Blockchain architectures

A focus on consensus

Claudio Di Ciccio

c.diciccio@uu.nl
https://www.uu.nl/staff/CdiCiccio

Agenda for today

13:15 - 13:30:
Recap and Q&A

13:30 - 14:15:
Proof of Work and consensus

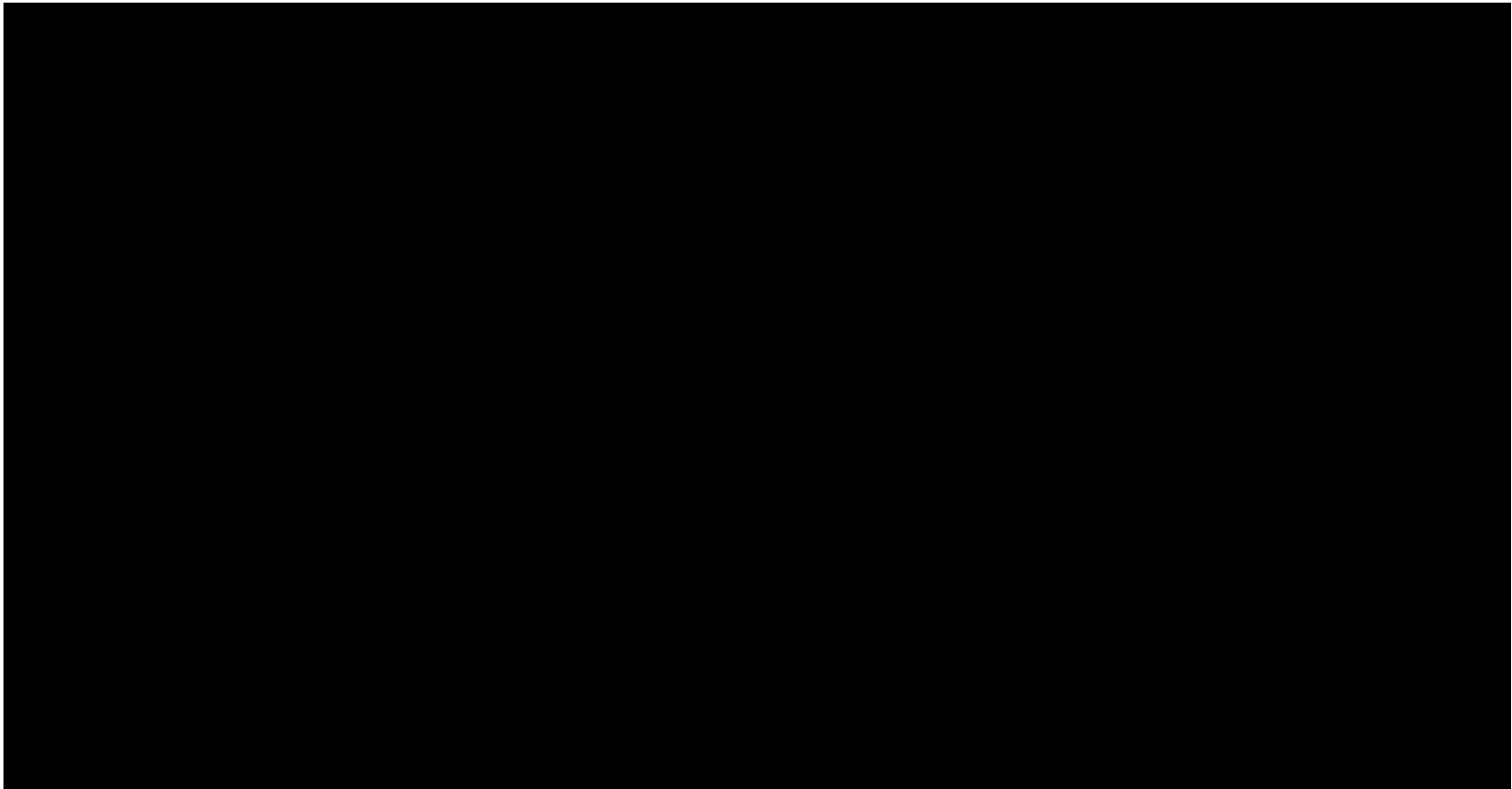
14:30 - 15:15:
Proof of Stake and consensus

15:30 - 17:00:
Architectural debate

Kudos to Jan Martijn for the picture

Utrecht University

Recap and Q&A



Core concepts

- Transactions → Transfer of assets (and code invocation)
- Signatures → Authentication
- Ledger → Transaction ordering
- Distributed architecture → Data persistency
- Hashing → Robustness
- Proof-of-[…] → Publishing rights
- Consensus → Eventual consistency
- Smart contracts → Programmability (and tokens / app coins)

Blockchain as a protocol [IES]

- In a telecommunications context, a protocol is a system of rules that describes how a computer (and its programmer) can
  - connect to,
  - participate in, and
  - transmit information over a system or network.
- We will call the computing systems in the network as nodes.
- These instructions define code syntax and semantics that the system expects.
- Protocols can involve hardware, software, and plain-language instructions.
  - Examples from everyday life in the web:
    - HTTP
    - TCP
    - SMTP
    - …
- The blockchain is *not* a software, a computer, or a software system
  - This is why different blockchains exist

Ledgers in conflict and the CAP theorem

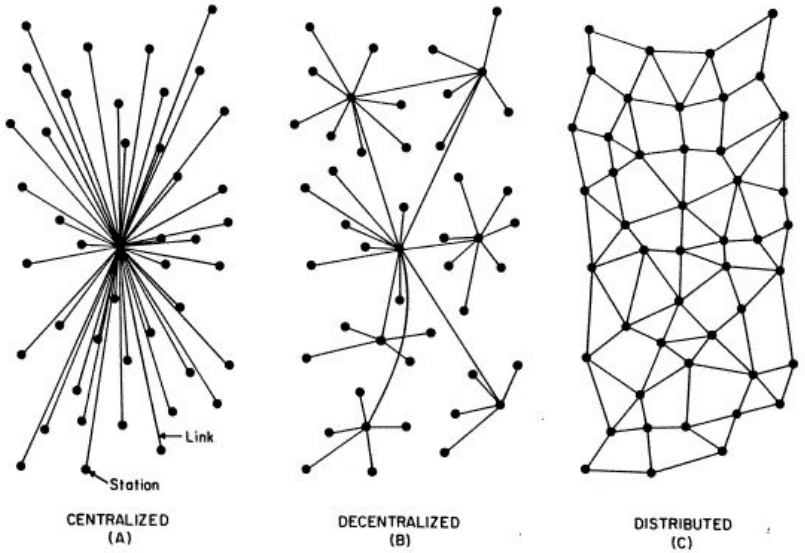
- Conflicts are not due to errors
- It is just a matter of propagation of the transactions in the network

- The CAP theorem states that out of the three following guarantees, only two can be guaranteed by a distributed data store
  - Consistency
    - Every read receives the most recent write or an error
  - Availability
    - Every request receives a (non-error) response (not necessarily the most recent write)
  - Partition tolerance
    - The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes
- The blockchain gives up on consistency

- The blockchain needs some state transitions before all nodes agree
  - Reaching consensus
- An Ethereum transaction is typically considered as final after around 12 blocks [e]

Centralized, decentralized, distributed

- Politically decentralized
  - No entity controls the network.
- Architecturally decentralized
  - No infrastructural central point of failure.
- Logically centralized
  - There is one commonly agreed state* and the system behaves like a single computer
- Distributed information
  - Every node* has access to the full history of transactions



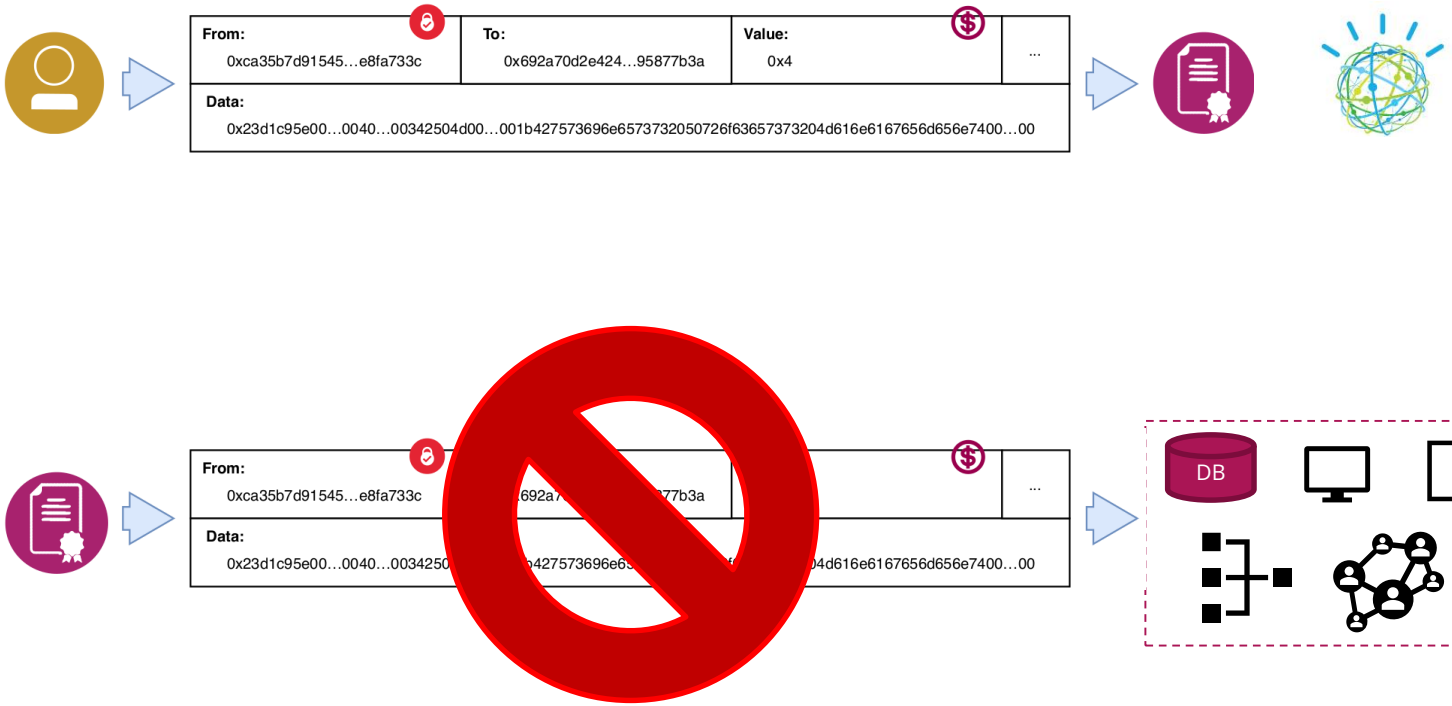
Utrecht University

Oracles: From on-chain to off-chain and vice versa

Blockchain technologies guarantee permancence and non-repudiability, but not truthfulness of the payloads

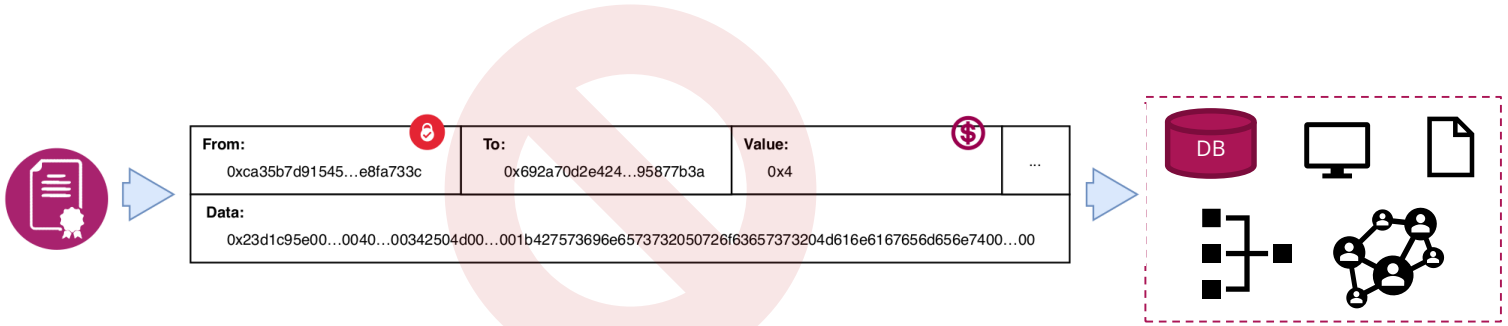
Image source: https://imgur.com/P25H4c0

The problem


From:
0xca35b7d91545…e8fa733c
To:
0x692a70d2e424…95877b3a
Value:
0x4
...
Data:
0x23d1c95e00…0040…00342504d00…001b427573696e6573732050726f63657373204d616e6167656d656e7400…00
From:
0xca35b7d91545…e8fa733c
...
Data:
DB

The Oracle

Source: http://matrix.wikia.com/wiki/File:The_Oracle_Making_Cookies.jpg

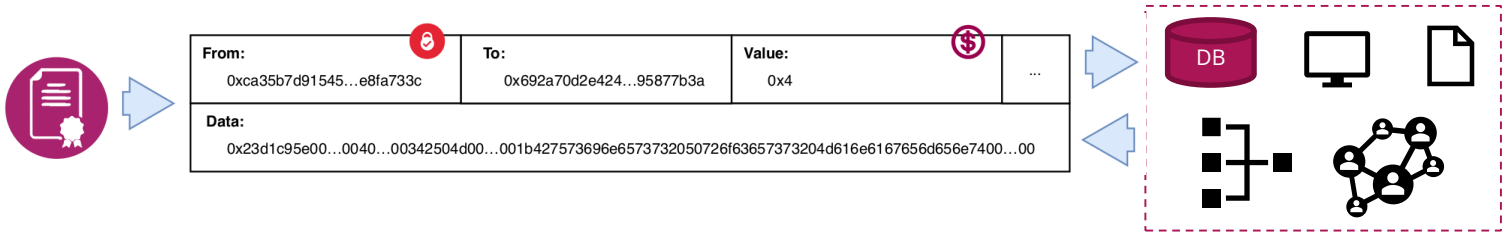


The Oracle

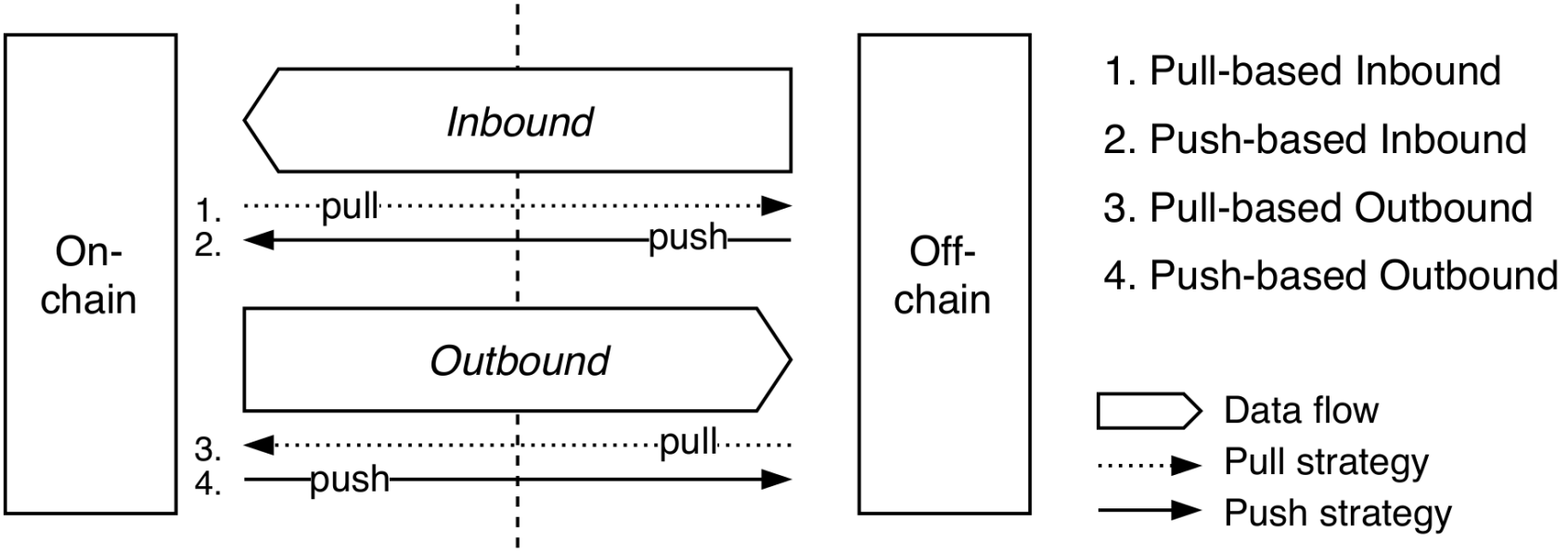
ISO/TC 307, ISO/TR 2345: “[A] DLT Oracle [is a] service that updates a distributed ledger using data from outside the distributed ledger system”. (2019)

Previous literature: oracles as off-chain information providers.

We see oracles as a bridge
between the on-chain and off-chain worlds.



Oracle patterns: Overview



Utrecht University

Consensus

Image © alchetron.com/John-D.-Rockerduck

Ledgers in conflict

- Conflicts are not due to errors
  - Just a matter of propagation of transactions in the network
- But conflicts can be exploited for double spending

Image © alchetron.com/John-D.-Rockerduck

Hashing

- Ideally, return
  - fully random (numeric) codes
  - for any value
    - except for values given before
      - same digest all the times then
- In practice: input/output (I/O)
  - I: (bit)string of any length (message)
  - O: fixed-length hash value (digest)
- No secret key
- All operations are public

Image © http://www.eadt.co.uk/news/days-gone-by-talk-of-ipswich-s-old-postal-service-provides-a-first-class-delivery-of-memories-1-4691350

Hashing and preimage resistance

- Ideally, return
  - fully random (numeric) codes
  - for any value
    - except for values given before
      - same digest all the times then
- In practice: input/output (I/O)
  - I: (bit)string of any length (message)
  - O: fixed-length hash value (digest)
- No secret key
- All operations are public

- SHA3-256("Hi there!")= 0xe10f7b08a108024dcdd178e4cf5a37a60afd2cc12fba6dd39fd0bd9bf3190925
- SHA3-256("Hello there!")= 0x2c71257ae32d532d6337e78b64a9810734a6a376199f4f1ef150b839abc0b01b
- SHA3-256("Blockchain")= 0x94074fd5892e84da500a78e4c02ff986c38815ad4063441a1caad310e89cf709
- SHA3-256(“blockchain")= 0x45740502697d57cbc7e6522372d3247adf1ab8f1cdb0cda1f20a022bf3e153d0

Proof of Work

- Right to publish the next block:
  if a computationally hard puzzle is solved
  - The solution is hard to be found, but easy to be verified

- Bitcoin PoW is built upon Hashcash (Adam Back, 1997):
  - The digest of the block (header) should be less than a certain value
  - In practice, the N Most Significant Bits (MSBs) should be equal to 0
  - The more the 0's, the harder
  - In fact, we say

- The header is fixed, except a numeric value (nonce, a word of 32 bits)
  which can be changed by the mining node to solve the puzzle

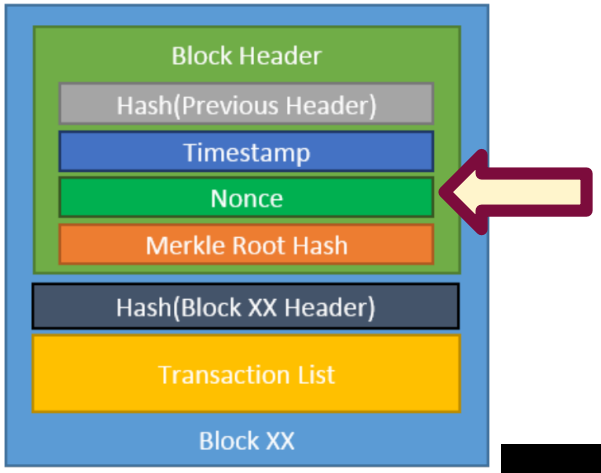


Image source: [NISTIR]

Proof of Work in Bitcoin [NISTIR]: An example

- Using the SHA-256 algorithm find the nonce n such that:
  - `SHA256(concat("blockchain", n)) = 0x`N such that N starts with **000000**
- Try!
  - `SHA256("blockchain0") =`
  - `0xbd4824d8ee63fc82392a6441444166d22ed84eaa6dab11d4923075975acab938`
    - Nope
  - `SHA256("blockchain1") =`
  - `0xdb0b9c1cb5e9c680dfff7482f1a8efad0e786f41b6b89a758fb26d9e223e0a10`
    - Nope
  - `SHA256("blockchain2") =`
  - `0x8f0532cd22055fb7599aa48f38501dcd46e61712ab49a02f840f5545830e9260`
    - Nope
  - `SHA256("blockchain3") =`
  - `0xeb61c3724d6da33605084d2d232bba0563cb82f4ad82c101b42f23c2e86277ef`
    - Nope
  - … 10,730,892 attempts later …
  - `SHA256("blockchain10730895") =`
    `0x000000ca1415e0bec568f6f605fcc83d18cac7a4e6c219a957c10c6879d67587`
    - At last!

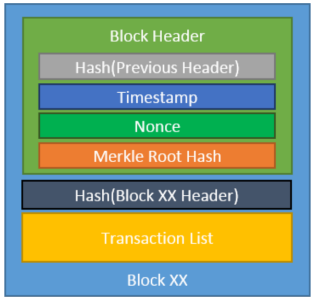
Proof of Work in Bitcoin [NISTIR]: An example

- Using the SHA-256 algorithm find the nonce n such that:
  - SHA256(concat("blockchain", n)) = 0xN such that N starts with 0000000
- Try!
  - SHA256("blockchain0") =
  - 0xbd4824d8ee63fc82392a6441444166d22ed84eaa6dab11d4923075975acab938
    - Nope
  - SHA256("blockchain1") =
  - 0xdb0b9c1cb5e9c680dfff7482f1a8efad0e786f41b6b89a758fb26d9e223e0a10
    - Nope
  - SHA256("blockchain2") =
  - 0x8f0532cd22055fb7599aa48f38501dcd46e61712ab49a02f840f5545830e9260
    - Nope
  - SHA256("blockchain3") =
  - 0xeb61c3724d6da33605084d2d232bba0563cb82f4ad82c101b42f23c2e86277ef
    - Nope
  - … 934,224,171 attempts later …
  - SHA256("blockchain934224174") =
    0x0000000e2ae7e4240df80692b7e586ea7a977eacbd031819d0e603257edb3a81
    - At last!

Proof of Work in Bitcoin [NISTIR]

Try it yourself:
andersbrownworth.com/blockchain/block
and
andersbrownworth.com/blockchain/blockchain

- Could we save results gained with leading "000000" to spare cycles on "0000000"?
  - No.
  - The block header changes every time, according to the contents of the block.
  - To stay with the metaphor, the prefix "blockchain" changes at every block!



- We can save with the *divide et impera* approach!
  - Distribute the work between more nodes to share the workload and rewards
    - Node 1: check nonce between 0 and 536870911
    - Node 2: check nonce between 536870912 and 1073741823
    - Node 3: check nonce between 1073741824 and 1610612735
    - Node 4: check nonce between 1610612736 and 2147483647
  - First solution found:
    - `SHA256("blockchain1700876653") = 0x00000003ba55d20c9cbd1b6fb34dd81c3553360ed918d07acf16dc9e75d7c7f1`
      after 90,263,918 attempts (less than previous 934,224,171)

Not the same nonce as before!

Image source: [NISTIR]

Proof of Work in Bitcoin (recap)

- Right to publish the next block:
  if a computationally hard puzzle is solved
- The solution is hard to be found, but easy to be verified

- Difficulty changes in Bitcoin every 2016 blocks (about 2 weeks)
  - The objective is, keep the average block time more or less stable
    - It should be 20160 minutes
    - This operation is named retarget

- The leading-zeroes-scheme illustrates that the block digest is less than or equal to a given target (e.g., 0x000000F…FF).
  - Example target (block 762048, 7 Nov 2022):
    0x00000000000000000**7a812**0000000000000000000000000000000000000000
- The digest is computed in Bitcoin with SHA-256
  - Demanding for processing, not for memory
  - ASICs (Application-specific Integrated Circuits) turn out to be very effective

Current target from: https://learnmeabitcoin.com/technical/target

Considerations and further readings

- The node who wins the puzzle gets a reward
  - Cryptos dug out of nowhere
  - Hence the name, miner!
- Playing by the rules pays off!

- What if two miners solve the puzzle with different blocks?
  - In Ethereum, also the runner-ups get a (lesser) reward
  - Uncles (a.k.a ommers)!

Picture: http://www.comicbookdb.com/graphics/comic_graphics/1/214/107268_20071001074129_large.jpg

Utrecht University

What the fork!

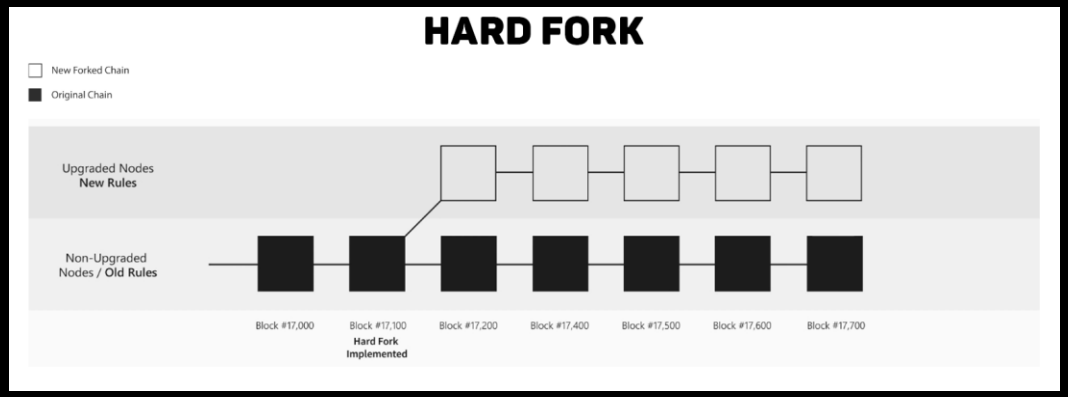


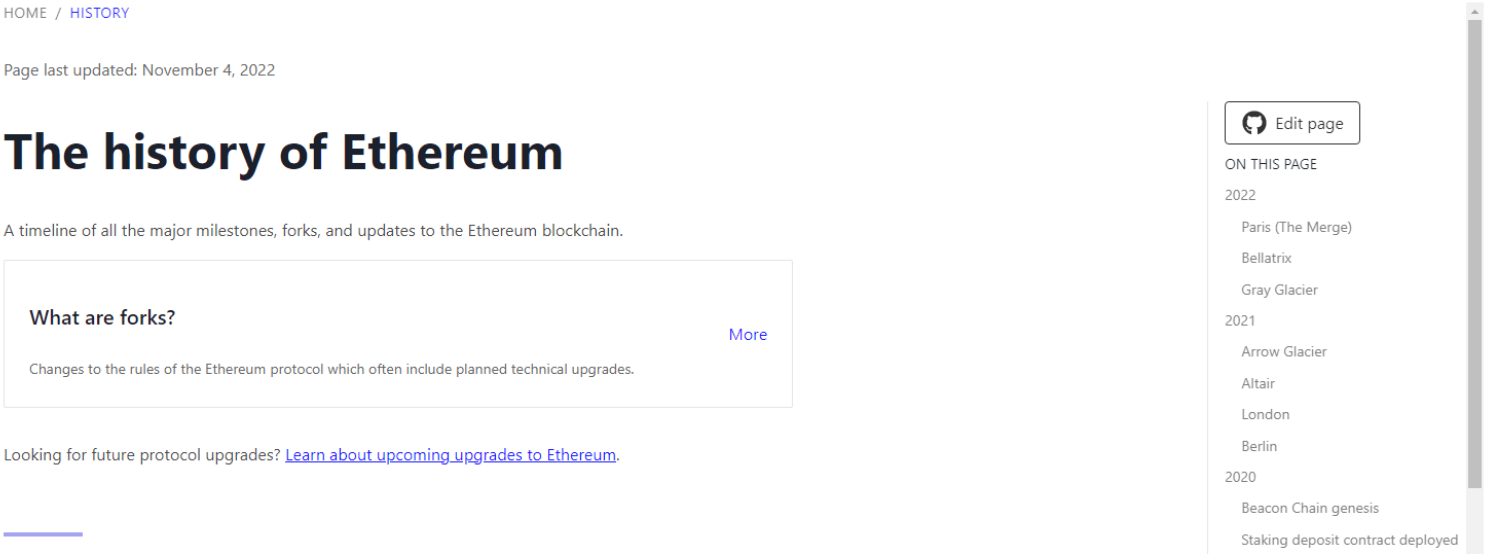
Image source: https://www.quora.com/How-does-forking-work-on-Ethereum

When does forking occur?

1. Software or protocol updates
   - In Ethereum, e.g.: Ethereum Improvement Proposals https://github.com/ethereum/EIPs

- Two main classes of fork (type 1)
  - Soft fork
    - Nodes that do not adopt the change can still access the blockchain (backward-compatible)
      - E.g., when Bitcoin introduced time-locked refunds [NISTIR]
      - Clients that do not implement the change in the protocol can ignore the new procedure
  - Hard fork
    - Clients that do not adapt to the change cannot access the new blockchain
      - They continue with the old one
      - E.g., the fork of Bitcoin Cash introduced a new cryptocurrency (BCH)
        - 2017, August the 1st, block 478558
        - For each bitcoin (BTC), an owner got 1 Bitcoin Cash (BCH)

More forks are yet to come

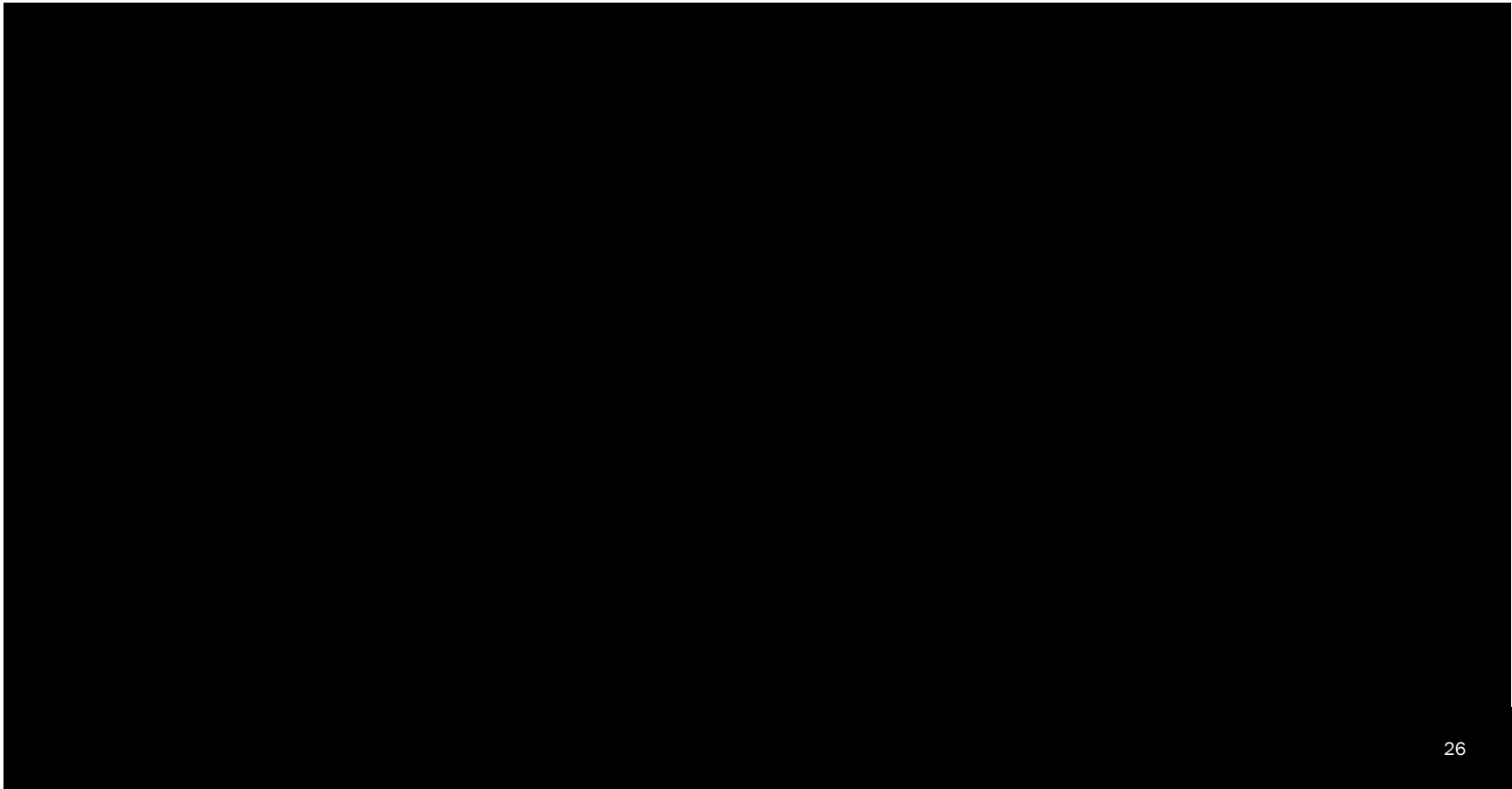


Cryptographic Algorithm	Type	Purpose	Impact from Large-Scale Quantum Computer
AES	Symmetric key	Encryption	Larger key sizes needed
SHA-2, SHA-3	N/A	Hash functions	Larger output needed
RSA	Public key	Signatures, key establishment	No longer secure
ECDSA, ECDH (Elliptic Curve Cryptography)	Public key	Signatures, key exchange	No longer secure
DSA (Finite Field Cryptography)	Public key	Signatures, key exchange	No longer secure

Image source: top: https://ethereum.org/en/history/; bottom: [NISTIR]

Utrecht University

Ethereum and Proof of Stake



Apropos PoW in Ethereum

- Improvement over Bitcoin: the Ethash algorithm
  - Ethereum protocol's defense against mining hardware optimization.
  - A memory-hard algorithm that can't be brute-forced with a custom application-specific integrated circuit (ASIC)
    - popular with Bitcoin mining enterprises
    - "One plague of the Bitcoin world is ASICs" [yp]
- Key to this algorithm memory-hardness is its reliance on a directed acyclic graph (DAG) file, which is essentially a 1 GB dataset created anew every 125 hours, or 30,000 blocks
  - This period of 30,000 blocks is also known as an epoch
- Shorter block time: approx. 15 sec
  - Better for computation!
  - But more miners guess the right nonce!

From PoW to next generation consensus protocol

- Mining pools / mining rigs growing
  - Huge mining power → lack of democracy
  - Let alone the electricity consumption
    - Nowadays entire states' consumptions are comparably expensive
      - Have a look:
        - https://ccaf.io/cbnsi/cbeci
- The Ethereum community planned an "end date" for PoW
  - More than a date, a Terminal Total Difficulty
- Then what?

Rank	Country and Region	Population (Millions) [26]	Energy (TWh)[23, 27, 28, 29]]	Share (%)
0	World	7,878.2	23,398.00	100.00
1	China	1,444.9	7,500.00	32.05
2	U.S.A	332.9	3,989.60	17.05
3	India	1,366.4	1,547.00	6.61
20	Taiwan	23.8	237.55	1.01
21	Vietnam	98.2	216.99	0.92
22	South Africa	60.1	210.30	0.89
23	Bitcoin + Ethereum	N.A.	190.13	0.81
24	Thailand	69.9	185.85	0.79
25	Poland	37.80	153.00	0.65
26	Egypt	104.3	150.57	0.64
27	Malaysia	3.1	147.21	0.62
28	Bitcoin	N.A.	135.12	0.57
29	Sweden	10.2	131.79	0.56
49	Switzerland	8.7	56.35	0.24
50	Ethereum	N.A.	55.01	0.24
51	Romania	19.1	55.00	0.23

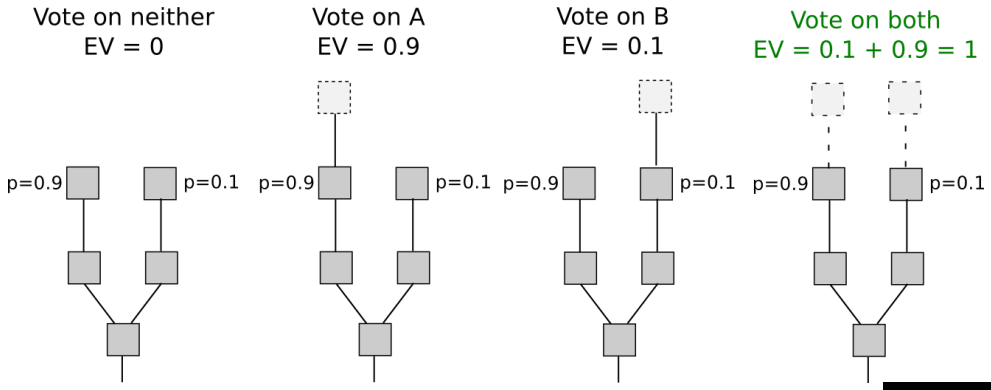
Rank	Country and Region	Population (Millions) [26]	Emission ($MtCO_2$)	Share (%)
0	World	7,878.2	37,077.40	100.00
1	China	1,444.9	10,060.00	27.13
2	U.S.A	332.9	5410.00	14.59
3	India	1,336.4	2,300.00	6.2
38	Nigeria	211.3	104.30	0.28
39	Czech Republic	10.7	100.80	0.27
40	Belgium	11.6	91.20	0.24
41	Bitcoin + Ethereum	N.A.	90.31	0.24
42	Kuwait	4.3	87.80	0.23
43	Qatar	2.9	87.00	0.23
49	Oman	5.2	68.80	0.18
50	Bitcoin	N.A.	64.18	0.17
51	Greece	10.3	61.60	0.16
76	Tunisia	11.94	26.20	0.07
77	Ethereum	N.A.	26.13	0.07
78	SAR	17.9	25.80	0.06

Terminal Total Difficulty: 58 750 000 000 000 000 000 000

V. Kohli, S. Chakravarty, V. Chamola, K. S. Sangwan, S. Zeadally, An analysis of energy consumption and carbon footprints of cryptocurrencies and possible solutions, 9(1): 79-89 (2023)
https://doi.org/10.1016/j.dcan.2022.06.017

Proof-of-Stake

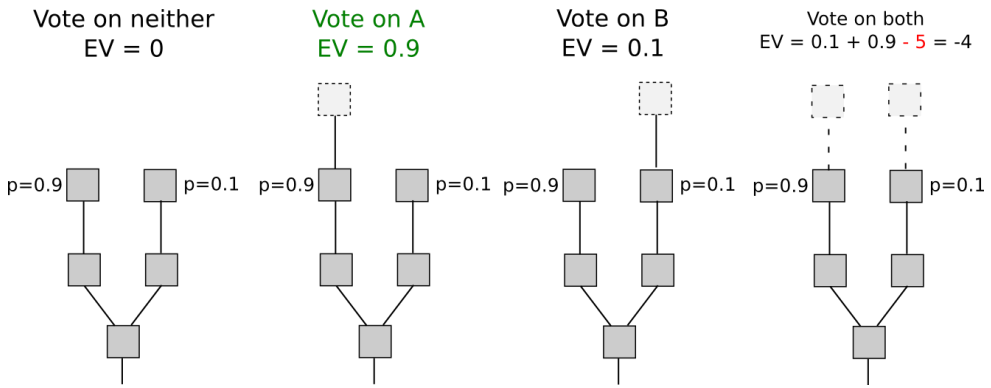
- Alternative consensus
  - Stake is an amount of crypto-commodities invested into the system
    - Locked via a special transaction, or
    - sent to a specific address
  - Cannot be spent
- Rationale
  - The more is left at stake by users,
    the less likely they would want to subvert the blockchain
  - Validators place their “bet” on the chain(s) they deem true
  - If the block is appended, validators get a reward proportional to the bet
- No block mining (puzzle solving) reward
  - But also nothing to lose!
- Issue: Nothing at stake
  - Place your bet on all chains
  - “Tragedy of the commons”



Source: https://eth.wiki/en/concepts/proof-of-stake-faqs

Casper the Friendly Finality Gadget

- Based upon PoS
  - The Ethereum Improvement Proposal (EPI) 1011 described a hybrid to transition from PoW to PoS: https://github.com/ethereum/EIPs/blob/master/EIPS/eip-1011.md
- Introducing slashing
  - Some of validator's deposit is burnt with immediate logout from validators set
    if two conflicting votes are raised that violate slashing conditions
    - https://arxiv.org/abs/1710.09437
- First implemented release: 2018, May the 8th
  - https://github.com/ethereum/casper/releases



Source: https://eth.wiki/en/concepts/proof-of-stake-faqs

2020: Towards Ethereum 2.0

Ethereum Individuals Developers Enterprise

HOME / ETH2

Page last updated: October 28, 2020

Ethereum 2.0 (Eth2)

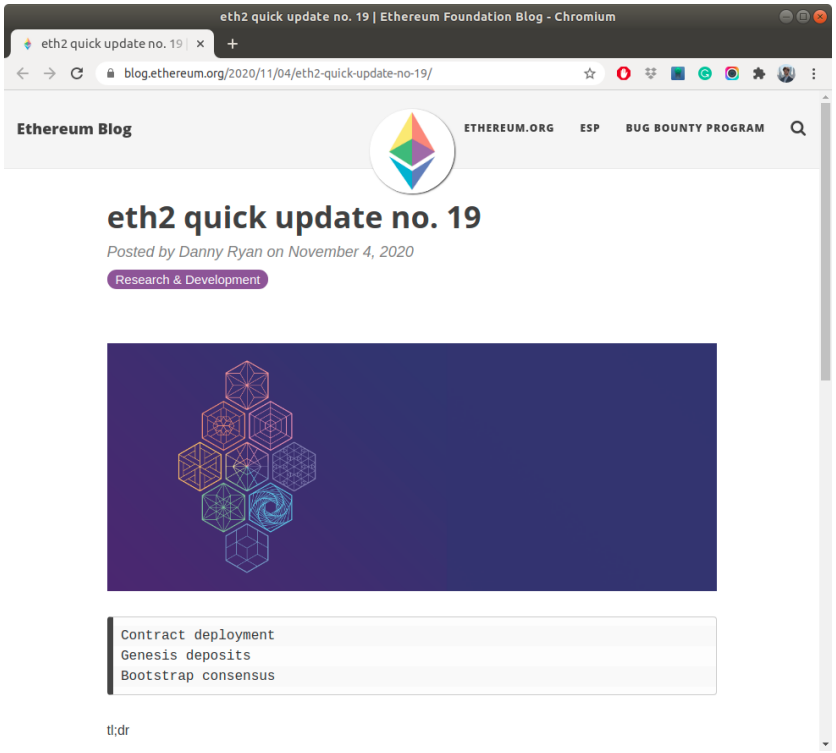
Eth2 is a long-planned upgrade to the Ethereum network, giving it the scalability and security it needs to serve all of humanity. The first stage of Eth2, called Phase 0, is planned to launch in 2020.

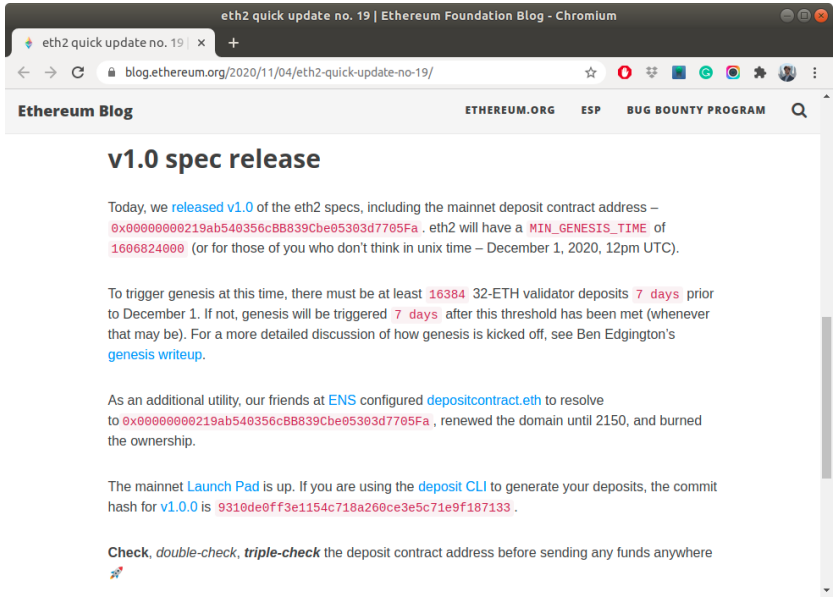
Eth2 will reduce energy consumption, allow the network to process more transactions, and increase security. Technically speaking, Ethereum will become a proof-of-stake blockchain and introduce shard chains. This is a huge change to how Ethereum works and it should bring equally huge benefits.

But it's only a change to Ethereum's infrastructure. If you're already an ETH holder, dapp user or dapp developer, you don't need to do anything because Eth2 will be compatible with the main Ethereum network you use today. **You'll be able to use the ETH you own today in Eth2 too.**

Source: https://ethereum.org/

2020: Towards Ethereum 2.0

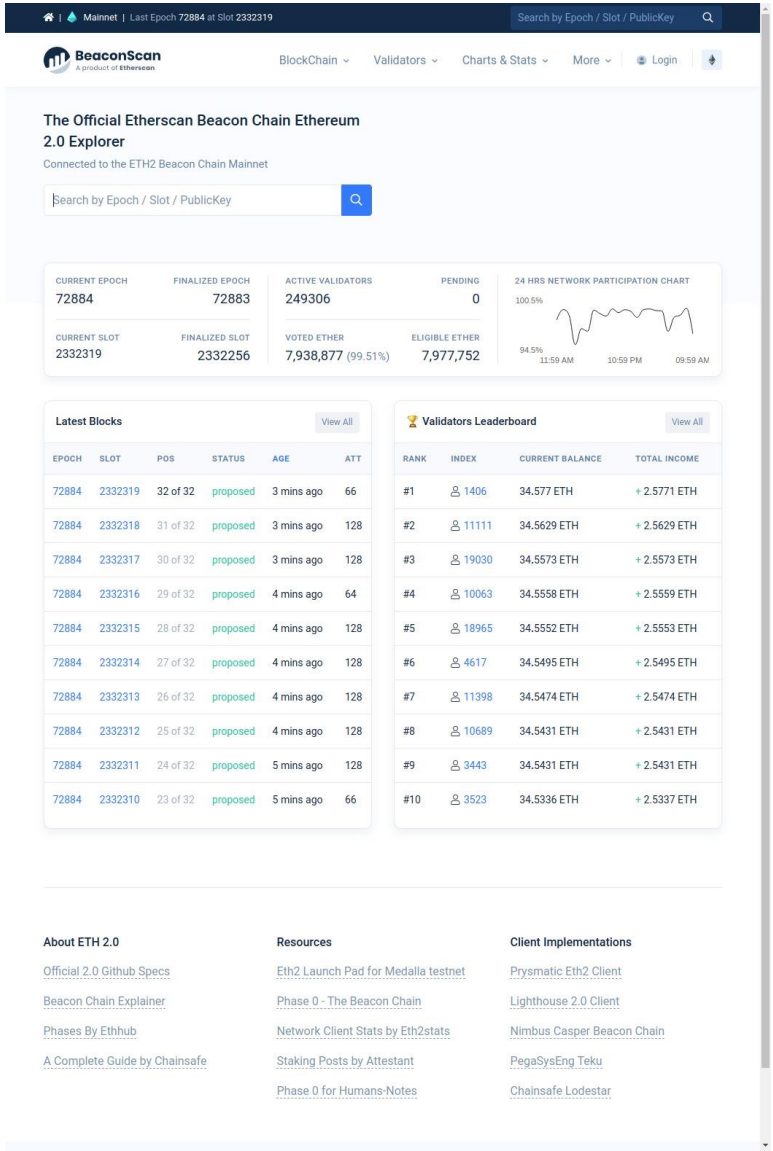




https://blog.ethereum.org/2020/11/04/eth2-quick-update-no-19/. Accessed: 2020-11-09 12:14:20

The beacon and the shards

- The beacon is a proof-of-stake chain
- The beacon acts as a synchronizing backbone for 64 shards (side-chains), which roll-up with it
  - To be included in “Phase 2”
- The beacon was already up and open for stakes (see https://beaconscan.com/)
- Planned for a merge with the previous, PoW-chain
  - Done (see next slide!)



For more info: https://ethereum.org/en/eth2/ (ETH 2.0), https://ethresear.ch/t/phase-one-and-done-eth2-as-a-data-availability-engine/5269/8 (Phase 2)

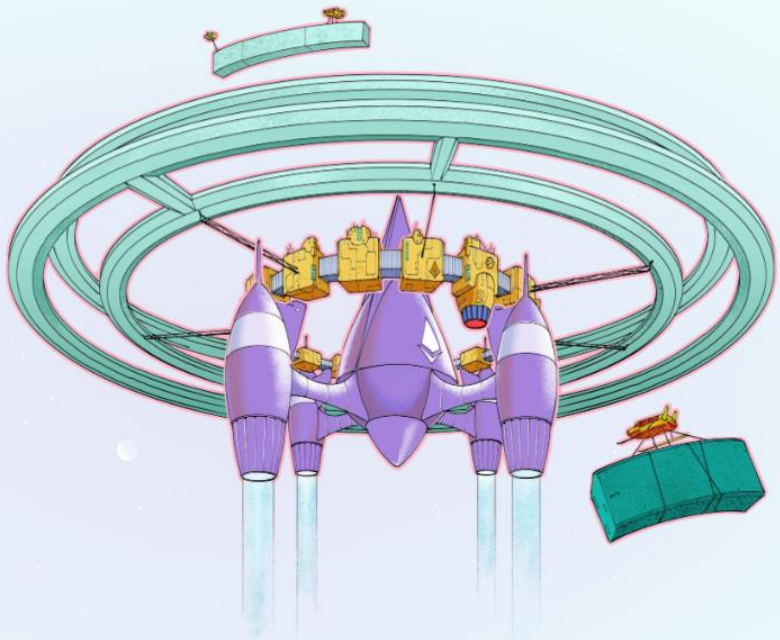
15 September 2022: The Merge

UPGRADES / MERGE

The Merge

- Ethereum Mainnet uses proof-of-stake, but this wasn't always the case.
- The upgrade from the original proof-of-work mechanism to proof-of-stake was called The Merge.
- The Merge refers to the original Ethereum Mainnet merging with a separate proof-of-stake blockchain called the Beacon Chain, now existing as one chain.
- The Merge reduced Ethereum's energy consumption by ~99.95%.

Page last updated: October 4, 2022



Terminal total difficulty at block 15537393

Shipped!

The Merge was executed on September 15, 2022. This completed Ethereum's transition to proof-of-stake consensus, officially deprecating proof-of-work and reducing energy consumption by ~99.95%.

https://ethereum.org/en/upgrades/merge/ (accessed: 2022-10-05)

Proof of Stake in Ethereum

- In Proof of Work, miners put capital at risk by expending energy
- In Proof of Stake, validators explicitly stake capital in ETH
  - 32 ETH are put at stake
    - Deposited in a dedicated smart contract
    - Not quite a small amount!
      Approx. € 45,000 (https://www.coinbase.com/converter/eth/eur, Sep. 2022)
    - Staking pools allow for consortia of stakers who hold less than 32 ETH
  - Can be slashed because of reiterated malfunction or misbehaviour
    - If the balance drops to 16 ETH or less, a forceful exit is triggered

- The validator is responsible for
  - checking that propagated blocks are valid
    (not only the head, i.e., the most recent block – see later) and
  - (occasionally) creating and propagating new blocks

- The validators send so-called attestations (votes)
  for a block they deem valid across the network.

The post-merge Ethereum block

Beacon block

Field	Description
randao_reveal	a value used to select the next block proposer
eth1_data	information about the (staking) deposit contract
graffiti	arbitrary data used to tag blocks
proposer_slashings	list of validators to be slashed
attester_slashings	list of validators to be slashed
attestations	list of attestations in favor of the current block
deposits	list of new deposits to the deposit contract
voluntary_exits	list of validators exiting the network
sync_aggregate	subset of validators used to serve light clients, with their aggregate signature for the previous block
execution_payload	transactions passed from the execution client

Zooming in the attestation field

Field	Description	
aggregation_bits	a list of which validators participated in this attestation	
data	slot	the slot the attestation relates to
	index	indices for attesting validators
	beacon_block_root	the root hash of the Beacon block containing this object
	source	the last justified checkpoint
	target	the latest epoch boundary block
signature	aggregate signature of all attesting validators	

"The fields inside the execution_payload reflect the block structure outlined in the Ethereum yellow paper, except that there are *no ommers* and prev_randao exists in place of difficulty."

Source: https://ethereum.org/en/developers/docs/blocks/ and https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/block-proposal/

The post-merge Ethereum block: execution payload

Execution payload header

Field	Description
parent_hash	hash of the parent block
fee_recipient	account address for paying transaction fees to
state_root	root hash for the global state after applying changes in this block
receipts_root	hash of the transaction receipts trie
logs_bloom	data structure containing event logs
prev_randao	value used in random validator selection
block_number	the number of the current block
gas_limit	maximum gas allowed in this block
gas_used	the actual amount of gas used in this block
timestamp	the block time
extra_data	arbitrary additional data as raw bytes
base_fee_per_gas	the base fee value
block_hash	Hash of execution block
transactions_root	root hash of the transactions in the payload

Execution payload

Field	Description
Same as the payload header, except the last 2 fields:	
withdrawals	list of withdrawal objects
transactions	list of transactions to be executed

Zooming in the `withdrawal` field (only for amounts exceeding 32 ETH)

Field	Description
address	account address that has withdrawn
amount	withdrawal amount
index	withdrawal index value
validatorIndex	validator index value

Staking withdrawals were enabled with the Shanghai/Capella upgrade which took effect on April 12, 2023.

Source: https://ethereum.org/en/developers/docs/blocks/

Rewards

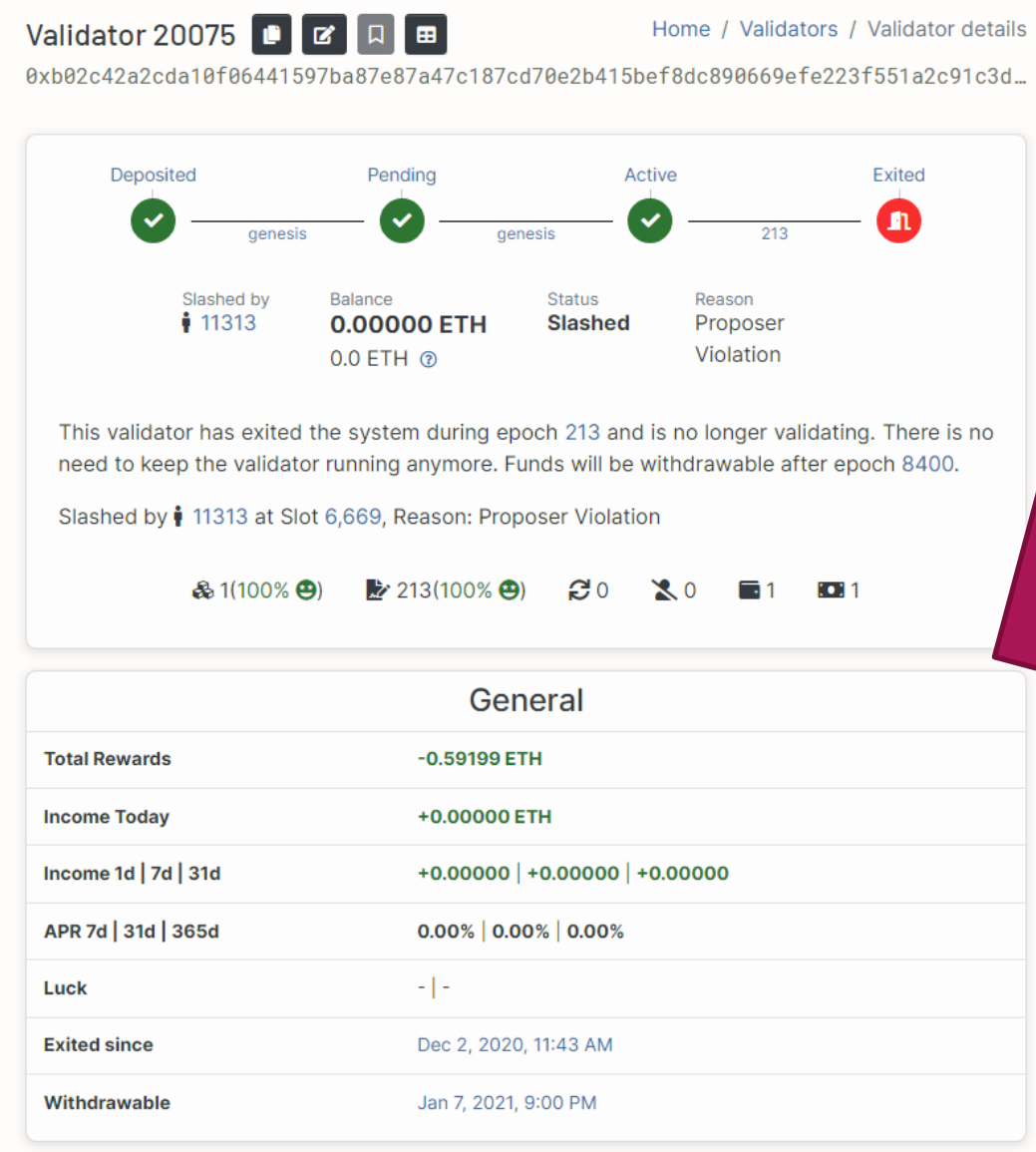
- Rewards
  - Attestation rewards
    (bound to timeliness: the fewer slots it takes, the higher the reward)
    - Correct head attestation
    - Correct (checkpoint) source attestation
    - Correct (checkpoint) target attestation
  - Sync committee reward
  - Proposer reward for an attested block with
    - head inclusion,
    - source inclusion,
    - target inclusion, and
    - sync-committee output

- For more information:
  - https://eth2book.info/altair/part2/incentives/rewards
  - https://consensys.net/blog/codefi/rewards-and-penalties-on-ethereum-20-phase-0/

Slashing

- Dishonest behaviours are prone to slashing:
  1. Double proposal: A proposer signs two different beacon blocks for the same slot
     - If the proposer does not propose a block, it misses the chance. No slashing.
  2. FFG double vote: A validator signs two differing attestations for the same target checkpoint epoch
  3. An attester signs a checkpoint attestation that “surrounds” another one
     - Let a validator cast two FFG votes.
       The epoch of the first vote’s source block precedes the epoch of the second vote’s source block;
       however, the epoch of the first vote’s target block follows the epoch of the second vote’s target block
     - This would cause a situation where the attester is contradicting what a validator already said was finalised in a previous attestation

The first slash: a proposal violation (Dec. 2020)



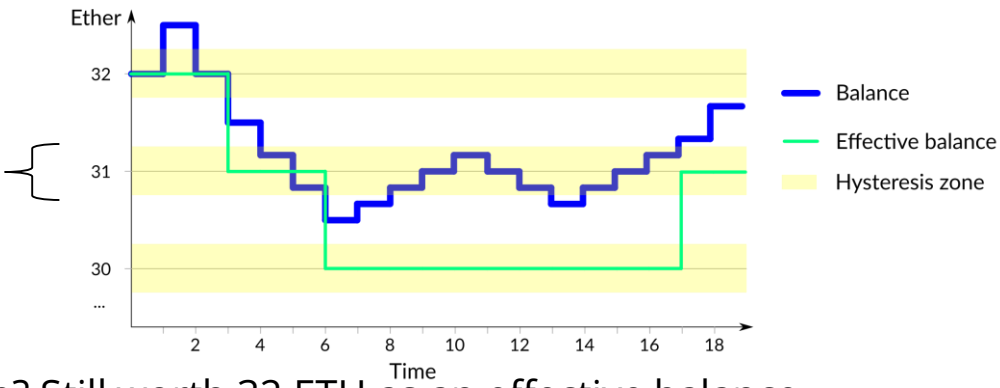
The likely reason was: the validator was running two instances of the same validator program

Source: https://beaconcha.in/validator/20075#blocks
For more information: https://medium.com/stakefish/ethereum-2-0-the-first-slash-a-retrospective-99e4fdcd563a

Considerations about the capital at stake

- In PoS, validators explicitly stake capital in ETH
  - 32 ETH are put at stake and varies over time as a validator balance
- Does putting more than 32 ETH give extra decision power or rewards?
  - Tl;dr: no
  - Please welcome the…

Hysteresis zone
(the effective balance does not change here)



- … Effective balance
  1. Never more than 32 ETH
     - 100 ETH validator balance? Still worth 32 ETH as an effective balance.
  2. Always an integer (no, it's just not about rounding! See next)
     - 29.7 ETH validator balance? Worth 29 ETH as an effective balance.
  3. Increases if raised by 1.25 or more ETH than the current effective balance
     - 25 ETH as effective balance? Need not less than 26.25 ETH to get to 26.
  4. Decreases if shrunk by 0.25+ ETH than the current effective balance
     - 25 ETH as eff. balance and val. balance down to 24.74 ETH? 24 ETH eff.
- Validator effectiveness: $^{\text{eff\_balance}}/_{\text{val\_balance}}$ (e.g., $^{29}/_{29.7} \cong 97.54\%$)

Source: https://www.attestant.io/posts/understanding-validator-effective-balance/

Bibliography

- Acronyms in square brackets indicate the reference
  - [NISTIR] Yaga, D., Mell, P., Roby, N., Scarfone, K. *Blockchain Technology Overview*. NISTIR 8202. https://doi.org/10.6028/NIST.IR.8202
  - [ABA] Xiwei Xu, Ingo Weber, Mark Staples: Architecture for Blockchain Applications. Springer 2019, ISBN 978-3-030-03034-6, pp. 1-307
  - [MB] Antonopoulos, A. M. Mastering Bitcoin: Programming the open blockchain. O'Reilly 2017. ISBN: 978-1-491-95438-6
  - [ME] Antonopoulos, A. M., Wood, G. Mastering Ethereum: Building Smart Contracts and DApps. O'Reilly 2017. ISBN: 978-1-491-97194-9
  - [btc] Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*. https://bitcoin.org/bitcoin.pdf
  - [wp] Buterin, V. *A Next-Generation Smart Contract and Decentralized Application Platform.* https://github.com/ethereum/wiki/wiki/White-Paper
  - [yp] Wood, G. *Ethereum: A secure decentralised generalised transaction ledger*. https://ethereum.github.io/yellowpaper/paper.pdf
  - [IES] Dannen, C. *Introducing Ethereum and Solidity. Foundations of Cryptocurrency and Blockchain Programming for Beginners*. Apress. ISBN: 978-1-4842-2535-6
  - [e] Diedrich, H. *Ethereum: Blockchains, Digital Assets, Smart Contracts, Decentralized Autonomous Organizations*. Wildfire Publishing. ISBN: 978-1523930470
  - [BDLT] Di Ciccio, C. Blockchain and Distributed Ledger Technologies. In: Leo, S., Panetta, I.C., The Role of Distributed Ledger Technology in Banking. Cambridge (in print)

Contents are mainly based on these publications. They are not expected to be all studied. Paying attention during the class is PARAMOUNT. Don't read me, I am just a chatty slide. listen to the instructor now!

Agenda for today

13:15 - 13:30:
Recap and Q&A

13:30 - 14:15:
Proof of Work and consensus

14:30 - 15:15:
Proof of Stake and consensus

15:30 - 17:00:
Architectural debate

Kudos to Jan Martijn for the picture

Utrecht University

DISCLAIMER

The information in this presentation has been compiled with the utmost care,
but no rights can be derived from its contents.