

Zero-Cost, Next-Level Multi-Agent Advertising Brain App

This blueprint outlines a **mind-blowing**, fully functional advertising brain app you can build and demo at **zero cash cost**, leveraging free-tier services, open-source tools, and multi-agent orchestration. It includes:

- **Architectural Overview**
- **Agent Definitions & Roles**
- **Core Prompt Library**
- **Tech Stack & Free-Quota Components**
- **User Flow & Interaction Points**
- **Deployment Plan on Replit + Streamlit**

1. Architectural Overview

```
flowchart LR
    subgraph UI
        A[Streamlit Frontend]
    end
    subgraph Orchestration
        B[n8n (in-process)]
        C[LangGraph]
    end
    subgraph Agents
        D[TrendHarvester]
        E[AnalogicalReasoner]
        F[CreativeSynthesizer]
        G[BudgetOptimizer]
        H[PersonalizationAgent]
    end
    subgraph Data
        I[Replit DB]
        J[Qdrant Vector Store]
        K[Replit File Storage]
    end
    A --> |"Run Campaign"| B
    B --> C --> D --> E --> F --> G --> H
    D --> J
    E --> J
    F --> I
    G --> I
    H --> K
```

- **Streamlit UI** triggers workflows, displays results, and houses stored data.
- **n8n library** kicks off LangGraph pipelines in-process—no external orchestration server.

- **LangGraph** defines data flow between agents.
- **Replit DB** stores campaign metadata; **Replit Files** hold static assets.
- **Qdrant** handles vector embeddings for analogical insights and feedback loops.

2. Agent Definitions & Roles

1. **TrendHarvester**
2. **Prompt:** "You are a Trend Harvester. Given topic `{query}`, list the top 5 emerging micro-trends with descriptions and social signals."
3. **Data Sources:** Mock JSON or free social API.
4. **AnalogicalReasoner**
5. **Prompt:** "You are an Analogical Reasoner. Map trend `{trend}` to brand `{brand}` using an evocative analogy."
6. **Memory:** Embeddings stored in Qdrant for similarity reuse.
7. **CreativeSynthesizer**
8. **Prompt:** "You are a Creative Synthesizer. Craft an ad headline and short copy invoking analogy: `{analogy}`."
9. **BudgetOptimizer**
10. **Prompt:** "You are a Budget Optimizer. Given historical performance metrics `{metrics}`, recommend spend reallocation across channels to maximize ROI."
11. **PersonalizationAgent**
12. **Prompt:** "You are a Personalization Agent. Create a 1:1 user journey touchpoint plan based on user profile `{profile}` and campaign `{campaign}`."

3. Core Prompt Library

```
trend_prompt: |
```

```
You are a Trend Harvester. Given the topic: {query}, return a bulleted list of the top 5 emerging micro-trends, each with a one-sentence description and an engagement score.
```

```
analogy_prompt: |
```

```
    You are an Analogical Reasoner. Translate the trend "{trend}" into an analogy for the brand "{brand}" that sparks creative insights.
```

```
creative_prompt: |
```

```
    You are a Creative Synthesizer. Using the analogy: {analogy}, write a punchy ad headline and 2-3 lines of ad copy.
```

```
budget_prompt: |
```

```
    You are a Budget Optimizer. Analyze the metrics: {metrics_csv}. Propose percentage spend adjustments across channels: Google, Meta, Programmatic, Email.
```

```
personalization_prompt: |
```

```
    You are a Personalization Agent. Draft a multi-step user journey with personalized messaging for user attributes: {profile_json}.
```

4. Tech Stack & Free-Quota Components

- **Replit + Streamlit**: Hosting UI and Python runtime (free always-on boost).
 - **n8n (npm library)**: In-process workflow orchestration (open-source).
 - **LangChain & LangGraph**: Multi-agent orchestration and memory management.
 - **Mistral 7B**: Via Hugging Face free Inference API.
 - **Gemini Flash Pro**: Google AI Studio free-tier text2text & text2im.
 - **Stable Diffusion**: `diffusers` fallback with Hugging Face free endpoint.
 - **Qdrant CE**: Self-hosted in a Replit container for vector store.
 - **Replit DB & Files**: Key-value and file storage for metadata and assets.
-

5. User Flow & Interaction Points

1. **Home Screen**: Text input for campaign topic, brand selector, user-profile upload.
 2. **Run Campaign**: Button triggers n8n-&-LangGraph workflow: Trend → Analogy → Creative → Budget → Personalization.
 3. **Results Dashboard**:
 4. **Trends** table with scores.
 5. **Analogies** list.
 6. **Ad Copy & Creative** side-by-side.
 7. **Budget Allocation Chart** (`st.bar_chart`).
 8. **Personalization Plan** steps.
 9. **Save & Export**: Store campaign in Replit DB; allow CSV/PNG export of results.
-

6. Deployment Plan on Replit + Streamlit

1. **Initialize Replit**: Create Python project, add `requirements.txt`, set secrets.
 2. **Start Qdrant**: Run in separate container: `qdrant-server --host 0.0.0.0 --port 6333`.
 3. **Develop** ``: Import all libs, define prompts, initialize agents, build LangGraph, setup Streamlit UI.
 4. **Configure n8n**: Install via npm in Replit shell, import library, define workflows in code or JSON.
 5. **Test Locally**: Use Streamlit `run` and verify API calls with caching enabled.
 6. **Demo URL**: Share the Replit-generated public URL—no extra hosting required.
-

With this **comprehensive app blueprint**, you'll showcase a fully autonomous, multi-agent advertising brain at **zero cost**, ready to impress any audience or potential partner.