

Title: Engineering a Modern Research Website Backend with an AI-Powered Content Generation Pipeline

Supervisors: Prof. Gianvito Urgese, Dr. Giuseppe Fanuli

Student: Yuxuan Wu

1. Introduction and Motivation

Despite the rapid acceleration of information flow facilitated by modern internet technologies, websites dedicated to academic research often lag behind. The content remains largely static, predominantly relying on single-format PDF files. This presentation creates a significant barrier between researchers and general audiences, as non-experts often find dense technical papers difficult to comprehend.

Furthermore, traditional dynamic web platforms suffer from high administrative overhead. Administrators are required to manually upload files and update database entries, leading to unavoidable data latency where new research appears online days or even weeks after publication.

The objective of this thesis is to address these challenges by developing the inNuCE Web Platform, a full-stack system designed to automatically transform static academic papers into diverse, interactive digital resources.

2. System Architecture

To ensure stability and security, the system adopts a **Micro-services inspired architecture**, fully containerized using **Docker** to facilitate reproducible deployment.

A **Dual-Workflow Access Architecture** was designed to strictly separate public user access from administrator operations, as shown in Figure 1.

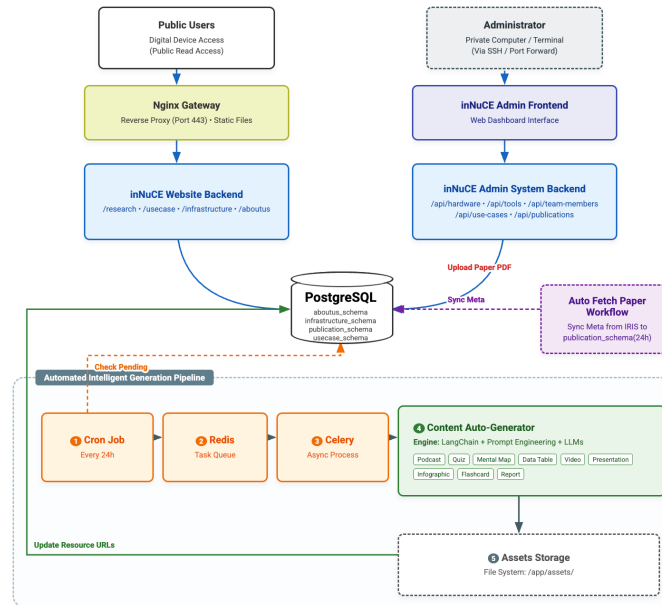


Figure 1 Overall Technic System Architecture of inNuCE Platform

Public Flow: Users access the website via standard HTTPS. The request goes through **Nginx Gateway** to the **inNuCE Backend**, which is built on Node.js to optimize for high-concurrency reading operations.

Admin Flow: Administrators manage the system using a private computer. They connect via a secure **SSH Tunnel** to access the **Admin Backend**. This design protects the sensitive management interface from public internet attacks.

The data layer uses **PostgreSQL** to store metadata. It is divided into four schemas to manage publications, use cases, and team info separately. It also implemented an "Auto Fetch" workflow that synchronizes data from the official IRIS system every 24 hours.

3. Automated Intelligent Generation Pipeline

The core innovation of this thesis is the Automated Intelligent Generation Pipeline, which replaces manual work with AI agents. The pipeline integrates Cron Job, Redis, Celery, and Large Language Models (Google Gemini). The workflow operates as follows:

Scheduling: A Cron job checks the database every 24 hours. If it finds a paper with uploaded document but without generated resources, it pushes the tasks to Redis Queue.

Asynchronous Execution: Celery Workers pull the task from Redis. This async design prevents the web server from blocking.

Generation: The system uses LangChain and Prompt Engineering to communicate with LLMs. It generates 9 types of multimodal resources, including Audio Podcasts, Videos, and HTML (Quiz, Mental Map, Data Table, etc..).

Closed-Loop Integration: After generation, the system saves files to the storage volume and updates the database URLs automatically.

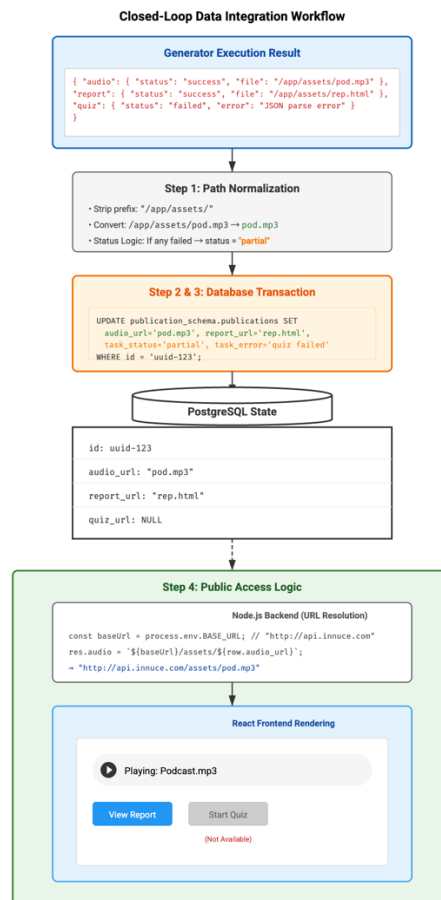


Figure 2 The Closed-Loop Data Integration Workflow

4. Results

The system was successfully tested. The engineering results demonstrate both functionality and efficiency.

4.1 Generated Outcomes

The pipeline successfully converts text-based PDFs into interactive formats. For example:

Audio Podcast: Creates a nature dialogue script between a “Host” and a “Guest”, making complex theories easy to listeners.

Interactive Quiz: Generates structured JSON data. The frontend renders this into a quiz with "Explanation" feedback, which has high pedagogical value.

Visual Report: Produces a responsive HTML summary, eliminating the need for users to read the original two-column PDF.

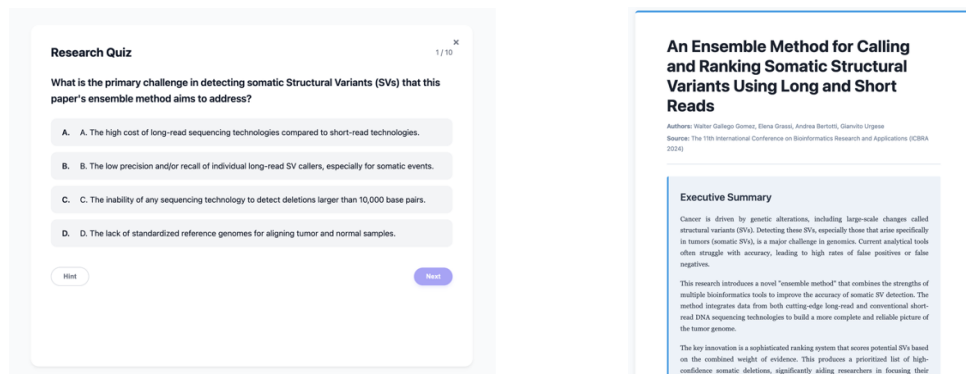


Figure 3: Examples of Generated HTML Report and Interactive Quiz

4.2 Performance Analysis

It compared the time consumption. Manual creation of these resources typically takes 3-4 hours per paper. By Notebook LM takes 5-10 minutes per paper. The inNuCE automated pipeline finishes the whole process in approximately 180 seconds. This validates the efficiency of the architecture.

5. Conclusions

This thesis presented a robust solution for modern scientific dissemination. By combining web infrastructure with Generative AI, it achieved two main goals: lowering the barrier for knowledge understanding and reducing the maintenance cost for administrators.

However, there are technical limitations. First, the stability of LLM output is not fully guaranteed, as the model occasionally generates invalid JSON formats or hallucination. Second, the PDF parsing library struggles with complex multi-column layouts. Finally, the system relies heavily on third-party APIs (Google Gemini), which introduces external risks. Future maintenance will focus on improving error handling to mitigate these issues.

In summary, the inNuCE platform offers a concrete reference for applying Generative AI in web applications. It successfully bridges the gap between complex academic research and the general public.