AccountableAI: Verifiable and Economically Accountable Autonomous AI Agents

Jinyuan Chen
jinyuan@ocior.com

April 4, 2026

Abstract

Autonomous artificial intelligence (AI) agents are increasingly capable of executing real-world actions with financial and operational consequences. However, existing systems lack deterministic verification and enforceable accountability. In this paper, we propose *AccountableAI*, a framework that combines policy commitments, cryptographic verification, and economic incentives to ensure trustworthy agent behavior.

In AccountableAI, AI agents commit to inputs, models, and execution policies prior to task execution, and subsequently provide verifiable evidence of their actions. A set of consensus nodes performs both correctness verification and distributed ledger consensus, determining whether agent outputs satisfy predefined constraints and policies. Verified outcomes are recorded on a blockchain or distributed ledger, while invalid or noncompliant actions trigger automated economic penalties, including slashing of agent-provided collateral and compensation to affected users.

AccountableAI supports multiple verification mechanisms, including deterministic rule evaluation, redundancy checks, and cryptographic proofs such as zero-knowledge proofs. It further enables flexible economic models incorporating per-task fees, subscription fees, staking rewards, and node incentives. By combining policy commitments, verifiable execution, decentralized consensus, and automated settlement, AccountableAI provides an accountability layer for autonomous AI systems, enabling safe deployment of AI agents in financial, enterprise, and consumer applications. With AccountableAI, we can trust AI, not because it is intelligent, but because it is accountable.

Contents

1 Introduction

Autonomous artificial intelligence (AI) agents are rapidly evolving from decision-support systems into *actors with real economic authority* [1]. Modern AI systems can execute financial trades, rebalance portfolios, manage subscriptions, coordinate supply chains, and initiate payments with minimal or no human intervention. While these capabilities unlock significant efficiency gains, they also introduce a fundamental challenge: *the lack of accountability for autonomous AI actions*.

When human professionals such as financial advisors, compliance officers, or operations managers make incorrect or negligent decisions, accountability is enforced through legal contracts, regulatory frameworks, or institutional liability. In contrast, when an AI agent makes an error—whether due to hallucination, flawed reasoning, data drift, or adversarial manipulation—there is no native mechanism to (i) prove what decision logic was followed, (ii) deterministically attribute fault, or (iii) provide automatic economic recourse. This absence of enforceable accountability creates what we term the *Accountability Gap*.

As a result, enterprises, financial institutions, and individual users remain reluctant to delegate high-value or irreversible decisions to autonomous AI systems. The downside risk of failure is often unbounded, while the internal reasoning processes of AI models remain opaque and difficult to audit. Existing approaches attempt to mitigate these risks through two primary strategies.

First, *centralized guardrails and monitoring systems*—such as Application Programming Interface (API) filters, rule-based validators, and prompt constraints—attempt to restrict undesirable behavior before execution. However, these mechanisms are inherently opaque, centrally controlled, and lack cryptographic guarantees of correctness. They do not provide verifiable proof of compliance, nor do they offer economic compensation when failures occur.

Second, *post-incident auditing and human oversight* rely on reviewing logs and system outputs after errors have occurred. While useful for diagnosis, these methods are slow, costly, and reactive. In high-frequency environments such as financial markets or automated operations, economic damage often occurs before any corrective action can be taken.

In summary, existing solutions aim to make AI systems *safer*, but not *accountable*. There is currently no infrastructure layer that enables autonomous AI agents to be held financially liable for their actions in a deterministic and automated manner.

To address this limitation, we propose *AccountableAI*, a framework that introduces verifiable execution and economic accountability into autonomous AI systems. AccountableAI requires agents to commit to their inputs, models, and execution policies prior to acting, and to provide verifiable evidence of correct behavior. A decentralized verification layer evaluates compliance, and incorrect actions trigger automatic economic penalties through collateral slashing and user compensation. By combining cryptographic verification with economic enforcement, Account-

ableAI transforms AI agents into accountable economic actors.

This work makes the following contributions:

- We formalize the *Accountability Gap* in autonomous AI systems and identify its implications for real-world deployment.
- We introduce a novel framework for *verifiable AI execution* based on commitments, proofs, and deterministic verification.
- We propose an *economic enforcement model* using staking and slashing to ensure accountability and provide automatic recourse.
- We demonstrate how the framework enables safe deployment of AI agents across financial, enterprise, and consumer applications.

1.1 Definitions

Unless otherwise specified, the following terms may be used as follows:

- **Intent**: A machine-readable or human-readable task description including one or more goals, constraints, permissions, economic budgets, risk tolerances, or verification requirements.
- **AI Agent**: Any software-based or model-based autonomous or semi-autonomous decision system that may perform one or more actions in response to an intent.
- **Policy**: One or more rules, limits, constraints, optimization criteria, permissions, prohibitions, or verification conditions that govern agent behavior.
- **Consensus Node**: A node that performs both verification of AI execution and distributed ledger consensus [2, 3] or ledger maintenance.
- **Collateral / Stake**: Any locked or escrowed asset, token, or value used to economically secure agent behavior or node participation.
- **Verification**: Any deterministic or rule-based procedure for deciding whether an output or action satisfies a committed policy.
- **Proof / Evidence**: Any cryptographic, computational, or procedural artifact used to support verification, including hashes, signatures, traces, zero-knowledge (ZK) proofs, attestation data, redundancy outputs, or challenge results.
- **Settlement**: Any process by which results, payments, slashing, compensation, or state changes are finalized and recorded.

2 AccountableAI

AccountableAI is a blockchain-based infrastructure protocol that enables *economically accountable autonomous AI agents*. It provides a *Proof-of-Accountability (PoA)* layer that sits between an AI agent and any high-value or irreversible action, such as executing a trade, transferring funds, triggering compliance workflows, or managing booking and subscriptions. Instead of trusting AI systems by reputation or monitoring alone, AccountableAI enforces correct behavior through cryptographic verification and economic liability. With AccountableAI, we can trust AI, not because it is intelligent, but because it is accountable.

2.1 System Architecture

As illustrated in Figure 1, the AccountableAI system comprises an intent layer, an AI agent execution layer, a consensus and verification layer, and a blockchain settlement layer.

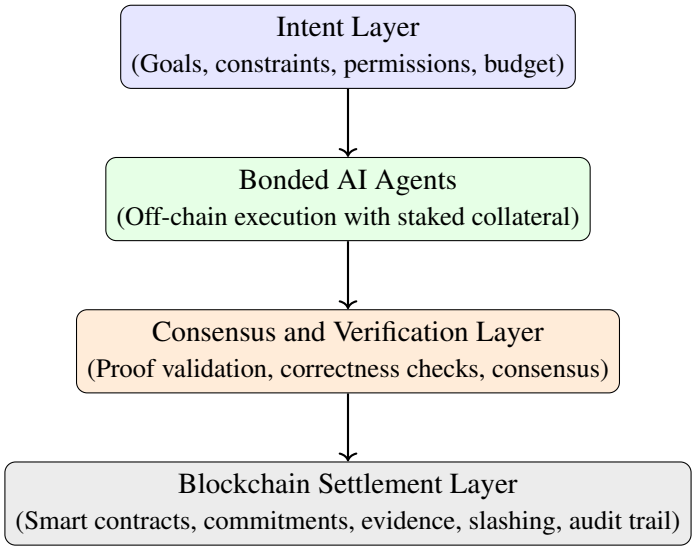

Figure 1: High-level architecture of the AccountableAI system illustrating the flow from user intent to AI agent execution, consensus-based verification, and blockchain settlement.

An end user, application, enterprise system, or machine may submit an intent. The intent may be received directly by an AI agent or may be routed through one or more dispatch, auction, broker, or selection mechanisms. An AI agent may choose to accept the task, post stake, and execute the task according to one or more policies. Outputs, proofs, or evidence are then evaluated by the consensus and verification layer. Settlement results, including approvals, payments, compensation, slashing, and logging, are stored in the blockchain settlement layer.

2.2 Intent Structure

An intent I may be represented as:

$$I = (G, C, P, B, \Gamma),$$

where:

- G denotes one or more goals or requested outcomes;
- C denotes one or more constraints, limits, or permissions;
- P denotes payment or fee parameters;
- B denotes a budget, escrow, or amount authorized for execution;
- Γ denotes one or more verification or assurance requirements.

The intent may include, by way of non-limiting example:

- asset allocation limits for financial automation;
- travel constraints such as maximum price, departure window, refundability, or preferred carriers;
- procurement rules such as approved vendors, maximum order sizes, or sustainability requirements;
- operational constraints such as timing windows, spend limits, geofencing, or security rules.

2.3 AI Agent Structure

An agent A may be characterized by:

$$A = (K, m, \pi, S, R),$$

where:

- K denotes one or more identities, keys, or credentials;
- m denotes one or more models, programs, or algorithm identifiers;
- π denotes one or more policies governing permitted actions;
- S denotes stake or collateral;
- R denotes one or more reputation, performance, or history metrics.

The model identifier m may refer to a fixed model version, program binary, container digest, execution image, model checkpoint hash, or other reproducible reference. The policy π may be encoded as logic, formulas, thresholds, allow-lists, deny-lists, optimization objectives, or combinations thereof.

2.4 Verification Workflow

AI agent execution follows a structured workflow including commitment, execution, proof generation, verification, and settlement.

As shown in Figure 2, this workflow enables deterministic validation of agent behavior.

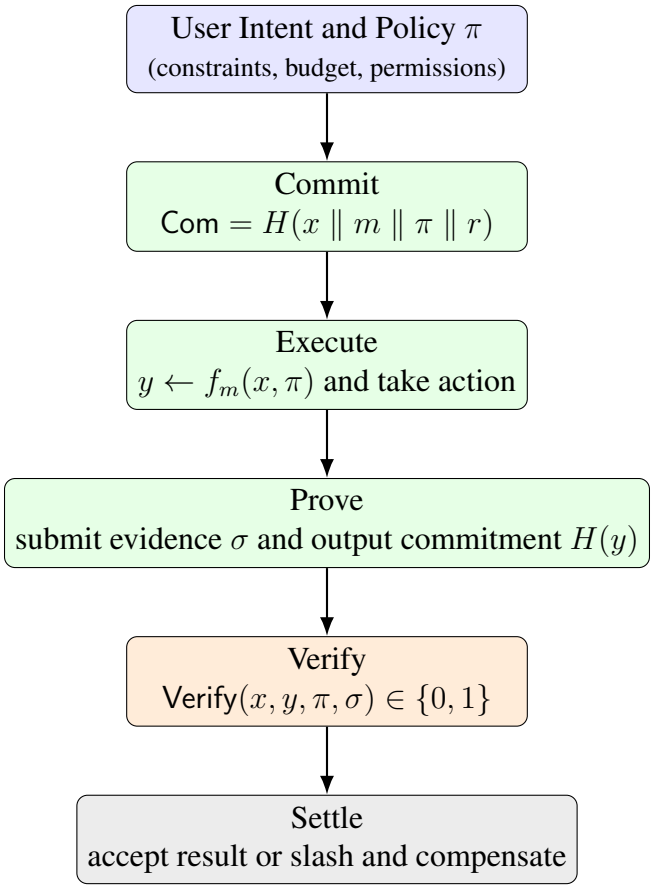

Figure 2: Verification workflow illustrating commitment, execution, proof generation, verification, and settlement of AI agent actions.

2.4.1 Commit

Before acting, the AI agent commits to inputs, a model identifier, and one or more policies:

$$\mathsf{Com} = H(x \parallel m \parallel \pi \parallel r),$$

where $H(\bullet)$ denotes the hash function, x denotes an input snapshot or commitment to inputs, m denotes the model or algorithm identifier, π denotes one or more policies, and r denotes a nonce or randomness value.

The commitment may be stored on the blockchain, a sidechain, a rollup, a state channel, a distributed database, or any suitable verifiable storage.

2.4.2 Execute

The AI agent computes:

$$y \leftarrow f_m(x, \pi),$$

where $f_m(\cdot)$ denotes the algorithm, model, or execution function. The agent may then perform a real-world or digital-world action, including but not limited to transmitting payment instructions, selecting a flight, submitting an order, signing a message, or triggering a downstream workflow.

2.4.3 Prove

After execution, the agent provides one or more evidence objects σ supporting the result y. Evidence may include:

- cryptographic signatures;
- execution traces;
- checkpoints or logs;
- challenge-response data;
- redundancy comparison records;
- zero-knowledge proofs;
- trusted execution environment attestations.

2.4.4 Verify

Consensus nodes evaluate:

$$\mathsf{Verify}(x, y, \pi, \sigma) \in \{0, 1\}.$$

If verification succeeds, settlement proceeds. If verification fails, one or more penalties or dispute processes may be triggered.

2.4.5 Settle

Settlement may include:

- marking a task completed;
- distributing user-paid fees;
- slashing or reducing agent stake;
- compensating a user, enterprise, or counterparty;

- distributing rewards to consensus nodes;
- storing records on-chain.

2.5 Consensus Nodes

In AccountableAI, a consensus node may perform both:

1. **Verification**: validating AI outputs, policies, and evidence; and
2. **Blockchain Consensus**: participating in the maintenance of a blockchain or distributed ledger that stores commitments, smart contracts, evidence, and settlement results.

Consensus nodes may use any suitable consensus protocol, for example, Ocior [4]. Consensus nodes may verify correctness by:

- checking deterministic policy predicates;
- verifying ZK-proofs or validity proofs;
- comparing multiple independent outputs;
- running challenge-response mechanisms;
- validating signatures or attestations.

2.6 Blockchain and Smart Contract Layer

In AccountableAI, the blockchain or distributed ledger stores:

- commitments to inputs, models, and policies;
- smart contracts governing tasks, escrow, permissions, payments, and slashing;
- proofs, evidence, and dispute records;
- task results, settlement decisions, and audit logs.

Smart contracts may be used to:

- lock user budgets or escrow funds;
- lock agent stake or node stake;
- distribute fees among agents and nodes;
- slash stake automatically under specified conditions;
- route compensation to affected users;
- record governance updates or policy templates.

2.7 Deterministic Verification

A key feature of AccountableAI is the ability to define correctness as an objective, deterministic predicate. For a given committed input and policy, any independent verifier should reach the same decision.

For example, a travel booking policy may require:

- price no more than $400;
- departure between 8:00 and 12:00;
- direct flight;
- refundable ticket;
- selection of the cheapest qualifying option.

Verification may determine whether:

$$\mathsf{Verify}(x, y, \pi) = \begin{cases} 1, & \text{if all constraints are satisfied and optimality holds;} \\ 0, & \text{otherwise.} \end{cases}$$

This allows deterministic enforcement. A system may slash the agent if the selected result violates the policy or is non-optimal under the committed data.

3 Economic Flow

In AccountableAI, the system distributes economic value among users, AI agents, and consensus nodes through task-based or subscription-based fees.

As illustrated in Figure 3, fees are allocated across participants based on execution, verification, and settlement roles.

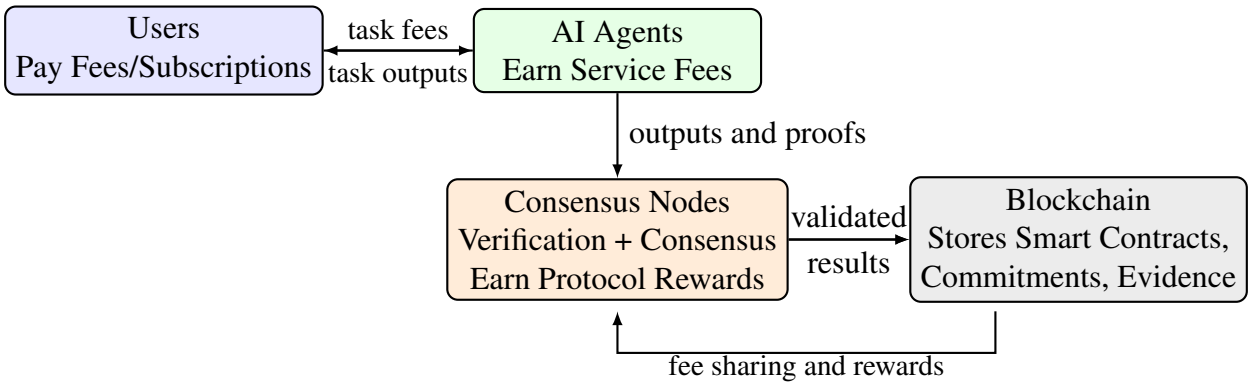

Figure 3: Fee and reward diagram of the AccountableAI system.

3.1 Fees

In AccountableAI, users may pay:

- per-task or per-intent fees;
- subscription fees, including monthly or annual plans;
- priority fees;
- dispute or challenge fees;
- enterprise service fees.

Fees may be denominated in a native token, stablecoin, fiat-referenced unit, internal credit, or other suitable economic unit.

3.2 AI Agent Earnings

AI agents may earn service fees when they successfully execute tasks. Fees may be distributed after verification and settlement. The system may further allocate different percentages of user-paid fees to the executing agent, the protocol treasury, consensus nodes, or other ecosystem participants.

3.3 Consensus Node Rewards

Consensus nodes may earn:

- transaction and settlement fees;
- verification and dispute-resolution fees;
- staking rewards;
- subscription fee allocations.

3.4 Slashing and Compensation

When a verification rule fails, one or more contracts may:

- reduce or slash the agent's collateral;
- compensate a user, enterprise, or counterparty;
- reward a challenger or verifying node;
- record the violation in agent history.

4 Illustrative Use Cases

AccountableAI serves applications unified by a single requirement: autonomous AI agents that act daily with real economic consequences must be verifiable and economically accountable. Some use cases are provided below.

- **Crypto-Native and Decentralized Finance (DeFi) Operations:** Autonomous trading, liquidation, treasury management, and cross-chain portfolio agents managing high-value on-chain assets where execution errors are irreversible.
- **Enterprise FinTech and Financial Operations:** AI agents executing transaction approvals, compliance monitoring, fraud detection, and financial risk management in regulated environments requiring auditability and liability.
- **Supply Chain and Operations:** Autonomous procurement, logistics coordination, and inventory optimization agents managing payments, contracts, and physical goods.
- **Prosumer and Power Users:** Personal finance guardians, subscription and spending managers, trading bots, and licensing agents operating under user-defined constraints.
- **Everyday Consumers:** Autonomous personal assistants that can safely book flights and travel, manage subscriptions, optimize mobility, enforce budgets, and execute payments or contractual actions with provable guarantees and financial recourse.

The following subsections explain some use cases in detail.

4.1 Financial Risk Guardian

A user may authorize an agent to manage a portfolio under constraints such as:

- no more than 20% allocation to high-volatility assets;
- no single trade larger than $5,000;
- rebalance if volatility exceeds a threshold V.

If the resulting portfolio violates the exposure threshold or if an oversized trade is made, deterministic verification can fail and collateral can be slashed.

4.2 Travel Booking Agent

A user may authorize an agent to book a flight under constraints such as:

- price cap;

- direct flight requirement;
- departure window;
- refundability;
- cheapest qualifying option.

If the selected ticket violates one or more constraints, or if a cheaper qualifying option existed within the committed input snapshot, the agent may be penalized and the user compensated.

4.3 Digital Executor

An agent may be configured to transfer assets or release encrypted materials upon satisfaction of conditions, such as prolonged inactivity or a time-based trigger. The condition itself can be committed and verified using timestamps, oracle data, and escrowed asset references. Premature execution or misdirected transfer may trigger automated penalties.

4.4 Enterprise Compliance or Operations

An enterprise may authorize an agent to approve or reject transactions, generate procurement decisions, or coordinate logistics subject to policy constraints, approved vendors, or spend limits. Consensus nodes verify compliance before settlement or approval.

5 Comparison with Existing Systems

The problem of accountability in autonomous AI systems intersects multiple domains, including decentralized AI platforms, verifiable computation, and traditional risk management. Existing approaches address isolated aspects of this problem but fail to provide a unified framework for verifiable execution, economic enforcement, and automated recourse.

Decentralized AI platforms such as Fetch.ai and Autonolas enable autonomous agent coordination and task execution in open environments [5, 6]. Similarly, networks such as SingularityNET focus on decentralized marketplaces for AI services [7], while Bittensor introduces incentive-driven machine learning networks [8]. Infrastructure-focused platforms such as Akash provide decentralized compute resources but do not address correctness or accountability of AI behavior [9].

Parallel efforts in cryptographic verification, including zero-knowledge proofs and verifiable computation systems, enable correctness guarantees for specific computations [10, 11]. However, these systems typically lack integrated economic enforcement or application-layer coordination for autonomous agents.

Traditional institutions such as insurance providers and auditing firms offer forms of risk mitigation and post-hoc accountability. However, these approaches are manual, reactive, and operate outside the execution layer of AI systems, limiting their effectiveness in real-time autonomous environments.

Table 1 summarizes how existing platforms address key dimensions of accountable AI systems, including agent support, execution verification, economic enforcement, and system-level coordination.

Table 1: Comparative analysis of existing platforms across four key dimensions: AI agent support, execution verification, economic accountability, and system-level coordination.

System	**Agents**	**Execution Verification**	**Slashing / Liability**	**Coordination**
Fetch.ai	Yes	No	Limited	Partial
Autonolas	Partial	No	Partial	Partial
SingularityNET	No	No	No	No
Bittensor	Partial	Partial	Weak	No
Akash	No	No	No	No
Ritual.net	Partial	Partial	No	Partial
Modulus Labs	No	Yes	No	No
Insurance (e.g., Lloyd's)	No	No	Yes (manual)	No
Auditors (e.g., Deloitte)	No	No	No	Partial (off-chain)
AccountableAI	Yes	Yes	Yes	Yes

The comparison highlights a fundamental gap: existing systems either provide *infrastructure without accountability* (e.g., compute networks), *verification without enforcement* (e.g., cryptographic proofs), or *enforcement without real-time integration* (e.g., insurance and audits).

AccountableAI addresses this gap by integrating these components into a unified framework. Specifically, it introduces a tightly coupled pipeline in which:

- AI agents must commit to inputs, models, and policies prior to execution;
- outputs are verified against deterministic correctness constraints;
- violations trigger automated economic enforcement through staking and slashing;
- all actions and outcomes are recorded in a shared settlement layer.

5.1 Advantages

The core advantage of AccountableAI is *automated recourse*: enforceable accountability that is programmatic, cryptographic, and economically backed.

Key differentiators include:

- **Execution Verification:** Unlike benchmark-based or reputation-based systems, AccountableAI validates outputs against explicit policy constraints using deterministic verification or cryptographic proofs.
- **Economic Enforcement:** The system introduces native economic consequences for incorrect behavior through collateral staking and automated slashing, aligning incentives with correct execution.
- **End-to-End Integration:** Verification, enforcement, and settlement are integrated into a single protocol, eliminating reliance on off-chain processes or manual intervention.
- **Platform Neutrality:** The framework is model-agnostic and infrastructure-independent, enabling compatibility with diverse AI systems and preventing vendor lock-in.

Together, these properties position AccountableAI as a foundational layer for trusted autonomous AI systems, bridging the gap between capability and accountability.

6 Conclusion

In this paper we propose AccountableAI: an accountability layer for autonomous AI. By combining policy commitments, stake-backed execution, consensus-based verification, blockchain settlement, and automated compensation or slashing, AccountableAI enables AI agents to operate with real economic authority while remaining subject to objective and enforceable constraints. With AccountableAI, we can trust AI, not because it is intelligent, but because it is accountable.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[3] V. Buterin, "A next-generation smart contract and decentralized application platform," *Ethereum White Paper*, 2015.

[4] J. Chen, “Ocior: Ultra-fast asynchronous leaderless consensus with two-round finality, linear overhead, and adaptive security,” Sep. 2025, available on arXiv: https://arxiv.org/abs/2509.01118.

[5] Fetch.ai Foundation, “Fetch.ai documentation,” https://fetch.ai, 2023.

[6] Autonolas, “Autonolas protocol,” https://olas.network, 2023.

[7] SingularityNET, “Singularitynet whitepaper,” https://singularitynet.io, 2017.

[8] Bittensor, “Bittensor whitepaper,” https://bittensor.com, 2021.

[9] Akash Network, “Akash whitepaper,” https://akash.network, 2020.

[10] B. Parno, J. Howell, C. Gentry, and M. Raykova, “Pinocchio: Nearly practical verifiable computation,” in *IEEE Symposium on Security and Privacy*, 2013.

[11] J. Groth, “On the size of pairing-based non-interactive arguments,” in *EUROCRYPT*, 2016.