

Frontends for Business at Scale

Manoj Kumar Dobbala

Abstract: *In the rapidly evolving landscape of e-commerce and retail, the necessity to re-platform the frontend layer has become paramount [1]. This study aims to evaluate frameworks and methodologies applicable to the major players in this industry, currently burdened by disparate websites, siloed codebases, teams, and processes. This approach involves a comprehensive exploration of the historical context of frontend technology stacks, tracing their evolution and delving into the concept of micro frontends. Adaptability and ease of implementation of business use-cases is discussed. This report documents insights, providing valuable learnings for those involved in the evaluation and strategizing of re-platforming efforts. Notably, this study transcends industry boundaries, offering a business-agnostic perspective that can benefit any sector engaged in similar initiatives.*

Keywords: Frontend, Microservices, Micro frontends, Architecture, Scaling, Development

1. Introduction

In the dynamic realm e-commerce and retail, the frontend layer stands as a pivotal element in the digital landscape. The existing paradigm, marked by numerous websites dedicated to specific business areas, has resulted in a fragmented ecosystem with isolated codebases, teams, and processes. Recognizing the imperative need for a unified and streamlined approach, this study embarks on an exploration of frameworks [2] and methodologies aimed at re-platforming this critical layer.

a) Project Specification

The overarching project specification involves addressing the challenges posed by the current fragmented frontend architecture within the e-commerce and retail. This entails a meticulous evaluation of existing structures, proposing innovative solutions, and implementing a unified framework conducive to enhanced collaboration and efficiency.

b) Aim and Objectives

Aim:

The primary aim of this study is to re-platform the frontend layer by adopting modern frameworks and methodologies. This re-platforming is not a mere replacement but a strategic transformation to achieve a cohesive, business-aligned frontend architecture.

Objectives:

- **Evaluate Historical Evolution:** Explore the historical evolution of frontend technology stacks [3] to understand their progression and identify key milestones relevant to the e-commerce and retail.
- **Examine Micro Frontends:** Investigate the concept of micro frontends [7] as an innovative approach to overcoming the challenges posed by disparate systems. Assess its applicability and benefits within the context of the targeted industry.
- **Document Learnings:** Meticulously document the insights gained throughout the study, serving as a comprehensive resource for individuals involved in the

evaluation of re-platforming strategies within the e-commerce and retail.

c) Research Question:

How can the frontend layer in e-commerce and retail be strategically re-platformed to achieve cohesion and efficiency, considering the historical evolution of frontend technology stacks and the innovative concept of micro frontends?

d) Research Rationale:

This study's rationale lies in the recognition of the fragmented nature of the current frontend architecture within the targeted industry. By thoroughly evaluating historical trends and embracing innovative approaches like micro frontends, we aim to provide a roadmap for a unified, business-centric frontend layer that transcends industry boundaries, offering valuable insights for similar initiatives in diverse sectors.

2. Background

Evolution of Frontend Over the Years: Necessity for replatforming

The evolution of frontend development [4][12] has been a dynamic journey, mirroring the relentless pace of technological advancements and shifting user expectations. In the early days, websites were predominantly static, constructed with plain HTML, offering limited interactivity. However, as the internet matured, the demand for richer and more responsive user experiences prompted a paradigm shift.

The advent of JavaScript ushered in a new era, enabling dynamic web applications that responded to user interactions in real-time. Subsequently, the rise of Single Page Applications (SPAs)[5] and frontend frameworks like Angular, React, Vue.js and Svelte marked a pivotal moment. These frameworks provided developers with the tools to create complex, interactive interfaces, revolutionizing the way applications were built and experienced by users. Below is the graph (Fig 1) illustrating the popularity of these frameworks over the years, accompanied by Fig. 2, which presents a feature comparison of these frontend frameworks.

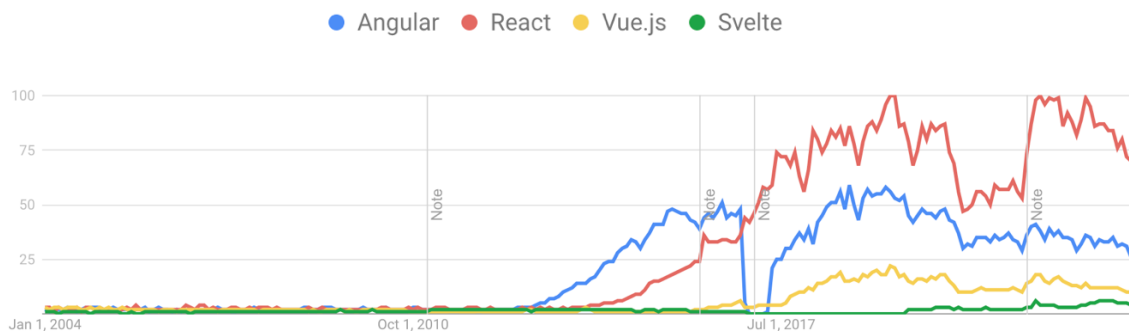


Figure 1: Source [6] Google trends: framework trends over years

Feature	Angular	Vue.js	React	Svelte
Data Binding	Two-way data binding	Two-way data binding	One-way data binding with hooks	Reactive assignment
Templating	HTML templates with Angular directives	HTML-based templates with directives	JSX syntax for rendering	HTML-like syntax with reactive assignments
Component Structure	Uses Components, Directives, and Services	Component-based	Component-based	Component-based
State Management	Built-in service for managing state (RxJS, NgRx)	Vuex for centralized state management	Context API, Redux, MobX	Context API, Redux, Recoil, Zustand
Tooling	Full-fledged CLI tool with extensive features	CLI tool with good features	Comprehensive tooling with Create React App	CLI tool with minimal setup
Performance Optimization	Ahead-of-Time Compilation (AoT)	Virtual DOM and optimized rendering	Virtual DOM and efficient diffing algorithm	Reactive updates and minimal runtime overhead
Framework Effectiveness	Comprehensive but can be heavy for smaller projects	Balanced performance and flexibility	High performance with a large ecosystem	High performance with minimal overhead
Applicability	Large-scale enterprise applications	Versatile for small to large-scale applications	Versatile for small to large-scale applications	Especially suited for smaller applications

Figure 2: Features comparison of frontend libraries/frameworks and applicability

As applications grew in complexity and scale, the need for replatforming became apparent. The traditional monolithic structures struggled to cope with the demands of modern business requirements, code maintainability, and the imperative for rapid deployments to meet speed-to-market goals. This necessitated a shift towards more modular and scalable architectures, prompting the emergence of micro frontends.

Adapting to Changing Trends: Micro Frontends [7] and Design Systems [8]

In response to the need for adaptability and agility, the concept of micro frontends gained prominence. Micro frontends involve breaking down the frontend of an application into smaller, independently deployable units. This approach aligns with the principles of microservices, allowing development teams to work on isolated components. The result is increased agility, scalability, and ease of maintenance, particularly in large-scale applications where different teams contribute to various parts of the frontend.

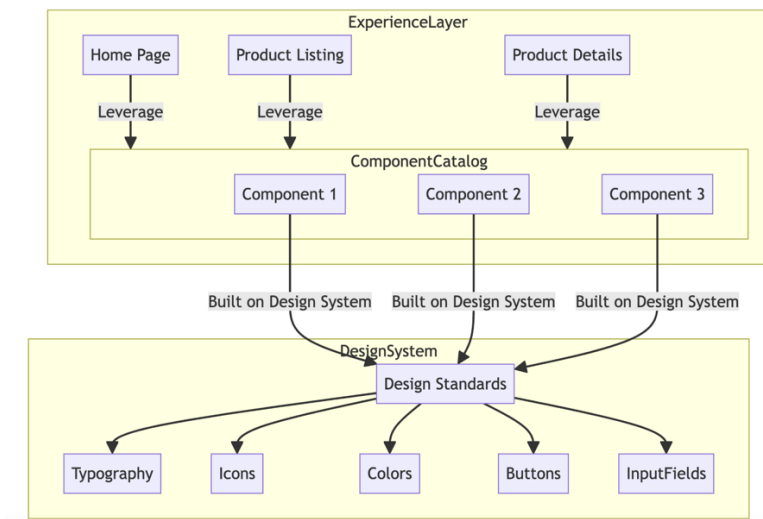


Figure 3: Micro-frontend with design system

Simultaneously, the concept of design systems emerged as a crucial strategy for building core components that ensure consistency and reusability across applications. Fig 3, shows how design systems provide a centralized repository of design patterns, UI components, and guidelines. This fosters a cohesive visual identity, accelerates development cycles, and ensures a seamless and consistent user experience across diverse interfaces.

3. Methodology

Systematic Literature Review: To ensure a comprehensive understanding of the landscape, a systematic literature review will be conducted [9]. This involves a rigorous search and analysis of academic databases, industry journals, conference proceedings, and relevant books. The selection criteria will prioritize studies that focus on frontend development in the e-commerce and retail, including re-platforming strategies, frameworks, and collaborative methodologies.

Case Study Analysis: In addition to the broader literature review, case studies will be analyzed to gain insights into real-world implementations of frontend re-platforming initiatives within similar industrial contexts. These case studies will be selected based on relevance to the retail sector, emphasizing successful re-platforming efforts, challenges faced, and the outcomes achieved. The analysis aims to extract practical lessons and patterns applicable to the proposed study.

4. Literature Review

To the best of the knowledge, only a limited number of studies have investigated. The methodology employed for conducting the literature review in support of this paper involved an exploration of diverse sources, encompassing scientific papers, articles on platforms like Medium, blogs, and the accumulated experience of developing user interfaces over the years within the targeted space. This multifaceted approach aimed to provide a holistic understanding of the evolution of frontend technologies, industry trends, and practical insights derived from real-world experiences [9]. A meticulous examination of scientific papers formed a foundational component of the literature review. By delving

into peer-reviewed journals and conference proceedings [14][25], the paper sought to integrate scholarly perspectives, theoretical frameworks, and empirical findings. This approach ensured the inclusion of rigorous research insights that contribute to the academic discourse on frontend technologies and re-platforming strategies.

The literature review incorporated the accumulated professional experience of developing user interfaces over the years within the specified industry space. This aspect ensured a practical grounding for the paper, with insights derived from hands-on involvement in frontend development projects. The inclusion of professional experience added a pragmatic layer to the literature review, bridging the gap between theoretical concepts and real-world applications.

In essence, the methodology for the literature review embraced a multidimensional approach, drawing insights from scientific papers for academic rigor, medium articles for contemporary perspectives, blogs for experiential knowledge, and professional experience for a practical understanding of the challenges and opportunities within the frontend ecosystem. This comprehensive strategy aimed to provide a well-rounded and insightful foundation for the paper's exploration of frontend re-platforming strategies.

Els Clarysse [10], outlines the evolution of frontend technologies over time from desktop-focused development in the 1980s to today where JavaScript has become the most prominent programming language. It attributes this rise to JavaScript's ability to power web applications, its high portability, and the growth of Node.js enabling fullstack JS development. Additionally, frameworks emerging help address issues with client-side scripting like component reuse. While providing useful context, the paper's focus is describing the history rather than conducting its own analysis.

Suarez [8] has explained the benefits of design systems for scaling design through reusable components. It describes key aspects like standards, components, patterns, voice and tone. Case studies are presented to exemplify visual language, components, and principles in practice. While comprehensive, the focus is on illustrating examples rather than conducting original research.

Peltonen S..et.al [7] proposes micro frontends as an approach to address issues in large monolithic frontends. It presents an example implementation and concludes micro frontends enable independent development and deployment. A limitation noted is requiring decoupling code. Compared to prior works, this paper aims to provide an overall picture of micro frontends as many prior works only covered parts. Novelty is assessing impacts qualitatively.

It becomes apparent that the challenges and solutions identified in the literature align closely with the practical implementations seen in real-world scenarios. Based on the study we found micro-frontends are typically implemented based on four fundamental elements: *componentization*, *framework flexibility*, *communication*, and *independent deployment*.

Componentization is a core principle of micro frontends, involving the decomposition of the user interface into small, self-contained units. These units, known as components, encompass their own logic, styles, and dependencies, leading to a modular structure. To illustrate, let's consider a standard e-commerce application with a product page featuring Ad campaigns on the left, trending deals on the right, and a search bar at the top. In this context, each component can function independently, encapsulating its specific functionality and capabilities.

Framework freedom is a significant benefit of micro frontends, as they are not bound to any specific frontend framework. This framework-agnostic nature allows development teams to select different frameworks or libraries for individual components, based on their specific requirements. Building upon the previous example, the trending deals component can be developed using React, while the Ad campaign component can utilize Vue.js. This flexibility empowers teams to leverage the unique strengths and capabilities of diverse technologies, while still maintaining a cohesive and unified system.

Communication between each other micro-frontends via well-defined APIs, promoting loose coupling and independence. For example, let's consider an e-commerce application where the product search component, developed using Angular, communicates with the shopping cart component built with React through a RESTful API. At the same time, the product display component, implemented using Vue.js, autonomously fetches product information directly from the backend server. This approach enables seamless interaction between the components while maintaining their individual autonomy and reducing dependencies.

Decentralized deployment is a key capability offered by micro frontends, enabling the independent deployment of individual components. This enables continuous delivery and mitigates the risk of system-wide failures. Development teams can release updates or address bugs in specific components without causing disruptions to the entire application. For example, imagine a scenario where the user authentication component is updated without impacting the live functionality of the product display or shopping cart

components. This decentralized deployment approach provides flexibility and resilience to the overall system architecture.

5. Case Studies

The following case studies delve into two common e-commerce challenges [13], and suggested approaches, providing examples of how frontend technologies are leveraged to overcome scalability, and collaboration in the ever-evolving digital landscape.

a) *Scalability [26] Enhancement for Seamless Shopping Experience:*

Challenge: Retail and e-commerce platforms often face the challenge of accommodating increased user traffic during peak shopping seasons, leading to scalability issues and performance bottlenecks.

Solution:

- **Adoption of Micro Frontends:** Implementing micro frontends allows for modular development and deployment, enabling retailers to scale specific components independently. This aligns with the demand for a seamless shopping experience, especially during high-traffic periods.
- **Integration of Progressive Web App (PWA) [27] Features:** Embracing PWA features enhances the overall user experience by providing offline capabilities and faster loading times. This ensures that customers can navigate the platform smoothly, even in challenging network conditions.

b) *Collaborative Development for Unified User Interfaces:*

Challenge: Collaboration challenges among development teams can result in fragmented user interfaces and inconsistent experiences across an e-commerce platform.

Solution:

- **Agile Transformation and Modern Tech Stack [28]:** Transitioning to Agile methodologies and modern tech stacks, such as React.js or Vue.js, fosters collaboration among development teams. This enables a more efficient and cohesive approach to frontend development, ensuring a unified and visually consistent user interface.
- **Implementation of Design Systems:** The adoption of design systems establishes standardized design elements across the platform, promoting collaboration and maintaining a cohesive visual identity.[8]

6. Conclusion

By combining the above methodological approaches—systematic literature review, case studies—the study provides a robust and multifaceted exploration of frontend re-platforming within the e-commerce and retail. The synthesis of theoretical insights and practical observations will contribute to the development of informed recommendations and strategies for achieving a unified and efficient frontend architecture.

Design systems aim to improve consistency, reusability, and maintainability of design work over long periods of time. This is achieved through standardizing user interfaces with reusable components. Standardizing interfaces through reusable patterns lowers the effort required for individual design work, allowing teams to focus on developing better experiences for users more quickly. Key documented benefits of reusable components include increasing development speeds due to existing patterns to prototype with, reducing design debt over time through consistency enforcement, and enabling faster iterative work through minimized refactoring.

While standardization provides benefits, it also introduces challenges such as potentially reduced flexibility if constraints are too restrictive. Striking the right balance between control and flexibility is important, as this balance will vary depending on factors like team size and experience level. Layout awareness in components represents promising future work, enabling rapid composition of screens from data. Done in conjunction with predictive assembly, this could automate customized flows and aid in customization to user needs.

Establishing clear code standards through linting and automation ensures consistency and prevents unintended divergence. But as systems evolve, standards themselves must refine through repeated refinement. Publishing documentation publicly aids wider understanding but also demands ongoing care as systems change. Measuring real-world metrics sheds light on typical usage and pain points to continuously learn and enhance performance. In closing, design systems provide a collaborative and iterative path forward as experimentation, community input and experience advance the collective knowledge.

In addition to the above methodological approaches, the concept of micro-frontends emerges as a crucial factor in achieving a unified and efficient frontend architecture. Micro-frontends offer componentization, framework flexibility, API-based communication, and decentralized deployment. The decomposition of the user interface into small, self-contained units enhances modularity and independence. The ability to work with different frontend frameworks empowers teams to leverage diverse technologies. API-based communication ensures seamless interaction between components. Furthermore, decentralized deployment allows for independent updates and bug fixes, reducing the risk of system-wide failures. By incorporating these principles, organizations can establish a robust and adaptable frontend architecture that promotes consistency, reusability, and maintainability over time.

7. Future Work

There are several promising avenues for taking this work forward. Additional case studies evaluating real-world implementations across different industries and project contexts could provide a larger data set to further validate or refine conclusions. Quantitative studies surveying

practitioners may also offer statistically significant insights into priorities and preferences when selecting frameworks. Automating measurement of proposed quality indicators could robustly analyze characteristic comparisons at larger scale. Experimenting through prototyping ideas like layout awareness or integrating AI could help prove out promising future directions. Collaborating with open-source projects provides opportunities to integrate findings, help ensure frameworks evolve to meet needs, and contribute new capabilities back to the community. Analyses could assess impacts of new technologies like Web Components. Expanding assessments to non-technical considerations and tracking long-term evolution as the field advances would likewise deepen understanding. Continued research brings academia and practice jointly toward shared progress in establishing best approaches for design at scale.

References

- [1] The evolution of e-commerce globally - NIQ." 23 Mar. 2022, <https://nielseniq.com/global/en/insights/analysis/2022/the-evolution-of-e-commerce-globally/>.
- [2] Y. Xing, "Research and Analysis of the Front-end Frameworks and Libraries in EBusiness," 2019. https://www.researchgate.net/publication/332456776_Research_and_Analysis_of_the_Front-end_Frameworks_and_Libraries_in_E-Business_Development
- [3] S. E. Ullah, T. Alauddin and H. U. Zaman, "Developing an E-commerce website," 2016 *International Conference on Microelectronics, Computing and Communications (MicroCom)*, Durgapur, India, 2016, pp. 1-4, doi: 10.1109/MicroCom.2016.7522526.
- [4] npn, 2020, The evolution of front-end development, <https://cloudreports.net/the-evolution-of-front-end-development/>
- [5] Fink, Gil & Flatow, Ido. (2014). *Introducing Single Page Applications*. 10.1007/978-1-4302-6674-7_1.
- [6] Google Trends :<https://trends.google.com/trends/explore?date=all&q=%2Fm%2F01211vxv,%2Fg%2F11c6w0ddw9,%2Fg%2F11c0vmgx5d,%2Fg%2F11c5t00h04>
- [7] Peltonen S., Mezzalira L., Taibi D. Motivations, benefits, and issues for adopting Micro-Frontends: A Multivocal Literature Review *Inf. Softw. Technol.*, 136 (2021), Article 106571, <https://www.sciencedirect.com/science/article/pii/S0950584921000549#d1e707>
- [8] Suarez, M. (2019). *Design Systems Handbook*. W3C mission. (2021). Hämtat från W3C Webplats: https://s3.amazonaws.com/designco-web-assets/uploads/2019/05/InVision_DesignSystemsHandbook.pdf
- [9] V. Garousi, M. Felderer, M.V. Mäntylä, Guidelines for including grey literature and conducting multivocal literature reviews in software engineering, *Inf. Softw. Technol.* 106 (2019) 101–121.
- [10] Els Clarysse (2019), QUALITY EVALUATION OF MODERN FRONT-END WEB DEVELOPMENT FRAMEWORKS,

- https://libstore.ugent.be/fulltxt/RUG01/002/837/311/RUG01-002837311_2020_0001_AC.pdf
- [11] F. Auer, V. Lenarduzzi, M. Felderer, D. Taibi From monolithic systems to Microservices: an assessment framework, <https://www.sciencedirect.com/science/article/pii/S0950584923001891>
- [12] Michael Wanyoike, (2018) History of front-end frameworks, <https://blog.logrocket.com/history-of-frontend-frameworks/>
- [13] Ehikioya, S.A.; Guillemot, E. A critical assessment of the design issues in e-commerce systems development. *Eng. Rep.* 2020, 2, e12154. <https://onlinelibrary.wiley.com/doi/full/10.1002/eng2.12155>
- [14] Wang D, Yang D, Zhou H, Wang Y, Hong D, Dong Q, Song S (2020) A novel application of educational management information system based on micro frontends. *Proced Comput Sci* 176:1567–1576 <https://www.sciencedirect.com/science/article/pii/S1877050920320688>
- [15] Martin, Fowler (2019). Micro Frontends. <https://martinfowler.com/articles/micro-frontends.html>
- [16] Tyler McMullen (2018). Micro-Frontends: A Microservice Architecture for the Frontend. <https://www.alligator.io/nodejs/micro-frontends/>
- [17] Abhijit Keni (2022). Micro Frontends: Theory and Best Practices. <https://www.bugrapostaci.com/micro-frontends-theory-and-best-practices/>
- [18] David Linthicum (2019). Microservices vs. Micro-frontends: To BFF or Not to BFF. <https://www.infoq.com/articles/microservices-vs-microfrontends/>
- [19] Luca Mezzalana (2021), Building Micro-Frontends, <https://www.oreilly.com/library/view/building-micro-frontends/9781492082989/>
- [20] Ian Feather (2021), Micro Frontends at BuzzFeed, <https://tech.buzzfeed.com/micro-frontends-at-buzzfeed-b8754b31d178>
- [21] Benjamin Johnson (2018), Exploring micro-frontends, <https://medium.com/@benjamin.d.johnson/exploring-micro-frontends-87a120b3f71c>
- [22] Phodal Huang (2019), Micro-frontend Architecture in Action with six ways. <https://dev.to/phodal/micro-frontend-architecture-in-action-4n60>
- [23] Michael Geers, extending the microservice idea to frontend development, <https://micro-frontends.org/>
- [24] KBall (2019), Microfrontends: the good, the bad, and the ugly, <https://zendev.com/2019/06/17/microfrontends-good-bad-ugly.html>
- [25] Nitin Jain (2019), Micro Frontend Curry, <https://levelup.gitconnected.com/micro-frontend-curry-506b98a4cfc0>
- [26] T. C. Chieu, A. Mohindra and A. A. Karve, "Scalability and Performance of Web Applications in a Compute Cloud," *2011 IEEE 8th International Conference on e-Business Engineering*, Beijing, China, 2011, pp. 317-323, doi: 10.1109/ICEBE.2011.63.
- [27] Adetunji, Oluwatofunmi & Ajaegbu, Chigozirim & Nzechukwu, Otuneme. (2020). Dawning of Progressive Web Applications (PWA): Edging Out the Pitfalls of Traditional Mobile Development. *American Scientific Research Journal for Engineering, Technology, and Sciences*. 68. 85-99.
- [28] Betz, Charles. (2016). Implications of digital transformation, Agile, and DevOps for IT curricula, with recommendations.

Author Profile

Manoj Dobbala is an experienced leader in software engineering, renowned for his ability to build and guide high-performing engineering teams. He earned his master's degree from West Virginia University (WVU) in 2012, where he made substantial contributions to a research team, co-authoring publications and completing a thesis. Over the years spanning from 2012 to 2024, he has amassed diverse experience across industries including healthcare, telecommunications, banking, and retail. Notably, he played a pivotal role in frontend transformation and replatforming endeavors at HomeDepot before recently assuming the role of spearheading Genuine Parts Company's replatforming efforts and transition to the cloud.