Point of View (PoV)

Azure-native End-to-End Agent Evaluation & Observability

Target audience: Enterprise Architects, Client Technology Leads

Executive summary

Enterprises adopting agentic AI need **LangSmith-class observability and evaluation** that can be safely operationalized within existing enterprise architecture, security, and governance models. This PoV describes an **Azure-native GenAIOps architecture** that enables architects and technology leads to design, review, and scale agent-based systems with confidence.

The approach standardizes on **Azure AI Foundry (Observability + Evaluation)** and **Azure Monitor / Application Insights (OpenTelemetry)** to deliver traceability, quality assurance, and continuous improvement while minimizing platform sprawl and operational overhead. Where certain agent-level capabilities are in public preview, this PoV outlines pragmatic architectural mitigations suitable for production environments.

---

1. Business problem

Agent-based AI systems are inherently non-deterministic and multi-step. Traditional APM tools fail to answer questions such as: - *Why did this agent fail for a specific user request? - Did a prompt or model change improve real task success? - Are we drifting from grounded, policy-compliant behavior in production?*

Most teams address this using external SaaS tools (e.g., LangSmith), which raises concerns around: - Data residency and IP exposure - Identity and access control fragmentation - Duplication of observability stacks - Long-term vendor lock-in

---

2. Design principles

1. **Azure-native by default**
   Reuse existing Azure security, monitoring, and governance controls.

2. **Evaluation is a first-class citizen**
   Quality, safety, and cost must be measurable before and after release.

3. **Open standards, not proprietary lock-in**
   Use OpenTelemetry and SDK-based evaluation.

4. **Minimal operational burden**
   Prefer managed services; avoid running custom observability platforms.

5. **Enterprise-grade security and compliance**
   RBAC, private networking, encryption, auditability, and retention policies.

3. Target capabilities (LangSmith-equivalent)

3.1 End-to-end agent tracing

- Correlated traces across:
- User request
- Planner / router
- Tool calls
- Retrieval (RAG)
- Model invocations
- Drill-down from high-level request → individual agent steps

Azure implementation - OpenTelemetry instrumentation in agent runtime - Azure Application Insights for trace, metric, and log storage - Azure AI Foundry Observability for AI/agent-specific views

---

3.2 Offline evaluation (Dev & CI/CD)

Purpose: Prevent regressions before deployment.

Capabilities: - Dataset-based evaluation (golden sets, edge cases, red-team cases) - Comparison of: - Prompt versions - Model versions - Retrieval configurations - Automated quality and safety scoring

Azure implementation - Azure AI Foundry Evaluation SDK - Model and dataset evaluation APIs (Generally Available) - Agent evaluation capabilities (Public Preview) - Pipeline integration (Azure DevOps / GitHub Actions) - Release gates based on evaluation thresholds

Enterprise mitigation - Treat preview agent evaluators as advisory signals - Retain lightweight custom evaluators for mission-critical gates

---

3.3 Continuous evaluation (Production)

Purpose: Detect drift, safety issues, and degradation in real usage.

Capabilities: - Sampled evaluation of live traffic - Trend analysis over time - Direct linkage from failed evaluations to traces

Azure implementation - Foundry continuous evaluation for agents - Configurable sampling strategies to control cost and latency - Dashboards integrated with Foundry Observability - Alerts via Azure Monitor

---

3.4 Feedback and human-in-the-loop

Capabilities: - SME and user feedback attached to traces - Promotion of feedback into curated evaluation datasets - Closed-loop improvement cycle

Azure implementation - Feedback metadata stored alongside traces - Golden and red datasets managed in governed Azure storage

---

4. Reference architecture (logical view)

4.1 Runtime layer

- Agent hosted on:
- Azure App Service / AKS / Functions
- Uses:
- Azure OpenAI models
- Azure AI Search (RAG)
- Enterprise tools and APIs

4.2 Observability layer

- OpenTelemetry SDKs emit:
- Traces
- Metrics
- Logs
- Azure Application Insights as the central telemetry store
- Azure AI Foundry Observability for AI-centric analysis

4.3 Evaluation layer

- Offline evaluation (CI/CD): Foundry Evaluation SDK
- Online evaluation (Prod): continuous evaluation sampling

4.4 Governance layer

- Azure RBAC and managed identities
- Azure Policy and resource tagging
- Centralized retention and audit controls

4A. Architecture decision summary (ADR-style)

Decision: Adopt Azure-native observability and evaluation for agentic AI workloads.

Alternatives considered: - External SaaS (e.g., LangSmith) - Custom-built evaluation and tracing platform

Rationale: - Alignment with existing Azure security, identity, and monitoring controls - Reduced operational overhead - Avoidance of data egress and SaaS governance risk

Trade-offs: - Faster feature velocity in external SaaS tools - Some Azure agent-evaluation capabilities currently in preview

Mitigation: - Use preview features as advisory signals - Retain lightweight custom evaluators for hard release gates

---

4B. Non-functional requirements mapping

NFR	Architectural response
Security	Azure RBAC, managed identities, no-secret telemetry
Compliance	Inherits Azure certifications; tenant-controlled retention
Scalability	Azure Monitor + OpenTelemetry scale patterns
Availability	Platform-managed SLAs; decoupled evaluation plane
Cost governance	Sampling strategies, evaluation thresholds, dashboards

---

4C. Reference architecture walkthrough (diagram narrative)

1. User request enters the agent runtime.
2. Correlation ID is created and propagated across all agent steps.
3. Planner, tool calls, retrieval, and model invocations emit OpenTelemetry traces.
4. Telemetry is ingested by Application Insights.
5. Azure AI Foundry Observability provides AI-specific trace visualization.
6. Evaluation runs either offline (CI) or sampled online (Prod).
7. Failed evaluations link directly back to traces for root-cause analysis.

---

5. Security, privacy, and compliance posture

Data protection

- Prompt and response scrubbing
- No secrets or credentials in telemetry
- Tokenization or hashing of identifiers

Identity & access

- Azure RBAC scoped by environment (Dev/Test/Prod)
- Least-privilege access to traces and datasets

Network & encryption

- Private endpoints where supported
- Encryption at rest (platform-managed or CMK)

Compliance & audit

- Inherits Azure compliance certifications (e.g., ISO, SOC, regional standards)
- Configurable log retention aligned to enterprise policy

- Full traceability for audits and incident reviews

---

6. KPIs and success metrics

Quality

- Task success rate
- Groundedness and relevance (RAG)
- Tool success and retry rates

Safety & compliance

- Policy violation rate
- Severity distribution
- Time-to-detection

Reliability

- p95 latency by agent step
- Error budgets
- MTTR

Cost

- Tokens per successful task
- Retrieval hit rate
- Cost per completed workflow

---

7. Reference adoption roadmap (architect-led)

Phase 0: Architectural foundation

- Define observability and evaluation standards
- Instrument agents with OpenTelemetry
- Establish RBAC and environment isolation

Phase 1: Quality gates

- Define golden and red datasets
- Integrate Foundry Evaluation into CI/CD
- Establish release thresholds

Phase 2: Production assurance

- Enable continuous evaluation with sampling
- Configure alerts and dashboards
- Formalize incident response workflows

Phase 3: Optimization and scale

- Feedback-driven dataset expansion
- Cost and latency optimization loops
- Architecture governance reporting

---

8. Scope boundaries and known limitations

This architecture intentionally does **not**: - Replace enterprise APM or SIEM platforms - Eliminate the need for domain-specific human review - Guarantee deterministic behavior from LLMs - Remove the need for prompt and agent design discipline

These constraints should be explicitly acknowledged during architecture reviews.

---

8. Why this is better aligned than external SaaS observability for enterprises

Dimension	Azure-native GenAIOps	External SaaS
Security & compliance	Inherits Azure controls	Separate trust boundary
Data residency	Tenant-controlled	Vendor-dependent
Identity	Azure AD / RBAC	Separate IAM
Ops overhead	Managed services	Additional platform
Lock-in	OpenTelemetry + SDKs	Proprietary

---

9. Conclusion

For Enterprise Architects and Client Technology Leads, this PoV provides a **reviewable, governable, and production-aligned** approach to agent observability and evaluation on Azure.

By treating evaluation and observability as **architectural capabilities rather than developer tools**, organizations can scale agentic AI systems with confidence—balancing innovation speed with enterprise-grade control, compliance, and cost governance.

This Azure-native approach delivers **LangSmith-equivalent agent observability and evaluation** while aligning with enterprise expectations for **security, compliance, scalability, and operational simplicity**.

It positions GenAI systems as **governable enterprise platforms**, not experimental tools—unlocking confident scale-out across regulated and mission-critical use cases.