UNIVERSITY OF ECONOMICS - VARNA
FACULTY „INFORMATICS“
DEPARTMENT „INFORMATICS“

Jordan Ivanov Jordanov

Cloud-based information system for managing customer orders in a manufacturing enterprise

ABSTRACT

of a dissertation thesis for acquiring

of the educational and scientific degree "doctor" in professional direction 4.6. Informatics and Computer Science, doctoral program "Informatics"

Research supervisor: Prof. DSc Pavel Petrov

Varna
2025

The dissertation is 180 pages long, including 44 figures, 20 tables, 4 appendices. The bibliography covers 186 literary sources.

The main results of the research were presented at scientific conferences and published in collections of reports (1 items), scientific journals (3 items).

The dissertation work was discussed and directed for defense before a scientific jury at a meeting of the "Informatics" department at the "Informatics" faculty of the University of Economics - Varna on 12.06.2025.

Scientific jury:

1. External members
   - Prof. Dr. Georgi Georgiev Dimitrov, UniBIT
   - Prof. Dr. Miroslav Nikolov Galabov, VTU
   - Prof. Dr. Eng. Teodora Bakardjieva, VSU
2. Internal members
   - Prof. Dr. Yulian Andreev Vasilev, UE-Varna
   - Assoc. Prof. Dr. Ivan Ognyanov Kuyumdzhiev, UE-Varna

The defense of the dissertation will take place on ..... at .... hours in the hall .... of the University of Economics - Varna at a meeting of the Scientific Jury appointed by Order No. ..... of the Rector of the University of Economics - Varna.

The defense materials are available to those interested on the website of the University of Economics - Varna, www.ue-varna.bg

I. GENERAL CHARACTERISTICS OF THE DISSERTATION

1. The importance of the research

In the contemporary business environment, manufacturing enterprises face increasingly high demands from business clients for efficiency and adaptability. The implementation of cloud technologies emerges as an approach for optimizing communication within organizational structures, maintaining relationships with business clients, and fulfilling dynamically changing market expectations. The relevance of the studied topic is determined by the trend of cloud technologies becoming a strategic tool for future growth, modernization, and digital transformation within manufacturing enterprises. This trend is expected to continue as cloud platforms enable relatively rapid implementation of innovative ideas.

In the context of the Fourth Industrial Revolution, characterized by the widespread application of digital technologies in manufacturing, several problems have arisen:

- the need for compatibility and interaction between various internal and external information systems,
- ensuring cybersecurity and reliable mechanisms for data exchange,
- the necessity for dependable storage and processing of large volumes of data,
- compliance with local regulations in global operations.

Historically, logistics and supply chain management were based on complex, sometimes manually managed processes involving multiple participants - from dispatchers, suppliers, and carriers to end consumers. Enterprise resource planning (ERP) and supply chain management (SCM) systems, established in the 1990s, sometimes operate separately from other informational subsystems within an enterprise, which hinders communication and information exchange between different departments and partners. This isolation obstructs the enhancement of information entry and processing procedures while creating challenges for effective business process management.

The development of cloud technologies, particularly the introduction of models Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), and Software-as-a-Service (SaaS), opens new opportunities for the digitalization of business processes in logistics. The application of these models eliminates the need for maintaining local infrastructure that is typical for on-premises services, while proving companies with the ability to implement software products

tailored to their specific needs. The flexibility of these models allows manufacturing enterprises to adapt to changing market conditions by facilitating quick decision-making based on real-time data and improving planning and forecasting capabilities.

2. Research thesis

The central research thesis of the dissertation is that processes related to managing customer orders, supported and implemented through various information technology software products, can be integrated into a customized cloud system based on the IaaS, PaaS, and SaaS cloud models. Configured according to the specific needs of a manufacturing enterprise, this system can contribute to the enhancement of business operations. This cloud-based information system can streamline the processes of managing sales orders and the internal supply chain by providing adaptive mobile and web applications. Based on methods used by manufacturing enterprises to manage information flows related to customer orders, and procedures for organizing deliveries.

3. Goals and objectives of the study

The research objective of the dissertation is to design and test a cloud-based information system for managing orders from business clients using mobile and web applications, as well as to evaluate its applicability in a specific manufacturing enterprise:

1. To investigate the main principles of the SCM, and analyze the problems related to information provision through corporate ERP and SCM systems.

1.1. To explore the possibilities for digitalization and rationalization of business processes related to managing sales orders from business clients through a customized information system adapted to the specific needs of a manufacturing enterprise.

1.2. To define the nature of cloud services and to study the principles and practices for implementing complex business logic in the programming code of the information system.

2. To propose a conceptual, logical, and communication model that serves as the foundation for the architecture of the cloud-based information system. The models should meet the criteria for security, scalability, and integration.

3. To examine the main functionalities that will bring necessary

improvements regarding the management of orders from business clients.

4. To develop an implementation plan for the system and select appropriate technological means for its realization. The research results should be tested in a manufacturing enterprise.

4. Object and subject of the research

The object of the dissertation research is the processes within the supply chains of a manufacturing enterprise that produces, sells, and delivers its products through separate commercial organizations in multiple countries. The study examines information systems and procedures related to managing the flow of data from the moment of production until reaching the end consumer. Additionally, it investigates the practical application of cloud technologies for order management, logistical operations, and internal supply chains, while adhering to the specific requirements of business clients and the effective use of resources.

The subject of this research is the methods for rationalizing and automating business processes utilizing modern capabilities of cloud platforms and technologies. Based on rationalization methods and a set of software architectural approaches, a customized information system is developed to allow for dynamic adaptation to changing requirements and parameters over time.

The scope of the research is limited to specific problems, architectures, and systems based on their significance for the activities of a manufacturing enterprise. Other potential approaches, supply chain strategies, and the development and implementation of cloud services are beyond the scope of this research.

5. Research methodology

Various research methods were used in the dissertation's methodological framework, the most important of which were information and data collection, comparative analysis, systematization (through classification and typification), inductive and deductive reasoning, modeling, and scientific abstraction.

A systematic approach was adopted to describe the progression of processes related to the study's object and subject. Visual representations of various facts and data were developed using graphical, schematic, and pictorial techniques. By combining these methods and approaches, the research aimed to draw definitive conclusions and provide recommendations.

6. Approbation

For the dissertation's topic, three articles and one report were published. A conceptual model was developed, and the individual modules of the customized cloud-based system for managing orders from business customers were examined. The manufacturing enterprise Heidelberg Materials Devnya was selected to test the system. To build the proposed system, up-to-date software tools and technologies were chosen, and its main development stages were described. The system was tested from March 1 to April 30, 2024.

II. STRUCTURE OF THE DISSERTATION

The dissertation consists of an introduction, three chapters and a conclusion, and is 180 pages long, including 44 figures, 20 tables, 4 appendices. The bibliography covers 186 literary sources. It also contains a list of abbreviations used.

Content:

Manufacturing Enterprise

3.1. Organization of the activity of the company Heidelberg Materials Devnya

3.2. Selection of technological toolkit for the implementation of the system

3.3. Physical implementation of the system

3.4. Adoption of the system through the chosen technological tools

3.4.1. Testing the cloud-based system

3.4.2. System monitoring

3.4.3. Calculating the costs of using cloud services

Conclusion

References

Appendices

III. SYNTHESIZED PRESENTATION OF THE DISSERTATION

Chapter 1. Problems of Information Provision in Managing Customer Orders

The **first chapter** examines the theoretical foundations, terminology, and technologies that define the importance of cloud systems in managing customer orders in a manufacturing enterprise. As a result of the conducted research, key information-provision issues are identified, the basic components of the finished-product delivery strategy are presented, and the interrelationships among various corporate subsystems in the internal supply chain are outlined. The need for developing a customized cloud-based management system that receives and provides information on specific orders and deliveries in real time is also demonstrated.

Specifically, the **first section** presents the essence and characteristics of forward and reverse supply chains, logistics and logistics management, resource planning, customer relationship management, and fleet management. The characteristics of SAP S/4 Hana, one of the leading ERP systems, are reviewed. As of April 2024, 86% of Fortune 500 companies and 92% of Forbes Global 2000 companies were clients of SAP. More than 77% of revenue transactions worldwide are processed through this system, which is used by over 400,000 organizations across 180 countries. The functionalities in SAP are distributed into separate modules, grouped into three main directions: logistics, accounting, and human resources management. Additionally, the dissertation presents

additional subsystems that strengthen the overall structure of the supply chain by processing various types of data,

- geographic data,
- vehicle data,
- transportation data,
- inventory and warehouse stocks,
- status and priorities of orders and requests,
- invoicing and payments,
- order history and customer preferences.

According to studies by various authors, corporate ERP systems are central to the interconnected model of the main components of the SCM strategy for delivering finished products. They manage the flow of goods, materials, information, and capital to ensure effective and coordinated activity. The adaptation of the interconnected model of the main components is presented in Figure 1.

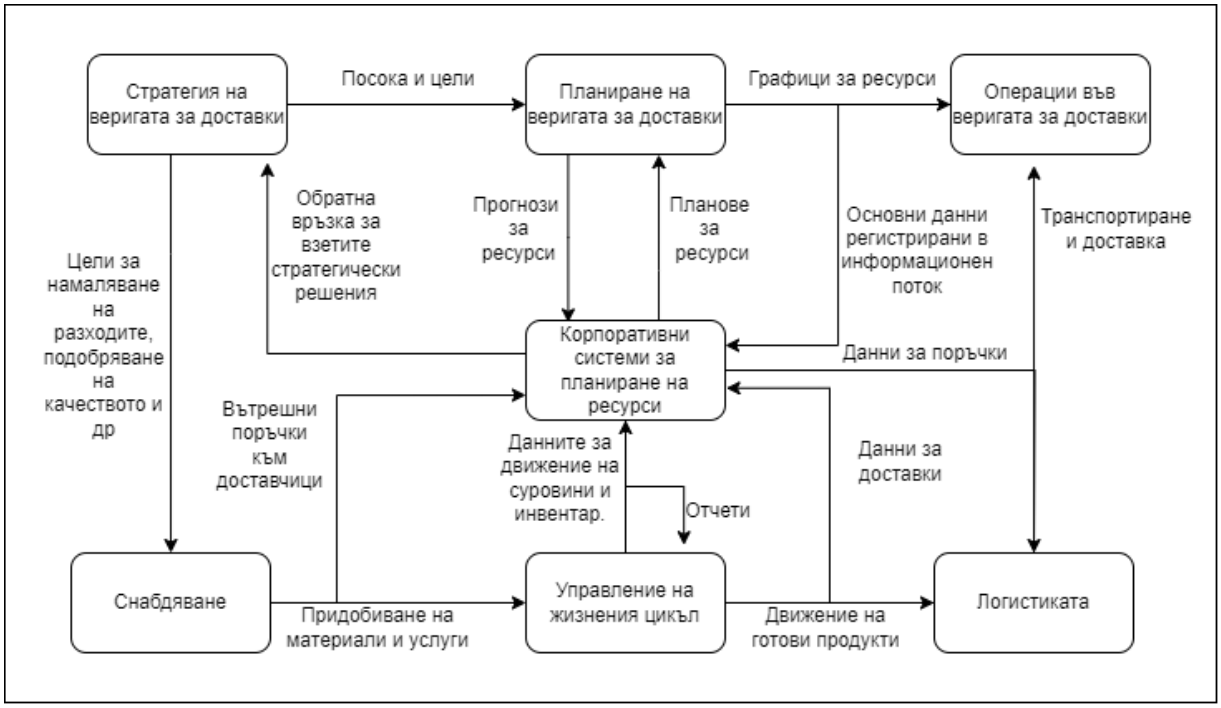

Figure 1. A model of the interconnection between the main components of the SCM strategy for delivering finished products. Adapted from: Türkay et al., 2016; Vasilev & Stoyanova, 2019.

In conclusion, the conducted studies reveal issues related to the interaction between various internal and external information systems, which require operational interoperability, reliable data exchange mechanisms, and a

high level of cybersecurity. Hence, it is necessary to apply standardized methodologies for managing information flows for orders and deliveries to achieve the study's objective.

The **second section** examines the approaches and methods for streamlining order management processes through a customized information system configured for a specific company. A range of software systems for resource planning, logistics management, and SCM are explored. An analysis of the literature and online sources reveals a lack of a specially developed technological model focused on managing customer orders in a manufacturing enterprise—one that adapts ERP and is part of the SCM strategy for delivering finished products. Based on the results of various studies, Figure 2 presents a model integrating four approaches to receiving customer orders in a manufacturing enterprise.

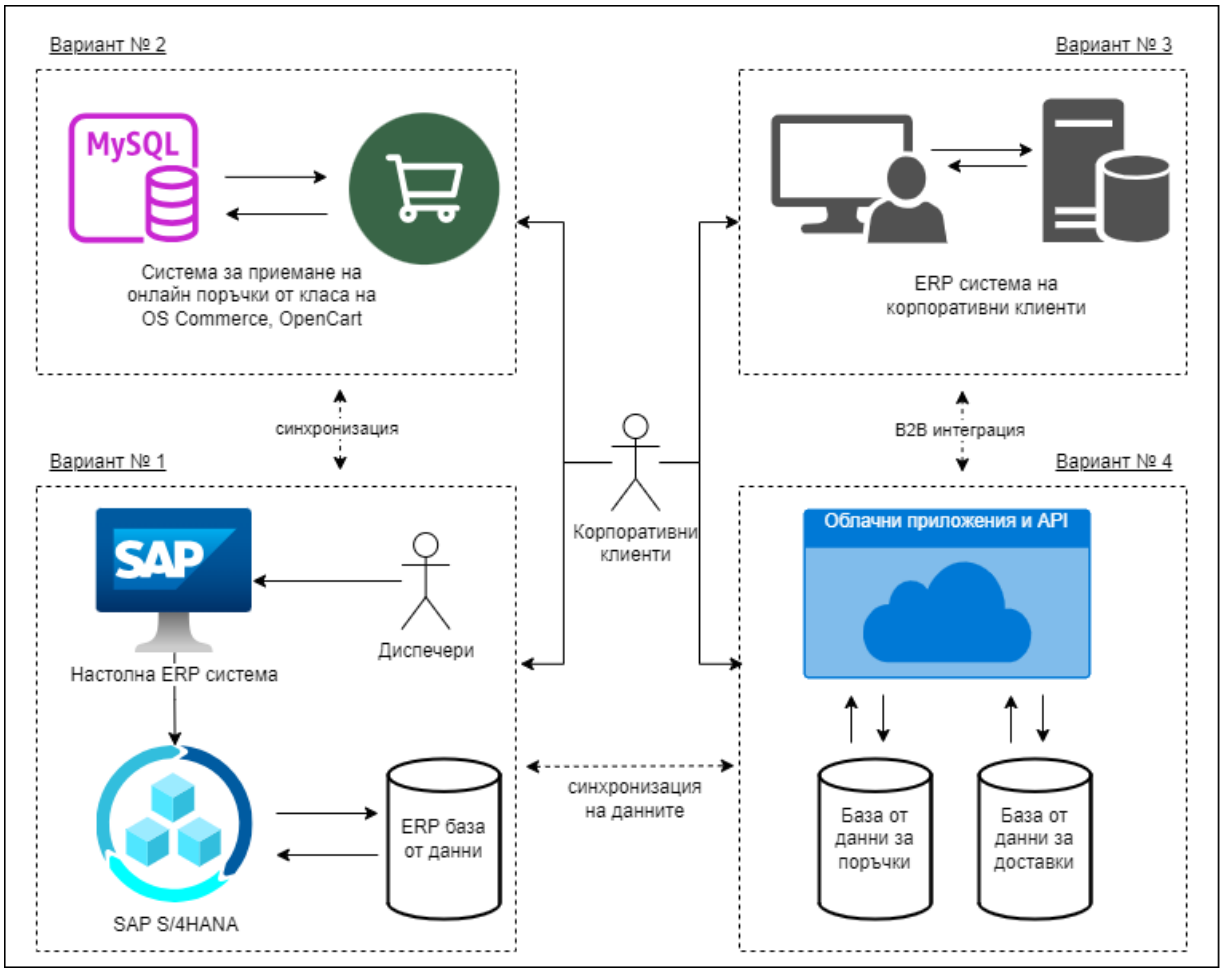

Figure 2. Technological model presenting various options for managing customer orders in a manufacturing enterprise. Adapted from: Vasilev, 2018; Cichosz et al., 2020; Agarwal, 2021.

Based on the presented technological model and the various options for managing customer orders, the need to develop a centralized system is supported. This system must integrate the enterprise's internal and external subsystems to ensure effective control and coordinated processes. In this regard, Figure 3 presents a model of a centralized cloud-based system for managing orders from business clients that ensures adaptability and integration.

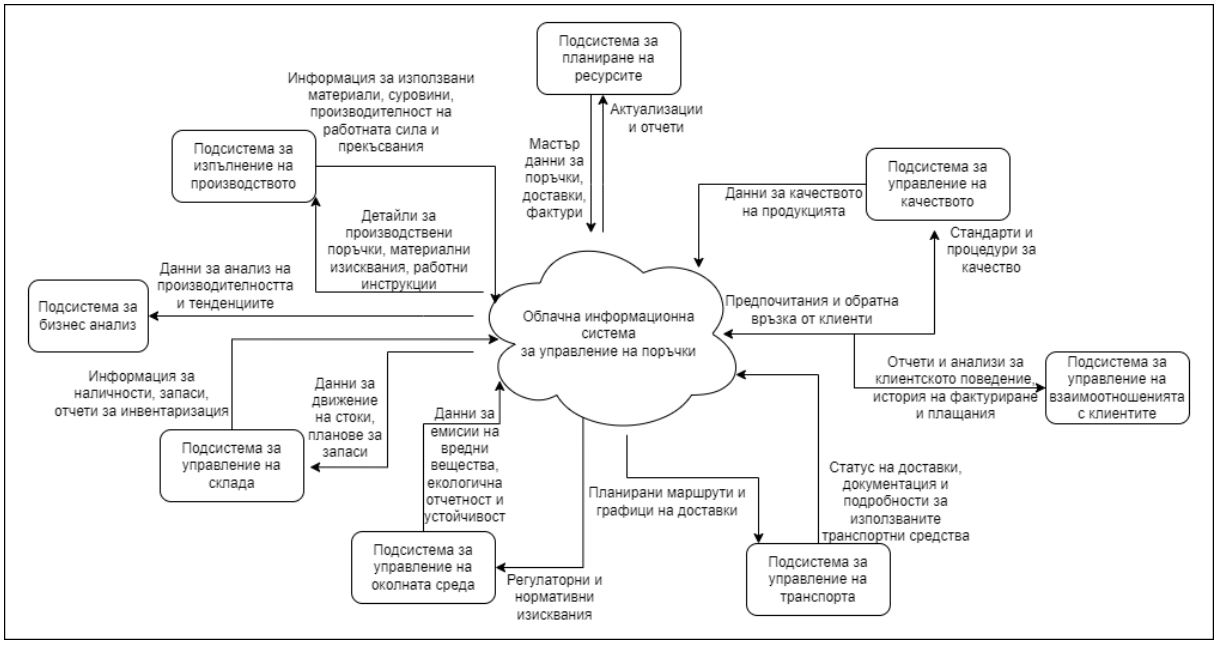

Figure 3. Model of a centralized cloud-based system for order management.
Adapted from: Verwijmeren, 2004; Caserio & Trucco, 2018; Shishmanov & Marinova-Kostova, 2024.

The centralized cloud-based system represents the fourth variant of the technological model for managing customer orders in a manufacturing enterprise (Figure 2). Its primary purpose is to consolidate data from various subsystems and optimize the exchange of information among the different units within the enterprise. By centralizing information management, the system facilitates access to up-to-date data by leveraging technologies such as SAP NetWeaver Gateway. System customization enables the implementation of automated processes and improvement algorithms. The ability to automate helps the system adapt to market dynamics and continually evolving requirements.

The **third section** examines the possibilities for centralizing management processes through the application of cloud technologies. The capabilities of cloud services to provide real-time data exchange and controlled access to information are analyzed. Nonetheless, the concept of cloud

technologies varies. For instance, the National Institute of Standards and Technology (2011) defined cloud computing as "*a model for enabling on-demand network access to a shared pool of configurable computing resources that can be rapidly provisioned and released with minimal management effort*" (p. 36). According to the definition by the Cloud Native Computing Foundation (2018),

> *Cloud technologies enable organizations to build and run scalable applications in modern, dynamic environments—public, private, and hybrid clouds—using networks of services and microservices. These systems are characterized by resiliency, high availability, accessibility, scalability, and manageability, which are essential for diverse business units. Automation of processes allows engineers to deploy software changes with minimal effort*. (p. 37)

High performance and low latency are key characteristics of cloud services. Performance refers to the time from when a user submits an online request to when the system responds. The level of latency is used as an indicator of efficiency and is directly tied to user satisfaction. Two scaling approaches are commonly used to enhance performance: vertical and horizontal. Vertical scalability focuses on improving the hardware resources of the existing infrastructure, while horizontal scalability involves adding additional hardware modules and/or virtual servers.

The field of cloud technologies distinguishes three architecture types: public, private, and hybrid. Public architectures are managed by external providers (e.g., Microsoft, Google, and Amazon) and offer scalable services. Private architectures are maintained within the organization's computing environment to provide control and security. Hybrid clouds combine the advantages of both public and private clouds, which allows for workload sharing.

As mentioned, another classification of cloud services distinguishes three cloud computing models: IaaS, PaaS, and SaaS. In the IaaS model, developers use computing resources in the form of virtual servers, networks, and data storage services. The PaaS model offers a complete platform for software development and deployment, including operating systems, databases, and programming tools. The SaaS model provides a full software package over the Internet that allows developers to use its functionalities without maintaining or managing the model. These models offer a distinct level of control, which makes

them suitable for various organizational requirements, as illustrated in Figure 4.

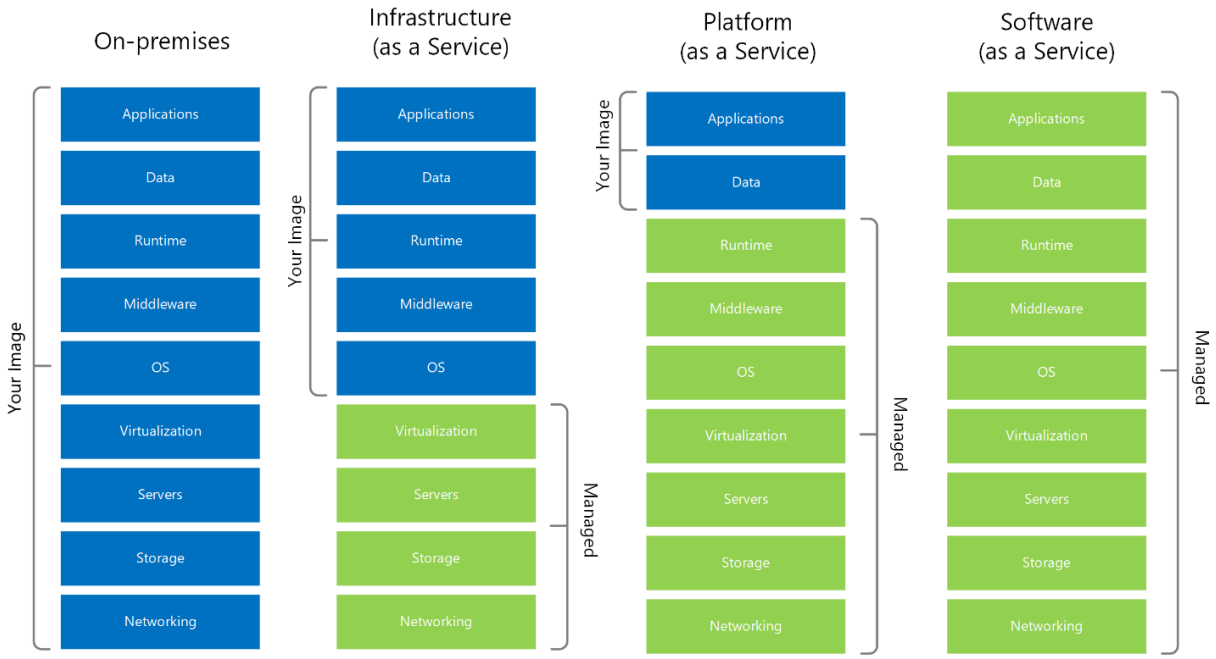

Figure 4. Comparison between cloud service models (IaaS, PaaS, SaaS) and traditional on-premises infrastructure, outlining management responsibilities.
Source: Mohammed & Zeebaree, 2021.

Containerized technologies are recommended for building, delivering, and executing cloud systems. Containerization is an approach in software development whereby the application code, along with all its dependencies and configurations, is packaged into a binary file called an image. Containers allow applications to be isolated from one another within a shared operating environment.

Additionally, the microservices architectural style is suitable for creating cloud systems. Microservices involve breaking an information system into small, independent applications, each of which targets a specific functionality. Moreover, microservices can be developed, deployed, and scaled independently from one another.

The **fourth section** focuses on business process management based on a domain-driven design (DDD) approach. Issues related to communication between microservices, information processing, performance, and technological advancements are analyzed. When developing complex business logic for managing customer orders, teams often encounter difficulties in creating algorithms and data structures, as well as in defining rules and validations. DDD

enables the definition of various areas within the system, which improves the modularity and reusability of microservices through so-called "bounded contexts." In DDD, the use of common terminology and collaboration between developers and SCM experts are encouraged by employing a shared, ubiquitous language.

Another approach that precedes DDD is "data-driven design," in which the logical division of modules and microservices is based on the data they operate on. Fowler (2012) compared DDD to data-driven design in terms of the time required and complexity of software development. The result of this comparison is illustrated in Figure 5

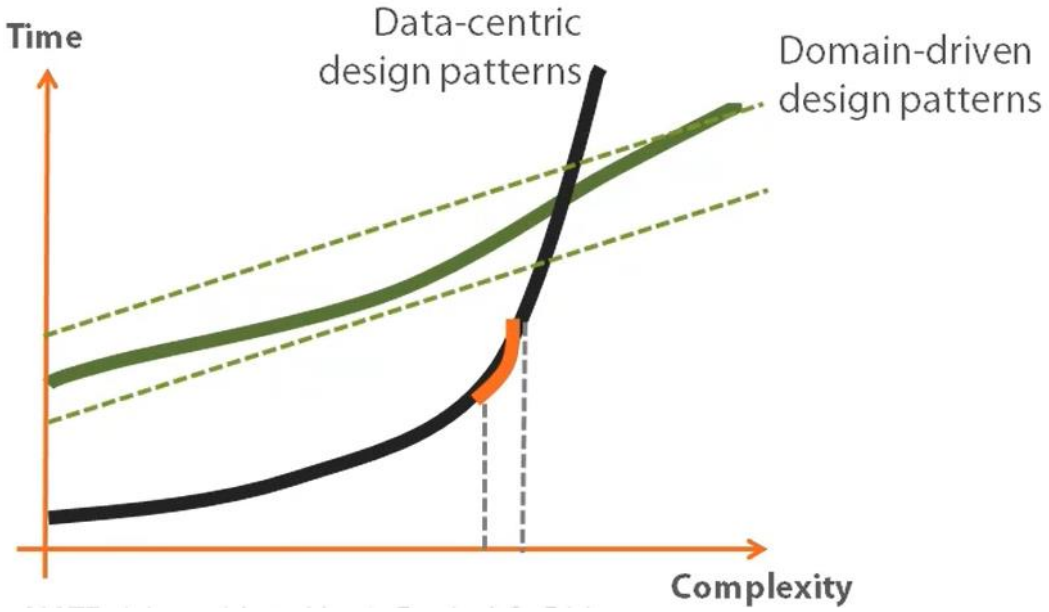

Figure 5. Comparison of Domain-Driven Design with data-driven design in terms of time and complexity in software development. Source: Fowler, 2012.

In addition to the principles of DDD, Young (2010) introduced the concept of command and query responsibility segregation (CQRS). According to this principle, every method in a microservice should be either a command or a query, but not both simultaneously. According to Young, commands are methods that perform operations that change the state of the data in the system, while queries are used for data retrieval.

Notably, according to the consistency, availability, and partition tolerance (CAP) theorem (i.e., Brewer's theorem), a distributed microservices system cannot simultaneously guarantee all three of the following properties:

1. Consistency: All client applications have access to the same view of

the data, even after updates or deletions.

2. Availability: All client applications can find a replica of the data, even in the event of partial malfunctions of the microservices.

3. Partition Tolerance: The system continues to function normally, even with partial issues in the microservice network.

Hence, by implementing CQRS, developers can create cloud services capable of handling large volumes of HTTP requests while maintaining data consistency, which addresses the CAP theorem. CQRS is an intermediate step between DDD and an event sourcing (ES) approach. ES complements CQRS because all changes to the system's data state are recorded sequentially and can be used for reconciliation and analysis.

Furthermore, the need exists for a personalized cloud-based order management system to build on the functionalities of existing SCM and ERP subsystems and ensure their interaction by applying modern real-time data processing technologies. Therefore, a conceptual and logical model is considered that describes the software elements, interfaces, and suitable algorithms, as well as a communication model for managing the composition of cloud microservices.

Chapter 2. Architecture of a Cloud-Based Order Management System for Customer Orders

Based on the derived theoretical postulates, the second chapter proposes an architectural model corresponding to the specifics of customer order management. The chapter develops a conceptual, logical, and communication model that serves as the foundation for modeling and implementing mobile and web applications designed to serve business customers. The scope and requirements for the system under development are defined, while use cases and business scenarios are presented to assist with the design process.

The cloud system is configured according to the specific needs of a manufacturing enterprise to manage key processes and activities within the internal supply chain by extracting and analyzing data in real time. The information obtained is used for the timely updating of workday schedules. Customers receive notifications regarding the estimated delivery time and any changes that occur.

The **first section** presents a conceptual model of the software system. The main business processes and activities of the system, including software

development, maintenance, and support, as well as information business modeling, are outlined. The implementation of a conceptual model proceeds through an iterative process involving several stages: forecasting the system's growth, defining business scenarios, and performing a high-level concept review.

The system growth forecast refers to evaluating or predicting the future development of a given system based on current data, trends, and an analysis of external and internal factors. The forecasting model presents various levels of configurability and scalability to facilitate the use of a single version of the system within a unified infrastructure and optimize the distribution of operating costs among different organizational units of the enterprise.

As a result, functional and non-functional requirements are formulated for a cloud-based order management system. Functional requirements define the specific behavior and operations of the system, including automating order processing, integrating with the enterprise's resource planning in real time, and facilitating customer service interactions. Non-functional requirements determine the system's operational attributes and constraints, including performance metrics, security standards, scalability, and reliability.

A mobile application is designed for end-users to track and manage their orders and deliveries in real time, which should be published on the Google Play Store and Apple App Store. These online distribution platforms for mobile applications also support functionalities for user feedback.

The web application, part of an integrated transportation management system, is designed for dispatchers who create requests, plan deliveries, and manage orders. It serves as a tool for making informed decisions and optimizing work schedules, providing daily reports, and synchronizing data with ERP and SCM subsystems. Thus, dispatchers can correct incorrectly entered data and communicate with customers and suppliers to ensure up-to-date, accurate information regarding the status of orders and deliveries.

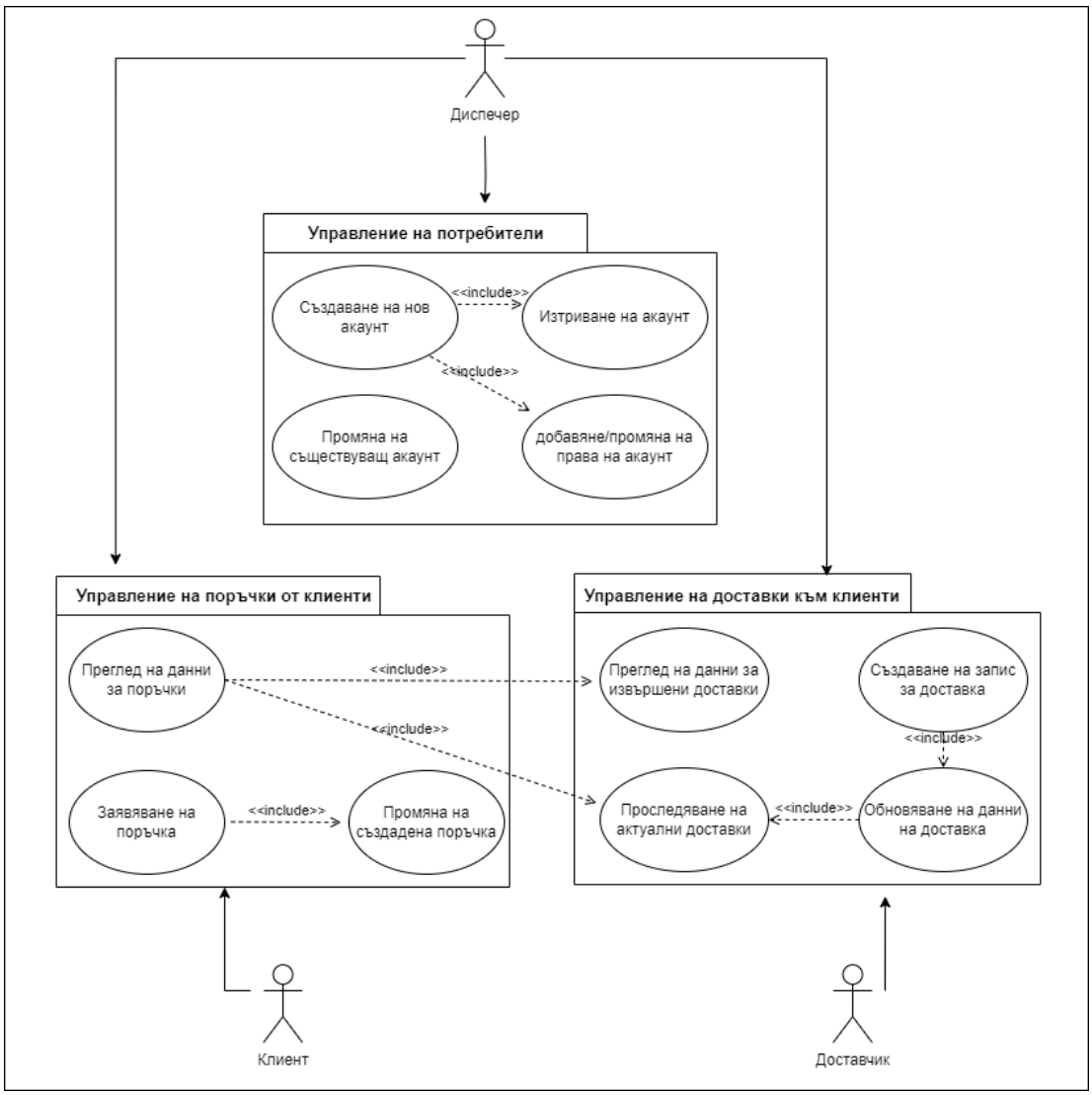

Figure. 6. Diagram of the main business scenario (author's development)

A business scenario diagram, presented in Figure 6, describes the main capabilities of the system: user account management, customer order management, and customer delivery management. The figure also depicts the participants, which in this case include dispatchers, clients, and suppliers. Based on the system growth forecast and the defined business scenarios, functional requirements, and non-functional requirements, the conceptual model (Figure 7) encompasses client applications, cloud microservices, and internal and external corporate systems.

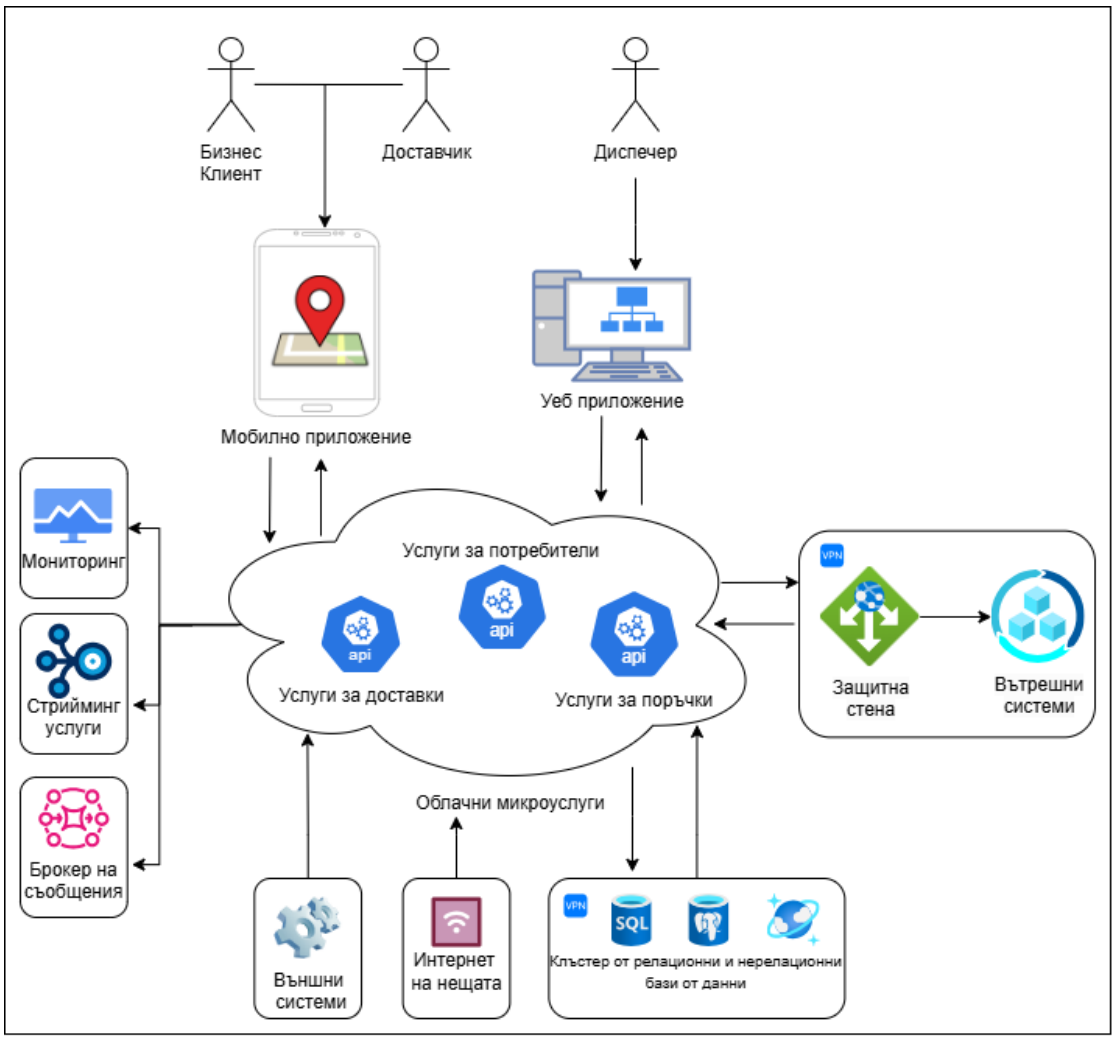

Figure. 7. High-level conceptual model (author's development)

The **second section** outlines the developed logical model of the system based on the conceptual model. The logical model is divided into object-relationship diagrams, main component diagrams, sequences, and activities. The diagrams represent the architecture of the cloud system, which is divided into several modules. Each module is designed to process specific data from the order management process.

The cloud-based order management system consists of three autonomous modules, each with specific responsibilities: order management, delivery management, and user management. Order and delivery management activities are interconnected. Therefore, the microservices in these modules process and store information through a set of microservices and NoSQL databases, which are integrated with internal corporate subsystems and Internet of Things (IoT) devices. The microservices that support these modules adhere to the principles and practices outlined in Chapter 1. These modules store data for sales order management from business clients and tracking vehicles serving deliveries. The

user management module handles activities related to the registration, authentication, and authorization of end-users, suppliers, and dispatchers by using application programming interfaces (APIs) and a relational database for access and identity management.

Bounded contexts are used to differentiate between modules for orders versus deliveries. Each context has its own business logic and data validation rules. Thus, changes in the order context do not directly affect the delivery context. The component diagram presented in Figure 8 visualizes the interaction between the modules, microservices, databases, internal and external subsystems, and mobile and web applications.

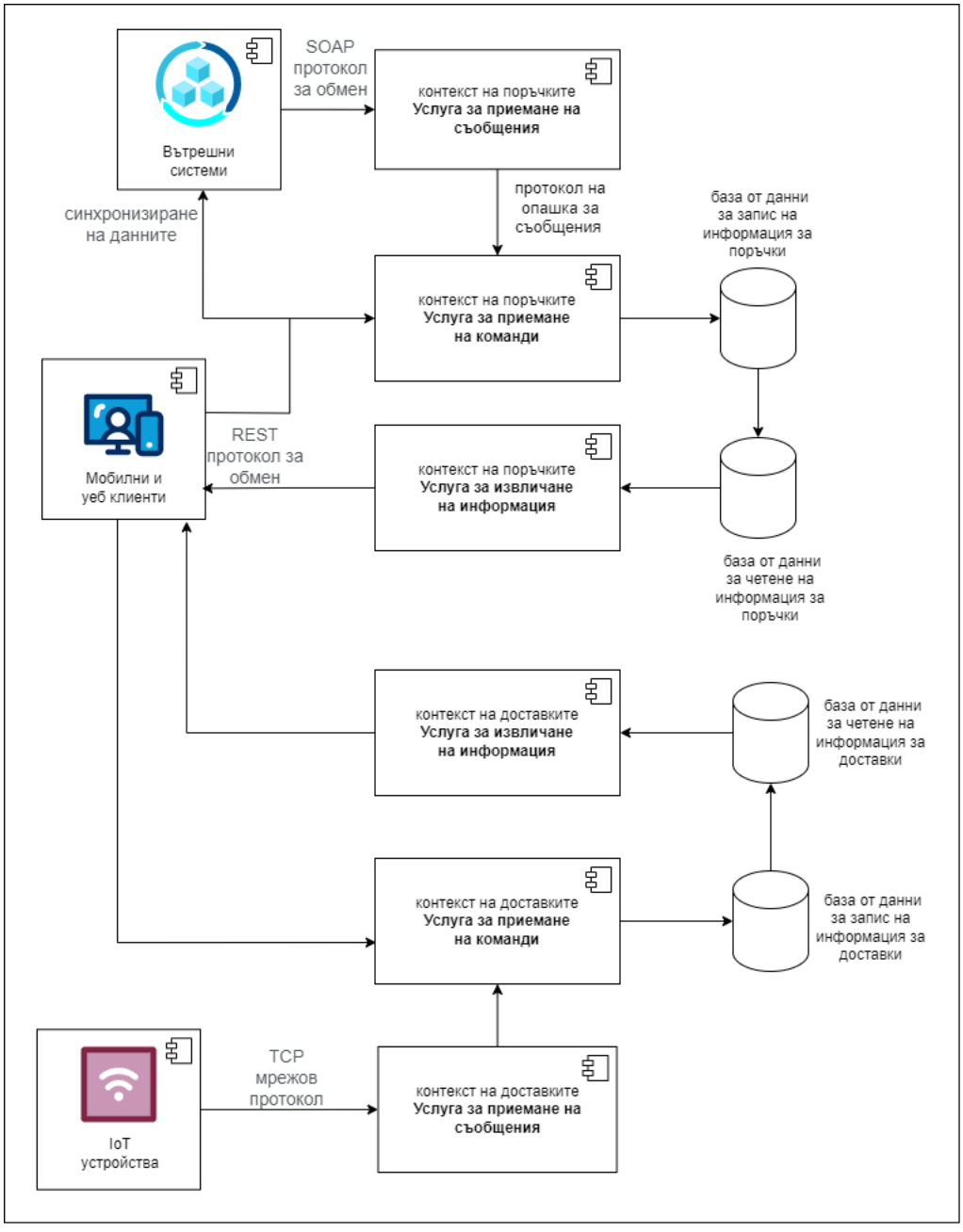

Figure. 8. Basic components of a cloud-based order management service and the relationships between them (author's development)

The cloud-based order management system has a modular structure that distributes responsibilities among microservices for message reception, command execution, and information retrieval based on DDD and CQRS. Command execution services update the system's data state and synchronize with ERP, while information retrieval services perform queries without changing data. The system uses a replication mechanism between read and write databases. The four databases use the same table structure, which allows for the parallel processing of large volumes of data. The schema of each database includes two main tables: streams and events. This organization facilitates the addition of new types of streams and events without necessitating changes in database structure. Events are stored chronologically, which enables their subsequent reconstruction. Figure 9 presents a relational (E-R) model of the tables for streams and related events.

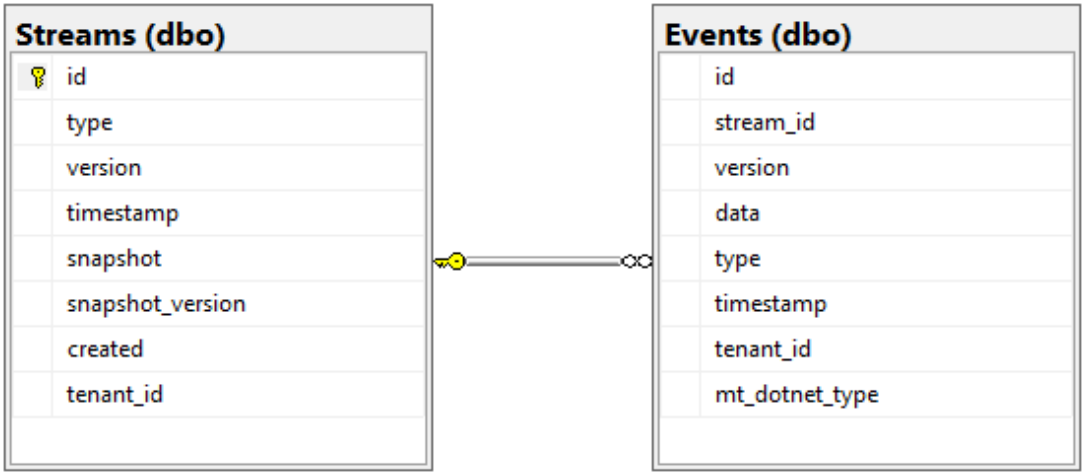

Figure. 9. Relational (E-R) model of the tables for streams and related events (author's development)

Subsequently, the decomposition of modules at the microservice level is discussed to provide an abstract view of the functional components and their interrelationships. Each component within these contexts presented in Figure 7 is located at the highest level in the hierarchy, which provides an API for the connection between the data and client web and mobile applications. Within the internal structure of each microservice are two main subdirectories, source code (SRC) and tests, which contain the source code and component tests, respectively. To ensure functional consistency and adhere to the core principles of DDD, each service in the system should use a similar package structure. Each

package contains object-oriented program code. To graphically visualize the classes, their attributes, methods, and relationships between them, a class diagram is presented in Figure 10.

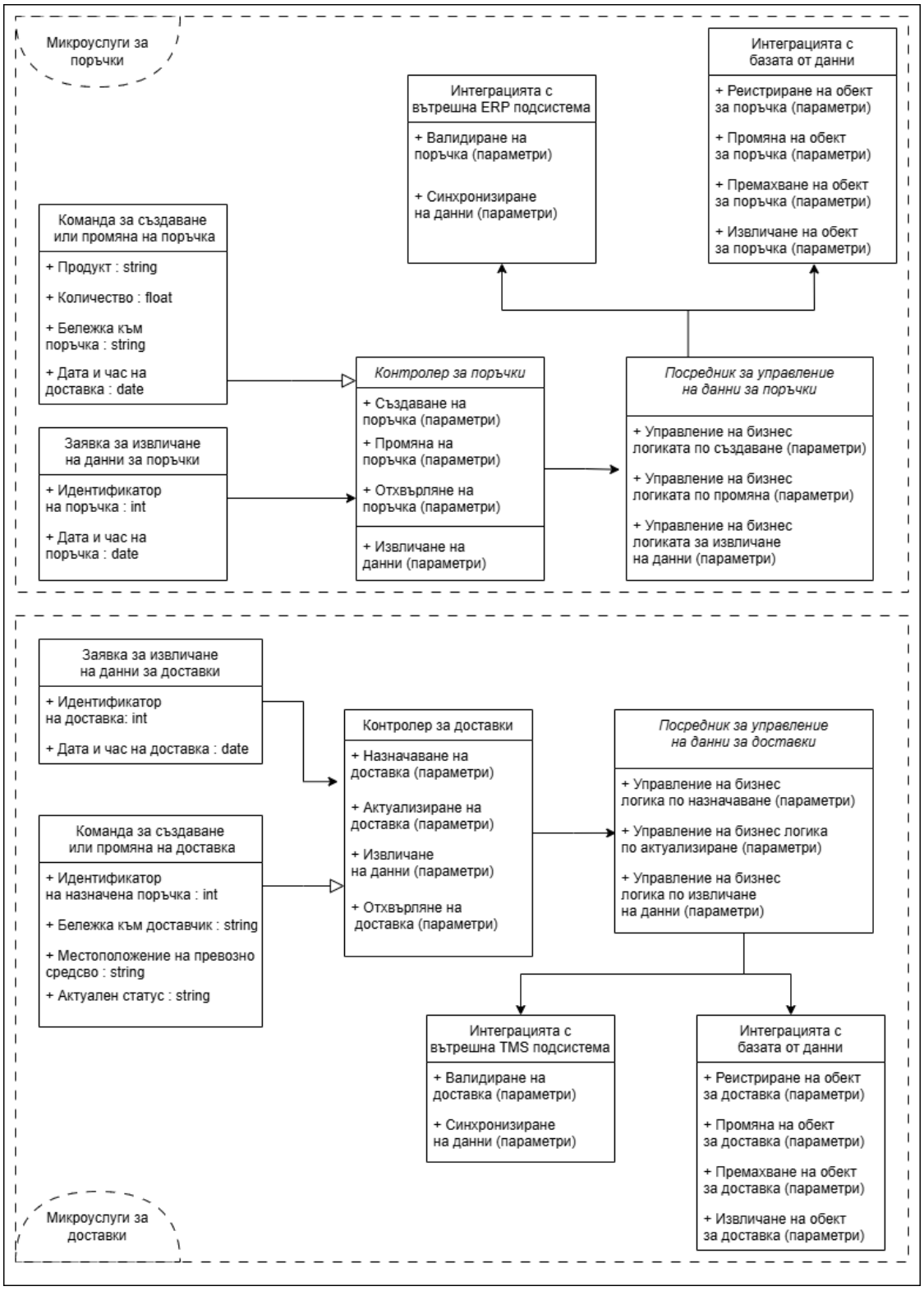

Figure. 10. Class diagram and their relationships (author's development)

Figure 10 has two main sections: microservices for orders and microservices for deliveries. In both sections, classes for commands, queries, network controllers, data management intermediaries, classes for database integration, and internal subsystems are encountered. The user profile management module operates via the API of a centralized identity provider to support secure password storage, multi-factor authentication, and constant interaction with the microservices to protect the data from unauthorized access and cyber threats. Authentication and authorization protocols are based on OAuth 2.0 and OpenID Connect.

The **third section** presents a communication model that describes the interaction between different system modules, information exchange, and data flow management via a sequence diagram (Figure 11).

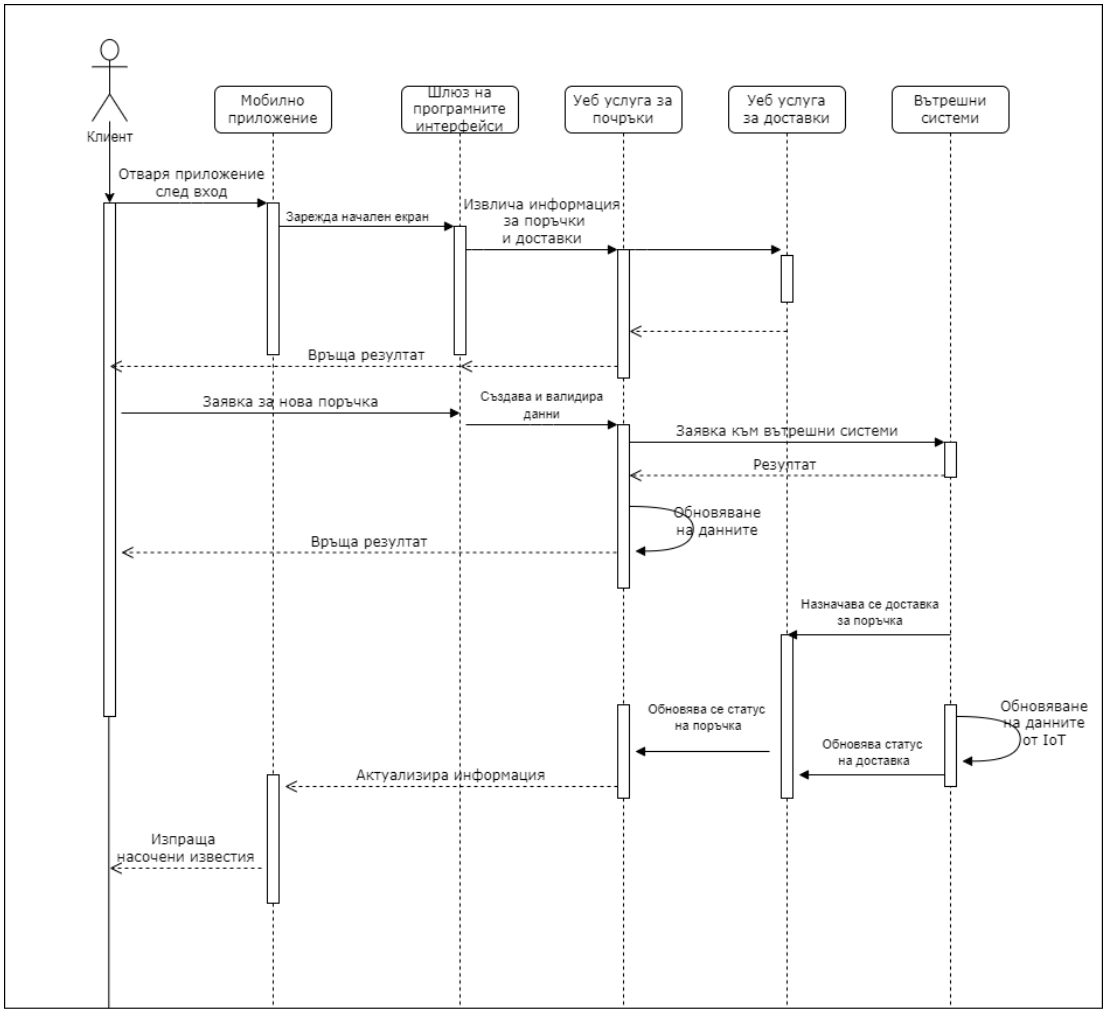

Figure. 11. Sequence diagram of a business scenario for executing client requests (author's development)

The main problem this model solves is choosing among the appropriate communication protocols: HTTP/HTTPS, SOAP, REST, and gRPC. The primary purpose of the diagram is to visualize the interactive collaboration among the system's various components. The sequence diagram demonstrates how the products and services within a cloud-based order management system interact to execute a key business scenario, showing the timeline and order of operations.

The **fourth section** presents functionality and the user interface. The user interface provides a visual connection between the system and the end user, where intuitive design enhances the user experience by making the features relatively easy to use:

- Authentication and authorization occur via usernames and passwords.
- A mobile application for business clients displays a list of orders, including buttons for the menu and new orders. Order statuses are shown using color coding, with options for contacting the supplier and tracking deliveries.
- Supplier features include a quick link to report damage and contact business clients, a display of delivery details, and updates through server queries.
- A web portal for dispatchers integrates functionalities for managing orders, logistics operations, and user accounts. It updates data in real time and allows for generating reports and analyses.

Chapter 3. Building and Using a Personalized Cloud System for a Manufacturing Enterprise

This chapter examines the practical and applied issues related to implementing a cloud-based order management system (i.e., a software product) within the Heidelberg Cement Devnya manufacturing company. This company is a leading cement producer in Bulgaria, located in the town of Devnya, Varna Region. In addition to cement, the enterprise offers a range of concrete mixes and specialized construction materials as a subsidiary of the German multinational corporation Heidelberg Materials. According to a business report, Heidelberg Cement Devnya was ranked as the most profitable construction materials company for 2023.

The **first section** of this third chapter discusses the main characteristics of the company's operations, including handling client requests and logistics activities for delivering materials (e.g., concrete, sand, gravel, cement, and asphalt). The process starts with entering a client request into the ERP subsystem with an initial status of "unconfirmed." The availability of materials is checked, and once confirmed, the request is forwarded to the production team and the logistics department for delivery planning. One day before the delivery, the dispatcher contacts the client for confirmation. If the client declines the delivery, additional costs arise, along with the need to manage returned inventory and operational challenges in logistics. During the delivery, the dispatcher monitors potential problems and manually updates the delivery status. The final stage includes delivery to the client and confirmation of receipt, after which the order is marked as "completed," and an invoice is issued.

The analysis emphasizes the role of dispatchers and the need to improve automation levels in processes. Consequently, the implementation of the cloud-based order management system can lead to significant improvements in the following areas:

- Order Acceptance: A cloud-based order management system is adapted with functions for online registration and agreement to general terms. Once approved by the dispatcher, users can register new orders as well as modify or cancel existing ones.
- Scheduling: The online portal serves as a tool to facilitate management and automation of the specified tasks by integrating internal subsystems with cloud services. In the event of a modified or canceled order, the system automatically reschedules. All interested parties are kept informed via automated notifications.
- Loading: The loading process for concrete mixes and inert materials begins with adding sand, gravel, and cement, along with water, into a transport truck (including mixer types). The materials are mixed until a homogeneous mixture is obtained and then transported to the construction site. The cloud-based order management system collects data on water levels and temperature from IoT sensors, which are sent in real time to a cloud platform.
- Delivery: The delivery process includes the real-time tracking of the trucks. Drivers can choose optimal routes, and clients receive access to the

vehicle's current location. In case of an unforeseen incident (e.g., an accident or a flat tire), drivers can contact the responsible personnel at the work site and notify the dispatchers. Clients sign documents digitally after delivery is completed. In addition, models can be created to support forecasting business clients' needs and dynamic pricing by analyzing order history and using machine learning algorithms.

In the **second section** of the third chapter, a set of technological tools is selected to implement the system based on the requirements for various mobile and web applications. A study and evaluation of different technological aspects are conducted, including programming languages, frameworks, public cloud service providers, and databases. A comparative analysis of web-based frameworks shows that .NET Core is a suitable choice for building cloud microservices.

Microsoft Azure has established itself as a leading provider of cloud services while offering a high degree of integration with .NET. Azure provides a variety of data storage services (e.g., the Azure SQL Database, Cosmos DB, and Blob Storage). The following technological tools were selected for implementing the cloud system:

- ASP .NET Core for server-side logic and microservices.
- Azure as the public cloud services provider.
- Azure SQL Database, Cosmos DB, and Blob Storage for databases.
- RabbitMQ as a message broker.
- Microsoft Blazor as the technology for the web portal.
- .NET MAUI for mobile application development.

The **third section** describes the physical implementation of the system, including the selection of hosting services based on the criteria of dynamic scaling, virtualization and containerization, orchestration tools, continuous integration, and delivery. Among the main hosting options are Azure Virtual Machines (suitable for IoT-related applications) and Azure Kubernetes Service, which offer automation and microservices management along with high reliability and performance.

The third section also discusses the choice of a system for storing and maintaining source code. Platforms like GitHub, GitLab, Bitbucket, and Azure DevOps facilitate software development and version control. Based on the

evaluation conducted, GitHub is identified as a suitable choice due to its intuitive interface and variety of tools. The public repository with the cloud system's source code is available at: https://github.com/profjordanov/cloud-based-management-information-system.

Virtualization and containerization are crucial for modern cloud systems since they significantly support DevOps practices. These technologies enable isolated application environments that increase security and efficiency. The workflow involves the development, testing, and deployment stages, which are automated through tools such as Docker and GitHub Actions to ensure continuous delivery and high-quality software. Hence, resources are optimized, and the system's reliability and resilience are ensured.

Based on the software technologies and tools discussed, Figure 12 presents an architectural diagram that corresponds to the conceptual, logical, and communication model. The figure includes the main components of the software product.

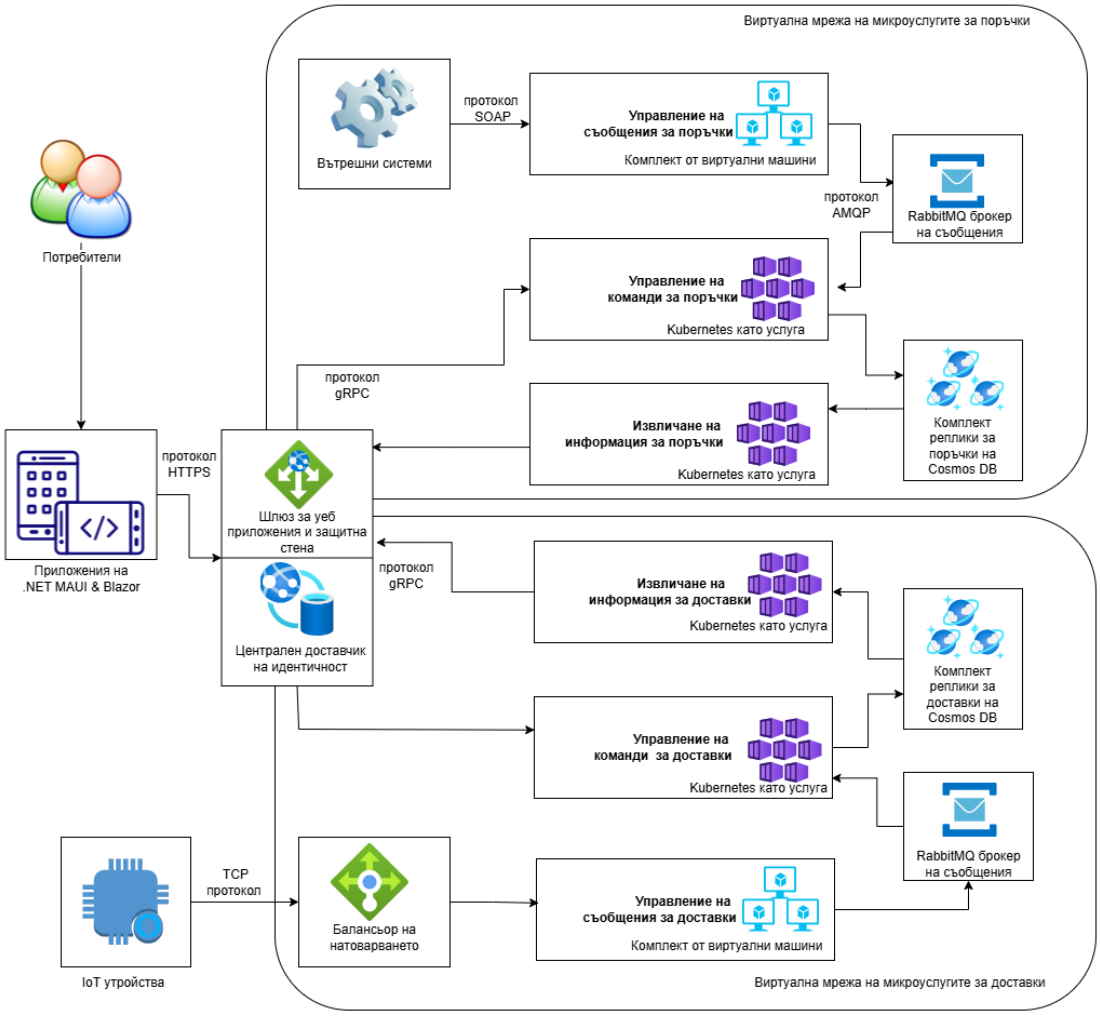

Figure. 12. Architectural diagram of the software technologies (author's development)

The **fourth section** presents the results of the system's trial at Heidelberg Cement Devnya. Various testing strategies are employed to assess the benefits of developing and deploying the cloud system. Testing enables the determination of whether the system functions according to users' expectations. This trial phase employs an A/B testing strategy, where both manual and automated test procedures are conducted to simulate user behavior in a temporarily created cloud environment. Manual tests enable verification of specific functionalities and assessment of the user interface, while automated tests register and modify data through the microservices' endpoints.

These test procedures are divided into two groups: Group A simulates system use by business clients, while Group B simulates system use by the supplier. The procedures cover various scenarios to ensure a comprehensive evaluation of cloud-based order management system functionality and security. Group A tests focus on managing order information, while Group B tests focus on delivery information.

After conducting the required tests and analyzing the results, Tables 1 and 2 present the collected data and include the following:

- the order ID, an automatically generated unique identifier for each order.
- the order date, the date when the order was placed via the mobile application.
- the preferred delivery date.
- the delivery status, which is the current, updated status of the delivery (e.g., delivered vs. upcoming).
- the GPS coordinates of the current location.

Table 1
Results from A/B Tests in creating orders and deliveries (author's development)

ID	Order Date	Delivery Date	Status	Coordinates
12314	1-Mar-24	3-Mar-24	Delivered	40.7128° N, 74.0060° W

ID	Order Date	Delivery Date	Status	Coordinates
32262	4-Mar-24	6-Mar-24	In Transit	34.0522° N, 118.2437° W
23123	7-Mar-24	9-Mar-24	Delivered	41.8781° N, 87.6298° W
33434	10-Mar-24	12-Mar-24	Pending	29.7604° N, 95.3698° W
90905	13-Mar-24	15-Mar-24	Delivered	33.4484° N, 112.0740° W
66786	16-Mar-24	18-Mar-24	In Transit	39.7392° N, 104.9903° W
90867	19-Mar-24	21-Mar-24	Delivered	32.7767° N, 96.7970° W
34248	22-Mar-24	24-Mar-24	Pending	37.7749° N, 122.4194° W
23129	25-Mar-24	27-Mar-24	Delivered	47.6062° N, 122.3321° W
31210	28-Mar-24	30-Mar-24	In Transit	25.7617° N, 80.1918° W

All the data presented in Table 1 is available in the company's SAP ERP subsystem. This demonstrates the interaction between the corporate subsystems and cloud-based order management system. In addition, Table 2 illustrates changes in some of the orders and deliveries, which are also reflected in the internal subsystems.

Table 2
Results from A/B Tests in modifying orders and deliveries (author's development)

ID	Order Date	Delivery Date	Status	Coordinates	Changes
12314	1-Mar-24	~~3-Mar-24~~ 5-Mar-24	Delivered	40.7128° N, 74.0060° W	Change in delivery details
32262	4-Mar-24	6-Mar-24	In Transit	~~34.0522° N, 118.2437° W~~ 56.7532° N, 96.4615° W	Change in coordinates

ID	Order Date	Delivery Date	Status	Coordinates	Changes
23123	7-Mar-24	9-Mar-24	~~Delivered~~ Cancelled	41.8781° N, 87.6298° W	Change in status

After implementing the cloud system, continuous monitoring of the client and server applications is required. The monitoring integrated into the Azure cloud platform offers tools for observation, logging, and tracking performance indicators that can be utilized by developers, business specialists, and managers. Monitoring is divided into two main categories: infrastructure monitoring and application monitoring. The first category involves evaluating and controlling system resources (i.e., the central processing unit, memory, disk space, and network traffic), while the second category focuses on monitoring the functionality and efficiency of individual services using parameters such as response time, error frequency, and tracking transactions between microservices and ERP.

Continuous monitoring is crucial because it allows for the detection of fluctuations caused by unforeseen events (e.g., market disruptions and sudden changes in user preferences). An example of this fluctuation in demand, captured in a controlled environment, is illustrated using Azure Monitor in Figure 13. Early detection of these issues by dispatchers enables a timely response that can prevent additional complications.

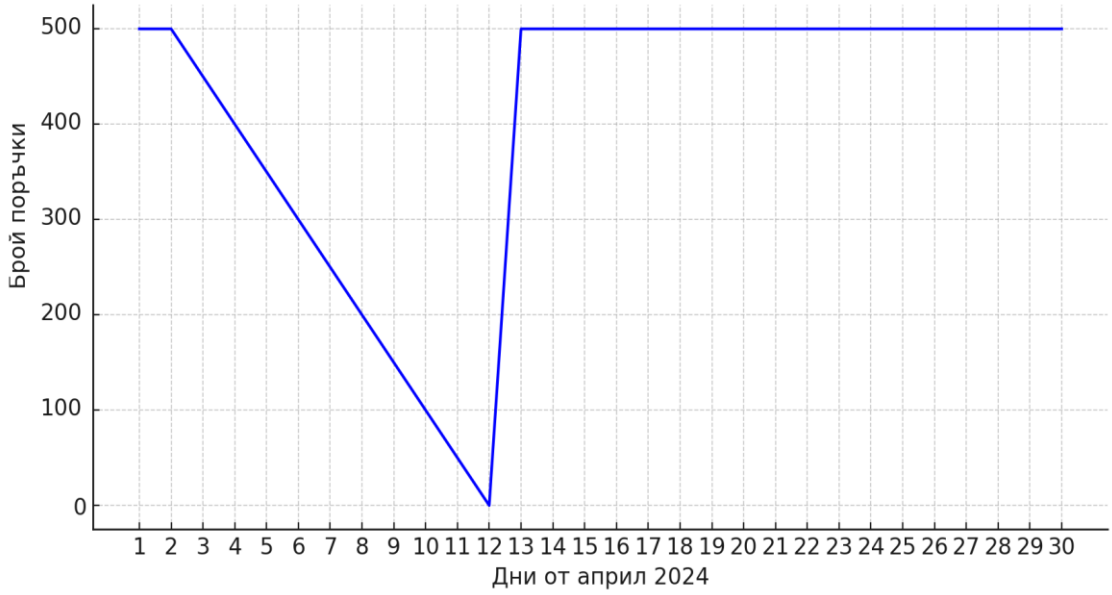

Figure. 13. Example of fluctuation in demand monitored with Azure Monitor (author's development)

In **conclusion**, this dissertation explores the issues related to managing sales orders for business clients in a manufacturing company. Particular attention is paid to problems concerning information support and the digitalization of the internal chain of orders and deliveries while building on corporate systems for resource planning, transportation management, and customer relations. Moreover, the solution also incorporates external IoT devices.

The challenges examined include providing real-time data, automating processes, reducing manual labor, and managing logistics operations efficiently when delivering products. The study examines measures and approaches for enhancing coordination and communication among various participants in the supply chain to ensure the company's strategic resilience in a competitive environment.

Below is the dissertation roadmap, which summarizes the content of the respective chapters:

- Chapter 1 presents the theoretical framework that addresses the first research objective as a contribution to this dissertation. It provides an analysis of modern ERP and SCM subsystems, their application within the internal supply chain, and a variety of cloud architectures and technologies. The chapter covers virtualization, containerization, orchestration tools, and the three cloud service models (i.e., IaaS, PaaS, and SaaS). Additionally, it examines opportunities for integration, implementing omnichannel strategies, and managing business processes using a DDD approach.
- Chapter 2 draws on the principles and practices of order management to present the conceptual, logical, and communication model, along with the functionality and user interface of the cloud-based system. These models are created based on established standards for visual modeling. The system architecture and the proposed models are integral to the second research objective and form a key contribution to this study.
- Chapter 3 overviews the activities at Heidelberg Cement Devnya based on the presented models, followed by a selection of suitable software technologies for the physical implementation of the cloud-based system by

examining the technical requirements and possibilities for integration with the company's existing subsystems. A plan for building and evolving the cloud-based system is developed that encompasses the stages of integration, configuration, and testing. As a result, clarity and predictability are ensured throughout the project execution.

The practical applicability of the study is demonstrated using A/B testing. The testing results reveal that the developed cloud-based management information system facilitates order fulfillment, enhances resource management, and provides automation and scalability. This outcome constitutes a substantial practical contribution.

IV. THE MAIN CONTRIBUTIONS TO THE DISSERTATION

1. **Modeling of the System Architecture**

Conceptual, logical, and communication models of the information system have been developed and visually presented using established software tools. These models form the architecture of the order management system and provide the foundation for its further implementation.

2. **Technological Selection and Integration**

Suitable software technologies have been identified for the physical implementation of the cloud system, considering both technical requirements and the possibilities for integrating with the existing subsystems in the enterprise. The selection includes programming languages, frameworks, and tools tailored to the specific needs of the project.

3. **Implementation and Testing Plan**

A plan for building the cloud system has been developed, outlining the various stages of integration, configuration, and testing. In this way, structure and predictability are ensured during project execution.

4. **A/B Testing in a Simulated Cloud Environment**

For demonstrating the system's applicability, an A/B testing strategy was chosen at the Heidelberg Cement Devnya manufacturing plant. This strategy was trialed through manual and automated test procedures, simulating real user behavior in a temporarily established cloud environment. The cloud-based information system is connected to the company's test ERP subsystem.

V. LIST OF THE PUBLICATIONS ON THE TOPIC OF THE DISSERTATION

Articles in Journals

1. **Jordanov, J**., Petrov, P., Vasilev, J., Kuyumdzhiev, I. (2025). *Domain-Driven Design in Cloud Computing: .NET and Azure Case Analysis*. TEM Journal. 14(1), pp.44-54.
2. **Jordanov, J**., Simeonidis, D., Petrov, P. (2024). *Containerized Microservices for Mobile Applications Deployed on Cloud Systems*. International Journal of Interactive Mobile Technologies, 18(10), pp.48-58.
3. **Jordanov, J**., Petrov, P. (2023). *Domain Driven Design Approaches in Cloud Native Service Architecture*. TEM Journal, 12(4), pp.1985-1994.

Conference Scientific Reports

1. **Jordanov, J**. (2024) *Opportunities for Streamlining Business Processes in Manufacturing Enterprises Through the Application of Cloud Technologies*. Proceedings from the International Scientific Conference "Information and Communication Technologies in Business and Education" (pp. 169-176). “Nauka i Ikonomika” Publishing House, University of Economics – Varna.