JARVIS Technical Whitepaper

Disclaimer

This is a conceptual document (**“White Paper”**) describing our proposed JARVIS platform and JAR tokens. It may be amended or replaced at any time. However, we are under no obligation to update this White Paper or to provide the recipient with access to any additional information. This White Paper is for discussion purposes only. Readers are notified as follows:

Not available to all persons: the JARVIS platform and JAR tokens are not available to all persons. Participation may be subject to a range of steps, including the need to provide certain information and documents.

No offer of regulated products in any jurisdiction: JAR tokens (as described in this White Paper) are not intended to constitute securities or any other regulated product in any jurisdiction. This White Paper does not constitute a prospectus nor offer document of any sort and is not intended to constitute an offer or solicitation of securities or any regulated product in any jurisdiction. This White Paper has not been reviewed by any regulatory authority in any jurisdiction.

No advice: this White Paper does not constitute advice in relation to whether you should participate in the JARVIS platform or buy any JAR tokens, nor should it be relied upon in connection with, any contract or purchasing decision.

No representations or warranties: No representations or warranties are made as to the accuracy or completeness of the information, statements, opinions or other matters described in this document or otherwise communicated in connection with the project. Without limitation, no representation or warranty is given as to the achievement or reasonableness of any forward-looking or conceptual statements. Nothing in this document is or should be relied upon as a promise or representation as to the future. To the fullest extent permitted under applicable law, all liability for any loss or damage whatsoever (whether foreseeable or not) arising from or in connection

with any person acting on this White Paper, or any aspect of it, notwithstanding any negligence, default or lack of care, is disclaimed. To the extent liability may be restricted but not fully disclaimed, it is restricted to the maximum extent permitted by applicable law.

English version prevails: this White Paper is provided in an official English version only. Any translation is for reference purposes only and is not certified by any person. If there is any inconsistency between a translation and the English version of this White Paper, the English version prevails.

Other companies: other than JARVIS.AI, the use of any company and/or platform names and trademarks does not imply any affiliation with, or endorsement by, any of those parties. References in this White Paper to specific companies and platforms are for illustrative purposes only.

You must take all necessary professional advice, including in relation to tax and accounting treatment. We hope the JARVIS.AI project will be highly successful. However, success is not guaranteed and digital assets and platforms involve risk. You must assess the risks and your ability to bear them.

Risk Disclosures

1. *Technology*

   Sophistication. Tokens are often described in exceedingly technical language; a comprehensive understanding of applied cryptography and computer science is required in order to appreciate inherent risks. By using the Services, you represent and warrant that you have sufficient knowledge, market sophistication, experience, and/or professional advice sufficient to undertake a prudent evaluation of the merits and risks of all transactions conducted by you pursuant to the Services. You agree to bear sole responsibility for the aforementioned evaluation.

   Forks. The blockchain technology underlying JAR tokens is subject to change at any time, including changes in operating rules (commonly referred to as “forks”), and blockchain networks may go offline as a result of bugs, hard forks, or a number of other unforeseeable reasons. Such changes may materially and adversely affect the value or function of the JAR tokens in your Account. You agree that you are fully responsible for monitoring such changes and agree to bear all risks arising therefrom or relating thereto.

Malicious Nodes. Some nodes in the JARVIS network may be malicious and attempt to get rewarded without corresponding contribution; also, attackers may try to ruin the JARVIS ecosystem if they only suffer from minimal penalties. We need strong guarantees to protect the network from malicious attacks to ensure that the transactions are secured and the ecosystem is sustainable. Some attacks that could threaten a blockchain network are listed and discussed as follows.

Sybil Attack. Malicious nodes could create multiple Sybil identities to strive for more rewards or cheat the network. In general, the proof mechanism should have established barriers to prevent Sybil attacks; however, there is no guarantee such barriers will always be successful.

Out-of-Work Attack. While an attacker can control a lot of nodes, the nodes could be used to make some troubles on a distributed computing network. The nodes controlled by malicious attackers could be called zombies. An attack methodology is to ask the zombie nodes quit or go on a strike at one time. On JARVIS network, the zombie nodes may take AI jobs but fail to complete them or return invalid results. If an AI job is assigned to a group of which most are zombie nodes, the AI job would receive unauthentic results or just simply fail.

Outsourcing Attack. Malicious nodes may outsource their jobs to other nodes, such that they may earn the rewards easily without consuming the corresponding computing power. On JARVIS network, nodes should present their capabilities to strive for taking jobs. Validation of node capabilities based on Proof-of-Intelligence may mitigate the behavior of outsourcing attack because the malicious nodes would lose their jobs if they do not endeavor to execute the same; however, there is no guarantee this approach will always be successful.

Disclaimer. JARVIS hereby disclaims all responsibility for any loss or damage arising from or relating to your use of any Services (including, but not limited to, risk of losses due to trading or due to factors beyond its control regarding the viability of any specific blockchain network). JARVIS further disclaims all responsibility for any loss or damages arising from or relating to any cyber-attacks (including without limitation the theft of your personal information), unprecedented surges in trading volume, any disruption or shut down of the Services, or other technical difficulties with respect to the Services.

2. *Security of the Platform.*
   You acknowledge that information you store or transfer through JARVIS' services may become irretrievably lost or corrupted or temporarily unavailable due to a variety of causes, including software failures, protocol changes by third party providers, internet outages, force majeure event or other disasters including third party DDOS attacks, scheduled or unscheduled maintenance, or other causes either within or outside JARVIS' control. You are solely responsible for backing up and maintaining duplicate copies of any information you store or transfer through JARVIS' services.
3. *Not Securities.*
   Use and purchase of the tokens generated by JARVIS carries significant financial risk. JARVIS hereby expressly disclaims that the transactions taking place on its platform pertain in any way to an offering of securities in any jurisdiction or that any documents published on its platform are solicitations for investment.
4. *Regulatory Measures*
   Crypto-tokens are being, or may be overseen by the regulatory authorities of various jurisdictions. JARVIS may receive queries, notices, warnings, requests, or rulings from one or more regulatory authorities from time to time, or may even be ordered to suspend or discontinue any action in connection with the Website or Services. The development of the Website may be seriously affected, hindered, or terminated as a result.
5. *Illiquidity and Price Volatility*
   You may find it difficult or impossible to liquidate. There may not be a demand for JAR tokens. JARVIS is not responsible for the circulation and trading of JAR tokens on the market. Tokens such as JAR tokens, if traded on markets, usually have extremely volatile prices. Fluctuations in price over short periods of time frequently occur, which price may be denominated in Bitcoin, Ether, US Dollars or any other fiat currency. Such fluctuations could result from market forces (including speculations), regulatory changes, technical innovations, availability of exchanges, and other objective factors and represent changes in the balance of supply and demand. The Seller is not responsible for any secondary market trading of JAR tokens, nor is JARVIS obliged to tame any price volatility of JAR tokens. Careful due diligence should be undertaken by you, with the full understanding that your contributions may not ultimately result in a useable or valuable token and the value of your contributions may therefore be subject to total loss.

JARVIS does not make any representation or warranty, explicit or implicit, as to the usability or the value of any tokens. You understand and accept that there is no warranty or assurance that you will receive any benefits through any JAR tokens that you hold.

6. *Compliance by Users.*
   You acknowledge and agree that JARVIS is not responsible for determining whether or which laws, rules, or regulations apply or may apply to your transactions (including, without limitation, any anti-money laundering laws, securities laws and tax laws). You acknowledge and agree that you are solely responsible for compliance with all such laws rules, or regulations as may be applicable to your transactions. Without limiting the foregoing, you acknowledge and agree that you are solely responsible for all tax obligations arising from your use of the Services. You further acknowledge and agree that JARVIS shall not be liable, whether directly or indirectly, for any of your tax obligations.

7. *Foundation Compliance.*
   You acknowledge and agree that JARVIS' recordkeeping and customer verification procedures may be, without prior notice, subject to change at any time as required by applicable regulations or state of the art practices.
   Applicable law, regulation, and executive orders may require JARVIS to, upon request by government agencies, freeze or suspend withdrawals or trading (or both), or disclose information regarding your Account(s). In the event such disclosure is compelled, you agree that JARVIS may disclose information regarding your Accounts. While JARVIS will endeavor to, where commercially reasonable, give you prior notice of such disclosure, JARVIS makes no guarantees that such prior notice will be made.

JARVIS

A Next-Generation Intelligent Contract and Decentralized AI Platform

Ever since Google built their first neural network to recognize cat faces in 2012 [1], artificial intelligence (AI) has become one of the hottest industry trend. Beyond the hype and the heightened media attention, numerous startups and internet giants race to build better and smarter products and services employing AI to build new user interfaces.

From face recognition to human-like voice synthesis, Deep Learning, a subset of machine learning in AI, beats all existing best-in-class machine learning algorithms in almost every field within a very short amount of time. As you may know, even the best human Go player can be defeated [2]by a few lines of codes, which was unthinkable just a few years ago.
There is no doubt that AI will lead the next wave of innovation. From fully autonomous cars to solving problems beyond human intelligence, AI will be ubiquitous in our life. We believe AI has the potential to spark social revolution and to push the human race forward for hundreds of years to come.

Yet, we still have a very long way to go to reach AI's full potential. As domain-specific problem becomes more challenging, the amount of data and computing power required for training grows exponentially. Most AI development we have today is made by large companies that are resourceful enough to afford the cost of computing power and the cost of data acquisition. In fact, companies like GAFA (Google, Apple, Facebook, Amazon) or BAT (Baidu, Alibaba, Tencent) spent hundreds of billions of dollars on powerful GPU supercomputers and hired hundreds of thousands human workers to create datasets. The demand of massive computation for AI even forced them to develop their own AI chip like TPU [3] so that they can one day rule the new AI economy. There is no question that large capital requirement for AI development prohibits small and medium size startups to participate, not to mention to compete against the big companies.

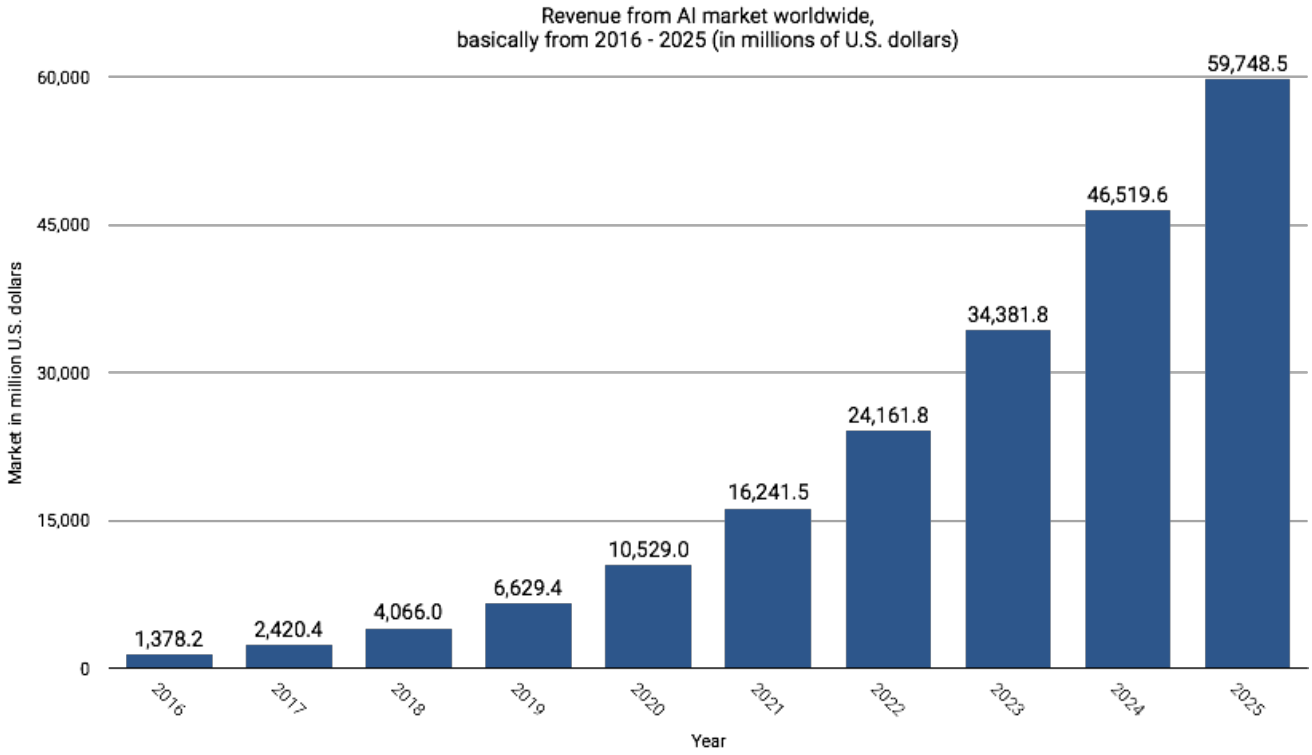
In addition to the capital requirement constraint, another big challenge for future AI development is that we still lack viable models for trustworthy and secure computing platform for AI that can one day run our lives and our world. In fact, as touted by some of the greatest minds of our age [4] [5], AI is probably the biggest existential risk for human civilization because the more data we can get, the smarter AI can become, which makes AI an obvious “winner takes all” economy, and whoever controls the platform can control our world, for good or for ill.

This is why we created JARVIS, the first decentralized AI platform backed by blockchain technology, to give AI power back to the people for a better future. With JARVIS, we can collectively explore and build decentralized cognitive intelligence and define boundaries and rules for artificial intelligence to make it ethical, moral and beneficial for the mankind. Also, as envisioned by tech giants, the future of AI will largely depend on new consumer goods with AI built-in, so JARVIS will provides turn-key software solutions so that people can easily create the next-generation Internet of Things (IoT) with cognitive intelligence that is both dependable and does not come at the cost of our [6].

PLEASE NOTE: CRYPTOGRAPHIC TOKENS REFERRED TO IN THIS WHITE PAPER REFER TO CRYPTOGRAPHIC TOKENS ON A LAUNCHED BLOCKCHAIN THAT ADOPTS THE JARVIS SOFTWARE. THEY DO NOT REFER TO THE ERC-20 COMPATIBLE TOKENS BEING DISTRIBUTED ON THE ETHEREUM BLOCKCHAIN IN CONNECTION WITH THE JARVIS TOKEN DISTRIBUTION.

Problems: A Broken Foundation for AI

From AI Application Idealization to Realization



The global AI market is growing fast. According to IDC[7], the global spending on cognitive and AI solutions will continue to see significant corporate investment over the next several years, achieving a compound annual growth rate (CAGR) of 54.4% through 2020 when revenues will be more than $46 billion, while Statista[8] reports similar prospect that the total AI market size in 2025 will be 24 times bigger than that in 2017.

Despite enormous amounts of investment, the AI systems we have today are still far from complete. These systems today can only be described as pattern recognition algorithms at best. Several barriers prevent us from making a quantum leap toward more general-purpose AI system that can one day free us from repetitive works and create general higher societal values.

As of today, even if we were to just create a smarter hardware gadget or an app to deliver minimal cognitive intelligence, it may still cost a fortune for most of us. The investment required from AI idealization to realization is still prohibitively high and thus most startups who want to pursuit this direction usually get stuck half way toward commercialization.

Machine learning, especially deep learning, requires massive amount of data and computation to get useful results in reasonable amount of time. To give you an example how expensive these cost could be, just for computation along, a single NVIDIA DGX-1 with 8 V100 GPU cards can cost you $149,000 USD, and renting one on cloud providers like Amazon will cost you more because their goal is to maximize their profit over hardware investment. Even if you use consumer-grade GeForce GPU cards instead enterprise-grade Tesla GPU, two training rigs each with 4 high-end NVIDIA GTX 1080Ti can easily cost more than $10,000 USD or more.
For data acquisition, although most people start building models with public dataset like ImageNet, which contains 14 millions of labeled images, Google recently releases the new state-of-art image classification result with whopping 300 millions of labeled images[9], which cost at least 90 million U.S. dollars to build on Amazon Mechanical Turk[10], and proved the effectiveness of enormous data for deep learning.
In either case, these barriers to entry inhibit new-comers from competing against monopoly and thus prevent the entire AI industry from truly flourishing.

Challenges for Building Smarter Decentralized Application

From the blockchain developer perspective, the existing blockchain systems have several major problems that make them impractical for mainstream use.

Problems like limited scalability, limited privacy, lack of formal verification, and inadequate development environment make platforms like Ethereum the worst places upon which to build decentralized applications. For example, when writing so-called “smart contract” in Solidity[11], major complications may arise due to inherent flaws in programming language and virtual machine design[12]. Design flaws in Solidity and vulnerabilities in Ethereum EVM has led to several major hacks such as DAO and Parity[13] in the short history of Ethereum.

In addition to these flaws and vulnerabilities, the Ethereum smart contract capabilities are very restricted due to its design goals of producing the exact same result everywhere. Despite being intended and proposed for financial applications, floating point is not supported, and there is no way to do overflow-checked arithmetic operations for integers. Complicating the matter further, there is also no way to interact with APIs and services on the public Internet in the isolated runtime environment, which renders smart contracts little more than electronic pocket

calculators (without floating point), with which it is almost impossible to conduct any meaningful work or to execute any serious computation on Ethereum with its consensus design.

Furthermore, today we have an enormous amount of GPU processing power wasted on useless number crunching for Bitcoin and Ethereum to create global consensus on the distributed ledger via proof-of-work (PoW). The energy consumption of Ethereum mining alone is close to that of the entire nation of Libya. Bitcoin's PoW is even more expensive. The amount of electricity wasted on global Bitcoin mining has exceeded the individual energy usages of 159 countries[14]. There are even prediction fears that given the current pace, Bitcoin will consume all the world's electricity by February 2020[15]. The PoW consensus scheme can produce a far great carbon footprint than imagined, worsening global warming and climate change. Such power should be put to better use.

AI and the Internet of Things (IOT): What's the Catch?

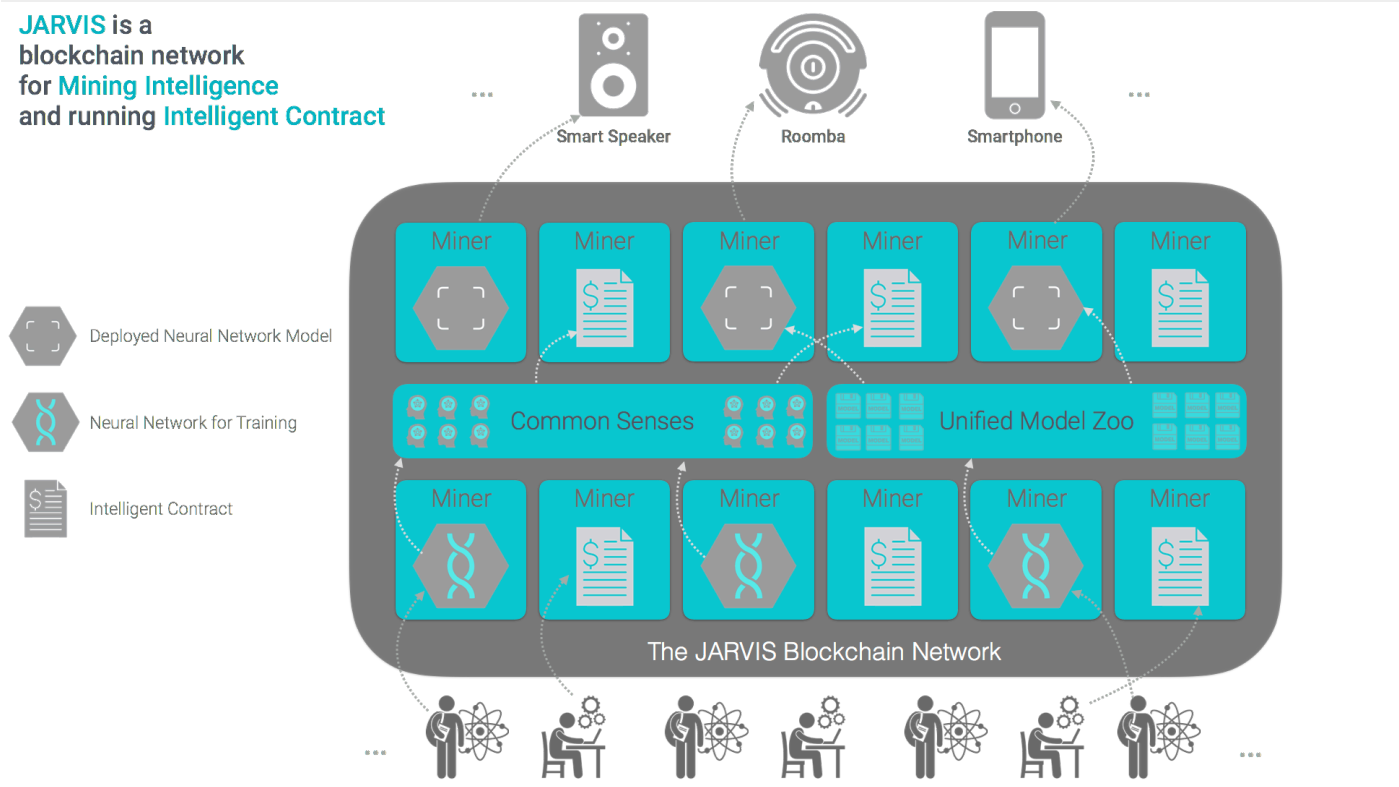
As for the AI industry, in Google's Founders' Letter April 2016, Larry and Sergey sent a message to the public: "Google is moving from a mobile first to AI first world" [16]. The message was made clear in Google I/O 2017[17] and in the October Google event, at which many hardware devices around Google's AI strategy were released[18], from smart speakers to maker-friendly smart camera kits. In fact, almost all industry tech giants today are now leveraging AI to reinvent the user interface for all kinds of gadgets, and there is a clear and considerable industry shift: the new role of AI is seeping into the Internet of Things (IoT).

However, there are two big problems ahead for smart IoT. First, it is extremely difficult to build AI-powered IoT because of its inherent complexity. Unless a team of experts for both hardware manufacturing and machine learning is available, and the cost of hardware development and the intelligence curation proves feasible, most smart IoT start-ups fail to resolve the inherent complexity and usually end up in death valley[19] or end up releasing a not-so-smart gadget[20].

Secondly, envision a future in which smart gadgets that are constantly collecting data and reacting to the environment are prevalent: our personal privacy will be greatly diminished. Such fears are not merely science fiction, but a current reality. For example, some central organizations are establishing mass surveillance of the entire population to build a "social-credit system," just like the one portrayed in the

“Nosedive” Black Mirror episode[21]. The omnipresent surveillance can easily lead to a totalitarian state power, reminiscent of Big Brother from George Orwell’s 1984[22].
For most people, it seems a conflict between technology and humanity is inevitable. To the contrary, these fundamental flaws in technologies are resolvable if they can be made more compatible with humanity. Based on the decentralized governance powered by blockchain technology, we believe we can build a more authentic future without the fear of losing our freedom to express and to be ourselves.

Solution: JARVIS - The Decentralized Cognitive Intelligence



Enter **JARVIS**, the world’s first decentralized AI blockchain platform that serves as the extended digital nervous system to power your cyber-physical world with cognitive intelligence. As a decentralized platform for cognitive intelligence, JARVIS is designed to curate intelligence collectively by creating a marketplace for machine learning developers, knowledge curators, app developers, and computation providers (a.k.a GPU miners) to work together while incentivizing all participants in a marketplace via a value exchange network powered by blockchain technology.

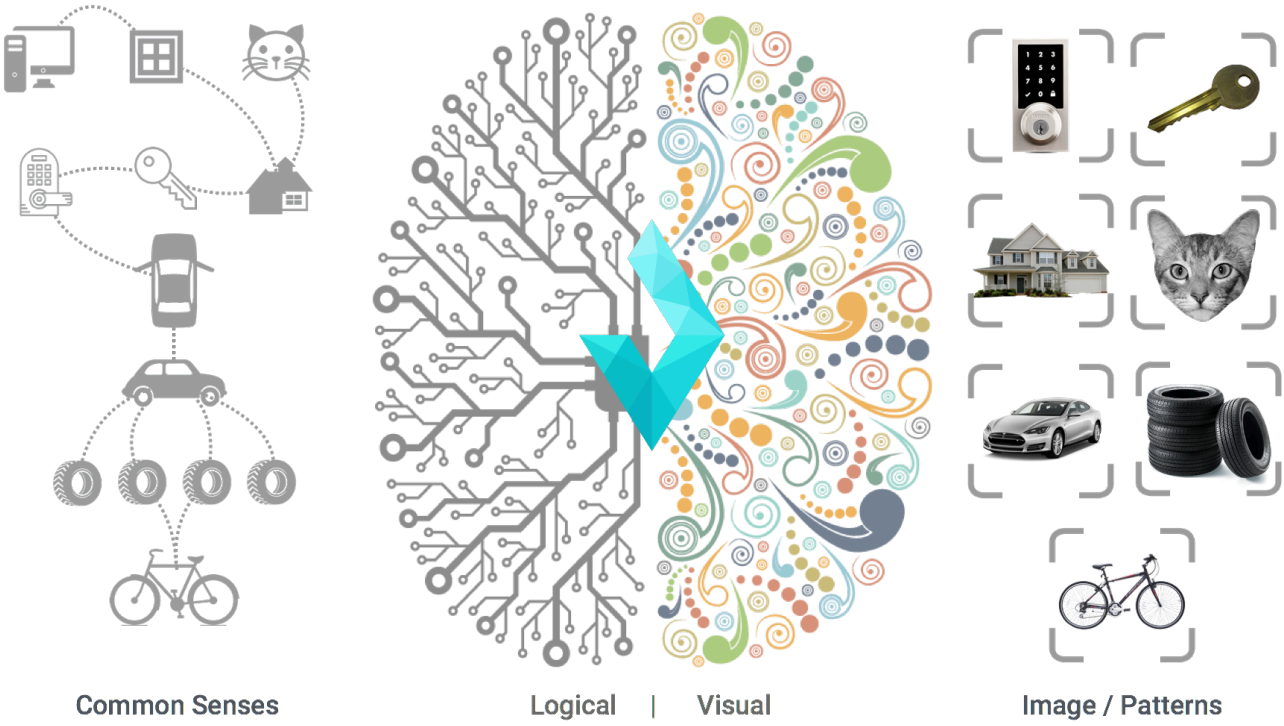
To elaborate, JARVIS will issue its own utility token called **JAR** to facilitate the value exchange process. For example, a machine learning developer can acquire GPU computing resources for an AI training job by spending JAR tokens in the network, and the network will randomly select suitable GPU miners to fulfill the job and receive JAR tokens in return. On the other hand, an application developer can spend JAR tokens and deploy a machine learning model trained by some other machine learning developer on the JARVIS platform and start serving its application to users without any prior knowledge about AI.

JARVIS will also provide several turn-key solutions and ready-to-use libraries for application developers to build not only smarter application, but also smarter hardware, such as a smart speaker with conversational interface, or a scene-recognizing vacuum cleaner robot to minimize the barrier of building AI-powered IoT and to broaden its application domain.

More importantly, we hope to collectively create ever smarter machines that may one day reach certain level of "*Artificial General Intelligence*" *(AGI)* to help us solve problems beyond human intelligence. Though AGI is still an open research problem, many researchers believe that two of the biggest obstacles are unsupervised pattern recognition and common sense reasoning for the world around us. So in addition to simply serving as a marketplace for computing resources and a platform for serving and training AI models, JARVIS will curate its own brain-like components such as the right brain and the left brain that are primarily responsible for visual and logical tasks accordingly, which translates into "**Unified Model Zoo**" and "**Common Sense Graph**" in JARVIS.

"Unified Model Zoo": The most comprehensive collection of neural network models for any pattern recognition task.

"Common Sense Graph": A knowledge graph that encapsulates basic human knowledge represented as a combination of linguistic corpus and fuzzy logic rules.



On JARVIS, two components that are inspired by human brain functions, i.e., the right brain and the left brain, are presented. One of the components, called **“Unified Model Zoo”**, is designed to recognize physical objects and to perceive contextual states. Unified Model Zoo would be the most comprehensive collection of neural network models for any pattern recognition task. It enables the JARVIS network to observe and explore the physical world while models of ever higher quality are trained and added to Unified Model Zoo by the JARVIS network based on the accumulated computing power.

The other component, called **“Common Sense Graph”**, is created for reasoning and understanding specific context. The conceptual ideas and relationships are extracted from the massive inputs to construct a knowledge graph to express the human thoughts. Common Sense Graph encapsulates basic human knowledge represented as a combination of linguistic corpus and fuzzy logic rules. With the two components, everyone can leverage the JARVIS network to create new applications more easily and also contribute more know-how to the network.

Based on these built-in components in the network, which are automatically curated over time via its **evolutional knowledge curation engine**, JARVIS further lowers the entry barrier for building intelligent systems and avoids redundant efforts being wasted on fundamental building blocks, such that cognitive intelligence may be built at a higher level. A machine learning developer would thus no longer need to train yet another human face detector or a cat face detector because JARVIS will have those basic neurons in place so you can directly connect your input to it and use the extracted features as inputs for some more complicated or domain-specific models, allowing more intelligence to be created in a hierarchical manner.

By leveraging its full capabilities, as a decentralized blockchain application platform, JARVIS makes smart contract intelligent, and we call it “**Intelligent Contract**.” An intelligent contract follows the same fundamental principles of the smart contract, such as being a public ledger that is traceable, transparent, and irreversible; however, an intelligent contract overcomes many limitations that smart contracts face. For instance, an intelligent contract allows application developer to create both deductive and inductive logics based on stateful context value, which can be used as a stated fact in the contextual memory of given contract.

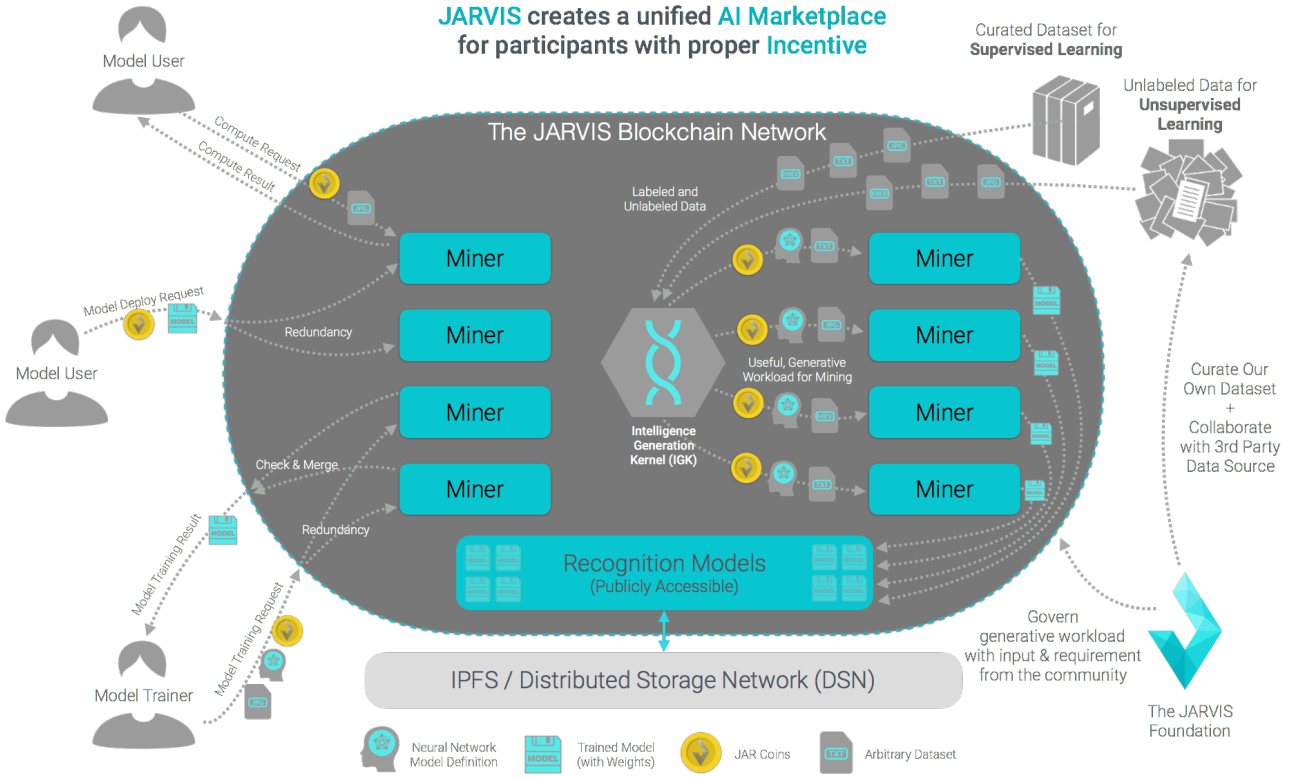
Simply put, an Intelligent Contract can react to the environment change which is updated by other contracts, and in every intelligent contract, developers can leverage various pattern recognition contracts deployed on the JARVIS network by asynchronous remote invocation. This forms a scalable distributed neural network on top of a trustless infrastructure backed by blockchain.

A side-by-side comparison between smart contracts and intelligent contracts is provided in the table below.

Attributes	Smart Contract	Intelligent Contract
Public Ledger	Y	Y
Completely deterministic	Y	Y
Ability to interact with services and data from outside of blockchain	N	Y
Ability to generate random number	N	Y
Support floating number	N	Y
Utilize Machine Learning workload	N	Y
Support deductive & inductive inference	N	Y
Easy to audit	N	Y

With intelligent contract, all kinds of truly smart applications are possible, ranging from smart crypto trading with AI-powered market sentiment analysis to scalable video analytic with adaptive feature learning. Unlike most PoW or PoS based blockchain today, JARVIS is designed to be infinitely scalable to handle arbitrary amount of AI computing workload, thus rendering it more dependable than most cloud providers and ultimately making it a preferred solution for deploying AI applications.

Value Exchange Network for AI



JARVIS is set to create an environment which AI-powered applications can be easily created and deployed, so in JARVIS, we developed a protocol to build an open marketplace for miners to rent out their computing power for machine learning developers on which to train and deploy their neural network models. The value exchange network is facilitated by running valuable AI jobs on the computing power of all participating GPU miners. Each AI job consists of a series of operations and an uncertain amount of data accessible from external sources such as *Distributed Storage Network (DSN)*, or *InterPlanetery File System (IPFS)* [23] or just via BitTorrent protocol.

Mining for Intelligence

Unlike traditional blockchains like Bitcoin or Ethereum, JARVIS decouples the concept of mining from the concept of building consensus. Traditional blockchains use a Proof-of-Work (PoW) based consensus design, which defines a way for miners to contribute its computation power to get rewards and also to reach consensus for the entire network. JARVIS uses existing consensus mechanisms with minimal computation (ex. Delegated Proof-of-Stake, a.k.a. DPoS or Verifiable Random Function, a.k.a. VRF), which will be determined later, for the ledger consensus while allowing spare GPU resources to get mining reward by running AI computation.

Thus, all miner nodes can either participate in the survey to earn transaction fees or contribute GPU processing power to earn computing fees instead. As a result, JARVIS doesn't rely on powerful GPU miners to rule the block creation process of the network thus free from majority hashrate attack. But a GPU miner node may run AI jobs to earn more reward than CPU miners.

To clarify, an AI job would be defined as either model training or model serving or any kind of combination of training and serving based on arbitrary intelligent contract execution. A model training job may include model design and refinement through automatic iteration of the model as well as the automatic back-propagation, which are built right in the JARVIS software stack.

On the other hand, a model serving job would be defined as deploying a trained models for either batch or on-demand real-time analysis onto arbitrary number of GPU miner nodes.

The Mining Proof

When an AI job is created and submitted to the JARVIS network, a group of miners will be selected via a deterministic random function from the consensus engine in the JARVIS network to solve it collectively. At the same time, a group of delegates will also be selected in similar fashion to verify the result and to ensure the result consistency.

Each AI job submitted will also be associated with certain amounts of JAR coins as miner fees. When an AI job is finished by participating miners and the job result is fully validated by the delegates, the AI job will be marked closed and the miner fees will be shared by participating miners and delegates based on their respective contributions.

Different AI jobs could have different complexity, difficulty, requirements, and also miner fees. For example, some algorithms could be advantageous to GPU miners who have large memory space but disadvantageous to CPU miners. Each AI job consists of sequential computing operations primarily and will always generate deterministic results which must be validated based on the concept of verifiable computing[24].

In JARVIS, the process of running and validating AI job on a decentralized network with trustless peers is called **Proof-of-Intelligence (PoI)**. We replace traditional Proof-of-Work (PoW) based system with PoI because obviously, PoI is more meaningful than PoW because the computing power is spent on more valuable works, i.e., mining intelligence.

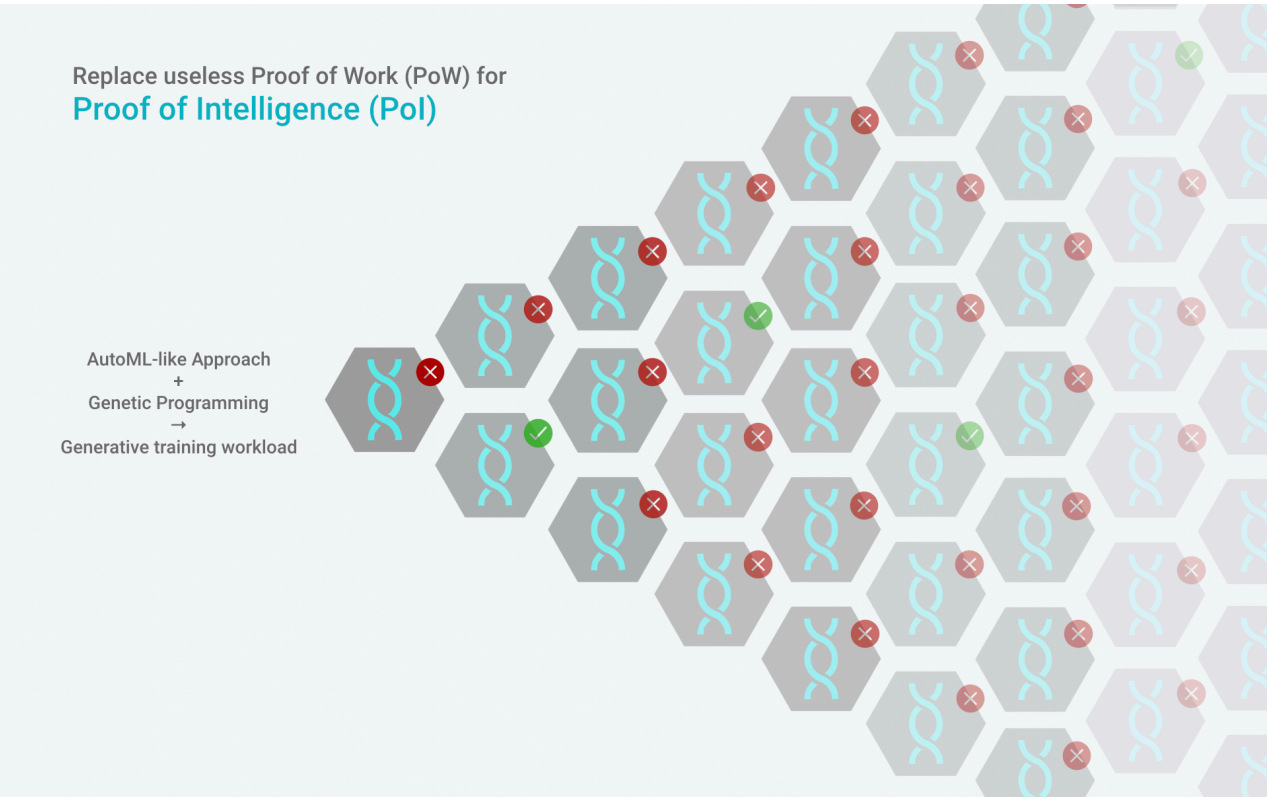
A massive number of miners could strive for AI jobs, especially for the AI jobs with respectable rewards. The JARVIS network will assign the AI jobs to miners on a random but deterministic basis. When miners are selected by the JARVIS network to solve an AI job, some nodes will also be selected randomly as delegates to verify the result of the AI job. As the delegates have reached a consensus that the result is valid, the AI job is completed.

In fact, these AI jobs can be submitted by the user of JARVIS network or can be generated by the network automatically. So, actually there are two types of PoI, “External” and “Internal”, that depend on the source of corresponding AI job of the PoI.

“**External PoI**” indicates an AI job is submitted and paid by external users in the JARVIS network. For example, the model trainer deposit JAR coins in the network in exchange for the miner’s computing power for those machine learning training jobs defined by the external user are external PoI. The result or trained intelligence of external PoI belongs to the external user and will not become part of the network.

In contrast, “**Internal PoI**” is automatically generated by the network itself as opposed to being issued by a specified user. The mechanism regarding Internal PoI is to create a large number of AI jobs that extract knowledge continuously and feedback the knowledge to the JARVIS network as public assets. Although no user would pay for an internal AI job, certain amount of JARs will be minted when AI job is completed by miners. In other words, JARs can be mined by running Internal PoI, and it is paid by the network itself.

Generative Knowledge Curation

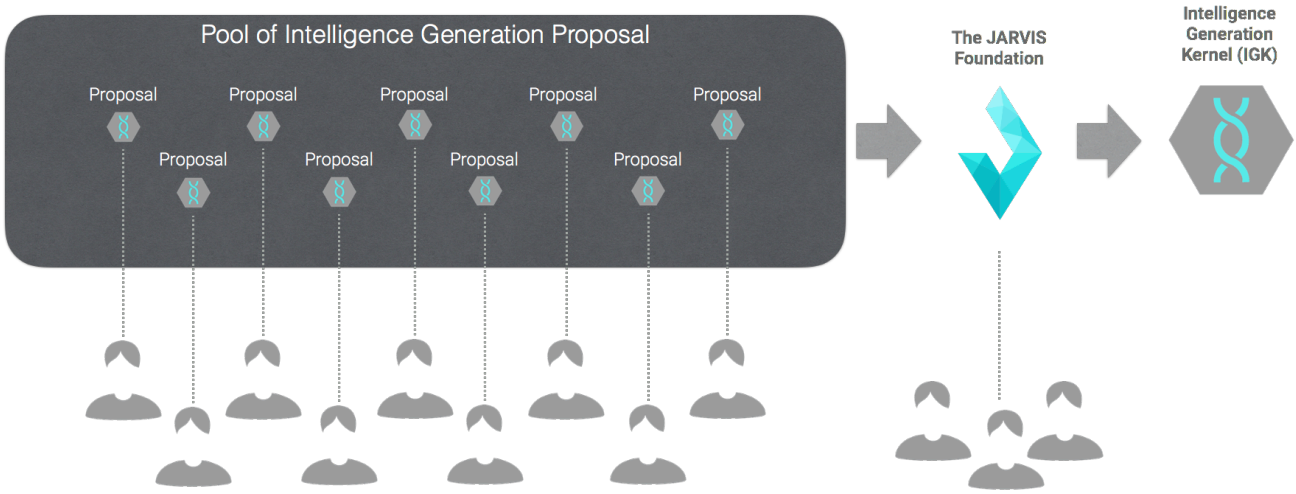


To ensure there is always enough workload for GPU miners to contribute their computing power to the JARVIS network, our protocol creates an unlimited supply of Internal PoI for generative knowledge curation. The generation of Internal PoI is controlled by a pre-defined algorithm, **Intelligence Generation Kernel (IGK)**, which can carry various proposals called IGK Proposal.

Each IGK proposal will specify a certain objective that can be reached by finding an optimal neural network model with the massive computing power embedded in the JARVIS network, e.g., spare GPU miners. The finding process would be based on automatic machine learning (AutoML) [25] [26] or deep neuroevolution[27] that automate algorithm design which generates different neural network model architecture continuously without human intervention to explore possible solution space.

Because the training process is often stochastic, the training parameters could be deterministically randomized repeatedly to produce different models. For example, the training parameters could be reset to initiate a new training process when the loss function or the objective function has crossed a certain threshold.

A great deal of Internal PoI could be produced with the AutoML-like mechanism as well as dynamic training parameter adjustment, but most of them may be discarded due to poor evaluation result in the end. The GPU miner will still get a mining reward nontheless because it has done its work.



Intelligence Generation Protocol

Although Internal PoI can be generated by IGK automatically, it still requires external resources for emerging subjects, curating training dataset, and collecting feedbacks from human. We defined the **Intelligence Generation Protocol** so anyone could submit **Intelligence Generation Proposals** to network and ask JARVIS Foundation to review it with a “proposal fee”, paid by JAR coins. JARVIS Foundation will then validate the given proposal by articulating its security, ethic, and legal compliances to decide if the proposal shall required further consideration.

After a certain period of time, we will have a number of proposals and survey the community for inputs on future course of intelligence generation. The process is just like the design of DPoS, where delegates can be elected with the same way, e.g., stake deposit, and those delegates need to review all proposals and provide their inputs. The selected proposals will be then applied and installed to IGK, so we can change the course of automatic knowledge curation process for better and safer AI governance.

Data Privacy

Homomorphic Encryption

Over the past few years, data collection of diverse domains has been growing explosively due to emerging technologies such as cloud computing, big data, IoT, and deep learning. Massive data has been aggregated to build many kinds of deep learning models in order to improve model accuracy for various AI applications. However a large portion of what we have today consists of sensitive information related to personal or organizational privacy. Much attention has been paid to issues of data privacy due to legal or ethical considerations, especially when the data is processed or stored on things like cloud services.

We believe the privacy issues should be treated even more carefully in blockchain environment because the computational hosts are more likely to be malicious or compromised in a trustless setup. Although the data could be encrypted during the transmissions between local and remote hosts to avoid being eavesdropped, the data must be plain, i.e., unencrypted, for computing in most applications. In other words, the remote hosts that perform the computing processes can still easily access the private data.

A concept of making computation on ciphertexts for ensuring privacy has been introduced in 1978 by Rivest et al. It is based on *homomorphism* which indicates the operations of two structures can be preserved with a map between the two structures.

For example, give two sets X, YX,Y and a map $\phi: X \rightarrow Y$φ:X→Y. For each

operation $\cdot$· of XX domain, an operation $*$∗ of YYdomain exists to

ensure $\phi(x_a \cdot x_b) = \phi(x_a) * \phi(x_b)$φ(xa·xb)=φ(xa)∗φ(xb) where $x_a, x_b \in X$xa,xb∈X.

For homomorphic encryption (HE), the computation result of ciphertexts can be decrypted and match the result of performing the corresponding operations on

plaintexts. That is, give encryption function $\phi_{enc}: X \rightarrow Y$φenc:X→Y and decryption

function $\phi_{dec}: Y \rightarrow X$φdec:Y→X where $\phi_{dec}(\phi_{enc}(x)) = x, \forall x \in X$φdec(φenc(x))=x,∀x∈X.

Then $\phi_{dec}(\phi_{enc}(x_a) * \phi_{enc}(x_b)) = x_a \cdot x_b$ φdec(φenc(xa)∗φenc(xb))=xa·xb where ∗∗ is the corresponding operation in Y for the operation $\cdot$ in X.

The purpose of homomorphic encryption is to compute data under encryption to ensure the private data will not be disclosed during the computation procedures. Many cryptosystems has been implemented for homomorphic encryption, but the operations cannot be applied arbitrarily and the operand would be also limited, e.g., integer only. Initially, these cryptosystems would not be appropriate for tuning AI models nor applying neural networks. However some studies have tried to implement deep learning with homomorphic encryption to keep data confidential in training or prediction processes. Based on a cryptosystem with homomorphic addition and multiplication operations, applying neural networks to encrypted data becomes possible[28].

Training over encrypted data is more difficult than prediction and may suffer from the requirement of having every operation to be a function of polynomial degree. A more recent study concluded that with approximating activation functions of neural networks, training could be performed with pleasing accuracy[29].

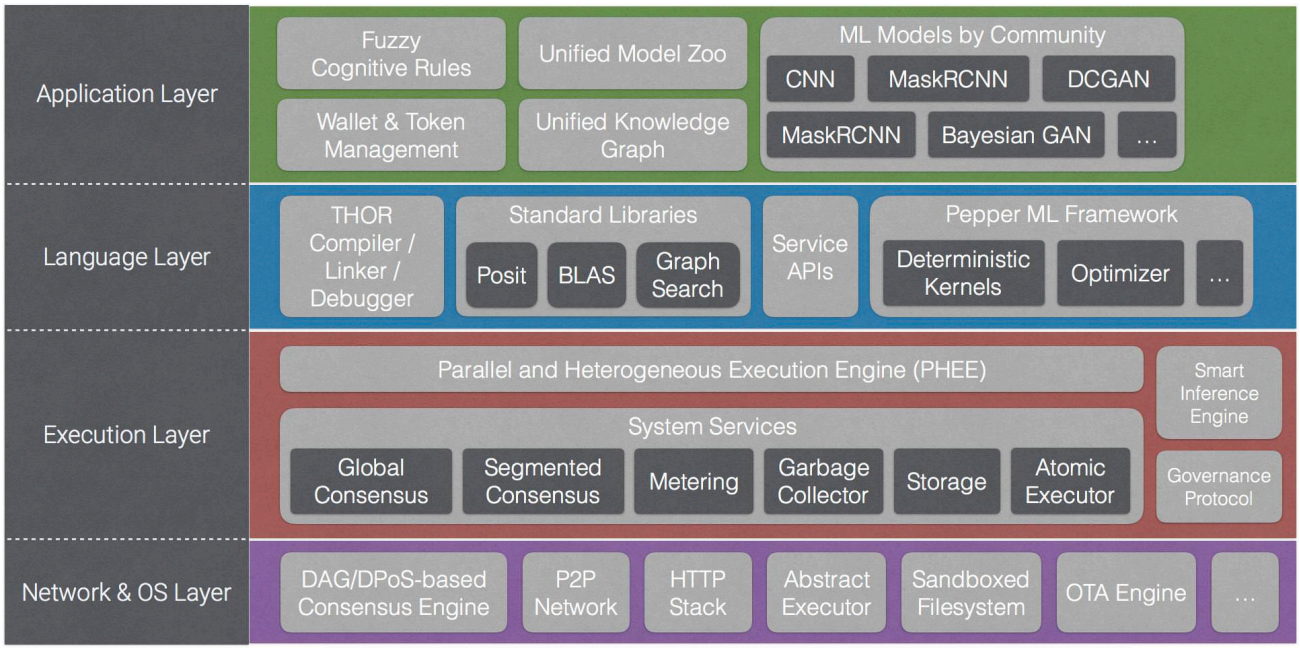
Although homomorphic encryption is still not really mature and efficient for complicated computations such as deep learning, there are several strategies to make it a reality. One of the promising direction is using FPGA as a starting point to accelerate homomorphic encryption directly[30] [31] at circuit level. Yet, in the same time, the global market has taken great interests in AI-specialized chips. Over the past few months, several AI chipmakers such as Graphcore[32], Cerebras[33], Cambricon[34], etc. have raised a large chunk of funding so more and more investments would be put into the area to implement AI properly. JARVIS is committed to kickstart an initiative to collaborate with the AI chip industry and startups to bring privacy-preserving AI chips to the market.

Of course, another approach would be inventing new homomorphic encryption scheme so that the cost of arithmetics over encrypted data is further reduced on conventional processors, and JARVIS is also committed to put our funding and resources into the active research field to accelerate the adoption of homomorphic encryption for deep learning so we can one day have things like secured and AI-assisted medical services[35].

Technology Overview

“Don’t be afraid to start over. It’s a brand new opportunity to rebuild what you truly want.” - Author Unknown

Due to the deficiencies in current blockchain technology stack, it is now time to rethink the fundamental technologies to iterate the future AI and blockchain economy with higher confidence for better result and faster speed. So we revisited the design of a decentralized application platform and redesigned almost every layer so it can better serve as a more general-purposed AI application platform on the blockchain.



The figure above shows the high level architecture and underlying components in JARVIS.

For ease of comprehension, the following sections will discuss each of these design concepts of JARVIS in a bottom-up manner separately

.

Multi-Consensus Design

Most of existing blockchain systems like Bitcoin or Ethereum use Proof-of-Work (PoW) based consensus to allow millions of completely untrusted participants to agree on single transaction history stored in a single distributed ledger. However the PoW based consensus is extremely inefficient and cost a lot of energy to reach agreement[36].

JARVIS takes a very different approach toward consensus. Instead of having a global ledger with PoW based consensus mechanism, JARVIS will adopt a new “multi-consensus” design which allows different consensus mechanisms to be used for different types of replicated state because we believe different type of shared information requires different level of consistency guarantee and transaction finality. For example, account balance state might be better stored in a single blockchain with near instant finality and with strong guarantee on consistency. We may either chose from Delegated Proof-of-Stake (DPoS) [37], Verifiable Random Function (VRF) [38] or even Directed-Acyclic-Graph (DAG) [39] [40] [41] based consensus to secure the account balance state on the trustless network.

On the other hand, as an analogy to the human brain, each of the computed neuron states across the entire cortex requires neither strong consistency nor near-instant finality as long as the collective intelligence and main consciousness persist. So the underlying network consensus mechanism[42] can be more flexible when weighing against consistency. So if we were building each individual intelligent agent in the form of intelligent contract running autonomously, the state of each agent does not require global consensus but only local consensus within a single shard. As a result, other DAG based consensus mechanism can be used here to improve scalability and transaction speed.

Although it is possible to make JARVIS work with PoW based consensus algorithm for both getting consensus and generating mining reward, our primary gaol is minimizing electricity waste, and we would like to see that mining power contributed to building cognitive intelligence that may one day benefit to most of us.

Odin VM: The Next-Generation Blockchain Virtual Machine

Back in 2014, when Ethereum first released the decentralized smart contract platform, such was hailed as a major advance in decentralization. However, difficulties later arose due to the inherited problems of the *Ethereum Virtual Machine (EVM)* design. In fact, it is actually considered virtually impossible to run most industry applications on Ethereum. The scalability and performance are simply unacceptable as a simple CryptoKitty[43] game can easily jam the entire network.

Though fully deterministic, the EVM is very limited in terms of computing capability. The lack of floating point, memory management, string support, standard library, and the fact that all data types cost 256 bits make Ethereum and Solidity by far the least productive programming environment we have ever encountered.

The trade-off between portability and capability is as undesirable as it can be. Obviously EVM is a broken foundation to run decentralized application, and there are many attempts to improve or to replace EVM such as *Qtum*[44], *EOS*[45] and *eWASM*[46]. We think if we were building the computing platform for the next 30 years or more, we'd better do the design right in the first place instead patching and forking and being complained all the time. That's why we created the next-generation blockchain virtual machine from ground up, and we call it **Odin Virtual Machine (OVM)** [47].

Just-in-time Compilation

OVM is a modern virtual machine supporting heterogeneous architecture which allows the same code to be run on both CPU, GPU, and even FPGA to maximize efficiency. Contract code are stored in an intermediate representation and translated into target machine code during runtime with built-in Just-In-Time (JIT) compilation based on LLVM[48] for CPU, NVVM[49] for NVIDIA GPU, and our own lowering mechanism for OpenCL and other FPGA accelerator. The virtual machine itself supports both task parallelism (lightweight threading) and data parallelism (data-driven reduction) in a single execution environment to exploit the processing power of modern many-core processors or highly-parallel processors like GPU.

OVM defines its own *Intermediate Representation (IR)* that allows compiler to transform high-level programming language construct into an universal representation of the contract code, making it possible to build multiple programming language front-ends on top for different domain specific applications. In addition, OVM will support existing EVM bytecode format such that existing EVM bytecode can be directly transpiled by the JIT sub-system. All existing Solidity contract can be executed on OVM seamlessly, minimizing the development efforts when switching platform.

Pluggable Consensus Engine

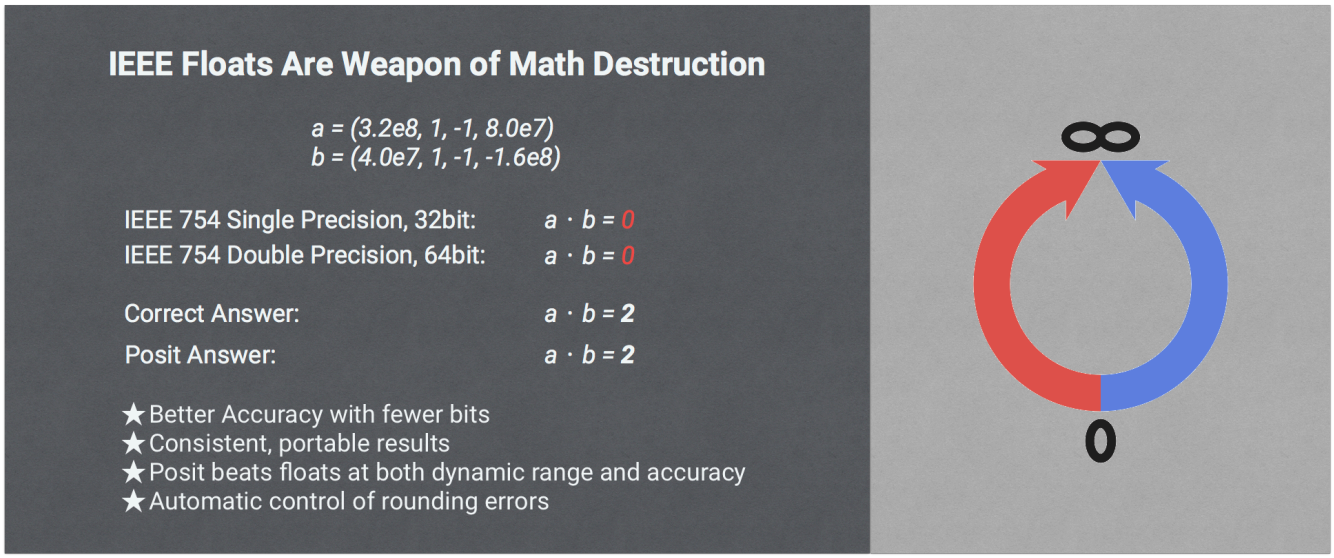
As mentioned above, JARVIS supports multiple consensus engines via OVM. By implementing interface for consensus provider, any consensus protocol can be used in contract execution as well as state replication. As an application developer, you can instruct OVM (via annotation of the programming language) to bind a state variable to a specific consensus mechanism in your code. It's possible to have multiple consensus protocols work together in a single contract execution process.

Initially, OVM will support both DPoS-based[37:1]and VRF-based[38:1]consensus protocol and add DAG-based protocol at a later stage when implementation became more matured and reliable and scalable.

Determinism

Despite being parallel and concurrent, OVM exposes fully deterministic runtime behavior for intelligent contract execution, so developer can test and debug their contract code with a peace of mind. Several techniques[50] [51] [52] would be used in the JIT compilation process as well as the runtime task graph scheduler to ensure consistent behavior across different hardware architectures. All state changes made during contract execution will be serialized and stored as transaction in the underlying consensus engine.

Floating-Point Arithmetic with Sigmoid Number



Due to the inherent nondeterministic behavior by IEEE 754 floating number design, floating point is not included in current Ethereum VM implementation. In fact, for the last few decades, IEEE754 standard is merely a guideline for designing Floating-Point Unit (FPU) in processor, so the result for any floating-point arithmetic actually varies from architecture to architecture.

However, as most machine learning today deals with probabilistic modeling and floating-point math, supporting deterministic floating-point arithmetic in blockchain is still paramount. To exploit the processing power of floating-point unit (FPU) on modern processors, OVM supports two types of float numbers: architecture-dependent float and architecture-independent float.

For architecture-dependent float, as defined by IEEE 754 standard, the JIT sub-system will try its best to maximize portability across different architectures, but it's not possible to produce same result everywhere every time. So the instance of OVM will be grouped by its runtime environment and processor compute capabilities (CM) as different shard in the JARVIS network. OVM ensures the same code, whether dealing with floating-point or not, always produce the same result within the same group. When some contract code were using architecture-dependent float, the JARVIS network will randomly choose OVM instances within the same group and compare the transaction results with each other instances to guarantee consistency. In some special case where the processor compute capabilities for floating-point is rare and different from the rest of participant in the network, OVM will use GNU GMP[53] instead to produce consistent result with lower performance instead.

On the other hand, OVM supports deterministic, architecture-independent floating-point arithmetic via compiler-assisted **"Posit Arithmetic"**[54] invented by *John L. Gustafson*[55], which is often also called **Posits**, **Gustafson Number** or **Sigmoid Number**.

Sigmoid is a math function[56] widely used in many neural network architectures, but it is relatively expensive to compute with today's IEEE 754 floating-point number. *Posits* makes sigmoid function computation extremely efficient and trivial and greatly improves the training and inference performance for machine learning while producing deterministic result with higher accuracy and dynamic range. In the same time, Posits requires fewer bits to achieve same precision and accuracy when comparing to traditional IEEE754 float, making future AI accelerator hardware architecture more energy efficient.

Although no Posits based FPU available on those off-the-shelf CPU or GPU today, OVM includes a software emulation for both CPU and GPU and also a VHDL library for FPGA hardware accelerators, making Posits the first-class citizen for blockchain floating-point arithmetic. In fact, there are several attempts made by companies like REX Computing[57] and others to implement Posits arithmetic for the next-generation processors, so we hope Odin VM and JARVIS can accelerate the vendor adoption of Posits in both software and hardware development and eventually fix broken floating number math that hampers the HPC community for decades.

Random Number

Random Number Generator (RNG) is very important for most programming, especially machine learning and many algorithmic trading, but in the same time very difficult to implement in a deterministic system. Several solutions were proposed[58] [59] [60] [61], but none of them are truly scalable nor free from attack.

To ensure the deterministic result involving random variable, the design of RNG is tightly-coupled with underlying consensus protocol design. For example, with DPoS consensus protocol, we can simply use last block hash as the random seed for current block generation. Or we can we leverages the *Verifiable Random Function (VRF)* [38:2] used in Dfinity[62]to generate deterministic random numbers without compromising network consistency.

Work Metering

Unlike EVM, OVM does not impose per-instruction metering during runtime but replace the concept of *Gas* in Ethereum with *Spark*, or *SPK* for short, in JARVIS.

SPK is an unit for measuring amount of processing work to be done on a baseline system. JARVIS will define the cost of a baseline system running some pre-defined AI workload or intelligent contract in number of *SPK*. For example, the cost for a mining rig with *NVIDIA GTX 1080 Ti* processing AI workload for 1 second can be defined as *1 SPK*. If miner has a different mining rig with lower spec, such as *NVIDIA GTX 1070*, it won't be able to earn exact *1 SPK* for every second but less than *1 SPK*, depending on the actual performance of the mining system. The network will check system performance periodically by monitoring the time spent on certain AI tasks, compare it against other nodes in the network and adjust the reward setting accordingly.

Similar to Ethereum, when some contract code or some AI job is submitted to the network for processing, the sender needs to deposit certain amount of JARs, which will be automatically converted into SPKs in real-time based on market price, and later consumed by the OVM during execution. The remaining balance will be automatically refunded to the sender's account upon the completion of the job. If the deposit is depleted before completing the job, all state transaction will be reverted and the fund will not be returned.

Deterministic Tensor Programming Support

Since OVM was created to facilitate machine learning and artificial intelligence, we included tensor[63] as one of the fundamental primitives in OVM IR. Inspired by MXNet TVM[64] and Tensorflow XLA[65], OVM provides a new set of tensor IR stack with strong guarantee on determinism. The OVM JIT sub-system leverage LLVM IR optimizer to perform target-dependent optimization for the given computation graph and use Halide[66] to compose high-performance kernel on various target such as ARM v7/NEON, CUDA, and OpenCL.

By incorporating tensor programming into OVM IR, OVM makes adding language support for symbolic tensor programming[67], which is the common practice for deep learning, straightforward. The symbolic tensor programming framework that allows users to declare mathematical expressions symbolically, as computational graphs. You may invent a totally different way to declare a computational graph in your own domain specific language (DSL), but OVM aims to be supporting all kinds of DSL front-end, such as Tensorlang[68], to better serve machine learning developer and programming language inventor.

Logic Programming Support

In addition to tensor, which serves as the primitive for various deep neural networks architectures, we believe in additional to deep learning, it's time to bring back logic programming and symbolic reasoning at the dawn of AI renaissance. OVM IR supports dynamic symbols and logic construction to make logic programming language like *Prolog*[69] compatible with OVM.

Logic programming might seem daunting in the first place because it requires programmer to get the logic right up front, but once people get used to it, it's a valuable tool to immensely improve their programming skill to produce logically correct codes. Since correctness is paramount in any contract code, logic programming can get away with a lot of errors in your overall design when it comes to logical consistency. More interestingly, those high-level rules in logic programming can be used to construct AI system with reasoning capability or build decentralized business process management for e-commerce applications.

OVM also has inference engine built-in to support runtime inference based on dynamic symbols. The inference engine is highly parallel and efficient, allowing millions of rules to be defined and evaluated on parallel processors like GPU.

Execution Domain and Internet Access

Previously, smart contract running on top of EVM is limited in a sandbox that prohibits any communication to the outside world. Developer used to have a “Oracle” running off-chain, side-by-side with the smart contract to inject data from outside of the blockchain. The “Oracle” has to run periodically with a continuous source of funding to pay the gas because there’s no way to call the “Oracle” from the smart contract. This complicates the design of blockchain application and prevents blockchain application from accessing external data in general, making it much less useful.

To minimize the hurdle, JARVIS provides a set of trustworthy nodes who are capable of accessing Internet and serve as a bridge between blockchain and external network. In other word, JARVIS defines a number of **Execution Domains** where contract codes are executed with different runtime capabilities and restrictions. Function calls can be sent across different domains just like conventional remote procedure call (RPC). Internally OVM will create different system threads for different execution domain and also create proxies for remote domains. The contract code can be annotated to select the desired execution domain programmatically.

THOR: The Intelligent Contract Programming Language

To ease the development of decentralized AI application on JARVIS, we’ve designed a new programming language named THOR from ground up. THOR is a contract-oriented, high-level, multi-paradigm programming language. The language itself is heavily influenced by *C++*, *Julia*[70], *Go*[71], and *ECMAScript*[72].

THOR is a static-typed language with automatic type inference and metaprogramming support via template programming and static functions that run during compile-time. THOR is designed to eases the development of intelligent contract by providing a simple, clean, and intuitive development environment for any developer who wants to create intelligence of any scale in the JARVIS network.

THOR comes with a compiler to compile THOR source code into OVM IR, which will be later compiled and linked into actual machine code on different processor architecture during run-time via multiple compiler framework like LLVM[48:1], NVVM[49:1], and Hadile[66:1].

More details about the THOR language design will be elaborated in our technical yellow paper, but just to give you sense about how it might look like, here's an example intelligent contract for opening the cat door when cat is recognized:

```
package main;

import ml.dl.model.cnn;

var catModel;
function init() {
    self := state.create<CatState>("dummy");
    state.current<CatState>(self);
    myModel = cnn.deploy("ipfs://my-cat-model");
}

@consensus { scope = "global" }
class CatState {
   var hasKnownCat: bool;
}

function catFound() {
    return state.current<CatState>().hasKnownCat;
}

function openCatDoor() {
    console.print("open door");
}

@exported
function checkCat(var snapshot: Image) {
    classes := catModel.classify(snapshot)
    if(classes["known-cat"]) {
        self := state.current<CatState>();
        self.hasKnownCat = true; // triggering rule
    }
}

rule CatDoorState : catFound => openCatDoor;
```

SIF ML: The Scalable & Encrypted Machine Learning Framework

As of today, TensorFlow[73] is the de-facto standard when it comes to deep learning. However it's designed to be a toolset limited to the domain of deep learning and deep learning only addresses a small portion of the all machine learning problems in artificial intelligence landscape. When it comes to other discipline of machine learning or artificial intelligence in general, developers need to incorporate other libraries or frameworks like libSVM[74], OpenCV[75], Kaldi[76], LightGBM[77], or CoreNLP[78]. There're a plethora of tools for machine learning, and each of them has different strength and weakness, making choosing machine learning framework a complicated task to begin with.

SIF aims to create an initiative for developers to unify all efforts into one single framework with multiple sub-modules, each of which is specialized in different discipline of machine learning, so the scope of SIF is beyond deep learning. All sub-modules will share the same foundation for linear algebra, probabilistic modeling, and applied mathematics libraries to make sure all numeric results are stable and deterministic.

For deep learning in SIF, a meta-learning framework will be included for automatic model exploration and training based on evolution computing. Sub-modules that are based on AutoML would make it much easier to fine-tune the hyperparameters of a neural network with massive computing power. Diverse genetic algorithms regarding deep learning, e.g., loss functions, objective functions, and optimization algorithms, would also be appended to SIF to accelerate the model training based on evolution. Moreover, the process of model training is extremely stochastic. Sometimes the success of a model depends on the fortune. The randomness of model training in SIF would be managed for accelerating evolution based on distributed computing. A process may have several instances with different randomness, e.g., the different random number seeds, and would be executed on distinct computing clusters at the same time.

More details about the SIF framework design will be discussed in our technical yellow paper due later.

Turn-key Solution for AI-powered IoT

Recently, companies like Google[79], ARM[80], Samsung[81] and several others bigshots in the industry all started to call the future of hardware is artificial intelligence unanimously. We shared the same belief that connected device is as dumb as the offline devices if there's no intelligence attached to it. To put it another way, one of the key to unleash the full potential of AI is to make AI abundant and ubiquitous in our everyday life, so we can immerse ourselves with smart hardwares in our surroundings as the new user interface to interact with Internet services and apps.

So JARVIS will provide turn-key solutions for smart IP camera, smart speaker, and also other low-power sensors to accelerate the adoption of JARVIS. With those turn-key solutions and ready-to-use firmware images, the barrier of creating intelligent machine is minimized, thus boost the application and transaction on the JARVIS network.

For example, you can build a smart camera to recognize people's face, emotion, behavior by simply directing the video stream to corresponding pattern recognition services that is deployed with existing models in the Unified Model Zoo without ever considering how to train such model or how to create a real-time video analytic infrastructure. Or you can virtually wire microphone input the speaker output declared in the intelligent contract to create a conversational user interface that works right out-of-box.

The entire IoT software stack is actually based on years of research and development for some AI-powered IoT gadget (to be announced later) and will be completely open-sourced under non-restrictive open source license when it's ready later this year.

In addition, as we have very strong connection to the hardware industry and IoT companies in Taiwan, we have reached collaboration agreements with multiple hardware gadget companies to incorporate JARVIS IoT software stack with their future hardware products, and we welcome any type of partnership here to really take IoT to the next level at global scale.

Potential Application

Smart Home and Video Analytic

Fueled by promises of voice recognition technology and AI, smart home is one of the fastest growing market in recent years. Yet, we are far from reaching the full potential of a true smart home. Many smart home devices are simply not “smart” enough in terms of its intended capabilities, significantly reducing the value proposition, hence their adoptions.

As a decentralized marketplace, JARVIS encourages a more vibrant development and adoption of AI applications specifically to power the smart home devices. In addition to voice and pattern recognition, we expect video analytic to be a key value driver for many upcoming home devices, from home security to tracking movement of toddlers and pets.

When the technology behind connected devices is finally able to support the intended vision of seamless smart home, the ecosystem will flourish, brining security, convenience, entertainment, and many other benefits to people’s home.

Medical Image Analysis

Diagnostic radiologists take 10 years of intense training and stringent certification to qualify, from medical school to residency and fellowship. Being able to discern subtle differences in a medical image between a healthy patient and an abnormal one takes knowledge and experience. Unfortunately, missed diagnosis in cancer and bone fracture happens on average 15 out of 1000 procedures[82], and 7% of radiologist faces claims each year[83]. This leads to patients not receiving the most adequate care, while doctors and hospitals become liable for malpractice.

Based on deep learning with thousands of historic patient images and readings, an AI application learns to distinguish subtle image differences from healthy and abnormal patients, and the AI application highlights areas of medical image at risk of cancer or bone fracture. Assisted by enhanced images, radiologists could make faster and more accurate readings with confidence. Since patient records can be fully obfuscated via homomorphic encryption, patient privacy is completely protected.

Every party involved benefits from JARVIS. Patients benefit the most from accurate early detection of any abnormalities, so they could receive best treatments possible.

Doctors become more efficient while avoiding high cost of malpractice. Hospitals and insurances reduce cost with healthier patients and more effective diagnostics.

Next-Gen User Interface

We believe the ultimate realization of AI is best presented as a revolutionary user interface. Although computing power has been growing substantially, the performance of human machine interaction does not have remarkable enhancement. The inefficient user interface is always a big issue for the productivity of developing novel applications; moreover, the user experience is constrained. That is also why many companies show enthusiasm for developing chatbots and smart speakers to extend their services while AI has shown great potential in user interaction.

Our interaction with technologies continues to evolve as they become more powerful and become more integrated into different aspects of our lives. Naturally, innovation around User Interfaces (UI) is a critical driver to unlock the values of future technology applications. It indicates not only more kinds of interfaces will be enabled but also the efficiency can be substantially enhanced because the machine will understand human much more.

Tabbing into JARVIS' Unified Model Zoo and Common Sense Graph, the next generation UI may achieve just that. Imagine a future where all UIs become conversational, intuitive, or with sense of sight. It would not be constrained by traditional I/O devices, e.g., keyboard, mouse, flat screen, speaker, etc. Any component of a cyber-physical-system such as various sensors could be used to make information transmission between human and AI-powered machine. Therefore, the next generation UI would make UI designers' job easier and enable UIs to be modified according to changes in usage scenarios automatically no matter what hardware would be involved. Once input is received by a JARVIS-powered application, more extended knowledge and logtics would be involved to achieve intelligent reactions which makes more intuitive friendly user experience.

Turn-key Solution for Internet-of-Things

Internet-of-Things (IoT) consists a wide range of interrelated products and services, ranging from customer facing devices and sensors, to virtual objects, platforms, and networks. The key for IOT providers to deliver higher value is their ability to make sense of the data collected through the IOT devices. However, due to challenges such as resource constraint, lack of expertise, and duplication of work, it is counter-productive, and often unrealistic, for all the IoT providers to attempt to develop an in-house back-end system that is smart enough or powerful enough to do the necessary analytical work. Many IoT providers also face a dilemma which they need to gather enough data to train the analytical model (to provide better services); yet without the analytical model already in place, the IoT products are not attractive enough to be widely adopted (i.e. unable to gather enough data).

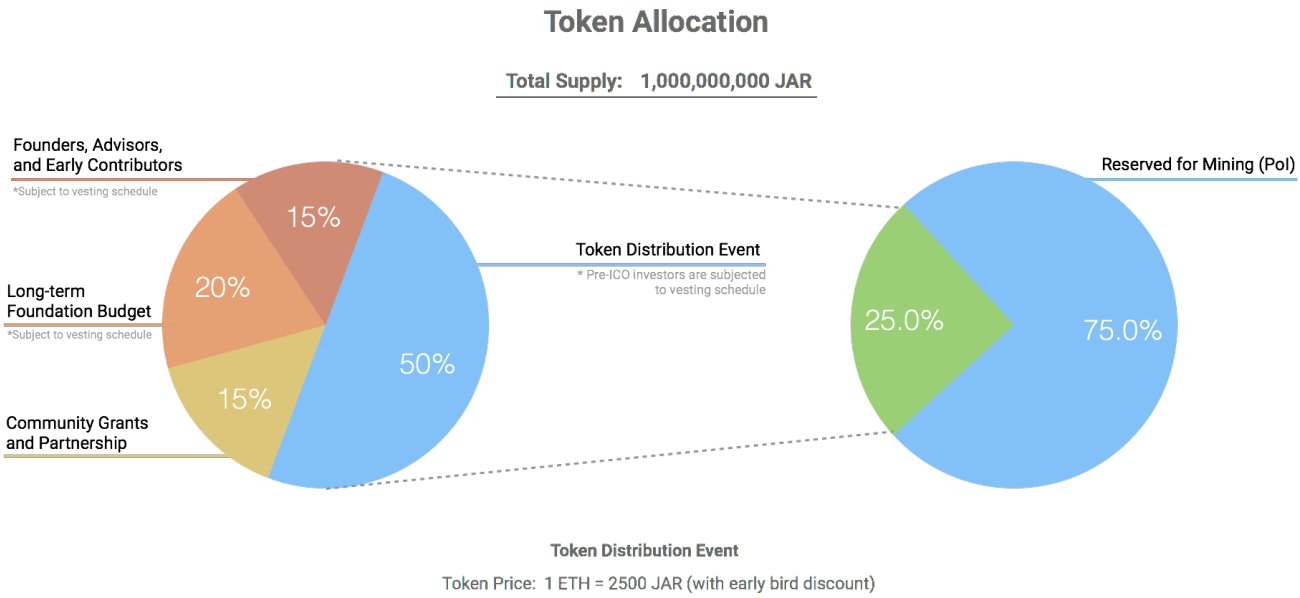
JARVIS hosts abundant trained models that are readily available for the IoT providers to adopt as turn-key solution. As the result, many IoT products may realize their full potential from day one by integrating with the JARVIS software stack, turning Internet-of-Things into Intelligence-of-Things in the end.

Token Economy

As previously mentioned, JARVIS is introducing a new utility token: **JAR**. With JAR, JARVIS is able to establish a set of protocols that help define how participants on JARVIS will create, distribute, and consume AI models and curated knowledges. Simply put, JAR is imperative to JARVIS' sustainability, helping the platform to truly become the decentralized AI marketplace that benefits all.

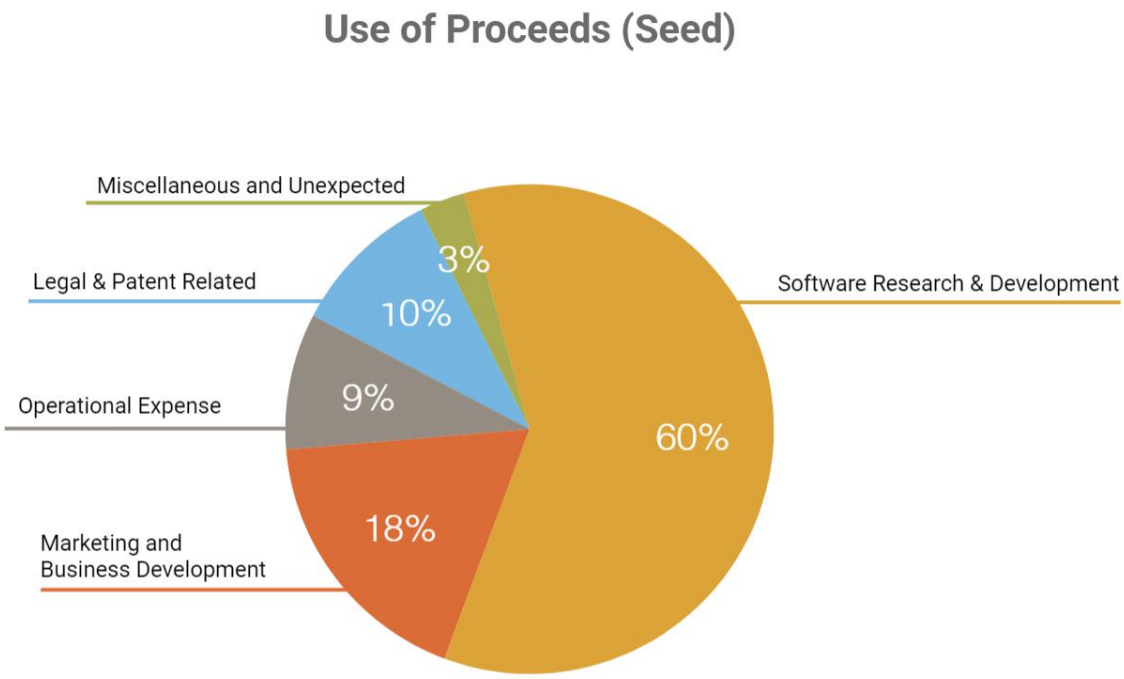
Token Allocation

There is a finite number of JAR tokens capped at 1,000,000,000. While 25% of JAR will be distributed within the first year to facilitate the launch and development of JARVIS platform, the remaining 75% will be reserve for mining over the next 50 years. Our plan for token allocation, use of proceed from the initial token distribution event, and the token emission rate over the next 50 years are explained below.



Token Allocation Summary

- Mining reserved: 750,000,000 JAR (75% of total JAR)
- Token Distribution Event: 125,000,000 JAR (12.5% of total JAR)
- Foundation Budget: 50,000,000 JAR (5% of total JAR)
- Community grant and partnership: 37,500,000 (3.75% of total JAR)
- Founder, Advisor, and Early Contributor Reserved: 37,500,000 (3.75% of total JAR)



Finite Supply with Pre-defined Algorithm

The emission plan for the reserved JAR token is shown in the chart below. This rate of emission is suitable not only for the healthy growth of supply and demand of AI services within JARVIS platform, but also for synchronizing JARVIS' growth timeline with the expected technology breakthrough and adoption timeline across the AI industry.

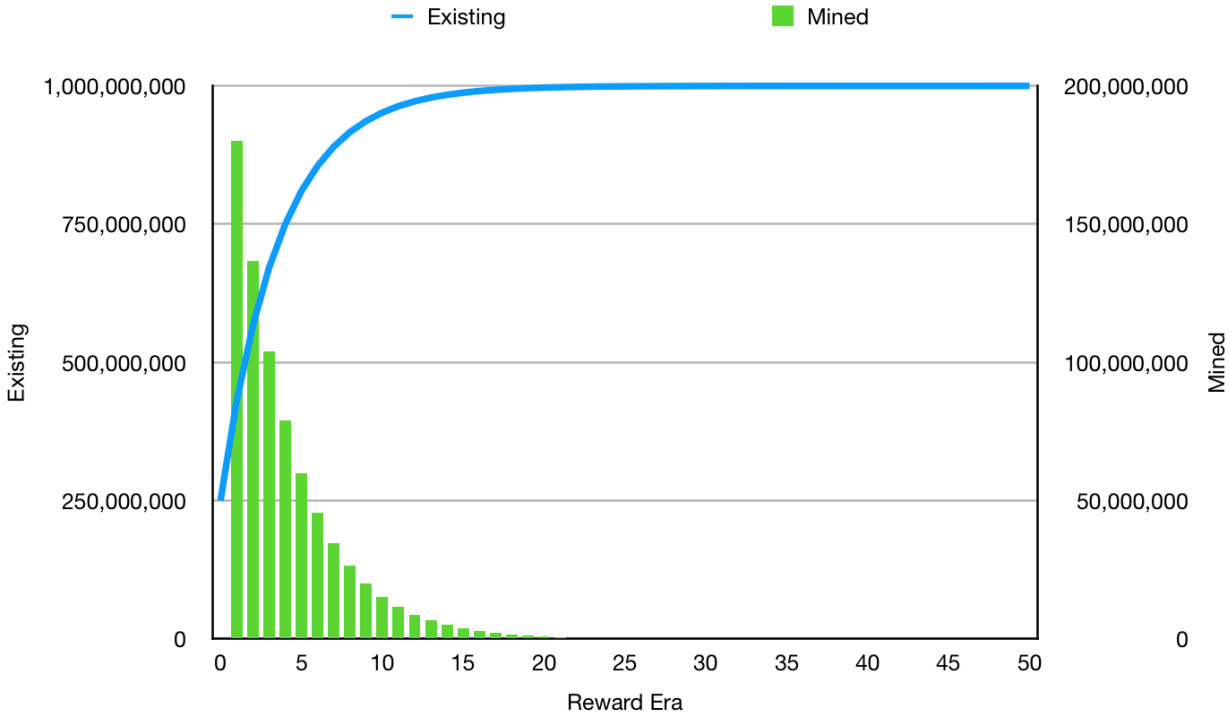
The JAR generation algorithm defines how JAR will be created. Each new JAR will be validated by the JARVIS network to ensure the generation rule is not violated and the generation rate would be also limited. On JARVIS, we would like to have a finite supply of JAR, say JJ, with an initial supply $s \cdot J$s·J. The mining efficiency of JAR supply is set to decrease geometrically, i.e., $a_x = r \cdot a_{x-1}$ax=r·ax−1 where xx indicates a reward era and $a_0 \cdot J$a0·Jindicates the supply that could be mined in the first era. It implies the JAR generation rate every era is a geometric progression. Moreover, the length of an era would be adjusted to achieve one year approximately. The common ratio rr is defined as: $r = \frac{1-s-a_0}{1-s}$

Therefore, the existing supply at the end of the x^{th}xth era can be presented as:

$$f(x) = (s + a_0 \times \frac{1-r^x}{1-r}) \cdot J$$

Obviously, the JAR supply generation will converge at zero which indicates the total supply is finite. In other words, the number of JAR in existence will not exceed JJ.

Give $J=10^9$J=109, $s=0.25$s=0.25, $a_0=0.18$a0=0.18, the existing supply and supply increase can be shown as the following table.



Reward Era	Existing JAR	JAR Added	JAR Increase	JAR % of Limit
0	250000000.0000	-	-	25.000000%
1	430000000.0000	180000000.0000	72.00000%	43.00000%
2	566800000.0000	136800000.0000	31.81395%	56.68000%
3	670768000.0000	103968000.0000	18.34298%	67.07680%
4	749783680.0000	79015680.0000	11.77988%	74.97837%
5	809835596.8000	60051916.8000	8.00923%	80.98356%
6	855475053.5680	45639456.7680	5.63564%	85.54751%
7	890161040.7117	34685987.1437	4.05459%	89.01610%
8	916522390.9409	26361350.2292	2.96141%	91.65224%
9	936557017.1151	20034626.1742	2.18594%	93.65570%
10	951783333.0075	15226315.8924	1.62578%	95.17833%
11	963355333.0857	11572000.0782	1.21582%	96.33553%
12	972150053.1451	8794720.0594	0.91293%	97.21501%
13	978834040.3903	6683987.2452	0.68755%	97.88340%
14	983913870.6966	5079830.3063	0.51897%	98.39139%
15	987774541.7294	3860671.0328	0.39238%	98.77745%
16	990708651.7144	2934109.9849	0.29704%	99.07087%
17	992938575.3029	2229923.5886	0.22508%	99.29386%
18	994633317.2302	1694741.9273	0.17068%	99.46333%
19	995921321.0950	1288003.8647	0.12950%	99.59213%
20	996900204.0322	978882.9372	0.09829%	99.69002%
21	997644155.0645	743951.0323	0.07463%	99.76442%
22	998209557.8490	565402.7845	0.05667%	99.82096%
23	998639263.9652	429706.1162	0.04305%	99.86393%
24	998965840.6136	326576.6483	0.03270%	99.89658%
25	999214038.8663	248198.2527	0.02485%	99.92140%

Reward Era	Existing JAR	JAR Added	JAR Increase	JAR % of Limit
26	999402669.5384	188630.6721	0.01888%	99.94027%
27	999546028.8492	143359.3108	0.01434%	99.95460%
28	999654981.9254	108953.0762	0.01090%	99.96550%
29	999737786.2633	82804.3379	0.00828%	99.97378%
30	999800717.5601	62931.2968	0.00629%	99.98007%
31	999848545.3457	47827.7856	0.00478%	99.98485%
32	999884894.4627	36349.1170	0.00364%	99.98849%
33	999912519.7917	27625.3289	0.00276%	99.99125%
34	999933515.0417	20995.2500	0.00210%	99.99335%
35	999949471.4317	15956.3900	0.00160%	99.99495%
36	999961598.2881	12126.8564	0.00121%	99.99616%
37	999970814.6989	9216.4109	0.00092%	99.99708%
38	999977819.1712	7004.4723	0.00070%	99.99778%
39	999983142.5701	5323.3989	0.00053%	99.99831%
40	999987188.3533	4045.7832	0.00040%	99.99872%
41	999990263.1485	3074.7952	0.00031%	99.99903%
42	999992599.9929	2336.8444	0.00023%	99.99926%
43	999994375.9946	1776.0017	0.00018%	99.99944%
44	999995725.7559	1349.7613	0.00013%	99.99957%
45	999996751.5745	1025.8186	0.00010%	99.99968%
46	999997531.1966	779.6221	0.00008%	99.99975%
47	999998123.7094	592.5128	0.00006%	99.99981%
48	999998574.0192	450.3097	0.00005%	99.99986%
49	999998916.2546	342.2354	0.00003%	99.99989%
50	999999176.3535	260.0989	0.00003%	99.99992%

JAR Is Utility Token

JAR is a utility token to facilitate the value exchange for AI on the blockchain. JAR can be earned by providing utility, such as:

- Rent GPU miner's computing resource for executing Intelligent Contract
- Participate in **Proof-of-Intelligence**
- Participate consensus protocol through underlying stake-based consensus mechanism
- Participate in community survey
- Contribute data or knowledge though campaigns organized by JARVIS foundation

JAR can be spent for consuming utilities or services provided by the network, such as:

- Train a complicated deep learning model for a specific classification problem
- Deploy a trained model on some GPU miner to provide recognition services via gRPC API serving
- Use a recognition service deployed by other people
- Submit **Intelligence Generation Proposal** through **Intelligence Generation Protocol**
- Execute a **Smart Contract** or an **Intelligent Contract** on the JARVIS network

Fees

A small portion of transaction fee will be collected to facilitate transactions on JARVIS. These fees are rewards to consensus providers and also to the JARVIS Foundation for the long-term support of the network. By applying Pluggable Consensus Engine, different consensus approach will have different transaction fee policy.

Governance

The JARVIS Foundation

The JARVIS Foundation is a non-profit organization that supports the development and operations of JARVIS. The foundation's mission is to support JARVIS to become the decentralized AI blockchain platform that serves as the extended digital nervous system to power your cyber-physical world with cognitive intelligence, creating technological breakthrough and generate tremendous societal value to the world.

Artificial General Intelligence Committee

The Artificial General Intelligence Committee is resided under the JARVIS Foundation with multiple purposes:

- Facilitate Intelligence Generation Proposal process as well as community survey. More on this is described in the below section
- Collaborate and communicate with other organizations on the topic of ethical AI
- Implement precautions and guard AI safety throughout JARVIS marketplace

Intelligence Generation Proposal (IGP)

Overtime, JARVIS community may need to make changes to the platform, such as updating the consensus protocol or deciding on IGK algorithms, as it grows. All JAR token holders, including JARVIS Foundation through the Artificial General Intelligence Committee, may submit proposal for changes. To submit proposal, token holders will pay a fee using JAR. A portion of this fee is non-refundable and is used for processing the proposal (e.g. review and investigate the suggested change). The remaining portion of the fee may be refunded; but only in the event that the proposal is adopted. The purpose is to encourage the submission of thoughtful proposals and prevent attack.

All token holders may join in the decision making process for approval the proposal by participating in community surveys. Once again, in order to encourage thoughtfulness on making decisions, token holders will pay a deposit (stake) to participate in the community survey. Such deposits will be locked for a period of three months before returning to the stakeholders. JARVIS employs the liquid democracy model. In other words, stakeholders may defer to subject matter experts to serve as delegate inside the community to provide feedback on proposals.

Roadmap

Midgard Release - 2018 July

- Intelligent Contract with THOR
- Functional Odin Virtual Machine
- Distributed Ledger Integration
- Implement External Proof-of-Intelligence
- Basic Model Training (CNN, LSTM)
- Smart Back-Propagation

Asgard Release - 2018 Oct

- THOR Language Enhancement
- More Model Support (RCNN, RNN)
- Stability Tuning
- Network Performance Enhancement
- Model Zoo Registry Zoo Open
- Implement Intelligence Generation Proposal Protocol
- Implement Internal Proof-of-Intelligence based on Genetic Programming

Yggdrasil Release - 2019

- Common Sense Graph Release
- Distributed Fuzzy Graph Search in THOR
- Fuzzy Inference Engine

Conclusion

JARVIS's mission is to empower the AI revolution that will soon bring the true smartness to everyone everywhere. It is expected that AI power will become ubiquitous and enrich our daily life just like how electricity, digital computing, and Internet lead to disruptive innovations that shaped our world we live in today.

To incorporate heterogeneous processor like GPUs or future AI chips in a decentralized, trustless AI computing cloud, a novel virtual machine over blockchain and a productive language have been designed with supporting distinct consensus protocols and deterministic instructions across different platforms.

With JARVIS' novel protocol design and economy design, both intangible and tangible assets such as human knowledge and computational devices could be aggregated to push AI forward collectively at a global scale. Any devices can easily access the accumulated knowledge and recognition capabilities on JARVIS to obtain cognitive intelligence on a reciprocal basis under an efficient economic system.

As power tends to corrupt and absolute power corrupts absolutely and AI is one of the most disruptive power ever exists in human history, we really need to take special care of it by incorporating AI on a decentralized infrastructure so everyone is a stakeholder, and no single organization owns it all.

1. Google Blog: Using large-scale brain simulations for machine learning and A.I. ↩
2. The Verge: Google's AlphaGo AI defeats world Go number one Ke Jie ↩
3. The Verge: Google's next-generation AI training system is monstrously fast ↩
4. CNBC: Stephen Hawking says A.I. could be 'worst event in the history of our civilization'↩
5. NPR: Elon Musk Warns Governors: Artificial Intelligence Poses 'Existential Risk' ↩
6. Elon Mush's Billion-Dollar Crusade to Stop The A.I. Apocalypse ↩
7. IDC: Worldwide Spending on Cognitive and Artificial Intelligence Systems Forecast to Reach $12.5 Billion This Year ↩
8. Revenues from the artificial intelligence (AI) market worldwide, from 2016 to 2025 (in million U.S. dollars) ↩
9. WIRED: AI and Enormous Data Could Make Tech Giants Harder to Topple ↩
10. Amazon Mechanical Turk: help for building your Machine Learning datasets ↩
11. The Solidity Contract-Oriented Programming Language ↩
12. HackerNews: Underhanded Solidity Coding Contest ↩
13. Turing Incompleteness and the Sad State of Solidity ↩
14. ExtremeTech: Bitcoin Mining Now Uses More Power Than 159 Countries ↩
15. Bitcoin Mining Guzzles Energy—And Its Carbon Footprint Just Keeps Growing ↩
16. Google Blog: This year's Founders' Letter ↩
17. VentureBeat: Google shifts from mobile-first to AI-first world ↩
18. Bloomberg: Google CEO Sets 'AI-First' Device Blueprint to Catch Apple ↩
19. TechCrunch Disrupt: CatFi a revolutionary pet tracking device ↩
20. 17 ridiculous 'smart' gadgets that really exist ↩
21. Wikipedia: Black Mirror: Nosedive ↩
22. Big Brother (Nineteen Eighty-Four) ↩
23. IPFS: InterPlanetery File System ↩
24. Wikipedia: Verifiable Computing ↩
25. Using Machine Learning to Explore Neural Network Architecture ↩
26. AutoML for large scale image classification and object detection ↩
27. Deep Neuroevolution: Genetic Algorithms Are a Competitive Alternative for Training Deep Neural Networks for Reinforcement Learning ↩

28. CryptoNets Applying Neural Networks to Encrypted Data with High Throughput and Accuracy ↩
29. CryptoDL: Deep Neural Networks over Encrypted Data ↩
30. Designing an FPGA-Accelerated Homomorphic Encryption Co-Processor ↩
31. An FPGA co-processor implementation of Homomorphic Encryption ↩
32. Graphcore ↩
33. Cerebras Systems ↩
34. 寒武纪科技 ↩
35. SecureMed: Secure Medical Computation using GPU-Accelerated Homomorphic Encryption Scheme ↩
36. Consensus in the Age of Blockchains ↩
37. Ethereum Wiki: Proof of Stake FAQ ↩ ↩
38. DFINITY: How to Achieve Near-Instant Finality in Public Blockchains using a VRF
    block-less cryptocurrency ↩ ↩ ↩
39. IOTA: The Whitepaper ↩
40. Byteball: The Whitepaper ↩
41. T.E.T.O: Trustless Eventual Total Ordering ↩
42. Raiblock: The Whitepaper ↩
43. CryptoKitties: Collect and Breed Digital Cats ↩
44. QTUM: The Blockchain Made Ready for Business ↩
45. EOS.IO: Dawn 2.0 ↩
46. Ethereum flavored WebAssembly ↩
47. Github: Odin VM by JARVIS ↩
48. The LLVM Compiler Infrastructure ↩ ↩
49. NVVM: NVIDIA Virtual Machine ↩ ↩
50. GPUDet: a deterministic GPU architecture ↩
51. CoreDet: a compiler and runtime system for deterministic multithreaded execution ↩
52. Parallel programming must be deterministic by default ↩
53. The GNU Multiple Precision Arithmetic Library ↩
54. Posit Arithmetic ↩
55. Beyond Floating Point: Next-Generation Computer Arithmetic ↩
56. Sigmoid Function ↩
57. REX Computing ↩

58. RANDAO: A DAO working as RNG of Ethereum ↩
59. Generate pseudo random numbers inside the Ethereum blockchain ↩
60. Ethereum PRNG ↩
61. RANDAO++ ↩
62. DFINITY - The future of cloud computing ↩
63. Wikipedia: Tensor ↩
64. TVM: Tensor Virtual Machine ↩
65. Tensoflow XLA ↩
66. Halide ↩ ↩
67. Wikipedia: Symbolic Programming ↩
68. Tensorlang ↩
69. Robust, mature, free. Prolog for the real world ↩
70. The Julia Language ↩
71. The Go Programming Language ↩
72. Standard ECMA-262: ECMAScript® 2017 Language Specification ↩
73. TensorFlow: An Open Source Software Library for Numerical Computation Using Data Flow Graphs ↩
74. LIBSVM: A Library for Support Vector Machines ↩
75. OpenCV: Open Source Computer Vision Library ↩
76. Kaldi: a toolkit for speech recognition ↩
77. LightGBM: Light Gradient Boosting Machine ↩
78. Stanford CoreNLP – Natural Language Software ↩
79. Google's new push to AI-powered services ↩
80. ARM Announces Updated AI and Deep Learning Framework for IoT Hardware ↩
81. Samsung Delivers Vision for Open and Intelligent IoT Experiences to Simplify Everyday Life ↩
82. Jeremy S. et al, "The Causes of Medical Malpractice Suits against Radiologists in the United States" ↩
83. Anupam B. et al, "Malpractice Risk According to Physician Specialty ↩