

LB/TH/43/2025
TH6020

MODELING A GLOBALLY SCALABLE AND COMPLIANT MULTI-TENANT SAAS ARCHITECTURE

Mohamed Jaward Uzama Zaid

239379U

MSc in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

June 2025

MODELING A GLOBALLY SCALABLE AND COMPLIANT MULTI-TENANT SAAS ARCHITECTURE

Mohamed Jaward Uzama Zaid

239379U

Thesis submitted in partial fulfilment of the requirements for the degree
MSc in Computer Science

Department of Computer Science and Engineering
Faculty of Engineering

University of Moratuwa
Sri Lanka

June 2025

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other University or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature:

Date: 06/06/2025

The above candidate has carried out research for the Masters thesis under my supervision. I confirm that the declaration made above by the student is true and correct.

Name of Supervisor: Dr. Buddhika Karunaratne

Signature of the Supervisor:

Date: 22/06/2025

ACKNOWLEDGMENT

First and foremost, I would like to express my deepest gratitude to my supervisor, Dr. Buddika Karunaratne, for his valuable guidance, patience, and support throughout this research project. His expertise and constructive feedback were instrumental in shaping this work. I am equally grateful to Dr. Sapumal Ahangama for his insightful comments and suggestions during my PG Diploma viva examination. I would also like to extend my sincere thanks to my final viva examiners, Dr. Thanuja Ambegoda and Dr. Thilinat Thanthriwatta, for their valuable feedback on the final thesis.

My sincere appreciation goes to all the lecturers and administrative staff members of the Department of Computer Science and Engineering, University of Moratuwa, who contributed to organizing the MSc program and facilitating this research journey. Your dedication and support created an environment conducive to academic growth and research excellence. I would especially like to acknowledge Mrs. Nisansala Samarasooriya for supporting administrative works.

To my beloved wife, words cannot adequately express my appreciation for your unwavering support during both the challenging and rewarding moments of this research journey. You have been my pillar of strength throughout this endeavor. I am also profoundly grateful to all my family members for their constant encouragement and understanding during this intensive period of study.

Finally, I would like to acknowledge my friends and peer colleagues who provided moral support and stimulating discussions throughout this academic pursuit. Your companionship made this journey more meaningful and enjoyable.

ABSTRACT

The rapid growth of cloud computing has established Software as a Service (SaaS) as the dominant software delivery model. The increasing adoption of SaaS creates opportunities for SaaS providers to expand globally and capture emerging markets. However, global expansion introduces significant challenges in balancing global scalability, data residency compliance, and operational efficiency. Existing research often addresses these requirements in isolation, resulting in suboptimal solutions.

This research proposes and validates a novel SaaS architecture that reconciles these competing requirements. The architecture comprises two key components: (1) a logically centralized yet physically distributed Control Plane, which enables operational efficiency through unified management of Application Planes and tenants, and (2) regionally distributed Application Planes, supporting global scalability while complying with regional data residency regulations.

The proposed solution (fully distributed nCPxnAP model) is evaluated against two alternative deployment models (single region 1CPx1AP and hybrid 1CPxnAP). Through comprehensive comparative analysis of three architectural variants, we demonstrate that our fully distributed nCPxnAP model achieves optimal performance, ensures full compliance with data residency laws, and preserves operational efficiency. The key contributions of this research are: (1) an architectural model for a globally scalable, compliant, and operationally efficient multi-tenant SaaS; (2) empirical validation via real world implementation and testing; and (3) practical guidelines for SaaS providers at various growth stages.

The findings offer a clear pathway for SaaS providers transitioning from single region deployments to global distribution while maintaining data residency compliance and operational efficiency. The research establishes new design principles for compliant global SaaS deployments and provides actionable implementation blueprints and performance benchmarks. It not only advances academic research but also provides industry professionals with a validated architectural model for navigating the complexities of modern global SaaS delivery.

Keywords: Cloud Computing, SaaS, Global Scalability, Data Residency Compliance, Operational Efficiency

TABLE OF CONTENTS

Declaration.....	i
Acknowledgment.....	ii
Abstract.....	iii
Table of contents.....	iv
List of Figures.....	vii
List of Tables.....	ix
List of Abbreviations.....	x
List of Appendices.....	xi
Chapter 1.....	1
Introduction.....	1
1.1 What is SaaS.....	2
1.2 What is Multi-Tenancy.....	3
1.3 Data Residency Compliance.....	4
Chapter 2.....	5
Research Problem.....	5
Chapter 3.....	6
Research Objectives.....	6
Chapter 4.....	7
Literature Review.....	7
4.1 SaaS and Multi-Tenancy.....	7
4.2 SaaS and Architectural Patterns.....	9
4.3 SaaS and Scalability.....	10
4.4 SaaS and Data Compliance.....	10
4.5 SaaS and Operational Efficiency.....	11
4.6 Research Gap.....	12
Chapter 5.....	13
Proposed Method.....	13
5.1 Proposed Architecture.....	13
5.1.1 The Logically Centralized, Physically Distributed Control Plane.....	14
5.1.2 The Distributed Application Plane.....	15
5.1.3 Pipeline.....	16
5.2 Tenant Scheduling and Routing.....	16
5.2.1 Tenant Scheduling.....	17
5.2.2 Tenant Routing.....	19
5.3 Addressing the Research problem.....	20
5.4 Evaluation.....	20
Chapter 6.....	21
Implementation Details.....	21
6.1 Tech Stack.....	21
6.2 Control Plane.....	23
6.2.1 Architecture.....	23

6.2.2 Infra Setup.....	24
6.3 Application Plane.....	25
6.3.1 Architecture.....	25
6.3.2 Infra Setup.....	26
Chapter 7.....	27
Experimental Results.....	27
7.1 Experimental Setup.....	27
7.1.1 Single Region (1CPx1AP).....	28
7.1.1.1 Architecture.....	28
7.1.1.2 Pre Configuration.....	28
7.1.1.3 Components Management.....	29
7.1.2 Hybrid (1CPxnAP).....	30
7.1.2.1 Architecture.....	30
7.1.2.2 Pre Configuration.....	31
7.1.2.3 Components Management.....	32
7.1.3 Fully Distributed (nCPxnAP).....	34
7.1.3.1 Architecture.....	34
7.1.3.2 Pre Configuration.....	34
7.1.3.3 Components Management.....	35
7.2 Test Cases.....	36
7.2.1 Business Register.....	36
7.2.2 Admin Login.....	36
7.2.3 Admin Get Business.....	37
7.2.4 Create Business User.....	38
7.2.5 Business User Login.....	38
7.2.6 Business User Get Business.....	39
7.3 Test Dataset.....	39
7.4 Testing Setup.....	40
7.5 Final Results.....	42
7.5.1 1CPx1AP.....	42
7.5.1.1 Component Metrics.....	42
7.5.1.2 Data Residency Law.....	43
7.5.1.3 Operational Efficiency.....	43
7.5.1.4 Final Latencies.....	44
7.5.1.5 Final Summary.....	46
7.5.2 1CPxnAP.....	47
7.5.2.1 Component Metrics.....	47
7.5.2.2 Data Residency Law.....	48
7.5.2.3 Operational Efficiency.....	49
7.5.2.4 Final Latencies.....	50
7.5.2.5 Final Summary.....	52
7.5.3 nCPxnAP.....	53
7.5.3.1 Component Metrics.....	53

7.5.3.2 Data Residency Law.....	54
7.5.3.3 Operational Efficiency.....	54
7.5.3.4 Final Latencies.....	55
7.5.3.5 Final Summary.....	56
7.5.4 Comparison.....	57
7.5.4.1 Performance Latencies.....	57
7.5.4.2 Data Residency Law.....	59
7.5.4.3 Operational Efficiency.....	59
7.5.4.4 Global Scalability.....	60
7.5.4.5 Cost.....	60
7.5.4.6 Availability.....	60
7.5.4.7 Summary.....	61
7.5.5 Findings and Recommendations.....	61
7.5.6 Limitations.....	62
Chapter 8	63
Conclusion.....	63
References.....	65

LIST OF FIGURES

Figure	Description	Page
Figure 5.1	Application Plane vs Control Plane	13
Figure 5.2	Proposed Architecture	14
Figure 5.3	Control Plane Data Model (ERD)	16
Figure 5.4	Tenant Registration Sequence Diagram	17
Figure 5.5	Tenant User Login Sequence Diagram	18
Figure 6.1	Single Control Plane Architecture	24
Figure 6.2	Multiple Control Plane Architecture	24
Figure 6.3	Application Plane Architecture	26
Figure 7.1	1CPx1AP Architecture	28
Figure 7.2	Control Plane Deploy Gitlab Pipeline (1CPx1AP)	29
Figure 7.3	Application Plane Deploy Gitlab Pipeline (1CPx1AP)	30
Figure 7.4	Application Plane Destroy Gitlab Pipeline (1CPx1AP)	30
Figure 7.5	Control Plane Destroy Gitlab Pipeline (1CPx1AP)	30
Figure 7.6	1CPxnAP Architecture	31
Figure 7.7	Control Plane Deploy Gitlab Pipeline (1CPxnAP)	32
Figure 7.8	Application Plane Deploy Gitlab Pipeline (1CPxnAP)	33
Figure 7.9	Application Plane Destroy Gitlab Pipeline (1CPxnAP)	33
Figure 7.10	Control Plane Destroy Gitlab Pipeline (1CPxnAP)	33
Figure 7.11	nCPxnAP Architecture	34
Figure 7.12	Control Plane Deploy Gitlab Pipeline (nCPxnAP)	35
Figure 7.13	Control Plane Destroy Gitlab Pipeline (nCPxnAP)	35
Figure 7.14	Business Register Sequence Diagram	36
Figure 7.15	Admin Login Sequence Diagram	37
Figure 7.16	Admin Get Business Sequence Diagram	37
Figure 7.17	Create Business User Sequence Diagram	38
Figure 7.18	Business User Login Sequence Diagram	38
Figure 7.19	Business User Get Business Sequence Diagram	39
Figure 7.20	Screenshot of Test Dataset Generation	39
Figure 7.21	High Level Diagram of Automated Testing Setup	41
Figure 7.22	Application Metrics (1CPx1AP)	42

Figure 7.23	Database Metrics (1CPx1AP)	43
Figure 7.24	Final Latencies Analysis (1CPx1AP)	45
Figure 7.25	Final Summary Latencies Analysis (1CPx1AP)	46
Figure 7.26	Application Metrics (1CPxnAP)	47
Figure 7.27	Database Metrics (1CPxnAP)	48
Figure 7.28	Screenshot of Databases (1CPxnAP)	49
Figure 7.29	Final Latencies Analysis (1CPxnAP)	51
Figure 7.30	Final Summary Latencies Analysis (1CPxnAP)	52
Figure 7.31	Application Metrics (nCPxnAP)	53
Figure 7.32	Database Metrics (nCPxnAP)	54
Figure 7.33	Final Latencies Analysis (nCPxnAP)	56
Figure 7.34	Final Summary Latencies Analysis (nCPxnAP)	57
Figure 7.35	Business Register - Summary Latencies Comparison	58
Figure 7.36	Admin Login - Summary Latencies Comparison	58
Figure 7.37	Admin Get Business - Summary Latencies Comparison	58
Figure 7.38	Create Business User - Summary Latencies Comparison	58
Figure 7.39	Business User Login - Summary Latencies Comparison	58
Figure 7.40	Business User Get Business - Summary Latencies Comparison	58

LIST OF TABLES

Table	Description	Page
Table 7.1	1CPx1AP Application Plane	28
Table 7.2	1CPx1AP Region	28
Table 7.3	1CPx1AP Countries	29
Table 7.4	1CPxnAP Application Planes	31
Table 7.5	1CPxnAP Regions	32
Table 7.6	1CPxnAP Countries	32
Table 7.7	nCPxnAP Control Planes	34
Table 7.8	Country and EC2 Region Mapping	41
Table 7.9	Architecture Comparison Summary	62

LIST OF ABBREVIATIONS

Abbreviation	Description
AP	Application Plane
API	Application Programming Interface
AWS	Amazon Web Services
CD	Continuous Deployment
CI	Continuous Integration
CP	Control Plane
DB	Database
DNS	Domain Name System
ECR	Elastic Container Registry
ERD	Entity Relationship Diagram
IaC	Infrastructure as Code
ID	Identification
IT	Information Technology
MTA	Multi Tenant Architecture
SaaS	Software as a Service
SLA	Service Level Agreement
S3	Simple Storage Service
TPMS	Third Party Management Systems
UI	User Interface
URL	Uniform Resource Locator

LIST OF APPENDICES

Appendix	Description	Page
Appendix - A	Part of Tenant Manager API Code	67
Appendix - B	Part of Control Plane Terraform Code	71
Appendix - C	Part of Control Plane Gitlab Pipeline Code	74
Appendix - D	Part of Retail Manager API Code	79
Appendix - E	Part of Application Plane Terraform Code	82
Appendix - F	Part of Application Plane Gitlab Pipeline Code	84
Appendix - G	Test Dataset Generation Code	89
Appendix - H	Part of Test Dataset	91
Appendix - I	Automated Testing Script	92
Appendix - J	Automated Testing Infra Terraform Code	103
Appendix - K	Example Results of Automated Testing	104