Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Analysis, Design and Implementation of Micro-frontend Architecture

Bachelor Thesis

2024
Bc. Pavol Repiský

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

Analysis, Design and Implementation of Micro-frontend Architecture

Bachelor Thesis

Study Programme:	Applied Computer Science
Field of Study:	Computer Science
Department:	Department of Computer Science
Supervisor:	RNDr. Ľubor Šešera, PhD.
Consultant:	Ing. Juraj Marák

Bratislava, 2024

Bc. Pavol Repiský

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Pavol Repiský
Študijný program:	aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	informatika
Typ záverečnej práce:	diplomová
Jazyk záverečnej práce:	anglický
Sekundárny jazyk:	slovenský

Názov: Analysis, Design and Implementation of Micro-frontend Architecture
Analýza, návrh a implementácia mikrofrontendovej architektúry

Anotácia: Mikrofrontendy predstavujú ďalší logický krok vo vývoji architektúry webových aplikácií. Tento prístup si však vyžaduje zvýšenie zložitosti architektúry a vývoja projektu. Problémy ako smerovanie, opätovná použiteľnosť, poskytovanie statických aktív, organizácia úložiska a ďalšie sú stále predmetom značnej diskusie a komunita ešte musí nájsť riešenia, ktoré dokážu efektívne spustiť projekt a riadiť výslednú zložitosť. Aj keď boli navrhnuté a diskutované niektoré prístupy, existuje veľké množstvo poznatkov a potenciálu na objavenie nových prístupov.

Cieľ: Preskúmajte existujúcu literatúru o prístupoch k návrhu a vývoju webových aplikácií pomocou mikro-frontend architektúry.
Porovnajte existujúce prístupy z hľadiska opätovnej použiteľnosti, rozšíriteľnosti, zdieľania zdrojov a správy stavu aplikácií.
Identifikujte prístupy, ktoré sú najvhodnejšie pre vývoj podnikových aplikácií, potom navrhnite a implementujte prototypovú mikrofrontendovú aplikáciu pomocou jedného vybraného prístupu.

Literatúra: https://www.researchgate.net/publication/351282486_Micro-frontends_application_of_microservices_to_web_front-ends
https://www.angulararchitects.io/blog/micro-apps-with-web-components-using-angular-elements/
https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1570726&dswid=5530
https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1778834&dswid=-4588
https://www.scientificbulletin.upb.ro/rev_docs_arhiva/reze1d_965048.pdf

Vedúci:	RNDr. Ľubor Šešera, PhD.
Konzultant:	Ing. Juraj Marák
Katedra:	FMFI.KAI - Katedra aplikovanej informatiky
Vedúci katedry:	doc. RNDr. Tatiana Jajcayová, PhD.

Spôsob sprístupnenia elektronickej verzie práce:
bez obmedzenia

Comenius University Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Bc. Pavol Repiský
Study programme:	Applied Computer Science (Single degree study, master II. deg., full time form)
Field of Study:	Computer Science
Type of Thesis:	Diploma Thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Analysis, Design and Implementation of Micro-frontend Architecture

Annotation: Micro-frontends represents the next logical step in the development of a web-application architecture. However, this approach necessitates an increase in the complexity of the project architecture and development. Issues such as routing, reusability, static asset serving, repository organization, and more are still the subject of considerable discussion, and the community has yet to find any solutions that can effectively bootstrap a project and manage the resulting complexity. While there have been some approaches proposed and discussed, there is a great deal of knowledge and potential for new approaches to be discovered.

Aim: Review existing literature about approaches to design and development of web applications using micro-frontend architecture.
Compare existing approaches from aspects of reusability, extendibility, resource sharing and application state management.
Identify approaches best suited for enterprise application development, then design and implement a prototypical micro-frontend application using one selected approach.

Literature: https://www.researchgate.net/publication/351282486_Micro-frontends_application_of_microservices_to_web_front-ends
https://www.angulararchitects.io/blog/micro-apps-with-web-components-using-angular-elements/
https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1570726&dswid=5530
https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1778834&dswid=-4588
https://www.scientificbulletin.upb.ro/rev_docs_arhiva/reze1d_965048.pdf

Supervisor:	RNDr. Ľubor Šešera, PhD.
Consultant:	Ing. Juraj Marák
Department:	FMFI.KAI - Department of Applied Informatics
Head of department:	doc. RNDr. Tatiana Jajcayová, PhD.
Assigned:	05.10.2023
Approved:	05.10.2023

prof. RNDr. Roman Ďurikovič, PhD.
Guarantor of Study Programme

Acknowledgments: Tu môžete poďakovať školiteľovi, prípadne ďalším osobám, ktoré vám s prácou nejako pomohli, poradili, poskytli dáta a podobne.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background and Motivation

This section provides an insightful overview of the background that led to the initiation of the research, addressing the historical context and the driving forces that underscore the significance of the study. It aims to offer readers a clear understanding of the factors and events that motivated the exploration of micro-frontend architecture in web applications.

1.2 Objectives and Scope

Focused on defining the purpose and boundaries of the research, this section outlines the specific objectives that the study aims to achieve. It delineates the scope of the investigation, detailing the aspects, dimensions, or features of micro-frontend architecture that are within the purview of the research. This section serves as a roadmap, guiding readers on what to expect from the study.

1.3 Key Research Questions

Central to the research inquiry, this section formulates and presents the key research questions that guide the investigation. These questions encapsulate the core issues and uncertainties that the study seeks to address. They provide a framework for the subsequent analysis and evaluation of existing approaches to web application design using micro-frontend architecture.

1.4 Scope and Limitations

Acknowledging the constraints and boundaries of the study, this section transparently discusses the limitations and delimitations inherent in the research. It addresses potential challenges, external factors, and constraints that might influence the research outcomes. By clearly defining these limitations, readers gain a realistic understanding of the study's scope and potential implications.

Chapter 2

Literature Review

2.1 Evolution of Frontend Architectures

This section embarks on a journey through the historical evolution of frontend architectures, tracing the origins and transformative phases that have shaped the landscape of web development. It explores the progression from traditional monolithic structures to more contemporary modular approaches, setting the stage for understanding the dynamic nature of frontend architectures.

2.2 Overview of Existing Approaches

Focused on contemporary developments, this section provides a comprehensive survey of the existing approaches to frontend architecture. It offers a detailed examination of diverse methodologies, frameworks, and paradigms employed in web application development. The exploration encompasses a spectrum of approaches, including monolithic architectures, component-based frameworks, and emerging trends.

Chapter 3

Theoretical Framework

3.1 Architecture Types

Explore various architectural patterns prevalent in web development, including monolithic, microservices, serverless, and progressive web app (PWA) architectures. Analyze the characteristics and suitability of each type, providing a comparative study to guide web developers in choosing the most appropriate architecture for their specific project requirements.

3.2 Communication Protocols

Investigate communication mechanisms and protocols essential for seamless interactions between web components. Discuss common approaches like RESTful APIs, GraphQL, and WebSocket protocols. Evaluate the trade-offs associated with each protocol, considering factors such as performance, scalability, and real-time capabilities, offering insights to help web developers make informed communication strategy decisions.

3.3 Composition Strategies

Investigate communication mechanisms and protocols essential for seamless interactions between web components. Discuss common approaches like RESTful APIs, GraphQL, and WebSocket protocols. Evaluate the trade-offs associated with each protocol, considering factors such as performance, scalability, and real-time capabilities, offering insights to help web developers make informed communication strategy decisions.

3.4 State Management

Investigate the nuances of managing state within web applications. Cover aspects like client-side and server-side state management, data synchronization, and state persistence. Discuss the impact on user experience, data consistency, and scalability, and evaluate popular state management libraries and patterns within the context of web development.

3.5 Versioning Strategies

Explore strategies for versioning web applications to ensure smooth deployment, updates, and maintenance. Discuss versioning techniques such as Semantic Versioning, API versioning, and content delivery network (CDN) caching. Provide guidance on selecting appropriate versioning strategies based on the unique needs and constraints of web development projects.

Chapter 4

Analysis

4.1 Introduction to Microfrontends

4.1.1 Background

Provide an overview of the evolution of web development architectures, emphasizing the challenges associated with monolithic frontend architectures.

4.1.2 Definition of Microfrontends

Define microfrontends and highlight its significance in addressing the limitations of traditional monolithic frontend development.

4.1.3 Key Characteristics

Discuss the essential features and characteristics of microfrontends, such as modularity, independent deployment, and technology agnosticism.

4.2 Microservices Architecture

4.2.1 Overview

Briefly explain the concept of microservices architecture, drawing parallels between microfrontends and microservices.

4.2.2 Benefits and Challenges

Explore the advantages and challenges of microservices architecture, emphasizing how these factors influenced the adoption of microfrontends.

4.3 Architecture Patterns

4.3.1 Single Page Application (SPA)

Discuss the traditional SPA architecture and its limitations that led to the emergence of microfrontends.

4.3.2 Module Federation

Explain the concept of module federation as a key technique in microfrontends, allowing independent development and deployment of frontend modules.

4.3.3 Server-Side Includes (SSI)

Explore server-side includes and how they contribute to the composition of microfrontends, enabling seamless integration of different parts of the application.

4.3.4 Frontend Composition Strategies

Discuss various strategies for composing microfrontends, such as Build-Time Integration, Runtime Integration, and Server-Side Composition, highlighting their pros and cons.

4.4 Implementation Technologies

4.4.1 JavaScript Frameworks

Survey popular JavaScript frameworks (e.g., React, Angular, Vue.js) and their compatibility with microfrontends.

4.4.2 Web Components

Explore the role of web components in microfrontends, discussing how they contribute to the encapsulation and reusability of frontend elements.

4.4.3 Communication Protocols

Examine communication protocols used in microfrontends, such as Custom Events, Pub/Sub, and HTTP APIs.

4.5 Case Studies and Examples

Present case studies of organizations that have successfully adopted microfrontends, highlighting the benefits they achieved.

Chapter 5

Design

5.1 System Architecture

This section provides a comprehensive overview of the overall architecture of the microfrontends system. It delves into the high-level structure, outlining how various components interact to achieve the desired functionality. Emphasis is placed on the distributed nature of microfrontends, highlighting the modular design that facilitates independent development and deployment.

5.2 Component Design

Focusing on the individual building blocks of the system, this section details the design principles and considerations for each microfrontend component. It discusses the selection of technologies, design patterns, and the rationale behind component boundaries, ensuring a cohesive yet decoupled system.

5.3 Communication Protocols

Detailing the communication mechanisms between microfrontends and other system components, this section explores the chosen protocols and technologies. It covers inter-component communication, API design, and data exchange strategies, emphasizing the need for seamless collaboration while respecting the autonomy of each microfrontend.

5.4 Data Management

Addressing data handling and storage within the microfrontends ecosystem, this section outlines the strategies for managing both local and shared data. It discusses data synchronization, consistency, and the role of databases or other storage solutions, ensuring a robust and efficient data management approach.

5.5 User Interface (UI) Design

Focused on the end-user experience, this section delves into the principles and methodologies guiding the design of the user interfaces across microfrontends. It covers user interface components, styling, responsiveness, and accessibility, ensuring a cohesive and visually appealing user experience.

5.6 Navigation and Routing

Exploring the mechanisms guiding user navigation within the microfrontends system, this section discusses routing strategies, deep linking, and the overall user journey. It ensures that the navigation design aligns with the modular nature of microfrontends, providing a seamless and intuitive user experience.

5.7 Testing Strategy

Detailing the comprehensive testing approach, this section covers unit testing, integration testing, and end-to-end testing strategies for microfrontends. It emphasizes the importance of validating the functionality and interoperability of individual components and the system as a whole.

5.8 Deployment Strategy

Focused on the release and deployment of microfrontends, this section outlines the deployment pipeline, versioning, and rollback strategies. It discusses continuous integration and delivery practices, ensuring a smooth and efficient deployment process.

5.9 Versioning Strategy

Addressing the challenges of versioning in a microfrontends environment, this section discusses the chosen versioning strategy for both components and the overall system. It ensures compatibility across different versions and provides mechanisms for handling upgrades and rollbacks.

5.10 Scalability and Performance

This section addresses the scalability considerations and performance optimization techniques for the microfrontends system. It explores strategies for handling increased user loads, optimizing resource usage, and ensuring a responsive and efficient user experience as the system scales.

Chapter 6

Implementation

6.1 Overview

Provide an overview of the software, system, or algorithm being implemented. Clearly state the objectives and goals of the implementation.

6.2 Technology Stack

Detail the tools, programming languages, and frameworks used in the implementation. Include information about the hardware and software requirements.

6.3 Implementation Details

Provide a detailed account of the actual implementation. Break down the implementation into sub-sections based on functionalities or modules. Include code snippets where necessary, focusing on critical or complex parts.

6.4 User Interface

Describe the user interface design and implementation. Include screenshots or mockups to illustrate the user experience.

6.5 Performance Evaluation

Evaluate the performance of the implemented system or algorithm. Include relevant metrics and benchmarks. Compare the performance against any existing solutions or standards.

6.6 Limitations and Challenges

Acknowledge any limitations in the implementation.Discuss challenges faced during the development process and how they were addressed.

Chapter 7

Conclusion

7.1 Results

This section succinctly presents the findings and outcomes from the implementation and analysis of microfrontend architectures. It covers key metrics, performance indicators, and notable observations to support the thesis's conclusions.

7.2 Summary of Research Questions

This section provides a brief overview of the core research questions addressed in the thesis, summarizing their significance and objectives. It serves as a quick reference for readers to understand the driving inquiries behind the investigation into microfrontends.

7.3 Recommendations for Future Work

In this section, potential areas for future research and improvement in microfrontend architectures are outlined. It identifies gaps, limitations, and suggests avenues for continued exploration and enhancement. This forward-looking perspective encourages ongoing development and discussion in the field.

[1]

Bibliography

[1] Tobias Oetiker, Hubert Partl, Irene Hyna, and Elisabeth Schlegl. *Nie príliš stručný úvod do systému LaTeX2e.* 2002. Preklad Ján Buša ml. a st.