Embedded Finance Without Becoming a Bank: ACM's White-Label BaaS and Frictionless Checkout for a Platform Modernizing Its Payments Incrementally

A Representative Modernization Scenario

ACM Global Tech

Abstract

This is a *representative* modernization scenario. It illustrates how a platform, marketplace, or regulated fintech can embed finance—branded accounts, frictionless checkout, card issuing, and payments—under its own brand on ACM Global Tech's white-label Banking-as-a-Service (BaaS) and Payment Service Provider (PSP) layer, and modernize incrementally rather than through a rip-and-replace. The technical substance traces to the frictionless-checkout architecture of Hanzo Checkout [1]: one-click customer recognition, a tokenization boundary that keeps card data out of the embedding application, and intelligent multi-processor orchestration with automatic failover. Outcomes are framed as *target* outcomes against cited public benchmarks.

1 The Pressure

Checkout is the conversion point where transaction friction translates directly into lost revenue. Published research finds that roughly 69.8% of online shopping carts are abandoned, with checkout complexity—multi-step flows, re-entered credentials, and verification overhead—cited as a primary factor [2]. A platform or marketplace that routes its customers off to a third-party processor inherits that friction and surrenders the branded relationship at the most valuable moment.

The strategic temptation is to "own" finance: issue branded accounts and cards, settle payments in-house, and capture the economics. But standing up a chartered bank or a full PSP is a multi-year, capital-intensive, and compliance-heavy undertaking. The institution wants embedded finance—accounts, checkout, issuing, payments—*under its own brand* **without itself becoming a bank or a card processor**, and it cannot tolerate a rip-and-replace of systems that are currently moving money. The real pressure, then, is threefold: reduce checkout friction and abandonment, embed branded financial products quickly, and modernize *incrementally* so that nothing in flight breaks.

2 ACM's Approach: Embedded BaaS + Checkout

ACM licenses an end-to-end, post-quantum, regulatory-compliant infrastructure stack—a full Banking-as-a-Service product, a Digital Securities Platform, a Stripe-class PSP, MPC crypto custody, a DEX, FHE privacy, and a real-time AI/ML transaction-monitoring engine for AML—built on Lux (blockchain, post-quantum cryptography, and FHE), Hanzo (AI and data), and Zen (LLMs). 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. For this scenario the relevant surfaces are the white-label BaaS and the frictionless checkout/PSP layer, which the embedding platform consumes under its own brand.

Branded accounts and issuing on white-label BaaS.
The platform presents accounts, cards, and payments as its own product. ACM's BaaS supplies the regulated rails underneath so the platform is the brand of record, not the bank of record. The institution adds embedded finance without acquiring a charter; the regulated obligations sit with the licensed stack rather than with the embedding application.

Frictionless, one-click checkout.
Returning customers complete a purchase in a single interaction. The checkout layer recognizes a customer from a session token, a device fingerprint, or an email hash, falling back cleanly when no signal is present, and then authorizes against a stored payment token [1]. The flow is structured as a small, well-defined state machine—INIT → IDENTIFIED → AUTHORIZED → COMPLETE—each transition gated and auditable [1].

Intelligent payment orchestration.
Rather than binding the platform to one processor, the layer routes each transaction across a set of gateways by a merchant-configurable score that trades off historical success rate against fee and latency, with automatic failover to the next-best gateway on decline [1]. This improves authorization rates and unit economics without the platform integrating each processor itself.

Incremental modernization.
Because the checkout and BaaS surfaces are consumed as branded, drop-in components behind the platform's own UI, the institution can adopt them route-by-route—one storefront, one product line, one market at a time—while legacy flows keep running. This is the "modernize incrementally instead of a rip-and-replace" posture ACM applies across its representative scenarios.

3 Architecture

The design adapts the Hanzo Checkout architecture [1] into an embedded-finance configuration. Four elements carry the load.

Customer recognition

One-click depends on identifying a returning customer fast and safely. The recognition function consumes multiple signals and degrades gracefully:

$$\mathit{identity}(\mathit{request}) = \begin{cases} c_{\mathit{token}} & \text{if a valid session token is present} \\ c_{\mathit{device}} & \text{if the device fingerprint is known} \\ c_{\mathit{email}} & \text{if the email hash is known} \\ \bot & \text{otherwise (full checkout)} \end{cases} \tag{1}$$

A confidence threshold guards fingerprint matches, and an unrecognized customer falls back to a standard flow rather than failing [1].

Tokenization boundary

A payment token is a tuple $(\mathit{id}, \mathit{type}, \mathit{last4}, \mathit{exp}, \mathit{scope})$, where $\mathit{type} \in \{\text{CARD}, \text{BANK}, \text{WALLET}\}$ and *scope* is the set of merchants authorized to use it [1]. Raw card numbers are held in an isolated vault and never reach the embedding application; the platform handles only tokens. The vault enforces a

token-isolation invariant: access to an underlying credential requires that the requesting merchant be within the token's scope, that the token be valid, and that the merchant be authenticated [1]. This is what lets a platform embed checkout without bringing its own systems into the most sensitive part of card handling.

Orchestration and failover

For a transaction T and gateway g, the routing score balances success, fee, and latency:

$$score(T, g) = w_s \cdot success_g - w_f \cdot fee_g(T.a) - w_l \cdot latency_g, \tag{2}$$

with merchant-configurable weights w_s, w_f, w_l [1]. The orchestrator selects the highest-scoring eligible gateway, and on decline walks the remaining eligible gateways in score order until one succeeds or the set is exhausted [1]. The embedding platform inherits multi-processor resilience and cost-aware routing as a managed capability.

Embedding model

The checkout and account surfaces are delivered as branded, drop-in components. The platform owns the experience and the brand; ACM's BaaS and PSP own the regulated rails, the vault, and the processor relationships. Because the surface area the platform integrates is a token-and-event interface rather than raw card data or processor APIs, adoption can proceed incrementally and reversibly, storefront by storefront.

4 Compliance and Security

- **PCI scope reduction.** The architecture targets PCI DSS Level 1 compliance by construction: a scope boundary $\mathcal{B}$ partitions components that touch raw card data (the vault, $\mathcal{B}_{in}$) from components that see only tokens (the embedding platform, $\mathcal{B}_{out}$), so that no PAN, CVV, or PIN data resides on the platform's systems [1]. Embedding finance therefore does *not* drag the platform's full estate into PCI scope.
- **Token unforgeability.** Tokens are generated with 128-bit cryptographically secure entropy; absent vault access, an adversary cannot forge a valid token except with negligible probability, and merchant compromise leaks tokens rather than credentials [1].
- **Threat coverage.** The source design mitigates network interception via TLS, merchant-system compromise via tokenization, and replay via idempotency keys and timestamps [1].
- **Post-quantum and AML posture.** The embedding stack is post-quantum and regulatory-compliant. ACM operates a real-time AI/ML transaction-monitoring engine for AML and licenses MPC crypto custody and FHE privacy as part of the broader platform.

5 Target Outcomes (vs. Cited Benchmarks)

Outcomes are *targets* measured against public benchmarks, not claimed actuals for any institution. The production figures reported in the source paper [1] are that system's results, not ACM's, and are cited here only as the benchmark the architecture is designed to reach.

1. **Lower abandonment.** Against the $\approx$ 69.8% baseline cart abandonment attributed in part to checkout friction [2], the one-click design targets a material reduction in abandonment; the source paper reports up to a 47% reduction in cart abandonment for one-click versus traditional multi-step flows [1].

2. **Higher conversion.** The source paper reports checkout conversion rising from a 31.2% traditional baseline to 45.9% with one-click, a +47.1% relative improvement [1]; the embedded configuration targets a comparable lift on the platform's own funnel.

3. **Better unit economics.** Intelligent routing in the source paper reduced average transaction fees by 18.3% [1]; the orchestration layer targets similar fee compression for the embedding platform.

4. **Resilience.** The source design reports 99.97% availability with automatic gateway failover [1]; the embedded layer targets high availability and graceful failover as managed properties rather than platform-built ones.

5. **Time-to-market without a charter.** The platform launches branded accounts, issuing, and payments without becoming a bank or PSP, adopting surfaces incrementally; specific timelines and realized results for a given institution are scoped per engagement.

6 Deployment

A representative path mirrors ACM's discovery-to-rollout posture and the incremental nature of the architecture:

1. **Discovery.** Map the platform's current checkout, processor relationships, and compliance posture, and identify the first storefront or product line to modernize.

2. **Embed checkout on one surface.** Drop in branded one-click checkout behind the platform's UI with the tokenization vault handling card data, keeping the platform out of PCI scope, while legacy flows continue elsewhere.

3. **Add branded accounts and issuing.** Layer in white-label BaaS accounts and card issuing under the platform's brand, with regulated obligations resident in the licensed stack.

4. **Turn on orchestration.** Enable multi-processor routing and automatic failover to lift authorization rates and compress fees.

5. **Roll out incrementally.** Extend surface by surface and market by market, retiring legacy paths only as each replacement proves out—never a rip-and-replace.

References

[1] Z. Kelling. *Hanzo Checkout: Frictionless Payment Processing for Modern E-Commerce*. Hanzo Industries. Source: `hanzo:hanzo-checkout/hanzo-checkout.tex`.

[2] Baymard Institute. *Cart Abandonment Rate Statistics*: approximately 69.8% of online shopping carts are abandoned, with checkout complexity a leading factor. As cited in [1].