Bitcoin and its Distributed Ledger Technology

Emmanuelle Anceaume
Joint work with Romaric Ludinard, Thibaut Lajoie-Mazenc and Bruno Sericola

Algotel 2017, Quiberon
What is Bitcoin

Ledger

Bob -> Alice ฿0.001
Chunk -> Sara ฿0.05
Eva -> Alice ฿0.009
Alice -> John ฿0.02
Bob -> Chunk ฿0.7
Peter -> Bob ฿0.008
Bob -> Alice ฿0.05
Bob -> Alice ฿0.046
Bob -> Alice ฿0.008
What is Bitcoin

Bitcoin and its Distributed Ledger Technology
What is Bitcoin

Ledger

Bob -> Alice €0.001
Chunk -> Sara €0.05
Eva -> Alice €0.009
Alice -> John €0.02
Bob -> Chunk €0.7
Peter -> Bob €0.008
Bob -> Alice €0.05
Bob -> Alice €0.046
Bob -> Alice €0.008
What is Bitcoin

≥ 2 000 US dollars
Basic principles

- Decentralized cryptocurrency
- Peer-to-peer network
- Trustless model
- Pseudonymous users

Bitcoin ingredients

- Participating entities
  - Users, Miners and Bitcoin nodes
- Data structures
  - Accounts
  - Transactions
  - Blockchain
A transaction is the data structure that allows coins transfer

- The input list
- The output list

### Transaction as Double-Entry Bookkeeping

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Value</th>
<th>Outputs</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input 1</td>
<td>0.10 BTC</td>
<td>Output 1</td>
<td>0.10 BTC</td>
</tr>
<tr>
<td>Input 2</td>
<td>0.20 BTC</td>
<td>Output 2</td>
<td>0.20 BTC</td>
</tr>
<tr>
<td>Input 3</td>
<td>0.10 BTC</td>
<td>Output 3</td>
<td>0.20 BTC</td>
</tr>
<tr>
<td>Input 4</td>
<td>0.15 BTC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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)
Validity checked by anyone → presence of a trusted third party superfluous

Transaction 7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18

INPUTS From
From (previous transactions Joe has received):
Joe

0.1005 BTC

OUTPUTS To
Output #0 Alice’s Address
0.1000 BTC (spent)
Transaction Fees: 0.0005 BTC

Transaction 0627052b6f28912f2703066a912ea577f2ce4da4ea5a5fb68a57286c345c2f2

INPUTS From
7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18
Alice

0.1000 BTC

OUTPUTS To
Output #0 Bob’s Address
0.0150 BTC (spent)
Output #1 Alice’s Address (change) 0.0845 BTC (unspent)
Transaction Fees: 0.0005 BTC

Transaction 2bbac8bb3a57a2363407ac8c16a67015ed2e88a488af58cf90299e0744d3de4

INPUTS From
0627052b6f28912f2703066a912ea577f2ce4da4ea5a5fb68a57286c345c2f2
Bob

0.0150 BTC

OUTPUTS To
Output #0 Gopesh’s Address
0.0100 BTC (unspent)
Output #1 Bob’s Address (change) 0.0845 BTC (unspent)
Transaction Fees: 0.0005 BTC
Transaction 08f794a8a28d8ba58daef1337ce4a88171f931dd858477db3889d7a
def5b917a from block 438070:
0100000001ba54aa54af3ca6247589210f47e3c617ba4219f84b61c8b8724381

cd2c448349010000006a47304402201e0b0555330b9ba6dc689aebebef04643

91a882c41a6650ab66f803179860a1802207d1b46c45d37e8a88fee49c3e02ad

b9cd3cbb4bb96ca313135e00a5c01f71a6b012102374f390070a14763707fe93

10a73eaf2b2221734d0ff0a0684078571e2a12e9efeffff0237b9190000000000

1976a9143d5b9da23ff21a211f101ee2adec37d6b797db7c88ac40420f000000

00001976a9144ce03f31d4bd8bc2932f14cea99f4d96edcdef0c88ac35af0600
Decoded:

- **Header**: `ver=1, vin.size=1, vout.size=2, nLockTime=438069`
- **Inputs**:
  - **ID**: 4983442cc814372b8c8614bf81942ba17c6e3470f21b97524a63caf54aa54ba
  - **Index**: 1 (input value: 0.03098035 BTC)
  - **scriptSig**:
    - **Signature**: 304402201e0b0555330b9ba6dc689aeebecf0464391a882c41a6650ab66f803179860a1802207d1b46c45d37e8a88fee49c3e02adb9cd3cc84bb96ca313135e00a5c01f71a6b[ALL]
    - **Key**: 02374f390070a14763707fe9310a73eaf2b2221734d0ff0a0684078571e2a12e9e
  - **nSeq**: 4294967294
Decoded (ctd):

- Outputs:
  - n: 0
    - value: 0.01685815 BTC
    - ScriptPubKey: OP_DUP OP_HASH160 3d5b9da23ff21a211f101ee2adec37d6b797db7c OP_EQUALVERIFY OP_CHECKSIG
  - n: 1
    - value: 0.01000000 BTC
    - ScriptPubKey: OP_DUP OP_HASH160 4ce03f31d4bdbc2932f14cea99f4d96edcdbc0c OP_EQUALVERIFY OP_CHECKSIG
Transaction Bitcoin

Bitcoin and its Distributed Ledger Technology
Double-spending situation

Given a Bitcoin account $a_i$, account $a_i$ is in a double-spending situation if there exist two transactions $T_1 = (I_1, O_1)$ and $T_2 = (I_2, O_2)$ such that:

$$T_1, T_2 \in \bigcup_p \mathcal{V}(p) \land a_i \in I_1 \cap I_2.$$

Conflict-free transaction

A transaction $T = (I, O)$ is conflict-free if $\forall a \in I$, $a$ is not involved in a double-spending situation and the transaction $T' = (I', O') \in \mathcal{V}(p)$ with $a \in O'$ is conflict-free.
Transactions usage

Seller

Client

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
Transactions usage

Signed transaction
Seller ← --------------------- Client
Transactions usage

Valid transaction?
Transactions usage

Yes !

Valid transaction?

Valid transaction?

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
Transactions usage

Here it is!

Seller

Client
Transactions usage

Seller

Client

Valid transaction?

Valid transaction?

Bitcoin and its Distributed Ledger Technology
Blockchain: a sequence of blocks

Block $n - 1$

Previous block

Proof of work

Transactions:

$T_0$

$T_1$

...
Blockchain: local view implementation

- Any locally valid data is embedded within a block
- Integrity proof: Merkle tree
- Resilience to sybil attacks: Hashcash Proof-of-Work
- Chaining with proof of integrity
Merkle root

$h_{0,0}(h_{1,0} || h_{1,1})$

$h_{1,0}(h_{2,0} || h_{2,1})$

$h_{2,0}(h_{3,0} || h_{3,1})$

$h_{3,0}(t_0)$

$h_{2,1}(h_{3,2} || h_{3,3})$

$h_{3,1}(t_1)$

$h_{2,2}(h_{3,4} || h_{3,5})$

$h_{3,2}(t_2)$

$h_{3,3}(t_3)$

$h_{2,3}(h_{3,6} || h_{3,6})$

$h_{3,4}(t_4)$

$h_{3,5}(t_5)$

$h_{3,6}(t_6)$

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
### Proof-of-Work

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>536870912</td>
</tr>
<tr>
<td>Parent block</td>
<td>000000000000000000000287d...</td>
</tr>
<tr>
<td>Merkle root</td>
<td>e1f2e6de5580056c0469ab8c8b3e4e04f611701cf36</td>
</tr>
<tr>
<td></td>
<td>0253182103dccec494a2b441ac1dc93314f1577ff8ce44cfe59...</td>
</tr>
<tr>
<td>fcbd4a74e1659</td>
<td></td>
</tr>
<tr>
<td>Date 2016-11-09</td>
<td></td>
</tr>
<tr>
<td>Time 14:54:31</td>
<td></td>
</tr>
<tr>
<td>Bits 402936180</td>
<td></td>
</tr>
<tr>
<td>Nonce</td>
<td></td>
</tr>
</tbody>
</table>

**Goal:** \( \text{SHA256} \circ \text{SHA256}(\text{header}) \leq \text{target} \)

**Every 2016 blocks:**

\[
target \leftarrow target \times \max\left(\frac{1}{4}, \min(4, \frac{\text{real}}{\text{expected}})\right)
\]
string=HelloWorld!, nonce=0, difficulty=000
string=HelloWorld!, nonce=0, difficulty=000,

HelloWorld!0 : 3f6fc92516327a1cc4d3dca5ab2b27aeedf2d459a77fa06fd3c6b19fb609106a
string=HelloWorld!, nonce=1, difficulty=000,

HelloWorld!0 : 3f6fc92516327a1cc4d3dca5ab2b27aeedf2d459a77fa06fd3c6b19fb609106a
HelloWorld!1 : b5690c48c2d0a09481186aaa99e4e090901ff2ac4d572e6706dfd30eefc22a27
string=HelloWorld!, nonce=2, difficulty=000,

HelloWorld!0 : 3f6fc92516327a1cc4d3dca5ab2b27aeedf2d459a77fa06fd3c6b19fb609106a
HelloWorld!1 : b5690c48c2d0a09481186aaa99e4e090901ff2ac4d572e6706dfd30eefc22a27
HelloWorld!2 : 5b6fd9c27fcb54ca23404d9428f081b7c9280ba6370e33a6a20b16f40ce76320
string=HelloWorld!, nonce=3, difficulty=000,

HelloWorld!0 : 3f6fc92516327a1cc4d3dca5ab2b27aeedf2d459a77fa06fd3c6b19fb609106a
HelloWorld!1 : b5690c48c2d0a09481186aaa99e4e090901ff2ac4d572e6706dfd30eefc22a27
HelloWorld!2 : 5b6fd9c27fcb54ca23404d9428f081b7c9280ba6370e33a6a20b16f40ce76320
HelloWorld!3 : 9c5d769416aa0ca894abf22bd17bd30fbb6959291423ae1903a9f86a1fe7ce78
....
string=HelloWorld!, nonce=94, difficulty=000,

HelloWorld!0 : 3f6fc92516327a1cc4d3dca5ab2b27aeedf2d459a77fa06fd3c6b19fb609106a
HelloWorld!1 : b5690c48c2d0a09481186aaa99e4e090901ff2ac4d572e6706dfd30eefc22a27
HelloWorld!2 : 5b6fd9c27fcb54ca23404d9428f081b7c9280ba6370e33a6a20b16f40ce76320
HelloWorld!3 : 9c5d769416aa0ca894abf22bd17bd30fbb6959291423ae1903a9f86a1fe7ce78
...
HelloWorld!94 : 7090a0e5d88cff635e42ea33fcd6091a058e9cdd58ab8cd5c21c1c70421e35c6
Hashcash Proof-of-Work

string=HelloWorld!, nonce=95, difficulty=000,

HelloWorld!0 : 3f6fc92516327a1cc4d3dca5ab2b27aeedf2d459a77fa06fd3c6b19fb609106a
HelloWorld!1 : b5690c48c2d0a09481186aa99e4e090901ff2ac4d572e6706df30eeefc22a27
HelloWorld!2 : 5b6fd9c27fc854ca23404d9428f081b7c9280ba6370e33a6a20b16f40ce76320
HelloWorld!3 : 9c5d769416aa0ca894abf22bd17bd30fbb6959291423ae1903a9f86a1fe7ce78
....
HelloWorld!94 : 7090a0e5d88c9f635e42eda33fcd6091a058e9cdd58ab8cd5c21c1c70421e35c6
HelloWorld!95 : b74f3b2cf1061895f880a99d1d0249a8cedf223d3ed061150548aa6212c88d43
Hashcash Proof-of-Work

string=HelloWorld!, nonce=96, difficulty=000,

HelloWorld!0 : 3f6fc92516327a1cc4d3dca5ab2b27aeedf2d459a77fa06fd3c6b19fb609106a
HelloWorld!1 : b5690c48c2d0a09481186aa99e4e090901ff2ac4d572e6706dfd30eeefc22a27
HelloWorld!2 : 5b6fd9c27fcb54ca23404d9428f081b7c9280ba6370e33a6a20b16f40ce76320
HelloWorld!3 : 9c5d769416aa0ca894abf22bd17bd30fbb6959291423ae1903a9f86a1fe7ce78
...
HelloWorld!94 : 7090a0e5d88c6ff635e42ea33fcd6091a058e9cdd58ab8cd5c21c1c70421e35c6
HelloWorld!95 : b74f3b2cf1061895f880a99d10249a8cedf223d3ed061150548aa6212c88d43
HelloWorld!96 : 447ca2fa886965af084808d22116edde4383cb9aa16fd1fbcf3db61421b9990b9
string=HelloWorld!, nonce=97, difficulty=000,

HelloWorld!0 : 3f6fc92516327a1cc4d3dca5ab2b27aeedf2d459a77fa06fd3c6b19fb609106a
HelloWorld!1 : b5690c48c2d0a09481186aaa99e4e090901ff2ac4d572e6706dfd30eefc22a27
HelloWorld!2 : 5b6fd9c27fcb54ca23404d9428f081b7c9280ba6370e33a6a20b16f40ce76320
HelloWorld!3 : 9c5d769416aa0ca894abf22bd17bd30fbb6959291423ae1903a9f86a1fe7ce78
...
HelloWorld!94 : 7090a0e5d88c6f635e42ea33fcd6091a058e9cdd58ab8cd5c21c1c70421e35c6
HelloWorld!95 : b74f3b2cf1061895f880a99d10249a8cedf223d3ed061150548aa6212c88d43
HelloWorld!96 : 447ca2fa886965af084808d22116edde4383cba16fd1fbcf3db61421b9990b9
HelloWorld!97 : 000ba61ca46d1d317684925a0ef070e30193ff5fa6124aff76f513d96f49349d
Block generation

- Bitcoin minting = creating blocks
- Computationally hard...
- ...but far from being impossible
- Result: easy to check
- Goal: find a valid PoW for the current blockchain
- Average generation time: 10 minutes
Blockchain construction

$$((B_0, \perp), (B_0, \perp), (B_0, \perp), (B_0, \perp), (B_0, \perp), (B_0, \perp), (B_0, \perp))$$

Thanks to Romaric

Algotel 2017, Quiberon
Blockchain construction

Thanks to Romaric
Blockchain construction

\[(B_0, \bot) \quad (B_1, B) \quad (B_0, \bot) \quad (B_0, \bot) \quad (B_0, \bot) \quad (B_0, \bot) \quad (B_0, \bot)\]

Thanks to Romaric

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
Blockchain construction

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(B₁, B)</td>
<td>(B₁, B)</td>
<td>(B₁, B)</td>
<td>(B₁, B)</td>
<td>(B₁, B)</td>
<td>(B₁, B)</td>
<td>(B₁, B)</td>
</tr>
<tr>
<td>(B₀, ⊥)</td>
<td>(B₀, ⊥)</td>
<td>(B₀, ⊥)</td>
<td>(B₀, ⊥)</td>
<td>(B₀, ⊥)</td>
<td>(B₀, ⊥)</td>
<td>(B₀, ⊥)</td>
</tr>
</tbody>
</table>

A  B  C  D  E  F  G

Thanks to Romaric

Algotel 2017, Quiberon
Blockchain construction

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>((B_5, E))</td>
<td>((B_5, E))</td>
<td>((B_5, E))</td>
<td>((B_5, E))</td>
<td>((B_5, E))</td>
<td></td>
</tr>
<tr>
<td>((B_4, A))</td>
<td>((B_4, A))</td>
<td>((B_4, A))</td>
<td>((B_4, A))</td>
<td>((B_4, A))</td>
<td></td>
</tr>
<tr>
<td>((B_3, C))</td>
<td>((B_3, C))</td>
<td>((B_3, C))</td>
<td>((B_3, C))</td>
<td>((B_3, C))</td>
<td></td>
</tr>
<tr>
<td>((B_2, E))</td>
<td>((B_2, E))</td>
<td>((B_2, E))</td>
<td>((B_2, E))</td>
<td>((B_2, E))</td>
<td></td>
</tr>
<tr>
<td>((B_1, B))</td>
<td>((B_1, B))</td>
<td>((B_1, B))</td>
<td>((B_1, B))</td>
<td>((B_1, B))</td>
<td></td>
</tr>
<tr>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
<td></td>
</tr>
</tbody>
</table>

Thanks to Romaric
Transient inconsistencies

Thanks to Romaric
### Transient inconsistencies

<table>
<thead>
<tr>
<th>Block 1</th>
<th>Block 2</th>
<th>Block 3</th>
<th>Block 4</th>
<th>Block 5</th>
<th>Block 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>((B_6, A))</td>
<td>((B_5, E))</td>
<td>((B_4, A))</td>
<td>((B_3, C))</td>
<td>((B_2, E))</td>
<td>((B_1, B))</td>
</tr>
<tr>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
<td>((B_0, \bot))</td>
</tr>
</tbody>
</table>

Thank you to Romaric

**Algotel 2017, Quiberon**  
**Bitcoin and its Distributed Ledger Technology**
Transient inconsistencies

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block !</td>
<td>B0, ⊥</td>
<td>B1, B</td>
<td>B2, E</td>
<td>B3, C</td>
<td>B4, A</td>
<td>B5, E</td>
<td>B6, A</td>
</tr>
<tr>
<td>Block !</td>
<td>B0, ⊥</td>
<td>B1, B</td>
<td>B2, E</td>
<td>B3, C</td>
<td>B4, A</td>
<td>B5, E</td>
<td>B6, A</td>
</tr>
<tr>
<td>Block !</td>
<td>B0, ⊥</td>
<td>B1, B</td>
<td>B2, E</td>
<td>B3, C</td>
<td>B4, A</td>
<td>B5, E</td>
<td>B6, A</td>
</tr>
</tbody>
</table>

Thanks to Romaric
### Transient inconsistencies

<table>
<thead>
<tr>
<th>Block</th>
<th>Block</th>
<th>Block</th>
<th>Block</th>
<th>Block</th>
<th>Block</th>
<th>Block</th>
<th>Block</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(B_8, A)$</td>
<td>$(B_8, A)$</td>
<td>$(B_8, A)$</td>
<td>$(B_8, A)$</td>
<td>$(B_7, G)$</td>
<td>$(B_7, G)$</td>
<td>$(B_7, G)$</td>
<td>$(B_7, G)$</td>
</tr>
<tr>
<td>$(B_7, G)$</td>
<td>$(B_7, G)$</td>
<td>$(B_7, G)$</td>
<td>$(B_7, G)$</td>
<td>$(B_6, A)$</td>
<td>$(B_6, A)$</td>
<td>$(B_6, A)$</td>
<td>$(B_6, A)$</td>
</tr>
<tr>
<td>$(B_6, A)$</td>
<td>$(B_6, A)$</td>
<td>$(B_6, A)$</td>
<td>$(B_6, A)$</td>
<td>$(B_5, B)$</td>
<td>$(B_5, B)$</td>
<td>$(B_5, B)$</td>
<td>$(B_5, B)$</td>
</tr>
<tr>
<td>$(B_5, B)$</td>
<td>$(B_5, B)$</td>
<td>$(B_5, B)$</td>
<td>$(B_5, B)$</td>
<td>$(B_4, A)$</td>
<td>$(B_4, A)$</td>
<td>$(B_4, A)$</td>
<td>$(B_4, A)$</td>
</tr>
<tr>
<td>$(B_4, A)$</td>
<td>$(B_4, A)$</td>
<td>$(B_4, A)$</td>
<td>$(B_4, A)$</td>
<td>$(B_3, C)$</td>
<td>$(B_3, C)$</td>
<td>$(B_3, C)$</td>
<td>$(B_3, C)$</td>
</tr>
<tr>
<td>$(B_3, C)$</td>
<td>$(B_3, C)$</td>
<td>$(B_3, C)$</td>
<td>$(B_3, C)$</td>
<td>$(B_2, E)$</td>
<td>$(B_2, E)$</td>
<td>$(B_2, E)$</td>
<td>$(B_2, E)$</td>
</tr>
<tr>
<td>$(B_2, E)$</td>
<td>$(B_2, E)$</td>
<td>$(B_2, E)$</td>
<td>$(B_2, E)$</td>
<td>$(B_1, B)$</td>
<td>$(B_1, B)$</td>
<td>$(B_1, B)$</td>
<td>$(B_1, B)$</td>
</tr>
<tr>
<td>$(B_1, B)$</td>
<td>$(B_1, B)$</td>
<td>$(B_1, B)$</td>
<td>$(B_1, B)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
</tr>
<tr>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
<td>$(B_0, \bot)$</td>
</tr>
</tbody>
</table>

**Thanks to Romaric**

---

**Algotel 2017, Quiberon**

**Bitcoin and its Distributed Ledger Technology**
Bitcoin properties

Safety
If a transaction $T$ is deeply confirmed by some correct node, then no transaction conflicting with $T$ will ever be deeply confirmed by any correct node.

Liveness
A conflict-free transaction will eventually be deeply confirmed in the blockchain of all correct nodes at the same height in the blockchain.

- Ensures a common prefix shared by all correct peers,
- Need to wait for a deep confirmation
Strong Safety

If a transaction $T$ is confirmed by some correct node, then no transaction conflicting with $T$ will ever be confirmed by any correct node.

✓ An immediate and important consequence of this property is the capability for Bitcoin to safely handle fast payments.
- 16 billions bitcoins (1 bitcoin = 2,210 US dollars)
- Used in 120 countries
- 8 millions wallets created since 2009
- 25 millions US dollars are exchanged per day (208 millions US dollars on the 1rst Jan. 2017)
- During the last 12 months, computation power has been multiplied by 3 ($\approx 2,545,602,748 \text{GHs}^{-1}$)
- 250,000 transactions are confirmed per day
Bitcoin issues

Weakness of the consistency criteria

- “prefix strong consistency” [Garay, Kiayias, Leonardos 2015]
- “strong consistency” [Emin Gün Sirer’s blog]
- “eventual consistency” [Decker, Seidel, Wattenhofer 2016]
is part of the blockchain and which one is discarded. Each subsequent block takes another 10 minutes, so in order to know that a transaction is confirmed, we may need to wait for several hours. The Bitcoin system is a prime example of eventual consistency: Eventually Bitcoin has a consistent view of the transactions, but one can never be sure, and it may always happen that a blockchain fork will destroy a substantial amount of transactions, sometimes even multiple hours later [1].
Bitcoin Guarantees Strong, not Eventual, Consistency

Emin Gün Sirer
March 01, 2016 at 12:15 PM

It has somehow become a common adage that Bitcoin is eventually consistent. We now have both academics and developers claiming that Bitcoin provides a laughably weak consistency level that is reserved solely for first-generation NoSQL datastores.

All of these people are wrong.
Bitcoin Guarantees Strong, not Eventual, Consistency

The Fallacious Eventual Consistency Claim

The error in thinking that Bitcoin is eventually consistent stems from looking at the operation of the blockchain and observing that the last few blocks of a blockchain are subject to rearrangement. Indeed, in Bitcoin and cryptocurrencies based on Nakamoto consensus, the last few blocks of the blockchain may be replaced by competing forks.
Bitcoin Provides a Strong Consistency Guarantee

And a completely different picture emerges when we go through Bitcoin's read protocol. To wit, the read protocol for Bitcoin is to discard the last $\Omega$ blocks of the blockchain. Satoshi goes into a detailed analysis of what the value of $\Omega$ ought to be, and derives the equation as a function of the probability of observing an anomaly.
Scalability barrier

- Increase block size?
  - ✓ increases transactions throughput
  - ✗ increases block propagation time
- Reduce block interval?
  - ✓ decrease transactions latency
  - ✗ increases instability (re-organization)
**BIPs: Bitcoin improvement proposals**

**BIP: Decoupling blockchain operations**
- Leader election
- Transaction serialization

1. **PeerCensus**

2. **Bitcoin-NG**

3. **BizCoin**
Observation: Bitcoin blocks serve two purposes:
- Election of a leader to solve potential inconsistencies
- Verification of transactions

Idea: separation of both functionalities to increase transaction throughput
- keyblocks to elect leader
- microblocks to validate transactions
Decoupling the leader election operation from transactions validation operation.

- Block size: 1 MB
- 3.5 transactions per second
- Latency: 10 mn
Decoupling the leader election operation from transactions validation operation.
miner = leader
miner = leader Fork! Fork! Fork!

miner = leader Fork! Fork! Fork!

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
✓ increases transaction throughput
✗ forks are frequent
✗ users still need to wait
✗ The leader is free to propose double-spending transactions
Idea:
  - Combine
    - the decoupling scheme to improve transaction throughput
    - PBFT algorithm to get strong consistency

Solution:
  - Blockchain component:
    - Use PoW to dynamically build the consensus group
    - Prevent Sybil attacks
  - Chain Agreement component:
    - PBFT algorithm within the consensus group
PeerCensus [ICDCN2016]

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
PeerCensus [ICDCN2016]

PeerCensus is a tool used for monitoring and analyzing peer-to-peer networks. The diagram shows a network of miners connected through a Chain Agreement protocol, with blocks numbered 20, 21, 22, and 23. The commitment of block 23 is indicated, and the network is secured using the PBFT consensus algorithm.
PeerCensus [ICDCN2016]

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
PeerCensus [ICDCN2016]

✓ increases transaction throughput
✓ forks do not exist
✓ users do not have to wait
✗ PBFT with 436,000 nodes definitively cannot work
✗ once the group is corrupted, it remains corrupted forever
Idea:
- Combine both the decoupling scheme and PBFT algorithm

Solution
- Sybil attacks
  - Use PoW to dynamically build the consensus group
- \( \mathcal{O}(n^2) \) scalability issue
  - Use the PoW
  - Fixed-size sliding window
  - Rely on collective signing (CoSi algorithm)
BizCoin

PBFT algorithm

keyblock miner

leader

microblocks

key block

key block

key block

keyblock miner

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
BizCoin [Usenix 2016]

Bitcoin and its Distributed Ledger Technology
BizCoin [Usenix 2016]

CoSi algorithm

microblocks

leader

witnesses
BizCoin [Usenix 2016]

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
Bitcoin and its Distributed Ledger Technology
BizCoin [Usenix 2016]

microblocks

leader

Algotel 2017, Quiberon

Bitcoin and its Distributed Ledger Technology
BizCoin [Usenix 2016]

key block
microblocks
key block
microblocks
key block
microblocks
key block
microblocks

Algotel 2017, Quiberon
Bitcoin and its Distributed Ledger Technology
✓ increases transaction throughput
✗ forks may still exist
✗ users may still have to wait
✗ PBFT with 436,000 $w$ nodes works
✗ once the group is corrupted, it may refuse honest miners from joining
Hypothesis

- Large number of participants, \((N \approx 10000\) in Bitcoin\)
- Local copy of the blockchain \(B^{(p)}_k\) per participant \(p\)
- Proportion \(\mu \in (0, 1)\) of malicious miners
- Proportion \(1 - \mu \in (0, 1)\) of correct miners

Analysis

Algorithm compute \(B_k = (B_0, \ldots, B_{k-1})\) such that:

\[
\left( \forall p, B^{(p)} \subseteq B_k \right) \land \left( \forall \ell \in [0, k - 1], B_\ell \text{ mined by a correct miner} \right)
\]
Shared model

- Presence of a supervising group $\mathcal{E}_\ell$
- Any transaction must be accepted by $\mathcal{E}_\ell$
  - Bitcoin-NG: $\ell = 1$
  - Byzcoin: $\ell = w$
  - PeerCensus: $\ell = \infty$
- At each new epoch, $\mathcal{E}_\ell$ evolves. At epoch $k$,
  - if $|\mathcal{E}_\ell| < \ell$ the new leader is added to $\mathcal{E}_\ell$
  - otherwise the leader at epoch $k + 1 - \ell$ is removed from $\mathcal{E}_\ell$
    and the new leader is added
- $\mathcal{E}_\ell$ comprises the last $\ell$ successful miners:
Shared model

- Network of $n$ identical nodes without communication delays
- Fraction $\mu \in (0, 1)$ of miners is adversarial
- The first block was not generated by the adversary
Shared model

\[ B_k = (h, m) \in \mathbb{N}^* \times \mathbb{N}, \quad B_0 = (1, 0) \]

Process \( B = \{ B_k | k \geq 0 \} \) represents the evolution of the blockchain
- \( k \) = current blockchain size,
- \( h \) = number of blocks generated by correct miners,
- \( m \) = number of blocks generated by malicious miners.
$\mathbb{P}\{B_{\mu,k} = (h, m)\} = \binom{m + h - 1}{h - 1}(1 - \mu)^{h-1}\mu^m1_{\{k=m+h-1\}}$

Probability that none of the first $k$ blocks has been generated by a malicious miner

- $\mathbb{P}\{B_{\mu,k} = (h, 0)\} = (1 - \mu)^{k-1}$

With $\mu = 0.001$ and $k = 430,000$

- $\mathbb{P}\{B_{0.001,430000} = (h, 0)\} \approx 10^{-187}$
PeerCensus approach: $E_\infty$

- PBFT algorithm
- $E_\infty = \{\text{successful miners}\}$
- Agree on the subsequent block
  $\Rightarrow$ on the composition of $E_\infty$
PeerCensus approach: $E_\infty$

- Byzantine resilient consensus algorithms
- $E_\infty = \{ \text{miners having created a block in the blockchain} \}$
- Agree on the subsequent block

$\Rightarrow$ on the composition of $E_\infty$

Correct as long as the following condition holds:

$$|\{p \in E_\infty \mid p \text{ malicious}\}| < |E_\infty|/3$$
PeerCensus approach: $E_∞$

- Byzantine resilient consensus algorithms
- $E_∞ = \{ \text{miners having created a block in the blockchain} \}$
- Agree on the subsequent block

$⇒$ on the composition of $E_∞$

$B_{μ,k} = (h, m) ∈ \mathbb{N}^* × \mathbb{N}$, $B_0 = (1, 0)$

$S = \{(h, m) ∈ \mathbb{N}^* × \mathbb{N} | h ≥ 2m+1\}$, $P = \{(h, m) ∈ \mathbb{N}^* × \mathbb{N} | h ≤ 2m\}$
PeerCensus approach: $E_\infty$

$$\mathbb{P}\{B_\mu, k \in S\} = \lim_{k \to \infty} \mathbb{P}\{B_\mu, k \in S\} = \begin{cases} 0 & \text{if } \mu > \frac{1}{3} \\ 1/2 & \text{if } \mu = \frac{1}{3} \\ 1 & \text{if } \mu < \frac{1}{3}. \end{cases}$$

$$\mathbb{P}\{B_\mu, k = (h, m)\} = \left(\frac{m + h - 1}{h - 1}\right)(1 - \mu)^{h-1}\mu^m1\{k=m+h-1\}.$$
PeerCensus approach: $E_\infty$

A random variable representing the number of epochs before pollution

$$\ell(\mu) = \lim_{k \to \infty} \mathbb{P}\{T > k\} = \begin{cases} 0 & \text{if } \mu > \frac{1}{3} \\ 1 - \frac{2\mu}{1-\mu} & \text{if } \mu \leq \frac{1}{3}. \end{cases}$$
PeerCensus approach: $E_\infty$

\begin{itemize}
  \item[(e)] $\mathbb{P}\{T > k\}$ as a function of $\mu$ and (f) Asymptotic behavior of $\mathbb{P}\{T > k\}$ as a function $\mu$
\end{itemize}
Bizcoin approach: $E_w, 1 < w < \infty$

- PBFT algorithm
- $E_\infty = \{\text{miners having created a block in the last } w \text{ blocks}\}$
- Deterministic agreement on the subsequent block
  - $\Rightarrow$ on the miner added to $E_w$
  - $\Rightarrow$ on the one removed from $E_w$

Correct as long as the following condition holds:

$$|\{p \in E_\infty \mid p \text{ malicious}\}| < \frac{|E_w|}{3}$$
Bizcoin approach: $E_w, 1 < w < \infty$

- Byzantine resilient consensus algorithms
- $E_\infty = \{ \text{miners having created a block in the last } w \text{ blocks} \}$
- Agree on the subsequent block
  \( \Rightarrow \) on the miner added to $E_w$
  \( \Rightarrow \) on the one removed from $E_w$

$B_k = (h, m) \in \mathbb{N}^* \times \mathbb{N}$, $B_0 = (1, 0)$

$S_w = \{ (m_0, \ldots, m_{w-1}) \in \{0, 1\}^w \mid \sum_{i=0}^{w-1} m_i \leq (w - 1)/3 \}$,

$P_w = \{ (m_0, \ldots, m_{w-1}) \in \{0, 1\}^w \mid \sum_{i=0}^{w-1} m_i > (w - 1)/3 \}$.

$$
\mathbb{P}\{ W_k \in S_w \} = \sum_{\ell=0}^{(w-1)/3} \binom{w}{\ell} \mu^\ell (1 - \mu)^{w-\ell}.
$$
Bizcoin approach: $E_w, 1 < w < \infty$

Figure: $P\{W_k \in S_w\}$ as a function of $\mu$ and $w$
Bizcoin approach: $E_w, 1 < w < \infty$

Figure: Proportion of blocks mined by the most represented mining pools according to the epoch length $w$
Conclusion

- A blockchain is an immutable, unfalsifiable public ledger, used in particular in the Bitcoin crypto-currency.
- The technology is promising but immature.
- Blockchains mix elements of decentralised systems and cryptographic protocols.
- Pressing issues:
  - Bandwidth, latency, robustness, dependability.
  - Consensus by PoW is inefficient by design and has high external (ecological) cost.