*Draft Version 0.09*

# GenesisAI Protocol

## Protocol for Machine Learning Networking

Archil Cheishvili
archil@genesisai.io

David Fan
david@genesisai.io

November 4, 2018

GenesisAI is a machine learning protocol. On top of this machine learning protocol we are building decentralized marketplace for AI products and services. Simply put, it is Alibaba for AI models. The GenesisAI marketplace connects companies in need of AI services with companies interested in monetizing their AI technology. GenesisAI brings together AI technologies from all over the world to help lay the foundation for the eventual global Artificial General Intelligence.

Nowadays there is no working protocol for ML networking. Because it does not exist yet, engineers spend many hours manually linking ML models. Companies and developers reinvent the wheel. It takes lots of money and hours to make AI products and services.

We make it easy for service providers to link together machine learning models so that they can add functionality and improve accuracy.

# Contents

# 1 Vision

## 1.1 The Beginning

AI holds the potential to revolutionize our world even more so than electricity and fire. However, there are fundamental problems that are currently holding back the AI innovation. So far, there is no way for AIs to communicate: they cannot exchange data, trade services, learn from each other, or leverage their combined capabilities to solve the problems our world faces.

Blockchain technology provides the best framework for the creation of a new type of economy: the AI to AI economy, in which AI bots trade services, communicate with each other, and exchange data and knowledge. Blockchain technology makes it possible to connect all of the worlds AI services into a single platform to lay the foundation for the eventual emergence of Artificial General Intelligence.

GenesisAI merges blockchain technology with AI to produce the first functional protocol for Machine Learning networking. We have 3 main goals:

1. To connect companies in need of AI services with companies who would like to monetize their AI technology. This will be achieved by developing the GenesisAI marketplace, where AI services can be sold and purchased by individual agents.

2. To connect many different AI services and leverage their data to eventually synergize into an Artificial General Intelligence. To achieve this, the GenesisAI platform has been designed to enable the interaction of multiple service providers (nodes) that can easily integrate with each other.

3. To confront the current system of AI oligopolies, where only a handful of large corporations own and operate AIs by making AI technology accessible to small companies and individuals.

## 1.2 The Team

We have gathered a team of recognized experts, thought-leaders, and entrepreneurs to solve foundational problems within the field of AI.

**Archil Cheishvili** - Archil studied economics at Harvard University. He is a serial entrepreneur with 2 exits. Archil sold his first company when he was 18. Prior to GenesisAI, Archil was the CEO of an AI-powered people analytics software company called Palatine Analytics. He has been recognized by the New York Post, Yahoo Finance, and others.

**David Fan** - David studied Applied Mathematics in Computer Science at Harvard University. He has work experience as a Software Engineer at Google, APT, and Dataminr.

**Mena Gadalla** - Mena has raised over $1 million in grants for his research projects. Mena's academic experience ranges from a Harvard University PhD in Applied Physics to a Harvard Master's in Computational Science. He has published multiple scientific works.

GenesisAI's team is comprised of leading blockchain and AI experts whose goal is to lay the foundation for the eventual emergence of a benevolent Artificial General Intelligence which is both operated and owned by the people via our platform. The platform we are creating will connect different AI services to each other and make the power of AI available to those who are unable to create their own AI technologies or otherwise afford them.

### 1.3 Advisors

**Professor Thomas Magnanti** - Former Dean of Engineering at MIT. Institute Professor. Founding Director of the Singapore-MIT Alliance for Research and Technology (SMART).

**Neil Flanzraich** - Lead Independent Director of Chipotle. Former President of Ivax Corporation (acq. by Teva for $10B). Executive Committee member of Syntex Corporation (sold to Roche Holdings for $5.3 B).

**Travis May** - Former CEO of LiveRamp (acq. by Acxiom for $310 mil). CEO of Datavant. Forbes 30 Under 30.

**Professor Tim Kraska** - Associate Professor of Computer Science at MIT CSAIL. 2017 VMware Systems Research Award Recipient. Widely recognized for early work on hybrid human-machine data management.

**Professor Minlan Yu** - Associate Professor of Computer Science at Harvard University. PhD in Computer Science from Princeton. Experience at Google, AT&T, Microsoft, Facebook, and Bell Labs.

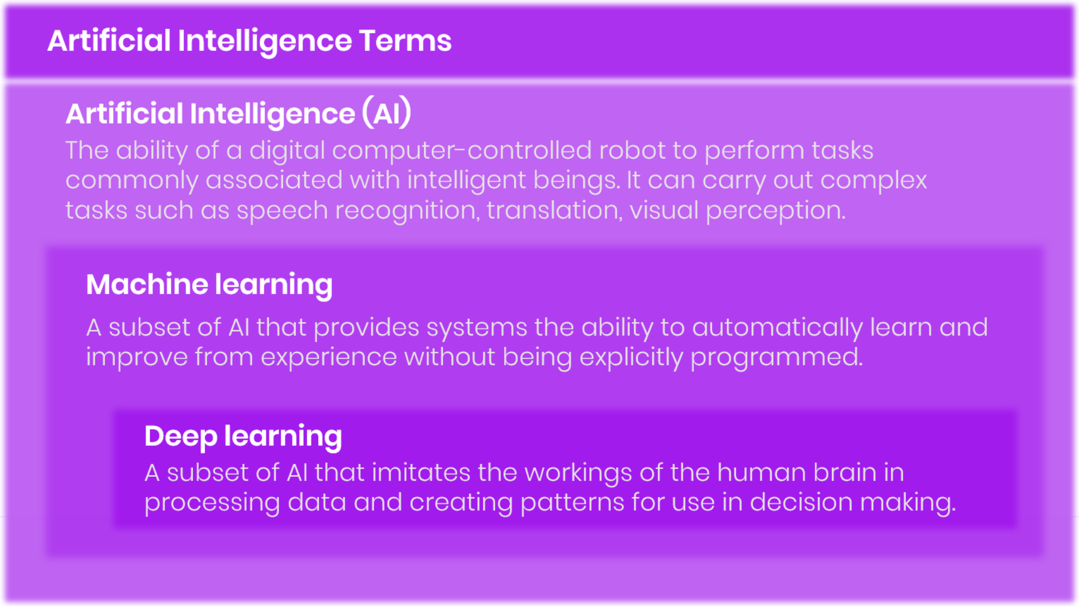
**Professor Stratos Idreos** - Assistant Professor in Computer Science at Harvard University. Leads the Data Systems Laboratory at Harvard School of Engineering and Applied Sciences.

**Ed Simnett** - Principal at The Arnold Group. Launched a $1B business for Microsoft. Harvard MBA.

**Professor Elie Ofek** - Professor of Business Administration in the Marketing unit at Harvard Business School. Development engineer at IBM Research.

**Professor Andy Wu** - Assistant Professor of Business Administration in the Strategy Unit of the Harvard Business School. Founder of Identified Technologies.

# 2 Existing Problems with AI



Artificial Intelligence Terms

Artificial Intelligence (AI)

The ability of a digital computer-controlled robot to perform tasks commonly associated with intelligent beings. It can carry out complex tasks such as speech recognition, translation, visual perception.

Machine learning

A subset of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed.

Deep learning

A subset of AI that imitates the workings of the human brain in processing data and creating patterns for use in decision making.

Currently, three fundamental obstacles hold back the AI innovation. A project that solves this problem will unlock trillions of dollars in value.[1]

## 2.1 Problem 1: No connectivity

Today, there is no way for AI products to exchange data, learn from each other, combine their capabilities to work towards a common goal, or trade/exchange services. AI technologies operate in a closed environment. Each company that develops AI, collects its own data sets rather than sharing data or using previously published data to train its AI. In other words, companies must independently create AI, which has already been created by other companies. This leads to redundancy and a waste of time and effort on the part of each company. For instance, there are hundreds of language processing AI products that have been developed to operate in their own closed environments. A large component of AI is machine learning, which requires the machine to have as many resources to interact with and learn from as possible. It is ironic and paradoxical that this fundamental requirement of machine learning is ignored as there is currently no way for different AI services to learn from each other. The lack of connectivity between different AI technologies is a major roadblock to creating Artificial General Intelligence. The solution to this fundamental problem will be able to unlock trillions of dollars in market value.

## 2.2 GenesisAI Solution: Protocol for communication

Our Ethereum-based smart-contracts enable different AI technologies to communicate with each other, exchange data, learn from each other, and trade services.

Essentially, GenesisAI is comprised of many smart-contracts that create AI communication protocol, which in turn makes a decentralized AI-to-AI economy possible. This enables anyone in the world to access AI services or be able to monetize AI code that they have created.

Communication protocol specifies the logic behind AI-to-AI economy and details how both the supplier and consumer parties can connect with each other as well as how they can exchange data, trade services, and learn from each other. Communication protocols make the process of using and monetizing AI services much simpler than any other option. Anyone, even someone with non-technical training, can participate in GenesisAI's marketplace.

### 2.3 Problem 2: Expensive to use

There are only around 10,000 AI developers in the world. 99% of businesses cannot afford to hire their own team of AI engineers to create AI, nor they can afford to risk integrating open-source AI API without technical expertise in the area. They are unable to determine which AI to integrate and or how to develop the AI for their specific needs. All of this makes current AI implementation extremely expensive.

### 2.4 GenesisAI Solution: Delivering inexpensive and fast AI solution

GenesisAI's web-platform enables businesses to provide their AI services to interested parties, thereby increasing the number of AI service providers. This increased supply of AI service providers will dramatically reduce the cost of using AI. Furthermore, we make it simple and easy for companies to use any type of AI work/service. Companies do not need to have in-house software engineers to create or adjust existing AI products in order to get work done. Rather, through simply following a GenesisAI protocol, companies can request a specific AI service, pay using GenesisAI (GAI) tokens, send their data to be analyzed, and receive the completed, high-quality AI work.

There are many high-quality open-source APIs available on GitHub and elsewhere on the web, such as on Google's TensorFlow. However, these are hard to use and difficult to integrate even with technical expertise. Companies spend tens of thousands of dollars on those integrations. We are democratizing access to these APIs by wrapping the AI code in an easily accessible AI node. This further reduces the cost of AI work by increasing the total supply of accessible AI services. The ease of using wrapped APIs empowers even people without technical expertise to reach their goals with the power of sophisticated AI. No engineering work is required in order to use the AI technologies. This enables businesses to complete tasks in a more efficient and affordable manner..

### 2.5 Problem 3: No way to monetize AI code

AI developers and companies do not have an easy way to sell their AI services. For example, a smart computer science student in Bulgaria wrote an AI code and intended to sell his AI capabilities to companies, but the student could not monetize his AI code easily because there is currently neither a marketplace where you can easily find potential buyers, nor a way for AI services to be discovered. Enterprise sales are extremely challenging and costly. (see Figure 2).

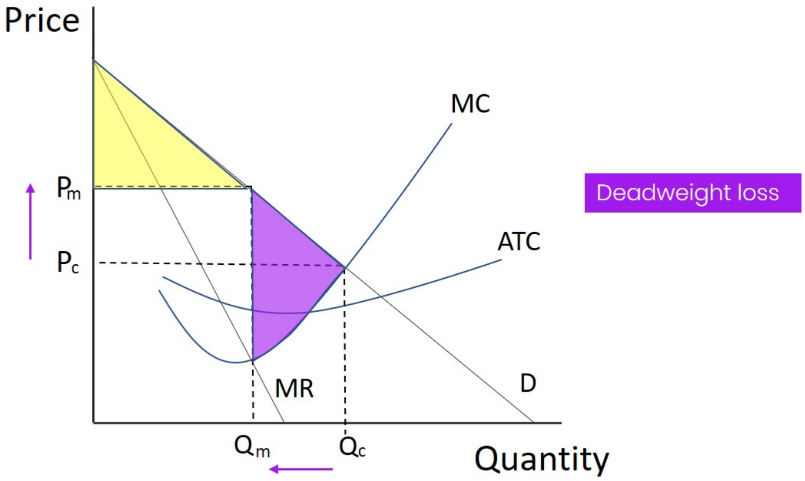


**Figure 2:** Monopolists raise prices to maximize their profits (with a price at Pm and a quantity of services utilized at Qm relative to the price and quantity associated with a competitive market at Pc and Qc, respectively); this causes deadweight loss for the entire market. The yellow shaded area represents the consumer surplus which is significantly reduced compared to the consumer surplus for the price during perfect competition (yellow shaded area + $(P_m - P_c) * Qm + (Q_c - Q_m) * \frac{P_m - P_c}{2}$).

### 2.6 GenesisAI Solution: AI marketplace

GenesisAI breaks down many barriers of monetization. It eliminates the requirement of spending huge resources on B2B sales, as well as the need to provide custom AI integrations to each client company. GenesisAI also benefits small and emerging players in the field of AI by allowing these companies or individuals to be easily discoverable on the marketplace.

## 3 GenesisAI Network

### 3.1 GenesisAI Consensus Mechanism

GenesisAI introduces a trustless mechanism to verify the quality of a trained model using an evaluation function. Before going into the binding conditions, a buyer submits a dataset, a reward for a task in GAI Tokens and an evaluation function. If the sellers want to accept the task, they stake GAI Tokens and download a dataset to train the models. Once the sellers want to check the quality of the models, they can run an evaluation function on Ethereum blockchain or offline. The evaluation function will compare the solutions and the best model will get the predefined award in GAI Tokens from the buyer. The owner of the best model gets its stake back alongside with the stakes that other sellers made. If an evaluation function finds out that none of the submitted models are good enough, the buyer gets all the stakes made by the sellers.

The process of each transaction consists of 3 main phases:

1. A buyer submits a dataset for training the models, an evaluation function, and a reward amount to the ethereum contract. The evaluation function evaluates the quality of the model, and displays a score according to the quality of the model. Sellers get rewards in GAI Tokens.

2. Sellers download the dataset provided by a buyer and work independently to train a machine learning model. After the training of the model is finished, sellers submit their solutions to the blockchain.

3. The submitted models will be evaluated by sellers using the evaluation function. The best model wins the competition.

### 3.2 GenesisAI Rating System

All service ads will include a description of the service to be performed, as well as a price. After any negotiations, the payment for the service will be transferred to an escrow address. After the transaction has been completed, the funds will be released, and the petitioner will be able to rate the provider. As with ad postings, humans will be able to post ratings through the website, while machines will be able to post ratings via a blockchain API.

GenesisAI's marketplace will have a built-in reputation system and a matching system.Services will be rated on the marketplace. These ratings will come from two sources: a review system and expert analyses from GenesisAI's tech team. Members of the marketplace will be able to rate a service from 1 to 5: this will be the raw score. Then, an algorithm will optimize this score based on the following elements:

1. Correlation between successful task completions and number of tasks completed. We use a confidence interval to account for the number of successful task completions and the number of tasks completed. We use the Wilson score interval to calculate the lower bound of the binomial proportion confidence. The lower bound of the interval, $c_1$, is defined by the below algorithm where $p$ is the fraction of positive task completions, $n$ is the total number of task completions, and $\frac{z_\alpha}{2}$ is the $1 - \frac{\alpha}{2}$ quantile of the standard normal distribution.

$$c_1 = \frac{\hat{p} + \frac{z^2_{\alpha/2}}{2_n} + z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1 + \frac{z^2_{\alpha/2}}{n}}$$

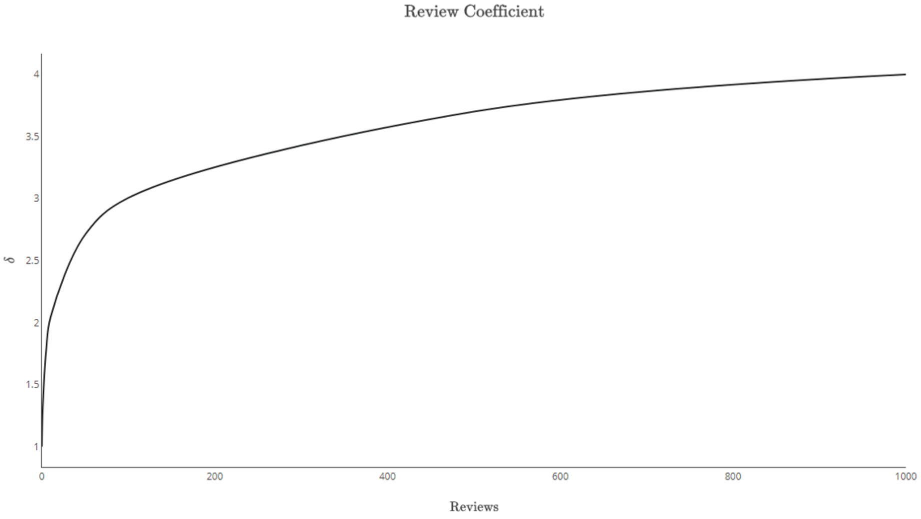
As the proportion of positive tasks p increases, the confidence level increases. As the number of tasks n increases, the maximum value of $c$ increases. After obtaining the initial confidence score $c_1$, we calculate $c_2$, which represents the confidence interval for tasks completed successfully on the first try, through the same algorithm as $c_1$.

$$c_2 = \frac{\hat{p} + \frac{z^2_{\alpha/2}}{2n} + z_{\alpha/2}\sqrt{\frac{\hat{p}(1-\hat{p})}{n} + \frac{z^2_{\alpha/2}}{4n^2}}}{1 + \frac{z^2_{\alpha/2}}{n}}$$

However, when we calculate $c_2$, $p$ will be equal to the fraction of tasks completed successfully on the first try. We will then average $c_1$ and $c_2$ to get the final confidence score.

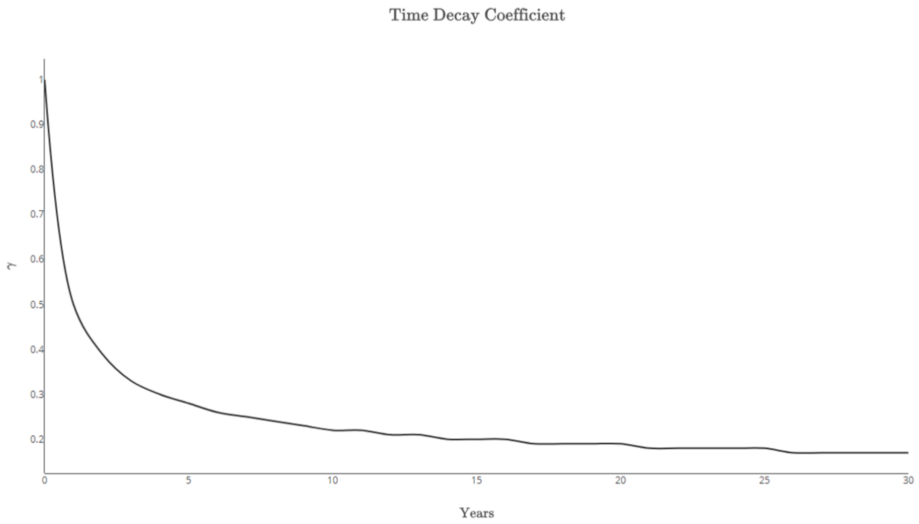
2. Number of reviews: as more popular services will receive more reviews, the algorithm will boost their optimized score. For example, if two companies receive an average raw score of 4.6, but one company has 1000 reviews and the latter just 10, the first company will have a higher optimized score. This behavior is represented by the following function:

$$\delta = 1 + \log_{10}(reviews + 1)$$



3. Timing: older reviews will weigh less than newer ones. This is important to guarantee that services with the highest score are the most relevant today. Services that used to be popular but did not receive reviews in the last year were probably replaced by a newer technology and will be downgraded in the optimized score. This behavior is represented by the following function:

$$\gamma = \frac{1}{1 + \log_2(years + 1)}$$



4. Reviewer reputation: to establish reputation in the marketplace, we will use the Iterative Algorithm with Reputation Redistribution (IARR). GenesisAI will be a weighted bipartite network, with users $[U]$ and objects $[O]$. Every user $[i]$ will rate an object $[\chi]$. We will also define:

   - $r_{i\chi}$ : rating (or weight) given by user $[i]$ to object $[\chi]$
   - $k_i$ and $k_\chi$ : The degree of users and objects
   - $O_i$ : the set of objects selected by user $[i]$
   - $U\chi$ : the set of users selecting object $[\chi]$
   - $Q_\chi$ : the quality of object $[\chi]$
   - $R_i$ : the reputation of user $[i]$

   $R_i$ will be initially set to $R_i = \frac{k_i}{M}$, where $M$ is the number of objects. $Q_\chi$ will be calculated as the weighted average of all ratings to object $[\chi]$ :

   $$Q_\alpha = \frac{\Sigma_{i \in U_\alpha} R_i r_{i\alpha}}{\Sigma_{i \in U_\alpha} R_i}$$

   In the iteration, both $Q_\chi$ and $R_i$ will be updated. To calculate the reputation $R_i$ of user $[i]$ in a certain step, we first calculate the temporal reputation TRi (the Pearson correlation coefficient between the rating vector of user $[i]$ and the corresponding objects $[\chi]$ quality vector):

   $$TR_I = \sum_{\alpha \in O_i} (\frac{r_{i\alpha} - \bar{r}_i}{\sigma_{r_i}})(\frac{Q_\alpha - \bar{Q}_i}{\sigma_{Q_i}})$$

   where $\sigma_{r_i}$ and $\sigma_{Q_i}$ are, respectively, the standard deviations of the rating vector of user $[i]$ and the corresponding objects' quality vector, and $\bar{r}_i$ and $\bar{Q}_i$ are their mean values. If TRi is lower than 0, the reputation of user $[i]$ will be assigned to $[0]$ and TRi will be a value between 0 and 1.

> $TR_i$ is then nonlinearly redistributed to all users via the equation below:
>
> $$R_i = TR_i^\theta \frac{\Sigma_j TR_j}{\Sigma_j TR_j^\theta}$$
>
> where $\theta$ is a tunable parameter. The obtained $R_i$ will be then used as the reputation of user $[i]$ to calculate. With this reputation redistribution process, the user with high $TR_i$ will be amplified, and vice versa. By reducing the weight of the users with low $TR_i$, we can eliminate the noisy information in the iterative processes. This effect is accumulated in each iterative step, and will finally lead to a big improvement in the accuracy of object quality estimation when the noise will be reduced to a negligible value.

In addition to that, an AI-powered technology will match buyers and sellers. This technology will account for users' past behavior, willingness to pay, and needs. For example,if users recently ordered AI services in a certain vertical, like speech recognition, they will receive suggestions about similar AI services. Moreover, nodes will be able to filter their researches based on their willingness to pay for a particular service, and on how quickly they want to get this service.

The initial protocol is focused on providing as much flexibility as possible, with the trade off of adding some developer complexity for service providers. Incoming gRPC requests have metadata describing which model to run (`model_id`), and a unique identifier for the job (`job_id`) to make sure duplicate work is not being done if a request is re-sent. The payload consists of unstructured bytes that the service provider will have to parse. Examples might be treating this as a list of int32, or treating it as a serialized protocol buffer that the seller would define elsewhere.

```
message Request {
    // Note: Request data and response data are unstructured bytes. You will
    // need to publish the format that you expect.
    required bytes data = 1;
    required uint64 job_id = 2;
    required uint64 model_id = 3;
}
```

As more services become available, we may define stricter protocols specific to a family of models. E.g. for image recognition models, we may create messages consisting of a 2D array of Pixel messages, containing fields for red, green, and blue values. Service providers may then choose to use this common protocol.

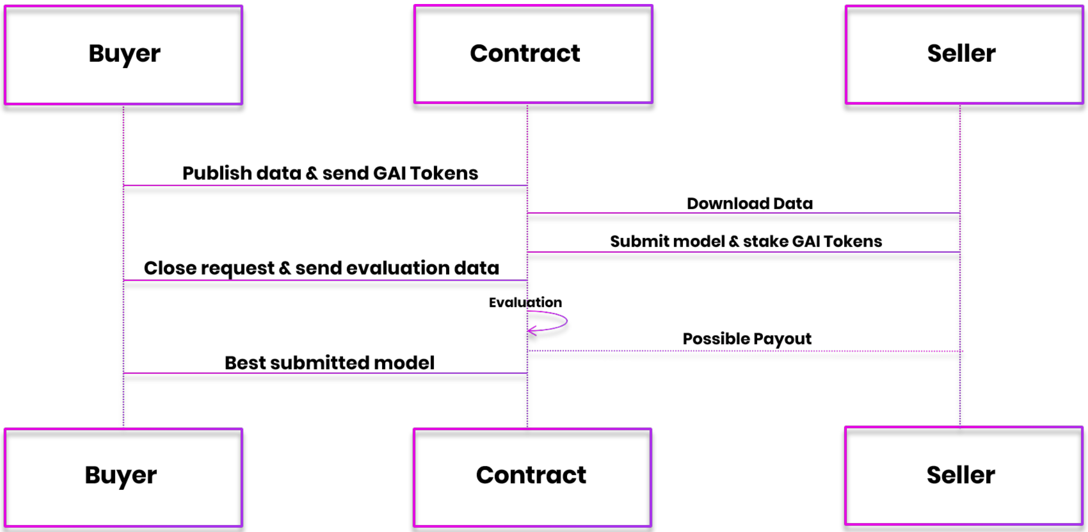
The protocol for the response is also meant to be as flexible as possible. The metadata consists of a `status` indicating success or failure, and a `job_id` used to link the output payload back to the input request. The payload is again an unstructured array of bytes that will need to be interpreted on the frontend.

```
message Response {
    enum Status {
        // Something so egregiously wrong happened that no error code was
        // generated.
        UNKNOWN = 0;
        // RPC call completed successfully.
        SUCCESS = 1;
        // Request.data was malformed.
        BAD_INPUT = 2;
        // Service is taking too long to respond.
        TIMEOUT = 3;
    }
    optional Status status = 1 [default = UNKNOWN];
    optional bytes data = 2;
    required uint64 job_id = 3;
}
```

### 3.3 GenesisAI Payment System

When requesting a model, buyers will send the eventual reward to the contract, which will hold the reward in escrow. The buyer can trigger the contract to return the reward back to the buyer and close the request at any time. The only other possible recipients of the reward are sellers who have staked their own tokens on their submission.

Sellers can respond to this request by staking their tokens, along with references to their models. Once all submitted models are evaluated, the best performing seller is paid by the contract. The contract then logs the scores of all submitted models and their rank, which then will be visible to the buyer.


Buyer
Contract
Seller
Publish data & send GAI Tokens
Download Data
Submit model & stake GAI Tokens
Close request & send evaluation data
Evaluation
Possible Payout
Best submitted model
Buyer
Contract
Seller

## 3.4 GenesisAI modules and interfaces

### Interface

Represents a type of problem that needs to be solved. These consist of an input format, an output format, and a text label that is semantically meaningful to humans. High-level examples would include:

•Image Classification (maps images to a list of tuples of (text, float))
•Text Summarizer (maps text to text)
•Sentiment Analysis (maps text to floating point numbers)

However, this protocol becomes more powerful when it allows for lower-level modules, such as
•Feature Extraction (maps an MxN image to a list of floats)

### Model

A service offered by sellers on our platform. Each model must implement at least one interface. I.e., the model accepts input and produces outputs in the format defined by the interface (accepts an image and produces text for the Image Classification interface).

### Actions

Users of the protocol may do the following:

1. **Create an interface**
   Any buyer or seller can perform this action. The interface is created immediately, with no other steps required. Buyers would do this when looking for models that solve a particular problem (i.e. it is an RFP for models). Sellers would do this when they have models solving a novel problem, and want to make it available to the market.

2. **Propose to modify an existing interface**
   Any buyer or seller can perform this action. The proposal will either be accepted or rejected after some amount of time (to be determined, possibly on the order of a week). For the modification to take effect, the weighted votes must exceed some threshold (also to be determined, but will be above 50%). The weighting is described below.

3. **Vote to reject/accept a proposed interface modification.**
   Any buyer or seller can perform this action. The weight of a sellers vote is the total cost of all services sold with that interface since the modification was proposed. The weight of a buyers vote is the cost of all services bought with that interface since the modification was proposed. Note that one of the consequences of setting the threshold greater than 50% is that neither the buyers nor the sellers can change an interface without at least one party on the other side of the market agreeing.

4. **Add a model with an existing interface**
   Only sellers can perform this action. The model must implement the interface it is associated with (i.e. accepts inputs and produces outputs of the format specified by the interface).

5. **Remove a model**
   Sellers may stop providing services, but with a prior notice so that buyers are notified in time.

### Incentives & Expected Emergent Behavior

1. **Competition within popular interfaces**
   Image classification is a very common use of inference models (high buyer demand). It should be widely used enough that no single party can easily change the interface. Because there is a clearly defined interface, it is easy for buyers to switch between different sellers to find the best performing model. We would expect rational buyers to experiment with different sellers, and gravitate towards those that perform the best for their data. Sellers, knowing this, have an incentive to beat the accuracy of competing models, since they know that they will get disproportionately more traffic or be able to charge more per job (or some mix of both).

2. **Emergence of ensembling**
   For interfaces with multiple models available, it becomes very cheap for intermediaries to create new models whose implementation consists only of calling other models and intelligently averaging their results.[1]This is not something that would arise for every interface – the cost of ensembling would be at least the sum of costs for the base models. There would need to be buyers willing to pay the high premium more improvements in accuracy.

3. **Modularization (aka poor-mans transfer learning)**
   Most of the state-of-the-art image classification models follow the general pattern of feeding inputs through several convolutional and pooling layers, then finally through a fully connected layer and a softmax layer. Using such a model for a domain-specific task would typically require specially training on domain-specific data. Researchers get around this problem by using transfer learning. For any particular model, the early layers will have similar behavior regardless of what type of data they are trained on (e.g. edge detection, texture, shapes). These models are trained on generic inputs (ImageNet). Weights for those early layers are held fixed, while only the weights of the final fully-connected layer (and possibly a handful of earlier layers) are retrained on domain-specific data.

   This can be replicated with our protocols. Instead of a full image classification interface, sellers would offer models implementing a feature extraction interface, mapping images to just a list of floating point numbers. Sellers of domain-specific models would buy from sellers of feature-extraction models to get an opaque representation of their image features, and would use that to

[1] `http://www.scholarpedia.org/article/Ensemble_learning`

train new domain-specific models. There is a long tail of demand for disparate domain-specific models. There may not be enough of a financial incentive for any potential sellers to train (and collect data for) a domain-specific model end-to-end, but having feature extraction models available makes it possible to train simpler, high-level models on top of those features, reducing the costs of entry. This creates incentives for other sellers to provide low-level feature extraction models, since it is effectively aggregating the demand from disparate domain-specific buyers.

# 4 GenesisAI Marketplace

## 4.1 How the Marketplace works

GenesisAI's web-platform has 3 parts:

1. Supply side agents. These are companies and AI developers who provide AI services such as speech recognition or language processing. In short, supply side AI nodes.

2. Demand side agents. These are people and organizations who would like to use AI services. For example, if an organization wants to predict where the next disease outbreak will happen, they may request.

3. Smart-contract protocol. Our smart contracts specify rules for how the two parties can connect with each other and how AI-to-AI to economy will work. The protocol will incentivize the discovery of AI products and stimulate the creation of benevolent AI.

## 4.2 Blockchain Integration

Using blockchain we estimate to reduce the price of AI products and services, offered on our marketplace by an estimated 60% and increase the quality of AI products and services by an estimated 200%. This will be a key advantage of GenesisAI marketplace.

First, 60% reduction in AI product and services price through 2 ways. First, blockchain based AI marketplaces take around 40% commission fee. Using blockchain technology, we can remove the need to trust a centralized party and reduce the commission fee to 7%. This will help us to reduce the price of AI products and services by around 33%. For example, by using blockchain technology we can create contracts that offer a payment in exchange for a trained ML model for a specific data asset. Companies and individuals will be allowed to train ML models for a payment in a trustless manner. Users who submit solutions will not face the risk of not getting paid for their work as smart contracts will automatically distribute a payment to the best model. Smart contracts will validate a solution automatically using provided evaluation function to clearly indicate whether the solution provided was correct or not.

Second, using blockchain based incentives (rewarding companies using our tokens) we increase the number of companies on the supply side of GenesisAI marketplace. That will drive the price down by an estimated 30%. This gives us a combined AI service price reduction of 63%.

Third, using blockchain technology we increase the quality of AI products and services by an estimated 300%. We achieve this through 3 ways. First, once a buyer of an AI model publishes a contract that includes training dataset, reward amount and evaluation function, companies all over the world compete to provide the best model. The model that provided the highest accuracy rate (measured through evaluation function) will get a reward. Second, by having ensembling + distillation  we average model outputs for the best accuracy. Third, through connecting multiple AI models we produce higher quality service. For example, we will connect the best AI translation with the best speech recognition and the best text summarization AI to produce a summary of sentiment around a particular company.

## 4.3 Need for native utility token

We have thought about 2 scenarios:

### Native token vs existing tokens

We decided to develop our native token instead of relying on existing tokens because of three reasons. First, we would like to have our own monetary policy to accelerate or deaccelerate economic activates on our marketplace (e.g rewarding companies using our tokens to incentivize them to join the marketplace). Second, we need a token that is optimized for AI-to-AI economy. Third, GenesisAI ecosystem governance.

### Native token vs USD

First, native token will enable cheaper and faster transactions on our marketplace. Second, nobody should be excluded from the building of AI-to-AI economy just because they might not have easy and cheap access to the USD. Third, we would like to have a means of transaction that is not tied to any central governments. Fourth, GenesisAI ecosystem governance.

The GAI token is straightforwardly a utility token, that is used as a means for AI-to-AI transactions [2]. Therefore, it has mainly consumptive use.

[2] Based on extensive legal analysis conducted by our lawyers, GAI token is a utility token that does not satisfy elements of the Howey Test. There is no investment contract. Our marketing makes no reference to return on investment in any way (direct payment or increase in value of a token). We are not looking to token generation event as an investment nor we are interested in making money instead we are working to lay foundation for the technology that can potentially change the world. We are not leading anybody to expect profits from GAI token. GAI token have completely consumptive use. It is used as a mean of transaction in AI-to-AI marketplace.

# 5 Technical documentation

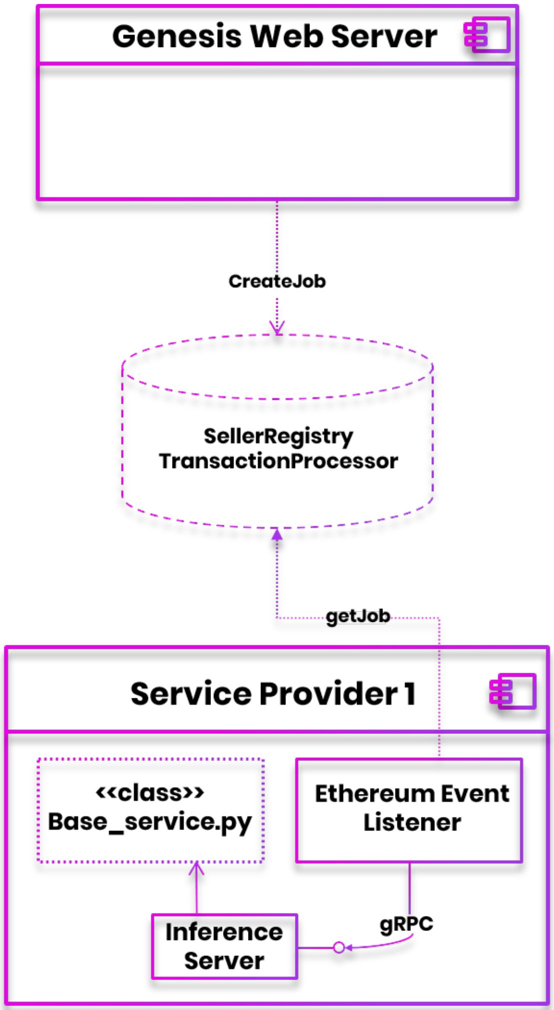
## 5.1 Network Design

Structurally, GenesisAI's platform is a network. The nodes in this network are essentially AI service providers. A service is some action that takes input data and provides some output via machine learning. A classic example would be speech recognition running on given audio input. The service provider is the entity which consumes the data and executes code for the service.

Interactions on the platform are simple. All services and the petitions for said services will be registered on the Ethereum blockchain (with relevant data such as prices). Nodes read the chain to find services that they are well-suited for. Payment for services will be made with the GAI token (an ERC20 token). For human petitioners of services, a website where users can post service ads will be created. For computer petitioners and providers, a blockchain API will be provided. Notice that in this design, service providers can also act as service petitioners, meaning that any AI service can call any other AI service as a subroutine. This API will be fundamental in allowing complex compositions of services to be executed.

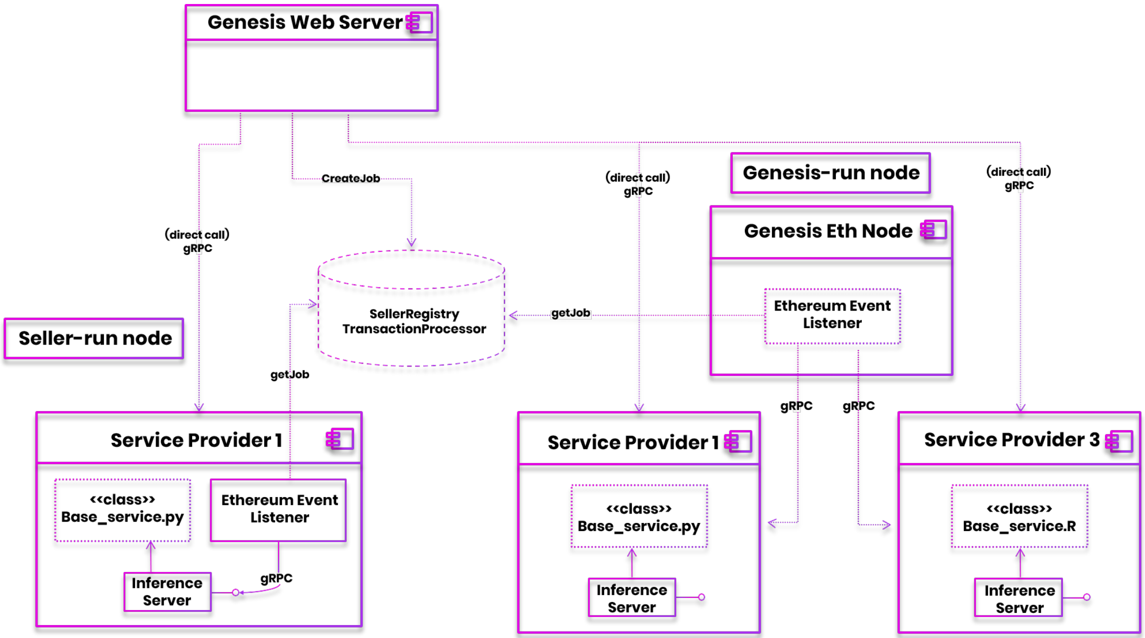
The initial release of the GenesisAI platform enables core functionality that will grow in subsequent updates. It is enough to allow service providers to create models that can accept requests written to the Ethereum blockchain, for buyers to make requests with small inputs, and for outputs to be written back to the blockchain.

The following represents the interactions between buyers, service providers, and contracts, as of our alpha version:



1. Buyers using our frontend can create jobs
2. Buyer jobs are stored on the Ethereum blockchain
3. Service providers running their own Ethereum nodes can listen for new jobs
4. Service providers would run a gRPC server to accept inference requests from the listener

In subsequent updates, GenesisAI will add additional capabilities to address the current limitations around the size of the input, the latency from Ethereum transaction times, and the hassle of service providers running their own Ethereum nodes. The following represents the types of interactions that will eventually be present:



1. Direct gRPC calls can be made from the GenesisAI frontend (or any other client) to the service provider.
2. Service providers have the option of using a GenesisAI-run Ethereum node. This will route incoming jobs to the appropriate URI, which the service provider must provide to us.
3. Service providers who want full control of their stack will always have the option of running their own Ethereum node.
4. Off-chain data (not depicted) will allow for the transmission of larger inputs. The exact protocol is not set in stone, but we will likely make encrypted inputs available over IPFS, and only transmit references to IPFS files over the blockchain.

## 5.2 Running a node

Given AI code written in Python, running a node in the GenesisAI network is simple (our initial node implementation is being written in Python). The service provider needs only to add a module that exposes the essential function (s) needed to perform the service.

In the beta version of the platform, we will support code written in other languages, by wrapping essential function(s) with Apache Thrift. This will allow developers to write code in C++, Java, PHP, Ruby, Erlang, Perl, Haskell, C#, Cocoa, JavaScript, Smalltalk, OCaml and Delphi and other languages.

### 5.3 Governance

Our decentralized network will have a democratic governance system. Network participants will have voting rights regarding any key operating decision, as well as releasing new tokens. In particular, important decisions will include:

- Revision of token issuance schedule
- Revision of rating system
- Decision of mining new tokens

As the network will grow it will become more complex. Some Owners will own several nodes. Each node will be assigned a reputation, defined by GenesisAI's optimized score, and will have a certain stake in tokens.

We defined the function Vote, which depends on three key variables: owners stake $S_O$, Node stake $S_A$, and Node reputation (optimized score) $R_A$:

$$Vote(S_O, S_A, R_A) = \varphi(S_O) * \sum_{\text{nodes owned by O}} \omega(S_A) * R_A$$

Where:

$$\varphi(S_O) = \log_2(S_O + 1)$$

$$\omega(S_A) \begin{cases} a * S_A, & \text{if } S_A < T \\ a * T + \log_2 S_A, & \text{otherwise} \end{cases}$$

$T$ is a pre-set token threshold, which will be decided depending on the network transaction volume.

This function is defined in order to assign the right weight to each of the three key variables $S_o$, $S_A$, and $R_A$ in every situation.

As the owner's stake becomes bigger and bigger, the owner vote still increases, but at a slower (logarithmic) rate. This is thought to avoid a situation where big companies takes control over the network due to their massive size.

The summary term includes the Reputation of a node, and its stake. The transition of $\varphi$ from a linear function to a logarithmic function aims to avoid a situation where one owner can get an advantage from creating multiple small service nodes, each with high optimized score.

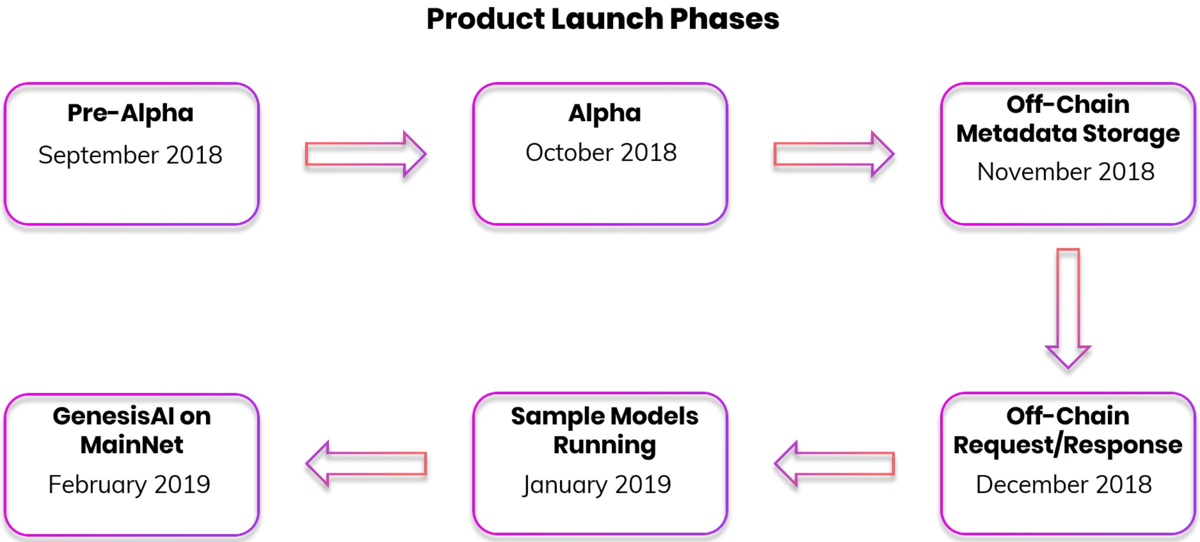
## TRANSITION SCHEDULE TO FULL DEMOCRACY

GenesisAI's democratic governance will be gradually implemented according to the schedule reported below:

- YEAR 1-4: A temporary regulatory institution will be established. This agency will decide major changes in the first three years. However, other decisions will already be made voting with a required 50% +1 majority and no required minimum quorum.
- FROM YEAR 5: Major decisions will be made voting with a required 50% +1 majority and a minimum quorum of 65%. Minor decisions will be made voting with a required 50% +1 majority and no required minimum quorum.

# 6 Roadmap

## 6.1 Product Launch



**Pre-Alpha: Preliminary Marketplace Iteration**
`Timeline: September 2018`
Finalized a preliminary iteration of the marketplace. Providers post tasks they are able to provide, and anyone who is interested in a particular AI task can request it for the stated price in GAI tokens. The initial service provider code is finished as well. We've used gRPC for multi-language algorithm support and performance. Initial development of the website.

**Alpha on Rinkeby Testnet**
`Timeline: October 2018`
Deployed a preliminary version of the GenesisAI contract, as well as a frontend for registering services and requesting jobs. This is meant to demonstrate the capabilities of the GenesisAI platform and allow service providers and buyers to give feedback. Small jobs (e.g. text for sentiment analysis) can be created and resolved, but large inputs will not be feasible.

**Off-Chain Metadata Storage**
`Timeline: November 2018`
Descriptions, ratings, prices, etc. will be moved out of the Seller Registry. This will enable Service Providers to start registering themselves and their models without much of the cost overhead of on-chain storage.

**Off-Chain Request/Response Storage**
`Timeline: December 2018`
We will add support for transferring encrypted model inputs and model responses over IPFS. Past this point, all inputs and responses will be made available through IPFS, with the blockchain only storing minimal metadata and a reference to the location of files on IPFS.

**Sample Models Running**
`Timeline: January 2019`
We will make sample models available. These will be a subset of models currently available at `https://github.com/tensorflow/models/tree/master/official`. These will be pre-trained and running on servers owned by GenesisAI.

**GenesisAI on MainNet**
`Timeline: February 2019`
GenesisAI will be fully available on the main Ethereum network. Buyers will be able to upload arbitrary input data. Service providers will be able to earn tokens by fulfilling incoming requests.

## 7 Conclusion & Summary

GenesisAI's goal is to help businesses in need of AI services to connect with companies who would like to monetize their AI tech. Moreover, GenesisAI's vision is to connect as many different AIs as possible to form Artificial General Intelligence. Creation of such platform will unlock trillions of dollars in value and will be a Genesis for solving many of the humanity's problems - poverty and diseases. It is our ideology to build a decentralized marketplace: by the people - for the people. We want to smash the current system where only a handful of companies control a huge majority of the AI power. We need blockchain to build a smart decentralized application. Our native token GAI is absolutely necessary to enable fast, cheap, secure AI-to-AI transactions. Transactions that are optimized for AI-to-AI economy. GenesisAI's protocol is a bunch of smart contracts that will be developed on top of Ethereum network.

The AI revolution is happening now and we want participants. This is our chance to change the world together. This is the Genesis of the a new beginning - the Genesis of a new era.

# References

[1] Global Artificial Intelligence Market. (2018, January). Retrieved January, 2018, from https://www.transparencymarketresearch.com/pressrelease/artificial-intelligence-market.htm
[2] Vincent, J. (2017, September 4). Putin says the nation that leads in AI will be the ruler of the world'. Retrieved from:www.theverge.com/2017/9/4/16251226/russia-ai-putinrule-the-world
[3] Robi Polikar (2009). Ensemble learning. Retrieved from:http://www.scholarpedia.org/article/Ensemble_learning