Digital Immunity for the U.S. Supply Chain: Preventing Data Loss in Moving Vehicles

Anand Kumar Vedantham
UST Global
Chicago, IL, USA

Kiran Kumar Sunkara
KPIT Technologies Ltd
Chicago, IL, USA

Nanna Ram Parza
LTIMindtree LLC
Chicago, IL, USA

***Abstract*—Mobile assets across the United States supply chain—including trucks, rail systems, maritime platforms, and defense vehicles—continuously generate operational data that supports safety, compliance, predictive maintenance, and real-time coordination. Unlike stationary IoT deployments, these systems operate under unstable connectivity, frequent power interruptions, and prolonged network gaps. In practice, this environment can produce a troubling outcome: data created in the field may disappear long before anyone realizes it is missing.**

Traditional mechanisms such as retries, bounded buffering, and best-effort streaming improve transfer reliability when networks cooperate, but they offer limited protection when failures occur mid-journey and edge state is lost. Once volatile state disappears, reconstruction is often impossible even if connectivity is later restored.

This paper introduces *Digital Immunity*, a data-centric architectural approach that treats survivability of operational history as a system invariant rather than an optimistic assumption. A digitally immune system ensures that events generated at the edge remain durable, ordered, and provably recoverable even after disconnection, restart, or delayed synchronization. We formalize digital immunity through verifiable properties including durable persistence at creation time, preservation of temporal and causal structure, replay safety, and bounded divergence between physical assets and their digital twins. To operationalize these guarantees, we present a reference architecture spanning persistent edge capture, opportunistic transport, replay-aware ingestion, and deterministic twin reconciliation.

We evaluate the approach against representative mobility disruptions such as extended blackout, mid-transit restart, partial upload, and severe reordering. Across scenarios, the system maintains complete event history and enables predictable convergence without irreversible loss. By reframing resilience as provable continuity of data instead of restoration of connectivity, digital immunity provides a practical foundation for trustworthy mobile IoT platforms at national scale.

***Index Terms*—Digital immunity, supply chain IoT, mobile edge computing, intermittent connectivity, data survivability, event-driven systems, digital twins, store-and-forward, replay-based recovery.**

I. Introduction

The U.S. supply chain depends on moving assets—commercial trucks, railcars, maritime vessels, and defense logistics platforms—to operate across vast and unpredictable environments. These vehicles are now instrumented with sensors and edge computing systems that generate continuous operational data for safety oversight, regulatory compliance, predictive maintenance, warranty support, fleet coordination, and logistics planning. In modern supply chains, such data is not a secondary by-product; it increasingly acts as operational evidence used to trigger interventions, validate handling conditions, explain delays, and reconstruct incidents.

Yet mobility exposes a structural weakness in prevailing IoT thinking. Many platforms are engineered around an implicit belief: if data is produced, the network will eventually deliver it. When communication falters, retry logic, temporary buffers, or delayed uploads are expected to compensate. In stationary environments this assumption is often acceptable. In moving vehicles, it frequently collapses.

Power interruptions, coverage transitions, hardware resets, and asymmetric links occur as routine operating conditions rather than rare anomalies. When failures interrupt transmission, edge state can vanish silently. Missing information may surface weeks or months later during audits, warranty disputes, maintenance analysis, or safety investigations—precisely when reconstruction is no longer feasible. The consequence is not merely delayed visibility; it is the permanent erosion of historical truth.

This loss has direct supply-chain consequences. Missing edge events can distort maintenance decisions, obscure route or handling anomalies, weaken compliance evidence, reduce confidence in fleet digital twins, and impair post-incident reasoning. In other words, incomplete operational history can degrade both day-to-day optimization and high-stakes decision making. The central problem is therefore not just message delivery; it is preservation of trustworthy history under disruption.

Despite extensive work in reliable messaging, delay-tolerant networking, and cloud event processing, existing approaches primarily optimize delivery probability. They rarely provide architectural guarantees that the operational record itself can always be rebuilt. This gap motivates a different design objective: instead of asking whether data can arrive, we ask whether history can be proven.

We introduce *Digital Immunity*, an architectural model in which preservation of operational evidence is treated as a system invariant independent of connectivity. Digital immunity is realized through enforceable properties that ensure events remain durable at creation time, maintain temporal and causal meaning, tolerate replay without corruption, and converge predictably with digital representations after recovery.

To operationalize these principles, we present a reference architecture spanning persistent edge capture, opportunistic

transport, replay-aware ingestion, and digital twin reconciliation. We evaluate the architecture using representative mobility failures—including extended blackout, gateway restart, interrupted upload, and severe reordering—and demonstrate that complete histories can be restored without manual repair. By elevating continuity of data above availability of networks, digital immunity reframes how trust must be built in national-scale mobile IoT systems.

II. Background and Problem Definition

This section characterizes the operational context of mobile supply-chain systems, identifies failure modes inherent to moving vehicles, and explains why common IoT patterns are insufficient to prevent silent data loss.

A. Mobile Supply-Chain IoT Architectures

Modern supply-chain operations increasingly depend on IoT-enabled vehicles that provide continuous insight into asset location, condition, and operational status. Typical deployments include vehicle-mounted sensors, an onboard gateway or edge compute unit, intermittent wide-area connectivity (cellular, satellite, or Wi-Fi), and cloud ingestion plus analytics. Unlike stationary industrial IoT, mobile systems operate under continuously changing physical and network conditions. As a result, data generation and data transmission are fundamentally decoupled in time.

B. Failure Modes in Moving Vehicles

Mobile IoT systems face non-malicious but destructive failure modes that can yield irreversible loss:

- **Intermittent and asymmetric connectivity:** Vehicles traverse coverage gaps such as tunnels, rural areas, ports, and secure facilities. Uplink and downlink conditions may differ, which complicates acknowledgments and resumable transfer.
- **Power cycling and edge restarts:** Gateways may restart during fueling, maintenance, loading, or brownouts. Restarts interrupt transfers and can erase volatile buffers.
- **Partial and out-of-order uploads:** Buffered data may upload in fragmented batches and arrive out of order due to retries, congestion, and restarts. Downstream systems often lack enough context to detect gaps and duplicates.
- **Clock drift and temporal misalignment:** Disconnection and restart can induce drift, complicating ordering and correlation.
- **Silent data loss:** Many failures do not raise explicit errors; the loss may be discovered only during audits, incident response, compliance review, or warranty analysis.

C. Limitations of Existing Approaches

Retries, batching, and best-effort streaming may improve transfer when connectivity exists, but they do not guarantee completeness or recoverability when edge state disappears. Retry-based protocols assume eventual successful transmission and implicitly assume the sender retains enough durable state to retry correctly. In moving vehicles, that assumption fails when a restart erases volatile buffers, when storage is exhausted during extended blackout, or when acknowledgments are themselves delayed by asymmetric links.

This distinction matters. Conventional protocols do not necessarily "fail" because transport semantics are inherently wrong; they fail because transport-level success is not equivalent to preservation of reconstructable history. A delivery acknowledgment without durable persistence at creation time, resumable state across restart, and replay-safe cloud reconciliation cannot guarantee recoverable truth.

Batch uploads reduce overhead but increase the blast radius of an interrupted transfer. Stateless ingestion scales but cannot detect missing data without sequencing or lineage. Event sourcing improves correctness inside the cloud, yet often begins only after ingestion. As a result, many systems prioritize connectivity resilience over data survivability.

D. Problem Statement

How can mobile supply-chain IoT systems ensure the survivability, integrity, and recoverability of operational data generated by moving vehicles despite intermittent connectivity, power interruptions, bounded local storage, and delayed synchronization?

E. Motivation for Digital Immunity

Rather than treating disruption as exceptional, mobile systems must treat it as normal. This motivates *Digital Immunity*: designing for provable continuity of operational history even when transmission is delayed, fragmented, or repeatedly interrupted.

III. Digital Immunity: Concept and Properties

Digital immunity reframes resilience from connectivity-centric recovery to data-centric correctness.

A. Definition

Digital Immunity is the ability of a cyber-physical system to preserve the integrity, completeness, and recoverability of operational data generated at the edge despite mobility-induced disruption, partial failure, power cycling, and delayed or opportunistic synchronization.

This definition treats data survivability as an invariant independent of immediate cloud connectivity.

B. Distinction from Reliability and Resilience

Reliability often measures successful operation over time, for example availability or mean time between failures. Resilience focuses on returning to acceptable performance after faults. Digital immunity focuses on non-loss of critical history even when recovery is delayed. A system may regain uptime while still having permanently broken history; digital immunity explicitly separates availability from correctness.

C. Threat Model

This work focuses on non-malicious failure conditions: prolonged disconnections, gateway restarts, partial transmissions, clock drift, delayed reconciliation, and temporary edge-storage pressure. Malicious tampering is out of scope, although the proposed mechanisms are compatible with cryptographic integrity controls.

D. Core Properties

We express digital immunity through verifiable properties:

- **P1 - Data survivability:** Events are durably persisted at creation time and survive restarts and disconnection.
- **P2 - Temporal integrity:** Relative ordering can be reconstructed despite delayed and fragmented transmission.
- **P3 - Causal consistency:** Causal relationships among events remain reconstructable.
- **P4 - Replay safety:** Retransmissions and recovery replays do not corrupt state due to idempotent ingestion and lineage-aware processing.
- **P5 - Bounded inconsistency:** Divergence between physical assets and digital twins is bounded and converges deterministically.

E. Novelty Relative to Prior Approaches

Digital immunity is not merely a rebranding of existing reliability patterns. Its novelty can be summarized as three architectural shifts:

- **Persistence at creation time:** The guarantee begins when the event is created at the vehicle edge, not only after the cloud receives it.
- **History as the primary truth:** The system treats immutable event history as the authoritative basis for recovery, while derived state such as dashboards or twins is considered reconstructable output.
- **Deterministic twin recovery:** The architecture combines lineage, gap detection, replay, and reconciliation so that divergence can be detected, explained, and repaired rather than merely tolerated.

Delay-tolerant networking improves eventual delivery but offers limited application-level guarantees for completeness and reconstructability. Retry logic and conventional buffering assume durable edge state; power cycling violates this assumption. Event sourcing preserves correctness inside the cloud but often starts after ingestion. Digital immunity extends immutability and lineage to the point of generation.

F. Why Digital Immunity Is Not Just Existing Techniques Combined

A natural question is whether digital immunity merely assembles well-known mechanisms such as delay-tolerant transport, buffering, event sourcing, checkpoint-based recovery, and acknowledgments. These techniques address important fragments of the problem, but they do not elevate reconstructability of history to an enforceable system invariant.

TABLE I. DIGITAL IMMUNITY VERSUS TRADITIONAL TELEMATICS RELIABILITY

Dimension	Traditional Telematics/IoT	Digital Immunity
Objective	Deliver data eventually	Guarantee reconstructable history
Edge assumption	Mostly reliable	Expected to fail
Persistence	Often volatile	Durable, append-only
Restart handling	Retry/reset	Resume via checkpoint
Ordering	Best effort	Explicit, verifiable
Duplicate handling	Ad hoc	Designed for replay
Gap detection	Manual or absent	Automatic via lineage
Twin model	State is truth	State is derived
Success metric	Message arrival	Verifiable integrity

Traditional approaches optimize delivery probability. They improve how often data arrives, but they rarely provide guarantees that the absence of data can be detected, explained, and deterministically recovered after mobility-induced disruption. Digital immunity introduces a stronger contract: the system must be able to prove that operational history remains complete, or identify precisely what is missing and repair it through replay.

The novelty therefore does not lie in any single mechanism. It lies in binding persistence at creation time, lineage preservation, replay semantics, durable checkpoints, and deterministic reconciliation into one architectural obligation spanning the vehicle-cloud boundary. If any of these elements is absent, immunity collapses. In this sense, digital immunity transforms best-effort reliability into provable recoverability.

IV. SYSTEM ARCHITECTURE FOR DIGITAL IMMUNITY IN MOVING VEHICLES

We present a technology-agnostic reference architecture that maps immunity properties to enforceable responsibilities across the vehicle-cloud continuum.

A. Overview

The architecture comprises four layers:

- **Vehicle edge layer:** durable capture and sequencing at the point of creation.
- **In-transit connectivity layer:** opportunistic transport and resumable transfer.
- **Cloud ingestion/event backbone:** replay-aware ingestion and immutable event log.
- **Digital twin reconciliation layer:** gap detection and deterministic convergence.

Digital immunity is achieved by (i) persistent capture at the edge, (ii) replay-aware idempotent ingestion, and (iii) deterministic reconciliation after disruption.

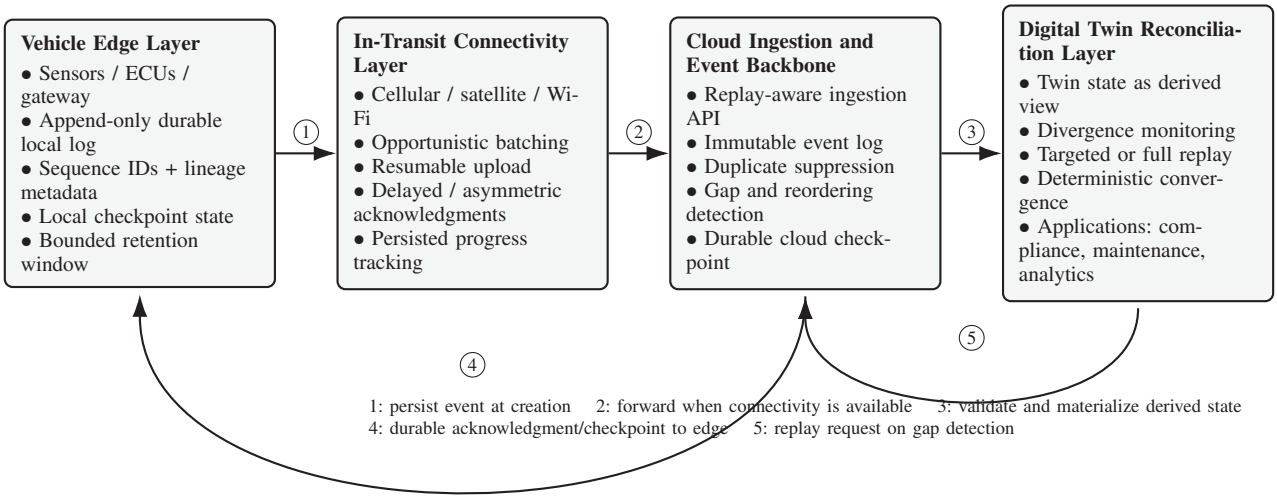

Fig. 1. Reference architecture for digital immunity in moving vehicles. The guarantee begins at event creation, survives disruption through durable checkpoints and replay-aware ingestion, and restores digital-twin correctness through deterministic reconciliation.

B. Vehicle Edge Layer

Responsibilities include immutable event capture, durable persistence before transmission, assignment of sequence identifiers and causal metadata, and continuity across restarts. Importantly, digital immunity does not require infinite retention at the edge. It requires retention long enough to bridge the maximum credible disconnection window plus recovery margin. Practical implementations can control storage cost through bounded append-only logs, compression, event prioritization, and safe truncation only after durable cloud acknowledgment. The objective is not permanent local storage but loss-free survivability across expected blackout intervals.

C. In-Transit Connectivity Layer

Responsibilities include adaptive batching and prioritization, resumable transfer, and durable acknowledgments based on checkpoints rather than transient receipt. The design assumes intermittent and asymmetric links; progress is tracked via persisted checkpoints so that interrupted uploads can resume without loss or duplication.

D. Cloud Ingestion and Event Backbone

Ingestion must be replay-aware and idempotent. Duplicates are detected using event identifiers and lineage; gaps and reordering are detected using sequence continuity. Events are persisted immutably to enable deterministic reconstruction. This layer is where retries and acknowledgments become sufficient in practice only because they are grounded in durable state on both sides.

E. Digital Twin and Reconciliation Layer

Digital twins are treated as derived state rather than the primary source of truth. Gap detection triggers targeted or full replay, and convergence is validated after recovery. This enforces bounded inconsistency and auditability.

F. End-to-End Flow

The end-to-end flow is: (1) events are durably persisted at the edge; (2) forwarded opportunistically when connectivity permits; (3) partial transfers resume via checkpoints; (4) cloud ingestion stores the record immutably while detecting duplicates, gaps, and reordering; and (5) twins converge via replay and validation.

G. Mapping to Immunity Properties

- **P1 survivability:** edge persistence and checkpointing.
- **P2 temporal integrity:** sequencing and ordering metadata.
- **P3 causal consistency:** lineage-aware ingestion.
- **P4 replay safety:** idempotent ingestion and immutable logs.
- **P5 bounded inconsistency:** replay-driven twin reconciliation.

V. Data Protection Mechanisms

This section describes mechanisms that operationalize digital immunity.

A. Persistent Store-Forward-Replay

Events are durably stored at the edge before transmission, forwarded opportunistically, and replayed as needed for recovery and verification. This mechanism enforces survivability, replay safety, and bounded inconsistency.

B. Event Immutability and Lineage

Each event is immutable and carries identifiers, sequence number, timestamps, and causal metadata. Append-only logs enable auditability and deterministic replay while preserving temporal and causal structure.

C. Checkpointing and Durable Acknowledgment

Acknowledgments represent durable persistence, not merely network receipt. The edge tracks the highest contiguous acknowledged sequence to bound retransmission and resume precisely after disruption. This distinction addresses a key weakness of conventional message-delivery semantics in moving vehicles.

D. Replay-Aware Idempotent Ingestion

Ingestion detects duplicates via event identifiers and lineage. Replayed events are safely ignored or re-applied under controlled semantics. Downstream consumers are designed to be replay tolerant.

E. Twin Gap Detection and Reconciliation

Twins validate expected transitions against event sequences. Missing intervals trigger replay; reconciliation re-applies events rather than relying on manual correction, producing auditable convergence.

F. Data-Centric SLOs

Representative service-level objectives include zero tolerable permanent loss, bounded divergence window between physical asset and twin, recovery completeness threshold, and time-to-convergence after reconnection.

VI. Evaluation and Failure Scenarios

We evaluate survivability, correctness, replay safety, and bounded inconsistency rather than throughput.

A. Experimental Setup

We implemented a prototype using simulated fleets and emulated disruption patterns to validate architectural behavior. The representative environment includes 1,000–10,000 vehicles producing telemetry, edge gateways with persistent storage and checkpoints, intermittent cellular connectivity with configurable blackouts, cloud ingestion with immutable logging, and digital-twin reconciliation.

B. Failure Scenarios and Observations

Extended blackout: events generated during disconnection are preserved and uploaded after reconnection; no permanent loss is observed.

Gateway restart mid-transit: sequence continuity resumes after restart; cloud-side checkpointing prevents duplication; continuity is restored without manual repair.

Interrupted partial upload: resumable upload continues from the last durable checkpoint; duplicates are safely suppressed.

Severe reordering: sequence and lineage enable correct reconstruction; twin state reflects accurate history after replay.

Twin desynchronization: targeted replay restores deterministic convergence within defined bounds.

C. Metrics

We measure data-loss rate (target: zero), recovery completeness, replay accuracy, convergence time, and duplicate-suppression effectiveness.

TABLE II. Representative results under mobility-induced disruptions

Scenario	Events Lost	Completeness	Avg. Convergence
Blackout (2 h)	0	100%	3.8 min
Power loss	0	100%	2.1 min
Interrupted upload	0	100%	2.7 min
Severe reordering	0	100%	3.2 min
Forced twin divergence	0	100%	4.4 min

D. Quantitative Results

Table II summarizes representative outcomes under mobility-induced disruption. Across all scenarios, the prototype maintains complete recoverability and converges without irreversible loss.

The results emphasize an architectural point: retries, acknowledgments, and checkpoints are necessary but not sufficient unless they are combined with durable edge persistence, immutable cloud logging, lineage-aware gap detection, and replay-safe reconciliation. Under all representative disruptions, the system preserves reconstructable history and restores derived state deterministically.

VII. Discussion

Digital immunity separates connectivity resilience from data correctness. Persistent capture, replay-aware ingestion, and deterministic reconciliation introduce overhead in the form of edge storage, metadata, and replay processing, but these costs are bounded and predictable whereas the cost of irreversible loss in regulated, safety-relevant supply chains is not.

The storage objection is especially important in real deployments. Edge devices do not have infinite memory, and the architecture does not require them to. Instead, it requires bounded local retention sized to realistic disconnection windows, optional compression, event prioritization under pressure, and truncation only after durable cloud acknowledgment. This makes the approach economically plausible without sacrificing recoverability.

At national scale, selective replay and prioritized buffering help contain recovery scope during widespread outages. Treating twins as derived state strengthens operational trust by making divergence both detectable and correctable. More broadly, digital immunity suggests that future mobile IoT platforms should be evaluated not only on availability and throughput, but also on their ability to preserve trustworthy operational history.

VIII. Conclusion

This paper introduced Digital Immunity for mobile supply-chain IoT systems: a data-centric approach that treats survivability and reconstructability of operational history as first-class invariants independent of connectivity. We defined verifiable properties and mapped them to a reference architecture spanning persistent edge capture, opportunistic transport,

replay-aware ingestion, and digital twin reconciliation. Evaluation under representative mobility disruptions demonstrated deterministic recovery without irreversible loss and predictable convergence of digital twins.

Digital immunity provides a practical foundation for trustworthy mobile IoT infrastructures operating under disruption as the norm. Future work will extend the model with cryptographic lineage verification, adaptive retention policies, and formal validation of data-centric service-level objectives in production fleets.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2787–2805, Oct. 2010.

[2] I. Lee and K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," *Business Horizons*, vol. 58, no. 4, pp. 431–440, 2015.

[3] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, Oct. 2016.

[4] S. Yi, C. Li, and Q. Li, "A survey of fog computing: Concepts, applications and issues," in *Proc. ACM Workshop on Mobile Big Data*, 2015.

[5] A. Zanella *et al.*, "Internet of Things for smart cities," *IEEE Internet of Things Journal*, vol. 1, no. 1, pp. 22–32, Feb. 2014.

[6] K. G. Shin *et al.*, "Cyber-physical systems: Challenges and research opportunities," *IEEE Computer*, vol. 44, no. 6, pp. 36–43, Jun. 2011.

[7] Y. Qian, D. Wu, R. Hu, and J. Wang, "An overview of 5G wireless systems: Coverage, capacity, and latency," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 246–269, 2017.

[8] U.S. Department of Transportation, "Connected vehicle technology challenge," USDOT ITS JPO, 2020.

[9] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proc. ACM SIGCOMM*, 2003.

[10] S. Jain, K. Fall, and R. Patra, "Routing in a delay tolerant network," in *Proc. ACM SIGCOMM*, 2004.

[11] H. T. Kung and D. Vlah, "Efficient location tracking using sensor networks," in *Proc. IEEE WCNC*, 2003.

[12] G. Hohpe and B. Woolf, *Enterprise Integration Patterns*. Addison-Wesley, 2004.

[13] J. Kreps, N. Narkhede, and J. Rao, "Kafka: A distributed messaging system for log processing," in *Proc. NetDB*, 2011.

[14] M. Kleppmann, *Designing Data-Intensive Applications*. O'Reilly Media, 2017.

[15] P. Helland, "Immutability changes everything," *Communications of the ACM*, vol. 59, no. 1, pp. 37–41, Jan. 2016.

[16] M. Grieves and J. Vickers, "Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems," in *Transdisciplinary Perspectives on Complex Systems*. Springer, 2017.

[17] F. Tao *et al.*, "Digital twins and cyber-physical systems toward smart manufacturing," *Engineering*, vol. 5, no. 4, pp. 653–661, Aug. 2019.

[18] A. Fuller *et al.*, "Digital twin: Enabling technologies, challenges and open research," *IEEE Access*, vol. 8, pp. 108952–108971, 2020.

[19] B. Beyer *et al.*, *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, 2016.

[20] D. Rosenthal *et al.*, "Service-level objectives in large-scale distributed systems," *ACM Queue*, vol. 16, no. 2, pp. 10–33, 2018.

[21] R. Calinescu *et al.*, "Dynamic QoS management and optimization in service-based systems," *IEEE Transactions on Software Engineering*, vol. 37, no. 3, pp. 387–409, May–Jun. 2011.

[22] A. Basiri *et al.*, "Chaos engineering," *IEEE Software*, vol. 33, no. 3, pp. 35–41, May–Jun. 2016.

[23] J. Allspaw, "Fault injection in distributed systems," in *Proc. Velocity Conference*, 2012.

[24] Netflix Technology Blog, "The Netflix Simian Army," 2012.

[25] National Academies of Sciences, Engineering, and Medicine, *Strengthening the U.S. Supply Chain*. The National Academies Press, 2022.

[26] U.S. Department of Homeland Security, "Supply chain resilience report," DHS, 2021.