INTERNATIONAL MULTI DISCIPLINARY JOURNAL FOR RESEARCH & DEVELOPMENT
eISSN 2394-6334 Impact factor: 7,854 Volume 12, issue 11 (2025)

HOLISTIC DEVSECOPS FOR DISTRIBUTED SYSTEMS: UNIFYING ZERO TRUST ARCHITECTURE, BLOCKCHAIN PROVENANCE, AND INTELLIGENCE-DRIVEN SECURITY AUTOMATION

Isabelle C. Moreau
Department of Resilient Systems Architecture, Sorbonne University, France

Abstract: Background: Modern software delivery environments—characterized by microservices, cloud-native architectures, distributed ledgers, and cyber-physical integrations—present complex security challenges that traditional perimeter-based defenses cannot adequately address. This paper proposes a cohesive, research-grounded framework that integrates Zero Trust principles, selective blockchain primitives, automated threat intelligence, and adaptive risk-aware controls into Dev SecOps pipelines. Objective: To articulate a theoretically rigorous, practically implementable architecture and methodology for embedding continuous, automated security validation into the software delivery lifecycle while managing scalability, cost, and regulatory constraints.
Methods: We synthesize cross-disciplinary literature (security engineering, blockchain, DevOps adoption studies, cyber-physical systems, and post-quantum planning) to design an integrative model; we then describe procedural instantiations, developer interaction patterns, and governance constructs to operationalize the model.
Results: The conceptual framework yields traceable security attestations, improved anomaly detection surfaces for CPS telemetry, and a policy-driven automation layer that minimizes human slowdowns without sacrificing control.
Conclusions: Combining Zero Trust controls, selective blockchain anchoring for provenance, and automated CTI-driven gating provides a resilient path for Dev SecOps evolution. Realizing the framework requires targeted investments in developer education, tooling alignment, and phased regulatory mapping.

Keywords: Dev SecOps, Zero Trust, blockchain provenance, threat intelligence, cyber-physical systems, post-quantum readiness.

Introduction

The contemporary software and systems landscape has evolved from monolithic, centrally hosted applications into a tapestry of distributed microservices, serverless functions, edge compute, and cyber-physical systems (CPS). This transformation delivers substantial benefits—scalability, fault tolerance, faster deployment velocity—but simultaneously expands the attack surface and complicates security governance. Early thinking about perimeter defense, often metaphorized as a "chewy center" with a soft interior protected by a hardened outer crust, has been repeatedly criticized as inadequate for these distributed, dynamic environments (Kindervag, 2010). Zero Trust, which rejects implicit trust and advocates continuous verification and least-privilege access, has emerged as an organizing principle for addressing this conceptual mismatch (Kindervag, 2010).

At the same time, emerging technologies such as blockchain propose architectural primitives—immutability, decentralized consensus, verifiable provenance—that can be harnessed to strengthen DevSecOps artifacts like supply chain provenance, code attestations, and distributed configuration audits (Swan, 2015; Staples et al., 2018). Integrating blockchain requires careful analysis: benefits exist, but there are clear trade-offs in scalability, cost, and environmental/resource footprints (Staples et al., 2018; Chavan, 2023). Moreover, modern threat landscapes have become highly automated and intelligence-driven: cyber threat intelligence

(CTI) integration into the CI/CD lifecycle offers possibilities to detect and mitigate risks early, potentially "before code hits production" (Malik, 2025; Ghura, 2023).

Cyber-physical systems (CPS) introduce unique constraints: telemetry streams, physical safety, timing constraints, and heterogeneous communication channels—all of which must be incorporated into security assurance mechanisms (Humayed et al., 2017). Additionally, forward-looking concerns, such as post-quantum cryptographic transitions, complicate long-lived attestations and must be considered in provenance and signature strategies (NIST, 2024). Regulatory regimes—PCI DSS for payment systems, industry breach reporting norms, and corporate governance expectations—also shape acceptable designs (PCI Security Standards Council, 2018; Verizon, 2024). Consequently, there is a pressing need for an integrated, adaptive DevSecOps framework that harmonizes Zero Trust controls, blockchain-enabled provenance, automated CTI gating, and CPS-aware anomaly detection while remaining cognizant of cost and scalability.

This paper fills a literature gap by articulating a unified, operationally grounded framework that synthesizes these threads. While prior work addresses individual components—Zero Trust models (Kindervag, 2010), blockchain opportunities for DevSecOps (Staples et al., 2018), CTI automation techniques (Ghura, 2023), CPS security surveys (Humayed et al., 2017)—few works present an end-to-end architecture and practical method for implementing them together within the CI/CD pipeline while addressing developer experience, cost trade-offs, and regulatory compliance. Our aim is to provide researchers and practitioners with a detailed, theoretically supported blueprint for implementation, evaluation, and extension.

Methodology

Our approach is analytic and synthetic: we conduct a structured literature synthesis across the provided references to identify core functional requirements, failure modes, and enabling technologies. From this synthesis, we formulate an architectural framework and a set of implementation patterns. The methodology comprises the following steps:

1. Literature Synthesis and Gap Analysis: We systematically reviewed the provided corpus—spanning Zero Trust, blockchain, DevOps adoption studies, CTI scaling, CPS security, and cryptographic standardization—to extract recurring themes, success factors, and shortcomings. For example, Zero Trust emphasizes continuous assertion of identity and policy enforcement but provides limited guidance on provenance for artifacts (Kindervag, 2010). Blockchain offers immutable provenance but struggles with cost and throughput (Swan, 2015; Staples et al., 2018). CTI literature emphasizes automation and enrichment for scale but notes the challenges of signal-to-noise and developer acceptance (Ghura, 2023; Danilova et al., 2020).

2. Functional Decomposition: We broke down the end-to-end DevSecOps lifecycle into discrete functional domains: identity and access control; code and artifact provenance; automated threat detection and response; runtime telemetry and anomaly detection (especially for CPS); policy, governance, and compliance mapping; and developer interaction and usability. Each domain was mapped to candidate technical controls and tooling primitives found in the literature (Freeman & Harvey, 2020; Gartner, 2019).

3. Architectural Integration: Using the functional maps, we designed an adaptive fabric that places security assertions, verifiable provenance, and CTI gating into the CI/CD pipeline while enforcing Zero Trust at the service mesh and orchestration levels. Blockchain primitives are used selectively for high-value provenance anchoring; CTI is operationalized as automated policy feeds that can gate merges or stage promotions (Malik, 2025; Staples et al., 2018).

4. Operational Patterns and Scripts (Textual): Recognizing the user's constraints and the prohibition on executable artifacts in this manuscript, we outline procedural, text-based "playbooks" for implementing the framework: how to instrument pipelines for attestations, when to anchor to a ledger, how to tune CTI thresholds, and how to incorporate CPS telemetry into anomaly models (He et al., 2022).

5. Evaluation Criteria: We propose a set of qualitative and quantitative evaluation metrics, derived from the literature, suited to measure security posture improvements, developer throughput impacts, cost metrics, and CPS safety outcomes. These metrics include time-to-detect, mean time to remediate, false positive rates for developer warnings, and end-to-end provenance verification time.

Each methodological step draws directly from the cited literature to ensure claims are anchored in established research and practitioner guidance (Kindervag, 2010; Staples et al., 2018; Ghura, 2023; He et al., 2022; PCI Security Standards Council, 2018).

Results

The synthesis yields a comprehensive, layered framework—hereafter called the Adaptive DevSecOps Fabric (ADF)—and a set of procedural playbooks for adoption. The results are descriptive and prescriptive rather than empirically measured within this paper; they are intended to function as a rigorously argued design artifact ready for empirical validation.

1. Architecture Overview of the Adaptive DevSecOps Fabric (ADF):

The ADF is organized into five concentric, interacting layers:

- Identity & Continuous Authorization Layer: Implements Zero Trust principles across build agents, CI runners, container registries, orchestration planes, and operator consoles. Policies require per-action credentials and short-lived artifacts, eliminating implicit trust for systems and human actors (Kindervag, 2010).

- Provenance & Attestation Layer: Produces immutable metadata at critical touchpoints: source control commit, build artifact creation, container image signing, test-suite completion, and deployment. Selective blockchain anchoring is used to record cryptographic hashes and attestation metadata for high-sensitivity artifacts, enabling later third-party verification without revealing private artifacts (Swan, 2015; Staples et al., 2018).

- CTI-Driven Policy & Automation Layer: Feeds automated threat intelligence into gating policies. CTI sources supply IoCs, TTP mappings, and behavioral signatures that the pipeline uses to flag risky dependencies, known-bad packages, or suspicious artifact provenance. Automation enforces policies via fail/hold actions or escalations where appropriate (Ghura, 2023; Malik, 2025).

- CPS Telemetry & Anomaly Detection Layer: For CPS and edge systems, the ADF integrates telemetry collection and sparse-feature anomaly detection methods tailored for constrained channels, following approaches proven in satellite telemetry anomaly detection (He et al., 2022; Humayed et al., 2017).

- Governance, Compliance & Cost Management Layer: Maps PCI DSS and other regulatory controls to pipeline gates and artifact retention policies. It also includes cost control mechanisms to avoid runaway expenses from naive blockchain anchoring or overly conservative resource allocations (PCI Security

Standards Council, 2018; Chavan, 2023).

2. Operational Playbooks (Textual):

We provide detailed playbooks for key operations.

- Signing and Attestation Playbook: Every build process emits an attestation bundle containing build environment fingerprint, dependency snapshot (with cryptographic hashes), test-suite summary, and operator approvals. The bundle is signed with a short-lived build key whose lifecycle is governed by identity layer policies. For artifacts requiring independent verifiability, the attestation bundle's global digest is anchored to a permissioned blockchain ledger accessible by authorized stakeholders. The anchoring process is batched to reduce cost and only includes minimal metadata required for verification, preserving confidentiality while ensuring integrity (Staples et al., 2018; Swan, 2015).

- CTI Gating Playbook: CTI feeds are ingested, normalized, and scored against artifact metadata and dependency manifests. If an artifact imports a dependency matching a high-confidence IoC or exhibiting suspicious provenance, the pipeline flags the commit and either halts promotion or inserts an automated mitigation (patch blocking, dependency pinning, or expanded testing). The gating policy is parameterized by confidence thresholds to minimize developer friction (Ghura, 2023; Danilova et al., 2020).

- CPS Telemetry Playbook: Edge and CPS devices stream compressed telemetry to an ingestion buffer. A sparse-feature anomaly detection model—similar in principle to the approach used for satellite telemetry—processes the stream to identify outliers in sensor correlations, timing anomalies, or protocol deviations. Anomalies trigger staged responses: local safe-mode commands, operator alerts with provenance links back to the deployed artifact, and quarantining of suspect control channels (He et al., 2022; Humayed et al., 2017).

3. Expected Security Posture Improvements:

Based on cross-citing literature on attack patterns and breach data, the ADF is designed to materially reduce common failure modes:

- Supply Chain Compromise: By enforcing provenance and CTI-driven dependency checking at CI time, the pipeline reduces the likelihood of malicious dependency ingress into production (Staples et al., 2018; Ghura, 2023).

- Lateral Movement & Privilege Escalation: Zero Trust continuous authorization prevents long-lived credentials and implicit trust relationships often exploited by attackers (Kindervag, 2010).

- Delayed Detection in CPS: Telemetry-based anomaly detection reduces mean time to detect safety-relevant anomalies in CPS deployments, addressing the unique requirements of physical-systems security (He et al., 2022; Humayed et al., 2017).

4. Trade-offs and Cost Considerations:

Selective use of blockchain anchoring minimizes transaction costs and maximizes utility. Instead of anchoring every artifact or event, the framework prescribes anchoring only for artifacts that are high-risk, long-lived, or require third-party verifiability—balancing the immutable ledger's strengths against cost and scalability constraints (Staples et al., 2018; Chavan, 2023).

5. Developer Experience and Usability Outcomes:

The literature emphasizes that developer acceptance hinges on signal quality and non-intrusiveness of security controls (Danilova et al., 2020). Accordingly, ADF emphasizes high-precision CTI scoring, context-aware warnings, and in-pipeline self-service remediation guidance to avoid "alert fatigue" among developers (Danilova et al., 2020; Freeman & Harvey, 2020).

Discussion

This section interprets the ADF design, elaborating theoretical implications, addressing limitations, considering counter-arguments, and mapping future research directions.

1. Theoretical Interpretation: Synergy of Zero Trust and Verifiable Provenance

Zero Trust shifts the security model from perimeter-based controls to verification at each access point; however, it does not inherently provide artifact-level non-repudiation or long-term provenance. Conversely, blockchain provides immutable provenance but lacks fine-grained, dynamic authorization logic. The ADF demonstrates that combining these paradigms produces emergent benefits: Zero Trust reduces trust dependencies in runtime interactions while verifiable provenance ensures that artifact claims can be independently corroborated after the fact. The strategic use of blockchain as a distributed, tamper-evident anchor—rather than as a transactional substrate for runtime logic—allows the system to maintain high assurance of supply chain integrity without surrendering scalability (Kindervag, 2010; Swan, 2015; Staples et al., 2018).

2. Automated CTI Integration: Balancing Speed, Precision, and Developer Workflows

CTI offers the promise of preemptively blocking or flagging risky artifacts, but the literature warns about noisy signals and the human cost of false positives (Ghura, 2023; Danilova et al., 2020). Practically, CTI must be operationalized through staged confidence thresholds and contextual enrichment: raw indicators should be correlated with dependency graphs, runtime vulnerability databases, and recent exploit telemetry. Lower-confidence matches inform non-blocking notifications and expanded tests; higher-confidence matches trigger pipeline holds and automated mitigations. This staged model harmonizes CTI with developer workflows and reduces cognitive load—a necessity underscored by developer preference studies for tailored, context-aware warnings (Danilova et al., 2020).

3. CPS and Telemetry: Safety, Timing, and Data Quality Challenges

CPS present distinct challenges: sensor noise, intermittent connectivity, and critical timing constraints render traditional anomaly detection approaches inadequate. The use of sparse-feature models in telemetry has proven effective in satellite contexts and is adaptable to other CPS domains (He et al., 2022). However, CPS require carefully tuned thresholds and fail-safe responses because false positives can trigger unnecessary physical interventions, while false negatives risk safety-critical failures. The ADF prescribes multi-signal confirmation—combining telemetry anomalies with provenance context and CTI signals—before initiating aggressive mitigations, thereby hedging against both false triggers and missed detections (Humayed et al., 2017; He et al., 2022).

4. Regulatory Alignment and Evidence Chains

Regulatory frameworks such as PCI DSS impose specific controls on data handling and logging. The

ADF's provenance and attestation logs can serve both operational and compliance functions, but the use of immutable anchors must be reconciled with data retention, privacy, and right-to-erasure requirements. The selective anchoring strategy helps: by anchoring only cryptographic digests and minimal metadata, the system avoids exposing sensitive payloads while maintaining verifiability. Governance procedures must ensure that anchored metadata is non-identifying or that appropriate access controls and legal frameworks permit such anchoring under applicable regulations (PCI Security Standards Council, 2018).

5. Economic Trade-offs: Scalability, Cost, and Environmental Footprint

Blockchain anchoring introduces ongoing transactional costs and, depending on the consensus mechanism, environmental considerations. Recent analyses and practitioner guidance suggest using permissioned ledgers or lightweight anchoring techniques—such as batching digests and anchoring a single Merkle root per epoch—to reduce costs (Staples et al., 2018; Chavan, 2023). The ADF emphasizes such strategies, recommends cost allocation metrics for teams, and proposes guardrails to prevent uncontrolled ledger usage that would erode the financial benefits of cloud-native scaling (Chavan, 2023).

6. Limitations and Counter-Arguments

No framework is a panacea. Critics may argue that adding provenance layers and CTI gating increases complexity and slows time-to-deploy. We counter that the ADF prioritizes developer experience: gating is staged, and only high-confidence interventions block pipelines. Moreover, the security posture improvements—reductions in supply chain compromise risk, faster detection of CPS anomalies, and more robust forensic evidence—justify incremental friction when carefully managed (Danilova et al., 2020; Staples et al., 2018). Another counterpoint is that Zero Trust and blockchain commitments may be misapplied: treating Zero Trust as a checklist rather than a continuous cultural practice or overusing blockchain as a "silver bullet" are real risks. The ADF explicitly frames these primitives as part of a socio-technical system: technical controls must be paired with governance, education, and periodic review (Kindervag, 2010; Staples et al., 2018).

7. Future Research Directions

Several research avenues arise naturally from the ADF:

- Empirical Evaluation: Implement controlled deployments in representative organizations—one in financial services subject to PCI DSS constraints and one in CPS-heavy industry—to measure the ADF's impact on security metrics, developer productivity, and cost. Metrics should include time-to-detect, false positive/negative rates, developer cycle times, and end-to-end verification latency.

- Automated Policy Learning: Investigate machine learning approaches for dynamically tuning CTI confidence thresholds and pipeline gating rules based on historical outcomes—balancing risk reduction with developer throughput.

- Post-Quantum Provenance Models: Research methods for future-proofing attestations against quantum adversaries, including hybrid signature schemes, post-quantum key rotation policies, and archival strategies that enable later re-verification (NIST, 2024).

- Human Factors Studies: Conduct qualitative research into developer perceptions of CTI-driven warnings and blockchain attestation feedback mechanisms to optimize UX and adoption (Danilova et al., 2020).

- Cross-Organizational Verification Protocols: Explore privacy-preserving multi-party verification protocols allowing third parties (auditors, partners) to validate provenance without exposing sensitive artifacts or intellectual property.

Conclusion

This paper presents the Adaptive DevSecOps Fabric (ADF), a comprehensive framework integrating Zero Trust continuous authorization, selective blockchain-based provenance anchoring, automated threat intelligence gating, and CPS-aware telemetry analysis into the CI/CD pipeline. The ADF is designed to be adaptive: it prescribes selective application of heavy-weight controls (ledger anchoring) and lightweight, developer-friendly automation (staged CTI warnings). By harmonizing technical controls with governance and cost-management practices, the framework aims to materially reduce supply chain risk, limit lateral movement, and improve detection of CPS anomalies while preserving developer velocity.

Achieving the ADF in practice requires careful incremental adoption: instrument pipelines for attestations; implement CTI ingestion and staged gating; tune telemetry anomaly detectors for CPS environments; and employ selective ledger anchoring for high-value artifacts. Future empirical work is necessary to quantify the framework's benefits and iterate on tuning strategies. The synthesis in this paper establishes a theoretically grounded and practically oriented foundation for those efforts, offering a research agenda and procedural playbooks to guide both researchers and practitioners toward more resilient, transparent, and efficient software delivery systems.

References

1. Swan, M. (2015). Blockchain: Blueprint for a New Economy. O'Reilly Media.
2. Kindervag, J. (2010). "No More Chewy Centers: Introducing the Zero Trust Model of Information Security." Forrester Research.
3. Verizon. (Annual publication). Verizon Data Breach Investigations Report.
4. Gartner. (2019). Continuous Delivery and DevOps: A Survey of Adoption.
5. PCI Security Standards Council. (2018). Payment Card Industry Data Security Standard (PCI DSS) v3.2.1.
6. Lange, F. (2017). "Fostering Collaboration in Cross-Functional Teams," IEEE Engineering Management Review, 45(3), 24–31.
7. NIST. (2024). "Post-Quantum Cryptography: NIST's Plan for the Future." https://csrc.nist.gov/projects/post-quantum-cryptography
8. Humayed, A., Lin, J., Li, F., & Luo, B. (2017). "Cyber-Physical Systems Security—A Survey," IEEE Internet of Things Journal, 4(6), 1802–1831.
9. Staples, M., et al. (2018). "Risks and Opportunities of Blockchain for DevSecOps," IEEE Software, 35(4), 47–53.
10. Chavan, A. (2023). "Managing scalability and cost in microservices architecture: Balancing infinite scalability with financial constraints," Journal of Artificial Intelligence & Cloud Computing, 2, E264.

http://doi.org/10.47363/JAICC/2023(2)E264

11. Malik, G. (2025). "Integrating Threat Intelligence with DevSecOps: Automating Risk Mitigation before Code Hits Production," Utilitas Mathematica, 122(2), 309–340.

12. Corecco, S., Adorni, G., & Gambardella, L. M. (2023). "Proximal policy optimization-based reinforcement learning and hybrid approaches to explore the cross array task optimal solution," Machine Learning and Knowledge Extraction, 5(4), 1660–1679.

13. Danilova, A., Naiakshina, A., & Smith, M. (2020, June). "One size does not fit all: a grounded theory and online survey study of developer preferences for security warning types." In Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering (pp. 136–148).

14. Freeman, E., & Harvey, N. (2020). 97 Things Every Cloud Engineer Should Know. O'Reilly Media.

15. Ghura, B. S. (2023). "Scaling & Automating Cyber Threat Intelligence (CTI) Operations with Free and Open-source Software (FOSS)."

16. He, J., Cheng, Z., & Guo, B. (2022). "Anomaly detection in satellite telemetry data using a sparse feature-based method," Sensors, 22(17), 6358.