УДК 004.735

Никитин Михаил Олегович, студент, Казанский государственный энергетический университет, г. Казань, Россия.

Сандаков Виталий Дмитриевич, кандидат технических наук, доцент, Казанский государственный энергетический университет, г. Казань, Россия.

ВЕБ-ПРИЛОЖЕНИЯ В УСЛОВИЯХ ЦИФРОВИЗАЦИИ: АРХИТЕКТУРА, КЛИЕНТСКИЕ ТЕХНОЛОГИИ, СЕТЕВЫЕ ОСНОВЫ И БЕЗОПАСНОСТЬ

Аннотация

В статье рассматриваются веб-приложения как основная форма предоставления цифровых сервисов. Показано, что качество веб-системы определяется архитектурой, клиентскими технологиями и сетевыми условиями, а также требованиями информационной безопасности. Кратко охарактеризованы подходы MPA, SPA и SSR/гибридные решения и их влияние на производительность и сопровождение. Отмечена роль HTML, CSS, JavaScript и DOM в формировании интерактивного интерфейса и возникновении типовых проблем быстродействия. Обобщены причины задержек при клиент–серверном взаимодействии и базовые методы их снижения. Рассмотрены распространённые угрозы (инъекции, XSS, CSRF) и меры защиты, закладываемые на этапе проектирования. Для наглядности приведены таблица сравнения подходов и схема контрольных точек безопасности.

Annotation

The article considers web applications as a primary form of delivering digital services. It shows that the quality of a web system is determined by its architecture, client technologies and network conditions, as well as information security requirements. The MPA, SPA and SSR/hybrid approaches are briefly described and their impact on performance and maintenance is discussed. The role of HTML, CSS, JavaScript and the DOM in building an interactive interface and causing typical

performance issues is highlighted. Common causes of latency in client–server interaction and basic methods to reduce it are summarized. Typical threats (injection, XSS, CSRF) and security measures that should be incorporated at the design stage are reviewed. For clarity, a comparison table and a diagram of key security control points are provided.

Ключевые слова: веб-приложения, архитектура, HTML, CSS, JavaScript, клиент–сервер, сети, безопасность.

Keywords: web applications, architecture, HTML, CSS, JavaScript, client–server, networks, security.

Развитие цифровых технологий привело к тому, что веб-приложения стали основным способом предоставления услуг: пользователю не требуется установка программ, а доступ к функционалу осуществляется через браузер. Это снижает “порог входа”, ускоряет внедрение обновлений и упрощает поддержку, что особенно важно в условиях распределённой работы и высокой динамики требований. Вместе с тем переход к веб-формату повышает требования к проектированию: ошибки в архитектуре, клиентской части или безопасности быстрее приводят к сбоям и рискам, так как приложение доступно широкому кругу пользователей [2; 6].

Традиционно веб-приложение строится на модели “клиент–сервер”: браузер отправляет запросы, сервер обрабатывает их и возвращает ответ в виде HTML-страницы, данных (например, JSON) или файлов. На практике качество взаимодействия зависит от целой цепочки факторов: от оптимальности серверной логики и структуры данных до особенностей клиентского кода, который управляет интерфейсом и формирует дополнительные запросы [5]. Поэтому проектирование веб-приложений целесообразно рассматривать как комплексный процесс, в котором архитектура, технологии фронтенда, сетевые параметры и защита взаимосвязаны [2].

Наиболее распространены три подхода к построению веб-приложений. Первый — многостраничные приложения (MPA), где сервер формирует

страницы, а браузер получает готовый HTML. Второй — одностраничные приложения (SPA), где интерфейс и логика в большей степени “переезжают” в браузер, а сервер предоставляет API. Третий — гибридные решения (SSR/частичная серверная генерация), объединяющие быструю выдачу начального контента и богатую интерактивность [2; 3]. Выбор подхода влияет на требования к инфраструктуре, объём клиентского кода и площадь атаки с точки зрения безопасности.

Для наглядности сравнение подходов приведено в таблице 1.

Таблица 1 — Сравнение архитектурных подходов веб-приложений

Критерий сравнения	MPA	SPA	SSR
Формирование интерфейса	в основном на сервере	В основном в браузере	Совместно (клиент + сервер)
Интерактивность	средняя	высокая	высокая
Нагрузка на клиент	ниже	выше	средняя
Требования к API	умеренные	высокие	высокие
Сопровождение	проще	сложнее (сборка, маршрутизация, состояние)	сложнее (два контура рендеринга)
Риски безопасности	классические для форм/шаблонов	классические для форм/шаблонов	комбинированные

Клиентская часть веб-приложений опирается на HTML и CSS как основу структуры и оформления, а также на JavaScript и DOM для интерактивности: обработки событий, динамического изменения страницы, валидации данных и взаимодействия с сервером [4]. При этом именно клиентская логика часто становится источником проблем: большое число скриптов, частые изменения DOM, неоптимальные обработчики событий и избыточные запросы приводят к снижению отзывчивости интерфейса. Поэтому при проектировании важно учитывать размер и сложность клиентского кода, а также правила его

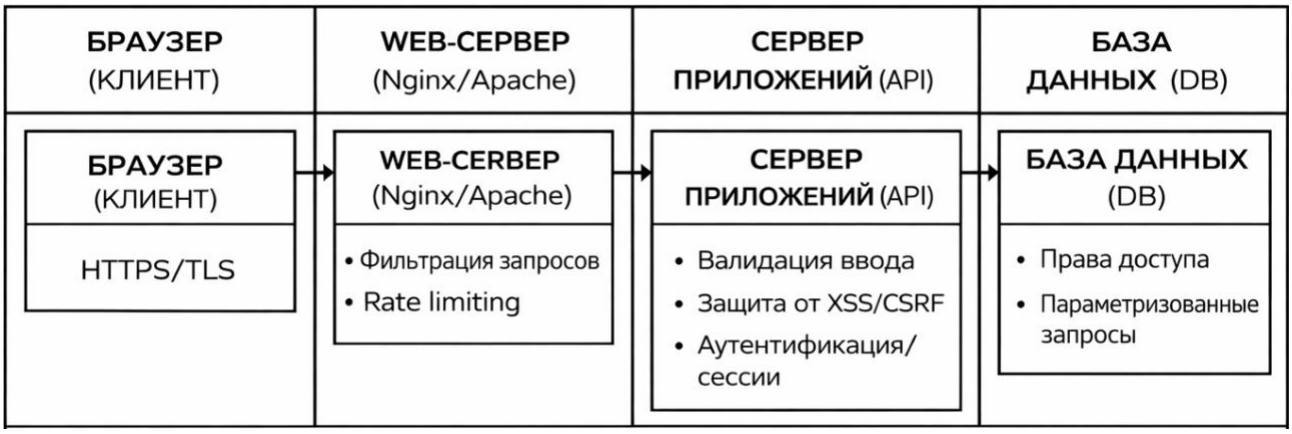
организации (модульность, повторное использование, ограничение лишних перерисовок) [1; 5].

Сетевой уровень также заметно влияет на поведение веб-приложения. Даже при корректной архитектуре задержки могут возникать из-за особенностей передачи данных: установления соединения, перегрузок каналов, потерь пакетов и конкуренции запросов. С точки зрения инженерной практики полезно оптимизировать количество запросов, использовать кэширование статических ресурсов и избегать “чата” (слишком частых обращений к серверу мелкими порциями), объединяя запросы там, где это оправдано [6; 7]. Для высоконагруженных систем это становится обязательным условием стабильной работы.

Отдельного внимания требуют вопросы информационной безопасности. Поскольку веб-приложения доступны через сеть, они регулярно становятся объектом атак. Наиболее типичны инъекции (в том числе при работе с базой данных), межсайтовый скриптинг (XSS) и межсайтовая подделка запросов (CSRF). Существенная часть рисков снижается ещё на этапе проектирования: строгая валидация входных данных, корректная работа с правами доступа, безопасное управление сессиями, ограничение привилегий и контроль источников данных [8]. Важно, что безопасность — это не “проверка в конце”, а набор решений, встроенных в архитектуру.

Наглядное представление точек контроля безопасности в типовой архитектуре веб-приложения приведено на рисунке 1.

Рисунок 1 — Типовая архитектура веб-приложения и основные точки обеспечения безопасности



Таким образом, разработка веб-приложений в условиях цифровизации требует комплексного подхода. Архитектура определяет масштабируемость и удобство сопровождения, клиентские технологии — качество интерфейса и интерактивность, сетевые факторы — реальную скорость отклика, а безопасность — устойчивость системы к угрозам. Совмещение этих аспектов на этапе проектирования позволяет создать веб-приложение, которое будет не только функциональным, но и надёжным, удобным и безопасным для пользователя [2; 6; 8].

Литература

1. Заяц А. М., Васильев Н. П. Проектирование и разработка WEB-приложений. Введение в frontend и backend разработку на JavaScript и node.js : учебное пособие для вузов. — 3-е изд., стер. — СПб. : Лань, 2021. — 120 с.
2. Тузовский А. Ф. Проектирование и разработка web-приложений : учебное пособие. — Томск : ТПУ, 2014. — 219 с.
3. Полуэктова Н. Р. Разработка веб-приложений : учебник для среднего профессионального образования. — 2-е изд. — М. : Издательство Юрайт, 2025. — 204 с.
4. Морозов Д. С., Кузнецова И. А. Веб-технологии для начинающих: HTML, CSS, JavaScript : учебное пособие. — М. : Техносфера, 2019. — 450 с.
5. Кузнецов М. И. Введение в веб-программирование: HTML, CSS, JavaScript : учебное пособие. — СПб. : Питер, 2022. — 532 с.

6. Наговицын И. В. Основы веб-технологий : учебник для вузов. — М. : ДМК Пресс, 2023. — 600 с.
7. Олифер В. Г., Олифер Н. А. Компьютерные сети. Принципы, технологии, протоколы : учебник для вузов. — СПб. : Питер, 2021. — 1008 с.
8. Хоффман Э. Безопасность веб-приложений : пер. с англ. — СПб. : Питер, 2021. — 336 с.

Literature

1. Zayats A. M., Vasilyev N. P. Design and Development of WEB Applications. An Introduction to Frontend and Backend Development in JavaScript and Node.js: study guide for universities. — 3rd ed., stereotyped. — Saint Petersburg: Lan', 2021. — 120 p.
2. Tuzovskiy A. F. Design and Development of Web Applications: study guide. — Tomsk: TPU Publishing House, 2014. — 219 p.
3. Poluektova N. R. Web Application Development: textbook for secondary vocational education. — 2nd ed. — Moscow: Yurait Publishing House, 2025. — 204 p.
4. Morozov D. S., Kuznetsova I. A. Web Technologies for Beginners: HTML, CSS, JavaScript: study guide. — Moscow: Tekhnosfera, 2019. — 450 p.
5. Kuznetsov M. I. Introduction to Web Programming: HTML, CSS, JavaScript: study guide. — Saint Petersburg: Piter, 2022. — 532 p.
6. Nagovitsyn I. V. Fundamentals of Web Technologies: university textbook. — Moscow: DMK Press, 2023. — 600 p.
7. Olifer V. G., Olifer N. A. Computer Networks: Principles, Technologies, Protocols: university textbook. — Saint Petersburg: Piter, 2021. — 1008 p.
8. Hoffman A. Web Application Security: translated from English. — Saint Petersburg: Piter, 2021. — 336 p.