ВЕБ-СЕРВИС ОБЛАЧНОГО ХРАНЕНИЯ ДАННЫХ С ПРОГРАММНЫМ МОДУЛЕМ ДЛЯ ИНТЕГРАЦИИ СО СТОРОННИМИ СИСТЕМАМИ С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИЙ OAUTH 2.0 И OPENID CONNECT

Буйновский М.В., студент гр. 281073

Белорусский государственный университет информатики и радиоэлектроники,
Институт информационных технологий,
г. Минск, Республика Беларусь

Середа И.А. – маг. техн. наук, старший преподаватель

Разработан веб-сервис облачного хранения данных с федеративной аутентификацией на основе OAuth 2.0 и OpenID Connect. Система включает сервис аутентификации и авторизации, файловый сервис, JavaScript-модуль для интеграции и приложение-проводник. Решение обеспечивает безопасное хранение данных, упрощенную интеграцию со сторонними системами и централизованное управление доступом.

Современная цифровая экономика характеризуется экспоненциальным ростом объемов данных и усложнением информационных систем. По данным аналитиков, к 2025 году глобальный объем данных превысит 180 зеттабайт, что создает необходимость в эффективных, безопасных и масштабируемых решениях для хранения, обмена и управления файлами [1]. Облачные технологии стали неотъемлемой частью IT-инфраструктуры организаций, при этом вопросы федеративной аутентификации и безопасной интеграции между различными системами приобретают критическое значение для защиты конфиденциальных данных.

Основная проблема, решаемая в рамках дипломного проекта, заключается в отсутствии универсальных, безопасных и легко интегрируемых решений для облачного хранения данных с поддержкой федеративной аутентификации. Традиционные системы хранения файлов страдают от следующих недостатков:

– сложность интеграции со сторонними приложениями: разработчикам приходится самостоятельно реализовывать потоки авторизации, что увеличивает время разработки и вероятность ошибок;

– отсутствие единого управления доступом в распределенных системах, где несколько серверов или приложений нуждаются в централизованной аутентификации;

– невозможность использования собственных серверов для хранения данных, что создает зависимость от внешних облачных провайдеров и риски утечек [2].

Разработанный веб-сервис решает выявленные проблемы путем создания системы с микросервисной архитектурой, где аутентификация и авторизация вынесены в отдельный сервер. Решение обеспечивает:

– централизованное управление доступом на основе протоколов OAuth 2.0 и OpenID Connect;

– федеративную модель, позволяющую интегрировать несколько файловых хранилищ под единым контролем;

– упрощенную интеграцию для сторонних разработчиков через внешний JavaScript-модуль, автоматизирующий OAuth-потоки [3].

Технологический стек разработанного веб-сервиса включает:

– бэкенд: PHP-фреймворк Symfony 6.4 с использованием Doctrine ORM, обеспечивающий модульную структуру и встроенные механизмы безопасности;

– фронтенд: React 19 с TypeScript, TailwindCSS для стилизации и компонентной библиотекой shadcn/ui;

– базы данных: PostgreSQL для сервера аутентификации (пользователи, серверы, токены, модули), SQLite для файлового сервера (индекс файлов и сессии);

– инфраструктура: Docker для контейнеризации, Kubernetes для оркестрации, Nginx как обратный прокси-сервер [4].

Функциональные возможности системы разделены на три основных компонента:

1. Сервис аутентификации и авторизации – обеспечивает регистрацию пользователей, двухфакторную аутентификацию, генерацию и валидацию JWT токенов, управление OAuth-клиентами (серверами и модулями).

2. Файловый сервис – реализует хранение файлов, индексацию метаданных, проверку прав доступа, загрузку и скачивание файлов, создание папок.

3. JavaScript-модуль интеграции – предоставляет SDK для сторонних систем, автоматизирующий процессы инициализации, аутентификации, обмена токенов и выполнения операций с файлами.

На рисунке 1 представлен диалог создания сервера в панели администратора, демонстрирующий процесс регистрации нового файлового сервера как OAuth-клиента с указанием названия, изображения и разрешенных URI перенаправления.

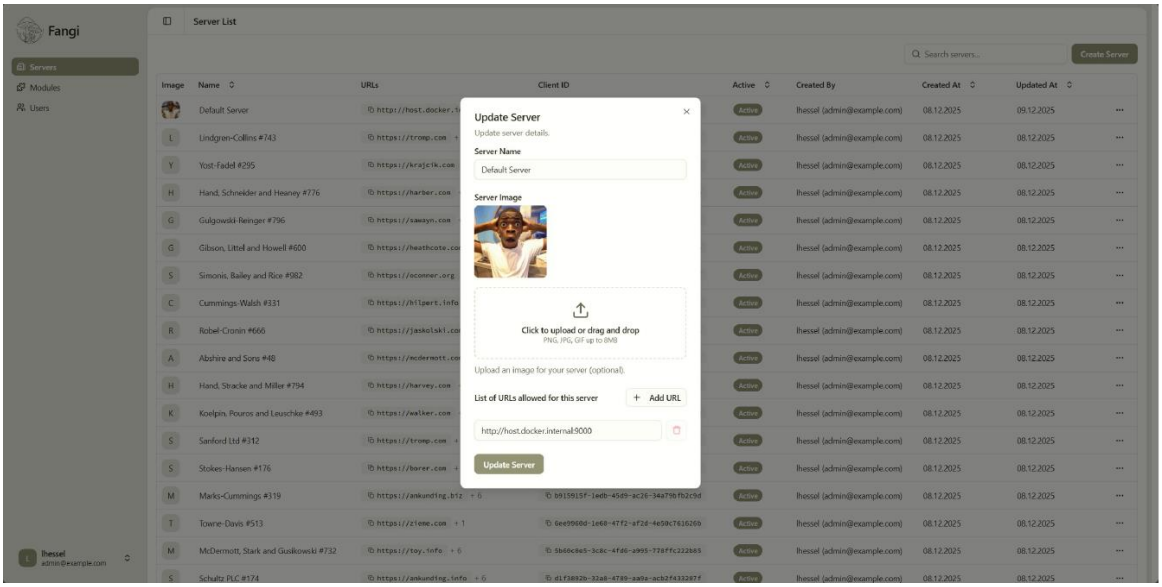

Рисунок 1 – Диалог создания сервера

Пользовательский интерфейс разработан с учетом двух различных дизайн-концепций:

– сервис аутентификации выполнен в пастельных бежевых и приглушенных зеленых оттенках, формирующих умиротворяющую среду, транслирующую надежность и укрепляющую доверие к безопасности системы;

– приложение-проводник выполнено в эстетике ретро-японской промышленной электроники конца 1980-х – начала 1990-х годов, сочетая функциональность с тактильной цифровой графикой.

Для управления процессом разработки применяется распределенная система контроля версий Git с адаптированной моделью GitFlow, включающей ветки:

– master – стабильный код для производственной среды;

– develop – основная ветка для разработки;

– feature/* – ветки для реализации новых функций;

– hotfix/* – ветки для оперативного устранения критических ошибок;

– release/* – ветки для подготовки релизов.

Тестирование программного средства включало:

– функциональное тестирование сценариев регистрации, аутентификации (включая двухфакторную), авторизации модулей и файловых операций;

– модульное тестирование Backend компонентов (генерация токенов, валидация кодов авторизации, проверка доступа);

– интеграционное тестирование взаимодействия между сервисом аутентификации, файловым сервисом и модулем;

– нагрузочное тестирование, подтвердившее стабильную работу при множественных подключениях с минимальным временем отклика.

Разработанный веб-сервис облачного хранения данных представляет собой готовое к внедрению решение, обеспечивающее:

– безопасное хранение данных за счет серверного шифрования и использования JWT токенов;

– централизованное управление доступом через федеративную аутентификацию на основе OAuth 2.0 и OpenID Connect;

– упрощенную интеграцию для сторонних систем через JavaScript-модуль-посредник;

– масштабируемость и надежность благодаря микросервисной архитектуре и контейнеризации в Docker и Kubernetes.

Применение открытых стандартов OAuth 2.0 и OpenID Connect позволяет делегировать доступ без передачи учетных данных, минимизируя риски фишинга и утечек. Результаты технико-экономического обоснования подтверждают экономическую целесообразность разработки и внедрения.

Список использованных источников:

1. Amount of Data Created Daily (2025) [Электронный ресурс]. – Режим доступа : https://explodingtopics.com/blog/data-generated-per-day.

2. Forecast: Public Cloud Services, Worldwide, 2023-2029, 1Q25 Update [Электронный ресурс]. – Режим доступа : https://www.gartner.com/en/documents/6302015.

3. The OAuth 2.0 Authorization Framework – RFC-6749 : Internet Engineering Task Force (IETF) – D. Hardt, Ed, 2012. – 75 с.

4. OpenID Connect Core 1.0 incorporating errata set 2 : OpenID Foundation – N. Sakimura, J. Bradley, M. Jones, B. de Medeiros, C. Mortimore, 2023. – 84 с.