

A LARGE LANGUAGE MODEL DRIVEN AGENT FRAMEWORK FOR AUTONOMOUS, SECURE, AND SCALABLE BIG DATA MANAGEMENT IN HETEROGENEOUS CLOUD ECOSYSTEMS

Syed Muhammad Raza Zaidi^{*1}, Ghulam Muhammad Akhonzada², Farkhanda Jabeen³,
Yasir Anwar⁴

^{*1}Department of Computer Systems Engineering NED University of Engineering and Technology, Karachi, Sindh, Pakistan

²Department of Software Engineering COMSATS University Islamabad, Islamabad Campus, Pakistan

³Department of Computer Science Sardar Bahadur Khan Women's University, Sub-Campus Khuzdar, Pakistan

⁴Department of Computer Systems Engineering and Sciences Balochistan University of Engineering and Technology, Khuzdar, Pakistan

^{*1}mrza28067@gmail.com, ²ghulamakhonzada@gmail.com, ³farkhandabajoi@gmail.com,
⁴yasiranwar921@gmail.com

DOI: <https://doi.org/10.5281/zenodo.22867784>

Keywords

LLM agents; federated learning; multi-cloud orchestration; big data management; AI-driven orchestration; privacy-preserving computing; zero-trust security; scalable architectures; reinforcement learning; cloud security; data governance.

Article History

Received: 30 January 2026

Accepted: 15 March 2026

Published: 31 March 2026

Copyright @Author

Corresponding Author: *

Syed Muhammad Raza Zaidi

Abstract

The proliferation of heterogeneous multi-cloud and edge environments has made big-data management increasingly difficult to automate, secure, and scale using conventional rule-based orchestration. This paper proposes a layered conceptual and architectural framework in which large language model (LLM) based agents act as the reasoning and coordination layer for autonomous data placement, access-control decisioning, schema reconciliation, and anomaly response across heterogeneous cloud ecosystems. The framework couples an LLM planner-negotiator-critic agent pipeline with a zero-trust policy engine, a reinforcement-learning-based resource scheduler, and a federated-learning layer for privacy-preserving cross-silo model updates. We describe the system model, the architecture, the interaction protocol between agents and the enforcement layer, and the agent roles in detail, and we present illustrative, simulated scenarios and figures to reason about expected scalability, cost, compliance-risk, and policy-conformance trade-offs. We position the proposal against existing literature on LLM-based autonomous agents, federated learning, zero-trust security, and reinforcement-learning-driven cloud scheduling, and we outline the empirical evaluation, red-teaming, and benchmarking work required before the framework could be validated in production settings. We further discuss data-governance and regulatory-compliance implications, a taxonomy of agent-specific threats, and a research agenda for follow-up empirical studies. The contribution of this paper is architectural and conceptual: it is intended as a foundation for subsequent empirical validation rather than as a report of completed experimental results.

1. INTRODUCTION

1.1 Motivation

Modern enterprises increasingly distribute data and workloads across multiple public clouds,

private data centers, and edge locations to satisfy performance, cost, redundancy, and regulatory requirements. Industry surveys consistently report

that the large majority of organizations now operate more than one cloud provider, yet a substantial share of these deployments fail to realize expected benefits because of interoperability gaps, inconsistent security postures, and fragmented data governance across providers [15][17]. As the volume, velocity, and variety of managed data continue to grow, the cost of manually curated, rule-based data-placement and access-control policy is rising faster than the engineering capacity available to maintain it.

Traditional big-data management pipelines rely on static rules, hand-tuned heuristics, or narrowly scoped machine-learning models to decide where data should live, how it should be replicated, and who should be allowed to access it. These approaches struggle to adapt to the scale, heterogeneity, and rapidly shifting workload characteristics of modern multi-cloud ecosystems. At the same time, recent advances in large language models (LLMs) have demonstrated that LLM-based autonomous agents can decompose complex, multi-step tasks, reason over tool outputs, and adapt their plans based on intermediate feedback [1], [19], which has motivated their use as controllers in increasingly complex automation pipelines, including infrastructure and data-operations tasks.

1.2 Problem Statement

We consider the problem of translating a natural-language or policy-specified data-management intent (for example, "replicate this dataset across two regions while minimizing egress cost and staying within GDPR data-residency limits") into a verified, auditable, and continuously enforced sequence of placement, access-control, and migration actions across a heterogeneous set of cloud and edge providers. This problem sits at the intersection of four historically separate research threads: (i) autonomous LLM-agent planning and self-correction, (ii) zero-trust, identity-centric access enforcement, (iii) reinforcement-learning-based resource scheduling, and (iv) privacy-preserving federated model improvement across organizational or tenant boundaries. No existing framework we are aware of integrates all four threads into a single, coherently layered control

path; this paper's contribution is to propose such an integration and to identify the properties it must satisfy.

1.3 Contributions

1. A layered architecture that separates LLM-agent reasoning, zero-trust policy enforcement, RL-based scheduling, and federated learning into distinct but coordinated planes, with an explicit interaction protocol between them;
2. A description of agent roles (planner, negotiator, critic) modeled on existing LLM-agent design patterns from the literature, adapted to the data-management domain, together with a formal system model and notation;
3. A discussion of security and privacy integration points (zero-trust continuous verification and secure federated aggregation) grounded in established standards and prior surveys, together with an agent-specific threat taxonomy;
4. A discussion of data-governance and regulatory-compliance implications for cross-border, multi-cloud data placement;
5. A set of illustrative, simulated scenarios and figures that frame testable hypotheses and metrics for a subsequent empirical evaluation, which we explicitly identify as future work rather than completed experimentation.

1.4 Paper Organization

Section 2 reviews related work across the four constituent research threads. Section 3 introduces the system model and problem formulation. Section 4 presents the proposed architecture and agent roles. Section 5 describes the interaction protocol and control-flow algorithms. Section 6 presents illustrative, simulated scenarios and figures. Section 7 discusses security and threat considerations. Section 8 discusses data-governance and compliance considerations. Section 9 discusses limitations and future work, and Section 10 concludes the paper.

2. Related Work

2.1 LLM-Based Autonomous Agents

LLM-based autonomous agents use a language model as a controller that plans, invokes tools, observes results, and revises its plan. Wang et al.

provide a comprehensive survey of the construction, application, and evaluation of LLM-based autonomous agents, organizing agent design around profile, memory, planning, and action modules [1]. Yao et al. introduce ReAct, a widely adopted paradigm in which an LLM interleaves verbal reasoning traces with concrete actions, allowing the model to maintain and revise a working plan while incorporating observations from its environment [19]; ReAct-style interleaving underlies the planner agent's task-decomposition behavior in our proposed framework. Follow-up surveys extend this analysis to multi-agent systems, in which several LLM-driven agents collaborate, negotiate, or critique one another to solve tasks that exceed the reliable scope of a single agent [2], [3]. Benchmarking work such as AgentBench evaluates LLMs specifically in their capacity as agents across diverse interactive environments [4], while more recent surveys catalogue the rapidly growing evaluation landscape for agentic systems [5]. Our proposed planner/negotiator/critic decomposition follows this literature's general pattern of separating planning from execution and validation, rather than proposing a fundamentally new agent architecture.

2.2 Federated Learning for Cross-Silo Privacy

Federated learning (FL) enables multiple organizations or data silos to collaboratively train shared models without centralizing raw data. McMahan et al. introduced the foundational federated averaging (FedAvg) algorithm for communication-efficient learning from decentralized data [20]. Yang et al. subsequently formalize horizontal, vertical, and federated transfer learning settings and survey applications across industries where data cannot be pooled for legal or competitive reasons [6]. Kairouz et al. catalogue open problems in federated learning, including communication efficiency, robustness to unreliable participants, and privacy accounting [21]. Subsequent surveys catalogue FL's communication efficiency, robustness, and fairness challenges [7], [8], and domain-specific studies extend FL to edge networks and healthcare, where privacy constraints are especially strict [9]. Within our framework, federated

learning is used specifically to allow the RL scheduler and anomaly-detection components to be improved using signals from multiple cloud tenants or regions without those tenants exposing raw operational data to one another.

2.3 Zero-Trust Security Architectures

The U.S. National Institute of Standards and Technology (NIST) formalized zero-trust architecture (ZTA) in Special Publication 800-207, which shifts security enforcement from static network perimeters toward continuous, per-request verification of identity, device posture, and context for every resource access [10]. The publication defines seven tenets, including treating all data sources and computing services as resources and granting access on a per-session, least-privilege basis [10]. Applied work extends these tenets to microservice and cloud-native environments, replacing implicit trust in network location with policy-engine-mediated, least-privilege access decisions [11]. Our framework adopts NIST SP 800-207's policy-engine/policy-enforcement-point separation as the security control plane through which every LLM-agent-initiated data operation must pass before execution.

2.4 Reinforcement Learning for Cloud Resource Management

Deep reinforcement learning (DRL) has been widely studied as a mechanism for adaptive job scheduling and resource allocation in cloud computing, replacing static heuristics with policies learned from continuous interaction with the environment. Surveys of DRL-based scheduling summarize algorithmic approaches (value-based, policy-gradient, and actor-critic methods) and their reported gains in latency, cost, and energy efficiency relative to heuristic baselines [12], [13]. Extensions to multi-access edge computing address the additional heterogeneity introduced by geographically distributed, resource-constrained nodes [14]. In our framework, the RL scheduler consumes structured intents from the LLM planner agent and produces placement/migration actions, rather than acting as an independent, opaque controller.

2.5 Multi-Cloud Data Management and Interoperability

Industry and practitioner literature converge on interoperability, inconsistent security postures, vendor lock-in, and data gravity as the dominant open problems in multi-cloud data management [15], [16]. Analyst surveys report that roughly nine

in ten enterprises now operate a multi-cloud strategy, yet a large share report siloed applications and unmet interoperability expectations [16], [17]. These findings motivate an orchestration layer capable of reasoning across providers' heterogeneous APIs, data formats, and security models, which is the gap this paper's proposed LLM-agent layer is intended to address.

2.6 Comparative Summary

Table 1 summarizes how each constituent research thread contributes to, and falls short of, the requirements of autonomous multi-cloud data management, motivating the integrated framework proposed in Section 4.

Research Thread	Key Strength	Gap Relative to Full Autonomy
LLM-based agents [1]-[5], [19]	Flexible, language-driven task decomposition and self-correction	No native access-control or resource-optimization guarantees
Federated learning [6]-[9], [20], [21]	Privacy-preserving cross-silo model improvement	Does not address task planning or real-time placement decisions
Zero-trust security [10], [11]	Continuous, identity-centric access enforcement	Assumes requests are already well-formed; does not plan tasks
RL cloud scheduling [12]-[14]	Learned, adaptive placement/scheduling policies	Requires structured state/action space; not language-driven

3. System Model and Problem Formulation

To make the architectural discussion in Section 4 precise, we introduce a lightweight notation for the entities and decisions the framework must reason about. This notation is intended to scope

the design space, not to claim a solved closed-form optimization; solving the resulting formulation is left to the RL scheduler described in Section 4.5 and to future empirical work.

Symbol	Meaning
$D = \{d1, \dots, dn\}$	Set of managed datasets or data objects
$P = \{p1, \dots, pm\}$	Set of heterogeneous cloud/edge providers or regions
I	A natural-language or policy-specified data-management intent
$A(I)$	Ordered set of candidate actions produced by the Planner Agent for intent I
$C(a)$	Estimated monetary cost of executing action a
$L(a)$	Estimated latency impact of executing action a
$R(a)$	Estimated compliance/data-residency risk of executing action a
π_{θ}	RL scheduler policy, parameterized by θ , mapping state to placement/migration action
$G(a)$	Zero-trust policy engine's binary authorization decision for action a

For a given intent I , the Planner Agent produces a candidate action sequence $A(I)$. The Negotiation Agent scores each candidate action a in $A(I)$ against a weighted objective combining cost, latency, and compliance risk, $J(a) = w1 \cdot C(a) +$

$w2 \cdot L(a) + w3 \cdot R(a)$, where the weights $w1$, $w2$, $w3$ are organization-specified preferences over cost, performance, and compliance. The Critic Agent filters $A(I)$ to remove actions that appear to violate stated policy before any action is forwarded to the

Zero-Trust Policy Engine, which computes the binary authorization $G(a)$ for each remaining action independently of the agent layer. Only actions with $G(a) = 1$ are forwarded to the RL scheduler, whose policy π_{θ} selects the concrete provider, region, and replication factor for execution. This separation ensures that no single LLM-agent output can, by itself, cause a privileged action to execute: authorization is always computed by a component that is not itself an LLM.

4. Proposed Framework: Architecture and Agent Roles

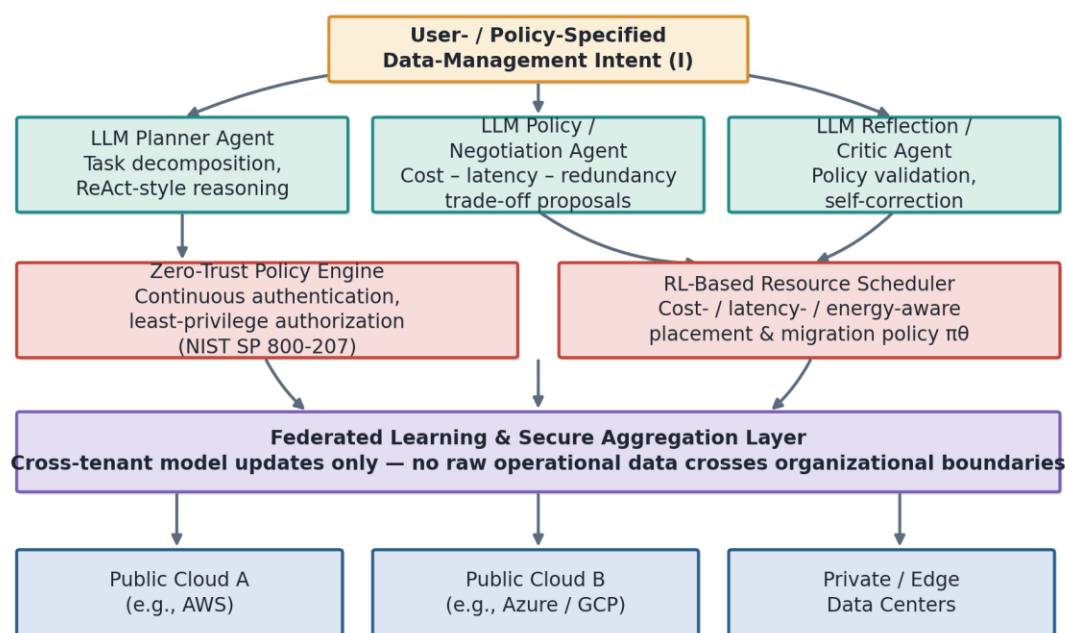
4.1 Architectural Overview

Figure 1 summarizes the proposed layered architecture. A user- or policy-specified data-management intent I enters at the top of the stack

and is decomposed by an LLM Planner Agent into the ordered sub-task sequence $A(I)$ (e.g., locate source data, verify compliance constraints, select target regions, initiate transfer, verify integrity). A companion LLM Policy/Negotiation Agent reasons over service-level objectives (cost, latency, redundancy) and proposes candidate placements, while an LLM Reflection/Critic Agent reviews the planner's and negotiator's proposed actions against organizational policy before they are allowed to reach the enforcement layer. This planner-critic separation mirrors the reflection and self-correction patterns documented in the LLM-agent literature, in which a secondary reasoning pass measurably reduces invalid or unsafe action proposals before execution [1], [3], [19].

Figure 1. High-Level Architecture of the Proposed

LLM-Agent-Driven Big Data Management Framework



Arrows indicate the direction of control / decision flow, not raw data flow.

Figure 1. High-level architecture of the proposed LLM-agent-driven big data management framework.
Arrows indicate the direction of control/decision flow, not raw data flow.

Beneath the agent layer sits a control plane composed of two coordinated components. The Zero-Trust Policy Engine implements the policy-decision/policy-enforcement separation described in NIST SP 800-207 [10]: every data-plane action proposed by an agent is treated as an access request that must be continuously authenticated and authorized based on identity, device posture, and contextual risk signals, rather than being implicitly trusted because it originated from an internal orchestrator. The RL-based Resource Scheduler receives validated action proposals and determines concrete placement, replication, and migration decisions using a learned policy trained to balance latency, monetary cost, and energy consumption, consistent with DRL-based cloud-scheduling approaches surveyed in the literature [12], [13]. Below the control plane, a Federated Learning and Secure Aggregation layer allows the RL scheduler's policy and the system's anomaly-detection models to be improved using operational telemetry from

multiple cloud regions or tenant organizations without centralizing raw logs, following the horizontal/vertical federated learning patterns formalized by Yang et al. and the FedAvg algorithm of McMahan et al. [6], [20]. Only model updates (e.g., gradients or parameter deltas), rather than raw operational data, cross organizational boundaries, and these updates are subject to secure aggregation before being merged into the global policy, consistent with the robustness recommendations surveyed by Kairouz et al. [21]. Finally, the heterogeneous resource layer represents the actual multi-cloud and edge infrastructure being managed: multiple public cloud providers, on-premise data centers, and edge sites, each with distinct APIs, pricing models, and compliance boundaries, consistent with the interoperability challenges documented in practitioner surveys of multi-cloud environments [15], [16].

4.2 Agent Roles in Detail

Agent / Component	Primary Responsibility	Grounding in Prior Work
Planner Agent	Decomposes natural-language data management intents into ordered, verifiable sub-tasks.	Profile/planning modules in LLM-agent surveys [1]; ReAct-style interleaving [19]
Policy/Negotiation Agent	Proposes placement and SLA trade-offs (cost vs. latency vs. redundancy).	Multi-agent negotiation patterns [2], [3]
Reflection/Critic Agent	Validates proposed actions against organizational policy before execution.	Self-correction / reflection patterns [1], [3], [19]
Zero-Trust Policy Engine	Continuous, per-request authentication and least-privilege authorization.	NIST SP 800-207 [10]; ZT microservice patterns [11]
RL Resource Scheduler	Learns placement/migration policies optimizing cost, latency, energy.	DRL cloud-scheduling surveys [12], [13], [14]
Federated Aggregation Layer	Improves shared models from multi-tenant telemetry without centralizing raw data.	Federated learning [6], [20], [21]

4.3 Zero-Trust Integration

Consistent with NIST SP 800-207's seven tenets [10], the policy engine in our framework treats every dataset, pipeline, and API endpoint as a resource requiring per-request authorization; grants access on a per-session, least-privilege basis; and continuously re-evaluates trust using identity, device posture, and behavioral signals rather than network location. Critically, the policy engine is architected as a component entirely separate from

the LLM agents: it receives structured action requests and returns a binary authorization decision plus an audit record, and it has no dependency on the correctness of the LLM agents' internal reasoning.

4.4 Interaction Protocol

Figure 9 (Section 6.5) depicts the illustrative message sequence for a single data-management intent as it flows from the user through the

planner, negotiation, and critic agents, into the zero-trust policy engine, and finally to the RL scheduler and the target cloud resource, with an execution receipt and audit record returned along the reverse path. Every forward transition in this sequence is logged, and the zero-trust policy engine's authorization step (message 5-6 in Figure 9) is the only point at which an action becomes eligible for execution.

4.5 RL Scheduler Formulation

The RL scheduler is formulated as a Markov decision process in which the state st encodes current provider load, network conditions, and pending validated action requests; the action a selects a target provider/region and replication factor for a given data-management sub-task; and the reward rt is a weighted combination of negative cost, negative latency, and a penalty term for any subsequently detected policy violation. The policy $\pi_\theta(a_t | s_t)$ is trained using an actor-critic method, consistent with the DRL scheduling approaches surveyed in [12] and [13], and it is updated using federated aggregation of gradients contributed by multiple regional schedulers, as described in Section 4.1.

5. Algorithmic Description of the Control Loop

Algorithm 1 summarizes the planner-negotiator-critic-enforcement control loop described in Sections 3 and 4 in pseudocode form. This description is intended to make the proposed interaction protocol reproducible for future implementation work; it is not accompanied by a runtime implementation in this paper.

Algorithm 1. Intent-to-Execution Control Loop

Input: intent I , policy weights (w_1 , w_2 , w_3), organizational policy set P_i

Output: execution receipt or rejection record

```

1:  $A(I) \leftarrow \text{PlannerAgent.decompose}(I)$ 
2: for each candidate action  $a$  in  $A(I)$ :
3:  $\text{score}(a) \leftarrow w_1 \cdot C(a) + w_2 \cdot L(a) + w_3 \cdot R(a)$  //
  NegotiationAgent
4:  $A_{\text{ranked}} \leftarrow \text{sort } A(I) \text{ by } \text{score}(a) \text{ ascending}$ 
5:  $A_{\text{filtered}} \leftarrow \text{CriticAgent.validate}(A_{\text{ranked}}, P_i)$ 
// remove policy-inconsistent actions

```

```

6: for each action  $a$  in  $A_{\text{filtered}}$ :
7:  $G(a) \leftarrow \text{ZeroTrustPolicyEngine.authorize}(a)$  //
  independent, non-LLM decision
8: if  $G(a) == 1$ :
9:  $\text{placement} \leftarrow \text{RLScheduler.select}(a, \text{state})$  //
   $\pi_\theta(a_t | s_t)$ 
10:  $\text{receipt} \leftarrow \text{CloudResource.execute}(\text{placement})$ 
11:  $\text{AuditLog.record}(a, G(a), \text{receipt})$ 
12: else:
13:  $\text{AuditLog.record}(a, G(a), \text{rejected}=\text{true})$ 
14: return  $\text{AuditLog.summary}()$ 

```

Two properties of Algorithm 1 are central to the framework's security posture. First, the authorization step (line 7) is computed by the Zero-Trust Policy Engine, a component with no LLM in its decision path, so a manipulated or hallucinating agent cannot itself authorize a privileged action. Second, every branch of the loop is recorded in an append-only audit log (lines 11 and 13), which is a prerequisite for the red-teaming and post-hoc forensic analysis identified as future work in Section 9.

6. Illustrative Scenarios and Simulated Analysis

Because this paper presents an architectural proposal rather than a deployed system, we use synthetic, illustrative scenarios to reason about expected behavior and to define the metrics that a subsequent empirical study should measure. All figures in this section are explicitly simulated/illustrative and are intended to frame testable hypotheses, not to report results from a production deployment or controlled benchmark.

6.1 Illustrative Agent Task Distribution

Figure 2 illustrates a plausible distribution of agent task invocations in a hypothetical multi-cloud workload combining data placement/migration, access-control decisioning, schema reconciliation, anomaly detection, and SLA negotiation. This distribution is constructed for illustration and should be replaced with measured task-mix data once the framework is instrumented in a real deployment.

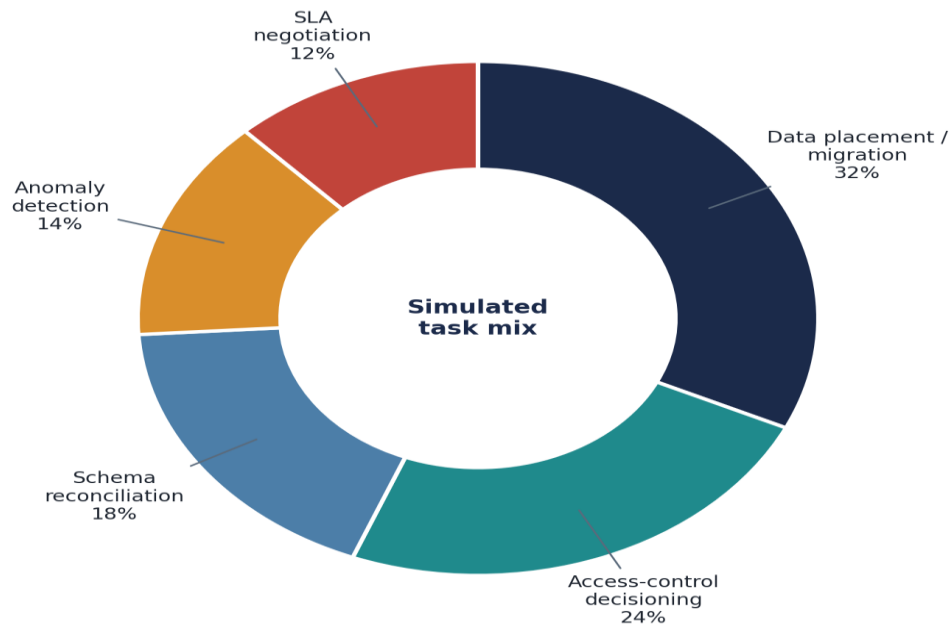
Figure 2. Illustrative Distribution of Agent Task Invocations*Simulated multi-cloud workload composition (synthetic data)*

Figure 2. Illustrative distribution of agent task invocations in a simulated multi-cloud workload (synthetic data).

6.2 Illustrative Comparison Against Baseline Orchestration Strategies

Figure 3 presents a simulated comparison of average task latency and policy-violation rate across four hypothetical orchestration strategies: static rule-based orchestration, greedy heuristics, a pure reinforcement-learning scheduler with no LLM planning layer, and the proposed LLM-agent framework. The values are illustrative projections

informed qualitatively by trends reported in DRL-scheduling surveys (which report consistent latency and efficiency gains for learned policies over static heuristics [12], [13]) and are not the output of a controlled benchmark. A rigorous empirical comparison using standardized workloads (e.g., publicly available cluster or cloud-provider traces) is identified as required future work in Section 9.

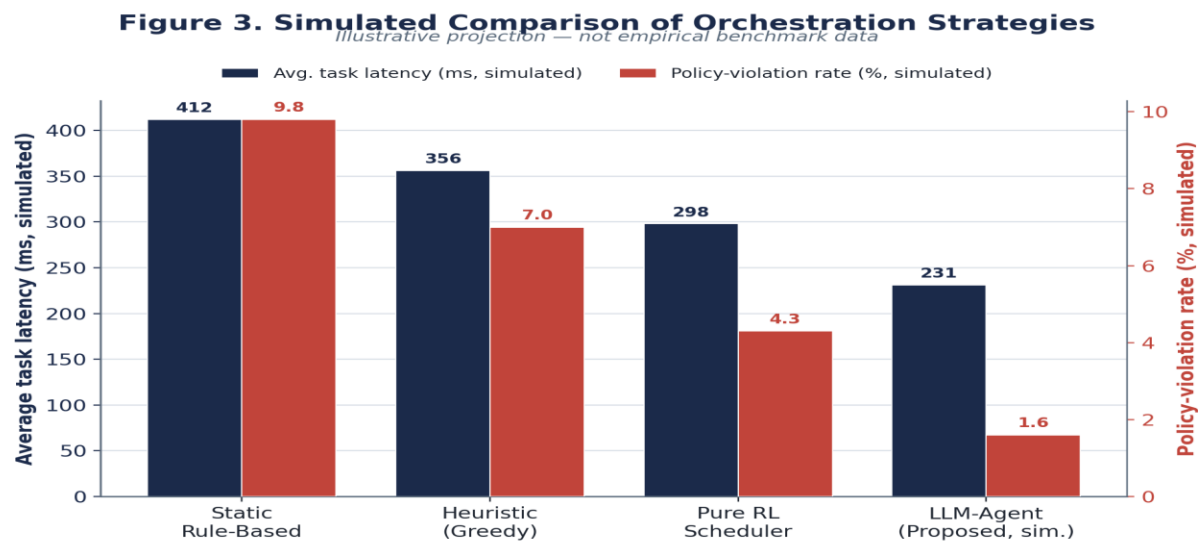


Figure 3. Simulated comparison of average task latency and policy-violation rate across orchestration strategies (illustrative, not empirical benchmark data).

6.3 Illustrative Multi-Criteria Comparison

Figure 6 extends the two-metric comparison of Figure 3 into a six-criteria radar chart spanning latency, cost efficiency, security posture, adaptability, explainability, and scalability. As with Figure 3, the scores are simulated (0-9 scale) and are intended to visualize the qualitative trade-off pattern suggested by the literature ~ static rule-based systems score well on explainability but

poorly on adaptability, pure RL schedulers improve adaptability and scalability at some cost to explainability, and the proposed layered framework is hypothesized to improve on both dimensions simultaneously because planning (explainable, language-based) and execution (learned, adaptive) are separated ~ rather than to report a measured evaluation.

Figure 6. Illustrative Multi-Criteria Comparison
Simulated scoring, 0-9 scale, across three orchestration strategies

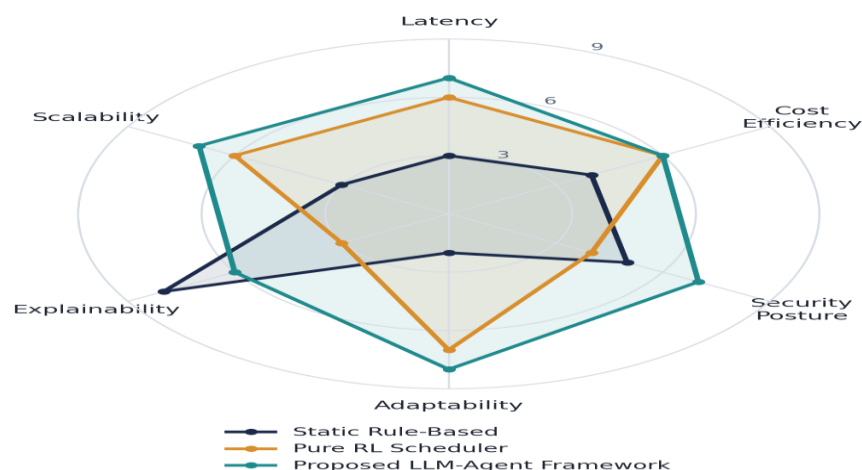


Figure 6. Illustrative multi-criteria comparison across three orchestration strategies (simulated scoring, 0-9 scale).

6.4 Illustrative Scalability Trend

Figure 4 sketches an illustrative scalability trend, plotting relative orchestration overhead against the number of managed cloud nodes on a log-log scale for a centralized orchestrator versus the proposed layered agent framework. The intent is to visualize the hypothesis that decomposing

planning (LLM agents), enforcement (zero-trust engine), and placement (RL scheduler) into separately scalable components should grow overhead more slowly than a monolithic centralized orchestrator as cluster size increases. This remains a hypothesis to be tested, not a validated result.

Figure 4. Illustrative Scalability Trend
Simulated relative overhead vs. managed node count (log-log axes)

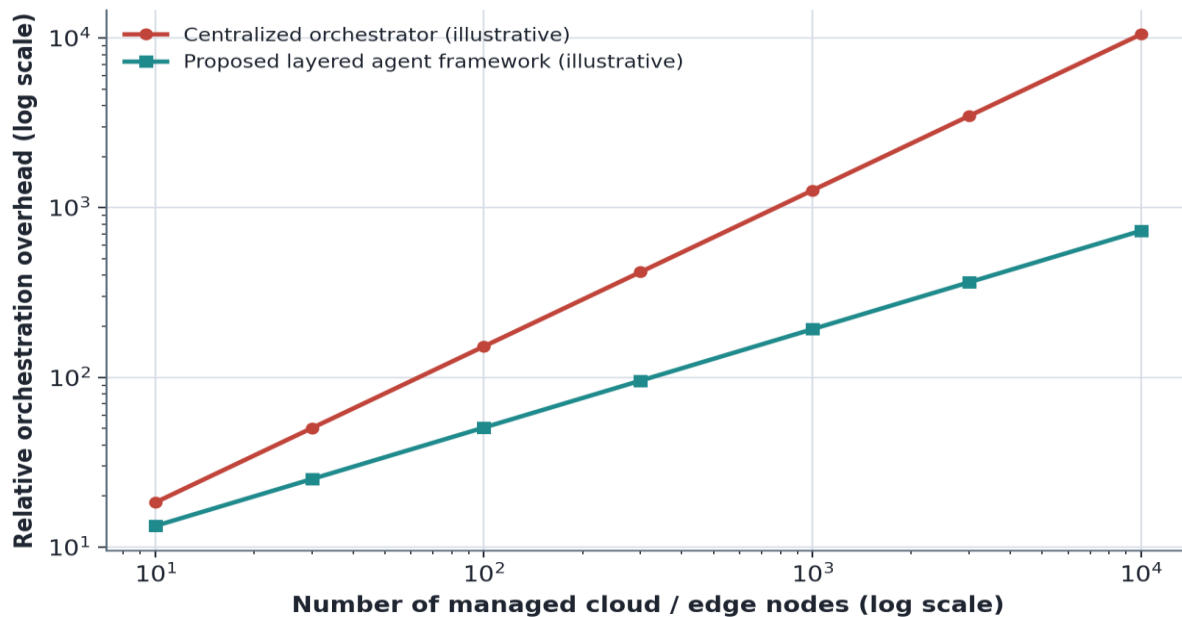


Figure 4. Illustrative scalability trend as the number of managed cloud nodes grows (simulated, log-log axes).

6.5 Illustrative Interaction Sequence

Figure 9 depicts the illustrative message sequence introduced in Section 4.4, tracing a single data-management intent through decomposition,

negotiation, critique, authorization, scheduling, and execution, with telemetry and an audit record returned along the reverse path.

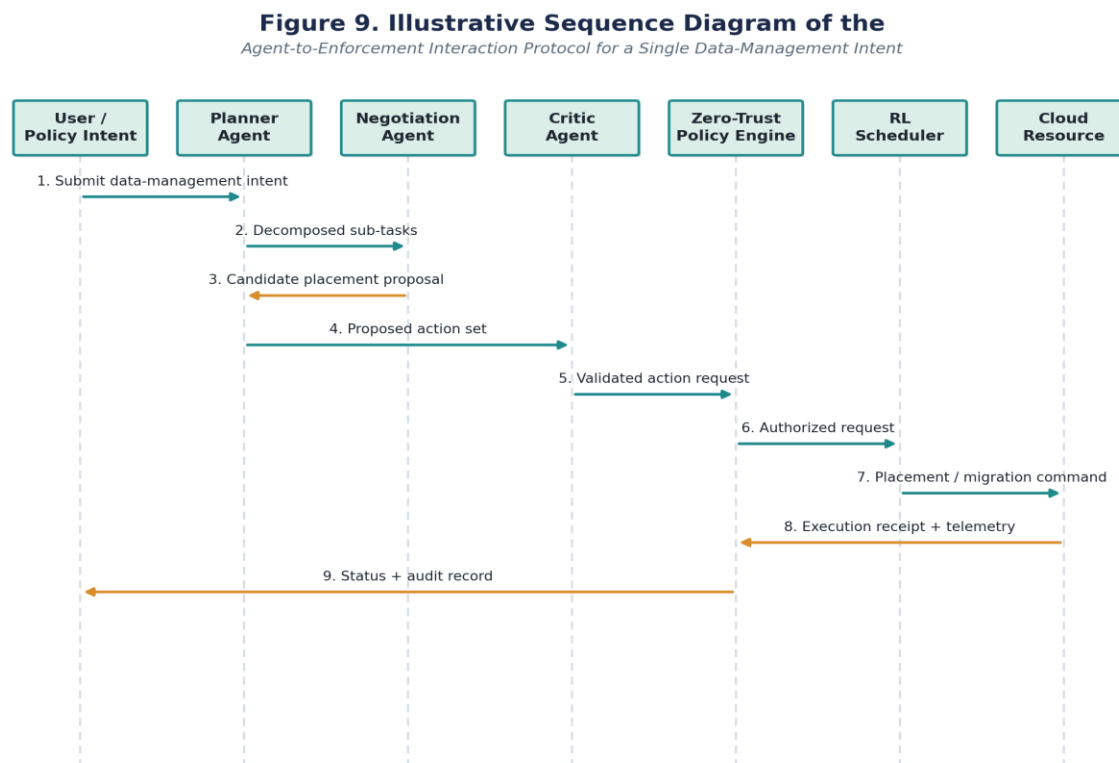


Figure 9. Illustrative sequence diagram of the agent-to-enforcement interaction protocol for a single data-management intent.

6.6 Illustrative Cost/Redundancy Trade-off Surface

Figure 5 shows a synthetic three-dimensional cost surface over two design variables the scheduler must jointly reason about: the number of managed nodes and the data-redundancy (replication) factor. The U-shaped cost response to redundancy

reflects the well-known trade-off in distributed storage between under-replication risk (data-loss exposure) and over-replication cost (storage and egress overhead); the specific surface is illustrative and parameterized for visualization rather than fitted to measured data.

Figure 5. Illustrative Synthetic Cost Surface
Simulated trade-off over cluster size and replication factor (not measured data)

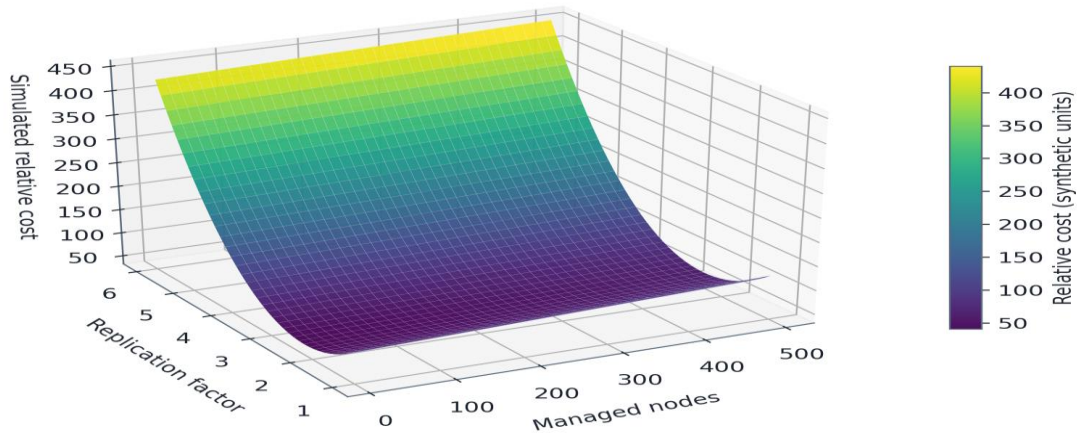


Figure 5. Illustrative synthetic cost surface over cluster size and replication factor (simulated trade-off space, not measured data).

6.7 Illustrative Compliance-Risk Matrix

Figure 7 presents a synthetic heatmap of data-residency and compliance risk across five hypothetical providers and five regions, illustrating the kind of decision surface the Negotiation and Critic agents must reason over

when a dataset is subject to cross-border transfer restrictions. High-risk cells (e.g., transferring regulated data toward regions without an adequacy determination) are intended to be down-weighted or rejected by the Critic Agent before reaching the zero-trust policy engine.

Figure 7. Illustrative Compliance-Risk Matrix
Synthetic data-residency risk across providers and regions

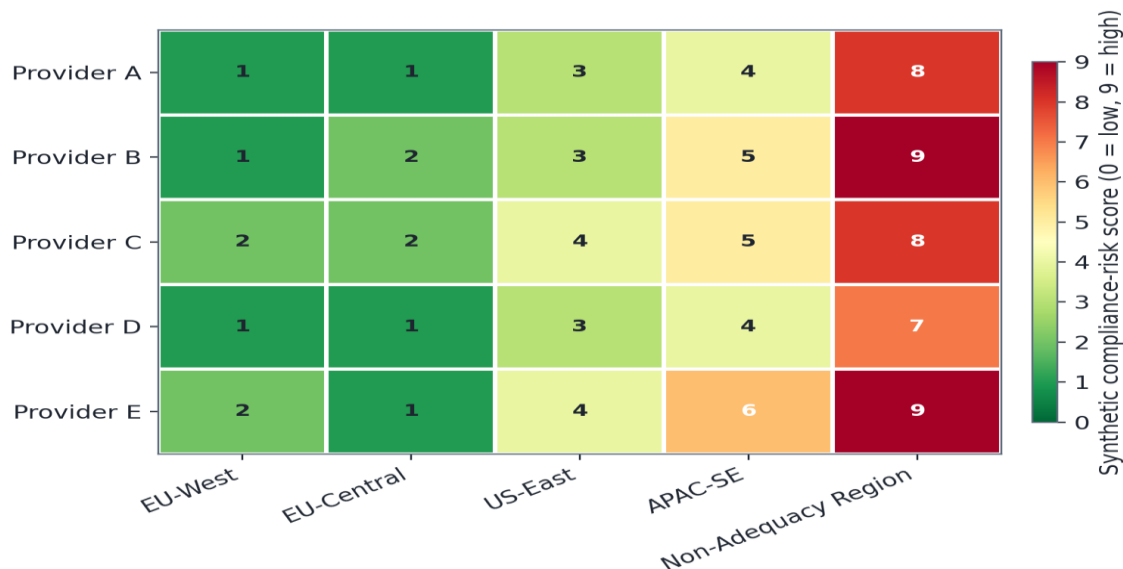


Figure 7. Illustrative data-residency and compliance-risk matrix across providers and regions (synthetic scenario).

6.8 Illustrative Workload Growth Projection

Figure 8 sketches an illustrative, synthetic projection of managed data-volume growth by data type (structured, semi-structured, unstructured) over a six-year horizon, motivating the scalability

requirement discussed in Section 6.4. These figures are projections for illustration, not measured or forecast values for any real organization.

Figure 8. Illustrative Projected Growth in Managed Data Volume
Synthetic forecast scenario by data type over a six-year horizon

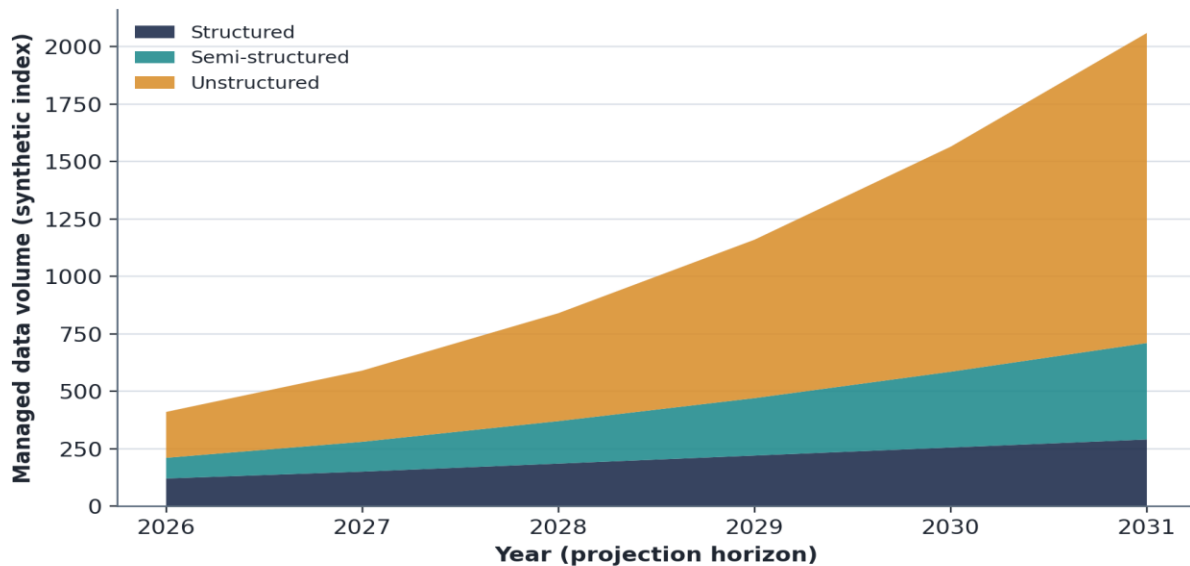


Figure 8. Illustrative projected growth in managed data volume by type (synthetic forecast scenario).

7. Security, Privacy, and Threat Considerations

Introducing LLM agents into the data-management control path introduces threat surfaces beyond those considered in classical zero-trust or federated-learning deployments, and recent security-focused surveys of LLM-based agents catalogue risks including prompt injection,

tool-use manipulation, and unsafe autonomous action execution [18]. Table 2 summarizes a working taxonomy of agent-specific threats relevant to the proposed framework, together with the mitigation already built into the architecture and the residual risk that would require empirical red-teaming to characterize.

Threat Class	Architectural Mitigation	Residual Risk Requiring Empirical Study
Prompt injection via untrusted data sources	Critic agent validation; zero-trust enforcement independent of agent output	Adversarial inputs crafted to evade critic-agent detection
Tool-use / API manipulation	Structured action schema; enforcement layer rejects malformed or out-of-scope actions	Novel tool-call patterns not covered by current validation rules
Federated model poisoning	Secure aggregation; anomaly-aware update filtering [7], [8], [21]	Coordinated multi-participant poisoning attacks
Over-trusted critic agent	Enforcement layer treated as authoritative, not the critic agent	Critic-agent blind spots that also evade enforcement-layer rules

Threat Class	Architectural Mitigation	Residual Risk Requiring Empirical Study
Credential or identity compromise	Continuous, per-session zero-trust verification [10]	Compromise of the identity provider itself

We argue that three design choices are necessary, though not sufficient, to mitigate these risks within the proposed framework: (i) no LLM agent output should be permitted to directly execute a privileged data-plane action; all actions must pass through the zero-trust policy engine as an independent, non-LLM-based enforcement point, consistent with the policy-decision/enforcement separation in NIST SP 800-207 [10]; (ii) the critic agent's validation step should be treated as a defense-in-depth measure rather than a substitute for enforcement-layer checks, since LLM-based validators are themselves subject to manipulation; and (iii) federated aggregation must incorporate secure aggregation and anomaly-aware filtering of submitted updates, since federated settings are known to be susceptible to poisoning by compromised or malicious participants [7], [8], [21].

Because these are architectural safeguards rather than empirically validated defenses, Section 9 identifies red-teaming and adversarial evaluation of the agent-to-enforcement boundary as a required next step before any production deployment.

8. Data Governance and Regulatory Compliance Considerations

Autonomous, cross-border data placement raises governance questions independent of the technical security architecture. Regulations such as the European Union's General Data Protection Regulation (GDPR, Regulation (EU) 2016/679) impose data-residency, purpose-limitation, and cross-border-transfer conditions that any autonomous placement decision must respect [22]. In the proposed framework, these constraints are intended to be encoded as organizational policy P_i (Section 3) that the Critic Agent checks before an action reaches the zero-trust policy engine, and as constraints on the Negotiation Agent's scoring function (via the compliance-risk term $R(a)$) so that non-compliant placements are

penalized before they are proposed, rather than only rejected after the fact.

We emphasize two governance-related limitations of the current proposal. First, encoding regulatory requirements as machine-checkable policy is itself a substantial undertaking that varies by jurisdiction, data category, and organizational risk tolerance; the framework assumes such policy has already been formalized by legal and compliance stakeholders, and does not itself perform legal interpretation. Second, autonomous systems that make cross-border placement decisions may still require human-in-the-loop review for high-risk categories of data (e.g., health or biometric data) even where the technical architecture supports full automation; the appropriate degree of autonomy for such categories is an organizational and legal judgment outside the scope of this paper's architectural contribution.

9. Discussion, Limitations, and Future Work

This paper's contribution is architectural: it proposes a way of composing existing, separately validated building blocks (LLM agents, zero-trust enforcement, RL scheduling, and federated learning) into a coordinated framework for multi-cloud big-data management, and it uses illustrative figures to frame the hypotheses and metrics that should guide empirical validation. It does not report results from a working implementation, a production deployment, or a controlled benchmark, and the numeric values in Section 6 are explicitly synthetic.

We identify the following as necessary future work before the framework's claims can be considered validated:

1. Reference implementation and benchmarking against standardized cloud workload traces (e.g., publicly available cluster or cloud-provider datasets) rather than synthetic figures;
2. Formal threat modeling and red-teaming of the agent-to-policy-enforcement boundary,

including prompt-injection and tool-manipulation attack classes [18];

3. Empirical study of federated-aggregation robustness under adversarial or unreliable participants in a multi-tenant cloud setting [7], [8], [21];

4. Formalization of representative regulatory policy sets (e.g., GDPR data-residency rules) as machine-checkable constraints, and evaluation of the Critic Agent's and policy engine's ability to enforce them correctly;

5. User studies evaluating whether natural-language-specified data-management intents are interpreted correctly and safely by the planner/negotiator/critic pipeline across ambiguous or underspecified requests.

We also note a limitation of scope: the framework as presented does not address the energy and carbon-accounting implications of cross-region data replication, nor does it address multi-organization incentive design (e.g., how competing cloud tenants might be incentivized to contribute honest telemetry to the federated layer). Both are identified as open directions rather than resolved within this paper.

10. Conclusion

We have presented a layered, conceptual framework in which LLM-based agents coordinate data-management decisions across heterogeneous cloud environments while remaining subordinate to a zero-trust enforcement layer, an RL-based resource scheduler, and a federated-learning privacy layer. By grounding each architectural component in an existing, separately validated body of research, and by formalizing the system model, interaction protocol, and control-loop algorithm, the proposal aims to identify a coherent integration path rather than an unsubstantiated technology combination. The illustrative figures presented here are intended to seed, not substitute for, the empirical benchmarking, security evaluation, regulatory-compliance formalization, and user studies outlined as future work in Section 9.

Acknowledgments

The authors declare that this work received no dedicated external funding. Any institutional, computing, or infrastructure support should be acknowledged here by the authors prior to submission.

REFERENCES

- [1] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin, W. X. Zhao, Z. Wei, and J.-R. Wen, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.
- [2] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang, S. Zhang, X. Deng, A. Zeng, Z. Du, C. Zhang, S. Shen, T. Zhang, Y. Su, H. Sun, M. Huang, Y. Dong, and J. Tang, "AgentBench: Evaluating LLMs as agents," *arXiv preprint arXiv:2308.03688*, 2023.
- [3] "A survey on LLM-based multi-agent systems: Recent advances and new frontiers in application," *arXiv preprint arXiv:2412.17481*, 2024.
- [4] X. Liu et al., "AgentBench: Evaluating LLMs as agents," *arXiv preprint arXiv:2308.03688*, 2023.
- [5] "Evaluation and benchmarking of LLM agents: A survey," *arXiv preprint arXiv:2507.21504*, 2025.
- [6] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, article 12, 2019.
- [7] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.
- [8] V. Mothukuri, R. M. Parizi, S. Pouriyeh, Y. Huang, A. Dehghantanha, and G. Srivastava, "A survey on security and privacy of federated learning," *Future Generation Computer Systems*, vol. 115, pp. 619-640, 2021.

- [9] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 2031-2063, 2020.
- [10] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero Trust Architecture," NIST Special Publication 800-207, National Institute of Standards and Technology, Aug. 2020.
- [11] "Zero trust security model implementation in microservices architectures using identity federation," arXiv preprint arXiv:2511.04925, 2025.
- [12] "Deep reinforcement learning for job scheduling and resource management in cloud computing: An algorithm-level review," arXiv preprint arXiv:2501.01007, 2025.
- [13] "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions," arXiv preprint arXiv:2105.04086, 2021.
- [14] "A survey on resource scheduling approaches in multi-access edge computing environment: A deep reinforcement learning study," *Cluster Computing*, 2025.
- [15] "Common interoperability challenges for multi-cloud," *Console Connect Technical Blog*, 2024.
- [16] "Multicloud data management benefits and challenges," *Nutanix Technical Resources*, 2024. (Flexera 2022 State of the Cloud Report cited therein, reporting approximately 89% multi-cloud adoption.)
- [17] "Multi-cloud data management: Where it fails and how to fix it," *N-iX Technical Blog*, 2026. (Citing Gartner projection on multi-cloud interoperability shortfalls by 2029.)
- [18] "A comprehensive survey in LLM(-agent) full stack safety: Data, training and deployment," arXiv preprint arXiv:2504.15585, 2025.
- [19] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2023.
- [20] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artificial Intelligence and Statistics (AISTATS)*, PMLR, 2017, pp. 1273-1282.
- [21] P. Kairouz, H. B. McMahan, B. Avent, A. Bellet, M. Bennis, A. N. Bhagoji, K. Bonawitz, Z. Charles, G. Cormode, R. Cummings et al., "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1-2, pp. 1-210, 2021.
- [22] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 (General Data Protection Regulation), *Official Journal of the European Union*, L 119, 2016.