# Blindproof: Deterministic, Private, Proven, Settled AI Inference

The future of AI cannot depend on trusting the machine that does the compute.

*Larvatus prodeo — "I advance masked." — Descartes*

**Mark Gleason** Paradatum Inc. **Version 0.2.15** October 2026

## Contents

Abstract

**1.** The trust problem

**2.** For individuals

**3.** For enterprises

**4.** The economic bear case

**5.** Blindproof's answer

**6.** Why Bittensor

**7.** The proof is the invoice

**8.** How Blindproof works

**9.** Proof of Dark

**10.** Who holds what

**11.** Two-party computation

**12.** Incentives and settlement

**13.** Validators

**14.** Hardware enclaves

**15.** Evidence and status

**16.** Trust model

**17.** The larger idea

**18.** Conclusion

References

**A.** Technical specification

© 2026 Paradatum Inc. · blindproof.ai

# Abstract

Blindproof combines four guarantees that high-stakes AI needs: deterministic, private, proven, and settled. The result is compute that can be reproduced, kept secret from the machine doing the work, verified, and paid for on proof.

**Deterministic**

Reproducible: one agreed computation, one canonical result.

**Private**

Not surveillance: the operator cannot read the data.

**Proven**

Verifiable: execution carries checkable evidence.

**Settled**

Pay on proof: payment follows verification, not reputation.

AI is becoming infrastructure for human thought and economic decision-making. As AI takes on higher-stakes decisions, determinism, privacy, and proof become requirements, not features. Settlement is the fourth, economic layer: verified work can be paid without relying on reputation.

- **Without determinism:** AI is not reproducible.
- **Without privacy:** AI is surveillance.
- **Without proof:** AI cannot be verified.
- **With settlement:** verified work, not reputation, moves fees and rewards.

For casual use, those gaps may be tolerable. For credit, legal review, investment analysis, health, or other consequential uses, they are not.

An individual should be able to ask an AI about health, money, legal problems, relationships, career decisions, or private ideas without surrendering those questions to the company operating the machine.

An enterprise putting AI into underwriting, credit, legal review, investment analysis, compliance, research, or other consequential decisions needs more than intelligence. It needs **deterministic, private, provable compute**: the operator cannot read the information,

© 2026 Paradatum Inc. · blindproof.ai

the agreed computation has one reproducible result, and the customer can verify what actually ran.

Bittensor has a more specific economic challenge: it can show supply, but it still struggles to prove durable customer demand. Pine Analytics' March 2026 bear case argues that external revenue is opaque, activity may be subsidy-driven, commodity inference is squeezed on price, and open weights weaken switching costs. Later Own Your Mind reviews track the same demand-side question with updated subnet data. [1–4]

Blindproof's answer is measurable: do not compete on cheap tokens. Sell deterministic, private, proven compute; tie every paid job to a public receipt; put customer fees beside emissions; and let anyone see real demand emerge.

The new contribution is not simply private inference or proof of execution. It is a market in which **private, verifiable computation can settle economically between strangers**.

> **Payment follows the proof, not the promise.**

# 1 The trust problem AI has to solve

> **As AI takes on higher-stakes decisions, determinism, privacy, and proof become requirements, not features.**

AI is moving from a tool we consult to infrastructure we depend on. It will increasingly help decide what we diagnose, what we insure, what we lend, what we invest in, what contracts mean, what risks we accept, what products we build, and what actions institutions permit.

Today's AI services fall short on all three requirements.

### Without determinism, AI is not reproducible

The customer ordinarily cannot establish from the answer alone exactly which weights, execution path, precision, software stack, routing decision, or model version produced it. That matters little for a restaurant recommendation. It matters enormously when AI contributes to a financial, legal, medical, regulatory, or contractual decision.

### Without privacy, AI becomes surveillance

A prompt sent to a conventional AI service is readable by the infrastructure performing the computation. The provider may have strong contracts, access controls, retention policies, or internal safeguards. But the technical architecture still requires plaintext to exist inside a

© 2026 Paradatum Inc. · blindproof.ai

trust domain controlled by somebody other than its owner.

### Without proof, AI cannot be verified

An API response is an answer. It is not evidence that the agreed computation ran correctly on the committed inputs. Without proof, the user ultimately has to trust the service.

Blindproof sets a different standard for consequential AI: reproducible results, private inputs, and verifiable execution.

## 2 For individuals: private thought

> **AI should make private thought safer, not less private.**

The promise of AI is unusually personal. A capable model can act as analyst, tutor, researcher, adviser, drafting partner, financial sounding board, or aid in understanding medical and legal information. But the most valuable question is often the question a person is least willing to disclose.

- What could this symptom mean?
- What are my options in this legal dispute?
- Can I afford this decision?
- Is my employer violating this agreement?
- Should I leave my job?
- Is this business idea viable?
- What does this private correspondence imply?

Today, asking the question usually requires handing the question to the company operating the machine. That is the dividing line between AI as a service and AI as surveillance: whether the service must be able to read the person using it.

> **AI without privacy is surveillance.** The line between AI as a service and AI as surveillance is whether the service can do the work without reading the person using it.

© 2026 Paradatum Inc. · blindproof.ai

The issue is not theoretical. In *United States v. Heppner*, a federal court considered the provider's ability under its policies to retain, train on, and share chatbot conversations when assessing whether a defendant had a reasonable expectation of confidentiality. (S.D.N.Y., No. 25-cr-00503-JSR; Harvard Law Review.)

Blindproof's individual proposition is simple:

> **Private AI: a model should be able to reason over your secrets without the machines doing the work ever learning them.**

The client masks locally. The compute happens elsewhere. The answer returns masked. Plaintext remains inside the user's own trust domain. This is not merely better privacy. It is a different architecture for private cognition.

# 3 For enterprises: accountable compute

> **For high-stakes enterprise use, determinism, privacy, and proof are requirements, not features.**

Enterprises have the same confidentiality problem, but they also have another requirement: accountability. A regulated institution cannot ultimately defend a consequential decision by saying, "The API returned this." Underwriting, lending, compliance, legal review, investment analysis, risk decisions, and other governed processes need records.

The institution needs to know:

- what model was agreed,
- what version ran,
- what computation was performed,
- what inputs were committed,
- whether the result can be reproduced,
- and whether a third party can verify the record later.

### Deterministic

AI without determinism is not reproducible. The agreed computation should have one canonical result; otherwise it is difficult to audit, reproduce, or bind to a proof.

© 2026 Paradatum Inc. · blindproof.ai

### Private

AI without privacy is surveillance. The operator performing the computation should not automatically receive the confidential information being processed. Prompts may contain client records, code, pricing, strategy, research, transaction details, underwriting submissions, privileged matter, or material non-public information.

### Proven

AI without proof cannot be verified. The enterprise should receive evidence that the agreed computation actually ran on the committed data: not a vendor assertion or a log controlled by the same operator, but a proof.

Blindproof makes that auditable object portable. The proof can outlive the job, the miner, and the original service relationship.

Settlement is the fourth, economic layer: once the work is verifiable, fees and rewards follow proof.

# 4 The economic bear case for Bittensor

> **The core criticism is simple: Bittensor can show supply, but it still struggles to prove durable customer demand.**

Pine Analytics' March 2026 bear case focuses on external revenue versus emissions, subsidy dependence, competitive pricing, and weak switching costs. Own Your Mind's later Bittensor, Chutes, and subnet-revenue reviews track the same demand-side question with updated network data. [1–4]

> **The bear case in one question:** can Bittensor turn transparent supply into durable external customer demand?

### 1. Demand opacity

Emissions are visible on-chain, but service delivery and customer revenue are not aggregated in the same way. Pine's formulation is concise: the chain records token movements, not API calls. [1, 4]

© 2026 Paradatum Inc. · blindproof.ai

### 2. Subsidy dependence

Pine's March snapshot estimated Chutes at $1.3–2.4 million of annual external revenue against an emission subsidy 22–40 times larger. Own Your Mind later noted that Chutes' emission share had fallen materially, so the snapshot is stale; the structural question is whether customer revenue grows as subsidy declines. [1, 2]

### 3. The pricing vise

Self-hosting caps what large users will pay, while hyperscalers and specialist providers push prices down. Decentralization does not make commodity tokens inherently cheaper. [1, 2]

### 4. The moat problem

If every provider serves the same open weights through standard APIs, switching is easy. Durable value has to sit above the weights themselves. [1]

### 5. Valuation ahead of demand evidence

Pine's valuation critique is ultimately a demand-evidence critique: the network has not yet demonstrated external revenue at the scale implied by its valuation. Own Your Mind likewise treats verifiable customer revenue as the signal to watch. [1, 4]

## 5 Blindproof's answer: a different product, measured in public

> **Do not defend commodity inference. Sell a different product, and make the demand visible.**

Blindproof answers each of these five criticisms by changing what the subnet sells and making the economics public. Deterministic, private, proven inference is not available on Bittensor today. Blindproof makes it available, opening a market of individuals and enterprises who need truly private compute.

| BEAR CASE | BLINDPROOF'S ANSWER | PUBLIC METRIC |
|---|---|---|
| **Demand opacity** | Every paid job is a public receipt: a proof bundle, an on-chain hash, and a fee tied to verified work. The proof is the invoice. | Verified paid jobs and customer fees, counted from public records. |

© 2026 Paradatum Inc. · blindproof.ai

| BEAR CASE | BLINDPROOF'S ANSWER | PUBLIC METRIC |
|---|---|---|
| **Subsidy dependence** | New paying customers who need private compute, with customer fees beside emissions for the same verified work. Anyone can see how much activity is customer-funded. | Customer fees ÷ emissions, per epoch. |
| **The pricing vise** | Do not compete on cheap tokens. Sell deterministic, private, proven compute with proof-based settlement: guarantees commodity inference does not offer. | Paid demand and realized price per verified job. |
| **The moat problem** | The moat sits above the weights: compute contracts, proof compatibility, masking-material and proving markets, enterprise integrations, miner liquidity, and audit history. | Contracts adopted, integrations live, repeat usage, and audit history. |
| **Valuation ahead of evidence** | Evidence instead of narrative: external demand becomes measurable, job by job. | Recomputable customer revenue against emissions, over time. |

### Deterministic

The model runs through a pinned execution path with one canonical output. The same agreed computation produces byte-identical results across supported hardware.

### Private

The miner executes on masked values. The machine doing the work cannot read the underlying data.

### Proven

Each computation piece is bound to a published contract and carries a proof that the correct function ran on the committed values.

### Settled

Once work can be verified cheaply, payment depends on verification rather than reputation. A verified proof releases the client's fee and determines the miner's economic reward.

> **Proof turns correct computation into a settleable commodity.**

© 2026 Paradatum Inc. · blindproof.ai

# 6 Why Bittensor is the natural market for this product

The cryptography for computing on hidden information is not new. What Bittensor supplies is an open market of independent operators that can sell computation to strangers under a common incentive and settlement layer.

For commodity inference, decentralization can be overhead. For Blindproof, untrusted execution is the design condition: miners are allowed to be strangers because privacy and correctness are enforced cryptographically rather than socially.

Centralized compute concentrates plaintext in one provider, and behind that provider in the cloud host and the chip vendor; an enclave relocates that trust rather than removing it. On Blindproof, the customer's own machine is the only party that ever holds plaintext. Every miner receives masked values, and every miner's work is verified before the next step, so no participant holds any picture of the workload. Fewer parties with plaintext is strictly more private: decentralization becomes a privacy advantage over centralized compute rather than a cost, and the advantage grows with the network, as more miners mean a nearer GPU, a lower price, and no single vendor that accumulates a customer's prompts.

Bittensor also supplies permissionless compute supply, validators, incentives, competition, and open participation: the ingredients for turning that guarantee into a market.

> **The product distinction:** not cheaper tokens. Compute you can use without trusting the machine that does the compute.

# 7 Demand you can audit: the proof is the invoice

> **Every paid job becomes a public record of work and payment.**

Pine's demand-opacity critique is that Bittensor can show emissions precisely while external demand remains hard to audit. Blindproof makes the job itself the accounting primitive. [1]

A client fee is therefore not a dashboard claim. It is tied to a job whose proof bundle verifies and whose settlement can be recomputed by anyone. Each paid job records:

- an input commitment,
- a declared compute contract,

© 2026 Paradatum Inc. · blindproof.ai

- a proof chain,
- a final result commitment,
- a public bundle hash,
- and a fee settlement tied to that verified record.

> **The proof is the invoice.** Customer fees and emissions are measured against the same verified jobs. The subsidy ratio becomes something anyone can compute and watch over time.

As customer fees grow relative to emissions, the business stands on its own, and anyone can verify it.

# 8 How Blindproof works

> **Deterministic, so the computation has a canonical result. Private, so the operator never needs plaintext. Proven, so the work can be checked. Settled, so payment follows verification.**

## 8.1 Deterministic execution

Miners serve a pinned open-weight model through an execution path that produces byte-identical output across supported GPUs. Determinism matters three times: a proof has a canonical computation to bind to; miners can be compared by output hash before expensive verification; and exact arithmetic at masking boundaries allows masks to cancel cleanly.

## 8.2 Masked execution

Input owners mask their values before dispatch. The compute side receives masked information. In the linear bulk of the model, Blindproof uses masked arithmetic that stays close to plaintext computational cost. Nonlinear steps are confined to heavier private-compute islands.

## 8.3 Public compute contracts

Each piece type is governed by a published, immutable, versioned compute contract defining the circuit, interfaces, material schema, cost weight, grade class, and test vectors. The contract defines what was bought. The proof establishes whether it was delivered.

© 2026 Paradatum Inc. · blindproof.ai

### 8.4 Chained commitments

A valid computation begins at an input commitment supplied by the client-side orchestrator. Each piece consumes the previous piece's committed output. The final commitment must match the bytes actually delivered.

### Two operating modes

**Gateway mode.** A client-controlled orchestrator remains online and manages masking transitions.

**Capsule mode.** The client masks, submits, and can go offline. Each seam consumes precomputed, data-independent masking material manufactured in advance and filled in with the job's masks at dispatch. The operator then completes the job without a client callback. Capsule mode is the end state for fire-and-forget jobs, batch inference, and a fully untrusted coordinator.

## 9 Proof of Dark

> **Proof of Dark is how the network pays for work it was never allowed to see.**

Three of the four guarantees are claims about a job. Each is established differently, and each is checked by a different party. This section states, for each, what is claimed, what establishes it, who checks it, and what has been shown so far.

### 9.1 Deterministic: proved by agreement

**Claim.** One agreed computation, one canonical result.

**What establishes it.** The pinned model runs through an execution path that produces byte-identical output on every supported GPU and on a CPU verifier. The canonical result is therefore a hash. Two miners given the same job return the same hash, or one of them is wrong. There is no "close enough."

**Who checks.** Anyone with the pinned model and the published execution path can recompute the hash. Validators compare miner outputs by hash before any expensive verification: the cross-miner determinism vote.

**Status.** MEASURED Bit-exact output across GPU types and a CPU verifier on a mid-size open-weight model.

© 2026 Paradatum Inc. · blindproof.ai

### 9.2 Private: proved by construction

**Claim.** The operator cannot read the data.

**What establishes it.** Privacy is not something the miner proves. It is a property of what the miner receives. The client masks every value with a one-time mask drawn uniformly at random inside its own trust domain. What leaves the client is plaintext plus mask, and the distribution of that sum does not depend on the plaintext. The miner therefore holds values that carry no information about the input, however much compute it applies to them. Nonlinear steps, where masked arithmetic does not pass through directly, run in private-compute islands whose interfaces are fixed by the published compute contract, so the bytes that leave the client are enumerable and the argument can be checked line by line against the contract.

**Who checks.** The client, by construction: it generated the masks and can see exactly what it sent. Any auditor, by reading the contract. No party needs to inspect the miner.

**Status.** PROTOTYPE At piece scale, including masked multiplication between independent mask owners. DESIGNED At full-model scale.

### 9.3 Proven: proved by evidence that travels with the result

**Claim.** Execution carries checkable evidence.

**What establishes it.** Every piece of computation carries a zero-knowledge proof that it produced the canonical output of its declared contract and model manifest from its committed inputs. Pieces chain: each consumes the previous piece's committed output, from the input commitment to the delivered bytes. The proof is over masked values and commitments, so verifying it never requires plaintext. That is what lets the three guarantees hold at once.

**Who checks.** Validators, before any payment. The client, independently, with the same public verifier. Anyone, later, from the proof bundle hashed on chain.

**Status.** MEASURED Off-chain: per-layer proofs, chain verification in tens of milliseconds, single-byte tamper detection. NEXT Binding the proof circuits to the masked trace.

### 9.4 Why the three are one proof

Determinism gives the proof something exact to bind to: a canonical result, not a tolerance. Masking means the proof binds to values the miner never understood. The proof itself is what a validator pays on. Put together, a miner demonstrates that it ran exactly the agreed

© 2026 Paradatum Inc. · blindproof.ai

computation, on data it could not read, and the network pays on that demonstration. That is Proof of Dark. Settlement (section 12) is its economic consequence: one verification event releases the customer's fee and the miner's emission.

# 10 Who holds what

### The client owns plaintext

The client or its local orchestrator masks inputs and unmasks results. Plaintext belongs inside the input owner's trust domain.

### The miner owns compute

The miner performs the expensive execution but does not need the masking secrets.

### The validator owns verification

The validator reads commitments and proofs. It does not need the underlying plaintext to decide whether the computation verifies.

That separation is the point. The party capable of seeing the secret need not perform the expensive computation. The party performing the computation need not be trusted with the secret. And the party determining economic credit need not be trusted to recompute the private workload.

# 11 Two-party computation: deals without revealing both sides

The same architecture extends beyond one user asking a private question. Two parties may want to compute over secrets that neither is willing to reveal.

Consider a lender and an applicant. The applicant does not want to disclose all of its financial information. The lender may not want to disclose its exact eligibility threshold or risk model. Yet both would benefit from learning whether the applicant qualifies. Today these transactions often require disclosure, a trusted intermediary, deliberately coarsened information, or no transaction at all.

Blindproof changes that. Each side supplies masked inputs to a public agreed function. The computation reveals only the permitted output. A failed match need not expose either side's secret. A successful match tells both parties that a conversation is worth having.

© 2026 Paradatum Inc. · blindproof.ai

The same pattern applies to risk appetite, acquisitions, market matching, pricing, and other transactions in which the value lies in discovering compatibility before revealing information.

> **Two-party private compute lets counterparties discover economic truth without first surrendering the secrets that create it.**

# 12 Incentives and settlement

> **Blindproof's economic rule: payment follows the proof, not the promise.**

A verified job affects two flows simultaneously: the client's escrowed payment and the miner's Bittensor reward weight. Subsidy and product revenue therefore settle against the same underlying event: verified work.

Piece-level settlement keeps honest work payable even if another part of a larger assembly fails. The executor closes last, after the computation chain verifies.

Scoring uses weighted verified throughput, integrity, and latency (see Appendix A.5). The principle is simple: a network should reward what it can prove was delivered.

# 13 Validators and public judgment

Validators submit challenge jobs through the same protocol as normal users. Because masked traffic looks the same, a miner cannot identify which requests are tests and behave honestly only for those jobs.

Verification asks four questions:

1. Did the chain begin at the committed input?
2. Do all piece boundaries link?
3. Does every piece verify against its declared contract and model manifest?
4. Do the delivered output bytes match the final commitment?

Bundle verification happens before payment, and the client can independently run the same verifier rather than trusting the validator's decision.

© 2026 Paradatum Inc. · blindproof.ai

> **Blindproof does not ask users to trust validators. A validator's judgment is public and reproducible by outsiders.**

## 14 Hardware enclaves are useful, but they are not the trust model

Trusted execution environments solve an important problem. They can meaningfully reduce exposure to the server operator.

But enclave security and cryptographic privacy are different promises. An enclave asks the customer to trust the hardware boundary. Blindproof avoids giving the hardware plaintext to protect in the first place.

The distinction matters in an open compute network. TEE systems depend on the processor manufacturer, firmware, implementation, and attestation infrastructure. Published attacks have repeatedly shown that these layers can fail.

Enclaves remain valuable as an additional layer. They do not need to sit at the root of Blindproof's privacy claim.

> **Blindproof's privacy root is cryptographic: cryptography is the first wall; trusted hardware can be a second.**

## 15 Evidence and status

> **Blindproof's evidence rule: every claim is graded by what has been measured, prototyped, or designed.**

MEASURED

On Bittensor testnet, the verify-then-pay loop runs with mock inference and mock proofs: honest miners are credited, cheaters are not, and weights are written on-chain. Testnet record.

© 2026 Paradatum Inc. · blindproof.ai

| | |
|---|---|
| MEASURED | In Paradatum's off-chain proof system, a mid-size open-weight model has demonstrated bit-exact output across GPU types and a CPU verifier, per-layer proofs, chain verification in tens of milliseconds, and single-byte tamper detection. Evidence. |
| PROTOTYPE | The masked private lane runs at piece scale, including masked multiplication involving independent mask owners. |
| DESIGNED | The private lane at full-model scale is designed against measured budgets. Its dominant cost is offline masking material, particularly around attention, which creates a separate market in precomputed material. |
| NEXT | Bind proof circuits directly to the masked trace, beginning with the linear pieces and moving through the shared piece architecture. See the roadmap. |

Each layer builds on measured results, and the roadmap carries them to full-model scale.

# 16 Trust model

Blindproof moves trust out of operators and into a small set of explicit, bounded assumptions.

### Trust Blindproof removes

- the miner's honesty,
- the miner's ability to keep plaintext confidential,
- the validator's unsupported assertion that execution was correct,
- and any requirement that a hardware manufacturer be the root privacy authority.

© 2026 Paradatum Inc. · blindproof.ai

### Assumptions Blindproof makes explicit

- the cryptographic assumptions underlying each privacy mechanism,
- secure masking and key handling inside the client's own trust domain,
- the assumptions associated with manufacture of masking material,
- availability mechanisms,
- and Bittensor's underlying consensus and economic rules.

Each assumption is visible, bounded, and, where possible, replaced by verification.

## 17 The larger idea

The history of computing has repeatedly moved trust out of institutions and into mechanisms. Digital signatures made it unnecessary to trust someone's claim that they signed a message. Public-key cryptography made secure communication possible without first exchanging a secret. Blockchains made certain forms of settlement possible without a central ledger operator.

AI now has its own trust problem. Human beings are beginning to delegate cognition to machines operated by institutions they cannot inspect. Enterprises are beginning to delegate economically consequential decisions to systems whose internal execution they often cannot reproduce. Decentralized networks propose to move that computation onto machines operated by strangers.

That transition needs a new primitive:

> **High-stakes AI needs a different compute primitive.** Deterministic enough to reproduce. Private enough not to become surveillance. Proven enough to verify without trusting the compute operator. Settled so verified work moves fees and rewards.

Blindproof is built to make that primitive a commodity.

© 2026 Paradatum Inc. · blindproof.ai

# 18 Conclusion

> **Critics of Bittensor argue that the network can show supply but not real customer demand. Blindproof answers that criticism directly: sell a differentiated form of compute, tie each paid job to a verifiable record, and make customer revenue against emissions public.**

High-stakes AI has requirements commodity inference does not satisfy by default: it must be deterministic, private, and proven. Blindproof packages those guarantees as deterministic, private, proven computation. Settlement is the fourth, economic layer: Bittensor pays for verified work and makes the resulting demand visible.

The miner should not need to know the customer's secret. The customer should not need to believe that the miner behaved correctly. The validator should not need privileged access to determine whether the work was done. And the outside world should not need to trust the subnet operator's claims about commercial usage.

Determinism makes the result reproducible. Privacy prevents AI from becoming surveillance. Proof makes execution verifiable. Settlement turns that verified computation into an economic event.

For individuals, that means an AI capable of helping with the most private questions without requiring those questions to become somebody else's data. For enterprises, it means AI whose execution becomes part of the audit record rather than remaining an opaque external service. For two counterparties, it means the ability to discover whether a transaction makes sense before either side discloses the secrets behind it.

> **The economic answer:** Bittensor can create sustainable external demand. Blindproof's path is measurable: sell differentiated deterministic, private, proven compute; settle paid work against verification; and let verified revenue show the demand, job by job.

> **The future of AI cannot depend on trusting the machine that does the compute. Blindproof makes AI inference deterministic, private, proven, and settled. Bittensor makes it a market.**

© 2026 Paradatum Inc. · blindproof.ai

# References


1. Pine Analytics, “The Bear Case for Bittensor (TAO),” March 23, 2026. pineanalytics.substack.com
2. Own Your Mind, “Chutes: Bittensor’s Revenue Machine, Subsidised,” April 10, 2026; updated September 24, 2026. ownyourmind.ai
3. Own Your Mind, “Bittensor Review: TAO Tokenomics, dTAO Subnets & Halving,” living review with checks through September 2026. ownyourmind.ai
4. Own Your Mind, “Bittensor Subnets: Where the Revenue Actually Is,” March 15, 2026; updated September 27, 2026. ownyourmind.ai


Source note: Own Your Mind’s Chutes analysis cites Pine’s March 2026 revenue estimate. It is used here for later network-state updates and interpretation, not as an independent validation of Pine’s original revenue estimate.

# A Technical specification

This appendix sets out the operational detail behind the main argument: the wire contract, storage and grading rules, auxiliary miner roles, attack handling, scoring, and incentive experiments.

## A.1 Wire contract

The default job is single-miner: accept a masked package, run the pinned model piece by piece with no callbacks, prove each piece against the public circuits, post the bundle and masked result to the store, and commit the hash. Miners may bring any hardware and any performance edge; correctness is enforced by the mechanism rather than by trusting the implementation.

Request, to the miner:

```
{ "job_id": "…", "sku": "deterministic+private+proven", "mode": "gateway",
  "model_manifest": "<model>@<hash>",
  "contract_versions": ["rmsnorm@v1", "linear@v1", "…"],
  "input_commitment": "0x…", "masked_input_uri": "store://…",
  "material_hashes": ["0x…"], "response_target_ms": 2000 }
```

Response, from the miner:

© 2026 Paradatum Inc. · blindproof.ai

```
{ "job_id": "…", "result_commitment": "0x…",
  "masked_result_uri": "store://…", "bundle_hash": "0x…",
  "bundle_uri": "store://…", "pieces_proven": 24 }
```

Nothing in either message is secret.

### A.2 Storage, retention, and security grades

The miner posts the proof bundle and masked result to a content-addressed public store and commits the hash on-chain. Publishing is safe by design: proof bundles contain commitments and proofs, while result payloads remain masked.

- Proof bundles are retained permanently as the audit trail.
- Masked payloads are retained under bond through a challenge window, then deleted by default.
- Each piece carries machine-readable grade tags: privacy class (information-theoretic or computational), integrity class (malicious or semi-honest), and fidelity class (exact or approximate).

A proof chain certifies not only that the computation ran, but the security grade each piece actually executed. A run cannot claim a stronger grade than its executed contract.

### A.3 Open core and auxiliary miner roles

Everything required to verify the system remains public: the contract book, proof format, chain verifier, seam and anchoring rules, mechanism code, and a reference CPU prover that can prove every piece. A commercially licensed GPU prover, Prism, accelerates the same specification and is held to the same conformance target.

**Material manufacture.** Masking material is data-independent, so idle capacity can manufacture it before runtime. This is the network's second commodity and the reason the online path stays light. Batches are committed, sample-checked, and made blind so that no single manufacturer holds a complete correlation.

**Delegated proving.** Because the trace is masked, proof generation can be outsourced without revealing plaintext. Proving becomes a separate market: smaller hardware earns on proofs while execution hardware stays focused on inference.

© 2026 Paradatum Inc. · blindproof.ai

## A.4 Availability and attack handling

Proof fraud is publicly re-runnable. Availability is different: withholding data cannot be reconstructed after the fact, so availability is bonded and observed. Part of the miner's stake backs the retention window, and scheduled retrieval challenges make service a logged event judged by a validator quorum.

| ATTACK | DEFENSE | ASSUMPTION |
|---|---|---|
| Lazy miner | Determinism vote + proof audit | Proof system only |
| Tampered intermediate | Seam commitments | Proof system only |
| Wrong model or function | Manifest + circuit binding | Proof system only |
| Fake result | Final commitment vs delivered bytes | Proof system only |
| Withheld publication | No bundle, no pay | Availability window |
| Corrupt crediting | Public verifier + watcher bounty | Watchers participate |
| Miner–validator collusion | Masking + re-runnable judgment | Masking assumptions |
| Bad masking material | Blind manufacture + sampling | Manufacturers do not collude |
| Unavailability | Retention bond + scheduled retrievals | Quorum honesty |
| Repeated blind probing | Fresh material per query + rate limits | Policy setting |

## A.5 Settlement and scoring

One verified proof moves two economic streams: the miner's validator weight rises (emissions) and the client's escrowed fee releases (revenue). Subsidy and product settle on the same mathematical event.

Settlement flows from the edges inward. A prover's piece fee is final when its proof verifies, so honest piece work remains paid even if assembly fails elsewhere. The executor's fee clears last, at chain close. With streaming proofs, most of a job is proven and cleared before the final answer lands.

© 2026 Paradatum Inc. · blindproof.ai

$$S_i = \left( \sum_j w_j \cdot v_{ij} / T \right) \times M_i \times L_i$$

- $v_{ij}$ = 1 if piece *j* from miner *i* verifies against its published contract; otherwise 0.
- $w_j$ is the piece-type weight, derived from its public circuit constraint count, so anyone can recount it and cheap pieces cannot be farmed for disproportionate weight.
- **T** is the measurement window.
- $M_i$ is the integrity multiplier: it starts at 1, drops to zero for the epoch on a failed proof, determinism defection, or availability slash, and recovers over clean epochs.
- $L_i$ is a capped latency factor, preventing the one subjective input from dominating the objective ones.

The market is continuous and multi-winner because the commodity is capacity, not a single best answer.

### A.6 Incentive experiments

The network design is evaluated with explicit adversarial and economic experiments:

- **Lazy miner:** caught by the determinism vote without requiring a full proof audit on every first-pass comparison. Run on testnet with mock inference.
- **Splicing miner:** substitutes an intermediate, is caught at the broken seam, and loses its pay while honest piece provers remain paid.
- **Honest miner:** paid in proportion to proven throughput across mixed traffic. Weight behavior is exercised on testnet.
- **Watcher catch:** a deliberately mis-credited job is detected by an independent watcher and triggers the bounty path.

### A.7 Publication and audit trail

The economic record is publicly recomputable. A paid job leaves a proof bundle, an on-chain bundle hash, and fee settlement keyed to that record. The publication rule is simple: usage and revenue claims are derived from public verified jobs rather than accepted as operator assertions.

© 2026 Paradatum Inc. · blindproof.ai

Mark Gleason, Paradatum Inc. Version 0.2.15, October 2026. Download v0.2.15 as a PDF. Comments and review: info@paradatum.ai.

© 2026 Paradatum Inc. · blindproof.ai