



Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.03 Прикладная информатика

О Т Ч Е Т

по индивидуальному заданию

Название: Разработка фронтенда для облачного хранилища

Дисциплина: Проектирование информационных систем

Студент

ИУ6-75Б

(Группа)

(Подпись, дата)

А. А. Оглоблин

(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А. Ф. Прутик

(И.О. Фамилия)

Москва, 2025

Цель работы:

- Создать фронтенд для облачного хранилища VOblako, работающий в режиме BFF (Next.js App Router + API routes);
- Обеспечить аутентификацию, просмотр файлов/папок, загрузку и управление файлами, в том числе в mock-режиме без внешнего бэкенда;
- Сформировать удобную UI-оболочку с адаптивным layout и интегрировать BFF-ручки с серверной частью через прокси.

Требования к системе:

- Реализовать BFF-слой на Next.js;
- Рендерит защищённые страницы (SSR) и обслуживает клиентские запросы;
- Проксирует запросы к внешнему API Gateway либо использует локальные моки;
- Управляет сессией через HTTP-only cookies;
- Покрывает API-маршруты и UI-компоненты тестами (Vitest + Testing Library);
- Обеспечить возможность контейнерной сборки и работы mock-данных внутри образа.

1 Описание решения и используемые технологии:

Next.js 16 (App Router) выступает как BFF: серверный рендеринг и API-handlers в `app/api/*`.

Режимы работы:

- `VOBLAKO_API_TARGET=remote` — прокси в удалённый API Gateway через `forwardToRemoteApi`;
- `VOBLAKO_API_TARGET=mock` — обработка файлов/хранилища/авторизации в памяти с начальным seed'ом и мок-файлами, включёнными в Docker-образ.

UI: адаптивный protected layout, интерактивная сетка файлов, drag&drop загрузка, диалоги загрузки и предпросмотра.

Тесты: покрыты API (auth/files/storage), layout-компоненты и загрузочные виджеты; добавлена команда `npm run test:coverage`.

2 Архитектурные схемы

На рисунке 2 отображена диаграмма компонентов микросервиса, показывающая внутреннюю структуру и связи между модулями. На рисунке 3 приведена диаграмма классов (структура кода) с основными модулями и их ответственностью.

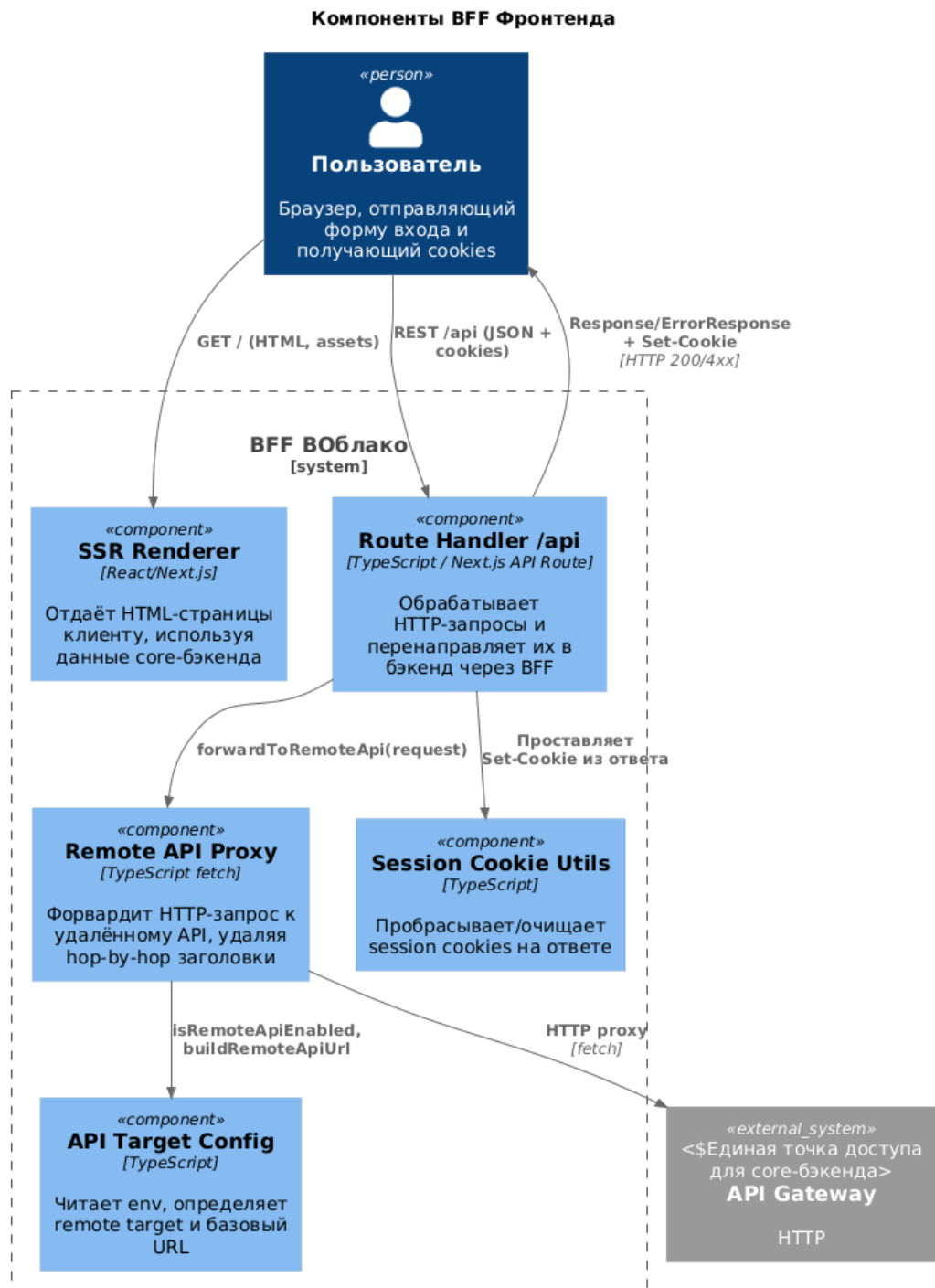


Рисунок 3 – Диаграмма компонентов

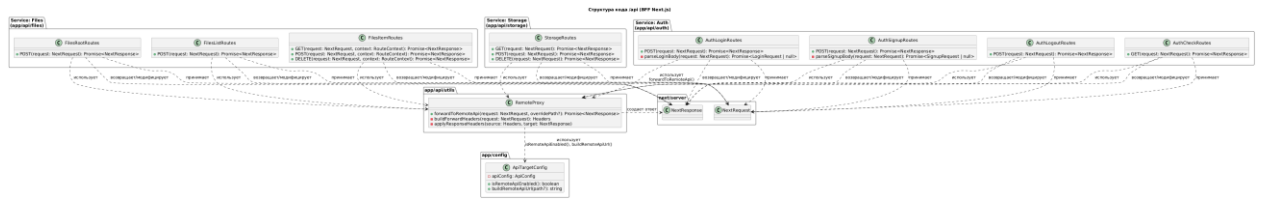


Рисунок 4 – Диаграмма кода

3 Тестирование и документация

Разработаны интеграционные тесты с использованием Vitest и Testing Library, были покрыты API-маршруты и UI-компоненты.

Вывод: Реализован BFF на Next.js с переключаемыми режимами mock/remote и корректной работой cookies. Макеты и загрузочные сценарии протестированы, мок-файлы включены в Docker-образ. Добавлены команды тестов и покрытия, обеспечивающие базовую регрессионную гарантию для API и UI слоёв.