ALLIED BizTech SOLUTIONS

FIELD GUIDE · 2026

Cut your D2C tool bill yourself.

The plan we follow.

For D2C brands at $25M to $500M GMV whose email, image, search and reviews tools all bill per customer: what to replace first, what to keep, and how to switch without losing a sale.

- → What your tool bill charges for
- → Replace first, replace next, keep
- → The architecture, in three layers
- → Decision and switch checklists
- → One customer record
- → A plan around your renewal dates
- → Protecting revenue at the switch
- → Where brands get stuck

Your customers grow. *So does every bill.*

D2C teams run 5–8 customer-facing tools, and each bills on a number that rises with growth: contacts, searches, apps, orders. The tool line outpaces revenue. Four of those tools share the same customers and sit on commodity parts: transactional email, an image CDN, search, and reviews with loyalty.

TOOL · WHAT IT BILLS ON	BEFORE, A YEAR	AFTER, DIRECT BILLING
Klaviyo · contacts (500K)	$48K	SES + Postmark + Twilio, ~$3K
Cloudinary · plan (Advanced)	$22K	Cloudflare R2 + Sharp, ~$1.5K
Algolia · searches (5M a month)	$24K	Meilisearch on a VPS, ~$3K
Yotpo · apps (Reviews, Loyalty, Subscriptions)	$18K	Postgres + AWS, ~$1K
Hosting and ops	–	$8K
A typical brand at $80M GMV	$112K	**$16.5K, after a $47.4K build**

The $47.4K is our one-time rebuild price for the four. If your own team builds it, put your engineering cost in that place.

THE RULE OF THUMB

At $80M GMV, year one returns ~$48K after the build and ~$95K a year from year two: the build pays back in about six months, ~$239K over three years. At $250M GMV the same four cost ~$280K a year, and year one alone returns more than $158K. On a $48K-a-year stack, at $25M GMV, the build takes about 15 months to pay back.

Who you need

- **An owner for the customer record**, usually the retention or CRM lead: they decide what "one customer" means and sign off each switch.
- **Finance, once**: every tool's renewal date and notice period, and next year's bill at next year's customer count.
- **Senior engineers** at home with Postgres, the Shopify API, AWS SES and DNS, object storage and a search engine. Our team ships the four in 8–10 weeks.

Replace what bills per customer. *Keep what runs the store.*

One test sorts the list: does the bill rise when your customer count does, while the job underneath is commodity infrastructure? Then it is a rebuild candidate. Your store, checkout and payments stay where they are.

REPLACE FIRST: FOUR TOOLS, ONE CUSTOMER

- **Klaviyo**: email and SMS on SES, Postmark and Twilio, billed per send, not per contact
- **Cloudinary**: images on Cloudflare R2, transformed with Sharp
- **Algolia**: search on self-hosted Meilisearch
- **Yotpo**: reviews, loyalty and referrals on your Postgres, with the same widget on the product page

REPLACE NEXT: PER CUSTOMER OR PER EVENT

- **Gorgias**: support
- **Rebuy**: recommendations
- **Postscript**: SMS, folded into the Twilio module
- **Builder.io**: the page builder
- **Smile.io**: loyalty, folded into the loyalty module

KEEP

- **Shopify**: store, checkout, payments. The rebuild reads its webhooks and API; a Plus contract is unaffected
- **Subscriptions**: ReCharge or Skio, joined by webhook
- **A tool you love**: drop its module and the bundle price falls by ~$10–14K. Yotpo is the one brands keep most often

DROP

- **Four logins, four dashboards**, four attribution sources
- **The features nobody opens**: rebuild the 15–25% your team uses
- **Four renewal dates** and four annual price hikes

GOING ONE TOOL AT A TIME

Start with the largest line, usually Klaviyo, and commit to the other three within six months: **that keeps the bundle price**. Past six months, each rebuild is priced on its own.

03 · ARCHITECTURE

One customer record. *Four jobs on top of it.*

Today each vendor keeps its own copy of your customer. Rebuilt, there is one: the store's customers and orders in your Postgres, and every module reads and writes against it. Shoppers see the same widgets, your team gets one app, and the bills go direct.

LAYER 1 · SHOPPERS SEE

The same storefront

- The reviews widget on the product page, unchanged
- The loyalty signup, same flow
- The search box and results
- Product images
- Email and SMS from your domain

LAYER 2 · ONE RECORD

Your customer data

- Customers, orders and consents in Postgres
- Every email, search, review and point as an event
- Retention views across all four
- One app and one sign-in for the team

LAYER 3 · DIRECT BILLS

Your accounts

- **AWS SES + Postmark**: email
- **Twilio**: SMS
- **Cloudflare R2 + Sharp**: images
- **Meilisearch** on a VPS: search
- **Postgres** on AWS: the record

Decisions to make before the build

DECISION	OUR DEFAULT	WHY
Bundle or one tool	All four in one build	Shopify is wired once; the customer record is whole from day one
Whose ID is the customer	Shopify's customer ID	Every order, export and vendor ID joins on it
Where the data lives	Postgres on your AWS	A retention question becomes one query
Sending identity	Your domain, with SPF, DKIM and DMARC	Direct SES with proper domain auth often delivers better
Image domain	Your own, on R2	The next change of engine moves no URL
Who owns it	One named person	Four tools had four owners; one app needs one

One customer. *Every touch.*

The data is what four vendors never give you: email, search, reviews and loyalty on one customer, for the first time. Model it so a retention question is one query, not four exports.

TABLE	KEY COLUMNS
`customers`	id, shopify_customer_id, email, phone, first_order_at, orders_count, total_spent, loyalty_tier
`identities`	customer_id, source (Shopify, Klaviyo, Yotpo, search session), external_id, matched_by (ID, email), at
`consents`	customer_id, channel (email, SMS), status, source, opted_in_at, withdrawn_at
`orders`, `order_lines`	shopify_order_id, customer_id, placed_at, total, discount_code; product_id, variant_id, quantity, price
`touches`	customer_id, module (email, search, review, loyalty, referral), type (sent, clicked, searched, reviewed, earned, redeemed), ref_id, at
`sends`, `searches`, `reviews`, `loyalty_ledger`	each module's detail; every row carries customer_id
`attributions`	order_id, touch_id, rule, value
`customer_daily`	customer_id, date, orders, revenue, points_balance, last_touch: the rollup retention views read

Questions one record answers

- Do customers who leave a review reorder sooner than those who do not?
- Which loyalty tier clicks your emails most, and which never opens them?
- What do repeat customers search for and fail to find?
- Which flow brings lapsed loyalty members back?

TWO RULES THAT SAVE A REBUILD

Every row about a person carries customer_id: an anonymous search session joins its customer at login or checkout. **Append touches, never overwrite them**: retention is a question about time.

Plan it around *your renewal dates.*

Four tools mean four contracts and four renewal dates. Line the work up so each tool is off before it renews, and no vendor rolls into another year by default.

WHEN	STEP	DONE WHEN
Before	**Count the bill.** Each tool's plan, the number it bills on, its renewal date and notice period; next year's bill at next year's customer count.	Finance has one sheet for all four
Before	**Choose the scope.** All four in one build, or the largest first with the rest committed within six months.	Scope and order signed off
1–2	**The customer record.** Shopify backfill and webhooks; every vendor's export matched to one customer; consents and suppressions in.	Every vendor row matched, or on a review list
3–5	**The modules, behind the same widgets.** Search, images, reviews and loyalty, then email and SMS. Each runs beside its vendor once built.	Each module matches its vendor on your data
6–7	**Pilot.** Two weeks on one brand or category.	The five numbers on the next page hold
8–10	**Switch, lowest risk first.** Search, then images, then reviews and loyalty; email last.	Each vendor read-only, then cancelled at renewal
After	**Use the record.** Retention views across all four; the next tool on the list joins as a module.	Finance gets a cohort report it never had

NOTICE PERIODS

Read every notice period before the build starts. **Give notice when the pilot passes, not before**, keep each vendor read-only until its renewal date, then cancel. No one works against a deadline the contract set.

Same widgets. *Same revenue.*

Shoppers should notice nothing. The reviews widget on the product page looks identical, the loyalty signup keeps its flow, and email goes out through SES. Five numbers prove it: record each before the build and watch it through the switch.

NUMBER	WHAT MOVES IT	HOW TO HOLD IT
Email revenue and inbox placement	A new IP and a new sender	SPF, DKIM and DMARC on your domain; a dedicated SES IP warmed 2–3 weeks on your most engaged contacts; 2–4 weeks beside Klaviyo; switch when placement matches
Stars on product pages and in Google	Review counts that reset; markup that changes	Import every review with its date, rating, verified flag and photos; compare counts and averages product by product; render the star markup from the numbers the widget shows
Search conversion	Relevance drift, lost synonyms	Run your top queries on both engines and compare the results; import synonyms; watch zero-result searches daily
Loyalty balances	Points lost or counted twice	Rebuild the ledger from orders, compare every balance daily, import the final delta at the switch
Images and page speed	Broken or slower image URLs	Keep the old URL pattern on your own domain; test a corpus of production URLs before DNS moves

Switch one module at a time

Each module runs beside its vendor for at least two weeks before it takes over. If a number drops, that module stays on its vendor until it is ready, and the others carry on. The standard is zero customer-visible downtime.

Send from one place

While email runs in parallel, each customer hears from one system only: split by brand, category or segment. One abandoned-cart email, one review request per order.

Decide it, switch it, *run it.*

Before you decide

- ☐ Every customer-facing tool listed, with the number it bills on
- ☐ Renewal dates and notice periods in one sheet
- ☐ Next year's bill at next year's customer count
- ☐ The four core tools priced against the build plus running costs
- ☐ The tools you keep, written down with the reason
- ☐ A named owner for the customer record

Before each switch

- ☐ Baselines recorded for the five numbers
- ☐ Consents and suppressions imported before the first send
- ☐ Review counts checked product by product
- ☐ Every loyalty balance matches
- ☐ Old image URLs resolve on the new domain
- ☐ A rollback written for the module
- ☐ Notice given only after the pilot passes

Running it

- ☐ **Weekly:** revenue by email flow, zero-result searches, reviews awaiting moderation
- ☐ **Monthly:** retention by cohort across email, search, reviews and loyalty; deliverability
- ☐ **Quarterly:** direct bills against the four tools you replaced; open loyalty points as a liability
- ☐ **Yearly:** a restore drill, an access review, and the next tool on the list

Who owns it

Name one person for the customer record. Four vendors each sent an account manager; one app has one owner, who answers for the data, the switch and the numbers after it.

Six ways the bill *keeps growing.*

1. Swapping one wrapper for a cheaper one

Omnisend and Sendinblue run 10–25% under Klaviyo; Okendo and Stamped start lower than Yotpo. Same economics at a lower starting price: you run this exercise again when their pricing rises with usage.

2. Best of breed

Loox, Smile, Friendbuy and Postscript are each fine on their own. Together they are four logins, four dashboards, four attribution sources and four contracts.

3. Four separate projects

Rebuilt one by one, the four cost ~$54,700 and ~16 weeks with one team, with the Shopify integration built four times. As one build: $47,400 in 8–10 weeks.

4. Starting with the long tail

Support, recommendations and the page builder are real savings. The four core tools share one customer and carry most of the bill: build the record with them, and the rest plug in later.

5. Rebuilding the tools, not the data

Four rebuilt tools with four customer tables keep the silos. The retention view, the reason to own the stack, never arrives.

6. Dropping reviews and loyalty to save the fee

From $5M GMV, reviews and loyalty drive order value and lifetime value. Own them; don't drop them.

And one that is not a mistake: negotiating

Vendors take 10–25% off an annual commit. Do it for any renewal that lands before your switch; it captures less than an eighth of what owning the stack does.

RATHER HAVE IT BUILT?

We build this exact plan. *Fixed price.*

The same order, the same customer record and the same widgets, delivered by a senior team that has shipped 250+ products. It starts with a free 30-minute D2C Stack Review.

$47,400 fixed, one time: Klaviyo, Cloudinary, Algolia and Yotpo as one app

$112K a year down to $16.5K at $80M GMV, ~$48K back in year one

Live in 8–10 weeks, after a two-week pilot on one brand or category

Same widgets, same workflow; your data unified in your own Postgres

90-day warranty, and the code in your GitHub from day one

We run it for $8,000 a year: hosting and ops

alliedbiz.tech/rebuild/for-d2c-brands

Allied BizTech · 15+ years · ISO 9001:2015 · 250+ products shipped. Klaviyo, Cloudinary, Algolia, Yotpo, Shopify, Shopify Plus, Gorgias, Rebuy, Postscript, Builder.io, Smile.io, ReCharge, Skio, Omnisend, Sendinblue, Okendo, Stamped, Loox, Friendbuy, AWS, Amazon SES, Cloudflare, Cloudflare R2, Postmark, Twilio, Meilisearch and Google are trademarks of their owners, named here for identification only; this guide is independent and not endorsed by any of them.