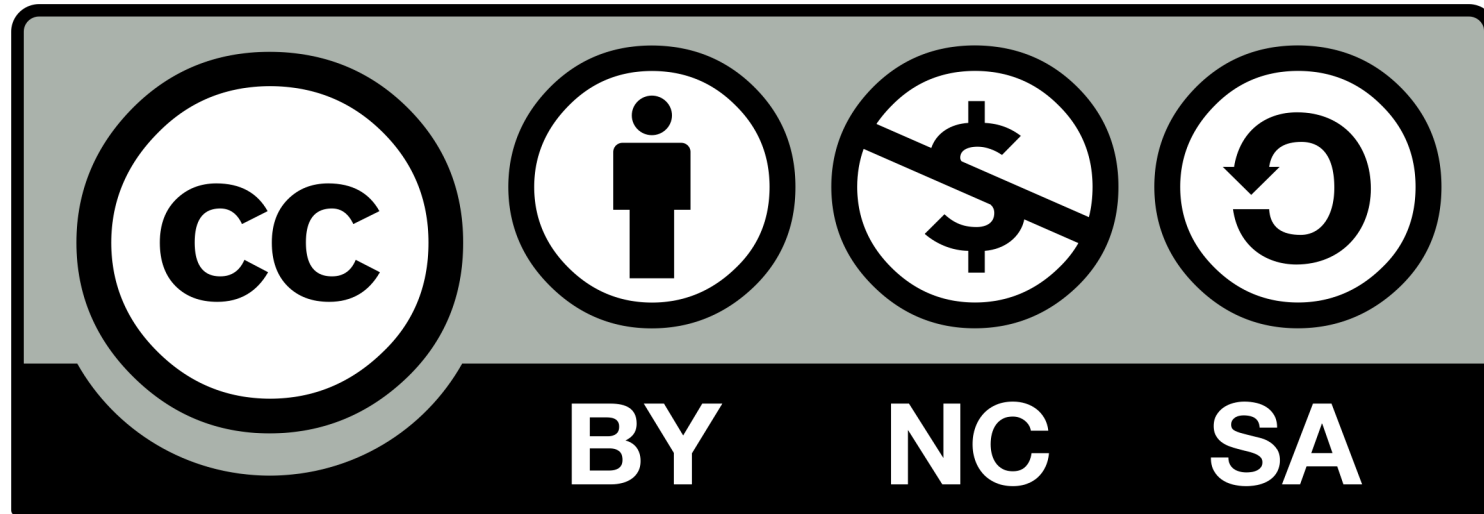# Technical Introduction To Bitcoin
DISCOVERING THE MAGIC INTERNET MONEY

Stéphane Roche

02-06-2019

CREATIVE COMMONS

Attribution-ShareAlike 4.0 International (CC BY-SA 4.0)

ABOUT STEPHANE

2015

Start working on Bitcoin in 2015 at Ledger (hardware wallet)

2017–2019

Focus on blockchain technical trainings

Founder of D10eConsulting

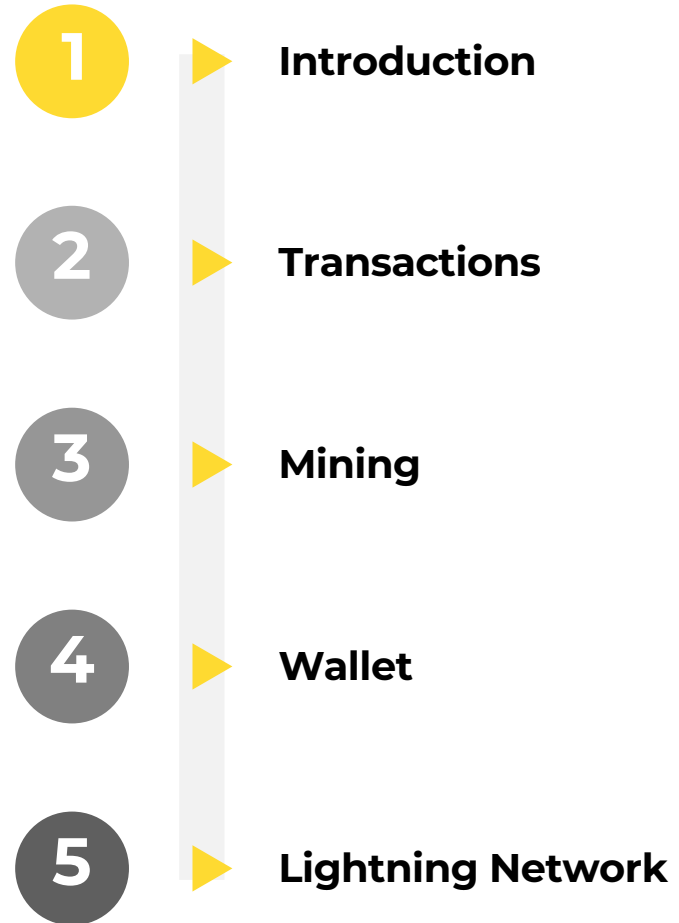Consultant at Chainsmiths

Work on Ethereum in 2016-2017

- Co-found non-profit organization Asseth
- R&D on Dao1901
- Contribute to the ERC20 Consensys smart contracts
- Dether.io (15,000 ETH raised)

2016–2017

OUTLINE

1

INTRODUCTION

"Bitcoin is the culmination of 30 years of attempts at building digital money for the internet."

Bitcoin Prehistory Timeline

Whitfield Diffie and Martin Hellman, "New Directions in Cryptography" (1976)

David Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms" (1981)

David Chaum, "Blind Signatures for Untraceable Payments" (1983)

Satoshi Nakamoto, "Bitcoin: Peer-to-peer Electronic Cash System" (2008)

Timothy C. May, "The Crypto-Anarchist's Manifesto" (1988)

Cypherpunks founded in SF (1992)
by Eric Hughes, Timothy C. May and John Gilmore

Eric Hughes, "A Cypherpunk's Manifesto" (1993)

Timothy C. May, "The Cyphernomicon" (1994)

1976 | 1981 | 1983 | 1988 | 1989 | 1992 | 1993 | 1994 | 1996 | 1997 | 1998 | 1999 | 2001 | 2006 | 2008

CyberCash (1994)

E-gold (1996)

David Chaum, Founded Digicash (1989)

Liberty Reserve (2006)

Adam Back, HashCash, proof-of-work for email spam (1997)

Nick Szabo, "Formalizing and Securing Relationships on Public Networks" (1997)

Video game currencies and markets (era started in 2001)

Nick Szabo, "Securing Property Titles with Owner Authority" (1998) – Smart Contracts

Many online retailers in the dotcom bubble (Beenz, Flooz, etc) (1998-2001)

Wei Dei, "B-money" (1998) – decentralized database to record txs and using a type of proof-of-work
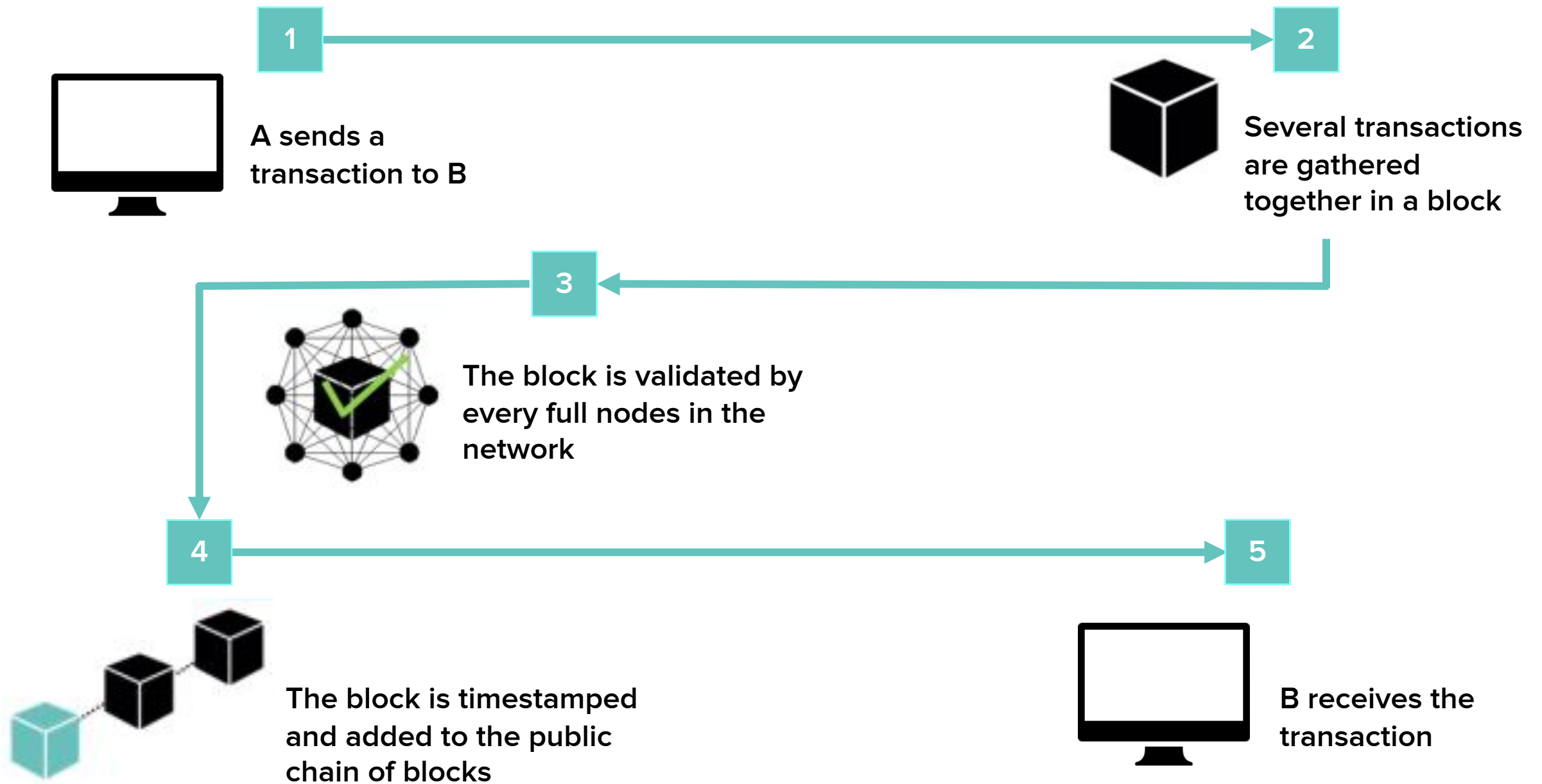
FEATURES

- A blockchain serves as the authoritative, trusted, open, public, global ledger

- Permissionless
  - Anyone can transact freely
  - Anyone can contribute to the protocol development (open source)

- Censorship-resistant

- Trustless
  - No third-party
  - An alternative to the global banking cartel
  - But securing digital assets is not trivial

- **A blockchain serves as the authoritative, trusted, open, public, global ledger**

- **Permissionless**
  - **Anyone can transact freely**
  - **Anyone can contribute to the protocol development (open source)**

- **Censorship-resistant**

- **Trustless**
  - **No third-party**
  - **An alternative to the global banking cartel**
  - **But securing digital assets is not trivial**

- **Decentralized network of independent nodes**
  - Nodes following the same set of rules (consensus rules)
  - Validating each blocks and transactions
  - Assembling their own copy of the blockchain

- **Apocalypse-resistant**
  - We can send bitcoins via satellite, mesh network, radio waves (Ham, JS8Call)

- **Deflationary system**
  - Programmatically fixed coin issuance (inflation rate)
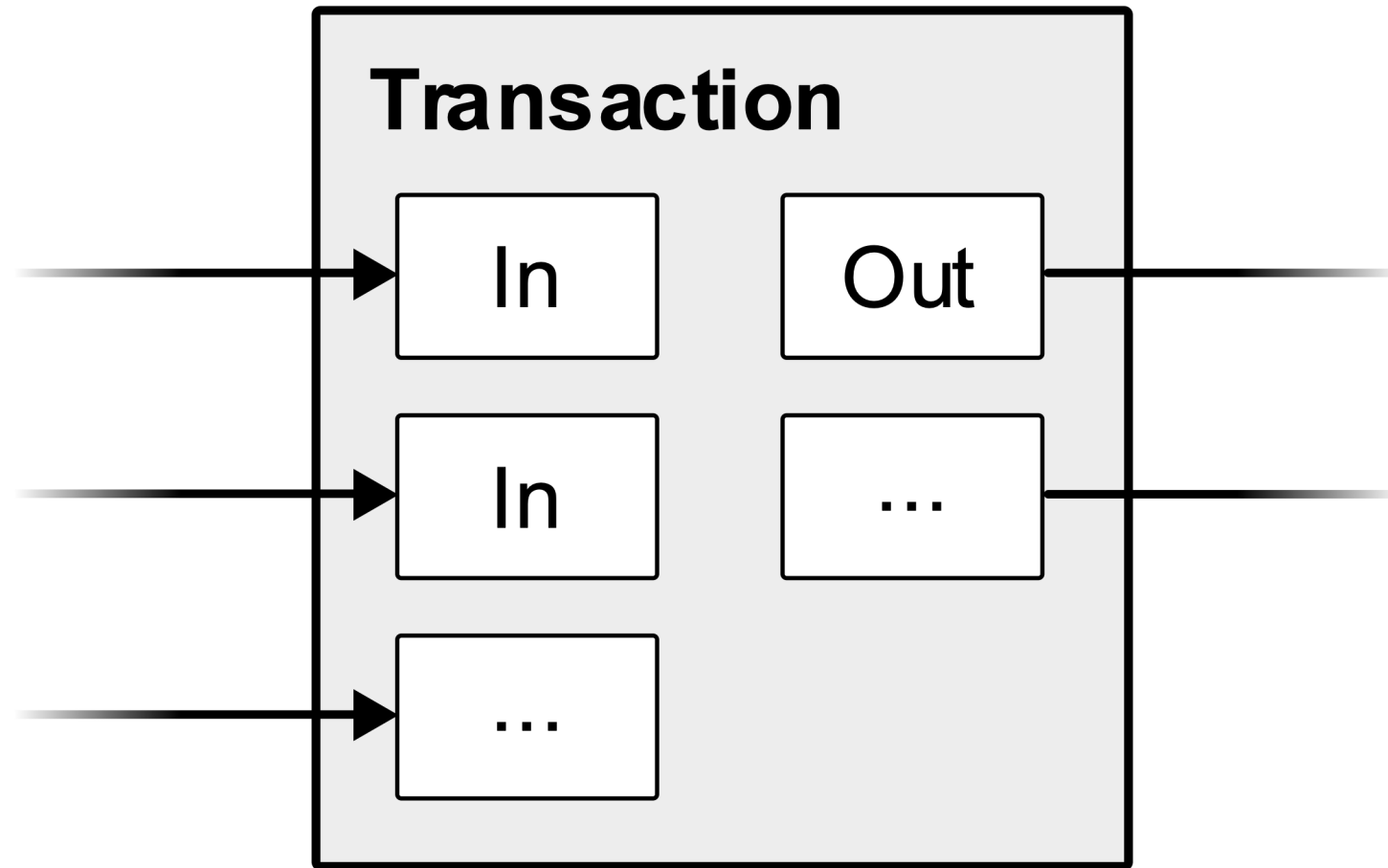 - Block mining reward halves every 210,000 blocks
  - Limited money supply to 21 milions coins

PROTOCOL & IMPLEMENTATIONS

- BIPs are a set of specifications that affects Bitcoin in general, not a specific implementation

- But ultimately, the reference client is the specification
  - The Bitcoin protocol is specified by the behavior of the reference client
  - Consensus is determined by the software that the majority of the network runs
  - Consensus is not determined by a natural language specification

- Implementations
  - Bitcoin Core / Reference client / Satoshi client (C++)
  - Btcd (Go)
  - BitcoinJ (Java)
  - Bcoin (Javascript)
  - Parity-bitcoin (Rust)
  - …

1 A sends a transaction to B

2 Several transactions are gathered together in a block

3 The block is validated by every full nodes in the network

4 The block is timestamped and added to the public chain of blocks

5 B receives the transaction

2 TRANSACTIONS

SMART CONTRACTS

- Unlocking script in Input / Locking script in previous Output

- Several standard smart contracts, defining the output type
  - Legacy
    - Pay To Public Key (P2PK)
    - Pay To Public Key Hash (P2PKH)
    - Pay To Script Hash (P2SH)
      - Nested Segwit
        - P2SH-P2WPKH
        - P2SH-P2WSH
  - Native Segwit
    - Pay To Witness Public Key Hash (P2WPKH)
    - Pay To Witness Script Hash (P2WSH)
  - Data anchoring (null_data, OP_RETURN)

- The smart contract type defines the execution flow, using template code

SPENDING CONDITIONS

- One signature

- Multiple signatures
    - A group of signers
    - A wallet company counter-signs a user tx

- Hashlock

- Timelock

- A mix of all that

- **MiniScript**
  - A new language to describes spending policies
  - Tree-like structure
  - Easy to construct
  - Easy to analyse

```
// push value
OP_0 = 0x00,
OP_FALSE = OP_0,
OP_PUSHDATA1 = 0x4c,
OP_PUSHDATA2 = 0x4d,
OP_PUSHDATA4 = 0x4e,
OP_1NEGATE = 0x4f,
OP_RESERVED = 0x50,
OP_1 = 0x51,
OP_TRUE=OP_1,
OP_2 = 0x52,
OP_3 = 0x53,
OP_4 = 0x54,
OP_5 = 0x55,
OP_6 = 0x56,
OP_7 = 0x57,
OP_8 = 0x58,
OP_9 = 0x59,
OP_10 = 0x5a,
OP_11 = 0x5b,
OP_12 = 0x5c,
OP_13 = 0x5d,
OP_14 = 0x5e,
OP_15 = 0x5f,
OP_16 = 0x60,

// control
OP_NOP = 0x61,
OP_VER = 0x62,
OP_IF = 0x63,
OP_NOTIF = 0x64,
OP_VERIF = 0x65,
OP_VERNOTIF = 0x66,
OP_ELSE = 0x67,
OP_ENDIF = 0x68,
OP_VERIFY = 0x69,
OP_RETURN = 0x6a,

// stack ops
OP_TOALTSTACK = 0x6b,
OP_FROMALTSTACK = 0x6c,
OP_2DROP = 0x6d,
OP_2DUP = 0x6e,
OP_3DUP = 0x6f,
OP_2OVER = 0x70,
OP_2ROT = 0x71,
OP_2SWAP = 0x72,
OP_IFDUP = 0x73,
OP_DEPTH = 0x74,
OP_DROP = 0x75,
OP_DUP = 0x76,
OP_NIP = 0x77,
OP_OVER = 0x78,
OP_PICK = 0x79,
OP_ROLL = 0x7a,
OP_ROT = 0x7b,
OP_SWAP = 0x7c,
OP_TUCK = 0x7d,

// splice ops
~~OP_CAT = 0x7e,~~
~~OP_SUBSTR = 0x7f,~~
~~OP_LEFT = 0x80,~~
~~OP_RIGHT = 0x81,~~
OP_SIZE = 0x82,

// bit logic
~~OP_INVERT = 0x83,~~
~~OP_AND = 0x84,~~
~~OP_OR = 0x85,~~
~~OP_XOR = 0x86,~~
OP_EQUAL = 0x87,
OP_EQUALVERIFY = 0x88,
OP_RESERVED1 = 0x89,
OP_RESERVED2 = 0x8a,

// numeric
OP_1ADD = 0x8b,
OP_1SUB = 0x8c,
~~OP_2MUL = 0x8d,~~
~~OP_2DIV = 0x8e,~~
OP_NEGATE = 0x8f,
OP_ABS = 0x90,
OP_NOT = 0x91,
OP_0NOTEQUAL = 0x92,

OP_ADD = 0x93,
OP_SUB = 0x94,
~~OP_MUL = 0x95,~~
~~OP_DIV = 0x96,~~
~~OP_MOD = 0x97,~~
~~OP_LSHIFT = 0x98,~~
~~OP_RSHIFT = 0x99,~~

OP_BOOLAND = 0x9a,
OP_BOOLOR = 0x9b,
OP_NUMEQUAL = 0x9c,
OP_NUMEQUALVERIFY = 0x9d,
OP_NUMNOTEQUAL = 0x9e,
OP_LESSTHAN = 0x9f,
OP_GREATERTHAN = 0xa0,
OP_LESSTHANOREQUAL = 0xa1,
OP_GREATERTHANOREQUAL = 0xa2,
OP_MIN = 0xa3,
OP_MAX = 0xa4,

OP_WITHIN = 0xa5,

// crypto
OP_RIPEMD160 = 0xa6,
OP_SHA1 = 0xa7,
OP_SHA256 = 0xa8,
OP_HASH160 = 0xa9,
OP_HASH256 = 0xaa,
OP_CODESEPARATOR = 0xab,
OP_CHECKSIG = 0xac,
OP_CHECKSIGVERIFY = 0xad,
OP_CHECKMULTISIG = 0xae,
OP_CHECKMULTISIGVERIFY = 0xaf,

// expansion
OP_NOP1 = 0xb0,
OP_CHECKLOCKTIMEVERIFY = 0xb1,
OP_NOP2 = OP_CHECKLOCKTIMEVERIFY,
OP_CHECKSEQUENCEVERIFY = 0xb2,
OP_NOP3 = OP_CHECKSEQUENCEVERIFY,
OP_NOP4 = 0xb3,
OP_NOP5 = 0xb4,
OP_NOP6 = 0xb5,
OP_NOP7 = 0xb6,
OP_NOP8 = 0xb7,
OP_NOP9 = 0xb8,
OP_NOP10 = 0xb9,

OP_INVALIDOPCODE = 0xff,
```
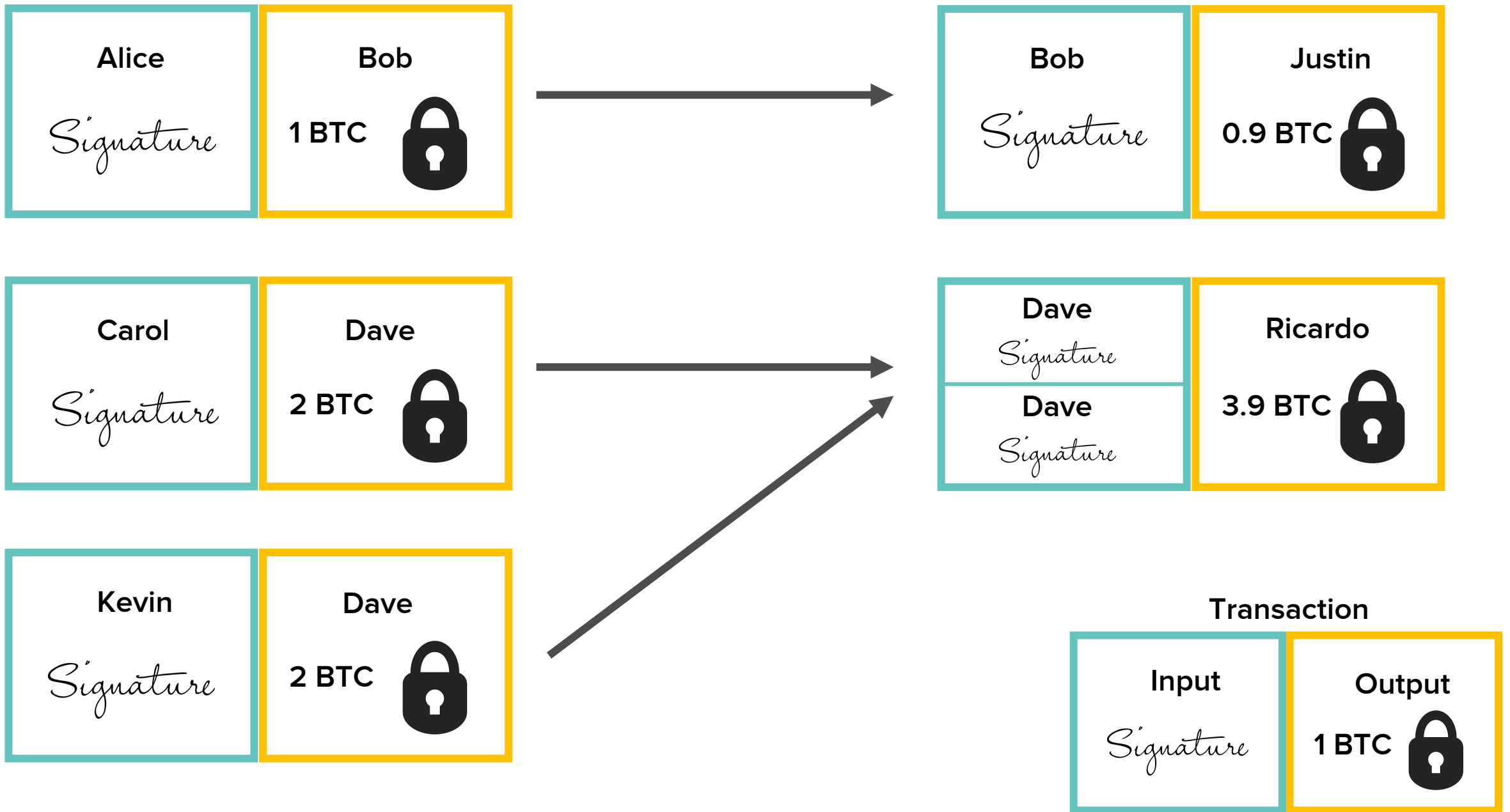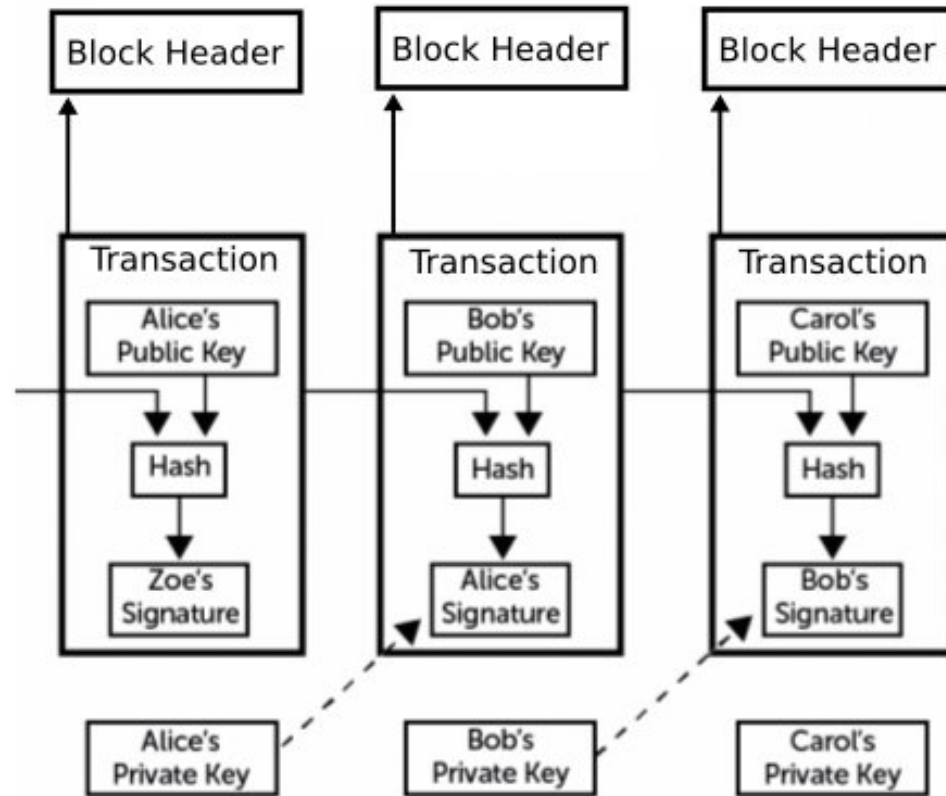
SCHNORR-BASED CONTRACTS

- Plan to transition from ECDSA to Schnorr

- Schnorr cryptography allows more mathematical tricks
  - We can add and substract signatures, public keys, etc.
- Scriptless scripts
  - A way to do alchemy with signatures
  - Smart contracts executed off-chain, only by the parties involved
  - A valid transaction has a signature that proves correct contract execution
  - Atomic coinswap, etc.
- Discreet log contracts
  - A way to do alchemy with public keys
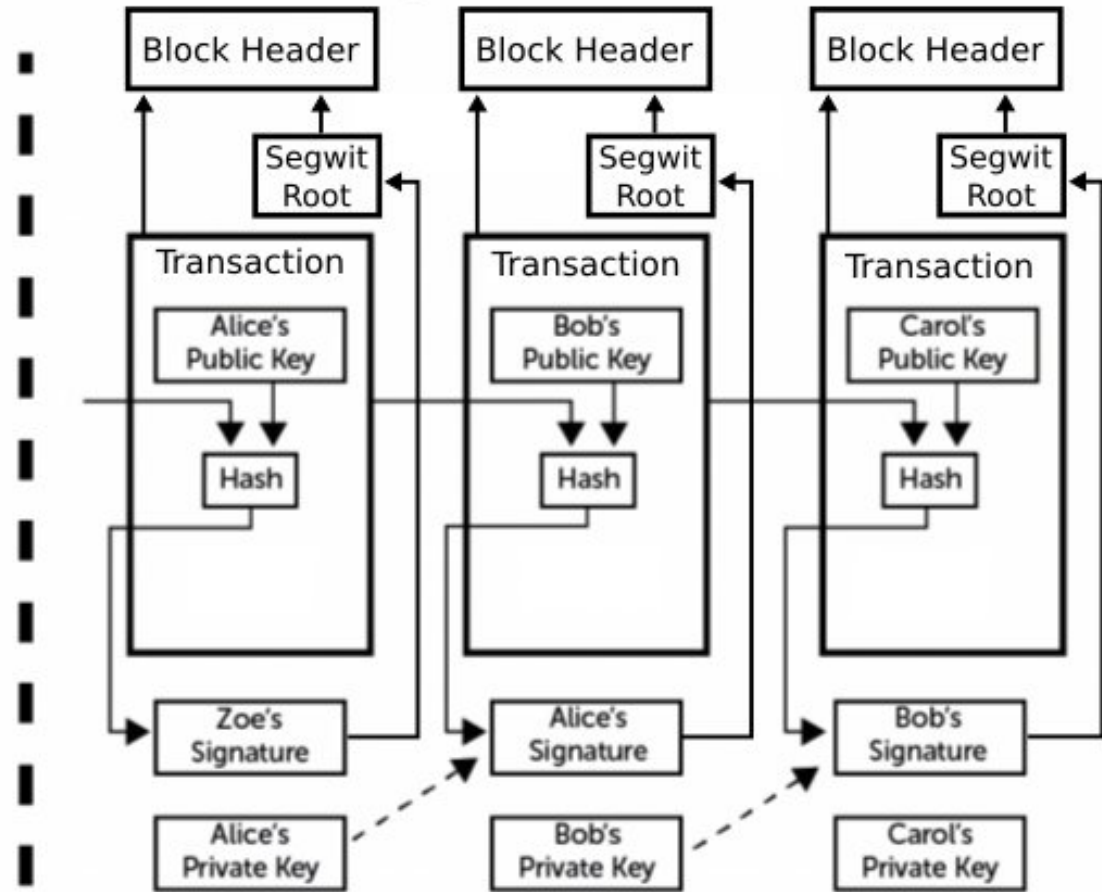 - An oracle determines division of funds
- Taproot

SEGREGATED WITNESS

- New transaction format, activated on August 2017

- Fix the transaction ID malleability problem
  - Malleable witness data (input script) segregated outside the transaction
  - Which allows the Lightning Network

- Bypass block size limit (1MB)
  - Witness data are discounted (1/4 of its real size)
  - Average block size is 1.3MB in Feb 2019

- More efficient signature verification algorithm

- Introduces script versioning

Non-segwit blocks

| Block Header | Block Header | Block Header |

Transaction
- Alice's Public Key
- Hash
- Zoe's Signature

Transaction
- Bob's Public Key
- Hash
- Alice's Signature

Transaction
- Carol's Public Key
- Hash
- Bob's Signature

Alice's Private Key

Bob's Private Key

Carol's Private Key

Segwit blocks

| Block Header | Block Header | Block Header |

Segwit Root

Segwit Root

Segwit Root

Transaction
- Alice's Public Key
- Hash

Transaction
- Bob's Public Key
- Hash

Transaction
- Carol's Public Key
- Hash

Zoe's Signature

Alice's Signature

Bob's Signature

Alice's Private Key

Bob's Private Key

Carol's Private Key

3 MINING

Mining plays 2 major roles in maintaining the Bitcoin network

Confirming Transactions

Issuing New coins

And the people who employ their computing resources to this process are called miners

BLOCKS

Bloc 46	Bloc 47	Bloc 48	Bloc 49
Transaction 92 Transaction 93 Transaction 94 Transaction 95	Transaction 96 Transaction 97 Transaction 98	Transaction 99 Transaction 100 Transaction 101	Transaction 102 Transaction 103 Transaction 104 Transaction 105
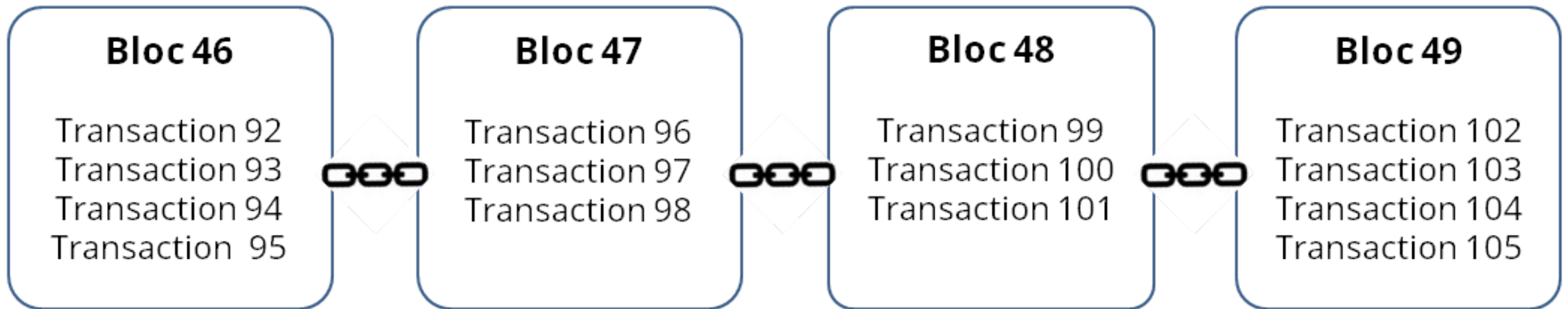
The first transaction of each block pays the miner who mined the block (coinbase transaction)
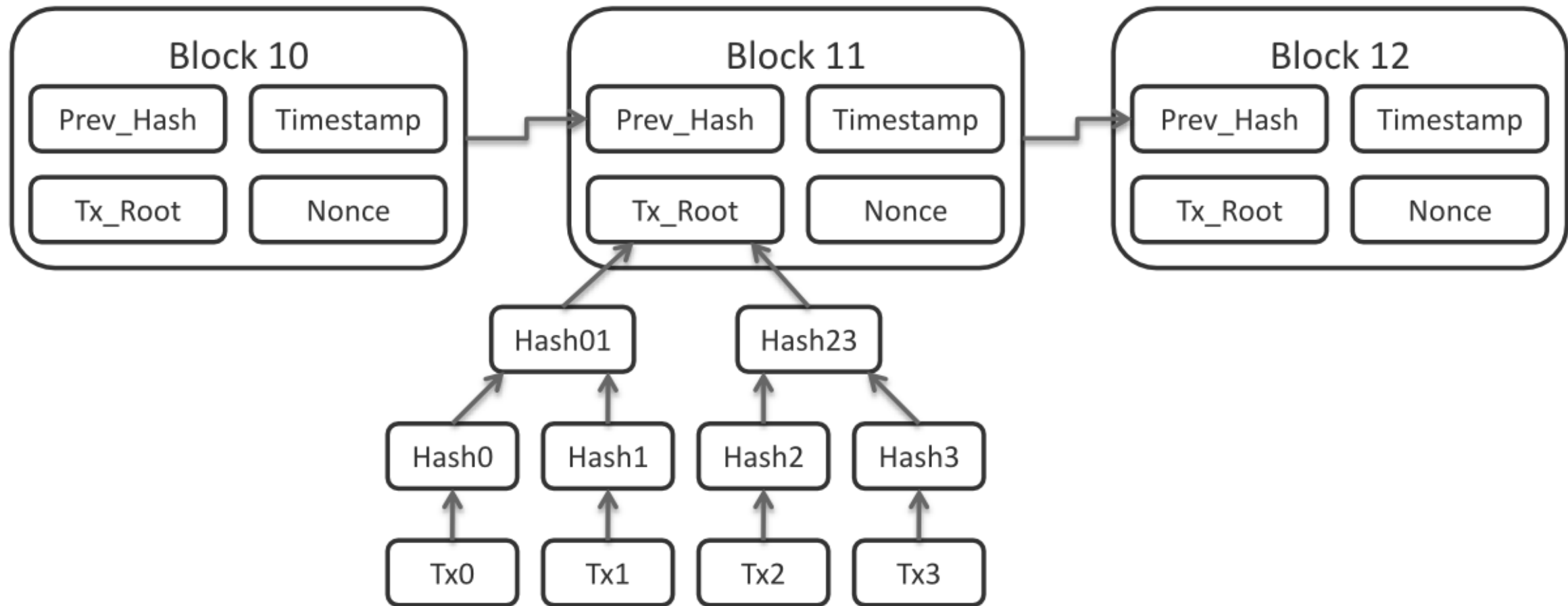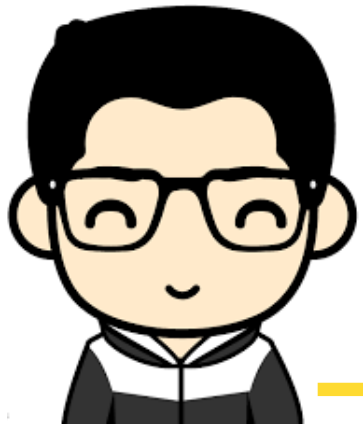
Memory Pool

- Transaction #34
- Transaction #a542
- Transaction #1f56
- Transaction #38b4
- Transaction #855c
- ...

Transaction 86

Candidate Block

Block Header

Block hash (PoW)
Previous block hash
Timestamp
Difficulty
Nonce (1++)
Transaction Merkle root

Transaction list
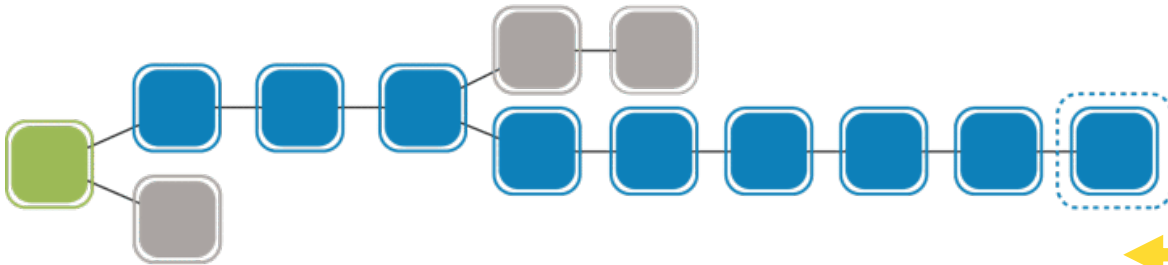
Coinbase transaction
Transaction #34
Transaction #a542
Transaction #1f56
Transaction #38b4
Transaction #855c
...

If block is valid
and block hash < difficulty target
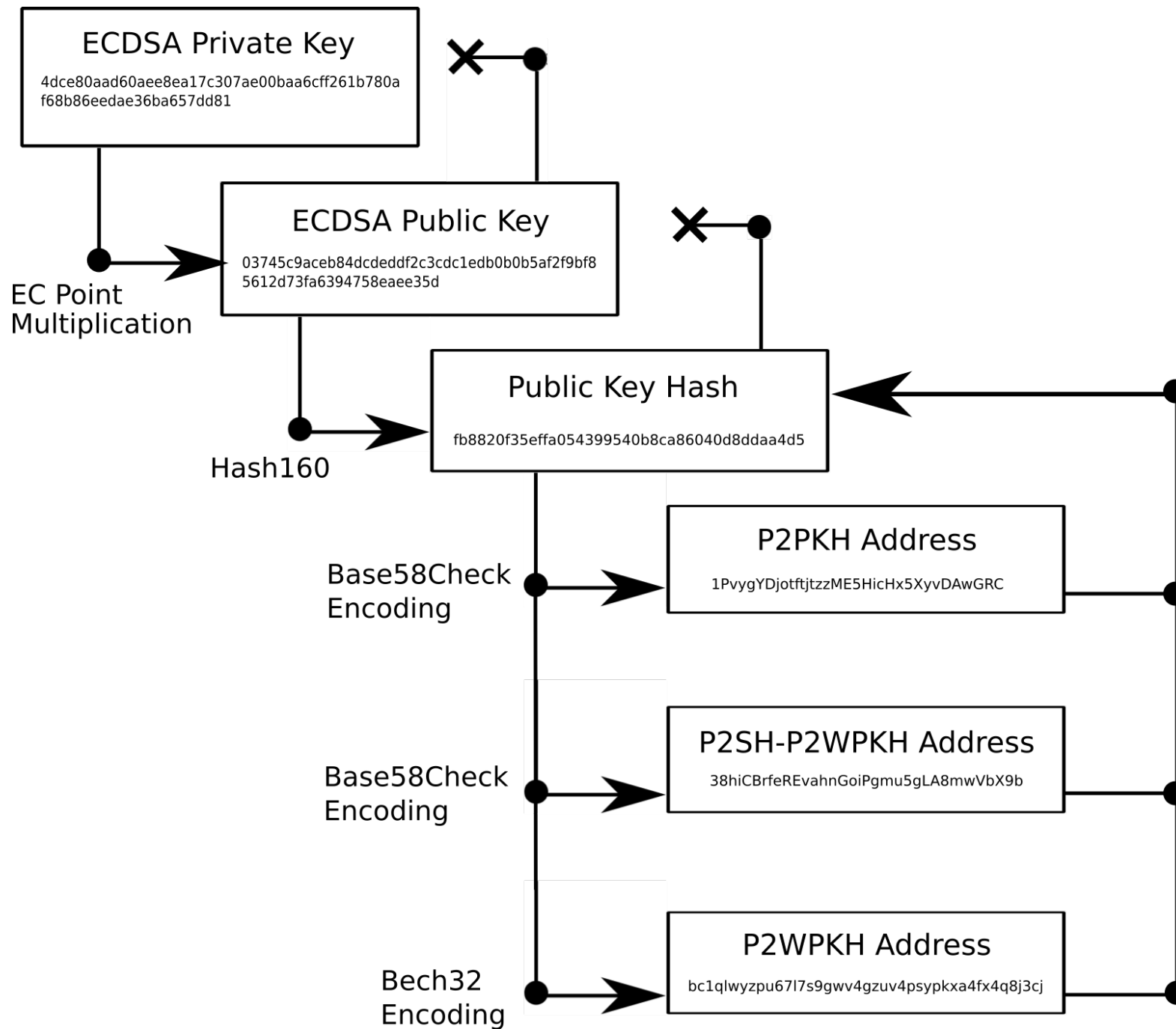then block is accepted by the other nodes
and added to their blockchain

Difficulty adjusts every
2016 blocks

- Use the Proof of Work consensus algorithm
  - Security based on laws of thermodynamics
  - Around 45 EH/s in March 2019 (exa = 10^{18})
  - Enables network-wide consensus without central authority

- Relentless race to the lowest electricity costs

- Mainly powered on renewable energy
  - CoinShares studies
  - More than 75% of Bitcoin's energy usage is estimated to come from renewable resources
  - Nearly half of all mining is done in a part of China where power is almost exclusively hydroelectric
  - Can positively contribute to the development and scaling of renewable energy projects
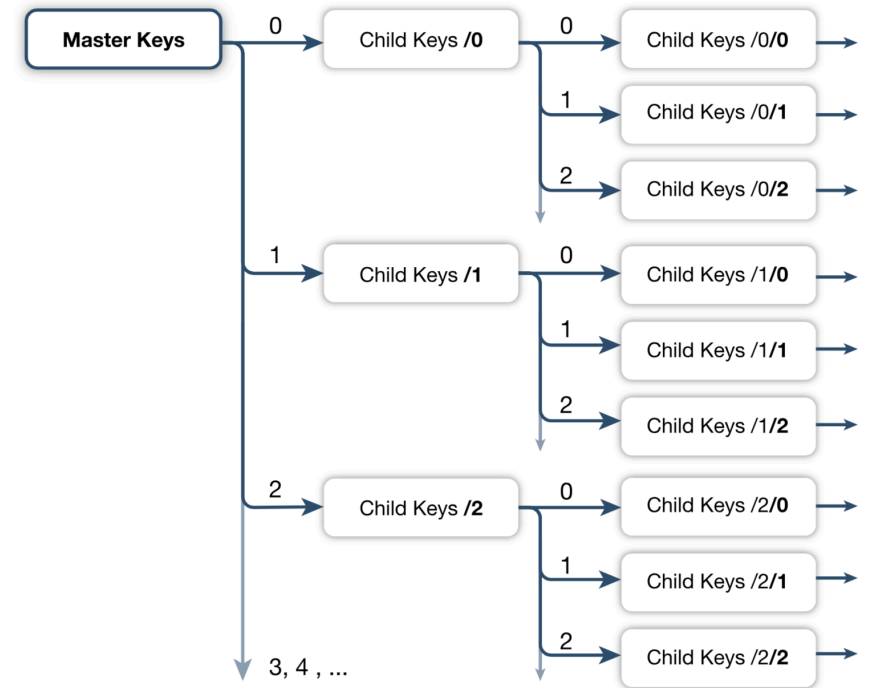
4 WALLET

ECDSA Private Key

4dce80aad60aee8ea17c307ae00baa6cff261b780a
f68b86eedae36ba657dd81

ECDSA Public Key

03745c9aceb84dcdeddf2c3cdc1edb0b0b5af2f9bf8
5612d73fa6394758eaee35d

EC Point
Multiplication

Public Key Hash

fb8820f35effa054399540b8ca86040d8ddaa4d5

Hash160

Base58Check
Encoding

P2PKH Address

1PvygYDjotftjtzzME5HicHx5XyvDAwGRC

Base58Check
Encoding

P2SH-P2WPKH Address

38hiCBrfeREvahnGoiPgmu5gLA8mwVbX9b

Bech32
Encoding

P2WPKH Address

bc1qlwyzpu67l7s9gwv4gzuv4psypkxa4fx4q8j3cj

HD WALLET

- Deterministically derive an indefinite number of addresses from a single secret

- Fresh addresses to improved privacy

- Can be reconstructed from master keys and mnemonic code

- Allows multiple accounts and multiple currencies

MNEMONIC CODE

- Wallet backup, usually 12 or 24 english words

- HD Tree derived from the mnemonic

- Mnemonic standard (BIP39) allows to add an additional passphrase

- No need to backup multiple times

- Bitcoin Core doesn't support mnemonic code
  - wallet.dat file to backup
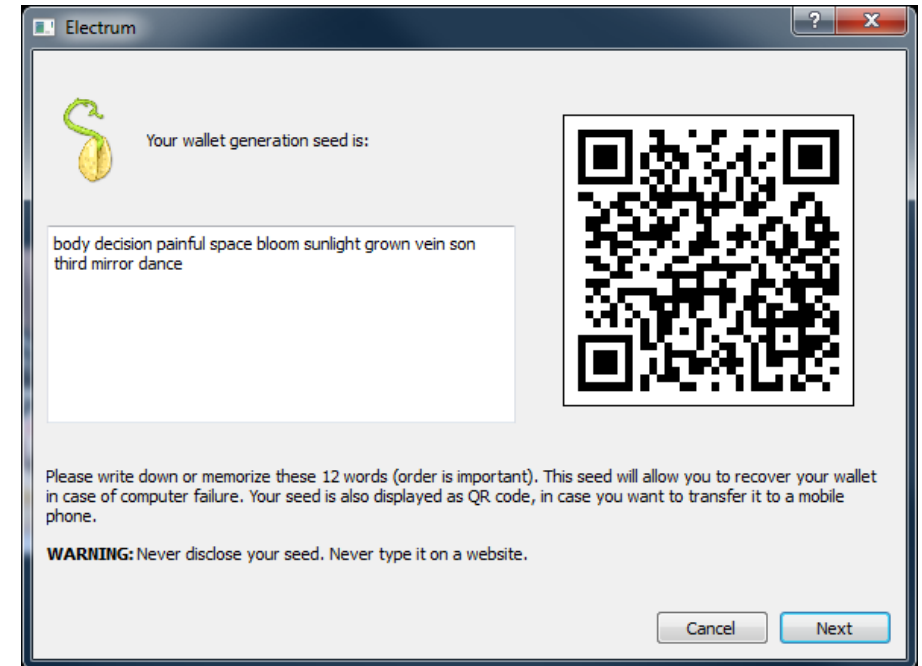
5

LIGHTNING NETWORK

- Off-chain transaction => near-instant payments

- Money flows through a payment channel
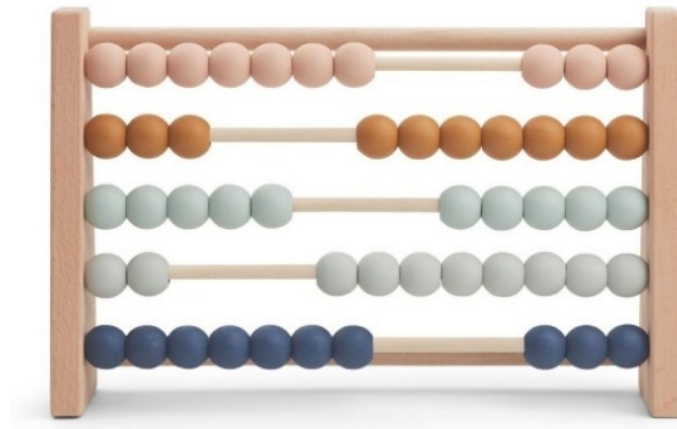
- LN is a network of payment channels

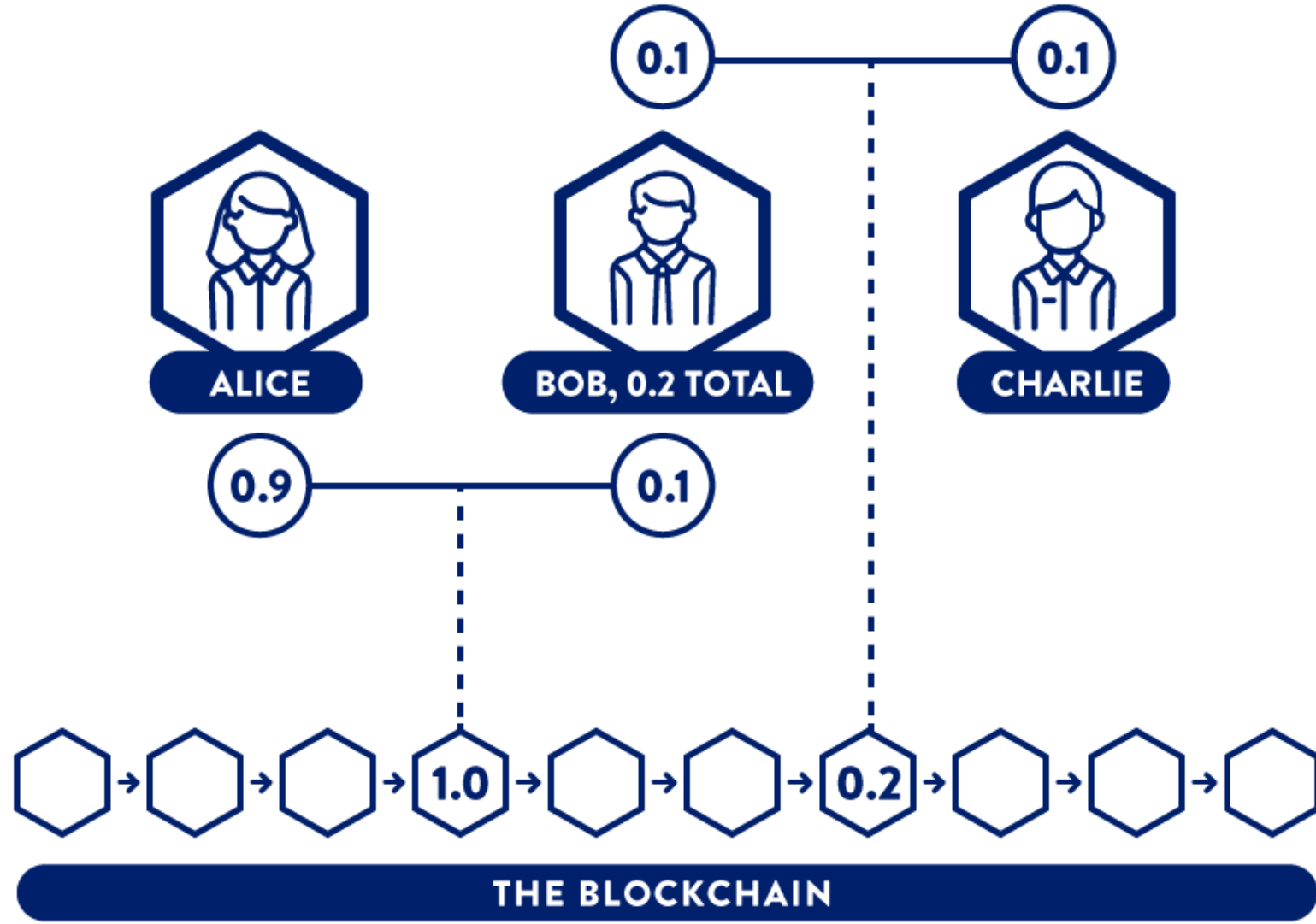- Built on top of the Bitcoin network

- Leverage a type of P2SH smart-contracts (HTLCs)

CONCLUSION

- Electronic cash system, programmable money

- 100% uptime since 10 years

- Improving quickly everyday

- Amazing open-source project

- Full of brilliant people