

SERVER-SIDE TRACKING

VENDOR MIGRATION

OWNED INFRASTRUCTURE

They were paying monthly for a server **they weren't allowed to look inside**

When their tracking vendor gave notice, WoodWatch had seven working days to replace the infrastructure every conversion depended on. We rebuilt it on a server they own, fixed the defects the old one had been hiding, and cut over without losing an order.

WoodWatch Wooden watches, direct to consumer · Europe and global · woodwatch.com

~90%

Lower recurring cost for the server layer, and they now own it outright

7 days

From notice to a live cutover, with a rollback path at every step

~1/3

Of the purchases Meta was reporting were the same order counted twice

01

The problem

02

What the audit found

03

The rebuild

04

The cutover

05

What they own now

INDUSTRY

Watches, direct to consumer

MODEL

Two storefronts, EU and global

STACK

Lightspeed eCom · GA4 · GTM web + server (Stape) · Google Ads · Meta · Microsoft · TikTok · Pinterest

ENGAGEMENT

Audit, then a full server-side rebuild and vendor migration

PERIOD

July to September 2026

You cannot audit what you do not own

WoodWatch sells wooden watches across two storefronts, one for Europe and one for the rest of the world. Their server-side tracking, the layer that carries every purchase to Google, Meta and Microsoft, was run by an outside vendor on the vendor's own infrastructure, for a fixed monthly fee. It worked, mostly. Nobody at WoodWatch could see inside it.

1

A monthly fee for a black box

No access to the container, no version history, no preview mode, no export. When a number looked wrong, the only available move was to email someone and wait. A defect could run for months without anyone being able to prove it was there.

2

The vendor's code ran first

A vendor-authored script was the very first tag on every page, ahead of the consent platform and ahead of the tag manager. It carried a complete second tracking implementation running in parallel with WoodWatch's own. It could only be edited by the vendor.

3

Then the notice arrived

The vendor announced they were switching the service off on a fixed date. An optional migration became a forced one, with seven working days to replace a live conversion pipeline feeding six platforms across two storefronts.

Sound familiar?

- Your server-side tracking is billed monthly by an agency, and you have never seen inside the container.
- Nobody on your team can answer "why did that number change?" without asking a third party.
- Your platform reports and your back office have quietly stopped agreeing.
- If your current partner left tomorrow, you are not sure what you would actually keep.

We had already audited the setup a few weeks earlier. That audit recommended a phased repair: fix the worst defects first, clean up second, migrate when convenient. The notice removed the option of doing it slowly, so the whole programme collapsed into a single build.

The setup was not neglected. It had drifted, and nobody could see where.

Worth saying plainly, because what follows is critical of specific components: the measurement was fundamentally working. Orders were recorded, the ecommerce funnel fired end to end, and reported revenue matched the back office at topline. The problems were specific, and every one of them was distorting a decision.

CRITICAL

Meta was being told about roughly a third more purchases than actually happened

Found. The browser pixel and the server were both reporting purchases with no reliable shared key, so Meta could not tell that the two were the same event. Some data layer events were also pushing two or three times per action, compounding it.

Why it mattered. Meta's bidding optimises against the conversions it is told about. Every ROAS figure, every cost per acquisition and every budget decision on the account was being made against a number inflated by roughly a third. This was the single most expensive defect in the setup.

CRITICAL

The rules for how tags behave lived in a file the client could not edit

Found. WoodWatch's own consent platform and their storefront's own settings were both configured correctly. But the vendor's script loaded ahead of both and set its own tag behaviour first, so what the site's own configuration said and what the tags actually did could differ.

Why it mattered. This is the ownership problem in miniature. The client had made the right decisions and could not make them stick, because the code that overrode them belonged to someone else. It also meant the recorded audience sizes and event counts were larger than the real thing. Moving this onto their own server made their configuration the one that actually applies.

HIGH

Four defects in the data layer, each quietly changing a number

Found. Events pushing multiple times per action. Currency written to several different keys, so some hits arrived with none. Checkout value summing unit prices while ignoring quantity, so any multi-item basket under-reported. And two different product identifier fields used interchangeably, which split item reporting and broke catalogue matching.

Why it mattered. The under-reported basket value is the one that costs money directly: it under-feeds value-based bidding on both Meta and Google, so the platforms were being taught that WoodWatch's customers are worth less than they are.

HIGH

Campaign tagging was inconsistent, so channel reporting understated paid social

Found. Tagging conventions had never been standardised across platforms, so a share of paid social traffic was not being recognised as paid. Separately, the payment provider was appearing as a referral source, a common default that quietly takes credit for the orders that pass through it.

Why it mattered. Channel reporting is what budget conversations run on. If paid social is being credited to direct and the payment gateway is being credited for purchases, the channel mix on the page is not the channel mix in reality.

MEDIUM

The usual accumulation that every long-running setup collects

Found. More than ten analytics properties. A tag for a Google product retired in 2023. Duplicate conversion tags passing different order IDs and values. Enhanced conversions hardcoded to a single market's account. A third of the advertising audience library empty, and a majority of tracked events with no report, audience or import reading them.

Why it mattered. None of it was urgent on its own, and none of it is unusual: this is what every setup looks like after a few years of platform changes and people moving on. It makes the data harder to use than it needs to be, and it is exactly the kind of housekeeping that only ever happens when a migration forces the issue. The rebuild was the moment to do it.

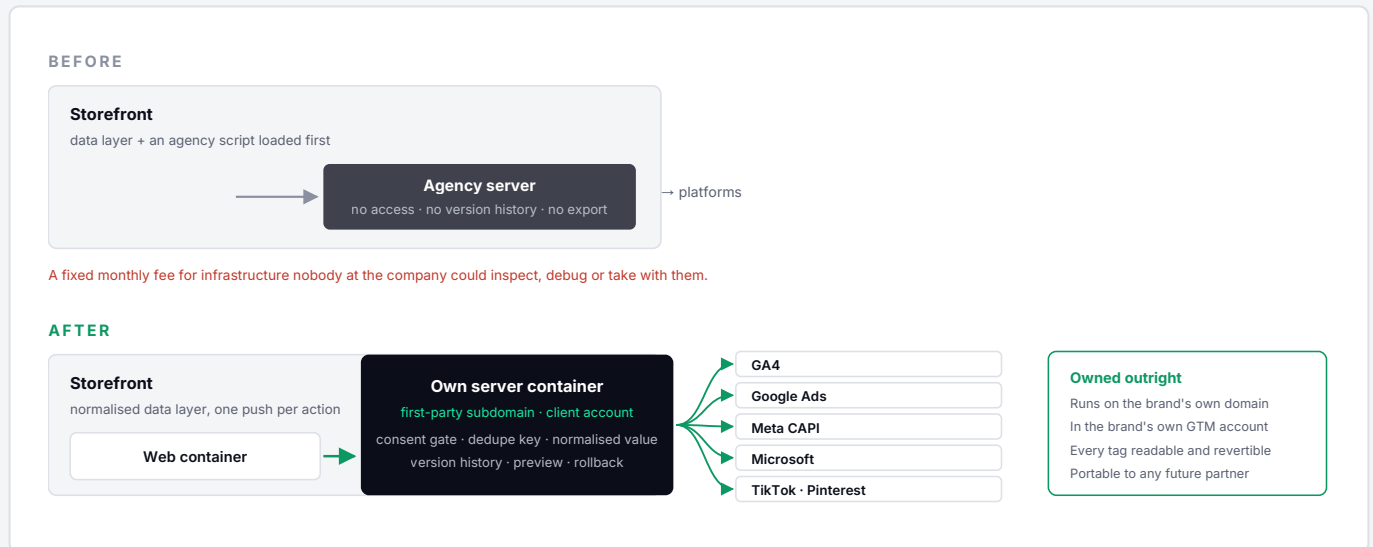
What Meta was reporting, versus what happened



Indexed to real orders = 100. Roughly a third of reported purchases were the same order counted twice, and every ROAS, cost per acquisition and bidding decision on the account was being made against the taller bar. Absolute figures withheld at the client's request.

One event stream, one server, one fan-out, all of it owned by the client

The principle was simple. The platforms stop disagreeing when they stop being fed separately. Everything routes through a single server container that WoodWatch owns, on their own subdomain, in their own account, with version history and a rollback button.



The deduplication contract

Purchases now carry an event ID derived from the order number itself, so the browser pixel and the server send the same value every time, even if the customer refreshes the confirmation page. Meta only deduplicates when both the event name and that ID match, which is why the old setup could not do it. Everything else gets a random ID per push, and a guard in the container drops any repeat of an ID it has already seen in that page view.

Fail open, never silently drop

The guard is written so that a missing ID costs a duplicate rather than a lost conversion. That is a deliberate choice: over-counting is visible and diagnosable, under-counting looks like a bad week. The same instinct runs through the build, which is why event ID is registered as a reportable dimension. If duplication ever comes back, it will be provable in an afternoon rather than an argument.

WHAT WAS BUILT

- **A first-party server container** on WoodWatch's own subdomain, so the visitor cookie is set by their domain rather than a third party and survives browser restrictions that cap third-party cookies at seven days.
- **The data layer normalised in the tag manager, not the theme**, which kept a developer deploy off the critical path: one push per action, one currency resolver, basket value recalculated from price multiplied by quantity, and one consistent product identifier across every event.
- **Six platforms rebuilt server-side** from that single stream, with visitor preferences applied once on the server rather than repeated in each browser tag, so the same rule set governs every destination.
- **Existing conversion actions kept, not recreated**. New ones reset the platforms' learning and orphan the history. This is the most expensive avoidable mistake in a migration like this.
- **Market-driven lookups** replacing values that had been hardcoded to one country, so enhanced conversions work in every market rather than one.
- **A written data layer specification** handed to the client's developers, covering the ten funnel events, the exact item shape, and the ground rules that caused most of the original problems.

The thing we told the client to expect, before it happened

Fixing the duplication meant reported Meta purchases would fall by roughly a third on the day it went live. Reported return on ad spend would look worse; actual return would be unchanged. We put that in writing beforehand, with the reasoning, so that nobody would look at the drop and conclude the new tracking was broken. Telling a client their numbers are about to get worse, and why that is the point, is part of the job.

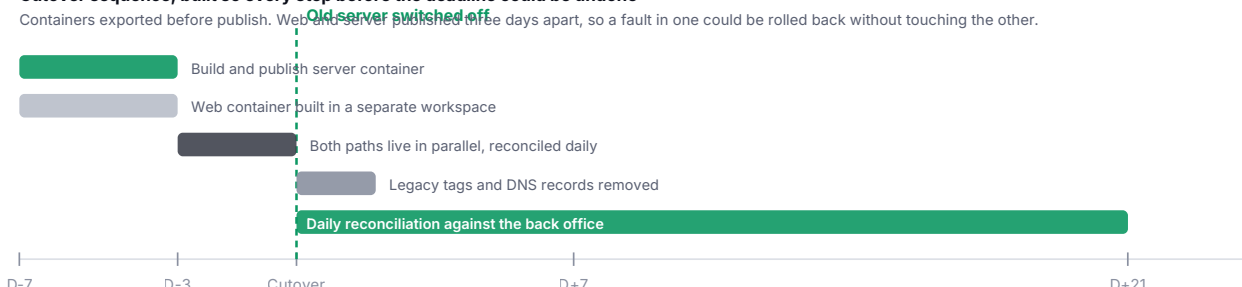
04 · THE CUTOVER

Seven days, a hard deadline, and no appetite for a lost weekend of orders

The audit had assumed three to four weeks of running old and new side by side. We had days. The answer was not to skip the parallel run but to compress it, and to make every step before the deadline reversible.

Cutover sequence, built so every step before the deadline could be undone

Containers exported before publish. Web and server published three days apart, so a fault in one could be rolled back without touching the other.



What we checked before publishing

- Every trigger firing once and only once through a full funnel, including a multi-item basket and a refresh of the confirmation page
- Both storefronts, both currencies, more than one market, because the hardcoded single-market defect existed precisely because someone had tested one market
- Meta confirming events as deduplicated rather than as two separate events, and match quality holding at its previous level
- Every visitor-preference scenario, checking that the client's own configuration was the one being applied in each
- Outbound response codes from the server, not just a green tick in the tag manager

Reconciliation, the only measure that counts

From three days before cutover to three weeks after, one row per day: orders and revenue from the back office alongside the analytics, ad platform and server figures. Tolerance of two percent on revenue and one order on count. Anything outside tolerance was diagnosed the same day, because a discrepancy found three weeks later is nearly impossible to trace back to a cause.

The old and new paths ran together for the final three days, which temporarily double counts. We flagged those three days as unusable for reporting in advance. That is the price of a safe cutover, and it is worth paying.

The rollback plan, written before anything was published

Both containers exported to file and stored outside the platform. Web and server published three days apart, deliberately, so a fault in one could be rolled back without touching the other. If the web container had broken the funnel, a version rollback restores the previous state in under a minute, and that lever gets pulled before anyone starts diagnosing. Nothing was deleted until its replacement was confirmed working, and everything retired was moved to a dated folder and paused for a fortnight first rather than removed outright.

05 · WHAT THEY OWN NOW

The same measurement, for about a tenth of the money, and it is theirs

Recurring cost of the server layer

Previous, rented server



index 100

Now, owned server



about 9

Recurring cost of the server layer, indexed to the previous arrangement = 100. Roughly a 90% reduction, and the brand now owns the container rather than renting access to someone else's.

BEFORE	AFTER
Server run by a vendor, on vendor infrastructure	Server owned by WoodWatch, on their own subdomain and in their own account
No access, no version history, no export	Full version history, preview mode and one-click rollback
Tag behaviour set by a script nobody at the company could edit	The client's own configuration is the one that applies, enforced again on the server
Meta reporting roughly a third more purchases than happened	One deduplication key shared by browser and server, and event ID reportable so it stays fixed
Basket value ignoring quantity, currency written several ways	Value recalculated from price and quantity, one currency resolver, one product identifier
Enhanced conversions working in one market	Market-driven lookups, working in every market
A fixed monthly fee	Roughly 90% less, on metered infrastructure, with the container portable to any future partner

What we are, and aren't, claiming

This was a migration under a deadline, not a growth campaign, and we are not attaching a revenue number to it. What can be said is this: the cutover happened on time, the reconciliation held inside tolerance, the recurring cost fell by roughly 90%, and the defects that had been distorting bidding are gone and now provable if they return. **The real deliverable is that WoodWatch can answer questions about their own data without asking anyone's permission.**

How this was verified. The full page load sequence recorded live and read line by line, so the order in which each script ran could be established rather than assumed. Container previews on both storefronts through a complete funnel. Outbound API request and response bodies inspected for every platform, not just tag status. Meta test events confirmed as deduplicated in Events Manager before publish, with the test code removed afterwards. Daily reconciliation against the client's back office from three days before cutover to three weeks after.

Do you own your server-side tracking, or rent it?

If you are paying monthly for infrastructure you cannot see inside, a tracking review will tell you what is actually running, what it is costing you in distorted bidding, and what it would take to own it.

[Book a tracking review](#)

Want a cheaper starting point? [Run a free Google Ads audit](#) →

About Swishiy

Swishiy is a data and measurement agency for ecommerce and B2B brands that spend real money on paid media. We sit between the platforms that report on themselves and the agencies that report on the systems they manage: the independent layer that verifies whether the numbers are true, fixes them when they aren't, and builds the measurement that lets a business bid on value rather than volume.

Tracking & conversion audits

GA4, GTM, consent, Google Ads, Meta, Microsoft. Every defect with its evidence and its commercial consequence.

Server-side & CRM pipelines

First-party server containers you own. Offline and CRM conversions delivered to every platform with value attached.

Marketing Mix Modeling

Bayesian MMM (Google Meridian) to show which channels drive incremental revenue, independent of platform attribution.

Track record

Top Rated Plus on Upwork · 77+ projects · 100% Job Success. Built and led by Qudees Kiani.

Contact

swishiy.com · qudees@swishiy.com
Book a call: swishiy.com/book



swishiy.com · Trust-Driven Growth · Case Study No. 03