Scholar Publishing Group
International Journal of Multimedia Computing
https://doi.org/10.38007/IJMC.2026.070202
ISSN 2789-7168 Vol. 7, Issue 2:10-16

Governance Strategies and Fault Propagation Suppression Methods for Digital Infrastructure Dependent on Topology

Yiru Zhang

Consumer Electronics Technology, Amazon, New York, NY 10044, USA

Keywords: Digital infrastructure; Topology orchestration; Fault propagation; Multi-cloud edge; Resilient governance

Abstract：For digital infrastructures closely linked to multi-cloud environments, edge nodes and service meshes, faults may spread across regions, control planes, service calls, data replication, and third-party dependencies. The governance framework based on topology orchestration in this paper consists of dynamic graph modelling, node propagation pressure, edge-level propagation probability, orchestration optimisation and cascading suppression evaluation. Combine the number of cloud service provider regions and statistics on major internet outages to propose a strategy of "topology explicitness—fault domain partitioning—change gating—traffic traction—resilience verification". Therefore, the focus of governance for digital infrastructure should shift from single-point repair to the anticipation of spread conditions. Unify the dependency graph, implement cross-domain scheduling and continuous drills, and thus reduce the cascading effect; improve the continuity and auditability of critical services.

1. Introduction

Digital infrastructure has developed from a single data centre into a large-scale system that includes public and private clouds, edge nodes, content delivery networks, service meshes, automated operation and maintenance platforms, etc. Computing resources, network paths, identity authentication, database replication, message queues and external interfaces are required by the company's core business. Any local degradation may be spread to cause a large-scale service outage via a call chain or shared control plane. Research on the robustness of complex networks has shown that local disturbances can spread along connection edges, load edges and functional dependency edges to cause a system-wide failure due to the overload of critical nodes [1]. Therefore, the reliability of the digital infrastructure is no longer just about hardware redundancy, but also determined by the coupling relationship of the dependent topology, shared failure domain and control strategy.

Network digital twins and topology observability offer new tools for propagation governance. Relevant studies have proposed that real-time digital twins can create continuously updated virtual environments to verify strategies, reproduce incidents and compare recovery paths [2], and multi-

domain network digital twins further need to show the relationship among links, services, control plane and data plane simultaneously [3]. Therefore, this paper defines topology orchestration as the consistent scheduling of service placement, traffic paths, fault isolation, state replication, change windows and recovery actions on a computable dependency graph. Thus, the definition of 'orchestration' here refers to actively constructing the paths and propagation probabilities for propagation, as well as the blast radius.

From an engineering practice perspective, many such cases are not due to a total failure of the underlying hardware, but rather to configuration drift, defective grey-scale deployment strategies, excessive retries, DNS caching anomalies, cross-region replication blockage, or unreachable access services. If the governance system only shows a single alarm or the status of one node, it is impossible to know whether the fault will be carried through the buffer, impact the core business, or cause cascading recovery failures. This paper studies "how dependency topologies are observed, organised and suppressed" to develop a governance method that combines theoretical modelling with engineering realisation.

2. Current Status Analysis of the Research Topic

Multi-cloud edge deployment is still developing, and the isolation levels for regions, availability zones, and control planes across different cloud providers are inconsistent. An increase in the number of regions will expand access to nearby areas and data residency and disaster recovery options; however, it may also raise the costs of cross-cloud interconnection, policy synchronisation and consistency maintenance. Research on multi-cloud microservices re-orchestration has added cost, latency and business continuity to migration goals, and now orchestrators are seen as shapers of fault propagation boundaries rather than simple resource schedulers [4]. In terms of topology, where the service instances are located, whether they share gateways and identity services, and whether there is a common source database master will all change the propagation path and recovery order.

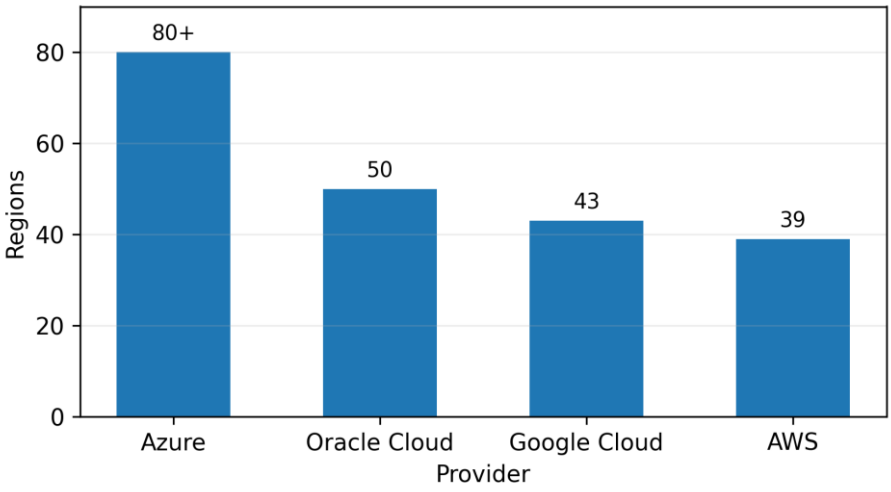

Figure 1. Statistics on the Number of Global Regions for Top Cloud Service Providers

Figure 1 Explanation: The figures are organised by the regional scope of the publicly available infrastructure pages, and Azure is listed as "80+" at the lower bound. As shown in the above experiments, the main cloud service providers have established large-scale global regional networks. The regions help improve the proximity access and data-residency capabilities; however, they do

not directly address fault isolation. Topology governance still needs to clarify the actual independence of availability zones, control planes, leased line interconnects, unified identity services and shared storage, and avoid assuming that superficial geographical dispersion equates to engineering-based propagation isolation.

Table 1. Dependency Layers and Governance Gaps in Digital Infrastructure

Layer	Dependency	Trigger	Governance gap	Control action
Network	BGP, DNS, CDN	Route leak	Weak path visibility	Diverse rerouting
Cloud platform	Region, zone, IAM	Control-plane fault	Shared blast radius	Domain partitioning
Service mesh	Gateway, sidecar, API	Retry storm	Policy risk blind spot	Circuit breaking
Data state	Database, cache, queue	Replica lag	RPO/RTO mismatch	State isolation
Observability	Metrics, logs, traces	Alert storm	Missing topology context	Topology-aware RCA

Table 1 shows that the propagation dependency has been reduced to five layers: network, cloud platform, service mesh, data state and observability. Each layer has different triggers and governance loopholes, so fault propagation is not the fault of a single device but rather the result of combined effects from resources, links, state and organisational responsibility. In a highly automated environment, the propagation speed of errors may be faster than that of physical failures; thus, the topology context must be incorporated into alarms, modifications and recovery mechanisms.

Fault injection and root-cause localisation techniques are moving away from post-event alarm convergence towards topology-level verification. Database-aware microservice resilience testing extends fault injection from coarse host failures to service and database interfaces and visualises the location and impact of injected disruptions [5], while research on root cause localisation in cloud-edge collaboration has shown that call dependencies and resource contention can jointly amplify faults [6]. The common implication of the above studies is that without a dependency topology, fault drills are likely to remain on isolated nodes, and without orchestration control, root cause localisation cannot be transformed into executable recovery operations.

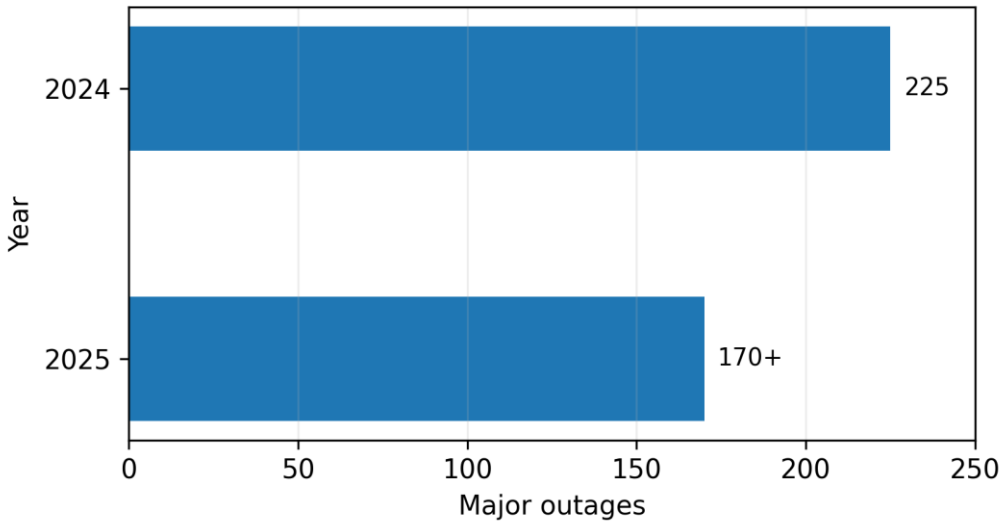

Figure 2. Horizontal Statistics of the Number of Major Internet Outages Worldwide

Figure 2 Explanation: A horizontal bar chart is used to show the statistical data of two years, and this way prevents misinterpretation as a continuous-time trend of the annual data. According to the 2024 and 2025 data from Cloudflare Radar, there were about 225 and more than 170 major internet outages, respectively. A reduction in the number of outages does not indicate that the risk has been eliminated. Configuration errors, energy supply problems, damaged submarine cables, third-party platform incidents and changes in regional policies can still cause cross-domain propagation, so topology governance needs to be long-term and institutionalised.

3. Problem Statement: Topological Representation of Fault Propagation

Current governance practice has three problems. First, due to the fragmented nature of CMDB, cloud assets, link monitoring, call tracing and ticketing systems, topology computation is infeasible; alerts can only indicate "where the anomaly is", but not "where the anomaly will spread". Second, there is an insufficient identification of shared failure domains; different clusters may appear to be in different regions but are actually sharing DNS, IAM, CI/CD pipelines, core database masters, etc., and are managed by the same operations team. Third, governance is not aligned with topology risks; automatic scaling, GitOps, service mesh policies and certificate rotation can spread errors in highly central nodes through local changes.

Turn the risk into a number by constructing a time-varying topological graph first.

$$G_t=(V_t,E_t,W_t,A_t),\quad e_{ij}\in E_t\Leftrightarrow v_i\rightarrow v_j \tag{1}$$

Vt is the set of nodes in the formula, which can include regions, availability zones, clusters, services, databases, queues, gateways and external APIs; Et is the set of dependency edges, representing call, replication, forwarding, authentication or control relationships; W_t is the call frequency, traffic share, replication intensity and shared resource intensity; and A_t records capacity, health, business level and governance responsibility. This diagram converts the static asset list into a probabilistic propagation structure and offers a single object for risk prioritisation, change review and recovery orchestration.

The Node Propagation Pressure is given by:

$$\pi_i(t)=\sum_{j\in N_i^-} w_{ji}\,a_j(t)-\lambda_i c_i(t) \tag{2}$$

N_i^- is the set of incoming neighbours, a_j(t) is the upstream abnormal activation level, c_i(t) is the absorption capacity of node i, and λ_i is the capacity offset efficiency in the formula. If π_i(t) continues to rise, it indicates that the node has reached a propagation convergence point, and rate limiting, circuit breaking, instance expansion, read-only degradation or cross-domain migration should be triggered first. This index shows how far a propagation risk has spread; otherwise, one could only count how many times an alarm occurred.

The probability of a side leak can be expressed as:

$$p_{ij}(t)=\frac{1}{1+\exp\left[-\left(\alpha w_{ij}+\beta u_i(t)+\gamma b_j(t)-\delta s_{ij}(t)\right)\right]} \tag{3}$$

u_i(t) is the load rate at the source node, b_j(t) is the target business criticality, s_ij(t) is the isolation strength, and α, β, γ, and δ are weighting parameters in the formula. Therefore, propagation risk is not only a function of the connection strength but also affected by business-criticality, real-time load, and isolation capabilities. Increase the isolation strength, reduce excessive retries, limit high-risk call edges, and weaken shared-state dependencies to lower the propagation probability directly.

4. Problem Solving/Strategy: Suppression Methods Driven by Topology Orchestration

Topology orchestration should form a five-layer closed loop. First, unify topology assets, bind regions, clusters, services, data replicas, external APIs, certificates, policies and personnel responsibilities to the same dependency graph, and update them every minute. Second, implement fault domain partitioning by dividing the core area, buffer zone and experimental area according to centrality, business level and shared failure relationship, so that non-core traffic and experimental changes cannot directly affect the critical path. Third, implement risk-aware traffic traction, reduce the weight of nodes with high propagation pressure, circuit breaking, read-write separation, low-priority task suspension and bypass access. Fourth, set up a change explosion radius gate and perform topology impact analysis for modifications to DNS, IAM, gateway, service mesh, database replication and cross-cloud private line. Fifth, form a closed loop of topology RCA and chaos verification, drawing on Bayesian online change-point RCA and large-model-assisted root-cause reasoning [7] [8].

The objective of optimisation can be written as:

$$\min_{x,y} J=\eta_1 L(x)+\eta_2 R(x,y)+\eta_3 C(x), \quad \sum_z y_{iz} \geq 2 \tag{4}$$

The formula is as follows: x is the service placement, routing weight or replica scheduling decision; y is the fault domain selection; L, R and C are latency, propagation risk and cost, respectively; η1, η2 and η3 are weighting coefficients; and the constraint $\sum zyiz \geq 2$ requires that critical nodes are deployed in at least two isolation domains. The above objective function reflects an actual trade-off in governance: if only low latency is aimed for, then services may be concentrated in a single area; if only extreme isolation is desired, however, the cost and data-consistency will be relatively high. Therefore, Topology orchestration should find a reasonably feasible solution under the business-level constraints.

Table 2. Topology Orchestration Governance Strategy Matrix

Strategy	Trigger signal	Action	Owner	Metric
Domain zoning	High centrality	Anti-affinity placement	Cloud board	Isolation ratio
Traffic shifting	High propagation pressure	Load shedding	SRE lead	Cascade size
Change gate	Risky control change	Canary or rollback	DevOps manager	Rollback time
Topology RCA	Cross-layer alerts	Graph ranking	AIOps team	MTTR
Chaos validation	Post-change drift	Targeted injection	Reliability team	Resilience score

As shown in Table 2, strategies are converted into trigger signals, actions and responsible parties, along with indicators, for cross-departmental cooperation. Fault domain partitioning and traffic steering focus on pre-incident and in-incident suppression, change gating reduces human intervention, and topology RCA and chaos verification are used for post-incident learning. By writing strategies, actions and responsible parties in the governance matrix at the same time, coordination costs during incidents can be reduced, and the orchestration system can be both automated and maintain the necessary audit trail.

The operating modes of the governance platform in practice include "identification-isolation-guidance-recovery-replay". The identification stage calculates highly central nodes, strongly dependent edges and shared failure domains based on a dynamic graph; the isolation stage sets rate limiting, circuit breaking and anti-affinity deployments for high-risk edges; the guidance stage migrates requests to healthy domains via global traffic management; the recovery stage initiates

read-only degradation, queue smoothing and state replay according to business levels; and the replay stage writes the incident path back to the topology library and updates propagation parameters and drill scripts. Thus, this case will serve as training material for the next orchestration plan.

The extent of the inhibition can be shown as a cascade inhibition score:

$$S_c=\theta_1\left(1-\frac{|C_{after}|}{|C_{before}|}\right)+\theta_2\left(1-\frac{T_{rec}}{T_{SLO}}\right)+\theta_3 Q_{surv} \tag{5}$$

Cbefore and Cafter are the sets of nodes affected before and after the strategy is applied, Trec is the recovery time, TSLO is the service level target, Qsurv is the survival rate of critical paths, and θ1, θ2, and θ3 are the evaluation weights in the formula. This index considers the compression of the impact surface, recovery time control and critical path preservation simultaneously, and can be used for incident review, exercise scoring and A/B testing of orchestration strategies. Cloud Uptime Archive and Kubernetes cloud-edge fault-injection data provide real availability data and repeatable fault experiments to support the testing of resilience strategies [9] [10].

The governance organisation must also be adjusted. Traditionally, operations and maintenance have been divided into areas such as the network, database, application and security, and information silos are easily formed during an incident. Topology orchestration needs to build a cross-domain reliability committee, and SRE, network, cloud platform, data platform and security teams will jointly maintain dependency graphs, fault domain definitions and change access rules. Topology risk assessment should be required before releasing high-risk businesses, and in operational performance evaluation, propagation pressure, isolation ratio, rollback time and drill pass rate need to be considered.

5. Conclusion

This paper studies the governance of topology-dependent digital infrastructure and puts forward methods for dynamic graph modelling, propagation pressure calculation, propagation probability assessment, orchestration optimisation and suppression score evaluation. According to the study, the main risk of cloud and network systems is not a single point of failure, but rather the connection among topology dependencies, shared failure domains, automated changes, and recovery responses. Only by unifying the representation of regions, services, data, control planes and organisational responsibilities in a computable graph can critical propagation edges be identified before failures occur and auditable suppression actions taken upon the occurrence of a failure.

The three tasks in engineering practice should be prioritised: establish a dependency topology baseline that is updated every minute; set mandatory change gating for DNS, IAM, gateways, database master-slave replication and cross-region replication; and encapsulate rate limiting, circuit breaking, migration, rollback and read-only degradation into auditable actions. In the future, by combining digital twin simulation, reinforcement learning scheduling and large language model runtime manual generation, the automation level of cross-domain fault suppression will be enhanced within interpretable and secure bounds. The governance system will change from "post-incident explanation" to "pre-propagation control", and Topology orchestration needs to be built as the foundation for a fault-tolerant digital infrastructure.

References

[1] *Artime O, Grassia M, De Domenico M, Gleeson J P, Makse H A, Mangioni G, Perc M, Radicchi F. Robustness and resilience of complex networks[J]. Nature Reviews Physics, 2024, 6(2): 114-131.*

[2] Alkhateeb A, Jiang S, Charan G. Real-Time Digital Twins: Vision and Research Directions for 6G and Beyond[EB/OL]. arXiv:2301.11283, 2023.

[3] Chang, C. W. (2026). Privacy Infrastructure Vulnerability Mining and Automated Framework Based on Multimodal AI. Procedia Computer Science, 279, 702-711.

[4] Wang, Y. (2026). Construction of Supply Chain Demand Forecasting Algorithm Based On Time Series Data Analysis and Improvement of Its Management Efficiency. Procedia Computer Science, 281, 484-493.

[5] Wang, Z. (2026). Mineral Commodity Time-Series Cycle Modeling and Feature Learning Algorithm Based On Improved Transformer. Procedia Computer Science, 281, 1347-1356.

[6] Liang, Q. (2026). Reconstruction of Commercial Building Space Reuse Mode Driven by Composite Business Types. International Journal of Engineering Advances, 3(1), 107-113.

[7] Gao, Y. (2026). Application of Delaware Corporate Law in Cross-Border M&A: Business Structures, Contractual Risk, and Case-Based Lessons. Economics and Management Innovation, 3(2), 128-136.

[8] Yin, J. (2025). Research on Financial Time Series Prediction Model Based on Multi Attention Mechanism and Emotional Feature Fusion. Socio-Economic Statistics Research (2025), 6(2), 161-169.

[9] Liang, Q. (2026). Research on the Renovation Design Path for Enhancing the Efficiency of Mixed-Use Office and Retail Spaces. European Journal of AI, Computing & Informatics, 2(2), 163-170.

[10] Liang, Q. (2026). How Architectural Design and Utility Infrastructure Impacts AI Supporting Campus and Drive Future Innovation, Operational Efficiency and Sustainable Advancement in Utility-Critical Environment. European Journal of Engineering and Technologies, 2(2), 71-77.

[11] Liu, Z. (2026). Research on Measuring User Behavior Response Differences Supported by Propensity Scoring Method. European Journal of AI, Computing & Informatics, 2(2), 47-53.

[12] Gao, Y. (2026). Governance Structures and Checks and Balances in Private Equity Funds: Practice, Problems, and Improvement Paths. Financial Economics Insights, 3(2), 82-91.

[13] Liang, Q. (2026). Research on Low-Energy Space Organization Methods in the Context of Building System Integration. Global Journal of Science & Innovation, 3(1), 9-15.