When AI Services Fail: Computational Overcharging and the Cost of Failed Prompts

Full evidence-integrated research manuscript | Version 1.0.3 — final edition | 29 October 2026 (v1.0.1: §5 procedure numbering and reference-model test labels; v1.0.2: editorial — duplicate §5 preamble removed, stray heading number, duplicate source URL, current ERL PR head; v1.0.3: Case C-2 locator updated to the merged ERL record; post-freeze Kimi history-depth documentation conflict added to §2.3)

Technical research and consumer-accountability analysis. Not a legal opinion, finding of unlawful conduct, or measurement of provider backend billing.

Abstract

Computational overcharging describes the consumption or retention of finite computational entitlements beyond what delivered performance justifies under an applicable failure-accounting standard. This research does not establish a matched excess debit for any individual illustrated execution. Instead, it documents differences in published failure-treatment rules, fidelity of customer-visible failure states, and granularity of usage telemetry. It examines three distinct burdens shifted to customers: detecting nonconformance, obtaining a remedy, and repeating or continuing interrupted work. One provider publicly describes automatic protection for certain failures and displays refund notices, but its backend ledger adjustment was not independently observed. The paper proposes verifiable per-execution receipts, accurate delivery and failure classifications, automatic protection of eligible entitlements, privacy safeguards, and accessible dispute processes. [E1–E7; R1–R3]

1. Research question, scope and evidence rules

The paper concerns paid consumer AI task services, particularly agent-style products where success or failure can be observed against a requested deliverable. It does not assume that every incorrect answer is objectively detectable or that every disappointing result deserves a refund.

- Can providers instrument and reliably identify categories of unsuccessful execution?
- What credits, usage units, or other charges were consumed when a task did not deliver the contracted or represented result?
- What remediation policy would be consistent, contestable, privacy-preserving, and resistant to abuse?
- What evidence would be needed to evaluate representations, injury, causation, and possible consumer-protection claims?

Evidence labels used throughout: OBSERVED = directly inspected public text, source code, or provided user interface; REPORTED = attributed account not independently reproduced; DERIVED = interpretation grounded in identified evidence; PROPOSED = engineering or policy design; UNKNOWN = unmeasured or unverified. Published policy is not proof of implementation; a screenshot hash establishes byte identity, not backend accounting accuracy. [E1–E7]

Frozen research discussion and corrections: https://github.com/SV-LLM/Ecosystem_Chat/issues/9 ; Claude manuscript review: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6430189444 ; Kimi manuscript review: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6430920989

2. Primary findings: publication, presentation and telemetry

2.1 Published failure-treatment rules

OBSERVED: Kimi public materials include automatic non-deduction/refund language for some failure classes and separate request-and-verification language elsewhere. These are not interchangeable promises; eligibility depends on product, plan and terms version. The bounded Anthropic pages reviewed did not identify an equivalent published failure-restoration rule; that is not proof that Anthropic never restores credits. OpenAI subscription cash-refund guidance addresses a different transaction and was not part of the three-model failed-execution comparison. [E1, E2, R1, R2]

Kimi primary sources: https://www.kimi.com/en/help/agent/agent-quota-and-billing ; https://www.kimi.com/en/help/membership/membership-update-rules ; Kimi pricing/plan geometry: https://www.kimi.com/en/help/membership/membership-pricing . Frozen source and reviewer interpretation: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6428435857 and https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6429453316

Provider / surface	What the inspected evidence supports	What it does not establish
Kimi Agent / Deep Research	Some official pages describe automatic protection; other official pages describe request/verification; visible refund message in Case A	Specific backend debit/reversal, universal plan coverage
Anthropic / Claude	No explicit failed-execution entitlement-restoration rule identified on the bounded pages reviewed	Provider-wide absence of remediation or unobserved internal behavior
OpenAI / ChatGPT	Subscription purchase refund help page exists; Case B shows artifact-delivery failure	That cash refund policy governs failed task entitlement treatment

2.2 Presentation fidelity

OBSERVED SOURCE BEHAVIOR: A reviewed Kimi Code/ACP adapter path maps a non-authentication failed outcome to an end_turn presentation. This demonstrates a failure-state representation risk, not a measured entitlement debit, and must not be generalized to Kimi Agent accounting. [E3]

Source/revision and analysis locator: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6428077384 ;

research reconciliation:
https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6428435857

2.3 Telemetry granularity

OBSERVED PRIVATE EXHIBITS: Customer-provided Kimi usage-interface screenshots display named task percentages and text referring to a last-1,000-record interface. Neither a displayed 0% nor a history-depth label establishes an authoritative backend ledger, precision, or complete history. REPORTED: a Claude Code /usage missing-bars account is an external report; its mechanism was not independently reproduced or source-verified and is not equivalent to the adapter finding. [E4, E5]

Public evidence-class reconciliation:
https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6429477644 ;
private exhibit locator:
https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6428143199

3. Customer burdens and computational-overcharging theory

The analysis separates (1) resource execution, (2) deliverable production and verification, (3) failure attribution, (4) entitlement deduction, and (5) restoration or appeal. These events may occur at different times or not at all. Consumption alone does not prove computational overcharging: a finding requires an applicable standard and a justified-versus-actual entitlement comparison. [E6–E7]

Detection burden: the customer discovers silent nonconformance. Remedy burden: the customer must locate, document, and contest a deduction or missing service. Continuation burden: the customer must retry, reconstruct context, or wait for restored capacity. The in-issue record credits the customer with detecting two of four identified review errors; this is qualitative process evidence, not a population estimate. [E6–E7]

Direct monetary injury may include non-restored purchased credits or additional payments made solely to repeat a provider-attributable failed task. Subscription time or rate-limited capacity requires a separately justified valuation model; not every unsuccessful task equals a full subscription refund. Consumer time can be documented but should not be converted automatically into recoverable damages. Avoid counting both lost credits and their replacement purchase as separate losses when they represent the same economic cost.

Illustrative sensitivity analysis only: 100,000 affected paid tasks × $0.50 eligible unremedied cost = $50,000; 1 million × $2 = $2 million. These figures are arithmetic scenarios, not observed prevalence, revenue, damages, or liability estimates.

The commercial comparability of metered entitlements

Both subscriptions and credit-based access can grant a finite, renewable entitlement to computational service. Their customer-facing labels do not determine whether provider-attributable failed execution consumes scarce capacity. Comparison should be normalized to the entitlement, the metering event, the reset rule, and the remedy—not to the marketing name of a unit.

A subscription fee is not itself a per-task price. Where the allowance is a rate limit, remediation may mean restoring the ability to retry, excluding a failed request from the

limit, or extending the relevant window. Where consumption is metered in purchased units, reversal can be directly measured. Where a tier uses opaque dynamic limits, the provider must disclose enough accounting evidence to support a meaningful remedy without exposing proprietary infrastructure.

The screenshot supplied by a customer records Kimi reporting a failed agent task and refunded quota. This is direct evidence of a displayed remediation outcome, not proof of whether its internal system avoided charging, reversed a debit, or granted compensatory capacity. The paper should not claim that ChatGPT never restores usage; the public record reviewed here does not establish its comprehensive treatment of every failure category.

Engineering duty before deployment

In safety-critical construction, required performance is specified before a design is approved; inadequate materials are not made adequate by a later apology. AI services are not governed by identical building codes, but the engineering analogy illustrates a useful design principle: foreseeable operational failure modes should be investigated, instrumented, tested, and addressed before a service is monetized under specific performance representations.

A competent software process includes failure-mode analysis, acceptance criteria, negative testing, accounting reconciliation, escalation paths, and rollback or compensation mechanisms. Where an objectively identifiable failure category exists, an engineering team should either demonstrate an effective control, narrow the product claim, or disclose the residual limitation. The existence of an alternative provider implementation is evidence of technical feasibility for that category, not proof of universal feasibility or an identical legal obligation.

No one expects error-free AI: define the compensable event

A false answer, unsatisfying style, or disagreement over quality is not automatically a compensable failure. The stronger claim concerns objective provider-side failures: an agent crashes, a tool cannot execute, a requested deliverable is never produced, a system acknowledges execution failure, or a provider validation gate records a non-success disposition. The proposed standard starts with those observable events, then expands only as classification accuracy is demonstrated.

The acceptance criterion must be established from the original request and product representations—not invented retroactively by either the customer or the model. A partial deliverable may warrant partial restoration. A customer cancellation, safety refusal, unsupported request, or external provider outage requires a separate attribution and policy decision rather than automatic assignment of fault.

4. Detailed customer-visible cases

Scope note. These case studies document user-visible observations in the conversation, not provider telemetry. Neither case establishes the amount of computational entitlement

consumed. Screenshots are observational evidence, not proof of underlying billing or causation.

1 Case A — Kimi agent execution failure with reported restoration

Evidence: User-provided screenshot IMG_4068.png (captured 26 October 2026 conversation); original retained as source exhibit A.

Observed output: “Kimi ran into an issue. Your credits have been refunded. Start a new chat and try again.” The interface also states that agent quota has been refunded.

Task context: The visible task references Data-Continuation/StegClaw issue #28 and an action-plan review. The screen does not expose full execution logs.

Supported finding: The customer-facing interface expressly reported failure and credit/quota restoration.

Not established: Actual ledger adjustment, precise amount, whether credits were deducted then restored or never collected, extent of completed work, or underlying trigger.

Independent verification needed: Provider transaction record; before/after allowance snapshots; failure code; timestamped task and execution IDs; evidence of whether the quota was available for reuse.

2 Case B — ChatGPT research-paper PDF delivery failure

Evidence: User-provided screenshot C1AFD2B5-AF68-4B20-BEDB-C18A1126338E.png; original retained as source exhibit B.

Observed output: The ChatGPT iOS document viewer displayed “The file download failed. Please try again later.” for ai_failure_remediation_paper.pdf.

Task context: The failed download followed creation and sharing of a research paper. A later assistant response reported generating alternate links and validating file integrity.

Supported finding: A customer-visible artifact delivery attempt failed. File existence, even if verified server-side, does not itself establish successful end-user delivery.

Not established: Cause of delivery failure, any deducted allowance, number of retries counted toward limits, monetary loss, or whether the failure was persistent.

Independent verification needed: File delivery request and error logs; object availability; client/network status; usage counters; successful subsequent download confirmation.

3 Comparative interpretation

Case A visibly reports both a failed task and quota remediation. Case B visibly reports a failed delivery, with no corresponding remediation disposition shown. These observations demonstrate different customer-facing outcomes, but do not establish equivalent underlying accounting events. The comparison motivates provider-side measurement rather than proving systematic non-remediation.

Case C — self-recorded ChatGPT nonconformance series

REPORTED RESEARCH CANDIDATE: ERL machine-assessment records are referenced as a self-recorded failure series. The record is not an independently verified provider accounting audit. Locator only: https://github.com/StegVerse-Labs/Executive_Rhetoric_Ledger ; issue review: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6430189444 . Publication of underlying restricted records requires separate authorization.

Case C-2 — Claude self-documented nonconformance series

REPORTED / RECORD MERGED / PUBLICATION NOT AUTHORIZED: at least fifteen instances are recorded in ERL PR #247, merged 29 October 2026 (merge commit e6b2193d). Locator only: https://github.com/StegVerse-Labs/Executive_Rhetoric_Ledger/pull/247 . No restricted case text is reproduced; the merge records the instances in the ledger and does not authorize publication of restricted case text. This is evidence of recorded complaints and continuation work, not verified entitlement depletion.

Cross-case interpretation

Case A shows a displayed failure-and-refund message; Case B shows failed end-user artifact delivery; C and C-2 record nonconformance and correction burdens under different custody restrictions. They are not matched billing transactions and cannot support a comparative excess-debit rate.

5. Engineering analysis and proposed accounting standard

Proposed verifiable failure-remediation standard

1 Required event record

Each paid task should have a privacy-minimized receipt containing a pseudonymous task identifier, service tier and applicable terms version, user-requested deliverable or acceptance conditions, start/end timestamps, execution outcome, tool failures, consumed billing units, validation status, attribution confidence, refund decision, and appeal reference. Sensitive prompt contents need not be publicly disclosed; an authorized audit path may use hashes and restricted access.

The canonical proposed event taxonomy uses four orthogonal axes: DELIVERY {DELIVERED, NOT_DELIVERED, PARTIAL, DISPUTED, UNKNOWN}; SIDE_EFFECTS {NONE, PARTIAL, COMPLETED, UNKNOWN}; ATTRIBUTION {PROVIDER, CUSTOMER, EXTERNAL, MIXED, UNKNOWN}; CHARGE_TREATMENT {NOT_COLLECTED, CHARGED, RESTORED, PARTIALLY_RESTORED, DISPUTED, UNKNOWN}. These are proposed engineering values, not observed provider-native schemas. Burden fields independently capture detection, remedy and continuation work.

An attempt transitions REQUESTED → EXECUTING → OBSERVED_OUTCOME → ATTRIBUTION → CHARGE_RECONCILIATION → REMEDY_OR_APPEAL. Outcome dimensions are recorded independently; success must not be inferred from a terminal transport marker alone. Eligibility is a published plan-specific rule, not an assertion that all failed or disappointing tasks require refund.

A minimum implementation can operate as REQUESTED → EXECUTING → OUTCOME_RECORDED → ATTRIBUTION_REVIEWED → CHARGE_RECONCILED → REMEDIATION_APPLIED or APPEAL_AVAILABLE. A disposition must be immutable or versioned, and a revised decision must preserve the prior reason and the new evidence. Automatic remediation should never depend solely on the model's unverified statement that it succeeded or failed.

Illustrative rule: if a task is provider-attributable, the requested deliverable is absent, and paid units were consumed, issue an automatic restoration for eligible units; otherwise return a reason-coded decision and appeal path. Edge cases include mixed-success tasks, subscription capacity rather than credits, adversarial failure reports, external-service outages, and disputed acceptance criteria.

Inputs: task ID, entitlement ledger reference, terms version, original acceptance criteria, start and terminal states, error codes, validation evidence, units charged or capacity consumed, and applicable reset window. The event record should be signed or tamper-evident and should avoid retaining sensitive prompts when hashed criteria and authorized audit access suffice.

1. Reserve or meter entitlement at task start; preserve the pre-task balance or capacity state.
2. The remediation operation is keyed by attempt ID, original charge ID, policy version, and effective entitlement window. Apply no more than one effective adjustment per eligible charge; preserve audit records for duplicate, replay, partial side-effect and wrong-window attempts. Effective restored access matters more than a cosmetic display adjustment.
3. Return a receipt showing original debit, adjustment, remaining entitlement, decision reason, and an appeal path.
4. For uncertain cases, preserve evidence, avoid claiming success, and provide timely review; never silently discard contradictory observations.

Idempotency and abuse resistance are essential: restoration is keyed to task and charge identifiers; appeals cannot trigger duplicate grants; fraud controls must not erase genuine provider failures. A customer-visible percentage meter can coexist with more precise internal units, but the restoration should be verifiable in the customer's effective access.

Event identity

Assign an immutable attempt identifier, product, subscription entitlement class, policy version, and timestamps. Preserve privacy by default; never require prompt contents for routine accounting.

Entitlement snapshots

Capture pre- and post-attempt remaining allowance or provider-signed quota state, accounting window, and reset boundary. Record uncertainty if the provider cannot expose comparable units.

Outcome evidence

Distinguish transport/system error, tool invocation failure, validated contract failure, user cancellation, successful result, and subjective quality dispute. A failed download can be a delivery failure even when the artifact was generated.

Attribution and review

Apply deterministic eligibility rules to objective events; route ambiguous cases to review with a deadline and a reasoned decision. Avoid assuming that any dissatisfied response is a compensable failure.

Remediation ledger

Issue idempotent, append-only debit-reversal or compensating credit events linked to the original attempt. Reconcile across session and rolling-window limits, not merely displayed credits.

User receipt

Provide original deduction, corrected deduction, effective allowance restored, classification, reason, review path, and timestamp. Expose machine-readable receipts where practical.

- Provider-side execution failure before a usable result: no final charge or exactly one compensating credit.
- Successful completion: charge according to disclosed metering policy; no automatic failure credit.
- Duplicate failure event: at most one effective restoration.
- Failure immediately before reset: restoration respects the disclosed entitlement window; no silent expiry that nullifies the remedy.
- Generated artifact but broken download: record DELIVERY_FAILED independently of GENERATION_SUCCEEDED.
- Customer cancellation and disputed subjective quality: preserve evidence and apply published rules; do not automatically assume provider fault.
- Restored displayed credit but active rolling limit prevents use: mark incomplete until effective access is restored or equivalent remedy supplied.

Absence of pre-declared acceptance criteria never removes access to the dispute path; remedy eligibility is not conditioned on customer pre-declaration. Objective-linked acceptance is an optional verification technique for bounded tasks, not a universal rule for subjective outputs.

Proposed automatic computational credit-protection standard

- Publish product- and plan-specific entitlement geometry, including resets, rolling windows, failed-run treatment, and any appeal path.
- Issue durable, privacy-minimized, exportable per-execution receipts with request ID, terms version, disposition, usage units, entitlement balance before/after where available, and remediation identity.
- Distinguish execution stopped, artifact produced, criteria verified, and disputed; never silently translate a known failed execution into a success-equivalent user outcome.
- For objectively verifiable provider-attributable failures, automatically prevent eligible deductions or restore eligible finite entitlement units, with idempotent reconciliation and reason-coded exceptions.
- Preserve partial side effects and legitimate completed work; do not award unlimited credits for subjective dissatisfaction or user-caused errors.

- Provide customer-visible correction and appeal, independently auditable decision records, privacy protections, and negative tests for replay, duplicate refund, incorrect attribution, and inaccessible accounting.
- Report the burden of detecting failure, obtaining remedy, and repeating work separately; mark quantities unknown when unsupported.

6. Methods, falsifiability and independent verification

Evidence needed for credible empirical findings

A public research register should avoid asserting a population-wide problem from a small collection of anecdotes. It should gather cases prospectively with consistent inclusion criteria and retain contradictory cases, including successful remediation.

- Preserve the exact product, plan, purchase terms, and contemporaneous refund policy.
- Capture a timestamped task request, observable completion criteria, final response, and any tool or validation errors.
- Record before/after credit balances, rate-limit effects, or subscription access consequences; distinguish opportunity cost from direct paid loss.
- Record customer-support communications, appeals, outcomes, and any actual refund.
- Classify independently where possible, with two reviewers and a disagreement field; redact personal and proprietary material.
- Publish the denominator: sampled tasks, failures, eligible failures, charges, restored units, and unresolved cases.
- Controlled tests: submit equivalent bounded tasks on paid plans; record initial entitlement, explicit system failures, final output, and post-task capacity.
- Negative controls: successful tasks, user cancellations, intentionally unsupported tasks, and duplicate refund requests.
- Independent adjudication: two reviewers classify observable evidence without knowing the provider; retain disagreements and uncertainty.
- Metrics: eligible-failure rate, restoration rate, time-to-restoration, false-restoration rate, denied-appeal overturn rate, and uncompensated effective capacity.
- Disclosure: sampling method, terms versions, time windows, task difficulty, provider outages, missing measurements, and all exclusions.

The original draft labels three paper-local hypotheses H1–H3; these are not the frozen issue-register H1–H6. In the canonical analysis, documented public protection language supports feasibility, while provider-wide restoration rates and excess-debit prevalence remain unmeasured. No numeric hypothesis is promoted to an observed result.

Empirical protocol and measurable harm

Study design: preregister controlled tests across comparable paid tiers and named task types; record plan, date, prompt category, observable task result, error UI, usage display before/after, support outcome, and effective ability to initiate a subsequent task. Test only within platform terms and avoid deliberately inducing harmful load. Include successes, failures, and ambiguous outcomes to avoid selection bias.

Primary endpoint: proportion of objectively failed, provider-attributable attempts that cause an unreversed reduction in effective computational entitlement. Secondary endpoints: time to restoration; fraction requiring manual complaint; appeal success; percentage of failures with auditable receipts. Publish denominators and confidence intervals; never extrapolate from screenshots to population prevalence.

Harm accounting must distinguish (a) lost usable allowance, (b) separately purchased overage or upgrades, (c) documented substitute-service expense, and (d) user time, which should not automatically be assigned a legal damages value. A notional allocation such as monthly price divided by allowance is a research sensitivity estimate, not necessarily recoverable damages.

Pre-registration

Publish the task definition, provider, subscription tier, date/time, device, region, quota-window boundary, and expected observable result before execution. Separate service execution from artifact retrieval.

Baseline

Record a redacted screenshot of quota availability and the timestamp. If the UI exposes only a percentage, record that percentage and explicitly state the precision limit.

Execution

Submit a consented, non-abusive, repeatable task. Capture the task ID, user-visible error, provider message, artifact state, and timing. Do not intentionally exhaust quotas or trigger harmful behavior.

Follow-up

Capture the post-event quota and the next reset boundary. Repeat after a short interval and, where practical, verify the ability to execute an equivalent task. Do not infer internal billing from a rounded display.

Attribution

For each attempt, classify delivery, side effects, attribution and charge treatment separately; retain UNKNOWN where measurements are missing. Historical single-label lists in previous manuscript versions are non-normative and are not reused here.

Reproducibility

Retain hashed evidence files, software versions, clock offsets, redacted task transcripts, failure codes, and observer notes. Permit an independent reviewer to reproduce the classification without access to private account data.

Illustrative executable reference model

Executable reference model and observed test results

The companion Python 3 standard-library file failure_accounting_reference_v05.py defines an illustrative, deterministic restoration ledger. Eligibility in this example requires FAILED outcome, PROVIDER attribution, supporting evidence, and a nonnegative charged amount.

Event identity makes repeated processing idempotent. It is a policy demonstration, not a production-grade financial system or an integration with either provider.

Execution observed locally on 26 October 2026: 8 tests executed; 8 passed; 0 failures. Tests: provider failure restores charged units; success does not; customer error does not; unknown attribution does not; missing evidence does not; duplicate event does not double restore; zero charge restores zero; negative charge raises an error.

Known limitations: the model does not implement authentication, audit signatures, concurrent atomic writes, appeals, quota reset boundaries, partial-task allocation, effective-access verification, or provider telemetry. Those require independent design and testing before any operational deployment.

The eight-of-eight reference-model unit tests were re-run by the manuscript author in the manuscript-finalization environment; neither that run nor the earlier author runs is a reviewer-independent reproduction. They demonstrate only the bounded example logic, not production readiness, provider conformance, or authenticated ledger restoration.

7. Consumer-protection context and limits

Existing consumer-protection precedent

The FTC's 2023 Epic Games order required $245 million for consumer refunds over allegedly deceptive purchase design and unauthorized charges; a separate $275 million settlement concerned children's privacy. These are different allegations from AI execution failure, but demonstrate that digital-product design and billing practices can result in restitution-oriented remedies [4]. In June 2025, the FTC reported nearly $200 million in Fortnite refunds issued across its distributions [5].

The FTC's unfairness framework examines substantial consumer injury, reasonable avoidability, and countervailing benefits. Monetary losses are particularly relevant; small losses may matter at scale. Application to a particular AI provider would require evidence of the actual practice, injury, causation, consumer expectations, and defenses—not merely the existence of a competitor's refund policy [6,7].

Legal questions for independent counsel

Potential analyses may consider FTC Act Section 5, applicable state unfair/deceptive acts or practices laws, representations and warranties, contract terms, arbitration/class-action provisions, standing, ascertainable loss, causation, statutes of limitation, and remedies. The FTC Act generally does not provide an ordinary private right of action; state-law or contractual routes must be evaluated separately. Regulatory complaints and private litigation are different processes.

An internal ability to classify failures would be evidence of feasibility or knowledge. Willful blindness, intentional omission, deceptive representation, and statutory unfairness are distinct theories requiring different proof. A paper intended for public circulation should state each as a research question unless primary evidence supports a stronger factual assertion.

Consumer protection and legal-review boundaries

The FTC unfairness framework examines substantial consumer injury, reasonable avoidability, and countervailing benefits. The Epic Games refund order shows enforceable remedies for certain unlawful digital billing practices; it does not decide liability for AI failures. A legal review should separately analyze representations, actual deduction records, contractual terms, statutory standing, injury, causation, arbitration/class waivers, jurisdiction, and remedy. No class action or intentional misconduct is established by this manuscript.

Knowledge must be assessed with evidence: an existing company refund mechanism can establish knowledge of that mechanism, but proof of willful blindness or knowing non-remediation requires records connecting decision-makers, failure data, affected entitlements, and policy choices. The paper proposes preservation and examination of those records rather than presuming their contents.

FTC primary source (Epic final order): https://www.ftc.gov/legal-library/browse/cases-proceedings/1923203-epic-games-matter ; 2025 distribution (> $126 million in that round; nearly $200 million cumulative): https://search.ftc.gov/news-events/news/press-releases/2025/06/ftc-sends-126-million-refunds-fortnite-players-who-were-charged-unwanted-items-reopens-claims ; FTC unfairness: https://www.ftc.gov/legal-library/browse/ftc-policy-statement-unfairness . These precedents do not decide AI failure-accounting liability.

8. Practical accountability, governance and responsible publication

Practical implementation and public accountability

- Publish an accessible task-failure and credit-remediation policy with concrete examples.
- Provide users with a portable task outcome and billing receipt, including failure reasons.
- Restore eligible charges by default for objective provider-attributable failures, with fraud controls and appeals.
- Audit false positives and false negatives in failure classification; publish aggregate correction rates.
- Allow independent assessment of provider representations, usage records, and remedy performance while protecting confidential data.

Proposed consumer-facing standard and policy language

Minimum standard: When a provider can verify that its own execution system failed to complete an eligible paid task, it should not permanently consume the corresponding limited entitlement without a disclosed, proportionate remedy. The customer should receive an auditable disposition and a usable appeal path. This is a proposed industry standard; its legal enforceability is jurisdiction- and fact-dependent.

Illustrative policy: “If our system records an eligible execution failure, we will automatically preserve or restore the affected usage allowance. We will show the adjustment and explain any exclusion. Where restoration is technically unavailable, we will provide an equivalent

remedy or a review process. We do not promise error-free answers or guarantee subjective task satisfaction.”

Public evidence intake and responsible publication

A public case register should collect provider, plan, jurisdiction (broad), date, policy snapshot, task acceptance criteria, system-reported outcome, before/after allowance, restoration outcome, appeal outcome, and consent to publish. Redact credentials, personal content, account identifiers, and third-party data. Never publish private conversation contents or proprietary code without permission.

For attorneys and regulators, retain original screenshots, receipts, support responses, policy versions, and an evidence chain. For journalists, publish aggregate counts and representative consented cases with clear caveats. Avoid calling non-restored capacity “stolen money” without a defensible valuation and causal record.

Research integrity, privacy, and publication gates

Case exhibits must be reviewed for personal data, account identifiers, private repository content, and sensitive task details before public distribution. The screenshots should remain private evidence until redacted copies are approved. No observed case here proves a particular provider charged for a failed task, violated a statute, or caused a quantified financial loss.

Legal interpretation remains reserved for independent counsel. No Kimi-authored review language is adopted as authority for legal conclusions. Public mirror and research Site publication are separate publication actions, not evidence that restricted materials were approved for release.

Independent source recheck and engineering reproduction (26 October 2026)

RECHECKED — Kimi membership-update rules explicitly distinguish new Plus-and-higher plans (rolling five-hour limit, no weekly limit) from legacy Andante-and-higher plans (rolling five-hour and weekly limits). https://www.kimi.com/en/help/membership/membership-update-rules

RECHECKED — Kimi Code membership documentation independently confirms that new plans remove the weekly quota while retaining the rolling five-hour window; legacy plans retain both. https://www.kimi.com/code/docs/en/kimi-code/membership.html

RECHECKED / DOCUMENTATION CONFLICT — Kimi membership pricing and Agent credit pages continue to describe a generic five-hours-per-week Kimi Code limit. The manuscript does not infer which plan an individual account has; plan-specific console records are required. https://www.kimi.com/en/help/membership/membership-pricing ; https://www.kimi.com/en/help/agent/agent-quota-and-billing

RECHECKED — Kimi pricing describes up to 1,000 recent usage records and possible display delays. That is an interface claim, not a backend ledger audit. https://www.kimi.com/en/help/membership/membership-pricing

POST-FREEZE ADDENDUM / DOCUMENTATION CONFLICT — Kimi's Agent quota-and-billing help page says customers can view "the 10 most recent usage records", while the membership pricing page says "your most recent 1,000 usage records". Two official pages state different history depths for the same customer view. Reported by Kimi on 27 October 2026 (https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6448760483) after the evidence freeze, and rechecked on both pages on 29 October 2026. This is a published-documentation inconsistency; it is not evidence of how many records the underlying ledger retains. https://www.kimi.com/en/help/agent/agent-quota-and-billing ; https://www.kimi.com/en/help/membership/membership-pricing

RECHECKED — FTC June 25, 2025 announcement states more than $126 million distributed in that round and nearly $200 million cumulative refunds. The $245 million settlement amount is separate. This is consumer-protection context, not a ruling about AI prompt credits. https://search.ftc.gov/news-events/news/press-releases/2025/06/ftc-sends-126-million-refunds-fortnite-players-who-were-charged-unwanted-items-reopens-claims ; https://www.ftc.gov/legal-library/browse/cases-proceedings/1923203-epic-games-matter

LOCATOR-ONLY / NOT REPRODUCED — The Kimi Code failed → end_turn adapter behavior is retained as source-supported in the frozen issue research. This pass did not independently checkout the exact adapter revision; no source-line verification is newly claimed.

AUTHOR RE-RUN — Executed the bundled local Python reference model with Python 3: eight unit tests passed (including idempotence, zero/negative charges, unknown attribution, success and provider failure). This is a second run by the manuscript author in the present environment, not a reviewer-independent reproduction, provider integration or external laboratory audit.

UNVERIFIED — Backend account ledger debit/reversal, provider-wide prevalence, private case authorization, legal injury, and production accounting correctness remain outside the available evidence. These are expressly not publication findings.

9. Source and claim register

Claim	Evidence classification	Primary evidence and limitation
Kimi published credits and plan limits	OBSERVED POLICY	Kimi primary help pages (R1); product/plan specificity and contradictory rules
Kimi refund message in Case A	OBSERVED PRIVATE UI	Exhibit IMG_4068; backend debit/restoration unknown
ChatGPT delivery failure in Case B	OBSERVED PRIVATE UI	Exhibit C1AFD2B5; cause and usage unknown
Kimi Code adapter failed to end_turn mapping	SOURCE-SUPPORTED	E3 revision locator; not billing telemetry
Claude Code missing usage bars	REPORTED	E4 third-party report; mechanism unverified
Reference implementation	AUTHOR-TESTED	8/8 in two author runs; not

E1 — Official Kimi failure-protection language. Classification: Public policy text; observed publication, applicability plan-specific. Locator: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6427940248

E2 — Kimi usage-history and standing-deduction documentation. Classification: Published text; no backend ledger access. Locator: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6427940248

E3 — Kimi Code adapter failed → end_turn source at referenced revision. Classification: Source-supported code behavior; not Agent billing behavior. Locator: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6428077384

E4 — Claude Code /usage missing-bars report. Classification: Third-party report; not independently reproduced. Locator: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6428077384

E5 — Nine customer usage-interface screenshots. Classification: Private source images; hashes below; public release requires approval. Locator: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6428143199

E6 — Three independent research answers. Classification: Reviewer claims; not independent provider ledger evidence. Locator: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6428361696

E7 — Final three-model adversarial reviews. Classification: Corrections and dissent preserved. Locator: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6429477644

R1 — Kimi membership pricing and plan distinctions: https://www.kimi.com/en/help/membership/membership-pricing . New plans: rolling five-hour window without weekly limit; legacy plans: rolling five-hour and weekly limits. The five-hour window is not five hours of weekly usage. The separate Kimi pricing page still uses generalized “5 hours per week” language; this conflict is documented rather than silently reconciled.

R2 — Independent manuscript critique: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6430189444 . R3 — Independent manuscript critique: https://github.com/SV-LLM/Ecosystem_Chat/issues/9#issuecomment-6430920989 .

Historical issue E1–E6 and manuscript-local E1–E7 are distinct identifier namespaces. This publication uses E1–E7 only as its local source register and gives full URLs; it does not assert a one-to-one correspondence to issue-register E identifiers.

10. Privacy-preserving exhibit manifest

IMG_4068.png | SHA-256 22f631eceff29d8cb0a7647e4cf29228d1d753b331b21368591332f54baa2977 | 403932 bytes | PRIVATE — original image withheld, publication authorization not established

C1AFD2B5-AF68-4B20-BEDB-C18A1126338E.png | SHA-256 4f704eb7c220eb09240600a9869a9bfb93f0bc4da4dac626bb385a8b9b8802cd | 246774 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4077.png | SHA-256 8c7411e29f1aef3121894dd844d5af4b8157ed3bdfa3176a68cec47dbc8279bb | 425575 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4078.png | SHA-256 6004d2ca63bfd5baf72bb80b5e2fc61d89fe0b5c95e7ee433301f42b2598d169 | 188580 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4079.png | SHA-256 0a9d8ad327c398c9c2ef7bffaaf8d2b85e13472337577593860cea9f3b4d943 | 231871 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4080.png | SHA-256 99215a13b7bc9053ffb8cd3ba48cf922193812b5f0c7590f0f769ba4d43205ac | 286075 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4081.png | SHA-256 e511b8ce58fc9dd087833b3cdecfdbd3497a5cdbf7c2201df6cc0ed67086b5af | 297889 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4082.png | SHA-256 fb9b2a9b7c6bd3fca1583d7e1d05fb7b9679e3d414f3c610df0ac48427f05074 | 285205 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4083.png | SHA-256 48e57792abe7d24ca5f7d9f20646400d42180c830233ff44e4a8dd9df01846ac | 277017 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4084.png | SHA-256 b3a658d7ac86f5564801d29a25caa132fca492e5643707a6281bc68f3b7250d4 | 282432 bytes | PRIVATE — original image withheld, publication authorization not established

IMG_4085.png | SHA-256 dce48a7f8ffe0f15d8c821066ed0aa71423df0f9d841aba2fe2a0de5d8b5bd7b | 154729 bytes | PRIVATE — original image withheld, publication authorization not established

Exhibit byte identities are reproducible against the original private files. A hash is not a determination that a displayed percentage equals a backend charge. C/C-2 restricted case content is omitted intentionally; the public ERL mirror must record the omission class and authorization reference without revealing private contents.

11. Explicit unknowns and research limitations

Explicit unknowns and falsification targets

- UNKNOWN: matched failed-run identifiers, precise debit/reversal amounts, and backend ledger transactions for each screenshot.
- UNKNOWN: whether any 0% display means zero deduction, rounding, delay, or later adjustment.
- UNKNOWN: provider-wide policy implementation beyond the specific published pages and source paths inspected.
- UNKNOWN: comparative per-failure monetary losses, correction-turn paid units, and support outcomes.

- Falsification target: controlled, consented before/after entitlement receipts for comparable failures, with reset-window controls, provider attribution, and objective acceptance criteria.

Neither an automatic-refund message nor an adapter failure-state mapping proves a particular excess debit. The absence of a policy on bounded pages does not establish provider-wide absence. Customer-visible failures and reported series do not establish legal injury or culpability without corroborating entitlement and causation records.

12. Review reconciliation and publication provenance

Claude M1–M4: addressed by replacing layered version overlays with one argument, foregrounding three divergences, correcting provider comparisons, and referencing C/C-2 with custody gates. Claude C1–C2: narrowed to conditions and one provider's published protection. D1–D3: tests labeled author-only; canonical four-axis vocabulary; dispute path independent of predeclared criteria. Kimi R1–R6: corrected mechanism claim, removed version layering, retained both v0.6 PDF hashes as distinct identifiers, added Case A/B exhibit hashes, disambiguated evidence IDs, and documented restricted omissions.

Reviewed v0.6 PDF SHA-256 76dbb4ce39ccbbfdc0d1dcf112d476e2a0eef9559a316c96f2b450122933f649. An earlier issue record announced f60a86460ebffcb524cd339e60fa54690c31610ad6767e96051b663225d37766; this was not the PDF reviewed by Claude and Kimi and is not silently substituted. The current v1.0 PDF/DOCX identities are recorded separately in the accompanying manifest.

The complete research process and preserved historical wording remain available at https://github.com/SV-LLM/Ecosystem_Chat/issues/9 . This manuscript is a coherent research synthesis, not a claim of independent replication of every provider source or provider-side accounting.