Computer Systems - Notes Week 13

Ruben Schenk, ruben.schenk@inf.ethz.ch

January 9, 2022

Chapter 27: The Internet Computer

27.1 What Is The Internet Computer?

The **Internet Computer (IC)** is a platform to run any computation, using blockchain technology for decentralization and security.

The **Internet Computer Protocol (ICP)** is a protocol for the coordination of nodes in *independent* datacenters, jointly performing any computation for *anyone.* ICP creates the Internet Computer blockchains and guarantees safety and liveness of smart contract execution despite Byzantine participant.

The above idea is based on **canister smart contracts,** which are a combination of data (in memory pages) and code (in WebAssembly bytecode). Developers and users interact directly with canisters on the IC.

Scalability is achieved through nodes and subnets. Nodes are partitioned into *subnets.* Canister smart contracts are assigned to different subnets. One subnet is special: it hosts the **Network Nervous System (NNS)** canisters which govern the IC.

ICP token holders vote on:

- Creation of new subnets
- Upgrades to new protocol versions
- Replacement of nodes
- etc.

The **chain key technology** is based on the following three principles:

- One public key of NNS never changes and the nodes in the NNS share the same private key
- The NNS generates key for new subnets and certifies them
- Nodes in a subnet use these keys for secure communication

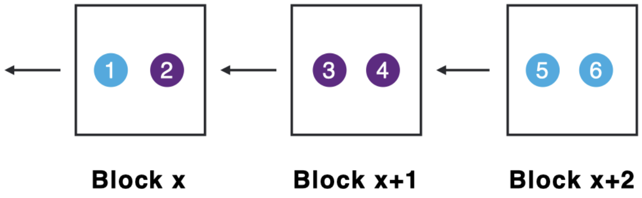
Each subnet is a replicated **state machine:**

- State: canisters and their queues
- Inputs: new canisters to be installed, messages from users and other canisters
- Outputs: responses to users and other canisters
- Transition function: message routing and scheduling as well as canister code

27.2 Consensus On The Internet Computer

Consensus orders the different messages in the network. Replicas may receive input messages in different orders, but must process them in the same order as the other replicas.

The **consensus properties** say that messages are placed in *blocks.* We reach an agreement using a blockchain.

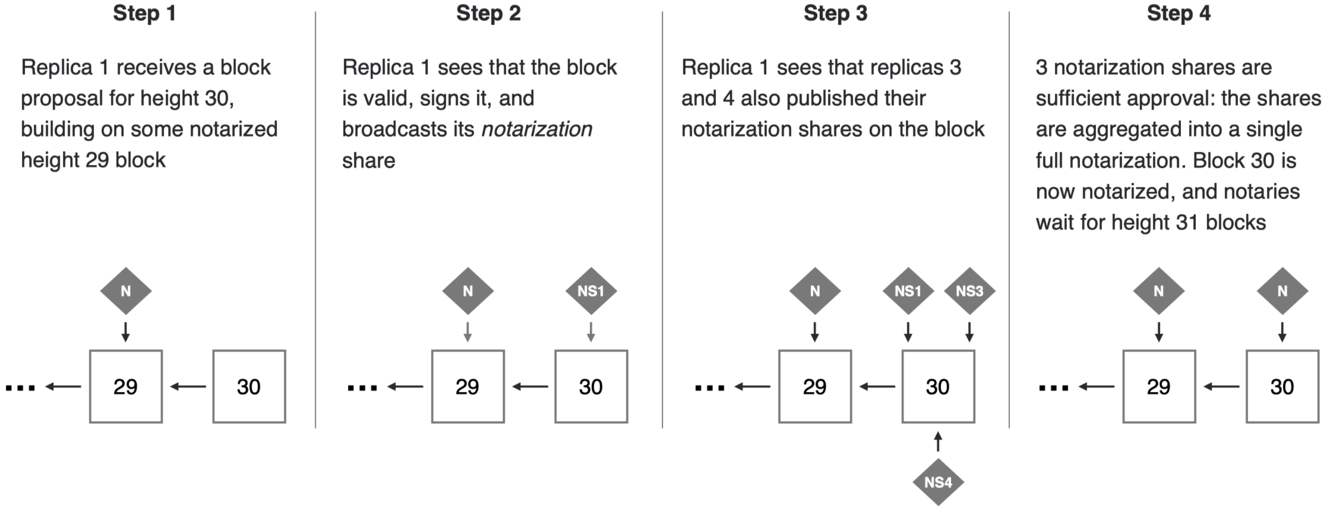


The following properties must hold even if up to $f < n/3$ nodes misbehave:

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

A **block maker** selects available messages and combines them into a block and broadcasts it. However, we need more than one block maker in each round, otherwise the IC would not be fault-tolerant.

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

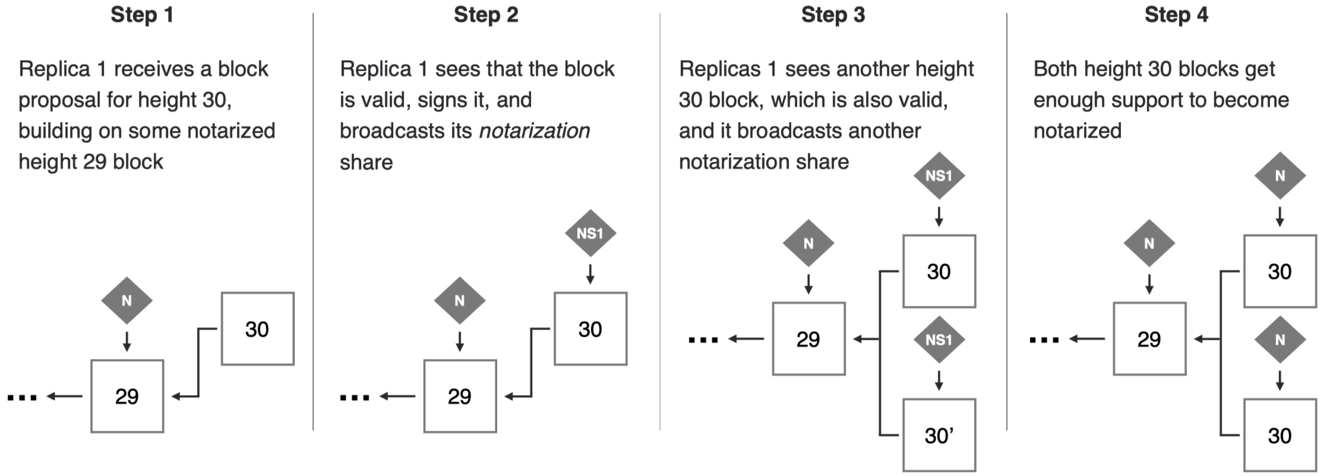
Step 3

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

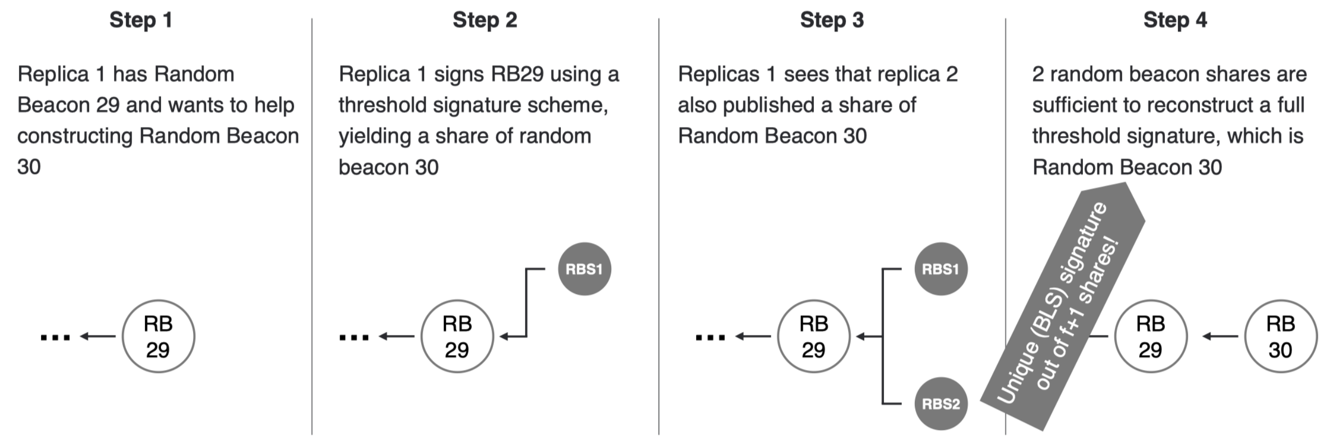
Step 3

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

Multiple notarized blocks may exist at the same height. At every height, there is a **Random Beacon,** an unpredictable random value shared by the replicas.



Step 1

Replica 1 has Random Beacon 29 and wants to help constructing Random Beacon 30

Step 2

Replica 1 signs RB29 using a threshold signature scheme, yielding a share of random beacon 30

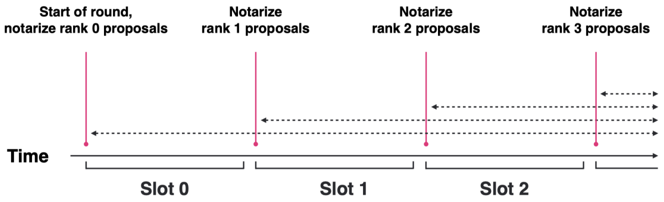
Step 3

Replicas 1 sees that replica 2 also published a share of Random Beacon 30

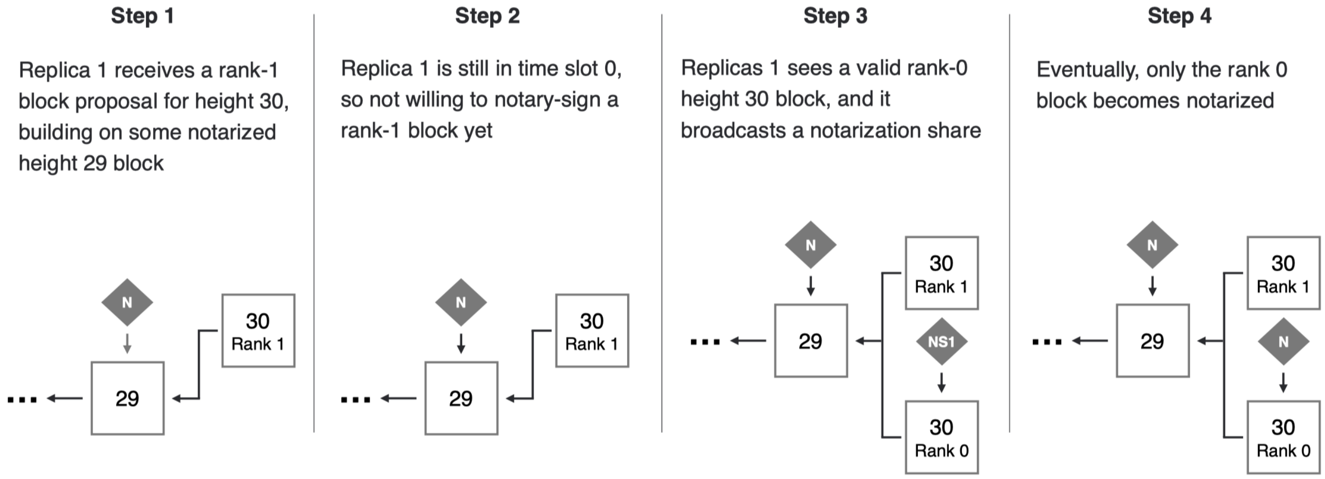
Step 4

2 random beacon shares are sufficient to reconstruct a full threshold signature, which is Random Beacon 30

The Random Beacon ranks block makers. Rounds are divided into time slots defining when block maker proposals are considered.


Start of round, notarize rank 0 proposals
Notarize rank 1 proposals
Notarize rank 2 proposals
Notarize rank 3 proposals
Time
Slot 0
Slot 1
Slot 2

Through notarization with block maker ranking we can reduce the number of notarized blocks.



Step 1

Replica 1 receives a rank-1 block proposal for height 30, building on some notarized height 29 block

Step 2

Replica 1 is still in time slot 0, so not willing to notary-sign a rank-1 block yet

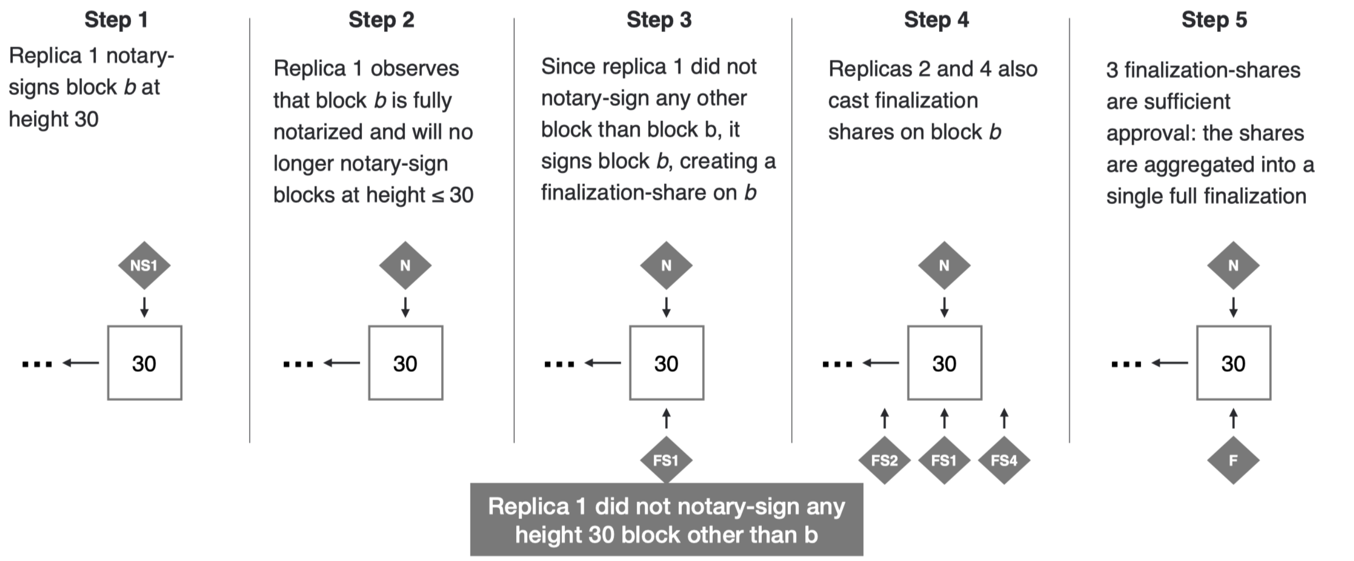
Step 3

Replicas 1 sees a valid rank-0 height 30 block, and it broadcasts a notarization share

Step 4

Eventually, only the rank 0 block becomes notarized

With **finalization,** replicas create finalization shares if they did not sign any other block at that height. This way, a finalization on block b at height h is a proof that no other block is notarized at height h.


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 ≤ 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
NS1
N
N
N
N
30
30
30
30
30
FS1
FS2
FS1
FS4
F
Replica 1 did not notary-sign any height 30 block other than b

Chapter 28: DeFi

28.1 Decentralized Finance

We introduce quickly some important terms about **finance:**

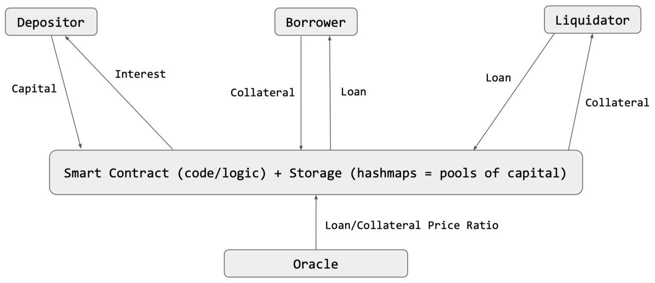
Money	Cryptocurrencies / Token
Banks	Blockchains / Lending Protocols / Vaults
Stocks	Tokens / Synths
Stock or Currency Exchanges	Automated Market Makers
PE / VC / Hedge-Funds	Decentralized Autonomous Organizations

28.2 Money & Banks

Money is native to the blockchain (the first level hashmap). The hashmap is the blockchain, which is everywhere. A **token** on the blockchain is essentially a nested hashmap.

Banks are simply blockchains or smart contracts, or, in other words, hashmaps and nested hashmaps.

28.3 Lending Protocol


Depositor
Borrower
Liquidator
Capital
Interest
Collateral
Loan
Loan
Collateral
Smart Contract (code/logic) + Storage (hashmaps = pools of capital)
Loan/Collateral Price Ratio
Oracle