Generative AI for Software Architecture and AIOps

LLM Driven Performance Model

N. Shiah; M. Fokaefs

[1]Department of Electrical Engineering and Computer Science, Lassonde School of Engineering

LASSONDE SCHOOL OF ENGINEERING

YORK U

Abstract

This project focuses on automatically using real-time performance data from microservices to build accurate LLM-based performance models.

Our main focuses are:

- Enriches models with live CPU, memory, and network metrics.
- Fine-tuned GPT-3.5 Turbo generates Palladio Bench files
- Predicts system response time within ~15% error.
- Identifies hidden performance bottlenecks
- Enables faster, automated performance simulation

Introduction

Problem

- Microservices are complex and dynamic, making manual performance modeling difficult.
- Traditional static models become outdated with evolving workloads.

Challenge:

Traditional modeling requires expert input and lacks adaptability

Solution & Goal:

- AI-driven method using real-time metrics and LLMs
- Continuous, accurate, low-effort performance modeling

Materials & Methods

- **System Monitored:** Spring PetClinic (microservices demo app)
- **Monitoring Tools:** Prometheus (metrics) + Zipkin (tracing)
- **Data Collected:**

1. CPU usage per microservice
2. Memory usage over time
3. Network & HTTP response sizes

- **AI Process- Fine-tuned LLM Setup:**

1. Convert time-series metrics to text prompts (.json)
2. Fine-tune GPT-3.5 Turbo model
3. Generate syntactically 10 Palladio files [Figure 5]
4. Match structures from BookShop case study

- **Other LLMs tested:** GPT-4, Claude (30–50% error range; GPT-3.5 gave best trade-off).

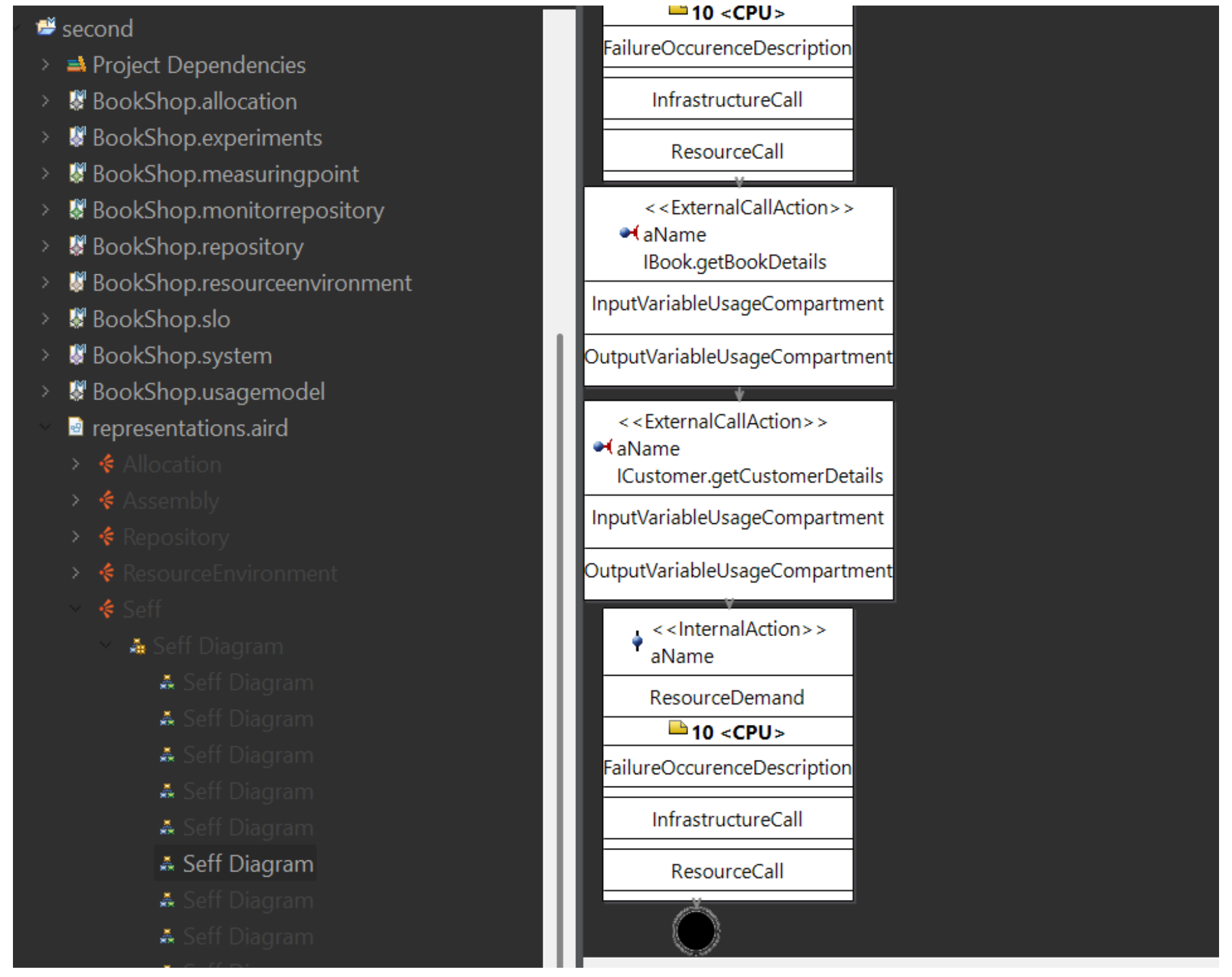

Figure 5: SEFF Diagram under BookShop Case

Results

Our results show promising fidelity and automation benefits:

- **Accuracy**: GPT-generated model predicted response times within 15% of measured results.
- **Automation**: Successfully generated importable 10 Palladio files.
- **Bottleneck Detection**: Automatically surfaced performance issues during simulation.
- **Service Dependencies**: Highlighted different CPU consumption patterns across services.
- **Model Comparison**: GPT-3.5 offered best balance of reliability, control, and latency.

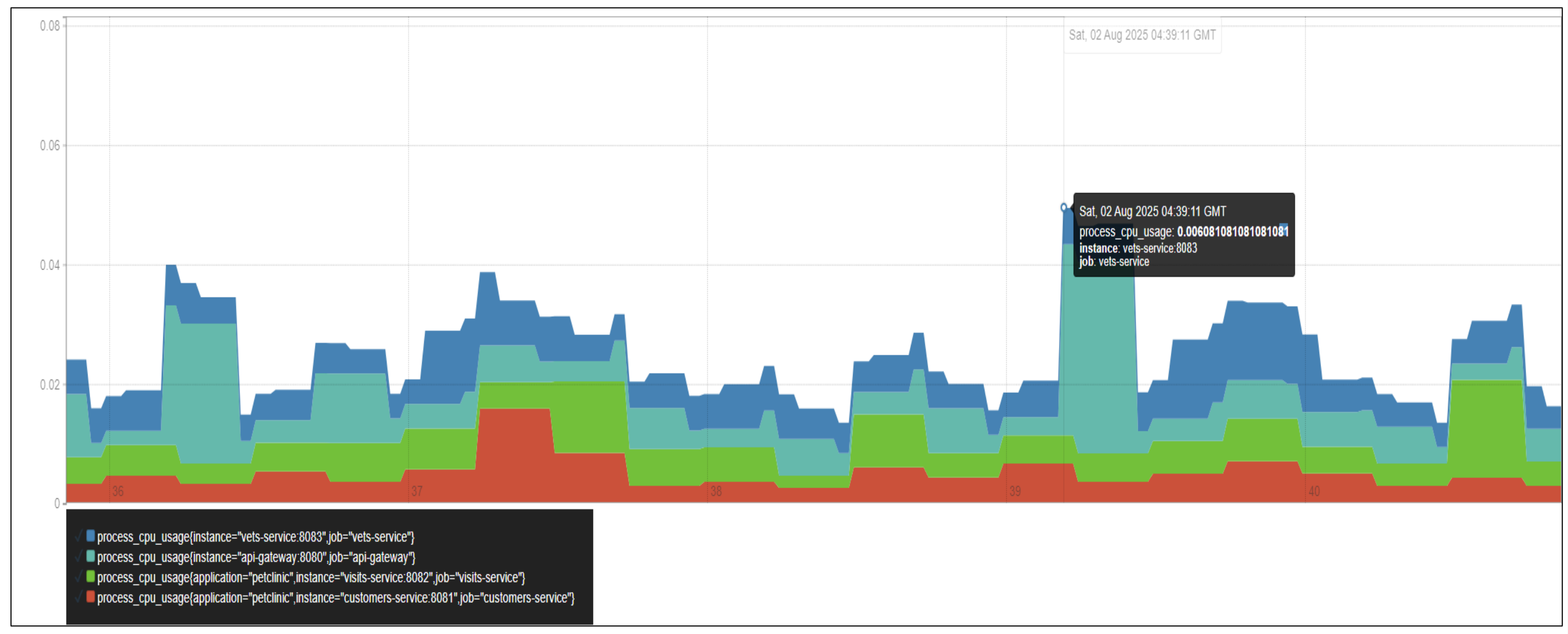

Figure 1: CPU usage per microservice

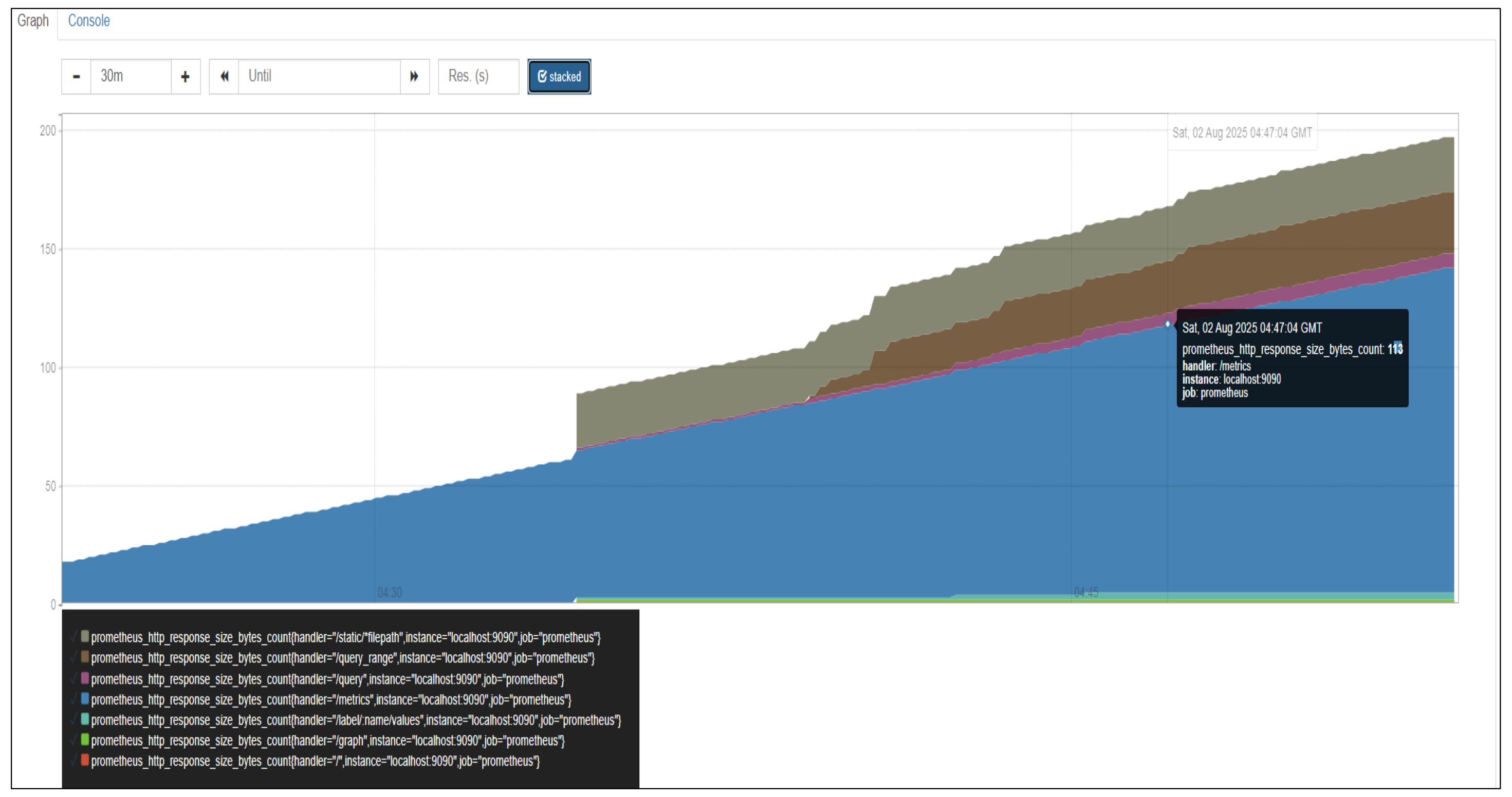

Figure 3: Network & HTTP Response sizes

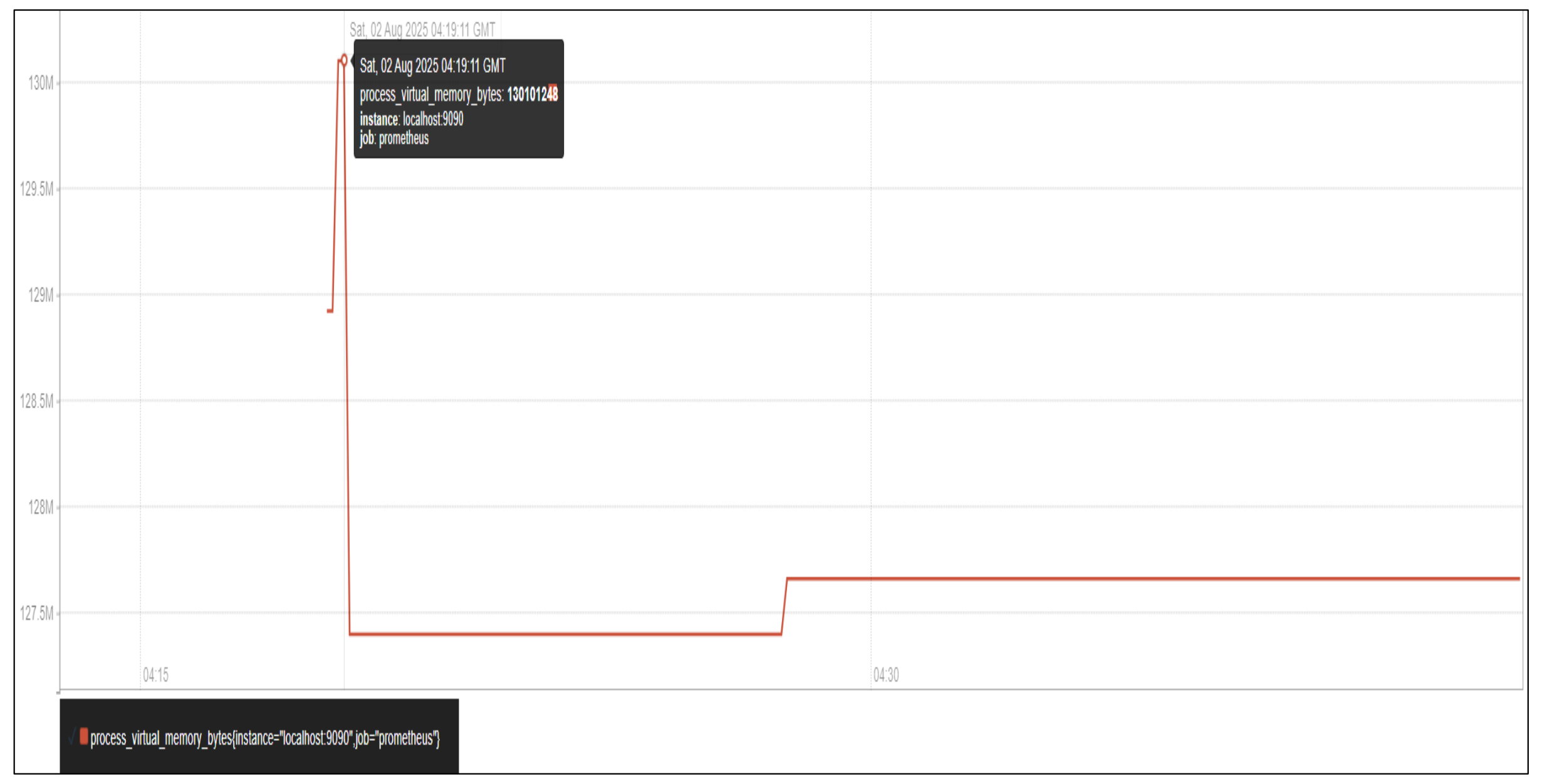

Figure 2: Memory Usage over time

We also tested generating the Palladio files using GPT-4, Claude, and other LLMs for comparison. While all models showed the ability to structure outputs correctly(within ~30%-~50% error bound), GPT-3.5 offered the best balance between reliability, control and latency during iterative fine-tuning.

Discussion & Implications

Benefits:

- Automated performance regression testing.
- "What-if" scenario simulation.
- Dynamic capacity planning.
- New DevOps practices enabled.

Limitations:

- Output not 100% reliable.
- Requires manual validation and adjustment.
- Component bindings occasionally misaligned.
- Interface-method associations sometimes missing.
- Complex scenarios need manual post-editing.

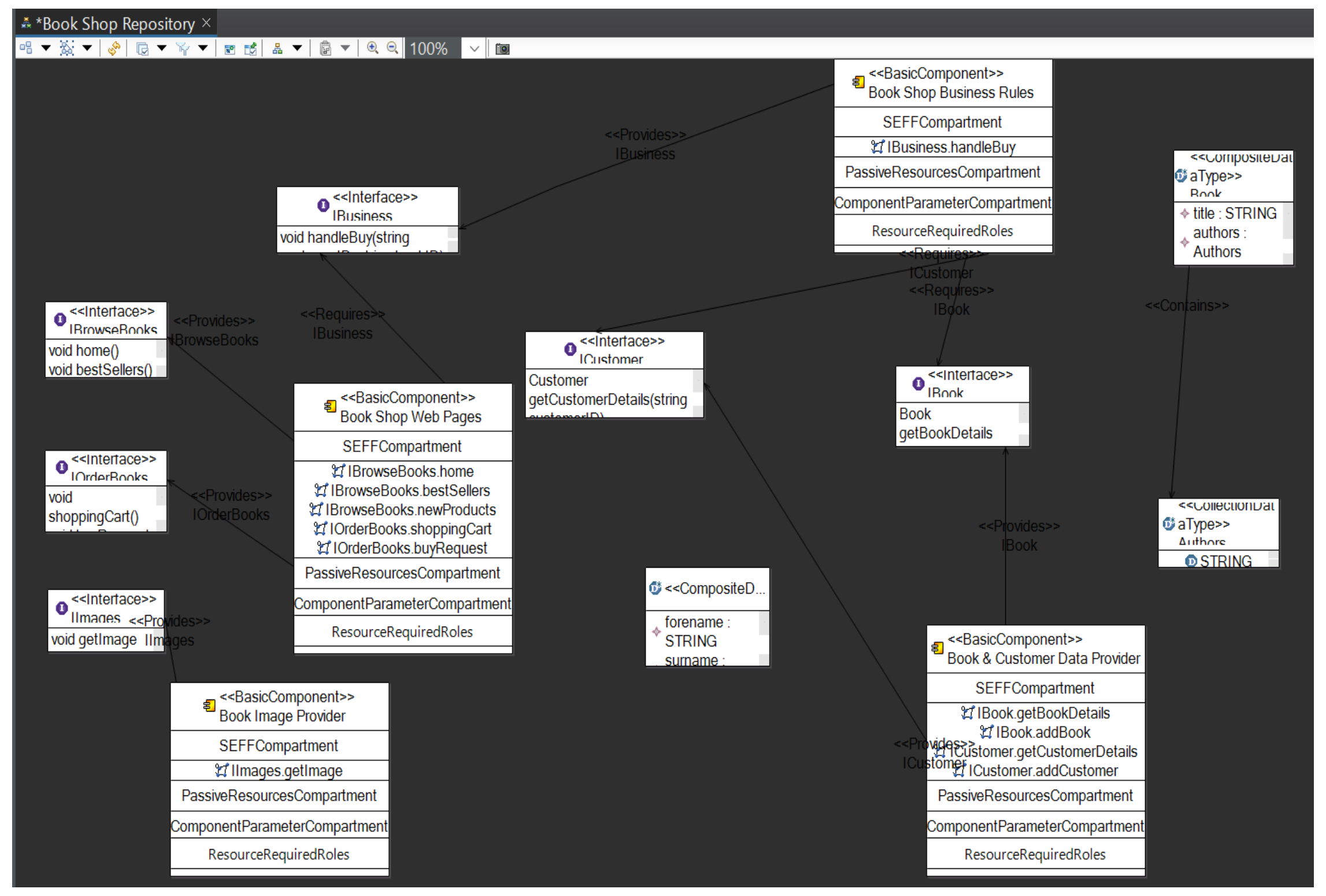

Figure 4: Generated repository model in BookShop Case

Conclusion

This work demonstrates that generative AI, grounded in live runtime data, can deliver fast, accurate, and low-maintenance performance models for microservices.

In summary, while GPT-based generation of Palladio files significantly reduces human effort in initial model setup, the output is not yet 100% reliable. Developers must still validate, adjust, and sometimes rewrite parts of the generated XML—especially in complex scenarios involving component interactions and nested behaviors.

References

Spring Team, "*spring-petclinic-microservices*," GitHub repository, https://github.com/spring-petclinic/spring-petclinic-microservices (accessed May. 10, 2025).

This project has been supervised under Professor Marios Fokaefs & supported under Lassonde Undergraduate Research Award (LURA).

Contact

Nurjahan A. Shiah

Software Engineering-Big Data Stream

Email: nurjahanahmed7@gmail.com

LinkedIn