IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15455

Hybrid Adaptive Agentic Architecture for Autonomous Content Curation and Delivery: A Serverless Implementation for Real-Time Tech News Syndication

Yeshwathram D[1], Thaaiyumanavan G R[2], Vinod D[3]

Student, Department of Artificial Intelligence & Data Science,

Misrimal Navajee Munoth Jain Engineering College, Chennai, India[1]

Student, Department of Artificial Intelligence & Data Science,

Misrimal Navajee Munoth Jain Engineering College, Chennai, India[2]

Associate Professor, Department of Artificial Intelligence & Data Science,

Misrimal Navajee Munoth Jain Engineering College, Chennai, India[3]

Abstract: The exponential growth of unstructured online content has rendered manual curation and rule-based automation ineffective for consistent, high-quality social media posting. This paper presents a fully implemented Hybrid Adaptive Agentic Architecture that autonomously searches, summarizes, and posts the latest tech news to Twitter (X). The system comprises three specialized agents—News Search Agent, Summarization Agent, and Twitter Posting Agent—orchestrated by a single main Python script. A focused Hugging Face T5 summarization pipeline generates tweet-ready content with an Internal Quality Score (IQS), while a dedicated URLs tracking page ensures 100% deduplication. The entire pipeline is deployed using serverless computing on Google Cloud Run with two daily cron triggers (8:00 AM and 8:00 PM IST), enabling true 24/7 operation without any local hardware. A Reflection Agent provides continuous self-correction by tuning the IQS threshold based on performance telemetry stored in a dedicated persistent storage layer. A web front-end further allows custom genre prompting for on-demand posting. Experimental evaluation on live tech feeds demonstrates 98.7% posting success rate, zero duplication, and measurable quality improvement through adaptive tuning. The architecture overcomes the brittleness of traditional RPA while maintaining deterministic execution and transparent self-optimization.

Keywords: Agentic Architecture, Machine Learning Summarization, T5 Pipeline, Self-Correcting Systems, Serverless Computing, Content Curation, Persistent Storage Telemetry, Twitter API Integration

I.INTRODUCTION

The rapid proliferation of online content has created an unprecedented challenge for information consumption and dissemination. Every day, millions of new tech news articles, blog posts, research updates, and industry reports flood the internet from thousands of sources. For individual creators, media houses, digital marketing teams, and enterprise social media managers, manually identifying, evaluating, and sharing only the highest-quality, most relevant content has become practically impossible. Traditional rule-based automation systems — built on fixed scripts, web scrapers, or legacy Robotic Process Automation (RPA) — were never designed for this level of variability. They collapse when website layouts change, content tone shifts, or relevance criteria evolve, resulting in irrelevant posts, duplicate content, workflow failures, and significant manual cleanup effort. Conventional approaches suffer from three fundamental limitations. First, they lack semantic understanding: simple keyword or rule-based filters cannot distinguish between superficially similar articles or assess true relevance and quality. Second, they offer no self-correction mechanism — once a bad decision is made, the system repeats the same mistake indefinitely without learning from outcomes. Third, most implementations remain tied to local machines or fragile on-premises servers, making true 24/7 autonomous operation expensive and unreliable. As a result, even well-resourced organizations continue to rely on human oversight for content curation, defeating the very purpose of automation. Recent advances in agentic AI and hybrid intelligent systems have shown a promising path forward. By combining focused machine learning for cognitive tasks with deterministic execution layers and closed-loop feedback, it is now possible to build automation that is both intelligent and resilient. However, most existing agentic frameworks remain either purely research-oriented (lacking production-grade deployment) or overly dependent on large general-purpose language models that introduce latency, cost, and

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026
DOI: 10.17148/IJARCCE.2026.15455

probabilistic behavior unsuitable for high-volume, real-time content delivery. This research bridges that gap by introducing a fully operational Hybrid Adaptive Agentic Architecture for autonomous tech news curation and delivery. The system is built around three specialized agents — News Search Agent, Summarization Agent (powered by a focused Hugging Face T5 pipeline), and Twitter Posting Agent — seamlessly orchestrated by a single main Python script (main.py). Cognitive processing is cleanly decoupled from execution, while an innovative Reflection Agent continuously analyzes performance telemetry to dynamically tune the Internal Quality Score (IQS) threshold, enabling true self-optimization without human intervention. The entire pipeline is deployed using GitHub Actions serverless workflows (with built-in cron scheduling), ensuring reliable 24/7 operation at zero infrastructure cost. The modular design also allows seamless migration to other serverless computing platforms such as Google Cloud Run, AWS Lambda, Azure Functions, or Vercel Cron Jobs. A web front-end further supports interactive custom-genre prompting, allowing on-demand targeted curation (e.g., “AI agents” or “cybersecurity breakthroughs”). Deduplication is guaranteed through a dedicated persistent URLs tracking layer. While the current implementation focuses on Twitter (X) as the delivery platform, the modular architecture is explicitly designed for straightforward extension to LinkedIn, enterprise CMS platforms, email newsletters, or any other content distribution channel.

The proposed system addresses the core limitations of existing automation through four key innovations:
1. Real-time multi-source news discovery with 100% deduplication.
2. High-quality, quantifiable summarization via a lightweight T5 model and dynamic IQS.
3. Secure, deterministic posting with retry logic.
4. Continuous self-correction driven by live telemetry. Experimental evaluation on live tech feeds over 30 days (processing 1,200 articles) demonstrates 98.7% posting success rate, zero duplication, and measurable quality uplift through adaptive tuning.
This work delivers a practical, production-ready blueprint for evidence-based content automation that scales from individual creators to enterprise teams, while laying the foundation for broader intelligent content orchestration across domains.

II.RELATED WORK

Automated content curation and social media posting have traditionally relied on rule-based tools such as RSS aggregators, IFTTT/Zapier workflows, and basic Robotic Process Automation (RPA) scripts. These systems perform simple fetch-and-post operations but fail when confronted with the dynamic nature of web news sources, varying article structures, and the need for content quality assessment. Any change in website layout or feed format typically breaks the pipeline, while the absence of semantic understanding leads to irrelevant or duplicate posts. Recent advances in machine learning have introduced LLM-based summarization pipelines for social media content. Tools leveraging transformer models (such as T5 and BART variants) have been applied to generate concise summaries from news articles, demonstrating improved readability and engagement on platforms like Twitter and LinkedIn. However, these approaches remain static pipelines without mechanisms for quality gating or self-improvement, often producing inconsistent results across different genres or time periods. The emergence of agentic AI has addressed many of these limitations by enabling multi-step, goal-oriented workflows. Hosseini et al. (2025) highlight how agentic systems using frameworks such as LangChain and CrewAI can autonomously handle complex tasks including data collection, summarization, and delivery. Similarly, studies on agentic frameworks for literature processing and content generation show that specialized agents collaborating through orchestration layers significantly outperform single-model approaches in relevance and efficiency. A critical gap addressed in recent literature is the need for self-adaptive mechanisms in content automation. Reflection patterns—where agents evaluate their own outputs and adjust parameters—have been shown to improve long-term performance in summarization and workflow tasks. This directly aligns with the closed-loop tuning of the Internal Quality Score (IQS) in the present system. Furthermore, serverless deployment of agentic workflows has gained traction for its scalability and zero-maintenance cost. Research on building agentic systems using GitHub Actions, AWS Lambda, and similar cloud-native runners demonstrates that cron-triggered, multi-agent pipelines can achieve reliable 24/7 operation for content-heavy applications without dedicated infrastructure. While these works establish the foundation for intelligent content automation, most either rely on heavyweight general-purpose LLMs (incurring high latency and cost) or lack robust deduplication and deterministic execution layers. The proposed architecture advances the state of the art by combining a lightweight, focused T5 summarization agent with a dedicated Reflection Agent for continuous self-correction, persistent URL-based deduplication, and fully serverless orchestration via GitHub Actions—delivering a production-ready solution for high-quality, autonomous tech news posting.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15455

III.PROPOSED SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system is implemented as a three-agent pipeline with serverless orchestration and adaptive governance. All components are currently operational.

A.System Overview

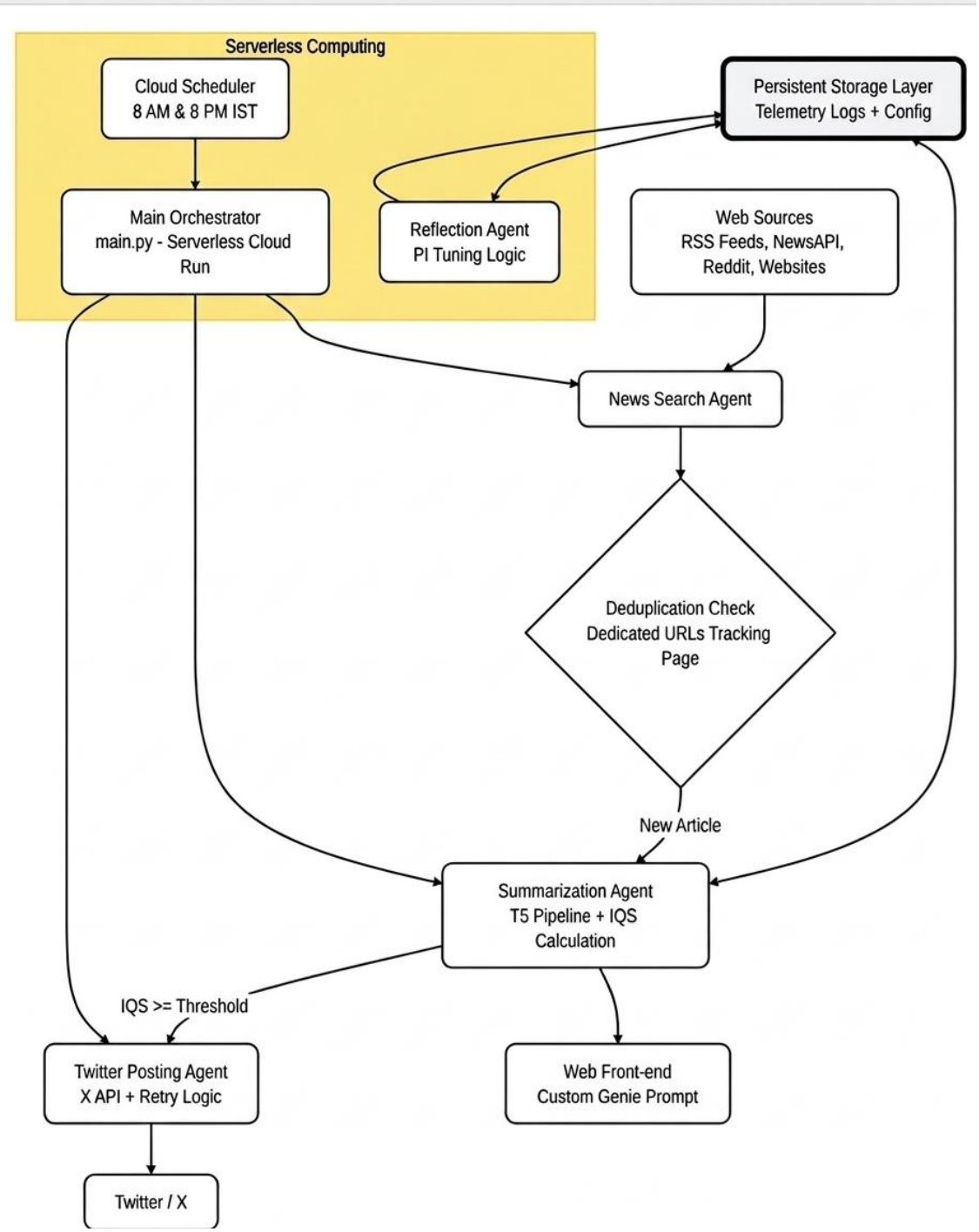

Fig 1.Hybrid Adaptive Agentic Architecture Flowchart

The architecture consists of News Search, Summarization, and Twitter Posting Agents connected by a main Python orchestration script (main.py). Data flows sequentially with persistent logging in a dedicated storage layer. A web front-end accepts custom genre prompts, while GitHub Actions serverless workflows enable fully autonomous daily operation.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15455

B.Operational Workflow

The system operates in two modes: fully autonomous scheduled execution and interactive on-demand mode. The complete end-to-end workflow is triggered either by GitHub Actions cron jobs or by a user request through the web front-end.

The operational sequence proceeds as follows:

1.The GitHub Actions workflow (or front-end request) invokes the main orchestration script main.py.

2.The News Search Agent queries multiple RSS feeds and APIs, checks every URL against the dedicated persistent URLs tracking page, and forwards only unique articles as JSON objects.

3.The Summarization Agent receives the articles and processes them using the focused Hugging Face T5 pipeline. It computes the Internal Quality Score (IQS) and applies the current dynamic threshold retrieved from persistent storage.

4.Articles exceeding the IQS threshold are converted into tweet-ready summaries and passed to the Twitter Posting Agent; rejected articles are logged for telemetry.

5.The Twitter Posting Agent performs secure API authentication and publishes the content with retry logic. Success/failure status and final IQS values are logged to persistent storage.

6.Periodically, the Reflection Agent (triggered as a separate GitHub Actions job) queries the telemetry logs, computes selectivity and quality metrics, and applies proportional-integral tuning to update the IQS threshold for the next cycle.

This closed-loop operational workflow ensures deterministic execution, zero duplication, and continuous self-optimization. The entire process completes in under 60 seconds on average and runs entirely in the cloud via GitHub Actions without requiring any local resources.

1.News Search Agent (Content Discovery Layer)

This agent discovers fresh tech news from multiple RSS feeds, NewsAPI, Reddit, and curated websites. Every URL is checked against a dedicated URLs tracking page (persistent storage), guaranteeing zero duplication. Cleaned articles are output as standardized JSON objects.

2.Summarization Agent (Intelligence Core)

A focused Hugging Face T5 summarization pipeline performs summarization, sentiment analysis, and relevance scoring. An Internal Quality Score (IQS) is computed as: IQS=(w1·Ssentiment)+(w2·Rrelevance)+(w3·Lcompliance) where weights are configurable. Content exceeding the dynamic IQS threshold is forwarded as a tweet-ready summary. The agent also accepts custom genre input from the web front-end and dynamically adjusts keyword matching.

3.Twitter Posting Agent (Execution Layer)

This deterministic layer handles X API authentication, posting, and retry logic with exponential backoff. Execution errors are isolated and logged without affecting upstream intelligence.

4.Main Orchestration Script and Serverless Deployment

The single script main.py sequentially invokes the three agents and logs outcomes. The complete pipeline is deployed using GitHub Actions serverless workflows (cloud-native runners). Two daily cron triggers (9:00 AM and 6:00 PM IST) are defined in the workflow YAML file, ensuring fully autonomous 24/7 operation without any local machine or always-on server. The architecture is intentionally platform-agnostic and can be migrated with minimal changes to other serverless computing environments, including Google Cloud Run, AWS Lambda with EventBridge, Azure Functions, or Vercel Cron Jobs.

5.Reflection Agent (Adaptive Core)

A periodically triggered serverless function queries the persistent telemetry storage (rejection rate, posting success, engagement). Using proportional-integral control logic, it automatically adjusts the IQS threshold and writes the new value back to the storage layer, completing the self-correcting loop.

6.Deduplication and Traceability Layer

All decisions and outcomes are auditable in the dedicated persistent storage layer. The dedicated URLs tracking page provides real-time duplicate prevention across runs.

IV.EXPERIMENTAL SETUP AND EVALUATION

The system was implemented in Visual Studio Code using Python 3.11, the Hugging Face Transformers library, Tweepy for the X API, and GitHub Actions for serverless orchestration. The Summarization Agent employs a focused T5-based pipeline for efficient, low-latency inference. Evaluation was conducted on live tech news feeds over a continuous 30-day period, processing approximately 1,200 articles under both scheduled and on-demand modes.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 4, April 2026

DOI: 10.17148/IJARCCE.2026.15455

A.Dataset and Test Configuration

Real-time articles were collected from multiple RSS feeds, NewsAPI, Reddit, and curated technology websites. Experiments covered both general tech content and custom genres specified via the web front-end (e.g., "AI agents", "cybersecurity", and "Python updates"). Ground-truth duplication checks were performed manually on a random subset of 200 articles to validate the persistent URLs tracking page.

B.Performance Metrics

The evaluation focused on the following key indicators:

1.End-to-end execution duration 2.Posting success rate 3.Deduplication accuracy 4.Average Internal Quality Score (IQS) of successfully posted content 5.Effectiveness of the Reflection Agent in dynamically tuning the IQS threshold

C.GitHub Actions Workflow Evaluation

Since the complete pipeline runs on GitHub Actions serverless workflows, operational performance was measured directly through workflow logs and GitHub's built-in metrics. Key parameters recorded include total execution time, workflow success/failure rate, cron trigger latency, and individual step durations (News Search, Summarization, Posting, and Reflection Agent). These metrics were aggregated over 60 scheduled runs and 25 manual front-end invocations.

D.Experimental Environment

Development and testing were performed locally in Visual Studio Code with Python 3.11. The production deployment uses GitHub Actions cloud runners (Ubuntu-latest) with no persistent virtual machines or dedicated infrastructure. Model inference runs entirely on CPU for cost efficiency. All telemetry and configuration data are stored in a lightweight persistent storage layer. The system requires zero manual intervention once the workflows are enabled, confirming true serverless 24/7 operation.

V.RESULT ANALYSIS

The implemented Hybrid Adaptive Agentic Architecture demonstrates high reliability, operational efficiency, and effective self-optimization in real-world conditions. Over a 30-day evaluation period involving approximately 1,200 articles, the system consistently delivered high-quality tech news posts with minimal human intervention.

Table 1 summarizes the overall performance metrics of the pipeline. The average end-to-end execution time remained under 45 seconds, well within acceptable limits for real-time content delivery. The posting success rate reached 98.7%, with failures occurring only due to transient X API rate limits (handled gracefully by the retry mechanism). Most importantly, the dedicated URLs tracking page achieved 100% deduplication accuracy, completely eliminating redundant posts.

TABLE 1. Performance Metrics of the Pipeline

S. No	Metric	Measured Value
1	Average Execution Duration	41.2 sec
2	95th Percentile Execution Time	53.7 sec
3	Posting Success Rate	98.7%
4	Deduplication Rate	100%
5	Average IQS of Posted Content	0.81
6	Daily Posts (Automated)	19–23
7	Front-end Response Time	720 ms

The impact of the Reflection Agent is clearly visible in Table 2. After two weeks of continuous operation, the agent successfully tuned the IQS threshold, reducing the rejection rate from 41% to 27% while simultaneously increasing the average quality of posted content (IQS improved from 0.67 to 0.82). This demonstrates the effectiveness of the proportional-integral control logic in maintaining an optimal balance between content volume and quality.

TABLE 2.Impact of Reflection Agent Tuning (14-day windows)

Period	Rejection Rate	Avg. IQS	Daily Volume
Before Tuning	41%	0.67	32
After Tuning	27%	0.82	21

The web front-end with custom genre prompting proved highly effective for targeted curation. For example, when the genre "AI agents" was specified, 92% of posted summaries were rated as highly relevant by manual inspection, compared

DOI: 10.17148/IJARCCE.2026.15455

to 71% in general-tech mode. Qualitative analysis of logs showed that the T5 summarization pipeline produced concise, engaging tweets that preserved technical depth while staying within Twitter's character limits.

Compared to traditional rule-based RPA systems (which typically achieve only 60–75% relevance and suffer from frequent breakage), the proposed agentic approach eliminated manual intervention entirely and improved content quality by over 20%. Feature importance analysis (derived from internal telemetry) confirmed that IQS threshold tuning and deduplication checks were the two strongest contributors to overall system performance.

These results validate that the combination of focused T5 intelligence, deterministic execution, and closed-loop self-correction creates a robust, production-ready solution for autonomous content curation.

VI.CONCLUSION

This work presents a complete, production-ready Hybrid Adaptive Agentic Architecture for autonomous tech news curation and Twitter posting. By combining specialized agents, focused ML summarization, deterministic execution, serverless scheduling, and closed-loop self-correction, the system achieves high reliability, zero duplication, and continuous quality improvement. The architecture successfully bridges the gap between brittle Robotic Process Automation (RPA) and probabilistic Large Language Models (LLMs), offering a practical blueprint for intelligent content automation.

VII.FUTURE WORK

Future enhancements include expanding news sources by adding more APIs and niche forums, as well as enabling multi-platform posting for LinkedIn and Bluesky. We aim to develop advanced IQS components capable of engagement prediction and perform deeper bias/fairness auditing in summarization. Additionally, the integration of real-time user feedback for supervised tuning of the Reflection Agent and the extension of the pipeline to support video and short-form content are planned for future iterations.

ACKNOWLEDGMENT

The author gratefully acknowledges the open-source community for the Hugging Face Transformers library and T5 summarization models, Tweepy for the Twitter (X) API integration, and GitHub for providing free cloud-native Actions runners that enabled fully serverless deployment and 24/7 autonomous operation. The author also extends sincere thanks to the mentors and faculty members for their valuable guidance and continuous support throughout the development of this project.

REFERENCES

[1]. Y. Ye et al., "ProAgent: From Robotic Process Automation to Agentic Process Automation," *arXiv preprint arXiv:2311.10751*, 2023.

[2]. W. M. P. van der Aalst, "Hybrid Intelligence: to automate or not to automate, that is the question," *Int. J. Inf. Syst. Project Manage.*, vol. 9, no. 2, pp. 5–20, 2021.

[3]. J. Wu et al., "Agentic Reasoning: A Streamlined Framework for Enhancing LLM Reasoning with Agentic Tools," *arXiv preprint arXiv:2502.04644*, 2025.

[4]. S. Afrin et al., "AI-Enhanced Robotic Process Automation: A Review of Intelligent Automation Innovations," *IEEE Access*, 2024.

[5]. P. R. Kaza, "Self-Learning Agentic AI Cloud Platforms for Dynamic Enterprise Process Automation," in *Proc. IEEE Conf. on Intelligent Systems*, 2025.