ZEROSTIC

API-First Proposal: GPT-5 Integrated Retrieval for Adamet Pvt Ltd

Enhance GPT-5 with multi-source enterprise APIs. Private, provenance-tracked, policy-enforced. No new UI required. Tailored solution for Adamet's business intelligence needs.

Prepared for: Adamet Pvt Ltd

Date: November 4, 2025

Zerostic
Email: support@zerostic.com
Web: www.zerostic.com

Executive Overview

This proposal outlines an API-first integration architecture that enhances GPT-5 Enterprise with multi-source enterprise data retrieval capabilities under strict privacy and compliance controls.

Enhance GPT-5 with multi-source enterprise APIs. Private, provenance-tracked, policy-enforced. No new UI required. Tailored solution for Adamet's business intelligence needs.

Client Requirements

Our understanding of Adamet Pvt Ltd's requirements:

1. API-First Architecture

Integrate multiple internal data sources into GPT-5 tools under strict privacy controls.

- All data access via RESTful APIs with JWT authentication
- GPT-5 calls tool facade; no direct database access
- Zero model training on proprietary data
- Support for future connector additions without code changes

2. Multi-Source Integration

Seamlessly connect SharePoint, Tender DB, and competitor data with unified retrieval.

- Pluggable connector architecture for each data source
- Unified response format across all sources
- Parallel query execution with result fusion
- Source-specific authorization checks

3. Privacy & Compliance First

Enforce ABAC/RBAC policies and redaction before any payload reaches GPT-5.

- Policy engine filters unauthorized chunks pre-retrieval
- Row-level and document-level access control
- PII/sensitive data redaction in payloads
- Full audit trail with compliance logging

4. Deterministic Provenance

Every response includes durable citations with doc_id, record_id, source URI, and checksum.

- Citation mappings stored in provenance store
- Timestamped retrieval traces
- Immutable citation URIs for audit
- Human-readable source attribution in responses

5. No New End-User UI

Users interact only with existing GPT-5 interface; all complexity hidden behind tool calls.

- Zero changes to end-user chat experience
- Tool invocation transparent to users

- Response format identical to standard GPT-5
- Admin dashboards for ops/analytics only

Our Solution Architecture

Six core pillars that deliver secure, scalable, and auditable AI integration

1. API Gateway / Tool Facade

GPT-5 Enterprise calls our secure API Gateway, which presents tools like fetch_tenders, search_docs, and fetch_competitors. All requests authenticated via JWT/OIDC.

2. RAG Orchestrator

Plans multi-source queries, orchestrates semantic and keyword searches, fuses results, and assembles context with citations before returning to GPT-5.

3. Policy Before Payload

Policy Engine enforces ABAC/RBAC and redacts unauthorized chunks. Only compliant, filtered data reaches the model.

4. Pluggable Connectors

Modular adapters for SharePoint, Tender DB, Competitor DB, and future sources. Each connector handles authentication, schema mapping, and retrieval.

5. Provenance Store

Durably maps every retrieved chunk to its source (doc_id, URI, checksum, timestamp). Enables citation verification and audit trails.

6. Observability & Audit

Logs, metrics, and traces for every query. Policy audits, latency SLOs, and retrieval analytics in one dashboard.

High-Level Architecture

API-first design with GPT-5 Enterprise as the consumer. All components enforce access control and redaction before returning payloads.

- Identity Provider (SSO/RBAC) authenticates all requests
- API Gateway exposes tools to GPT-5 with normalized query/plan interface
- RAG Orchestrator performs semantic + keyword search, filters via Policy Engine
- Connectors retrieve from SharePoint, Tender DB, Competitor DB
- Provenance Store maps citations; Observability tracks all operations

Security & Compliance

No Model Training
Zero proprietary data used for fine-tuning or model updates

ABAC/RBAC
Attribute and role-based access control enforced before retrieval

Redaction
PII and sensitive fields masked in all payloads to GPT-5

Private Networking
All internal APIs on private VPC with zero public exposure

Audit & Telemetry
Full trace logs with policy decisions and retrieval lineage

Deliverables & Timeline

Discovery & Design

Duration: 2 weeks

- Finalize connector schemas
- Define tool signatures for GPT-5
- Security policy framework

PoC Development

Duration: 4 weeks

- SharePoint + Tender DB connectors
- RAG orchestrator with policy engine
- 12-15 test queries with success metrics

Testing & Validation

Duration: 2 weeks

- Latency benchmarks (p95 < target)
- Provenance verification tests
- Security audit report

Production Rollout

Duration: 2 weeks

- Live integration with GPT-5 Enterprise
- Observability dashboards
- Runbook and ops handoff

Success Metrics

- Query Latency: 95 p95 < 2s
- Connector Uptime: 99.9 %
- Test Queries: 15 queries

Architecture Diagrams

For the best viewing experience, please access the interactive diagrams at the proposal website. The diagrams can be zoomed, panned, and downloaded in high resolution.

High-Level Architecture Diagram

The HLD diagram illustrates the complete system architecture including:

- GPT-5 Enterprise (OpenAI Hosted) with Private Link/VNet integration
- Identity Provider (Azure AD / Entra) for SSO and RBAC
- API Gateway (Tool Facade) exposing tools to GPT-5 with mTLS termination
- Policy Engine enforcing AuthZ (ABAC), data redaction, DLP, and chunk filtering
- RAG Orchestrator for query planning and multi-source retrieval
- Data Connectors for SharePoint, Tender DB, and Competitor DB
- Vector and Metadata Indexes for semantic and structured search
- Provenance Store for durable citations
- Observability & Audit system for logs, metrics, and compliance

All components operate within the Enterprise Tenant boundary with comprehensive policy enforcement at every interaction point. The diagram shows bidirectional data flows, authentication patterns, and security controls.

Sequence Diagram

The sequence diagram details the complete request flow:

1. User submits query through authenticated session (SSO via IdP)
2. API Gateway validates token and enforces initial authorization
3. GPT-5 receives query and invokes tools through mTLS + Private Link
4. Policy Engine authorizes tool access and applies redaction rules
5. RAG Orchestrator queries Vector and Metadata Indexes
6. Policy Engine filters results based on user permissions
7. Orchestrator retrieves fresh data from Connectors if needed
8. Provenance Store maps citations to source documents
9. Gateway returns enriched response to GPT-5 with citations
10. GPT-5 formulates final answer with provenance links
11. All interactions logged to Observability & Audit system