Marcio Nogueira

Google Ads API Tool — Design Documentation

Application Type: Standard Access Request

Manager Account ID: 529-401-3948

Company Website: https://marcionogueira.com.br

Contact: marcio4146@gmail.com

Country: Brazil

Document Date: May 24, 2026

Document Version: 1.0

1. Executive Summary

Marcio Nogueira is a Brazilian Software-as-a-Service (SaaS) platform designed to simplify Google Ads and Google Merchant Center management for small and medium-sized e-commerce stores (SMBs) operating in Brazil. The platform automates onboarding, campaign creation, and performance reporting through a unified dashboard accessible to non-technical merchants.

This document describes the technical architecture, API integration patterns, security model, and compliance approach for our use of the Google Ads API. We are requesting access in order to legitimately manage advertiser sub-accounts under our Manager Account (MCC), synchronize product catalogs with Merchant Center, and provide end-to-end campaign automation for our customers.

Key Information

FIELD	VALUE
Company / Product	Marcio Nogueira
Type	Independent Google Ads Developer (SaaS Platform)
MCC Customer ID	529-401-3948
Target Customers	Brazilian SMB e-commerce merchants
Estimated Sub-accounts (Year 1)	Up to 100
Primary APIs Used	Google Ads API v18, Content API for Shopping v2.1, Site Verification API
Access Tier Requested	Standard Access

2. Business Model

2.1. Target Market

Our customers are independent online retailers in Brazil, typically 1–50 employees, selling physical goods through their own online stores (e-commerce platforms such as Shopify, WooCommerce, Tray, Nuvemshop, and proprietary stores). These merchants typically:

- Have between 50 and 10,000 SKUs in their catalog
- Operate annual ad budgets between USD 500 and USD 100,000
- Lack the technical expertise to configure Google Ads and Merchant Center themselves

- Cannot afford to hire a full-time digital marketing team or specialized agency

2.2. Value Proposition

We bridge the technical gap by offering:

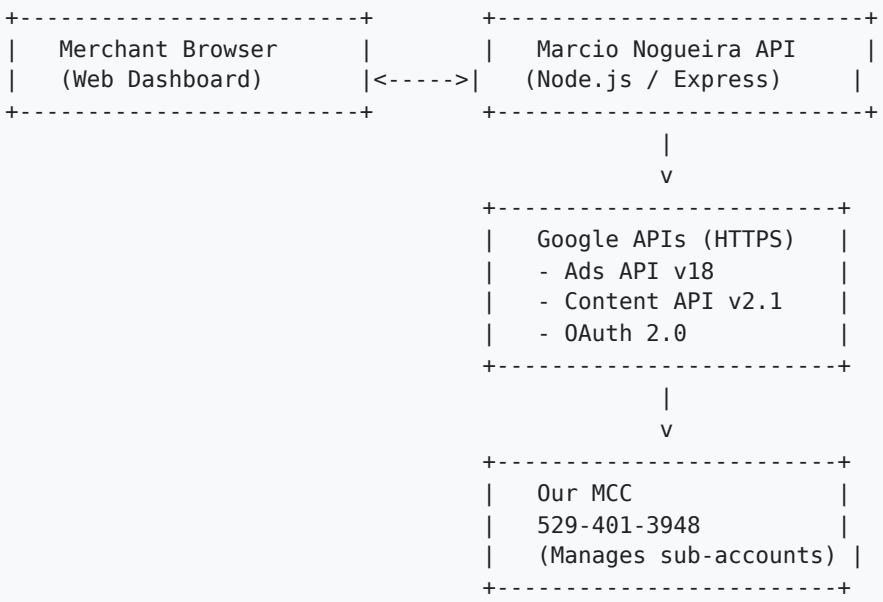
- **One-click onboarding** — merchants connect their store and Google account; our platform automatically creates the necessary Ads sub-account and Merchant Center linkage.
- **Catalog automation** — daily product catalog synchronization from the merchant's store to Merchant Center.
- **Smart campaign defaults** — preconfigured Shopping and Performance Max campaigns using best practices from Google's documentation.
- **Localized support** — Portuguese-language interface and support team operating in Brazilian business hours.
- **Consolidated reporting** — real-time metrics across Ads and Merchant Center, with attribution to product-level performance.

2.3. Revenue Model

We charge a monthly subscription fee plus a small percentage of ad spend (capped). Merchants pay Google directly for their ad spend — we do not handle billing on behalf of advertisers. This avoids any payment intermediation and keeps the merchant in full control of their budget.

3. Technical Architecture

3.1. System Components



3.2. Stack

LAYER	TECHNOLOGY
Frontend	HTML5, CSS3, vanilla JavaScript (no framework, no client-side credentials)
Backend	Node.js 24.x + Express 4.x
Auth	OAuth 2.0 via google-auth-library; express-session with HttpOnly cookies
Google SDK	googleapis v144 (official Node client)
Persistence	JSON file storage for invitations metadata (no PII beyond email)
Hosting	Cloudflare Pages (static frontend) + Node backend on managed VPS
TLS	HTTPS-only, TLS 1.3, certificates via Google Trust Services / Cloudflare

3.3. High-Level Workflow

1. Merchant signs up at `marcionogueira.com.br` and authenticates via OAuth 2.0 (Sign in with Google).

2. Our platform requests minimal Google Ads scope (`auth/adwords`) plus Merchant Center (`auth/content`).
3. Our MCC (529-401-3948) creates a new sub-account using `CustomerService.CreateCustomerClient` with the merchant's email address, which automatically sends an invitation.
4. Merchant accepts the invitation by clicking the link in the email (Google-generated URL).
5. Our platform synchronizes the merchant's product catalog with Merchant Center via the Content API.
6. Merchant configures or approves campaign settings; we create Shopping or Performance Max campaigns via the Ads API.
7. Performance data is fetched daily via `GoogleAdsService.SearchStream` for reporting in our dashboard.

4. Google Ads API Usage

4.1. API Services and Methods Used

SERVICE	METHODS	PURPOSE
CustomerService	listAccessibleCustomers, createCustomerClient	List and create managed sub-accounts
CustomerUserAccessService	mutate, listOperations	Manage user permissions within sub-accounts
CustomerUserAccessInvitationService	mutate	Send/revoke invitations
CampaignService	mutate (CREATE, UPDATE, PAUSE)	Create and manage Shopping/Performance Max campaigns
CampaignBudgetService	mutate	Manage daily and shared budgets
AdGroupService	mutate	Configure ad groups within campaigns
ConversionActionService	mutate, query	Define and track conversions
GoogleAdsService	search, searchStream	Reporting and metric queries (GAQL)
RecommendationService	apply, dismiss	Surface Google's optimization suggestions

4.2. Sample Operations

CREATE CUSTOMER CLIENT (ONBOARDING)

```
POST /v18/customers/5294013948:createCustomerClient
{
  "customerClient": {
    "descriptiveName": "Loja Cliente XYZ",
    "currencyCode": "BRL",
    "timeZone": "America/Sao_Paulo"
  },
  "emailAddress": "merchant@example.com.br",
  "accessRole": "ADMIN"
}
```

CAMPAIGN CREATION (SHOPPING)

```
POST /v18/customers/{customer_id}/campaigns:mutate
{
  "operations": [{
    "create": {
      "name": "Shopping - June 2026",
      "advertisingChannelType": "SHOPPING",
      "status": "PAUSED",
      "shoppingSetting": {
        "merchantId": "{merchant_id}",
        "salesCountry": "BR"
      },
      "campaignBudget": "customers/{cid}/campaignBudgets/{bid}",
      "biddingStrategyType": "MAXIMIZE_CONVERSION_VALUE"
    }
  }]
}
```

REPORTING QUERY (GAQL)

```
POST /v18/customers/{customer_id}/googleAds:searchStream
{
  "query": "SELECT campaign.id, campaign.name, campaign.status,
            metrics.impressions, metrics.clicks, metrics.cost_micros,
            metrics.conversions, metrics.conversions_value
            FROM campaign
            WHERE segments.date DURING LAST_30_DAYS
            ORDER BY metrics.cost_micros DESC"
}
```

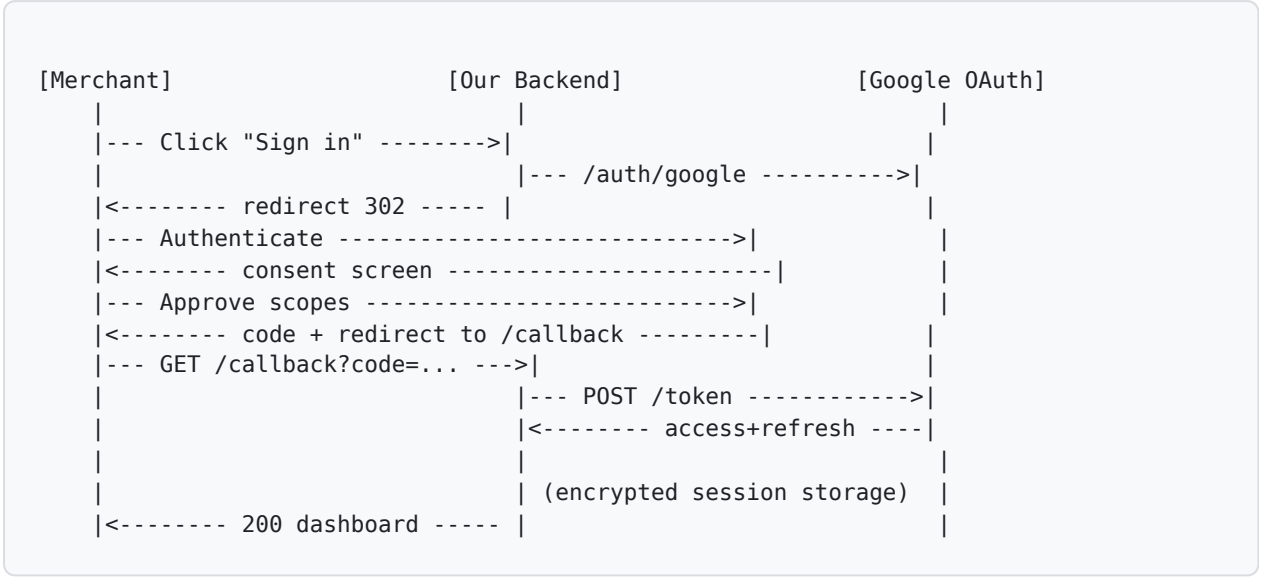
4.3. Estimated API Volume

OPERATION	FREQUENCY	MONTHLY VOLUME (YEAR 1)
CreateCustomerClient (onboarding)	Per new merchant	~8 / month
Campaign management (mutate)	Daily aggregate	~3,000
Reporting queries (search/searchStream)	Daily per merchant	~9,000
Conversion tracking lookups	Hourly aggregate	~24,000
Total API operations		**~36,000 / month**

> Volumes are expected to grow linearly with merchant count, capped at our infrastructure tier. We will request access tier upgrades through proper channels as needed.

5. OAuth 2.0 Flow

5.1. Authentication Flow Diagram



5.2. Scopes Requested

SCOPE	PURPOSE	JUSTIFICATION
`auth/userinfo.email`	Identify the user	Required for session identity
`auth/userinfo.profile`	Display name/picture	Personalized UI greeting
`auth/adwords`	Google Ads API access	Core platform functionality
`auth/content`	Merchant Center API	Product catalog sync
`auth/siteverification`	Domain verification	Required for Merchant Center setup

5.3. Token Lifecycle

- Access tokens are stored encrypted in server-side sessions only. No client-side storage of OAuth tokens.
- Refresh tokens are stored separately and used only to obtain new access tokens when needed.
- Tokens can be revoked at any time by the merchant via `myaccount.google.com/permissions`.
- Session expiration: 24 hours of inactivity triggers a fresh login.
- HTTP-only and Secure cookies; SameSite=Lax.

6. Data Model

6.1. Entities

```
Merchant {
  id:                   UUID
  email:                String        (Google OAuth)
  name:                 String
  googleAdsCustomerId: String        (after invitation accepted)
  merchantCenterId:     String        (after Merchant Center linked)
  domain:               String        (verified domain)
  status:               enum          (PENDING|ACTIVE|SUSPENDED)
  createdAt:            Timestamp
  lastActiveAt:         Timestamp
}

Invitation {
  id:               String
  merchantEmail:    String
  invitationUrl:    String           (from createCustomerClient response)
  ivid:             String           (extracted from URL)
  customerId:       String           (Ads customer ID created)
  sentAt:           Timestamp
  acceptedAt:       Timestamp?
}

Campaign {
  id:                   Long          (Google Ads campaign ID)
  merchantId:           UUID          (FK to Merchant)
  customerId:           String
  name:                 String
  advertisingChannelType: enum
  status:               enum
  budgetMicros:         Long
  syncedAt:             Timestamp
}
```

6.2. Data Retention

- OAuth tokens: deleted upon merchant logout or session expiry (24h).
- Merchant data: retained while account is active; deleted within 30 days of account closure.
- Performance metrics: aggregated and stored up to 24 months for trend analysis.
- Audit logs: retained for 12 months for security and compliance review.
- Invitation records: deleted 90 days after acceptance or expiry.

7. Security and Compliance

7.1. Security Measures

LAYER	CONTROL
Transport	HTTPS only (TLS 1.3); HSTS enabled
Authentication	OAuth 2.0 with PKCE; no password storage
Session	Encrypted server-side; HttpOnly, Secure, SameSite cookies
Authorization	Scope-minimization; explicit user consent per scope
API Tokens	Developer token stored in encrypted environment variables, never logged
Input Validation	All user input sanitized; SQL injection / XSS prevention
Rate Limiting	Per-IP and per-user rate limits to prevent API abuse
Monitoring	Real-time anomaly detection on API call patterns
Backups	Daily encrypted backups; tested restoration procedures

7.2. Regulatory Compliance

- **LGPD (Lei 13.709/2018):** Brazilian data protection law (GDPR equivalent). Our platform implements all required controls: lawful basis (consent), data minimization, purpose limitation, transparency, retention limits, user rights (access, correction, deletion, portability).
- **Google API Services User Data Policy:** We comply with the Limited Use Policy. We do not transfer user data to third parties except as necessary to provide the service or as required by law. We do not use Google user data for advertising. We do not allow humans to read user data unless explicit consent is obtained or required for security, legal compliance, or limited support operations.
- **Google Ads API Terms of Service:** We agree to and follow the Required Minimum Functionality (RMF) standards for all tools accessing the API.

7.3. Privacy Policy and Terms

- Privacy Policy: `https://marcionogueira.com.br/privacy`
- Terms of Use: `https://marcionogueira.com.br/terms`
- Contact for privacy inquiries: `marcio4146@gmail.com`

8. Required Minimum Functionality (RMF) Compliance

Per Google's RMF requirements, our tool implements the following functionalities beyond simple data retrieval:

8.1. Account Management

- ✅ Create new customer clients (sub-accounts) via API
- ✅ Configure account-level settings (timezone, currency, descriptive name)
- ✅ Manage user access and invitations

8.2. Campaign Management

- ✅ Create campaigns (Shopping, Performance Max)
- ✅ Edit campaign settings, budgets, and bidding strategies
- ✅ Pause, enable, and remove campaigns
- ✅ Create and manage ad groups and product groups

8.3. Conversion Tracking

- ✅ Create conversion actions for merchant goals
- ✅ Configure conversion tracking with Google Tag
- ✅ Report on conversion performance

8.4. Reporting

- ✅ Custom reports beyond what the standard Google Ads UI provides
- ✅ Cross-account aggregated views for merchants managing multiple stores
- ✅ Product-level performance correlated with Merchant Center catalog

9. Operational Practices

9.1. Error Handling

We implement exponential backoff for transient errors, parse Google's structured error responses, and surface actionable feedback to merchants. We respect quota limits and implement client-side rate limiting to stay within Google's published thresholds.

9.2. Monitoring and Logging

All API calls are logged with anonymized identifiers (no PII in logs). We monitor for unusual patterns (e.g., spike in error rates) and pause batch operations automatically when anomalies

are detected.

9.3. Customer Support

Support is provided in Portuguese during Brazilian business hours via email (`marcio4146@gmail.com`), WhatsApp ((69) 99272-9912), and in-app messaging. Critical issues are addressed within 4 business hours.

10. Contact Information

COMPANY

Marcio Nogueira
Brazil
marcionogueira.com.br

API CONTACT

Email: marcio4146@gmail.com
Phone: +55 (69) 99272-9912
WhatsApp: wa.me/5569992729912

This document was prepared as part of the Google Ads API Token application. All information herein is accurate to the best of our knowledge as of the document date. We accept responsibility for the practices described and commit to operating in compliance with Google's API Terms of Service and User Data Policy.

Document version: 1.0 · **Date:** May 24, 2026