Enhancing adaptive systems with Intelligent Agents in Microservice Architectures: Opportunities and Challenges *

Roman Lysenko[1*,†], Oleksa Skorokhoda[2,†]

[1] *EPAM Systems, Lviv, 79048, Ukraine*

[2] *Lviv Polytechnic National University, Lviv, 79013, Ukraine*

Abstract

The integration of intelligent agents into adaptive systems based on microservice architecture offers significant advantages in automation, scalability, and resilience. These agents enable real-time system monitoring, anomaly detection, and autonomous decision-making, improving system efficiency and fault tolerance. The modular nature of microservices facilitates flexible updates and independent scaling, reducing operational overhead. However, this integration also introduces challenges, including coordination complexities, security risks, and increased computational overhead. This paper explores the benefits, architectural considerations, and key challenges of integrating intelligent agents with adaptive microservice-based systems, providing insights into optimizing system performance while addressing potential limitations.

Keywords

Artificial intelligence, intellectual systems, intelligent agents, adaptive systems, microservices

1. Introduction

The rapid evolution of adaptive systems, particularly those designed using microservice architecture, has significantly transformed how complex systems are built, maintained, and optimized. Microservices, with their modular design, allow for flexible and scalable system development, but their full potential is realized when combined with intelligent agents. These agents can autonomously monitor, manage, and adapt system operations in real-time, enabling systems to respond proactively to dynamic changes in environment and demand.

This paper aims to investigate the integration of intelligent agents with adaptive systems based on microservice architecture, focusing on their role in improving system performance, scalability, security, and resilience. We explore how intelligent agents enhance the capabilities of microservice-based solutions by providing real-time monitoring, self-healing mechanisms, and cost-effective resource management. At the same time, we examine the challenges that arise from such integration, including coordination between agents, security concerns, and system complexity.

Adaptive systems in software are systems that can automatically adjust their behavior or structure depending on the environment in which they operate. The primary goal of such systems is to ensure flexibility, efficiency, and resilience when facing changing requirements or operating conditions.

Examples of adaptive systems include:

- Cloud computing systems, which can scale resources according to the load.
- Smart grids, which adjust parameters depending on electricity consumption.
- Artificial intelligence systems, which learn and adapt based on data they receive.

Such adaptability allows systems to operate more efficiently in a changing environment or with limited resources.

1 *SmartIndustry 2025: 2nd International Conference on Smart Automation & Robotics for Future Industry, April 03-05, 2025, Lviv, Ukraine*

* Corresponding author.

2 These authors contributed equally.

roman.r.lysenko@lpnu.ua (R. Lysenko); oleksa.v.skorokhoda@lpnu.ua (O. Skorokhoda);

0009-0009-2353-9272 (R. Lysenko); 0000-0002-1455-8553 (O. Skorokhoda);

Table 1
Main Characteristics of Adaptive Systems

Characteristic	Description
Self-configuration	The system can adjust its parameters according to current needs.
Self-optimization	The system continuously improves its processes based on the analysis of its performance.
Self-healing	The ability to automatically detect and correct errors that occur during operation.
Self-protection	The system adapts to external threats, such as cyber-attacks.

This paper aims to analyze the integration of intelligent agents into adaptive systems designed using microservice architecture. The primary objectives are:

- To examine how intelligent agents enhance system adaptability, security, and efficiency in microservice-based environments.
- To identify key challenges, including agent coordination, security risks, and performance trade-offs, that arise in such integrations.
- To explore architectural and implementation considerations necessary for optimizing adaptive microservice-based systems with intelligent agents.
- To provide insights into practical applications, demonstrating real-world benefits and challenges of this approach across various industries.

This research contributes to the understanding of how intelligent agents can be leveraged to improve the resilience and autonomy of modern distributed systems while highlighting strategies to mitigate associated risks.

2. Architectural, Development, and Maintenance Challenges in Adaptive Systems

Adaptive systems offer substantial advantages, but their development and maintenance pose significant challenges. Designing architectures that dynamically adjust during runtime while maintaining system stability remains a core challenge in adaptive systems [1].

Table 2
Comparison of adaptive systems based on microservices vs. monolithic architectures.

Criterion	Microservices architecture	Monolithic architecture
Flexibility	High	Low
Scalability	Horizontal, flexible	Vertical, limited
Maintenance	Easy, independent components	Difficult, changes affect the entire system
Performance	High with proper configuration	Higher in small projects

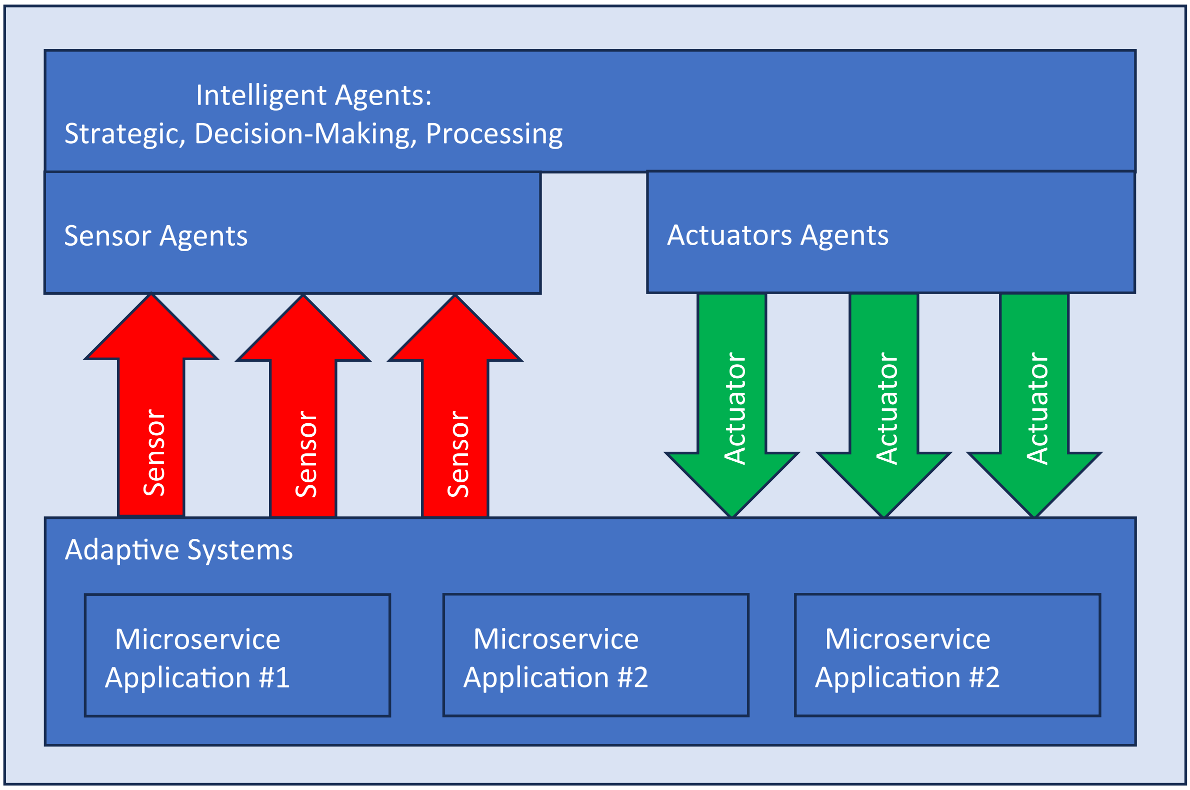

Figure 1: High-level structure and interaction of model of integration intelligent agents with adaptive systems which based on microservice architecture.

Below are the main issues faced by architects and developers of adaptive systems:

1. Architectural Issues

- Modularity and flexibility: Adaptive systems require a high degree of modularity so that each component can be modified or updated without negatively affecting the entire system. This complicates the design due to the need for clear decomposition of components.
- Dynamic configuration: Creating an architecture capable of dynamically changing its configuration during operation is difficult because it requires anticipating and controlling possible adaptation paths and their consequences.
- Balancing performance and adaptability: Increasing adaptability often leads to increased system complexity, which can reduce performance, creating a challenge for architectural optimization.

2. Development Issues

- Managing complexity: The more adaptation scenarios there are, the harder it is for developers to control system behaviour. The increase in system complexity requires advanced mechanisms for managing adaptation at both design and runtime [2].
- Modeling environment and adaptation scenarios: Developers need to account for a wide range of potential conditions in which the system may operate. This includes simulating different contexts and creating models of adaptive behavior.
- Implementing self-learning mechanisms: If adaptability includes machine learning or self-learning, appropriate algorithms must be integrated, requiring additional resources and expertise in artificial intelligence and data science.

3. Maintenance and Operational Issues

- Monitoring and analysis: To ensure adaptability, monitoring systems must be implemented to collect real-time data on the system’s performance and identify the need for changes. However, continuous monitoring increases infrastructure costs.

- Unpredictable behaviour: It is difficult to predict how the system will behave in new conditions, which can lead to unexpected errors or conflicts between adaptive processes. Managing unpredictable behaviour in adaptive systems is critical for ensuring long-term stability [3].
- Continuous updating requirements: Due to constantly changing requirements and external conditions, adaptive systems need continuous updates to address new challenges or improve adaptability.
- Security: Adaptive systems are more prone to vulnerabilities due to their complexity and dynamic changes. Ensuring security becomes a critical task, especially when the system constantly adapts to new conditions and configurations.

4. Performance and Optimization Issues

- Real-time optimization: Systems must perform adaptation in real-time without significant delays or performance degradation, which may require considerable computational resources.
- Conflicts between different adaptation scenarios: There may be situations where adaptive mechanisms work against each other, complicating the maintenance of overall system stability and coherence.

Thus, the architecture, development, and maintenance of adaptive systems are complex due to the need to integrate flexibility and resilience in a changing environment. This requires careful planning, modelling, and continuous monitoring, as well as a high level of technical expertise.

The problem of monitoring and controlling an adaptive system using intelligent agents presents several challenges due to the complexity and dynamic nature of such systems. Here are the key issues:

- Real-time Monitoring

  Intelligent agents are tasked with continuously collecting data from various components of the adaptive system, assessing performance, and detecting potential issues. Given the complexity of these systems, the challenge is to process and analyse data in real-time without overloading the system or affecting performance [4]. Handling large data volumes from distributed sources efficiently is critical to ensuring the system's responsiveness. Usage of well-known solutions for system monitoring (as example: Grafana or Prometheus) provides the capabilities to track and analyse general system environment parameters. At the same time, intelligent agents can also use these solutions for data collection of internal states from each microservice, but this will require custom implementation to maintain conceptual features.
- Coordination and Communication

  Multiple agents deployed across different system nodes must coordinate and communicate effectively. Poor synchronization or miscommunication can lead to inconsistent or conflicting adaptations, causing unintended consequences [5]. Efficient, low-latency communication is essential, especially in large-scale distributed systems, to provide a comprehensive view of the system's state and avoid errors. Deep view on cross-component communication design of each unique system can require strong attention to integration design of communication between intelligent agents and system nodes.
- Decision-Making and Adaptation

  When agents detect an issue or opportunity for optimization, they must decide whether to initiate adaptation. This decision-making process is complex, as agents must consider both immediate and long-term consequences. Agents must balance system performance optimization with maintaining stability, which often requires trade-offs between short-term and long-term goals [1]. Additionally, conflicts can arise when multiple agents trigger conflicting adaptations, complicating overall system behaviour. Each agent can be responsible for dynamic maintenance of solution, based on an internal state of the system node. This can require dynamical analysis of internal system processes operability and adaptability based on the results of internal execution flow.

- Scalability

  As the system scales, so does the number of intelligent agents required to monitor and control it. Ensuring the scalability of agent-based monitoring without introducing overhead is a significant challenge. The system must adapt to increasing complexity without diminishing the agents' effectiveness or introducing inefficiencies [4]. When intelligent agents are integrated into the internal system nodes, it can require massive data collection from the internal system nodes states. As a result, this complex multi-agent monitoring is heterogeneously integrated into the cross-services design, and it can require complete view of full system internal state. Especially, it is required for deep state analysis by strategic intelligent agents which build possible suggestions as solution enhancements. Classical cloud-based solutions provide the capabilities for scaling each system node separately based on general environment parameters. The implementation of scalability based on data-driven dependencies in multi-layered intelligent agent's environment will require additional customization of classical solutions (as example: Kubernetes provide the possibility to build custom controllers).

- Security and Trust

  Ensuring the security and integrity of intelligent agents is critical since these agents operate autonomously and make decisions that can impact the system's stability. Any compromise in agent security can lead to system vulnerabilities, impacting overall trust in the system's adaptability [5]. Safeguarding agent communications and decision-making processes is vital to prevent malicious interference.

- Error Handling and Recovery

  Intelligent agents must detect and manage errors without causing system disruptions. Robust error-handling mechanisms are necessary to ensure graceful recovery from failures and prevent cascading issues across the system [1]. Agents must coordinate effectively to restore normal operation while maintaining the system's adaptability. Error handling and recovery in complex microservice architectures sometimes can be critical from the data importance perspective. Some solutions will require deep data analysis or debugging which are time consuming. Intelligent learning agents (ML-based) which track and analyse the complete solution state can provide suggestions for architectural improvements based on identified errors.

Using intelligent agents to monitor and control adaptive systems presents challenges in real-time data processing, coordination, decision-making, scalability, security, and error recovery. Addressing these challenges requires careful design of the agents and the system architecture to ensure efficiency, security, and responsiveness.

3. The methods and materials

Developing adaptive systems based on microservice architecture, monitored by intelligent agents, is a modern approach to creating flexible, scalable, and resilient software systems. This concept combines the strengths of microservice architecture with the autonomous decision-making capabilities of intelligent agents to enhance adaptability and performance in dynamic environments:

- Microservice Architecture as a Foundation

  Microservices are small, loosely coupled services that work independently to perform specific functions within a larger system. Each microservice is responsible for a particular aspect of the system's functionality, and these services communicate with each other through lightweight protocols, typically via APIs. This architecture is highly modular, allowing developers to easily modify, scale, or replace individual services without affecting the entire system. The decentralized nature of microservices supports adaptive behaviour by enabling isolated changes in specific components without disrupting other parts of the system [6].

- Autonomous Monitoring and Adaptation by Intelligent Agents

  In an adaptive system built on microservices, intelligent agents play a critical role in monitoring the system's health and performance. These agents observe the behaviour of individual microservices and the overall system in real-time, collecting data on various metrics such as resource utilization, response times, error rates, and security threats. Based on the data, agents can make decisions autonomously, such as reallocating resources, restarting failed services, or adjusting configurations to optimize performance.

  For instance, if an agent detects that a particular microservice is experiencing high traffic, it can automatically scale up resources allocated to that service, ensuring smooth performance. Conversely, if certain services are underutilized, the agent can reduce resources, optimizing the system's overall efficiency. Agents can also identify and mitigate failures, helping the system recover quickly from unexpected issues [7].

- Dynamic Reconfiguration and Self-Healing

  One of the primary advantages of combining microservices with intelligent agents is the ability to dynamically reconfigure the system without manual intervention. Intelligent agents continuously analyse the system's state and can rewire services in response to changes in the environment or user demands. For example, if a particular service needs to be updated or replaced, agents can route traffic to alternative services, ensuring minimal downtime.

  Self-healing capabilities are also enhanced in this model. If a microservice fails, the intelligent agents can detect the issue and restart the service, or in more critical cases, spin up a replacement service in a different part of the infrastructure. This reduces system downtime and ensures that services remain operational even in the face of unexpected failures [8].

- Scalability and Flexibility

  Microservice-based architectures are inherently scalable, allowing for the easy addition or removal of services as needed. Intelligent agents further enhance this scalability by automating resource allocation and scaling decisions. As demand for certain services increases or decreases, agents can dynamically adjust the system to maintain performance levels while avoiding over-provisioning or under-utilization of resources.

  Flexibility is another key advantage. The modular nature of microservices allows developers to update or replace individual components without affecting the entire system. Intelligent agents facilitate this by managing the transition between old and new versions of services, ensuring smooth updates with minimal disruption [9].

- Decentralized Control and Reduced Complexity

  By utilizing microservices and intelligent agents, control over the system becomes more decentralized. Each microservice can operate independently, and intelligent agents handle local decision-making and adaptation. This reduces the complexity of managing a large, monolithic system and allows for more granular, efficient control over the system's behaviour.

  However, the use of intelligent agents requires careful coordination to avoid conflicts between adaptation decisions. For example, one agent might attempt to scale up a service, while another could be reducing resources elsewhere. Mechanisms for agent coordination, such as message passing or centralized decision-making in certain critical scenarios, can help mitigate these challenges [10].

- Security and Resilience

  Intelligent agents can enhance the security and resilience of microservice-based adaptive systems. By constantly monitoring system behaviour, agents can identify unusual patterns that might indicate security threats, such as denial of service attacks or unauthorized access attempts. In response, agents can take corrective actions like isolating affected services or dynamically adjusting security policies to mitigate risks.

Resilience is another strength of this model. In addition to self-healing mechanisms, intelligent agents can pre-emptively identify and resolve potential issues before they escalate into failures. This proactive approach helps maintain high availability and reduces the risk of catastrophic failures [11].

Example of function of self-configuration mechanism. An adaptive system automatically changes its parameters in response to external variations in workload. This process can be described by the equation:

$$P_{new} = P_{current} + \alpha \cdot \Delta E \quad (1)$$

where:

- P_{new} is the new level of system performance,
- $P_{current}$ is the initial performance level (e.g., the average value without adaptation),
- α is the adaptation coefficient (determining how quickly the system responds to changes),
- ΔE represents external workload variations (such as increased requests, failures, or environmental changes).

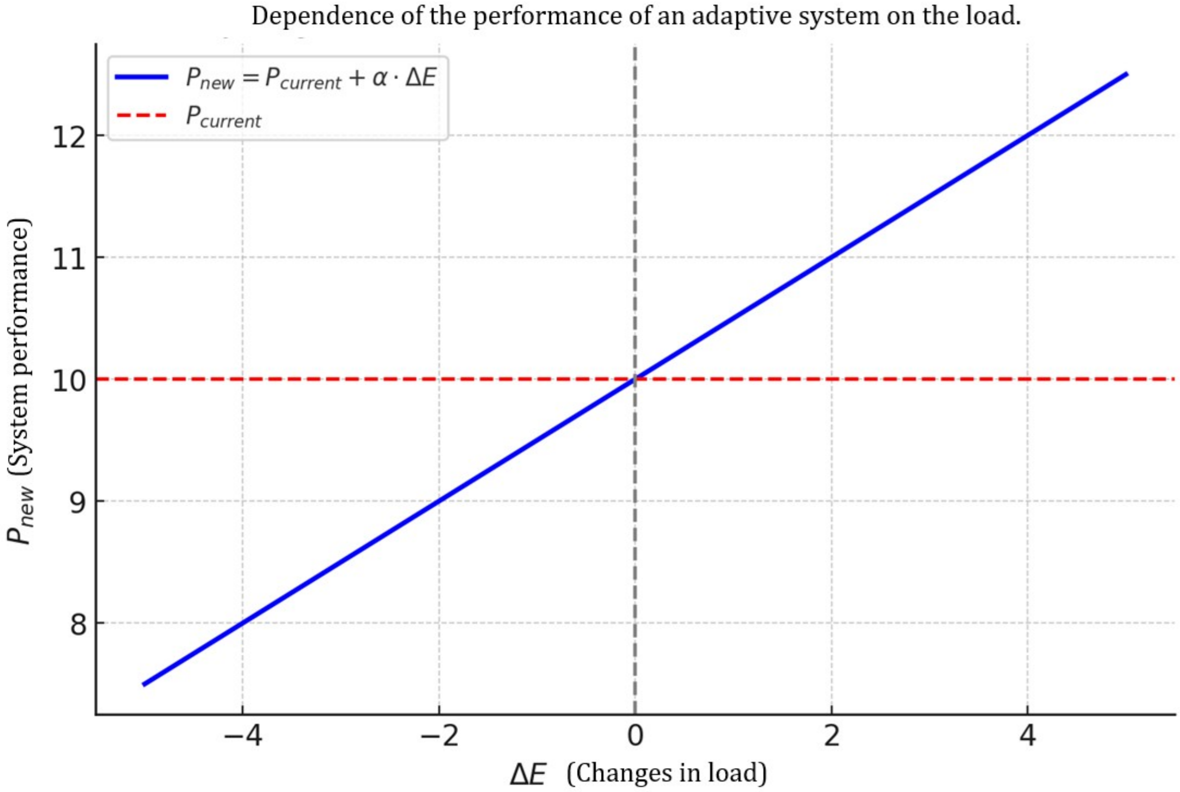

Figure 2: Dependence of the performance of an adaptive system on the load.

Explanation of the Graph:

- If the workload increases ($\Delta E > 0$), the system adapts and increases its performance.
- If the workload decreases ($\Delta E < 0$), the system can lower its performance to save resources.
- The red dashed line represents the initial performance level ($P_{current}$).
- The higher the adaptation coefficient (α), the faster the system adjusts its performance in response to changes.

This approach allows adaptive systems to dynamically react to workload variations, maintaining stable operation without resource overuse.

4. The results of research

The approach of developing adaptive systems using microservice architecture combined with intelligent agents yields several notable outcomes:

- Increased Flexibility

  The microservice architecture's modular design allows each service to evolve independently. This leads to a highly flexible system where individual components can be updated, replaced, or scaled without affecting the overall system. The use of intelligent agents further enhances this flexibility by enabling real-time adaptations to the system's behaviour in response to changing conditions [12].
- Improved Scalability

  Microservices naturally support horizontal scaling, where new instances of a service can be created as demand increases. Intelligent agents monitor resource utilization and automatically manage this scaling process, ensuring optimal performance while avoiding over-provisioning. This scalability makes the system well-suited to environments with fluctuating loads and high variability in user demands [13].
- Enhanced Resilience and Self-Healing

  By leveraging intelligent agents, adaptive systems become more resilient. Agents can detect failures or performance bottlenecks in real-time and take corrective actions, such as restarting services or rerouting traffic to healthy components. This self-healing capability ensures that the system can recover quickly from disruptions, minimizing downtime and maintaining service continuity [14].
- Autonomous Optimization

  Intelligent agents continuously monitor system performance and can optimize resource allocation and configurations based on current conditions. This leads to more efficient use of resources, as the system dynamically adjusts to maintain optimal performance. Over time, this results in cost savings and improved operational efficiency as the system operates with fewer manual interventions [15].
- Decentralized Control and Reduced Complexity

  The decentralized nature of microservices and the autonomous actions of intelligent agents reduce the complexity of managing the system. Instead of relying on centralized control, each microservice and agent operates independently, making local decisions. This decentralized approach simplifies system management, especially in large-scale systems, as there is less need for manual oversight or intervention [16].
- Improved Security and Proactive Threat Mitigation

  With continuous monitoring by intelligent agents, the system becomes more proactive in identifying and responding to security threats. Agents can detect anomalies in system behaviour that may indicate potential attacks and take immediate action, such as isolating compromised services or adjusting security configurations. This proactive approach reduces the risk of system breaches and enhances overall security [17].
- Faster and Easier Maintenance

  Due to the modular nature of microservices, maintaining and updating adaptive systems is faster and easier. Developers can make changes to individual services without needing to take down the entire system. Intelligent agents assist by managing service transitions, ensuring smooth updates, and reducing the potential for errors during maintenance processes [18].
- Cost-Effective Resource Management

Intelligent agents optimize resource usage by scaling services up or down based on real-time demand, which prevents wasteful over-provisioning and reduces infrastructure costs. This cost-effectiveness is especially valuable in cloud environments where resources are billed based on usage [15].

While the approach of developing adaptive systems using microservice architecture and intelligent agents offers numerous advantages, it also presents several drawbacks:

- Increased Complexity in Management

Managing a system based on microservices, especially one with intelligent agents, introduces significant complexity. Each microservice operates independently, which can lead to many services that need to be monitored, maintained, and updated. Coordinating the actions of multiple intelligent agents and ensuring they work in harmony without causing conflicts can be challenging. As the system scales, the complexity grows, requiring advanced tools and expertise to manage the interactions between services and agents effectively.

- Communication Overhead

Microservice architectures rely on inter-service communication, typically through APIs or messaging systems. As the number of microservices increases, the volume of communication between them grows as well, especially when system architecture and cross-services communication design were not refactored properly in time. This can lead to performance bottlenecks and increased latency, especially in large-scale systems where services are distributed across different servers or geographic locations. Intelligent agents, which monitor and manage these services, may also add to the communication overhead as they exchange data and coordinate actions in real-time.

- Resource Consumption

Running multiple microservices and intelligent agents requires significant computational resources. Each microservice operates independently, often needing its own instance of resources such as memory, CPU, and storage. Intelligent agents, which continuously monitor and optimize the system, also consume additional resources. This can lead to higher infrastructure costs, especially in cloud environments, where resources are billed based on usage.

- Coordination and Conflict Resolution

In an adaptive system with multiple intelligent agents, ensuring proper coordination is essential. However, there is a risk of conflicts when different agents make decisions that affect the same part of the system. For example, one agent might attempt to scale up a service, while another is scaling it down based on different criteria. These conflicts can lead to instability or suboptimal performance. Designing mechanisms for effective coordination and conflict resolution among agents is a challenging aspect of this approach.

- Security Challenges

While intelligent agents enhance security by detecting and mitigating threats in real-time, they also introduce new security concerns. The autonomous nature of agents means they need to be carefully designed to avoid being exploited or manipulated by malicious actors. Additionally, the decentralized nature of microservice architectures can make it harder to maintain a consistent security posture across all services. Each service may have its own vulnerabilities, and securing the communication between services and agents is critical to prevent breaches.

- Testing and Debugging Difficulties

  Testing and debugging adaptive systems built on microservice architecture can be more difficult compared to monolithic systems. The dynamic and distributed nature of microservices makes it challenging to trace and isolate issues, especially when they involve multiple services or when intelligent agents are autonomously making changes in real-time. Debugging interactions between services, identifying the root cause of performance issues, or understanding the impact of agent decisions often requires specialized tools and a deep understanding of the system's architecture.

- Overhead from Intelligent Agents

  While intelligent agents provide valuable automation, they also introduce overhead. The agents must be continuously running, consuming resources, and processing large amounts of data in real-time to monitor and make decisions. If not properly managed, this can result in additional load on the system, potentially reducing overall performance. Moreover, the algorithms used by agents for decision-making may need to be fine-tuned to avoid inefficient behavior, which could further complicate the system's management.

- Steep Learning Curve

  Implementing and maintaining an adaptive system with microservices and intelligent agents requires expertise in both microservice architecture and AI-driven automation. Developers and operations teams need to be familiar with distributed systems, microservice design patterns, agent-based systems, and real-time monitoring. This steep learning curve can increase development time and require more skilled personnel, raising the overall cost of implementation and maintenance.

5. Practical value

The practical implementation of adaptive systems based on microservice architecture, monitored and operated by intelligent agents, can be seen across various industries. These systems are used to optimise operations, enhance user experience, and reduce costs. Below are real-world examples and practical implementations that demonstrate how these technologies are applied:

1. E-commerce Platforms: Dynamic Scaling and Personalisation

   Example: An online retailer like Amazon utilises a microservice architecture where different services, such as inventory management, product recommendations, and payment processing, function independently.

   Implementation by Amazon:

   - During high-traffic periods, such as Black Friday, intelligent agents monitor user demand and dynamically scale specific services, like payment processing, to handle the increased load without overburdening the entire system.
   - These agents also analyse browsing behaviour and purchasing patterns to provide personalised recommendations in real-time, ensuring that the user receives tailored suggestions, which can boost sales.

   Benefit: This automated scalability prevents system crashes, reduces latency, and enhances the customer experience, all while keeping infrastructure costs manageable by scaling only what is needed.
2. Healthcare: Real-Time Patient Monitoring Systems

Example: Hospitals and healthcare providers use adaptive systems for patient monitoring in intensive care units (ICUs). A microservice architecture ensures that different aspects of patient care, such as vital signs tracking, medication management, and alerts, are handled independently.

Implementation by Philips IntelliVue Guardian:

- Intelligent agents continuously monitor patients' vital signs (heart rate, blood pressure, oxygen levels) in real-time. If an agent detects anomalies—like a sudden drop in blood pressure—it can immediately trigger an alert to medical staff and activate pre-programmed actions, such as adjusting medications.
- These systems can also integrate with patient history records and offer real-time recommendations based on a combination of current data and historical trends.

Benefit: The system reduces the risk of human error, ensures timely interventions, and allows medical staff to focus on more complex tasks, as many routine decisions and adjustments are made autonomously by the system.

3. Financial Services: Fraud Detection and Risk Management

Example: In the banking sector, institutions like JPMorgan Chase utilize adaptive systems to monitor transactions for fraud detection.

Implementation by JPMorgan Chase COiN:

- Microservices handle different banking functions, such as transaction processing, user authentication, and loan approval. Intelligent agents constantly analyse transaction patterns in real-time, flagging suspicious activities such as unusual withdrawals or transfers.
- The agents cross-reference the data with global fraud patterns and historical data from individual users to determine whether a transaction should be blocked or flagged for further review. If necessary, the system can take immediate action, like freezing an account or notifying the user, without waiting for human intervention.

Benefit: This system reduces response times for potential fraud cases, protects customer assets, and significantly lowers the risk of financial loss for both the bank and its clients.

4. Cloud Services: Automated Resource Management

Example: Cloud service providers like Google Cloud and AWS use adaptive systems to manage vast infrastructures and client resources efficiently.

Implementation by AWA Auto Scaling:

- Cloud services are broken down into microservices responsible for storage, computing, database management, etc. Intelligent agents monitor system performance, traffic, and resource utilisation. When there is a sudden surge in demand for cloud resources—like during a product launch or a viral event—the agents automatically allocate additional computing power and storage.
- Once the demand decreases, the agents deallocate resources to avoid over-provisioning and unnecessary costs.

Benefit: This adaptive resource management ensures that clients always have access to the necessary resources without experiencing slowdowns or outages, while also minimising costs through efficient allocation.

5. Smart Manufacturing: Predictive Maintenance and Automation

Example: In smart factories, like those operated by Siemens, adaptive systems manage production lines and equipment maintenance through predictive analytics.

Implementation by Siemens MindSphere:

- Microservices control various production processes, such as assembly, quality control, and packaging. Intelligent agents continuously monitor equipment performance, tracking vibration, temperature, and operational efficiency.

- If the system detects signs of wear or inefficiency in machinery, it schedules predictive maintenance before a breakdown occurs. In addition, intelligent agents can reassign workloads to other machines to ensure that production continues without disruption.

Benefit: This approach reduces downtime, improves operational efficiency, and extends the life of expensive machinery, ultimately saving manufacturers significant time and money.

6. Telecommunications: Network Optimisation and User Experience Enhancement

   Example: Telecom providers like Verizon and AT&T use adaptive systems for managing their networks, ensuring high-quality service for millions of users.

   Implementation by AT&T Network AI:

   - The network infrastructure is divided into microservices responsible for managing call routing, data services, and network traffic balancing. Intelligent agents constantly monitor the quality of the service each user experiences, detecting any network bottlenecks or latency issues.
   - When agents notice congestion in a particular region or node, they automatically reroute traffic to less congested areas or dynamically allocate more bandwidth to high-demand regions.

   Benefit: Users experience fewer dropped calls, faster internet speeds, and more reliable connectivity, all without manual intervention by network engineers.

7. Energy Sector: Smart Grid Management

   Example: Utility companies use adaptive systems for managing smart grids, ensuring efficient energy distribution and consumption.

   Implementation by General Electric Predix:

   - Microservices monitor different parts of the grid, including power generation, distribution, and consumption. Intelligent agents predict demand based on historical data and real-time inputs such as weather forecasts and user consumption patterns. When the system predicts a surge in demand—such as during a heatwave—agents adjust the distribution of power to prevent blackouts, or suggest alternative energy sources, like solar or battery storage.
   - Additionally, the system can monitor the health of grid infrastructure, identifying faults in power lines or transformers and alerting repair teams before major outages occur.

   Benefit: This ensures more reliable energy distribution, reduces the chances of blackouts, and helps optimise energy consumption, contributing to cost savings for both utilities and consumers.

6. Conclusions

Developing adaptive systems using microservice architecture, monitored and controlled by intelligent agents, offers numerous benefits. The modularity of microservices provides flexibility and scalability, while intelligent agents enhance system adaptability, resilience, and security. This approach allows systems to dynamically reconfigure in real-time, respond to changing conditions, and recover from failures, making it ideal for environments with fluctuating demands or unpredictable circumstances. Combining these two paradigms results in systems that are not only highly adaptive but also easier to maintain and evolve over time.

Despite its benefits, the combination of microservice architecture and intelligent agents in adaptive systems comes with challenges related to complexity, resource consumption, security, and management. Addressing these drawbacks requires careful system design, robust coordination mechanisms, and the use of advanced monitoring and debugging tools. As a result, this approach may not be suitable for all applications, particularly those with limited resources or less dynamic requirements.

Practical implementations of adaptive systems with microservice architectures and intelligent agent monitoring are transforming industries by improving operational efficiency, enhancing customer experience, and reducing costs. These systems enable businesses to dynamically respond to real-time changes, predict potential issues before they arise, and automate decision-making processes, leading to better overall performance.

The development of adaptive distributed systems based on microservice architecture, with intelligent agents for monitoring and management, holds significant scientific value across various fields leading to more resilient and efficient systems. Key technical design attributes are:

1. Flexibility and Modularity: Microservices allow complex systems to be broken down into independent components, enabling easy scaling and adaptation. This is crucial for creating systems that can quickly evolve with changing requirements.
2. Adaptability and Self-Learning: AI agents use machine learning to monitor and predict system behavior, allowing real-time adaptation to environmental changes, advancing autonomous decision-making systems.
3. Efficiency in Computing: These systems optimize computational and network resource use, fostering the development of energy-efficient distributed computing solutions, critical in handling big data and cloud systems.
4. Management of Complex Systems: AI agents autonomously manage large-scale infrastructures, detecting anomalies and preventing failures in critical sectors like energy, healthcare, and transportation.
5. Scalability: Microservices enable scalable integration of AI agents into both small and large systems, improving overall performance while reducing the need for human oversight.
6. Interdisciplinary Research: This technology supports cross-disciplinary studies in fields such as computer science, engineering, and social sciences, offering new insights into automation, decision-making, and resource optimization.
7. Security and Ethics: As these systems evolve, they raise new challenges in data security, transparency, and ethical standards, driving research in developing safe, reliable, and ethically responsible AI systems.

The scientific significance of adaptive systems with microservice architectures and AI agents lies in their potential to automate complex processes, enhance system efficiency and security, and foster interdisciplinary research in emerging technologies.

Declaration on Generative AI

During the preparation of this work, the authors utilized ChatGPT and LanguageTool to identify and rectify grammatical, typographical, and spelling errors. Following the use of these tools, the authors conducted a thorough review and made necessary revisions, and accept full responsibility for the final content of this publication.

References

[1] Salehie, M., & Tahvildari, L. (2009). Self-adaptive software: Landscape and research challenges. ACM Transactions on Autonomous and Adaptive Systems (TAAS), 4(2), 1-42.

[2] Lewis, G. A., Morris, E. L., Simanta, S., & Wrage, L. (2009). Common themes in dynamic system adaptation. Journal of Systems and Software, 83(12), 1865-1872.

[3] Andersson, J., de Lemos, R., Malek, S., & Weyns, D. (2013). Modeling dimensions of self-adaptive software systems. Software Engineering for Self-Adaptive Systems II, 1-30.

[4] Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. Computer, 36(1), 41-50.

[5] Weyns, D., Andersson, J., & Schmerl, B. (2012). On patterns for decentralized control in self-adaptive systems. Software Engineering for Self-Adaptive Systems II, 76, 76-107.

[6] Groves, M. D., Richardson, R., & Whitesell, S. (2025). Pro microservices in .NET 6: With examples using ASP.NET Core 6, MassTransit, and Kubernetes.
[7] Makarenko, A., & Otrokh, S. (2023). Using SDR as sensor nodes in an intelligent radio emission data collection network management system. Scientific Notes of the State University of Information and Communication Technologies, (1), 15-23. Retrieved from journals.dut.edu.ua.
[8] Magableh, B. (2019). A deep recurrent Q network towards self-adapting distributed microservices architecture.
[9] Dragoni, N., Lanese, I., Larsen, S. T., Mazzara, M., Mustafin, R., & Safina, L. (2017). Microservices: How to make your application scale. Perspectives of System Informatics, 95-104.
[10] Mori, K. (2019). Autonomous decentralized systems: Achievements and future challenges. Proceedings of the 2019 IEEE International Symposium on Autonomous Decentralized Systems, 1-8.
[11] Leines-Vite, L., Pérez-Arriaga, J. C., & Limón, X. (2021). Information and Communication Security Mechanisms For Microservices-based Systems.
[12] Ford, N., & Parsons, D. (2017). Building Evolutionary Architectures: Support Constant Change. O'Reilly Media.
[13] Leitao, P., Karnouskos, S., Ribeiro, L., Lee, J., & Strasser, T. (2016). Smart Agents in Industrial Cyber-Physical Systems. Proceedings of the IEEE, 104(5), 1085-1101.
[14] Rieger, C. G. (2014). Resilient control systems: Practical metrics basis for defining mission impact. In Proceedings of the 7th International Symposium on Resilient Control Systems (pp. 1-6). IEEE.
[15] Sharma, S., Nag, A., Cordeiro, L., Ayoub, O., & Tornatore, M. (2020). AI-assisted virtualization software.
[16] Zilberstein, S. (2021). Decentralized Partially Observable Markov Decision Processes. University of Massachusetts.
[17] Stiennon, N., et al. (2020). Learning to Summarize with Human Feedback. Advances in Neural Information Processing Systems, 33, 3008-3021.
[18] Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. Computer, 36(1), 41-50.