

Описание технической архитектуры АИС «УНИД»

Содержание

1. Общие сведения о системе	2
1.1. Наименование системы и назначение	2
2. Общие сведения об архитектуре проекта.	3
2.1.1. Общая схема приложения, составляющие, их взаимодействие	3
2.1.2. Как приложение встраивается в ИТ-ландшафт предприятия, необходимое окружение	4
2.2. Описание технической архитектуры программного обеспечения	4

1. Общие сведения о системе

1.1. Наименование системы и назначение

Наименование системы у разработчика: Управление научной и инновационной деятельностью» (УНИД).

АИС «УНИД» — информационная система Управления Научной и Инновационной Деятельностью для научно-образовательных и академических организаций. Система предназначена для учета результатов научной деятельности сотрудников, аспирантов и студентов; автоматизирует процессы планирования, контроля сроков и качества, анализа эффективности научно-исследовательской и инновационной деятельности организации, её кадровых и финансовых ресурсов. УНИД обеспечивает единое информационное пространство, позволяющее повышать всем участникам бизнес-процессов свою эффективность за счёт оптимального взаимодействия.

АИС «УНИД» включает в себя совокупность функциональных модулей, объединенных в единую систему с общим сервисом авторизации.

2. Общие сведения об архитектуре проекта.

2.1.1. Общая схема приложения, составляющие, их взаимодействие

На рис. 1 схематично представлена схема взаимодействия всех составляющих, участвующих в работе. Основной блок, в который собраны данные элементы – «Экземпляр сервера УНИД (кластер, виртуальная машина и пр.)», среди данных элементов можно выделить СУБД, Backend и Frontend.

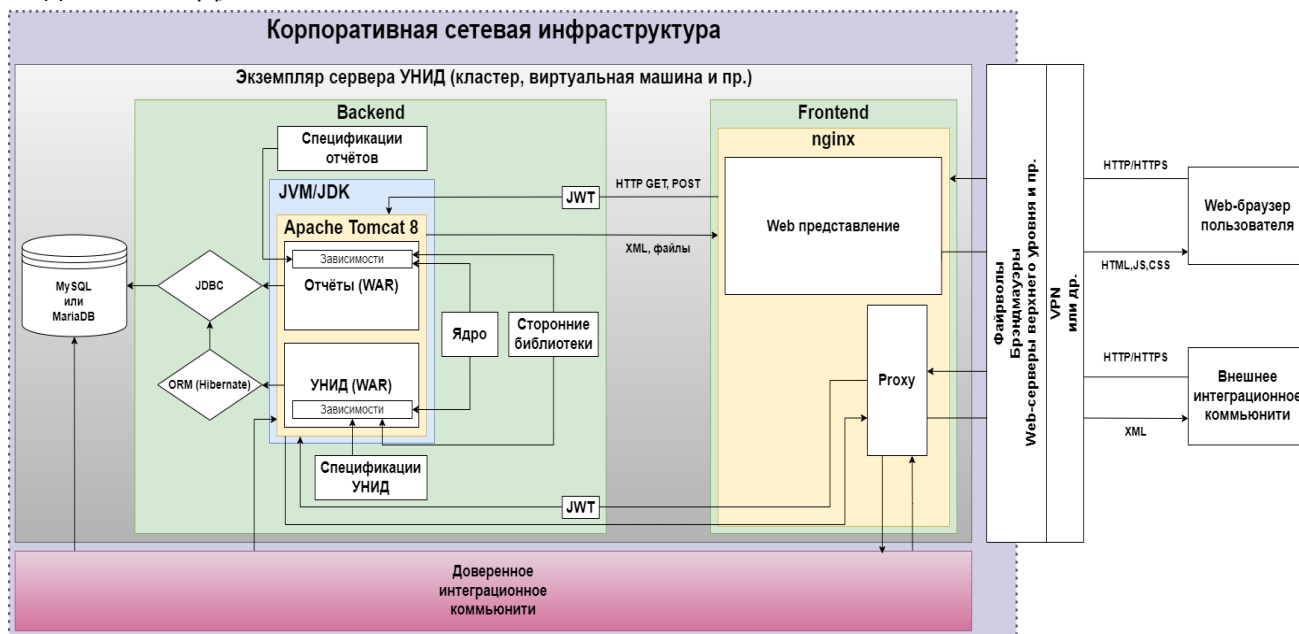


Рис.1. Общая схема приложения, составляющие, их взаимодействие

Коммуникация с БД осуществляется внутренними средствами Backend компонента с использованием JDBC драйвера и средств ORM библиотеки (Hibernate).

Контейнер сервлетов Apache Tomcat выступает в роли web-сервера, предоставляющего REST API сервисы. Работает Apache Tomcat под управлением виртуальной машины JAVA.

Структурно Backend приложений 2 – основное приложение УНИД и сервис отчётов. Каждое из них использует как собственный набор библиотек (ядро), так и сторонние библиотеки (например, Apache Commons IO, JDOM2, Hibernate и пр.). Помимо используемых зависимостей, backend приложения используют различные спецификации, характерные для тех или иных заказчиков.

Frontend – самостоятельный компонент, который представляет из себя web-приложение, состоящее из набора HTML, JS, CSS и пр. элементов. Работа frontend компонента находится под управлением web-сервера (наиболее популярный вариант - nginx). Коммуникация между frontend и backend классическая – on demand HTTP client-server request – т.е. web-представление по мере необходимости отправляет на backend HTTP запрос, получает в ответ нужную ему информацию и отображает визуальную составляющую у пользователя в web-браузере. Сервер же, в свою очередь, не имеет возможности общения с клиентом. Аутентификация пользователей осуществляется при использовании JWT (JSON Web Token) механизмов.

2.1.2. Как приложение встраивается в ИТ-ландшафт предприятия, необходимое окружение

Учитывая специфики и особенности бизнес-процессов, а так же спецификацию АИС УНИД, как учётной информационной системы, внутри корпоративной сетевой инфраструктуры (ИТ-ландшафта) организации экземпляр развёрнутой системы АИС УНИД должен быть закрыт от внешнего мира средствами, выбор которых стоит за организацией, это может быть VPN, различные сетевые экраны, фаерволы и пр. инструменты. Данный подход обеспечит повышенный уровень безопасности.

Доверенное интеграционное комьюнити (другие информационные системы и ресурсы, находящиеся в внутри корпоративной сетевой инфраструктуры) – элементы интеграционной среды, полностью подконтрольные организации, имеют возможность работать с серверным REST API АИС УНИД напрямую. В некоторых особых случаях даже напрямую с БД, что в свою очередь значительно уменьшает объём звеньев интеграционных потоков, увеличивает скорость обмена данными и т.д.

Внешнее интеграционное комьюнити, если таковое есть, доступ к имеющимся сервисам и данным АИС УНИД могут получить только на общих основаниях – должны быть учётные данные, прохождение аутентификации и авторизации.

Как для внешнего интеграционного комьюнити, так и для web-представления, отображаемого на экране пользователя в web-браузере, коммуникация с REST API сервера осуществляется через перенаправление HTTP/HTTPS в nginx.

2.2. Описание технической архитектуры программного обеспечения

АИС УНИД – информационно-аналитическая система, выполненная в классической клиент-серверной архитектуре web-приложения.

В качестве структурных элементов, составляющих основу АИС УНИД можно вынести:

- Слой пользовательского представления (далее frontend), как элемент интерактивного взаимодействия с пользователем;
- Слой сервисной части (далее backend), как элемент DM, DD, DC и др. операций, осуществляемых в процессе штатной работы системы.
- БД слой хранения данных, готовое решение в виде стороннего ПО, в качестве задач которого выступают хранение, изменение, представление, аудит изменений и др. операции работы с пользовательскими и системными данными.

Вышеупомянутые элементы структуры технически не имеют жесткой связи, могут быть собраны, установлены и запущены независимо друг от друга, но для корректной штатной работы системы в целом, работа всех элементов должна быть подготовлена, сконфигурирована, настроена и синхронизирована.

Коммуникация frontend и backend элементов осуществляется посредством HTTP запросов (по протоколу HTTP или HTTPS, в зависимости от конфигурации и инфраструктуры ИТ-ландшафта организации). Backend, в свою очередь, в зависимости от характера запроса,

производит те или иные действия (выбирает данные и отдаёт frontend'у для отображения, сохраняет/изменяет/удаляет полученные от frontend'a данные, производит какие то сервисные операции с данными, опираясь на backend бизнес-логику, и др.).

Развёртывание конечных исполнимых серверных приложений по умолчанию осуществляется с использованием контейнера сервлетов Apache Tomcat 8. Однако, возможны вариации в области выбора сервисного ПО для развёртывания, например JBoss Wildfly или др. решения, предоставляющие необходимый функционал.

Коммуникация backend и БД реализована с использованием ORM прослойки и JDBC драйвера.

Архитектурный стиль системы в целом, как описано в п. 3.a – классическое клиент-серверное web-приложение. Как сказано в п. 3.b. – frontend и backend не имеют никакой жесткой связи, по сути – это отдельные проекты, каждый из которых выполнен в своём архитектурном стиле с учётом frontend и backend особенностей проектирования.

Frontend архитектурно, представляет проект на основе Feature-Sliced, содержащий:

- «Человек понятный» URL, с учетом защиты от XSS (далее ЧПУ);
- Общие компоненты, не содержащие бизнес-логики (src/components) (UI component);
- Система отображения контента форм (Look);
- Маппинг объектов для отправки и при получении (MVVM) при общении с БЭКендом.

Модульность. В зависимости от выбранного URL, выполняются разные модули для отображения (при наличии прав доступа). Все запросы на Backend, выполняются только с JWT-ключом, генерация и хранение которого, выполняется на Backend. В случае, если JWT-ключ не валиден, БЭКенд отвечает ошибкой авторизации (HTTP status 401 Unauthorized). Дальнейшее выполнение любых действий, кроме авторизации невозможно и пользователю необходимо авторизоваться.

Хранение данных на стороне браузера отсутствует. Все исходные тексты и файлы хранятся на стороне сервера.

Система построена модульно. Модуль может быть использован из любого места кода.

Валидация данных описана непосредственно в каждом модуле (при необходимости).

Модули имеют слабую связность и строгую типизацию. Общие компоненты, не содержащие бизнес-логики, могут быть использованы в любом месте.

Требования к браузеру:

- Chrome (актуальная версия);
- Firefox (актуальная версия);
- MS Edge (актуальная версия).

- Все используемые библиотеки бесплатные для использования и имеют лицензии MIT и/или GPL3.

Backend архитектурно, представляет web-ресурс, предоставляющий различного рода REST API ресурсы, выполненный в монолитной архитектуре.