

DeFacts: The Fact Layer

Compute once. Share the result. Get paid.

A fact is that which is claimed and either mathematically proven or signed.

Mark Gleason Paradatum Inc. **Version 1.1.0** October 2026

Abstract

Every day, millions of people ask the same questions of the same models and pay, in money and energy, for the same answers to be computed again. DeFacts is the fact layer of the Paradatum stack: a shared, proven, paid-for cache of AI work. A result that has been computed once under a pinned contract, with a proof of what ran, is filed as a claim: the fact, its proof, its source hash and its model checkpoint. The next person who asks is served the stored claim, its proof re-checked in milliseconds, with no model run. The producer of the claim earns a royalty on every reuse. Blindproof runs the work; DeFacts remembers it; Knowledgecoin judges what it means.

A fact, in DeFacts, is that which is claimed and either **mathematically proven** or **signed**: proven, when a verifiable computation produced it; signed, when an accountable party, an oracle or a notary, put its key behind it. Nothing else is a fact. Three guarantees then define every entry:

Proven or signed

Provenance is checkable: a proof of the computation, or the signature of an accountable oracle or notary.

Portable

The record outlives the job, the server and the vendor. Bytes on public storage, hash anchored on chain.

Paid

Every reuse settles a royalty to the producer of record. Sharing pays; hoarding does not.

The result is a public library of machine-ready facts with provenance, built as a by-product of work people were already paying for. In the terms of the Knowledgecoin papers, DeFacts is

the shelves of the Perfect Library: the layer at which a book can no longer be burned, altered, misfiled or invented by any Librarian, mad or evil [1, 4].

No person should ever ask the same prompt against the same model twice. Compute it once, share the result, and pay the one who computed it.

1 The waste

AI inference is the first computing workload in history where the same question is answered from scratch, at full cost, every time it is asked. A web page is cached at the edge. A database query is cached in memory. A compiled program is compiled once. But a prompt sent to a model today triggers a fresh forward pass through billions of parameters on a GPU (graphics processing unit), and so does the identical prompt from the next user, and the next.

Three things made that acceptable until now:

1. **Non-determinism.** Standard GPU inference drifts: floating-point ordering and scheduling change the output bits from run to run and machine to machine. Two runs of the same prompt are not the same answer, so there was nothing canonical to cache.
2. **No proof.** An answer from an API is an assertion. Nothing binds it to the model, the weights, the inputs or the execution path that produced it, so a stored answer could not be trusted by anyone but the person who asked.
3. **No settlement.** Even if a result could be cached, there was no mechanism to pay the party who produced it when someone else used it, so nobody had a reason to share.

Deterministic, proven inference removes the first two obstacles [2]. DeFacts removes the third.

2 The library needs shelves

The greatest enemy of knowledge is not ignorance. It is an Evil Librarian who hands you censored, altered or invented information while maintaining the illusion that it is validated knowledge.

The Knowledgecoin position paper set out, in 2021, what a **Perfect Library** would have to be: trusted (every entry can be checked), spatially complete (the same knowledge available everywhere), temporally stable (what was true yesterday is still on the shelf today, unaltered), mutual (held in common rather than owned by one party) and parsimonious (indexed so that the minimum is read to answer a question). It then described three ways a library fails, each named for its Librarian [1, 4].

- **The Mad Librarian** burns random books, inserts phony ones and fails to index. The reader cannot tell what is missing or what is false, so every consensus on the shelves is in doubt. In 2026 the Mad Librarian is hallucination: a model that answers fluently about a paper never written and cannot say what it left out.
- **The Evil Librarian** does the same things on purpose, to an end, while keeping the doors open and the illusion of completeness intact. Malice is not required; capability and reach are. In 2026 the Evil Librarian is the silent update: a hosted assistant whose training mixture, tuning, system prompt and retrieval ranking nobody outside can audit, and whose overnight checkpoint rewrites every book at once. Yesterday's answer cannot be got back.
- **The Epistemic Agent Problem** is the question of who is paid for the knowledge the readers themselves create. In 2026 the answer is nobody: the world's epistemic labor is absorbed into model weights and rented back to the people who produced it.

Knowledgecoin answers at the level of the catalog and the jury: how claims are adjudicated into knowledge, and how the parties who do that work are paid [1]. But a catalog is only as good as the shelves beneath it. A library whose books can be burned, altered, misfiled or invented cannot be made trustworthy by any amount of adjudication above it. **Facts are the shelves.** DeFacts is the layer that makes each book a thing no Librarian can touch.

Measured against the five criteria, a DeFacts entry is:

CRITERION	HOW A DEFACTS FACT MEETS IT	WHICH LIBRARIAN IT DEFEATS
Trust	Every entry carries a proof of the computation or the signature of an accountable oracle or notary. The reader checks it; nobody is asked to believe.	Mad and Evil: phony books cannot be shelved as facts.
Space	Deterministic execution gives the same bytes on any hardware; the record lives on public storage, anchored on a public chain, readable from anywhere.	Evil: no section can be closed to one city.
Time	The model checkpoint, contract and source hash are pinned in the record. Yesterday's answer is on the shelf today, unchanged, and can be re-verified.	Evil: the silent update cannot rewrite what is already filed.
Mutuality	The library is held in common. Anyone can file, anyone can read, anyone can verify with the same public verifier; no operator sits between reader and record.	Mad: a misfiled or burned book is visible as a gap in a public index.
Parsimony	Asked again, the stored fact is served and re-verified in milliseconds. Resumable checkpoints mean a document is read once. The minimum is computed to answer a question.	Mad: the reader never searches the whole inventory.

And the Epistemic Agent Problem is answered at the shelf, not only at the catalog: the producer of record earns a royalty every time the fact is reused. The reader who did the work is paid when the next reader relies on it.

Deterministic, proven inference removes the Mad Librarian from the computation. A pinned checkpoint and a filed record remove the Evil Librarian from the shelf. Royalties on reuse remove the Epistemic Agent Problem from the economics. What remains is a library in which knowledge is a claim that has been tested by parties with something to lose, which is the layer above, and in which the claim itself cannot be altered, which is this one.

3 The fact layer

A **fact** in DeFacts is not a bare assertion. It is that which is claimed and either mathematically proven or signed. A claim arrives in one of two ways:

- **Proven.** A computation produced it under a pinned contract, and a proof shows the agreed model ran on the committed inputs. The fact is as trustworthy as the mathematics of the proof.
- **Signed.** An accountable party attested to it with a cryptographic signature. The fact is as trustworthy as the signer's key and reputation, both of which are on the record.

Anything that is neither proven nor signed is not a fact. It can still be stored, but it is marked as an assertion and cannot earn a royalty, be adjudicated, or be served as a fact.

Every entry is a record with five parts:

PART	WHAT IT IS
Claim	The result, in the DeFacts schema: the prompt (or its hash when private), the model checkpoint, the answer, and the extracted facts it contains.
Proof	The receipt from the run that produced it: contract pinned, inputs committed, output hash, proof that the agreed computation ran.
Source	The hash of any source document the answer drew on, so provenance can be checked without re-reading the source.
Attestation	Either the proof (for a computed fact) or the signature of the oracle or notary who vouches for it (for a signed fact), with the signer's public identity.
Producer	The party of record who paid for the run or placed the signed fact, and who earns the royalty on reuse.

Bytes live on durable public storage (Arweave); the hash and the ownership record are anchored on Solana. The entry can be re-verified by anyone, at any time, without trusting DeFacts, Paradatum, or the machine that ran the job.

Provenance is proven. Truth is not claimed. Whether the claim *holds* is the job of the layer above: Knowledgecoin adjudicates claims into knowledge by parties with something to lose [1]. DeFacts' job is narrower and foundational: make sure the claim, exactly as computed or signed, is on the record, with its attestation, forever.

4 Oracles and notaries

Not every fact is the output of a model. Prices, weather, sensor readings, court dockets, corporate filings, the text of a law as published on a given day: these enter the world as observations, not computations. For these, DeFacts admits two kinds of signer.

Notaries

A notary attests that a thing existed in a given form at a given time. The notary hashes a document, signs the hash with a timestamp, and files the signed hash. The notary does not say the document is true; the notary says this is the document, unchanged, as of this moment. A notarized source is what a proven extraction then runs against, so the chain reads:

Notary signs the source → Blindproof proves the extraction → DeFacts files the fact with both the signature and the proof.

Oracles

An oracle attests to the value of an external quantity: a price at a time, a reading from an instrument, a result from a system outside the network. The oracle signs the value, the time and its own identity. Where several oracles report the same quantity, DeFacts files each signed report; agreement among independent signers is itself a fact.

Same economics, same accountability

Signed facts carry the same economics as proven ones. The signer, or the party who paid the signer, is the producer of record and earns the royalty on reuse. A signer's history is public: every signed fact is on the record, so a notary or oracle that signs falsely signs its own reputation away. Where a signed fact is disputed, it goes up to Knowledgecoin adjudication like any other claim, and a signer whose fact is overturned can be slashed under the Knowledgecoin rules [1].

The two kinds of attestation compose. A proven fact can cite a notarized source; a signed oracle value can be the committed input to a proven computation; an adjudicated verdict is a signed fact whose signers are the panel.

Everything in the library bottoms out in either a proof or a key.

5 Compute once

The economic argument is simple. Let a job cost c to compute and r to retrieve and re-verify, with $r \ll c$ (milliseconds of hash checking against seconds of GPU time). If a prompt is asked n times:

$$\text{today: } n \cdot c \qquad \text{DeFacts: } c + (n - 1) \cdot r$$

For any prompt asked more than once, the saving is close to $(n - 1) \cdot c$. For the long tail of unique prompts, nothing changes: the job runs, and the result is filed for whoever asks next. There is no scenario in which DeFacts costs more than the status quo; there are many in which it costs a small fraction.

Where repetition is high, the saving is the business:

- **Public documents.** Every analyst who asks what a regulation, a filing or a court opinion says is asking the same model the same thing. One proven extraction, served thousands of times.
- **Reference questions.** Dates, definitions, standards, specifications. The first answer is the last computation.
- **Agent workflows.** Agents repeat sub-questions constantly. A shared fact layer turns repeated reasoning into lookups.
- **Resumable context.** DeFacts can store a verified checkpoint of a model's state after it has read a document, so the next question about that document resumes from the checkpoint rather than re-reading it. Same bytes, every time, because the run is deterministic.

6 Why cached compute is the future

Every prior layer of computing became infrastructure the moment its results became shareable: the compiled binary, the cached page, the content delivery network, the package registry. AI inference is still at the stage where every user rebuilds the binary. That cannot last. Model capability is converging on open weights; the cost of running them is converging on electricity; the thing that will differ between providers is not the answer but whether the answer is **proven, owned and reusable**.

A fact layer changes who benefits from a question being asked. Today the beneficiary is the provider, who gets paid each time. With DeFacts the beneficiary is the first asker, who gets paid every time after. The incentive flips from hoarding compute to sharing results, which is the incentive that built every other piece of shared infrastructure.

Privacy is not sacrificed. A job can run masked, so the operator never sees plaintext, and the asker decides whether the result is filed publicly, filed with the prompt hashed, or not filed at all. Public facts are shared because the asker chose to share them and is paid for it.

The question is not whether AI answers will be cached. Every other computing result already is. The question is who owns the cache and who gets paid when it is used.

7 Proof of Fact

Blindproof's Proof of Dark pays for work the network was never allowed to see [2]. DeFacts' **Proof of Fact** pays for work that was done once and used many times. The rules:

1. **Filing is paid for up front.** Placing a fact costs an escrowed fee. Spam has a price; real facts recover it.
2. **FACT is minted only against use.** New FACT is issued to the producer after real paid reuse of the claim, not at filing. Nothing is minted for facts nobody asks for.
3. **Every reuse pays a royalty.** Each retrieval, re-verification or adjudication of a claim settles a share to the producer of record.
4. **Gateways buy FACT to serve facts.** A gateway such as Paradatum that serves customers from the library needs FACT to pay producers. Customer demand for facts is demand for FACT.
5. **Receipts chain.** The filing receipt (R2) chains to the compute receipt (R1) that produced the claim, and forward to any adjudication receipt (R3). The whole history of a fact is one auditable chain.

Customers never need to hold FACT. They pay a gateway in dollars; the gateway settles the fact layer in FACT. The token exists for one job only: to be the producer's stake in the library and carry the royalty.

8 The DeFacts standard

Facts are filed in a machine-readable standard so that agents, not only people, can use the library:

- explicit and implicit facts, each with a confidence;
- resolved entities and their relations (subject, predicate, object);
- who, what, when, where, why, how;
- timeline and causal chains;
- full provenance: source hash, model checkpoint, contract, receipt, and for signed facts the signer's identity and signature.

A fact is useful to an agent only if the agent can check where it came from. Every field above is bound to the attestation.

9 Where DeFacts sits

LAYER	WHAT IT DOES	PROOF
Knowledgecoin	Adjudicates claims into knowledge; staked panels, verdicts on the record	Proof of Know
DeFacts	Files proven and signed results as claims; serves them again without re-running; pays producers on reuse	Proof of Fact
Blindproof	Runs the job privately with proof, on independent GPUs	Proof of Dark
Paradatum gateway	Pins the contract, escrows payment in dollars, issues receipts	—

A result flows up: Blindproof proves it, DeFacts files it, Knowledgecoin judges it. Knowledge flows back down as adjudicated claims, which are themselves filed. For the protocol that governs adjudication, staking and settlement across the stack, see the Knowledgecoin whitepaper [1]; for the full map, see the stack page [3].

10 Roadmap

1. **Dollars first.** Launch on USDC (USD Coin) settlement through the Paradatum gateway: prove-and-file is the first job type. Customers pay in dollars, results are filed with proof, no token.
2. **The library.** Seed the library with public documents: regulations, filings, standards, public records. Each source notarized, each extraction proven, served on demand.
3. **Oracles and notaries.** Open signing to accountable third parties: notaries for documents, oracles for prices, readings and external results.
4. **FACT.** Issue FACT when the library is worth re-verifying and there is paid reuse to settle. Mint only against use.
5. **Checkpoints.** Verified resumable model state for long documents.

6. **Adjudication.** When enough proven claims exist to argue about, Knowledgecoin adjudication goes live above the library.

11 Conclusion

A fact is that which is claimed and either mathematically proven or signed. Everything in DeFacts bottoms out in a proof or a key, and everything in it pays the party who put it there. That is what makes it a shelf no Librarian can reach: the Mad Librarian is removed from the computation by proof, the Evil Librarian from the record by the pinned checkpoint, and the Epistemic Agent Problem from the economics by the royalty. The question is not whether AI answers will be cached; every other computing result already is. The question is who owns the cache and who gets paid when it is used. DeFacts answers: the person who paid for the computation, or signed for the observation, owns the result; anyone can verify it; every reuse pays them.

Compute once. Share the result. Get paid.

References

1. Knowledgecoin whitepaper v2.0.0, Paradatum Inc., October 2026.
knowledgecoin.io/whitepaper
2. Blindproof whitepaper, Paradatum Inc., October 2026.
blindproof.ai/whitepaper
3. The Paradatum stack. paradatum.ai/stack
4. Knowledgecoin position paper, 2021: the Perfect Library, its five criteria, and the Mad Librarian, Evil Librarian and Epistemic Agent problems; restated in Part 1 of the Knowledgecoin whitepaper v2.0.0 [1].

Mark Gleason, Paradatum Inc. Version 1.1.0, October 2026. . Comments and review: info@paradatum.ai.