CBS COPENHAGEN BUSINESS SCHOOL
HANDELSHØJSKOLEN

Elective Course on Mastering Blockchain: Foundations to Consensus

Bitcoin Blocks, Transactions, and Transaction-based Ledger

Raghava Mukkamala

Associate Professor & Director, Centre for Business Data Analytics
Copenhagen Business School, Denmark
Email: rrm.digi@cbs.dk, Centre: https://cbsbda.github.io/

Course Coordinator at SRMIST:

Prof. K. Shantha Kumari

Associate Professor
Data Science and Business Systems Department,
SRM Institute of Science and Technology, India
Shanthak@srmist.edu.in

EQUIS ACCREDITED AACSB ACCREDITED ASSOCIATION AMBA ACCREDITED CEMS PIM

Outline

- Simple Mechanics of Digital / Crypto Currencies
- Bitcoin Transactions
- Bitcoin Blocks and Block structure
- Merkle Trees and Transactions

Simple Mechanics of Digital / Crypto Currencies

Many of the slides have been taken and adapted from:
Bitcoin and Cryptocurrency Technologies
with due credit to the generous authors of slides from:
http://bitcoinbook.cs.princeton.edu/

Image by WorldSpectrum from Pixabay

CBS

Value of money/currency

Is money a belief system? Or real?

Yes. Just as economics is a shared illusion. Since money has a value that is collectively agreed upon, with no inherent value, it is only the shared illusion that allows money to have that value.

https://www.quora.com/Is-money-a-belief-system

Semantics of digital currency

Fiat Currency

Let's put ourselves in the shoes of someone who needs to design a digital currency. We need to answer a few questions to do so.

- What is the notion of the currency in digital form?
- How and who can/will create it?
- How can we spend/transfer it to others in return for a service or goods?
- Who will verify for us if it is a counterfeit/double-spend or not?
- What are the rules of the currency, and who will maintain it?

To understand these intricacies, let's take a look at some simple designs of currencies

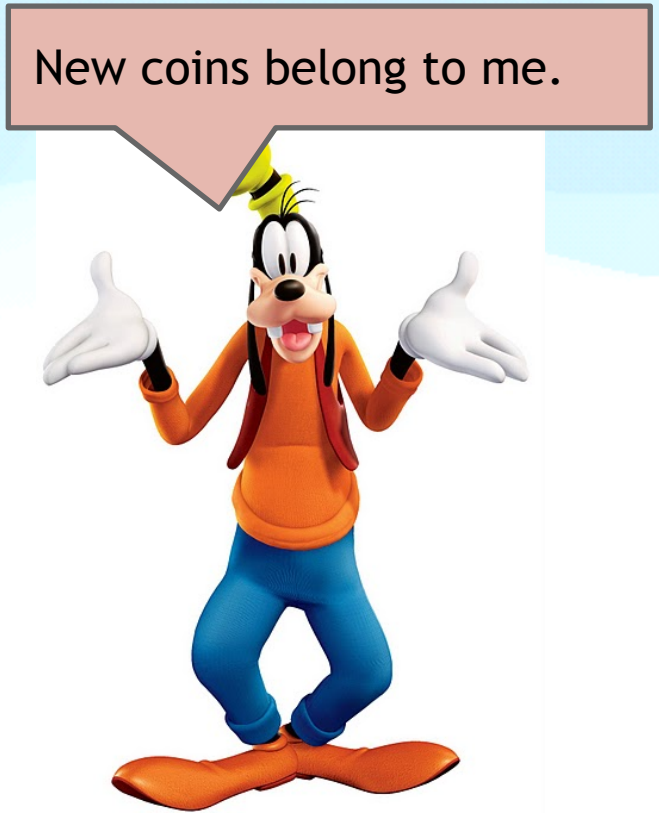
GoofyCoin

Creating Coins

Goofy can create new coins

signed by sk_{Goofy}

CreateCoin [uniqueCoinID]



Note that sk_Goofy is private key of Goofy

Creating Coins

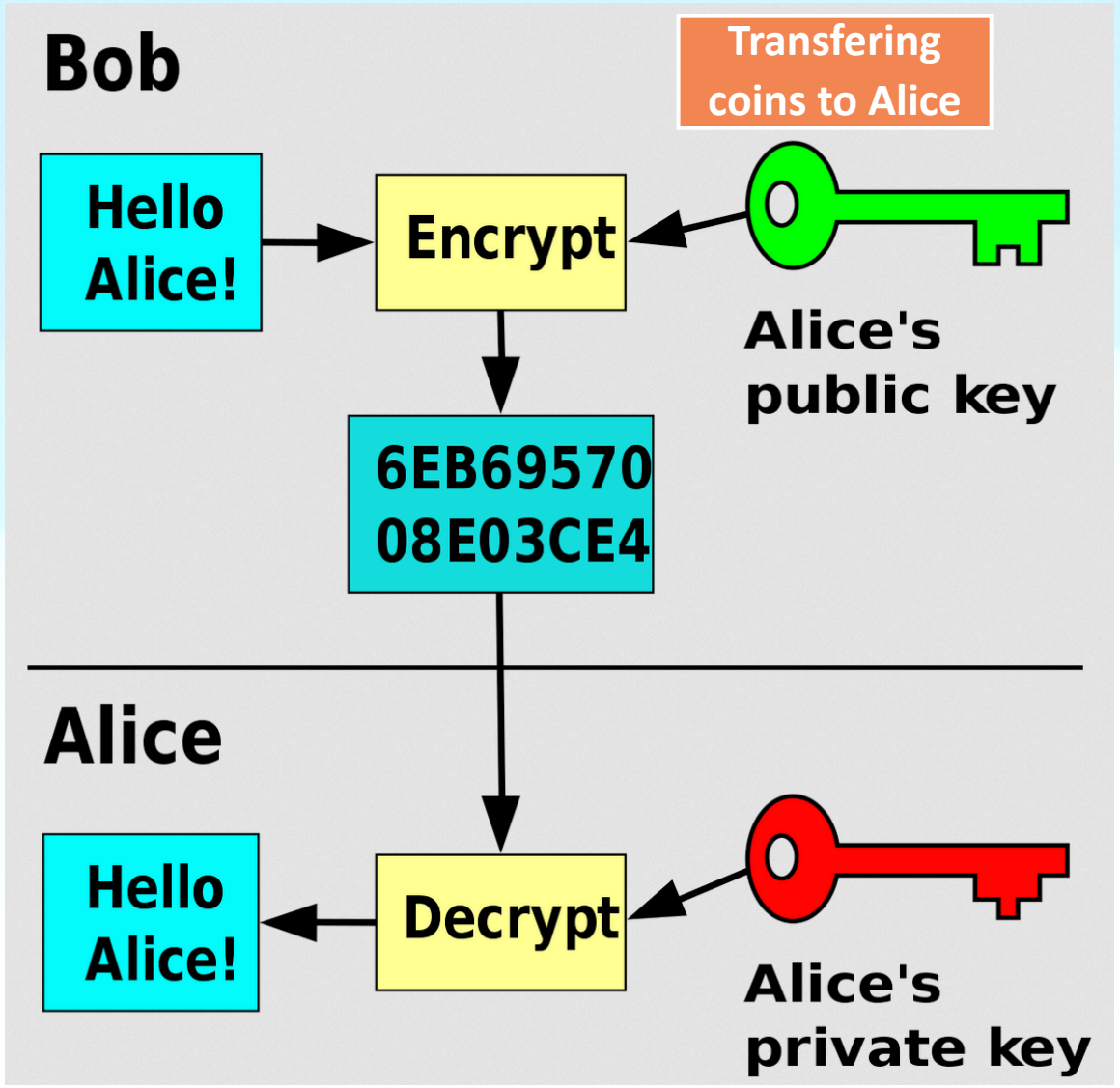
Why do want him to use his private key?

Why can't he use his public key to sign?

signed by sk_{Goofy}
CreateCoin [uniqueCoinID]

Note that sk_Goofy (sk_{Goofy}) is private key of Goofy

Asymmetric Cryptography - Encryption

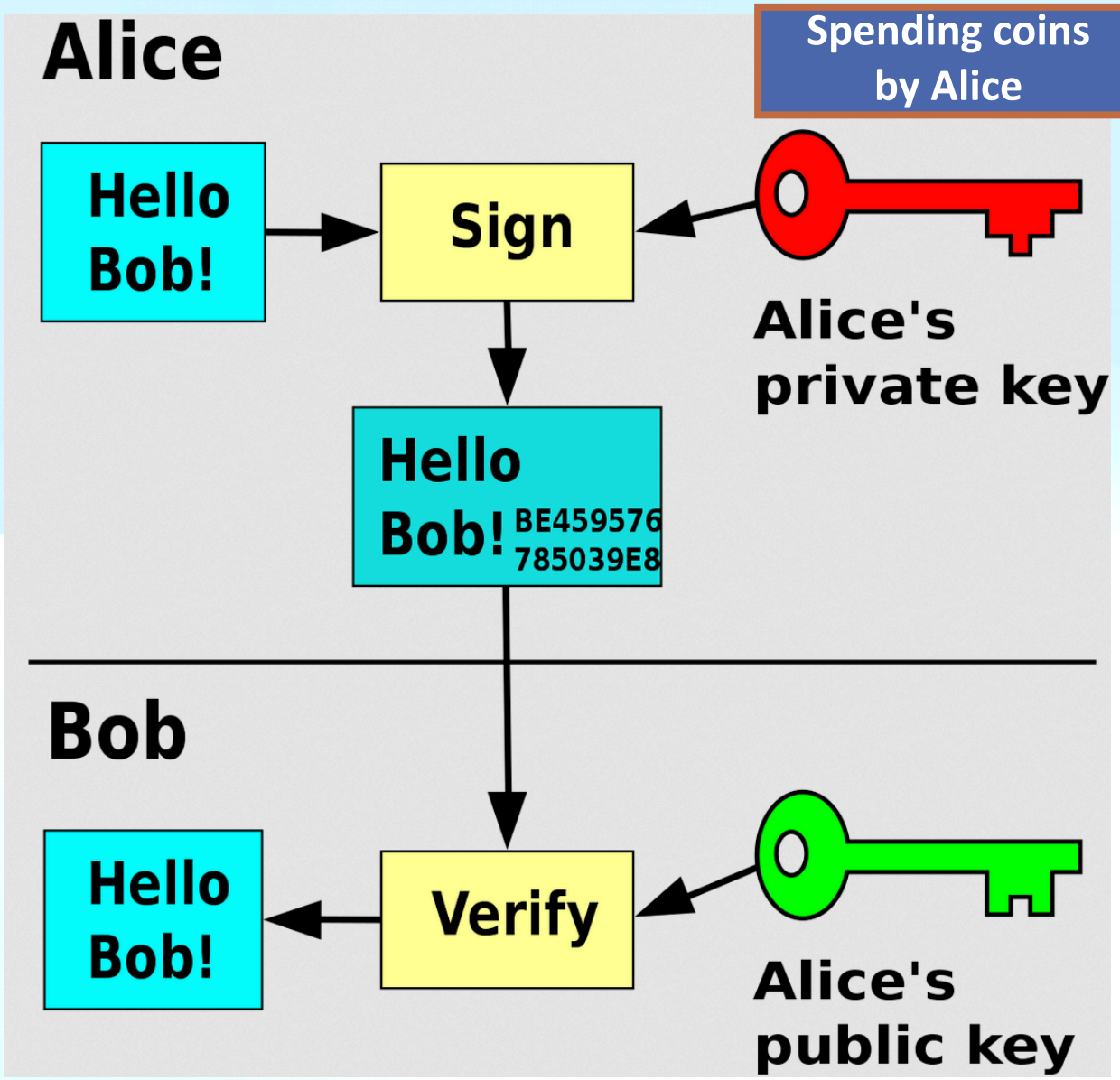


Goal is to send secure message so that only Alice (none other) can read the message.

Spending coins by Alice

https://en.wikipedia.org/wiki/Public-key_cryptography

Asymmetric Cryptography - Signing



Goal is to prove that it is only Alice (none other) sent this message i.e to prove authenticity of the sender —>

We use to spend the coins allocated to her

Anybody can verify if coin is signed by Alice

https://en.wikipedia.org/wiki/Public-key_cryptography

RSA Algorithm

RSA Algorithm

Key Generation

Select p,q	p and q both prime; $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n),e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e,n\}$
Private key	$KR = \{d,n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod n$

Decryption

Plaintext:	C
Ciphertext:	$M = C^d \pmod n$

https://www.youtube.com/watch?v=-jSX9fNJiN8

Spending/Transfering a Coin

A coin's owner can spend it.

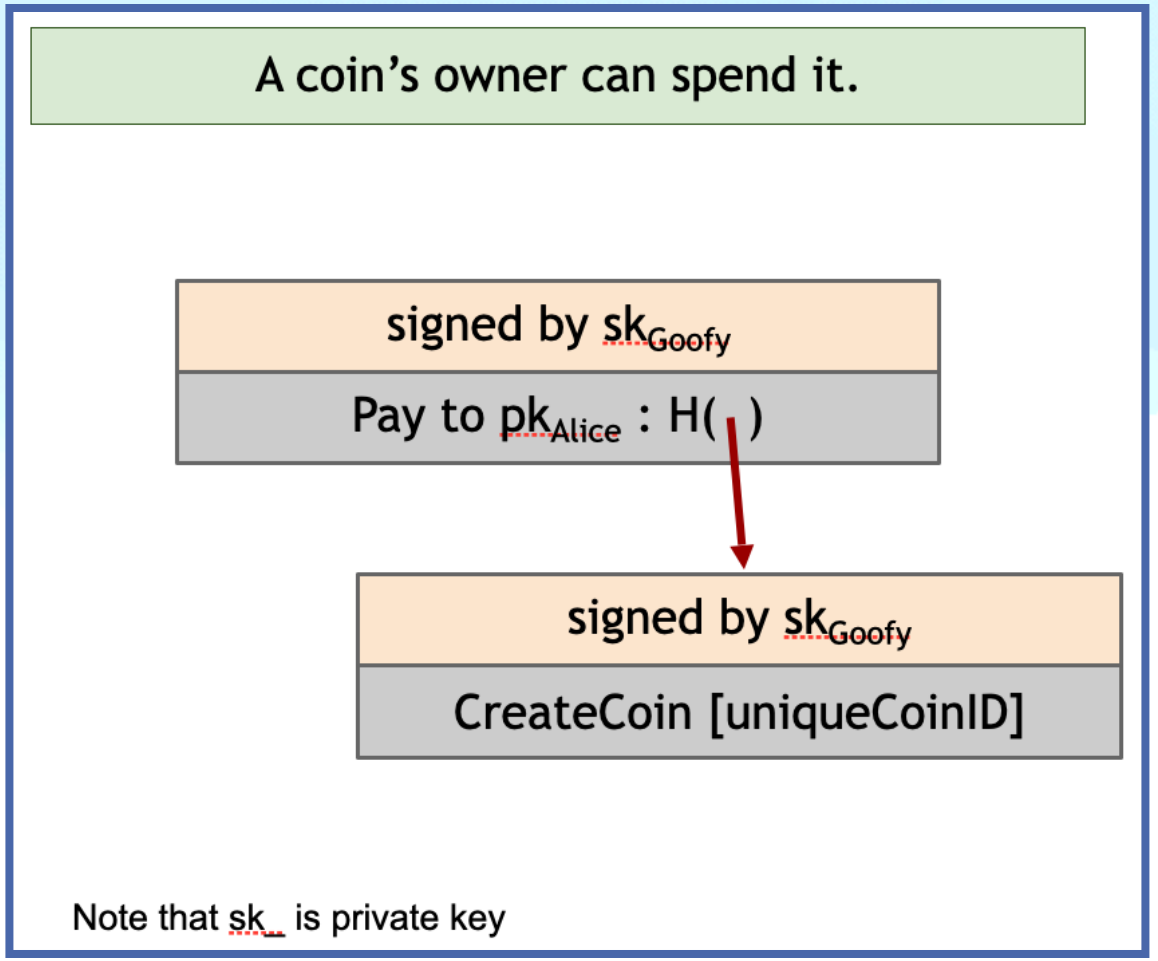
signed by sk_{Goofy}
Pay to pk_{Alice} : H( )

signed by sk_{Goofy}
CreateCoin [uniqueCoinID]

Note that sk_ is a private key, and pk_ is a public key (address)

Spending/Transfering a Coin

- Create a transaction showing the coin is transferred to Alice's public key
- Then, provide a link to the original transaction, which shows ownership of the coin
- Finally, sign the transaction to make it valid with the respective private key (e.g. sk_{Goofy})
- The signature will also prove that you are the owner of it.



Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.44

The recipient can pass on the coin again.

signed by sk_{Alice}

Pay to pk_{Bob} : H( )

signed by sk_{Goofy}

Pay to pk_{Alice} : H( )

signed by sk_{Goofy}

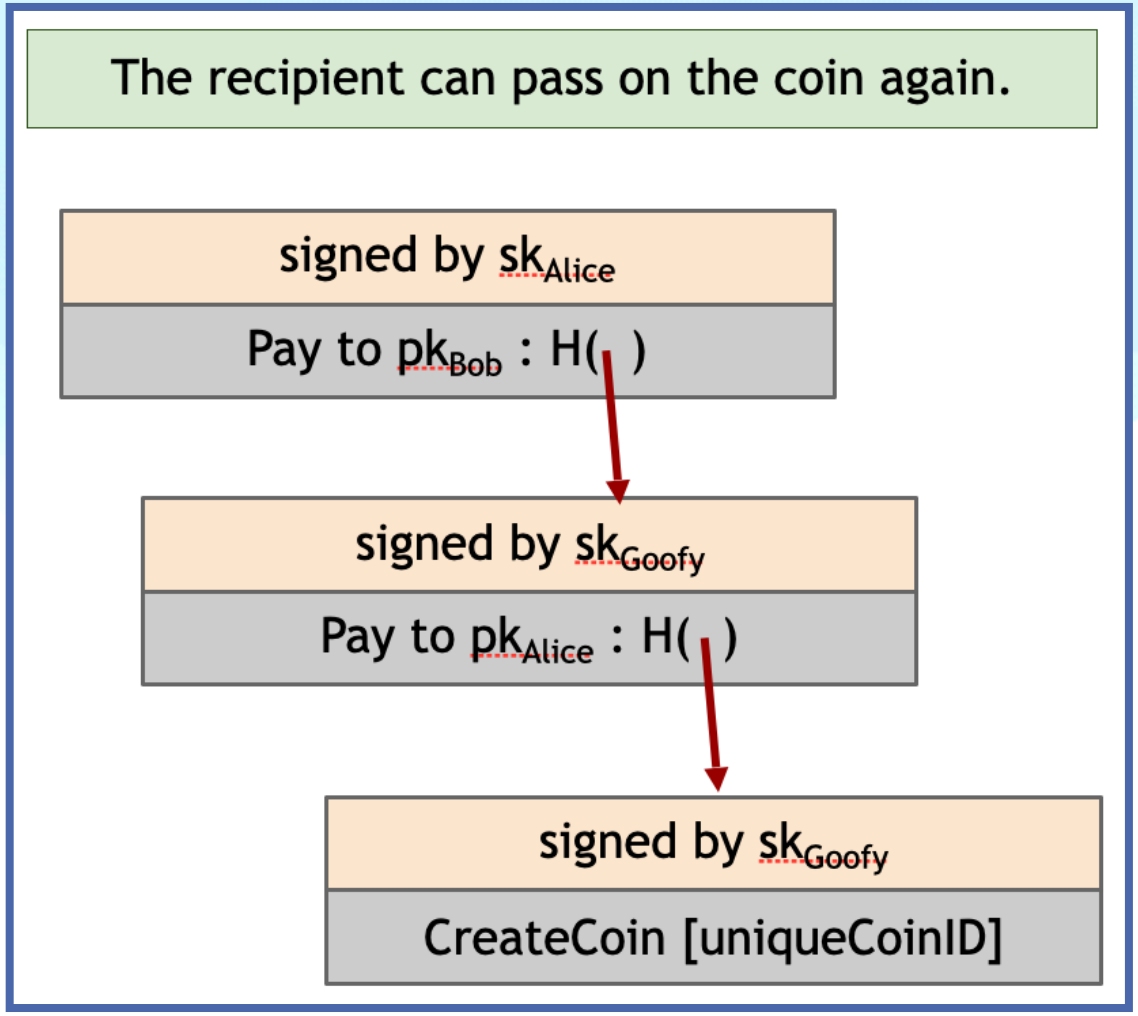
CreateCoin [uniqueCoinID]

Note that pk_{Alice} and sk_{Alice} should be corresponding keys

Rule of GoofyCoin

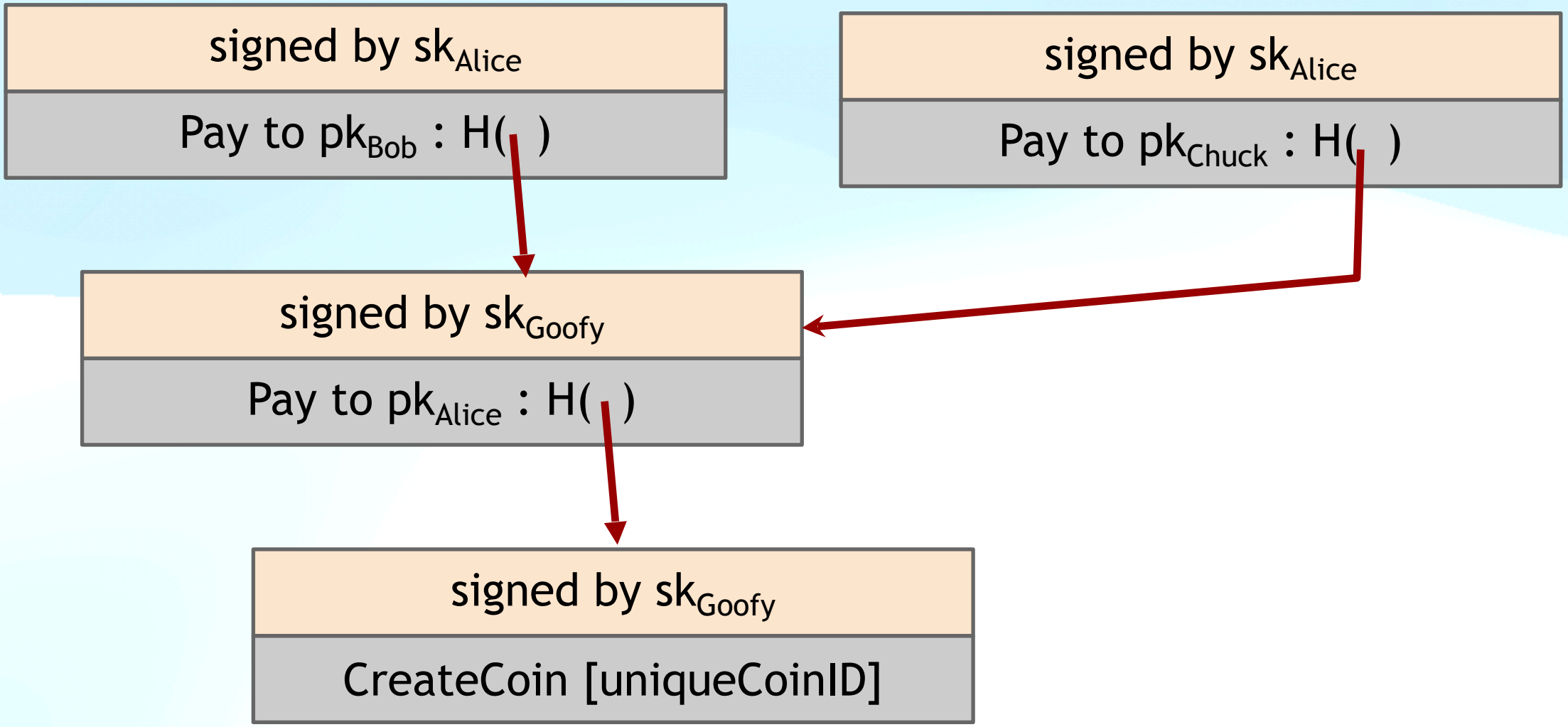
To summarize, the rules of GoofyCoin are

- Goofy can create new coins by simply signing a statement that he's making a new coin with a unique coin ID.
- Whoever owns a coin can pass it on to someone else by signing a statement that saying, "Pass on this coin to X" (where X is public key)
- Anyone can verify the validity of a coin by following the chain of hash pointers back to its creation by Goofy, verifying all of the signatures along the way.



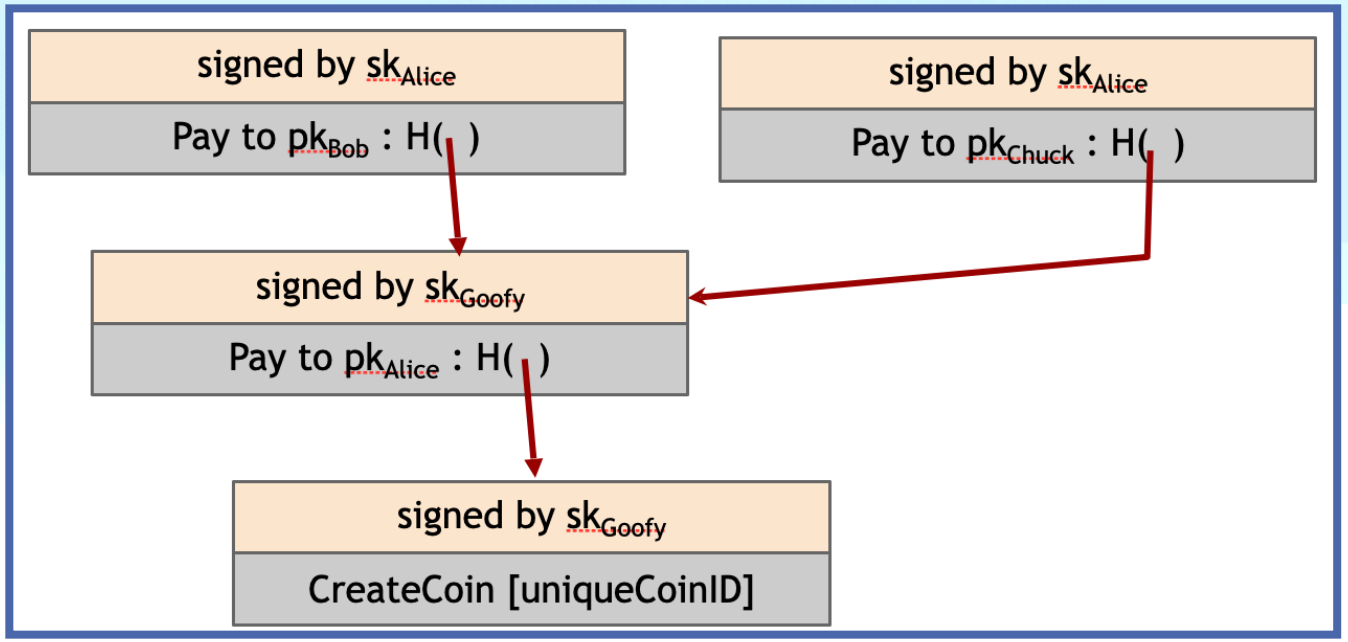
Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.44

double-spending attack



Double Spending Problem

- A fundamental security problem with GoofyCoin.
- Alice passed her coin on to Bob by sending her signed statement to Bob but didn't tell anyone else.
- To Chuck, it appears that it is a perfectly valid transaction, and he owns it
- Bob and Chuck would both have valid claims to be the owner of this coin.
- GoofyCoin does not solve the double-spending attack, as transactions are **not publicly visible.**



Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.44

double-spending attack

the main design challenge in digital currency

ScroogeCoin

ScroogeCoin

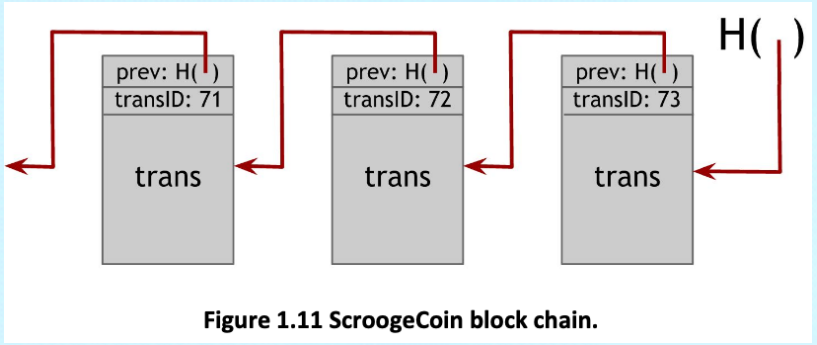


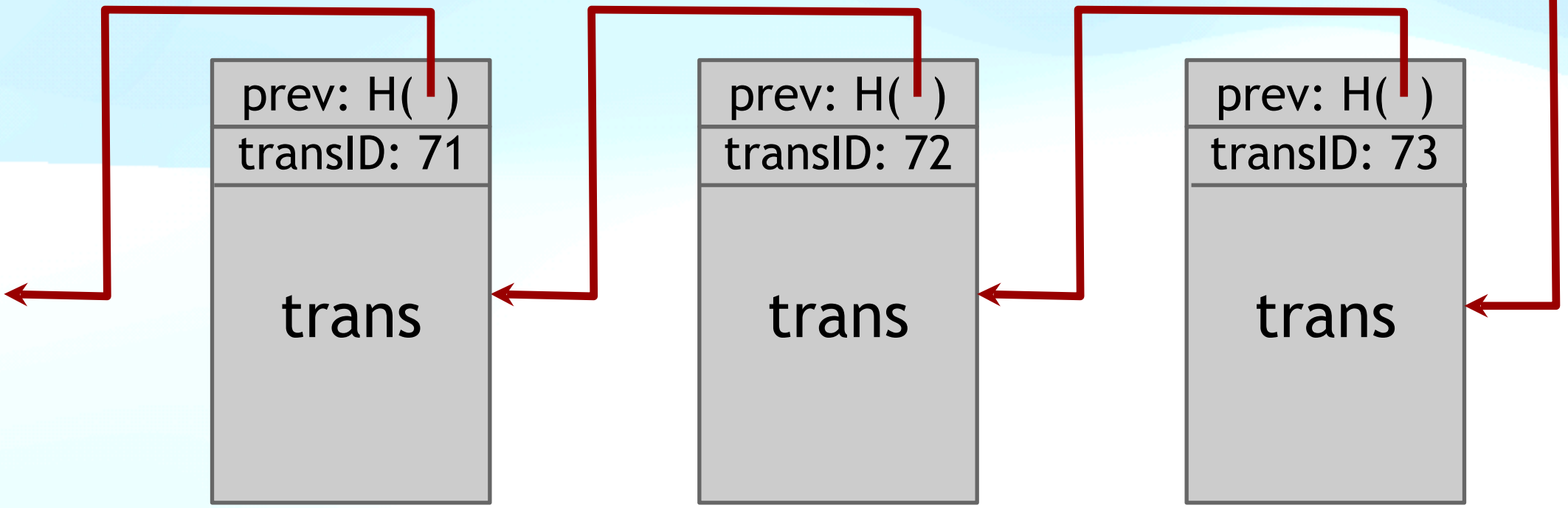
Figure 1.11 ScroogeCoin block chain.

- The first key idea is that a **designated entity** called Scrooge publishes an append-only-ledger containing the history of all the transactions that have happened.
- The append-only property ensures that any data written to this ledger will remain forever.
- If the ledger is truly append-only, we can use it to defend against double-spending by requiring all transactions to be written the ledger before they are accepted.
- That way, it will be publicly visible if coins were previously sent to a different owner.

Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.44

Scrooge publishes a history of all transactions
(a block chain, signed by Scrooge)

H( )



optimization: put multiple transactions in the same block

ScroogeCoin - Ledger

- To implement this append-only functionality, Scrooge can build a blockchain (the data structure) which he will digitally sign.
- It's a series of data blocks, each with one transaction in it
- Each block has the ID of a transaction, the transaction's contents, and a hash pointer to the previous block.
- Scrooge digitally signs the final hash pointer (i.e., for all blocks), which binds all of the data in this entire structure, and publishes the signature along with the blockchain.

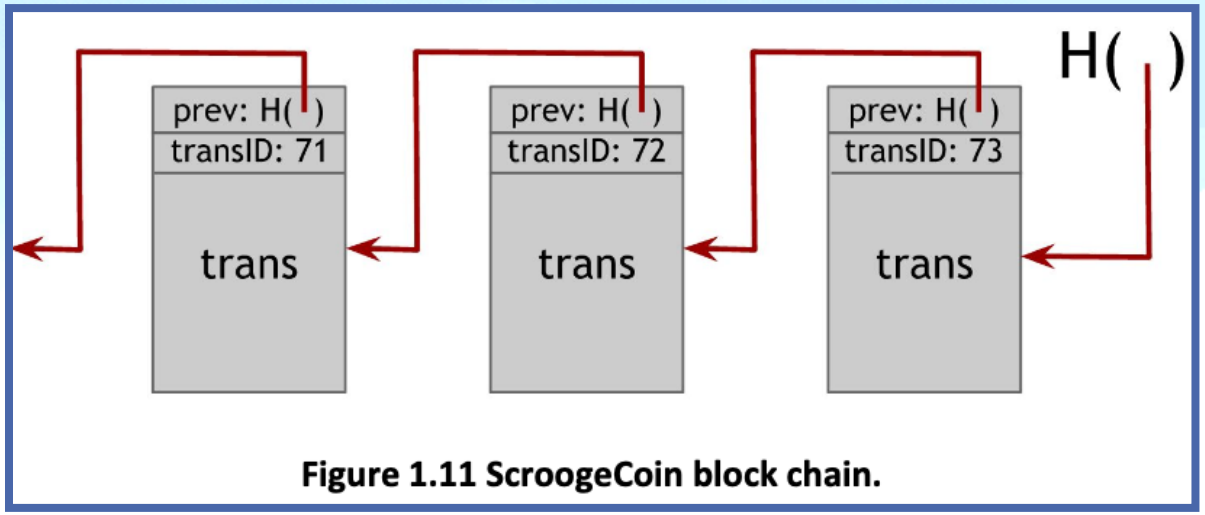


Figure 1.11 ScroogeCoin block chain.

Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.44

ScroogeCoin - Transaction Verification

- In ScroogeCoin, a transaction only counts if it is in the blockchain signed by Scrooge.
- Anybody can verify that Scrooge endorsed a transaction by checking Scrooge's signature on the block that it appears in.
- Scrooge ensures he doesn't endorse a transaction that attempts to double-spend an already-spent coin.

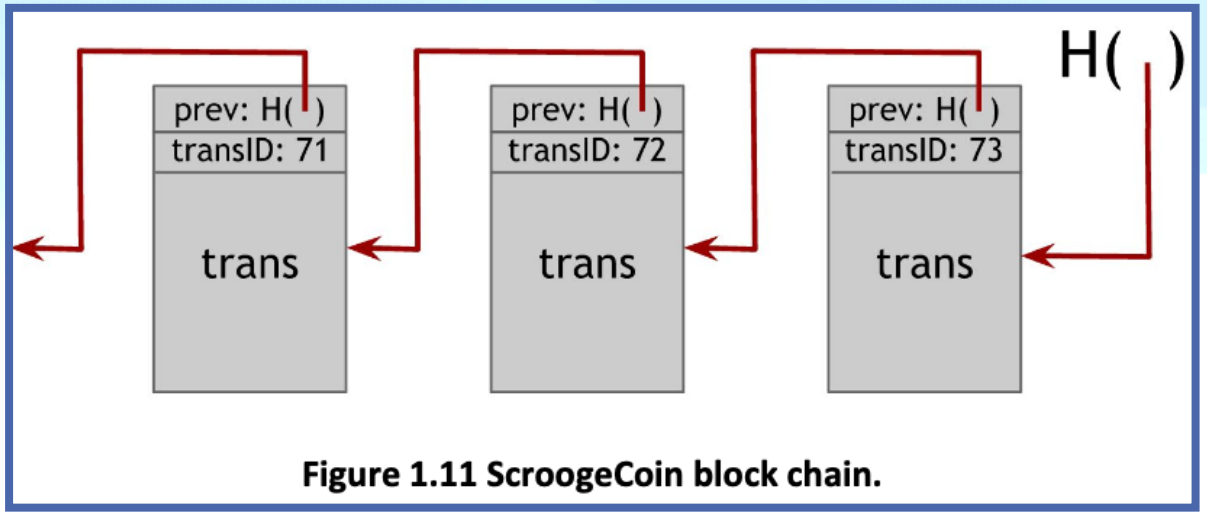


Figure 1.11 ScroogeCoin block chain.

Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.45

Why append-only ledger?

- Why do we need a blockchain with hash pointers in addition to having Scrooge sign each block?
- This ensures the append-only property. Even if Scrooge tries to add or remove a transaction to the history or change an existing transaction, it will affect all of the following blocks because of the hash pointers.
- As long as someone monitors the **latest hash pointer** published by Scrooge, the change will be obvious and easy to catch. We don't need to store all the blockchain for monitoring. **Why???**
- In a system where Scrooge signed blocks individually, you'd only need to keep track of signatures by Scrooge.
- A blockchain makes it very easy for any two individuals to verify that they have observed the exact same history of transactions signed by Scrooge.

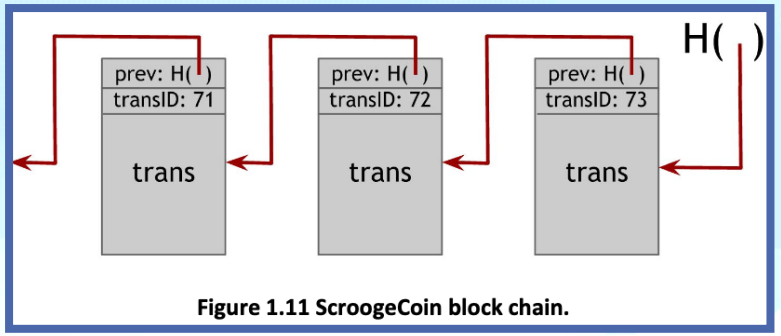


Figure 1.11 ScroogeCoin block chain.

Remember:

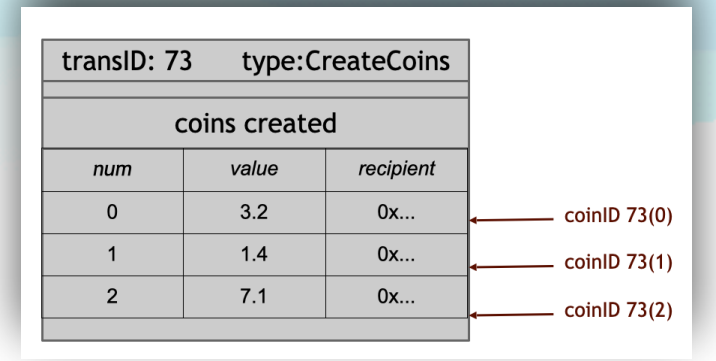
Second pre-image resistant property of Hash functions

Given x, find y≠ x such that h(x) = h(y)

Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.45

Transaction Semantics

- In ScroogeCoin, there are two kinds of transactions: CreateCoins and PayCoins
- CreateCoins is just like the operation Goofy could do in GoofyCoin makes a new coin.
- With ScroogeCoin, we'll extend the semantics a bit to allow multiple coins to be created in one transaction.
- Each coin has a serial number within the transaction.
- Each coin has a value; a certain number of scrooge coins.
- Each coin has a recipient, which is a public key that gets the coin when it's created.



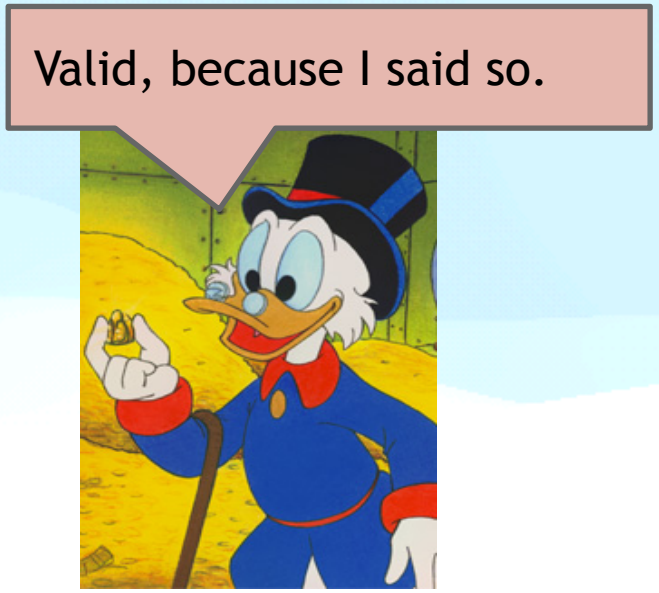
CreateCoins transaction creates new coins

transID: 73	type:CreateCoins	
coins created		
num	*value*	*recipient*
0	3.2	0x...
1	1.4	0x...
2	7.1	0x...

coinID 73(0)

coinID 73(1)

coinID 73(2)



Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.44

PayCoins transaction consumes (and destroys) some coins, and creates new coins of the same total value

transID: 75 type:PayCoins

consumed coinIDs:
68(1), 42(0), 73(2)

coins created

num	*value*	*recipient*
0	2.5	0x...
1	0.6	0x...
2	6.4	0x...

signatures

Valid if:
-- consumed coins valid,
-- not already consumed,
-- total value out = total value in, and
-- signed by owners of all consumed coins

Immutable coins

Coins can't be transferred, subdivided, or combined.

But: you can get the same effect by using transactions

to subdivide: create new trans

consume your coin

pay out two new coins to yourself

- Let's say that you have a coin of value 3.14, and you want to pay 1.1 to your friend.
- You can create a transaction where you spend a coin of value 3.14 and create 2 new coins:
  - 1) a coin of value 1.1 to your friend's public key and
  - 2) a coin of value 2.04 to your public key.

Problem of ScroogeCoin

- ScroogeCoin will work because people can see which coins are valid. It prevents double-spending because everyone can look into the blockchain and see that all of the transactions are valid and that every coin is consumed only once.
- But the problem is Scrooge — he has too much influence. He can't create fake transactions because he can't forge other people's signatures. But he could stop endorsing transactions from some users, denying them service and making their coins unspendable.
- If Scrooge is greedy (as his cartoon namesake suggests), he could refuse to publish transactions unless they transfer some mandated transaction fee to him.
- Scrooge can also, of course, create as many new coins for himself as he wants. Or Scrooge could get bored of the whole system and stop updating the blockchain completely.
- Many reasons people do not accept a cryptocurrency with a centralized authority.

ScroogeCoin

Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.47

Crucial question:

Can we descroogify the currency, and operate without any central, trusted party?

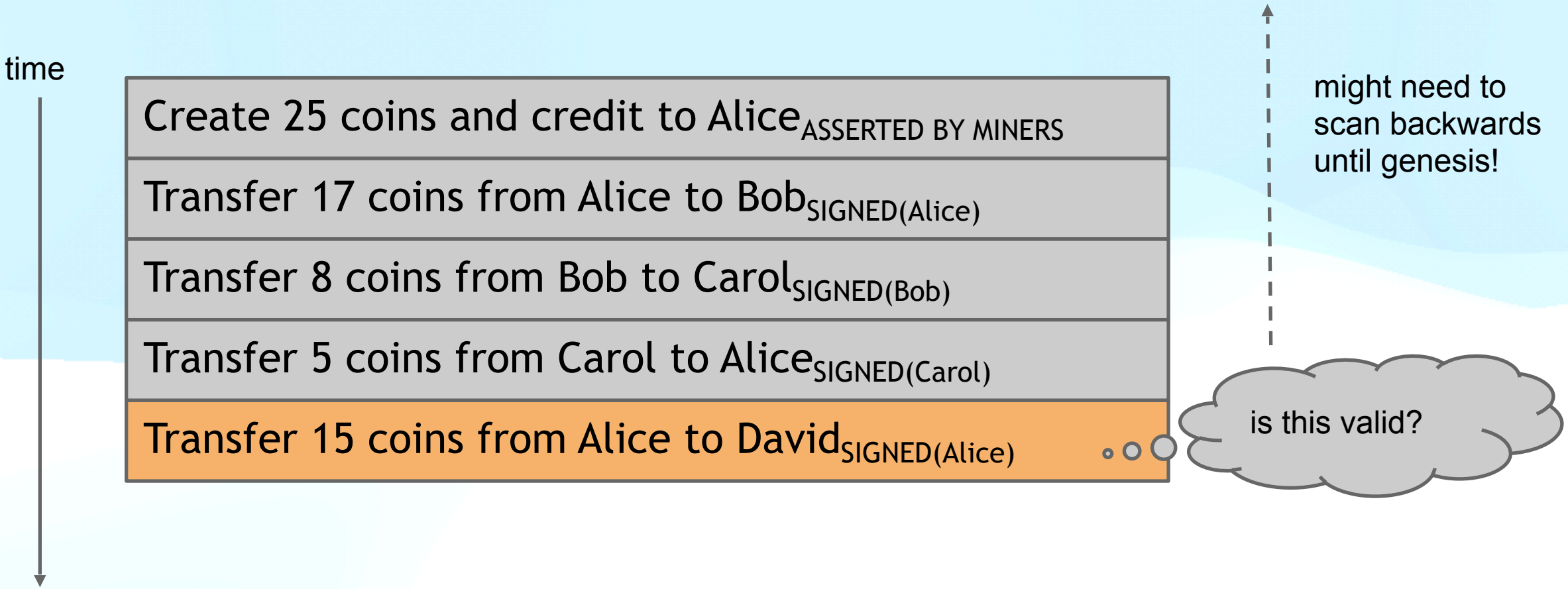
Account-based verses Transaction-based Ledger

Many of the slides have been taken and adapted from:
Bitcoin and Cryptocurrency Technologies
with due credit to the generous authors of slides from:
http://bitcoinbook.cs.princeton.edu/

Image by MichaelWuensch from Pixabay

CBS

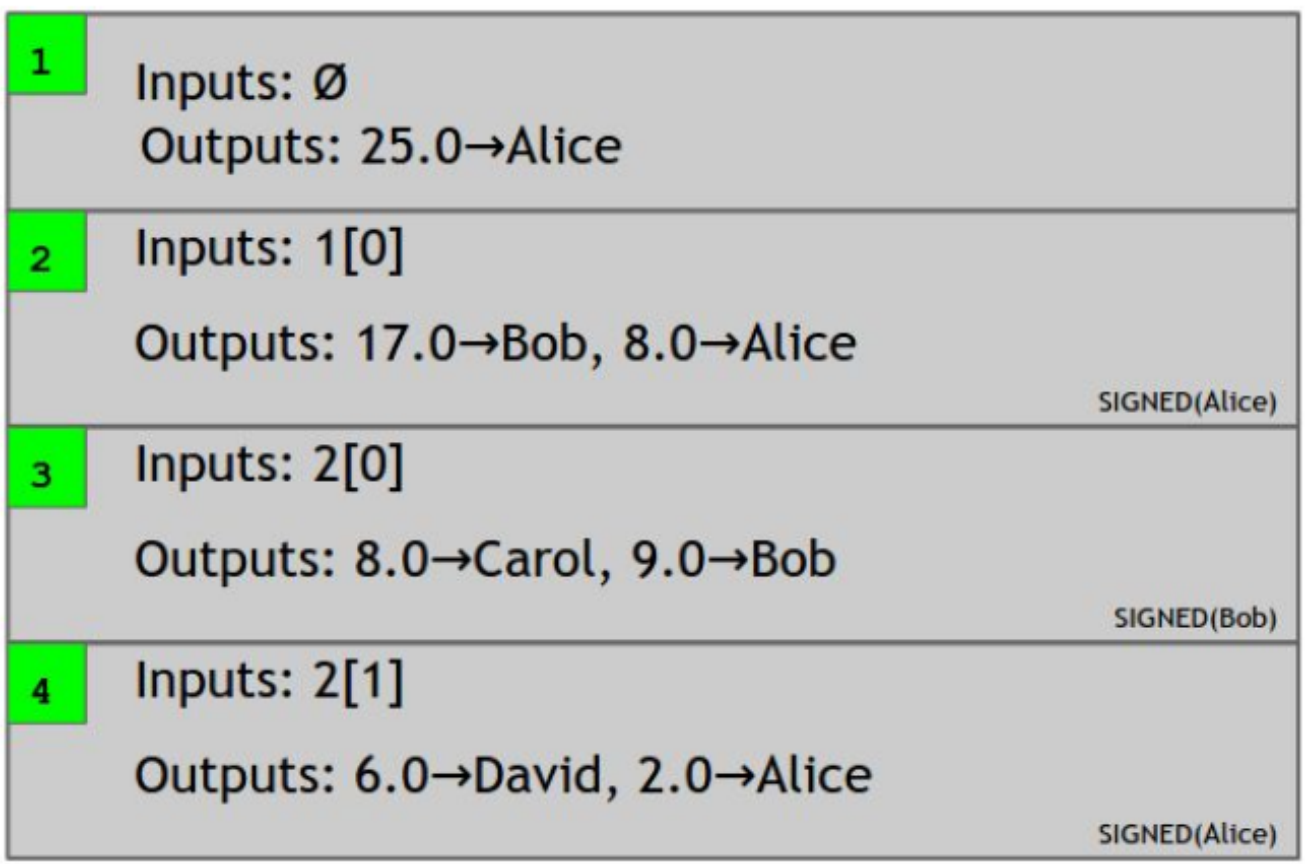
An account-based ledger (*not* Bitcoin)



SIMPLIFICATION: only one transaction per block

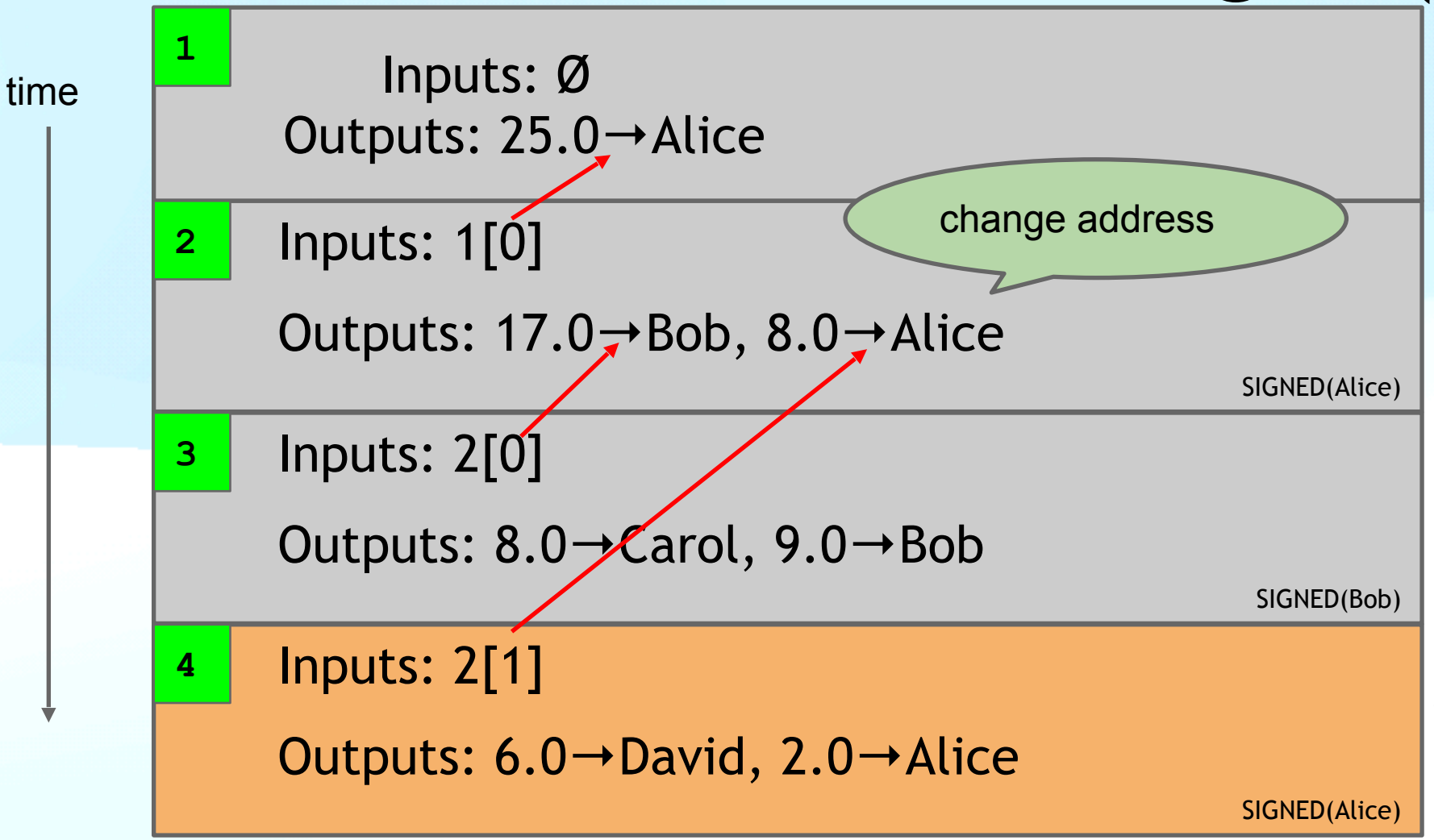
Transaction-based Ledger

- Transactions specify a number of inputs and several outputs (recall PayCoins in ScroogeCoin).
- You can think of the inputs as coins being consumed (created in a previous transaction) and the outputs as coins being created.
- For transactions in which new currency is being minted, there are no coins being consumed (recall CreateCoins in ScroogeCoin).
- Each transaction has a unique identifier. Outputs are indexed beginning with 0, so we will refer to the first output as "output 0".



Book: Bitcoin and Cryptocurrency Technologies: http://bitcoinbook.cs.princeton.edu/ p.47

A transaction-based ledger (Bitcoin)



we implement this with hash pointers

finite scan to check for validity

is this valid?

SIMPLIFICATION: only one transaction per block

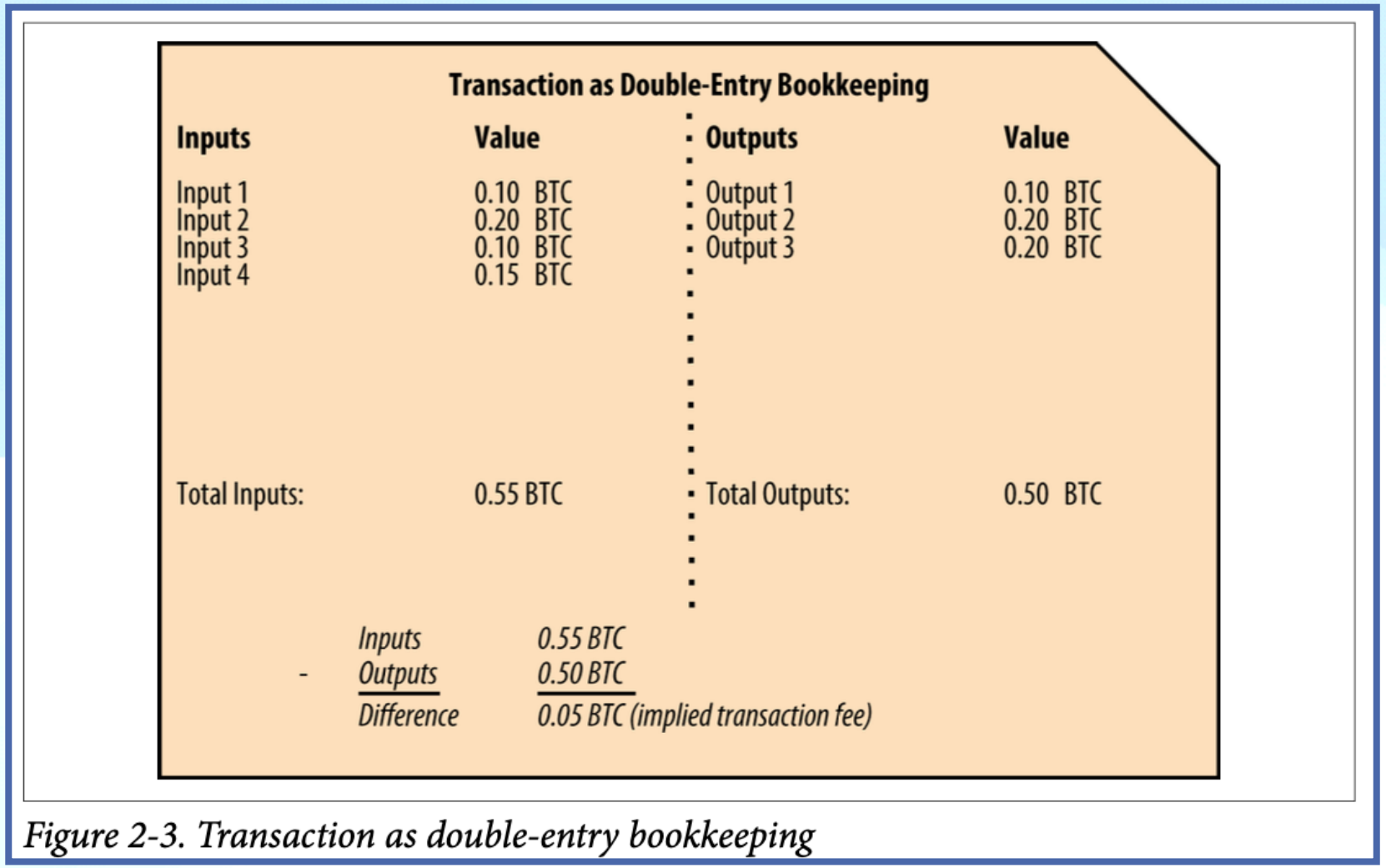
Bitcoin Transactions

Many of the slides have been taken and adapted from:
Bitcoin and Cryptocurrency Technologies
with due credit to the generous authors of slides from:
http://bitcoinbook.cs.princeton.edu/

Image by MichaelWuensch from Pixabay

CBS

Transaction as Double-entry Bookkeeping



Transaction as Double-Entry Bookkeeping

Inputs	Value	Outputs	Value
Input 1	0.10 BTC	Output 1	0.10 BTC
Input 2	0.20 BTC	Output 2	0.20 BTC
Input 3	0.10 BTC	Output 3	0.20 BTC
Input 4	0.15 BTC		
Total Inputs:	0.55 BTC	Total Outputs:	0.50 BTC

	Inputs	*0.55 BTC*
-	*Outputs*	*0.50 BTC*
	Difference	*0.05 BTC (implied transaction fee)*

Figure 2-3. Transaction as double-entry bookkeeping

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.19

Buying a Cup of Coffee

- Alice met with her friend Joe to exchange some cash for bitcoin.
- The transaction created by Joe funded Alice's wallet with 0.10 BTC.
- Now Alice wants to buy a cup of coffee at Bob's coffee shop and order for a cup of coffee
- The point-of-sale system automatically converts the total price from US dollars to bitcoin at the prevailing market rate and displays the price in both currencies

https://pothead.coffee/how-to-buy-coffee-with-cryptocurrency/

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.19

Buying a Cup of Coffee

- Bob's point-of-sale system will create a special QR code containing a payment request.
- QR code contains a QR-encoded URL that contains a destination bitcoin address, a payment amount, and a label such as "Bob's Cafe."
- This allows a bitcoin wallet application to prefill some information, showing a human-readable description to the user.
- Alice uses her smartphone to scan the barcode and authorizes the payment.

```
bitcoin:1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA?
amount=0.015&
label=Bob%27s%20Cafe&
message=Purchase%20at%20Bob%27s%20Cafe

Components of the URL

A bitcoin address: "1GdK9UzpHBzqzX2A9JFP3Di4weBwqgmoQA"
The payment amount: "0.015"
A label for the recipient address: "Bob's Cafe"
A description for the payment: "Purchase at Bob's Cafe"
```


Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.17

Chain of Transactions

- millibitcoin = 1/ 1000 of bitcoin
- Satoshi = 1/100,000,000 of a bitcoin

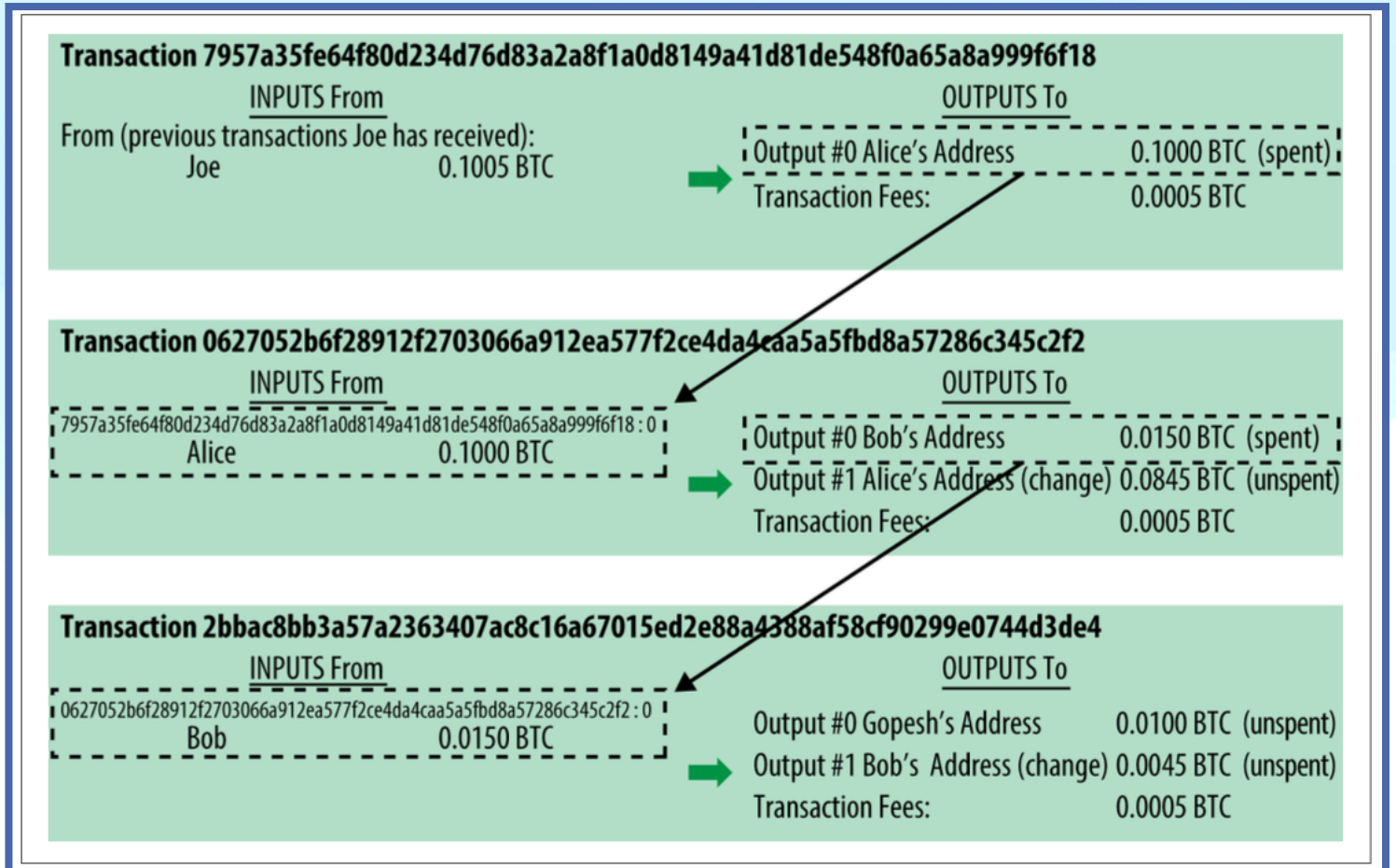
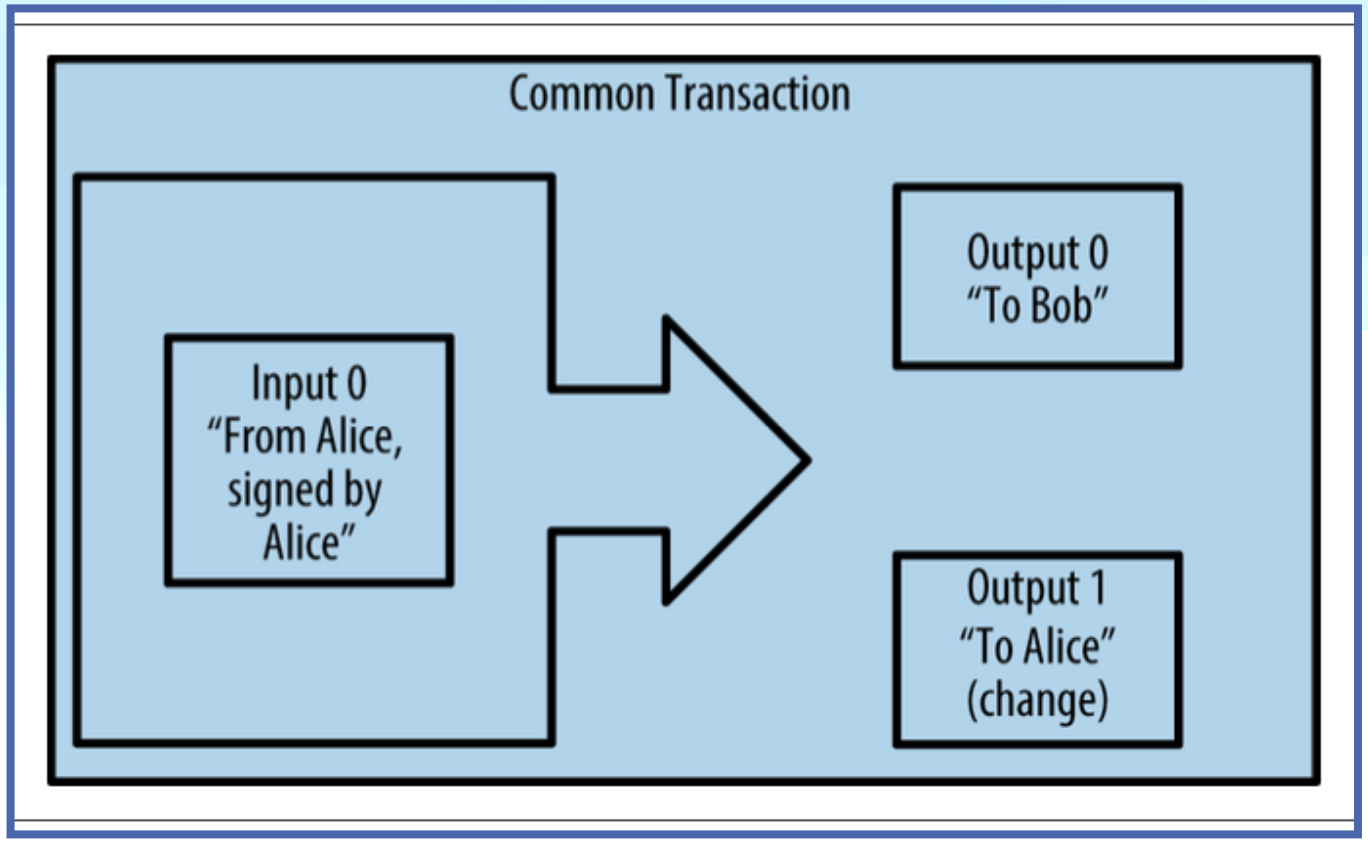


Figure 2-4. A chain of transactions, where the output of one transaction is the input of the next transaction

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.20

Common Transaction Forms - change Address

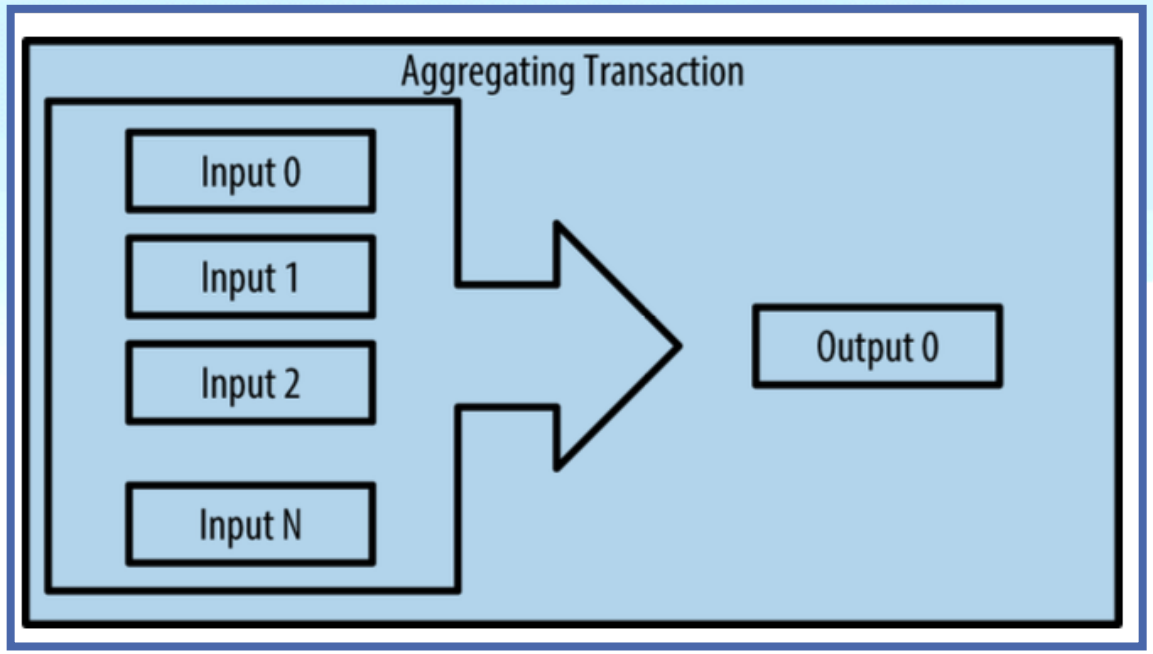
- The most common form of transaction is a simple payment from one address to another, which often includes some “change” returned to the original owner.
- This type of transaction has one input and two outputs



Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.”. p.21

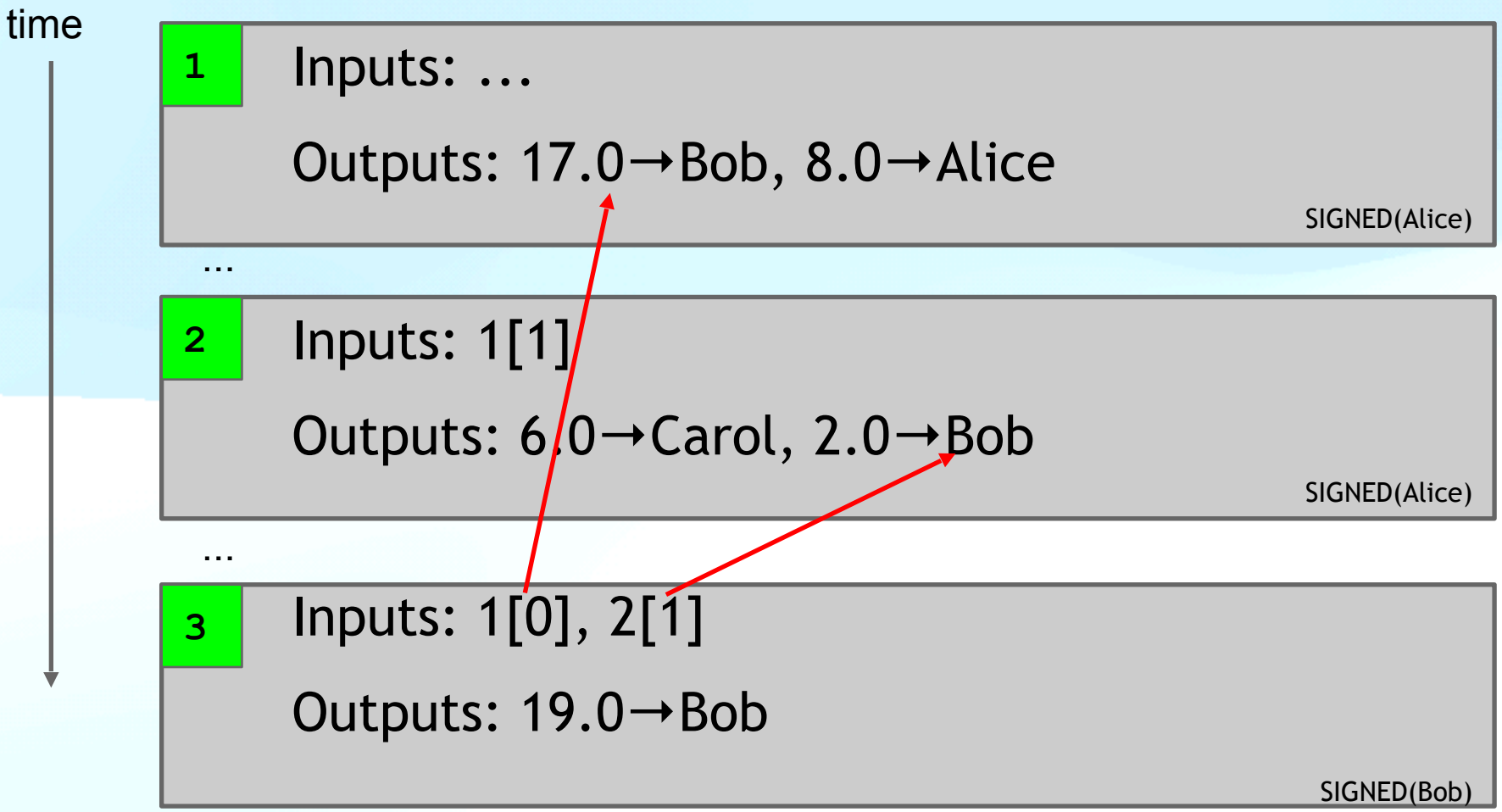
Common Transaction Forms - Aggregating funds

- Another transaction type is one that aggregates several inputs into a single output.
- This represents the real-world equivalent of exchanging a pile of coins and currency notes for a single larger note.
- Transactions like these are generated by wallet applications to clean up lots of smaller amounts that were received as a change for payments.



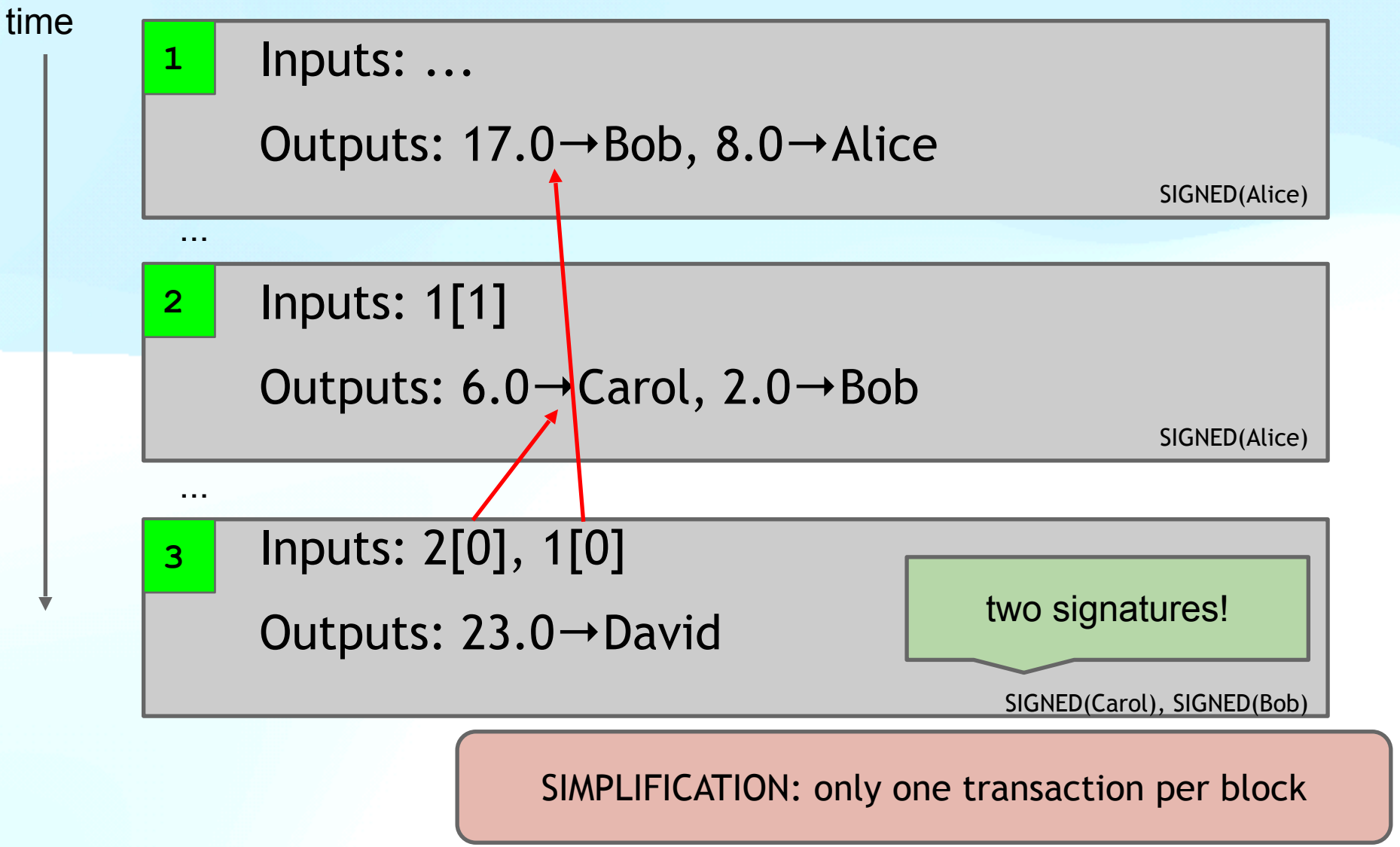
Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.2

Merging value



SIMPLIFICATION: only one transaction per block

Joint payments



The real deal: a Bitcoin transaction

metadata

input(s)

output(s)


```
{
    "hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
    "ver":1,
    "vin_sz":2,
    "vout_sz":1,
    "lock_time":0,
    "size":404,
    "in":[
     {
      "prev_out":{
       "hash":"3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
       "n":0
      },

          "scriptSig":"30440..."
     },
     {
      "prev_out":{
       "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
       "n":0
      },
      "scriptSig":"3f3a4ce81...."
     }
    ],
    "out":[
     {
      "value":"10.12287097",
      "scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"
     }
    ]

}
```

The real deal: transaction metadata

transaction hash

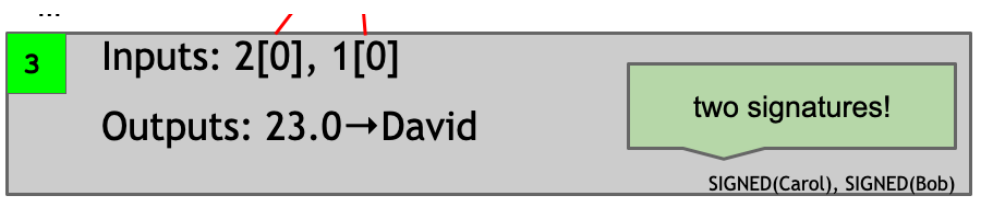
housekeeping

“not valid before”

housekeeping

```
{
    "hash":"5a42590...b8b6b",
      "ver":1,
      "vin_sz":2,
      "vout_sz":1,
      "lock_time":0,
      "size":404,
...
}
```

more on this later...



The real deal: transaction inputs

previous transaction

signature

(more inputs)

```
"in":[
  {
    "prev_out":{
      "hash":"3be4...80260",
      "n":0
    },
"scriptSig":"30440....3f3a4ce81"
  },
...
],
```

The real deal: transaction outputs

output value

recipient address??

(more outputs)

```
"out":[
    {
      "value":"10.12287097",
      "scriptPubKey":"OP_DUP OP_HASH160 69e...3d42e
OP_EQUALVERIFY OP_CHECKSIG"
    },
        ...
]
```

more on this soon...

https://blockchain.info/rawblock/000000000000000000333d265acedd64e126788ce876b46450023bddb4e6a06

Peek at Transactions of a bitcoin block

https://blockchain.info/rawblock/0000000000000000000333d265acedd64e126788ce876b46450023bddb4e6a06

APPENDIX D: Segregated Witness: Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.329-342

https://blockchain.info/rawblock/0000000000000000000333d265acedd64e126788ce876b46450023bddb4e6a06

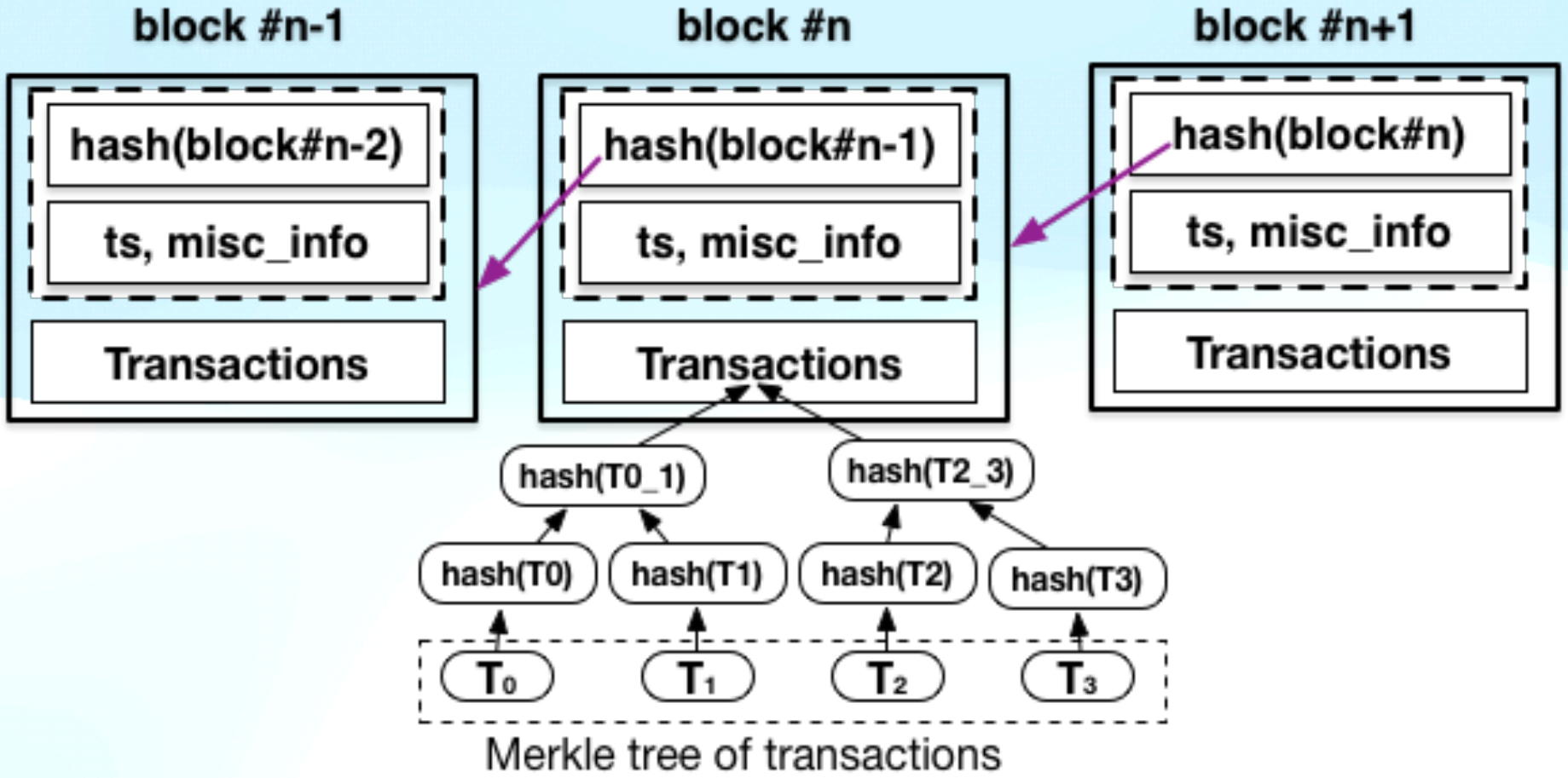
Bitcoin Blocks and block structure

Image by Gerd Altmann from Pixabay

Many of the slides have been taken and adapted from:
Bitcoin and Cryptocurrency Technologies
with due credit to the generous authors of slides from:
http://bitcoinbook.cs.princeton.edu/

CBS

Blocks in Bitcoin



Structure of Block

- BlockIdentifiers
  - Block Header Hash
  - Block Height (current height = ~775940)
  - Block Size
- Block Header
- Transactions

Block height: 663852
Block size: 1,346,857 bytes
Block hash: 00000000000000000092f5f21cd6db1d3d74252d6e343bd7a0fcac448ef01a5
Header:
Version: 0x20a00000
Timestamp: 2021-01-01 00:13
Bits: 386,867,735
Difficulty: 18,599,593,048,299.49
Nonce: 2,082,724,677
hashPrevBlock: 00000000000000000059d6c6ef39775d01feec638c693b881cacc0546e7a249
hashMerkleRoot: 956c3cdc34c804e47ff344fc19077a3122ea87413cb133c038c92ef9b08d2d2d
Number of transactions: 2,907
Block reward: 6.25 BTC
Transactions:
Tx hash: 4b65af0b601df3eb6e7237f44218b9d951ef6dcd2bd7ab45744c3ec80556bcce
1.82428143 BTC from 1CUTyyxgbKvtCdoYmceQJCZLXCde5akiX2
To: 1CUTyyxgbKvtCdoYmceQJCZLXCde5akiX2
Fee: 0.00200000 BTC (775.194 sat/B - 193.798 sat/WU - 258 bytes)
Tx 2, Tx 3, Tx 4 ...

https://levelup.gitconnected.com/a-complete-decoding-of-the-bitcoin-block-578904267142

https://blockchain.info/rawblock/00000000000000000333d265acedd64e126788ce876b46450023bddb4e6a06

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.2

Structure of Block

Table 9-1. The structure of a block

Size	Field	Description
4 bytes	Block Size	The size of the block, in bytes, following this field
80 bytes	Block Header	Several fields form the block header
1–9 bytes (VarInt)	Transaction Counter	How many transactions follow
Variable	Transactions	The transactions recorded in this block

- Block is made of a header, containing metadata, followed by a long list of transactions that make up the bulk of its size.
- The block header is 80 bytes, whereas the average transaction is at least 250 bytes
- The average block contains more than 500 transactions.
- A complete block, with all transactions, is therefore 1,000 times larger than the block header

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.2

Block Header

Table 9-2. The structure of the block header

Size	Field	Description
4 bytes	Version	A version number to track software/protocol upgrades
32 bytes	Previous Block Hash	A reference to the hash of the previous (parent) block in the chain
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The approximate creation time of this block (seconds from Unix Epoch)
4 bytes	Difficulty Target	The Proof-of-Work algorithm difficulty target for this block
4 bytes	Nonce	A counter used for the Proof-of-Work algorithm

Block #775933

Block Hash 00000000000000000046767bd78bd408f3cf3885e44aae82991bf2389d81e2d

Summary

Merkle Root	3180614cb8a7da1fb5f6b6969356f515e4fd66849...	Difficulty	1.2595093284770418
Bits	386344736	Size (bytes)	597927
Version	811769856	Nonce	2130922635
Number of Transactions	2769	Previous Block	775932
Height	775933	Next Block	775934
Block Reward	6.464 BTC	Confirmations	1
Timestamp	10/02/2023, 22.58.53		

Genesis Block: https://www.blockchain.com/explorer/blocks/btc/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

Current Block: https://bitpay.com/insight/BTC/mainnet/block/00000000000000000046767bd78bd408f3cf3885e44aae82991bf2389d81e2d

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.2

Genesis Block

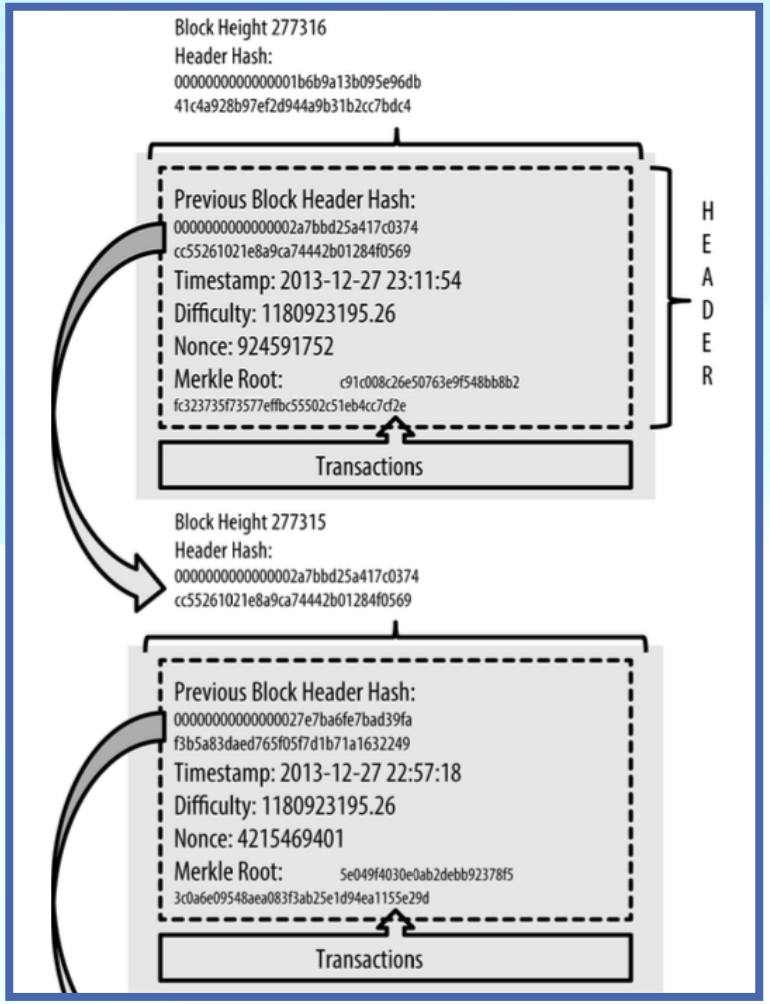
```
GetHash()      = 0x000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
hashMerkleRoot = 0x4a5e1e4baab89f3a32518a88c31bc87f618f76673e2cc77ab2127b7afdeda33b
txNew.vin[0].scriptSig     = 486604799 4
0x736B6E616220726F662074756F6C69616220646E6F63657320666F206B6E697262206E6F20726F6C6C65636E6168432039303032 2F6E614A2F33302073656D695420656854
txNew.vout[0].nValue       = 5000000000
txNew.vout[0].scriptPubKey =
0x5F1DF16B2B704C8A578D0BBAF74D385CDE12C11EE50455F3C438EF4C3FBCF649B6DE611FEAE06279A60939E028A8D65C10B73071A6F16719274855FEB0FD8A6704 OP_CHECKSIG
block.nVersion = 1
block.nTime    = 1231006505
block.nBits    = 0x1d00ffff
block.nNonce   = 2083236893

CBlock(hash=000000000019d6, ver=1, hashPrevBlock=00000000000000, hashMerkleRoot=4a5e1e, nTime=1231006505, nBits=1d00ffff, nNonce=2083236893, vtx=1)
  CTransaction(hash=4a5e1e, ver=1, vin.size=1, vout.size=1, nLockTime=0)
    CTxIn(COutPoint(000000, -1), coinbase
04ffff001d0104455468652054696d65732030332f4a616e2f3230303920436861696e63656c6c6f72206f6e206272696e6b206f66207365636f6e64206261696c6f757420666f722062616e6b73)
    CTxOut(nValue=50.00000000, scriptPubKey=0x5F1DF16B2B704C8A578D0B)
  vMerkleTree: 4a5e1e
```

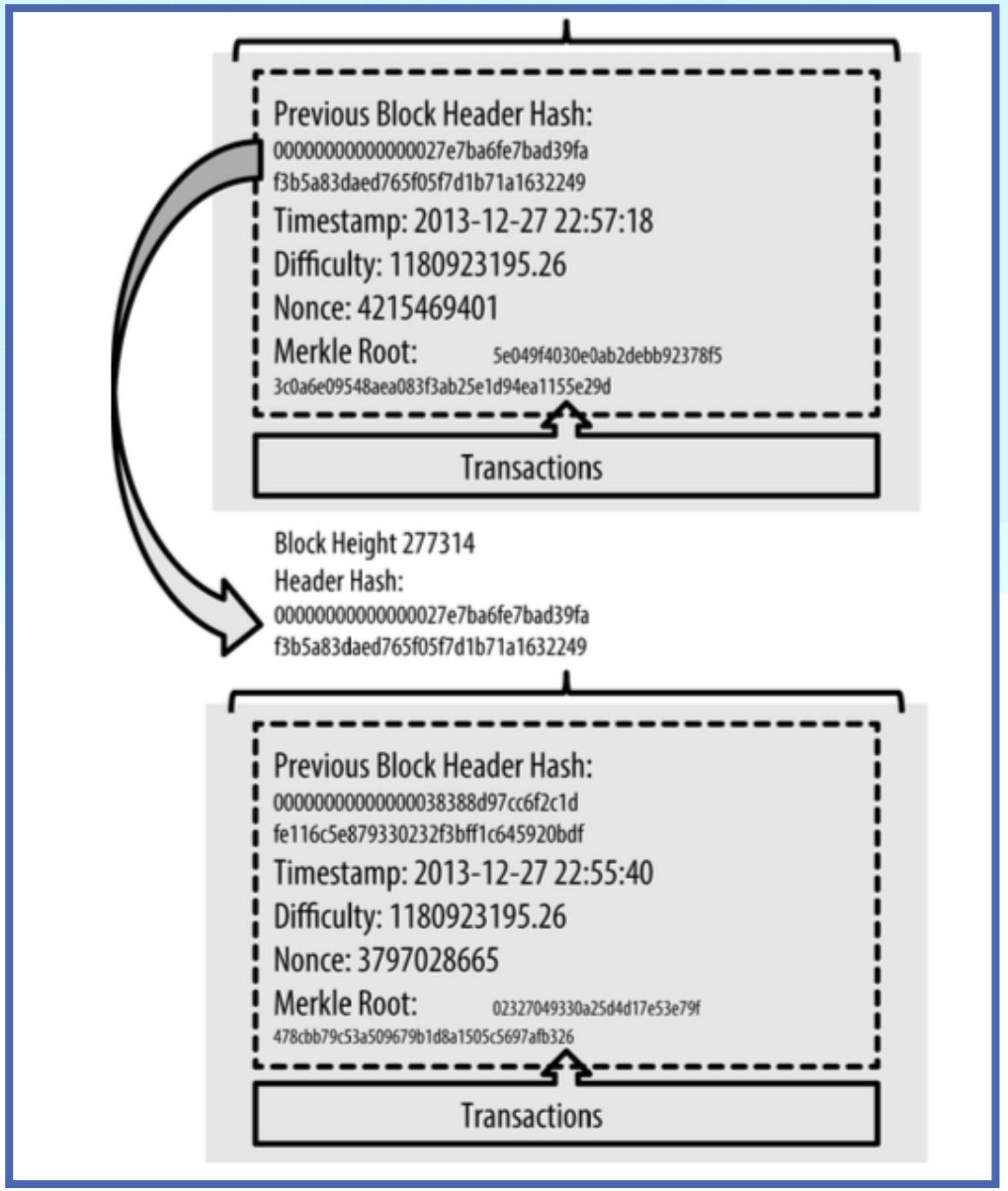
https://en.bitcoin.it/wiki/Genesis_block

Genesis Block: https://www.blockchain.com/explorer/blocks/btc/000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f
Another block:https://www.blockchain.com/explorer/blocks/btc/00000000000000000000333d265acedd64e126788ce876b46450023bddb4e6a06

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.2

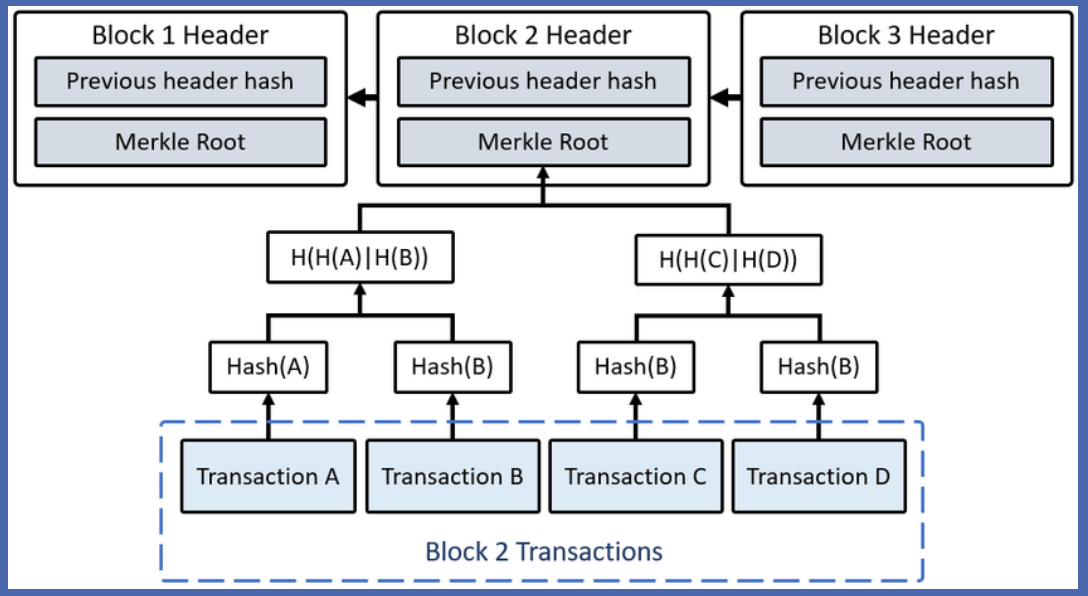
Linking of Blocks





Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.201

Merkle Trees and Transactions

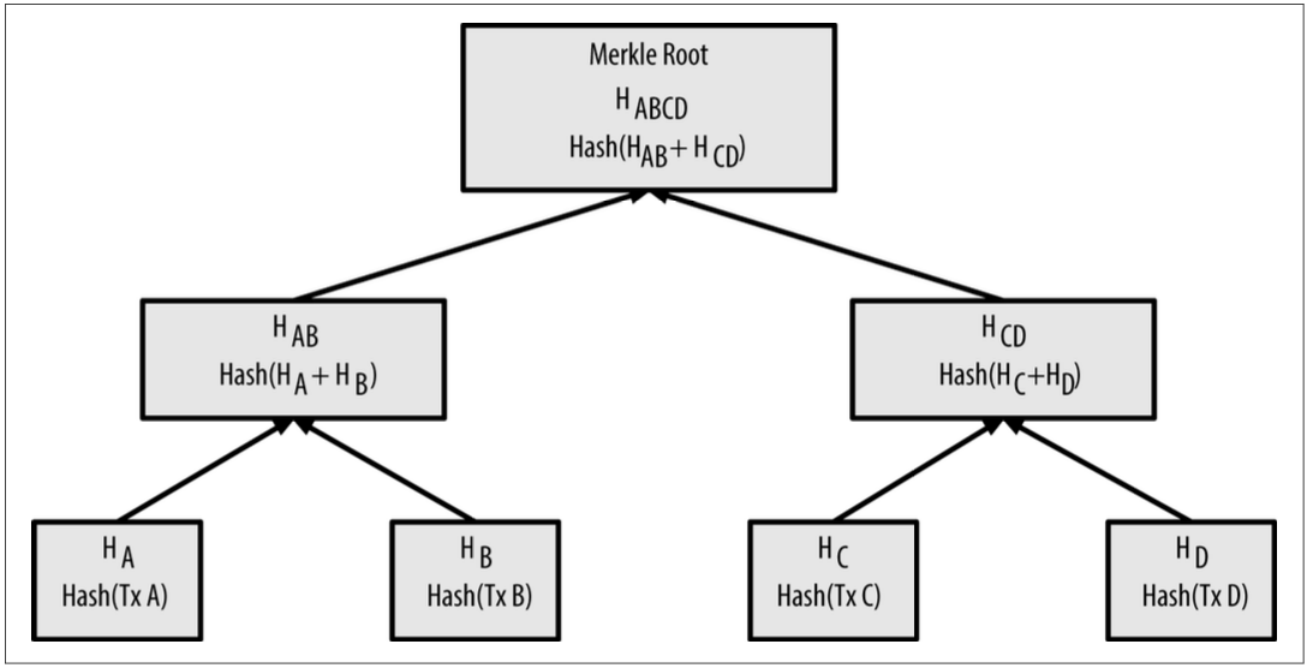


Many of the slides have been taken and adapted from:
Bitcoin and Cryptocurrency Technologies
with due credit to the generous authors of slides from:
http://bitcoinbook.cs.princeton.edu/

CBS

Transaction Hash Graph

- Merkle trees summarize all the transactions in a block, producing an overall digital fingerprint of the entire set of transactions,
- Provide a very efficient process to verify whether a transaction is included in a block.
- A Merkle tree is constructed by recursively hashing pairs of nodes until there is only one hash, called the root, or merkle root.
- The cryptographic hash algorithm used in bitcoin's merkle trees is SHA256 applied twice, also known as double-SHA256.



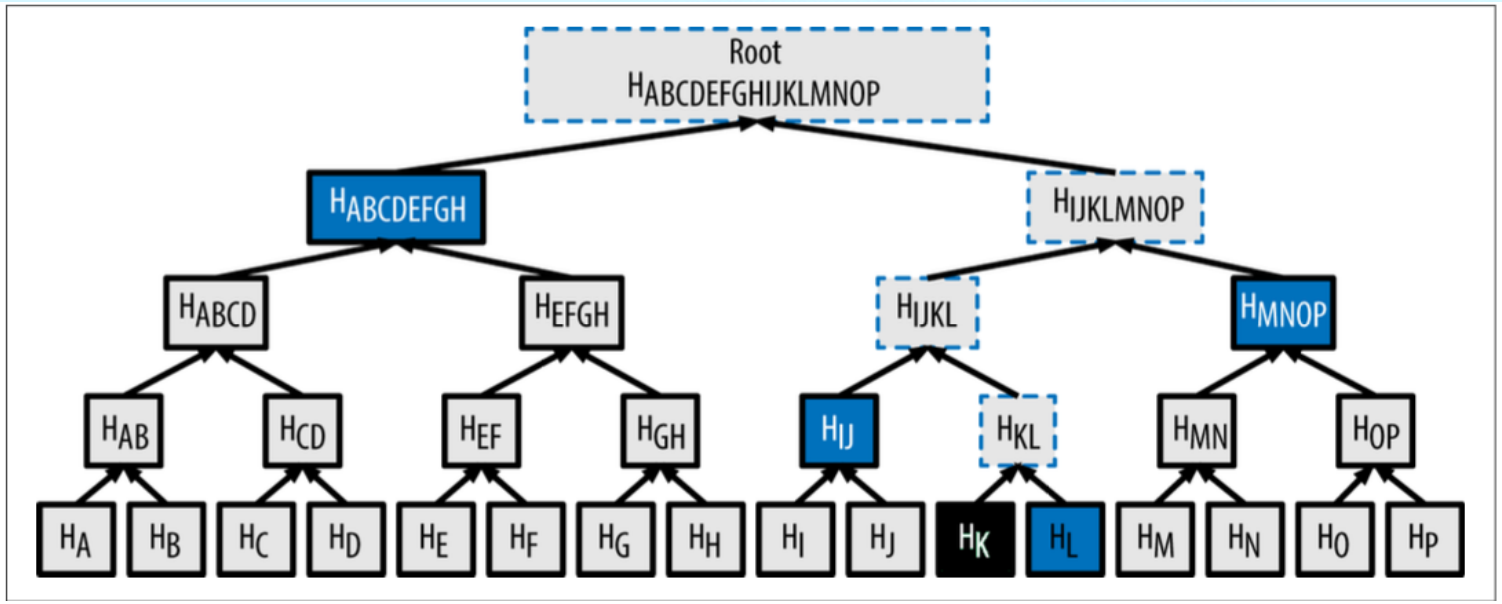
$$H_A = SHA256(SHA256(Transaction\ A))$$

$$H_{AB} = SHA256(SHA256(H_A + H_B))$$

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.201

Verifying a Transaction

- We want to prove that transaction K is included
- Provide path with 4 hashes: H_L, H_{IJ}, H_{MNOP}, $H_{ABCDEFGH}$
- We can compute pairwise hashes at each branch and prove that K exists in log(n) lookups.



Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.201

Verifying a Transaction

- As the number of transactions (also size) increases from 16 to 65535, then the path to verify a transaction only increases from 4 to 16 hashes.
- With merkle trees, a node can download just the block headers (80 bytes per block) and still be able to identify a transaction's inclusion in a block by retrieving a small merkle path from a full node, without storing or transmitting the vast majority of the blockchain, which might be several gigabytes in size.
- Nodes that do not maintain a full blockchain, called simplified payment verification (SPV) nodes, use merkle paths to verify transactions without downloading full blocks.

Table 9-3. Merkle tree efficiency

Number of transactions	Approx. size of block	Path size (hashes)	Path size (bytes)
16 transactions	4 kilobytes	4 hashes	128 bytes
512 transactions	128 kilobytes	9 hashes	288 bytes
2048 transactions	512 kilobytes	11 hashes	352 bytes
65,535 transactions	16 megabytes	16 hashes	512 bytes

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.206

Popular Blockchain Explorers

Popular blockchain explorers include:

- Bitcoin Block Explorer

  https://blockexplorer.com/txs/ce8ed0c7797eb29a2dd0229d83347acf61f3f7e9381f3aaebe4c9f88bd4273ab

- BlockCypher Explorer （https://live.blockcypher.com/）
- blockchain.info (https://blockchain.info)
- BitPay Insight (https://insight.bitpay.com)
- Viewing raw data of block:

replace block hash in https://blockchain.info/rawblock/[block_hash]

e.g.: https://blockchain.info/rawblock/00000000000000000000333d265acedd64e126788ce876b46450023bddb4e6a06

Antonopoulos, A. M. (2014). Mastering Bitcoin: unlocking digital cryptocurrencies. " O'Reilly Media, Inc.". p.16

Thank you!

Raghava Mukkamala

rrm.digi@cbs.dk

https://www.cbs.dk/staff/rrmdigi

https://raghavamukkamala.github.io/

https://cbsbda.github.io/