Benefits, Challenges, and Performance Analysis of a Scalable Web Architecture Based on Micro-Frontends

Adrian Petcu, Madalin Frunzete, Dan Alexandru Stoichescu

University "Politehnica" of Bucharest, Romania

U.P.B. Sci. Bull., Series C, Vol. 85, Iss. 3, 2023

Table of Contents

Introduction

Web applications have evolved to support parallel development across multiple layers and teams

The shift from monolithic to microservice-based architectures has transformed backend development

Front-end applications lack a simple, scalable implementation pattern comparable to backend microservices

Micro-frontends extend the microservice philosophy to the UI layer, enabling independent development and scalable codebases

Objective: Explore the benefits, challenges, and performance of a scalable architecture based on micro-frontends

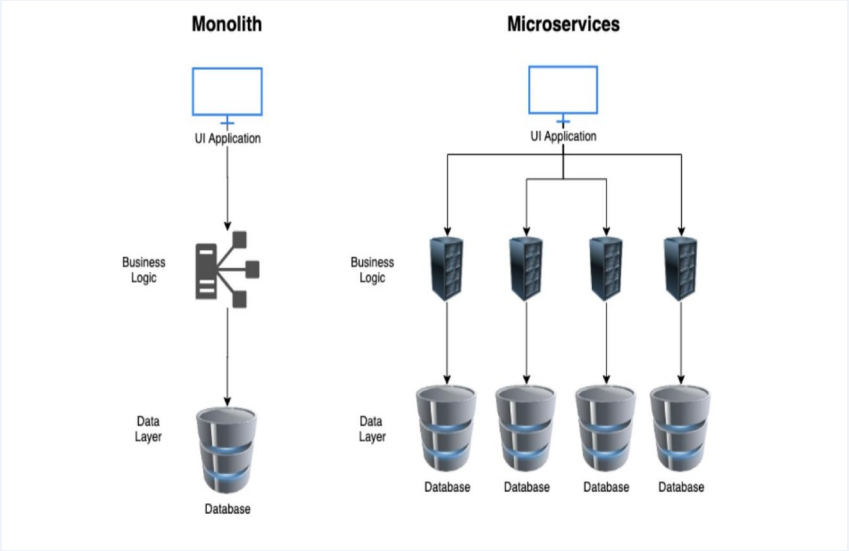
Architectural Overview

Monolithic Architecture

- Single-tiered application; all components bundled as one unit
- Full redeployment for minor changes
- Growing build times, difficult maintenance
- Poor fault isolation

Microservices Architecture

- Small, autonomous services by sub-domain
- Independently developed, deployed, and scaled
- Continuous delivery, improved fault isolation
- Technology freedom per service



Monoliths scale vertically; microservices scale horizontally

Figure 1. Monolith vs Microservice Architecture

Micro-Frontends: Concept and Motivation

- As backend migrates to microservices, front-end monoliths grow larger and harder to maintain
- Micro-frontends apply the microservice paradigm to the UI layer
- Application split into independent units by functionality or domain
- Each unit owned end-to-end by a cross-functional team
- Loose coupling via well-defined contracts
- Enables technology agnosticism across teams

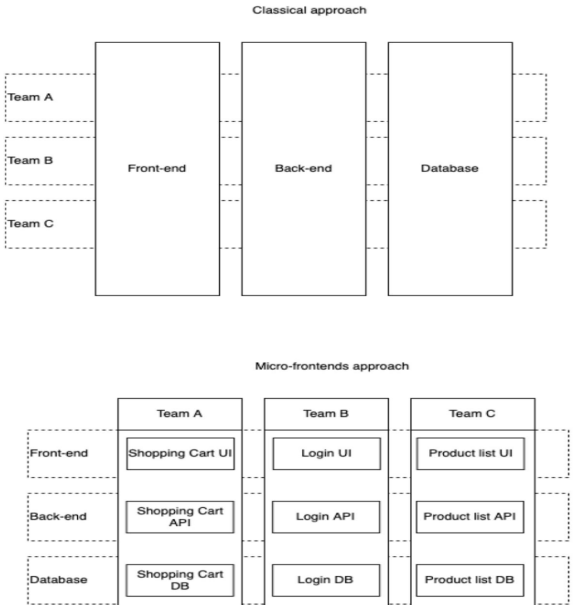


Figure 2. Cross-functional teams with micro-frontends

Composition Types and Splitting Strategies

Horizontal Split

- Multiple micro-frontends on the same page
- Each team responsible for a screen section
- Requires inter-team coordination

Vertical Split

- Each micro-frontend represents an entire page
- Simpler team boundaries and ownership

Solution	Description
Routing	Each route maps to a different micro-frontend
Iframe	Micro-frontends embedded via frames
Web Components	Framework-agnostic browser APIs
Module Federation	Dynamic loading of code and shared resources

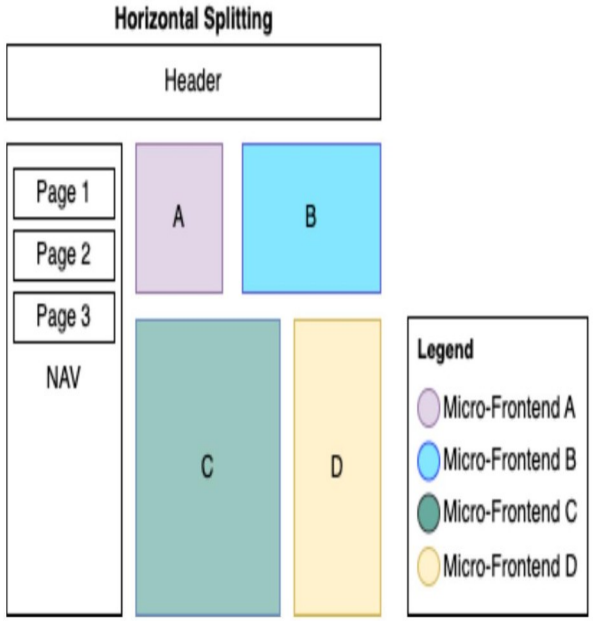


Figure 3. Horizontal splitting

Benefits of Micro-Frontend Architecture

Incremental Updates

Gradual migration from monolith; isolated experiments on parts of the application

Decoupled Codebases

Smaller, focused repositories reduce complexity and code duplication

Independent Deployments

Each micro-frontend released independently; failures impact only one UI area

Autonomous Teams

Cross-functional teams own code quality, business logic, framework, and styling

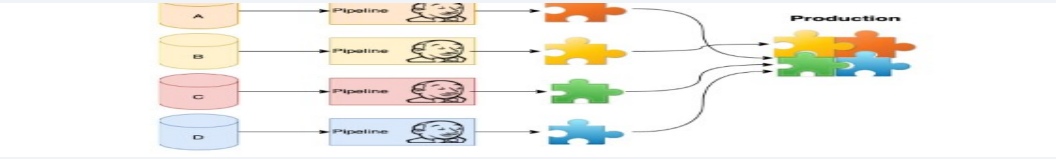


Figure 5. Independent deployments

Challenges of Micro-Frontend Architecture

Inter-unit Communication
Robust framework for coordinating events between separate sections

Backward Compatibility
Shell application modifications must not break existing micro-frontends

Standardized Contracts
Well-defined inputs and outputs for inter-unit communication

Centralized Communication
Publisher/Subscriber pattern for multi-component coordination

Bundle Size Control
Framework core instantiated only as needed, not duplicated per unit

Consistent Styling
Shared styling library for visual consistency across all units

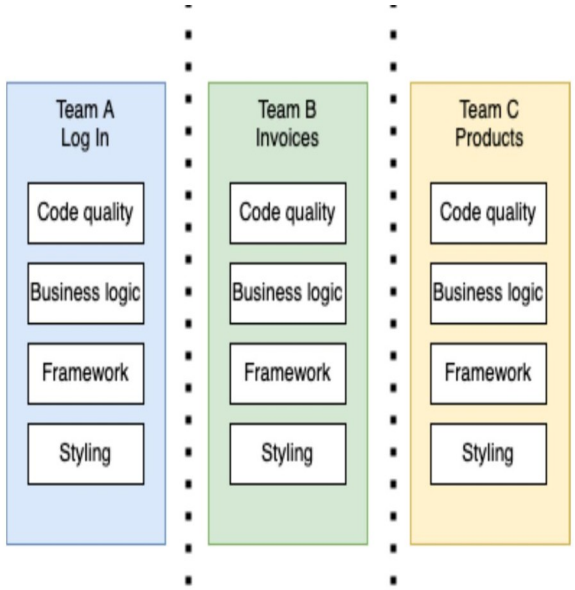


Figure 6. Team ownership

Results: Performance Comparison

Criterion	Monolith	Iframe	Module Fed.
First Paint	418 ms	**1222 ms**	540 ms
Requests	**13**	**34**	26
Resources	**5.4 MB**	**16.6 MB**	6.6 MB
Load Time	1.35 s	1.12 s	**0.774 s**

Key Findings

-55%
first paint time vs. Iframe

-60%
bundle size vs. Iframe

-42%
load time vs. Monolith

-23%
requests vs. Iframe

Module Federation expected to outperform monolith for larger applications

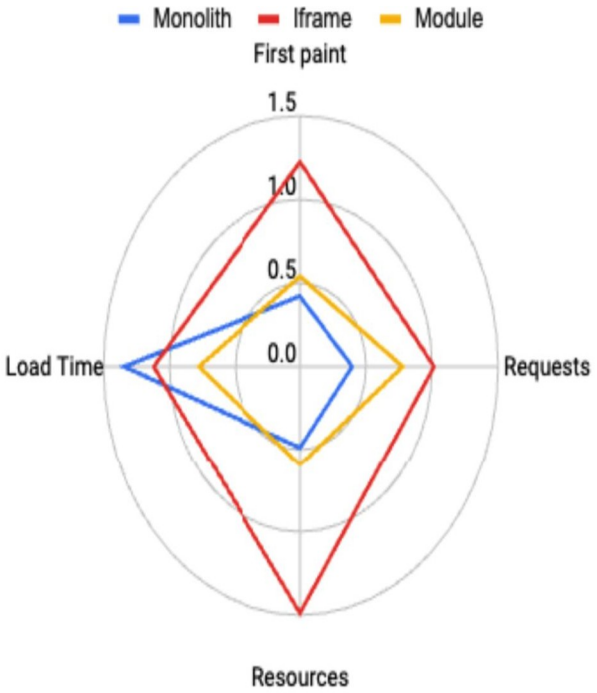


Figure 10. Relative comparison between solutions

Thank You

Questions?

adrian.petcu@stud.etti.upb.ro

Petcu, A., Frunzete, M., & Stoichescu, D. A. (2023).
U.P.B. Sci. Bull., Series C, 85(3), 319–334.