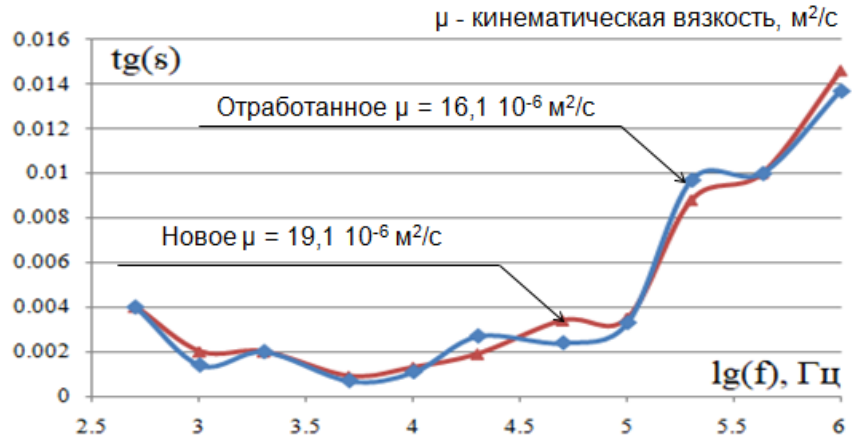

Рисунок 3 – Зависимость диэлектрических потерь гидравлического масла от частоты электрического поля

В результате проведенных исследований установлено:

1. В процессе эксплуатации гидравлического масла увеличивается значение кинематической вязкости. Для марки масла Esso Nuto H 46 кинематическая вязкость увеличилась с 16,1 10^{-6} м^2/с до 19,1 10^{-6} м^2/с.
2. Относительная диэлектрическая проницаемость образцов отработанного масла в среднем на 1,5 % больше по сравнению с образцами нового масла.
3. Значения тангенса диэлектрических потерь близки для исследуемых образцов.

На основании полученных данных можно сделать вывод, что диэлектрический метод контроля позволяет проводить оценку качества гидравлических масел.

Список использованных источников

1. Науменко, А. М. Разработка системы контроля качества гидравлического масла / А. М. Науменко [и др.] // Материалы докладов 52-й МНТК преподавателей и студентов УО «ВГТУ». – Витебск, 2019. – С. 17–19.

УДК 004.4

РАЗРАБОТКА ЭЛЕКТРОННОЙ СИСТЕМЫ ДОКУМЕНТООБОРОТА

Ринейский К.Н., ст. преп., Казаков В.Е., к.т.н., доц., Старикович Ю.А., маг.

Витебский государственный технологический университет,
г. Витебск, Республика Беларусь

Реферат. *В статье рассматриваются вопросы, связанные с разработкой клиент-серверного приложения, интегрируемого в информационную систему университета: общие архитектурные решения, проблемы и их решения при реализации.*

Ключевые слова: монолитная архитектура, микросервисая архитектура, RPC, Evtnts, брокер сообщений.

Электронный документооборот в учреждении образования является неотъемлемым компонентом на современном этапе развития общества.

Существует два основных подхода при формировании структуры таких систем: построение монолитной архитектуры; разбивка на микро сервисную архитектуру.

Концепция монолитного программного обеспечения (рис. 1) заключается в том, что различные компоненты приложения объединяются в одну программу на одной платформе. Обычно монолитное приложение состоит из базы данных, клиентского пользовательского интерфейса и серверного приложения. Все части программного обеспечения унифицированы, и все его функции управляются в одном месте.

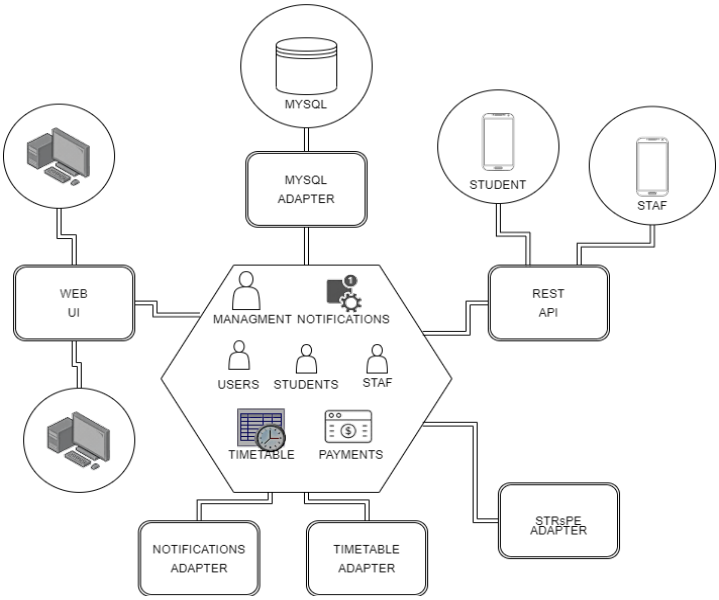

Рисунок 1 – Краткая схема монолитной архитектуры

Проблемы больших монолитных систем: плохое горизонтальное масштабирование; плохая отказоустойчивость; сложность внедрения новых технологий; сложность рефакторинга; проблема в поиске сотрудников и т. д.

В данном подходе университет будет испытывать следующие проблемы:

– поиск сотрудников для поддержания и развития проекта. Трудно привлечь молодого специалиста (студента) или квалифицированного сотрудника со стороны, так как не все захотят и не смогут работать на проекте с большим объемом исходного кода и устаревшими технологиями;

– невозможно использовать современные подходы и технологии, которые улучшили бы проект.

Другим подходом является – микросервисная архитектура. Данный подход является более перспективным для решения задач выбранного типа.

Суть подхода очень проста (рис. 2). Большая система разбивается на много маленьких и вроде бы делает жизнь лучше и проще, но на самом деле проблемы перетекают на другой уровень – уровень взаимодействия.

Плюсы: горизонтальное масштабирование; отказоустойчивость; масштабирование команд разработчиков; переиспользование сервисов; гибкость стека.

Минусы: сложность реализации; дорого на начальном этапе; каждый сервис атомарный; несогласованность данных (отсутствие классических транзакций).

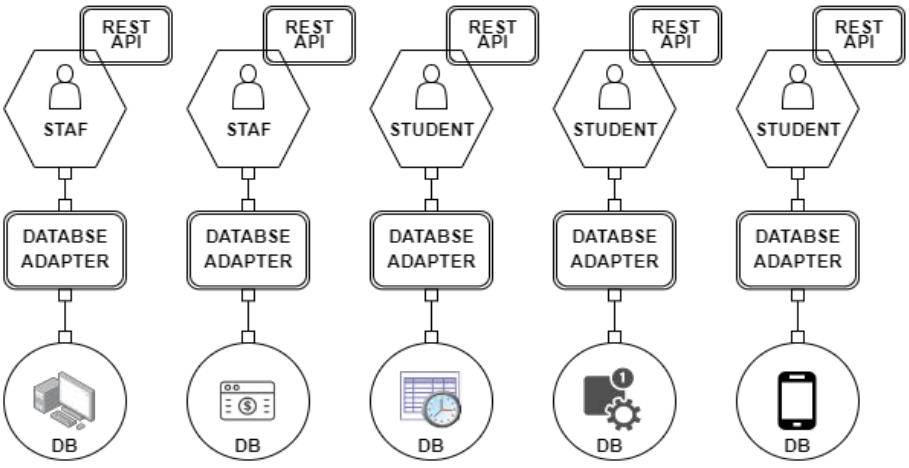

Рисунок 2 – Краткая схема микросервисной архитектуры

Мы стремимся к тому, чтобы каждый сервис был полностью отделен от других.

VCS/CI/CD на сервис: один сервис – один репозиторий, один сервис – один CI-конвейер, один сервис– один CD-конвейер.

Это значит, что у одного сервиса свой репозиторий имеет собственный конвейер сборки и собственный конвейер развертывания, собственная БД.

Для взаимодействия сервисов есть два подхода: синхронный RPC (HTTP REST); асинхронный Events (FMQP, MSMQ и т.д.).

Недостатками первого, в нашем случае, являются: отложеная согласованность; широковещательные сообщения; отложенные запросы.

Поэтому выбираем Events, асинхронные взаимодействия на шинах данных. Будем использовать независимый от платформы протокол AMQP и брокер сообщений RabbitMQ. Это позволит нам использовать в нужных местах, нужные языки и тем самым не привязываться к стеку, а взаимодействия всех сервисов будет происходить через events, то что нам и нужно.

На стороне клиентов есть какой-то frontend и они скорее всего захотят общаться с backend`ом через Rest API, для этого используем ApiGateway. ApiGatway – сервис, который позволяет создать тонкий фасад на сетевом уровне, к которому мы подключаем все наши микросервисы, тем самым делая единую точку входа для fornt`a.

Так как у нас микросервисная архитектура, нам надо решить проблему с логированием. Для этого все запросы связываются единым ID, который формируется на уровне ApiGateway и потом добавляется ко всем запросам, events которые идут во внутреннюю систему, таким образом можно использовать ActivityID.

Использование данного подхода положено в основу проекта электронного документооборота ВГТУ.

Список использованных источников

1. Amazon API Gateway [Электронный ресурс]. – Режим доступа: https://aws.amazon.com/ru/api-gateway/getting-started/. – Дата доступа: 15.05.2020.
2. RannitMQ [Электронный ресурс]. – Режим доступа: https://www.rabbitmq.com /#getstarted. – Дата доступа: 15.05.2020.

УДК 004.4

SCADA-СИСТЕМА КОНТРОЛЯ ТЕМПЕРАТУРЫ

Самусев А.М. маг., Кузнецов А.А., проф., Ринейский К.Н. ст. преп., Чернов Е.А., ст. преп.

Витебский государственный технологический университет, г. Витебск, Республика Беларусь

Реферат. Статья посвящена разработке программы дистанционного мониторинга температурных параметров промышленного объекта.

Ключевые слова: SCADA, SCADA-система, MasterSCADA, контроль температуры.

SCADA (supervisory control and data acquisition, диспетчерское управление и сбор данных) – программный пакет, предназначенный для разработки или обеспечения работы в реальном времени систем сбора, обработки, отображения и архивирования информации об объекте мониторинга или управления. SCADA может являться частью АСУ ТП, АСКУЭ, системы экологического мониторинга, научного эксперимента, автоматизации здания и т. д.

Разработанный проект на основе SCADA (MasterSCADA) контроля температуры направлен на сбор и архивирование данных, непрерывный мониторинг температурных показателей распределённой системы и уведомление оператора о нештатных ситуациях. Структура проекта имеет вид (рис. 1):