Figures

Figure 1. Architecture of the generative DevOps AI system (100) inside a cloud account environment (105). Labeled components include in-account micro-agents (110), RL orchestrator/optimizer (120), Policy Weaver guardrail module (130), federated learning coordinator (140), cloud provider API interface (150) with cloud resources (151) and metrics (152), and a private AI model endpoint (160) for foundation model inference, all within the user's cloud environment.

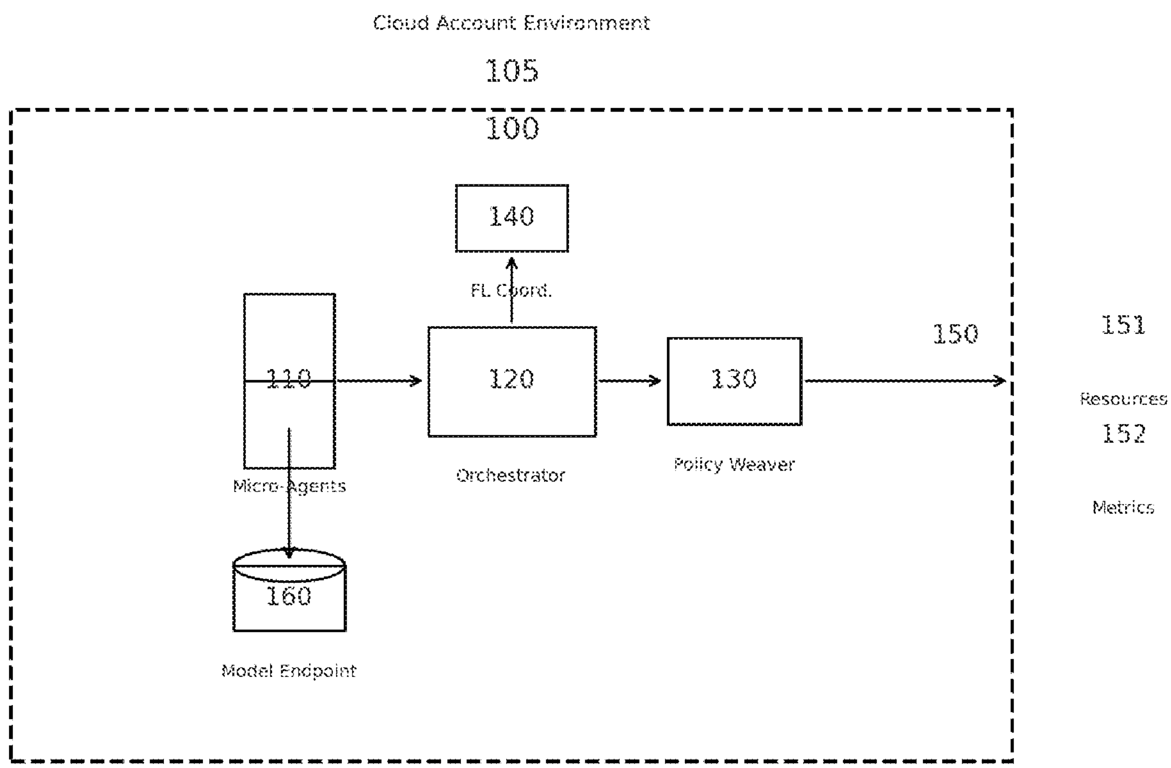


FIG. 1

Figure 2. Tri-Vector RL optimizer (200) in operation. The RL agent evaluates potential actions against three metrics – cost (210), performance (220), and compliance risk (230) – and receives

aggregated reward signals from these dimensions to balance its decisions. Arrows illustrate how the optimizer considers each metric's feedback when choosing actions.

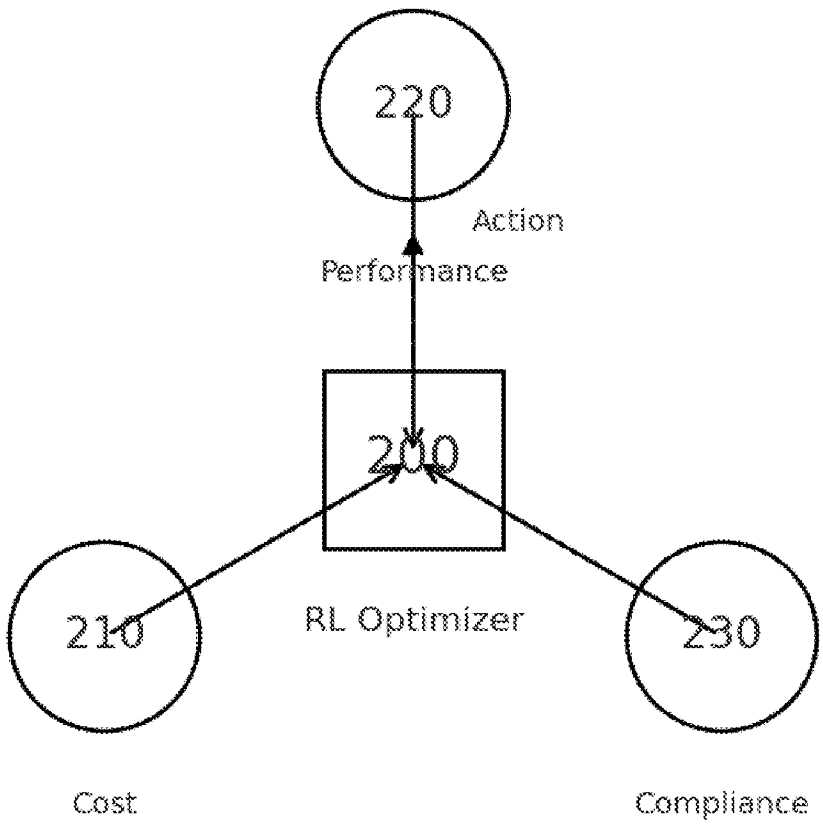


FIG. 2

Figure 3A. The Policy Weaver (300) intercepting an AI-generated Infrastructure-as-Code change (310) and applying rule checks (policy rules set 132) before deployment. The proposed change is fed into the Policy Weaver, which evaluates it in real time against multiple guardrail policies (e.g. tagging requirements, IAM rules) to ensure compliance prior to execution.

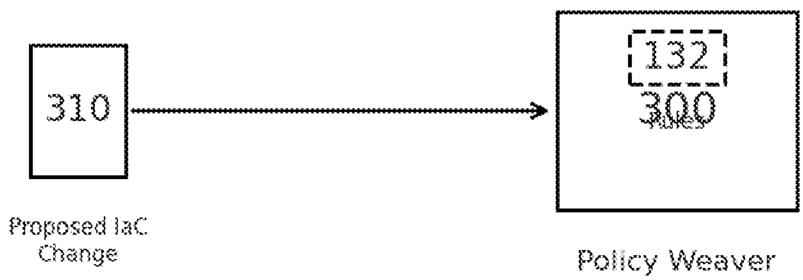


FIG. 3A

Figure 3B. Zoom-in on a specific guardrail check (320) performed by the Policy Weaver. In this example, the check verifies a required condition (such as mandatory resource tags or least-privilege IAM permissions). The diagram shows the outcome being **allowed** (right arrow) if the policy is satisfied, or **blocked** (left arrow, marked with "X") if the proposed action violates the guardrail.

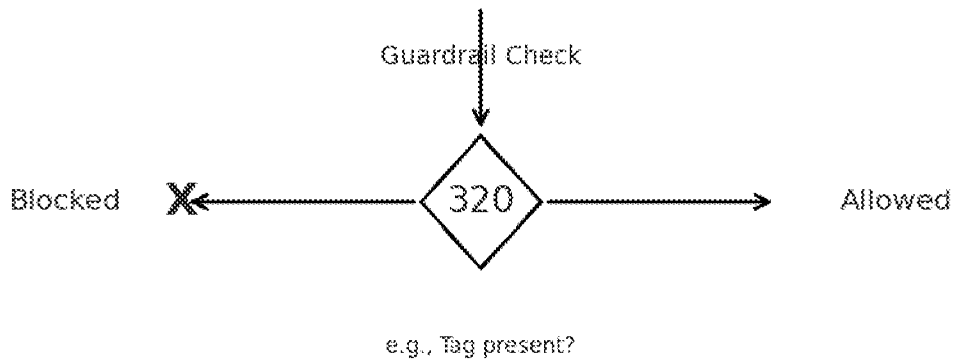


FIG. 3B

Figure 4.1. Federated learning across multiple accounts. Account A (401) and Account B (402) each train local AI models and periodically send encrypted model updates (411, 412) from their in-account micro-agents to a central aggregator (420). These updates (e.g. gradient or parameter changes) are transmitted securely (encrypted) to preserve privacy.

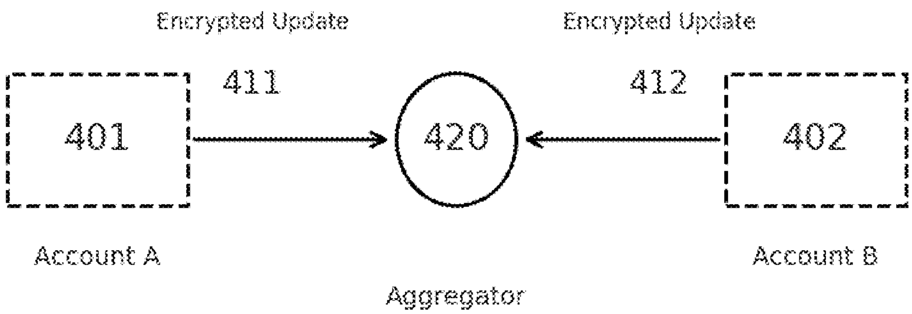


FIG. 4.1

Figure 4.2. The central aggregator (420) performs a secure aggregation process (430) on the incoming updates and produces an improved global model. It then distributes the updated model parameters back to each account's deployment as model updates (shown as 411 to Account A and 412 to Account B). This privacy-preserving aggregation ensures all agents learn collectively without sharing raw sensitive data.

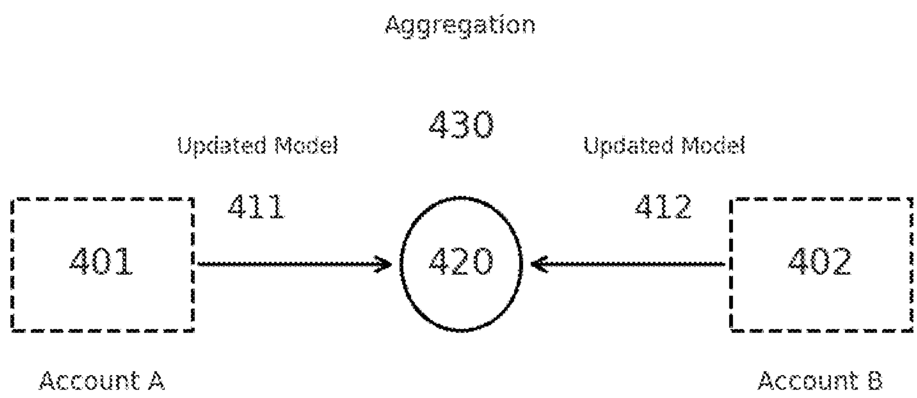
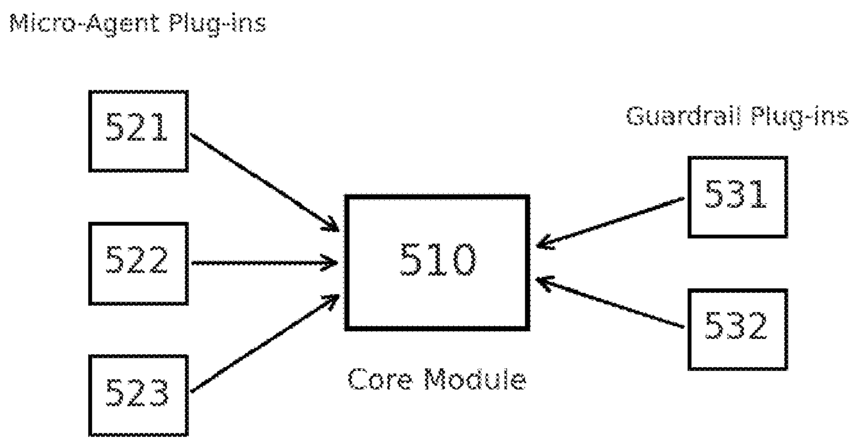


FIG. 4.2

Figure 5. Plug-in architecture of the system (500). The core orchestration module (510) loads extensible plug-ins for micro-agents and policies. On the left, multiple micro-agent plug-ins (521, 522, 523) integrate into the core, adding new DevOps capabilities. On the right, guardrail policy plug-ins (531, 532) attach to the Policy Weaver, allowing custom compliance rules. This modular design enables easy extension of the system's capabilities without altering the core



Micro-Agent Plug-ins
521
522
523
510
Core Module
Guardrail Plug-ins
531
532

FIG. 5
