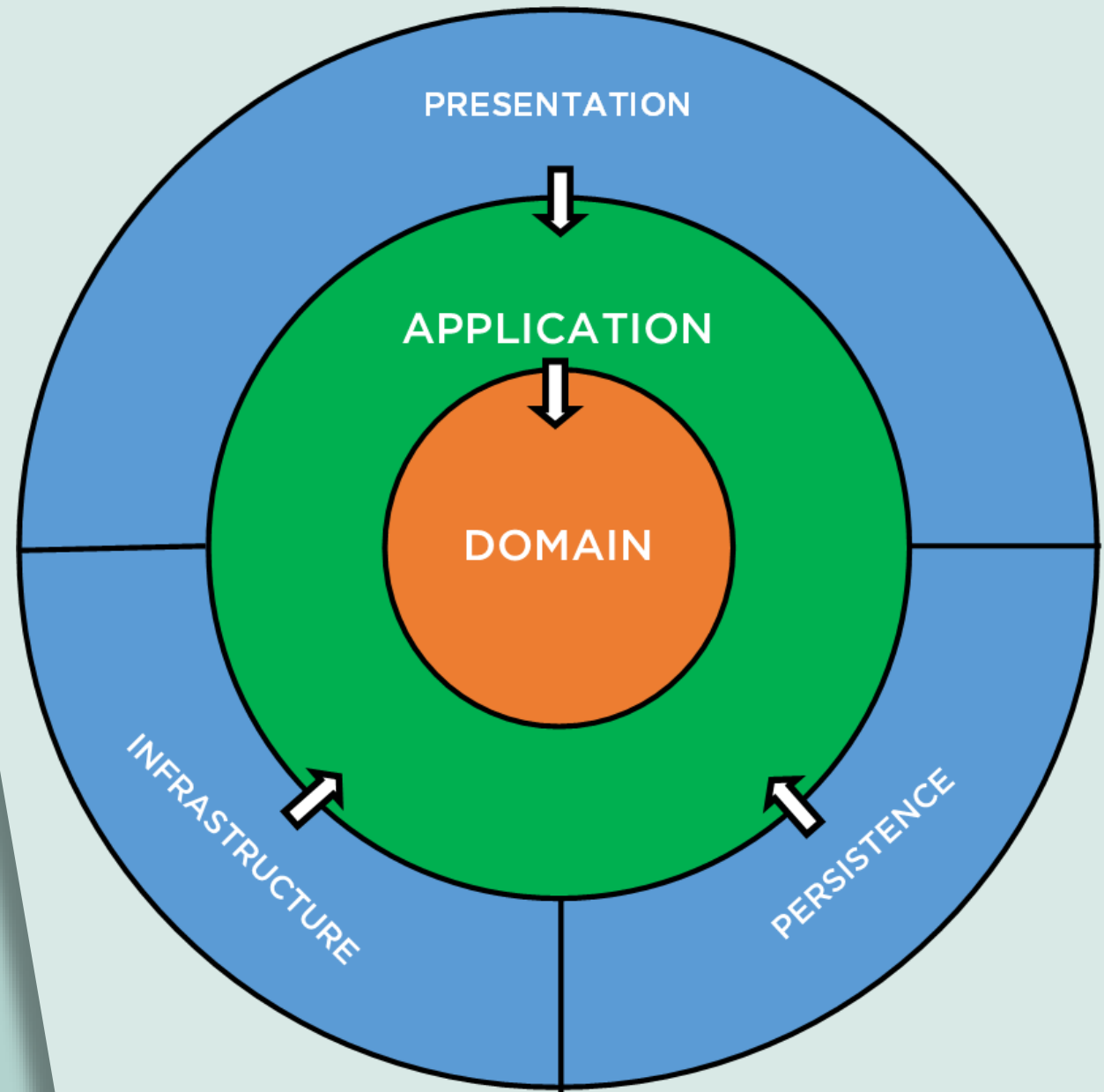


Uvod u softversko inženjerstvo

Čisti paketi
i arhitektura

Nikola Luburić

nikola.luburic@uns.ac.rs



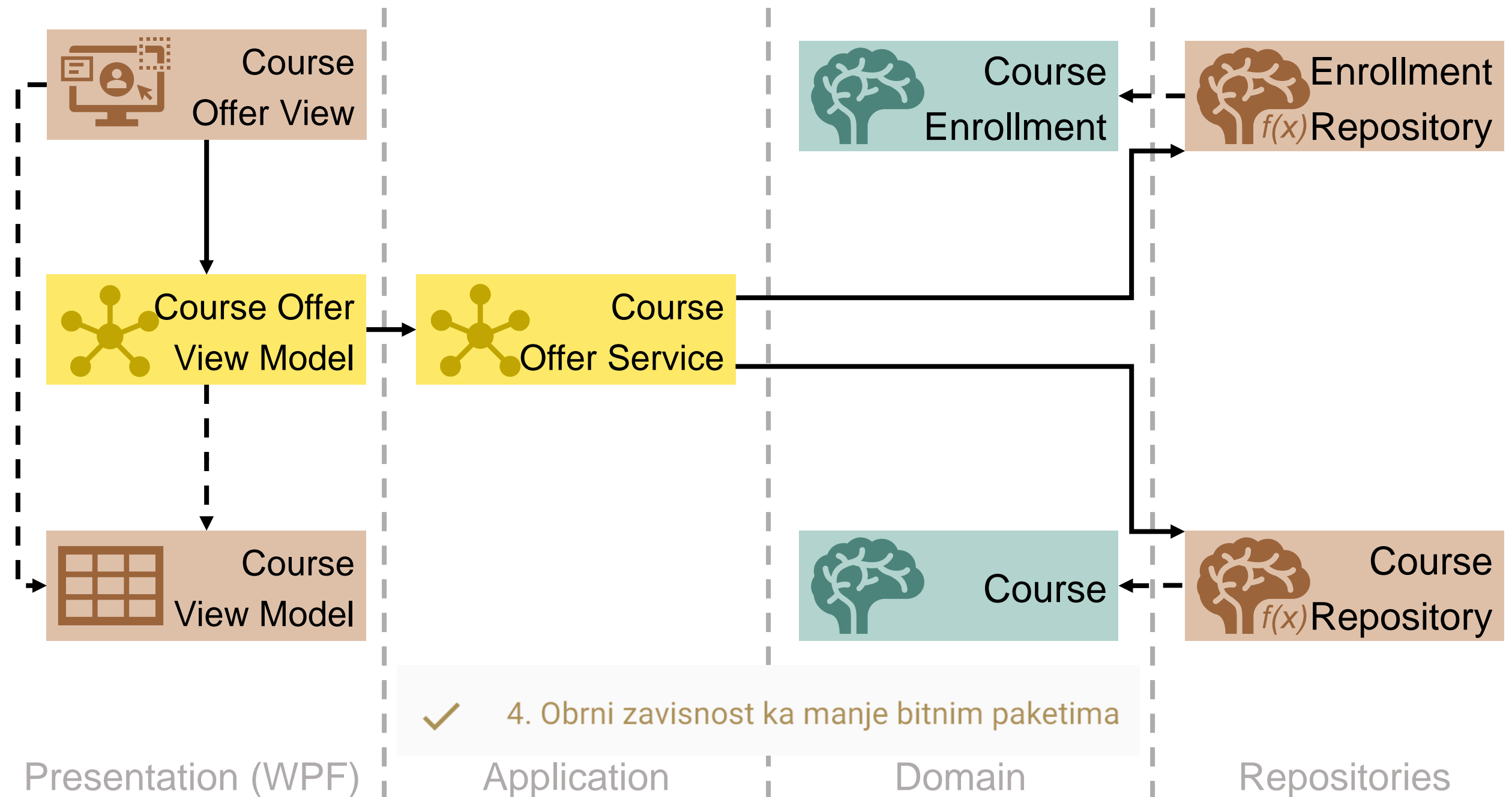
MVVM paketiranje

Princip inverzije zavisnosti

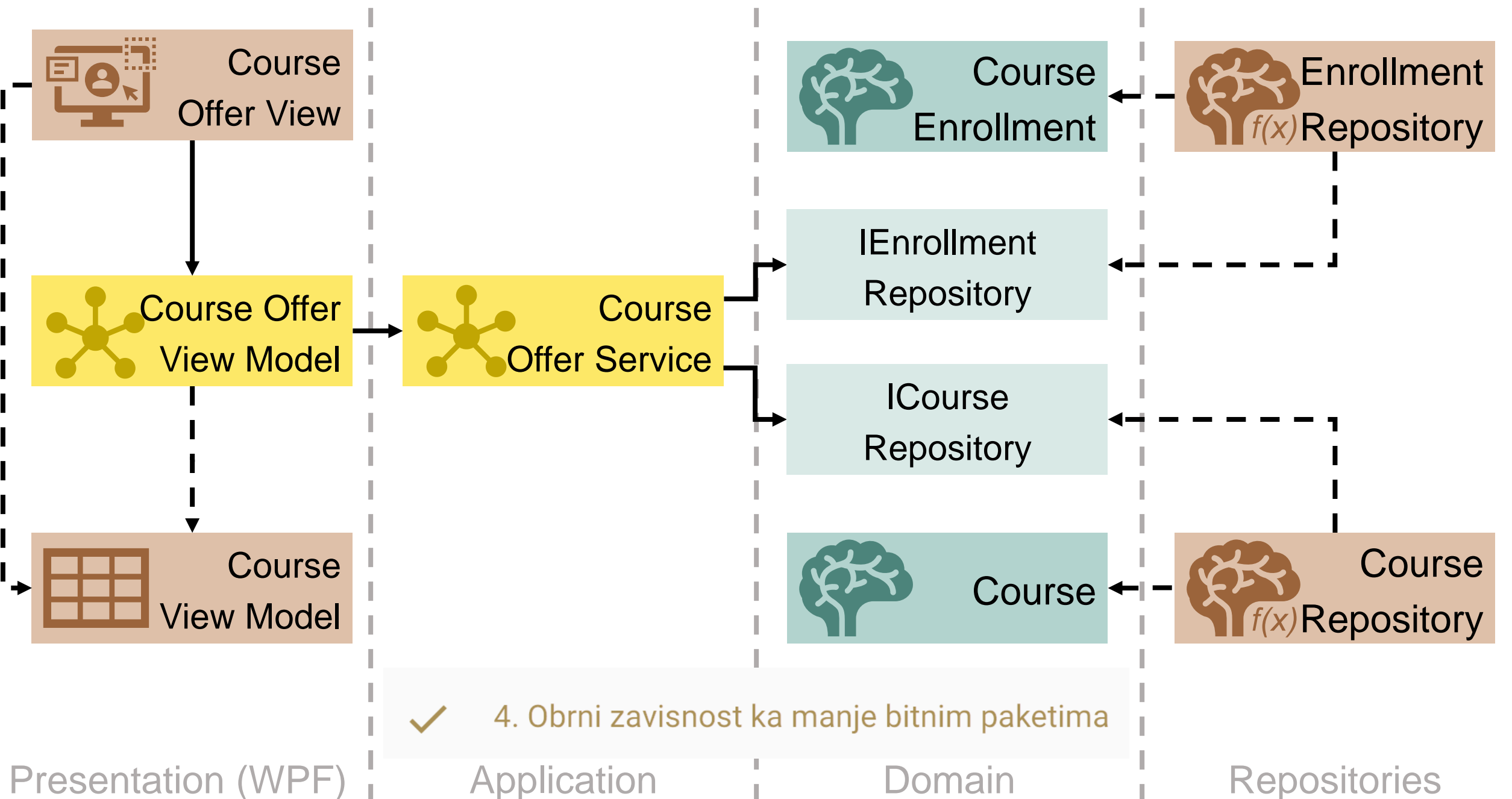
Čisti paketi
i arhitektura

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.



SRP

OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija

*šta treba da
uradi sa bazom?*

Aplikacioni i
domenski
sloj



Baza
podataka

SRP

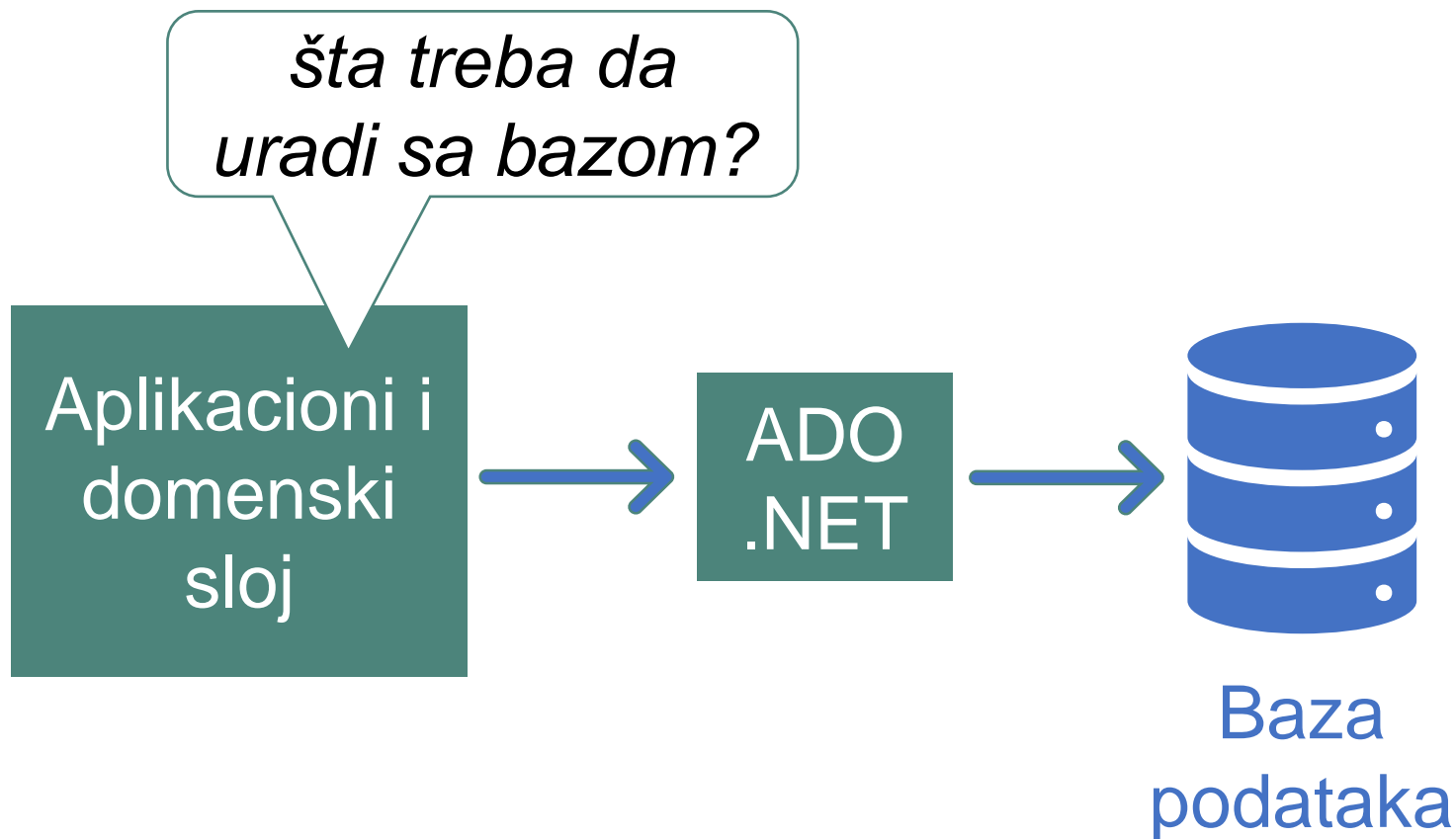
OCP

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() {  
        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

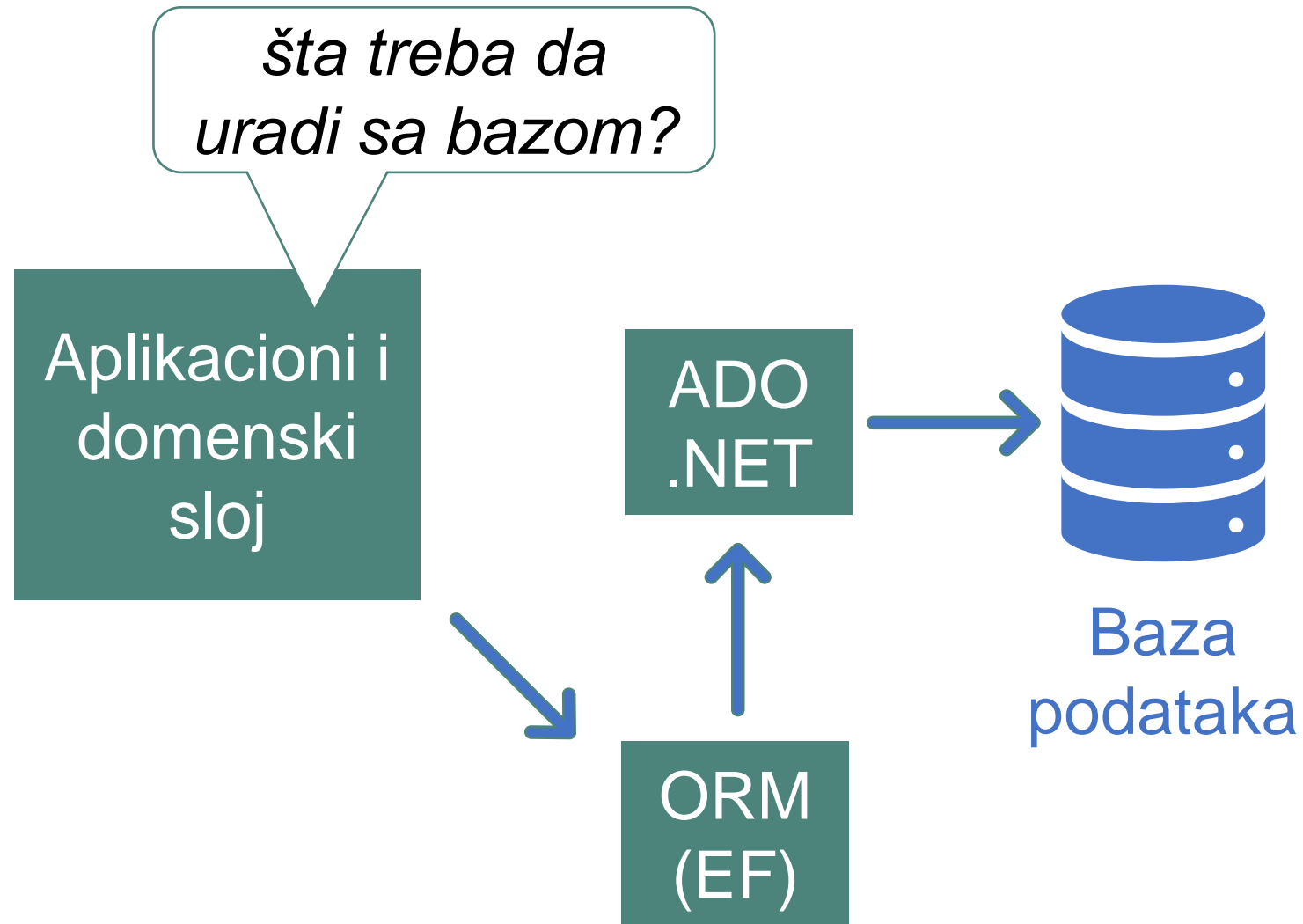
OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija



SRP

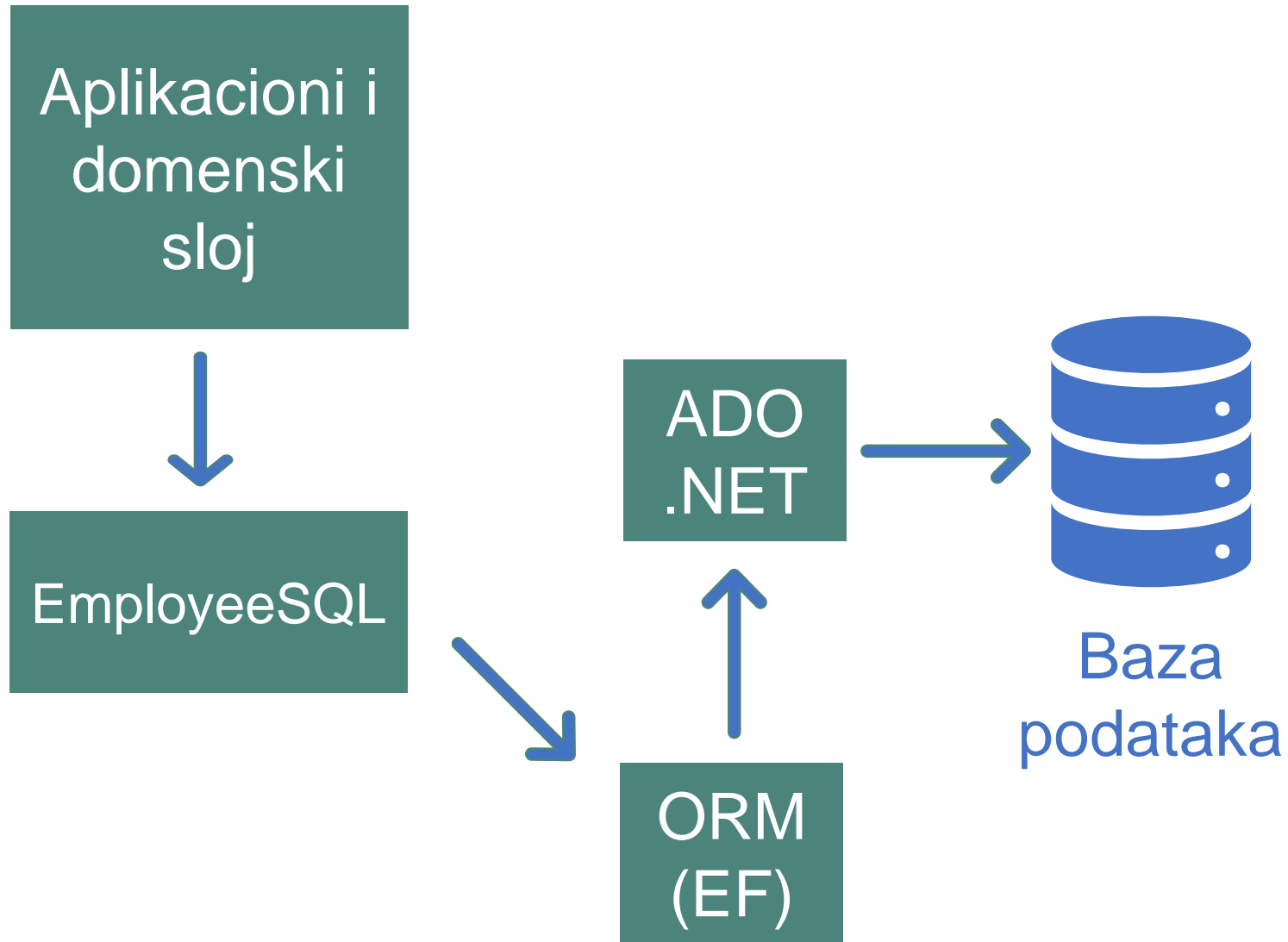
OCP

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

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

*Znak da se
DIP ne prati*

*da li Reporter
zavisi od DB?*

SRP

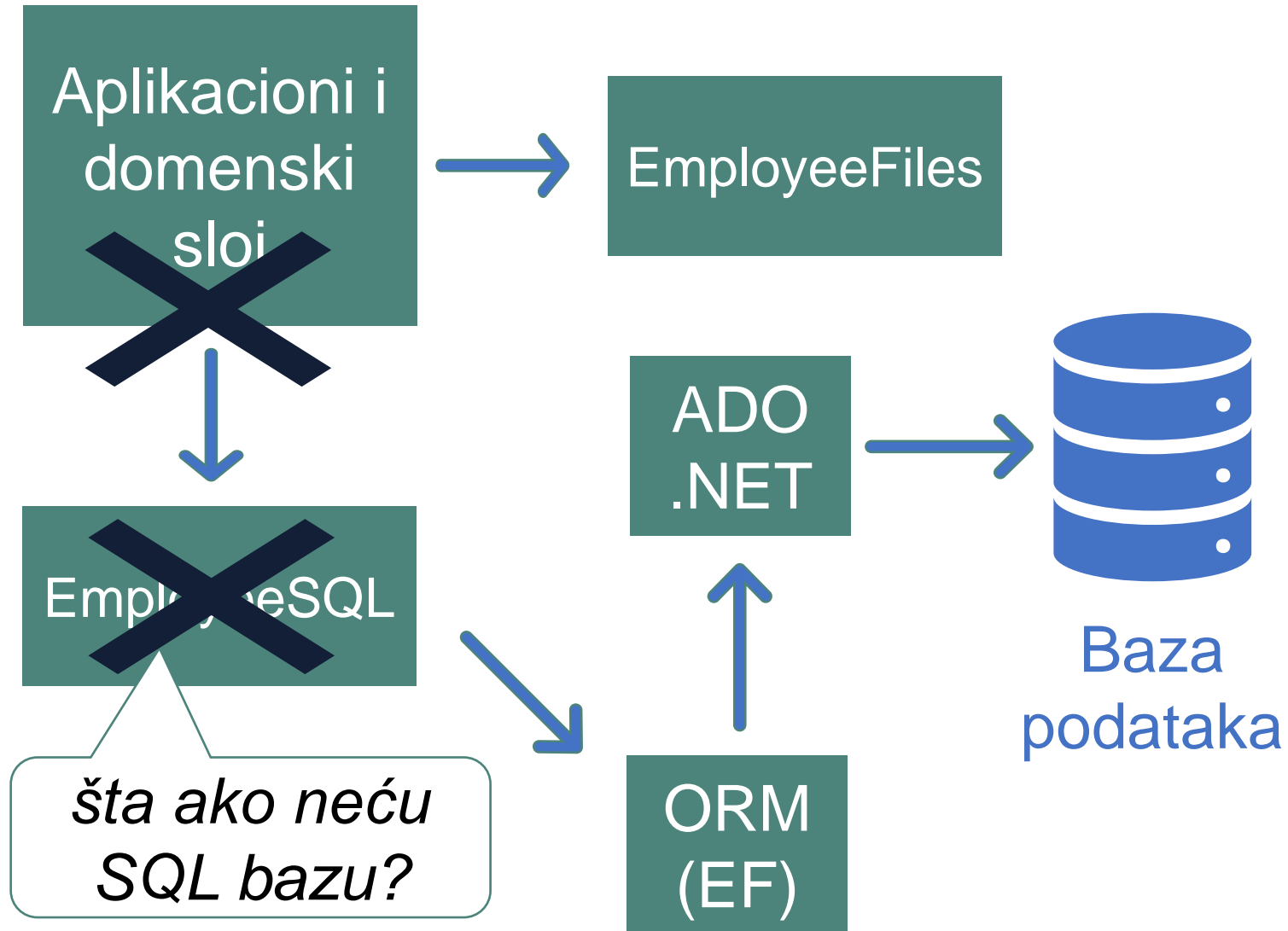
OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija



Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija

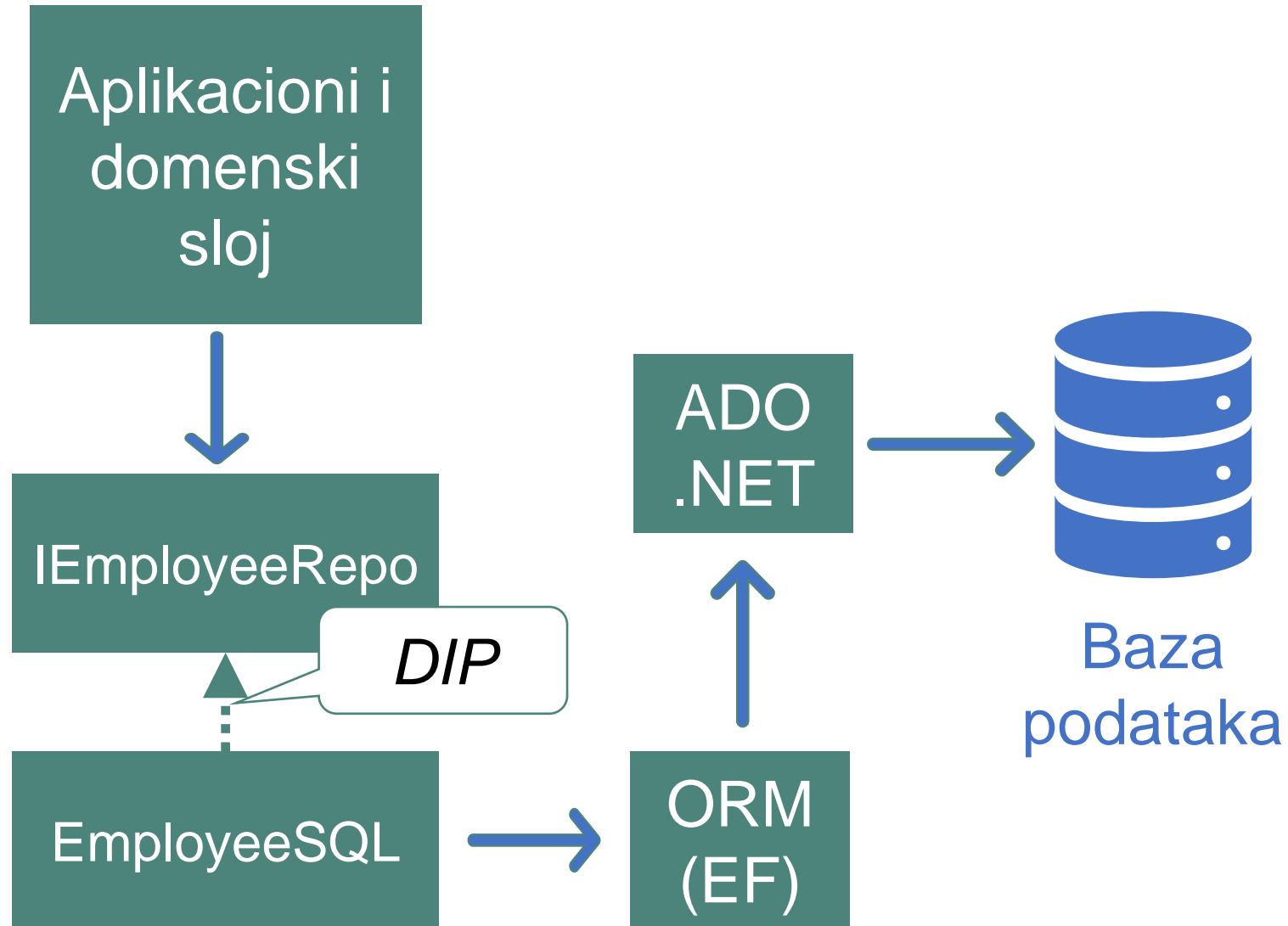
SRP

OCP

LSP

ISP

DIP



SRP

OCP

LSP

ISP

DIP

Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija

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

Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija

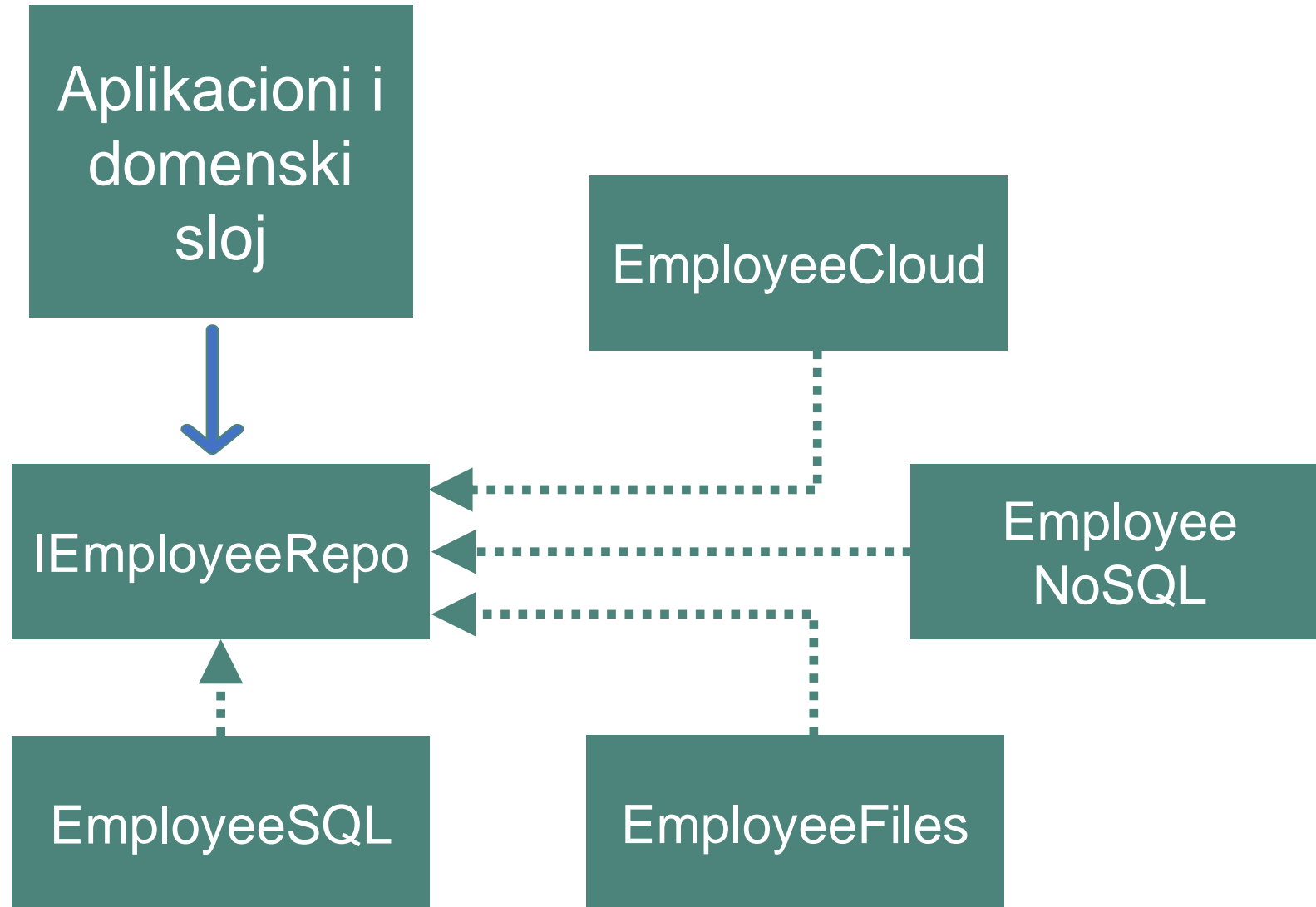
SRP

OCP

LSP

ISP

DIP



SRP

Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija

OCP

Dependency injection

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

LSP

ISP

DIP

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

SRP

Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija

OCP

Dependency injection

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

LSP

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

ISP

DIP

SRP

OCP

LSP

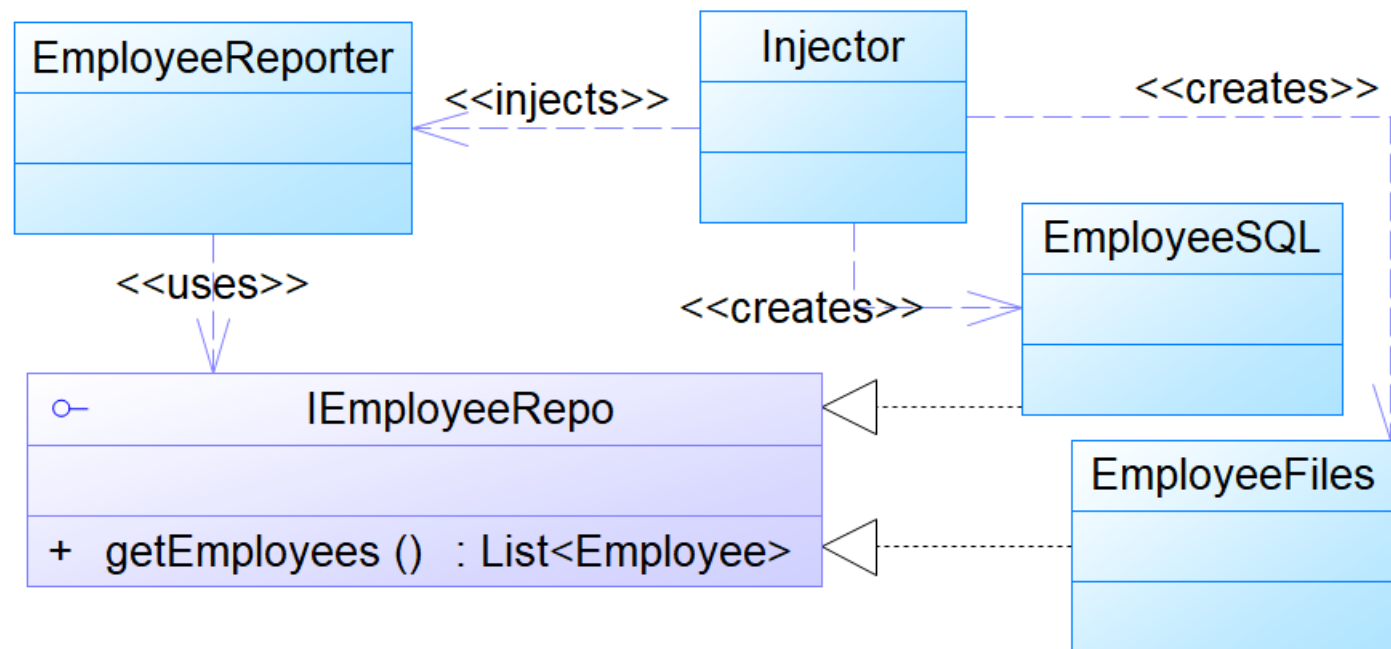
ISP

DIP

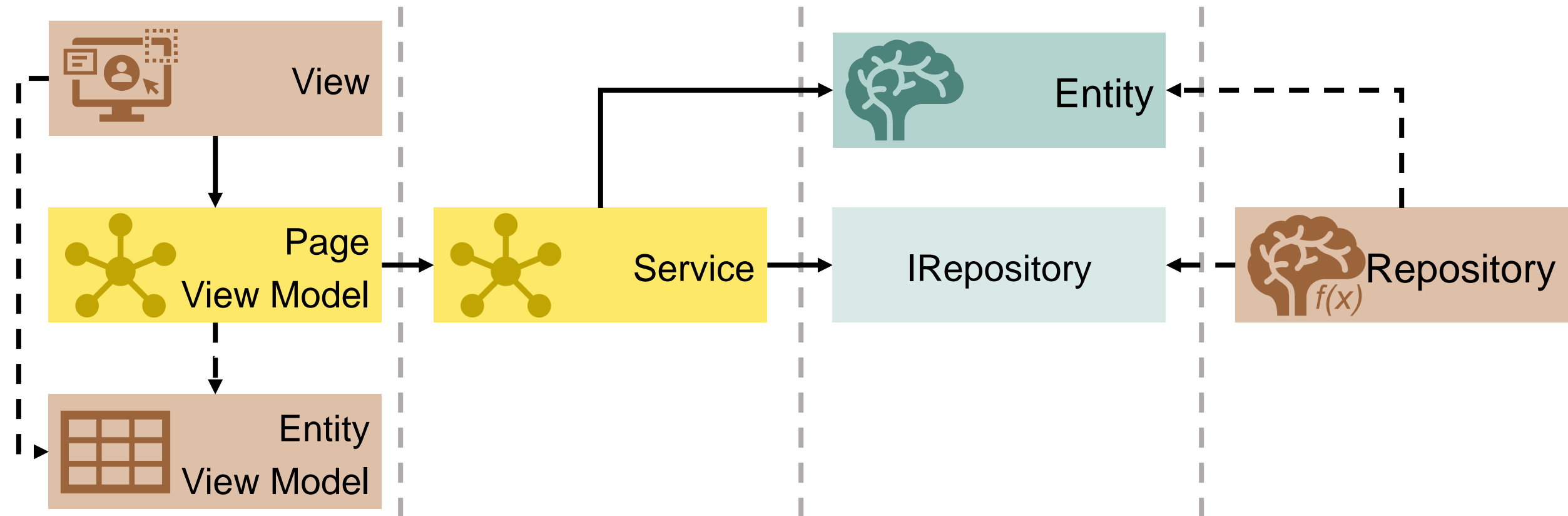
Apstrakcije ne zavise od implementacije,
implementacije zavise od apstrakcija

Dependency injection

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

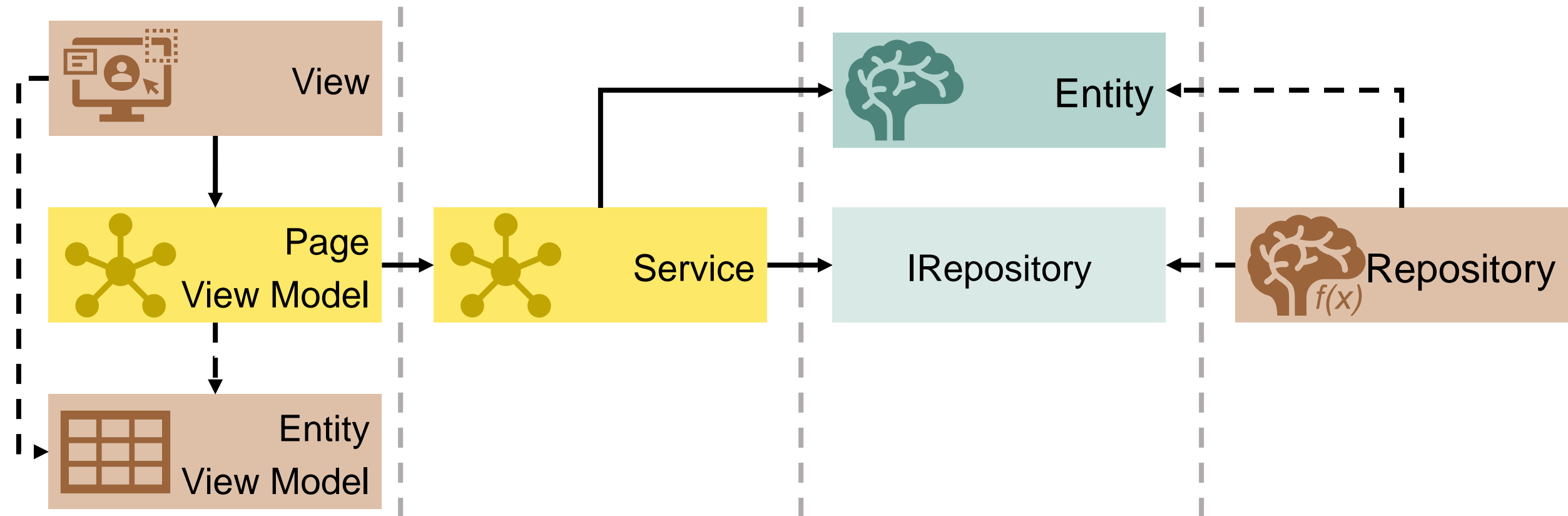


Direktor može da zatraži izveštaj koji ističe prosečan broj poena ostvaren na svakom od delova svih ispita u poslednjih godinu dana (koliko je studenata slušalo kurs, koliko položilo i procenat polaganja).

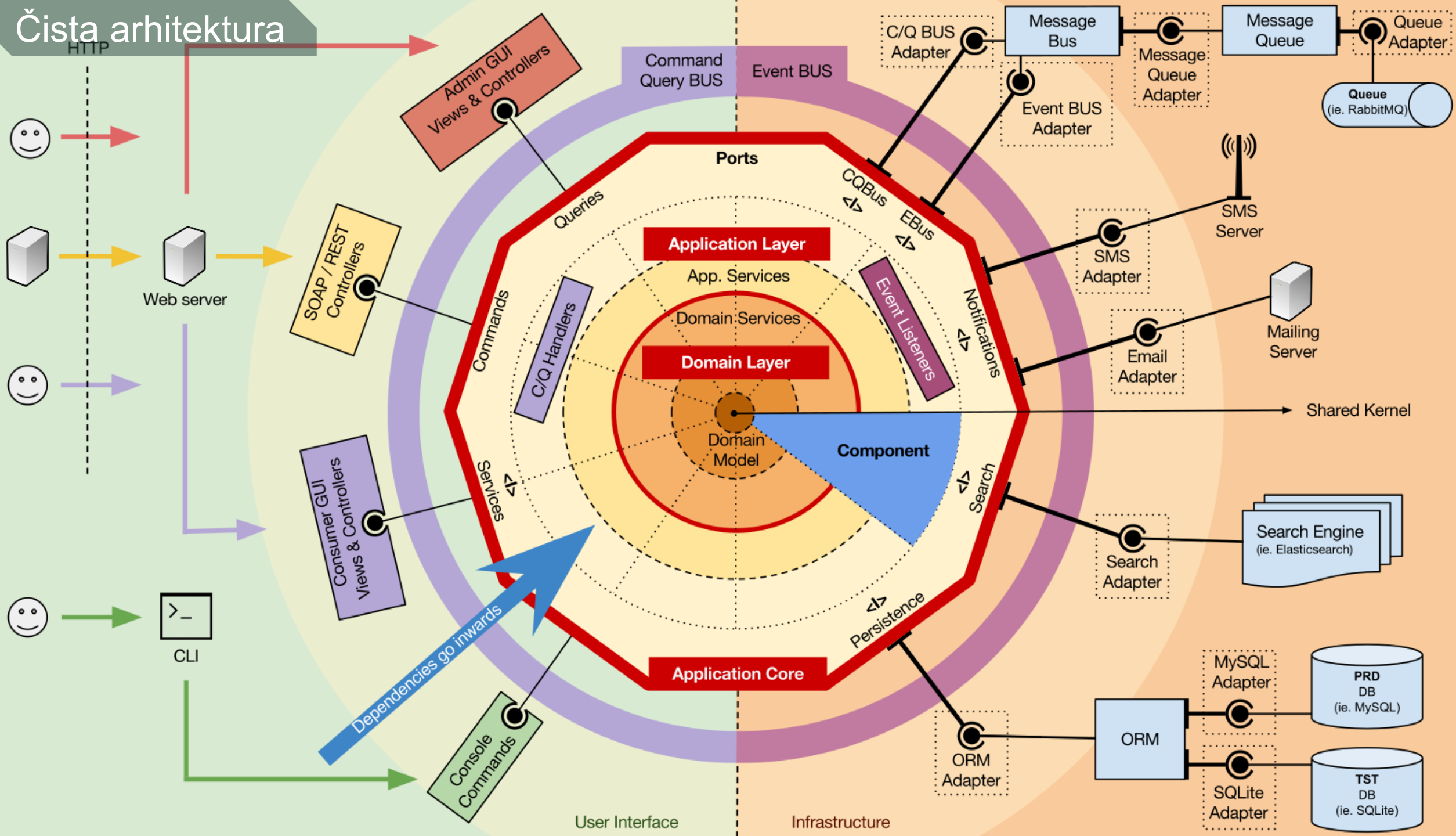


Direktor može da zatraži izveštaj koji ističe prosečan broj poena ostvaren na svakom od delova svih ispita u poslednjih godinu dana (koliko je studenata slušalo kurs, koliko položilo i procenat polaganja).

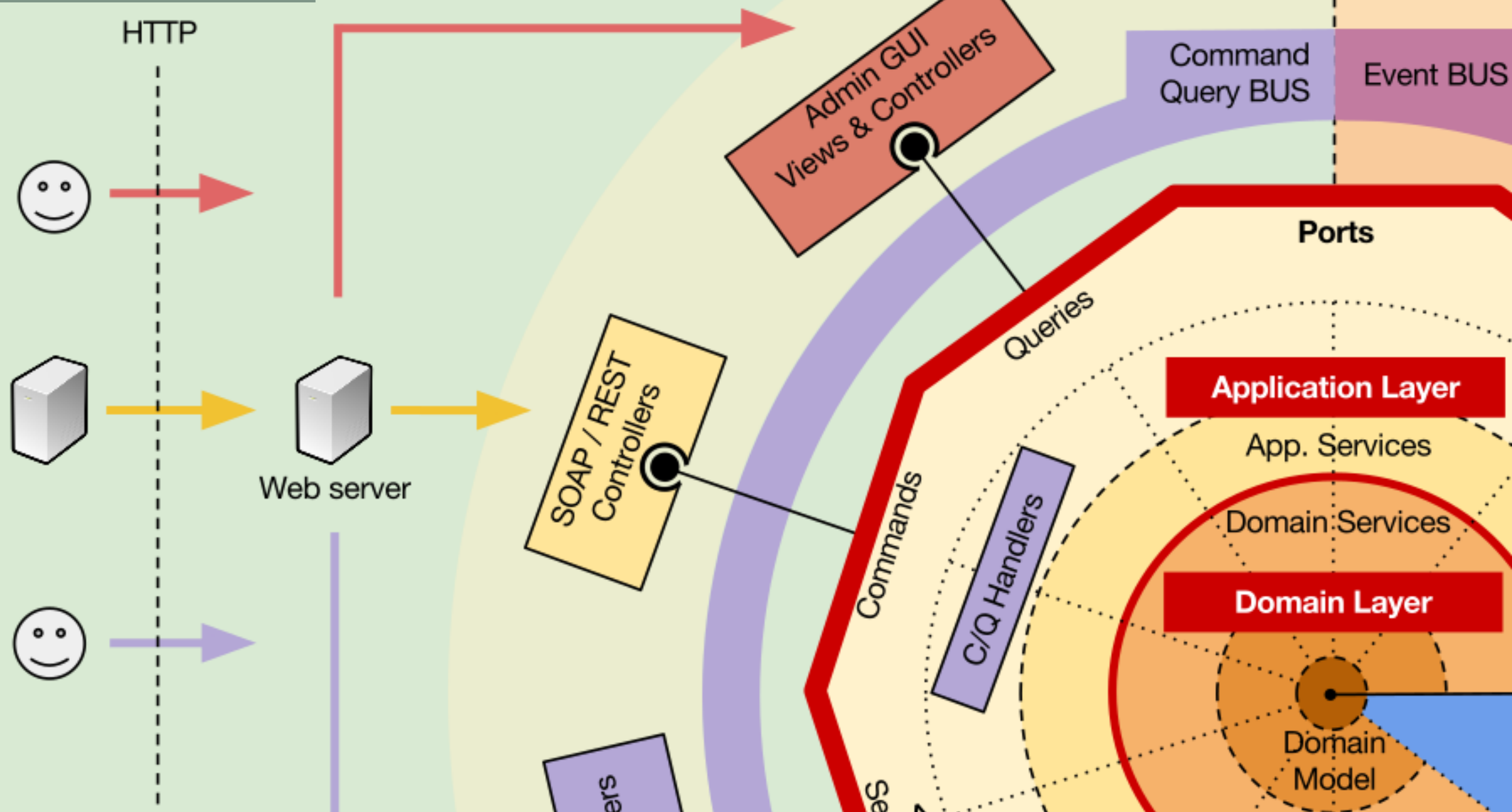
Izveštaj se generiše u PDF formatu i šalje na mejl direktora.



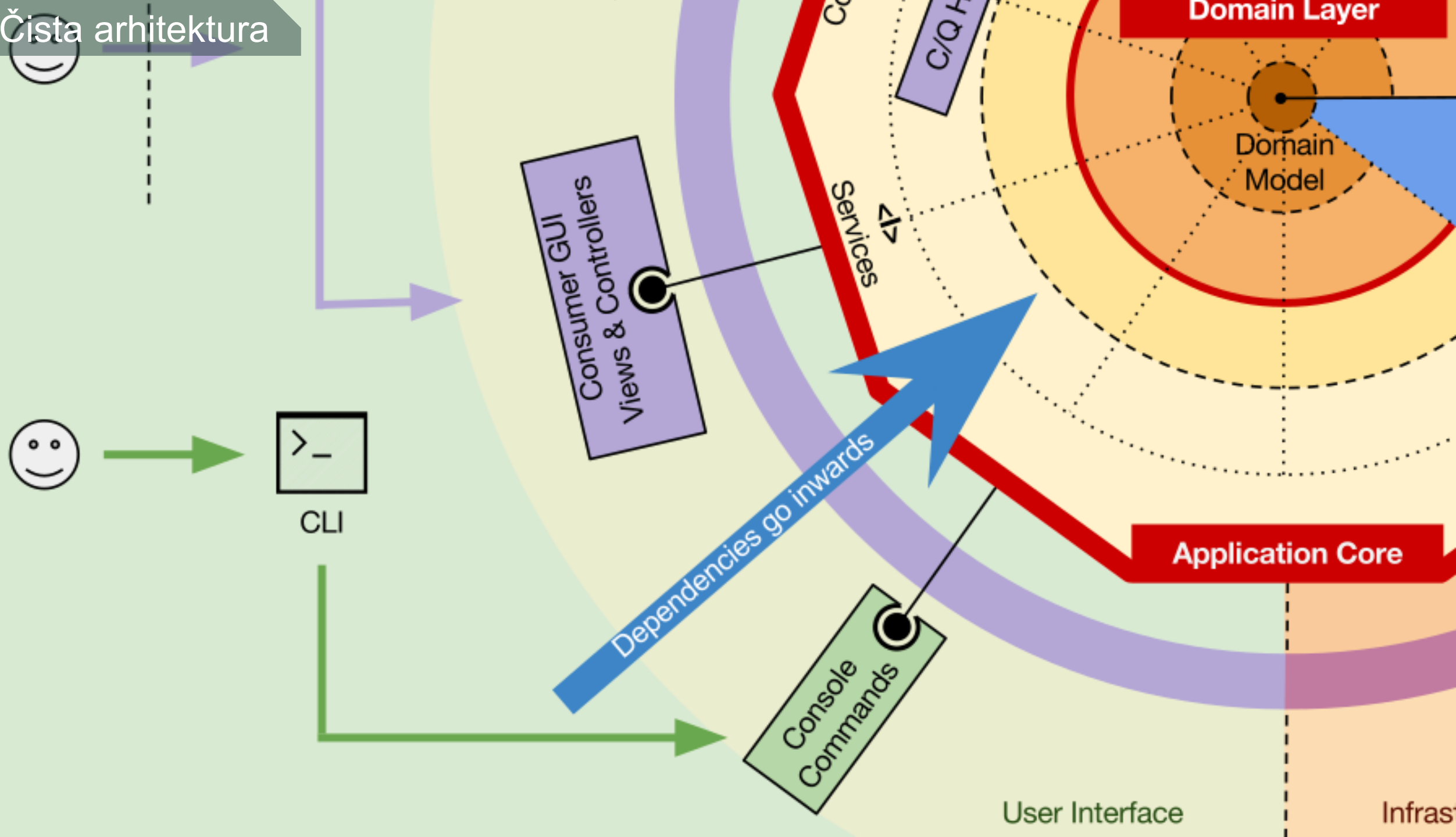
Čista arhitektura



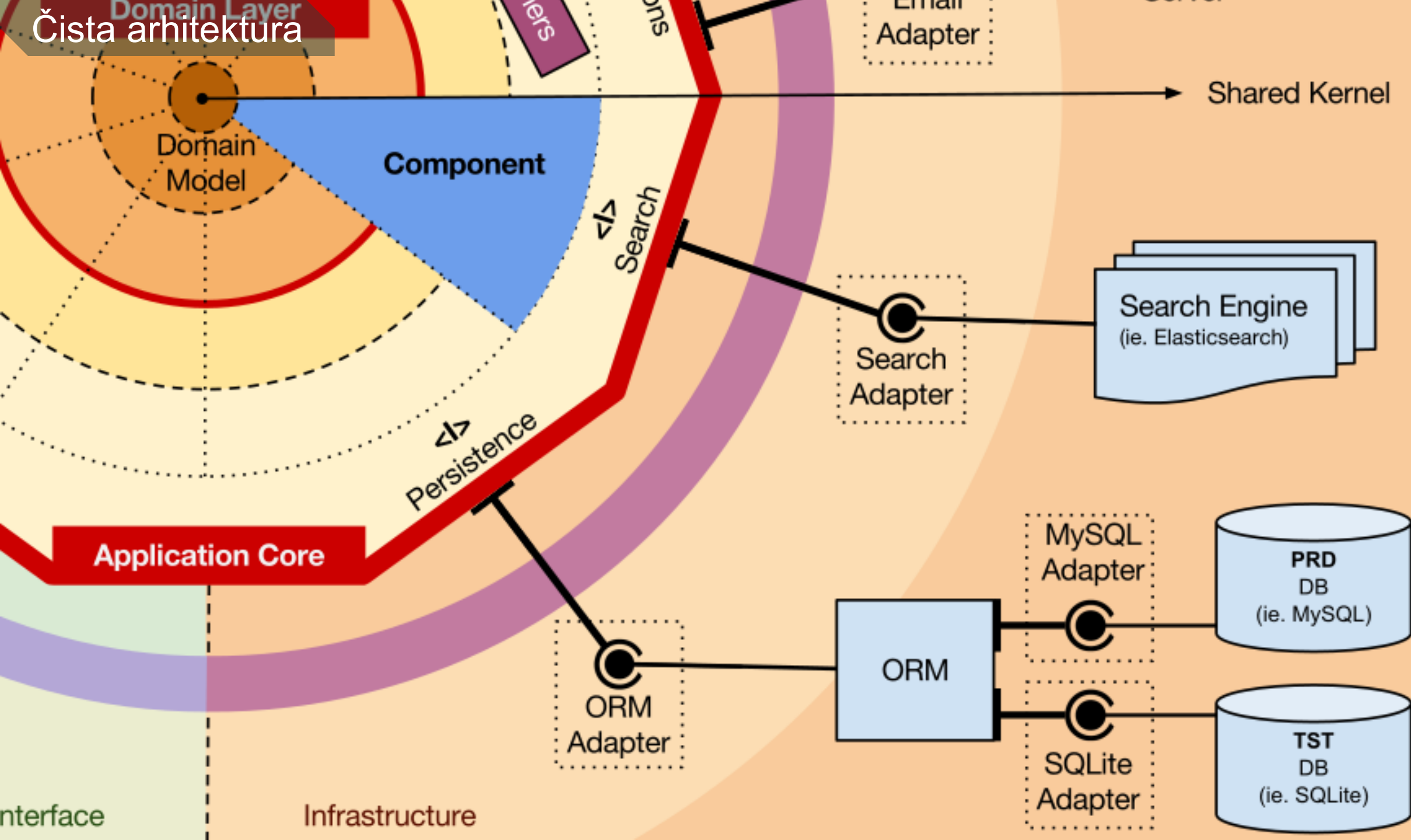
Čista arhitektura



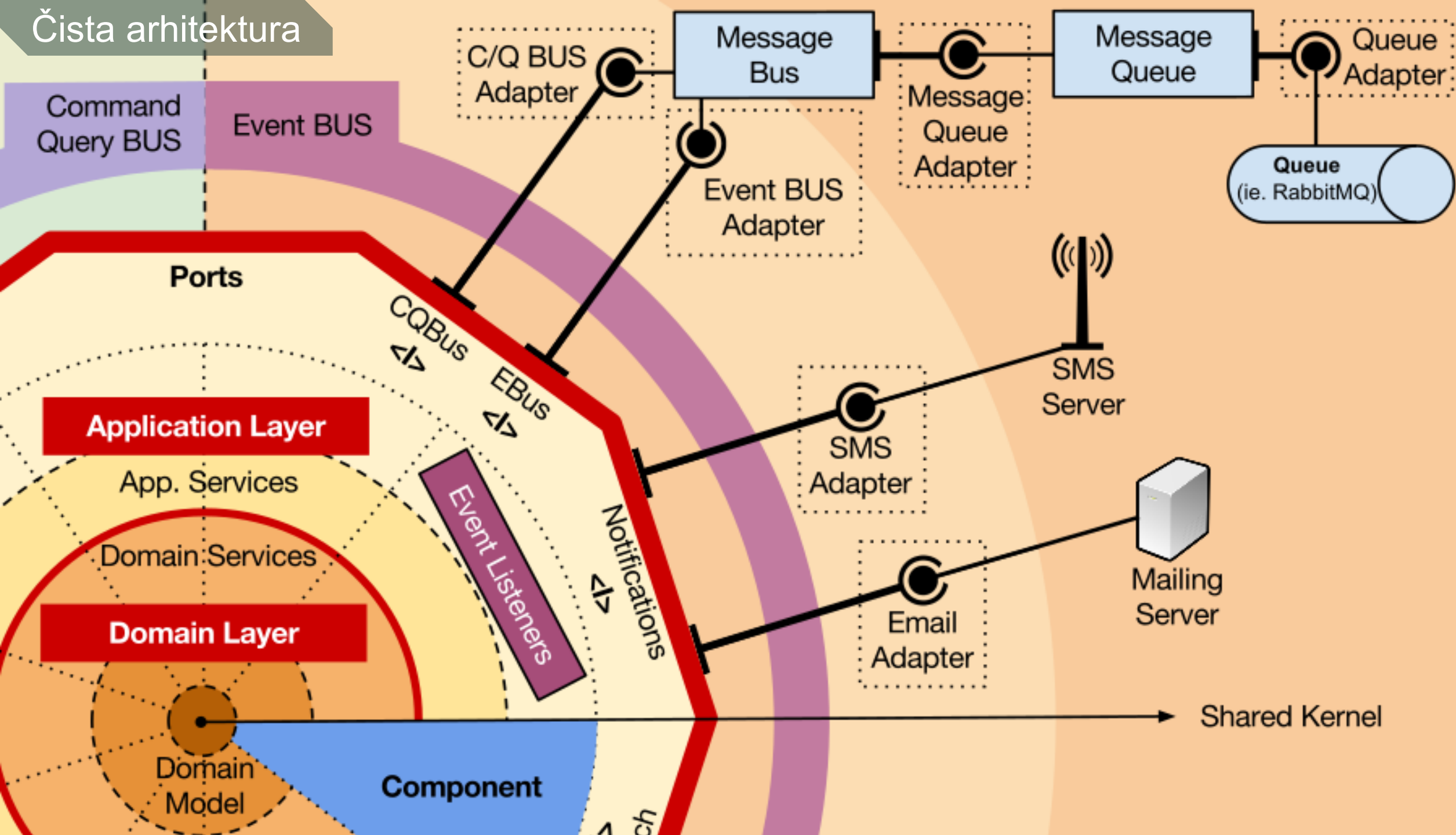
Čista arhitektura



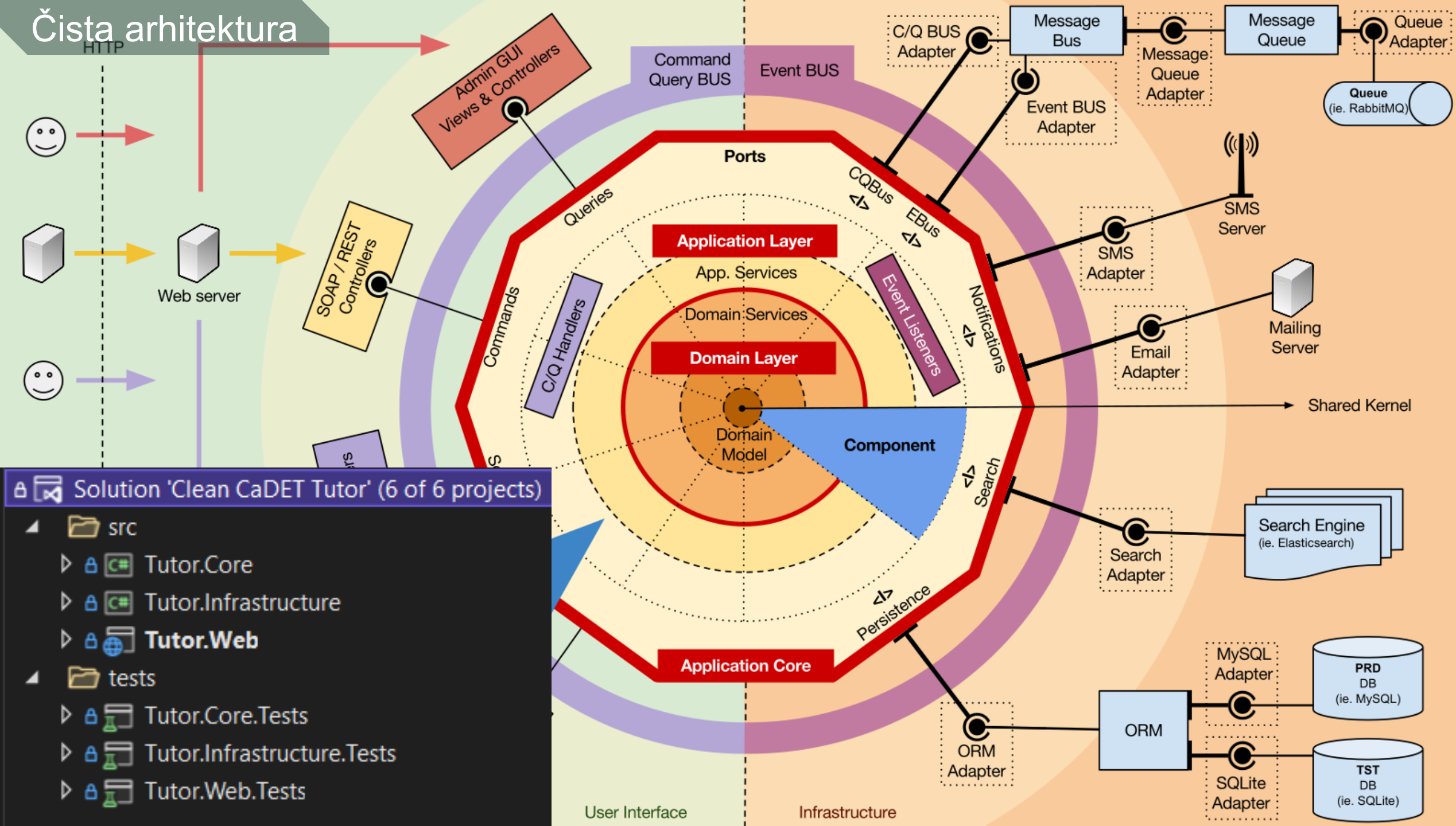
Čista arhitektura



Čista arhitektura



Čista arhitektura

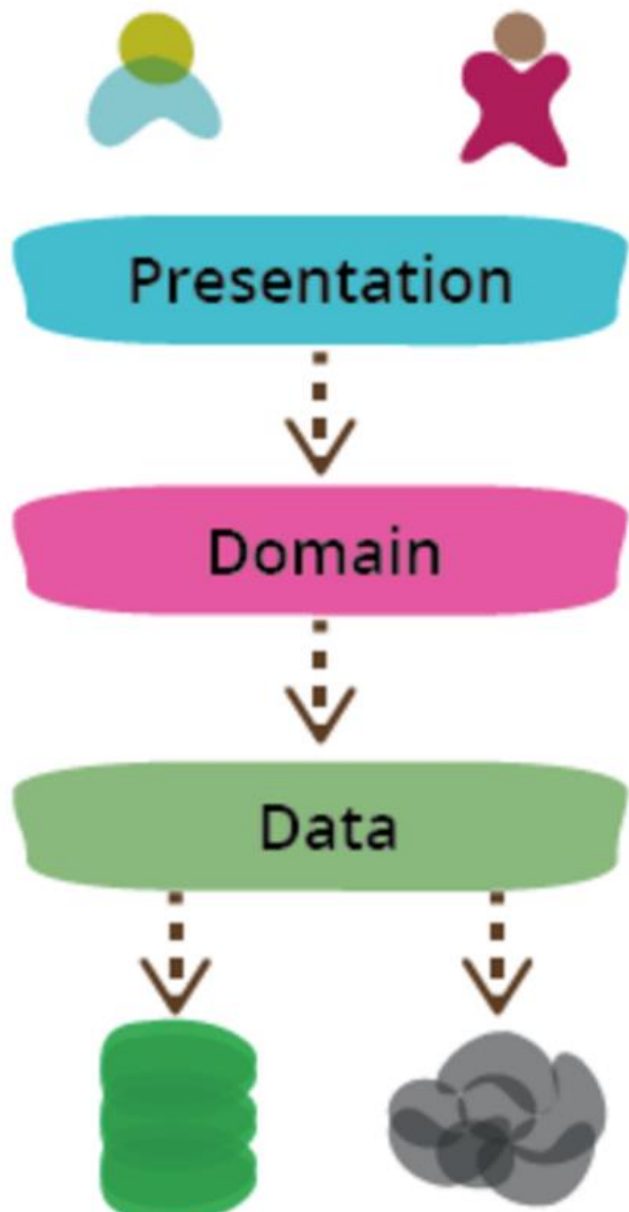


C# ZdravoKorporacija

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

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

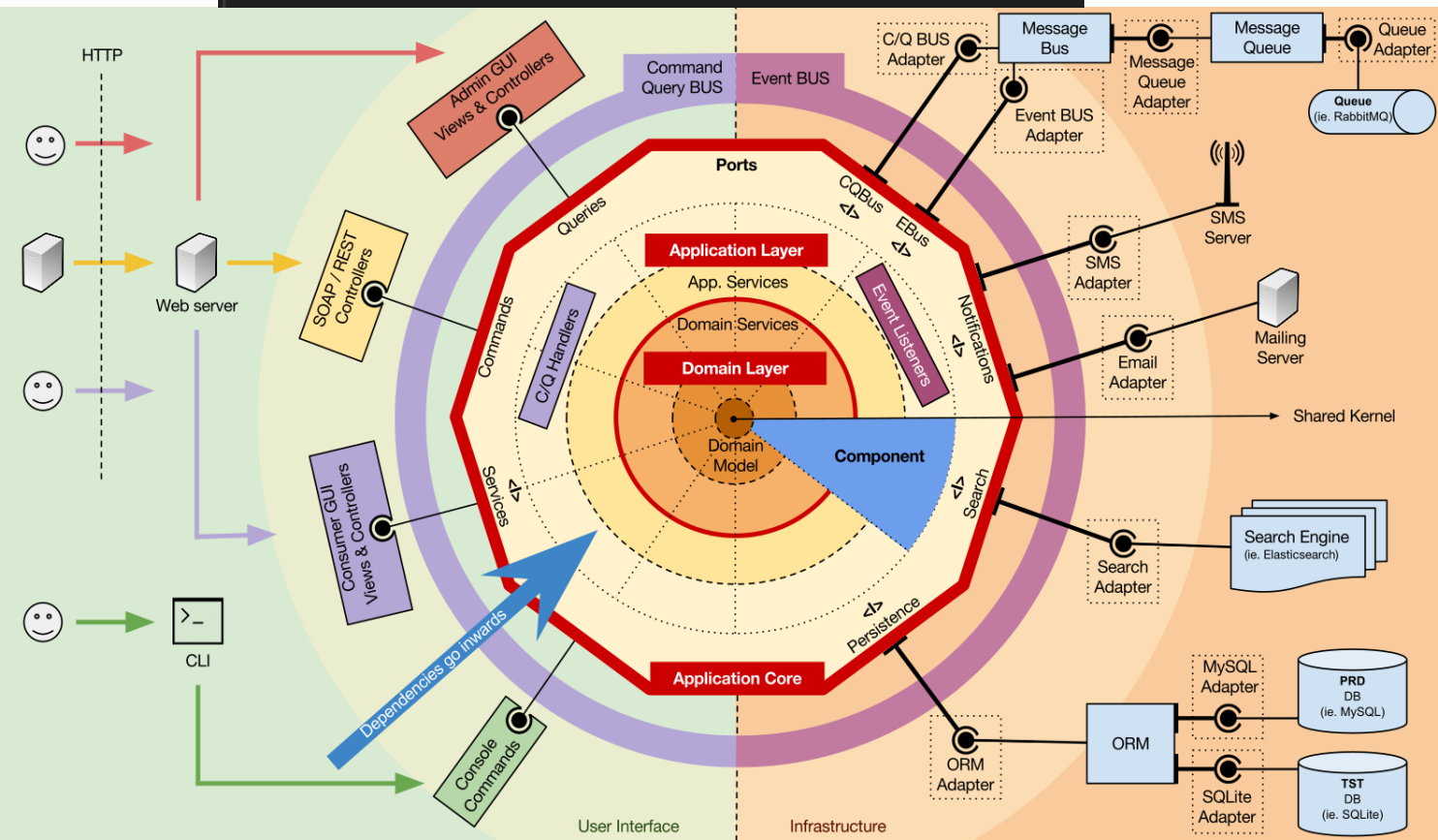
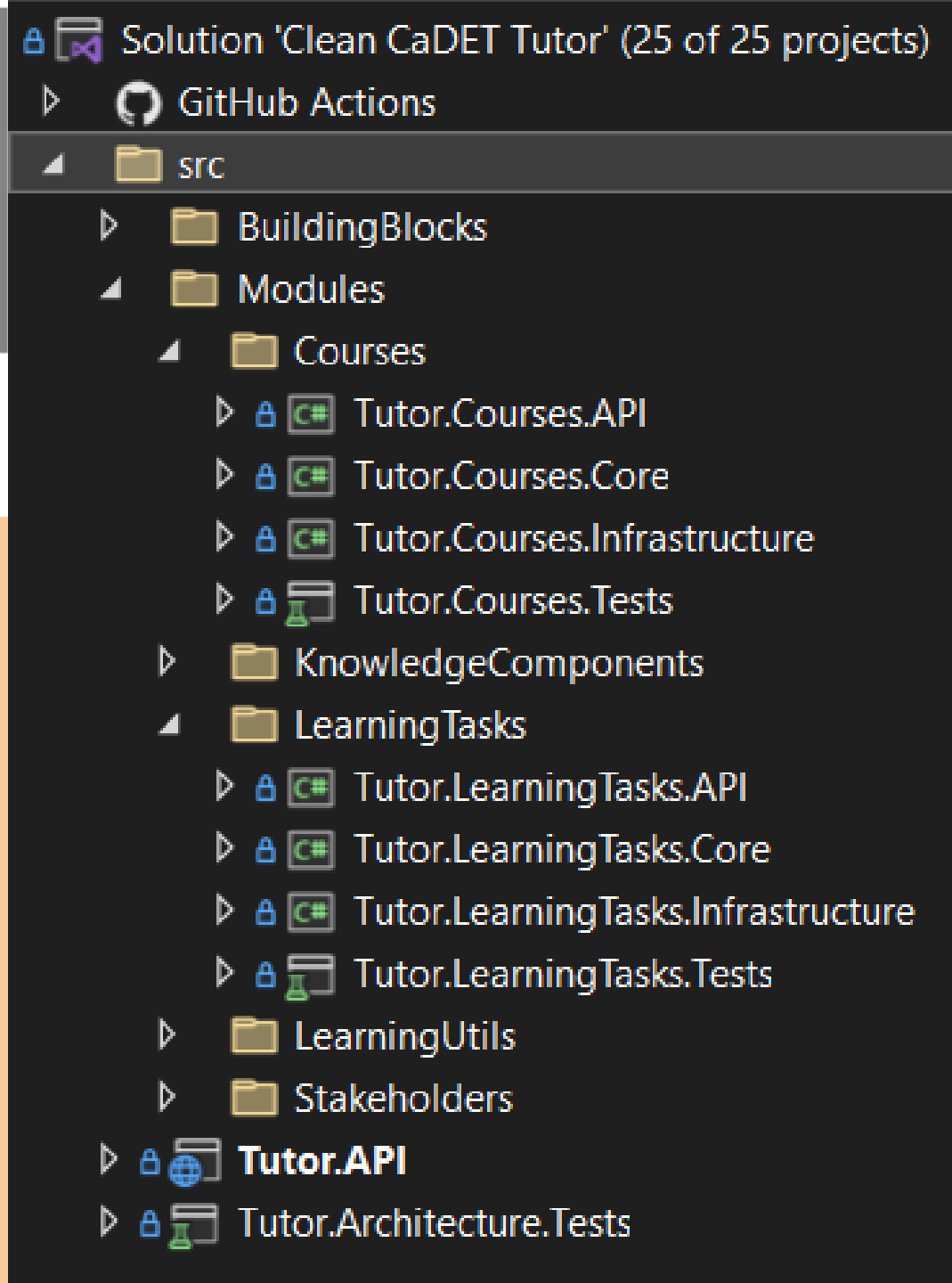
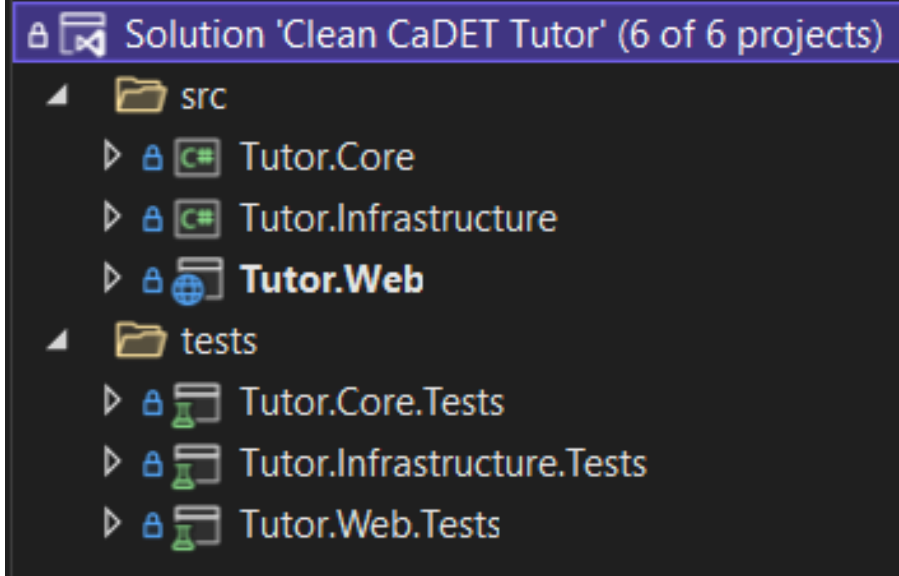
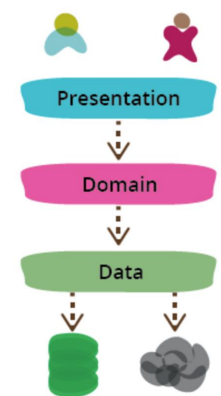
*šta otvaram
ako dodajem
novu funkciju
za pacijente?*



Package by layer

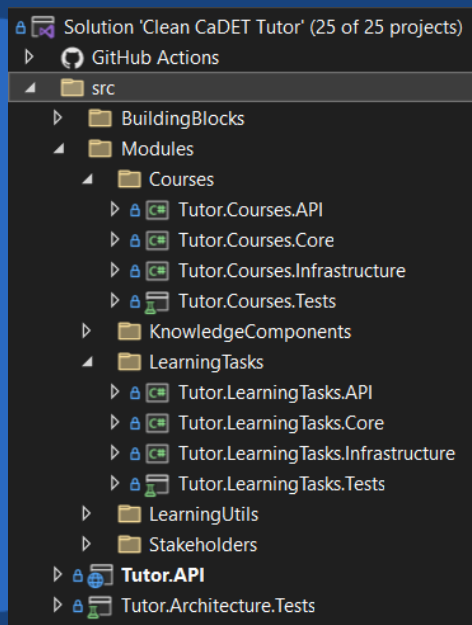


Package by feature



Modularity

**Modular
monolith**



Microservices

**Monolithic
big ball of mud**

**Distributed
big ball of mud**

Number of deployment units