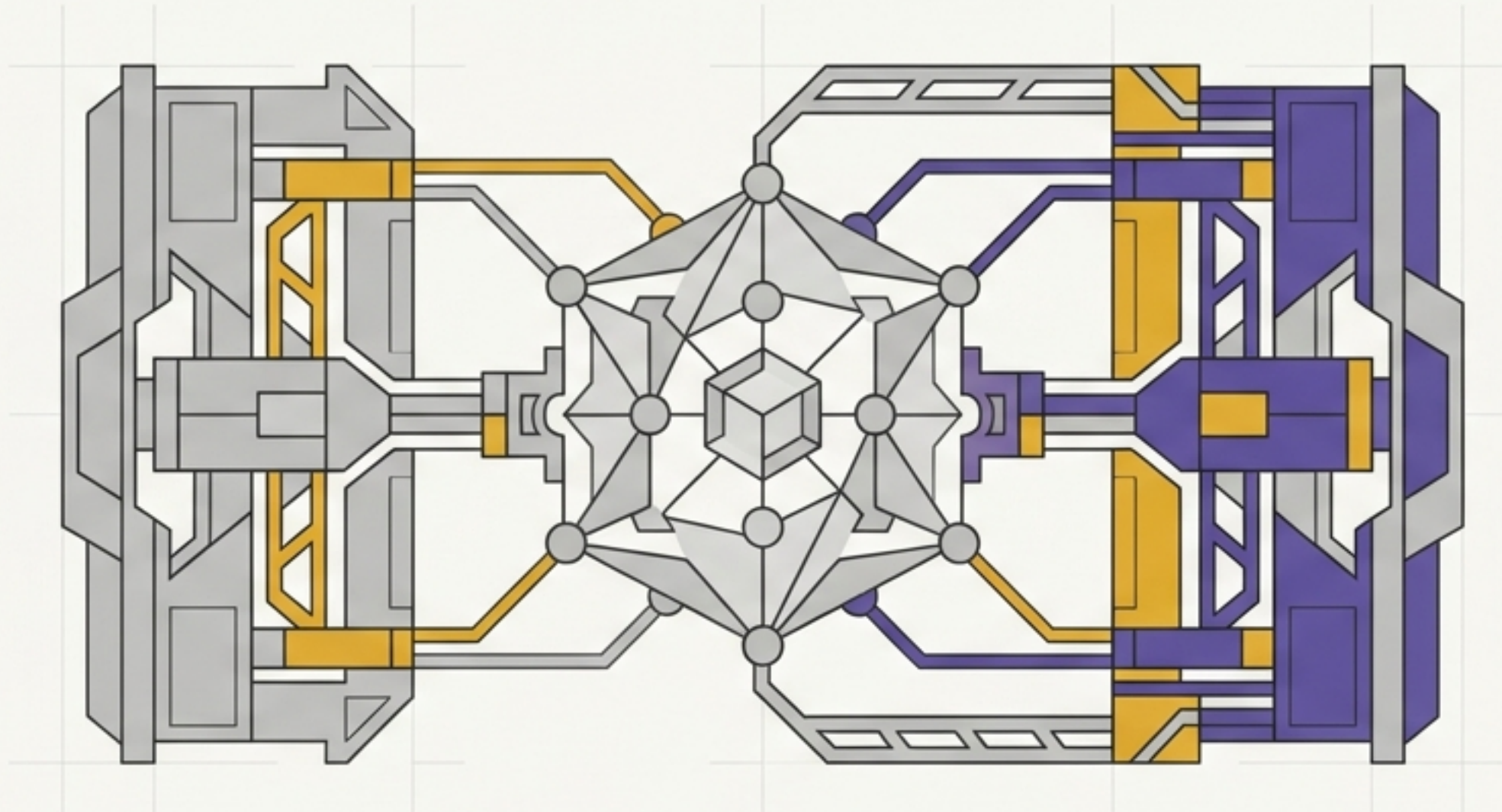


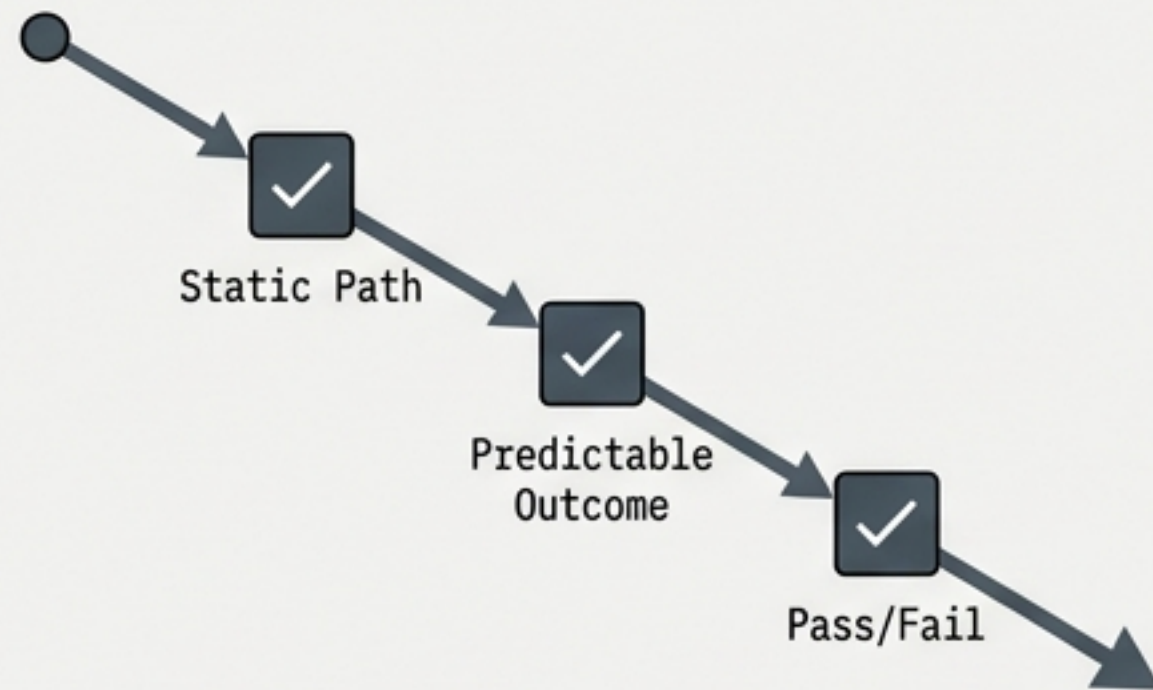
AGENTIC QUALITY ENGINEERING

The Step-by-Step Playbook for AI Agent Testing with SAP, Tricentis, and Open Source.



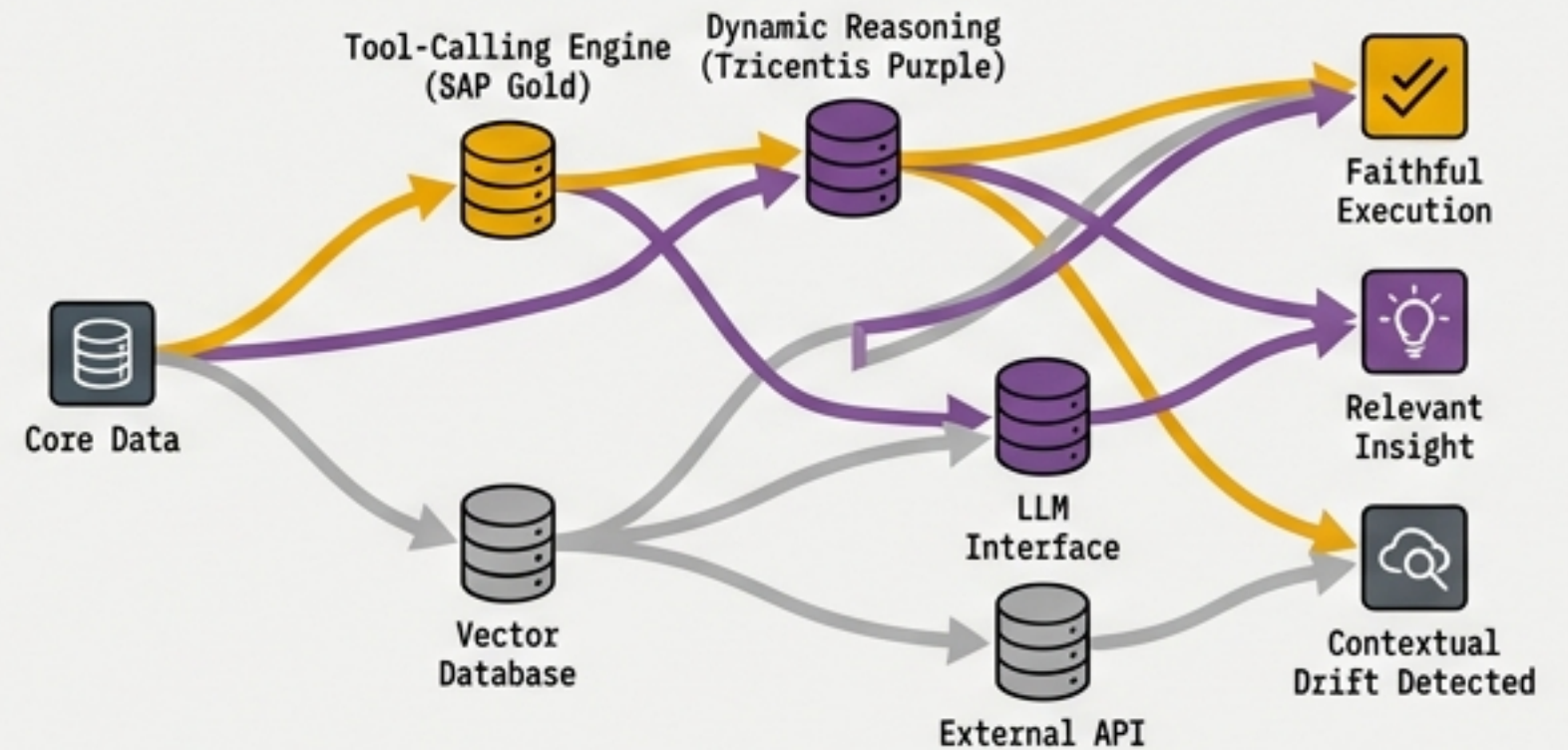
The SAP Autonomous Enterprise Requires a Probabilistic Approach to Quality

Traditional Software QA



- Static paths, predictable outcomes.
- Binary assertion testing (Pass/Fail).
- Pre-scripted UI/API interactions.

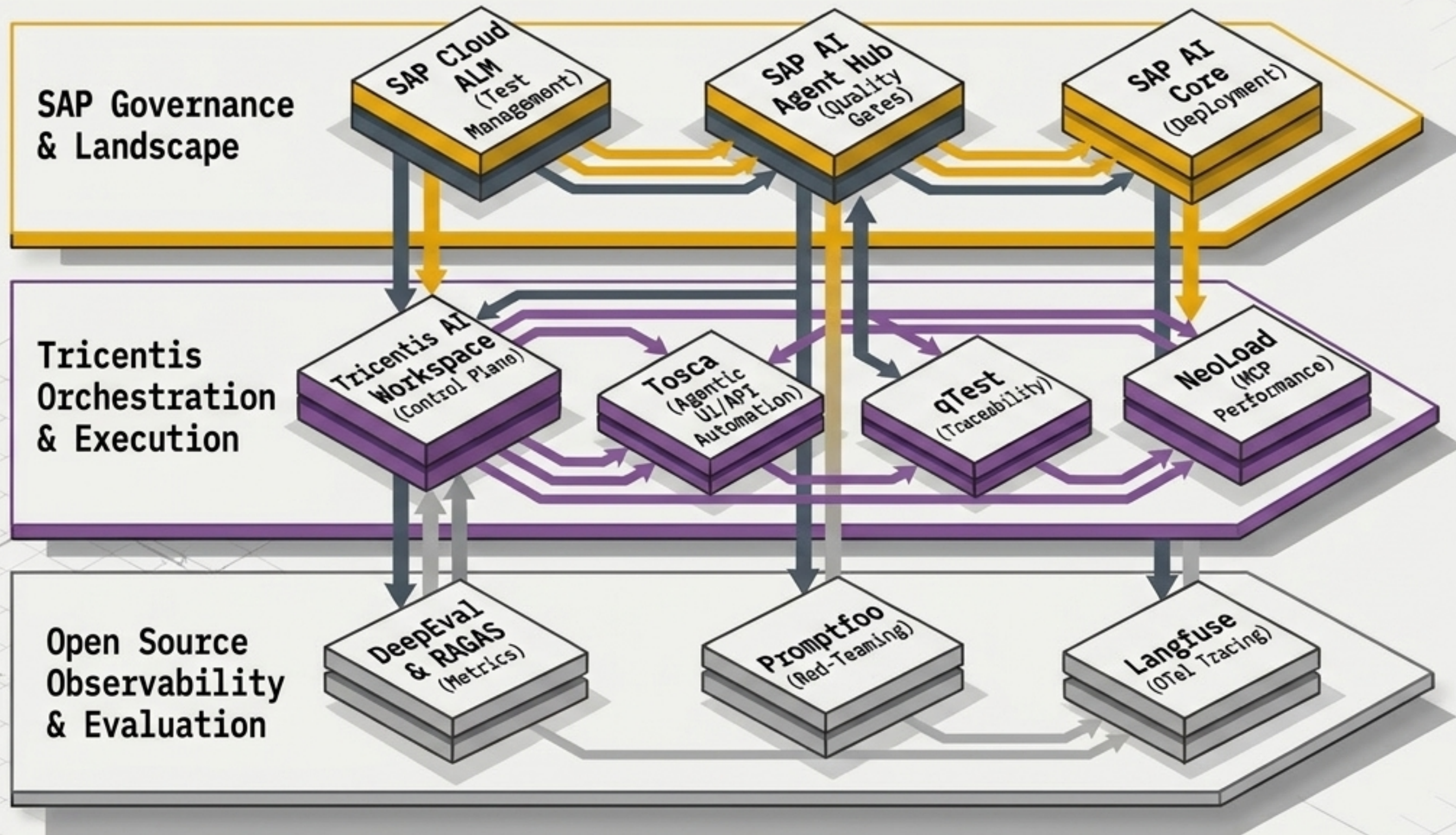
Agentic QA



- Autonomous tool-calling and dynamic reasoning.
- Probabilistic evaluation (Faithfulness, Relevancy).
- Multi-agent handoffs and continuous drift monitoring.

With SAP's €100M partner investment and 680+ agents in the SAP AI Agent Hub, the 2026 Agent-Led Toolchain shifts QA from verifying code to verifying intent and trust.

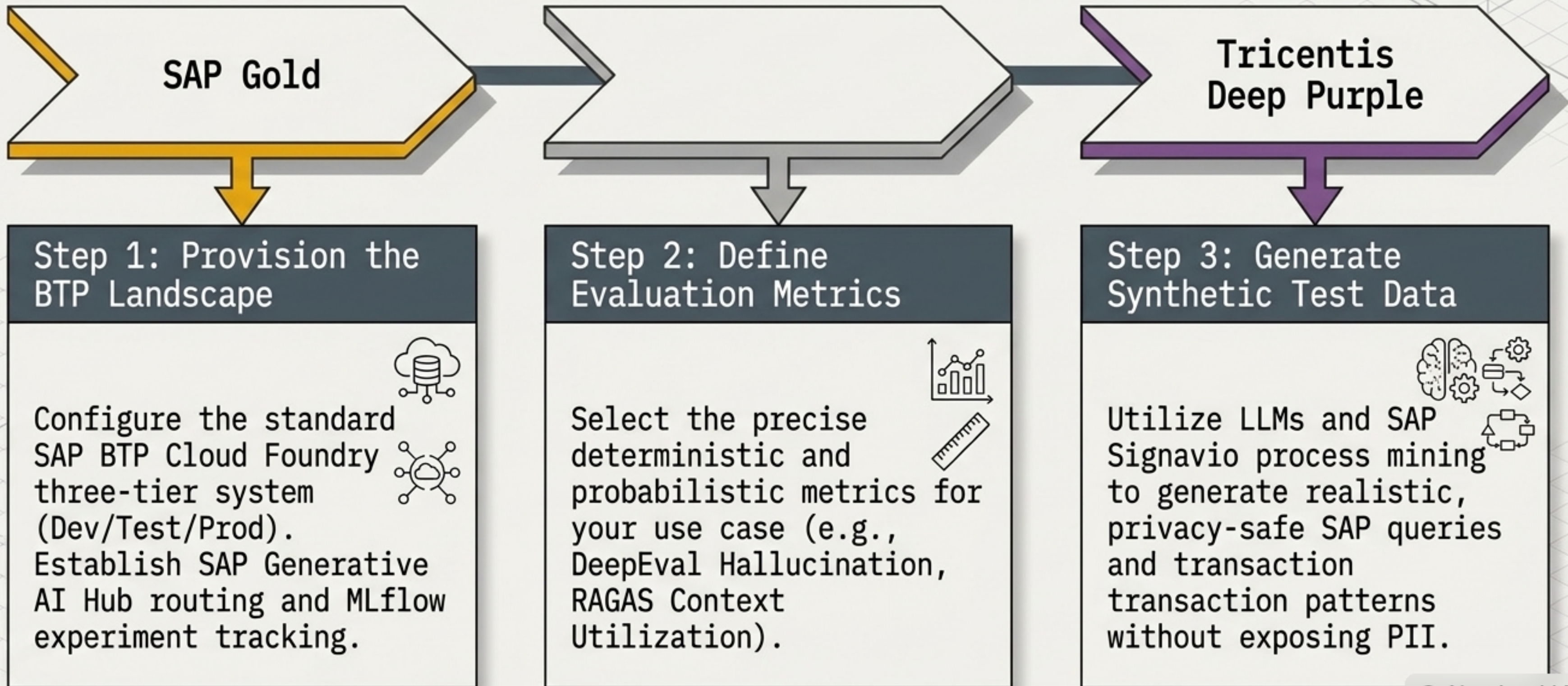
The Unified Enterprise Agent Testing Architecture



Architecture Note

This hybrid stack provides the rigorous control of enterprise platforms with the cutting-edge evaluation metrics of open-source frameworks.

Phase 1: Landscape Foundation & Metric Definition (Weeks 1-4)



The Probabilistic Evaluation Metrics Matrix

Target of Evaluation	<u>DeepEval Metrics</u>	<u>RAGAS Metrics</u>
Output Accuracy	<u>Hallucination Metric</u> (checks contradictions against context), <u>Answer Relevancy</u> (G-Eval based scoring)	<u>Answer Relevancy</u> (LLM-as-a-judge scoring)
Retrieval Pipeline (RAG)	<u>Contextual Precision</u> , <u>Contextual Recall</u>	<u>Context Precision</u> (ranking of retrieved SAP KB items), <u>Context Utilization</u> (how effectively the LLM uses retrieved context)
Agentic Execution	<u>Tool Use Correctness</u> (validating agent API calls to S/4HANA), <u>Agent Goal Completion</u> (end-to-end task success)	N/A

[SYSTEM_NOTE]: Both frameworks support LLM-as-a-judge evaluation natively via SAP Generative AI Hub APIs.

Phase 2: Building the Evaluation Pipeline & Security (Weeks 5-8)

Step 4: Build the Evaluation Suite



Implement DeepEval as an evaluation service on SAP BTP. Connect test results to SAP Cloud ALM via REST API for quality gating.

Step 5: Set Up Observability



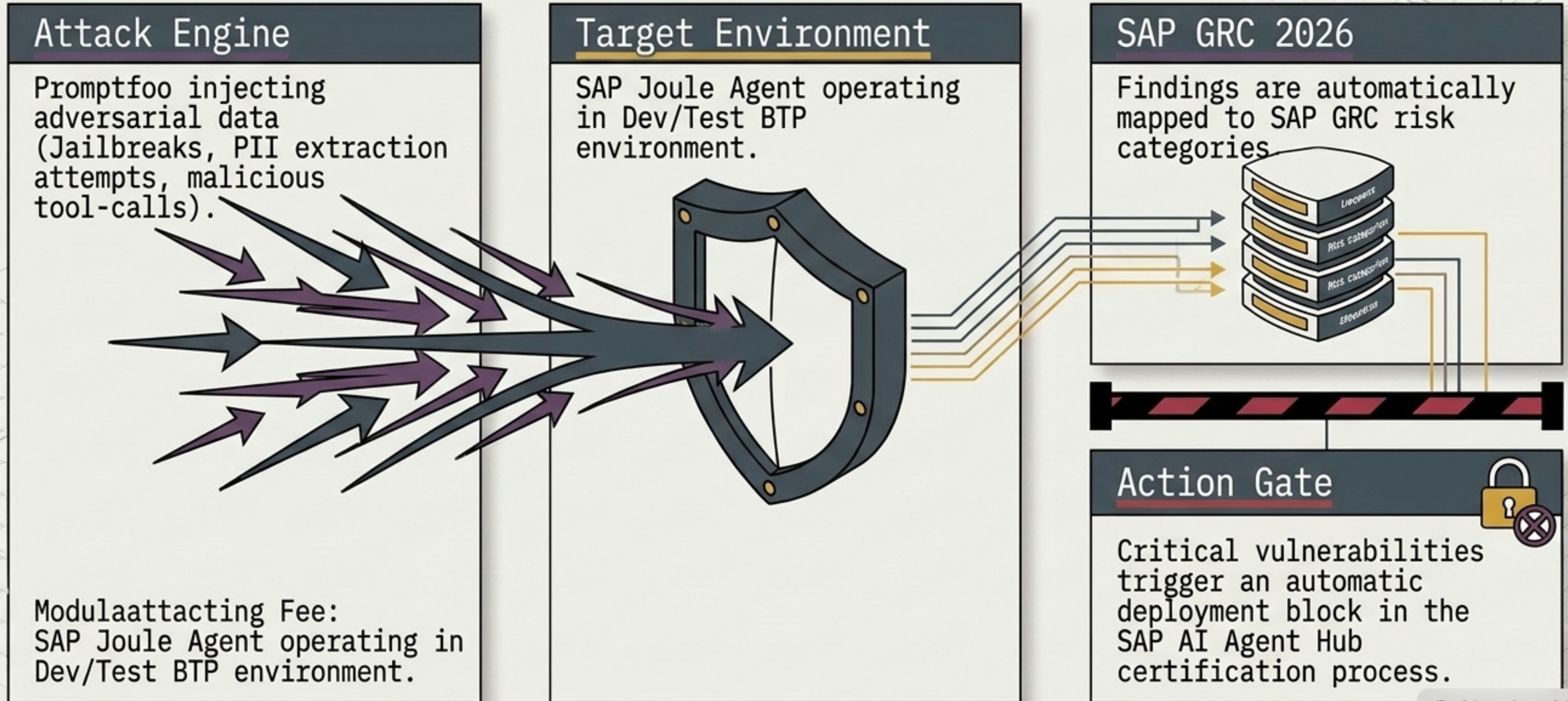
Deploy Langfuse or Arize Phoenix for OpenTelemetry-native LLM tracing. Monitor prompt versioning and latency.

Step 6: Implement Red-Teaming

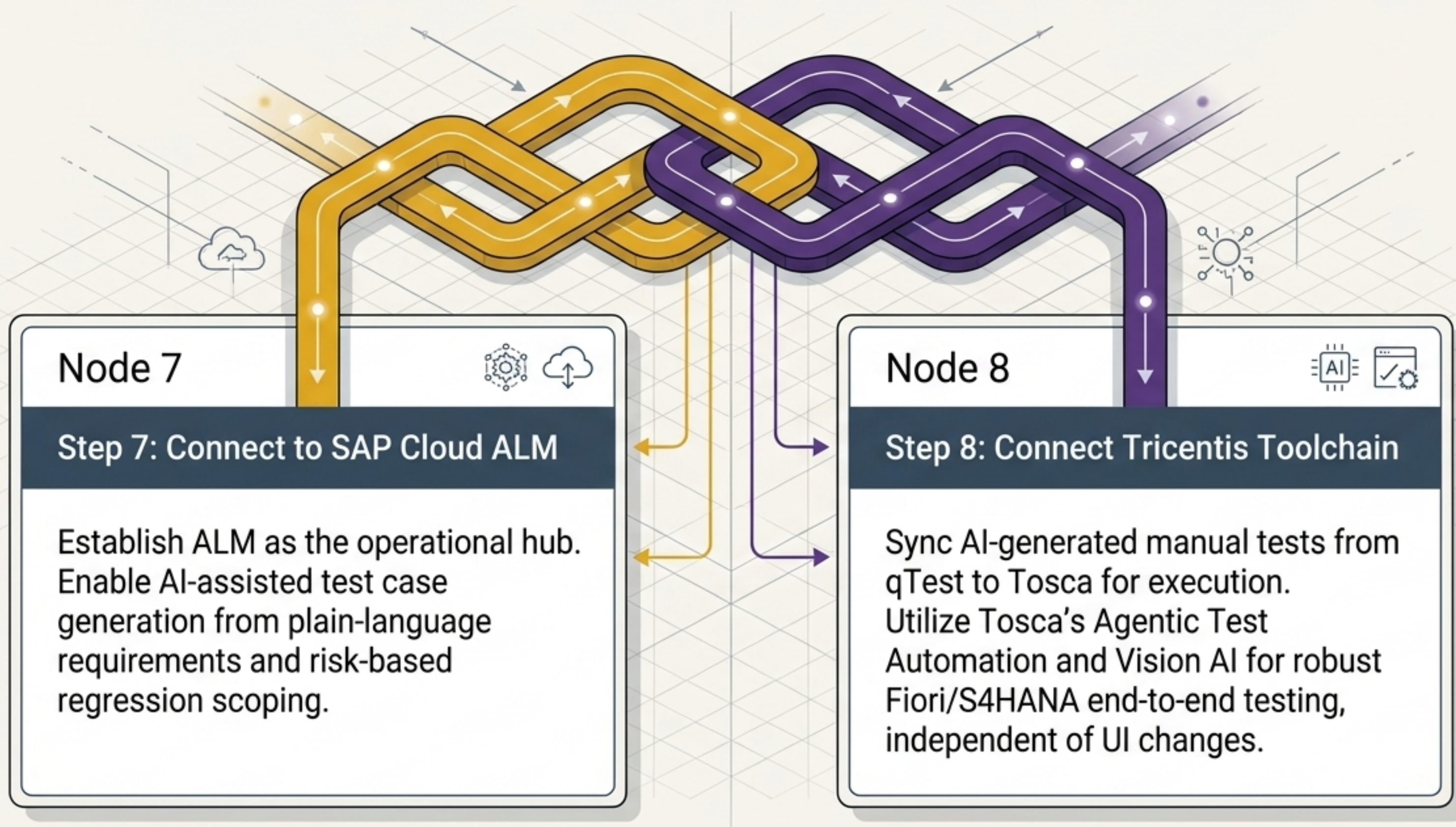


Configure automated vulnerability scanning using Promptfoo or Giskard to test resistance to prompt injections prior to deployment.

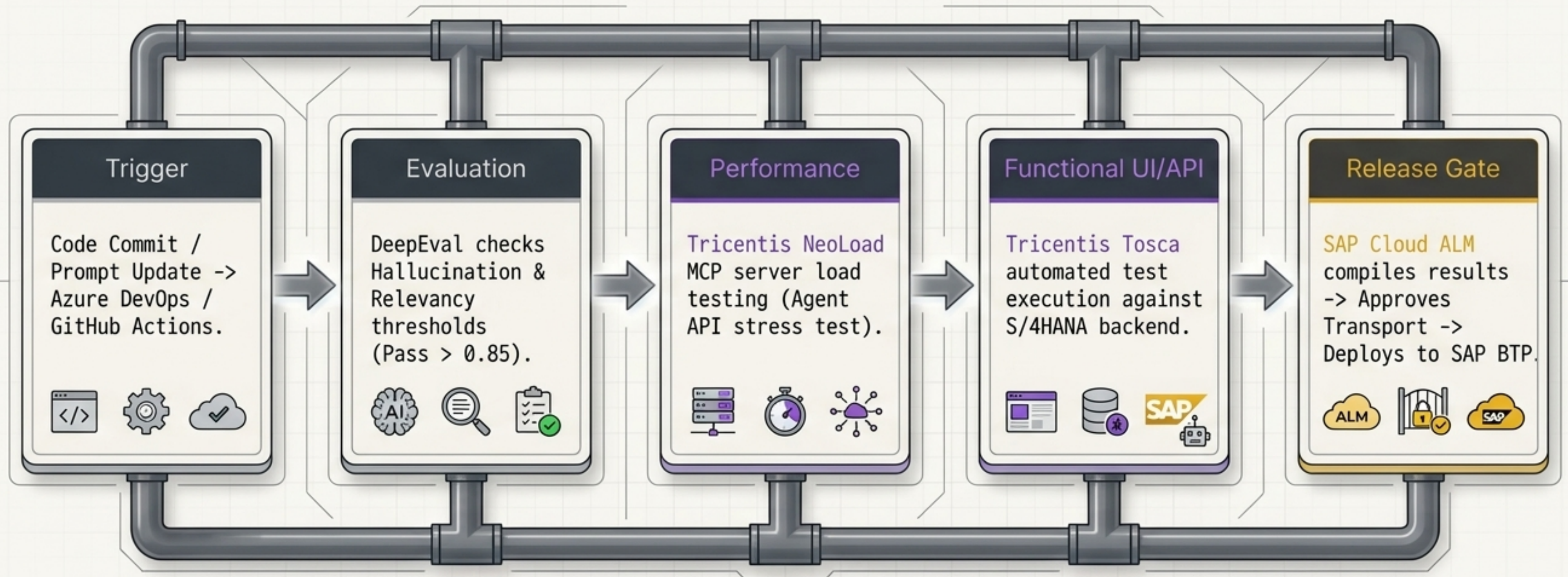
Automated Red-Teaming: Mapping Vulnerabilities to SAP GRC



Phase 3: SAP and Tricentis Enterprise Integration (Weeks 9-12)



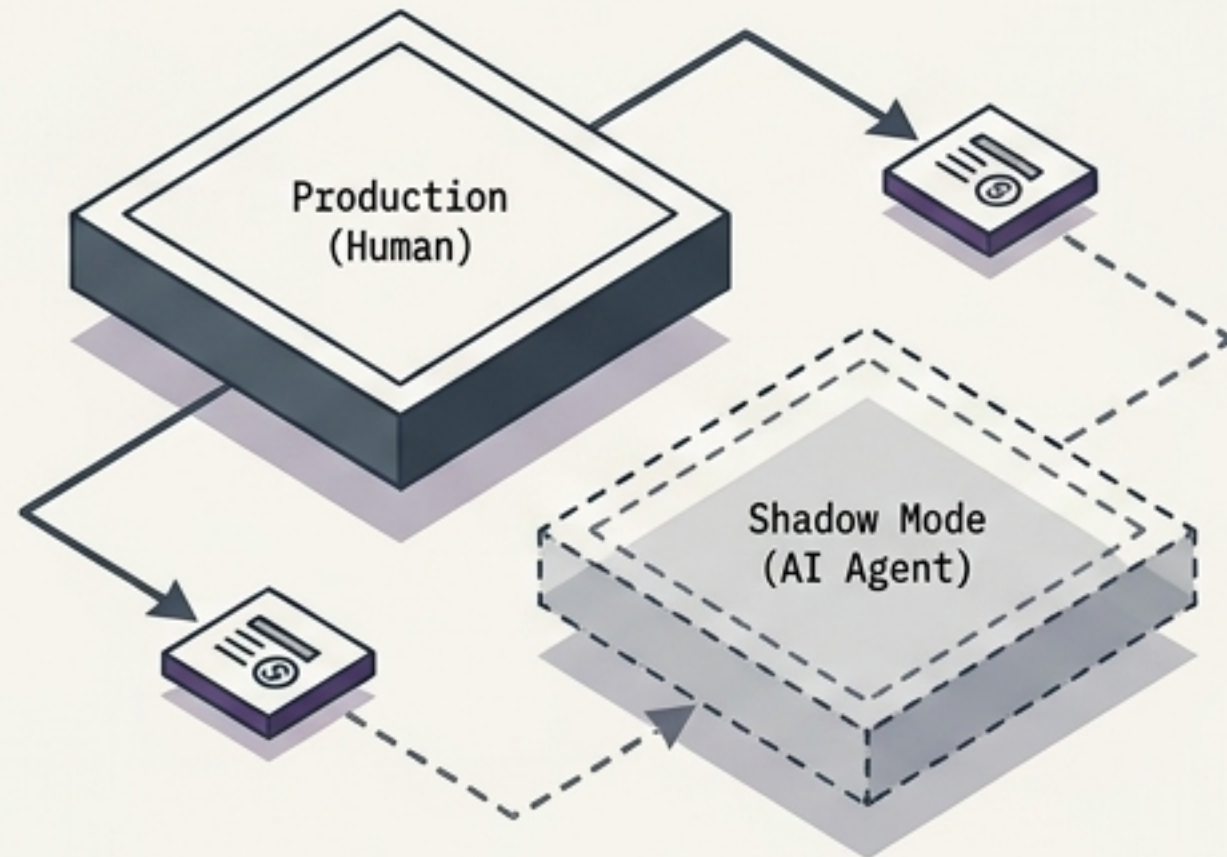
The Agent-Led Toolchain CI/CD Execution Path



Phase 4: Production Deployment Strategies (Weeks 13-16)

Node 9

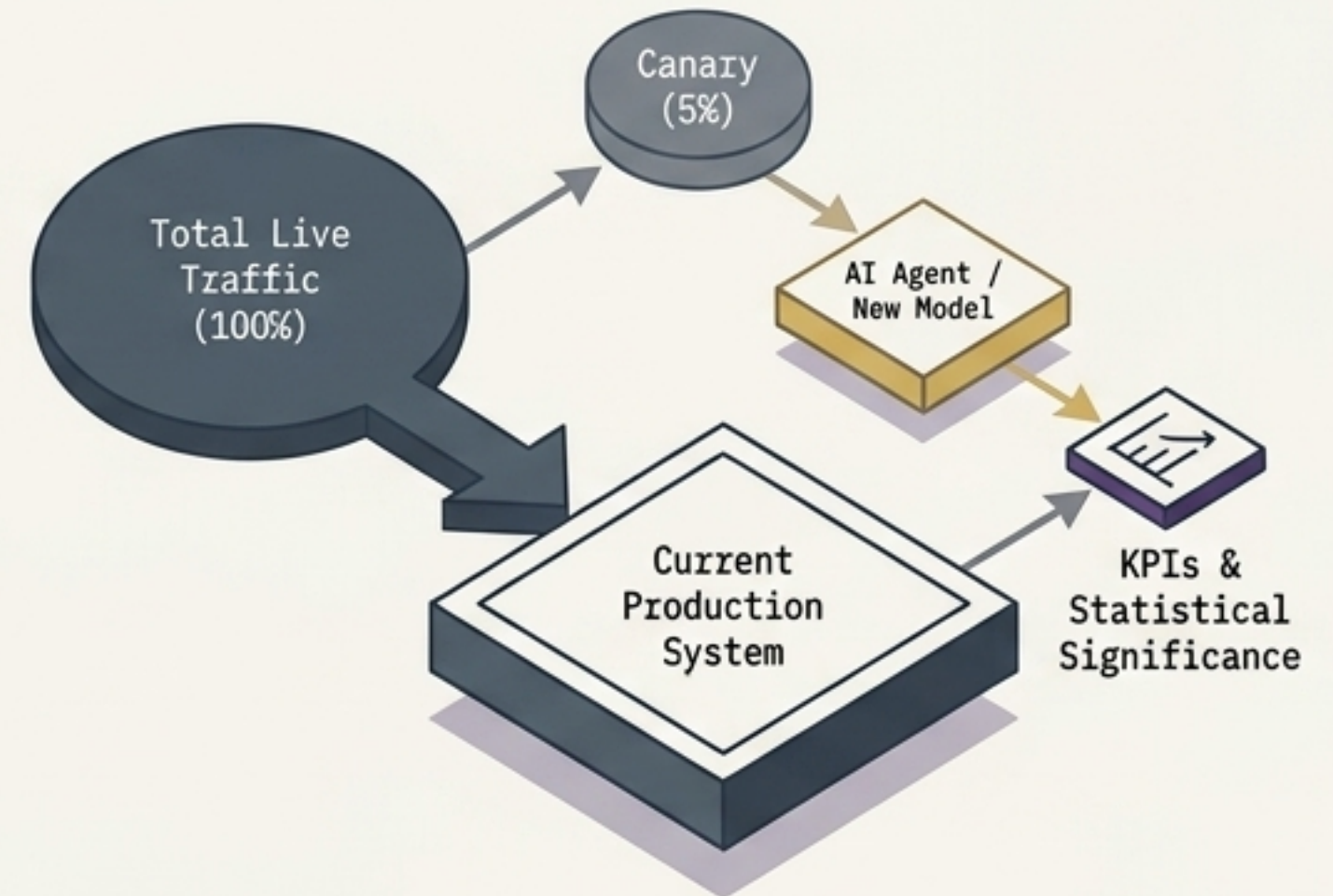
Step 9: Shadow Mode Activation



Run the AI agent in parallel with human operators in production. The agent processes the exact same transactions silently, allowing for real-world evaluation without business risk.

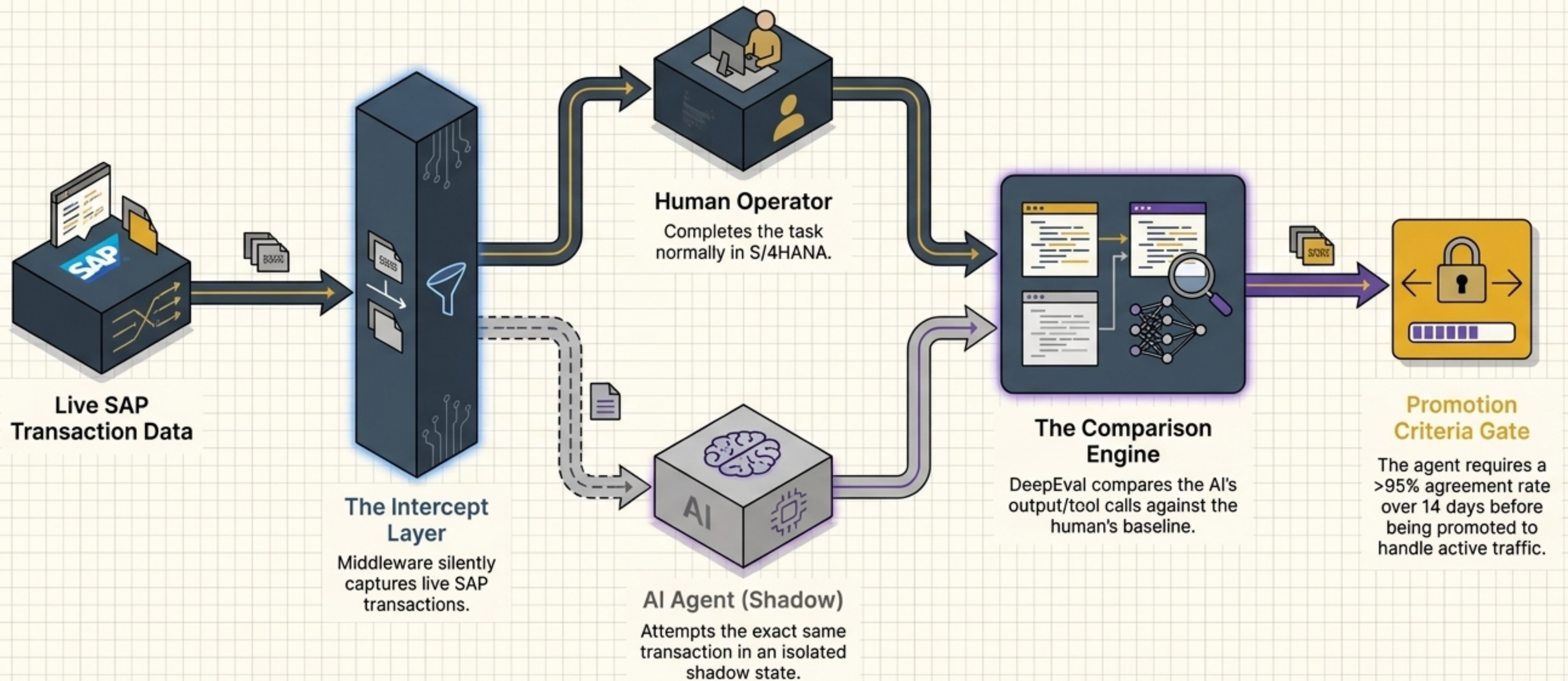
Node 10

Step 10: Canary Deployment

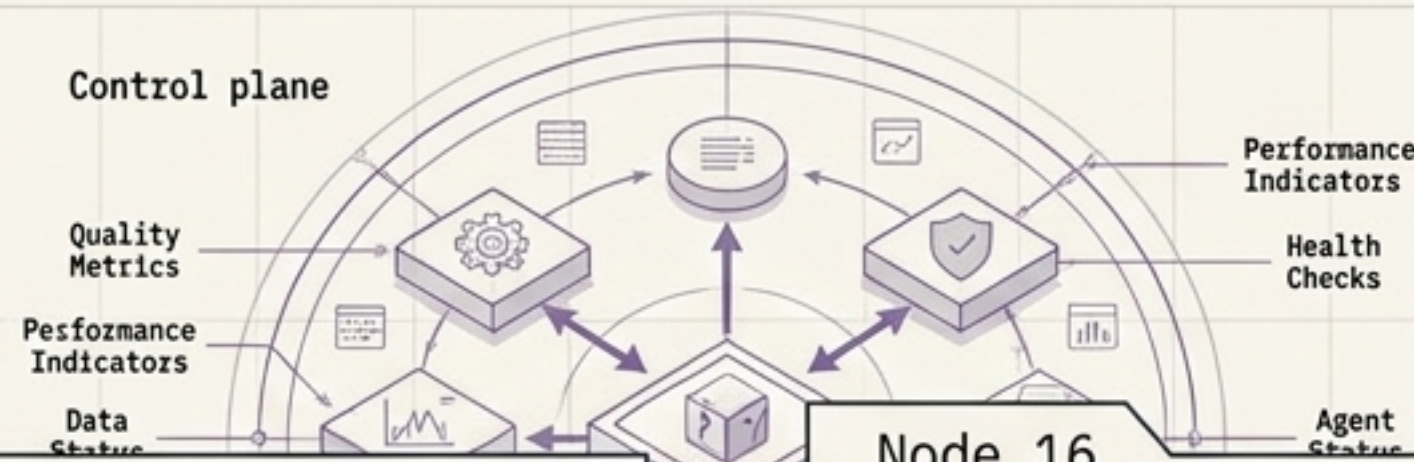


Route a small percentage (e.g., 5%) of live traffic to the agent (or a new LLM model like GPT-4o vs Claude 3.5). Measure business KPIs and use statistical significance testing before a full rollout.

Architectural Pattern: Shadow Mode in SAP

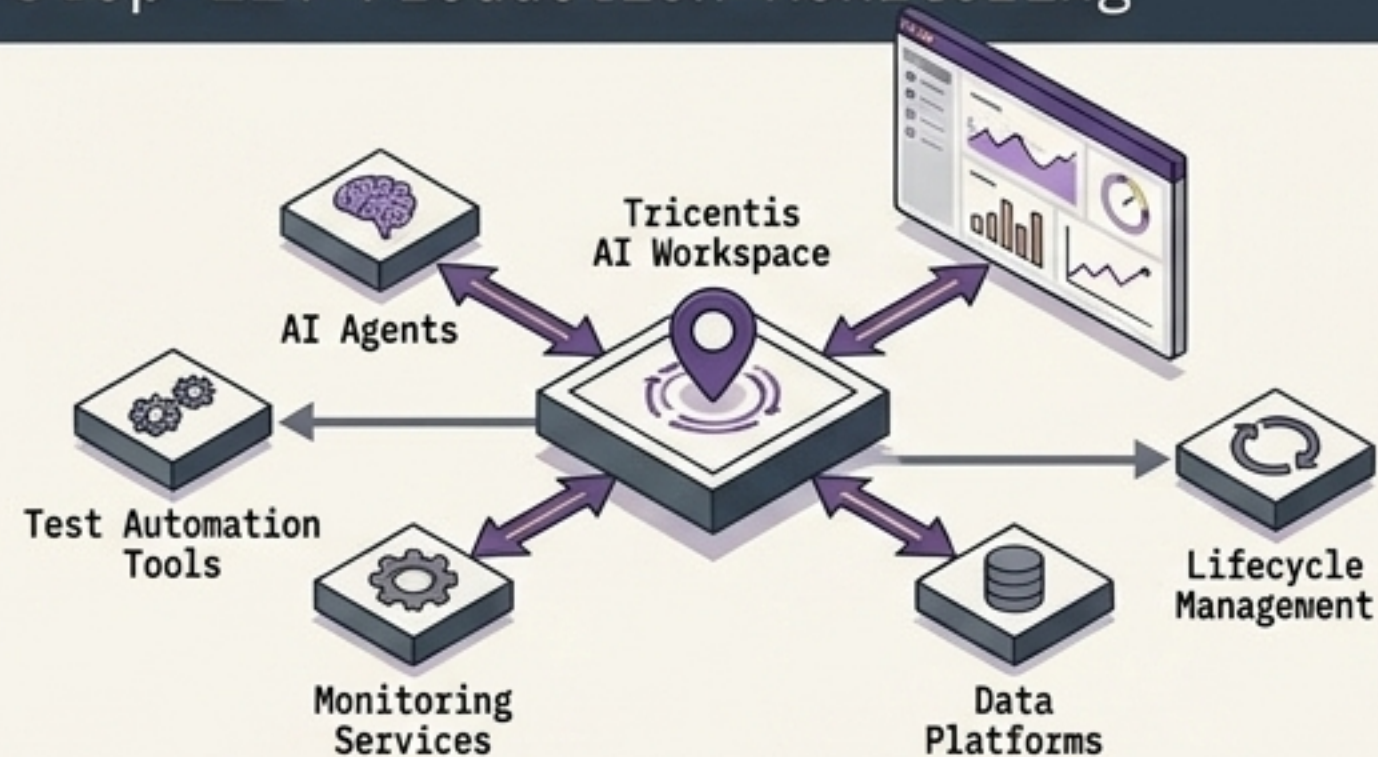


Phase 5: Continuous Operations & Scaling (Ongoing)



Node 12

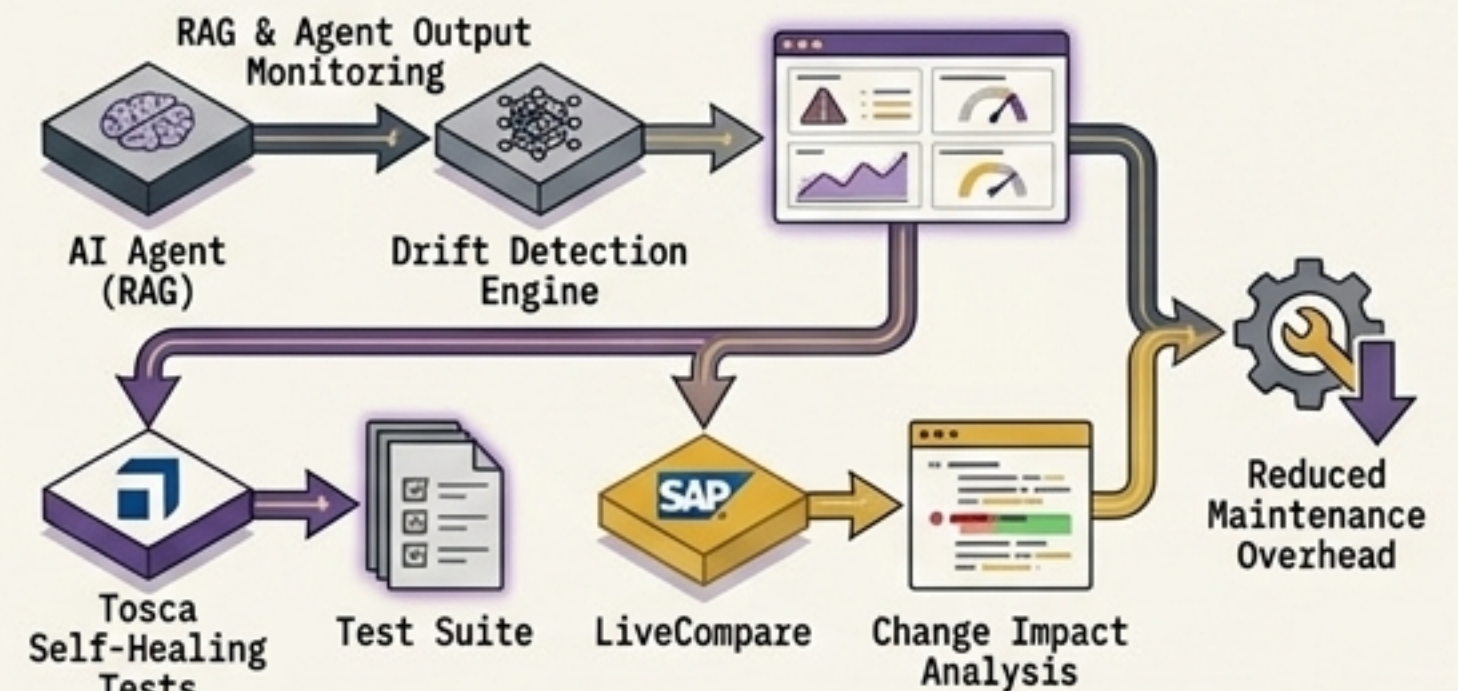
Step 11: Production Monitoring



Utilize Tricentis AI Workspace as the control plane for autonomous quality engineering. Orchestrate complex workflows bringing together agents and tools across the software lifecycle.

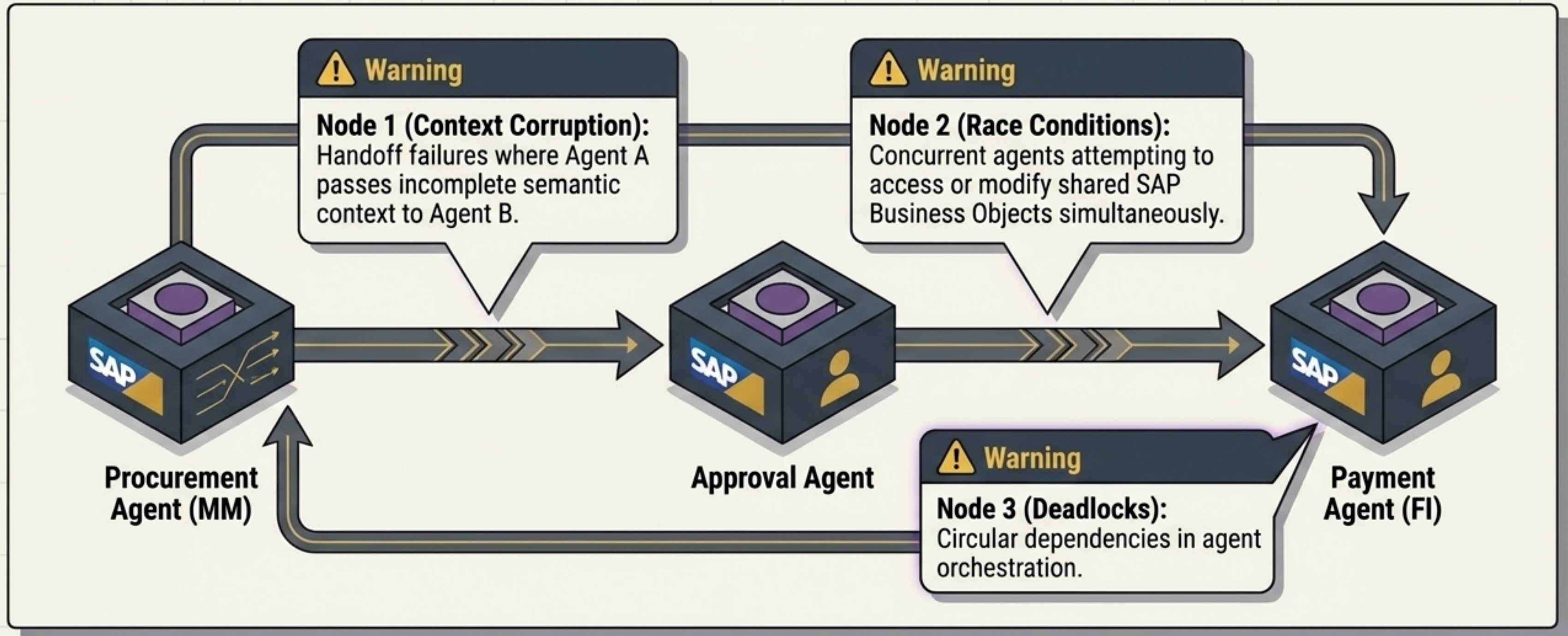
Node 16

Step 12: Drift Detection & Maintenance



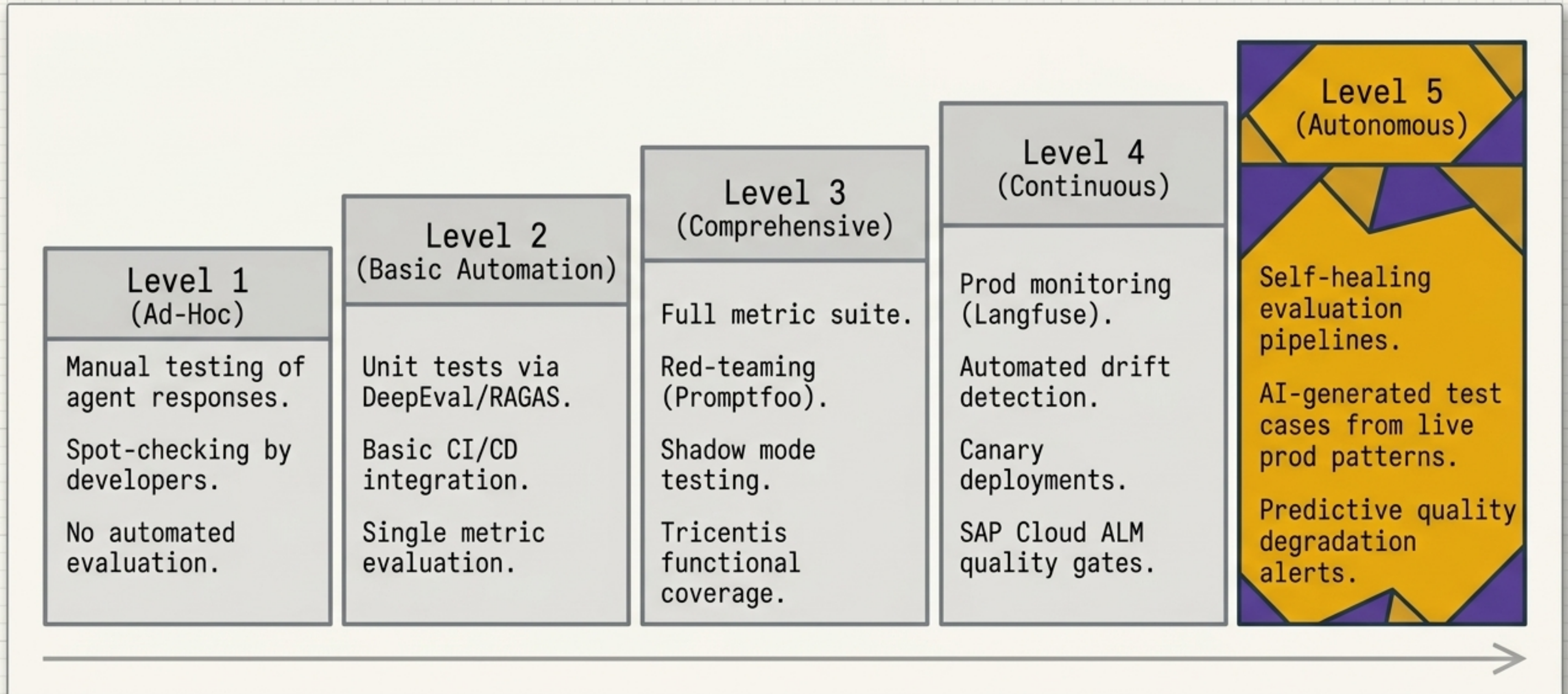
Monitor RAG embeddings and agent outputs for quality degradation over time. Leverage Tosca's self-healing tests and LiveCompare's SAP change impact analysis to reduce maintenance overhead.

Advanced Scenario: Multi-Agent Chaos Engineering



Testing Strategy: Requires chaos engineering—intentionally injecting failures and executing concurrent load testing via NeoLoad to validate system resilience.

The Enterprise Agent Testing Maturity Model



Architecture Strategy: The Hybrid Stack Recommendation

Build (Open Source)

- **Pros:** Full control, highly customizable metrics (DeepEval/RAGAS).

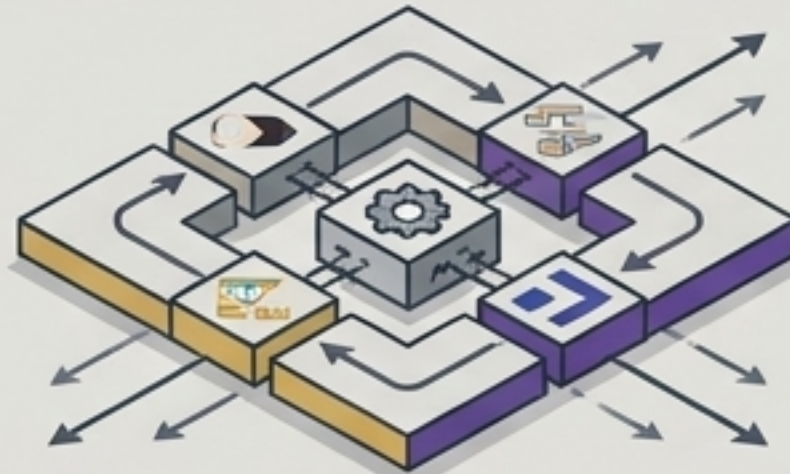


- **Cons:** High integration effort, maintenance burden.



The 2026 Hybrid Recommendation

- Use **Open Source** (DeepEval, Promptfoo, Langfuse) for raw mathematical evaluation, tracing, and red-teaming.



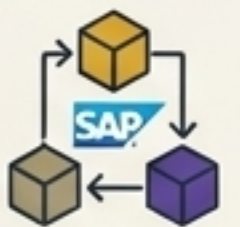
- **Wrap it in Enterprise Platforms** (SAP Cloud ALM, Tricentis Tosca/qTest/NeoLoad) for CI/CD execution, test management, UI/API automation, and strict governance.

Buy (Pure Enterprise)

- **Pros:** Faster time-to-value, managed infrastructure.



- **Cons:** Vendor lock-in, slower adoption of bleeding-edge metrics.



In the Autonomous Enterprise, quality engineering is the ultimate control plane. Trust, but verify probabilistically.