



УДК 004.896

DOI: <http://dx.doi.org/10.21686/1818-4243-2024-6-53-66>

А.А. Брызгалов

Российский экономический университет им. Г.В. Плеханова,
Москва, Россия

Микросервисная реализация многоагентных систем сетевых предприятий*

Цель. Целью данной статьи является разработка методов и алгоритмов микросервисной реализации многоагентных систем, поддерживающих функции цифровых двойников сетевого предприятия. Особое внимание уделяется реализации функций сбора данных, мониторинга, диагностики, защиты информации и принятия решений, обеспечивающих эффективное взаимодействие между агентами продуктов и агентами ресурсов на основе соглашений об уровне сервиса (SLA).

Методы исследования. Исследование использует методы построения микросервисной архитектуры для разделения и автономного выполнения функций с целью повышения уровня обслуживания SLA и надежности системы. Для усиления устойчивости системы к угрозам безопасности в алгоритмы добавлены этапы контроля доступа, шифрования данных и анализа аномалий, составляющие виртуальный периметр безопасности. Методология включает адаптивное управление параметрами SLA, использование распределенной обработки данных и применение аналитических алгоритмов для принятия решений, а также механизмы для выявления потенциальных угроз и несанкционированного доступа.

Результаты. Предложенные методы и алгоритмы позволяют создавать гибкую и масштабируемую многоагентную систему,

способную адаптироваться к изменяющимся условиям и обеспечивать устойчивую и безопасную работу цифровых двойников. Использование SLA для регулирования взаимодействия агентов улучшает согласованность их действий, повышает точность мониторинга и защищает данные, поддерживая высокую надежность и безопасность системы.

Заключение. Результаты показывают, что микросервисный подход, использование SLA и интеграция мер безопасности значительно повышают эффективность и надежность многоагентных систем, позволяя агентам адаптироваться к изменениям, оперативно реагировать на отклонения и предотвращать возможные угрозы. Применение данных методов может существенно улучшить управление активами в сетевых предприятиях, поддерживая их конкурентоспособность, устойчивость и безопасность в условиях цифровой трансформации.

Ключевые слова: многоагентная система, микросервисы, цифровой двойник, SLA, сбор данных, мониторинг, диагностика, принятие решений, сетевое предприятие, архитектура системы.

А.А. Bryzgalov

Plekhanov Russian University of Economics, Moscow, Russia

Microservice Implementation of Multi-Agent Enterprise Networking Systems

Purpose. The purpose of this article is to develop methods and algorithms for microservice implementation of multi-agent systems that support the functions of digital twins of a networked enterprise. Particular attention is paid to the implementation of data collection, monitoring, diagnostics, information protection and decision-making functions that ensure effective interaction between product agents and resource agents based on service level agreements (SLAs).

Research methods. The research uses methods of creating a microservice architecture to separate and autonomously perform functions in order to optimize SLA and increase system reliability. To strengthen the system's resilience to security threats, access control, data encryption and anomaly analysis stages that form a virtual security perimeter have been added to the algorithms. The methodology includes adaptive management of SLA parameters, the use of distributed data processing and the application of analytical algorithms for decision-making, as well as mechanisms to identify potential threats and unauthorized access.

Results. The proposed methods and algorithms make it possible to create a flexible and scalable multi-agent system capable of adapting to changing conditions and ensuring stable and safe operation of digital counterparts. Using SLA to regulate the interaction of agents improves the consistency of their actions, increases the accuracy of monitoring and protects data, maintaining high reliability and security of the system.

Conclusion. The results show that the microservice approach, the use of SLA and the integration of security measures significantly increase the efficiency and reliability of multi-agent systems, allowing agents to adapt to changes, respond promptly to deviations and prevent possible threats. The use of these methods can significantly improve asset management in network enterprises, maintaining their competitiveness, sustainability and security in the context of digital transformation.

Keywords: multi-agent system, microservices, digital twin, SLA, data collection, monitoring, diagnostics, decision-making, networked enterprise, system architecture.

* Исследование выполнено за счет гранта Российского научного фонда (проект № 22-11-002821) <https://rscf.ru/project/22-11-00282/>

Введение

Современные предприятия сталкиваются с необходимостью адаптации к постоянно меняющимся условиям рынка, что вызывает необходимость эффективного управления и быстрого принятия решений на основе актуальных данных о состоянии производственных систем. В этой связи большое внимание уделяется многоагентным системам, способным поддерживать сложные процессы в динамичных условиях, в которых каждый актив с помощью многоагентной технологии представляется цифровым двойником [1]. Такие цифровые двойники играют ключевую роль, поскольку позволяют моделировать и оптимизировать операции, обеспечивая согласованное взаимодействие между различными активами и бизнес-процессами. Центральное взаимодействие в многоагентной системе происходит между агентом продукта и агентом ресурса, которые обмениваются данными посредством соглашений об уровне сервиса (SLA), что помогает обеспечить выполнение требований и достигать заданных бизнес-целей [2].

Соглашение об уровне сервиса SLA (Service Level Agreement) в рамках многоагентных систем не только регулирует качество предоставляемого сервиса, но и определяет параметры для эффективного сбора и обмена данными, что является основой для мониторинга состояния агентов и принятия решений. Использование SLA позволяет агентам продукта и ресурса взаимодействовать в режиме реального времени, настраивая обмен информацией в зависимости от состояния и потребностей системы. Однако для масштабируемой реализации таких процессов необходима архитектура, обеспечивающая гибкость и автономность каждого

агента, что достигается с помощью микросервисного подхода. Микросервисы позволяют разрабатывать и внедрять независимые функции для каждого агента, предоставляя тем самым структурированное решение, поддерживающее эффективное взаимодействие агентов на всех уровнях [3].

Применение микросервисов в многоагентных системах находит широкую поддержку в исследованиях [4],[5],[6], поскольку такая архитектура предоставляет возможность гибкого масштабирования и адаптации агентов к изменяющимся условиям. В отличие от монолитных систем, микросервисный подход позволяет каждому агенту действовать как независимый сервис, что повышает общую отказоустойчивость и снижает риски сбоев при взаимодействии между агентами. Это особенно важно для реализации SLA, так как каждый агент может функционировать независимо и в случае отказа перезапускаться отдельно, не нарушая работу других. Многочисленные исследования [7],[8],[9] показывают, что микросервисы способствуют распределению нагрузки и позволяют интегрировать новые компоненты и функции без необходимости изменения всей системы, что делает их идеальными для многоагентных систем, требующих высокой гибкости и адаптивности.

Исследования в области многоагентных систем [10],[11] подтверждают, что микросервисы обеспечивают высокую масштабируемость и модульность, что критически важно для систем, работающих в условиях постоянных изменений и необходимости быстрого принятия решений. Работы, посвященные цифровым двойникам [12],[13], показывают, что их успешное применение позволяет улучшить управление процессами, сократить издержки и минимизировать

влияние человеческого фактора на качество данных. В многоагентных системах основная роль цифровых двойников заключается в том, чтобы поддерживать функциональность агентов, выполняющих задачи, такие как сбор данных о состоянии активов, мониторинг их параметров и принятие решений по управлению. Разработка методов для микросервисной реализации функций агентов позволяет каждому агенту не только эффективно взаимодействовать с другими агентами, но и своевременно адаптироваться к изменениям в системе.

Существует несколько подходов к управлению и реализации функций сбора данных, мониторинга и диагностики в многоагентных системах, и каждый из них подчеркивает важность использования микросервисной архитектуры для достижения оптимального взаимодействия агентов. Одним из таких подходов является применение методов распределенного мониторинга, где каждый агент контролирует определенные параметры и передает данные другим агентам согласно SLA [14]. Это позволяет минимизировать избыточность данных и обеспечить высокую производительность, что особенно полезно в условиях высокой нагрузки. Современные алгоритмы также предлагают использование адаптивных SLA, которые могут изменяться в зависимости от условий работы системы, что позволяет агентам оперативно реагировать на изменения состояния активов и внешних факторов, таких как увеличение нагрузки или изменение параметров производительности [15].

Актуальные исследования [16],[17],[18] также подчеркивают важность внедрения методов и алгоритмов для принятия решений на основе данных, полученных в процессе мониторинга и диагностики. В многоагентных системах циф-

ровые двойники, действуя в рамках SLA, анализируют данные мониторинга и принимают решения, направленные на оптимизацию использования ресурсов и предотвращение возможных сбоев. Применение микросервисного подхода здесь позволяет каждому агенту использовать специальные алгоритмы принятия решений, что упрощает процесс управления и повышает гибкость системы. Разработка алгоритмов для микросервисной реализации таких функций позволяет создать систему, способную адаптироваться к новым условиям и эффективно использовать данные для принятия своевременных решений, поддерживающих работу всех агентов.

Применение SLA позволяет стандартизировать обмен данными и управление ресурсами между агентом продукта и агентом ресурса, обеспечивая согласованность действий и оперативное реагирование на изменения состояния активов. Таким образом, SLA выступает основным связующим элементом, который обеспечивает надежное взаимодействие и определяет границы ответственности агентов, минимизируя риски некорректного выполнения операций. Научные работы [19],[20], посвященные SLA в многоагентных системах, подчеркивают важность использования соглашений для стандартизации и обеспечения качества обмена информацией. SLA определяют структуру и протоколы взаимодействия между агентами, что помогает избежать потерь данных и минимизировать риски сбоев. Актуальные исследования [21],[22] подчеркивают, что SLA позволяет достичь высокого уровня надежности в многоагентных системах, так как каждое нарушение параметров SLA может быть оперативно зафиксировано, а сбои — минимизированы благодаря быстрому реагиру-

нию. Тем не менее, существует необходимость в разработке более эффективных алгоритмов, которые позволят улучшить взаимодействие агентов в микросервисной архитектуре, особенно в условиях работы с большими данными.

Современные подходы к разработке и реализации SLA в многоагентных системах включают использование автоматизированных средств мониторинга и диагностики, которые дают возможность отслеживать соответствие установленным параметрам в режиме реального времени [23]. Такие методы позволяют системам быстро реагировать на отклонения от нормативных показателей и принимать корректирующие меры, повышая стабильность работы системы. Интеграция SLA в микросервисную архитектуру дополнительно улучшает способность системы к адаптации, так как каждый агент может быть перепрограммирован или обновлен без необходимости прерывать общую работу системы. Это особенно актуально для цифровых двойников, где SLA задает критерии, по которым агент продукта и агент ресурса координируют свои действия, гарантируя, что данные, необходимые для принятия решений, передаются своевременно и корректно.

Настоящее исследование направлено на разработку методов микросервисной реализации функций сбора, мониторинга, диагностики, защиты информации и принятия решений для многоагентных систем, отражающих поведение цифровых двойников. Основная цель работы заключается в создании микросервисных алгоритмов, которые обеспечат стабильное выполнение SLA между агентом продукта и агентом ресурса, а также поддержку непрерывного мониторинга и принятия решений. Автором предлагаются методы, которые позволяют осущест-

влять адаптацию агентов в режиме реального времени к изменениям состояния активов и гибко реагировать на потребности производственной системы, поддерживая высокое качество сервиса.

Микросервисная реализация многоагентной системы и структура SLA

Микросервисная архитектура в многоагентных системах предоставляет гибкие и масштабируемые инструменты для реализации функций сбора данных, которые становятся основой для принятия решений и мониторинга состояния активов. В многоагентной системе, реализующей функции цифровых двойников, сбор данных играет ключевую роль в обеспечении согласованности и непрерывности процессов на уровне взаимодействия агентов. Эти агенты, представляющие цифровых двойников, которые соответствуют реальным активам предприятия, выполняют автономные функции, координируя взаимодействие друг с другом через установленные соглашения об уровне сервиса (SLA). Для успешного взаимодействия агентов в многоагентной системе и обеспечения эффективного выполнения функций цифровых двойников важную роль играет структурированное управление соглашениями об уровне сервиса. SLA выступает не только механизмом для регулирования параметров обмена данными между агентами продукта и агентами ресурса, но и задает ключевые условия для сбора данных, мониторинга и принятия решений. Блоковая структура SLA помогает согласовать действия агентов, обеспечивая необходимый уровень надежности, адаптивности и устойчивости системы. Ниже представлена блоковая структура SLA (табл.1), которая регулирует взаимодействие агентов и способствует опти-

Блоковая структура SLA
Block structure of SLA

Блоки структуры SLA	Параметры блоков	Описание параметров
Блок параметров сервиса	• Цель и задачи • Объекты взаимодействия • Период действия	• Описание основной цели и ключевых задач взаимодействия, таких как сбор данных, мониторинг состояния активов и принятие решений. • Указание агентов, между которыми устанавливается соглашение (например, агент продукта и агент ресурса). • Определение сроков действия SLA, например, длительность, приоритеты выполнения, частота пересмотра.
Блок параметров сбора данных	• Частота сбора данных • Тип и объем данных • Фильтрация и обработка данных • Допустимые задержки	• Указание частоты, с которой агенты должны собирать данные, в зависимости от текущих условий системы и ее нагрузки. • Определение параметров данных, подлежащих сбору (например, производственные параметры, состояние ресурсов). • Указание условий, при которых данные фильтруются и преобразуются на уровне агента до передачи в систему. • Максимально допустимые временные задержки при сборе и передаче данных для предотвращения избыточной нагрузки.
Блок параметров мониторинга	• Мониторинговые метрики • Частота обновления метрик • Пороговые значения и допуски • Автоматическое уведомление	• Список метрик для мониторинга, таких как производительность, температура, состояние актива и другие критичные параметры. • Частота мониторинга для каждой метрики, обеспечивающая своевременное обновление данных в системе. • Определение пороговых значений и допустимых пределов для метрик, при достижении которых активируется диагностика. • Настройка условий для автоматического уведомления агентов при отклонениях от нормы и необходимости корректирующих действий.
Блок параметров диагностики	• Критерии запуска диагностики • Методы диагностики • Классификация отклонений • Прогнозирование последствий	• Указание условий, при которых запускается процесс диагностики, например, при достижении критических уровней метрик. • Способы анализа состояния системы, такие как временные ряды, машинное обучение, прогнозирование. • Разделение отклонений по уровню критичности (например, высокая, средняя, низкая) и их возможные причины. • Определение возможных последствий отклонений для других агентов и системы в целом.
Блок параметров принятия решения	• Сценарии адаптивных действий • Пороговые условия для принятия решений • Эскалация и приоритеты • Автономия агентов	• Определение стандартных сценариев действий, которые агент должен предпринять в зависимости от состояния системы и условий SLA. • Пороговые значения, при достижении которых агент должен принять определенное решение, включая инструкции для выполнения. • Процедуры и правила эскалации для критических ситуаций, требующих немедленных действий, и установление приоритетов действий. • Уровень автономности агентов при принятии решений, включая допустимые пределы для действий, принимаемых без дополнительных запросов.
Блок параметров отчетности и контроля	• Частота отчетности • Критерии выполнения SLA • Аудит и проверка соответствия • Изменение SLA	• Периодичность отчетов о результатах взаимодействия и текущем статусе выполнения SLA. • Определение критериев для измерения успешности выполнения SLA, включая ключевые показатели эффективности (KPI). • Возможность регулярного аудита для оценки выполнения SLA и корректировки параметров при необходимости. • Правила изменения условий SLA в зависимости от изменений условий в системе или после анализа результатов работы.
Блок параметров управления нарушениями	• Определение нарушений • Процедуры реагирования • Меры восстановления • Штрафные санкции или корректирующие меры	• Критерии и условия, которые рассматриваются как нарушение SLA. • Описание действий агентов и системы в случае нарушений, включая активацию резервных сценариев или временные решения. • Процедуры для восстановления стабильности системы после нарушения и предотвращения повторных нарушений. • Определение санкций или корректирующих мер для агентов при повторных нарушениях SLA.

мальной работе алгоритмов сбора данных, мониторинга и принятия решений в рамках микросервисной архитектуры. Эффективное функционирование многоагентных систем, основанных на цифровых двойниках предприятий,

зависит от надежных механизмов мониторинга и диагностики, которые обеспечивают своевременное выявление отклонений и поддержание работоспособности активов на основе SLA. В многоагентной системе каждый агент выпол-

няет уникальную функцию, а мониторинг состояния активов позволяет реагировать на изменения в их состоянии и предотвращать возможные сбои. Принятие решений в многоагентных системах, представленных цифровыми двой-

никами, требует применения специализированных микросервисных алгоритмов, способных оперативно реагировать на изменяющиеся условия и координировать действия между агентами продукта и ресурса. В таких системах SLA определяет допустимые условия для принятия и исполнения решений. Систематизация процессов принятия решений позволяет каждому агенту выполнять свои задачи автономно, но с учетом требований и ограничений, установленных SLA. Это создает надежную основу для скоординированного управления в условиях высокой неопределенности и динамичности среды, что является критически важным для современных сетевых предприятий.

Таким образом, для эффективного выполнения задач сбора данных, мониторинга, диагностики, защиты информации и принятия решения и их исполнения требуются специализированные микросервисные алгоритмы, поддерживающие гибкость и адаптируемость системы к изменяющимся условиям.

Метод и алгоритм микросервисной реализации для сбора данных

Один из ключевых моментов микросервисной реализации для сбора данных заключается в модульности подхода, который позволяет каждому агенту отвечать за определенный аспект процесса [24]. Реализация происходит путем выделения отдельных микросервисов для сбора данных о конкретных характеристиках активов, таких как состояние, производительность и возможные отклонения от нормы. Подход обеспечивает независимость и автономность каждого микросервиса, что особенно полезно в случаях, когда требования к агентам варьируются в зависимости от особенностей активов и условий их

эксплуатации. При этом каждый микросервис интегрируется в общую систему через SLA, что позволяет гарантировать точность и своевременность данных, необходимых для принятия решений.

Основным преимуществом использования микросервисов для сбора данных является их способность обеспечивать асинхронный и распределенный обмен данными, позволяя каждому агенту собирать и передавать данные в зависимости от текущих условий [25]. В микросервисной архитектуре для этого реализуются брокеры сообщений и очереди, позволяющие организовать надежный обмен между агентами без необходимости постоянного подключения к централизованному серверу. Такое решение не только сокращает задержки, но и снижает нагрузку на систему, позволяя каждому агенту функционировать в автономном режиме. Взаимодействие между агентом продукта и агентом ресурса через SLA дополнительно оптимизирует процесс сбора данных, так как каждый агент может контролировать частоту и объем передаваемых данных в зависимости от текущего состояния системы.

В условиях повышенной нагрузки или изменения требований к данным SLA могут автоматически регулировать частоту сбора и объем данных, что позволяет оптимизировать производительность системы. Этот метод особенно полезен в случаях, когда необходимо сократить избыточность данных и обеспечить только актуальную информацию для анализа и принятия решений. При этом гибкость микросервисной архитектуры позволяет интегрировать адаптивные SLA без необходимости изменения основных компонентов системы, что обеспечивает стабильность работы всех агентов.

Для реализации высококачественного сбора данных

используются фильтрация и предобработка, которые позволяют снизить объем передаваемой информации и минимизировать ошибочные данные. Фильтрация проверяет данные на соответствие критериям, установленным SLA, и передает только значимые данные, что позволяет снизить нагрузку на каналы связи и серверы обработки. На этапе предобработки данные могут нормализоваться и агрегироваться, что обеспечивает их готовность к последующим этапам анализа и принятия решений. Применение фильтрации и предобработки в микросервисах агентов позволяет повысить точность и оперативность принимаемых решений, что особенно важно в условиях работы с большими объемами данных и высоких требований к скорости обработки.

Использование распределенной обработки данных в микросервисах позволяет улучшить масштабируемость системы и упростить интеграцию новых агентов и компонентов. В этом подходе каждый агент может выполнять задачи по сбору данных параллельно с другими, что сокращает время отклика системы и повышает общую производительность. Распределенные микросервисы могут работать на различных серверах и обеспечивать непрерывный поток данных, независимо от состояния других компонентов системы. Взаимодействие через SLA позволяет координировать задачи по сбору данных, что обеспечивает устойчивую и высокопроизводительную работу системы даже в условиях повышенной нагрузки.

Метод распределенного сбора данных с использованием SLA и событийного взаимодействия основан на организации взаимодействия агентов цифровых двойников (агента продукта и агента ресурса) в сети предприятия через сервис агента цифровой платформы

[2]. Этот метод фокусируется на обеспечении гибкости, масштабируемости и высокой надежности сбора данных. В основе метода лежит использование SLA (соглашения о предоставлении сервиса), формируемого сервисом агента цифровой платформы, что позволяет задавать конкретные параметры и частоту сбора данных для каждого взаимодействия между агентами.

Цель метода заключается в том, чтобы агенты могли самостоятельно собирать, фильтровать и обрабатывать данные в зависимости от текущих условий и требуемого качества сервиса, определенного SLA. Сервис агента цифровой платформы управляет SLA и отслеживает выполнение параметров по сбору данных, передавая их в микросервисную архитектуру, что повышает согласованность действий агентов, распределенных по сети..

Алгоритм микросервисной реализации сбора данных представлен на рис. 1. Перед началом работы алгоритма сервис агента цифровой платформы должен инициализировать SLA для взаимодействия агента продукта и агента ресурса, определяя частоту сбора данных, объем, допустимые задержки, формат и критерии фильтрации данных. SLA хранится и управляется центральным компонентом платформенного сервиса, который периодически передает обновленные параметры сбора данных агентам. Агенты продукта и ресурса определяют ключевые события, которые требуют сбора данных, например, изменения состояния актива или параметры, влияющие на производство. Каждый агент подписывается на события, значимые для его роли, и получает требования по частоте и точности данных от SLA. Эти события могут быть как внешними (например, запрос пользователя), так и внутренними



Рис. 1. Алгоритм микросервисной реализации сбора данных
Fig. 1. Algorithm of microservice implementation of data collection

(например, выход показателя за установленные пороги).

Рассмотрим алгоритм микросервисной реализации сбора данных более детально. Когда зафиксировано значимое событие, инициируется процесс сбора данных, агент ресурса производит локальную фильтрацию данных, оставляя только значимые для SLA параметры, тем самым снижая нагрузку на сеть. Данные обрабатываются и структурируются агентом с учетом требований SLA, после чего отправляются агенту продукта или на центральную платформу для дальнейшей обработки.

Собранные и отфильтрованные данные передаются в микросервисы платформы, где каждый микросервис отвечает за свой аспект обработки: один отвечает за валидацию, другой — за агрегацию, третий — за предобработку для дальнейшего анализа. SLA регулирует взаимодействие микросервисов, задавая допустимые пределы задержек и частоты обмена данными. Это позволяет добиться гибкости и согласо-

ванности процессов при росте нагрузки. После обработки микросервисы проверяют данные на соответствие SLA. Данные, удовлетворяющие всем критериям, сохраняются в хранилища данных. При наличии отклонений агент уведомляется о необходимости корректировки параметров или частоты сбора. Сервис агента цифровой платформы мониторит исполнение SLA и обновляет параметры или частоту сбора данных в реальном времени. Например, при увеличении нагрузки частота сбора может быть снижена, или наоборот, повышена в критический момент. Обновленные параметры передаются агентам продукта и ресурса, после чего процесс повторяется в соответствии с новыми условиями.

Рассмотренный метод и алгоритм микросервисной реализации для сбора данных в многоагентных системах с использованием цифровых двойников обеспечивают адаптивность, автономность и масштабируемость. Этот метод будет поддерживать эффек-

тивное взаимодействие между агентами продукта и ресурса через SLA, что позволит достичь высокой согласованности и точности обмена данными.

Метод и алгоритм микросервисной реализации мониторинга и диагностики

Микросервисы, ответственные за мониторинг, предоставляют возможность автономного отслеживания показателей состояния каждого цифрового двойника, что позволяет выявлять изменения в режиме реального времени и передавать эти данные в соответствующие агентам.

Алгоритм мониторинга, реализуемый в микросервисах, может адаптироваться к изменяющимся условиям и быстро анализировать большие объемы данных, позволяя агентам продукта и ресурса получать обновленную информацию и корректировать действия. Этот подход особенно важен при мониторинге критичных показателей, где даже минимальные отклонения могут указывать на необходимость немедленного вмешательства. Использование SLA позволяет каждому агенту контролировать параметры, устанавливая допустимые пределы для различных условий эксплуатации. В случае выявления отклонений, микросервисы мгновенно передают данные соответствующим агентам, инициируя процесс диагностики и предотвращая возможные сбои в системе.

Диагностика состояния активов осуществляется с помощью независимых микросервисов, которые позволяют агентам автоматически проводить анализ и выявлять причины отклонений, тем самым минимизируя время простоя. Это позволяет системе не только реагировать на текущие проблемы, но и предвидеть потенциальные сбои [26]. Совместное использование мониторинга и диагностики через

SLA дает возможность агентам продукта и ресурса корректировать свои действия, что повышает эффективность взаимодействия и снижает риски нарушений.

Одним из важных аспектов микросервисного мониторинга является возможность интеграции распределенных систем наблюдения, которые позволяют агентам вести диагностику в режиме реального времени, независимо от их местоположения в системе [27]. При этом микросервисы могут быть распределены на нескольких серверах и взаимодействовать через облачные платформы, что повышает надежность системы. В рамках распределенного мониторинга каждый агент может проводить локальные проверки и анализировать состояние актива, обеспечивая мгновенный доступ к данным о состоянии системы.

Микросервис мониторинга может автоматически инициировать процессы диагностики, если обнаружены отклонения от установленных параметров, что минимизирует риски сбоев и гарантирует быстрое восстановление системы. Это позволяет агентам продукта и ресурса поддерживать оптимальные параметры функционирования и обеспечивает устойчивую работу в условиях различных факторов нагрузки.

Одновременно с этим, микросервисы для мониторинга и диагностики могут быть дополнены инструментами визуализации данных, которые помогают операторам и аналитикам отслеживать состояние активов в реальном времени [28]. Визуализация позволяет не только получить мгновенное представление о состоянии системы, но и анализировать тренды для долгосрочного планирования и улучшения процесса принятия решений. Использование наглядных инструментов особенно важно для крупных предприятий, где цифровые двойники множе-

ства активов требуют постоянного контроля и оптимизации. Визуализация данных помогает лучше понять состояние системы и позволяет агентам продукта и ресурса совместно работать на достижение оптимальных условий.

Для обеспечения защиты данных и предотвращения несанкционированного доступа в алгоритмы мониторинга и диагностики внедрены функции проверки информационной безопасности. Передача данных между агентами и микросервисами выполняется по защищенным каналам с использованием методов шифрования и аутентификации, что позволяет минимизировать риски утечек данных. В процессе анализа метрик микросервис мониторинга проверяет наличие подозрительных активностей, таких как резкие изменения объемов данных или попытки доступа к защищенным метрикам, которые могут свидетельствовать о потенциальных угрозах безопасности. При обнаружении таких аномалий инициируется сигнал об инциденте, который передается на диагностику для дальнейшего анализа. На этапе диагностики система также проверяет данные на наличие несанкционированных изменений или необычных моделей поведения. Эти меры безопасности помогают повысить надежность системы, обеспечивая непрерывную защиту данных и стабильное выполнение соглашений об уровне сервиса.

Метод адаптивного мониторинга и диагностики с использованием микросервисной архитектуры направлен на автоматизацию и оперативность наблюдения за состоянием агентов в многоагентной системе. В основе метода лежит использование соглашений об уровне сервиса (SLA), формируемых сервисом агента цифровой платформы, что позволяет задавать гибкие па-

параметры мониторинга и диагностики для каждого агента в зависимости от задач и условий.

Цель метода — обеспечить своевременную диагностику потенциальных проблем между агентом продукта и агентом ресурса, используя SLA для настройки периодичности и детальности мониторинга. В рамках системы каждый агент (цифровой двойник) непрерывно передает ключевые метрики своему микросервису мониторинга, который анализирует и передает данные на диагностику в случае выявления отклонений. Метод реализован в виде алгоритма мониторинга и диагностики и представлен на рис.2.

Алгоритм микросервисной реализации для мониторинга и диагностики включает в себя следующие действия. Каждый агент собирает установленные SLA метрики в реальном времени и отправляет их в микросервис мониторинга, отвечающий за сбор данных для каждого конкретного агента. Параметры мониторинга, такие как частота отправки и объем данных, задаются согласно SLA, что снижает избыточность передаваемых данных. Данные передаются по защищенному каналу с использованием методов шифрования и аутентификации для предотвращения несанкционированного доступа. Микросервис мониторинга сверяет метрики с установленными пороговыми значениями соглашения на наличие отклонений и записывает их в базу мониторинга. После сверки микросервис проводит контроль аномалий безопасности т.е. проверяет данные на наличие необычной активности, которая может указывать на потенциальные угрозы (например, резкие изменения в объемах данных, попытки доступа к закрытым метрикам и др.). Если обнаружены подозрительные активности - аномалии, иницииру-

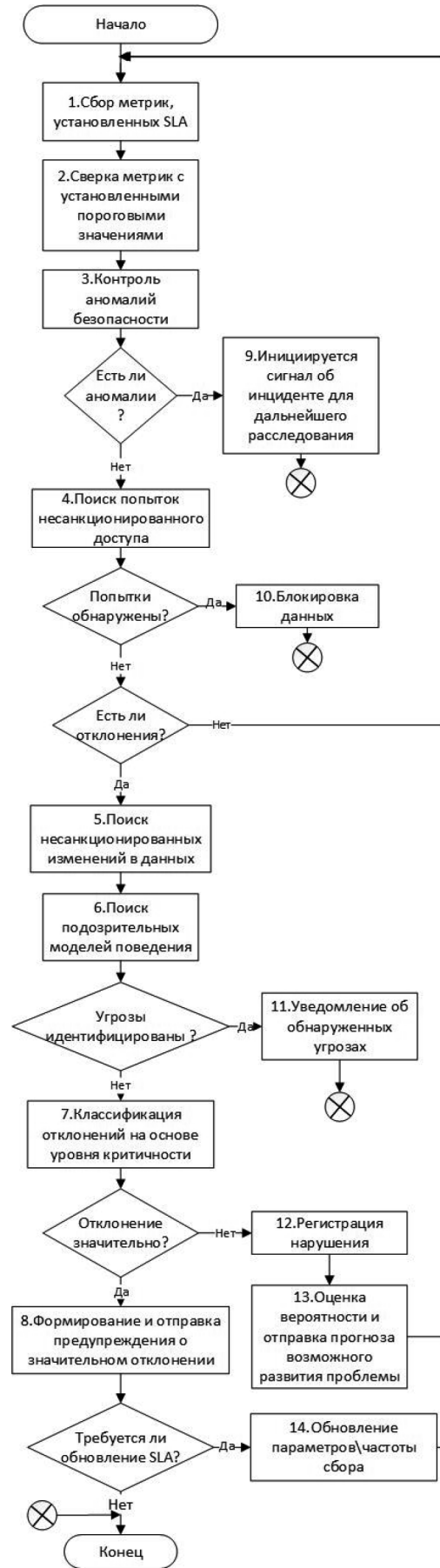


Рис. 2. Алгоритм микросервисной реализации для мониторинга и диагностики
Fig. 2. Microservice implementation algorithm for monitoring and diagnostics

ется сигнал об инциденте для дальнейшего расследования. Перед передачей данных на этап диагностики микросервис мониторинга проводит поиск несанкционированного доступа и подтверждает, что передаваемые данные предназначены только для авторизованных микросервисов и агентов. В случае обнаружения попыток несанкционированного доступа данные блокируются, а агент или микросервис уведомляется о нарушении. При наличии отклонений от заданных параметров SLA, микросервис инициирует обращение к сервису диагностики.

Микросервис диагностики получает от микросервиса мониторинга данные о метриках и проверяет их по диагностическим правилам и шаблонам, заданным в SLA. В рамках диагностики проводится поиск потенциальных угроз безопасности, таких как несанкционированные изменения в данных или подозрительные модели поведения. Если идентифицированы угрозы, микросервис диагностики отправляет уведомление об обнаруженных угрозах в платформенный сервис. В процессе диагностики производится классификация отклонений на основе уровня критичности по типу (например, высокое потребление ресурсов или снижение производительности) и их влияние на дальнейшую работу агентов. Незначительные нарушения регистрируются и диагностический сервис оценивает вероятность дальнейшего развития отклонений, включая угрозы безопасности, и отправляет прогноз агенту продукта или агенту ресурса в зависимости от уровня критичности. Если прогноз указывает на серьезную проблему, предупреждение о значительном отклонении отправляется также платформенному сервису для обновления SLA и принятия немедленных мер по усилению защиты. Платформенный сервис на основе уведомлений

от диагностического сервиса может обновить параметры SLA для оптимизации мониторинга и предотвращения аналогичных отклонений, а также анализирует полученные отчеты о нарушениях безопасности и корректирует SLA для повышения безопасности данных и предотвращения повторных угроз. Агенту может быть передан обновленный SLA, чтобы адаптировать параметры мониторинга и диагностики под изменившиеся условия или уровень загрузки.

Метод и алгоритм реализации микросервисов мониторинга и диагностики в многоагентных системах создают условия для своевременного сбора информации о состоянии активов, анализа данных и принятия решений. Благодаря применению SLA система получает возможность следить за состоянием и возможными сбоями. Такой подход повышает устойчивость и гибкость системы, делая возможным быстрое и эффективное взаимодействие агентов через SLA, что способствует оптимизации процессов и снижению затрат на обслуживание.

Методы и алгоритмы принятия решений и их исполнения

Первым шагом в реализации механизма принятия решений является формализация данных, полученных на этапе мониторинга и диагностики, для последующего анализа. Микросервисы, отвечающие за сбор и обработку данных, передают их в специализированные компоненты, которые выполняют оценку состояния системы на основе установленных правил и SLA. Эти правила могут включать как фиксированные пороговые значения, так и адаптивные критерии, изменяющиеся в зависимости от текущих условий.

Основной задачей алгоритма исполнения решений яв-

ляется обеспечение согласованности между агентами, что достигается благодаря SLA, которые определяют, какие действия должен предпринять каждый агент в случае возникновения определенных условий. Эти соглашения формируют своеобразные «сценарии действий» для каждого агента, позволяя им предугадывать, как изменятся их задачи в зависимости от состояния всей системы.

Еще одним значимым аспектом является использование алгоритма для контроля результатов принятых решений, что позволяет агентам корректировать свои действия на основе обратной связи [29]. Микросервис с функцией мониторинга анализирует, насколько успешно выполнены действия, и передает данные в компоненты принятия решений, которые могут адаптировать алгоритмы для будущих ситуаций. SLA в этом процессе играют важную роль, поскольку они помогают определить, какие критерии успешности должны соблюдаться, и обеспечивают возможность корректировки параметров взаимодействия между агентами продукта и ресурса. Такой подход обеспечивает циклическое совершенствование системы, повышая ее адаптивность и устойчивость.

Для повышения скорости исполнения решений и минимизации задержек используется метод асинхронного обмена данными, позволяющий агентам действовать параллельно [25]. Микросервисная архитектура поддерживает такой подход, создавая независимые процессы, которые могут выполнять задачи одновременно, снижая время реакции системы на изменение условий. SLA регулируют объем и частоту обмена данными, поддерживая баланс между скоростью исполнения и точностью принимаемых решений. Это особенно важно для многоа-

гентных систем, работающих в реальном времени, где задержки могут привести к накоплению ошибок и нарушению согласованности между агентами [10].

Главная задача метода — обеспечить, чтобы каждый агент (в особенности агенты продукта и ресурса) мог самостоятельно принимать решения, исходя из SLA и текущего состояния системы, а также корректировать действия для выполнения заданных условий и требований. Метод реализован в виде алгоритма, блок-схема представлена на рис.3.

Описать данный алгоритм принятия решений и их исполнения можно следующей последовательностью действий. SLA инициализирует условия, при которых агент должен принимать решения: пороговые значения метрик, критические условия и параметры, указывающие на необходимость адаптации. Платформенный сервис обновляет SLA, если необходимо, а агенты получают текущее состояние условий и триггеров. Агент продукта и агент ресурса непрерывно передают обновленные данные и текущие метрики микросервису мониторинга, который проверяет достижение пороговых условий, заданных в SLA. При достижении пороговых условий SLA микросервис мониторинга инициирует процесс принятия решений.

Микросервис принятия решений анализирует текущие данные агентов в контексте установленных SLA и выбирает подходящий сценарий действия. Доступные сценарии зависят от принятого решения, текущего состояния ресурсов и производственных процессов; микросервис может определить действие (например, изменение распределения ресурсов, задержку или ускорение производственных этапов) для оптимизации вы-



Рис. 3. Алгоритм микросервисной реализации для принятия решений и их исполнения

Fig. 3. Microservice implementation algorithm for decision-making and execution

полнения SLA. В случае возникновения критических условий микросервис может рекомендовать корректировку задач агента или активацию альтернативного сценария, предусмотренного SLA. Микросервис принятия решений отправляет инструкции соответствующему агенту для исполнения (например, агенту продукта для корректировки плана производства или агенту ресурса для перераспределения нагрузки). Агент получает инструкции и исполняет команды из нее, изменяет свои действия для соответствия обновленным условиям. Если агент сталкивается с повторяющейся проблемой или требуемым сценарием адаптации, сервис агента цифровой может пересмотреть и обновить параметры SLA, чтобы улучшить адаптацию и автономность системы. Обновленный SLA затем отправляется агентам для дальнейшего выполнения с учетом новых условий.

Микросервисный метод и алгоритм для принятия решений и их исполнения в многоагентных системах обеспечивают высокую гибкость, надежность и согласованность взаимодействия между цифровыми двойниками предприятия. Взаимодействие агентов через SLA позволяет им адаптироваться к изменяющимся условиям и действовать с учетом потребностей друг друга, что улучшает эффективность всей системы. Применение современных методов аналитики и отказоустойчивости в микросервисах способствует созданию адаптивной системы, которая поддерживает оптимальную работу активов в условиях повышенной динамики и сложности.

Заключение

Предложенные методы и алгоритмы микросервисной реализации для сбора данных, мониторинга, диагностики, защиты информации и принятия решений в многоагентных системах направлены на обеспечение гибкости, автономности и масштабируемости МАС для эффективного взаимодействия агентов, реализующих функции цифровых двойников. В многоагентной системе, где основное взаимодействие происходит между агентом продукта и агентом ресурса, применение соглашений об уровне сервиса (SLA) позволяет обеспечить согласованность действий агентов и адаптацию к изменениям, что критически важно для современных сетевых предприятий.

Микросервисная архитектура доказала свою эффективность в создании автономных компонентов, что позволяет каждому агенту самостоятельно выполнять свои функции по сбору данных, мониторингу, диагностике, защите информации и принятию решений в рамках SLA. Поддержка такого уровня независимости и взаимодействия между агентами обеспечивает устойчивую работу системы даже в условиях высокой нагрузки и повышенной изменчивости внешних условий. Это позволяет цифровым двойникам реагировать на отклонения от нормативных показателей и автоматически корректировать свои действия для выполнения бизнес-целей.

Применение методов мониторинга и диагностики через микросервисы обеспечивает своевременное выявление и классификацию отклонений, что способствует предот-

вращению сбоев и повышает надежность всей системы. Благодаря распределенной обработке данных и адаптивному мониторингу, система может оперативно реагировать на изменения состояния активов, что особенно важно для поддержки стабильности и эффективности работы цифровых двойников. SLA в этой архитектуре играют ключевую роль, определяя параметры для каждого процесса, и способствует оптимизации работы МАС.

Методы принятия решений, реализуемые микросервисами, позволяют агентам гибко адаптировать свои действия, основываясь на SLA и анализе текущих данных. Использование аналитических алгоритмов, прогнозирования и автономного исполнения решений улучшает качество управления ресурсами и снижает риск возможных ошибок. Этот подход не только поддерживает эффективность работы агентов в реальном времени, но и способствует циклическому улучшению системы на основе полученной обратной связи.

Таким образом, предложенные методы и алгоритмы микросервисной реализации делают возможным создание адаптивной и надежной многоагентной системы, способной к быстрому масштабированию и гибкой адаптации к изменениям. Современные сетевые предприятия, внедряя микросервисную архитектуру для поддержки цифровых двойников, получают преимущества в управлении активами и процессами, что позволяет повысить конкурентоспособность и эффективность бизнеса в условиях цифровой трансформации.

Литература

1. Звягин Л.С. Изучение многоагентных систем и решение проблем их математической алгоритмизации // Мягкие измерения и вычисления. 2021. Т. 46. № 9. С. 62–73. DOI: 10.36871/2618-9976.2021.09.003.

2. Тельнов Ю.Ф., Казаков В.А., Данилов А.В., Брызгалов А.А. Разработка моделей производственных и бизнес-процессов сетевых предприятий на основе многоагентных систем // Программные продукты и системы. 2023. № 4. С. 632–643. DOI: 10.15827/0236-235X.144.632-643.

3. Коновалов Н.С., Побойкина А.О., Чернов А.В. Построение микросервисной архитектуры // Современные инструментальные системы, информационные технологии и инновации: Сборник научных трудов XVI Международной научно-практической конференции (Курск, 18–19 марта 2021 г.) Отв. ред. М.С. Разумов. Курск: Юго-Западный государственный университет, 2021. С. 139–142.
4. Тельнов Ю.Ф., Казаков В.А., Данилов А.В. Проектирование многоагентной системы для сетевого предприятия // Бизнес-информатика. 2024. Т. 18. № 3. С. 70–86.
5. Jagutis M., Russell S., Collier R. W. Using Multi-Agent MicroServices (MAMS) for Agent-Based Modelling // International Workshop on Engineering Multi-Agent Systems. Cham: Springer Nature Switzerland, 2023. С. 85–92.
6. Melnyk A.O., Zimchenko B. Microservices Infrastructure Architecture for the Cloud-Based Multi-Agent Group Decision Support Systems for Autonomous Cyberphysical Systems // IntSol. 2023. С. 337–345.
7. Луценко Д.Ю., Полякова Л.П. Разбиение монолитного приложения на микросервисы с использованием паттерна Strangler // Информационные технологии в управлении и экономике. 2021. № 3(24). С. 82–87.
8. Комилиен Ф.С., Рахимов М.Ф. Микросервисная архитектура: от монолита к гибким распределенным системам // Доклады Национальной академии наук Таджикистана. 2023. Т. 66. № 11–12. С. 659–667.
9. Ирбитский И.С., Романенков А.М., Стульников К.Т., Удалов Н.Н. Подходы к формированию безопасности в микросервисной архитектуре // Современная наука: актуальные проблемы теории и практики. Серия: Естественные и технические науки. 2022. № 3. С. 91–99. DOI: 10.37882/2223-2966.2022.03.
10. Liu Z. et al. Reliability modelling and optimization for microservice-based cloud application using multi-agent system // IET Communications. 2022. Т. 16. № 10. С. 1182–1199.
11. Bondarenko A.S., Korolev D.V., Zaytsev K.S. Study the efficiency of using multi-agent models in modern microservice architectures // International Journal of Open Information Technologies. 2024. Т. 12. № 8. С. 48–55.
12. Кораблев А.В. Ключевые функциональность и преимущества использования цифровых двойников в промышленности // Цифровая экономика. 2019. № 2(6). С. 5–11.
13. Денисов С.Г. Технологии сбора и обработки данных для создания цифровых двойников // Бюллетень инновационных технологий. 2023. Т. 7. № 2(26). С. 12–17.
14. Li L. et al. Agent-based multi-tier SLA negotiation for intercloud // Journal of Cloud Computing. 2022. Т. 11. № 1. С. 16.
15. Pozdniakova O., Cholomskis A., Mažeika D. Self-adaptive autoscaling algorithm for SLA-sensitive applications running on the Kubernetes clusters // Cluster Computing. 2024. Т. 27. № 3. С. 2399–2426.
16. Nouredine S., Meriem B. ML-SLA-IoT: An SLA Specification and Monitoring Framework for IoT applications // Proceedings — 2021 International Conference on Information Systems and Advanced Technologies. 2021. DOI: 10.1109/ICISAT54145.2021.9678460.
17. Lazaroiu G. et al. Artificial intelligence-based decision-making algorithms, Internet of Things sensing networks, and sustainable cyber-physical management systems in big data-driven cognitive manufacturing // Oeconomia Copernicana. 2022. Т. 13. № 4. С. 1047–1080.
18. Брызгалов А.А. Разработка метода адаптации производственных и бизнес-процессов к динамическим условиям функционирования на основе накопления и анализа массивов данных // Инжиниринг предприятий и управление знаниями (ИП&УЗ-2023): Сборник научных трудов XXVI Российской научной конференции: в 2-х томах (Москва, 29–30 ноября 2023 г.). М.: Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г. В. Плеханова», 2023. С. 65–72.
19. Kumar R., Hassan M.F., Adnan M.H. M. A Principled Design of Intelligent Agent for the SLA negotiation process in cloud computing // 2022 2nd International Conference on Computing and Information Technology (ICCIT). IEEE, 2022. С. 383–387.
20. Swain A.K., Garza V.R. Key Factors in Achieving Service Level Agreements (SLA) for Information Technology (IT) Incident Resolution. 2022. DOI: 10.1007/s10796-022-10266-5.
21. Singh J., Goraya M.S. An Autonomous Multi-Agent Framework using Quality of Service to prevent Service Level Agreement Violations in Cloud Environment // International Journal of Advanced Computer Science and Applications. 2023. Т. 14. № 3.
22. Hioal O. et al. A method based on multi-agent systems and passive replication technique for predicting failures in cloud computing // Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science). 2023. Т. 16. № 1. С. 18–32.
23. Сидоров Ю.Ю. Использование технологии мультиагентных систем для решения задачи диагностики состояния технического объекта // Современные информационные технологии: сборник трудов по материалам 5-й Всероссийской научно-технической конференции (Москва, 27 сентября 2019 г.). М.: ООО «Научный консультант», 2019. С. 101–106.
24. Архипов К.А., Скляр А.Я. Способы обмена данными в микросервисной архитектуре // Развитие науки и практики в глобально меняющемся мире в условиях рисков: Сборник

материалов XXII Международной научно-практической конференции (Москва, 25 октября 2023 г.). М.: ООО «Издательство АЛЕФ», 2023. С. 141–145.

25. Sakurada L., Leitao P., De la Prieta F. Agent-based asset administration shell approach for digitizing industrial assets // *Ifac-Papersonline*. 2022. Т. 55. № 2. С. 193–198.

26. Hioual O. et al. A method based on multi-agent systems and passive replication technique for predicting failures in cloud computing // *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*. 2023. Т. 16. № 1. С. 18–32.

27. Usman M. et al. A survey on observability of distributed edge & container-based microservices // *IEEE Access*. 2022. Т. 10. С. 86904–86919.

References

1. Zvyagin L.S. Study of multi-agent systems and solution of problems of their mathematical algorithmization. *Myagkiye izmereniya i vychisleniya = Soft measurements and calculations*. 2021; 46; 9: 62–73. DOI: 10.36871/2618-9976.2021.09.003. (In Russ.)

2. Tel'nov Yu.F., Kazakov V.A., Danilov A.V., Bryzgalov A.A. Development of models of production and business processes of network enterprises based on multi-agent systems. *Programmnyye produkty i sistemy = Software products and systems*. 2023; 4: 632–643. DOI: 10.15827/0236-235X.144.632–643. (In Russ.)

3. Konovalov N.S., Poboykina A.O., Chernov A.V. Building a microservice architecture. *Sovremennyye instrumental'nyye sistemy, informatsionnyye tekhnologii i innovatsii: Sbornik nauchnykh trudov XVI Mezhdunarodnoy nauchno-prakticheskoy konferentsii = Modern instrumental systems, information technologies and innovations: Collection of scientific papers of the XVI International scientific and practical conference (Kursk, March 18–19, 2021)* Ed. M. S. Razumov. Kursk: South-West State University, 2021: 139–142. (In Russ.)

4. Tel'nov Yu.F., Kazakov V.A., Danilov A.V. Designing a multi-agent system for a network enterprise. *Biznes-informatika = Business informatics*. 2024; 18; 3: 70–86. (In Russ.)

5. Jagutis M., Russell S., Collier R. W. Using Multi-Agent MicroServices (MAMS) for Agent-Based Modelling. *International Workshop on Engineering Multi-Agent Systems*. Cham: Springer Nature Switzerland; 2023: 85–92.

6. Melnyk A. O., Zimchenko B. Microservices Infrastructure Architecture for the Cloud-Based Multi-Agent Group Decision Support Systems for Autonomous Cyberphysical Systems. *IntSol*. 2023: 337–345.

7. Lutsenko D.Yu., Polyakova L.P. Splitting a Monolithic Application into Microservices Using the Strangler Pattern. *Informatsionnyye tekhnologii v*

28. Miranda R. P. M. Real-time information proessing [i. é processing]. 2024.

29. Кучина Т. Н., Якушева Е. А. Алгоритм для проведения сбора обратной связи от пользователей в задачах контроля сложных информационных систем // *Научные исследования и современное образование: сборник материалов X Международной научно-практической конференции ФГБОУ ВО «Чувашский государственный университет им. И.Н. Ульянова»; Актюбинский региональный государственный университет им. К. Жубанова; Кыргызский экономический университет им. М. Рыскулбекова; ЦНС «Интерактив плюс»*. (Чебоксары, 13 марта 2020 г.) Чебоксары: ООО «Центр научного сотрудничества «Интерактив плюс», 2020. С. 82–87.

upravlennii i ekonomike = Information Technologies in Management and Economics. 2021; 3(24): 82–87. (In Russ.)

8. Komiliyen F.S., Rakhimov M.F. Microservice Architecture: From Monolith to Flexible Distributed Systems. *Doklady Natsional'noy akademii nauk Tadzhikistana = Reports of the National Academy of Sciences of Tajikistan*. 2023; 66; 11–12: 659–667.

9. Irbitskiy I.S., Romanenkov A.M., Stul'nikov K.T., Udalov N.N. Approaches to the formation of security in microservice architecture. *Sovremennaya nauka: aktual'nyye problemy teorii i praktiki. Seriya: Yestestvennyye i tekhnicheskiye nauki = Modern science: current problems of theory and practice. Series: Natural and technical sciences*. 2022; 3: 91–99. DOI: 10.37882/2223-2966.2022.03. (In Russ.)

10. Liu Z. et al. Reliability modelling and optimization for microservice-based cloud application using multi-agent system. *IET Communications*. 2022; 16; 10: 1182–1199.

11. Bondarenko A.S., Korolev D.V., Zaytsev K.S. Study the efficiency of using multi-agent models in modern microservice architectures. *International Journal of Open Information Technologies*. 2024; 12; 8: 48–55.

12. Korablev A.V. Key functionality and advantages of using digital twins in industry. *Tsifrovaya ekonomika = Digital Economy*. 2019; 2(6): 5–11.

13. Denisov S.G. Data collection and processing technologies for creating digital twins. *Byulleten' innovatsionnykh tekhnologiy = Bulletin of innovative technologies*. 2023; 7; 2(26): 12–17. (In Russ.)

14. Li L. et al. Agent-based multi-tier SLA negotiation for intercloud. *Journal of Cloud Computing*. 2022; 11; 1: 16.

15. Pozdniakova O., Cholomskis A., Mažeika D. Self-adaptive autoscaling algorithm for SLA-sensitive applications running on the Kubernetes clusters. *Cluster Computing*. 2024; 27; 3: 2399–2426.

16. Nouredine S., Meriem B. ML-SLA-IoT: An SLA Specification and Monitoring Framework

for IoT applications. Proceedings - 2021 International Conference on Information Systems and Advanced Technologies. 2021. DOI: 10.1109/ICISAT54145.2021.9678460.

17. Lazaroiu G. et al. Artificial intelligence-based decision-making algorithms, Internet of Things sensing networks, and sustainable cyber-physical management systems in big data-driven cognitive manufacturing. *Oeconomia Copernicana*. 2022; 13; 4: 1047-1080.

18. Bryzgalov A.A. Development of a method for adapting production and business processes to dynamic operating conditions based on the accumulation and analysis of data arrays. *Inzhiniring predpriyatiy i upravleniye znaniyami (IP&UZ-2023): Sbornik nauchnykh trudov XXVI Rossiyskoy nauchnoy konferentsii: v 2-kh tomakh = Enterprise Engineering and Knowledge Management (IP&UZ-2023): Collection of scientific papers of the XXVI Russian Scientific Conference: in 2 volumes (Moscow, November 29-30, 2023)*. Moscow: Plekhanov Russian University of Economics; 2023: 65-72. (In Russ.)

19. Kumar R., Hassan M.F., Adnan M.H.M. A Principled Design of Intelligent Agent for the SLA negotiation process in cloud computing. 2022 2nd International Conference on Computing and Information Technology (ICCIIT). IEEE; 2022: 383-387.

20. Swain A.K., Garza V.R. Key Factors in Achieving Service Level Agreements (SLA) for Information Technology (IT) Incident Resolution. 2022. DOI: 10.1007/s10796-022-10266-5.

21. Singh J., Goraya M.S. An Autonomous Multi-Agent Framework using Quality of Service to prevent Service Level Agreement Violations in Cloud Environment. *International Journal of Advanced Computer Science and Applications*. 2023; 14: 3.

22. Hioual O. et al. A method based on multi-agent systems and passive replication technique for predicting failures in cloud computing. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*. 2023; 16; 1: 18-32.

23. Sidorov Yu.Yu. Using multi-agent systems technology to solve the problem of diagnosing the state of a technical object. *Sovremennyye informatsionnyye tekhnologii: sbornik trudov po materialam 5-y Vserossiyskoy nauchno-tekhnicheskoy konferentsii = Modern information technologies: collection of papers based on the materials of the 5th All-Russian scientific and*

technical conference (Moscow, September 27, 2019). Moscow: Scientific Consultant LLC; 2019: 101-106. (In Russ.)

24. Arkhipov K.A., Sklyar A.Ya. Data exchange methods in microservice architecture. *Razvitiye nauki i praktiki v global'no menyayushchemsya mire v usloviyakh riskov: Sbornik materialov XXII Mezhdunarodnoy nauchno-prakticheskoy konferentsii = Development of science and practice in a globally changing world under risks: Collection of materials of the XXII International scientific and practical conference (Moscow, October 25, 2023)*. Moscow: Alef Publishing House LLC; 2023: 141-145. (In Russ.)

25. Sakurada L., Leitao P., De la Prieta F. Agent-based asset administration shell approach for digitizing industrial assets. *Ifac-Papersonline*. 2022; 55; 2: 193-198.

26. Hioual O. et al. A method based on multi-agent systems and passive replication technique for predicting failures in cloud computing. *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science)*. 2023; 16; 1: 18-32.

27. Usman M. et al. A survey on observability of distributed edge & container-based microservices. *IEEE Access*. 2022; 10: 86904-86919.

28. Miranda R. P. M. Real-time information processing [i. é processing]. 2024.

29. Kuchina T.N., Yakusheva Ye.A. Algorithm for collecting feedback from users in problems of monitoring complex information systems. *Nauchnyye issledovaniya i sovremennoye obrazovaniye: sbornik materialov X Mezhdunarodnoy nauchno-prakticheskoy konferentsii FGBOU VO «Chuvashskiy gosudarstvennyy universitet im. I.N. Ul'yanova»; Aktyubinskiy regional'nyy gosudarstvennyy universitet im. K. Zhubanova; Kyrgyzskiy ekonomicheskiy universitet im. M. Ryskulbekova; TSNS «Interaktiv plus» = Scientific research and modern education: collection of materials of the X International scientific and practical conference FSBEI HE «Chuvash State University named after I.N. Ulyanov»; Aktobe Regional State University named after K. Zhubanov; Kyrgyz Economic University named after M. Ryskulbekov; CNS «Interactive Plus». (Cheboksary, March 13, 2020)*. Cheboksary: LLC «Center for Scientific Cooperation» Interactive Plus «; 2020: 82-87. (In Russ.)

Сведения об авторе

Алексей Алексеевич Брызгалов

Ассистент кафедры прикладной информатики и информационной безопасности

Российский экономический университет

им. Г.В. Плеханова, Москва, Россия

Эл. почта: Bryzgalov.AA@rea.ru

Information about the author

Alexey A. Bryzgalov

Assistant of the Department of Applied Informatics and Information Security

Plekhanov Russian University of Economics,

Moscow, Russia

E-mail: Bryzgalov.AA@rea.ru