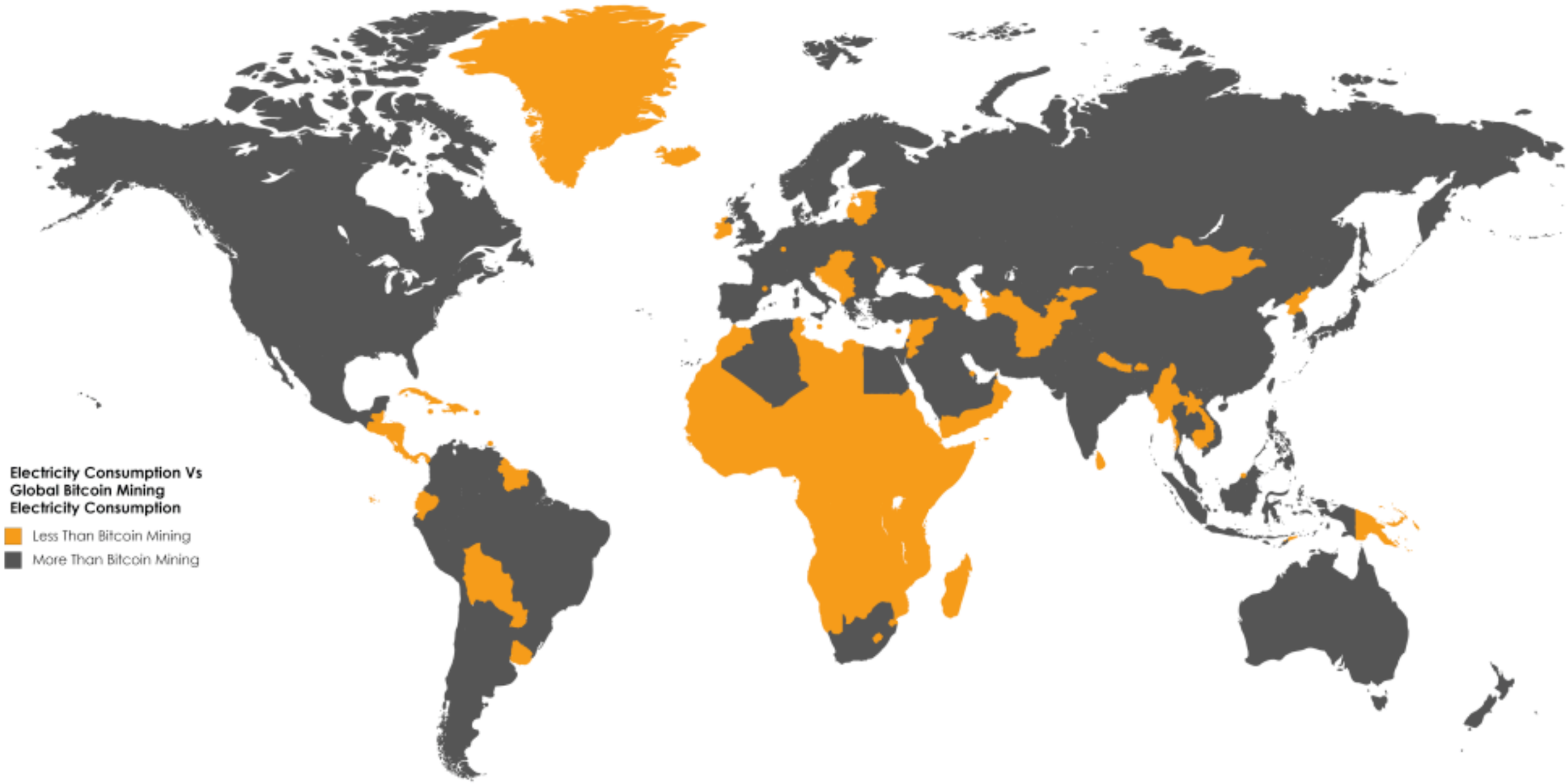
Miscellaneous

50.037 Blockchain Technology

Paweł Szałachowski

Proof-of-Stake (aka Virtual Mining)

Energy Consumption



Source: https://powercompare.co.uk/bitcoin/

Proof-of-Stake (PoS)

- Any proof-of-*resource* can be translated to *proof-of-money*
- So why not remove this intermediary and ``mine" with cryptocurrency stakes?
    - Miners instead of buying hardware, buy stake and vote with it
        - Sybil attacks eliminated
- Energy-friendly, more efficient, reduce ASIC-caused centralization, any cryptocurrency holder is a stakeholder

PoW

PoS

Peercoin

- https://peercoin.net/assets/paper/peercoin-paper.pdf (2012)
- Hybrid PoW/PoS with stake denominated by coin-age

  coin-age = (amount of UTXO) × (# of blocks it remains unspent)

- Block mining as in Bitcoin: H(header) < T
  - T is adjusted based on how much coin-age they are willing to consume
    - Blocks include a special coinstake Tx which spends some transactions to reset their coin-age to zero
    - The sum of the coin-ages consumed in the coinstake transaction decides how difficult the proof-of-work puzzle is to make a given block valid
- Miners can balance PoW/coin-age but it is much easier to find a solution consuming some coin-age

PoS

- Pure PoS: vote proving that you control some stake
    - Richest participants always stay richest
- The nothing-at-stake-problem
    - An adversary tries to create a fork of k blocks (would fail with high probability)
    - In PoW mining that would cost the adversary a lot of resources, but with PoS?
    - Rational miners would constantly attempting to fork the chain
    - Mitigations: checkpoints, punishments, etc …
- Some of these systems are getting close to classic BFT consensus protocols
    - Similar setting and problems

PoS

- Not tested as PoW, new attack vectors likely to be found
  - Stake centralization
  - 51% miner can control the chain forever
    - No new powerful miner can emerge, like in PoW
- Is that a permissionless system anymore?
  - Someone needs to give us stake
- Active research: Ethereum’s Casper, Algorand, …

Atomic Cross-chain Swaps

Atomic Swaps

- In Bitcoin, it is easy to move tokens controlled by different entities
  - Txs are confirmed on the blockchain
- Can we swap one type of coin for another?
  - Alice wants to sell a quantity **a** of altcoin to Bob in exchange for a quantity **b** of his bitcoin
  - Tx should be atomic and without a trusted intermediary
    - Easy to solve w/ a trusted intermediary (exchange), but w/o?

1. Alice generates a refundable deposit of *a* altcoins as follows:
   - 1.1 Alice generates a random string *x* and computes the hash *h=H(x)*
   - 1.2 Alice generates **DepositA** as shown below, but doesn't publish it yet
   - 1.3 Alice generates **RefundA**, and gets Bob's signature on it
   - 1.4 Once Bob signs **RefundA**, she publishes DepositA (but doesn't publish **RefundA**)
2. Bob generates a refundable deposit of *b* bitcoins as follows:
   - 2.1 Bob generates **DepositB** as shown below, but doesn't publish it yet
   - 2.2 Bob generates **RefundB**, and gets Alice's signature on it
   - 2.2 Once Alice signs **RefundB**, he publishes **DepositB** (but doesn't publish **RefundB**)
3. Case 1: Alice goes through with the swap
   - 3.1 Alice claims the bitcoins by time T_1, revealing *x* to Bob (and everyone) in the process
   - 3.2 Bob claims the altcoins by time T_2

   Case 2: Alice changes her mind, does not claim the altcoins, does not reveal *x* to Bob
   - 3.1 Bob claims his altcoin refund at time T_1
   - 3.2 Alice claims her Bitcoin refund at time T_2

DepositA [Altcoin block chain]

Input:	Alice's coins of value *a*
ScriptPubkey:	Redeemable by providing either (*sigA* and *sigB*) or *sigB* and *x* s.t. *H(x) = <h>*

→

RefundA [Altcoin block chain]

Input:	DepositA
Output:	AddrA
Timelock:	T_2
ScriptSig:	*sigA, sigB*

DepositB [Bitcoin block chain]

Input:	Bob's coins of value *b*
ScriptPubkey:	Redeemable by providing either (*sigA* and *sigB*) or *sigA* and *x* s.t. *H(x) = <h>*

→

RefundB [Bitcoin block chain]

Input:	DepositB
Output:	AddrB
Timelock:	T_1
ScriptSig:	*sigA, sigB*

State Channels

Payments in Bitcoin

- Issues
  - Scalability and Tx latency
  - Micropayments
    - High fees if done “on-chain” w/o trusted parties
- Idea: 2nd layer scalability
  - So far we discussed “on-chain” scalability (1st layer)
  - Consider micropayments: do we need to settle all Txs on chain?
  - Payment channels: “off-chain” Txs with the final balance recorded on-chain
    - Can be generalized to any state (state channels), e.g., smart contracts

Payment Channel (unidirectional)

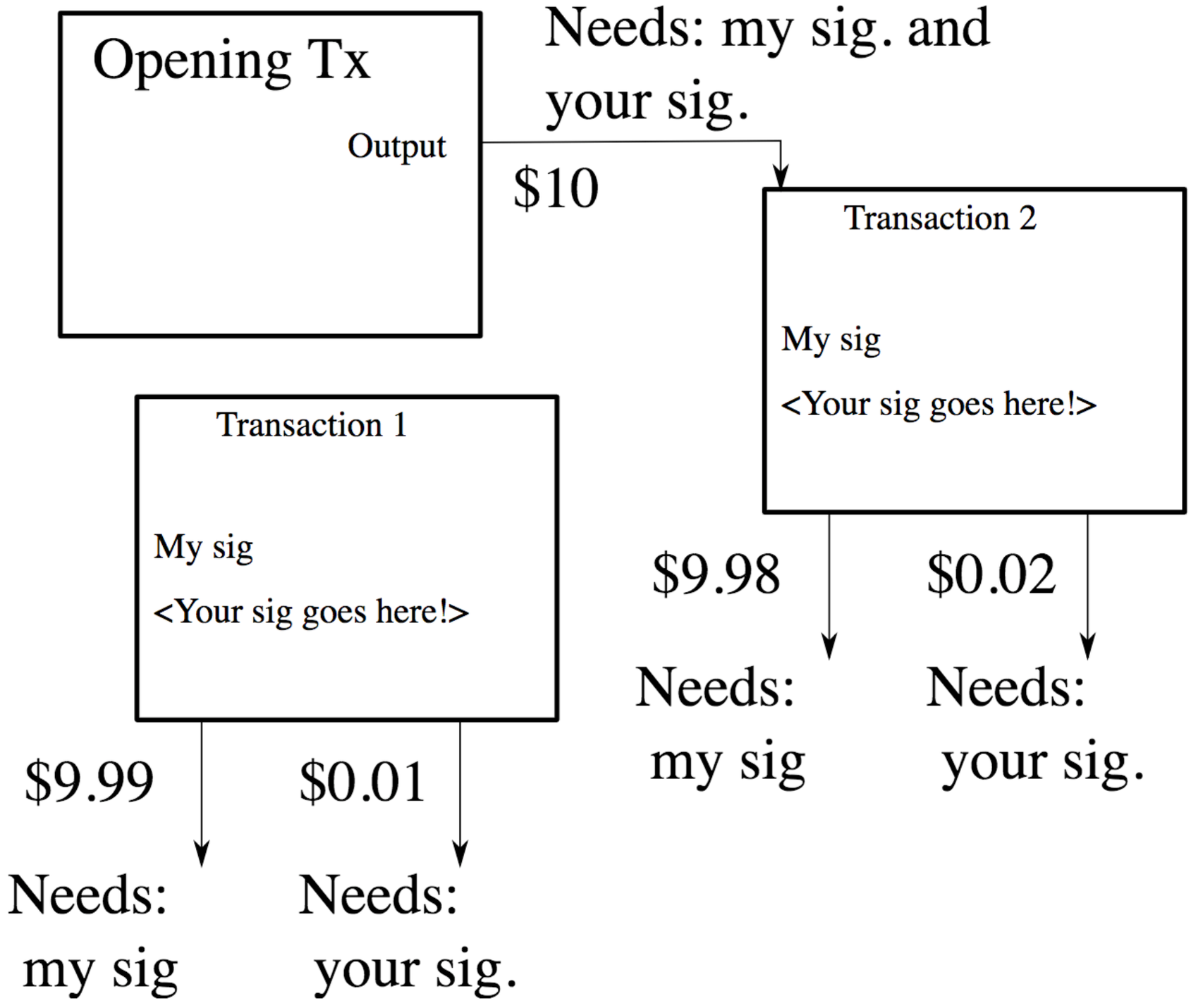
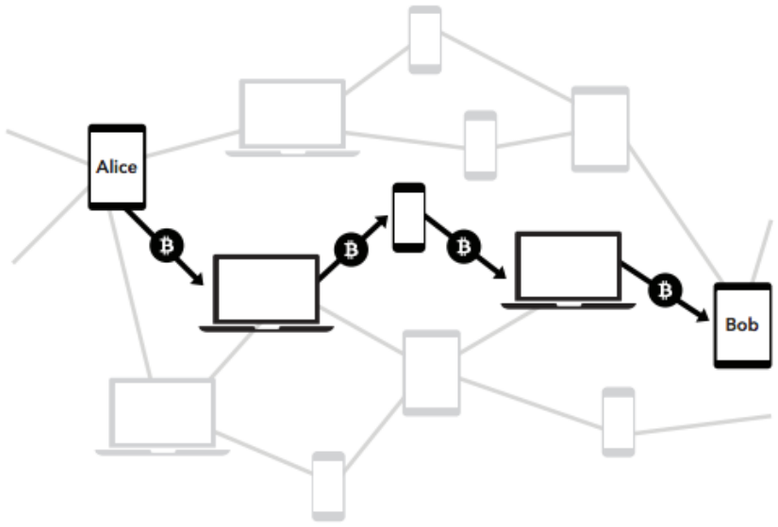


Fig src: https://rusty.ozlabs.org/

Payment Networks

- Payment Networks (e.g., Lightning https://lightning.network/)
  - Do we need to establish many p2p channels? Multiple “hops”
    - Low fees, fast, high throughput, buyers are protected, sellers need to watch the blockchain
- Many off-chain solutions
  - Not only payments, smart contracts too
    - Plasma, Arbitrum, …



Data Feeds for Smart Contracts

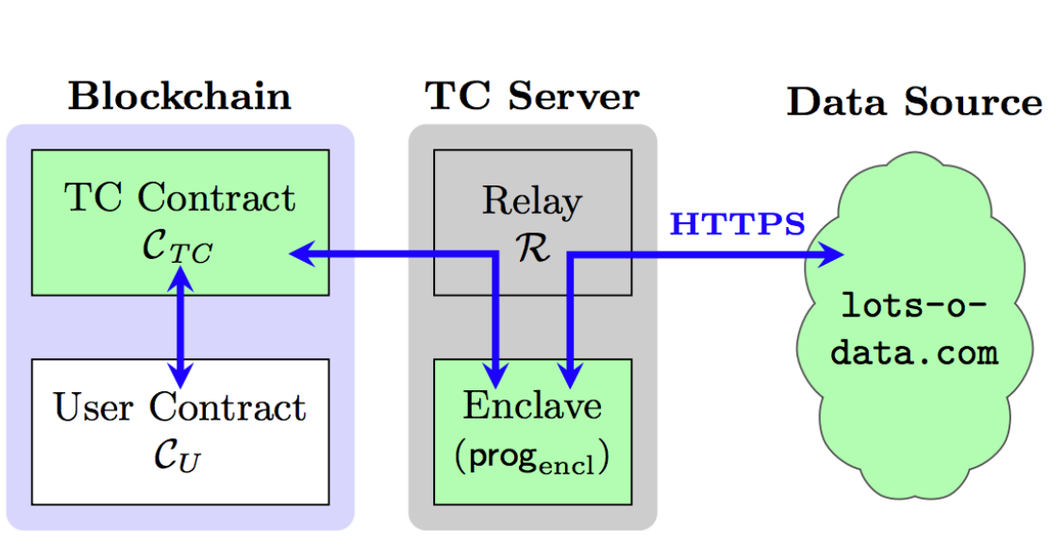
Problem

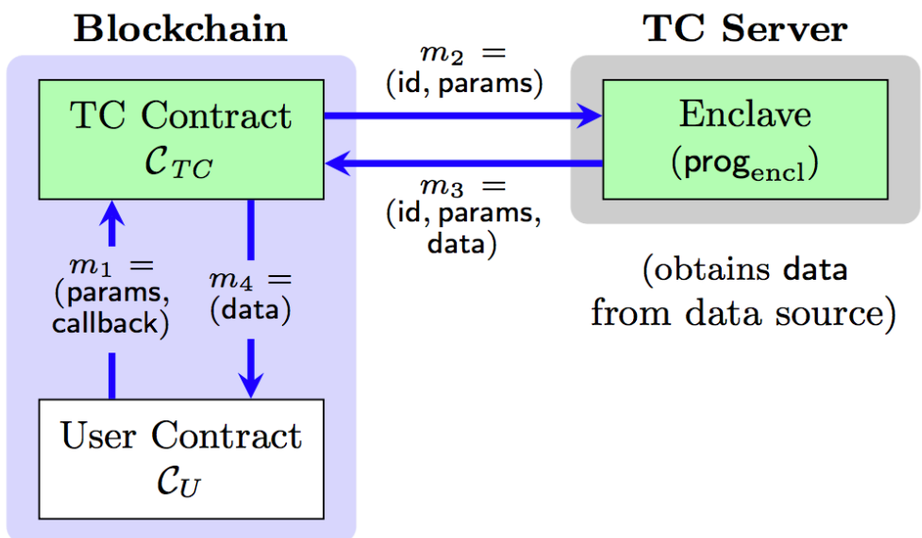
- How to empower smart contracts so they can process data from external infrastructures?

- A lot of data available via HTTPS (HTTP over TLS)
    - It would be great to use it in smart contracts
    - Use a proxy? Need to trust it
- TLS does not provide non-repudiation for app data
    - Cannot prove to a third-party authenticity of received data

TownCrier

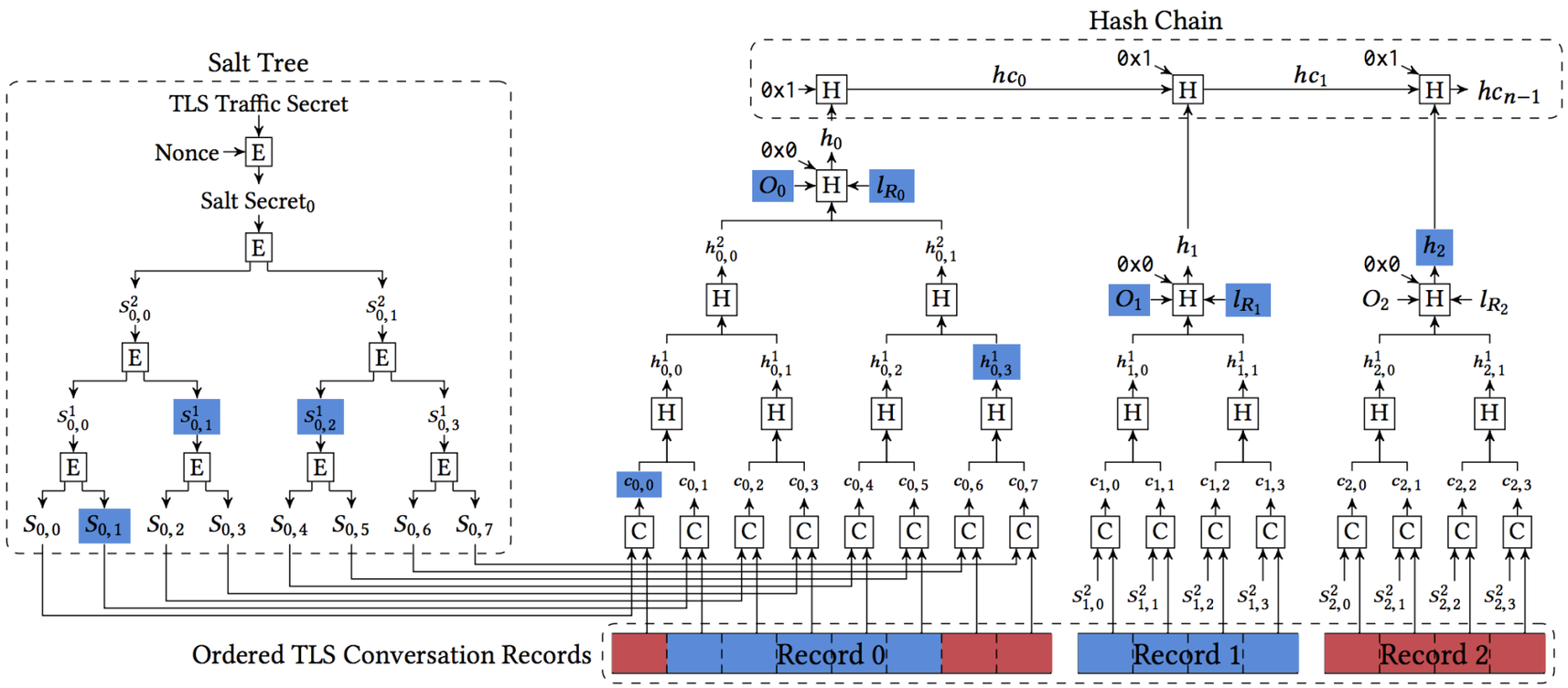
- https://eprint.iacr.org/2016/168.pdf
- A proxy between a blockchain and HTTPS websites
  - but implemented with Intel's SGX
- Pros: easy integration, website operators do not have to deploy
- Cons: needs to trust Intel (single point of failure) and SGX (recent attacks)





TLS-N

- https://eprint.iacr.org/2017/578.pdf
- TLS with non-repudiation
  - Efficient Merkle-tree-based authentication on TLS layer
- Pros: more general and powerful solution, extra features (like privacy)
- Cons: significant changes to the TLS spec, TLS servers have to be updated, difficult/expensive integration with smart contracts (TLS records are signed)



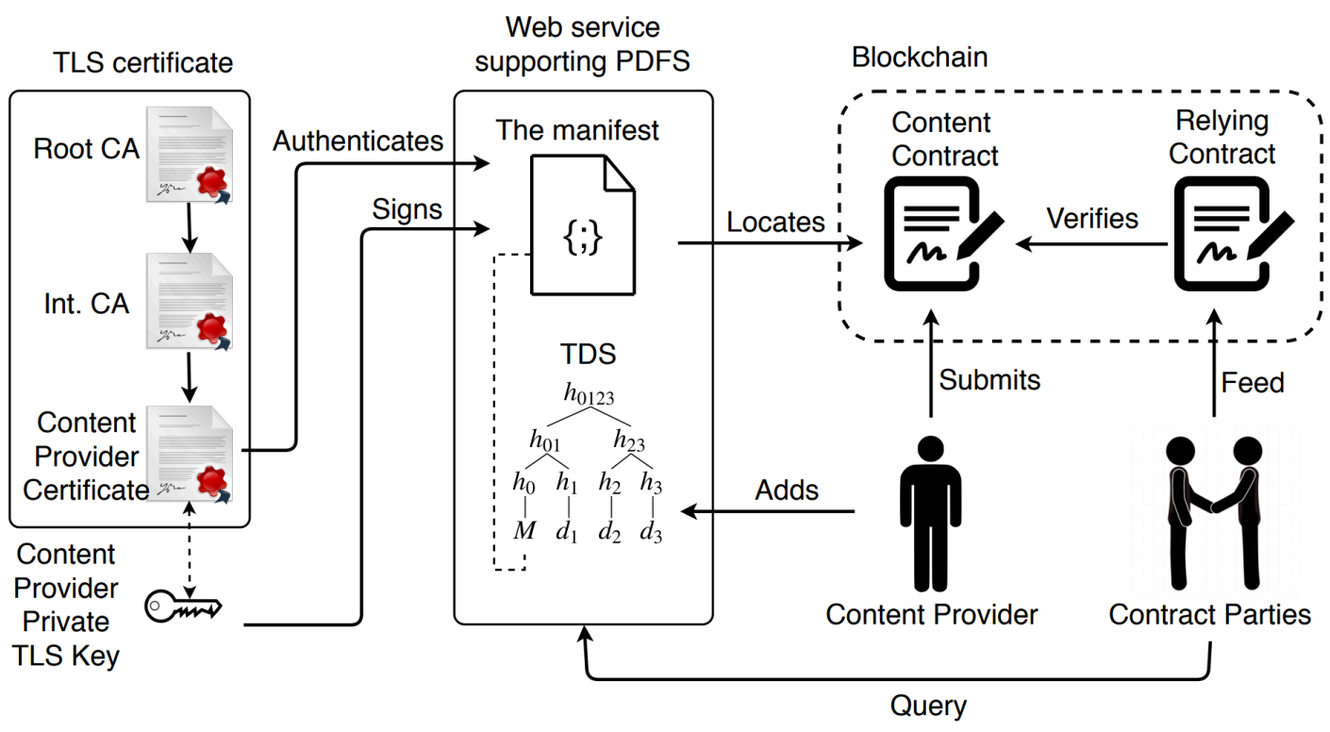
PDFS

- https://arxiv.org/pdf/1808.06641.pdf

- Data providers sign (via TLS) their contract locations with API allowing:
    A. Update the contract with a new root of a database (append-only)
    B. Prove that given data is authentic (is in the database), can be paid
    C. Query for data if provider seems unavailable

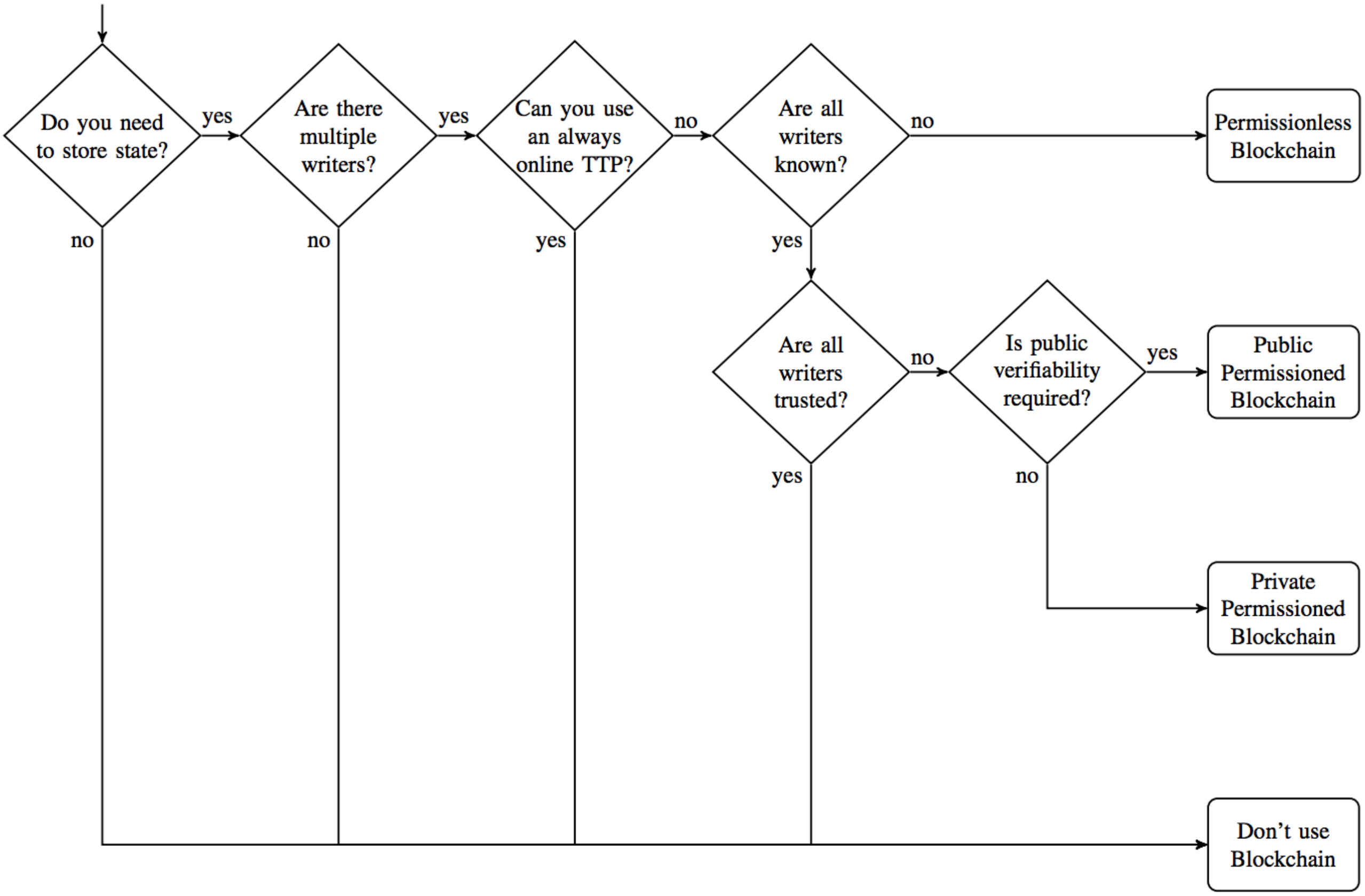
- Pros: direct TLS authentication, easy integration and data format, extra features

- Cons: website operators have to deploy



Do you need a blockchain?

(https://eprint.iacr.org/2017/375.pdf)


Do you need to store state?
yes
Are there multiple writers?
yes
Can you use an always online TTP?
no
Are all writers known?
no
Permissionless Blockchain
no
no
yes
yes
Are all writers trusted?
no
Is public verifiability required?
yes
Public Permissioned Blockchain
yes
no
Private Permissioned Blockchain
Don’t use Blockchain

Reading

- Textbook 10.5

- + inline references