# Definitions

## Autonomous AI Oracle network

An autonomous AI oracle network refers to a decentralized and self-operating system that connects off-chain artificial intelligence services with on-chain smart contracts and blockchain networks. This system enables smart contracts to integrate AI capabilities and make complex decisions based on off-chain data inputs, processed by AI models. It revolutionizes the functionality of smart contracts by endowing them with advanced decision-making abilities, akin to an oracle in blockchain but specifically designed to handle AI computations.

## AIVM (Artificial Intelligence Virtual Machine)

The AIVM is a decentralized system architecture that serves as a uniform execution environment for AI model inference on the blockchain. It parallels the Ethereum Virtual Machine (EVM) in providing a standard set of rules and execution protocols that all nodes must follow. This ensures that the execution of AI models is consistent and secure across different nodes, leading to reliable AI computations within a trustless environment. The AIVM is adaptable to a wide range of AI models, offering flexibility and facilitating the integration of AI capabilities into a variety of applications.

## AI model inference queries

AI model inference queries are requests submitted to the blockchain network by users who seek to leverage AI capabilities for analytics or decision-making tasks. These queries trigger the process of AI model inference, where input data is fed into a trained AI model to generate predictions or insights. The results of these queries are reported back on the chain, providing users with AI-informed outputs that support various applications, such as predictive analysis or automated content generation.

## Bifurcated Inference Ledgering (BIL)

BIL is a two-phase transaction structure that Nesa implements to enhance the scalability and efficiency of decentralized AI computations. By decoupling the submission of an inference request from the actual execution of AI model inference, BIL streamlines the network's process, preventing computational loads from affecting blockchain performance. It ensures non-blocking transactions, maintains high throughput, and facilitates a flexible resource allocation, making it a vital component in the system's decentralized inference framework.

## Commit-Reveal Mechanism

A commit-reveal mechanism is a two-step process utilized to ensure honest and independent contributions from nodes executing AI inference tasks. In the Commit phase, nodes submit cryptographic commitments of their results, concealing the content while proving the existence of a computation. During the Reveal phase, the actual results and the nonce used in the commitment are disclosed for verification. This mechanism prevents nodes from free-riding on the work of others, thereby maintaining fairness and integrity in the decentralized inference process.

## Aggregation of Inference Results

The aggregation of inference results pertains to the process by which multiple outputs from different nodes are synthesized to produce a final, official outcome for an AI inference task. Nesa employs a default majority vote strategy within a smart contract to tally node submissions and determine the consensus result. Nodes that align with the majority are rewarded, while nodes with faulty submissions are penalized. This method reinforces the reliability and validity of the aggregated inference outcomes.

## AIVM kernel

The AIVM kernel is a complete package that includes the model parameters, configuration file, inference code, and aggregation code required for executing an AI model within the Artificial Intelligence Virtual Machine (AIVM). It encapsulates all necessary information and

logic for nodes to correctly execute an AI model. The kernel is stored on the blockchain to ensure transparency, immutability, and verifiability of the model's execution environment.

## Model Parameters

Model parameters are the set of weights and biases that characterize an AI model, essentially determining its behavior and prediction capabilities. These parameters result from the training process and dictate how the model processes input data to generate outputs. Within the AIVM system, model parameters play a vital role in achieving consistent execution and consensus on AI inference results among different nodes.

## AIVM Configuration File

Similar to a Dockerfile, the AIVM configuration file outlines the specifications for the virtual environment needed to run a model on the AIVM. It lists dependencies, libraries, and runtime details, ensuring that each node sets up an identical execution environment. This file is essential for maintaining the uniformity and reproducibility of model execution on the decentralized network.

## Inference Code

The inference code refers to the actual algorithmic logic executed by the AI model to process inputs and deliver predictions or other outputs. Combined with model parameters, the inference code conducts AI tasks within the AIVM and plays a crucial role in interpreting input data and producing consistent and accurate results for users' queries.

## Aggregation Code

The aggregation code in the AIVM ecosystem is a script that determines how results from different nodes are consolidated to reach a consensus within the decentralized virtual machine. It forms an integral part of the AIVM kernel and is vital in synthesizing outputs from various executions to provide a singular, verifiable outcome for AI tasks.

## Decentralized Storage Solutions

Decentralized Storage Solutions like InterPlanetary File System (IPFS) and Arweave are employed by Nesa to store AIVM kernels in a distributed manner. Such solutions offer a resilient platform for data storage that is resistant to censorship and data loss, ensuring the persistent availability and accessibility of AI models and execution environments.

## On-chain Model and AIVM Repository

The On-chain Model and AIVM Repository constitute a decentralized management system for storing and securing AI models along with their virtual machine configurations. It acts as a discoverable library that allows users to interact with a range of AI models and supports the responsible handling of private or proprietary models through encryption and key management processes.

## Privacy for Proprietary Models

Within Nesa's ecosystem, proprietary models are protected through encryption before being stored on decentralized platforms. This privacy protection ensures that confidential AI models remain inaccessible to unauthorized parties, with decryption keys controlled and distributed at the discretion of the model owner, fostering a secure and trustworthy AI marketplace.

## AIVM Interface for Model Interaction

The AIVM frontend interface is a user-facing platform that facilitates interaction with the on-chain AI model repository. It provides functionalities such as model browsing, uploading, deployment, and real-time monitoring—enabling users across various expertise levels to work with AI models effectively and with ease.

Decentralized inference delineates a process wherein AI computations are undertaken across a distributed network of nodes, ensuring a trustless environment where results are transparently reported on-chain. This method allows for AI models to be utilized in a decentralized way, maintaining user privacy and avoiding centralized points of control or bias.

## Inference Committee Selection

The Inference Committee Selection process involves deciding on a group of nodes (the committee) that will handle a particular AI inference task on Nesa. It uses VRF to ensure fair and random selection, as well as to verify the committee's composition, bolstering the security and impartiality of the decentralized inference system.

## Free-Riding Prevention

Free-riding prevention mechanisms are deployed to deter nodes from profiting from the efforts of others without contributing to the computation work. These mechanisms include a commit-reveal protocol that obligates nodes to demonstrate their own computations before gaining access to the results, promoting fairness and contribution within the decentralized inference network.

## Homomorphic Encryption (HE)

Homomorphic Encryption is a type of encryption that allows for computations on encrypted data without requiring access to the decryption key. Nesa's hybrid-privacy system considers the use of HE techniques for carrying out secure computations on private user data within the TEEs of its decentralized AI platform, strengthening the privacy-preserving capabilities of the project.

## Secure Multi-Party Computation (SMPC)

SMPC is leveraged in Nesa's infrastructure for executing secure computations among multiple parties where the data inputs remain private. This cryptographic protocol is valuable for collaborative tasks in a decentralized environment where privacy is of paramount importance, without having to compromise the integrity of the computations.

## Zero-Knowledge Proof (ZKP)

ZKP is a cryptographic technique used by Nesa to enable the verification of the correctness of computations without exposing the underlying data. It helps maintain confidentiality while providing proof that the system's computations are accurate, encouraging trust and security in the decentralized inference system.

## Threshold Cryptosystem

A threshold cryptosystem is utilized by Nesa for the secure distributed management of cryptographic keys. In such a system, a secret key is split into multiple shares, with no single party having access to the entire key. Nesa employs this system to ensure secure collaborative decryption during AI model inference tasks, without exposing sensitive data.

## Large Language Models (LLMs)

Large Language Models, such as those developed by OpenAI (e.g., GPT), are AI algorithms designed to process and generate human language in a context-aware and nuanced manner. Nesa uses an enhanced version of LLM, optimized for the Web3 environment, to provide smart contract interactions with access to sophisticated language-processing capabilities.

## $NES

The native cryptocurrency token used within the Nesa ecosystem. $NES serves various functions such as paying for transaction fees, staking for network security, and participating in governance decisions. It underpins the economic model of Nesa, integrating economic incentives with network operations for AI model inference.

## AIVM Kernel Marketplace

The AIVM Kernel Marketplace envisaged by Nesa represents a future platform where containerized AI models can be publicly traded and monetized. This open marketplace aims to empower creators with the ability to own, update, and sell their AI iterations, promoting a robust ecosystem for AI model evolution and interaction in a decentralized fashion.

## Threshold Decryption

Threshold decryption is a cryptographic technique used to decrypt information in a distributed manner where no single participant can reconstruct the complete decrypted data. This method is integral to Nesa's privacy-preserving system and is a key component

in securely handling inference on encrypted data without disclosing it to any unauthorized entities.

## Utility Suite

A collection of tools and services from external Web3-AI partners provided through Nesa, tailored for various stages of AI development. These include computational resources like TPUs and GPUs, model databases, APIs, and other tools necessary for training AI, such as data oracles and governance platforms, which aid in the smooth and efficient evolution of AI systems.

## DNA X Platform

DNA X is Nesa's inaugural dApp centered on creating, interacting with, and monetizing autonomous digital personalities, designated as DNA's. These DNA's leverage the capabilities of the Helix-1 model to provide life-like and evolving interactions with users, creating a unique ecosystem of digital beings on-chain, minted as NFTs, and evolving their personalities through interactions within the DNA X platform.

## NANs

NANs are an AI-based jury system within the Nesa infrastructure that evaluates the reliability and performance of kernels by administering a series of tests designed to simulate various operating conditions and uncover any inconsistencies in execution. NANs concentrate on key performance indicators which may include computational efficiency, ethical adherence, data precision, susceptibility to generate erroneous or creative outputs, and compliance with established model protocols. They employ an adversarial evaluation framework to challenge the kernels in unpredictable ways, ensuring they behave consistently and according to the predefined configurations before being cataloged on the blockchain.

## Split-Flow

The Split-Flow Protocol dynamically partitions tasks into confidentiality and verifiability streams that operate using either TEEs for heavyweight operations, cryptographic methods

like SMPC for lightweight tasks, or a hybrid of the composites depending on requirements. ZKPs are employed within the verifiability stream to ensure the accuracy of computations without compromising data privacy. The protocol's ability to adjust in real time to the specifics of computational tasks makes it particularly effective in balancing security, efficiency, and verifiability in Nesa's decentralized inference systems.

## Model Version Control

Model Version Control is a process and system within Nesa's decentralized AI framework that enables the tracking and management of different versions of AI models and corresponding datasets. It resembles version control systems in software development, like Git for AI training processes, and ensures all model updates are duly logged and managed.

## On-chain Fork Management

Nesa's on-chain fork management refers to the protocols established to handle branches in AI model evolution. It ensures that as models fork and merge in their development paths, these changes are accurately reflected on-chain. This system allows for complex evolutionary processes, including adaptation and experimentation on AI model updates.

## TEE (Trusted Execution Environment) Implementations

Nesa relies on various TEE implementations, including Intel's Trusted Domain Extensions (TDX), AMD's Secure Encrypted Virtualization with Secure Nested Paging (SEV-SNP), ARM's Confidential Compute Architecture (CCA), and NVIDIA's Hopper H100 GPU architecture. Each provides an additional layer of security and privacy, allowing confidential computations to occur within the Nesa ecosystem.

## SMPC (Secure Multi-Party Computation)

SMPC on Nesa is utilized to execute confidential computations collaboratively, ensuring participating parties cannot access others' data input. The implementation is optimized for performance, allowing the network to facilitate secure interactions without compromising efficiency.

## VRF (Verifiable Random Function)

Nesa uses VRF to provide unbiased, unpredictable, and verifiable random values for network processes such as node selection. It ensures that all elements of randomness within the system are fair and transparent, fortifying the security and integrity of the decentralized AI platform.

## Proof-of-Stake Consensus

Nesa operates a Proof-of-Stake (PoS) consensus mechanism, where network validators and their delegators secure and maintain the blockchain by staking $NES tokens. This consensus model provides network governance and collaborative decision-making, contributing to the platform’s democratization and distributed trust.

## Dynamic Pricing Model

Nesa's dynamic pricing model refers to the adaptable fee structure for AI model inference queries. It aligns resource allocation with market demand, enabling users to prioritize their requests by paying higher fees. This model ensures efficient system utilization and optimizes network throughput by managing the queuing of inference tasks based on variable pricing.

## Erasure Coding and Merkelization

Nesa employs erasure coding and Merkelization techniques for data redundancy and security, ensuring that AI model data remains intact and tamper-proof as it is stored and processed across its decentralized network.

## Model Evolution Attestation

Model evolution attestation is a verification process within Nesa that ensures the integrity of AI model updates. Validators within the network attest to the authenticity and proper execution of model evolutions, providing a trustless environment for model development and deployment.

## Namespace-specific Storage Management

Nesa uses a unique storage management system that allocates space for models and data within specific namespaces. This allows for organized and scalable storage solutions tailored to the needs of different AI models and their associated datasets.

## queryStream

QueryStream is a protocol function within Nesa designed to facilitate the on-chain querying of AI models by streaming verification layer data to Nesa's immutable ledger or another blockchain like Ethereum. This method ensures the data's integrity through a peer-to-peer network and smart contracts that execute model evolutions end-to-end. QueryStream's P2P relayer component serves as a bridge, streaming query results to the relevant blockchain's settlement layer for final confirmation and execution.

## Byzantine Fault Tolerant (BFT) Consensus Mechanism

Nesa's consensus mechanism ensures that even in the presence of faulty or malicious nodes (byzantine failures), the network can reach consensus and operate correctly. This enhances the network's security and resilience.

## Validator Nodes

Nodes on a blockchain network with the responsibility to validate transactions, produce new blocks, and maintain the integrity of the network. Validators participate in the network's consensus mechanism and, in some blockchains, are required to stake cryptocurrency as a form of security and commitment to their role.

## Gaussian Kernel Transformation

Nesa's Helix-1 model incorporates Gaussian Kernel transformations to effectively manage the linguistic style and syntax adaptation, enabling the AI to emulate human-like language patterns within the NES X platform interactions.

## Intrinsic Rank in AI

Nesa's LLMs utilize the concept of intrinsic rank which postulates that the actual number of parameters needed to perform a task effectively is often much lower than the size of the model itself, allowing for efficient adaptations with minimal parameter changes.

## PayForQuery

PayForQuery is a transaction type on Nesa that enables developers to pay for the querying of AI models, with the cost being determined by the complexity and resources needed for the transaction. It capitalizes on Nesa's Byzantine Fault Tolerant signature mechanism to ensure the validity and security of the AI model data queries on the chain. The PayForQuery transactions are essential for the verifiable execution of AI model inferences, allowing for cryptographic validation and submission of data to Nesa or interchain ecosystems like Ethereum.

## Gumbel Softmax Reparameterization

The Gumbel Softmax approach facilitates differentiable sampling in AI model training and is used in Nesa to introduce randomness while maintaining differentiability during model training and execution. The refactoring of the sampling process does not make it differentiable.

## Ancestral Sampling

Ancestral sampling is a stochastic process used by Nesa's LLMs to generate sequences of predictions. It’s used to sample from a distribution over sequences, enabling realistic and coherent text generation by sampling each token in the sequence conditionally based on previously generated tokens.

## DNA

A DNA (Decentralized Neural Application) refers to a one-of-one NFT minted on DNA X and running on Nesa that represents a digital being on-chain. A DNA has its own AIVM Kernel on

Nesa. Each DNA has its own set of characteristics analogous to the genetic composition found in biology, making each configuration as distinctive as an individual's DNA. A DNA runs on fully decentralized querying on the Nesa network.

Previous
Zero-Knowledge Proof (ZKP)

Last updated 1 month ago