Bitcoin and Blockchain for Pythoneers

EuroPython 2017

Why Bitcoin?

Crypto currency

- fast
- reliable
- without central authority

- The Blockchain is a distributed ledger (peer to peer).

Problems of a Distributed Database

All transactions are announced publicly.

- Simple approach: «Alice transfers a coin to Bob»
  - Problem: Transaction forgery
  - Solution: coins get an ID
- More elaborated approach: «Alice transfers coin #42 to Bob»
  - Problem: Synchronization issues, caused by network latency, e.g. double spending.
  - Solution: the whole networks verifies the transaction
- How to reach consensus?
  - Problem: fake identities, e.g. «Sybil» attack
  - Solution: Proof of Work (PoW)

Bitcoins Blockchain Principles

- Transactions are verified in a distributed manner and registered in blocks.
- Blocks form a chain.
- Blocks are created using a PoW.
- New blocks a created (mined) on the longest chain.

Block Header

1. Version number (fix value)
2. Time stamp (changes every second)
3. Difficulty, i.e. zero bits («fix» value)
4. Previous block's hash (fix value)
5. Root of Merkle tree → transactions (changes for every transaction processed)
6. Nonce (freely chosen)

- Header information is hashed
- Hash has to satisfy the difficutly

} **PoW**

Block Reward

Coinbase

- Incentive for block miners.
- Brings new coins into Bitcoin system.
- Halved every four years (210’000 blocks).

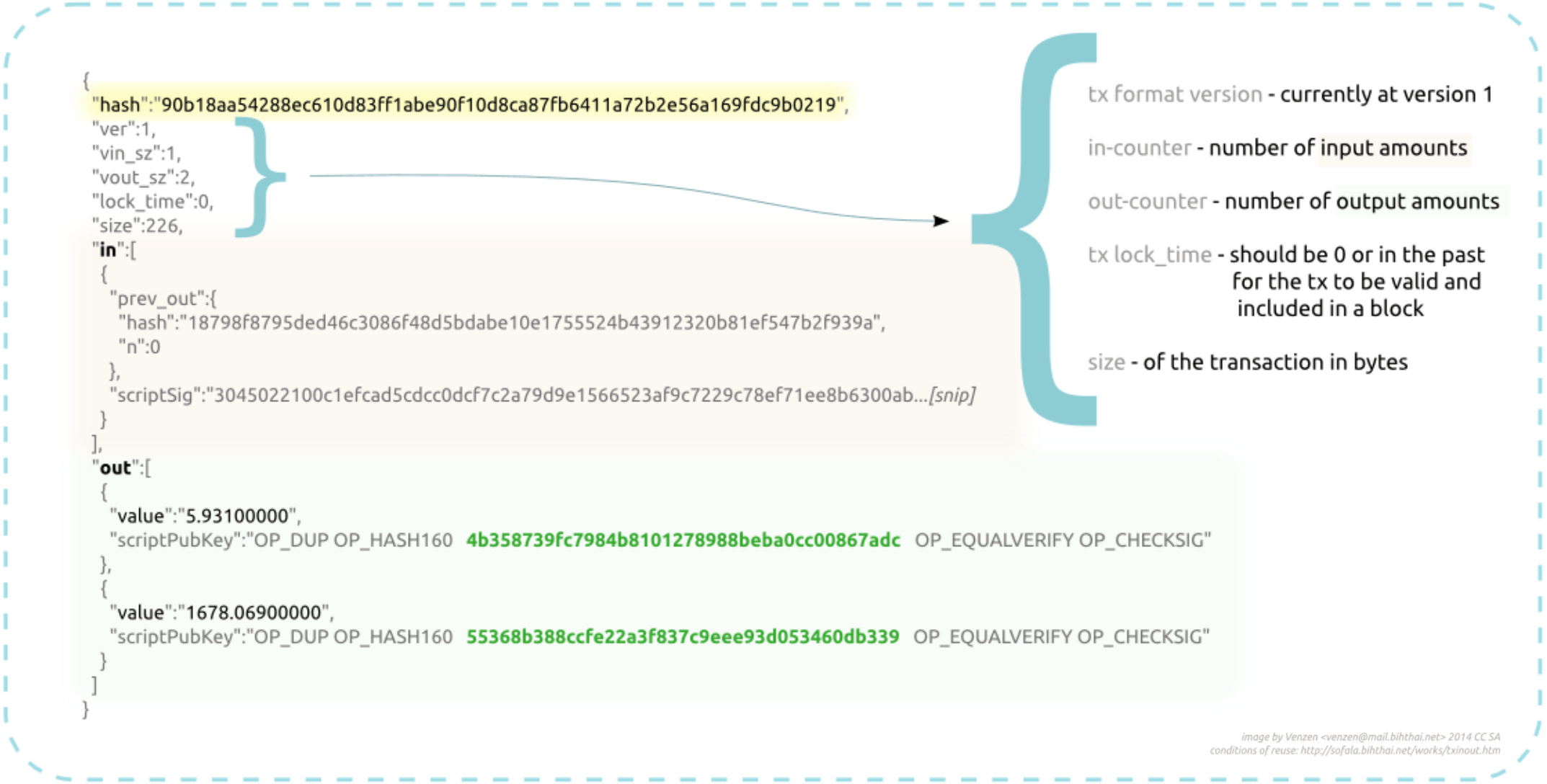
Transaction fees

Recapitulation

- Transactions are registered in blocks, the pages of the distributed ledger.
- Blocks are formed to a chain to give a complete order of the transaction history.
- Blocks are created by solving a cryptographic puzzle, the PoW, to prevent Sybil attacks.
- New blocks are mined on the longest chain to prevent forks.
- The miner of the new block is rewarded by a certain amount of Bitcoins.

See animation on https://www.youtube.com/watch?v=rSL5eSv31ew

Transactions: Structure



Transaction: Axioms

1. Any Bitcoin amount is sent to an address.
2. Any Bitcoin amount received is locked to the receiving address (and associated with a wallet).
3. Any time we spend Bitcoin, the amount we spend will always come from funds previously received and currently present in our wallet.
4. Addresses receive Bitcoin, but they do not send Bitcoin – Bitcoin is sent from a wallet.

Transactions: Example

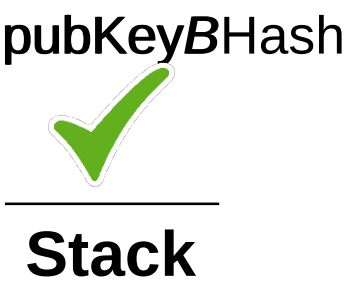
Output:	amount: 0.5 *scriptPubKey* / pubKey*B*Hash	Tx 1: Alice sends Bob 0.5 BTC
Input:	prevTxHash / index *scriptSig*	Tx 2: Bob sends Charlie 0.4 BTC
Output1:	amount: 0.4 *scriptPubKey* / pubKey*C*Hash	
Output2: (change)	amount: 0.1 *scriptPubKey* / pubKey*B*Hash	

Transaction Verification

Combined script (Alice sends Bob 0.5 BTC / Bob spends 0.4 BTC):

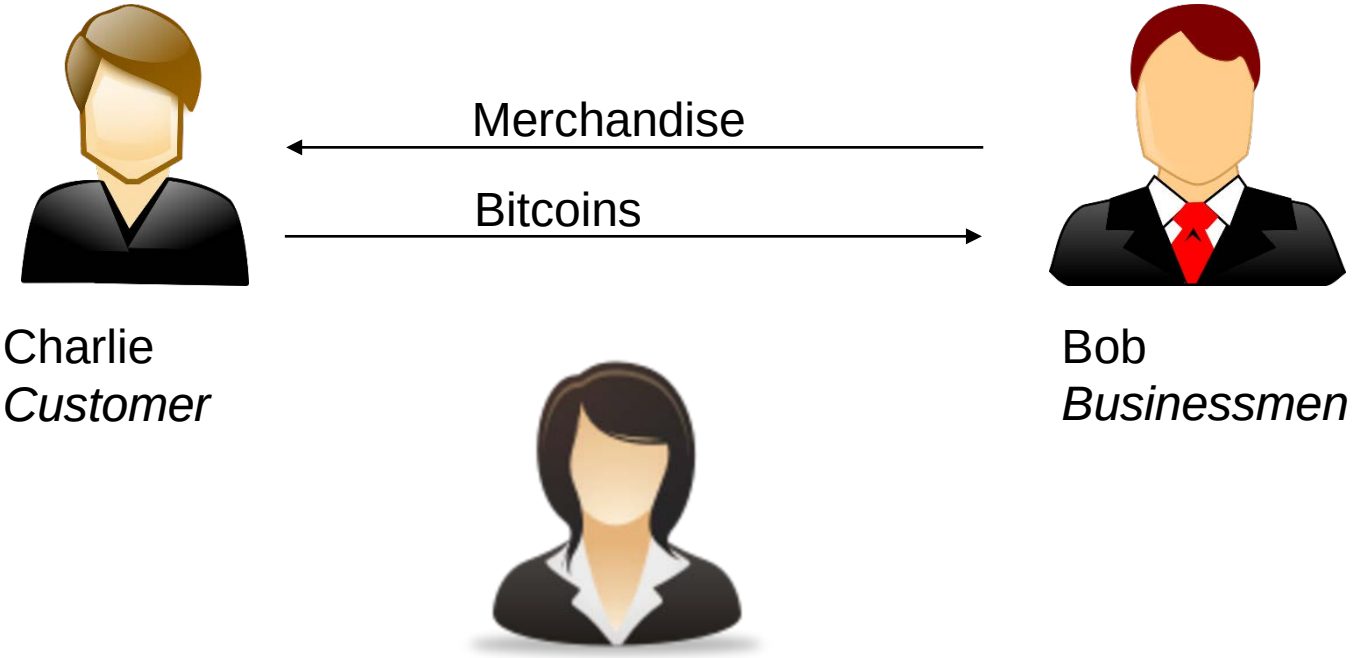
sig pubKey*B* `OP_DUP OP_HASH160` pubKey*B*Hash `OP_EQUALVERIFY OP_CHECKSIG`

Pay-to-Pubkey-Hash (P2PKH)



Transaction: Contracts

Escrow And Arbitration:



Multi-signature contract:
Amount can only be spent, if two people sign.

Nakamoto (2008): Abstract

“A purely *peer-to-peer* version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based *proof-of-work*, forming a *record that cannot be changed* without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.”

Problems

- Low transaction volume
- High energy consumption (for block mining)

Alternative consensus protocols:

- Proof of Stake (PoS)
- Practical byzantine fault tolerance (PBFT)

Alternatives

- Bitcoin forks: Litecoin, Dogecoin
- Alternative Blockchains: Ethereum
- Alternative Consensus protocol: Hyperledger

Comparison:

	Bitcoin	Litecoin	Dogecoin	Ethereum	Hyperledger (IBM)
Block interval	10 min	2.5 min	1 min	10-20 seconds	
Public nodes	6000	800	600	4000	
Consensus protocol	PoW	PoW	PoW	PoW (PoS planned for 2018)	PBFT

2016

Blockchain Use Cases

- Crypto Currencies
- Smart Contracts
- Infrastructure for Digital IDs
- Distributed Databases
- Distributed storage of legal paperwork


C&A

Contact information and credits

Benno Luthiger
benno.luthiger@id.ethz.ch

ETH Zurich
IT Services
Stampfenbachstrasse 69
8092 Zürich

Images: Venzen / *venzen@mail.bihthai.net* (slide 8)