The Internet Computer
Guest Lecture @ ETH Zurich

December 19, 2022
Thomas Locher (thomas.locher@dfinity.org)
Yvonne Anne Pignolet (yvonneanne@dfinity.org)
DFINITY

- Not-for-profit organization that develops the Internet Computer
- Founded in 2016
- Headquarters: Zurich, Switzerland
- Staff: +250
Outline

- What is the Internet Computer?
- Internet Computer Architecture
- Closer Look: Consensus
- Closer Look: HTTPS Outcalls
- The Internet Computer Today
What is the Internet Computer?

Platform to run any computation, using blockchain technology for decentralization and security.
ICP creates the Internet Computer blockchains

Guarantees safety and liveness of smart contract execution despite Byzantine participants

Coordination of nodes in independent data centers, jointly performing any computation for anyone
Canister smart contract

Data: Memory pages

Code: WebAssembly bytecode
Deploying and Using Canisters

DEPLOY

UX

User

Developer
Internet Computer Architecture

Collection of *replicated state machines* that are tied together using advanced cryptography
Nodes in Independent Data Centers
Chain Key Cryptography

Single 48-byte public key

for a secret-shared private key

https://internetcomputer.org/how-it-works/chain-key-technology
Internet Computer Consensus

Assumption: \( n > 3f \)

Guarantees **agreement** even under asynchrony

Guarantees **termination** under partial synchrony

https://internetcomputer.org/how-it-works/consensus
Network Nervous System (NNS)

verify(    ,   )

Governance
Ledger
Registry
Cycle Minting Canister
(Plus a few others)

https://internetcomputer.org/nns
Subnet 1: Public key: ..., nodes: ...
Subnet 2: Public key: ..., nodes: ...
...
Registry
Subnets

Governance

Add subnet with public key and nodes …
Subnets

- Each canister is assigned to one subnet
- Each subnet is a **replicated state machine**
- A canister can call canisters on other subnets
- Subnets make the Internet Computer **scalable**!
Closer Look: Consensus
Consensus Properties

Messages are placed in blocks. We reach agreement using a blockchain.

The following properties must hold even if up to $f < \frac{n}{3}$ nodes misbehave:

- **Agreement (Safety)**: For any $i$, if two (honest) replicas think that the $i$th block is agreed upon, they must have the same block.
- **Termination (Liveness)**: For any $i$, at some point every (honest) replica will think that the $i$th block is agreed upon.
- **Validity**: all agreed upon blocks are valid.

We use $n = 4$, $f = 1$ in examples.
A block maker selects available messages and combines them into a block and broadcasts it.

We need more than one block maker in each round, otherwise the IC would not be fault tolerant!
A sequence of unpredictable random bits (called the random beacon) is used to rank block makers:

<table>
<thead>
<tr>
<th>Rank 0</th>
<th>Replica 1</th>
<th>Replica 4</th>
<th>Replica 2</th>
<th>Replica 3</th>
<th>Replica 4</th>
<th>Replica 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rank 1</td>
<td>Replica 4</td>
<td>Replica 3</td>
<td>Replica 3</td>
<td>Replica 1</td>
<td>Replica 2</td>
<td>Replica 1</td>
</tr>
<tr>
<td>Rank 2</td>
<td>Replica 3</td>
<td>Replica 1</td>
<td>Replica 4</td>
<td>Replica 4</td>
<td>Replica 1</td>
<td>Replica 3</td>
</tr>
<tr>
<td>Rank 3</td>
<td>Replica 2</td>
<td>Replica 2</td>
<td>Replica 1</td>
<td>Replica 2</td>
<td>Replica 3</td>
<td>Replica 4</td>
</tr>
</tbody>
</table>

Round 24  Round 25  Round 26  Round 27  Round 28  Round 29
The notarization process ensures that a *valid* block proposal is published for every round.

**Step 1**
Replica 1 receives a block proposal for height 30, building on some notarized height 29 block.

**Step 2**
Replica 1 sees that the block is valid, signs it, and broadcasts its *notarization* share.

**Step 3**
Replica 1 sees that replicas 3 and 4 also published their notarization shares on the block.

**Step 4**
3 notarization shares are sufficient approval: the shares are aggregated into a single full notarization. Block 30 is now notarized, and notaries wait for height 31 blocks.
Replicas may notary-sign multiple blocks to ensure that at least one block becomes fully notarized.

Step 1
Replica 1 receives a block proposal for height 30, building on some notarized height 29 block.

Step 2
Replica 1 sees that the block is valid, signs it, and broadcasts its notarization share.

Step 3
Replicas 1 sees another height 30 block, which is also valid, and it broadcasts another notarization share.

Step 4
Both height 30 blocks get enough support to become notarized.
Notarization with Block Maker Ranking

Multiple notarized blocks may exist at the same height!

One notarized block $b$ at a height $h = \text{Agreement up to } h$
Finalization

Replicas create finalization shares if they did not sign any other block at that height

Step 1
Replica 1 notary-signs block $b$ at height 30

Step 2
Replica 1 observes that block $b$ is fully notarized and will no longer notary-sign blocks at height $\leq 30$

Step 3
Since replica 1 did not notary-sign any other block than block $b$, it signs block $b$, creating a finalization-share on $b$

Step 4
Replicas 2 and 4 also cast finalization shares on block $b$

Step 5
3 finalization-shares are sufficient approval: the shares are aggregated into a single full finalization
Finalization

Finalization on block \( b \) at height \( h \) = Proof that no other block is notarized at height \( h \)

The chain up to this block is final
Closer Look: HTTPS Outcalls
Interaction with Web 2.0

On-chain world

Smart contract

Off-chain world

Smart contracts cannot access Web 2.0 resources directly!
Interaction with Web 2.0

An oracle infrastructure is required. The oracles must be trusted!
Interaction with Web 2.0 on the Internet Computer

On-chain world

Off-chain world

• Replicas can communicate with web servers directly (no intermediary!)

• Responses go through consensus to ensure deterministic behavior
The Internet Computer Today
Live Since May 2021!

https://dashboard.internetcomputer.org
Growing Blockchain Ecosystem

[Images of various blockchain ecosystem projects]

[Links:
- DSCVR: dscvr.one
- Distrikt: az5d-cqaaa-aaaq-cai
- Internet Identity: identity.ic0.app
- OpenChat: oc.app
- IC Drive: icdrive.co
- Papyrs: app.papy.rs
- Dank: dark.cco
- Texas Hold'em: Im5fi-aayaa-aahsa-cai

https://internetcomputer.org/showcase]
Many Distributed Systems Problems

• Disseminating messages among nodes in the same subset
• Disseminating messages between subnets
• Scheduling and concurrent execution of canister messages
• Enabling nodes to catch up
• Handling churn (adding/removing nodes)
• Guaranteeing consistency
• Upgrading to next protocol version
• Creating new subnets
• Load balancing
• …
## Internet Computer vs. ...

<table>
<thead>
<tr>
<th></th>
<th>Bitcoin</th>
<th>Ethereum</th>
<th>ICP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Average block time</strong></td>
<td>1 block / 10 minutes</td>
<td>1 block / 15 seconds</td>
<td>37 blocks / second</td>
</tr>
<tr>
<td><strong>Finality</strong></td>
<td>1 hour</td>
<td>3 minutes</td>
<td>1-3 seconds</td>
</tr>
<tr>
<td><strong>TXs per second</strong></td>
<td>7</td>
<td>15</td>
<td>30,000 (write)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>2,000,000 (read)</td>
</tr>
<tr>
<td><strong>Validation data</strong></td>
<td>440 GB</td>
<td>750 GB</td>
<td>48 bytes</td>
</tr>
</tbody>
</table>
More information

- Infographic: https://internetcomputer.org/icig.pdf

- Technical Library:
  - https://www.youtube.com/playlist?list=PLuhDt1vhGcrfHG_rnRKsqZO1jL_Pd970h (videos)
  - https://medium.com/dfinity (blogposts)

- 200,000,000 CHF Developer Grant Program: https://dfinity.org/grants/

- DFINITY SDK: https://internetcomputer.org/docs/current/developer-docs/build/