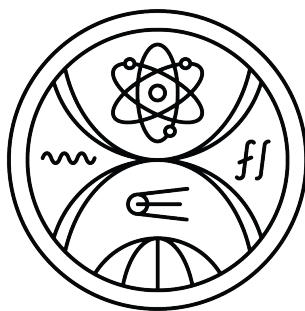


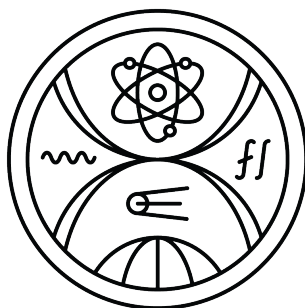
UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



ANALÝZA, NÁVRH A IMPLEMENTÁCIA
MIKROFRONTENDOVEJ ARCHITEKTÚRY -
ALTERNATÍVNE RIEŠENIE

Diplomová práca

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



ANALÝZA, NÁVRH A IMPLEMENTÁCIA MIKROFRONTENDOVEJ ARCHITEKTÚRY - ALTERNATÍVNE RIEŠENIE

Diplomová práca

Študijný program: Aplikovaná informatika
Študijný odbor: Informatika
Katedra: Katedra aplikovanej informatiky
Školiteľ: RNDr. Ľubor Šešera, PhD.
Konzultant: Ing. Juraj Marák



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta:	Bc. Marek Drahoš
Študijný program:	aplikovaná informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor:	informatika
Typ záverečnej práce:	diplomová
Jazyk záverečnej práce:	slovenský
Sekundárny jazyk:	anglický
Názov:	Analýza, návrh a implementácia mikrofrontendovej architektúry - alternatívne riešenie <i>Analysis, design and implementation of micro-frontend architecture – alternative solution</i>
Anotácia:	Mikrofrontendy predstavujú ďalší logický krok vo vývoji architektúry webových aplikácií. Tento prístup však nevyhnutne vedie k zvýšeniu komplexnosti projektovej architektúry a vývoja. Otázky ako "routing", opakovaná použiteľnosť, "static asset serving", "repository organization" a ďalšie sú stále predmetom diskusií a komunita a komunita bude ešte len musieť nájsť riešenia. Hoci už boli navrhnuté a diskutované niektoré riešenia, stále ešte existuje veľa znalostí a potenciálu pre objavenie nových prístupov.
Cieľ:	1. Prehľad existujúcej literatúry venovanej návrhu a vývoju webových aplikácií použitím mikrofrontendovej architektúry. 2. Porovnanie existujúcich prístupov z hľadiska rozšíriteľnosti a opakovanej použiteľnosti. 3. Návrh alternatívneho riešenia k riešeniu navrhnutému v diplomovej práci Analýza, návrh a implementácia mikrofrontendovej architektúry. 4. Implementácia prototypového mikrofrontendu použitím uvedeného alternatívneho návrhu.
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
Poznámka:	Táto diplomová práca bude príbuzná k inej diplomovej práci: Analýza, návrh a implementácia mikrontendovej architektúry. Študenti budú vychádzať z rovnakej literatúry. Každý zo študentov však spracuje naštudovanú literatúru samostatne. Môže však pri vyhodnotení tejto časti diplomovej práce dôjsť k zvýšenému prekryvu týchto dvoch diplomových prác pri vyhodnotení v CRZP.
Vedúci:	RNDr. Ľubor Šešera, PhD.
Konzultant:	Ing. Juraj Marák



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Katedra: FMFI.KAI - Katedra aplikovanej informatiky

Vedúci katedry: doc. RNDr. Tatiana Jajcayová, PhD.

Spôsob sprístupnenia elektronickej verzie práce:

bez obmedzenia

Dátum zadania: 24.11.2023

Dátum schválenia: 11.12.2023

prof. RNDr. Roman Ďurikovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Čestné prehlásenie

Bratislava, 2025

.....
Bc. Marek Drahoš

Pod'akovanie

Abstrakt

Táto práca si kládla za cieľ predstaviť mikrofrontendy ako spôsob vývoja frontendu webových aplikácií, ktorý zahŕňa rozdelenie používateľského rozhrania na viacero menších nezávislých častí. V práci analyzujeme základné princípy tohto prístupu, spôsoby kompozície webovej aplikácie v rámci mikrofrontendovej architektúry a rôzne možnosti implementácie. Dôležitou súčasťou je aj zhodnotenie prínosov a problémov, ktoré môžu vyplynúť z použitia mikrofrontendov. Prezentované poznatky sú aplikované v návrhu aplikácie založenej na aplikačnom kontajnery implementovanom prostredníctvom frameworku Blazor a mikrofrontendov založených na webkomponentoch implementovaných v TypeScripte s využitím Bootstrapu.

Kľúčové slová:

Abstract

This thesis aimed to introduce micro frontends as a method for developing the frontend of web applications, which involves dividing the user interface into several smaller, independent parts. The thesis analyzes the fundamental principles of this approach, methods of composing web applications within the micro frontend architecture, and various implementation options. An essential part is also the assessment of the benefits and challenges that may arise from using micro frontends. The insights presented are applied in the design of an application based on application container implemented through the Blazor framework and microfrontends based on webcomponents implemented in TypeScript with the use of Bootstrap framework.

Keywords:

Obsah

Úvod	1
1 História tvorby webových aplikácií	2
1.1 Frontend	2
2 Mikrofrontend	3
2.1 Architektúra orientovaná na služby	3
2.2 Princípy mikrofrontendovej architektúry	3
2.3 Kompozícia mikrofrontendovej architektúry	4
2.3.1 Kompozícia na strane servera	5
2.3.2 Kompozícia pomocou edge servera	5
2.3.3 Kompozícia na strane klienta	5
2.4 Prístupy k implementácii	5
2.4.1 Iframes	5
2.4.2 Webové komponenty	5
2.4.3 Module federation	6
2.4.4 Built Time Integration	6
2.5 Benefity	7
2.5.1 Možnosť použitia viacerých technológií	7
2.5.2 Nezávislé tímy	7
2.5.3 Samostatný vývoj, nasadzovanie, správa mikrofrontendových aplikácií	7
2.5.4 Izolácia chýb a odolnosť	7
2.6 Komplikácie	7
3 Navrh prototypu	9
4 Implementacia	10
5 Overenie/Diskusia k výsledkom	11
Záver	12

Úvod

Zložitosť webových aplikácií neustále narastá, čo vedie k vytváraniu komplikovaných monolitických používateľských rozhraní, ktoré sú náročné na udržiavanie, rozširovanie a škálovanie. Preto je dôležité neustále pracovať na nových prístupoch k ich tvorbe. Jedným z alternatívnych organizačných a architektonických štýlov, ktoré riešia tieto problémy, sú mikrofrontendy [1]. Tieto ponúkajú možnosť rozdeliť frontendovú časť aplikácie na samostatné a nezávislé mikroslužby, čím zvyšujú modularitu a znovupoužiteľnosť komponentov.

Cieľom práce je poskytnúť prehľad existujúcich prístupov k návrhu a vývoju webových aplikácií použitím mikrofrontendovej architektúry a porovnanie týchto prístupov z rôznych (SPECIFIKOVAT HLADISKA) hľadísk. Zdôraznia sa pri tom najnovšie trendy, najlepšie postupy a príklady úspešných implementácií. Medzi ďalšie ciele patrí navrhnutie riešenia pre dané problémy na základe získaných poznatkov. Tento návrh bude následne použitý na implementáciu prototypového mikrofrontendu, ktorý slúži ako dôkaz konceptu navrhnutého riešenia a umožňuje jeho testovanie a hodnotenie v reálnom prostredí.

V úvode práce sa venuje histórii vývoja webových aplikácií od ich počiatkov až po súčasnosť, pričom identifikujú kľúčové etapy a trendy, ktoré formovali ich architektúru. Ďalej sa zameriame na mikrofrontendovú architektúru, analyzujeme jej princípy, prístupy k kompozícii, implementačné možnosti a zhodnotíme výhody a nevýhody jej využitia.

Výsledkom práce je ...

Kapitola 1

História tvorby webových aplikácií

Filler obsahu ak bude treba nafúknuť prácu'. Prehľad historie vývoja webových aplikácií ako celku, vývoj backendu

1.1 Frontend

Predstavenie roznych pristupov (staticke stranky, server-side rendering (JSP, JSF...), SPA (angular, react...))

Kapitola 2

Mikrofrontend

Vývoj prezentačnej vrstvy prístupmi predstavenými v kapitole 1.1 najčastejšie vedie ku vzniku monolitckej aplikácie. To spôsobuje nárast komplexnosti a jej ďalšie úpravy môžu viesť k neželaným vedľajším efektom. Taktiež akokoľvek malá zmena znamená potrebu zostaviť a následne nasadiť celú aplikáciu pričom organizácia týchto činností pri súčasnej práci viacerých tímov je nesmierne náročná. Zaškolenie nových členov je komplikovanejšie a navyše dochádza k znižovaniu prínosov dodatočných zdrojov.

Tieto problémy sa objavujú aj pri vývoji backendu, pričom v súčasnosti najpoužívanejší/ jeden z najpoužívanejších prístupov pre ich riešenie je architektúra orientovaná na služby.(TREBA NAJST ZDROJ PRE TOTO TVRDENIE).

2.1 Architektúra orientovaná na služby

PRINCIPY SOA+MICROSERVICES

(zameriť sa na SOA alebo len čisto microservices?)

2.2 Princípy mikrofrontendovej architektúry

Mikrofrontendová architektúra je relatívne nový prístup k tvorbe prezentačnej vrstvy. Jej základným prvkom sú rovnaké princípy ako pri mikro službách, s cieľom premeniť jednu aplikáciu na viacero menších navzájom nezávislých aplikácií. Každá z nich by mala spĺňať nasledujúce vlastnosti prevzaté z mikro služieb:

- **Modelovaná okolo biznis domény** - aplikácia je vytvorená na jeden konkrétny cieľ,
- **Dôraz na automatizáciu** - robustná kontinuálna integrácia a kontinuálne nasadzovanie

- **Skrývanie implementačných detailov** - žiadna aplikácia nepotrebuje vedieť detaily inej aplikácie
- **Decentralizácia rozhodnutí** - tím môže nezávisle od ostatných zvoliť prístup k riešeniu problému, ktorému čelia
- **Nezávislé nasadzovanie** - nasadenie novej verzie jednej aplikácie nijako neovplyvňuje ostatné
- **Izolácia zlyhaní** - v prípade ak dôjde ku chybe v jednej aplikácii ostatné fungujú naďalej bez akejkoľvek zmeny.

Okrem toho vieme doplniť aj ďalšie vlastnosti definované podľa citáciaGeers:

- **Nezávislé na technológiách** - každá mikro služba môže byť implementovaná s použitím rôznych technológií,
- **Izolácia kódu** - každá aplikácia je izolovaná, nepoužíva premenné a stav inej aplikácie,
- **Tímové prefixy** - v prípade ak nie je možná úplná izolácia každá aplikácia by mala mať definovaný vlastný prefix,
- **Preferencia natívnych technológií prehliadača** - v prípade ak je to možné vždy volíme natívnu technológiu pred vývojom vlastného API.

2.3 Kompozícia mikrofrontendovej architektúry

Jedným z prvých rozhodnutí pri tvorbe mikrofrontendovej aplikácie by mal byť spôsob akým budeme pristupovať k danej architektúre z technického hľadiska, pričom máme dve možnosti.

Horizontálne členenie popisuje spôsob, kedy je jedna stránka tvorená viacerými nezávislými mikrofrontendmi. V tomto prípade je nevyhnutná koordinácia jednotlivých tímov a je potrebné určiť spôsob prípadnej komunikácie medzi aplikáciami.

Vertikálne členenie je každá stránka tvorená len jednou aplikáciou a je jasne definovaný spôsob smerovania žiadostí na jednotlivé mikrofrontendy. Samotnú kompozíciu aplikácií vieme realizovať tromi rôznymi prístupmi:

- kompozícia na strane servera,
- kompozícia pomocou edge servera,
- kompozícia na strane klienta.

2.3.1 Kompozícia na strane servera

Požiadavka klienta je spracovaná na servery, ktorý zavolá jednotlivé mikrofrontendy a výsledky volaní spojí do jedného výsledného dokumentu. Medzi hlavné výhody takéhoto riešenia patrí možnosť využitia rôznych cachovacích stratégií pre dosiahnutie čo najkratšieho času načítania finálnej verzie stránky. V prípade vysoko dynamickej stránky závislej od klienta naopak treba zvážiť škálovateľnosť takéhoto riešenia.

2.3.2 Kompozícia pomocou edge servera

Skladanie výslednej stránky prebieha s použitím Content Delivery Network (Slovník pojmov alebo vysvetlenie v texte). Využíva na XML založený značkový jazyk nazvaný Edge Side Include. Aj napriek tomu, že bol predstavený v roku 2001 stále neexistuje jednotná implementácia naprieč rôznymi poskytovateľmi, čo môže viesť k uzamknutiu na konkrétnu implementáciu.

2.3.3 Kompozícia na strane klienta

Finálna verzia stránky vzniká priamo v prehliadači klient, najčastejšie formou kontajnerovej aplikácii a viacerých mikrofrontendou pridaných pomocou Javascriptu, prípadne inej technológie.

2.4 Prístupy k implementácii

V súčasnosti existuje množstvo prístupov ako jednotlivé komponenty implementovať, pričom niektoré z nich si predstavíme.

2.4.1 Iframes

Pre integráciu jednotlivých komponentov používame iframe HTML tag pomocou ktorého načítavame druhý dokument na rovnakej stránke. Ich použitie zabezpečuje vysoký level izolácie mikrofrontendov, kde je nemožné zdieľať spoločné závislosti. To vedie k potrebe ich viacnásobného načítania čo má vplyv na výkonnosť systému.

2.4.2 Webové komponenty

Webové komponenty je súhrnné pomenovanie pre súbor špecifikácii, ktorých použitie umožňuje vytváranie vlastných tagov.

Custom Elements

Custom Elements tvoria základ pre definovanie vlastných typov DOM elementov, pričom autor je schopný definovať ich správanie a štýly. Pri definovaní nových typov môžeme vytvoriť úplne nové bez akejkoľvek funkcionality alebo rozšíriť už existujúce typy (len pre chrome?).

Shadow DOM

Tento štandard umožňuje pridávanie podstromov do elementov vo webovom dokumente. Takto vytvorené stromy sú zapuzdrené, to jest štýly definované v danom podstrome nie sú aplikované na zvyšok dokumentu.

ES Module

ES modul umožňuje modulárne vytváranie webových komponentov v samostatnom JavaScript súbore, ktorý následne zahrnieme do dokumentu.

HTML Template

HTML Template špecifikuje spôsob deklarácie fragmentov HTML tagov (značiek??), ktoré vieme neskôr kopírovať a vkladať do dokumentu. Ich obsah nemá pred ich použitím žiaden efekt na dokument.

2.4.3 Module federation

Tento prístup umožňuje asynchrónne načítanie vzdialených(angl. remote) modulov na požiadanie len v prípade keď ich potrebujeme. Každý modul v tomto prípade predstavuje samostatný mikrofrontend, pričom je taktiež možné zdieľanie závislostí naprieč modulmi.

2.4.4 Built Time Integration

Integrácia prebieha počas zostavovania aplikácie, kde sú jednotlivé komponenty definované ako závislosti kontajnerovej aplikácie. Tento prístup však vedie k porušeniu konceptov architektúry, konkrétne znemožňuje samostatné nasadzovanie mikrofrontendov. Pri zmene jedného z nich je potrebné znovu zostaviť a nasadiť minimálne kontajnerovú aplikáciu čo môže viesť k potrebe zostavenia ostatných komponentov.

2.5 Benefits

Mikrofrontendová architektúra prináša množstvo benefitov. (príde mi to ako opakovanie sekcie o princípoch)

2.5.1 Možnosť použitia viacerých technológií

Nakoľko mikrofrontendová architektúra kombinuje viacero nezávislých komponentov každý z nich môže využívať iné technológie, čo taktiež umožňuje použitie novo predstavených knižníc v budúcnosti.

2.5.2 Nezávislé tímy

Jasná separácia na základe biznis logiky umožňuje vytváranie nezávislých tímov ktoré sú slobodné vo svojich rozhodnutiach nakoľko tie neovplyvňujú žiadnu inú súčasť systému. Tímy môžu taktiež pokrývať všetky súčasti danej funkcionality, vrátane backendu a databázy.

2.5.3 Samostatný vývoj, nasadzovanie, správa mikrofrontendových aplikácií

Separácia komponentov prináša možnosť paralelnej práce viacerých tímov, ktorý navyše môžu svoje zmeny nasadzovať a testovať bez vplyvu na ostatných, čo vedie k zníženiu času potrebného na vývoj a k rýchlejšiemu nasadeniu. Taktiež jednotlivé zdrojové súbory budú menšie a ľahšie na údržbu.

2.5.4 Izolácia chýb a odolnosť

Pri správnej izolácii komponentov každá chyba spôsobí výpadok len jednej časti systému a jej prípadné odstavenie nemá vplyv na aplikáciu ako celok.

2.6 Komplikácie

(kapitola bude prepracovaná zatiaľ len jednoduché vymenovanie) Napriek všetkým benefitom ktoré mikrofrontendy prinášajú nepredstavujú univerzálne riešenie pre vývoj frontendu. Medzi problémy ktoré prinášajú patria:

- Zväčšenie množstva dát
- Opakovanie kodu
- Zdielane závislosti

- Konzistencia používateľského rozhrania
- Monitorovanie
- Zvýšená komplexita
- Správa (governance)

Kapitola 3

Navrh prototypu

Kapitola 4

Implementacia

Kapitola 5

Overenie/Diskusia k vysledkom

Záver

Literatúra

- [1] Michael Geers. *Micro Frontends in Action*. Manning, 2020.
- [2] Jouni Männistö, Antti-Pekka Tuovinen, and Mikko Raatikainen. Experiences on a frameworkless micro-frontend architecture in a small organization. In *2023 IEEE 20th International Conference on Software Architecture Companion (ICSA-C)*, pages 61–67, 2023.
- [3] Andrei Pavlenko, Nursultan Askarbekuly, Swati Megha, and Manuel Mazzara. Micro-frontends: application of microservices to web front-ends. 2020.
- [4] Severi Peltonen, Luca Mezzalana, and Davide Taibi. Motivations, benefits, and issues for adopting micro-frontends: A multivocal literature review. *Information and Software Technology*, 136:106571, 2021.