

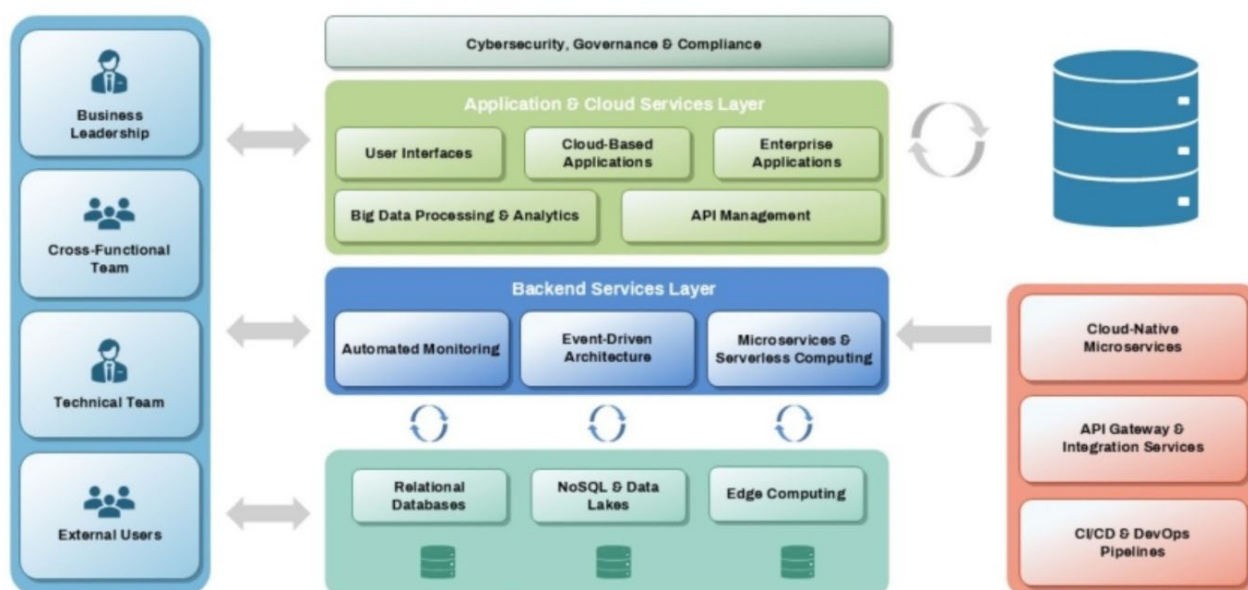


## KAPITEL 6 / CHAPTER 6<sup>6</sup> INTELLIGENT DESIGN OF CORPORATE INFORMATION SYSTEMS

DOI: 10.30890/2709-2313.2025-45-02-032

### Introduction

In the contemporary context of enterprise digital transformation, corporate information systems (CIS) constitute an indispensable foundation for managing resources and business processes (figure 1).



**Figure 1 – Corporate Information System of an IT Company**

*Reference: [1].*

At present, particular interest is focused on:

- methodological;
- architectural;
- algorithmic approaches to the design of corporate information systems.

A key objective is to identify the principal mechanisms for:

- cost reduction;
- increasing the efficiency and adaptability of corporate platforms.

The task of intelligent design of CIS is becoming especially relevant, as the

<sup>6</sup>*Authors: Tokariiev Volodymyr Volodymyrovych*

*Author's sheets: 0,49*



complexity of modern systems is growing exponentially [2]. The volumes of data generated by CIS and the increasing dynamism of business requirements make effective manual management of system architecture and computational resources impractical. Traditional approaches based on static rules and expert assessments rapidly lose their ability to ensure the required levels of performance, resilience, and, no less importantly, economic efficiency in an environment of constantly changing cloud pricing models and market demand [3]. Consequently, there arises a strategic need to develop self-learning and proactive systems capable of autonomously adapting to a changing operational environment [4]. This work proposes a detailed examination of the transition from reactive management, which focuses on correcting problems after they occur, to a proactive cycle in which architectural decisions and configurations are generated and validated automatically [5]. A central element of this process is the application of:

- artificial intelligence (AI);
- machine learning (ML).

These methods are used to analyze massive volumes of operational CIS data (logs, performance metrics, transactions) in order to:

- first, accurately predict future workloads and potential failures (so-called predictive maintenance);
- second, employ reinforcement learning techniques for the automatic and optimal allocation of computational resources in real time.

This approach enables global optimization by balancing conflicting objectives, such as minimal system latency and minimal resource consumption. Particular attention is devoted to the concept of digital twins of CIS. These are high-fidelity simulation models that allow designers to simulate the impact of any architectural changes (for example, database migration or modification of a microservices structure) in a safe virtual environment before such changes are deployed to the production system [6]. This significantly reduces the risks of incompatibility and the accumulation of technical debt. Currently, three main directions shaping the intelligent paradigm can be identified.



The first direction examines data-driven architectural patterns based on quantitative metrics of coupling and component complexity.

The second direction analyzes artificial intelligence algorithms for automated diagnostics and performance optimization, including the application of genetic algorithms to search for optimal configurations.

The third direction focuses on methodological frameworks for integrating these intelligent tools into existing agile processes (Agile/DevOps), ensuring a continuous cycle of observation, modeling, and deployment.

In this context, it becomes important not only to create innovative intelligent management tools but also to develop a unified methodological foundation that systematically integrates:

- architectural modeling concepts;
- real-time data analytics;
- principles of IT solution lifecycle optimization;
- automated managerial decision-making.

Thus, intelligent design of corporate information systems is considered an interdisciplinary field that integrates:

- information technologies;
- software engineering;
- mathematical modeling and artificial intelligence.

A key success factor is the transition to adaptive architectures capable of modifying their structure and configuration depending on:

- changes in workload;
- business priorities;
- availability of cloud resources;
- levels of risk and incidents.



## **6.1 Key stages of the methodology for designing corporate information systems**

Intelligent design of corporate information systems is an approach that employs methods of systems analysis, mathematical modeling, machine learning (ML), and artificial intelligence (AI) to create, evolve, and continuously improve the architecture, processes, and performance of large-scale information systems. Unlike traditional design, where decisions are made based on expert knowledge and fixed requirements, the intelligent approach involves the creation of a self-learning system that:

- predicts future needs and workloads;
- simulates the impact of architectural changes prior to their implementation;
- automatically adjusts system parameters (e.g., scaling and resource allocation) in real time.

The design of corporate information systems is based on four interrelated components.

### **1. Modeling and simulation.**

This component forms the basis for decision-making. Digital models of CIS are created, including:

- business process models (BPMN, EPC) to understand workflow execution;
- architectural models (ArchiMate, UML) to describe component structures;
- simulation models (digital twins).

These computer simulations reproduce system behavior under various load scenarios, failures, or changes. This allows designers to simulate, for example, the impact of a database migration or the introduction of a new microservice without risk to the real system.

### **2. Artificial intelligence and machine learning (AI/ML).**

AI technologies are used to analyze vast volumes of CIS operational data:

- logs;
- metrics;
- transactions.

ML models detect anomalies and predict component failures well before they



actually occur. Reinforcement learning algorithms can make decisions about dynamic allocation of computational resources more effectively than static rule-based approaches. AI also analyzes query patterns and suggests optimal indexes or schema refactoring.

### 3. Data-driven architecture.

Decisions regarding architectural evolution (e.g., transitioning from a monolith to microservices) are made not intuitively but based on quantitative metrics:

- coupling metrics;
- complexity metrics.

Automated tools measure dependencies between components, helping to identify bottlenecks and the need for refactoring. Subsequently, economic optimization is performed, including analysis of cloud resource usage costs for different architectural solutions.

### 4. Intelligent optimization process.

This process represents a continuous cycle comprising the following phases:

- observe (analysis): collection of large volumes of operational data (workloads, latency, errors, resource costs);
- diagnose & predict: ML models analyze data, identify root causes of problems, and predict when the system will reach capacity limits;
- model & generate: potential architectural changes are formulated based on forecasts and introduced into the system's digital twin;
- optimize: algorithms (e.g., genetic algorithms or search-based methods) automatically evaluate thousands of configurations and identify a globally optimal solution (e.g., minimal latency at minimal cost);
- deploy: the optimal solution (such as new configuration settings or scaling rules) or an architectural patch is automatically deployed to the real CIS, initiating a new observation cycle.

This approach transforms CIS management from reactive (problem correction after occurrence) to proactive and self-optimizing (table 1).

**Table 1 – Examples of application**

CIS Domain	Problem	Intelligent Solution
Resource management.	Excessive cloud service costs due to inefficient scaling.	ML models predict workloads with 95% accuracy and automatically adjust the number of virtual machines, saving 15–30% in costs.
Customer service.	High latency during peak loads leading to transaction loss.	The system dynamically reallocates transaction priorities, routing critical requests to the least-loaded microservices and minimizing waiting time.
Security	Difficulty detecting hidden threats among millions of log records.	Neural networks identify anomalies in network traffic and user behavior that deviate from normal patterns, enabling proactive cybersecurity.
Security.	Difficulty detecting hidden threats among millions of log records.	Neural networks identify anomalies in network traffic and user behavior that deviate from normal patterns, enabling proactive cybersecurity.
Development.	Technical debt and module inefficiency.	AI-based code analysis tools detect code smells and provide concrete, economically justified refactoring recommendations.

**6.2 Research on the methodology for testing software**

The development of cloud technologies, containerization, and microservice architectures has significantly changed software quality requirements. Under modern conditions, performance, resilience, and scalability testing become particularly

important, as these characteristics determine a system’s ability to operate under high loads and rapidly adapt to changes in real environments. The analysis (table 2) demonstrates substantial advantages of cloud-based testing across almost all critical parameters. The most pronounced differences are observed in scalability, environment provisioning speed, economic efficiency, and parallel test execution capabilities. At the same time, it should be noted that the cloud approach requires new competencies from teams: understanding cloud platform architectures, proficiency with infrastructure-as-code tools, knowledge of containerization and orchestration principles, and awareness of cloud provider pricing models.

**Table 2 – Comparative characteristics of traditional and cloud testing**

Criterion	Traditional testing	Cloud testing
Scalability.	Limited by physical resources; expansion takes weeks or months.	Virtually unlimited; scaling within minutes.
Cost model.	High capital investments and constant operational expenses.	Pay only for actual usage time.
Environment setup time.	2–4 weeks for a new environment.	Minutes or hours using infrastructure as code.
Geographical distribution.	Complex and economically impractical.	Global deployment supported in most regions.
Environment isolation.	Difficult; shared resources cause conflicts.	Easy via virtualization and containerization, full task-level isolation.
Reproducibility.	Low; depends on manual configuration.	High due to infrastructure as code.
CI/CD integration.	Requires significant effort and specialized tools.	Native support via managed services.
Test parallelism.	Limited by available machines.	Hundreds or thousands of parallel executions.
Maintenance and updates.	Requires specialized personnel.	Partially automated by the provider, reduced operational burden.



The initial learning curve may be steep, especially for teams with deep experience in traditional approaches. In addition, systems with high security requirements or regulatory constraints may require a hybrid approach, where critical components or sensitive data remain on-premises, while the cloud is used for scalable testing with synthetic or anonymized data.

Specific cloud testing scenarios provide unique opportunities unattainable in traditional approaches. Elasticity testing verifies automatic adaptation to traffic growth or decline, which is critical for applications with uneven workloads—such as e-commerce platforms during seasonal sales, media services during major content releases, or financial systems during market openings. Continuous testing integrates quality verification at every stage of the delivery pipeline, enabling constant validation and immediate detection of defects after changes are introduced. This transforms testing from a separate project phase into a continuous practice integrated into all aspects of development. Testing in environments *максимально* close to production eliminates the classical discrepancy between test and production environments. The ability to rapidly replicate production infrastructure allows testing under conditions nearly identical to real operational environments.

From an economic perspective, cloud testing democratizes access to high-quality testing infrastructure. A startup with a limited budget can perform comprehensive load testing for several hundred dollars by using resources only for a few hours before a critical release. Large enterprises benefit from scale, as they can execute thousands of parallel tests without maintaining massive proprietary data centers.

The cloud computing market in 2024–2025 is characterized by the dominance of three major platforms:

- Amazon Web Services;
- Microsoft Azure;
- Google Cloud Platform.

According to Synergy Research Group, as of Q4 2025, the combined market share of these three providers exceeds 65% of the global cloud infrastructure market. Each platform has unique characteristics that make it more or less suitable for specific testing



scenarios.

### 1. Amazon Web Services.

Amazon Web Services remains the undisputed market leader with an approximate share of 32% as of early 2025. Launched in 2006, AWS has the longest history among cloud providers, which is reflected in the maturity of its services and ecosystem. The platform offers more than 200 full-featured services covering nearly all aspects of cloud computing—from basic compute resources to specialized services for machine learning, IoT, blockchain, and quantum computing.

For testing system construction, AWS provides a wide range of relevant services. Amazon EC2 offers virtual servers with flexible configuration options, including instance types optimized for general purpose, compute, memory, storage, and other workloads.

AWS Elastic Container Service and Amazon Elastic Kubernetes Service provide managed container orchestration platforms. ECS is AWS's proprietary solution with deep ecosystem integration, while EKS offers standard Kubernetes with a managed control plane. Both services support AWS Fargate, a serverless container execution model where AWS fully manages the underlying infrastructure, allowing users to focus on resource requirements and application logic.

### 2. Microsoft Azure.

Microsoft Azure ranks second with a market share of approximately 23%. Launched in 2010 as Windows Azure, it was later renamed Microsoft Azure to reflect its evolution from a Windows-centric platform to a full-fledged multi-platform ecosystem supporting Linux, open-source technologies, and diverse programming languages.

Azure is particularly strong in the enterprise segment due to deep integration with the Microsoft ecosystem. Organizations using Windows Server, Active Directory, SQL Server, SharePoint, Office 365, or .NET achieve optimal integration with Azure. Hybrid scenarios combining on-premises infrastructure and cloud resources are also a strong area for Azure, enabled by solutions such as Azure Arc and Azure Stack.

Azure Virtual Machines provide a wide range of configurations supporting both



Windows and Linux. Azure Container Instances allow rapid container execution without managing underlying infrastructure, similar to AWS Fargate. Azure Kubernetes Service delivers a fully managed Kubernetes platform with strong ecosystem integration. Azure DevOps and Azure Pipelines offer a comprehensive CI/CD platform with native support for various development technologies.

A unique Azure feature is Azure DevTest Labs, a specialized service for rapid test environment provisioning with built-in cost control mechanisms. Azure also offers competitive pricing, particularly for organizations with existing Microsoft enterprise licenses through the Azure Hybrid Benefit program. Limitations include a smaller global footprint compared to AWS, sometimes less consistent documentation, and certain legacy platform aspects.

### 3. Google Cloud Platform.

Google Cloud Platform holds approximately 11% of the market but stands out for technological innovation, particularly in big data, machine learning, container technologies, and global networking. Launched in 2008 with Google App Engine, GCP has evolved into a full-featured cloud platform.

Google Compute Engine provides virtual machines with flexible configurations and an innovative pricing model with automatic sustained-use discounts. Spot instances offer up to 80% cost savings for non-critical workloads. Google Kubernetes Engine is considered the most mature managed Kubernetes platform, reflecting Google's role as Kubernetes' creator.

Cloud Run enables fully managed serverless container execution with automatic scaling from zero. Cloud Build and Cloud Deploy provide CI/CD capabilities tightly integrated with GCP infrastructure. A major advantage of GCP is Google's global network infrastructure, one of the largest and fastest in the world, ensuring low latency and high throughput for distributed system testing. Limitations include fewer available services compared to AWS and Azure and fewer enterprise-specific compliance certifications.



## **Conclusions**

This paper examines modern approaches to the intelligent design of corporate information systems in the context of digital transformation and increasing IT architecture complexity. It demonstrates that the integration of artificial intelligence, machine learning, simulation modeling, and data-driven architectures enables a transition from reactive to proactive and self-optimizing CIS management. The effectiveness of digital twins in reducing architectural change risks and technical debt is substantiated. The obtained results confirm the feasibility of implementing intelligent approaches as a methodological foundation for the design, optimization, and evolution of modern corporate information systems.