WHITE PAPER

Zero-Trust UI Documentation for SOC 2 & HIPAA

Replacing manual screenshots with automated, sandboxed, audit-ready evidence

DocuAgent AI
Prepared for compliance, security, and engineering leadership

Executive Summary

Ask most engineering or product teams what SOC 2 or HIPAA prep actually looks like, and you'll hear some version of the same story: weeks spent taking screenshots by hand, blacking out customer data, and dropping images into tickets so an auditor can see how the application behaves.

It's a slow process, and it's also a risky one. Screenshots taken in staging or production can catch real Protected Health Information (PHI) or Personally Identifiable Information (PII) without anyone noticing, and once that data sits in a wiki page or a shared drive, it's out of anyone's control.

DocuAgent AI takes a different approach. It's an automated, sandboxed engine that crawls, verifies, and documents production applications on an ongoing basis, and it's built so that sensitive fields never leave the client's own security boundary in the first place.

The Problem: Manual Documentation Breaks Compliance

Most teams run into the same three issues when they try to document user flows for auditors or for customer success:

- **Sensitive data ends up in the documentation.** It's common for someone recording a walkthrough or grabbing a screenshot to capture real user data in the process — names, Social Security numbers, medical record numbers, financial details. It usually isn't intentional, but that doesn't make it less of a problem.
- **Screenshots don't hold up as evidence.** A screenshot pasted into a wiki page tells an auditor almost nothing on its own. There's no way to verify when it was taken, which version of the code produced it, or whether anyone edited it afterward.
- **The UI moves faster than the documentation.** With most teams shipping changes daily, a compliance manual is often out of date before an auditor even opens it. That gap between what's documented and what's actually running is exactly what SOC 2 controls CC6.1 and CC7.1 are meant to catch.

The Zero-Trust Architecture: How It Works

Instead of relying on someone to manually gather screenshots, DocuAgent AI runs the whole thing as an isolated, continuous pipeline. A scan moves through five stages before it's ever stored anywhere, and sensitive data gets filtered out well before that point:

Stage	Pipeline Step	What Happens
1	Application UI	The target web application is loaded exactly as a user would encounter it.
2	gVisor / MicroVM Isolation	Execution occurs inside an isolated, disposable sandbox with no persistent state.

Stage	Pipeline Step	What Happens
3	DOM & Vision AST Parsing	The page structure is parsed both semantically (DOM) and visually (rendered layout).
4	Local PII / PHI Masking Engine	Sensitive fields are detected and redacted before anything leaves the sandbox.
5	Cryptographic SHA-256 Evidence Vault	A tamper-evident, signed record of the capture is stored for audit review.

1. MicroVM Sandbox Isolation

Each scan runs inside its own gVisor or Firecracker microVM. The engine interacts with the application headless, and once the run finishes, that environment is torn down completely — nothing persists between scans.

2. Client-Side PII/PHI Sanitization

Before a screenshot or any visual metadata reaches an LLM for guide generation, or gets written to storage, it passes through DocuAgent AI's local AST and vision engine. That engine finds sensitive elements — input fields, table cells, modal overlays — and redacts them right at the DOM layer, using regex and CSS selector rules you define.

3. Immutable Execution Ledgers

Each run produces a signed evidence bundle rather than a loose file. Every scan is hashed with SHA-256 and added to an append-only log, so an auditor can see exactly when a route was captured, what controls were in place, and how the flow played out — without having to take anyone's word for it.

Mapping to Compliance Controls

Here's how these capabilities line up against the specific SOC 2 and HIPAA control objectives that come up most often in audit findings tied to UI documentation:

Compliance Framework	Control Objective	How DocuAgent AI Satisfies the Control
SOC 2 Type II	CC6.1 / CC6.8 — Infrastructure & Access Boundaries	Automatically documents access control flows and boundary UI elements inside gVisor sandboxes with zero manual data exposure.
SOC 2 Type II	CC7.1 — Infrastructure & Change Management	Detects layout and navigation drift across releases, generating time-stamped proof of UI state changes.
HIPAA	164.312(a)(2)(i) — Unique User Identification & Masking	Redacts all PHI/PII at the DOM layer before screenshot persistence or model processing.

Compliance Framework	Control Objective	How DocuAgent AI Satisfies the Control
HIPAA	164.312(b) — Audit Controls	Produces cryptographically signed JSON execution logs for every application route crawl.

Getting Started

Rolling out Zero-Trust UI documentation takes three steps:

1. **Deploy the isolated runner.** Run it inside your private VPC or Kubernetes cluster using the local Docker or microVM profile.
2. **Define your masking rules.** Configure selector patterns for high-risk inputs, such as #ssn, .patient-record, or [data-pii="true"].
3. **Wire it into CI/CD.** Trigger docuagent crawl on every production deployment to keep master guides current and generate signed compliance bundles for your audit vault along the way.

Where DocuAgent Fits Into Your Stack

DocuAgent is meant to sit inside the tools your team already has open, rather than adding one more dashboard to check. A few of the integration points that come up most often:

- **MCP server support.** Connect DocuAgent to any MCP-compatible client, so it can be called directly from an assistant or an internal tool instead of running as a standalone app.
- **GitHub Action PR diff comments.** Every scan can post its findings straight into the pull request that triggered it, so reviewers see compliance issues right next to the code changes that caused them.
- **Splunk and Datadog webhooks.** Evidence events and violation alerts stream into whichever SIEM or observability platform is already in place, rather than living in a separate silo only the compliance team checks.
- **Local vLLM/Ollama fallback.** When a scan needs model inference and sending data to an external API isn't the right call, DocuAgent can fall back to a locally hosted vLLM or Ollama instance instead.

See a Sample Evidence Bundle

The easiest way to understand what DocuAgent actually produces is to look at one. A sanitized sample Compliance Evidence Bundle is available to browse directly on docuagent-ai.com — no account or sales call required. It's a real .zip export, built against a demo application, with the same step-pins, redaction, and signed ledger structure you'd get from a production scan.

Inspect a sample bundle at docuagent-ai.com, or browse the rest of this series and full integration guides at docuagent-ai.com/resources/whitepapers and docs.docuagent-ai.com.