Launching a White-Label Payment Service Provider on ACM: A Representative Scenario in Multi-Processor Payment Orchestration

ACM Global Tech

Abstract

This case study is a *representative modernization scenario*. It describes how a fintech or financial institution launches a branded, "Stripe-class" Payment Service Provider (PSP) on ACM Global Tech's white-label, regulated-first ecosystem: multi-processor orchestration, durable payment workflows, intelligent routing across processors, and card, bank, and crypto rails, all under the institution's own brand and post-quantum-secured. The technical substance traces to the multi-processor payment-orchestration architecture of *Hanzo Commerce* [1], delivered through ACM's PSP; the platform composes with ACM's MPC custody and non-custodial settlement [2], post-quantum digital-securities infrastructure [3], and FHE privacy [4] on the Lux, Hanzo, and Zen stack. All outcomes are *target* outcomes framed against cited public benchmarks.

1 The Pressure

A community-scale or mid-sized institution already holds the relationships, the licenses, and the deposit base to run merchant payments. What it typically does not have is its own PSP, so the most valuable, recurring part of the payment relationship runs on a third party's rails and under a third party's brand. The pressure is structural rather than incidental:

- **Processor lock-in.** Integrating directly against a single processor means inheriting that processor's API, error semantics, and settlement timeline, with no path to cross-processor failover and no leverage on price [1].
- **Fragmented rails.** Acquiring, the gateway, payouts, and any crypto channel run on disconnected systems, each adding a handoff, a distinct settlement file, and a place for reconciliation to drift.
- **PCI and compliance burden.** Handling card data pulls the institution into the full scope of PCI DSS, while a regulated launch must also satisfy SOC 2 and ISO 27001 governance from day one.
- **Long-lived data exposure.** Cardholder and merchant records are a "harvest now, decrypt later" target: data intercepted today can be decrypted once a cryptographically relevant quantum computer exists, a risk the institution carries without controlling the cryptography that protects it.

The objective is to convert a referral arrangement into a payments line of business the institution owns, brands, and controls, without taking on the reliability or compliance debt that fragmented integration implies.

2 ACM's Approach: the White-Label PSP

ACM ships a Stripe-class PSP and a full Banking-as-a-Service product, licensed so the institution runs acquiring, orchestration, and payouts under its own brand. Rather than negotiate a better referral deal, the institution stands up the entire payments stack on one platform, and the same Banking-as-a-Service rails extend to its other programs. Four capabilities, drawn from the orchestration architecture in [1], carry the work.

Processor-agnostic payment intents.
Applications submit a *payment intent*, a processor-independent description of the desired charge carrying amount, currency, method, customer, an explicit candidate processor list, and an idempotency key; the orchestration layer, not the application, selects the processor and manages the lifecycle [1]. The idempotency key guarantees that a retried request returns the original result rather than creating a duplicate charge [1].

Multi-processor abstraction across card, bank, and crypto rails.
Each processor is reached through one common interface (`Charge`, `Capture`, `Refund`, `Webhook`), and processor-specific error codes are mapped to a canonical set (`declined`, `insufficient_funds`, `expired_card`, `processor_error`, `fraud_suspected`) so application code stays processor-agnostic [1]. The same abstraction spans card processors, alternative methods, and on-chain crypto channels, the latter settled by a watch-and-confirm pattern (a unique deposit address, blockchain monitoring, confirmation after the required block depth) [1].

Durable payment workflows.
The payment lifecycle runs as a durable workflow with compensation: a charge is followed by a ledger entry, and if a later step fails the workflow automatically refunds, with durable execution guaranteeing that compensation runs even if a worker process crashes [1]. This is the saga pattern [5] applied to payments, yielding exactly-once processing semantics under partial failure [1].

Intelligent routing across processors.
A scoring model selects the optimal processor and drives automatic failover, balancing success rate, latency, and cost [1]:

$$\text{score}(p) = \alpha\, r_p + \beta \left(1 - \frac{l_p}{l_{\max}}\right) + \gamma \left(1 - \frac{c_p}{c_{\max}}\right), \tag{1}$$

where r_p is the success rate, l_p the median latency, and c_p the per-transaction cost for processor p, with default weights $(\alpha, \beta, \gamma) = (0.6, 0.2, 0.2)$. When the primary processor fails with a retryable error, the router tries the next-best processor up to the configured failover limit [1].

Real-time AI/ML for AML. ACM operates a real-time AI/ML anti-money-laundering and transaction-monitoring engine on top of the PSP, a live capability that scores every transaction as it flows through the orchestration layer. It composes with the processor-native and rule-based fraud controls (velocity checks, amount limits, geographic anomaly detection) [1], adding model-driven detection of laundering patterns that static rules miss.

3 Architecture

The PSP is organized as the three-layer orchestration system of [1], branded and operated by the institution.

1. **Payment Intent API.** Accepts payment requests from the institution's applications and merchants, normalizing them into intents with idempotency keys [1].

2. **Processor Router.** Applies the scoring model above to select a processor and to fail over across processors on retryable errors [1].

3. **Durable Workflow engine.** Manages the charge, capture, refund, webhook reconciliation, and ledger lifecycle as a single durable workflow with automatic compensation [1].

A webhook handler reconciles asynchronous processor events back into the active workflow, and a double-entry ledger records every movement for financial reconciliation across heterogeneous backends [1]. On the settlement side, ACM composes this orchestration with non-custodial, MPC-based money movement so that programmable payouts and on-chain settlement never require any single operator to hold a complete key [2], and with the Lux, Hanzo, and Zen substrate that supplies the post-quantum digital-securities and tokenized-settlement rails [3].

4 Compliance and Security Posture

The PSP is regulated-first, with controls designed into each release rather than bolted on.

- **PCI scope minimization.** Raw card numbers are never stored; card data is tokenized by the processor's client-side SDK before reaching PSP servers, and the orchestration layer is PCI DSS compliant [1]. The institution's launch is additionally designed to support SOC 2 and ISO 27001 requirements.

- **Key management.** Customer payment tokens are encrypted with AES-256-GCM using per-tenant keys, and all processor API keys are held exclusively in a key-management service rather than in application state [1].

- **Fraud and idempotency.** Processor-native fraud engines compose with additional velocity, amount, and geographic-anomaly rules, and every payment operation is idempotent so duplicate requests never re-process [1].

- **Post-quantum protection.** Sensitive cardholder and merchant flows are protected with cryptography aligned to NIST's 2024 post-quantum standards, ML-KEM (FIPS 203) for key establishment and ML-DSA (FIPS 204) for signatures [6], directly addressing the harvest-now-decrypt-later exposure on long-lived payment records, with the specific post-quantum configuration set per engagement.

- **Confidential reconciliation.** Where multi-party reconciliation or privacy-preserving analytics are required, the platform can compute over encrypted data using ACM's FHE privacy layer, so sensitive figures need not be decrypted to be processed [4], with availability of this layer scoped to the engagement's PSP product tier.

5 Target Outcomes

The figures below are *target* outcomes to model and plan against, derived from cited public benchmarks and from the measurable difference between fragmented vendor stacks and an owned, orchestrated platform. Actual outcomes depend on merchant mix, transaction volume, and which flow is launched first.

1. **Up to 95% lower infrastructure cost** versus a legacy, on-premises payments stack, a target based on ACM's stated platform value proposition for cloud-native infrastructure [7], freeing budget to compete on merchant pricing.

2. **Higher reliability through failover.** The source architecture reports that cross-processor orchestration achieves higher reliability than direct single-processor integration [1]; the institution should target measurable authorization-rate gains from routing, retries, and automatic failover, rather than adopt the source's specific processed-volume or success-rate figures as its own.

3. **No duplicate charges by construction.** Idempotency keys and durable, compensating workflows target the elimination of double-charge incidents under partial failure [1].

4. **Unified, reconciled settlement.** Acquiring, orchestration, and payouts share one ledger and one reconciliation flow, targeting a move from fragmented batch settlement toward near-real-time, fully reconciled disbursements on migrated flows [1].

5. **Recurring economics retained.** By owning the brand and the processing relationship, the institution targets a structural shift of processing margin from a referral partner onto its own balance sheet.

6 What Deployment Looks Like

Delivery is incremental and regulated-first.

- **Start with one flow.** The engagement begins with a single merchant segment or payment method, proves the model in production, then expands across the book, grounding decisions in real transaction behavior, following ACM's phased delivery methodology and timeline.

- **Add processors and rails over time.** Because every processor sits behind the same interface and canonical error mapping [1], new card processors, alternative methods, and crypto channels are added without changing application code.

- **Compliance in scope from day one.** PCI DSS, SOC 2, and ISO 27001 requirements are designed into each release, so launching a payments business never outpaces governance.

- **Owned, not rented.** The institution brands and controls the stack, and can extend the same Banking-as-a-Service rails to embedded-finance and program partners on the Lux, Hanzo, and Zen ecosystem. ACM is a member of the W3A foundation (`w3a.foundation`), which licenses this IP and partners to resell and co-build it for regulated institutions.

References

[1] Z. Kelling. *Hanzo Commerce: Multi-Processor Payment Orchestration for AI-Native Applications.* Hanzo Industries. Source: `hanzo:hanzo-commerce-payments/hanzo-commerce-payments.tex`.

[2] Z. Kelling. *Non-Custodial Securities Custody via Threshold Signing.* Lux Industries, Inc. Source: `lux:mpc-custody/mpc-custody.tex`.

[3] ACM Global Tech. *Post-Quantum Digital Securities: ACM's End-to-End Regulated Stack on Lux, Hanzo, and Zen*. Source: `acm:papers/acm-post-quantum-digital-securities.tex`.

[4] ACM Global Tech. *Compliant Privacy via Fully Homomorphic Encryption*. Source: `acm:papers/acm-compliant-privacy-fhe.tex`.

[5] H. Garcia-Molina and K. Salem. *Sagas*. ACM SIGMOD, 1987.

[6] National Institute of Standards and Technology. *Module-Lattice-Based Key-Encapsulation Mechanism Standard (FIPS 203)* and *Module-Lattice-Based Digital Signature Standard (FIPS 204)*. NIST, 2024.

[7] ACM Global Tech. *Platform Value Proposition: Up to 95% Lower Infrastructure Cost versus Legacy Core Stacks*. Benchmark reference, `acmglobaltech.com`.