

Your Software Investment Is the **Barrier** to AI Enablement

Tech debt, redefined: **legacy tech that can't convert to the JSON shape AI, Google, and Shopify now run on.** Not old code — the wrong shape. And most of it isn't legacy at all: it's on this year's budget.

Status: Published · 2026-07-13 Canonical: [crm-sync.dev/wrong-shape](#) Pair: [Challenge → Solution](#)
#merchant-ai #google-merchant-feed #gpc #graphql #rest #soap #ucp #feedonomics #rithum #salsify
#jda-blue-yonder #attio #forward-deployed-infrastructure #egress #cloudflare #r2

The right shape, shipped: [PIM Sync – the GraphQL, AI-shaped PIM · Shopify App Store](#) →

[Intro deck: Data Reshape for UCP \(PDF\)](#) →

A lot of organizations are signing for Salsify, Feedonomics, or Rithum right now on one hope — that the purchase will automatically align their product data with what's coming: cross-border catalogs, agentic commerce, UCP. The vendors' own documentation says the opposite.

THE RECEIPTS

The internal model is rows plus transformers. Feedonomics organizes product data into “databases” of rows mutated by field-level, Excel-formula-style transformers. The Retail tier's flagship workflow is literally download to Excel, edit, re-upload. Ingestion accepts CSV, TSV, XML, JSON, NDJSON — but everything normalizes to the tabular shape.

The receipt is their own connector. Feedonomics' open-source Shopify connector states it pulls “using a combination of the Rest API and GraphQL, and will output that data into a singular CSV.” Read that twice: it consumes Shopify's nested graph and flattens it as step one. The shape every AI reader needs is discarded at the front door, then parent-child gets rebuilt downstream as one-row-per-variant plus `item_group_id` — a channel file convention, not a data model. The platform API is REST-only; there is no GraphQL surface.

The whole chain is the same shape. Rithum (ChannelAdvisor + CommerceHub) documents REST for new integrations, a V7 SOAP API for legacy, and template-driven flat-file feeds as

the interchange. Upstream, the JDA/Blue Yonder planning-and-WMS estate contracts in batch CSV.

SHOPIFY → GOOGLE MERCHANT CENTER FEED

Full Shopify Product Schema → Google Merchant Center Feed Attributes			
shopify GraphQL Field	Google Feed Attribute	Required?	Notes
product.title	g:title	Required	Max 180 chars; Include key attributes (color, size, material)
product.descriptionHtml	g:description	Required	Strip HTML; max 5,000 chars
productVariant.sku	g:id	Required	Must be unique and stable across feed submissions
productVariant.price	g:price	Required	Include currency code (e.g., 00.00 USD)
productVariant.compareAtPrice	g:sale_price	Recommended	Only send when less than g:price
product.category.fullName	g:google_product_category	Required	Use full GPC path or numeric ID
metafield: feed_data.google_product_category_id	g:google_product_category	Required (alt)	Numeric GPC ID override (e.g., 287)
metafield: feed_data.google_product_type	g:product_type	Recommended	Merchant-defined category hierarchy
productVariant.selectedOptions(Color)	g:color	Required (apparel)	Raw value from variant option
productVariant.selectedOptions(Size)	g:size	Required (apparel)	Raw value from variant option
product.vendor	g:brand	Required	Maps directly
productVariant.barcode	g:gtin	Required (if available)	EAN/UPC/ISBN
productVariant.inventoryPolicy	g:availability	Required	DENY → In stock; CONTINUE → preorder
productVariant.imageUrl	g:image_link	Required	Must be variant-specific Image URL
product.onlineStoreUrl + variant.params	g:link	Required	Canonical PDP URL with variant query params
metafield: custom.material_composition	g:material	Recommended (apparel)	Custom metafield
metafield: custom.age_group	g:age_group	Required (apparel)	newborn, infant, toddler, kids, adult
metafield: custom.gender	g:gender	Required (apparel)	male, female, unisex
productVariant.weight + weightUnit	g:shipping_weight	Recommended	Used for calculated shipping in Merchant Center

The map the flat stack can't hold: Shopify's GraphQL product schema paired to Google Merchant feed attributes – variants, metafields, and categories as one structure.

THE TYPICAL BUSINESS UNIT — A FLAT-FILE RELAY

1. Supply chain owns the item master in a CSV-only system of record.
2. A nightly export becomes the inter-team data contract.
3. A feed-manager seat rebuilds variant relationships by hand in middleware.
4. Per-channel exports ship on schedules — staleness equals cadence.
5. BI receives yet another un-joined copy.

It's also why the lights-out warehouse never arrives on this stack: automation needs event-driven, graph-shaped, machine-speed exchange, and a nightly flat file structurally can't carry it.

A-PLUS SOFTWARE, WRONG SHAPE

Let's be clear about the vendors: these are A-plus platforms. Feedonomics and Rithum earned their market — they syndicate channel files at massive scale, reliably, in the era when the channel file *was* the product. Nobody building them in 2015 could have anticipated AI in e-commerce, agents reading catalogs as nested objects, or Shopify freezing REST. This isn't a quality problem and it isn't anyone's failure. **They are simply not the right shape.** And shape is the one thing a purchase order can't transform — buying excellent flat-shape software as your AI-alignment layer is still signing next year's tech debt at this year's kickoff.

THE SHAPE ALREADY MOVED

Shopify forced the question: the REST Admin API is frozen; the full variant model, metaobjects, and categories exist only in GraphQL's nested shape. Google's side reads the same way — the Shopping Graph, the Merchant API, the agent surfaces all consume structured objects. Cross-border makes it non-negotiable: market variants, localized prices, auditable price history. A row can't hold any of that.

The same test applies to the identity side of the estate. Salesforce's clouds — and even AI-native CRMs like Attio — govern the relationship record: service, sales, the account. None of them sit on the commerce protocol surface where catalogs are read and agent orders transact. However modern your CRM, the catalog side of the stack has to produce the shape itself.

RENT THE AI-SHAPED TOOL, OR OWN THE SHAPE

Attio's rise makes the point from the other direction: the market clearly wants AI-native, graph-flexible tooling — that's why teams leave rigid CRMs for it. But there is a strategic option nobody prices in that comparison: **build your own.** Once your substrate stores the graph — products, identities, consent, and entitlements as real relations, with REST and GraphQL surfaces — an Attio-class experience becomes a composition over data you already own: bespoke views, AI answers, agent access, no per-seat meter, no migration hostage. The choice this cycle isn't which vendor's shape to move into. It's whether the shape lives in a vendor's database or in yours.

THE INVERSION: STORE THE GRAPH, RENDER THE PROJECTIONS

Real parent-child relations in the database, Shopify GraphQL consumed natively, nested JSON emitted — **the CSV becomes a render target, never the database**. Your spreadsheet still works: it enters once, gets AI-fied, and the Google feed, BigQuery, and the AI-queryable catalog are all projections of the same living structure.

So in the vendor meeting, skip the feature matrix and ask one question: *show me one product as a nested object, live*. The layer that can is the one converting the others.

WHERE THE SHAPE SHOULD LIVE — THE EGRESS QUESTION

One fact most dev teams and enterprise leadership haven't internalized: **if you run on Shopify, you already run on Cloudflare**. Shopify's storefront CDN is Cloudflare; Oxygen — Shopify's Hydrogen hosting — is Cloudflare Workers. The edge your buyers already hit is not a vendor decision you have left to make; it's the ground your platform stands on.

That's why the pairing is Webflow + Xano + Cloudflare rather than another SaaS on an egress-billed cloud. Webflow is the design surface — and it exports to any filesystem your team runs (React, Vue, Svelte, Angular, via Vite/TypeScript), so the front end is never hostage. Xano is the system of record — Postgres with Redis on Docker/Kubernetes, REST out, Shopify GraphQL consumed natively, a working instance minutes after signup. Cloudflare carries the functions and the assets: in this stack the Designer Extension, the media, and the decks all serve from Workers and R2 — versioned deploys with failover, publish management an enterprise team can operate, instead of a single dev holding an external vendor's external database. SOC 2 across all three. And with an AI tool runner over that substrate, any API shape you need gets composed in real time — no per-seat SaaS between you and your own data.

Egress is the tax the AI era multiplies. Agent commerce means machine reads — feeds fetched, catalogs crawled, RAG queries answered, media pulled by every AI surface, all day. On an egress-billed cloud, every one of those reads is a metered event; the more legible your data becomes to machines, the bigger the bill for being read. R2 charges zero egress. So the rule for UCP-era commerce is one sentence: **keep your assets and your functions where your buyers' platform already lives — on the edge, where being read is free**. And the governance artifact rides the same substrate: the JWE-timestamped ledger is the document a CISO or DPO actually signs off on — consent, mandates, and deploys, versioned and auditable, forward data governance instead of a policy PDF.

Because what you're buying this cycle isn't a layer. It's **forward-deployed infrastructure — a substrate that amends the shape of your data as the ecosystem reshapes**. The shape moved in 2024 and it will move again; forward-deploy is how the next shift stops being a repurchase event.

Sources: [Feedonomics Platform API](#) · [Feedonomics ingestion](#) · [feedonomics/shopify-catalog-connector](#) · [Rithum/ChannelAdvisor API](#) · [Rithum Flex Feeds](#) · [Terminology: REST, SOAP, GraphQL, Merchant AI](#)

The store-the-graph inversion, productized: **PIM Sync for UCP Data** — the GraphQL, AI-shaped PIM. Tailored for GraphQL-era Shopify, built for Google feed managers, BigQuery-ready. Intro deck: [Data Reshape for UCP \(PDF\)](#) — the Shopify GraphQL → GPC → Merchant-feed schema maps, slide by slide.