∞ INTERNET COMPUTER

Internet Computer
## IN A NUTSHELL

A non-technical introduction

Contents

Genesis of the Internet Computer project

The Internet Computer project can be traced back to 2014, when Dominic Williams, an early pioneer in blockchain technology, was working on ways to make them vastly faster, more efficient, and scalable. He was the first adapting classical distributed computing techniques for the blockchain setting, and that year became the first to describe a "sharding" method for creating infinitely scalable token ledgers.
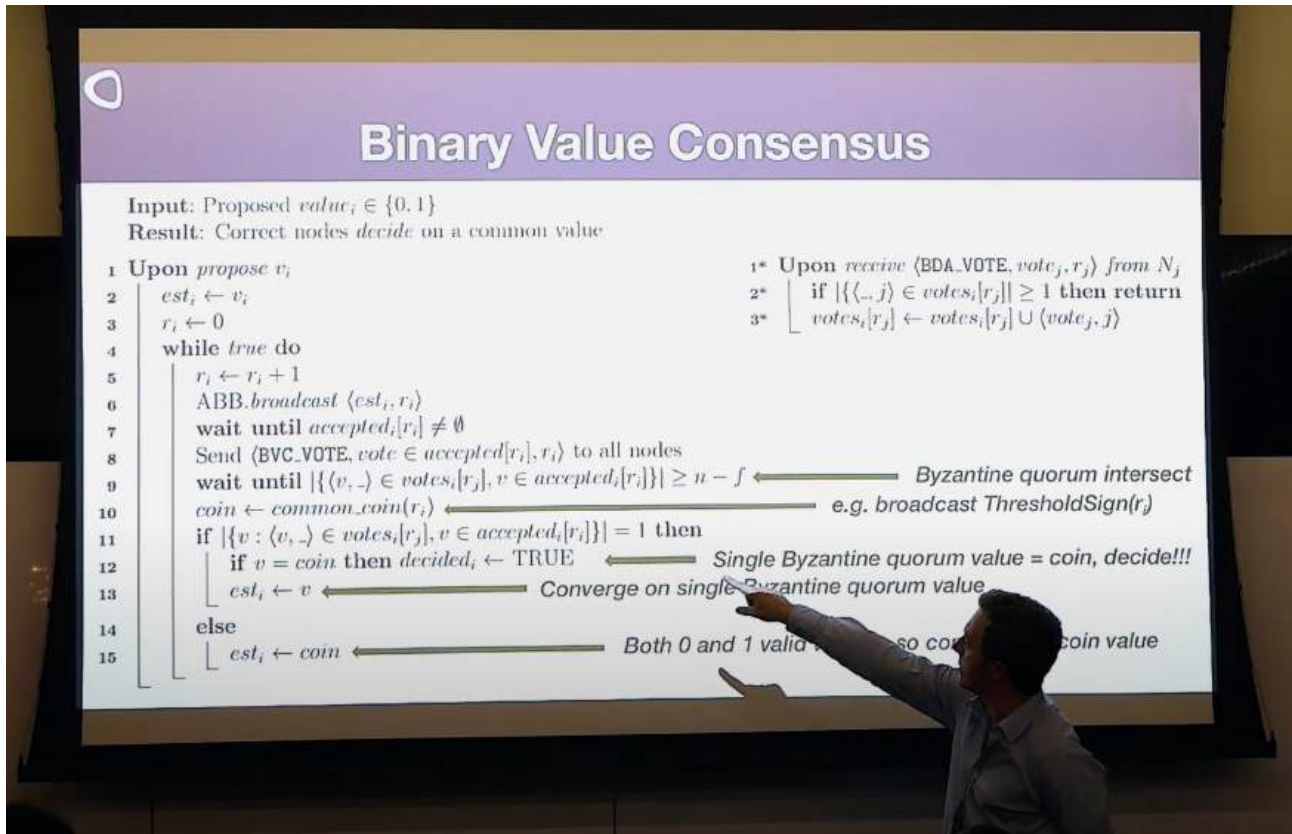
His research work led him to becoming involved with the early Ethereum community, before the Ethereum blockchain network launched. Ethereum was working towards launching a network that would host a new kind of "smart contract" software, which could be used to create DeFi (decentralized financial) services.

Dominic soon realized that, with sufficient R&D effort, the principles underlying smart contract technology could be more broadly applied, and a new kind of network could be created that would host an evolution of smart contract software that was much more general-purpose. This could be used to create things like social networks, enterprise systems, and AI models, which would run entirely from the network.

This new "network-resident" backend software would offer seminal advantages when compared against the backend software that runs on

server machines in the traditional IT stack. For example, this software would be immune to the usual forms of cyberattack, and unstoppable, within the network's "fault bounds." The software would also greatly simplify how web applications and other internet services are built and maintained.

Moreover, web applications and internet services built using this network-resident software would be sovereign, in the sense that they resided on a decentralized network, rather than on proprietary infrastructure provided by a corporation, such as a cloud provider. Such software could also be configured to run "autonomously" under the exclusive control of digital governance systems, potentially paving the way for a new generation of transparent and democratic internet services that run under the direct control of their communities.

The primary objective of the imagined network was to extend the decentralized public internet with "World Computer" functionality. While the initial function of the internet was to make it easy to connect software, independently of location, this new functionality would allow the decentralized networking environment to also host software in the mode of a public cloud computing platform. Where private network platforms were more appropriate for an application, adapted versions of the technology could be used to obtain security, resilience and other benefits.

Developing the technology became a major undertaking involving years of work, and large teams of highly qualified researchers and engineers.

Extending the decentralized internet paradigm

The official birthday of the modern internet is January 1, 1983, when an earlier network called ARPANET upgraded its network protocols. The internet's network protocols are decentralized, and trace their lineage back through work on early packet-switching networks in the 1960s, which aimed to develop communications networks that could withstand a nuclear strike.

The purpose of the internet was to connect software, independently of where the software was running, in a highly resilient and simple way. Today it is responsible for connecting our web browsers to web servers, so that we can consume content on the World Wide Web, and forwards our emails, transmits our instant chat messages, streams our video, and many other things.

The permissionless and decentralized nature of the internet was key to its success. Since no single actor owns or controls the internet, it can play the role of a global public utility that belongs to all humanity.

Anyone can extend the physical infrastructure over which the internet runs, for example in the role of an ISP (Internet Service Provider), which earns a profit by selling access to others. The decentralized nature of the internet means that such parties do not have to navigate gatekeepers or obtain permission to build-out their infrastructure, nor worry that gatekeepers might rescind their own access, which would destroy the value of their investments. Thus, as demand for internet access and bandwidth began to grow exponentially in the 1990s, new entrants quickly entered the internet infrastructure market, and the internet quickly scaled its capacity to meet demand without being throttled by bottlenecks.
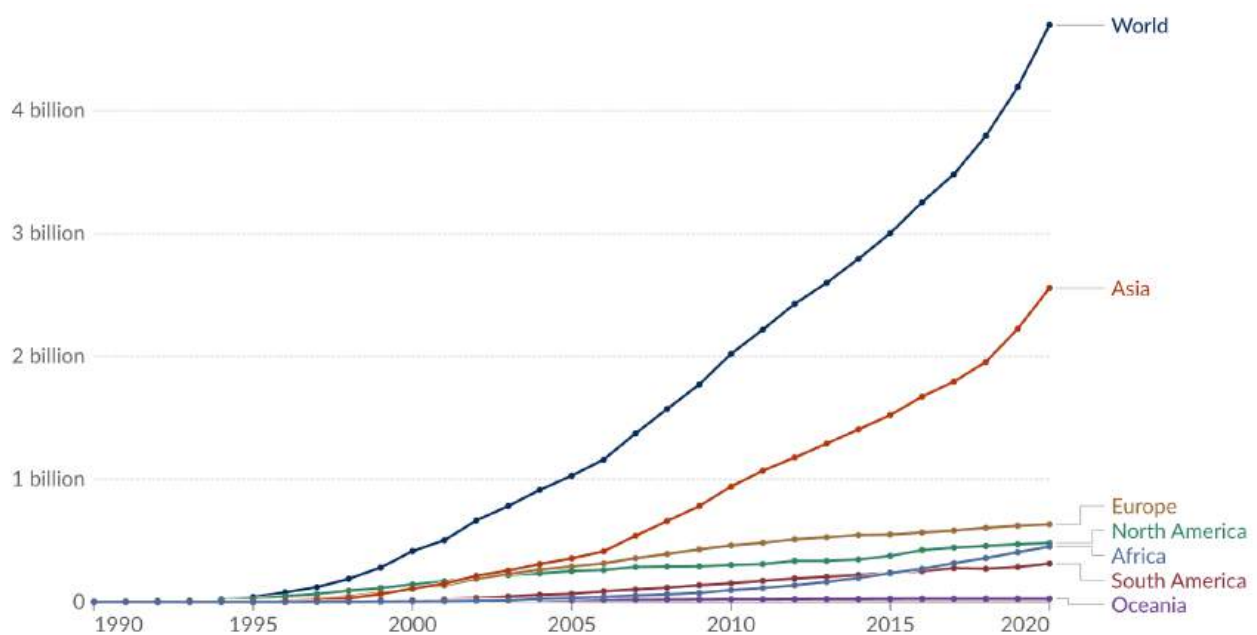
Internet adoption post 1990 (the first website went live August 6, 1991)

These factors led to a massive wave of internet adoption starting in the 1990s, and the internet quickly became the default means to connect and began to host a giant global marketplace for goods, services and information, where anybody could take part.

The world is now exploring other ways that decentralized networks can create useful functionality, and in 2008, Bitcoin pioneered a new kind of "stateful decentralized network" that ran over the internet.

In a stateful decentralized network, the participants in the network jointly maintain shared data that can only be updated in ways defined by the network's protocol. In the case of Bitcoin, the shared data is a ledger of bitcoins that play the role of "digital gold." This shared data can only be modified by submitting properly authenticated "transactions" to the network that move bitcoin between addresses.

Stateful decentralized networks are formed using network protocols with special mathematical properties that prevent them from being subverted and give them incredible resilience. This is why bitcoins cannot be stolen without stealing the relevant authentication material from their owners, and the network was not shutdown during its controversial early years.

Dominic's aim was to build on the stateful decentralized networking approach, to create a public network capable of hosting the world's computation – which network would act to extend the decentralized functionality provided by the internet environment.

This would unlock a powerful new paradigm: today, while the internet connects software, it cannot host backend software and data. Consequently, while the internet connects us to things like social media services and enterprise applications, which form part of our daily lives, the services themselves do not run on the decentralized networking environment. Instead, they run on centralized, proprietary, private computing infrastructure, such as Big Tech's cloud services, where they are constructed using the technology of traditional IT.

While this centralized infrastructure suits many purposes, many services would better run on "World Computer" functionality. DFINITY addressed this need by launching the Internet Computer network May 10, 2021, after several years work developing the necessary technology.

In the years since, the Internet Computer has been used by entrepreneurs and businesses around the world to build and run everything from decentralized social media, through games, sharing economy frameworks, enterprise apps, financial systems, AI models, and more.

A new use case involves advanced AI models being enabled to build custom web applications and internet services completely solo, by leveraging the unique properties of the Internet Computer environment. In this new paradigm, parties spin-up and evolve sovereign network- resident applications simply by chatting with AI, then obtaining full ownership and control over the underlying software code and data involved.

Blockchain functionality on a "World Computer" network

The network-resident backend software that the Internet Computer hosts offer a superset of the properties and functionalities of the smart contract software that traditional blockchains like Ethereum host.

Smart contract software is tamperproof, in the sense it is guaranteed to run its correct logic against its correct data, unstoppable, in the sense that it will always run on request, and can be autonomous, in the sense it can exist independently of human control, either where it cannot be updated, or where it can only be updated by a decentralized digital governance system, which implements the wishes of a community of independent actors. These properties make it possible to use smart contracts on blockchains like Ethereum and Solana to build decentralized financial (DeFi) rails that process tokenized value.

However, the computational capacity of this smart contract software is incredibly limited, and they can only host tiny amounts of data. This is sufficient to implement DeFi schemes, which essentially involves maintaining accounting information and applying straightforward related processing, which nonetheless can often cost several dollars to execute. Such computations are invoked through the submission of transactions, which users create using crypto wallets.

While such traditional blockchains have helped pioneer DeFi, in practice even those that market themselves as being the most "scalable" and "efficient" remain unable to store a simple photograph taken using a phone (for example, when these blockchains host an "NFT," the related image is stored elsewhere, such as on a cloud service).

Widespread confusion around their capabilities has arisen because "web3" projects, such as social media websites and games, market themselves as being built "on" blockchains whose ecosystems they often have financial ties to. However, what is meant by this language, which is now almost universally used, that the web3 services are built on centralized Big Tech cloud infrastructure using traditional IT technology, and that they maintain associated tokens and DeFi logic on the referenced blockchains.

By comparison, the Internet Computer really hosts services like social networks, enterprise applications, and AI, which can be built exclusively from the network-resident software it hosts.

This new form of network-resident software has tremendous speed, efficiency and capacity differences when compared against traditional smart contracts, which are measured in several orders of magnitude, but also supports important new kinds of capability. For example, hosted software pays the network for its own computation, and can process HTTP, such that it can serve interactive web experiences directly to users, who may be unaware that they are seamlessly interacting with software hosted by a decentralized network rather than a server.

This contrasts with smart contracts, which cannot create web experiences, such that users wishing to interact directly must manually invoke computations by using crypto wallets to craft individual transactions that must be configured to carry payment for the expensive smart contract computation that will be performed, in a shot-by-shot basis.

Under the hood, the Internet Computer creates its decentralized platform functionality using a lot of advanced math and computer science. It leverages principles pioneered by blockchain but reimagines the science and engineering on top to create a World Computer.

Special "chain key" functionality supported by the network allows network-resident code to create accounts on traditional blockchains, and submit signed transactions, without maintaining a private key that might be stolen (in fact, the network's nodes directly interoperate with Bitcoin and Ethereum nodes).
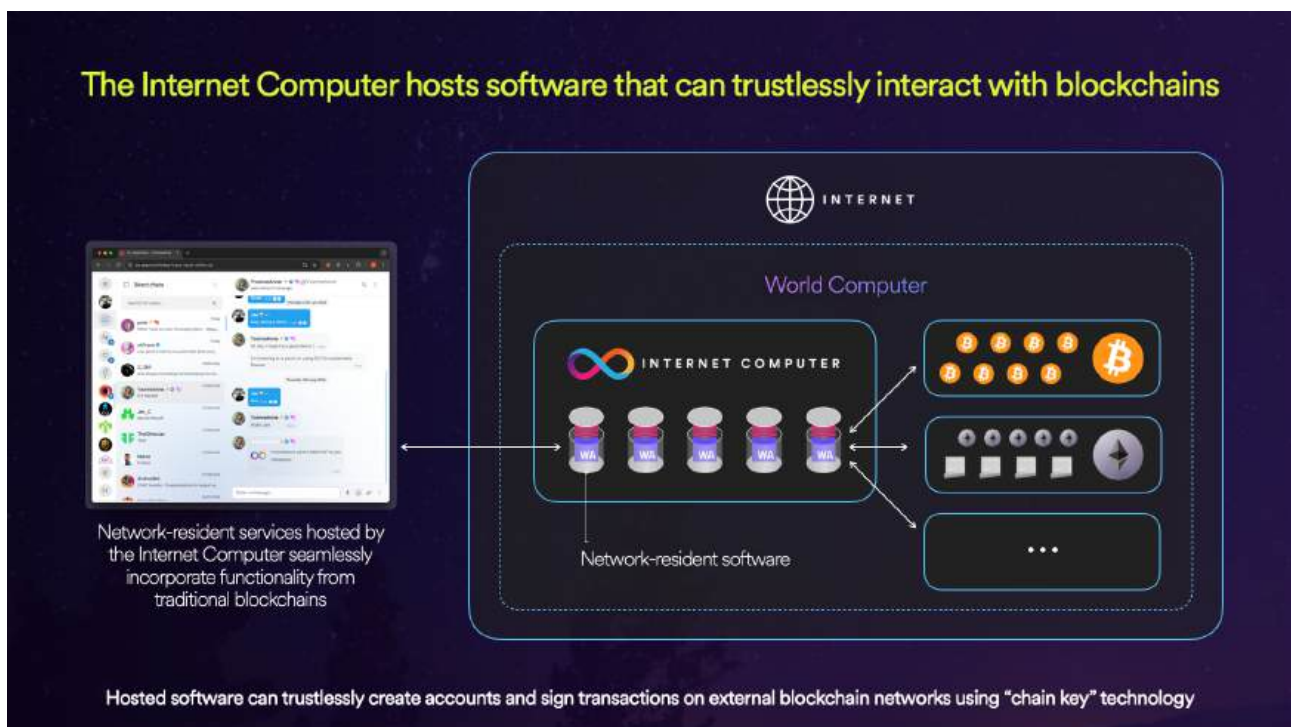
This makes it possible to create trustless digital twins of tokens on traditional blockchains, for example allowing DeFi functionality to be added to networks such as Bitcoin that don't natively support smart contracts, and to fully decentralize web3 services that have been built using centralized infrastructure, so as to make them tamperproof, unstoppable, autonomous and censorship resistant in the mode of blockchain. Through this functionality, traditional blockchains become part of *one* World Computer.

The DFINITY Foundation

In 2016, the DFINITY project was assisted by a venture-backed incubator of decentralized computing projects based in Palo Alto, California.

When it was time to incorporate, it was seen that a neutral not-for-profit organization would be more suitable for building out internet functionality than a startup company, and so in October 2016, the DFINITY Foundation was incorporated in Switzerland.

During 2024, DFINITY calculated that it had directed more than 1,000 person-years of R&D effort towards the development and improvement of the Internet Computer, in an operation that continues at scale today.

The DFINITY Foundation has funded its operations through an endowment of "ICP" tokens, which tokens derive from the economic rails that power the Internet Computer network.

In February 2017, the DFINITY Foundation first sold ICP tokens in its "Seed" public ICO (Initial Coin Offering). During this event, nearly 25% of all ICP tokens were sold to hundreds of anonymous members of the public at 3 cents each.

Although the blockchain industry was much smaller than it is today, the relatively small amount of funds raised enabled DFINITY to begin rapidly scaling its R&D operations. A second public ICO fundraise had been planned, but regulatory changes prevented it taking place.

DFINITY then ran two private fundraising rounds in 2018, called the "Strategic" and "Presale" fundraisers, raising over $110m from more than 100 different high-profile hedge funds, venture capital firms, and high-net worth individuals.

DFINITY leveraged research centers in California and Zurich, Switzerland. The early team combined technical talent from California's early crypto scene with leading engineers, researchers and cryptographers drawn from organizations such as Google.

During 2018, DFINITY scaled R&D operations in Zurich, by drawing famous and well-known cryptographers from IBM Research Europe, which is based outside Zurich, including Jan Camenisch, who became DFINITY's CTO, and numerous engineers and researchers from Google, which maintains its second largest campus outside Mountain View in Zurich, and academics from Zurich ETH. As a result of this expansion, more than half the DFINITY staff now resides in Switzerland.

A distinguishing feature of the project is that it has been able to blend pioneers of the early crypto scene with highly qualified and published technical personnel from high tech and research backgrounds, which has been key to the realization of the "World Computer" vision.

DFINITY has become world-renowned for its advanced research and engineering capabilities.

How "World Computer" functionality was added to the internet

The internet is the original decentralized network, and is created by a network protocol called IP ("Internet Protocol"). A network protocol is a carefully defined language that computing devices separated by network links can speak to each other to create a functionality.

IP combines physical network hardware and links from independent participants to create a highly resilient public network that can continue functioning even if a substantial portion of the individual participants fail by rerouting data flows. Arbitrary networking hardware can be combined, from a phone's cellular transceiver to an ethernet switch run by an ISP, to a WiFi router installed in a home, to an undersea fiber optical cable. This heterogenous hardware is weaved together to create a robust global-scale environment that *connects software*, independently of where it is running.

The Internet Computer is inspired by the internet model but aims to extend the functionality of the public internet environment so that it can also *host software*.
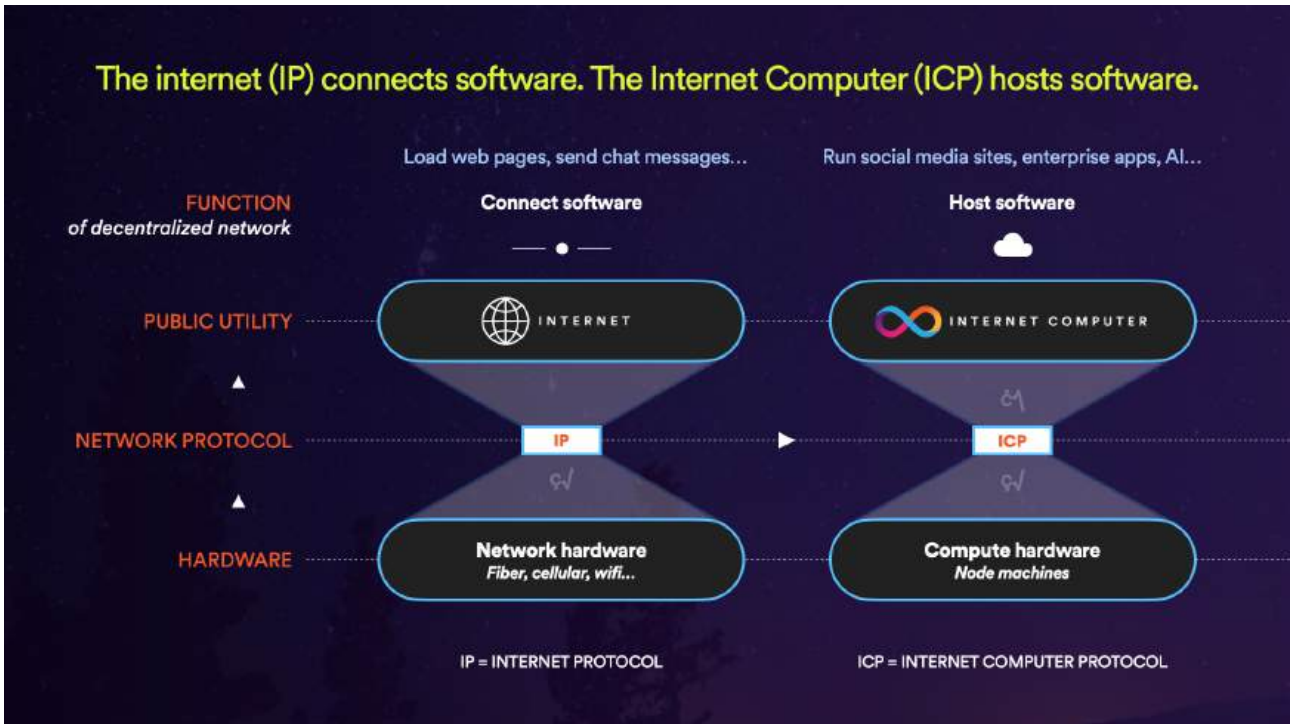

A network protocol called ICP ("Internet Computer Protocol") runs *over the internet* to combine special "node machine" hardware that independent "node providers" (typically companies) operate from data centers around the world. At the time of writing, there are more than 130 node providers that own and operate node hardware at scale. ICP weaves this hardware together to create public "World Computer" functionality.

Node machines are built to specifications that are publicly standardized. This allows them to be combined to replicate computation in data, without less capable machines falling behind when the network is under load. Node machines resemble server computers but have design changes to make them more suited to processing the ICP protocol, and don't include expensive hardware redundancy features since the network's distribution and replication of data and computation makes them unnecessary.

Within the overall Internet Computer network, the ICP protocol organizes these node machines into "subnets." The network creates subnets by combining node machines operated by different node providers, from different data centers, which are in different geographies and jurisdictions. The purpose is to deterministically decentralize the hardware involved, in ways that allow the mathematics of the protocol to deliver the required levels of security and resilience with the minimum of replication, which maximizes efficiency.

Each subnet hosts a subset of the units of network-resident software hosted by the network, and the network increases its capacity by creating new subnets.
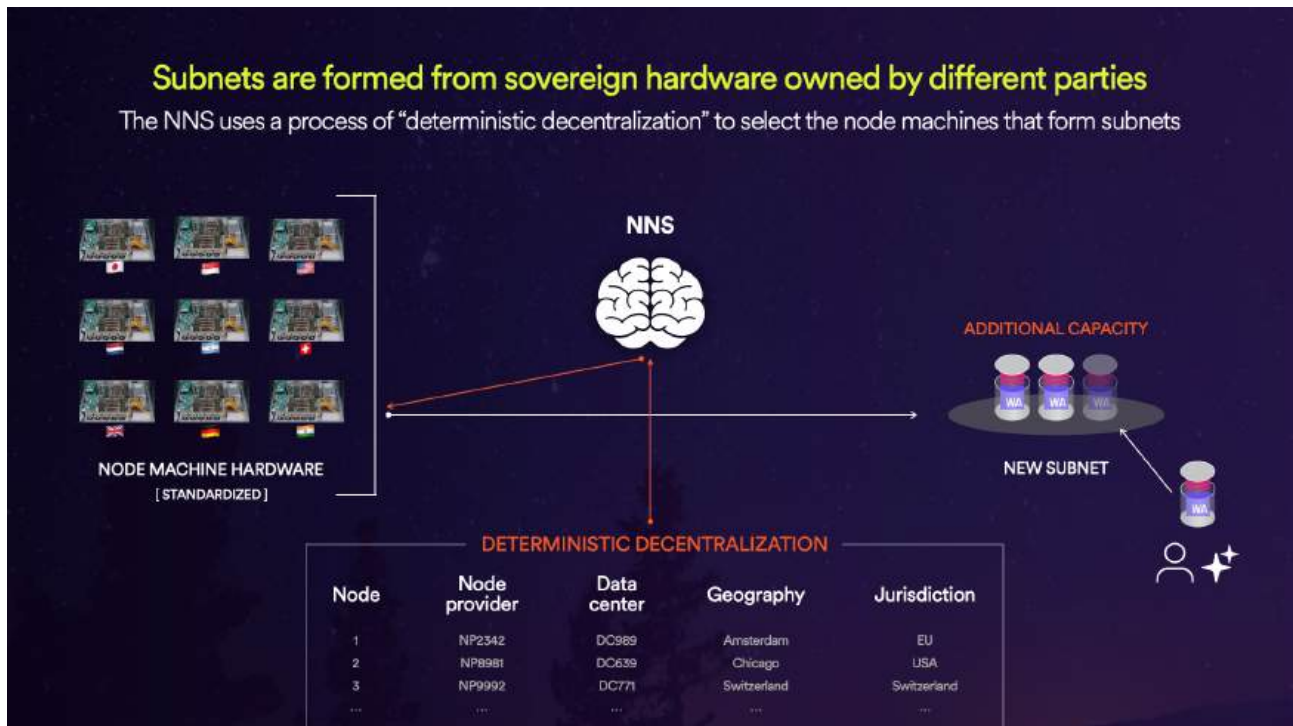
The units of software are an evolution of smart contracts called "canisters," because they bundle logic (in the form of WebAssembly bytecode) and data (in the form of persistent memory pages). Canisters run within a radical new form of "serverless cloud" environment. Within this environment, the subnets are transparent to the canister software, and software can directly call into any other software on the Internet Computer, permissions allowing.

What's important, is that the ICP protocol ensures that if some of the nodes in a subnet become faulty or fail, then the software it hosts continues running without a hitch.

Moreover, the computer science employed by the ICP protocol imbues subnets (and the overall network) with a technical property called "Byzantine Fault Tolerance." This guarantees that even if an adversary of the network, such as an imaginary "Dr. Evil," manages to gain physical control over a portion of the nodes in a subnet, and can arbitrarily corrupt their computations and data, and their interactions with other nodes, then the software units running on the subnet shall continue running completely correctly, and none of their computations or data shall be corrupted or influenced in any way.

The Internet Computer network is highly dynamic and can add and remove subnet nodes without interrupting hosted software. When the network needs more capacity, it forms nodes into new

subnets. When a subnet is overloaded by the software it already hosts, the subnet forks into two subnets, which divide the hosted software between them.

The protocol math and computer science that makes all this possible in practice is complicated. ICP is designed and implemented in a modular way, which allows engineers and researchers to contribute within their domains of experience, without having to understand how all parts would work in detail, while allowing the end-to-end mathematical properties to be verified. This approach to protocol design has allowed ICP to become the most sophisticated network protocol ever engineered.

A new kind of decentralized compute platform has thus emerged, which can host serverless software, computations and data, at scale, which is immune to cyberattack and unstoppable, which can interact with HTTP to create web-based user experiences and is flexible enough to run AI models.

Why hosted software is immune to cyberattack

The backend software and data on traditional IT is highly vulnerable, and a degree of insecurity and risk remains even when the best cybersecurity technologies and practices are used. Even active management by security teams leveraging firewalls, anti-malware, and intrusion monitoring systems is insufficient to guarantee security. At any time, a mistake can result in hackers entering systems, and exfiltrating confidential data, or encrypting systems with ransomware potentially causing permanent system and data loss, and global cybercrime costs are now approaching $10 trillion a year.

The Internet Computer addresses these challenges by providing "World Computer" functionality with security and resilience guarantees that derive from the mathematical design of the ICP protocol. Within the "fault bounds" of the protocol, the Internet Computer guarantees that software it hosts is tamperprooof (in the sense it will always run its correct logic against its correct data) and unstoppable (in the sense it will always run its logic against its data on demand).

The architecture leverages two key principles to make network-resident software immune to traditional forms of cyberattack and unstoppable.

The first key principle involves running each unit of hosted network-resident software within a global-scale distributed "virtual execution environment." This is analogous to how Web browsers protect users from the software embedded inside the web pages they display, by hosting each page's software inside a secure sandbox, so that hackers cannot craft code that

allows them to escape onto users' computing devices and do malicious things.

The Internet Computer places all the network-resident software it hosts, and all related computations and data, into a similar framework – the innovation being that the virtual execution environment hosts an entire serverless cloud computing platform, rather than simply the code embedded within a web page.

Perhaps unsurprisingly, the Internet Computer and web browsers share technology. The Internet Computer runs software on a specialized version of the "WebAssembly" virtual machine. This virtual machine is also used by modern web browsers to safely run software packaged within web pages at native speed, for example for purposes such as decoding video.

An interesting historical note is that WebAssembly was co-invented by one of the earliest DFINITY Foundation team members (Andreas Rossberg). From the beginning, it was intended that WebAssembly would be used to safely run performant backend software, as well as software in web browsers. To create the overall virtual execution environment, the Internet Computer adds many additional components to WebAssembly, for example providing units of software with APIs

that enable them to call into other software, set permissions and other configurations, and process HTTP to serve content and interactive web experiences.

Thus, although the Internet Computer is "permissionless," and there is no barrier to malicious software being freely uploaded, because the cloud computing platform runs within a distributed sandbox, it cannot escape onto the node machines that host the network to cause faults, or subvert the environment to interfere with other network-resident software in unexpected ways.

The second key principle involves hosting the virtual execution environment inside a "stateful decentralized protocol." As discussed earlier, in the case of Bitcoin, the "state" is the ledger of bitcoins, and in the case of the Internet Computer, the "state" is the entire virtual execution environment that hosts the cloud computing environment.

Network protocols often require that participants maintain some data related to their interactions. For example, TCP ("Transmission Control Protocol," which is part of the IP family of protocols) allows two instances of software to maintain a reliable bi-directional stream of data over the internet, which developers access via software libraries to perform tasks such as streaming video and transferring chat messages. The libraries that

implement the protocol are required to maintain contextual data relating to the stream.

A stateful decentralized protocol differs in that there are many participants, and through their interactions with other participants and the contextual information and computations they perform as a result of following the rules of the network protocol, they also jointly maintain a copy of a shared global "state" (which is consistent across all participants).

Part of the inventive step involved is that even if individual participants don't follow the rules of the protocol, and even if they actively try to disrupt other participants, the copy of the state that correctly behaving participants maintain is not corrupted and remains consistent. This means that faulty participants helping host the Bitcoin network cannot conjure up new bitcoins, or steal existing bitcoins, even if they are "Dr. Evil," and similarly, participants in the Internet Computer network cannot interfere with hosted software, or its computations and data. This results from the protocols being mathematically designed and achieving a special property known as "Byzantine Fault Tolerance, which provides security and resilience guarantees."
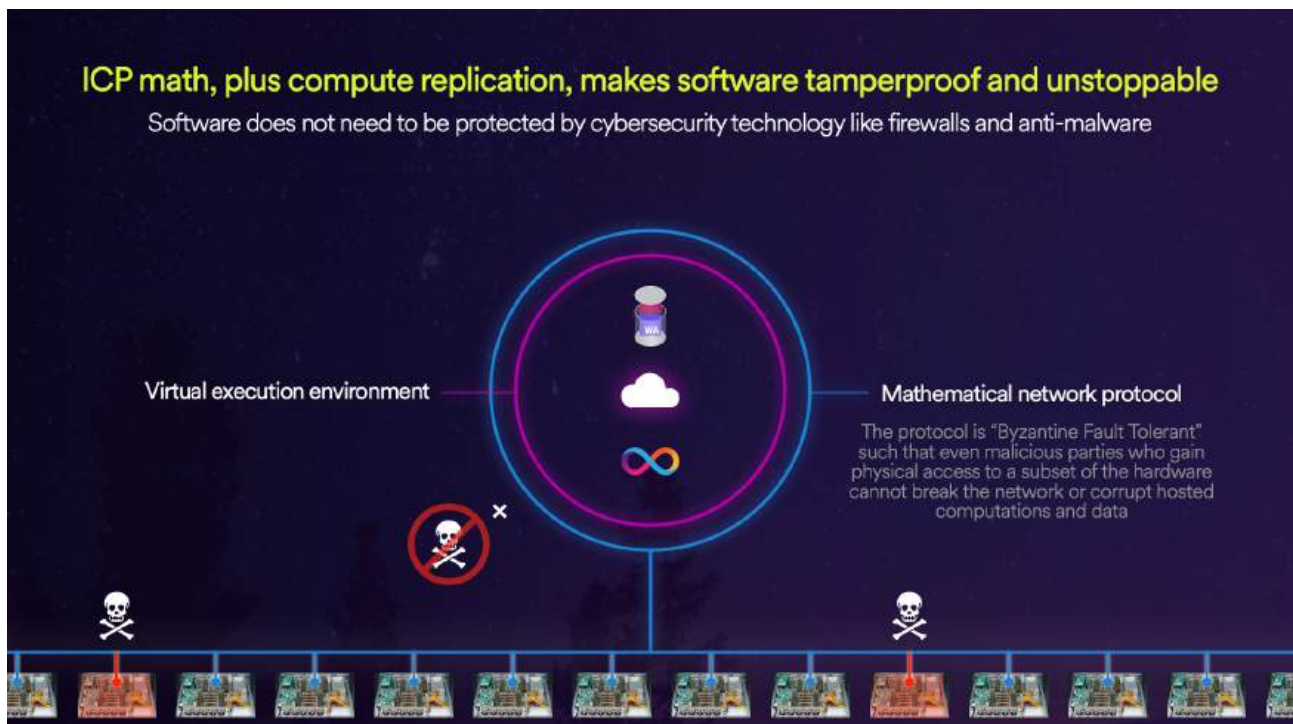
The power of this approach is that hackers cannot make 2+2=5 and bend the rules of math. So long as the proportion of faulty participants does not exceed the mathematically-designed fault bounds of the network, the platform created by the network remains tamperproof and unstoppable, which property is extended to hosted software and its computations and

data, since this is just part of the state, in the same way that a bitcoin is part of the state of the Bitcoin ledger.

An obvious danger with such schemes is that a "Dr. Evil" might add new faulty participants to the network (here, node machines), until such time that the proportion of faulty participants exceeds the fault bounds. However, the network is designed in such a way that the "node providers" that own and operate node machines are identified, and they are combined in ways that prevent this from happening.

Traditional IT infrastructure is very different beast. Software runs directly on computers, rather than inside a virtual execution environment inside a mathematically secure protocol. A hacker can exploit software to escape onto a computer's operating system, or into other software, and then exfiltrate data, or install ransomware.

By contrast, the Internet Computer guarantees that network-resident software and its data and computations cannot be subverted, and can always be invoked.

This has been proven in production. At the time of writing, there are web3 social networks running on the Internet Computer, which have many thousands of users, and which, for the past two years, have enabled users to maintain cryptocurrency tokens inside their accounts so they can conveniently transmit them using means such as instant messaging. These services have run problem free, despite numerous prolific state-funded hacking groups being focused on stealing cryptocurrency from internet services, and despite the services running without security teams or traditional forms of cybersecurity such as firewalls and anti-malware.

The Internet Computer provides a seminal advance in how software logic and data is secured.

The economics of the Internet Computer network

Just like the internet itself, the Internet Computer aims to run in the mode of a global public utility, which nobody owns and controls. To be self-sustaining, decentralized networks must incorporate economic models that support the financing of the physical hardware that hosts them.

The internet creates a self-sustaining economy through "peering relationships." In its most obvious form, parties pay other parties to connect their computing devices and networks to the internet. For example, a retail user might pay a cellular operator to connect their phone, and a business might pay an ISP to connect their office, which in turn might pay an internet

backbone provider to add additional connectivity to their subnets, which may pay to connect to undersea cables.

(Sometimes the nature of the economic benefit provided by peering relationships takes other forms, especially at the core of the internet. For example, some Big Tech organizations run undersea cables and allow telecoms corporations to connect to them for free, receiving economic benefit by guaranteeing and speeding access to their services for billions of users. Meanwhile, multi-party peering relationships often involve payment-in-kind, in the sense that each adds redundancy to their routes, such that there are no obvious payments being made.)

The internet can leverage a peering-based economic model to fund its underlying network infrastructure because its purpose is to connect software. The purpose of the Internet Computer is to host software, and it uses a different kind of economic model to flow value from hosted software to those operating the underlying hardware, which involves tokenization.

A ledger of "ICP" tokens (named after the protocol) is hosted by the Internet Computer network. These are imbued with value by various sources of demand deriving from utility they provide, which allows the network to issue them to the node providers as a reward for node machines they correctly operate.

Node provider businesses have fixed costs, which relate to the cost of capital involved in acquiring their node machine hardware, and then hosting it inside a data center, and administering their infrastructure. Since the price of crypto tokens can be volatile, the network's protocols modulate the number issued to node providers to ensure their financial rewards are stable when measured against fiat currencies, allowing them to operate their businesses in a stable way.

A key source of demand, which helps imbue the ICP tokens with value, relates to how software installed on the network is run. Software on the network must be charged with a digital fuel called "cycles," which it burns as it performs computations and persists data, much like an electric car is charged with electricity, and then burns the electricity as it travels around.

The Internet Computer provides functionality to convert ICP tokens worth 1 XDR (a virtual currency defined by the IMF) into 1 trillion new cycles, thereby also making the cost of compute on the network stable and predictable. On a continual basis, people running software on the network convert ICP tokens they have purchased into cycles, which they use to power their software, causing it to be burned and disappear forever.

Additional sources of demand for ICP tokens derive from their role as a form of cryptocurrency, and their use as form of "stake" that enables participation in network governance.
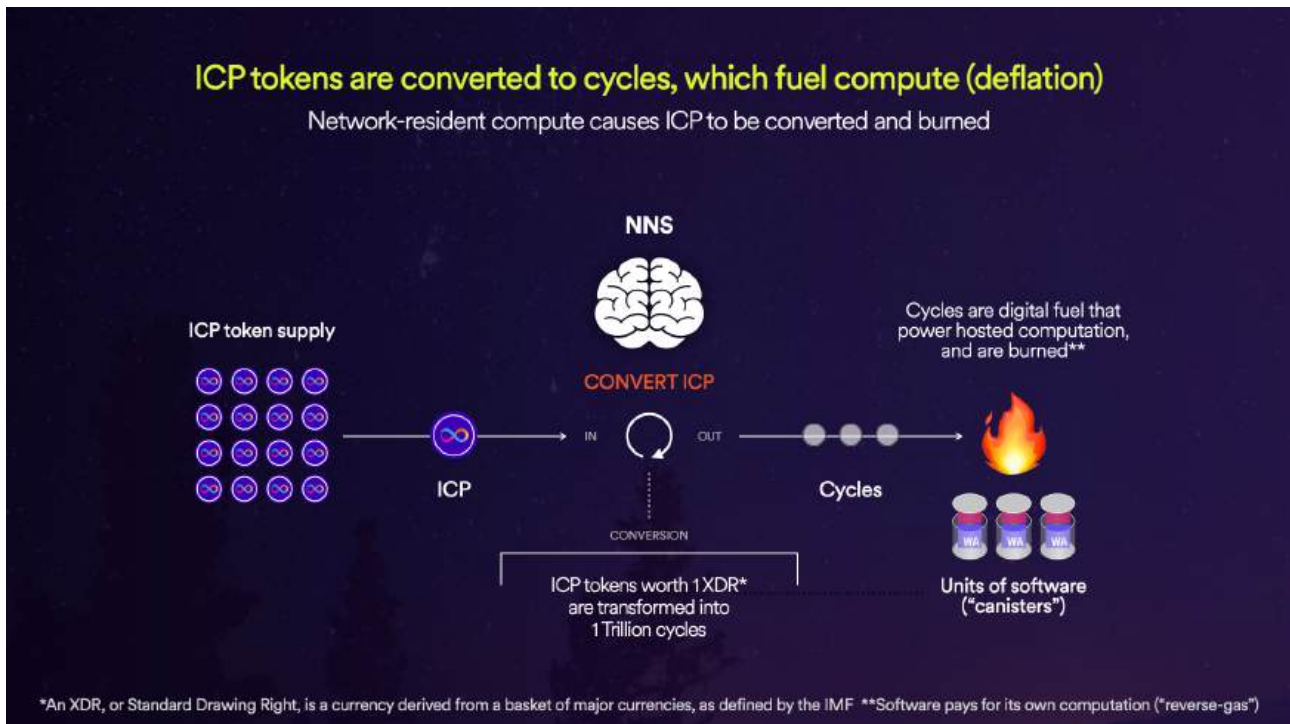
Thus, the four primary purposes of ICP tokens are:

1. Providing the source material for a digital fuel called "cycles," which are needed to power software hosted on the Internet Computer network.

2. Providing the Internet Computer with a means to reward node providers hosting the network.

3. Mediating participation in network governance and providing rails that can reward participants.

4. Application as a programmable store of value and medium of exchange (cryptocurrency).

The open governance of the Internet Computer

It would be extremely difficult to create a global scale decentralized network without some form of governance because some tasks involved in their management require the application of intelligence (human or otherwise).

While the internet is mostly decentralized, it does have some centralized dependencies related to governance, including a not-for-profit organization in America called the "Internet Corporation for Assigned Names and

Numbers" (or "ICAAN"). Among roles, the organization organizes the dispensing of internet addresses used by computers, and friendly domain names.

Any centralized dependencies within decentralized networks are vulnerabilities, which can reduce neutrality, security and censorship resistance, among other issues. For this reason, the Internet Computer runs under the control of a fully automated and permissionless decentralized governance system called the "Network Nervous System" (or "NNS"), which orchestrates the entire network.

The NNS is created by a sophisticated software framework that is hosted on the network itself, which is also directly integrated into the underlying ICP protocols.

Since the node machines that host the network blindly process ICP protocol interactions with other nodes, they also follow the instructions of the NNS automatically. This allows the NNS to orchestrate network governance and management, without centralized actors being involved.

The NNS is permissionless and decentralized, and functions as a kind of DAO ("Decentralized Autonomous Organization"), which allows the Internet Computer to run with full autonomy, while also dynamically adapting and evolving.

Anybody can submit proposals to the NNS, which describe actions the network should take. Each proposal belongs to a standard topic, and has a specific type, which defines exactly what information must be supplied. Proposals trigger actions such granting a cryptographic identity to a new node provider wishing to join the network, tweaking network economics, updating the node machine software that processes the ICP protocol across the network (which is how the network regularly upgrades the version of ICP that it is running), creating additional capacity for software by creating new subnets, splitting overloaded subnets, and myriad other functions, including ingesting external data that helps the network function.

The design of the NNS enables it to rapidly securely decide on proposals. Because the proposals the NNS adopts are automatically executed by the network, the network can rapidly adapt to changing circumstances, such as increases in demand, and evolves under its own steam by

quickly integrating improvements. At the time of writing, the NNS often adopts tens of proposals a day, reflecting the degree to which the network runs in a highly dynamic and self-directed manner.

Anyone can submit proposals because the NNS is permissionless, and anyone can participate in voting on proposals, reflecting the open and democratic aims of the network.
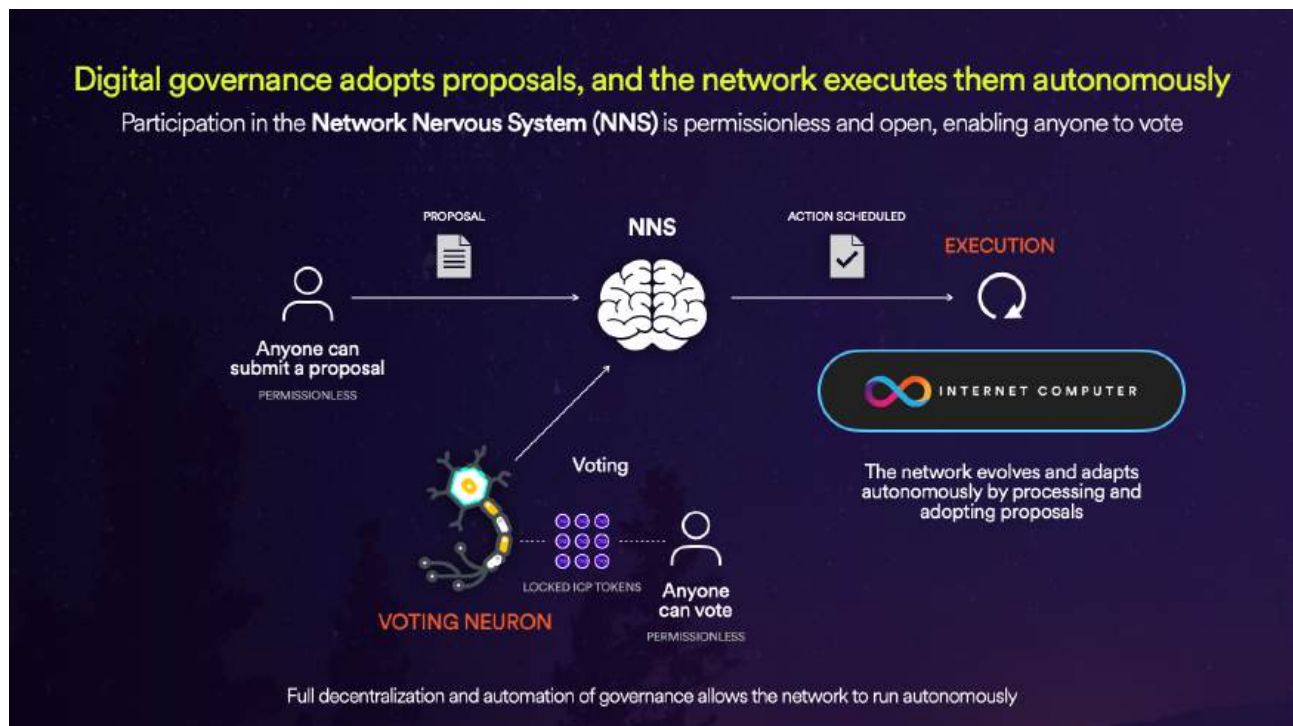

A challenge with any such governance system, is finding a way to ensure that participants vote in honest and responsible ways, and ways to stop malicious parties simply joining in large numbers to out-vote honest parties, which could result in the network performing harmful actions, such as adopting an upgrade that deletes all data.

This problem is solved by voting "neurons." To vote on NNS proposals, it is necessary to create anonymous neurons within the NNS, and then imbue them with voting power by locking ICP tokens inside. Consequently, everyone that votes on proposals has placed ICP tokens they own at risk by locking them.

If harmful proposals are adopted, this will cause the value of the ICP tokens locked inside neurons to fall, and vice versa, creating an incentive for neuron holders to vote for actions that will create long-term advantage, because the ICP tokens they have locked inside their neurons cannot be immediately retrieved.

The NNS provides incentives that encourage people to participate in governance, as well as node provision, which has resulted in massive amounts of capital being locked inside neurons by honest parties. This makes it infeasibly expensive for malicious parties to acquire sufficient ICP tokens to exert a harmful influence by voting on proposals, even though they may have financial incentives to do so (for example, because they have placed short

bets on the ICP token and wish to harm the network to make them profitable).
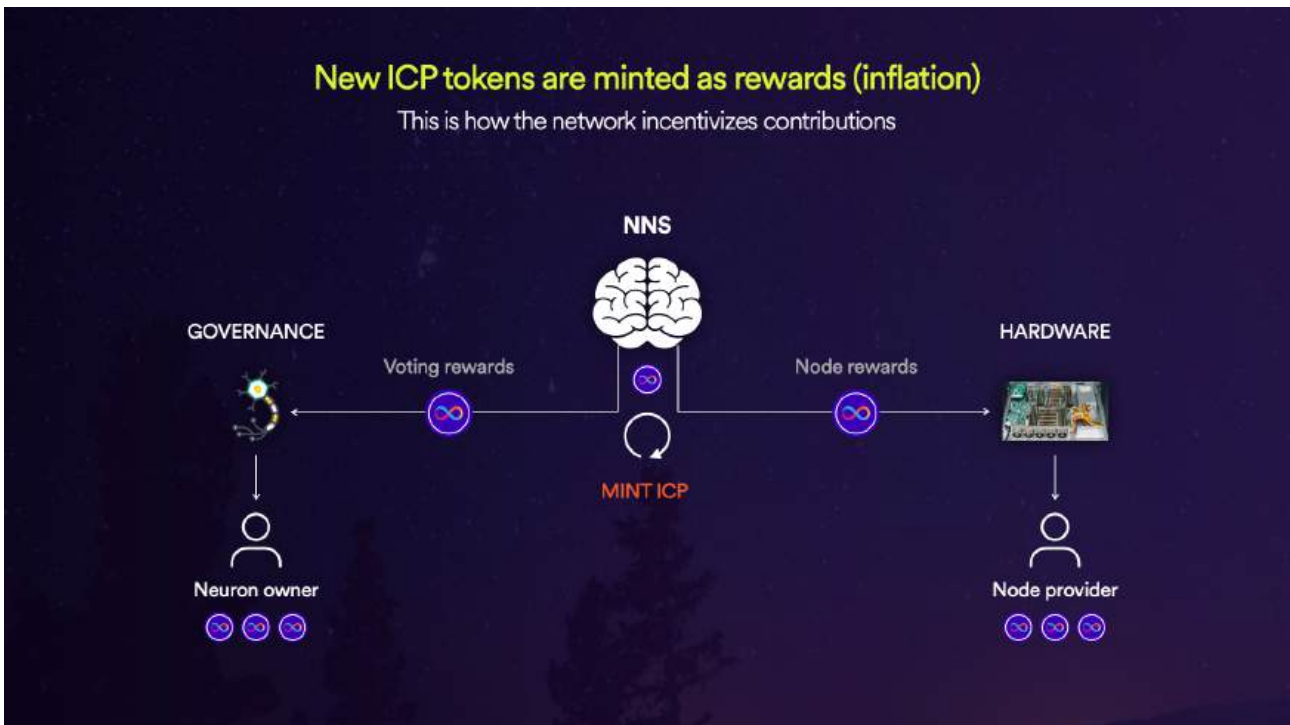
Understanding voting neurons

To participate in Internet Computer governance, it is necessary to create "voting neurons" inside the NNS. Neurons are standalone objects, which are not designed to be transferred or sold. Neurons create powerful incentives that encourage participation in NNS governance, while ensuring the overall governance system functions as intended.

NNS participants can configure their neurons to vote completely automatically by following the voting of other neurons they have selected, which typically belong to leaders and experts within the Internet Computer ecosystem.

Neuron owners can also configure voting to follow the majority vote of a quorum of other neurons, and can reselect which neurons are followed, or return to manual voting, at any time. Following can be configured differently for each of the various governance topics the NNS supports, allowing voting power to be precisely delegated to those with appropriate expertise. The NNS thus functions as an advanced "automated liquid democracy" that has no direct parallel in the world today.

Many participants in NNS governance create their neurons to earn rewards in ways described herein, encouraging the contribution of capital value. Since these neuron holders typically delegate day-to-day voting power to experts

for the purposes of convenience, this provides tremendous security and stability, while leaving ultimate power among the totality of neuron holders.

The influence any individual neuron can exert, and the size of the rewards it can eventually earn, derives proportionately from the relative size of its voting power.

The voting power of each neuron depends on its configuration when it votes, specifically: 1) the number of ICP tokens locked inside, 2) its current "dissolve delay," which determines how long it would take to release the ICP tokens inside were it placed into "dissolving mode," which causes the dissolve delay to fall with time, and 3) it's "age," which is a virtual attribute calculated as the lessor of the time elapsed since the neuron was created, and the time elapsed since it was in dissolving mode.

The voting power of a neuron is calculated using the formula:

Voting power = locked ICP tokens x dissolve delay bonus x age bonus

(Limits include that the maximum dissolve delay that can be configured is 8 years, and that neurons can only vote when the dissolve delay is 6 months or greater, and the maximum *calculated* age is 4 years. The "dissolve delay bonus" starts at 1.0 when the dissolve delay is 6 months, and increases to 2.0 at 8 years, and the "age bonus" starts at 1.0 when the age is zero days, and increases to 1.25 at 4 years.)

For every vote a neuron submits on a proposal, the NNS increases its "maturity" attribute as a reward. When neurons gain maturity, they can be used to create new ICP tokens later.

Understanding neuron maturity

Each neuron has an attribute called "maturity," which is analogous to the weight of a physical object. Since maturity is an attribute, it cannot be separated from a neuron (i.e. it is not a token that can be separately transferred and held). The NNS increases the maturity of neurons as a reward for voting on proposals.

Every 24-hour period, the NNS has a total number of maturity points that it wishes to distribute among voting neurons, which is defined by the Internet Computer's protocols. To make the distribution fair, the NNS divides the maturity points among neurons that voted in proportion to the total number amount of voting power they deployed.

Therefore, neuron owners can maximize their maturity gains, by making sure their neurons vote on every proposal, and by maximizing their voting power

by locking additional ICP tokens inside, configuring their dissolve delays to be longer, and waiting for their calculated age to reach the 4- year maximum.

Once neurons have gained maturity, they can be used to produce new ICP tokens for their owners.

In principle, the exact process involved depends on the current design of the NNS, which can evolve as the network is updated to improve its tokenomics. But in the current design, a neuron with maturity can be used to "spawn" a new neuron that has newly minted ICP tokens inside.

When a new neuron is spawned, the number of new ICP tokens inside is approximately equal to the maturity points "consumed" on the source neuron, and its dissolve delay is set to zero, which allows the new ICP tokens inside to be immediately withdrawn if that is the wish.

The spawning process takes a week from start to end, which means that the owner of a neuron cannot predict what the price of the new ICP tokens will be on markets where they are traded once they can be retrieved. Furthermore, the number of new ICP tokens that is produced depens on how the price is trending during the spawning process.

If the price rises while the spawning process is taking place, a slightly larger number of new ICP tokens will be produced, and vice versa, with a variance of +/- 5%, which encourages neuron owners to generate income in the form of ICP tokens only when the markets upon which they are traded are healthy.
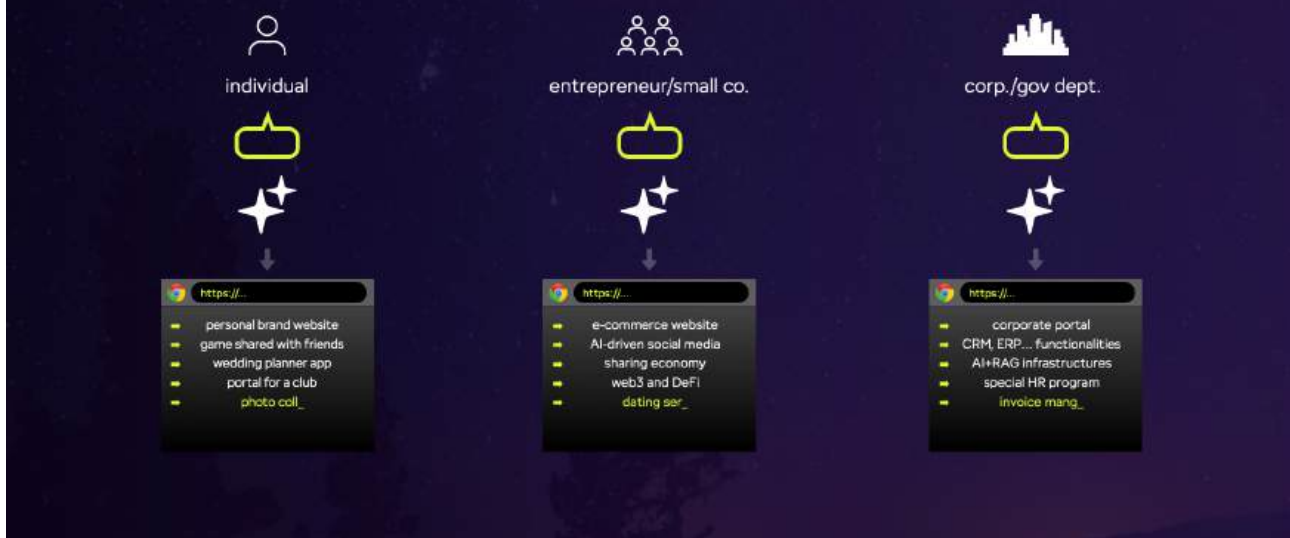
How ICP enables AI to build custom applications and services solo

A key emerging purpose of the Internet Computer, and ICP stacks generally, is to enable advanced AI models to spin up and evolve custom web applications and internet services in response to simple instructions given by a chat interface.

The applicability is broad. For example, an individual might create a personal branding website, wedding planner, or game to share with friends, while a startup entrepreneur might create a web3 sharing economy service, or e-commerce website, and a larger company might create a corporate portal with CRM or ERP functionalities, or even AI and RAG infrastructures. The custom applications and services thus created are completely sovereign, in the sense their creator owns the underlying software code and data contained and does not risk the customer lock-in that might occur had they

Users chat with AI to create running websites, applications and internet services
The *self-writing internet* – non-technical people can build out the internet ecosystem, democratizing tech

individual

entrepreneur/small co.

corp./gov dept.

https://...
- personal brand website
- game shared with friends
- wedding planner app
- portal for a club
- photo coll_

https://...
- e-commerce website
- AI-driven social media
- sharing economy
- web3 and DeFi
- dating ser_

https://...
- corporate portal
- CRM, ERP... functionalities
- AI+RAG infrastructures
- special HR program
- invoice mang_

built on proprietary infrastructure such as cloud services and SaaS platforms.

When AI writes serverless Internet Computer software to create and update applications according to instructions provided by chat, this can be "compiled" and uploaded to the network single shot. This contrasts with traditional IT, where updates can involve software and configurations being installed across a plethora of platform components such as databases and web servers and can involve tasks such as cloud orchestration, and configurations of things like backup and security systems, which proceed far slower than chat speed, and involve problem ladders that require iterative resolution.

While AI is increasingly powerful, it will remain the case that it can hallucinate and make mistakes for some time, and this presents another challenge when AI must maintain services on traditional IT. On the Internet Computer, software is automatically tamperproof, just like the network, which means that a mistake by AI does not open the door to traditional cyberattacks. By contrast, a mistake on traditional IT could have catastrophic consequences.

Similar challenges are involved having AI upgrade services built on traditional IT. Most obviously, traditional IT has been designed to support infrequent upgrades, which can be complex and involve synchronization changes across multiple platform components, rather than to enable upgrades every few minutes in response to chat inputs. Furthermore, software upgrades usually involve data migrations, and mistakes can result in data loss.

24

Users can chat to upgrade *running* web apps in real time

Changes are described, and then the URL is refreshed 1-2 minutes later
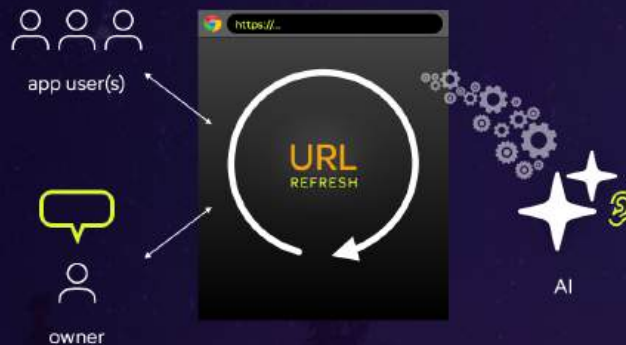
The Internet Computer supports a paradigm called "orthogonal persistence," which essentially somewhat collapses the difference between logic and data – software engineers define logic, and the data it references persists automatically, without the need to copy it into a database or file (this is because software runs inside persistent memory pages). A special domain-specific language called Motoko has been created to leverage powerful features of the Internet Computer environment, and tailor orthogonal persistence in ways that better support AI building solo.
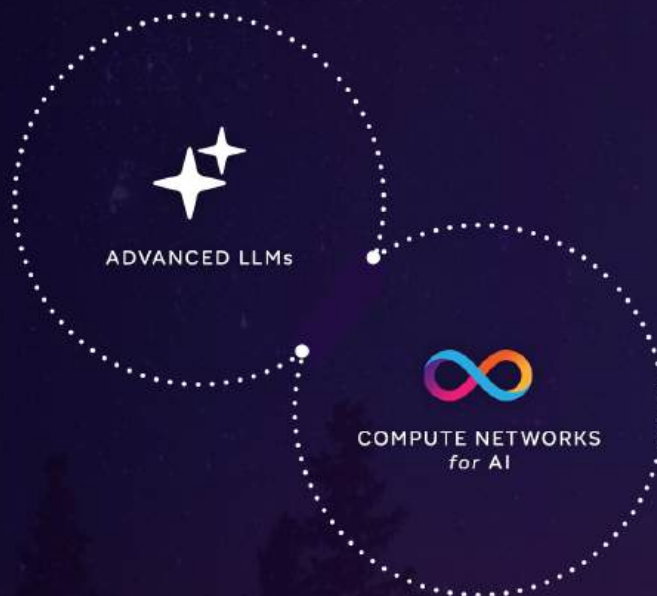
These include making data migrations associated with upgrades extremely efficient, allowing AI to effectuate upgrades to applications and at services at "chat speed." In addition, language features ensure that as software is updated, necessary migrations of memory-resident data are described as part of the software update's code and will fail to build if it includes a mistake that would result in data loss.

—-

Slides sourced from:

deck.internetcomputer.org

deck.icp.ai