lyzr

Agent Architect Cohort 1

Business Case of Agents

Day 3

lyzr

Recap - Day 2 : Agent Architecting

- Business Requirements & AI Agents
- Core Components of an AI Agent (Tools, Functions, Extensions etc.)
- Agent Communication & MCP
- Safe & Responsible AI
- Secure Deployment of AI Agents
- Improving an AI agent
- Model Fine Tuning

lyzr

Agent Architect Cohort

Day 3 : Business Case of Agents

- Recap on MCP & A2A
- AgentOps
- Agent Eval
- Responsible & Safe AI
- Hallucination Management
- Agent Data Governance
- Cost-benefit analysis for agent adoption
- Writing Test Cases for AI Agents
- Access to 100+ Agent Use Cases & Blueprints

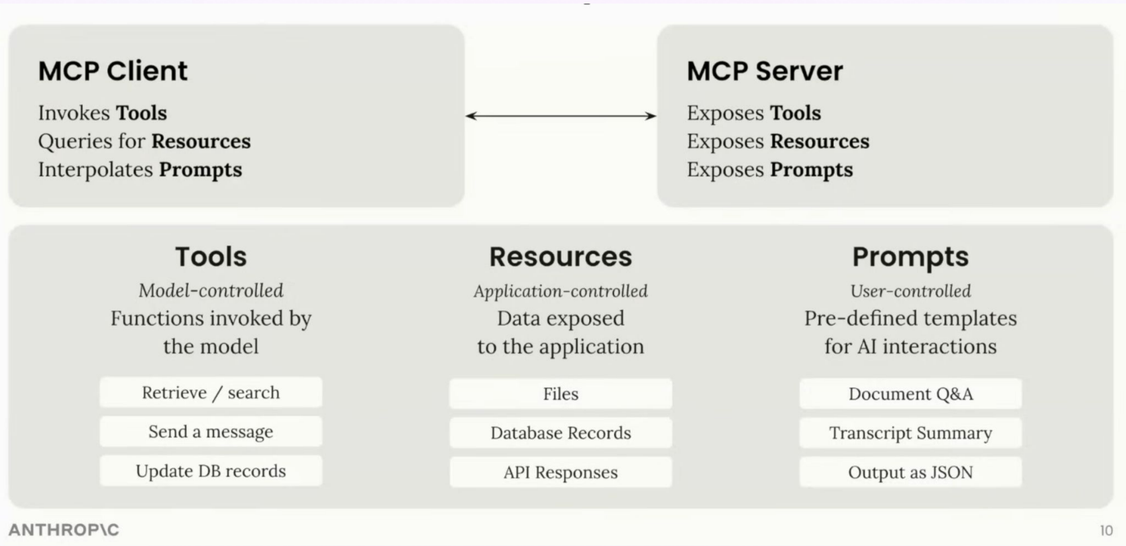
Agent Communication : MCP Server

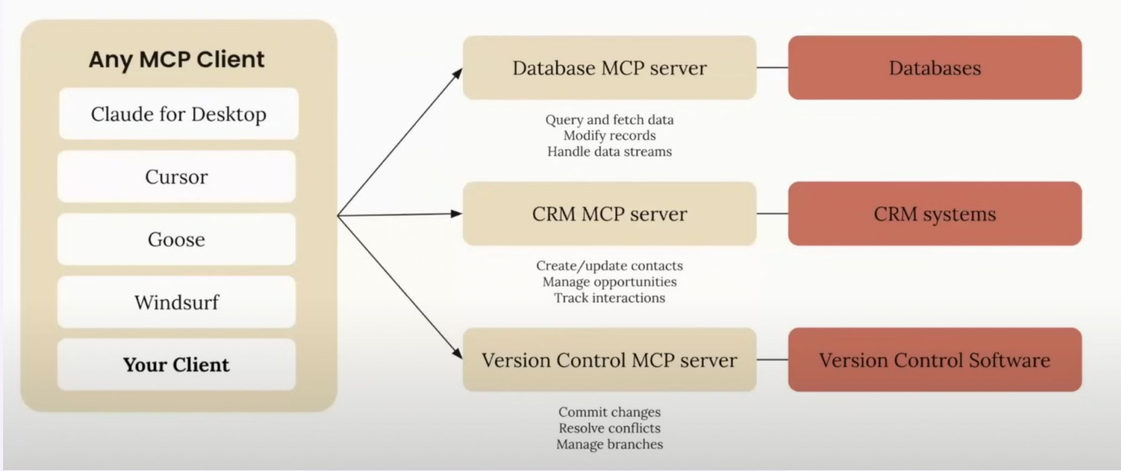
- **What it is:**
  An **MCP Server** acts as the orchestration backbone for agent systems—managing workflows that involve multiple components like LLM calls, tool invocations, memory access, and inter-agent communication.

- **How it works:**
  It runs a **step-based execution loop**, deciding at each step whether to call a model, invoke a tool, trigger another agent, or pause based on predefined logic and state.

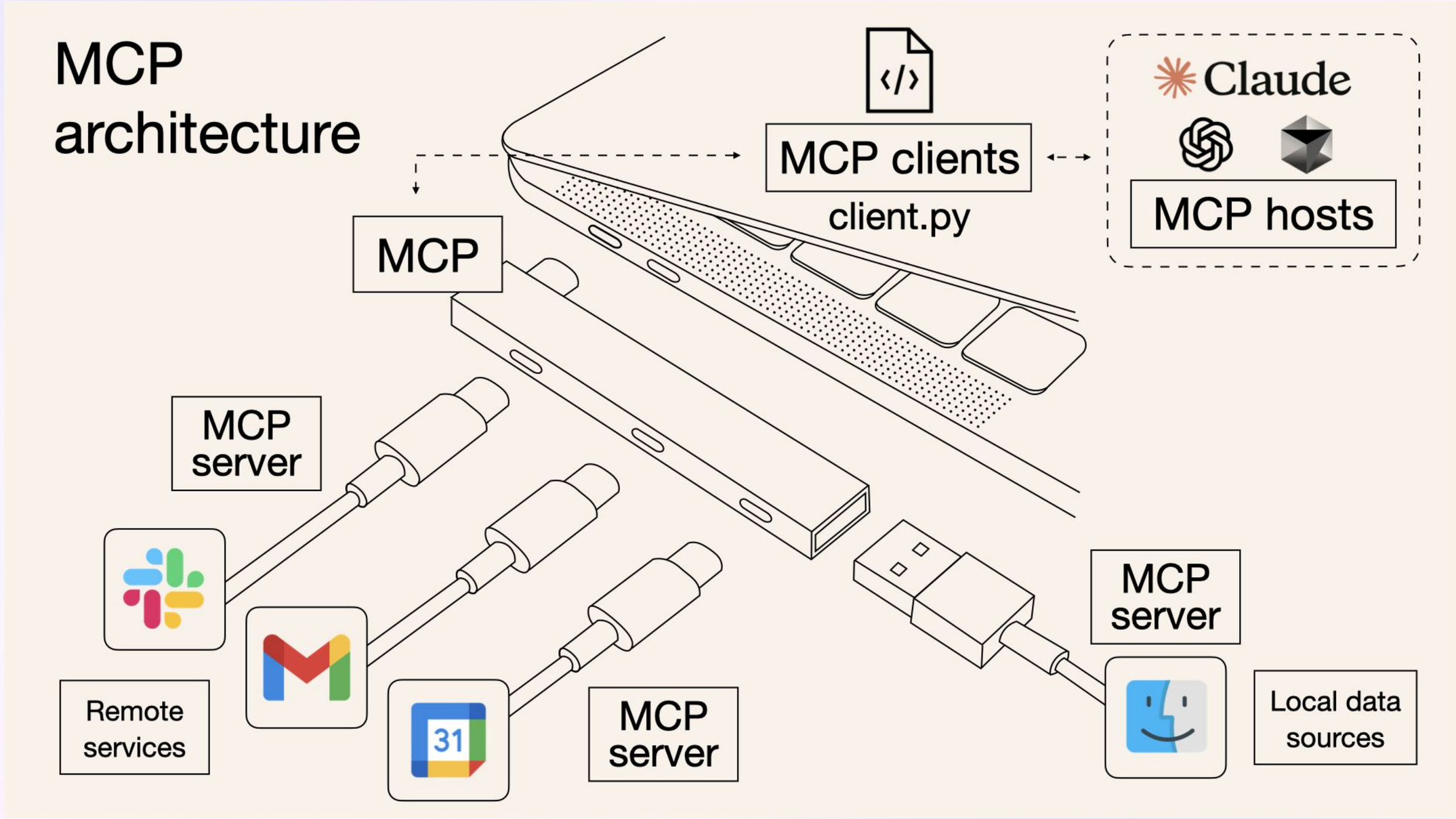
- **Why it's needed:**
  Without it, you'd need to hard-code orchestration in scripts or rely on brittle, prompt-only logic. MCP centralizes and abstracts this, making agent workflows **modular, reusable, and deterministic**.

- **Key benefits:**
  - Enables **complex multi-turn workflows**
  - Ensures **stateful, policy-driven execution**
  - Supports **robust error handling, logging, and retries**
  - Makes agents truly **autonomous and production-ready**





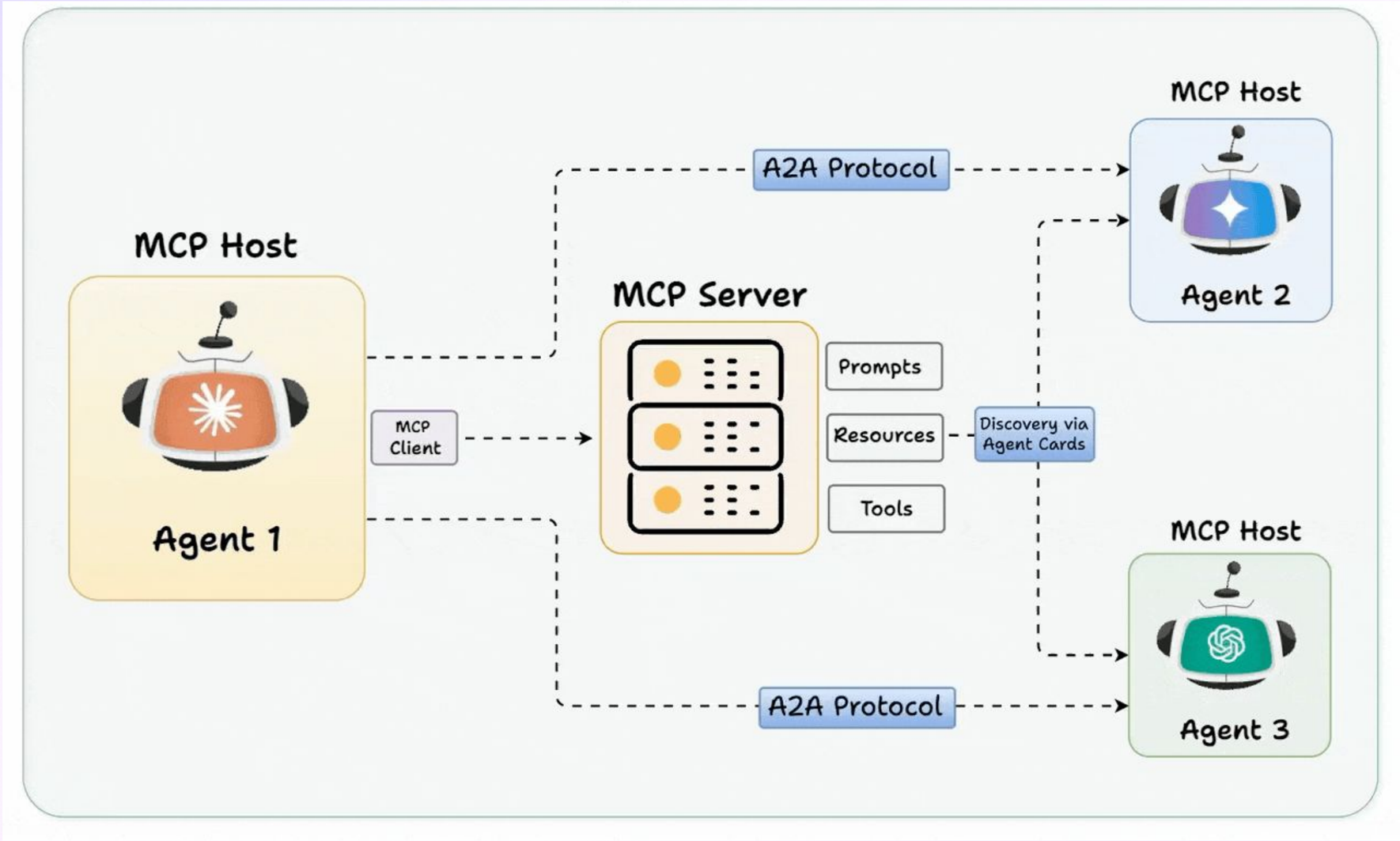
MCP architecture



Agent Communication : A2A Agent to Agent

- **A2A communication** enables agents to collaborate by exchanging messages, tasks, and context, forming the foundation of multi-agent coordination.
- It's essential for **distributed systems**, allowing agents to operate asynchronously and share responsibilities across environments.
- Remote agent protocols manage **task delegation, event triggers, and negotiation**, using structured messaging (e.g., JSON, Protobuf) over channels like gRPC or WebSockets.
- Key to success: **shared context, reliable message routing, and security**, ensuring agents act as a unified, intelligent system.

A2A v/s MCP


MCP Host
Agent 1
MCP Client
A2A Protocol
MCP Server
Prompts
Resources
Tools
Discovery via Agent Cards
MCP Host
Agent 2
MCP Host
Agent 3
A2A Protocol

AgentOps

- **What is AgentOps:**
  A discipline within **GenAIOps** focused on operationalizing AI agents - ensuring they move from prototype to production reliably and efficiently.
- AgentOps incorporates standard MLOps/DevOps practices (version control, CI/CD, testing, logging, security, metrics).
- Its unique focus areas include:
  - **Internal and external tool management**: Handling APIs, access, and reliability.
  - **Agent brain prompt**: Managing the core goal, profile, and instructions.
  - **Orchestration**: Managing the flow, logic, and interaction between agents/steps.
  - **Memory**: Handling short-term and long-term context.
  - **Task decomposition**: Managing how agents break down complex problems.
- It's a blend of people, processes, and technologies

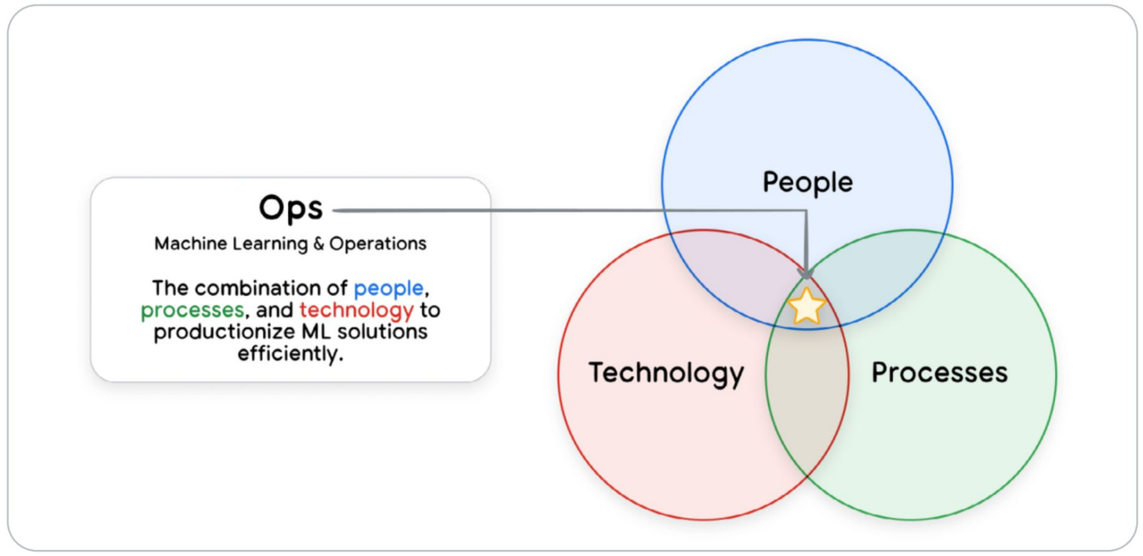


Figure 2. Each of these "Ops" are about technology, processes, and people[14]

AgentOps

- **Why it matters:**
  While building agents is quick, productionizing them at scale demands structured practices for **quality, reliability, and monitoring**.

- **How it fits in:**
  AgentOps builds on **DevOps and MLOps** principles and complements other GenAIOps areas like
  - FMOps: For foundational models.
  - PromptOps: For managing prompts.
  - RAGOps: For operationalizing RAG solutions.

- **Key focus:**
  Managing agent lifecycle, performance, feedback loops, and safe deployment in **live, evolving environments**.

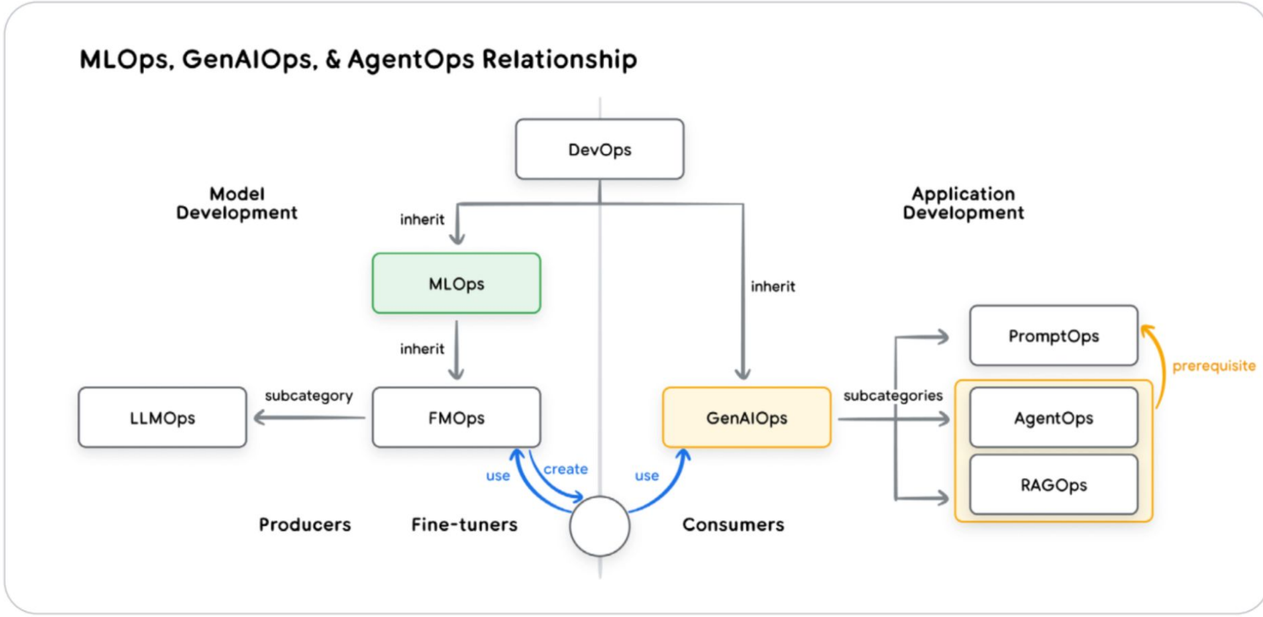


Figure 1. Relationship between DevOps, MLOps, and AgentOps.[13]

AgentOps starts with tracking agent logs at a granular level.

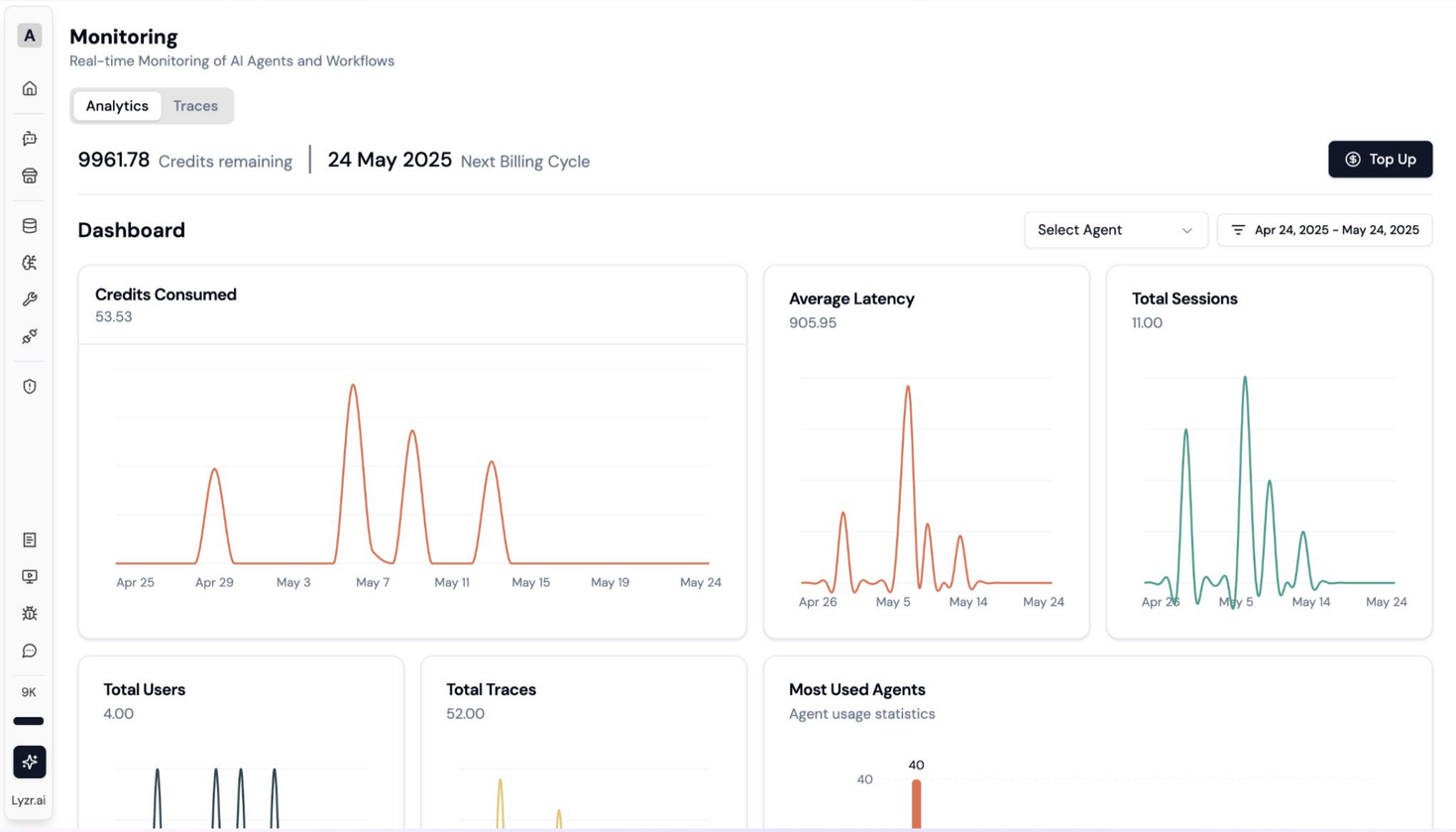
Date	Model	Credits Consumed	Input Message	Output Message
May 27, 2025, 03:15 PM	gpt-4o	17.66	This Nondisclosure Agreement (“Agreement”) is made and...	Upon comparing the provided NDA with the standard NDA, the...
May 27, 2025, 03:14 PM	claude-3-7-sonnet-latest	28.04	I would like to write an email to the customer and ask for a follo...	# Follow-up Call Request Hi Mohit, Thank you for taking the time...
May 27, 2025, 03:05 PM	gpt-4o	7.87	how do you handle Responsible AI?	Hi, my name is Lyra. I am an intelligent chat assistant. I'm here to...
May 27, 2025, 03:05 PM	text-embedding-ada-002	0.01		
May 27, 2025, 03:05 PM	gpt-4o	7.58	how do you handle Responsible AI?	Lyzr incorporates Responsible AI directly into its core agent...
May 27, 2025, 03:05 PM	text-embedding-ada-002	0.01		
May 27, 2025, 03:04 PM	gpt-4o	1.43		Hi, my name is Lyra. I am an intelligent chat assistant. I'm here to...
May 27, 2025, 01:42 PM	claude-3-7-sonnet-latest	11.31	custom platform	Thank you for sharing that you're using a custom platform. This...
May 27, 2025, 01:41 PM	claude-3-7-sonnet-latest	5.24	personalized recommendations based on user requerst	Thank you for sharing that you're looking to build personalized...
May 27, 2025, 01:41 PM	claude-3-7-sonnet-latest	4.83	SKU details	Thank you for sharing that you have SKU details available. This is ...

Observability & Traceability

- **Credits Consumed:** Operational cost tracking for budgeting and ROI.
- **Average Latency:** Detects performance bottlenecks in agent execution.
- **Sessions & Traces:** Measures agent activity and engagement flow.
- **Most Used Agents:** Surfaces high-impact agents for optimization or scaling.
- **Total Users:** Gauges reach and adoption.

Why It's Critical:

- **Operational efficiency** (spot slow or overused agents)
- **Cost control** (credit usage tracking)
- **Trust and reliability** (latency + success tracking

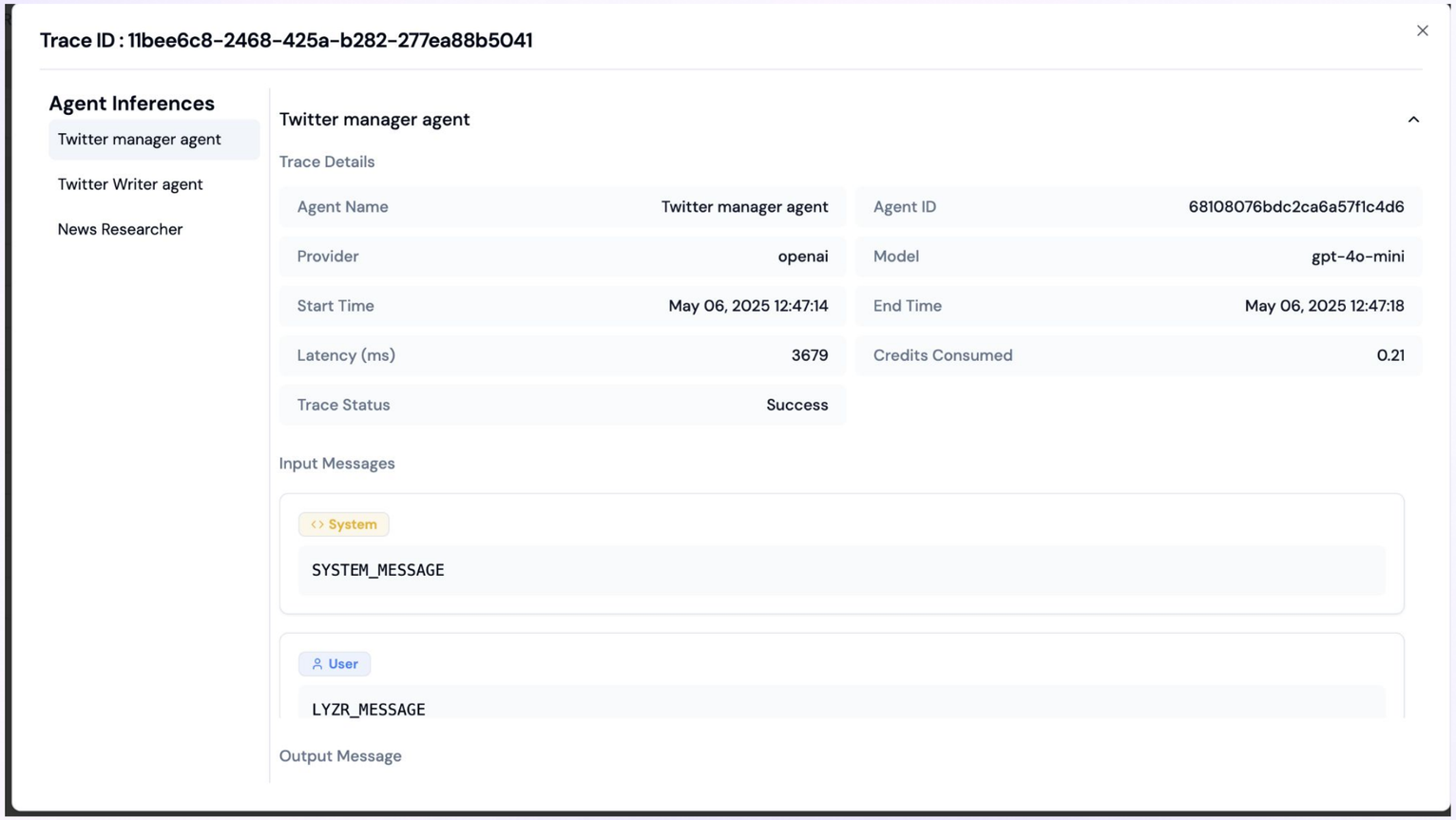


Observability & Traceability

- **Agent Identity:** Shows which agent ran, its provider, and specific model used.
- **Execution Metrics:** Captures **latency**, execution time window, and cost.
- **Message Flow:** Displays **system-level prompts**, user instructions, and output messages—allowing you to audit the interaction end-to-end.
- **Multi-Agent Visibility:** Shows related agents, helping trace coordination in orchestration flows.

Why it matters?

- **Debugging**: Pinpoint failure - wrong tool, bad output, or hallucination origin.
  Accountability: Attribute actions to specific agents, prompts, or configurations.
- **Optimization**: Analyze latency, cost, retries, tool selection for continuous tuning.
- **Governance**: Essential for HITL review, compliance, and trust in enterprise workflows.



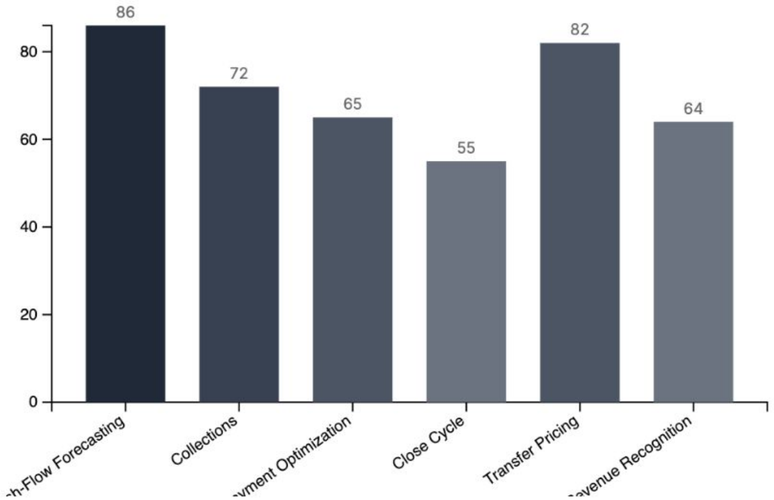
Enrich the logs to ‘user specific’ analytics

Analytics

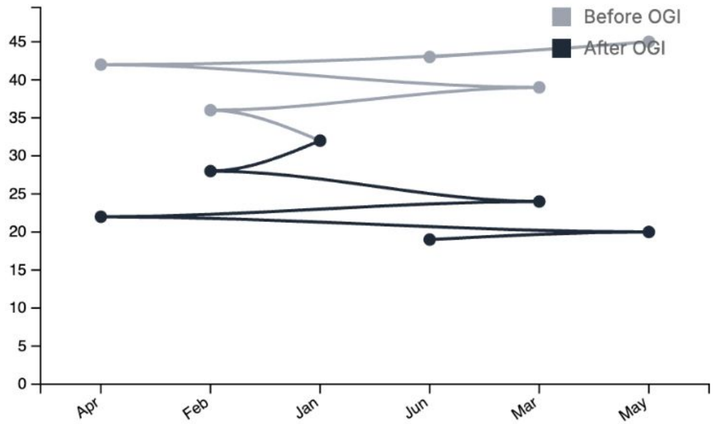
Measured impact of OGI agents across key financial metrics and operational efficiency indicators.

Average DSO Reduction	Close Cycle Time	Working Capital Yield	Automation Rate
14.5 days	**5.2 days**	**+4.7%**	**73%**
↑ 24% vs. Previous Period	↑ 48% vs. Previous Period	↑ 3.2% vs. Previous Period	↑ 35% vs. Previous Period

Time Savings by Agent

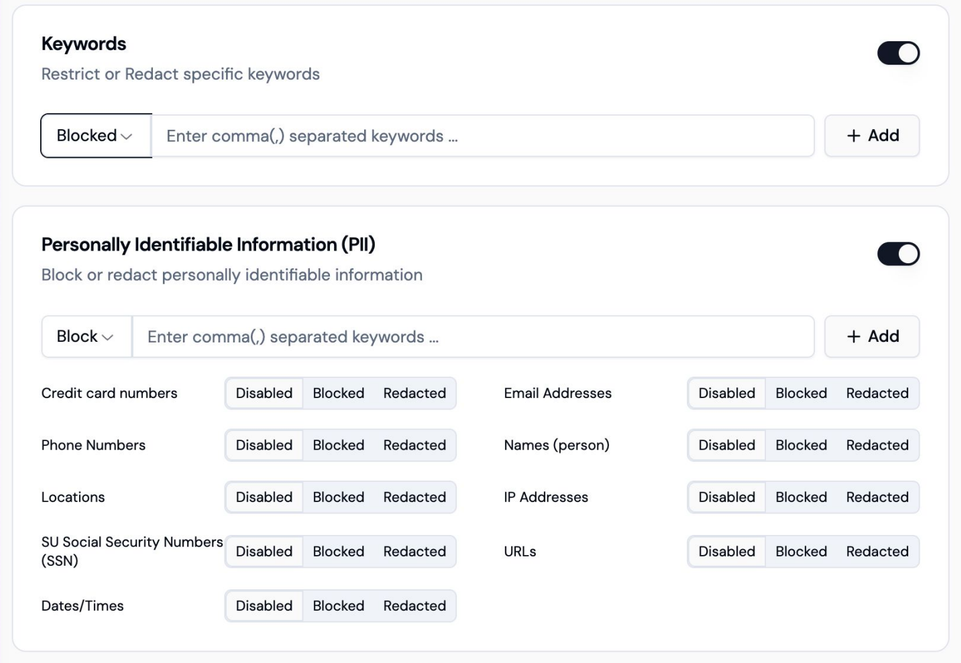


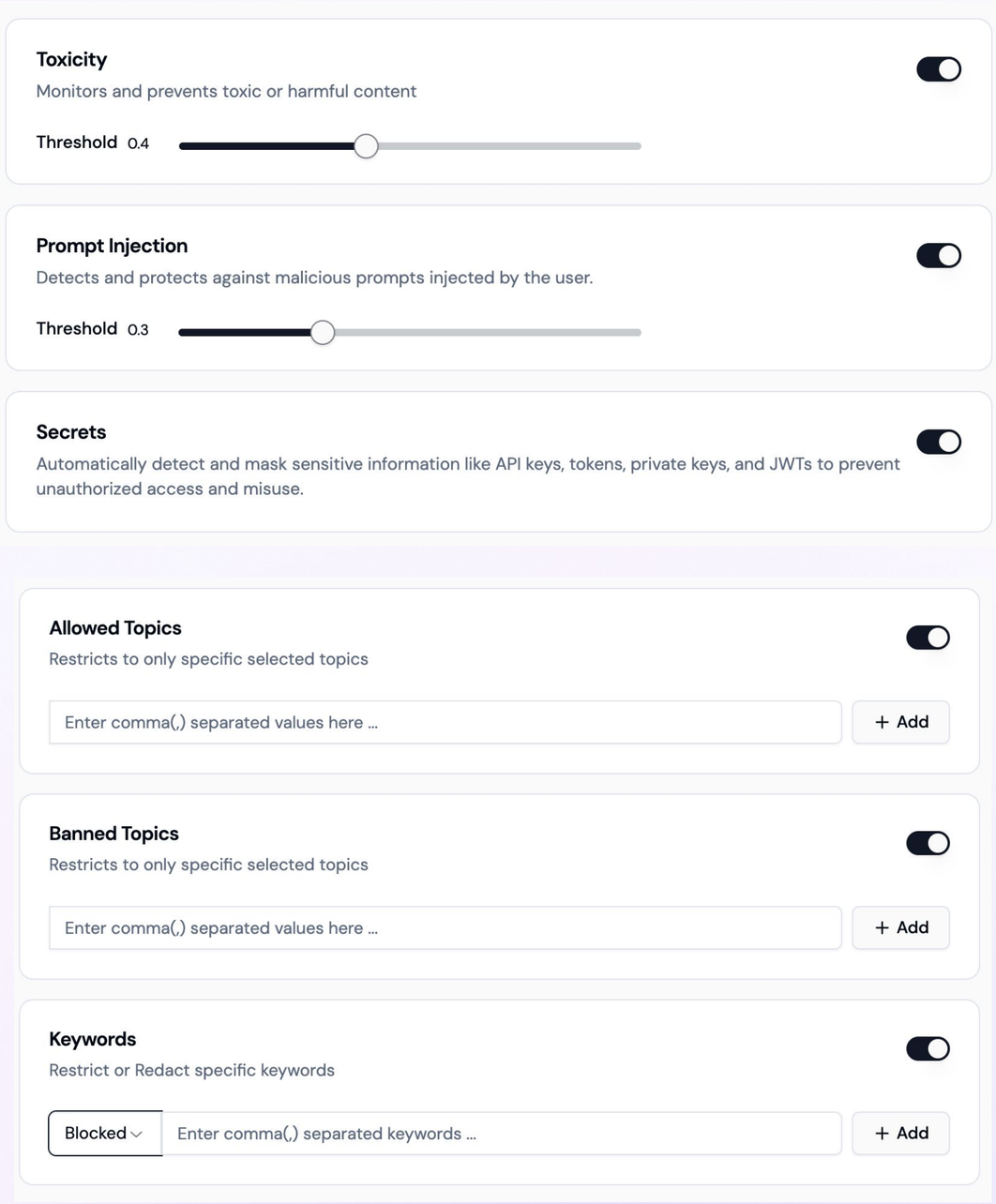
Working Capital Impact



Responsible & Safe AI

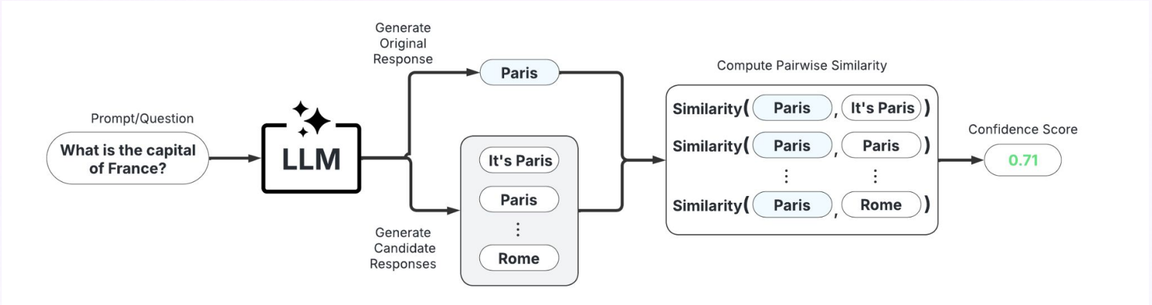
- Toxicity Controller
- Prompt Injection Manager
- PII Redaction
- Bias & Fairness Detection
- Secrets Manager
- Restricted Keywords
- Restricted Actions
- Custom Policies



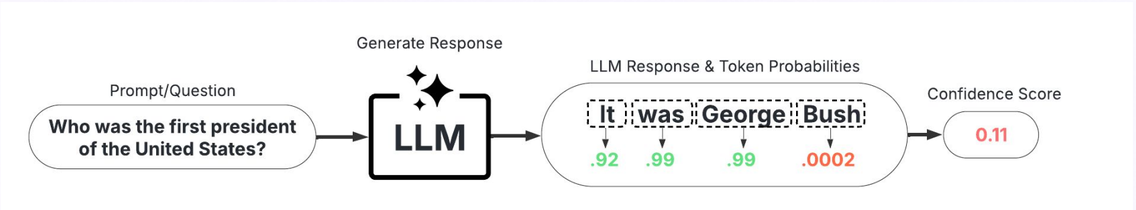


Hallucination Management

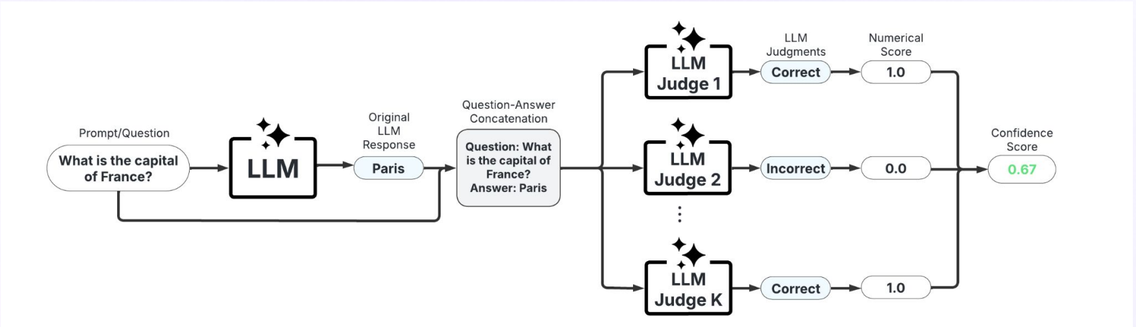
- **Reflection :** Self-evaluation by the model—asking, “Did I answer correctly?” Promotes internal consistency and iterative improvement.
- **Groundedness :** Measures how well an AI response is anchored in verifiable source data (e.g., via RAG), reducing hallucination.
- **Context Relevance :** Assesses whether the response is appropriate to the input, using task-specific or retrieval-based grounding for alignment.
- **Black Box Scorers :** Treat the model as a black box—output is judged by external heuristics or reference comparisons without insight into internal logics.
- **White Box Scorers :** Evaluate based on internal reasoning steps, such as chain-of-thought traces or intermediate tool use, offering interpretable scores.
- **LLM as a Judge :** Uses another LLM (often fine-tuned) to rate or compare outputs—scalable but sensitive to prompt design and bias.
- **Ensemble Scorers :** Combine multiple scoring methods (e.g., relevance, groundedness, factuality) for a more holistic and robust evaluation.
- **NeuroSymbolic AI :** Merges neural networks (LLMs) with symbolic reasoning—enabling agents to **reason, generalize, and explain decisions** with greater transparency and control.



Black Box Scorers



White Box Scorers



LLM-as-judge

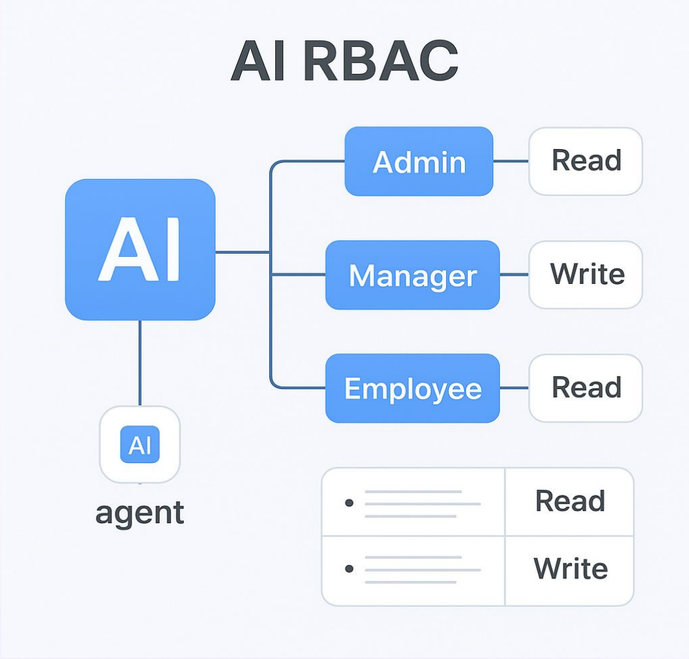
Agent Data Governance & Agent Entitlement Policy

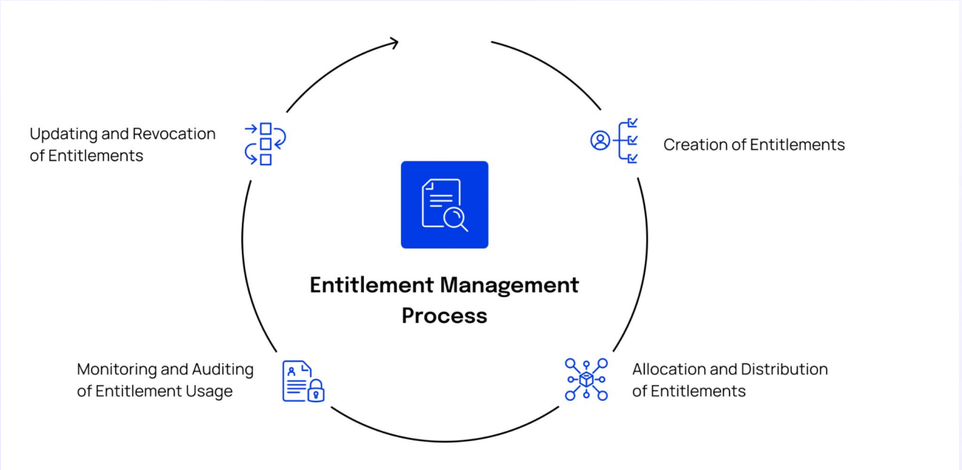
An AI Agent Entitlement Policy specifies:

- **Access Rights**: What data, tools, or systems an AI agent can access.
- **Action Permissions**: What operations an agent is permitted to perform.
- **Operational Boundaries**: The contexts or conditions under which an agent can act.
- **Delegation Authority**: Whether and how an agent can act on behalf of users or other systems.

Why Entitlement Policies Matter

- **Security & Compliance:** Enforce least privilege to prevent unauthorized access and reduce breach risks.
- **Accountability:** Define agent-level permissions to enable auditability and traceability.
- **Operational Integrity:** Ensure agents act only within their approved scope, preserving system stability.
- **User Trust:** Transparent boundaries build confidence in agent behavior and governance.





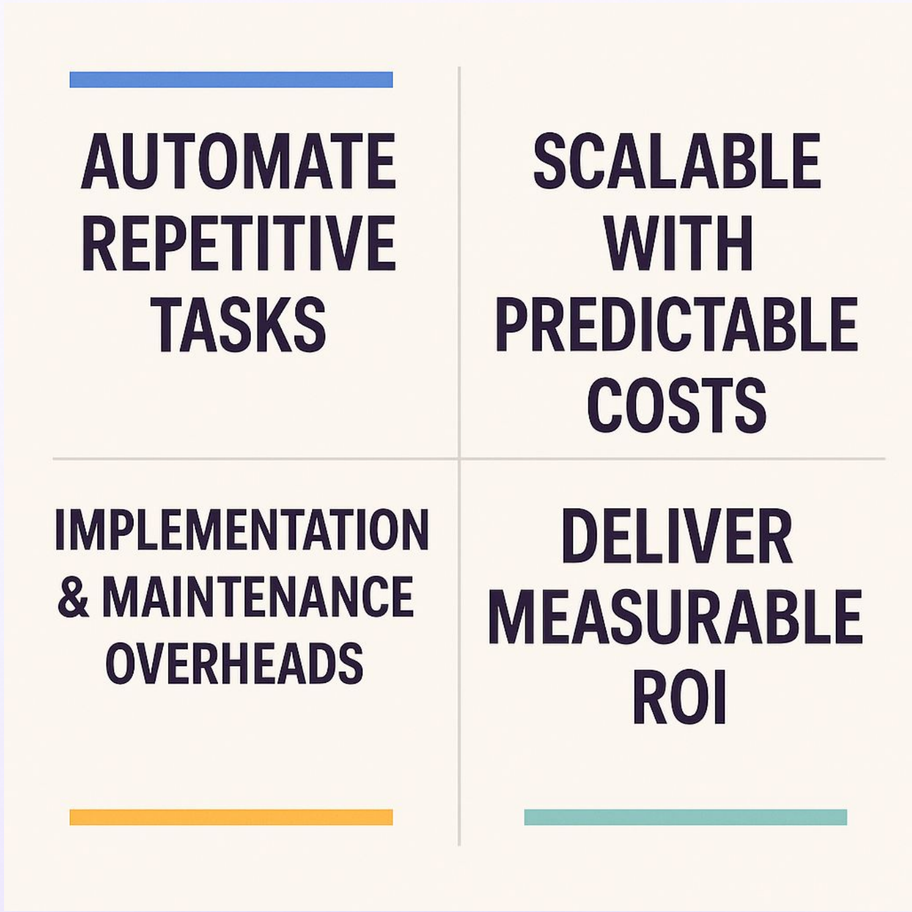
Cost-benefit analysis of AI agent adoption

- **Operational Efficiency & Cost Savings:**
  Agents automate repetitive, high-volume tasks - reducing labor costs, eliminating human error, and enabling 24/7 execution without additional headcount.

- **Scalability with Predictable Costs:**
  Unlike human teams, agents scale horizontally with demand. Cost per task remains stable, even as workload increases—ideal for customer support, research, and internal operations.

- **Implementation & Maintenance Overheads:**
  Adoption requires upfront investment in agent design, tool integration, inference costs (e.g., LLM usage), and ongoing monitoring, testing, and guardrail enforcement.

- **Strategic Fit & ROI Triggers:**
  Agent adoption is most impactful when tasks are repetitive, latency-sensitive, or require consistent logic execution—delivering measurable ROI through speed, consistency, and cost control.



Writing Test Cases

For Agents

- **Golden Path Tests:**
  Validate standard, expected inputs to ensure the core agent logic functions as intended.

- **Edge Case Tests:**
  Cover rare or boundary scenarios to test how the agent handles unusual input or environmental conditions.

- **Negative Tests:**
  Feed invalid, malicious, or policy-violating inputs (e.g., prompt injections, off-topic queries) to assess guardrail effectiveness and system resilience.

- **Tool Invocation Tests:**
  Confirm tools are correctly selected, invoked with proper parameters, and that the agent handles their outputs appropriately.

For Workflows

- **End-to-End & Flow Testing:**
  Validate the complete user journey across agents, ensuring smooth transitions, correct outputs, and cohesive orchestration logic.

- **Delegation & Handoff Testing:**
  Ensure agents accurately pass **state, context, and data** during handoffs, and that recipient agents respond appropriately.

- **Routing & Logic Branch Testing:**
  Test all conditional paths and branching logic—especially with router or manager agents - to confirm the system follows the correct flow under varying inputs.

- **Failure & Concurrency Testing:**
  Simulate agent failures and parallel executions to test **resilience**, **retry behavior**, and **data consistency** in concurrent environments.

What KPIs to Track for Agent Success?

1. Business Impact

- **ROI / Cost Savings** – Measurable value from automation
- **User Adoption & Engagement** – Frequency and breadth of agent usage
- **CSAT / NPS** – User satisfaction with interactions

2. Effectiveness & Quality

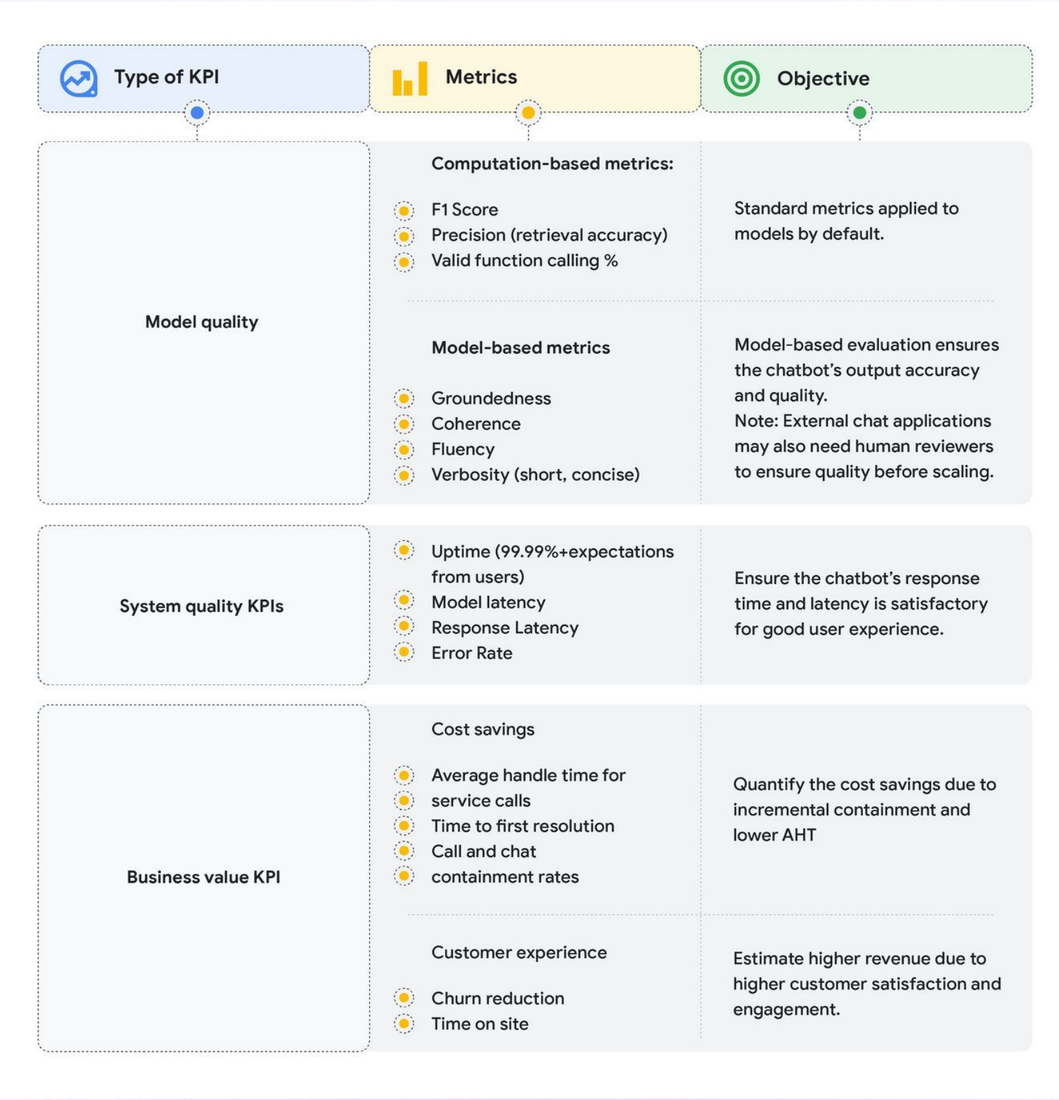
- **Goal Completion Rate (GCR)** – % of tasks completed end-to-end
- **Task Success Rate (TSR)** – % of successful subtasks
- **Accuracy / F1 Score** – Response correctness (via evals)
- **Hallucination & Relevance Scores** – Quality and truthfulness of responses

3. Efficiency & Operations

- **Latency & Cost per Interaction** – Speed and cost performance
- **Tool Call Success Rate** – % of tool invocations that succeed
- **Error Rate** – % of failed or broken interactions

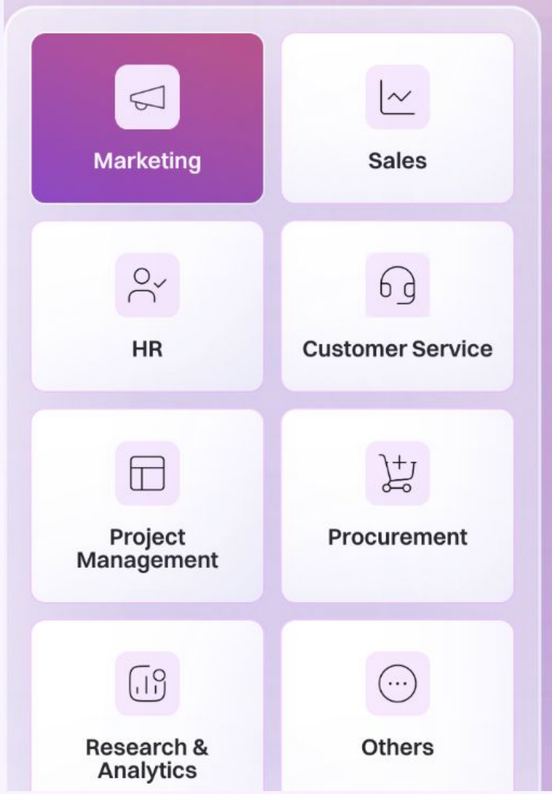
4. User Experience & Trust

- **Human Escalation Rate** – % requiring fallback to human agents
- **Feedback Signals** – Thumbs up/down or similar
- **Guardrail Trigger Rate** – Safety/policy enforcement frequency



Use Cases & Blueprints

Lyzr Agent Usecases



Category	Agent Name	Description	Complexity	Instant Demo Readiness	Production Go-Live Time
Marketing	Blog Writer	A multi-agent system that automates content creation and blog writing using Stanford Innovation Labs STORM algorithm.	Medium	Yes	6-8 weeks
	Email Marketer	An autonomous email marketer that does detailed prospect research, identifies personalization points, writes emails, follows up, and books appointments.	Low	Yes	4-6 weeks
	LinkedIn Marketer	Repurposes internal blog posts and other research content into LinkedIn posts and posts automatically across the company's LinkedIn page.	Medium	No	6-8 weeks
	Twitter Posting Agent	Repurposes internal blog posts and other research content into Twitter posts and posts automatically across the company's Twitter page.	Low	No	4-6 weeks
	SEO Optimizer	Analyzes the supplied set of SEO keywords and improves the content, including blogs, case studies, and technical whitepapers to have a better SEO score.	Medium	No	6-8 weeks
	Lead Enrichment	Automatically enhances lead profiles with relevant business and contact information from multiple sources.	Medium	Yes	6-8 weeks

Agent Architect Cohort 1

Agent Architecting

Questions?

lyzr

See You Tomorrow!

Day 4 Focus: Building on Lyzr Studio!

- Building an agent on Lyzr
- Configure new tools and enable tool calling
- Enable core modules (memory, knowledge base)
- Build a knowledge base and try various retrieval types
- Build Responsible AI policy and add to an agent
- Launch the agent as an standalone app on Lyzr
- Build managerial orchestration
- Build DAG orchestration
- Build Hybrid orchestration