Google Ads API — Tool Design Document

Cover

Company	Definitely Mabe AB (org. nr 556836-0688)
Website	https://www.d-ma.be
Practice page	https://www.d-ma.be/ad-tech
Manager account (MCC)	881-197-5317
Contact	google-ads-api@d-ma.be
Privacy policy	https://www.d-ma.be/privacy
Terms of service	https://www.d-ma.be/terms
Cloud project	dm-ads-automation
Document date	1 September 2026

1. Summary

Definitely Mabe AB (org. nr 556836-0688) is a technology consultancy based in Sweden. We manage Google Ads accounts for e-commerce clients through our manager account (881-197-5317), to which one client account is currently linked, and we are building an internal tool to support that work. This is described publicly at https://www.d-ma.be/ad-tech, which states the same scope, the same human-approval control and the same internal-only status as this document.

The tool will read performance data for analysis and reporting. Three further capabilities are designed and in development: generating and maintaining responsive search ad assets across large product catalogues, maintaining negative keyword lists from search term reports, and uploading offline conversion values so that automated bidding reflects actual order revenue rather than a flat assumed value.

The tool is used only inside Definitely Mabe AB, by people working on our own client engagements. It is not a product, it is not resold, and it is not made available to anyone outside our client work. No change is applied to a live account without a person at Definitely Mabe AB having reviewed and approved that specific change first.

Build status — stated plainly, because it is what the access was applied for. No Google Ads API integration has been written yet. The parts of the tool that exist today are the ones that need no Ads access: a scrape of the client's own product catalogue (1,962 category pages), an analysis of their organic search performance from Search Console, and mechanical validation of generated ad copy against Google's published character limits and against our own rule that ad copy may not make claims the landing page does not itself make. Sections 4 and 6 describe the Ads API integration those parts feed into. It is specified, and it is not built.

Access position, verified 1 September 2026. Our developer token holds **Basic access**, granted on 1 September 2026, permitting 15 000 operations per day across test and production accounts. The grant was verified against the live API rather than taken from the approval email: a `GoogleAdsService` read against the linked client account (customer 937-262-7000) returned campaign rows, and `KeywordPlanIdeaService.GenerateKeywordIdeas` returned keyword ideas. Nothing has been persisted, and no campaign, budget, keyword or ad has been created or modified in that account.

The two limits that made this application necessary are both now lifted, and are recorded here because sections 4 and 8 are written against them:

- **Keyword planning.** Under the previous Explorer access, `KeywordPlanIdeaService.GenerateKeywordIdeas` returned `DEVELOPER_TOKEN_NOT_APPROVED` — *"This*

method is not allowed for use with explorer access. Please apply for basic or standard access." It now succeeds. Section 4.2 grounds ad group and keyword generation in real search demand, which is what that service provides.

- **Operation ceiling.** Explorer capped production usage at 2 880 operations per day. The catalogue in section 3 is 1 962 category pages, and one build pass over it — an ad group, its keywords and its ads per page — is several times that ceiling before any reporting is pulled at all. Basic access raises the ceiling to 15 000 per day.

The only production requests this token has made are access-level diagnostics: two on 16 August 2026 and two on 1 September 2026. **No integration has been written, and the grant did not change that** — it removed the blocker described in sections 4 and 6, it did not build anything.

2. Business model

We are engaged directly by the advertiser and paid by the advertiser. We do not resell advertising, we do not act as a reseller of Google services, and we do not charge on a share-of-spend basis for the API work described here.

Client accounts remain owned by the client. They are linked to our manager account for the duration of the engagement and can be unlinked by the client at any time. Billing, campaign history and account ownership stay with the advertiser throughout.

At the end of an engagement the client is expected to establish their own manager account and developer token. Nothing we build depends on our access persisting after handover, and our copies of the client's performance data are deleted when the engagement ends.

We comply with the Google Ads API Terms and Conditions and the Google Ads API policies. No Google Ads data is shared with, or made accessible to, any third party.

3. Current engagement

The tool is being developed against a single live engagement, which is representative of the work we expect it to support. The client is an e-commerce retailer; the account is linked to our manager account and identified on the application form.

What matters here is the shape of the catalogue rather than what it sells. It is organised by compatibility — each category page combines a product type with the specific equipment it fits, so a customer searching for a part is really asking a compatibility question. The advertisement, the landing page and the product all have to answer that question consistently, which is what makes generic ad copy useless and per-page copy unavoidable. The catalogue currently spans:

Dimension	Scale
Compatibility variants	649
Category pages (variant × product type)	1,962
Product listings across those pages	5,319

Managing advertising at this granularity by hand is not practical, which is what the tool is for, and it is why we need programmatic access rather than the Ads interface. The case for it rests on the number of distinct pages that each need their own ad group, keywords and copy — not on the size of the budget.

4. What the tool does

4.1 Reporting and analysis

*Status: **specified, not implemented.** No reporting integration has been written; the only production requests made with this token are the access-level diagnostics in section 1. The analysis this would replace is currently done from Search Console data and a scrape of the client's own catalogue — neither of which uses the Google Ads API.*

Will read campaign, ad group, keyword and search term performance to produce internal analysis and client reporting. This is expected to be the largest share of our API usage by request volume and it is entirely read-only.

- Spend, clicks, conversions and conversion value by campaign, ad group and keyword
- Search term reports, used to identify wasted spend and to inform negative keyword lists
- Change history, used to understand what has previously been tried in an account
- Account structure, used to audit configuration before proposing any change

4.2 Ad group and ad generation

Status: in development — the current focus of the engagement.

For catalogues organised by compatibility, each category page corresponds to a distinct and narrow customer intent. The tool builds the ad group structure and the ads within it directly from that catalogue: the landing pages, the keywords they should answer, and the product media the client already publishes.

Keywords are phrase or exact match only. The tool does not create broad match keywords — on a compatibility catalogue, broad match is what turns a compatibility question into irrelevant traffic, and excluding it is a deliberate constraint of the design rather than a default.

Responsive search ad assets are composed from structured catalogue data — product family, compatibility variant, and the wording on the page itself. Image assets are linked at campaign or ad group level from the client's existing product media.

Generated assets are validated mechanically before a human sees them: character limits are checked against Google's published requirements, and headline content is restricted to terms drawn from the page it points to. The tool does not compose claims that the landing page does not itself make.

4.3 Negative keyword management

Status: in development. Search term analysis runs today and the resulting negatives are applied by hand; the API path replaces that manual step.

A compatibility catalogue attracts a predictable amount of irrelevant traffic: job seekers, people intending to make the part themselves, second-hand buyers, and searches for equipment the client does not stock parts for. The tool identifies these from the search term report and proposes negative keywords, with the reason for each recommendation, rather than proposing terms on cost alone.

4.4 Offline conversion value upload

Status: specified, scheduled after 4.2 and 4.3.

Order values in this catalogue span two orders of magnitude, from a small consumable to a large fitted item. Where an account reports a flat conversion value, automated bidding cannot distinguish between them and optimises against a signal that does not reflect the business.

The tool uploads actual order values from the client's order system against the corresponding click identifiers, so that bidding optimises on revenue rather than on conversion count. No personal data is transmitted — the upload consists of a click identifier, an order value, a currency and a timestamp.

5. API services used

Service	Use	Access
GoogleAdsService	Performance, search terms, change history, account structure	Read
CustomerService	Enumerate accounts under the manager account	Read
KeywordPlanIdeaService	Search volume and keyword ideas to ground asset generation	Read

Service	Use	Access
CampaignService, AdGroupService	Create and update campaigns and ad groups	Write
AdGroupAdService, AssetService	Create and update responsive search ad assets	Write
CampaignCriterionService, AdGroupCriterionService	Manage keywords and negative keyword lists	Write
ConversionUploadService	Upload offline conversion values	Write

We do not create Google Ads accounts, modify account-level settings, or manage billing through the API.

6. Architecture

The tool runs on our own workstations and on a server under our control. It is a command-line application; there is no hosted service, no customer-facing interface, and no multi-tenant deployment. What runs today is the catalogue and Search Console analysis described in section 1; both the read and the write paths against the Google Ads API are specified and not yet built.

Layer	Detail
Runtime	Python, using the official `google-ads` client library
Authentication	OAuth 2.0 desktop client. Credentials are held in a secrets manager and resolved at runtime; no credential is written to disk or committed to version control.
Cloud project	Google Cloud project owned by our organisation, used solely for OAuth client registration and API enablement. No Google Cloud compute or storage is involved.
Data at rest	Performance exports and catalogue data are stored in a private repository accessible only to us.
Distribution	None. The tool is not packaged, published, or installed anywhere outside our own environment.

Request flow

A typical operation is designed to run as follows. Steps 1 to 3 are read-only, fully automated by design, but have not been run against any account. Steps 4 to 6 are the approval gate described in section 7 and the write path it governs; these are specified and not yet implemented. Step 6 is the only point at which anything is written to a live account.

1. Read current account state via GoogleAdsService
2. Read catalogue data from the client's product source
3. Compose the proposed change and validate it against Google's requirements
4. Render the proposal as a written diff — exactly what would change, and why
5. The operator reviews the diff and approves or rejects it
6. Apply approved changes only, and record what was applied

7. Human review of every change

The tool operates read-only by default. No mutation is applied to a live account without explicit human approval of that specific change.

This is a documented policy in our internal decision record, not a convention. It exists for a straightforward reason: an unreviewed automated process with write access to a live advertising budget carries an unbounded downside,

and no efficiency gain justifies it.

In practice this means:

- Every proposed change is presented as a written diff showing the current state, the proposed state, and the reason for the change.
- Approval is per change set. There is no standing authorisation and no category of change that bypasses review.
- Changes are grouped by how much they can affect, in three tiers. Reversible, low-magnitude changes (negative keywords, asset text, labels) are reviewed as a batched digest. New campaigns and asset groups are proposed individually. Anything touching budgets, bidding strategy, or pausing live advertising is proposed individually *and* explained on its own terms.
- Applied changes are recorded, so that account change history and our own record agree.

Nothing in the tool is capable of creating campaigns, altering budgets or pausing live advertising without a person having read the specific change and approved it. The approval gate is being built together with the write path it governs, not added to it afterwards.

8. Access and users

Question	Answer
Who uses the tool?	People working inside Definitely Mabe AB on our own client engagements. Internal users only.
Do clients have access?	No. Clients receive reports and approve changes; they do not operate the tool.
Is it resold or licensed?	No. It is not a product and is not made available to third parties.
Is it built on someone else's tool?	No. It is developed by us.
Campaign types supported	Search — the tool creates and manages Search campaigns only. Shopping and Performance Max are read for reporting; the tool does not create or modify them.

We expect our request volume to remain modest: a small number of managed accounts, reporting pulled on a daily or weekly cadence, and asset generation run in batches rather than continuously. Basic access was granted on 1 September 2026 and is sufficient for this work; Standard access exceeds what our usage requires and is not being sought. Both reasons the application gave were measured rather than inferred (section 1): keyword planning services were refused outright at the previous access level, and the 2 880 operations/day cap was below what a catalogue of 1 962 category pages requires for a single build pass. Basic access resolves both.

---

9. Tool mockups

The tool is a command-line application, so these are mock-ups of its terminal output rather than of a graphical interface. **They illustrate the designed write path, which is specified and not yet implemented** (see section 1, Build status). They are included because section 7 describes the control that matters most in this application, and a control is easier to assess shown than asserted.

9.1 LOW tier — batched digest

Reversible, low-magnitude changes are collected and approved in one pass.

```
# MOCK-UP — illustrative output of the designed write path. Not a capture.
$ ads propose negatives --account 123-456-7890

  PROPOSED  12 negative keywords          tier: LOW     reversible: yes

  # phrase                     matched spend  reason
  1 "begagnad"                     1 240 SEK  second-hand intent, client sells new only
  2 "jobb"                           310 SEK  job-seeker traffic
  3 "ritning"                         180 SEK  fabrication intent, no product match
  ... 9 more
```

```
  Estimated monthly spend released: 4 100 SEK

  [a] approve all   [s] select   [r] reject   [d] detail
```

9.2 HIGH tier — proposed individually, with reasoning

Anything touching budgets, bidding or pausing live advertising is presented on its own.

```
# MOCK-UP — illustrative output of the designed write path. Not a capture.
$ ads propose budget --campaign "Category / Generic"

  PROPOSED  budget change                tier: HIGH    reversible: yes

    current   240 SEK/day
    proposed  180 SEK/day                -25%

  Reason
    Search impression share lost to rank is 4% while lost to budget is 0%; the
    campaign has not been budget-constrained on any day in the last 30. Spend is
    being absorbed at a CPA of 640 SEK against a target of 450 SEK.

  Evidence   30d rolling window, GoogleAdsService

  This change affects a live budget. Approving applies it immediately.

  [y] approve   [n] reject   [e] explain further
```

9.3 Applied-change record

Every approved change is written to a local record so that our account of what happened and Google Ads change history can be reconciled.

```
# MOCK-UP — illustrative record format. Not a capture.
YYYY-MM-DDT09:14:22Z  account=123-456-7890  tier=LOW   op=negatives.add
                      n=12  approved_by=mathias  ref=prop-4471
YYYY-MM-DDT09:16:03Z  account=123-456-7890  tier=HIGH  op=campaign.budget
                      240->180 SEK/day  approved_by=mathias  ref=prop-4472
```