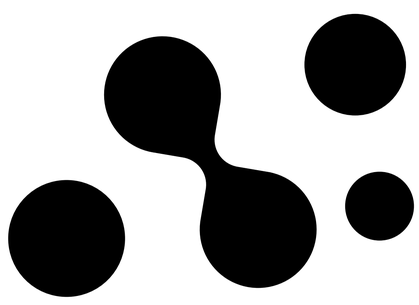


APUS NETWORK

Enabling Verifiable Decentralized AI through Deterministic GPU Computing

VERIFIABLE AI INFERENCE

Litepaper

V.1

November 2024

1. Background

1.1 AI: From Nobel Moment to Centralization Concerns

With declarations of AI's "Nobel moment" and recent Nobel Prizes in physics and chemistry being considered its "coming out party," there is renewed hope that AI is poised to shape history.[1] As one of the most revolutionary technologies of this century, AI is projected to develop into a $15 trillion industry by 2030,[2] with the promise of significantly enhancing human productivity.

However, alongside this immense potential lies a looming shadow cast by centralization: AI-aided state surveillance, control of the technology in the hands of too few firms, data privacy, algorithmic transparency, and resulting income and societal inequalities.

1.2 Decentralized AI: Breaking Free from Centralized Control

Decentralized AI is a paradigm shift that leverages blockchain technology to redistribute the ownership and governance of AI systems. It aims to transform AI development from walled gardens into open ecosystems.

The technology promises to democratize access to AI resources, enabling independent developers to build and monetize their work, while making AI investment opportunities available to the public rather than just venture capitalists.

However, this decentralized vision faces a critical security challenge: how can we guarantee the integrity of AI computations when they're executed across a network of anonymous participants? Without robust verification mechanisms, malicious nodes could compromise model outputs, or expose sensitive data - effectively undermining the entire decentralized AI ecosystem.

The solution to this challenge lies in verifiable AI inference, which can be achieved through either complex verification protocols or, more efficiently, through deterministic computation.

1.3 Apus Network: Pioneering Verifiable AI Inference through Deterministic Computation

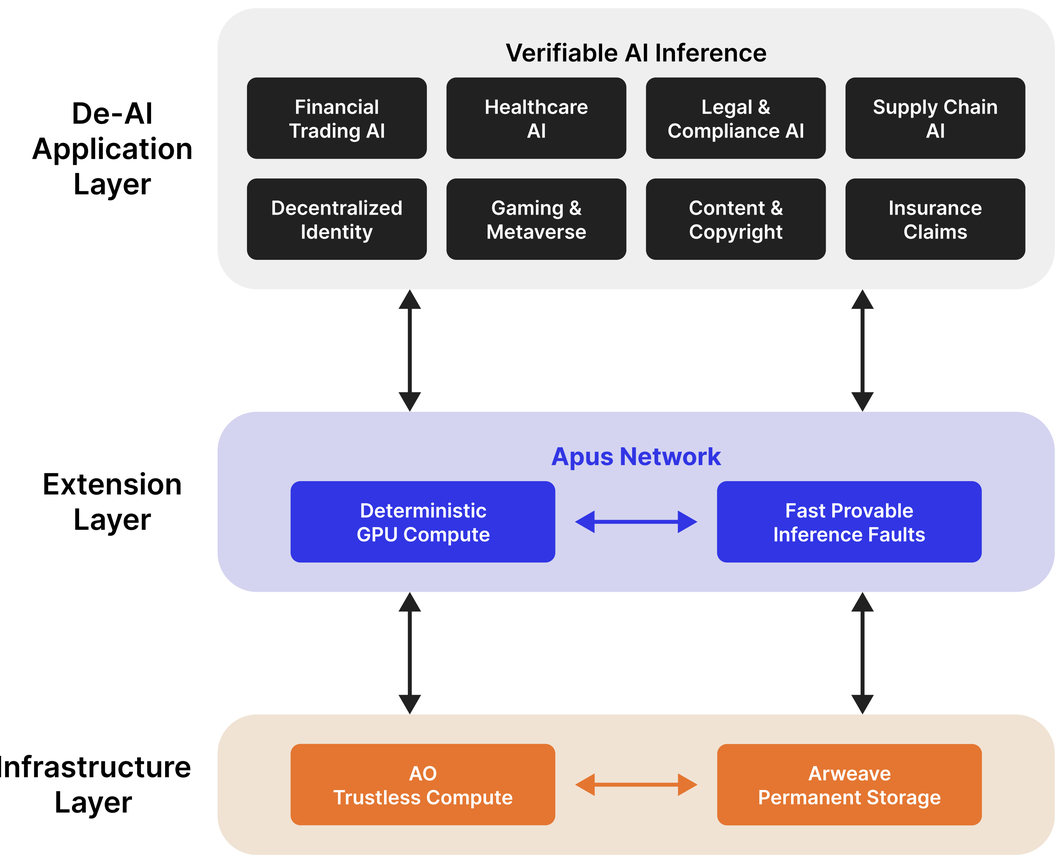
Among various approaches to verifiable AI inference, deterministic computation stands out as the most efficient solution, eliminating the need for complex verification and consensus mechanisms that would otherwise increase computational costs significantly.

Apus Network advances this vision through two core innovations: deterministic GPU computation and **Fast Provable Inference Faults** (**FPIF**) - a lightweight verification protocol specifically designed for deterministic environments.

By integrating these innovations with Arweave's [3] immutable permanent storage and AO's [4] decentralized trustless computing infrastructure, Apus Network creates a robust framework for verifiable AI inference. This powerful synergy delivers multiple benefits:

- Ensures computational integrity through deterministic execution
- Enables fast verifiable via **FPIF**'s streamlined protocol
- Prevents unauthorized model manipulation
- Maintains permanent, tamper-proof records of AI operations

This comprehensive approach positions Apus Network at the forefront of decentralized AI, driving the development of more transparent, fair, and reliable AI systems.



2. Verifiable AI Inference

2.1 Introduction

So what's verifiable AI Inference? Verifiable AI inference allows independent verification of an AI computation's correctness based on three fundamental components:

- **Input Integrity:** Ensuring the correct input data was used
- **Model Integrity:** Confirming the authentic model was employed
- **Compute Integrity:** Verifying the computation was performed correctly

In essence, achieving verifiable AI inference can be expressed through the fundamental equation:

Verifiable AI Inference =
Verification Mechanism(Input Integrity + Model Integrity + Compute Integrity)

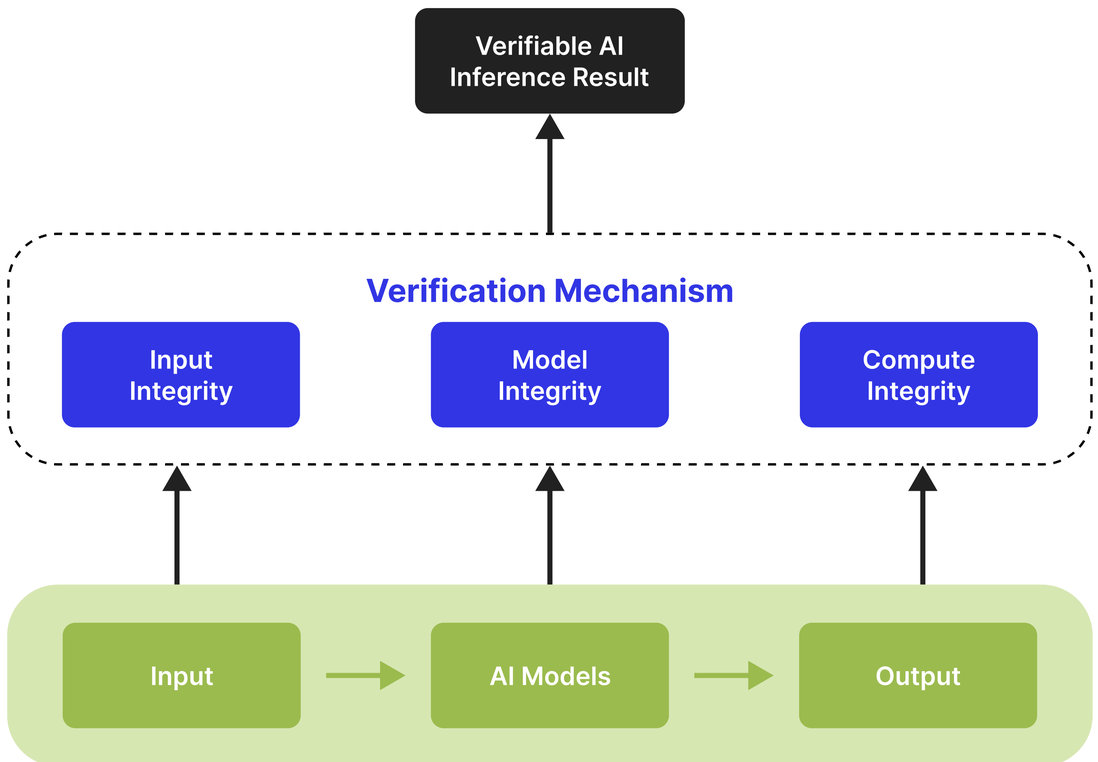
The complexity of this verification mechanism varies dramatically depending on whether the computation is deterministic or non-deterministic.

In deterministic computations, where the same inputs and model always produce identical outputs, the

verification mechanism becomes remarkably simple - verifiers can just rerun the computation and compare results.

However, in non-deterministic computations (like most modern decentralized AI systems), where the same inputs and model may produce slightly different but valid outputs, complex verification protocols become necessary to validate the computation process.

While deterministic computations simplify verification through re-computation, both deterministic and non-deterministic systems still require robust verification mechanisms to ensure trustworthy results.



2.2 Verification Mechanism

Currently, three main approaches dominate the verification mechanisms' landscape: Zero-Knowledge Proofs (ZK ML), Optimistic Fraud Proofs (OP ML) and Cryptoeconomics (Cryptoeconomic ML) approaches.

[5] However, this paper introduces a novel verification mechanism called Fast Provable Inference Faults (FPIF ML), which integrates the strengths of existing approaches while mitigating their drawbacks, thereby offering an optimal balance between security, cost, and practicality.

Let's delve into the details of each mechanism and explore their unique trade-offs.

2.2.1 Zero-Knowledge Proofs (ZK ML)

ZK ML represents the cutting edge of cryptographic verification, offering mathematical guarantees through ZK-SNARKs. The approach generates simple proofs that are easily verifiable yet computationally intensive to generate:

Core Mechanism:

- Neural networks are compiled into zero-knowledge circuits
- Proofs are generated to verify computation correctness
- Verification can be performed without accessing the original model

Key Advantages:

- Cryptographic guarantee of computational correctness
- Impossible for model providers to cheat
- No trust assumptions required
- Fixed verification costs regardless of model size

Technical Challenges:

- ~1000x increase in computation costs [6]
- ~1000x increase in latency for proof generation [6]
- Significant overhead for circuit compilation
- High implementation complexity

2.2.2 Optimistic Fraud Proofs (OP ML)

OP ML adopts a "trust but verify" approach, assuming computations are correct unless proven otherwise through challenge-response mechanisms:

Core Mechanism:

- Nodes perform computations and post results
- Watchers monitor the network for incorrect results
- Challenge period allows for fraud proof submission
- Interactive challenge-response game resolves disputes

Key Advantages:

- Lower operational costs compared to ZK ML
- Security guaranteed with single honest watcher
- No complex cryptographic requirements
- Practical for immediate implementation

Security Considerations:

- Requires active network watchers
- Challenge period introduces result finality delay
- Complex challenge-response game implementation

2.2.3 Crypto-economics (CE ML)

Crypto-economic ML takes a pragmatic approach by leveraging fundamental economic incentives rather than complex cryptographic proofs. The mechanism is elegantly simple:

Core Mechanism:

- Nodes stake collateral to participate in the network.
- Users specify their desired security level by selecting the number of verifying nodes.
- Multiple nodes perform computations independently and reveal their results.
- Consensus determines correctness; dissenting nodes face slashing penalties.

Key Advantages:

- **Flexible security-cost tradeoff**: Users can choose between n=1 (minimal cost) to n=all_nodes (maximum security)
- **Low latency:** Only requires commit-reveal from participating nodes
- **Simple implementation:** Similar to established oracle mechanisms
- **Scalable architecture:** Easy to add or remove computing nodes

Security Considerations:

- Security correlates directly with economic stakes
- Vulnerable to theoretical collusion if majority of nodes cooperate
- Can be strengthened through mechanisms like Eigenlayer restaking
- Potential for enhanced security through query encryption and intent-based payments

2.2.4 Fast Provable Inference Faults (FPIF ML)

FPIF ML introduces a streamlined verification mechanism uniquely tailored to deterministic computation environments such as AO and Arweave. By leveraging deterministic executions, cryptographic attestations and staking/slashing economic mechanism, **FPIF** ML guarantees the integrity and correctness of computations without the need for complex fraud proofs or extensive economic staking frameworks.

Core Mechanism:

- **Deterministic computations:** AO processes employ deterministic computations that ensures given identical inputs and environments, the computation will always yield the same result.
- **Signed attestation:** After performing computations, Compute Units (CUs) generate signed attestations that encapsulate the results, including the updated state and any resulting outbound messages.
- **Public Verification:** These attestations are publicly accessible, allowing any network participant to independently verify the correctness and reliability of the computation without additional complex mechanisms.
- **Slashing Mechanism:** If a CU provides incorrect computations or fails to perform as expected, their staked tokens are subject to slashing penalties, disincentivizing malicious behavior.
- **Rewards for Integrity:** Correct computations and valid attestations result in rewards for CUs, aligning their interests with network integrity and performance.

Key Advantages:

- Simplicity and Efficiency:

- **Direct Verification:** Verification resembles signature validation, eliminating the need for complex proof generation or extensive consensus processes.
- **Low Latency:** Attestations can be verified quickly, enabling fast confirmation of computational results.

- Cost-Effective:

- **Scalable Verification:** Each attestation is independent, allowing the system to scale without increased complexity.
- **Direct Verification:** Verification resembles signature validation, eliminating the need for complex proof generation or extensive consensus processes.

- Enhanced Security:

- **Immutable Logging:** Integration with Arweave ensures that all message logs and attestations are stored immutably, preventing tampering and ensuring long-term data integrity.
- **Economic Deterrents:** Slashing mechanisms provide strong economic deterrents against malicious actions, ensuring that CUs have a vested interest in maintaining computational integrity.

- Technical Challenges:

- **Heterogeneous GPU Hardware:** Ensuring deterministic computations across diverse GPU architectures and configurations presents significant complexity.
- **Privacy Protection Limitations:** Unlike ZK ML solutions, FPIF ML does not inherently provide strong privacy guarantees. While it ensures the correctness of computations through signed attestations, it does not obfuscate the data or model details involved in the computation. Several approaches could address these privacy concerns:
  - **FHE (Fully Homomorphic Encryption)** could be a method to solve privacy issues, though it is still in early practical stages due to significant performance overhead.
  - **TEE (Trusted Execution Environment):** Provides hardware-level isolation and protection for sensitive computations and data. TEE solutions like Intel SGX or AMD SEV can offer a good balance between privacy protection and performance, allowing secure execution of AI inference while keeping both input data and model parameters confidential.

2.2.5 Fast Provable Inference Faults (FPIF ML)

Verifiable AI Inference Mechanism	Security Level	Verification Speed	Cost Efficiency	Scalability	Projects
ZK ML	Extremely Secure	Fast	Very Low	Limited	EZKL, Giza, Modulus
OP ML	Conditionally Secure	Slow	Medium	Moderate	Ora, Gensyn
CE ML	Conditionally Secure	Slow	Medium	Moderate	Ritual, Atoma
FPIF ML	**Highly Secure**	**Fast**	**High**	**High**	**Apus Network**

In comparing various verifiable AI inference mechanisms, each offer distinct trade-offs:

- **ZK ML** provides the highest level of security through zero-knowledge proofs, but comes with extreme computational costs and limited scalability.

- **OP ML** and **CE ML** offer conditionally secure solutions with moderate costs and scalability, but suffer from slower verification speeds.

FPIF ML emerges as a breakthrough by delivering:

1. Fast verification speed
2. High security through deterministic computation
3. Low cost efficiency
4. High scalability

Apus Network is building a **Deterministic GPU Extension** for **FPIF** ML based on the AO protocol and Arweave storage. How can it ensure verifiable AI inference with Arweave and AO, and how does Apus Network address the deterministic GPU technical challenges?

3. Our Solution

3.1 Ensuring Verifiable AI Inference with Arweave and AO

To demonstrate that Verifiable AI Inference can be achieved within the AO Protocol, integrated with Arweave, it is essential to analyze how the following key elements are upheld: Input Integrity, Model Integrity, Compute Integrity, and the application of Fast Provable Inference Faults (**FPIF**). Leveraging the unique features of Arweave and AO **[4]**, each of these components is meticulously addressed to ensure a trustworthy and reliable AI inference environment.

3.1.1 Input Integrity

Input Integrity ensures that the data fed into the AI inference process is authentic, unaltered, and free from tampering or corruption.

- Immutable Message Logging:

- In AO, interactions with any process occur through **Messages**, which are assigned unique, incrementing sequence numbers by **Scheduler Units** (**SUs**).
- These messages are stored on Arweave, a decentralized and immutable data storage layer, ensuring that once a message is recorded, it cannot be modified or deleted.

- Digital Signatures and Verification:

- Each message adheres to the **ANS-104** **[8]** standard and includes a digital signature from the sender.
- **Scheduler Units** (**SUs**) sign messages upon assignment, allowing any participant to verify the authenticity and order of messages through cryptographic checks.

The combination of message signing, immutable storage on Arweave, and robust delivery semantics ensures that input data remains authentic and unaltered, thereby maintaining Input Integrity.

3.1.2 Model Integrity

Model Integrity guarantees that the AI model used for inference is accurate, unmodified, and corresponds to the intended version.

- Process Initialization and Environment Definition:

- Each AO process is initialized with specific **compute environment** (Env) parameters, including the type of Virtual Machine (VM) (e.g., **WebAssembly**), required extensions, and resource limits.
- The initialization data includes references to the AI model, typically as a hash or transaction ID stored on Arweave, ensuring that the model used cannot be altered post-deployment.

- Immutable Storage of Models:

- AI models are stored on Arweave, leveraging its permanent and tamper-proof storage capabilities.
- By referencing the model's unique hash, AO ensures that the exact version of the model is deployed and used consistently across all computations.

Through explicit environment definitions and immutable storage of models on Arweave, AO maintains the Model Integrity necessary for reliable AI inference.

3.1.3 Compute Integrity

Compute Integrity ensures that the AI inference computations are executed correctly, without errors or malicious interference.

- Deterministic Computation Functions:

- In the formal security model of the AO Computer, deterministic computation is achieved by processes using a deterministic function **F** to derive their state **S(P_i)** from the message **Log_i** and the compute environment **Env_i**, ensuring consistency and verifiability.
- Determinism ensures that given the same inputs and environment, the computation will always yield the same result.

- Independent Compute Units (CUs):

- CUs operate independently to compute the state of processes based on the immutable message logs and defined environments.
- After computation, CUs generate **signed attestations** of the results, which include the new state and any resulting outbound messages.

- Economic Incentives and Penalties:

- AO employs an economic model where CUs must stake tokens to participate in computations.
- If a CU provides incorrect computations, they face **slashing penalties**, disincentivizing malicious behavior and promoting honest computation.

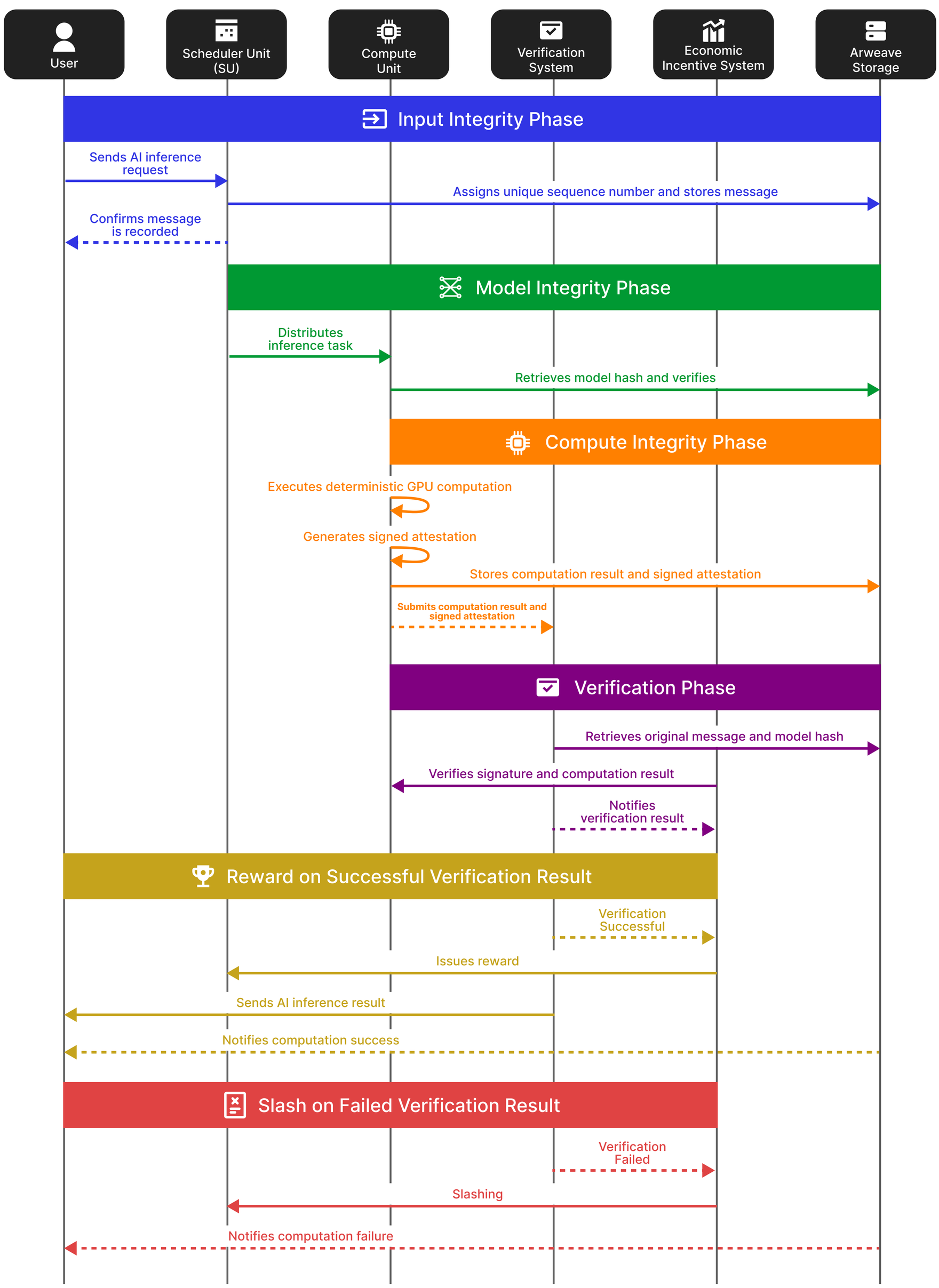
- Holographic State Mechanism:

- The state of any process is derived from its message log stored on Arweave. This allows any CU to independently verify and compute the process state, ensuring consistency and correctness across the network.

Utilizing deterministic functions, independent and verifiable compute units, and a robust economic incentive structure, AO upholds the Compute Integrity required for accurate AI inference.

In summary, the innovative and robust implementations of Input Integrity, Model Integrity, and Compute Integrity on AO Protocol and Arweave form a solid foundation that enables the **FPIF** Verification Mechanism to provide a verification process that is fast, inexpensive, and highly secure.

This combination ensures that AI inference within the AO ecosystem is both reliable and efficient, meeting the stringent requirements for Verifiable AI Inference in decentralized environments.



3.2 Deterministic GPU as AO extension

While the AO Protocol has demonstrated promising capabilities in executing deterministic AI inference on CPUs, such as running Microsoft's Phi-3-Mini-4K-Instruct model through aos-llama [9] (utilizing the llama land framework) , significant performance limitations remain.

Currently, with a 3.4B parameters model quantized to Q4, performing an inference task of 100 input tokens yielding 20 output tokens requires minutes-level processing time. This latency is unacceptable for commercial AI inference applications, where real-time or near-real-time responses are crucial.

Deterministic GPU computation is therefore essential to meet the performance requirements of commercial AI applications.

3.2.1 The Challenge of Non-Deterministic GPU Computation

Non-deterministic computation refers to processes where identical inputs may produce slightly different outputs across different runs or environments.

This variability is particularly prevalent in modern AI systems and introduces significant verification challenges. Non-deterministic behaviors in GPUs arise from several factors:

- **Algorithmic Randomness:** Random initialization, dropout layers, and stochastic elements
- **Floating-point Operations:** Floating points are not associative, [7] which means in highly parallelized GPU operations (a + b) + c may not equal a + (b + c).
- **Parallel Processing:** Race conditions and timing variations in parallel computations
- **GPU Architecture Variations:** Different GPU architectures or even CUDA versions can produce varying outputs due to their distinct handling of parallel computations.

3.2.2 Apus Network's Solution: Building a Deterministic GPU as AO Extension

To address the non-deterministic challenges of GPU computations within the AO ecosystem, Apus Network is dedicated to developing a Deterministic GPU as AO Extension for Fast Provable Inference Faults Machine Learning (**FPIF** ML). This extension leverages the AO Protocol and Arweave's immutable storage to ensure that GPU-based AI inference achieves the **speed**, **consistency**, and **verifiability** required for commercial applications.

Key Measures Implemented by Apus Network to Achieve Deterministic GPU Extension:

- **Eliminating Algorithmic Randomness:** By fixing random number seeds and utilizing controllable pseudorandom number generators, we eliminates non-determinism introduced by random initialization, dropout layers, other stochastic elements and relevant GPU operators.
- **Controlling Floating-Point Operations:** Adopting deterministic floating-point computation methods to prevent result discrepancies caused by the non-associativity of floating-point arithmetic. This may involve using high-precision arithmetic or controlling the order of operations through Nvidia determinism framework. [10]
- **Adjusting parallel processing strategies:** Adjusting parallel processing strategies in certain non-deterministic computation scenarios may lead to decreased performance. However, this trade-off is necessary to ensure determinism, thereby maintaining consistent and reproducible computation results.
- **Standardizing GPU Architectures:** Leveraging W3C standards WebGPU [11] to unify cross-platform behavior differences and customizing its underlying implementation to ensure consistent results within the same computation architecture. Additionally, defined computation architecture tags guarantee determinism across platforms.

Through these measures, Apus Network's Deterministic GPU Extension effectively resolves the non-deterministic issues inherent in GPU computations. This enables the provision of high-performance, reliable, and verifiable AI inference services, meeting the stringent requirements of commercial applications.

4. Roadmap

To achieve the seamless integration and enhancement of Fast Provable Inference Faults (**FPIF**) within the AO Protocol and Arweave ecosystem, Apus Network has outlined a comprehensive roadmap. This roadmap details the phased development and implementation milestones from Q4 2024 through Q3 2025.

2024 Q4 **Interface Definition**

- Complete interface specification for deterministic GPU as AO extension
- Develop Proof of Concept (POC) for deterministic GPU computation

2025 Q1 **Basic Integration**

- Establish initial integration work with AOS
- Implement some core deterministic GPU operators based on Nvidia determinism-framework [10]
- Complete WebGPU standard interface adaptation

2025 Q2 **Alpha & Beta Testing**

- Achieve first deterministic GPU model deployment
- Release Alpha & Beta version for community testing
- Establish baseline performance metrics

2025 Q3 **Production Release**

- Complete full integration with AO verification mechanism
- Launch production-ready version
- Enable enterprise-grade application deployment

>>> **Future Considerations**

- Implement cross-hardware deterministic guarantees
- Establish performance benchmarks across different hardware configurations

- Achieve seamless integration with mainstream AI frameworks
- Build comprehensive developer ecosystem
- Test on potential TEE implementation for privacy protection

5. Summary

In the landscape of decentralized verifiable AI inference, this paper has examined the limitations of existing verification mechanisms — from the high computational costs of **ZK ML**, to the latency and security issues of **OP ML** and **CE ML**.

We introduced **Fast Provable Inference Faults** (**FPIF ML**), a novel verification mechanism that combines deterministic computations, cryptographic attestations, and economic incentives to deliver fast, inexpensive, and secure verification.

Through integration with AO and Arweave, Apus Network aims to revolutionize the verification landscape, enabling scalable and secure AI services that meet commercial demands in decentralized world.

References

[1] https://foreignpolicy.com/2024/10/23/nobel-prize-warning-ai-hinton-hassabis-nihon-hidankyo/

[2] https://www.grayscale.com/research/reports/ai-is-coming-crypto-can-help-make-it-right

[3] https://www.arweave.org/

[4] https://ao.arweave.dev/#/read

[5] https://medium.com/dragonfly-research/dont-trust-verify-an-overview-of-decentralized-inference-c471a9f7a586

[6] https://medium.com/@ModulusLabs/chapter-5-the-cost-of-intelligence-da26dbf93307

[7] https://stackoverflow.com/a/69754285

[8] https://github.com/ArweaveTeam/arweave-standards/blob/master/ans/ANS-104.md

[9] https://github.com/samcamwilliams/aos-llama

[10] https://github.com/NVIDIA/framework-reproducibility

[11] https://www.w3.org/TR/webgpu/