

 > [How it works](#) > Infinite scalability

Infinite scalability

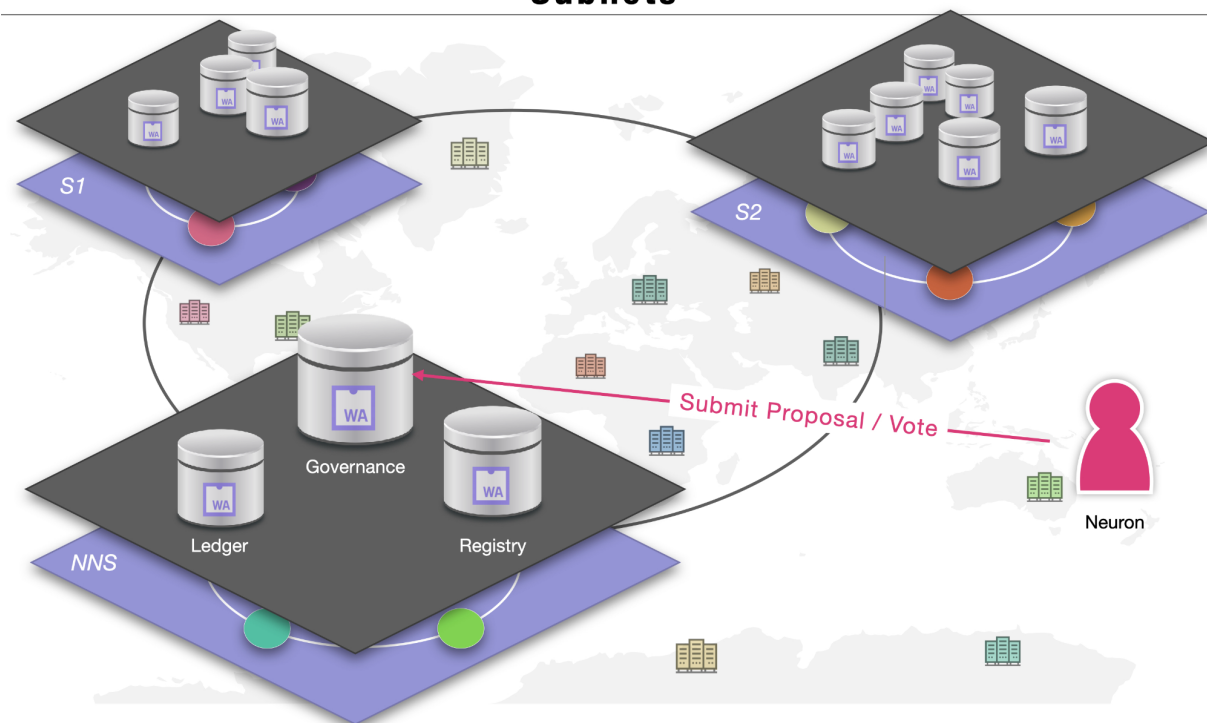
Ever wondered about the meaning behind DFINITY? It's Decentralized + Infinity. It's named that way because the Internet Computer is designed to scale infinitely. It means that the Internet Computer can host an unlimited number of canisters (smart contracts), store an unlimited amount of memory, process an unlimited amount of transactions per second. In simple words, Internet Computer is designed to host even large scale social media platforms in a fully decentralized way.

There are two types of widely-used approaches to improve the scalability of a system: (1) Vertical Scaling, and (2) Horizontal Scaling. Vertical scaling means adding more CPU, RAM and disk to a single computer. Horizontal scaling means adding more computers to the system. There is a limit to vertical scaling. But with horizontal scaling, one can achieve unlimited scalability. Internet Computer is one of the first blockchains to successfully use horizontal scaling.

The nodes in the Internet Computer are divided into subnets, each containing a few dozen nodes. The set of nodes in a subnet together maintain one blockchain. Each subnet can host thousands of canisters and process messages received by those canisters. Each subnet has a limited capacity in terms of the number of canisters (a few thousand), amount of storage (hundreds of GBs), and bandwidth (a few hundred transactions per second). But as more subnets are added to the Internet Computer, its overall capacity increases proportionately. There is no limit on the number of subnets that can be added.

Another crucial design aspect that is a prerequisite for limitless scaling is the inter-subnet communication of canisters: A canister of a subnet can send asynchronous messages to any canister on any other subnet. XNet messages are ingested by the receiving subnet's consensus layer and their integrity is validated based on the sending subnet's threshold signature — another application of [chain-key cryptography](#). This architecture of XNet messaging leads to a "loose coupling" of the subnets that does not require a central component such as a shard chain as used in other blockchains with multiple "shards" that would create a bottleneck when scaling out. Therefore newly added subnets can immediately send and receive XNet messages to any other subnet and an increasing number of subnets does not hit a natural bottleneck as in other, more simplistic, architectures.

Subnets



The Internet Computer is divided into subnets. Each subnet hosts many canisters. One of the subnets hosts Network Nervous System canisters.

Creating a new subnet has two steps. (1) Adding new nodes to the Internet Computer, and (2) Creating a subnet with the available nodes. Anyone can purchase the node hardware and add it to the Internet Computer by following the [node provider onboarding process](#).

We now describe how to create a new subnet with the available nodes. The Internet Computer has a decentralized governance system called Network Nervous System (NNS). Essentially, the NNS consists of a group of canisters that manage the Internet Computer. In the NNS, there is a component called “registry”, which stores the full configuration of the Internet Computer. The registry has a record for each subnet which includes a protocol version, the list of nodes in the subnet, protocol configuration parameters, etc.

Explorer / Proposals / 49018

Create new application subnet

Topic	Subnet Management
Type	Create Subnet
Status	Executed
Reward Status	Settled
Reward Weight	1
Date Created	2022-03-11, 10:43:23 AM UTC
Date Decided	2022-03-11, 2:04:25 PM UTC
Date Executed	2022-03-11, 2:11:41 PM UTC
Proposer	40

Voting Results

Updated 2022-03-11, 2:04:25 PM UTC

Yes 98.392 %

No 0.000 %

Voting Power 404815607

Voting Period Ended

Voting Power 1'880

There are two ways a proposal can be decided:

- Absolute Majority:** Before the voting period ends, a proposal is adopted or rejected if an absolute majority (more than half of the total voting power, indicated by "delimter above") has voted Yes or No on the proposal, respectively.
- Simple Majority:** When the voting period ends, a proposal is adopted if a simple majority (more than half of the votes cast) has voted Yes and those votes constitute at least 3% of the total voting power (indicated by "delimter above"). Otherwise, the proposal is rejected. Before a proposal is decided by Simple Majority, the voting period can be extended in order to "wait for quiet". Such voting period extensions occur when a proposal's voting results turn from either a Yes majority to a No majority or vice versa.

Proposal to create a new subnet. The status of all proposals can be viewed on the [IC Dashboard] (<https://dashboard.internetcomputer.org/governance>).

To add a new subnet, one has to submit a *proposal* to the NNS to add a record for a new subnet to the registry. The proposal consists of the list of nodes to be included in the new subnet. The proposal can be voted on by anyone who staked their ICP tokens. If a majority of voters accept the proposal, then the registry canister instructs the NNS subnet to generate — in a fully decentralized way using [chain-key cryptography](#) — the cryptographic key material to be used by the new subnet and a catch up package containing the genesis block. The registry canister then adds a record containing the configuration of the subnet.

We now describe how a new subnet is created after a record is added to the registry. Each node runs 2 processes, the (1) Replica and the (2) Orchestrator. The replica consists of the 4-layer software stack that maintains the blockchain. The orchestrator downloads and manages the replica software. When a new node is onboarded, the node provider has to install IC OS on the node, which contains the orchestrator software. The orchestrator regularly queries the NNS registry for any updates. If the orchestrator sees that the node is included in a registry record, then the orchestrator downloads the corresponding replica software, and runs the replica with the Catch Up Package included in the registry as input. The replica then starts accepting messages and the consensus protocol extends the genesis block present in the catch up package.