SYSTEM ARCHITECTURE OVERVIEW

Daily Flow

1. Trigger
2. Scrape trending AI automation topics
3. Clean + deduplicate
4. Rank topics (engagement potential)
5. Retrieve context from your knowledge base (RAG via Pinecone)
6. Generate post (Gemini)
7. Generate Twitter version
8. Generate slide-based LinkedIn carousel text (optional)
9. Human approval (optional but recommended)
10. Schedule + post to LinkedIn
11. Schedule + post to Twitter
12. Log performance
13. Update vector memory with posted content

STACK (LOW COST)

- Orchestrator: n8n (self-hosted)
- Vector DB: Pinecone (free tier: 1 index, 100k vectors)
- LLM: Google Gemini (use cheapest model available)
- Embeddings: Gemini text-embedding-3-small
- Scraping: HTTP Request + RSS feeds
- Storage: Google Drive / local JSON
- Scheduling: native n8n Cron
- Social APIs:

    A. LinkedIn API

    B. X API (formerly Twitter)

N8N WORKFLOW - NODE BY NODE

WORKFLOW 1 — DAILY CONTENT GENERATION

1 Cron Node

Type: Cron
Trigger: Every day at 8:00 AM

2 Collect Trending Topics

Use multiple nodes in parallel (Merge later).

A. Reddit AI Automation

Node: HTTP Request
Method: GET
Endpoint:

```
https://www.reddit.com/r/ArtificialIntelligence/top.json?limit=10&t=day
```

Repeat for:

- r/automation
- r/MachineLearning

No API key required.

B. RSS Feeds (Industry Blogs)

Use RSS Feed Node:

Examples:

- Google AI Blog RSS
- OpenAI Blog RSS
- TechCrunch AI RSS

C. X Trending AI Search

If API available:
Use X search endpoint for:

```
"AI automation" OR "AI agent" OR "n8n" -filter:replies
```

If no API → skip to RSS + Reddit only (still works).

3 Clean + Normalize Data

Node: Function Node

Extract:

```
titlesummary
url
```

```
engagement_score (upvotes or retweets)source
```

Return standardized JSON.

4 Remove Duplicates

Node: Item Lists → Remove Duplicates
Field: title

5 Topic Scoring Node

Use Function Node to score:

```
score =
(engagement * 0.6)+ (recency_weight * 0.3)+ (keyword_match_bonus *
0.1)
```

Keyword bonus if contains:

- AI agents
- Automation
- no-code
- Workflow
- RAG

Sort descending.

Take top 1.

RAG SYSTEM FLOW

6 Generate Embedding (Trending Topic)

Node: HTTP Request to Gemini embedding endpoint

Input:

```
"Topic: {{title}}. Summary: {{summary}}"
```

Store embedding vector.

7 Query Pinecone

Node: HTTP Request to Pinecone Query API

Pass:

```
topK = 5includeMetadata = true
```

Returns:

- Relevant past posts
- Your expertise notes
- Saved insights

8 Combine Context

Function Node:

Construct:

```
Trending Topic:
{{title}}

Industry Summary:
{{summary}}

My Expertise:
{{top 5 retrieved notes}}
```

CONTENT GENERATION

9 Generate LinkedIn Post (Gemini)

Node: HTTP Request to Gemini LLM

Prompt Structure:

```
You are a senior AI automation strategist.

Write a high-engagement LinkedIn post.

Topic: {{title}}
```

```
Context: {{retrieved_context}}

Constraints:
- Hook in first line
- 8-12 short lines
- Clear insight
- Strategic take
- End with engaging question
- No emojis
- Confident tone
```

Return: linkedin_post

10 Generate Twitter Version

Same model.

Prompt:

```
Rewrite this as a high-impact X thread.
Max 4 tweets.
Each tweet under 280 chars.
Punchy.
```

Return: twitter_thread

11 Generate Slide Carousel (Optional)

Prompt:

```
Convert this into 5 LinkedIn carousel slides.
Each slide max 25 words.
Slide 1 = hook.
Slide 5 = CTA.Return JSON:
[
 {slide:1,text:""},
 {slide:2,text:""}
]
```

OPTIONAL: HUMAN APPROVAL NODE

Before auto-posting, add:

Node: Telegram / Email Approval

Send preview.

Buttons:

- Approve
- Reject
- Regenerate

If no approval needed → skip.

AUTO POSTING

12 LinkedIn Post Node

Use LinkedIn OAuth2.

Endpoint:

`ugcPosts`

Content type:

- Text only OR
- Document upload for carousel

13 Twitter Post Node

Use X API v2:

If thread:

- Post first tweet
- Loop remaining with reply_to

Use SplitInBatches Node.

PERFORMANCE LOGGING

---

14 Save Post Data

Store:

- Topic
- Generated post
- Date
- URL

Use:

- Google Sheets Node
  OR
- Local JSON file

FEEDBACK LOOP

15 Engagement Collection Workflow (Separate Cron)

Daily:

- Fetch last 5 posts
- Get impressions, likes, comments

Store metrics.

16 Update RAG Memory

Create summary:

"This post performed well because..."

Generate embedding.

Upsert to Pinecone.

Your system now learns.

INITIAL RAG SETUP (ONE-TIME WORKFLOW)

Step 1: Upload Your Knowledge

Sources:

- Past LinkedIn posts
- Blog articles
- Notes
- Expertise areas
- Case studies

Use Google Drive Node → Read Files.

Step 2: Chunk Text

Function Node:
Split into 300–500 token chunks.

Step 3: Generate Embeddings

Call Gemini embedding endpoint.

Step 4: Upsert to Pinecone

Index:

- `LinkedIn-personal-brand-index`

Metadata:

- `Source`
- `date`
- `topic`

COST MINIMIZATION STRATEGY

Scrape RSS instead of paid APIs

Use Gemini cheapest model

Limit topK retrieval to 3–5

Keep posts under 500 tokens

Run once daily only

Estimated cost/month:
No cost

ADVANCED UPGRADE (PHASE 2)

Add:

1. Positioning Memory Layer

Track:

Topics you already posted

-
-

- Avoid repetition

2. Content Mix Logic

Rotate:

- Educational
- Contrarian
- Case study
- Tool breakdown
- Trend commentary

3. Multi-Angle Generator

Generate 3 hooks.
Select strongest using scoring function.

ENGINEERING RISKS

LinkedIn API restrictions (strict permissions)

- X paid API limitations
- Scraping reliability
- Rate limits
- Shadow banning (avoid spam-like behavior)

Mitigation:

- Limit to 1 post per day
- Avoid link-only posts
- Maintain human review initially

FINAL SYSTEM MAP

```
Cron
  ↓ Scrape (Reddit + RSS)
  ↓ Clean + Rank
  ↓ Embedding
  ↓ Pinecone Retrieval
  ↓ LLM Generate
  ↓ Format Split (LinkedIn / Twitter)
  ↓ Approval
  ↓ Schedule
  ↓ Post
```

↓ Log
↓ Feedback Loop

Strategic Insight

This project isn’t about automation.

It’s about building:

- Authority compounding system
- Asymmetric leverage
- A distribution engine powered by your expertise