SuperSkyNet

Whitepaper

University of St.Andrews
CEO, Dr.Hannover
CTO,Dr.York

August 6 2018
Version 1.0

Preface

SuperSkyNet is an intelligent collaboration system based on IOT, fog computing, block chaining and AI technology. The Internet has already become a part of our lives, as an extension of Internet technology, the Internet of Things is also rising rapidly. Now, there are more and more online devices around us, such as computers, mobile phones, routers, cameras, etc. are in a 24-hour online state. However, these online devices are still not fully interconnected and work together, the global online equipment resources are greatly wasted. So we put forward an intelligent cooperative system based on Internet of Things, fog computing, block chain, AI - SuperSkyNet System.

In retrospect, it is easy to find that science and technology are the great creators of a new world, and the arrival of a block-chain society will truly bring about a globalized distributed network of independent, win-win, sharing of the social operation mechanism.

The interaction of human activities and information, which is constantly forming a huge chain society, will really bring the global distributed network independent, win-win, sharing of the social operating machine data assets. The social and trading data we generate every day can be entirely owned by the producer. Following the spirit of Internet sharing, equality and transparency, this data source is a "global credit resource". Social Computing, Affective Computing. In a world of code bytes, we're like little green geeks traveling through time and space, trailing behind each other a long, shiny tail of digital information.

By 2020, the world's total data storage will reach 40ZB, 1ZB is equal to a trillion GB bytes, human life has become a pile of numbers, Professor C.R. Law tells us: all science is abstract mathematics, all judgment is based on statistics. The authenticity and wide application of data in block chain intelligent contract will be an important process change in Anthropocene.

The goal of the SuperSkynet Block Chain underlying protocol is to involve us in an open source mechanism that connects all real data source records and verifies data sharing, making all intelligent contracts that compute real data simpler and more reliable, and making each perSSN a smart contract computing library and data sharing in business activities And, by jointly validating the data source mechanism, we get positive results reward.

Contents

1 In troduction

1.1 Vision

The concept of SuperSkynet, often referred to as the fictional conscious super-intelligence system in the movie "Terminator", is increasingly associated with the recent explosive growth of artificial intelligence and robotics. With the rise of machine learning, computers have been able to achieve human-level performance on highly complex sensing tasks. 1 Advances in deep learning have enabled artificial neural networks such as AlphaZero to easily defeat World-Champion Go players. Processor optimizations such as Google 180 Teraflop TPU can further accelerate neural network training.

Now, people are beginning to realize that artificial intelligence can turn our world into a world where different intelligent entities can interact without human control. However, concerns such as Google's use of artificial intelligence to automate the US UAV defense system4 have made the project impossible to update, and other companies such as Amazon are about to become the only cloud providers for the Pentagon 6. The fear of SuperSkynet has become more reaSSNable as the world faces many challenges in creating variants that benefit humanity.

However, we believe that distributed ledger technology, such as blockchain, can bring a beneficial SuperSkynet, or what we call the smart machine economy. Using blockchains, you can distribute the collective knowledge of all devices to ensure that no central system can affect all other devices. Manage autonomous devices with systems that can motivate positive behavior while preventing future hSNMful behavior. There is no centralized control on the network, allowing machines to interact directly with each other without human operators or intermediaries. As a result, the blockchain will enable many new decentralized and secure applications that can be executed at the edge of the smart.

In the projected $7.1 trillion Internet market, the $10 trillion blockchain market

Together with the $15.7 trillion AI market9, it constitutes a smart machine economy, and devices such as robotics doctors will be able to diagnose patients and treat patients independently. Self-driving cars will be able to communicate safely with nearby cars to minimize collisions. Smart devices will have the intelligence to represent individual actions to improve the quality of life. With the advancement of artificial intelligence, blockchain technology and hardware, the smart machine economy can be created today.

SuperSkynet is an end-to-end protocol designed to meet the requirements of smart machine economy through its two components: SuperSkynet Network (SSN), an extensible machine learning IoT blockchain platform and SuperSkynet-Core, a license-free The neural processing block chain core.

SuperSkynet Core hardware will lay the foundation for encryption acceleration, neural network processing and system-on-chip development in IoT devices, while SuperSkynet Network will serve as a distributed application, providing devices with the capacity to self-organize, learn and communicate information among each other.

2 SuperSkynet Overview

By creating a neural processing core optimized for the blockchain and its native blockchain network, OpenSingularity addresses the main issues of the Internet of Things, enabling these devices to connect and be intelligent. This section will detail why blockchain can be used for the Internet of Things, SuperSkynet design principles to achieve intelligent machine economy and deployment strategies.

2.1 Architectural Beneftts of Blockchain for IoT

The blockchain is a public, immutable, distributed ledger technology that can be used for transacting with data in a distributed and decentralized manner.

Decentralization As quoted by Vitalik Buterin, “blockchains are politically decentralized (no one controls them) and are architecturally decentralized (without central infrastructure failure points), but they are logically centralized (There is a mutual recognition state in which the system behaves like a single way. In this way, the blockchain provides a decentralized, unreliable way of interconnecting devices and exchanging value. Decentralization deprives large companies of Trust, power and responsibility, and transfer them back to the open community of the supervisor. As a result, the

blockchain can reduce transaction costs and achieve instant transactions by depriving middlemen (such as Paypal) and additional administrative fees. Decentralization also helps resolve Privacy and data issues.

Immutability The data published on the blockchain is immutable, providing transparency and audibility to all devices that conduct transactions over the network. In many scenarios, immutability is useful because it prevents someone from tampering with data and allows everyone to query the chain to access applications such as authentication, timestamps, audit trails, and identity management.

Programmability In the form of smart contracts, programmability on the blockchain enables device au-tonomy to make untrusted exchanges between devices that can be validated through code and other nodes. Programmability can be extended to often static iot devices and support for various changes and interactions between them.

Security The network running on the blockchain has fault tolerance and can withstand node failures. Using the Byzantine fault-tolerant model, components are allowed to fail in the system if their local state is damaged, disconnected, or the output is malicious. In the real world, fault-tolerant systems work well, and the nodes in the system can run in unpredictable ways. As a result, Byzantine fault-tolerant capabilities can be used to implement many desired cybersecurity aspects, such as defending against MITM attacks and DDoS attacks.

2.2 Application Beneftts of Blockchain for IoT

Blockchains bring many application benefits which will be discussed later in theSuperSkynet Network. Just a few applications that a blockchain and its programmability would bring in IoT would be:

1. Distributed Computing - Machines can distribute workloads and share resources such as computation, memory and storage on edge, while being rewarded for the amount that they delegate.
2. Federated learning - Machines can train off private data without ever sending it, leaving training data distributed while improving models' accuracy.
3. Cryptocurrency - A instant and near-feeless digital currency can be used as a way to pay for data and algorithms while providing incentives for others to share it.
4. Secure Interactions - Devices can develop a reputation based on previous transactions and start to self-organize and use peer-to-peer discovery clients to interact with non-malicious nodes.
5. Data Sharing - Data can be securely sent off the chain and be hashed on the blockchain.
6. Imitation learning - Machines can teach one another the correct policies during training.
7. Smart Contracts - Developers can code their own contract in which devices are forced to obey.

For example, applications such as distributed computing will address problems with limited processing power on the edge; federated learning will address some problems with untapped data and allow devices to be compliant with data consent and security laws; digital currencies can be used to exchange value and data, encouraging nodes to participate in the network ecosystem. More applications will be covered later.

2.3 Problems with Blockchain Networks for IoT

Despite the many benefits of blockchain, the current network has high computing overhead and low finality. The network architecture does not handle the billions of interactions that are performed every day by iot devices, nor does it support real-world adoption. Older network architectures such as bitcoin or Ethereum are based on principles such as work proof consensus and the "One blockchain, Many Applications" design. The blockchain developed from these old principles has low transaction rates (7-20 transactions per second) and high transaction costs (. (70 cents), try to accommodate many applications in a chain and let nodes perform expensive, useless

computing tasks. Nor can these blockchains interact with each other because they focus on their own applications rather than working together. Even newerDAG solution requires a heavyweight operation, in which sending transactions forces small devices to perform work proofs. As a result, traditional blockchain and even new versions are not suitable for iot devices. For example, smaller iot devices, such as sensors and wearable:

1. Proof of Work Mining - Smaller devices cannot be turned into miners as they face computation and power restraints.
2. Storing Data - Training data and chain data cannot be stored on devices as they face memory and storage restraints.
3. Connectivity - Devices in rural areas might face latency issues and will not be able to have a steady connection.
4. Running full nodes - Devices cannot verify full blockchains as downloading a whole chain might require upwards of 50 gigabytes of storage.
5. Ternary Operations - No CPUs can work with DAGs or Blockchains with ternary operators.
6. Cold Storage - With IoT devices getting hacked from things like BotNet, devices cannot safely store or utilize cryptocurrency.

As a result, some deep learning distributed applications and blockchain operations might not be suitable for the Internet of Things devices.

2.4 Blockchain Adoption Problem

All cryptocurrencies face adoption issues in addition to architectural issues, and the space is highly speculative. Bitcoin and Ethereum are currently valued at $131 billion and $60 billion, respectively (as of June 2018), because while bitcoin has basic technology, they are the most used network in the field. However, no cryptocurrency can be widely used due to its design and underlying infrastructure. For cryptocurrencies and blockchain technology to start to be adopted, they must:

1. High efficiency - transaction costs should be minimal, with low validation time, and energy efficiency.

2. Legacy compatibility - blockchain or blockchain needs to be compatible with current systems, such as current systems CPU and hardware.

3. Private and secure - blockchain should be flexible (public or private) to meet related iot tasks.

4. Simplicity - blockchain and its respective cryptocurrencies should use simplicity and seamless.Converting
5. passwords to passwords in exchange is a complex task.

6. Security - because cryptocurrency exchanges and hot wallets have been hacked, the cryptocurrency people use is unrecoverable.

RISC-V Set of open-source instruction set architectures and designs for processors

SuperSkynet Core SuperSkynet Core is the name of all variants of the blockchain chip.

SuperSkynet Network SuperSkynet Network is the collective network of all the blockchains, protocols, and smart contracts.

SuperSkynet Network, Fabric SSN Fabric is SuperSkynet Network's root multi-chain blockchain that allows for the creation of blockchains and exchange of tokens between networks.

SuperSkynet Network, Idem SSN Idem is SuperSkynet Network's decentralized identity Network.

SuperSkynet Network, Nova SSN Nova is SuperSkynet Network's distributed application platform.

SuperSkynet Network, Singularity SSN Singularity is SuperSkynet Network's virtual application layer that interfaces with the SuperSkynet Cores.

SuperSkynet SuperSkynet is the system that will enable a new era of machine intelligence by combining the components of SSN and SuperSkynet Core

3 SuperSkynet Core

In the previous section, we discussed an overview of how the SuperSkynet system provides an end-to-end devel- opment platform for IoT applications. In this section we will describe a modular set of hardware IP blocks tailored for optimally running SSN on embedded "edge" IoT devices—that is small devices that serve as sensor or actuators, sit at the edge of the network and are mostly characterized by their low cost and low power budget. In particular, through a combination of cryptographic helpers running a high-security lite blockchain client, an AI accelerator for perceptual tasks and an embedded CPU, SuperSkynet Core will become the ideal platform for IoT OEMs to develop and deploy their applications and devices. SuperSkynet Core will be distributed via a license-free arrangement to System-on-chip (SoC) manufacturers for them to integrate it into their offerings, reducing cost and accelerating adoption

The SuperSkynet core consists of three main components: An SNM or RISC-V based CPU to host a Linux kernel an interface with peripheral devices; A secure Crypto-engine for storing private keys, singing messages and performing any other cryptographic computations required to operate any blockchain efficiently and in particular the SSN blockchain; A Neural Processing Unit (or NPU) to accelerate the linear algebra operations required by modern neural networks such as DNN, CNN and RNNs.

3.1 CPU

For the SSN blockchain to run a modern Linux operating system such as Ubuntu, opensingu-larity will include a set of modular processors in the SuperSkynet kernel. SNM processor's RISC architecture implements simple design, fast clock speed, small mold size and efficient memory usage, providing reliable IP, expert design support and leading software tools for new SoCs or IP blocks. SNM provides an ideal product line for the needs of iot devices, where modern versions feature a full-system approach to secure - trust zones. First, we will integrate the m-series processors of SNM cortex-low-power embedded applications with the 64-bit high-performance processors of SNM cortex-high-end applications. This arrangement is based on feedback from our development partners that iot devices require low power consumption and small footprint. The SNM ecosystem provides AMBA(advanced microcontroller bus architecture) to connect multiple peripherals (IO, coprocessor, and memory controllers) needed to build modern processing units; Multiple vendors provide these trusted peripherals for a variety of process nodes. Finally, SNM's broad penetration creates a vibrant and tested community that supports the entire software stack of bootloader, kernel, driver, distros, libraries, applications, and software development tools such as compilers, analyzers, and debuggers. As the SSN blockchain comes online and the blockchain prototype application begins development, we will describe it

3.1.1 SNM Architecture

SNM is a simplified instruction set computer (RISC). As a RISC, SNM targets a fixed length, simple and powerful instructions executed at high clock speeds over a period of time. SNM, as a RISC architecture, is based on many principles for simple design and fast clock speed. The pipeline is designed to be decoded at one stage, without the need for microcode. A large number of general purpose registers are defined for quick execution of instructions. SNM USES a load/storage architecture in which data processing instructions are used only for registers and the load/storage scheme is used to transfer data from memory. However, there are some differences from pure RISC. SNM USES variable loops for some instructions, such as multi-register loading/storage, to achieve faster and higher code density. Inline tube shifter improves performance and code density, but results in more complex instructions. SNM added the thumb 16-bit instruction set, which resulted in an increase in code density of about 30 percent. Conditional execution is added to improve performance and code density by reducing branches. Some enhancement instructions were added for the DSP operation.

3.2 SuperSkynet Network Cryptography Engine

To handle the computational load associated with blockchain, encryption features, and consensus algorithms, each supergrid core contains an optimized encryption engine. Performing these functions in hardware reduces software overhead, and hashes required for encryption, authentication, and proof of work (PoW) can be executed faster and with less power consumption. The host processor of each node can access the encryption engine acceleration through the security API and secure communication channels. Through this interface, the host processor will be able to run any encrypted application efficiently with hardware acceleration, such as running DApps, Light Client, or consensus algorithm. On the Addi side, the integration of secure storage and secure access to private keys will enable iot devices to autonomously execute cryptocurrency transactions. Users, owners, or managers will be able to configure their devices to allow for certain transactions and transaction frequencies, ensuring and additional levels of security.

The encryption core provides a highly secure platform for cryptocurrency private key processing and anti-operation authentication. It

provides a broad portfolio of services through its API, including authentication password libraries, MiFARE Plus and MiFARE DESFire libraries, hardware security features, and encryption engines. It can optionally work with NPU and biometric processing engines for n factor user authentication. It will handle the highest levels of security certification, including universal standards such as EAL6+, EMVCo and CUP. It supports the following basic functions:

1. MiFARE

Classic/DESFir/Plus

2. Cryptographic support

   (a) Message Digest: RIPEMD160, SHA224, SHA256, SHA384, SHA512, SHA3, SHA3-XOF, KEC- CAK

   (b) Cryptography Key Generation: DES (64, 128,192 bits), AES (128 bits), ECC (256 bits), RSA (1024, 2048, 3072,4096 bits)

   (c) HMAC Signature: HMAC-SHA256, HMAC-SHA512RSA Signature with PKCS1 v1.5, PKCS1 PSS schemes

3. Work to validate Operations performed and multifactor authentication (pin, passphrase, biometric auth, etc)

4. Private key recovery

5. Supports cryptographic libraries

6. Trusted and user mode of operation of the SW running on the node using hypervisors

7. Secure Boot ROM to build a chain of trust

8. Physically Unclonable Functions (PUF) to prevent device duplication 9. Tamper detection at the chip level with RAM clear and key erasure 10. Protection against grey market [13]

11. FIPS140-2 level 3 or more

12. Security Certification including EU Common Criteria Certification [14]

SuperSkynet Core's Crypto-Engine will allow IoT devices to store cryptocurrency in the hardware itself, en- abling them to use cryptocurrency securely. This means wearable devices will be able to store cryptocurrency and eventually, cryptocurrency will become user-friendly.

3.3 Neural Processing Unit

To leverage the current advances on Machine Learning on image classification, natural language processing, speech recognition, etc. we'll include a Neural Processing Unit (NPU) optimized to accelerate all current types of neural network algorithms, including DNNs, CNNs, and RNNs. Additionally, the NPU will be a fundamental component to enable high-security user authentication through biometrics. To explore design spaces efficiently, scalability is achieved by replicating as many NPUs as required. The scalable NPU ar- chitecture addresses a wide range of requirements of lower and higher-end applications, from accelerating embedded IoT devices with deep learning and proof of work mining by individuals through cell phones with built-in NPUs.

The NPU will serve as the brain of the IoT device, allowing it to perform classification tasks with human- level accuracy at a practical throughput and within a practical power budget. The main host processor of each node will access the functionality accelerated by the NPU via a secure API and secure communication channels. Through this interface, the host processor will be able to efficiently implement custom data processing applications by loading pre-trained neural network models into the core, injecting data into it and reading back partial of complete activation results. These networks can be stored in the IoT's ROM at time of manufacturing, or securely acquired, improved and updated later trough blockchain transactions.

The NPU block diagram shows the main abstraction. In here, local memories for neurons weights from the model to be run will be fed from the main host processor at appropriate times via the main system's AXI bus. Also, there is a local inter-layer memory for activations that will first hold the input data to the network (either full images, patches, of batches of images or patches), and then as each layer in network is processed by the Multiply and Accumulate unit, and the nonlinearity is applied, the output

of one layer gets stored back to the activation memory to be used as the input to the next layer in the neural network. At the end of the network, the final result is stored in the activation memory from where the host processor can fetch it.

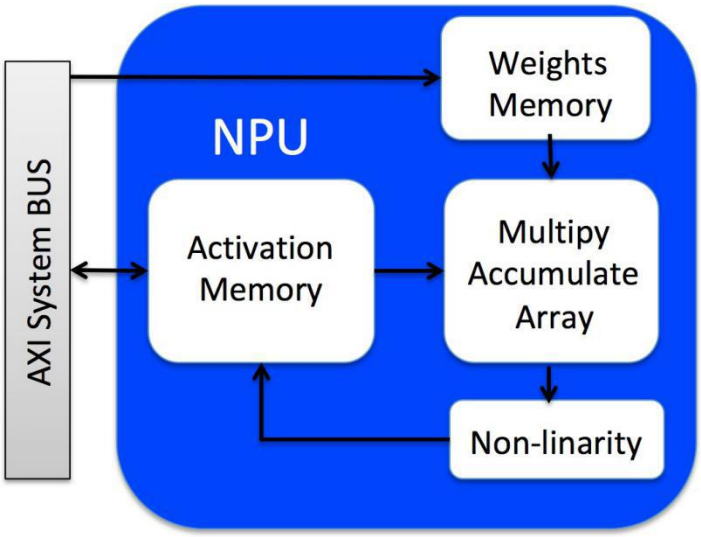


Figure 5: NPU Block Diagram

A vital requirement of any machine learning accelerator is its integration with training and deployment tools that have become standards in the field. Therefore, OpenSingularity will develop the necessary backends to Tensorflow, Keras, PyTorch, Caffe, etc. to support directly running these tools on our custom NPU, and integrate these into the SSN Blockchain API. As part of the adoption of these tools, we'll support emerging open interchange standards such as Open Neural Network Exchange (ONNX) so that developers can easily migrate their applications into our hardware.

By accelerating neural network algorithms, we foresee that the developers may choose to use the NPU to build DApps with integrated learning. These applications could progressively finetune pre-trained networks or leverage the latest advances in transfer learning to achieve higher accuracy and specialization. The OpenSingularity NPU is not intended as a platform for experimenting with new NN architectures nor as a replacement for high-performance NN training workstations such and NVIDIA's DGX or TPUs. In principle, although incurring considerable overhead over integrated solutions, learning through the default back-propagation algorithm can be done by simply computing the forward pass through the network in hard- ware, and using software to store the gradients and compute the backward pass. The host processor may manage other forms of learning such as reinforcement learning or evolutionary learning, using acceleration from the NPU as practical. Overall, the vision behind supporting the NPU as part of a learning framework, as opposed to using high-performance GPUs, is an analogy to the "Tortoise and the Hare" story where slow and steady (via a large collection of distributed IoT devices) may lead to interesting developments.

The NPU will also play a critical role in providing high-security to the system, enabling N-factor user authentication in some applications. For example, in hardware wallets of smartphones with SuperSkynet Cores, the NPU could receive input directly, by a secured physical channel, from biometric sensors and would directly enable the hardware wallet when a valid user identification detected. The biometric data could enable iris or retina scans, face identification, fingerprint matching. Additionally, the host processor could leverage the NPU to provide an additional layer of security by keeping track of ongoing patterns of transactions and authentications, and use an anomaly detection algorithm, to default the system to a secure state if an anomaly is detected.

3.3.1 Neural Network Operations

Over the past two decades Neural Networks have established themselves as a computational tool that for solving problems only humans were capable of. In traditional programming paradigms where developers establish a set of rules for the computer to follow in solving a problem. In Neural Networks developers define a set of nodes (neurons) and connections and use an optimization algorithms to find the best parameters for a given problem. In this section we'll provide a brief overview of the current

state of Neural Networks and their computational requirement, but a full review of the subtitles of each step is beyond the scope of this document. For more details, the reader is encouraged to follow Stanford's CS231n online course as a basic introduction into the subject.

Neural networks are loosely inspired by the vertebrate brain structure where neurons are highly specialized cells that receive inputs from other neurons through the dendrites, and then transmit a signal through the axon when the sum of inputs is greater than some threshold.

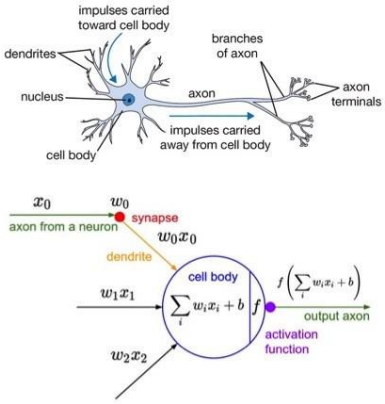


Figure 6: Analogy between a biological neuron (top) and a mathematical neuron (bottom)

In an artificial neural network, layers of nodes or neurons are interconnected in such a way that neurons from one layer make connections with neurons of another layer, and each connection is assigned a weight7.

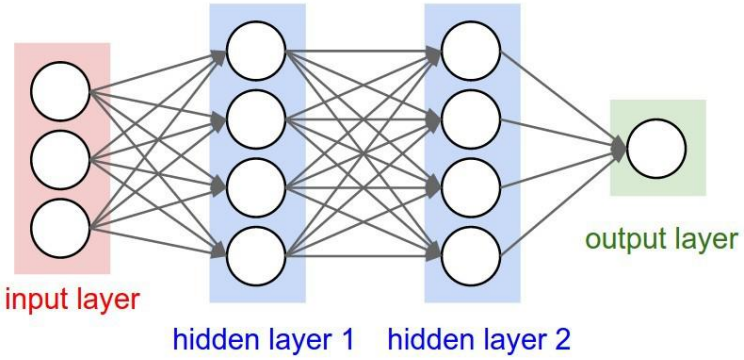


Figure 7: Simple Neural Network Architecture

Biological neurons have extraordinarily intricate dendrite integration trees, exhibit rich temporal and modulatory dynamics at every synapse (connection between axons and dendrites) within the dendrites, at the soma (cell body) and through the axon. In contrast, artificial neurons are represented by a very simple mathematical model where the output of each neuron can be defined as $y_j = f(\sum_i w_i x_i + b)$, where for each layer w_i is the weight of the connection between the ith neuron in the previous layer and its activation x_i, b is the activation threshold and $f()$ is a nonlinear activation function that can take several shapes. A very popular activation function is the Rectified Linear Unit (RelU) but the sigmoid, and tangents are commons as well.

In practice, the neural network algorithm can be cast into a linear algebra operation $Y = f(W \times X)$ where W is a matrix of weights whose rows represent all the connections between a previous layer and a neuron in the next layer. For this reaSSN, GPU and other forms of Matrix-Matrix or Vector-Matrix multiplication hardware has become so popular in accelerating the evaluation of neural networks.

Furthermore, a highly successful variant of neural networks has been the Convolutional Neural Network (CNN) where instead of having all neurons in one layer connect to all neurons in another layer, each layer is defined by a filter or constitutional kernel that gets shifted through the input space. This has proven to have enormous advantages by reducing the number of parameters that a network has—each layer only needs the parameters of the filter and not the full permutation matrix—and by creating filters that are applied through the same way through the input space, therefore achieving spatial invariance. There are many different network CNN architectures, and some are featured in.

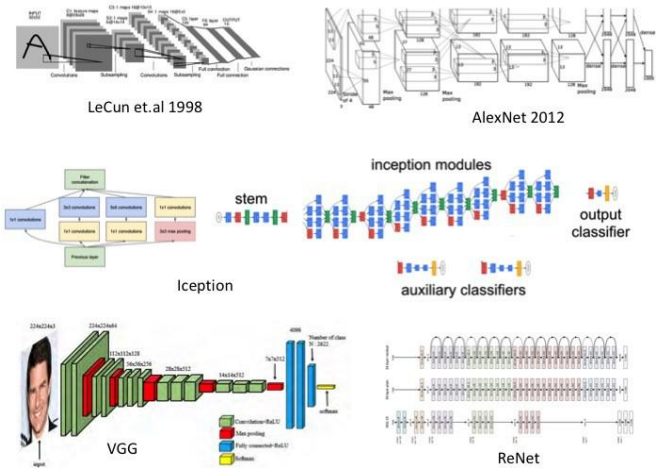


Figure 8: Multiple Convolutional Network Architectures

3.3.2 Neural Network Computational Requirements for Modern Networks

Despite the apparent simplicity of neural networks, there is enormous computational complexity behind them. As can be seen in the previous section, neural network architectures (8) have grown in complexity and scale, and modern models have millions of parameters and perform billions of mathematical operations
(9) to classify the contents of a small patch of image [15].

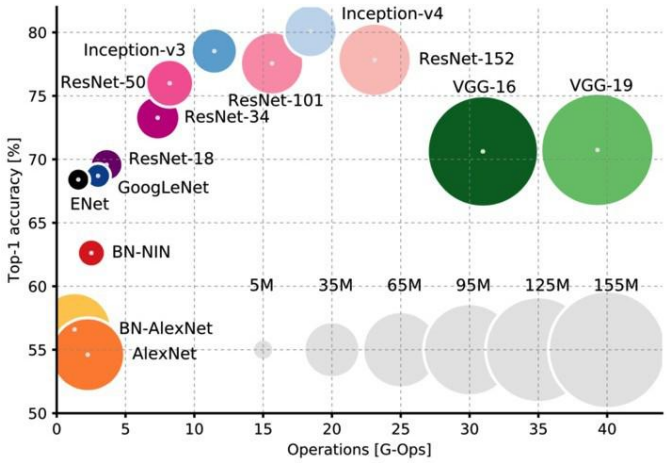


Figure 9: Giga-Operations and number of parameters required for each network

Note that the computation required to classify an image is in the 2.5 to 40 GOP, yet the report does not clarify if this is for a single patch of the image of for multiple. For this reaSSN, the actual computation required for a full HD image may be 100 to 1000 times greater. Inference time10and memory11usage measurements used Torch7 with cuDNN-v5 and CUDA-v8 back-end.

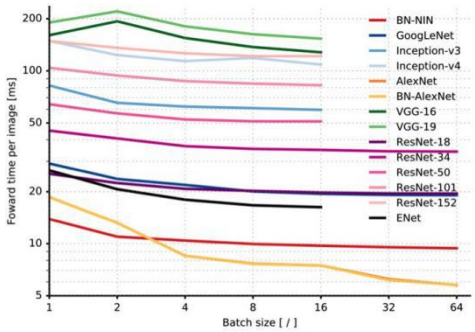


Figure 10: Inference Time vs. Batch size.

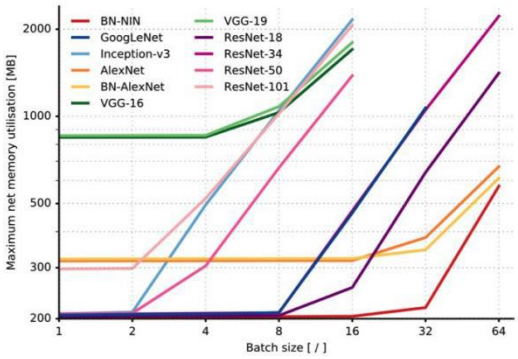


Figure 11: Maximum system memory utilization vs. batch size. Usage shows a knee graph due to the network model memory using a static allocation and then variable memory used by larger batches

Power consumption hovers around the 12 W mark for all models12. All experiments were conducted on a JetPack-2.3 NVIDIA JetSSN TX1 board (NVIDIA): an embedded visual computing system with a 64-bit SNM-A57 CPU, a 1 T-Flop/s 256-core NVIDIA Maxwell GPU and 4 GB LPDDR4 of shared RAM.

3.3.3 NPU Architecture: GPU vs TPU

A key innovation in the field has been to use the Matrix multiplication engines used to render images in Graphics Processing Unit (GPUs) to compute the workloads of Neural Networks. This has been one of the enabling factors that allow much faster training as well as larger models. An interesting trivia is that the

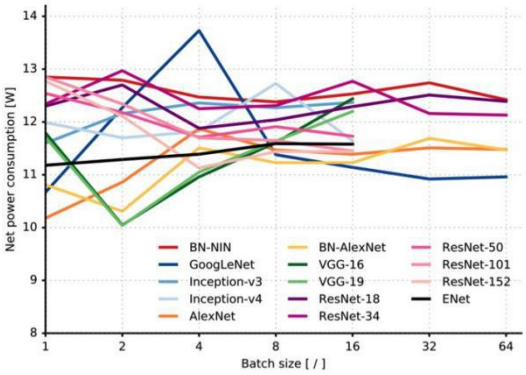


Figure 12: Power Consumption required for each network on the JetPack-2.3 NVIDIA JetSSN TX1 board. Baseline at Idle is 1.3W

revolutionary AlexNet network that unleashed this movement is broken up into two main branches because each branch could be run in an independent GPU. To expedite training data scientists put great effort into creating network architectures that maximize (but not exceed) the memory capacity of GPUs. This in turn has created a feedback cycle where GPU manufacturers (NVIDIA in particular) are designing GPUs with larger capacities specific for these workloads. As of this writing, the pinnacle of machine learning computing is the NVIDIA DGX-1 workstation with Tesla V100 GPUs that can process 1000 TFLOPS (deep learning). In general, GPUs work better than CPUs for machine learning because they have a much larger number of computing cores and faster access to memory. This computational advantage is extremely important because it can reduce network training from months to hours. The reader is referred to an excellent slideshow from NVIDIA showcasing the advantages of GPUs for deep

learning [16].

Beyond GPUs, when Google realized that neural networks would overtake the performance of traditional computing for translation services[17] (and others), but that this would triple their computational require- ments, they designed their own Tensor Processing Unit. Figure13shows the physical implementation of the TPU, how its architecture mimics very directly the computation required to process the common layers of a neural network, its integration stack and how with this implementation they achieved an impressive computational efficiency 89 times greater than Using CPUs and 29 times greater than using GPUs (of that era). This, and the newer generation of TPUs are available for use by the public through the Google Cloud. The reader is referred to an excellent overview of the TPU [18] or the original TPU paper [19].

3.3.4 NPU Available Offerings

Through the past decade, the industry has recognized that neural computation and custom accelerators were required to move forward the field of Artificial Intelligence and machine learning. This was perhaps catalyzed by DARPA's SyNAPSE project which led to TrueNorth, one of the first formal efforts to productize neural network accelerators. Today there are dozens of players in this field that offer mature and accessible NPU

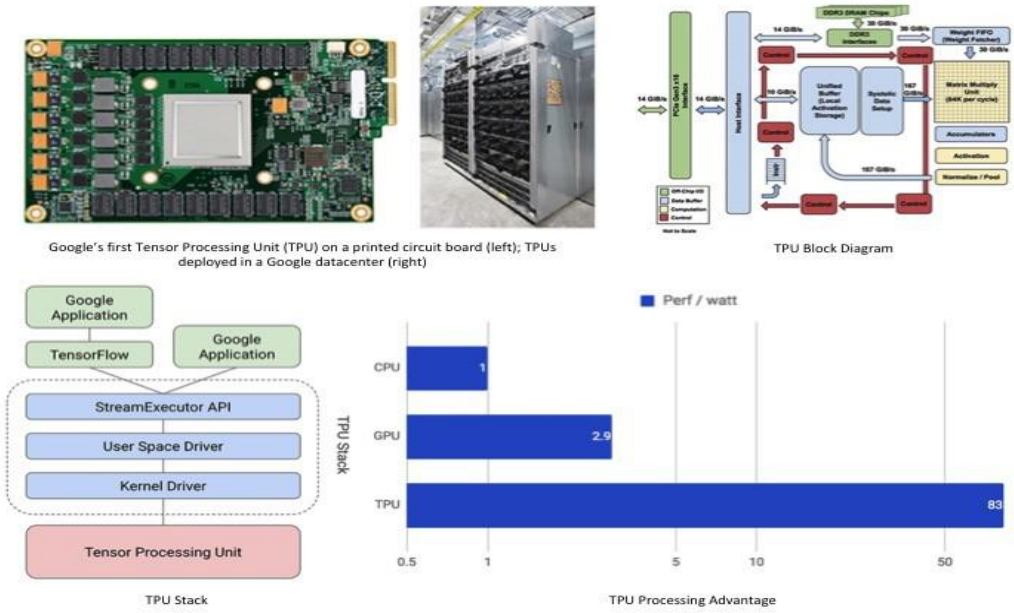


Figure 13: Google First implementation of the Tensor Processing Unit

acceleration at multiple scales. At the ASIC or SoC level, SNM [20], Synopsis [21], Cadence [22], offer supported IP blocks ready for integration into silicon products. Further up the stack, several hardware manufacturers offer readily available chips, module, and workstations for accelerating neural processing such as: Nvidia [23], Intel [24], Bitmain [25]. From a cloud perspective, NPU acceleration is available from NVIDIA, Google, Amazon, Microsoft, and IBM. Finally, it is rumored that there are approximately 35 startups pursuing NPU acceleration products such as Fathom computing, Mythic (previously Isocline), Groq, Wave Computing, Cerebus, GraphCore, Ambiq Micro and Knupath.

3.3.5 Potential for Optimization

Despite the enormous amount of funding and momentum in the NPU field, it is unclear if there will be successful players that develop solutions optimized for IoT devices. As such, the OpenSingularity Foundation will focus on identifying the best-suited applications for Machine learning at the IoT edge and run extensive workload profiling. From there, we will compare how these specific workloads map onto existent available solutions in search of potential for improvement. Finally, once we have identified the specific space between the existing solutions and the required needs, we will conduct an extensive review of cutting-edge techniques and IP landscape to develop a set of NPUs customized to solving the NPU needs of IoT edge devices.

4 SuperSkynet Network

In theprevious sections, we detailed the SuperSkynet Core and how the components will be used to utilize the applications on SuperSkynet Network (SSN) and of blockchain technologies as a whole. We also detailed a high level overviewof the SSN network structure and its adoption plan. In this section, we introduce SSN, an infinite-chain network that will serve to link all intelligent devices and blockchains under one decentralized system. By bootstrapping the network off of the SuperSkynet Cores, SSN allows for global adoption of blockchain technology by providing billions of devices immediate access to its network and other networks.

4.1 Network Introduction

SuperSkynet Network is comprised of four main frameworks: SSN Fabric, SSN Nova, SSN Idem, and SSN Singularity. However, the latter three frameworks are all connected to SSN Fabric, the root blockchain platform enabling an infinite amount of other blockchains and frameworks that connect to it.

Table 6: SuperSkynet Network Frameworks

SSN Frameworks	Description	Applications	Native Tokens
Fabric	Blockchain Platform	Proof of Stake Blockchains Cross Chain Communication	SuperSkynet Light
Nova	Distributed App Platform	Scalable Smart Contracts Web3 and Ethereum Compatibility	Light
Idem	Decentralized ID Platform	Secure Node Discovery IoT Device Management	Light
Singularity	Machine Learning Platform	AI KnowledgeNet Decentralized Machine Learning	Singularity

s

With the frameworks shown in Table 3, SSN can provide the end-to-end solution with a development platform and applications to support the interactions between autonomous devices.

4.1.1 SuperSkynet Network, Fabric

SSN Fabric is a publicly validated, Byzantine Fault Tolerant, Delegated Proof of Stake blockchain and the "root"of the SuperSkynet Network. Fabric contains a Go-Language software development kit, enabling developers to make fast public or private, fault tolerant proof of stake blockchains independent of Fabric's governance.

With Fabric, blockchains can become their own VM-independent platform or be used to interact with the underlying scheme of other blockchains. Fabric only keeps track of the tokens on each blockchain created on it, allowing for a type of cross-blockchain communication where each blockchain can be independent but are able to exchange data packets with one another through it.

Table 7: SSN Blockchains CompariSSN

Properties	SSN Fabric	Sub Chains
Type	Public	Public or Private
Consensus	Delegated Proof of Stake	Proof of Stake
Validators	100 to 500	4 to Infinity
Finality	Instant	Instant
Privacy	No	Yes or No
Turing Complete	No	Varies
Governance	Yes	Soverign

Shown in Table 4, Fabric will start with 100 validators and have its own governance mechanism. However, subchains on Fabric are independent of one another and have their own network designs, allowing them to be isolated from the failures of other networks. This is enabled by a Byzantine Fault-Tolerant Consensus Algorithm called Tendermint Core, that takes state translation machines in any language and replicates it across all machines. Tendermint Core is, as a result, well suited to handle many IoT subsystems and many low latency, high finality, blockchains that are well architected to function in the real world. This makes Fabric a modular platform for deploying high throughput blockchains with minimal resource consumption.

Sub-Chains In this paper, sub-blockchains created on Fabric are referred to as IoT Chains. IoT Chains can be created by developers for their own purposes or to interoperate with Singularity's existing chains. To create IoT Chains, Fabric comes with a toolkit, which provides boilerplates for on-chain storage data type customization, multi-data type on-chain storage abstractions, private blockchains, and public blockchain creation. With Fabric's software development kit, any existing blockchain like Bitcoin and Ethereum can be created as an IoT chain but with infinite scalability and an energy efficient Proof of Stake fault tolerant consensus.

SuperSkynet Token SuperSkynet Token is the native staking token of the SuperSkynet Network and SSN Fabric. In Proof-of-Stake blockchains, the creators of each block are chosen by random selection in a round-table like fashion according to how much coins or value the perSSN holds. To provide incentives for participants to stake the currency, the SuperSkynet token is solely designed for staking whereas block rewards and fees are distributed in another token. Interestingly enough, SuperSkynet tokens can be used for staking with other IoT Chains.

Light Block rewards and fees are paid in a currency called Light. Light is the native fee token of the SSN Nova platform as well as the SSN Idem platform and can be used across all blockchains.

4.1.2 SuperSkynet Network, Nova

Connected to SSN Fabric is Nova, a modular smart contract platform with its own enhanced Ethereum Virtual Machine (EVM) called Quantum. Nova's platform will allow for distributed applications to be built on the SuperSkynet Network while removing the drawbacks of Ethereum such as transaction time and fees.

At first, Nova will simply be a Proof of Stake Ethereum powered by Tendermint Core and a virtual machine built in part to the specifications of EVM. With a similar virtual machine, Nova will allow for inter-operability between existing Ethereum distributed applications and Web3. Nova will also enable developers already familiar with Ethereum to migrate to SuperSkynet Network and begin developing IoT-based applications that can be adopted immediately across devices with the SuperSkynet Core. These benefits make Nova a modular platform for developing scalable decentralized applications immediately usable in IoT devices.

4.2 SuperSkynet Network, Idex

Connected to SSN Fabric is Idex, a hybrid sub-chain distributed ledger built to create a decentralized identity and a crypto phonebook for IoT devices. SSN Idex provides the tools necessary for devices to publish information that other independent blockchains and applications can access and query. Since the network is immutable and public, any device can join the network and start finding devices over the network. Attached to the distributed ledger is an off-chain explorer that devices can query to examine the transactions and history of other devices. Here Idex provides a machine reputation service where device addresses can receive ratings from 0 to 100 depending on how reputable a machine might be. Idex can then be combined with other scalable platforms to make a whole new scope of applications such as algorithms for secure machine to machine transactions and self-organization.

For all these reaSSNs, SuperSkynet Network Idex provides all the designs and specifications necessary to support decentralized identities and its resulting potential applications.

4.3 SuperSkynet Network, Singularity

SuperSkynet Network, Singularity, also known as an AI KnowledgeNet or Virtual Application Layer, is an extension of Nova and Idex, enabling a series of interoperable applications for interactions and learning between SuperSkynet Cores and other IoT devices. More specifically, these are for decentralized machine learning, distributing computation, and data sharing. The applications can be tied in with Singularity's multi-chain marketplace where devices can agree on values for their training data or computational power.

Both the the distributed applications and the marketplace make up SSN Singularity. Developers can make their own distributed applications on Nova or perhaps combine it with Idex and have them be interoperable with the existing applications on Singularity's virtual application layer. Real-world devices and SuperSkynet Cores can then utilize the applications and the cryptocurrenices that the distributed applications offer. For example, if a developer wanted to create an application for distributed evolutionary learning on Nova, devices could access something called the virtual application layer and have access to the network's protocols and cryptocurrenices.

4.4 SuperSkynet Network Architecture

These four frameworks and their native applications and protocols make up the SuperSkynet Open Network. Beneath their platforms lay a three-layer architecture. On the very bottom, Tendermint core provides the consensus engine and P2P communication to form the base of the SSN. Above Tendermint Core, lies SSN's SDK which implements blockchain logic for the cryptocurrencies, smart contracts, identity, staking, and governance. SSN's SDK interfaces with Tendermint core via ABCI, short for Application Blockchain Client Interface. On top of the Singularity SDK, applications can be implemented.

4.5 Application Blockchain Client Interface

In SSN, the interface between multi-machine state translations and is used to communicate between Tendermint consensus and the application layer. The ABCI is an interface that allows applications to be implemented on top of Tendermint Core in any programming language. ABCI is implemented in a socket protocol called Tendermint Socket Protocol (TSP).

The Tendermint Socket Protocol is used for communication between the application and Tendermint Core. Using this layer of abstraction, the Tendermint Core can be plugged into any application on SSN that can communicate via sockets. This provides a modular architecture on SSN for implementing blockchain systems. Typically, Tendermint Core would be responsible for sharing blocks between nodes and establishing the transaction order. Cryptographic transaction validation, incentive mechanism, and other blockchain primitives would be implemented at the application layer.

The Tendermint Core maintains three connections, mempool connections for using CheckTx for transaction relays, consensus connection for executing committed transactions, and a query connection for application states.

Mempool connection (CheckTx)

- Checks if transactions are valid (only lightweight checks) and should be executed and broadcast to other nodes (through DeliverTx)); only uses CheckTx
- Performs checks by using the "Mempool" as a starting state (list of accounts, current balance, and any other relevant information stored in the state).
- Starts as a copy of the last committed state

Consensus Connection (BeginBlock, DeliverTx, EndBlock, Commit)

- Executes and broadcasts transactions that have been checked. Message sequence is - for every block - BeginBlock, [DeliverTx, ...], EndBlock, Commit

Query Connection (Query, Info) - query application state without engaging in consensus (= read-only) (Query)

- handshake (Info)
- genesis (initChain).

Otherwise, the ABCI design has a message protocol defined using protobuf and the server implemented by async raw bytes and grpc [29]

Info: used to communicate current state between ABCI client (tendermint) and Server (the application encapsulating business logic)

Flush: used as a means to communicate that a message has been received and processed. It is send after receiving the associated response to a request sent.

InitChain: called to initialize a new node. In the case of the first node, it will also initialize the blockchain, in the case of a new node in an existing blockchain, it will just catch up with the other nodes by replaying past transactions.

checkTx: sends the Transaction to be "prechecked" before it is send to all validating nodes for processing and integration in the current block.

deliverTx: send the Tx to all validators and executes the Tx (if relevant)

Commit: commits the state with all accepted Txs. This writes the state such that the next block can begin and increase the block height.

beginBlock: opens a block for the inclusion of new Txs

endBlock: closes

setOption: allows setting of local, nonconsensus critical options on the node. For example, log level of the app.

Query: allows querying of the state without impacting it (read-only). This operation is performed locally on the node)

4.6 Validators and Delegators

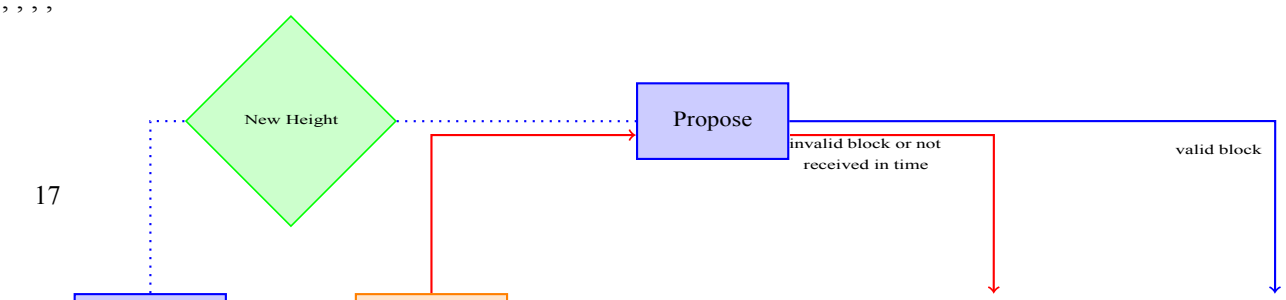
In SSN's Tendermint Consensus, validators can participate in the consensus by broadcasting cryptographic signatures that act as votes for the next blocks. To become a validator, a node must lock up a predetermined amount of tokens. Delegators, someone who wants to contribute voting power to a validator, delegates the same token to a potential validator, so that the delegate might earn a part of a block reward. Delegates are putting their tokens at risk by delegating their stakes to validators and may lose tokens whether or not the validator behaves in line with the protocol implementation.

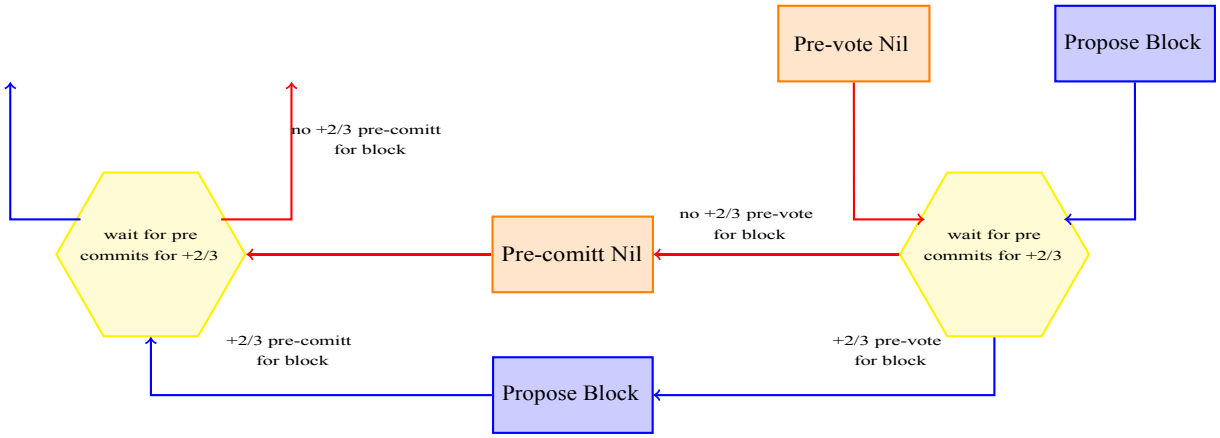
Validators have a voting power equal to that locked up in a bond transaction and may unlock the coins by posting an unbonded transaction.

A minimum of 4 validators are needed but can scale infinity to run the consensus protocol on SSN. However, in the SSN Fabric, we will begin with 100 validators and scale to 500. These validators can help run the other networks on SSN.

4.7 Tendermint BFT dPoS

SSN's Tendermint Byzantine Fault Tolerance protocol is a modified version of the DLS protocol and is resilient to up to $\frac{1}{3}$ of Byzantine participants. The consensus protocol requires no proof-of-work mining and protects against double spending. Tendermint's algorithm is based around the FLP impossibility result from Fischer's research in asynchronous systems. [30]The algorithm assumes that the network is partly synchronous and that non-byzantine nodes can utilize an internal clock until the next block is published.





[30]M. J. Fischer, N. A. Lynch, and M. S. PaterSSN, "Impossibility of distributed consensus with one faulty process," Journal of the ACM, vol. 32, no. 2, pp. 374–382, 1985.

Figure 11: SSN Cross-Blockchain Communication protocol [31]

4.8 IVAL+ Data Structure

SSN uses an IVAL+ Data Structure that is similar to that of Ethereum's Patricia trees. This data structure is there to fast computation for deterministic Merkle root hashes and storage for key-value pairs.

SSN uses a merkalized IVAL+ (Go 1.8+), a balanced variant of AVL trees to ensure the blockchain state cannot be tampered. In short, the AVL+ algorithm modifies the AVL algorithm to keep values on leaf nodes while using branches to store keys. It is a key value pair storage allowing for a deterministic merkle root hash for computation, which guarantees the integrity of the structure from one block to the next. As it is a variant of AVL, all the operations are Olog(n), and the nodes are immutable and indexed by their hash in the tree. The nodes serves as a some timestamp for uncommitted mempool transactions, so that they can roll back the last commits for the new block. SSN's IVAL+ is a more efficient algorithm adaptation of AVL.

4.9 Light Clients

Native support for light clients makes SSN particularly useful for IoT applications, in which nodes may have limited resources. In contrast to IOTA's system that targets IoT applications that require a heavy Java-based gateway node implementation, SSN's consensus is designed to support light clients that do not have to store transactions locally. This is achieved by allowing applications to include the root of a Merkle tree in each block, which can be used to verify state queries or transaction outputs. This allows SSN to enable light client protocols, which are designed to allow users in low-capacity environments to help maintain a certain state of the network. This means light clients protocols are great in IoT devices such as smartphones, watches, and tablets.

SSN enables applications to embed a Merkle Tree hash in each block to verify state queries or transaction outputs, similar to the structure of Ethereum's light clients. With SSN's underlying consensus, the network solves the nothing-at-stake predicament by utilizing deposit collateral, allowing light clients to know when a validator is going to change and then verify $> \frac{2}{3}$ of the pre-commits to know the latest block state. However, with our SuperSkynet Core devices, small IoT devices should be able to run full nodes.

4.10 Cross-Blockchain Communication

SSN contains an cross-blockchain communication protocol (CBC) to allow blockchains on SSN to exchange tokens and information with one another. All exchanges between blockchains are done with something called CBC packets in which packets of information is sent through Fabric to the other blockchains.

One way to do cross-chain atomic swaps is shown by hash time locked contracts in the Lightning Network. However, SSN's CBC's protocol can create 2-way sidechains, enabling exchanges between blockchains with instant finality that can enable a transfer of information or value. [35]

More specifically, the CBC protocol contains two types of transactions: a packet transactions, which

enables a blockchain to prove that a packet was published by a sender via the most recent block hash Merkle-proof and a block commit transaction, which enables blockchains to prove its most recent block-hash to an observer.

In this manner, SSN allows for the receiving chain to acknowledge which CBC packets are committed while allowing what outbound packets are allowed.

The concept of cross-blockchain communication can then be applied to things such as:

Multiple Virtual Machines - SSN Fabric only communicates to other IoT Chains with CBC, so each other blockchain can be sovereign and have their own virtual machines, applications, and governance.

Distributed Exchange- SSN Fabric can be used as a decentralized exchange to swap tokens between IoT Chains.

Cross-Chain Bridge - Chains on SSN Fabric can serve as a bridge to other blockchains like Bitcoin by verifying states in SSN and on other blockchains.

In this manner, cross-blockchain communication is a vital component of having an infinite amount of interoperable, self-governing blockchains on the SSN.

4.10.1 Inftnite Sharding Paradigm

SSN handles infinite sharding through its IoT Chains. SSN Fabric ignores the state of its IoT Chains but rather listens to communications through CBC packets, so each shard can be its own sovereign blockchain.

Unlike with proof of work consensus blockchains, with SSN's Tendermint consensus, running an infinite amount of parallel blockchains does not diminish either the speed or security of each IoT Chain.[36] As each chain can handle thousands of transactions, spawn an infinite amount of chains, and have sub-chains work together, SSN can scale to infinity to handle any amount of IoT interactions.

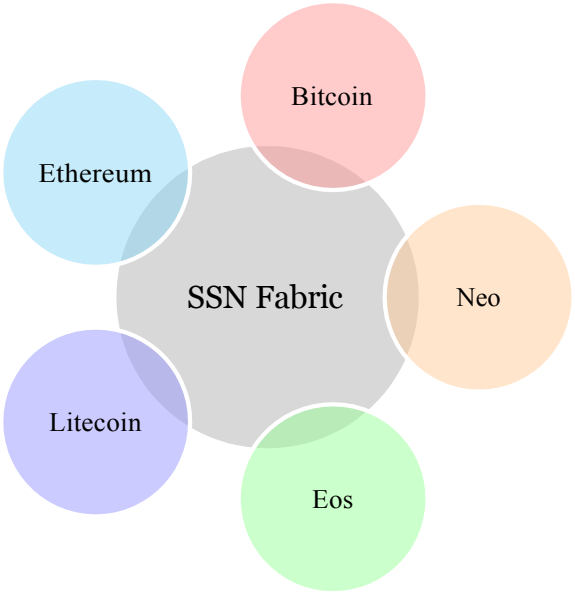
The main difference between sharding with SSN and other blockchains is that on other blockchains, the shards depend on the general machine state while SSN preserves the number of tokens between chains. This means that on SSN any blockchain with entirely different virtual machines can be created and can fail, while on other blockchains, none of the shards should fail. However, in SSN, other types of sharding can be implemented and tied in within the network.

4.11 SSN Fabric

SSN Fabric is a competitively validated delegated proof-of-stake platform. The hub maintains the number of tokens on each IoT Chain and enables a seamless relay of data between blockchains. This means that the hub serves as a global bridge between all public and private blockchains on SSN while also serving as a distributed exchange.

4.11.1 Fabric Entangled Chains

On SSN Fabric, its native cross-blockchain communication protocols allow it to interoperate with its existing chains. However, since we acknowledge since there are a lot of applications that people make on other chains, we will enable something called an entangled chain which provides a bridge and interoperability with existing blockchains and their native cryptocurrencies such as Bitcoin or Ethereum. All that is needed for a entangled chain to serve as a bridge is some type of pseudo-finality on the other blockchain where there is some process that determines the finality of the block.



For example, on SSN, one entangled chain can serve as a bridge with Ethereum. To provide some background, the main differences between Tendermint and Ethereum goes as follows. Tendermint uses go-wire for serialization while Ethereum uses Recursive Length Prefix. Tendermint uses ed25519 where in compariSSN, Ethereum uses secp256k1. Lastly, Tendermint uses IVAL+ Trees while Ethereum uses Patricia Trees for key values.

Currently on Tendermint, there exists a protocol called ETGate which serves as a bridge between Tendermint-based blockchains and Ethereum. In the protocol, it decoded packets within Ethereum's virtual machine. However, converting every block into a compatible variant within the Ethereum Virtual Machine is too gas costly for SSN. In order to provide a gas-friendly bridge from SSN to Ethereum, an ABCI app will receive a relay message from the SSN Fabric, and the ABCI app will write an Ethereum transaction containing the address, denomination, amount, and nonce. The Signing Apps will then detect transactions from the ABCI Apps and sign transactions using secp256k1. The Signing Apps will relay messages back for replication and the relayer Signing App will query the ABCI app's transactions and process those that reach the required threshold The relayer Signing App will send a transaction to the Ethereum smart contract and the smart contract will send a Light ERC20 Token to the user's Ethereum address.

On Fabric, it's easy to transfer light to the entangled chain, and once the entangled chain receives an CBC packet, signers can convert the signature into Ethereum's native secp256k1 format. Validators can then wait for 2/3 of transactions to be complete and then relay the information to Ethereum, in which we will create on smart contracts to enable the interoperability of SSN's native tokens and Ether. Once the light is sent, the smart contract can then send an ERC20 light variant to an Ethereum address where the IoT device is able to convert it to Ethereum via a distributed exchange. The development of entangled chains is still in its early stages, and more updates will be provided as the project continues. SSN's entangled chains serve as a global bridge to enable interoperability between all major blockchains.

4.11.2 SSN Fabric Tokens

On the SSN Fabric, there exists two types of tokens: one for staking and one for fees. They are both respectively called SuperSkynet and Light. SuperSkynet is the only staking token on SSN Fabric and is used to vote, validate, and delegate validators. Light is used for a transaction fees to mitigate spam. Because SSN's consensus algorithm can replicate different deterministic states, more than one coin can be built upon each chain since SSN Hub tracks multiple different token states.

For this reaSSN, the multi-token economic model was created to address the problems of current proof of stake models.

For example, when Ethereum switches to Casper, it has one native token: Ether. As Ether has more utility than staking, such as paying for goods, a large number of tokens will not be staked and as a result, weakens the security of the protocol.

As SSN is an interoperable multi-token network, we can introduce both SuperSkynet and Light to address this concern. SSN's utility is for staking only and will be used to earn transaction fees and block rewards on SSN Fabric and hosted chains. One can think of the token like an SHA-256 ASIC miner. The

ASIC miner's main utility is to mine Bitcoin. The rewards are in Bitcoin, but in order to mine, one needs the ASIC. In this SSN's case, the reward is in Light instead of Bitcoin and the miner is the SuperSkynet token instead of the ASIC.

With this model, SuperSkynet's utility is to serve as the only staking token, which in turn would incentivize the governance and security of the network. In this manner, the majority of SuperSkynet will be staked in the network since it will be used just for staking. The fees collected by validators from computational costs from each transaction will be distributed proportionally to the number of SuperSkynet staked.

4.11.3 Validators and Incentives

In SSN Fabric, validators can stake their SuperSkynet tokens and can delegate the tokens to stakers. The hub at first will have 100 validators, but over time will create to 500 validators. Validators can stake their SuperSkynet tokens and in return receive Light for block provisions and transaction fees. As there are only a limited number of validators, nodes can delegate their Light tokens and contribute to the consensus; as a result, they will earn a percentage of transaction fees for participating or lose their share if the validator is malicious. Like other delegated proof of stake models, the more SuperSkynet Tokens one stakes, the more block rewards and transaction fees they get in return. When SSN Fabric is launched, validators will be chosen through a public vote, which will shift around validators when SuperSkynet tokens are delegated to others.

When a block is published, the provisions are proportionally distributed across validators relative to their stakes. If the block provision is 5000 Light tokens and each validator has 20% of staked SuperSkynet tokens, and the commission fee is 2% across 10 validators, then the 500 tokens will be distributed across:

Commission: 500 * 80 % * 2 % = 8 Light
Validator: 500 * 20 % + Commission = 108 Light **Delegators**: 500 * 80 % - Commission = 402 Light

Each delegator will receive a distribution of the 402 Light in relation to what it delegated to the validator pool. If a validator is malicious, such as when it commits signing, it is easy to tell on SSN because only two conflicting votes are needed. The validator will immediately be dissipated after a slash transaction is committed. Initially, 5 percent of Light tokens will be inflated every year; however, this value will change to incentivize validators to stake two-thirds of the SuperSkynet tokens and depending on the governance of the hub.

On genesis day there will be 100 validators, and will increase at a rate of 15 percent per year until it reaches 500 validators. The block reward for Light will be determined at a later date but will be at an inflation rate that asymptotically reaches zero. Validators on SSN Fabric might help validate other IoT Chains such as SSN Nova in the very beginning.

4.11.4 Slashing

If a validator misbehaves, it loses its staked SuperSkynet tokens along with Light. This happens when it double signs, such as if a validator reports that on Chain X, a validator signed two blocks with the same height on Chain X and Y. If that is the case, the validator will get slashed on Chain X. Next, if a validator's signature has not been included in the last x amount of blocks, the validator will get slashed a proportional amount of x. If it surpasses a number y, then the stake will be removed. If someone reports that a validator did not vote, a minor slash will occur. Moreover, validators can be slashed if the node gets DDOSed, the private key gets hacked, it loses connection, and if the node crashes.

4.11.5 Governance

On the SSN Fabric, validators can vote on things such as block gas limits in relation to parameter changes, coin inflation, updates to the policies, as well as vote on terms and services that govern the SSN Fabric. Each validator is required to vote or else the validator will be deactivated for 2 weeks. Each vote proposal requires an x amount of tokens on SSN as a stake deposit. If the proposal was spam, meaning that the votes were majority negative, the deposit would go into something called a reserve pool.

For proposals, validators can vote with either: Yes, No, and Abstain and a strict majority is required for a proposal to pass. More updates regarding governance specification will be revealed close to the mainnet launch.

Aside from the SSN Fabric, each blockchain on SSN can have its own governance and constitutions, as they are sovereign blockchains.

4.11.6 IoT Chains

The iot chain is a public/licensed or private/consortium blockchain of high-throughput areas or specific devices, each powered by the Tendermint BFT consensus connected to Fabric. While each iot chain can handle thousands of transactions per second, the billions of internet-connected devices that use the network could cause write fees to rise, and in the real world, no blockchain or DAG can handle more than 30,000 transactions per second. Iot devices are also the opposite of "" one hand at a time," "as devices on a single blockchain are forced to use different data types that are simulated in common containers. As a result, each chain USES Tendermint's ABCI, allowing developers to create more distributed replicas of the iot chain blockchain and split up network users to create unlimited scalability. The SSN toolkit allows you to build custom device type chains for specific iot applications because native types Perform better. Each iot chain can also be launched from the iot ledger, allowing validators in each iot chain to verify their
location, which in turn creates geographically specific public chains that end quickly.

Iot chain is a block chain that can form various shapes or forms on the fabric, such as local, global, public, pri-vate, commonwealth, geographically specific blockchain, manufacturer operation or user operation. The iot chain on Fabric comes with an open-source boilerplate software development kit for building an interoperable iot chain, a simple guide for anyone who wants to create an iot chain with a SuperSkynet core. Each iot chain is connected to the Idem identification chain and the second layer network

Systems that allow them to create new identity-based applications instead.

4.12 SuperSkynet Network, Idem

SSN Idem includes a built-in identity protocol layer that allows machines to find each other, self-organize, and start developing so-called machine reputation. This layer allows the device to determine whether a random node is malicious or is in the same domain of knowledge. Identity protocols will also enable people to determine whether their machines are working properly, as nodes in the network will interact with other known nodes in the public ledger. Smart contracts in Idem are updated when transactions are processed through a Singularity distributed application.

Unique identities will be added as transactions and stored on the side chain of the hub via SSN's SDK to provide permanent identity data records. The off-chain database will take information nodes, such as reputation scores, which will depend on an algorithm generating a score based on how many interactive nodes there are prior to experiencing, how many cryptocurrency nodes there are, and how much data the nodes have. The algorithm and more details about the exact nature of the identity protocol will not be made public because it is not in the best interest of the device to start the game system. However, more details will be announced near the release of superskynet's core.

4.13 Beacons

The identity of machines is much simpler than that of humans or groups of humans who own them. At the most basic level, the machine id is an address on the network that does not appear to belong to none be-

Because its owner has privately listed it. For machines that require public release of information, machines can have human-readable addresses, such as John's. Doe/weather station or Microsoft/weather station or cleve-land /parkingmeter123534. The beacon is the system that registers the machine on the ID chain of the SSN fabric. It allows a machine to publish information about itself, usually once, but it can hold money, and of course it can publish it multiple times.

Specifically, the device stores a "beacon" on the chain. This beacon function is similar to device ID and allows for peer-to-peer access to a particular device, regardless of its location on the network. The cost of registering a beacon is minimal to prevent spam. The beacon can be highly descriptive or completely minimal. This is the user's choice.

The beacon is essentially a permanent device identifier stored on the blockchain with the required context. Beacons can be used by manufacturers to communicate with equipment on site or by end users to choreograph business between devices. While the two devices on Idem can find each other even without the beacon on the blockchain, they need to know their device ids. The beacon enables users and devices to find each other without directly processing the IPFS device ID.

4.14 SuperSkynet Network, Nova

Nova is SSN's local intelligent contract platform that allows for the creation of infinitely scalable distributed applications with IoT and AI vertical.

At the start, Nova will be a lightning blockchain that can interoperate with Ethereum. This means that at genesis, Nova will start with EVM on SSN's Tendermint, supporting Web3 compatibility, sharding, and high throughput. Compared to Ethereum's current working proof consensus, Nova allows transactions to run at 20 times the speed of the transaction because it can package 20 times the transaction

One block. This will allow users who are familiar with the Ethereum smart contract platform to migrate to Nova and allow developers to be familiar with the network of SSN. Nova will accept light as its gas, similar to the way Ethereum accepts Ether. The validators on Fabric will also help run Nova's distributed application platform. Later, Nova will contain its own native virtual machine built on the Tendermint Core. The virtual machine (QVM), called Quantum, will be a lightweight JVM implementation for high performance when performing chain logic. More details on QVM will be updated in the near future.

4.14.1 Nova Scalability

Through a blockchain, Nova can handle 200 transactions per second. In SSN, you can create an unlimited number of distributed replication blockchains on SSN and work in parallel.

Through SSN's cross-blockchain communication protocol, one person can create five more Novas, enabling plat-form to have a maximum of 1,000 transactions per second. Multiply the number of iot chains by 5 and Nova can process 5,000 transactions per second. In this way, Nova implements both horizontal scalability and infinite sharding.

Nova also contains the necessary consensus for iot and ai interaction. Both the current bitcoin and Ethereum implementations have certain confirmation codes before the final transaction. For example, bitcoin's six confirmations were 60 minutes, while Ethereum's six confirmations were just two minutes. Using Nova's consensus model, blocks will be done in a second.

Nova makes transaction fees significantly lower than Ethereum, since transactions are done instantly and there is no backlog.

4.15 SuperSkynet Network, Singularity

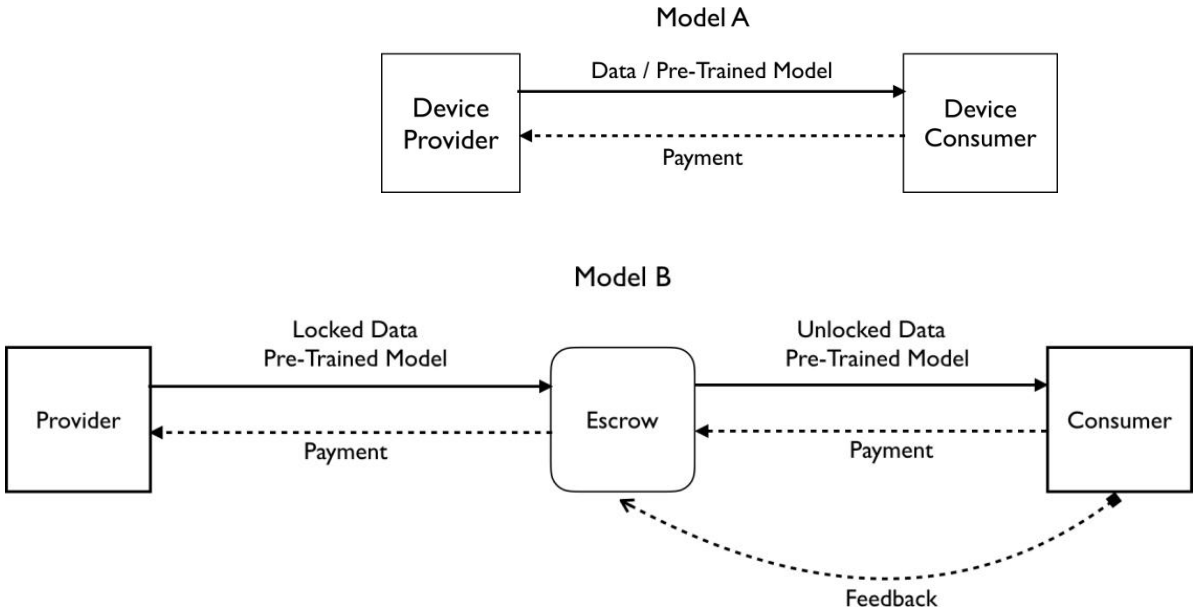
Both blockchain on SSN and distributed applications on Nova form so-called distributed knowledge networks or virtual application layers. Here, nodes with data and knowledge from places like ImageNet and the data they collect may be distributed across the network. With built-in identity protocols, nodes will be able to find each other in similar areas of knowledge and start transferring data, knowledge, and training to each other in a decentralized manner. You can create more distributed applications on the knowledge base and interoperate with existing applications. This virtual application layer will be an autonomous infrastructure that implements existing infrastructure, such as AWS, and distributes important data sets for training. In this way, nodes can enter the knowledge network ecosystem and begin a cyclical evolution process.

Figure 15: SSN Virtual Application Layer

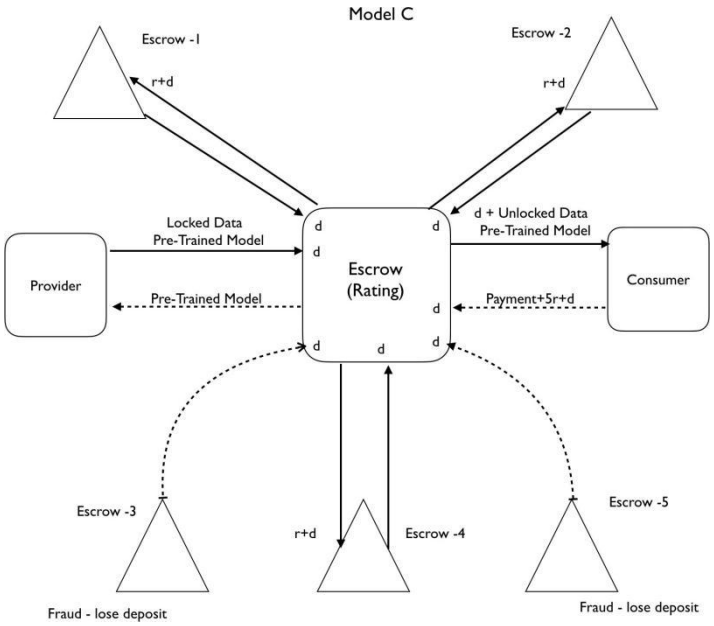
The figure above depicts how the virtual application layer would function. Billions of nodes will enter the SuperSkynet Ecosystem and be able to interact with one another with the KnowledgeNet virtual application layer. Companies and developers will be able to add to this ever-growing network by creating their own blockchains and DApps. Nodes on the network will be able to leverage these applications and settle prices with a built-in AI marketplace. The marketplace, distributed applications, and reputation economics will be released in more detail further along in the development of SuperSkynet Core as they need to be tailored to provide the necessary applications and make the device the core powers autonomous and fully functioning.

Onyx AI Marketplace In order to make Singularity's AI data market run smoothly in this distributed trustworthy circumstance, we have defined and implemented the smart contract systems enable devices to exchange data, pre-trained AI model, or anything of value in a transparent, conflict-

free way. Basically, these smart contracts are a series of computer programs that are stored on a Nova ledger/blockchain and specifies contractual terms, along with possessing the means to enforce those terms. These smart contracts would enable exchanges between devices and update the identity protocol.



For seller and buyer with very high reputation score, we can execute model A smart contract to speed up transactions which is very straight forward and highly efficient. Model B smart contract is for the compromise method for entities with middle level reputation and can be decided by buyer.



In between fully distrust entities, we use model C of our smart contract to bind all parties' responsibilities and obligations including escrows who will be the witness for all steps of execution of this smart contract. All entities need to deposit a small amount of coins and also will be rated (reputation score) after this smart contract is executed completely. All entities who has a positive behavior in the execution will get its deposit back and its reputation score will also be increased. On the contrary, the deposit will be lost and reputation score will be decreased. Escrows entities will also receive rewards in return as trustful witness. After each transaction is done, an update will be made to the decentralized identities of each participant.

Autonomous Decentralized Machine Learning By coupling various machine learning distributed

applications with the virtual application layer's identity network, edge nodes can start finding one another to begin fine-tuning their networks. With this system, machines will be able to transfer knowledge and work with one another. Some methods of doing so include

Transfer Learning - Nodes can use pre-trained models and retrain the final layers to have the neural network become more generalized for other situations.

Data Labeling - Machines or people can label data that other devices can train from.

Federated Learning - Edge nodes will be able to train off private, untapped data such as medical data and collaborate to make a better neural network model.

Some advantages or underlying systems that would enable these types of learning include

Distributed Storage - Datasets can be distributed across the network rather than be centralized on one server

Distributed Processing - Nodes on edge will be able to distribute idle processing power or borrow others.

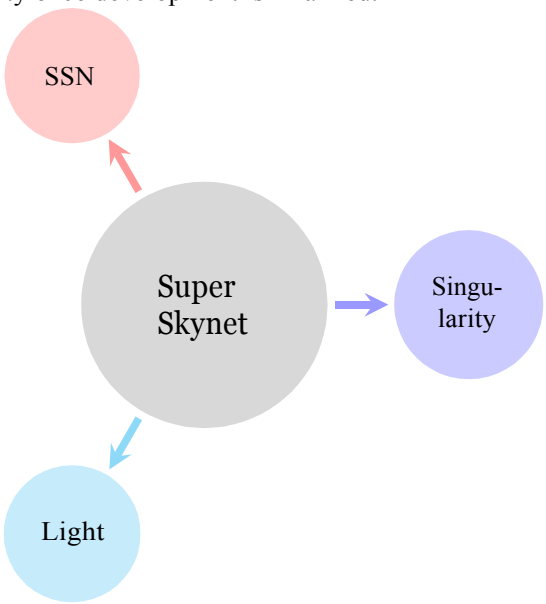
Incentives - Devices are incentivized to participate in this system, distribute data, sell idle processing power, and share algorithms

Knowledge - Edge nodes will be able to transfer knowledge from one node to the other.

The Singularity application layer will provide the necessary applications for devices to interact with one another. More specifically, the smart contracts that this layer contains will be for federated learning, data labeling, distributed computing, and transfer learning.

4.16 SuperSkynet Token

The SuperSkynet Token offered in the OpenSingularity token distribution will swap over to all cryptocurrencies on Singularity once development is finalized.



5 Conclusion

To sum up, SuperSkynet protocol can be summarized as SuperSkynet core and SuperSkynet Network. Superskynet core is a modular blockchain SoC core that offers SNM competitive options in the iot chipset market. All of these devices with the super skynet kernel will be equipped with SSN hardware wallets, enabling devices to use blockchain and cryptocurrencies, and have the security of classified wallets, but also adding brain chip systems for ai authentication and intelligent capabilities similar to humans.

All iot devices, from self-driving cars to smart cities and smartphones, can be connected via the SSN network, a scalable iot infinite chain platform. SSN will enable these devices to exchange value by deploying algorithms over the network, training vital private data, finding each other in a secure way, leveraging other networks like bitcoin or Ethereum, and borrowing its KnowledgeNet from improved basic infrastructure like AWS and Imagenet. This will be supported by the SSN's extensible fault-tolerant architecture, which enables the network to handle various iot subsystems by providing interoperability between private and public blockchains while providing the ability to process millions of transactions in a heartbeat.

With the supergrid core, devices have the ability to become intelligent and leverage blockchain networks. With the super skynet network, devices can connect and interact with each other as people do today.

Both components will support the creation of SuperSkynet, an end-to-end protocol that supports intelligent machine economics.