

Nesa's Utility Suite

Nesa accessorizes its system with a hub for external AI tools that help facilitate model querying, training, upload, and evolution (Figure 11.2). As the point of final validation for any model inference query, Nesa sits at the very bottom of the stack and is therefore able to plug an assortment of third-party AI services on top of it.

These services are serially executed and consigned to a dedicated adapter within the system. The Utility Suite includes provisioned computational power (TPUs, GPUs), almost all major publicly available open-source models, the major API-based LLMs, an assortment of Information Oracles, DAO tooling, and Relayers.

Nesa will continue to grow its collection of AI-Web3 partner services and products that can be invoked during training, upload, orchestration, and inference before Nesa executes final query validation.

Nesa Protocol

Charters

OPERATING SYSTEM

COMPILE VERSIONS

HARDWARE SPECS

AIVM Configuration

PRECISE COMPILATION OPTIONS & FLAGS

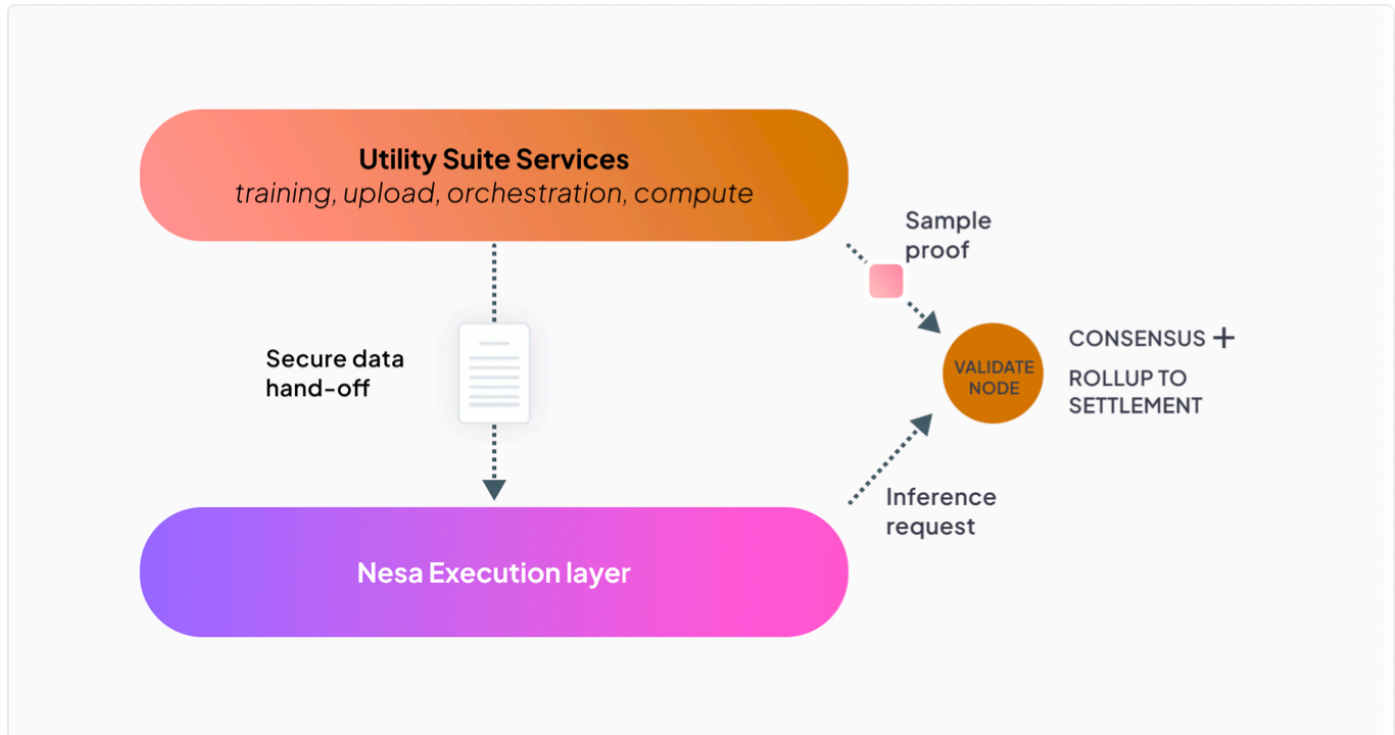
EXECUTION ENVIRONMENT

INITIALIZATION PROCEDURE

DATA INPUT & OUTPUT CONVENTIONS

AIVM specification and configuration. Details are provided in AIVM kernels in order to guarantee consistency and inference reliability in the AIVM, including operating system, compiler, hardware specifications, initialization procedures, the inference code and aggregation code, data input and output conventions, and other precise compilation options and flags for execution. The execution protocol within the AIVM prescribes the exact series of steps that every node must follow. By standardizing the execution flow, we can reliably predict and replicate the behavior of AI models across the network. If a model demands particular hardware characteristics, such as GPU acceleration or specialized processing units like TPUs, these requirements are explicitly stated in the AIVM configurations. Moreover, features provided by the hardware that could potentially lead to inconsistent execution, such as non-deterministic hardware instructions, are either strictly enabled or disabled as appropriate. Developers are permitted to customize any aspect of the AIVM configuration file including requested hardware for inference and specifics of the query response. Before an AIVM kernel is

approved and stored on the blockchain, it undergoes rigorous validation to ensure compliance with the specified configuration, whether preset by default or customized by the model's developer.



Nesa Utility Suite Architecture. Nesa connects a hub of utility services that help facilitate model querying, training, upload, and evolution for AI model developers in the Nesa ecosystem. A utility service is initialized, run, and then securely hands off data to Nesa's Execution Layer to assist in a component of the preparation, orchestration, or computation of the request. Inference is then conducted by Nesa Core and the results are sent to validator nodes for consensus before rollup to settlement.

Some examples of Nesa's future Utility Suite include:

1. Efficient sharing of computational resources on-chain, such as Render, Akash.
2. Decentralized AI tooling and infrastructure services, such as Olas, Bittensor.
3. Information oracles for off-chain knowledge relayance, such as Chainlink, Band.
4. DAO tooling management systems, such as Gnosis, Aragon

Previous
Dynamic Model Versioning and Fork Management

Next
Interoperability and AIVM's Future Plans

Last updated 1 month ago