Technical Architecture & Algorithm Documentation

Project: AI-Powered Artist Promotion Assistant

Document Type: Technical Specification

Status: Proof of Concept (POC)

1. System Architecture

The system follows a modern **Event-Driven Architecture**, designed to process social media streams in real-time while maintaining a "Human-in-the-Loop" validation layer.

Layers

1. **Ingestion Layer (Simulated):** * Connects to Instagram Graph API and LinkedIn Marketing API.
    - **Function:** Fetches comments, DMs, and posts containing target hashtags (e.g., #digitalart, #lookingforart).
    - **POC Implementation:** fetchNewLead() function in App.jsx randomly selects from a curated pool of edge-case scenarios (MOCK_LEADS_POOL).
2. **Intelligence Layer (AI Engine):**
    - **NLP Processor:** Uses Large Language Models (LLM) to parse text sentiment and intent.
    - **Opportunity Scorer:** Assigns a numerical value to leads.
    - **Response Generator:** Drafts replies based on the brandVoice state configuration.
3. **Presentation Layer (Client):**
    - React.js Single Page Application (SPA).
    - Manages state for leads, history, and analytics.

2. Core Algorithms

A. Opportunity Scoring Algorithm (OSA)

The OSA determines the commercial viability of a lead. It allows the artist to prioritize high-value interactions.

Logic Flow:

1. **Input:** User comment text + User profile metadata.
2. **Keyword Matching:** Scans for high-value triggers:
    - *Purchase Signals:* "buy", "price", "commission", "project", "lobby", "budget".
    - *Engagement Signals:* "love this", "cool", "fire emoji".

    - *Spam Signals:* "nft", "promo", "follow back", "check my bio".
3. **Contextual Weighting:**
    - Inquiries with specific locations ("Manhattan lobby") or dimensions ("large format") receive a **1.5x multiplier**.
    - Inquiries from verified accounts or profiles with business keywords ("Director", "Curator") receive a **1.2x multiplier**.
4. **Output:** Score (0-100).

Pseudo-Code Representation:

```
function calculateOpportunityScore(text, userType) {
  let score = 50; // Base score

  if (contains(text, ["price", "commission", "buy"])) score += 30;
  if (contains(text, ["love", "cool", "nice"])) score += 10;
  if (contains(text, ["promo", "check bio"])) score -= 40;

  if (userType === "Interior Designer" || userType === "Corporate Buyer") {
    score = Math.min(score * 1.2, 100);
  }

  return score;
}
```

B. Persona Classification Logic

The system automatically categorizes users to tailor the AI response.

Persona	Triggers / Detection Logic	Response Strategy
Interior Designer	Keywords: "project", "lobby", "space", "client".	Professional, portfolio-focused.
Corporate Buyer	Keywords: "office", "headquarters", "license", "brand".	Business-casual, licensing-focused.
Art Enthusiast	Keywords: "beautiful", "wow", emojis.	Friendly, gratitude-focused.
Influencer/Spam	Keywords: "collab", "shootout", "ambassador".	Polite decline or ignore.

3. Data Flow & Security

Data Lifecycle

1. **Fetch:** fetchNewLead triggers a pull request.
2. **Process:** Data is sanitized (HTML tags removed) and passed to the Scorer.
3. **Store:** Active leads live in React State (useState); approved leads move to the History array (simulating a PostgreSQL database commit).
4. **Purge:** Dismissed leads are removed from memory immediately.

Security Protocols

- **API Isolation:** In the production roadmap, API keys are stored in .env files and never exposed to the frontend.
- **Read-Only Scope:** The initial API connection requests read-only access to public comments to minimize security risks.
- **GDPR Compliance:** The "Dismiss" action permanently removes the user's data from the local session, respecting the right to be forgotten.

4. Future Roadmap

- **Integration:** Replace simulated data with live axios calls to OpenAI API.
- **Auth:** Implement Firebase Authentication for multi-user support.
- **Mobile:** Port React web app to React Native for on-the-go approvals.