Technische Universität Berlin

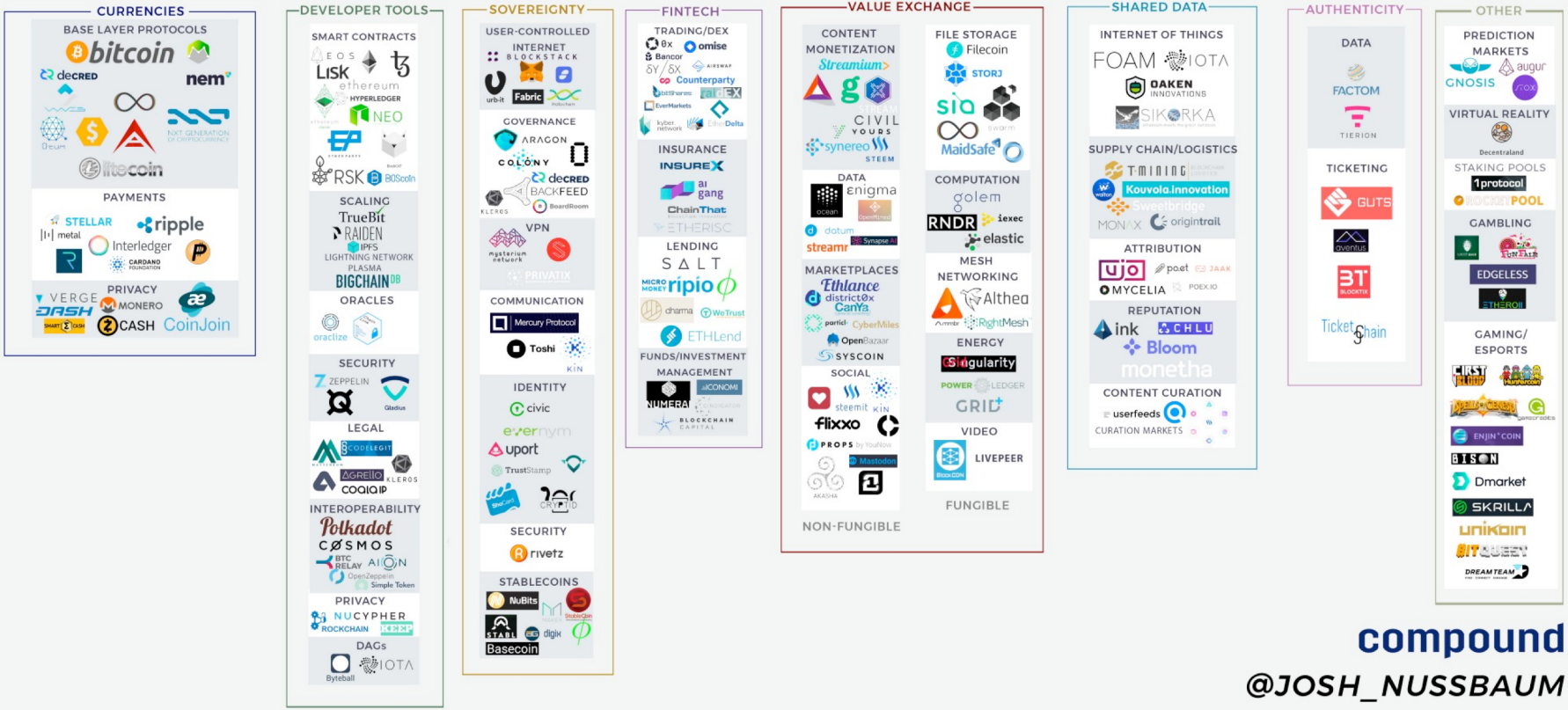
Blockchain Insights

Prof. Dr.-Ing. Stefan Tai

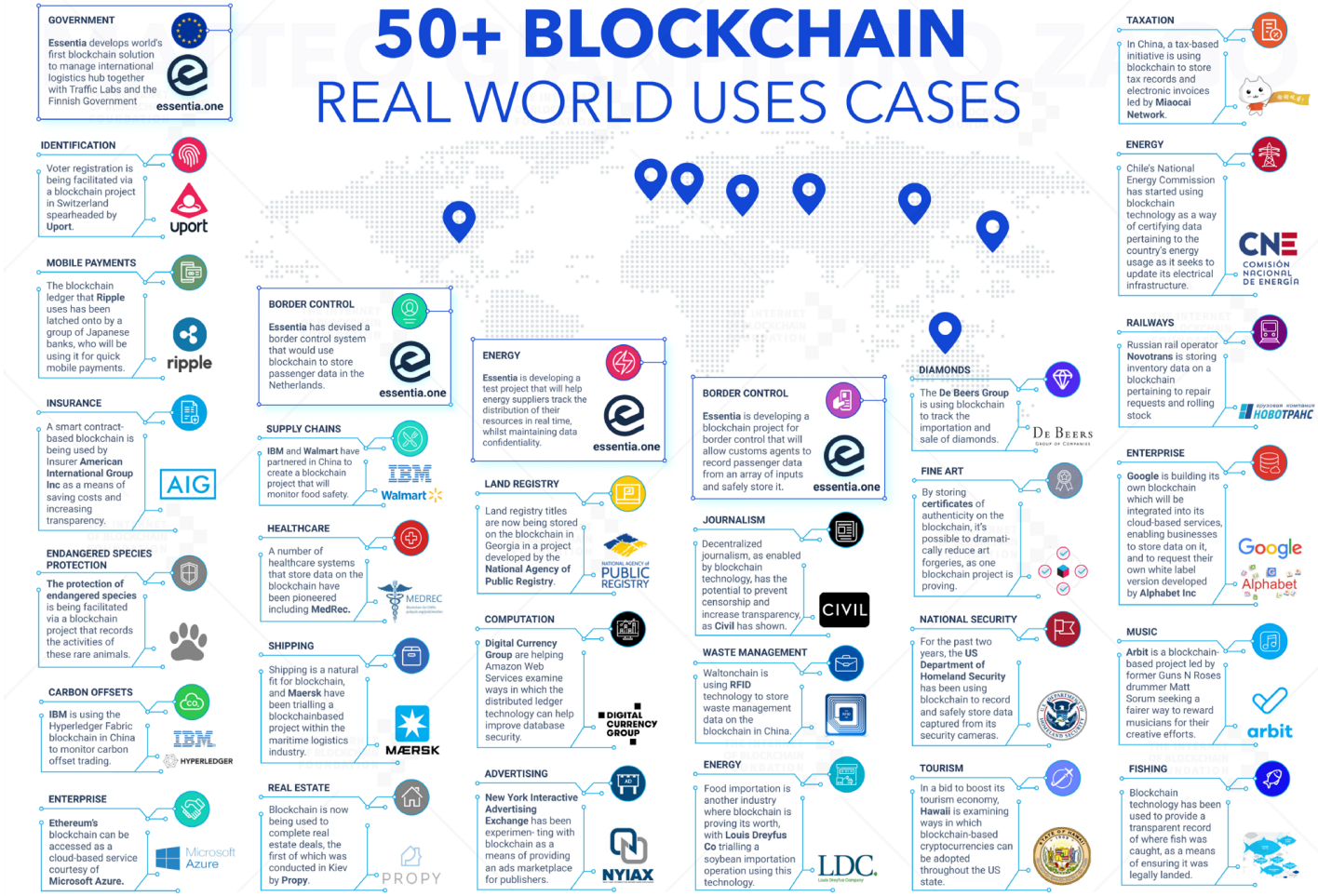
ISEngineering
Information Systems Engineering

https://news.bitcoin.com/berlin-students-chess-ethereum/

BLOCKCHAIN PROJECT ECOSYSTEM



50+ BLOCKCHAIN
REAL WORLD USES CASES



THE INTERNET OF BLOCKCHAIN FOUNDATION

Technische Universität Berlin

ISEngineering
Information Systems Engineering

“If you are not operating at the edge of new technologies, you will surely be disrupted. If you are not willing to embrace new technologies like blockchain […], you are, maybe subtly, at some point … going to extinction.”

FedEx CEO Fred Smith at Consensus 2018

The power and disruption of blockchain is evident...
"…but so are the challenges to its broad implementation."

MIT Sloan Management Review, March 2017

Technische Universität Berlin

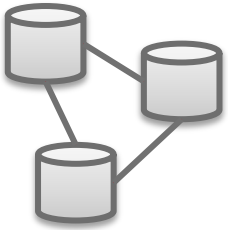
So, what is a blockchain?

Perspectives matter

Technische Universität Berlin

Crypto economy

Decentralized database



Programming platform

Shared ledger

ISEngineering
Information Systems Engineering

What is a blockchain?

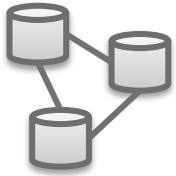
"A blockchain is an economic system"

– Crypto-economic perspective

Blockchains enable the implementation of purely decentralized digital currency (aka cryptocurrency).

- Payment method
- Decentralized incentive and governance mechanism
- Financing model (token sales, ICOs)

What is a blockchain?



"A blockchain is a distributed decentralized database"

– Data management & IT architect perspective

A blockchain is a special type of peer-to-peer database with the following key properties:

- Stores an append-only ordered linked list of transaction records ➔ the transaction history
- The transaction history is fully replicated among all peers using a decentralized consensus protocol
- The transaction history is practically immutable and tamper-proof (under some assumptions)

What is a blockchain?

Technische Universität Berlin

"A blockchain is (part of) a programming platform"

– Developer perspective

Blockchains enable building decentralized applications (DApps):

- Smart contracts = decentralized business logic
- emerging decentralized software stack for storage, messaging, naming, routing, etc.
- "Web 3.0 browsers" and light clients

What is a blockchain?

"A blockchain is a shared ledger"

– Business perspective

Enables business disintermediation

➔ promises lower cost of business transactions

- ➢ Cut out the middleman
- ➢ Single source of truth, golden record
- ➢ Open data platform for value-add services

…and even more perspectives
and combinations thereof…

At the core: Transactions

Recall ACID transactions and RDBMS

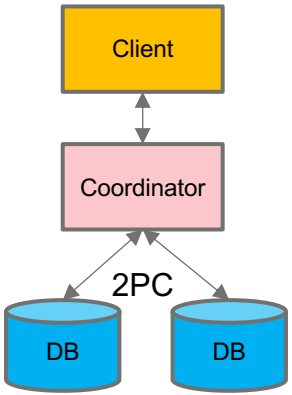
ACID Transaction

Atomicity – all or nothing

Consistency – only valid data

Isolation – no interference

Durability – committed data is never lost



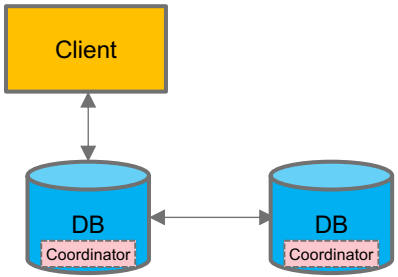
Recall BASE systems and NoSQL stores

BASE Systems

Basically **A**vailable –
partial system failures ok

Soft-state – system state can change even without further updates

Eventually consistent – system will become consistent if no new updates are made



Blockchain transactions and blockchain systems: **Not ACID, not BASE, but SALT**

Sequential – transactions are processed in sequential order

Agreed - community consensus determines transaction validity

Ledgered – all agreed-on transactions are added to an append-only ledger

Tamper-Resistant – A transaction cannot be manipulated or censored

Symmetric – a peer-to-peer network with symmetric responsibilities

Admin-free – no concept of a system admin

Ledgered – all peers maintain a copy of the ledger

Time-consensual – working with block intervals

Comparing ACID, BASE, and SALT

A → ?
C
I
D

Sequential
Agreed
Ledgered
Tamper-proof

(Tx)

Symmetric
Admin-free
Ledgered
Time-consensual

(System)

B
A
S
E

T for Turing Complete?

A → Stored procedures

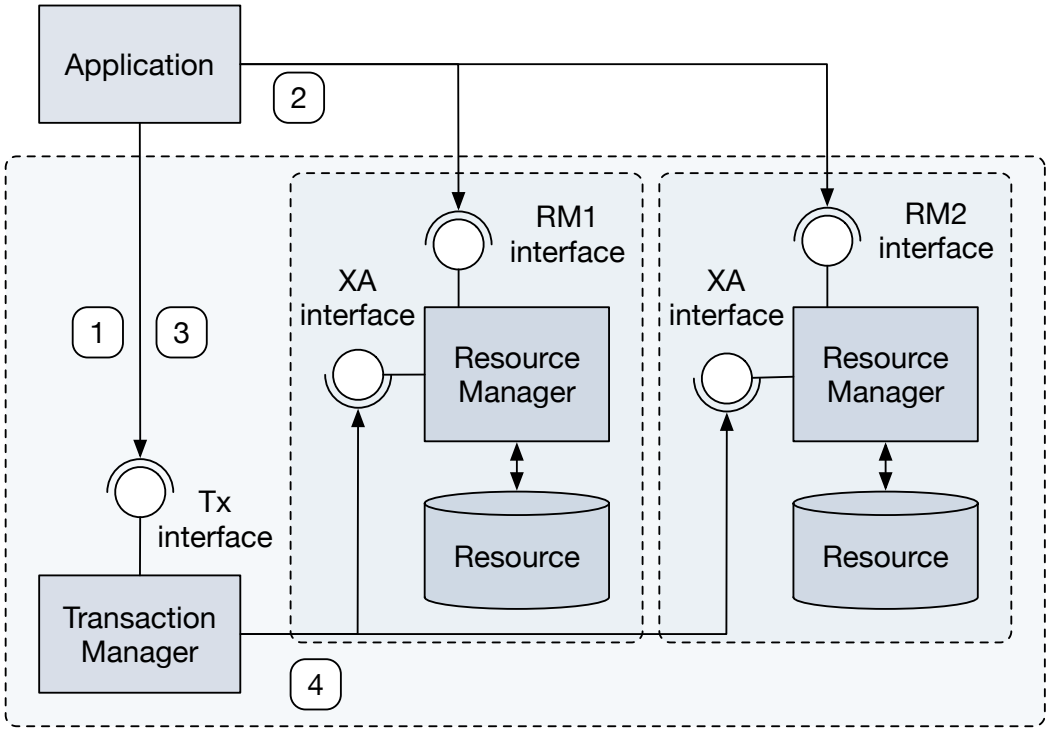
Deterministic computations

Reduced language instructions

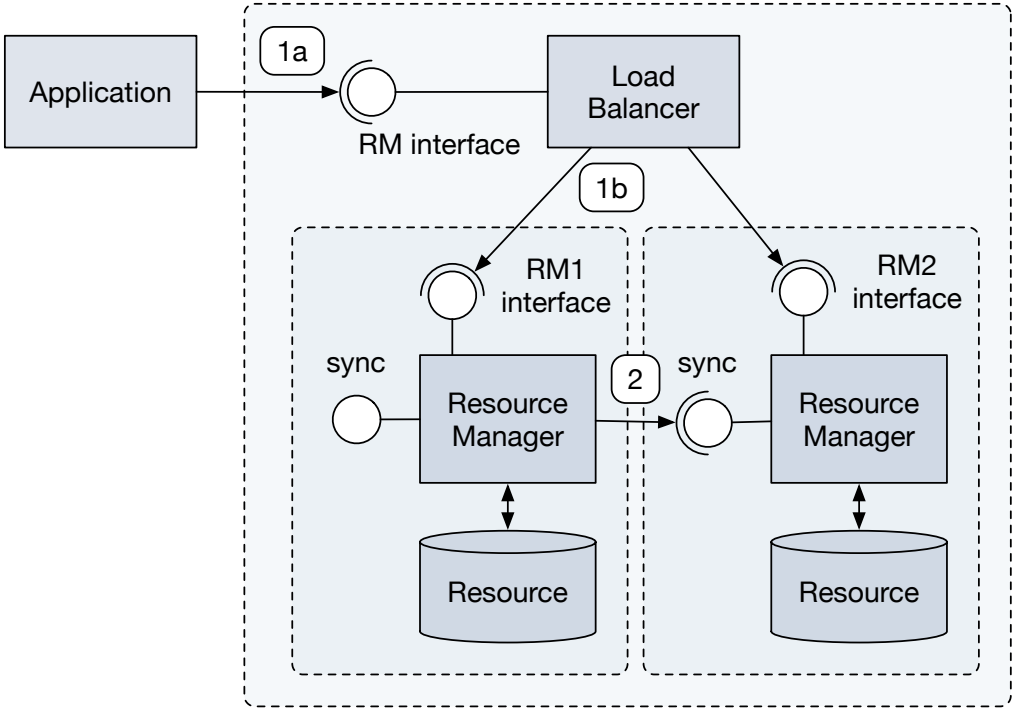
Cost-model for a Turing-complete language

Smart contracts

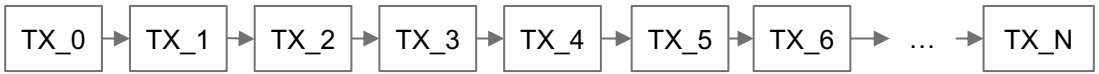
TP systems in support of ACID transactions



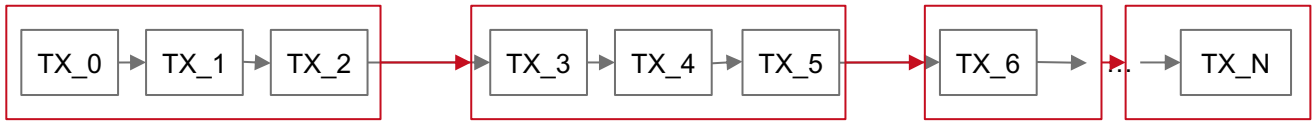
BASE Systems



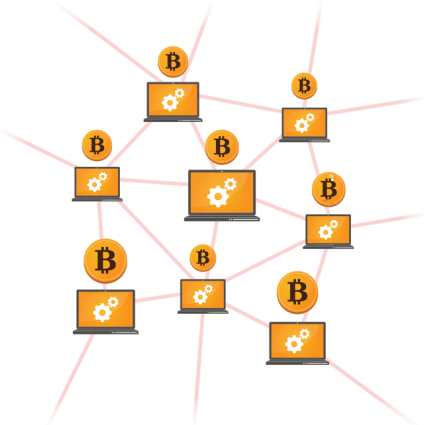
Tx agreement vs. Block agreement



Agreement on single TX ➡ overhead



Blocks of Transactions
- Chronological order
- Reference to predecessor

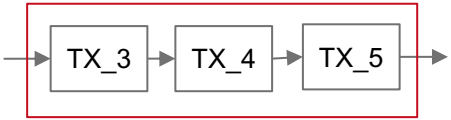


Blocks are…

- Collections of transactions
- Chronologically linked to their predecessor
- The data structure consensus is found on

Solving Consensus

Who is allowed to write a block?



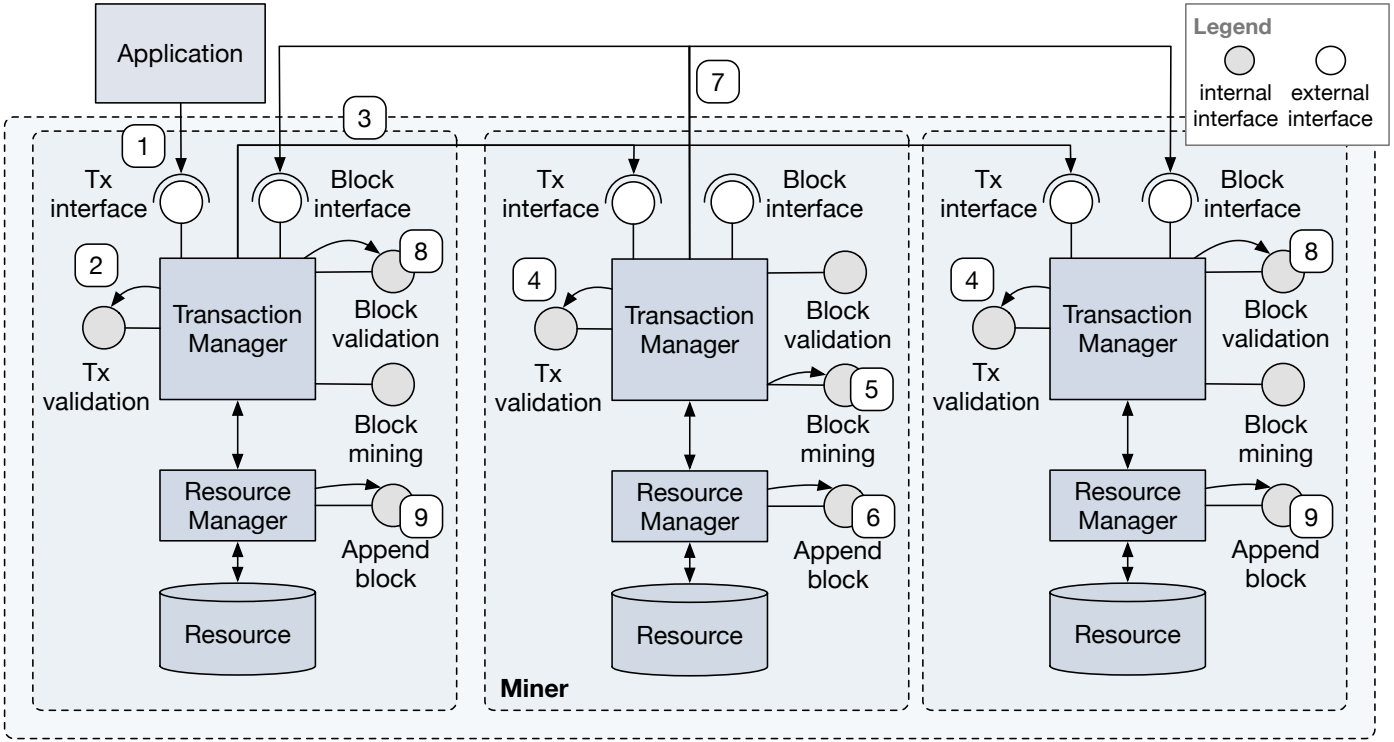
“Traditional” Consensus Algorithm

- ➡ High message complexity

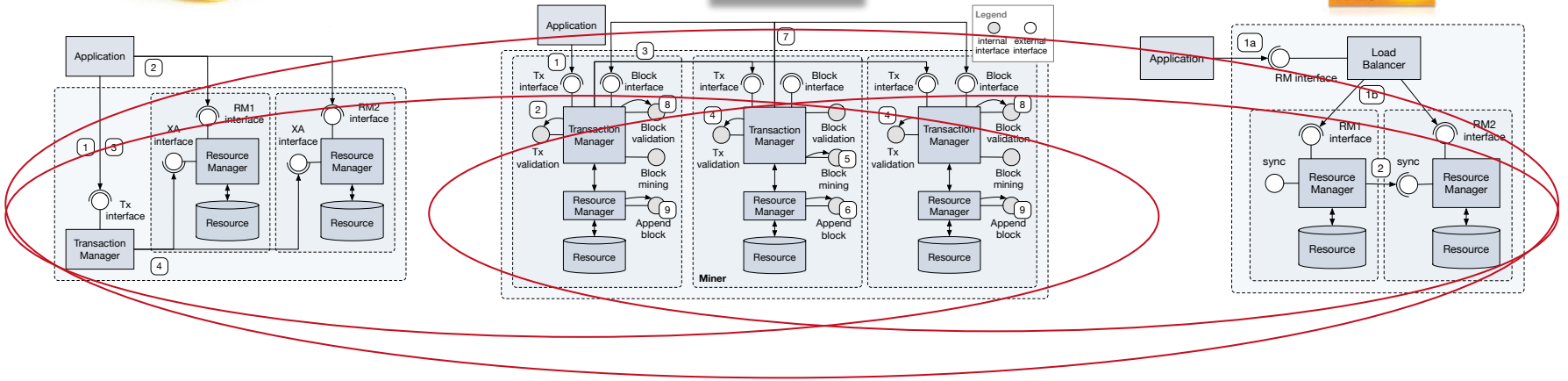
Bitcoin Solution

- ➡ Lottery (Proof-of-Work)
  - All nodes try to solve a hard problem
  - The solution is easy to verify
  - The first node to find a solution, writes a block
  - Other nodes verify solution and accept the block

Understanding SALT



Applications will likely use a combination of all three transaction and system models

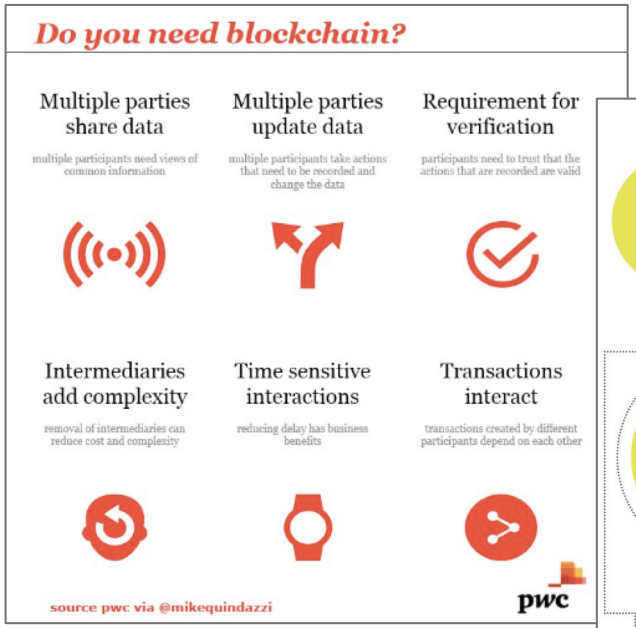


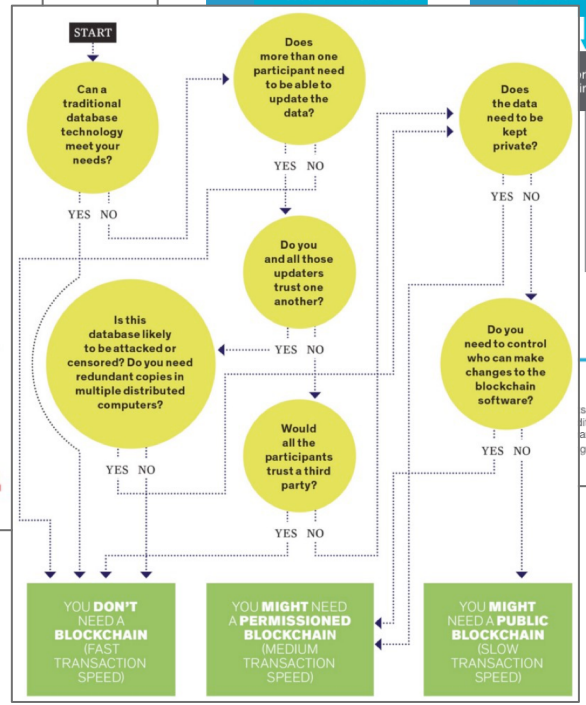
Still SALTy? Well-seasoned or just bad taste?

Technische Universität Berlin

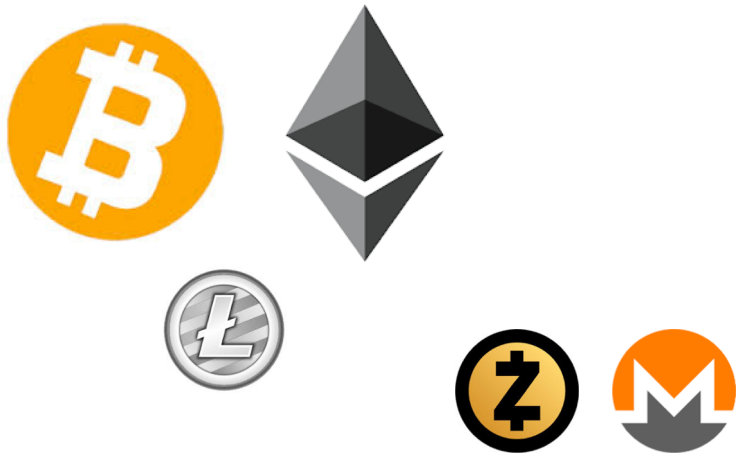
Why should you care?

Asking the right questions may help





Not a single system, but different types of blockchain systems exist



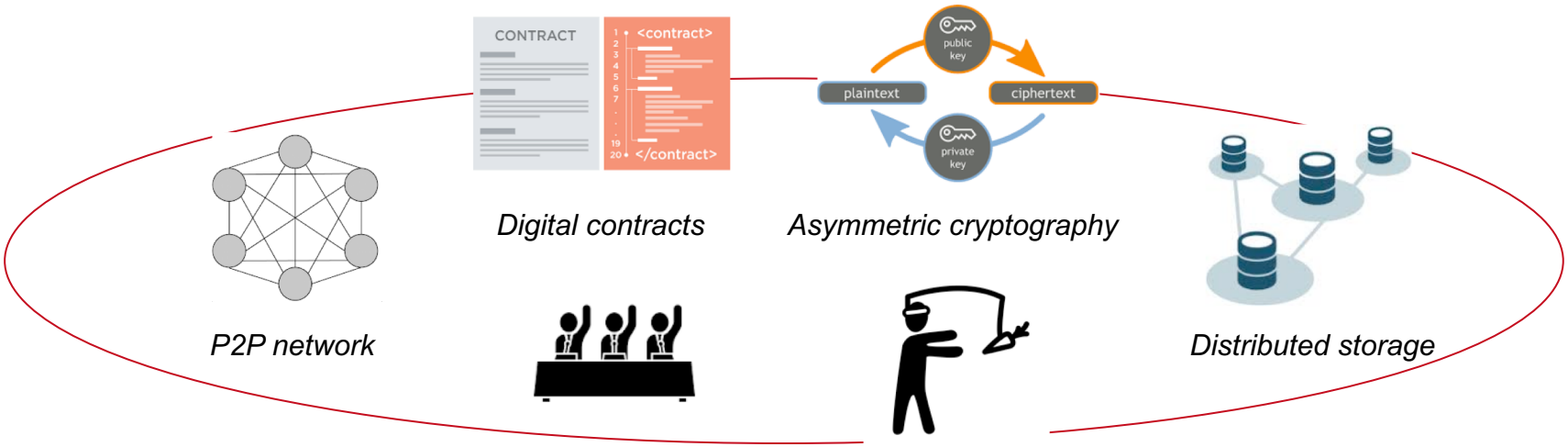
public

private

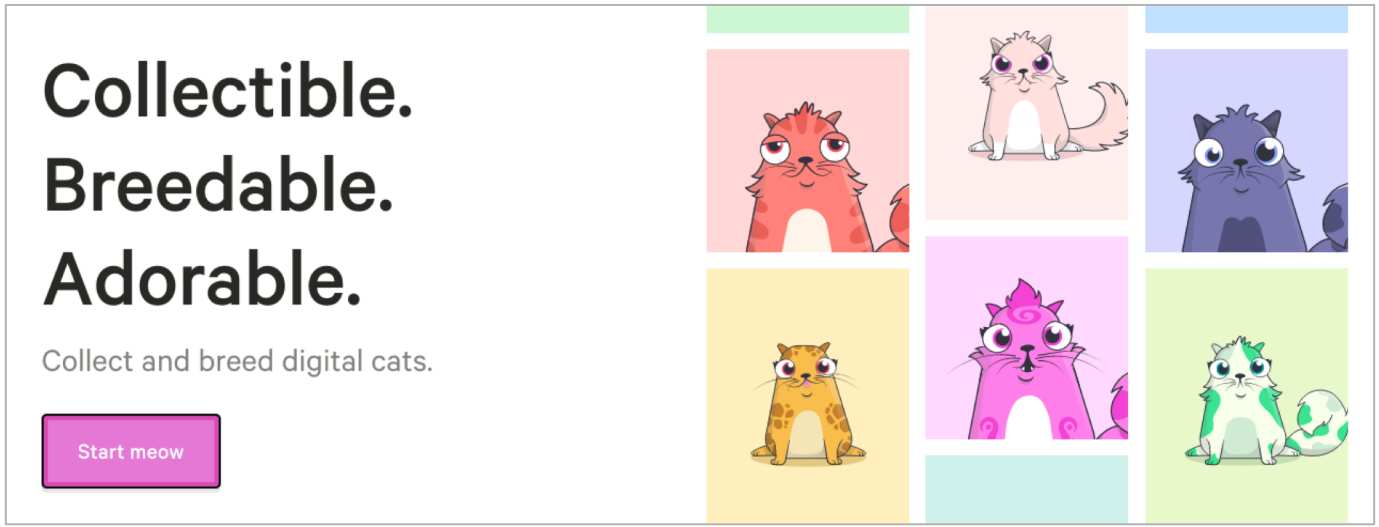
…with different characteristics

	Public (Permissionless)	Private (Permissioned)
Identity management	anonymous or pseudonymous peers	KYB/KYC checking, PKI
Access management	open participation, fully decentralized	central authorities, only partly decentralized
Consensus protocol (typical)	Proof-of-Work, Proof-of-Stake	Proof-of-* and/or BFT protocols
Trust	trustless (= no trust required)	some trust in central authorities is required
Applications (typical/envisioned)	cryptocurrency, web 3.0	cross-organization and cross-border enterprise applications

Look at all constituent parts of a blockchain

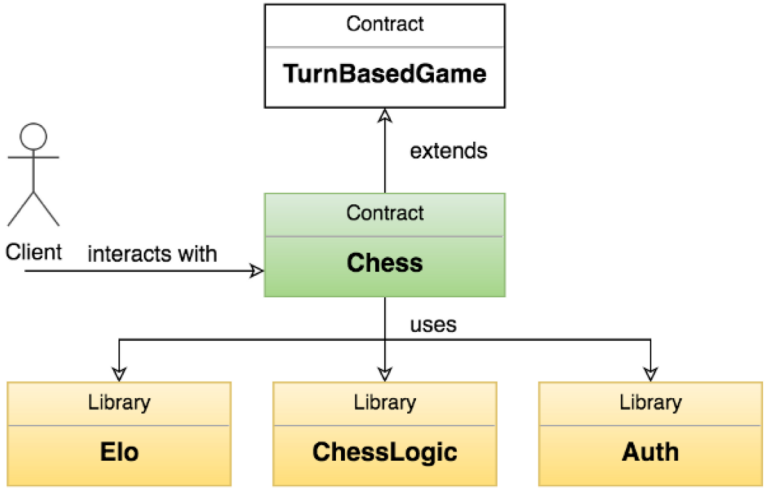


cryptokitties.co



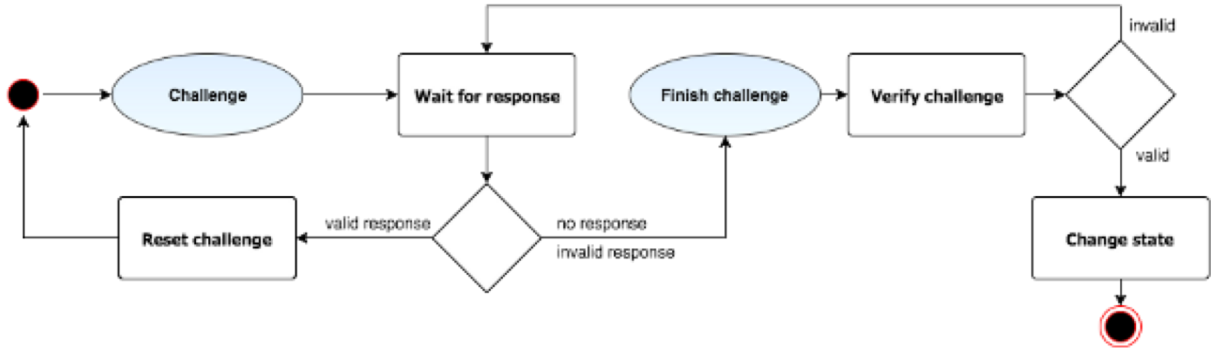
Lessons (to be) learned

Simple chess game, tough challenges



- Checkmate condition is too complex to be checked on-chain. We need to find an alternative trustless way to check conditions.
- Computations cost money. Hence, like in a physical chess game, we should have a player trigger endgame condition checks instead of doing them after every valid move.

Challenge-Response Pattern



Context:

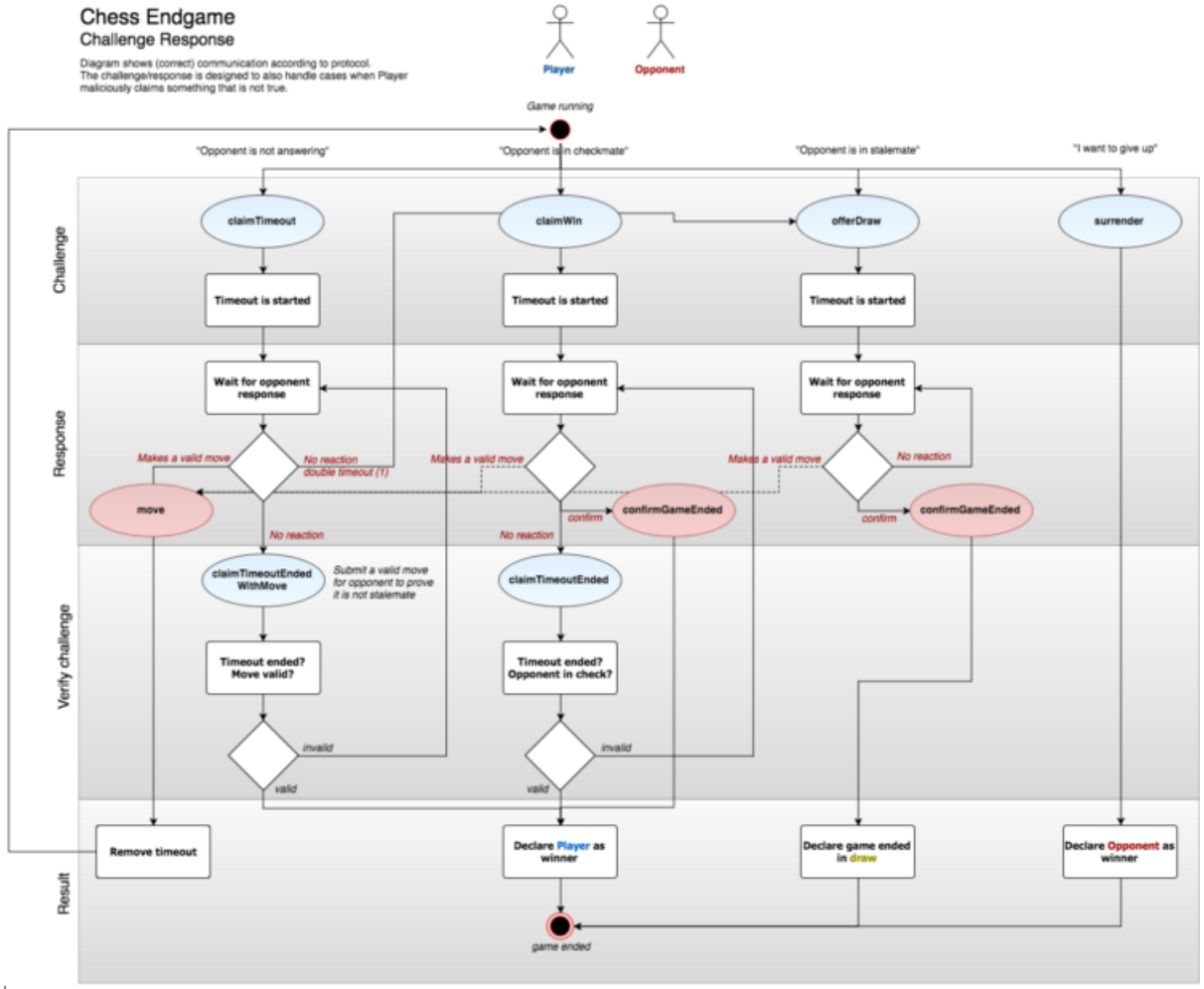
- A smart contract models a state machine with well-defined final states.
- State transitions are cheap to compute, but checking whether a given state is a final state is expensive or may not be possible at all.

Solution:

- Perform the check off-chain on the client side. A client can notify a smart contract when a final state has been reached.
- Other clients can prove claims wrong by providing a valid state transition.

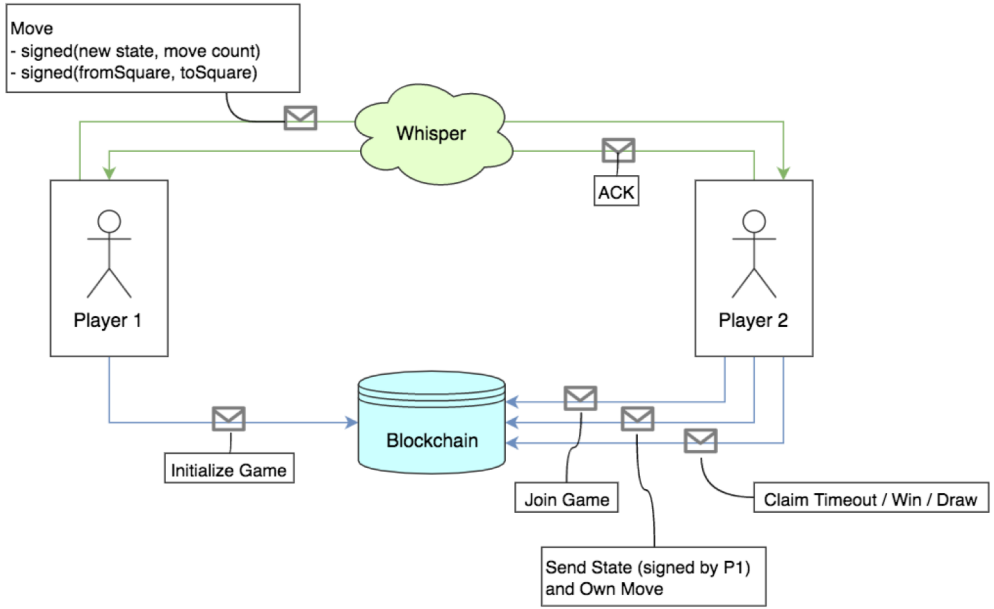
Chess Endgame
Challenge Response

Diagram shows (correct) communication according to protocol.
The challenge/response is designed to also handle cases when Player maliciously claims something that is not true.



(1) In case of stalemate, if Player falsely claimed a win, neither the Player nor the Opponent would have a chance to do anything, because that state can only be resolved when there is a valid move. Because of that, an additional way to resolve the state is added: After two times the timeout, both players are allowed to offer a draw.

Off-Chain Signatures Pattern



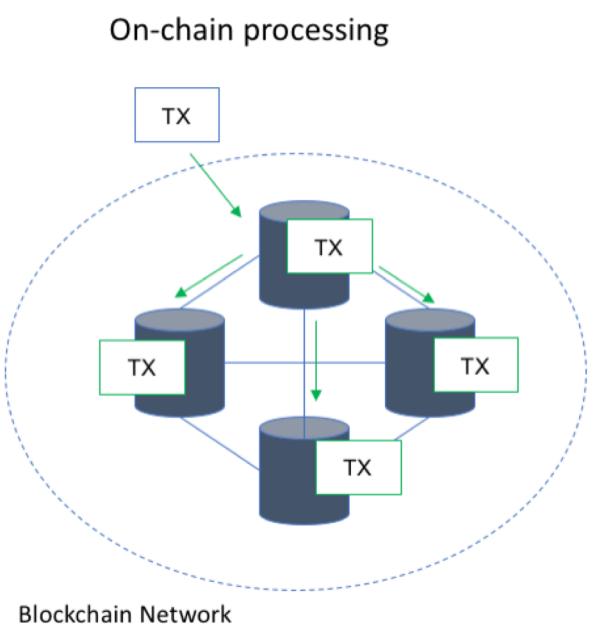
Context:

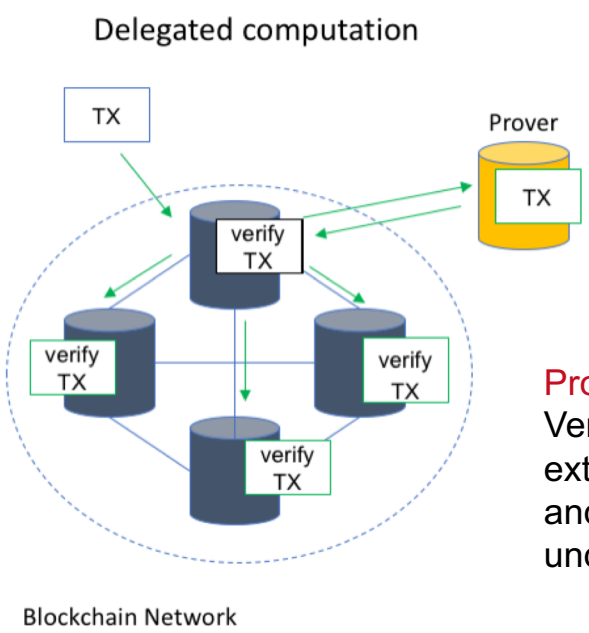
- Two network participants want to transact with each other multiple times in the future.
- They want to reduce the cost of these transactions or want to hide them from others.

Solution:

- Specify a smart contract including a function, which applies an external state given as argument to the contract state.
- This function includes a signature check to ensure both participants agree with the state change.
- The participants perform transactions purely off-chain and peer-to- peer, without involving the blockchain.
- Any transaction, signed by both parties, can then be sent to the smart contract by a participant at any point in time. After validating both signatures, the contract updates its state accordingly.

Delegated Computation Pattern



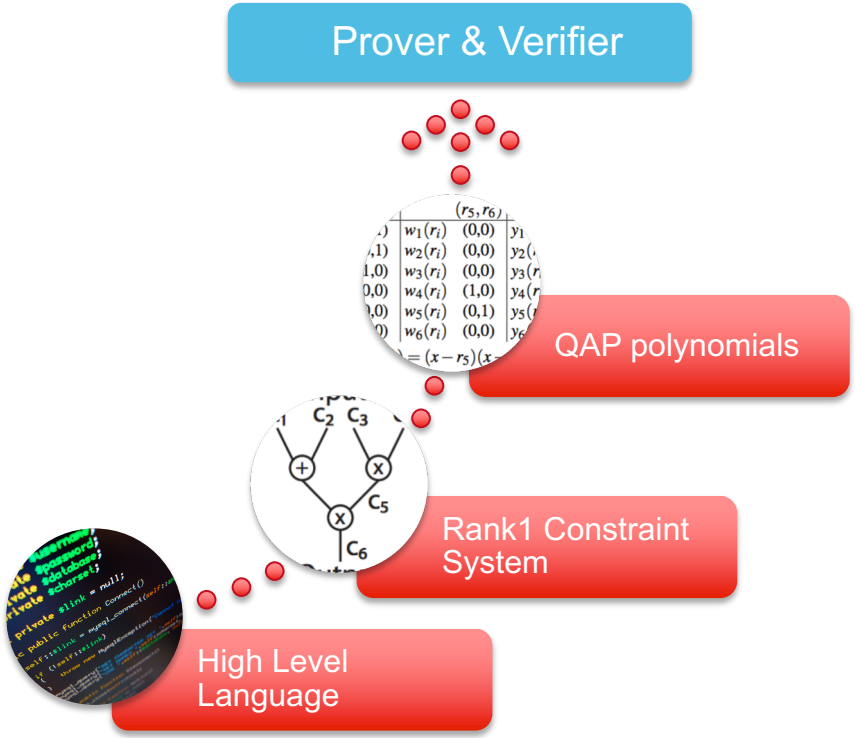


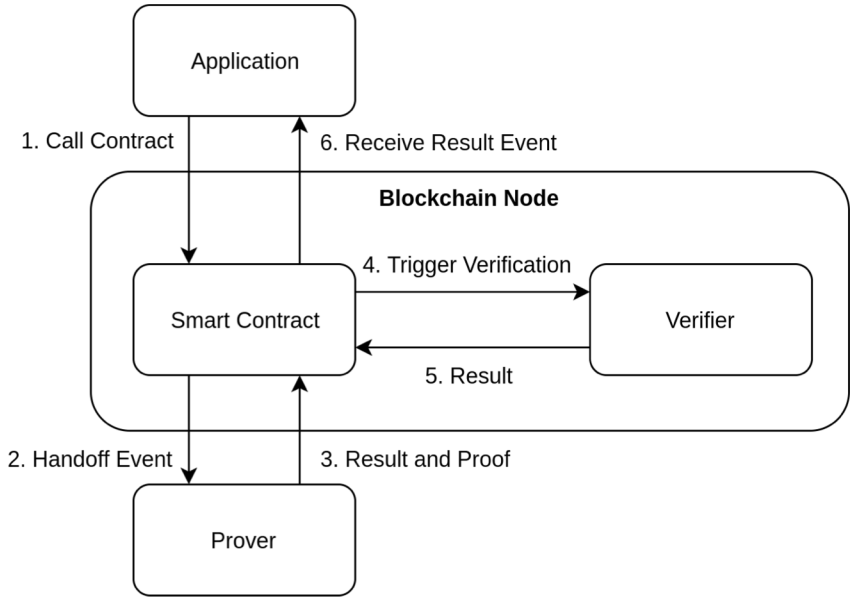
Problem:
Verifiable computations are extremely complex to specify and require deep technological understanding

ZoKrateS – A Toolbox for Off-Chaining

https://github.com/JacobEberhardt/ZoKrates

Technische Universität Berlin





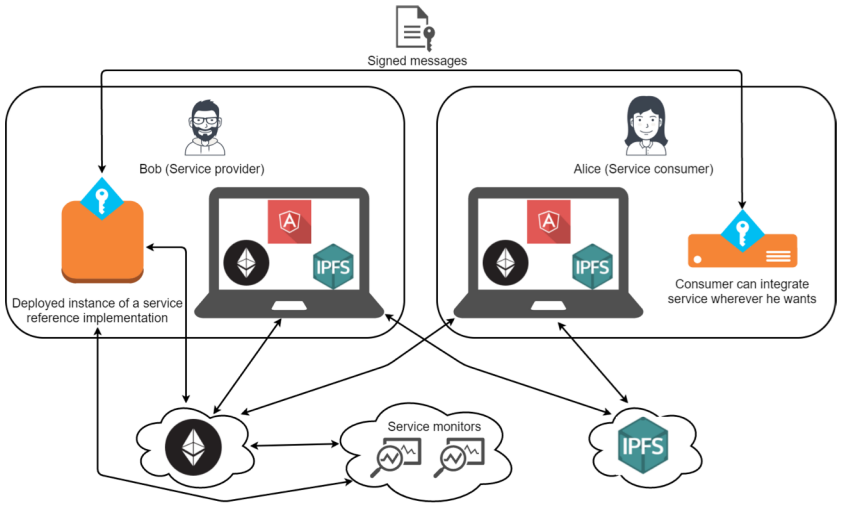
ISEngineering
Information Systems Engineering

Technische Universität Berlin

Come work with us / collaborate!

ISEngineering
Information Systems Engineering

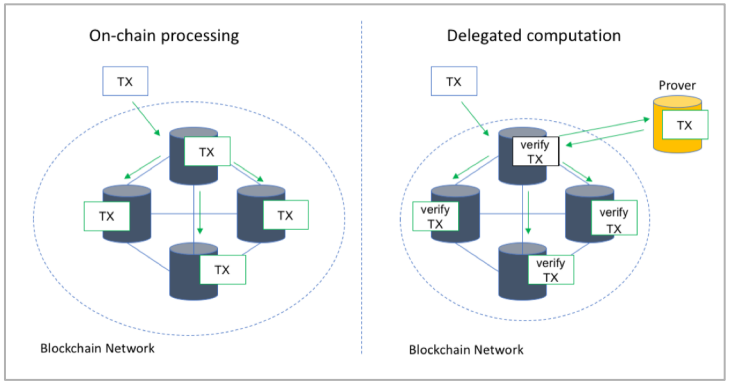
Blockchain Expertise in the ISE Team



Blockchain-based Service Marketplace

http://www.ise.tu-berlin.de/fileadmin/fg308/publications/2017/2017-klems-eberhardt-tai-service-marketplace.pdf

https://news.bitcoin.com/berlin-students-chess-ethereum/



ZoKrateS – A Toolbox for Off-Chaining
https://github.com/JacobEberhardt/ZoKrates

Technische Universität Berlin

Not ACID, not BASE, but SALT
A Transaction Processing Perspective on Blockchains

Stefan Tai, Jacob Eberhardt and Markus Klems
Information Systems Engineering, Technische Universität Berlin

On or Off the Blockchain?
Insights on Off-Chaining Computation and Data

Jacob Eberhardt and Stefan Tai

Information Systems Engineering (ISE)

{j

Abstract. The potenti
how organizations produ
tical applications dealin
strate this capacity. Blc
from a set of unique pro
of cryptographically-secu
been agreed upon by ne
however, may also suffer
negatively impacting ov
we report on lessons lea
mental blockchain proje
putation and data off-t
introduced and benefits

1 Introduction

Blockchains are a combinatic
predominantly including pee

ment syste
ailability,
of blockch
paper, we
ke both, a

Trustless Intermediation in Blockchain-based Decentralized Service Marketplaces

Markus Klems, Jacob Eberhardt, Stefan Tai, Steffen Härtlein, Simon Buchholz,

tu-berlin.de

tform for sellers
lly assumes that
tlement, and dis-
ntralized system.
nable new forms
ockchain-enabled
etplace interme-
saction costs, by
Desema, our de-
plementation of
in combination

software services and
espite high hopes, ser-
ger scale. Systems like

ZoKrates - Scalable Privacy-Preserving Off-Chain Computations

Jacob Eberhardt, Stefan Tai
Information Systems Engineering (ISE)
TU Berlin
Berlin, Germany
{je, st}@ise.tu-berlin.de

***Abstract*—Scalability and privacy are two major challenges for today's blockchain systems. Processing transactions at every node in the system limits the system's ability to scale. Further, the requirement to publish all corporate or individual information for processing at every node, essentially making the data public, is – despite of all other advantages – often considered a major obstacle to blockchain adoption. In this paper, we make two main contributions to address these two problems: (i) To increase

increase transaction throughput by reducing the cost associated with transaction validation; (b) through off-chaining, remove the requirement to reveal data to all and to become public but allow data to remain private, yet, private data must still have an influence on the transactions processed on-chain.

Transaction throughput is directly linked to the cost implied for clients for block validation. While only mining nodes

ng

Technische Universität Berlin

Thank you!

Stefan Tai
tai@tu-berlin.de
ise.tu-berlin.de

ISEngineering
Information Systems Engineering