Threshold AI Oracles: Verified AI for Event-Driven Web3

Supra Research

May 22nd, 2025

Abstract

The Threshold AI Oracle system redefines Web3 oracle infrastructure by transitioning from passive data feeds to intelligent, deliberative frameworks capable of interpreting complex real-world events for decentralized applications. While conventional oracles are generally accurate and effective for delivering price data, they are limited in scale and scope. Computation and communication overheads grow linearly in the number of data feeds, which results in elevated latency, bandwidth consumption, and/or resource inefficiency as these systems scale to cater to a variety of Web2 sources. Moreover, oracle developers must anticipate user demands in advance, as their data feeds cannot be updated dynamically. This limitation is especially pronounced in the Web3 AI context, where it is inherently difficult to predict the questions or prompts users may submit.

Threshold AI Oracles solve both problems in a single key modification: They introduce a *just-in-time query* model that generates structured, verifiable outputs only upon request or when predefined conditions are met. By leveraging AI-driven multi-agent committees, the system delivers trust-minimized, low-latency outputs across potentially unlimited real-world data types. Its phased implementation begins with numerical outputs suitable for prediction market resolutions, progresses to structured textual commands for decentralized finance automation, and culminates in executable logic for adaptive governance.

Applications include resolving trade agreement outcomes, rebalancing DeFi portfolios based on sentiment metrics, and dynamically updating DAO policies. Operating as a decentralized Epistemic Layer, the system transforms unstructured inputs into structured, verified on-chain/Web3 knowledge. While challenges remain, including natural language aggregation, subjectivity, model drift, adversarial attacks, and equitable access, this work lays the foundation for a responsive, intelligent blockchain system. It invites collaborations across disciplines to advance AI summarization, cryptographic verification, and decentralized governance for next-generation Web3 infrastructure.

Contents

1. Introduction

The convergence of artificial intelligence (AI) and blockchains offers a significant opportunity to allow rather solitary decentralized systems to realize what AI has already learned from the real world [1]. In decentralized prediction markets such as Polymarket, participants wager on outcomes involving geopolitical developments, regulatory decisions, or macroeconomic shifts. These platforms require oracles not merely to relay raw data but to deliver structured interpretations of complex events, and thus their resolving solutions often require human input and are not algorithmically decided [2].

Oracles today, such as the Distributed Oracle Agreement (DORA) protocol developed by Supra Labs [3], are accurate and reliable, particularly in delivering structured financial data such as price feeds. DORA leverages a coherent cluster approach to achieve consensus with a simple majority of nodes, significantly improving efficiency over traditional oracles requiring a supermajority [3]. However, the decentralization that ensures trust minimization also introduces redundancies and, in some configurations, inefficiencies. Today, most oracle systems operate through either push-based or pull-based models. In both cases, the list of data items must be pre-determined, continuously produced. Sometimes, it must be finalized on a source chain or via a consensus process before it can be transmitted to others, increasing latency [4]. High-frequency publishing across hundreds or thousands of tracked data points also increases computational and storage load. Furthermore, the scope of these systems remains narrow, primarily supporting price feeds and other simple scalar values [5].

The Threshold AI Oracle system expands this aperture significantly. It introduces a just-in-time query architecture, wherein data is computed, verified, and delivered only when explicitly requested or when predefined conditions are met. Moreover, a multitude of types of data are able to be verified through this method, enabling smart contract platforms to trigger based on intelligent reasoning [6]. If AI agents can reliably interpret a given query, the system can produce a verifiable, low-latency output that is suitable for on-chain execution. Applications extend far beyond financial metrics to include regulatory analysis, policy monitoring, sentiment inference, and automated governance triggers.

To enable this functionality, Threshold AI Oracles integrate advanced AI agent systems with distributed cryptographic consensus mechanisms. Unlike traditional oracles that rely on static feeds or human voting processes, this system employs specialized multi-agent committees. These agents deliberate over real-world events using natural language inputs, assess contextual data from diverse sources, and reach quorum through threshold cryptographic signatures. When consensus is achieved, a signed output is published on-chain, initiating smart contract execution or informing decentralized governance.

This architecture supports a phased implementation strategy. The initial phase delivers numerical outputs suitable for automation-driven smart contract execution. The intermediate phase introduces advanced custom commands, adding sophistication to decentralized finance automation. The final phase enables AI-generated logic for dynamic, self-modifying smart contract behavior. By minimizing unnecessary data publication and extending oracle capability beyond price feeds, the Threshold AI Oracle system addresses key inefficiencies in current designs while significantly expanding the domain of real-time on-chain automation.

As a decentralized Epistemic Layer, the system enables blockchain applications to reason over and respond to complex real-world conditions. This foundation supports a new generation of intelligent, responsive decentralized infrastructure capable of safely integrating a vast array of external data with deterministic execution.

2. Background and Motivation

2.1 The Oracle Problem in Web3

Blockchains are closed systems. The vast majority of them generally cannot natively access or interpret data from the external world [7]. To overcome this limitation, decentralized applications rely on oracles, which serve as middleware that delivers off-chain information to smart contracts. Oracles are critical to the

operation of many Web3 use cases, including decentralized finance (DeFi), prediction markets, supply chain auditing, and on-chain governance [1].

Furthermore, traditional oracles tend to rely on a limited amount of specific data sources that assume a deterministic reality, hence most oracle solutions only handle price feeds [3]. They are not designed to evaluate complex or evolving conditions that require interpretive judgment. For example, determining whether a geopolitical event has materially occurred may require contextual analysis that falls outside the capabilities of simple feed-based reporting. Validation in these systems is typically limited to a small number of whitelisted data providers or basic aggregation rules, which can introduce trust dependencies and increase the risk of manipulation [4, 8, 5].

In this form, conventional oracles are limited in servicing decentralized applications that require adaptive analysis, conditional execution, or responsive automation. Existing solutions are not well suited for workflows that depend on complex event detection, multi-source validation, or reasoning across real-world ambiguity [9]. These constraints motivate the need for a new class of oracles capable of on-demand, verifiable interpretation across a wider range of data domains.

2.2 The Rise of AI Agents

Recent developments in artificial intelligence, particularly the emergence of large language models (LLMs) and autonomous agent frameworks, have significantly advanced the ability of machines to process and interpret unstructured information. Systems such as GPT-4, AutoGPT, and BabyAGI demonstrate that AI agents can summarize ambiguous input, infer contextual significance, and resolve ill-defined queries, including questions such as whether a given event is consequential [10, 6].

In contrast to traditional data feeds, which typically rely on a predefined stream of structured inputs, AI agents can dynamically access and query a wide range of heterogeneous data sources [11]. These include Web2 APIs, real-time news outlets, social media platforms, and open databases. Furthermore, agents can autonomously coordinate their own workflows, decomposing complex tasks into modular subtasks and executing them in sequence or in parallel. Many are also capable of initiating external actions, such as executing smart contract functions or interfacing with blockchain protocols through plugin integrations or remote procedure calls [6].

These features suggest that AI agents could serve as powerful bridges between the real world and decentralized infrastructure, offering a degree of interpretive and procedural flexibility that traditional oracle systems cannot provide. However, several limitations currently hinder their adoption within Web3 contexts. Most AI agents operate in centralized environments with opaque decision-making processes. Their outputs are inherently non-deterministic, often exhibiting variability across runs, and are susceptible to hallucinations or inconsistent reasoning. Without cryptographic guarantees or reproducibility, their outputs cannot be considered reliable within trustless systems [12].

Consequently, although AI agents possess significant potential to enhance event understanding and decision automation, they remain fundamentally incompatible with the deterministic and verifiable requirements of decentralized architectures. Overcoming this gap requires the integration of AI reasoning capabilities with cryptographic coordination mechanisms to ensure both interpretive accuracy and verifiable trust.

2.3 The Need for a New Oracle Model

The limitations of traditional oracle architectures, when considered alongside the emerging capabilities of AI agents, highlight the need for a new class of oracle systems. Such a system must combine the interpretive flexibility of artificial intelligence with the verifiability and auditability required in trustless blockchain environments [11].

A next-generation oracle must be query-driven and selective in its behavior, responding only to user-defined conditions rather than continuously broadcasting data irrespective of context. This selectivity reduces bandwidth consumption, lowers transaction costs, and avoids the inefficiencies associated with constant data production [4]. It must also be reasoning-aware, capable of evaluating complex, ambiguous, or evolving

situations. For example, identifying the emergence of a geopolitical crisis or a regulatory shift often requires contextual understanding that exceeds the capacity of structured feeds.

In addition to interpretive capability, integrity via verifiability is crucial. Oracle outputs must be derived from quorum-based consensus among a diverse set of agents and secured through digital signatures that preserve integrity and reproducibility via transitive authentication. These guarantees are necessary for smart contracts to treat the data as trustworthy and actionable, particularly in financial or governance contexts where accuracy and accountability are paramount [13].

Lastly, the system must be optimized for low-latency performance. Real-time applications in decentralized finance, governance automation, and cross-chain coordination require condition-based triggers that can be executed immediately upon event detection [14]. By reconceptualizing oracles as intelligent, deliberative systems rather than passive data pipelines, this new model enables more scalable, adaptive, and meaningful interaction between decentralized applications and real-world phenomena [3].

3. Core Concepts

3.1 Just-in-Time Versus Always-On Oracles

Conventional oracle systems generally operate under a push-based paradigm, continuously transmitting data to blockchain networks irrespective of current application demand. This model, while effective for real-time financial feeds, introduces several inefficiencies. These include increased network congestion, elevated gas fees, and excess computational load caused by the unfiltered injection of data into on-chain environments [4]. Recently, a pull-based paradigm has become popular, wherein verified data is submitted to the blockchain only when the client/user request it. While the pull-based model mitigate the on-chain inefficiencies, the verified data is still continuously produced off-chain. Equally importantly, for both push/pull models, there is no adaptability: if a user queries for a new, personalized type of data/opinion, they cannot produce and include an answer just in time.

The Threshold AI Oracle system introduces a just-in-time oracle architecture towards overcoming the above issues. In this model, data is processed and published only when explicitly requested by a user or when a predefined condition is satisfied. For instance, a decentralized finance (DeFi) application may issue a conditional query that requires resolution only if market volatility, triggered by a geopolitical event, exceeds a specified threshold [14]. This targeted query structure eliminates the need for high-frequency data streaming and constant price feed updates.

The operational workflow begins with the submission of a conditional query by a decentralized application or smart contract. Upon receipt, the system activates specialized AI agents that monitor relevant data sources in real time. If an event matching the query condition is detected, and a quorum of agents independently converges on a shared interpretation, a cryptographically signed result is produced and published on-chain.

This just-in-time approach offers several advantages. It reduces on-chain data volume, thereby enhancing scalability and lowering transaction costs. It also preserves user privacy by avoiding the unnecessary exposure of potentially sensitive information [3]. More broadly, this model redefines the oracle's role from a passive data broadcaster to an intelligent event interpreter, capable of delivering selective, actionable insights for decentralized execution environments.

3.2 Multi-Agent Committees

A central component of the Threshold AI Oracle system is its deliberative architecture, which relies on multi-agent committees. These committees are composed of AI agents with distinct functional roles and access to heterogeneous data sources. This structural diversity increases the interpretive resilience of the system and reduces the likelihood of manipulation or systemic error.

Each agent within a committee is responsible for a specialized evaluative task. Some agents focus on verifying factual accuracy, others are designed to identify inconsistencies or misinformation, and still others assess higher-order contextual variables such as legal implications or market sentiment. By operating independently,

each agent contributes a differentiated viewpoint that enriches the collective decision-making process. The functional and informational diversity across agents serves as a safeguard against correlated failure and groupthink [15].

Within a committee, agents engage in structured deliberation using natural language exchanges to evaluate how an event should be interpreted [6]. For example, when reviewing a corporate news release, agents may debate whether the announcement constitutes a material risk to a financial position. This process mirrors the logic of a decentralized jury system, where the final decision emerges from a negotiated synthesis of individual interpretations rather than a pre-defined or deterministic rule set [16].

By fostering pluralistic reasoning, the committee model enhances robustness against misinformation, adversarial influence, and data bias. It also supports the generation of adaptive, context-aware outputs that are better suited to the complexities of real-world events. This enables the oracle system to move beyond binary signaling and deliver nuanced, actionable information suitable for decentralized execution.

3.3 Parallelized Reasoning Inspired by STM

To enable responsiveness in time-sensitive environments, the Threshold AI Oracle system adopts a parallelized execution architecture inspired by principles from Software Transactional Memory (STM) and partial-order computation models [17, 18]. Unlike sequential systems, which process tasks through a linear pipeline, this design allows multiple agent committees to evaluate independent dimensions of a query simultaneously.

The parallelization process begins with a network of sentry nodes responsible for receiving requests from smart contracts and verifying associated payments. These nodes serve as interpreters that analyze the structure and intent of each query. Before dispatching the query to downstream agent committees, sentry nodes conduct a localized verification process, which may include a consensus round to confirm the authenticity and validity of the request [15]. Once verified, the request is cryptographically signed by the sentry nodes, ensuring that only authenticated and paid queries are processed further.

This pre-processing stage serves multiple purposes. First, it minimizes wasted computation by filtering out invalid or unauthorized requests before they reach more resource-intensive AI committees. Second, it reduces overall latency by organizing reasoning tasks per request to respective AI agent committees.

Following this verification, the signed request is delineated into sub-tasks and dispatched. Each committee is assigned to evaluate a specific dimension of the query, such as factual accuracy, contextual relevance, legal interpretation, or financial impact. These evaluations proceed in parallel, and the system aggregates results only when each committee satisfies its quorum threshold.

The architecture draws from principles in distributed database systems, where transactions are committed by independent processes without requiring global synchronization [19]. Even when some agents timeout or produce ambiguous outputs, the system can finalize a decision as long as quorum is achieved within a subset of relevant committees. This fail-soft behavior enhances system resilience and availability.

Through modular committee specialization and sentry node coordination, the system supports concurrent reasoning while maintaining cryptographic verifiability. These properties enable high-throughput, low-latency responses, making the system well suited to applications such as DeFi liquidations, governance automation, and condition-based event resolution in decentralized environments.

3.4 Threshold Cryptographic Signing

Final outputs produced by the Threshold AI Oracle system are secured through threshold cryptographic signatures. This mechanism consolidates the collective agreement of multiple AI agents into a single verifiable artifact, enabling efficient and tamper-resistant interaction with blockchain execution environments [3].

Once a quorum of agents reaches consensus on the interpretation of an event, their decision is encoded into a threshold signature. For example, BLS signatures facilitate this process by aggregating individual agent signatures into a compact and unique proof [20]. The resulting signature is concise, easy to verify on-chain, and highly efficient in terms of gas consumption.

Threshold signatures offer several operational advantages. They enable near-instantaneous verification of outputs, support scalability through compactness, and are economically viable for frequent or complex operations. The quorum threshold itself can be parameterized according to the criticality of the task. For decisions involving high-value assets or systemic risks, the system may require a supermajority of a large quorum. For lower-risk outputs, such as informational updates or auxiliary logic triggers, a smaller quorum may be acceptable [21].

This cryptographic layer serves as a final validation checkpoint. Only outputs that have passed through structured deliberation, satisfied quorum conditions, and been signed by an authorized group of agents are eligible for submission to smart contracts. In this way, threshold signing provides a verifiable and secure interface between probabilistic AI-generated reasoning and the deterministic execution logic of blockchain systems.

4. Protocol Design

The Threshold AI Oracle system provides decentralized applications with a framework for securely integrating real-world data by combining artificial intelligence with distributed consensus mechanisms [3]. This integration is governed by a cryptoeconomic protocol that enables intelligent, verifiable automation based on structured event interpretation [22].

The protocol is designed to process complex, multi-dimensional queries by coordinating specialized AI agents, validating diverse data sources, and delivering cryptographically signed outputs to smart contracts. Its architecture supports trust-minimized execution by ensuring that all outputs are generated through a consensus-driven process involving both deliberation and threshold signing.

This section details the core components of the protocol, including the query lifecycle, agent role specialization, committee-level deliberation, inter-committee orchestration, on-chain integration, phased output generation, and the summarization process. These elements collectively enable the system to scale efficiently, maintain security guarantees, and remain resilient to adversarial interference or manipulation.

4.1 Request Lifecycle

The protocol transforms real-world queries into cryptographically verifiable on-chain events, supporting automated decision-making in decentralized applications. The process begins when a user or smart contract submits a query, such as determining the outcome of a prediction market bet or assessing market sentiment [8]. Specialized AI agent committees, selected for domain expertise, evaluate the query in parallel [6]. For instance, one committee verifies factual data, another detects inconsistencies, and a third analyzes contextual metrics, such as sentiment trends.

Each committee engages in deliberation, exchanging natural language arguments to interpret the query and propose a response. In the initial phase, committees may be focused on producing numerical outputs, progressing to structured textual commands and sophisticated logic. If a committee achieves a predefined threshold of agreement, its output is cryptographically signed. Dependencies among committees are managed through a staged progression model, where signed outputs from one committee enable subsequent deliberations. A meta-aggregator collects signed conclusions, and upon achieving global quorum, a threshold signature is generated. The signed output is published on-chain, triggering actions such as prediction market resolutions or DeFi automation tasks, ensuring efficient, trust-minimized execution [15].

4.2 Agent Roles and Specialization

The Threshold AI Oracle protocol relies on a diverse set of AI agents, each assigned to a specific role within a committee architecture optimized for interpretive reasoning [6]. These agents are functionally specialized to ensure that multiple analytical perspectives are represented in the deliberation process. Committees are formed dynamically in response to queries, with role assignments tailored to the nature and complexity of the input.

Each agent type contributes distinct capabilities. Validator agents are responsible for verifying factual claims against structured, verifiable data sources. Adversarial simulators are designed to detect inconsistencies, misinformation, or adversarial manipulation within the input set. Contextual analysts evaluate non-deterministic metrics such as market price feeds or economic indicators. Summarizer agents synthesize deliberation outcomes into candidate proposals, while judger agents could assess logical coherence and evidential consistency of the arguments presented.

These roles are executed across a range of AI architectures. Agents may employ fine-tuned language models, symbolic reasoning engines, or deterministic rule-based systems, depending on their assigned function. All agents interact with external data environments, including Web2 APIs, public databases, and indexed news content, under controlled data provenance protocols that mitigate the risk of collusion or misinformation injection.

Agent participation can be governed by reputation systems that track accuracy, response consistency, quorum alignment, and overall reliability across past tasks [23]. These reputation scores can influence role assignment, participation frequency, and committee inclusion. The modularity of this agent system design allows the protocol to balance interpretive precision, adversarial robustness, and contextual sensitivity without compromising scalability or responsiveness [24, 25].

4.3 Committee Deliberation Process

The committee deliberation protocol is designed to facilitate structured consensus formation while integrating diverse analytical perspectives. This mechanism draws inspiration from human governance models and adapts them for autonomous agent coordination. The goal is to enable outputs that reflect reasoned interpretation rather than only outputs based on simple data aggregation.

Deliberation proceeds through, for example, a round-robin or all-to-all structure in which agents contribute natural language arguments relevant to the assigned query [16]. Each agent introduces new information, counterpoints, or clarifications in response to previous contributions. A summarizer agent, selected either at random or based on reputation score, is tasked with synthesizing these arguments into a preliminary conclusion. The nature of this output depends on the query type, ranging from numerical values in early implementation phases to structured commands, or sophisticated smart contract functions in later stages.

Once a candidate summary is proposed, agents are required to vote to accept, reject, or contest it. Finalization depends on reaching a predetermined threshold, such as a two-thirds supermajority or a fault-tolerant quorum. If the threshold is not met, the committee initiates a new deliberation round with a different summarizer. In cases where deadlock persists, the query may escalate to fallback mechanisms, such as a larger, independent oversight committee or optional human review.

To ensure robustness against adversarial behavior, the system incorporates role diversity, crypto-economic staking, randomized agent assignment, and redundancy in data sourcing [26]. Malicious agents attempting to distort the process through fabricated or misleading arguments are neutralized through these structural defenses. By bounding the duration of deliberations and enforcing quorum-based decisions, the protocol balances rigorous interpretive reasoning with timely responsiveness for high-frequency applications such as prediction market resolution or automated governance [15].

4.4 Inter-Committee Orchestration

The protocol implements an orchestration model to manage coordination among multiple agent committees while avoiding full system synchronization. This design enables concurrent processing of distinct query dimensions, thereby maximizing throughput and reducing systemic latency. Committees operate independently unless a specific dependency exists between their respective outputs.

For instance, a committee tasked with evaluating market sentiment may depend on the prior resolution of whether a particular geopolitical event occurred. In such cases, the dependent committee will defer execution until the required upstream output is cryptographically finalized. This dependency structure draws on principles from distributed computing and partial-order execution, ensuring that committees only

wait when absolutely necessary, which minimizes bottlenecks and supports continuous processing pipelines [27].

Each committee output includes a threshold-signed result, metadata indicating quorum participation, and a unique identifier. These components function as verifiable attestations that downstream committees and smart contracts can reference to validate data integrity and execution eligibility. By encoding outputs in this structured format, the system supports flexible dependency resolution without requiring global synchronization.

To protect against coordinated manipulation or data corruption, the protocol enforces diversity in committee composition and data source allocation. Agent assignment may be randomized, and cross-committee overlap is minimized to prevent collusion or systemic bias [28]. These safeguards are essential to maintaining adversarial resilience, particularly in environments where malicious nodes may attempt to exploit shared dependencies or quorum pathways [15].

4.5 On-Chain Integration

The Threshold AI Oracle system enables secure and scalable integration of off-chain data with on-chain execution environments. Unlike traditional oracles that rely on static feeds or centralized intermediaries, this system leverages AI-driven deliberation and threshold cryptographic consensus to interpret complex queries and deliver actionable, verifiable outputs to smart contracts [8]. Each output is secured using Boneh–Lynn–Shacham (BLS) threshold signatures [20], ensuring minimal verification overhead and compatibility with decentralized infrastructure.

Smart contracts require structured, deterministic inputs to execute predefined logic. To meet this requirement, Threshold AI Oracles generate outputs in developer-specified formats, balancing expressiveness with verifiability. The system supports a phased implementation strategy, with increasing levels of output complexity and interpretive sophistication.

Phase 1: Numerical Outputs. In the initial phase, agents return discrete or continuous numerical values selected from predefined domains registered by smart contracts [3]. Examples include categorical responses (e.g., 1, 2, 3 for prediction market outcomes) or sentiment scores on a scale from 1 to 10, for example. These responses are computed through statistical summarization, typically the median of $(2f + 1)$ submitted values, ensuring robustness against up to f faulty or malicious agents. This model avoids the need for complex model inference, thereby reducing computational load on validators. Once consensus is reached and the result is signed, validators can efficiently verify the output and trigger corresponding smart contract logic.

For example, a prediction market querying whether a trade agreement is signed by a certain date might return Option A (signed), Option B (delayed), or Option C (canceled), encoded as 1, 2, or 3. Nodes analyze diplomatic reports and market indicators, deliberate, and reach consensus on the output, which is then signed and published. A different query could assess Ethereum sentiment. If the final score is 7, a contract may execute a rule to convert 25% of stablecoins into ETH.

Phase 2: Structured Commands. In this intermediate phase, agent committees may generate JSON-formatted outputs that include a specified command, parameters, and a reasoning field. For example, an output may take the form: { "command": "adjust_payout", "parameters": { "amount": 5000, "account": "0x123" }, "reasoning": "Volatility index 75% justifies increased payout" }. These commands are drawn from a finite set of predefined operations, including payout calculations, task finalization, score updates, event logging, and trade execution. Inputs may include percentages, token quantities, or blockchain addresses.

To ensure consistency and resistance to adversarial manipulation, agent responses are clustered using embedding models and verified via pairwise Natural Language Inference (NLI) tests [29]. Deterministic summarization requires at least $(f + 1)$ agents to submit semantically coherent responses. This phase increases expressivity while preserving the verifiability necessary for smart contract execution. Challenges in this phase include securing the semantic embedding space from adversarial inputs. Mitigations include using diverse AI architectures, randomized committee assignments, and slashing conditions for incoherent or malicious behavior.

Phase 3: Sophisticated Logic Generation. The final phase of integration enables agents to dynamically generate executable smart contract logic. This includes functions that incorporate conditional automation, such as rule-based responses to changes in external variables. Logic generation involves several coordinated steps: (1) natural language interpretation of the query and associated context, (2) summarization of proposed reasoning, (3) algorithm design, and (4) deliberation over parameter choices and operational boundaries. Only when (f + 1) agents agree on a valid logic construct, and the code passes verification or simulation-based stress testing, can it be signed and published [30].

As an example, a DeFi protocol may request adaptive payout logic based on market volatility. The resulting output may be: { "logic": "function adjust_payout(volatility, account) { if (volatility > 50) return 110; else return 100; }", "reasoning": "Volatility-based adjustment" }. The smart contract then compiles and executes this logic, adjusting token disbursements in real time.

This final phase introduces the highest computational demands and security requirements. While it offers maximum flexibility, it also necessitates robust safeguards, including adversarial simulation and formal logic validation, to prevent the injection of unsafe or biased code into autonomous smart contracts. Confidence in this capability is supported by recent advances in code-generating LLMs and machine-verified software, though rigorous evaluation remains essential to production deployment.

4.6 Phased Implementation Approach

The Threshold AI Oracle protocol follows a phased implementation strategy designed to deliver immediate functionality while supporting long-term innovation. This staged progression enables the system to gradually expand its interpretive and automation capabilities while preserving core guarantees of scalability, verifiability, and trust minimization [22, 31, 3].

The protocol is structured into three distinct phases of increasing computational and semantic complexity. In the initial phase, the system produces numerical outputs, such as discrete labels or scalar values drawn from predefined ranges. These are suitable for prediction market resolutions, binary event verification, or threshold-based triggers. Outputs are derived through statistical consensus methods, such as median aggregation, and are designed to be easily verifiable by smart contracts with minimal processing overhead.

The intermediate phase introduces structured textual commands in JSON format, enabling smart contracts to perform more granular automation tasks. Outputs include explicitly defined functions and parameter sets, along with contextual reasoning metadata. These commands are drawn from a limited vocabulary of contract-safe operations, such as calculating or adjusting payouts, triggering actions, or updating internal scores. Summarization in this phase is performed via embedding-based clustering and entailment validation, ensuring deterministic convergence among validators.

The final phase extends the system's capability to include the generation of executable smart contract logic. In this stage, agent committees interpret the query, design appropriate algorithms, and deliberate over parameterization. The resulting code is subjected to formal verification and adversarial testing before being cryptographically signed and submitted on-chain. This phase supports adaptive, stateful automation, enabling smart contracts to adjust behavior dynamically in response to real-world developments.

Each phase builds incrementally upon the protocol's foundational components, including AI agent deliberation, parallel committee orchestration, and threshold cryptographic signatures. This design ensures that the system remains operational and secure across all levels of complexity, while progressively expanding its ability to serve as a general-purpose oracle infrastructure for decentralized applications.

4.7 Summarization Process

The summarization process aggregates agent responses into a single, verifiable output and serves as a critical mechanism for ensuring trust-minimized execution throughout all protocol phases. Each phase employs distinct summarization techniques tailored to the format and complexity of the input data. This design supports consistency, fault tolerance, and security, even in the presence of adversarial behavior.

Numerical Inputs (Initial Phase). For numerical tasks, such as prediction outcomes or sentiment scores, each node submits a proposed value or range, typically constrained by a predefined domain. The designated summarizer computes the median of $(2f + 1)$ responses, which provides Byzantine fault tolerance against up to f malicious agents [15, 32, 3]. This median computation is a lightweight operation that validators can independently reproduce. If a validator's local computation matches the reported median, it votes to confirm the output and initiate smart contract execution.

Textual Inputs (Intermediate Phase). In this phase, agents generate structured textual outputs in response to interpretive queries. Each node submits a text response, which is converted into an embedding vector for semantic comparison. The summarizer identifies at least $(f+1)$ mutually coherent responses using pairwise Natural Language Inference (NLI) to test for entailment and consistency [29]. The resulting cluster forms the basis for a structured command, which is then signed and broadcast. Validators independently reproduce the embedding clustering and entailment checks to ensure deterministic summarization.

This phase is vulnerable to adversarial attacks, particularly when malicious agents attempt to manipulate the embedding space or inject subtly divergent responses [26]. To mitigate these risks, the system incorporates a combination of diverse AI model architectures, randomized committee assignments, and punitive slashing mechanisms. Ongoing research focuses on developing robust embeddings and adversarial-resistant clustering algorithms.

Logic Generation (Long-Term Phase). For executable smart contract logic, agents propose code that must be summarized into a single, verified script. For example, the summarizer may be required to consolidate at least $(2f + 1)$ signed proposals in order to propose a final outcome that is to thereafter be scrutinized by formal validation, simulation-based testing, or static analysis to ensure functional correctness and safety [33, 34]. Validators thereafter must produce an output that at least $(f+1)$ committee members have signed in order to be allowed to publish their script on-chain.

This phase imposes the highest computational burden and requires optimizations in formal methods to maintain acceptable latency and resource use. An open area of development is the design of efficient validation workflows that reduce verification overhead without compromising security guarantees.

Across all phases, the summarization layer acts as the final checkpoint in the deliberation pipeline. It ensures that all outputs, whether numeric, textual, or logical, are produced through verifiable, consensus-driven processes. This guarantees protocol integrity while enabling scalable support for increasingly complex applications across the decentralized ecosystem.

5. Cryptographic Verification Models

In trustless Web3 environments, the outputs of artificial intelligence systems must be verifiable through cryptographic means in order to be considered secure, auditable, and actionable [3]. Unverified outputs from AI models, regardless of their interpretive accuracy, are inadequate in decentralized contexts due to the absence of formal guarantees. Without cryptographic validation, there is no assurance that the reported result was honestly computed, faithfully deliberated, or agreed upon by a quorum of agents [15].

The Threshold AI Oracle protocol addresses this challenge by implementing cryptographic verification mechanisms that ensure outputs are generated through protocol-compliant processes. Every accepted result must originate from a deliberative procedure in which a quorum of agents contributes to a decision that is subsequently validated and signed. The system design is intended to preserve auditability, resist manipulation, and enable secure integration with on-chain logic.

This section evaluates three primary approaches for cryptographic verification of AI outputs: threshold signatures, zero-knowledge proofs, and trusted execution environments. Each approach offers different trade-offs in terms of performance, trust assumptions, transparency, and implementation complexity. While all three approaches are theoretically viable, the protocol prioritizes threshold signatures due to their favorable alignment with the needs of real-time, decentralized execution systems. Their ability to provide fast, compact, and verifiable output with minimal overhead makes them the preferred model for securing AI-generated data in production deployments.

5.1 Threshold Signatures (Preferred Model)

The Threshold AI Oracle system adopts threshold cryptographic signatures, such as Boneh–Lynn–Shacham (BLS) schemes, as its primary mechanism for output validation [20]. This approach offers an advantageous combination of computational efficiency, cryptographic security, and compatibility with decentralized architectures. Each participating agent or committee holds a share of the signing key, and a valid output is produced only when a predefined quorum agrees on the interpretation of a given event and jointly generates a single aggregate signature.

This signing model enables rapid verification on-chain, often within milliseconds. Such speed is essential for real-time decentralized finance applications, where operations like liquidations must be executed immediately in response to external triggers [14]. Delays in these scenarios can lead to missed opportunities or financial loss. The compact size of threshold signatures also minimizes gas consumption, which is a critical factor in maintaining scalability.

The protocol supports flexible quorum thresholds, allowing applications to adjust the required level of consensus based on the criticality of the task. It is well suited to modular, multi-agent systems, where committees may be updated or rotated without requiring full key recomputation [21]. This adaptability supports seamless integration with dynamic agent networks and evolving governance structures.

In addition to performance and scalability, threshold signatures enhance system security. By incorporating agent diversity and economic staking, the protocol increases the cost and complexity of collusion [22]. Agents are incentivized to behave honestly, as successful manipulation requires coordination among multiple participants with verifiable economic commitments. These properties make threshold signatures the preferred model for producing fast, trustworthy, and cryptographically secure outputs in the Threshold AI Oracle framework.

5.2 Zero-Knowledge Proofs (ZK-LLM, zkML)

Zero-knowledge proofs (ZKPs) provide a method for verifying that a computational process produced a specific output without disclosing the input data or the internal steps taken to generate the result [35]. This approach offers strong guarantees for privacy and correctness, making it particularly attractive for applications that require transparency without compromising confidentiality. Use cases include regulatory compliance, off-chain computation audits, and the validation of sensitive operations [36].

Despite their theoretical strengths, ZKPs remain impractical for real-time oracle systems. Generating a zero-knowledge proof for computationally intensive AI tasks, such as natural language understanding or multi-agent deliberation, typically requires several minutes to complete [6]. Even optimized frameworks, including distributed zkSNARK systems, face difficulties in reducing latency to acceptable levels for real-time use cases [37]. Additionally, these systems often require specialized cryptographic hardware or custom circuit designs, which increases the complexity and cost of deployment.

Another limitation is that most zero-knowledge proof systems are designed to operate over deterministic logic. This restricts their applicability in contexts where AI agents must engage in probabilistic reasoning or interpret ambiguous data. The non-deterministic nature of deliberative multi-agent processes complicates circuit generation and undermines the reproducibility required for succinct proofs.

Given these challenges, ZKPs are better suited for ex post verification of AI outputs rather than for real-time execution environments. They may be appropriate for auditing high-stakes decisions, confirming model integrity after the fact, or securing offline computation, but are currently not viable as the primary cryptographic verification mechanism for event-driven, low-latency applications such as those supported by the Threshold AI Oracle protocol.

5.3 Trusted Execution Environments (TEEs)

Trusted Execution Environments (TEEs) provide a hardware-based solution for ensuring computation integrity. Technologies such as Intel SGX and AMD SEV allow programs to run within secure enclaves that isolate execution from the rest of the system [38, 39]. This architecture aims to prevent unauthorized access

or tampering during runtime, making it possible to run sensitive computations, such as AI inference, while maintaining confidentiality and integrity.

TEEs offer strong performance advantages. Their ability to generate and return results with minimal latency makes them attractive for applications that require real-time responsiveness. In latency-sensitive scenarios, such as automated trading or protocol-triggered DeFi liquidations, TEEs can meet strict timing constraints that other verification mechanisms struggle to satisfy.

However, these benefits come with significant trade-offs. The internal operation of TEEs is opaque to external observers, and their correctness ultimately depends on the security of proprietary hardware and firmware. This introduces a centralized trust dependency that conflicts with the foundational principles of decentralized systems [8]. TEEs cannot be independently audited by the broader community, which limits transparency and increases systemic risk in the event of hardware-level exploits.

In addition, TEEs are vulnerable to a range of attacks, including side-channel and supply chain compromises [26]. These risks are particularly concerning in permissionless networks, where adversaries may possess strong incentives and substantial resources. For these reasons, while TEEs may be appropriate in controlled enterprise settings or specific off-chain applications, they are treated as a fallback mechanism only within the Threshold AI Oracle protocol. The architecture prioritizes cryptographic verification methods that are transparent, decentralized, fast, and publicly auditable.

5.4 Comparative Summary

Verification models must be evaluated based on their speed, transparency, auditability, and suitability for real-time use. Table 1 summarizes these tradeoffs [3].

Model	**Speed**	**Transparency**	**Auditability**	**UX Readiness**	**Suitability**
Threshold Signatures	Fast (ms)	High (committee logs)	Partial (can log debate)	Ready now	Preferred default
ZK Proofs	Slow (mins+)	Very high	Full	Poor latency	Future-proofed but early
TEEs	Fast	Opaque	Limited	Mixed	Last-resort only

Table 1: Comparison of cryptographic verification models

5.5 Designing for the End User: Why Speed Wins

In many decentralized applications, response time is a critical determinant of utility and user trust. In decentralized finance (DeFi), mechanisms such as automated liquidations and collateral rebalancing must respond to market fluctuations within seconds [14]. Even minor delays in execution can lead to significant financial losses or missed arbitrage opportunities. Similarly, in governance contexts, decentralized autonomous organizations (DAOs) must be able to respond promptly to verified external events, such as regulatory updates or geopolitical developments, in order to initiate proposals, enact policy changes, or defend against protocol threats [40].

End users expect oracle systems to deliver data with a level of speed and reliability comparable to that of centralized service providers. At the same time, they require guarantees that the delivered outputs are cryptographically verifiable, tamper-resistant, and generated through secure, consensus-driven processes. Achieving both low latency and high trust is therefore essential for oracle infrastructure intended to serve real-time decentralized applications.

Threshold signatures fulfill these dual requirements. They enable rapid verification of consensus-based outputs and are computationally efficient to validate on-chain [20]. Their compact structure reduces-outline gas costs, and their compatibility with quorum-based governance allows seamless integration into existing smart contract workflows. Because threshold signatures are generated only after a defined set of agents reaches agreement, they provide high assurance of correctness without introducing significant delays.

By contrast, alternatives such as zero-knowledge proofs and trusted execution environments may offer useful properties in specific scenarios but are not yet capable of meeting the strict latency and transparency requirements of event-driven blockchain systems. For this reason, threshold signatures are prioritized within the Threshold AI Oracle architecture as the primary mechanism for achieving fast, secure, and scalable oracle outputs.

6. Real-World Use Cases

The Threshold AI Oracle system enables the transformation of real-world events into cryptographically verified on-chain triggers, facilitating automated decision-making across a wide array of decentralized applications. By combining AI-driven deliberation with threshold cryptographic signatures [6], the protocol supports a variety of domains, including prediction markets, decentralized finance (DeFi), and decentralized governance.

Each application scenario is aligned with the system's phased implementation strategy, as described in Section 4 [3]. In the initial phase, the protocol produces verifiable numerical outputs suitable for discrete decision-making. The intermediate phase introduces structured textual commands to support more granular automation. The long-term phase expands capabilities to include the generation of executable logic, allowing smart contracts to adapt dynamically to complex external conditions.

The use cases presented in this section demonstrate the protocol's capacity to handle sophisticated queries, produce structured outputs, and execute smart contract logic with high integrity. These examples also highlight the system's advantages over traditional oracle architectures such as UMA, which have been shown to be vulnerable to governance-related manipulation [41]. Through this approach, the Threshold AI Oracle offers a more resilient and flexible foundation for event-driven automation in decentralized systems.

6.1 Request to Response to Action

The Threshold AI Oracle system functions as a seamless end-to-end pipeline that converts user or protocol-submitted queries into verifiable smart contract outcomes [8]. A query typically specifies a condition to monitor, such as the resolution of a prediction market or a shift in market sentiment, and defines the expected output format. This output may be a numerical value, a structured textual command, or executable smart contract logic, depending on the query's complexity and the desired automation behavior.

Each query is assigned a unique identifier and is linked to a corresponding execution layer on the blockchain. Once submitted, specialized AI agent committees are tasked with evaluating the query by sourcing relevant external data, interpreting its significance, and reaching consensus through structured deliberation. When quorum is achieved, the resulting decision is encoded as a cryptographically signed output and published on-chain. The smart contract associated with the original query identifier then executes a predefined action based on the verified result.

This low-latency, deterministic process allows the system to support a variety of application types across its phased implementation model [14]. In the initial phase, the protocol supports simple prediction market resolutions using discrete numerical outputs. The intermediate phase enables automated DeFi operations through structured commands. The long-term phase extends support to adaptive governance mechanisms powered by AI-generated logic. In each case, the system provides a trust-minimized path from off-chain event detection to on-chain execution.

6.2 Event-Driven DeFi Automation

In decentralized finance, the ability to respond quickly to market conditions is essential for maintaining protocol stability and competitiveness [14]. The Threshold AI Oracle system enables automated decision-making based on multiple variables by leveraging structured textual commands generated during the intermediate phase of its implementation model. This capability supports complex condition-based triggers that can influence lending rates, collateral requirements, or trading strategies.

Consider a DeFi protocol that adjusts interest rates based on two independent factors: market volatility and sentiment extracted from news and social media. The protocol submits a query specifying both variables as inputs. Two separate AI agent committees are assigned to evaluate each input. One committee analyzes historical and real-time market data to compute volatility metrics. The other committee processes social media trends and news sentiment using natural language processing techniques [6].

Each committee operates in parallel. Agents deliberate and produce proposed outputs, which are then clustered using embedding models and verified through pairwise Natural Language Inference (NLI) tests [29]. Once at least $(f + 1)$ agents converge on a coherent interpretation, a threshold signature is generated and the signed command is submitted on-chain. The smart contract reads the structured output and adjusts lending rates according to the specified parameters.

To defend against adversarial manipulation of input data, the system incorporates model diversity, randomized agent assignments, and quorum-based validation, as outlined in Section 6 [26]. These mechanisms enhance both the accuracy and integrity of the automation process, ensuring that protocol behavior remains trustworthy even under rapidly changing or adversarial conditions.

6.3 Dynamic DAO Governance

The Threshold AI Oracle system enables decentralized autonomous organizations (DAOs) to dynamically adjust their operations in response to real-world developments by generating executable logic in the long-term phase of the protocol [40]. This capability allows DAOs to go beyond static rule enforcement and adopt responsive governance mechanisms that are capable of incorporating external inputs, such as legal or regulatory events, into their decision-making frameworks [22].

Consider a DAO that manages a decentralized platform and must update its governance policy following a regulatory change [22]. The DAO submits a query to the oracle system requesting verification of the regulatory development and requests executable logic to update its governance contract accordingly.

AI agent committees are assigned to monitor relevant information sources, including government websites, legal commentary, and financial news. The evaluation proceeds through a multi-stage pipeline involving natural language processing, deliberation over interpretation, summarization of agent opinions, crafting of logic, and agreement on input parameters [16]. Once a consensus of at least $(f + 1)$ agents is achieved, the output is signed and delivered on-chain as structured executable code.

The smart contract receives the logic and applies it to update its governance parameters in real time. This process is safeguarded through extensive validation techniques, including formal verification methods [33] and adversarial testing, to ensure that the generated logic is syntactically valid, semantically consistent, and resistant to manipulation. This model of adaptive governance allows DAOs to maintain compliance, react to external developments, and continuously evolve while preserving the security guarantees of a decentralized system.

6.4 AI-Resolved Prediction Markets

Prediction markets benefit significantly from the Threshold AI Oracle system's ability to resolve subjective or complex queries through verifiable numerical outputs, particularly in the initial phase of protocol deployment [2, 3]. Unlike traditional oracles that rely on static data feeds or human governance mechanisms, the system provides a trust-minimized and automated resolution pathway for event-based markets.

Consider a prediction market contract structured around the question of whether a trade agreement will be signed by a specified deadline. The contract presents three possible outcomes: Option A (signed), Option B (delayed), and Option C (canceled), which are mapped to numerical values of 1, 2, and 3, respectively.

Once the query is submitted, AI agent committees are tasked with evaluating the outcome. These agents analyze a variety of external data sources, including diplomatic press releases, financial news, and government statements. Through deliberation and aggregation, each agent submits a proposed outcome. The system computes the median of $(2f + 1)$ submissions, ensuring robustness against up to f potentially faulty or

malicious inputs [15]. After consensus is reached, the result is signed using a threshold signature and published on-chain.

The corresponding smart contract uses the signed output to execute a payout according to the winning option. This process offers greater security and efficiency compared to oracles that depend on token-weighted voting or human arbitration. Additionally, committee randomization and data diversity mitigate the risk of input manipulation, providing a more reliable foundation for decentralized prediction markets.

6.5 Sentiment-Based DeFi Automation

The Threshold AI Oracle system extends the capabilities of decentralized finance by enabling automated smart contract actions based on sentiment analysis. In the initial phase of the protocol, sentiment-driven outputs are expressed as verifiable numerical scores, allowing contracts to adjust behavior in response to public perception or market mood [42, 3].

Consider a DeFi protocol that modifies portfolio allocations based on the prevailing sentiment toward Ethereum. The protocol submits a query requesting a sentiment score on a scale from 1 to 10. AI agent committees are activated to evaluate this request, sourcing data from social media platforms, news articles, and financial market commentary.

Each agent processes a subset of the data and proposes a sentiment score. Once $(2f + 1)$ opinions have been submitted, the system computes the median score, ensuring resilience against up to f faulty or adversarial contributions [28]. The final result is secured through a threshold signature and published on-chain.

The smart contract receives the signed sentiment score and triggers predefined logic based on the value. For instance, if the score falls below a certain threshold, the protocol may reallocate assets into stablecoins. If sentiment is strong, the contract may increase exposure to risk-on assets. This enables responsive asset management strategies that are informed by real-time external data.

Robustness against manipulation is achieved through a combination of data source diversity, randomized committee composition, and quorum-based validation. Compared to traditional oracles that rely on single-source feeds or centralized data aggregators [4, 8], the system offers a more secure and adaptable mechanism for incorporating sentiment into decentralized financial infrastructure.

6.6 AI-Augmented Escrow and Dispute Resolution

In digital labor markets and service-based smart contracts, dispute resolution is a critical function that often determines the reliability of on-chain agreements. The Threshold AI Oracle system supports this functionality by acting as an autonomous arbitration layer [41]. During the intermediate phase of the protocol, structured textual commands are used to direct contract behavior based on the outcome of AI-driven deliberation.

Consider a scenario in which a software developer delivers a prototype as part of a contractual engagement. A dispute arises concerning whether the delivered work meets the agreed-upon specifications. The platform submits a query requesting resolution and expects a command that will either release or refund escrowed funds.

AI agent committees are formed to evaluate the claim. They examine the submitted codebase, contractual documentation, and any related communication or technical specifications. Through deliberation, the agents submit candidate responses, which are then clustered using embedding-based similarity models [29]. Once at least $(f + 1)$ agents converge on a coherent decision, the output is signed and published on-chain.

The resulting structured command may include an instruction such as `release_escrow` or `initiate_refund`, depending on the findings. This triggers the appropriate financial transaction through the escrow contract. In cases of high ambiguity or ethical sensitivity, human oversight may be introduced to supplement or validate the AI-generated output [43].

7. Limitations and Open Challenges

The Threshold AI Oracle system introduces a novel architecture for decentralized intelligence by integrating AI-driven deliberation with cryptographic consensus. This combination enables the production of structured, actionable outputs for smart contracts that respond to complex real-world conditions. While the system presents significant advances over traditional oracle mechanisms, it also faces a range of technical, conceptual, and social challenges that must be addressed to ensure long-term robustness and broad adoption [3].

Key limitations include the difficulty of aggregating diverse textual responses into coherent outputs, particularly in the intermediate phase where natural language interpretation plays a central role. The handling of subjective or culturally dependent inputs remains a complex problem, especially in applications related to ethics, governance, or regulatory compliance [40]. Furthermore, the tradeoff between latency and deliberative rigor presents challenges for use cases that demand both speed and interpretive accuracy.

Additional concerns involve model reliability and drift, particularly as AI agents evolve in dynamic data environments. Security vulnerabilities, such as adversarial attacks on model embeddings or coordinated collusion within agent committees, require ongoing research and mitigation strategies. Legal ambiguity surrounding autonomous decision-making and smart contract execution further complicates the system's deployment in jurisdictions with evolving regulatory frameworks.

Finally, questions of equitable access and inclusion must be addressed to prevent the system from reinforcing existing power asymmetries. Without careful protocol design, economic barriers could limit participation in staking, validation, or governance, reducing the diversity of perspectives and increasing systemic risk.

These limitations are most evident during the system's phased implementation as outlined in Section 4. Addressing them will require interdisciplinary collaboration across the domains of cryptography, AI safety, governance design, and law. The remainder of this section presents a structured analysis of these challenges, outlines mitigation strategies, and highlights opportunities for future research and refinement.

7.1 Aggregating Textual Deliberation

The aggregation of natural language outputs is a central challenge in the intermediate phase of the Threshold AI Oracle protocol, where agent committees generate structured textual commands. Unlike numerical outputs, which can be reliably consolidated through statistical functions such as median computation, textual responses exhibit significant variability in terminology, syntactic structure, and expression of certainty. This variability makes it difficult to determine when logically equivalent conclusions are semantically aligned [6].

To address this challenge, the protocol aims to use embedding models to map textual outputs into high-dimensional vector spaces, enabling semantic comparison [29]. It then applies pairwise Natural Language Inference (NLI) tests to detect entailment and contradiction among candidate responses. A valid summary is produced when at least $(f + 1)$ agents submit responses that fall within a coherent semantic cluster.

Despite this framework, aggregating textual data remains an open research problem. Semantic similarity thresholds are difficult to calibrate across contexts, and clustering outcomes may vary depending on model architecture or tokenization strategy. In addition, adversarial agents may exploit embedding weaknesses to introduce syntactically similar but logically divergent outputs, disrupting consensus formation [26].

Current mitigation strategies include the use of diverse AI models within committees, randomized agent assignments, and adversarial training during model fine-tuning. In more complex or ambiguous cases, the protocol may trigger fallback mechanisms such as human-in-the-loop review to validate the output [43]. Ongoing research aims to improve the robustness and interpretability of semantic embedding techniques to support more reliable aggregation of natural language deliberation.

7.2 Trust in Subjective Outcomes

Subjective queries present a unique challenge to the deterministic nature of smart contract execution. Questions involving interpretive judgment, such as whether a regulatory announcement improves market stability, often lack a single objective answer. These types of queries become especially relevant in the intermediate

and long-term phases of the protocol, where AI committees are tasked with generating structured textual commands or executable logic [44].

Interpretation of such events may vary significantly across legal jurisdictions, cultural contexts, and political environments. A policy change that is considered beneficial in one context may be seen as detrimental in another. As a result, outputs derived from subjective queries are inherently more susceptible to disagreement among participants and require additional layers of validation and transparency.

Prior examples from governance-based oracle systems, such as UMA, demonstrate the risks of relying exclusively on token-weighted voting to resolve ambiguous outcomes. The $7 million Polymarket incident illustrates how powerful stakeholders can influence decisions in ways that undermine system neutrality and user trust [41]. While the Threshold AI Oracle protocol reduces this risk through cryptographic consensus and diverse agent composition, it must still account for the need to resolve subjectivity in a fair and auditable manner.

To address this challenge, the protocol plans to include support for appeals, optional human oversight, and the generation of transparent audit trails. Future research may focus on formalizing measures of interpretive confidence, including subjectivity scoring, or introducing stakeholder-weighted consensus models that incorporate the preferences of affected parties. These enhancements could increase system reliability in domains where qualitative reasoning plays a central role.

7.3 Latency Versus Rigor Tradeoffs

The phased output structure of the Threshold AI Oracle system introduces an inherent tradeoff between latency and computational rigor. Numerical outputs, which are prevalent in the initial phase, can be generated and verified rapidly. These outputs are well suited for high-frequency applications such as decentralized finance automation, where execution speed is critical [14].

In contrast, outputs from the intermediate and long-term phases (structured textual commands and executable logic) require multi-stage deliberation and validation. These steps involve complex reasoning, semantic clustering, and formal verification, which significantly increase response time. While these outputs offer greater expressiveness and contextual sensitivity, they may be unsuitable for applications that require near-instantaneous decision-making.

To balance these competing demands, the protocol supports configurable execution paths based on the risk profile of the query [15]. For example, a governance proposal involving a smart contract migration may tolerate several minutes of processing time to ensure correctness. In contrast, a portfolio rebalancing operation in response to volatile market conditions may require immediate execution to avoid financial loss.

The system can be designed to address these variations by allowing developers to specify acceptable latency thresholds and precision requirements at the time of query submission. Adaptive quorum models can be applied to accelerate low-risk decisions while preserving high-rigor protocols for critical or ambiguous tasks. This flexibility enables the protocol to serve a broad range of use cases without sacrificing performance or reliability.

7.4 AI Model Drift and Decay

AI model performance may degrade over time due to a phenomenon known as drift, which occurs when models trained on historical data fail to generalize to new or evolving contexts. In decentralized finance applications, for instance, changes in Ethereum market dynamics or sentiment patterns may reduce the accuracy of outputs generated by previously reliable models [6].

To maintain system reliability, the Threshold AI Oracle protocol plans to incorporate mechanisms for detecting and mitigating model drift. These include version-controlled retraining workflows, real-time performance telemetry, and community oversight during model updates [45, 46]. Periodic evaluation against benchmark tasks and synthetic test sets allows for the identification of underperforming models before they impact on-chain operations.

Transparency is further enhanced through the use of structured audit logs and publicly accessible model documentation. Model cards [44] provide standardized metadata on training data, intended use, known limitations, and observed failure modes. This documentation supports both developer review and community governance processes, fostering accountability across the agent lifecycle.

Future research may explore the application of continual learning algorithms, dynamic testing frameworks, and adversarial robustness techniques to improve long-term model performance. These tools will be essential to ensuring that the system remains adaptive and trustworthy as the underlying data environment evolves.

7.5 Adversarial Attacks on Agents or Committees

The Threshold AI Oracle system relies on autonomous agents and external data inputs, which introduces vulnerabilities to various forms of adversarial manipulation. These include Sybil attacks, misinformation campaigns, and data poisoning, as well as spoofing of structured and unstructured inputs [28]. For example, a prediction market query that depends on social media sentiment could be misled by coordinated bot activity or artificially amplified narratives.

To defend against such threats, the protocol incorporates multiple layers of mitigation. Randomized committee composition reduces the predictability of agent assignments, making it more difficult for adversaries to coordinate attacks on specific decision processes [15]. Behavior profiling and historical performance tracking further support the identification of anomalous agent behavior [3]. Agents that exhibit suspicious patterns may be subject to slashing penalties or temporary exclusion from quorum participation.

Anomaly detection systems can be deployed to monitor the integrity of inputs, especially in high-sensitivity applications such as decentralized finance and governance. Despite these safeguards, advanced attacks on language model embeddings and clustering mechanisms remain an ongoing concern [26]. Such attacks may exploit subtle manipulations in phrasing or context to bias semantic similarity measures and disrupt consensus formation.

Investment in adversarial robustness is essential to maintaining the reliability of the system under adversarial pressure. Future work will require collaboration across machine learning, cryptography, and security research to strengthen model interpretability, enhance input verification, and improve the resilience of AI agents operating in decentralized environments.

7.6 Human Oversight and Legal Ambiguity

Some queries processed by the Threshold AI Oracle system involve ethical, normative, or legal dimensions that cannot be fully addressed through automated reasoning. For example, determinations involving reputational harm, regulatory compliance, or policy alignment often require human interpretation to ensure fairness and contextual sensitivity [43]. While the protocol supports human-in-the-loop mechanisms, their inclusion raises important concerns about accountability, potential bias, and the reintroduction of centralization.

The integration of human oversight presents both procedural and governance challenges. Decisions involving human input must be auditable and subject to community review to preserve the transparency and decentralization goals of the protocol [40]. Additionally, the criteria for when human intervention is permitted or required must be clearly defined to avoid arbitrary override of automated processes.

Legal ambiguity further complicates the implementation of autonomous decision systems. It is currently unclear who bears legal responsibility when a smart contract executes a faulty or harmful action based on an AI-generated output. The distributed nature of agent-based deliberation and consensus mechanisms obscures traditional notions of liability, making it difficult to assign fault or enforce accountability [47, 22].

To address these concerns, future protocol designs should explore decentralized review systems that support layered validation without compromising openness. Legal frameworks must also evolve to recognize the collective and procedural nature of decentralized intelligence. This includes defining the roles and obligations of protocol contributors, validators, and governance participants in relation to dispute resolution, regulatory compliance, and user protection.

7.7 Cost, Access, and Fairness

The computational and operational demands associated with complex queries in the Threshold AI Oracle system may present barriers to participation for smaller protocols and resource-constrained users [48]. Intermediate- and long-term phase outputs, which require advanced processing such as semantic clustering or logic generation, may incur higher costs due to the intensive use of AI agents and verification infrastructure. This cost structure creates a risk of exclusion, where only well-capitalized entities can afford to access advanced oracle functionality.

Such dynamics could undermine the decentralization goals of the protocol by concentrating decision-making capabilities among a small subset of economically privileged actors. If left unaddressed, this concentration may lead to reduced diversity in agent composition, weaker system resilience, and diminished trust in outcomes.

To promote broader accessibility, the protocol may implement pricing strategies such as tiered query fees based on complexity or urgency. Quadratic funding mechanisms [49] can be used to prioritize community-valued queries and support underrepresented use cases. In addition, protocol-level subsidies for public goods, such as governance resolution or market transparency, may help distribute costs more equitably.

Commons-based governance models [50, 46] and open resource allocation frameworks offer further opportunities for enhancing fairness. These approaches prioritize inclusivity and collective benefit, making them well suited to guide the equitable evolution of decentralized oracle infrastructure. Future protocol iterations should explore how these models can be integrated to ensure access and fairness across economic, geographic, and institutional boundaries.

8. Conclusion

The Threshold AI Oracle system redefines the role of oracles within the Web3 ecosystem [3], moving beyond passive data relays to introduce an intelligent, deliberative architecture capable of interpreting and acting upon complex real-world events. Traditional oracle systems, which depend on static feeds, continuous data polling, and opaque validation mechanisms, are prone to inefficiencies, rigidity, and governance vulnerabilities [4, 41]. In contrast, the Threshold AI Oracle system offers a selective, context-aware design grounded in cryptographic verifiability and optimized for actionable outputs. This model supports a new class of real-time, trust-minimized automation in decentralized applications.

As described in Section 4, the system progresses through a phased implementation model [6]. The initial phase produces numerical outputs, enabling applications such as prediction market resolutions. The intermediate phase introduces structured textual commands for more granular automation in DeFi. The final phase supports logic generation, empowering dynamic and adaptive governance frameworks [22]. Outputs are generated by multi-agent committees that deliberate in real time and reach quorum-based consensus. Each result is finalized through threshold cryptographic signatures [20], ensuring secure and efficient execution with minimal overhead.

Illustrative examples include prediction market queries resolved by analyzing diplomatic data to produce a signed result such as {“value”: 1}, or DeFi protocols using real-time sentiment scores like {“value”: 7} to adjust asset allocations. In governance contexts, DAOs may receive executable code such as {“logic”: “function update_policy(compliance_level, contract) { if (compliance_level > 80) enable_new_policy(contract); }”}, allowing on-chain rules to evolve in response to external regulatory developments.

Functioning as an epistemic layer, the Threshold AI Oracle system transforms unstructured data into structured knowledge [11]. It synthesizes diverse, real-world sources to produce reliable, interpretable outputs that drive on-chain execution. As discussed in Section 8, this capability underpins a broad range of applications, from market forecasting [2] to adaptive governance [40]. Compared to conventional oracles, the system offers superior resistance to manipulation and improved decision quality through structured deliberation.

Nonetheless, as outlined in Section 9, the protocol faces ongoing challenges. These include aggregating heterogeneous textual inputs, adjudicating subjective judgments, balancing low-latency requirements with interpretive rigor, and addressing model drift [44]. Adversarial threats, such as embedding manipulation

in textual summarization [26], also present non-trivial risks. These challenges are particularly acute in the intermediate and long-term phases, and require continued research in areas such as AI summarization, cryptographic verification, and fault-tolerant system design.

Deployed across scalable blockchain execution layers, the Threshold AI Oracle protocol aims to provide a foundation for composable, secure, and intelligent automation. It creates opportunities for interdisciplinary collaboration [16]. Researchers are encouraged to investigate the dynamics of agent behavior, consensus in language models, and model robustness. Developers can contribute by refining summarization pipelines, improving adversarial defenses, and enhancing protocol performance. Governance participants and community members will play a central role in designing inclusive structures that preserve fairness and systemic resilience [46].

Through these collective efforts, the Threshold AI Oracle system aspires to establish a new paradigm for decentralized intelligence. Its design equips Web3 infrastructures with the ability to interpret and respond to external events autonomously, opening the path to blockchain ecosystems that are as responsive, context-aware, and adaptable as the societies they aim to support.

References

[1] K. Salah, M. H. u. Rehman, N. Nizamuddin, and A. Al-Fuqaha, "Blockchain for ai: Review and open research challenges," *IEEE Access*, vol. 7, pp. 10 127–10 149, 2019.

[2] J. Peterson, J. Krug, M. Zoltu, M. Williams, and S. Alexander, "Augur: A decentralized oracle and prediction market platform," https://www.augur.net/whitepaper.pdf, 2018, accessed: 2025-05-09.

[3] Supra Research, "Dora: Distributed oracle agreement," https://supra.com/documents/SupraOracles-DORA-Whitepaper.pdf, 2023, accessed: 2025-05-11.

[4] S. Ellis, A. Juels, and S. Nazarov, "Chainlink: A decentralized oracle network," https://chain.link/whitepaper, 2017, accessed: 2025-05-09.

[5] S. Reynolds, "Polymarket suffers uma governance attack after rogue actor becomes top-5 token staker," https://www.coindesk.com/markets/2025/03/26/polymarket-suffers-uma-governance-attack-after-rogue-actor-becomes-top-5-token-staker, 2025, accessed: 2025-05-09.

[6] Z. Xi, W. Chen, X. Guo, W. He, Y. Ding, B. Hong, M. Zhang, J. Tang, G. Zheng, and W. Wang, "The rise and potential of large language model based agents: A survey," arXiv preprint arXiv:2309.07864, 2023. [Online]. Available: https://arxiv.org/abs/2309.07864

[7] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," https://bitcoin.org/bitcoin.pdf, 2008, accessed: 2025-05-09.

[8] F. Zhang, E. Cecchetti, K. Croman, A. Juels, and E. Shi, "Town crier: An authenticated data feed for smart contracts," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 270–282, 2016.

[9] A. Pasdar, Y. C. Lee, and Z. Dong, "Connect api with blockchain: A survey on blockchain oracle implementation," *ACM Computing Surveys*, October 2022.

[10] OpenAI, "Gpt-4 technical report," arXiv preprint arXiv:2303.08774, 2023. [Online]. Available: https://arxiv.org/abs/2303.08774

[11] Y. Liu, Q. Lu, L. Zhu, and H.-Y. Paik, "Decentralised governance for foundation model based ai systems: Exploring the role of blockchain in responsible ai," *IEEE Software*, 2024.

[12] Y. Cao, S. Li, Y. Liu, Z. Yan, Y. Dai, P. S. Yu, and L. Sun, "A comprehensive survey of ai-generated content: Challenges and opportunities," arXiv preprint arXiv:2306.00619, 2023. [Online]. Available: https://arxiv.org/abs/2306.00619

[13] V. Buterin, "Ai agents and decentralized governance: Scaling coordination in web3," Speech at ETH Denver 2025, quoted in ChangeHero Blog, https://changehero.io/blog/emerging-blockchain-development-trends-for-2025-and-beyond, 2025, accessed: 2025-05-09.

[14] Aave Protocol, "Aave protocol whitepaper," https://github.com/aave/aave-protocol/blob/master/docs/Aave_Protocol_Whitepaper_v1_0.pdf, 2020, accessed: 2025-05-09.

[15] L. Lamport, R. Shostak, and M. Pease, "The byzantine generals problem," *ACM Transactions on Programming Languages and Systems*, vol. 4, no. 3, pp. 382–401, 1982.

[16] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley, 2009.

[17] N. Shavit and D. Touitou, "Software transactional memory," *Proceedings of the Fourteenth Annual ACM Symposium on Principles of Distributed Computing*, pp. 204–213, 1995.

[18] M. Herlihy and J. E. B. Moss, "Transactional memory: Architectural support for lock-free data structures," *Proceedings of the 20th Annual International Symposium on Computer Architecture*, pp. 289–300, 1993.

[19] P. A. Bernstein and E. Newcomer, *Principles of Transaction Processing*, 2nd ed. Morgan Kaufmann, 2009.

[20] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Advances in Cryptology — ASIACRYPT 2001*, pp. 514–532, 2001.

[21] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme," *Public Key Cryptography — PKC 2003*, pp. 31–46, 2003.

[22] V. Buterin, "Ethereum: A next-generation smart contract and decentralized application platform," https://ethereum.org/en/whitepaper/, 2014, accessed: 2025-05-09.

[23] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[24] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Pearson, 2020.

[25] E. Elkind and J. Rothe, "Cooperative game theory," in *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, Eds. Cambridge University Press, 2016, pp. 135–159.

[26] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv preprint arXiv:1412.6572, 2015. [Online]. Available: https://arxiv.org/abs/1412.6572

[27] L. Lamport, "Time, clocks, and the ordering of events in a distributed system," *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978.

[28] J. R. Douceur, "The sybil attack," *International Workshop on Peer-to-Peer Systems*, pp. 251–260, 2002.

[29] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, 2015.

[30] S. Amani, M. Bégel, M. Bortin, and M. Staples, "Towards verifying ethereum smart contract bytecode in isabelle/hol," *Proceedings of the 7th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pp. 66–77, 2018.

[31] I. Sommerville, *Software Engineering*, 10th ed. Pearson, 2015.

[32] M. Castro and B. Liskov, "Practical byzantine fault tolerance," *Proceedings of the Third Symposium on Operating Systems Design and Implementation*, pp. 173–186, 1999.

[33] E. M. Clarke, T. A. Henzinger, H. Veith, and R. Bloem, *Handbook of Model Checking*. Springer, 2018.

[34] V. D'Silva, D. Kroening, and G. Weissenbacher, "A survey of automated techniques for formal software verification," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 7, pp. 1165–1178, 2008.

[35] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, vol. 18, no. 1, pp. 186–208, 1989.

[36] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," *2014 IEEE Symposium on Security and Privacy*, pp. 459–474, 2014.

[37] H. Wu, W. Zheng, A. Chiesa, R. A. Popa, and I. Stoica, "Dizk: A distributed zero knowledge proof system," https://www.usenix.org/conference/usenixsecurity20/presentation/wu, pp. 2565–2582, 2020.

[38] V. Costan and S. Devadas, "Intel sgx explained," IACR Cryptology ePrint Archive, 2016/086, https://eprint.iacr.org/2016/086, 2016.

[39] M. Sabt, M. Achemlal, and A. Bouabdallah, "Trusted execution environment: What it is, and what it is not," *2015 IEEE Trustcom/BigDataSE/ISPA*, pp. 57–64, 2015.

[40] S. Hassan and P. De Filippi, "Decentralized autonomous organization (dao)," *Internet Policy Review*, vol. 10, no. 2, 2021.

[41] S. Klinger and O. Marian, "Blockchain-based dispute resolution: Insights and challenges," *Stanford Journal of Blockchain Law & Policy*, vol. 2, no. 1, pp. 45–67, 2019.

[42] H. M. Kim and J. Lee, "Blockchain-based sentiment analysis for financial markets," *IEEE Access*, vol. 8, pp. 155 110–155 120, 2020.

[43] S. Amershi, D. Weld, M. Vorvoreanu, A. Fourney, B. Nushi, P. Collisson, J. Suh, S. Iqbal, P. N. Bennett, K. Inkpen, J. Teevan, R. Kikin-Gil, and E. Horvitz, "Guidelines for human-ai interaction," *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–13, 2019.

[44] M. Mitchell, S. Wu, A. Zaldivar, P. Barnes, L. Vasserman, B. Hutchinson, E. Spitzer, I. D. Raji, and T. Gebru, "Model cards for model reporting," *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pp. 220–229, 2019.

[45] A. D'Amour, K. Heller, D. Moldovan, B. Adlam, B. Alipanahi, A. Beutel, C. Chen, J. Deaton, J. Eisenstein, M. D. Hoffman *et al.*, "Underspecification presents challenges for credibility in modern machine learning," arXiv preprint arXiv:2011.03395, https://arxiv.org/abs/2011.03395, 2020.

[46] L. Aragon, A. Dafoe, B. Ding, and S. Farquhar, "Governance of ai systems: Lessons from open-source software," *AI & Society*, 2022.

[47] P. Hacker, "Regulating artificial intelligence: Legal and ethical challenges," *European Journal of Risk Regulation*, vol. 11, no. 4, pp. 717–728, 2020.

[48] S. Jain and Y. Chen, "Decentralized governance and market mechanisms for blockchain ecosystems," *Proceedings of the 2018 ACM Conference on Economics and Computation*, pp. 585–602, 2018.

[49] V. Buterin, Z. Hitzig, and E. G. Weyl, "A flexible design for funding public goods," *Management Science*, vol. 65, no. 11, pp. 5171–5187, 2019.

[50] E. Ostrom, *Governing the Commons: The Evolution of Institutions for Collective Action*. Cambridge University Press, 1990.

[51] J. Bethencourt, V. Shmatikov, and S. Savage, "Reputation-based trust management in peer-to-peer systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 2, pp. 132–147, 2007.

[52] J. A. Kroll, I. C. Davey, and E. W. Felten, "The economics of bitcoin mining, or bitcoin in the presence of adversaries," Proceedings of WEIS 2013, https://www.weis2013.econinfosec.org/papers/KrollDaveyFeltenWEIS2013.pdf, 2013.

[53] M. Carlsten, H. Kalodner, S. M. Weinberg, and A. Narayanan, "On the instability of bitcoin without the block reward," *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 154–167, 2016.

[54] S. P. Lalley and E. G. Weyl, "Quadratic voting: How mechanism design can radicalize democracy," *AEA Papers and Proceedings*, vol. 108, pp. 33–37, 2018.

A. Agent Staking and Crypto-Economic Incentives

To ensure the reliability of oracle outputs and the integrity of the decision-making process, the Threshold AI Oracle system incorporates a comprehensive crypto-economic framework. This framework combines financial and reputational mechanisms to hold agents accountable for their contributions [3]. Through staking requirements, performance-based reputation scoring, slashing penalties for misbehavior, proportional reward distribution, and market-driven committee selection, the protocol aims to create strong incentives for agents to act in alignment with system objectives [7, 22, 28].

These mechanisms serve multiple functions. Staking introduces economic risk for agents who participate in deliberation and signing processes, deterring malicious or negligent behavior. Reputation scores track the historical performance of each agent, influencing their eligibility for future participation and role assignments. Slashing protocols impose penalties for agents found to be contributing invalid, incoherent, or malicious outputs. At the same time, agents who contribute valid and timely responses receive rewards drawn from user-submitted fees, protocol subsidies, or redistributed penalties.

The resulting economic structure supports an environment in which high-quality performance is financially incentivized and systemic integrity is reinforced through distributed accountability. By embedding these economic constraints directly into the operational fabric of the protocol, the system strives to maintain resilience against manipulation and promotes consistency across a diverse and decentralized network of AI agents.

A.1 Agent Staking

All agents participating in the Threshold AI Oracle system are required to lock a financial stake as a form of collateral. This stake may be deposited directly by the agent or delegated through external participants. It functions as a guarantee of honest behavior and provides a measurable financial consequence for misrepresentation or failure to meet protocol standards. If an agent is found to have acted maliciously or persistently returned inaccurate outputs, part or all of its bonded assets may be subject to slashing penalties. This economic disincentive serves as a protective measure against manipulation and reduces the feasibility of Sybil attacks by increasing the cost of deploying large numbers of low-reputation agents [28].

Stake requirements are dynamic and can be calibrated according to the agent's assigned role or the economic weight of the query [49]. For instance, an agent tasked with evaluating the impact of a high-stakes geopolitical event on a decentralized finance vault may be required to post a larger stake proportional to the potential downstream effects of its decision. The protocol supports both native staking, which is integrated with the underlying validator infrastructure of the blockchain, and domain-specific staking, which ties capital to an agent's performance in particular functional areas.

An agent's stake also contributes to its on-chain reputation. Higher-value or long-duration stakes are interpreted as signals of reliability and commitment, perhaps increasing the agent's likelihood of being selected for critical roles [23]. In this way, staking not only introduces economic accountability but also potentially plays a central role in establishing a trust metric that influences agent selection and committee composition across the protocol.

A.2 Reputation and Performance Weighting

Reputation functions as a complementary mechanism to staking in evaluating agent trustworthiness within the Threshold AI Oracle system. It serves as a persistent, performance-based metric that reflects an agent's historical reliability, accuracy, and behavioral consistency. Each agent is assigned a composite reputation score, which is continuously updated based on quantifiable indicators such as quorum participation, time-to-response, vote alignment with validated outcomes, and peer or third-party evaluations [23, 51].

Agents with consistently high reputation scores may receive preferential treatment in protocol operations [48]. They are more likely to be assigned to critical queries, elevated to committee leadership roles, or selected as summarizers during deliberation. In addition, high-reputation agents may receive larger reward shares or attract more delegated stake for high-value or time-sensitive tasks. For example, an agent with a proven track record in resolving disputes related to climate data might be prioritized for future environmental or insurance-related queries.

Conversely, agents with low reputation scores could face operational disadvantages. These may include exclusion from committees assigned to high-risk queries, limited visibility in task marketplaces, or greater vulnerability to slashing for erroneous behavior. The reputation model introduces a form of evolutionary pressure that continuously filters out underperforming or unreliable agents from critical decision-making processes [52].

By rewarding sustained performance and penalizing inaccuracy or strategic misbehavior, the system ensures that agent selection is not solely determined by stake size but also by demonstrable competence. This hybrid model promotes both economic accountability and epistemic reliability across the network.

A.3 Rewards and Fee Structures

The Threshold AI Oracle system compensates agents based on the timeliness and accuracy of their participation in protocol activities [3]. The reward pool is funded through a combination of user-submitted query fees, protocol-level subsidies, and the redistribution of slashed collateral from agents who have violated performance or integrity standards [49]. This multi-source funding model ensures that incentives remain aligned with system reliability and that high-performing agents are consistently compensated.

Rewards are allocated using a performance-based distribution framework. Factors influencing reward magnitude include successful task completion, consistent participation in quorums, and adherence to latency thresholds. Agents who contribute meaningfully to the consensus process and help produce a valid, timely output are eligible for full compensation. In contrast, agents who abstain, delay their response, or fail to meet quality standards may receive only partial rewards or no compensation at all.

For instance, an agent tasked with validating a rapid market volatility event may receive a higher reward for contributing to a low-latency, high-stakes decision [14]. The protocol may dynamically adjust reward weightings to reflect query complexity and time sensitivity, thereby incentivizing not only correctness but also responsiveness.

By linking economic incentives to both outcome validity and execution speed, the system promotes active, high-quality participation. This design supports the overall responsiveness and integrity of the oracle network, particularly in applications requiring real-time data and condition-based automation.

To address the provenance of agent behavior, individual actions may be cryptographically signed using non-threshold signatures in addition to their contribution toward the aggregated threshold signature. This allows the protocol to verify agent participation in the final outcome without inflating on-chain verification costs. These auxiliary signatures can be aggregated or compressed into formats that are gas-efficient to verify, supporting both accountability and performance in resource-constrained execution environments.

A.4 Slashing and Challenge Mechanisms

To maintain the integrity of outputs and discourage inaccurate or malicious behavior, the Threshold AI Oracle system may incorporate a challenge-response protocol supported by slashing penalties [53]. This

mechanism allows any participant—including users, other agents, or third-party monitors—to dispute a signed output within a defined challenge window. Upon initiation of a challenge, a new committee could be assembled to re-evaluate the original query. In some cases, the system may require a higher quorum threshold for the review process to ensure greater rigor and consensus diversity.

If the review committee determines that the original output was incorrect, incoherent, or inconsistent with available evidence, agents involved in the disputed result are penalized. Slashing penalties are assessed proportionally, based on the stake size and voting behavior of the involved agents. For example, an agent that falsely reports the occurrence of a regulatory event to influence a DAO governance outcome may lose a portion of its bonded assets. The participant who initiated the challenge is eligible to receive a share of the slashed funds, thereby creating an economic incentive to monitor oracle outputs and report misconduct [22].

Slashing may be triggered by various forms of protocol deviation, including submission of incorrect data, logical inconsistency with previous results, or deliberate non-participation in quorum activities. This enforcement mechanism serves as an internal correction system that operates without requiring a centralized arbiter to establish ground truth. By embedding self-corrective capabilities into the oracle's governance and reward model, the system promotes transparency, accountability, and continuous improvement in decision quality.

A.5 Committee Composition via Economic Markets

The composition of agent committees within the Threshold AI Oracle protocol is influenced by market-driven mechanisms. Users and decentralized applications may allocate stake to individual agents or agent pools, thereby influencing their likelihood of inclusion in committee selection processes [48]. This structure introduces an open, economically mediated layer of agent selection that reflects both public confidence and historical performance [23].

To safeguard against centralization and collusion, governance parameters may enforce diversity constraints in committee composition [28]. For example, a quorum may be required to include a balance between validator-backed agents and those supported through independent delegation. Additionally, protocols with specialized needs, such as decentralized finance applications, may prioritize agents with domain-specific expertise in financial analysis. At the same time, the system may mandate the inclusion of agents tasked with adversarial review to ensure that dominant interpretations are subjected to challenge and scrutiny.

Over time, this model transforms committee formation into a competitive, reputation-sensitive process. Agents who consistently deliver accurate, timely, and verifiable outputs are more likely to attract stake and secure active roles in high-value tasks. Underperforming agents, by contrast, experience reduced participation and diminished influence. This market-based allocation of attention and responsibility fosters a dynamic intelligence marketplace that enhances both the security and adaptability of the protocol.

A.6 Economic Resilience and Attack Mitigation

The Threshold AI Oracle protocol incorporates structural safeguards to defend against economically motivated attacks and to ensure fair participation across agents. These defenses are implemented through constraints on stake concentration, dynamic participation requirements, and randomized committee assignment [3]. Collectively, these measures reduce the feasibility of manipulative behavior and promote systemic integrity [53, 52].

Stake caps limit the amount of influence any single entity can exert by restricting the maximum proportion of total stake that can be allocated to a given agent or committee [3]. Stake decay functions could act as a counterbalance to passive accumulation, ensuring that influence must be maintained through continued participation and performance rather than long-term capital lockup alone. The system also enforces diversity across roles, committee memberships, and data sources to minimize the risk of correlated failure or information capture.

Committee shuffling further enhances security by introducing unpredictability into agent selection. Randomized assignment ensures that collusion must be pre-coordinated across uncertain committee boundaries,

significantly raising the cost and complexity of attacks [15]. For example, long-range bribery and Sybil flooding become impractical when agents cannot reliably predict or influence their committee placement, and when diversity constraints prevent homogeneous quorums.

These features emphasize economic deterrence as a first line of defense, reducing dependence on any single authoritative oracle or validator. By embedding accountability, randomness, and role diversity into the core of its coordination logic, the protocol provides a resilient foundation for decentralized intelligence. This approach strengthens the protocol's ability to sustain adversarial pressure while maintaining high standards of trust and performance.

B. Reputation, Oversight, and Governance

As AI agents within the Threshold AI Oracle system assume greater responsibility in interpreting real-world events and triggering on-chain actions, the need for robust oversight mechanisms becomes increasingly important. To ensure the system remains trustworthy, adaptive, and aligned with its decentralization objectives, the protocol incorporates three complementary layers of governance: agent reputation tracking, optional human oversight, and community-controlled protocol governance [11, 40].

Reputation scores serve as a persistent, performance-based evaluation of agent behavior [23]. These scores are used to inform task assignment, role eligibility, and economic incentives, ensuring that agents are rewarded not only for accuracy but also for consistency and timely participation. By assigning reputation weight to each decision, the system introduces a continuous feedback loop that filters out low-performing agents and prioritizes those with strong records of reliability.

In cases involving high ambiguity or ethical subjectivity, the protocol supports optional human-in-the-loop validation [43]. This mode of operation allows human reviewers, whether appointed by decentralized autonomous organizations or elected through token-weighted governance, to review outputs before finalization. Human involvement ensures that socially sensitive or legally complex decisions can be reviewed through an additional layer of accountability.

At the protocol level, decentralized governance mechanisms control parameters such as quorum thresholds, slashing conditions, reward weights, and agent onboarding criteria. Governance may be conducted through on-chain voting, delegated committees, or hybrid models that balance community input with domain-specific expertise. These structures enable the system to evolve transparently and democratically, while preserving its foundational goal of trust-minimized intelligence generation for decentralized applications.

B.1 Agent Reputation Models

The Threshold AI Oracle system employs a composite reputation model to evaluate and rank participating AI agents. Each agent can be assigned a dynamic score based on multiple performance indicators, including historical accuracy, frequency of participation in successful quorums, alignment with final consensus outcomes, response latency, stake duration and volume, and peer or third-party endorsements [23, 51, 3]. The model applies greater weight to recent behavior, thereby emphasizing current reliability over long-past performance.

Agents with consistently strong reputation scores may be prioritized in task allocation and role selection. These agents could be assigned leadership responsibilities such as committee coordination, summarization of deliberations, or evaluation of high-value queries more frequently. For example, an agent with a demonstrated track record in validating financial indicators within decentralized finance contexts may regularly serve in pivotal roles for related tasks.

Conversely, agents with low or declining reputation scores face operational limitations. They may be restricted to lower-priority roles, subjected to more stringent slashing thresholds, or removed entirely from eligibility for committee participation. This ensures that the system continuously promotes agents who exhibit strong epistemic performance while filtering out those who undermine output reliability.

The reputation model supports a meritocratic environment in which accuracy, responsiveness, and consistency are directly linked to opportunities and rewards [48]. By incentivizing high-quality contributions, the system reinforces its broader goal of generating trustworthy, verifiable outputs for decentralized applications.

B.2 Human-in-the-Loop Oversight (Optional)

While the Threshold AI Oracle system is designed to operate autonomously, certain classes of queries may necessitate human oversight. Events that are subjective, ethically sensitive, or politically contentious often require interpretive nuance that is best handled by human judgment [43, 44]. This additional layer of review helps ensure that system outputs remain aligned with cultural norms, legal frameworks, and ethical expectations.

The protocol accommodates several modes of human participation. In manual override mode, designated individuals or decentralized autonomous organizations are permitted to review and amend AI-generated outputs before they are submitted on-chain [40]. In assisted review mode, human validators participate directly in the consensus process alongside AI agents, contributing to quorum formation. In escalation mode, queries that result in disagreement, ambiguity, or failure to reach quorum are flagged for human evaluation by an authorized review entity.

For example, a query evaluating whether a corporation's conduct constitutes unethical behavior may require human interpretation due to jurisdictional variation or cultural sensitivity. Human involvement in such contexts provides critical safeguards for legitimacy and fairness.

These oversight options enable the system to balance efficiency with accountability. While the protocol can respond quickly in routine cases, it remains capable of deferring to human reviewers when a higher level of scrutiny is required. This capacity for selective human input strengthens the protocol's ability to serve in sensitive domains such as governance, dispute resolution, and regulatory compliance. The tradeoff, of course, is latency, this is why Human-in-the-Loop Oversight is expressed as an optional enhancement.

B.3 Protocol Governance

The governance structure of the Threshold AI Oracle system is designed to be decentralized, transparent, and adaptable. It is responsible for managing protocol-level parameters that shape system behavior, including quorum thresholds, agent eligibility criteria, slashing conditions, reward distribution models, and task pricing schedules [40, 46, 22]. Governance may be executed through a Supra-native decentralized autonomous organization composed of token holders, network stakers, and protocol developers [3]. Alternatively, a hybrid governance model may be adopted, combining community-driven voting with oversight from a council of appointed subject-matter experts.

This governance framework provides mechanisms for updating critical aspects of the system such as deliberation algorithms, agent onboarding procedures, and protocol upgrade paths. For example, the community may vote to adjust the minimum staking requirement for agents assigned to financial data queries in order to align system access with associated risk levels.

To mitigate the risk of governance capture, the protocol can incorporate advanced voting systems such as reputation-weighted delegation or quadratic voting schemes [54]. These mechanisms promote wider participation without compromising informed decision-making. By aligning influence with demonstrated knowledge or historical contribution, the governance process remains both inclusive and robust.

Overall, the governance architecture enables the protocol to evolve in response to shifting technological, economic, and regulatory conditions. It maintains a balance between community sovereignty and expert oversight, ensuring that changes to the system reflect both collective input and domain-specific insight.

B.4 Upgrade Paths and Agent Rotation

The Threshold AI Oracle system includes mechanisms for structured upgrades and scheduled agent rotation to maintain resilience, fairness, and adaptability over time. These procedures are essential for preventing

long-term stagnation in agent performance and for enabling the integration of new capabilities as artificial intelligence and blockchain technologies evolve.

Agent upgrades occur through regular epoch-based refreshing cycles. During each cycle, agents are re-evaluated based on metrics such as accuracy, participation frequency, latency, and reputation. Agents that fail to meet performance thresholds may be retired, while new agents may be admitted following successful onboarding procedures. This process ensures that the agent pool reflects current best practices and remains aligned with system requirements.

Transparency is maintained through a version-controlled agent registry that allows the community to audit historical updates and track changes to deployed models [46]. In parallel, synthetic challenge datasets are used to evaluate model generalization and detect issues such as drift, underspecification, or emergent bias [45]. For example, a newly introduced language model tailored to legal contexts may be incorporated into the system following successful performance validation and governance approval [44].

Committee compositions are periodically rotated to prevent long-term dominance by any single agent or group [15]. Randomized assignment and enforced diversity further reduce the risk of collusion or influence centralization. These design features help maintain an equitable and secure operational environment, allowing the protocol to adapt dynamically as system needs and technological capabilities advance.