УДК 004.622

Разработка информационной модели сбора и анализа сведений программно-аппаратного состава узлов локальной вычислительной сети для обеспечения мероприятий по информационной безопасности

Королёнок Александр Юрьевич
Приамурский государственный университет им. Шолом-Алейхема
Студент

Аннотация
Меры обеспечения информационной безопасности являются основополагающими, для надежной, бесперебойной работы информационных систем и информационных процессов любого предприятия. Обеспечение информационной безопасности представляет собой комплекс организационных и технических мероприятий, которые должны неукоснительно и в полном объеме исполняться в организации в соответствии с разработанной политикой и нормативными документами, регламентирующими это направление деятельности. Для рационального использования вычислительных и трудовых ресурсов большое значение имеет анализ (или разработка) модели, которая смогла бы оптимальным образом обеспечить комплексную реализацию данных мероприятий. В данной статье рассматривается архитектура построения мультиагентной информационной модели по обеспечению сбора и анализа данных о программно-аппаратном составе узлов локальной вычислительной сети для обеспечения мероприятий по информационной безопасности.
Ключевые слова: Мультиагентная архитектура, клиент-серверное взаимодействие, информационная безопасность, Windows Management Instrumentation, Windows Communication Foundation.

Development of an information model for collecting and analyzing information on the hardware and software composition of local area network nodes to ensure information security measures

Korolenok Alexandr Yurievich
Sholom-Aleichem Priamursky State University
Student

Abstract
Measures to ensure information security are fundamental for the reliable, uninterrupted operation of information systems and information processes of any enterprise. Ensuring information security is a set of organizational and technical measures that must be strictly and fully implemented in the organization in accordance with the developed policy and regulatory documents governing this area of activity. For the rational use of computing and labor resources, the analysis

(or development) of a model that could optimally ensure the integrated implementation of these activities is of great importance. This article discusses the architecture of building a multi-agent information model to ensure the collection and analysis of data on the hardware and software composition of local area network nodes to ensure information security measures.

Keywords: Multi-agent architecture, client-server interaction, information security, Windows Management Instrumentation, Windows Communication Foundation.

Научный руководитель:
Глаголев Владимир Александрович
Приамурский государственный университет имени Шолом-Алейхема
к.г.н., доцент, доцент кафедры информационных систем, математики и правовой информатики

1. Введение

1.1. Актуальность исследования

Отсутствие автоматизированного способа сбора сведений об аппаратно-программном составе ПЭВМ, анализа этих сведений, соотнесения их к требованиям информационной безопасности влечет за собой неоптимальное использование сил и средств и проявляется на всех этапах рассматриваемого процесса: начиная со сбора информации, заканчивая устранением недостатков (несоответствий предъявляемым требованиям).

Объектом исследования является реализация мультиагентной системы оптимизации информационных процессов в рамках осуществления мероприятий по обеспечению информационной безопасности и сопутствующих информационных процессов.

Предмет исследования – информационная модель обеспечения мероприятий по требованиям информационной безопасности.

Гипотеза исследования заключается в том, что реализация информационной модели, сможет обеспечить более эффективное выполнение комплекса мер, что позволит более прогнозируемо и эффективно использовать трудовые ресурсы и повысить качество информационных процессов, путем внедрения автоматизации сбора и анализа информации по программно-аппаратному составу парка ПЭВМ организации.

1.2. Обзор исследований

Направлением разработки информационных систем занимаются многие, современные ученые. М.Г. Адеева и К.М. Айгумов [1] описали возможность привязки контекста данных в wpf с использованием свойств и ресурсов данного. Исследователи Л.Ю. Бакиев и А.Н. Борисов [2] описали разработку клиент-серверного приложения ежедневника. И.И, Баранкова, У.В, Михайлова и Г.И. Лукьянов [3] продемонстрировали возможность разработки приложений на языке программирования C# для работы с базами данных. Исследователи Н.В. Бужинская и Н.О. Котелевец [4] описали разработку программного продукта для автоматизации учета оборудования

на предприятии. А.А Иванов, Д.Д. Колташёва и Шимов [5] описали разработку приложения на языке C# с использованием Microsoft SQL Server. В авторской статье А.Ю. Королёнок [6] продемонстрировал использование инструментария управления Windows (WMI) для получения системной информации с применением техник объектно-ориентированного программирования. Е.И. Костин и И.В. Чумак [7] продемонстрировал создание службы wcf для использования в клиент-серверном приложении. Исследователи М.Д. Кузина и А.А. Семенов [8] продемонстрировали разработку системы тестирования на основе платформы ASP. NET. Ученный А.Х. Назмутдинов [9] в своей работе продемонстрировал особенности применения паттерна mvvm в клиентских приложениях с использованием платформы пользовательского интерфейса wpf. Ученные Belenesi D. T., Gabor G., Moisi E. V. [10] произвели сравнительное исследование фреймворков WPF и UWP. Исследователь Daadoo M. [11] произвел разработку и внедрение чат-приложения с использованием WPF и WCF. Ученный Sannarangaiah K. [1] описал проектирование и разработку графического пользовательского интерфейса на языке программирования C#.

1.3. Цель исследования

Целью исследования является создание архитектуры информационной модели сбора и анализа сведений программно-аппаратного состава узлов локальной вычислительной сети.

2.1. Методы исследования

Инструментом для достижения цели станет интегрированная среда разработки Visual Studio 2022 Community, и язык C# 8.0. Клиент-серверное взаимодействие реализовано с помощью технологии Windows Communication Foundation (WCF), а интерфейс приложений с использованием Windows Presentation Foundation (WPF). Для хранения данных системы использовалась СУБД MySQL Server 8, опосредованно, через API Web-сервиса, написанного на языке Python 3, с применением фреймворка Jango. Сбор сведений программно-аппаратного состава осуществлен с применением технологии Windows Management Instrumentation (WMI).

2.2. Концептуальная схема взаимодействия компонентов модели

Чтобы определить требования к исследуемой информационной модели необходимо исходить из базовых требований информационной безопасности организации.

Определим основные из требований информационной безопасности к клиентской АРМ:

1. состоит на инвентарном учете, определено материально-ответственное лицо и адрес объекта инвентаризации; АРМ соответствует эксплуатационным характеристикам согласно учету;
2. соответствие минимальным требованиям аппаратного состава (количество ядер процессора, разрядность процессора, количество ОЗУ, ВЗУ, комплектность);

3. соответствие минимальным требованиям к программному обеспечению (тип, версия, разрядность ОС, антивирус, средство от несанкционированного доступа, VPN-клиент, криптографический провайдер, специализированное ПО.

4. верификация программно-аппаратного состава АРМ должна перепроверяться при всех последующих изменениях.

Проанализировав существующие в данной области исследования программные решения было выявлено лишь частичное покрытие необходимого функционала имеющихся программных продуктов, а именно: удаленная установка программного обеспечения средствами комплекса мониторинга административных средств антивируса Касперского, получения сведений программно-аппаратного состава АРМ средствами VipNet Adminitstrator, антивируса Касперского, Secret Net Studio. В качестве инструментов для сбора сведений об АРМ пользователей также рассматривались такие продукты как AIDA64, Everest, HWMonitor. Эти программы отлично подходят для сбора сведений о системе, но не позволяют делать это удаленно.

Как итог, к разрабатываемой информационной модели предъявлены следующие требования: автоматизация мониторинга программно-аппаратного состава узлов сети; получение и анализ сведений по узлам сети (соответствующим и нет, требованиям безопасности); удаленное получение сведений по команде с конкретного узла (или нескольких); получение сведений о динамике изменений по устранению недостатков; учет технических средств; учет проводимых работ (ремонт технических средств); возможность добавления отчетной документации.

На основе выше изложенных требований предложена концептуально функциональная схема мультиагентной информационной модели информационной системы сбора и анализа сведений об автоматизированных рабочих местах (узлах сети) (см. рис. 2.1.).

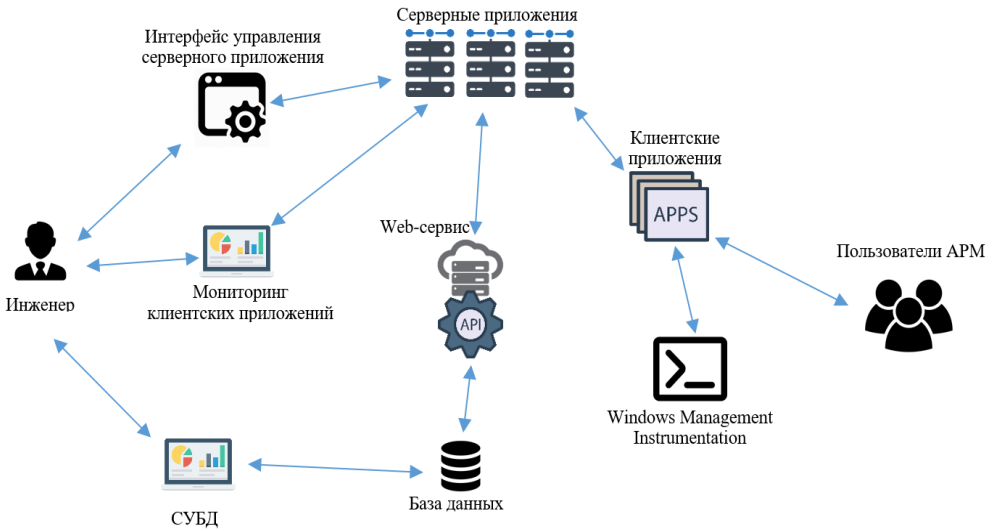

Рисунок 2.1 – Концептуально функциональная схема информационной модели

На основе предложенной схемы взаимодействия компонентов системы выработаны 2 основных сценария работы информационной модели:

Общий сценарий взаимодействия компонентов системы, ее пользователей и пользователей АРМ.

1. Выявление АРМ, на которых не установлено клиентское приложение системы, средствами анализа результатов SQL-запросов к базе данных Web-сервиса.

2. Удаленная установка клиентских приложений на узлы сети. Посылка команды клиенту на сбор и передачу сведений о программно-аппаратном составе АРМ:

– уведомление пользователя о необходимости заполнения сведений (инвентарный номер АРМ, адрес объекта, кабинет, ФИО материально-ответственного лица);

– получение системной информации и последующая передача на сервер;

– получение сведений сервером, передача средствами API Web-сервиса и последующая запись в БД.

3. Обработка на предмет выявления соответствия (или несоответствия) программно-аппаратного состава АРМ, предъявляемым требованиям, средствами предопределенных SQL-запросов и хранимых процедур.

4. Планирование работ по устранению недостатков на основе полученных сведений:

– в случае несоответствия только программных средств: установка или обновление до требуемой версии ПО;

– в случае несоответствия аппаратных средств – уведомление пользователя (средствами клиента, и/или по телефонной связи) о необходимости доставить АРМ на сервисное обслуживание или замены АРМ.

5. Подготовка отчетной документации об узлах, прошедших проверку.

6. Повторный запрос сведений (без уведомления пользователя) о системе после устранения несоответствий программных средств предъявляемым требованиям.

Сценарий мониторинга узлов сети прошедших проверку:

1. Обнаружение изменений в программно-аппаратном составе АРМ и отправка информации на сервер.

2. Передача на Web-сервис и запись сведений об изменениях в БД.

3. Отправка уведомления руководителю подразделения связи.

4. Делегирование задачи об устранении недостатков конкретному инженеру.

Сценарий мониторинга узлов сети не прошедших проверку:

1. Отправка повторного уведомления пользователя о необходимости предоставления сведений.

2. Устранение недостатков.

В предлагаемой концептуальной схеме и в соответствии с вышеописанными сценариями взаимодействия компонентов системы реализуется комплексное обеспечение мер по выполнению требований информационной безопасности организации.

Эксплуатация ИС (реализующая информационную модель) производится путем мониторинга сетевых узлов и последующего анализа получаемых данных (как из интерфейса серверного приложения, так и напрямую через получаемые результаты SQL-запросы к БД). На основании получаемых данных руководитель подразделения (или назначенный им инженер) определяет план работ по устранению недостатков и приступает к их устранению. После этого снова собираются и анализируются данные (отслеживание динамики).

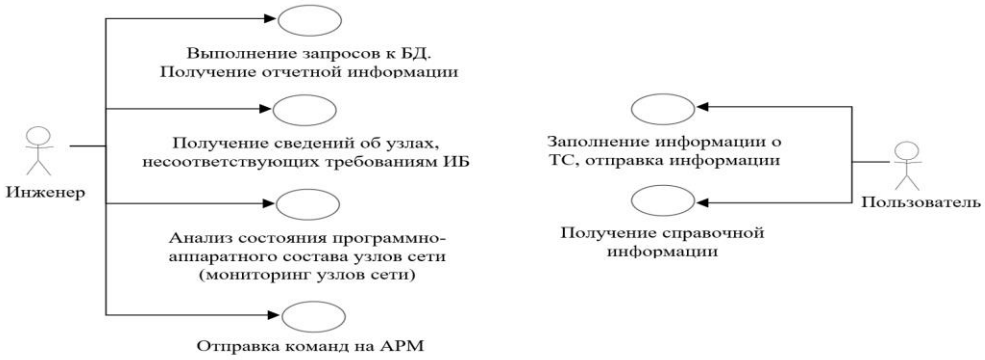

Рисунок 2.2 – Диаграмма вариантов использования информационной системы

2.3 Архитектура информационной системы

Основываясь не концептуальной модели, для реализации компонентов системы с учетом необходимой интеграции с существующим Web-сервисом и базой данных была выбрана следующая архитектура построения информационной системы (см. рис. 1.3).

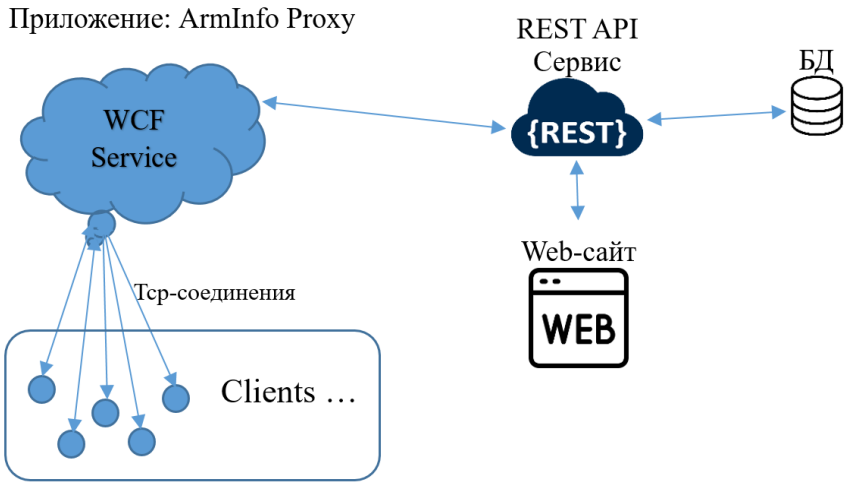

Рисунок 2.3 – Схема взаимодействия компонентов архитектуры информационной системы

Основной технологией взаимодействия между клиентскими и серверными приложениями выбрана технология Windows Communication Foundation. Ее безусловным преимуществом является возможность построения достаточно сложных программных коммуникационных решений с поддержкой различных протоколов межсетевого взаимодействия. Так как одним из обязательных требований, предъявляемых к информационной модели является оперативность получения сведений об изменениях состояния состава АРМ, а с другой стороны возможность автоматического обновления клиентских приложений в пределах локальной вычислительной сети, которая предполагает передачу данных достаточно большого объема (~10 Мб), стало очевидно, что протокол соединения между клиентским и серверным приложениями должен быть дуплексным, а соединение (по возможности) постоянным. Таким образом был выбран протокол TCP, который полностью соответствует поставленной задаче.

В тоже время серверное приложение взаимодействует сервисом с помощью REST API, посылая запросы и принимая ответы в форме JSON объектов. Таким образом для клиентских приложений происходит абстрагирование от существующей модели данных реализованной в базе данных. Достоинством данного подхода является то обстоятельство, что серверное приложение при своей инициализации полностью загружает все необходимые данные об объектах инвентаризации (подразделениях, адресах и т.п.), сведениях о номенклатуре, инвентарных номерах, соответствия версий ПО и аппаратном составе АРМ требованиям безопасности, и в таком состоянии является проксирующим сервером.

Назначение частей архитектуры информационной модели:

1. БД – централизованное хранилище информации о технических средствах.

2. REST API Сервис:

– является провайдером данных для ArmInfoProxy и Web-сайта (изолирует БД от других компонентов ИС);

– агрегирует информации об АРМ, поступающей от клиентских приложений (ArmInfoClient), через WCF-сервис (ArmInfoProxy), позволяя производить анализ изменений и выдачу предупреждений для администратора безопасности.

3. Web-сайт – часть ИС, позволяющая производить действия оперативного учета технических средств и формирования отчетной документации.

4. ArmInfo Proxy – Серверное приложение, реализующее WCF – сервис на основе дуплексных соединений TCP с клиентскими приложениями ArmInfoClient. Является проксирующим звеном между REST API сервисом и клиентскими приложениями, позволяя в любой момент времени запросить необходимую информацию с клиентской ПЭВМ и дополнительно предоставляет доступ ко внутренним информационным сервисам.

5. ArmInfo Client – клиентское приложение – агент. Является основным компонентом сбора информации о программном и аппаратном составе автоматизированного рабочего места, используя технологию WMI. Устанавливает tcp-сессию с одним или несколькими серверными приложениями ArmInfo Proxy, в рамках которой предоставляет доступ к внутренним сервисам и передает информацию о программно-аппаратном составе АРМ. В условиях сбоя сети и невозможности связаться с сервером, может устанавливать соединение с другими агентами в рамках доступного сегмента (подсети) для регистрации диагностической информации о сбое сети, информировании пользователя произошедшем инциденте и о возможных вариантах решения проблемы.

Клиент-серверное взаимодействие осуществляется средствами технологии WCF (Windows Communication Foundation). В общих словах она базируется на трех составляющих компонентах: адрес (Address), привязка (Binding), контракт (Contract). В качестве адреса для клиента выступает TCP/IP адрес сервера. Список адресов серверных приложений изначально предопределен в каждой версии клиента, и, если есть необходимость обновления данного списка необходимо обновить и само приложение, но вместе с тем архитектура приложения позволяет в перспективе внедрить механизм конфигурирования состояния клиентов. В качестве привязки выступает создание объектно-ориентированной абстракции протокола межсетевого взаимодействия, а именно TCP (NetTcpBinding). Логика самого клиент-серверного взаимодействия находится в сервисном контракте INetContract со стороны сервера и ICallback со стороны клиента. На представленной ниже диаграмме (см. рис. 2.5) изображены основные классы, составляющие логику клиент-серверного взаимодействия.

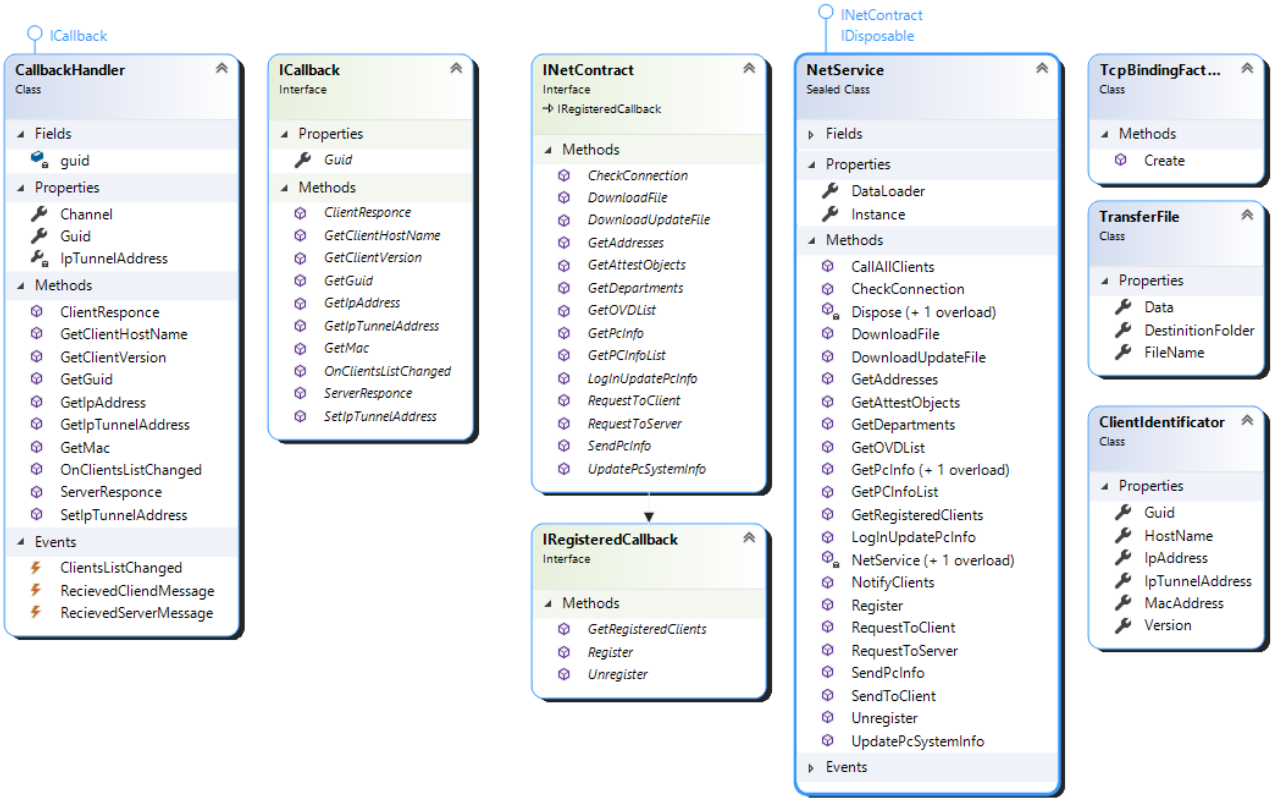

Рисунок 2.4 – Диаграмма классов, составляющих логику клиент-серверного взаимодействия

Реализация мультиагентного взаимодействия осуществляется путем обмена данными между серверными приложениями, которые и выступают в качестве агентов. В свою очередь, средой для этих агентов выступает API Web-сервисов и протоколы межсетевого взаимодействия клиентов. На приведенной ниже диаграмме (см. рис. 2.4) изображен процесс мультиагентного взаимодействия.

Путем внедрения дополнительных контрактов взаимодействия клиентов с несколькими серверными приложениями и последующим обновлением существующих клиентов можно добиться дальнейшего расширения функционала всей системы. При этом существующие контракты взаимодействия (передачи команд, отправки данных и обновления) не будут нуждаться в изменениях.

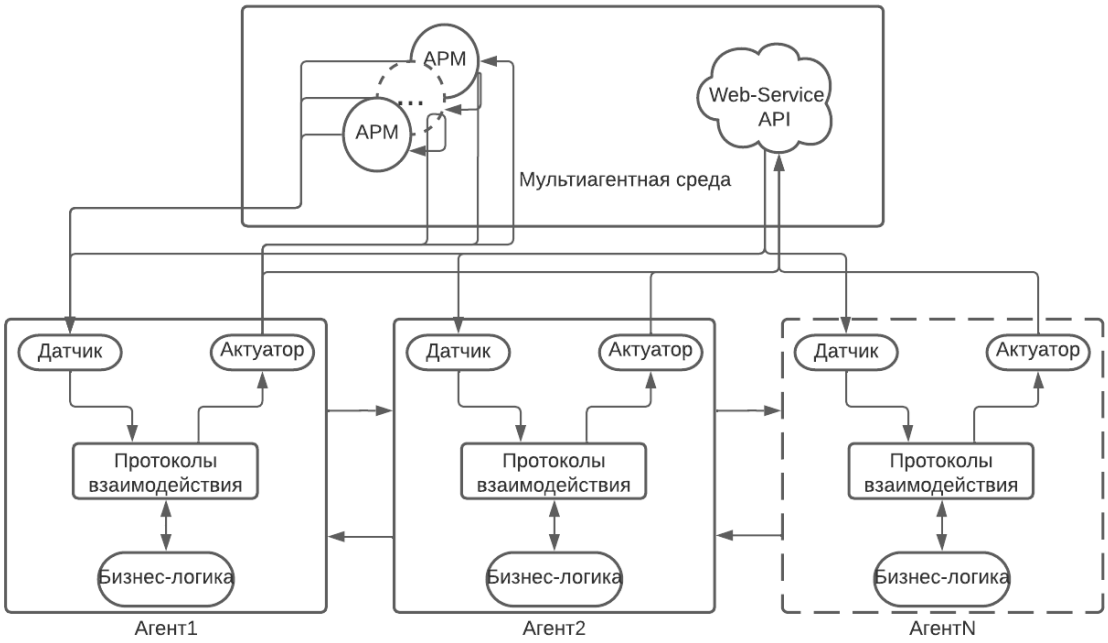

Рисунок 2.5 – Диаграмма мультиагентного взаимодействия

Применение мультиагентной архитектуры, в которой серверные приложения выступают в качестве агентов, а в качестве внешней среды выступают протоколы взаимодействия с клиентами и сервисами позволяет обеспечить масштабируемую надежность системы в целом. Так, при возникновении сбоя работы Web-сервиса серверное приложение, которое уже произвело загрузку данных ранее – замещает Web-сервис, выступая поставщиком данных на время его недоступности. Когда Web-сервис снова становится доступным, все серверные приложения отправляют на него полученные от клиентов данные. В свою очередь клиенты могут подключаться к любому доступному серверу.

3. Выводы

В данной статье была описана мультиагентная информационная модель сбора и анализа программно-аппаратного состава узлов локальной вычислительной сети организации.

Применение информационной системы, построенной на архитектуре описанной информационной модели, сможет обеспечить более эффективное выполнение комплекса мер обеспечения информационной безопасности, что позволит более прогнозируемо и эффективно использовать трудовые ресурсы и повысить качество информационных процессов, путем внедрения автоматизации сбора и анализа информации по программно-аппаратному составу парка ПЭВМ организации.

Библиографический список

1. Адеева М. Г., Айгумов К. М. Привязка и контекст данных в wpf с использованием свойств и ресурсов фреймворка // Информационные технологии в экономике и управлении. 2020. С. 179-181.
2. Бакиев Л. Ю., Борисов А. Н. Разработка клиент-серверного приложения ежедневника // Актуальные проблемы науки в студенческих исследованиях. 2017. С. 302-303.
3. Баранкова И. И., Михайлова У. В., Лукьянов Г. И. Разработка приложений на C# для работы с базами данных. М., 2018.
4. Бужинская Н. В., Котелевец Н. О. Разработка программного продукта для автоматизации учета оборудования на предприятии // Аллея науки. – 2019. Т. 4. №. 1. С. 953-957.
5. Иванов А. А., Колташёва Д. Д., Шимов И. В. Разработка приложения на языке C# с использованием Microsoft SQL Server // Актуальные вопросы преподавания математики, информатики и информационных технологий. 2017. №. 2. С. 186-191.
6. Королёнок А. Ю. Использование инструментария управления Windows (WMI) для получения системной информации с применением техник объектно-ориентированного программирования // Постулат. 2020. №. 12.
7. Костин Е. И., Чумак И. В. Создание службы wcf для использования в клиент-серверном приложении //инновационные подходы в решении научных проблем. 2021. С. 121-126.
8. Кузина М. Д., Семенов А. А. Разработка системы тестирования на основе платформы ASP. NET // Инновационное развитие легкой и текстильной промышленности. 2017. С. 50-52.
9. Назмутдинов А. Х. Особенности применения паттерна mvvm в клиентских приложениях с использованием платформы пользовательского интерфейса wpf //молодежь и системная модернизация страны. 2019. С. 131-134.
10. Belenesi D. T., Gabor G., Moisi E. V. Comparative study on WPF and UWP Frameworks used in RSS Application //2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI). IEEE, 2021. С. 1-6.
11. Daadoo M. Developing and Implementation of Distributed Chat Applications using WPF and WCF. 2017.
12. Sannarangaiah K. Design and development of a graphical user interface by state-of-the-art C# patterns. 2020.