Google Marketing Platform Data Pipeline

DV360 Metadata | Ads Data Hub | BigQuery | AWS SQS

Case Study

3	Pipelines
8	Entity Types
6	Tiers
3	Report Formats
Outbox+	Reconciliation

Overview

A production data platform consisting of three event-driven pipelines that synchronize Google DV360 advertising data from GCP to AWS. Each pipeline has its own trigger pattern, orchestration layer, and worker implementation. All three share a common cross-cloud delivery mechanism: a transactional outbox with receipt-based reconciliation from BigQuery to AWS SQS.

The platform covers DV360 entity metadata, performance reports, and Ads Data Hub match-rate workflows with UPDM table creation, async job polling, and privacy-aware retry handling.

Business Problem

Ordered delivery: Parent-child campaign entities must reach the downstream system in dependency order.

Auto-detection: DV360 report tables arrive unpredictably in BigQuery; the pipeline reacts to table creation events.

Privacy sandbox: Match-rate analysis runs inside Ads Data Hub with async operations and differential privacy cooldowns.

Provable delivery: Every SQS message needs an audit trail so gaps are detected instead of silently ignored.

Architecture

Sources	Google Cloud Platform	AWS
DV360 Data Transfer	Eventarc: Pub/Sub -> Workflow + Tier Controller -> BigQuery Outbox + Receipts	SQS Queues
DV360 Offline Reporting	Eventarc: BQ Audit Log -> Workflow + Report Workers -> BigQuery Normalization	SQS Queues
GCS: Sales CSV Upload	Eventarc: GCS Finalize -> Cloud Tasks State Machine -> ADH API + UPDM	SQS Queues

Tech Stack: Python / Flask / boto3 | BigQuery stored procedures | Cloud Run Jobs & Services | Cloud Tasks | Cloud Workflows | Eventarc | Pub/Sub | ADH API | AWS SQS | Docker | Bitbucket Pipelines

Pipeline Modules

Four modules, each mapped to a specific production data problem

Case Study

1. DV360 Metadata Pipeline

Tiered Delivery

Synchronizes 8 DV360 entity types to AWS SQS with enforced parent-before-child ordering. Entities are assigned to 6 dependency tiers; a Cloud Workflow builds the BigQuery outbox and a Cloud Run tier controller chains tiers sequentially with Cloud Tasks. Shards run in parallel inside each tier.

Pattern: dependency tiers -> shard parallelism -> loopback polling -> reconciliation

2. DV360 Reports Pipeline

Field Normalization

Detects new DV360 report tables via BigQuery ADMIN_WRITE audit logs, parses report type, merges data, and delivers it through the shared outbox pattern. Workers map DV360 snake_case fields into a common interface schema and split dimensions from metrics.

Pattern: BQ audit trigger -> report type parse -> normalization -> outbox delivery

3. ADH Match-Rate Workflow

Async State Machine

Ingests sales CSVs from GCS, resolves variable schemas with alias-based matching, creates UPDM tables, runs ADH match-rate queries, and exports results. A 6-state Cloud Tasks workflow manages async ADH operations, retries, cooldowns, and stuck-batch recovery.

Pattern: file upload -> schema resolve -> UPDM -> ADH query -> result export

4. BigQuery to AWS SQS Handoff

Outbox Pattern

Provides reliable cross-cloud delivery used by all three pipelines. BigQuery builds outbox rows, workers send batched SQS messages, receipts are written back to BigQuery, and reconciliation compares expected vs. actual delivery counts before marking a run as SENT or PARTIAL.

Pattern: outbox -> batch send -> receipts -> reconcile -> alert/recover

Shared Delivery Pattern: BigQuery Outbox -> AWS SQS -> Receipts -> Reconciliation

Reliability & Transferable Patterns

The main value is not only extraction - it is safe orchestration, validation, and delivery.

Case Study

Technical Challenges Solved

Cloud Tasks timeout vs. long-running jobs
Loopback re-enqueue turns a hard timeout into resumable polling sessions without duplicate job launches.

ADH privacy failures are not normal bugs
Differential privacy errors use a dedicated cooldown path instead of normal HTTP retry logic.

Deterministic shard assignment
FARM_FINGERPRINT on advertiser ID keeps row subsets stable across retries without external coordination.

Variable CSV schemas
Header normalization and alias maps convert inconsistent sales files into BigQuery-compatible schemas.

Concurrent run prevention
BigQuery guard procedures block overlapping runs and auto-close stale executions before new work starts.

Reliability Patterns

Transactional outbox	Build message table before sending
Receipt tracking	Write SQS MessageId + MD5 per send
Expected vs. actual	Compare outbox count to receipt count
Tiered ordering	Guarantee parent-before-child delivery
Per-partner idempotency	Safe fan-out under Pub/Sub retries
Stuck batch recovery	Scheduler reactivates stale states
Concurrent run guard	Prevents duplicate outbox/delivery runs

Similar Systems I Can Build

Platform data sync	Google Ads, DV360, TTD, Meta, Amazon DSP exports normalized into BigQuery or external systems.
Cross-cloud ETL	GCP-to-AWS or GCP-to-Azure workflows where silent message loss is unacceptable.
Async API orchestration	Long-running report APIs driven by Cloud Tasks state machines and recovery endpoints.

Also applicable: event-driven GCP platforms, privacy-constrained analytics, and multi-tenant data fan-out with idempotency controls.

Built by a data engineer specializing in GCP advertising data pipelines, automation, and cross-cloud delivery.

Available for contract work: BigQuery, Cloud Run, Ads Data Hub, DV360, API pipelines, and reliable data handoffs.