**Journal of Information Systems Engineering and Management**
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

# AI-Driven Engineering Challenges in PCI-Compliant Mobile IoT Shipping Systems

Sai Krishna Paladugu
FedEx Services, USA

**ARTICLE INFO**

Received: 20 Feb 2026

Revised: 22 Feb 2026

**ABSTRACT**

Enterprise shipping solutions for large enterprises have evolved into a distributed network of mobile IoT endpoints that coordinate payment processing, package handling, and shipping label printing through a global network of regional hubs. The article discusses the engineering challenges of incorporating artificial intelligence into PCI-compliant mobile shipping platforms that are separated by a very strict security perimeter to provide predictive optimization without exposing cardholder data. The mobile revenue capture, hardware state management, and regulatory compliance systems do not allow for customary reactive engineering at scale. AI acts as a predictive engineering layer that lives outside the payment environments and takes telemetry from the hardware, transaction data, and workflow context to identify the causes of future instability before they happen. This shows how clever automation and regulatory compliance can co-exist when systems are designed to deliver measurable increases in transaction success and operational efficiencies in distributed, high-volume environments.

**Keywords:** Mobile IoT Shipping Platforms, PCI Compliance Architecture, Predictive Reliability Engineering, Distributed Payment Systems, AI-Driven Workflow Optimization

## Introduction

Modern enterprise shipping software is defined as a distributed system of payment terminals, barcode scanners, thermal printers, kiosks, and enterprise-managed tablets. In the world of the Internet of Things (IoT) these devices are the shipping industry's IoT platforms and are architecturally distinct from point-of-sale systems because they are distributed, have high transaction volumes for shipping/return activities, are subject to PCI DSS, and purposefully operate in real-time. As of 2024, the global shipping endpoint systems, or the IoT in retail market, was valued at 66.44 billion dollars, and is projected to have a compound annual growth rate of 24.5% from 2025 to 2033. The recently increased amount of shipping endpoint systems needing data-driven management presents certain engineering challenges in mobile connectivity, hardware state management and security that cannot be addressed with customary reactive error handling methods.

The artificial intelligence acts as a layer above payment systems (which are usually out of scope) examining transaction patterns, hardware telemetry, and the workflow context. The AI itself does not make any decisions on whether a particular transaction should be allowed or denied but acts as an intelligence-seeking mechanism. As predictive maintenance in IoT environments has been reported to reduce maintenance planning by 20-50%, increase uptime by 10-20% and reduce total maintenance costs by 5-10%, the same benefits are extendable to terminal hardware management scenarios. This allows for the proactive optimization of terminal hardware maintenance workflows, while remaining isolated from cardholder data environments. These engineering challenges include non-deterministic hardware, state transitions across distributed, mobile IoT devices, retry approaches as a function of availability for mobile devices in transport, and enterprise-level observability frameworks [2]. We cover the technical challenges

Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ Research Article

and engineering solutions for what we refer to as an emerging class of AI-enabled mobile IoT shipping platforms in this article.

## Architectural Foundations of Mobile IoT Shipping Platforms

### Multi-Endpoint Integration Architecture

Mobile IoT shipping platforms build on decentralized systems where multiple, heterogeneous hardware endpoints collaborate to deliver a single transaction, in contrast to point-of-sale systems which are connected by a permanent wired infrastructure. As identified in the Internet of Ships architectural survey, there are dozens of device classes in large-scale IoT systems that have independent operational states and communication paths and thus independent latencies and unreliabilities of endpoints [3].

According to empirical evaluations on distributed IoT systems, cloud-based IoT processing models typically incur high communication overheads and unknown interactions between cloud service providers and evaluators [4]. Although application and evaluation of 5G mobile cellular network and delivery services for mobility in fog-based IoT environments show promising benefits and research opportunities in overcoming mobility-related challenges, they do not address mobilization and system management issues [4].

The aim of fog computing is to provide minimal latency by processing data near the network edge. It improves performance in real-time applications by up to 40% [4]. Edge computing can provide real-time data processing and can reduce data transmission time by 30% compared to cloud computing [4]. Yet obstacles such as energy use, security vulnerabilities, and the management of workflow across different architectures still exist [4].

### Real-Time Workflow Orchestration Challenges

The real-time workflow orchestration is done in mobile IoT shipping systems by asynchronously executing a series of transactions while ensuring concurrency and transactional consistency. IoT workflows executed only in the centralized cloud are highly latency-sensitive and also unstable under load, with latency increasing with more devices concurrently executing the tasks [4]. In edge computing, an optimized edge computing architecture for solving computation offloading problems of heterogeneous edge computing servers reduces the task delay by 50% and performs well in meeting deadlines [4].

For example, scanning a package, authorizing a payment, and printing a shipping label are all tied to specific hardware, but for various reasons (hardware and compliance), activities such as printing and secure payment authorization may need to happen in a specific order. The decentralization is a source of challenges such as security issues, contention over resources, and management complexity in fog systems [4].

### PCI Compliance and Security Domain Separation

Strict architectural constraints are imposed on mobile IoT shipping platforms by security and compliance requirements. For example, PCI DSS prescribes the separation of all payment processing components into their own security domains that must not be accessible from general-purpose devices. According to Aslam et al., security domain segmentation is a design principle in IoT systems with sensitive information and its absence increases the risk of data loss and modification considerably [3].

Their security analysis concludes that offloading all but sensitive processing to the fog and edge does not weaken security measures (when cryptographic isolation and access protection are implemented), and delivers quantifiable performance benefits [4]. The authors conclude that secure processing at the edge reduces latency without an observable increase in packet loss or error rates compared to baseline measurements [4]. While this requires more complex handling of errors and debugging, and hides payment internals, this separation is necessary to achieve regulatory responsiveness and clarity of architectural design.

### Hardware State Management and Synchronization

The shipping system relies on a consistent state across potentially distributed endpoints. Each endpoint has a state machine, handling its own readiness, activity, errors and recovery. Resource-intensive three-dimensional applications have been used as paradigmatic examples to evaluate response delay [4].
The source version runs only on the node where the central storage is located, and only copies the data items within the adjacent and close node. The edge version runs on each node where a copy of the data is kept. It can be disabled when no copies exist. It additionally supplies a messaging mechanism to find the replicas [4]. The experimental results indicate D-ReP performs better than non-replicated data sources with caching on the client side, realizing 26% more delay saving with a 14% lower incremental cost [4]. The communication inefficiency and misunderstanding errors resulting from the deployment and detection of the replica were also shown to be marginal [4]. The aforementioned approach has not been proved to be efficient in real time applications such as the emergency healthcare, or the self-driving cars [4].
In desynchronization scenarios, large-scale IoT systems with sporadically connected, non-homogeneous devices may encounter transaction aborts and user dissatisfaction [3]. These scenarios can be reduced by employing state estimation, timeout detection, and controlled reset strategies, which aim to provide a consistent system state without interrupting active operations or obstructing system functionality. Mechanisms must balance the frequency of verification against bandwidth and power constraints, especially in mobile environments where constant polling is harmful to performance.

| Architecture Type | Transmission Delay Reduction | Key Challenges |
|---|---|---|
| **Cloud-Centric** | NA | High communication cost, CSP interaction issues |
| **Fog Computing** | Up to 40% | Security gaps, resource contention, management complexity |
| **Edge Computing** | ~30% vs. cloud | High energy consumption, security vulnerabilities |

Table 1: Performance Comparison of Computing Architectures in Mobile IoT Systems [4]

## Payment Transaction Complexity in Distributed Mobile Environments

### Multi-Phase Payment Authorization Workflows

Payment authorization in distributed mobile IoT environments is more complex than at fixed point of sale due to the limited mobility of the devices, the heterogeneity of the network environments, the heterogeneity of software orchestrations. Mobile payments are an opportunity for financial and commercial organizations to increase revenue. The mobile payment market is expected to grow close to 38% from 2010 to 2020 [6].
Mobile payment authorization has multiple steps: terminal initialization, card interaction, secure data capture, payload transmission and encryption, authorization at the gateway, and device reset after a transaction has taken place [5], [6]. Each step involves an associated delay and failure condition. Empirical measurements of mobile payment end-to-end latencies show them to be less than a second when sufficient internet bandwidth is available, and a few seconds when a mobile user is on a slower data network, with the most variance in the latency accounted for by initialization and data transfers over the network [5].

The probability of failure is cumulative, and card interaction and transmission across a network have been shown to account for the majority of failed transactions in the case of NFC and chip based transactions [6]. Mobile transactions have a higher chance to fail (due to weak signal, roaming, power management) and to avoid wasting resources, orchestration layers should configure phase-aware timeout thresholds instead of simple end-to-end timers. Whereas naive retries can increase duplicate authorizations and overload backends, repeated retries taking the type of failure into account can improve the completion rate without exceeding payment processors' tolerances [6] or overloading backends.

**Payment Blob Integrity and Data Consistency**

Handling discrepancies and differences in data across multiple payment channels and gateways is another challenge. Payment orchestration platforms must be designed to reconcile transaction data from various sources, such as payment gateways, acquiring banks, and internal systems, to ensure data accuracy and consistency. Strong data validation and reconciliation processes should be in place for timely detection and resolution of discrepancies. Accurate algorithms for reconciliation that match and reconcile transactions are difficult to design [5]. The payment orchestration platform must form matching algorithms that are flexible to work with different formats, types of transactions, reconciliation algorithms, partial or duplicate matches, and reconciliation exceptions. The reconciliation service must also provide an audit trail of reconciliation operations [5]. Bit-level packet loss or serialization error at the time of an embedded message authentication code makes the authorization invalid.

As a result of wireless packet loss, mobile operating system level memory pressure, or cross-language data sharing between terminal firmware, mobile software, and backend infrastructure [5], mobile form factors are statistically more vulnerable to integrity threats. The majority of surveyed systems rely on end-to-end integrity validation schemes ranging from cryptographic checksums to embedded signatures, rooted in PCI isolation boundaries [6]. In particular, integrity validation is treated as an all-or-nothing correctness requirement, in that payloads are either correctly recovered or not recovered at all.

When a failure was detected, it was much more likely to be a major failure (>1% of requests). In trials, major failures were detected much more often than minor failures [7]. Such localization reduces mean time to resolution, and enables remediation that is easier to manage within a large mobile deployment.

**Timeout Management and Transaction Ambiguity**

From a design standpoint, timeout management is one of the hardest problems in mobile payment systems. According to measurements, the response time of the authorization varies largely, due to the processing time required by the issuer, network load, and the handshakes of the protocols [5]. Static timeout thresholds, while suitable for fixed environments, result in false timeouts or excessive user wait times in mobile environments.

While the payment transaction proceeds between the web server and client-side, compromising the access point, router, Wi-Fi monitoring attack, or cell tower, proxy and MBA (modem, bridge, and access point) may allow the SSL stripping attack to be performed on a man-in-the-middle (MITM) attack due to improper verification of SSL certificate [6]. The security and privacy of sensitive information is one of the main factors affecting the acceptance of mobile payment systems. Security is critical to ensuring transaction security properties, such as confidentiality, authentication, integrity, non-repudiation and authorization, between financial data passing between entities in an MPS such as the customer, bank server, payment gateway and merchant [7].

Further research also shows that mobile devices have been instrumental in mobile payments. The entire volume of mobile payments in 2017 was about $1.721 billion. [6] As mobile payments increase every day, the revenue of mobile payment systems also increases accordingly. Because of the stakeholders' support

**Journal of Information Systems Engineering and Management**
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

for the mobile/digital payments to every place and at any time, the enormous majority of mobile payment systems use Elliptic Curve Cryptography (ECC) to improve the efficiency of mobile payments. These schemes use lightweight cryptography, which is more secure and has a lower overhead [6].
Mobile shipping environments further complicate the ambiguity resolution problem, as it can happen that the connection is lost at the time the status is needed. This requires a trade-off between short time-out periods and the overhead of reconciling the status. Some work has researched adaptive time-out periods, based on network feedback [5].

**Hardware-Dependent Failure Patterns**
Hardware malfunctions are a common cause of failures on mobile payment terminals. Different types of failures were classified in empirical studies such as terminal wake-up failures, card reader failures, cryptographic key desynchronization, and battery-related failures [7]. Initialization failures are often caused by firmware modes that save battery power or a lack of battery, and card reading failures by environmental contaminants and wear [6].
The worst-case scenario is key management failures where keys are expired or de-synchronized. In such cases, no encryption can occur and all transactions deterministically fail until the keys are re-provisioned. Environmental factors such as extreme temperature, high humidity and vibration have been shown to result in considerably greater hardware error rates for mobile applications than fixed terminals indoors [5].
Mitigation involves maintaining hardware-specific failure taxonomies, gathering fine-grained diagnostic telemetry, and applying recovery strategies that are specialized for each class of failure rather than relying on generic retry logic. This has demonstrated efficacy in improving transaction success rates and minimizing downtime of distributed mobile payment services [6].

| Phase | Primary Function | Latency Contribution | Dominant Failure Types |
|---|---|---|---|
| **Terminal Initialization** | Device wake-up and setup | High variance | Power-saving state issues, battery depletion |
| **Card Interaction** | Physical/NFC card reading | Medium | Card reader malfunction, environmental contamination |
| **Secure Data Capture** | Cardholder data encryption | Low | Cryptographic key desynchronization |
| **Encrypted Payload Transmission** | Network data transfer | High variance | Wireless packet loss, network congestion |
| **Gateway Authorization** | Backend processing | Variable | Issuer delays, protocol handshakes |
| **Post-Transaction Reset** | Terminal state cleanup | Low | Firmware state management |

Table 2: Multi-Phase Payment Authorization Workflow Components

## Artificial Intelligence as Predictive Engineering Infrastructure

### Transaction Reliability Prediction Models

Reliability prediction models are an attempt to estimate, using quantitative methods, how much different types of components contribute to the failure of the system. It is predictive engineering. Common models for predicting reliability include Bellcore, CNET, HRD4, MIL-217, and Siemens reliability models. These models use component type and operating conditions to predict failures. A Bellcore-based empirical study

**Journal of Information Systems Engineering and Management**
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

found metal oxide resistors accounted for 13% of all failures found on the circuit board, and a CNET-based model predicted that 14% of failures were related to the tantalum electrolytic capacitors [8]. Conversely, the HRD4 model estimated a failure rate of just 6.5% for pn-junction diodes, though this may be as much a measure of device stress modeling as the devices themselves [8]. By contrast, tantalum electrolytic capacitors constituted as much as 30% of the failure rate as assessed by the MIL-217 estimate, while metal oxide resistors have an estimated failure rate of around 33% according to the Siemens model [8].
For mobile IoT transactional applications, supervised learning models can classify the outcome of each transaction based on an ordered sequence of operations, which contains information about hardware availability, the timing of communications, and the context in which the operations are executed. In heterogeneous systems with non-deterministic execution paths, learning-based reliability estimators yield important reductions in wrongly predicting transaction failures [9]. By returning a probabilistic reliability score instead of a binary success prediction, these models can enable pre-transaction decisions, such as conditional retries or delayed firing of a workflow. Given enough recent feedback data, periodic retraining of these models can reduce instabilities in prediction error and thus maintain stable prediction performance over time despite load or infrastructure changes [8].

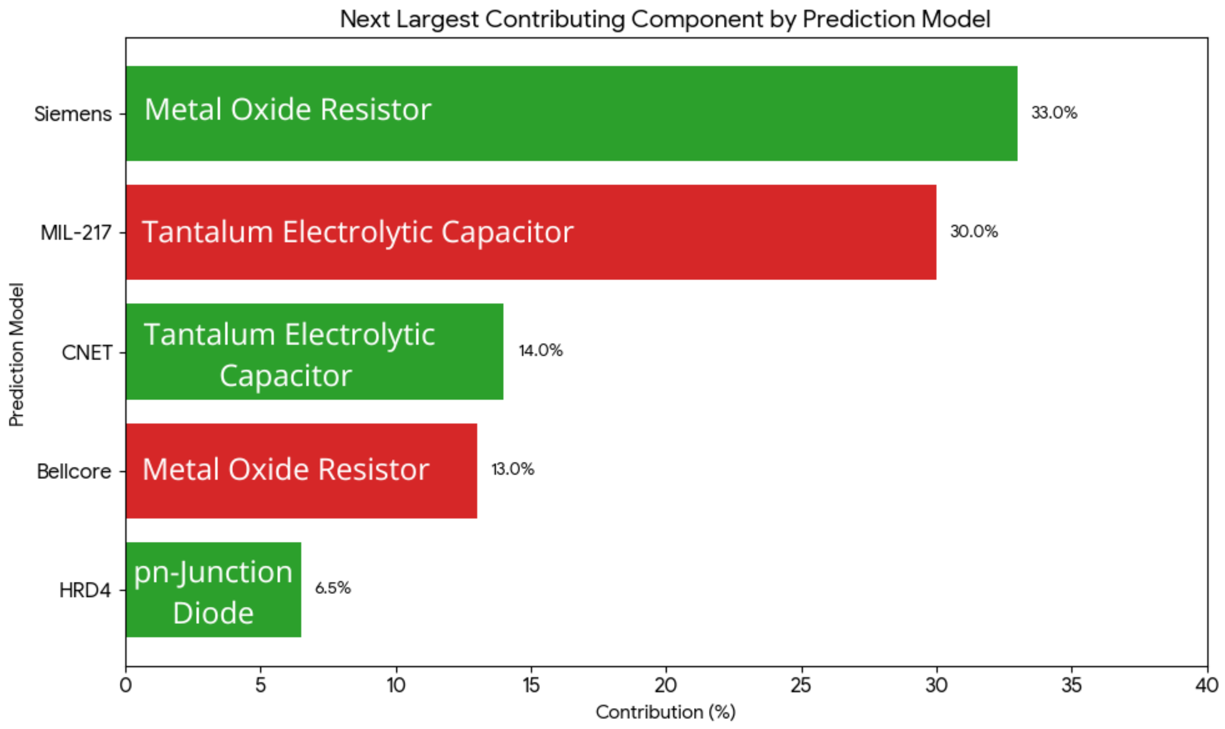

Figure 1: Next Largest Contributing Component by Prediction Model (after bipolar transistor) [8]

## Hardware Telemetry Analysis and Anomaly Detection

One example of this is telemetry-driven anomaly detection in predictive reliability engineering, where unsupervised anomaly detection can predict hardware failures before they happen. For example, real-world experiments on cyber-physical systems with multiple sensors found that 60-85% of device failures can be predicted with unsupervised anomaly detection depending on sampling frequency and the number of signals [10]. Time-series analysis of metrics, such as response time, error percentages and power drift, can allow for the detection of gradual deviations that can indicate terminal, sensor or actuator failure. Clustering techniques can be applied to assess groups of devices at the population level, for example, identifying groups of devices with similar degradation patterns [9]. Statistical control methods compare the operational profile of a system with a baseline profile to check for changes, as opposed to rules determined by expert operators under a predetermined threshold level. Telemetry methods allow predictive maintenance, which give considerably better results in downtime than reactive fault management [10].

### Workflow Context Integration and Optimization

To improve the accuracy of reliability prediction, hardware and network telemetry data, contextual information, and metadata at the workflow level are added as features to models. In evaluating the predictions from models from signal-only and context-aware approaches, the latter have been shown to reduce the probability of classifying transient anomalies as permanent ones (false positives) by 20% [9]. These context features encapsulate not only the device state, but also the order of execution and the environment, distinguishing between structurally complex workflows and system states that would result in unpredictable and unstable systems. This enables more precise fine tuning of control strategies, such as timeouts and retry gating. The robustness of context-aware models under different loads is also improved in distributed environments where hardware of the same type can have different levels of reliability depending on the context [8], showing that workflows can be more effectively predicted than uniform policy enactment in heterogeneous environments.

### Compliance-Preserving AI Architecture Design

If AI is used in regulated transactions, separation between predictive analytics and sensitive execution of the transaction is important. Reliability engineering literature suggests that optimal predictions are made on metadata and behavioral information instead of protected payload content, thus preserving regulatory compliance without sacrificing prediction quality [8]. Architectures where AI computation consists only of non-sensitive telemetry, timing during execution, and labels for the outcome will have equivalent predictive power as architectures trained on richer but still constrained data, and will ensure that learning pipelines and model inference logic are outside of regulated security zones. Validated architectures explained in the previous work show that scope-isolated AI components do not increase compliance exposure while allowing optimizations to be adapted to system behavior [10]. Such designs make artificial intelligence an engineering tool for compliance, not a regulatory compliance liability.

## Operational Challenges in High-Volume Distributed Environments

### Scale and Performance Requirements

Operational requirements for mobile IoT shipping platforms can be traced to transformations in distributed computing. A recent set of longitudinal studies shows distributed systems have transitioned away from client-server architectures that were common in the 1990s to multi-region microservice architectures that are federated across system borders. Today, approximately 94% of enterprise organizations have distributed computing infrastructures [11]. This has been accompanied by a 3-5× increase in system complexity per architectural generation, fundamentally altering the operational constraints within which scale and performance must be achieved.

Quantitatively, the modern distributed software stack of the average distributed application has grown from less than five in 1995 to 137 microservices in 3.8 geographic areas on average today [11]. For mobile IoT shipping platforms this architectural complexity means that payment authorization, inventory synchronization, labeling and telemetry flows run in combinatorially many possible configurations across

services and regions. For instance, longitudinal performance has improved around 250× per decade on average since 1990. End-to-end latency improved at the rate of 78% per decade on average between the same years, showing the changing expectations of modern systems [11].
Consequently, the expectations of transactional capabilities in environments with increasing shipping flows has risen. In 2023, it was estimated that a peak throughput of 12800 transactions per second would be achievable with sub-200ms response times, representing a 40x increase in throughput capabilities in the last decade. These TPS limits (320 in 1995, 1500 in 2000, 6400 in 2010 and 12800 in 2023) apply not only to backend processing, but all the other steps in the transaction lifecycle, including the round-trips of an authorization, device level processing and user interfaces [11]. In particular, during heavy periods of shipping when thousands of endpoints submit requests in parallel, small round-trip times can be noticeable at scale.
Through real-time IoT benchmarking, we can learn how workloads impact a system's scalability. The RIoTBench benchmark suite has shown that task complexity has a high impact on the throughput of distributed stream processing systems. Ingesting and processing hundreds to thousands of messages per second and increased latency under load have also been observed [12]. This reflects the situation in shipping platforms, where compute intensive workloads, such as encryption, validation or reconciliation, are more costly to run than lightweight telemetry ingestion workloads.

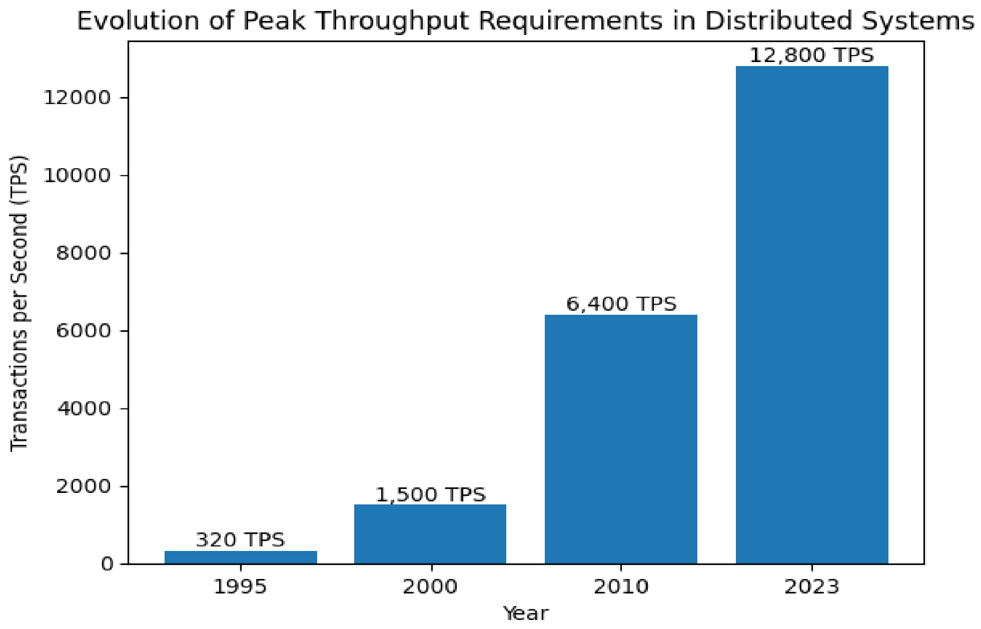

Figure 2: Growth in Peak Throughput Requirements of Distributed Systems [11]

**Network Connectivity Variability and Resilience**
As distributed systems are deployed across many regions and infrastructure providers, the network has become a major source of operational challenges. More and more enterprise applications are running across multiple regions and infrastructure providers, on average across 3.8 regions and 7.2 infrastructure providers, exposing them to heterogeneous network conditions and failure modes [11]. The challenge is even higher for mobile IoT shipping platforms across fixed retail locations, distribution centers and mobile field environments with varying connectivity.

**Journal of Information Systems Engineering and Management**
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/ **Research Article**

With network instability, correctness of the transaction and the latency of the payment authorization flows becomes a major concern, and systems that rely on synchronous interactions with an external network for authorization become vulnerable. Over the years, fault-tolerance has improved in distributed systems (up 84%), but mainly through retry, buffering, and isolation techniques rather than reliable connectivity assumptions [11]. Realistically this means the local queueing of transactions, the idempotent design of the transaction, and adaptive retries to handle transient failures without double charging.
Finally, RIoTBench experiments show that variations in the network and underlying resource can lead to important differences in the end-to-end latency experienced by users, even when the average throughput remains constant [12]. This suggests that adaptive communication mechanisms like adaptive protocol selection, batching and payload compression may be needed on mobile IoT platforms to achieve consistent user-perceived quality of service.

### Hardware Lifecycle Management and Diversity

The accelerating evolution of distributed computing architectures has also led to an increase in hardware heterogeneity. Enterprise systems might scale to thousands of inter-connected components, while mobile IoT must support multiple generations of devices, multiple vendors, and multiple firmware and software baselines. This is part of a larger trend in the field towards architectural heterogeneity, for increasing complexity of features [11].
Studies of distributed system evolution show the increasing failure surface area of larger distributed systems, and support the need for lifecycle management rather than total system replacement as physical components fail [11]. In mobile shipping systems, spatially distributed hardware failures are not serviced in real-time, making predictive maintenance and device telemetry essential. These include issues such as tracking inventory, assessing firmware compatibility, and detecting drift from desired configuration when fleets may number in the tens of thousands.

### Observability and Diagnostic Infrastructure

Increasing scale and architectural complexity have further required observability. As distributed systems have scaled from a handful of components to hundreds of microservices, customary monitoring approaches have been insufficient to provide insights into performance or reliability problems [11]. This drives the need for high-scale telemetry pipelines to collect logs, metrics and traces from every component of the system.
As systems become more complex, the amount of data being collected continues to grow. As reported by RIoTBench, real-time IoT systems can produce millions of events per day which calls for well-designed stream-processing pipelines to support both high throughput and low latency analysis [12]. Distributed tracing, which allows for reconstructing the trajectory of a transaction as it traverses multiple services and regions, has become a key diagnostic tool in these settings.
From the perspective of operations, observability systems should strike a balance between diagnostic fidelity and compliance. Payment-enabled IoT shipping platforms must ensure that telemetry streams do not leak sensitive cardholder data while providing sufficient observability to identify performance bottlenecks, network degradation, and device-level problems. As observability features are important for distributed systems and enable the reliability needs of modern software to be met, workloads have high uptime targets, with 78% of enterprises targeting 99.99% and 23% targeting 99.999% uptime [11].

## Conclusion

Building an artificial intelligence (AI) based PCI compliant mobile Internet of things (IoT) shipping system is a different challenge from basic payment processing and app development. This article summarizes how transformational AI can be applied as a predictive engineering layer beyond payment processing systems. By analyzing observed transaction patterns, hardware telemetry, and the context of the workflows, AI can optimize the system proactively, without violating compliance boundaries. This exemplifies that clever automation and compliance can go hand in hand through considerate system design with security boundaries and AI leverage. Consequently, this work yields higher transaction success rates, improved operational efficiency and greater security assurances than customary architectures that operate reactively. Given the increasing reliance of enterprise logistics on mobile shipping, the responsive systems engineering approach described in this work further advances distributed mobile computing, IoT orchestration systems and AI-based operational platforms.

## References

[1] Grand View Research, "Internet Of Things In Retail Market (2025 - 2033)," 2024. Available: https://www.grandviewresearch.com/industry-analysis/internet-of-things-iot-retail-market
[2] Deloitte Insights, "Predictive maintenance and the smart factory," Deloitte University Press, 2017. Available: https://www2.deloitte.com/us/en/insights/focus/industry-4-0/using-predictive-technologies-for-asset-maintenance.html
[3] Sheraz Aslam et al., “Internet of Ships: A Survey on Architectures, Emerging Applications, and Challenges,” IEEE Internet of Things Journal, 2020. Available: https://www.researchgate.net/profile/Sheraz-Aslam/publication/380004315
[4] Shams Forruque Ahmed et al., “Optimising Internet of Things (IoT) Performance Through Cloud, Fog and Edge Computing Architecture,” IET Wireless Sensor Systems, 2025. Available: https://ietresearch.onlinelibrary.wiley.com/doi/pdfdirect/10.1049/wss2.70016
[5] Shobhit Agarwal, “Payment Orchestration Platforms: Achieving Streamlined Multi-Channel Payment Integrations and Addressing Technical Challenges,” Quarterly Journal of Emerging Technologies and Innovations, 2019. Available: https://www.researchgate.net/profile/Shobhit-Agrawal-7/publication/380357700
[6] Sriramulu Bojjagani et al., “Systematic survey of mobile payments, protocols, and security infrastructure,” Journal of Ambient Intelligence and Humanized Computing, 2021. Available: https://www.researchgate.net/profile/Chien-Ming-Chen-2/publication/351690030
[7] Emre Kıcıman and Armando Fox, “Detecting application-level failures in component-based internet services,” IEEE transactions on neural networks, 2005. Available: https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/anomaly-4.pdf
[8] J.A.Jones and J.A.Hayes, “A comparison of electronic-reliability prediction models," IEEE Transactions on reliability, 1999. Available: https://www.nomtbf.com/wp-content/uploads/2011/12/Draft-Comparison-of-Electronic-Reliability-Prediction-Methodologies.pdf
[9] Ziji Chen et al., “Detecting Anomalies in Systems for AI Using Hardware Telemetry,” arXiv preprint arXiv, 2025. Available: https://arxiv.org/pdf/2510.26008
[10] Nikhil Kassetty et al., “Billing Workflows: Harnessing Edge AI to Solve Operational Bottlenecks and Enhance Efficiency,” International Journal of Global Innovations and Solutions, 2024. Available: https://ijgis.pubpub.org/pub/gbeleiu7
[11] Aravind Sekar, “The Future of Large-Scale Distributed Systems: Trends, Challenges, and Opportunities,” International Journal of Scientific Research in Computer Science Engineering and Information Technology, 2025. Available: https://www.researchgate.net/publication/390714723
[12] Hany Ali et al., “Dynamic connectivity hub: Multiple isps smart aggregation for optimized iot connectivity," IEEE Access, 2025. Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=10840185