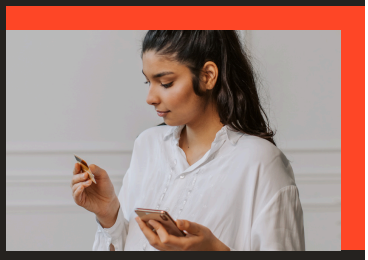
55 % of enterprises have paused or killed GenAI projects over fears of sensitive-data leakage, privacy lapses, and security gaps.



The Secure Chat Stack: A Practical Blueprint for Private AI Inference

Design, deploy, and scale private GenAI tools without leaking a single token of sensitive data.

The shift toward internal AI inference is accelerating. But most deployments fail before they start as they're blocked by IT, compliance, or security due to unresolved questions about control, leakage, and governance.

Enterprise teams are discovering that simply switching on GPT-4 doesn't solve for these concerns. In fact, it often amplifies them. Employee conversations containing PII, source code, and confidential strategies can leak into public models (and already has done so).

This guide offers a practical framework to launch private GenAI tooling without compromising on data boundaries or auditability. It outlines a full-stack approach: sealed inputs, private model serving, filtered outputs, and continuous traceability.

This playbook is for enterprise platform teams standing up internal chat assistants across legal, finance, operations, and product. Use it to align stakeholders, validate architecture choices, and move from pilot to production with confidence.

Book a fifteen minute consult and turn policy bottlenecks into cleared lanes.

What Breaks in Public LLM Interfaces

Most off-the-shelf AI chat tools prioritize ease of use over enterprise control. Here's what breaks when these tools are applied to regulated or sensitive workloads:

- **Leakage risk:** employees paste PII, client data, or unreleased code into an interface backed by opaque APIs
- **Storage ambiguity:** unknown prompt retention, unclear model feedback loops, and no user-level logging
- **Compliance surface:** chat histories become audit liabilities across SOX, HIPAA, FINRA, and internal IP policies

Want to learn more?
Feel free to reach out to us at **hello@sindri.app** and let's chat!

The 7-Step Deployment Blueprint

Step 1: Define Access Roles and Data Classes

Objective: *Establish clear boundaries for who can access what data through the AI interface.*

What it is

A cross-functional inventory of data classes, residency mandates, and downstream sharing rules.

Implementation:

- **Map user groups** by function (legal, finance, engineering, sales)
- **Tag data classes** with sensitivity levels (public, internal, confidential, restricted)
- **Create RBAC policies** linking users to allowable data access
- **Define query scopes** per role to prevent privilege escalation

Action Checklist:

- [ ] Document all user personas and their data needs
- [ ] Classify existing data repositories by sensitivity
- [ ] Design role-based prompt filtering rules
- [ ] Test access controls with sample queries

Ask Your Vendor:

- "Does this pipeline support per-user query logging and scoped embeddings?"
- "Can I restrict model access based on data classification tags?"
- "How do you prevent privilege escalation in multi-tenant scenarios?"

The "Shadow IT" Problem in AI

42% of companies report employees using unauthorized AI tools with work data (Source: Salesforce). When organizations delay formal AI rollouts, employees bypass security controls entirely. That marketing intern copying customer data into ChatGPT? That's a GDPR violation waiting to happen. Clear RBAC policies from day one prevent shadow AI adoption by giving employees sanctioned, role-appropriate access to AI capabilities.

Step 2: Build a Sealed Prompt-Input Layer

Objective: *Isolate user prompts from production systems and eliminate data persistence risks.*

Implementation:

- **Proxy prompt capture**: Forms capture inputs without local storage
- **Write-only ingestion**: Prompts go directly to secure processing
- **Remove temp files**: No intermediate caching or logging
- **Validate before processing**: Check prompts against policy before model access

Quick Win: Deploy a simple proxy form that validates prompts against a whitelist of approved topics before sending to the AI model.

Step 3: Connect Your Own Vector DB (with Filters)

Objective: *Maintain control over knowledge base without third-party indexing.*

Implementation:

- **Private embeddings**: Generate and store vectors in your infrastructure
- **Policy-based retrieval**: Apply filters based on user roles and data classification
- **Audit query paths**: Log what documents are retrieved for whom
- **Context isolation**: Prevent cross-contamination between user sessions

Action Checklist:

- [ ] Deploy vector database in your environment
- [ ] Configure policy filters for document retrieval
- [ ] Implement audit logging for all queries
- [ ] Test context isolation between users

Architecture Flow:

Step 4: Host the LLM Model in an Attested Enclave or Zero-Data API

Objective: *Ensure model inference happens in a verified secure environment.*

The choice of where to run your LLM models is critical. It determines whether your entire security architecture holds or fails. Traditional deployments force a choice between cloud convenience with security questions, or air-gapped control with operational complexity.

Implementation Options:

Option A: Self-Hosted with TEE

- Deploy models in Trusted Execution Environments (Intel SGX/TDX, AMD SEV)
- Full control over model weights and execution
- Requires significant infrastructure expertise
- Maximum flexibility, higher operational overhead

Option B: Confidential Cloud Services

- Hardware-attested remote execution through APIs
- Cryptographically verifiable security guarantees
- Standard cloud interfaces with enterprise-grade protection
- Faster deployment, managed security operations

Why Option B is Gaining Enterprise Adoption

Confidential cloud platforms like Sindri are becoming the preferred choice for regulated industries. They combine the operational simplicity of cloud APIs with the security guarantees of hardware enclaves. Leading solutions in this space offer:

- Independent verification: Hardware attestation you can audit yourself
- Standard integration: Drop-in API compatibility with existing systems
- Verifiable security: Every claim backed by cryptographic proof

Rather than building complex TEE infrastructure, enterprises are choosing platforms like Sindri that deliver these guarantees as a service. This approach typically reduces deployment time from months to weeks while maintaining the highest security standards.

Quick Implementation: Deploy a proxy that routes to attested inference endpoints. Most teams can have confidential inference running within days, not months.

Ask Your Vendor:

- "Can I audit memory, disk, and model operation logs?"

- "What attestation do you provide for confidential execution?"
- "How do you verify no data retention or training use?"

Step 5: Wrap a Safe Output-Filtering Layer

Objective: *Scan and filter model outputs before delivery to users.*

Implementation:

- **Real-time PII detection**: Scan outputs for sensitive patterns
- **Content policy enforcement**: Check against organizational guidelines
- **Hallucination detection**: Flag suspicious claims or facts
- **Redaction controls**: Automatically remove or mask sensitive content

Example Workflow:

Step 6: Deploy a Browser-Safe Internal UI

Objective: *Create a secure web interface with proper session management.*

Implementation:

- **SSO Integration**: Use corporate identity providers
- **No local caching**: Messages stay server-side only
- **Session isolation**: Each user gets isolated chat context
- **Audit visibility**: IT can monitor sessions if required

Step 7: Monitor and Iterate with Audit Trails

Objective: *Maintain comprehensive logs for compliance and improvement.*

Implementation:

- **Session metadata**: Track user, timestamp, role, but not content
- **Security events**: Log access violations and policy triggers
- **Performance metrics**: Monitor latency and accuracy
- **Compliance exports**: Generate reports for auditors

Key Metrics to Track:

- Query volume by role and data class
- Policy violation frequency
- Average response time

- User satisfaction scores

Ask Your Vendor:

- "Can I export full anonymized session logs for compliance?"
- "What granularity of audit data do you provide?"
- "How do you handle log retention and deletion?"

Deployment Maturity Matrix

Capability	Pilot	MVP	Scaled	Optimized
Prompt Control	Ad hoc redaction	Role-based filters	Sealed front end	Fully tokenized + scoped
Model Infrastructure	Confidential API POC	Multi-model support	API w/ retention policy	Hosted confidential
Data Access	Manual approval	Basic RBAC	Automated policies	Dynamic context-aware
Monitoring	Basic logs	Session tracking	Compliance reports	Real-time analytics
Privacy	Best effort	Policy enforcement	Encryption in transit	End-to-end confidential

Confidential Chat ROI Snapshot

Cost of Breach

- **Average data breach cost**: $4.88M per incident (2023)[1]
- **Per-record cost**: $161 for sensitive data[2]
- **Regulatory fines**: Up to 4% of annual revenue (GDPR)[3]
- **IP theft impact**: Significant revenue loss[4]

Internal Productivity Gains

- **Legal teams**: 40% faster document review
- **Finance**: 65% reduction in report preparation time
- **Engineering**: 30% acceleration in code documentation
- **Sales**: 55% improvement in proposal turnaround

Consolidated Stack Value

- **Vendor reduction**: 3-4 point solutions → 1 comprehensive platform
- **Security overhead**: 90% reduction in AI compliance management
- **Training costs**: 90% less security training needed

Ninety Day Implementation Roadmap

90-Day Implementation Roadmap	Key Activities
Phase 1: Assess (Weeks 0–2)	
Week 1	☐ Identify primary use cases ☐ Map data classification requirements ☐ Inventory existing AI tool usage
Week 2	☐ Evaluate vendor options ☐ Define security requirements ☐ Create project charter
Phase 2: Secure (Weeks 2–6)	
Weeks 2–3	☐ Deploy prompt input laye ☐ Configure role-based access ☐ Set up private vector DB
Weeks 4–5	☐ Implement model inference pipeline ☐ Deploy output filtering ☐ Create audit logging
Week 6	☐ Build initial UI ☐ Test security controls ☐ Conduct pilot with limited users
Phase 3: Scale (Weeks 6–12)	
Weeks 6–8	☐ Onboard additional teams ☐ Refine access controls ☐ Gather feedback
Weeks 9–10	☐ Optimize performance ☐ Enhance monitoring ☐ Document processes
Weeks 11–12	☐ Prepare compliance reports ☐ Plan future enhancements ☐ Measure ROI

Let's design your first internal LLM interface the right way: confidential by *default*.

Your organization deserves AI tools that enhance productivity without compromising security. Starting with a proper architecture saves months of retrofitting and potential security incidents.

Ready to build with confidence? Book a 15-minute blueprint consultation with Sindri where we'll:

- Review your specific security requirements
- Identify the best deployment approach for your organization
- Provide a custom implementation timeline

Book Your Consultation →

About Sindri

Sindri builds confidential GenAI infrastructure used by teams to safely deploy LLMs across legal, finance, healthcare, and product organizations. Our API-first architecture supports private model execution, zero data leakage, and full audit logs for verifiable security.

Our Platform Features:

- **Private Model Execution**: TEE-based inference with hardware attestation
- **Complete Audit Trails**: Every query logged and verifiable
- **Enterprise Integration**: Seamless connection to existing data and identity systems

Trusted by:

- Fortune 500 institutions
- Leading technology companies
- Government agencies

Sources:

1. Cost of a data breach 2024 | IBM
2. Data Breach Costs Key Drivers and Trends
3. Art. 83 GDPR – General conditions for imposing administrative fines - General Data Protectio…
4. Council Post: The Costs Of IP Theft And How To Protect Your Company's Ideas