SUMMARY

# Provably Private AI as Infrastructure

Zero-retention AI + prompt-driven PII/PHI de-identification: the baseline layer for any organization that handles sensitive text

**Elixir Technologies** | October 2026

Read the full white paper →

elixir.com/provably-private-ai-white-paper

## The Problem

Every organization runs AI somewhere. The question is no longer *whether* to use it but *what happens to the data you feed it.* Most AI tools **store** your content, make privacy depend on a **toggle or tier**, and offer a **promise** rather than a property. Promises break:

- A 2023 bug exposed ChatGPT conversation titles and payment data, because the data was stored.
- Across 2025 to 2026, "share" features quietly published ~100,000 ChatGPT chats, hundreds of thousands of Grok chats, and Claude chats containing medical records and credentials to search engines; ~143,000 conversations ended up on the public Internet Archive.
- In 2025, a court ordered OpenAI to retain **deleted** ChatGPT logs and produce 20 million of them.

Every one of these had the same precondition: **the data was being kept.**

## The Idea

Two capabilities, combined, belong in the baseline stack:

### 01 Provably private, zero-retention AI

**WHAT IT MEANS** Content is processed in memory and never stored; privacy is an architectural fact, not a setting.

**WHY IT MATTERS** Nothing to breach, index, subpoena, or misconfigure. "We *can't*," not "we won't."

### 02 Prompt-driven PII/PHI de-identification

**WHAT IT MEANS** Find the identifiers in a document and produce a de-identified version, on that same private path.

**WHY IT MATTERS** Turns "we hold nothing" into "and we can still safely clean your most sensitive content."

**Why together:** cleaning sensitive data normally means handing it to another tool that keeps a copy. On a zero-retention path, the sensitive input is used to produce the de-identified output and then discarded, with no second copy anywhere.

elixir

## What It Is, and What It Isn't

### It is

- Client-side document storage
- In-memory AI processing with immediate discard
- Content-free logging (the log layer emits only operational fields, so content cannot be written even by mistake)
- Verifiable by inspection, not trust
- A posture where Elixir cannot see your content (nothing writes it to a store or log, the runtime role cannot write to storage or logs, and the crash-reporter captures no stack-frame variables), verifiable from source, dependency pins, and IAM policy
- A hybrid de-identification pipeline (deterministic detection + model pass + verification) that returns a de-identified draft **and a coverage report**

### It is not

- A compliance certification. HIPAA Safe Harbor requires a human's "no actual knowledge" judgment. The tool makes a knowledgeable person faster and better-documented; it does not replace them.
- A claim of "unhackable."

## One Standard, Universal Pattern

The worked example is **HIPAA Safe Harbor** (remove 18 enumerated identifiers; confirm nothing remaining identifies the individual). The same find → strip → verify → confirm pattern serves **GDPR** (data minimization), **CCPA/CPRA**, **GLBA**, and the **EU AI Act** (transparency obligations live since Aug 2026; high-risk obligations phasing in through Dec 2027 and Aug 2028). An architecture that holds no sensitive data is durable across all of them.

## Why It Belongs in the Baseline Stack

**Removes the adoption blocker**
Legal and security can approve AI they can verify.

**Shrinks breach & subpoena surface**
The cheapest data to protect is the data you don't keep.

**Simplifies vendor assessment**
No stored content makes most of the questionnaire not-applicable; months to days.

**Regulation-ready**
One architectural posture instead of a per-rule scramble.

## In Production Today

Elixir ships this architecture across writing, image, and voice tools, and has built a dedicated de-identification capability into **Elixir Muse**: point it at a document, get a HIPAA Safe Harbor-targeted draft plus a coverage report, over a zero-retention path, with nothing stored on Elixir's servers.

*Incident references drawn from third-party reporting and primary court filings; cited and dated in the full technical white paper.*

**THE BOTTOM LINE**

A court can compel a vendor to retain and produce data it holds, even data a user deleted. ***No court can compel production of data that was never kept.*** That is the difference between a promise and a property.