

PROFIT-FIRST AD MANAGER

Software Requirements Specification

Version 2.0 | Enterprise Edition

Document Status	Draft for Review
Version	2.0
Date	February 2026
Author	Product Team
Classification	Confidential
Reviewers	Engineering, Product, QA, Business

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) defines the complete functional, non-functional, architectural, and operational requirements for the Profit-First Ad Manager (PFAM) — a cloud-hosted SaaS platform that enables direct-to-consumer (D2C) e-commerce brands to measure, monitor, and automate their advertising campaigns based on true net profit rather than vanity metrics like Revenue-on-Ad-Spend (ROAS).

This document is the single authoritative reference for all design, development, QA, DevOps, and business stakeholder decisions relating to PFAM v2.0. All implementation decisions must be traceable back to a requirement defined herein.

1.2 Scope

PFAM is a multi-tenant SaaS application that:

- Connects to commerce platforms (Shopify initially; WooCommerce in future) to ingest orders, line items, COGS, product metadata, and refunds in real time.
- Connects to advertising platforms (Meta Ads, Google Ads, TikTok Ads) via OAuth to ingest ad spend, impressions, clicks, and conversion events at the campaign / ad set / ad level.
- Applies a multi-tiered attribution engine to match revenue from orders to the ad campaign that drove that purchase.
- Computes true Net Profit per campaign using the formula: Revenue - Ad Spend - COGS - Estimated Returns - Platform Fees.
- Provides a rules engine that allows merchants to define profit-based conditions and trigger automated actions (pause, scale, alert) directly via Ads APIs.
- Delivers dashboards, exportable reports, and notification integrations (Slack, Email, Webhooks) to give business stakeholders actionable, real-time profit visibility.

1.3 Definitions, Acronyms & Abbreviations

Term	Definition
PFAM	Profit-First Ad Manager — the product described by this document.
D2C	Direct-to-Consumer — e-commerce brands selling directly to end consumers.
COGS	Cost of Goods Sold — direct costs attributable to producing/purchasing a product.

Term	Definition
ROAS	Return on Ad Spend — Revenue / Ad Spend. Does not account for COGS or returns.
Net Profit	Revenue - Ad Spend - COGS - Estimated Returns - Platform Fees.
Attribution	The process of crediting a marketing touchpoint for driving a conversion/order.
Click ID	A unique identifier appended to landing page URLs by Meta (fbclid) or Google (gclid).
Conversion ID	Platform-specific identifier linking a purchase event to an ad click.
SKU	Stock Keeping Unit — a unique identifier for a product variant.
AOV	Average Order Value — total revenue divided by number of orders.
CAC	Customer Acquisition Cost — spend required to acquire one new customer.
LTV	Lifetime Value — projected net profit from a customer over their relationship.
MRR	Monthly Recurring Revenue — total recurring subscription revenue per month.
RBAC	Role-Based Access Control — permission model based on user roles.
ETL	Extract, Transform, Load — data pipeline process.
SLA	Service Level Agreement — contractual commitment on uptime/response times.
WAF	Web Application Firewall — network security component for HTTP traffic.
PII	Personally Identifiable Information — data that identifies an individual.
OAuth	Open Authorization — standard protocol for secure, delegated access to third-party APIs.
JWT	JSON Web Token — compact, self-contained token for authentication.
EKS	Elastic Kubernetes Service — AWS managed Kubernetes.
GKE	Google Kubernetes Engine — GCP managed Kubernetes.

Term	Definition
SOC 2	System and Organization Controls 2 — security compliance framework.
GDPR	General Data Protection Regulation — EU data privacy regulation.
CCPA	California Consumer Privacy Act — California data privacy regulation.

1.4 Intended Audience

Audience	Relevant Sections
Engineering (Front-end, Back-end, Data)	3, 5, 6, 7, 8, 9, 12, 13
QA / Testing	6, 13, 14
DevOps / Platform Engineering	7, 12, 15
Product Management	3, 4, 5, 14
Security & Compliance	6, 10
Business / Investors	2, 3, 4, 16
Customer Support / Operations	15

1.5 Document Overview

This SRS is organized as follows:

- Section 1 — Introduction, purpose, scope, and terminology.
- Section 2 — Executive Summary: Market opportunity, competitive landscape, and PFAM differentiators.
- Section 3 — Overall Description: Product perspective, key features, constraints, and assumptions.
- Section 4 — User Personas and Use Cases.
- Section 5 — Functional Requirements (FR) with detailed acceptance sub-criteria.
- Section 6 — Non-Functional Requirements (NFR): Performance, security, scalability, compliance.
- Section 7 — System Architecture and Data Flow.
- Section 8 — Data Model and Schemas.
- Section 9 — API Specifications.
- Section 10 — Security, Privacy, and Compliance.
- Section 11 — Integration and Third-party Dependencies.

- Section 12 — Deployment, Scalability, and Monitoring.
- Section 13 — Testing and QA Strategy.
- Section 14 — Acceptance Criteria.
- Section 15 — Operations and Maintenance.
- Section 16 — Risks and Mitigations.
- Section 17 — Appendix: Glossary and Revision History.

2. Executive Summary & Market Research

2.1 Problem Statement

The digital advertising ecosystem is built around revenue-centric metrics. Ad platforms such as Meta Ads, Google Ads, and TikTok Ads optimize campaign delivery for maximizing conversions and Return on Ad Spend (ROAS). While ROAS tells a merchant how many dollars of revenue were generated per dollar of ad spend, it fundamentally ignores the costs that determine actual business profitability:

$$\text{Net Profit} = \text{Revenue} - \text{Ad Spend} - \text{Cost of Goods Sold} - \text{Estimated Returns} - \text{Platform \& Payment Fees}$$

A brand may report a 3.5x ROAS which appears healthy, yet after accounting for a 45% COGS ratio, a 12% return rate, and 3% payment processing fees, the actual net profit margin is less than 5%. Without a unified system that integrates commerce data with advertising data, brands routinely scale unprofitable campaigns, destroying working capital at growth speed.

Industry research indicates that 60-70% of D2C brands do not have real-time visibility into per-campaign profitability, and that 15-25% of monthly ad spend at a typical brand is allocated to campaigns that are net-negative when fully-loaded costs are considered.

2.2 Market Opportunity

Market Segment	Size	Notes
Total Active Shopify Stores	4.4 million	Global, 2024 estimate
Stores running paid advertising	~1.1 million	~25% of total
Stores spending >\$5K/month on ads	~300,000	PFAM primary target
TAM at \$300/month avg. subscription	\$1.08B ARR	US Letter approx.
SAM — English-speaking markets, Shopify-first	~\$540M ARR	~150K merchants
SOM — Year 3 target (1% penetration)	\$5.4M ARR	1,500 customers

2.3 Competitive Landscape

The market has multiple adjacent players, none of which provide the full combination of accurate profit calculation, SKU-level returns handling, and automated action execution:

Competitor	Category	Strengths	Key Gap vs PFAM
TripleWhale	Attribution Analytics	Strong brand, Shopify-native, creative analytics	Analytics-only; no automated ad actions
Northbeam	Multi-touch Attribution	Sophisticated attribution, enterprise features	Complex setup, expensive (\$1K+/mo), no automation
Lifetimely	LTV & Profit Analytics	Customer cohort analysis, LTV tracking	Weak returns modeling, no campaign automation
Hyros	Attribution & Call Tracking	High-ticket offer tracking, AI attribution	Not e-commerce-first, very expensive
Elevor / Littledata	Server-side Tracking	Accurate event tracking, GA4 integration	Data pipeline only, no profit calculation
Custom BI (Tableau/Looker)	Enterprise Analytics	Fully customizable, powerful	\$50K+ setup, 3-6 month deployment, no automation
Native Ad Platform Tools	Channel-specific Analytics	Free, integrated	No COGS/returns, siloed per platform, no cross-channel view

2.4 PFAM Unique Value Propositions

Differentiator	Description
Profit-first automation	The only SMB-accessible tool that executes API-level ad actions (pause, budget change) based on profit thresholds — not just surface insights.
SKU-level returns modeling	Per-SKU return rate history from Shopify Refunds API applied as a cost in net profit calculation. Retroactively updated as actual refunds arrive.
Multi-tiered attribution	Order-level click ID matching as primary method; SKU-weighted, blended, and ML-predicted probabilistic attribution as fallbacks.
Simple, CEO-friendly UX	Dashboards designed for founders and operators, not data analysts. Clear red/green indicators, plain-language rule builder, actionable alerts.
Shopify-first quick onboarding	Full OAuth connection and historical data import in under 10 minutes. No tagging, no GTM setup, no engineering work.

Differentiator	Description
Transparent profit methodology	Every number shows its calculation components. Users can see exactly why a campaign was paused. Confidence scores surface data quality.
SMB-oriented pricing	Ad-spend-tiered plans starting at \$99/month, designed for brands spending \$5K-\$500K/month.

3. Overall Description

3.1 Product Perspective

PFAM is a standalone cloud-hosted SaaS product that operates at the intersection of three existing systems: commerce platforms, advertising platforms, and the merchant's own cost and returns data. PFAM does not replace these systems; it reads from them (and in the case of ad platforms, writes back to them for automated actions). PFAM adds the profit intelligence layer that none of these systems provide natively.

The product is organized into four logical layers:

- Data Ingestion Layer — connectors that pull and normalize data from external platforms.
- Intelligence Layer — the attribution engine and profit calculation engine that derive meaning from raw data.
- Automation Layer — the rules engine and action executor that take actions on external platforms based on profit signals.
- Presentation Layer — dashboards, reports, alerts, and API endpoints that surface insights and enable user interaction.

3.2 Key Features Summary

Feature Group	Components
Platform Connectors	Shopify (orders, products, COGS, refunds), Meta Ads, Google Ads, TikTok Ads (Phase 2)
Profit Calculation Engine	Revenue attribution, COGS aggregation, returns modeling, fee estimation, net profit computation
Attribution Engine	Click ID / Conversion ID direct matching, SKU-weighted fallback, blended fallback, ML-predicted fallback
Automation Rules Engine	Condition builder (rolling windows, thresholds), action execution (pause, scale, alert), audit trail
Dashboards & Reports	Account overview, Campaign view, Ad Set view, Product/SKU profitability, exportable CSV/PDF
Notifications & Integrations	Slack, Email, Webhooks for rule triggers and system alerts
User Management	Multi-tenant, RBAC (Admin, Analyst, Read-only), SSO for enterprise
Billing & Subscriptions	Stripe-powered subscription tiers based on monthly ad spend, usage metering

Feature Group	Components
Admin Console	Internal tools for support, overrides, audit log viewing, tenant management

3.3 User Characteristics

PFAM is designed for non-technical business users. The typical user is a D2C brand founder or performance marketer who understands advertising concepts (campaigns, ad sets, ROAS) but is not expected to understand data engineering, SQL, or API configuration. The product must present complex analytical concepts in plain language with visual aids.

Technical users (growth engineers, developers) will interact with PFAM primarily via the API, connector configuration, and attribution tuning interfaces. These users expect verbose logging, confidence scores, and raw data export capabilities.

3.4 Constraints

- PFAM is dependent on the API availability and rate limits of Meta Ads, Google Ads, and Shopify. Any platform deprecation or policy change may affect feature availability.
- Automated ad actions are subject to platform policies. Currently, Meta and Google Ads permit third-party pausing and budget modification via API with appropriate OAuth scopes.
- PFAM is not an accounting system and does not replace a merchant's ERP, general ledger, or tax software. Profit figures are estimates based on available data.
- Attribution is inherently probabilistic post-iOS 14.5. PFAM must communicate confidence levels to users and default to conservative actions on low-confidence data.
- All processing of EU resident data must occur in EU-region cloud infrastructure to comply with GDPR data residency requirements.

3.5 Assumptions & Dependencies

- Merchants have an active Shopify store and at least one Meta Ads or Google Ads account connected.
- Merchants either have COGS data in Shopify or are willing to provide a COGS percentage per SKU or product category.
- Platform APIs remain available and backward-compatible for a minimum of 12 months following each version release.
- AWS (primary) or GCP (secondary) infrastructure is available with 99.9%+ uptime for cloud services used.
- Stripe remains the payment gateway for subscription billing for the foreseeable future.

4. User Personas & Use Cases

4.1 Primary Personas

Persona	Role	Goals	Pain Points	Key PFAM Features Used
Founder / CEO (Alex, 34)	Owns D2C brand, \$50K/mo ad spend	Ensure every ad dollar is profitable; sleep knowing bad campaigns are paused automatically	Drowning in dashboards; spends Sunday evenings in Excel; afraid of wasting cash	Account overview, automation rules, daily email digest, mobile alerts
Performance Marketer (Priya, 29)	In-house media buyer managing Meta + Google	Optimize campaigns quickly; justify budget decisions with data	Can't get unified profit view across platforms; attribution gaps after iOS 14	Campaign detail view, attribution inspector, multi-platform dashboards
Finance Manager (David, 41)	Controller / accountant at growing D2C brand	Reconcile ad spend with actual profit; export data for accounting system	Ad metrics don't match P&L; can't link COGS to specific campaigns	CSV exports, COGS breakdown reports, reconciliation view
Agency Account Manager (Sarah, 31)	Manages 15 D2C clients at Shopify Plus agency	Monitor profit across all clients from one dashboard; report ROI to clients	Each client has different setup; manual reporting takes 20 hrs/week	Multi-tenant agency dashboard, white-label reports, client-level rules
Growth Engineer (Kenji, 27)	Technical lead at mid-market brand	Validate tracking accuracy; customize attribution; build integrations	Attribution black box; can't trust numbers without auditability	API access, attribution debug mode, raw data export, connector logs

4.2 Primary Use Cases

UC-01	Onboarding & Platform Connection Actor: Founder / Growth Engineer
Goal	Merchant successfully connects Shopify + Meta, views first profit dashboard.
Steps	<ol style="list-style-type: none"> 1. Merchant clicks "Connect Shopify" and completes OAuth flow in under 10 minutes. 2. PFAM imports last 90 days of orders, line items, COGS, and refunds. 3. Merchant connects Meta Ads account via OAuth and selects ad accounts. 4. PFAM imports last 90 days of campaign, ad set, ad spend, and conversion data. 5. Attribution engine runs initial matching pass. Dashboard populates. 6. Merchant receives onboarding email confirming setup completion.

UC-02	Profit Calculation & Dashboard Review Actor: Founder / Performance Marketer
Goal	Merchant reviews true net profit per campaign across all connected platforms.
Steps	<ol style="list-style-type: none"> 7. Merchant opens Campaign View. Sees all campaigns ranked by net profit. 8. Each row shows: Revenue, Ad Spend, COGS, Returns, Fees, Net Profit, Net Profit %. 9. Color coding: green (profitable), amber (near-zero), red (unprofitable). 10. Merchant can filter by platform, date range, product category, or attribution confidence. 11. Clicking a campaign opens the attribution inspector showing matched orders.

UC-03	Automation Rule Creation Actor: Founder / Performance Marketer
Goal	User creates a rule to automatically pause unprofitable ad sets.
Steps	<ol style="list-style-type: none"> 12. User navigates to "Automation Rules" and clicks "Create Rule". 13. Selects template: "Pause if unprofitable over rolling window." 14. Configures: rolling window = 7 days; threshold = net profit < \$0; platform = Meta Ads. 15. Sets guardrails: minimum 15 attributed orders before rule can trigger. 16. Sets notification: email + Slack on trigger. 17. Activates rule. PFAM validates configuration and shows rule summary.

UC-04	Automated Action Execution Actor: System (Background Worker)
Goal	PFAM automatically pauses an unprofitable ad set and notifies the user.
Steps	<ol style="list-style-type: none"> 18. Daily sync completes. Profit engine recalculates all ad set metrics. 19. Rule engine evaluates UC-03 rule: Ad Set A shows -\$412 net profit over 7 days. 20. Worker enqueues pause action for Ad Set A on Meta Ads. 21. Worker calls Meta Marketing API: pause ad set. API returns 200 success.

22. Audit log entry created: {actor: system, action: pause, adset_id, rule_id, timestamp, profit_value}.
23. Email + Slack notification fired: "Ad Set A paused — 7-day net profit: -\$412."

UC-05	Attribution Inspector & Override Actor: Performance Marketer / Growth Engineer
Goal	User investigates attribution for a specific campaign and overrides an action.
Steps	<p>24. User notices a paused campaign they believe should be active.</p> <p>25. Opens attribution inspector: sees matched orders, confidence scores, attribution method per order.</p> <p>26. Sees 8 orders with "Low" confidence (SKU-weighted fallback used).</p> <p>27. Reviews raw click data. Decides to whitelist this campaign from automation.</p> <p>28. Unpauses campaign and adds exception. Audit log records manual override.</p> <p>29. User can adjust attribution method for this campaign independently.</p>

UC-06	COGS & Returns Configuration Actor: Finance Manager / Founder
Goal	User configures accurate COGS and return rates to improve profit accuracy.
Steps	<p>30. User navigates to "Product Settings" → "COGS & Returns".</p> <p>31. Option A: Import from Shopify (if COGS field populated in Shopify). One-click import.</p> <p>32. Option B: Upload CSV with columns: SKU, COGS, Return Rate %.</p> <p>33. Option C: Set category-level defaults (e.g., "Apparel: 40% COGS, 15% return rate").</p> <p>34. PFAM recalculates all historical profit metrics with new data.</p> <p>35. Validation warnings shown if COGS seems implausibly high/low for a category.</p>

UC-07	Report Export & Finance Reconciliation Actor: Finance Manager
Goal	Finance team exports profit data for monthly accounting reconciliation.
Steps	<p>36. User goes to "Reports" and selects date range and platforms.</p> <p>37. Selects report type: "Campaign Profit Export" (CSV) or "Summary Report" (PDF).</p> <p>38. CSV includes: campaign, ad spend, revenue, COGS, returns, fees, net profit per day.</p> <p>39. PDF report includes charts, executive summary, and methodology note.</p> <p>40. Finance team imports CSV into accounting software for reconciliation.</p>

UC-08	Agency Multi-Client Dashboard Actor: Agency Account Manager
Goal	Agency user monitors profit health across all client accounts in one view.

Steps

41. Agency user logs in and sees all client accounts on a single grid.
42. Each client card shows: Active Campaigns, MoM profit trend, alerts pending.
43. Red/amber/green status indicators per client.
44. Click into any client for full campaign-level view (same as single-tenant experience).
45. Export per-client or aggregate PDF reports with agency branding.

5. Functional Requirements

Each requirement below is identified by a unique ID (FR-XX), a priority level (High / Medium / Low), and a status (Mandatory / Desirable). Sub-bullet points define the specific, testable acceptance criteria for each requirement.

FR-01 User Registration & Authentication **[High]** Mandatory

Users shall be able to register, authenticate, and manage their account and organization settings securely.

Acceptance Criteria:

- Support email/password registration with email verification flow.
- Support Google OAuth 2.0 and Microsoft OAuth 2.0 as Social Sign-In options.
- Implement mandatory email verification before account activation.
- Support organization-level SSO via SAML 2.0 for Enterprise plan customers.
- Enforce multi-factor authentication (MFA) via TOTP (Google Authenticator, Authy) as optional for all users and mandatory for Admin role.
- Provide password reset flow via time-limited (1 hour) email tokens.
- Implement session management: JWT access tokens (15-min expiry) + refresh tokens (30-day expiry, rotated on use).
- Failed login attempts > 5 in 10 minutes triggers a 15-minute IP-level lockout and admin alert.
- All auth events (login, logout, password reset, role change) persisted to an immutable audit log.

FR-02 Organization & Team Management **[High]** Mandatory

The system shall support multi-user organizations with role-based access control (RBAC).

Acceptance Criteria:

- Organizations can have unlimited users on Growth+ plans; 1 user on Starter.
- User roles: Owner, Admin, Analyst, Read-Only.
- Owners and Admins can invite users by email; invitations expire after 7 days.
- Owners can transfer organization ownership to another Admin.
- Removing a user immediately revokes all active sessions for that user.
- All role assignments and changes are logged with actor, timestamp, and previous role.
- Agency plan supports a parent organization with sub-organizations (client accounts) per client.
- Sub-organization users can be scoped to specific client accounts only.

FR-03 Shopify Connector **[High]** Mandatory

The system shall connect to a Shopify store via OAuth and synchronize all commerce data required for profit calculation.

Acceptance Criteria:

- Initiate Shopify OAuth using the required API scopes: read_orders, read_products, read_inventory, read_reports, read_fulfillments.
- On first connection, import historical data for the last 90 days (configurable up to 365 days on Enterprise).
- Ongoing sync: new and updated orders ingested within 60 minutes of creation/modification.
- Ingest order-level data: order_id, created_at, total_price, total_discounts, currency, customer_id.
- Ingest line-item-level data: product_id, variant_id, sku, quantity, price, total_discount.
- Ingest COGS from Shopify cost_per_item field on variants where available.
- Ingest all refund and partial refund events with line-item granularity.
- Ingest product and variant metadata: title, tags, product_type, vendor.
- Handle multi-currency stores: convert all amounts to organization's base currency using daily FX rates from an exchange rate API.
- Shopify API rate limits are managed with a token bucket algorithm; requests queued and retried with exponential backoff.
- Connector status visible on settings page: last sync time, total orders synced, error count.

FR-04 Meta Ads Connector [High] Mandatory

The system shall connect to Meta Business Manager via OAuth and retrieve advertising data at all hierarchy levels.

Acceptance Criteria:

- Initiate Meta OAuth requesting scopes: ads_read, ads_management (for pause/budget actions).
- Support connecting multiple ad accounts per organization.
- Import campaign, ad set, and ad metadata: ids, names, status, objectives, targeting summary.
- Import daily insight metrics: spend, impressions, clicks, reach, CPM, CPC, CTR, conversions.
- Import conversion events with action_type breakdown for purchase, add_to_cart, initiate_checkout.
- Import click-level data via Meta's offline conversion matching where available.
- Support both Account Currency and USD reporting with automatic conversion.
- Handle Meta's 7-day, 1-day click attribution windows. Map to PFAM's configurable attribution window.
- Detect and handle rate limit (Error Code 17) via backoff queuing.
- Log API calls with timestamps, endpoints, and response codes for debugging.

FR-05 Google Ads Connector [High] Mandatory

The system shall connect to Google Ads via OAuth and retrieve campaign and conversion data.

Acceptance Criteria:

- Initiate Google Ads OAuth requesting scope: <https://www.googleapis.com/auth/adwords>.
- Import campaign, ad group, and ad metadata with status and bidding strategy.
- Import daily metrics: cost, impressions, clicks, conversions, conversion_value per campaign and ad group.
- Import conversion actions with type classification (purchase, lead, etc.).
- Map Google Click ID (gclid) to orders via Shopify order UTM parameters where available.
- Support Google Ads Manager Account (MCC) for agency clients with multiple sub-accounts.

- Handle Google Ads API quota limits (10,000 operations per day default) with throttling.

FR-06 TikTok Ads Connector (Phase 2) [Medium] Desirable

The system shall connect to TikTok Ads Manager and retrieve ad spend and conversion data.

Acceptance Criteria:

- Initiate TikTok OAuth and retrieve campaign, ad group, and ad data.
- Import spend, impressions, clicks, and purchase conversion events.
- Map TikTok Click ID (ttclid) to Shopify orders via UTM parameters.
- Expose as an add-on feature at an additional monthly fee.

FR-07 Data Synchronization [High] Mandatory

The system shall keep all imported data current with configurable sync frequencies.

Acceptance Criteria:

- Default sync schedule: daily at 3:00 AM merchant's timezone.
- Hourly sync available on Growth plan and above.
- Manual "Sync Now" button available in UI, with a 5-minute cooldown per platform.
- Sync jobs are idempotent: re-running a sync for the same time window must not create duplicate records.
- Sync failures trigger a retry queue (3 retries with exponential backoff: 5 min, 15 min, 60 min).
- Persistent sync failures (after all retries) send an email + Slack alert to the organization Admin.
- Sync job logs available in UI under "Connector Health": status, duration, records imported, errors.
- Stale data indicator shown in dashboard if last successful sync > 25 hours ago.

FR-08 Attribution Engine [High] Mandatory

The system shall attribute revenue from orders to advertising campaigns using a multi-tiered methodology.

Acceptance Criteria:

- Tier 1 (Direct Match): Match order to ad click using fbclid, gclid, or ttclid stored in Shopify order's UTM source/medium fields or note attributes.
- Tier 2 (Conversion ID Match): Match Shopify purchase event fired via Meta Pixel or Google Tag to ad campaign using conversion event metadata.
- Tier 3 (SKU-Weighted Attribution): For orders where direct match fails, distribute revenue across campaigns proportionally to each campaign's share of spend on products matching the order's SKUs in the attribution window.
- Tier 4 (Blended Attribution): Distribute unmatched revenue proportionally across all active campaigns by spend volume within the attribution window.
- Tier 5 (ML-Predicted): Apply a trained gradient boosting model using features: order time, SKU category, customer geography, session path (if available) to predict most likely source campaign.
- Attribution window is configurable per organization: 1-day, 7-day, 14-day, 30-day click window.

- Each attributed order is tagged with: `attribution_tier` (1-5), `confidence_score` (0.0-1.0), `attribution_method`, `matched_click_id` (if available).
- Confidence thresholds: Tier 1 = 0.95, Tier 2 = 0.85, Tier 3 = 0.70, Tier 4 = 0.50, Tier 5 = 0.60-0.85 (model-dependent).
- Attribution conflicts (same order attributed by multiple platforms) resolved by a platform priority ranking configurable by user.
- Campaign-level attribution totals reconciled against total order count and revenue. Unmatched orders clearly flagged.
- Attribution inspector UI: per-order view showing attribution decision path and contributing factors.

FR-09 Profit Calculation Engine [High] Mandatory

The system shall calculate Net Profit per ad campaign / ad set / ad using all available cost components.

Acceptance Criteria:

- Net Profit formula: $\text{Net Profit} = \text{Attributed Revenue} - \text{Ad Spend} - \text{Attributed COGS} - \text{Estimated Returns} - \text{Estimated Platform Fees}$.
- Attributed Revenue: sum of order line-item revenues for orders attributed to the campaign in the window.
- Ad Spend: actual spend ingested from ad platform API for the campaign/window.
- Attributed COGS: sum of ($\text{unit_cogs} \times \text{quantity}$) for all line items in attributed orders. Where Shopify COGS is unavailable, use merchant-configured COGS percentage or category default.
- Estimated Returns: sum of ($\text{order_revenue} \times \text{sku_return_rate}$) for all attributed orders. `sku_return_rate` derived from trailing 180-day actual return rate per SKU.
- Estimated Platform Fees: configurable percentage (default 2.9% + \$0.30 per transaction) applied to attributed revenue.
- All calculations stored as immutable snapshots at each sync cycle with a version timestamp.
- Historical recalculation triggered automatically when COGS or return rate data is updated.
- Profit metrics surfaced for: daily, 7-day rolling, 14-day rolling, 30-day rolling, custom date range.
- All monetary values stored in cents (integer) to avoid floating-point precision issues.
- Net Profit % (margin) = $\text{Net Profit} / \text{Attributed Revenue} \times 100$, displayed alongside absolute profit.
- True ROAS = $\text{Attributed Revenue} / \text{Ad Spend}$ (always shown alongside Net Profit for context).
- Blended metrics available at Account level aggregating across all connected platforms.

FR-10 Returns & Refund Modeling [High] Mandatory

The system shall model expected and actual returns at the SKU level for accurate net profit calculation.

Acceptance Criteria:

- Ingest all Shopify refunds including partial refunds with line-item granularity.
- Compute trailing 90-day and 180-day return rate per SKU: $\text{returns_rate} = \text{refunded_units} / \text{sold_units}$.
- Apply return rate to attributed orders: $\text{expected_return_cost} = \text{order_revenue} \times \text{sku_return_rate}$.

- When actual refund is processed, retroactively update the profit metric for the original campaign.
- Return Lag Adjustment: model expected returns for orders < 45 days old as "return reserve" (unrealized).
- After 45 days, transition from expected to actual return figures.
- Users can manually override return rates per SKU in Product Settings.
- Return reason classification from Shopify refund notes: Defective, Wrong Item, Changed Mind, Sizing.
- Defective/Fulfillment returns flagged separately (not attributed to ad campaign performance).
- Return rate benchmarks by product category displayed in UI for context.
- Return reserve amounts clearly labeled in profit breakdown: "Includes \$X estimated unrealized returns."

FR-11 Automation Rules Engine [High] Mandatory

Users shall be able to define conditional rules that trigger automated actions on ad platforms when profit thresholds are met.

Acceptance Criteria:

- Rule components: Name, Platform (Meta / Google / TikTok), Scope (Account / Campaign / Ad Set / Ad), Condition(s), Action, Guardrails, Notification.
- Conditions supported: $\text{rolling_net_profit} < X$, $\text{rolling_net_profit_margin} < X\%$, $\text{rolling_roas} < X$, $\text{spend_to_date} > X$, $\text{rolling_return_cost} > X$.
- Rolling window options: 1, 3, 7, 14, 30 days.
- Compound conditions: AND / OR logic for up to 5 conditions per rule.
- Actions supported: Pause entity, Reduce daily budget by $X\%$, Increase daily budget by $X\%$, Send alert only.
- Guardrails: minimum attributed orders threshold before rule can trigger (default: 10 orders), minimum spend threshold (\$50), maximum pauses per day per account (configurable, default 10).
- Guardrails prevent premature automation on low-data campaigns.
- Rule activation is explicit (user must toggle ON). New rules are always created in OFF/Draft state.
- Rule evaluation runs after each sync cycle.
- User can preview rule behavior against historical data before activating ("Backtest Rule").
- Rules are platform-specific: a Meta rule cannot act on Google campaigns.
- Users can whitelist specific campaigns or ad sets from all automation rules.

FR-12 Action Execution & Audit Trail [High] Mandatory

The system shall execute authorized actions on ad platforms and maintain a complete, immutable audit trail of all actions.

Acceptance Criteria:

- Actions queued in a persistent job queue (Kafka) before execution.
- Each action includes: `entity_type`, `entity_id`, `platform`, `action_type`, `triggering_rule_id`, `triggering_metric_values`, `expected_outcome`.
- Actions executed via the respective platform API with the OAuth token of the connected ad account.

- API response (success or error) recorded in the audit log.
- Audit log fields: audit_id, org_id, actor (user_id or "system"), action_type, entity_id, platform, rule_id, metric_snapshot_json, api_response_code, created_at.
- Audit log is append-only and immutable (no update/delete operations).
- All actions are reversible from the UI (un-pause, reset budget) with confirmation dialog.
- Manual user actions (from UI) also recorded in audit log with user as actor.
- Audit log searchable and filterable by date, actor, action type, campaign, and platform.
- Audit log exportable to CSV.
- Failed actions retried up to 3 times. Persistent failure triggers admin notification.
- Daily action digest email sent to Admin: summary of automated actions taken.

FR-13 Dashboards & Reports [High] Mandatory

The system shall provide visual dashboards and exportable reports at multiple levels of granularity.

Acceptance Criteria:

- Account Overview Dashboard: total spend, total revenue, total COGS, total returns, total net profit, and net profit % for all connected platforms in the selected date range.
- Campaign View: sortable table of all campaigns with columns: Platform, Campaign Name, Spend, Revenue, COGS, Returns, Net Profit, Net Profit %, True ROAS, Attribution Confidence, Status.
- Ad Set / Ad Group View: same columns as Campaign View, drillable from campaign row.
- Ad-level View: individual ad performance with creative thumbnail (where available via API).
- Product / SKU Profitability View: profit per product and per SKU across all campaigns.
- Attribution Inspector: order-level view for a specific campaign showing each attributed order, its attribution tier, confidence score, and cost breakdown.
- Time-series charts: daily profit trend line for selected campaigns, spend vs profit overlays.
- Platform Comparison View: side-by-side Meta vs Google (vs TikTok) profit comparison.
- Date range selector: presets (Today, Yesterday, Last 7/14/30/90 days) + custom range.
- Metric filters: filter by attribution confidence level, platform, product category, campaign status.
- All tabular data exportable to CSV. Summary views exportable to PDF with charts.
- Saved Views: users can save custom filter/column configurations and name them.
- Mobile-responsive design: dashboards usable on tablets and mobile browsers.

FR-14 Notifications & Alerts [High] Mandatory

The system shall send configurable notifications when rules trigger, anomalies are detected, or system events occur.

Acceptance Criteria:

- Notification channels: Email, Slack (via incoming webhook or Slack App), HTTP Webhook.
- Notification triggers: Rule triggered (action taken), Rule evaluated but guardrail blocked action, Sync failure (after retries), Attribution accuracy below threshold, Low data confidence warning, Subscription payment failed.
- Each notification type can be independently enabled/disabled per channel.
- Slack integration: connect to a Slack workspace, configure channel per notification type.
- Webhook integration: configure endpoint URL, secret key for HMAC signature verification.

- Webhook payload format: JSON with event_type, timestamp, org_id, entity details, and metric snapshot.
- Digest mode: option to batch multiple notifications into a single daily digest email at a configured time.
- Notification history viewable in-app for last 90 days.
- Email templates: HTML with PFAM branding, plain-text fallback, unsubscribe link.

FR-15 User Roles & Permissions (RBAC) [High] Mandatory

The system shall enforce role-based access control throughout all UI and API surfaces.

Acceptance Criteria:

- Roles: Owner (all permissions), Admin (all except billing/owner transfer), Analyst (view + rule management), Read-Only (view only).
- Permissions matrix enforced at API layer (not only UI).
- Analysts can create/edit/delete rules but cannot connect/disconnect platform accounts.
- Read-Only users cannot trigger manual syncs, create rules, or export data.
- All permission checks fail with 403 Forbidden and a clear error message.
- API tokens (for Growth Engineer use case) inherit the permissions of the user who generated them.
- Tokens can be scoped to specific permission subsets at creation time.

FR-16 Billing & Subscription Management [High] Mandatory

The system shall manage subscription billing via Stripe with plan tiers based on monthly ad spend.

Acceptance Criteria:

- Plan tiers: Starter (\$99/mo, up to \$10K ad spend), Growth (\$299/mo, \$10K-\$50K), Pro (\$599/mo, \$50K-\$150K), Scale (\$1,299/mo, \$150K-\$500K), Enterprise (custom).
- Monthly ad spend calculated as trailing 30-day sum of ad spend across all connected accounts.
- Automatic plan upgrade prompt shown when ad spend exceeds current plan threshold for 3 consecutive days.
- Automatic downgrade not applied automatically — only prompt sent with option to downgrade.
- Stripe Checkout for new subscriptions; Stripe Customer Portal for plan changes and cancellations.
- Failed payment retried per Stripe Smart Retries. On final failure: account suspended with grace period (7 days) and admin notification.
- Annual subscription option with 15% discount. Annual plans invoiced upfront.
- Usage metering endpoint reports monthly ad spend to Stripe for overage billing if applicable.
- All invoices and payment history accessible in-app under Billing.
- Tax calculation via Stripe Tax for applicable jurisdictions.

FR-17 COGS Management [High] Mandatory

The system shall provide flexible methods for merchants to configure and manage COGS data.

Acceptance Criteria:

- Method 1 — Shopify Import: One-click import of cost_per_item values from Shopify product variants.
- Method 2 — CSV Upload: Upload CSV with columns SKU, COGS Value, Currency. Validation on upload.
- Method 3 — Percentage Override: Set COGS as a % of revenue per product category or globally.
- Method 4 — Manual Entry: Edit COGS per SKU individually in UI.
- COGS data versioned: changes stored with timestamp, prior value, and actor.
- Validation warnings: flag SKUs where COGS > selling price, or COGS = \$0 for physical products.
- COGS confidence indicator on dashboards: "High" (Shopify import), "Medium" (CSV), "Low" (percentage estimate).
- Historical profit metrics automatically recalculated when COGS is updated.

FR-18 Admin Console [Medium] Mandatory

An internal admin UI shall allow Anthropic customer support staff to manage tenants, view logs, and perform overrides.

Acceptance Criteria:

- Accessible only to internal staff with admin_console role (separate from merchant roles).
- View all organizations with subscription status, MRR, last active, connected platforms.
- View any organization's sync logs, audit trail, and rule execution history.
- Manually trigger a sync for any organization.
- Modify subscription plan and billing status for support cases.
- Add/remove connectors and revoke OAuth tokens on behalf of a merchant.
- View and search system-wide error and event logs.
- Export organization data for GDPR data subject access requests.

FR-19 API for Developers [Medium] Desirable

PFAM shall expose a REST API allowing technical users and integration partners to query profit data programmatically.

Acceptance Criteria:

- API authentication via Bearer token (generated in UI, scoped to user's permissions).
- Endpoints: GET /api/v1/campaigns, GET /api/v1/campaigns/{id}, GET /api/v1/profit-metrics, GET /api/v1/orders/{id}/attribution.
- Rate limiting: 100 requests/minute per token by default; Enterprise: 1,000 requests/minute.
- All responses in JSON with consistent error format: {error_code, message, request_id}.
- API documentation available at /docs using OpenAPI 3.0 spec with interactive Swagger UI.
- Webhook event subscriptions manageable via API in addition to UI.
- API versioning via URL prefix (/api/v1/); previous version supported for 12 months after new version release.

6. Non-Functional Requirements

NFR-01 Performance [High] Mandatory

System must meet the following performance targets under normal operating conditions:

- Dashboard page load (first contentful paint): < 2.5 seconds for date ranges up to 30 days.
- Profit metric computation for an organization with 30,000 orders: < 10 minutes from sync job completion.
- Real-time rule evaluation after sync: < 5 minutes for up to 1,000 ad sets per organization.
- API response time (GET endpoints, p95): < 500ms.
- API response time (POST action endpoints, p95): < 2,000ms.
- CSV export generation (up to 1M rows): < 60 seconds.
- Search and filtering in UI: < 300ms for up to 10,000 campaigns.
- WebSocket push updates for live dashboard refresh: < 5 seconds from sync completion.

NFR-02 Scalability [High] Mandatory

System architecture must support the following scaling targets:

- Year 1 target: 10,000 active merchant organizations without performance degradation.
- Year 3 target: 100,000 active merchant organizations.
- Horizontal autoscaling of all stateless services (API, Worker, Profit Engine).
- Database partitioned by org_id (tenant) to prevent cross-tenant query interference.
- Read replicas for analytics queries to avoid impact on transactional write performance.
- Job queue throughput: 100,000 sync events per hour.
- Object storage (S3) used for large data exports — no direct file streaming from DB.
- Stateless API layer: no in-process session state; all state in Redis or PostgreSQL.

NFR-03 Availability & Reliability [High] Mandatory

The platform must meet the following availability commitments:

- 99.5% uptime SLA for all paid plans (Growth and above).
- 99.0% uptime SLA for Starter plan.
- Planned maintenance windows: maximum 4 hours per month; announced at least 72 hours in advance via status page and email.
- RTO (Recovery Time Objective): < 1 hour for Severity 1 incidents.
- RPO (Recovery Point Objective): < 1 hour for database data.
- Multi-AZ deployment for all stateful services (PostgreSQL, Redis, Kafka).
- Automated health checks and self-healing via Kubernetes liveness/readiness probes.
- Circuit breakers on all external API calls (Meta, Google, Shopify) to prevent cascade failures.

NFR-04 Security [High] Mandatory

All components must conform to the following security requirements:

- TLS 1.2 minimum (TLS 1.3 preferred) for all data in transit.
- AES-256 encryption for all OAuth tokens, API keys, and PII at rest (AWS KMS).
- No plaintext secrets in code, environment variables in plaintext, or logs.
- Secrets managed via HashiCorp Vault or AWS Secrets Manager.
- WAF (AWS WAF) protecting all public-facing endpoints with OWASP Top 10 rulesets.
- Rate limiting on all public endpoints: 1,000 req/min per IP for API; 60 req/min for auth endpoints.
- SQL injection, XSS, and CSRF protections enforced at framework level.
- CORS policy: API accessible only from PFAM frontend domain (no wildcard).
- Security vulnerability scanning in CI/CD pipeline (Snyk or equivalent).
- Dependency vulnerability alerts monitored and patched within 14 days for critical CVEs.
- Penetration testing: quarterly by third-party security firm.
- Bug bounty program established by end of Year 1.

NFR-05 Data Privacy & Compliance [High] Mandatory

The system must comply with applicable data privacy regulations and platform policies:

- GDPR (EU): Data subject access requests (DSAR) fulfilled within 30 days via self-service portal.
- GDPR: Right to erasure (deletion) — user data purged within 14 days of deletion request.
- GDPR: Data portability — export of all user/org data in machine-readable format (JSON/CSV).
- EU merchant data stored only in EU-region AWS infrastructure (eu-west-1 or eu-central-1).
- CCPA (California): Privacy disclosures, opt-out of sale of personal data.
- Platform API policies: Rate limiting, usage within permitted scopes, no reselling of platform data.
- SOC 2 Type II certification: audit initiated within 6 months of launch.
- Data retention: configurable per tenant; default 24 months; immutable audit log retained indefinitely.
- Data Processing Agreements (DPA) in place with all sub-processors (AWS, Stripe, Auth0).

NFR-06 Observability & Monitoring [High] Mandatory

The system must provide comprehensive observability for operations and debugging:

- Structured JSON logs from all services; log level configurable per service without restart.
- Distributed tracing via OpenTelemetry with Jaeger or AWS X-Ray backend.
- Metrics exported to Prometheus; Grafana dashboards for all key system metrics.
- Key metrics: sync job duration, attribution match rate, profit computation latency, API error rate, queue depth, rule execution count, action success/failure rate.
- Alerting via PagerDuty: P1 (system down) and P2 (significant degradation) alerts with on-call rotation.
- P1 alert SLA: acknowledged within 5 minutes, mitigated within 60 minutes.
- Customer-facing status page (statuspage.io or equivalent) updated during incidents.
- Error tracking via Sentry with source maps for production frontend debugging.

NFR-07 Maintainability & Code Quality [Medium] Mandatory

The codebase must meet quality standards to support sustainable development velocity:

- Test coverage: $\geq 80\%$ unit test coverage on profit engine, attribution engine, and rules engine.
- Integration test suite for all connector flows using platform sandbox/test accounts.
- E2E test suite (Playwright) for critical user journeys: onboarding, rule creation, action execution.
- Code review required for all PRs to protected branches (minimum 1 approver).
- CI/CD pipeline: lint, test, build, security scan on every PR.
- Database migrations managed with Alembic (Python) / Flyway (schema versioning).
- API versioning: all breaking changes require a new API version.
- Technical debt tracked; $< 20\%$ of each sprint allocated to debt reduction.

NFR-08 Idempotency & Data Integrity [High] Mandatory

All data processing must be idempotent to prevent duplication from retries or replayed events:

- Sync jobs identified by a composite key (org_id, platform, window_start, window_end); duplicate job is a no-op.
- Order ingestion: upsert by order_id + store_id — re-ingesting same order updates, does not duplicate.
- Profit metric computation: overwrite existing ProfitMetric for same (adset_id, window_start, window_end).
- Automation actions: idempotency key stored per action execution — same rule+entity+window cannot trigger twice in 24 hours.
- Webhook delivery: PFAM includes an idempotency key in all outbound webhook payloads.
- All DB writes use transactions; partial failures roll back completely.

7. System Architecture & Design

7.1 Architecture Overview

PFAM follows a microservices architecture deployed on Kubernetes, organized into five logical service groups. Each group is independently deployable, horizontally scalable, and communicates via REST (synchronous) or Kafka events (asynchronous).

Service Group	Technology	Responsibility
Frontend (Web & Mobile)	React 18, Next.js, TypeScript, TailwindCSS	User dashboard, onboarding, rule builder, reports, admin console
API Gateway	Node.js / Express, TypeScript	Authentication, rate limiting, tenant routing, request validation, REST API
Connector Services	Python 3.11, FastAPI, Celery	Shopify / Meta / Google / TikTok OAuth, data ingestion, incremental sync
Data Processing Workers	Python 3.11, Pandas, SQLAlchemy, Celery	ETL transformations, attribution engine, profit calculation engine
Rules & Automation Service	Python 3.11, FastAPI	Rule evaluation, action queuing, Ads API action execution, audit logging
Notification Service	Node.js, Bull queue	Email (SendGrid), Slack, Webhook delivery with retry
Billing Service	Node.js, Stripe SDK	Subscription management, webhook handling, usage metering
Primary Database	PostgreSQL 15 (AWS RDS)	All transactional and analytical data, multi-tenant partitioned
Cache & Session Store	Redis 7 (AWS ElastiCache)	JWT refresh tokens, rate limit counters, dashboard caching
Message Broker	Apache Kafka (AWS MSK)	Sync job events, action queue, inter-service events
Object Storage	AWS S3	CSV/PDF exports, data snapshots, ML model artifacts
ML Platform	Python, scikit-learn, MLflow	Attribution model training, versioning, serving via REST

Service Group	Technology	Responsibility
Monitoring	Prometheus, Grafana, Sentry, DataDog APM	Metrics, alerting, error tracking, distributed tracing
CI/CD	GitHub Actions, Docker, Kubernetes (EKS)	Build, test, security scan, deploy pipeline

7.2 Data Flow Architecture

The following describes the end-to-end data flow from raw platform data to actionable profit insights:

Step	Description
Step 1 — OAuth Connection	Merchant completes OAuth for Shopify + Ad platform. Access tokens encrypted via AES-256 and stored in PostgreSQL token vault. Refresh token rotation scheduled.
Step 2 — Initial Data Import	Connector Service receives import job from API. Calls Shopify Orders API and Ad Platform Insights API in paginated batches. Raw data written to staging tables in PostgreSQL with job_id tracking.
Step 3 — ETL Normalization	Data Processing Worker reads staging data, applies currency conversion, deduplication, and schema normalization. Produces normalized Order, LineItem, AdSpend, Campaign records.
Step 4 — Attribution Run	Attribution Engine processes each order. Attempts Tier 1-5 matching. Writes AttributedOrder records with attribution_tier, confidence_score, attributed_campaign_id.
Step 5 — Profit Calculation	Profit Engine aggregates attributed orders per (adset_id, window). Joins with COGS table, applies return rate, computes net profit. Upserts ProfitMetric records.
Step 6 — Rule Evaluation	Rules Service reads all active rules for the organization. Evaluates conditions against latest ProfitMetric snapshots. Rules passing threshold + guardrails emit action events to Kafka.
Step 7 — Action Execution	Action Worker consumes events from Kafka. Calls Meta / Google API to execute action (pause, budget change). Records API response to audit log. Triggers notification events.
Step 8 — Notification Delivery	Notification Service consumes notification events. Renders email/Slack/webhook payload. Delivers with retry logic. Records delivery status.
Step 9 — Dashboard Serve	API Gateway serves dashboard queries. Reads from ProfitMetric, Campaign, and AuditLog tables. Results cached in Redis for 5 minutes. Frontend renders charts and tables.

7.3 Multi-Tenancy Model

PFAM uses a shared database, shared schema multi-tenancy model with row-level tenant isolation. Every table contains an `org_id` column. All application-layer queries enforce: `WHERE org_id = :current_org_id`. PostgreSQL Row-Level Security (RLS) policies provide a second defense-in-depth layer ensuring no cross-tenant data leakage even if application logic has a bug.

For performance isolation, high-volume organizations (>100K orders) are migrated to dedicated database clusters via a tenant tier classification system. Connection pooling is managed via PgBouncer per service.

7.4 ML Attribution Model

The Tier 5 ML attribution model is a gradient boosting classifier (XGBoost) trained on historical data where Tier 1/2 direct matches are used as ground truth labels. Input features include:

- Order creation timestamp (hour of day, day of week, days since last order).
- Product category and average category ROAS.
- Customer acquisition status (new vs returning).
- Order value relative to campaign AOV.
- Geographic proximity to campaign targeting regions.
- Recency of most recent ad interaction (from pixel events where available).

Model artifacts are versioned in MLflow and stored in S3. A/B testing infrastructure allows parallel evaluation of new model versions before promotion to production. The model is retrained weekly using the accumulated matched order dataset.

8. Data Model & Schemas

8.1 Entity Relationship Overview

The following table describes all primary entities in the PFAM data model with their key fields and relationships.

Entity	Description	Key Fields
Organization	Core tenant entity	id (UUID PK), name, billing_plan, stripe_customer_id, base_currency, data_region, created_at, updated_at
User	Individual user account	id (UUID PK), org_id (FK), email, name, role (enum), mfa_enabled, last_login_at, created_at
Store	Connected Shopify store	id (UUID PK), org_id (FK), shopify_store_id, access_token_enc, access_token_iv, region, plan, last_sync_at, sync_status
AdAccount	Connected ad platform account	id (UUID PK), org_id (FK), platform (enum: meta/google/tiktok), account_id, account_name, access_token_enc, currency, last_sync_at
Campaign	Ad platform campaign entity	id (UUID PK), ad_account_id (FK), platform_campaign_id, name, status, objective, daily_budget, lifetime_budget, start_date, end_date
AdSet	Ad set / ad group entity	id (UUID PK), campaign_id (FK), platform_adset_id, name, status, daily_budget, targeting_summary_json
Ad	Individual ad creative entity	id (UUID PK), adset_id (FK), platform_ad_id, name, status, creative_thumbnail_url
AdInsight	Daily spend + impression metrics	id (UUID PK), adset_id (FK), date, spend, impressions, clicks, reach, conversions, conversion_value, cpm, cpc, ctr
Order	Shopify order record	id (UUID PK), store_id (FK), shopify_order_id, created_at, total_amount, total_discounts, currency, customer_id, financial_status, fulfillment_status
LineItem	Order line item with COGS	id (UUID PK), order_id (FK), product_id, variant_id, sku, quantity, unit_price, unit_cogs, unit_cogs_source (enum: shopify/csv/manual/estimated)
Return	Shopify refund record	id (UUID PK), order_id (FK), line_item_id (FK nullable), refund_amount, quantity_returned, reason, reason_category (enum: defective/wrong/change_mind/sizing/other), created_at

Entity	Description	Key Fields
SkuReturnRate	Trailing return rate per SKU	id (UUID PK), org_id (FK), sku, trailing_90d_rate, trailing_180d_rate, manual_override_rate, last_computed_at
AttributedOrder	Attribution result linking order to campaign	id (UUID PK), order_id (FK), adset_id (FK), attribution_tier (1-5), confidence_score (0.0-1.0), attribution_method, matched_click_id, attributed_revenue, window_start, window_end
ProfitMetric	Computed profit snapshot per ad set per window	id (UUID PK), adset_id (FK), window_type (enum: daily/7d/14d/30d), window_start, window_end, spend, attributed_revenue, attributed_cogs, estimated_returns, platform_fees, net_profit, net_profit_pct, true_roas, order_count, attribution_coverage_pct, computed_at
AutomationRule	User-defined profit rule	id (UUID PK), org_id (FK), name, platform, scope (campaign/adset/ad), conditions_json, action_type, action_params_json, guardrails_json, is_active, created_by (FK), created_at, updated_at
RuleExecution	Record of each rule evaluation	id (UUID PK), rule_id (FK), evaluated_at, entities_evaluated, entities_triggered, entities_blocked_by_guardrail, action_queued (bool)
AuditLog	Immutable action audit trail	id (UUID PK), org_id (FK), actor_user_id (FK nullable), actor_type (enum: user/system), action_type, entity_type, entity_id, platform, rule_id (FK nullable), metric_snapshot_json, api_request_json, api_response_code, api_response_json, created_at
Notification	Notification delivery record	id (UUID PK), org_id (FK), trigger_type, channel (enum: email/slack/webhook), payload_json, delivered_at, delivery_status, retry_count
CogsSetting	COGS configuration per org	id (UUID PK), org_id (FK), scope (enum: sku/category/global), scope_value, cogs_type (enum: absolute/percentage), cogs_value, currency, source (enum: shopify/csv/manual), created_at, updated_at

8.2 Critical Indexes

The following composite indexes are required for query performance at scale:

Table	Index Columns	Purpose
profit_metric	(adset_id, window_type, window_start, window_end)	Fast lookup for rule evaluation and dashboard queries
attributed_order	(adset_id, window_start, window_end)	Attribution aggregation per window

Table	Index Columns	Purpose
attributed_order	(order_id, adset_id)	Duplicate attribution detection
ad_insight	(adset_id, date)	Time-series spend queries
order	(store_id, created_at)	Incremental sync and date-range queries
audit_log	(org_id, created_at)	Audit log search performance
automation_rule	(org_id, is_active, platform)	Rule evaluation pre-filter

9. API Specifications

9.1 API Design Principles

- RESTful resource-oriented design following REST constraints.
- JSON request/response bodies with Content-Type: application/json.
- Authentication via Bearer token in Authorization header.
- Consistent error format: { "error": { "code": "ERR_CODE", "message": "...", "request_id": "..." } }
- Pagination via cursor-based approach: { "data": [...], "next_cursor": "...", "has_more": true }.
- All timestamps in ISO 8601 UTC format (2026-02-11T14:30:00Z).
- Monetary values in cents (integer) with explicit currency field.

9.2 Core API Endpoints

Method	Endpoint	Description	Auth Required	Role
POST	/api/v1/auth/register	Register new user + organization	No	-
POST	/api/v1/auth/login	Email/password login, returns JWT	No	-
POST	/api/v1/auth/refresh	Refresh access token	Refresh Token	-
POST	/api/v1/connect/shopify	Initiate Shopify OAuth flow	Yes	Admin
POST	/api/v1/connect/meta	Initiate Meta Ads OAuth flow	Yes	Admin
POST	/api/v1/connect/google	Initiate Google Ads OAuth flow	Yes	Admin
GET	/api/v1/connectors	List all connected platform accounts	Yes	Analyst+
POST	/api/v1/connectors/{id}/sync	Trigger manual sync for connector	Yes	Admin
GET	/api/v1/dashboard/overview	Account-level profit summary	Yes	Read-Only+
GET	/api/v1/campaigns	List campaigns with profit metrics	Yes	Read-Only+

Method	Endpoint	Description	Auth Required	Role
GET	/api/v1/campaigns/{id}	Campaign detail with ad sets and metrics	Yes	Read-Only+
GET	/api/v1/campaigns/{id}/attributed-orders	Orders attributed to a campaign	Yes	Analyst+
GET	/api/v1/adsets/{id}/profit-metrics	Profit metrics for ad set across windows	Yes	Read-Only+
GET	/api/v1/products/profitability	Product and SKU level profit	Yes	Read-Only+
GET	/api/v1/rules	List all automation rules for org	Yes	Analyst+
POST	/api/v1/rules	Create new automation rule	Yes	Analyst+
PUT	/api/v1/rules/{id}	Update automation rule	Yes	Analyst+
DELETE	/api/v1/rules/{id}	Delete automation rule	Yes	Admin
POST	/api/v1/rules/{id}/toggle	Activate or deactivate rule	Yes	Analyst+
POST	/api/v1/rules/{id}/backtest	Backtest rule against historical data	Yes	Analyst+
GET	/api/v1/audit-log	Paginated audit log for org	Yes	Read-Only+
POST	/api/v1/actions/{id}/reverse	Reverse (undo) an automated action	Yes	Analyst+
GET	/api/v1/cogs	List COGS settings for org	Yes	Analyst+
POST	/api/v1/cogs	Create or update COGS setting	Yes	Analyst+
POST	/api/v1/cogs/import	Bulk import COGS from CSV	Yes	Admin
GET	/api/v1/reports/export	Export profit report (CSV or PDF)	Yes	Analyst+

Method	Endpoint	Description	Auth Required	Role
GET	/api/v1/billing/subscription	Get current subscription details	Yes	Admin
POST	/api/v1/billing/portal	Create Stripe Customer Portal session	Yes	Owner
GET	/api/v1/notifications/history	Notification delivery history	Yes	Read-Only+
POST	/api/v1/notifications/settings	Update notification channel settings	Yes	Admin

10. Security, Privacy & Compliance

10.1 Authentication & Session Security

Control	Implementation
Primary Auth	JWT access tokens (15-min expiry) + HttpOnly Secure refresh tokens (30-day expiry, rotated on use)
Social Auth	Auth0 for Google / Microsoft OAuth with PKCE flow
Platform OAuth	Server-side OAuth for Shopify / Meta / Google. Tokens stored encrypted, never sent to frontend
SSO (Enterprise)	SAML 2.0 via Auth0 SAML connection, supporting Okta, Azure AD, Google Workspace
MFA	TOTP via Authenticator apps. Mandatory for Owner and Admin roles. Backup codes provided at setup
Session Revocation	Redis-based token blacklist for immediate revocation on logout, role change, or security events
Brute Force Protection	Auth endpoint rate-limited to 5 attempts/10 minutes per IP. Progressive delay on failed attempts
Concurrent Sessions	Tracked per user. Admin UI shows active sessions with device/IP info. "Sign out all" available

10.2 Data Protection

Data Type	At Rest	In Transit
OAuth tokens (Shopify, Meta, Google)	AES-256 encrypted, IV stored separately, key in AWS KMS	TLS 1.3 only, HSTS enforced
PII (email, customer data from orders)	AES-256 encrypted in DB columns	TLS 1.3
Financial data (order values, COGS)	Stored in encrypted PostgreSQL with RLS	TLS 1.3
Audit logs	Append-only, stored encrypted, S3 archive for > 2 years	TLS 1.3

Data Type	At Rest	In Transit
ML model training data	Anonymized / pseudonymized, stored in encrypted S3	TLS 1.3
Export files (CSV, PDF)	Signed S3 URLs, 15-minute expiry, no direct public access	TLS 1.3

10.3 Platform API Compliance

PFAM's automated actions must comply with the terms of service of Meta Ads, Google Ads, and TikTok Ads. Key compliance controls:

- Explicit merchant authorization: Users must explicitly enable automation features and accept a disclosure that authorized PFAM to modify their ads.
- Action whitelisting: Only permissible actions (pause entity, adjust budget by %) are available. No campaign creation, ad deletion, or audience modification.
- Rate limiting: Maximum 10 automated actions per ad account per 24-hour period, configurable lower by user.
- Transparency: Every automated action includes a reason parameter in the API call where supported by the platform.
- Audit trail: All actions logged with merchant consent record timestamp.
- Platform partnership: PFAM will apply for Meta Marketing Partner and Google Partner status to gain compliance guidance and preferential API access.

10.4 GDPR & Privacy Compliance Matrix

GDPR Requirement	PFAM Implementation
Lawful basis for processing	Contractual necessity (processing merchant data to deliver the service)
Data subject access request (DSAR)	Self-service portal for export of all personal data within 30 days
Right to erasure	Delete user account and PII within 14 days; anonymize instead of delete for audit log integrity
Data portability	Export all org data as JSON and CSV via in-app or Admin Console
Data minimization	Only customer IDs and email hashed for analytics; full PII only in transactional records
Data residency	EU merchants: data stored in AWS eu-west-1; US merchants: us-east-1

GDPR Requirement	PFAM Implementation
Sub-processor agreements	DPA's in place with AWS, Stripe, Auth0, SendGrid, DataDog
Breach notification	Notify affected users and DPA within 72 hours of confirmed breach; incident playbook maintained
Privacy by design	Data minimization in design; privacy impact assessment for all new features handling PII

11. Integration & Third-party Dependencies

Integration	Type	Used For	API Version	Failure Behavior
Shopify	Commerce Platform	Orders, products, COGS, refunds, webhooks	REST Admin API v2024-01	Cache last known data; alert merchant; retry with backoff
Meta Marketing API	Ad Platform	Campaign data, insights, pause/budget actions	v18.0+	Fallback to cached; disable automation; alert admin
Google Ads API	Ad Platform	Campaign data, spend, conversions, actions	v16+	Fallback to cached; disable automation; alert admin
TikTok Ads API (Phase 2)	Ad Platform	Campaign data, spend, conversions, actions	v1.3+	Graceful degradation; feature flagged off
Stripe	Payment / Billing	Subscription management, invoicing, usage metering	Stripe API v2023-10-16	Queue billing updates; alert on payment failure
Auth0	Identity	OAuth social login, SAML SSO, user management	Auth0 v2	Fall back to local session; alert on Auth0 degradation
AWS S3	Object Storage	Export file storage, ML artifacts, DB snapshots	AWS SDK v3	Queue exports; retry; alert if > 1 hour unavailable
SendGrid	Email	Transactional emails, notification delivery	SendGrid API v3	Retry with exponential backoff; fallback to SMTP if configured
Slack API	Messaging	User workspace notifications via incoming webhook	Slack Web API	Retry 3x; notify via email if Slack fails

Integration	Type	Used For	API Version	Failure Behavior
AWS MSK (Kafka)	Message Broker	Sync job events, action queue, inter-service messaging	Kafka 3.x	Consumer lag monitoring; alert on queue depth > threshold
DataDog	APM & Monitoring	Distributed tracing, infrastructure metrics	DataDog Agent v7	Non-blocking; local metric buffer on DataDog outage
PagerDuty	Incident Management	On-call alerting for P1/P2 incidents	PagerDuty API v2	Fallback to email alerts if PagerDuty unavailable
Open Exchange Rates / Fixer.io	FX Rates	Daily currency conversion for multi-currency stores	REST API	Cache last known rates; retry; alert if rates > 2 days stale

11.1 Dependency Risk Management

All third-party dependencies are subject to the following risk management practices:

- API versioning contract: PFAM pins to specific API versions. Version upgrade is a planned engineering task, not automatic.
- Deprecation monitoring: An automated job checks each platform's developer changelog weekly and flags upcoming deprecations in the engineering team's Slack channel.
- Fallback behavior: Every external API call has a defined failure behavior (see table above). No integration failure shall cause a complete product outage.
- Circuit breakers: Implemented via Hystrix (Java) or Tenacity (Python) pattern. After 5 consecutive failures in 60 seconds, the circuit opens and requests fail fast for 30 seconds before testing again.
- Vendor lock-in mitigation: Abstraction layers (repository pattern) used for database, storage, and messaging to allow swapping of underlying technology.

12. Deployment, Scalability & Monitoring

12.1 Infrastructure Architecture

Component	Technology	Configuration	Scaling Strategy
Container Orchestration	Kubernetes (AWS EKS)	Multi-AZ, 3 worker nodes minimum	Horizontal Pod Autoscaling on CPU/memory/custom metrics
API Service	Node.js / Express (Docker)	2 vCPU, 4GB RAM base pod	HPA: scale to 20 pods at >70% CPU
Worker Service	Python Celery (Docker)	4 vCPU, 8GB RAM base pod (data intensive)	HPA + Kafka consumer group scaling
Primary Database	AWS RDS PostgreSQL 15	db.r6g.large (prod), Multi-AZ, 1TB gp3 SSD	Vertical scaling + read replicas for analytics
Cache	AWS ElastiCache Redis 7	cache.r6g.large, Multi-AZ with replication	Cluster mode enabled for horizontal sharding
Message Broker	AWS MSK (Kafka)	3 broker nodes, kafka.m5.large	Topic partitioning by org_id for throughput
Object Storage	AWS S3	Standard class, versioning enabled	Infinite scalability; lifecycle rules for cost mgmt
CDN	AWS CloudFront	Distribution in front of frontend + API	Edge caching of static assets; WAF integration
ML Serving	SageMaker Endpoint / FastAPI	ml.m5.xlarge, auto-scaling	SageMaker scaling 1-10 instances by request volume

12.2 CI/CD Pipeline

All services follow the same CI/CD pipeline managed via GitHub Actions:

- 46. PR Created: Lint, unit tests, TypeScript type check, Python type check (mypy).
- 47. Security Scan: Snyk dependency vulnerability scan; Trivy container image scan.
- 48. Build: Docker image built and pushed to Amazon ECR.
- 49. Integration Tests: Run against ephemeral test environment using Shopify sandbox and Meta test accounts.
- 50. Staging Deploy: Automatic deploy to staging environment on merge to main branch.
- 51. Smoke Tests: Playwright E2E smoke tests run against staging.

- 52. Production Deploy: Manual approval gate for production. Rolling deploy with <5% error rate threshold (auto-rollback on breach).
- 53. Post-Deploy: Synthetic monitoring runs for 10 minutes. PagerDuty alert on anomaly.

12.3 Environment Strategy

Environment	Purpose	Data	Access
Development	Local developer machines	Synthetic test data	Engineering team only
Staging	Pre-production integration testing	Anonymized copy of production data, refreshed weekly	Engineering + QA + Product
Production	Live customer environment	Real merchant data	Restricted; admin access via bastion host with audit log
Sandbox	Customer testing/evaluation environment	Isolated, no real API keys	Prospects + internal testing

12.4 Key Monitoring Metrics

Metric Category	Metric	Alert Threshold
Availability	API error rate (5xx)	> 1% over 5 minutes → P1
Availability	API error rate (4xx)	> 10% over 5 minutes → P2
Performance	API p95 response time	> 2 seconds → P2
Data Pipeline	Sync job duration	> 60 minutes for standard org → P2
Data Pipeline	Kafka consumer lag	> 10,000 messages → P2
Data Pipeline	Attribution match rate	< 40% over 24 hours → alert to product team
Business	Rule execution errors	> 5% failure rate → P1
Business	Automation action failure	Any failure after 3 retries → P2 + merchant notification
Security	Auth brute force attempts	> 100 failed logins/IP/hour → P1 security alert

Metric Category	Metric	Alert Threshold
Infrastructure	DB connection pool utilization	> 80% → P2
Infrastructure	Disk utilization	> 80% → P2

13. Testing & Quality Assurance

13.1 Test Strategy Overview

Test Type	Scope	Tooling	Coverage Target	Frequency
Unit Tests	Profit engine, attribution logic, rule evaluation, data transformations	pytest (Python), Jest (Node.js)	>= 80% line coverage	Every commit via CI
Integration Tests	Connector flows, API endpoint behavior, DB interactions	pytest + testcontainers, Supertest	All critical paths	Every PR merge
E2E Tests	Full user journeys: onboarding, rule creation, action execution	Playwright	8 critical flows	Every merge to main + nightly
Load & Performance Tests	API throughput, sync job scalability, DB query performance	k6 / Locust	Per NFR targets	Weekly + pre-major release
Security Tests	OWASP Top 10, auth bypass, injection, data isolation	OWASP ZAP, Snyk, manual pen test	All OWASP Top 10 vectors	Automated in CI; manual quarterly
Attribution Accuracy Tests	Verify attribution accuracy against known-good dataset	Custom Python test harness	>= 85% overall, >= 92% Tier 1/2	Weekly regression on model retrain
Chaos Engineering	Service failure, DB failover, Kafka partition loss	AWS Fault Injection Simulator	Key failure scenarios	Monthly in staging
Smoke Tests	Verify core functionality after each deploy	Playwright subset	Top 5 user journeys	After every production deploy

13.2 Critical Test Scenarios

The following test scenarios are mandatory and must pass before any production release:

ID	Scenario	Pass Criteria
TS-01	Profit Accuracy Validation	Given: 1,000 orders with known revenue, COGS, return rates. When: Profit engine runs. Then: Net profit output matches manual calculation within $\pm 1\%$.
TS-02	Attribution Tier 1 Accuracy	Given: 500 orders with fbclid or gclid present. When: Attribution engine runs. Then: $\geq 92\%$ of orders correctly attributed to the campaign that generated the click.
TS-03	Automation Rule — Pause Execution	Given: Ad set with 7-day net profit = $-\$500$, rule condition = profit < 0 , guardrail = 15 orders (met). When: Rule evaluation runs. Then: Meta API pause call is made, audit log entry created, Slack notification sent.
TS-04	Automation Rule — Guardrail Block	Given: Ad set with 7-day net profit = $-\$500$, rule condition = profit < 0 , guardrail = 15 orders (only 10 matched). When: Rule evaluation runs. Then: No API call made, RuleExecution log shows "blocked_by_guardrail: true."
TS-05	Multi-tenant Data Isolation	Given: Two organizations with separate data. When: Org A's API token is used to query /api/v1/campaigns. Then: Only Org A's campaigns are returned. No Org B data is accessible under any query permutation.
TS-06	Idempotent Sync	Given: Sync job for the same (org_id, window) runs twice. When: Second run completes. Then: No duplicate Order, AdInsight, or ProfitMetric records exist. Counts are identical to after first run.
TS-07	Returns Retroactive Recalculation	Given: Order attributed to Campaign X; refund issued 15 days later. When: Refund sync runs. Then: ProfitMetric for Campaign X for the original window is updated to reflect the actual return cost.
TS-08	RBAC Enforcement	Given: Read-Only user token. When: POST /api/v1/rules is called. Then: API returns 403 Forbidden with error code "INSUFFICIENT_PERMISSIONS".
TS-09	Platform API Failure Graceful Degradation	Given: Meta Ads API returns 503. When: Sync job for Meta connector runs. Then: Job retries 3x with backoff, marks connector as degraded, displays stale data warning in UI, sends admin alert. No unhandled exception thrown.
TS-10	COGS Recalculation on Update	Given: COGS for SKU X changed from \$5 to \$8. When: Update is saved. Then: All historical ProfitMetric records containing SKU X attributed orders are recalculated. Net profit values updated within 30 minutes.

14. Acceptance Criteria

The following acceptance criteria define the minimum conditions required for a production release of PFAM. All criteria must be met before GA launch.

ID	Criterion Name	Priority	Definition of Done
AC-01	Platform Connectors	High	Users can connect a Shopify store and a Meta Ads account via OAuth. After connection, historical data (orders, ad spend) for the last 90 days is visible in the dashboard within 24 hours. Sync completes without manual intervention. Connector health page shows successful sync with order count and spend totals matching platform-reported values within 2%.
AC-02	Profit Calculation Accuracy	High	Profit engine computes net profit per campaign within 1% tolerance of a manually-verified ground truth dataset of 500 orders. COGS sourced from Shopify, return rates from historical refunds. Finance Manager stakeholder can reconcile PFAM profit figures to their monthly P&L within an acceptable variance.
AC-03	Automation Rule Execution	High	A rule configured to pause an ad set when 7-day rolling net profit < \$0 (with minimum 15 orders guardrail) correctly triggers a Meta Ads API pause call when conditions are met. Audit log records the event with timestamp, rule ID, and metric snapshot. Slack notification delivered within 5 minutes of action.
AC-04	Attribution Coverage	High	For an organization with a well-configured Shopify + Meta setup (fbclid present on >= 50% of orders), the overall attribution coverage rate (% of orders attributed to at least one campaign) is >= 75%. Orders with attribution confidence < 0.7 are clearly flagged in the UI.
AC-05	Security & Token Protection	High	Security audit confirms: no OAuth tokens or API keys stored in plaintext in DB, logs, or environment variables. AES-256 encryption verified for all token columns. WAF active and blocking OWASP Top 10 test attacks. MFA enforced for Admin users.
AC-06	Multi-tenancy Data Isolation	High	Penetration test and automated test confirms zero cross-tenant data leakage under all tested query scenarios. RLS policies active. Separate organizations cannot access each other's campaigns, orders, metrics, or rules.
AC-07	Performance Under Load	Medium	Load test with 500 concurrent users demonstrates: API p95 response < 500ms, dashboard page load < 2.5 seconds, sync job for a 30K-order organization completes in < 10 minutes. No P1 errors observed during load test.

ID	Criterion Name	Priority	Definition of Done
AC-08	Notification Delivery	Medium	Rule trigger notifications delivered via Email and Slack within 5 minutes of action execution. Webhook delivery confirmed with correct payload and valid HMAC signature. Notification history page shows correct delivery status.
AC-09	RBAC Enforcement	High	All permission-restricted API endpoints return 403 Forbidden when called with a token of insufficient role. Read-Only users cannot create rules, trigger syncs, or export data via UI or API.
AC-10	Billing Integration	Medium	User can subscribe to a paid plan via Stripe Checkout. Plan upgrade/downgrade works correctly via Stripe Customer Portal. Failed payment triggers grace period (7 days) notification. Subscription data reflected accurately in-app.
AC-11	Onboarding Experience	High	A new user can complete full onboarding (register, connect Shopify, connect Meta, view first dashboard) within 15 minutes without support intervention. Onboarding checklist tracks completion. Time to first insight < 24 hours after initial data import.
AC-12	Export Functionality	Medium	Campaign profit CSV export for a 90-day range with 1,000 campaigns generates correctly within 60 seconds. PDF summary report renders with accurate charts and data. Finance Manager can reconcile exported data to PFAM dashboard values.

15. Operations & Maintenance

15.1 Backup & Recovery

Asset	Backup Method	Frequency	Retention	RTO	RPO
PostgreSQL DB	AWS RDS automated snapshots + continuous WAL archiving to S3	Daily snapshot + continuous	35 days snapshots; 7 years audit archive	< 1 hour	< 5 minutes
Redis Cache	RDB snapshot + AOF (append-only file)	Every 60 seconds AOF; daily RDB	7 days	< 15 minutes	< 60 seconds
S3 Objects (exports)	Cross-region S3 replication	Continuous	Per lifecycle policy (90 days hot, 2 years glacier)	< 30 minutes	< 1 hour
Application Config	Git + Terraform state in S3	On change	Indefinite	< 30 minutes	N/A
ML Model Artifacts	S3 versioned storage	Each model training run	1 year of model versions	< 30 minutes	Last model version

15.2 Incident Response

Severity	Definition	Response SLA	Communication
P1 — Critical	Complete outage; data loss risk; security breach; automated actions misfiring at scale	Acknowledge: 5 min; Resolve: 60 min	Immediate PagerDuty page; status page updated; email to all affected customers
P2 — Major	Significant feature degradation; sync delays > 2 hours; automation paused	Acknowledge: 15 min; Resolve: 4 hours	PagerDuty alert; status page updated; email to affected customers
P3 — Minor	Non-critical feature degraded; cosmetic UI issues; minor data discrepancy	Acknowledge: 2 hours; Resolve: 24 hours	Status page note; affected customers notified if data-impacting

Severity	Definition	Response SLA	Communication
P4 — Informational	Known issue, workaround available; performance slightly below target	Acknowledge: 24 hours; Resolve: next sprint	Internal tracking only

15.3 Support Tiers

Plan	Support Tier	Channels	Response SLA
Starter	Standard	Email, help docs, in-app chatbot	48 business hours
Growth	Standard Plus	Email + live chat (9-5 EST, M-F)	24 business hours
Pro	Priority	Email + live chat + phone (on request)	4 business hours
Scale	Priority Plus	All channels + dedicated CSM (quarterly review)	2 business hours
Enterprise	Enterprise	All channels + dedicated CSM (monthly review) + custom SLA	1 hour (custom SLA)

15.4 Release Management

- Release cadence: Major features every 4 weeks (sprint cycle); Hotfixes as needed.
- Feature flags: New features released behind feature flags to enable gradual rollout and instant rollback.
- Release notes published to in-app notifications and changelog page for every release.
- Database migrations: Always forward-compatible; backward-compatible changes where possible; breaking migrations require a multi-phase deploy.
- API deprecation: New API version announced 6 months before old version sunset; sunset email notifications to affected API token holders.

16. Risks & Mitigations

Risk	Name	Impact	Probability	Severity
R-01	Platform API Deprecation or Policy Change	High	High	Critical
R-02	Attribution Inaccuracy Leading to Incorrect Actions	High	Medium	High
R-03	Merchant Data Quality (Inaccurate COGS)	High	High	High
R-04	Competitor Feature Parity	Medium	High	Medium
R-05	Security Breach or Data Leak	High	Low	High
R-06	Platform-Imposed Automation Restrictions	Medium	Low	Medium
R-07	Rapid Customer Churn (>5% Monthly)	High	Medium	High
R-08	Key Talent Loss (Founding Team or Key Engineer)	Medium	Medium	Medium

Detailed mitigation strategies for each risk:

R-01 Platform API Deprecation or Policy Change [Impact: High Probability: High Severity: Critical]	
Description	Meta, Shopify, or Google deprecates a critical API endpoint or revokes third-party automation permissions, breaking connectors or automation features.
Mitigations	<ul style="list-style-type: none">Version-pinned connectors with graceful degradation to read-only mode.Continuous monitoring of each platform's developer changelog for deprecation notices.Marketing Partner status with Meta and Google for early access to API changes.Multi-platform strategy: if one platform breaks, others remain functional.Connector abstraction layer enables rapid replacement of platform-specific code.
R-02 Attribution Inaccuracy Leading to Incorrect Actions [Impact: High Probability: Medium Severity: High]	
Description	Attribution model credits wrong campaigns; system pauses a profitable campaign or keeps an unprofitable one running, causing merchant financial harm.

Mitigations	<ul style="list-style-type: none"> • Conservative guardrails by default: minimum order count before any automated action. • Confidence scores displayed on every attributed metric; low confidence flagged prominently. • Manual override available for all automated actions, with one-click reversal. • Backtest feature: merchants can preview rule behavior on historical data before enabling. • Attribution accuracy tracked weekly and regressed against known-good dataset. • Liability disclaimers in ToS; E&O insurance to mitigate financial exposure.
--------------------	---

R-03 Merchant Data Quality (Inaccurate COGS) [Impact: High | Probability: High | Severity: High]

Description	Merchants provide inaccurate COGS data (common for early-stage brands), leading to misleading profit calculations and bad automation decisions.
Mitigations	<ul style="list-style-type: none"> • COGS confidence indicator shown on all profit metrics. • Validation alerts when COGS appears anomalous (e.g., > 90% of price, or \$0 for physical goods). • Multiple COGS input methods to reduce friction and increase accuracy. • Onboarding wizard explicitly walks users through COGS setup. • Quarterly prompt to review and update COGS data.

R-04 Competitor Feature Parity [Impact: Medium | Probability: High | Severity: Medium]

Description	TripleWhale, Northbeam, or a well-funded new entrant adds automation features, commoditizing PFAM's core differentiator.
Mitigations	<ul style="list-style-type: none"> • Speed advantage: ship automation depth (multi-condition rules, ML recommendations) faster than incumbents. • Data moat: ML models trained on growing merchant dataset become more accurate over time. • Brand building: establish PFAM as the "profit-first" category leader through content and community. • Switching costs: 12+ months of historical profit data and custom workflows make migration painful. • Expand moat into areas incumbents cannot easily follow: agency tools, enterprise, omnichannel (email, influencer).

R-05 Security Breach or Data Leak [Impact: High | Probability: Low | Severity: High]

Description	A security vulnerability leads to unauthorized access to merchant OAuth tokens, order data, or customer PII.
Mitigations	<ul style="list-style-type: none"> • AES-256 encryption for all sensitive data at rest. • Penetration testing quarterly by third-party security firm. • Bug bounty program to incentivize responsible disclosure.

- SOC 2 Type II certification providing independent security controls validation.
- Incident response plan with < 72-hour GDPR breach notification procedure.
- Cyber liability insurance (\$2M policy).

R-06 Platform-Imposed Automation Restrictions [Impact: Medium | Probability: Low | Severity: Medium]

Description	Meta or Google restricts or bans third-party automated actions via their APIs, disabling PFAM's core automation layer.
Mitigations	<ul style="list-style-type: none"> • Action rate limiting and explicit user consent already in place (proactive compliance). • Monitor platform policies for changes; maintain direct relationships with platform policy teams. • If automation restricted: pivot to "one-click recommendation" model (user approves, PFAM executes) — still 10x better than competitors with no automation. • Maintain read-only analytics as independent value layer (product is still useful without automation).

R-07 Rapid Customer Churn (>5% Monthly) [Impact: High | Probability: Medium | Severity: High]

Description	Customers churn faster than expected due to poor onboarding, unclear value, or inadequate attribution accuracy, destroying LTV:CAC economics.
Mitigations	<ul style="list-style-type: none"> • White-glove onboarding for Growth+ plans (15-minute setup call). • Onboarding completion checklist: 85%+ completion target reduces early churn. • Monthly "value delivered" email: quantifies money saved, campaigns optimized. • Health score monitoring: proactive CS outreach when engagement drops. • Annual contract option with 15% discount locks in 12 months.

R-08 Key Talent Loss (Founding Team or Key Engineer) [Impact: Medium | Probability: Medium | Severity: Medium]

Description	Loss of CTO or a critical data engineer would significantly slow product development.
Mitigations	<ul style="list-style-type: none"> • 4-year vesting with 1-year cliff for all founding team members. • Competitive retention equity refresh grants annually. • Documentation and knowledge sharing culture: no single person owns a critical system. • Succession planning: identify and grow internal candidates for key roles. • Advisor network provides continuity on strategic decisions.

17. Appendix

17.1 Technology Stack Summary

Layer	Technology	Rationale
Frontend	React 18 + Next.js + TypeScript + TailwindCSS	SSR for SEO, type safety, rapid UI development, large talent pool
Charts / Visualization	Recharts + D3.js	Recharts for standard charts; D3 for custom attribution visualizations
API Gateway	Node.js + Express + TypeScript	Performant for I/O-bound requests; same language as frontend reduces context switching
Data Workers	Python 3.11 + FastAPI + Celery	Python best-in-class for ML/data; FastAPI for async; Celery for distributed task execution
ORM / DB Access	SQLAlchemy (Python) + Knex.js (Node)	Mature ORMs with migration support and connection pooling
ML Framework	scikit-learn + XGBoost + MLflow	scikit-learn for preprocessing; XGBoost for attribution model; MLflow for experiment tracking and model registry
Authentication	Auth0 (managed) + JWT	Reduces auth engineering burden; enterprise SSO/SAML out of the box
Secrets Management	HashiCorp Vault / AWS Secrets Manager	Centralized secret rotation and audit; no secrets in environment variables
Database	PostgreSQL 15 + PgBouncer	ACID compliance; JSONB for flexible schemas; RLS for tenant isolation; PgBouncer for connection pooling
Cache	Redis 7	Sorted sets for leaderboards; pub/sub for real-time events; fast TTL-based caching
Message Broker	Apache Kafka (AWS MSK)	Durability, replay capability, consumer group scaling; preferred over RabbitMQ for data-intensive workloads
Email	SendGrid API	High deliverability, template management, analytics
Payments	Stripe	Industry-standard; usage metering; Customer Portal reduces billing engineering

Layer	Technology	Rationale
Infrastructure	AWS (EKS, RDS, ElastiCache, MSK, S3, CloudFront, KMS)	Single-provider simplicity; all required managed services available; GDPR region compliance
IaC	Terraform + Helm	Reproducible infrastructure; GitOps workflow
CI/CD	GitHub Actions + Docker + ECR	Native GitHub integration; no additional CI server to manage
Monitoring	Prometheus + Grafana + Sentry + DataDog APM	Open standards + commercial APM; covers metrics, errors, and distributed tracing

17.2 Pricing Tiers Reference

Plan	Monthly Price	Ad Spend Limit	Ad Accounts	Users	Sync Frequency	Key Features
Starter	\$99/month	Up to \$10K	1	1	Daily	Core dashboards, 3 rules, email alerts
Growth	\$299/month	\$10K - \$50K	3	3	Hourly	All Starter + Slack, CSV export, 10 rules
Pro	\$599/month	\$50K - \$150K	Unlimited	10	Hourly	All Growth + API access, PDF reports, 50 rules, ML attribution
Scale	\$1,299/month	\$150K - \$500K	Unlimited	25	Hourly + real-time	All Pro + Dedicated CSM (quarterly), custom rule templates, SLA
Enterprise	Custom	\$500K+	Unlimited	Unlimited	Real-time	All Scale + SSO, white-label reports, custom SLA, monthly CSM, custom integrations

Plan	Monthly Price	Ad Spend Limit	Ad Accounts	Users	Sync Frequency	Key Features
Agency Add-on	+\$1,500/month	Per client	Up to 10 clients	Per client	Per client plan	Multi-client dashboard, client-level reporting, agency branding

17.3 Document Revision History

Version	Date	Author	Changes
1.0	2026-02-11	Product Team	Initial draft SRS — high-level requirements and architecture overview.
2.0	2026-02-12	Product Team	Enterprise edition: expanded functional requirements with acceptance criteria; detailed data model; full API specification; security/compliance matrix; risk register; testing strategy; operational runbooks.

17.4 Related Documents

- PFAM Investor Q&A Guide — Business case, market analysis, unit economics
- PFAM UI/UX Design Specification — Wireframes, component library, interaction patterns
- PFAM Attribution Engine Technical Spec — Detailed ML model documentation, training methodology
- PFAM Security Policy — Detailed security controls, pen test reports
- PFAM API Reference — Full OpenAPI 3.0 specification with request/response schemas
- PFAM Runbooks — Incident response procedures for all P1/P2 scenarios