Realizing the vision of the World Computer

Learn how the Internet Computer blockchain realizes the vision of the World Computer. Dive into its technology, open-source repositories, in-depth video academy sessions, white papers, publications, and detailed technology articles.

Architecture

Core IC Protocol Chain-key technology Tokenomics & Governance Chain-evolution technology Smart c

Architecture of the Internet Computer

The Internet Computer (IC) realizes the vision of a ***World Computer*** – an open and secure ***blockchain-based network*** that can host programs and data in the form of smart contracts, perform computations on smart contracts in a secure and trustworthy way, and scale infinitely.

Smart contracts on the Internet Computer are called ***canister smart contracts***, or ***canisters***, each consisting of a bundle of ***WebAssembly (Wasm)*** bytecode and smart contract data storage. Each canister has its own, isolated, data storage that is only changed when the canister executes code.

Canisters are hosted on ***subnets***, the top-level architectural building block of the IC. A subnet is an independent blockchain, running on ***node machines***, or ***nodes***, deployed in globally-distributed data centers. A single subnet can securely host tens of thousands of canister smart contracts, totalling in hundreds of gigabytes of memory – there are currently dozens of subnets, growing to thousands in the future. For each canister hosted on a subnet, its code and data is stored on every node in the subnet, and its code is executed by every node in the subnet. This replication of storage and computation is essential to achieve ***fault tolerance***, so that canister smart contracts will continue to execute even if some nodes in the subnet are faulty (either because they crash, or even worse, are hacked by a malicious party). This replication is powered by the core ***Internet Computer Protocol (ICP)***, which implements a high-throughput, low-latency consensus mechanism and an efficient virtual machine for WebAssembly execution, backed by a blockchain.

The IC's multi-subnet architecture is much more powerful than the well-known sharding approach because it enables smart contracts on different subnets to communicate with each other seamlessly – much like services in a traditional **microservices architecture**, but fully on chain. Canisters communicate via ***asynchronous messages***, i.e., they don't block on sending a message, but process the response when it eventually arrives. This novel approach to inter-canister calls allows for scaling out the IC by simply adding more subnets.

The core ICP makes heavy use of ***chain-key cryptography***, a toolbox of advanced cryptographic protocols (based on **threshold cryptography**) that enables the decentralized operation of the IC with unprecedented scalability. Chain-key cryptography also includes a sophisticated collection of technologies for robustly and

nodes to easily join a subnet without validating every block beginning from the genesis

block, as in other blockchains). Another building block in the chain-key crypto toolbox are ***chain-key signatures***. They enable a canister to interact with (write to) other blockchains using threshold cryptography.

Having scalable and decentralized technology to power the operation of the network is not enough. In order to meet the requirements of complete decentralization, the IC needs a fully decentralized approach to governance. Governance of the IC platform is accomplished through a ***tokenized Decentralized Autonomous Organization (DAO)***, which is called the ***Network Nervous System (NNS)***. Each individual dapp on the IC can have its own governance system similar to the NNS by customizing and deploying an out-of-the-box tokenized DAO based on the ***Service Nervous System (SNS)*** for the dapp.

The **Internet Computer** was launched and open-sourced on May 10th 2021 by the DFINITY Foundation. The Internet Computer is now an independent network controlled by ICP token holders but DFINITY continues supporting its evolution.

Go deeper

Core IC Protocol

Overview

The Internet Computer is powered by the Internet Computer Protocol (ICP), from which its utility token, the ICP token, derives its name. The core part of the IC protocol, the ***core IC protocol***, is a 4-layer protocol that is running on the nodes of each subnet. By running the core IC protocol, the nodes of a subnet realize a blockchain-based ***replicated state machine*** that makes progress independently of the other subnets (but communicates asynchronously with them). This architecture of many concurrently-operating subnets enables the IC to scale practically without limits. Subnets process ***messages***, which are submitted by users or come from other subnets.

The core IC protocol comprises the following four layers, from bottom to top:

1. Peer-to-peer
2. Consensus
3. Message routing
4. Execution

The lower two layers, P2P and consensus, together implement a ***selection and ordering*** of incoming messages and provide messages to the upper two layers in the form of ***blocks***. The upper two layers, message routing and execution, receive blocks containing ordered messages from the lower part of the stack and execute them in a completely deterministic manner on every node of the subnet. This realizes a replicated state machine, where every node in the subnet transitions from the same starting state to the same ending state in every round (it must be ensured that every node executes the same messages in the same order, i.e., fully deterministically).

Go deeper

Peer-to-peer

The peer-to-peer layer (P2P) of the Internet Computer, the bottommost layer in the protocol stack, is responsible for the secure and reliable communication between the nodes of a subnet. The P2P layer realizes a virtual peer-to-peer broadcast network between the nodes of a subnet, building upon the Internet Protocol (IP) connectivity between the nodes. This makes the P2P layer the communications fabric that connects all the nodes of a subnet. Using P2P, a node can broadcast a network message, also called ***artifact***, to all the nodes in the subnet. Artifacts can be things like input to canisters submitted by users or protocol messages (e.g., block proposals) generated by the IC protocol. P2P ensures that artifacts to be broadcast are eventually delivered to all nodes of the subnet. Eventual delivery reflects the asynchronous nature of real-world communication networks, which we assume for the Internet Computer protocol.

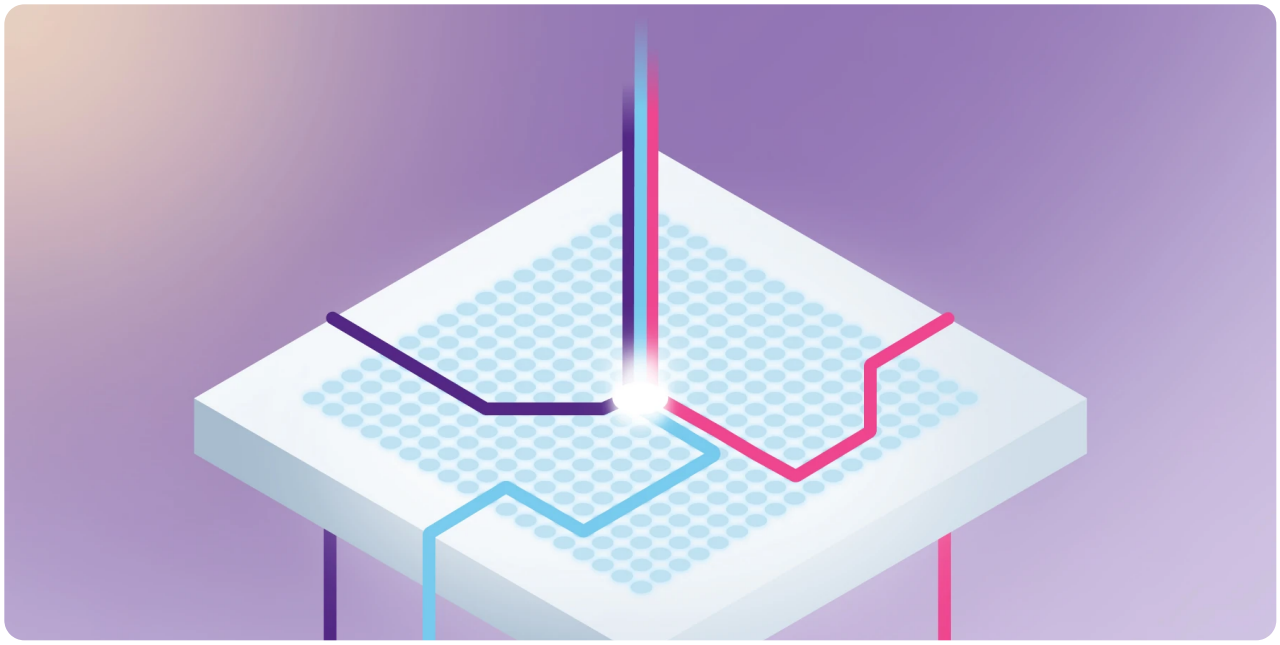
Go deeper

Consensus

Every blockchain needs a consensus mechanism that allows the nodes to agree on the messages to be processed, as well as their ordering. Consensus is the component of the core IC protocol that drives the subnets of the IC. Each subnet is a blockchain that runs the IC core protocol, including consensus, independently of the other subnets. The purpose of the consensus protocol is to output the same block of ordered messages on each node of a subnet in a given round so that each node can make the same state transition when deterministically executing those messages.

The IC's consensus protocol is designed to meet the following requirements: low latency (almost instant finality); high throughput; robustness (graceful degradation of latency and throughput in the presence of node or network failures). The IC consensus protocol also provides ***cryptographically guaranteed finality***. This is in contrast to Bitcoin-like protocols which only provides ***probabilistic finality***, where a block is considered final once a sufficient number of blocks have built on top of it in the blockchain.

Go deeper

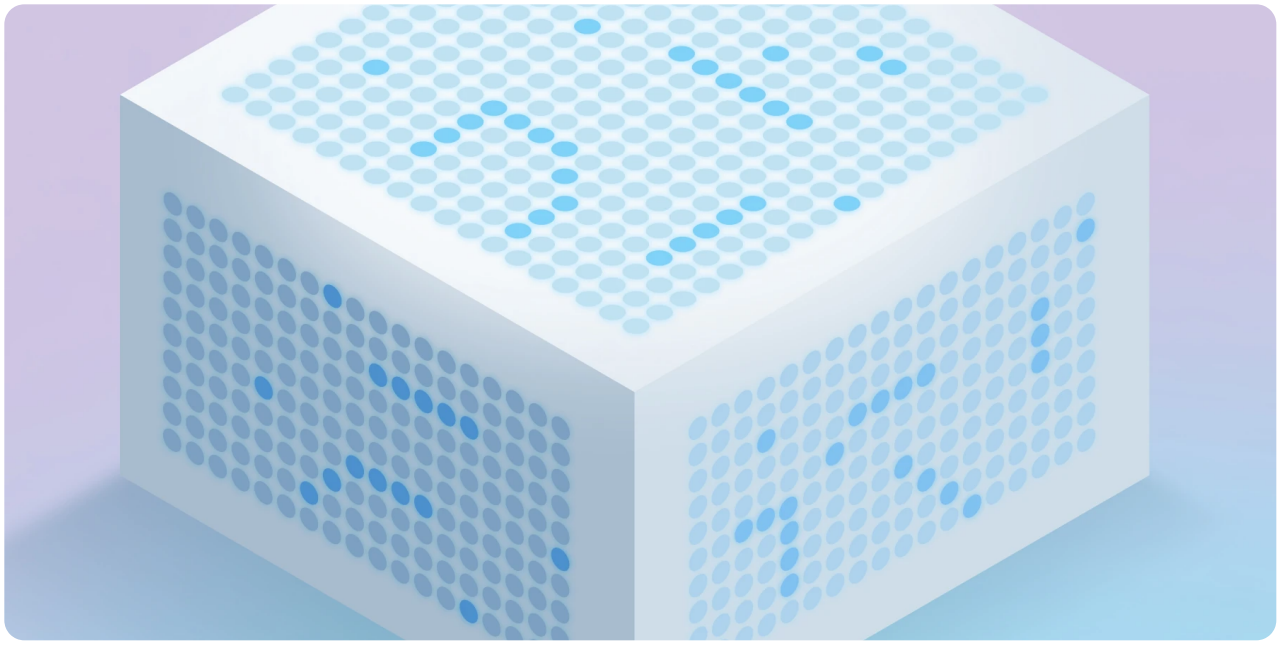


Message routing

In every IC round, the message routing component receives a block of messages to be processed from consensus – the same block on each node of the subnet – and places the messages into the input queues of their target canisters, a process called ***induction***. Then, it triggers the execution round which will potentially lead to new canister messages in the executed canisters' output queues. Once execution is done, the messages in the output queues are routed by the message routing component to the recipients.

The recipients may include canisters residing on a different subnet. The message routing layer implements the routing of inter-canister messages between subnets, such that those messages can be included in blocks and be inducted on the recipient's subnet. This is referred to as cross-subnet messaging or simply XNet messaging. ***Secure XNet messaging*** is a key ingredient for the architecture of loosely-coupled subnets and thus a prerequisite for the scalability of the IC.

Another crucial functionality implemented by the message routing layer is ***state certification***, that is, the subnet certifying parts of the replicated subnet state in every round in a decentralized manner. Among others, this certification is used by other subnets to verify the authenticity of the subnet-to-subnet streams or to allow users to authentically read the processing status of messages previously submitted by them. State certification and secure XNet messaging enable, among others, the secure and transparent communication of canisters across subnet boundaries, a challenge that any



Execution

The execution layer, the topmost layer of the core IC protocol stack, is responsible for executing canister smart contract code. Code execution is done by a ***WebAssembly (Wasm)*** virtual machine deployed on every node. WebAssembly bytecode can be executed deterministically, which is important for a blockchain system, and with near-native speed. Canister messages, i.e., ingress messages by users or messages by other canisters, have been inducted into the queues of the canisters on the subnet by message routing. Message routing then hands over control to the execution layer, which deterministically executes messages, either until all messages in the canisters' queues are consumed or the cycles limit for the round has been reached, to ensure bounded round times.

The execution layer has many unique features, which sets apart the IC from other blockchains:

1. *Deterministic time slicing (DTS)* - The execution of very large messages requiring billions of Wasm instructions to be executed can be split across multiple IC rounds. This capability of executing messages over multiple rounds is unique to the Internet Computer blockchain.
2. *Concurrency* - Execution of canister Wasm bytecode is done *concurrently* on multiple CPU cores, which is possible due to each canister having its own isolated state.

that require randomness.

Go deeper

Chain-key technology

Chain-key cryptography

The Internet Computer protocol uses a toolbox of advanced cryptographic mechanisms, collectively known as ***chain-key cryptography***, which allows the IC to achieve functionalities and scalability that are impossible on other blockchains.

A key component of chain-key cryptography is a ***threshold signature scheme***, which is like an ordinary digital signature scheme, except that the secret signing key is distributed among all the replicas in a subnet in such a way that the key cannot be stolen by compromising one (or even a large fraction) of the replicas in the subnet. The technology has many benefits including:

2. The topology of IC can evolve autonomously -- New nodes and subnets can be added, faulty nodes can be recovered and protocol can be upgraded autonomously.
3. A source of unpredictable and unbiasable pseudo-random numbers for canisters. Canisters can securely run algorithms that need randomness.

Go deeper

Chain-key signatures

Chain-key signatures extends chain-key technology to allow transactions targeted at other blockchains to be computed fully on-chain using the Internet Computer Protocol. Using chain-key signatures, the IC can integrate with other blockchains such as Bitcoin and Ethereum in a completely trustless manner without needing any bridges. Canisters can now securely store and transact Bitcoin. The secret key of the Bitcoin is shared between all the nodes running the canister. The canister can trasact Bitcoin using a chain-key signed transaction only when at least 2/3rd of the nodes agree to make the transaction. Indeed, using chain-key signatures is the strongest, most decentralized way of integrating blockchains as no additional trust assumptions besides that of the two blockchains are required, particularly no additional parties that manage signature keys or their shares.

Bitcoin integration

The Bitcoin integration on the Internet Computer rests on two pillars: Chain-key signatures and a direct interaction between Internet Computer nodes and the Bitcoin peer-to-peer network. While chain-key signatures make it possible for canisters to have their own Bitcoin addresses and create valid transactions spending bitcoins held by these addresses, the direct message exchange between the Internet Computer and the Bitcoin network serves to maintain information about the Bitcoin blockchain state, such as address balances, in the Internet Computer and to transmit Bitcoin transactions originating from canisters to the Bitcoin network.

Go deeper

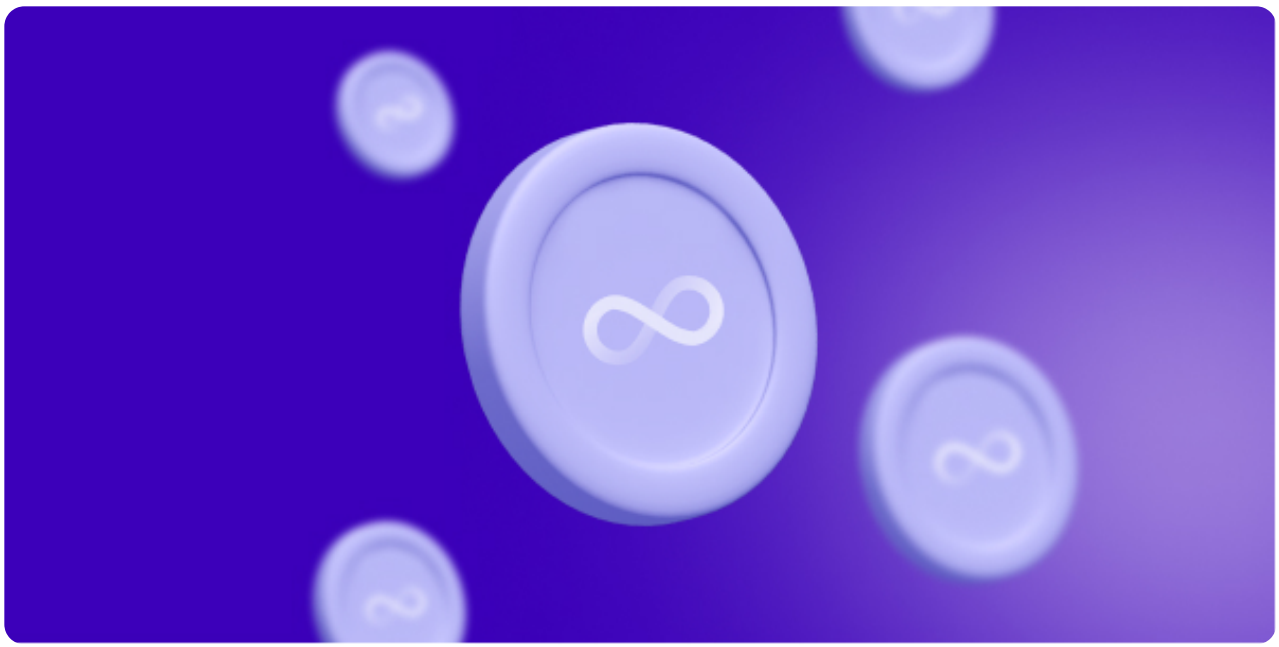
Chain-key tokens

Chain-key tokens — and Chain-Key Bitcoin (ckBTC) — are a cryptography-based replacement to wrapped tokens with strong decentralization advantages: Chain-key tokens eliminate the risks associated with the traditional intermediary-based token wrapping, while also having the same benefit of making a token from another blockchain available for transfers and trading. ***Chain-key cryptography*** makes this possible: Taking the example of Bitcoin, a canister smart contract can own ECDSA key pairs and derive Bitcoin addresses to which transfers of real Bitcoin can be made on the Bitcoin network. When receiving bitcoin, the canister mints and issues ckBTC in a 1:1 ratio to the sender of the bitcoin. Conversely, redeeming ckBTC for the underlying bitcoin removes the ckBTC from circulating supply and refunds the bitcoin. This makes a chain-key token a 'twin' of the original token with the same properties and valuation, but hosted on the Internet Computer.

Go deeper

Tokenomics & Governance

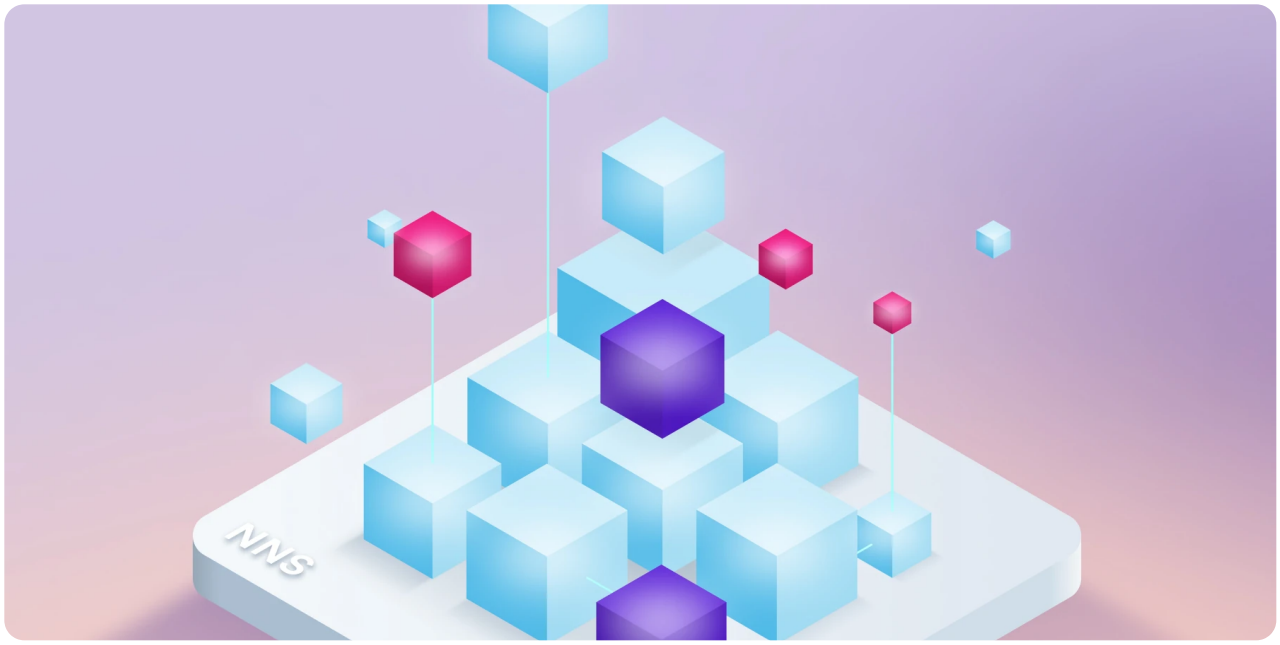
Core IC Protocol Chain-key technology Tokenomics & Governance Chain-evolution technology Smart c



Tokenomics

The Internet Computer makes use of an utility token called ICP. The ICP token is used for a variety of functions on the platform. Users can stake ICP giving them the right to vote and earn voting rewards. Investors can use ICP to participate in SNS swaps launched on the Internet Computer, i.e., the initial offerings of the native DAOs. Developers use ICP token to purchase cycles to power their dapps on the Internet Computer. And node providers are remunerated in ICP for the compute power they provide to the Internet Computer platform.

Go deeper

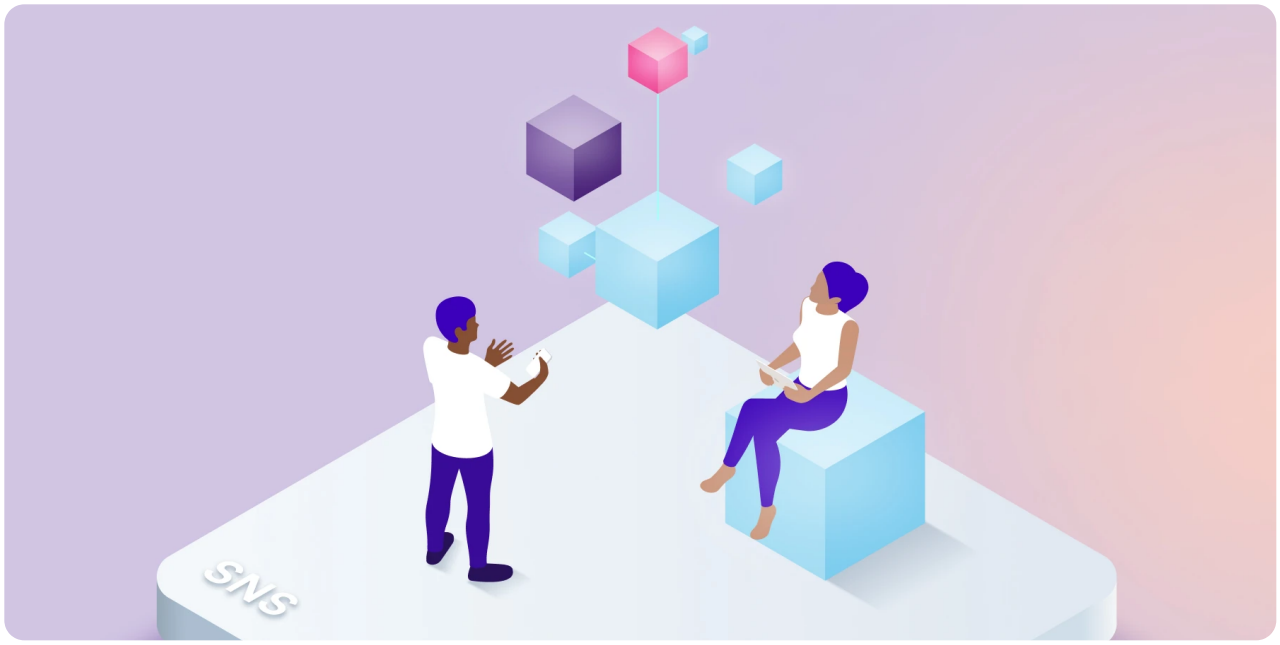


Network Nervous System

The Internet Computer is a ***decentralized*** system run by many independent nodes. The ***Network Nervous System (NNS)*** coordinates their effort by determining which subnet a node belongs to (the topology), which protocol version they should run, and when they should upgrade to a new protocol version.

NNS decision making is done via an open tokenized governance system. The NNS is one of the largest decentralized autonomous organizations (DAOs). Anyone can become a participant of the NNS by staking ICP tokens and contribute to decisions.

Go deeper



Service Nervous System (SNS)

Similarly to how the Internet Computer is controlled by the Network Nervous System, a decentralized application on the IC can be controlled by a community. The Internet Computer's built-in solution for a DAO (decentralized autonomous organization) that ***tokenizes*** and ***decentralizes*** a dapp is called the Service Nervous System (SNS). In the process of creating an SNS, new tokens are minted and sold in a community-based fundraising. The dapp's control is handed over to the SNS and everyone who has SNS tokens can contribute to decisions on how the dapp evolves going forward. This allows distributing the power and ownership of the dapp over a number of parties, thereby eliminating single points of failure and gaining censorship resistance. Moreover, the newly created tokens can be used to create incentives to foster user adoption and participation.

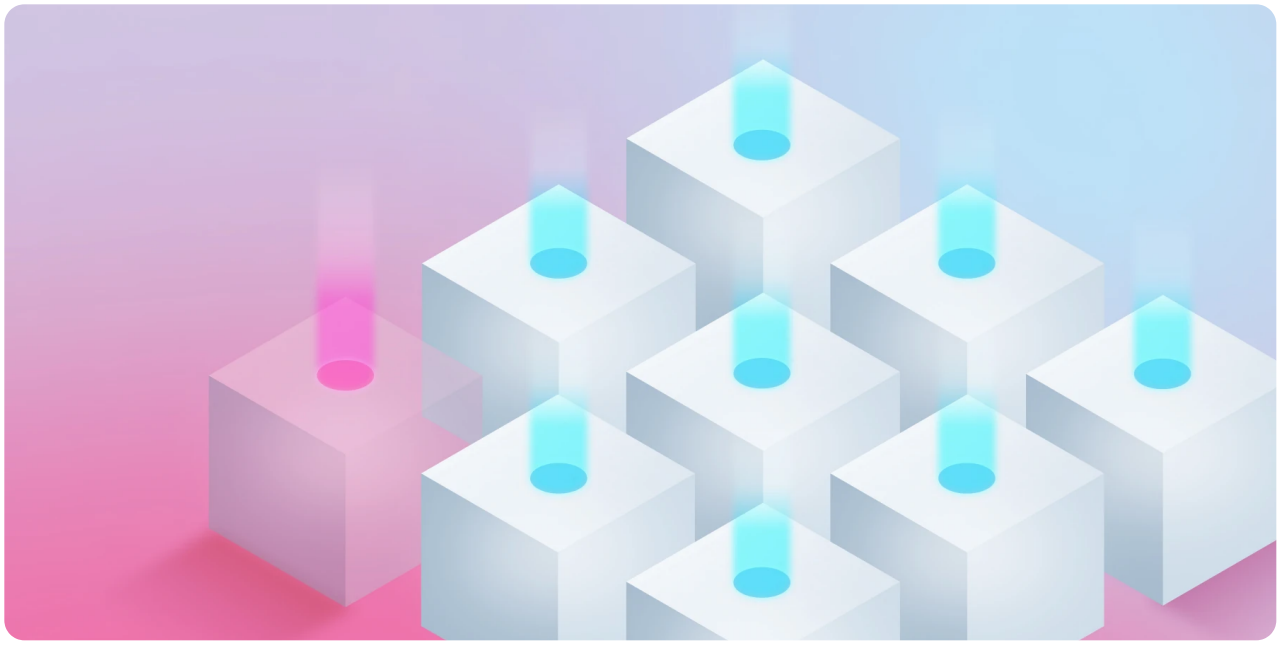
Go deeper

Chain-evolution technology

Infinite scalability

The Internet Computer scales its capacity horizontally by creating new subnets that host additional canisters — just like traditional cloud infrastructure scales by adding new machines. Once the IC's Network Nervous System (NNS) decided to create a new subnet, it selects a group of spare nodes that have joined the IC but have not yet been allocated to any subnet and creates the initial configuration of the new subnet. The selected group of nodes then begins to form a new subnet blockchain.

Go deeper



Fault tolerance

In any large-scale distributed system, it is inevitable that individual nodes fail due to hardware breaking, network connectivity issues, or even the owner deciding to remove the nodes from the network. In such cases, the IC's Network Nervous System selects a spare node that replaces the failed node in its subnet. The new node then joins the subnet and performs a state synchronization with its existing nodes and begins contributing to the subnet blockchain's consensus protocol.

Go deeper

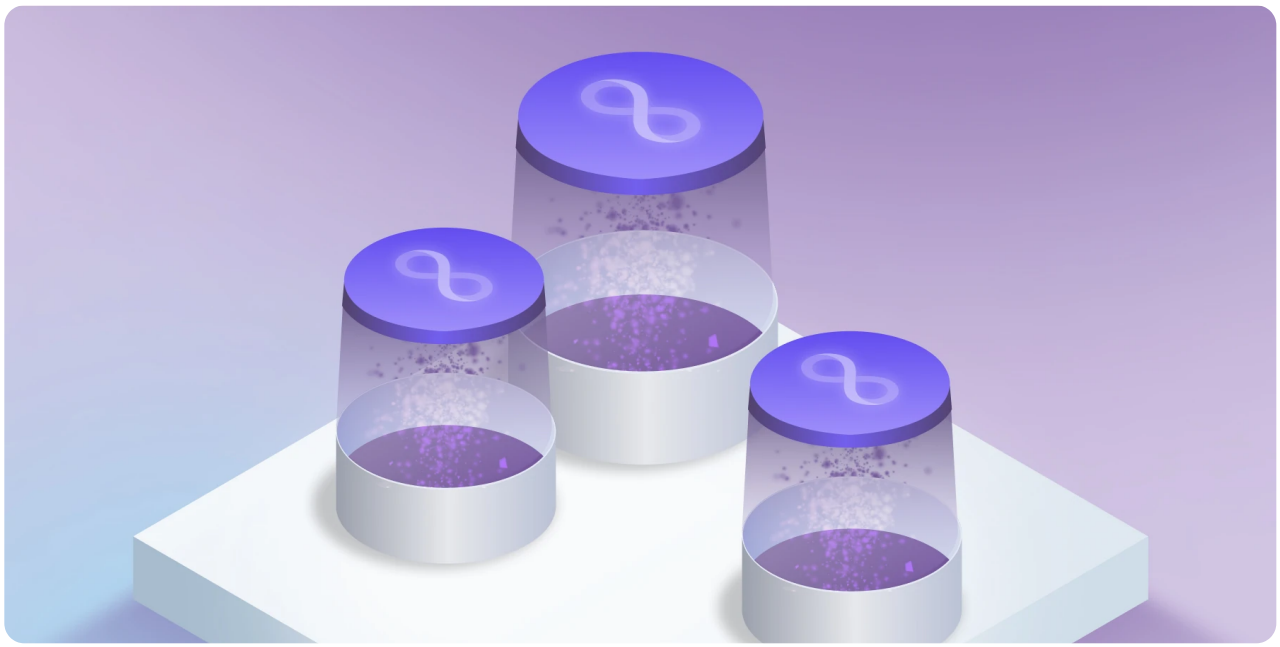
Core IC Protocol Chain-key technology Tokenomics & Governance Chain-evolution technology Smart c

Protocol upgrade

The Internet Computer blockchain is governed by the Network Nervous System (NNS), its algorithmic governance system. One of the many duties of the NNS is to orchestrate upgrades of the Internet Computer to a new protocol version when the community has adopted an upgrade proposal. Making upgrades to any blockchain requires solutions to several challenging problems posed by the nature of decentralized systems including how to allow arbitrary changes to the protocol, preserve state of all canister smart contracts, minimize downtime, and roll out upgrades autonomously.

Go deeper

Smart contracts

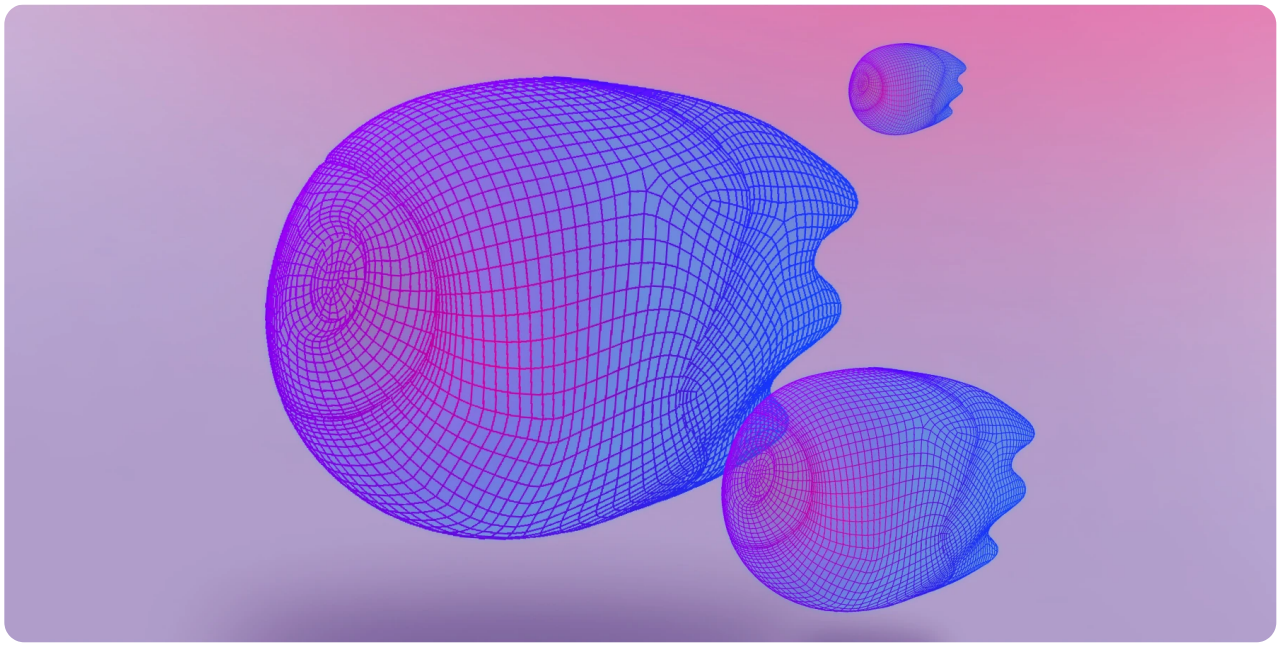


Canisters

Smart contracts on the Internet Computer come in the form of ***canisters***: computational

of each such message is done in complete isolation, allowing for massive levels of concurrent execution. Canisters are managed by controllers. Control structure of canisters could be centralized (e.g. when the controllers include some centralized entity), decentralized (when the controller is a DAO) or even non-existent, in which case the canister is an immutable smart contract. Controllers are the only entities which can deploy the canister to the Internet Computer, start/stop their execution and update their code. The controllers also need to ensure that canisters hold sufficient ***cycles***. These are the unit used on the IC to acquire resources for canister execution (memory, network bandwidth and computational power). To this end the IC monitors the resource usage of canisters and deducts their cost from a cycle balance maintained locally by each canister.

Go deeper



Motoko

Motoko is a new programming language for smart contracts. It is designed to seamlessly support the programming model of the Internet Computer and makes it easier to take advantage of the unique features of the blockchain. Motoko is strongly typed, actor-based, and has built-in support for orthogonal persistence and asynchronous message passing. Productivity and safety features include automatic memory management, generics, type inference, pattern matching, and both arbitrary- and fixed-precision

interoperability.

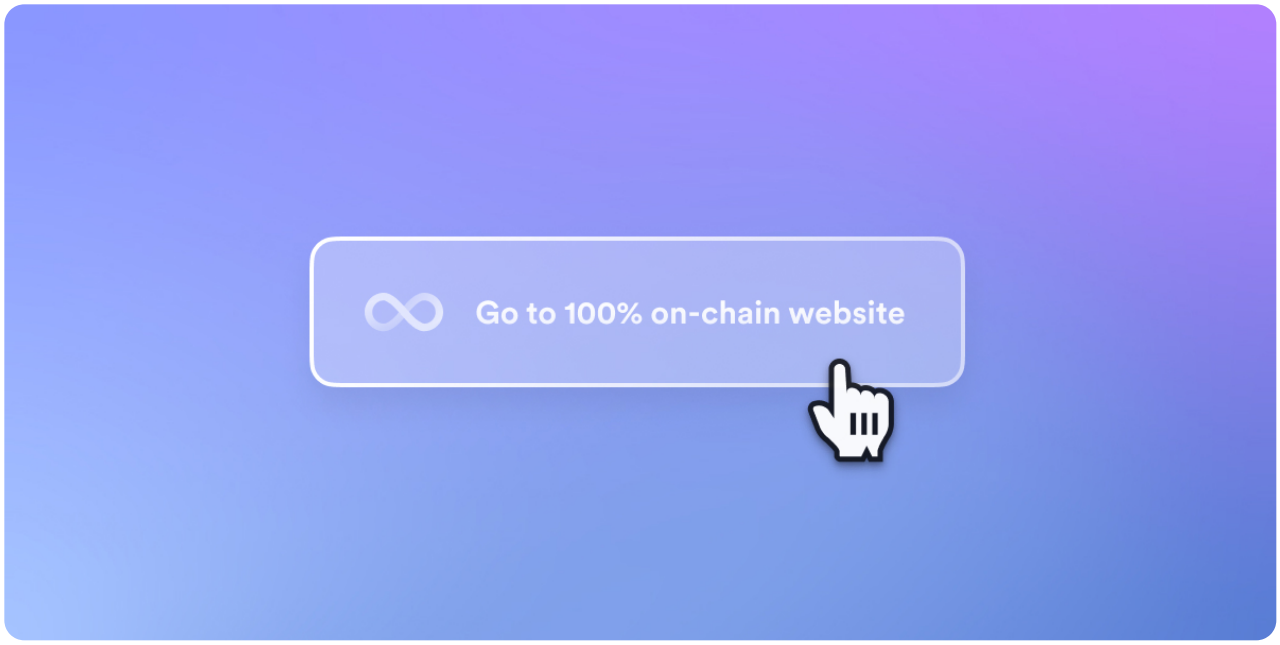
Go deeper

Certified variables

Canister smart contracts can declare variables as certified. Whenever set, these variables will automatically get a Merkle tree certificate, signed by the Internet Computer blockchain. This allows anyone to verify the authenticity of this type of data using the Internet Computer's public key.

Go deeper

Web access



Smart Contracts serve the web

The Internet Computer is the only blockchain that can host a full dapp – frontend, backend and data. This is a crucial and distinguishing feature allowing dapps to run 100% on-chain inheriting the security and decentralization of blockchain without sacrificing speed or affordability. This is possible because the IC can securely serve HTTP requests.

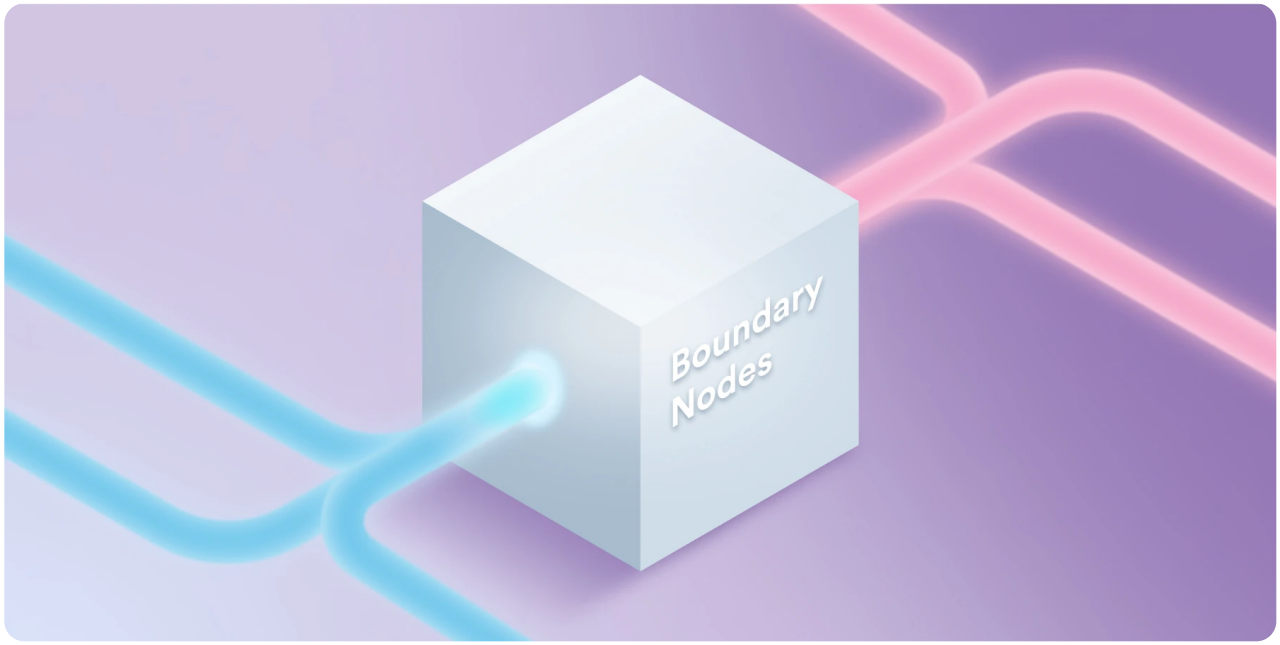
Go deeper

Core IC Protocol Chain-key technology Tokenomics & Governance Chain-evolution technology Smart c

Asset certification

Assets are served from the Internet Computer in a tamper-proof way using certification: Each asset is accompanied by a certificate, which is signed by the entire subnet allowing the user to verify that a response is correct and authentic even when communicating with a malicious node.

Go deeper



Boundary nodes

The boundary nodes are the gateway to the Internet Computer and enable seamless access to the canister smart contracts with stock browsers. They provide an HTTP endpoint and translate all incoming user requests to API canister calls, which are processed on-chain. In addition, the boundary nodes act as a cache to improve the performance of the dapps hosted on the Internet Computer.

Go deeper

Core IC Protocol Chain-key technology Tokenomics & Governance Chain-evolution technology Smart c

Internet Identity

The main means of identity and authentication used on the web are usernames and passwords, which are hard to manage and well-known for their security vulnerabilities. To solve these pitfalls, the Internet Computer blockchain pioneered a more advanced and much more secure method of cryptographic authentication, known as Internet Identity, which is more convenient to use, works across all of a user's devices, and helps to protect user privacy.

Internet Identity is a gateway to applications on the Internet Computer. When you use Internet Identity, websites cannot collect and share information about your online activity. This is because Internet Identity helps you create and manage anonymous, independent accounts for every website so that you get the privacy of having many different accounts without the burden of managing them.

Go deeper

Core IC Protocol Chain-key technology Tokenomics & Governance Chain-evolution technology Smart c