**A close-up of a logo

Description automatically generated**

|  |
| --- |
| **Real-Time Collaboration Platform Using WebSockets and Event-Driven Architecture** |

**Christos Stylidis**

SID: 205364504177

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

*Master of Science (MSc) in Information and Communication Systems*

JANUARY 2025  
THESSALONIKI – GREECE

****

|  |
| --- |
| **Real-Time Collaboration Platform Using WebSockets and Event-Driven Architecture** |

**Christos Stylidis**

SID: 205364504177

|  |  |
| --- | --- |
| Supervisor: | Prof. Leonidas Akritidis |
| Supervising Committee Members: | Assoc. Prof. Panayiotis Bozanis  Assist. Prof. Dimitrios Karapiperis |

SCHOOL OF SCIENCE & TECHNOLOGY

A thesis submitted for the degree of

*Master of Science (MSc) in Information and Communication Systems*

JANUARY 2025  
THESSALONIKI – GREECE

Abstract

This dissertation explores the development of a **real-time ticketing system** leveraging **WebSockets and an event-driven architecture**. The platform facilitates instant updates, allowing users, agents, and admins to seamlessly interact with ticket data. The focus of this project was to address challenges in real-time data synchronization, ensuring scalability and maintaining role-base access control.

The system was implemented using **Node.js, Express.js, MongoDB, and Kafka, Authentication Setup** (JWT) to enable real-time updates for ticket creation, assignment, and status changes. Websocket integration ensures live feedback for dashboard operation, while Kafka serves as the backbone for reliable message queueing between components. The codebase includes role-based dashboards for users, agents, and admins, supporting dynamic operations like viewing, creating and updating tickets.

Acknowledgment is extended to my supervisor, **Leonidas Akritidis**, and committee members **Panayiotis Bozanis** and **Dimitrios Karapiperis** for their valuable insights and support. Gratitude is also expressed to the **IHU faculty and peers** for their feedback encouragement throughout this project.

Christos Stylidis

23.12.2024

Table of Contents

Table of Contents

[Abstract 3](#_Toc186672938)

[Table of Contents 4](#_Toc186672939)

[1 Introduction 5](#_Toc186672940)

[1.1 Overview 5](#_Toc186672941)

[1.2 Motivation 6](#_Toc186672942)

[1.3 Objectives 7](#_Toc186672943)

[1.4 Structure 8](#_Toc186672944)

[2 Literature Review 9](#_Toc186672945)

[2.1 Overview 9](#_Toc186672946)

[2.2 Related Work 10](#_Toc186672947)

[2.3 Critical Analysis 10](#_Toc186672948)

[**2.3.1** **Strength of Existing Systems** 10](#_Toc186672949)

# **Introduction**

## **Overview**

The growing reliance on real-time communication systems in various domains has emphasized the need for robust, scalable, and efficient solutions. Real-time systems have become critical in ensuring seamless interactions, particularly in industries requiring instantaneous updates, such as customer support, logistics, and collaborative platforms. These systems aim to enhance user experiences by delivering immediate feedback and reducing delays.

Despite advancements, traditional systems often face challenges in managing high-volume, bi-directional communication and ensuring consistency across different user roles. These limitations become particularly evident in scenarios involving multiple stakeholders, such as users, agents, and administrators, working collaboratively within a system. The need for a reliable solution synchronize data, ensure real-time updates, and maintain role-based access has never been more crucial.

This dissertation system designed to address these challenges. By leveraging WebSockets and an event-driven architecture, the system ensures instantaneous updates for ticket creation, status changes, and assignment operations. Built using Node.js, Express.js, MongoDB, and Kafka, the platform incorporates scalable and efficient communication mechanisms while maintaining data integrity. The proposed solution aims to offer an intuitive responsive user experience across dashboards for users, agents, and administrators.

## **Motivation**

The increasing reliance on real-time systems across industries such as e-commerce, customer support, logistics, and collaborative platforms has highlighted the critical need for systems that an deliver instantaneous updates and seamless interactions. Traditional communication systems often struggle to meet these demands due to latency issues, lack of scalability, and difficulty in maintaining data consistency among various stakeholders. As businesses increasingly shift towards digital and interconnected operations, the need for reliable and scalable real0time communication platforms has become more pressing than ever.

Despite advancements in technology, many existing solutions fail to effectively address challenges like role-based access control, real-time data synchronization, and handling high-volume data streams efficiently. Users often face delays in receiving updates, and administrators encounter difficulties in managing operations dynamically. These limitations can lead to customer dissatisfaction, operational inefficiencies, and ultimately, business losses.

The dissertation is motivated by the opportunity to bridge these gaps by leveraging modern technologies like WebSockets, Kafka, and an event-driven architecture to develop a robust real-time ticketing system. The proposed system not only addresses the technical limitations of existing solutions but also provides a scalable and intuitive platform that ensures consistent and secure interactions for all users-be they customers, agents, or administrators. The project aims to contribute to the ongoing evolution of real-time communication systems, offering innovative approaches to tackle long0standing challenges and setting a foundation for future advancements in this field.

## **Objectives**

The primary objective of this dissertation is to develop a robust and scalable real-time ticketing system that addresses the challenges of modern communication and operational needs in various domains. Be leveraging advanced technologies such as WebSockets, Kafka, and an event-driven architecture, the system aims to provide instantaneous updates. Seamless interactions, and enhanced role-based access for users, agents, and administrators.

Specific objectives include:

1. **Real-Time Updates:** To enable instantaneous updates for ticket creation, assignment, and status changes, ensuring a smooth and uninterrupted flow of information among stakeholders.
2. **Data Consistency and Synchronization:** To ensure that all users have access to the most up-to-date information by maintaining real-time data synchronization across the system.
3. **Role-Based Access Control:** To implement secure and efficient access controls that differentiate user privileges and ensure that each role-whether a customer, agent, or administrator-can perform its designated tasks without interference.
4. **Scalability:** To design the system to handle high volumes of concurrent users and data streams without compromising performance or reliability.
5. **User Experience:** To deliver an intuitive and responsive interface for all users, enabling effortless navigation and task completion.
6. **System Reliability and Security:** To build a system that ensures data integrity, prevents unauthorized access, and provides fault-tolerant mechanisms to handle potential systems failures.

This dissertation aims not only to address existing limitations in real-time communication systems but also to set a foundation for future enhancements, supporting scalable and innovative solutions in the field.

## **Structure**

This dissertation is organized into several chapters, each addressing specific aspects of the development and implementation of real-time ticketing system:

1. **Chapter 1:** **Introduction**

This chapter provides an overview of the project, its motivation, and objectives. It sets the foundation for understanding the problem domain and outlines the structure of the dissertation.

1. **Chapter 2: Literature Review**

This chapter surveys existing research and technologies related to real-time communication systems, event-driven architectures, and role-based access control. It highlights the strengths and limitations of current solutions and positions this project within the broader research landscape.

1. Chapter 3: System Design and Architecture

This chapter delves into the architectural design of the system, detailing the use of technologies such as WebSockets, Kafka, and MongoDB. It explains the rationale behind the chosen design and how it addresses the identified challenges.

1. Chapter 4: Implementation

This chapter provides a comprehensive discussion of the implementation process, including the development of key system features like real-time updates, fault-tolerant mechanisms, and user role management.

1. Chapter 5: Results and Evaluation

This chapter evaluates the system’s performance, scalability, and reliability. It includes testing results and discusses how the system meets the objectives outlined earlier.

1. Chapter 6: Conclusions and Future Work

This chapter summarized the contributions of the dissertation and suggests potential direction for future research and development

1. Appendices

Additional materials, including code snippets, diagrams, and datasets, are provided here for reference.

# **Literature Review**

## **Overview**

The rapid advancements in technology have reshaped the landscape of communication systems, making real-time data processing and exchange an essential component of modern applications. From e-commerce platforms to collaborative tools and customer support systems, the demand for seamless and instantaneous communication has become a cornerstone for businesses and organizations seeking to enhance user experiences and operational efficiency.

Traditional communication systems, despite their widespread adoption, often fall short in addressing the complexities of real-time data exchange. These systems are typically hindered by latency issues, scalability challenges, and a lack of mechanism to ensure data consistency and reliability. As industries become increasingly communication has grown significantly.

This dissertation builds upon the foundation of existing research and leverages cutting-edge technologies such WebSockets, Kafka, and event-driven architectures to develop a real-time ticketing system. By addressing key challenges such as role-based access control, real-time updates, and high-volume data management, this work aims to contribute to the ongoing evolution of real0time communication systems. The subsequent sections will explore existing literature in this domain, critically analyzing related work and identifying gaps that this project seeks to address.

## **Related Work**

The purpose of this section is to explore existing research and systems relevant to real-time communication platforms, focusing on the technologies and methodologies that address challenges such as scalability, latency, and role-based control.

By critically examining related work, this dissertation aims to identify the strengths and limitations of existing solutions, providing a foundation for understanding how the proposed system contributes to the field.

This section will focus on systems and research that utilize WebSockets, Kafka, and event-driven architectures, discussing their applications, performance, and shortcomings in real-time communication.

## **Critical Analysis**

This section evaluates existing real-time communication systems and methodologies by analyzing their strengths and limitations. The analysis focuses on key aspects such as scalability, latency, fault tolerance, and role-based access control, which are critical to modern real-time applications. By identifying gaps in current approaches, this section sets the foundation for justifying the need for the proposed solution outlined in this dissertation.

### **Strength of Existing Systems**

**Scalability**

Scalability is a critical aspect of real-time systems, enabling them to handle increasing loads without compromising performance. It ensures that systems can grow efficiently to meet users demands, whether through hardware upgrades or distributed architectures. Technologies like Kafka exemplify scalability through features such as horizontal scaling, where additional brokers can be added to the cluster to distribute workloads effectively.

For instance, Kafka’s message partitioning mechanism allows data to be segmented and processed in parallel across multiple nodes, ensuring high throughput even under heavy traffic. This capability is particularly valuable for systems like e-commerce platforms and ticketing systems that experience significant traffic surges during peak periods. By supporting seamless scalability, these systems can maintain responsiveness and reliability, providing consistent user experiences regardless of demand fluctuations.