

# Google Ads API – Tool Design Documentation

Technical reference describing how Arclight Digital's **AI-managed Google Ads service** uses the Google Ads API: architecture, authentication model, specific API operations, security posture, and the human-in-the-loop approval policy.

**Tool:** Arclight Ads    **Operator:** Arclight Digital (ABN 39 504 024 061)    **Version:** 1.0

**Last updated:** 2026-05-06    **Region:** Brisbane, Australia

## Contents

- [Tool overview](#)
- [Architecture](#)
- [API access model](#)
- [Read operations \(GAQL\)](#)
- [Write operations \(mutations\)](#)
- [Operation volume](#)
- [Data handling and retention](#)
- [Security posture](#)
- [Human-in-the-loop policy](#)
- [Rollback procedures](#)
- [Privacy and compliance](#)

## 1. Tool overview



## COMPONENT SUMMARY

LAYER	TECHNOLOGY	PURPOSE
Storage	Supabase (Postgres)	Snapshots, recommendations queue, append-only audit log
Reasoning	Anthropic Claude (Opus 4.7)	Generates recommendations from 90-day data + per-client context
Ad data plane	Google Ads API v24	Read snapshots, apply approved mutations
Conversion validation	GA4 Data API	Independent conversion auditing
Approval UI	Streamlit (operator-only)	Pending recommendations review
Scheduling	GitHub Actions	Cron triggers for snapshot, anomaly, weekly review

## 3. API access model

Arclight Digital operates a single Manager Account (MCC) at `customers/8196591048`. Clients grant access via the standard Google Ads **account-linking workflow** (no API account creation, no scraping). The account owner accepts the link request inside their own Google Ads UI; no credentials are shared.

### AUTHENTICATION CHAIN

OAuth 2.0 desktop client in a Google Cloud project ( `arclight-ads` ) configured as **Internal** user-type for the `arclightdigital.com.au` Workspace organisation.

One refresh token issued via the `https://www.googleapis.com/auth/adwords` scope to a single operator identity ( `zac@arclightdigital.com.au` ).

The Google Ads Python client library uses this refresh token together with the developer\_token and login\_customer\_id (MCC) header to authorise every request.

API calls iterate over linked client accounts via customer\_client resource queries on the MCC.

Refresh token, developer token, and OAuth client secret are stored in google-ads.yaml on the operator workstation. The file is git-ignored, never logged, never transmitted, and not copied to CI runners that execute mutations.

## 4. Read operations (GAQL)

Six read query patterns. All are scoped to a single linked customer at a time and constrained by date ranges. No bulk download of accounts outside the operator's MCC.

### ACCOUNT DISCOVERY

```
SELECT
  customer_client.client_customer,
  customer_client.id,
  customer_client.descriptive_name,
  customer_client.currency_code,
  customer_client.time_zone,
  customer_client.manager,
  customer_client.status,
  customer_client.level
FROM customer_client
WHERE customer_client.status = 'ENABLED'
```

### DAILY CAMPAIGN PERFORMANCE

```
SELECT
  campaign.id, campaign.name, campaign.status,
  campaign.advertising_channel_type,
  campaign_budget.amount_micros,
  metrics.cost_micros, metrics.conversions,
  metrics.conversions_value, metrics.impressions,
  metrics.clicks, segments.date
FROM campaign
```

WHERE segments.date DURING LAST\_30\_DAYS

ORDER BY cost\_micros DESC

**ARCLIGHT**  
DIGITAL

## SEARCH-TERMS REPORT (LAST 7 DAYS)

```
SELECT
  search_term_view.search_term,
  campaign.id, ad_group.id,
  metrics.impressions, metrics.clicks,
  metrics.cost_micros, metrics.conversions,
  segments.date
FROM search_term_view
WHERE segments.date DURING LAST_7_DAYS
```

## KEYWORD PERFORMANCE

```
SELECT
  ad_group_criterion.criterion_id,
  ad_group_criterion.keyword.text,
  ad_group_criterion.keyword.match_type,
  ad_group_criterion.cpc_bid_micros,
  ad_group_criterion.quality_info.quality_score,
  metrics.cost_micros, metrics.conversions,
  metrics.impressions, metrics.clicks,
  segments.date
FROM ad_group_criterion
WHERE ad_group_criterion.type = 'KEYWORD'
  AND segments.date DURING LAST_30_DAYS
```

## AD PERFORMANCE

```
SELECT
  ad_group_ad.ad.id, ad_group_ad.ad.type,
  ad_group_ad.status, metrics.cost_micros,
  metrics.conversions, metrics.impressions,
  metrics.clicks, segments.date
FROM ad_group_ad
WHERE segments.date DURING LAST_30_DAYS
```

## BUDGET PACING

```
SELECT campaign_budget.id,
       campaign_budget.amount_micros,
       metrics.cost_micros, segments.date
FROM campaign_budget
WHERE segments.date DURING LAST_7_DAYS
```

## 5. Write operations (mutations)

Five mutation types, each only applied after an explicit human approval recorded in our recommendations table. The exact mutation payload is stored on the recommendation row before approval, and a corresponding rollback payload is recorded after successful mutation.

MUTATION TYPE	SERVICE	PURPOSE
Add negative keyword	<code>AdGroupCriterionService.MutateAdGroupCriteria</code> (or campaign-level via <code>CampaignCriterionService</code> )	Block search terms surfaced as wasted spend
Adjust keyword bid	<code>AdGroupCriterionService.MutateAdGroupCriteria</code>	Increase or decrease CPC bid on a converting/non-converting keyword
Pause campaign	<code>CampaignService.MutateCampaigns</code>	Stop spend on an underperforming campaign
Change campaign budget	<code>CampaignBudgetService.MutateCampaignBudgets</code>	Reallocate budget toward higher-ROAS campaigns
Change bid strategy	<code>CampaignService.MutateCampaigns</code>	Switch between Manual CPC, Maximise

Out-of-scope mutations (will not be performed at any tier): account creation, billing changes, user-management changes, customer/login modifications, conversion tracking changes, audience uploads, or any operation that would expose data outside the linked client account.

## 6. Operation volume

OPERATION	FREQUENCY	OPERATIONS PER FIRE (PER CLIENT)
Account discovery	Daily 03:00 AEST	1
Campaign snapshot	Daily 02:00 AEST	~5–20 (one per campaign × pagination)
Search-terms snapshot	Daily 02:00 AEST	~5–50 (paginated)
Keyword snapshot	Daily 02:00 AEST	~10–100 (paginated)
Approved mutation	Ad-hoc, after operator approval	1 per recommendation; typically 0–10 per client per week

Estimated total daily API operations across all clients during ramp ( $\leq 10$  clients): **under 1,000 ops/day**. Well below the 15,000 ops/day Basic-tier ceiling. Volume scales linearly with client count; we expect to remain comfortably within Basic limits for at least the first 50 clients.

### STORAGE

**Supabase Postgres database**, region: Sydney (ap-southeast-2). All client snapshot data, recommendations, and audit-log rows live here.

Encryption at rest: AES-256 (Supabase managed).

Encryption in transit: TLS 1.3 to/from Supabase, Google APIs, and Anthropic API.

### RETENTION

Campaign & search-term snapshots: rolling 24 months. Older rows pruned by scheduled job.

Recommendations and audit log: kept indefinitely while client is active. On client termination, exported to the client and purged from our database within 30 days unless legal retention applies.

Search query payloads stored only as aggregated data per the Search Terms Report — no PII (no IP addresses, no user identifiers).

### WHAT IS *NOT* STORED

End-user personal data (we never see individual searcher data; the API only exposes aggregates).

Conversion-event payloads or transaction details.

Account-holder OAuth tokens for client accounts (we use MCC linking, not OAuth-as-client).

---

## 8. Security posture

**Single operator identity.** One human (the Arclight Digital owner) holds the OAuth refresh token. No multi-user broker, no shared credentials.

**Credential isolation.** `google-ads.yaml` lives only on the operator workstation; never committed (git-ignored, with explicit `service-account*.json` and `client_secret*.json` patterns also blocked); never transmitted to mutation-running CI.

**Append-only audit log.** Every mutation against a client account is recorded with operator identity, timestamp, the exact request payload, the API response, success/failure flag, and any error message. The audit table has no UPDATE/DELETE pathway in our code.

**Service-role key segmentation.** The Supabase service-role key used by job runners is distinct from any client-shared keys and rotates on operator request.

**Two-factor authentication** on Google Cloud Console, Google Ads (operator account), GitHub, and Supabase.

---

## 9. Human-in-the-loop policy

### POLICY — STRICTLY ENFORCED IN CODE

**No mutation against any client Google Ads account is automatically applied.** Every recommendation generated by Claude — and every operator-modified variant — is written to the `recommendations` table with `status = 'pending'`. A mutation is only executed when a human operator updates the row to `status = 'approved'` via the internal Streamlit dashboard. The mutation executor refuses any row not in `approved` status.

Operator decisions are recorded with `decided_by` (operator identity) and `decided_at` (timestamp) on the recommendation row. The dashboard shows the operator the AI's reasoning, the supporting data, and the exact mutation payload before the approve button is enabled.

This holds at the code level: the executor's signature requires a recommendation row whose `status` field is `approved`; pending or rejected rows raise an exception before any API call is made. There is no path in the codebase for Claude to issue a mutation directly.

Every mutation function returns a **rollback payload** describing the inverse operation, captured at execution time. Example: a budget change from \$80 to \$130 records a rollback payload of `{operation: 'budget_change', new_budget_micros: 80_000_000}`. Rollback payloads are stored on the corresponding recommendation row.

If a change underperforms, the operator triggers rollback from the dashboard. The executor reads the rollback payload, applies it as a fresh mutation (which is itself audit-logged), and updates the recommendation row to `status = 'rolled_back'`. End-to-end recovery is typically under 30 seconds.

---

## 11. Privacy and compliance

**Australian Privacy Principles (APP)** compliant. Privacy policy published at [arclightdigital.com.au/privacy](https://arclightdigital.com.au/privacy).

**No personal data of end users** is stored — the Google Ads API only exposes aggregated reporting.

**Client onboarding** includes a written services agreement specifying data scope, retention, and termination handover obligations.

**Right to data export and deletion:** on termination, clients receive a complete export of their snapshot history, recommendation log, and audit trail; database rows are then purged within 30 days.

**No data sharing with third parties** except: Anthropic (for the weekly review prompts — payloads contain only aggregated campaign performance, no PII), Resend (for email delivery — recipient address only), and GitHub (for source code and CI logs — secrets are masked).

---

## Contact

Questions about this design or the tool's operation:

Zac Engledow · Arclight Digital · [zac@arclightdigital.com.au](mailto:zac@arclightdigital.com.au) · +61 400 600 862

