ALLIED BizTech SOLUTIONS

FIELD GUIDE · 2026

Rebuild Shopify Plus yourself.

The plan we follow.

For D2C brands at $50M to $500M GMV who want to own their storefront, checkout, orders and catalogue, and stop paying a platform fee on every sale.

- → Is it worth it, at your GMV
- → What to rebuild and what to drop
- → The headless stack, in three layers
- → Migration and go-live checklists
- → The data model
- → A week-by-week build plan
- → Checkout, payments, order integrity
- → Where teams get stuck

Three charges grow with GMV. *A rebuild grows with traffic.*

Shopify Plus bills a base subscription, a transaction fee on every sale, and the app-store fees a Plus store collects: Klaviyo, Cloudinary, Algolia, Yotpo, Recharge and the rest. Each grows with revenue. A rebuild replaces them with infrastructure cost, which grows far more gently.

GMV	PLUS + APPS A YEAR	REBUILT: YEAR ONE / AFTER	SAVED IN YEAR ONE
$25M · growing D2C	~$104,000	~$94,500 / $48,000	Stay on Plus
$50M · scaling D2C	~$170,000	~$100,500 / $54,000	~$69K
$100M · established D2C	~$294,000	~$106,500 / $60,000	~$187,500
$250M · large D2C	~$640,000	~$181,200 / $80,000	~$458,800
$500M+ · enterprise, global	~$1,200,000+	~$236,200 / $135,000	~$963,800

Shopify Plus base, transaction fee and apps, May 2026. Card processing is left out of both sides: Stripe direct charges the same 2.7% as Shopify Payments. "Rebuilt" runs Saleor or Medusa.js on AWS with Stripe direct, plus our fixed build price for a single brand: $46,500 up to $100M GMV, $101,200 (the top of the $67,000–$101,200 scale tier) from $100M to $500M. If your own team builds it, put your engineering cost in that place.

THE RULE OF THUMB

Below $50M GMV, stay on Shopify Plus: it is the right call. At $50M, the rebuild saves ~$69K in year one and ~$116K a year after. **At $100M GMV, it saves ~$187K in year one and ~$234K a year after.** From $250M, year one alone saves more than $450K.

Who you need

- **A senior full-stack engineer** at home with Python or TypeScript, Postgres, Next.js and the Stripe API. Production-grade commerce ops, payment compliance and scaling take 4–6 months of senior engineering when you do them yourself.
- **Your ecommerce lead**, two hours a week: the templates, promotions and apps that matter, the pilot product line, the go-live call.
- **An owner after launch**: hosting, upgrades, payment reconciliation, and on-call through every peak sale.

Rebuild the core. *Re-think the apps.*

Shopify Plus's core (catalogue, cart, checkout, orders, customers, fulfillment) maps one for one to Saleor or Medusa.js. The app store is what gets re-evaluated: most apps become a direct integration.

REBUILD ONE FOR ONE

- **Catalogue**: products, variants, inventory, multi-warehouse, multi-locale
- **Cart, checkout and payments**: Stripe Elements, Apple Pay, Google Pay
- **Orders and customers**: order management, returns, refunds, addresses
- **Promotions**: codes, automatic discounts, BOGO, gift cards
- **Multi-currency and tax**: Avalara or TaxJar
- **Fulfillment**: Shippo, EasyPost or your 3PL; inventory synced with your WMS
- **Headless storefront**: your top 5–10 Liquid templates as React components on Next.js
- **Admin dashboard**: orders, catalogue, customers

REBUILD, SIMPLER

- **POS**: Square or Stripe Terminal, by API
- **Markets**: native multi-currency and multi-domain; at 5+ markets, stay on Plus
- **Shop Pay**: Stripe Link and Apple Pay cover ~80% of the repeat-buyer convenience

DROP ON PURPOSE

- **The app store's long tail** (8,000+ apps): rebuild your top 3–5
- **Audiences**: Meta and Google customer lists, direct
- **Functions**: scoped only if you need custom checkout code
- **The account manager**: our managed support tier

ADD WHAT PLUS PRICING DISCOURAGES

- No 0.15% transaction fee
- Direct Postgres access, no API rate limits
- Apps replaced once, at one-time cost

Three layers. *Headless, and yours.*

An open-source commerce engine, a storefront you design, and direct providers for payments, search, email and shipping. Each part is well documented and swappable; none of it charges a share of your sales.

LAYER 1 · STOREFRONT

Next.js or Astro

- Custom theme, full design control
- Incremental Static Regeneration
- SEO and Core Web Vitals
- Per-locale routing and content
- A/B tests and personalization
- Several brands on one engine

LAYER 2 · ENGINE

Saleor or Medusa.js

- Saleor: Python, GraphQL
- Medusa.js: TypeScript, REST, MIT
- Products, orders, customers, inventory
- Promotions, discounts, gift cards
- Multi-currency, multi-warehouse
- Plugins for your own logic

LAYER 3 · INFRA

Direct providers

- **Stripe**: payments, no platform fee
- **Postgres on RDS**: orders, stock
- **S3 + CloudFront**: images, video
- **Meilisearch**: search, for Algolia
- **SES + Postmark**: email
- **Shippo, EasyPost, Shippit**

Decisions to make in week one

DECISION	OUR DEFAULT	WHY
Engine	Saleor or Medusa.js	Pick the language your team will maintain: Python or TypeScript
Where it runs	Your AWS account, Postgres on RDS	Orders and customers stay yours, queryable from your warehouse
Payments	Stripe direct, Elements for card entry	Card data never touches your servers: PCI SAQ-A, the simplest level
Rendering	Next.js with ISR; pages refresh when a product changes	Static speed, current prices and stock
Content	A headless CMS: Sanity or Contentful	Marketing edits pages without a deploy
Tax	Avalara or TaxJar from day one	Tax rules are a service to buy, not code to write

Risk: Stripe Radar screens every payment and Stripe Disputes handles chargebacks with evidence packets, the same infrastructure Shopify Payments runs on. Keep every Terraform module and storefront template in Git from day one.

Ten tables *carry the whole store.*

Saleor and Medusa.js ship most of these; know them before you customise anything. Keep Shopify's ID and order number on every imported row, so the import can run twice and any order traces back to its source.

TABLE	KEY COLUMNS
`products`, `variants`	id, external_id, handle, title, status; variants: sku, options, price, compare_at_price, weight
`inventory_levels`	variant_id, warehouse_id, on_hand, reserved, available, updated_at
`customers`	id, external_id, email, phone, accepts_marketing, stripe_customer_id, tags; addresses in `addresses`
`checkouts`	id, customer_id, currency, lines, discount codes, gift cards, server-computed totals, payment_intent_id, expires_at
`orders`	id, external_id, order_number, customer_id, currency, subtotal, discount, shipping, tax, total (integer cents), financial_status, fulfillment_status, checkout_id (unique)
`order_lines`	order_id, variant_id, sku, quantity, unit_price, discount_allocations, tax_lines
`payments`, `refunds`	order_id, stripe_payment_intent_id or refund_id, amount, currency, status, at
`fulfillments`	order_id, lines, warehouse_id, carrier, tracking_number, status, shipped_at
`discounts`, `gift_cards`	code, type (percentage, fixed, BOGO, automatic), rules, usage limits; gift cards: code hash, balance, currency, expires_at
`order_events`	order_id, type, from_state, to_state, actor (customer, staff, webhook), stripe_event_id, at

TWO RULES THAT SAVE A REBUILD

Store money as integer cents, with its currency, on every row: rounding in floats is how totals drift from Stripe. **Only a verified Stripe webhook marks an order paid**, never the browser's return from checkout: a closed tab must not lose an order, and a forged redirect must not create one.

Sixteen weeks to build, *then a pilot and a cutover.*

This is the order we build in. Each stage ends with something your ecommerce lead can try. Then one product line sells for real for four weeks, and four weeks of cutover run both stores side by side: about six months from kickoff to Shopify-free.

WEEK	BUILD	DONE WHEN
1–2	**Audit and model.** Choose the engine. Export from the Admin API into staging: products, variants, inventory, customers, addresses, orders, gift cards. AWS, RDS and Stripe set up.	Counts in staging match Shopify; 50 orders spot-checked
3–5	**Engine and catalogue.** Multi-warehouse inventory, promotions, discounts, gift cards, multi-currency, tax through Avalara or TaxJar.	Every SKU's price, stock and tax match Shopify
6–8	**Storefront.** Next.js, your top 5–10 Liquid templates as React components, Meilisearch, images on S3 and CloudFront, per-locale routes.	Product and collection pages pass Core Web Vitals
9–11	**Checkout and payments.** Cart, Stripe Elements, Apple Pay, Google Pay, Link, Radar, subscriptions; order states, webhooks, refunds (page 7).	Test orders reconcile with Stripe to the cent
12–16	**Admin, fulfillment, launch prep.** Order management, returns, Shippo, EasyPost or your 3PL, WMS sync, your top 3–5 apps, a 301 map for every URL, a load test.	Customer service works a day of orders without Shopify
17–24	**Pilot and cutover.** One product line sells for four weeks with a daily fix list; then four weeks with both stores running, a delta import, and DNS moved.	The pilot line converts as well as it did on Shopify

AFTER CUTOVER

Let the Shopify Plus contract **run to its renewal date**, then let it expire. Order history already sits in your Postgres and every old URL 301-redirects, so search rankings carry over and nobody is in a hurry.

Every payment, one order. *Every order, reconciled.*

A storefront is pages. Checkout is where the money is: one bug charges a customer twice, sells stock you do not have, or ships an order nobody paid for. Build it in this order.

1. **Keep card data off your servers.** Stripe Elements collects the card in Stripe's frame, which keeps you at PCI SAQ-A. Apple Pay, Google Pay and Stripe Link run through the same element.
2. **Price on the server.** Recompute lines, discounts, gift cards, shipping and tax on the server when checkout starts and again before payment. The browser only shows the totals.
3. **One checkout, one payment, one order.** Create the PaymentIntent with an idempotency key from the checkout ID, and make `checkout_id` unique on orders. A double click or a retry finds the order already made.
4. **Trust the webhook, once.** Verify every Stripe signature and store each event ID before acting on it. A repeated delivery changes nothing; an order moves to paid only from `payment_intent.succeeded`.
5. **Hold stock, then commit.** Reserve inventory when payment starts, commit it on success, release it on failure or expiry. No oversold stock, and no stock stuck in abandoned carts.
6. **Reconcile every day.** Match Stripe's balance transactions to orders and refunds, and flag any difference before finance finds it.

Conversion without Shop Pay

Shop Pay's accelerated checkout is a real Shopify advantage. Stripe Link, browser autofill, Apple Pay and Google Pay replace it; in our deployments conversion matches or beats Shop Pay, because the new checkout has no redirects and a shorter path from cart to payment.

Fraud and chargebacks

Stripe Radar screens every payment. Stripe Disputes handles representment; keep tracking numbers and delivery proof on each order so the evidence is ready.

Move it, launch it, *run it.*

Migration

- ☐ Export twice: everything in week one, the changes since at cutover
- ☐ Keep Shopify's ID and order number on every row; start new order numbers above the last
- ☐ Move gift cards with their balances
- ☐ Passwords do not export: invite customers to set one, or sign them in by email link
- ☐ Move Recharge subscriptions to Stripe Subscriptions
- ☐ 301 every product, collection, page and blog URL
- ☐ Rebuild your top 3–5 apps; list the rest for phase two
- ☐ Check counts, then trace 50 orders end to end

Go-live

- ☐ Stripe live keys, Radar rules and webhooks verified
- ☐ Tax and shipping checked on test orders in every market
- ☐ A load test at your peak traffic
- ☐ Redirects live and the new sitemap submitted
- ☐ Alerts on checkout errors, failed payments and webhook backlog
- ☐ Rollback ready: DNS back to Shopify in one change

Running it

- ☐ **Always on:** checkout and payment alerts, security patches
- ☐ **Daily:** Stripe reconciliation against orders and refunds
- ☐ **Weekly:** conversion, Core Web Vitals and failed payments to the ecommerce lead
- ☐ **Monthly:** engine and dependency upgrades, through staging first
- ☐ **Before every peak sale:** a load test and a code freeze
- ☐ **Quarterly:** cost review, AWS against the Plus bill you replaced
- ☐ **Yearly:** a restore drill, an access review, the PCI SAQ-A

Who owns it

Name one person before launch. The store is your revenue: someone has to answer when checkout errors at 9 p.m. on Black Friday.

Five ways a rebuild *stalls.*

1. Moving to another hosted platform

BigCommerce Enterprise drops the transaction fee and adds GMV-based contract escalation. Adobe Commerce Cloud runs $40–$200K a year; commercetools $200K+. Same economics as Plus at scale, plus a migration.

2. Self-hosting Saleor or Medusa.js as a side project

You can. Production-grade commerce ops, payment compliance and scaling are 4–6 months of senior engineering. Plan for that, or the store launches without the parts that protect revenue.

3. Rebuilding below $50M GMV

Shopify Plus is a good product, and below $50M it is the right call: at $25M the rebuild saves less than $10K in year one. Grow first; run the numbers again at $50M.

4. Rebuilding the whole app store

A Plus store collects apps. Rebuild your top 3–5 as direct integrations and keep a phase-two list; most of the rest are never missed.

5. Skipping the pilot and the redirects

Four weeks on one product line find the discount rule nobody listed and the 3PL edge case. A missing 301 map costs the rankings that bring your traffic.

And one that is not a mistake: negotiating

Shopify discounts enterprise contracts ~10–20% on multi-year commits, especially at $100M+ GMV, but the 0.15% transaction fee is largely fixed. Negotiate if the rebuild is a year away; it captures less than a quarter of what the rebuild does at scale.

RATHER HAVE IT BUILT?

We build this exact plan. *Fixed price.*

The same scope, stack and sixteen weeks, delivered by a senior team that has shipped 250+ products.

$46,500 fixed for a single brand up to $100M GMV, $67,000–$101,200 from $100M to $500M, +$19,100–$32,800 for 3–5 brands, agreed before we start

Live in 12–16 weeks, then a four-week pilot on one product line before full cutover

Every order moves, with full history and order numbers; every old URL 301-redirected

90-day warranty, quarterly check-ins in year one, and the code in your GitHub from day one

We can run it for $24,000–$48,000 a year: hosting, monitoring, monthly upgrades and SLA support

alliedbiz.tech/rebuild/shopify-plus-alternative

Allied BizTech · 15+ years · ISO 9001:2015 · 250+ products shipped. Shopify and Shopify Plus are trademarks of their owner, named here for identification only; this guide is independent and not endorsed by Shopify. Prices as of May 2026.