https://orcid.org/0000-0002-9374-2833
https://orcid.org/0000-0001-6559-1508

UDC 004.273

H.T. SAMOYLENKO*, A.V. SELIVANOVA*

DISTRIBUTED INFORMATION SYSTEMS IN E-COMMERCE

*State University of Trade and Economics, Kyiv, Ukraine

Анотація. *У статті розглянуто основні вимоги до інформаційних систем електронної торгівлі, що підтримують ведення бізнесу. Охарактеризовано особливості модульного проєктування інформаційних систем електронної торгівлі, визначено переваги та недоліки незалежно від розроблених інформаційно-обчислювальних ресурсів. Обґрунтовано доцільність використання розподілених інформаційних систем для задач електронної торгівлі. Концепція розподілених інформаційних систем передбачає використання різних технологій та протоколів для забезпечення доступності, надійності й масштабованості системи. Архітектура розподіленої інформаційної системи передбачає створення системи з розподіленими компонентами, які взаємодіють за допомогою стандартних інтерфейсів та використовують різні технології для комунікацій. Визначено перспективи застосування розподілених інформаційних систем, проаналізовано переваги застосування розподіленої архітектури. У статті проаналізовано етапи побудови архітектури розподіленої інформаційної системи, визначено її основні складові. Архітектура розподілених систем може складатися з таких складових, як сервери баз даних, веб-сервери, додатки, засоби забезпечення безпеки, мережеве обладнання і може варіюватися залежно від конкретної системи та її потреб. Визначено типи архітектур розподілених інформаційних систем, специфіку та особливості їх застосування. У статті розглянуто мікросервісну орієнтовану архітектуру (Microservices-Oriented Architecture, MOSA), основна ідея якої полягає в тому, що програмне забезпечення розбивається на невеликі автономні мікросервіси, що взаємодіють між собою за допомогою API. Застосування MOSA для інформаційних систем електронної торгівлі дозволяє підвищити швидкість розробки та впровадження додаткових функцій, забезпечує масштабованість та стійкість до відмов.*
Ключові слова: *системи електронної торгівлі, розподілені інформаційні системи, архітектура інформаційних систем.*

Abstract. *The article discusses the basic requirements for electronic commerce information systems that support business. The features of the modular design of electronic trade information systems are characterized and the advantages and disadvantages of independently developed information-but-computational resources are determined. The expediency of using distributed information systems for electronic trade tasks is justified. The concept of distributed information systems involves the use of various technologies and protocols to ensure the availability, reliability, and scalability of the system. The architecture of a distributed information system involves the creation of a system with distributed components that interact using standard interfaces and use various technologies for communications. The prospects for the use of distributed information systems are determined and the advantages of using a distributed architecture are analyzed. The article studies the stages of building the architecture of a distributed information system and defines its main components. The architecture of distributed systems can consist of such components as database servers, web servers, applications, security tools, and network equipment, and may vary depending on the specific system and its needs. The types of architectures of distributed information systems and the specifics and features of their application are determined. The article discusses microservices-oriented architecture (Microservices-Oriented Architecture, MOSA), the basic idea of which is that software is divided into small, autonomous microservices that interact with each other using APIs. The use of MOSA for electronic trade information systems allows for increasing the speed of development and implementation of additional functions and ensures scalability and resistance to failures.*
Keywords: *e-commerce systems, distributed information systems, architecture of information systems.*

DOI: 10.34121/1028-9763-2023-2-69-74

1. Introduction

In a fairly short period of time, e-commerce has become a phenomenon that began to claim one of the leading roles in business and economic activity in general. The use of distributed information systems for solving e-commerce tasks makes it possible to reduce the load on servers and ensure the work of geographically remote units. Currently, Microservices-Oriented Software Architecture (MOSA) is gaining particular popularity; it is an architectural approach used for software development when the application is divided into many small independent modules.

The aim of the article is to analyze the specifics of microservice architecture for e-commerce information systems and to study the problems arising in the course of using this approach.

2. Results of the research

The creation of effective information systems for e-commerce should be based on a system approach, modern models and methods of information systems (IS) design [1]. The main requirements for trade information systems that ensure business conduct are the following:

1. Efficiency and functionality. Information systems of electronic trade must provide buyers with the necessary information about goods and services, support electronic means of payment, use the means of Internet advertising, etc.

2. Operation in real-time mode.

3. Scalability. The main tasks of a trading company are to increase turnover and profits, develop competitiveness, expand the client base, increase efficiency and reduce costs, and expand its business.

4. High performance is one of the main requirements for IS since the flow of customers in electronic commerce is much higher than in most offline stores.

5. Flexible structure and adaptability. Regardless of the chosen option, the stages of IS implementation, its adaptation and integration with business processes, employees, and resources of a specific e-commerce enterprise are mandatory.

6. Reliability. Information systems must ensure the reliable operation of e-commerce enterprises.

7. Information security is one of the key issues of e-business organizations, to which considerable attention is paid in almost all spheres connected with e-commerce. Large online stores often encounter various types of violations aimed at stopping their normal functioning or stealing confidential customer data.

For the practical implementation of these requirements, it is important to consider the design problems of electronic trade information systems. Very often, when developing a new information system, there are already implemented functional modules that have already been implemented and have shown their effectiveness in practice. A problem in modular design can be the incompatibility of individual components. In this case, a set of mechanisms is used that will make such independently developed information and computing resources compatible. This distribution of data allows, for example, to store in a network node the data that are most often used in this node. This approach makes it easier and faster to work with this data, leaving the opportunity to work with other data in a common database. If the information system supports consistent storage of information in several files, it can be said that it supports a database. If the supporting data management system allows working with several files, such a system should also have some own data (metadata) and even knowledge that determines the integrity of the data. A system in which more than one database server functions is usually considered distributed. It is used to reduce the load on the server and ensure the operation of geographically remote units. To ensure the reliability and scalability of the system, the concept of a distributed information

system involves the use of various technologies and protocols to ensure the availability and protection of the system. Ensuring security and data protection in distributed systems is also an important component. For this, various mechanisms such as encryption, user authentication and authorization, data access control, etc. are used. So, the concept of building a distributed information system architecture involves creating a system with distributed components that interact using standard interfaces and use various technologies to ensure system availability, reliability, scalability, and security.

Distributed information systems (DIS) have great potential for future development. Below, there are some perspectives of DIS development:

• expanding the scale of DIS – the development of cloud computing and virtualization technologies allows you to increase the scale of DIS and provide more efficient work with a large amount of data;

• increasing the reliability of DIS – the development of new methods of protection against intrusions and emergencies in DIS allows for a higher level of system reliability and security;

• the development of the Internet of Things – distributed information systems can be used to collect and process data from various Internet of Things devices, which gives a lot of opportunities for process automation and optimization of various production processes;

• the development of innovative methods of data processing – the development of technologies of artificial intelligence, machine learning, and data analysis allows you to create new methods of data processing and analysis in DIS, which allows for more accurate prediction and management of processes in DIS;

• the development of blockchain technologies – the use of blockchain technologies in DIS can ensure a higher level of security and reliability of interaction between various elements of the system.

There are several advantages of DIS. They are the following:

• scalability – distributed information systems can be easily scaled depending on the changing needs of users, so they can be used to serve a large number of users;

• reliability – in case of failure of one server, other servers can accept the load and continue working without interruption in providing services;

• distribution – data can be distributed between several servers, which allows for the efficient use of resources and reduction of data access time;

• security – distributed information systems can be provided with additional security measures to ensure protection against unauthorized access and hacking;

• flexibility – distributed information systems can be easily changed or updated without interruption in the provision of services, which allows companies to quickly respond to changes in market conditions and user needs;

• distributed information systems allow for more efficient use of resources, increase productivity and provide greater reliability and security in comparison with centralized information systems.

3. Materials and methods

The architecture of a distributed information system is a set of principles, standards, and approaches to the design, development, and operation of systems that allow them to work as a single integrated complex, consisting of several components located on different physical machines, networks, or different locations [2, 3]. The main characteristics of the architecture of distributed IS are distributed independence, transparency, redundancy, scalability, and security. The construction of the architecture of the distributed information system consists of the following stages:

1. Definition of system requirements. At this stage, it is necessary to define and describe what business processes the system should support, what functions should be implemented, what

data should be stored, and what characteristics regarding reliability, scalability, and other parameters should be taken into account.

2. Architecture design. At this stage, the general concept of the system is created and the main components that it should include are determined. The selection of technologies and platforms on which the system will be built is carried out as well.

3. Prototyping. At this stage, a prototype of the system is created with minimal functionality to check whether it meets the customer's requirements and to make sure that the selected technologies and platforms are suitable for the system development.

4. Development. At this stage, the complete development of the system is performed, including all unnecessary functions and components.

5. Testing and validation. At this stage, the system is tested at various stages of development to ensure that it works correctly and meets the customer's requirements.

6. Release and support. At this stage, the system is put into operation and further support is provided, including fixing bugs and adding new features that arise over time.

Distributed IS can include various components such as database servers, web servers, applications, security tools, network equipment, etc. These components can be located on different physical machines connected to the network and managed by a centralized or decentralized management system [4].

The architecture of distributed IS consists of several components that may vary depending on the specific system and its needs. However, the main components of DIS can be the following:

– client-server architecture: this architecture is based on the interaction between the client and the server. The client component (the client) interacts with the server to obtain the resources and services provided by the server component. The server component usually resides on a centralized server that provides services to many clients;

– peer-to-peer (P2P) architecture: in this architecture, components of a distributed information system have equal rights and can interact with each other without using a centralized server. Components can share resources and process tasks together;

– service-oriented architecture (SOA): this is an architecture in which system components are services provided to other components over a network. GIS components can be built with different technologies and programming languages, but they can interact with each other through standardized services. SOA is an approach to the design and development of software in which system functionality is organized as a set of services that can be used to interact with other systems or components. In SOA, each service provides clearly defined interaction capabilities that are described using interface standards such as WSDL (Web Services Description Language) or REST (Representational State Transfer). Clients can use these services by calling the corresponding interfaces, which allows them to obtain the necessary data and perform operations. The advantages of SOA include the increased reuse of software components, reducing the complexity of the software by dividing it into smaller components, and increasing system flexibility;

– microservice architecture: this is an architecture in which system components are in the form of individual microservices that can be deployed and scaled individually. Each microservice performs its task and interacts with other microservices over a network. Microservices-Oriented Architecture (Microservices-Oriented Architecture, MOSA – Microservices-Oriented Software Architecture) is an approach to software development based on the creation of several small, independent and interchangeable components that provide separate functions. As a rule, these components are lightweight, modular, and distributed horizontally, i.e. the same functionality can be implemented by several different services. MOSA technologies are also called “mosquito technologies”; the main idea of MOSA is that software is broken down into small, autonomous microservices that can interact with each other using APIs. Each microservice can be designed, tested, and executed independently of other microservices, allowing developers to design, test,

and release new features more efficiently. With MOSA, developers can reduce the size and complexity of software, increase the speed of development and release of new features, and provide greater scalability and fault tolerance [4];

– mixed architecture: it is a combination of different component architectures (usually client-server and SOA (Service-Oriented Architecture)) used to achieve certain business goals. In the combined architecture, each component of the system performs its function, and the interaction between the components is carried out through a network. For example, distributed ICs for e-commerce may use elements of a client-server architecture to provide interaction between a client and a server for order processing and payment. At the same time, SOA elements can be used in the system to provide access to the catalog of goods and various services, for example, the delivery of goods or payment through third-party payment systems. The combined architecture can provide greater flexibility and scalability of the system, as well as ensure optimal use of resources and adapt to changes in business requirements and technological trends. However, it is worth considering that the combination of different architectures can lead to the complexity of the system and the emergence of problems in ensuring its security and stability.

4. Designing

Currently, Microservices-Oriented Software Architecture (MOSA) is gaining particular popularity – this is an architectural approach used to develop software, where the application is divided into many small independent modules that can interact with each other through APIs. Each module is a separate microservice that has its own data and business logic. The main idea behind MOSA is to break an application into smaller, simpler, and lighter parts that can be deployed and scaled individually. Each microservice can be written in different programming languages and deployed on different servers. This allows you to reduce dependencies between various system components and make development and maintenance simpler and more efficient. The main advantages of MOSA are increasing the scalability and resilience of the system, reducing the time of deployment and development, and ensuring a more flexible and quick response to changes in business requirements. However, MOSA can also be more complex to develop and test, as it needs to ensure proper interaction between different microservices and address duplication of functionality. MOSA is a set of small, self-contained services that have a single responsibility and are deployed individually. Each service is a separate program that must include functionality for a certain isolated business purpose. Communication between services usually takes place using a REST API [5]. MOSA has a number of significant advantages. They are the following:

1) free choice of programming language and data storage – this approach allows you to use different programming languages depending on different business needs;

2) reliability – in case of failure of one service, others will continue to work, and the point of failure is much easier to find and eliminate;

3) microservices use the capacities they need, which optimizes the use of available resources and saves customers;

4) the process of introducing microservices is much faster because they have a small code base and are compiled faster, all links pass faster, including testing, which, in its turn, allows you to make changes faster and safer.

Let’s consider the design of IS architecture for the operation of a conventional online store. The direct functioning of the online store can be described step by step as follows:

• the buyer goes to the website of the online store, searches for the product, and sends it to the shopping cart;

• when placing an order, the client indicates their contact details and payment method;

• the manager contacts the client to confirm the order and specify contact data and the place where the goods should be delivered;

• the manager of the online store or another employee packs the product and sends it through a courier service (if the customer does not prefer self-delivery);

• the client receives the order and makes the payment;

• further interaction with the buyer (cross-sell, up-sell, advertising, and email marketing).

According to the described operation of a conditional online store, it is possible to divide the future application into the following microservices: goods service (goods-service), category service (category-service), statistics service (statistics-service), API gateway service (gate-way-service), and registration service (registration-service) (Fig. 1). The client part can be developed separately.

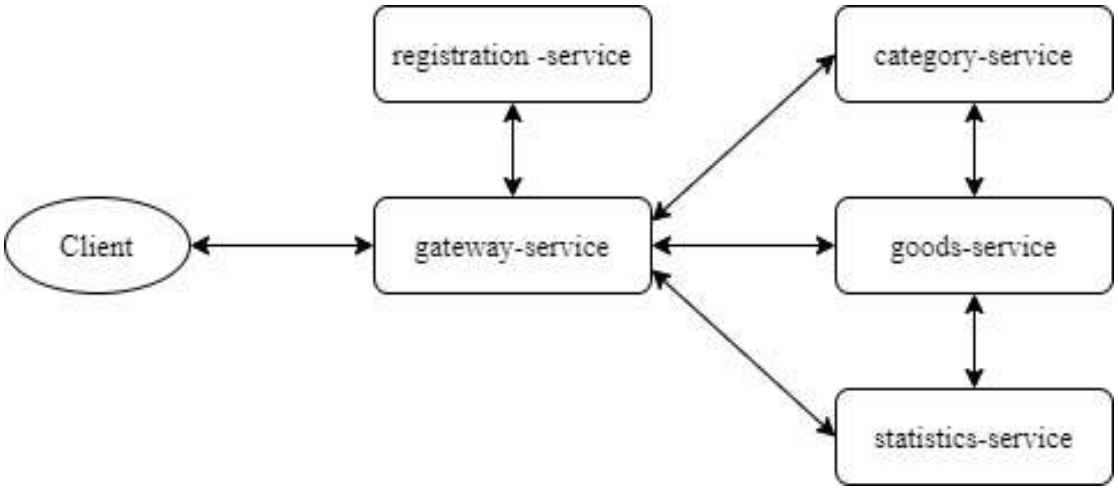

Figure 1 – Proposed application architecture

5. Conclusions

The article examines the advantages and disadvantages of using distributed information systems and analyzes the stages of building a distributed information system architecture. Various architectures of distributed information systems and their components, which may vary depending on the specific system and its needs, are characterized. To reduce dependencies between various system components and make the development of distributed e-commerce systems simpler and more efficient, it is proposed to use Microservices-Oriented Software Architecture, whose concept is to divide the application into many small independent modules, each of which is a separate microservice. This approach makes it possible to speed up the implementation and simplify the further support of the distributed information system.

REFERENCES

1. ISO/IEC 15288:2008 - System life cycle processes. URL: https://www.iso.org/standard/43564.html.
2. Paulk M.C., Weber C.V., Curtis B., Chrissis M.B. et al. The Capability Maturity Model: Guidelines for Improving the Software Process. Boston: AddisonWesley, 2015. 456 p.
3. Гломозда Д.К. Проектування, системний аналіз і розробка корпоративних інформаційних систем: навчальний посібник. Київ: НаУКМА, 2015. 95 с.
4. Pattern: Monolithic Architecture. URL: https://microservices.io/patterns/monolithic.html.
5. Amazon API Gateway. URL: https://aws.amazon.com/api-gateway/.

Стаття надійшла до редакції 06.04.2023