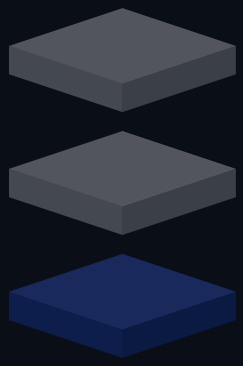
Ariavor



One governed model across a multi-brand retail group

A retail group of several brands, each with its own systems and its own answer to what a margin is. One governed Fabric model now spans them, with one definition of revenue, margin and inventory.

SECTOR
Multi-brand retail

PERIOD
10 months

PATH
Foundation and Activation

COMPANY
500—1,000 people

OCTOBER 2026

AT A GLANCE

3 answers to what a margin is, replaced by one

Every brand could report on itself, but the group could not compare them without a week of manual work. We replaced an 8-year-old SQL Server warehouse and a nightly Excel chain with Microsoft Fabric, and one definition per metric made the comparison a filter selection.

1
definition of revenue, margin and inventory

3
sales channels: stores, online shops, wholesale

4
semantic models on one lakehouse

10
months, as lead architect with a delivery team

Anonymised by sector and size, because the confidentiality that protects this client would protect you too. Sector and outcomes are real; identifying detail is not.

Before and after

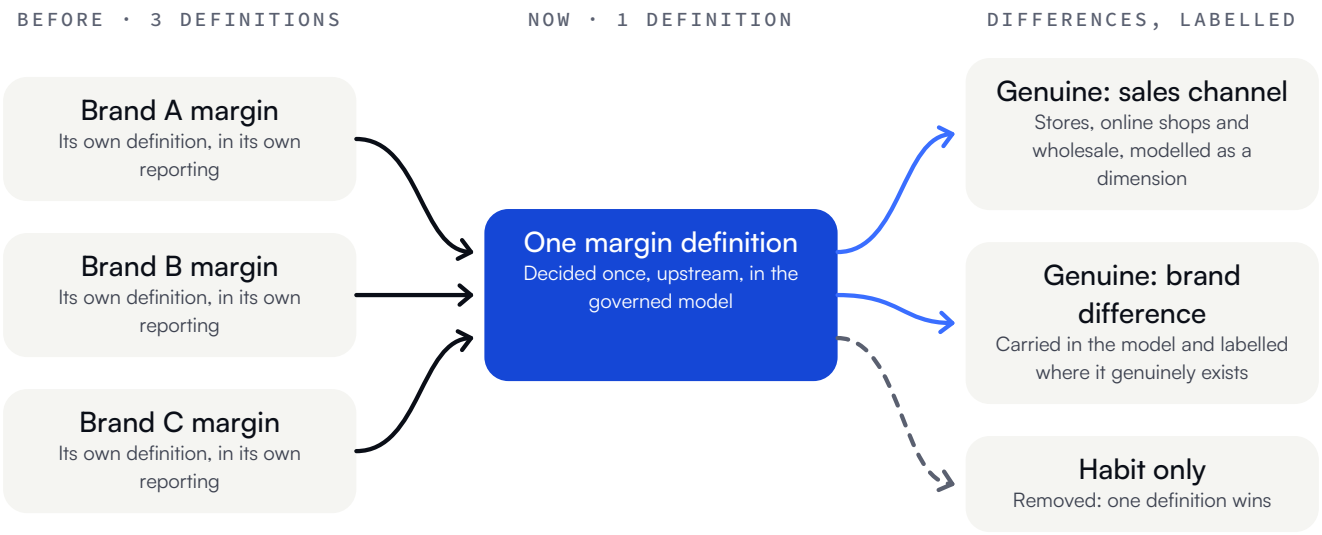


Figure 1 · 3 brands answering what a margin is 3 ways, resolved into one definition. The differences that are real are modelled and labelled; the ones that were habit are removed.

Before and after (cont.)

CLIENT

A multi-brand retail group of 500 to 1,000 people, with around 50 stores, 3 online shops, a wholesale business and around 200 BI consumers.

MANDATE

Introduce Microsoft Fabric from scratch in place of an 8-year-old SQL Server warehouse and a nightly Excel chain, and give the group one answer per metric.

Comparison as a filter selection

Once revenue, margin and inventory were defined independently of the channel, consolidating the group stopped being a piece of work. The group, one brand or one channel is now a filter on the same model.

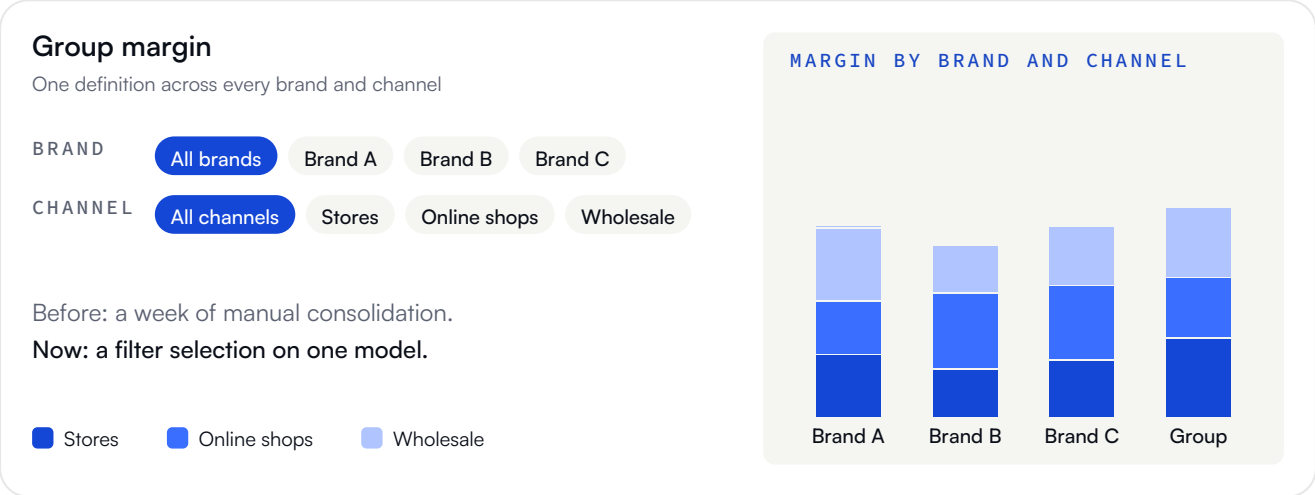


Figure 2 • Consolidation became a filter selection: the group, a brand or a channel is a click on the same model. Illustrative: no client figures are shown.

Store revenue in near real time



Figure 3 • Till data streamed into a KQL database replaced a 2-day reporting delay on store revenue.

A customer master as its own data product

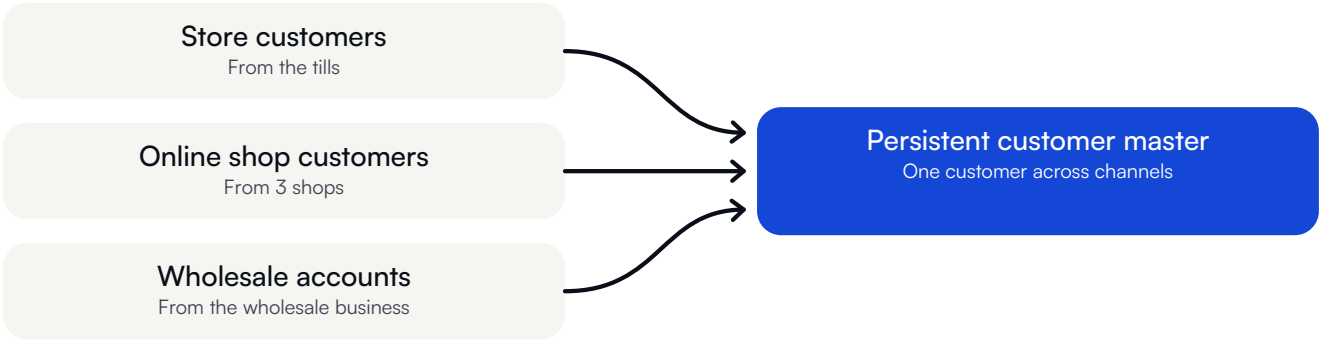


Figure 4 • Customer identity resolution built as its own data product, so one customer is one record whichever channel they buy through.

How it was delivered

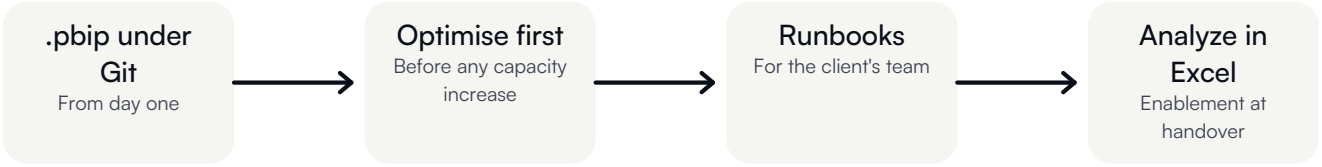


Figure 5 • Delivered as lead architect with a specialist delivery team, and handed over with runbooks and Excel enablement.

The architecture

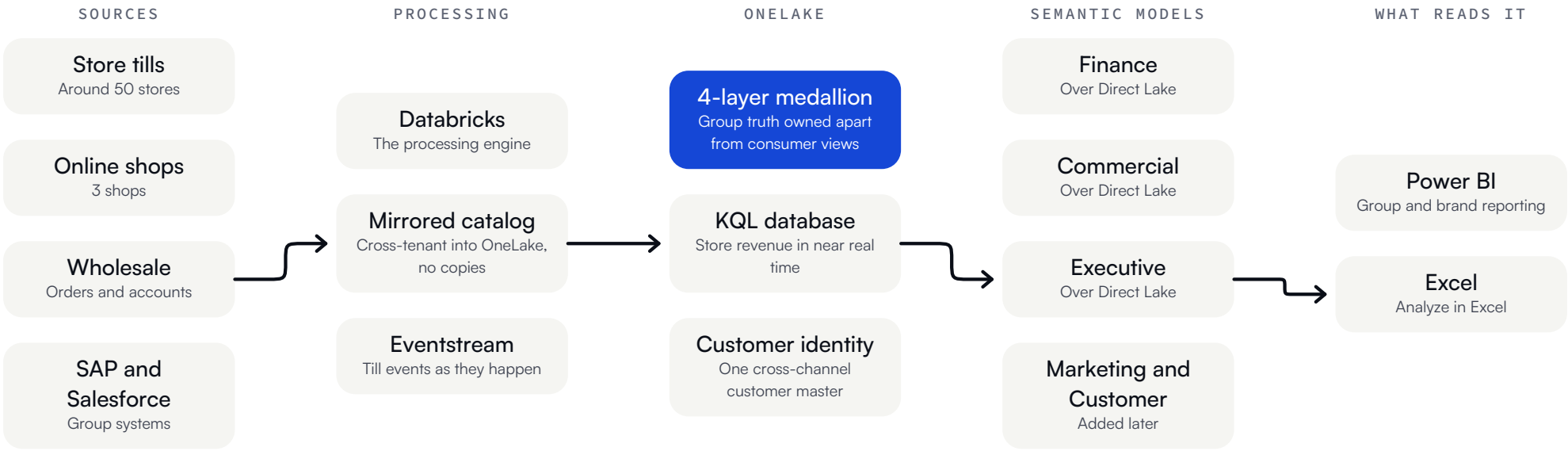


Figure 6 • Databricks stays the processing engine and is mirrored into OneLake rather than copied. The semantic models read the lakehouse over Direct Lake, with no nightly import.

Architecture decisions

DECISION	REJECTED ALTERNATIVE	WHY
One model spanning the brands	One model per brand	Faster to build, but it keeps the reconciliation problem. The comparison is the deliverable.
Differences modelled, not flattened	One margin forced on every brand	A genuine difference is modelled and labelled; one that was habit is removed.
Databricks kept and mirrored	Rebuilding processing in Fabric	Mirrored cross-tenant into OneLake, so Power BI reads it over Direct Lake with no nightly import.
Group truth owned apart from consumer views	One layer serving every use	A consumer view can change without touching the group figures.

What the client has today

- ✓ One governed semantic layer across the brands, with the real differences modelled.
- ✓ Group-level comparison as a filter selection, without the week of manual consolidation.
- ✓ Store revenue within the trading day instead of 2 days later.
- ✓ A persistent cross-channel customer master, built as its own data product.

WHERE IT LEFT THEM

The group is compared by brand or channel with one click, on figures that mean the same in every brand.

STACK

Microsoft Fabric · Direct Lake · Databricks · Eventstream · KQL Database · Power BI · DAX · SAP · Salesforce · Azure DevOps

Niklas Hoeppener • Founder • info@ariavor.com • ariavor.com