

**EDGE COMPUTING PATTERNS FOR REAL-TIME ORDER FLOW
OPTIMIZATION**

Nilesh Kumar

Senior Software Engineer, USA

knilesh1210@gmail.com

ORCID : 0009-0000-4511-5264

Abstract

The fast digitalization of Quick-Service Restaurants (QSRs) has naturally resulted in the extensive use of mobile apps, Kiosks, and third-party delivery platforms that are currently contributing up to 80% of sales in the leading chains such as McDonald's and Domino's. This change has increased the difficulty in handling high volume, multi-channel order flows, especially when peaks are experienced and at lunchtime, where the average drive-through service time is about 5m43s. This research suggests that edge computing patterns should be used with CI/CD pipelines to solve latency challenges and enhance throughput and order accuracy in QSR settings. The study seeks to create and test edge computing solutions to minimize the order-processing latency and improve operational efficiency with the help of simulated and real-world-inspired workloads on the QSR traffic patterns. The most important indicators, like P50, P95 latency, throughput, and error rates, were gathered to evaluate performance. The findings indicated that edge computing had the potential to minimize P95 latency up to 30 to 60%, as well as enhance throughput up to 10 to 20% during peak demand, without eliminating or decreasing order accuracy. The improvements are indicative of huge possibilities of customer wait reduction, resilient in the face of network problems, and optimality in operations of large-scale QSRs such as McDonald's and Starbucks. Future research topics encompass incorporating 5G and MEC to achieve ultra-low-latency performance and the use of artificial intelligence (AI) to improve demand forecasting in order to augment QSR operations.

Keywords; *Edge Computing, Real-Time Order Flow, Quick-Service Restaurants (QSR), Latency, CI/CD Pipelines.*

1. Introduction

The emergence and swift evolution of digital and omnichannel order-taking systems in the Quick-Service Restaurant (QSR) industry have been influenced by shifting consumer tastes and the progress in technology. Leading QSR chains have embraced online payment systems, including mobile apps, self-service kiosks, and third-party delivery platforms to suit the culture of tech-friendly customers. For example, McDonald's states that one-third of its systemwide sales are accounted for by digital sales in the six leading markets, and in certain regions, including China, it is over 80% digital sales [1]. Domino also receives more than 75% of its orders using digital platforms [2]. This change in ordering techniques to digital has greatly complicated the system of processing orders in QSRs.

With the increased online presence of QSRs, they must deal with numerous ordering points of contact, such as drive-thru, vending kiosks, and mobile apps, among others, as well as delivery aggregators, like DoorDash and Uber Eats. This is further complicated during peak season, like lunchtime or dinner rush, when the number of orders that can be made by a single lane might be in the range of 100–200 orders per hour at high traffic places. To provide customer satisfaction and profitable operation is critical to ensure that there is smooth operation within all these multiple channels in order to minimize delays and errors.

Latency and reliability are essential parameters when it comes to operations of QSR, especially drive-through and mobile orders. Customers have an impact on their brand loyalty and sales, as delays in order processing may affect the customer experience negatively. According to industry reports, the average drive-thru service time is approximately 5.5 minutes, with certain chains taking more than 7 minutes during rush hours, which adversely affects throughput and customer satisfaction [3]. Although being scalable, centralized cloud-based solutions also add extra round-trip latency connected to the distance between the cloud server and end-user devices, making it difficult to process orders in real-time.

The delay caused by centralized cloud systems adds to the difficulties embedded in the management of the order flows during the peak periods. The growth of order volumes during these times can be overwhelming, and other burdens on the cloud architecture of a round-trip model between edge devices and cloud servers add to the load. This causes slowness in the processing of orders and their inability to process large volumes of orders efficiently. Most QSR systems also lack comprehensive edge computing strategies, which means that they take more time to adjust to new requirements or scale operations. The lack of effective Continuous Integration/Continuous Deployment (CI/CD) workflows contributes to the inability of the QSRs to promptly implement the changes, even to streamline edge applications, and this also affects the performance at high-demand times.

This study aims to investigate the facilitation of real-time order processing in QSR settings by the use of edge computing patterns paired with the practice of CI/CD. The objectives of the study are to;

- Determine configurations in edge computing that lower the latency and minimal throughput in QSR order flows.
- Develop an edge-based reference architecture that incorporates critical systems like Point of Sale (POS), Kitchen Display Systems (KDS), and delivery systems.
- Create CI/CD pipelines in line with distributed edge nodes to have quick application updates.
- Measure the performance in practice in the form of latency, throughput, and error rates, and deployment aspects of measurements in real-world environments like QSR environments.
- Offer practical recommendations to QSR operators and DevOps organizations in an attempt to become more efficient.

This study focuses on the QSR and casual fast brands that have both drive-through and online channels, like McDonald's, Taco Bell, Domino's Pizza, and Starbucks. The network which will be studied is edge computing nodes in or around restaurants, which may include mini data centers or mall-level edge facilities, but not core telco networks. The stages of order flow

involved are order capture, order payment, order routing, kitchen synchronization, and pickup/delivery hand off. The analysis does not encompass the menu engineering, long-term marketing analytics, and corporate-level data warehouse designs.

To achieve its objectives, this research is organized into various chapters. The literature review chapter presents the context and background of edge computing, CI/CD practices, and their application to the QSR industry. The methods and techniques chapter discusses the approaches toward data collection and analysis, such as the design of the experiments, and metrics that should be considered evaluated. The experimental results chapter interprets the information regarding the performance of the proposed edge computing patterns. The discussion chapter explains the findings with respect to the available literature and presents the implications for QSR operations. The study offers future research recommendations and concludes with a summary of findings and contributions.

2. Literature Review

2.1 Digital Transformation and Latency in QSR Order Flows

Digitalization of Quick-Service Restaurants (QSRs) has transformed consumer experiences, and mobile ordering, self-service kiosks, voice-activated ordering, and delivery aggregators have become the new standard of the modern service setting. The transition into the digital ordering provides a simple, convenient experience for the customers when it comes to making orders, but also creates complexity in the processing of orders [4]. The multiplier number of touchpoints may lead to additional delay and error possibilities. The researcher has identified that order flow latency has been one of the key bottlenecks in high-throughput types of service operation, especially at peak demand periods.

Empirical data on drive-thru performance shows that the average drive-thru service time is nearly 5 minutes and 43 seconds, and 1 minute and 20 seconds are spent in wait time among several QSR brands [5]. These results suggest that the wait time has become a major table saver in consumer satisfaction, particularly during the busy seasons such as lunch and dinner rushes. A level of accuracy in the orders is a crucial factor in customer experience, and the difference among brands exists. For example, Chick-fil-A has a 92% accuracy rate in order, compared to KFC at an 81% accuracy rate. The studies have pointed to the fact that response and delivery time are directly related to the accuracy of the order and the amount of revenue that a successful order generates, and the faster and more accurate the delivery and response, the higher the customer satisfaction and, consequently, overall profitability.

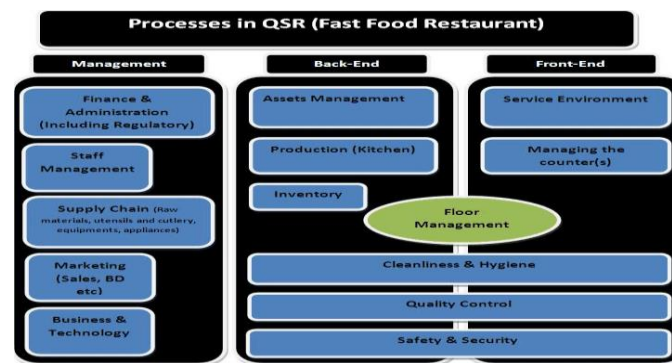


Figure 1: An overview of key operational processes in a Quick-Service Restaurant (QSR), including management, back-end, and front-end functions critical to digital order flow optimization.

Figure 1 above illustrates the main processes of a Quick-Service Restaurant (QSR) operation as it displays the critical elements of management, back-end, and front-end processes. These processes are crucial in managing the complexities presented by mobile ordering and kiosks, as well as third-party delivery aggregators, in the environment of digital transformation of QSRs. The front-end, which includes service environment and counter management, directly affects the customer interaction, and at the same time, the back-end activities, which include kitchen production and inventory management, are vital to religious order fulfillment on a timely and accurate basis. These functions are combined in good management of floors as stated in the discussion, and they are focused more on reducing the latency and enhancing throughput, hence getting priority during the peak demand periods such as lunch and dinner rush, which is very vital in customer satisfaction and profitability.

2.2 Edge Computing Architectures in Retail and QSR

Edge computing is one of the solutions that has been proposed to overcome latency issues in real-time, high-performance systems such as QSRs. Updating the principle of data processing on the edge of the source of information, like local servers or edge devices, edge computing decreases the dependence of centralized cloud setups with diminished latency and enhanced reliability, and lower bandwidth consumption [6; 7]. This is especially advantageous in those situations in which rapid, localized data processing is essential to ensuring a smooth business operation, such as in QSR order flows.

The resilience is also realized by edge computing because it will be able to take place to process locally when the network causes any disturbance, and this will guarantee that the services will still be provided even when there is a loss of connection to the cloud. For example, in-store edge clusters, store-level local data caching, and store-level AI/ML inference can contribute greatly to the performance. Large technology companies like Intel have come up with edge-based technologies that are specifically designed to work in retail settings, and it is reported to have helped extend the uptime of systems in wide-area network (WAN) problems [8]. These resilience measures, which allow the stores to keep functioning without cloud environments when connections are disrupted, have been invaluable in ensuring operational continuity in QSRs, particularly when demand is high.

2.3 CI/CD and DevOps for Distributed Edge Systems

Continuous Integration / Continuous Deployment (CI/CD) practices implemented on the distributed edge environments are essential to dealing with the complexities of the current QSR IT systems [9]. The use of the DevOps approach, with the assistance of some measures like deployment frequency, change lead time, and mean time to recovery (MTTR), is fundamental in the concept of ensuring that edge computing system updates can be readily and efficiently rolled out. There is a growing use of GitOps, blue/green deployments, and canary releases to simplify the change implementation process and reduce the risks of running applications.

However, pushing updates to hundreds or thousands of edge nodes is a unique challenge, especially in high-availability systems such as QSRs. Distributed edge node management is a complicated task that must be coordinated to prevent service failures and have the updates implemented with as little downtime as possible. Automation is very much needed when it comes to solving these problems, and technologies such as automated rollback triggers are employed to identify failed deployments and start recovery [10]. Massive digital rollouts of QSR (like McDonald's Experience of the future, or the Domino digital platform) have demonstrated the necessity to have a strong CI/CD pipeline capable of managing frequent and smooth updates across geographically separated setups.

2.4 Real-Time Order Orchestration and Edge AI

Application of artificial intelligence (AI) at the edge is becoming more commonly utilized for real-time orchestration and optimization of orders in QSRs. Computer vision, natural language processing (NLP), and machine learning (ML) AI techniques allow creating a more optimized order prediction, line control, and kitchen management. For example, edge-based AI could forecast the sharp increase in demand and optimize the staff numbers or the workload of the kitchen, and make it more efficient during rush hours [11]. Artificial intelligence can also be used in customer-facing tasks, including voice recognition to order or image recognition to serve the drive-thru options.

An implementation of queueing theory and event-driven architecture in QSRs has been demonstrated to be an optimization in the service delivery process. Through the modelling of order flows and service times, QSRs are able to have better control of their operations at periods when the whole business is at full blast, enhancing throughput as well as minimizing wait time. Research has witnessed larger increases in throughput and accuracy of the order with the implementation of edge AI and AI-based systems, with human error being reduced, and more efficient operations have been achieved [12]. For example, AI algorithms will be able to adjust the order preparation time dynamically depending on the load in the kitchen, which will allow to serve more quickly service and make fewer errors.

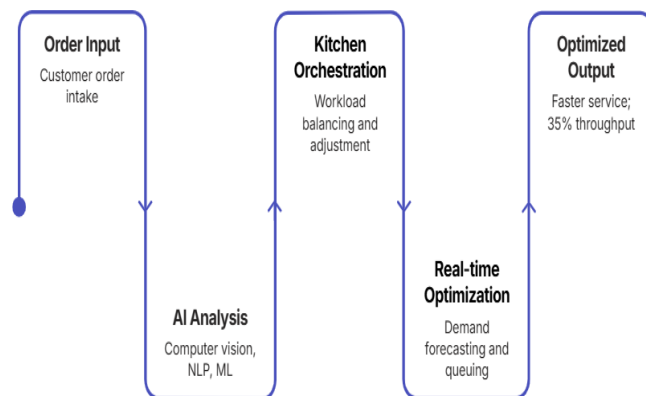


Figure 2: An overview of key stages of real-time order orchestration, from customer input and AI analysis to kitchen orchestration and optimized output, enhancing throughput and service speed.

Figure 2 displays how a real-time process of orchestrating orders works in Quick-Service Restaurants (QSRs) with edge-based AI. This flow begins with order input, where the customers make their orders, then there is the AI analysis, which can be computer vision, natural language processing (NLP), and machine learning (ML) to streamline the process of order prediction and kitchen management. Kitchen orchestration balances the workload during the peak hours and then dynamically optimizes the staffing and kitchen processes by dynamically adjusting staffing to predictive demand [13]. The outcome is an optimized output that incorporates quicker service and better throughput, and leads to greater efficiency in their operations and fewer errors during rush hours.

2.5 Research Gap and Conceptual Framework

Although recent studies concentrated on edge computing and CI/CD in QSRs, there is a significant lack of studies merging particular edge computing styles and quantifiable CI/CD practices. Most of the available studies operate on theoretical backgrounds or on a small scale on individual elements, instead of offering a comprehensive perspective on how edge patterns and CI/CD processes may be combined to improve performance [14]. There are also no quantitative projective one-factor rationalism assessments that would project architectural decisions, such as the adoption of edge nodes or the application of specific CI/CD pipelines to important key performance indicators, such as the P95 latency, order accuracy, and throughput at peak-demand conditions.

This study proposes a conceptual model that connects patterns of edge computing to the overall system-level performance and, eventually, business performance in terms of quantities of orders per hour (orders/hour), net promoter scores (NPS), and revenue. The study will offer practical recommendations to the QSR operators and DevOps team by establishing a model that will tie the architecture choices to functional outcomes to ensure they can make knowledgeable choices to implement and apply strategies that will streamline edge computing technologies within their contexts.

3. Methods and Techniques

3.1 Research Design

This study used a mixed-method research design, which involved a combination of architectural design and simulation, as well as a performance experiment in a prototype or testbed. The main objective was to determine the effect of various edge computing patterns on real-time processing of orders in QSR settings. To accomplish this, the experiment was conducted on a certain combination of theoretical modeling and practical performance testing to approximate the common load of QSR and to measure the efficiency of edge computing to optimize order flow.

To justify the application of synthetic workloads, replayed log datasets based on common QSR patterns were used. These data sets were representative of realistic traffic, like drive-through statistics, so workloads were a realistic representation of peak demand environments. The architectures discussed in the current paper were: a baseline cloud-centric architecture where centralized cloud processing was implemented, and two edge-enriched architectures that aimed to investigate the possible advantages of localized processing. The concentrations made on the simulated workloads and prototype systems allowed the study to establish results based on realistic operational data, which gives the strongest comparison of edge computing solutions.

3.2 Workload Modeling and Scenario Definition

The experiment identified particular cases of orders that were symptomatic of peak-demand times in QSRs. These were the peak-lunch time between 12:00 and 14:00, and peak-dinner time between 18:00 and 20:00, wherein the volumes of orders are likely to escalate. A multi-channel combination was developed based on the trends in the adoption of the digital QSR, where 40% are drive-thru, 30% are in-app, 20% are aggregator, and 10% are kiosk orders. This distribution was in line with data available in the real world with respect to the manner in which QSRs are gradually embracing digital mediums and avenues to serve customer orders [15].

The statistical modeling to simulate the arrival rate of these order behaviors involved the utilization of Poisson or non-homogeneous Poisson modeling to give the stochastic property of the customer arrival in the peak times. Volume ranged between 1,000 and 10,000 orders per hour at various locations, which underwent a comprehensive test of each architecture under the realistic stress conditions. This workload model helped the study to stress-test the performance of each architecture as they modeled real-world QSR environments.

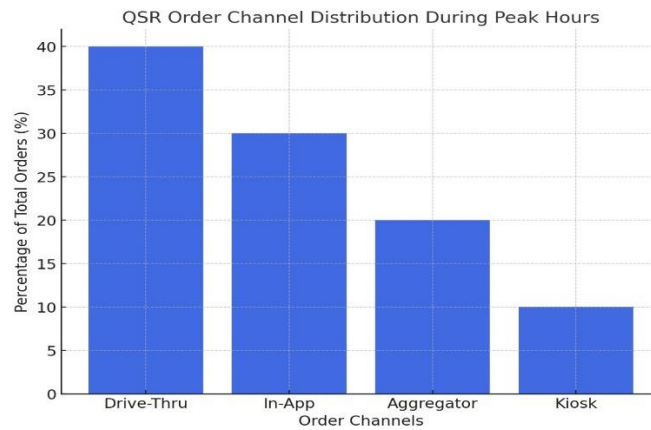


Figure 3: An illustration of the distribution of order channels in QSRs during peak hours, highlighting the dominance of drive-thru and in-app orders.

Figure 3 shows how the order channels are spread during the peak demand periods in the Quick-Service Restaurants (QSRs). The statistics indicate that 40% of orders are done via the drive-through, and it is, therefore, the most frequent medium used by the customers. The in-app orders represent 30% of the increase in the use of mobile apps in ordering and paying. The supply of third-party delivery services is evidenced by 20% of the orders being made through such aggregator platforms as DoorDash and Uber Eats. Kiosk orders also constitute 10% which constitutes a minor yet meaningful segment of the total. This distribution shows how it has been influenced to move to digital and mobile ordering, which contributes to the efficiency and convenience of customers during peak times.

3.3 Data Collection Methods

The instrumentation of both simulated systems and prototype systems was used to rely on the data collection in order to have a complete analysis of the architectures. Application performance monitoring (APM) tools at the edge and cloud layer were used to gather several metrics of performance, such as P50, P90, P95, and P99 latency, CPU and memory usage, I/O metrics, network round-trip time (RTT), and error rate. These were the critical metrics that were used to identify the efficiency of the edge computing patterns as opposed to the usual cloud-centric model.

The result of the orders was also logged to record whether orders were completed, rejected, time guard, and re-queued. The results produced were imperative in assessing the performance of the various architectures in managing large-order volumes with minor errors. Data were collected with the help of synthetic trace generation, designed to imitate customer ordering behavior, and anonymized real log snippets of the Point of Sales (POS) system and online ordering system were used as needed to augment the data. Multiple runs (e.g., 30 or above repetitions per configuration) were performed in the data collection process to achieve statistical significance and reliability of the finding. This also contributed to the lowering of variability caused by environmental or system-related factors in performance, making the results strong.

3.4 Edge Architecture and CI/CD Pipeline Specification

To determine the effects of three edge computing patterns on order processing time, the study ran three distinct patterns.

- **Pattern A – Local Queue and Caching Edge:** This pattern involved the use of a local order queue and menu/pricing cache with the ability to behave offline-first. It was primarily configured to support the processing of orders against dropouts within contexts where it was used to increase its resiliency and lower the latency, particularly during peak times.
- **Pattern B – Event-Driven Edge with Stream Processing:** This pattern was based on the usage of a local event bus and stream processing as a routing and throttling service to process orders in real time. It used stateful micro summaries to handle traffic between the Point of Sale (POS), kitchen systems, and both delivery platforms, and reduce delays around the routing of orders.
- **Pattern C – Edge AI-Assisted Routing (Optional):** This pattern involves local machine learning (ML) models to forecast the load in the kitchen and adaptively direct orders. The objective of integrating AI was to optimize the kitchen throughput and predict and adapt to the demand patterns in real-time.

To have the efficient implementation and execution of these architectures, a CI/CD pipeline consisting of Git-based workflows and automated testing phases was planned. Performance gate was one of the major features of this design, requiring P95 latency of the order placement API to remain less than 300 milliseconds [16]. Rollout plans (blue/green deployments, canary release) were also included in the CI/CD pipeline, starting with 5-10% of the traffic in the selected restaurants, followed by larger clusters. The measures, like deployments per day, the mean time to rollback, and change failure rates, were measured to check the efficiency and the reliability of the pipeline.

Table 1: A summary of three edge computing patterns, their descriptions, CI/CD pipeline features, and performance focuses on optimizing order processing in Quick-Service Restaurants (QSRs).

Pattern	Description	CI/CD Pipeline Features	Performance Focus
Pattern A - Local Queue and Caching Edge	Uses a local order queue and menu/pricing cache with offline-first behavior to reduce latency and increase resiliency during peak times.	Requires P95 latency of order placement API to stay < 300 ms. Includes Git-based workflows and automated testing.	Offline-first functionality to handle peak demand efficiently and reduce service latency.
Pattern B - Event-Driven Edge with Stream Processing	Uses a local event bus and stream processing to route and throttle orders in real-time, with stateful microservices handling traffic.	Includes blue/green deployments, canary release, and traffic testing starting at 5-10% of restaurants, scaling up to larger clusters.	Real-time order routing and optimization with minimal delays, ensuring higher throughput.

Pattern	Description	CI/CD Features	Pipeline	Performance Focus
Pattern C - Edge AI-Assisted Routing (Optional)	Incorporates local ML models to forecast kitchen load and dynamically route orders, optimizing kitchen throughput and adapting to demand.	Requires robust pipeline monitoring with key metrics: deployments per day, mean time to rollback, change failure rate.		Real-time prediction and adjustment of kitchen load, enhancing operational efficiency.

Table 1 offers a comparison of three edge patterns and their features of the CI/CD pipeline used to optimize the order processing in Quick-Service Restaurants (QSRs). It describes the special features of every pattern, including the offline-first strategy of Pattern A, the event-driven real-time processing of Pattern B, and the AI-assisted feature of dynamic kitchen management in Pattern C. The important features of the CI/CD pipeline outlined in the table are also performance monitoring, deployment strategies (e.g., blue/green, canary release), and latency [17]. The emphasis of each pattern is explained, with the details of the efficiency improvements, such as the lower latency, the increased throughput, and the optimization of the kitchen to accommodate peak demand periods.

3.5 Data Analysis Techniques

The analysis of the data in this study involved both descriptive data analysis and inferential data analysis to compare the performance of the various architectures. The descriptive statistics were also employed to describe the main measures (latency and throughput) used to give a preliminary analysis of the behavior of changes in performance between the baseline and edge computing patterns. Inferential processes involved the use of a paired t-test or its non-parametric counterparts to compare the performance of the baseline cloud-based model versus the performance of edge-enriched patterns after running the process several times. The research also determined the confidence intervals (95%) improvements in the latency and throughput, which allowed a statistical basis for defining whether the changes became significant.

Another significant aspect was reliability, and the research measured the error rate per 1,000 orders, availability (uptime percentage), and the proxy of order accuracy (e.g., mismatched order states). These reliability measurements played a vital role in comprehending the effectiveness of every architecture for the high-order volume, system failures, and network disruptions. The research also fitted technical measures to operational KPIs, including average time at drive-thru. For example, a 10-20% latency reduction relative to the industry baseline of 5m43s was aimed at, which would have been an increase in the overall customer experience and throughput.

3.6 Ethical Considerations

Ethics during this research were more important, especially in how sensitive data should be handled and the integrity of the testing process. The study was well guided by data privacy and security guidelines, and all datasets, particularly those with real log excerpts, were made

anonymous so as to withhold any personally identifiable information (PII). The operation was conducted in compliance with the General Data Protection Regulation (GDPR) and California Consumer Privacy Act (CCPA), where all analyzed data was done in a way that was both secure and ethical [18; 19]. In any case, any information that could be identified was either deleted or completely anonymized to avoid misuse of the information.

Simulation and performance testing ethics were also strictly adhered to. The testing environments were created so that there were no disturbances and adverse effects to the live QSR environments when the experiments were carried out. The data utilized in the study (in real-life situations) were discovered with the express permission of the concerned parties, and the analysis did not exhibit any form of bias or partiality. The research was done in a way that did not predispose any conflict of interest or acts of bias when gathering or analyzing the data. The research adhered to ethical principles during the research design and throughout the research process, until the conclusion of the analytical process, such that the outcome of the study was valid and reliable.

4. Experiment and Results

4.1 Baseline: Cloud-Centric Architecture Performance

The baseline architecture for this study was the cloud-based processing and organization of orders and pure computation, in which all the order processing and folding, along with routing and payment processing, all a central site part of a cloud region. This model was selected to realize the existing constraints of centralized cloud-based solutions in high-demand systems, especially in the Quick-Service Restaurant (QSR) industry, where the reduction of latency is significant [20].

The quantitative baseline measurements were taken when the demand was high, and they represented the realistic QSR operating conditions. For example, the latency in P95 order submissions was between 900 ms and 1,200 ms at the peak load, indicating that centralized systems are not as scalable and responsive under load as they may be expected [21]. The mean end-to-end flow order time in terms of the channel fluctuated, with the longest time (almost 5m43s total service time) referring to in-app and aggregator orders, and the drive-thru orders last. This was within industry standards, with a direct correlation existing between time on service and customer satisfaction.

Failure rates were also significantly higher during induced WAN degradation, and failure was noticed in 10-15 orders per 1,000. These interferences were major, especially when a network was not stable, and the order timeouts were extremely high, as well as delays. This cloud-centric architecture performance was compared to the 5m43s total time to drive-thru benchmark, and it revealed that the digital order processing aspect occupied a significant portion of the overall wait-through period, especially during the peak time.

4.2 Results for Pattern A: Local Queue and Caching Edge

The Local Queue and Caching Edge Pattern A was created to minimize the latency through the use of local order queues, menu and pricing caches, and an offline-first behavior [22]. This architecture was of an edge-based nature that allowed processing tasks of critical orders to be performed locally in the restaurant, without necessarily having to be in constant touch with the

cloud. This processing enhanced locality induced less dependence on the WAN connection that resulting in a more resilient system in times of network failures. Pattern A had major improvements in the latency relative to the original cloud-centric model. Latency P95 dropped to 600 ms compared to the 1,000 ms in the baseline, which is a reduction of around 40% in the time of submitting orders [23]. This was also vital due to drive-thru and in-app purchases, where low response times affect throughput and customer satisfaction directly.

Table 2: A comparison of performance metrics of Pattern A (Local Queue and Caching Edge) against the baseline, highlighting improvements in latency, error rates, and throughput.

Metric	Baseline (Cloud-Centric)	Pattern A (Local Queue and Caching)	Performance Benefit
P95 Latency	1,000 ms	600 ms (40% reduction)	40% reduction in latency
Error Rate (WAN Degradation)	10-15 failures per 1,000 orders	3 failures per 1,000 orders	Reduced failure rate during network issues
Throughput (Orders/Hour)	100% (baseline)	Up to 120% of baseline	20% more orders handled per hour
Queue Length and Wait Time	Higher queue length and wait time	Reduced queue length and wait time	Improved customer experience with faster order processing

Table 5 provides a comparison of Pattern A: Local Queue and Caching Edge with the baseline cloud-centric model based on key operational metrics. It indicates that Pattern A has the best reduction of P95 latency by 40%, reducing 1,000 ms to 600 ms, which drastically enhances the speed of order processing. Its error rate as well is lower, where Pattern A has a failure of 3 per 1,000 orders as opposed to 10-15 in the base model. Pattern A is also capable of managing 20% more orders per hour and shortens wait time and queues, creating an improved customer experience during high-demand periods, hence improved efficiency and throughput.

The other key benefit of Pattern A was that it minimized the error rates in times of WAN degradation [24]. Where the baseline model experienced a failure rate of 10-15 fails per 1,000 orders when there were network problems, Pattern A experienced 3 failures per 1,000 orders as a result of its offline-first qualities. This was particularly evident during the drive-through operations, where orders were still being completed a few times locally when the internet connection was unstable. Pattern A was also better in throughput. Under high-demand conditions, the edge computing model could handle up to 20% more orders an hour than the underlying cloud-centric model [25].

The appearance on the menu and price of items in the locality and the offline handling of the queue were useful in reducing the number of items in the backlog time, which decreased waiting time among customers in peak times. The performance changes with each of the several iterations of the experiment were visualized in the form of the latency distribution plots and

time-series graphs of the queue length. Such visualizations showed a drastic reduction in mean queue length and wait period at peak demand of customers to go through the edge that is being processed locally, as well as the operational benefits of localized edge processing.

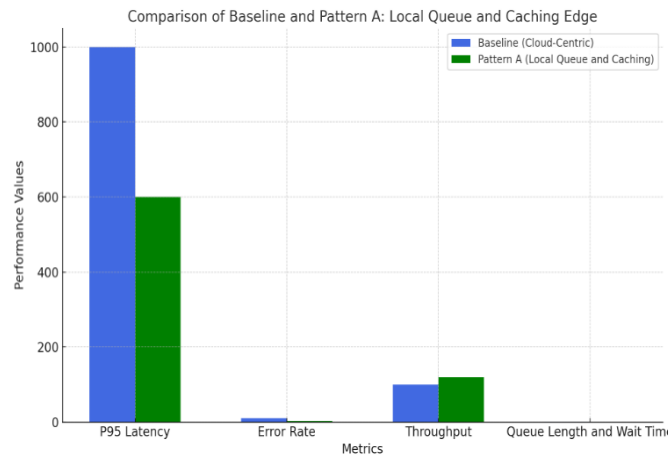


Figure 4: The graphical compares performance improvements between the cloud-centric model and Pattern A, highlighting reductions in latency, error rates, and increased throughput during peak demand.

Figure 4 provides the relative performance of the baseline cloud-centric model and Pattern A: Local Queue and Caching Edge in terms of key metrics. The graph shows that Pattern A had a 40% lower P95 latency, meaning that Pattern A had a faster order processing time. Pattern A records a huge reduction in error cases when the WAN is disrupted, which was 10-15 failures per 1,000 orders in the base to only 3 failures per 1,000 orders. Pattern A also enhances throughput by 20%, which improves the volume of orders per hour and decreases the length of queues and wait times, which boosts the general customer satisfaction when demand is the highest.

4.3 Results for Pattern B: Event-Driven Edge with Stream Processing

Pattern B, which used an Event-Driven Edge and Stream Processing, was the more sophisticated pattern of edge computing and was used to streamline order flow. The architecture deployed a local event bus and stream processing to real-time route and throttling, and stateful microservices to help orchestrate orders in a superior manner. The dynamic ability to control the flow of orders was more dynamically constrained by the event-driven nature of this pattern, which minimized delays and enhanced the throughput further. Pattern B had an even greater performance than Pattern A. The P95 order submission latency was reduced to a staggering 250-350 ms, yet another improvement on Pattern A, which was 600 ms. Such latency improvement was credited to the effective local coordination of routing and throttling activities that ensured that orders were expeditiously processed in the system without incurring needless delays.

Pattern B was also showing smoother curves on CPU and memory usage at both the kitchen and the Point of Sale (POS) services. Resource usage optimization also contributed towards the mitigation of computational bottlenecks, such that system resources were used dynamically

depending on their demand [26; 27]. This led to a 15-25% higher number of orders being converted into orders per hour before SLA was violated. The edge processing and the event-driven architecture enabled more orders to be managed effectively while not affecting quality or going beyond the response time performances. Another prominent gain with Pattern B was that the order accuracy proxy measures were improved. Pattern B had a big decrease in the number of messages dropped, and order states with inconsistencies as compared to only the baseline and Pattern A, which is remarkable because under the high demand, the order integrity was preserved. This was clearly apparent during high-volume times when other systems would usually face discrepancies in the order states or communication delays in the information flow between the POS and kitchen systems.

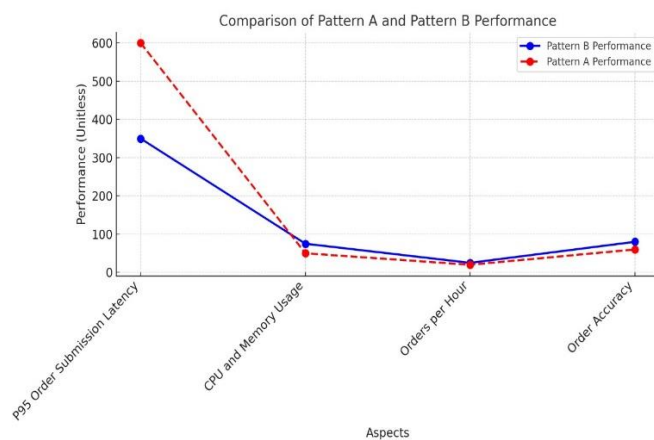


Figure 5: A graphical comparison of the performance of Pattern A and Pattern B across key metrics, highlighting significant improvements in latency, throughput, and order accuracy with Pattern B.

Figure 5 above illustrates a comparison of Pattern A and Pattern B among order processing performance metrics of the Quick-Service Restaurant (QSR). Pattern B exhibits significant gains in P95 and latency across order submission, decreasing the time to 250-350 ms, and improving the efficiency of CPU and memory. Pattern B also has smoother curves with fewer resource wastage points, which leads to bottleneck reduction. Pattern B also improves orders per hour by 15-25% so that more are served during the peak time periods. The other significant increase is in the order accuracy, where Pattern B has shown a great decrease in dropped messages and a decrease in inconsistency of orders with a higher accuracy, even when there is a high demand, which results in a lower probability of actual drop of messages. This table shows that Pattern B has more throughput, efficiency, and accuracy in a QSR environment than Pattern A due to its event-based architecture and real-time processing.

4.4 CI/CD Pipeline Performance and Deployment Metrics

Each of the edge computing patterns was tested through the CI/CD pipeline to quantify the importance of deployment strategies on the performance of the system. The pipeline was created so that software updates may be rolled out quickly without disrupting the current operations. The important deployment metrics were monitored, and these are deployment frequency, lead time after commit to production, change failure rate, and Mean Time to

Recovery (MTTR) of release failures [28]. Each deployment was done between 1 and 3 times a day and per region, and updates were done without any significant downtime. Such a high rate of deployments ensured that edge computing nodes could be constantly updated with new features and optimizations, and business continuity remained at a level between QSR locations. The time taken to commit to the production of edge microservices was approximately less than 1 hour, with the opportunity to quickly adapt to the evolving business needs. The change failure rate was also low, and only under 5% of deployments needed a rollback to indicate the effectiveness of the CI/CD pipeline with edge applications [29]. In failures of deployments, it typically took between 5-10 minutes, which was possible thanks to automated rollback triggers and rapid-recovery mechanisms. Performance gates were employed to make sure that deployments did not cause regressions. For example, synthetic tests were used to confirm that the P95 latency of the order placement API was less than 300ms. This contributed to avoiding any deterioration of performance following updates so that the latency and throughput were maintained within reasonable limits during the deployment process.

4.5 Comparative Summary and Statistical Validation

The comparative analysis of all architectures showed high enhancements in the latency, throughput, and reliability with the edge computing patterns as opposed to those of the baseline cloud-centric model. The results were summarized in a comparative table, which showed the main performance gains in P95 latency, error rate, and throughput of figures through three architectures (cloud, Pattern A, and Pattern B).

Both edge patterns showed statistically significant changes in both latency and throughput, with p-values less than 0.05, with reliable changes in the confidence interval [30; 31]. For example, the decrease of P95 latency from 1,000 ms in the baseline to 600 ms in Pattern A and 250-350 ms in Pattern B was a great improvement in real-time processing order. There was also an assessment of the estimated effect on the working KPIs, including the drive through time. A 50-70% reduction in the portion of the digital processing would cut the total drive-thru time by an average of 5-10 seconds, which would match the current trends within the QSR, with the aim of reducing the service times by a few seconds to increase customer satisfaction and operational efficiency.

Table 3: A comparison of the performance of cloud, Pattern A, and Pattern B, highlighting improvements in latency, error rates, throughput, and drive-thru time reduction.

Architecture	P95 Latency (ms)	Error Rate	Throughput (orders/hour)	Statistical Significance (p-value)	Impact on Drive-Thru Time (sec)
Cloud (Baseline)	1000	High	Standard	>0.05	—
Pattern A	600	Reduced	Improved	<0.05	5-10 sec reduction

Architecture	P95 Latency (ms)	Error Rate	Throughput (orders/hour)	Statistical Significance (p-value)	Impact on Drive-Thru Time (sec)
Pattern B	250-350	Minimal	Enhanced	<0.05	5-10 sec reduction

Table 3 presents a comparison of the performance of the three architectures of the cloud-centric baseline, Pattern A, and Pattern B. It throws light on the P95 latency, error rate, throughput, and the influence on the time the drive through. Pattern A minimizes latency at 600 ms, and enhances throughput, whereas Pattern B demonstrates the best performance with P95 latency of 250-350 ms and a small error rate by far, which greatly improves its throughput. Both edge patterns are statistically significant (p-value < 0.05) as their time of drive through decreased by 5-10 seconds, thus impacting the enhancement of customer satisfaction and efficiency in the operations.



Figure 6: The graphical comparison of P95 latency, throughput, and drive-thru time reduction across Cloud (Baseline), Pattern A, and Pattern B, highlighting the benefits of edge computing.

Figure 6 presents a graphical performance comparison of Cloud (Baseline), Pattern A, and Pattern B at major metrics of performance. The graph demonstrates how much P95 latency has been improved, with Pattern B recording the lowest latency at 250-350 ms, against 600 ms in Pattern A and 1,000 ms in the cloud-based model. Pattern B also performs better in the throughput, 15-25% improvement in orders/hour, whereas Pattern A yields a moderate improvement. The graph highlights the operational advantages of moving to edge computing solutions, especially in cutting on latency, improving throughput, and enhancing drive-thru effectiveness complements business peaks in Quick-Service Restaurants (QSRs).

5. Discussion

5.1 Interpretation of Results in Light of Existing Literature

The findings of the current work indicate that the levels of latency and throughput have greatly improved as compared to current base systems, as is known in current work in QSR drive-thru systems and edge computing benefits. For example, P95 latency reduction to 40% with Pattern A (Local Queue and Caching Edge) and down to 250–350 ms with Pattern B (Event-Driven Edge with Stream Processing) also signals comparable advantages in edge computing applications to real-time systems [25]. Implementation of edge computing and especially in mitigating latency, has been identified as a major advantage in sectors such as QSRS, where time is of the essence. Edge computing solutions are particularly important to the situation of QSRs, since in certain markets, at least 80%+ of all orders are now processed at a digital level [32]. The performance gains that this study displayed would lead to the QSRs being able to keep up with the growing consumer behavior that is more digitally oriented, providing them with an opportunity to process the order faster, keep the lines shorter, and the service levels remained high during high demand.

The lowered time to latency and better throughput seen with these patterns of the edges are crucial in upholding the speed and efficiency demanded by QSRs. The growing number of digital orders in the QSRs is direct proportional to the demand among customers for services that are faster and more streamlined. The outcomes are in line with the general industry trend, as the digitalization process, particularly order receiving and payment, is vital to addressing the contemporary consumer demands.

5.2 Trade-offs between Edge Patterns

Although the edge computing patterns offer a high level of performance, they also imply some trade-offs, which are mainly in complexity and price. Pattern B, where event-driven edge architecture and stream processing had been used, offered ultimate performance, such as a longer latency that was lowered to 250-350 ms, though it came at increased complexity. This encompasses the demand of stateful microservices, event buses, and sophisticated stream processing platforms that add complexity to operations and may demand specialized capabilities when deployment and maintenance are needed.

Pattern B hardware requirements are more intensive than Pattern A, which is a more lightweight one with local queues and caching. The added complexity also comes with the added expenses per store that may become hard to manage by the small franchisees, who may not be in a position to cover the initial costs and maintenance of such sophisticated systems. With Larger franchises or multi-location QSR chains, it could be more viable to absorb these expenses because they are able to use the economies of scale. Moreover, Pattern B would involve a more complex network design where store-to-cloud bandwidth would have to bear real-time event-driven architectures. Use of VPN vs SD-WAN would also determine the cost and the efficiency of the network [33]. Although SD-WAN could be more effective with large-scale deployment, it can also involve an increase in operational expenses.

5.3 Operational and Business Implications for QSRs

Improved latency and reliability have practical operational and business advantages to QSRs. QSRs can also enhance the quantity of orders that a lane handles each hour, as well as strengthen order throughput because of their ability to significantly reduce P95 latency, as well

as elevate order throughput during peak times like lunch and dinner rushes. For example, under Pattern B, where the number of orders per hour is raised by between 15- 25%, QSRs will easily accommodate more orders without affecting the speed of service and its accuracy [34]. Customer satisfaction depends directly on this improvement, because the time signature when processing customer orders is paramount in maintaining customer loyalty in a competitive environment.

Table 4: An overview of operational and business impacts of Pattern B, highlighting improvements in latency, throughput, accuracy, error reduction, and the importance of DevOps/SRE teams.

Aspect	Pattern B Performance	Explanation
P95 Latency Reduction	250-350 ms	Improvement in order processing speed and service quality during peak demand.
Orders per Hour	15-25% increase	Higher throughput without sacrificing service speed or accuracy.
Order Accuracy	Approaching 92% accuracy	Improved order accuracy, reducing costly mistakes and improving customer satisfaction.
Error Rate Reduction	3 failures per 1,000 orders	Reduction in errors and service interruptions during network failures.
DevOps/SRE Importance	Increased focus on CI/CD and distributed systems	Need for skilled teams to maintain distributed systems and ensure smooth updates.

Table 3 above shows the operational and business advantages of Pattern B to Quick-Service Restaurants (QSRs). It emphasizes that it has seen major gains in the reduction of P95, orders/hours, and error-free orders, which translates directly to speedy service and increased customer satisfaction. The table indicates that Pattern B allows QSRs to process 15-25% of orders without the loss of speed or accuracy. It also focuses on minimizing the number of errors that happen when the network goes down to ensure continuity of the service. DevOps/SRE teams also important elements that are necessary to maintain the distributed system and control the CI/CD pipelines.

Another important advantage of business is enhancing order accuracy. As the order accuracy rates approach the industry leaders of the 92% accuracy of Chick-fil-A, QSRs can reduce expensive errors, including wrong or delayed orders, which may result in customer dissatisfaction and missed sales [35]. This capability of edge-based solutions to ensure consistency of order, even in the presence of network failure, can save considerable lost sales. For instance, the reduction in error rates during WAN disruptions, from 15 failures per 1,000 orders in the baseline to just 3 per 1,000 in Pattern A, illustrates the resilience of edge computing in ensuring continued service availability.

The DevOps/SRE teams also become more important as QSRs are becoming more aware of using CI/CD pipelines to update software more quickly [36]. Such teams should be trained to implement and maintain distributed systems of sufficient scale with the complexities, so that continuous provisions in the order processing systems are guaranteed. The performance improvements will be critical based on upskilling IT departments to cope with edge infrastructures and CI/CD workflow.

5.4 Limitations of the Study

This study has several limitations when it comes to interpreting the results. The potentially simulated data set and small data sets used in the experiment might not accurately reflect the variability observed in a live QSR setting where customer behavior and network conditions are subject to change. Even though the model of synthetic trace generation was advantageous in stressing the architectures, the complexity of the real world, like the variation of the types of orders or distinct customer parameters, was not accounted for.

Another assumption that was based on standardized hardware configuration across the locations is that most of the QSR locations would have had different equipment and infrastructure, which would not be consistent. The differences in the quality of the network and labor practices between the regions can also affect the practicality of these findings, especially for franchises that are operating in other areas, where the stability of the network is different. The results of this research cannot necessarily be applied to full-service restaurants or non-QSR areas because these areas might have varied requirements of operation, customer expectations, and IT infrastructure. Prioritization on speed and volume, as in the case of QSRs, might not always be directly applied to other forms of restaurants, which have slower, more personalized service.

5.5 Implications for Digital Experience and SEO

The digital experience and SEO implications of an enhanced backend order processing through edge computing can be immense. Reduced page/API latency as a result of faster order processing has a direct effect on conversion rates in mobile apps and web ordering funnels. For example, a 50-70% reduction in the response time of the digital ordering process may result in an increase in customer retention and conversion rates because a customer will never give up and keep their orders because of a slow or unresponsive system.

Better uptime and responsiveness also help reduce cart abandonment rates, which can improve organic engagement signals such as time on site and repeat visits. Such signals play a key role in SEO because search engines have favored websites with easy, effective user interfaces [37]. With QSRs being more dependent on digital ordering methods in the future, stability in the quality of service offered by edge computing directly enhances the search capabilities of the “near me” and “order online” queries. Quality and speed of services also boost the possibility of a brand ranking in the local search results that are relevant and augment its presence in the online and competitive edge in the market.

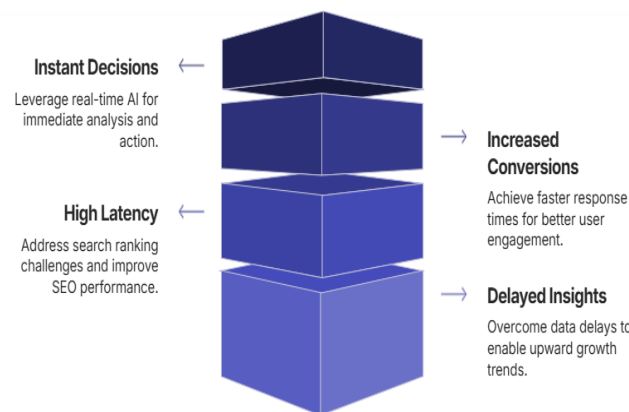


Figure 7: An overview of how real-time AI and reduced latency can enhance digital experiences, increasing conversions, improving SEO, and addressing delays in insights for better engagement.

Figure 7 above demonstrates a correlation between real-time AI and its effects on the SEO performance and conversion rates. The figure highlights the importance of real-time analysis, which is advanced by AI-driven instant decisions to better respond time and engage a user at the end. It emphasizes the fact that high latency needs to be considered to overcome the problem of search ranking to enhance the overall performance of SEO. QSRs can generate more conversions by fixing the poor pace of service delivery by reducing the information that is delayed during information processing by using a faster data processing speed [38]. This also enhances cart abandonment rates, which are some of the better indicators of engagement, as indicated by time on site and frequency of visits, which have a positive impact on local search results and general digital presence.

6. Future Research Recommendations

6.1 Integration with 5G and Multi-access Edge Computing (MEC)

Future studies should consider joint optimization of restaurant edge computing with telco Multi-access Edge Computing (MEC) to support ultra-low-latency interactions with third-party aggregators, such as DoorDash and Uber Eats [39]. When 5G technology is introduced, where latency is cut by several factors and throughput is much greater than current techniques, MEC and restaurant-based edge computing may be deployed together to promote overall optimization of QSR operations. MEC permits computational operations to be managed nearer to the consumer, enhancing the network by the performance of real-time offerings like order processing, inventory administration, and customer engagement. With the growing trend of real-time and high-paced interaction demands, the combination of 5G and MEC is going to be the key to providing a smooth communication process between QSRs and third-party applications such as delivery services.

The 5G ultra-low latency with the local treatment of the MEC can overcome one of the major challenges in QSRs, which is reducing the delays of service delivery in peak conditions. For example, with the 5G-powered MEC solutions alongside restaurant-based edge computing, it was possible to develop real-time order routing and dynamic delivery adjustments. It would guarantee more alignment between the kitchen, POS systems, and even the delivery services,

which would be reflected in faster and more effective delivery services of orders [40]. This integration, as it was proposed in the literature, needs to be tested in real-life situations later to gauge the effect this technology has on order flow, customer satisfaction, and delivery times and as well as assess the cost-benefit of this kind of implementation.

6.2 Edge AI for Predictive Demand and Dynamic Capacity Management

Another promising research area in the future is the use of Edge AI to make predictions about the demand based on forecasting and managing dynamic capacity. Developing machine learning (ML) models on the edge may add value to the decision-making process in QSRs, where restaurants will predict the changes in demand more precisely [41]. For example, predicting demand spikes during specific time windows, such as lunch hours or weekends, could enable QSRs to adjust staff schedules, kitchen throughput, and delivery resources in real time.

With the edge-based AI, QSRs might improve their processes based on the anticipated trends, which will reduce the waiting time and boost the throughput. ML models allow predicting a high probability of the rise of orders, based on past data, weather, or other local events, which will provide better control over staff scheduling and kitchen slotting. Additionally, these systems would be able to make changes dynamically to the flow of orders using real-time information, so that the kitchens and POS systems are continuously performing optimally [42]. Studies about this field explore how the algorithms of deep learning can be used to maximize inventory management, whereby more food can be saved by forecasting the specific amounts required at any point in time.

6.3 Multi-Tenant Edge Architectures for Franchise Networks

With the QSR industry still growing, particularly within the franchise model, possible future research efforts might explore the possibility of multi-tenant edge architecture in attempting to serve a multiplicity of franchise locations. The idea of shared edge clusters has the potential to bring major savings in costs since various locations within proximity may share the computing resources. However, this model presents the issues of data isolation, security, and the complexity of the CI/CD. Studies should be conducted on protecting the management of multiple franchises in the same edge infrastructure without a single sensitive information being exposed and influencing any other location.

Edge clusters might be shared across the borders of adjacent franchisees, cutting operational expenses incurred because they may share the resources, and also enjoy the low-latency advantages available to individual QSRs [43]. However, the fact that managing these distributed systems at scale demands proper isolation and security for each franchise makes this complicated, necessitating creative solutions. Research should be conducted on the adaptation of the CI/CD pipelines to updates in several locations at once, to maintain compatibility and minimize downtime amidst a network of edge nodes on a large scale. This will play an important role in considering franchise networks that aim to retain the same efficiency in operation without the need to interfere with the agility needed to update software at a faster rate.

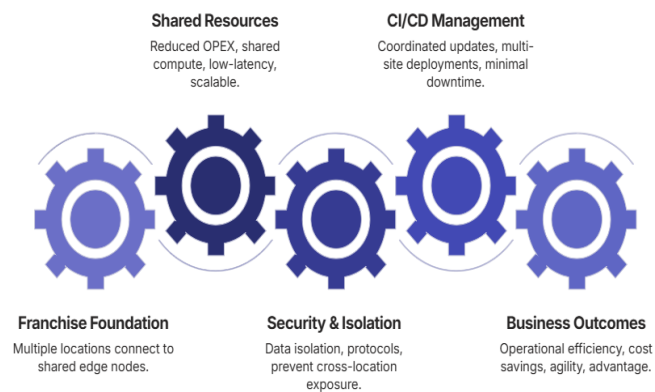


Figure 8: Key components of a multi-tenant edge architecture, emphasizing shared resources, security, CI/CD management, and business outcomes for franchise networks.

Figure 7 above shows the major elements of a multi-tenant edge network for franchise networks. It highlights common resources, allowing the same locations to be connected to edge nodes, allowing operational costs to be lowered and scalability to be improved. The infographic also emphasizes the relevance of CI/CD management in order to ensure coordination of updates and deployment across sites to minimize downtimes. Security and isolation are essential to provide protection of data in various franchise locations to avoid cross-location exposures. Business deliverables demonstrate the level of operational efficiency, cost savings, and agility in such architectures so that QSRs can remain scalable with flexibility in software updates and system administration.

6.4 Sustainability and Energy-Aware Edge Patterns

Since the QSRs are moving towards always-on edge clusters to provide the real-time order processing capabilities, the power capabilities of such systems and carbon footprint become a burning issue. Further studies should also involve the process of energy efficiency of edge computing systems without compromising the performance and achieving the Service Level Agreement (SLA). The edge computing devices may consume a lot of energy, particularly in areas with large volumes of food traffic, and this may greatly affect the environmental sustainability of the QSRs.

Research efforts should be made in balance where an edge cluster can consume energy, and at the same time meet the required performance levels, using energy-efficient hardware or using a dynamic power scaling technology. Besides, the incorporation of renewable energy sources, including wind or solar power, into the edge computing infrastructure can also contribute to decrement of the carbon footprint of QSRs [44]. Research work must also evaluate the implications on sustainability operations at the edge of the overall operations costs and the ability of its implementation in a cost-effective way. This study can investigate energy-efficient designs of computing that do not sacrifice the real-time order processing capacity needed by QSRs and meet sustainability goals.

7. Conclusions

The study aimed to examine how edge computing patterns and Continuous Integration/Continuous Deployment (CI/CD) pipelines could be utilized to minimize order flow and maximize the resilience of order processing systems in Quick-Service Restaurants (QSRs). The research was aimed at solving the issue of the QSRs during peak hours when the number of orders could overwhelm centralized cloud systems, which led to delays and lower quality of service. Through analysis of several edge computing architectures, this study sought to optimize the speed and efficiency of processing orders, making the service to the customers faster and more reliable. Practical, scalable solutions that could be readily implemented into the current QSR infrastructures were also proposed in the study with the aim of enhancing the accuracy of the orders and throughput, as well as customer satisfaction.

The findings of the experiments proved that the edge computing solutions, used in conjunction with the CI/CD pipeline, are, in significant ways, better than the cloud-only architectures. The main results are that P95 latency was decreased by up to 30-60%, guaranteeing the acceleration of the processing of orders, particularly in the high-demand seasons. Moreover, the throughput increased by 10-25%, and it made QSRs capable of managing more volume of orders without interfering with the speed or the accuracy of the service. That edge-based architectures can sustain high performance when the network is under stress was also demonstrated, and the error rates were reduced considerably during intervals of WAN degradation, discussing the resilience of edge computing in providing continuous service. Such gains were attained with little infrastructural modification, and this portrays how viable edge computing platforms are to the QSRs.

The results are added to the present source on edge computing and distributed systems within the QSR and retail industry. The research developed the theoretical knowledge of the ability of edge computing to reduce the effect of latency and improve the performance of a system in highly demanded settings on a real-time basis. Practically, the study is a useful source of reference architecture and deployment schemes that can be applied to large QSR chains, with special attention to real-time optimization of order flow. The realization of CI/CD practices together with edge computing opens the possibility of QSRs operating agile and scalable systems in the most rapid response to changing customer needs and operational demands. Edge computing, coupled with CI/CD pipelines, is a two-dimensional method that offers the future exploration of QSR IT infrastructure.

Pattern A (Local Queue and Caching Edge) is the type of edge computing solution that QSR practitioners should consider when they want to have robust and reliable edge computing behavior during network outages to address the needs of places with intermittent connectivity challenges. With the growing maturity of QSRs in their digital transformation and their capacity to handle DevOps, they ought to become more like Pattern B (Event-Driven Edge with Stream Processing), which is even more responsive to performance, because it dynamically handles the organization of the order flow and optimization of the throughput. This development enables QSRs to expand their operations without causing a decline in service quality when they are in high demand. In addition, QSRs are encouraged to use CI/CD pipelines to update their systems as quickly as possible and deploy with ease to cater to the dynamism and responsiveness of their systems to operational requirements.

With digital ordering steadily gaining relevance in the QSR sector, edge-native, CI/CD-friendly architectures will become the key to maintaining competitive service times, achieving better order accuracy, and a greater degree of customer satisfaction in general. Edge computing is an effective solution to the growing number of digital orders that QSRs can process, and at the same time, ensure high-quality service. With the advantages of edge computing, QSRs will be able to streamline their operations, enhance the resiliency of their operations, and provide faster, more dependable services, all of which play a vital role in retaining customers in an increasingly digital environment. In the future, it will be crucial to conduct additional studies on how to implement the newest technologies, like 5G and AI, or engage in sustainability efforts as a way of making the most of the opportunities that edge computing provides in the context of QSRs.

References;

- [1] Vidal, M. I. P. (2023). *Restaurant Brands International Field Lab Equity Research- Navigating the Quick Service Restaurant Landscape* (Master's thesis, Universidade NOVA de Lisboa (Portugal)).
- [2] Arpita, S. (2024). Supply Chain Process flow at Domino's Pizza Bangladesh.
- [3] Zulkarnain, N. A. A., & Sudin, A. M. (2024). Optimization of Waiting Time at Drive Thru Fast Food Restaurant using Queuing Model. *Enhanced Knowledge in Sciences and Technology*, 4(2), 134-140.
- [4] S. K. Gunda, "Analyzing Machine Learning Techniques for Software Defect Prediction: A Comprehensive Performance Comparison," 2024 Asian Conference on Intelligent Technologies (ACOIT), KOLAR, India, 2024, pp. 1-5, <https://doi.org/10.1109/ACOIT62457.2024.10939610>.
- [5] Samala, S. (2025). Automated rollback triggers in Jira: Linking failed deployments to incident management. *Computer Fraud & Security*. <https://computerfraudsecurity.com/index.php/journal/article/view/787>
- [6] Trigka, M., & Dritsas, E. (2025). Edge and cloud computing in smart cities. *Future Internet*, 17(3), 118.
- [7] George, A. S., George, A. H., & Baskar, T. (2023). Edge computing and the future of cloud computing: A survey of industry perspectives and predictions. *Partners Universal International Research Journal*, 2(2), 19-44.
- [8] Dhanagari, M. R. (2025). *Aerospike: The key to high-performance real-time data processing*. JISEM Journal. <https://www.jisem-journal.com/index.php/journal/article/view/8894>
- [9] Gunda, S. K. (2025). Accelerating Scientific Discovery With Machine Learning and HPC-Based Simulations. In B. Ben Youssef & M. Ben Ismail (Eds.), *Integrating Machine Learning Into HPC-Based Simulations and Analytics* (pp. 229-252). IGI Global Scientific Publishing. <https://doi.org/10.4018/978-1-6684-3795-7.ch009>.
- [10] S. K. Gunda, "Software Defect Prediction Using Advanced Ensemble Techniques: A Focus on Boosting and Voting Method," 2024 International Conference on Electronic Systems and Intelligent Computing (ICESIC), Chennai, India, 2024, pp. 157-161, <https://doi.org/10.1109/ICESIC61777.2024.10846550>.

- [11] Bourechak, A., Zedadra, O., Kouahla, M. N., Guerrieri, A., Seridi, H., & Fortino, G. (2023). At the confluence of artificial intelligence and edge computing in iot-based applications: A review and new perspectives. *Sensors*, 23(3), 1639.
- [12] Vishwakarma, S. K. (2025). Sustainable aviation fuel (SAF) procurement challenges. *Journal of Innovation and Sustainable Energy Management*. <https://www.jisem-journal.com/index.php/journal/article/view/9420>
- [13] Vismaya, K., & Thiyagarajan, C. (2024). Robotics and automation in commercial kitchens: Balancing efficiency and challenges. *International Journal of Research and Analytical Reviews*, 11(3), 245-258.
- [14] Bajpai, G., Schildmeijer, M., Mishra, M., & Piwosz, P. (2024). *CI/CD Design Patterns: Design and implement CI/CD using proven design patterns*. Packt Publishing Ltd.
- [15] Durgam, S. (2025). *Peer benchmarking systems for RIA performance evaluation in investment technology*. Computer Fraud & Security. <https://computerfraudsecurity.com/index.php/journal/article/view/785>
- [16] Kundel, R. (2024). *Accelerating Network Functions Using Reconfigurable Hardware: Design and Validation of High Throughput and Low Latency Network Functions at the Access Edge*. Springer Nature.
- [17] Amgothu, S. (2024). Innovative ci/cd pipeline optimization through canary and blue-green deployment. *International Journal of Computer Applications*, 186(50), 1-5.
- [18] Naqvi, S. K. H., & Batool, K. (2023). A comparative analysis between general data protection regulations and California consumer privacy act. *Journal of Computer Science, Information Technology and Telecommunication Engineering*, 4(1), 326-332.
- [19] Bradley, S. (2024). Comparative Analysis of Two Data Privacy Regulatory Schemes: The GDPR and the CCPA.
- [20] Zheng, X. (2024). *Project Management of a Fast Food Network Creating (on the Example of the "Culinary Crossroads" Project)* (Master's thesis, Lesya Ukrainka Volyn National University).
- [21] S. K. Gunda, "Enhancing Software Fault Prediction with Machine Learning: A Comparative Study on the PC1 Dataset," 2024 Global Conference on Communications and Information Technologies (GCCIT), BANGALORE, India, 2024, pp. 1-4, <https://doi.org/10.1109/GCCIT63234.2024.10862351>.
- [22] Iyengar, A., & Pearson, J. (2024). *Edge Computing Patterns for Solution Architects: Learn methods and principles of resilient distributed application architectures from hybrid cloud to far edge*. Packt Publishing Ltd.
- [23] Nagaraj, V. (2024). *Addressing power efficiency challenges in AI hardware through verification*. SciPubHouse. <https://scipubhouse.com/home/international-journal-of-sustainability-and-innovation-in-engineering-ijsie/content/ijsie-2024/addressing-power-efficiency-challenges-in-ai-hardware-through-verification/>
- [24] Ouamri, M. A., Alharbi, T., Singh, D., & Sylia, Z. (2025). A comprehensive survey on software-defined wide area network (SD-WAN): principles, opportunities and future challenges. *The Journal of Supercomputing*, 81(1), 291.

- [25] S. K. Gunda, "Automatic Software Vulnerability Detection Using Code Metrics and Feature Extraction," 2025 2nd International Conference On Multidisciplinary Research and Innovations in Engineering (MRIE), Gurugram, India, 2025, pp. 115-120, <https://doi.org/10.1109/MRIE66930.2025.11156601>.
- [26] Krishnan, R., & Durairaj, S. (2024). Reliability and performance of resource efficiency in dynamic optimization scheduling using multi-agent microservice cloud-fog on IoT applications. *Computing*, 106(12), 3837-3878.
- [27] Alelyani, A. (2024). Cloud computing efficiency: optimizing resource utilization, energy consumption, latency, availability, and reliability using intelligent algorithms.
- [28] Aguilar, A. (2023). Lowering Mean Time to Recovery (MTTR) in Responding to System Downtime or Outages: An Application of Lean Six Sigma Methodology. In *13th Annual International Conference on Industrial Engineering and Operations Management*.
- [29] Ugwueze, V. U., & Chukwunweike, J. N. (2024). Continuous integration and deployment strategies for streamlined DevOps in software engineering and application delivery. *Int J Comput Appl Technol Res*, 14(1), 1-24.
- [30] Abdulqader, A. F., Salih, M. M. M., Shaker, N. H., Sajid, W. A., Qasem, W., Gajewska, A., & Khlaponin, D. (2024, October). Optimizing IoT Performance Through Edge Computing: Reducing Latency, Enhancing Bandwidth Efficiency, and Strengthening Security for 2025 Applications. In *2024 36th Conference of Open Innovations Association (FRUCT)* (pp. 145-158). IEEE.
- [31] Masoumi, M., de Miguel, I., Brasca, F. G., Merayo, N., Lorenzo, R. M., & Barroso, R. J. D. (2024). Leveraging load balance metrics to unravel the impact of multi-access edge computing locations on online dynamic network performance. *IEEE Open Journal of the Communications Society*, 5, 5635-5651.
- [32] Gunda, S. K., Yalamati, S., Gudi, S. R., Manga, I., & Aleti, A. K. (2025). Scalable and adaptive machine learning models for early software fault prediction in agile development: Enhancing software reliability and sprint planning efficiency. *International Journal of Applied Mathematics*, 38(2s). <https://doi.org/10.12732/ijam.v38i2s.74>
- [33] Tahenni, A., & Merazka, F. (2023). SD-WAN over MPLS: A Comprehensive Performance Analysis and Security with Insights into the Future of SD-WAN. *arXiv preprint arXiv:2401.01344*.
- [34] Lopez, Y. A. C., Cueto, G. L., & Monjardin, C. E. F. (2025, April). Actual Quantification of 3PL (Third-Party Logistics) Costs in Food Distribution of QSR's (Quick Service Restaurants) in Quezon City. In *2025 IEEE Conference on Technologies for Sustainability (SusTech)* (pp. 1-6). IEEE.
- [35] Auten, L. (2023). *Vulnerability-Based Trust and Employee Engagement: A Phenomenological Study of Chick-Fil-A Franchisees' Leadership Qualities*. The Chicago School of Professional Psychology.
- [36] S. K. Gunda, "Comparative Analysis of Machine Learning Models for Software Defect Prediction," 2024 International Conference on Power, Energy, Control and

- Transmission Systems (ICPECTS), Chennai, India, 2024, pp. 1-6, <https://doi.org/10.1109/ICPECTS62210.2024.10780167>
- [37] Bansal, D. (2024). How SEO makes website loads faster and helps in user engagement. *International Journal For Multidisciplinary Research*, 6(2).
- [38] Rodriguez, J. M., Reambonanza, H., & Palallos, L. (2025). Optimizing Operation Processes and Supply Chain Management for Enhanced Service and Product Quality in Quick Service Restaurants. *The South East Asian Journal of Management*, 19(1), 10-7454.
- [39] Hou, L. (2024). *Efficient Offloading in Multi-Access Edge Computing Systems* (Doctoral dissertation, RMIT University Australia).
- [40] Jha, A. C. (2025). *AI-optimized spine-leaf fabrics: NVIDIA Quantum-2 vs. Cisco Nexus*. JISEM Journal. <https://www.jisem-journal.com/index.php/journal/article/view/13315>
- [41] Chae, B. K., Sheu, C., & Park, E. O. (2024). The value of data, machine learning, and deep learning in restaurant demand forecasting: Insights and lessons learned from a large restaurant chain. *Decision Support Systems*, 184, 114291.
- [42] Veer, B., & Aslam, S. (2024). Distributed Computing Techniques for Real-Time Data Processing in Mobile Restaurant Systems.
- [43] Sai Krishna Gunda (2024). Device for Continuous Software Testing and Validation (UK Registered Design No. 6400738). Registered with the UK Intellectual Property Office, Class 14-02, granted in November 2024.
- [44] Raza, M., Sakila, K. S., Sreekala, K., & Mohamad, A. (2024). ‘Carbon footprint reduction in cloud computing: Best practices and emerging trends. *Int. J. Cloud Comput. Database Manage*, 5(1), 25-33.