

ПРОЕКТИРОВАНИЕ АВТОМАТИЗИРОВАННОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ УПРАВЛЕНИЯ ОБУЧЕНИЕМ С ПРИМЕНЕНИЕМ НОТАЦИИ С4

Пихтовников С.В., к.х.н., доцент,
Файзрахманова К.Э., к.т.н., доцент,
Пихтовников А.С., студент
УУНиТ, г. Уфа, Россия

Аннотация. В статье рассматривается подход к архитектурному проектированию автоматизированной информационной системы управления обучением с использованием нотации С4. Описаны два уровня модели: контекст системы и контейнерная диаграмма, отражающие взаимодействие пользователей, внешних сервисов и внутренних компонентов. Предложенный подход обеспечивает наглядность архитектурных решений и снижает риски при разработке и сопровождении системы. Результаты могут быть применены при проектировании аналогичных образовательных информационных систем.

Ключевые слова: С4-модель, архитектура программного обеспечения, дополнительное профессиональное образование, контейнерная диаграмма, информационная система.

Сфера дополнительного профессионального образования (ДПО) характеризуется активным внедрением цифровых технологий. Образовательные организации постепенно переходят от бумажного документооборота к онлайн-обучению, электронным договорам и автоматизированному учету успеваемости. В результате возрастает потребность в специализированных информационных системах, которые охватывают полный жизненный цикл работы со слушателем, начиная с подачи заявки и заканчивая выдачей документа об обучении.

Создание таких систем связано с рядом архитектурных сложностей. Требуется объединить модули учебного процесса, защиты контента, электронного документооборота, а также внешние сервисы оплаты и электронной подписи. При этом архитектура должна быть понятна не только разработчикам, но и заказчикам, включая методистов, менеджеров и администраторов.

Для решения этой задачи используется нотация C4, предложенная С. Брауном [1]. Она представляет систему на четырех уровнях детализации: от общего контекста до уровня кода, что позволяет адаптировать представление системы под разные группы заинтересованных лиц. В данной работе рассматриваются первые два уровня нотации C4 при проектировании автоматизированной информационной системы управления обучением.

Нотация C4 (Context, Containers, Components, Code) представляет собой инструмент документирования архитектуры программных систем. В отличие от UML-диаграмм, C4 ориентирована на коммуникацию, при этом каждый уровень предназначен для своей аудитории – от бизнес-заказчиков до разработчиков. Первый уровень, контекст системы, показывает систему как единое целое в окружении пользователей и внешних сервисов. Вторым уровнем, контейнеры, раскрывает внутреннюю структуру системы и выделяет развертываемые компоненты, такие как приложения, базы данных и сервисы. Третий уровень описывает компоненты внутри контейнеров, а четвертый соответствует структуре кода, включая классы и модули. К ключевым принципам нотации относятся явное указание технологий в элементах диаграммы, подписи всех связей с указанием протоколов, а также единообразное обозначение акторов и систем. Соблюдение этих правил повышает читаемость архитектуры и снижает вероятность ошибок.

На уровне контекста АИС взаимодействует с пятью категориями пользователей: системным администратором, преподавателем, методистом, менеджером и слушателем. Все пользователи работают через протокол HTTPS, обеспечивающий безопасную передачу данных. Система также интегрируется с

три внешними сервисами. Платежный модуль подтверждает факт оплаты через REST-API; для делегированного доступа к ресурсам используется протокол авторизации OAuth 2.0. Сервис электронной подписи принимает документы и возвращает подписанные версии через REST-API. Почтовый сервер используется для отправки договоров и уведомлений по протоколу SMTP. Диаграмма контекста (рисунок 1) позволяет участникам проекта сформировать единое понимание границ системы без детализации реализации.

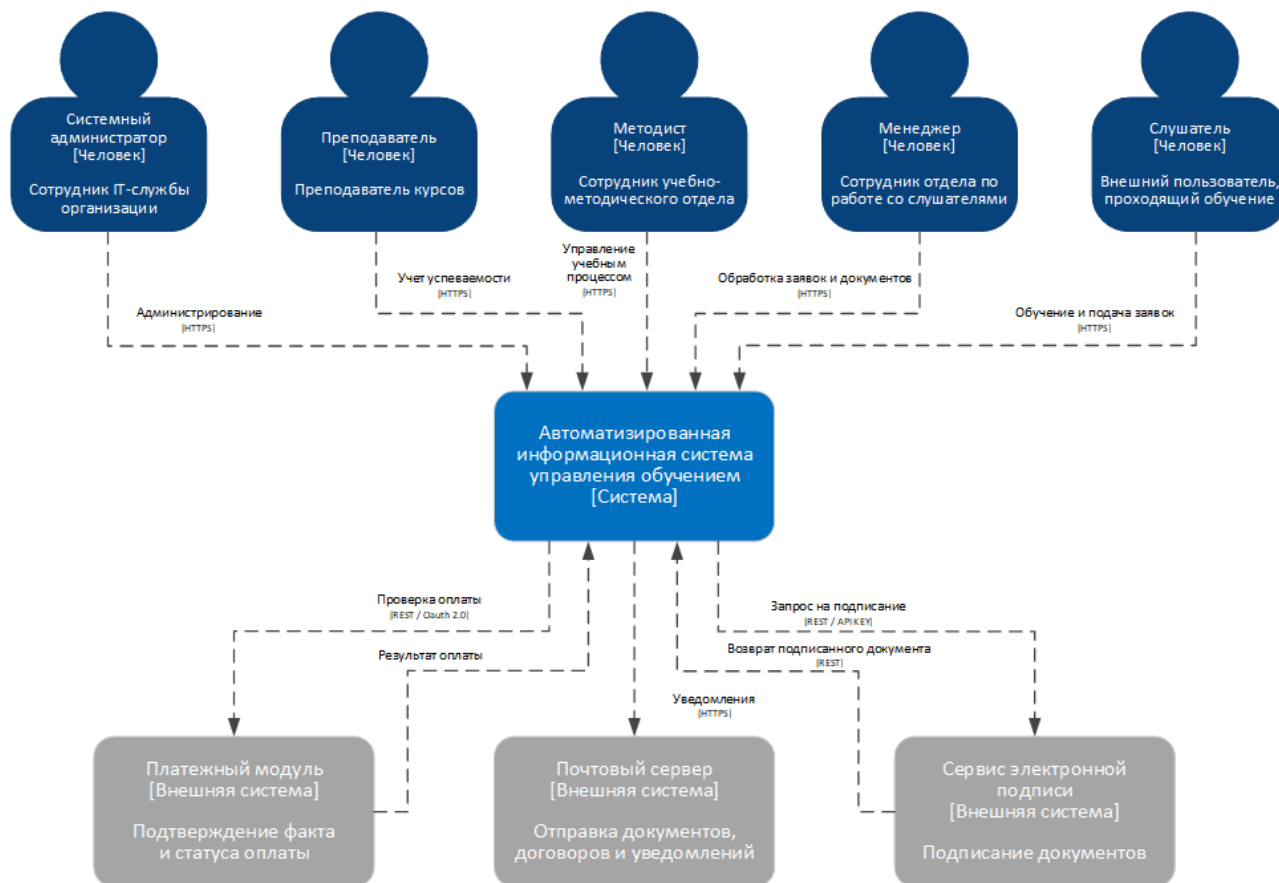


Рисунок 1 – Диаграмма контекста АИС

Контейнерная диаграмма (рисунок 2) показывает внутреннюю структуру АИС. Система включает восемь контейнеров, каждый из которых является отдельной развертываемой единицей. Веб-клиент реализован на Bootstrap 5 и JavaScript и предоставляет интерфейс для всех ролей пользователей. Веб-сервер на базе Nginx выполняет функции обратного прокси, терминацию TLS и раздачу статических файлов, принимая HTTPS-запросы и перенаправляя их на сервер приложений по HTTP.

Сервер приложений (Python 3.11, Django 5, Gunicorn) реализует основную бизнес-логику системы. Он включает модуль регистрации и управления договорами, модуль учебного процесса, модуль документооборота, подсистему DRM (Digital Rights Management) и REST-API. Обработчик фоновых задач построен на Celery (рабочие процессы) и Redis (брокер сообщений). Celery используется для генерации документов, пакетного подписания квалифицированной электронной подписью (КЭП), анализа поведения слушателей с использованием искусственного интеллекта (ИИ) и рассылки уведомлений. Redis выделен в отдельный контейнер и используется для кэширования сессий и ограничения частоты запросов.

Сервис видеохостинга (FFmpeg, HLS, MinIO S3) обеспечивает загрузку, транскодирование видеолекций и их потоковую передачу по протоколу HLS (HTTP Live Streaming). Сервис защиты контента (Python, FastAPI) выполняет шифрование и маркировку учебных материалов, а также проверяет права доступа через базу данных. База данных PostgreSQL 15 хранит информацию о слушателях, договорах, зачислениях, DRM-доступах, журнал аудита и инцидентов информационной безопасности. Файловое хранилище MinIO S3 используется для хранения учебных материалов, шаблонов и подписанных документов.

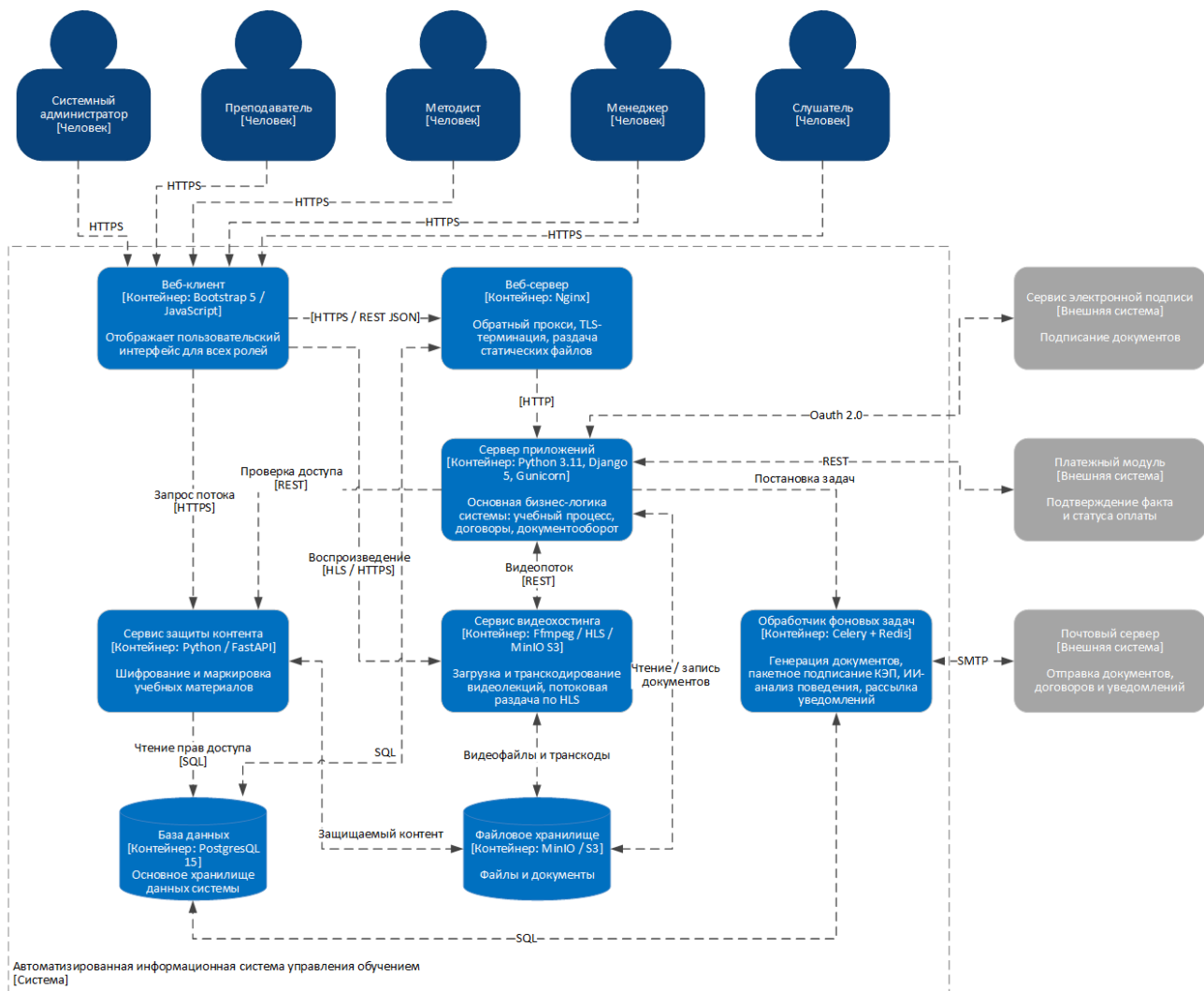


Рисунок 2 – Контейнерная диаграмма АИС

Контейнерная архитектура обеспечивает горизонтальное масштабирование. Сервер приложений, фоновые задачи и сервис видеохостинга могут масштабироваться независимо друг от друга. Использование MinIO как S3-совместимого хранилища позволяет в будущем перейти на облачные решения без изменения программного кода.

Применение нотации C4 при проектировании АИС позволило выявить архитектурные закономерности и потенциальные риски на раннем этапе: в частности, необходимость защиты каналов передачи данных между внешними сервисами и ядром системы, а также риски, связанные с зависимостью от сторонних платежного модуля и сервиса электронной подписи. Явное указание протоколов взаимодействия (HTTPS, REST, SQL, SMTP, OAuth 2.0, HLS) позволяет заранее оценить требования к безопасности каналов передачи

данных. Выделение внешних сервисов за границы системы на обоих уровнях C4 помогает четко определить зоны ответственности разработчиков.

Предложенная модель формализует взаимодействие компонентов системы и создает основу для дальнейшей детализации архитектуры на компонентном (уровень 3) и кодовом (уровень 4) уровнях. Перспективой развития является разработка компонентных диаграмм для DRM-подсистемы и модуля документооборота, а также формирование ADR-документов (Architecture Decision Records) для обоснования ключевых архитектурных решений.

Литература

1. The C4 model for visualising software architecture [Электронный ресурс]. URL: <https://c4model.com/> (дата обращения: 15.05.2026).