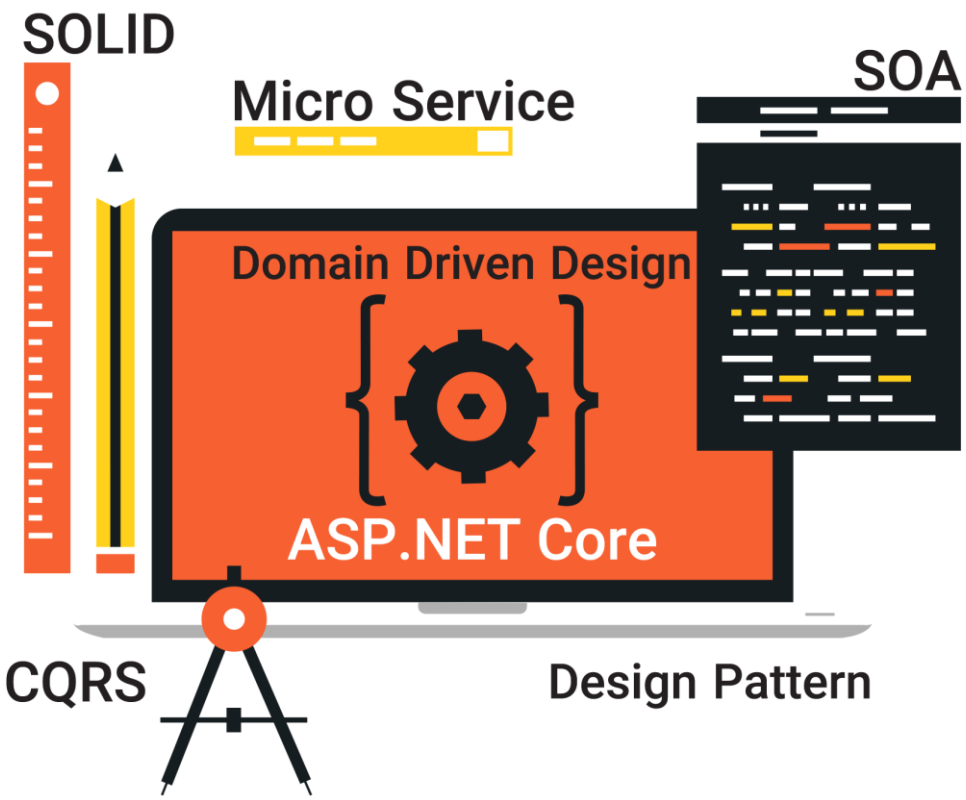


Domain Driven Desgin

Visual Studio

www.NikAmooz.com

مدرس: علیرضا ارومند

nikamooz;
آموزش برنامه نویسی و اجرای پروژه

جلسه اول

معرفی علیرضا ارومند

۱. **مدرس و مشاور** ASP.NET Coreو معماری‌های نرم‌افزاری **(نیک آموز)**

۲. **مدیر فنی خبرگزاری نسیم**

۳. **کارشناس ارشد توسعه نرم افزار داتین (فناپ)**

۴. **کارشناس ارشد توسعه نرم افزار ارتباط فردا (بانک آینده)**

۵. **متخصص انجام پروژه‌های وب و .NET**

۶. **و...**

Introduction

چه خواهیم آموخت؟

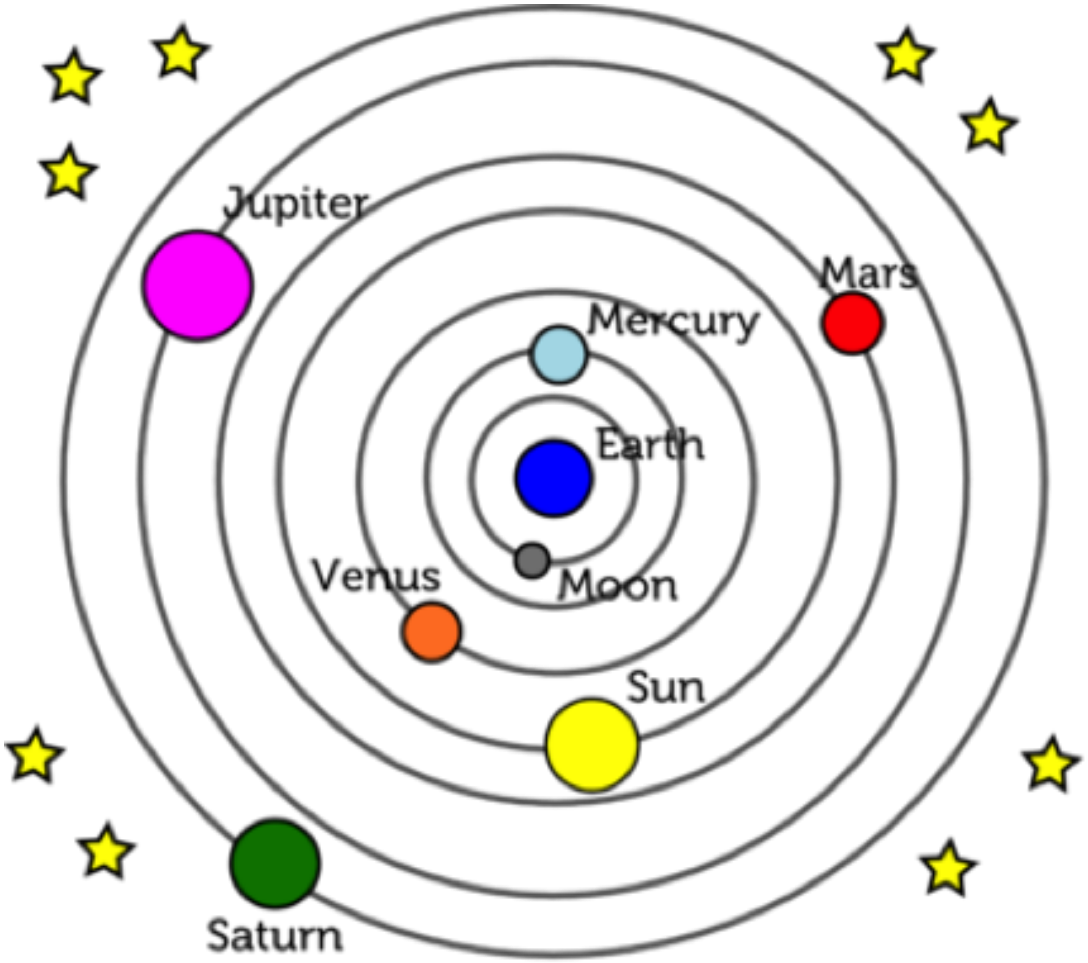
۱. معرفی Domain Centric Architecture

۲. انواع روش‌های پیاده سازی

۳. Application Layerو کاربرد آن

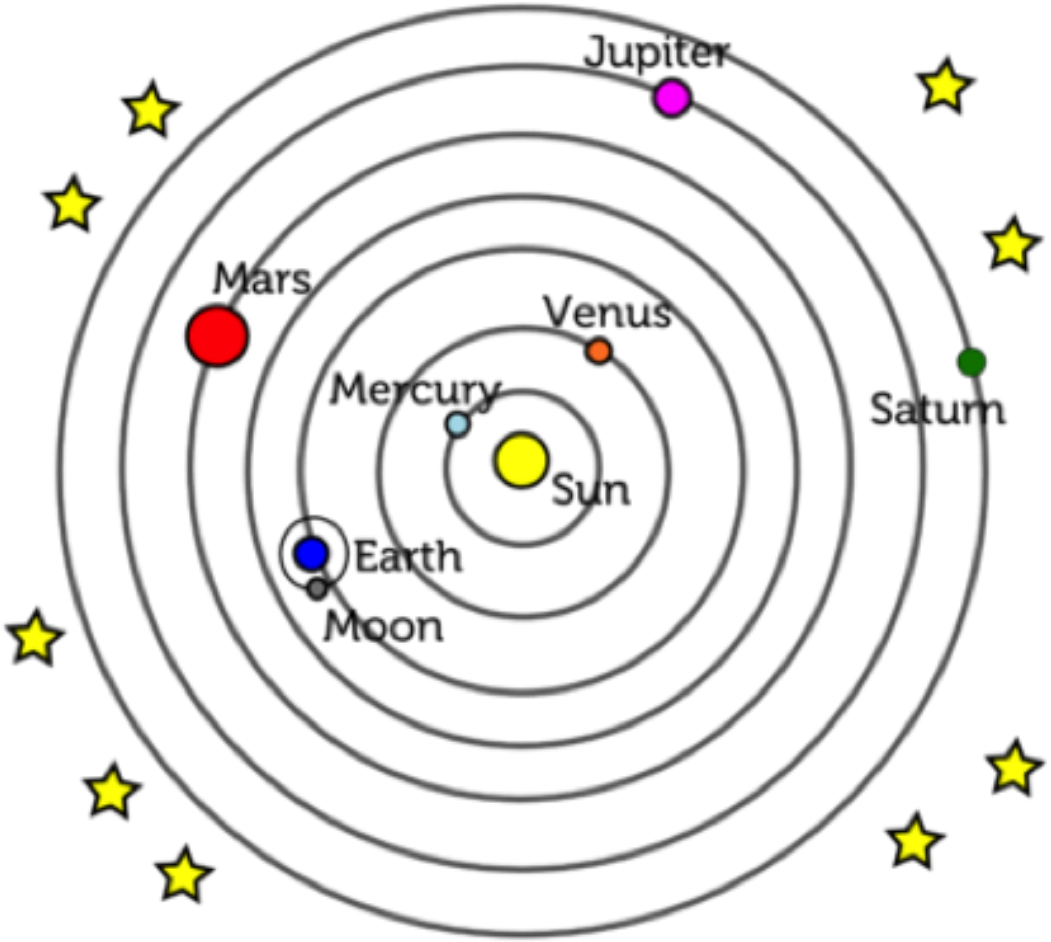
۴. ...

زمین مرکز هستی



Earth at the Center

برخورد با واقعیت توسط کپرنیک

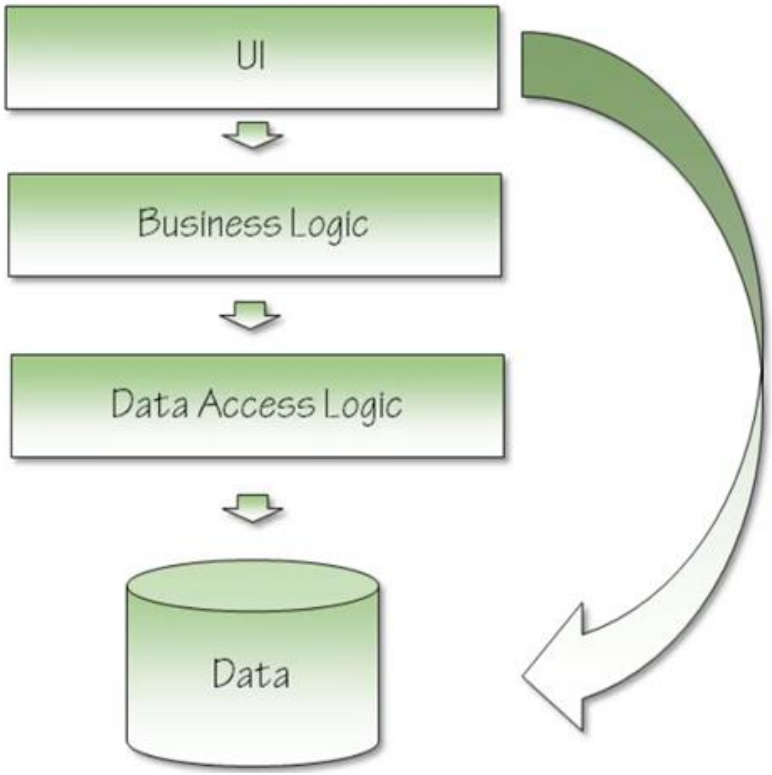


Sun at the Center

معماری سه لایه Data Centric

۱. انتخاب اشتباه مرکز کائنات

۲. دیتابیس مرکز نرم افزار



Domain Centric Architecture

۱. تغییر نگرش در انتخاب هسته نرم افزار

۲. دیتابیس تنها جزییات پیاده سازی است

۳. اصلی نرم افزار منطق تجاری است

هدف معماری

هدف معماری اطمینان از کاربردی بودن ساختمان است، نه اطمینان از استفاده از آجر در ساختن ساختمان ها.

شناسایی موارد کاربردی و جزئیات

۱. فضای موجود در خانه

۲. کاربردی بودن فضاها

۳. ابزار مورد استفاده برای ساخت

۴. اجناس مورد استفاده برای ساخت

موارد کاربردی نرم افزار

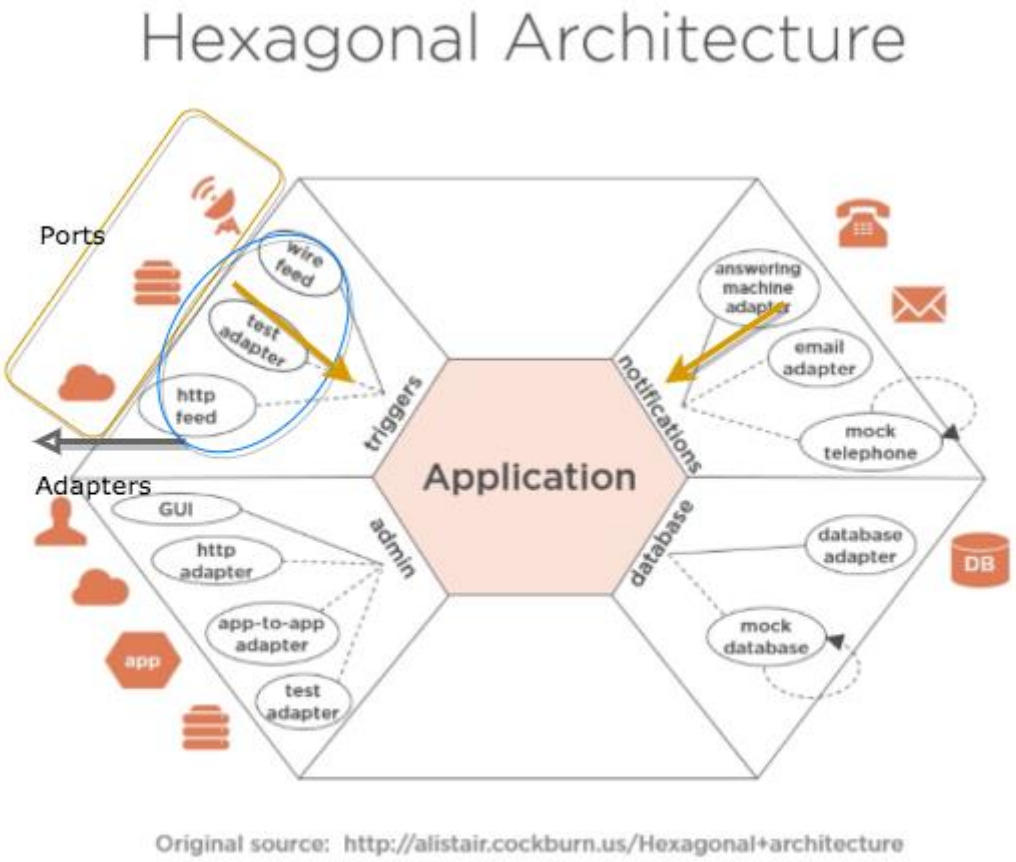
۱. قواعد و قوانین دامنه برنامه

۲. راول های اجرایی منطق برنامه

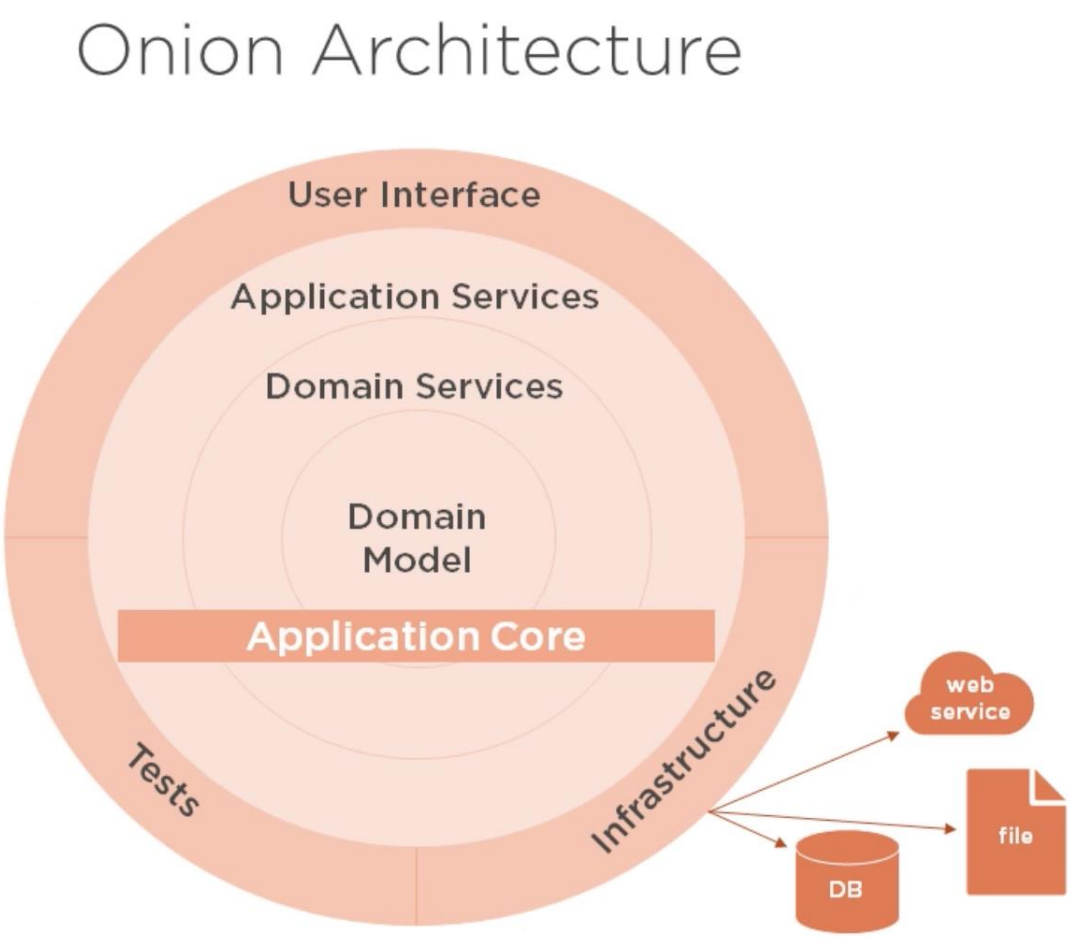
۳. نحوه نمایش و خروجی

۴. نحوه ذخیره و بازیابی

Hexagonal Architecture

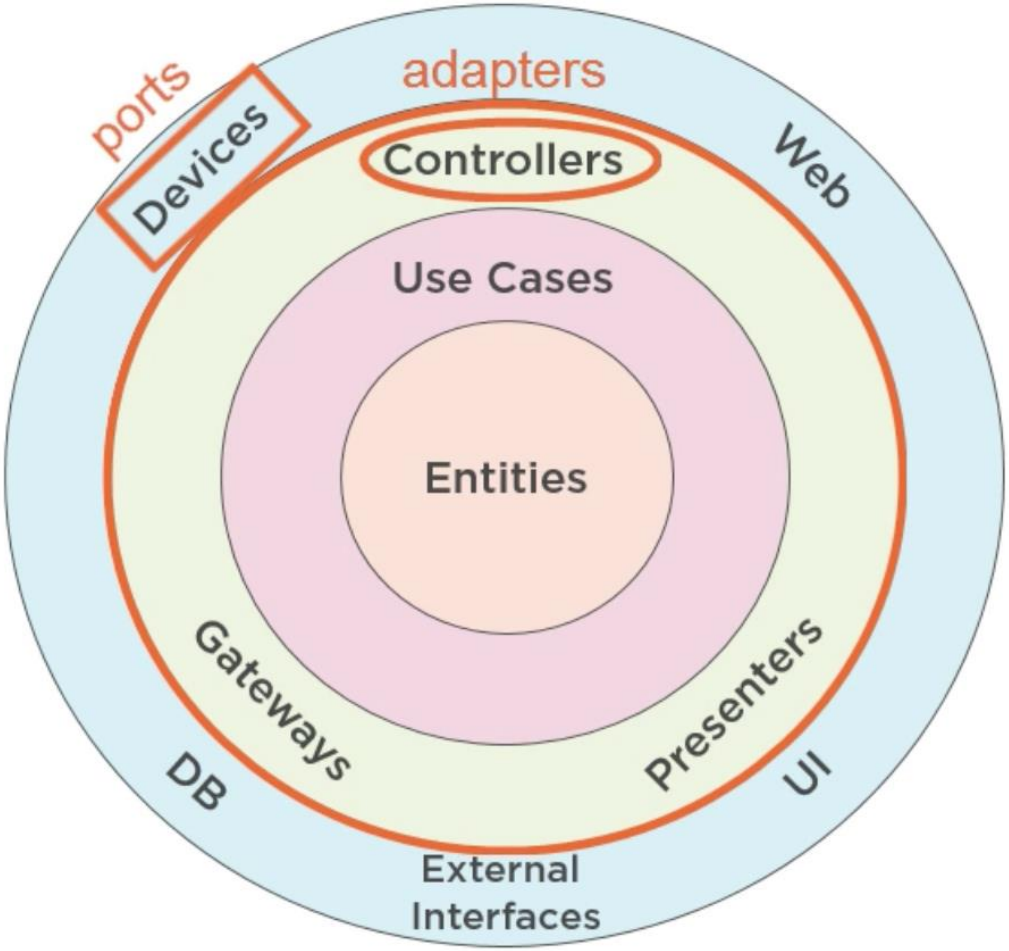


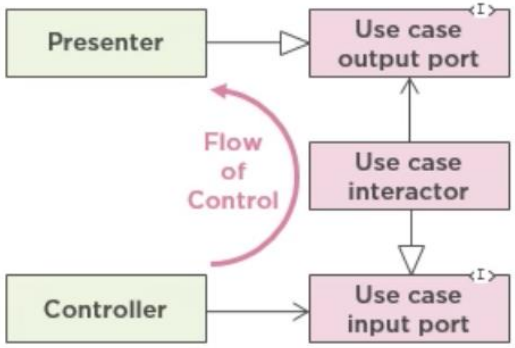
Onion Architecture



Original source: http://jeffreypalermo.com/blog/the-onion-architecture-part-2/

Clean Architecture





Original source: http://blog.8thlight.com/uncle-bob/2012/08/13/the-clean-architecture.html

یک روح در سه کالبد

۱. از نظر ظاهری تفاوت دارند

۲. دامنه در مرکز هر سه قرار دارد

لایه بندی

۱. حفظ سطح انتزاع

۲. سادگی تغییرات

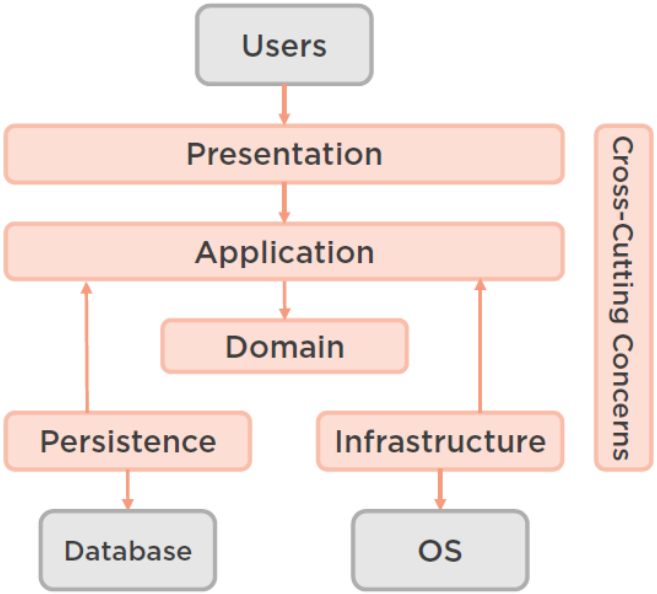
۳. استفاده تخصصی

۴. رعایت اصل SRP

لایه بندی با توجه به دامنه

۱. چهار لایه کاری داریم

۲. هر لایه می‌تواند چند پروژه باشد



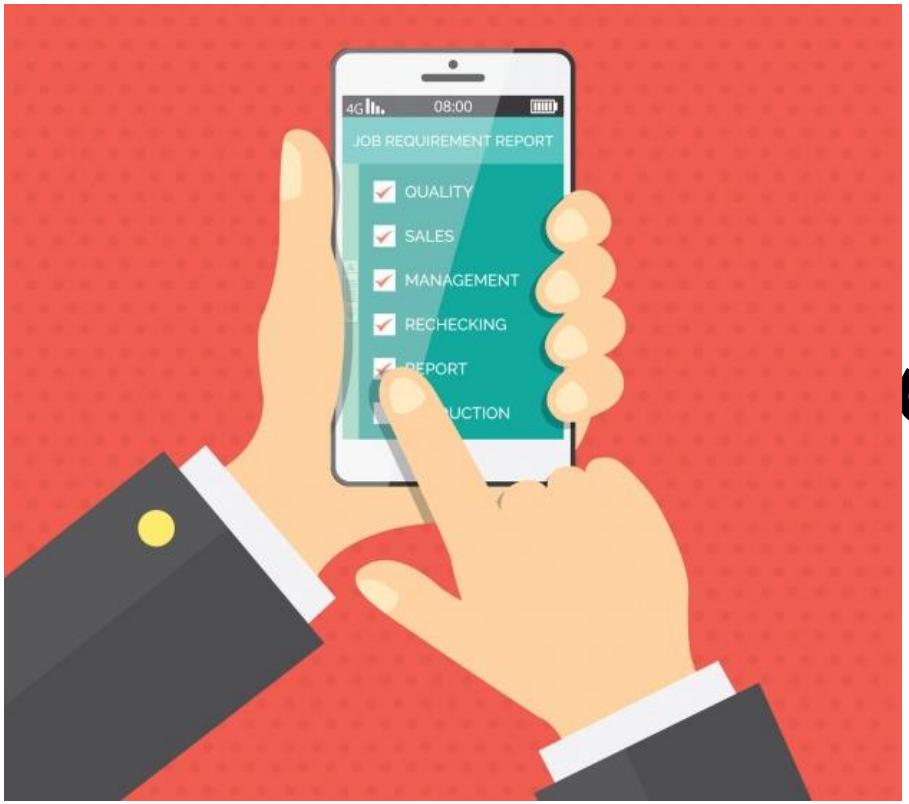
Application Layer

۱. پیاده سازی Use Caseها

۲. وابسته به دامنه است

۳. عدم اطلاع از خروجی

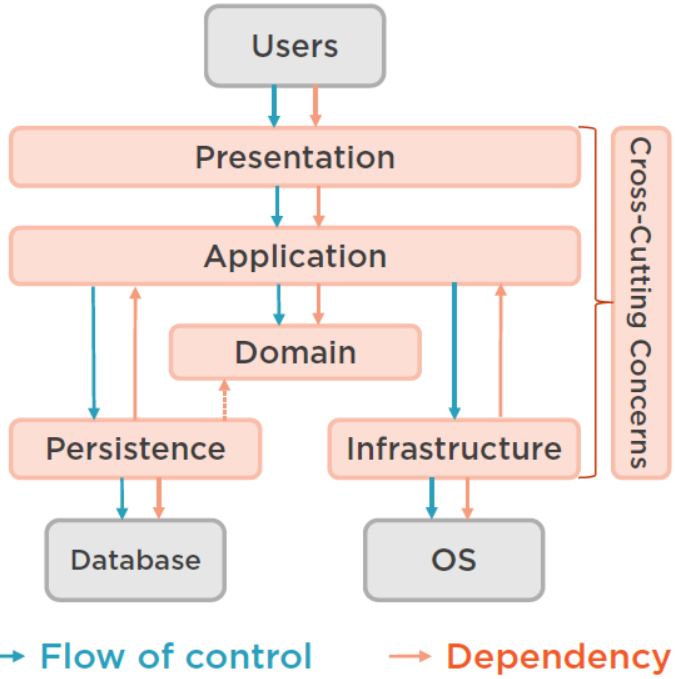
۴. عدم اطلاع از دیتابیس



وابستگی‌ها و جریان کنترل

۱. بعضا متفاوت است

۲. متفاوت از روش سنتی سه لایه



هماهنگی با DIP

۱. نباید به جزییات وابسته باشیم

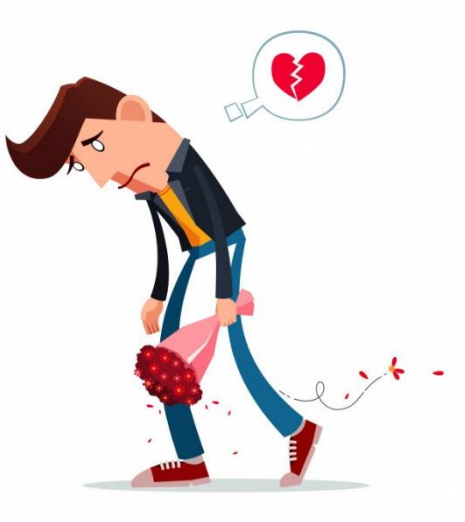
۲. جهت وابستگی را عوض می‌کنیم

Command Query Separation

۱. تقسیم اعمال به دو دسته دستور و واکشی

۲. دستورات تغییر دارند بدون بازگشت

۳. واکشی دریافت اطلاعات بدون تغییر



فرض اشتباه در CQRS

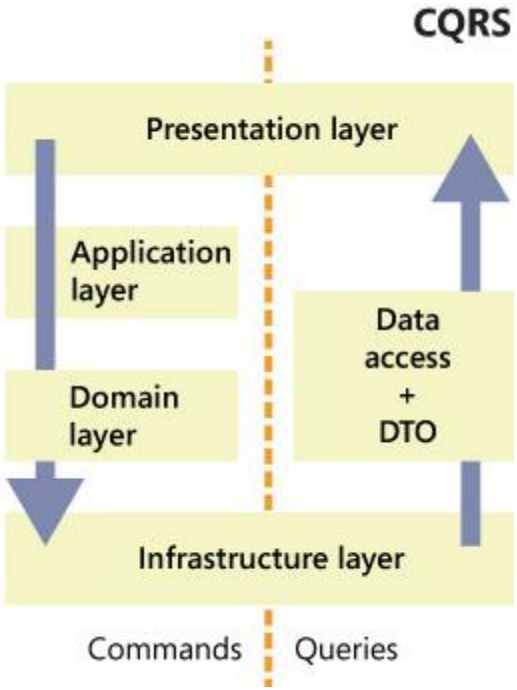
۱. دستورات مقدار بازگشتی دارند

۲. واکشی ها تغییر به همراه دارند

CQRS Architecture

۱. استفاده از جداسازی در معماری

۲. بهینه سازی برای کارهای تخصصی



CQRS Architecture

۱. تقسیم هوشمندانه در مرکز برنامه

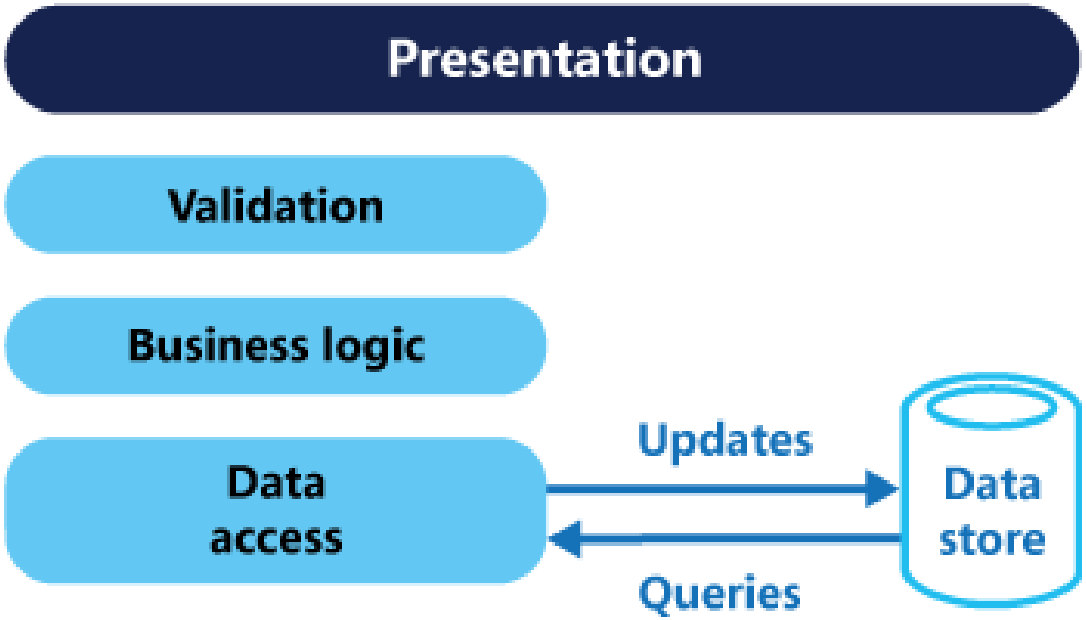
۲. نوشتن توسط قواعد دامنه

۳. واکشی مستقیم

استفاده از یک پایگاه داده

۱. ساده ترین روش پیاده سازی

۲. تنها بهبود معماری

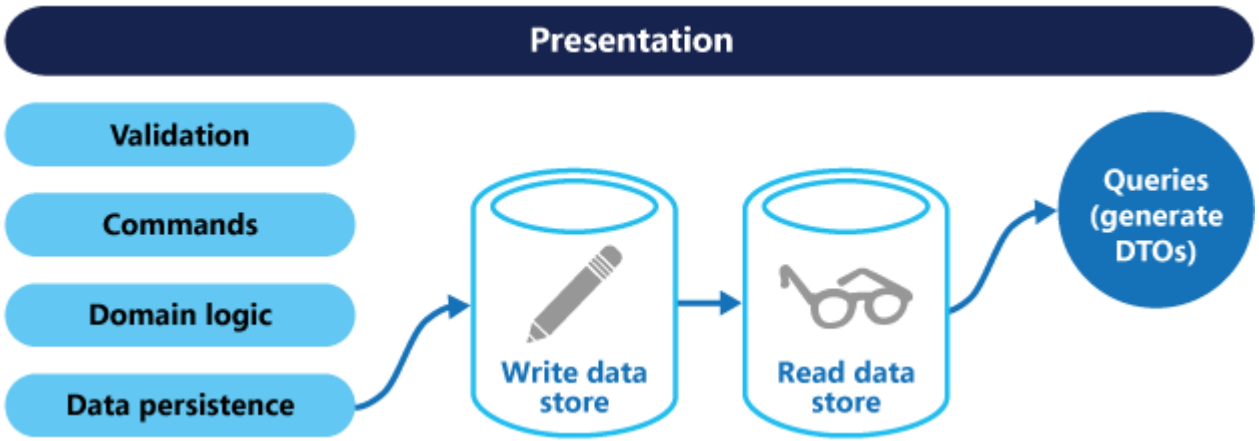


استفاده از دو پایگاه داده

۱. پیچیدگی بیشتر

۲. Eventually Consistency

۳. تخصصی سازی پایگاه داده ها

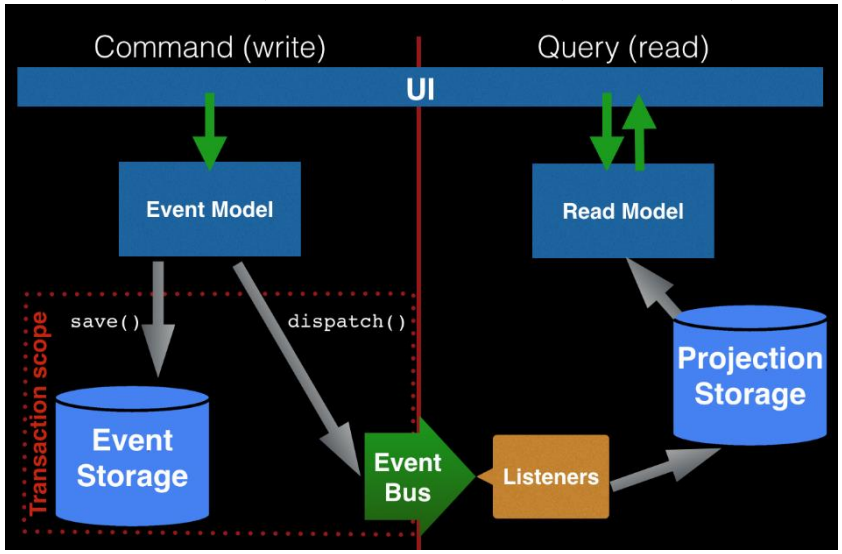


Event Sourcing

۱. پیچیده‌ترین روش پیاده سازی

۲. نگهداری وقایع در دستورات

۳. ساخت وضعیت با اجرای زنجیره دستور



دامنه‌های بزرگ و پیچیده

۱. می‌توان یکپایچه توسعه داد

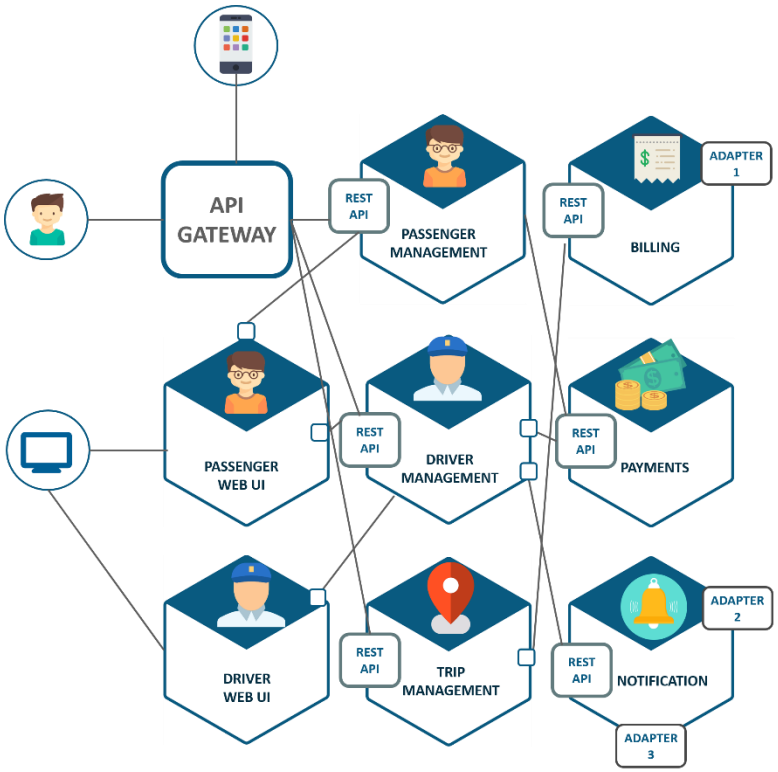
۲. اصطلاحا Monolith

۳. کاملا وابسته است

Micro service

۱. خوردن کردن زیر سیستم‌ها

۲. پیاده سازی تخصصی



صفحه اینستا نیک آموز را فالو کنید

تصاویر آموزشی و آفرهای لحظه‌ای فقط در اینستاگرام نیک آموز

نیک آموز را در اینستاگرام فالو کنید

@nikamooz

در کانال تلگرام نیک آموز عضو شوید

https://telegram.me/nikamooz