Microfrontends with @

Munich.NET, 2020-05-20

Dr. Florian Rappl

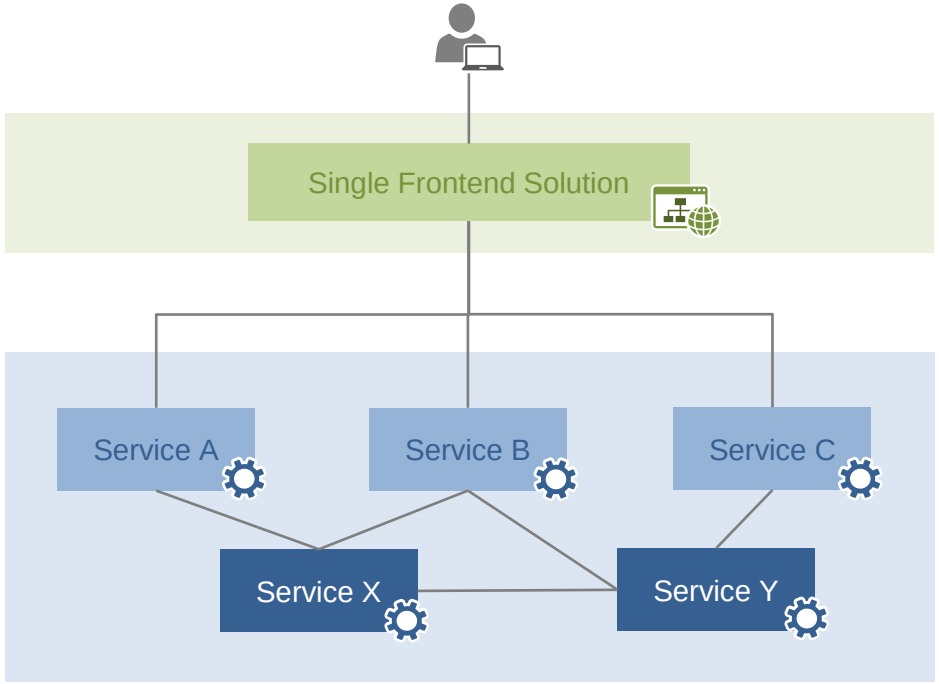
meetup.com/Munchen-NET-Meetup/events/270154213

Hello, I'm Florian!

Solution Architect at smapiot
IoT / Embedded Computing
Digital Transformation
Distributed Web Applications

Open-Source Enthusiast
Microsoft MVP Development Tools
Projects for .NET and JS
Articles for various blogs and magazines

System Architecture Example

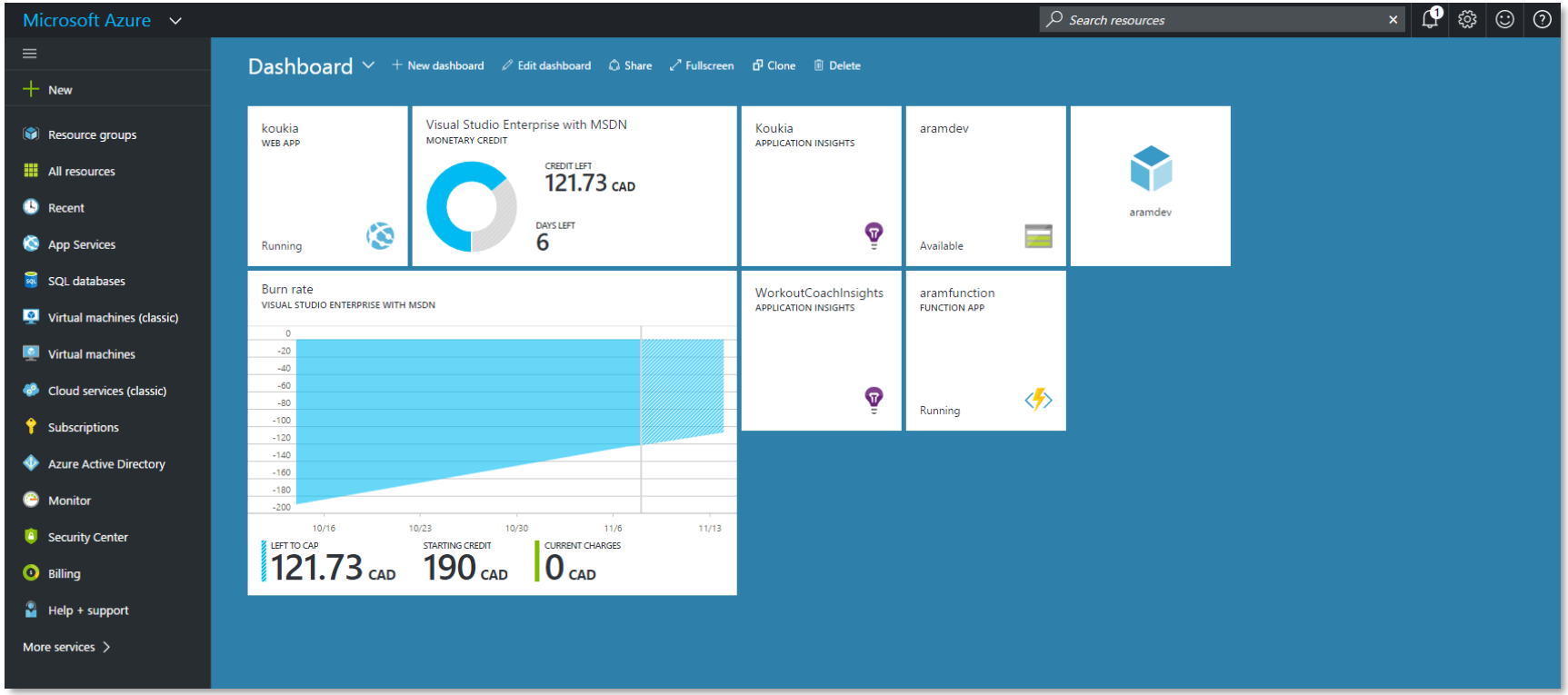


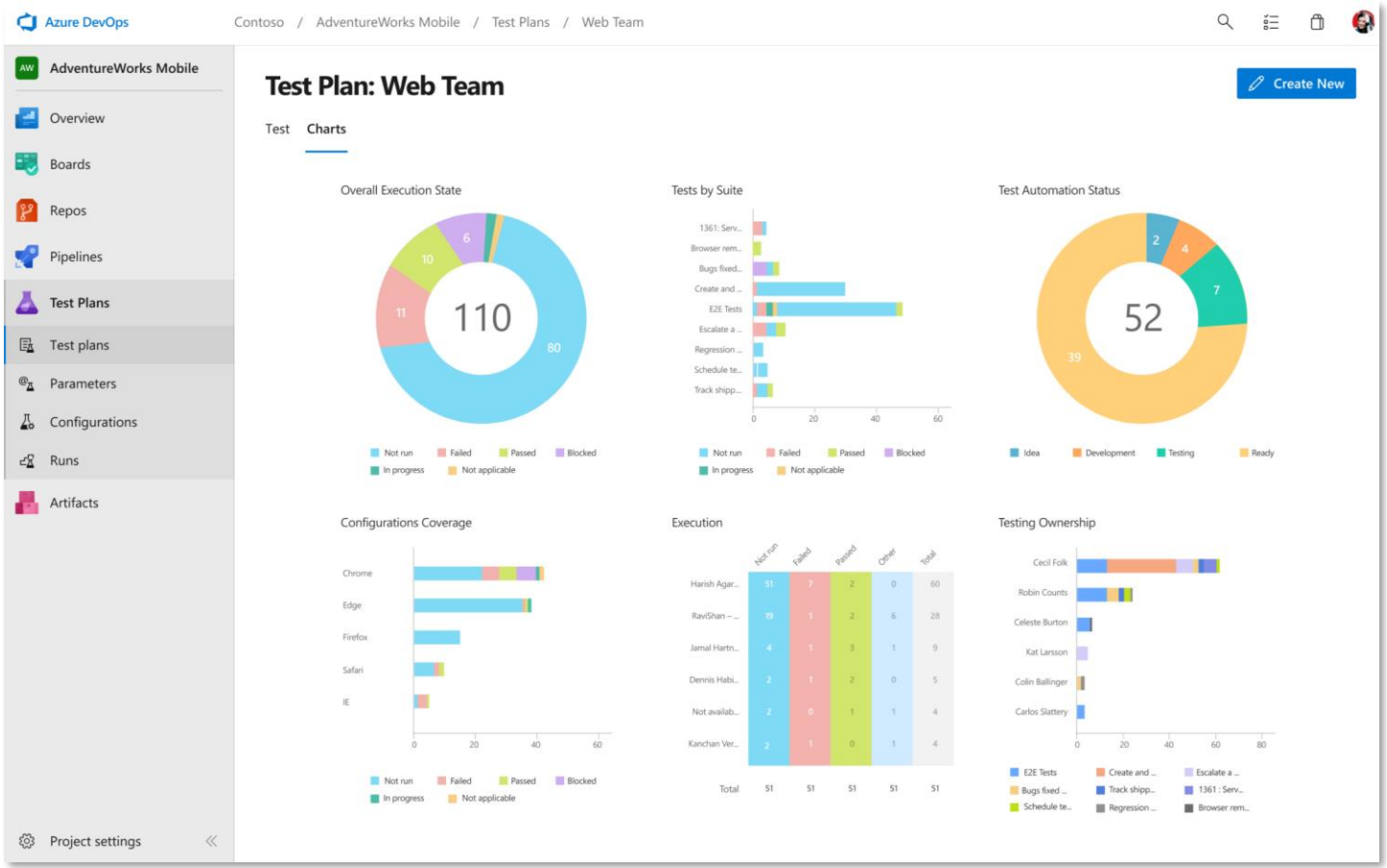
Frontend

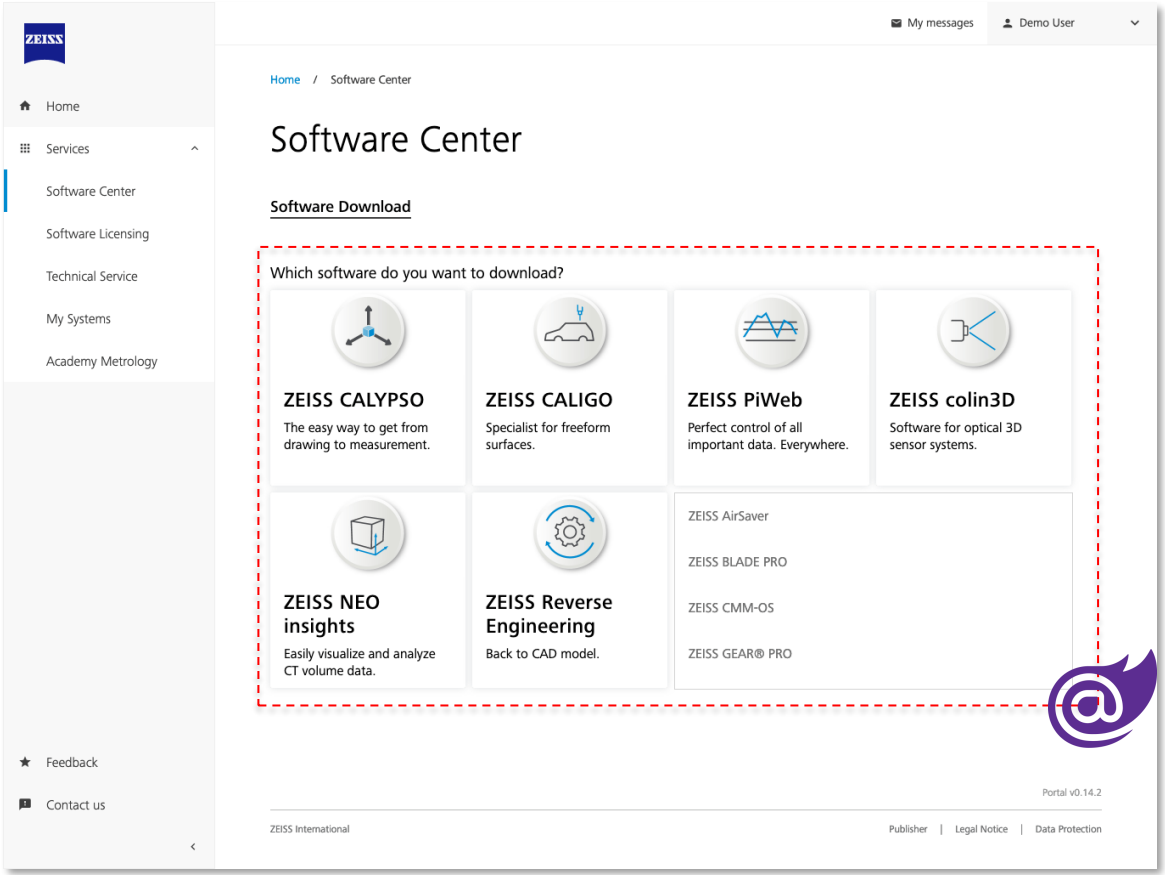
- Monolithic architecture
- Integrated and consistent user experience
- Developed by one central team
- Larger deployment releases

Microservices

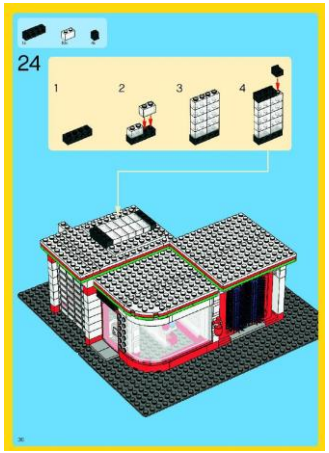
- Scoped around business capabilities
- Developed by autonomous teams
- Own development & deployment lifecycle
- Loosely coupled
- Technology independent


Microsoft Azure
Search resources
New
Resource groups
All resources
Recent
App Services
SQL databases
Virtual machines (classic)
Virtual machines
Cloud services (classic)
Subscriptions
Azure Active Directory
Monitor
Security Center
Billing
Help + support
More services
Dashboard
New dashboard
Edit dashboard
Share
Fullscreen
Clone
Delete
koukia
WEB APP
Running
Visual Studio Enterprise with MSDN
MONETARY CREDIT
CREDIT LEFT
121.73 CAD
DAYS LEFT
6
Koukia
APPLICATION INSIGHTS
aramdev
Available
aramdev
Burn rate
VISUAL STUDIO ENTERPRISE WITH MSDN
0
-20
-40
-60
-80
-100
-120
-140
-160
-180
-200
10/16
10/23
10/30
11/6
11/13
LEFT TO CAP
121.73 CAD
STARTING CREDIT
190 CAD
CURRENT CHARGES
0 CAD
WorkoutCoachInsights
APPLICATION INSIGHTS
aramfunction
FUNCTION APP
Running


Azure DevOps
Contoso / AdventureWorks Mobile / Test Plans / Web Team
AdventureWorks Mobile
Overview
Boards
Repos
Pipelines
Test Plans
Test plans
Parameters
Configurations
Runs
Artifacts
Project settings
Test Plan: Web Team
Create New
Test
Charts
Overall Execution State
110
Not run
Failed
Passed
Blocked
In progress
Not applicable
Tests by Suite
Test Automation Status
52
Idea
Development
Testing
Ready
Configurations Coverage
Chrome
Edge
Firefox
Safari
IE
Execution
Testing Ownership
Cecil Folk
Robin Counts
Celeste Burton
Kat Larsson
Colin Ballinger
Carlos Slattery

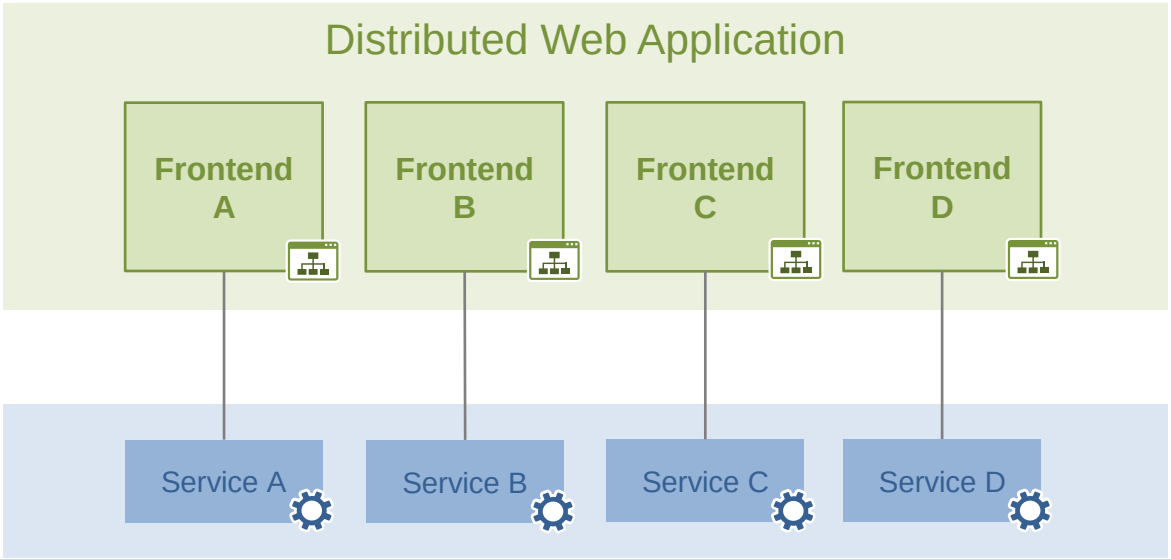


* this is not actually Blazor, but you get the idea



Microfrontend Architecture

Microfrontend Architecture



Frontend

Microfrontend

Backend

Microservice

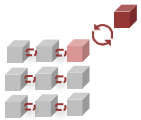
Desired Solution

Microservices Aspects

Aspects of a Monolith Approach

Best of both worlds for a **modular distributed web application**

Business capabilities as **modules**



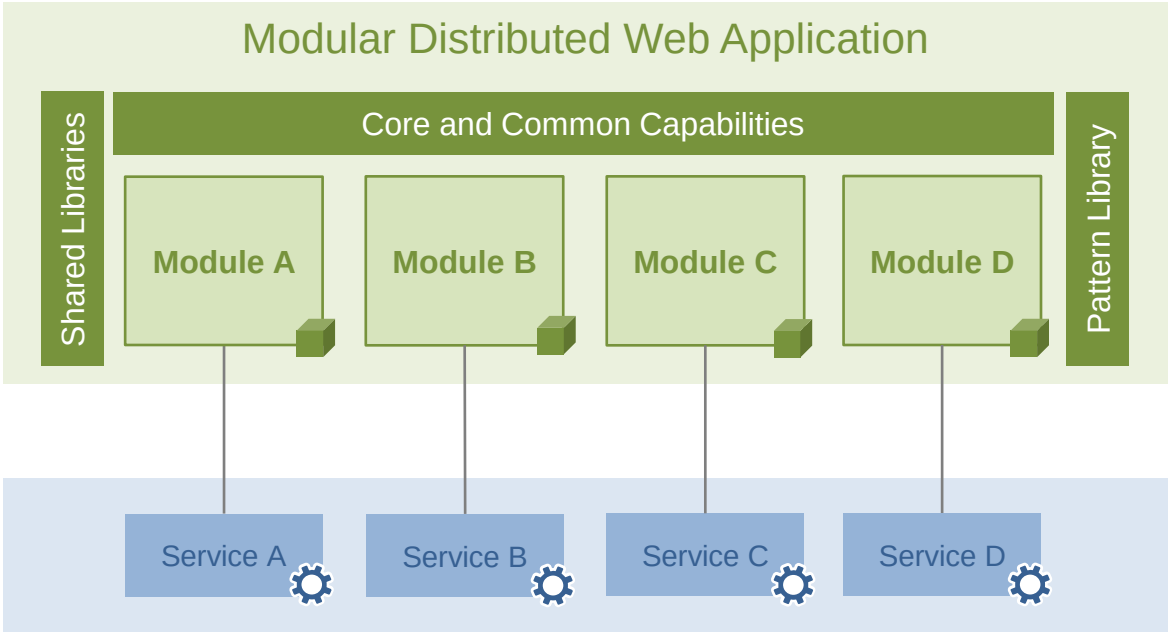
Loose coupling with **dynamic loading**

Shared **architecture foundation**

Consistent **UI & UX**

Development by **independent teams**

Proposed Frontend Architecture



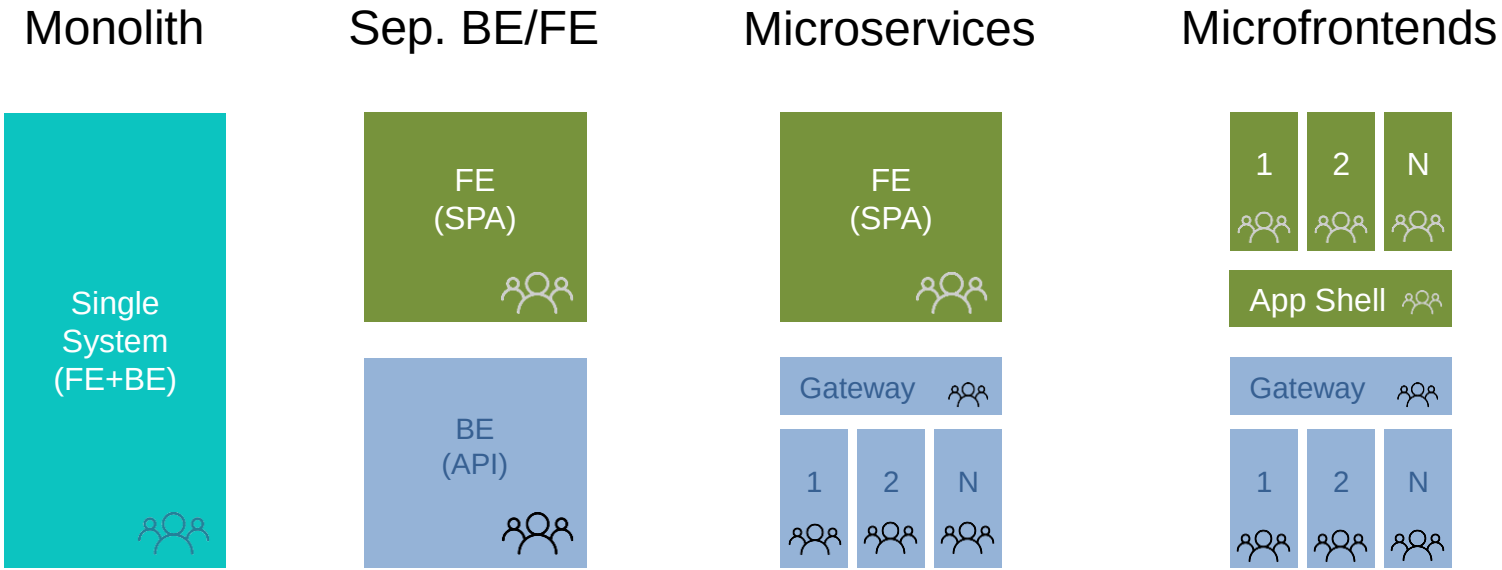
Frontend

- App Shell Component
- Business Capability Module

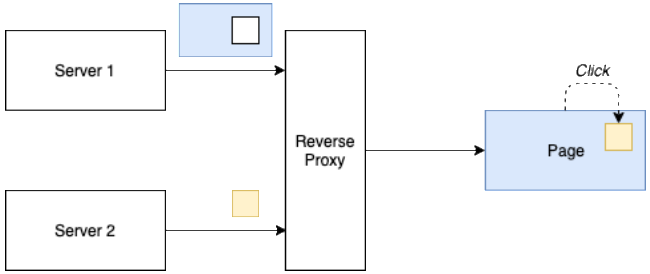
Backend

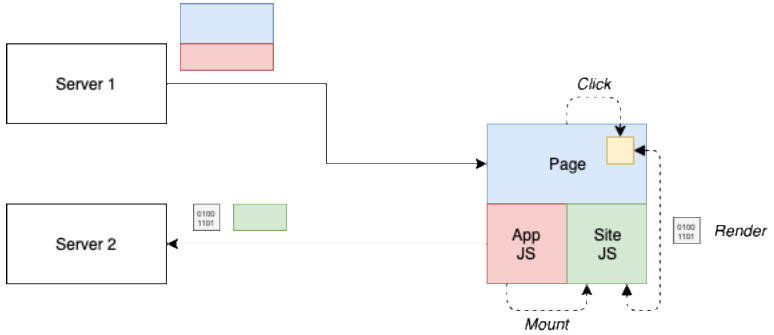
- Microservice

Natural Web App Evolution



Implementation Patterns

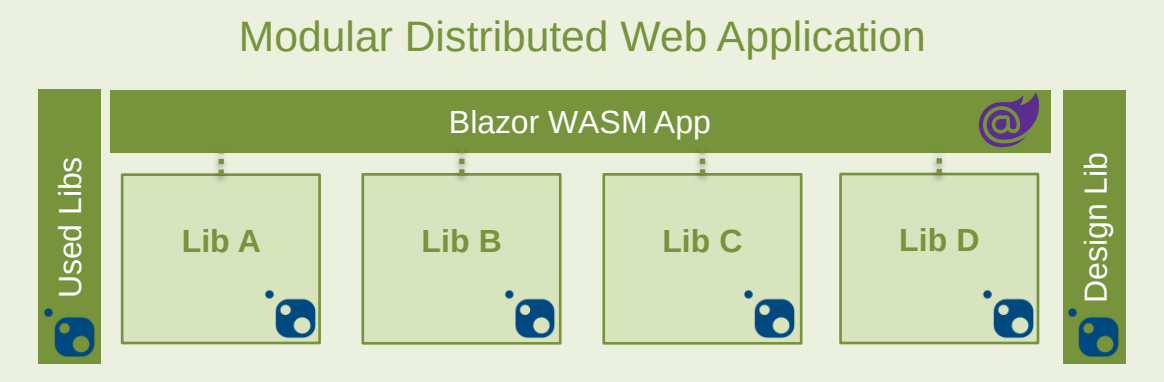




https://blog.bitsrc.io/6-patterns-for-microfrontends-347ae0017ec0

Options in Blazor (WASM)

Use NuGet Packages



Use NuGet Packages

- ✅ Stays in the flow
- ✅ C# all the way

- ❌ Requires recompilation
- ❌ No cross-framework support
- ❌ Needs full evaluation before starting

Demo
Blazor Distributed

Combine with JS

- ✅ Recombine freely
- ✅ Allows using other frameworks
- ✅ Enables lazy loading of Blazor code

- ❌ Difficult to publish correctly
- ❌ Requires JS and C# parts

Challenges
Blazor Microfrontend Solution

Challenge 1:
Routing

Challenge 2:
DOM Projection

Challenge 3:
Dynamic Registration

Challenge 4:
Using Shared Components

Challenge 5:
Lazy Loading

Reference Implementation

a framework for modular distributed Microfrontends

https://piral.io

https://docs.piral.io

https://github.com/smapiot/piral

Principles and Challenges

First class development experience
"setup of local dev environment in minutes"

Supporting Backend Services
e.g. feed for dynamic loading of modules

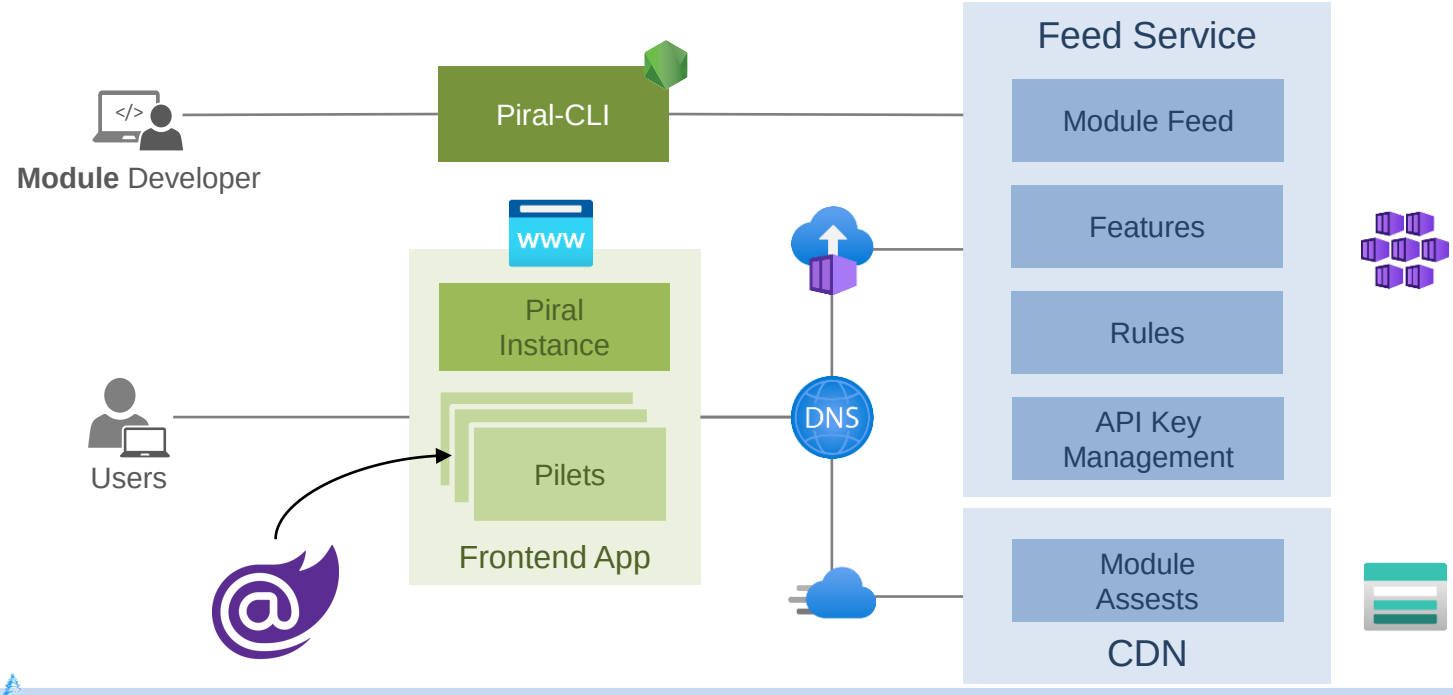
Modular Frontend Solution

Comprehensive development tooling
e.g. scaffolding modules based on templates

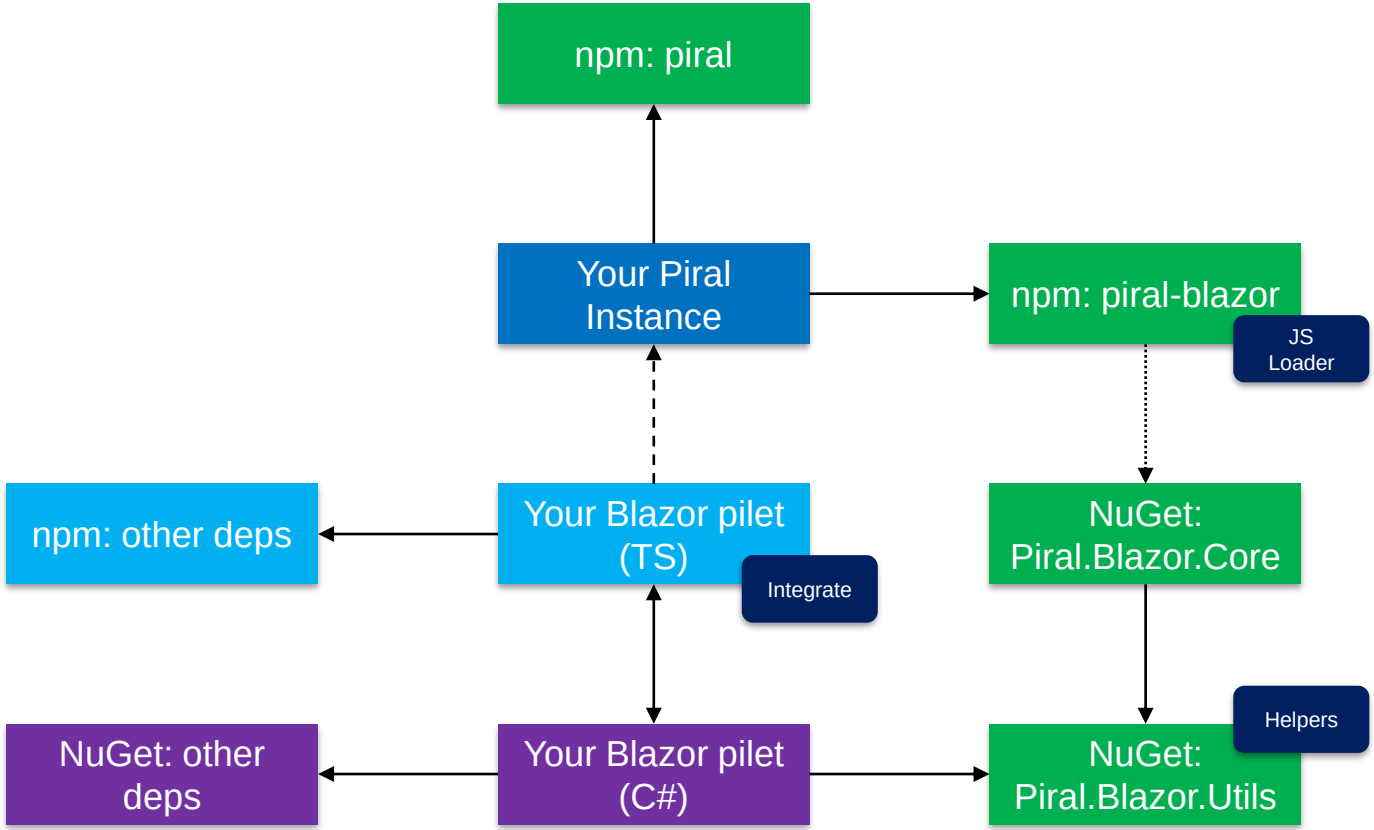
Support for multiple frameworks
e.g. React, Angular, Vue

Distributed development of modules

High Level Architecture with Piral



Demo
App Shell Development

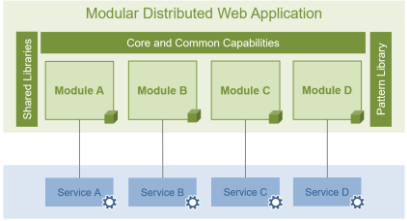

npm: piral
Your Piral Instance
npm: piral-blazor
JS Loader
npm: other deps
Your Blazor pilet (TS)
Integrate
NuGet: Piral.Blazor.Core
Helpers
NuGet: other deps
Your Blazor pilet (C#)
NuGet: Piral.Blazor.Utils

Demo
Microfrontend Development

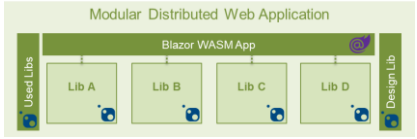
Contribution Areas

- 🚀 Star ☆ Piral
- 🚀 Debugging
- 🚀 Convenience
- 🚀 Templating
- 🚀 Reverse Tree Shaking

Conclusion



Micro Frontend Architecture



Architecture for Distributed Blazor

Framework for Modular Microfrontends

Thank you!

smapiot
smapiot.com
github.com/smapiot

Florian Rappl
florian.rappl@smapiot.com
@FlorianRappl

Piral
piral.io
docs.piral.io

Further Reading

Popular Misconceptions About Microfrontends

https://dev.to/florianrappl/11-popular-misconceptions-about-micro-frontends-463p

Reasons for Doing Microfrontends

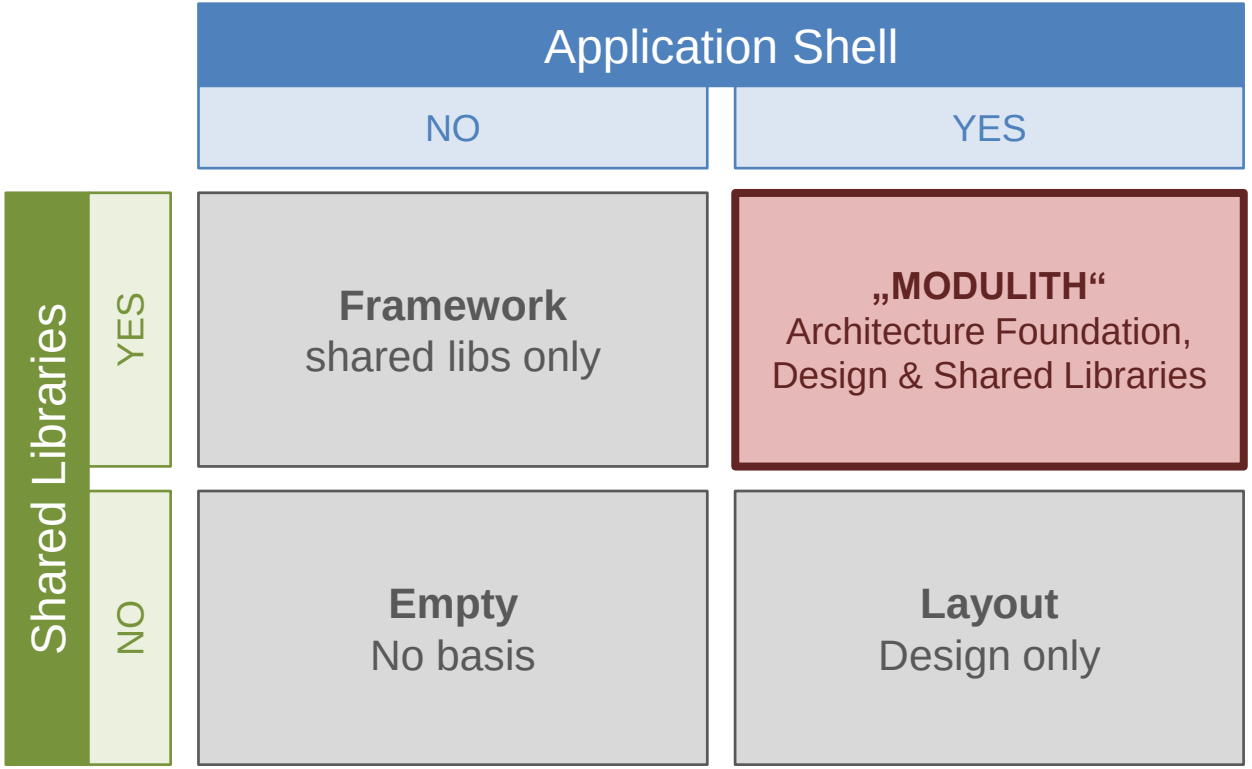
https://dev.to/florianrappl/5-reasons-for-doing-microfrontends-1mba

Introduction to Microfrontends with piral

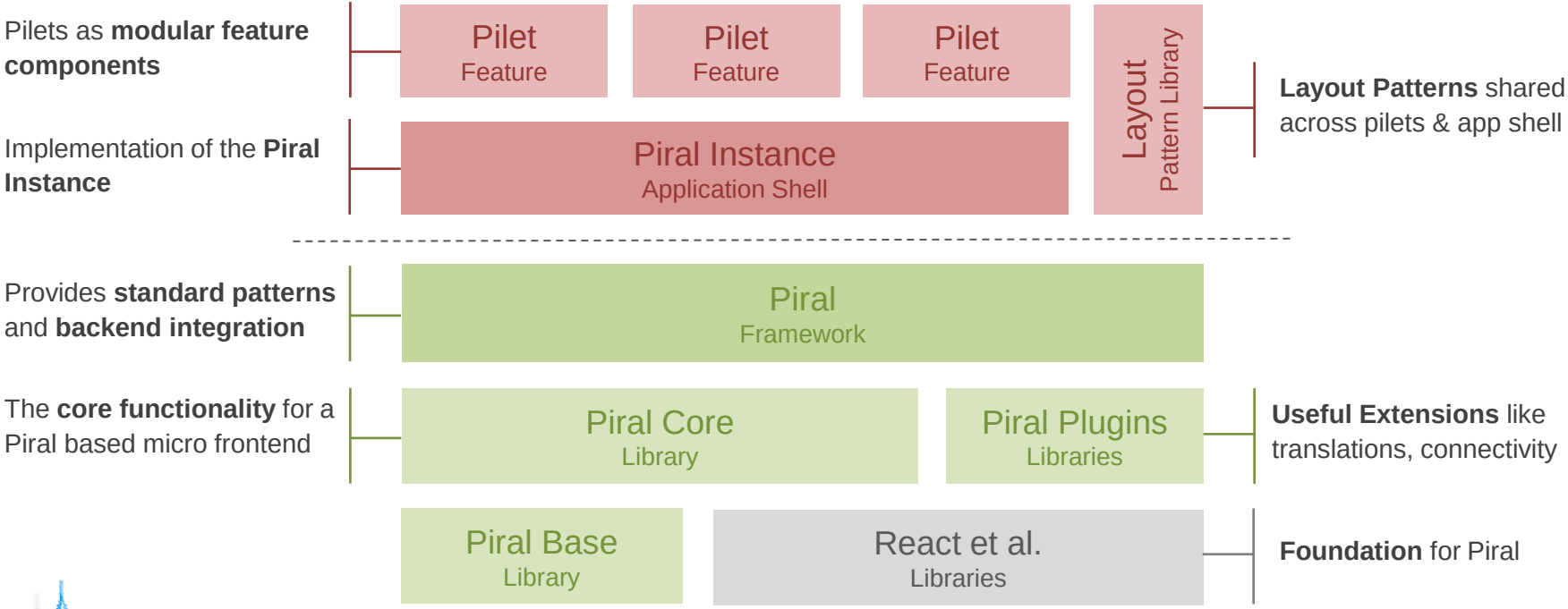
https://dev.to/florianrappl/introduction-to-microfrontends-with-piral-4mpp

Microfrontends with React

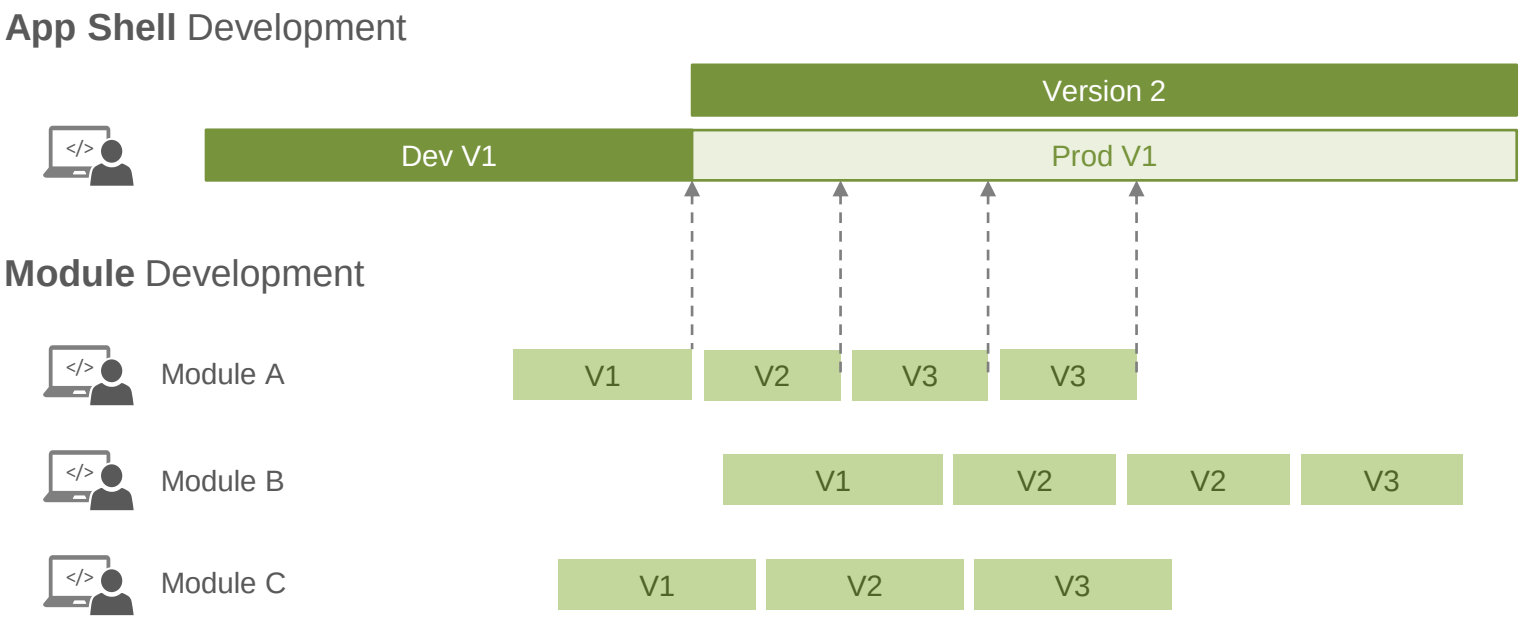
https://dev.to/florianrappl/microfrontends-based-on-react-4oo9


Application Shell
NO
YES
Shared Libraries
YES
NO
Framework
shared libs only
„MODULITH“
Architecture Foundation,
Design & Shared Libraries
Empty
No basis
Layout
Design only

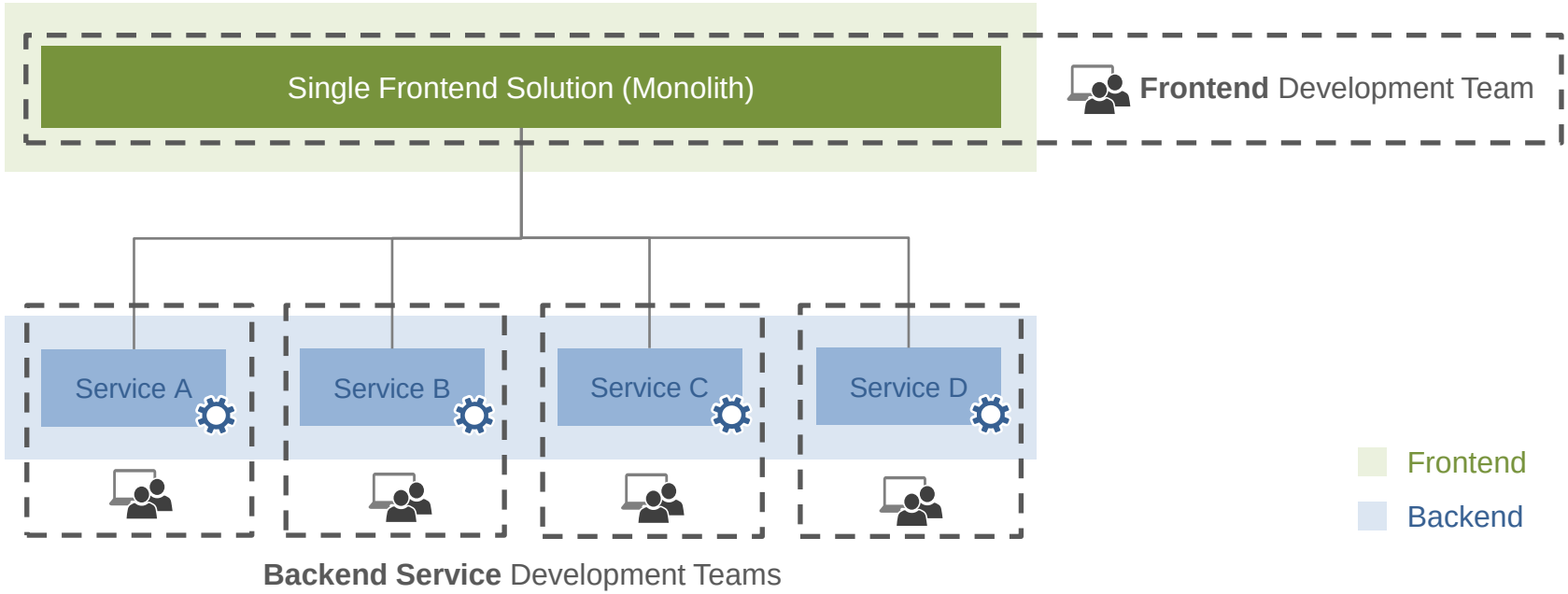
Components of a Piral Frontend



Development Lifecycles



Development Approach



Development Approach