority
- Blockchain systems are also distributed ledgers
- Properties:
  - transparency and verifiability
  - integrity
  - redundancy



Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Why not a conventional database?

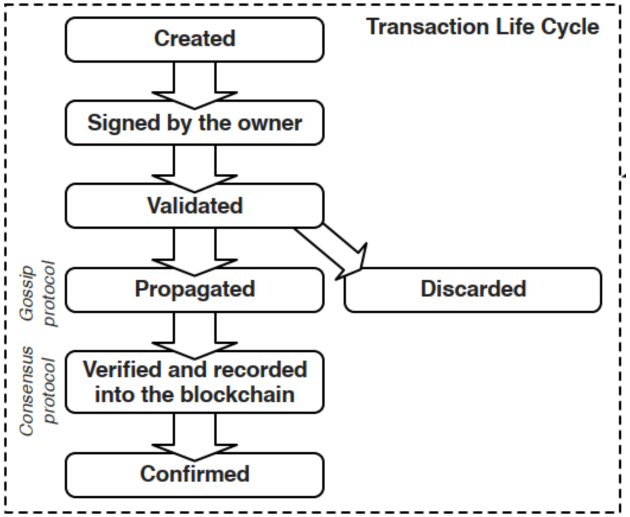


Wüst, Karl, and Arthur Gervais. "Do you need a Blockchain?." IACR Cryptology ePrint Archive 2017 (2017): 375.

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

How does a blockchain system work?

- **Clients** propose transactions (signed with public key)
- Transactions are propagated to all peers
- **Validator nodes** verify and order transactions (no double spending!)
- Transactions are grouped in a **block** by storing them in a Merkle tree
- Validators reach **consensus** on the next block to add to the ledger
- After consensus is reached, the block becomes the new tip of the **blockchain**



Xu, Xiwei, et al. "A pattern collection for blockchain-based applications." *Proceedings of the 23rd European Conference on Pattern Languages of Programs*. ACM, 2018.

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Permissionless ledgers

- Globally distributed network, **anyone can join** and set up a node
- Consensus: based on Zero-Knowledge Proofs
  - usually Proof of Work/Proof of Stake variant
  - < 100 transactions/second (Ethereum: <15, Bitcoin: <7)
  - Confirmation latency: seconds to minutes
- Cryptocurrencies (tokens) used as **incentive system**
- currently ~2000 different permissionless ledgers (coinmarketcap.com)

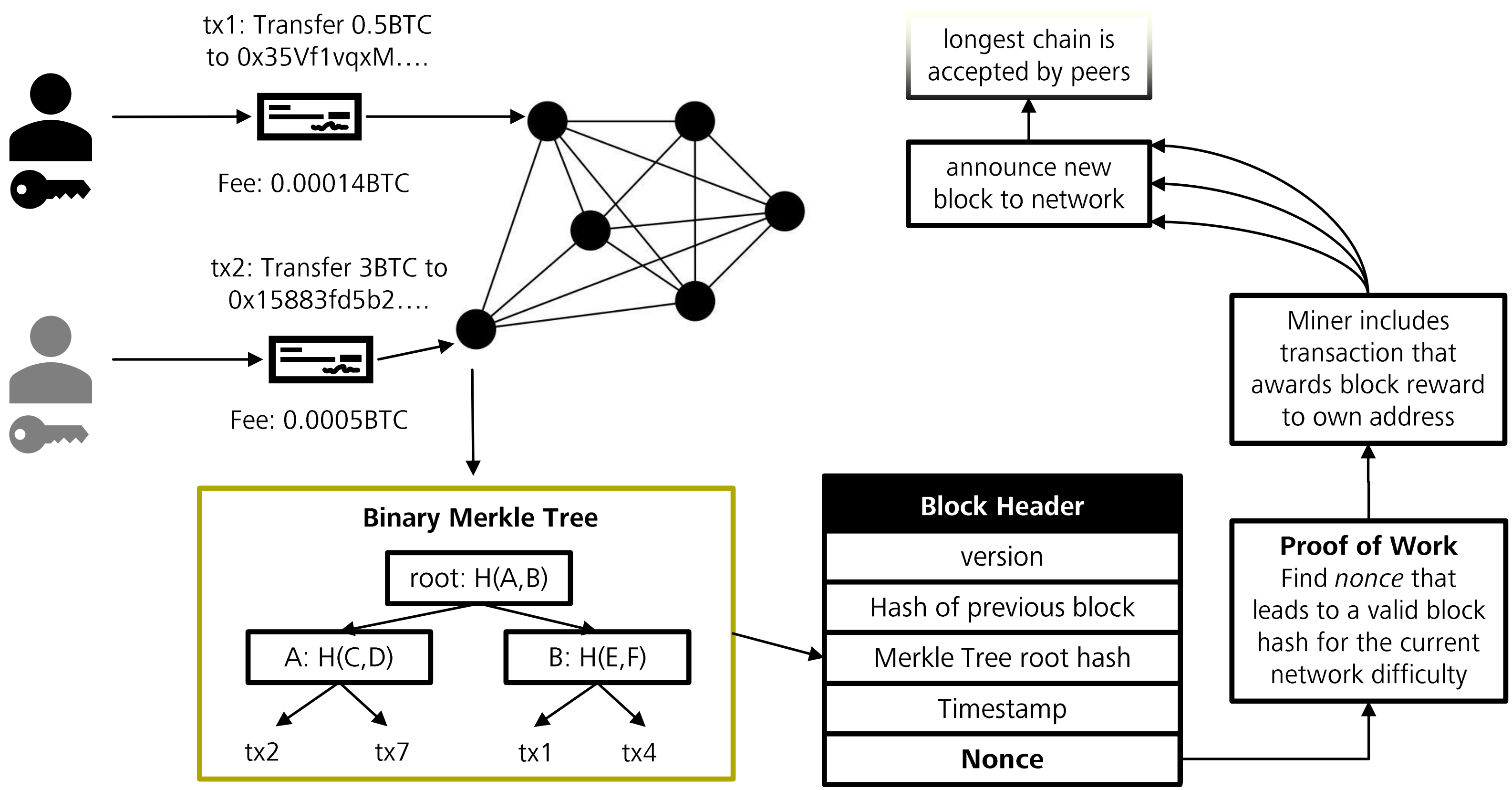
Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Permissionless ledgers

transactions only	decentralized applications
Blockchain 1.0	**Blockchain 2.0**
bitcoin ZCASH nem	ethereum IOTA EOS NEO smart economy TRON
(…)	(…)

Universität Regensburg
Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

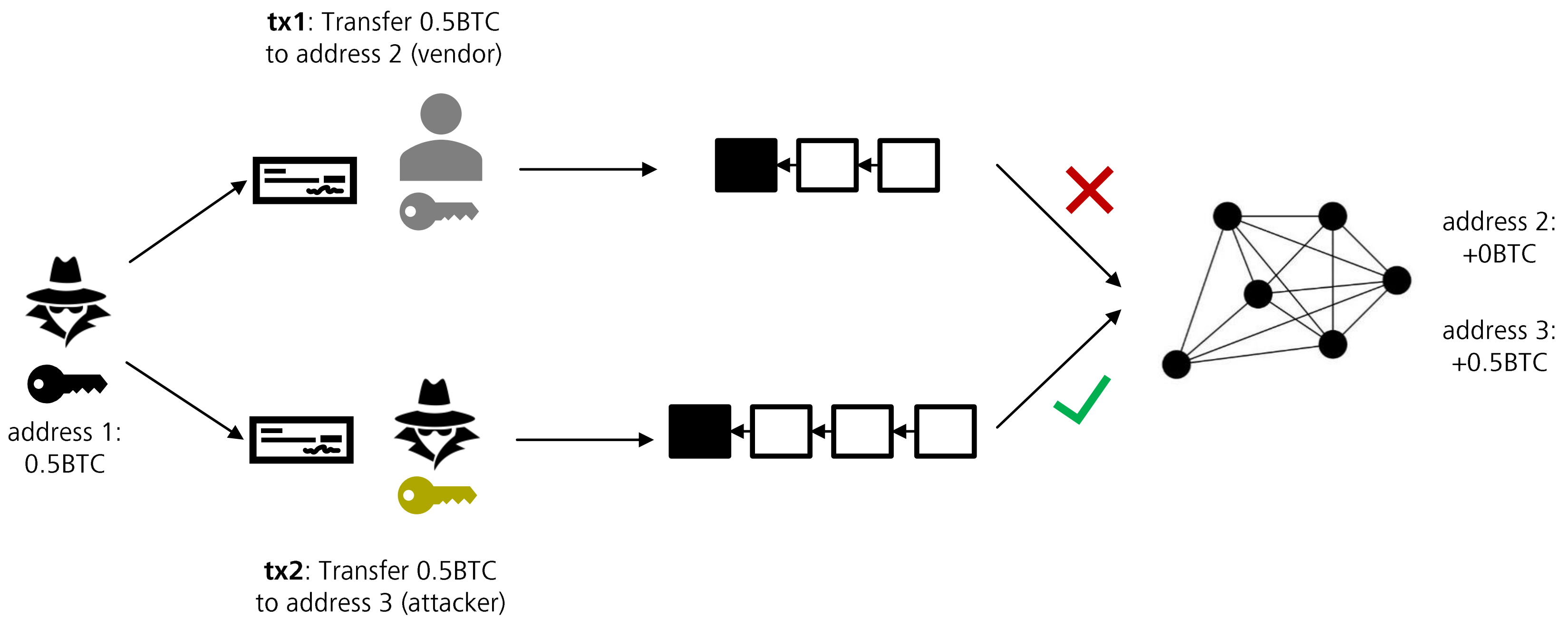
Example: Proof of Work blockchain



Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Example: the double spending problem



Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Permissionless ledgers - performance

- Permissionless ledgers where anybody can mine blocks have performance problems: Global transaction limit < 100 tx/s
- Proposed solutions:
  - Delegation to set of block producers with enough delegated stake (DPoS)
  - off-chain transaction channels (i.e. Bitcoin Lightning)

Cryptocurrency Name	Protocol	TPS
Bitcoin	PoW	7
Ethereum	PoW	15
Ripple	RPCA	1500
Bitcoin Cash	PoW	60
Cardano	PoS	7
Stellar	SCP	1000
NEO	DBFT	10000
Litecoin	PoW	56
EOS	DPoS	~millions
NEM	PoI	4000

Bach, L. M., B. Mihaljevic, and M. Zagar. "Comparative analysis of blockchain consensus algorithms." *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. IEEE, 2018.

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Permissioned ledgers

- Limited number of **authorized participants**
- Consensus: Raft, Byzantine Fault Tolerant (BFT), Round Robin
  - between 100 and 10,000 transactions/second
  - throughput decreases with increasing number of participants
- Developed specifically for **enterprise** usage
- Still undergoing heavy development

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Permissioned ledgers – Open Source

R3 Corda

BIGCHAIN DB

EXONUM

parity

Quorum

MultiChain

HYPERLEDGER

HYPERLEDGER SAWTOOTH

HYPERLEDGER IROHA

HYPERLEDGER FABRIC

HYPERLEDGER BURROW

HYPERLEDGER INDY

HYPERLEDGER BESU

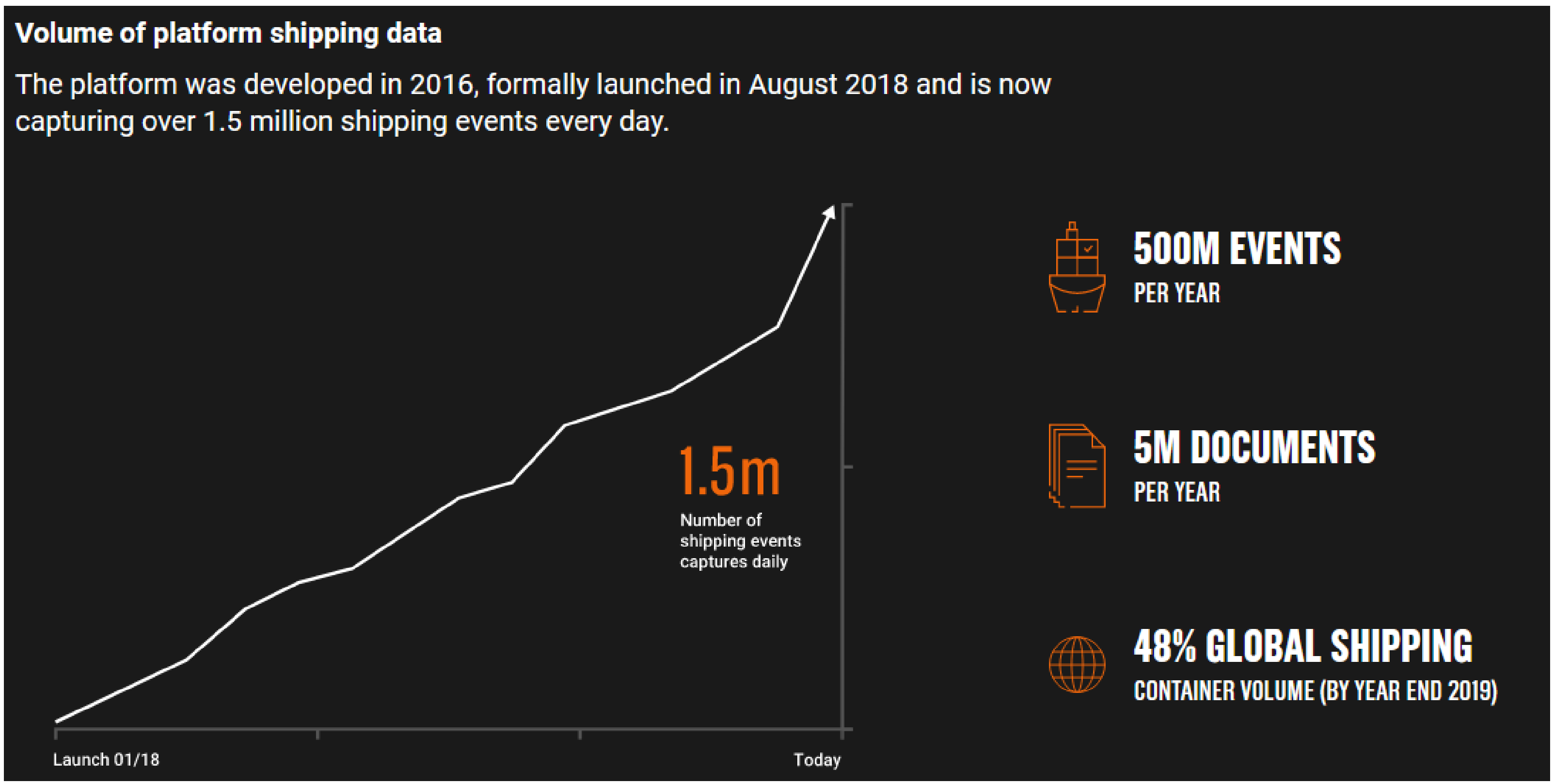
Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Permissioned ledgers – Supply Chain Demo

TRADELENS

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Permissioned ledgers – Supply Chain Demo



https://tour.tradelens.com/status

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

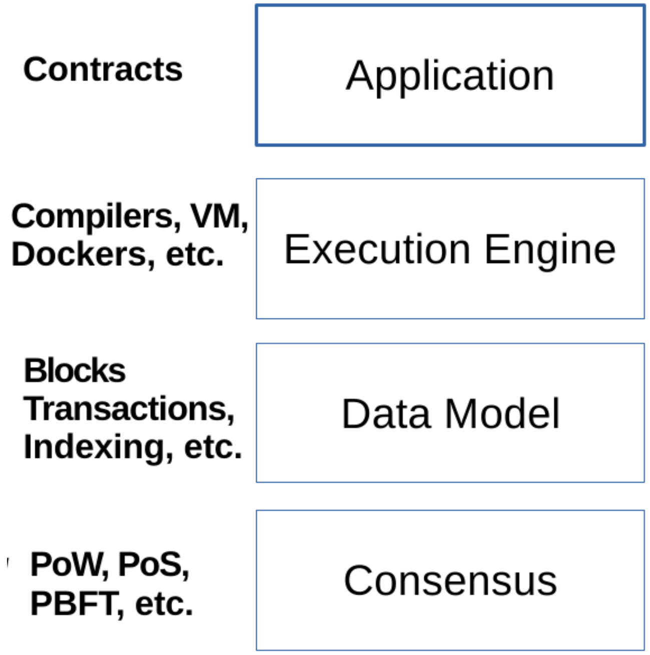
Agenda

- Motivation
- Distributed Ledger Concepts
- **Distributed Ledger Architecture**
- Distributed Ledger Use Cases and Limitations

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Layers: abstract view

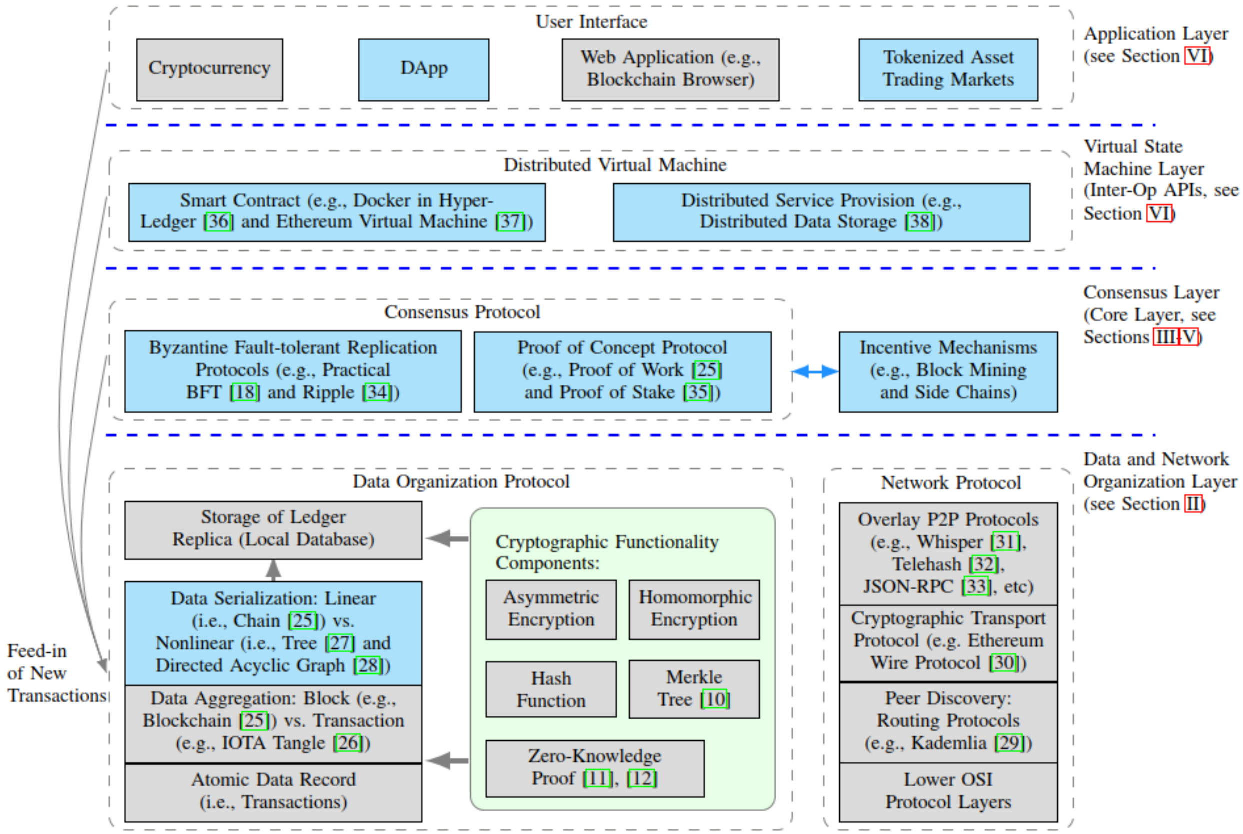


Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., & Tan, K. L. (2017, May). Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Layers: detail



Wang et al. (2018). A Survey on Consensus Mechanisms and Mining Management in Blockchain Networks. *arXiv preprint arXiv:1805.02707*.

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Underlying cryptographic methods

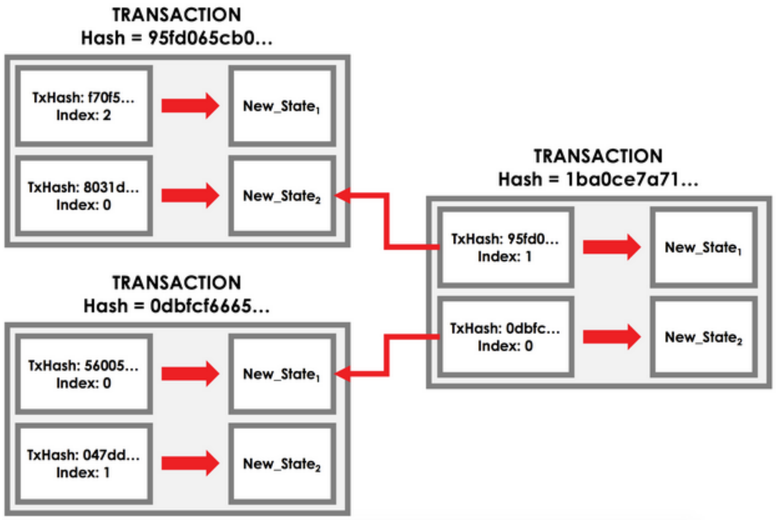
- **Hash functions**
  - integrity verification and block linking
  - Proof of Work consensus (i.e. SHA256, Keccak-256)
- **Public key** cryptography
  - digital signatures (i.e. ECDSA), encrypted communication (Diffie-Hellman)
  - authentication & authorization
- **Symmetric** encryption
  - private blockchain data
- **Zero knowledge proofs**
  - zk-SNARKs (zero-knowledge succinct argument of knowledge)
  - private transactions
- **Homomorphic** encryption
  - private smart contracts

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Transaction model - UTXO

- UTXO: **U**nspent **T**ransaction (**TX**) **O**utputs
- each transaction's inputs must reference a prior transaction's outputs
- UTXO was the first transaction data model (used in Bitcoin)



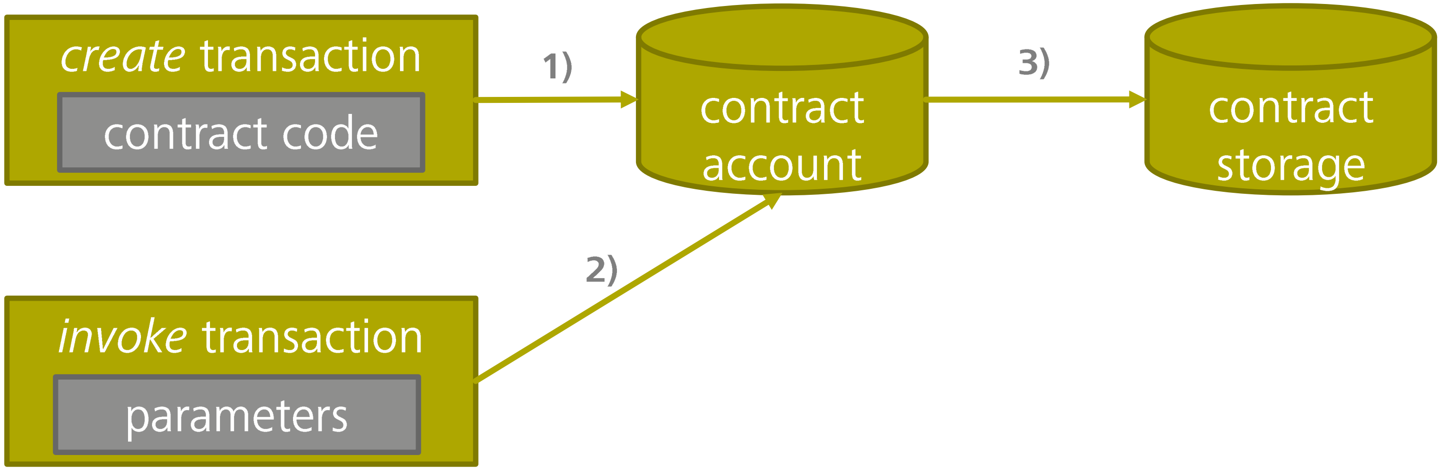
Source: docs.corda.net

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Transaction model – Contracts

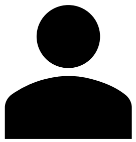
- Transactions interact with smart contracts
- Smart contracts create and modify state (i.e. assets)



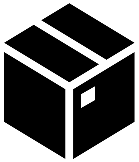
Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

State models

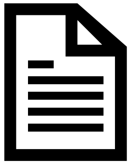
- **Accounts**
  - model user ownership of assets or currency



- **Assets**
  - model real world assets, e.g. shipping goods



- **User-defined state**
  - based on smart contract data types

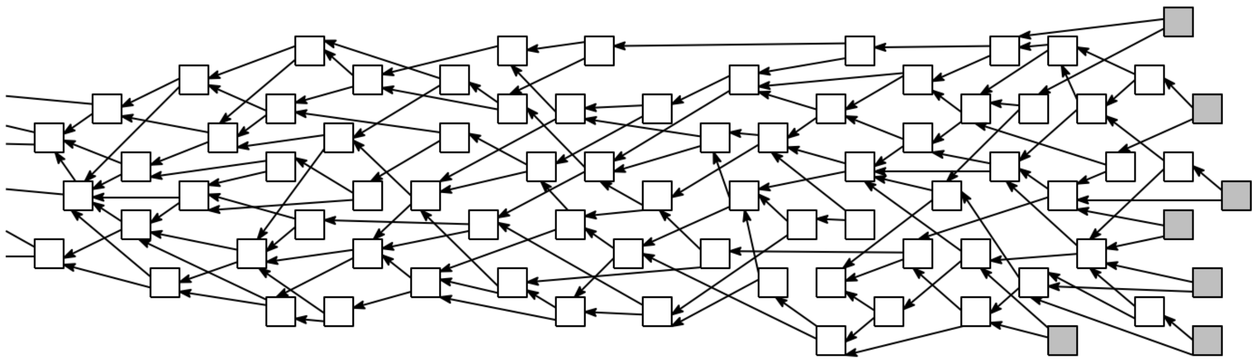


Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

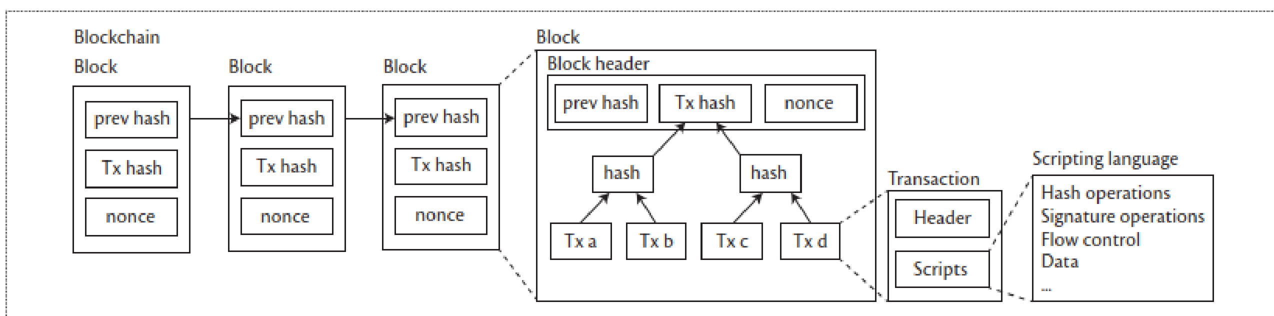
Data structure – Graph vs. Blockchain

Graph (Tangle)



Serguei Popov. “The Tangle”, April 2018, iota.org/research/academic-papers

Blockchain (Bitcoin)



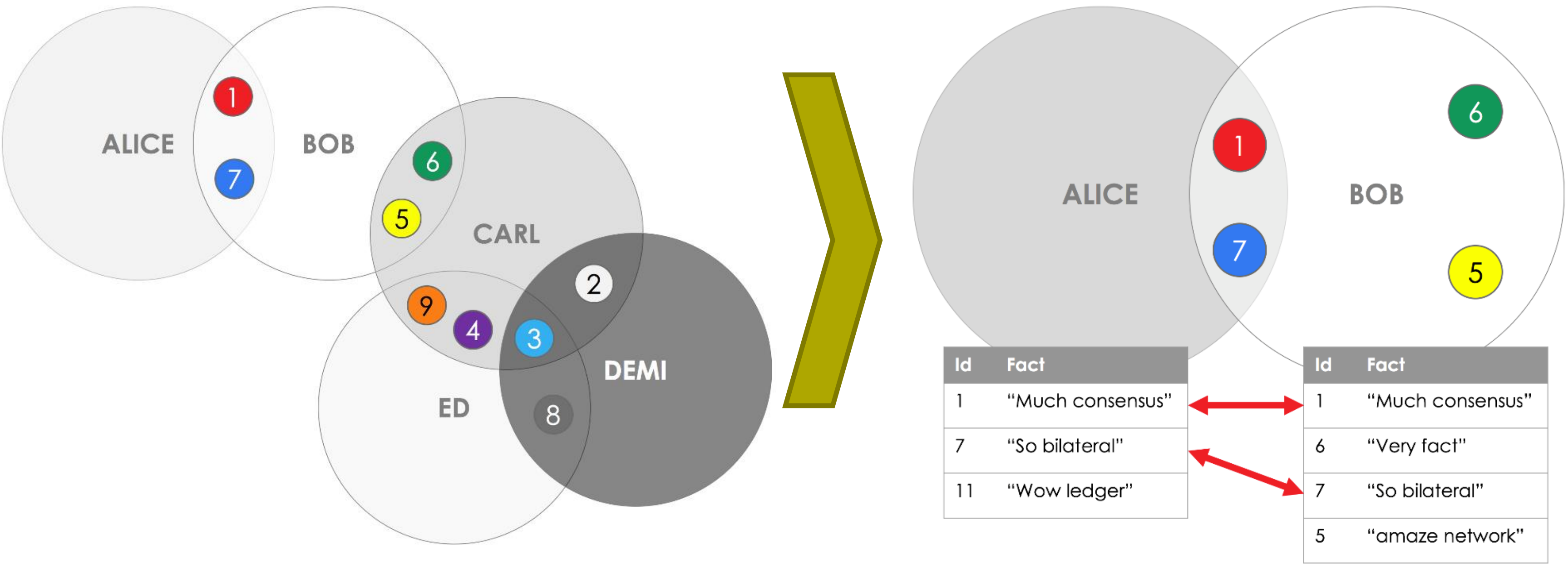
Giechaskiel et al. "When the “Crypto” in Cryptocurrencies Breaks: Bitcoin Security under Broken Primitives", *IEEE Security & Privacy* 16.4 (2018): 46-56

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Example: Corda

- Uniqueness consensus using special **notary** servers



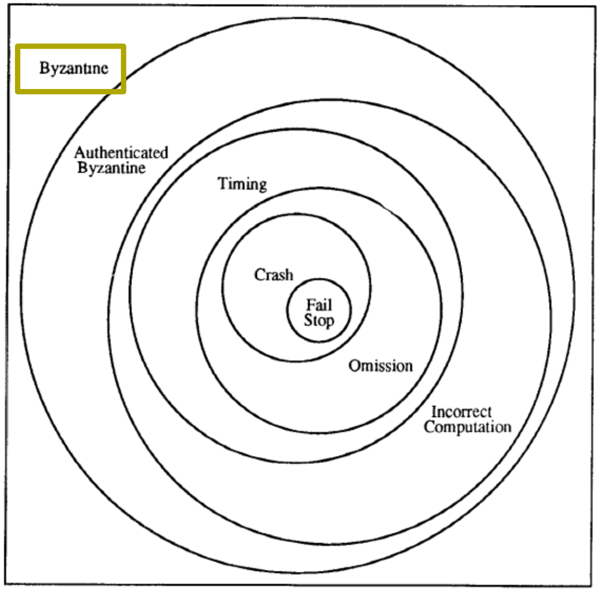
Id	Fact
1	"Much consensus"
7	"So bilateral"
11	"Wow ledger"

Id	Fact
1	"Much consensus"
6	"Very fact"
7	"So bilateral"
5	"amaze network"

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Consensus

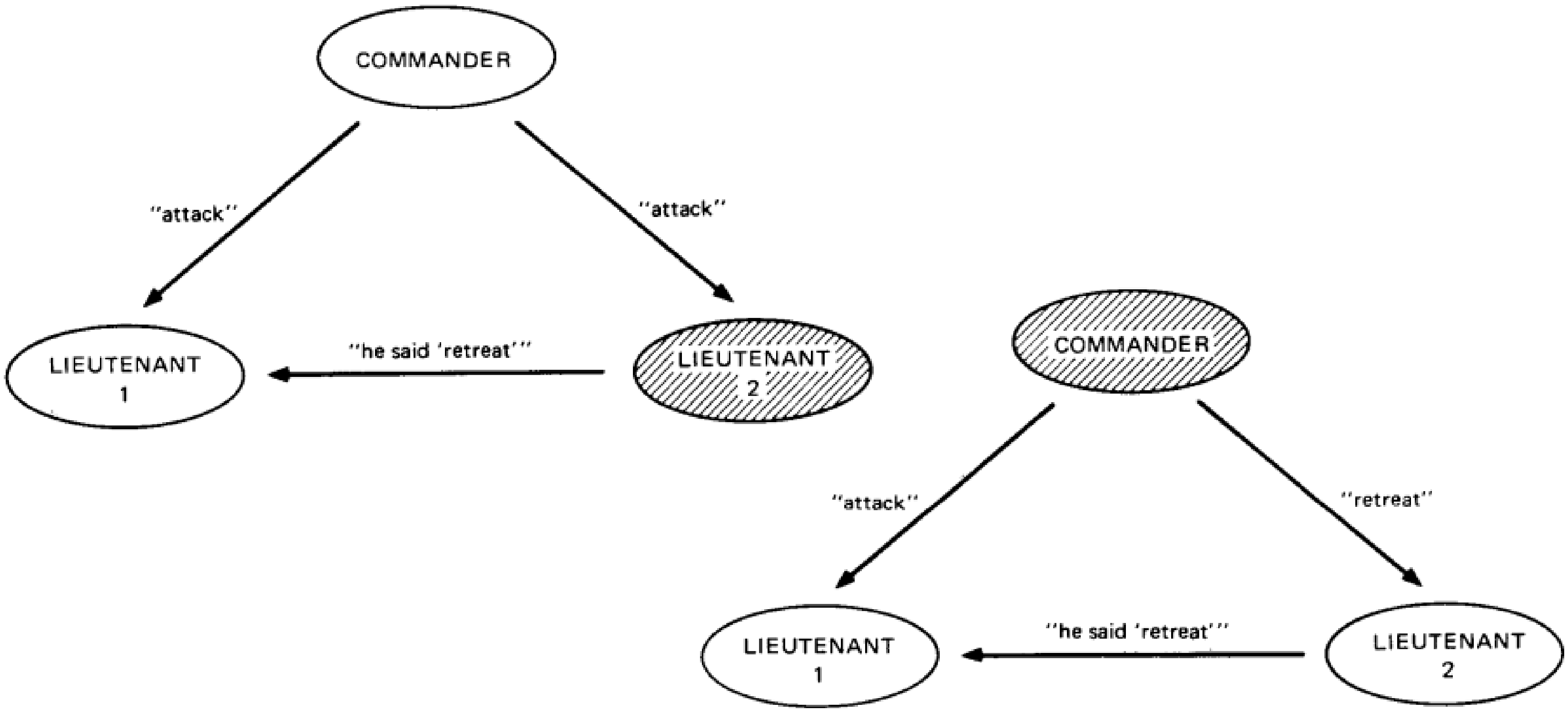
- Distributed ledgers are **replicated state machines**
- To agree on shared state, consensus must be reached regarding state updates
- Consensus protocols aim to be (byzantine) fault-tolerant
- Byzantine Fault: presents different symptoms to different observers



Ordered fault classification by Barborak et al. (1993)

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

The Byzantine Generals Problem



Lamport, L., Shostak, R., & Pease, M. (1982). The Byzantine generals problem. ACM Transactions on Programming Languages and Systems (TOPLAS)

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Byzantine Generals Problem

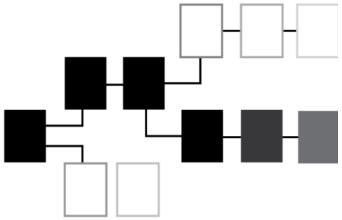
- Consensus should be reached even if nodes are faulty
- **asynchronous** networks: deterministic consensus is impossible (Fischer-Lynch-Paterson impossibility)
  → rely on stochastic algorithms or weak synchrony assumptions
- **synchronous** networks:
  more than 2/3 of all nodes must be honest to reach consensus

- Protocol requirements:
  - Liveness / Termination
  - Agreement
  - Validity
  - Total Order

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Proof of Work (PoW) / Proof of Stake (PoS)



- Used in permissionless environments
- Stochastic algorithms: **Forks** are possible

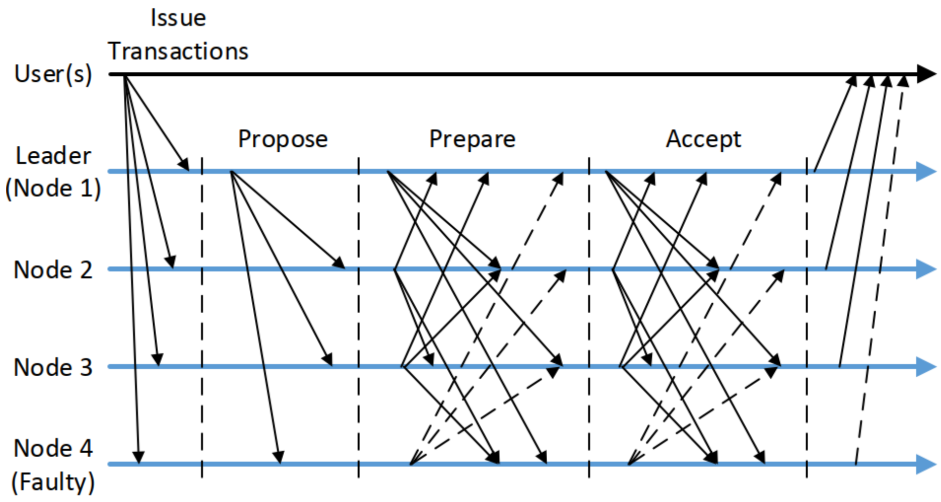
- PoW was the first algorithm proposed in 2008
  - perform computation-intensive, but easily verifiable operation to become block leader and earn reward
  - longest/most computation-intensive fork is accepted

- PoS addresses PoW inefficiency (power consumption)
  - block leader is determined randomly **based on staked (frozen) currency**
  - in **delegated PoS**, nodes can vote for their favorite block producer

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Practical Byzantine Fault Tolerance (PBFT)

- PBFT (2002) was the first high-performance and attack-resistant algorithm to solve the Byzantine Generals Problem (20 years later)
- However: Quadratic communication complexity $\Theta(n^2)$
- Many variants and improvements developed:
  most recent: SBFT – 2018, $\Theta(n)$ in the common case



Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Other consensus algorithms

- Permissioned
  - BFT SMaRt, Simple BFT, …
  - Tendermint: BFT consensus middleware
  - Proof of Authority: focus on availability over consistency
- Permissionless
  - Proof of Importance: Reputation/Stake based
  - Proof of Elapsed Time: Based on hardware enclaves (Intel SGX)
  - Proof of Burn
  - IOTA Tangle, Swirlds Hashgraph

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Smart contracts

- Executed within a sandboxed **virtual machine**
- **Replicated execution** across all validators
  - resulting state change must be deterministic to achieve consensus
- Purpose: trusted and autonomous distributed contract execution

- **DApps** (Decentralized Applications) use smart contracts as backend instead of a traditional server

- First implementation: Ethereum VM & Solidity
- Permissionless networks: **contract invocation costs** based on operation type to avoid infinite execution Denial of Service (DoS)
- Newer frameworks support multiple languages, Web Assembly

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Smart contracts: ERC20 token standard example

EthFiddle Security Audit Share Login

```
// ----------------------------------------------------------------------------

// ERC Token Standard #20 Interface

// https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20-token-standard.md

// ----------------------------------------------------------------------------

contract ERC20Interface {

    function totalSupply() public constant returns (uint);

    function balanceOf(address tokenOwner) public constant returns (uint balance);

    function allowance(address tokenOwner, address spender) public constant returns (uint remaining);

    function transfer(address to, uint tokens) public returns (bool success);

    function approve(address spender, uint tokens) public returns (bool success);

    function transferFrom(address from, address to, uint tokens) public returns (bool success);


    event Transfer(address indexed from, address indexed to, uint tokens);

    event Approval(address indexed tokenOwner, address indexed spender, uint tokens);

}
```

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Smart contracts: Operation pricing

Ethereum yellow paper gas fee structure:

- **Gas** is consumed for every **basic operation**
- Gas consumption translates to **transaction costs in Ether**
- **Dynamic gas limits** on blocks (set by peers) and transactions (set by users)

Appendix G. Fee Schedule

The fee schedule G is a tuple of 31 scalar values corresponding to the relative costs, in gas, of a number of abstract operations that a transaction may effect.

Name	Value	Description*
G_{zero}	0	Nothing paid for operations of the set W_{zero}.
G_{base}	2	Amount of gas to pay for operations of the set W_{base}.
$G_{verylow}$	3	Amount of gas to pay for operations of the set $W_{verylow}$.
G_{low}	5	Amount of gas to pay for operations of the set W_{low}.
G_{mid}	8	Amount of gas to pay for operations of the set W_{mid}.
G_{high}	10	Amount of gas to pay for operations of the set W_{high}.
$G_{extcode}$	700	Amount of gas to pay for operations of the set $W_{extcode}$.
$G_{balance}$	400	Amount of gas to pay for a BALANCE operation.
G_{sload}	200	Paid for a SLOAD operation.
$G_{jumpdest}$	1	Paid for a JUMPDEST operation.
G_{sset}	20000	Paid for an SSTORE operation when the storage value is set to non-zero from zero.
G_{sreset}	5000	Paid for an SSTORE operation when the storage value's zeroness remains unchanged or is set to :
R_{sclear}	15000	Refund given (added into refund counter) when the storage value is set to zero from non-zero.
$R_{selfdestruct}$	24000	Refund given (added into refund counter) for self-destructing an account.
$G_{selfdestruct}$	5000	Amount of gas to pay for a SELFDESTRUCT operation.
G_{create}	32000	Paid for a CREATE operation.
$G_{codedeposit}$	200	Paid per byte for a CREATE operation to succeed in placing code into state.
G_{call}	700	Paid for a CALL operation.
$G_{callvalue}$	9000	Paid for a non-zero value transfer as part of the CALL operation.
$G_{callstipend}$	2300	A stipend for the called contract subtracted from $G_{callvalue}$ for a non-zero value transfer.
$G_{newaccount}$	25000	Paid for a CALL or SELFDESTRUCT operation which creates an account.
G_{exp}	10	Partial payment for an EXP operation.
$G_{expbyte}$	10	Partial payment when multiplied by $\lceil \log_{256}(exponent) \rceil$ for the EXP operation.
G_{memory}	3	Paid for every additional word when expanding memory.
$G_{txcreate}$	32000	Paid by all contract-creating transactions after the *Homestead transition*.
$G_{txdatazero}$	4	Paid for every zero byte of data or code for a transaction.
$G_{txdatanonzero}$	68	Paid for every non-zero byte of data or code for a transaction.
$G_{transaction}$	21000	Paid for every transaction.
G_{log}	375	Partial payment for a LOG operation.
$G_{logdata}$	8	Paid for each byte in a LOG operation's data.
$G_{logtopic}$	375	Paid for each topic of a LOG operation.
G_{sha3}	30	Paid for each SHA3 operation.
$G_{sha3word}$	6	Paid for each word (rounded up) for input data to a SHA3 operation.
G_{copy}	3	Partial payment for *COPY operations, multiplied by words copied, rounded up.
$G_{blockhash}$	20	Payment for BLOCKHASH operation.

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Blockchain security: Smart contracts

- Example Vulnerability: Solidity reentrancy attack

```
contract Bob {
  bool sent = false;
  function ping(address c) {
    if (!sent) {
       c.call.value(2)();
       sent = true;
}}}
```

```
contract Bob { function ping(); }

contract Mallory {
  function() {
    Bob(msg.sender).ping(this);
  }
}
```

  - Mallory contract invokes Bob’s ping function, which sends 2 wei to own address
  - Fallback is triggered when a contract receives currency without data
  - Fallback executes the function again before it gets a chance to set sent=true
  - Infinite loop continues until out-of-gas or Bob is depleted of funds

- Mitigation: Vulnerability Scanners

Mythril

SECURIFY SECURITY SCANNER FOR ETHEREUM SMART CONTRACTS

SmartCheck

Atzei, Nicola, Massimo Bartoletti, and Tiziana Cimoli. "A survey of attacks on Ethereum smart contracts." IACR Cryptology ePrint Archive 2016 (2016): 1007.

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Blockchain security: General aspects

- Three **categories** of blockchain security:
  - **operational** security (key management, trust issues)
  - **smart contract** security (vulnerabilities, compiler bugs)
  - **consensus protocol** security (double spending, eclipse attack)

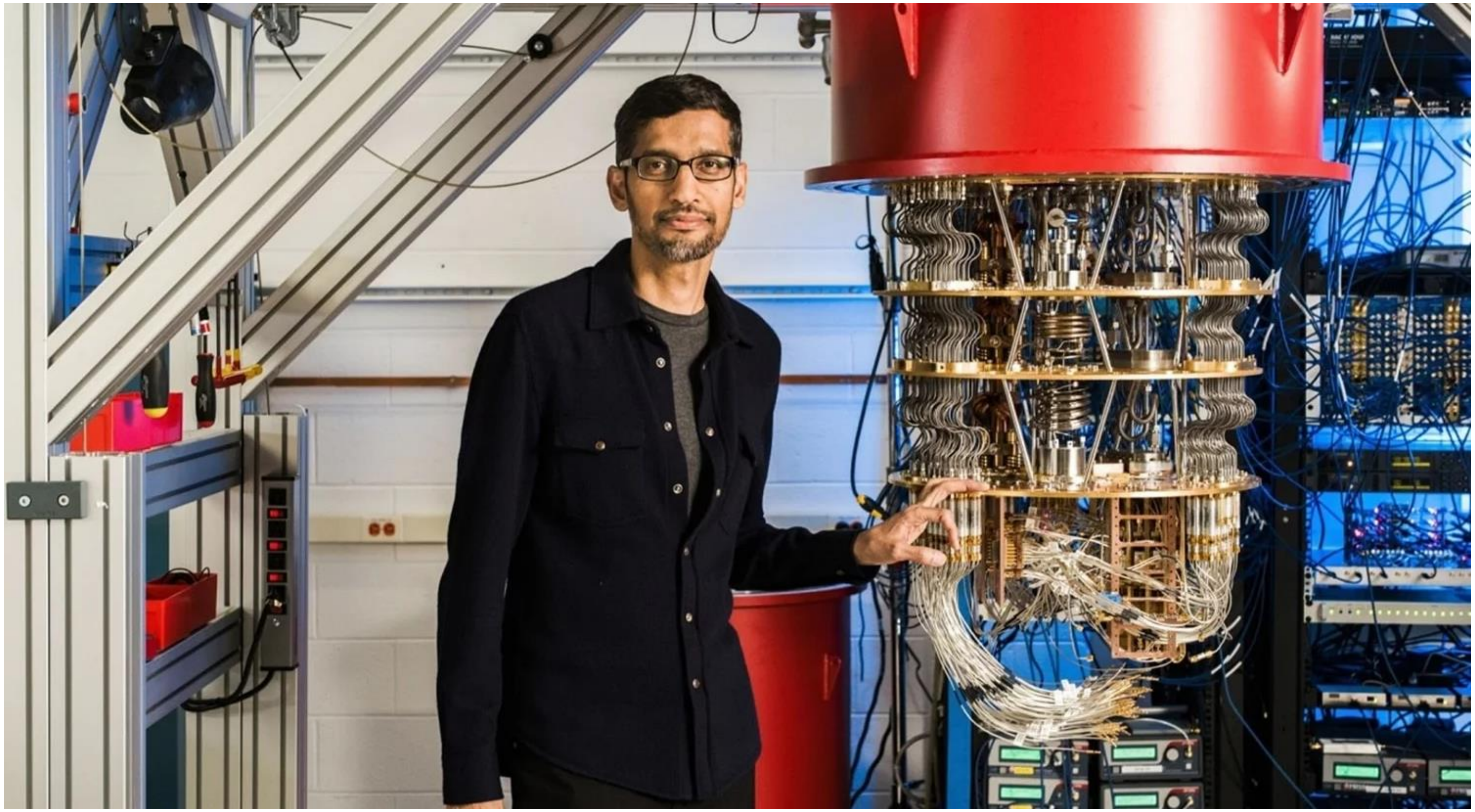
- **> 2/3 attacks** related to **operational** security
  - example: exchange hacks - Mt. Gox (2014)
  - future concern: quantum-resistant blockchain cryptography

- **Mitigations**:
  - key encryption
  - cold wallets
  - post-quantum cryptography
  - smart contract termination (suicide)
  - smart contract vulnerability scanners
  - new consensus protocols

Chia, Vincent, et al. "Rethinking Blockchain Security: Position Paper." *arXiv preprint, arXiv:1806.04358* (2018).

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Blockchain security: Quantum attacks



Source: https://www.theguardian.com/technology/2019/oct/23/google-claims-it-has-achieved-quantum-supremacy-but-ibm-disagrees

Blockchain security: Quantum attacks

- Public key cryptography is vulnerable to **Shor's Algorithm**
  - **exponential** speedup for finding the discrete logarithm
- Hashes & symmetric encryption are vulnerable to **Grover's algorithm**
  - **quadratic** speedup for brute-force attacks
- Mitigation requires a **redesign of blockchain primitives**
  - hash-based signature schemes
  - hash combiners
  - hash function replacement

El Bansarkhani, Rachid, Matthias Geihs, and Johannes Buchmann. "PQChain: Strategic Design Decisions for Distributed Ledger Technologies against Future Threats." IEEE Security & Privacy 16.4 (2018): 57-65.

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Privacy concerns

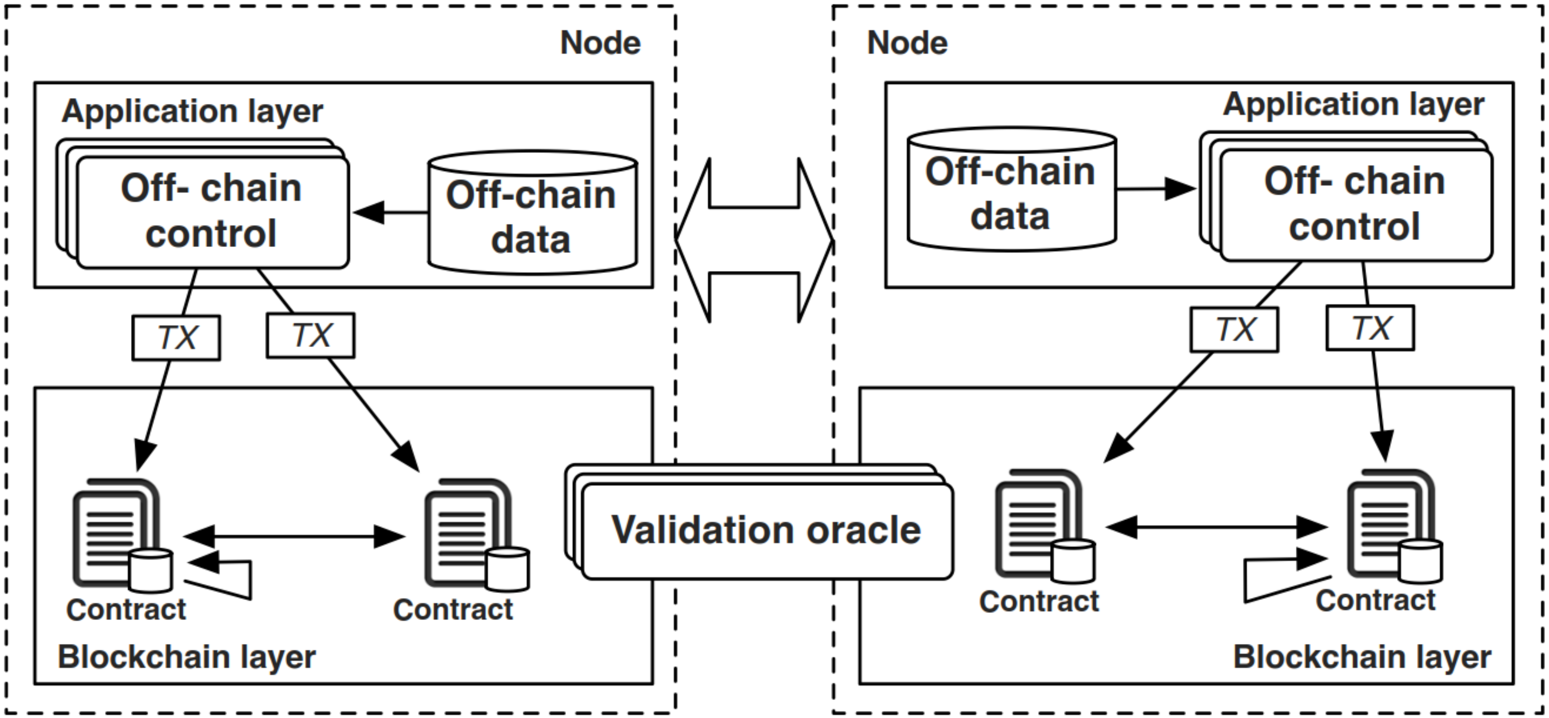
- By default, **all data is public** to all nodes in the distributed system
- Account addresses are pseudonymous, but **re-identification attacks** can reveal identities through data mining
- Personal data on the blockchain vs. GDPR compliance

- Privacy enablers:
  - **Private encrypted transactions** between participants (e.g. Parity)
  - **Zero knowledge cryptography** (e.g. ZCash)
    zk-SNARKs prove existence of data without revealing it
  - **Secure multi-party computation**
    perform distributed computations without revealing data

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

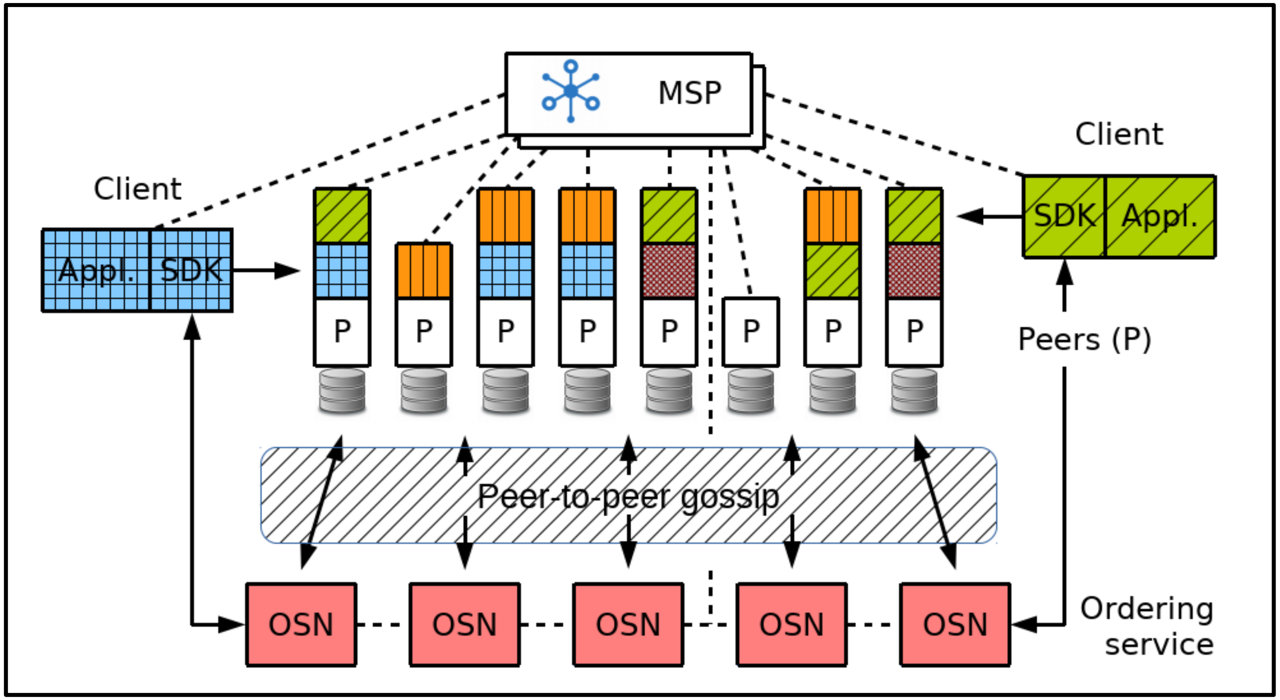
Architecture of a blockchain-based application



Xiwei Xu et al. “The blockchain as a software connector”. In: Proceedings - 2016 13th Working IEEE/IFIP Conference on Software Architecture, WICSA 2016. IEEE, Apr. 2016, pp. 182–191. URL: https://ieeexplore.ieee.org/document/7516828

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Example: Hyperledger Fabric

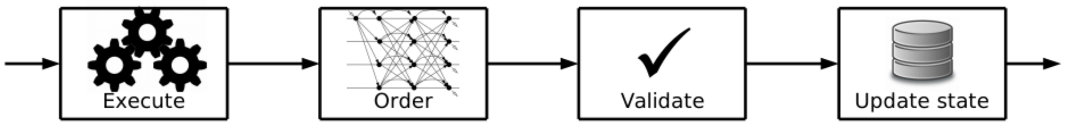


Fabric network with **federated MSPs** and running **multiple chaincodes** (differently shaded and colored), selectively installed on peers according to policy

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Example: Hyperledger Fabric

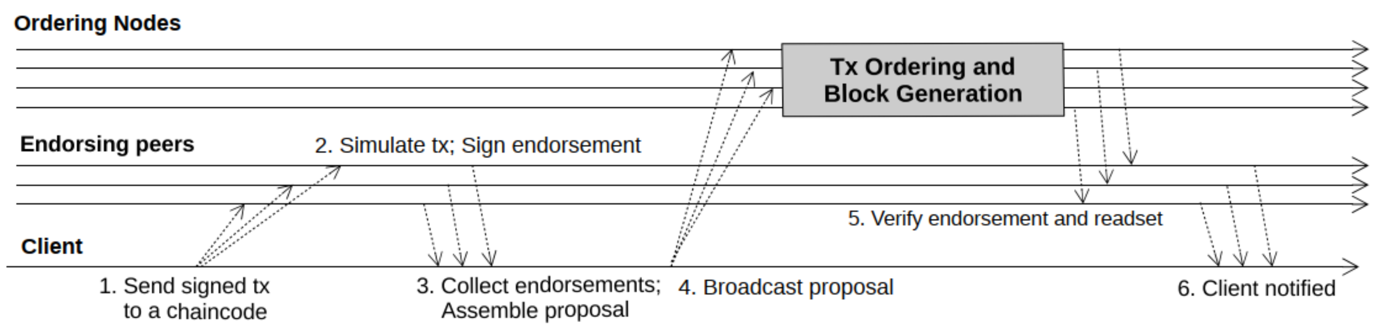


- Simulate trans. and endorse
- Create rw-set
- Collect endorse-ments

- Order rw-sets
- Atomic broadcast (consensus)
- Stateless ordering service

- Validate endorse-ments & rw-sets
- Eliminate invalid and conflicting trans.

- Persist state on all peers



Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Off-chain storage

- Commonly, hashes are used as references for mapping off-chain data
- **Any database** can be used (e.g. relational, No-SQL, DHT)

- DHT: Distributed Hash Table
  - **key-value-store**, often using hash of value as key
  - fully decentralized, keys are retrieved with a **routing algorithm**
  - popular DHTs (Swarm/IPFS) rely on **S/Kademlia** (XOR-metric)
  - Kademlia provides defense against common adversarial attacks (eclipse / sybil / churn / adversarial routing)

Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Agenda

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

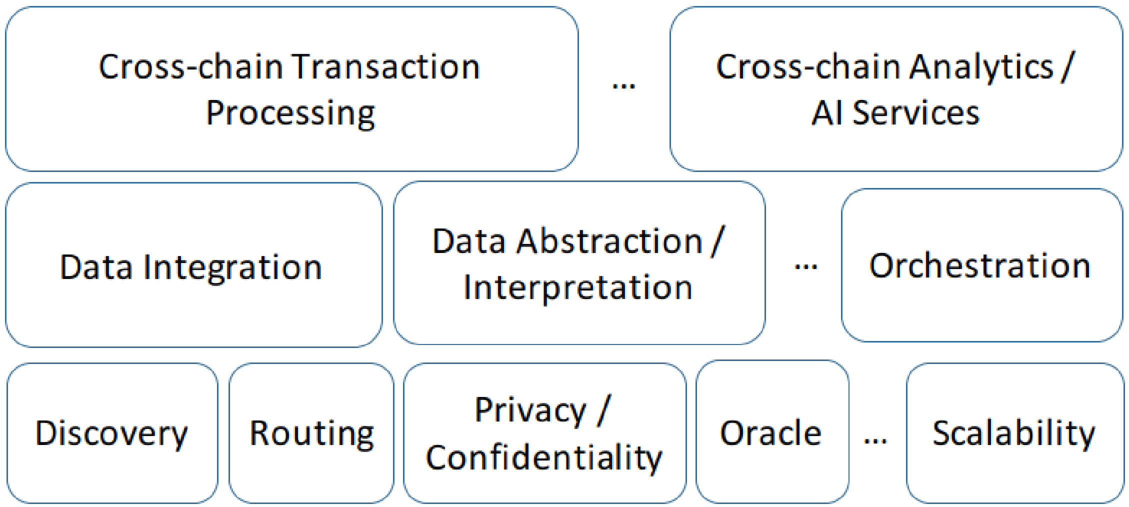
Top ten obstacles to adoption

1. Scalability – full agreement
2. Privacy
3. Cost-effectiveness
4. Scalability – storage replication
5. Interoperability
6. Agility
7. Key Management
8. Meaningful comparisons
9. Governance
10. Usability

Meiklejohn, Sarah. "Top Ten Obstacles along Distributed Ledgers Path to Adoption." *IEEE Security & Privacy* 16.4 (2018): 13-19, https://smeiklej.com/files/topten.pdf.

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Interledger technologies for the internet of blockchains

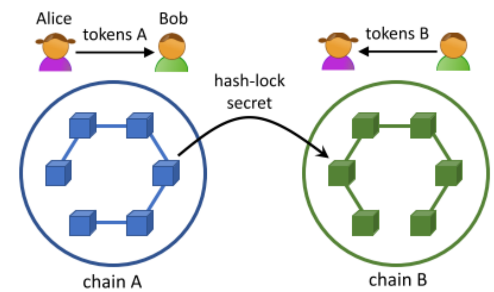


- Based on current design differences of ledgers, there will be many independent deployments in the future
- **Standards** and **platform designs** must be developed

Hoang Tam Vo et al. "Internet of Blockchains: Techniques and Challenges Ahead", *IEEE Blockchain* 2018, URL: http://cse.stfx.ca/~cybermatics/2018/Proceedings/index.html#!/toc/0

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

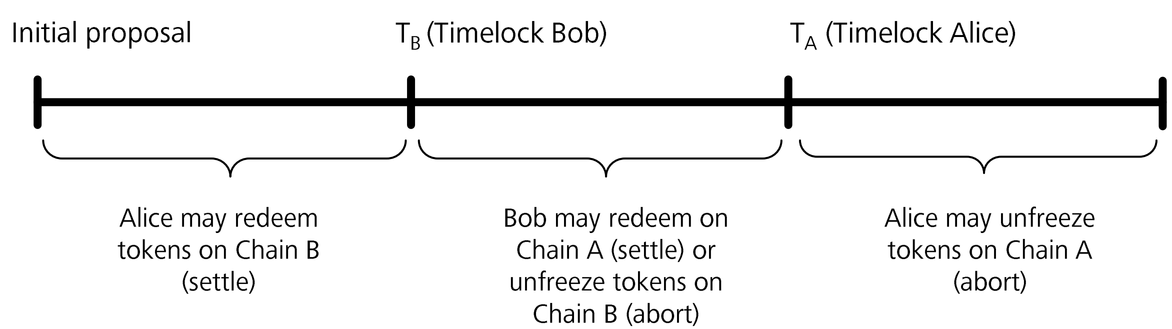
Interledger: Cross-blockchain value swaps



Hash Time-Lock Contracts (HTLC)

(ex: token swap via smart contracts on Chain A and Chain B)

- **Alice** reserves tokens for **Bob** on **Chain A**, dependent on some secret *s*. The secret is set by including a hash-lock *H(s)*
- **Bob** reserves tokens for **Alice** on **Chain B,** also setting *H(s)*
- **Alice** redeems tokens on **Chain B** by sending *s* to the contract
- **Bob** redeems tokens on **Chain A** by sending *s* from his address
- Timelocks avoid indefinite token lockup:



Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Proposed distributed ledger use cases

- Decentralized currency
- Financial services (interbank settlement, insurance policies)
- Data provenance
- Data marketplaces
- Identity Management
- Health records
- Supply chain coordination and tracking
- Energy trading
- Security

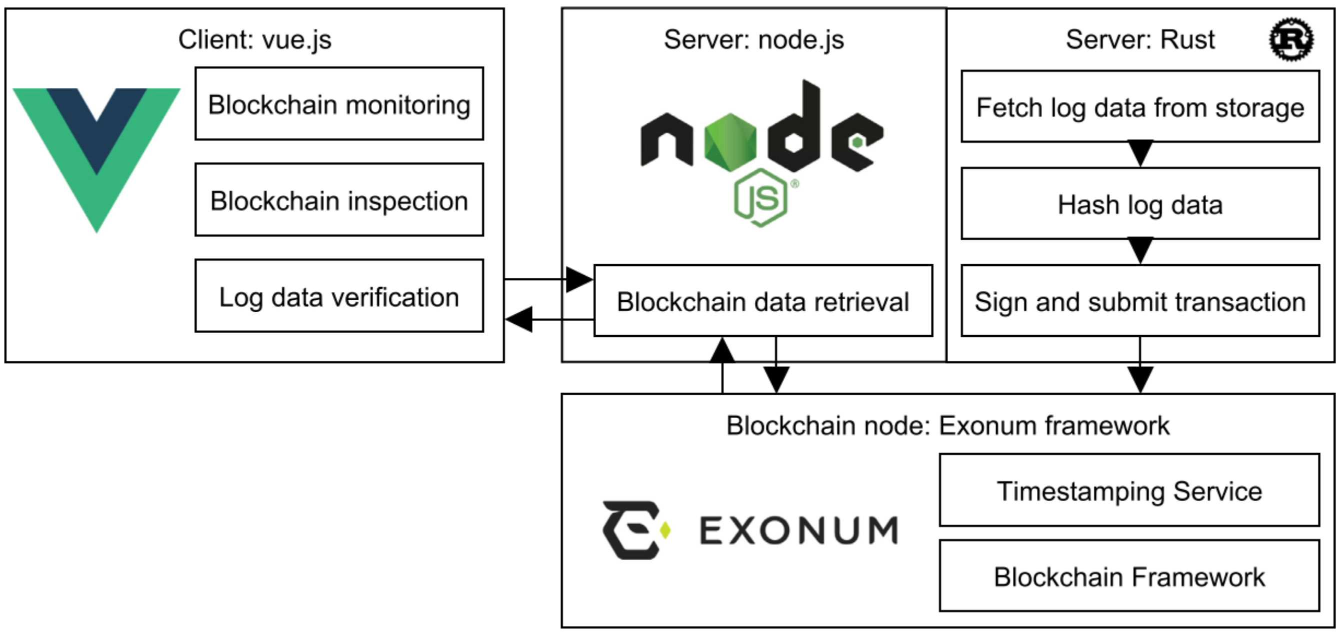
Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Distributed ledgers in security research

- **PKI** based on a distributed ledger
  - replace trust in centralized certificate authority
- Dynamic **access control** for off-chain data
  - transactions required to grant and revoke access
- Blockchain-based **data provenance**
- Data **integrity assurance**
- Malware and threat intelligence exchange platforms

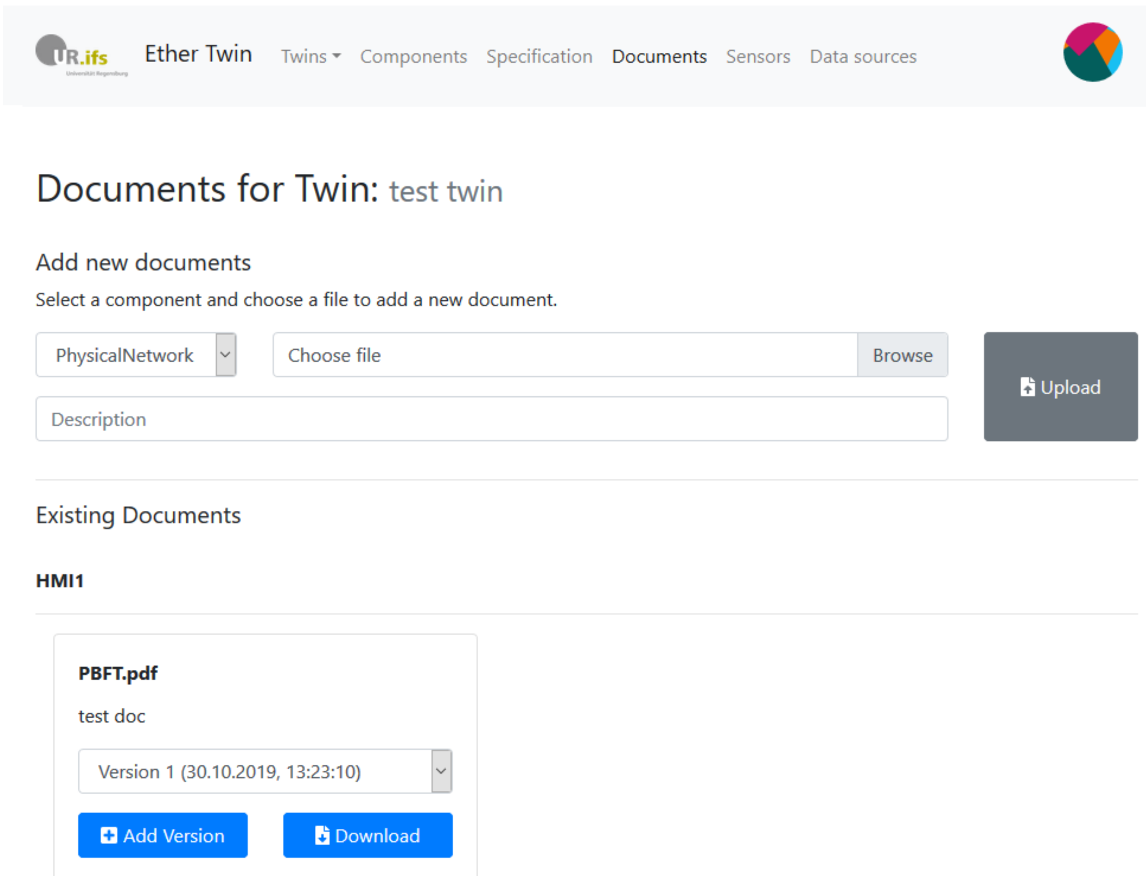
Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Log non-repudiation using a distributed ledger



Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

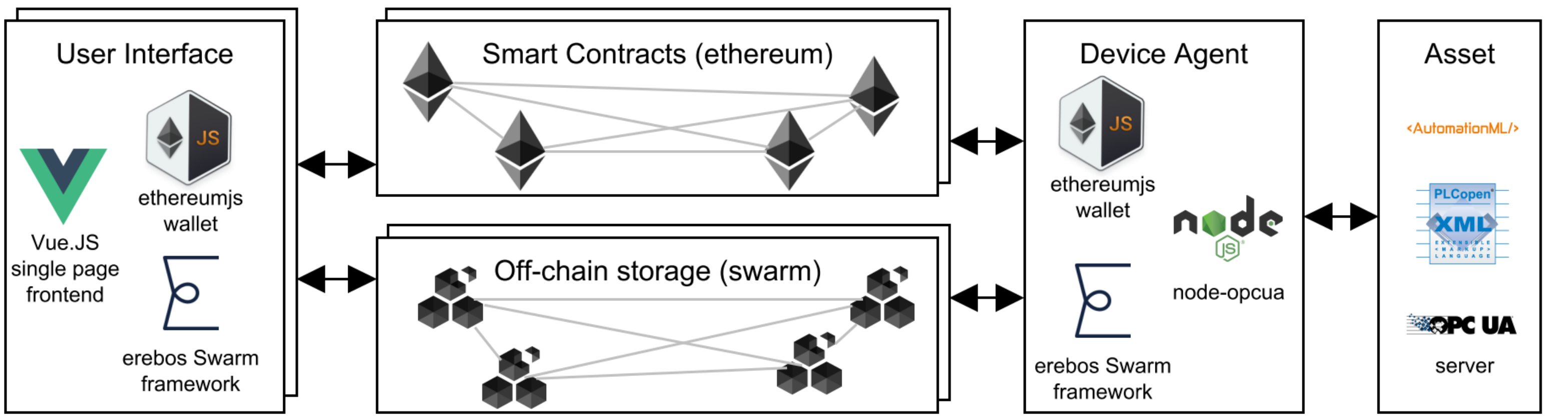
Digital Twin Management on the Blockchain



Universität Regensburg

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Digital Twin Management on the Blockchain



Ethereum: Permissioned blockchain with Solidity Smart Contracts
https://ethereum.org/
https://solidity.readthedocs.io/en/

swarm

Swarm: DHT-based off-chain storage network
https://swarm-guide.readthedocs.io/

Benedikt Putz
Chair of Information Systems
FACULTY OF BUSINESS, ECONOMICS AND MANAGEMENT
INFORMATION SYSTEMS

Assignment

- **Name and briefly explain three key differences between permissionless and permissioned distributed ledgers.**
- **Explain the applicability of DLT in a cybersecurity use case of your choice. Use the framework by Wüst and Gervais (2017) as guidance.**
- **Note:** Please back up your statements with appropriate references.