

Private AI compute you can check.

The system running at fumata.io today. Anything not live yet is marked. Benchmarks load live.

Revised 7 October 2026 ·

In one page

When you send a prompt to an AI service, you trust the operator twice: not to read it, and to run the model it says it runs. You can check neither. Fumata replaces both kinds of trust with things your browser verifies.

A Fumata job runs on a GPU machine inside an Intel TDX confidential virtual machine with NVIDIA confidential computing switched on. Before anything is sent, your browser checks Intel's signed quote and NVIDIA's signed GPU evidence for that exact machine and the exact software image it booted. It then encrypts your messages, in the tab, to a key that was created inside the enclave and is bound to that evidence. The relay in the middle, which is ours, only forwards ciphertext.

The enclave signs the hash of the exact encrypted request and response. Your browser checks that signature, and the result becomes a public receipt. Receipts are put into Merkle trees whose roots are written to a ledger contract on Robinhood Chain. Independent witness nodes, run by holders from a browser tab, check each receipt again on their own and co-sign it.

Anyone can open a receipt and repeat every check in their own browser. No one, including us, can read what you asked.

What runs today

Part	Status	Where to check
Sealed jobs on attested GPUs (text, documents, images)	Live	Run
Hardware check in your browser before each session	Live	Seal check panel on Run
End-to-end encryption to the enclave key	Live	"What left this tab" on Run
Enclave-signed receipts, verified on receipt	Live	Verify
Market of sealed models, re-checked every ten minutes	Live	Market
Witness nodes in the browser	Live	Nodes
Receipt roots anchored on Robinhood Chain	Live	FumataLedger
Credits in USDG or ETH, with the fixed split	Deployed · credits open at launch	FumataRevenue
Weekly witness rewards, claimed on chain	First week after launch	Nodes

Who you trust, and who you don't

Confidential computing does not remove trust. It moves it to a few parties that are easier to name and harder to corrupt.

Party	Can it read your job?	What you rely on
The owner of the GPU machine	No	Memory is encrypted by the CPU and the GPU; the machine proves its state to you.
Fumata (our relay and servers)	No	Messages are encrypted in your tab to the enclave key. We add an API key and forward bytes.
NEAR AI Cloud, which runs the machines	No	The software image is measured and the measurement is in the quote you check.
Intel and NVIDIA	Not by design	Their hardware and their signing keys. This is the trust that remains.
Witness nodes	No	They only see receipts, which hold hashes and signatures.

What the relay does see: which model you picked, when, and how many bytes went each way. It also counts your jobs against your daily allowance, which is why you sign in with a wallet. Receipts are never linked to wallets in public.

How a sealed job runs

1. The hardware proves itself

When you sign in or pick a model, your browser asks for the gateway's and the model machine's attestation reports. Each report holds an Intel TDX quote and, for the model, NVIDIA GPU evidence. The browser verifies the quote against Intel's collateral, sends the GPU evidence to NVIDIA's attestation service and checks NVIDIA's signed verdict, and checks that the report was made for a nonce it chose. The quote also commits to the hash of the machine's software description (its compose file), which we show as the measured build.

2. The job is sealed in your tab

The report binds a public key to the enclave. Your browser makes a fresh key pair, derives a shared secret with that key (secp256k1 ECDH) and encrypts every message field with AES-GCM before the request leaves the tab. Documents are turned into text in your browser first, and images are resized there, so they travel inside the encrypted messages too.

3. It runs inside the enclave

The relay adds an API key and forwards the bytes unchanged. Inside the confidential machine, the model decrypts the request, runs on the GPU in confidential mode and encrypts the answer back to your tab's key. It streams back in encrypted pieces that only your tab can open.

4. White smoke

The enclave signs `model:sha256(request):sha256(response)`, computed over the exact encrypted bytes, with its attested key. Your browser recomputes both hashes from the bytes it sent and received and checks the signature. It then gives our server the job's id. The server only accepts ids of jobs that went through its relay for your session, once each; it fetches the signature from NEAR AI itself, checks it again, verifies the signer's hardware evidence on its own and records the receipt.

The receipt

A receipt holds no prompt, no answer and no wallet. It holds:

- `sec`: its number, from 1, with no gaps.
- `model` and `signedText`: the model id and the two hashes the enclave signed.
- `signature` and `signer`: an EIP-191 secp256k1 signature and the address of the enclave key that made it.
- `composeHash`: the measured build of the machine, from its verified quote.
- `ts`: when it was recorded.

Its leaf is `keccak256(abi.encode(DOMAIN, seq, keccak(model), keccak(signedText), keccak(signature), keccak("ecdsa:" + signer), composeHash, ts))` with `DOMAIN = keccak256("fumata.receipt.v1")`. The Verify page recomputes it from the fields it shows, so a receipt cannot be edited without the check failing.

Anchoring on Robinhood Chain

Receipts are anchored in batches of consecutive numbers. A batch closes when it holds 256 receipts or when its oldest receipt has waited ten minutes. Its leaves go into a Merkle tree where pairs are hashed in sorted order and an odd node at the end of a level moves up unchanged, which is the rule OpenZeppelin's MerkleProof uses. The root is written to FumataLedger with `anchor(root, firstSeq, count)`, and the contract refuses any batch that does not start exactly where the last one ended.

Anyone can call `verify(batch, leaf, proof)` on the contract, or check the proof in the browser as the Verify page does. Once a root is on chain, no receipt in its batch can be changed, removed or reordered.

Witness nodes

A witness node is a browser tab. The wallet behind it signs one message that authorises a key created in that tab, and the key signs too, proving the tab holds it. From then on the server hands the tab receipts to check, picked at random among those that still need witnesses; a check is only accepted for a receipt handed to that node. The tab verifies each one with the same code as the Verify page (signature, model, leaf, and the signer's Intel and NVIDIA evidence, cached for thirty minutes per signer), and signs `fumata.witness.v1:leaf:verdict` with its key.

The first five valid checks of a receipt earn one point each; a flag is recorded on its own and takes no paid place. Weeks start on Monday at 00:00 UTC. When a week closes, the witness pool's USDG for that week is split by points and published as a Merkle root on FumataRevenue, and each node claims its amount with a proof. A receipt flagged by a witness is shown as flagged on its page.

Before the token launches, any wallet can run a node as an early witness. After launch, a node needs the holder balance shown on the Nodes page. It is checked when work is handed out, at every check, and again when the week is paid, so moving one stake from wallet to wallet earns nothing extra.

\$FUMATA and the money

Every wallet gets three free sealed jobs a day; holders get forty. Beyond that, jobs are paid with credits: 1 USDG buys 40 jobs, in USDG or in ETH swapped to USDG in the same transaction. FumataRevenue splits every purchase on the spot:

Share	Goes to	How it can leave
60%	Compute and operations	Sent to the treasury in the same transaction.
20%	Witness pool	Only through weekly Merkle claims, capped per week.
20%	Buyback and burn	Only through a buyback that sends every token bought to 0x...dEaD.

The token is not part of the security argument. Privacy comes from the hardware and the encryption; proof comes from the signatures and the anchors. The token decides who gets free compute and who is paid to witness.

Live benchmarks

Read from the network when this page loaded. Hardware checks are measured in users' browsers; evidence verification on our server.

4 sealed jobs run	2.6 s median hardware check in the browser
1.9 s median sealed job, send to signed answer	0.4 s median server-side evidence verification

32

attested GPUs across sealed models

4

receipts anchored on chain

0

witness checks signed

0

witness nodes online now

What we don't claim

- **Hardware is trusted, not proven.** Confidential computing has had side-channel attacks before. If Intel or NVIDIA hardware or keys are broken, the guarantees weaken.
- **"OutOfDate" is shown as it is.** Some machines report an Intel TCB status of OutOfDate, which means Intel has published a newer microcode or firmware for that platform. The quote is still genuine and verifies; we show the status on every receipt instead of hiding it.
- **Metadata is visible.** Our relay sees the model name, timing and sizes, and which wallet is signed in. It never sees content.
- **A receipt does not prove the answer is right.** It proves which measured machine produced it and that it was not altered. Models can still be wrong.
- **Witnesses check evidence, not content.** They cannot see jobs, so they verify signatures and hardware evidence, not answers.
- **Coordination is ours.** The relay, the receipt numbering and the anchoring keeper are run by Fumata. Anchors and witness signatures make tampering visible, not impossible to attempt.
- **One provider today.** The sealed machines run on NEAR AI Cloud. Other attested providers join the market when they pass the same browser checks.

Contracts and code

Two contracts on Robinhood Chain (chain id 4663), Solidity 0.8.26, no external libraries, tested on a fork of the live chain:

- **FumataLedger**: [0xd8726f4d386822cf51cba5f542c6162e94a40464](#). Anchors receipt roots; `batchOf(seq)` and `verify(batch, leaf, proof)` are public.
- **FumataRevenue**: [0x66a3a6fc1bd142dd6443d4305db0e001a2ab9ef3](#). Credits, the 60/20/20 split, buybacks and weekly witness claims. No withdraw path for the burn reserve or the witness pool.

The browser verification uses NEAR AI's open-source inference SDK, unchanged, together with viem for signatures and our 40-line Merkle module, the same one the contracts were tested against.

FAQ

Why sign in with a wallet?



To count free jobs and sell credits. It opens a 24-hour session and moves nothing.

Can I verify a receipt without trusting this website?



Yes. Download the proof bundle from any receipt page. The NEAR SDK, viem and a Robinhood Chain RPC are enough to check it.

What happens to my data after the job?



We keep only the receipt: hashes and signatures. The conversation lives in your tab and goes when you close it.

Why these models and not the famous closed ones?



Closed APIs can't prove what machine they run on. A model is listed only if it passes the browser checks.