الجمهورية الشعبية الديمقراطية الجزائرية
République Algérienne Démocratique et Populaire
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
المدرسة العليا للإعلام الآلي 08 ماي 1945، بسيدي بلعباس
École Supérieure en Informatique
-08 Mai 1945- Sidi Bel Abbès

Mémoire de Fin d'étude

En Vue de l'obtention du diplôme d' **ingénieur d'état**
Filière : **Informatique**
Spécialité : **Système d'Information et Web (SIW)**

Thème

Migration from monolithic to micro-services architecture approach for Ruby on Rails web applications

---

Réalisé par :
Abdelouadoud MAHDAOUI

Soutenu le : **17/09/2023**

Devant le jury composé de :

Mr Abdelhamid MALKI	Président
Mr Djamel-Abdelhak SERIAI (LIRMM)	Encadrant
Mr Amar Djamel BENSABER (ESI-SBA)	Encadrant
Mme Badia KLOUCHE	Examinatrice

Année Universitaire : 2022/2023

Dedication

“

I dedicate this work to my dear parents, for their unconditional love, unfailing support and the values they have instilled in me throughout my life.

To my brothers El mouatassim billah and Anes and my sister Khaoula, your unwavering support and companionship mean the world to me. Thank you for always being there.

To my friends, supervisors and all those who have shared their time, expertise and encouragement, I express my sincere gratitude. Your encouragement and invaluable advice have helped shape this work.

All my loved ones, to all of you

Thank you.

”

- Abdelouadoud

Acknowledgements

I would like to express my sincere gratitude to my supervisors **Mr. Djamel-Abdelhak Seriai** from LIRMM and **Mr. Djamel Amar Bensaber** from ESI-SBA who have guided and supported me throughout this final year project. I would like to thank them for their attentive supervision, insightful advice, and invaluable expertise that greatly contributed to the completion of this work.

My thanks also go to **all the teachers and staff members at ESI-SBA.** Your dedication, pedagogical support, and encouragement have been crucial to my education and the accomplishment of my journey at ESI-SBA.

I would like to extend my heartfelt gratitude to **my friends and all those who contributed to this work.** Your collaborative spirit has been a source of inspiration.

Lastly, I wish to convey my deep gratitude to **my parents, as well as my brothers and sister.** Your unconditional love, sacrifices, and unwavering support have been my pillars throughout this academic journey.

Abstract

In today's rapidly evolving technological landscape, software applications are becoming increasingly intricate, posing challenges in their development, deployment, and maintenance. Microservice-oriented architecture (MSA) is one of the latest style to have emerged. An approach that involves breaking down applications into smaller, independent services that collaborate seamlessly, promoting flexibility, scalability, and maintainability. These features allow for easily-deployable, autonomous, and scal- able applications.

While the advantages of microservices are evident, numerous enterprises grapple with legacy monolithic codebases. These legacy systems, characterized by their sprawling and intricate code, often hinder the adoption of microservices due to their inherent complexity. Redesigning from the ground up is cost-prohibitive and fraught with risks, prompting organizations to embark on gradual, incremental migrations toward microservices. In light of these challenges, there emerges a pressing need for a well-structured, versatile, and reusable migration process.

In this project, we focus specifically on the migration of applications built using Ruby on Rails. The migration process is dissected into two primary research problems: Firstly, the identification of microservices architecture or clusters within the existing monolithic codebase. Secondly, the materialization of the extracted microservices architecture utilizing diverse technologies and techniques.

To validate the proposed approach, a real-world, in-production application serves as the testing ground. Through meticulous analysis, we identified microservices and executed the migration process. Subsequent to the migration, rigorous testing and experimentation were conducted to assess the quality of the migrated application and discern disparities from its monolithic precursor.

Résumé

Dans le paysage technologique actuel, qui évolue rapidement, les applications logicielles deviennent de plus en plus complexes, ce qui pose des problèmes au niveau de leur développement, de leur déploiement et de leur maintenance. L'architecture orientée microservices (MSA) est l'un des derniers styles à avoir émergé. Cette approche consiste à décomposer les applications en petits services indépendants qui collaborent de manière transparente, ce qui favorise la flexibilité, l'évolutivité et la facilité de maintenance. Ces caractéristiques permettent de créer des applications facilement déployables, autonomes et évolutives.

Bien que les avantages des microservices soient évidents, de nombreuses entreprises sont aux prises avec des bases de code monolithiques héritées du passé. Ces systèmes hérités, caractérisés par leur code tentaculaire et complexe, entravent souvent l'adoption des microservices en raison de leur complexité inhérente. Une refonte complète est trop coûteuse et comporte des risques, ce qui incite les organisations à se lancer dans des migrations graduelles et incrémentielles vers les microservices. À la lumière de ces défis, il apparaît un besoin pressant d'un processus de migration bien structuré, polyvalent et réutilisable.

Dans ce projet, nous nous concentrons spécifiquement sur la migration d'applications construites à l'aide de Ruby on Rails. Le processus de migration est disséqué en deux problèmes de recherche principaux : Premièrement, l'identification d'une architecture de microservices ou de clusters au sein de la base de code monolithique existante. Deuxièmement, la matérialisation de l'architecture microservices extraite à l'aide de diverses technologies et techniques.

Pour valider l'approche proposée, une application réelle en production sert de terrain d'essai. Grâce à une analyse méticuleuse, nous avons identifié les microservices et exécuté le processus de migration. Après la migration, des tests et des expérimentations rigoureux ont été menés pour évaluer la qualité de l'application migrée et discerner les disparités par rapport à son précurseur monolithique.

ملخص

في المشهد التكنولوجي سريع التطور اليوم ، أصبحت تطبيقات البرامج معقدة بشكل متزايد ، مما يشكل تحديات في تطويرها ونشرها وصيانتها. تعد البنية الموجهة نحو الخدمات المصغرة (MSA) واحدة من أحدث الأساليب التي ظهرت. نهج يتضمن تقسيم التطبيقات إلى خدمات أصغر ومستقلة تتعاون بسلاسة ، مما يعزز المرونة وقابلية التوسع وقابلية الصيانة. تسمح هذه الميزات بتطبيقات سهلة النشر ومستقلة وقابلة للتطوير.

في حين أن مزايا الخدمات المصغرة واضحة ، فإن العديد من الشركات تتصارع مع قواعد التعليمات البرمجية القديمة. غالبا ما تعيق هذه الأنظمة القديمة ، التي تتميز برمزها المترامي الأطراف والمعقد ، اعتماد الخدمات المصغرة بسبب تعقيدها المتأصل. إن إعادة التصميم من الألف إلى الياء باهظة التكلفة ومحفوفة بالمخاطر ، مما يدفع المؤسسات إلى الشروع في عمليات ترحيل تدريجية وتدريجية نحو الخدمات المصغرة. في ضوء هذه التحديات ، تبرز حاجة ملحة لعملية ترحيل شبه آلية ومتعددة الاستخدامات وقابلة لإعادة الاستخدام.

في هذا المشروع ، نركز بشكل خاص على ترحيل تطبيقات Ruby on Rails حيث تم تقسيم عملية الترحيل إلى مشكلتين بحثيتين أساسيتين: أولاً، تحديد بنية الخدمات الصغيرة أو المجموعات داخل قاعدة التعليمات البرمجية المتجانسة الحالية. ثانياً، تجسيد بنية الخدمات المصغرة المستخرجة باستخدام التقنيات التكنولوجيا المختلفة.

للتحقق من صحة النهج المقترح ، إستخدمنا تطبيقا واقعي في طور الإستخدام كأرض للاختبار. من خلال التحليل الدقيق ، حددنا الخدمات المصغرة ونفذنا عملية الترحيل. بعد الترحيل ، تم إجراء اختبارات وتجارب صارمة لتقييم جودة التطبيق الجديد وتمييز التباينات من سلائفها.

Liste of acronyms

MSA	*Microservice-oriented architecture*
MIA	*Microservice Identification Approach*
API	*Application Programming Interface*
HTTP	*Hypertext Transfer Protocol*
JSON	*JavaScript Object Notation*
REST	*Representational State Transfer*
AST	*Abstract Syntax Tree*
RoR	*Ruby on Rails*
CLI	*Command Line Interface*
MVC	*Model-View-Controller*
URL	*Uniform Resource Locator*
DevOps	*Development and Operations*
XML	*Extensible Markup Language*
CBSE	*Component-Based Software Engineering*
RPC	*Remote Procedure Call*
BATC	*Business Applications Transactional Contexts*
DRY	*Don't Repeat Yourself Principle*