CHANDAN KUMAR
Founder, Global Info Edge

45

Tracking Problems Solved

Claude Prompts You Decode

Claude + Google Tag Manager. Read the container, write the spec, build in a workspace, publish only when you say so.

Fix tracking in minutes, not hours — without handing over the keys.

linkedin.com/in/ckmehta · @ckmehta · ckumar.in

What's inside

Eight clusters. Forty-five prompts. Each one written to be pasted into Claude with a GTM connection live — and each one built so nothing goes live without you approving it.

#	Cluster	Prompts
01	Container Audit & Cleanup	7
02	GA4 Foundations	6
03	Ecommerce & Purchase Tracking	7
04	Lead Generation & Form Tracking	5
05	Google Ads Conversions	5
06	Meta, LinkedIn & Other Platforms	5
07	Consent, Privacy & Server-Side	5
08	Debugging, QA & Governance	5

Why this exists

There is a post going around saying Claude plus Google Tag Manager means you can stop doing tracking work by hand. The claim is broadly right and the detail matters enormously — because what you are being invited to do is give an AI write access to the system that measures your revenue.

This book gives you the forty-five prompts. It also tells you exactly how the connection works, what it can and cannot do, and the four rules that keep it from becoming an incident.

How this actually works

The short version of the viral post is: tell Claude what you want to track, and it builds the tags, triggers and variables for you. That is true. The part usually left out is **how** the connection is made, and it changes what you should agree to.

There is no official Google connector

As of this writing, Google does not publish an official Google Tag Manager MCP server. The connection is made through **third-party MCP servers** built on Google's Tag Manager API v2 — several exist, some hosted, some run locally, some open source.

That matters for three reasons. First, you are choosing a vendor, and that vendor sits between Claude and your container. Second, the capabilities differ between them — some are read-only, some can publish. Third, you authenticate with your own Google account via OAuth, which means the server acts with **your** GTM permissions.

> ### Before you connect anything
>
> Check what the MCP server you have chosen actually does — read-only or read-write, hosted or local, who operates it, and whether it stores your tokens. Then check what GTM permission level the Google account you authenticate with holds. If that account can publish to a production container, so can anything connected to it.

The safe sequence

Every prompt in this book follows the same order, and it is not decoration. It is the sequence that keeps a mistake recoverable.

Step	What happens	Who decides
1. Read	Claude reads the live version and reports what is there. Nothing changes.	Nobody — it is read-only
2. Draft	Claude builds tags, triggers and variables in a new workspace.	You approve the plan
3. Version	Changes are packaged into a version. Still not live.	You review the diff
4. QA	You test in Preview mode against a checklist.	You

Step	What happens	Who decides
5. Publish	The version goes live.	You, explicitly

Read the live version first, always. Auditing a workspace tells you what someone intended. Auditing the live version tells you what is actually running on your website right now, which is the only thing that affects your numbers.

The four rules

1. **Read before you write.** Every audit prompt in this book ends with an instruction not to create, update, delete, version or publish. Keep it there. If Claude misidentifies the container, a read-only instruction turns a disaster into a wasted minute.
2. **Never publish from a prompt.** Claude drafts in a workspace and stops. You look at the diff, you run the QA checklist, you publish. That line does not move, regardless of how confident the output sounds.
3. **Record the live version number before you start.** It is your rollback point. Write it down before the first change, not after something breaks.
4. **Treat deletions as a separate, slower decision.** A tag nobody recognises is not necessarily a dead tag. Pause it for two weeks, watch what breaks, then delete. Prompt 07 is built around exactly this.

What Claude is genuinely good at here

Reading a container of two hundred tags and telling you which fourteen matter. Writing a dataLayer specification a developer can implement without asking follow-up questions. Explaining what changed between two versions when something broke last Tuesday. Producing the QA checklist you would have written if you had the time.

What it is not good at: knowing that the tag with the strange name was added for a legal requirement in 2023. That context lives with you, which is why the human stays in the sequence.

Set it up once

Fifteen minutes of setup, and the forty-five prompts become a system rather than a document you read once.

1. **Choose and connect a GTM MCP server.** Several third-party options exist; pick one, read what it does, and connect it in Claude. Start with one that is read-only if you are new to this.
2. **Authenticate with a scoped Google account.** Not the account that owns everything. Give it the minimum GTM permission the task needs — and for your first week, give it read access only.
3. **Create a Project in Claude, one per container.** Put your context in the Project instructions: the site, the platforms you use, your naming conventions, your consent setup, and the domains involved.
4. **Record the live version number** in the Project instructions. Update it after every publish. This is your rollback point and it takes five seconds to maintain.
5. **Run prompt 01 first.** It reads the live container and tells you which cluster to run next.

Six habits that make these work

1. **Confirm the target before every write.** Ask Claude to repeat the account ID, container ID and container name back to you. Wrong-container changes are the expensive mistake here, and they are entirely avoidable.
2. **Ask for the reasoning, not just the configuration.** "Why this trigger type rather than the others" catches errors that a finished tag never reveals.
3. **Make it tell you what it cannot see.** The container does not contain your business context. A prompt that ends with "list what you could not determine from the container alone" is worth more than one that produces a confident complete answer.
4. **Test the negative case.** Not just does the tag fire — does it fail to fire when it should. A form tag that fires on validation errors is worse than no form tracking, because you will believe it.
5. **One change set at a time.** Publish, watch for 48 hours, then the next. Bundled publishes make it impossible to know which change caused the problem.
6. **Write down what you changed and why.** Then in a fortnight, paste that back in and ask what actually happened to the numbers. That is how the container gets a memory.

CLUSTER 01 · 7 PROMPTS

Container Audit & Cleanup

Before you add anything, find out what is already in there. Most containers carry years of tags nobody can account for.

01 The Live Container Audit

Use when *First thing on any container you did not build yourself.*

You'll need Read access to the GTM account. Have the container name ready.

> Read my live GTM container and tell me what is actually published.
>
> List every tag, trigger and variable in the live version — not the workspace. For each tag: what it does, what fires it, and whether it looks current.
>
> Then give me: tags that appear broken or misconfigured; tags referencing platforms we may no longer use; triggers that can never fire; and anything with a naming convention that suggests it was a one-off nobody cleaned up.
>
> Read only. Do not create, update, delete, version or publish anything. Confirm the account ID, container ID and live version number back to me before you start.

02 Duplicate & Overlapping Tag Finder

Use when *When conversions look inflated, or two people have had edit access.*

You'll need Read access. Run the live audit first.

> Find where this container is double-counting.
>
> Identify: two or more tags sending the same event to the same destination; a platform receiving the same conversion from both a GTM tag and a hardcoded on-page script; and any tag firing on a trigger broad enough to catch the same action twice.
>
> For each, tell me which one to keep and why, and what the likely inflation is on the affected metric.
>
> Read only for now. Give me the findings as a table before you propose any change.

03 Paused, Orphaned & Zombie Tag Sweep

Use when *Quarterly. Containers accumulate; nobody ever removes.*

You'll need Read access, plus the list of platforms you actually use today.

> Find everything in this container that no longer earns its place.
>
> Categorise: tags that are paused (and how long they have been); tags with no trigger attached; triggers with no tag attached; variables referenced by nothing; and tags pointing at tools we no longer pay for — I'm pasting our current tool list below.
>
> For each item, state your confidence that it is genuinely dead. Anything you're unsure about goes in a separate 'verify before removing' list rather than the deletion list. I would rather review twenty items than lose one that mattered.

04 Trigger Logic Review

Use when *When tags fire in the wrong places, or don't fire where they should.*

You'll need Read access.

> Review every trigger in this container for logic errors.
>
> Check for: page path conditions that use 'contains' where they should use 'equals'; regex that is broader than intended; triggers that depend on a variable that may be undefined on some pages; and 'All Pages' triggers on tags that should be scoped.
>
> For each problem, show me the current condition, what it actually matches, and what it should be.
>
> Flag specifically any trigger where a small site change — a new URL pattern, a renamed button — would silently break it. That is the failure mode nobody notices for three months.

05 Variable Inventory & Naming Audit

Use when *Before you hand the container to anyone else.*

You'll need Read access.

> Inventory every variable and tell me whether this container is maintainable.
>
> For each variable: type, what it reads, and which tags or triggers depend on it. Flag variables with near-identical purposes that could be consolidated.
>
> Then assess the naming convention. Propose a consistent scheme — type prefix, platform, purpose — and give me the rename map as a table.

> Do not rename anything yet. Renaming variables breaks references, so I want to see the full dependency chain for each one before we touch it.

06 Container Bloat & Load Impact

Use when *When Core Web Vitals are poor and marketing tags are a suspect.*

You'll need Read access, plus your PageSpeed or Core Web Vitals report.

> Tell me what this container is costing my page speed.
>
> List every tag that loads a third-party script, grouped by vendor, and note which fire on All Pages versus specific triggers. Identify tags loading synchronously or early that could be deferred.
>
> Cross-reference against the Core Web Vitals data I'm pasting below and tell me which tags are the most likely contributors.
>
> Be honest about the limits here — you can identify candidates from the container, but you cannot measure their actual impact without a waterfall. Tell me which ones are worth testing rather than presenting a ranking as fact.

07 The Staged Cleanup Plan

Use when *After the audit, before you delete anything.*

You'll need The output of prompts 01–05.

> Turn these findings into a cleanup plan I can execute without breaking reporting.
>
> Sequence it in three stages: (1) pause and observe — things I'm confident are dead, paused for two weeks first; (2) remove — items confirmed dead after observation; (3) rebuild — anything broken that needs replacing rather than deleting.
>
> For each item give me: the action, the risk if we're wrong, and what to watch to know if we were.
>
> Create this as a plan in a new workspace only. Do not create a version and do not publish. I will review every item before anything goes live.

CLUSTER 02 · 6 PROMPTS

GA4 Foundations

The layer everything else reports through. Get the naming and the configuration wrong here and every downstream number inherits the mistake.

08 GA4 Configuration Check

Use when *New container, or any account where GA4 numbers look implausible.*

You'll need Read access, plus your GA4 measurement ID.

> Check my GA4 setup in this container against how it should be configured.
>
> Verify: the configuration tag exists and fires on all pages before any event tags; the measurement ID is correct and consistent; event tags reference the configuration correctly; and there is no duplicate GA4 configuration from a hardcoded gtag script.
>
> Then check the firing order — event tags firing before configuration is a common and quiet failure.
>
> Read only. Give me findings with severity, and tell me which single fix would change my reported numbers most.

09 Event Naming Convention Builder

Use when *Before you create your fifth event. Ideally before your first.*

You'll need Your list of actions you want to track, and any existing event names.

> Design a GA4 event naming convention for this business and apply it to my existing events.
>
> Rules to follow: use GA4 recommended event names wherever one exists, because they unlock built-in reporting. Use snake_case for custom events. Name the action, not the implementation — 'generate_lead', not 'form_submit_v2_new'.
>
> Give me: the convention written as a one-page rule; a rename map for my existing events; and a list of events that should be consolidated because they measure the same thing.
>
> Flag any rename that would break historical reporting continuity, and say what I'd lose.

10 Recommended vs Custom Event Map

Use when *When you're deciding what to build and want the free reporting.*

You'll need Your business model and the list of user actions that matter.

> Map my business actions to GA4 events.
>
> For each action I list below, tell me: whether a GA4 recommended event covers it (and which), or whether it needs a custom event. Where a recommended event exists, use it — custom events lose the built-in reports and I'd rather not pay that price for a nicer name.
>
> For each, specify the required and recommended parameters.
>
> Then flag anything I've asked to track that probably isn't worth an event, because a container with sixty events nobody looks at is worse than one with twelve that get used.

11 Cross-Domain & Referral Exclusion Setup

Use when *Any time a user crosses domains — checkout, payment gateway, booking engine.*

You'll need Read access, plus the full list of domains and payment gateways in your funnel.

> Set up cross-domain measurement properly for these domains.
>
> From the domain list below: configure cross-domain linking in the GA4 configuration, and identify every payment gateway or third-party host that needs a referral exclusion — otherwise the gateway gets credited for my conversions and my paid channels get robbed.
>
> Tell me exactly which domains go in which setting, and why they are different things. People confuse these two constantly.
>
> Build it in a new workspace. Do not create a version yet. Then give me the test plan to confirm session continuity across the handoff.

12 Internal Traffic & Bot Filtering

Use when *Small-to-mid traffic sites where the team's own visits distort the data.*

You'll need Read access, plus your office IPs and team device details.

> Keep our own traffic out of the reporting.
>
> Recommend the approach: GA4 internal traffic filters, a GTM-side exclusion, or both — and explain the trade-off, because a GTM-side block means the data never arrives and cannot be recovered.
>
> Set up what we agree on using the IPs below. Include a way to identify internal traffic for staff

on mobile networks, where IP filtering fails.

Then tell me how to verify it is working, and what share of my current traffic is likely internal based on anything visible in the container.

13 GA4 Parameter & Custom Dimension Plan

Use when *When you can see that something happened but not who it happened to.*

You'll need Your reporting questions — the things you wish you could segment by.

Design the parameter and custom dimension layer that answers my reporting questions.

For each question below, tell me what parameter would answer it, on which event it belongs, and whether it should be an event-scoped or user-scoped custom dimension.

Respect the limits — GA4 caps custom dimensions, so tell me which of my questions are worth a dimension and which are not. Rank them.

Then give me the dataLayer variables needed in GTM, and the exact spec to hand a developer for anything the site does not currently push.

CLUSTER 03 · 7 PROMPTS

Ecommerce & Purchase Tracking

Where the money is measured, and therefore where a mistake costs most. Every bidding decision downstream inherits whatever this layer reports.

14 Purchase Tracking End-to-End

Use when *The prompt from the headline. Set up purchase tracking from a checkout flow.*

You'll need Read access, your checkout URL structure, and a description of the confirmation page.

> Set up purchase tracking end to end for this checkout flow.
>
> I need: the dataLayer specification the developer must implement on the confirmation page; the GTM variables to read it; the trigger; and the GA4 purchase event tag with all required parameters — transaction_id, value, currency, items array, tax, shipping.
>
> Build it in a new workspace. Create a version but do not publish.
>
> Then give me a QA checklist: what to check in Preview mode, what should appear in the GA4 DebugView, and the three most likely reasons it will fire but report zero revenue.

15 DataLayer Spec for Developers

Use when *When the site does not push the data and you need engineering time.*

You'll need The events you need and what data each must carry.

> Write the dataLayer specification I can hand straight to a developer.
>
> For each event below: the exact push syntax with a realistic example object, every required key with its data type, where in the page lifecycle it must fire, and the edge cases the developer must handle — guest checkout, multi-currency, discounts, partial refunds.
>
> Write it so someone who has never used GTM can implement it without asking me follow-up questions. Include a 'common mistakes' section at the end.
>
> This is a document, not a container change. Do not modify GTM.

16 Full Ecommerce Funnel Events

Use when *When you have purchase tracking and nothing before it.*

You'll need Your product and checkout page structure.

> Build the complete ecommerce funnel, not just the purchase.
>
> Specify view_item_list, select_item, view_item, add_to_cart, remove_from_cart, view_cart, begin_checkout, add_shipping_info, add_payment_info and purchase — with the parameters each needs and the trigger for each.
>
> Prioritise them. If I can only implement four this quarter, tell me which four give the most diagnostic value and why.
>
> Build the tags in a new workspace, grouped in a folder. No version, no publish.

17 Duplicate Purchase Prevention

Use when *When revenue in the platform exceeds revenue in the bank.*

You'll need Read access, plus a description of what happens if a user refreshes the confirmation page.

> Find and fix duplicate purchase firing.
>
> Check: does the purchase tag fire again on confirmation page refresh or back-navigation? Is transaction_id populated on every fire, or empty in some cases? Is the same purchase reported by both a GTM tag and a platform-native script?
>
> Tell me what deduplication is currently in place and what is missing.
>
> Then implement a transaction_id-based guard in a new workspace and explain how it works, including what happens on a genuine repeat purchase by the same customer — that must still count.

18 Transaction ID & Revenue Accuracy

Use when *Before you trust a single ROAS number.*

You'll need Read access, plus a sample of backend orders for the same period.

> Check whether the revenue this container reports matches what the business actually took.
>
> Review how value, currency, tax and shipping are being passed. Flag the classic errors: revenue including tax when it should not, shipping counted as revenue, discounts not deducted, currency hardcoded on a multi-currency store, value passed as a string with a symbol in it.
>
> Compare the structure against the sample orders I'm pasting below and tell me where a

> discrepancy would come from.
>
> Rank the fixes by how much they distort reported revenue.

19 Refund & Cancellation Tracking

Use when *Any business with meaningful returns. Especially cash on delivery.*

You'll need Your refund process and where cancellations are recorded.

> Set up refund tracking so my reported revenue reflects reality.
>
> Specify the GA4 refund event — full and partial — with the parameters needed, and the mechanism to trigger it, given that refunds usually happen outside the website.
>
> Cover the options honestly: a dataLayer push from the admin system, the Measurement Protocol, or a scheduled adjustment. Tell me the trade-offs of each including the effort involved.
>
> Then tell me what this changes about the ROAS I'm currently reporting to the business. If returns are material, the current number is wrong and I need to know by how much.

20 Checkout Drop-off Instrumentation

Use when *When you know people leave and not where.*

You'll need Your checkout steps in order.

> Instrument this checkout so I can see exactly where people leave.
>
> For each step below, specify the event, the trigger, and the parameters that would let me segment drop-off by payment method, delivery option, device and whether the user was logged in.
>
> Also instrument the failure states — payment declined, address validation error, out-of-stock at checkout. Those are usually the biggest drop-off cause and almost nobody tracks them.
>
> Build in a new workspace, in a folder named Checkout Funnel. No publish.

CLUSTER 04 · 5 PROMPTS

Lead Generation & Form Tracking

Form tracking is where most lead-gen accounts quietly lie to themselves. This cluster makes the events fire on the thing that actually happened.

21 Form Tracking That Actually Fires

Use when *When your form conversions and your inbox disagree.*

You'll need Read access, plus the form URL and how it submits — AJAX, redirect, or embedded third party.

> Set up form tracking that fires on genuine submission, not on button click.
>
> First tell me which method is right for this form: GTM's form submission trigger, a dataLayer push from the developer, a success-message element visibility trigger, or a thank-you page trigger. Explain why the others are wrong here — this depends on how the form submits and getting it wrong is the single most common tracking error in lead generation.
>
> Then build it in a new workspace with the trigger and the GA4 generate_lead event.
>
> Include how to test that it does not fire on a failed or validation-blocked submission.

22 Lead Quality Signals in the DataLayer

Use when *When you want the bidding algorithm to learn what a good lead looks like.*

You'll need Your form fields and what distinguishes a good lead from a poor one.

> Design the lead quality signals to capture at submission.
>
> From the form fields below, tell me which are safe and useful to push to the dataLayer as parameters — budget band, service interest, company size, urgency, source — and which must never be pushed because they are personally identifiable.
>
> Be strict on that second list. Names, emails, phone numbers and addresses do not go into GA4.
>
> Then give me the dataLayer spec, the GTM variables, and how these parameters would later feed a lead-scoring model or an offline conversion import.

23 Call & WhatsApp Click Tracking

Use when *Any Indian business where a meaningful share of leads arrive by phone or WhatsApp.*

You'll need Read access, plus your tel: and wa.me link structure.

> Track phone and WhatsApp engagement properly.
>
> Set up click tracking on tel: links and wa.me links, separated by page location so I can see which pages drive calls. Include the sticky mobile call button if there is one.
>
> Be clear about the limitation up front: a click is not a call, and a WhatsApp click is not a conversation. Tell me the realistic gap and what I would need — call tracking numbers, WhatsApp Business API webhooks — to close it.
>
> Build the click tracking in a new workspace, and give me the honest caveat to put next to this number in reporting.

24 Multi-Step Form Progress Tracking

Use when *Long forms, quote builders, application flows.*

You'll need The steps in order and how step changes are signalled in the page.

> Instrument this multi-step form so I can see where people abandon.
>
> For each step: the event name, the trigger, and a step_number and step_name parameter so it builds a clean funnel in GA4.
>
> Tell me how to detect step changes given how this form works — URL change, dataLayer push, or DOM change — and which of those is most robust to future redesigns.
>
> Also instrument backward navigation and validation errors by field. The field people get stuck on is usually the field to remove.
>
> New workspace, folder named Form Funnel, no publish.

25 Lead Value Assignment

Use when *Before you turn on value-based bidding for a lead-gen account.*

You'll need Your close rate by lead type and your average deal value.

> Assign values to my lead events so bidding optimises for revenue rather than volume.
>
> From the close rates and deal values below, calculate an expected value per lead type. Show the arithmetic — I will check it.
>
> Then tell me how to pass that value: statically in the tag, dynamically from a dataLayer

> parameter, or later via offline conversion import. Recommend one for my situation and say what it costs to implement.
>
> Flag the risk plainly: a wrong static value teaches the algorithm the wrong lesson at scale, and it will take weeks to unlearn.

CLUSTER 05 · 5 PROMPTS

Google Ads Conversions

The tags that decide what your bidding learns from. A misconfiguration here does not just misreport — it actively trains the algorithm on the wrong thing.

26 Google Ads Conversion Tag Setup

Use when *New account, or any account where conversion counts look wrong.*

You'll need Read access, plus your conversion ID and labels.

> Set up Google Ads conversion tracking correctly in this container.
>
> Build: the conversion linker (this must fire on all pages and is missing more often than people expect), and the conversion tag with the correct ID and label, value, currency and transaction ID.
>
> Check first whether any of this already exists, including as a hardcoded gtag on the site. Two implementations means double counting.
>
> Build in a new workspace, create a version, do not publish. Then give me the verification steps in Google Tag Assistant and what to look for in the Ads interface over the following 48 hours.

27 Enhanced Conversions Implementation

Use when *When measurement is losing conversions to consent and cross-device.*

You'll need Read access, plus what customer data is available on the confirmation page.

> Implement enhanced conversions for web in this container.
>
> Specify: which user-provided data fields are available and usable, how they must be hashed or passed, the dataLayer spec if the developer needs to expose them, and the tag configuration.
>
> Be explicit about the compliance side — what consent is required before this data can be sent, and what must be disclosed in the privacy policy.
>
> Build in a new workspace. Then tell me how to verify it is actually working, because enhanced conversions can appear configured and be silently receiving nothing.

28 Conversion Linker & GCLID Capture

Use when *Before any offline conversion import project. This is where they fail.*

You'll need Read access, plus your CRM or lead capture form.

> Verify and fix GCLID capture through my funnel.
>
> Check the conversion linker is present and firing on all pages. Then design the capture chain: GCLID and GBRAID read from the URL, stored in a first-party cookie, written to a hidden form field, and passed through to the CRM.
>
> Tell me exactly where this typically breaks — cookie lifetime, cross-domain handoff, forms that reload, single-page-app routing — and how to test each link in the chain.
>
> Build the GTM side in a new workspace and write the developer spec for the form side separately.

29 Conversion Action Duplication Check

Use when *When Ads reports more conversions than the business recognises.*

You'll need Read access, plus your Google Ads conversion actions list with counting settings.

> Find where Google Ads is counting the same thing twice.
>
> Cross-reference the container against the conversion actions list below. Identify: the same action tracked by two conversion actions; a conversion action set to count 'every' where it should count 'one'; soft actions like page views or button clicks included in the primary bidding goal; and actions with placeholder values someone set once and forgot.
>
> For each, tell me the corrected setting and warn me if fixing it will visibly reduce reported conversions — I would rather explain that to the business before it happens than after.

30 Offline Conversion Import Prep

Use when *Lead gen, long sales cycles, anything closing off-site.*

You'll need Your CRM stages and the GCLID capture status from prompt 28.

> Prepare everything the offline conversion import needs from the tracking side.
>
> Confirm GCLID capture is working end to end — do not proceed to the rest if it is not, and say so plainly.
>
> Then map my CRM stages to conversion actions with values, specify the upload format and cadence, and tell me which stage should be the primary bidding signal.

> Give me the implementation as a spec I can hand to whoever owns the CRM, plus the validation test that proves data is arriving. Without this, the account optimises for lead volume and pays for the wrong leads.

CLUSTER 06 · 5 PROMPTS

Meta, LinkedIn & Other Platforms

Every extra platform is another tag, another consent obligation and another chance to double-count. This cluster keeps the sprawl honest.

31 Meta Pixel & CAPI Audit

Use when *When Meta-reported conversions and backend orders diverge.*

You'll need Read access, plus your Conversions API setup details if any.

> Audit my Meta tracking in this container.
>
> Check: the base pixel fires once on all pages; standard events use Meta's standard names where one fits; event parameters are complete; and whether the Conversions API is running alongside the pixel.
>
> If both browser and server events are sending, check for deduplication — I'll cover that in the next prompt, but flag here whether it appears to be in place at all.
>
> Read only. Report findings with severity and tell me which fix most affects reported performance.

32 Meta Event Deduplication

Use when *Any account running pixel and CAPI together — which should be all of them.*

You'll need Read access, plus how your server-side events are generated.

> Set up event deduplication between the Meta pixel and the Conversions API.
>
> Specify how event_id must be generated so the same value is available to both the browser event and the server event, and where it comes from — usually the order ID for purchases.
>
> Check event_name matches exactly between the two, because deduplication needs both to align.
>
> Build the browser side in a new workspace and write the server-side requirement as a developer spec. Then tell me how to confirm deduplication is working in Events Manager, and what the numbers look like when it is not.

33 LinkedIn Insight Tag Setup

Use when *B2B accounts running LinkedIn Ads.*

You'll need Read access, plus your LinkedIn partner ID and conversion definitions.

> Set up LinkedIn conversion tracking in this container.
>
> Build the Insight Tag firing on all pages, plus the conversion events for the actions I list below. Use LinkedIn's event-specific conversion setup rather than URL-based conversions where possible, because URL rules break when the site changes.
>
> Check first that the Insight Tag is not already hardcoded on the site.
>
> New workspace, version, no publish. Include the verification steps and the expected delay before LinkedIn reports data.

34 Multi-Platform Event Mapping

Use when *When the same conversion needs to reach four platforms without diverging.*

You'll need Your list of platforms and the conversions each needs.

> Build a single event map that feeds every platform from one source of truth.
>
> For each business action below, show me: the dataLayer event name, and the corresponding event name and parameters for GA4, Google Ads, Meta and LinkedIn.
>
> The point is one dataLayer push feeding all four tags, so the platforms cannot drift apart. Tell me where that is not possible because a platform requires something structurally different.
>
> Output as a mapping table. Then build the tags in a new workspace, one folder per platform.

35 Third-Party Tag Risk Review

Use when *Before adding a vendor tag, and annually for the ones already there.*

You'll need Read access.

> Assess the risk carried by every third-party tag in this container.
>
> For each: what vendor it belongs to, what data it can access from the page, whether it can inject further scripts, and whether it fires before or after consent.
>
> Flag specifically: any tag with a Custom HTML implementation that could read form fields or URL parameters containing personal data; and any tag from a vendor we no longer have a contract with.
>
> Read only. Rank by risk, not by how long the tag has been there. Old tags are usually the

> higher risk precisely because nobody remembers approving them.

CLUSTER 07 · 5 PROMPTS

Consent, Privacy & Server-Side

The cluster that stops a tracking project becoming a legal problem. In India the DPDP Act now sits alongside GDPR obligations for anyone with EU or UK visitors.

36 Consent Mode v2 Implementation Check

Use when *Any site with EU, UK or EEA traffic. Increasingly, any site at all.*

You'll need Read access, plus your consent management platform details.

> Check whether Consent Mode v2 is correctly implemented in this container.
>
> Verify: default consent state is set before any tag fires; the CMP updates consent correctly; ad_storage, analytics_storage, ad_user_data and ad_personalization are all handled; and tags have the correct additional consent checks configured.
>
> Tell me what happens right now under a deny — do tags fire anyway, do they send cookieless pings, or nothing at all?
>
> Read only. Report the gaps in order of legal exposure first, measurement impact second.

37 Tag Firing Under Deny

Use when *After the consent check. This is the test people skip.*

You'll need Read access, plus your CMP configuration.

> Walk me through exactly what fires under each consent state.
>
> For grant-all, deny-all, and each partial combination my CMP offers: list which tags fire, which are blocked, and which fire in a modelled or cookieless mode.
>
> Flag any tag that fires under deny and should not — that is the finding that matters and it is more common than people expect, particularly with Custom HTML tags that bypass consent checks.
>
> Then tell me what share of my traffic each state represents, if that is visible, and what that means for how much I should trust day-to-day fluctuations in my numbers.

38 PII Leak Audit

Use when *Annually, and any time a new form or Custom HTML tag is added.*

You'll need Read access, plus your URL structures for logged-in and post-form pages.

> Find where personal data may be leaking into analytics.
>
> Check: URL parameters that could contain email addresses, phone numbers or names being sent as page_location; form field values captured by auto-event variables; Custom HTML tags that read the DOM broadly; and any variable passing user identifiers into GA4.
>
> GA4 prohibits sending personal data, and the exposure is regulatory, not just a policy breach.
>
> For each finding: what is leaking, how, and the fix — usually a redaction variable or a corrected trigger. Rank by severity and tell me which requires action this week.

39 Server-Side GTM Migration Plan

Use when *When browser-side measurement loss is material and you're considering server-side.*

You'll need Read access, plus your traffic volume, hosting situation and budget.

> Assess whether server-side GTM is worth it for this account, then plan it if so.
>
> Start with the honest assessment: what measurement loss would it actually recover, what it costs to run and maintain at my traffic volume, and what it does not fix. Do not assume the answer is yes — for many mid-size accounts it is not.
>
> If the case holds, give me the migration plan: which tags move server-side first, what stays in the browser, the container structure, and the sequence that avoids a measurement gap during the transition.
>
> Include what could go wrong and how I would notice.

40 DPDP & GDPR Readiness Review

Use when *Indian businesses with EU or UK visitors, or handling significant personal data at home.*

You'll need Read access, plus your privacy policy and your data processing arrangements.

> Review my tracking setup against Indian DPDP Act obligations and UK/EU GDPR where applicable.
>
> Cover: what personal data this container collects or transmits; whether consent is obtained before it happens; whether the privacy policy accurately describes it; the cross-border transfer position for data going to platforms outside India; and whether data retention settings are

> configured deliberately or left at default.
>
> Give me a gap list with the specific remediation for each.
>
> Be clear about what you are and are not — this is a technical review that helps me brief a lawyer, not legal advice. Say so in your output.

CLUSTER 08 · 5 PROMPTS

Debugging, QA & Governance

The habits that separate a container someone can inherit from one that has to be rebuilt. Also the rules that make it safe to let an AI near it at all.

41 The Not-Firing Debugger

Use when *When a tag should fire and doesn't, and you've been staring at Preview mode for twenty minutes.*

You'll need Read access, plus the tag name and what you see in Preview mode.

> Debug why this tag is not firing.
>
> Work through it in order and tell me what you find at each stage: does the trigger's event actually occur; do all the trigger conditions evaluate true; is a blocking exception attached; are the variables the trigger depends on populated at that moment; is consent blocking it; and is the tag paused or scheduled.
>
> Check the container configuration for each and tell me which stage it fails at.
>
> If you cannot determine it from the container alone, say exactly what you need me to check in Preview mode rather than guessing at a cause.

42 Pre-Publish QA Checklist

Use when *Every single time, before you publish. No exceptions.*

You'll need The workspace changes you are about to publish.

> Build the QA checklist for these specific changes before I publish.
>
> List every change in this workspace. For each: what to test, in what order, on which device and browser, and what a correct result looks like.
>
> Include regression checks — what existing tracking could these changes plausibly break? That is the part people skip and it is where the damage comes from.
>
> Also give me: what to check in Preview mode, what to check in the destination platform, and what to monitor for 48 hours after publishing.
>
> Do not publish. This is a checklist for me to work through.

43 Version Diff & Change Explainer

Use when *When something broke and you need to know what changed.*

You'll need Read access, plus the version numbers to compare.

> Compare these two container versions and tell me exactly what changed.
>
> For each difference: what was added, modified or removed; who made it and when, if that is available; and what the likely intent was.
>
> Then assess each change for whether it could plausibly cause the symptom I'm describing below.
>
> Rank the candidates by likelihood and tell me what evidence would confirm or rule out each. Read only — I want to understand before I revert anything.

44 The Rollback Plan

Use when *Written before you publish, not after something breaks.*

You'll need Read access, plus the current live version number.

> Write the rollback plan for this publish.
>
> Record the current live version number and what it contains, so we have a known-good state to return to.
>
> Then: the exact steps to revert, how long a revert takes to propagate, what data is lost or double-counted during the gap, and who needs to be told.
>
> Also define the trigger for rolling back — the specific metric and threshold that means 'revert now' rather than 'wait and see'. Deciding that in advance is the whole point; nobody makes a good call about it at 11pm.

45 Container Documentation & Handover

Use when *Before leave, before offboarding, before you become the single point of failure.*

You'll need Read access to the full container.

> Write the documentation that lets someone else own this container.
>
> Cover: what each tag does in plain language and why it exists; what fires it; what depends on what; the naming conventions in use; the dataLayer events the site pushes and who maintains them; known issues and their status; and the three things most likely to break, with what to do about each.

> Write it for a competent person who has never seen this container.
>
> Then list what you could not document because it is not visible in the container — undocumented site dependencies, business context, historical decisions. That list is the actual handover risk.

Your first hour

Forty-five prompts is a library, not a to-do list. This is the order that finds the broken thing fastest.

Minutes	Run	What you should have
0–15	01	A read-only picture of what is actually live on your site.
15–25	02	Every place you are double-counting, and which tag to keep.
25–35	08	Whether your GA4 foundation is sound or quietly broken.
35–45	38	Whether personal data is leaking into analytics. Check this early.
45–60	36	What fires under consent deny — and what should not.
Next session	07	A staged cleanup plan. Pause first, delete later.

One warning worth reading twice

Prompt 38 — the personal data leak audit — has a habit of finding something. Email addresses in URL parameters, form values captured by auto-event variables, Custom HTML tags reading the DOM broadly. It is uncomfortable and it is regulatory exposure, not a reporting inconvenience.

Run it before you build anything new. Adding tracking on top of a container that is already leaking simply increases the surface area.

The tooling is not the edge. Anyone can connect a container to an AI in ten minutes. The edge is the operator who knows which check to run first, what a good answer looks like, and when to stop and ask a human.

Who wrote this

Chandan Kumar

Founder — Global Info Edge

Performance marketing, measurement and growth systems.
New Delhi · Google Premier Partner · Meta Business Partner

I have spent fifteen years running paid media and measurement for B2B, B2C and D2C brands, and founded **Global Info Edge** in New Delhi in 2012. Every rule in this book — read before you write, never publish from a prompt, record the version number first — exists because at some point not following it cost somebody real money.

Track record	
Ad spend managed	₹50 Cr+
Qualified leads generated	2.5 lakh+
Client revenue influenced	₹126 Cr+
Clients served	250+
Years in performance marketing	15+

Find me here

Connect on LinkedIn

My Profile — ckumar.in

Business Profile — ckmehta.com

Global Info Edge (.com)

Global Info Edge (.in)

Instagram — @ckmehta

Use it. Improve it. Tell me where I'm wrong.

This is free to read, quote and share with attribution. If you run one of these prompts and it surfaces something surprising in your container, I would genuinely like to hear about it — that is how the next version gets better.

If you would rather someone ran all forty-five against your container with you, that is what we do. Write to Support@globalinfoedge.com or find me on LinkedIn.

Chandan Kumar

Founder, Global Info Edge

Claude can do the clicking.
You still decide what goes live.

linkedin.com/in/ckmehta

ckumar.in/about/chandan-kumar · ckmehta.com/about/chandan-kumar
globalinfoedge.com/about/chandan-kumar · globalinfoedge.in/about/chandan-kumar
Instagram @ckmehta

GLOBAL INFO EDGE

Saket, New Delhi · Support@globalinfoedge.com · +91 91559 93311

Brand. Build. Grow.