ISSN: 2096-3246
Volume 58, Issue 01, 2026

ENTERPRISE ARCHITECTURE FOR PAYMENT CARD INTEGRATION: PAYMENT ORCHESTRATION, EVENT-DRIVEN PIPELINES, AND REGULATORY COMPLIANCE IN COMPLEX SAP ENVIRONMENTS

Rajasekhar Reddy Putta
Pondicherry University, India.

Abstract
Modernizing the enterprise payments infrastructure requires a consistent architecture across heterogeneous ERPs, CRMs, and commerce and retail stacks. Converging regulatory compliance, fraud, and omnichannel use cases compound the technical challenge that legacy point-to-point integrations cannot sustainably solve. A central Payment Orchestration Layer (POL) based on security-first tokenization principles can eliminate cardholder data (CHD) at all layers within the enterprise application stack: hosted payment fields, iFrame-based card-not-present flows, and PCI-validated Point-to-Point Encryption (P2PE) in retail card-present environments. Apache Kafka's exactly-once semantics can enforce financial integrity within automated distributed settlement and posting processes by atomically pairing payment lifecycle events with relevant financial accounting commands. Network tokenization per EMVCo Payment Tokenisation Technical Framework maximizes approvals while reducing exposure to fraud with domain-restricted, cryptographically bound surrogates. Mutual TLS with certificate-bound access tokens per IETF RFC 8705 limits the attack surface across use cases for all integration boundaries. SCA orchestration as per EMV 3-D Secure 2.2.0 allows for optimized, frictionless transactions to remain within the PSD2 regulatory envelope. SAP Cloud Integration is the connectivity mechanism through which FI-AR and GL postings are then automatically triggered by payment events, without CHD traversing enterprise systems. This eliminates paper invoice costs, increases transaction realization rates, and improves receivables visibility across all channels.

Keywords: Payment Orchestration Layer, Apache Kafka Exactly-Once Semantics, PCI DSS Tokenization, Strong Customer Authentication, and EMVCO Network Tokens.

1. Introduction

Enterprise payment infrastructure modernization is one of the most architecturally challenging parts of large-scale digital transformation initiatives in organizations with heterogeneous enterprise resource planning (ERP), customer relationship management (CRM), commerce, and retail environments. They must balance the need for consistent payment capabilities, regulatory compliance, fraud and risk mitigation, and operations for highly dynamic omnichannel commerce. Point-to-point integrations between enterprise systems and third-party payment processors have proven fragile, widened the regulatory compliance surface area, and constrained the speed of configuration for new payment methods and fraud management controls. The concept of payment orchestration, event-driven processing such as Apache Kafka, and network tokenization have changed architectural considerations [1][2].

This paper describes a payment card integration enterprise architecture. It combines technical architecture, regulation, and finance, where security-first tokenization eliminates cardholder data (CHD) from all enterprise applications; a platform-centric Payment Orchestration Layer (POL) centralizes the routing, tokenization, and settlement; event-driven microservices decouple the channels from the SAP core systems; and omnichannel parity is achieved across the web, call center, B2B, and retail store environments [3][4]. The remainder of this article covers each of these aspects

in turn, covering the technology stack required to offer exactly-once financial transactions, PCI DSS compliance across multiple business channels, real-time fraud management in line with PSD2 regulatory requirements, and the automated reconciliation of transaction settlement into financial accounting systems.

2. Payment Orchestration Layer: Architecture and Security Posture

POLs are the only logical boundary for all payment traffic within an enterprise. They are responsible for payment authorizations, captures, voids, refunds, multi-acquirer routing, token vault management, 3DS orchestration, settlement ingestion, payment disputes, and end-to-end observability. With these functions centralized in a strongly typed REST/OpenAPI interface using signed, idempotent, and replayable webhooks, POL decouples its consuming enterprise applications (SAP ECC, SAP S/4HANA, SAP Commerce, CRM, and retail Point of Sales systems) from the operational burden of integrating multiple payment service providers (PSPs) [5].

Protection of the POL relies on mutual TLS and certificate-bound access tokens, as defined in IETF RFC 8705 [6]. This prevents bearer token replay attacks by binding the token to the client certificate being presented. If the bearer token is stolen, it cannot be used without the corresponding client private key to unlock it. Finally, key management operations are handled by a Hardware Security Module (HSM), with role-based access control (RBAC), a Web Application Firewall (WAF), DDoS protection, and full audit logging, providing additional layers of protection.

To connect to SAP, SAP Cloud Integration (CPI) is used with REST/OData adapters, mappable data transformation and routing templates, OAuth2 support, and API management capabilities [7]. Events occurring in the payment lifecycle are published to Apache Kafka topics, which are consumed by CPI integration flows (iFlows) to trigger updates to FI-AR and GL. No CHD is ever made to traverse the SAP systems. The POL also exposes a policy configurator that centralizes and distributes routing rules, SCA exemptions, Pay-by-Link expiry windows, and retry policies via configuration topics. This allows rapid reconfiguration of payment methods and fraud rules without deploying code to the channel application [3].

Dispute handling	Chargeback event processing	Signed, idempotent webhooks
Policy configuration	Routing rules, fraud rules, and retry policies	Centrally distributed config topics
SAP connectivity	FI-AR/GL updates via CPI iFlows	OAuth2 with REST/OData adapters

Table 1: POL Functional Capabilities and Security Controls [5][6][7]

3. Apache Kafka Event-Driven Architecture and Exactly-Once Semantics

The reliability and monetization of payments between multiple enterprise systems depend on the messaging substrate. Payments require delivery guarantees, event ordering guarantees, idempotency and atomicity of financial postings, and the mutually consistent handling of order state transitions with settlements of those transitions in the face of system failure on one part or the other. These requirements were addressed with idempotent producers and transactions in Apache Kafka version 0.11 that achieve exactly-once processing semantics (EOS) for the read-process-write loop [8].

This architecture defines a payment topic topology that covers the entire payment life cycle. The topics are payment.authorization.request|response and payment.sca.challenge.started|completed, payment.capture.request|response, payment.refund.request|response, payment.settlement.update,

payment.chargeback.event, payment.token.lifecycle, payment.paybylink.created|delivered|paid|expired, payment.terminal.session.opened|closed, fiar.posting.request|posted [8]. It thereby decouples (isolates) channels, POL, and financial systems from each other, allowing them to evolve independently of each other.

In this mode, the producer configuration will include enable.idempotence=true, acks=all, bounded in-flight requests. A transactional producer will specify a transactional.id and write records atomically via beginTransaction/commitTransaction. This guarantees that whenever there is a transaction (e.g., a capture response), there is a matching FI-AR posting request, and thus the financial ledger is consistent regardless of how many times a transaction may be retried. Setting the consumers to isolation.level=read_committed hide aborted transactions [8] and implementing the financial exactly-once discipline and thus preventing double captures or double postings in the event of network partitions or application restarts.

On an implementation level, the architecture implements per-topic dead-letter queues (DLQs); idempotent consumers (using idempotency keys for deduplication); partitioning by order or payment identifier to co-locate affinity groups; and observability instrumentation for consumer lag, throughput, and end-to-end latency by working backward from the observability data to infer latencies of processing stages. These guardrails ensure that settlement and collections processes are automated, journal postings are automated, and finance is kept aware of incoming receivables.

4. PCI DSS Compliance, CHD Elimination, and Tokenization Strategy

The Payment Card Industry Data Security Standard v4.0.1 defines the scope of PCI as all systems that store, process, or transmit CHD [1]. The architectural approach for the segmentation of CHD from all enterprise systems is to use hosted payment fields and iFrames for card-not-present (CNP) transactions and PCI-validated Point-to-Point Encryption (P2PE) for card-present transactions. CNP transactions insert cardholder data into fields hosted by the PSP, which are rendered inside the merchant's application context but are processed inside the PSP's domain. This means that PAN and SAD never traverse the enterprise network infrastructure [1][9]. For example, CRM agents taking orders via the telephone will only see and use PSP-hosted fields and will never hear or see PAN or SAD.

Two new PCI DSS v4 requirements have a broad impact on e-commerce implementations. Requirement 6.4.3 requires a merchant to maintain an inventory of all scripts on payment webpages that contain an authorization and/or integrity-checking process. Requirement 11.6.1 requires merchants to implement a permanent client-side change and tamper detection process for payment and parent webpages. [1]. These requirements affect the SAQ option between SAQ An and SAQ A-EP for embedded or redirect payment pages. We enforce these requirements in the architecture through SRI, CSP script allow-listing (previously known as CSP), and tooling that monitors the web application stack [1].

For an integrated retail and POS application, PCI-validated P2PE solutions are selected from the official PCI Security Standards Council registry [9]. In a semi-integrated POS technology architecture, the payment terminal is directly connected to the PSP environment, with card data encrypted at the point of swipe, dip, or tap. Since decryption occurs only in the PSP's secure environment, the CHD never traverses the store's network or resides in the point of sale application memory [9][10]. The merchant obligations as outlined in the P2PE Instruction Manual (PIM), including chain-of-custody, device tampering, and key management, are part of store procedures [9].

Tokenization uses three types of tokens: enterprise tokens, which provide a common reference to a payment instrument across SAP ECC, SAP CRM, SAP S/4HANA, SAP Commerce, and SAP Point

of Sale. Acquirer tokens are for processor-scope requirements, while network tokens conform to the EMVCo Payment Tokenization Technical Framework [11] to replace PAN with a domain-restricted cryptographically bound surrogate limited to a specific merchant device or transaction context. Network tokenization is offered by Token Service Providers (TSPs) that provide token generation, binding, account updater refresh, token rotation, and token revocation [12]. The cryptographic features of the network tokens can improve authorization rates and reduce fraud, resulting in the successful completion of the transaction. Token lifecycle governance is integrated with the CRM consent hierarchy and SAP account hierarchy to accommodate both B2C card-on-file and B2B account-on-file use cases.

Channel	CHD Elimination Mechanism	Applicable SAQ	Key v4.0.1 Controls
E-commerce (CNP)	PSP-hosted iFrame / hosted fields	SAQ A or SAQ A-EP	Req. 6.4.3 script inventory; Req. 11.6.1 Tamper Detection
CRM / call center	POL-hosted embedded UI	SAQ A or SAQ A-EP	SRI enforcement; CSP script allow-listing
Retail POS	PCI-validated P2PE; semi-integrated terminal	SAQ P2PE	PIM chain-of-custody; device tamper checks
Mobile POS	Encrypted card readers; SDK integration	SAQ P2PE	Key rotation via TMS; PIM procedures
B2B / partner API	Token-only references; no PAN transit	SAQ A	Idempotency keys; mTLS API boundary

Table 3: PCI DSS v4.0.1 Scope Controls by Channel [1][2][3][9]

5. Fraud Management, SCA/PSD2 Orchestration, and Pay-by-Link

The legal baseline for fraud management across European countries is the Revised Payment Services Directive (PSD2) and its related RTS [13][14]. The RTS imposes the need for applying SCA to remote electronic payment transactions, which is a combination of at least two of something you know, have, and are and dynamic linking. As such, the RTS defines the various named exemptions (Transaction Risk Analysis (TRA), low-value transactions, MITs, and corporate payment instruments) through which frictionless processing may be preserved while maintaining regulatory compliance [13].

The 3DS 2.2.0 (3DS 2.x) version is the global standard wireless specification for 3DS 2 authentication flows by EMVCo. In addition to clarifying 3DS Requestor-Initiated (3RI) flows for MIT and decoupled authentication for out-of-band, the specification added important enhancements to the available data elements for 3DS 2 frictionless challenge suppression. A 3DS orchestration engine in the POL checks the transaction against the jurisdiction exemption hierarchy and alerts the issuer of the request using standardized exemption indicators, and sparks challenge flows only if the issuing bank declines the frictionless flow. Layered PSP fraud prevention tools also add more protection and do not add friction to the checkout experience. These features include device fingerprinting, behavioral analytics, BIN intelligence, velocity and anomaly detection models, negative and allow-list management, and champion-challenger rule testing. [3]

With the Pay-by-Link feature, PSPs can send a secure, on-brand payment link via email or chat that can be viewed on any device [16]. From a PCI perspective, CHD is not routed through any enterprise infrastructure. The PCI SAQ A/A-EP controls apply to merchant-hosted initiating pages, including

script inventory, SRI enforcement, and tamper detection. These requirements are intended to provide security against client-side skimming [1][16]. In reality, the payment link is typically generated by the agent using the tap-to-pay bot in a call/communication with the customer and committed back to the customer by the Kafka topics, allowing the status of the link during the lifetime of the invoice (created, delivered, paid, expired) and then triggering the related order fulfillment events and FI-AR status updates, bringing savings on paper invoices and shortening the cash collection cycle.

The BRIM (Billing and Revenue Innovation Management) module of SAP S/4HANA also provides specific support for subscription-based or consumption-based recurring billing models. An implementation of BRIM requires that the payment authorization support incremental authorization (the update of the remaining authorized amount) and reauthorization in cases where delivery is delayed until the original authorization expires. Changes in authorization states are replicated in Kafka to finance and order management systems to ensure that the SAP-managed credit exposure reflects the actual payment risk.

Token Type	Scope	Lifecycle Operations	Enterprise Use Case
Enterprise token	Cross-channel (ECC, CRM, S/4HANA, Commerce, POS)	Create, bind, rotate, revoke	Consistent instrument reference across SAP
Acquirer token	Processor-scoped storage	Create, refresh, expire	PSP-specific transaction processing
Network token	Domain-restricted (merchant/device/context)	Create, account updater refresh, rotate, revoke	Improved auth rates; reduced fraud

Table 4: EMVCo Tokenization: Token Types, Scope, and Lifecycle Operations [11][12]

6. Settlement, FI-AR Reconciliation, and Conclusion

The settlement and reconciliation architecture closes the loop from payment execution to financial accounting. Acquirer settlement files and PSP settlement events are normalized into the POL and sent to Kafka topics, which SAP ECC and S/4HANA can consume from CPI iFlows [7][8]. These iFlows post to FI-AR and GL and process the fee accruals, chargeback provisioning, and variance reporting. The transactional producer and read_committed consumer isolation levels of Kafka ensure that the event triggering the iFlows and the posting commands are atomically processed, and the financial ledger is never out of step with the payment outcomes, even under failure and retries.

Settlement references are linked to FI-CA (Contract Accounts Receivable and Payable) in S/4HANA BRIM implementations to allow for automatic receivables clearing of high volume [17]. The BRIM components SOM (Subscription Order Management), CI (Convergent Invoicing), and CC (Convergent Charging) support the end-to-end usage-to-cash process. Kafka is an event backbone that handles charge events from charging and rating to invoicing and payment collection.

The architecture described in this article is cohesive because the hosted fields and the PCI-validated P2PE solution deliver the design advantages of removing CHD from all enterprise application layers and, therefore, enabling scope reduction for PCI DSS v4.0.1 and for the smoothest QSA audit

assessment cycle [1][9]. EMVCo's network tokenization specification improves authorization rates and provides lifecycle management and cryptographic domain restrictions [11][12], while Apache Kafka's exactly-once semantics and transactional API guarantee financial consistency across distributed and concurrent settlement and posting workflows [8]. mTLS with certificate-bound tokens per RFC 8705 constrains the API attack surface [6]. SAP CPI provides the integration fabric of payment events, driving a financial accounting backend without ever having CHD transit SAP systems [7]. PSD2/SCA orchestration via EMV 3DS 2.2.0 optimizes for frictionless authorization and sustains compliance with current regulations [13][15].

The result is an enterprise payment platform to manage multiple brands, regulations, and channels from a single orchestrating layer. Improved automated journal postings, visibility into receivables, reduced paper invoice printing, and improved transaction realization rates make for an attractive architectural investment for large payments businesses.

Exemption Type	Regulatory Basis	3DS 2.2.0 Signal	Frictionless Outcome
Transaction Risk Analysis (TRA)	PSD2 RTS Article 18	Exemption indicator in auth request	The issuer approves without challenge
Low-value transaction	PSD2 RTS Article 16	Low-value flag	No SCA required below threshold
Merchant-Initiated Transaction	PSD2 RTS Article 13	3RI flow	No customer interaction required
Corporate payment instrument	PSD2 RTS Article 17	Corporate flag	Exempt from SCA mandate
Decoupled authentication	EMV 3DS 2.2.0 extension	Decoupled auth indicator	Out-of-band verification, no redirect

Table 5: PSD2/SCA Exemption Hierarchy and 3DS 2.2.0 Mechanism [13][14][15]

Conclusion

Applicable to all company types, this article's architecture eliminates CHD from all layers of all enterprise applications by combining PSP-hosted fields, PCI-validated P2PE terminals, and EMVCo standards-compliant centralized token vaults to satisfy PCI DSS v4.0.1 scope reduction requirements and close all QSA audit cycles for all business channels. An Apache Kafka transactional producer and read_committed isolation level consumer create a distributed settlement, and FI-AR posting financial exactly-once discipline, meeting a financial ledger audit requirement to remain in sync with the payment result even when the payment fails, retries, or is sent across multiple partitions of the financial ledger. POL's integrated policy configuration means that changes to payment methods, fraud rules, SCA exemptions, and routing rules can be made rapidly and without coding each change into each of the channel applications. The mTLS with certificate-bound tokens, as defined by RFC 8705, allows strict mutual authentication at the edge of every API. PSD2/SCA orchestration is optimized through the use of EMV 3DS 2.2.0, which supports exemption signaling. Thanks to Pay-by-Link, secure payment collection in email and chat without CHD is also possible. The settlement events are sent as normalized FI-AR, GL, and FI-CA journal entries via SAP CPI integration flows, eliminating the transit of CHD. This provides deeper visibility into receivables across transactional non-CHD ECC environments as well as high-volume BRIM subscription billing deployments, resulting in a single orchestration boundary for an enterprise payment platform servicing multiple card brands, regional regulatory regimes, and channel types, and providing the financial operational outcomes that justify the architectural investment.

ISSN: 2096-3246
Volume 58, Issue 01, 2026

References

[1] PCI Security Standards Council, "Payment Card Industry Data Security Standard," 2024. [Online]. Available: https://www.middlebury.edu/sites/default/files/2025-01/PCI-DSS-v4_0_1.pdf?fv=AKHVQBp6

[2] Securitymetrics, "How to Perform a PCI v4.0 SAQ A Self-Assessment," *PCI SSC Document Library*, 2022. [Online]. Available: https://www.securitymetrics.com/blog/how-to-perform-a-pci-40-saq-a-self-assessment#

[3] PCI Security Standards Council, "Payment Card Industry (PCI) Data Security Standard Self-Assessment Questionnaire A-EP and Attestation of Compliance," PCI SSC Document Library, 2022. [Online]. Available: https://listings.pcisecuritystandards.org/documents/SAQ_A-EP_v3.pdf

[4] PCI Security Standards Council, "Information Supplement: Guidance for PCI DSS Scoping and Network Segmentation," PCI SSC Document Library, 2016. [Online]. Available: https://listings.pcisecuritystandards.org/documents/Guidance-PCI-DSS-Scoping-and-Segmentation_v1.pdf

[5] Global Payments, "Pay by Link Overview," [Online]. Available: https://developer.globalpayments.com/docs/payments/online/pay-by-link-overview#:~:text=Allow%20customers%20to%20pay%20with,to%20enter%20their%20payment%20details.

[6] B. Campbell, J. Bradley, N. Sakimura, and T. Lodderstedt, "OAuth 2.0 Mutual-TLS Client Authentication and Certificate-Bound Access Tokens," IETF RFC 8705, Feb. 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8705

[7] SAP SE, "Exploring the Capabilities of SAP Integration Suite," *SAP Help Portal*, 2024. [Online]. Available: https://learning.sap.com/learning-journeys/developing-with-sap-integration-suite/exploring-the-capabilities-of-sap-integration-suite

[8] Confluent, "Exactly-Once Semantics Are Possible: Here's How Kafka Does It," *Confluent Engineering Blog*, Jun. 2017. [Online]. Available: https://www.confluent.io/blog/exactly-once-semantics-are-possible-heres-how-apache-kafka-does-it/

[9] PCI Security Standards Council, "List Of Validated Products And Solutions," [Online]. Available: https://listings.pcisecuritystandards.org/assessors_and_solutions/vpa_agreement?return=%2Fassessors_and_solutions%2Fpayment_software

[10] Ingenico, "Ingenico MOVE 5000 Terminal," Fiserv, 2023. [Online]. Available: https://merchants.fiserv.com/content/dam/firstdata/au/en/documents/Ingenico%20Terminal%20Move5000%20User%20Guide.pdf

[11] EMVCo, "EMV® Payment Tokenization," *EMVCo*, 2023. [Online]. Available: https://www.emvco.com/wp-content/uploads/2023/03/EMVCo-Payment-Tokenisation-A-Guide-To-Use-Cases-v2.2.1.pdf

[12] Mastercard, "Tokenization explained: Protecting sensitive data and strengthening every transaction," 2026. [Online]. Available: https://www.mastercard.com/global/en/news-and-trends/stories/2025/what-is-tokenization.html#:~:text=Tokenization%20protects%20your%20account%20by,represent%20digital%20or%20physical%20assets.

[13] legislation.gov.uk, "Commission Delegated Regulation (EU) 2018/389," https://www.legislation.gov.uk/eur/2018/389

[14] European Banking Authority, "Opinion on the Implementation of the RTS on SCA and CSC,"

EBA, 2018. [Online]. Available: https://docs.google.com/document/d/1lv5m5_AY4BzjUDxrLJT_DG_zoLQ4VP4lkhLvQRC_z9Q/edit?tab=t.0
[15] Visa Europe, "European EMV 3DS 2.2.0 Implementation Guide," 2019. [Online]. Available: https://www.visa.co.uk/content/dam/VCOM/regional/ve/unitedkingdom/PDF/sca/visa-european-emv-3ds-220-implementation-guide.pdf
[16] Cashfree Payments, "Create Payment Link," 2024. [Online]. Available: https://www.cashfree.com/docs/api-reference/payments/latest/payment-links/create
[17] SAP Support, "SAP Billing and Revenue Innovation Management" 2024. [Online]. Available: https://support.sap.com/en/product/onboarding-resource-center/brim.html
[18] SAP, "SAP Commerce Cloud, Open Payment Framework Integration," *SAP Help Portal*, 2024. [Online]. Available: https://help.sap.com/docs/SAP_COMMERCE_COMPOSABLE_STOREFRONT/962112809f9a48f9b36aa05b208b3731/5b69af54a66645f7af8cc541b0ddbe08.html
[19] Nawaz Dhandala, "How to Create Kafka Idempotent Producers," OneUptime, 2024. [Online]. Available: https://oneuptime.com/blog/post/2026-01-30-kafka-idempotent-producers/view
[20] SAP Community, "Expose SAP CPI Integration Flows as APIs Using SAP API Management: A Step-by-Step Guide," SAP Help Portal [Online]. Available: https://community.sap.com/t5/technology-blog-posts-by-members/expose-sap-cpi-integration-flows-as-apis-using-sap-api-management-a-step-by/ba-p/14256902