Wrote by: Krieger (Sumit Kumar)

January 2025

"Across the board, I see far too many plans to save the world that involve giving a small group of people extreme and opaque power and hoping that they use it wisely. And so, I find myself drawn to a different philosophy, one that has detailed ideas for how to deal with risks, but which seeks to create and maintain a more democratic world and tries to avoid centralization as the go-to solution to our problems." — Vitalik Buterin

Introduction:

Today, AI is experiencing rapid growth, and many companies are either currently holding or aiming to establish a monopoly in the AI sector. This is a cause for alarm because AI, unlike other traditional inventions, is fundamentally rooted in intelligence. The potential consequences of monopolizing this industry could be catastrophic for humanity.

Moreover, the current state of the AI market lacks transparency, making it challenging for users to verify the quality of the AI models they receive from suppliers. This situation opens avenues for potential manipulation, which could harm both clients and suppliers.

To address this issue, there is a crucial need for a transparent proof-of-inference marketplace that establishes trust between clients and suppliers. In response to this challenge, we propose InfoNex, a blockchain-based infrastructural solution. InfoNex is a decentralized AI platform designed to enable an open marketplace for AI models where users can access inference services offered by multiple, untrusted AI service suppliers. The platform aims to ensure that users are guaranteed good quality of service and suppliers receive fair payment for their services.

Challenges in the current AI Marketplace:

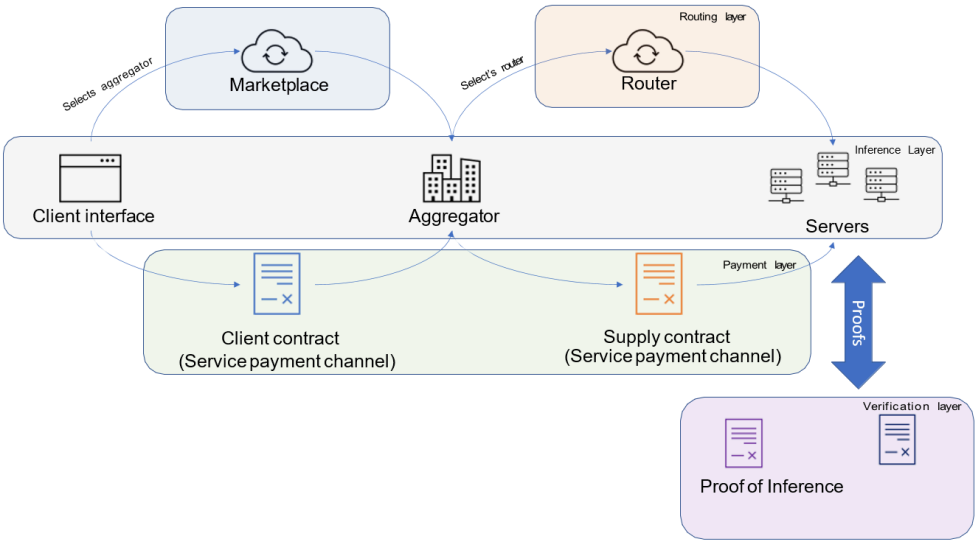
- Individual suppliers may not be able to attract enough clients.
- The supplier may not apply a good model and return low-quality results.
- The client may not pay after getting the service.

To address these challenges, InfoNex's decentralized AI service model includes Allowing an aggregator to collectively offer service on behalf of multiple suppliers, with an SLA (Service Layer Agreement) implemented as a smart contract to ensure fair revenue sharing.

Implementing a proof system for quality of AI services, using a challenge-response setup. Utilizing smart contracts and payment channels to implement scalable and reliable payment services for suppliers, supported by an objective dispute resolution mechanism to ensure suppliers can get paid if they deliver service.

Technical Overview:

InfoNex is based on a four-layer architecture to ensures that the client receives a trust-free, incentive compatible, byzantine resistant AI services this layered approach allows the protocol to enable proof of inference and proof of model ownership which provides cryptographic resistance against a variety of misbehaviours.



InfoNex's architecture

Proposed Model:

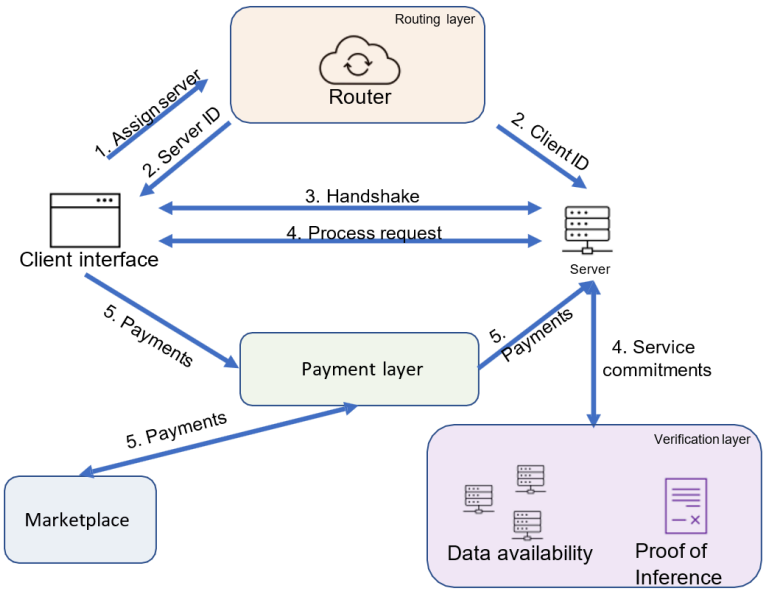
1. An aggregator module collectively offers service on behalf of multiple suppliers. The aggregator and suppliers engage in an SLA implemented as a smart contract to ensure that each gets a fair share of the revenue.

2. A proof system for quality of AI services to ensure that suppliers provide the promised quality of service. The proof is implemented through a challenge-response setup executed using a decentralized pool of challengers (Optimistic approach).

3. Smart contracts and payment channels to implement scalable and reliable payment service for the suppliers. This will be supported by an objective dispute resolution mechanism to ensure that suppliers can get paid if they deliver service.

Four Layer Architecture:

Inference Layer:



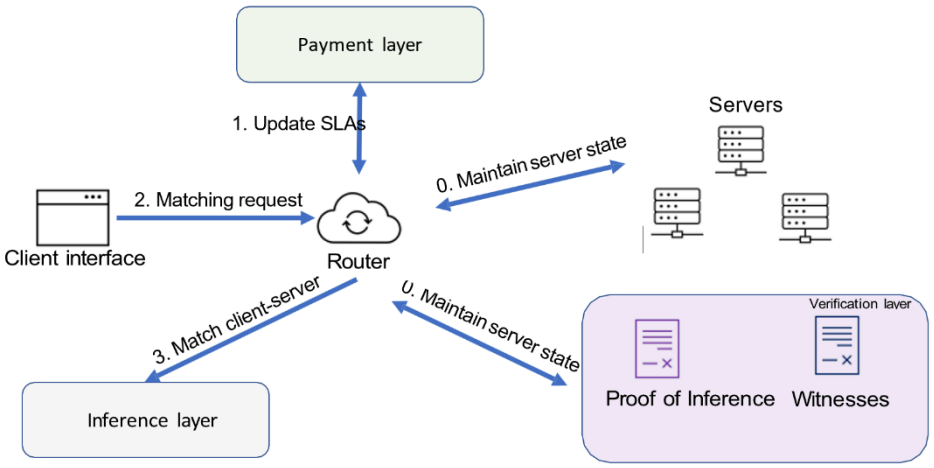
Inference layer overview

The inference layer facilitates ML inference queries and manages service information for the verification layer (commit service information to the verification layer). It operates in a modified Web2 server-client architecture for the proof framework.

Inference layer architecture include:

- **Service Exchange:** The client connects to the server, verifying an SLA path through the common aggregator. Inference requests are sent using the server's API endpoint, signed by the client for potential dispute resolution. The server processes requests and returns output data. Micropayments, as per SLA, are handled by the payment layer, ensuring payment before serving subsequent requests.
- **Service Dispute Witnesses:** Data exchanged in the service layer serves as evidence in payment disputes. Signed inference requests, output data committed to a Proof layer, and micropayment records are used for resolution, detailed in subsequent sections on the Transaction and Proof layers.

Routing Layer:



Routing layer overview

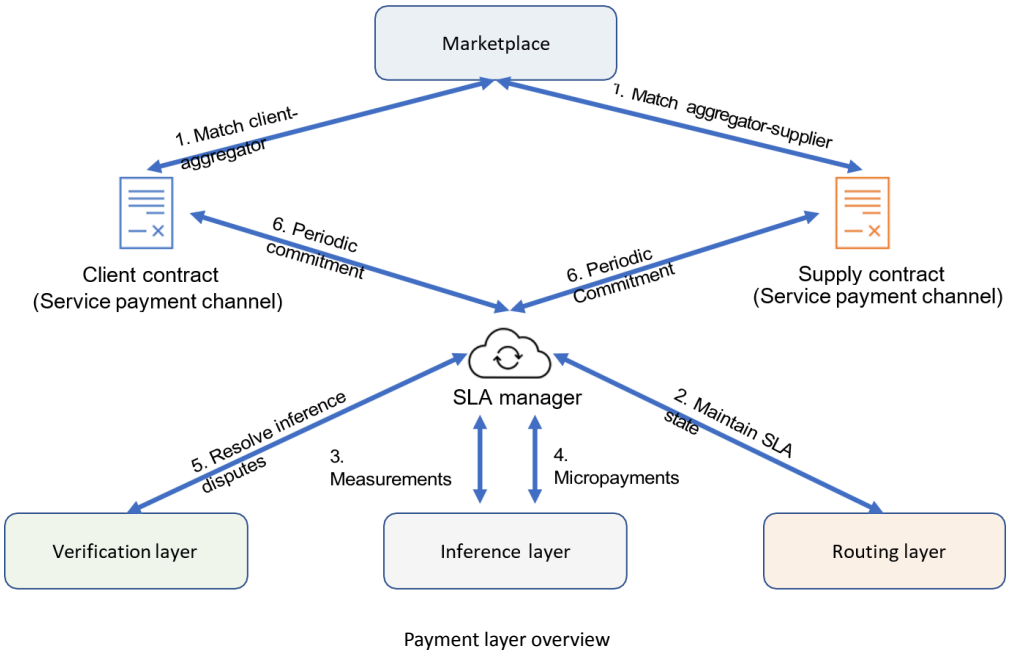
The routing layer in the system is responsible for matching clients with servers. This involves a router contract performing load balancing based on factors like latency, compute cost, and compliance with SLAs. Servers subscribe to a router with a variable set of parameters, and clients can select their preferred server.

A few key features of the routing layer include:

- **Server State Maintenance:** Router maintains server network state, including:
  a) Server model capacity (AI models the server can handle)
  b) Server hardware capacity (compute capacity)
  c) Server request load (number of connected clients)
- **SLA State Maintenance:** Router tracks SLAs signed at the transaction layer, ensuring matching clients and servers share a common aggregator. Monitors payment layer contracts to register or deregister SLAs.
- **Client-Server Matching:** Clients submit requests specifying server preferences (model id, location boundary, server uptime, etc.) The routing layer assigns a server for an AI model upon client request. The client is informed of the server's ID and address, and the server is notified of the incoming connection.

The router uses a matching logic that considers both the server's state and its SLA to select the most suitable server. It then notifies the inference layer to establish a connection and the payment layer for payment anticipation through their shared aggregator.

Payment Layer:



Payment layer overview

The payment layer manages Service Level Agreements (SLAs) through smart contracts, facilitating seamless transactions and conversion of service messages to service units (economic incentives) between two parties in a decentralized manner. This layer is built on a blockchain network5 to ensure transparency, security, and automation of payment processes. The key components of this layer include Service Contracts, SLA Manager, Measurement Gateway.

Service Contracts:

- Consist of two main components: SLAs agreed upon by transacting parties and unidirectional payment channels.
- SLA Specification: Utilizes the SLA4OpenAPI standard for codifying SLAs, mapping service usage to token payment amounts, particularly for AI applications (e.g., model type, input/output size).
- Payment Channel Setup: Establishes a unidirectional payment channel with escrow, defining terms for the delegation of payment keys to an intermediary SLA Manager.

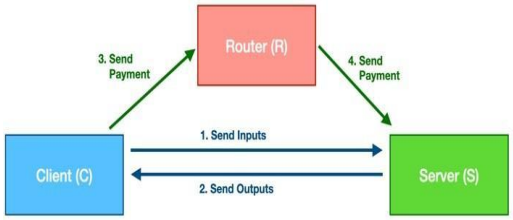
SLA Manager:

- Role: Manages SLAs by receiving signed measurements from both consumer and supplier SLA clients.
- Payment Conversion: Converts received measurements into appropriate payment amounts by signing micropayments, facilitating fund transfers on the established payment channel on behalf of the consumer.
- Delegated Payment Handling: Manages the delegation of payment keys according to the terms specified in the SLA, ensuring secure and automated payment processes.

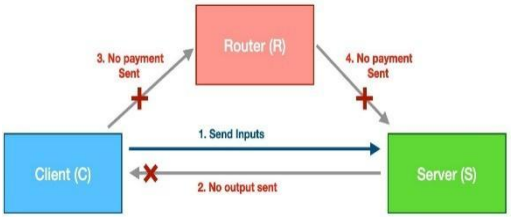
SLA Client and Measurement Gateway:

- Measurement Interpretation: Measurement gateway interprets service messages and converts them into service units, relevant to AI applications (e.g., model requested, input/output size).
- Information Flow: SLA client fetches information from the measurement gateway, signs it with the key from the service contract, and forwards it to the SLA Manager. Optionally, the consumer-side SLA client can convert measurements into micropayments for direct forwarding to the supplier.

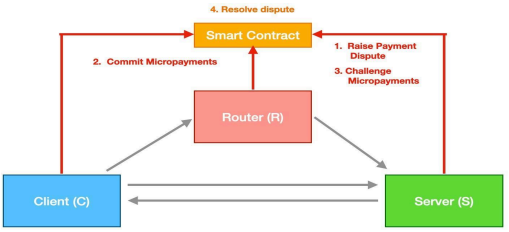
Verification Layer:



Ideal path

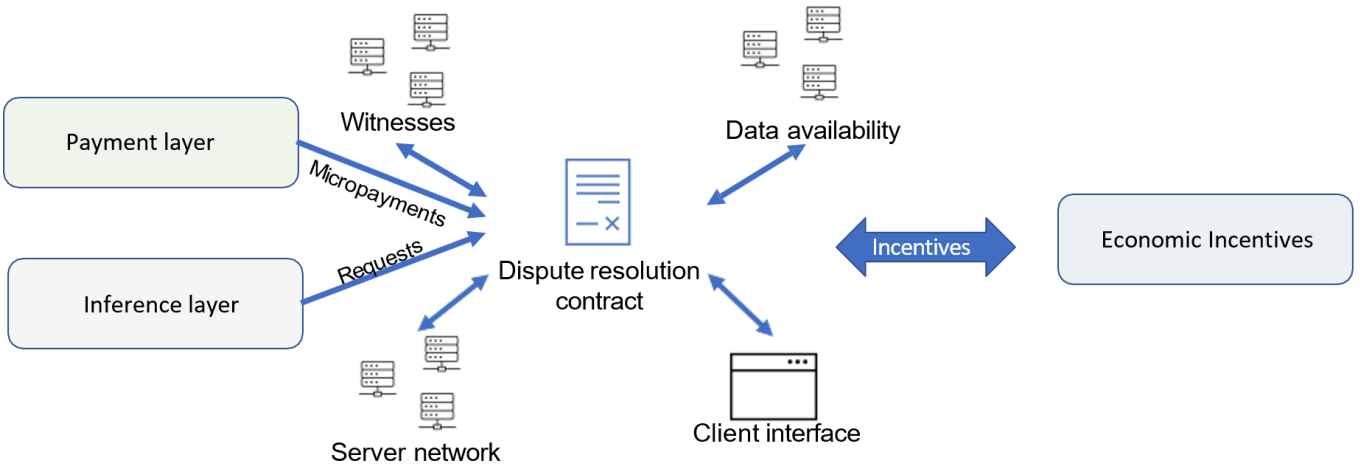


Service dispute



Dispute resolution

Dispute resolution through micropayment (payment layer)



Verification layer overview

The verification layer in InfoNex leverages blockchain and cryptography for immutable and trusted service state management. In this proposal we specifically highlight two proof categories, each designed to address distinct types of disputes in the system.

Proof of Inference: Proof of Inference serves as evidence for accurate computations on a specified (and open) AI model, facilitating resolution in disputes over correct inferences.

Proof of Model-ownership: This establishes evidence of the degree of similarity between two AI models, determining whether one is a clone or a fine-tuned version of the other. This proof mediates potential disputes concerning intellectual property owned by the AI model's proprietor.

To make InfoNex Byzantine resistant there are other proof systems that may be explored for new attack vectors such as proof of custody, proof of data ownership and proof of AI hosting etc.