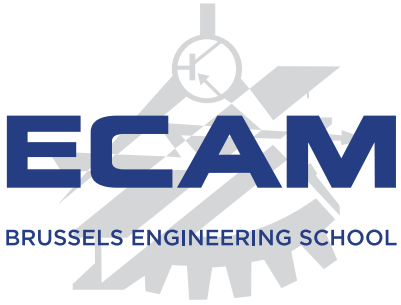


DB701μ Introduction to
Blockchain with Python

Session 2

Blockchain Transactions and Proof-of-Work Algorithm

Sébastien Combéfis

Winter 2020

Objectives

- Understand what is stored in the blocks of a blockchain

  Any kind of data, such as transactions for the Bitcoin

- Understand the structure of a block and how they are linked

  Index, timestamp, hash, previous hash, data and nonce

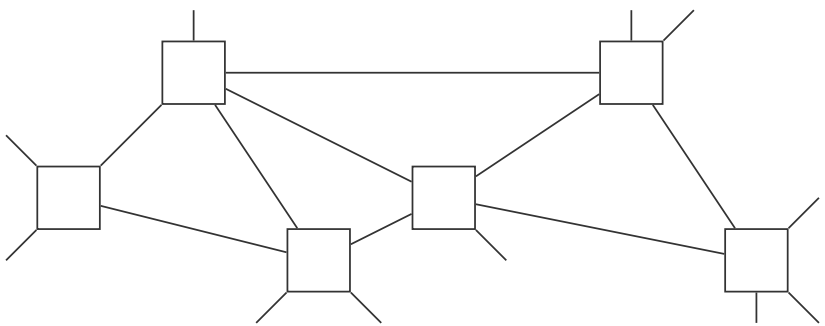
- Understand how nodes agree with Proof-of-Work algorithm

  A multiple validation process to ensure security of the blockchain

Transaction

Distributed Ledger

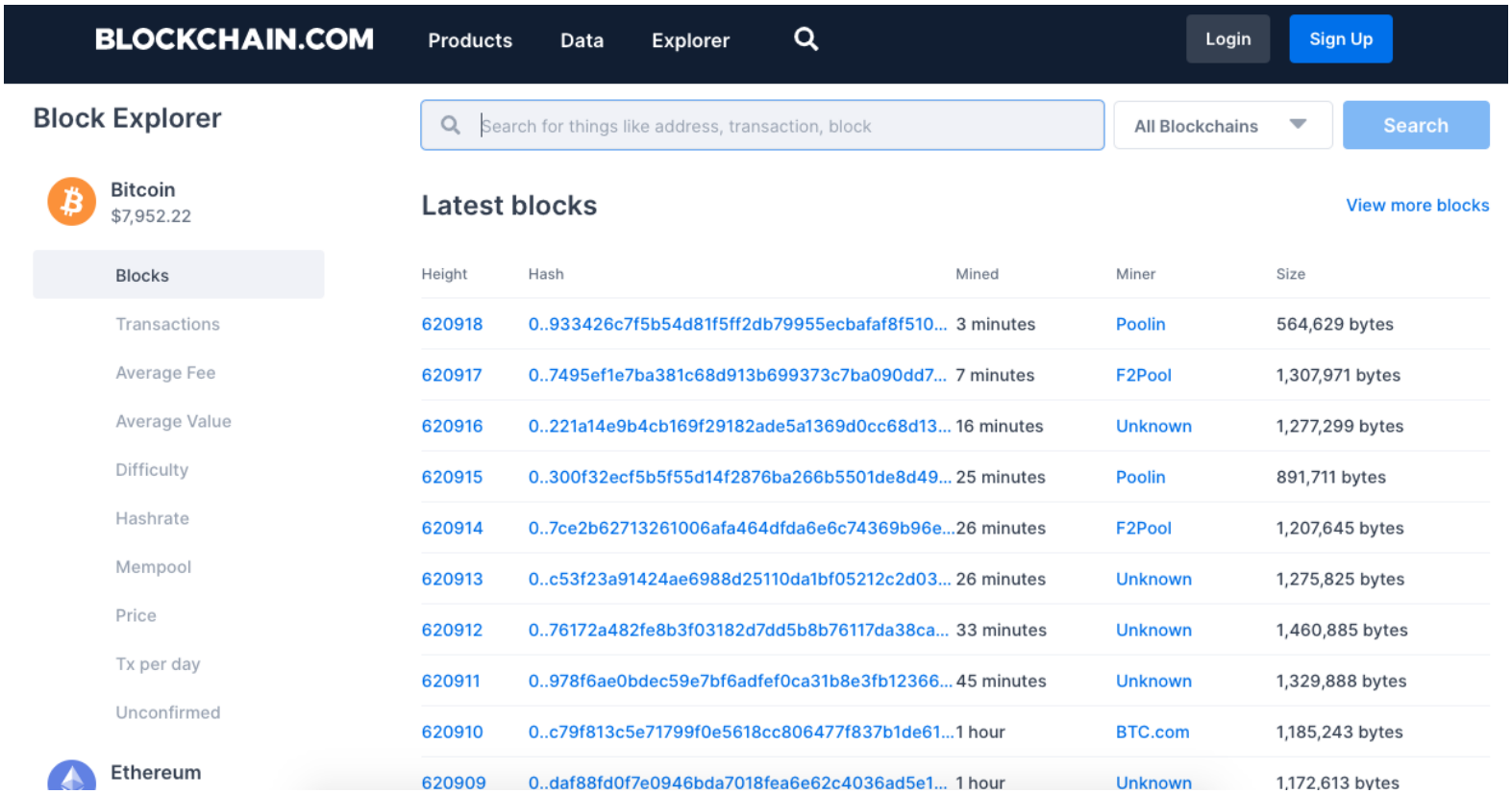
- A blockchain is a distributed ledger of data
  - Spread across the network among all the peers of the network
  - Each peer holds a copy of the ledger on its own machine
- A blockchain is a chain of blocks linked together
  - Starting from a first block called the Genesis block
  - Blocks are connected to each other in chronological order
  - Each block contains a bunch of data and some metadata



Blockchain Explorer

- Anyone can inspect the content of a public blockchain

  By downloading it locally of by using online websites



Transaction Request

- A blockchain stores transactions between its users

  A user of the system can make a transaction request

- Five main steps are executed to validate a transaction

  1. A node creates and digitally sign a transaction
  2. The transaction is propagated to other nodes validating it
  3. It is then included in a block propagated onto the network
  4. The next block is cryptographically linked to this block

Validation

- A transaction must be validated to be added to the ledger

  Otherwise it is simply rejected by the network

- Each transaction goes through several validation levels

  1. Nodes are validating locally each transaction that they receive
  2. Transaction confirmed once put in a block and propagated
  3. Second confirmation when a new block is added to the latter
  4. Transaction final when typically six other blocks added

Security

- Creating a digital signature for each transaction request
  - Transaction request encrypted with the user private key
  - Other nodes can check authenticity with user public key
- Changing the transaction request invalidates the signature

  To prevent an attacker to change the content of the transaction

Inside Block

Cryptographic Security

- The ledger is made tamper-proof with cryptography

  A lot of verification and protection is put in place

- Digital signature used to check authenticity of a transaction

  Protects the transaction request message and its source

- Cryptographic hash used to ensure the consistency

  Blocks belonging to the chain cannot be changed

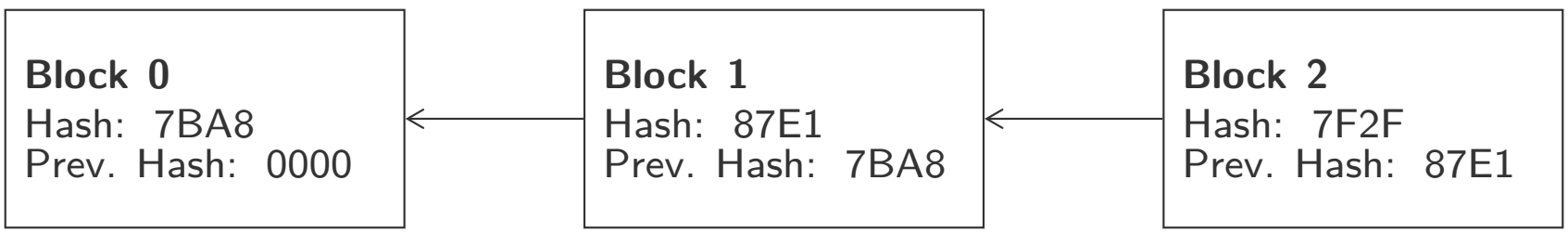
Block Content

- A block contains data, a hash and the hash of previous block

  Data depends on application: money, share, certificate, vote, etc.

- Hash computed on a string representation of the block content

  Changing the content of the block changes the value of the hash



Index

- Each block in the blockchain has an index as unique position

  Genesis block has the index 0, *the first linked block index* 1, *etc.*

- Index used to check that the blockchain has not been broken

  Just going through the blocks and validating their index

Timestamp

- Sometimes important to know the order of transactions

  For cryptocurrency, money cannot be spent if not available

- Possible to add a timestamp inside each transaction

  Not a good solution since it can be easily counterfeited

- A block is used to group several transactions together

  Transactions from a block happened at the same time

Consensus

Consensus

- Most critical attribute of blockchain technology is consensus

  Ability to update the decentralised ledger through consensus

- Validating a change in the blockchain through specific process
  - Set of strict criteria defined by the blockchain protocol
  - Need for a consensus between the participating nodes

Proof-of-Work

- Proof-of-Work (PoW) process to add new blocks to the chain

  Trying to guess a value by a brute-force algorithm

- Solving the problem is hard, but checking a solution is easy

  It takes about 10 minutes to be solved on a network of nodes

- Nodes from the network are racing to solve the problem

  Winner receives a reward and adds a block to the blockchain

Nonce

- A blockchain may impose some requirements on valid hashes

  It will defined the difficulty of the mathematical puzzle to solve

- Having three leading zeros can be a requirement, for example

  The number of zeros can be selected to change the difficulty

- The nonce is the value to find to have a valid hash
  - This is the value for which the nodes of the network are racing
  - The process of finding the nonce is referred to as mining

Mining

- The mining process consist in finding the nonce

  Nodes are dedicating computing resources to find it

- Different incentive for nodes finding the correct nonce
  - Block reward for the node that found the nonce and hash
  - Transaction fees to prioritise your insertion in a node

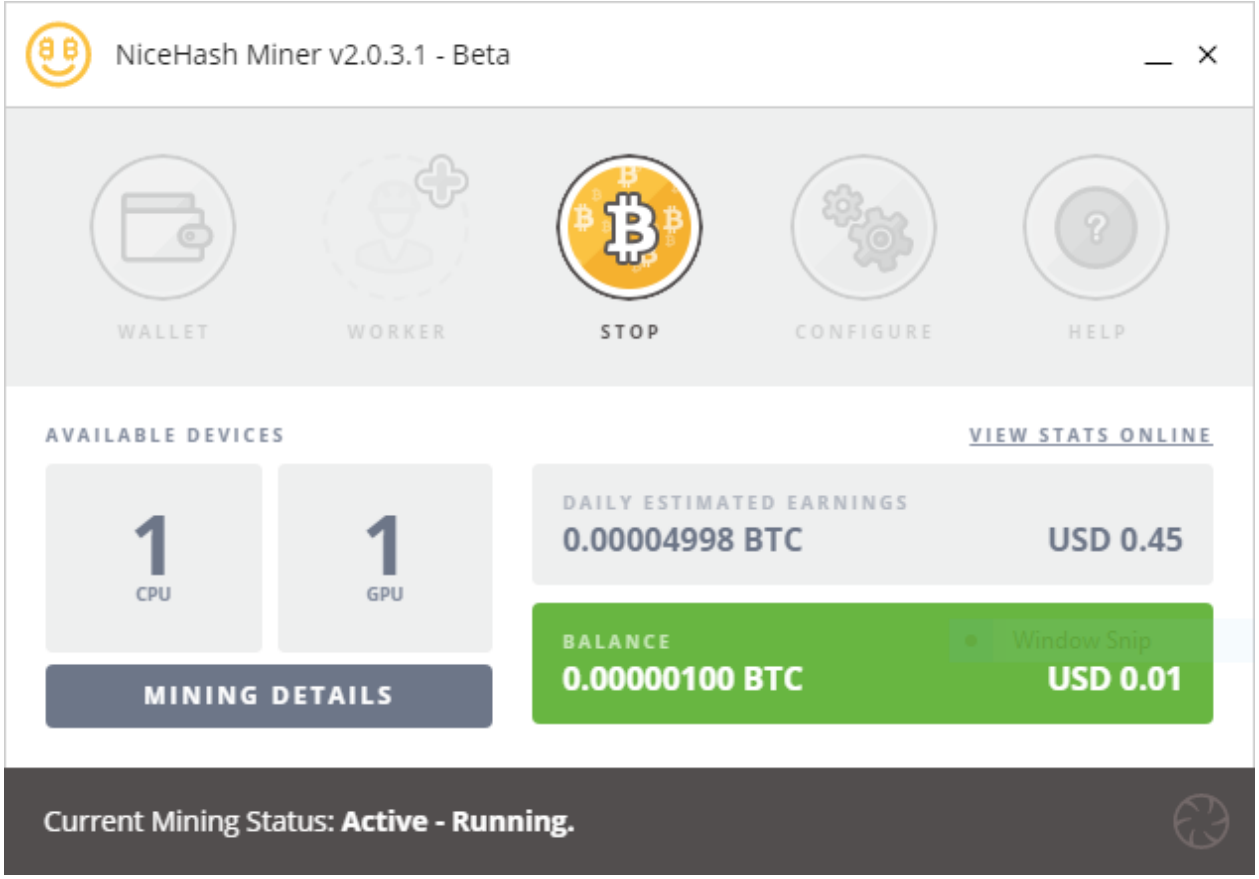
- Single mining is not possible anymore, need for pool mining

  People are buying hash power with distributed applications

NiceHash Miner

- NiceHash Miner is a tool to buy and sell hash power

  For example, 1 CPU with 25.7 MH/s, 1 GPU with 823.4 MH/s



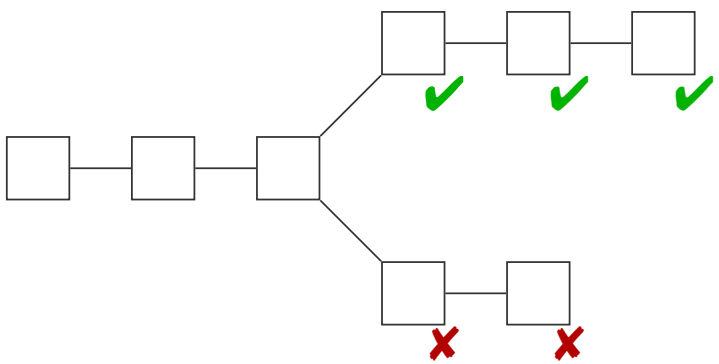
Fork

- A fork occurs if several nodes solve PoW at the same time

  Several blockchains are competing to be the valid one

- The only valid fork will be the longest chain

  Transactions from others rejected and sent for verification again



Validation Method

- Transaction validated by one node with Proof-of-Stake (PoS)
  - Lower energy consumption over the network
  - No reward as with PoW, but forger receives a transaction fee
- Many other methods can be used to perform validation

  Proof-of-Authority, ...-Burn, ...-Capacity, ...-Elapsed Time

References

- Jean-Luc Verhelst (2017). *Bitcoin, the Blockchain and Beyond: A 360-Degree onboarding guide to the first cryptocurrency and blockchain*, Self-Published, ISBN: 978-2-930-97100-1.
- Eden Au (2019). *Building a Minimal Blockchain in Python: Understanding blockchain by coding*, July 13, 2019. https://towardsdatascience.com/building-a-minimal-blockchain-in-python-4f2e9934101d
- Michele D'Aliessi (2016). *How Does the Blockchain Work? Blockchain technology explained in simple words*, June 1, 2016. https://onezero.medium.com/how-does-the-blockchain-work-98c8cd01d2ae
- Ibad Siddiqui (2018). *What The Hell Is Blockchain And How Does It Works? (Simplified)*, April 24, 2018. https://medium.com/coinmonks/what-the-hell-is-blockchain-and-how-does-it-works-simplified-b9372ecc26ef
- Tania H. (2018). *How the Blockchain Works*, January 4, 2018. https://rubygarage.org/blog/how-blockchain-works
- Blair Marshall (2018). *How are transactions validated?*, February 2, 2018. https://medium.com/@blairlmarshall/how-do-miners-validate-transactions-c01b05f36231
- Edzo Botjes (2017). *Pulling the Blockchain apart.. The transaction life-cycle*, August 11, 2017. https://medium.com/ignation/pulling-the-blockchain-apart-the-transaction-life-cycle-7a1465d75fa3

Credits

- Jimmy Emerson, DVM, July 13, 2012, https://www.flickr.com/photos/auvet/7662314238.
- F Delventhal, December 30, 2018, https://www.flickr.com/photos/krossbow/49575992783.
- Wilson Lam, October 9, 2015, https://www.flickr.com/photos/kitkit201/21951944510.