Extended Abstract (M1 Summer Internship)

Student: Mohamed Mbarkiou

Company: Oracle R&D; Center, Casablanca

Dates: June 2025 – September 2025

Abstract

This extended abstract reports on a twelve-week internship conducted at the Oracle R&D; Center in Casablanca, Morocco. The primary objective was to address latency and scalability issues within the company's internal service architecture. My project focused on developing a centralized configuration microservice using Java and the Micronaut framework. The goal was to replace legacy configuration methods that caused deployment bottlenecks. Through the implementation of a horizontally scalable architecture, integrated with Oracle Cloud Key Management Service (KMS) and secured via TLS/JWT, the project achieved significant performance gains. Specifically, deployment times for consumer services were reduced from 10 seconds to under 100 milliseconds. This report details the technical development process, the security protocols implemented, and analyzes the professional feedback received. The experience confirmed my proficiency in backend engineering while highlighting the importance of cross-team communication in a large-scale enterprise environment.

Introduction

Oracle is a global leader in database software and cloud engineering. At the R&D; Center in Casablanca, the engineering teams maintain a complex ecosystem of microservices. A critical challenge in this architecture is "Configuration Management"—how individual services retrieve their settings (database URLs, feature flags, secrets) at runtime.

Prior to my internship, consumer services faced significant delays during startup because they relied on a legacy configuration fetch mechanism. This resulted in deployment times averaging 10 seconds per service, which is unacceptable for a cloud-native environment requiring rapid auto-scaling. The objective of my internship was to engineer a modern, high-performance solution to decentralize and optimize this process.

The project aimed to build a new microservice capable of serving configurations with low latency (<50ms) while ensuring strict security compliance. This assignment was not merely a proof-of-concept; the target was a production-ready tool to be adopted by over ten internal services. This project required a deep understanding of distributed systems, encryption standards, and the specific capabilities of the Oracle Cloud Infrastructure (OCI).

Account of Work

The internship followed a standard Software Development Life Cycle (SDLC), moving from requirement analysis to deployment and integration.

1. Technology Selection and Architecture Design

The first phase involved selecting the right tools for high-performance requirements. While Spring Boot is standard, I opted for Micronaut for this specific service. Micronaut's ahead-of-time (AOT) compilation offers faster startup times and lower memory consumption, which fits the requirement for a low-latency utility service. I designed the architecture to be horizontally scalable, ensuring that as more internal teams adopted the service, we could simply add more instances without degrading performance.

2. Core Development and Optimization

I developed the core logic in Java. The primary challenge was reducing the "Time-to-First-Byte." By optimizing the data retrieval logic and implementing efficient caching strategies, I successfully reduced the configuration fetch time for consumer services from 10 seconds to under 100 milliseconds. To ensure the system could handle high throughput, I engineered the service to maintain a response time of less than 50ms, even under load.

3. Security Implementation (DevSecOps)

Given that this service handles sensitive configuration data, security was paramount. I integrated the service with Oracle Cloud KMS (Key Management Service) to ensure that all secrets were encrypted at rest and in transit. I implemented a robust authentication layer using JWT (JSON Web Tokens), requiring services to prove their identity before requesting configurations. Additionally, I enforced TLS-secured REST APIs to prevent man-in-the-middle attacks.

4. Deployment and Integration

The final phase involved containerizing the application and deploying it to the staging and production environments. I collaborated with other teams to integrate this new microservice into their workflows. By the end of the internship, over 10 active services in production were consuming configurations from my microservice. I also utilized my DevOps skills to set up monitoring dashboards to track the latency metrics, confirming the <100ms deployment time achievement.

Feedback Analysis and Reflection

Supervisor Evaluation:

The feedback from my internship tutor at Oracle was largely positive, particularly regarding technical execution and autonomy. The evaluation highlighted my ability to quickly master the Micronaut framework and the internal Oracle Cloud infrastructure. My tutor appreciated that the solution was "production-ready," noting that the reduction in deployment latency directly impacted the team's velocity. However, the evaluation pointed out that while my code was efficient, my initial documentation was too technical and lacked high-level architectural diagrams for non-expert stakeholders.

Self-Evaluation:

I agree with the supervisor's assessment. I spent 90% of my energy on optimization and code quality, often treating documentation as an afterthought. I realized that in a large organization like

Oracle, code is read more often than it is written, and clear documentation is essential for maintainability.

Improvement Points:
Reflecting on this, I have identified two main areas for improvement for my future career and the upcoming Projet Long:

1. Documentation First: I aim to adopt a "Docs-as-Code" approach, ensuring that architectural diagrams and API documentation (Swagger/OpenAPI) are updated simultaneously with the code.
2. Stakeholder Communication: I need to improve how I communicate technical wins to non-technical managers. Instead of just sharing latency numbers, I should explain the business value (e.g., "saved X engineering hours").

Conclusion

In conclusion, my internship at Oracle was a definitive step in my transition from student to professional Software Engineer. I successfully engineered a critical piece of infrastructure that solved a tangible latency problem, reducing deployment times by over 99%. Mastering technologies like Micronaut, Oracle Cloud KMS, and JWT in a production environment significantly deepened my technical expertise. Furthermore, the feedback received regarding the importance of documentation has reshaped my approach to software engineering. I am now better prepared to tackle complex, distributed systems in my final year Master’s project and am eager to apply these enterprise-level standards to future challenges.

Bibliography

1. Oracle Cloud Infrastructure, "Key Management Service (KMS) Documentation," *docs.oracle.com*, 2025.

2. G. A. Graeme Rocher, *Learning Micronaut*, 1st ed., O'Reilly Media, 2024.

3. S. Newman, *Building Microservices: Designing Fine-Grained Systems*, 2nd ed., O'Reilly Media, 2021.

4. JSON Web Token (JWT) Internet Standard, "RFC 7519," *IETF Tools*, 2015.