УДК 004.42
Никитюк В., канд.техн.наук; Старицький О.
Тернопільський національний технічний університет імені Івана Пулюя

ОПТИМІЗАЦІЯ АДАПТИВНО-ГІБРИДНОЇ АРХІТЕКТУРИ ARX ДЛЯ ПІДВИЩЕННЯ МАСШТАБОВАНОСТІ ТА ПРОДУКТИВНОСТІ ВЕБКЛІЄНТІВ

Nykytiuk V., Starytskyi O.
Ternopil Ivan Puluj National Technical University

OPTIMIZATION OF THE ADAPTIVE-HYBRID ARX ARCHITECTURE FOR IMPROVING THE SCALABILITY AND PERFORMANCE OF WEB CLIENTS

Оптимізація архітектури фронтенд-застосунків відіграє важливу роль у забезпеченні їхньої масштабованості та продуктивності. В умовах зростання складності вебклієнтів і необхідності роботи в реальному часі традиційні підходи не завжди забезпечують достатню гнучкість та ефективність.

Метою роботи є дослідження адаптивно-гібридної архітектури ARX та аналіз можливостей її оптимізації для підвищення продуктивності й масштабованості вебклієнтів. Особливу увагу приділено управлінню станом, розподілу логіки між клієнтом і сервером та адаптації архітектури до змін навантаження.

Для досягнення мети проаналізовано існуючі архітектурні підходи у фронтенд-розробці та визначено їхні переваги й недоліки. Результати порівняння монолітних, серверних і мікрофронтенд-рішень наведено в таблиці 1, що дозволило виявити обмеження традиційних підходів і обґрунтувати доцільність використання гібридної моделі.

Таблиця 1. Порівняння архітектур фронтенд-застосунків

Напрямок оптимізації	Опис	Вплив на систему
Lazy loading	Завантаження по мірі потреби	↓ час завантаження
Code splitting	Розділення bundle	↓ навантаження
SSR/CSR hybrid	Комбінування рендерингу	↑ SEO + швидкість
Caching	Кешування даних	↑ продуктивність
State optimization1	Оптимізація стану	↑ масштабованість

Провівши порівняння та ознайомившись із функціоналом кожного з рендерів, обрано SSR, оскільки цей підхід дозволяє ефективно вирішувати задачі SEO та працювати з динамічним контентом. Він забезпечує генерацію готового HTML на

сервері для кожного запиту, що покращує індексацію сторінок і створює кращий користувацький досвід завдяки швидкому завантаженню основного контенту. SSR також надає гнучкість у реалізації, що робить його оптимальним вибором для проєктів із динамічними даними.

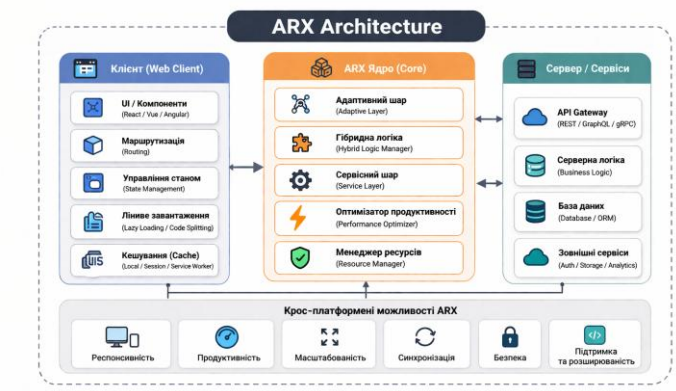

Рисунок 1. Архітектура ARX

Висновок: досліджено наявні підходи до рендеру та обрано оптимальний варіант для нашої роботи. На основі вибраної стратегії було обрано основний напрямок для оптимізації. Обраний рендер допоможе нам покращити динамічність та SEO та автоматизувати роботу в певних аспектах коду.

Література

1. Micro Frontends Architecture: Scalable Frontend Development. // Режим доступу: https://micro-frontends.org/. [дата звернення 12.11.2024].
2. Performance Optimization in React Applications. // Режим доступу: https://react.dev/learn/render-and-commit. [дата звернення 12.11.2024].
3. Next.js Rendering Strategies (SSR, SSG, ISR). // Режим доступу: https://nextjs.org/docs/pages/building-your-application/rendering. [дата звернення 12.11.2024].
4. Web Application Architecture Patterns. // Режим доступу: https://learn.microsoft.com/en-us/azure/architecture/guide/architecture-styles/web-app. [дата звернення 12.11.2024].
5. Frontend Performance Best Practices. // Режим доступу: https://web.dev/performance/ . [дата звернення 12.11.2024].