

Software Architecture

- سیستم پذیری: سیستم بتواند با افزایش کارایی برآورد کند.
- انعطاف پذیری: که در ساختار طوری باشد که تغییرات را با کمترین
- مدیریت پیچیدگی: پیچیدگی های بزرگ را قابل مدیریت کند.

چرا معماری مهمه؟

معماری به معنای نرم افزار به ساختار کلی سیستم است و باید که دستورالعمل برای مختلف طایر و سازماندهای سیستم
 هدف اصلی معماری نرم افزار اینست که یک طرح جامع ایجاد کند که نیازهای محتمل برای تغییر معماری را برآورده کند.

• Architecture patterns •

راه حل های استاندارد برای مسائل رایج هستند.

لایه ای: کلاسیک، مودر، میکروسروسیس
 مایکروسروسیس: سیستم های کوچک و مستقل هستند.
 کلاسیک مودر: به مرور زمان به کلاسیک ها ختم میشوند.

• Connectors •

راه های ارتباطی میان اجزای سیستم دارند
 مشخص می کنند.

API
 protocols
 زبان ها

• Components •

واحدهای مستقل هستند که
 وظیفه خاصی را انجام میدهند

ماژول ها
 سرویس ها

اصول معماری نرم افزار

* جداسازی وظایف ← هر بخش باید از سایرین جدا باشد و مسئولیت مشخص خودش را داشته باشد. * هماهنگی ← ناحه امکان طراحی و ساده نگه دار بدون اینکه کارایی لازم برود

* ماژولاریتی ← سیستم در به ماژول های کوچک و مستقل تقسیم کن که در کنار یکدیگر کار می کنند. * قابلیت انعطاف ← کماصورت ها طوری طراحی بشوند که در آینده بتوانند با تغییرات
 استقربالند.

* انترایج ← جزییات پیچیده در مخفی کن و فقط چیزهای ضروری در دسترس باشد.
 * ارتفاع پذیری و قابلیت گسترش ← طراحی باید طوری باشد که امکان گسترش و تغییرات در آینده داشته باشد.
 * مدیریت وابستگی ← وابستگی بین بخش ها کم باشد و ارتباطات داخلی در بخش
 در قوی نگردد.

ماژولاریتی

- تقسیم کردن سیستم به تکه های کوچکتر به طوری که مسئولیت مشخص
- حداکثمتا توسط توسعه و نگهداری گونه
- ارتباط با صلبی ماژول ساکن و کنترل شده

چطوری ماژول طراحی کنیم؟

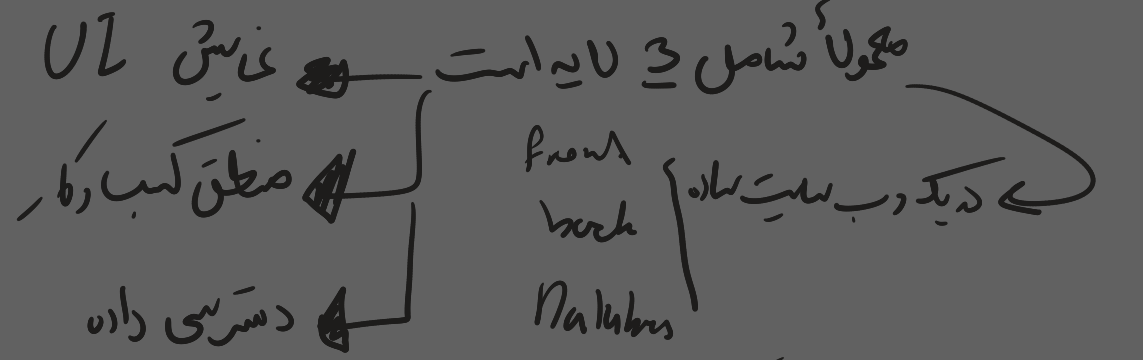
- وظایف رو جدا کنیم : مشخص کنیم که سیستم همراه چه کارهایی انجام بده ، پس هر کار یک ماژول کرد.
- ماژول های مشخصی رو مرتبط کنیم : هر ماژول باید یک راه مشخص با سایرین داشته باشه.
- مدیریت وابستگی ها ، از ابزارهای استفاده کنیم (*Dependency Injection*) که ماژول ها سیستم
- استاندارد ماژولی *Design Pattern* داشته باشن .
نمایندگی از ساختار و نحوه کار ماژول ها با یکدیگر کن.

* هر ماژول برای خوانایی باید سیستم جداست به این معنا که پایگاه داده ... خوانش ندارد . سیستم مربوط به هر ماژول *work flow* خودش رو در این توسعه سریع تر و سوازی صحت راز طرفی مدیریت سخت تر خواهد شد.

انواع صحای نرم افزار

۱- لایه ای: تقسیم سیستم به لایه های مستقل از یکدیگر که هر کدام مسئولیت خاص خود را دارند

و تنها بالایه های مجاوره ارتباط هستند.



* کاربرد: پروژه های کوچک تا متوسط، مثل سیستم مدیریت محتوا.

۲- مبتنی بر رویداد: یک بخش رویداد تولید میکند بخش های دیگر بهش واکنش نشون میدن. مثل یک سیستم حس که وقتی یک کاربر پیام تولید میکنه بقیه کاربرها رویداد پیام جدید (احسان) میکنند.

* برنامه های Real-Time، مثل IoT

۳- تک سگتی: همی اجرای سیستم یک برنامه ی راحت تر دارند.

ال ر منطقی و... همه رو با هم نوی یک برنامه Merge کن:

* پروژه های کوچک یا استارت آپ های تازه شروع کردند.

۴- سرویس گرا: سیستم از سرویس های مستقل تشکیل شده از طریق پروتکل های استاندارد (REST)

یا SOAP) باهم ارتباط دارند.

* سیستم های بزرگ سازمانی

۵- مایکروسرویس: سیستم بر سرویس های کوچکتر تقسیم می شه که هر کدام بطور مستقل کار خودتونو انجام دادن و از طریق API به هم ارتباط هستند.

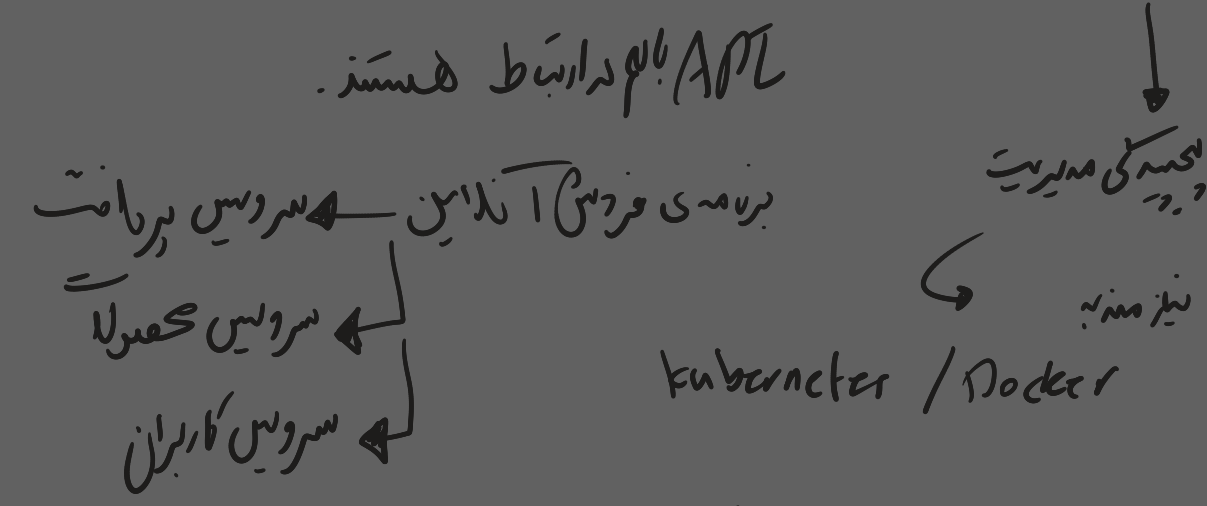
* مناسب برای سیستم های بزرگ یک API که فقط وقتی کاربرد خواست می شه ابرار می شه.

* اپلیکیشن های صغیر پذیر با تعداد صغیر

آنها را شروع کردی و پروژه کوچیکه تک سگتی / لایه ای

به حال توسعه ای یک سیستم بزرگ و معیانی پذیری مایکروسرویس / بدون سرور

نیاز منه به پاسخ گوی سریع مبتنی بر رویداد



۶- توانیت سرور: یک سرور مرکزی خدمات در ارائه می ده و توانیت ها از بی درخواستی کن.

برنامه ها این که توانیت (برنامه موبایل) با سرور ارتباط دارند.

* کاربرهائی نوی دب سالت ها، سیستم های بانی است.

مقایسه سریع	مقیاس پذیری	سادگی	پیچیدگی مدیریت	کاربرد مثال
لایه ای	متوسط	بالا	کم	وب سایت ساده
مایکروسرویس	بالا	متوسط	بالا	پلتفرم های بزرگ
کلانت-سرور	متوسط	بالا	متوسط	اپلیکیشن های وب
تک سگتی	کم	بالا	کم	پروژه های کوچک
بدون سرور	بالا	متوسط	کم	API های مقیاس پذیر

