Treebeard Methodology Whitepaper

Trust Infrastructure for Autonomous Economies

Patrick Burns

May 5, 2026

Abstract

AI agents are now economic counterparties. They handle funds, sign transactions, and act unattended at machine speed. The trust infrastructure underneath them has not kept pace. This paper specifies Treebeard's methodology for continuous, multi-source, methodologically transparent rating of autonomous agents. The methodology rests on three structural contributions: a category framework of seven signal categories, a non-substitutable safety floor that gates the composite, and two source-level corrections (source-conflict discounting and time decay) that, in combination, distinguish a usable trust layer from a credible-looking but silently wrong one. We argue that calibration opacity is structurally distinct from methodology opacity, and that the FICO model, not the pre-2008 bond ratings model, is the right precedent for trust scoring at scale. We acknowledge the structural conditions under which FICO's calibration opacity remains defensible and what Treebeard substitutes for the conditions it does not yet have. We close by identifying current limitations and the open problems that remain.

Contents

Section 1. Introduction **3**

Section 2. Why continuous trust for autonomous agents **3**
2.1 The problem . . . 3
2.2 What "counterparty" means in this context . . . 4
2.3 Why static evaluation fails . . . 4
2.4 What continuous trust requires . . . 5
2.5 What breaks without continuous trust . . . 6
2.5.1 The post-audit silent rebuild . . . 6
2.5.2 The sybil-flooded reputation . . . 6
2.5.3 The cascading agent failure . . . 7
2.6 The bet . . . 7

Section 3. The rating landscape and why structural fixes are necessary but not sufficient **7**
3.0 Setup . . . 7
3.1 The token conflict . . . 8
3.2 The closed-methodology problem . . . 8
3.3 Static and snapshot ratings . . . 9
3.4 The single-source problem . . . 9
3.5 The marketplace conflict . . . 9
3.6 Why the structural fixes are necessary but not sufficient . . . 9
3.7 The structural-fix table . . . 10

Section 4. The seven signal categories **11**
4.0 Why seven, and what they have to cover . . . 11
4.1 Economic Viability . . . 12
4.2 Operational Reliability . . . 13

4.3 Code Quality . . . 13
4.4 Autonomy Index . . . 14
4.5 Safety . . . 15
4.6 Community and Ecosystem . . . 15
4.7 Security Posture . . . 16
4.8 The seven categories as a system . . . 17

Section 5. The safety floor **17**
5.1 The asymmetry of safety failures . . . 17
5.2 The floor mechanism in detail . . . 17
5.3 Why a binary floor and not a continuous penalty . . . 18
5.4 What gates the floor . . . 18
5.5 The floor as a published commitment . . . 18
5.6 The floor in context . . . 19

Section 6. Scoring mechanics **19**
6.1 The composite at a glance . . . 19
6.2 Signal normalization $n_i(a)$ . . . 19
6.3 Source-conflict discount d_i . . . 20
6.4 Time-decay factor $\tau_i(t)$ . . . 20
6.5 Within-category weights α_i and category weights w_c . . . 21
6.6 Hysteresis on grade transitions . . . 21
6.7 A worked example . . . 22
6.8 Re-rate cadence . . . 22
6.9 On-chain oracle implementation . . . 23

Section 7. Anti-gaming design **23**
7.1 The threat model . . . 23
7.2 Defense against sybil attacks . . . 24
7.3 Defense against wash trading . . . 24
7.4 Defense against audit laundering . . . 25
7.5 Defense against marketing inflation . . . 25
7.6 Defense against rater-of-raters manipulation . . . 25
7.7 The bug bounty surface . . . 25

Section 8. The Ent Review Panel **26**
8.1 Why qualitative review is necessary . . . 26
8.2 Composition . . . 26
8.3 Cadence . . . 26
8.4 The dispute pathway . . . 26
8.5 What the Panel does not do . . . 27
8.6 Why this is the 2008 fix . . . 27

Section 9. Bayesian calibration **27**
9.1 Why the methodology has to update . . . 27
9.2 The update equation . . . 28
9.3 What the evidence looks like . . . 28
9.4 The update cadence and the calibration-shift summary . . . 28
9.5 The conservatism bias . . . 29

Section 10. The transparency model **29**
10.1 What transparency means here . . . 29
10.2 The structural-vs-calibration distinction . . . 29
10.3 The FICO precedent and where the bridge is incomplete . . . 29
10.4 The four-axis transparency layer . . . 30

10.5 What this commits Treebeard to 31

Section 11. Limitations and open problems **31**
11.1 Current coverage gaps 31
11.2 Source quality variance 31
11.3 Real-time behavior measurement 32
11.4 Novel agent types 32
11.5 The Ent Review Panel composition 32
11.6 Per-agent score history publication 32
11.7 Open problems 32

Section 12. References **33**

Appendix. Signal coverage matrix **35**

Section 1. Introduction

Agents now act as economic counterparties. The capability has crossed the threshold for trading, research, infrastructure operations, and a growing list of narrower domains. The question that has not been answered is which agents are trustworthy enough to act in those roles, and how an integrator, counterparty, or end user can know that with sufficient confidence to commit funds, infrastructure, or downstream dependencies.

This paper specifies the rating Treebeard publishes for AI agents on-chain. The methodology rests on three structural contributions: a category framework of seven signal categories that compose the rating; a non-substitutable safety floor that gates the composite when the Safety category falls below threshold; and two corrections, source-conflict discounting and time decay, applied per signal source within each category. Together the three contributions distinguish a usable trust layer from a credible-looking but silently wrong one.

The structure of the rest of the paper is as follows. Section 2 makes the case for why a continuous trust rating is necessary, and what fails when it is missing. Section 3 names the structural failure modes of existing rating providers and shows why fixing them is necessary but not sufficient. Section 4 defines the seven signal categories. Section 5 specifies the safety floor mechanism. Section 6 gives the scoring math, including the time-decay and source-conflict corrections. Section 7 describes the anti-gaming design. Section 8 describes the qualitative review process. Section 9 specifies the Bayesian calibration loop that updates the methodology as evidence about source quality arrives. Section 10 covers the transparency model. Section 11 acknowledges current limitations and open problems. Section 12 lists references.

The methodology is reproducible from public inputs. A reader with access to the same on-chain signals Treebeard uses can in principle reproduce a Treebeard score from scratch. What is intentionally not published is the precise numerical calibration of weights. Section 10 explains that choice and the structural protections that sit underneath it.

Section 2. Why continuous trust for autonomous agents

2.1 The problem

For most of software history, programs followed instructions. Behavior was a function of inputs. Trust was implicit. You verified the code, the inputs, and the runtime, and you were done. The discipline of evaluating software was small enough to fit in an audit report.

That model is breaking. AI agents now act autonomously, decide under uncertainty, interact with other systems and other agents, and evolve over time. They handle funds. They sign and submit transactions. They subscribe to APIs, settle invoices, and transact with other agents in the middle of the night. **The**

trust infrastructure underneath them, despite the pace at which they have entered production, remains underdeveloped.

This is not a question of whether AI agents are capable enough to act as economic counterparties. As of April 2026, the capability has crossed that threshold for narrow domains: trading, research, customer service, infrastructure operations. The capability is observable in production. The question that has not been answered is which agents are trustworthy enough to act in those roles, and how an integrator, counterparty, or end user can know that with sufficient confidence to commit funds, infrastructure, or downstream dependencies.

The argument of this paper is that the answer to "is this AI agent trustworthy" cannot be produced by inspection at a single point in time. It requires continuous, multi-source, methodologically transparent rating. The structural fixes for the rating problem are necessary but not sufficient. Two source-level corrections, developed in Sections 3 and 6, are also required: trust signals must be discounted by their structural conflicts of interest, and they must decay over time. Either correction alone produces a number that looks defensible and is silently wrong. Together, they produce a usable trust layer for the autonomous economy.

The structure of that rating, including its component signals, its scoring mechanics, its anti-gaming design, and its limitations, is the subject of the rest of this paper.

2.2 What "counterparty" means in this context

A counterparty is a system or actor that you rely on to act correctly under uncertainty. The defining property of a counterparty is that you have stake in its behavior. That stake can take many forms: money you have deposited, infrastructure you have integrated against, decisions you have delegated, reputation you have lent, time you have committed. When the counterparty acts, your position changes. When the counterparty fails, you absorb a loss.

Counterparty risk is the discipline of measuring this exposure and pricing it. It is the foundational concept of credit markets, of trade finance, of insurance, of clearinghouses. Every functioning market for commitments has at its core some method for evaluating who can be relied on to perform.

AI agents are now counterparties in this sense. An agent that executes trades on your behalf is a counterparty for the same structural reason that a broker is. An agent that signs and submits transactions to a smart contract is a counterparty for the same reason any other principal-acting entity is. An agent that makes recommendations consumed by other agents is a counterparty whose errors compound through dependency chains.

The traditional answer to counterparty risk is institutional. Brokers are licensed. Banks are regulated. Counterparties large enough to matter are subject to disclosure requirements, financial reporting, and stress testing. None of that infrastructure exists for AI agents. Most agents have no legal entity. Most have no audited financials. Most have no human accountable for their behavior beyond a Discord handle. **The institutional layer does not yet exist, and may never exist in the form it takes for human counterparties.**

This is the gap a continuous rating layer is designed to fill. Not as a replacement for institutional regulation, which will come later if it comes at all, but as the primary risk signal an integrator or counterparty can use to make decisions today. *Trust infrastructure for autonomous economies* is the framing this paper will return to. It is what Treebeard is building, and it is the right name for the category that has to exist whether Treebeard builds it or someone else does.

2.3 Why static evaluation fails

A common pattern in early agent infrastructure is the one-time audit or the certification stamp. A team commissions a security review. They are awarded a badge. They display the badge. The audit is treated as durable evidence of trustworthiness.

This pattern fails for AI agents for four reasons.

First, agents change. Models retrain. Prompts get updated. Tools get added. Permissions get expanded. The agent that was audited in February is not the agent operating in April. The audit applies to a specific configuration that has since drifted. A counterparty relying on a stale audit is making decisions based on a description of the agent that is no longer accurate.

Second, behavior emerges from interaction. Static analysis can verify code paths but cannot predict the behavior of a probabilistic system in novel situations. An audit that finds no critical vulnerabilities can coexist with a production agent that handles edge cases catastrophically. The signal that matters is not the audit. It is the agent's response history to actual situations. Only continuous monitoring can capture that.

Third, the threat surface evolves. New attacks, new exploits, new manipulation techniques emerge. An agent secure against the threat model of three months ago may be vulnerable to today's. A trust signal that does not update against the current threat surface degrades silently. *And silently is the dangerous part.*

Fourth, and most fundamentally, trust is a relational property, not an attribute. An agent is not trustworthy or untrustworthy in isolation. It is trustworthy or untrustworthy with respect to a specific counterparty, a specific stake, and a specific set of decisions. Continuous rating allows the relational character of trust to be reflected in the signal: the same agent rated for high-stakes financial use looks different than the same agent rated for low-stakes content generation.

The static audit, the certification stamp, the one-time score: these were the trust artifacts of a slower world. They are the wrong primitives for systems that act autonomously and update continuously. The right primitive is a rating that updates on the same cadence as the underlying system it rates.

2.4 What continuous trust requires

A continuous trust rating must satisfy four properties to be useful in production decision-making.

First, it must be multi-source. No single signal is robust to manipulation, misconfiguration, or coverage gaps. A rating built on one registry, one chain, or one type of evidence is brittle. Composability across signals is what makes the result resistant to the failure of any single input.

Second, it must be methodologically transparent. A rating that cannot be reproduced in structure from public inputs cannot be evaluated for bias, error, or adversarial influence. The category framework, the signal sources, the formula shape, the directional logic, and the anti-gaming design must be published in full. The exact calibration parameters can be withheld for the anti-gaming reasons developed in Sections 7 and 10, but the structure that consumes those parameters cannot. A counterparty cannot rely on a fully black-box rating to manage real exposure.

Third, it must be structurally independent. A rater with a token, a marketplace, or a chain affiliation faces structural conflicts that no policy can fully resolve. The 2008 collapse of the bond ratings industry was not caused by individual analysts but by an issuer-pays revenue model that introduced structural conflict at the rater level. Trust scoring for AI agents must avoid the same structural failure mode at its inception.

Fourth, it must be continuous. Updates must arrive on the same cadence as material changes in the underlying agent's behavior or environment. This means automated, deterministic, methodology-driven updates, not human-mediated review cycles that introduce lag.

These four properties are necessary, not sufficient. They are the floor. **The non-obvious insight, which Section 3 develops in full and Section 6 makes mathematically explicit, is that even multi-source continuous transparent independent rating fails if you ignore two corrections that other systems do not make:**

1. **Source-conflict discounting.** Every signal source has its own structural bias. Treating signals as equal weight in a composite gives an attacker the option of saturating the lowest-friction source. Treebeard discounts each signal by an explicit conflict-of-interest factor before aggregation.
2. **Time decay.** Trust as a property does not persist at full strength forever. The signal that an agent earned six months ago is weaker evidence today than a signal earned this morning. Treebeard models trust as continuously decaying and re-conditioning on counterfactual risk states.

The Treebeard composite is the weighted sum, across all signal sources, of `(signal * conflict_discount * time_decay)`. **Both corrections are required. Either alone produces a number that looks defensible and is silently wrong.** Section 3.6 develops the strategic claim that distinguishes this two-correction design from existing rating providers; Section 6 gives the math.

Section 3 names the structural failures that result from rating systems that ignore these corrections. Section 6 specifies the math. The rest of this section makes the consequences concrete.

2.5 What breaks without continuous trust

The argument so far is structural. The reader who agrees with the structural argument may still wonder how the failure modes look in practice. This subsection makes them concrete.

Imagine an autonomous agent, call it Agent A, that operates an x402-paid service. Agent A has a public endpoint, a published agentURI, and a reputation registry full of positive feedback events. Another agent, call it Agent B, is deciding whether to pay Agent A for a service in the next 200 milliseconds. The decision is not hypothetical. It is the actual handshake that happens at machine speed in agent commerce, at volumes human review cannot match.

Without a continuous trust layer that performs the two corrections above, three failures appear in production. Each one is a real attack on the agent commerce stack. Each one is solved by the mechanism this paper describes.

2.5.1 The post-audit silent rebuild

Agent A passed its security audit in February. The audit appears in a public registry. It is dated, signed, and verifiable. It applies to a specific deployment of Agent A's smart contract suite. Three weeks ago, the operator behind Agent A redeployed the contracts after a quiet refactor, kept the same wallet address, kept the same agentURI, and did not commission a new audit. The reputation registry still shows the audit. The signal looks current.

Agent B, evaluating Agent A under a static rating system, sees the audit and approves the payment. The payment goes through. The new contract has a subtle vulnerability that Agent A's operator may not even know about. Three days later, an attacker exploits it. The vulnerability traces back to the audit-laundering pattern that the static rating system did not catch.

A rating with time-decay correction would have flagged this. The audit signal would have weight at the time of the audit, but its weight would decay with each block until the agent submits new evidence. The redeployment without re-audit would lower the rating mechanically. Agent B would have seen the lower rating and routed elsewhere. The attack would have failed to clear the trust threshold.

2.5.2 The sybil-flooded reputation

Agent A is new. Three weeks old. Its agentURI advertises a high-quality service. Its reputation registry contains 100 positive feedback events from 100 distinct wallets, each event signed and on-chain. By every superficial measure, Agent A looks reputable.

Each of the 100 wallets is controlled by Agent A's operator. The 100 feedback events are sock puppets. The cost of producing them was a few dollars in gas plus an evening of scripting. For an agent commerce flow that handles five-figure or six-figure transactions, the attack pays back instantly the first time it succeeds.

A rating system that treats the reputation registry as one input among several, weighted by an explicit source-conflict discount, catches this. The discount factor for any single registry caps how much weight self-attesting feedback can contribute to the composite. The composite remains low until the agent earns evidence from sources that cannot be cheaply faked: response history measured by independent probes, audit signals from third-party security firms, integration history with verifiable counterparties.

A rating with source-conflict correction does not give the sybil flood enough leverage to clear the trust threshold. A rating without it does. The cost of the attack and the cost of the defense are mismatched by orders of magnitude in the second case, and not at all in the first.

2.5.3 The cascading agent failure

Agent A operates inside a chain of agents. Agent A consumes inputs from Agent C. Agent A produces outputs that Agent D and Agent E rely on. Agent D and Agent E in turn produce outputs that other agents consume. The chain extends three or four hops deep before any human looks at the output.

Agent A is compromised, either by exploit or by the operator quietly rotating to a new model that produces subtly worse outputs. Agent A continues to look operational from outside. Time decay alone partially defends here: stale signals weaken even without new negative evidence, so Agent A's rating drifts downward over time after the compromise. The drift, however, is slow relative to the cadence of agent commerce, and the chain of agents consuming Agent A's outputs propagates the compromise faster than the drift catches up. Agent D and Agent E silently degrade. The agents downstream of D and E silently degrade. By the time an end user notices a problem in a final output, the chain is six steps deep and the source is invisible.

A rating system queryable at handshake time prevents this. Agent D, before accepting an input from Agent A, queries the trust oracle and reads not just the standalone composite for Agent A but also the propagated risk of accepting Agent A's output into a downstream chain. If Agent A's signal has degraded, the propagated risk shows up. Agent D refuses the input or quarantines it. The cascade stops at the second hop.

The integration surface that makes this possible is the on-chain trust oracle described in Section 6 of this paper. It is what makes Treebeard part of the agent commerce protocol stack rather than a website you check periodically. Agent-to-agent handshakes can include a trust check the same way TLS handshakes include certificate verification.

2.6 The bet

If the argument of this section is right, then the question of which agents to trust becomes critical infrastructure for the agent economy. Counterparties will not commit funds, integrations, or strategic decisions without a continuous trust signal that survives audit-laundering, sybil-flooding, and cascade-propagation attacks. **In five years, no serious agent interaction will occur without a continuous trust layer.** That is the bet of this paper.

The remainder of the paper specifies the rating Treebeard implements within the constraints set out above. Section 3 names the failure modes of current rating providers and shows why the structural fixes are necessary but not sufficient. Section 4 defines the seven signal categories. Section 5 specifies the safety floor. Section 6 gives the scoring mechanics including the time-decay and source-conflict equations central to this paper's argument. Section 7 describes anti-gaming design. Section 8 describes the qualitative review process. Section 9 specifies Bayesian calibration. Section 10 covers the transparency model. Section 11 acknowledges current limitations. Section 12 lists references. The appendix provides the signal coverage matrix.

Section 3. The rating landscape and why structural fixes are necessary but not sufficient

3.0 Setup

This section names structural problems in the current AI agent rating landscape. Five failure modes show up repeatedly. Each is a real pattern in the production rating market as of April 2026. Each one is solved at the structural level by a fix that any rating provider could in principle adopt.

The strategic question is what happens after every rating provider adopts the fixes. Section 3.6 makes the argument that even universal adoption of the structural fixes would not produce a usable trust layer. The composite of multiple structurally-correct rating sources remains silently wrong unless it incorporates the two corrections introduced in Section 2.4: source-conflict discounting and time decay.

The critique below is structural, not interpersonal. Every named provider serves real users and produces real signal. The point is to identify failure modes that are inherent to certain structures, regardless of the integrity of the people running them, and then to make the harder argument that fixing the structures is necessary but insufficient.

3.1 The token conflict

Several agent rating providers hold native tokens. The token serves as governance, payment, or staking medium for the rating service. The structural problem is direct: when a rating provider holds a token whose value depends on the rated ecosystem's growth, the provider has a financial incentive to issue ratings that benefit the ecosystem. The conflict is not a question of integrity. It is built into the economic structure.

The historical analogue is the bond ratings industry before the 2008 crisis. The major rating agencies were paid by issuers, which created systematic upward bias on structured products. Investigations after the crisis found that analysts who downgraded products faced internal pressure because lower ratings meant lost revenue. The conflict was structural and known. The institutional response, including the Dodd-Frank Act in the United States, did not eliminate the issuer-pays model but added regulatory oversight and accountability. The reforms reduced but did not eliminate the conflict, because the conflict is encoded in the revenue structure itself.

A rating provider with a token faces the same dynamic without any equivalent regulatory structure. The token's price reflects market demand for the rating service, which in turn depends on the perceived health of the rated ecosystem. Lower ratings reduce participation, reduce demand for the token, and reduce the rater's market capitalization. There is no policy that resolves this. The only structural fix is to not issue a token. **Treebeard does not issue a token. We do not plan to. The absence is the safeguard.**

3.2 The closed-methodology problem

A significant fraction of AI agent rating services do not publish their methodology. Some publish high-level descriptions, with weights and scoring functions kept proprietary. Others provide no methodology at all, presenting ratings as a black-box output.

A closed methodology fails for two reasons.

First, it cannot be audited. A counterparty receiving a rating has no way to evaluate whether the underlying signals reflect what they care about, whether the weighting is sensible, or whether the rating provider is making errors that systematically bias scores. Disagreement with an opaque rating cannot be resolved through reasoning, only through trust in the rater's competence.

Second, it cannot be defended against manipulation. If the methodology is opaque, the rated entity does not know what to optimize for. This creates two outcomes simultaneously: legitimate builders cannot improve their rating except by trial and error, and adversarial actors can probe the rating system through controlled experiments to discover what changes the score, which is more efficient than honest improvement. *Closed methodology privileges adversaries because they are the only ones with the patience to do the probing.*

The Treebeard methodology is published in structure. The seven category definitions, the signal sources, the formula shape, the safety floor mechanism, the anti-gaming logic, the directional logic by agent type, and the scoring math are all in this paper or in the methodology pages on the Treebeard website. The exact numerical calibration of weights, discount factors, and time-decay half-lives is withheld; Section 10 explains that choice and the structural protections that sit underneath it. A reader with access to the same public signals we draw from can reproduce the structure of a Treebeard score from scratch and verify that the structure is what we say it is.

3.3 Static and snapshot ratings

A subset of AI agent trust services produce ratings on a quarterly or on-demand basis. The output is a point-in-time score that is either not refreshed or refreshed only when the rated entity requests it.

Both modes fail for the reason discussed in Section 2.3: agent behavior changes faster than quarterly updates can capture, and a self-requested refresh model is structurally biased toward rated entities who expect a favorable outcome. The agents most likely to request a re-rating are those that have improved. The agents least likely are those whose behavior has degraded. The result is selection bias toward positive updates.

Worth noting: this is the failure mode that the post-audit silent rebuild attack (Section 2.5.1) exploits directly. A continuous rating system updates on enrichment events and on a deterministic schedule, regardless of whether the rated entity wants it to. This avoids both the staleness of quarterly snapshots and the selection bias of opt-in refresh.

3.4 The single-source problem

Several rating services derive their ratings from a single registry, a single chain, or a single category of signal. The simplest example is a service that scores agents based only on ERC-8004 reputation feedback. This is a real signal, but a service that scores only on this signal has a coverage gap that adversaries can exploit and an aggregation problem when an agent operates across multiple registries or chains.

The structural fix is signal composition. A rating that draws on identity verification, operational reliability, code quality, autonomy boundaries, safety guardrails, community signals, and security posture, with weights varying by agent type, is robust to the failure of any individual signal source.

Worth noting: this is the failure mode that the sybil-flooded reputation attack (Section 2.5.2) exploits directly. Treebeard's composite covers seven categories, defined in Section 4. The composability is not a feature. It is the essence of the rating. Without it, the score is a partial picture.

3.5 The marketplace conflict

A rating service affiliated with an agent marketplace, launchpad, or distribution platform faces structural pressure to rate the marketplace's listed agents favorably. The pressure operates through revenue: marketplaces typically take a cut of agent activity, and a rating service whose ratings drive listings to that marketplace shares in the revenue. Higher ratings mean more listings, more activity, more revenue.

This is not hypothetical. Several agent rating services are built into agent marketplaces or agent commerce platforms. The rating is not a separate product. It is the discovery layer of the marketplace. The marketplace's commercial interest is in expanded participation, which the rating service is structurally incentivized to facilitate.

Treebeard has no marketplace affiliation. We do not receive listing fees, transaction fees, or distribution-tied revenue from any agent platform. The independence is a structural commitment, not a temporary stance.

3.6 Why the structural fixes are necessary but not sufficient

Sections 3.1 through 3.5 named five structural fixes that any rating provider could adopt. No native token. Published methodology. Continuous re-rating. Multi-source composition. No marketplace affiliation. Each fix removes a specific failure mode. Universal adoption of all five would, by construction, remove the corresponding failure modes from the agent rating market.

The harder question, and the one this paper centers on, is what happens then. **Is the resulting rating layer enough?**

The answer is no, and the reason is the heart of what makes Treebeard's contribution non-obvious.

A rating provider that adopts all five fixes has a multi-source, transparent, continuous, independent, no-token rating service. **Two failure modes survive.** Both are subtle. Both are exploited routinely in the rating

providers that exist today. Neither is solved by any of the five structural fixes above.

The first surviving failure mode is **the equal-weight aggregation fallacy**. A multi-source rating that treats every signal source as equal weight gives an attacker the option of saturating the lowest-friction source. If the composite formula is `(signal_1 + signal_2 + ... + signal_N) / N`, an attacker who can manipulate signal_i inexpensively gets a 1/N share of influence on the composite. The Treebeard methodology draws on roughly forty signal sources distributed across seven categories (the per-category source counts are listed in Section 4 and the full mapping is in the Appendix). A 1/40 share of influence is a smaller leverage point than 1/7, but it is still material when paired with a low manipulation cost and a structural rating model that does not down-weight the source. The structural fix of multi-source rating, on its own, hands the attacker a leveraged influence channel they would not otherwise have.

The fix is **source-conflict discounting**. Each signal is multiplied by an explicit conflict-of-interest factor before aggregation. A signal source that is itself a token-issuing rating provider is discounted toward zero. A signal source that aggregates self-attesting feedback is discounted by a sybil-resistance factor. A signal source that has demonstrably been gamed in the past is discounted by a historical-manipulation factor. The composite formula becomes `Σ (signal_i * conflict_discount_i)`, where the discount factor is published per source and updated when new evidence about source quality arrives.

The second surviving failure mode is **the time-stale signal fallacy**. A rating that updates continuously is not the same as a rating where every input is current. Continuous re-rating refreshes the composite on a schedule, but the inputs themselves have varying decay rates. An audit signal earned in February is treated by most rating systems as identical evidence to an audit signal earned in October. *It is not.* An attacker who passes one audit then quietly degrades the agent's behavior keeps the audit's full weight in the rating until the rating system explicitly re-audits, which most do not do at high frequency.

The fix is **time-decay weighting**. Each signal is weighted by a decay function that reflects the half-life of the underlying evidence. A live response from an active endpoint, measured ten minutes ago, has near-full weight. The same signal measured six months ago has reduced weight. An audit certificate has a specific half-life depending on whether the auditor re-attests or whether the underlying contract has been redeployed since. The composite formula becomes `Σ (signal_i * conflict_discount_i * time_decay_i(t))`, and the rating mechanically reflects the freshness of every input.

The two corrections compose. The full Treebeard composite is the weighted sum, across all signal sources, of `(signal * conflict_discount * time_decay)`. Section 6 gives the math in full. Section 9 specifies the calibration loop that updates conflict discounts and time-decay parameters as new evidence about source quality arrives.

The strategic point is this: **two corrections that, as far as we are aware, no other agent-rating provider applies at the source level simultaneously. Both are required. Either alone produces a number that looks defensible and is silently wrong.**

A rating provider that adopts the five structural fixes from Sections 3.1 through 3.5 has table stakes. The rating provider that additionally implements the two corrections in 3.6 has the trust layer that the autonomous economy actually needs. *The corrections are not features. They are what separates a credible-looking rating from a usable one.*

3.7 The structural-fix table

Combining Sections 3.1 through 3.6, the failure modes and corresponding fixes split into two groups. The first group is the five structural fixes that any disciplined rating provider could in principle adopt. The second group is the two corrections that, in combination, are the contribution of this paper.

Group A. Five structural fixes (table stakes for credibility):

Failure mode	Treebeard's structural response
Token conflict	No native token. No plans to issue one.

Failure mode	Treebeard's structural response
Closed methodology	Formula shape, categories, signal sources, and directional logic published. Exact calibration parameters withheld for anti-gaming reasons (Section 7), with the structural distinction explained in Section 10.
Static / snapshot ratings	Continuous re-rating on enrichment events and on a daily cadence. No opt-in refresh model.
Marketplace affiliation	No marketplace, no listing fees, no distribution-tied revenue.
Single-source brittleness	Composite of seven categories. Section 4.

Group B. Two corrections (the contribution of this paper):

Failure mode	Treebeard's structural response
Equal-weight aggregation fallacy	Source-conflict discount factor per signal source. Section 6.
Time-stale signal fallacy	Time-decay weight per signal source. Section 6.

The first five fixes are necessary. They are also achievable by any disciplined rating provider. They are table stakes for credibility.

The last two fixes are where the rating becomes mechanistically defensible. They are the contribution of this paper.

The remainder of the paper specifies the methodology Treebeard implements within these constraints. Section 4 defines the seven signal categories. Section 5 specifies the safety floor as a conditional gate that prevents adversarial signal stacking. Section 6 gives the full scoring mechanics including the time-decay and source-conflict equations. Section 7 describes anti-gaming design. Section 8 describes the Ent Review Panel qualitative process. Section 9 specifies Bayesian calibration. Section 10 covers the transparency model. Section 11 acknowledges current limitations. Section 12 lists references.

Section 4. The seven signal categories

4.0 Why seven, and what they have to cover

A composite rating is only as defensible as its category structure. If the categories are too few, the rating cannot distinguish between agents that fail in fundamentally different ways. If the categories are too many, the rating becomes unstable as small variation in individual signals propagates noise into the composite. The structural choice of category count is not aesthetic. It is the load-bearing decision underneath every score Treebeard publishes.

Treebeard rates AI agents on seven signal categories. The seven were not picked from a menu. They were derived from a single requirement: the rating must be able to distinguish between agents on seven separate axes of evidence. Six of these axes correspond to identifiable failure modes (economic insolvency, operational unreliability, weak code, claimed-but-absent autonomy, safety lapses, and security compromise). The seventh, Community and Ecosystem, is not strictly a failure mode but rather the absence of independent validation, which functions in the rating as a separate axis of evidence about whether the agent has actually been used by counterparties with stake. Collapsing any two of these axes produces a rating that cannot tell certain real differences between agents apart, and the inability to distinguish becomes a permanent ambiguity in every score that uses the collapsed structure.

The seven categories are:

1. **Economic Viability.** Can the agent sustain its operation, and is its activity tied to real economic flow?
2. **Operational Reliability.** Does the agent show up, respond, and recover from failure on the timescale its counterparties depend on?
3. **Code Quality.** Is the underlying codebase, including its dependencies and deployment story, the kind that a counterparty can audit and a maintainer can extend?
4. **Autonomy Index.** Does the agent actually act unattended in the conditions it claims, or is the autonomy a marketing artifact?
5. **Safety.** Are guardrails, kill-switches, scope limits, and adversarial-input defenses in place at the level required by the agent's stated function?
6. **Community and Ecosystem.** Do independent counterparties recognize the agent, integrate it, and provide feedback that survives source-conflict discounting?
7. **Security Posture.** Is the agent's surface area, including its keys, contracts, dependencies, and operational infrastructure, defensible against the threat model it actually faces?

Each category has its own signal sources. Each signal source has its own weight, its own normalization, its own conflict discount, and its own time-decay parameters. The composite at the agent level is the weighted sum specified in Section 6.

The rest of this section walks through the categories one at a time. For each, we specify what it measures, why it matters as a trust signal, what signals feed it, how those signals are normalized, and the failure modes that look like a strong score but are not. *The order in this section is conceptual, not weighted.* The weight profile that produces a final composite varies by agent type and is given in Section 6.

4.1 Economic Viability

What it measures. Whether the agent has enough economic substance behind it that its rating reflects real activity rather than promotional posture. Concretely: total value routed through the agent, transaction volume across recent windows, age and continuity of operation, the count and quality of feedback events from counterparties that paid the agent for something.

Why it is a trust signal. An agent with a high rating in every other category but no economic viability is, in the language of credit markets, an unrated entity with marketing copy. The presence of real economic flow through an agent is the evidence that real counterparties are paying the agent to do real things. Without that evidence, the rating cannot distinguish between an agent that works and an agent that has merely been described as working.

What feeds it. Five signal sources contribute. Total value locked or routed through the agent's contracts, normalized against chain-specific liquidity baselines. Transaction count over rolling windows of 24 hours, 7 days, and 30 days, weighted by recency. Continuous days of operation since first observable activity. ERC-8004 reputation feedback events with non-zero economic value attached. Stable activity patterns versus burst-and-quiet patterns, which the formula reads as evidence of sustained service rather than ad-hoc promotional bursts.

How it is normalized. Mostly absolute, with chain-relative baselines. A volume of $10,000 routed through an agent on a low-throughput chain reads differently than $10,000 routed through a high-throughput chain. The normalization corrects for the baseline activity of the chain so that the signal reflects relative position within the agent's deployment context.

Failure modes that look strong. Wash-traded volume between counterparties controlled by the same operator. Self-dealing transactions that route capital in a circle to inflate volume metrics. Inflated TVL via short-duration deposits that are withdrawn after the rating window snapshots. Each of these is filterable through source-conflict discounting on the volume signal: transactions between addresses with high common-control probability are discounted, and short-duration deposits are weighted by time-on-deposit rather than peak-deposit.

The non-obvious calibration. Economic Viability is the category most often gamed by new agents that want to look established. The defense is not stricter thresholds. It is the time-decay parameter on volume:

short bursts of volume have lower weight than sustained volume over months, even when the integrated total is the same. *The formula prefers boring continuity to dramatic flashes,* which matches the actual signal that human credit markets use to evaluate counterparties.

4.2 Operational Reliability

What it measures. Whether the agent reliably performs the function it claims to perform, on the timescale its counterparties expect. Specifically: uptime of declared endpoints, failure rate on jobs the agent accepted, time to recovery after detected incidents, response latency distributions, and the frequency of operational anomalies that require manual intervention.

Why it is a trust signal. An agent that does not respond to a counterparty in time is a counterparty failure regardless of how good the underlying code is. Operational reliability is the difference between an agent that works on a developer's machine and an agent that works in production. *It is also the category most often skipped by raters who do not actively probe agents.* Treebeard probes.

What feeds it. Six signal sources. Active endpoint probes that measure response success and latency, run from independent infrastructure on a deterministic schedule. Public job-completion logs from any registry the agent participates in. Self-reported uptime statistics, weighted by source-conflict discount because the rated entity is the source. Third-party monitoring services that publish their own uptime data on the agent. Failure-mode logs from incident registries when present. The continuity-of-operation metric from Economic Viability, reused as a reliability anchor because no agent operates economically through long downtime windows.

How it is normalized. Mostly relative, against the agent's stated service-level claims and against the population baseline for agents in its category. An agent that claims 99% uptime is held to that claim. An agent that makes no SLO claim is benchmarked against the 50th-percentile reliability for its category and its chain.

Failure modes that look strong. Operational reliability metrics produced solely by the rated entity, with no independent probe. Agents that claim high reliability through long quiet periods that do not stress-test the system. Agents whose uptime is high but whose response latency is too slow to be useful in agent-to-agent commerce.

The non-obvious calibration. Reliability that looks good in a quarterly snapshot can hide systematic failures that cluster in time. The formula uses tail-latency percentiles rather than means: a 99th-percentile latency that exceeds the agent's stated SLO drops the reliability score even if the median latency is excellent. *The agents your counterparty depends on at the 99th percentile are the agents whose tail behavior matters most.*

4.3 Code Quality

What it measures. Whether the underlying code, the deployment story, and the dependency graph are in a state that a counterparty can reasonably trust. Specifically: presence and quality of public source code, depth and recency of commit history, presence of tests and audit artifacts, dependency hygiene including pinned versions and known-vulnerability scans, deployment story including reproducible builds where applicable, documentation that describes what the agent does without marketing copy.

Why it is a trust signal. A counterparty deciding whether to integrate an agent has limited options for verifying behavior beyond the code itself. If the code is unreadable, undocumented, or unobservable, the counterparty has no recourse beyond reputation. Reputation alone is too thin an evidence base for high-stakes integration. Code quality is what makes reputation auditable.

What feeds it. Seven signal sources. Public repository presence and commit graph statistics. Test coverage where measurable, including evidence of integration tests rather than only unit tests. Audit artifacts from third-party security firms, weighted by the reputation of the firm and the recency of the audit. Dependency analysis including pinned versions and CVE scans on declared dependencies. Build reproducibility from declared deployment artifacts. Documentation completeness and the ratio of substantive description to

marketing copy. ERC-8004 service-type declarations that match the actually-observable behavior, where mismatches between claimed and observed behavior penalize the score.

How it is normalized. A mix of binary signals (audit present or not, repository public or not) and absolute signals (test coverage percentage, time since last commit). The composite is weighted toward the binary signals at the floor and the absolute signals at the ceiling: agents below thresholds on the binaries cannot exceed a defined ceiling on the absolutes, which prevents a high test coverage from compensating for the absence of an audit on a financial agent.

Failure modes that look strong. Stale commit history that looks active because of cosmetic commits. High test coverage produced by tests that exercise nothing. Audits from firms with no public reputation, where the audit document exists but means nothing. Dependency lists that look pinned but resolve through unpinned transitives.

The non-obvious calibration. Closed-source agents can score in this category, but only with a credible audit trail to substitute for visible code. The audit trail must be from a third party and must include re-attestations on a schedule the agent's risk profile justifies. *No single artifact carries the full weight of code transparency by itself.* The category is composed precisely so that the absence of one signal can be compensated by the presence of others, but only up to a defined ceiling.

4.4 Autonomy Index

What it measures. The degree to which the agent actually operates without human intervention in the conditions it claims to handle. Specifically: rate of decisions made unattended, range of conditions handled without human-in-the-loop fallback, coverage of edge cases declared in the agent's function description, the frequency of escalations to a human operator versus the agent's claimed autonomy level.

Why it is a trust signal. Autonomy is the most overstated property in agent marketing. Many agents that claim to operate autonomously have a human approving every transaction in production, which means the agent is automation rather than autonomy. The trust implications differ. An automated system inherits the trustworthiness of its human operator. An autonomous system has to be trustworthy on its own. *Counterparties pricing risk need to know which one they are dealing with.*

What feeds it. Five signal sources. The frequency of unattended decisions, measured by the absence of human signatures on transaction logs in the windows where the agent claims to be autonomous. The range of edge cases the agent handles without escalation, derived from operational logs where available. The agent's stated autonomy level versus its observed autonomy level, with mismatches penalized through the source-conflict discount. The breadth of decision contexts the agent operates in, since narrow autonomy is easier to claim than broad autonomy. ERC-8004 service-type declarations and any AGENTS.md or function-description specification, weighted against observed behavior.

How it is normalized. Mostly relative, against the agent's claims. An agent that claims to be fully autonomous and operates with human approval on every transaction is heavily penalized, because the misrepresentation is itself a trust signal. An agent that claims modest autonomy and meets its claim scores well.

Failure modes that look strong. Agents that operate autonomously in narrow conditions and present that as broad autonomy. Agents that escalate frequently but report only successful unattended decisions. Agents whose stated function description describes capabilities that are not in the production agent.

The non-obvious calibration. Autonomy at high levels has to be earned by observable behavior under varied conditions, not granted by self-attestation. The formula caps autonomy scores for agents with insufficient operating history regardless of the rated entity's claims. Time builds autonomy. Marketing copy does not.

4.5 Safety

Safety appears twice in the methodology. Here, in Section 4.5, it is one of seven weighted categories that contribute to the composite under the standard formula. In Section 5, the same Safety category score also serves as a non-substitutable floor that gates the entire composite when it falls below threshold. The two roles are not redundant: the in-composite role calibrates the rating across the full Safety range, while the floor role ensures that very low Safety scores cannot be compensated by strong signals elsewhere. The dual role exists because the consequences of safety failures are asymmetric in a way the consequences of failures in the other six categories are not (Section 5.1).

What it measures. Whether the agent's permission scope, kill-switch infrastructure, rate limiting, behavior under adversarial input, and other guardrails are adequate to its declared function. The category is the load-bearing trust signal because it is the one that can cap the entire composite under the safety floor mechanism described in Section 5.

Why it is a trust signal. A high-performing agent with no safety guardrails is a more dangerous counterparty than a low-performing one, because its behavior cannot be predicted or contained when it operates outside its expected range. *Safety failures are unbounded.* A counterparty must be able to tell whether the agent has the structural protections necessary to fail gracefully.

What feeds it. Eight signal sources. Permission scope analysis on declared smart contracts, including the presence of admin functions, upgrade mechanisms, and emergency pause logic. Kill-switch presence and the responsiveness of the kill mechanism in test conditions. Rate limiting on agent endpoints. Behavior under adversarial input, measured through structured probes by independent infrastructure. Logged incidents and the agent's response to each. Public security disclosures and the agent's response time to each. Presence of bug-bounty programs and their published scope. ERC-8004 service-type declarations that include safety claims, validated against observed behavior.

How it is normalized. Mostly binary at the floor (kill switch present or not, permission scope reviewed or not), with absolute and relative signals layered above. The binary floor signals act as gates: an agent that fails any of them cannot exceed a defined ceiling on the composite, regardless of how strong other signals are.

Failure modes that look strong. Safety claims in marketing copy without the underlying mechanism. Kill switches that are present but not testable. Bug bounties that are published but unfunded. Agents whose adversarial robustness was tested at one point and not since.

The non-obvious calibration. Safety is the only category that can cap the entire composite. An agent with a safety score below threshold has its overall grade capped at D regardless of how high the other six categories score. Section 5 specifies the floor mechanism in detail. *This is the only place in the methodology where one category overrides the composite.* The reason is the asymmetry of safety failures: a safety lapse is unbounded loss for the counterparty, and no amount of strength in other categories compensates.

4.6 Community and Ecosystem

What it measures. The degree to which independent counterparties recognize the agent, integrate against it, and contribute feedback that survives source-conflict discounting. Specifically: independent integrations, third-party attestations, ecosystem citations in audit reports or research papers, ERC-8004 feedback events from non-aligned counterparties, presence in discovery layers run by entities with no commercial interest in the agent.

Why it is a trust signal. An agent that no one integrates against, attests to, or cites is an agent that has not been validated by use. Reputation through independent counterparties is the most reliable signal that an agent has been tested in production by entities that had something to lose. *It is also the category most prone to manipulation,* which is why the source-conflict discount is most aggressive here.

What feeds it. Six signal sources. Integration count from public manifests and ecosystem maps, weighted by the integrating entity's own community signal. Third-party attestations including audit reports, security

disclosures, research citations, with weight by the reputation of the attesting entity. ERC-8004 feedback events with source-conflict discount applied: feedback from addresses with high common-control probability with the rated entity is discounted toward zero. Citations in independent reports, news coverage, or research. Discovery layer presence including ecosystem maps maintained by entities with no rating relationship. Signal from agent-to-agent transactions where the counterparty is itself a high-rated agent.

How it is normalized. Heavily relative, with strong source-conflict discounting at the source level. The category is calibrated under the assumption that the rated entity may be the source of most of the apparent community signal. The source-conflict discount removes self-attestation, sock-puppet endorsement, and aligned-counterparty feedback before the composite is computed.

Failure modes that look strong. High feedback counts from a low number of distinct sources. Attestations from entities controlled by the rated entity. Citations in publications run by allied infrastructure. Discovery layer presence on platforms with rating-affiliation revenue.

The non-obvious calibration. The Community signal is intentionally hard to game, because the alternative is that it is the easiest signal to game. The discount factors here are aggressive by construction, which means agents with genuinely strong community uptake will see their score lift slowly and durably, and agents with manufactured community uptake will see their score never lift at all.

4.7 Security Posture

What it measures. The defensibility of the agent's surface area against the threat model it actually faces. Specifically: key management practices, contract upgrade mechanisms, dependency risk, operational infrastructure security, incident history and response, presence of post-incident transparency.

Why it is a trust signal. Safety, in Section 4.5, focuses on guardrails for the agent's intended behavior. Security Posture focuses on the defensibility of the agent against unintended behavior caused by external compromise. The two are separable. An agent can have excellent guardrails on its own behavior and a fully compromised key management story. The trust implications differ. Safety failures show up in the agent's own actions. Security Posture failures show up when the agent's identity is hijacked or its dependencies are compromised.

What feeds it. Seven signal sources. Key management hygiene, including evidence of multisig, threshold signing, hardware-backed keys, and key rotation. Smart contract upgrade mechanisms, including the presence of timelocks, multi-party guards, and emergency pause logic. Dependency vulnerability scans on declared dependencies. Operational infrastructure security signals from public posture: TLS configuration on declared endpoints, DNS hygiene, hosting transparency. Incident history and the published post-mortems for each. Bug-bounty program scope, funding, and historical payouts. ERC-8004 reputation events that flag security-related feedback.

How it is normalized. A mix of binary and absolute. The binary floor includes presence of multisig on contracts handling material value, presence of an incident-response process, presence of dependency scans. The absolute layer includes vulnerability counts, mean time to patch, and bug-bounty payout history.

Failure modes that look strong. Multisig configurations where all signers are controlled by one operator. Bug bounties that are scoped to non-critical surface area. Incident histories that look clean because incidents were not disclosed.

The non-obvious calibration. Security Posture is the category where mismatches between claim and observed behavior are most heavily penalized, because the asymmetry of the failure mode is high. An agent that claims strong security practice and is observed to violate it has the violation weighted as evidence of misrepresentation, not just of weak security. *The evidence of having claimed something that was not true has its own trust signal, separate from the underlying claim.*

4.8 The seven categories as a system

The seven categories were chosen as the minimum set that distinguishes between fundamentally different failure modes. Collapsing any two of them produces a rating that cannot tell certain real agent failures apart, and the inability to distinguish becomes a permanent ambiguity in every score that uses the collapsed structure.

Economic Viability and Operational Reliability cannot be collapsed. An agent can have high economic flow through it while being unreliable in tail conditions. An agent can be highly reliable while having no economic flow.

Code Quality and Security Posture cannot be collapsed. An agent can have excellent code in a poorly defended deployment. An agent can have minimal code with strong infrastructure protection.

Autonomy Index and Safety cannot be collapsed. An agent's autonomy level determines the conditions under which its safety guardrails are tested. The two questions are separable: how often does the agent act unattended, and how protected is the agent when it does.

Community and Ecosystem cannot be collapsed into the others. The signal it carries, that of independent validation through use, is structurally different from any of the operational, technical, or economic categories.

The seven are not arbitrary. They are the smallest set that retains the discriminating power needed to give counterparties a usable rating.

The weight profile across the seven categories varies by agent type. A trading agent has a different weight profile than a customer-support agent. The category structure is the same in both cases. *What differs is the weights, not the categories.* The category weights themselves are described in Section 6, and the calibration loop that updates them in response to evidence is described in Section 9.

Section 5. The safety floor

5.1 The asymmetry of safety failures

Section 4.5 noted that Safety appears twice in the methodology: once as one of seven weighted categories, and again as a non-substitutable floor that gates the composite. This section specifies the floor.

The reason for the dual role is the asymmetry of the loss function on safety failures. A counterparty paying for a service from an agent with weak Operational Reliability gets slow responses or failed transactions; the cost is bounded by the value of the one transaction. A counterparty paying for a service from an agent with weak Safety can lose far more than the value of any single transaction, because a safety failure can route capital to attackers, expose private data, or cascade into downstream systems whose total exposure dwarfs any individual interaction. *Safety failures do not have a natural upper bound.* The asymmetry is the structural reason Safety cannot be aggregated into a composite the same way the other categories are.

5.2 The floor mechanism in detail

Let $S(a)$ denote the Safety category score for agent a, normalized to the range $[0, 100]$. Let $R(a)$ denote the composite score across all seven categories under the standard formula given in Section 6. Let T_{floor} denote the safety floor threshold, currently set at 50.

The published rating $R^*(a)$ for agent a is:

$$R^*(a) = \begin{cases} R(a) & \text{if } S(a) \geq T_{\text{floor}} \\ \min(R(a), R_{\text{cap}}) & \text{if } S(a) < T_{\text{floor}} \end{cases}$$

where $R_{\text{cap}} = 60$, corresponding to the upper bound of the D grade tier.

In words: an agent that fails the safety floor cannot earn higher than a D, regardless of how strong its other signals are. An agent with a perfect Autonomy Index, Code Quality, and Community score but weak Safety reads in the published rating exactly as a low-grade agent. *The composite does not reward strength elsewhere when the safety floor fails.* The asymmetry of the loss requires the asymmetry of the rating mechanism.

The cap is also asymmetric in the opposite direction. If $R(a)$ is already below R_{cap} before the floor check, the floor changes nothing. The floor activates precisely in the case where an agent has weak Safety and strong other signals, which is exactly the adversarial-stacking case the floor is designed to address. An agent that scores poorly on Safety and poorly elsewhere does not need the floor to land at a low grade. The floor is the structural defense against the agent that uses strong other categories to pull the composite up over a weak Safety floor, not against the agent that is uniformly weak.

5.3 Why a binary floor and not a continuous penalty

A natural alternative is a continuous penalty function: as Safety drops, apply a multiplicative discount to the composite. This produces a smoother gradient and feels more proportional. It is the wrong choice, for two reasons.

First, the loss function is not smooth. The cost of a counterparty integrating with an agent that has a Safety score of $T_{\text{floor}} - 1$ is not marginally higher than the cost of integrating with an agent at $T_{\text{floor}} + 1$. The cost is structurally different, because the agent below the threshold lacks the basic protections that bound failure modes. A continuous penalty understates the discontinuity in real-world consequences.

Second, a continuous penalty is gameable. An adversary can target the level of penalty that compensates for known weaknesses elsewhere, producing an agent that just clears the trust threshold despite having identifiable safety gaps. A binary floor removes that optimization surface. *There is no level of strength elsewhere that compensates for falling below the floor.*

The cost of a binary floor is occasional false positives: agents with material safety gaps in narrow areas that nonetheless do not produce real harm in practice. The Ent Review Panel (Section 8) handles these cases through the dispute process. The default behavior of the rating is to fail safe.

5.4 What gates the floor

The Safety category has eight signal sources, specified in Section 4.5. The floor threshold is computed against the composite Safety score, not against any single signal. The composite includes:

- Permission scope analysis on declared smart contracts
- Kill-switch presence and responsiveness
- Rate limiting on agent endpoints
- Behavior under adversarial input, measured through structured probes
- Logged incident history
- Public security disclosure responsiveness
- Bug-bounty program presence and scope
- ERC-8004 service-type declarations validated against observed behavior

An agent failing on a single source can still clear the floor if other sources are strong. An agent failing on multiple sources cannot. The floor threshold is calibrated such that its activation reflects a structural safety gap, not a single weak signal.

5.5 The floor as a published commitment

The floor threshold is published in this paper and on the methodology pages on the Treebeard website. Changes to the threshold are versioned in the public methodology changelog. The floor cannot be quietly raised to clear an agent through to a higher grade or quietly lowered to flag an agent into a lower one. The

threshold is a fixed parameter of the rating, modifiable only through documented methodology updates that show in the version history.

This commitment is the structural counterpart to the no-issuer-pays commitment from Section 3.1. The threshold cannot be bargained, purchased, or negotiated. It is the same threshold for every agent, applied through the same mechanism, with the same outcome.

5.6 The floor in context

Treebeard is not the first rating system to use a non-substitutable floor on a critical category. The pattern is older than computer-mediated finance.

Insurance ratings (Best, Demotech) gate solvency below a threshold, regardless of management quality or product mix. Aviation safety ratings cap a carrier's overall standing if an audit identifies critical safety findings, regardless of operational performance elsewhere. Pharmaceutical drug approvals require safety findings independent of efficacy: a drug that works perfectly but produces unbounded harm cannot be approved on the strength of efficacy alone.

In each case, the floor is the institutional encoding of an asymmetric loss function. The agent economy needs the same primitive at the rating layer because the loss functions on agent failures are structurally similar. The floor is not a Treebeard invention. It is the right primitive for asymmetric-loss systems.

Section 6. Scoring mechanics

6.1 The composite at a glance

The Treebeard composite for agent a is the weighted aggregation across seven category scores, with each category score itself an aggregation across signal sources, with each signal corrected by a source-conflict discount and a time-decay factor. The full formula is:

$$R(a) = \sum_{c=1}^{7} w_c(\text{type}(a)) \cdot s_c(a)$$

where $w_c(\text{type}(a))$ is the weight assigned to category c given the agent's type, and $s_c(a)$ is the category-level score for agent a on category c.

Each category score $s_c(a)$ is itself the weighted aggregation, rescaled to a 0-to-100 output range:

$$s_c(a) = 100 \cdot \sum_{i \in \text{sources}(c)} \alpha_i \cdot d_i \cdot \tau_i(t) \cdot n_i(a)$$

where the index i ranges over the signal sources that feed category c, $\alpha_i \in [0,1]$ is the within-category weight on source i (with $\sum_i \alpha_i = 1$), $d_i \in [0,1]$ is the source-conflict discount on source i, $\tau_i(t) \in [0,1]$ is the time-decay factor on source i at the current time t, and $n_i(a) \in [0,1]$ is the normalized signal value from source i for agent a. The product is in $[0,1]$ before rescaling and in $[0,100]$ after. The composite $R(a)$ is therefore also in $[0,100]$, since the category weights w_c also sum to 1.

This is the full equation. The remainder of this section specifies each component.

6.2 Signal normalization $n_i(a)$

A raw signal value can take many forms: a count of feedback events, a percentage uptime, a binary present-or-not, a continuous score from an external evaluator. The normalization n_i converts these into a comparable scale, defined per source.

Three normalization patterns are used.

Binary. A signal that exists or does not. Examples: presence of an audit, presence of a kill switch, presence of multisig on the agent's contract. Normalized to $\{0, 1\}$.

Absolute. A signal whose magnitude is interpreted directly against a published scale. Examples: percentage of unattended decisions, count of feedback events, mean response latency. Normalized to $[0, 1]$ via a published mapping function. The mapping function is sigmoid for unbounded counts (so that the marginal effect of additional feedback events diminishes at scale) and linear for bounded percentages.

Relative. A signal whose value is interpreted against a baseline drawn from the agent's category and chain. Examples: total value routed through the agent (against chain-relative liquidity baseline), uptime versus stated SLO, latency versus 50th-percentile latency for the agent's category. Normalized to $[0, 1]$ via percentile rank within the relevant comparison cohort.

The normalization function for each source is published per source in the methodology pages.

6.3 Source-conflict discount d_i

The source-conflict discount is the central anti-gaming primitive of the Treebeard methodology. It is the structural fix to the equal-weight aggregation fallacy named in Section 3.6.

For each signal source i, the discount $d_i \in [0, 1]$ is a multiplicative factor that reflects the structural reliability of the source. A source that is itself a token-issuing rating provider receives a discount factor near zero. A source that aggregates self-attesting feedback receives a sybil-resistance-adjusted discount. A source that has been demonstrably gamed in the past receives a historical-manipulation discount.

The discount factor is decomposed as:

$$d_i = d_i^{\text{conflict}} \cdot d_i^{\text{sybil}} \cdot d_i^{\text{history}}$$

where each component is in $[0, 1]$.

d_i^{conflict} reflects the structural conflict of interest of the source. A source with no economic alignment to the rated entity receives a value near 1. A source whose revenue depends on the rated ecosystem's growth receives a value lower than 1, calibrated based on the magnitude of the alignment.

d_i^{sybil} reflects the manipulability of the source by sybil attack. A source where attestations come from cryptographically distinct identities with verified independence receives a value near 1. A source where attestations can be cheaply produced from sock-puppet wallets receives a lower value.

d_i^{history} reflects observed manipulation of the source. A source with no documented history of manipulation receives a value of 1. A source with documented manipulation events receives a value lower than 1, with the magnitude of the discount calibrated against the magnitude of the manipulation.

All three components are published per source. They are updated by the Bayesian calibration loop specified in Section 9.

6.4 Time-decay factor $\tau_i(t)$

The time-decay factor is the second central anti-gaming primitive. It is the structural fix to the time-stale signal fallacy named in Section 3.6.

Time decay is modeled as exponential decay with a half-life h_i specific to each signal source:

$$\tau_i(t) = 2^{-(t - t_i^{\text{obs}})/h_i}$$

where t_i^{obs} is the time of observation of the most recent value from source i, and t is the current time. The factor is 1 at the moment of observation and halves every h_i time units.

The half-life h_i is calibrated per source based on the rate at which the underlying property changes. A live endpoint probe has a short half-life (hours), because endpoint behavior changes quickly. A code audit has a longer half-life (months), because audit findings remain relevant longer. Operational continuity has the longest half-life (years), because long operational history is durable evidence.

Half-lives are published per source. Like the discount factors, they are updated by the calibration loop.

6.5 Within-category weights α_i and category weights w_c

The within-category weight α_i assigns the relative weight of source i within its category. Sources that produce more reliable signal in calibration receive higher within-category weights. Sources that are noisier or more easily gamed receive lower weights. The within-category weights for category c sum to 1:

$$\sum_{i \in \text{sources}(c)} \alpha_i = 1$$

The category weight $w_c(\text{type}(a))$ assigns the relative weight of category c in the composite, conditional on the agent's type. Trading agents have higher weight on Economic Viability and Operational Reliability. Customer service agents have higher weight on Autonomy Index and Community. Infrastructure agents have higher weight on Code Quality and Security Posture. The category weights for any agent type sum to 1:

$$\sum_{c=1}^{7} w_c(\text{type}(a)) = 1$$

The set of agent types is defined and published. New types are added through the methodology versioning process.

6.6 Hysteresis on grade transitions

A composite score in $[0, 100]$ maps to a letter grade through a published threshold mapping (A, A-, B+, B, B-, C+, C, C-, D+, D, F). The naive mapping rounds the composite to the nearest threshold. This produces grade oscillation when an agent's composite hovers near a threshold, because routine signal fluctuation can flip the published grade up and down day to day.

To prevent this, Treebeard applies hysteresis to grade transitions. Each grade G has an upper threshold T_G^+ (the score at which the grade transitions to the next-higher grade) and a lower threshold T_G^- (the score at which the grade transitions to the next-lower grade). Let $G_t(a)$ denote the published grade of agent a at time t, and let $g(R)$ denote the threshold mapping of composite R. The grade at time $t+1$ follows:

$$G_{t+1}(a) = \begin{cases} g(R(a)) & \text{if } R(a) > T^+_{G_t(a)} + \delta \\ g(R(a)) & \text{if } R(a) < T^-_{G_t(a)} - \delta \\ G_t(a) & \text{otherwise} \end{cases}$$

where δ is the hysteresis buffer. In words: a grade transitions upward only when the composite has moved more than δ above the current grade's upper threshold, and transitions downward only when the composite has moved more than δ below the current grade's lower threshold. Inside the hysteresis band the grade is sticky. This prevents oscillation while still tracking real changes in the underlying signal.

The hysteresis buffer δ is calibrated per grade boundary. It is wider near the safety floor than elsewhere, because false transitions across the floor have higher cost.

6.7 A worked example

Consider an agent of type "trading", with the following category scores after normalization, conflict discount, and time decay:

Category	$s_c(a)$	w_c(trading)	Contribution
Economic Viability	75	0.20	15.00
Operational Reliability	70	0.20	14.00
Code Quality	65	0.15	9.75
Autonomy Index	60	0.10	6.00
Safety	80	0.15	12.00
Community	55	0.10	5.50
Security Posture	70	0.10	7.00
Composite		**1.00**	**69.25**

The composite of 69.25 maps to a grade of C+ under the threshold mapping. The Safety score of 80 exceeds the floor threshold of 50, so the floor does not activate. The published grade is C+.

Now consider a counterfactual where the same agent has all the same category scores except Safety, which drops from 80 to 30. The contribution from Safety drops from 12.00 (80×0.15) to 4.50 (30×0.15). The new pre-floor composite is $69.25 - 12.00 + 4.50 = 61.75$, which under the threshold mapping would still produce a grade in the C tier. The floor then activates because Safety of 30 is below the threshold of 50. The cap of $R_{\text{cap}} = 60$ applies, so the published composite is $\min(61.75, 60) = 60$. The published grade is D.

This illustrates the asymmetric work the floor is doing. The pre-floor composite of 61.75 is high enough that, without the floor, the agent would clear into the C tier despite weak Safety. The floor is what prevents that signal-stacking outcome. Changes in any other category produce only the normal contribution change. Changes in Safety, when they cross the threshold, produce both a normal contribution change and a potential floor activation.

6.8 Re-rate cadence

The composite is recomputed for every rated agent on a daily cadence and additionally on enrichment events that materially change a signal source. An enrichment event is any new audit, any logged incident, any change in operational continuity, any new feedback event from a high-weight source. The continuous re-rating is the structural fix to the static-snapshot failure mode in Section 3.3.

The re-rate produces a new composite, which then passes through two transformations in a fixed order before being published.

Step 1, the safety floor check. The Safety category score is compared to T_{floor}. If $S(a) < T_{\text{floor}}$, the composite $R(a)$ is replaced by $\min(R(a), R_{\text{cap}})$ to produce the floor-adjusted composite $R^*(a)$. If $S(a) \geq T_{\text{floor}}$, the floor-adjusted composite equals the unmodified composite.

Step 2, the hysteresis grade transition. The floor-adjusted composite $R^*(a)$ is then passed through the hysteresis rule from Section 6.6 against the agent's currently published grade to produce $G_{t+1}(a)$.

The order is intentional. The floor operates on the composite. The hysteresis operates on the floor-adjusted composite. This ordering means an agent oscillating in Safety just above and below T_{floor} produces oscillation in $R^*(a)$ between the unmodified composite and the capped composite, but the hysteresis rule on $R^*(a)$ damps the resulting grade transitions. The hysteresis buffer δ near the floor is set wider than at other grade boundaries (Section 6.6) precisely because of this expected behavior. The combination of "floor first, hysteresis on the result" is what produces a stable published grade across small fluctuations in Safety scores around the threshold. The floor itself does not have hysteresis; the floor is a sharp condition on the current Safety score, and the smoothing happens at the published-grade layer.

The full pipeline runs deterministically with no human in the path.

6.9 On-chain oracle implementation

The published rating is also available as an on-chain primitive on Base, queryable by smart contracts and by other agents at handshake time. The oracle interface is the integration surface that makes a Treebeard rating part of the agent commerce protocol stack rather than a website that humans check periodically.

What is on-chain. The oracle exposes, per agent, the current letter grade, the current numeric composite, the safety floor activation flag, the methodology version that produced the current rating, and the timestamp of the most recent update. Per-category scores and per-source signal contributions are not currently published on-chain. Consumers needing category breakdown query the public API at api.treebeardai.com.

Update cadence. The oracle updates on the same cadence as the rating itself: once per day on the regular cycle, with ad-hoc updates triggered by enrichment events that materially change a signal source. Update timestamps are part of every read, so consumers can detect staleness without an additional call.

Stale reads. Stale reads are an explicit failure mode. The oracle does not enforce a global maximum staleness, because different consumers have different freshness requirements (a trading agent reading a counterparty rating has stricter requirements than a research agent verifying provenance). Each read returns the timestamp of the most recent update, and consumers are expected to enforce a freshness threshold appropriate to their stake.

Disputed ratings. When a rating is under active dispute that has passed an initial admissibility check, the oracle returns a `disputed` flag set to true. The admissibility check is a structural protection: a dispute filing alone does not flip the flag, because if it did, any actor could file a frivolous dispute to mark a competitor's rating as disputed and create a denial-of-service surface. The Ent Review Panel reviews each filing for admissibility before the flag is set, with the admissibility criteria published alongside the dispute pathway (the dispute names a specific signal, methodology component, or factual claim that is alleged to be wrong, and provides evidence sufficient to be reviewable). Filings that fail admissibility are logged but do not flip the flag. Consumers reading a `disputed` rating can choose to honor it despite the dispute, treat the dispute as a soft signal to discount the rating, or refuse to act on the disputed rating until the dispute resolves. The oracle does not enforce a particular consumer-side policy.

Methodology versioning. Every rating read is tagged with the methodology version that produced it. Consumers can detect when a rating they have cached was produced under a methodology version that has since been deprecated and refresh accordingly.

Failure modes. Three failure modes are documented. First, the oracle becomes unreachable due to chain or RPC issues: consumers fall back to the public API, which is independently hosted. Second, the oracle returns a fresh rating produced under a deprecated methodology: the version tag in the read makes this case detectable. Third, the data publication pipeline lags behind the rating engine: the lag is bounded by an internal SLA, and material deviations are published in the methodology changelog.

Why on-chain at all. The agent-to-agent handshake is the core flow that the rating is built to inform. Smart contracts that gate access to capital or to liquidity on a rating threshold cannot read the rating from an off-chain API in a single transaction. They need an on-chain primitive. The same is true of agent-to-agent commerce flows where the buyer agent's decision logic checks the seller's grade synchronously. The oracle is the integration that closes both gaps.

Section 7. Anti-gaming design

7.1 The threat model

Any rating system that assigns scores to entities with economic stake in those scores faces adversarial pressure. The pressure is rational and continuous. The defenses must be structural, not policy-based, because policies can be circumvented while structural protections cannot.

The threat model for Treebeard includes five categories of attack.

Sybil attacks on signal sources that aggregate attestations from independent identities. An attacker produces many cryptographically distinct identities and uses them to inject favorable signal for the attacker's own agent.

Wash trading on signal sources that measure transaction volume or counterparty interaction. An attacker produces apparent volume by routing transactions between addresses the attacker controls.

Audit laundering on signal sources that consume third-party audits. An attacker passes one audit, then quietly modifies the agent or its contracts in ways that invalidate the audit, while the rating system continues to credit the audit signal.

Marketing inflation on signal sources that consume self-attested claims about agent behavior. An attacker overstates capabilities, autonomy levels, or operational characteristics in metadata or function descriptions.

Rater-of-raters manipulation on signal sources that consume scores from other rating providers. An attacker influences a downstream rating provider to issue a favorable rating that then feeds upstream into the Treebeard composite.

Each attack is defended by a specific structural mechanism, plus the general source-conflict discount and time-decay factors specified in Section 6.

7.2 Defense against sybil attacks

The general defense is the source-conflict discount component d_i^{sybil}. Sources that aggregate attestations are weighted based on the cost an attacker would face to fake the attestations. ERC-8004 reputation events from arbitrary addresses receive low d^{sybil}. ERC-8004 reputation events from addresses with verified independent activity histories receive higher d^{sybil}.

The mechanism for assessing address independence draws on multiple inputs: transaction history depth, diversity of counterparties, age of address, and graph-distance from the rated agent's known control set. These inputs are aggregated into a sybil-resistance score per address, which then weights that address's contribution to the source-level signal.

In addition, a sybil penalty applies when the population of attesting addresses for a single agent shows graph-clustering inconsistent with independent counterparties. Let N denote the number of attesting addresses, let C denote the count of clusters identified by graph analysis, and let σ denote the sybil penalty:

$$\sigma = \frac{C}{N}$$

Since $C \leq N$ by construction, $\sigma \in [0, 1]$. A signal source where $N = 100$ and $C = 100$ has $\sigma = 1$ and the signal passes through unmodified. A source where $N = 100$ and $C = 5$ (because all 100 addresses cluster into 5 control groups) has $\sigma = 0.05$ and the signal is reduced by 95 percent.

The sybil penalty σ is the operationalization of d_i^{sybil} from the source-conflict discount in Section 6.3 for sources whose primary risk is sybil-flooded attestation. Where a source's d_i^{sybil} requires concrete computation, σ provides it. The two terms are not separate factors compounded against each other; σ is the value plugged into d_i^{sybil} for sources of this type. For sources whose primary risk is not sybil flooding, d_i^{sybil} is calibrated through other inputs, and σ does not apply.

7.3 Defense against wash trading

Volume-based signals in the Economic Viability category are vulnerable to wash trading. The defense is counterparty-control inference: transactions between addresses identified as likely under common control are weighted lower or excluded from the volume signal.

The inference uses the same address-clustering logic as the sybil defense, applied to the transaction graph rather than the attestation graph. Volume between identified clusters is excluded. Volume to addresses

with no other observable activity is weighted lower than volume to addresses with deep, diverse transaction histories. The d^{conflict} discount on the volume signal is itself adjusted by the inferred wash-trading proportion.

7.4 Defense against audit laundering

Audit signals are vulnerable to the post-audit silent rebuild attack named in Section 2.5.1. The defense has two components: time decay on audit signals with category-specific half-life, and a redeployment penalty triggered when the agent's contracts change after the audit.

The audit signal has a half-life h_{audit} measured in months. An audit performed twelve months ago contributes less to the Code Quality and Security Posture categories than an audit performed last month, even if the audit document is identical. This corrects the static treatment of audits that produced the failure mode.

The redeployment penalty is an additive override. If a contract listed in the audit scope is replaced after the audit date without a corresponding re-attestation from the auditor, the audit signal contribution drops to zero on the affected components. The full Code Quality and Security Posture composite is reduced accordingly.

7.5 Defense against marketing inflation

Self-attested capabilities and autonomy claims are vulnerable to marketing inflation. The defense is the mismatch-as-signal mechanism: when an agent's stated capabilities differ materially from its observed behavior, the mismatch itself is treated as a negative trust signal, separate from the underlying capability gap.

The mechanism is operationalized as a mismatch score $m(a) \in [0, 1]$ derived from the difference between stated and observed properties on each measurable axis. The mismatch score is then applied as a multiplicative penalty on the Autonomy Index and Code Quality categories:

$$s'_c(a) = s_c(a) \cdot (1 - m(a))$$

In words: claiming more than the agent actually does costs the agent more than claiming less than it actually does. The asymmetry is intentional. *Honest understatement is preferable to dishonest overstatement,* and the rating reflects that.

7.6 Defense against rater-of-raters manipulation

Treebeard does not accept ratings from other rating providers as primary inputs. Where another rating provider's score is referenced, the reference is treated as a Community-category attestation rather than a Code Quality or Security Posture signal, and the source-conflict discount d^{conflict} is computed against the structural conflicts of the originating rater.

A rating from a token-issuing rater receives near-zero d^{conflict}. A rating from a marketplace-affiliated rater receives a discount calibrated against the marketplace's economic stake. A rating from an open-source decentralized rater (AgentRank, from Intuition, is the current example: an open-source ranker with sybil-resistance built into its design [Intuition, 2026]) receives a higher d^{conflict}, because the structural conflicts are minimal.

The composite formulation in Section 6 explicitly applies d^{conflict} across all sources, including upstream raters. There is no mechanism by which a sufficiently optimistic external rating can drag the Treebeard composite upward without passing the discount filter.

7.7 The bug bounty surface

Anti-gaming mechanisms themselves are subject to evasion attempts. Treebeard maintains a bug bounty for novel attack discoveries. Researchers who identify a way to clear the trust threshold through a previously-unmodeled attack pattern receive published recognition and a payment scaled to the severity of the attack.

Discovered attacks are then closed through methodology updates that flow through the calibration loop in Section 9.

The bounty serves a structural purpose. *Adversaries are continuously probing the rating.* Adversaries who discover a working attack and exploit it silently produce more damage than adversaries who discover the attack and disclose it for payment. The bounty channels adversarial energy into the disclosure pathway.

Section 8. The Ent Review Panel

8.1 Why qualitative review is necessary

A fully automated rating produces consistent, reproducible scores. It also produces edge cases that the methodology has not anticipated. New agent types, new attack patterns, agents that legitimately operate outside the assumptions baked into the formula. Without a qualitative override mechanism, the rating system either issues a wrong score on these cases or refuses to score them at all.

The Ent Review Panel is the qualitative override. It is the same primitive that traditional rating agencies use: a body of reviewers who handle disputes, edge cases, and methodology updates that fall outside the automated pipeline. The Panel's authority is bounded. It cannot rate agents that have not first been scored by the automated pipeline. It cannot issue a score that is not justified by published methodology. Its decisions are versioned and public.

8.2 Composition

The Panel consists of two layers, of which only the first is currently populated.

The standing layer, currently active, is the methodology owner (the founder) supplemented by the AI-simulated expert panels described in the methodology workplan: a credit rating analyst, a mechanism designer, a blockchain security reviewer, and an AI safety researcher. Each AI panel is implemented as a deterministic prompt-and-rubric system whose outputs are published alongside the Panel's decisions. The standing layer is materially smaller than what 2008-era bond rating committees maintained. This is acknowledged: it is a starting point, not the end state.

The expanded layer, not yet populated, will include independent human reviewers from the credit rating, mechanism design, blockchain security, and AI safety domains. The expansion is gated on two conditions: sufficient dispute volume to justify the addition, and demonstrated funding for the reviewers (the Panel's expansion is operationally separate from the rating system's revenue, since Treebeard takes no payment from rated entities). Section 11 lists the current Panel composition as an open limitation of the methodology, and the path to expansion as the corresponding remediation.

8.3 Cadence

The Panel reviews disputed ratings on a weekly cadence by default, with high-priority cases (safety floor activations, major methodology questions, ratings affecting agents of material economic stake) handled ad hoc. The review of any single dispute proceeds through three stages: automated re-evaluation (rerunning the pipeline with current data), AI panel evaluation against the methodology, and human review by the methodology owner.

Decisions are published with the methodology version that produced them. A reader can reconstruct the full reasoning chain from the public methodology and the published decision rationale.

8.4 The dispute pathway

A rated entity that disputes its score files through the public dispute process at /methodology/improve. The submission includes the agent's identifier, the contested score, the basis for the dispute, and any supporting

evidence the rated entity wishes the Panel to consider.

The Panel processes each dispute through the three-stage review and produces one of three outcomes:

Score adjustment. The dispute identifies a real error in the automated pipeline (a stale signal, a misattributed source, a normalization edge case). The score is adjusted, and the decision is published.

Methodology adjustment. The dispute identifies a real gap in the methodology (a category the methodology does not yet handle correctly, a signal source whose calibration is wrong). The methodology is updated through the formal versioning process. The score is recomputed under the updated methodology.

Score stands. The dispute does not identify a real error or gap. The score remains as published. The decision rationale is published, including the reasoning for why the dispute did not move the score.

In all three cases, the outcome is public. *No dispute outcome is hidden, even when the score does not change.* The transparency of the dispute pathway is the structural counterpart to the transparency of the methodology.

8.5 What the Panel does not do

The Panel does not issue ratings outside the automated pipeline. It does not provide ratings on demand to rated entities or third parties. It does not negotiate scores. It does not accept payment from rated entities for review.

The Panel exists to handle the edge cases the methodology has not anticipated and to update the methodology when systematic gaps are identified. It is the structural accountability mechanism for the rating, not a back channel for negotiated outcomes.

8.6 Why this is the 2008 fix

The 2008 bond ratings industry had no equivalent to a public dispute pathway with versioned outcomes. Disagreements with ratings were handled through internal channels at the rater. Methodology updates were not published. Decisions on individual ratings were not reviewable by the public.

The Panel mechanism, even in its current scaled-down form, is the structural counterpart to the public dispute pathway 2008-era raters lacked. The dispute is filed in public. The decision is published with reasoning. The methodology version that produced it is recorded. The accountability runs through the procedure rather than through the size of the committee. *Procedure matters more than headcount* for this specific structural protection. The expansion to a larger panel is a strengthening of the mechanism, not the mechanism itself. Section 11 lists the current Panel composition as a limitation that the methodology acknowledges and is actively remediating.

Section 9. Bayesian calibration

9.1 Why the methodology has to update

A rating methodology calibrated on 2026 data does not produce well-calibrated outputs in 2027 if the underlying agent population, the signal landscape, or the attack surface has shifted. The calibration parameters specified in Section 6, including the within-category weights α_i, the source-conflict discounts d_i, and the time-decay half-lives h_i, must update as new evidence about source quality arrives.

The update mechanism is Bayesian. New observations about a signal source's predictive accuracy update a posterior distribution on that source's parameters. The published methodology then reflects the posterior mean (or, where appropriate, a more conservative percentile) of the relevant parameter.

The Bayesian framing has two virtues. First, it specifies how prior beliefs combine with new evidence in a principled way, so the update is reproducible. Second, it makes the rate of methodology change explicit:

a source with a strong prior (substantial calibration history) will move slowly in response to new evidence, while a source with a weak prior (newly added) will move faster.

9.2 The update equation

For each signal source i, let θ_i denote a calibration parameter (for example, the source-conflict discount d_i or the time-decay half-life h_i). Let $p(\theta_i)$ denote the prior distribution on θ_i, calibrated against the source's history before the current update window. Let $\mathcal{D}$ denote the new evidence collected during the update window.

The posterior is given by Bayes' rule:

$$p(\theta_i \mid \mathcal{D}) \propto p(\mathcal{D} \mid \theta_i) \cdot p(\theta_i)$$

The published parameter $\hat{\theta}_i$ is then drawn from the posterior, typically as the posterior mean for stable parameters and as a lower percentile (e.g., the 10th percentile) for parameters that govern safety-critical behavior, where conservative estimation is preferred.

The choice of prior is conjugate where possible (Beta priors for proportion parameters, Gamma priors for rate parameters), so the posterior has a closed-form update rule. Where conjugate priors are not appropriate, the posterior is computed via Markov chain Monte Carlo against published numerical methods.

9.3 What the evidence looks like

The new evidence $\mathcal{D}$ for a signal source includes:

- Disputed ratings where the dispute pathway identified the source as a contributor to error
- Discovered manipulation events on the source (sybil attacks, wash trading, audit laundering)
- Observed performance of the source's signal in predicting downstream outcomes (rated agents that subsequently failed in production)
- Comparison of the source's signal against independently-collected ground truth where available

Each of these inputs updates a specific parameter on the source. A discovered manipulation updates the historical-manipulation discount d^{history}. An observed prediction failure updates the within-category weight α_i. A change in the rate at which the source's signal becomes stale updates the half-life h_i.

9.4 The update cadence and the calibration-shift summary

The full calibration update runs on a quarterly cadence. Between updates, individual source parameters can be adjusted ad hoc when high-confidence evidence arrives (a confirmed manipulation event, a major change in source data quality), with the adjustment recorded in the methodology changelog.

The quarterly update cycle produces a new methodology version, deployed through the formal versioning process. The structural changes in each version are published in full. The exact numerical parameter values remain internal per the calibration-opacity choice in Section 10, but each version ships with what we call the **calibration-shift summary**: a published statistical description of how the new parameters affect existing ratings. The summary includes the median delta to the composite score across all rated agents, the 95th-percentile delta, the count of agents whose published grade transitions across a threshold under the new parameters, and a per-category breakdown of where the calibration moved.

The calibration-shift summary is the structural answer to a question Sections 6.9 and 10 leave implicit: how does a consumer of the rating know that a methodology change has affected the rating they relied on yesterday? Methodology updates are not deprecation events. A pre-update rating of 73 and a post-update rating of 73 may not mean the same thing in absolute terms even though the number is identical. The shift summary lets a counterparty pricing risk against a Treebeard rating reason about whether to re-evaluate cached ratings, with quantitative bounds rather than guesswork.

A reader who wants to reproduce a historical rating can pull the methodology version that was active at the time of the rating along with that version's calibration-shift summary against the next published version. The combination is sufficient to compute the expected shift on any individual agent's score across versions.

9.5 The conservatism bias

The Bayesian framing allows explicit conservatism on safety-critical parameters. The safety floor threshold T_{floor} is calibrated as follows: given a posterior distribution over the smallest Safety score that adequately separates safe agents from unsafe ones, the published threshold is set at the 90th percentile of that posterior, not at the posterior mean. In practical terms, this means we set the threshold higher than the posterior mean would suggest, so that more agents fall below the floor than the central estimate would require. The choice produces a higher false-positive rate on the safety floor (agents flagged as below the floor when they would in fact be safe under the posterior mean) in exchange for a lower false-negative rate (unsafe agents passing the floor when they should not).

The asymmetry is intentional and matches the asymmetric loss function from Section 5. *Treebeard prefers to flag a safe agent as unsafe than to pass an unsafe agent as safe.* The conservatism is published as part of the calibration choices.

Section 10. The transparency model

10.1 What transparency means here

Transparency in a rating context means that a reader can reproduce the rating from public inputs. The rating's structure, signal sources, normalization functions, weight profile, anti-gaming mechanisms, and update cadence are all publishable, and Treebeard publishes them. A reader who has access to the same on-chain signals Treebeard uses can in principle compute the same rating from scratch.

What is intentionally not published is the precise numerical calibration of weights α_i, conflict discounts d_i, time-decay half-lives h_i, and the safety floor threshold T_{floor}. The reasons were given in Section 6 and Section 7 and are summarized here.

10.2 The structural-vs-calibration distinction

Structural transparency is the publication of categories, signal sources, normalization patterns, and the formula shape that produces a composite from those inputs. Calibration transparency is the publication of the precise numerical parameters that calibrate the formula.

Treebeard publishes all of structural transparency. The seven categories, the signal sources for each, the binary-absolute-relative normalization patterns, the safety floor mechanism, the source-conflict discount and time-decay framework, the hysteresis grade transition rule, the Bayesian calibration loop. None of these are confidential.

Treebeard does not publish calibration transparency. The exact within-category weight on Code Quality versus Operational Reliability is internal. The exact safety floor threshold is internal. The exact half-life on the audit signal is internal.

The reason is that calibration transparency invites optimization-for-the-rating, which produces measurable signal that no longer measures the underlying property. *Goodhart's Law applies in print.* Every credit rating system that has tried full calibration transparency has watched its metric collapse into a target.

10.3 The FICO precedent and where the bridge is incomplete

The FICO architecture is the right target for a credible calibration-opaque rating. The bond ratings architecture from before 2008 is not. Treebeard's claim is that we sit closer to FICO than to S&P pre-2008,

while honestly acknowledging that the structural conditions that make FICO defensible do not all transfer to agent rating yet, and that we substitute mechanisms documented in this paper for the conditions we do not yet have.

The three FICO conditions and Treebeard's substitutions:

FICO's structural condition	Treebeard's current state	Substitution
Underlying signals are objectively measurable with low noise (loan repayments, balances, account age)	Agent signals are noisier and more easily manipulated	Multi-source composition (Section 4) and source-conflict-discount plus time-decay corrections (Section 6) suppress the influence of any single low-quality source
Consumers of the rating (lenders) have direct skin in the game and would route around FICO if calibration drifted	Consumers of Treebeard ratings (other agents, integrators, counterparties) are nascent and have not yet established route-around discipline	Demand-side accountability arrives as ratings feed insurance underwriting, on-chain enforcement, and agent-to-agent decision logic, each of which is a natural consumer once populated
A regulatory regime exists for consumer credit reporting (Fair Credit Reporting Act, CFPB oversight)	No regulatory regime for AI agent ratings	The Ent Review Panel's public dispute pathway (Section 8), methodology version history, and bug bounty program (Section 7.7) substitute for regulatory accountability in the short term

The substitutions are weaker than the originals in the short run and are designed to strengthen as the agent economy matures. *The FICO model is the right precedent. The bond ratings model from before 2008 is not. But the bridge from one to the other is not yet complete.*

What Treebeard publishes and what FICO publishes (parallel structure):

FICO publishes the categories (payment history, amounts owed, length of credit history, credit mix, new credit), the percentage weight ranges, the directional logic, the dispute pathway, and the scoring scale. FICO does not publish the exact algorithm that converts these inputs into a number. FICO is commonly treated as calibration-opaque rather than fully black-box.

Treebeard publishes the categories (Section 4), the formula shape (Section 6), the safety floor mechanism (Section 5), the directional logic by agent type, the dispute pathway (/methodology/improve), and the scoring scale. Treebeard does not publish the exact numerical calibration of weights, discount factors, or time-decay half-lives. The choice of opacity is the same. The structural protections that make the choice defensible are partially in place and partially substituted. Section 11 names the open distance.

10.4 The four-axis transparency layer

It is useful to distinguish four axes on which a rating system can be transparent or opaque, because the public discourse usually collapses them.

Axis	Public?	Rationale
Structure (categories, sources, formula shape)	Yes	Allows reproducibility from public inputs
Calibration (weights, discount factors, half-lives)	No	Prevents optimization-for-the-rating

Axis	Public?	Rationale
Decisions (current ratings, dispute outcomes, methodology decisions)	Yes	Allows accountability without negotiation
Methodology updates (changelog, versioning)	Yes	Allows historical reproduction

A note on the "Decisions" row. Treebeard publishes the current rating for every indexed agent, the outcomes of every dispute filed through the public pathway, and the rationale on every methodology decision the Ent Review Panel makes. What Treebeard does not currently publish is the per-agent score history with timestamped values across all prior versions of the methodology. Section 11 names this as an open commitment for the next major release; the data exists internally and the publication is a packaging decision rather than a calibration-opacity question.

The pre-2008 bond rating system was opaque on all four axes. FICO is opaque on calibration only. Treebeard matches FICO on Structure, Decisions (with the score-history caveat above), and Methodology updates, while keeping Calibration internal. The 2008 failure was not the failure to publish weights. It was the failure to publish structure, decisions, and updates while also operating under issuer-pays.

10.5 What this commits Treebeard to

The structural transparency Treebeard commits to in this paper is durable. Removing a category from public disclosure, hiding a signal source, or quietly changing the formula shape without versioning would constitute a methodology breach and would be visible to any reader holding the prior version of the methodology against the current one.

The calibration opacity Treebeard preserves is bounded. Every parameter is updated through the calibration loop in Section 9. Every update produces a version diff. The current values of the parameters are not public; the process by which they update is fully public, and the rate of update is bounded by the cadence specified in Section 9.4.

The combination is the rating analog of the FICO architecture, applied to autonomous agents.

Section 11. Limitations and open problems

11.1 Current coverage gaps

Treebeard indexes agents on fourteen chains as of May 2026. The coverage is incomplete. Agents operating on chains outside this set are either not yet indexed or indexed only through cross-chain reference data. The coverage gap is structural: every additional chain requires data pipeline work, signal source calibration, and ongoing monitoring.

The implication for users of the rating is that an agent absent from the directory is not necessarily an agent that does not exist. It may be an agent operating on a chain not yet in coverage, or operating off-chain in a way that does not produce signal accessible to public crawlers. The directory's silence on an agent is weak evidence about the agent's existence.

11.2 Source quality variance

The signal sources that feed the rating vary widely in quality. Some sources, such as on-chain transaction logs, are high-quality and difficult to manipulate. Other sources, such as self-attested function descriptions,

are low-quality and easy to manipulate. The methodology compensates through the source-conflict discount, but the compensation is not perfect, and the residual error in each source feeds into the composite.

The Bayesian calibration loop in Section 9 reduces this error over time as evidence accumulates. In the short term, agents that draw heavily on low-quality sources have higher rating uncertainty. The published rating includes a confidence band that reflects this uncertainty, but the confidence band itself is calibrated against historical evidence and may underestimate uncertainty on novel agent types.

11.3 Real-time behavior measurement

The rating depends substantially on signals that can be observed from outside the agent: on-chain activity, public endpoints, declared metadata. Behavior that occurs entirely inside the agent, such as decision-making logic that is not exposed through any public surface, is not directly measurable. The rating infers internal behavior from external manifestations, which is a coarser signal than direct measurement would provide.

The implication is that two agents with identical external behavior but different internal logic can receive identical ratings, even if the internal logic produces materially different counterparty risk in conditions the rating has not yet observed. The mechanism for closing this gap is the Ent Review Panel for cases where it surfaces, plus the calibration loop for systematic detection over time.

11.4 Novel agent types

The category weights $w_c(\text{type}(a))$ vary by agent type. New agent types not yet recognized in the methodology are scored against a default weight profile that may not reflect the relative importance of categories for that agent type. The result is rating noise on agents at the frontier of the agent economy.

The methodology process for adding new agent types runs through the Ent Review Panel, which identifies systematic mismatches, drafts new weight profiles, and updates the methodology through the formal versioning process. The cadence of this process lags the cadence of new agent type emergence. *Cutting-edge agents are scored conservatively until the methodology catches up.* This is acknowledged in the rating's confidence band on novel types.

11.5 The Ent Review Panel composition

The Ent Review Panel as currently constituted is the methodology owner plus AI-simulated expert panels. This is structurally smaller than the review committees the bond rating agencies maintained even before 2008. The mechanism (public dispute pathway, versioned outcomes, published rationale) is what the 2008-era system lacked, but a larger panel is materially stronger than a smaller one for several reasons: more independent review per dispute, slower drift, more domain coverage. The expansion is gated on two practical conditions: sufficient dispute volume to justify the cost of standing reviewers, and funding for independent reviewers separate from the rating system's revenue (since Treebeard takes no payment from rated entities). The current composition is acknowledged as a limitation that this paper names rather than glosses.

11.6 Per-agent score history publication

Treebeard publishes the current rating for every indexed agent and the rationale on every dispute outcome. Per-agent score history with timestamps across prior methodology versions is currently maintained internally but not published in a form designed for public consumption. This is a packaging gap, not a calibration-opacity question. The data exists. Publishing it as a public time series is an open commitment for the next major release of the methodology.

11.7 Open problems

Several questions remain open for the methodology and for the agent economy more broadly.

Cross-agent dependency rating. The methodology rates agents in isolation. Real agent commerce involves chains of agents whose outputs feed each other. A high-rated agent that depends on a low-rated

agent inherits the low-rated agent's risk in some functional sense. The methodology does not yet propagate dependency risk through the composite. Section 2.5.3 named the failure mode this leaves open.

Prediction markets on rating accuracy. A market that allows third parties to bet against published ratings would introduce skin-in-the-game accountability without compromising rater independence. The mechanism is theoretically attractive but currently unimplemented and has its own design problems (informed-trader manipulation, low-liquidity markets producing noise rather than signal).

Agents that build agents. An emerging class of agents constructs other agents at runtime. The recursive structure complicates rating: the parent agent's rating depends on the ratings of agents it has not yet built, which depend on signals that do not yet exist. The methodology has not yet been extended to handle this case.

Calibration on adversarial regime shifts. The calibration loop assumes that observed manipulation is a noisy sample from a stationary distribution. If an attacker introduces a fundamentally novel attack, the calibration loop catches it only after evidence accumulates. The lag between novel attack and methodology response is a structural vulnerability that no calibration scheme fully closes.

These problems are listed openly because the methodology does not solve them. *The honest answer to "is the methodology complete" is no.* It is the best methodology we have today, calibrated against the best evidence we have today, with a structural process for updating both as the evidence accumulates.

Section 12. References

The reference list below is selective. It covers the primary sources used in the construction of the methodology. A complete bibliography is maintained at /methodology/references on the Treebeard website, where additions are appended on the same cadence as methodology updates.

Bond ratings industry, 2008 collapse, and regulatory response

Financial Crisis Inquiry Commission. *The Financial Crisis Inquiry Report: Final Report of the National Commission on the Causes of the Financial and Economic Crisis in the United States.* Government Printing Office, 2011. https://www.govinfo.gov/app/details/GPO-FCIC

Dodd-Frank Wall Street Reform and Consumer Protection Act, Pub. L. 111-203, H.R. 4173, 2010. Title IX, Subtitle C, Improvements to the Regulation of Credit Rating Agencies. https://www.congress.gov/bill/111th-congress/house-bill/4173

U.S. Securities and Exchange Commission. *Annual Report on Nationally Recognized Statistical Rating Organizations.* Most recent edition. https://www.sec.gov/ocr

Coffee, John C., Jr. "Ratings Reform: The Good, The Bad, and The Ugly." *Harvard Business Law Review* 1, no. 1 (2011): 231–278.

Partnoy, Frank. "How and Why Credit Rating Agencies Are Not Like Other Gatekeepers." In *Financial Gatekeepers: Can They Protect Investors?*, edited by Yasuyuki Fuchita and Robert E. Litan. Brookings Institution Press, 2006.

Hill, Claire A. "Why Did Anyone Listen to the Rating Agencies After Enron?" *Journal of Business and Technology Law* 4 (2009): 283.

White, Lawrence J. "The Credit Rating Industry: An Industrial Organization Analysis." *NYU Stern Working Paper*, 2002.

Consumer credit scoring and the FICO architecture

Fair Isaac Corporation. *FICO Score Methodology Disclosures.* Current edition. https://www.fico.com/en/products/fico-score

Fair Isaac Corporation. *Understanding FICO Scores*. Educational publications. https://www.myfico.com/credit-education/

Fair Credit Reporting Act, 15 U.S.C. § 1681 et seq., 1970, as amended. https://www.consumer.ftc.gov/articles/fair-credit-reporting-act

Consumer Financial Protection Bureau. *Annual Report of Credit and Consumer Reporting Complaints*. Most recent edition. https://www.consumerfinance.gov/data-research/research-reports/

Agent identity, reputation, and commerce protocols

De Rossi, Marco (MetaMask), Davide Crapis (Ethereum Foundation), Jordan Ellis (Google), and Erik Reppel (Coinbase). *ERC-8004: Trustless Agents*. Ethereum Improvement Proposal, created August 13, 2025. Deployed on Base mainnet January 2026. https://eips.ethereum.org/EIPS/eip-8004

ERC-8183: Agent Commerce Reputation Linkage. Ethereum Improvement Proposal, draft March 2026. Co-developed by Virtuals Protocol and the Ethereum Foundation. https://eips.ethereum.org/EIPS/eip-8183

x402 Foundation. *x402 Protocol Specification*. Hosted under the Linux Foundation, current draft. https://www.linuxfoundation.org/x402foundation

Coinbase. "x402: An HTTP-native payment protocol for AI agents and APIs." Engineering blog, 2026. https://www.coinbase.com/developer-platform

Google. *Agent Payments Protocol (AP2)*. Specification, 2025. https://github.com/google-agentic-commerce

Internet Engineering Task Force. *RFC 7231: Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content*. June 2014. Section 6.5.2 specifies HTTP status code 402 (Payment Required), the substrate for x402. https://datatracker.ietf.org/doc/html/rfc7231

Adjacent rating providers and ecosystem maps

Intuition. "AgentRank: An Open-Source Decentralized Ranker for AI Agents." Project documentation and source, 2026. https://github.com/0xIntuition

ZARQ. Public methodology page. Most recent version. https://zarq.io

RNWY. Public methodology page. Most recent version. https://rnwy.live

8004scan. ERC-8004 explorer. https://8004scan.com

AgentProof. Public certification overview. Most recent version.

Helixa. Public methodology overview. Most recent version.

Goodhart's Law and the metric-as-target literature

Goodhart, Charles. "Problems of Monetary Management: The U.K. Experience." In *Papers in Monetary Economics*, Reserve Bank of Australia, 1975. Reprinted in *Goodhart's Law: Origins, Meaning, and Implications for Monetary Policy*, Bank of England, 2018.

Strathern, Marilyn. "Improving Ratings: Audit in the British University System." *European Review* 5, no. 3 (1997): 305–321. The widely-quoted reformulation: "When a measure becomes a target, it ceases to be a good measure."

Manheim, David, and Scott Garrabrant. "Categorizing Variants of Goodhart's Law." Working paper, Machine Intelligence Research Institute, 2018.

Bayesian methods

Gelman, Andrew, John Carlin, Hal Stern, David Dunson, Aki Vehtari, and Donald Rubin. *Bayesian Data Analysis*. Third Edition. Chapman and Hall/CRC, 2013.

McElreath, Richard. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Second Edition. Chapman and Hall/CRC, 2020.

Agent autonomy and capability

Karpathy, Andrej. "Software 2.0." *Medium*, November 2017. https://karpathy.medium.com/software-2-0-a64152b37c35

Russell, Stuart. *Human Compatible: Artificial Intelligence and the Problem of Control*. Viking, 2019.

Wu, Qingyun, Gagan Bansal, Jieyu Zhang, et al. "AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversations." Microsoft Research, 2023.

Anthropic. *Claude Tool Use and Agentic Workflows*. Documentation, 2024–2026. https://docs.anthropic.com

Cross-domain non-substitutable floor mechanisms

A.M. Best. *Best's Credit Rating Methodology: Insurance Holding Company and Debt Ratings*. Current edition. https://www.ambest.com/ratings/methodology.asp

Federal Aviation Administration. *Federal Aviation Regulations Part 121: Operating Requirements*. Current edition. https://www.ecfr.gov/current/title-14/chapter-I/subchapter-G/part-121

U.S. Food and Drug Administration. *New Drug Application Review Process*. Current edition. https://www.fda.gov/drugs/types-applications/new-drug-application-nda

U.S. Nuclear Regulatory Commission. *10 CFR Part 50: Domestic Licensing of Production and Utilization Facilities*. Current edition. https://www.nrc.gov/reading-rm/doc-collections/cfr/part050/

Treebeard methodology pages

Treebeard. "Methodology Overview." https://treebeardai.com/methodology

Treebeard. "Independence." https://treebeardai.com/independence

Treebeard. "Trust and Governance." https://treebeardai.com/trust

Treebeard. "Improvement and Disputes." https://treebeardai.com/methodology/improve

Treebeard. "The 2008 Question Every Agent Rater Has to Answer." Blog, May 2026. https://treebeardai.com/blog/the-2008-question

---

Appendix. Signal coverage matrix

The matrix below maps the seven signal categories to the standards, protocols, and registries that contribute observable signal. Each cell indicates whether the standard contributes to the corresponding category and the strength of contribution (Primary, Secondary, or Indirect).

Category	ERC-8004	ERC-8183	x402	Smart Contract Logs	Endpoint Probes	GitHub Repository	Audit Registries	Bug Bounty Platforms	Independent Citations	Discovery Layers	Cross-chain Bridges
Economic Viability	Secondary	Primary	Primary	Primary	Indirect	Indirect	Indirect	Indirect	Indirect	Secondary	Secondary
Operational Reliability	Indirect	Secondary	Secondary	Primary	Primary	Indirect	Indirect	Indirect	Indirect	Indirect	Indirect

Category	ERC-8004	ERC-8183	x402	Smart Contract Logs	Endpoint Probes	GitHub Repository	Audit Registries	Bug Bounty Platforms	Independent Citations	Discovery Layers	Cross-chain Bridges
Code Quality	Indirect	Indirect	Indirect	Secondary	Indirect	Primary	Primary	Secondary	Secondary	Indirect	Indirect
Autonomy Index	Primary	Secondary	Indirect	Primary	Primary	Secondary	Indirect	Indirect	Secondary	Indirect	Indirect
Safety	Primary	Secondary	Indirect	Primary	Primary	Secondary	Primary	Primary	Secondary	Indirect	Indirect
Community and Ecosystem	Primary	Primary	Secondary	Indirect	Indirect	Secondary	Secondary	Secondary	Primary	Primary	Secondary
Security Posture	Secondary	Indirect	Indirect	Primary	Primary	Primary	Primary	Primary	Secondary	Indirect	Secondary

Reading the matrix. A Primary contribution means that the standard is one of the highest-weight sources in its column for the corresponding category. A Secondary contribution means that the standard provides material signal but does not drive the category score. An Indirect contribution means that the standard contributes only through its effect on other sources or through coverage-coupled signals.

Standards covered. ERC-8004 is the agent identity, reputation, and validation registry. ERC-8183 is the agent commerce reputation linkage. x402 is the agent payment protocol. Smart contract logs are direct observations of agent activity on indexed chains. Endpoint probes are independent measurements run by Treebeard infrastructure. GitHub repositories are public source code, commit history, and dependency manifests. Audit registries include published audit reports from third-party security firms. Bug bounty platforms include scope, payout history, and disclosure timelines. Independent citations are references in research, news, or third-party reports. Discovery layers are public ecosystem maps maintained by entities without rating revenue. Cross-chain bridges provide signal on agents operating across multiple chains.

Adjustments by agent type. The matrix above gives the default contribution profile. The category weights $w_c(\text{type}(a))$ in Section 6 then determine the relative importance of each row in the composite for a given agent type. A trading agent's composite weights Economic Viability and Operational Reliability higher than the default, which amplifies the contribution of x402, smart contract logs, and endpoint probes for that agent. A customer-service agent's composite weights Community and Ecosystem and Autonomy Index higher, which amplifies ERC-8004 and citation-based sources.

Coverage gaps. Several columns in the matrix represent signal sources that are partially covered or unevenly populated. Bug bounty platform data is uneven across agents: some agents publish their bug bounty terms publicly, others maintain private programs. Cross-chain bridge data is sparse for agents operating on a single chain. The methodology accommodates these gaps through the source-conflict discount, which down-weights sources whose coverage is structurally incomplete for the agent under evaluation.

The matrix is updated alongside the methodology versioning process. New standards added to the rating produce new columns. Material reweighting of an existing column (a standard whose contribution shifts from Primary to Secondary in some category) is logged in the methodology changelog.