

Сравнительный анализ архитектурных паттернов микрофронтендов для высоконагруженных веб-платформ

В.В. Артюхин, Д.В. Батулин, А.И. Егунова, В.Е. Аксенов

ФГБОУ ВО «Национальный исследовательский Мордовский государственный университет им. Н.П. Огарева», г. Саранск

Аннотация: Статья представляет исследование различных подходов к реализации микрофронтенд-архитектуры в высоконагруженных веб-приложениях. Описывается процесс сравнительного анализа четырех основных паттернов интеграции микрофронтендов: Module Federation, Single-SPA, Web Components и iframe-подхода. Проведено экспериментальное исследование производительности каждого паттерна с измерением ключевых метрик загрузки и взаимодействия. Разработаны рекомендации по выбору архитектурного подхода в зависимости от требований проекта.

Ключевые слова: микрофронтенды, веб-архитектура, Module Federation, Single-SPA, высоконагруженные системы, производительность веб-приложений.

Введение

Современные веб-платформы характеризуются растущей сложностью функциональности и необходимостью независимой разработки компонентов различными командами. Традиционная монолитная фронтенд-архитектура создает проблемы масштабирования и увеличивает риски при обновлении отдельных модулей системы [1].

Микрофронтенд-архитектура переносит концепции микросервисов на уровень пользовательского интерфейса [2], позволяя разделить приложение на независимые модули, каждый из которых может разрабатываться, тестироваться и развертываться автономно. В 2025 году микрофронтенды стали одним из ключевых трендов веб-разработки, особенно для крупных корпоративных систем и высоконагруженных платформ [3].

Актуальность исследования обусловлена отсутствием четких критериев выбора конкретного паттерна микрофронтендов в зависимости от характеристик проекта. Существующие подходы имеют различные компромиссы между производительностью, сложностью реализации,

изоляции и гибкостью, что затрудняет обоснованный выбор архитектурного решения.

Обзор архитектурных паттернов микрофронтендов

В настоящее время выделяют четыре основных подхода к реализации микрофронтенд-архитектуры, каждый из которых имеет свои области применения.

Module Federation – подход, встроенный в Webpack 5, позволяющий динамически загружать модули из различных сборок в runtime. Данный подход обеспечивает эффективное переиспользование зависимостей между микрофронтендами и минимизацию дублирования кода.

Single-SPA – фреймворк-оркестратор, предоставляющий механизм регистрации и монтирования независимых приложений в рамках одной страницы. Single-SPA поддерживает различные фреймворки (React, Vue, Angular) и управляет их жизненным циклом [4].

Web Components – нативный браузерный стандарт для создания изолированных компонентов с инкапсуляцией стилей и логики через Shadow DOM. Подход обеспечивает максимальную изоляцию, но имеет ограничения в производительности и сложность интеграции с современными фреймворками [5].

Iframe-подход – классический метод изоляции, при котором каждый микрофронтенд загружается в отдельный iframe. Обеспечивает полную изоляцию, но создает проблемы с SEO и пользовательским опытом.

Методика экспериментального исследования

Для сравнительного анализа был разработан тестовый стенд, состоящий из контейнерного приложения (shell) и четырех микрофронтенд-модулей. Каждый модуль реализован идентично на React 18, но

интегрируется с использованием различных паттернов. Основные характеристики тестовой системы

- Контейнер: React 18 + TypeScript, роутинг React Router
- Микрофронтенды: 4 модуля по 120KB [6]
- Сервер: Node.js 20, Nginx в качестве прокси
- Тестовое окружение: Chrome 130, MacBook Pro M3, сеть 4G

Реализация паттернов микрофронтендов. Для обеспечения объективности сравнения все паттерны реализованы с использованием идентичной функциональности. Ниже представлен пример конфигурации Module Federation для одного из микрофронтендов на рисунке 1.

```
const ModuleFederationPlugin = require("webpack/lib/container/ModuleFederationPlugin")

module.exports = {
  plugins: [
    new ModuleFederationPlugin({
      name: "eventModule",
      filename: "remoteEntry.js",
      exposes: {
        "./EventList": "./src/components/EventList",
        "./EventDetail": "./src/components/EventDetail"
      },
      shared: {
        react: { singleton: true, requiredVersion: "^18.0.0" },
        "react-dom": { singleton: true, requiredVersion: "^18.0.0" },
        "react-router-dom": { singleton: true }
      }
    })
  ]
};

// Подключение в контейнере
const EventModule = React.lazy(() => import("eventModule/EventList"));

function App() {
  return (
    <Suspense fallback={<Loader />}>
      <EventModule />
    </Suspense>
  );
}
```

Рис. 1 – Конфигурация Module Federation для микрофронтенда

Для Single-SPA реализация основывалась на регистрации приложений через функции `registerApplication` и управления активностью через маршрутизацию. Web Components использовали Custom Elements API с

Shadow DOM для инкапсуляции стилей [7]. Iframe-подход реализован через динамическое создание элементов с postMessage для коммуникации

Результаты экспериментального исследования

Для оценки производительности измерялись следующие метрики согласно методологии Web Vitals:

- Time to Interactive (TTI) – время до интерактивности первого микрофронтенда
- Total Bundle Size – суммарный размер загружаемых файлов
- Navigation Time – время переключения между микрофронтендами
- Memory Overhead – потребление оперативной памяти
- Initial Load Time – время первичной загрузки всего приложения

Данные представлены в таблице № 1

Таблица № 1

Сравнительные характеристики паттернов микрофронтендов

| Метрика | Module Federation | Single-SPA | Web Components | Iframe |
|--------------------|-------------------|------------|----------------|--------|
| TTI (сек) | 1.8 | 2.1 | 2.4 | 3.2 |
| Bundle Size (KB) | 245 | 312 | 298 | 380 |
| Navigation (мс) | 120 | 180 | 210 | 450 |
| Memory (MB) | 45 | 52 | 48 | 78 |
| Initial Load (сек) | 2.2 | 2.6 | 2.8 | 3.9 |

Анализ результатов показывает, что Module Federation обеспечивает наилучшие показатели производительности благодаря эффективному механизму sharing зависимостей [8]. Single-SPA демонстрирует сбалансированные характеристики с небольшим overhead на оркестрацию.

Web Components показывают хорошую изоляцию, но страдают от увеличенного Bundle Size из-за полифилов [9].

Критерии выбора архитектурного паттерна

На основе проведенного исследования разработана матрица решений для выбора оптимального паттерна микрофронтендов в зависимости от требований проекта:

Module Federation рекомендуется для:

- Проектов с использованием Webpack 5 и современных фреймворков
- Систем, требующих оптимальной производительности и минимального Bundle Size

Single-SPA подходит для:

- Миграции legacy-приложений на микрофронтенд-архитектуру
- Проектов с гетерогенными технологическими стеками (React + Angular)

Web Components оптимальны для:

- Дизайн-систем и переиспользуемых UI-библиотек [10]
- Проектов с требованиями максимальной изоляции стилей и логики

Заключение

Проведенное исследование показало, что выбор паттерна микрофронтенд-архитектуры существенно влияет на производительность и поддерживаемость веб-приложения. Module Federation демонстрирует наилучший баланс характеристик для большинства современных проектов.

Разработанная матрица решений позволяет архитекторам и техническим лидерам обоснованно выбирать архитектурный подход с учетом специфики проекта. Перспективы исследования включают анализ гибридных подходов и исследование паттернов в контексте микрофронтендов.

Литература

1. Берьянов М.С., Монченко А.С., Дерябин А. Исследование микрофронтендной архитектуры - инструменты и рекомендованные практики использования // Столыпинский вестник. 2022. № 10. С. 5575-5585.
2. Peltonen S., Mezzalana L., Taibi D. Motivations, Benefits, and Issues for Adopting Micro-Frontends: A Multivocal Literature Review // Information and Software Technology. 2021. V. 136. Article 106571.
3. Lawson N. Does shadow DOM improve style performance? 2021. URL: nolanlawson.com/2021/08/15/does-shadow-dom-improv-style-performance.
4. Павлидис В.Д., Бражникова А.Н. Современные архитектуры построения web-приложений // Наукосфера. 2025. № 3-2. С. 47-52.
5. Егорчев Ф.А., Лукьянов С.К., Замотайлова Д.А. Архитектура веб-приложений // Информационное общество: современное состояние и перспективы развития: Кубанский государственный аграрный университет имени И.Т. Трубилина, 2021. С. 189-193.
6. Короленко И.А. О фреймворках Vue, Svelte, Solid и Lit для разработки клиентских веб-приложений // Инженерный вестник Дона. 2023. № 11. URL: ivdon.ru/ru/magazine/archive/n11y2023/8804.
7. Яровая Е.В. Микросервисная архитектура при разработке фронтенд - приложений // Столыпинский вестник. 2022. Т. 4. № 5. С. 19.
8. Мишин П.А., Мишина П.А. Разработка микросервиса ретайлирования на языке Python // Инженерный вестник Дона. 2024. № 1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8928
9. Артюхин В.В., Батурин Д.В., Егунова А.И., Кочеваткин А.В. Алгоритм адаптивного кэширования данных в распределённых веб-системах // Научно-технический вестник Поволжья. 2025. № 10. С. 75-77.

10. Филиппев А.В., Махалкина Т.О., Дмитриев Д.С. Оркестрация контейнеров микросервисной архитектуры в системе распознавания лиц // Системный анализ в науке и образовании. 2020. № 2. С. 108-114.

References

1. Ber'yanov M.S., Monchenko A.S., Deryabin A. Stolypinskij vestnik. 2022. № 10. Pp. 5575-5585.
2. Peltonen S., Mezzalana L., Taibi D. Information and Software Technology. 2021. V. 136. Article 106571.
3. Pavlidis V.D., Brazhnikova A.N. Naukosfera. 2025. № 3-2. Pp. 47-52.
4. Lawson N. Does shadow DOM improve style performance? 2021. URL: nolanlawson.com/2021/08/15/does-shadow-dom-improv-style-performance
5. Egorchev F.A., Luk'yanov S.K., Zamotajlova D.A. Informacionnoe obshchestvo: sovremennoe sostoyanie i perspektivy razvitiya: Sbornik materialov XIV mezhdunarodnogo foruma. Krasnodar: Kubanskij gosudarstvennyj agrarnyj universitet imeni I.T. Trubilina, 2021. Pp. 189-193
6. Korolenko I.A. Inzhenernyj vestnik Dona. 2023. № 11. URL: ivdon.ru/ru/magazine/archive/n11y2023/8804
7. Yarovaya E.V. Mikroservisnaya arhitektura pri razrabotke frontend prilozhenij, Stolypinskij vestnik. 2022. V. 4. № 5. P. 19.
8. Mishin P.A., Mishina P.A. Inzhenernyj vestnik Dona. 2024. № 1. URL: ivdon.ru/ru/magazine/archive/n1y2024/8928
9. Artyukhin V.V., Baturin D.V., Egunova A.I., Kochevatkin A.V. Nauchno-tekhnicheskij vestnik Povolzh'ya. 2025. № 10. Pp. 75-77.
10. Filip'ev A.V., Mahalkina T.O., Dmitriev D.S. Sistemnyj analiz v nauke i obrazovanii. 2020. № 2. Pp. 108-114.

Дата поступления: 5.11.2025

Дата публикации: 27.12.2025