“Blockchain Fundamentals”

Module 1: Blockchain Intuition	Module 1: Create a Blockchain
Module 2A: Cryptocurrency Intuition Module 2B: Cryptocurrency Transactions	Module 2: Create a Cryptocurrency
Module 3: Smart Contract Intuition	Module 3: Create a Smart Contract

What is a Blockchain

Stuart Haber

W. Scott Stornetta

What is a Blockchain

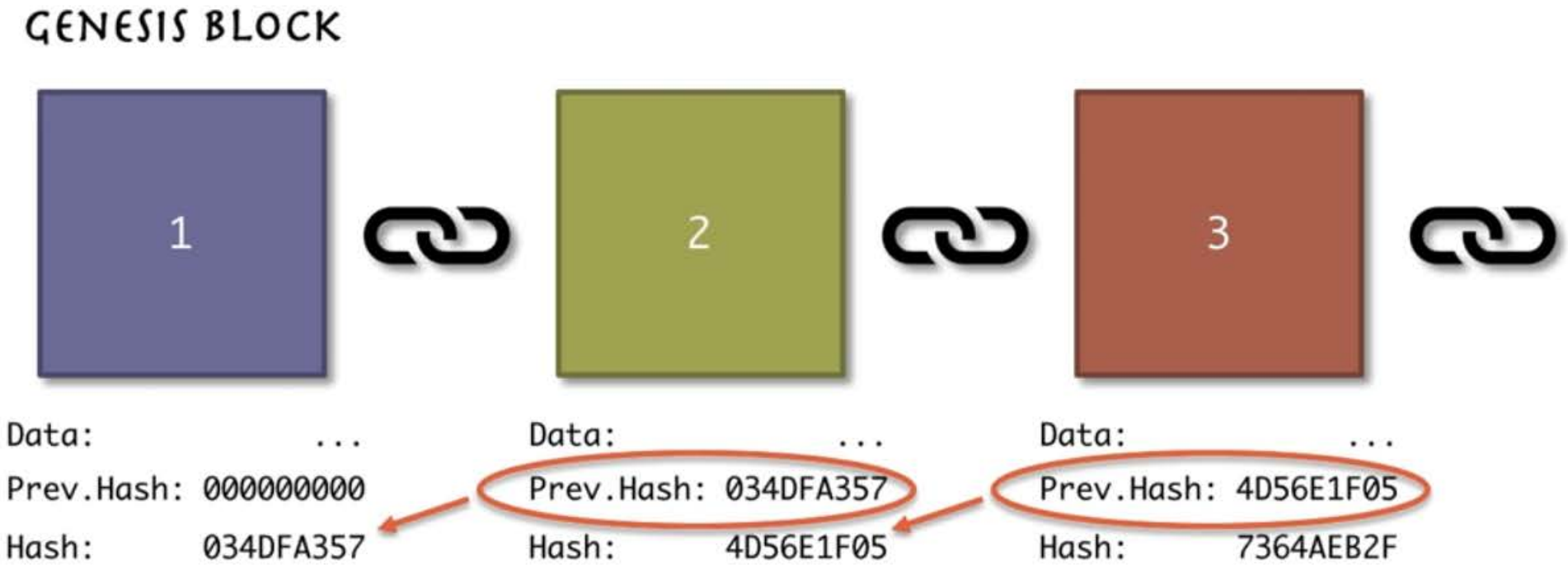
A blockchain is a continuously growing list of records, called blocks, which are linked and secured using cryptography.

– *Wikipedia*

What is a Blockchain

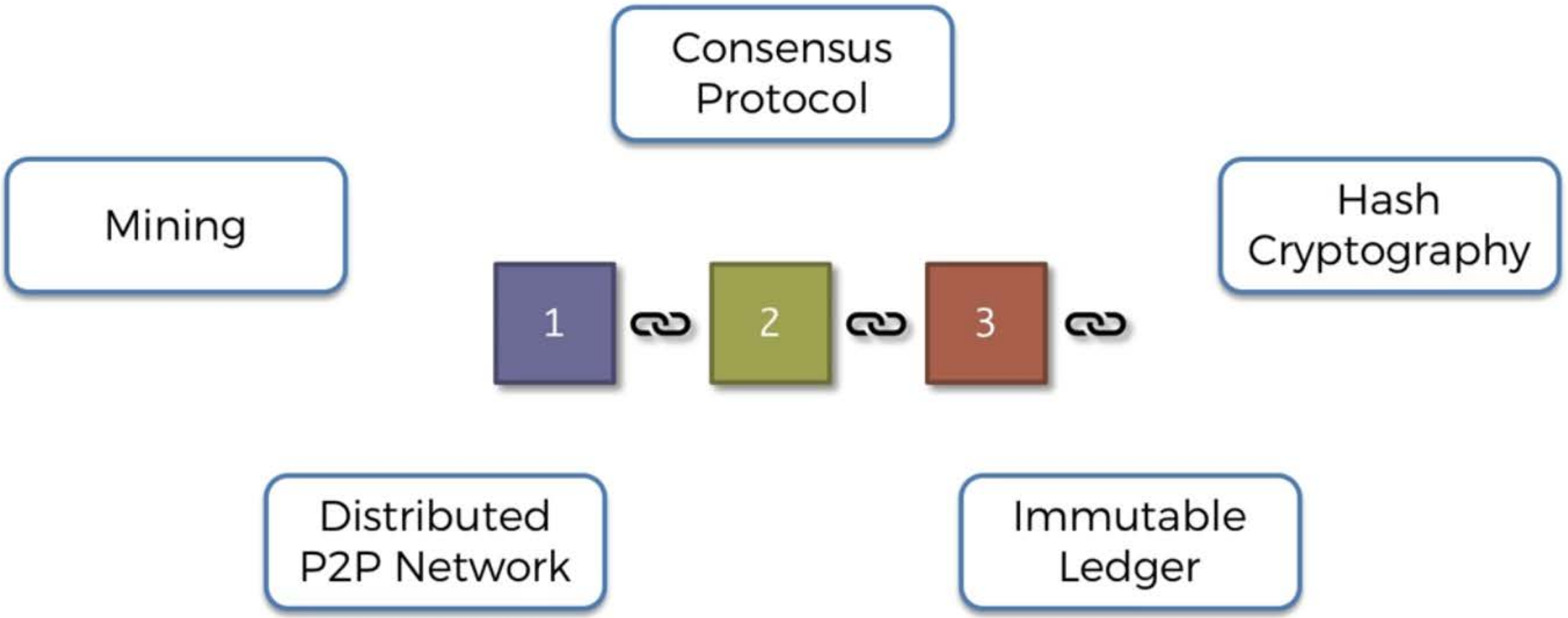
1. Data: “Hello World!”
2. Prev.Hash: 034DFA357
3. Hash: 4D56E1F05

What is a Blockchain



"Blocks are cryptographically linked together"

What is a Blockchain



Understanding SHA256 Hash

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F


```
2a89bf1700a91
40aa496380fd0
b4443921bbeef
b9cdb9ef6ea74
07cf82286afc
```

NSA

'SHA256'

64 characters

Understanding SHA256 Hash

The 5 requirements for Hash algorithms:

1. One-Way
2. Deterministic

Understanding SHA256 Hash

The 5 requirements for Hash algorithms:

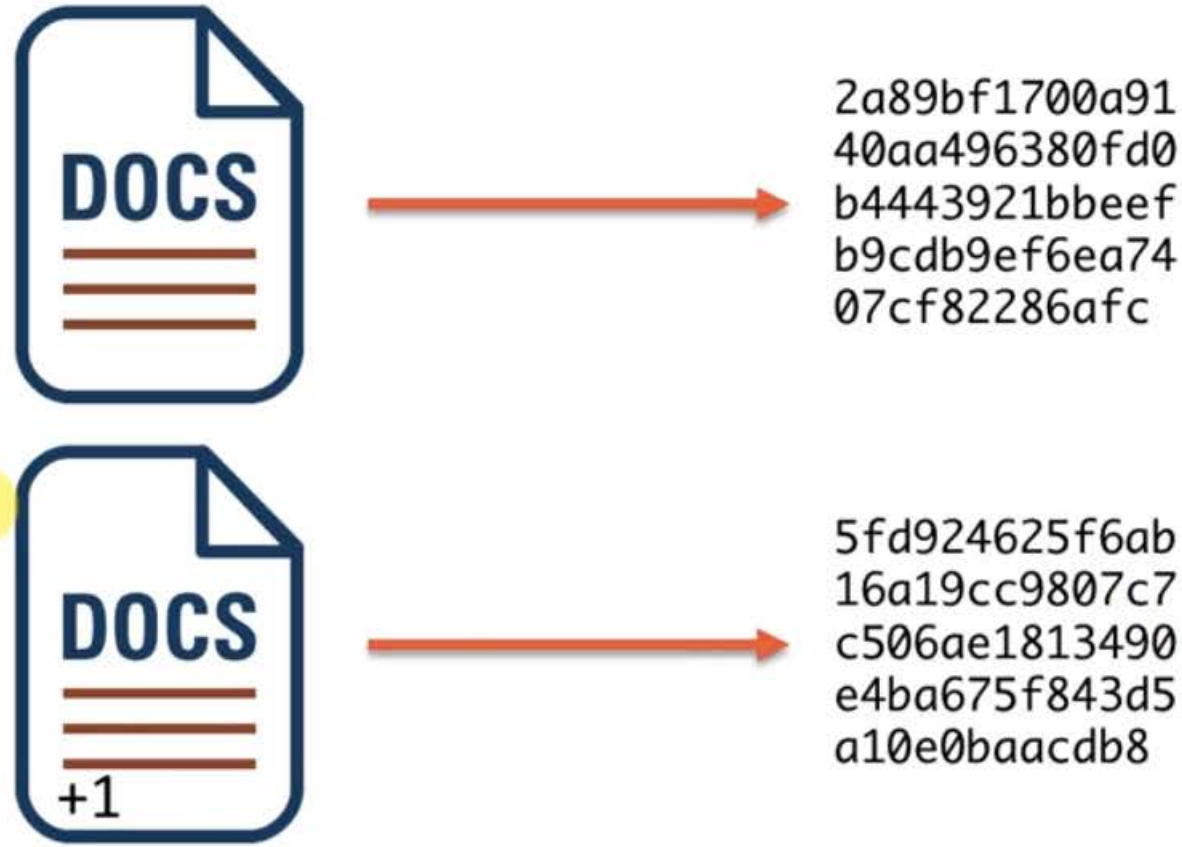
1. One-Way
2. Deterministic
3. Fast Computation


The Avalanche Effect

Understanding SHA256 Hash

The 5 requirements for Hash algorithms:

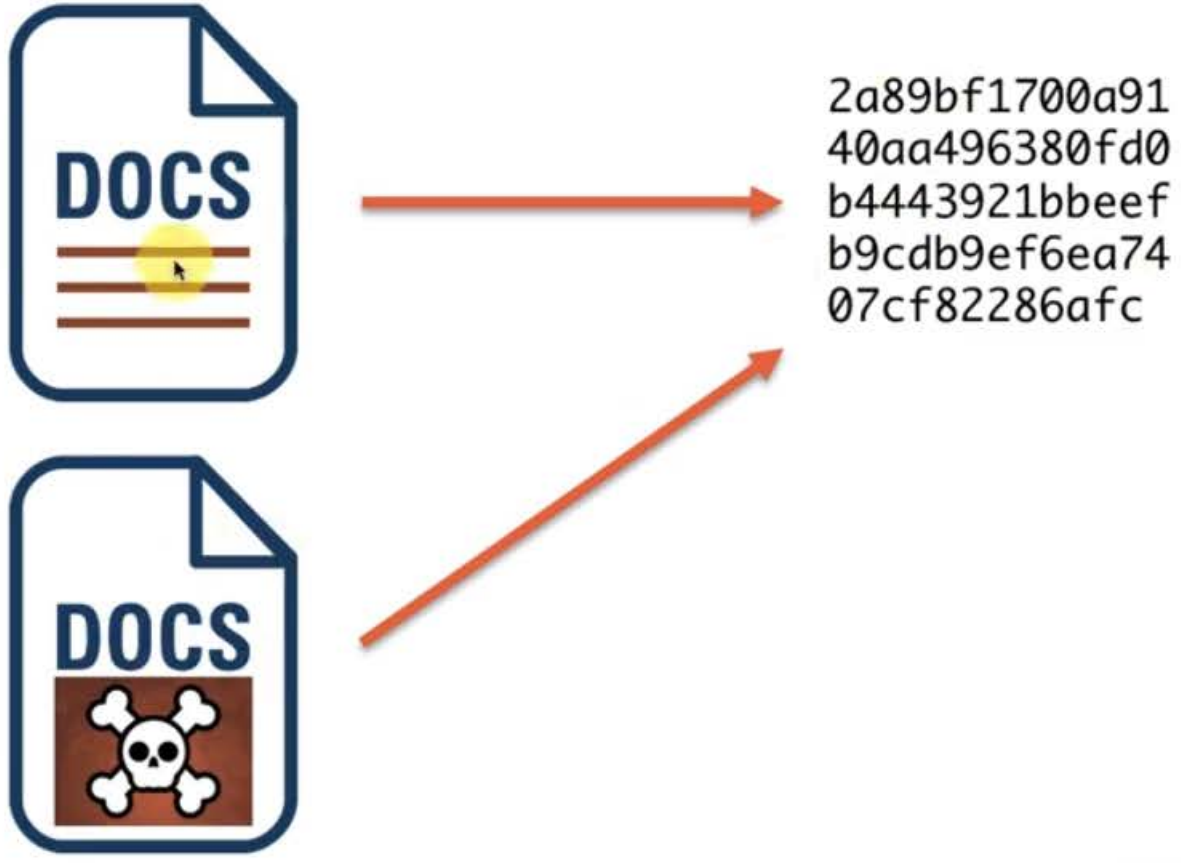
1. One-Way
2. Deterministic
3. Fast Computation
4. The Avalanche Effect



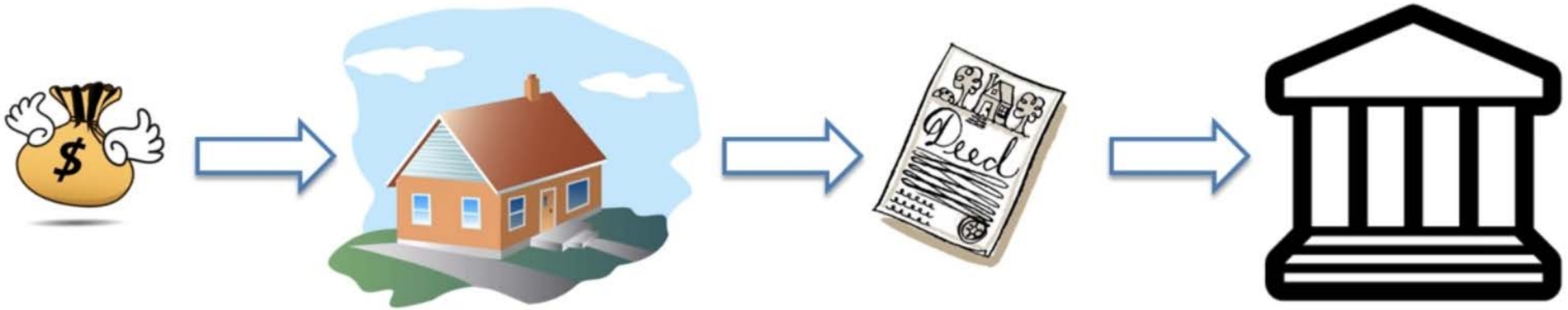
Understanding SHA256 Hash

The 5 requirements for Hash algorithms:

1. One-Way
2. Deterministic
3. Fast Computation
4. The Avalanche Effect
5. Must withstand collisions

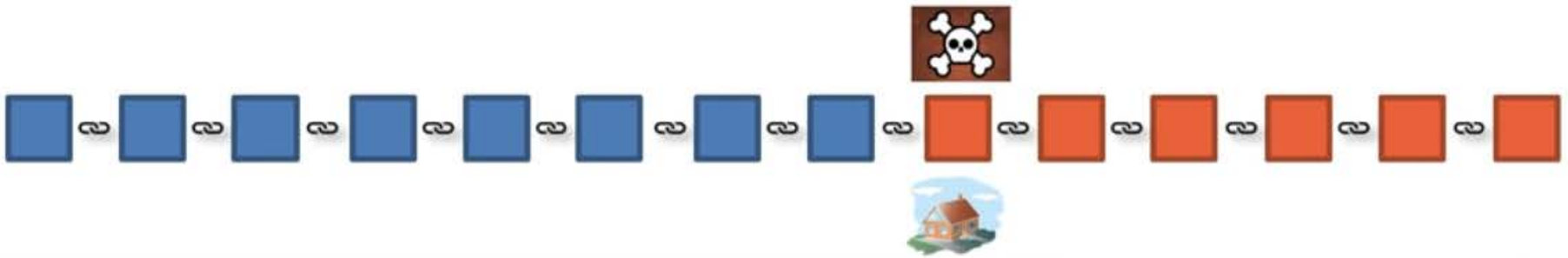


Immutable Ledger

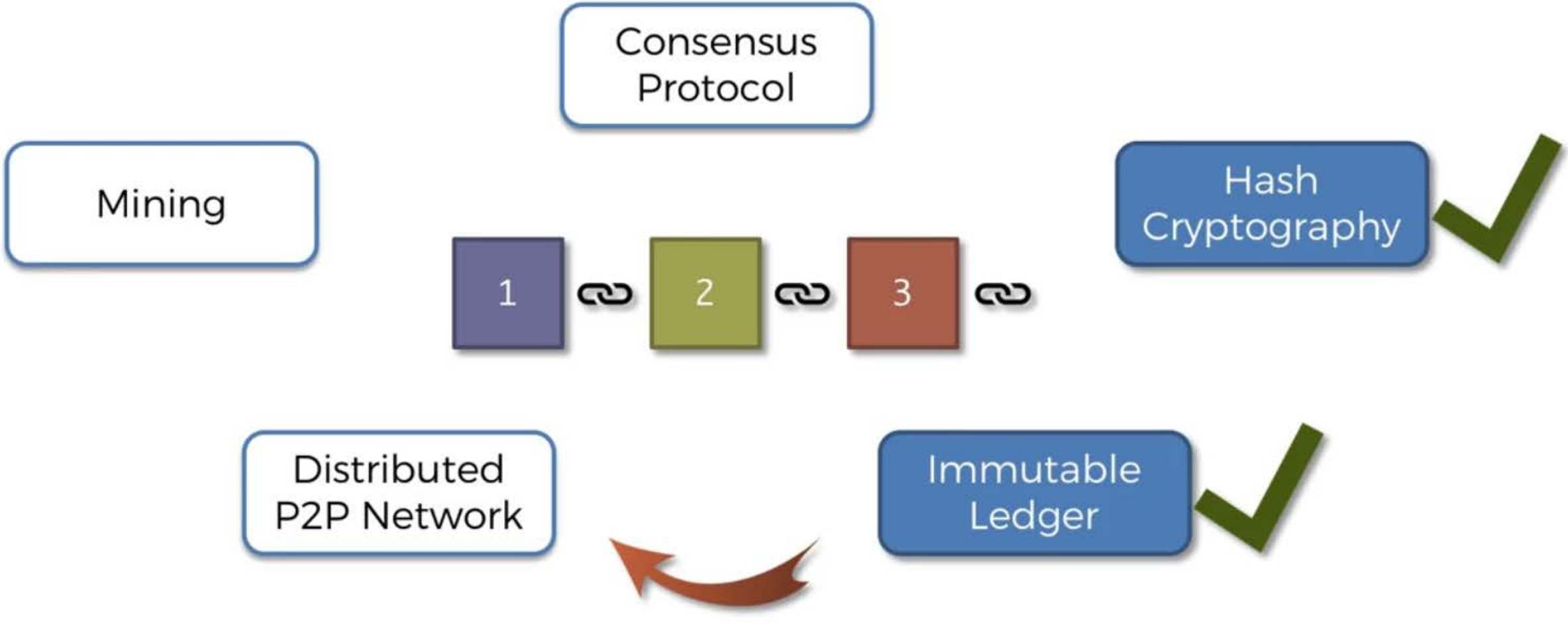


Traditional Ledger

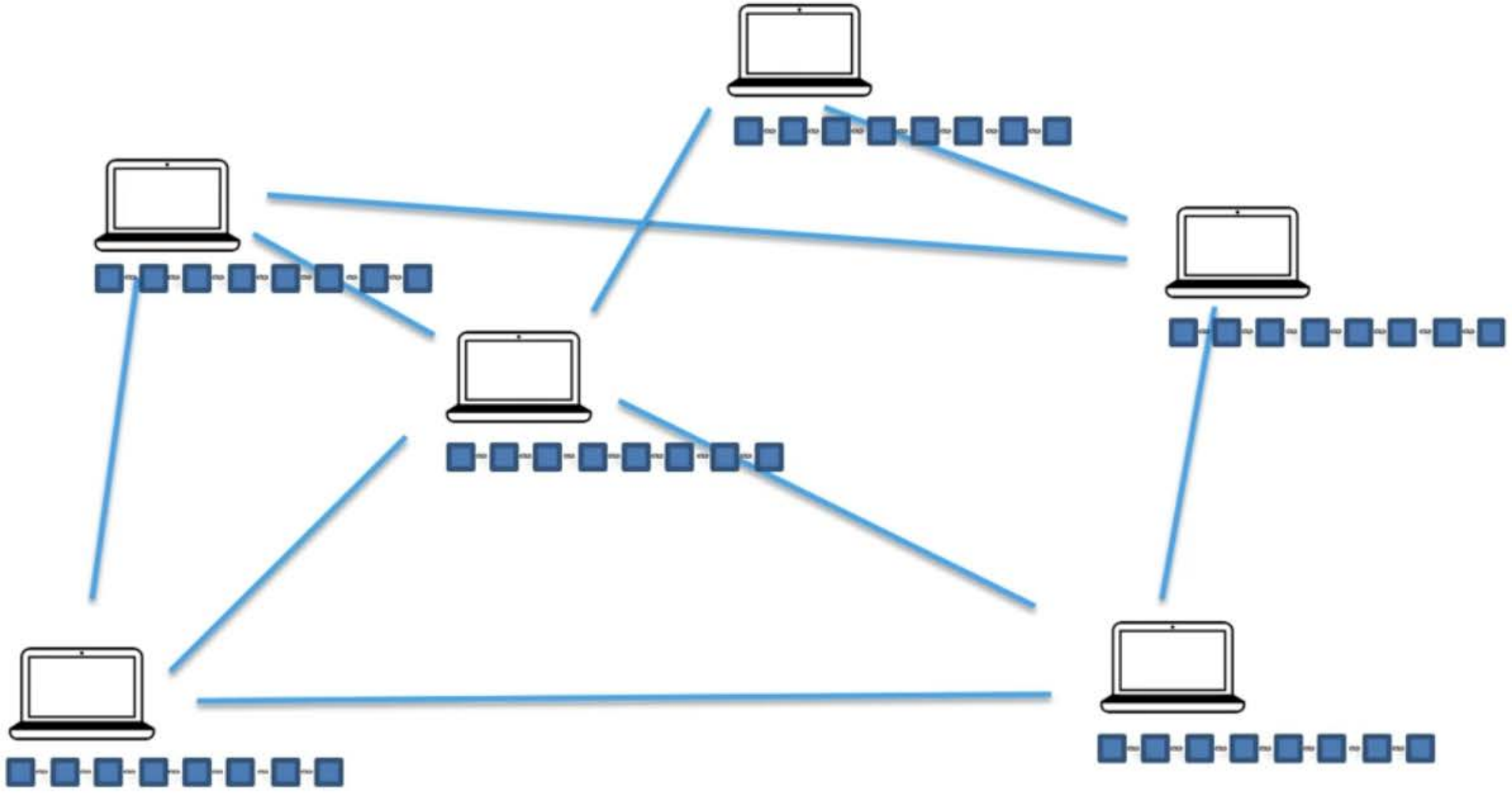
Blockchain



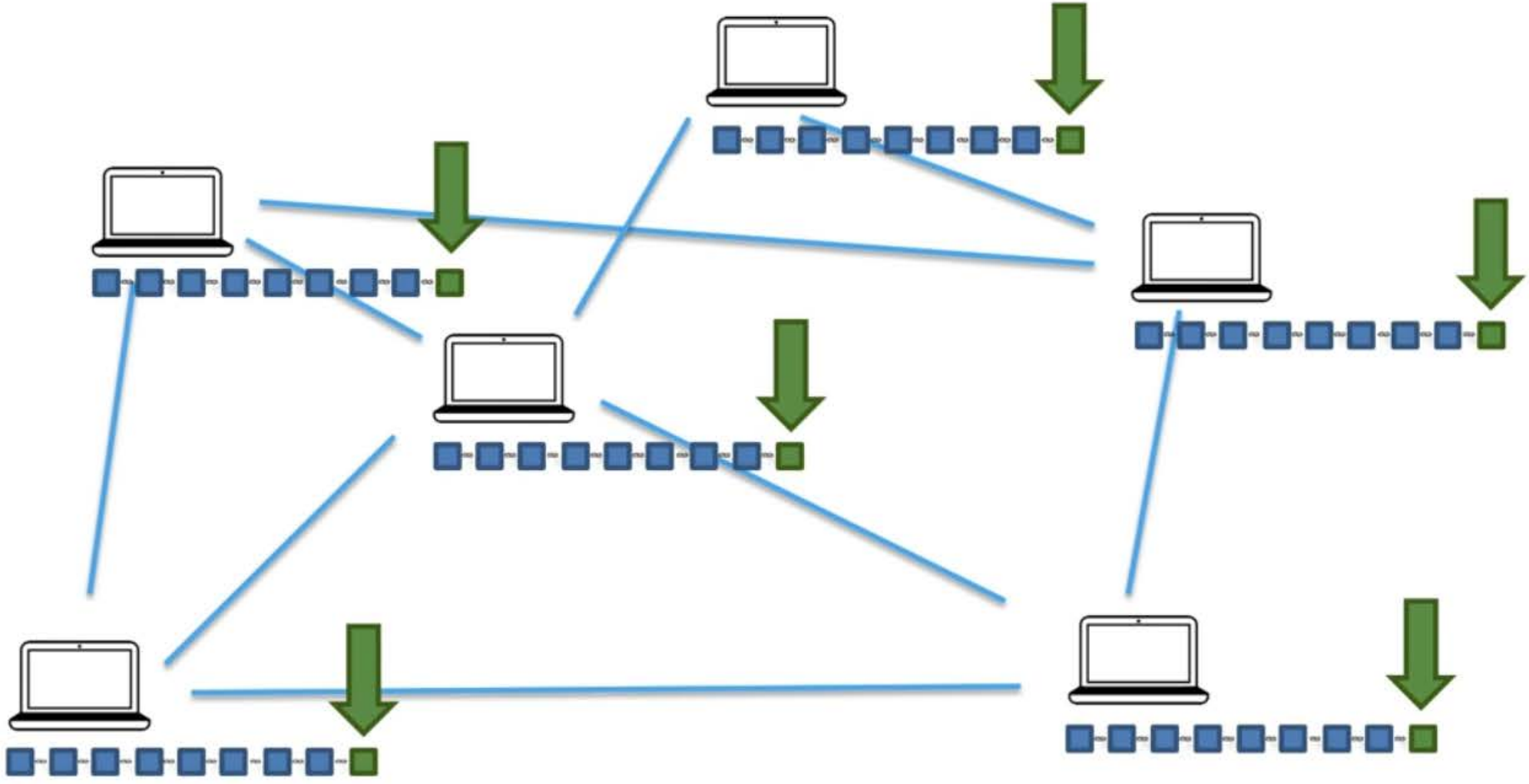
Distributed P2P Network



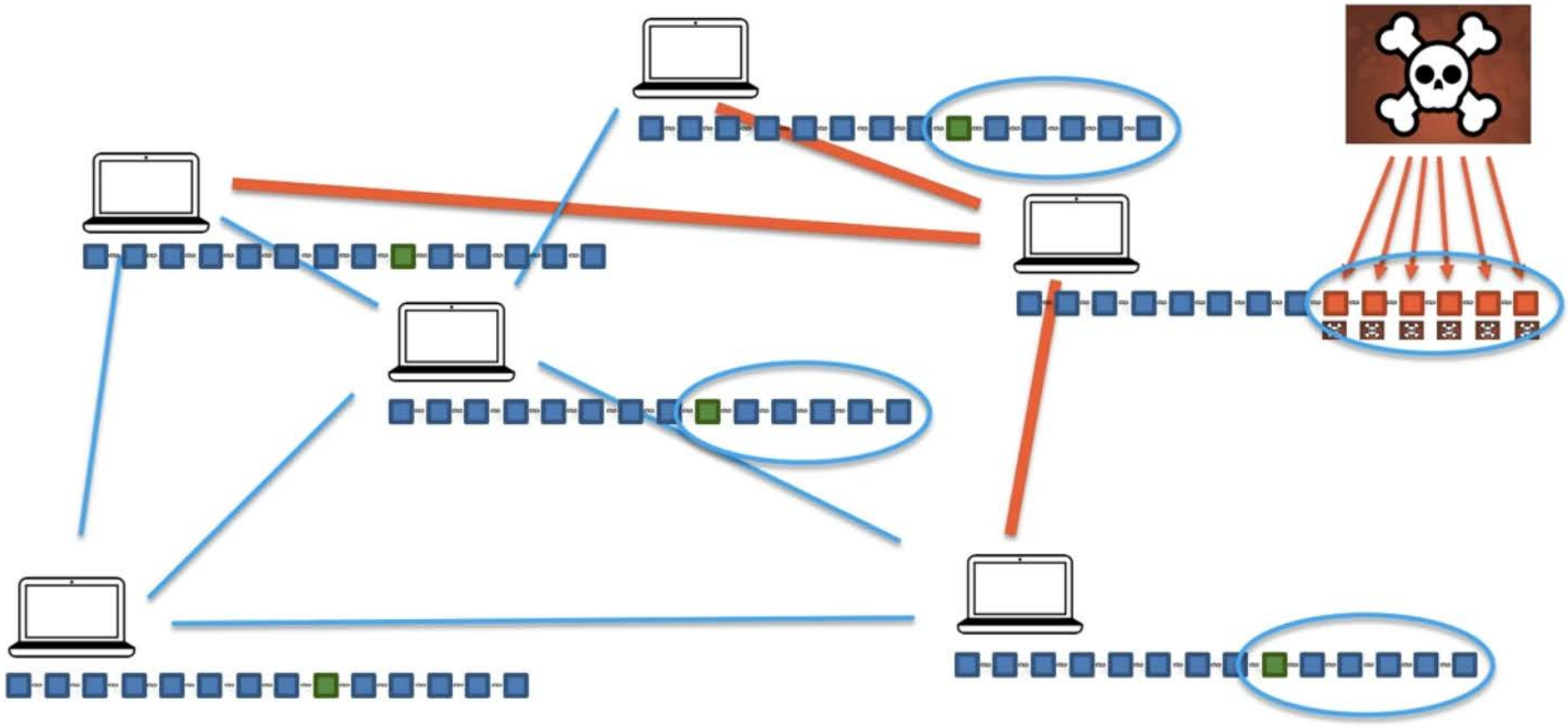
Distributed P2P Network



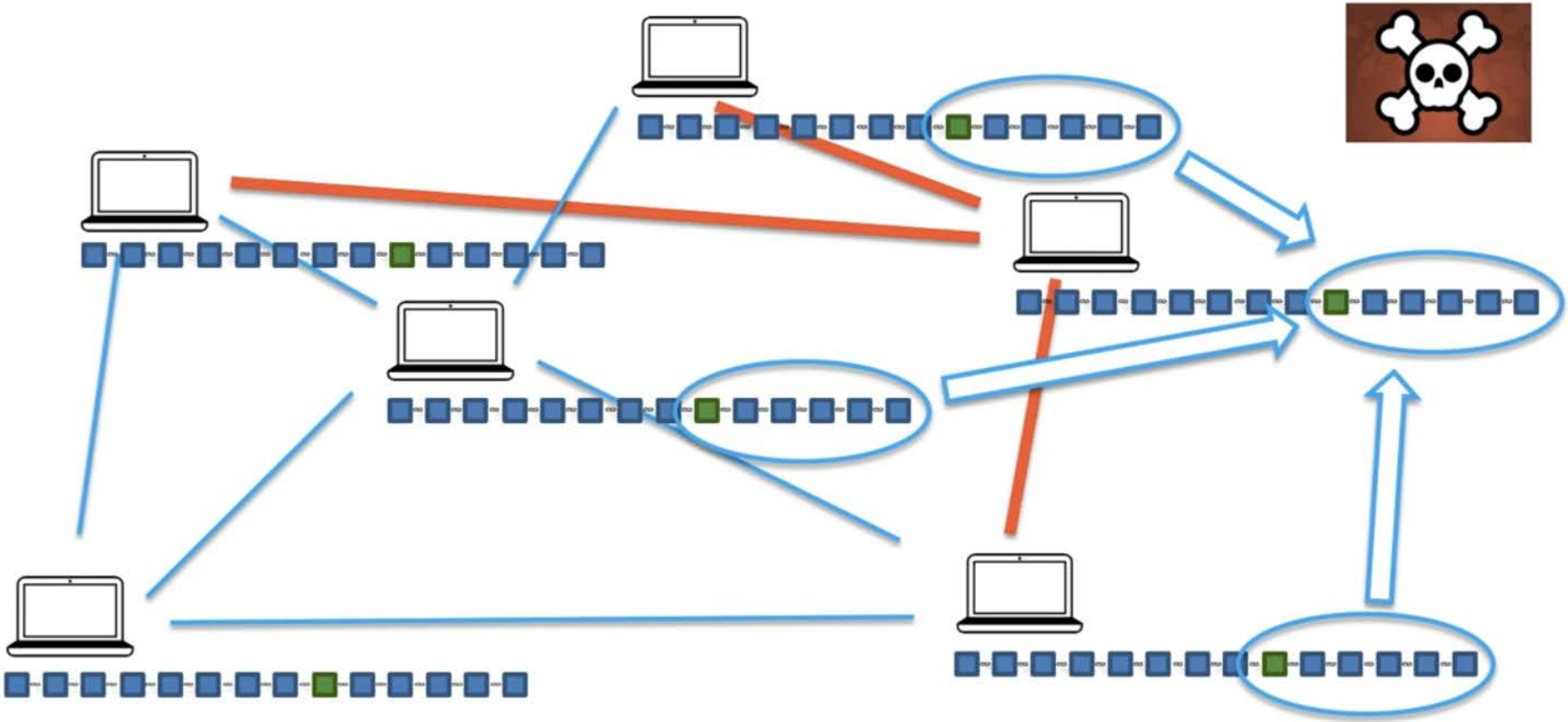
Distributed P2P Network



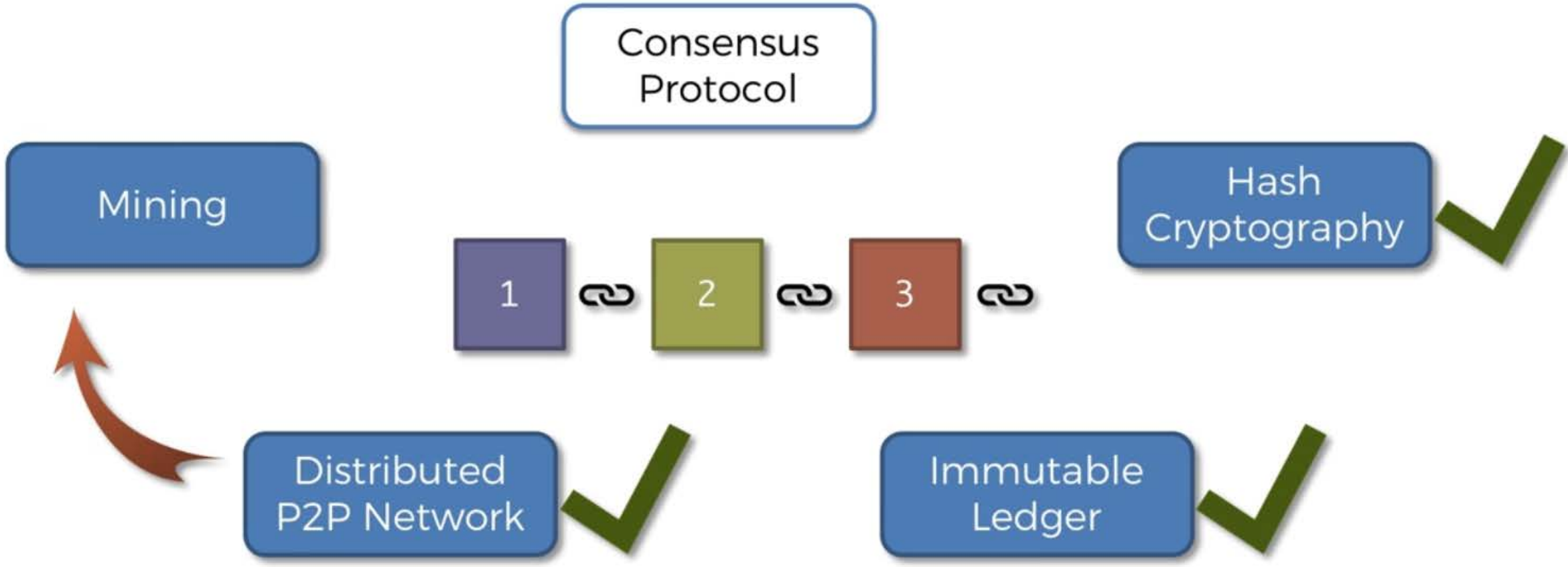
Distributed P2P Network



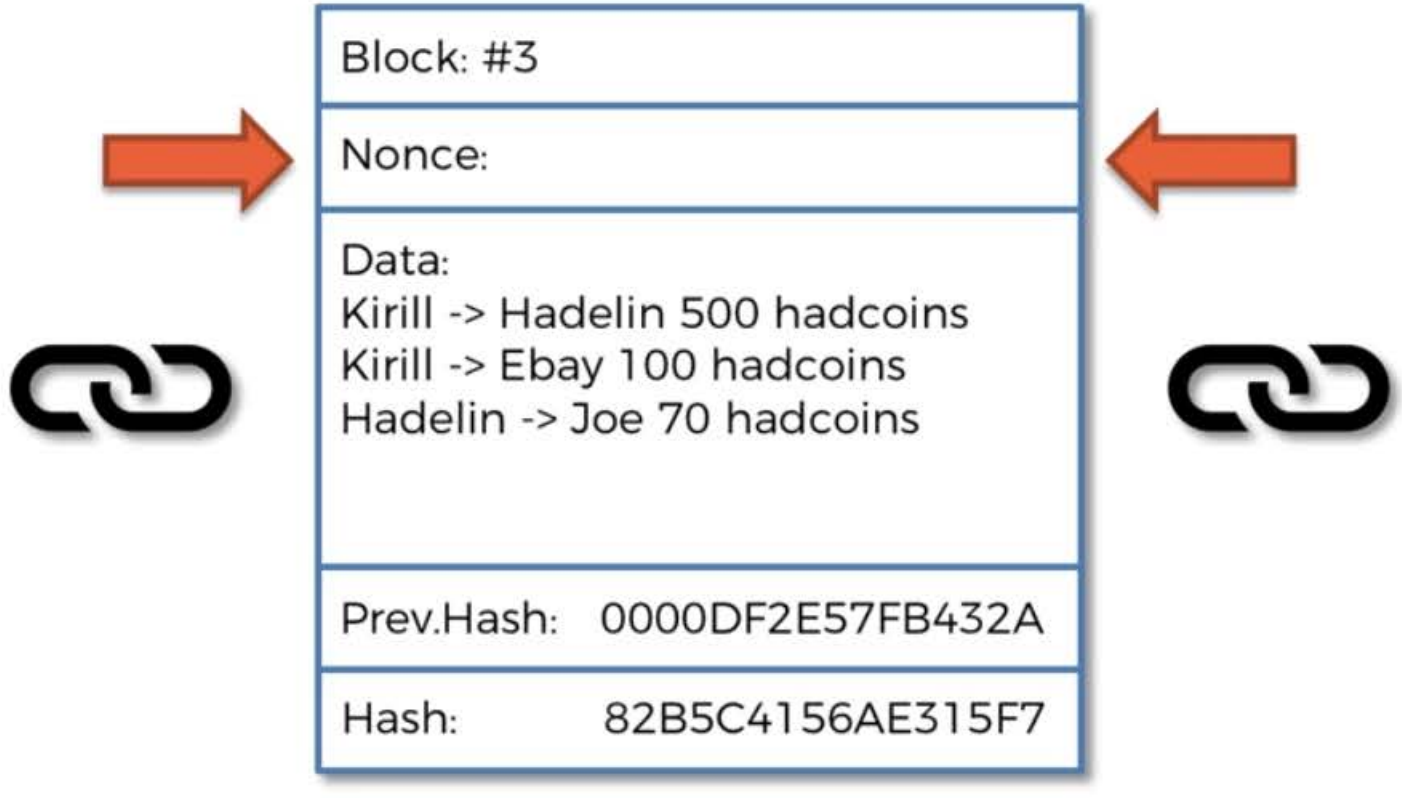
Distributed P2P Network



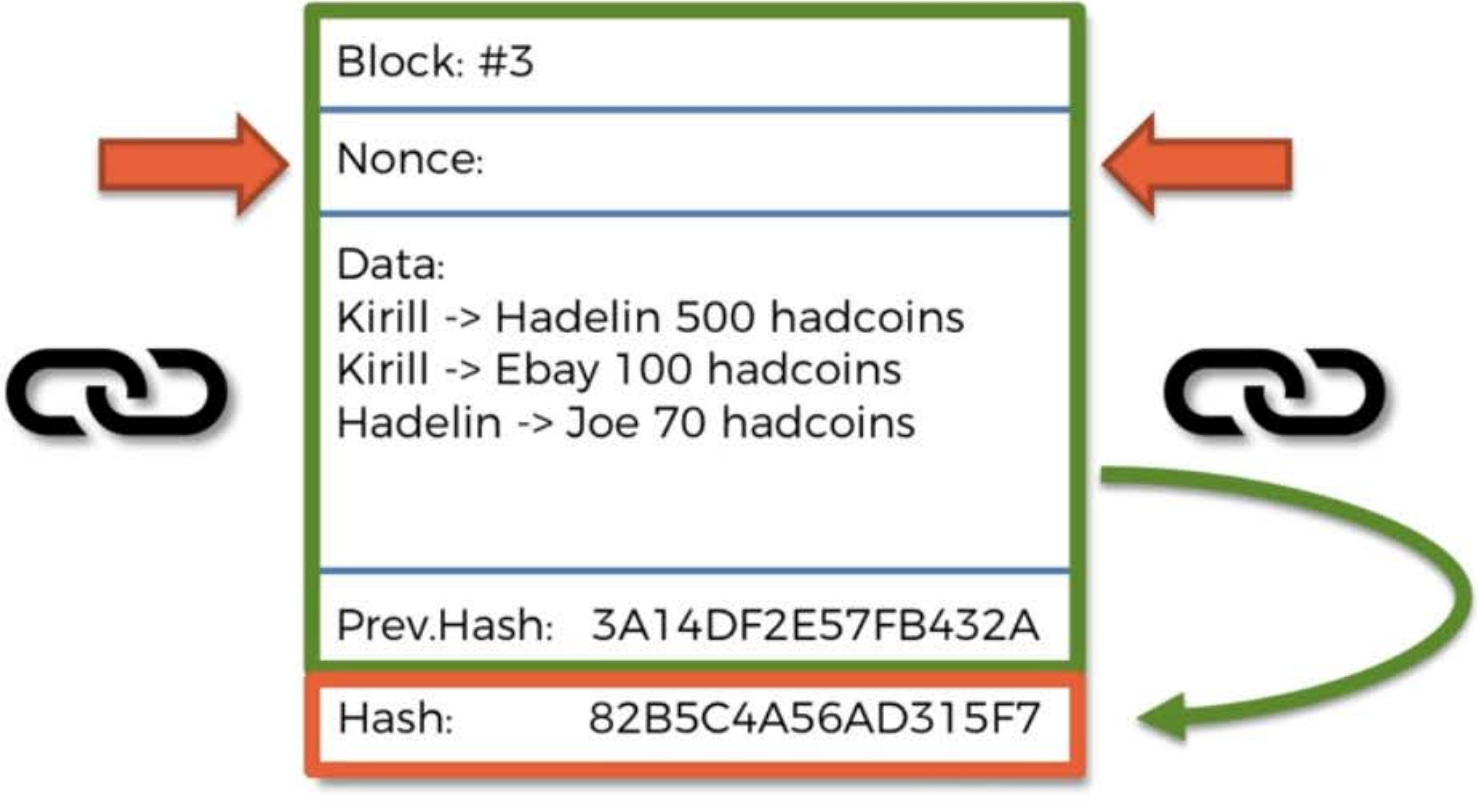
How Mining Works



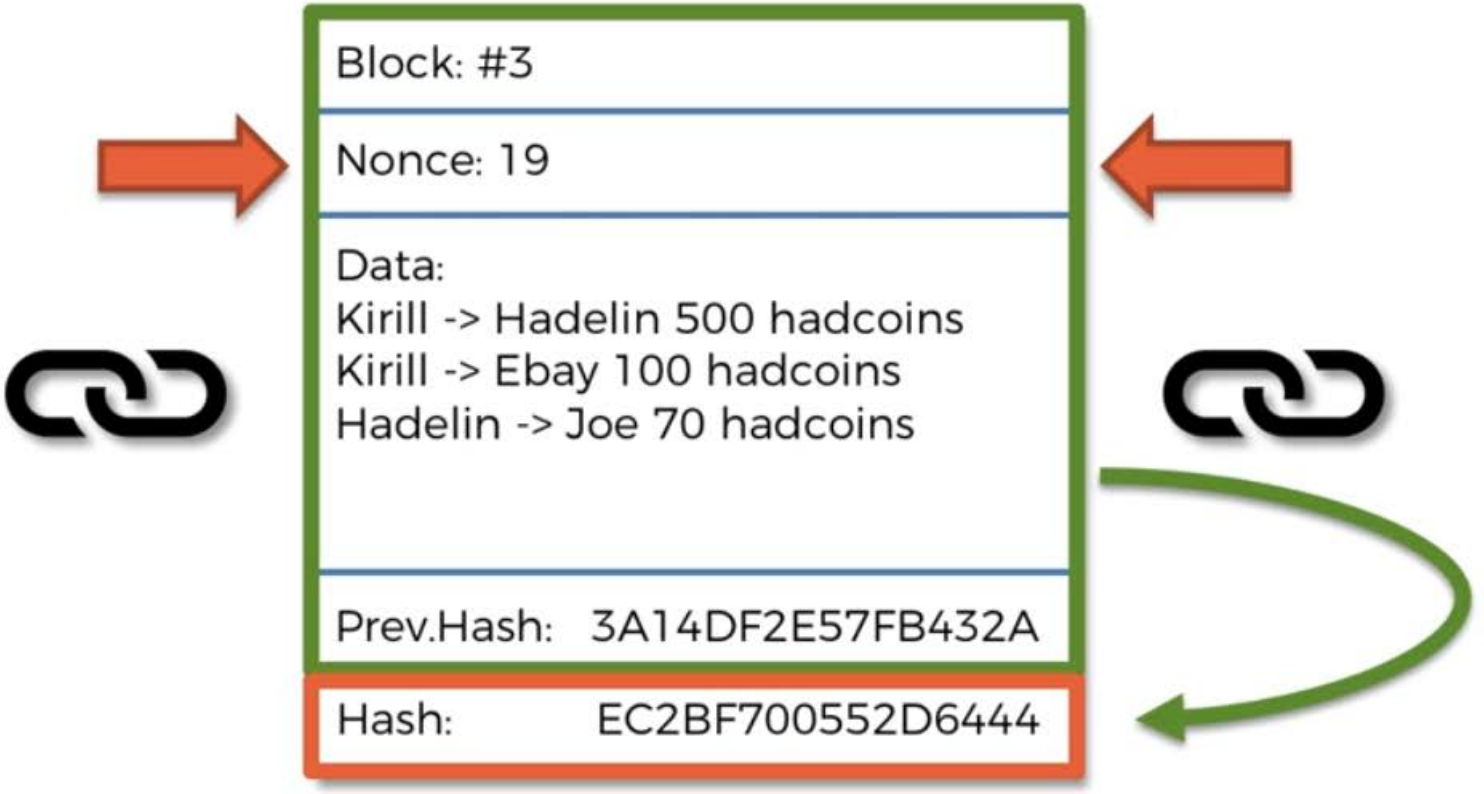
How Mining Works



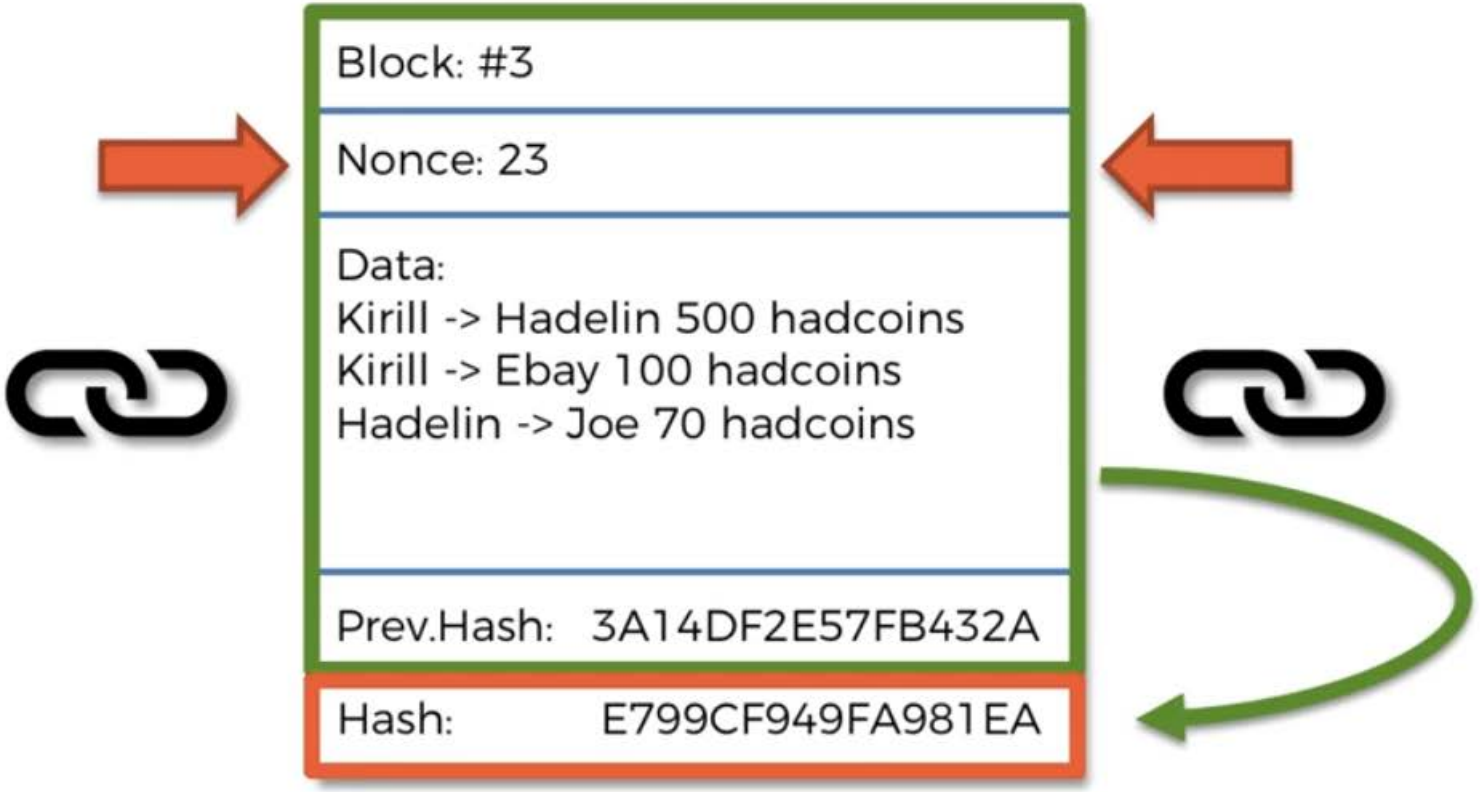
How Mining Works



How Mining Works



How Mining Works



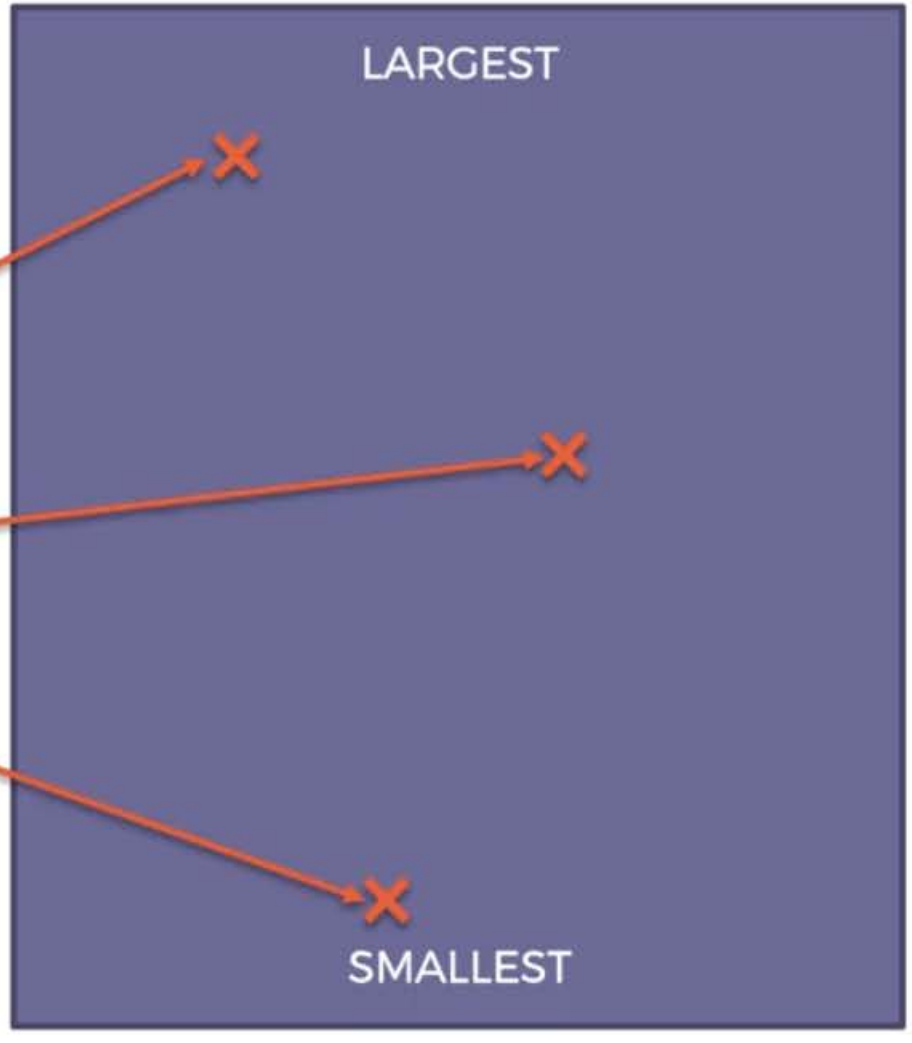
How Mining Works

A Hash is a Number

18D5A1AEDCBF543BC630130BEF99CFAD55D1B7413EF05B9AF927432FDE808C68
=112329626862361549158410627713034556651052663334451303122582684570577849908 24

00000000000087EC6D4886046788DCB49E9897F03C0A063F1F0CB57EEE7F0923
=000000000000002184207116031099371168244920544458523238690089125260753789934 43

0000000000000000000000000000000159CAA4B1EDA0FED66CB5E915C8F
=000000000000000000000000000000000000000000000438342898295709947707018187988111



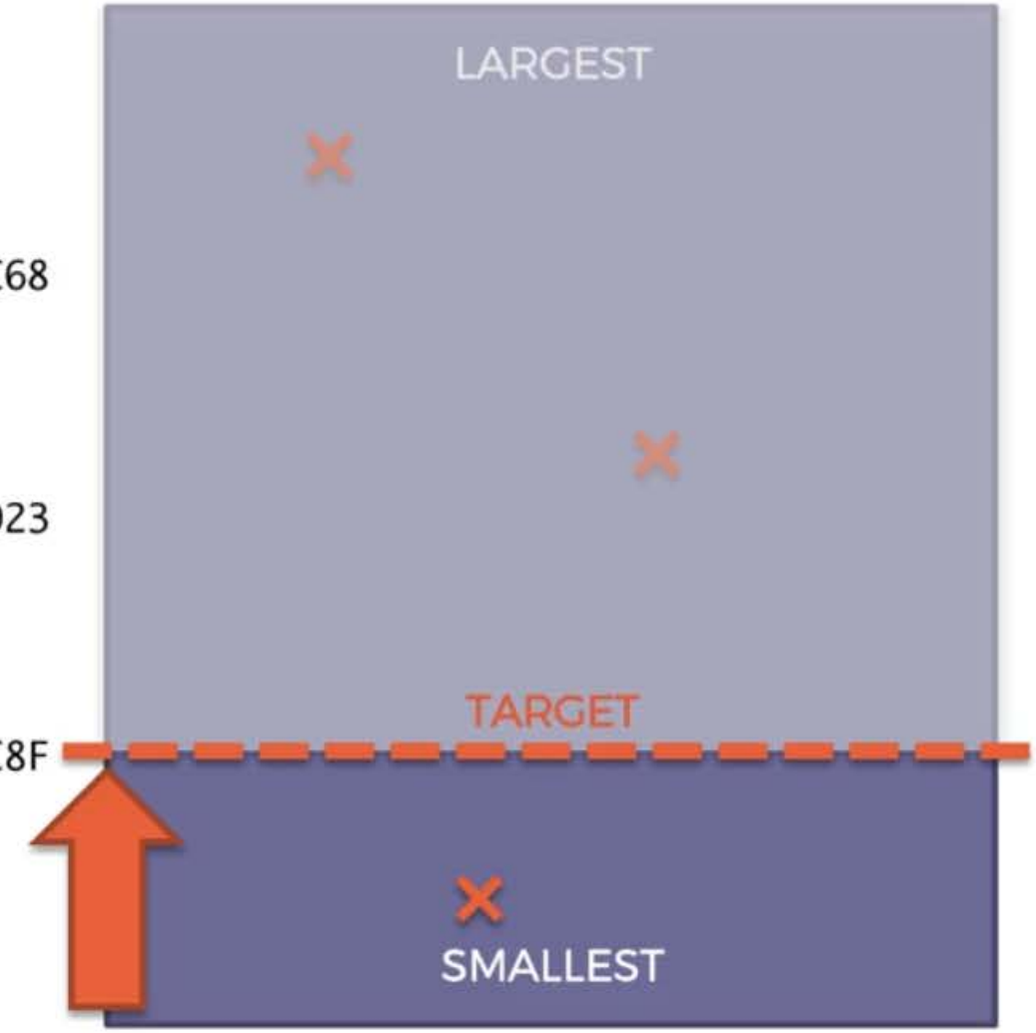
How Mining Works

- ALL POSSIBLE HASHES -

18D5A1AEDCBF543BC630130BEF99CFAD55D1B7413EF05B9AF927432FDE808C68

00000000000087EC6D4886046788DCB49E9897F03C0A063F1F0CB57EEE7F0923

0000000000000000000000000000000000159CAA4B1EDA0FED66CB5E915C8F



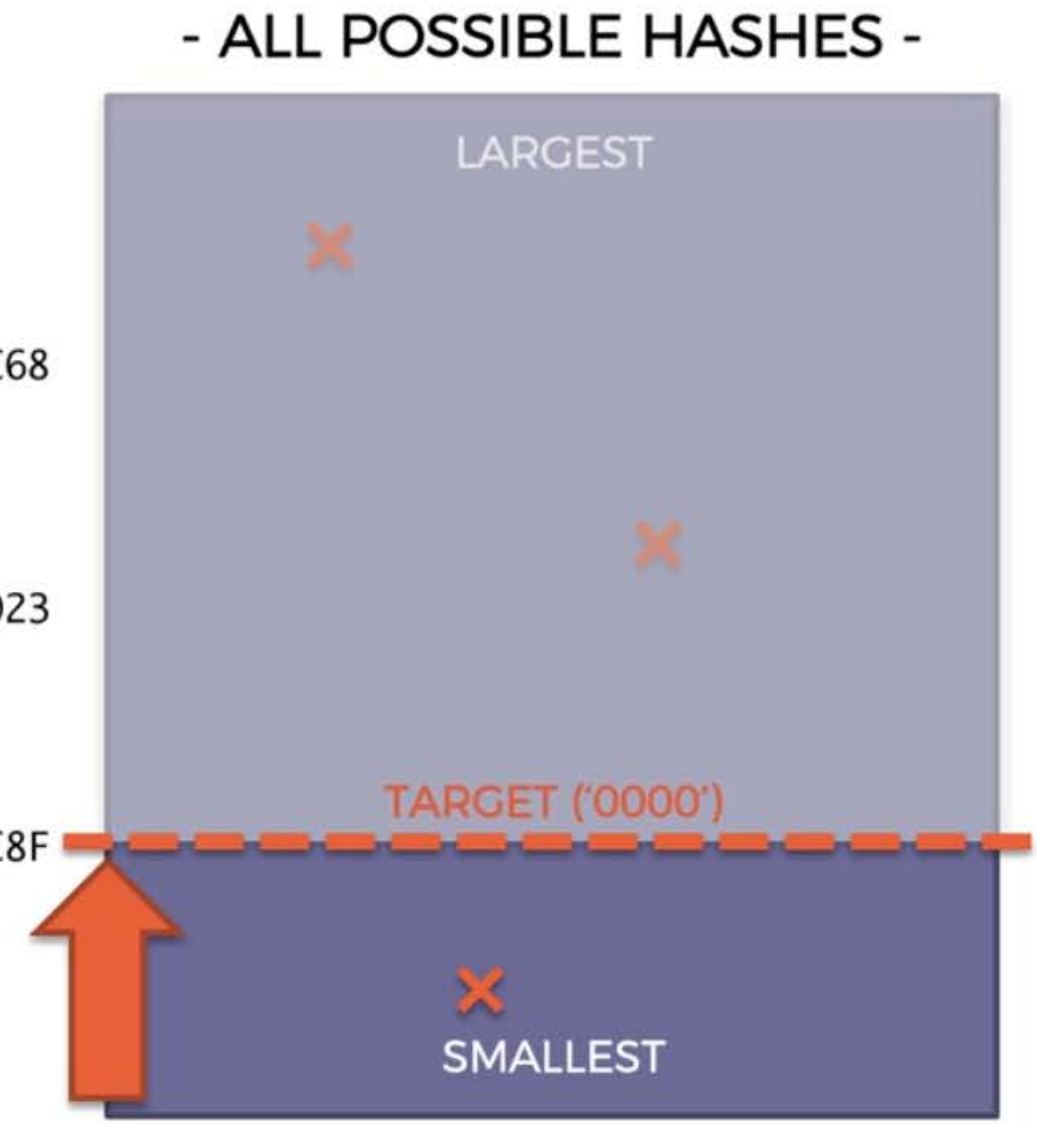
How Mining Works

18D5A1AEDCBF543BC630130BEF99CFAD55D1B7413EF05B9AF927432FDE808C68

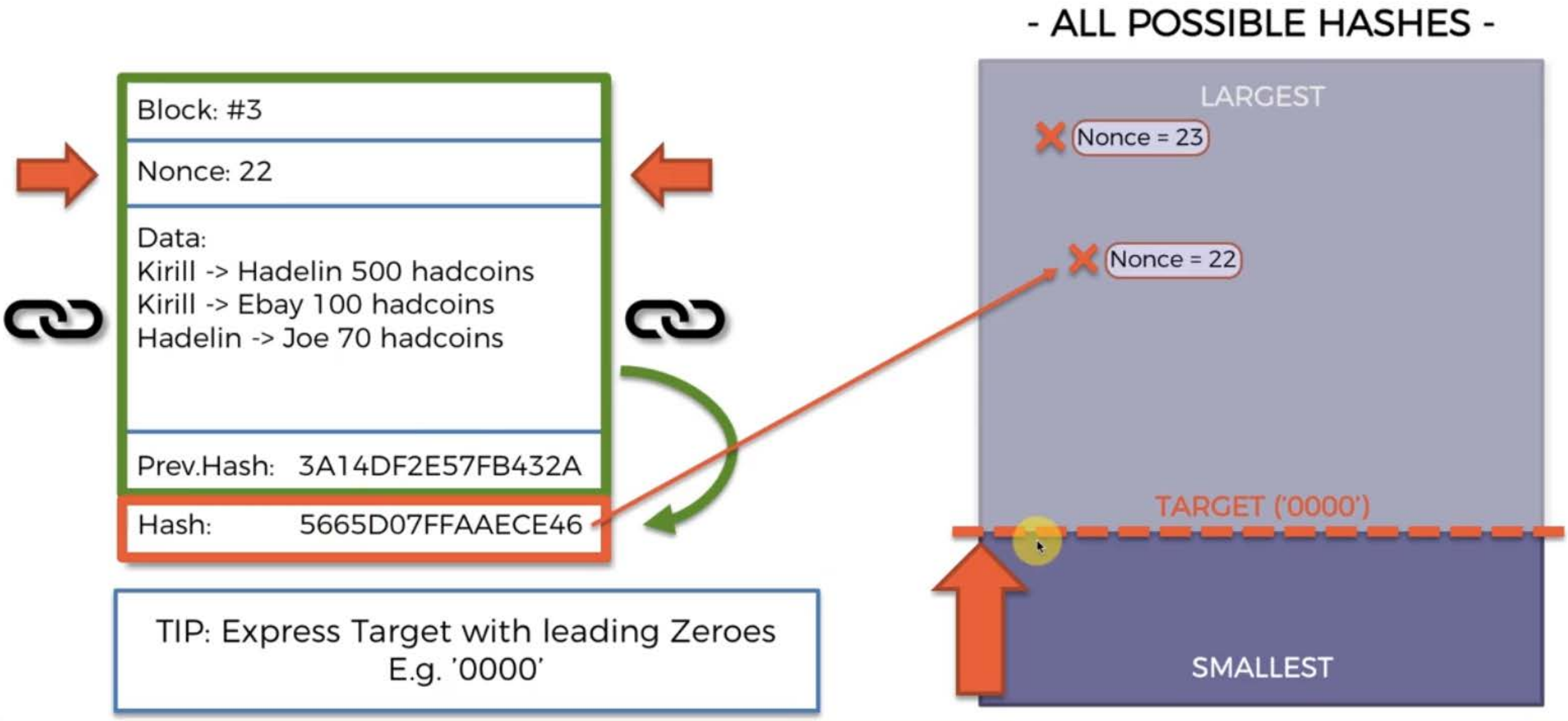
00000000000087EC6D4886046788DCB49E9897F03C0A063F1F0CB57EEE7F0923

00000000000000000000000000000000000159CAA4B1EDA0FED66CB5E915C8F

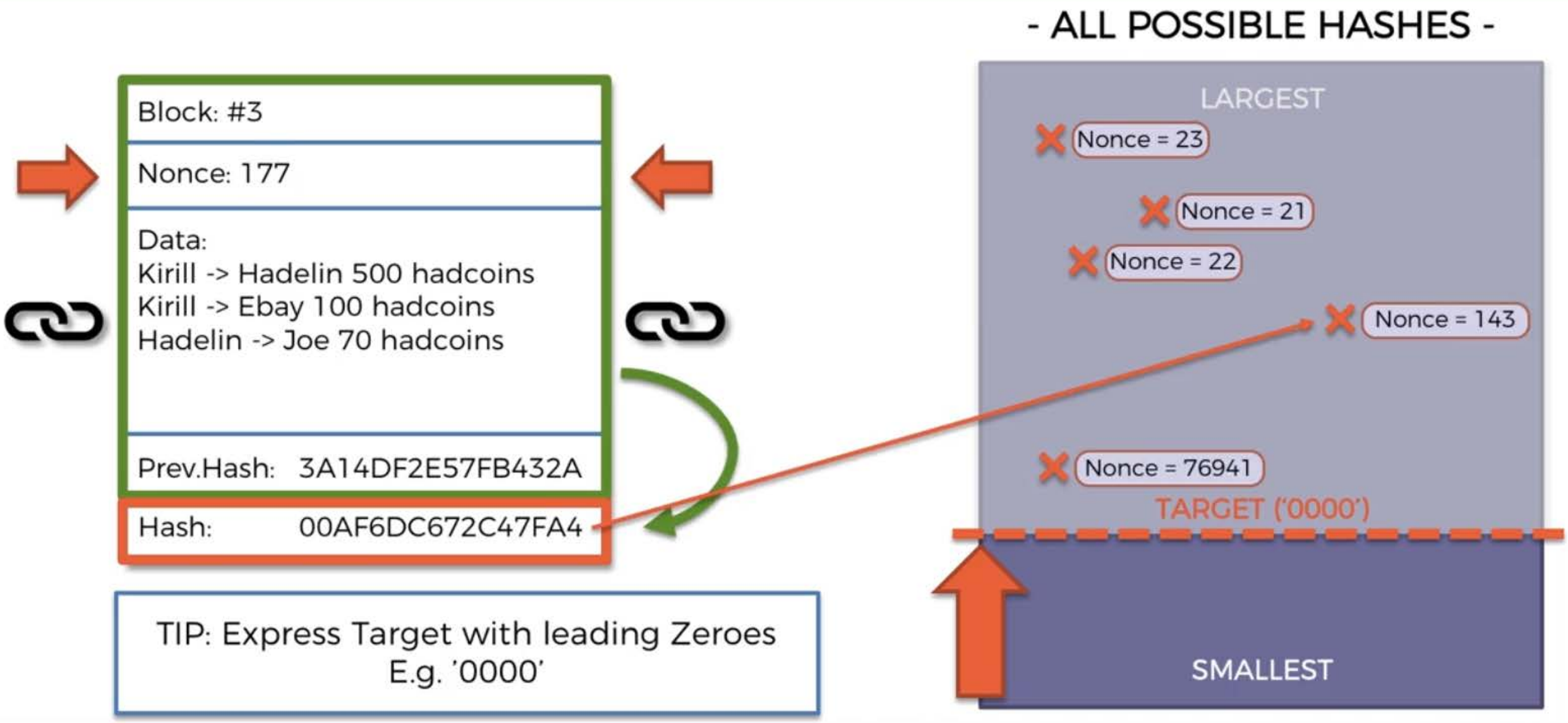
TIP: Express Target with leading Zeroes
E.g. '0000'



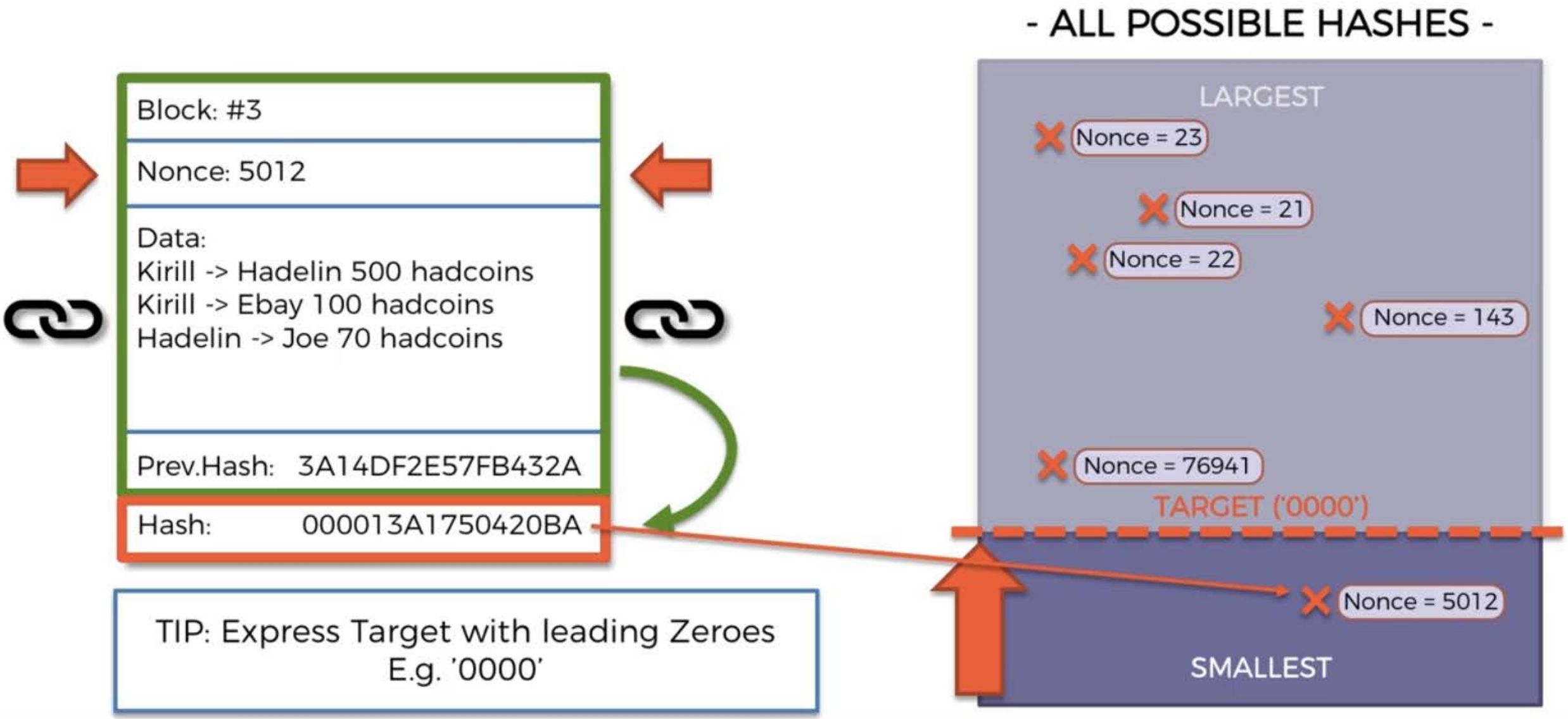
How Mining Works



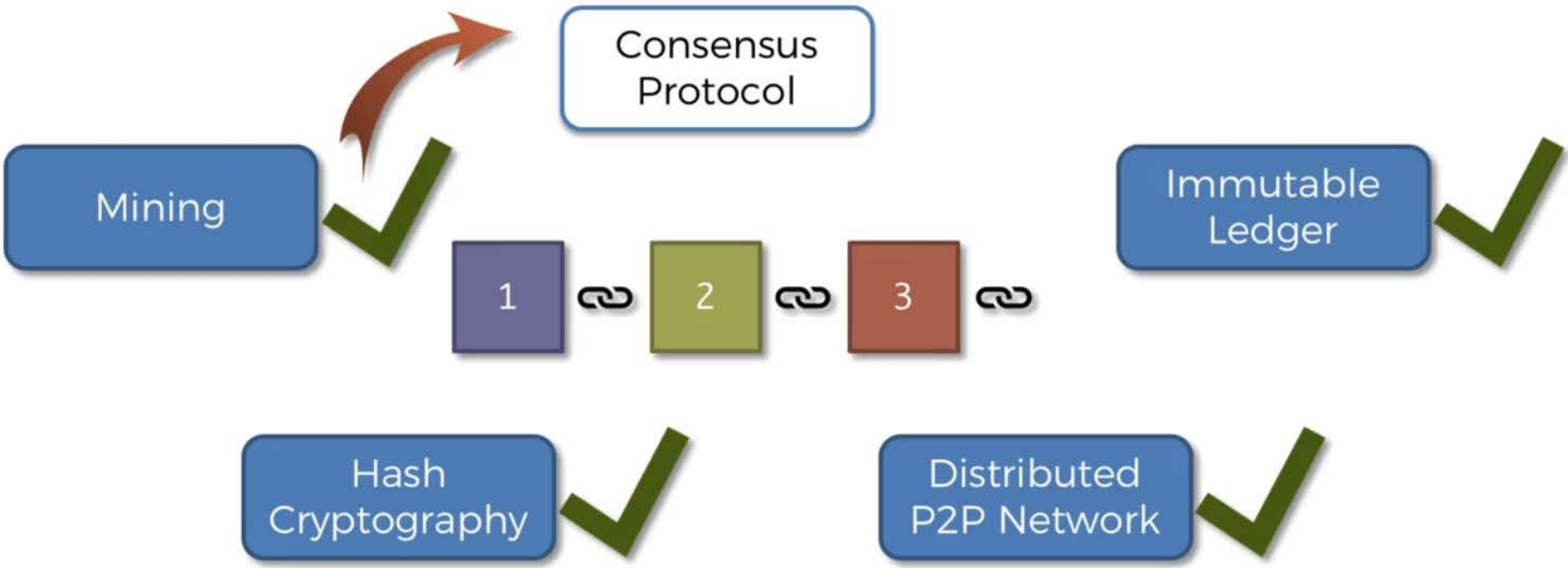
How Mining Works



How Mining Works

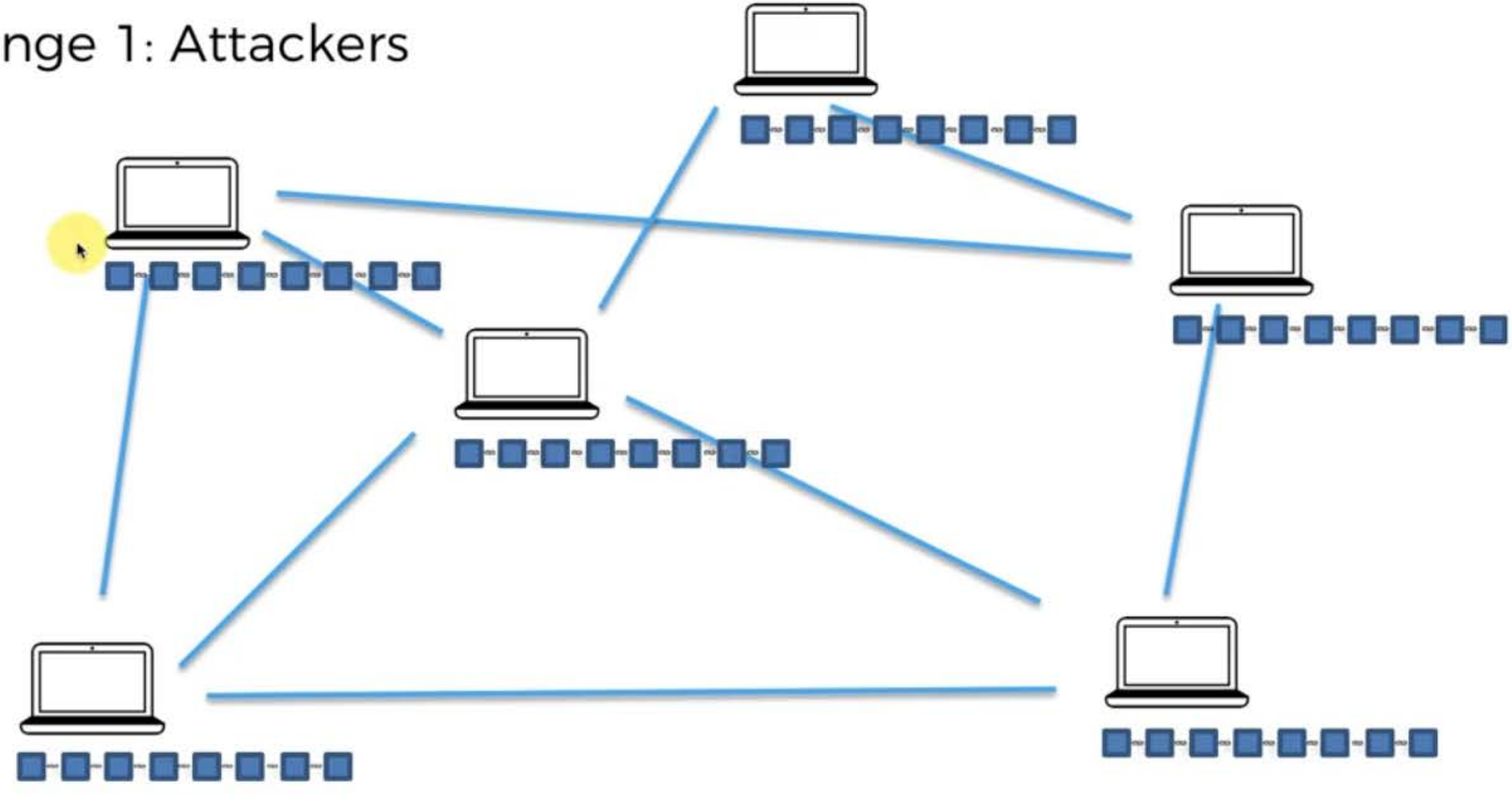


Consensus Protocol



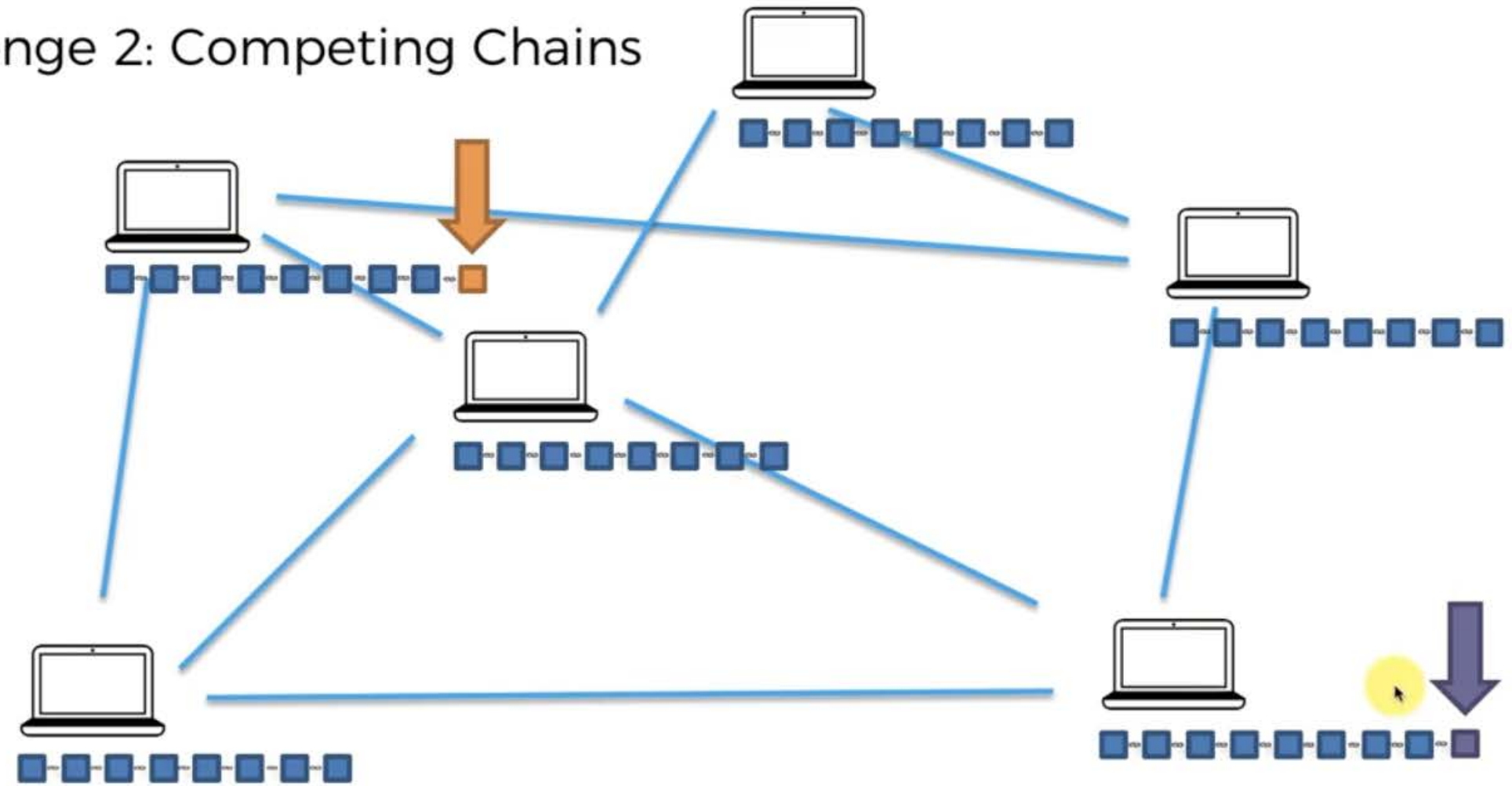
Consensus Protocol

Challenge 1: Attackers

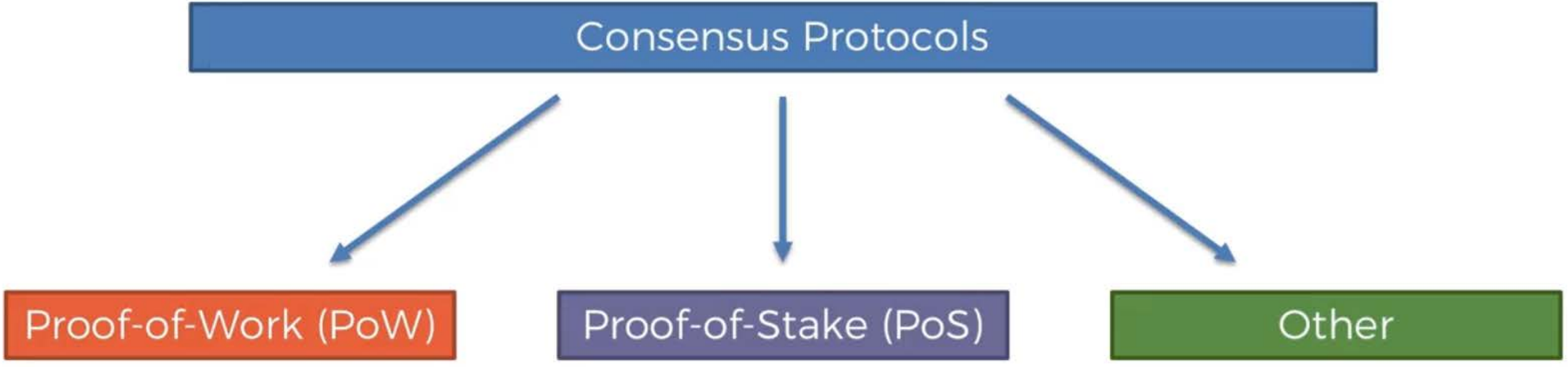


Consensus Protocol

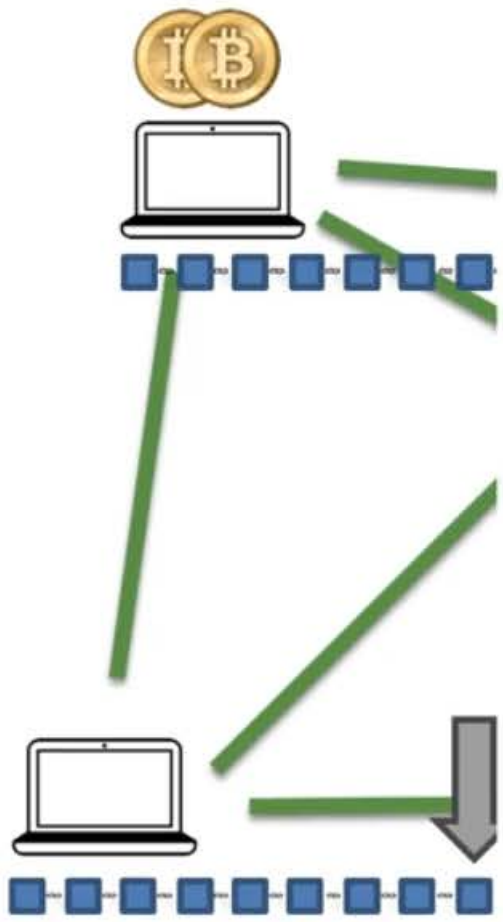
Challenge 2: Competing Chains



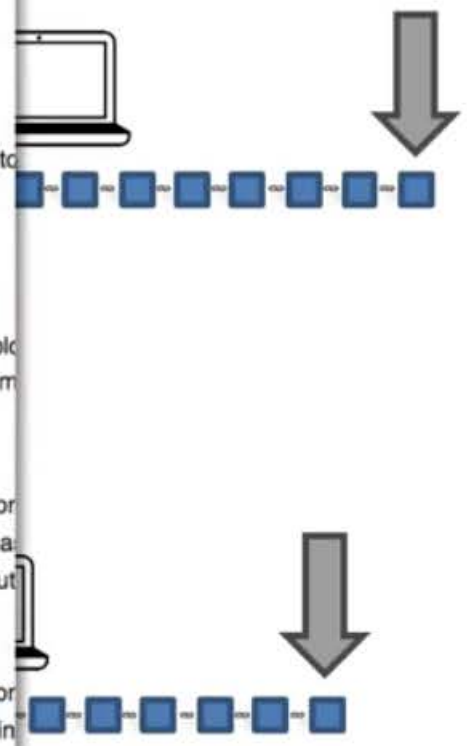
Consensus Protocol



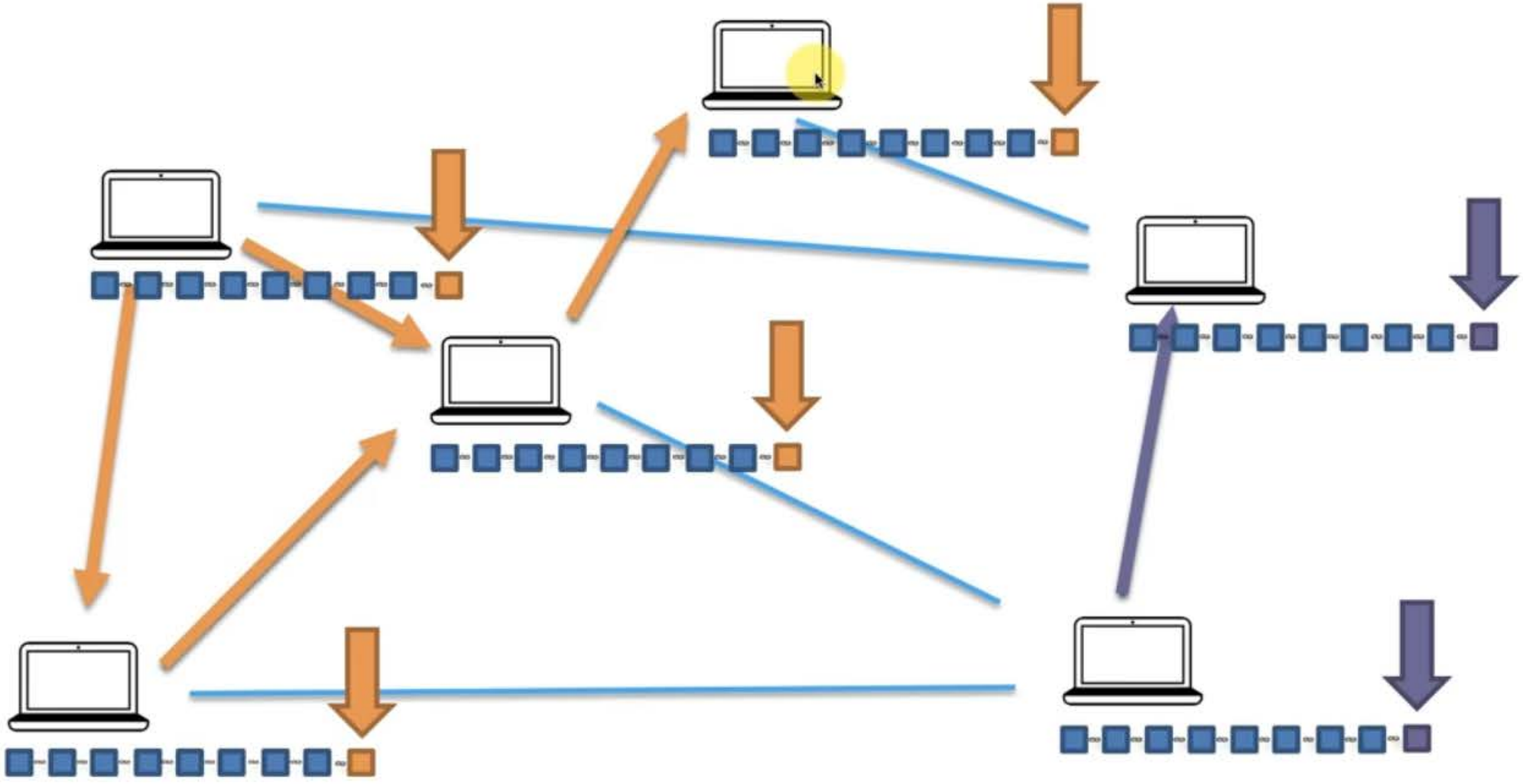
Consensus Protocol



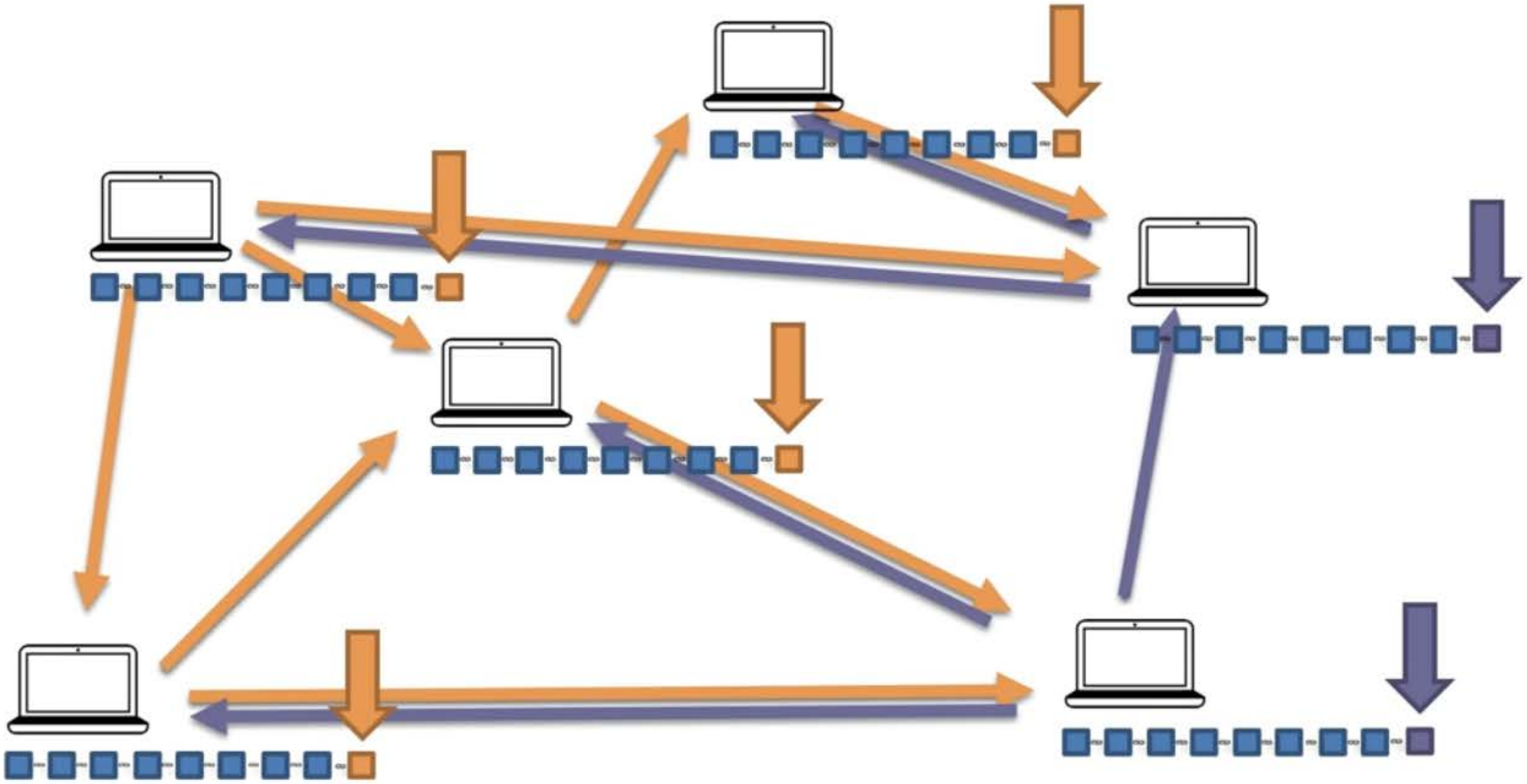
1. Check syntactic correctness
2. Reject if duplicate of block we have in any of the three categories
3. Transaction list must be non-empty
4. Block hash must satisfy claimed *nBits* proof of work
5. Block timestamp must not be more than two hours in the future
6. First transaction must be coinbase (i.e. only 1 input, with hash=0, n=-1), the rest must not be
7. For each transaction, apply "tx" checks 2-4
8. For the coinbase (first) transaction, scriptSig length must be 2-100
9. Reject if sum of transaction sig opcounts > MAX_BLOCK_SIGOPS
10. Verify Merkle hash
11. Check if prev block (matching *prev* hash) is in main branch or side branches. If not, add this to orphan block in *prev* chain; done with block
12. Check that *nBits* value matches the difficulty rules
13. Reject if timestamp is the median time of the last 11 blocks or before
14. For certain old blocks (i.e. on initial block download) check that hash matches known values
15. Add block into the tree. There are three cases: 1. block further extends the main branch; 2. blo make it become the new main branch; 3. block extends a side branch and makes it the new m
16. For case 1, adding to main branch:
    1. For all but the coinbase transaction, apply the following:
        1. For each input, look in the main branch to find the referenced output transactior
        2. For each input, if we are using the *n*th output of the earlier transaction, but it ha
        3. For each input, if the referenced output transaction is coinbase (i.e. only 1 input (100) confirmations; else reject.
        4. Verify crypto signatures for each input; reject if any are bad
        5. For each input, if the referenced output has already been spent by a transactior
        6. Using the referenced output transactions to get input values, check that each in
        7. Reject if the sum of input values < sum of output values
    2. Reject if coinbase value > sum of block creation fee and transaction fees



Consensus Protocol

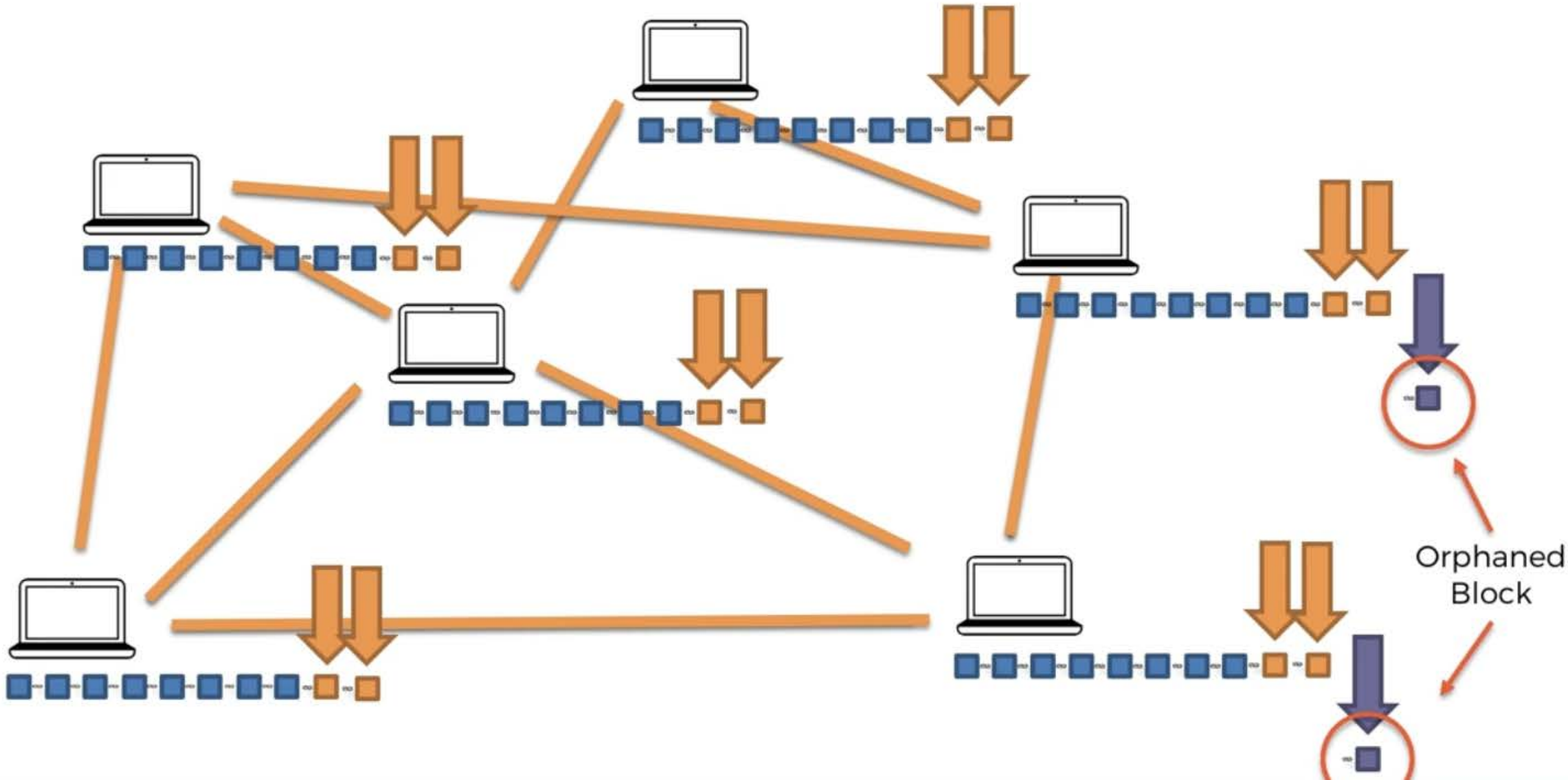


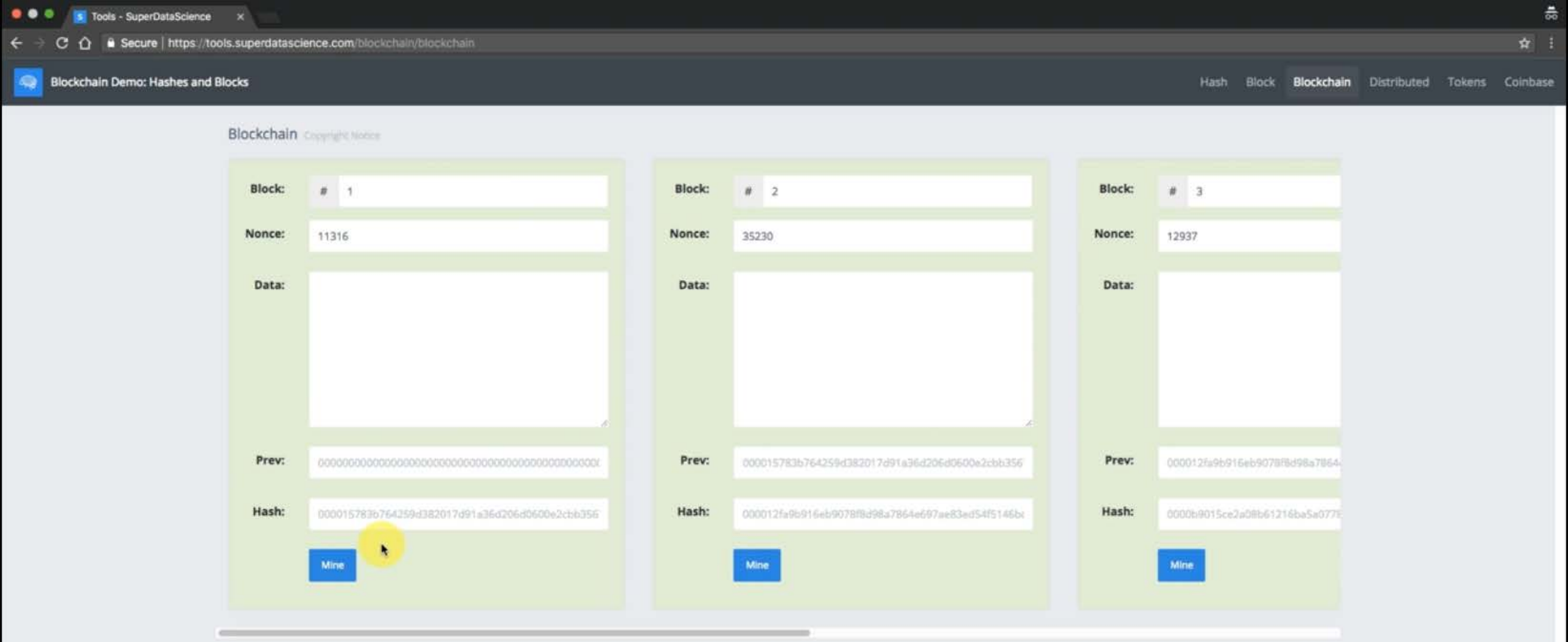
Consensus Protocol



So that's why we need bicentennial tolerance.

Consensus Protocol




Tools - SuperDataScience
Secure | https://tools.superdatascience.com/blockchain/blockchain
Blockchain Demo: Hashes and Blocks
Hash
Block
Blockchain
Distributed
Tokens
Coinbase
Blockchain
Block:
1
Nonce:
11316
Data:
Prev:
Hash:
000015783b764259d382017d91a36d206d0600e2cbb356
Mine
Block:
2
Nonce:
35230
Data:
Prev:
000015783b764259d382017d91a36d206d0600e2cbb356
Hash:
000012fa9b916eb9078f8d98a7864e697ae83ed54f5146b
Mine
Block:
3
Nonce:
12937
Data:
Prev:
000012fa9b916eb9078f8d98a7864
Hash:
0000b9015ce2a08b61216ba5a077
Mine

```
Python 3.6.3 |Anaconda, Inc.| (default, Oct  6 2017, 12:04:38)
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

In [1]: import hashlib

In [2]: previous_proof = 2

In [3]: new_proof = 3

In [4]: new_proof**2 - previous_proof**2
Out[4]: 5

In [5]: str(new_proof**2 - previous_proof**2)
Out[5]: '5'

In [6]: str(new_proof**2 - previous_proof**2).encode()
Out[6]: b'5'

In [7]: hashlib.sha256(str(new_proof**2 - previous_proof**2).encode())
Out[7]: <sha256 HASH object @ 0x181656df08>

In [8]: hashlib.sha256(str(new_proof**2 - previous_proof**2).encode()).hexdigest()
Out[8]: 'ef2d127de37b942baad06145e54b0c619a1f22327b2ebbcfbec78f5564afe39d'

In [9]:
```

Overview | GET http://127.0.0.1:50... | No Environment

http://127.0.0.1:5000/get_chain

Save

GET http://127.0.0.1:5000/get_chain Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
Key	Value	Description	

Body Cookies Headers (4) Test Results Status: 200 OK Time: 2.29 s Size: 252 B Save Response

Pretty Raw Preview Visualize JSON


```
{
    "chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2022-01-09 04:00:04.012103"
        }
    ],
    "length": 1
```

Overview | GET http://127.0.0.1:50... | No Environment

http://127.0.0.1:5000/mine_block

Save

GET http://127.0.0.1:5000/mine_block Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Body Cookies Headers (4) Test Results Status: 200 OK Time: 122 ms Size: 349 B Save Response

Pretty Raw Preview Visualize JSON


```
{
    "index": 4,
    "message": "Congratulations, you just mined a block!",
    "previous_hash": "34992017aeae0337f40d4acba667037ff7a8c8311c143c8519ac81ada10904da",
    "proof": 21391,
    "timestamp": "2022-01-09 04:08:36.839584"
}
```

Overview | GET http://127.0.0.1:50... | No Environment

http://127.0.0.1:5000/get_chain

Save

GET http://127.0.0.1:5000/get_chain Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Ed
Key	Value	Description	

Body Cookies Headers (4) Test Results Status: 200 OK Time: 20 ms Size: 700 B Save Response

Pretty Raw Preview Visualize JSON


```
{
    "chain": [
        {
            "index": 1,
            "previous_hash": "0",
            "proof": 1,
            "timestamp": "2022-01-09 04:00:04.012103"
        },
        {
            "index": 2,
            "previous_hash": "18d234b57088d8325a02c28f059249159e823684661f1abce0588bcb490c1a27",
            "proof": 533,
            "timestamp": "2022-01-09 04:07:07.583005"
        },
        {
            "index": 3,
```

Overview GET http://127.0.0.1:50... No Environment

http://127.0.0.1:5000/is_valid Save

GET http://127.0.0.1:5000/is_valid Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (4) Test Results Status: 200 OK Time: 14 ms Size: 181 B Save Response

Pretty Raw Preview Visualize JSON


```
1 {
2     "message": "Blockchain is valid !"
3 }
```