

## АНАЛИЗ АРХИТЕКТУРНЫХ РЕШЕНИЙ И ПРОЕКТИРОВАНИЕ АРХИТЕКТУРЫ ПРОГРАММНОГО СРЕДСТВА РАСЧЕТА ОБЪЕМА УЧЕБНОЙ РАБОТЫ, ВЫПОЛНЯЕМОЙ ШТАТНЫМ СОСТАВОМ ППС КАФЕДРЫ

*Адамонис А.В.*

*Белорусский государственный университет информатики и радиоэлектроники,  
г. Минск, Республика Беларусь*

*Научный руководитель: Шелест А.В. – магистр техники и технологии, ассистент кафедры ППИКС*

**Аннотация.** Проанализированы архитектурные решения для реализации серверной части программного средства расчета объема учебной работы, выполняемой штатным составом ППС кафедры. Спроектирована архитектура программного средства.

**Ключевые слова:** программное средство, архитектура программного средства, нотация C4Model

**Введение.** Проектирование архитектуры для реализации программного средства, направленного на расчет объема учебной работы, выполняемой штатным составом ППС кафедры, является ключевым этапом в разработке программного средства. Для грамотного проектирования архитектуры необходимо проанализировать основные архитектурные подходы для реализации программных средств.

Цель данного анализа заключается в оценке преимуществ и недостатков различных подходов к архитектуре программного средства, способствующих достижению высокой эффективности и функциональности системы расчета объема учебной работы, выполняемой ППС кафедры.

**Основная часть.** Выбор архитектуры для реализации серверной части программного средства расчета объема учебной работы, выполняемой штатным составом ППС кафедры, предстоял между монолитной и микросервисной архитектурами. Эти два подхода имеют существенные отличия, и каждый из них имеет свои преимущества и недостатки [1].

Монолитная архитектура представляет собой единое приложение, в котором все компоненты и функции находятся в одной кодовой базе. Это удобно для небольших проектов. Однако, с ростом функциональности и сложности приложения, монолит может стать трудным в поддержке и масштабировании. Изменения в одной части приложения могут оказать влияние на другие части, что усложняет поддержку и разработку.

Микросервисная архитектура, в свою очередь, представляет из себя набор независимых, маленьких сервисов, каждый из которых выполняет конкретную функцию или задачу. Они работают в рамках своих собственных процессов и могут использовать разные технологии и базы данных. Этот подход обеспечивает высокую гибкость, позволяя разрабатывать и обновлять сервисы независимо друг от друга [2].

Сравнив два различных подхода, выбор пал на микросервисную архитектуру. Этому послужили следующие причины:

1 Программное средство для расчета объема учебной работы, выполняемой штатным составом ППС кафедры, имеет множество систем, которые могут быть интегрированы с ним. Это могут быть системы для различных аспектов образовательного процесса, такие как работа с учебными планами, расписанием занятий, учетом успеваемости студентов. Таким образом, программное средство, реализованное в микросервисной архитектуре, имеет возможности для расширения функциональности путем интеграции с другими образовательными приложениями и системами, которые можно реализовать на других языках программирования.

2 Микросервисная архитектура обеспечивает более гибкое управление сложностью системы, разбивая ее на мелкие части, которыми легче управлять.

Проектирование архитектуры программного средства выполнено в нотации C4 Model. Данная нотация представляет архитектуру программного средства в виде 4-х уровней декомпозиции, где каждый последующий уровень более детально раскрывает предыдущий.

На рисунке 1 изображен контейнерный уровень архитектуры программного средства в нотации C4 Model. На данном уровне архитектуры выделены основные контейнеры программного обеспечения – среды, где запускается и функционирует код. Программное средство состоит из следующих контейнеров:

- Web Application;
- Api Gateway;
- Api Application University System с базой данных;
- Api Application Edy Load System с базой данных;
- E-mail System.

Контейнерный уровень дает детальное представление о внутреннем строении системы, позволяя увидеть, какие части приложения работают вместе, как они связаны между собой и каким образом происходит внутреннее взаимодействие.

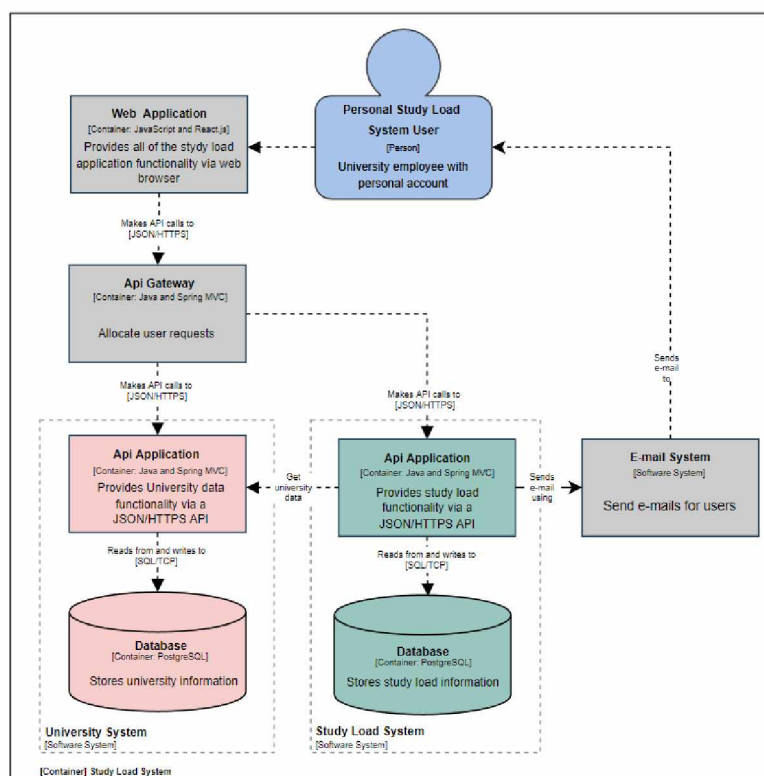


Рисунок 1 – Контейнерный уровень архитектуры программного средства в нотации C4Model

Разрабатываемое программное средство для расчета объема учебной работы, выполняемой штатным составом ППС кафедры было разделено на 4 микросервиса, каждый из которых выполняет свою уникальную функцию и играет ключевую роль в обеспечении эффективности и гибкости системы. Выделены следующие микросервисы:

1 Gateway – сервис-шлюз, который выполняет ключевую роль в системе, предоставляя точку входа для запросов. Он маршрутизирует входящие запросы между различными микросервисами, обеспечивая им балансировку нагрузки. Это позволяет управлять и контролировать трафик, а также обеспечивает централизованную точку доступа к системе, улучшая безопасность и масштабируемость.

2 University System – сервис, который ответственен за основную работу с пользователями, информацией о ВУЗе и связанной с учебным процессом статистикой. Сервис обеспечивает централизованное хранение данных и доступ к ним, что позволяет эффективно управлять информацией, связанной с учебным заведением.

3 E-mail System – сервис, который отвечает за отправку сообщений по электронной почте сотрудникам учебного заведения. Он обеспечивает надежный механизм связи и информирования, играя важную роль в коммуникациях между сотрудниками и другими участниками учебного процесса.

4 Edu Load System – сервис, который отвечает за расчет учебной работы, выполняемой ППС кафедры. Выделение этого сервиса позволяет централизованно управлять сложным процессом распределения нагрузки между преподавателями кафедры, повышая эффективность и сокращая возможные конфликты или перегрузки у преподавателей.

**Заключение.** Таким образом, проанализированы основные архитектурные решения для разработки серверной части программного средства для расчета объема учебной работы, выполняемой штатным составом ППС кафедры. Выбранный микросервисный подход позволяет эффективно управлять сложными процессами и обеспечить высокий уровень гибкости и функциональности программного средства.

Проектирование архитектуры программного средства выполнено с применением нотации C4 Model. Разработанная архитектура включает в себя четыре ключевых микросервиса, каждый из которых выполняет определенную функцию, необходимую для эффективного функционирования системы.

#### **Список литературы**

1. Ford N., Richards M. *Fundamentals of Software Architecture: An Engineering Approach. A Comprehensive Guide to Patterns, Characteristics, and Best Practices* / N. Ford, M. Richards. USA : Sebastopol, 2020. – 244 p. – ISBN 978-1-492-04345-4.

2. Ричардсон К. *Микросервисы. Паттерны разработки и рефакторинга* / К. Ричардсон. СПб : Санкт-Петербург 2020. – 544 с. – ISBN 978-5-4461-0996-8.

UDC 621.3.049.77–048.24:537.2

## **ANALYSIS OF ARCHITECTURAL SOLUTIONS AND DESIGN OF ARCHITECTURE OF A SOFTWARE TOOL FOR CALCULATING THE VOLUME OF STUDY WORK PERFORMED BY THE STAFF OF THE DEPARTMENT'S TEACHING STAFF**

*Adamonis A.V.*

*Belarusian State University of Informatics and Radioelectronics, Minsk, Republic of Belarus*

*Shelest A.V. – master of engineering and technology, assistant of the department of ICSD*

**Annotation.** Architectural solutions for the implementation of the server part of the software tool for calculating the volume of educational work performed by the staff of the department's teaching staff are analyzed. The software architecture was designed.

**Keywords:** software, software architecture, C4Model notation