Student Internship Programme (SIP)

Final Project Report

at

Bitopia Technology Pte Ltd

Reporting Period:

May 2025 to Aug 2025

by

Koh Rui Cheng, Brendan

Department of Computer Science

School of Computing

National University of Singapore

2024/2025

Student Internship Programme (SIP)

Final Project Report

at

Bitopia Technology Pte Ltd

Reporting Period:

May 2025 to Aug 2025

by

Koh Rui Cheng, Brendan

Department of Computer Science

School of Computing

National University of Singapore

2024/2025

Project Title: Software Engineer/Developer

Project ID: SY242256988

Project Supervisor: Roy Chew

Summary

An asynchronous image processing workflow leveraging AI and RabbitMQ was developed as a proof-of-concept in an enterprise application. The system included a reusable RabbitMQ reliability architecture, involving retries, DLQs, and Grafana observability. These safeguards ensured fault-tolerance and recoverability of AI processing tasks. It also involved a reusable Spring Boot AI microservice, abstracting three LLM platforms and OCR for document and image processing. Additional research explored traditional ML models for enterprise application uses. The components and documentation developed form a blueprint for scalable, production-ready AI workflows in future enterprise systems. Architectures and implementation details are presented in this report.

Subject Descriptors:

Message queues

Reliability

Enterprise architectures

Artificial intelligence

Machine learning

Keywords:

RabbitMQ, large language models (LLM), asynchronous task processing, fault tolerance, AI image and document processing

Implementation Software and Hardware:

Spring Boot, RabbitMQ, Spring AMQP, Prometheus + Grafana, OpenAI/Google/Anthropic APIs, Docker, Thymeleaf, Vue, Tesseract OCR

Acknowledgement

I would like to express sincere gratitude to Bitopia Technology for the opportunity for me to undertake this internship, and to my supervisor Roy for their constant attention and guidance to me over the past 11 weeks. I would also like to thank my other coworkers for their advice and help, not only during our collaboration on project tasks, but also for making my time enjoyable and engaging through lunch outings and other activities. My coworkers' and supervisor's assistance and encouragement have been insightful and made my time in Bitopia Technology much more meaningful and enriching.

Table of Contents

1 Introduction

1.1 Host Organisation Background and Structure

Bitopia Technology is a company specialising in building full-stack business applications. It has both private- and public-sector clients, with experience building enterprise software in various industries, including real estate and e-commerce. The company uses a traditional Java enterprise stack with a Thymeleaf + Vue frontend and Spring Boot backend.

Bitopia Technology contains 7 full-time employees: 2 founders responsible for sales and project procurement, and 5 software engineers focused on Java enterprise development. Being part of a lean team, I assimilated into the company culture easily and was given opportunities to collaborate closely with many coworkers.

1.2 Project Background

I worked on two projects during my internship. I spent most of my time on a project for a client in the real estate industry. I developed a proof-of-concept image processing feature powered by artificial intelligence (AI) and RabbitMQ, where images are asynchronously processed into JavaScript Object Notation (JSON) data for generating business models. In the last two weeks of my internship, I also worked on a small AI document processing feature.

I also supported a public sector project where I conducted exploratory research into business use cases of AI like regression and classification. As a government-affiliated project, I considered constraints like privacy of sensitive citizen data, confining my research to free and open-source models.

2 Key Project Deliverables

2.1 AI and Document Processing Microservice Development

During the first two weeks, I began with an exploratory AI research task involving cost-benefit analysis of different large language models (LLMs) from companies like OpenAI, Google (Gemini), and Anthropic (Claude).

This involved building an experimental reusable AI software development kit (SDK) microservice in Spring Boot, providing an application programming interface (API) for users to create AI responses, then view the number of tokens and cost to call each AI model. I also created a corresponding Vue frontend,

giving users a graphical user interface (GUI) to test different AI models.

This application was deployed onto the enterprise project's user acceptance testing (UAT) server. It also provided valuable experimental data that I used to write a document to evaluate the cost-effectiveness of different LLMs for the company's products.

For the document processing task in the last two weeks of my internship, I expanded the microservice with a workflow with the Tesseract optical character recognition (OCR) library to get document text, which is then fed to the LLM to extract structured data from the document.

2.2 RabbitMQ Reliability and Fault-Tolerance Architecture

To enable asynchronous task execution for the AI image processing feature, my supervisor and I chose to implement a message queue (MQ) architecture. Since my supervisor intended to use MQs for future baseline projects, I designed a reusable MQ reliability and fault-tolerance architecture, as illustrated in Figure 1. We chose RabbitMQ due to its seamless Spring Boot integration via the Spring AMQP library. Additionally, unlike other message brokers like Kafka, RabbitMQ provides unique features vital for asynchronous task processing such as a management user interface (UI) and message time-to-lives (TTL), simplifying the architecture significantly (Dobbelaere & Esmaili, 2017).

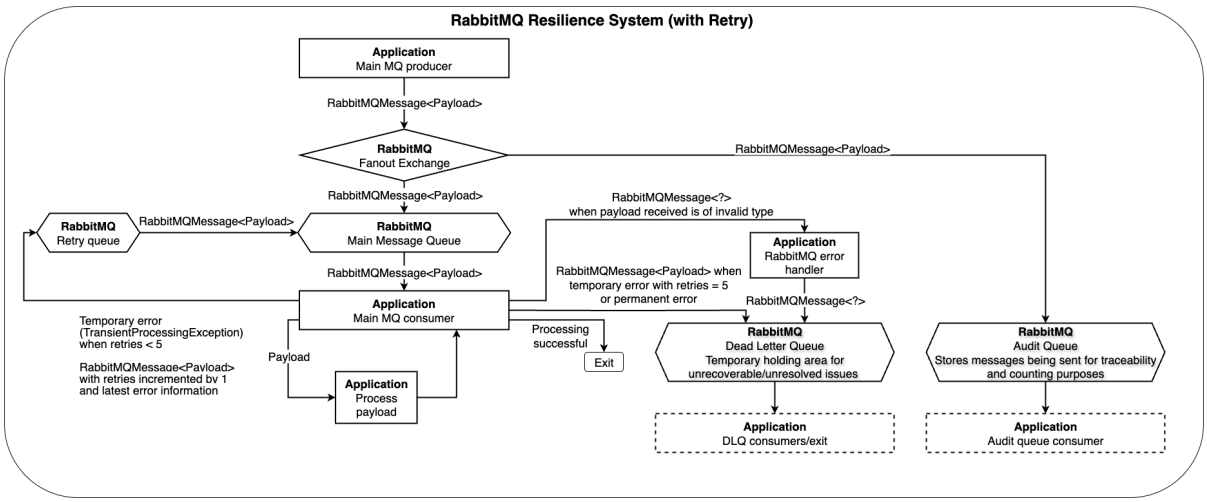


Figure 1: RabbitMQ Resilience System with Retry Logic

The RabbitMQ resilience architecture is designed to enable reliable asynchronous task processing. A `RabbitMQMessage` containing a payload and metadata like error details and retry count is first published by a producer, and then routed to both the main MQ for processing and an audit MQ for traceability and

monitoring.

A consumer subscribes to the main MQ and processes each message's payload. If processing is successful, the MQ workflow exits. However, if there are temporary processing errors like network issues, the message is sent to a retry queue that re-publishes the message into the main queue after a specified TTL like 30 seconds. This retry queue is processed independently of consumer threads, ensuring they remain unblocked to process other messages.

If there are permanent errors (e.g. invalid payload type/ID), or temporary errors after 5 retries, the messages are sent to a dead letter queue (DLQ), a temporary holding area for problematic messages, for further manual review.

The Firehose Tracer plugin was added to the RabbitMQ management UI, enabling full traceability and debugging of message flows through logging every message received or redirected from queues. Additionally, the RabbitMQ Prometheus plugin enabled exporting and tracking of metrics like individual queue size and number of messages in the past hour. Grafana dashboards provided visualisations of Prometheus metrics data, with RabbitMQ providing official Grafana dashboards for observability in most production applications.

This architecture and observability research was the result of iterative improvement over several standups. During the first few iterations, my supervisor noticed certain imperfect architectural decisions, such as the lack of traceability and observability, or incorrectly using DLQs within the retry workflow. These issues were progressively addressed in successive standups, resulting in the final resilience system in Figure 1.

2.3 Proof-of-Concept: Asynchronous AI Image Processing with RabbitMQ

The AI microservice and RabbitMQ reliability architecture were integrated into an asynchronous AI image processing workflow in the private sector application as a proof-of-concept feature branch. This feature allows users to upload an image, from which the LLM will extract visual features such as material type (e.g. floor or surface material) that are then integrated into the data model and persisted in the application database.

The extraction of visual features is done as an API call to the AI microservice. The entire workflow is asynchronous, where the user uploads an image and processing is done automatically in the background, even if the user leaves the page. This behaviour is enabled through the RabbitMQ architecture integration,

where the image processing tasks (encoded by image ID) are published to a queue that the consumer, which does the actual image processing, subscribes to. The image processing tasks are traceable and fault-tolerant due to the architecture design in subsection 2.2.

I also explored different computer vision (CV) models and libraries like OpenCV and evaluated the viability of local CV models compared to LLMs, since these models can solve image segmentation and classification tasks, like in our use case. Although we did not pursue OpenCV further, this research gave the company a lot of insight into the future uses of other available AI technologies.

2.4 Machine Learning (ML) Research for Enterprise Use Cases

During the second half of the internship, I was tasked with ML research for the public sector project, where I identified four common business applications of tabular data prediction: regression, classification, time-series prediction, and clustering. I also provided solutions and models used, and showed examples of these problems within the application context.

An example business application of classification is determining the type of pet (cat/dog) adopted based on the adopter's age and monthly income, as seen in Figure 2. A common classification model is **logistic regression**, which predicts that a 29-year-old with $4,100 monthly income would likely own a **dog**.

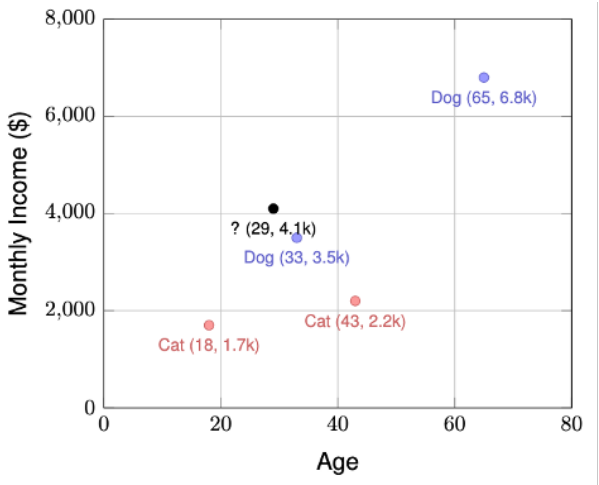


Figure 2: Business application of classification

As this research pertained to a government project, my research focused on the use of local open-source ML models from the `scikit-learn` and `statsmodels` libraries.

Since most ML libraries are Python-based and the company uses a Java Spring Boot technical stack, I

also designed a software architecture for running the ML models as a Python API callable by the Spring Boot application, as well as integration with business intelligence tools like PowerBI, as seen in Figure 3. This provides a template for my coworkers when implementing ML pipelines in the future.

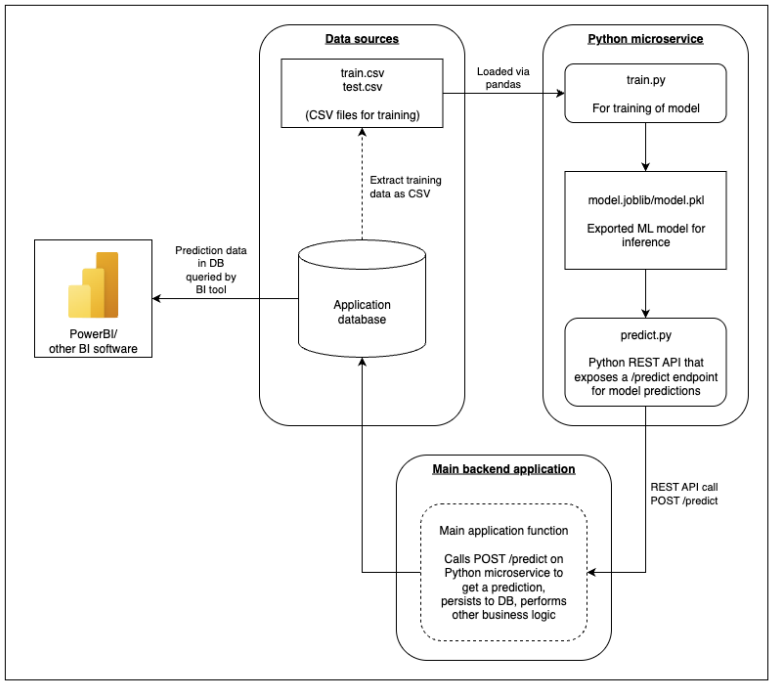


Figure 3: Enterprise and business intelligence integration of ML models

3 Knowledge and Experience Gained

3.1 Technical and Industry Knowledge Gained

During this internship, I gained first-hand exposure to various technologies and frameworks essential to my growth when developing future software applications.

In my first two weeks, I created a full-stack internal application with a reusable AI SDK with Vue and Spring Boot, giving me initial exposure to the company stack. Integrating the microservice with the enterprise codebase for the proof-of-concept was extremely simple, giving me valuable insight into the benefits of reusable microservice architecture and the separation of concerns it provides.

While I have had experience with general MQ architectures in school modules, I reinforced my technical knowledge by designing the reusable RabbitMQ reliability architecture and implementing it within the Java enterprise application. This feature gave me a valuable opportunity to implement RabbitMQ for the first time, while sharpening my architectural thinking skills through the design of architecture diagrams.

I also gained a deep appreciation of the importance of reliability and traceability of messages in enterprise applications, as dropping one message could create side effects like lost business revenue. Using Docker Compose to manage microservices like RabbitMQ, Prometheus, and Grafana helped me better understand the Docker ecosystem and microservice orchestration in development.

Over the past 11 weeks, I have been exposed to many new or unfamiliar technologies, including Vue, Spring Boot, Grafana, and Tesseract. This practical experience has expanded my technical toolkit, which I can apply for future personal and professional projects. I have also written multiple research and technical documents with architecture diagrams, simplifying explanations of these new technologies and concepts during standups. I also documented code through JavaDoc comments and READMEs, facilitating maintainability and easy handover for other developers. This hands-on experience allowed me to appreciate the importance of writing technical documentation and sharing knowledge with other developers, which improves code understandability and readability.

3.2 Non-technical Knowledge Gained

While the technical knowledge gained during the internship was extremely valuable for my future roles, most personal growth was rooted in the non-technical aspects of the internship, where I gained important soft skills like presentation skills and resilience.

During the internship, I engaged in weekly in-person standup meetings, where I shared with my supervisor and coworkers about my progress and findings from my assigned research tasks for the week. These standups required me to hone my presentation and communication skills by providing system walkthroughs and explanations of components I built, which sometimes exceeded an hour. Additionally, the supervisor occasionally asked tough questions regarding my architectural decisions, training my resilience and critical thinking skills.

The internship also taught me to be more proactive and adaptable. As my assigned tasks fell in later weeks as project deliverables approached completion, I scoped out new improvements and worked on vital documentation tasks during this downtime. Due to my more autonomous office environment, I also took the initiative to message my supervisor to offer additional assistance.

Overall, the past 11 weeks at Bitopia Technology have transformed me into a confident software engineer adaptable to different team structures. As I move into future roles, I believe that soft skills like effective

communication and resilience are as vital as technical expertise in steering the direction of my future projects.

4 Conclusions

4.1 Problems Faced

Throughout the internship, I encountered challenges that taught me valuable technical and non-technical lessons.

A challenge I encountered was to translate high-level guidance from my supervisor into detailed and actionable tasks. To meet the overall goals, I learned to proactively scope out and structure the low-level tasks on my own. I appreciated the trust and autonomy I was given, especially as an intern where self-directed learning is key.

I also faced challenges setting up tools like RabbitMQ and Tesseract OCR, as well as running the enterprise Java project for the first time. This involved troubleshooting and debugging setup errors and configuring the local development environment with minimal documentation and guidance. However, these experiences helped sharpen my knowledge of environment setup and debugging.

4.2 Assessment of Work Experience and Concluding Remarks

In conclusion, my work experience at Bitopia Technology has been transformative and enriching. I contributed to customer-facing features in production codebases, while having the autonomy to experiment with various AI and ML technologies. I am confident that I will benefit from these experiences in my remaining semesters in school. I also had the rare chance to design and implement backend systems like the RabbitMQ reliability architecture from the ground up. I have grown significantly from this opportunity and am grateful to my supervisor Roy and my colleagues for imparting valuable technical and professional skills during my past 11 weeks here.

References

Dobbelaere, P., & Esmaili, K. S. (2017). Kafka versus rabbitmq: A comparative study of two industry reference publish/subscribe implementations: Industry paper. *Proceedings of the 11th ACM International Conference on Distributed and Event-Based Systems*, 227–238. https://doi.org/10.1145/3093742.3093908

A Internship Presentation Slides

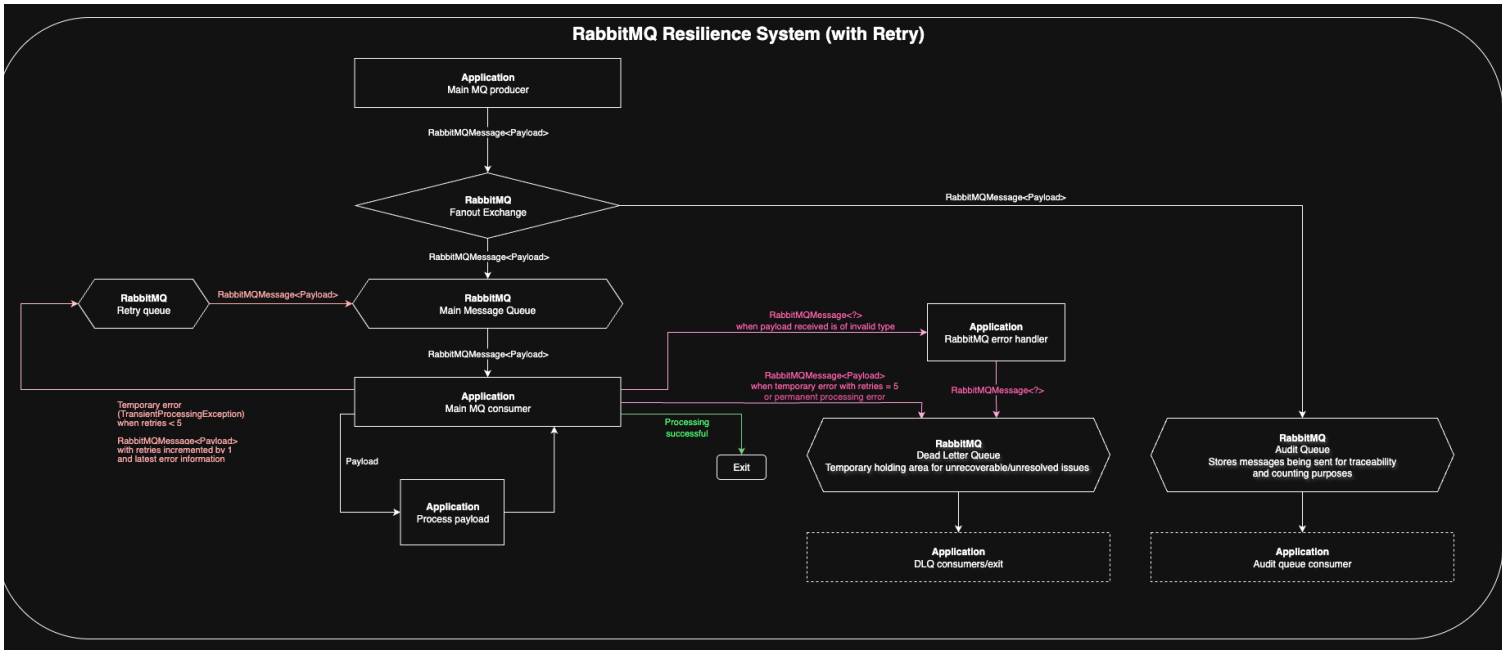
SIP/ATAP

Name: Koh Rui Cheng, Brendan
Project Title: Software Engineer/Developer
Project ID: SY242256988
Company: Bitopia Technology Pte Ltd
Supervisor: Roy Chew
Period: May-Aug 2025

Work Completed Milestones

- **Week 1-2:** AI model testing platform and microservice with support for multiple AI models from different firms
- **Week 3-4:** Initial proof-of-concept feature branch, implementing an AI image processing workflow
- **Week 5-10:** Reliable and reusable message queue architecture design (RabbitMQ)

Work Completed Milestones

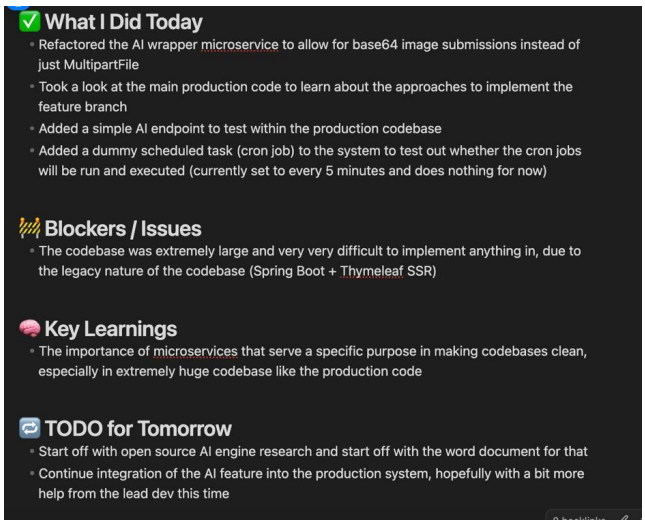


Work Completed Milestones

- **Week 5-11:** Integration of reusable RabbitMQ design into the image processing workflow
- **Week 7-9:** Research document on traditional machine learning (ML) problems for business use cases
- **Week 10-11:** Small document processing feature using Tesseract OCR and AI microservice

Technical Skills

- Scoped tasks tracked on notes app daily throughout the internship



Technical Skills

- Hands-on experience to various technologies, including Vue, Spring Boot, RabbitMQ, Grafana, Tesseract OCR
- Understanding of reliability and traceability in production systems
- Gained knowledge on the benefits of microservice architecture, and separation of concerns it provides

Soft Skills

- Wrote a lot of technical and other documentation throughout the internship, easing the supervisor's and coworkers' understanding of various concepts I researched on
- Gained valuable presentation, communication, and critical thinking skills from the weekly standups

Conclusion

- Career path & goals
  - Definitely widened as my expertise has grown for backend-heavy and full-stack roles
  - **Pre-internship:** Was a frontend heavy developer with minimal backend exposure
  - **Post-internship:** Realizing that I am quite capable to do backend and infrastructure heavy roles

Conclusion

- Professional network
  - Shared my GitHub profile during farewell lunch, a few developers followed me there
- Challenges faced
  - Independent-oriented work culture was difficult for me at first
  - Feature branch done independently from main system → had to fix bugs and merge conflicts when updating my feature branch