1. Overview of Multi-Agent System for the AI Viral Engine

A multi-agent system for this platform would consist of interconnected AI agents, each responsible for a specific function (e.g., data scraping, prediction, content creation, or promotion). These agents operate semi-autonomously, communicate with each other, and leverage an LLM (like Grok) for decision-making, content generation, or analysis. The system supports the SRS's requirements by:

- **Parallel Processing**: Agents handle simultaneous tasks (e.g., scraping TikTok while scoring trends).
- **Specialization**: Each agent focuses on a niche (e.g., sentiment analysis vs. token management).
- **Scalability**: Agents can be added or adjusted to handle increased data or user load.
- **Coordination**: Agents share data and decisions to ensure cohesive trend launches.

The system aligns with the SRS's emphasis on real-time trend prediction, automated content creation, and blockchain integration (e.g., prediction markets, tokens).

2. Agent Roles and Responsibilities

Based on the SRS, the following agents can be designed to cover the platform's key functions. Each agent uses AI (potentially LLM-based) and interacts with others via a central coordinator or direct communication.

2.1 Data Collection Agent

- **Purpose**: Fetches trend data from sources (TikTok, Reddit, X, Google Trends) as per FR3.2.
- **Tasks**:
  - Use APIs (e.g., TikTok Trends API, Reddit API) or ethical web crawlers to collect real-time data on hashtags, posts, searches, and engagement metrics.
  - Monitor mentions of topics across platforms (FR1.4).
  - Clean and preprocess data (e.g., remove duplicates, normalize formats).
- **Tools**: Python libraries (BeautifulSoup, Scrapy), no-code tools (Apify, Octoparse), or APIs from ViralStat/Semrush.
- **Output**: Structured datasets (e.g., JSON with post IDs, views, likes, timestamps).
- **Interaction**: Sends data to Prediction Agent and Scoring Agent; receives instructions on new sources from Coordinator Agent.

2.2 Prediction Agent

- **Purpose**: Predicts hottest topics hourly, daily, weekly, or monthly (FR1.1).
- **Tasks**:
  - Use machine learning models (e.g., time-series forecasting, LSTM) to identify emerging trends.
  - Filter out outlier influences (e.g., celebrity-driven events) to focus on organic trends.
  - Incorporate a "search engine meta finder" to aggregate search engine insights.
- **Tools**: TensorFlow, PyTorch, or LLM-based forecasting (e.g., Grok with DeepSearch mode).
- **Output**: List of predicted topics with metadata (e.g., category, platform, growth rate).
- **Interaction**: Receives raw data from Data Collection Agent; sends predictions to Scoring Agent.

2.3 Scoring Agent

- **Purpose**: Assigns virality scores (0-100) to predicted topics (FR1.3).

- **Tasks**:
  - Analyze factors: audience size/demographics, regions, similar past trends, key personalities, engagement metrics.
  - Use data from similar platforms (e.g., Semrush API, Exploding Topics) for benchmarking.
  - Flag topics with 90+ scores for launch.
- **Tools**: LLM for sentiment analysis, statistical models for scoring, APIs for historical trends.
- **Output**: Scored topics with justification (e.g., "90: High TikTok engagement in US").
- **Interaction**: Receives predictions from Prediction Agent; sends high-scoring topics to Launch Agent.

2.4 Content Generation Agent

- **Purpose**: Automates creation of trend pages and content (FR2.1).
- **Tasks**:
  - Generate narrative summaries, project details, descriptions, and logos for trend pages.
  - Create promotional content (e.g., tweets, TikTok scripts) tailored to platforms.
  - Populate token pages with activity and sentiment data (FR2.4).
- **Tools**: LLM (e.g., Grok for text generation), AI image tools (e.g., DALL-E for logos).
- **Output**: Fully populated trend pages, social media posts, visual assets.
- **Interaction**: Receives high-scoring topics from Scoring Agent; sends content to Promotion Agent.

2.5 Launch Agent

- **Purpose**: Manages "cult launches" for high-scoring trends (FR2.1).
- **Tasks**:
  - Initiate trend launches with 90+ scores, creating token pools or liquidity pools.
  - Integrate prediction markets for betting on trend outcomes (FR2.2).
  - Implement Anti-MEV dynamic fees for blockchain transactions (FR2.4).
- **Tools**: Blockchain APIs (e.g., Ethereum, Web3.js), smart contract templates.
- **Output**: Launched trend pages with active markets and token pools.
- **Interaction**: Receives trend data from Scoring Agent; coordinates with Promotion Agent for rollout.

2.6 Promotion Agent

- **Purpose**: Deploys AI bots on X for engagement and promotion (FR2.3).
- **Tasks**:
  - Post, retweet, and interact with trend-related content on X.
  - Update agent profile with all trend-related activity.
  - Respond to mentions and engage with trending hashtags.
- **Tools**: X API (premium), LLM for generating responses, automation frameworks (e.g., n8n).
- **Output**: Increased trend visibility via X interactions.
- **Interaction**: Receives content from Content Generation Agent; reports engagement metrics to Analytics Agent.

2.6 Analytics Agent

- **Purpose**: Displays trend analytics and insights on the user dashboard (FR2.5).
- **Tasks**:
  - Aggregate interactions, sentiment, and activity across platforms (FR2.4).

    - Visualize data by region, category, or time frame.
    - Provide real-time updates (every 5-15 minutes, per NFR1).
- **Tools**: Chart.js for visualizations, MongoDB for data storage, LLM for sentiment analysis.
- **Output**: Interactive dashboard with charts, metrics, and trend summaries.
- **Interaction**: Receives data from all agents; provides feedback to Coordinator Agent.

2.7 Coordinator Agent

- **Purpose**: Manages agent interactions and workflow orchestration.
- **Tasks**:
    - Assign tasks based on system goals (e.g., prioritize new trends).
    - Resolve conflicts (e.g., contradictory predictions).
    - Ensure compliance with ethical guidelines (e.g., rate-limiting crawlers).
- **Tools**: Agent orchestration platforms (e.g., LangChain, CrewAI), message queues (RabbitMQ).
- **Output**: Seamless operation across agents.
- **Interaction**: Communicates with all agents; logs system performance.

2.8 API Agent

- **Purpose**: Exposes APIs for external integration (FR3.1).
- **Tasks**:
    - Provide endpoints for trend predictions, scores, and launches.
    - Allow users to query specific tokens or trends.
    - Ensure secure access (e.g., API keys, rate limits).
- **Tools**: RESTful API frameworks (FastAPI, Flask), OAuth for authentication.
- **Output**: Public API endpoints (e.g., /trends/predict).
- **Interaction**: Receives data from Analytics and Scoring Agents; serves external developers.

3. Multi-Agent System Architecture

The MAS can be structured as follows:

- **Centralized Coordination**: The Coordinator Agent acts as a hub, managing task delegation and data flow. Agents communicate via APIs or message queues (e.g., Kafka, RabbitMQ).
- **Decentralized Execution**: Each agent operates independently, using local LLMs or APIs for tasks, but syncs with the Coordinator for updates.
- **Data Flow**:
    - Data Collection → Prediction → Scoring → Content Generation → Launch → Promotion → Analytics → API.
    - Feedback loops: Analytics Agent reports metrics to Prediction Agent for model refinement.

4. Practical Applications of Multi-Agent System

The MAS enhances the platform by addressing SRS requirements in the following ways:

- **Real-Time Trend Prediction (FR1.1)**: Data Collection and Prediction Agents work in tandem to fetch and analyze data, ensuring hourly/daily updates.
- **Virality Scoring (FR1.3)**: Scoring Agent uses LLM-driven sentiment analysis and historical data to assign accurate scores, reducing human bias.

- **Automated Launches (FR2.1)**: Content Generation and Launch Agents streamline page creation and token pool setup, enabling rapid “cult launches.”
- **Cross-Platform Promotion (FR2.3)**: Promotion Agent amplifies trends on X, increasing visibility and engagement.
- **Scalable Analytics (FR2.5)**: Analytics Agent provides customizable dashboards, meeting NFR1’s performance requirements.
- **API Integration (FR3.1)**: API Agent enables third-party developers to build on the platform, aligning with the SRS’s vision for extensibility.
- **Blockchain Features (FR2.4)**: Launch Agent’s Anti-MEV fees and prediction markets add a gamified, decentralized layer, unique compared to platforms like ViralStat.