From: http://news.bitcoin.com

Blockchain

Zwischen Genesis und Mondlandung

2016-08-06 Jonas Nick jonasd.nick@gmail.com https://nickler.ninja @n1ckler

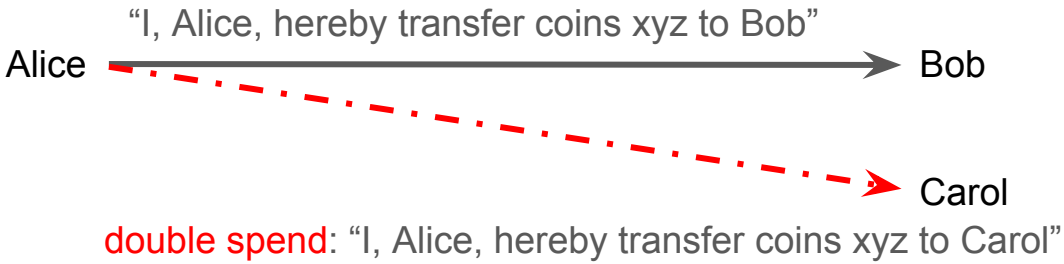
Peer-to-Peer Cash

- Ideal: Internet money without central control and anonymous

```
I've been working on a new electronic cash system that's fully
peer-to-peer, with no trusted third party.

[...]

Satoshi Nakamoto

--------------------------------------------------------------------------
The Cryptography Mailing List
```

A toy currency

- Start with arbitrary bits that you call coins from now on
- Use cryptographic signatures to make forging messages impossible



- A central bank could tell which transaction came first.

A toy currency

- Decentralize control: Shared ledger
  - Every participant keeps a record of the transaction history
  - This works as long you know all the participants and trust a majority.
- But in open peer-to-peer systems
  - It is impossible to know all the participants.
  - It is impossible to meaningfully count votes.
- Want: dynamic membership of the participant set

Bitcoin

- Proof of Work: small proof that some amount of computation was done

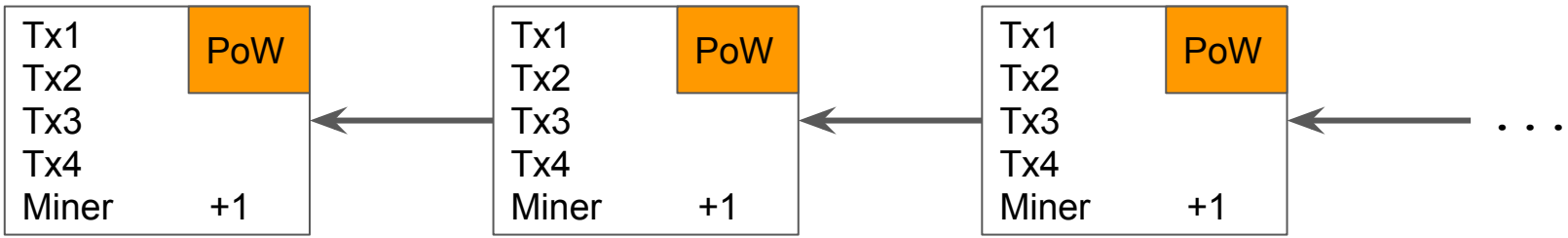
1. Define that the “official” transaction history
    a. is valid
    b. has the most proof of work
2. Providing PoW (mining) to the official history is rewarded with coins

Effect:

1. Consensus on official history.
2. Incentivizes mining on a history. Incentivizes mining on the official history.

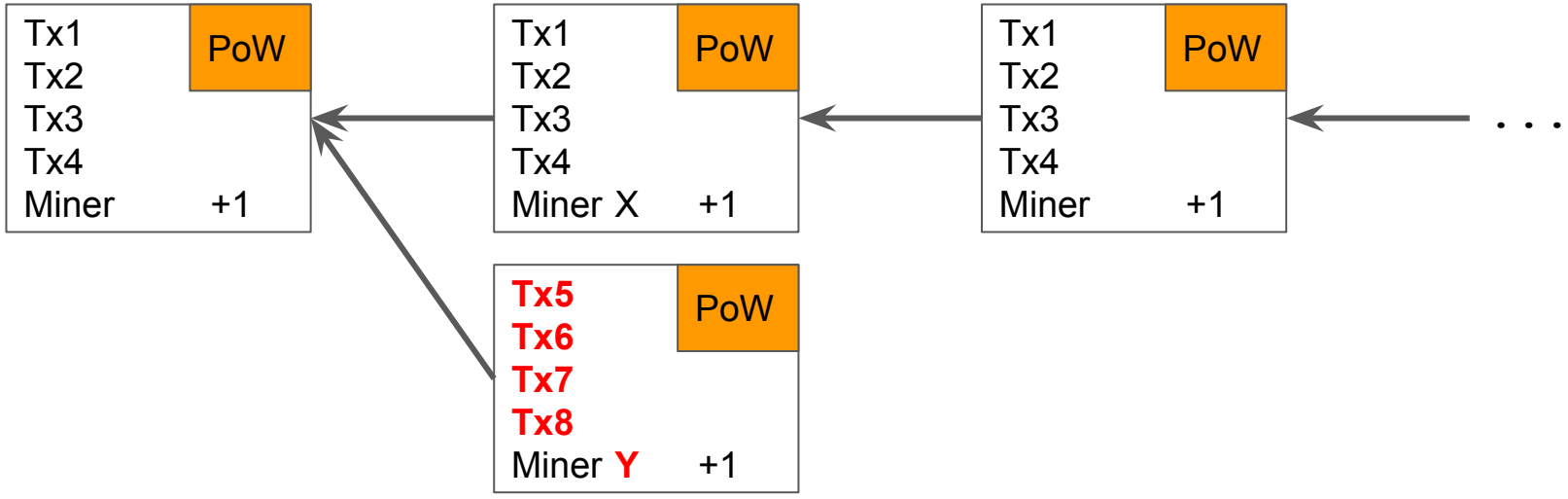
Mining

- History is represented as a chain of blocks.
  - Blocks contain transactions.
- Miners create blocks by collecting transactions.
- And attempt to solve the PoW function.
- Blocks are mined on expectancy every 10 minutes.
- The miner gets a mining reward.



Mining

- Miner attempting to rewrite the history always loses in the long run
  - As long as miner has less than 50% hash rate
- Miners can *not* spend your coins or include invalid transactions
  - f.e. A tx that send more coins than the attacker has available.



Blockchain technology

- 2 years ago: An application that uses Bitcoin in some way
- Now: **Consensus** on shared **censorship-resistant state** with **immutable** rules in a **distributed** environment with potentially **dishonest** nodes.
- Goal: Reduce trust or expensive processes
- Can enable interactions that were previously impossible.

Part 2: Transactions

Transactions

- Balance-based vs. UTXOs
- Balance-based (f.e. Ethereum)

Ledger state

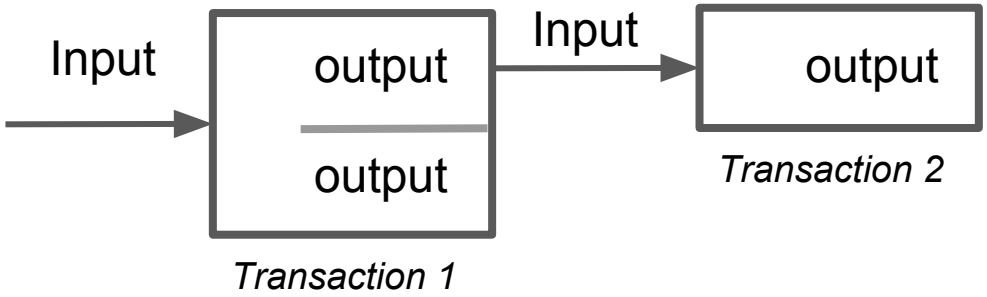
Alice	2
Bob	0

  - Transaction: Alice 1 coin ⟶ Bob

New ledger state

Alice	1
Bob	1

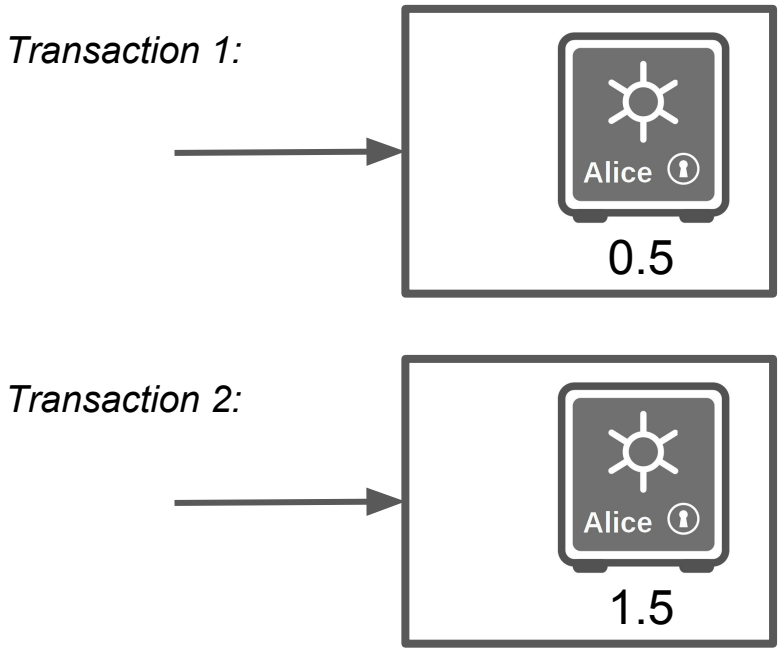
Transactions Inputs & Outputs



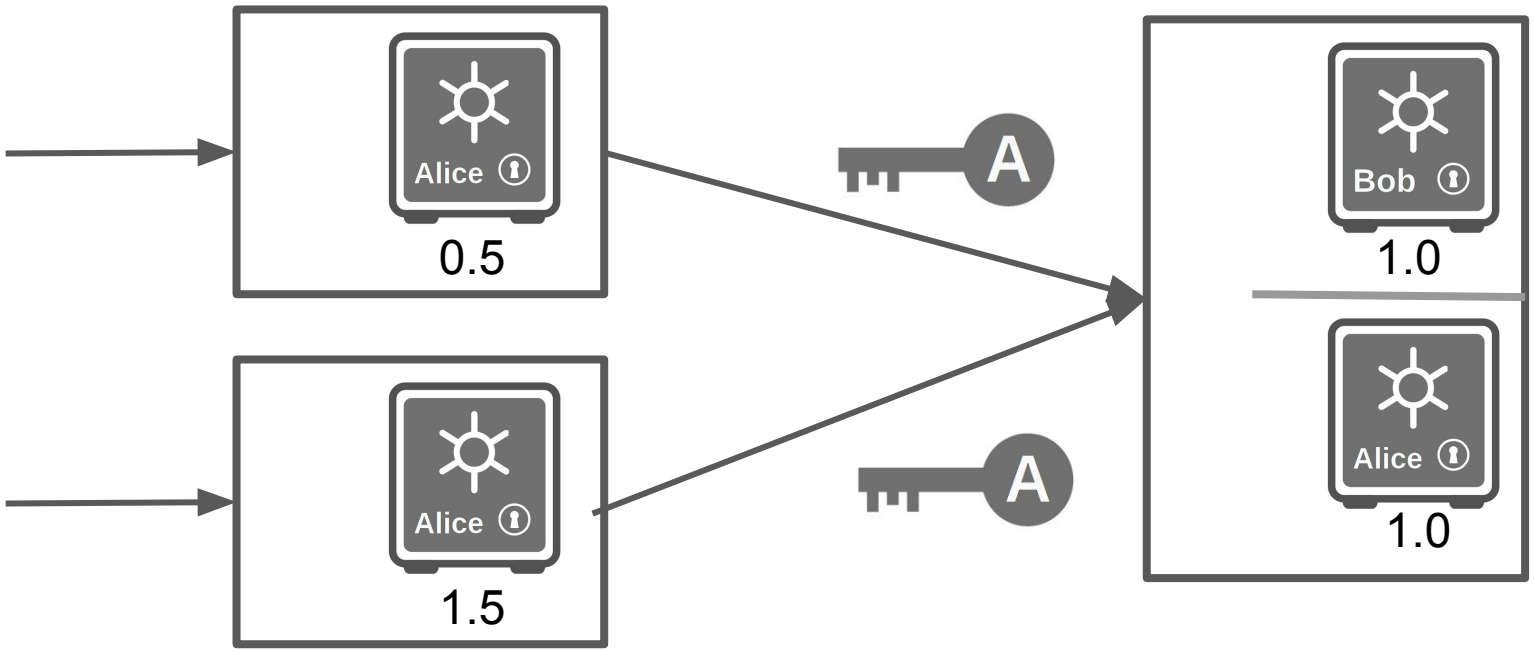
Transaction output: tuple of recipient and value
input: tuple of txid, vout and signature

Unspent Transaction Outputs (UTXOs)

- Alice owns 2 coins = Alice can spend transaction outputs whose values sum to 2



Spending Outputs



Part 3: Script

Cryptography Basics

- Cryptographic hash functions
  - hash: `{0,1}* -> {0,1}^n`
  - Example: `sha1("foo") = f1d2d2f924e986ac86fdf7b36c94bcdf32beec15`
  - collision resistant
- Public key cryptography
  - key pair: secret key `sk` and public key `pk`
  - cryptographic signature over message m
    - `sign(message, sk) -> sig`
    - `verify(message, pk, sig) -> {0, 1}`
  - Nobody can create a `sig` for a `pk` without the `sk`.

Script Evaluation: Pay-to-pubkey (P2PK)

= Bitcoin script `<pubKey> OP_CHECKSIG`

= Bitcoin script `<sig>`

+

= `<sig> <pubKey> OP_CHECKSIG`

= `true`

Multisig

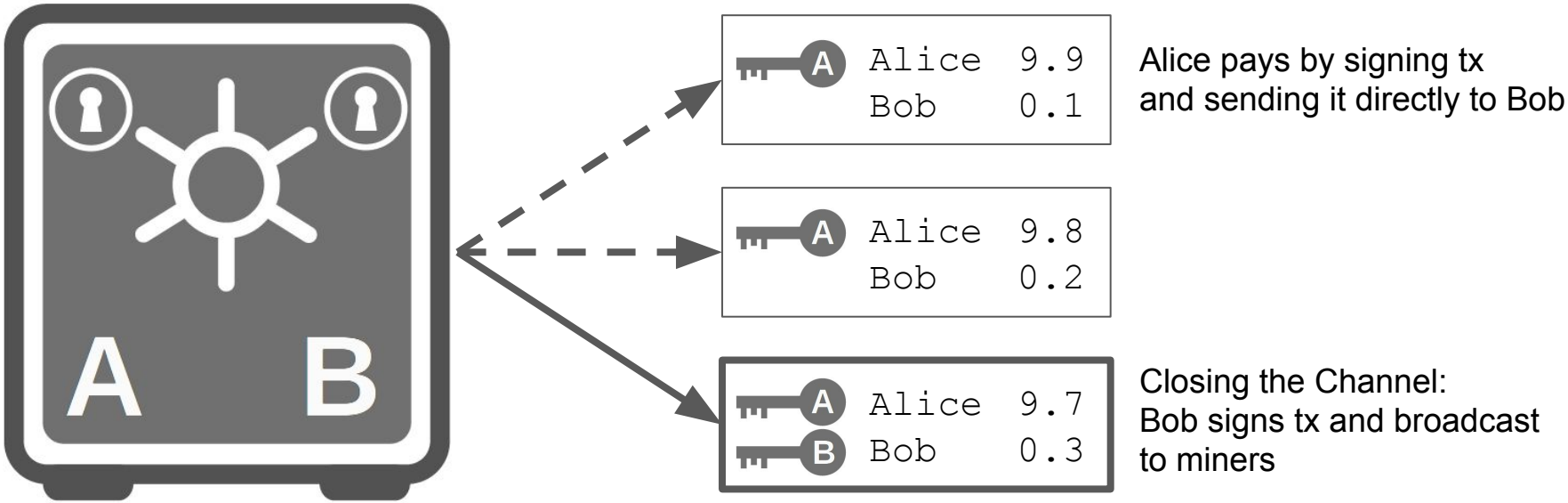
2 of 3 Multisig Output

Use cases: Wallet security, Escrow, Micropayment Channels

scriptPubKey: `<m> <pubKey_1> … <pubKey_n> <n> OP_CHECKMULTISIG`
scriptSig: `<sig_1> … <sig_m>`

Payment Channels

Setup: Alice creates transaction with 10 bitcoin to a 2-of-2 multisig with Bob



Micropayment Channel

- Problem: If Bob vanishes, Alice's coins are lost
- CheckLockTimeVerify
  - `12345 OP_CLTV`
  - script evaluation fails if blockchain < 12345 blocks
- Idea: After some time, Alice gets refund

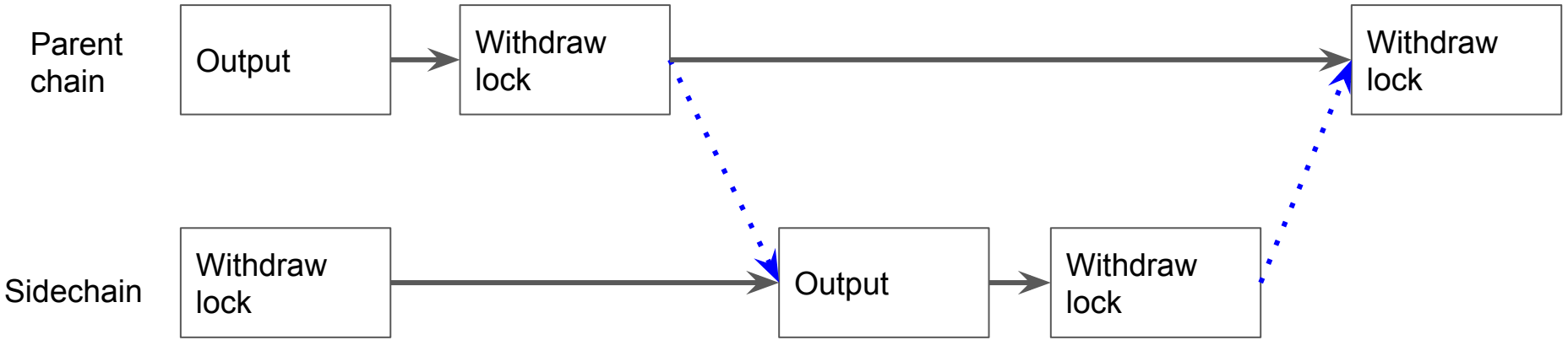
Part 4: Sidechains

Sidechains

- Observations
  a. There is no single blockchain that meets all requirements.
  b. Blockchains make different trade offs.
  c. New blockchain rules need consensus, slow process.
  d. Creating new blockchains from scratch is a huge challenge
     - Network effect, security
- Interoperability
  a. Pass information from chain to chain in a trustless and automated way.
  b. Leverage security from a different chain.
  c. Common API.

Sidechains

- Use case: Add features to Bitcoin



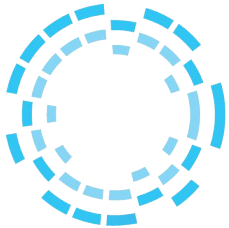
Federated Peg

- New security model: Set of mutually distrusting functionaries
- Enforce the rules that Bitcoin is currently unable to.
- Uses m-of-n multisig instead of PoW.
- Auditable
- Allows creation of interoperable private chains.

Elements

https://elementsproject.org / https://github.com/elementsproject/elements

- Bitcoin Core code fork
- Uses federated peg
  - our public chain pegged to Bitcoin testnet
- Alpha released, Beta soon

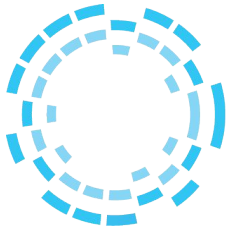


Blockstream

Liquid

https://elementsproject.org/sidechains/liquid/

- Production Bitcoin sidechain
- Based on elements
- Key feature: Decrease interchange settlement lag (ISL)
  - Because Liquid uses federated Peg: improves latency, throughput
- + Elements features (CT)
- Primarily for Bitcoin exchanges, payment processors, traders
- Strong Federation
- Launch in late summer 2016



Blockstream

Alpha feature: Confidential Transactions (CT)

Tx verification: `input_value = output_value + fee`

Verification with CT: `Enc(input_value) = Enc(output_value) + fee`

`alpha_address = bitcoin_address + blinding_pubKey`

Without corresponding blinding private key, values are hidden (blinded).

Auditors can import private blinding key

Part 5: Bitcoin Roadmap

“Through the use of cryptographic proof and decentralized
networks Bitcoin minimizes and replaces trust costs.”

Segregated Witness

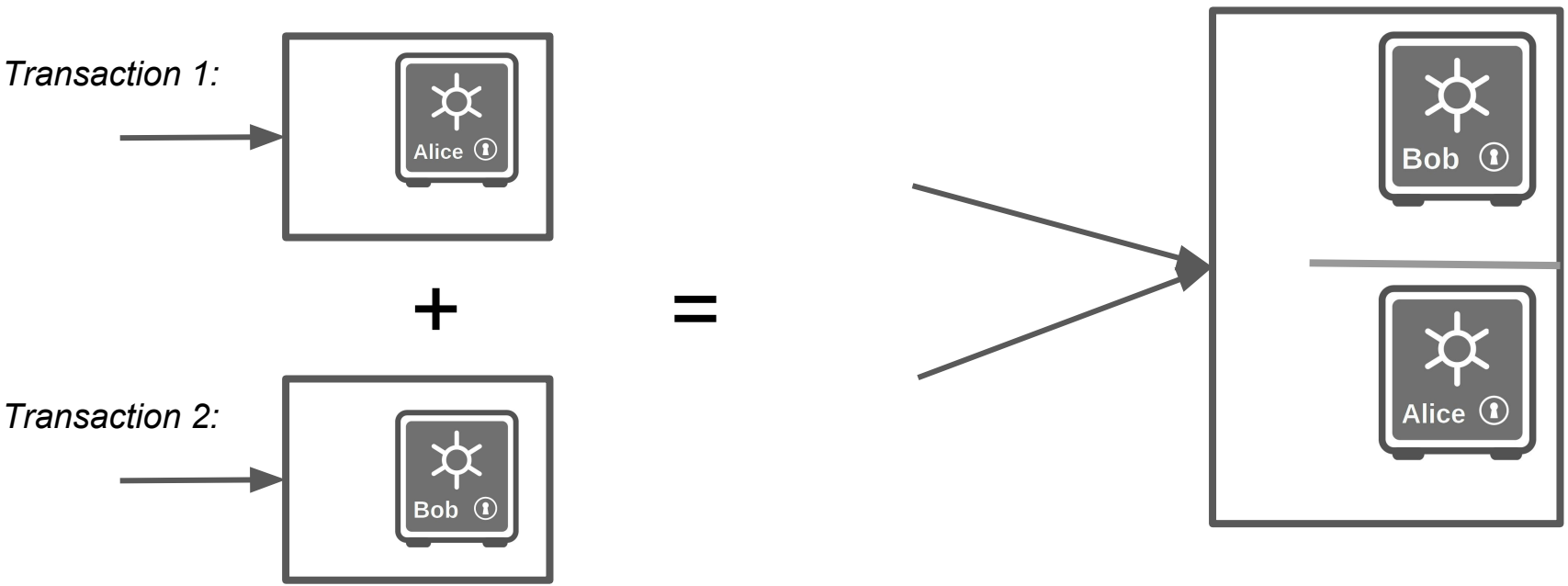
- Signaturen sind nicht mehr Teil der Transaktion
  - Sie sind nur ein Zeuge (Witness) den man zur Validierung braucht
- Soft-fork: Benutzer bestimmen selbst ueber Upgrade
- Implementiert, aber noch nicht aktiviert

- Loest “malleability” Problem
- Erhoeht den effektiven Transaktions-Durchsatz
- Skript Versionierung

Signature Aggregation

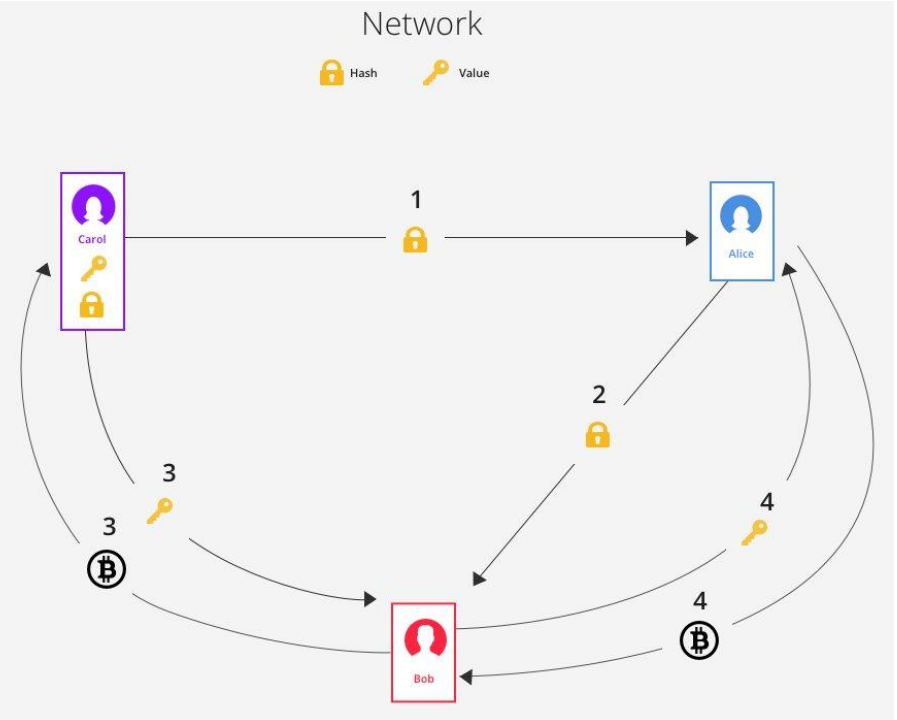
- Neue Skript Version
- Fuehrt neuen Signatur Algorithmus ein (Schnorr)
- Es wird nur eine Signature pro Transaktion benoetigt
- Im Durchschnitt bis zu 30% Einsparung der Transaktionsgroesse
  - Erhoeht Durchsatz
- Nebeneffekt: Man spart Transaktionsgebuehren (~ 5%) wenn man eine Transaktion gemeinsam mit einer anderen Person erstellt (“CoinJoin”)

CoinJoin



Lightning Network

- = Payment channel + Netzwerk
- Zahlung benoetigt im Allgemeinen nicht Eroeffnung eines Channels
- Sind Off-chain
- Privatsphaere durch Onion Routing



From: https://BitcoinMagazine.com

Allgemeinere “Smart Contracts”

- Ziele
  - On-Chain Verifikation, Off-Chain Berechnung
  - Basierend auf 80 Jahren Fortschritt in der Informatik
    - Und 40 Jahren Software Entwicklung
  - Privat
- Beispiele
  - MAST
    - Bitcoin Skript, aber es wird nur der ausgefuehrte Teil offenbart
  - ZKCP
    - Zk-snark: Beweis, dass Funktion auf Input true zurueckgibt ist, ohne den Input zu offenbaren
    - Beispiel: `is_valid_Sudoku_solution(solution)`
    - Bezahlung erfolgt nur genau dann wenn Beweis und Input offenbart wird
  - etc.

Zusammenfassung

- Blockchain: **Consensus** on shared **censorship-resistant state** with **immutable** rules in a **distributed** environment with potentially **dishonest** nodes.
- Bitcoin ist eine flexible Plattform fuer Blockchain Applikationen.
- Sidechains fuegen state-of-the-art Features hinzu und werden ab dem spaeten Sommer in Produktion gehen.
- Unzaehlige neue Features in Bitcoin sind in Entwicklung. Besondere Wertlegung auf Erhaltung der Dezentralisierung.
- Slides: https://nickler.ninja/slides/2016-Munster.pdf