Attest: a verifiable DBOM that proves what a model was trained on

PATH vc_fundable

ICP Enterprise model teams with regulatory or IP exposure

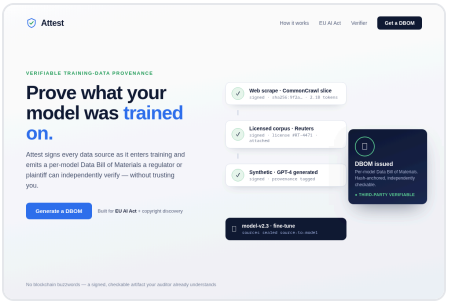
TAM $15B+ AI governance/compliance tooling by 2030

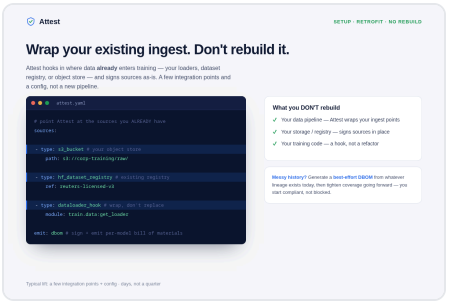
SAM $1B+ EU/IP-exposed model trainers, $50-150K ACV

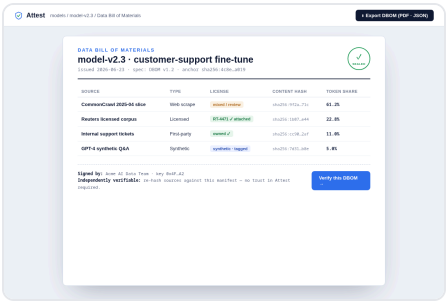
YR-1 SOM $1-3M ARR Year 1

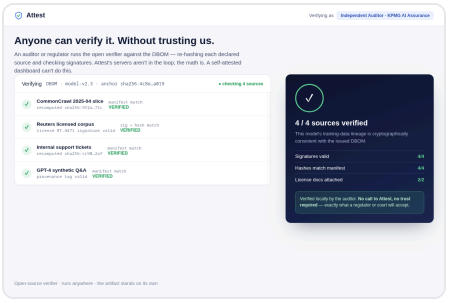
TIMELINE MVP: sign-and-emit DBOM + open verifier on toy corpus, validated by practitioners in 48h -> Launch: real integration on one pipeline producing a verifiable DBOM in ~1 week -> GTM: design-partner DBOM reviewed by a governance practitioner via audit/governance firms in ~2 weeks

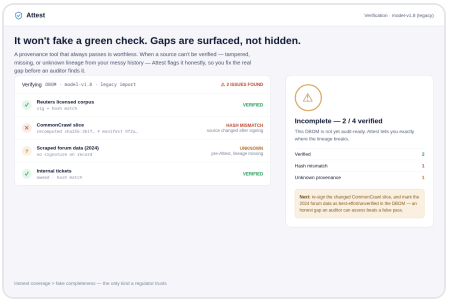
Visual concepts

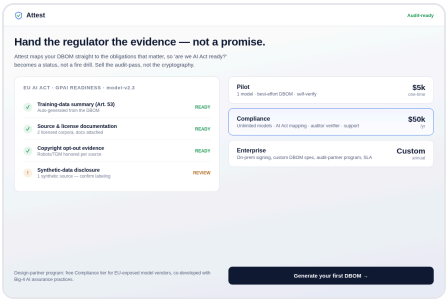












Try the clickable prototype →

https://kcio-state.tor1.digitaloceanspaces.com/ideas/6fa64f53-3231-4aa1-ab3a-70b1bdcb2395/prototype/index.html

Attest: a verifiable Data Bill of Materials that proves what a model was trained on

Don't sell blockchain — sell defensible compliance. Attest retrofits onto messy pipelines, signs each training source as-is, and emits a per-model DBOM an auditor independently verifies by re-hashing the artifacts, with our servers never in the loop. The compounding moat is becoming the regulator/auditor-recognized standard. Regulation is creating the demand on a deadline, so this one is honestly vc-fundable.

Strategy

This is the strongest of the batch, and the timing is exactly right. You framed it as "a cryptographic, verifiable chain-of-custody so models can prove what they were trained on — a per-model data bill of materials a third party can verify." The obvious-but-fatal version is "blockchain for AI training data" — leading with the crypto buzzword scares off the actual buyer and invites the "why not just a signed log?" objection. Here's the sharper framing.

The reframe. Nobody buys provenance for its own sake — they buy it because **someone with subpoena or fine power is about to ask "prove what this model was trained on" and they can't answer.** Reframe from "a verifiable data lineage system" to "the audit artifact that satisfies an EU AI Act regulator and a copyright plaintiff's discovery — the Data Bill of Materials (DBOM) you can hand to a third party who independently verifies it without trusting you." The cryptography is the *how*; the product is **defensible compliance**. Sell the artifact and the audit-pass, not the chain.

Falsifying proof point. The riskiest assumption is whether a *credible third party* (an auditor / regulator-aligned reviewer) will actually accept a DBOM as sufficient evidence — tech that produces an artifact nobody trusts is worthless. Week-1 test: produce a real DBOM for a small trained model from a mixed dataset (web + licensed + synthetic) and put it in front of 3 AI-governance/legal practitioners — would this stand up in an audit or discovery? ~$1.5K, days, mostly expert interviews + a working sign-and-emit pipeline on a toy corpus. If practitioners say "not enough," we learn the exact gap before building.

Target customer. Not "all model trainers" — the beachhead is **mid-to-large orgs training/fine-tuning models for regulated or IP-sensitive use** (enterprise AI teams, model vendors selling into the EU, foundation-model labs facing litigation). They have legal exposure, budget, and a board asking about AI Act readiness *now*.

Problem / why now. The EU AI Act's GPAI transparency obligations are phasing in (training-data summaries, documentation) and copyright suits (NYT, Getty, authors) are forcing discoverable lineage — and **today nobody can produce it**, because data pipelines were built with zero custody tracking. The regulatory clock is the "why now": demand is being legislated into existence on a deadline.

Value prop / wedge. Ship ONE thing: a pipeline integration that **signs each data source as it enters training and emits a per-model DBOM** (sources, licenses, hashes, transformations) that an

independent party can verify against the artifacts — without re-running the training. The wedge is the *verifiable artifact + the verifier*, not a full data platform. Land on "generate your AI Act training-data documentation automatically, provably."

Market (honest math).

- ICP: enterprise/model teams with regulatory or IP exposure (≤8 words).
- TAM: ~$15B+ AI governance/compliance + MLOps tooling by 2030 (regulation-driven, fast).
- SAM: orgs training/fine-tuning models touching EU or IP-sensitive use — ~10K orgs × ~$50-150K ACV ≈ **$1B+** serviceable, because this is enterprise compliance spend, not prosumer.
- SOM: ~$1-3M ARR Year 1 (design partners among EU-exposed model vendors + governance-forward enterprises).
- **Path = vc_fundable.** Defended SAM clears the $1B bar on enterprise ACV, the market is being created by regulation on a deadline, and the moat compounds. This is the one in the batch I'd put on the VC track with a straight face.

Moat / why us. A signed log is easy to build — the moat is **becoming the standard the auditors/regulators recognize**. (1) Trust/neutrality: an independent verifier is only valuable if it's credibly third-party. (2) Standard lock-in: if your DBOM format is what AI Act auditors accept, every model vendor selling into the EU adopts it, and switching means re-certifying. (3) Network: each accepted audit makes the format more authoritative. Regulatory-standard moats compound and are very hard to dislodge.

GTM wedge. First 10 customers: not bottoms-up — go through the people already scared. Partner with AI-governance consultancies and Big-4 audit practices building AI Act services (they need a tool to produce the evidence), and target model vendors with active EU go-to-market. Lead with "we generate the training-data documentation the Act requires, and a third party can verify it."

Success metric. Number of DBOMs that pass an independent/auditor review + design partners citing it in actual compliance filings. Target: ≥3 DBOMs accepted in a real audit or regulatory submission within the first quarter. That's proof the artifact has authority, which is the entire business.

Two incumbents who'd copy the wedge in 30 days: data-lineage/catalog vendors (e.g. existing MLOps lineage tools) and governance platforms (Credo AI, Holistic AI). Our unfair edge: **independent cryptographic verifiability + a DBOM standard built for the Act**, where they offer self-attested dashboards a regulator can't independently trust. Verifiable-by-a-third-party is the line they can't easily cross.

Aggressive timeline. 48h: sign-and-emit DBOM pipeline on a toy mixed corpus + expert-validation interviews. ~1 week: real integration with one training pipeline producing a verifiable DBOM. ~2

weeks: first design partner generating a DBOM for a real fine-tune, reviewed by a governance practitioner.

Design (Alexis, UX)

Core flow. (1) Point Attest at the sources you ALREADY have — your object store, dataset registry, and a data-loader hook — via a short config; it wraps your ingest, it doesn't replace it. (2) As data enters training, Attest signs each source (hash + license + provenance). (3) It emits a per-model DBOM: a formal, sealed bill of materials listing every source, its license, content hash, and token share. (4) An independent auditor or regulator runs the open verifier, re-hashing each source against the manifest — Attest's servers are never in the loop. (5) The DBOM maps straight to EU AI Act obligations, so 'are we audit-ready?' becomes a status, not a fire drill.

Screens.

- **01 Hero** — Attest wordmark + the source-to-model chain of custody: web scrape → licensed corpus → synthetic, each sealed, ending in a stamped 'DBOM · third-party verifiable' certificate. Key interaction: 'Generate a DBOM.'
- **02 Retrofit setup (the visitor's exact fear)** — an `attest.yaml` that hooks into existing ingest points (s3 bucket, HF registry, dataloader hook) with a 'what you DON'T rebuild' panel + a best-effort-DBOM path for messy history. Key interaction: wrap-don't-refactor, days not a quarter.
- **03 The DBOM artifact** — the product IS this document: a per-model bill of materials rendered as a formal audit certificate (sources, licenses, hashes, token shares, signature seal). Key interaction: export as PDF/JSON.
- **04 Independent verifier (the moat)** — an auditor's view re-hashing each source and returning verified/4-of-4, locally, with no call to Attest. Key interaction: 'verify without trusting us' — the line self-attested dashboards can't cross.
- **05 When lineage doesn't check out (non-happy state)** — a FAILED verification: one hash mismatch (source changed after signing) + one UNKNOWN-provenance source from pre-Attest history. Key interaction: honest gaps surfaced, never a fake green pass.
- **06 AI Act readiness + pricing** — the DBOM mapped to specific obligations (Art. 53 training-data summary, license docs, copyright opt-out, synthetic disclosure) with status; Pilot $5k / Compliance $50k / Enterprise. Key interaction: hand the regulator evidence, not a promise.

UX risks.

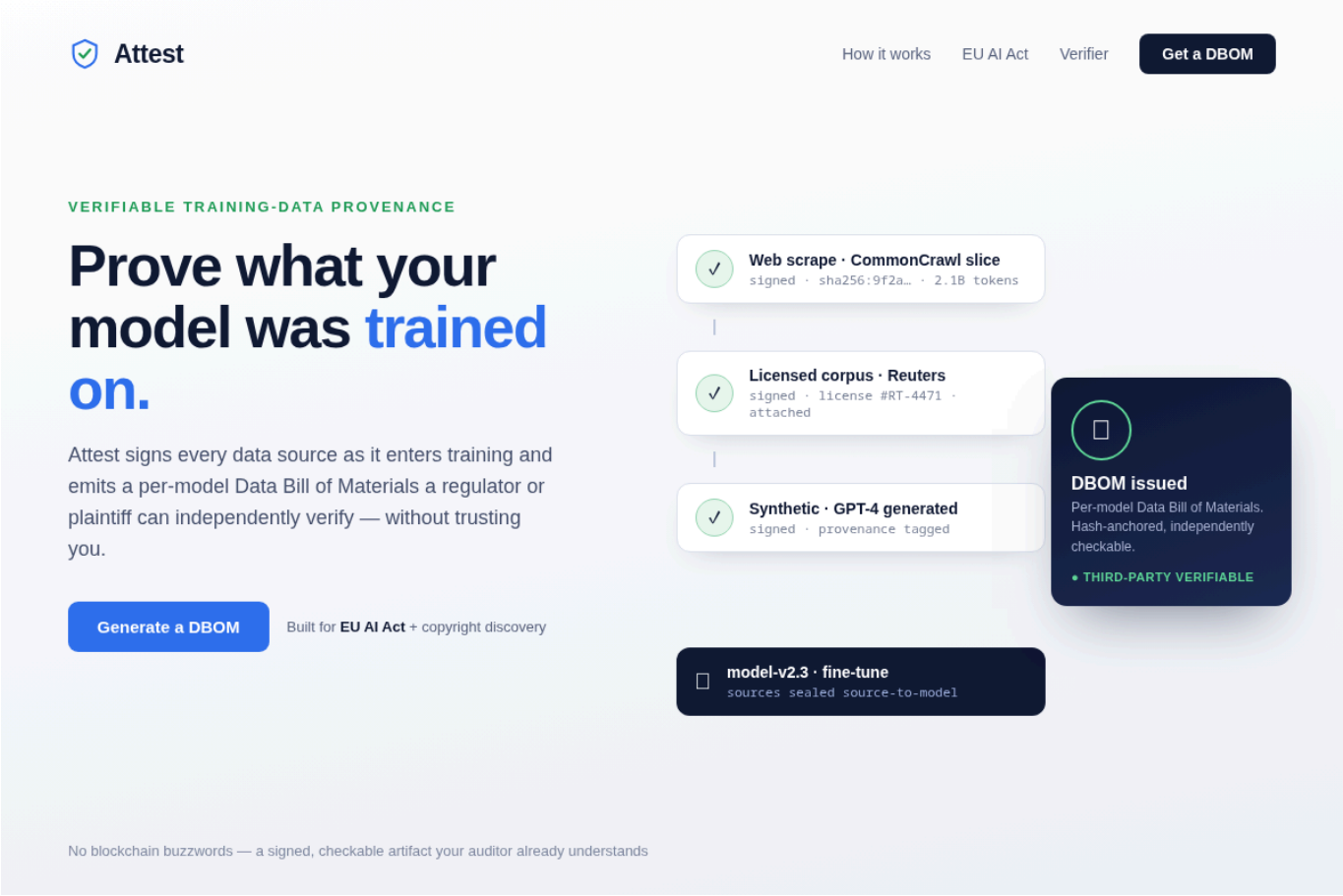
- *Buyers fear a rip-and-replace that blows the compliance deadline.* Mitigation: screen 02 leads with retrofit — config against existing ingest, plus a best-effort DBOM from messy history so you

start compliant, not blocked. The 'what you DON'T rebuild' list answers the objection before it's raised.

- *An artifact nobody independently trusts is worthless.* Mitigation: screen 04 makes third-party verification the centerpiece — the auditor re-computes locally, Attest never in the loop — so the DBOM's authority comes from math, not from our word.
- *A tool that always shows green is the opposite of credible to a regulator.* Mitigation: screen 05 is a deliberate failure state — hash mismatch + unknown provenance shown honestly — proving Attest surfaces coverage gaps instead of faking completeness. Honest amber beats a false pass in an audit.

Visual system. An audit/legal-grade aesthetic, deliberately NOT crypto-bro: clean paper white `#ffffff` / `#f7f8fa` with authoritative deep-navy ink `#101935`, a trust-steel accent `#2f6fed` for signatures/verification, and a verification-green `#1f9d57` for sealed/verified vs. an honest amber `#b5732a` / red `#c0492f` for review/failed. The DBOM and verifier read like an SBOM/SOC2 report a Big-4 auditor already understands — Inter + monospace for hashes. No blockchain visual language anywhere; the credibility cue is the document and the seal, not a chain animation.

Carousel.



Attest

SETUP · RETROFIT · NO REBUILD

Wrap your existing ingest. Don't rebuild it.

Attest hooks in where data **already** enters training — your loaders, dataset registry, or object store — and signs sources as-is. A few integration points and a config, not a new pipeline.

attest.yaml

```yaml
# point Attest at the sources you ALREADY have
sources:

  - type: s3_bucket # your object store
    path: s3://corp-training/raw/

  - type: hf_dataset_registry # existing registry
    ref: reuters-licensed-v3

  - type: dataloader_hook # wrap, don't replace
    module: train.data:get_loader

emit: dbom # sign + emit per-model bill of materials
```

What you DON'T rebuild

- ✓ Your data pipeline — Attest wraps your ingest points
- ✓ Your storage / registry — signs sources in place
- ✓ Your training code — a hook, not a refactor

Messy history? Generate a **best-effort DBOM** from whatever lineage exists today, then tighten coverage going forward — you start compliant, not blocked.

Typical lift: a few integration points + config · days, not a quarter

Attest models / model-v2.3 / Data Bill of Materials

Export DBOM (PDF · JSON)

DATA BILL OF MATERIALS

model-v2.3 · customer-support fine-tune

issued 2026-06-23 · spec: DBOM v1.2 · anchor sha256:4c8e…a019

SEALED

SOURCE	TYPE	LICENSE	CONTENT HASH	TOKEN SHARE
CommonCrawl 2025-04 slice	Web scrape	mixed / review	sha256:9f2a…71c	**61.2%**
Reuters licensed corpus	Licensed	RT-4471 ✓ attached	sha256:1b07…e44	**22.8%**
Internal support tickets	First-party	owned ✓	sha256:cc90…2af	**11.0%**
GPT-4 synthetic Q&A	Synthetic	synthetic · tagged	sha256:7d31…b8e	**5.0%**

Signed by: Acme AI Data Team · key 0x4F…A2
Independently verifiable: re-hash sources against this manifest — no trust in Attest required.

Verify this DBOM →

Attest

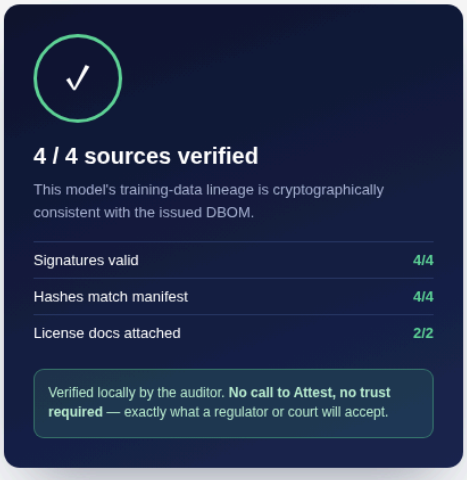
Verifying as **Independent Auditor · KPMG AI Assurance**

Anyone can verify it. Without trusting us.

An auditor or regulator runs the open verifier against the DBOM — re-hashing each declared source and checking signatures. Attest's servers aren't in the loop; the math is. A self-attested dashboard can't do this.

Verifying `DBOM · model-v2.3 · anchor sha256:4c8e…a019` — ● checking 4 sources

✓	**CommonCrawl 2025-04 slice** `recomputed sha256:9f2a…71c`	`manifest match`	VERIFIED
✓	**Reuters licensed corpus** `license RT-4471 signature valid`	`sig + hash match`	VERIFIED
✓	**Internal support tickets** `recomputed sha256:cc90…2af`	`manifest match`	VERIFIED
✓	**GPT-4 synthetic Q&A** `provenance tag valid`	`manifest match`	VERIFIED



Open-source verifier · runs anywhere · the artifact stands on its own

It won't fake a green check. Gaps are surfaced, not hidden.

A provenance tool that always passes is worthless. When a source can't be verified — tampered, missing, or unknown lineage from your messy history — Attest flags it honestly, so you fix the real gap before an auditor finds it.

Verifying `DBOM · model-v1.8 · legacy import` ⚠ 2 ISSUES FOUND

✓	**Reuters licensed corpus** `sig + hash match`	VERIFIED
✕	**CommonCrawl slice** `recomputed sha256:3b1f… ≠ manifest 9f2a…`	HASH MISMATCH source changed after signing
?	**Scraped forum data (2024)** `no signature on record`	UNKNOWN pre-Attest, lineage missing
✓	**Internal tickets** `owned · hash match`	VERIFIED

⚠

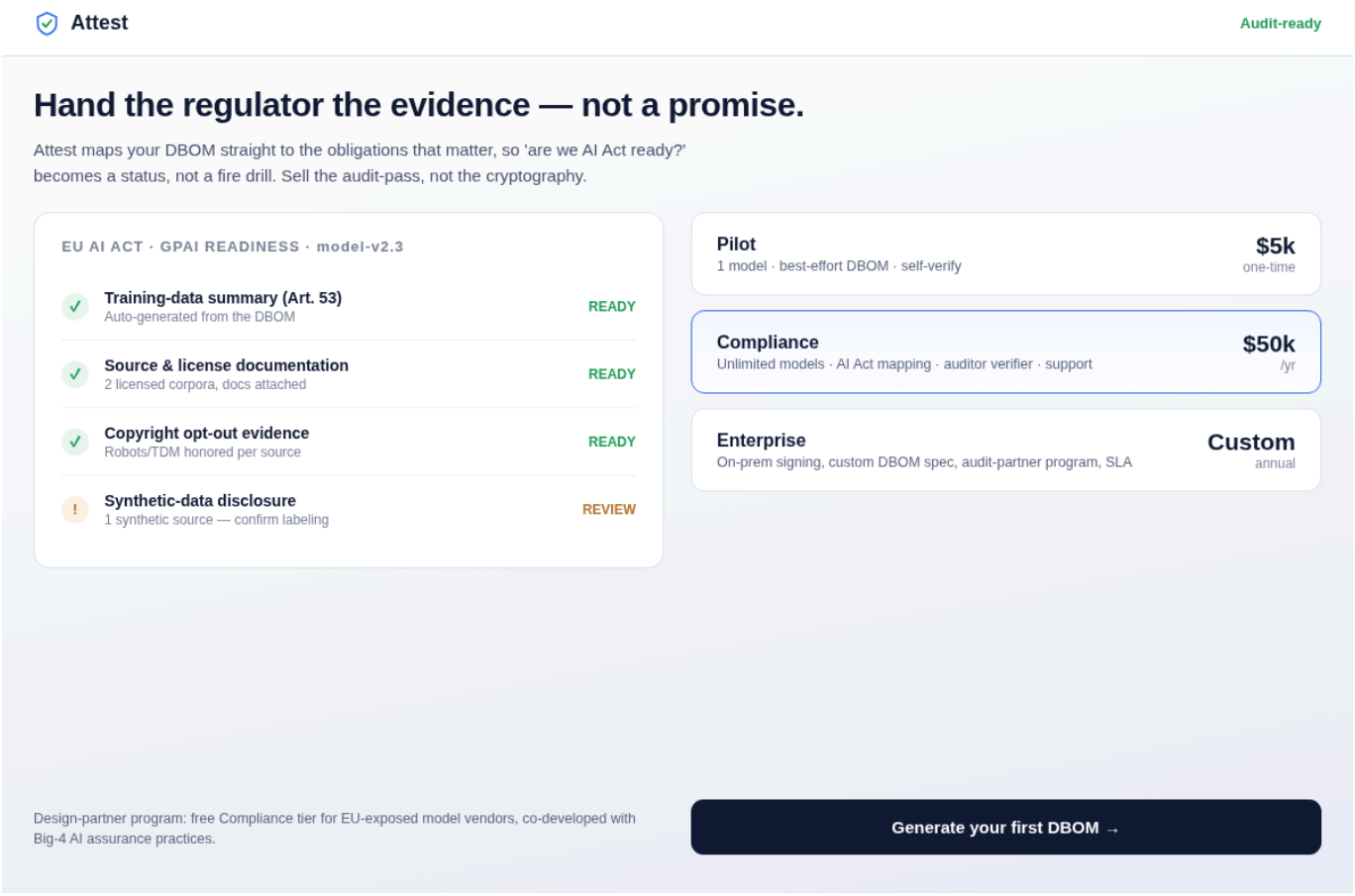
Incomplete — 2 / 4 verified

This DBOM is not yet audit-ready. Attest tells you exactly where the lineage breaks.

Verified	2
Hash mismatch	1
Unknown provenance	1

Next: re-sign the changed CommonCrawl slice, and mark the 2024 forum data as best-effort/unverified in the DBOM — an honest gap an auditor can assess beats a false pass.

Honest coverage > fake completeness — the only kind a regulator trusts



Engineering

Stack:

- **Language: Python** for the SDK/agent — that's where every training pipeline (PyTorch, HF `datasets`, custom loaders) already lives, so we hook in without a language boundary. A small **Go** binary for the standalone open-source verifier so auditors can run it anywhere with zero Python env.
- **Integration:** thin **ingest wrappers** — an S3/object-store reader, a HuggingFace dataset-registry hook, and a `dataloader` decorator. You point `attest.yaml` at the sources you already have; we wrap ingest, we don't replace it. This is the "retrofit, not refactor" promise made literal.
- **Signing/crypto: content hashing (SHA-256)** per source + **Ed25519** signatures over the manifest, anchored to a Merkle root so the whole DBOM has one tamper-evident anchor hash. Standard, boring, auditable primitives — deliberately **not** a blockchain (a blockchain adds nothing here and scares the buyer).
- **DBOM format:** a versioned **JSON manifest** (sources, licenses, hashes, token shares, transforms, signature) with a deterministic canonical serialization, plus a PDF render for humans. Modeled on **SBOM/SLSA** so it reads like evidence a Big-4 auditor already understands.

- **Verifier:** open-source, runs locally, re-computes each source hash against the manifest and checks signatures — **no call to Attest**. Its independence IS the moat; a self-attested dashboard can't cross that line.
- **Backend: FastAPI + Postgres** for the signing service, key management, DBOM storage, and the AI Act obligation mapping; **on-prem signing** image for Enterprise (regulated data never leaves their boundary).

Architecture: Data enters training through your existing loaders → Attest's ingest wrapper hashes + signs each source as-is and records license/provenance → at model seal it emits a DBOM (Merkle-anchored JSON + PDF). Separately and offline, an auditor runs the open verifier: re-hash declared sources → compare to manifest → check Ed25519 sigs → output verified / mismatch / unknown per source. The signing path and the verification path never share a server — that separation is what makes the artifact trustworthy.

Data model: `source(id, type, uri, content_hash, license, provenance, token_share)` · `dbom(model_id, spec_version, merkle_anchor, signature, signer_key)` · `verification(dbom_id, source_id, recomputed_hash, result=verified|mismatch|unknown)` · `obligation(dbom_id, art_ref, status)` mapping sources to specific AI Act articles. The `verification` records + the accepted-audit history are the compounding asset.

Hard parts / risk (the 2 that actually matter):

1. **The artifact has to be credible to someone with subpoena/fine power — tech nobody trusts is worthless.** De-risk: independent local verification (auditor re-computes, we're not in the loop), an SBOM/SLSA-shaped format auditors already recognize, and an explicit **honest failure state** — when a source's hash doesn't match or its lineage is pre-Attest/unknown, the verifier shows mismatch/unknown, never a fake green. A tool that always passes is the opposite of credible; surfacing gaps is the feature.
2. **Retrofit onto a messy pipeline without blowing the compliance deadline.** A rip-and-replace is self-defeating. De-risk: wrap existing ingest points via config (a few hooks, not a new pipeline), plus a **best-effort/retro-DBOM** that reconstructs lineage from manifests, storage metadata, and logs you already have — and flags the gaps as a punch-list — so you're defensibly documented for what you can prove now instead of blocked on everything.

Build plan:

- **48h cut-corner (proof):** the clickable prototype below + a sign-and-emit pipeline on a toy mixed corpus, put in front of governance/legal practitioners to confirm a DBOM would stand up — the Week-1 question is "will a credible third party accept this?", which is a validation question, not a code question.

- **1-week MVP:** real integration with ONE training pipeline (HF + S3) producing a verifiable DBOM, and the open verifier re-hashing it end-to-end.
- **2-week:** first design partner generating a DBOM for a real fine-tune, reviewed by a governance practitioner, mapped to AI Act obligations.

Cut-the-corner version: what ships in 48h is the prototype below — open the **Independent verifier**, hit ▶ **Run verification** to watch each source re-hash and seal 4/4, then hit ⚠ **Tamper a source** and watch the same engine honestly flip it to HASH MISMATCH and break the seal (no fake green). The DBOM exports, the failure screen lets you re-sign / mark best-effort to recover, and the AI Act screen resolves obligations to readiness. Proves both the verifiable-artifact wedge and the honest-gaps behavior with zero infra.

 Open the clickable prototype

Plan

- **Pricing:** Open verifier free -> Team SaaS $50K/yr -> Enterprise on-prem custom
- **Timeline:** MVP: sign-and-emit DBOM + open verifier on toy corpus, validated by practitioners in 48h -> Launch: real integration on one pipeline producing a verifiable DBOM in ~1 week -> GTM: design-partner DBOM reviewed by a governance practitioner via audit/governance firms in ~2 weeks
- **Team:** Sam 5d signing SDK + open verifier, Alexis 2d audit-console UX, Mark 2d GTM + practitioner validation; security/crypto review and a governance advisor before any customer DBOM.
- **Build cost:** $10-15K for the 48h sign-and-emit proof + practitioner validation + 1-week pipeline integration; the Week-1 spend is mostly expert review, not code.
- **First milestone:** Week-1: a real DBOM for a small mixed-corpus model that 3 governance/legal practitioners confirm would stand up in an AI Act audit or copyright discovery.
- **VC fundability:** Genuinely vc-fundable: regulation manufactures demand on a deadline, enterprise ACVs clear a $1B+ SAM, and a regulator-recognized-standard moat compounds with every accepted audit.