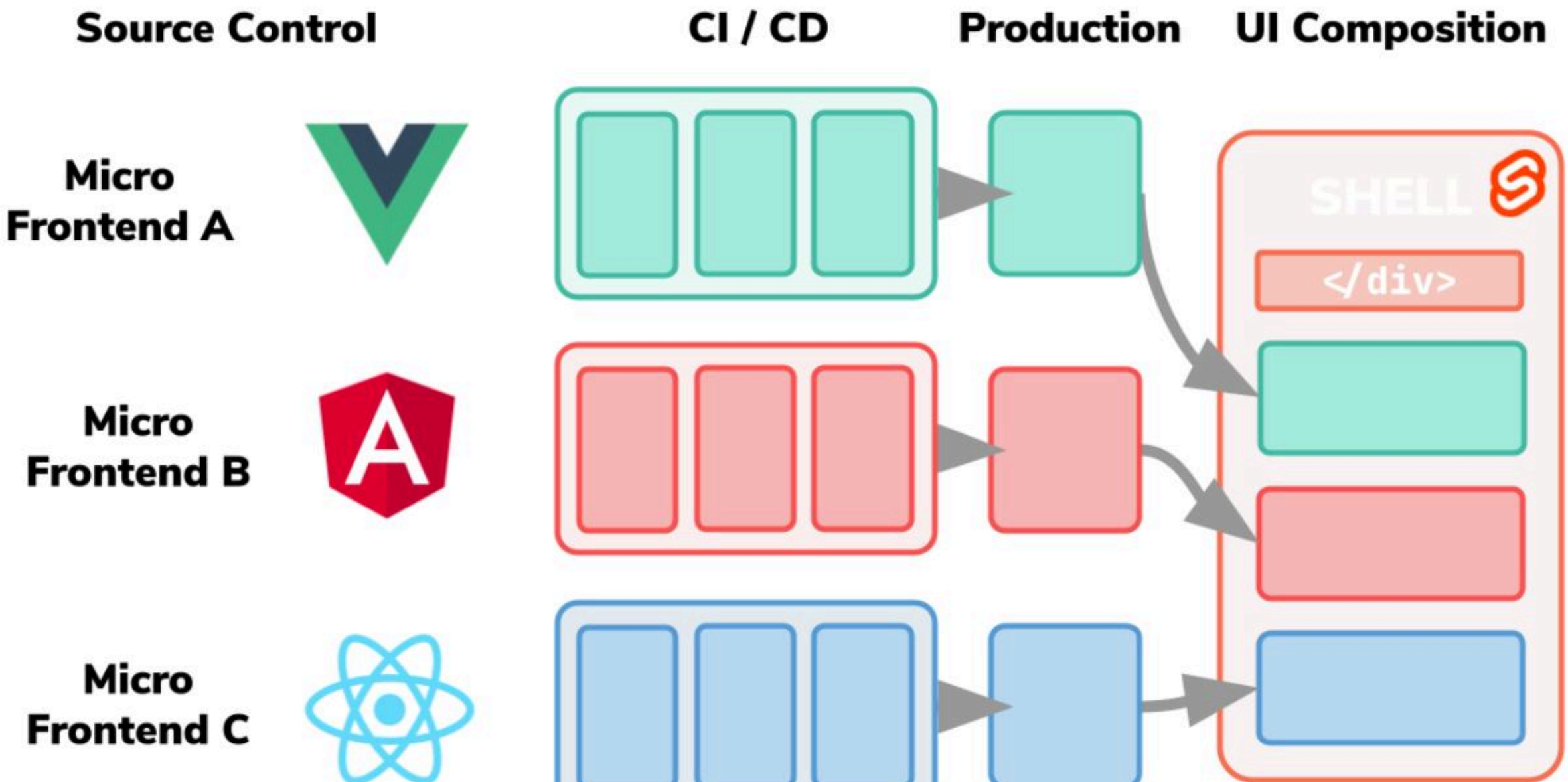
in/poyrazavsever

MIKRO DA MIKRO

Mikro-Frontend Nedir?



in/poyrazavsever

Mikro-Frontend Nedir?

Mikro-Frontend, büyük bir frontend uygulamasını küçük ve bağımsız parçalara ayıran mimari yaklaşımdır.

Ben bu yaklaşımı ilk olarak **Akbank'da** çalışan bir frontend developer'dan duymuştum.

Her parça, kendi yaşam döngüsüne, kod tabanına ve ekip sahipliğine sahiptir.

in/poyrazavsever

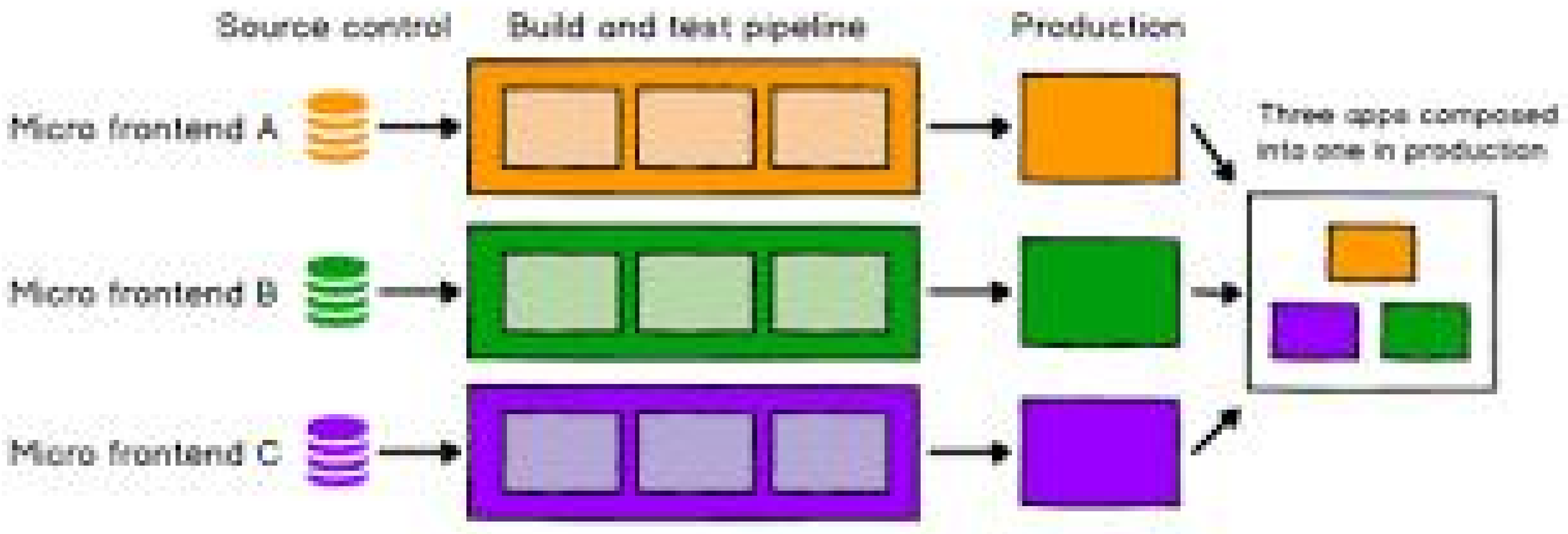
Burada Amacımız

Ekiplerin birbirinden bağımsız çalışabilmesini, deploy süreçlerinin izole olmasını ve uygulamanın ölçeklenebilir hale gelmesini sağlamaktır.

in/poyrazavsever

Felsefemiz

“Microservices for frontend” prensibiyle, her modül kendi kararlarını verir ama ortak bir kullanıcı deneyimi sunar.



in/poyrazavsever

Temel Prensip

“Bağımsız geliştir, entegre etmeden önce izole test et.”

in/poyrazavsever

Peki neden? Monolit Sorunu Nedir?

Tek kod tabanı, tek build pipeline,
ve en ufak değişiklikte devasa deploy’lar...

Monolitik yapılar büyüdükçe hantallaşır.

in/poyrazavsever

Madem Öyle Dönüştürelim

Mikro-Frontend mimarisiyle her modül ayrı build edilir, CI/CD süreçleri izole hale gelir.

```
Monolitik: app.js (1 takım)
Mikro-Frontend: dashboard, auth, settings (3 takım)
```

in/poyrazavsever

Geçişte Dikkat Edilecekler

- Routing yeniden tasarlanmalı
- State yönetimi izole olmalı
- Ortak API sözleşmeleri tanımlanmalı

in/poyrazavsever

Özet olarak

Mikro-Frontend sadece teknik değil,
aynı zamanda organizasyonel bir dönüşümdür.

in/poyrazavsever

Ortak UI Kit Paylaşımı Neden Gerekli?

Her micro-app kendi UI bileşenlerini üretebilir.

Ama görünümün bütünlüğü için ortak bir UI Kit şarttır.

in/poyrazavsever

Design Token Mantığı

Renk, tipografi, spacing gibi değerler
tek bir kaynakta tutulur – token yapısı diyoruz buna.

in/poyrazavsever

Klasör Yapısını Ben Şöyle Yapıyorum

```
ui-kit/
  ├─ components/
  ├─ tokens/
  ├─ theme/
  └─ index.ts
```

in/poyrazavsever

Paylaşım Yöntemleri

UI Kit bir npm paketi,

Storybook repo’su

veya private registry olabilir.

in/poyrazavsever

Burada

“Her micro-app farklı olabilir,

ama UI dili ortak kalmalı.”

in/poyrazavsever

Micro-Frontend + Monorepo Yapısı Sorunu

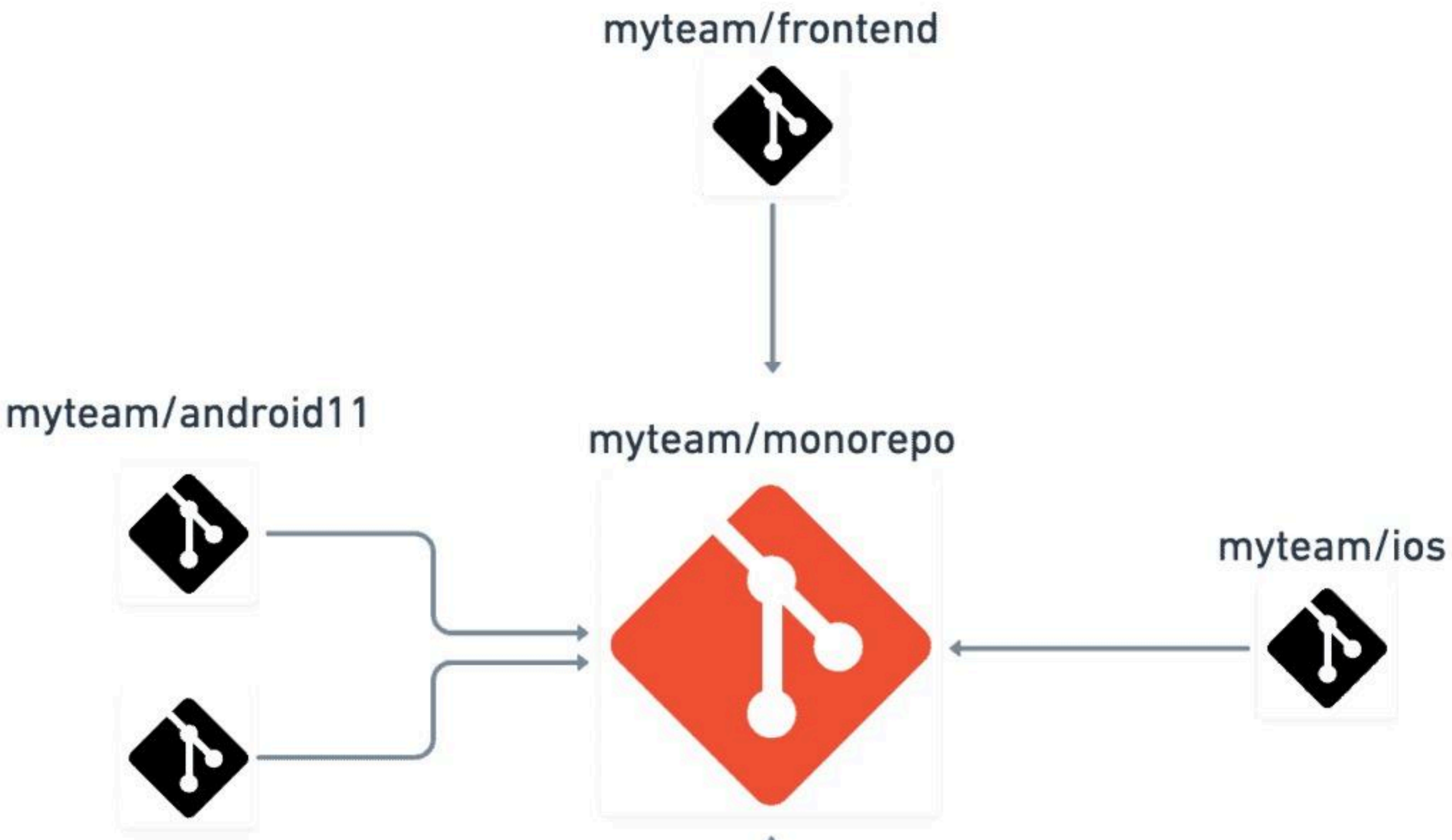
Çok sayıda repo, versiyon karmaşası,
bağımlılık takibi oldukça zor.

in/poyrazavsever

Monorepo Çözümü

Tüm micro-app’ler tek repoda toplanır.

Build bağımsız – kod yönetimi merkezi.



in/poyrazavsever

Yapımız İse

Tüm micro-app’ler tek repoda toplanır.

Build bağımsız – kod yönetimi merkezi.

```
apps/
  ├─ dashboard/
  ├─ auth/
  ├─ settings/
packages/
  ├─ ui-kit/
  ├─ utils/
  └─ config/
```

in/poyrazavsever

Aşağıdaki Araçları Kullanabilirsiniz

Nx, Turborepo ve Lerna bu yapıyı yönetmek için kullanılır.

Lerna

Özet Olarak Kuralımız

“Merkezi repo, bağımsız build.
Dengeyi koru.”

Sen ne düşünüyorsun?

in/poyrazavsever