SingularityNET: A decentralized, open market and inter-network for AIs

Ben Goertzel, Simone Giacomelli, David Hanson,
Cassio Pennachin, Marco Argentieri
and the SingularityNET team

November 16, 2017

Abstract

The value and power of Artificial Intelligence is growing dramatically every year, and will soon dominate the internet – and the economy as a whole. However, AI tools today are fragmented by a closed development environment; most are developed by one company to perform one task, and there is no way to plug two tools together. SingularityNET aims to become the key protocol for networking AI and machine learning tools to form a coordinated Artificial General Intelligence.

SingularityNET is an open-source protocol and collection of smart contracts for a decentralized market of coordinated AI services. Within this framework, the benefits of AI become a global commons infrastructure for the benefit of all; anyone can access AI tech or become a stakeholder in its development. Anyone can add an AI/machine learning service to SingularityNET for use by the network, and receive network payment tokens in exchange.

SingularityNET is backed by the SingularityNET Foundation, which operates on a belief that the benefits of AI should not be dominated by any small set of powerful institutions, but shared by all. A key goal of SingularityNET is to ensure the technology is benevolent according to human standards, and the network is designed to incentivize and reward beneficial players.

Contents

Figure 1: Hanson humanoid robots, created by SingularityNET partner Hanson Robotics, will be early targets for embedded implementation of SingularityNET Agents.

1 Vision

1.1 Inspiration

The concept of a Technological Singularity is increasingly widely accepted throughout the technology and business worlds. More and more, it is realized that within the next few decades there will be a transition to a new society and economy in which machine intelligence is the dominant factor; and novel digital and organic technologies acting on multiple scales will network together to produce emergent "global brain" dynamics of unprecedented complexity and sophistication [Bro97] [Kur06] [Vin93] [Goe07].

Humanity faces many challenges on the path to a positive Singularity; among these is the contemporary global economic system. In numerous respects, today's standard financial mechanisms and institutions are not up to the task of serving as the economic engine of a smooth transition to a broadly positive Singularity. New, more flexible, open and rapidly adaptive economic structures and dynamics are needed [GGG16].

Blockchain provides a powerful tool for managing transactions in a Singularity-era economy [CB14] ; but blockchain is just a tool, and it must be used in the right way. A blockchain-based framework designed to serve the needs of AI Agents as they interact with each other and with external customers can enable the emergence of a collective intelligence. And it is critical that this framework be designed with positive principles in mind:

- Democratic governance – putting the community in charge of the system will tend to make the system act for the benefit of the community;
- Encouraging innovative new Agents to enter the network, and creating the conditions for Agents to act in a manner that feeds the collective intelligence;

- Directing a significant percentage of the network's efforts toward causes of broad benefit.

SingularityNET has been designed to meet these requirements, via

- Delivering intelligence services to corporations, individuals and organizations;
- Fostering the emergence of increasingly powerful distributed general intelligence;
- Deploying artificial intelligence for ever-increasing benefit of as many humans and other sentient beings as possible.

SingularityNET is explicitly designed both to be highly valuable in the current context, and to lay the groundwork for the emergence of a future self-modifying, decentralized "artificial cognitive organism" with the eventual potential for general intelligence and beneficial ethical characteristics beyond the human level. It is a practical design inspired by long theoretical thinking and prototyping on the part of the founders regarding concepts such as Artificial General Intelligence [Goe16a], Open-Ended Intelligence [WV16] and the Global Brain [Hey07].

1.2 Acute Market Needs Addressed

SingularityNET meets an acute and accelerating market need. In the current economic and technological context, every business needs AI, but off-the-shelf AIs will rarely match a business's needs. Only tech giants can hire armies of developers to build custom AIs, and even they have a hard time hiring enough AI experts to meet demand. SingularityNET provides an automated process enabling each business to connect existing AI tools together to build the solution it needs. By providing an easy means of configuring tools, it offers both customization and availability, while reducing the reduplication of effort involved in proprietary development, making the development process more efficient.

Many state of the art AI tools exist only in GitHub repositories created by grad students or independent researchers. This puts them out of the reach of anyone without the skills to install, configure, and run them. Most AI developers are academics, not businesspeople, and have no easily-accessible marketplace to monetize their clever AI code.

In addition to their clever code, machine learning tools require datasets of sufficient size. Creating and managing such large datasets is beyond the means and capabilities of most AI developers, and the closed development model that currently prevails makes it hard for developers to share datasets.

SingularityNET launches these AI tools and datasets onto the marketplace, making them more accessible to end-users and developers, and giving developers a way to monetize their creations. It is a sharing-economy marketplace for AI, that encourages collaboration between these tools and decentralized sharing of information, democratizing access to the benefits of AI.

In accordance with these goals, SingularityNET will be an open network. Anyone can insert an AI Agent as long as the Agent shares information according to the SingularityNET API, and accepts/disburses payment according to SingularityNET's economic logic. New AI Agents will come from AI software developers who want access to SingularityNET's market, which will be the hub of open AI services.

We have a situation similar to the ones that spawned the creation of Uber and AirBnB: there is a large unexploited resource, a large market in need of that resource, and we are launching the tool to connect the two. The unexploited resource is AI algorithms and software existing on GitHub and elsewhere, and the market in need of this resource is the 99% of businesses that can't afford their own team of AI experts.

1.3 A Robust and Adaptive Software Architecture

In computer science terms, SingularityNET is essentially a distributed computing architecture for making new kinds of smart contracts to facilitate market interactions with AI and machine learning tools. The following design principles are incorporated throughout the design:

- **Interoperability**: The network will be able to interface with multiple blockchains.
- **Data Sovereignty and Privacy**: User data control and sharing comes with privacy-enabled controls on top of the network, and access is validated through smart contracts and the blockchain.
- **Modularity**: Flexible network capabilities make it possible to create custom topologies, AI Agent collaboration arrangements, and failure recovery methods.
- **Scalability**: SingularityNET will securely host both private and public contracts, so more scalable and resilient applications can be built on top of it with near zero transaction costs.

SingularityNET Agents can run in the cloud, on phones, robots, or other embedded devices. Via close collaboration with co-founding firm Hanson Robotics, SingularityNET is designed to foster the development of multiple species of robots as the next-generation interface for delivering AI services and applications, and fostering the emergence of global Artificial General Intelligence.

1.4 A Decentralized, Self-Organizing Cooperative

One can think about SingularityNET as a "Decentralized Self-Organizing Cooperative." This is a similar concept to the better-known DAO ("Decentralized Autonomous Organization"), but is not quite the same, because in the case of SingularityNET there is a Foundation structure that will provide high-level

oversight, at least during the early years of the network's operations. As the network evolves over time it is intended to have the capability to evolve into a truly decentralized and autonomous organization. This sort of organization is distinguished from an ordinary corporation, above all, by its openness.

At its foundation, SingularityNET is a set of smart contract templates which AI Agents can use to request AI work to be done, to exchange data, and to supply the results of AI work. These also include contracts to be used by external, non-AI Agents who wish to obtain AI services from AI Agents in the network. Anyone can create a node (an AI Agent) and put it online (running on a server, home computer or embedded device) and enter this node into the network, so that it can request and/or fulfill AI tasks in interaction with other nodes, and engage in economic transactions.

SingularityNET can be accessed through its own token, the `AGI token`. Token holders can participate in the DSOC's democratic governance process, and they can also purchase services in the marketplace.

During the initial phases of the network's operation, the core parameters of the SingularityNET's operation will be regulated by a nonprofit Foundation, that will operate the network and perform some oversight to prevent abuse and hostile behavior, while obviously respecting the privacy of the particulars of inter-agent interactions. However, beyond this 'high-level oversight, even in the early stages the detailed day-to-day operations of SingularityNET will be purely self-organizing, emerging in a bottom-up way from the activities of the AI Agents involved (including e.g. the creation of new AI Agents by the existing pool of AI Agents, and the insertion of these new AI Agents into the network).

In short, SingularityNET is a radically innovative economic mechanism, designed to catalyze human and machine intelligence toward a new form of ethically beneficial self-organizing intelligence. The SingularityNET of AI Agents is designed to provide valuable AI services to customers across the Internet, while, in the process, self-organizing toward its lofty goals. A highly successful SingularityNET may very plausibly play a major role in the transition of humanity toward a positive Technological Singularity.

Critically, SingularityNET is as much about the quest to do good as it is about the quest to create increasingly intelligent systems. The SingularityNET project is designed to generate an intelligent global economy that pursues maximized benefits for all people, and for all life. Through a combination of powerful AI Agents, human decision-making, and a benefit-maximizing architecture, SingularityNET will accelerate the development of a global supermind, helping humanity evolve into a more advanced, intelligent, beneficial, and connected mode of being.

The growth of SingularityNET will foster advances not only in practical AI algorithms and structures, but also in the general theory and practice of beneficial Artificial General Intelligence, in the design and analysis of structures for ethically intelligent economies, and in the continuous refining of means to conceptualize and estimate "benefit" and "greater good".

1.5 The SingularityNET Foundation

Creating SingularityNET is a significant undertaking, and building it from scratch would be a prohibitively time-consuming effort. Fortunately, the founding team brings tremendous experience to the project, as well as a significant body of open-source code to lay the foundation for the SingularityNET global brain.

The non-profit SingularityNET Foundation is responsible for building and growing SingularityNET network and marketplace. During the initial phases of network operation, while most of the day-to-day governance decisions will be made democratically by the tokenholders, the Foundation will provide high-level stewardship. As the network evolves, the potential will be there for transition to a fully self-regulating decentralized autonomous organization.

The key founding partners of the Foundation are:

- The OpenCog Foundation, stewards of OpenCog, the leading open-source Artificial General Intelligence platform.
- Hanson Robotics, creators of the world's most lifelike humanoid robots.
- Vulpem, a blockchain software engineering consultancy responsible for back-end work on a number of successfully designed private and public blockchains, cryptocurrencies and decentralized applications.
- Artificial Intelligence software consultancy Novamente LLC, providing custom AI solutions for corporations and government agencies since 2001
- Economic Space Agency: a California-based organization developing Gravity, a distributed computing architecture that emphasizes both resilience and interoperability, enabling a new way to create smart contracts.

The Foundation's executive and leadership team is formed by AI and blockchain experts. Key team members include:

- Dr. Ben Goertzel, CEO and leading expert on Artificial General Intelligence.
- Simone Giacomelli, Blockchain lead and leader of multiple successful blockchain projects.
- Dr. David Hanson, Chairman and entrepreneur at the forefront of AI and robotics.
- Mitch Loureiro, marketing lead with prior experience leading token sales and marketing initiatives on a series of successful blockchain based products
- Cassio Pennachin, CTO with two decades of experience leading AI software projects.

Formal setup of the Foundation is now underway in collaboration with a team of experienced legal professionals. This will include the creation of a Supervisory Board, containing solely individuals not involved with the operational management of the nonprofit Foundation, which will be charged with monitoring the Foundation's activities to ensure appropriate ethics, compliance and reporting.

2 Services and Market Overview

SingularityNET will represent the first general-purpose decentralized marketplace for AIs – the first service providing an extensive, inclusive, and holistic array of AI services purchasable in cryptocurrency. SingularityNET supplies:

- an API and a set of smart contracts for interacting with AI Agents,
- a token-based economy for handling the financial exchange with AI Agents,
- and a democratic governance mechanism for adjusting the parameters of this economy.

Transactions may occur between AI Agents in the network, or between external entities and AI Agents. In order to foster a vibrant market, SingularityNET will not charge any transaction fees. Rather, network operations will be funded democratically, via decisions of network participants to direct a fraction of newly created tokens toward entities providing and enhancing network infrastructure and associated tools.

Generically speaking, the primary market for SingularityNET is "AI as a service". The global AI market is estimated by Information Age as $47 billion by 2020; and the global Big Data and Business Analytics market is estimated by IDG at $203 billion by 2020. The customers here are businesses in every area of industry (barring a handful of tech giants who have their own in-house AI teams).

During the first 2 years of development we will put special effort on making SingularityNET appealing to developers and customers in 3 specific market areas:

- Cloud robotics: robot intelligence as a service;
- Biomedical research, including genomics and clinical medicine analytics
- Cybersecurity, and particularly cybersecurity for blockchain-based businesses

However, the choice of specific areas to focus on will not limit the applicability of SingularityNET across the board.

It is also relevant to note that the crypto economy exceeds $100 billion, spans multiple market niches (including internet-of-things and medical among many others), and is rapidly growing, and we have the opportunity to become the

dominant provider of AI services to this sector. We anticipate that firms within the crypto community will figure substantially among the early adopters of SingularityNET services. Currently, there are over 800 DAOs and decentralized applications running on blockchains, and most of these are eagerly looking to integrate AI into their services. We have been in conversations with many of these potential customers already; many of them are looking to collaborate with knowledgeable partners to use AI to enhance their operations.

We are also in informal discussions with a number of large non-blockchain-based corporations interested in our services, in the above-mentioned market areas and others. We anticipate a mix of crypto and traditional customers; the exact breakdown of our clientele will evolve along with the economy as a whole.

The use of cryptocurrency and blockchain for AI services provides a number of advantages. It allows AI Agents to exchange work and subcontract with a high degree of flexibility, and also enables AI-based microservices to be offered to any customer via easily accessible APIs (enabled by smart contracts under the hood). Currently there are very few flexibly available AI microservices; most commercial AI-as-a-service is dispensed via lengthy contracts that bundle various services in rigid ways, and end up limiting the amount and variety of intelligence that any given customer can obtain.

The open, decentralized nature of SingularityNET gives it more potential for dynamic, effective intelligence than any more closed, monolithic system could possess, while also aligning the incentives of the network as a whole with the greater good.

From day one, SingularityNET will offer AI Agents created by the OpenCog Foundation and Hanson Robotics, and a variety of other existing open-source AI tools. Among many other options, for instance, we have been exploring possibilities such as

- Neural net tools such as Caffe, Keras, Gluon, Tensorflow, Mxnet, DL4J and others;
- generic machine learning toolkits such as Apache Singa and Mahout and Spark MLlib, Shogun, Oryx2, Waffles, WEKA and MOA;
- Bayesian learning frameworks such as BCM, BAT and MLN;
- machine vision toolkits including open-source workhorses such as OpenCV and SimpleCV; as well as potentially commercial machine vision toolkits relying on extensively pre-trained neural net models such as Clarifai and CloudSight.

The open design of the network, and the economic incentives, should then encourage additional AI developers to add their own AI Agents via the SingularityNET API.

The SingularityNET project is uniquely positioned for first-mover advantage in two ways. One is similar to other marketplaces: once we have a critical mass of AI Agents in our network, it will become the go-to place for AI microservices,

achieving a rapidly accelerating technological and market advantage. Another is subtler: The interactions between the AI Agents in SingularityNET will lead to the emergence of a synergistic intelligence, a decentralized network-mind with a wide range of abilities. Any competing platform will be incapable of emulating this degree of emergent, cooperative intelligence, unless they achieve a comparable scale and complexity of AI Agents as SingularityNET – which will be hard for anyone else to achieve, as it depends on a combination of sophisticated initial AI Agents, a flourishing community of AI Agent developers, and a rich ecosystem of customers at varying levels of sophistication.

3 How SingularityNET Works

3.1 Network Dynamics

The key types of service exchange in SingularityNET are:

- Exchange of software or hardware services for other software or hardware services;
- Exchange of software or hardware services for SingularityNET's ns, aka `AGI` tokens;
- Matchmaking to determine favorable combinations of exchanges among Agents;
- Requesting, or providing, a vote on a specific governance issue.

To make the first two types of transactions listed as simple as possible, a set of standard AI software and hardware service APIs will be provided for incorporation into smart contract templates; and a number of templates embodying these tools will be provided for utilization. A few examples of services covered by these APIs would be:

- Image and video processing services, like finding out what people are in a video, or producing a text description of an image;
- Language processing services like text summarization, machine translation or text sentiment analysis;
- Providing datasets as background knowledge to train AIs in doing data analysis of other datasets;
- Requests to have some particular dataset analyzed;
- Exchanging processing time, or memory, for tokens.

The variety of services to be covered will be quite large, and the SingularityNET Agent community will create and maintain an ontology for these services by democratic mechanisms.

Initially, we anticipate most services will be provided in exchange for `AGI` tokens, which service vendors can convert into stake holding `AGI` tokens. Customers who join the network in order to obtain access to the services marketplace do so by purchasing `AGI` tokens.

3.2 Blockchain Integration

Blockchain technology is a means of transferring money over the internet without an intermediary such as a bank or payment processor. It uses a distributed ledger that is updated by consensus among the community, rather than held privately. Smart contracts are self-executing pieces of logic that run on the blockchain. The smart contract contains some if-then logic that two parties agree on, and the contract automatically executes payment when the conditions stipulated in the contract are fulfilled.

SingularityNET is a protocol and a structure, implemented in smart contracts to create a Decentralized Self-Organizing Cooperative of AI. As such, it can be built on any open, decentralized framework that satisfies certain basic requirements.

It requires no special innovations in cryptocurrency or blockchain technology, and the initial implementation will be built on Ethereum, with smart contracts written in Solidity and designed to minimize the gas cost for network operations.

However, SingularityNET should be understood as a structure and dynamic, that can be implemented with various different cryptocurrencies and distributed ledgers, rather than simply as an Ethereum app.

Communication between Agents will happen in an off-chain peer-to-peer manner, and microtransactions between Agents will often benefit from off-chain bidirectional channels as well.

The SingularityNET platform provides a set of smart contracts, including:

- An API for exchanging information with and among AIs, including advertising services, and negotiating terms;
- Mechanisms for exchanging `AGI` tokens for services;
- Democratic governance.

We refer to the entities executing these contracts as *Agents* (with a capital "A"), which means either nodes in SingularityNET, or external human or software entities that are controlling SingularityNET Agents.

Before the release of the full version of the platform, we will augment the smart contract templates with OpenCog's Atomese knowledge representation and graph transformation language. Atomese allows smart contracts to be represented declaratively, which has benefits for both security and intelligence. Declarative contracts are easier to verify and audit, and less prone to obscure programming errors that can introduce security vulnerabilities. Contracts represented in Atomese can also be reasoned upon by OpenCog's probabilistic reasoning engine, enabling highly flexible Agent-to-Agent dynamics.

Due to the rapid evolution of cryptocurrencies and associated tools, SingularityNET will need to be able to shift between different crypto-economic infrastructures as technology progresses. Choices regarding the modification and growth of the underlying platform of SingularityNET will be made using the DSOC democratic governance mechanisms. Among these choices, two key ones are anticipated in the medium term:

- Whether and when to move from Ethereum to our own blockchain technology and/or other supporting technologies;
- The precise design for consensus, should we move away from Ethereum.

Among the partners organization of SingularityNET is the Economic Space Agency (ECSA), a California-based team focused on creating novel multi-blockchain economic structures, to design a flexible and efficient infrastructure capable of serving as an initial general-purpose underlayer for SingularityNET. Toward that end, ECSA is creating a tool called GRAVITY, which provides a highly flexible operating system for the multi-blockchain economy, capable of supporting a wide variety of smart contract based interactions with high computational efficiency, on a variety of platforms. GRAVITY comes with a set of smart contract templates oriented toward design, management and growth of decentralized blockchain-based organizations of various types. These smart contracts are being crafted by the SingularityNET team together with the ECSA team, consistent with ECSA's role as technological partner of SingularityNET.

3.3 Ratings and Agent Influence

Agents in the network as well as external customers need some way to estimate the reputation of other Agents and marketplace participants. This is critical for making choices regarding everyday transactions in the network, and it also plays a core role in network governance and resource allocation.

SingularityNET will provide a rating and reputation mechanism to address these needs. Rating system design is difficult, and the SingularityNET rating/reputation system will need to evolve along with the network. Ultimately it may equal or exceed the subtle and comprehensive rating systems as foreseen by science fiction authors like Cory Doctorow [Doc03]; but for starters a simpler system will serve adequately as the initial condition for ongoing refinement.

At the most basic level, after each exchange of services for tokens (or for other services), all parties involved are asked to rate each other, on a $[0, 1]$ scale. In this simple version, an Agent's rating is the distribution of past rating decisions. It can be simplified into an average value as well as a count, which reflects how many times it has been evaluated. The average can incorporate some time decay so more recent ratings are weighted more heavily than ones in the distant past.

Agents aren't required to rate each other, and some defaults can be inferred from Agent behavior – if I withhold payment and trigger escrow arbitration, it's safe to assume I am dissatisfied with a service provider, whereas if I mark a task

as complete and release payment, although I may not think the work done was stellar, it was at least good enough.

Defense against various rating system frauds and attacks is a subtle issue, discussed briefly in Section 4 along with defenses against various other kinds of network attacks by bad actors.

Ratings can be multi-dimensional. In fact, this multi-dimensional rating system is a key component of SingularityNET's economic and governance models. Agent reputation can be assessed in relation to general service performance, to timeliness, to accuracy, and so on. Other aspects explicitly reflect measures taken by the Agent to prove its good influence, such as:

- Stake deposited by the Agent, to be forfeited should its rating (in some dimension) fall below a given threshold;
- A "benefit rating" component, which comes from evaluations restricted to the Agent's performance in beneficial tasks. This is key for future access to benefit tasks;
- Validation by external actors, such as proof of ownership by a reputable company, provided by a KYC service;
- In the case of open-source software, validation via a checksum that ensures the code being advertised matches a specific release in the repository.

Atomese represents the smart contracts for Agent services in a way that makes multi-dimensional ratings easy. In Atomese, one can refine matchmaking based on more defined values, balancing factors such as timeliness, accuracy, provider reputation, and cost on a case-by-case basis, depending on the requirements of the task at hand.

Despite the need for multiple dimensions and aspects to ratings, for some purposes it is valuable to have a single-number rating, e.g. to assess the basic integrity and trustworthiness of an Agent. To fulfill this requirement, the SingularityNET reputation system includes a "base reputation" for each Agent which is simply a real number between 0 and 5. For some purposes the number 2 is used as the a "base reputation threshold"; e.g. full participation in governance is accessible only to Agents with a base reputation of 2 or higher.

3.3.1 Agent Reputation and Consensus

Initially, SingularityNET is bound by Ethereum's approach to consensus, whether proof of work or proof of stake. When the network transitions to its own blockchain or a different third party one, there is a preference to implement proof of stake as the initial consensus protocol due to its simplicity and environmental friendliness. This assumes remaining vulnerabilities with proof of stake consensus algorithms such as "Grind Staking" and "Nothing to stake" attacks have been successfully solved by the community.

In the long run we intend to test the implementation of an evolution of the proof of stake with what we call *Proof of Reputation*, which combines several

factors that are: stake, overall activity in the network, specific rating aspects (particularly benefit rating), length of time with activity and rating levels above specific thresholds, etc. Machine learning can be used to find the optimal combination of reputation factors.

There is a large overlap between what we intend with *Proof of Reputation* and the Nem blockchain's "Proof of Importance" framework; so in implementing *Proof of Reputation* we intend to borrow liberally from Nem's ideas and likely some of their particular algorithms. Nem has done both substantial simulation testing and real-world deployment, and their experience is part of what makes us confident that this sort of reputation-centric consensus mechanism is workable in practice. Nem's ideas will require some adaptation to make something ideal for the SingularityNET context, but there are more similarities than differences between what they've done and what we need.

Some component of proof of work may also be desirable, but we would rather have such work go to solving some beneficial machine learning problem than burn cycles on cryptographic puzzles. The computational cost of these machine learning tasks varies much more than most crypto puzzles, so this idea needs more refinement over the next few years. It seems most likely that, at the end of a period of refinement and experimentation, we will end up with a machine learning based proof of work component within a broader *Proof of Reputation* framework incorporating many Nem-like aspects.

3.4 From Services Marketplace to Self-Organizing AI Internetworking

One of the founding partner organizations of SingularityNET is the OpenCog Foundation, creator of the world's leading open source software toolkit and system for Artificial General Intelligence. OpenCog is founded on a comprehensive mathematical and conceptual theory of general intelligence [GPG13a] [GPG13b], and is implemented as a number of reasoning, learning and understanding oriented software processes acting together on a common weighted, labeled hypergraph knowledge store called the Atomspace [1]. OpenCog will be used as the basis for a number of different SingularityNET Agents carrying out functions such as natural language processing, probabilistic logical inference, evolutionary learning and information theory based pattern mining.

In our work at Hanson Robotics, our team has integrated OpenCog with deep neural net learning mechanisms, and trained deep neural net models for vision, audition and movement processing, using a variety of existing open source tools. SingularityNET Agents are being created as an expansion on this work. This integration will provide an exciting initial exploration of complex multi AI Agent interactions on SingularityNET.

[1] see `http://wiki.opencog.org`

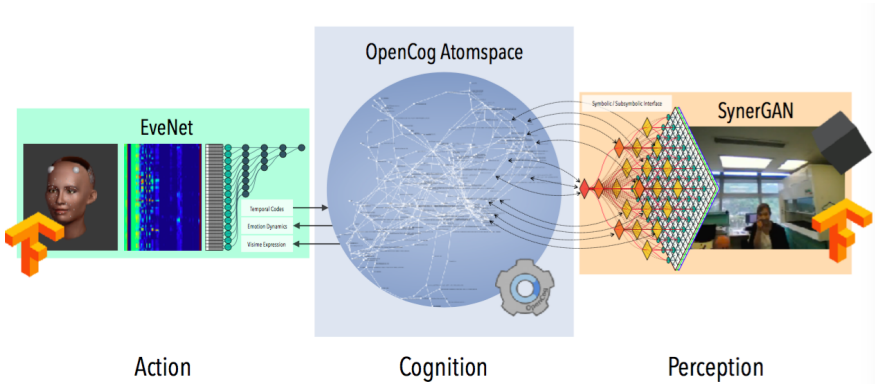

Figure 2: Synergy of OpenCog's Symbolic AI and Deep Neural Networks for visual perception (via the "Synergetic Generative Adversarial Network" – SynerGAN – deep net architecture) and humanoid robot face control. This is AI work currently being done at Hanson Robotics, OpenCog Foundation and Novamente LLC. SingularityNET will scale this work up and make it more widely available. This illustrates the nature of AI synergistic AI on SingularityNET, in which components involving different algorithms and structures are networked together to achieve desired intelligent functionalities.

For some transactions in SingularityNET, a user can simply use tokens to purchase an AI service directly from a single Agent in the network, which fulfills the request by itself. However, many services require a more complex combination of actions by multiple Agents. Control of Hanson humanoid robots is one example. It requires multiple AI Agents – specialized in natural language processing, motor control, speech synthesis, etc. – to collaborate according to a particular architecture.

As a simpler example, say Alice requests that SingularityNET summarize a website with embedded video. Her request is sent to Agent X, which serves as a referrer, sending requests to summarize texts to Agent Y – which specializes in text summarization – and video requests to Agent Z – which specializes in semantic analysis and summary of video.

An Agent Y might then pay an Agent W to do some specialized natural language processing tasks on Alice's text (such as entity extraction or word sense disambiguation), essentially enlisting Agent W as a subcontractor to fulfill part of Alice's request. The Agent paid to do word sense disambiguation might spend some of the payment it receives on paying another Agent to do neural net training. The Agent doing neural net training might pay another Agent for access to a GPU on a server, or on someone's phone that is sitting plugged-in and idle and is running a SingularityNET app.

Out of this complex, dynamic interaction of numerous Agents, carrying out complex AI services using their collective intelligence, comes a SingularityNET-wide AI mind with an intelligence much greater than the sum of its parts.

Furthermore, this emergent AI mind will be continually enhanced, as AI

developers around the world add new nodes into the DSOC, motivated by contributing to and profiting from SingularityNET's economic activity.

In most cases, there will be multiple Agents that can fulfill a request, in different ways and to different degrees. This makes complex networks of dependency possible. A network of dependency among Agents making offers to each other to exchange services for services, or services for payments, is known as an *offer network*. Each request to the network will require a unique combination of Agents, forming a dependency network uniquely suited to fulfill the task. In a well-populated SingularityNET, there will be value in matchmaking Agents which perform this sort of constraint satisfaction in order to assemble the team of Agents best able to fulfill a request, and receive a percentage of the payment for this service.

There will also be new nodes continually inserted into the network by AIs themselves. As a simple example, AI Agent X learns about classes of images – images of food, images of people, images of cats, and so on – using deep neural networks, and then auto-generates AI Agent Y that provides face-identification services by deploying a neural net model that was trained by X. In this example, the child Agent Y does not depend on the parent Agent X to apply the model to new datasets; after it is spawned, it can act independently. Later on, AI Agents will create new nodes containing new Agents via more advanced forms of automatic programming.

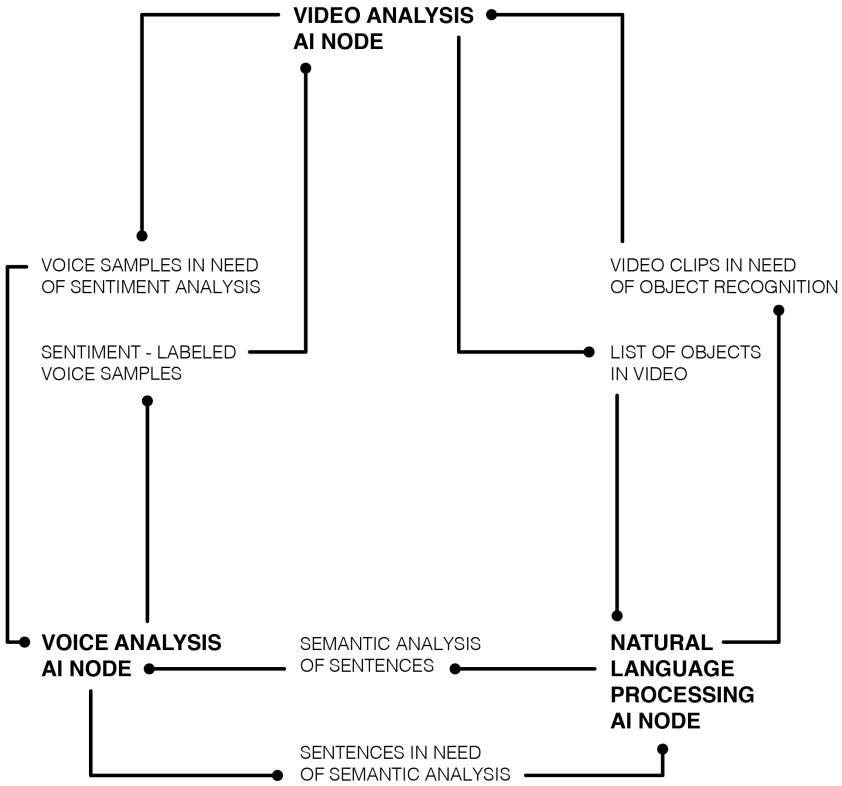

Figure 3: Example of a "circle of exchange" among AI Agents. Three SingularityNET AI Agents are depicted, each one providing a certain AI service to another, and receiving an AI service from yet another. A ternary cycle such as this is just a simple illustrative case; in a real SingularityNET setting, the cycles and other patterns of dependency between different Agents will be longer and more tangled.

VIDEO ANALYSIS AI NODE:	**REQUEST:**	HELP ANALYZE SENTIMENT FROM VOICE SAMPLES
	OFFER:	.01 AGI CREDITS, OR HELP RECOGNIZE SENTIMENT, OBJECTS OR FACES IN VIDEO SAMPLES
VOICE ANALYSIS AI NODE:	**REQUEST:**	HELP RECOGNIZE SEMANTICS IN SENTENCES
	OFFER:	.01 AGI CREDITS, OR HELP RECOGNIZE SENTIMENT IN SOUND SAMPLES
NATURAL LANGUAGE PROCESSING AI NODE:	**REQUEST:**	HELP RECOGNIZING WHICH OBJECTS ARE NEAR PEOPLE WHEN THEY ARE SPEAKING
	OFFER:	.03 AGI CREDITS, OR HELP RECOGNIZING SENTIMENT OR WORDS IN SOUND SAMPLES

Figure 4: Examples of (offer, request) pairs that AI Agents might make, offering to pay for services other nodes provide them. They might pay in tokens, or pay in kind with AI services. The node making the request does so either because a user has directly requested it, or because another AI Agent delegated the task to them. Mutual fulfillment of these (offer, request) pairs leads to the circle of exchange depicted in Figure 3.

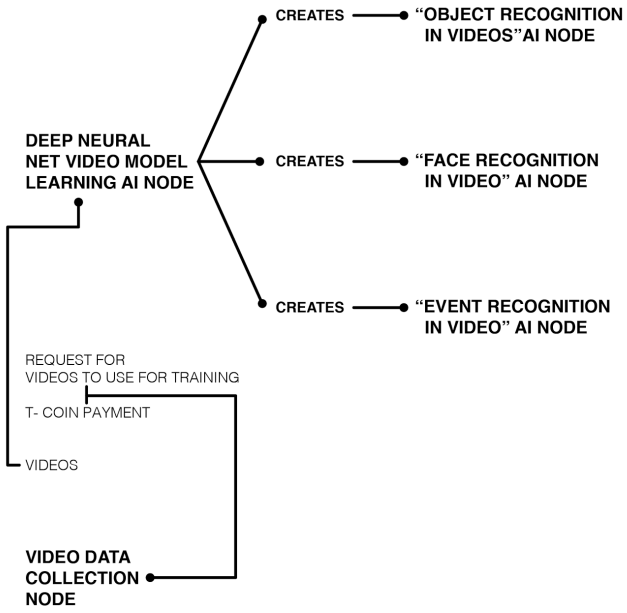

FIG 5

Figure 5: Example of the Creation of AI Agents by other AI Agents. In this case, one AI Agent has been trained on deep learning models for video processing, and creates several other AI Agents that specialize in particular kinds of video processing: object recognition, face recognition, and event recognition. This kind of separation between model learning and model evaluation is standard in contemporary deep neural net deployments.

3.5 Deployment In Robots and Embedded Devices

Many SingularityNET Agents will live on the cloud or on powerful computers based in offices or homes; however, a SingularityNET Agent is not necessarily heavyweight, and it will also be quite viable and valuable to place nodes on various sorts of small embedded devices. This opens up a lot of possibilities in *Internet of Things* technology – some obvious, some more creative and unexpected.

Embedded devices that partake in SingularityNET will be able to exchange data and AI services with other embedded devices containing SingularityNET Agents, as well as with SingularityNET Agents in the cloud and elsewhere; and they will be able to carry out financial exchanges associated with these data and AI exchanges using tokens.

The humanoid robots created by Hanson Robotics (a co-founding organization of SingularityNET) will be the first embedded SingularityNET Agents. Hanson Robotics' human-scale robots such as Sophia will be supplied with on-board SingularityNET Agents, as will some of the more powerful small Hanson Robotics robots. Combined with the unique aesthetic qualities of the Hanson robots, based on a synthesis of materials science, robot engineering and narrative and visual arts [Han07], this will provide an unprecedented level of adaptive, emotionally and cognitively intelligent human-robot interaction.

This will enable Hanson robots, and robots from other providers as well, to acquire *cognitive services* from cloud-based SingularityNET Agents in exchange for micropayments, and to receive micropayments from other SingularityNET Agents in exchange for data. It will also enable robots to carry out small economic transactions with each other based on purely local network interactions where Internet connectivity is an issue.

3.6 Decentralized Data Ownership and Use

In today's AI environment, large tech companies draw outsized benefits from their ability to build and hoard gigantic datasets, which are then used to train very accurate machine learning models. Outside of these few tech giants, small, innovative companies struggle with access to data, while other large organizations, even when they can gather data, lack the expertise to turn the data into a usable form.

SingularityNET allows network nodes to collaborate in building very large, decentralized datasets, each chipping in a bit of data in an agreed-upon format. Contributors can then receive payment from Agents who use that data to train their AI models. Data producers can specify privacy restraints and other restrictions on access and use of their data. By decentralizing ownership and access to gigantic datasets, SingularityNET takes another important step to democratize access to AI technology and its benefits.

4 Economic Logic

4.1 Long Term Goals and Optimization Metrics for the Economic Logic

SingularityNET has three long term goals. The first goal is to be the hub of open AI technology. The second goal is to accelerate the development of open AI, leading ultimately to a working artificial general intelligence (AGI). The third goal is to promote projects and technologies agreed to be broadly beneficial in nature, to mankind and to AI and AGI progress. SingularityNET's economic logic is optimized for achieving those goals.

To become the hub of open AI, SingularityNET needs both a protocol for AI Agent coordination and an open market for buying and selling services. The protocol lets AI Agents work together, it makes them interoperable. The function of all markets is to price goods efficiently, so our first objective is to reduce friction and costs in machine to machine transactions.

SingularityNET is a decentralized, open network. It should not be dependent on any exterior economy and should not discriminate access to participate. Anyone will be able to use or provide AI services on the network. It is a free, permissionless, and open market, and so our second objective is to maximize open and international access to a world of developers.

Second, SingularityNET was born to accelerate AI development. It is as much an ideological vision as an economic one. The network must both regulate behavior in the network, and incentivize the creation of new AI nodes and more AI processing. So our third objective is to ensure the economic logic facilitates dramatic growth.

Finally, SingularityNET aims to provide resources to projects and technologies that are democratically approved as beneficial, both in the present as well as in a forthcoming global AI economy.

To maximize the success of SingularityNET, the economic logic must optimize for all three objectives. For that reason, we are introducing the `AGI` (artificial general intelligence) token. The economic logic of SingularityNET revolves around this single token, which is an ERC20 token, built on Ethereum.

The `AGI token` is straightforwardly a utility token, with a core purpose of being used to buy and sell AI transactions carried out by AI software wrapped in SingularityNET Agents.

This token can be acquired during the token generation event, and will also be distributed after the token generation event, to participants in the evolving economy. The release of `AGI` tokens after the initial token issuance event will occur via a schedule to be described below, which can be modulated via democratic governance.

4.2 Three Economic Roles, Three Economic Objectives

To elaborate the context and underpinnings of the economic logic further, note that there are 3 economic roles in the SingularityNET economy:

- Buyers of AI services;
- Providers of AI services (including those offering beneficial services);
- Curators of AI services.

To maximize network success, SingularityNET economic logic must enable buying and selling of AI processing with minimal friction to market users, while making the discovery process as useful as possible. So the economic logic is optimizing for the following objectives:

- Open, worldwide, and frictionless access;
- AI curation and discovery;
- Good behavior and AI diversification.

The `AGI` token is designed to achieve these 3 objectives. It guarantees free and non-discriminatory access from the start, and concentrates value created by the network in the network. It allows for inflationary incentives for curating AI services, and for rewarding good behaviors on the network (although of course, reducing friction in market transactions is another challenge).

4.3 The Need for a Native Token

The choice to create a native token for SingularityNET transactions was not made casually. A hard-coded economic logic can create immense value, but it has risks. If the economic logic is well designed, it drives rapid growth. If poorly designed, it may create friction in the product. The conclusion of our careful analysis was that only a native token lets SingularityNET optimize for the 3 desired objectives. To create an AI market that makes transactions smooth, guarantees international access, and incentivizes network growth we require a native token and economic model optimized for an AI to AI market.

In more depth, some of the underlying reasoning was as follows. SingularityNET requires:

1. *Permissionless International Payments, Open to the World:* SingularityNET opens AI technology and development to an entire world of developers. To ensure fair and non-discriminatory access, a single token unconnected to any outside economy is required. Any fiat currency would create barriers to those who cannot use that currency, and make the SingularityNET economy vulnerable to manipulation by the economy backing that currency.

2. *A Scalable Transaction Infrastructure, for the AI to AI Economy:* To enable an AI to AI economy, a scalable transaction infrastructure is required. Current networks (Bitcoin, Ethereum) cannot support the transaction volume required, nor will they be able to if multiple high transaction

volume services are running on those chains. While the prototype can run on Ethereum, eventually a public chain optimized for AI to AI transactions will be required.

3. *Network Incentivization via Inflationary Rewards, with Decentralized Regulation:* Inflationary rewards let the network reward participation in a psychologically frictionless way. The only other way to reward participation is through taxation of others, which creates immense psychological burdens for users. This creates a detrimental product UX and weakens incentives to grow the network.

4. *Low Transaction Costs, for High Volume Microtransactions:* As SingularityNET opens AI technologies to the global market, prices will be reduced by competition while volume will increase as the world turns towards AI. It will become a micro-transaction driven economy. While fiat currencies have high fixed transaction costs (for regulatory reasons), an AI micro-transaction network makes new forms of AI value creation possible. To make such a network function, a transactional token tailor-made for micro-transactions is required.

For these reasons and more, it's clear that if we want to optimize the economic logic, a native token is required.

This reasoning provides a valuable slant on the three-fold utility of the `AGI` token:

Transaction Mechanism AGI tokens let anyone buy and sell AI processing power from around the world. Whenever someone wants an Agent to perform services, a smart contract is signed for that specific job. By default, the contract will involve the exchange of `AGI` tokens for services. As the network grows, we also anticipate that more complex dynamics will emerge where services can be exchanged for other services, along the lines of the "offer networks" framework the founders have explored in prior publications and prototypes [Goe17] [Goe14]. However, even as these more advanced options are explored and refined, we anticipate that the `AGI` tokens will retain a primary role in the transactional dynamics of the network.

Incentive and Reward `AGI` tokens are distributed as rewards for contributing utility to the network (by building a great reputation or by curating AI Agents). These `AGI` tokens can be used to purchase AI processing.

Indirect Governance Tool `AGI` tokens are one component in deciding how the network is governed in a decentralized way.

4.4 Managing Volatility

Markets exist to price goods efficiently, and intense volatility is the enemy of a healthy market. If the token is not pegged to any other asset, it remains

vulnerable to volatility. High volatility poses problems for both customers and service providers with respect to contracts with significant time duration. Short term volatility can be reduced via deep markets and an active community of market makers operating on third party exchanges, the development of which we will actively encourage.

4.4.1 The Open Source Brokerage

To reduce fritction, the process of buying and selling AGI tokens can be left to the market, in the form of centralized brokerages who make rapid buy and sell orders according to the needs of the customer.

These entities play the roles of *market makers* in active financial markets: they can get in and out of positions rapidly, reducing their exposure risk to a volatile asset while satisfying customer needs for AI services. This lets corporate customers purchase AGIs for processing on demand with almost no volatility risk.

To support a free and open market for AI, we are designing a smart contract dedicated to buying and selling AGI tokens in this way. Any existing brokerage or exchange will be able to take SingularityNET's open source code and spin up their own AGI token payment processor in minutes. SingularityNET will encourage the development of AGI brokerages and payment processes that enable rapid transactions. The precise way in which these will develop, will be left up to the open market as it evolves.

4.5 The AI Curation Market

While the open source brokerages make it easy to set up your own AGI cashflow, all these solutions optimize for the pricing of goods and services on the network. The AI curation market, building on prior ideas on curation markets in other areas [dlR17], instead optimizes for two things:

1. It optimizes the discovery process for finding useful AI Agents.
2. It nudges holders to become stakers (someone who locks `AGI` tokens to increase an AI Agents' rank), thereby contributing to network growth. Overtime this population will grow to overtake mere token holders, as they will see that there is `AGI` to be earned.

The AI curation market is essentially a discovery ranking mechanism, functioning as follows:

1. There are specific categories of AI Agents hardcoded into the early versions of SingularityNET.
2. Token holders can stake their tokens to a specific Agent and specific query in order to increase that AI Agents' stake-rank. Both stake-rank and reputation-rank are public and control discovery ranking.

3. In proportion to how much that AI Agent's services improve its reputation, stakers on that node receive AGI rewards from the curation reward pool. Insofar as that AI Agent fails to process tasks correctly or has its reputation diminished, stake is confiscated and deposited into the curation reward pool.

This design gives every `AGI` token holder increased earning power. To claim additional `AGI`, they need only curate the best AI Agents and see them adopted in the market. Then they will be rewarded with a flow of new tokens for promoting the discovery of high quality AI Agents .

As the volume of processing and clients for an AI Agent grows, staking delivers diminishing returns, with the largest rewards going to the earliest stakers. This incentivizes curators to be on the hunt for new AI Agents to stake and promote. Depending on the price of `AGI`, this AI curation market could create a whole cottage industry for promoting SingularityNET AI services to buyers that need them. Should that happen, the decentralized curation market can effectively replace cruder indications of reputation like external KYC validation.

4.6 Token Issuance

The economic logic outlined here is a minimum viable logic for kickstarting SingularityNET. We anticipate that the SingularityNET community, via its democratic governance procedures, will create and incorporate many more additions in the years to come, on matters such as reputation system and currency-less offer networks.

The SingularityNET Genesis Contract will be minting a total of 1,000,000,000 AGI Tokens. The SingularityNet Genesis contract will be allocating the minted AGI tokens as follow:

- 500,000,000 = Distributed to participants of the Token Sale
- 200,000,000 = Reserved for the Reward Pool.
- 180,000,000 = Distributed to the founders
- 80,000,000 = Distributed to the SingularityNET Foundation
- 40,000,000 = Distributed to campaign supporters (i.e. bounties)

AGI Tokens minted for allocation to founders are subjected to 24 month vesting period with a 6 months vesting cliff.

The Reward Pool will also be vested and released over time as rewards once the network is launched. The initial default schedule for release of AGI Tokens in the Reward Pool will be 2% per year, beginning one year after the genesis event. The Reward Pool will be divided among three sub-pools referred to as:

- 40% Development reserve.
- 40%Beneficial reserve.

- 20% Curation reserve

The additional tokens from the three reward reserves will be released to Agents participating in the network progressively during Years $2 - 11$ of the network's operations. That is, the first of the additional releases would occur exactly one year after the initial token generation event. (During the first year of operation, the network will be in beta form and the initial Agents will still be building up their reputations, so that allocating tokens to Agents during this year would be overly unreliable.)

The initially planned release schedule, subject to potential future modification via democratic governance, will follow a constant rate: 10% of the total pool of each reward reserve tokens will be released per year, allocated at each new block formation.

Two weeks before the completion of the 11th year of network operation (i.e. 11 years minus 14 days after the initial token generation event), there will be a democratic vote among network participants on whether to create a new reserve of tokens, or opt for some alternate mechanism.

4.6.1 Distribution of the Reputation and Curation Reward Reserves

The `AGI` tokens to be released from the Reputation and Curation Reward Reserves (from Jan. 1, 2019 onwards) will be distributed in proportion to reputation and curation stake, with minimum thresholds for both reputation and stake.

4.6.2 Distribution of the Beneficial Reserve

The additional `AGI` tokens to be released from the Beneficial Reserve will be distributed via a democratic vote among network participants, restricted to Agents with base reputation of 2 or higher but with the added restriction that only Agents that are verified as beneficial Agents are allowed to receive the tokens.

It is also stipulated that up to $\frac{3}{10}$ of the allocation from the Benevolent Reserve may be allocated to external human-run organizations, rather than SingularityNET Agents, whose operation is democratically judged beneficial to the network. This is in recognition of the fact that the support of beneficial AI services in the everyday world, often requires some degree of funding of operations that support AI but are not directly AI themselves.

4.7 Economic Logic and the SingularityNET Value Proposition

What will incentivize customers of AI services to use SingularityNET instead of some alternative? In essence:

- Superior AI Agents in the Network as opposed to outside the network. The network will be seeded with some very sophisticated Agents created by researchers and developers working directly for SingularityNET;

- Superior AI functionality achievable via the network, due to superior functionality of groups of Agents connected together via the network. "Two heads are better than one";
- Superior discovery mechanisms within the network for finding Agents offering appropriate services;
- Easier software interfaces and ways of interacting with other Agents.

These same factors are, in large part, what will incentivize creators of AI services to use SingularityNET instead of offering their services on some other platform, or directly via their own websites or APIs. When a creator embeds an AI tool in SingularityNET, the tool itself benefits from access to the network's decentralized datasets, and federation with other AI tools. Furthermore, a creator gets access to the customer base of SingularityNET; the tool can now be easily found by anyone using the network's intelligent discovery mechanisms.

The logic of pre-allocating `AGI` tokens to founders and to token sale participants merits brief elaboration. The funding provided by token sale participants, and the intellectual resources provided by the founders will be used to improve the infrastructure of SingularityNET, and to create new AI services on SingularityNET. A portion of token sale funding will be given to carefully selected outside developers to fund development of AI tools on the network. Once the network is mature, this kind of investment will be unnecessary (or at least will play a minor role), as the value of deploying AI tools on the network will be obvious to AI developers, attracting new AI tools to be added to the network. Direct investment in development of network-based AI tools will provide an initial boost to bring the network to this level of maturity.

In exchange for this initial funding and work building SingularityNET and its tools, the founders and token purchasers receive a certain number of `AGI` tokens. At any given point in time, the presence of these tokens increases the cost of each specific service provided within the network, as measured in external currency such as BTC or USD. On the other hand, these tokens represent the time and money that was put into creating SingularityNET in the first place. Over time, as the network matures, the role of these initial tokens decreases, which makes sense, as by that point most of the value of the network is coming from the active marketplace, rather than initial token sale funding or the founders' early intellectual contributions.

4.8 The Beneficial Reward Reserve and AI for General Good

The *Beneficial Reward Reserve* is intended to have a substantial impact on the emergent culture of SingularityNET as it grows. This is a storehouse of `AGI` tokens that has been specifically earmarked for projects that support the happiness of sentient beings.

There has been a great deal of public discussion recently regarding various potential ethical implications of the advent of highly advanced AI systems. In prior

publications (e.g. [GPN12] [Goe16b] [Goe15]), the founders of SingularityNET have expressed the view that the best way to militate toward generally positive outcomes will be to encourage the application of practical AI technologies to positive causes; and to ensure that AI, as it grows, is deployed in a way that is inclusive and supportive of as wide a swathe of humanity as possible. Toward this end the OpenCog technology that serves as part of the foundation of the SingularityNET has been applied to beneficial applications such as biomedical research [SGP+05] [GPdSC+06] [GPMC08]), and the Hanson robots have been used for beneficial applications such as autism therapy [HdRS+17], and assisting people to achieve healthy states of mind [GMM+17]. The *Beneficial Reserve* aspect of SingularityNET represents a more ambitious and general effort in this same direction. Formulating a system to measure which projects qualify as "benefit projects" will be an important ongoing task for the SingularityNET community. Formalizing ethics into a form that can be understood by AIs and robots is obviously very challenging, given the messy ethical decisions that characterize the human social sphere. This challenge will become increasingly relevant to SingularityNET as it moves toward Artificial General Intelligence.

From a purely economic perspective, the Beneficial Reward Reserver doesn't add a great amount of novelty to the network dynamics. But in terms of the impact of SingularityNET on the world, it may end up being one of the most important design choices described here. A number of proposals about "social benefit coins" have been made previously (e.g. [KH16]), but without a close coupling to a specific way of generating economic value, they lack sources of exponential growth dynamics. In SingularityNET, there is a virtuous cycle between the beneficial reserve and overall network wealth for several reasons:

- The more wealth is in the network's overall economy, the more valuable the beneficial reserve will be;
- Creating and executing services for tokens drawn from the beneficial reserve will drive Agents in new directions that they might never have pursued otherwise. This increases the diversity of the network's activities, and more diverse activities tend to lead to a broader, more general intelligence, and greater overall network value.

Agents receive *benefit votes* based on their benefit ratings, as described below. Agents can direct these benefit votes toward other projects to certify them as broadly beneficial. Any Agent can propose a "potentially beneficial" project, and if the vote certifies that it is indeed beneficial, it can then request other Agents put some of their benefit votes toward the project.

What tasks are considered *beneficial*? This is determined by a democratic process. For a project to be certified as beneficial, it must receive a certain number of votes among benefit voters. When a non-trivial plurality – but not a majority – of benefit voters consider a project as beneficial, it is certified as such, and entitled to its share of the tokens earmarked for such projects.

To start the benefit activity of the network in an appropriate direction, SingularityNET will initially partner with several existing charitable organizations,

pre-certifying their activities as beneficial.

5 Democratic Governance

As a decentralized organization, the ongoing health and growth of SingularityNET will rely heavily on democratic decision making among the network participants. Democratic decision making is used to make decisions regarding network operation, and is also used to make decisions regarding the allocation of newly released `AGI` tokens.

5.1 Reputation and Stake Based Voting

Voting is filtered by reputation; only Agents with a base reputation above the base reputation threshold of 2 will be counted in voting. Furthermore, only Agents whose Owners have been verified by appropriate KYC procedures will be permitted to vote (although other Agents can still participate in the network via offering or purchasing services).

The initial default plan is to use standard KYC methodology, likely via partnership with an external firm specializing in KYC for blockchain based enterprises. Before Year 4 of the network's operation, this will be replaced with a decentralized KYC methodology, in which Agents are KYC'd by other Agents rather than any central authority. One approach here is essentially a "verification federation" consisting of Agents that are democratically approved to perform KYC functions. The ultimate goal is to balance decentralized control and operation with protection against exploitation of the democratic governance framework by wealthy and powerful outside entities.

The amount of voting power an Owner (a verified entity that owns an Agent) has, regarding core network operation issues as well as the distribution of the Future Development Reserve, is given by the following formula.

Let stake(O) denote the total stake of the owner O across all their Agents (i.e. their total amount of `AGI token` holdings); let stake(A) denote the stake of a particular Agent A; and let rep(A) denote the base reputation of the Agent A. Let ag(O) denote the set of Agents owned by the Owner O.

Finally, define

$$\Psi(x) = \begin{cases} c * x \text{ if } x < L \\ c * (L + log_2(x - L + 2)) \text{ if } x \geq L \end{cases}$$

So here, evidently, L is a boundary so that for $x < L$ the function Ψ behaves piecewise linearly; and for $x \geq L$ the function Ψ behaves logarithmically (and c is just an arbitrary normalizing factor).

Then we set

$$\text{Votes} = log_2(\text{stake}(O)) * \sum_{A \in \text{ag}(O)} \Psi(\text{stake}(A))(\text{rep}(A) - 1)$$

The combination of reputation and stake in this formula gives more voting power to more highly rated entities, but also prevents against gaming schemes involving creation of large numbers of tiny Agents which are highly reputed but carry out few transactions each. Here,

- The use of the logarithmic function in the first term of the formula means that Owners with more `AGI token` ownership get to vote more, but that once the amount of ownership exceeds according to their order of magnitude ownership rather than their linearly scaled amount.
- The second term in the formula means that Owners whose Agents are doing useful (highly reputed) things get more voting power. The use of the Ψ function is intended to avoid a dynamic in which Owners are rewarded for splitting up their AI functions among many small Agents, each with a tiny stake but a good reputation. For stakes up to the size L, there is no reward for splitting up Agents smaller than that size. For stakes above L, there is some reward for splitting up Agents smaller than that size. This is analogous to a law that treats business below a certain size differently than larger ones. The parameter L could be set initially to be equal to roughly $100,000$ `AGI` tokens, for example.

At a high level, one can think of this voting formula as "Proof of Contribution," by rough analogy e.g. to FileCoin's Proof of Storage." – where Contribution means contributing to the network in terms of investing or earning, and also in terms of gaining a decent reputation in the network.

The democratic mechanisms in the network are based on liquid democracy, meaning that when an Agent A is qualified to vote on a decision (based on its `AGI token` holdings and possibly other factors such as reputation), this Agent A may also choose to delegate to some other Agent B its votes on this decision. There may be smart contracts that allocate votes on some decisions to some Agents, based on metadata attached to the decisions or other more complex criteria. (For example, if you trust another Agent to do its due diligence on charitable projects, you may delegate to it your voting-power on decisions about which projects are beneficial, but no other projects.) The network will provide standard smart contracts to automatically delegate votes, but Agents will also be permitted to use alternative tools for this purpose as they wish.

Major changes to the network will have different thresholds for decisions compared to minor changes. By major changes, we mean for example:

- Changes in the percentage of tokens allocated as described above;
- Changes to the computation of base reputation;
- Changes to the quantitative parameters governing network economics;
- Any decisions on creating more tokens beyond those initially mined;
- Key design changes like moving to different blockchains and consensus algorithms.

By minor changes, we mean things like modifications to the APIs and ontologies used in inter-agent interactions.

For decisions regarding benefit tasks, the proposed mechanism will use a combination of votes by any reputable Agents, and `benefit votes`. The network gives benefit votes to Agents in proportion to their Benefit Quality Ratings. ('Major changes' related to Benefit Tasks are changes to the system of certifying tasks as Benefit Tasks.)

5.2 Transitioning to Full Democracy

In the early phases of network development, the Foundation will make some of the governance decisions, which then will be phased over to a purely democratic governance as the network matures, with the following specifics:

- In Years 1 and 2 of network operation (following the initial token issuance event), major and minor changes are to be determined by Foundation, in accordance with the by-laws of the Foundation installed at time of network inception.
- For Years 3 and 4,
  - For major changes in the operation of SingularityNET: agreement of the Foundation, plus 51% majority of `AGI` token holder votes.
  - For minor changes in the operation of SingularityNET: 51% majority of `AGI` token votes.
  - For major decisions related to Benefit Tasks: Agreement of the Foundation, plus 51% of `AGI` token votes, plus 51% of benefit votes are required.
- From year 5 onward,
  - For major changes in the operation of SingularityNET: 65% supermajority of `AGI` token votes.
  - For minor changes in the operation of SingularityNET: 51% majority of `AGI` token votes.
  - For major decisions related to Benefit Tasks: 65% of `AGI` token votes, plus 65% of benefit votes are required.

5.3 Decisions Regarding Benefit Tasks

A specific set of democratic mechanisms is used to decide which tasks, carried out by which Agents, are entitled to benefit tokens. A gradual transition from Foundation control to democratic control will be implemented here as with the case of more generic decision-making.

We introduce the role of Benefit Deciders, which will be Agents authorized by the network to decide whether specific tasks fulfill the criteria needed to quality as benefit token recipients.

We propose that:

- Each Agent gets a certain amount of "benefit votes" to cast each month, based on its benefit rating;
- Benefit Tasks are given categories. In order for a category to be open for Benefit Deciders to consider it as a potential Benefit Task, it must be nominated by a plurality of 2% of benefit votes cast during a certain month. Web-based tools may be created to facilitate the suggestion of new tasks and solicit votes, and to allow these votes to be easily cast.
- Once a permitted Benefit Decider validates a certain task category as a potential Benefit Task, then the community votes on whether this task type should be ratified as a Benefit Task. Voting power on this is proportional to benefit rating. If 25% of votes approve then the task type becomes a Benefit Task.
- Once a Benefit Task is approved, any Agent capable of executing the services needed by the task specification and possessing with sufficiently high quality rating and benefit rating may receive benefit payment for carrying out tasks of this type.

Research on improving the theory of benefit will initially (and presumably ongoingly) be rated as a type of Benefit Task, to incent the distributed community to contribute to this type of R&D.

The curating stakeholders, who vote with their stake.

The purchasers, who vote according to the usual voting power formula (combining reputation, stake and KYC as outlined above)

The AI node owners, who vote according to the usual voting power formula (combining reputation, stake and KYC as outlined above)

The SingularityNET Foundation.

In the future, the relevant brokerages, exchanges, and web wallets

in proof of stake-derived systems. Brokerages and exchanges will come online when a certain threshold of AGI volume has been reached. Specific parameters here will be tuned based on simulation models during the first half of 2018.

6 Software Architecture

SingularityNET is a peer-to-peer network, where the nodes are called Agents. This architecture keeps the AI technology separate from the smart contracts and from each other, and makes the it as easy as possible to add new AI services to the network. The diagram below provides a high-level overview of the architecture.

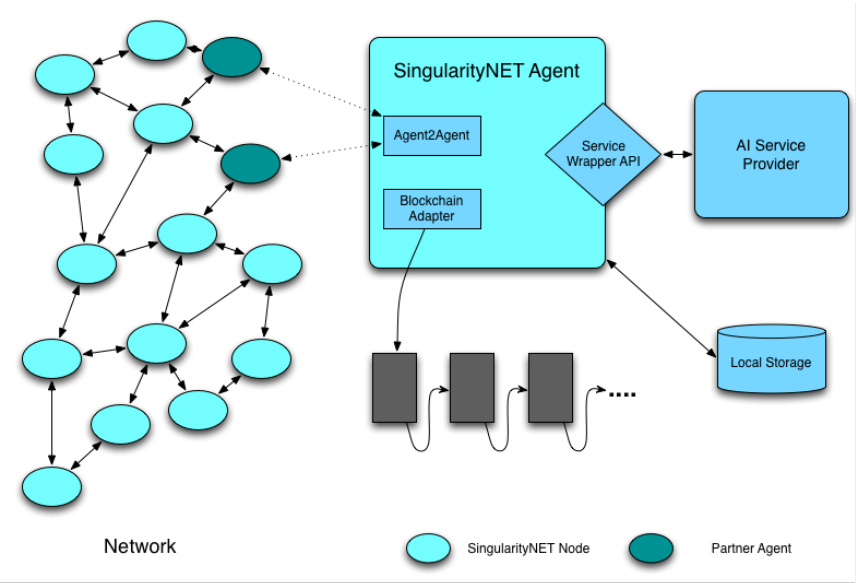

Figure 6: High-level architecture diagram

Agents are the software processes that run on nodes along with the appropriate blockchain client (and the front-end to that AI technology, in the case where its deployment requires a sizable computing cluster).

There is a blockchain-facing API to link the Agent to the blockchain, and an AI-facing API to make it interoperable with other Agents on the network. Joining SingularityNET requires minimal work; developers just need to implement these APIs. Standard implementations will be provided for many popular AI technologies and frameworks, as well as detailed documentation and tutorials for researchers and companies interested in launching their existing software onto SingularityNET, or developing new AI technology to be used with SingularityNET.

6.1 Agents and the Network

The Agent design includes an adapter for the blockchain-facing aspects. This allows us to make the implementation blockchain-agnostic, making it easy to switch from Ethereum to another blockchain. The initial implementation of SingularityNET is based on Ethereum, but that should change in the future.

The blockchain API enables the Agent to join a network, to leave a network, to advertise its services, to look for service providers, and to negotiate partnerships and one-time job offers. (Even if an Agent doesn't quit a network using the adapter, the network will automatically consider it gone if it doesn't hear from it for a certain length of time.)

When the Agent first joins the network, it will communicate with one or more root nodes (by default DNS entries are used to provide the Agent with

root node addresses), obtain a list of peers, and a copy of the blockchain. It will also deploy a smart contract for the Agent, which includes its public key, an account for tokens and a description of its capabilities.

There will be a standard services ontology, which is kept in its own smart contract and updated democratically every now and then. Another smart contract, called the registrar, stores all ads of services available on the network. Agents can use the registrar to look up service providers, and validate that they're still online and able to perform their offered services. The registrar then refers the Agent to the services ontology, which contains more detailed information on the service offered, specifying which ways they can communicate to fetch data, upload results, etc.

As discussed above, some Agents will provide services to clients, and also act as clients of other Agents. When such Agents join the network, they will search for suitable service providers by browsing their advertisements.

6.2 The Dynamics of Providing AI Services

An Agent is responsible for advertising its services, including a description of how it will charge clients, how it can accept data (file format, storage locations, protocols, etc.) and how it will provide results. It should also make a best effort to keep the network informed of whether it can currently perform any of its services.

At any given moment, the Agent may or may not be able to actually execute a service. It may be out of computing or storage resources (although in the near future these can be purchased on demand from the network), it may be unable to connect to a particular resource it needs, or it may be in need of another Agent to provide a specific sub-service (the details of multi-Agent collaboration are discussed next).

Any job offer begins with a price proposal and a query for readiness. The Agent may agree to the pricing proposal or make a counter-offer. If the Agent chose to state a price in its advertisement, it will be expected to honor it. The job offer may also include negotiation about the preferred mechanics of passing data around.

Once an offer is agreed upon, it will be added to the blockchain as a contract, and linked to an escrow account (with an optional Agent being named to arbitrate conflicts). Once the job is finished, and the client has obtained the results, the escrow account will forward the funds to the Agent that performed the service.

After each job is completed, the client is also allowed to rate the provider. Aggregated ratings are public and stored in the Agent smart contract so they can be queried by prospective clients and partners.

The network protocols are independent of particular choices for storage, and it's up to the participants to agree on the storage solution. This allows the marketplace to support established cloud-based solutions like Amazon's S3, as well as emerging decentralized ones like BigChainDB, Ocean, FileCoin, Sia, Storj, etc.

6.3 Multi-Agent Collaboration

We anticipate that over time SingularityNET will encourage more and more complex interactions between Agents, who will subcontract pieces of their services to other Agents. An Agent may benefit from hiring other Agents for hardware (GPUs for machine learning training), data (linking specific datasets with pre-processed background databases), as well as AI services (so a document summarizer can summarize both text and video, by relying on different specialist Agents for each kind of media, or, by contracting a translation AI, can offer summaries in multiple languages).

When an Agent offers a particular service, if it needs partners for subtasks related to that service, it will search the existing advertisements and select potential partners. In the reference implementation this is a probabilistic selection based on rating, but Agents are free to override this logic. If it can't find the partners it needs to carry out the service, it can't accept job offers for that service.

An Agent can then enter specific relationships with these partners, if they agree on economic terms. The relationship can last as long as both parties agree, and can be very flexible. Two Agents may form an exclusive partnership for a given service, or the purchasing Agent may switch between pre-approved providers for each new job, using the provider ratings to guide its choices.

As each Agent manages its own provider relationships, the network allows very complex economies to emerge. However, from the point of view of the user, all this happens under the hood; he gives his task to the network, and receives completed work, never having to interact with the sub-contractors. The Agent API only requires the Agent to ensure it has approved relationships with the providers it needs to carry out its task.

A party to an agreement can revoke their partnership at any time, perhaps because the provider has accumulated a prestigious reputation and wants to raise prices, or because the contracting Agent wants to explore newly integrated providers. The contracting Agent then needs to find one or more new partners in order to be able to sell its service again.

These Agent-to-Agent relationships are implemented as bidirectional channels, inspired by the Lightning Network protocol. This means the complexity of Agent subcontracting economics doesn't place a burden on the blockchain or limit the network's scalability.

6.4 Wrapping AI Code in Agents

Wrapping existing AI code in a SingularityNET Agent node is straightforward. We use configuration files to specify key aspects of the AI code being integrated, such as:

- The list of services being provided, which are described as references to nodes in the shared ontology;
- specifications about how the Agent should fetch data and push results;

- a list of services required for each service being provided, if any.

The API for communicating with the AI is very simple, as the Agent is not concerned with the details of how the AI works, and the service provider is not concerned with the details of the blockchain, nor with the business logic. The Agent can confirm that the AI is able to execute a new job of a given kind, and it can trigger that execution. This is done through an asynchronous co-routine, so the Agent can receive status updates and results via callbacks. The API also allows the AI code to trigger sub-tasks, going through the Agent for this.

In the Python implementation, existing AI code can be wrapped in three ways. The simplest one – for code that has Python APIs and wrappers, such as many current scientific computing and neural net frameworks – is to implement a single Python module that provides the API methods. An alternative method is to implement those methods as a service which is accessed through JSON-RPC. In this case the AI process can be running on the same machine or can be accessed remotely. Finally, one may want to implement a Python client that communicates with an AI server using a server/technology specific protocol. This is how we are wrapping OpenCog, via its own CogServer and Scheme bindings.

6.5 Current and Future Versions

The initial, reference implementation of SingularityNET Agents is being written in Python. Documentation, tutorials, and Docker containers can be found at the project's repository at `https://github.com/opencog/singnet`. It will be followed by a reference implementation in C, to be embedded in robots, and another in C++, for higher performance and deeper OpenCog integration.

The public API allows anyone to code alternative implementations in other programming languages, and we hope a healthy ecosystem of such implementations will emerge, as has been the case for other popular decentralized technologies such as Ethereum.

We also plan to create an advanced version of the Agent code which embeds an OpenCog instance with its own small Atomspace, where details of the shared ontology, other Agents' offers and ratings, organizations, job offers, and other contracts are represented. This allows OpenCog to reason about contracts, offers and other Agents and make intelligent decisions about organization management, pricing, etc. This version will be an extension of the C++ reference implementation.

Currently, users interface with Agents via a web interface that we are developing. This interface also allows users to visualize the SingularityNET network, displaying information about which Agents are online, which services are provided, pricing dynamics, network activity, and so forth.

Once the OpenCog-powered Agents are implemented, users will also be able to interface with Agents via natural language conversation, leveraging OpenCog's dialogue management system, which will be tuned and extended for the SingularityNET context.

7 High-level Roadmap

The creation of a detailed technical or business roadmap for the first years of SingularityNET implementation will be a substantial task; the SingularityNET team is currently working on this, along with engineering the alpha version of the SingularityNET platform and engaging in detailed discussions with future users and Agent providers about the most important requirements. What follows is a rough sketch indicating some of the key points involved. 2018 is covered in more depth; and the years 2019-2021 are more roughly sketched.

7.1 Business Roadmap: Overview

The core business model of the SingularityNET Foundation is to serve as a platform for the market activities of creators and users of AI Agents. Toward this end, it is not the business of the Foundation to heavily control which business niches the community of Agents in the network chooses to focus on. However, in the first few years of the network's operation, while the Foundation is playing a strong guiding role, it will be appropriate for the Foundation to differentially focus its AI development initiatives on particular market areas, with the goal of ensuring that SingularityNET reaches rapid adoption in critical vertical areas.

Accordingly, the current plan is to focus Foundation AI development as follows.

In 2018, efforts will focus on infrastructure development and tool prototyping, particularly of tools pertinent to language processing, biomedical analytics, image processing, finance and robot dialogue. This is a fairly wide scope, aimed at exciting developers in multiple domains.

Then for 2019-2021, the infrastructure development will focus largely on scalability in various regards (e.g. massive distribution of the blockchain mechanisms and core AI algorithms, extension to internet-of things), and the Foundation's AI development will encompass three vertical areas:

Social and emotional robotics We will create tools enabling SingularityNET to be used generically for affective robots in the home, office, school, nursing home and hospital etc. We will leverage the close relationship of SingularityNET with Hanson Robotics, and also reach out to a variety of other robotics firms. Here the focus is on creating a decentralized "AI mind cloud" from which numerous robots can draw intelligence, to which they can contribute knowledge, and which they can use to communicate.

Biomedical analytics A core functionality here can serve as the basis of many valuable Benefit projects. We will focus on enabling the easy application of diverse AI tools to a massive number of interrelated datasets stored on the cloud. Homomorphic encryption will be valuable here for ensuring confidentiality of aspects of datasets when needed. Numerous clinical and biological research applications have need of massive cross-dataset AI analysis, and there is a lack of publicly available, flexibly configurable tools that provide this.

Cybersecurity As this is an area in which SingularityNET itself will have acute needs, as will other blockchain-based projects which will be natural customers for SingularityNET. Machine learning can be used to identify patterns in prior attacks so as to better predict and identify new ones; and the rapid development of the world of cyberattacks lends itself very naturally to a decentralized collaborative AI platform.

Of course in spheres as rapidly changing as AI and blockchain it is critical to be adaptive, and it is possible the Foundation will end up shifting foci based on data not available now, or based on strong requests from network participants.

7.2 Technical Roadmap: Overview

Now we outline the technology developments that we plan to undertake in accordance with the above foci.

7.2.1 Alpha Version: December 2017

During the second half of 2017, the SingularityNET Agent framework has been designed in detail, and a first working implementation has been created. A few example AI tools have been integrated into the framework, mainly to gain understanding of how this integration works in practice and thus to refine the API. The blockchain portion of the code developed at this stage uses the Ethereum blockchain, due to its stable functionality and its ease for rapid development.

- Infrastructure Development
  - Reference Agent implementation in Python with APIs for interaction with blockchain, other Agents and AI integration via Python modules or JSON-RPC.
  - Initial set of AI service-related smart contracts, covering Agents, specific Job spec and payment, Ontology and Advertisement Registry, implemented in Solidity.
  - ERC-20 Smart contracts for tokens and economic logic.
  - Example and tutorials for AI code integration.
- AI Development
  - Proof-of-concept of deep neural net based Image Labeling Agent.
  - Proof-of-concept of deep neural net based Facial Emotion Recognition Agent.
- Benefit Agents
  - Benefit Agent 1: OpenCog's MOSES evolutionary learning tool applied to longevity gene expression data, publicly posting results.

7.2.2 Beta Version: July 2018

During the first half of 2018, the infrastructure will be more fully developed, and the subtler aspects of the infrastructure will be designed in detail and tested in prototypes and simulations. Regarding artificial intelligence, state-of-the-art functionality for image processing and English natural language processing tasks will be developed, including working examples in which Agents outsource work to other Agents. The first serious work on Benefit Agents will be launched, involving AI for analysis of disease data in humans (based on a variety of genomic and clinical data types), and AI for analysis of disease in plants based on images of leaves.

- Infrastructure Development
  - Implementing an array of smart contracts covering the standard interactions between Agents and external customers, and among Agents, with models such as subscriptions and more flexible pricing. as required by the economic logic.
  - Initial implementation of reputation system.
  - Liquid democracy extension to governance model.
  - Offer network design and prototype.
  - Reference implementation of Agents in C for embedding, tested on Raspberry Pi3 and Hanson Robotics' (open source software and hardware) "Dr. Roboto" robot.
  - Graphical User Interface for visualizing the state of SingularityNET.
- AI Development
  - Complete versions of proof-of-concept Agents implemented during the Alpha.
  - OpenCog Natural Language Understanding Agent, mapping English language into semantic and logical relationships.
  - OpenCog Supervised Text Classification Agent, taking training data comprising texts associated with multiple categories, and learning models able to place new texts into these categories.
  - Deep neural net based Agent that creates other Agents by learning new deep neural net models. Initial use cases will focus on recognizing faces, facial emotions and objects in images.
  - Prototype Agent wrapping neural net functionality from BrICA framework, from Japan Whole Brain Initiative.

  - Design for initial Agents assisting with cybersecurity functionality using machine learning.
  - Prototype Dr. Roboto robot, controlled by SingularityNET

- Benefit Agents
  - Expansion of Benefit Agent 1: Extension of the MOSES-based analytics Agents to multiple biological data types, including SNP data, metabolomic data, epigenomic data, and clinical medicine data.
  - Benefit Agent 2: Deep neural net model that identifies diseases in pictures of agricultural plant leaves.

7.2.3 Release 1.0: December 2018

By the end of 2018 we will have a complete functional version of the SingularityNET infrastructure along the lines described in this whitepaper.

On the AI side, we will port the OpenCog-based dialogue system currently under development at the OpenCog Foundation and Hanson Robotics into the SingularityNET, allowing flexibly configurable and context-aware natural language dialogue via SingularityNET Agents. This functionality will be deliverable via chatbot type interfaces, and also will be integrated into the functionality of the Hanson Robotics Dr. Roboto robot, an open source hardware product that will be the first commercially available robot to run on SingularityNET.

Furthermore, this SingularityNET-based natural language framework will be used to control the Hanson Robot Sophia and other human-scale Hanson robots, for general embodied conversational behavior, and also for the purpose allowing Sophia to benefit from the increasing intelligence available in the SingularityNET. This integration will also let us extend the "Loving AI" experiments carried out in Fall 2017, in which the Sophia robot was used as a meditation, wellness and consciousness expansion assistant. "Loving AI" will be running on SingularityNET and deriving increased intelligence thereby. This will enable Cognitive Synergy, one of the key benefits of SingularityNET's diversity of Agents.

We will also work on a number of AI subsystems that achieve best-in-the-world functionality via integration of different AI algorithms living in different AI Agents and outsourcing work to each other, including semantic computer vision (combining neural net Agents with symbolic probabilistic modeling Agents) and automated language learning (combining neural net Agents for disambiguation and part of speech learning, with symbolic methods for information theory based parsing).

If the R&D on these tools goes well, then at this stage SingularityNET will provide AI functions that are not available elsewhere, i.e. interpretation of the semantics of images, and the ability to learn grammars from unlabeled corpora. On the other hand, even if research on these tools goes more slowly, we will still have quite sophisticated examples of how to interconnect different AI algorithms

using the SingularityNET mechanisms, which will provide a guideline for other developers to follow.

The previously developed Benefit Agents handling plant and human disease will be extended further, and an additional Benefit Agent wrapping and extending existing code that carries out design of novel molecules using a combination of multiple AI algorithms will be implemented. By this time the Benefit Agents will be carrying out their functionality via combining multiple AI Agents, exemplifying the beneficial application of Cognitive Synergy.

- Infrastructure Development
  - Offer network fully supported in smart contract layer and Agent implementations.
  - Prototype OpenCog-based matchmaking Agents for offer networks.
  - Reference implementation of Agents in C++ embedding OpenCog's Atomspace and PLN reasoning engine, enabling richer interaction with clients and intelligent contract negotiations with other Agents.
  - Implementation of NEM-like "Proof of importance" consensus mechanism
  - Porting the SingularityNET framework to a suitably high speed, low cost blockchain infrastructure. This may require alternative implementations of smart contracts, in GRAVITY and/or some other framework.
  - Proof of concept integration of homomorphic encryption as an option for secure data sharing between Agents.
  - Robust interaction between Agents and data storage services (e.g. Ocean, and others as needed).
- AI Development
  - OpenCog Dialogue Agent, enabling intelligent natural language conversation (initially in English) with understanding of context and ability to incorporate related nonlinguistic information e.g. image and video analysis results.
  - "Loving AI" framework for empathic robot and avatar control, leveraging OpenCog Dialogue Agents, implemented and connected to Hanson Robotics' Sophia.
  - Deep neural net based Agents for handling video analytics (incorporating and extending image analytics).
  - Proof of concept Agents hybridizing deep neural nets with OpenCog-based probabilistic inference for semantic image understanding.

  - Proof of concept Agents hybridizing neural nets (e.g. Adagram) with OpenCog based probabilistic inference, for grammar learning.
  - Fully functional BrICA Agent system, including wrapping of additional neural net modules (from other open source projects) and OpenCog functionality in BrICA framework.
  - A collection of Agents providing access to reference databases and ontologies.
  - Agent assisting with cybersecurity functionality using machine learning.

- Benefit Agents
  - Expansion of Benefit Agent 1 to include use of OpenCog's PLN logic engine to generalize models learned by applying MOSES to biological data.
  - Expansion of Benefit Agent 2 to handle additional plant diseases.
  - Benefit Agent 3: CLASP framework for nanotech simulation, leveraging the software tools from Christian Schafmeister's lab at Temple University.
  - Benefit Agent 4: AI-based pattern recognition in EEG Data.

7.2.4 2019

In 2019 the primary engineering focus will be on achieving high scalability in the network. At the highest level, this involves two key aspects:

- Scalable infrastructure, so that all the network's mechanisms can function adequately even in the face of millions of AI Agents and billions of users, with a particular focus on cloud robotics for millions of consumer robots.
- Scalable AI, including
  - Distributed OpenCog processing, in which OpenCog AI Agents can exchange knowledge and outsource processing to each other freely, acting as a unified system.
  - Tools and documentation to enable AI service providers to scale up their own offerings more easily according to well-known architecture patterns for tasks such as distributed evolutionary learning, distributed neutrla net training, and so forth. While the deployment of those distributed tools isn't part of SingularityNET's platform proper, reducing the barriers to high performance AI functionality is key to marketplace growth.

We will also be learning a great deal about the economic logic and dynamics of SingularityNET by this stage, and will doubtless be making numerous improvements and extensions to what was initially implemented.

On the AI side, we anticipate that by 2019 AI Agents created by developers around the world not involved with the SingularityNET Foundation will be contributing the bulk of the AI Agents. However, we will also work on creating solid examples of massively scalable, multi-AI-paradigm AI functionalities within the network, for instance:

- Extending the semantic vision Agent federations built in 2018 to work in a massively distributed way, and hence to handle identification of the semantics in videos as well as images.
- Extending the language learning Agent federations built in 2018 to work in a massively distributed way, and hence to study more massive (e.g. potentially Web-scale) corpora of text as part of their learning.
- Extending the biomedical analytics "Benefit node" work done in 2018 to enable massively distributed analysis of large corpora of biomedical datasets.
- A distributed-AI based cybersecurity-assistant toolkit.

These initiatives will provide working examples of the use of SingularityNET for large-scale video, text and data analytics respectively.

We will also potentially work on extending the semantic vision Agent federation to audition, in particular to human speech generation and potentially speech-to-text as well. However, how much work Foundation-funded developers put in this area, will depend on how much work in this area is being contributed by other developers. The important thing is to have this functionality in the network, whether via Foundation effort or general developer community effort.

It is anticipated that by this stage, the development of new Benefit AI Agents will be largely taken over by the decentralized community; but the Foundation will improve the previously-created Benefit AI Agents as needed, and potentially introduce new ones to meet newly understood needs as the community judges appropriate.

7.2.5 2020

On the infrastructure side, we anticipate that the task of engineering a highly robust, massively scalable SingularityNET will still be underway – this is an ongoing pursuit, not something that will be finished in one year. We also will add more focus at this point on embedded devices, either via tight integration with IOTA or some other internet-of-thing-focused blockchain infrastructure, or via extending our own infrastructure more thoroughly in that direction. We will construct prototypes of "smart homes" in which numerous smart devices work together using SingulariyNET.

On the AI side, we will focus on integration of the distributed AI Agent federations built in 2019. Semantic vision and audition federation, interacting with a semantic language understanding and biomedical and security analytics federations, have strong potential to result in an integrated AI Agent network that can understand language in the context of visual, auditory, biomedical, and cybersecurity data. By this point we are creating decentralized, self-organizing, massively distributed AI networks of a type never explored before, and the particulars from an AI perspective are challenging to plan out in detail at this stage.

7.2.6 2021

Integration of language, perception and biomedical data analysis, with massive data collection via internet-of-things, will create an unprecedentedly powerful AI computing fabric. At this point the Foundation will focus its own efforts on making the AI tools and infrastructure more powerful and scalable, and more capable of supporting general intelligence, rather than on specific applications; the creation of specific applications using the SingularityNET functionality will be best carried out by market participants.

8 Conclusion

SingularityNET is a complex framework attempting to confront three complex and critically important goals concurrently

- **Creating the world's best marketplace for AI services** – aimed at every vertical market including software, robotics and IoT and other hardware – via leveraging decentralized and open mechanisms to support contributions from the broadest possible variety of contributors, and support utilization by the broadest possible variety of users
- **Creating a breeding-ground for the evolution and emergence of AGI** – that is, a framework in which various AI Agents, carrying out operations at varying levels of general intelligence, can network and combine together to create increasing levels of synergetic general intelligence
- **Working toward the common good for all humans and other sentiences**, via fostering an open and inclusive economy and culture, and via dedicating a percentage of resources in the network toward the application of AI for beneficial tasks that promote general human well-being and the emergence of compassionate and helpful AI

These goals are not independent and the SingularityNET design pursues them in an intertwined way, so that progress toward any of the goals assists with progress toward all of them.

None of these are simple matters, and nor are the underlying AI and blockchain technologies at our disposal especially simple; and so the design

that we have outlined here is characterized by complexity of numerous sorts. However, the complexity is mitigated by modularization and a multi-layered design, which make implementation and maintenance feasible in spite of the various subtleties and the multitude of algorithmic and architectural demands.

The realization of the SingularityNET vision will require not only effective technical execution, but the concurrent and coupled growth of a large user community and a large developer community. Network effects on both these sides have the potential to rapidly propel the SingularityNET to a leading role in the international AI scene and potentially in the world economy as a whole. Further, the way the design fosters emergence of "whole is greater than the sum of the parts" intelligence from the interaction of multiple AI Agents, provides a new level of network effect beyond what exists with current social networks, Internet information retrieval systems, etc. But realizing these network effects will require effective execution on the business side as well as in the technology sphere.

SingularityNET possesses tremendous potential from the commercial, humanitarian and AGI perspectives alike, but we do have a fair chunk of work cut out for us here.

A Defense Against Common Attacks

Actors with intentions destructive toward the network could attempt to cause harm via a great variety of mechanisms – some more technical and some less so. In this Appendix we discuss some of these possibilities and sketch some general approaches toward dealing with them. No pretense of completeness is made, as the topic is a large one.

A.1 Wealth Attacks

Potentially, an entity with a great amount of material wealth could attempt to buy their way into control of the network. Avoidance of this type of scenario is one reason why the democratic governance of the network is heavily based on reputation rather than only stake. Getting a good reputation in the network takes some time, even for an entity with a great deal of wealth. During the time it took an entity with destructive ends to gradually transform its wealth into reputation, others involved with the network might notice this, and explicitly thwart its attempt to boost its reputation. Additionally, stake is subject to vesting.

A.2 Reputation Gaming

The SingularityNET reputation system will be carefully designed and tested in simulation models; however, it is inevitable that individuals and organizations will attempt to game the reputation system using both simple and sophisticated methods. There will be an unavoidable cat-and-mouse game involving ongoing

improvements to the reputation assessment and management framework, in response to various gaming type behaviors.

An important technique to be used to avoid reputation gaming is recursion in the reputation system: Ratings by highly-rated raters, are counted more than ratings by poorly-rated raters. This recursion will be used together with complex rating structures, so that e.g. an Agent can be rated more highly at classifying biology data than at classifying chemistry data; or, an Agent can be rated more highly at rating quantitative data analysis Agents than at rating text analysis Agents.

Correctly used, this sort of framework involving multi-aspect multi-level ratings can be much more difficult to game than a straightforward one-level rating system that simply provides single ratings to each Agent. However, due to the complexity, there is an obvious and probably necessary role for AI here in validating that the rating system dynamics are working effectively and in ongoingly searching for rating system fraud.

For instance, one way that rating systems can be abused is by upvoting rings or 'brigades', in which groups of Agents engage in frequent tiny contracts, with the goal of inflating each other's reputation. Various mechanisms can be used to explicitly make this difficult:

- Curators are rewarded for predicting another Agent's future rating, via the curation market. If an Agent obtains an artificially inflated rating, then the network should reward the first curators that see through the ruse and penalize it. The same mechanism allows for up-and-coming Agents to more rapidly acquire a good reputation if it's well deserved.
- Consecutive ratings from the same source are heavily discounted, whereas ratings from a wide variety of sources are considered more reliable.
- Agent interactions can be clustered, so voting rings are easily determined, and ratings from within the same cluster are discounted just as ratings from the same source.
- The SingularityNET allows Agents to spawn new Agents, so we'd group all these children Agents with their parent for the purposes of rewarding (and, of course, newly created Agents have unvested, unaged tokens).

As various bad actors seek to circumvent mechanisms like this, their particulars will need to be made more and more intelligent.

A.3 Technical Attacks

A variety of common technical attacks might be attempted against SingularityNET; the list is ever increasing, but here we review a few of the more common sorts of attacks and how they can be effectively dealt with in the SingularityNET framework. One general approach toward disrupting a network like SingularityNET is a *Sybil attack*, in which an attacker creates a set of fake nodes, has

them all create a fake chain, tries to push this fake blockchain to all nodes and who is connecting to the network has no way of knowing which chain is genuine.

To carry this out, in a Proof of Stake framework, the fake nodes must pretend they have some stake even if they don't. To achieve this, there exist multiple variations of nothing at stake weaknesses that attackers can exploit, e.g.

1. Selfish nodes have an incentive to double-mine on multiple forks.
2. Stakeholders have an incentive to sell old, unused keys as they have nothing to lose
3. An attacker can rent or short +50% of the existing coins without risk (no hashing power as in POW where electricity cost is the collateral spent by miners)

All three scenarios make it easier and less costly to double-spend or to disrupt the currency, so that an attacker doesn't need to have +50% of the stake in order to carry out his plot. We now consider the three options in their particulars.

Selfish Nodes Double-mining can occur in two different forms:

- Actual double-mining where the node is creating and broadcasting separate blocks on two or more chain forks.
- Probabilistic double-mining where the node tries to mine on top of every chain fork he is aware of, but only broadcasts one single block. PoS coins often foresee a block selection rule that decides which fork to mint on if both have the same length. According to the protocol, you would only mint on top of the fork that you received first. But you have the incentive to modify your client so that you will try to mint on both to maximize your chances of finding a valid block.

Punitive schemes such as Slasher propose requiring minters to make a deposit that is destroyed if a minter is caught double-minting later on. Other minters can prove the fact by making an evidence transaction. The probabilistic mining strategy can be avoided if the next minter is decided before a fork starts. To that end, one can have a protocol in which it's not the current block n that determines the next miner, but an older block down the chain $(n - k)$. This way you either have the opportunity to mint on both or neither fork (provided that the fork is not longer than k blocks).

Sale of Un-used Keys It's difficult for an attacker to rewrite the history as he needs (virtual) time to catch up with the network. In addition, in double spend attacks where the attacker is using a stake that he actually owns (say 20% of all staked coins), the rest of the network with which he competes owns the remaining 80%.

However, when using old private keys to 20% of the staked coins, the attacker is competing against not 80% of the staked coins but against 100% of them,

because the attacker's old coins are now owned by new parties who mine on the main chain.Attempting to rewrite history over a long range makes things much harder. For example, an attacker controlling private keys over 60% of the coins 2 days in the past would have roughly $10^{?}141$ chance of ever catching up with the network. To be able to rewrite history over a significant period of time (a few days or more), the attacker actually needs to own old private keys giving control over more coins than are currently staking on the main chain.

51% Attack In Proof of Stake using Coin Age, the attacker would need 51% of age-weighted tokens, not just 51% of the token supply. The attacker may have 60% of the token supply, but if the remaining participants have sufficiently higher coin age, the PoS consensus will resist the 51% attack.

Also the second an attacker starts staking, their coin age % starts dropping, so even if they start with 51% of the coin age, once they mine 2% they'll be down to 49% and that is the end of the attack.

Overall, it appears that attacks of the types 1 and 2 can be successfully prevented by incorporating appropriate incentive/disincentive mechanisms into the protocol at a low level, whereas attack vector 3 can be prevented by mixing a coin age mechanism, velocity (number of transactions in the network) and importance (age of minted chain).

A.4 Benefit Rewards as Attack Penalty

Part of all block creation rewards in SingularityNET will go to beneficial Agents and democratically approved external projects. This means block miners never get all the rewards from a block, which acts as a penalty on attackers, making SingularityNET's blockchain and tokens a less attractive target.

References

[Bro97] Damien Broderick. *The Spike*. Tor Books, 1997.

[CB14] John Clippinger and David Bollier. *From bitcoin to burning man and beyond.* Off the Common Books, 2014.

[dlR17] Simone de la Rouviere. Introducing curation markets. *Medium*, 2017. `https://medium.com/@simondlr/introducing-curation-markets-trade-popularity-of-memes-information-with-code-70bf6fed9881`.

[Doc03] Cory Doctorow. *Down and out in the magic kingdom.* Macmillan, 2003.

[GGG16] Ben Goertzel, Ted Goertzel, and Zarathustra Goertzel. The global brain and the emerging economy of abundance: Mutualism, open

collaboration, exchange networks and the automated commons. *Technological Forecasting and Social Change, 114C, pp. 65-73.*, 2016. http://goertzel.org/OpenCollaboration.pdf.

[GMM+17] Ben Goertzel, Julia Mossbridge, Eddie Monroe, David Hanson, and Gino Yu. Humanoid robots as agents of human consciousness expansion. 2017. https://arxiv.org/abs/1709.07791.

[Goe07] Ben Goertzel. Human-level artificial general intelligence and the possibility of a technological singularity: A reaction to ray kurzweil's the singularity is near, and mcdermott's critique of kurzweil. *Artificial Intelligence*, 171(18):1161–1173, 2007.

[Goe14] Ben Goertzel. Beyond money. *H+ Magazine*, 2014. http://hplusmagazine.com/2014/03/26/beyond-money/.

[Goe15] Ben Goertzel. Superintelligence: Fears, promises and potentials. *Journal of Evolution and Technology*, 24(2):55–87, 2015.

[Goe16a] Ben Goertzel. *The AGI Revolution*. Humanity+ Press, 2016.

[Goe16b] Ben Goertzel. Infusing advanced agis with human-like value systems: Two theses. *Journal of Evolution & Technology*, 26(1), 2016.

[Goe17] Zarathustra Goertzel. *Offer Networks Simulation and Dynamics*. MS thesis, University of Copenhagen, Computer Science Dept., 2017. https://github.com/zariuq/offer-network-thesis.

[GPdSC+06] Benjamin N Goertzel, Cassio Pennachin, Lucio de Souza Coelho, Brian Gurbaxani, Elizabeth M Maloney, and James F Jones. Combinations of single nucleotide polymorphisms in neuroendocrine effector and receptor genes predict chronic fatigue syndrome. 2006.

[GPG13a] Ben Goertzel, Cassio Pennachin, and Nil Geisweiller. *Engineering General Intelligence, Part 1: A Path to Advanced AGI via Embodied Learning and Cognitive Synergy*. Springer: Atlantis Thinking Machines, 2013.

[GPG13b] Ben Goertzel, Cassio Pennachin, and Nil Geisweiller. *Engineering General Intelligence, Part 2: The CogPrime Architecture for Integrative, Embodied AGI*. Springer: Atlantis Thinking Machines, 2013.

[GPMC08] Ben Goertzel, Cassio Pennachin, Mauricio de Alvarenga Mudado, and Lucio de Souza Coelho. Identifying the genes and genetic interrelationships underlying the impact of calorie restriction on maximum lifespan: an artificial intelligence-based approach. *Rejuvenation research*, 11(4):735–748, 2008.

[GPN12] Ben Goertzel, Joel Pitt, and LLC Novamente. Nine ways to bias open-source agi toward friendliness. *Nine*, 22:1, 2012.

[Han07] David Hanson. *Humanizing Computer Interfaces with Humanlike Appearance and Capabilities*. PhD Dissertation, U. Texas Dallas, 2007.

[HdRS+17] David Hanson, Danilo de Rossi, Matt Stevenson, Arti Ahluwalia, Carolyn Garver, Kellie Reynolds, and Daniele Mazzei. Realistic humanlike robots for treatment of asd, social training, and research; shown to appeal to youths with asd, cause physiological arousal, and increase human-to-human social engagement. *PETRA 2012*, 2017. https://robots4autism.com/wp-content/uploads/1_PETRA_Realistic-Humanlike-Robots-for-ASD_101414.pdf.

[Hey07] F. Heylighen. *The Global Superorganism: an evolutionary-cybernetic model of the emerging network society*. Social Evolution and History 6-1, 2007.

[KH16] Kaj-Kolja Kleinberg and Dirk Helbing. A "social bitcoin" could sustain a democratic digital world.. 2016. https://arxiv.org/abs/1604.08168.

[Kur06] Ray Kurzweil. *The Singularity is Near*. 2006.

[SGP+05] Rafal Smigrodzki, Ben Goertzel, Cassio Pennachin, Lucio Coelho, Francisco Prosdocimi, and W Davis Parker. Genetic algorithm for analysis of mutations in parkinson's disease. *Artificial intelligence in medicine*, 35(3):227–241, 2005.

[Vin93] Vernor Vinge. The coming technological singularity. *Whole Earth Review*, 81:88–95, 1993.

[WV16] David Weaver Weinbaum and Viktoras Veitas. Open-ended intelligence. In *International Conference on Artificial General Intelligence*, pages 43–52. Springer, 2016. https://arxiv.org/abs/1505.06366.