enovate-it.com
We care, we help you all to win

High-Level Design

SENTIFY

Version:V1.0 Date: 6 Oct 2025

(Approval of the HLD indicates an understanding of the purpose and content described in this deliverable. By signing this deliverable, each individual agrees on proposed subsystems roles and business logic it is responsible for; Test Planning and Detailed design work should be initiated on this project and necessary resources should be committed as described in the Charter document.)

Document Revision History

Ver	Description	Author	Approver	Date
V1.0	**Initial Draft**	Yash Lathiya/ Nikhil Rapariya		

1. Introduction

Sentify is an AI-powered market intelligence platform designed specifically for cryptocurrency traders. It cuts through the noise of endless crypto news and chatter by analyzing global news sources and media outlets, delivering only what matters through probability-based insights.

Using reputable crypto and financial news sources as the primary data inputs and Gemini API for AI analysis, Sentify continuously scans crypto-related content, analyzes its credibility and sentiment, and transforms it into clear, actionable trading insights with 0–100% probability scores.

Traders can access the platform as free users or subscribe for premium features including advanced filters and real-time alerts. The system empowers users to trade with data-driven insights rather than emotional reactions.

Key highlights of Sentify include:

- Real-time Market Intelligence – Continuously monitors news sources and online publications for crypto-related news and events, processes them through AI analysis, and delivers instant sentiment scores.
- Probability-based Alerts – Provides 0–100% bullish/bearish probability scores indicating likely market impact direction and intensity.
- Customizable Filters – Users can set preferences for specific cryptocurrencies and sentiment thresholds to receive tailored alerts.
- Push Notification System – Instant delivery of market-moving insights directly to users' devices.
- Historical Analysis – Track sentiment accuracy and performance over time with comprehensive analytics.

Scope:
The scope of Sentify includes the development of a cross-platform mobile application (Android & iOS) with web dashboard support, providing real-time crypto market sentiment analysis and predictive alerts to traders.

In-Scope Features

- User Access & Management
    - Registration via email or Google for personalized experiences
- Data Ingestion & Processing
    - Multi-Source News Scraping: Real-time scraping from *trusted cryptocurrency news portals, blogs, and financial analysis sites.*
    - Content Filtering Pipeline: Natural language preprocessing, duplicate detection, spam filtering, and removal of irrelevant content.
    - Data Normalization: Converting data into a unified structure for AI analysis (e.g., headline, body, timestamp, source reliability).
    - Scheduled Crawling & Continuous Updates: Periodic checks for new articles with dynamic refresh based on market volatility.
- AI Analysis & Scoring
    - Gemini API integration for NLP and sentiment analysis
    - Probability scoring (0-100%) for bullish/bearish market impact
    - Credibility assessment and source reliability scoring
- Alert Management
    - Push notification delivery for significant market events
    - User-customizable thresholds for specific coins and sentiment levels
    - Alert history and performance tracking
- User Preferences & Customization
    - Coin/watchlist management for specific cryptocurrency tracking
    - Sentiment threshold configuration (e.g., only alerts above 70% probability)
    - Notification frequency and timing controls

2. System Architecture

1. High-Level Goal

Sentify is an AI-powered mobile application designed to provide cryptocurrency traders and enthusiasts with timely, actionable intelligence. The system ingests and analyzes a high volume of news, identifies high-impact events through sentiment analysis, and delivers personalized alerts to users based on their interests and sensitivity preferences.

2. System Architecture Overview

The architecture is designed around modern, decoupled modules to ensure scalability, resilience, and maintainability. It consists of three primary, independent services that communicate asynchronously through a central database and a task queue.

- **Data Ingestion & Processing Pipeline:** The data collection and AI analysis engine.
- **The Alerting Engine:** The core value-delivery component that monitors for and dispatches notifications.
- **Backend API:** The gateway for the user-facing mobile application.

Module 1: Data Ingestion & Processing Pipeline

Objective: To continuously discover, process, analyze, and store relevant crypto news from across the web.

Technology: Python, Scrapy, Sentence-Transformers, LangChain, Gemini API.

Process Flow:

- **Source Fetching:** A scheduled task (running every few minutes) initiates the pipeline, fetching new articles from a predefined list of news APIs, blogs, and financial websites.
- **Preprocessing:** Raw HTML is cleaned, and the core article text, title, and metadata are extracted.
- **Vectorization:** The cleaned text is converted into a numerical vector embedding using a `Sentence-Transformers` model.

- **Deduplication:** A similarity search is performed against the **PG Vector** database. If an article with a highly similar vector already exists, it is discarded to avoid redundant analysis and alerts.
- **AI Sentiment Analysis:** For new, unique articles, a structured prompt is sent to the **Gemini API**. The API returns a detailed analysis, including:
- A sentiment score (e.g., from -1.0 to 1.0).
- An impact level (e.g., High, Medium, Low).
- A concise summary of the news.
- **Persistence:** The processed article—including its text, metadata, vector embedding, and the full sentiment analysis result—is saved as a new record in the `news_articles` table in the PostgreSQL database.

Module 2: The Alerting Engine

Objective: To monitor the analyzed news in real-time and deliver personalized, high-impact alerts to the correct users instantly.

Technology: Python, Redis, PostgreSQL.

Process Flow:

- **Event Detection:** A dedicated worker service continuously monitors the `news_articles` table for newly inserted records.
- **Audience Matching:** For each new article, the engine performs a critical matching query to identify the target audience:
- It retrieves the `coin_id` (e.g., 'ethereum') and the `sentiment_score` (e.g., -0.92) from the new article.
- It then queries the `users` and `user_watchlist` tables to find all users who meet **both** of the following criteria:

a. The user is following that specific coin (`user_watchlist.coin_id = 'ethereum'`).

b. The article's sentiment score meets the user's personal **`alert_sensitivity`** threshold (e.g., `abs(-0.92) >= user.alert_sensitivity`).

- **Payload Construction:** For the matched list of users, the engine constructs the push notification payload (title, body, and any data needed to deep-link into the app).
- **Dispatch:** The engine makes a single API call to **Firebase Cloud Messaging (FCM)**, providing the payload and the list of unique device tokens for the matched users.

FCM then handles the reliable delivery of the notification to each user's mobile device.

Module 3: Backend API

Objective: To serve as the secure and efficient interface between the Sentify mobile app and the backend ecosystem.

Technology: NodeJS, PostgreSQL.

Key Responsibilities:

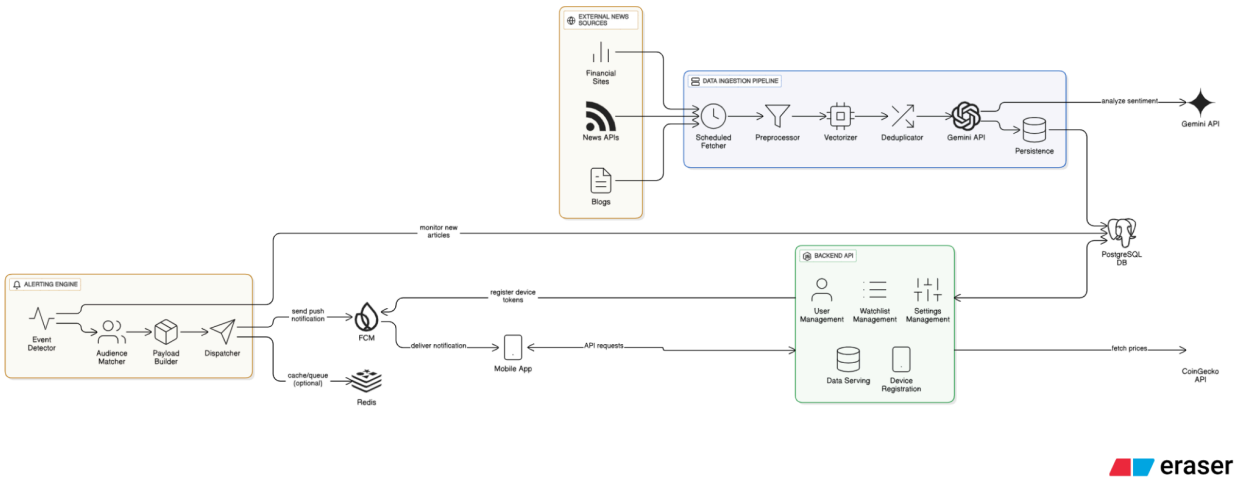
- **User Management:** Handles user registration, login (authentication), and profile management.
- **Watchlist Management:** Provides CRUD (Create, Read, Update, Delete) endpoints for users to manage their list of followed cryptocurrencies.
- **Settings Management:** Allows users to update their personal settings, most importantly their `alert_sensitivity` level.
- **Data Serving:**
- Provides data for the main dashboard, including prices (fetched and cached from the **CoinGecko API**) and a feed of recent high-impact news from the PostgreSQL database.
- Serves historical news and sentiment data for specific coin detail pages.
- **Device Registration:** Securely receives and stores the user's FCM device token required for push notifications.

End-to-End Workflow Example

- A news site publishes an article: "Major Exchange Reports Security Audit Failure."
- The **Ingestion Pipeline** fetches it, creates an embedding, sees it's unique, and sends it to Gemini. Gemini returns a sentiment score of **-0.95**. The data is saved to PostgreSQL.
- The **Alerting Engine** detects the new article.
- The engine queries the database: "Find all users following this exchange's coin whose sensitivity is set to 95% or less."
- It finds two users: User A (sensitivity 80%) and User B (sensitivity 90%). It discards User C (sensitivity 98%).
- The engine sends the device tokens for User A and B to **FCM** with the alert message.

- Users A and B receive a push notification within minutes of the news being published. User C does not.
- Later, all three users open the **Sentify App**. The **Backend API** serves them the dashboard, where they can all see the negative news item in their feed.

System Architecture Diagram:



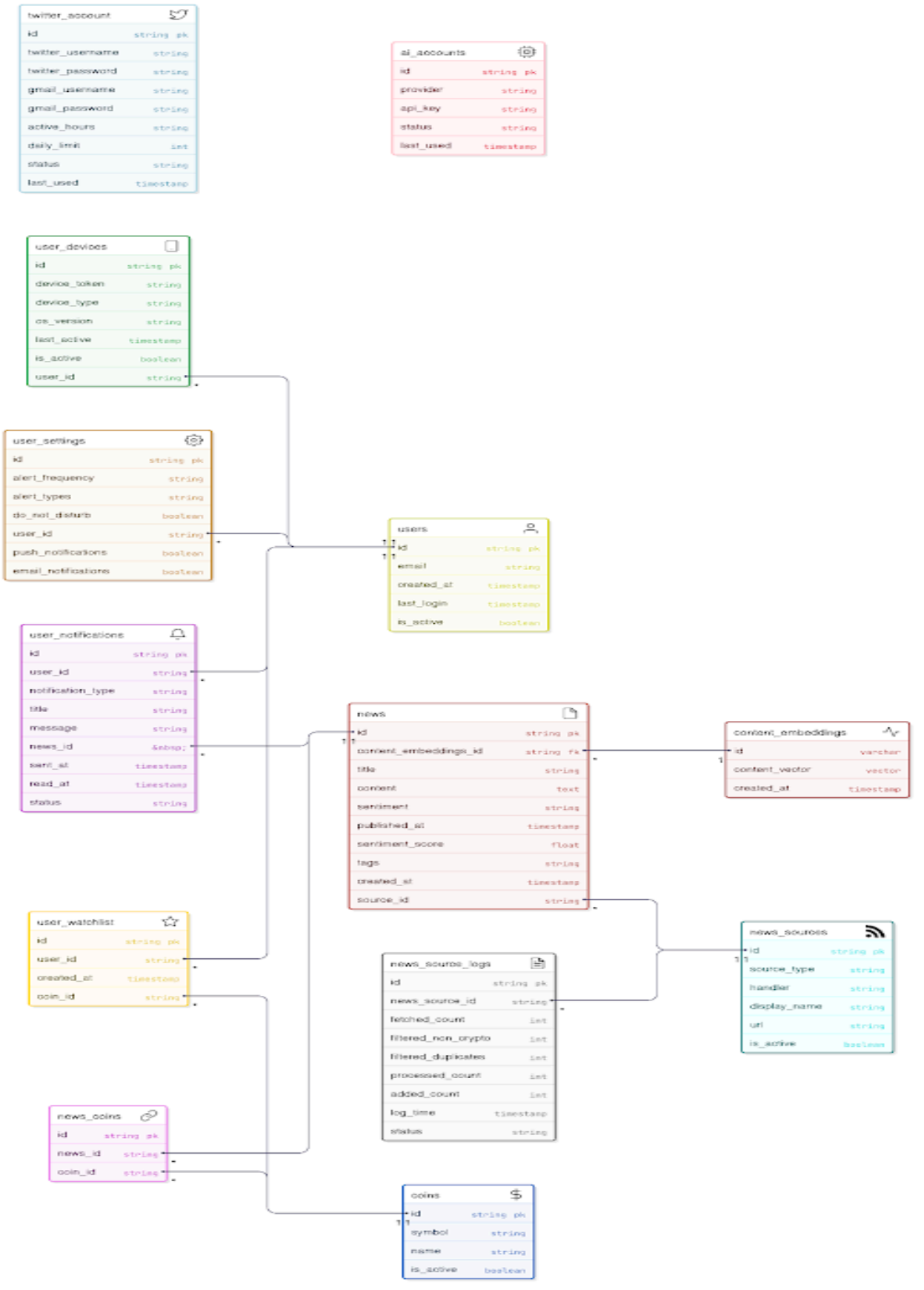
Technology Stack:

- Frontend: Flutter (Mobile), React (Web Admin Dashboard)
- Backend: Node.js (User Services)
- AI/ML: Gemini API, PyTorch (potential custom models)

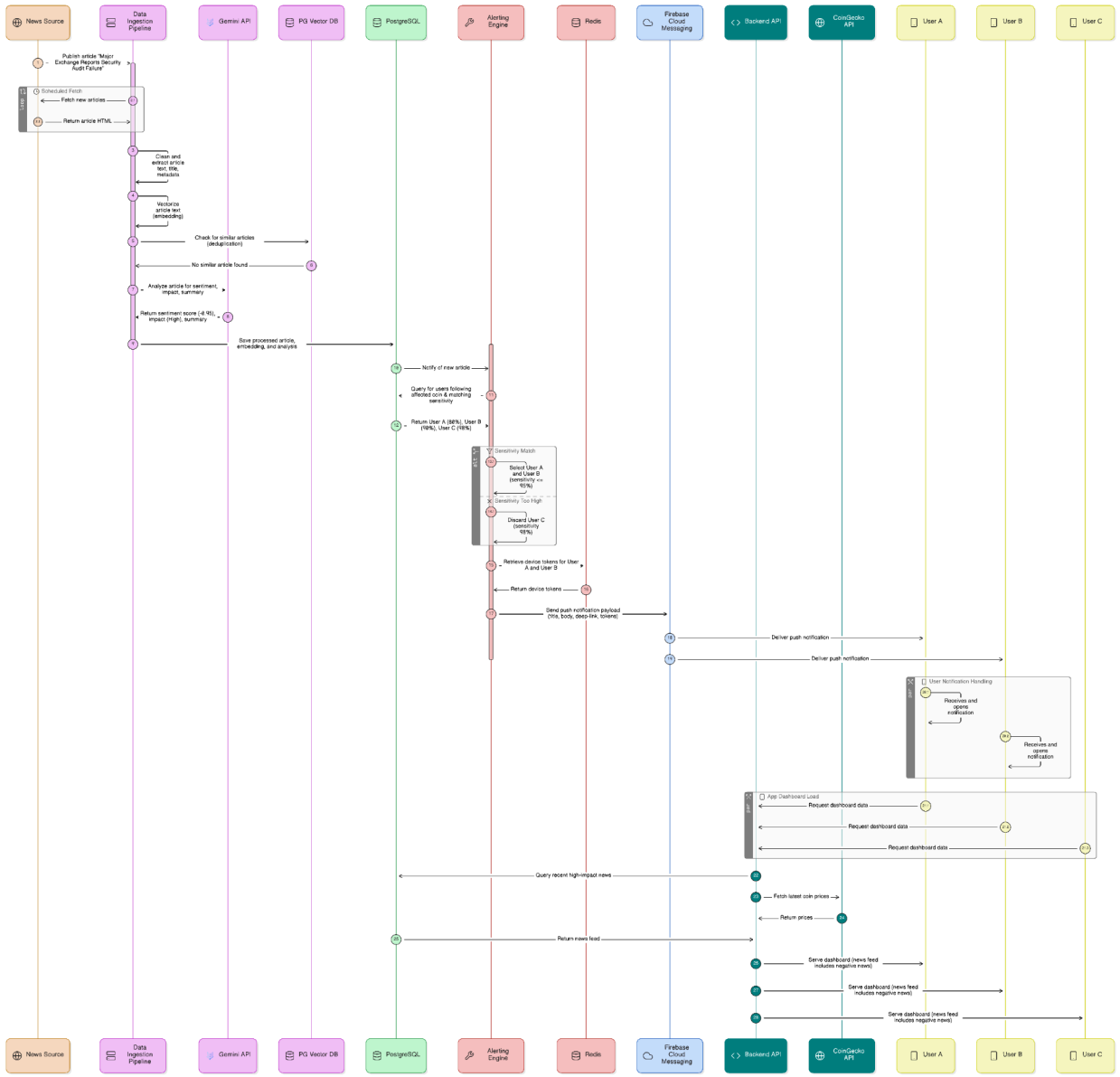
- Database: PostgreSQL (user data, historical alerts), vector DB, Redis stack (caching, real-time data)
- Deployment: Docker, AWS ECS/EKS, CI/CD via GitHub Actions

3. Data Design

Entity-Relationship Diagram (ERD)


twitter_account
id string pk
twitter_username string
twitter_password string
gmail_username string
gmail_password string
active_hours string
daily_limit int
status string
last_used timestamp
ai_accounts
id string pk
provider string
api_key string
status string
last_used timestamp
user_devices
id string pk
device_token string
device_type string
os_version string
last_active timestamp
is_active boolean
user_id string
user_settings
id string pk
alert_frequency string
alert_types string
do_not_disturb boolean
user_id string
push_notifications boolean
email_notifications boolean
users
id string pk
email string
created_at timestamp
last_login timestamp
is_active boolean
user_notifications
id string pk
user_id string
notification_type string
title string
message string
news_id &
sent_at timestamp
read_at timestamp
status string
news
id string pk
content_embeddings_id string fk
title string
content text
sentiment string
published_at timestamp
sentiment_score float
tags string
created_at timestamp
source_id string
content_embeddings
id varchar
content_vector vector
created_at timestamp
user_watchlist
id string pk
user_id string
created_at timestamp
coin_id string
news_source_logs
id string pk
news_source_id string
fetched_count int
filtered_non_crypto int
filtered_duplicates int
processed_count int
added_count int
log_time timestamp
status string
news_sources
id string pk
source_type string
handler string
display_name string
url string
is_active boolean
news_coins
id string pk
news_id string
coin_id string
coins
id string pk
symbol string
name string
is_active boolean

Sequence Diagrams



4. Component & Module Design

Component Breakdown:

Auth Module:
Handles user registration, subscription management, and session management using JWT tokens.

Data Ingestion Module:

- News Source Listener: Real-time monitoring and fetching of crypto-related news from verified sources and feeds
- Content Preprocessor: Cleans, filters, and normalizes article content
- Duplicate Detector: Identifies and handles duplicate or re-published content

AI Processing Module:

- Gemini API Client: Integration with Google's Gemini for NLP and sentiment analysis
- Sentiment Scorer: Converts AI analysis to 0–100% probability scores
- Context Analyzer: Identifies mentioned cryptocurrencies and market context

Notification Module:

- Push Service: Delivers alerts to mobile devices via Firebase Cloud Messaging
- Rate Limiter: Controls notification frequency per user
- Delivery Tracker: Monitors alert delivery and user engagement

User Preference Module:

- Coin Manager: Handles user watchlists and cryptocurrency preferences
- Threshold Manager: Manages user-defined sentiment thresholds
- Settings API: Provides CRUD operations for user configurations

Analytics Module:

- Performance Tracker: Measures sentiment accuracy and market impact
- User Behavior Analyzer: Tracks user engagement with different alert types
- A/B Testing Framework: Tests different scoring algorithms and thresholds

Component Responsibilities

- **Data Ingestion:** Real-time monitoring of crypto-related news, content preprocessing, duplicate detection, and initial filtering.

- **AI Processing:** Sentiment analysis using Gemini API, probability scoring, context extraction, and credibility assessment.

- **Notification Service:** Push notification delivery, rate limiting, delivery tracking, and engagement analytics.

- **User Management:** Profile management, subscription handling, preference storage, and access control.

- **Analytics:** Insight generation, accuracy tracking, and user engagement analysis.

5. API Design & Integrations

Public API Design:

- Strategy: RESTful APIs with WebSocket for real-time updates
- Format: JSON
- Versioning: URL path (e.g., /api/v1/...)

Authentication & Authorization:

- Method: JWT tokens for API authentication
- Flows:
    - Email login

Third-Party Integrations:

- AI Service: Gemini API for NLP and sentiment analysis
- Data Source: Multiple crypto and financial news sources (via web crawlers)
- Market Data: CoinGecko API for cryptocurrency metadata and price validation
- Push Notifications: Firebase Cloud Messaging for mobile alerts

6. Security Considerations

Authentication & Authorization Flow:

- JWT-based authentication for all API endpoints
- Role-based access control
- API rate limiting to prevent abuse
- Secure WebSocket connections for real-time data

Data Security:

- Encryption in Transit: TLS/SSL for all communications
- Data Privacy: User data encryption and anonymization where possible
- API Security: Secure storage of third-party API keys and credentials
- Compliance: GDPR-compliant data handling practices

Data Source Compliance:

- Adherence to data usage terms of partner news APIs and websites
- Proper attribution and citation for content usage
- Respect for robots.txt and scraping ethics

AI Ethics & Bias Mitigation:

- Regular model validation for bias detection
- Transparency in scoring methodology
- User feedback incorporation for continuous improvement