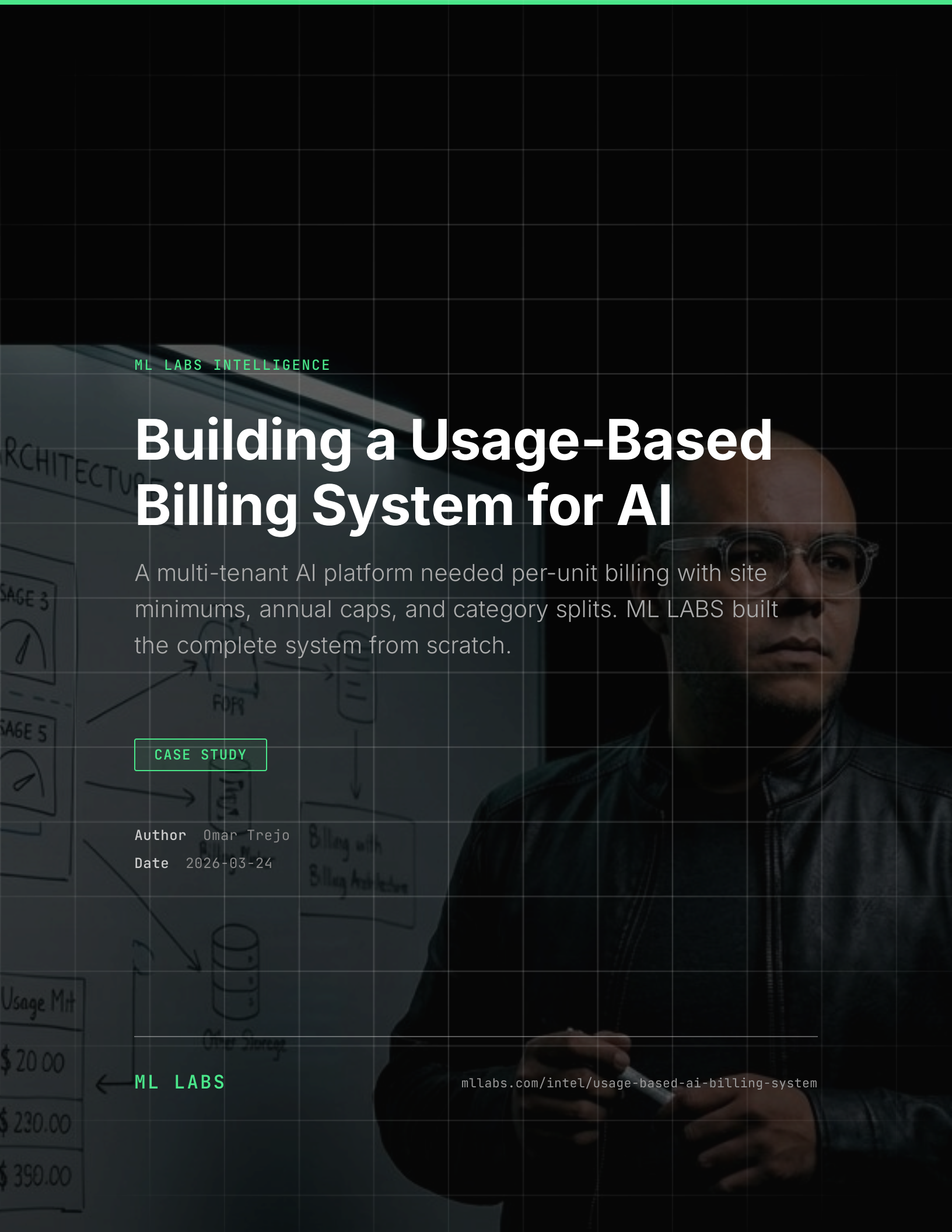


ML LABS INTELLIGENCE

Building a Usage-Based Billing System for AI

A multi-tenant AI platform needed per-unit billing with site minimums, annual caps, and category splits. ML LABS built the complete system from scratch.

CASE STUDY

Author Omar Trejo

Date 2026-03-24

A multi-tenant AI platform was scaling into enterprise contracts and hit a structural wall: the pricing model could not be expressed by any off-the-shelf billing tool. Enterprise customers needed per-unit processing fees with multiple pricing dimensions, per-site minimum commitments, annual spending caps, and role-based finance access with auditable invoices. Standard billing tools handle subscriptions and simple metering. They do not handle the interactions between multiple pricing dimensions that enterprise contracts require.

ML LABS designed and built the complete billing system from deployment infrastructure through invoice PDF generation — a full billing portal deployed across three environments with self-service access for finance teams and automated enforcement of every pricing rule in the contract.

The Problem

The platform processed thousands of units daily across dozens of sites. Enterprise contracts specified multiple pricing dimensions — per-unit rates that varied by processing category, per-site monthly minimums, and annual spending caps — with complex interactions between them. None of these rules exist in isolation; they interact at invoice time in ways that produce incorrect totals if any single rule is applied without awareness of the others.

- Multiple pricing dimensions with interactions that no off-the-shelf tool could express
- Per-site monthly minimums with category-specific enforcement rules
- Annual caps per site accumulating across billing periods for the contract year
- Multi-site invoices requiring structured breakdowns for finance reconciliation
- Role-based access for finance teams who needed self-service invoice retrieval without operational system access

Deployment and Access Infrastructure

Before building billing logic, the platform needed secure multi-environment deployment with finance-specific access controls. ML LABS stood up the full deployment pipeline across three environments — test, US production, and UK production — with environment-specific configuration and tenant isolation.

Authentication provided finance teams with self-service access to invoices and dashboards without exposing operational platform features. MFA enforcement ensured that billing data access met the security posture required for financial records in a regulated environment. Each environment maintained independent access controls, preventing cross-environment data exposure.

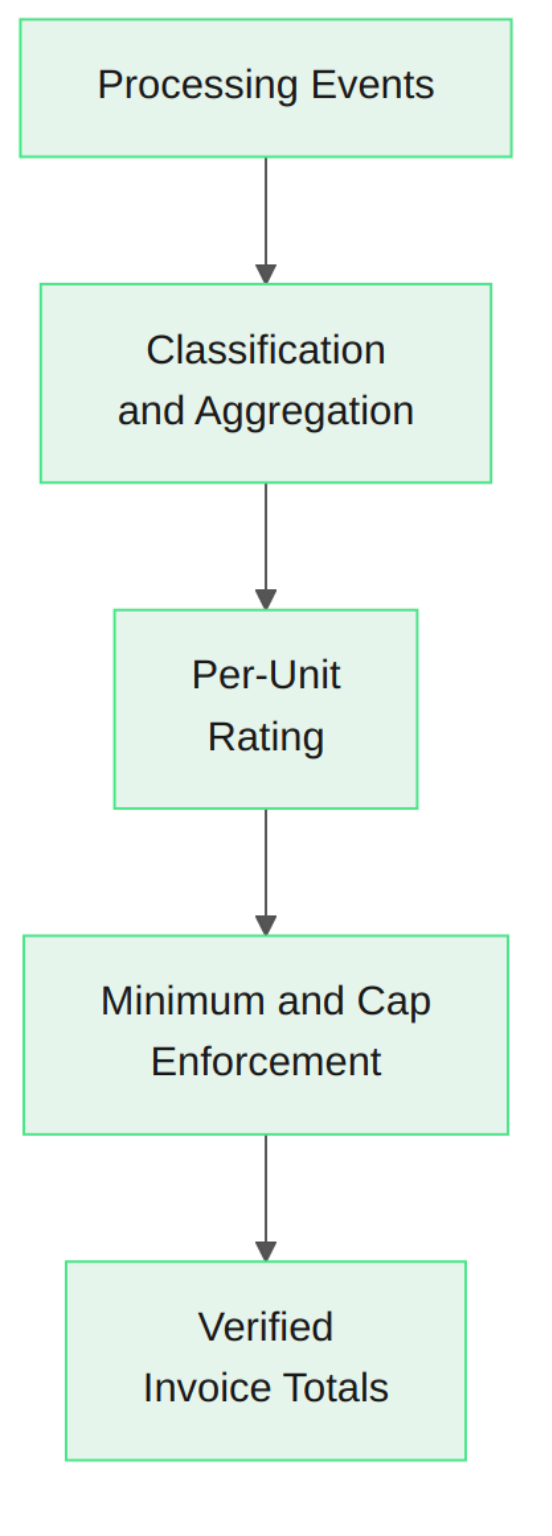
> The billing system is a compliance surface. Every access to financial data — who viewed which invoice, when, and from which environment — must be answerable from audit logs.

Metering and Pricing Engine

Every processing event on the platform generated a billing event tagged with site, category, and timestamp. The metering layer captured these events, classified them by billing category, and aggregated them by billing period and site.

The pricing engine enforced all contractual rules — per-unit rates, site-level minimums, and annual caps — with correct handling of the interactions between them. Getting these interactions right on the first build matters. Each rule interaction

has a specific failure mode that produces invoices finance teams will dispute — and billing disputes erode customer trust faster than feature gaps.


Processing Events
Classification and Aggregation
Per-Unit Rating
Minimum and Cap Enforcement
Verified Invoice Totals

Invoice Generation

The invoice PDF had to serve as both a billing document and a reconciliation tool. Finance teams at large multi-site organizations verify invoices against their internal tracking, and the invoice structure must match their reconciliation workflow.

ML LABS built the invoice generator to produce structured documents with summary totals, per-site breakdowns with line-item transparency, and verification totals that prove the breakdown sums correctly. For organizations with dozens of sites, this meant multi-page invoices with consistent, professional formatting throughout — the level of polish that determines whether a finance team trusts the invoice at a glance or opens a dispute ticket.

Dashboard-Invoice Reconciliation

The billing dashboard showed current-period charges in near real-time. The invoice showed finalized charges after the period closed. These numbers had to match — and initially, they did not. Small discrepancies from timing differences compounded across sites.

ML LABS built automated reconciliation into the billing cycle. Discrepancies above a threshold trigger investigation before invoices are released. This is not a one-time fix but an ongoing guard — real-time and finalized billing systems naturally drift, and the reconciliation layer catches it every period.

> Dashboard-invoice mismatches are the most common source of billing disputes in usage-based products. The fix is architectural: both systems must derive charges from the same reconciled dataset.

Billing Configuration

Enterprise pricing negotiations produce constant configuration changes: new sites onboard with specific minimums, existing sites renegotiate rates, and pricing toggles enable or disable specific charge categories per site. The billing system had to absorb these changes without code deployments.

ML LABS built a configuration layer that supports per-site pricing adjustments, minimum thresholds, annual cap values, and category-level billing toggles — all manageable as configuration rather than code changes. Each configuration change takes effect at the next billing period boundary, and the system maintains a full history of changes for audit purposes.

When This Scope Is Premature

Not every AI product needs a custom billing system. If pricing is a single per-unit rate with no minimums, caps, or category splits, Stripe's metered billing handles it adequately. The complexity threshold is the interaction between pricing dimensions: the moment two or more rules must be enforced simultaneously — category splits interacting with site minimums, or minimums interacting with annual caps — generic tools create more reconciliation work than they save.

The difference between building billing infrastructure for the first time and executing the same pattern with prior experience is not incremental. Teams that have navigated per-tenant metering, rule interaction bugs, and dashboard-invoice reconciliation before compress what would otherwise be a painful discovery process into a controlled build.

Results

The billing portal is deployed and operational across three environments. Finance teams access invoices through a self-service portal with role-based authentication and MFA. All pricing rules — per-unit rates, site minimums, annual caps — are enforced automatically with no manual intervention. Every invoice provides line-item transparency. Dashboard-invoice reconciliation runs before every billing cycle, catching discrepancies before they reach customers.

The system handles the full lifecycle of enterprise billing: site onboarding with custom pricing, mid-cycle configuration changes, multi-page invoice generation, and audit-ready access logging. What would have required a billing operations team running spreadsheet reconciliation runs as automated infrastructure.

First Steps

1. **Map your pricing rule interactions before choosing a billing tool.** List every pricing dimension (per-unit rates, minimums, caps, category splits) and document how each pair interacts. If more than two dimensions interact simultaneously, plan for custom billing infrastructure rather than forcing a generic tool.
2. **Build reconciliation into the architecture from the start.** If any two systems produce billing numbers — a dashboard and an invoice generator, a metering

pipeline and a finance export — their outputs must be compared automatically every billing period. The first reconciliation run will surface gaps.

3. **Separate metering from rating from invoicing.** When pricing rules change with each enterprise negotiation, only the rating configuration should update. When new sites onboard, only the metering classification should expand. Tangling these concerns makes every change a full-system risk.

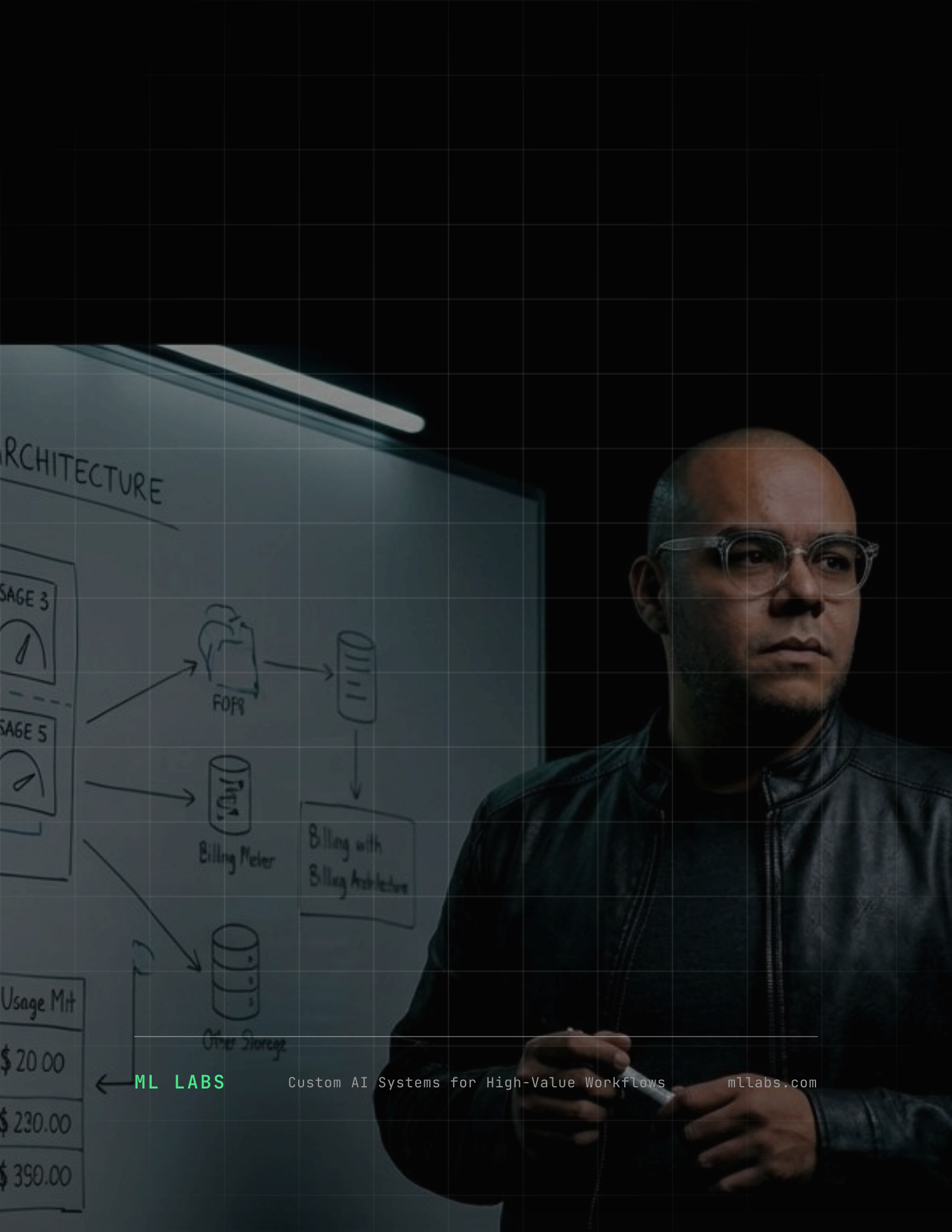
Practical Solution Pattern

Capture every billable event with tenant, site, and category metadata. Feed aggregated usage into a pricing engine that enforces all contractual rules as configuration. Generate invoices from the same reconciled dataset that powers the billing dashboard, deploy across environments with independent configuration, and provide finance teams with self-service access behind role-based authentication.

This works because it separates the three concerns that tangle in generic billing: metering (what happened), rating (what it costs), and invoicing (what the customer sees). The architecture absorbs pricing model evolution without structural rework — a new site minimum, a rate change, a category toggle each affect exactly one configuration layer. Organizations that have shipped usage-based billing for multi-tenant platforms before know where the reconciliation gaps hide and where the rule interactions break. That pattern recognition compresses what would otherwise be an expensive debugging cycle into a controlled, predictable build. If a usage-based billing workflow is already scoped and needs to reach production, **AI Workflow Integration** is the direct build path.

References

1. Bessemer Venture Partners. **The AI Pricing and Monetization Playbook**. *Bessemer Venture Partners*, 2024.
2. Orb. **Stripe's Limitations for Usage-Based Billing**. *Orb*, 2024.
3. Lago. **Revenue Leakage in SaaS: How Billing Gaps Cost 1-5% of ARR**. *Lago*, 2025.
4. CNCF. **CloudEvents Specification**. *Cloud Native Computing Foundation*, 2024.
5. OpenMeter. **How to Price AI Products**. *OpenMeter*, 2024.
6. MGI Research. **Revenue Leakage Series Part 1: From Hidden Risk to Asymmetric Opportunity**. *MGI Research*, 2024.


RCHITECTURE
SAGE 3
SAGE 5
FDR
Billing Meter
Billing with
Billing Architecture
Usage Mit
$ 20.00
$ 230.00
$ 390.00