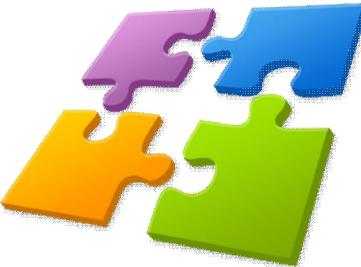


سرویس ها در نرم افزار

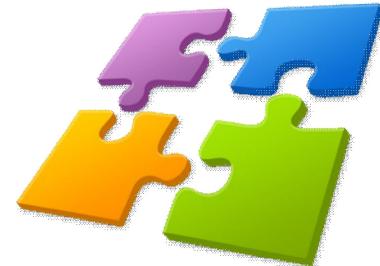
- سرویس
 - فرد یا سازمانی که کاری را برای فرد یا سازمان دیگر انجام می دهد.
 - اداره پست
- مزایای سرویس
 - صرفه جویی در زمان، پول و تلاش فرد نیازمند سرویس
 - در غیاب سرویس ممکن است، کار مورد نظر هیچ گاه انجام نشود
 - افزایش کارایی
 - قابلیت استفاده مجدد (یعنی این سرویس توسط افراد و سازمان های زیادی قابل استفاده است)
 - شفافیت سرویس برای درخواست کننده سرویس



سرویس ها در نرم افزار ...

- سرویس در نرم افزار
- یعنی برنامه ای که کاری را برای برنامه دیگر انجام می دهد.
- این کار ممکن است شامل برخی از کارکردها یا به اشتراک گذاری داده ها باشد.
- در بیشتر مواقع این برنامه های کاربردی به صورت **توزیع شده** اجرا می شوند.
- یعنی برنامه کاربردی ارائه دهنده سرویس، و برنامه کاربردی مصرف کننده سرویس روی ماشینهای مختلفی که از طریق شبکه بهم متصل شده اند، اجرا می شوند.
- حتی اگر در مواردی این دو برنامه هر دو روی یک ماشین اجرا شوند، باز هم شیوه برقراری ارتباط در آنها مشابه همان شیوه توزیع شده است.
- ایده سرویس در نرم افزار، در طول چندین دهه رشد کرده است. منشا اصلی این تکامل رایانش توزیع شده است اما تکامل زبان های برنامه نویسی نیز بی تاثیر نبوده است.

سرویس ها در نرم افزار ... تکامل زبانهای برنامه نویسی



• زبان های ترتیبی (Sequential Languages)

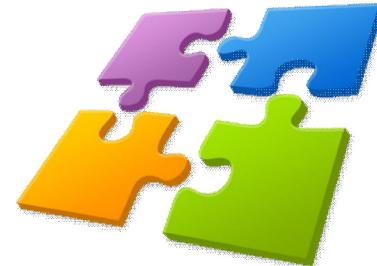
- دستورات را به ترتیب اجرا می کردند.
- امکان قابلیت استفاده مجدد از کدها وجود نداشت.
- نگهداری کد، مشکل بود.
- عملکرد این زبانها در استفاده از حافظه بسیار ناکارا بود.

• زبان های رویه ای (Procedural Languages)

- ابتدایی ترین شکل مفهوم سرویس، در این زبانها بوجود آمد: تابع (متدها / زیرروال)
- افزایش قابلیت استفاده مجدد
- افزایش قابلیت نگهداری
- استفاده کاراتر از فضای حافظه

سرویس ها در نرم افزار ...

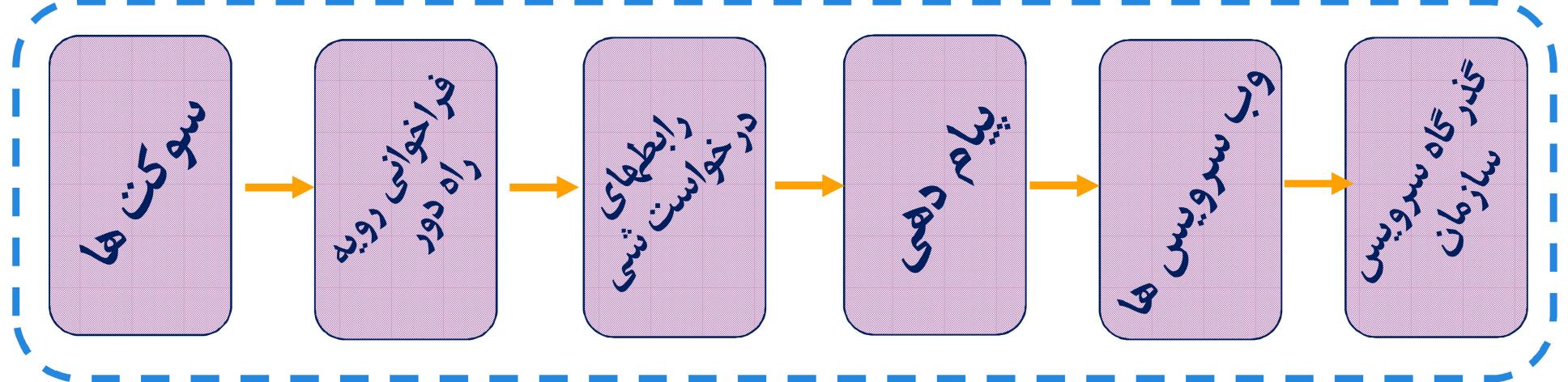
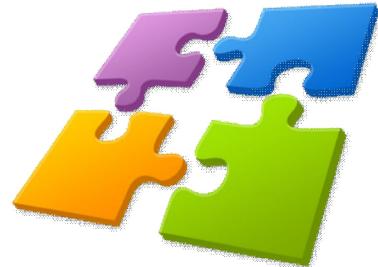
تکامل زبانهای برنامه نویسی ...



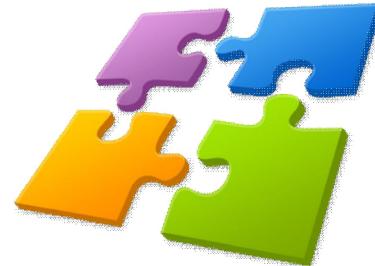
• زبان های شی گرا (Object Oriented Languages)

- مفهوم کلاس معرفی شد که حاوی داده و رفتار کپسوله شده است.
- افزایش قابلیت استفاده مجدد
- اگرچه معرفی زبانهای رویه ای و شی گرا استفاده مجدد از کد را افزایش داد، اما این استفاده مجدد به برنامه های اجرایی یا برنامه های کامپیوتری شخصی محدود میشد. به عبارت دیگر، رویه در خارج از برنامه دربرگیرنده خود، قابل استفاده مجدد نبود.
- بنابراین، این توسعه در زبان های برنامه نویسی مستقیما منجر به توسعه سرویس ها آن گونه که امروز شناخته می شوند نشد، بلکه منشا اصلی و بنیادی تر توسعه سرویس و معماری سرویس گرا، ناشی از رایانش توزیع شده است.

معماری سرویس گرا

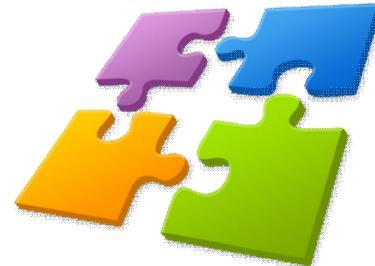


تکامل معماری سرویس گرا و یکپارچه سازی مبتنی بر سرویس ها



معماری سرویس گرا...

- معماری سرویس گرا به یک مساله کسب و کاری متداول و خاص می پردازد.
- در گذشته نیازمندی های کسب و کار به سرعت تغییر نمی کردند. خط محصول عرضه شده توسط یک شرکت و شیوه های بازاریابی و فروش این محصولات ثابت بودند. بنابراین نیازمندی های فناوری اطلاعات نیز کمابیش ثابت بودند.
- در دهه ۱۹۹۰ این وضعیت تغییر کرد. طول عمر یک محصول کوتاه تر شد و سازمان به سرعت شروع به تغییر کرد.
- دلایل اصلی این تغییرات عبارت است از:
 - ادغامات، تملک ها، شرایط متغیر بازار، پیشرفت های تکنولوژیک جدید، ماهیت روابط کسب و کار



معماری سرویس گرا...

• ادغامات

- دو یا چند شرکت یا سازمان به هم ملحق میشوند تا یک شرکت واحد جدید را تشکیل دهند.

• تملک ها

- یک شرکت اندازه خود را با خرید یا به دست آوردن شرکتی دیگر افزایش میدهد.

• شرایط متغیر بازار

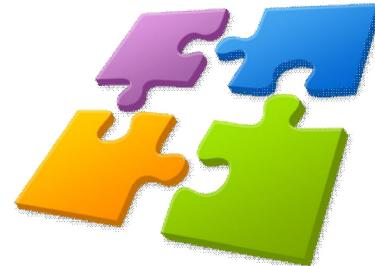
- معرفی سریع محصولات جدید و بسته بندی مجدد محصولات موجود به منظور بقا در بازار به شدت رقابتی

• پیشرفت های تکنولوژیک جدید

- پیشرفت ها مانند اینترنت و سیستم پاسخ صوتی، فرصت های جدیدی برای بازاریابی، فروش و تدارکات فراهم میکند.

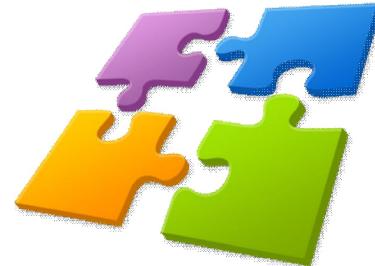
• ماهیت روابط کسب و کار

- یک سازمان بزرگ معمولاً روابط زیادی با نهادهای خارجی کسب و کار مانند شرکای کسب و کار و تامین کنندگان دارد. این روابط ماهیتا جریان دارند و مدام در حال تغییر هستند.



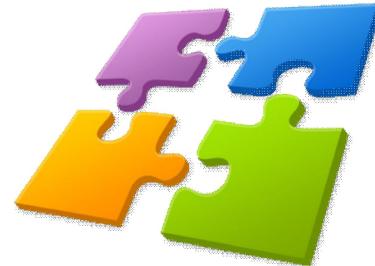
معماری سرویس گرا...

- شرایط به شدت متغیر کسب و کار، به این معنی است که نیازمندی های سیستم های فناوری اطلاعات که از این کسب و کارها پشتیبانی می کنند، نیز به سرعت شروع به تغییر کردند.
- در گذشته هر برنامه کاربردی برای پاسخ به یک نیاز خاص کسب و کار توسعه داده میشد و مسائل جدید به توسعه برنامه های جدید نیاز داشت.
- اما نیازمندی های به شدت متغیر فناوری اطلاعات به این معناست که شیوه های قدیمی توسعه نرم افزار، دیگر مناسب نیستند، و رویکردی جدید برای ارائه سیستم های انعطاف پذیر و چابک فناوری اطلاعات نیاز بود تا بتواند نیازهای به شدت متغیر کسب و کار را تامین کند.



معماری سرویس گرا...

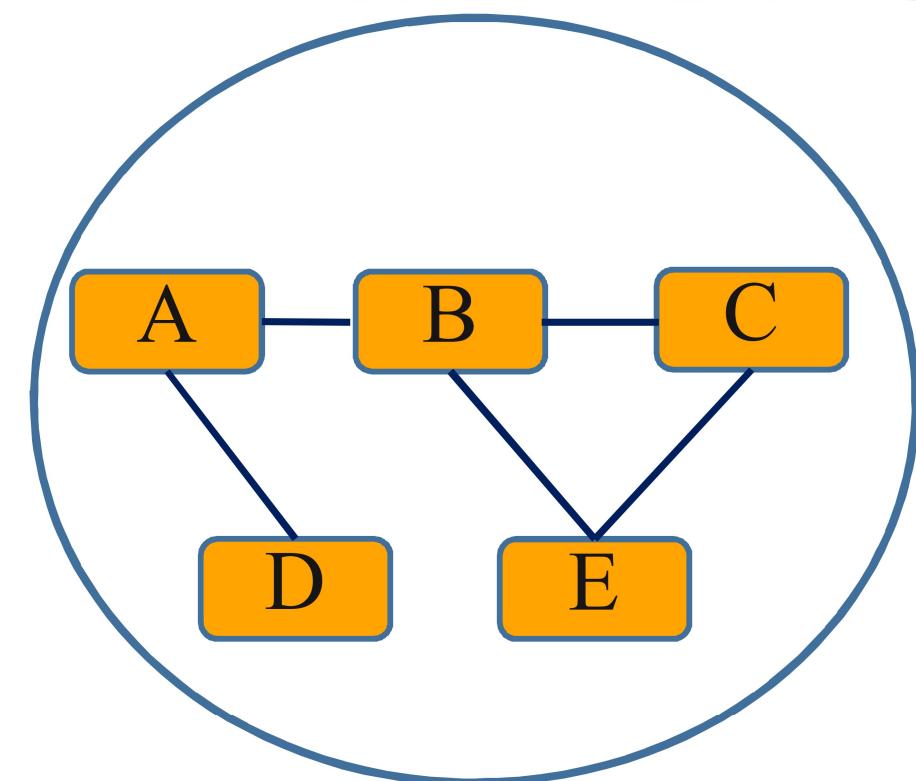
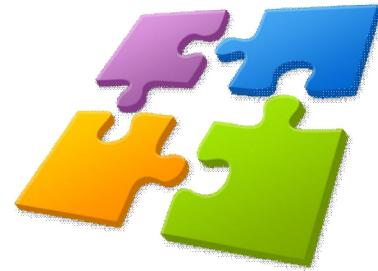
- معماری سرویس گرا، به عنوان پاسخی به این مساله روی سیستم های چابک فناوری اطلاعات با استفاده از مولفه های قابل استفاده مجدد، تاکید میکند.
- در این معماری مولفه ها یا برنامه های کامپیوتری برای حل مساله کسب و کار خاصی توسعه داده نمیشوند. در عوض تعدادی کارکرد عمومی ارائه می دهند.
- این مولفه ها میتوانند با هدف خاصی به یکدیگر متصل شوند، یا یکپارچه شوند یا برای تامین نیاز کسب و کاری خاصی پیکربندی شوند.



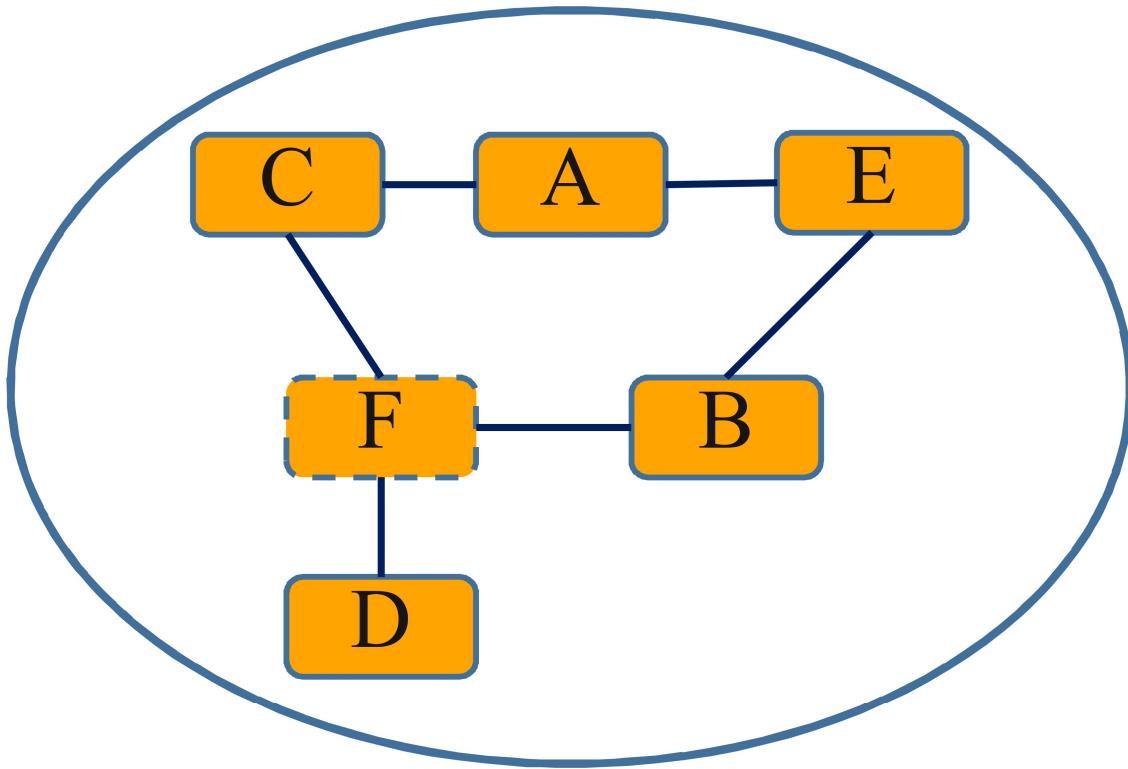
معماری سرویس گرا...

- اگر نیازمندی های کسب و کار تغییر کند، نیازی به توسعه یک برنامه کامپیوتری جدید نیست. سیستم میتواند مجددا پیکربندی شود تا این نیاز جدید را تامین کند.
- معماری سرویس گرا، روی چابکی و استفاده مجدد از دارایی های نرم افزار با استفاده از مولفه های نرم افزاری که میتوانند به آسانی با پیکربندی های مختلف مجددا متصل شوند، تاکید میکند.

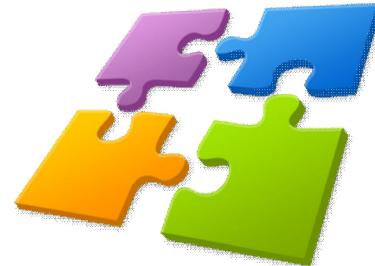
معماری سرویس گرا...



پیکربندی اولیه

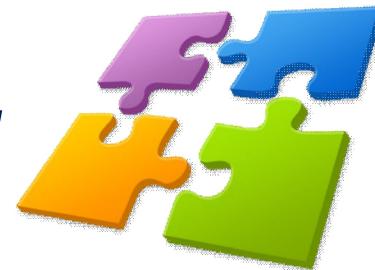


پیکربندی ثانویه



معماری سرویس گرا...

- مزایای مولفه های قابل استفاده مجدد
- در زمان و منابع توسعه و تست صرفه جویی می کنند، زیرا در صورت تغییر نیازمندی ها بیشتر مولفه های جدید یا حتی گاهی، هیچ یک از آنها نیاز به توسعه ندارند.
- با حذف کد تکراری، عملکرد و داده های سازگارتری را به مصرف کنندگان داخلی و خارجی ارائه می دهند.
- نگهداری کد آسان است، زیرا تغییرات می تواند در جایی از یک مولفه متمرکز شود.
- کد به خوبی تست می شود، زیرا بارها در آرایش ها و موقعیت های مختلف مورد استفاده قرار می گیرد.



تعاریف و مفاهیم پایه

- **برنامه کاربردی**

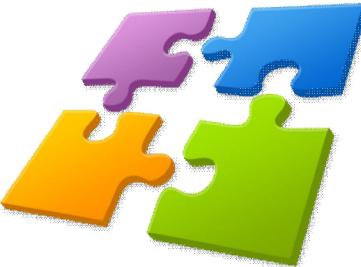
- یک برنامه کامپیوتری واحد = یک برنامه اجرایی واحد

- **رایانش توزیع شده**

- چیزی بیش از یک برنامه کاربردی
- این برنامه ها معمولاً روی سخت افزار یا ماشین های مجزا اجرا میشوند اما با یکدیگر در ارتباطند تا به برخی از کارکردها دست یابند.

- **سازمان**

- هر گونه تشکیلات بزرگ و پیچیده ای که سیستم فناوری اطلاعات بزرگ و پیچیده ای نیز دارد.



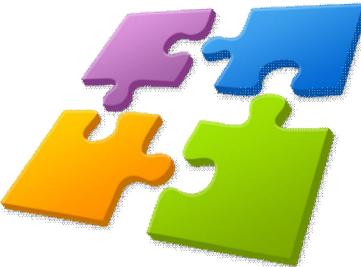
تعاریف و مفاهیم پایه...

• نرم افزار سازمان

- نرم افزاری که برای سازمان بزرگی طراحی شده که معمولاً تشکیلات، فرآیندها، و مدل کسب و کار داخلی خود را دارد.
- این نرم افزار باید وابستگی های بین شعبه ای و روابط کسب و کار خارجی مانند شرکای کسب و کار و فروشنده‌گان خارجی را نیز در نظر بگیرد. بنابراین این نرم افزار باید تعداد زیادی از انواع مختلف نیازمندی‌ها را تامین کند. بسیاری از این نیازمندی‌ها، نا تمام، مبهم و حتی مخالف یکدیگرند. و دائماً در حال تغییرند. به همین دلیل نرم افزار سازمان معمولاً بسیار پیچیده است.

• یکپارچه سازی

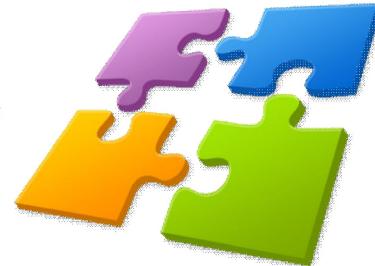
- طراحی برنامه‌های کاربردی جدید یا اصلاح برنامه‌های کاربردی موجود برای به اشتراک گذاری داده‌ها و عملکردها



تعاریف و مفاهیم پایه...

• معماری نرم افزار

- در یک شرکت یا سازمان کوچک به تعداد کمی برنامه کامپیوتری نیاز است. مدیریت این برنامه های کوچک آسان است و نیازی به یک طراحی کلی ندارند.
- هر چه سازمان های بزرگتری را در نظر بگیریم، تعداد برنامه های کامپیوتری بیشتر میشود و نیاز بیشتری هم به یک طراحی کلی یا راهبرد طراحی به منظور اجتناب از آشفتگی وجود دارد. این طراحی کلی یا راهبرد طراحی، معماری نرم افزار نامیده میشود.
- معماری نرم افزار ماهیت مشابهی با معماری ساختمان دارد.
- هر دو نیاز دارند تا طبق قواعدی برای نیازهای فعلی و نیازهای آتی برنامه ریزی کنند.



تعاریف و مفاهیم پایه...

• پیوند ضعیف / پیوند قوی

• منظور از پیوند، ارتباط بین مولفه های مختلف یک سیستم است.

پیوند قوی	پیوند ضعیف	عامل
اتصال مستقیم	اتصال غیر مستقیم از طریق یک واسط	اتصال فیزیکی
متقارن	نامتقارن	سبک ارتباطی
سیستم قوی	سیستم ضعیف	نوع سیستم
منطق متتمرکز	منطق توزیع شده	الگوی تعامل
اتصال ایستا	اتصال پویا	اتصال سرویس