IRSTI 50.49.37
https://doi.org/10.53939/1560-5655_2026_1_166

Sychev Y. A.[1], Kim T. A.[2]

[1]Kaspi Bank JSC, Almaty c., Kazakhstan
[2]Department of "Finance and Accounting",
Sh. Murtaza International Taraz University, Taraz c., Kazakhstan

MODERNIZATION OF A SHOPPING MALL MARKETING DATA ANALYTICS SYSTEM: TRANSITION TO A .NET MICROSERVICE ARCHITECTURE AND QUANTITATIVE ASSESSMENT OF RESULTS

Abstract. The paper presents the modernization of a shopping mall marketing data analytics system originally implemented as a monolithic web application on .NET Framework with a single MS SQL Server database and batch integrations via file exchange. The audit identified limitations of the legacy solution a delay of more than 24 hours before data appeared in reports, performance degradation during peak periods, and high coupling of modules, where failures of individual components led to downtime of the entire platform. To reduce risks, an iterative Strangler Fig strategy was adopted with parallel operation of the monolith and new services. The target architecture is built on ASP.NET Core microservices with domain-based data isolation, the use of PostgreSQL for transactional data and Redis for hot metrics, as well as an event pipeline on Apache Kafka for near-real-time data processing and aggregation, followed by publishing indicators to BI dashboards. Implementation results demonstrate a reduction in data latency to seconds, increased resilience and release manageability, reduced MTTR to a few minutes, lower operating costs, and the economic feasibility of the project with payback within a 12-24 month horizon, including expansion of analytics through social media monitoring.
Keywords: Modernization, microservices, architecture, analytics, streaming; scalability.

Introduction. In modern retail, competitive advantage is shifting toward managing customer experience based on data. Effective shopping mall marketing largely relies on a detailed understanding of visitor behavior. The business needs to track traffic flows, identify "dead zones", and evaluate audience response to promotions in real time. However, many shopping malls still rely on legacy systems originally designed for nightly batch data processing. Hereafter, a legacy system is understood as an outdated monolithic platform.

Such an architecture creates a critical gap between business needs and IT capabilities. The main barrier is high latency. Reports and metrics arrive with a delay of a day or more, depriving marketing of the ability to influence the situation promptly. In this regard, many companies are moving to microservice architecture. This approach implies splitting an application into a number of small services, each of which performs its own narrow function. Such modernization brings significant benefits such as improved scalability, flexibility, and maintainability [1].

The analytical system considered in this work had similar shortcomings of legacy systems. In addition to high delay, the problem was caused by data fragmentation. Data from CRM, ERP, and traffic sensors were stored in isolated systems without modern APIs, which made it impossible to assemble a holistic customer profile. The system also demonstrated low flexibility. Vertical scaling could not handle peak loads during sales periods, and introducing new functions could take months due to the risk of destabilizing the code.

In effect, the system operated on data as of the previous day and was not able to provide information on what is happening now and what measures need to be taken immediately. This led to missed profit, especially during promotions and sales.

The aim of the study is to develop and implement a modernization strategy for such a legacy system to ensure real-time analytics and support new types of data. To achieve the goal, the following tasks were set:

- conduct an audit of the initial architecture and record its baseline indicators;
- justify the choice of a phased migration methodology and the target technology platform;
- implement the transition to a .NET microservice architecture using modern distributed-systems patterns;
- compare quantitative performance indicators of the system before and after modernization;
- demonstrate new analytical capabilities enabled by architectural changes (for example, analysis of social media data).

Below, the materials used and the modernization methodology are presented, then results and their discussion are provided, and finally the main conclusions are formulated.

Materials and Research methods. The object of the study was a corporate marketing analytics system operating as a monolithic web application on the old .NET Framework stack with a single MS SQL

Server 2008 database. This database became a bottleneck because it served both operational transactions (OLTP) and analytical queries (OLAP), provoking constant resource conflicts. Integration with external systems, for example CRM and visitor counters, was carried out through scheduled file exchange. Before modernization, an audit of the legacy system was performed and key metrics were measured. The average time from event occurrence to the appearance of information in reports exceeded 24 hours, since data was processed in batch mode only once per day. During peak loads, the response time for analytical queries exceeded 5 seconds, and timeouts often occurred. System reliability was low because failure of any module led to downtime of the entire platform; the average recovery time was about 2 hours. Development flexibility also suffered updates and deployment of new functionality occurred approximately twice a month, and even small changes took weeks to implement. These indicators confirmed the need for a fundamental reconstruction of the system.

For migration, an iterative approach based on the “Strangler Fig” pattern was chosen [2]. Unlike a radical rewrite of the system “from scratch” this method made it possible to gradually introduce a new architecture on top of the old one, progressively taking over the legacy system’s functions. The approach has proven effective in similar .NET application migration projects. The essence of the methodology is that a “facade” layer is created around the existing application, through which all requests pass. Then, step by step, individual functional domains are “cut out” of the monolith and re-implemented as autonomous microservices. At each stage, facade routing is updated - traffic is redirected to the new service, and the corresponding module in the old system is decommissioned.

Such incremental refactoring made it possible to avoid a long service interruption, and business users continuously received improvements as migration progressed, which increased trust in the project. In the first stage, the priority area was visitor traffic analytics for it, a new service was developed, after which all requests in this topic began to be served by it. Gradually, other key domains were migrated as well-loyalty programs, customer profiles and fundamentally new capabilities unavailable in the legacy system were introduced, for example a service for collecting and analyzing social media data. This order of work made it possible to immediately demonstrate the business value of modernization, turning the project from forced IT expenses into an investment in new capabilities [3].

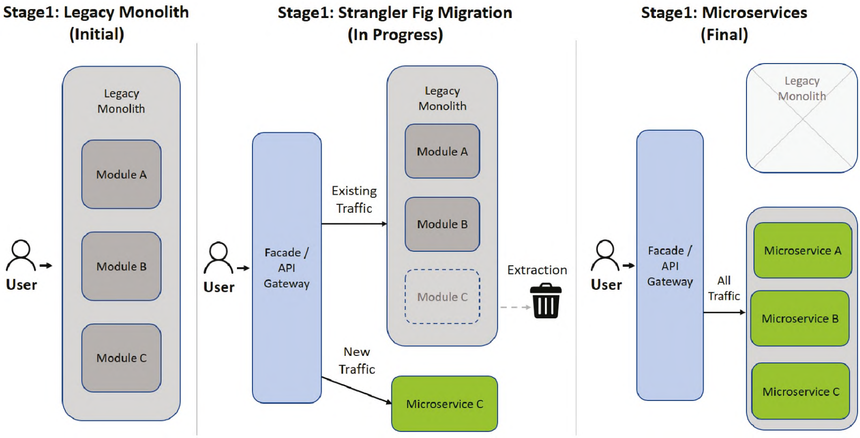

Figure 1 – Iterative migration using the Strangler Fig pattern

The new architecture was implemented as a set of cloud-independent microservices on ASP.NET Core. Each microservice corresponds to a separate area of business logic and its own data store. A combined approach was used for data storage. In particular, the transactional sales service uses a relational PostgreSQL database to ensure ACID properties of financial operations. To cache frequently requested analytical results, a high-performance Redis cluster is used, accelerating access to "hot" data, for example current sales indicators. All services interact via lightweight HTTP APIs or an asynchronous data exchange bus.

In addition to the OLTP microservices themselves, a separate pipeline was developed for near-real-time data processing. Its task is to offload operational services from heavy analytical computations and to ensure parallel processing of events as they arrive. The pipeline architecture includes several stages. All events are published to the distributed Apache Kafka message queue. Kafka can accept large event streams and guarantees reliable delivery. Next, at the processing stage, a special aggregator service subscribes to Kafka topics and performs transformation and enrichment of data. The computation results are stored in the main relational database PostgreSQL. This approach ensures load separation, where .NET microservices handle fast transactions, while a separate service in the background computes aggregates for analytics. At the final stage of data delivery, integration with a business intelligence tool was implemented, within which interactive dashboards were created for the marketing department.

They access either PostgreSQL directly to obtain up-to-date reports, or the Redis cache to access frequently updated metrics. Thus, the architecture combines the advantages of operational processing and deep analytics, meeting modern requirements of data-driven management.

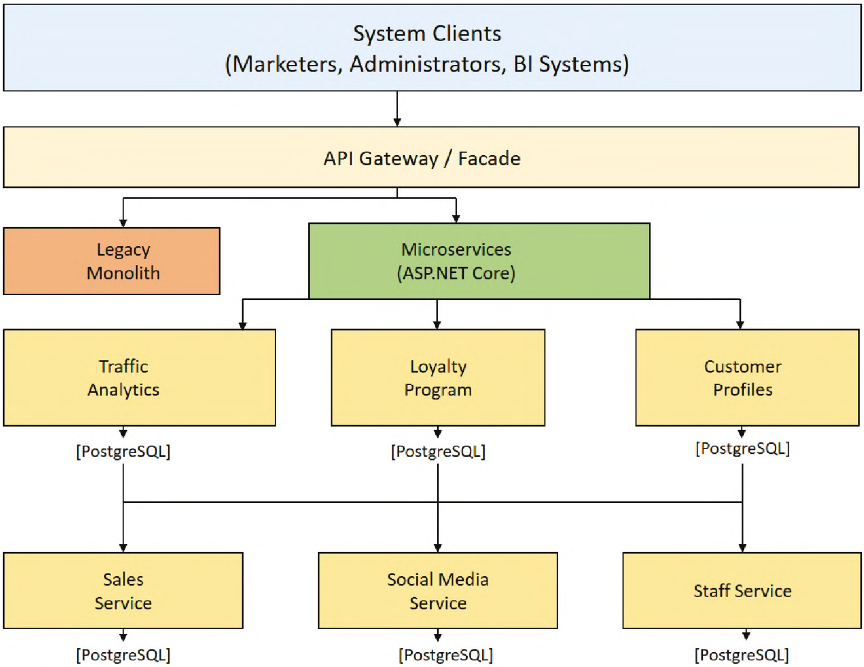

Figure 2 - Component interaction diagram

Results and discussion. Key system performance indicators improved after the transition to a microservice architecture. The time between data arrival and obtaining insight decreased from 24 hours to a few seconds, meaning analytics effectively became near real time.

Shopping mall management now has up-to-date dashboards reflecting the current situation rather than yesterday's results. The frequency of deploying updates increased from a few times per month to several times per week or even per day, which indicates increased development agility. The time to deliver a new report or metric to market decreased by approximately 80% from several weeks to several days. Due to the modularity of microservices, each change can be tested and deployed in isolation, which significantly reduced regression testing efforts and made it possible to respond faster to business requests. Platform reliability increased, if a separate service fails, it is restarted automatically due to Docker/Kubernetes cluster orchestration; the average recovery time decreased to 8 minutes. Failure of one microservice no longer brings down the entire system, as in the monolith other services continue to operate. Flexibility of horizontal scaling appeared – under increasing load, it is possible to dynamically add instances only of those services where it arose, without scaling the entire system unnecessarily. This had a direct effect on costs - support OPEX decreased by approximately 50% due to abandoning the legacy platform and optimal use of resources microservices consume exactly as many resources as their function requires.

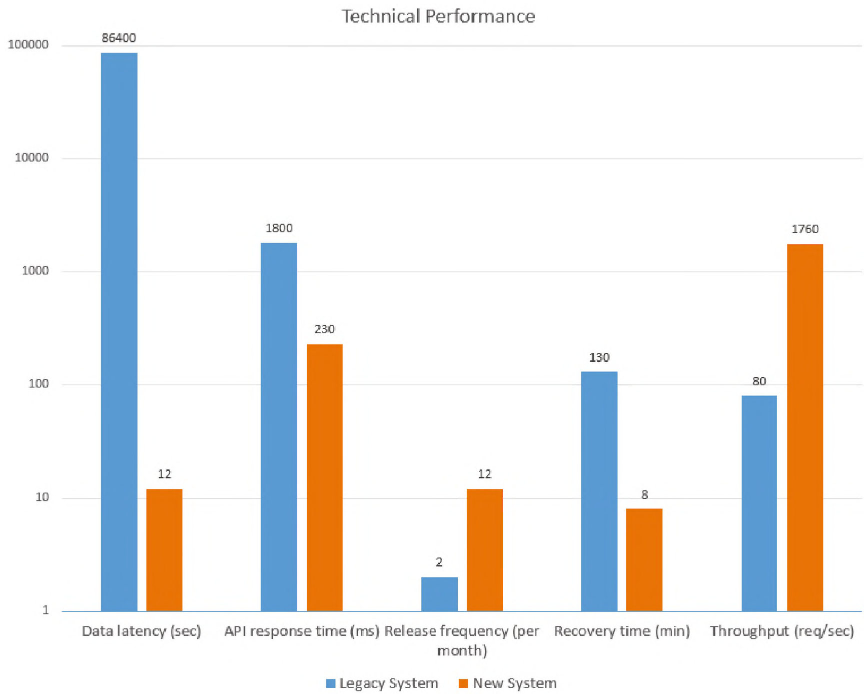

Figure 3 - Comparison of technical indicators of the Legacy System and the New System (data latency, API response time, release frequency, recovery time, throughput).

Table 1- Project budget

Expense item	Amount (USD)
Payroll (Senior Developer, 11 months)	36 440
Payroll (2 × Middle Developers Middle Developers, 11 months)	54 658
Cloud environment (Dev/Test)	7 845
License and software	4 290
Outsourcing	6 150

In addition, reducing the number and duration of system outages (by 60-80%) made it possible to reduce missed profit and costs associated with service unavailability. According to aggregated estimates, return on investment in modernization was achieved within 12-24 months of operating the new system.

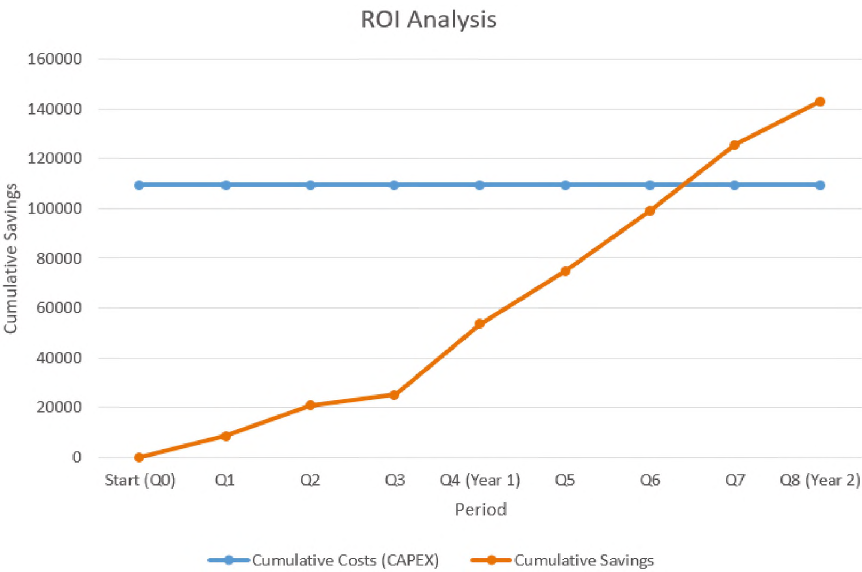

Figure 4 – ROI Analysis cumulative costs (CAPEX) and cumulative savings by periods (Q0-Q8).

These results are consistent with industry cases in which companies implementing modern architecture demonstrate improved scalability and a high benefit-to-cost ratio [4]. Also, teams or individual developers can work independently of each other [5].

The new architecture not only improved the indicators of the old system, but also created qualitatively new possibilities for marketing analytics. One such result was the introduction of collecting data from social networks [6]. Within the microservice platform, a separate service was developed that, in real time, collects public mentions of the shopping mall and stores them in a database. This capability made it possible to implement monitoring of customer opinion; for example, if several negative reviews about a certain aspect appear in social networks within a short period of time, the system automatically generates a notification for the mall's operational service. The marketing and administrative departments get the chance to respond during the customer's visit, without waiting for loss of loyalty, and can manually analyze the collected reviews [7], where interaction with the visitor occurs continuously based on data.

Another new direction of analytics was deeper analysis of buyer behavior. The legacy system could answer the question "how many sales were there yesterday in store X?" The new platform answers questions of a new type, for example "How did sales dynamics change in store X immediately after launching an advertising campaign?" or "Which products are most often bought together in one receipt?" Such data synthesis reveals valuable insights. Marketers can promptly identify dead zones» areas with low traffic and immediately take measures, for example relocating advertising or events to those zones. In addition, the ability appeared for flexible management of rental rates based on actual foot traffic. These examples illustrate how the modernized architecture became an instrument for implementing advanced marketing analysis methods that were previously impossible. Moreover, the obtained results are consistent with trends in scientific research, according to which social media data are successfully applied to assess the attractiveness of urban commercial spaces [8]. Our system provides a technological foundation for implementing such analytical approaches in practice.

Conclusion. The conducted study demonstrated that the legacy architecture was a serious obstacle to implementing an operational, data-oriented marketing strategy in the shopping mall. Applying the phased Strangler Fig migration methodology made it possible to successfully replace the legacy system with a modern. NET microservice platform without critical downtime and risks. Key architectural decisions splitting the application into microservices, isolating each service's data in its own database, and using diverse storage technologies depending on the nature of the data provided the required flexibility and scalability of the system. The new architecture is capable of effectively handling both transactional loads and complex analytical queries in real time, thereby eliminating the previously existing trade-off between responsiveness and depth of analysis.

Quantitative outcomes confirm the feasibility and effectiveness of modernization. Improvement of the main KPIs was noted - data latency decreased by an order of magnitude, recovery time decreased by more than 80%, release frequency increased sixfold, and horizontal scaling made it possible to eliminate performance bottlenecks. Economic analysis showed that reducing operating and infrastructure costs, as well as preventing losses from downtime, allowed recovering the invested funds in less than two years of operating the new system. These results are in line with global digital transformation trends. Transition to cloud microservices gives organizations a significant increase in efficiency and business performance [5].

The practical value of the work lies in the fact that the presented architectural strategy and the obtained metrics can serve as a guideline for other enterprises in retail and commercial real estate facing the problem of legacy systems. The modernization described in the paper demonstrates how modern technologies make it possible to put into practice concepts from current scientific literature. In particular, the work shows that modern IT architecture acts as a necessary catalyst for implementing proactive marketing approaches, such as digital customer guidance [4] and intelligent big data analytics [7].

References

1 *Hossain M. D., Sultana T., Akhter S., Hossain M. I., Ngo T. T., Huynh L. N. T., Lee G.-W., Huh E.-N.* The role of microservice approach in edge computing: Opportunities, challenges, and research directions // ICT Express. – 2023. – Vol. 9. – No. 3. – P. 1162–1182. – DOI: 10.1016/j.icte.2023.06.006.

2 *Fowler M.* Strangler fig application [Electronic resource]. – 2004. – URL: http://martinfowler.com/bliki/StranglerApplication.html (accessed: 18.02.2026).

3 *Sychev Y.* AN APPLICATION OF ARTIFICIAL INTELLIGENCE SYSTEMS IN APPLICATION ARCHITECTURE DESIGN: FROM REQUIREMENTS TO IMPLEMENTATION // Universum: Technical Sciences: electronic scientific journal. – 2025. – No. 9(138).

4 PingCap. Real-time vs batch processing: A comprehensive comparison for 2025 [Electronic resource]. – 2025. – URL: https://pingcap.com/article/real-time-vs-batch-processing (accessed: 18.02.2026).
5 *Li X., Xu M., Zeng W., Tse Y. K., Chan H. K.* Exploring customer concerns on service quality under the COVID-19 crisis: A social media analytics study from the retail industry // Journal of Retailing and Consumer Services. – 2023. – Vol. 70. – P. 103157. – DOI: 10.1016/j.jretconser.2022.103157.
6 *Tumanov O.* Aspects of using social media in research // Scientific Bulletin of the National Academy of Statistics, Accounting and Audit. – 2019. – No. 4. – P. 24–29. – DOI: 10.31767/nasoa.4.2019.03.
7 *Olshanetskaya O. A., Kotova M. R.* DIGITAL CUSTOMER GUIDANCE IN SHOPPING MALLS: STRATEGY AND IMPLEMENTATION METHODS // Industrial Economy. – No. 5. – P. 219–227. – DOI: 10.47576/2949-1886.2025.5.5.028.
8 *Zhang Y., Wang M., Yang X., Zhang R.* Urban commercial space vitality evaluation method based on social media data: The case of Shanghai // Land. – 2025. – Vol. 14. – No. 4. – P. 1–24. – DOI: 10.3390/land14040697.

Сычев Ю.А. [1]**, Ким Т. А.** [2]
[1]АО «Каспи Банк», Аға инженер-программашы, Алматы қ., Қазақстан
[2]Тараздағы Ш. Мұртаза атындағы Халықаралық университет, «Қаржы және бухгалтерлік есеп» кафедрасының оқытушысы, Тараз қ., Қазақстан

САУДА ОРТАЛЫҒЫНЫҢ МАРКЕТИНГТІК ДЕРЕКТЕРІН ТАЛДАУ ЖҮЙЕСІН ЖАҢҒЫРТУ: .NET МИКРОСЕРВИСТІК АРХИТЕКТУРАСЫНА КӨШУ ЖӘНЕ НӘТИЖЕЛЕРДІ САНДЫҚ БАҒАЛАУ

Түйіндеме. Мақалада бастапқыда .NET Framework платформасында біртұтас монолит веб-қосымша ретінде іске асырылған, бір MS SQL Server дерекқорын және файл алмасу арқылы орындалатын пакеттік интеграцияларды пайдаланған сауда орталығының маркетингтік деректерін талдау жүйесін жаңғырту сипатталады. Аудит ескірген шешімнің шектеулерін анықтады: есептерде деректер пайда болғанға дейін 24 сағаттан астам кідіріс, ең жоғары кезеңдердегі өнімділіктің деградациясы және жеке компоненттердің істен шығуы бүкіл платформаның тоқтап қалуына алып келетін жоғары модульдік байланыс. Тәуекелдерді азайту үшін монолит пен жаңа қызметтердің параллель жұмысымен Strangler Fig итеративті стратегиясы таңдалды. Нысаналы сәулет доменге бағытталған деректерді оқшаулау ASP.NET Core микросервистеріне, транзакциялық деректер үшін PostgreSQL және «ыстық» көрсеткіштерге Redis, сондай-ақ нақты уақытқа жақын режимде деректерді өңдеу және біріктіру үшін Apache Kafka-дағы оқиға конвейеріне, содан кейін BI бақылау тақталарында көрсеткіштерді жариялауға негізделген. Іске асыру нәтижелері деректер кідірісінің секундтарға дейін қысқарғанын, тұрақтылық пен релиздерді басқарудың жақсарғанын, MTTR көрсеткішінің бірнеше минутқа дейін төмендегенін, операциялық шығындардың азайғанын және

әлеуметтік желілер мониторингі арқылы аналитиканы кеңейтуді қоса алғанда, жобаның 12–24 айлық көкжиекте өзін ақтайтынын көрсетеді.
Түйінді сөздер: жаңғырту, микросервистер, архитектура, аналитика, ағындық өңдеу, ауқымдылық.

* * *

Сычев Ю.А.[1], Ким Т. А.[2]
[1]АО «Каспи Банк», г. Алматы, Казахстан
[2]Международный университет им. Ш. Муртазы, кафедра «Финансы и бухгалтерский учет», г. Тараз, Казахстан

МОДЕРНИЗАЦИЯ СИСТЕМЫ АНАЛИТИКИ МАРКЕТИНГОВЫХ ДАННЫХ ТОРГОВОГО ЦЕНТРА: ПЕРЕХОД К МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ .NET И КОЛИЧЕСТВЕННАЯ ОЦЕНКА РЕЗУЛЬТАТОВ

Аннотация. В статье представлена модернизация системы аналитики маркетинговых данных торгового центра, изначально реализованной как монолитное веб-приложение на .NET Framework с единой базой данных MS SQL Server и пакетными интеграциями через файловый обмен. Аудит выявил ограничения устаревшего решения: задержку более 24 часов до появления данных в отчеатах, деградацию производительности в пиковые периоды и высокую связанность модулей, при которой сбои отдельных компонентов приводили к простоям всей платформы. Для снижения рисков была выбрана итеративная стратегия Strangler Fig с параллельной работой монолита и новых сервисов. Целевая архитектура построена на микросервисах ASP.NET Core с доменно-ориентированной изоляцией данных, использованием PostgreSQL для транзакционных данных и Redis для “горячих» метрик”, а также событийным конвейером на Apache Kafka для обработки и агрегации данных в режиме, близком к реальному времени, с последующей публикацией показателей в BI-дашбордах. Результаты внедрения демонстрируют сокращение задержки данных до секунд, повышение отказоустойчивости и управляемости релизов, снижение MTTR до нескольких минут, уменьшение операционных затрат и экономическую целесообразность проекта с окупаемостью в горизонте 12-24 месяцев, включая расширение аналитики за счет мониторинга социальных сетей.
Ключевые слова: Модернизация, микросервисы, архитектура, аналитика, потоковая обработка, масштабируемость.

Information about the authors

Sychev Yegor Aleksandrovich – Senior Software Engineer, Kaspi Bank JSC, Almaty c., Kazakhstan; sigmadel10@gmail.com

Kim Tatiana Alekseevna – Master of Economics, Lecturer;

tanechka-kim@internet.ru

Авторлар туралы мәліметтер

Сычев Егор Александрович – Senior Software Engineer, АО Kaspi Bank JSC, Алматы қ., Қазақстан; sigmadel10@gmail.com

Ким Татьяна Алексеевна – экономика магистрі, оқытушы;

tanechka-kim@internet.ru

Сведения об авторах

Сычев Егор Александрович – Senior Software Engineer, АО Kaspi Bank,

г. Алматы, Казахстан; sigmadel10@gmail.com

Ким Татьяна Алексеевна – магистр экономики, преподаватель,

tanechka-kim@internet.ru