



Research Digest on Engineering Management and Social Innovations

Available on: <http://rdems.in>

DOI: 10.46647/rdems0204028

Volume: 02 Issue: 04, April, 2026

p-ISSN: 3117-4418

e-ISSN: 3117-4426

Developing A Real Time Auction Enabled Multi-Vendor E-Commerce Platform with Distributed Backend

Dr. KSRK Sharma, Donti Mriduldev, K.C.Udith Kumar Naik, K Chiranjeevi S K S, Mohammed Raashed

Department of Artificial Intelligence & Data Science, Vidya Jyothi Institute of Technology, Hyderabad, India

Abstract

There is a great need for highly scalable, interactive, and real-time solutions due to the exponential rise of e-commerce platforms. Synchronous communication and static pricing structures, which restrict user engagement and do not facilitate dynamic interactions like auctions, are the main features of traditional e-commerce platforms. This paper offers a scalable, efficient, real-time auction-enabled multi-vendor e-commerce platform based on distributed microservices architecture. Real-time bidding with low latency is made possible by the suggested system's integration of a high-performance auction engine with traditional e-commerce features. The architecture makes use of WebSocket connectivity for real-time bid updates, a specialized C++ module for processing auctions, and Spring Boot for backend services. The technology guarantees quick reaction times and effective management of multiple users at once. Comparing the platform to conventional systems, experimental research shows that it delivers high throughput, low latency, and enhanced scalability. Real-time feature integration creates a competitive marketplace and significantly increases user engagement.

Index Terms—E-Commerce, Real-Time Auction, Microservices, WebSockets, Distributed Systems, Scalability, C++

I. Introduction

Over the past ten years, the digital commerce ecosystem has rapidly changed due to the development of internet technology and the rising demand for online services. It is expected of modern e-commerce systems to support several vendors, manage massive volumes of traffic, and offer flawless customer experiences. Nevertheless, the majority of current systems lack interactive elements like real-time auctions and rely on fixed price structures. Dynamic pricing is introduced by real-time auction systems, allowing consumers to compete for goods in a time-limited setting. Price discovery is improved and user engagement is increased. However, there are issues with managing concurrent bids, guaranteeing data consistency, and reducing latency when implementing such systems within conventional infrastructures. In order to overcome these obstacles, the suggested system integrates a high-performance auction engine with a distributed microservices-based architecture. This approach ensures scalability, flexibility, and real-time responsiveness.

II. Significance

In today's digital marketplaces, real-time auction systems are essential. Auctions produce a dynamic environment where prices are set by user demand, in contrast to static pricing approaches. Among the main benefits are: Enhanced user involvement via competitive bidding Real-time communication and comments Improved methods for pricing discovery Increased revenue potential and exposure for vendors

a. Theoretical Background



Research Digest on Engineering Management and Social Innovations

Available on: <http://rdems.in>

DOI: 10.46647/rdems0204028

Volume: 02 Issue: 04, April, 2026

p-ISSN: 3117-4418

e-ISSN: 3117-4426

1. In contemporary digital marketplaces, real-time auction systems are essential. Auctions establish a dynamic environment where prices are set by user demand, in contrast to static pricing approaches. Important benefits include: Enhanced user interaction via competitive bidding Interaction and response in real time Improved processes for price discovery Increased chances for revenue and visibility for vendors
2. Microservices Architecture: The system is divided into autonomous services via microservices, which may be independently developed, deployed, and scaled. Fault tolerance and system flexibility are enhanced as a result.
3. Real-Time The system is divided into different services by microservices, which may be independently built, deployed, and scaled. This enhances fault tolerance and system flexibility.
4. High-Performance Computing using C++: Real-time bidding systems depend on effective memory management and quick execution, which are ensured by using C++ for auction processing.
5. Distributed Systems: By distributing the workload among several nodes, distributed systems enhance scalability and performance.

III. Literature Survey

Monolithic architectures, which closely connect system components, were the focus of earlier e-commerce research. Although these systems are simpler to create at first, they have maintenance and scalability issues. Microservices architecture is emphasized in recent research as a way to address scalability problems. Monolithic architectures, which closely connect system components, were the subject of earlier e-commerce research. These systems are simpler to create at first, but they have maintenance and scalability issues. Microservices architecture is emphasized in recent studies as a way to address scalability problems. WebSockets are perfect for applications that need real-time updates since they lower latency and increase responsiveness. Additionally, auction systems have been thoroughly researched, with a focus on fairness and concurrency management. WebSockets are perfect for applications that need real-time updates since they decrease latency and increase responsiveness. Additionally, auction systems have been thoroughly investigated, with a focus on fairness and concurrency management. WebSockets, and high-performance computing.

IV. Methodology

a. Existing System

Conventional e-commerce platforms have a number of drawbacks. Models of static pricing High latency as a result of HTTP transmission Limited scalability and absence of real-time communication These systems are not appropriate for dynamic settings that need real-time updates. Conventional e-commerce platforms have a number of drawbacks: Static models of pricing High latency as a result of HTTP exchange Insufficient scalability and no real-time communication In dynamic contexts where real-time updates are necessary, these technologies are not appropriate. Sprint 5-8 (Marketplace Logic): Product catalogs, Elasticsearch indexing, and vendor onboarding procedure implementation. Sprint 9–12 (Real-Time Auction Core): Using Boost to develop the C++ bidding service. implementing the WebSocket handshake protocols and Asio. Sprint 13–16 (Scaling & Resilience): Performing geo-redundant deployment exercises and coordinating the Kafka-based Saga transactions.

a. Proposed System

The suggested system presents: A C++-implemented real-time auction engine with distributed backend services and instantaneous updates via Spring Boot WebSocket-based communication



Research Digest on Engineering Management and Social Innovations

Available on: <http://rdems.in>

DOI: 10.46647/rdems0204028

Volume: 02 Issue: 04, April, 2026

p-ISSN: 3117-4418

e-ISSN: 3117-4426

Scalable communications and database systems The technology ensures consistency and fairness by processing user bids in real time and instantaneously updating all players.

IV. System Design

There are several layers in the system architecture:

a. Frontend Layer: Built using Next.js and React, the frontend offers an interactive user interface for perusing merchandise and taking part in auctions.

b. Backend Layer: Business logic, authentication, and API administration are handled by the backend, which is implemented using Spring Boot.

C. Backend Layer: Using Spring Boot, the backend manages APIs, business logic, and authentication.

D. Backend Layer: The backend manages APIs, business logic, and authentication and is implemented using Spring Boot.

• E. Modules for User Interaction Users are able to:

- Look through the products
- Take part in auctions
- Make bids in real time.
- Sign up and log in

F. Data Execution and Processing Scalability and fault tolerance are ensured by data flowing through message queues like Kafka and APIs.

G. Execution and Processing of Data Scalability and fault tolerance are ensured by data flows via message queues like Kafka and APIs.

SYSTEM EVALUATION

The suggested Real-Time Auction Enabled Multi-Vendor E-Commerce Platform with Distributed Backend was evaluated in a number of ways, including usability assessment, functional correctness, performance efficiency, and comparative benchmarking. These assessments guarantee that the system satisfies both user-centric and technical requirements.

Functional Evaluation

The suggested Real-Time Auction Enabled Multi-Vendor E-Commerce Platform with Distributed Backend was evaluated in a number of ways, including usability assessment, performance efficiency, functional correctness, and comparative benchmarking. These assessments guarantee that the system satisfies both technical and user-centered specifications.

User and Vendor Management

The system effectively facilitates:

- Authorization, authentication, and registration of users
- Role-based access control for buyers, vendors, and administrators; vendor onboarding and store management

The platform is able to effectively support:

- Registration, permission, and authentication of users
- Store management and vendor onboarding
- Role-based access control for administrators, vendors, and customers.

Product and Auction Management

The system allows vendors to:

- Add, modify, and remove products
- Select between postings with fixed prices and those that are up for auction.
- Set up the base price, start and end times, and other auction conditions.

Testing of the auction lifecycle confirmed:

- Precise validation of bids
- Preventing duplicate or erroneous bids
- After the auction is over, the right winner is chosen.



Research Digest on Engineering Management and Social Innovations

Available on: <http://rdems.in>

DOI: 10.46647/rdems0204028

Volume: 02 Issue: 04, April, 2026

p-ISSN: 3117-4418

e-ISSN: 3117-4426

Real-Time Bidding Functionality

Add, modify, and remove goods• Select between postings with a set price and those that are up for auction. • Set up the base price, start time, and end time of the auction.

Testing of the auction lifecycle verified:

- Precise bid verification• Preventing duplicate or incorrect bids• After the auction is over, choose the right winner.

Payment and Order Processing

The system ensures:

- Order creation following an auction or direct purchase • Integration of a secure gateway for payments• Validating transactions and managing failures gateway to paymentValidation of transactions and management of failures

Fault Handling and Reliability

Unsuccessful payments, duplicate orders, and timeout situations were all handled skillfully.:

Timeout situations, duplicate orders, and unsuccessful payments were all handled skillfully.

Performance Evaluation

System responsiveness, scalability, and efficiency under various workloads are the main goals of performance evaluation.

Throughput and Scalability

The distributed architecture made it possible to scale horizontally:

- Thousands of concurrent users were managed by the system. Microservices were separately scaled according to load. • Even traffic distribution was achieved via load balancers.

The system performed steadily during the busiest auction activity, according to stress testing.

Real-Time Communication Performance

WebSocket-based communication demonstrated:

- Minimal lag in bid updates
- Efficient handling of concurrent bid events
- Low network overhead compared to polling mechanisms

Comparative Analysis

- Low network overhead in comparison to polling technologies; • Effective management of concurrent bid events; • Minimal lag in bid updates

Feature	Monolithic System	Proposed Distributed System
Scalability	Limited	Highly scalable
Fault Isolation	Low	High
Deployment	Complex	Modular
Performance	Moderate	High
Maintenance	Difficult	Easier

The distributed architecture greatly increases system resilience and scalability.

Comparison with Traditional E-Commerce Platforms

Comparing Conventional E-Commerce Platforms

Conventional platforms:

- Pay attention to fixed pricesReduced user involvement and limited real-time interaction

The suggested system

- Encourages dynamic
- increases customer engagement through real-time bidding• Facilitates competitive product appraisal

Comparison with Existing Auction Platforms



Research Digest on Engineering Management and Social Innovations

Available on: <http://rdems.in>

DOI: 10.46647/rdems0204028

Volume: 02 Issue: 04, April, 2026

p-ISSN: 3117-4418

e-ISSN: 3117-4426

Existing auction systems often lack:

- Support for multiple vendors; distributed backend scalability; and integrated order and payment systems

All of these features are combined into a single system by the suggested platform..

Usability and Transparency Assessment

Transparency and usability are essential for building user confidence and platform adoption.

User Interface Usability

The following criteria were used to assess the platform interface:

- The simplicity of navigation
- Information about the auction was clear;
- It was responsive on all devices; Users were able to:
- Easily browse products
- Track bids and auction status in real time;
- Take part in auctions without confusion

Vendor Experience

Verified by vendor usability testing

- Easy auction setup;
- Clear statistics and sales tracking;
- Easy product listing procedure

Transparency in Auction Process

Transparency was ensured through:

- Confirmation of vendor usability testing

Easy auction setup; straightforward product listing procedure; and transparent analytics and sales monitoring

Security and Trust

Among the security precautions were:

Secure authentication methods; encrypted communication (HTTPS); fraud detection and monitoring
These actions increased users' trust in the platform.

Accessibility

Security precautions comprised:

Secure authentication methods, encrypted communication (HTTPS), and fraud detection and monitoring

These actions improved the platform's user confidence.

RESULTS

The following were security measures:

Fraud detection and monitoring; secure authentication methods; and encrypted communication (HTTPS)

These actions increased platform confidence among users. Throughout several test scenarios, the created Real-Time Auction Enabled Multi-Vendor E-Commerce Platform successfully executed all essential features. Error-free registration, authentication, product browsing, and active participation in auctions were all possible for users.

The created Real-Time Auction Enabled Multi-Vendor E-Commerce Platform successfully carried out all essential features in a variety of test situations. Error-free registration, authentication, product browsing, and active auction participation were all possible. The system was able to effectively manage thousands of concurrent users because to the distributed microservices architecture. Increased speed, decreased server stress, and optimal resource use were all facilitated by load balancing and caching techniques, especially during moments of high auction demand.

The system was able to effectively manage thousands of concurrent users because to its distributed microservices architecture. Increased speed, decreased server stress, and optimal resource use were



Research Digest on Engineering Management and Social Innovations

Available on: <http://rdems.in>

DOI: 10.46647/rdems0204028

Volume: 02 Issue: 04, April, 2026

p-ISSN: 3117-4418

e-ISSN: 3117-4426

all facilitated by load balancing and caching techniques, especially during moments of high auction demand. The system demonstrated great resilience under stressful situations in terms of scalability and reliability. The flexibility to scale services independently allowed for effective management of growing workloads without compromising system stability. While message queues maintained event consistency during temporary problems, fault tolerance methods made sure that failures in specific services did not affect the platform as a whole.

CONCLUSION

The system demonstrated great resilience under stressful circumstances in terms of scalability and reliability. Independent service scaling allowed for effective management of growing workloads without compromising system stability. While message queues maintained event consistency during temporary problems, fault tolerance methods made sure that failures in specific services did not affect the platform as a whole.

The design and construction of a multi-vendor e-commerce platform with real-time auction capabilities and a distributed backend architecture was reported in this study. The study focused on the main drawbacks of conventional e-commerce systems, especially with regard to user engagement, scalability, and flexibility..

The design and construction of a multi-vendor e-commerce platform enabled by real-time auctions and backed by a distributed backend architecture was reported in this study. The study addressed the main drawbacks of conventional e-commerce systems, especially with regard to user engagement, scalability, and adaptability.

The results of the performance study showed consistent system behavior under high load conditions, minimal latency, and high throughput. Additionally, the system demonstrated a high degree of resilience to malfunctions, guaranteeing continuous service availability.

All things considered, the suggested platform embodies a contemporary method of e-commerce by merging:

Distributed computing, real-time communication, and scalable cloud architecture

Low latency, fast throughput, and stable system behavior under high load situations were shown by the performance evaluation findings. Additionally, the system demonstrated a high degree of resistance to failures, guaranteeing continuous service availability.

All things considered, the suggested platform embodies a contemporary method of e-commerce by fusing:

Real-time communication, distributed computation, and scalable cloud infrastructure

REFERENCES

1. G. DeCandia et al., "Dynamo: Amazon's Highly Available Key-Value Store," *Proceedings of the 21st ACM Symposium on Operating Systems Principles*, 2007.
2. M. Fowler and J. Lewis, "Microservices Architecture," ThoughtWorks, 2014.
3. A. S. Tanenbaum and M. Van Steen, *Distributed Systems: Principles and Paradigms*, 2nd Edition, Prentice Hall, 2007.
4. T. Erl, *Cloud Computing: Concepts, Technology & Architecture*, Prentice Hall, 2013.
5. I. Foster et al., "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," *International Journal of High Performance Computing Applications*, 2001.
6. A. Banks and R. Gupta, "MQTT Version 3.1.1," OASIS Standard, 2014.
7. F. Bonomi et al., "Fog Computing and Its Role in the Internet of Things," *Proceedings of the MCC Workshop on Mobile Cloud Computing*, 2012.



Research Digest on Engineering Management and Social Innovations

Available on: <http://rdems.in>

DOI: 10.46647/rdems0204028

Volume: 02 Issue: 04, April, 2026

p-ISSN: 3117-4418

e-ISSN: 3117-4426

8. N. Dragoni et al., "Microservices: Yesterday, Today, and Tomorrow," *Present and Ulterior Software Engineering*, Springer, 2017.
9. S. Newman, *Building Microservices: Designing Fine-Grained Systems*, O'Reilly Media, 2015.
10. J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Communications of the ACM*, 2008.