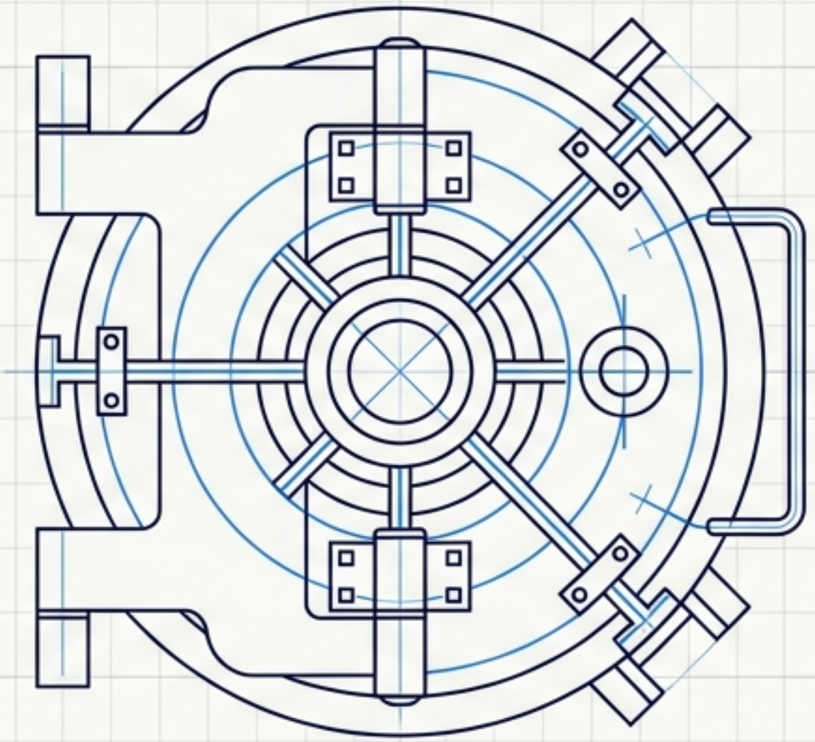
Engineering Deterministic Compliance



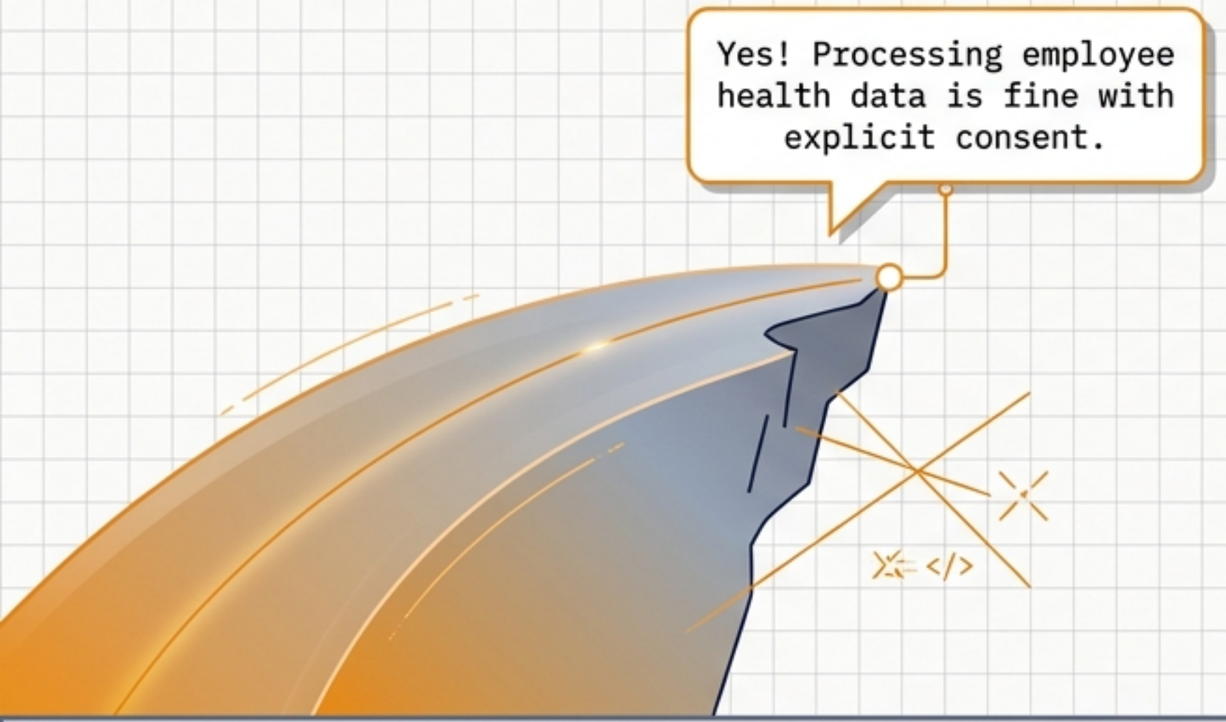
Architecture for legally verifiable AI agents.

STATUS: PRODUCTION | SPEC: KCP (APACHE 2.0) | IMPLEMENTATION: MYNDER

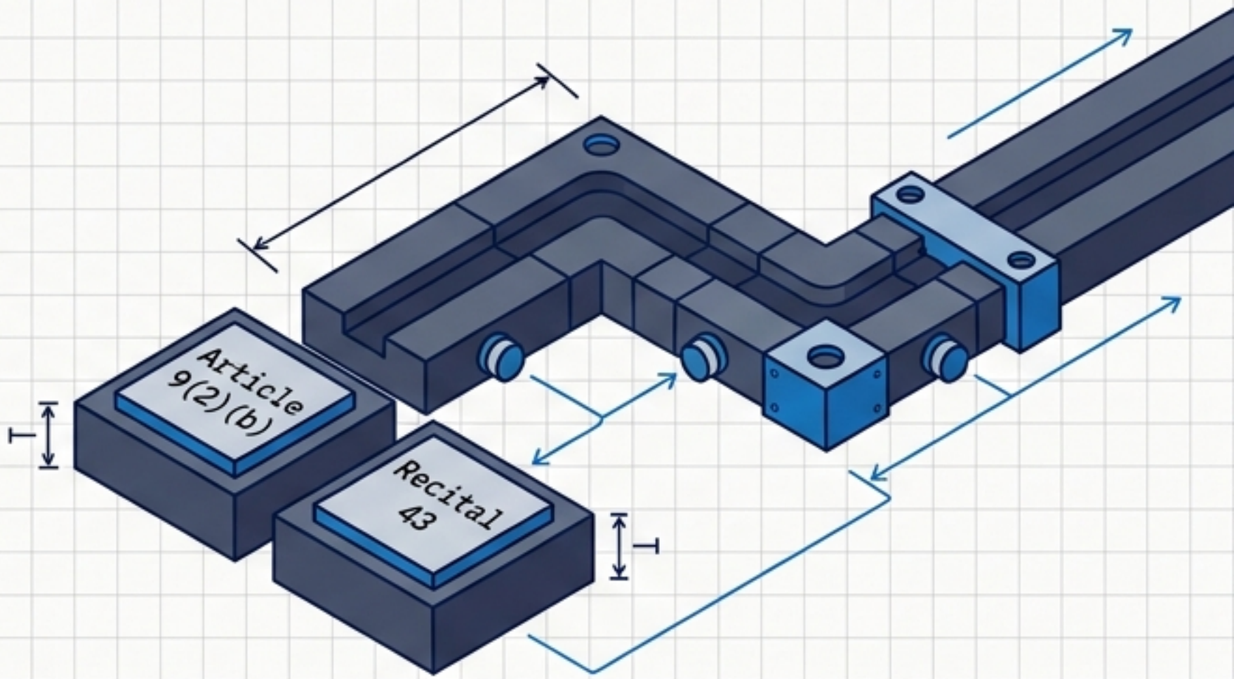
NotebookLM

The failure mode of an LLM is not silence. It is confidence.

The Hallucination Trap



The Reality



Why training data fails

An LLM trained on the internet assumes “consent” (Article 6) solves everything. It misses the nuance that employer-employee power imbalances presume consent is not freely given (CJEU C-61/19), requiring an employment law basis instead.

The model doesn’t know this; it just predicts fluent text.

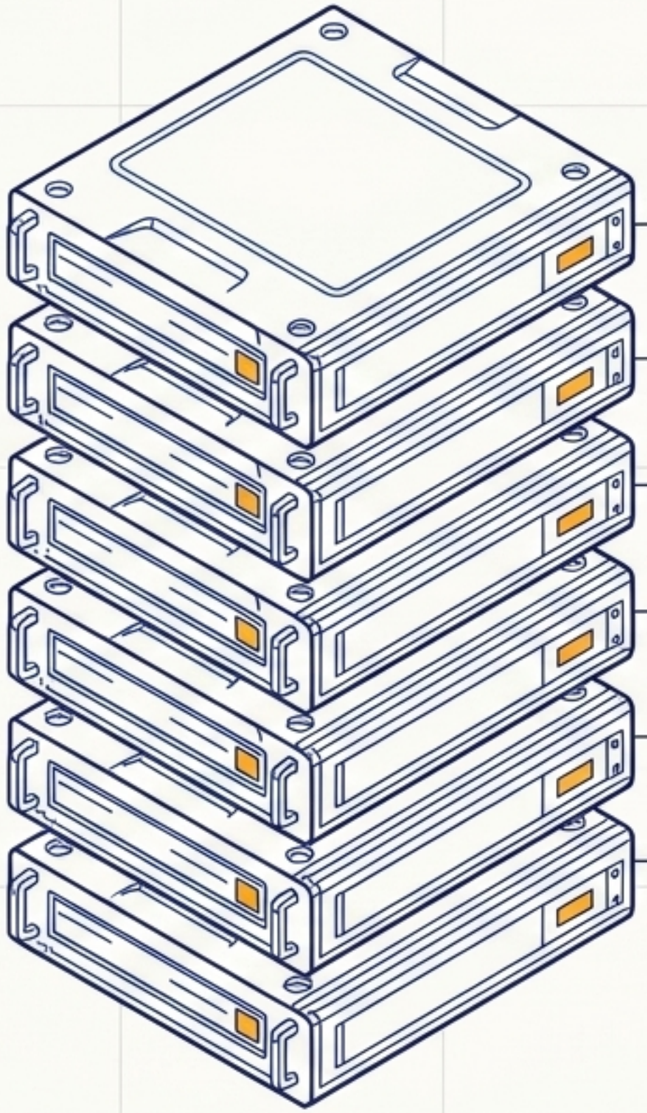
NotebookLM

Vibes versus verification

	Standard AI Agent	Deterministic Compliance Agent
Knowledge Source	Training Data Summaries & RAG	✓ Authoritative Fragment Routing
Output Basis	Probabilistic Guesswork	✓ Deterministic Evaluator Functions
Failure Mode	Confident Hallucination	✓ Verifiable Rejection
Audit Trail	We think it read the right version	✓ Ed25519 Detached Signatures

NotebookLM

The 6-layer compliance engine



LAYER 06 Continuous Benchmarks

LAYER 05 Expert Lenses

LAYER 04 Evaluator Functions

LAYER 03 Agent Integration Sidecar

LAYER 02 Extraction Pipeline

LAYER 01 Verifiable Manifests (KCP)

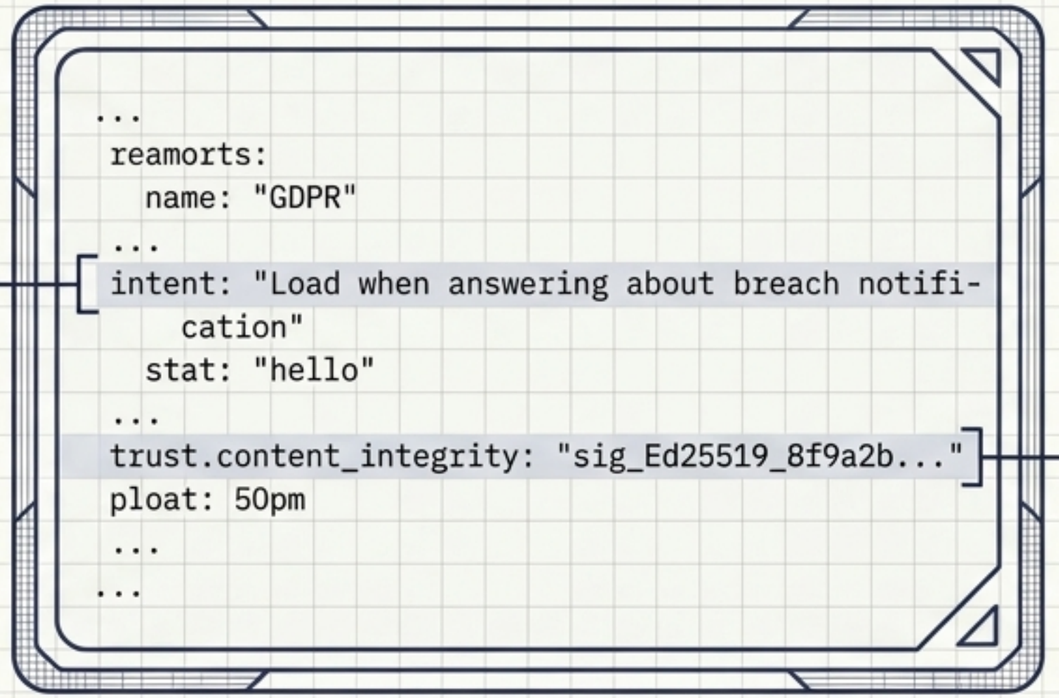
LLMs cannot internalize an 88-page, 73,000-token regulation like the GDPR.

We must build an external determinnistic engine that controls exactly what the model sees, how it assesses data, and how it is verified.

NotebookLM

Layer 1: Cryptographically verifiable fragments

Routing Logic. A plain-language trigger. Prevents loading all 99 articles.

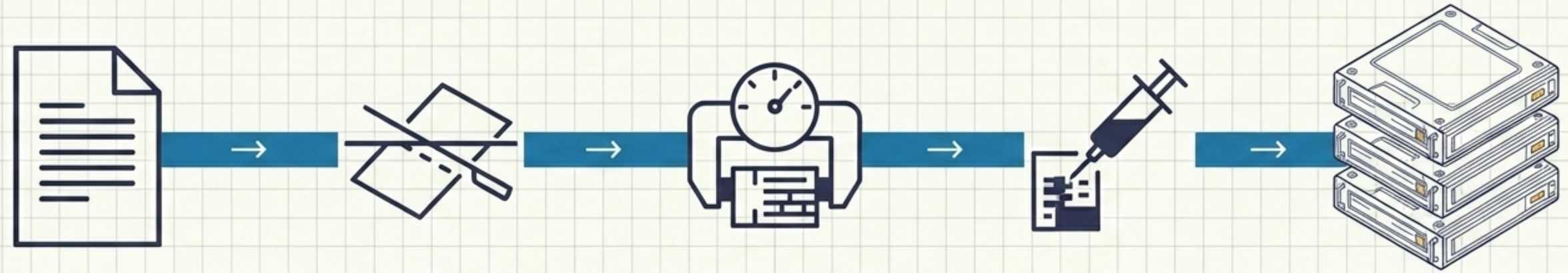


The Audit Trail. An Ed25519 detached signature.

Regulators do not accept “we think the agent read the right version.” Signatures mathematically prove the agent read this specific version of the law, published at this timestamp, without tampering.

NotebookLM

Layer 2: The extraction pipeline



Source PDF

Structural Slicer.
Detects jurisdictional markers (EU ‘Article N’, Nordic ‘Kapittel N’).

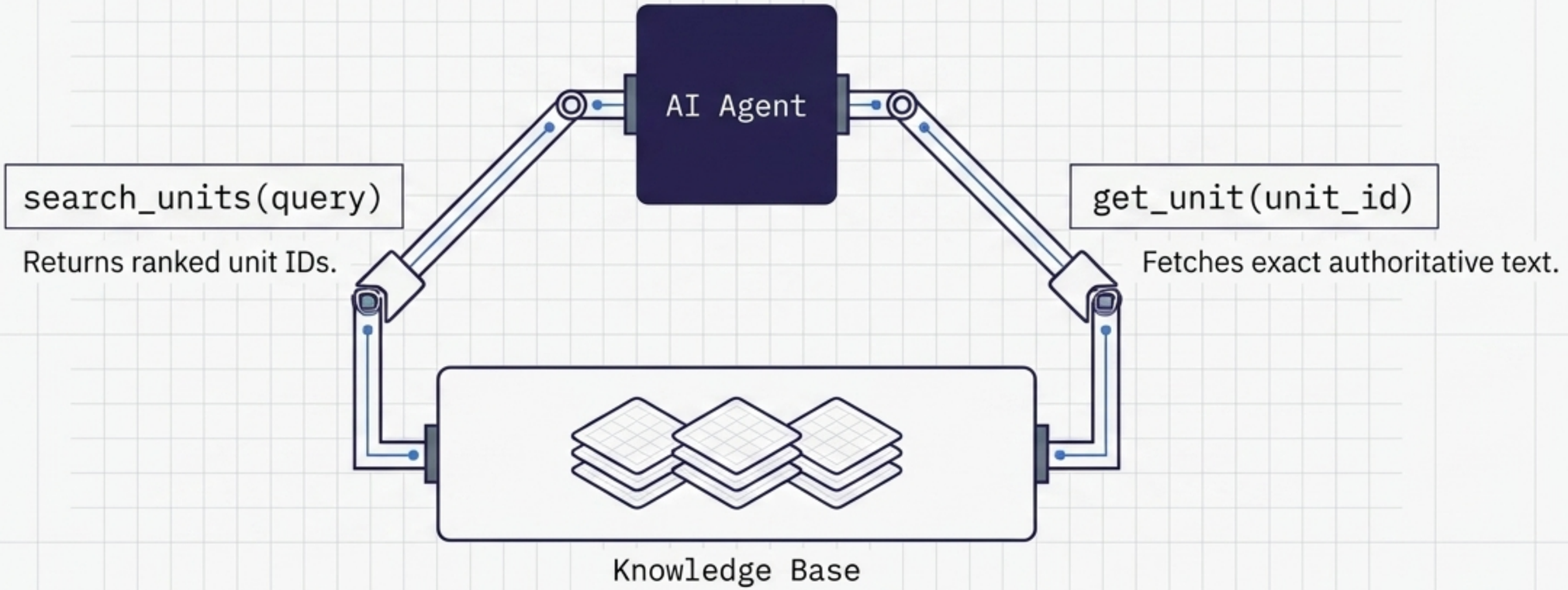
Tokenizer.
Calculates token budgets per chunk.

Metadata Annotator.
Injects `intent_fn` domain knowledge.

> Example: GDPR Article 28 is automatically tagged with `intent: Mandatory` DPA contract clauses... `CORE` article for vendor assessment.

NotebookLM

Layer 3: The integration sidecar

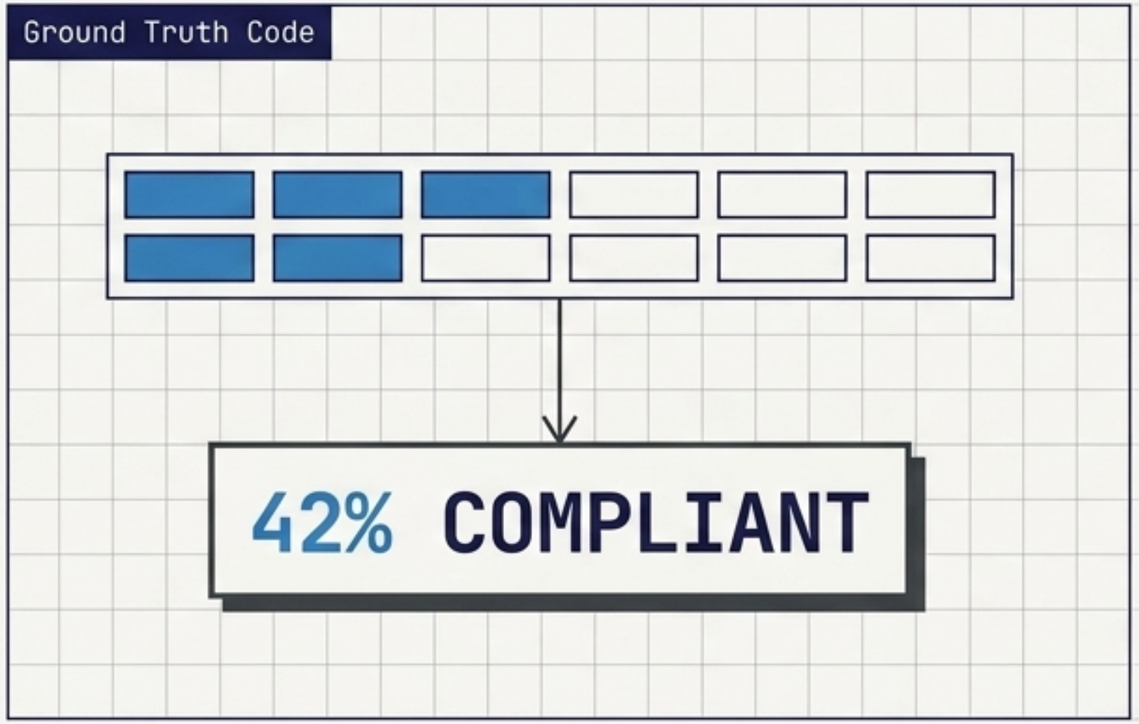


A dramatically reduced integration surface.

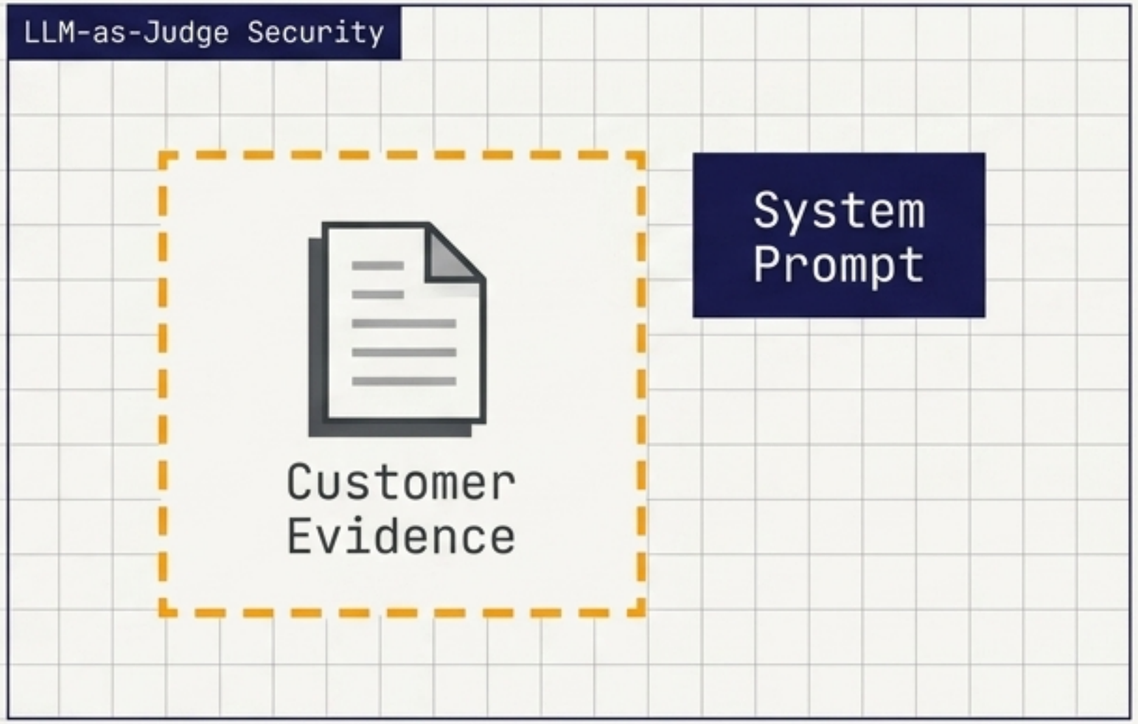
When asked about DPIAs, the agent searches, fetches Article 35, and reads the actual mandatory triggers (large-scale special category, systemic evaluation). It reads the law, not its training memory.

NotebookLM

Layer 4: Deterministic evaluators



Write one evaluator per obligation (Art. 6, Art. 32, etc.). If the evaluator calculates 42% compliance, but the agent says “looks good,” the agent is objectively wrong.

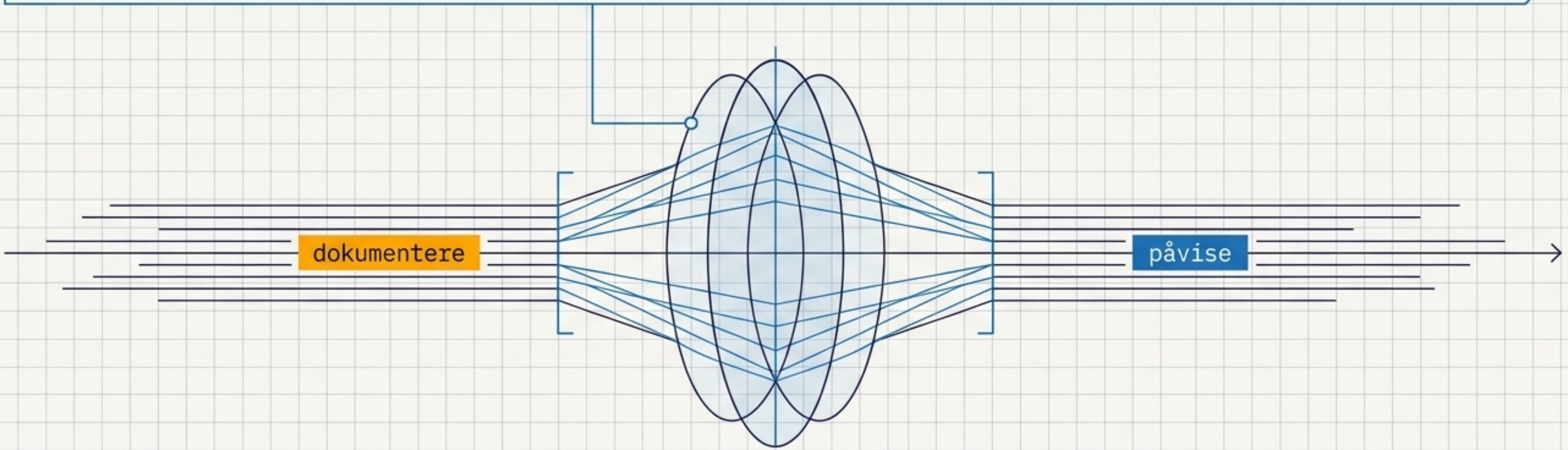


When evaluating subjective requirements (like Art. 13 transparency), prompt injection defense is mandatory. The document is fenced as evidence, never executed as instructions.

NotebookLM

Layer 5: Institutional expert lenses

Encoding senior specialist judgment into versionable YAML.



The Problem

Legal terminology is strict. “To document” (dokumentere) is not the legally distinct standard of “to demonstrate/prove” (påvise) under Article 5(2).

The Solution

The Lens file catches terminology errors, wrong classifications, and missing obligations during every review pass. It transforms institutional knowledge from a senior expert’s head into a consistent, automated asset.

NotebookLM

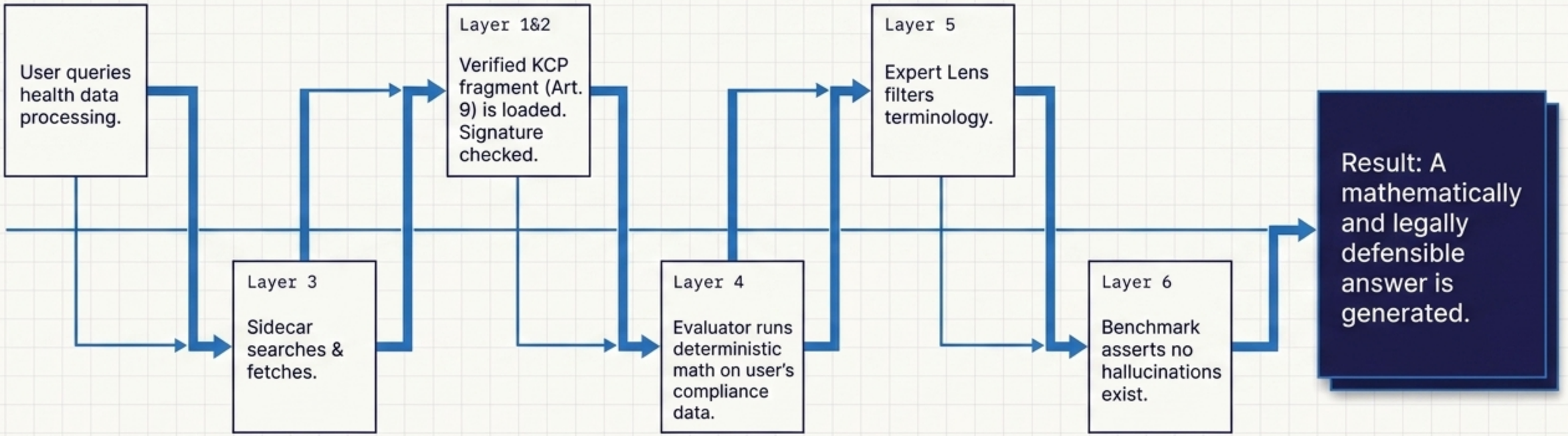
Layer 6: Continuous benchmarking

expected_elements	prohibited_elements
Data erasure must happen "without undue delay".	LLM invents "30-day deadline" for erasure.
	LLM states "Report all breaches to data subjects" (Misses that only high-risk breaches trigger Art. 34).

The **safety net**. When you change models, update prompts, or when legislation shifts (like NIS2 implementing acts), the benchmark suite immediately catches regressions.

NotebookLM

The deterministic data flow



NotebookLM

What this architecture prevents

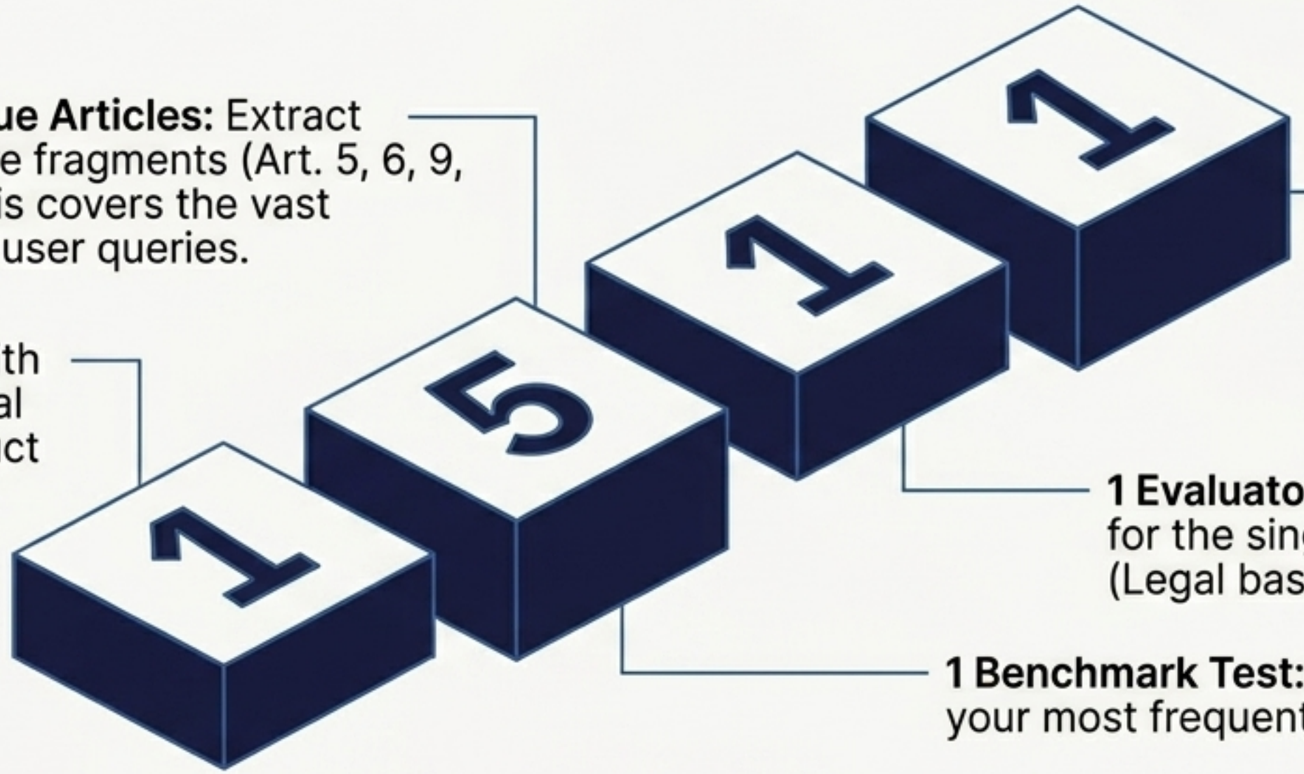
Missing Layer	Silent Failure ✗	System Catch ✓
Missing Knowledge	Agent approves employee health data based on consent.	Routed to Art 9(2)(b) employment law basis.
Missing Evaluators	Agent says ‘compliance looks good’ just because a privacy notice exists.	Code calculates actual score: 33% (no DPO, no DPIAs).
Missing Signatures	Fictional Art. 6(1)(g) injected by a malicious edit.	Ed25519 verification fails; content rejected.
Missing Lenses	Platform uses incorrect evidentiary standard (‘document’).	Lens forces correction to ‘demonstrate’.

NotebookLM

Starting small: The 1-5-1-1 protocol

5 High-Value Articles: Extract just the core fragments (Art. 5, 6, 9, 28, 32). This covers the vast majority of user queries.

1 Regulation: Start with the single most critical regulation your product handles (e.g., GDPR).



1 Benchmark Test: Build one test to catch your most frequent hallucination.

1 Evaluator: Write deterministic code for the single most queried obligation (Legal basis coverage).

1 Benchmark Test: Build one test to catch your most frequent hallucination.

"Agents are only as good as what they know. Give them authoritative knowledge, then build the systems to prove they know it."

NotebookLM