GNX - CIS: AI-Powered LinkedIn Content Generation & Optimization System

Author	AI Architect		Initial Update	Checkpoint Finished	Updated On	Future project
Version	1.0		October 15, 2025	1.1	October 17, 2025	2.0
					October 19, 2025	2.1
					October 20, 2025	2.5
					October 22, 2025	3.0

PRD: LinkedIn AI Content Co-pilot (v1.0)

- **Product:** LinkedIn AI Content Co-pilot
- **Version:** 1.0 (Genesis)
- **Status:** Live (Internal)
- **Author:** AI Architect
- **Date:** October 15, 2025

1. Objective & Problem Statement

Senior professionals, technical leaders, and consultants need to maintain a consistent, high-quality presence on LinkedIn to build their professional brand, attract talent, and engage with their industry. However, the process of ideating, drafting, and refining content is time-consuming and requires significant creative effort.

The **LinkedIn AI Content Co-pilot** solves this by acting as an intelligent assistant that learns the user's unique voice and style to generate high-engagement draft posts, ready for review and approval. Our objective is to **reduce content creation time by 90%** while **improving post quality and engagement potential.**

2. Target Audience & User Persona

- **Persona:** "The Expert Practitioner" (e.g., Senior SAP Consultant, CTO, Engineering Manager)
- **Goals:**
  - Share deep technical expertise and project learnings.
  - Build a reputation as a thought leader in their niche (e.g., Enterprise AI, SAP Integration).
  - Engage in meaningful conversations with peers.
- **Pain Points:**
  - "I have the knowledge, but not the time to write posts consistently."
  - "I struggle to make my technical content engaging for a broader audience."
  - "I'm not sure what content will perform well."

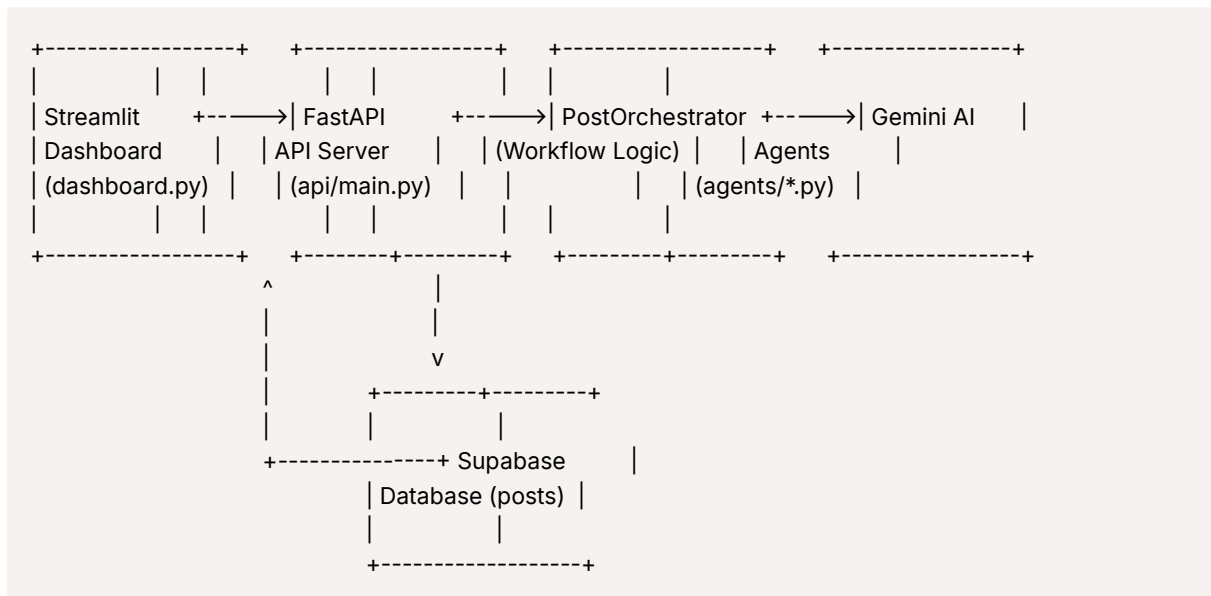## 3. What We Have Achieved (Features in v1.0)

We have successfully built and deployed a full-stack, multi-agent AI system with the following core capabilities:

Feature ID	Feature Name	User Benefit	Technical Implementation
FE-01	Style-Aware Content Generation	Generates posts that sound authentically like the user, not a generic AI.	`HistoryAgent` analyzes past posts (mocked); `ContentAgent` uses this style profile in its prompt to Gemini.
FE-02	Advanced Virality Analysis	Provides a detailed, objective score and actionable suggestions to improve every post.	`ViralityAgent` uses a detailed rubric to score posts on 8 criteria and provides a breakdown and suggestions.
FE-03	Centralized Draft Management	A single source of truth for all generated content, allowing for a structured review process.	All generated drafts are stored in a Supabase `posts` table with a `draft` status.
FE-04	Interactive Control Panel	Provides a simple, intuitive UI to generate, review, and approve content without touching code.	A Streamlit `dashboard.py` application that interacts with our backend API.
BE-01	Scalable API Backend	A robust, production-ready API that orchestrates the entire workflow and can support future applications.	A FastAPI server with endpoints for generation, retrieval, and approval.

4. Technical Architecture Overview

The system is designed with a decoupled, service-oriented architecture, ensuring maintainability and scalability.

- **Frontend:** A **Streamlit** dashboard provides the user interface for all interactions.
- **Backend API:** A **FastAPI** server exposes RESTful endpoints for all core functions.
- **Orchestrator:** A central `PostOrchestrator` class manages the workflow between agents.
- **AI Core:** A multi-agent system powered by **Google Gemini** (`Content` , `Virality` , `History` agents).
- **Database:** A **Supabase (PostgreSQL)** instance serves as the persistent data store for all posts.

```
+------------------+    +------------------+    +------------------+    +-----------------+
|                  |    |                  |    |                  |    |                 |
| Streamlit    +---—→| FastAPI      +--—→| PostOrchestrator  +--—→| Gemini AI   |
| Dashboard        |    | API Server   |    | (Workflow Logic) |    | Agents          |
| (dashboard.py)   |    | (api/main.py)|    |                  |    | (agents/*.py)   |
|                  |    |              |    |                  |    |                 |
+------------------+    +--------+---------+    +---------+---------+    +-----------------+
                            ^                        |
                            |                        |
                            |                        v
                            |            +---------+---------+
                            |            |                   |
                  +---------------+ Supabase        |
                  | Database (posts) |
                  |                   |
                  +-------------------+
```

5. What to Do Next: Future Roadmap (v1.1 and beyond)

Now that the core content factory is operational, we can focus on building a closed-loop system and adding high-value automation features.

Feature ID	Feature Name	Description	Business Value
RD-01	Automated Publishing	After a user clicks "Approve" in the dashboard, the system will automatically	Completes the automation loop, moving from content creation to distribution.

Feature ID	Feature Name	Description	Business Value
		publish the post to LinkedIn at a scheduled time.	
RD-02	**Performance Analytics Loop**	A scheduled task will fetch metrics (likes, comments) for published posts and feed this performance data back into the `HistoryAgent`.	Creates a self-improving system that learns not just the user's *style*, but what content *actually performs well*.
RD-03	**Content Scheduling**	Allow users to approve a post and schedule it for a future date and time from the dashboard.	Enables content batching and strategic timing for maximum impact.
RD-04	**Topic Suggestion Engine**	An agent that analyzes past high-performing posts and industry trends to suggest new topics for the user to write about.	Solves the "blank page" problem and keeps content relevant and fresh.