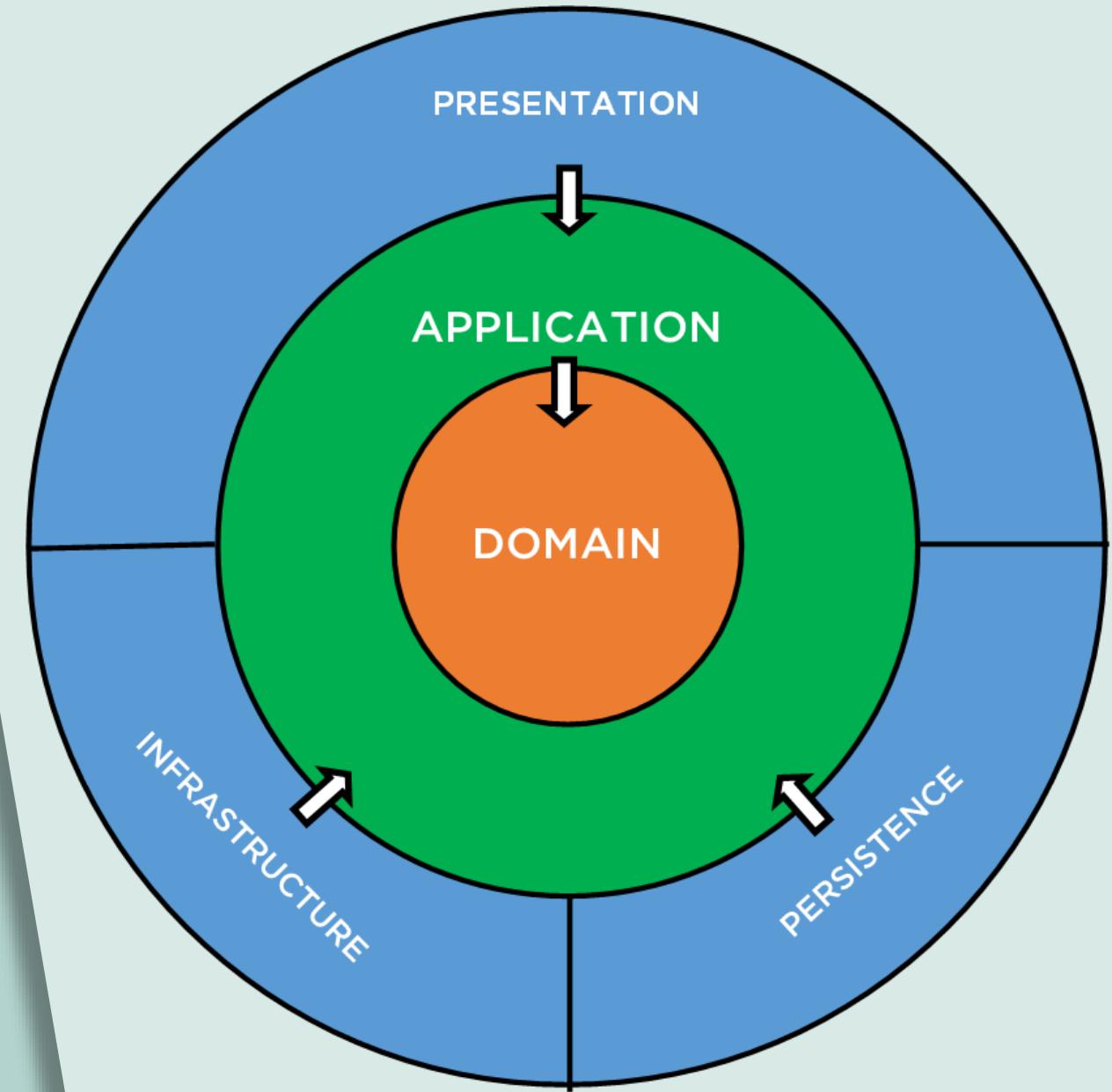


# Uvod u softversko inženjerstvo

Čisti paketi  
i arhitektura

Nikola Luburić  
[nikola.luburic@uns.ac.rs](mailto:nikola.luburic@uns.ac.rs)



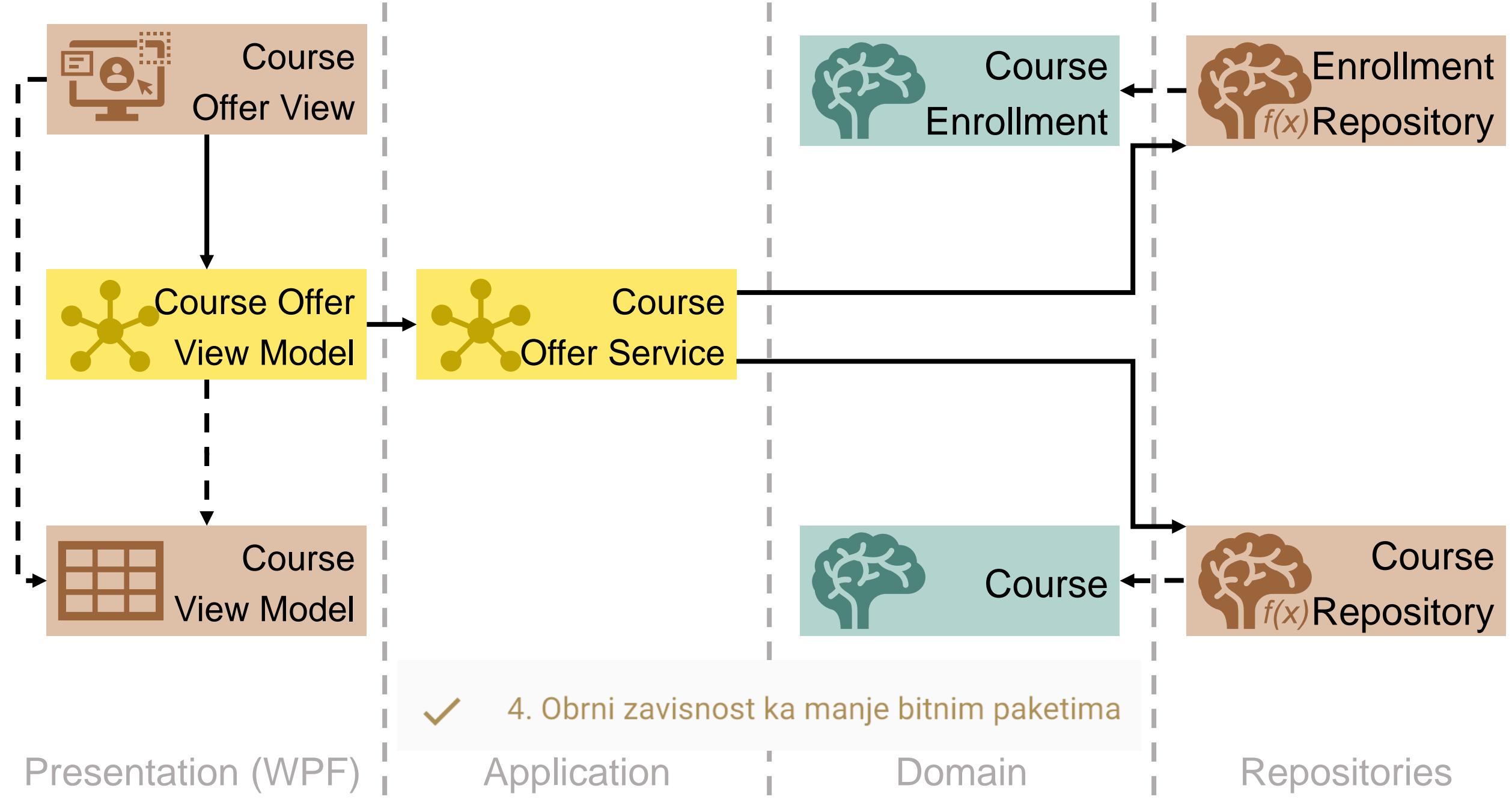
Čisti paketi  
i arhitektura

MVVM paketiranje

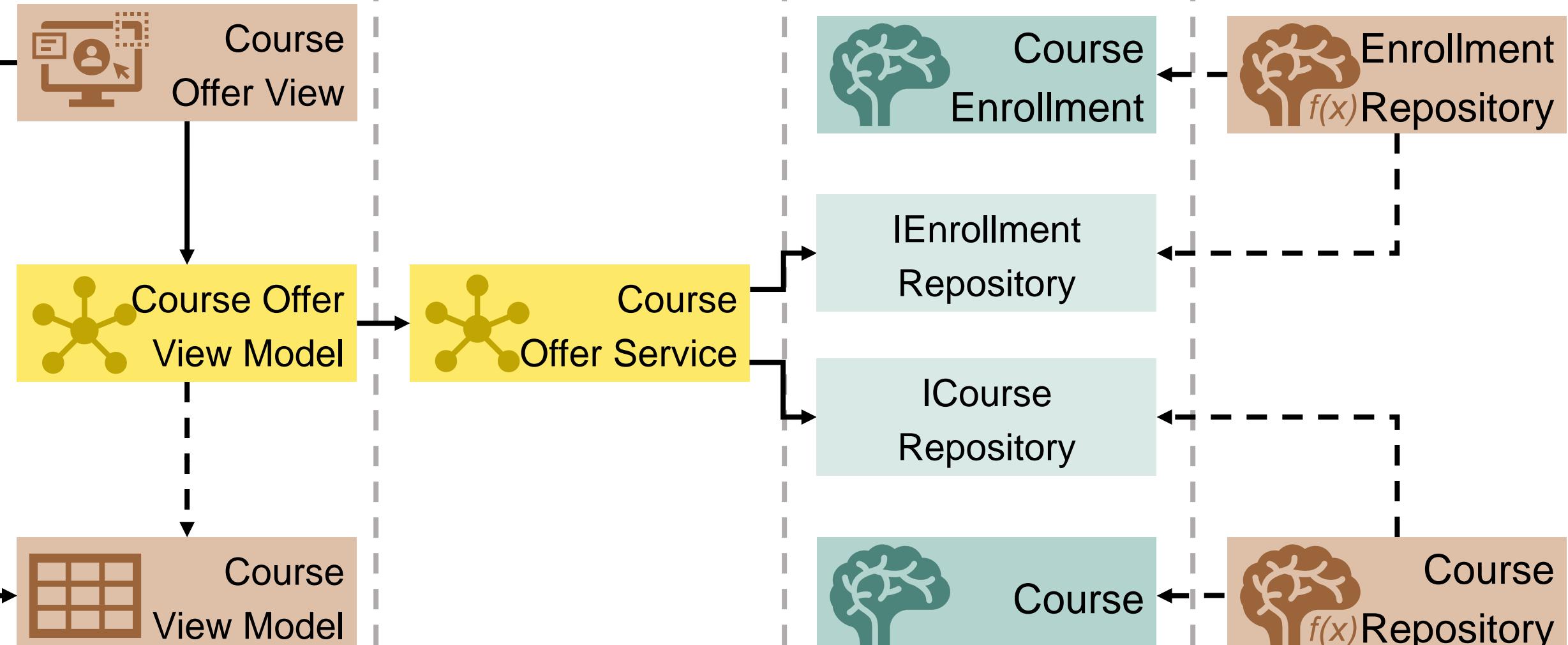
Princip inverzije zavisnosti

Čista arhitektura

Učenik vidi spisak kurseva na koje može da se prijavi u narednom periodu.



Učenik vidi spisak kurseva na koje može da se prijavi u narednom periodu.



✓ 4. Obrni zavisnost ka manje bitnim paketima

Presentation (WPF)

Application

Domain

Repositories

SRP

OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,  
implementacije zavise od apstrakcija



# Apstrakcije ne zavise od implementacije, implementacije zavise od apstrakcija

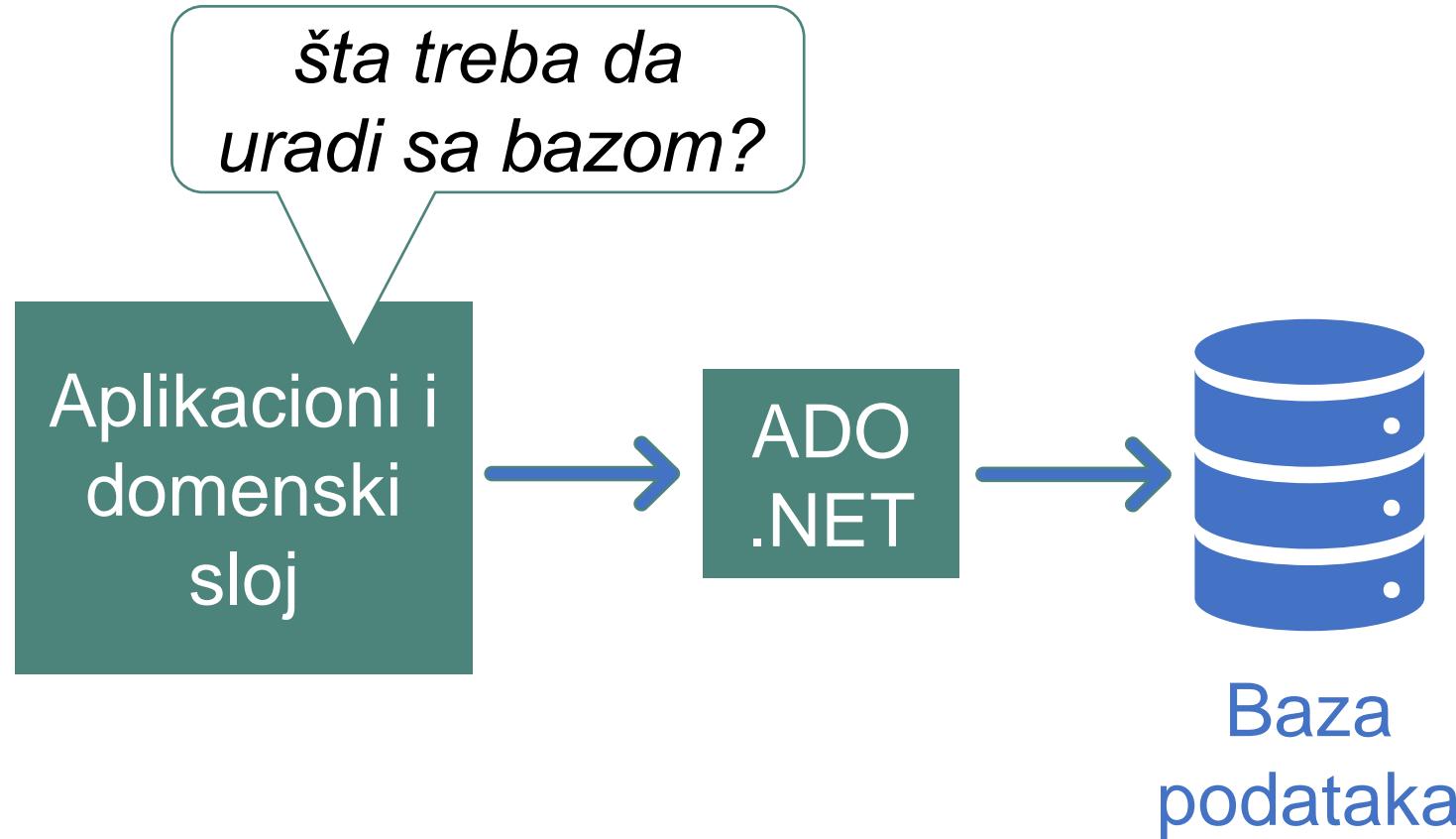
SRP

OCP

LSP

ISP

DIP



SRP

OCP

LSP

ISP

DIP

## Apstrakcije ne zavise od implementacije, implementacije zavise od apstrakcija

```
class EmployeeReporter {  
    void ReportEmployees() {  
        List<Employee> ems = new List<Employee>();  
        String query = "select * from Employees";  
        try {  
            using (SqlCommand cmd = new SqlCommand(query, con)) {  
                con.Open();  
                using (SqlDataReader reader = cmd.ExecuteReader()) {  
                    while (reader.Read()) ems.Add(CompileEmployee(reader));  
                }  
                con.Close();  
            }  
            PrintEmployees(ems); // ...  
        }
```

SRP

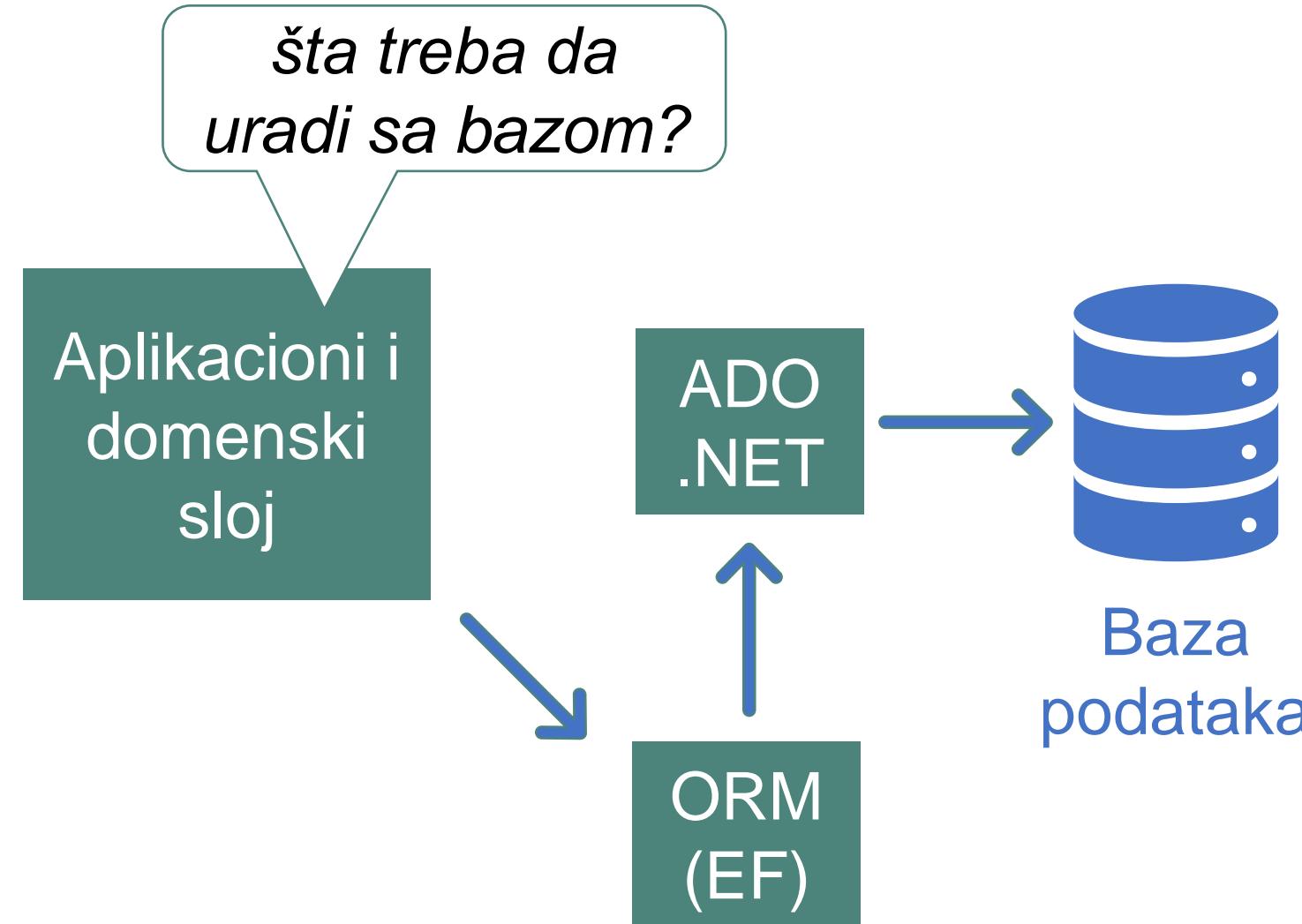
OCP

LSP

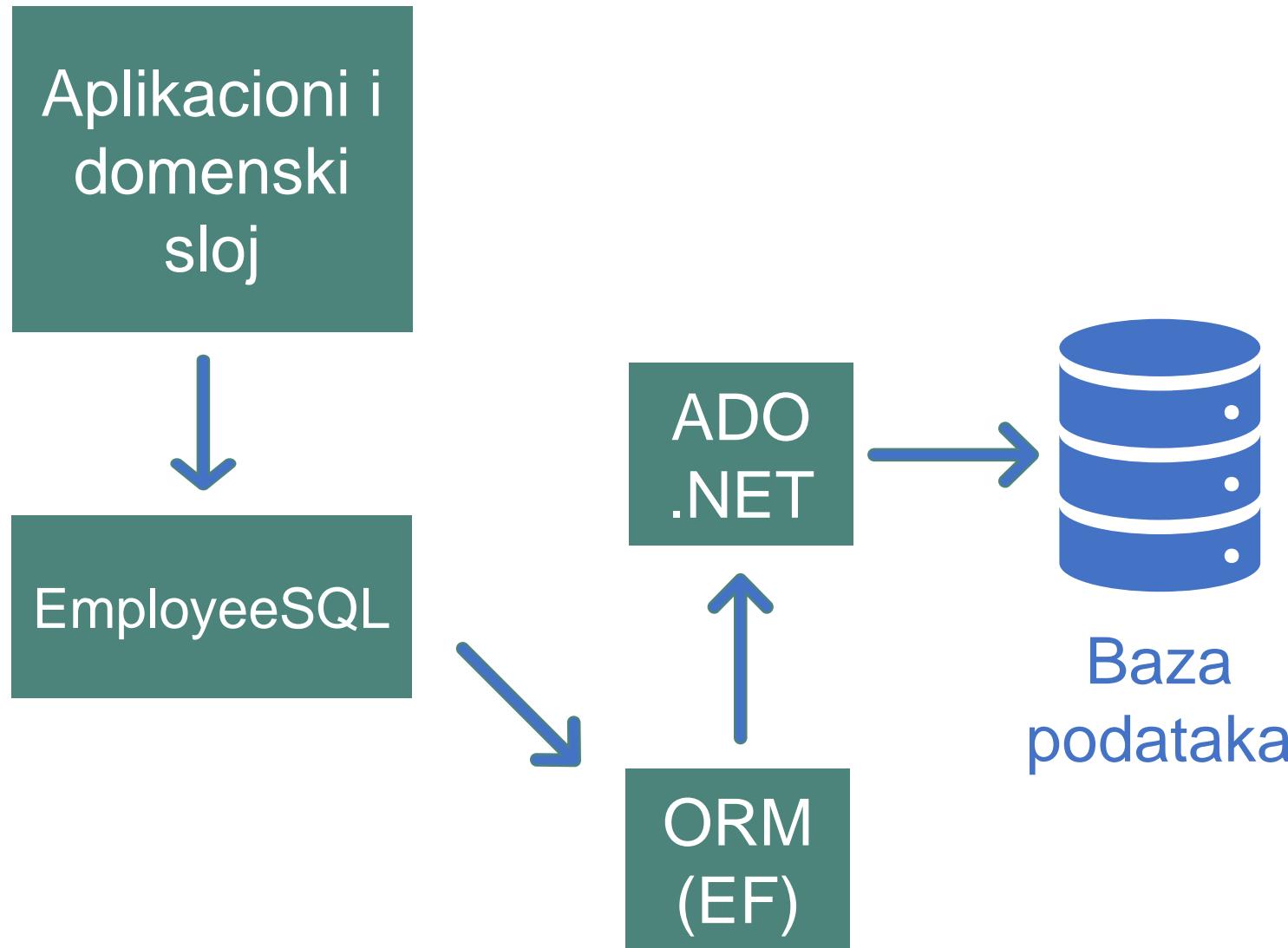
ISP

DIP

Apstrakcije ne zavise od implementacije,  
implementacije zavise od apstrakcija



# Apstrakcije ne zavise od implementacije, implementacije zavise od apstrakcija



SRP

OCP

LSP

ISP

DIP

SRP

OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,  
implementacije zavise od apstrakcija

```
class EmployeeReporter {  
    void reportEmployees() {  
        EmployeeDB edb = new EmployeeDB();  
        printEmployees(edb.getEmployees());  
    }  
}
```

*Znak da se  
DIP ne prati*

*da li Reporter  
zavisi od DB?*

SRP

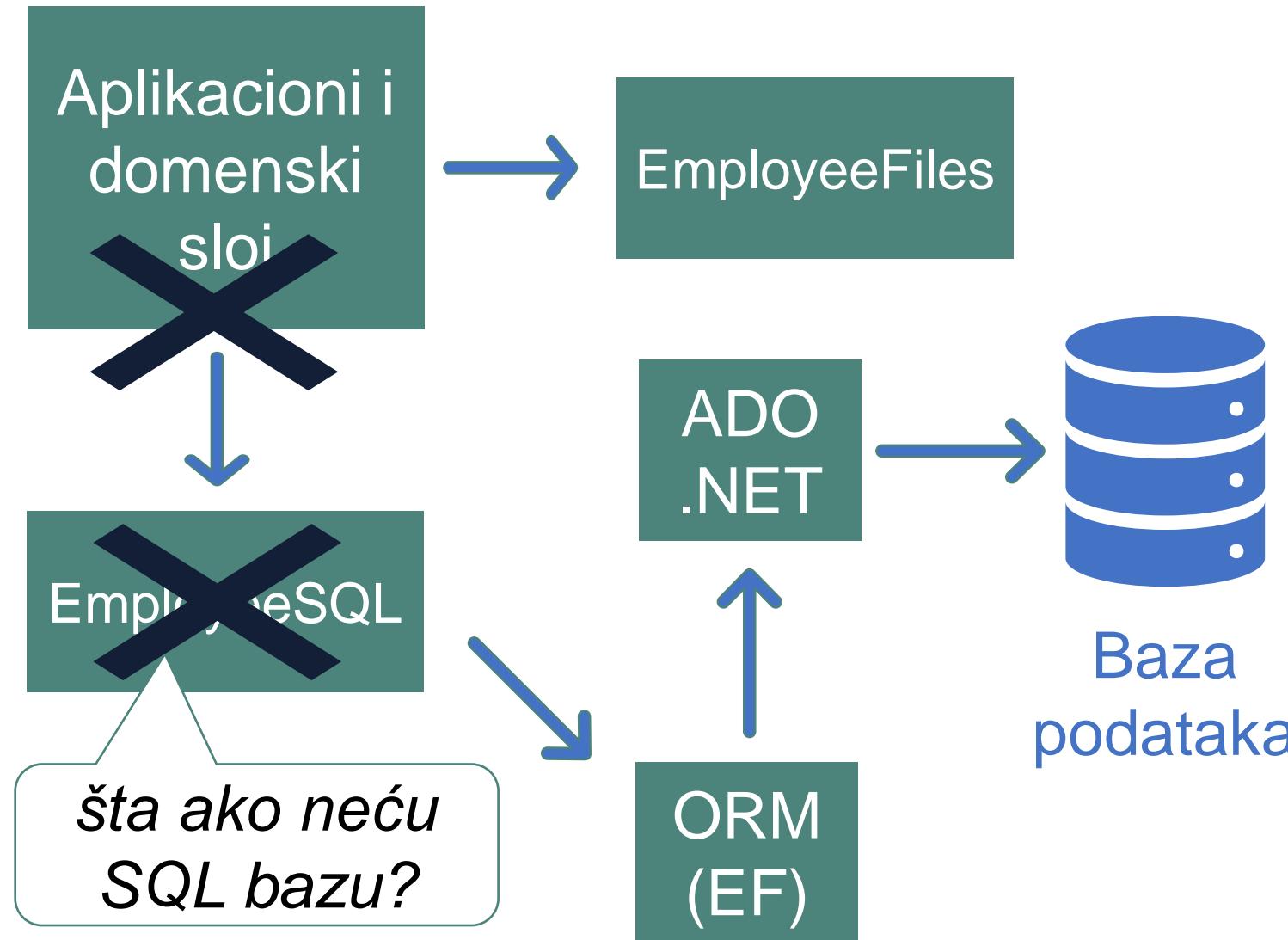
OCP

LSP

ISP

DIP

# Apstrakcije ne zavise od implementacije, implementacije zavise od apstrakcija



SRP

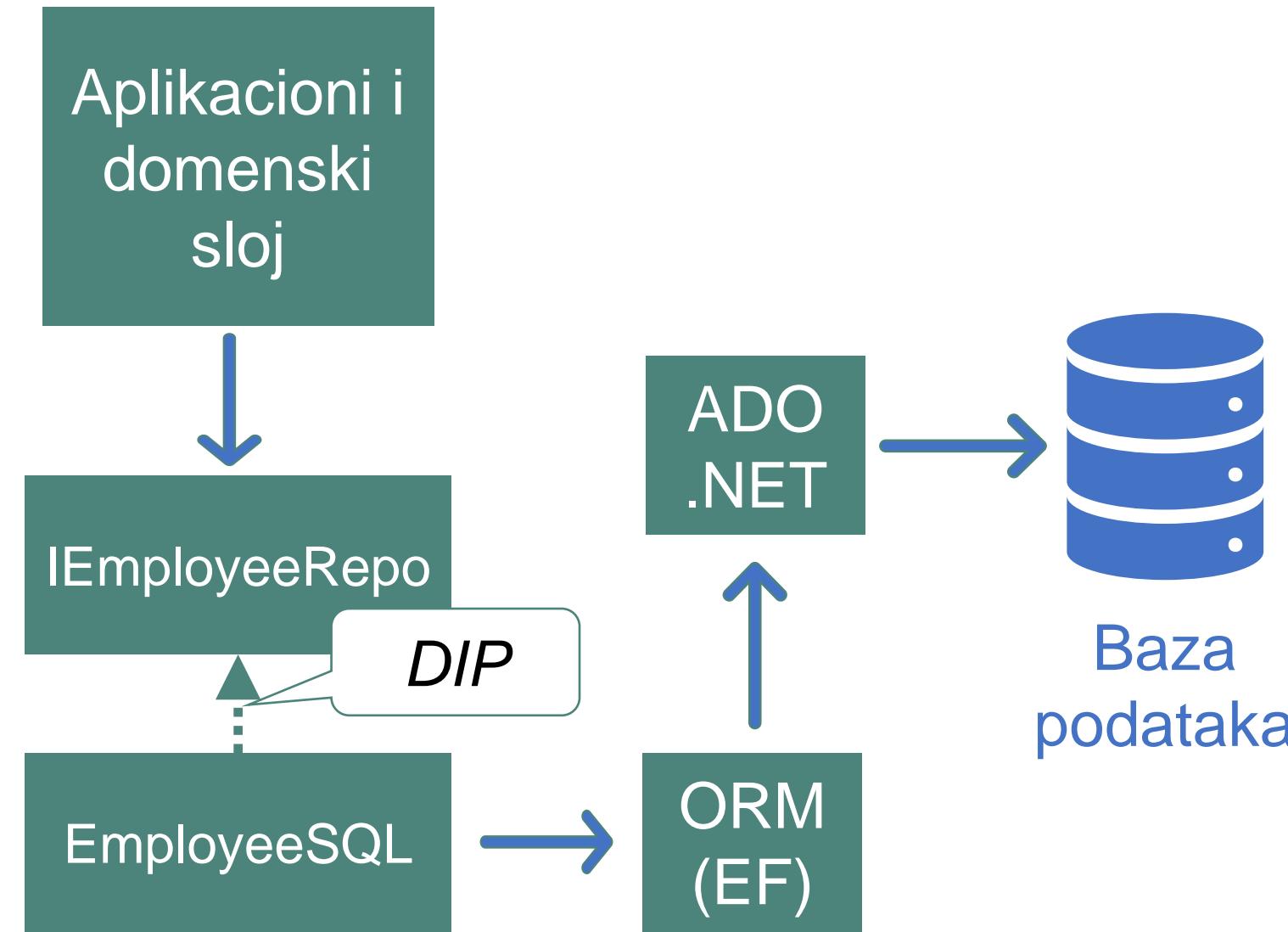
OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,  
implementacije zavise od apstrakcija



Apstrakcije ne zavise od implementacije,  
implementacije zavise od apstrakcija

SRP

OCP

LSP

ISP

DIP

```
class EmployeeReporter {  
    void reportEmployees(IEmployeeRepo er) {  
        printEmployees(er.getEmployees());  
    }  
}
```

Apstrakcije ne zavise od implementacije,  
implementacije zavise od apstrakcija

SRP

OCP

LSP

ISP

DIP

Aplikacioni i  
domenski  
sloj

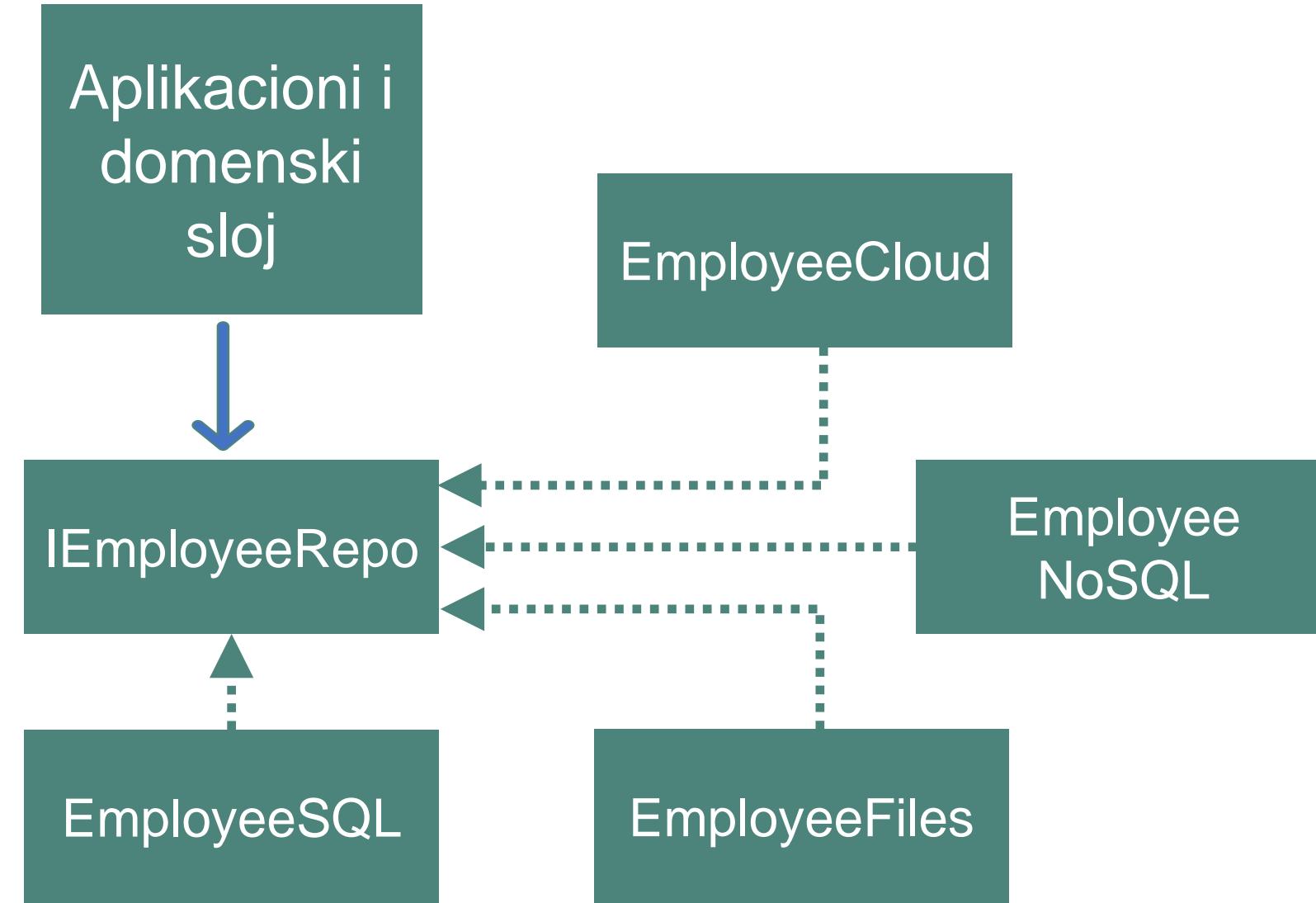
EmployeeCloud

IEmployeeRepo

Employee  
NoSQL

EmployeeSQL

EmployeeFiles



SRP

OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,  
implementacije zavise od apstrakcija

### *Dependency injection*

- ❖ Šablon koji pomaže da se ostvari *DIP*
- ❖ Opskrbi modul sa onim što mu je potrebno
- ❖ Nivo konstruktora, polja (*setter*) i funkcije

```
class EmployeeReporter {  
    void reportEmployees(IEmployeeRepo er) {  
        printEmployees(er.getEmployees());  
    }  
}
```

SRP

OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,  
implementacije zavise od apstrakcija

### *Dependency injection*

- ❖ Šablon koji pomaže da se ostvari *DIP*
- ❖ Opskrbi modul sa onim što mu je potrebno
- ❖ Nivo konstruktora, polja (*setter*) i funkcije

```
class EmployeeReporter {  
    IEmployeeRepo er;  
    public EmployeeReporter(IEmployeeRepo e) {  
        this.er = e;  
    }  
}
```

SRP

OCP

LSP

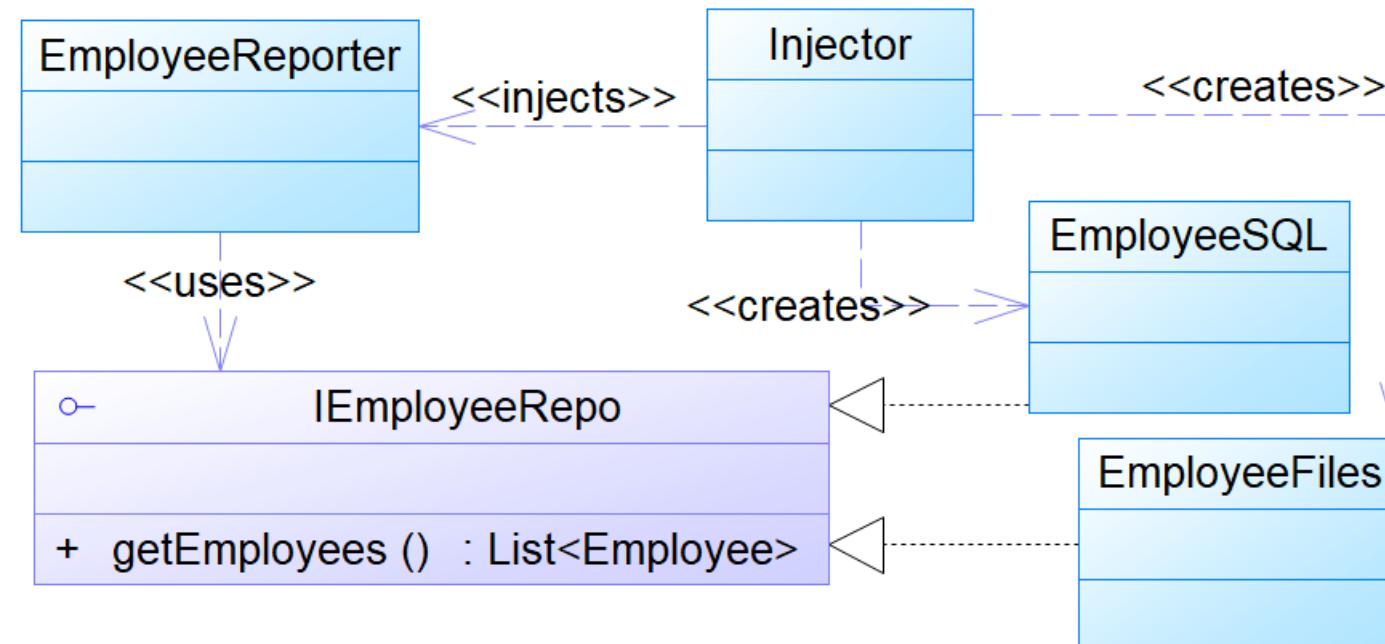
ISP

DIP

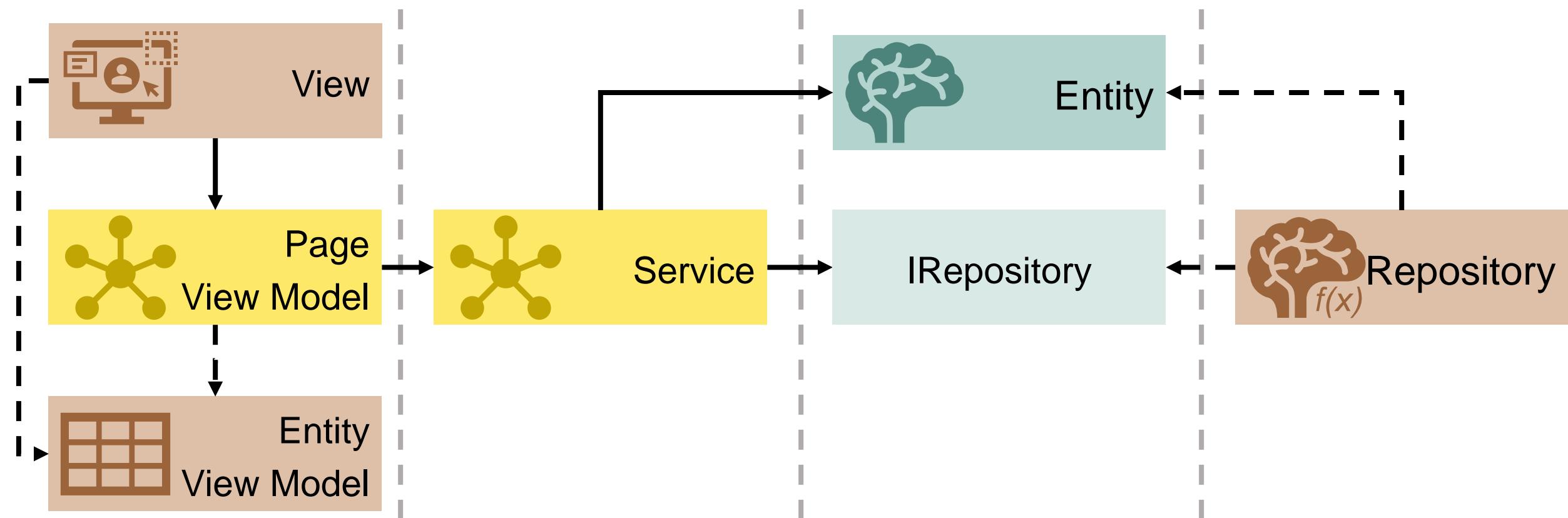
# Apstrakcije ne zavise od implementacije, implementacije zavise od apstrakcija

## *Dependency injection*

- ❖ Šablon koji pomaže da se ostvari *DIP*
- ❖ Opskrbi modul sa onim što mu je potrebno
- ❖ Nivo konstruktora, polja (*setter*) i funkcije

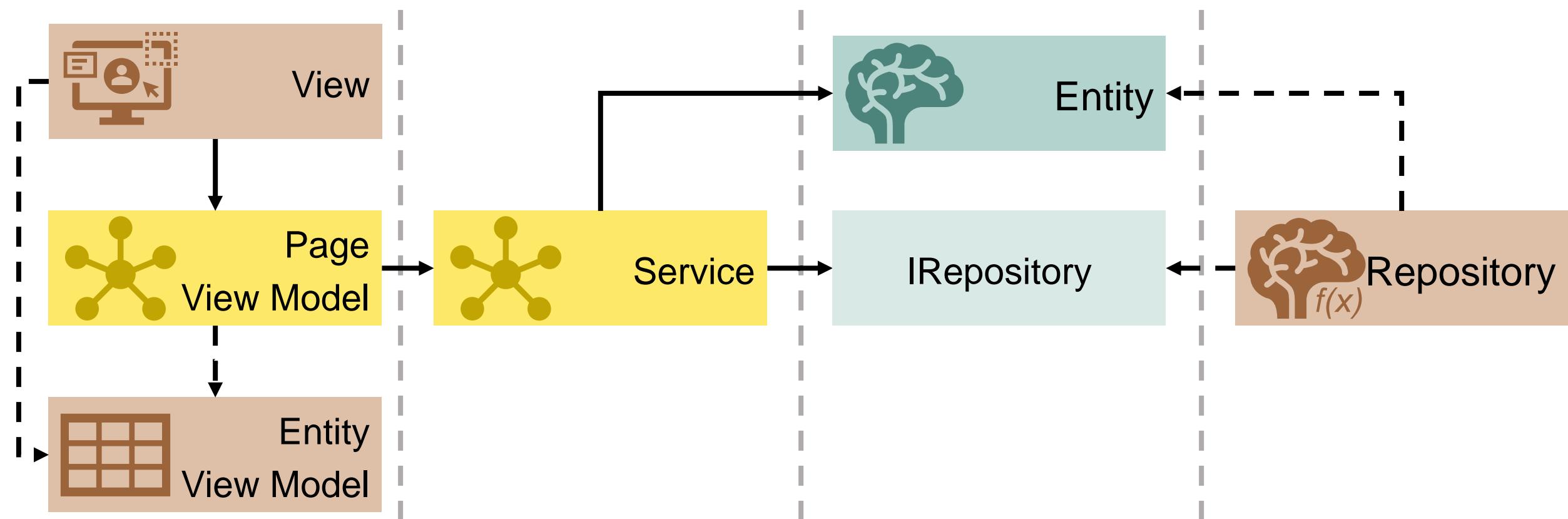


Direktor može da zatraži izveštaj koji ističe prosečan broj poena ostvaren na svakom od delova svih ispita u poslednjih godinu dana (koliko je studenata slušalo kurs, koliko položilo i procenat polaganja).

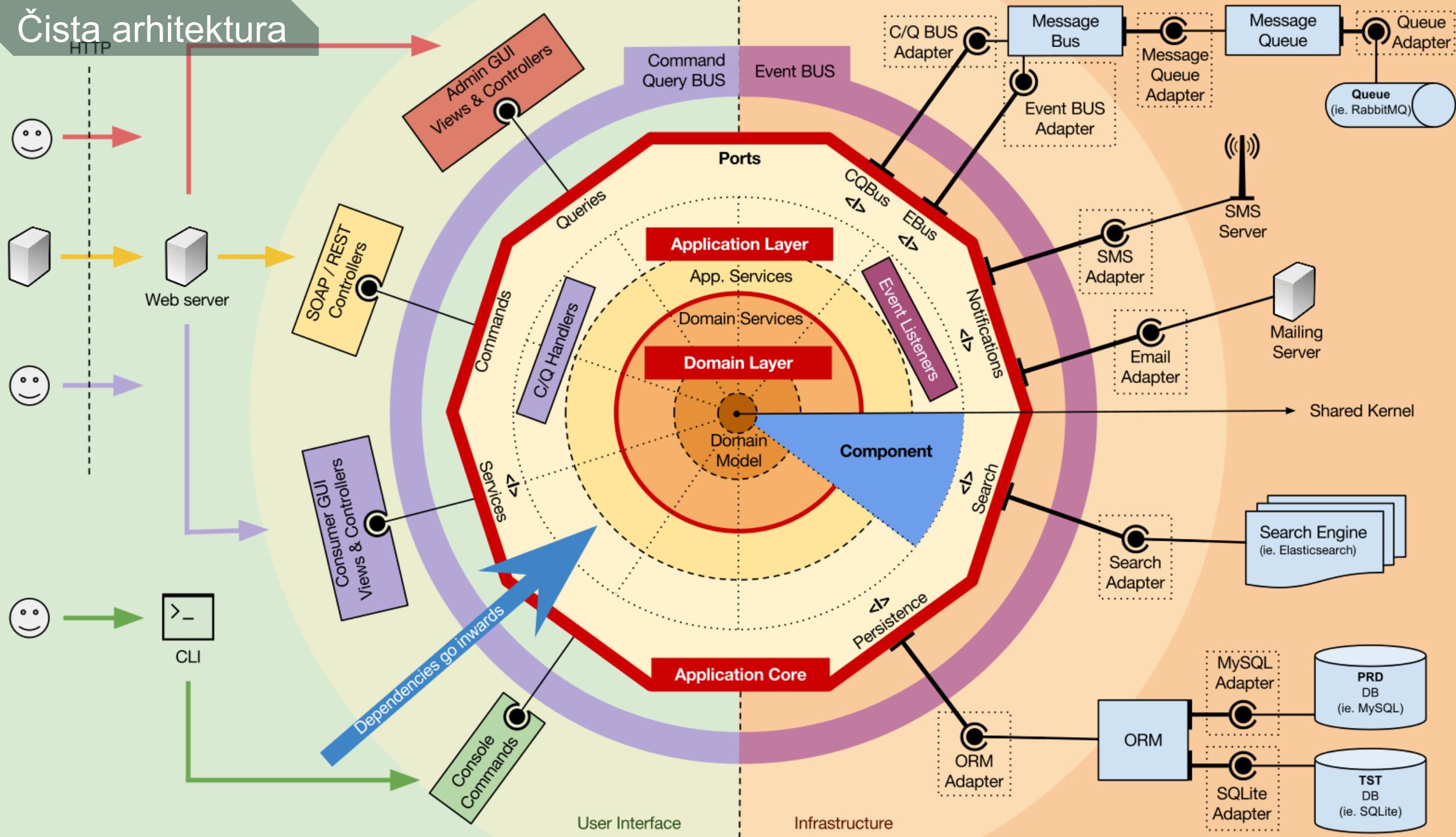


Direktor može da zatraži izveštaj koji ističe prosečan broj poena ostvaren na svakom od delova svih ispita u poslednjih godinu dana (koliko je studenata slušalo kurs, koliko položilo i procenat polaganja).

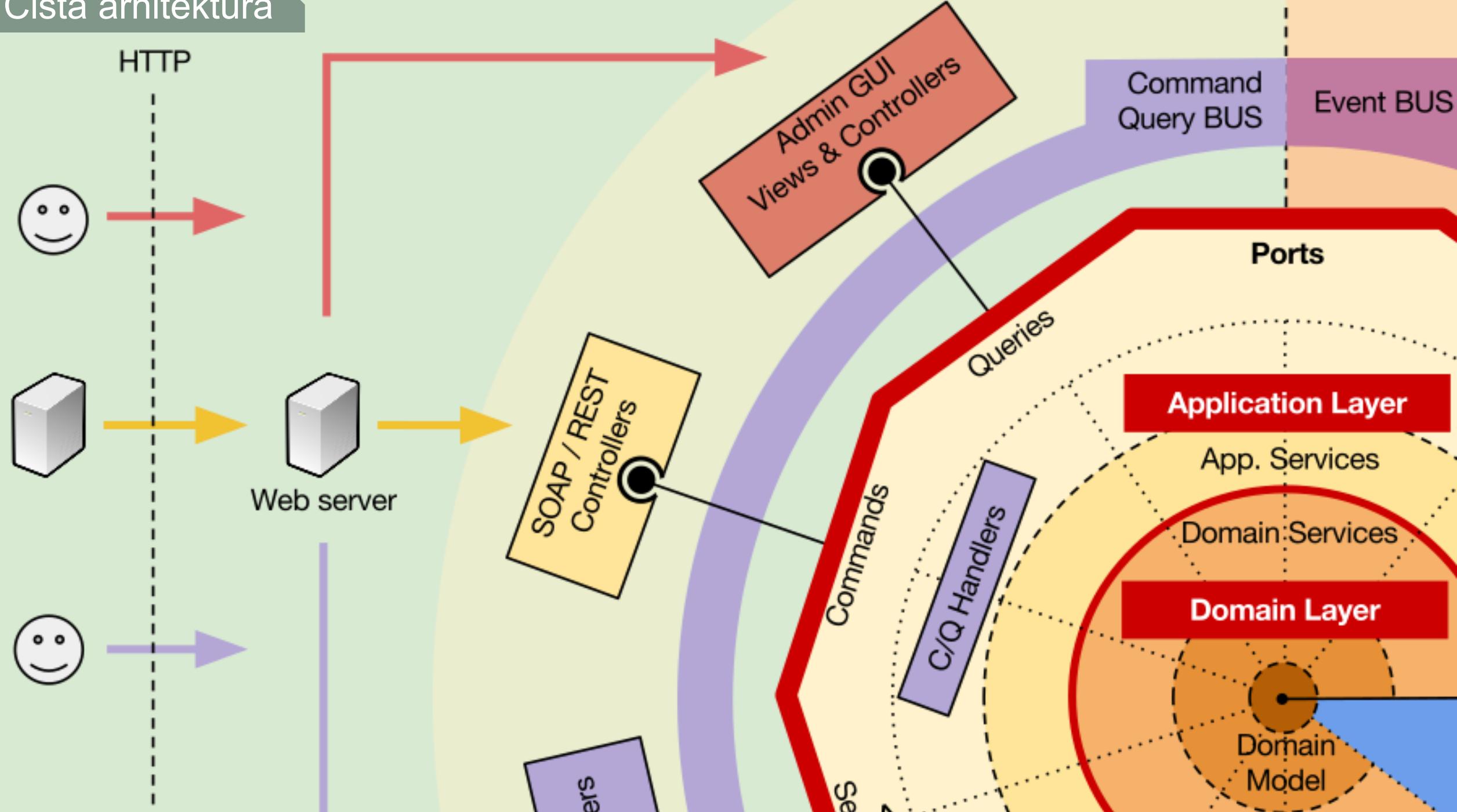
Izveštaj se generiše u PDF formatu i šalje na mejl direktora.



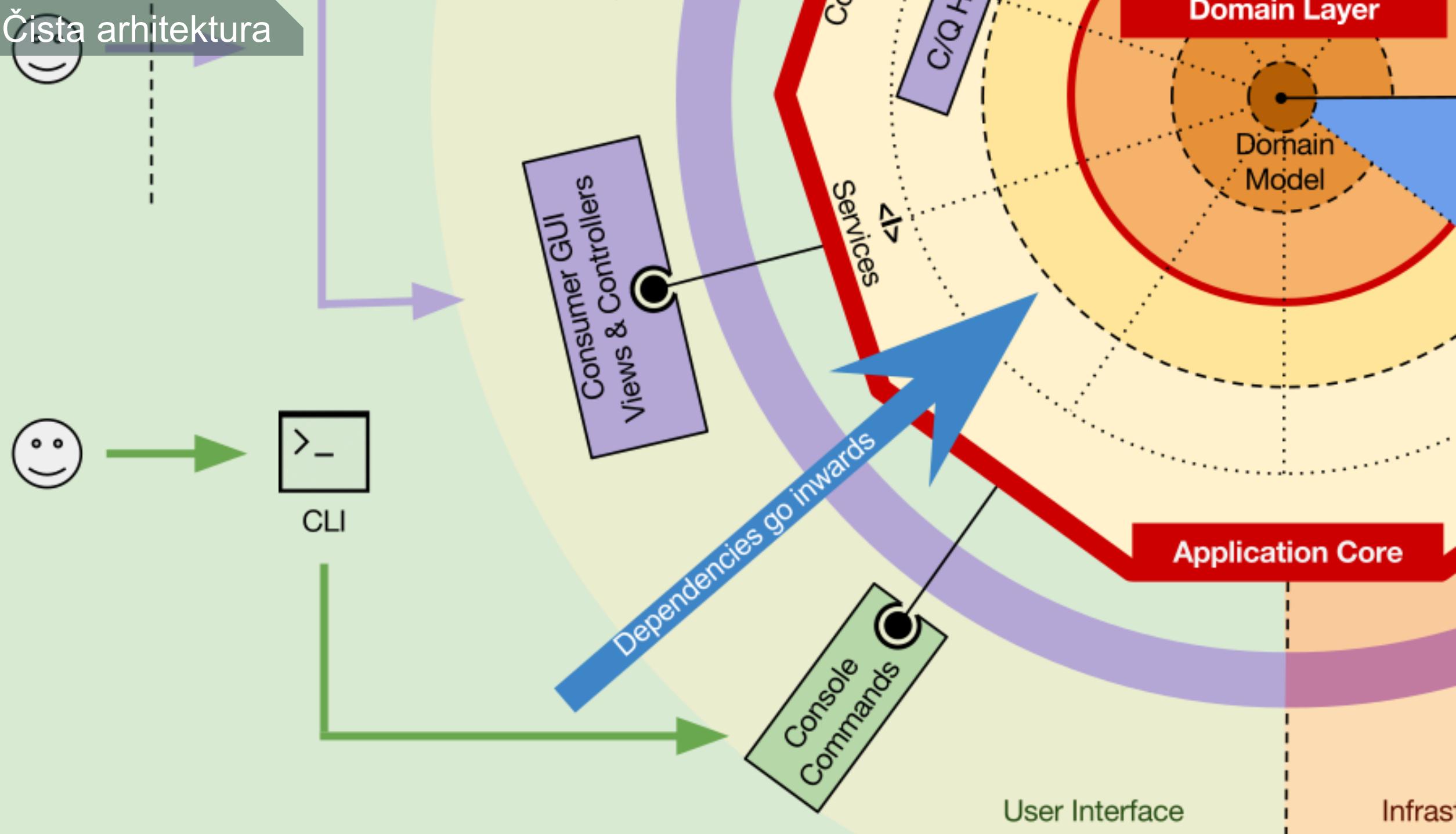
# Čista arhitektura

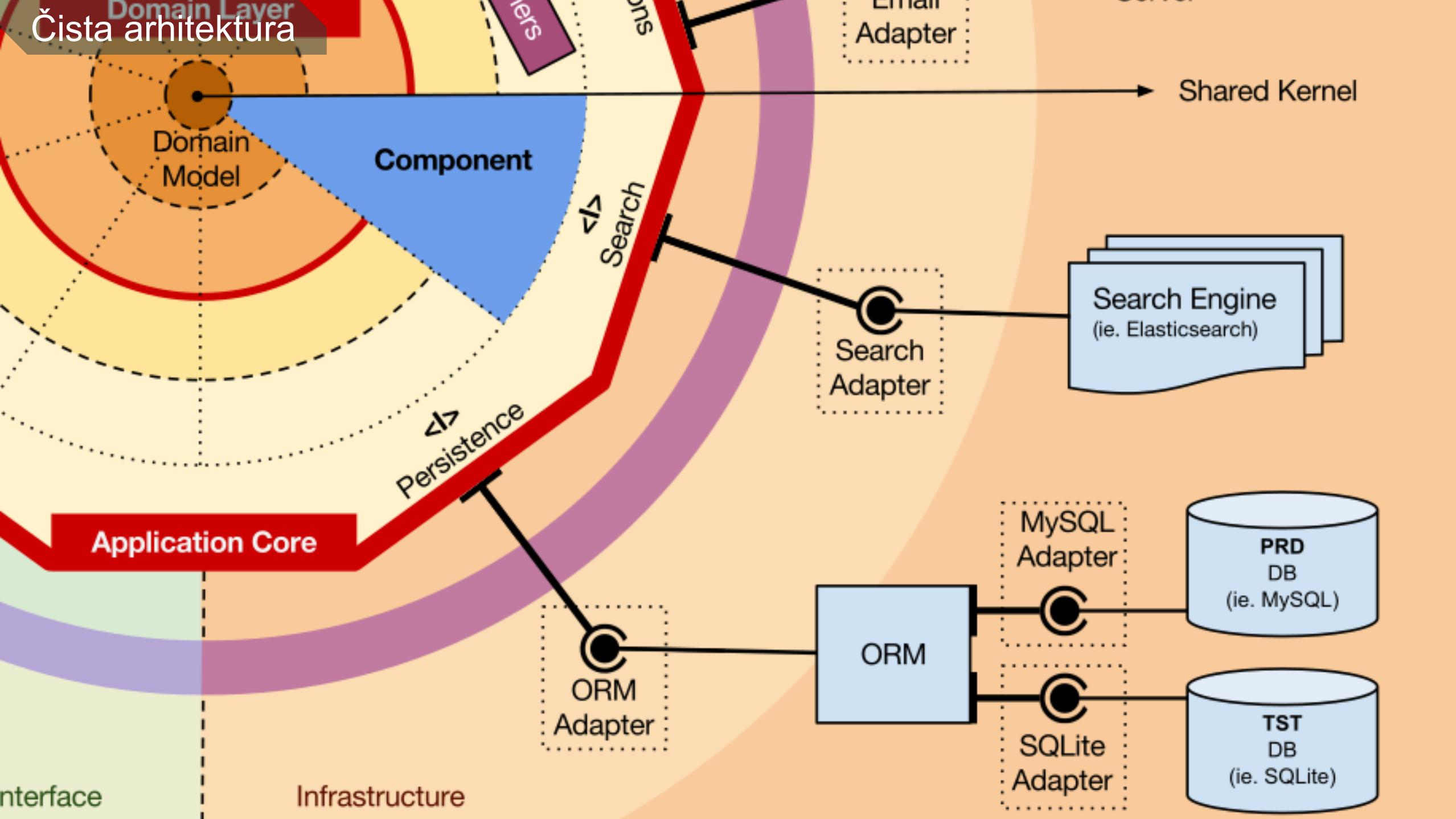


# Čista arhitektura

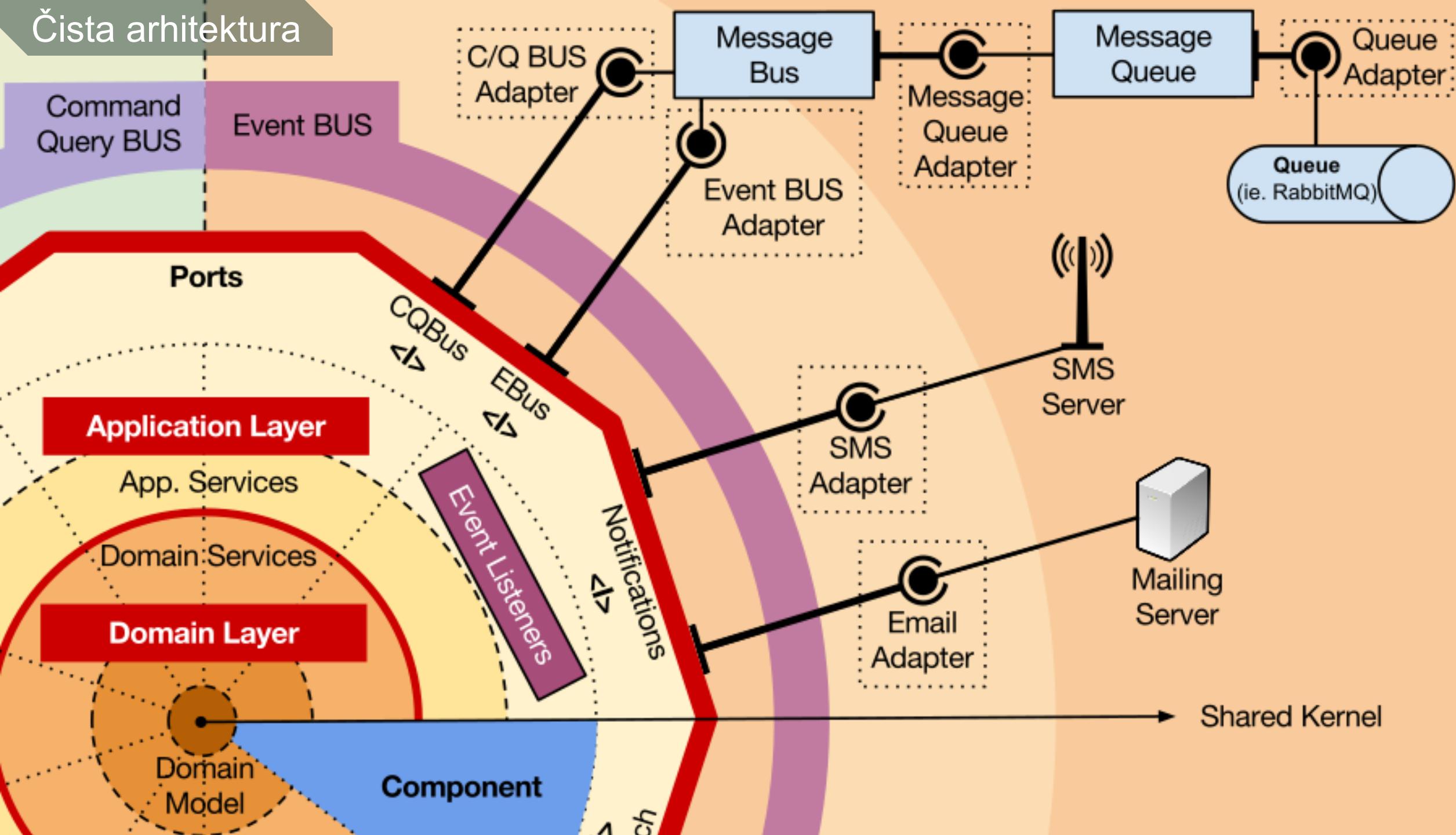


# Čista arhitektura

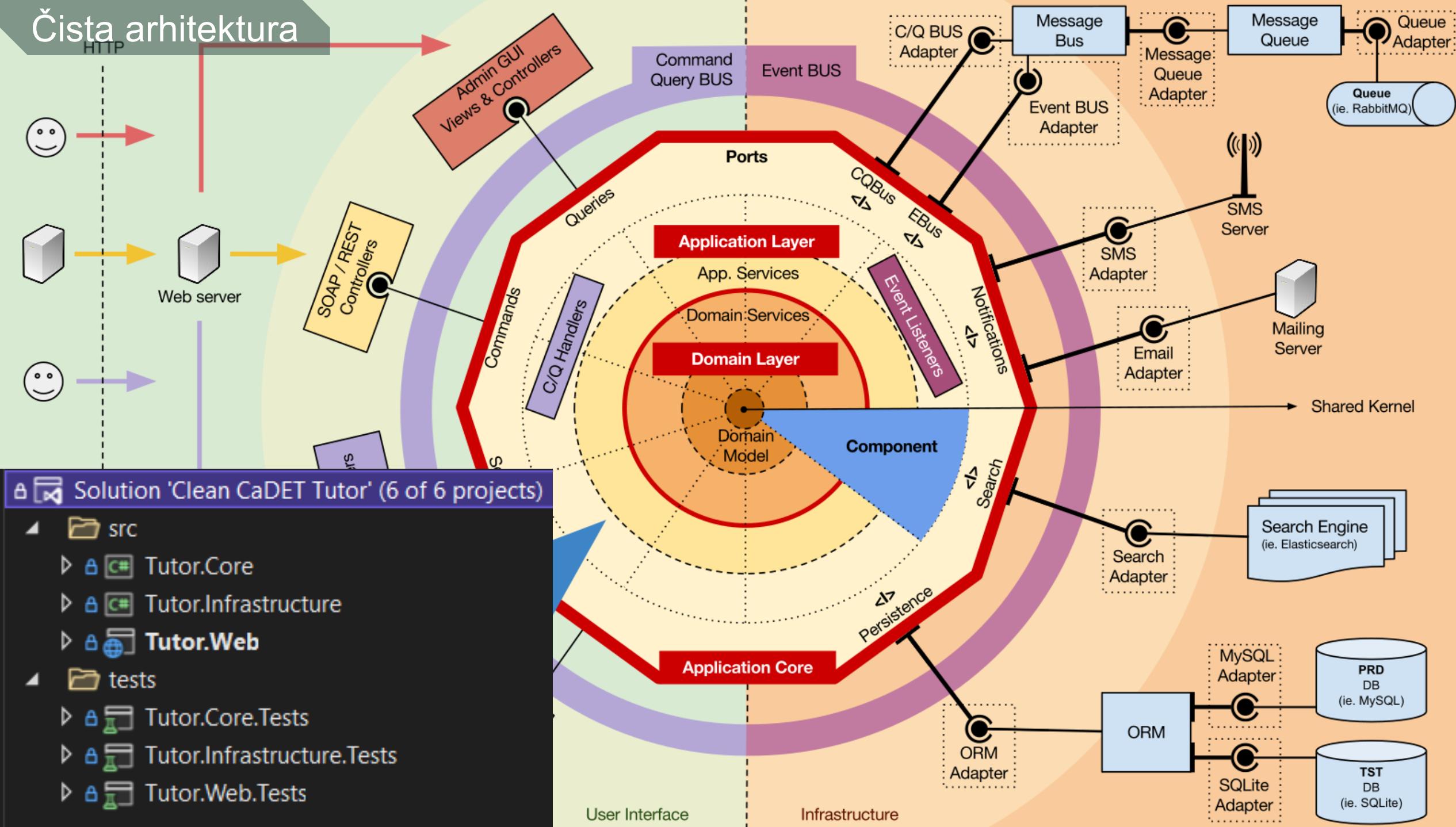




# Čista arhitektura



# Čista arhitektura



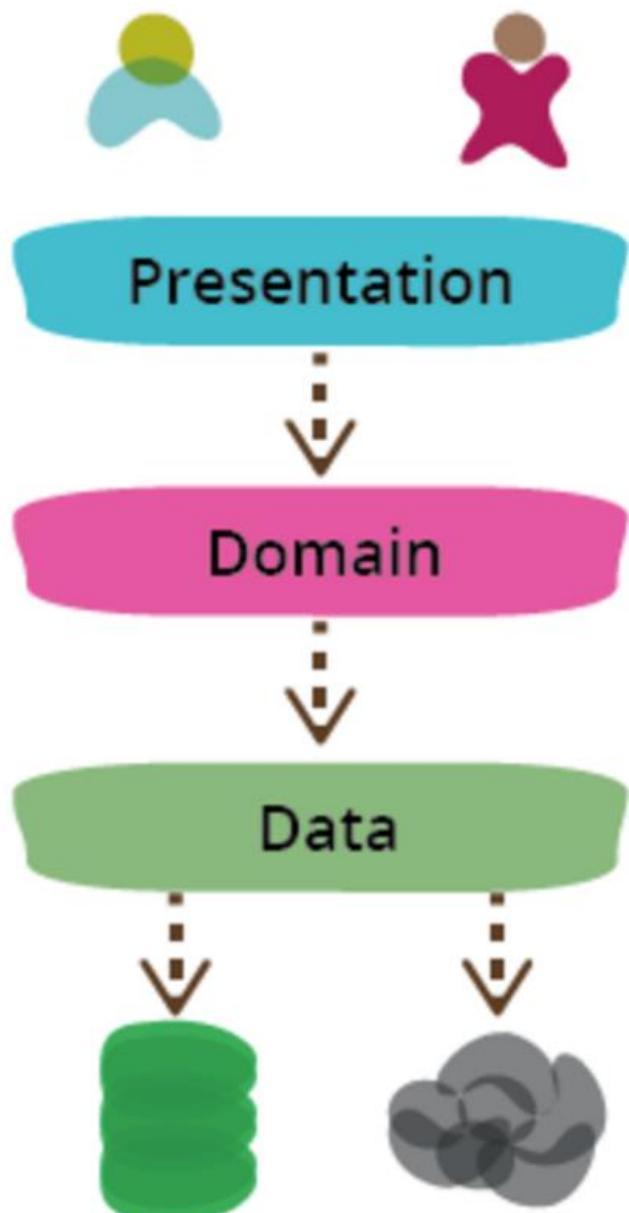


# Zdravokorporacija

- ▶ Dependencies
- ▶ Controller
- ▶ DTO
- ▶ Model
- ▶ Repository
- ▶ Resources
- ▶ Service
- ▶ View
- ▶ App.xaml
- ▶ AssemblyInfo.cs
- ▶ MainWindow.xaml

- ▶ Service
  - ▶ C# AbsenceRequestService.cs
  - ▶ C# AdvancedRenovationJoiningService.cs
  - ▶ C# AdvancedRenovationSeparationService.cs
  - ▶ C# AnamnesisService.cs
  - ▶ C# AppointmentService.cs
  - ▶ C# BasicRenovationService.cs
  - ▶ C# DoctorService.cs
  - ▶ C# EquipmentService.cs
  - ▶ C# ManagerService.cs
  - ▶ C# MedicalRecordService.cs
  - ▶ C# MedicationService.cs
  - ▶ C# NotificationService.cs
  - ▶ C# PatientService.cs
  - ▶ C# PrescriptionService.cs
  - ▶ C# RatingService.cs
  - ▶ C# RoomService.cs
  - ▶ C# SecretaryService.cs

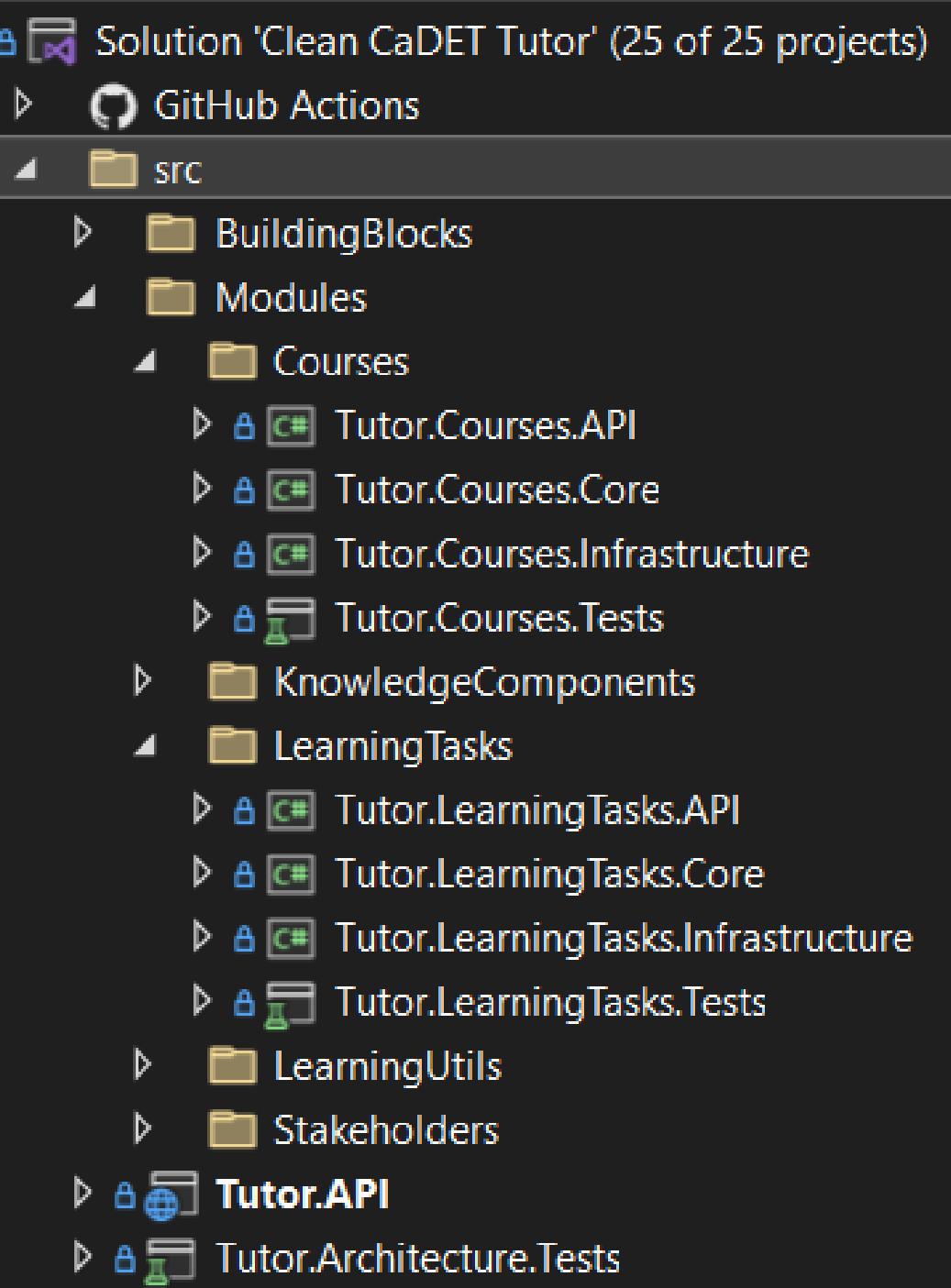
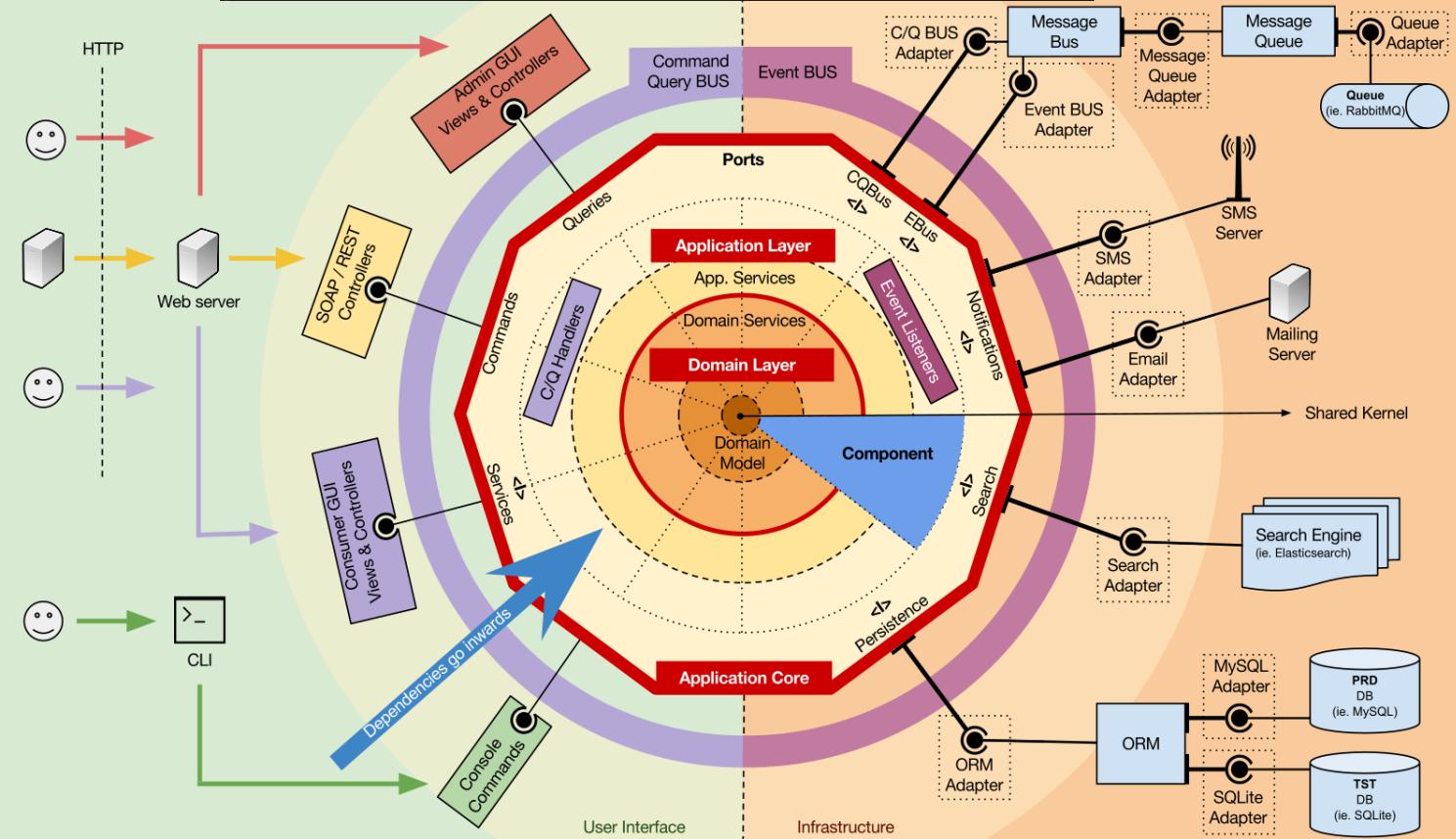
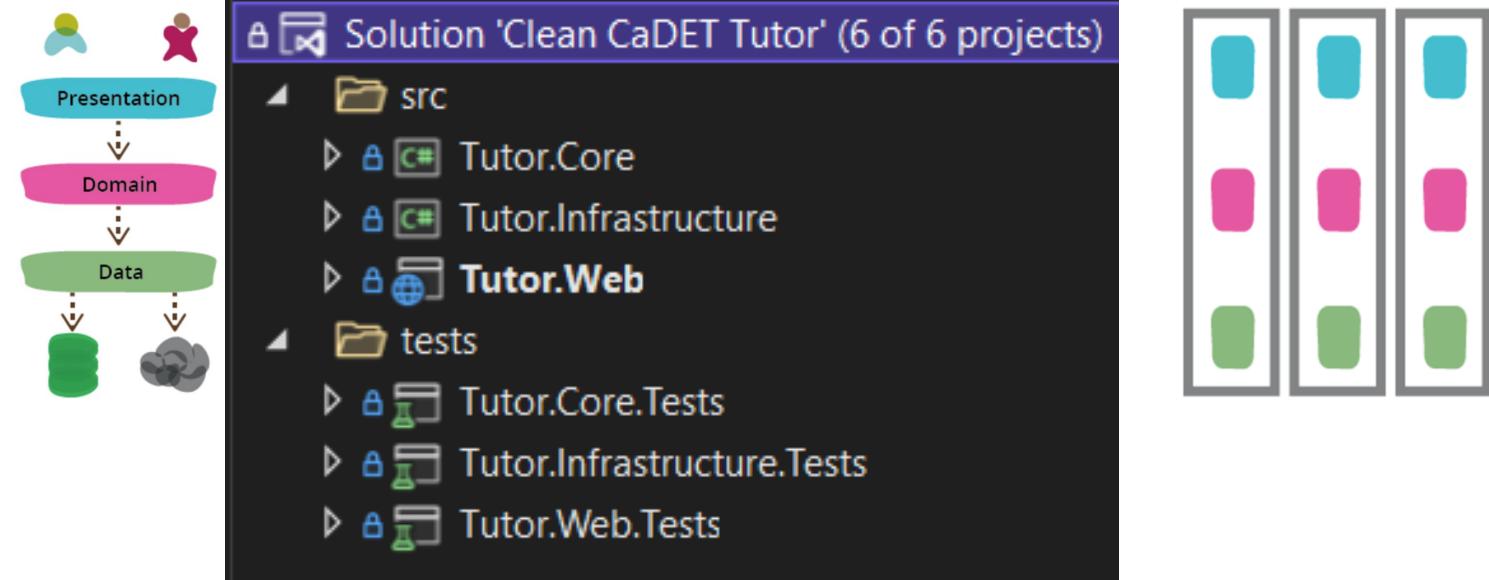
šta otvaram  
ako dodajem  
novu funkciju  
za pacijente?



*Package by layer*

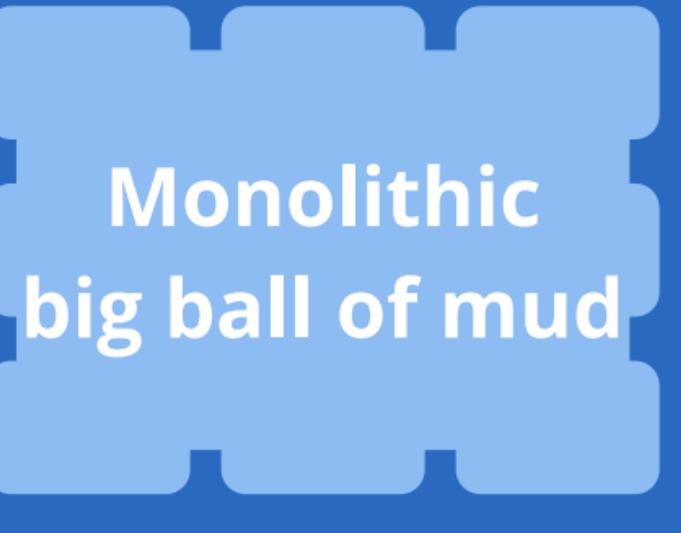
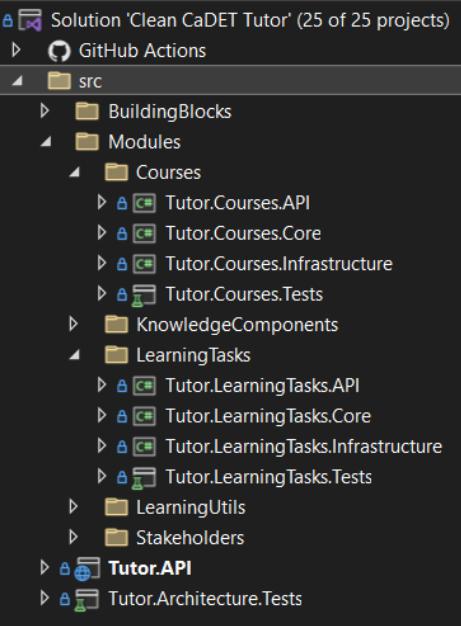
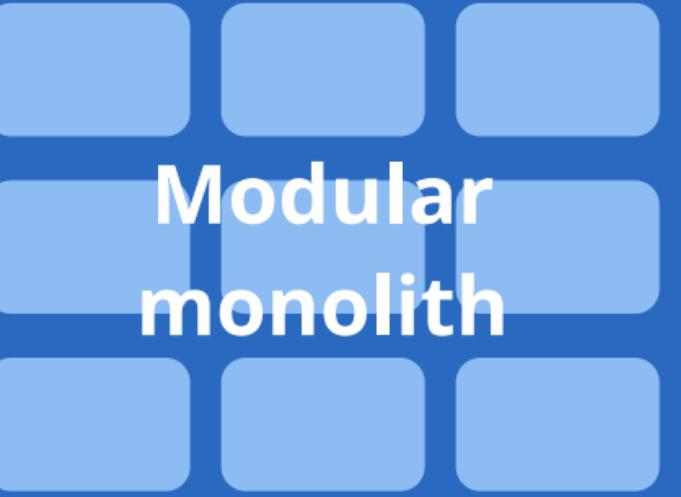


*Package by feature*



Modularity

Modular  
monolith



Monolithic  
big ball of mud



Microservices



Distributed  
big ball of mud

Number of deployment units