A REFERENCE ARCHITECTURE FOR SHOPIFY BRANDS

The Tracking Stack.

Eight layers, one architecture. The exact tracking stack I rebuild every Shopify ad account around. No email gate.

By Conner Crowe

Fractional marketing lead · Spokane, WA
Working directly with founders.
A decade in. Independent since 2024.

The Architecture, at a glance.

PIN THIS PAGE
TO YOUR SECOND MONITOR

01

The store

Shopify (or platform). The single source of truth for orders, refunds, customers. Everything downstream depends on the store data being clean.

02

GTM web

Browser-side tag manager. Fires events client-side.

03

Server-side container

First-party endpoint. Proxy for every platform. The spine.

↓

04

GA4

Behavioral analytics. Not the source of truth for ad performance. Used for behavioral cohorts, audience export, and triangulation.

05

Google Ads

Server-side conversion API + enhanced conversions.

06

Meta + paid social

CAPI + browser pixel, deduped via event_id. TikTok and Pinterest fan out from the same spine.

07

Email and SMS

Klaviyo, Postscript. The cleanest first-party identity graph you have. Feeds back into Meta CAPI and Google enhanced conversions.

08

Reporting

Written Monday summary. Looker Studio for ad-hoc questions. Not a real-time dashboard nobody opens.

Direction of data flow: 01 (the store) emits the canonical events. 02 (browser GTM) fires the same events client-side. 03 (server container) is the first-party proxy that all downstream platforms receive from. 04 GA4 reads behaviorally; 05/06 Google Ads + Meta consume conversions for bidding. 07 Email feeds first-party identity back upstream. 08 Reporting is a separate read-layer.

Built for the tracking lead at a $5K to $200K Shopify brand.

Whether that lead is the founder, the in-house marketer, or a contracted developer. The doc is opinionated. Read the architecture poster on page 2 once, then dip into individual layers as needed.

01

It is a reference, not a tutorial.

The exact menu paths and code snippets live in vendor docs. This document tells you what to build, what order to build it in, and what the failure modes look like in 2026.

02

It assumes Shopify.

The architecture works on WooCommerce or BigCommerce too, but the specifics around checkout extensibility, theme app extensions, the post-2025 Additional Scripts audit, and the June 2026 Shopify Scripts cutover are Shopify-specific. Adapt accordingly.

03

Every layer has a fixed shape.

Each of the eight layers gets two pages. A summary on the opener (what it is, what it costs you when it breaks). A tactic page with symptom, why it bleeds, five minute check, the fix, and an illustration of the pattern.

04

Page 21 is the cheat sheet.

The dedup and event-ID reference table on page 21 is the single highest-value page in this document. If your engineering team only reads one page, point them there.

Every Shopify brand I audit has a tracking problem before it has a media problem.

The two compound. Bad data leads to bad bids. Bad bids lead to wasted spend. The founder concludes paid is broken and either pulls budget or hires a new agency. The new agency runs ads against the same broken tracking and the cycle continues.

The fix is not a better media buyer. The fix is the architecture below. Once it is in, paid spend earns cleaner signal, campaigns optimize against the right action, and the Monday report tells the founder what is happening instead of what the platform wants them to believe.

Three rules that compress everything in this document:

01 **The server-side container is the spine.** Every platform downstream reads from it. If your container is dirty, every platform is dirty.

02 **Every event has one canonical ID across every platform.** Without dedup contracts you double-count and the algorithms inflate spend chasing ghosts.

03 **Reporting is a discipline, not a tool.** The Monday summary that gets read beats the dashboard nobody opens.

THE THREE LAYERS MOST OPERATORS SKIP

Fix **03 (server-side container)**, **07 (Klaviyo identity loop)**, and the post-2025 audit of **01 (Shopify Web Pixel API migration)**. The other five compound on top of these.

The store is the source of truth.

Shopify (or whatever platform) emits the canonical events: orders, refunds, line items, customer identifiers. Everything downstream is a copy. If the store is wrong, every downstream platform is wrong. This is also where the 2025 Additional Scripts deprecation already broke half the purchase pixels in the wild, and where the June 2026 Shopify Scripts cutover is about to take a second slice.

THE 2025 DEPRECATION ALREADY HAPPENED

Shopify shut down Additional Scripts and checkout.liquid on the Thank you and Order status pages on **August 28, 2025**. Automatic upgrades started January 2026. If your purchase pixel was in Additional Scripts and the migration has not been audited since, it is most likely already broken or firing wrong. The next deadline on the calendar is **June 30, 2026**, when Shopify Scripts (cart-level discount logic, a separate product) goes.

Get off Additional Scripts. Move to Web Pixel API.

SHOPIFY · CHECKOUT EXTENSIBILITY

THE STATE OF PLAY

Your purchase pixel lived in Additional Scripts before August 2025. Shopify either auto-upgraded the store in early 2026, or the team migrated late. Either way, the events now fire inconsistently or not at all. Paid platforms see broken or partial purchase events, Smart Bidding loses signal, conversion lift on Meta evaporates, and most teams notice the declining ROAS before they notice the cause. This page is about auditing the aftermath, not preparing for a future deadline.

FIVE MINUTE CHECK

Shopify admin → Settings → Customer events

Open both tabs (App pixels and Custom pixels). Confirm every pixel that should be firing is listed and shows status "Connected". Anything missing from the list is broken. Anything marked "Disconnected" is broken.

THE FIX

Audit the post-migration state and rebuild what broke.

01 In Shopify admin under Settings → Customer events, list every pixel across both tabs.

02 For the last 30 days: pull conversion counts from Google Ads (Tools → Conversions), Meta (Events Manager), and GA4. Compare each against Shopify Orders for the same window. A gap of more than 10 percent is the leak.

03 For each platform with a leak: rebuild the pixel as a Custom Pixel (Web Pixel API). Verify with a test order in the Shopify sandbox before going live.

04 Run old and new in parallel for two weeks where possible. Reconcile counts. Then kill the old.

ILLUSTRATION, NOT A CLIENT RESULT

Picture a Shopify store auto-migrated to the new checkout. Most purchase pixels survive, but one server-side event fires on test orders and not real ones because the new Web Pixel context hands over the customer email in a different shape than the old code expected. ***The fix is rebuilding that event against the new API and reconciling the months of partial data.***

GTM web does browser-side only.

The browser GTM container fires events client-side. Pixel fires, page view, scroll depth, button click. Most teams have GTM web. Most teams stop there. The miss is treating GTM web as the only tag manager. Every platform that matters (Google Ads, Meta, GA4) should be receiving from the server container, not the browser, with the browser as a deduped backup.

THE PATTERN

Browser GTM owns: page views, scroll, click, view-content, add-to-cart, lead form fires. Server container owns: purchase, lead, refund, value-driving conversions. The split keeps the heavy lifting on the server where you can see it and own it.

Split the work between web and server.

GTM WEB CONTAINER

SYMPTOM

Your GTM web container has 80+ tags. Every conversion fires from the browser. Server-side container does not exist or is misconfigured.

WHY IT BLEEDS

iOS Safari ITP and ad blockers eat browser pixel data. Conversions get under-reported, and how much depends on traffic mix. Smart Bidding optimizes against a fictional, smaller audience.

FIVE MINUTE CHECK

GTM web container → Tags

```
If your purchase pixel is firing as a
browser tag with no server-side
equivalent, this layer is
misconfigured.
```

THE FIX

Move conversions to server. Keep behavioral on browser.

01 Stand up the server-side container (Layer 03).

02 Move purchase, lead, refund, and value-driving events to fire from the server.

03 Keep the browser tags firing in parallel with deduplication via event_id.

04 After two weeks, compare server vs browser counts. The server number is the truth.

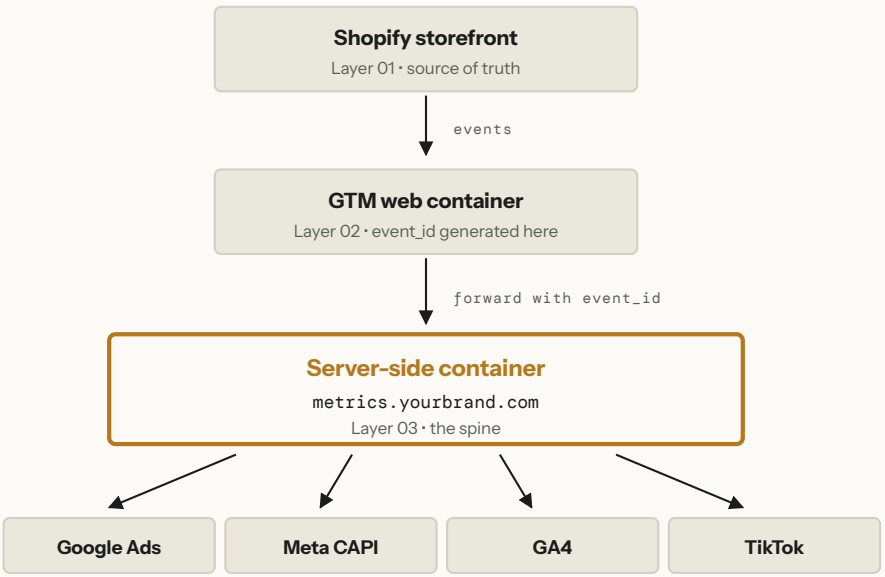
ILLUSTRATION, NOT A CLIENT RESULT

Picture a store whose purchase pixel fires only in the browser. *Ad blockers and Safari's tracking limits drop some of those events, so Google Ads records fewer orders than Shopify holds. Moving purchase events server-side closes that gap. Same orders, counted more completely.*

The server container is the spine.

The server-side container is where every platform downstream reads from. It is the first-party proxy. It is the only layer in this entire architecture that you cannot skip and still have a working stack in 2026.

HOW DATA FLOWS THROUGH THE SPINE



PICK ONE AND COMMIT

Stape ($120 to $300/mo, fastest setup, Shopify-friendly), **Google Cloud Platform GTM SS** ($250 to $500/mo, full control, requires devops), or **Elevar managed** ($300+/mo, white-glove, opinionated). Don't try to roll your own server endpoint. The maintenance burden is not worth saving $200 a month.

Put the container on a first-party custom subdomain.

STAPE · GCP · ELEVAR

SYMPTOM

Server container exists but lives on the vendor's default subdomain (e.g. .stape.io or .gcp-something). Cookies set there are third-party, dropped by Safari, and the whole point of server-side is wasted.

WHY IT BLEEDS

The reason server-side beats browser is the first-party cookie context. If your container is on a vendor subdomain, browsers treat it as third-party and you lose much of the signal you thought you were capturing.

FIVE MINUTE CHECK

```
Browser → DevTools → Network tab
Find your sGTM endpoint. If the
domain is not your own (e.g.
data.yourbrand.com), this layer is
misconfigured.
```

THE FIX

CNAME a subdomain to the server endpoint.

01 Pick a neutral subdomain like data.yourbrand.com. Skip gtm, sgtm, tracking, analytics, and metrics. Ad blockers target those words by name and undo the first-party gain you came for.

02 CNAME it to your server container endpoint (Stape / GCP / Elevar).

03 Update your GTM web tags to send to the new subdomain.

04 Verify in browser DevTools that requests now go to your domain.

ILLUSTRATION, NOT A CLIENT RESULT

Picture a store with a server-side container set up correctly but running on the vendor's default subdomain. ***Browsers treat that as third-party traffic and block more of it. Pointing a CNAME at a first-party subdomain is a short DNS change that lets more events through.***

GA4 is for behavioral analytics, not attribution.

GA4 is for behavioral cohorts: which pages lead to add-to-cart, which campaigns produce sessions that convert, which products correlate with returning customers. It is not the place to grade ad performance against Shopify revenue. Those numbers will never agree, because GA4 sessionizes and Shopify counts orders. Wire GA4 cleanly, then judge ad performance from inside the ad platforms.

THE MISMATCH NOBODY FIXES

Founders ask "why don't GA4 revenue and Shopify revenue match?" They never will. GA4 attributes by session and dedupes its own way. Shopify sums order totals. The right answer is to stop comparing them and use each for its actual purpose.

Wire GA4 for behavior. Wire Google Ads for conversion.

THE MISMATCH AND WHY IT BLEEDS

Google Ads is importing GA4 conversions. The two systems show different numbers and the founder asks every Monday which one to trust. That is the symptom and the cause in one sentence. GA4-imported conversions lag, and GA4 credits them by its own attribution rules rather than Google Ads' rules, so they usually come in lower than a direct conversion API import, so Smart Bidding optimizes against incomplete signal, CPA looks worse than it is, and you scale slower than you should.

FIVE MINUTE CHECK

Google Ads → Goals → Conversions

If your primary purchase conversion is sourced from "Google Analytics 4", this layer needs work. Server-side conversion API should be primary; GA4 is a backup.

HONEST READ

Comparing GA4 revenue to Shopify revenue is the wrong question. Compare GA4 paid sessions to Shopify paid-source orders, then judge each ad platform from inside its own dashboard. GA4 was built to read behavior across the funnel, not to grade Meta against Google.

THE FIX

Server CAPI is primary, GA4 is secondary, and the GA4-Ads link feeds audiences only.

01 Set up a server-side purchase conversion in Google Ads via the conversion API.

02 Mark it as primary. Demote the GA4-imported version to secondary.

03 Keep the GA4-Ads link, but use it for audience export only, not conversions.

04 After 14 days, check that the new primary conversion count is at least as high as the GA4 number was. If it is lower, check the server tag before going further.

ILLUSTRATION, NOT A CLIENT RESULT

Picture a lead-gen account importing conversions from GA4 only. *Anything GA4 loses to Safari or an ad blocker never reaches Google Ads. Sending the lead server-side gives bidding the same leads, collected at a layer the browser can't block.*

Google Ads needs enhanced conversions on every action.

Server-side conversion API is the primary input. Enhanced Conversions is the multiplier. Together they recover conversions that would otherwise be lost to cookie loss, ITP, and consent denials. Without both, your Smart Bidding is making decisions on a fraction of the actual signal.

CONSENT MODE V2 + EEA

If you take any traffic from the EU or UK, Consent Mode v2 is now required. Without it Google throttles delivery and reporting. Implementation is one config change in your CMP (Cookiebot, OneTrust, Consentmanager) plus one parameter in your GTM container. Don't skip it because you "barely take EU traffic."

Enhanced Conversions on every action. Conversion value as net revenue.

GOOGLE ADS · CONVERSIONS

SYMPTOM

Enhanced Conversions is "enabled" in Google Ads but firing inconsistently. Conversion value is set to gross revenue including shipping and discounts. Every conversion action is set to primary.

WHY IT BLEEDS

Smart Bidding optimizes against inflated revenue numbers. tROAS targets get hit on paper but real margin is below the line. Multiple primary conversions confuse the algorithm and split bid signal.

FIVE MINUTE CHECK

Google Ads → Goals → Conversions → Diagnostics

If "Enhanced Conversions" diagnostic shows anything other than green across the board, it is not actually firing. Also: count your primary actions. There should be exactly one purchase conversion as primary.

THE FIX

One primary purchase, enhanced on every action, valued at net revenue.

01 Enable Enhanced Conversions on every conversion action via the Google Ads tag, not GTM.

02 Set conversion value to gross revenue MINUS discounts and shipping. Net to the brand.

03 Pick one primary purchase conversion. Demote everything else (add-to-cart, view-content, etc) to secondary.

04 Run for 14 days. Check the EC diagnostic stays green.

ILLUSTRATION, NOT A CLIENT RESULT

Picture a Shopify store bidding on tROAS with conversion value set to order subtotal including shipping. *The algorithm chases high-shipping orders that look big and earn little. Switching the value to net revenue points it back at margin.*

Meta runs on the same spine. TikTok and Pinterest too.

Conversions API on every event. Browser pixel fires the same events with an event_id so Meta can dedupe them. Without dedup, Meta sees double conversions and inflates ROAS. The same architecture extends to TikTok Events API and Pinterest Conversions API. One server container, three platform fan-outs, deduped end to end.

ATTRIBUTION MODEL SELECTION

Meta default is 7d-click 1d-view. That is the right setting for most DTC brands. For high-consideration verticals (furniture, B2B, real estate), bump to 7d-click 7d-view and watch engaged-view as the secondary read. Don't overthink it. The setting matters less than the dedup contract underneath.

Browser + CAPI sharing event_id. Hashed identity in every server event.

META CAPI · TIKTOK · PINTEREST

SYMPTOM

Meta reports a 6.0 ROAS. Shopify reports total revenue ~30 percent below what would match. The two never reconcile. Founder thinks Meta is lying.

WHY IT BLEEDS

Without a shared event_id between browser pixel and CAPI, Meta double-counts purchases. Reported ROAS inflates. You scale spend chasing fake revenue and burn budget on audiences that don't actually convert.

FIVE MINUTE CHECK

```
Events Manager → Test Events
Trigger a test purchase. Look for "Browser + Server" in the deduplication column. If the events are firing as separate, dedup is broken.
```

THE FIX

Share event_id, hash PII in CAPI and run quarterly lift studies.

01 Generate a unique event_id per event in your sGTM container. Pass to both browser pixel and CAPI.

02 Include hashed email and hashed phone on every server-side purchase event.

03 Verify dedup is working in Events Manager Test Events tool.

04 Run a Conversion Lift study quarterly to validate incrementality once spend is above $20K/mo.

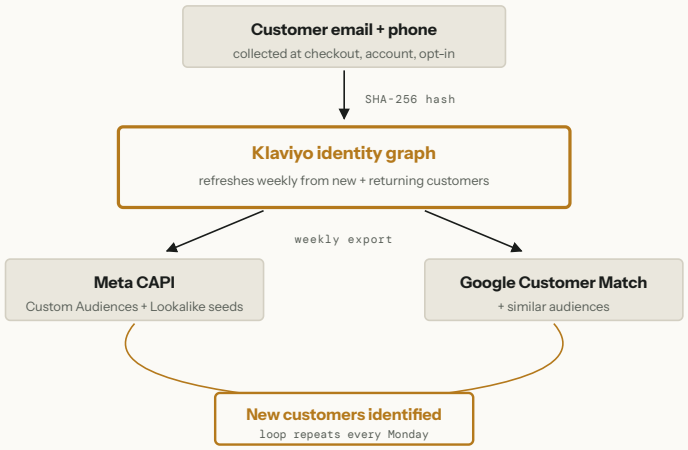
ILLUSTRATION, NOT A CLIENT RESULT

Picture a brand whose Meta ROAS looks far better than the store's books because purchases are counted twice. ***Fixing deduplication doesn't make Meta perform worse. It makes the reporting honest, so you can cut the audiences that lose money and move budget to the ones that work.***

Klaviyo is your identity graph, not your send engine.

Email and SMS platforms are not send engines. They are the cleanest first-party identity graph you have. Hashed identity feeds back into Meta CAPI and Google Customer Match. The most underused piece of the stack at most brands I audit.

THE WEEKLY IDENTITY REFRESH



THE COMPOUNDING PLAY

Wiring Klaviyo identity into Meta CAPI and Google enhanced conversions gives both platforms more purchases they can match to a person, so bidding learns from more of your real customers. The gain comes from first-party identity, not creative or audience changes.

Pipe Klaviyo identity to Meta + Google. Refresh weekly.

KLAVIYO · POSTSCRIPT · CUSTOMER MATCH

SYMPTOM

Klaviyo is connected to Shopify and sending email campaigns. The customer list never gets exported to Meta or Google. Customer Match lists in Google Ads are six months old.

WHY IT BLEEDS

You are sitting on the highest-quality first-party data in your business and not feeding it back into the platforms that need it most. Without fresh customer match, your lookalike audiences degrade. Without Klaviyo email synced to Meta CAPI, your conversion match rate is lower than it could be.

FIVE MINUTE CHECK

Klaviyo → Integrations

If Meta and Google Ads are not in your active integrations list, this layer is missed. If they are connected but the list size in Google Customer Match hasn't changed in a month, the sync is stale.

THE FIX

Keep the integrations active, sync weekly and hash identity in every server event.

01 Connect Klaviyo to Meta and Google Ads as active integrations.

02 Set customer match lists to refresh at least weekly.

03 Push hashed email and phone into your server-side container so every CAPI event includes identity.

04 Build Klaviyo segments (high-LTV, lapsed, abandoners) and sync them as Custom Audiences for Meta exclusion / inclusion targeting.

ILLUSTRATION, NOT A CLIENT RESULT

Picture a brand with Klaviyo set up well for email and no connection to Meta or Google. *Wiring that customer identity through gives the ad platforms cleaner data to match and optimize against.*

Reporting is a discipline, not a tool.

A real-time dashboard nobody opens is worse than a written Monday summary that gets read. Pull the numbers that matter (spend, revenue, ROAS, new vs returning, top campaigns). Write one paragraph each on what changed and why. Ship it before the founder's first Monday meeting. The format matters less than the rhythm.

THE MONDAY RULE

Every Monday by 9am the founder gets one written summary covering: total spend last week, total revenue, blended ROAS, what went up, what went down, what changed in the account, what's being tested this week. That is the entire reporting layer. Looker Studio is for ad-hoc questions, not weekly reads.

Written Monday summary. Looker Studio for ad-hoc. Quarterly re-plan.

REPORTING CADENCE

SYMPTOM

The agency built a beautiful Looker Studio dashboard. Founder hasn't opened it in three weeks. There is no weekly written summary. Every Monday call starts with "what happened last week."

WHY IT BLEEDS

Decisions get delayed because the data isn't packaged for the founder's actual workflow. Tests run too long because nobody is monitoring them in narrative form. The brand stays "in the data" without ever stepping back to make the strategic call.

FIVE MINUTE CHECK

Founder's inbox → Last Monday

If there is no written summary email from the previous Monday outlining spend, revenue, ROAS, and what changed, this layer is missed.

THE FIX

Send the Monday summary email, re-plan quarterly and keep a dashboard for ad-hoc questions.

01 Write a Monday summary email. Spend, revenue, ROAS, new vs returning, top campaigns, what changed and why.

02 Keep it under 300 words. The founder reads it on their phone before the first meeting.

03 Looker Studio (or equivalent) lives for ad-hoc questions. "Pull last quarter's spend by campaign type for me." That is what dashboards are for.

04 Quarterly: re-plan against the original 90-day plan. Did the spend target get hit? The ROAS target? The new-customer target? What changes for next quarter?

A SAMPLE MONDAY SUMMARY, IN FULL

An illustration of the format, for a brand running ~$30K/mo in paid:

- Spend: $32,400 (vs $30,000 plan, +8%)
- Revenue: $145,200 (vs $120,000 plan, +21%)
- Blended ROAS: 4.48 (above target 4.0)
- New customers: 312 (vs returning 580, ratio holding at 35/65)
- Top campaign: PMax-Bedroom, ROAS 5.2, scale candidate
- Watch: branded search CPC up 14% week-over-week

Event dedup contract, by platform.

If your engineering team only reads one page in this document, point them here. The dedup key per platform. Use these IDs consistently across browser and server events.

PLATFORM	DEDUP KEY	NOTES
Meta	`event_id`	Generate per-event in sGTM. Pass to both browser pixel and CAPI. Verify in Events Manager Test Events.
Google Ads	`transaction_id`	Order ID is the dedup key; gclid / wbraid / gbraid is the attribution key. Don't conflate them. The same order should never fire twice with the same transaction_id, regardless of which click identifier carried it.
GA4	`transaction_id`	Standard ecommerce purchase event. Order ID is canonical. Same event from server and browser dedupes by ID.
TikTok	`event_id`	Same pattern as Meta. Pass event_id from sGTM to both Pixel and Events API.
Pinterest	`event_id`	Same pattern as Meta. Pinterest Conversions API has been GA since 2022; dedup behavior matches.
Microsoft Ads	`UET tag conversion ID`	UET fires browser-side. Server-side via Microsoft CAPI uses the same conversion ID for dedup.
Klaviyo	`unique_id`	Klaviyo's Events API dedupes Track events by the unique_id field on the payload. Use it (not a custom property) to prevent the same event from posting twice when Shopify-native and server-tracked sources both fire.

THE PATTERN THAT WORKS ACROSS EVERY PLATFORM

Generate one event_id per real-world event in your server-side container. Pass it to every platform you fire to (browser and server). Each platform handles dedup using that ID as the canonical reference. Generate the ID once, use it everywhere.

What this PDF does not cover.

Three things sit one layer deeper than this stack. They are real and worth doing eventually. They are not in scope here because the eight layers above need to be solid first.

Customer data platforms (Segment, RudderStack)

If you are at $5M+/year and have multiple data destinations (CRM, warehouse, support tools, multiple marketing platforms), a CDP becomes the right answer. Below that scale, the server-side container is the CDP. Don't add a layer you don't need.

Warehouse-level analytics (BigQuery + dbt)

The right answer when you outgrow Looker Studio and need cross-source modeling at scale. Wait until you have the data engineering capacity to maintain it. Otherwise it's a graveyard of half-finished pipelines.

Marketing mix modeling (MMM)

The right answer above $500K/mo in spend. Below that, attribution model selection inside Google Ads and Meta is sufficient. MMM at smaller scales is statistically noisy and produces decisions you would have made anyway.

Send the three above to a CDP-specialist consultancy or a data engineering firm. They are not the eight layers in this doc. Don't try to bolt them on before the eight layers below are solid.

If you'd rather not rebuild this alone.

I rebuild this exact stack for a small number of Shopify brands each quarter. Not a quote. Not a discovery deck. A working tracking architecture inside your accounts.

- **A ranked, screenshotted list** of every layer in your current stack and what's broken.
- **The exact migration path** from your current stack to the eight layers above, in priority order.
- **A 30 day rebuild plan** with task owners (yours or mine) and rough ETAs.
- **Honest take** on whether you should DIY, hire your dev team, or bring me in.

Send me read-only access for the free written audit.

5 TO 7 PAGES, WRITTEN BY ME, BACK WITHIN 24 HOURS OF ACCESS. NO CALL NEEDED.

CONNERCROWE.COM/AUDIT-REQUEST →

Glossary.

If you handed this PDF to your developer or your in-house tracking lead and one of these landed unfamiliar, the page is here.

CAPI

Conversions API. Server-side endpoint for sending event data to Meta, TikTok, Pinterest. Replaces / complements browser pixel.

CMP

Consent Management Platform. Cookiebot, OneTrust, Consentmanager. Required for EEA traffic.

CMv2

Consent Mode v2. Google's framework for adjusting tag behavior based on user consent. Required as of 2024 for EEA traffic to Google Ads.

EC

Enhanced Conversions. Google Ads feature that hashes user-provided data (email, phone) for higher conversion match rates.

event_id

Unique identifier per event used to dedupe between browser pixel and server-side CAPI. Generate in sGTM.

gclid

Google Click Identifier. URL parameter Google appends to ad clicks for tracking.

GTM

Google Tag Manager. Browser tag manager. The web container.

GTM SS / sGTM

Google Tag Manager Server-Side. The server container that proxies data to platforms.

ITP

Intelligent Tracking Prevention. Safari's privacy framework that limits cross-site cookies and breaks browser-side tracking.

MMM

Marketing Mix Modeling. Statistical modeling of marketing spend impact on revenue. Higher-order analytics.

tCPA / tROAS

Target Cost Per Acquisition / Target Return On Ad Spend. Smart Bidding strategies in Google Ads.

UET

Universal Event Tracking. Microsoft Ads tracking tag (Bing).

Bad tracking makes every dollar weaker. Clean data is the spine.

More spend doesn't fix bad data. A new agency doesn't either. The eight layers above are the architecture every Shopify brand at $5K to $200K per month needs to have right before scaling paid acquisition. The brands that win the next two years are the ones that fix the spine first.

© CONNER CROWE 2026 · All rights reserved.