

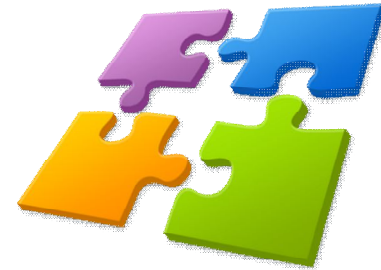
یکپارچه سازی کاربردهای سازمانی

فصل اول: مفاهیم اولیه

بتول لکزائی

b_lakzaei@yahoo.com

سرویس ها در نرم افزار



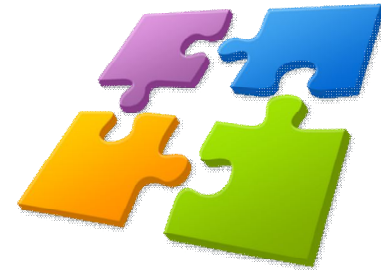
• سرویس

- 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

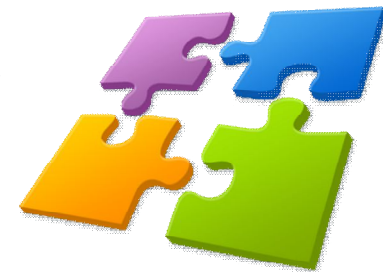
سرویس ها در نرم افزار ...



- **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)

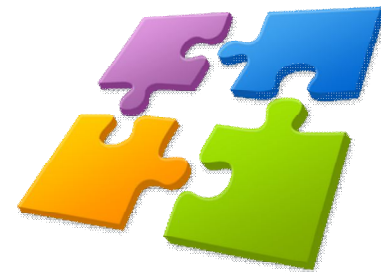
- دستورات را به ترتیب اجرا می کردند.
- امکان قابلیت استفاده مجدد از کدها وجود نداشت.
- نگهداری کد، مشکل بود.
- عملکرد این زبانها در استفاده از حافظه بسیار ناکارآمد بود.

• زبان های رویه ای (Procedural Languages)

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

سرویس ها در نرم افزار ...

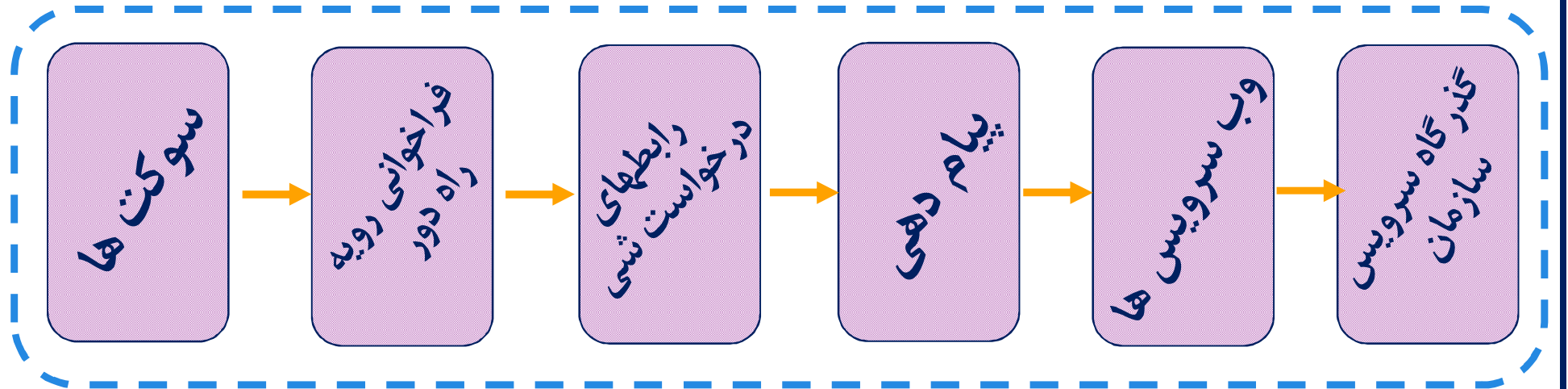
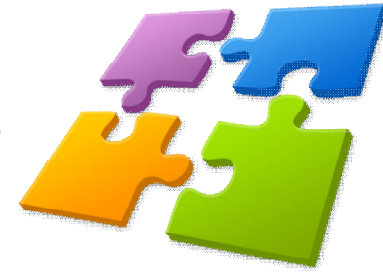
تکامل زبانهای برنامه نویسی ...



• زبان های شی گرا (Object Oriented Languages)

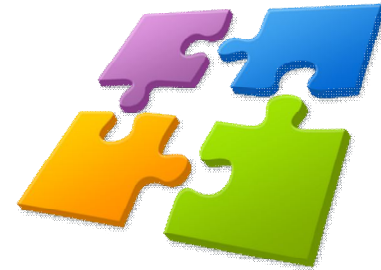
- مفهوم کلاس معرفی شد که حاوی داده و رفتار کپسوله شده است.
- افزایش قابلیت استفاده مجدد
- اگرچه معرفی زبانهای رویه ای و شی گرا استفاده مجدد از کد را افزایش داد، اما این استفاده مجدد به برنامه های اجرایی یا برنامه های کامپیوتری شخصی محدود میشد. به عبارت دیگر، رویه در خارج از برنامه دربرگیرنده خود، قابل استفاده مجدد نبود.
- بنابراین، این توسعه در زبان های برنامه نویسی مستقیماً 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



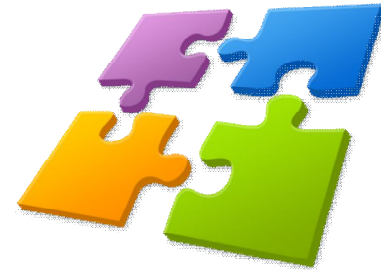
تکامل معماری سرویس گرا و یکپارچه سازی مبتنی بر سرویس ها

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

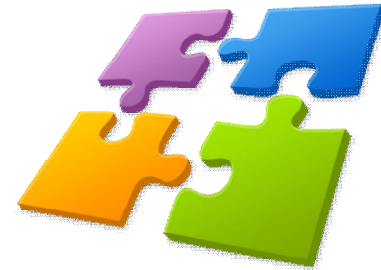
• پیشرفت های تکنولوژیک جدید

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

• ماهیت روابط کسب و کار

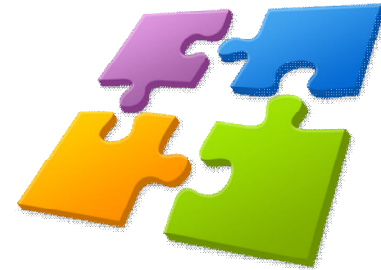
- یک سازمان بزرگ معمولاً روابط زیادی با نهادهای خارجی کسب و کار مانند شرکای کسب و کار و تامین کنندگان دارد. این روابط ماهیتاً جریان دارند و مدام در حال تغییر هستند.

معماری سرویس گرا...



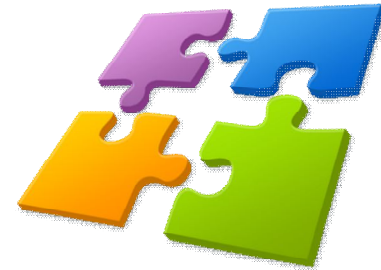
- شرایط به شدت متغیر کسب و کار، به این معنی است که نیازمندی های سیستم های فناوری اطلاعات که از این کسب و کارها پشتیبانی می کنند، نیز به سرعت شروع به تغییر کردند.
- در گذشته هر برنامه کاربردی برای پاسخ به یک نیاز خاص کسب و کار توسعه داده میشد و مسائل جدید به توسعه برنامه های جدید نیاز داشت.
- اما نیازمندی 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...



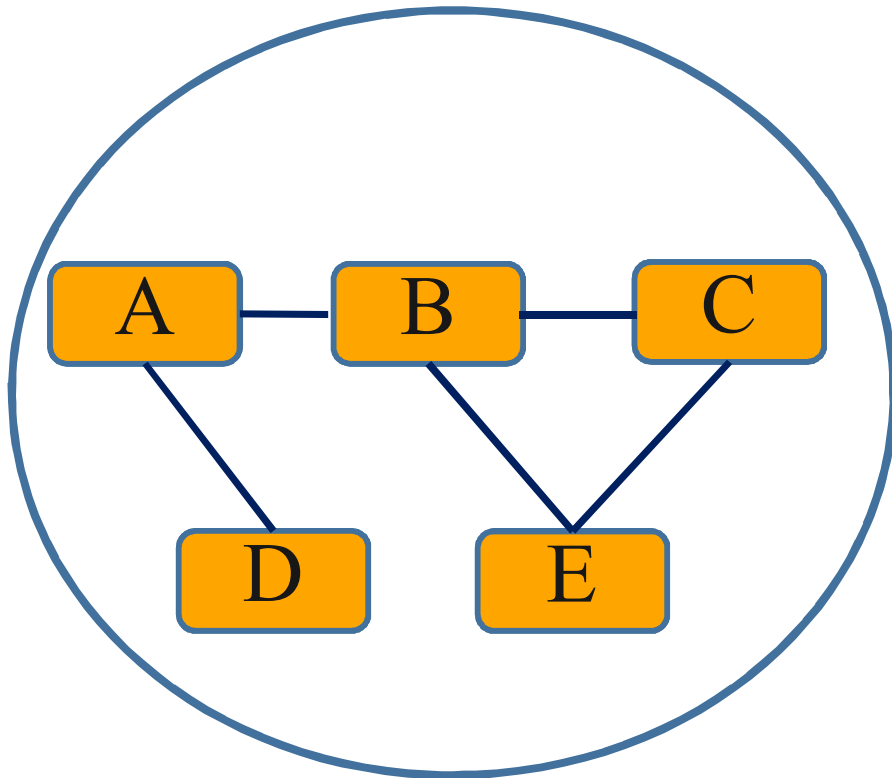
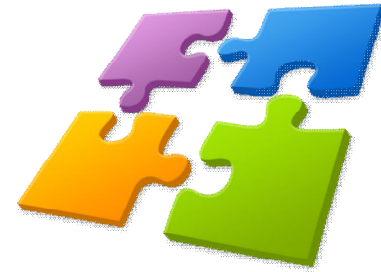
- معماری سرویس گرا، به عنوان پاسخی به این مساله روی سیستم های چابک فناوری اطلاعات با استفاده از مولفه های قابل استفاده مجدد، تاکید میکند.
- در این معماری مولفه ها یا برنامه های کامپیوتری برای حل مساله کسب و کار خاصی توسعه داده میشوند. در عوض تعدادی کارکرد عمومی ارائه می دهند.
- این مولفه ها میتوانند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

معماری سرویس گرا...

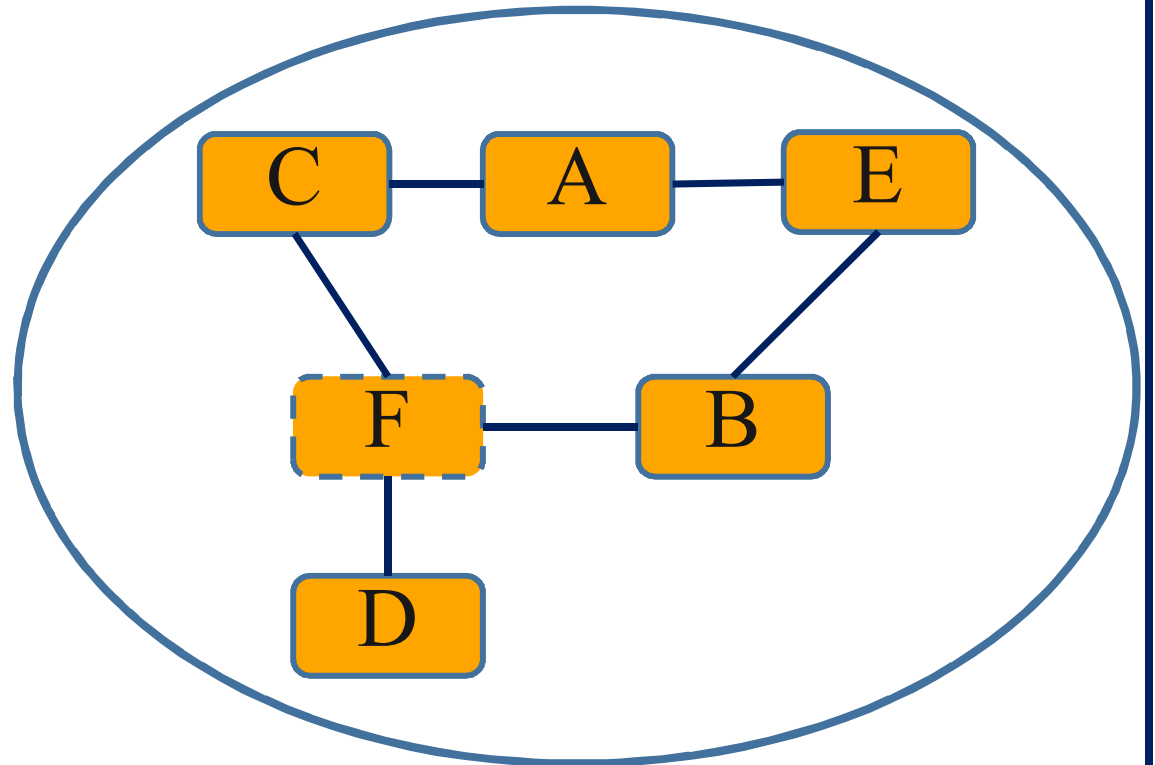


- اگر نیازمندی های کسب و کار تغییر کند، نیازی به توسعه یک برنامه کامپیوتری جدید نیست. سیستم میتواند مجدداً پیکربندی شود تا این نیاز جدید را تامین کند.
- معماری سرویس گرا، روی چابکی و استفاده مجدد از دارایی های نرم افزار با استفاده از مولفه های نرم افزاری که میتوانند به آسانی با پیکربندی های مختلف مجدداً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

معماری سرویس گرا...

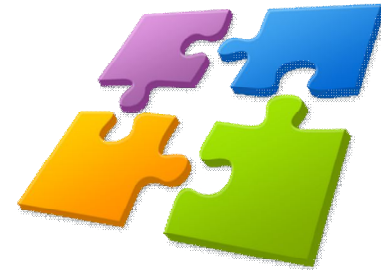


پیکربندی اولیه



پیکربندی ثانویه

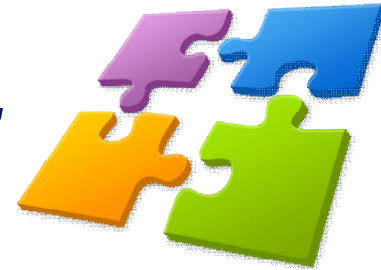
معماری سرویس گرا...



• 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

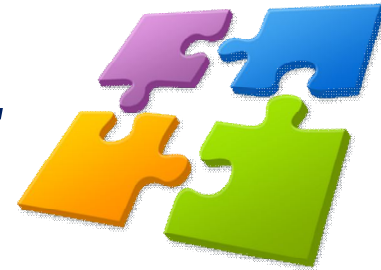
• رایانش توزیع شده

- چیزی بیش از یک برنامه کاربردی
- این برنامه ها معمولا 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...



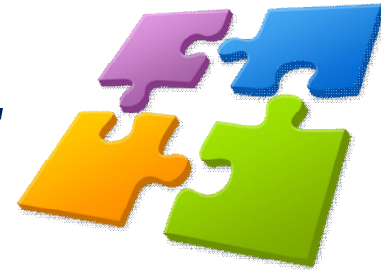
• نرم افزار سازمان

- 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 فروشندگان خارجی را نیز در نظر بگیرد. بنابراین این نرم افزار باید تعداد زیادی از انواع مختلف نیازمندی ها را تامین کند. بسیاری از این نیازمندی ها، 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 برنامه های کاربردی جدید یا اصلاح برنامه های کاربردی موجود برای به اشتراک گذاری داده ها و عملکردها

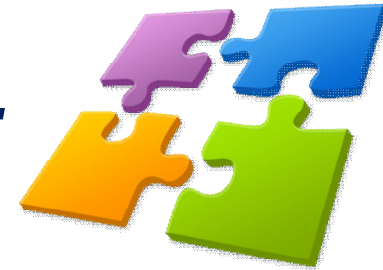
تعاریف و مفاهیم پایه...



• 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
مقارن	نامقارن	سبک ارتباطی
سیستم قوی	سیستم ضعیف	نوع سیستم
منطق متمرکز	منطق توزیع شده	الگوی تعامل
اتصال ایستا	اتصال پویا	اتصال سرویس



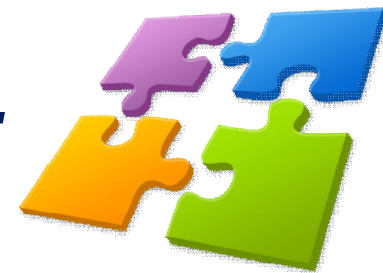
یکپارچه سازی کاربردهای سازمانی

فصل دوم: سوکت ها و تسهیم داده

بتول لکزائی

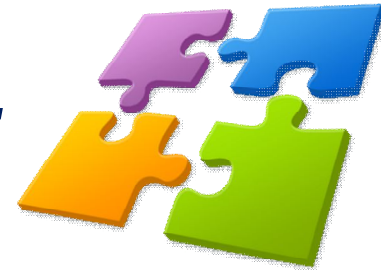
b_lakzaei@yahoo.com

تسهیم داده مبتنی بر فایل

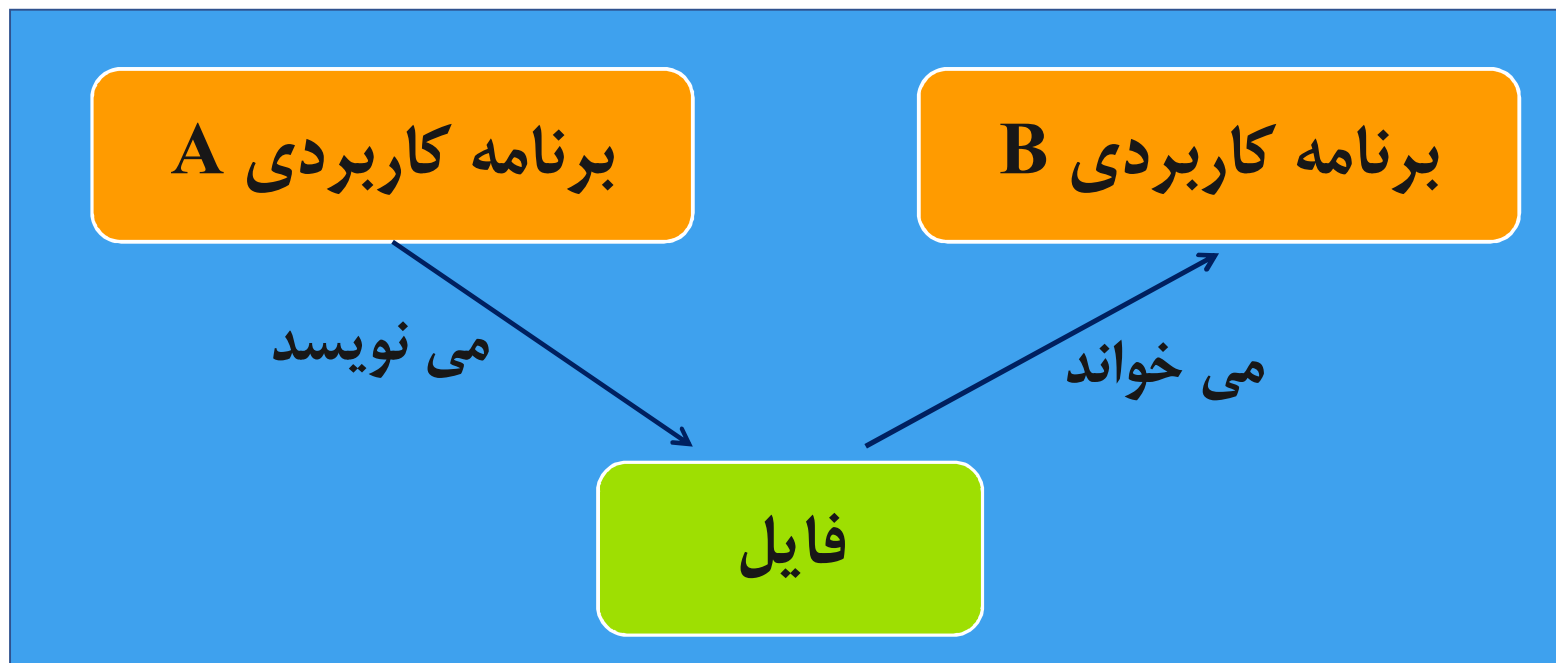


- اولین و رایج ترین روش تسهیم داده از طریق فایل ها است.
- در این روش، یک برنامه کاربردی داده ها را روی یک فایل می نویسد و برنامه کاربردی دیگر داده ها را از همان فایل می خواند.
- اگر هر دو برنامه روی یک ماشین اجرا شوند، میتوانند از دیسک ماشین برای خواندن و نوشتن استفاده کنند.
- اگر دو برنامه کاربردی روی دو ماشین مختلف اجرا می شوند، باید از روشهای انتقال داده بین دو ماشین استفاده کنند. متداولترین روش انتقال داده بین دو ماشین پروتکل انتقال فایل (FTP) است.

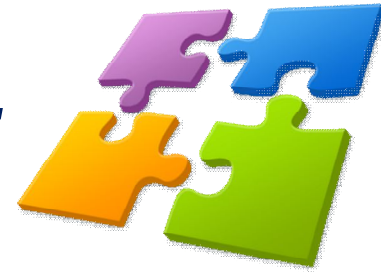
تسهیم داده مبتنی بر فایل ...



↓ سرور (ماشین)



تسهیم داده مبتنی بر فایل ...



سرور ۱ ↓

برنامه کاربردی A

می نویسد

فایل

سرور ۲ ↓

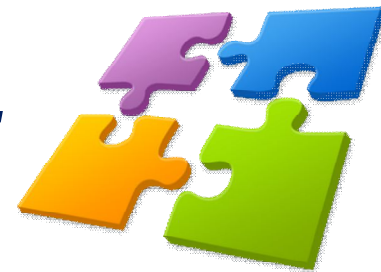
برنامه کاربردی B

می خواند

فایل

انتقال فایل (FTP)

تسهیم داده مبتنی بر فایل ...



- رایج ترین نوع تسهیم داده مبتنی بر فایل، استفاده از فایل های متنی است، زیرا ممکن است زبان ها و سیستم عامل های مختلف از تعداد و ترتیب متفاوتی از بایت ها برای نمایش مقادیر عددی استفاده کنند، اما تقریبا در همه زبانها و همه سیستم عامل ها، هر کاراکتر توسط یک بایت نمایش داده میشود.

• انواع متداول فایل های متنی:

- فایل های XML
- فایل های Flat
- فایل های رکورد با طول ثابت
- فایل های رکورد با طول متغیر

تسهیم داده مبتنی بر فایل ...

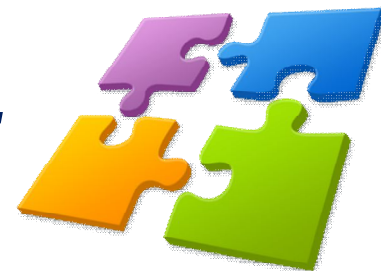


• فایل‌های Flat از نوع رکورد با طول ثابت

- در این فایل‌ها، طول هر فیلد ثابت و ترتیب ظاهر شدن فیلدهای مختلف نیز ثابت است.
- در هنگام نوشتن چنین فایل‌هایی، توسعه دهنده باید مطمئن شود که کلیه داده‌های فیلد در صورت لزوم با استفاده از درج فیلترهایی مانند فاصله یا صفر، طول مورد نیاز را دارند.
- به عنوان مثال فایل زیر، رنگ، طول و عرض یک کاغذ را نشان می‌دهد. فرض کنید طول فیلدها به ترتیب باید، شش، چهار و چهار باشد.

```
/* An example of fixed length flat file, which describes the color,  
length, and width of a sheet of paper. */  
Yellow01452456
```

تسهیم داده مبتنی بر فایل ...



• فایل Flat از نوع رکورد با طول متغیر

- طول رکوردها و در نتیجه طول فایلها متغیر است، ولی نوع رکوردها یکسان است.
- در این فایلها از حائل هایی مانند ویرگول یا نقطه ویرگول، برای جدا کردن فیلدهای مختلف استفاده میکنند.
- فیلدها می توانند هر طولی داشته باشند اما باید از ترتیب معینی پیروی کنند.

```
/* An example of variable length flat file which describes the color,  
length, and width of a sheet of paper */  
Yellow,145,2456
```

تسهیم داده مبتنی بر فایل ...

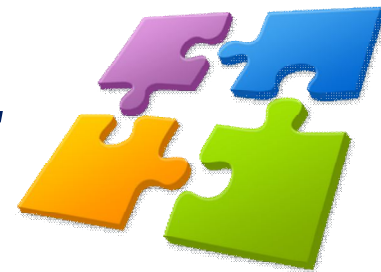


- انواع جدید فایل‌های متنی از XML استفاده میکنند.
- XML از تگ‌ها برای شرح فیلدهای مختلف یک رکورد استفاده میکند.
- مزیت استفاده از XML این است که تگ‌ها، خودتوصیف هستند و فایل توسط انسان قابل فهم است.

```
/* A portion of an XML file, which describes the color, length,  
and width of a sheet of paper. */
```

```
<paper>  
    <color>Yellow</color>  
    <length>145</length>  
    <width>2456</width>  
</paper>
```

تسهیم داده مبتنی بر فایل ...



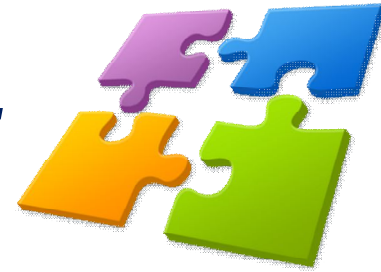
• مزایای استفاده از فایل متنی:

- نوشتن و خواندن متن یک فایل، معمولا وظایف ساده ای هستند.
- بیشتر زبان های برنامه نویسی، امکانات خوبی برای انجام این دو وظیفه، ارائه می دهند.

• معایب استفاده از فایل متنی:

- در یک فایل متنی حتی مقادیر عددی نیز به صورت رشته، خوانده و نوشته میشوند.
- برنامه نویس هنگام نوشتن باید مقادیر عددی را به مقادیر رشته ای ، و هنگام خواندن باید مقادیر رشته ای را به مقادیر عددی تبدیل کند.

تسهیم داده مبتنی بر فایل ...



```
Using System.IO;
```

```
...
```

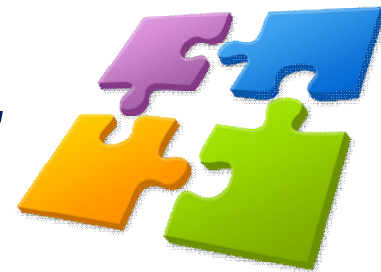
```
StreamWriter wr = new StreamWriter(FilePath);  
wr.Write("This is a test for writing to a file.");  
wr.Close();
```

```
Using System.IO;
```

```
...
```

```
StreamReader rd = new StreamReader(FilePath);  
string content = rd.ReadToEnd();  
rd.Close();
```


تسهیم داده مبتنی بر فایل ...



• معایب روش تسهیم داده مبتنی بر فایل

۱. در این روش داده به صورت بلادرنگ به اشتراک گذاشته نمیشود. و در نتیجه ممکن است مساله کهنگی داده (Staleness of Data) رخ دهد.

۲. هنگام نیاز تعداد زیادی برنامه به یکپارچه سازی، این روش مناسب نیست زیرا این نوع یکپارچه سازی، "نقطه به نقطه" است و تعداد یکپارچه سازی های نقطه به نقطه اساسا به ترتیب زیر افزایش می باید:

$$N(N-1)/2$$

(N: تعداد برنامه های کاربردی درگیر در یکپارچه سازی)

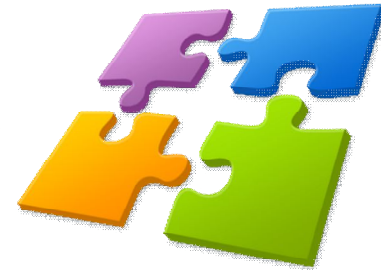


تسهیم داده مبتنی بر فایل ...

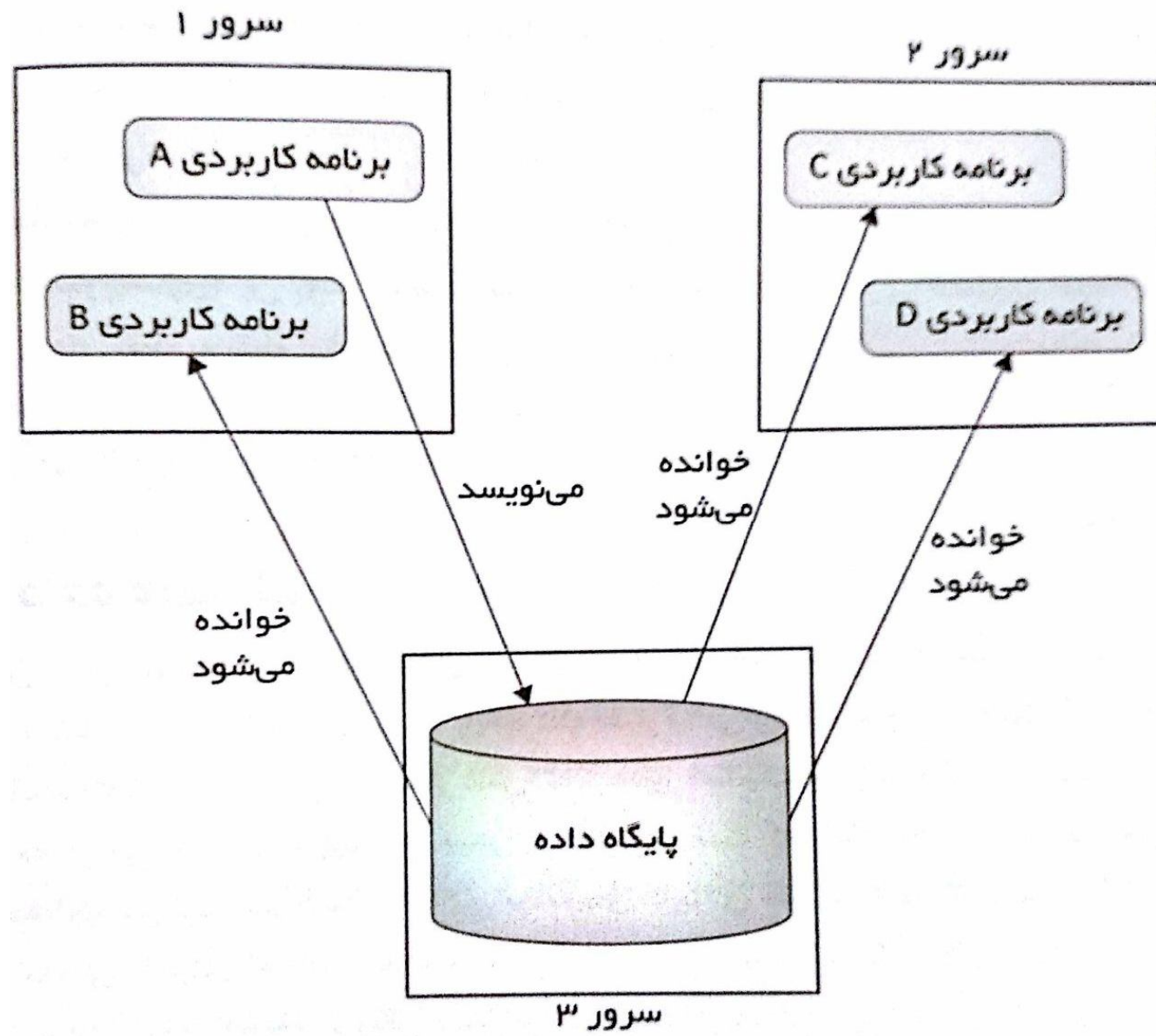
• معایب روش تسهیم داده مبتنی بر فایل ...

۳. اگر تعداد زیادی فایل درگیر باشند، این روش قابل اعتماد نیست و باعث افزایش حجم کاری توسعه دهندگان نرم افزار میشود زیرا:
 - توسعه دهندگان هم برای نوشتن روی فایل و هم برای خواندن فایل باید روی فرمت فایل توافق داشته باشند.
 - توسعه دهندگان هم برای نوشتن روی فایل و هم برای خواندن فایل باید روی قرارداد نامگذاری فایل توافق داشته باشند.
 - توسعه دهندگان باید روی دایرکتوری مورد نظر برای ایجاد یا انتقال فایل توافق داشته باشند.
 - توسعه دهندگان باید درباره برنامه کاربری مسئول حذف فایل، هنگامی که به آن نیازی نیست توافق داشته باشند.
 - توسعه دهندگان باید یک سازوکار قفل کردن را پیاده سازی کنند، تا هنگامی که یک برنامه کاربردی در حال نوشتن روی فایل است، سایر برنامه ها، به فایل دسترسی نداشته باشند.
 - اگر دو برنامه روی دو ماشین مختلف اجرا شوند، توسعه دهندگان باید تصمیم بگیرند که کدام برنامه کاربردی مسئول انتقال فایلها از یک سرور به سرور دیگر است.

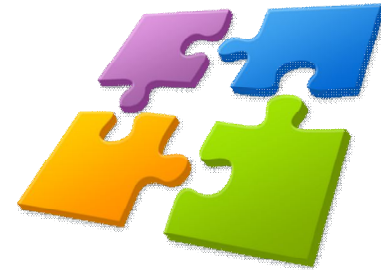
پایگاه داده مشترک



- در روش تسهیم سازی داده با استفاده از پایگاه داده مشترک، یک برنامه داده را روی یک پایگاه داده مشترک می نویسد، و سایر برنامه های کاربردی داده را از روی آن پایگاه می خوانند.
- تفاوت مهم این روش با روش مبتنی بر فایل این است که پایگاه داده، تقریباً همیشه روی ماشین مجزای خودش اجرا میشود. در نتیجه انتقال داده بین برنامه های کاربردی همیشه از طریق یک شبکه انجام میشود (حتی هنگامی که هر دو برنامه، روی یک ماشین باشند).
- به دلیل برقراری ارتباط از طریق شبکه، سرعت این روش نسبت به روش مبتنی بر فایل، کمتر است.
- این روش، بر خلاف روش مبتنی بر فایل، "نقطه به نقطه" نیست. یعنی وقتی داده روی پایگاه داده نوشته میشود، هر تعداد از برنامه های کاربردی می توانند آن را به اشتراک بگذارند. بنابراین داده همواره با برنامه های کاربردی مختلف سازگار است.



پایگاه داده مشترک ...



• معایب روش تسهیم داده مبتنی بر پایگاه داده مشترک

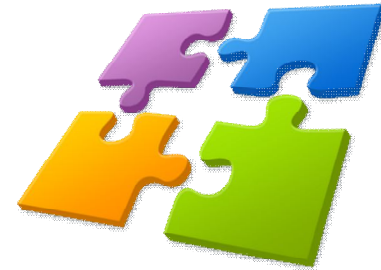
۱. داده به طور بلادرنگ به اشتراک گذاشته نمیشود.

۲. تعیین یک طراحی یکپارچه برای پایگاه داده، که بتواند نیازهای چند برنامه کاربردی مختلف را تامین کند، دشوار است.

- اگر هر یک از برنامه های کاربردی، خود بخشی از یک سیستم بزرگتر باشند که نیاز به یکپارچه سازی دارد، این مساله تشدید میشود.

- تغییر در نسخه های مختلف یک برنامه، ممکن است نیازمند تغییر در پایگاه داده آن باشد و این مساله میتواند سایر برنامه های کاربردی که از این پایگاه داده استفاده می کنند را تحت تاثیرات نامطلوب قرار دهد.

پایگاه داده مشترک ...



معایب روش تسهیم داده مبتنی بر پایگاه داده مشترک ...

۳. اگر تعداد زیادی برنامه نیاز به یکپارچه سازی داشته باشند، این روش مناسب نیست زیرا تعداد زیادی برنامه دائما در حال نوشتن روی پایگاه داده یا خواندن آن هستند و در نتیجه، ممکن است این مساله باعث بروز بن بست شود.

۴. اگر برنامه های کاربردی در چندین مکان مختلف توزیع شده باشند، این روش مناسب نیست زیرا در عمل، دسترسی به یک پایگاه داده مشترک از طریق شبکه گسترده (WAN) بسیار کند است.

سوکت ها

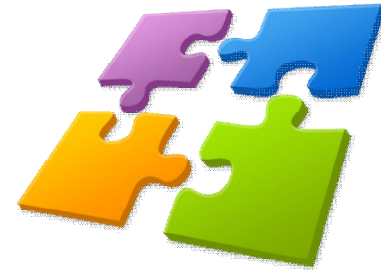
- راه حل مساله کهنگی داده:
- اتصال بلادرنگ

- ابتدایی ترین روش برای برقراری اتصال بین دو برنامه کاربردی:
- استفاده از سوکت ها

• سوکت:

- یک نقطه اتصال ارتباطات (Communications Connection Point) یا به عبارت دیگر یک نقطه پایانی (EndPoint) است که می تواند در شبکه نامگذاری و آدرس دهی شود.

سوکت ها ...



- فرآیندهایی که از سوکت استفاده میکنند میتوانند روی یک سیستم یا سیستم های مختلفی در شبکه های مختلف مستقر شوند.
- سوکت ها هم برای برنامه های کاربردی شبکه و هم برای برنامه های کاربردی مستقل مناسبند.
- رابطهای کاربردی برنامه نویسی سوکت، استاندارد شبکه برای پروتکل TCP/IP هستند. و توسط بسیاری از سیستم عاملها، پشتیبانی میشوند.

سوکت ها ...

- سوکت ها، چگونه مساله کهنگی داده را حل می کنند؟
- در این روش، یک برنامه کاربردی در یک پورت معین روی یک ماشین معین گوش به زنگ داده دریافتی است. برنامه کاربردی دیگر میتواند با استفاده از آدرس IP و آدرس پورت برنامه کاربردی اول، روی همان سوکت داده ها را بنویسد. برنامه کاربردی مستمع میتواند به محض اینکه برنامه دوم داده را نوشت، آن را بخواند. بنابراین داده به طور بلادرنگ به اشتراک گذاشته میشود.
- سربار برقراری ارتباط از طریق سوکت ها بسیار پایین است. در نتیجه برنامه نویسی مستقیم سوکت، روشی بسیار کارا در ارتباطات است و بسیاری از روشهای مدرن ارتباطی، در عملکرد داخلی خود، متکی به برنامه نویسی سوکت می باشند.

سوکت ها ...



جریان کد سمت سرور برای تسهیم
داده با استفاده از سوکت ها

ایجاد یک سوکت

انقیاد به یک پورت

گوش دادن به درخواست اتصال

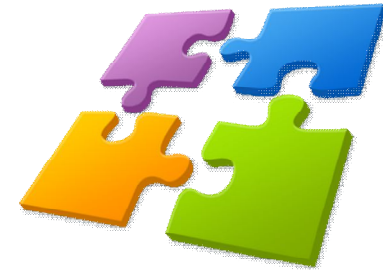
پذیرش یک درخواست اتصال

خواندن و نوشتن در سوکت

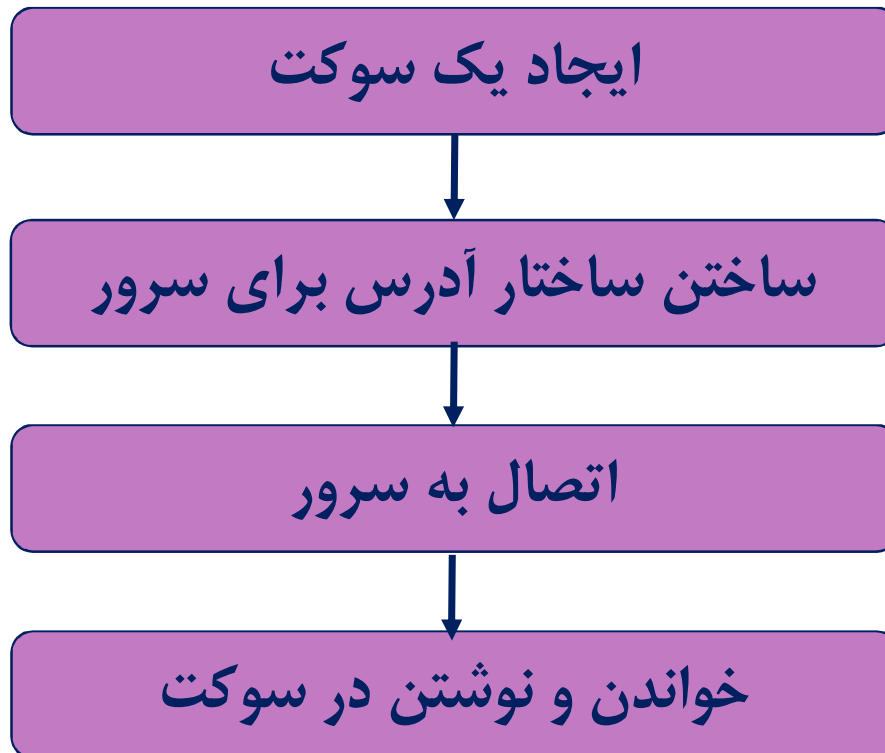
بستن سوکت

```
1 using System;
2
3 using System.Net;
4
5 using System.Net.Sockets;
6
7 using System.Text;
8
9
10
11 static Socket sktListener;
12
13 static void Main(string[] args)
14 {
15
16     sktListener = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
17
18     IPEndPoint ipLocal = new IPEndPoint(IPAddress.Any, 1800);
19
20     sktListener.Bind(ipLocal);
21
22     sktListener.Listen(100);
23
24     sktListener = sktListener.Accept();
25
26     byte[] buffer = new byte[500];
27
28     sktListener.Receive(buffer);
29
30     sktListener.Close();
31
32 }
33
34
35
```

سوکت ها ...



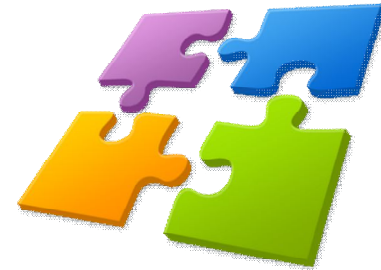
جریان کد سمت کلاینت برای
تسهیم داده با استفاده از سوکت ها



```
using System;
using System.Net;
using System.Net.Sockets;
using System.Text;

static void Main(string[] args)
{
    Socket sktClient = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
    sktClient.Connect("127.0.0.1", 1800);
    string str = "Hello Server....";
    byte[] buffer = Encoding.ASCII.GetBytes(str);
    sktClient.Send(buffer);
    sktClient.Close();
}
```

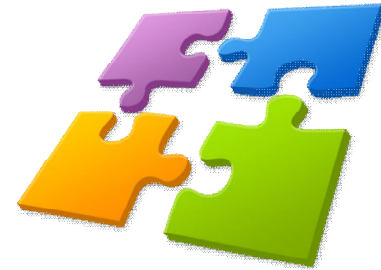
سوکت ها ...



• نقص های روش مبتنی بر سوکت:

۱. مساله اصلی برنامه نویسی سوکت این است که **فقط داده** به صورت مستقیم قابل تسهیم است نه عملکرد.
۲. به دلیل سطح پایین بودن API (رابط برنامه نویسی کاربردی)، این روش برای انواع پیچیده مناسب نیست.
۳. کد قابلیت اتصال در برنامه های کاربردی قرار دارد و به راحتی قابل استفاده مجدد نیست.
۴. اگر داده ها مقادیر عددی باشند، این روش مستقل از پلت فرم نیست.
۵. اتصال سوکت به صورت "نقطه به نقطه" است و پیوند قوی بین دو برنامه وجود دارد.

سایر روشهای تسهیم داده



- اگر برنامه های کاربردی روی یک ماشین اجرا شوند و از یک سیستم عامل استفاده کنند، سه روش دیگر نیز برای تسهیم داده بین آنها وجود دارد:
 - حافظه مشترک، Pipe و FIFO

• روش حافظه مشترک

- در این روش یکی از برنامه ها، ناحیه ای در RAM ایجاد میکند که سایر برنامه های کاربردی نیز میتوانند به آن دسترسی داشته باشند.
- این روش، سریع ترین روش برای تسهیم داده به صورت بلادرنگ است.
- حافظه مشترک تا زمانی که سیستم مجدداً راه اندازی شود، باقی می ماند.

- روش Pipe و FIFO نیز مشابه روش حافظه مشترک هستند با این تفاوت که Pipe و FIFO تا زمانی باقی می ماند که آخرین برنامه کاربردی که شی را باز نگه میدارد، آن را ببندد.



یکپارچه سازی کاربردهای سازمانی

فصل سوم: فراخوانی رویه راه دور (RPC)

بتول لکزائی

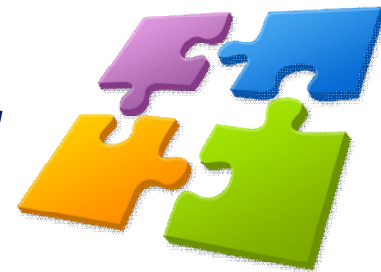
b_lakzaei@yahoo.com

اشتراک عملکرد



- روشهای مبتنی بر فایل و مبتنی بر سوکت، تنها محدود به تسهیم داده هستند، و به برنامه های کاربردی اجازه تسهیم عملکرد را نمی دهند.
- سوکت مفهوم اتصال برنامه های کاربردی را معرفی کرد که برای تسهیم عملکرد، ضروری هستند.
- به عبارت دیگر، هنگامی که برنامه های کاربردی عملکرد را به اشتراک می گذارند، بدون توجه به روش یکپارچه سازی، سوکت ها همیشه تقریباً در پس زمینه هستند.

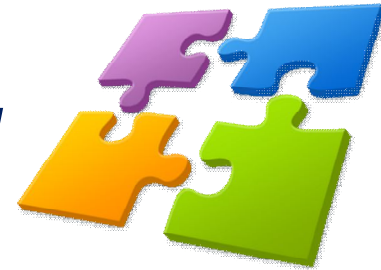
اشتراک عملکرد ...



• اشتراک عملکرد:

- توابع تعریف شده در یک برنامه کاربردی، توسط سایر برنامه های کاربردی سازمان فراخوانی شوند.
- یکپارچه سازی کاربردهای سازمانی، فرآیند دشواری است چرا که انواع بسیار متفاوتی از برنامه های کاربردی را در بر میگیرد که به زبان های متفاوتی نوشته شده اند و روی انواع متفاوتی از پلت فرم ها که ممکن است از لحاظ جغرافیایی، توزیع شده باشند، اجرا می شوند.

انواع روشهای فراخوانی توابع



- توابع همگام
- تابع فراخواننده، مسدود میشود و تا زمانی که کنترل از تابع فراخوانی شده، بازنگردد، قادر به انجام هیچ کاری نیست.
- توابع ناهمگام
- کد تابع فراخواننده می تواند برای انجام کارهای دیگر ادامه پیدا کند، بدون اینکه منتظر برگشت کنترل از تابع فراخوانی شده باشد.
- فراخوانی رویه راه دور، فقط شامل فراخوانی های همگام است.

انواع توابع همگام

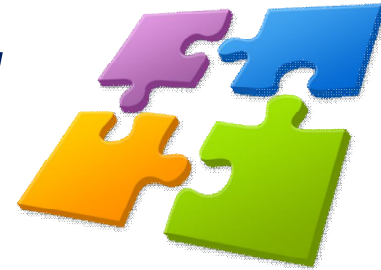
RPC

RPC کنترل شده

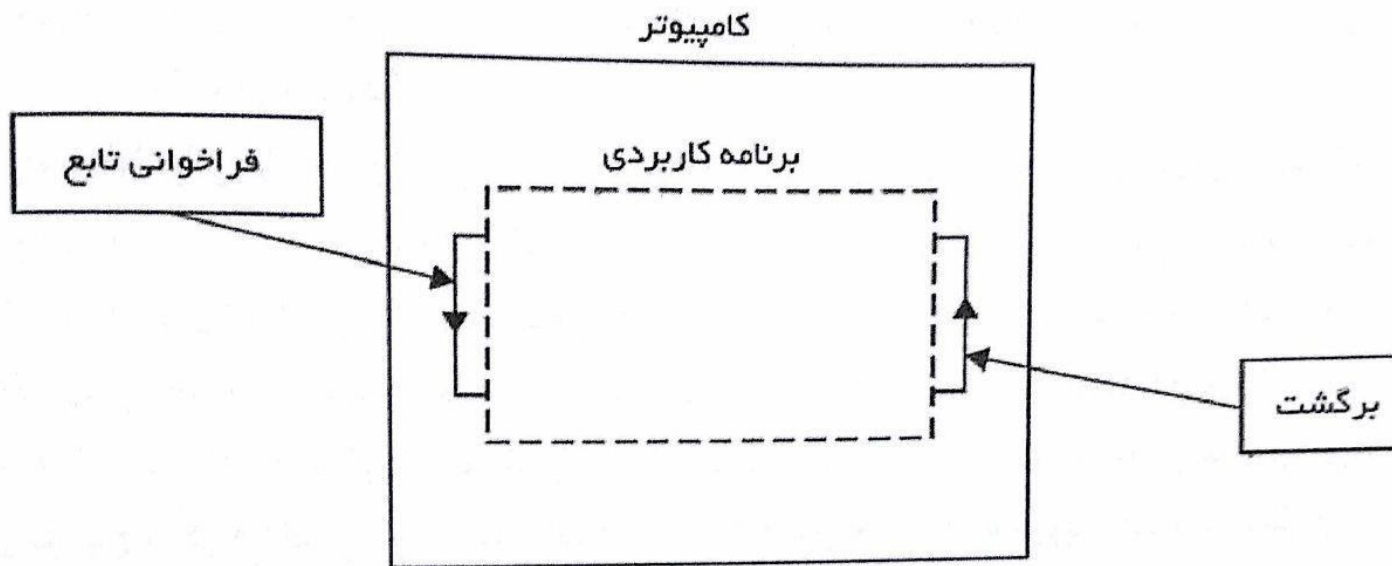
فراخوانی توابع محلی

انواع توابع همگام

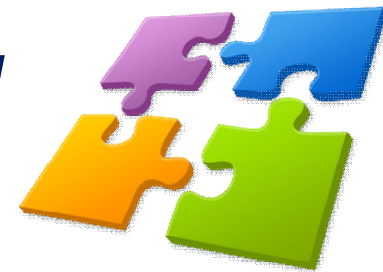
فراخوانی تابع محلی



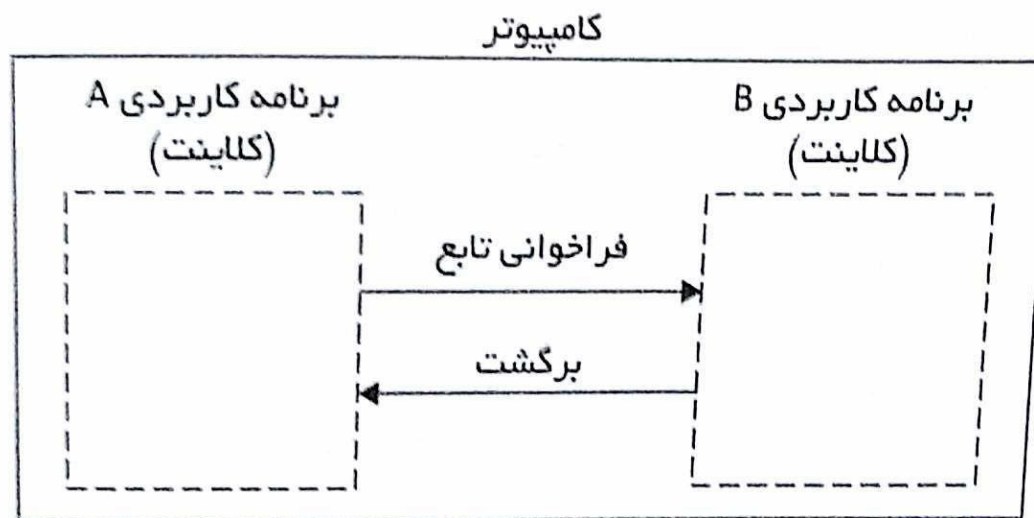
- کد تابع فراخواننده و کد تابع فراخوانی شده، هر دو بخشی از یک برنامه هستند.
- هر دو تابع روی یک ماشین، مقیم هستند.
- با اجرای برخی از دستورالعمل های زبان ماشین، کنترل از تابع فراخوان به یک تابع جدید منتقل میشود. مقادیر ثابتها در حافظه ذخیره می شود و متغیرهای محلی تابع در بخشی از فضای پشته قرار می گیرند.



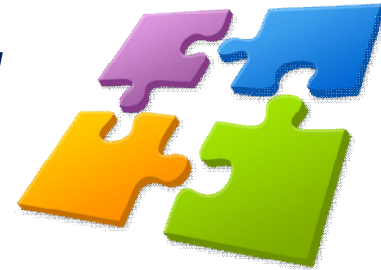
انواع توابع همگام... RPC کنترل شده



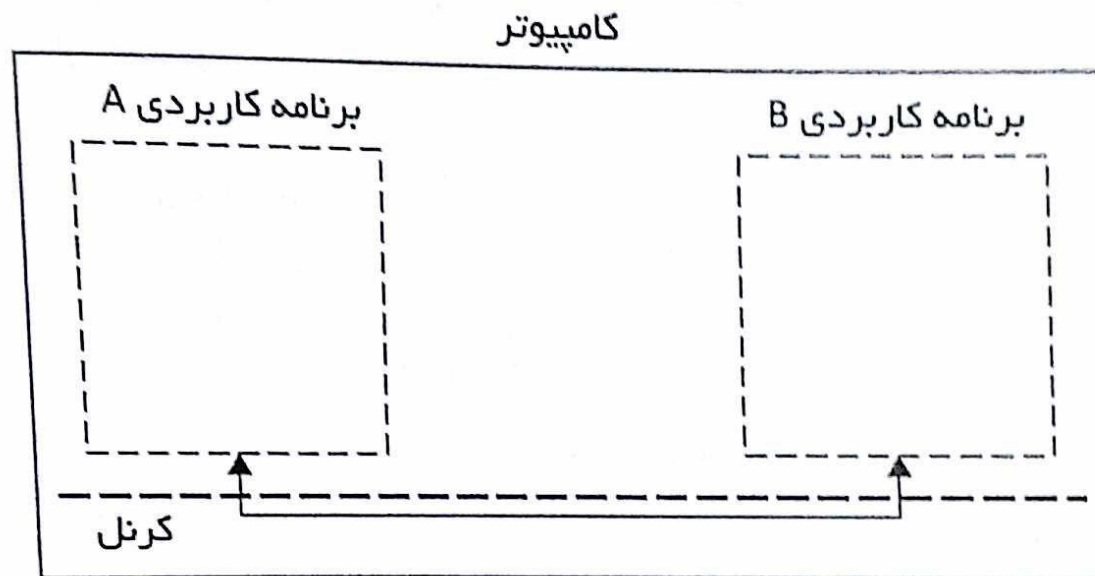
- کد تابع فراخواننده و کد تابع فراخوانی شونده، در دو برنامه کاربردی مختلف که روی یک ماشین اجرا می‌شوند مقیم هستند.
- به عبارت دیگر، در این روش یک برنامه کاربردی یعنی سرور، تابع را در دسترس سایر برنامه‌های کاربردی یعنی کلاینت‌ها، قرار می‌دهد. سرور و کلاینت‌ها روی یک کامپیوتر اجرا می‌شوند.



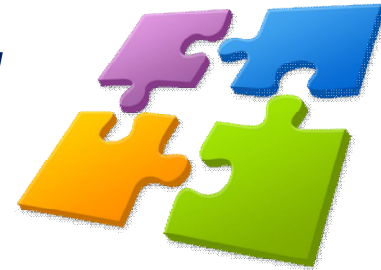
انواع توابع همگام... RPC کنترل شده...



- این روش مشابه روش RPC است، با این تفاوت که در اینجا نیازی به ارتباطات شبکه ای نیست. و از هسته سیستم عامل برای برقراری ارتباط استفاده میشود.
- این روش نوعی ارتباط بین فرآیندی یا IPC در نظر گرفته میشود.



انواع توابع همگام... RPC کنترل شده...



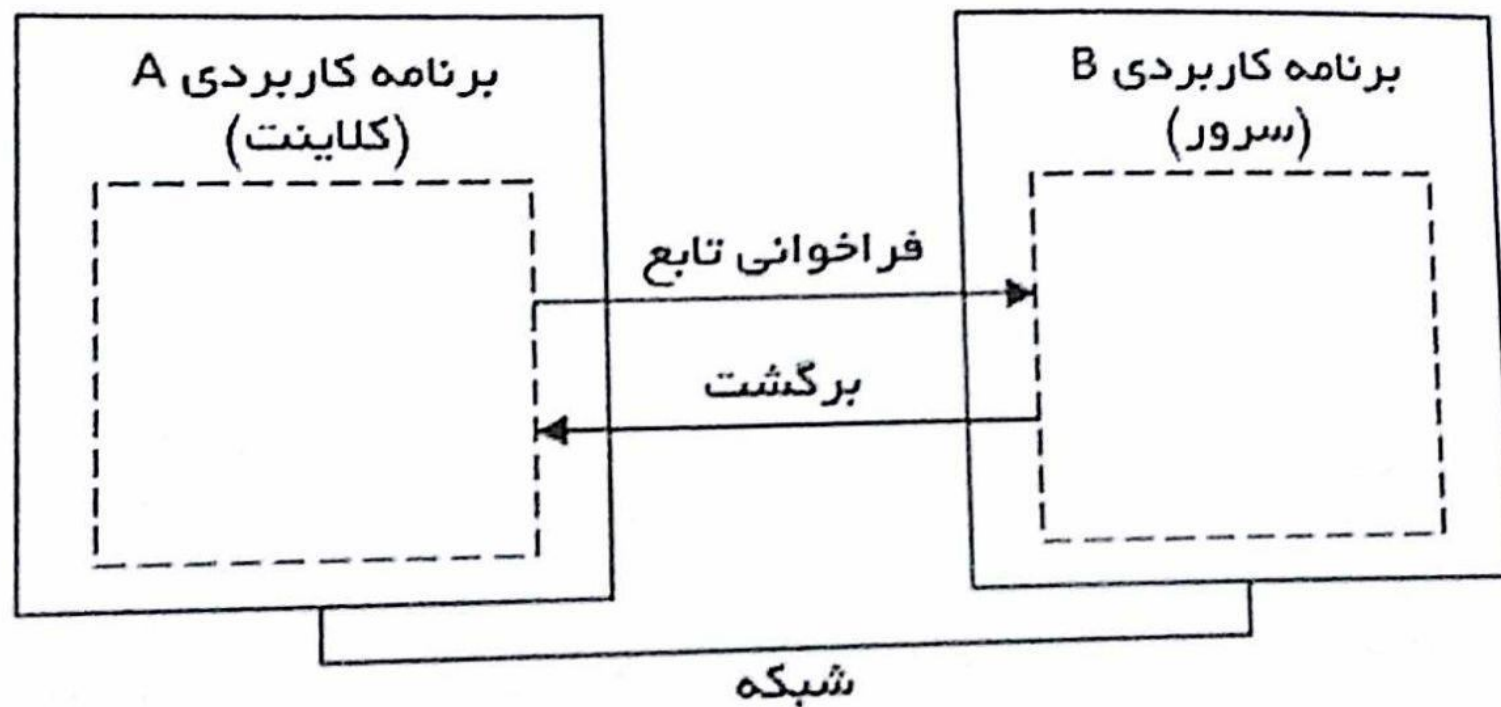
- فراخوانی رویه راه دور کنترل شده، شهودی ترین و کارا ترین روش برای تسهیم عملکرد بین دو برنامه کاربردی است.
- در این روش ارتباط بین دو برنامه کاربردی از طریق دستورات فراخوان سیستم در هسته سیستم عامل انجام می شود.
- یک پیاده سازی ویژه از این نوع تبادل عملکرد، Door نامیده می شود.
- **دو محدودیت اصلی Door**
 - مستقل از پلتفرم نیست و تنها می تواند در سیستم عامل های Solaris استفاده شود.
 - دو برنامه کاربردی که عملکرد را به اشتراک می گذارند باید روی یک میزبان اجرا شوند.

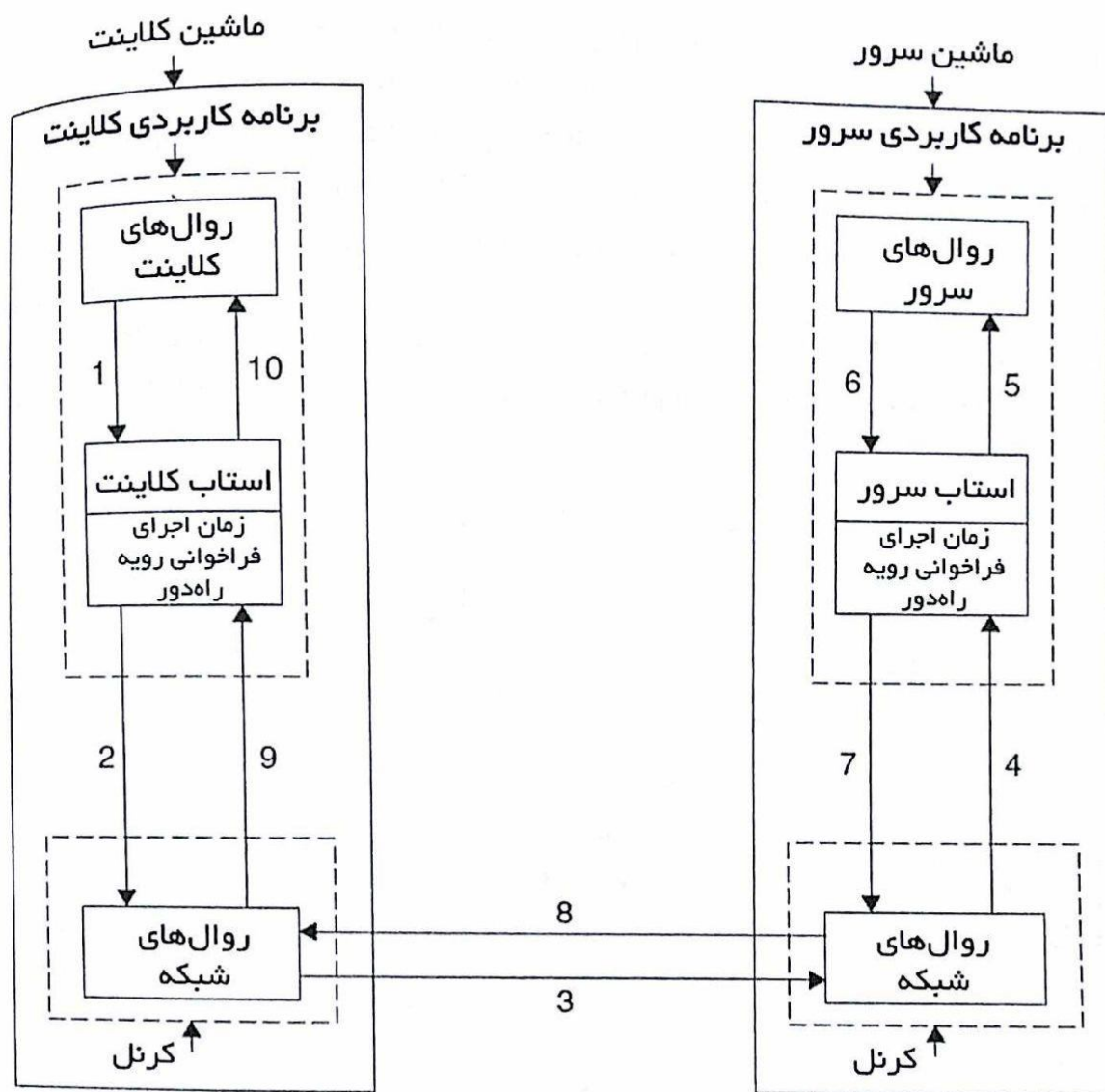
انواع توابع همگام... RPC



- فراخوانی رویه راه دور گامی مهم در توسعه الگوی یکپارچه سازی است، زیرا برای اولین بار، کلیه مراحل ضروری برای تسهیم عملکرد بین برنامه های کاربردی، در این روش مطرح شد.
- RPC می تواند بین هر دو برنامه کاربردی دلخواه برای تسهیم عملکرد استفاده شود.
- این برنامه های کاربردی ممکن است روی ماشین های مجزا که توسط شبکه به هم متصل می شوند، اجرا شوند.
- RPC مبتنی بر معماری کلاینت-سرور است.
- **کلاینت**: برنامه کاربردی که تابع را فراخوانی می کند.
- **سرور**: برنامه کاربردی که تابع فراخوانی شده در آن مقیم است.

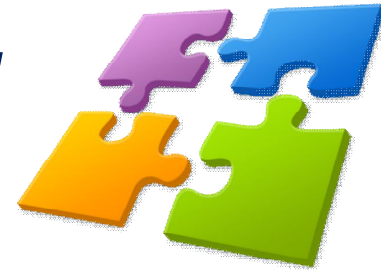
انواع توابع همگام... ... RPC





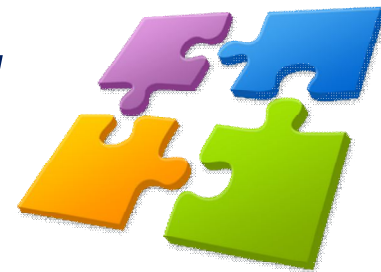
شکل ۸-۴ فرآیند کامل فراخوانی رویه راه دور

انواع توابع همگام... ... RPC



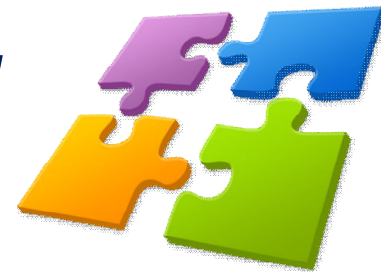
- ابتدا سرور راه اندازی می شود و یک پورت موقت را رجیستر می کند، و به این پورت گوش می دهد.
- سپس کلاینت راه اندازی می شود و با اجرای دستوراتی معین، پورت موقت سرور را پیدا می کند.
- سپس کلاینت با سرور یک اتصال TCP را از طریق این پورت برقرار می کند.
- استاب کلاینت، آرگومان های ارسالی برای فراخوانی تابع قرار گرفته روی سرور، را بسته بندی میکند، آنها را در قالب فرمتی استاندارد تنظیم می کند، و سپس در قالب یک یا چند پیام شبکه قرار می دهد.
- این بسته بندی آرگومان ها، تجمیع نامیده می شود.
- با این تجمیع تفاوت های نحوه ذخیره سازی اعداد، در در پلتفرم های مختلف به طور خودکار مدیریت می شوند بنابراین فراخوانی رویه راه دور مستقل از پلت فرم است (اما مستقل از زبان نیست).

انواع توابع همگام... ... RPC



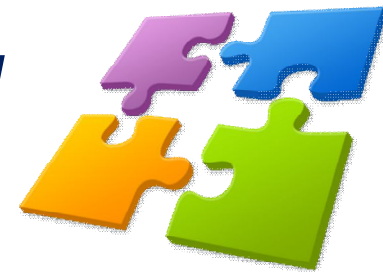
- استاب کلاینت از توابع موجود در کتابخانه زمان اجرای RPC، برای فراخوانی سیستم ها در هسته سیستم عامل محلی استفاده می کند تا پیام بسته بندی شده تحت شبکه را با استفاده از پروتکلی مانند TCP به سرور ارسال کند.
- به عبارت RPC، در زمان اجرا همه فراخوانی های سیستمی را که برای اتصال ضروری هستند را کپسوله می کند بنابراین نیازی نیست برنامه نویس برنامه نویسی سیستم ها را بلد باشد.
- در سمت سرور همین که پیام شبکه توسط روال های شبکه در هسته سیستم عامل دریافت میشود از طریق کتابخانه زمان اجرای RPC برای استاب سرور ارسال می شود.
- استاب سرور پارامترهای ورودی را تفکیک می کند و تابع محلی درخواست شده را فراخوانی می کند.

انواع توابع همگام... ... RPC



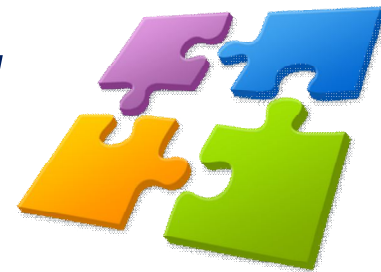
- پس از اجرای تابع محلی فراخوانی شده استاب سرور، مقدار بازگشتی را در یک یا چند پیام تحت شبکه تنظیم می کند و بسته بازگشت را از طریق کتابخانه RPC در زمان اجرا به هسته سرور ارسال می کند.
- هسته سرور پیام ها را با استفاده از یک پروتکل مانند TCP به کلاینت ارسال می کند.
- استاب کلاینت، پیام های شبکه را از هسته با استفاده از کتابخانه زمان اجرای RPC دریافت می کند، سپس مقادیر بازگشتی را به فرمت قابل قبول برای کلاینت تبدیل می کند.

انواع توابع همگام... ... RPC



- **معایب روش RPC**
- قابلیت استفاده مجدد کد در این روش ضعیف است، زیرا کدهای تجمیع، تفکیک و ارتباطات شبکه در برنامه های کلاینت و سرور نوشته شده اند.
- این روش مستقل از زبان نیست و کلاینت و سرور باید از یک زبان برنامه نویسی یکسان استفاده کنند.
- نوع پیوند بین برنامه های کاربردی پیوند قوی است، زیرا فراخوانی ها همگام هستند.
- در این روش نقش کلاینت و سرور ثابت است و رابطه به صورت همتا به همتا (Peer to Peer) نیست، یعنی کلاینت به عملکرد تعبیه شده در سرور دسترسی دارد ولی عکس آن امکان پذیر نیست.

انواع توابع همگام... ... RPC



• معایب روش RPC ...

- از روش ارتباط نقطه به نقطه استفاده می کند، بنابراین هنگامی که تعداد زیادی برنامه نیاز به یکپارچه سازی دارند مناسب نیست.
- اگر تعداد زیادی از فراخوانی های راه دور درگیر باشند این روش مناسب نیست زیرا به دلیل ماهیت ذاتی فراخوانی همگام کلاینت نمی تواند قبل از اینکه سرور کارش را تمام کند کار خود را ادامه دهد.
- می توان این مشکل را با استفاده از برنامه نویسی چند رشته ای حل کرد اما این کار پیچیدگی برنامه را افزایش داده و ریسک هایی به همراه دارد.



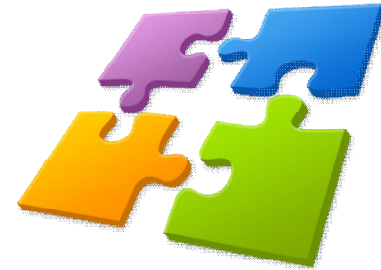
یکپارچه سازی کاربردهای سازمانی

فصل چهارم: اشیای توزیع شده

بتول لکزائی

b_lakzaei@yahoo.com

مدل اشیای توزیع شده



- روش RPC برای اولین بار امکان به اشتراک گذاری **عملکرد** بین برنامه های کاربردی مختلف که ممکن است روی ماشینهای متفاوتی اجرا شوند را فراهم کرد.
- با این وجود، این روش معایب بسیاری داشت:
 - پایین بودن قابلیت استفاده مجدد کد
 - وابستگی به زبان
 - ثابت بودن نقش کلاینت و سرور
 - ارتباط نقطه به نقطه
 - فراخوانی همگام
 - عدم مقیاس پذیری



مدل اشیای توزیع شده ...

- راه حل پیشنهادی برای حل معایب روش RPC باید دارای ویژگی های زیر باشد:
 ۱. جداسازی کد تجمع و کد ارتباطات در یک مولفه مستقل
- مزیت:

- افزایش قابلیت استفاده مجدد
- حل مشکل رابطه "نقطه به نقطه" بین کلاینت و سرور (زیرا برنامه های کاربردی متعددی می توانند با استفاده از این مولفه مستقل به یکدیگر متصل شوند).
- "همتا به همتا" بودن رابطه بین کلاینت و سرور

۲. عدم وابستگی به زبان و عدم وابستگی به پلت فرم

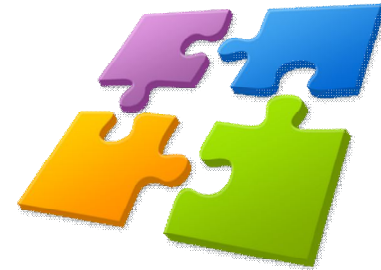
راه حلی که شامل این ویژگی ها می باشد: استفاده از اشیای توزیع شده



مدل اشیای توزیع شده ...

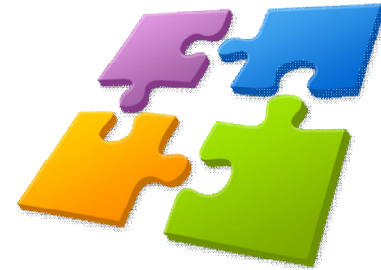
- اشیای توزیع شده مفاهیم مرتبط با کلاس ها و برنامه نویسی شی گرا را توسعه می دهند.
- سه رکن اصلی برنامه نویسی شی گرا:
 - **کلاس:**
 - ساختار تعریف شده توسط کاربر که داده و رفتار مربوط به یک نهاد معین را کپسوله می کند.
 - مزایای کپسوله سازی: برنامه نویسی قابل اطمینان تر و قوی تر
 - **ارث بری:**
 - منجر به استفاده مجدد از کد در سطح برنامه نویسی می شود.
 - **چند ریختی:**
 - یعنی توابعی که کار مشابهی انجام میدهند میتوانند نام یکسانی داشته باشند.
 - البته ویژگی سوم در این بخش اهمیت زیادی ندارد

مدل اشیای توزیع شده ...



- شی: نمونه خاصی از یک کلاس معین در زمان اجرا.
- قبل از اشیای توزیع شده فقط اشیایی که متعلق به یک برنامه کاربردی بودند می توانستند با یکدیگر در زمان اجرا تعامل داشته باشند.
- با اشیای توزیع شده اشیای متعلق به برنامه‌های مختلف نیز می‌توانند با یکدیگر تعامل داشته باشند و داده و عملکرد را در زمان اجرا مبادله کنند.

مدلهای مبتنی بر اشیای توزیع شده



وابسته به پلت فرم ویندوز

مدل شی مولفه توزیع شده / DCOM
Distributed Component Object Model

مستقل از پلت فرم، وابسته به زبان Java

احضار متد از راه دور / RMI
Remote Method Invocation

مستقل از پلت فرم و مستقل از زبان

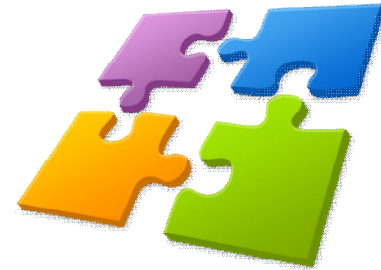
معماری عمومی رابط درخواست شی / CORBA
Common Object Request Broker Architecture

CORBA مدل

- مولفه های اصلی مدل CORBA
 - رابط درخواست شی / ORB
 - زبان تعریف رابط / IDL
 - مدل شی CORBA
 - نقشه های کلاینت و سرور در CORBA
 - سرویس های CORBA

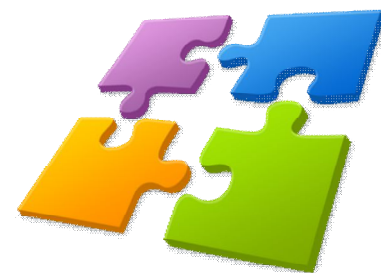
مدل CORBA

رابط درخواست شی / ORB



- مهم ترین مولفه در CORBA

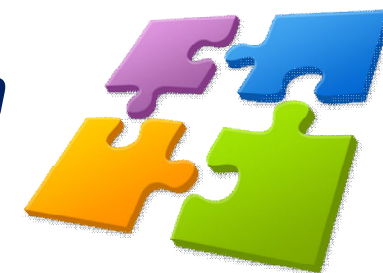
- کد تجمیع و کد فراخوانی های شبکه را در یک مولفه نرم افزاری مجزا، کپسوله می کند.
- کلیه نیازهای تجمیع و تفکیک آرگومان ها و مقادیر بازگشتی اشیای توزیع شده را تامین می کند:
 ۱. باعث میشود چندین برنامه کاربردی بتوانند از یک کد، به طور مجدد استفاده کنند.
 ۲. با کنار گذاشتن روش یکپارچه سازی نقطه به نقطه، باعث از بین رفتن پیوند قوی می شود.
 ۳. چندین برنامه کاربردی روی یک ماشین میتوانند از طریق این رابط، با یکدیگر و با برنامه های کاربردی مقیم در ماشین های دیگر، ارتباط برقرار کنند.



مدل CORBA

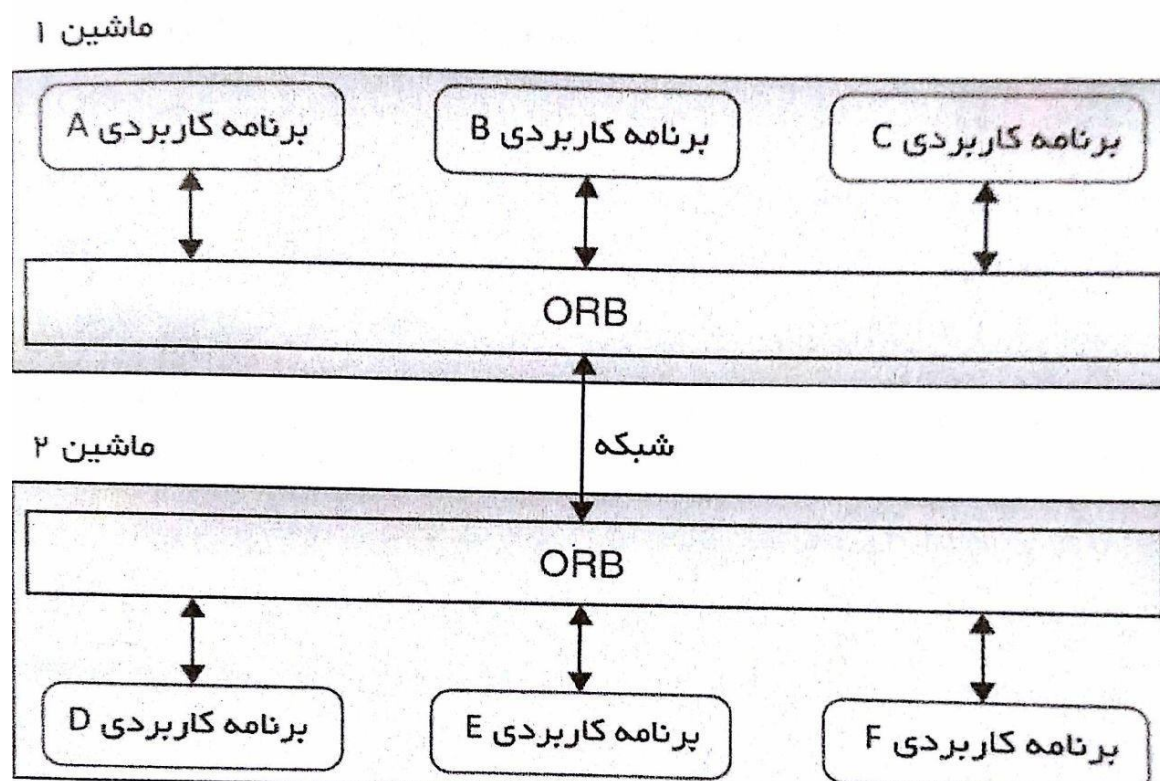
رابط درخواست شی / ORB ...

- کلیه نیازهای تجمیع و تفکیک آرگومان ها و مقادیر بازگشتی اشیای توزیع شده را تامین می کند: ...
- ۴. ارتباط بین مولفه ها مستقل از پلت فرم است زیرا پارامترها به محض انتقال به فرمت مستقل از پلت فرم تبدیل می شوند و به محض پذیرش به فرمت خاص پلت فرم مقصد، تبدیل می شوند.
- ۵. تفاوت های سخت افزاری، مشکل ساز نخواهد بود زیرا رابط درخواست شی این تبدیلات را به طور خودکار انجام میدهد.
- ۶. فرایند تجمیع و تفکیک کاملاً توسط رابط درخواست شی انجام میشود و برنامه نویس نگران جزئیات نحوه این تجمیع و تفکیک نیست.



مدل CORBA

رابط درخواست شی / ORB ...





مدل CORBA

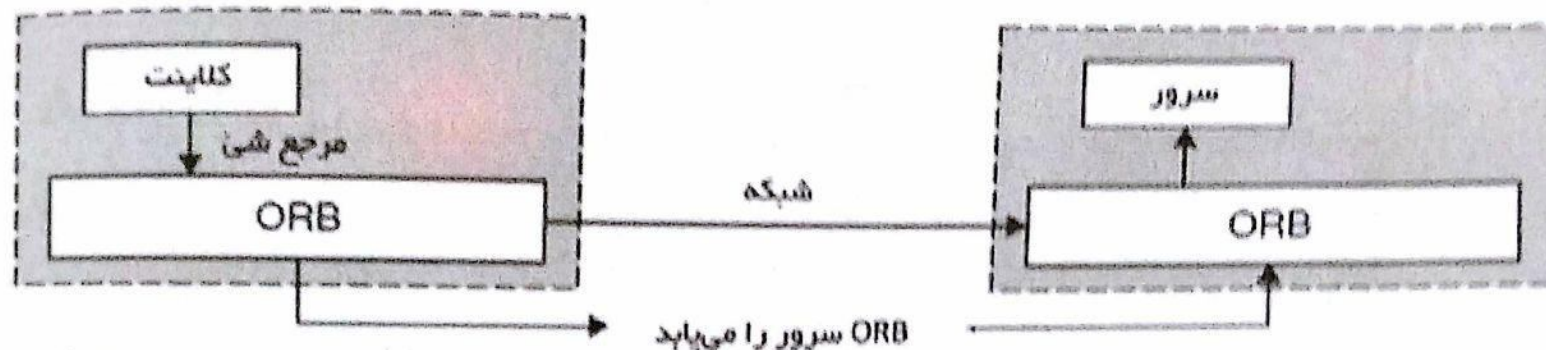
رابط درخواست شی / ORB ...

مراحل اصلی برقراری ارتباط بین برنامه های کاربردی از طریق رابط درخواست شی:

۱. هنگامی که یک شی یا مولفه می خواهد از سرویسهای شی یا مولفه دیگری استفاده کند، ابتدا مرجعی برای شی ارائه دهنده سرویس به دست می آورد.
۲. سپس رابط درخواست شی، پیاده سازی شی متناظر را از طرف کلاینت مکان یابی می کند.
۳. هنگامی که سرور مکان یابی شد، رابط درخواست شی اطمینان میدهد که سرور آماده دریافت درخواست است.
۴. رابط درخواست شی در سمت کلاینت، پارامترهای احضار شده متد را می پذیرد و پارامترها را در شبکه تجمیع میکند.
۵. رابط درخواست شی در سمت سرور، پارامترها را تفکیک میکند و آن ها را به شی سرور تحویل میدهد.
۶. مقدار برگشتی به ترتیب در سمت سرور و سمت کلاینت به روشی مشابه، تجمیع و تفکیک میشود.



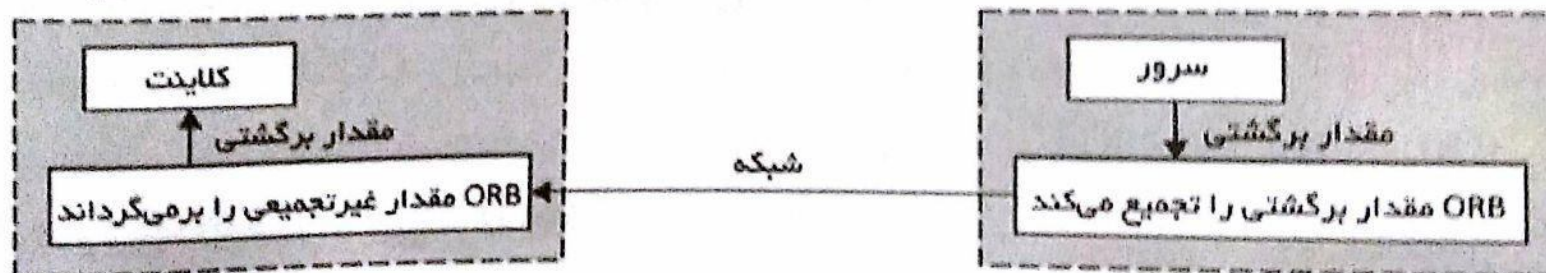
مرحله ۱

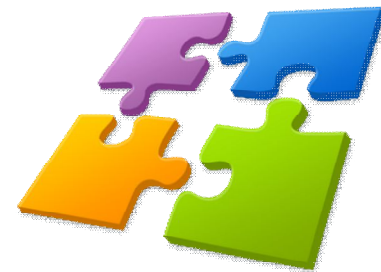


مرحله ۲



مرحله ۳





مدل CORBA

زبان تعریف رابط / IDL

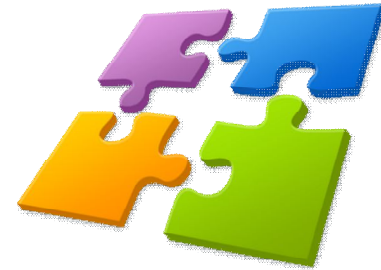
- در روش RPC، رابط ها به یک زبان خاص نوشته میشود، که برنامه های کلاینت و سرور نیز به همان زبان نوشته شده اند.

• IDL:

- ابزاری است برای معرفی رابط ها به روشی مستقل از زبان.
- یک زبان برنامه نویسی نیست و فقط می تواند برای تعریف رابط ها استفاده شود.
- صرف نظر از جزئیات تنها نام و امضای متدها را تعیین می کند.
- وابسته به هیچ زبان برنامه نویسی خاصی نیست.
- مسئول تبادل صحیح داده بین زبان های متفاوت است، زیرا رابط های تعریف شده توسط IDL، می توانند به هر زبانی نگاشت شوند.

مدل CORBA

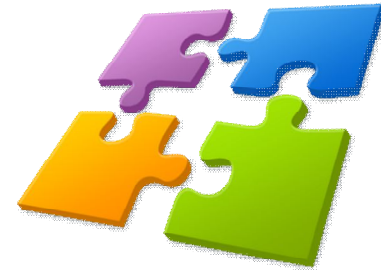
زبان تعریف رابط / IDL ...



- نگاشت های استاندارد برای برخی زبان های برنامه نویسی رایج مانند C، C++، Java، COBOL، Smalltalk و ... وجود دارد.
- این نگاشت ها، ساختارهای زبان IDL را به ساختارهای یک زبان ویژه نگاشت میکنند
- به عنوان مثال در Java، رابط IDL به رابط جاوا نگاشت می شود و در C++ رابط IDL به کلاس C++ نگاشت میشود.
- این نگاشت با استفاده از یک ابزار خودکار انجام میشود که فایل IDL را به عنوان ورودی میگیرد و تعدادی فایل تولید می کند که برای توسعه کلاینت و سرور به زبانی معین استفاده می شوند.
- به عنوان مثال، ابزار idl2j فایل های مناسب را برای توسعه کلاینت و سرور به زبان جاوا تولید میکند.

CORBA مدل

زبان تعریف رابط / IDL ...

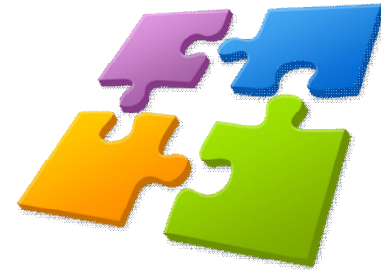


Listing 5.1: A sample of IDL interface definition

```
module Test {  
    interface square {  
        attribute double arg1;  
        double getSquare (in double arg1);  
    };  
};
```

مدل CORBA

مدل شی CORBA

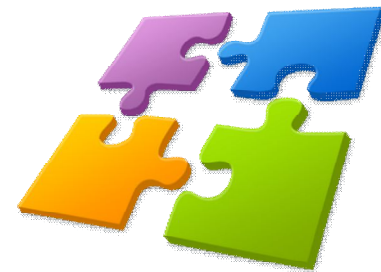


• عناصر یک شی مدل CORBA:

- پشتیبانی از توزیع تقریباً شفاف اولیه
- ارجاعات شی
- آداپتورهای شی (که باعث میشود اشیای توزیع شده با استفاده از رابطهای درخواست شی با یکدیگر ارتباط برقرار کنند).

مدل CORBA

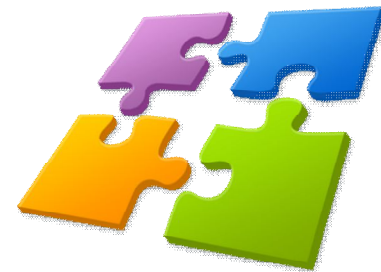
مدل شی CORBA ...



- توزیع شفاف:
 - توزیع اشیاء در صورتی شفاف است که کلاینت از سرویس های ارائه شده توسط شی دیگری که از مکان آن تقریباً بی اطلاع است، استفاده میکند.
 - این شی دیگر ممکن است روی ماشین راه دور یا به صورت محلی قرار گیرد.
- دو روش برای برقراری ارتباط مولفه یک برنامه کاربردی با مولفه ای در یک برنامه کاربردی دیگر، در اشیای توزیع شده:
 - ارسال با مقدار / Passing By Value
 - ارسال با مرجع / Passing By Reference
- CORBA از روش ارسال با مرجع استفاده میکند.

مدل CORBA

مدل شی CORBA ...

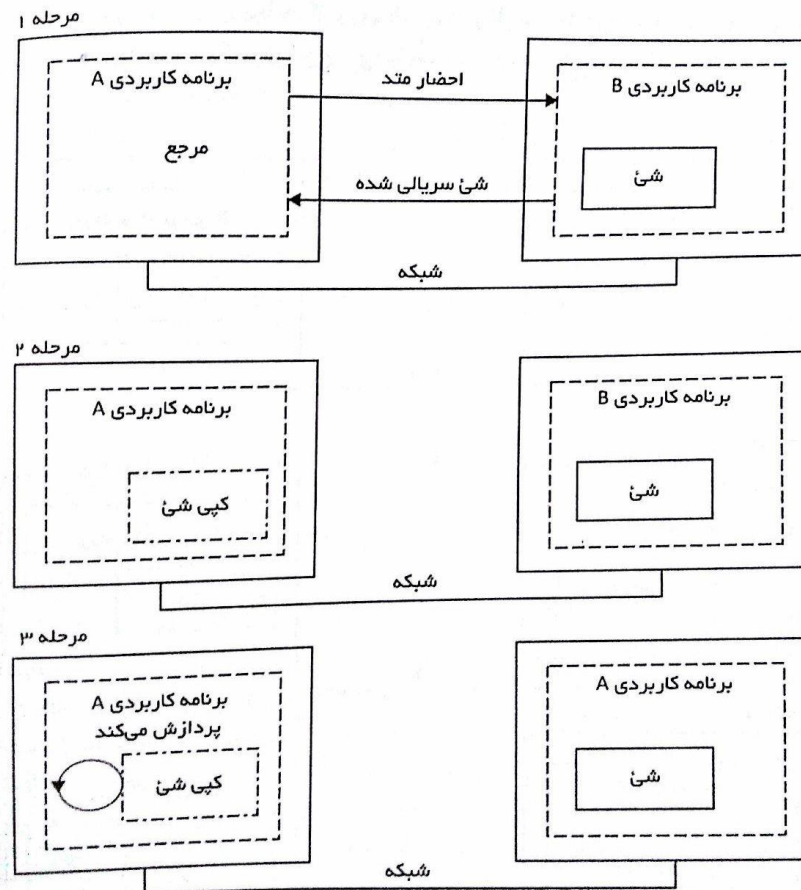
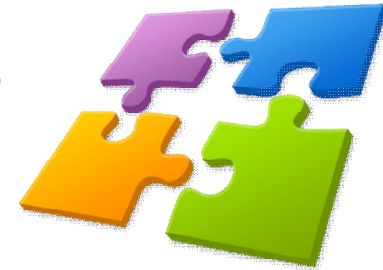


ارسال با مقدار / Passing By Value

- رونوشتی از شی فراخوانی شده، ایجاد میشود که با فرآیندی به نام سریالیزه کردن به برنامه کلاینت ارسال میشود.
- برنامه کلاینت متدی روی رونوشت شیئی که از سرور دریافت کرده، احضار میکند.
- در این روش، عملیات انجام شده منجر به تغییر وضعیت شی در سرور نمی شود.
- به عنوان مثال:
 - برنامه A، متدی روی شی راه دور در برنامه B را احضار میکند.
 - برنامه B رونوشتی از شی فراخوانی شده را از طریق سریالیزه کردن ایجاد میکند و رونوشت ایجاد شده را به برنامه A ارسال میکند.
 - برنامه A رونوشتی از شی به صورت محلی ایجاد میکند و متدی روی این رونوشت فراخوانی میکند.

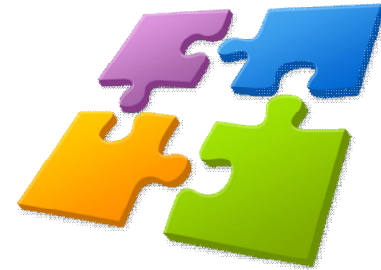
مدل CORBA

مدل شی CORBA ...



مدل CORBA

مدل شی CORBA ...

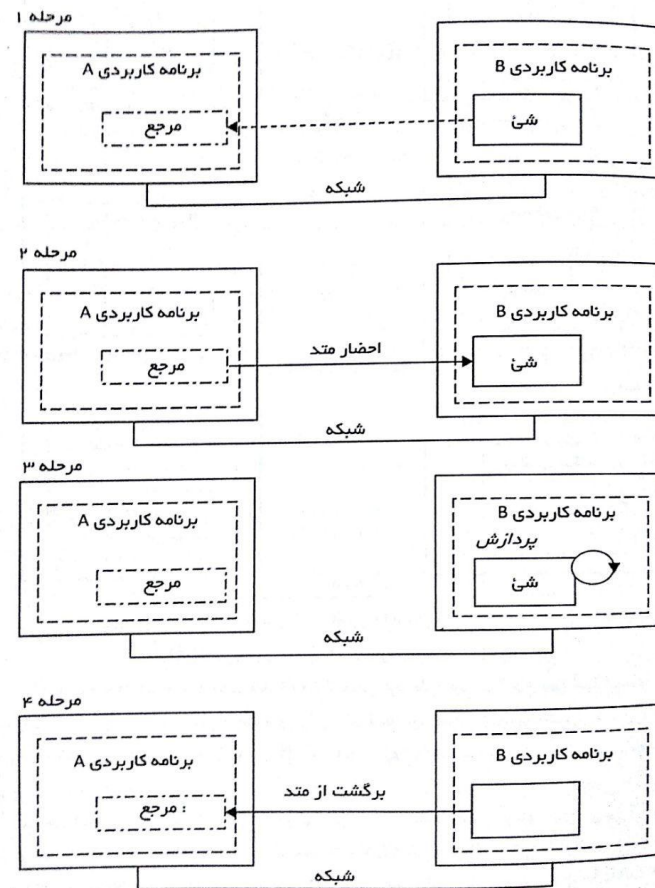
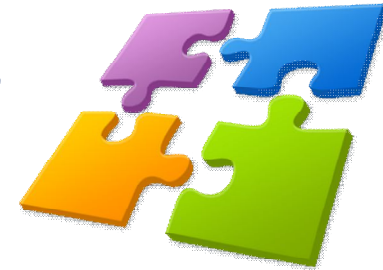


ارسال با مرجع / Passing By Reference

- در این روش، شی در اختیار برنامه سرور قرار میگیرد و مرجع شی به کلاینت ارسال میشود.
- به عنوان مثال:
 - ابتدا برنامه A مرجعی به شی راه دور در برنامه B دریافت میکند.
 - سپس برنامه A از این مرجع برای فراخوانی متد راه دور، روی یک شی راه دور استفاده میکند.
 - سپس برنامه B متد را پردازش میکند.
 - سرانجام شی راه دور، پاسخ را به برنامه A ارسال میکند.

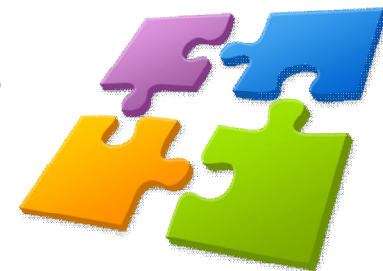
مدل CORBA

مدل شی CORBA ...

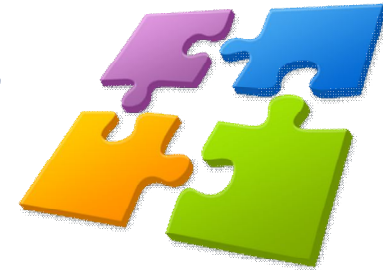


مدل CORBA

مدل شی CORBA ...



- آداپتور شی:
 - هدف اصلی آداپتور شی، برقراری ارتباط بین پیاده سازی شی با رابط درخواست شی است.
- در CORBA سه نوع آداپتور شی استفاده میشود که مهم ترین آنها آداپتور شی ابتدایی / BOA است.
- این آداپتور به اشیای CORBA، مجموعه رایجی از متدها برای دسترسی به توابع رابط درخواست شی ارائه میدهد.
- این توابع شامل فعال سازی، احراز هویت، و پایداری شی است.
- هر پیاده سازی از رابط درخواست شی مطابق با CROBA، باید یک آداپتور شی ابتدایی را ارائه دهد.



مدل CORBA

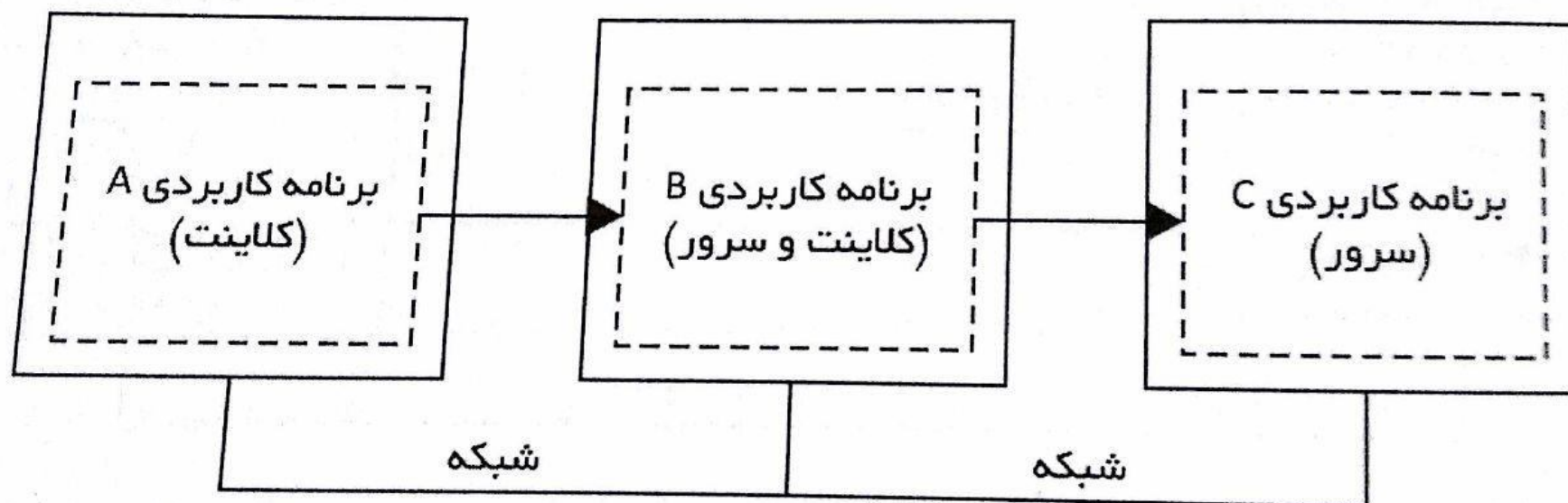
نقش های کلاینت و سرور در CORBA

- CORBA مانند RPC مفهوم اصلی کلاینت و سرور رو حفظ میکند اما تمایز بین کلاینت و سرور در CORBA تا حدودی مبهم است. چون روابط آنها به صورت همتا به همتا است.
- **سرور:** هر برنامه کاربردی که شی را ایجاد میکند و به سایر برنامه های کاربردی امکان مشاهده این شی را میدهد.
- **کلاینت:** سایر برنامه های کاربردی که از سرویس های ارائه شده توسط سرور، استفاده می کنند.
- یک برنامه کاربردی می تواند برای برخی از برنامه های کاربردی به عنوان کلاینت و برای برنامه های کاربردی دیگر به عنوان سرور عمل کند.



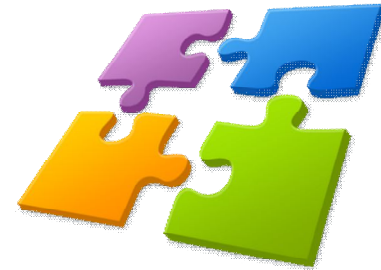
مدل CORBA

نقش های کلاینر و سرور در CORBA ...



مدل CORBA

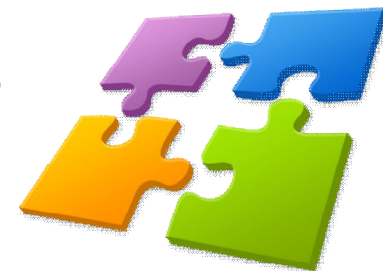
مدل ارتباطات CORBA



- در یک محیط توزیع شده، برنامه های کاربردی باید از طریق شبکه با هم ارتباط برقرار کنند.
- شبکه در بالای لایه فیزیکی (که میتواند به صورت سیمی یا بی سیم باشد)، از یک لایه شفاف تشکیل شده که مسئول انتقال بسته های داده از یک نقطه به نقطه دیگر است.
- متداولترین پروتکل انتقال، پروتکل TCP/IP و پس از آن پروتکل DCE است.
- CORBA از یک پروتکل استاندارد به نام IIOP، بین رابط های درخواست شی استفاده میکند.
- این پروتکل بر اساس پروتکل TCP/IP ساخته میشود و تعامل پذیری چندمحیطی صحیح را تضمین میکند.

مدل CORBA

سرویس های CORBA



- سرویس نامگذاری:
- باعث میشود اشیای CORBA با یک نام رجیستر و مکان یابی شوند.
- به اشیای CORBA نام هایی داده میشود که اشیای دیگر با این نام ها میتوانند آنها را جستجو کنند.
- سرویس کنترل همزمانی:
- رابطی برای کنترل همزمانی در اشیای به اشتراک گذاشته شده CORBA ارائه میدهد.
- این کار با استفاده از چند نوع قفل مانند قفلهای خواننده-نویسنده، و قفل های هدف، انجام میشود.



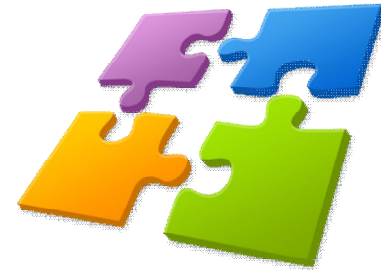
مدل CORBA سرویس های CORBA ...

- سرویس تراکنش:
- بخش جدایی ناپذیر هر برنامه کاربردی مهم
- تراکنش باید تجزیه ناپذیر باشد.
- به عنوان مثال برای هماهنگی انتقال بین دو حساب بانکی، عملیات به صورت تراکنش انجام میشود یعنی یا موجودی هر دو حساب به طور صحیح به روز می شود یا هیچ کدام.
- سرویس چرخه حیات:
- مسئول ایجاد، حذف، کپی و انتقال اشیای CORBA



مدل CORBA سرویس های CORBA ...

- سرویس امنیت:
- امنیت در محیط یکپارچه سازی بسیار اهمیت دارد، زیرا هر برنامه کاربردی در دسترس همه برنامه های کاربردی دیگر سازمان است.
- اجزای سرویس امنیت:
- سرویس احراز هویت:
- برای تایید هویت مورد ادعای کاربر
- سرویس تعیین حق دسترسی:
- کنترل دسترسی به سرویسها یا اشیای مختلف



مدل CORBA

سرویس های CORBA ...

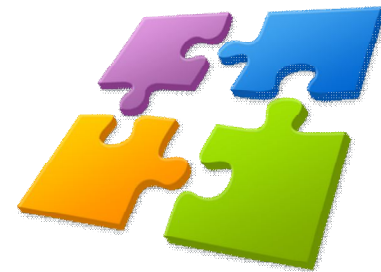
• اجزای سرویس امنیت:...

- سرویس ممیزی امنیت:
- حفظ سوابق کلیه اعمال کاربر

• سرویس انکارناپذیری:

- امکاناتی شبیه امضای دیجیتال را ارائه میدهد، به این معنا که منبع ارسال داده، یا دریافت کننده داده، میتواند به طور غیرقابل انکار ثابت شود.

مقایسه روش RPC و مدل شی توزیع شده



- مزایای مدل شی توزیع شده نسبت به روش RPC:

- با استفاده از رابط درخواست شی، مساله استفاده مجدد و ارتباط نقطه به نقطه، حل شد.
- با استفاده از زبان تعریف رابط، مساله وابستگی به زبان حل شد.

- مشکلات RPC که در مدل شی توزیع شده نیز حل نشد:

- مساله مقیاس پذیری
- این مساله به دلیل ماهیت ذاتی فراخوانی همگام است که باعث میشود کلاینت از انجام کار بیشتر باز بماند تا زمانی که سرور کارش را تمام کند و کنترل را به کلاینت برگرداند.



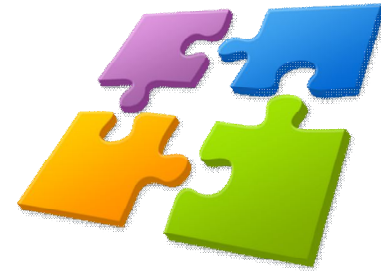
یکپارچه سازی کاربردهای سازمانی

فصل پنجم: پیام رسانی

بتول لکزائی

b_lakzaei@yahoo.com

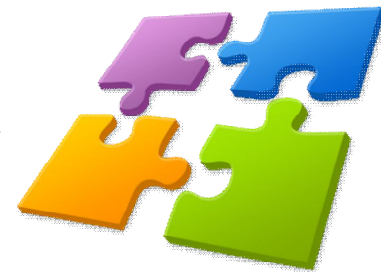
معایب RPC و اشیای توزیع شده



- اگرچه اشیای توزیع شده، بسیاری از معایب روش RPC را برطرف کردند، اما دو عیب بسیار مهم روش RPC را از قلم انداختند:
 ۱. RPC و اشیای توزیع شده، هر دو از روش ارتباط همگام استفاده می کنند.
 - پیوند بین سرور و کلاینت، یک پیوند قوی است.
 - مقیاس پذیری پایین است.
 ۲. در روشهای RPC و اشیای توزیع شده، ارتباط بین برنامه ها کابردی، قابل اعتماد نیست.
 - تضمینی وجود ندارد که پیام و مقدار بازگشتی به مقصد تحویل داده شود.
 - ممکن است کلاینت در مواردی مانند توقف در اتصال شبکه یا زمانی که سرور فعال نیست، در عملیات خود دچار وقفه شود.

راه حل این مسائل: پیام رسانی ناهمگام

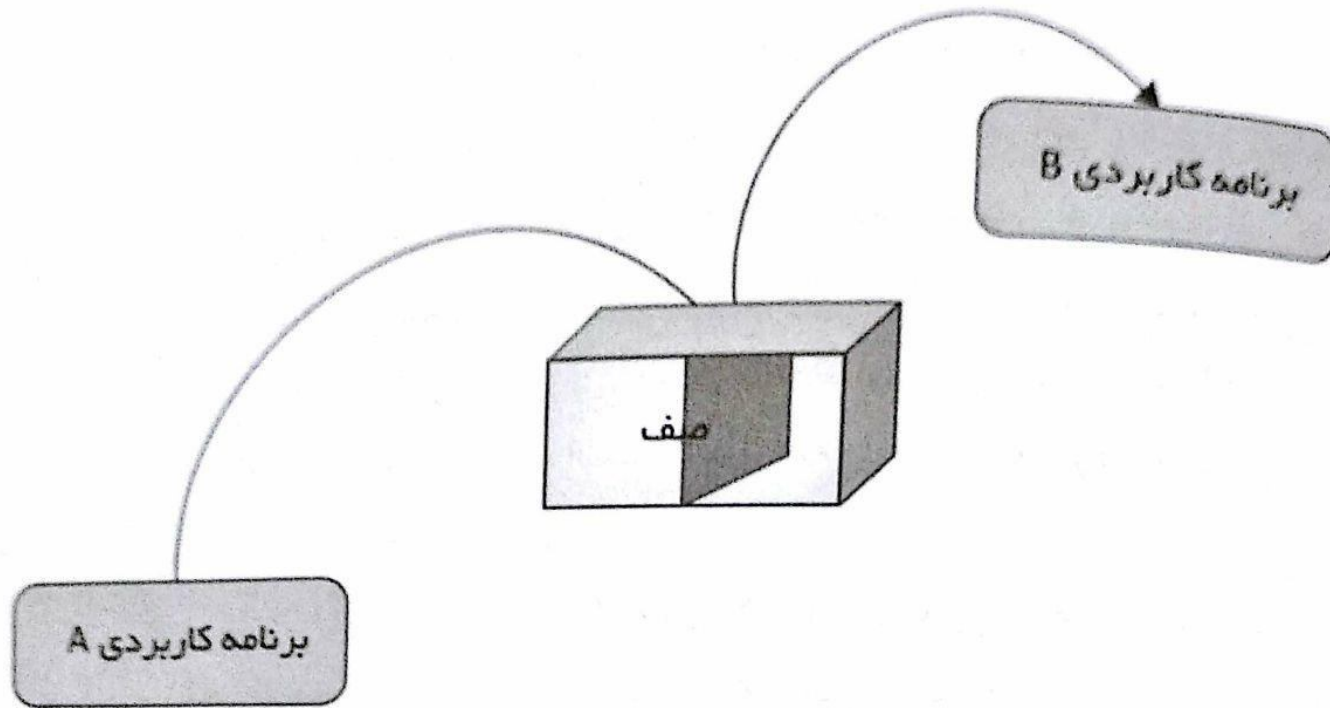
پیام رسانی ناهمگام



- کلاینت، پیامی را برای سرور ارسال میکند اما برای ادامه کار خود، منتظر دریافت پاسخ از سرور نمی ماند.
- این ناهمگامی منجر به انفصال پیوند قوی بین کلاینت و سرور و در نتیجه افزایش مقیاس پذیری میشود.
- در پیام رسانی، برنامه های کاربردی به طور مستقیم با هم در ارتباط نیستند و لینک اختصاصی بین آنها وجود ندارد. بلکه از طریق صف/کانال به طور غیر مستقیم با هم ارتباط برقرار میکنند.
- **صف:** مجموعه ای از پیام ها که میتواند بین چند کامپیوتر به اشتراک گذاشته شود.

پیام رسانی ناهمگام ...

- ابتدا برنامه A پیامی به صف می فرستد. پس از اینکه برنامه A پیام را به صف تحویل داد، برنامه B پیام را از صف بازیابی میکند.

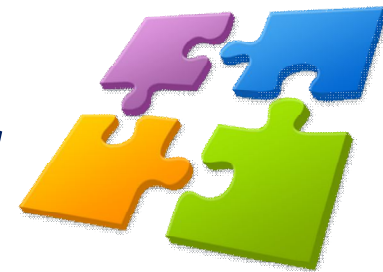


پیام رسانی ناهمگام ...

میان افزار پیام گرا / MOM (Message-Oriented Middleware)

- در پیام رسانی ناهمگام مانند رابط درخواست شی یا اشیای توزیع شده، کد ارتباطات و تجمیع به عنوان یک مولفه نرم افزاری مجزا، نوشته میشود که اجازه استفاده مجدد از کد را می دهد (یعنی چند برنامه کاربردی میتوانند از یک کد برای برقراری ارتباط با یکدیگر و با برنامه های کاربردی روی ماشین های دیگر، استفاده کنند). این مولفه نرم افزاری مجزا، سیستم پیام رسانی یا میان افزار پیام گرا نامیده میشود.

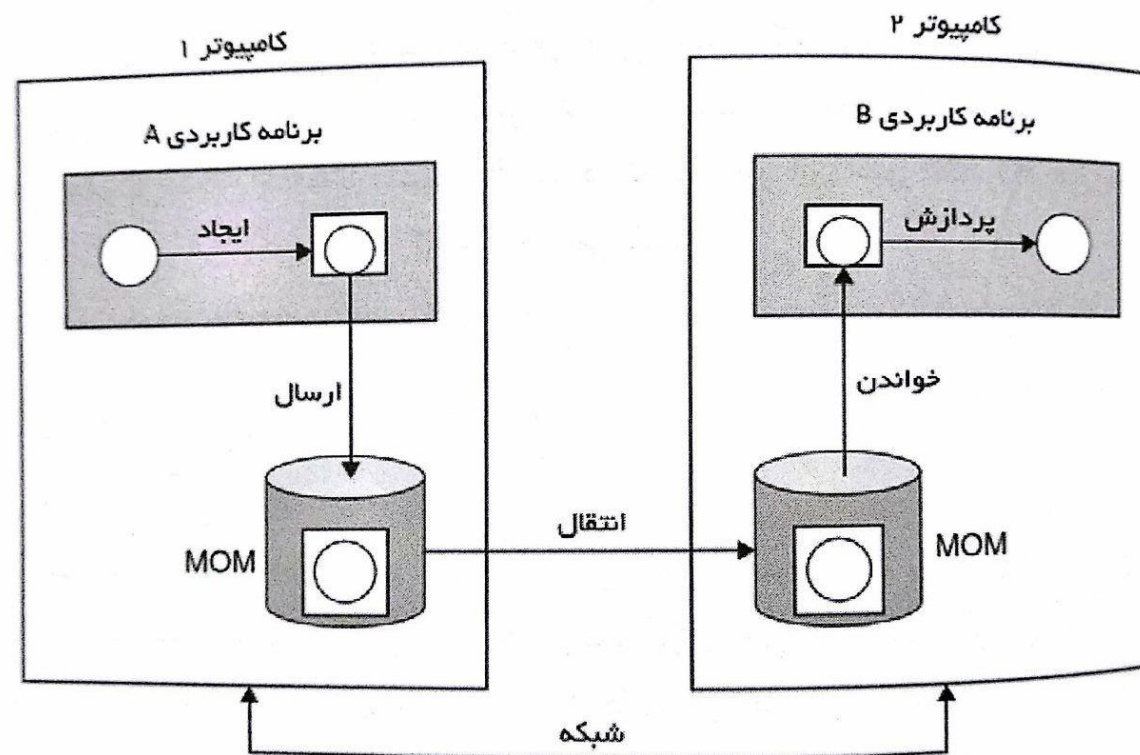
پیام رسانی ناهمگام ...



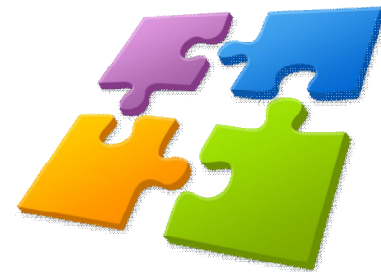
• مراحل تبادل پیام ناهمگام:

۱. برنامه A پیامی ایجاد میکند و داده را در آن قرار می دهد.
۲. برنامه A این پیام را به صف درون سیستم پیام رسانی (MOM) ارسال میکند.
۳. سیستم پیام رسانی، پیام را از گیرنده به فرستنده منتقل میکند و آن را در دسترس گیرنده قرار میدهد.
۴. برنامه B پیام را از روی صف میخواند.

پیام رسانی ناهمگام ...



ویژگی های پیام رسانی ناهمگام



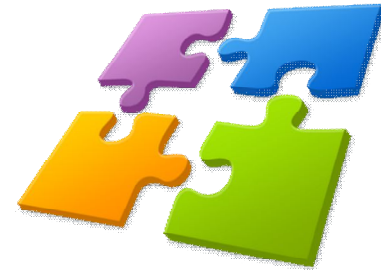
۱. ارتباط ناهمگام بین کلاینت و سرور

- کلاینت پس از ارسال درخواست یک عملکرد روی سرور، منتظر دریافت پاسخ نمی ماند و همزمان به انجام کارهای دیگر می پردازد.

۲. همگام سازی برخی برنامه ها هنگامی که به شبکه متصل می شوند

- برخی برنامه های کاربردی به طور خاص به گونه ای طراحی میشوند تا جدای از شبکه اجرا شوند، ولی هنگامی که اتصال شبکه موجود است با سرورها همگام میشوند.
- مانند برخی برنامه های کاربردی روی لپ تاپ و گوشی موبایل
- در این برنامه ها، داده ای که باید ارسال شود، به محض تولید در صف قرار میگیرد و منتظر اتصال برنامه کاربردی به سرور می ماند.

ویژگی های پیام رسانی ناهمگام ...



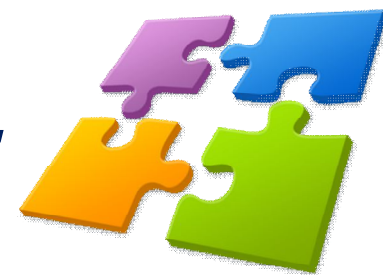
۳. قابلیت اعتماد

- سیستم پیام رسانی، تحویل پیام به مقصد را با استمرار پیام رسانی تضمین میکند. یعنی آنقدر ارسال پیام را تکرار میکند تا زمانی که پیام تحویل داده شود.

۴. اجتناب از عبور تکه تکه (Throttling)

- در روشهای RPC و اشیای توزیع شده، یک سرور واحد ممکن است با درخواست های کلاینت های مختلف، دچار اضافه بار شود. این مساله میتواند منجر به افت عملکرد و حتی توقف سرور شود.
- سیستم پیام رسانی، تا زمانی که سرور آماده پذیرش درخواست ها میشود، آنها را صف بندی میکند. بنابراین سرور میتواند سرعت پردازش درخواست ها را طوری کنترل کند که با دریافت تعداد زیادی درخواست های همزمان دچار اضافه بار نشود.
- کلاینت ها تحت تاثیر این عبور تکه تکه قرار نمیگیرند، زیرا ارتباطات ناهمگام است و در نتیجه از ادامه کار خود باز نمی مانند.

عناصر سیستم پیام رسانی



عناصر سیستم پیام رسانی

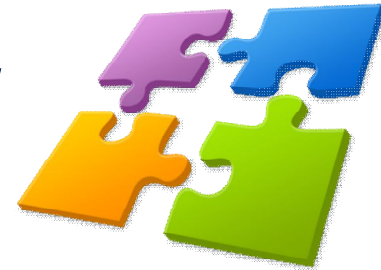
کانال ها (صف ها)

پیام ها

نقاط پایانی

عناصر سیستم پیام رسانی

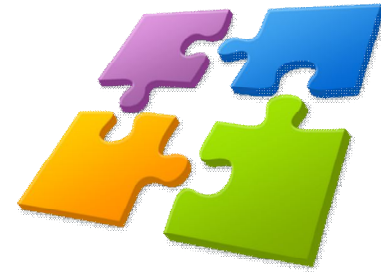
کانال ها



- کانال ها برای انتقال داده استفاده می شوند، و هر کانال به عنوان یک پایپ مجازی (Virtual Pipe) عمل میکند که گیرنده را به فرستنده متصل میکند.
- هنگام ایجاد یک سیستم پیام رسانی، کانالها به طور خودکار ایجاد نمیشوند، بلکه برنامه نویس باید تعیین کند که برنامه های کاربردی چگونه ارتباط برقرار کنند و سپس کانالهای مناسب را ایجاد کند.
- انواع کانال
 - کانال نقطه به نقطه / Point to Point
 - کانال انتشار و اشتراک / Publish and Subscribe

عناصر سیستم پیام رسانی

کانال ها ...



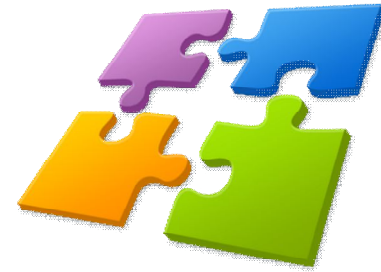
کانال نقطه به نقطه

- در این مدل، کلاینت ها از طریق کانال های مجازی به نام صف، به طور ناهمگام پیام ها را ارسال و دریافت می کنند.
- این مدل، مبتنی بر مدل نظرسنجی یا کشش (Pull-or-Polling-Based Model) است، که در آن پیام ها به جای اینکه به صورت خودکار به سمت سرور ارسال شوند، باید از صف درخواست شوند.
- هدف این مدل، تحویل یک به یک پیام ها است.
- هر صف ممکن است چند گیرنده داشته باشد، اما فقط یک گیرنده میتواند هر پیام را دریافت کند.
- سیستم پیام رسانی (که گاهی اوقات ارائه دهنده JMS نامیده میشود) پیام ها را بین گیرنده ها تقسیم میکند، بنابراین تضمین میشود که هر پیام فقط توسط یک گیرنده استفاده میشود.

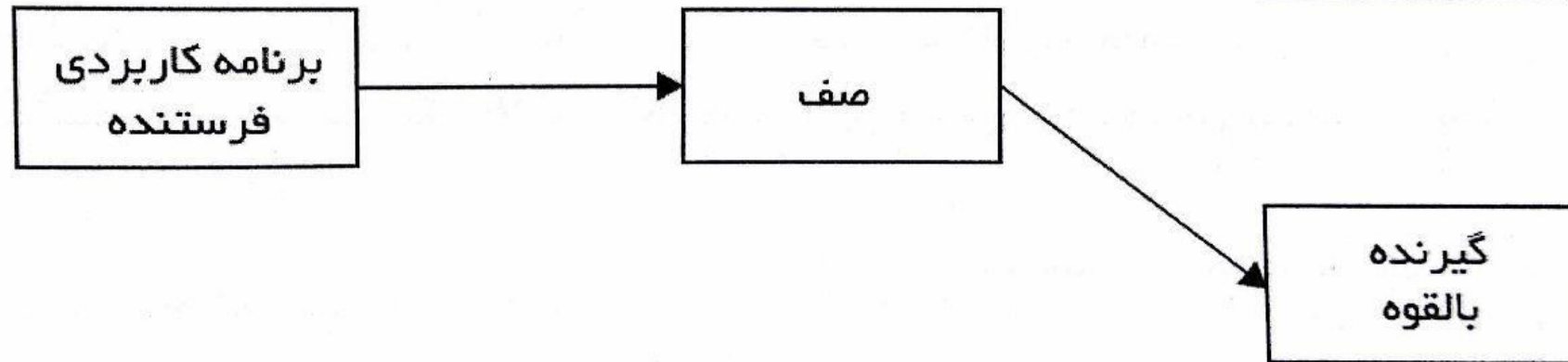
عناصر سیستم پیام رسانی

کانال ها ...

کانال نقطه به نقطه ...

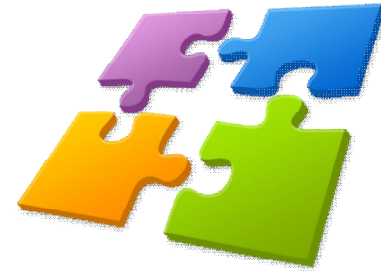


نقطه به نقطه (یک به یک)



عناصر سیستم پیام رسانی

کانال ها ...



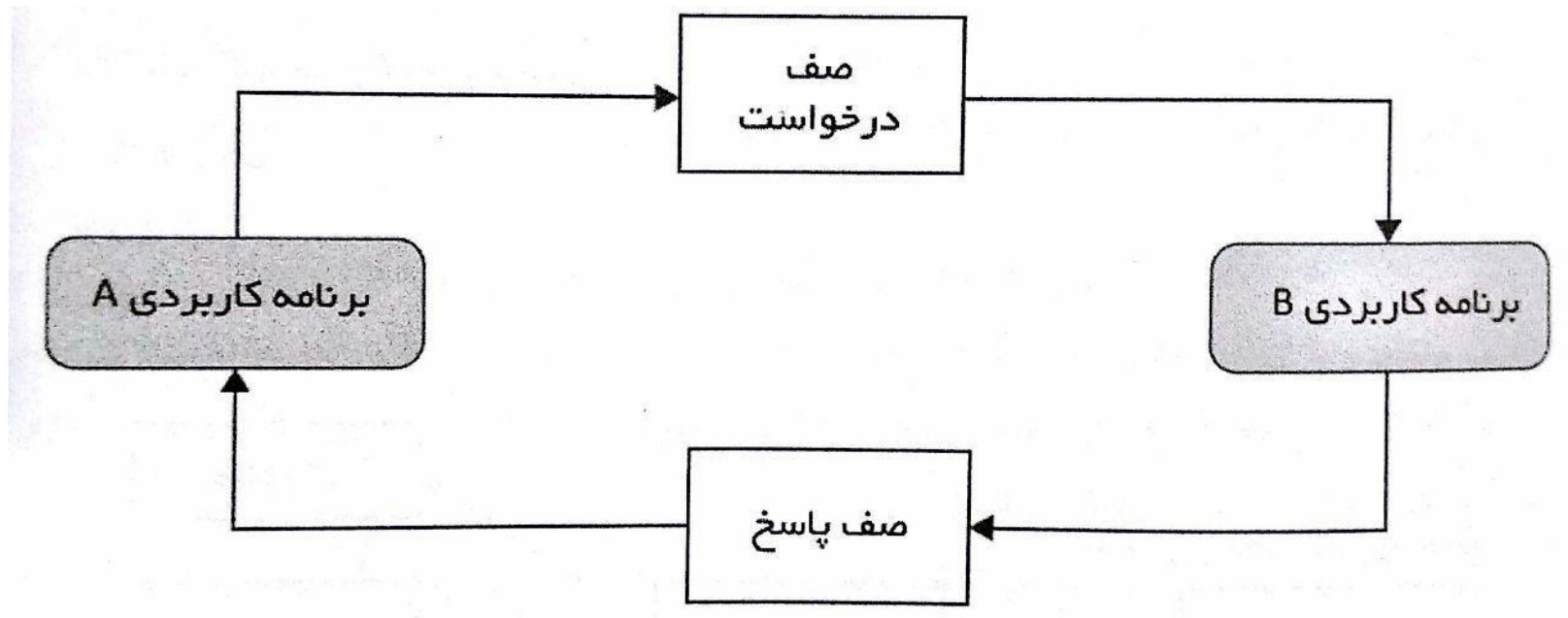
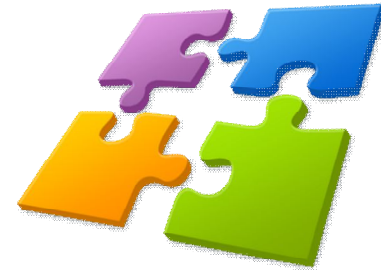
کانال نقطه به نقطه ...

- مدل پیام رسانی ناهمگام نقطه به نقطه میتواند برای شبیه سازی پیام رسانی یا تعادل همگام بین دو برنامه کاربردی نیز استفاده شود:
- در این مدل، از دو صف استفاده میشود: صف درخواست برای تحویل درخواست ها، و صف پاسخ: برای دریافت مقادیر بازگشتی
- صف درخواست برای برنامه درخواست کننده (برنامه A) صف خروجی است درحالی که همزمان به عنوان صف ورودی برای برنامه دریافت کننده (برنامه B) استفاده میشود.
- به طور مشابه، صف پاسخ به عنوان صف خروجی برای برنامه کاربردی B و به عنوان صف ورودی برای مقدار بازگشتی برنامه A استفاده میشود.
- برای ایجاد همبستگی میان درخواست و پاسخ، یک ID همبستگی به سرصفحه پاسخ ضمیمه میشود. مقدار این ID همبستگی معمولاً ID پیام درخواست است.

عناصر سیستم پیام رسانی

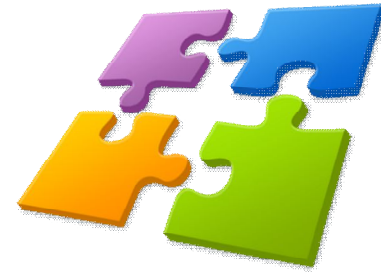
کانال ها ...

کانال نقطه به نقطه ...



عناصر سیستم پیام رسانی

کانال ها ...



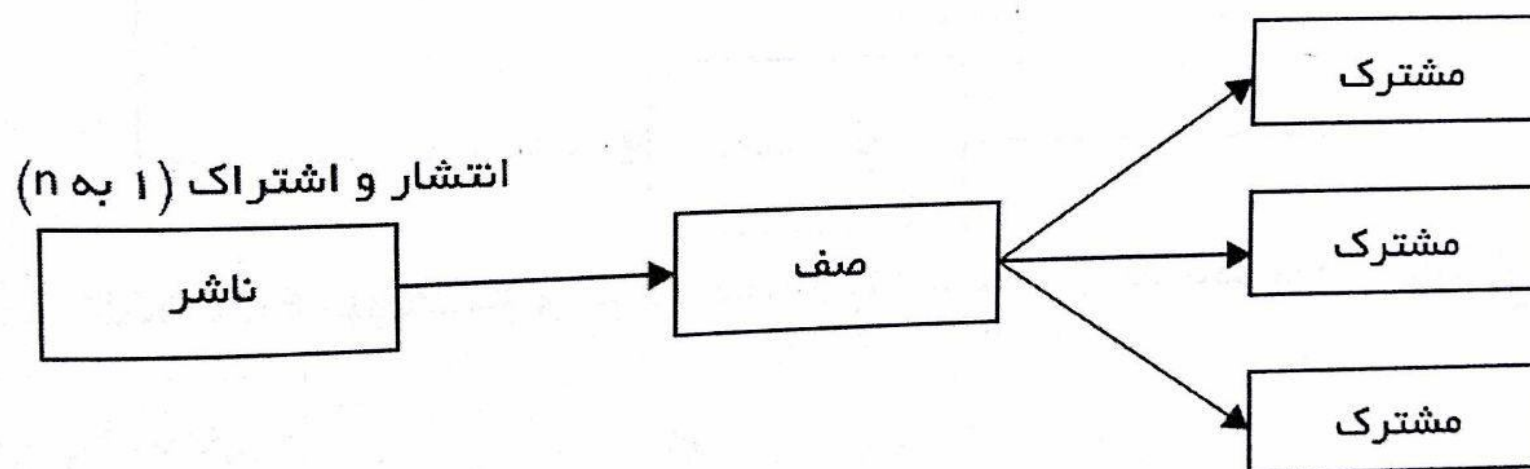
کانال انتشار و اشتراک

- در این مدل، یک تولید کننده می تواند از طریق یک کانال مجازی به نام تاپیک (Topic) به هر تعداد از مصرف کنندگان پیام ارسال کند.
- مصرف کنندگان میتوانند مشترک تاپیک شوند.
- پیام هایی که به یک تاپیک آدرس دهی میشوند، به کلیه مشترکین تحویل داده می شوند و هر مشترک یک رونوشت از پیام را دریافت میکند.
- این مدل، یک مدل مبتنی بر فشار (Push-based-Model) است که در آن پیام ها به طور خودکار بین مشترکین، پخش می شوند.

عناصر سیستم پیام رسانی

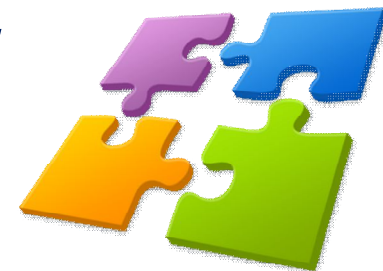
کانال ها ...

کانال انتشار و اشتراک ...



عناصر سیستم پیام رسانی

کانال ها ...

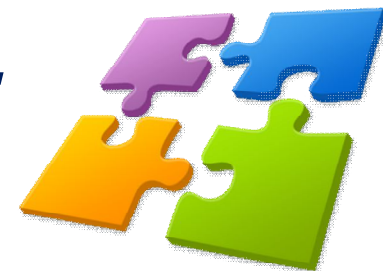


کانال انتشار و اشتراک...

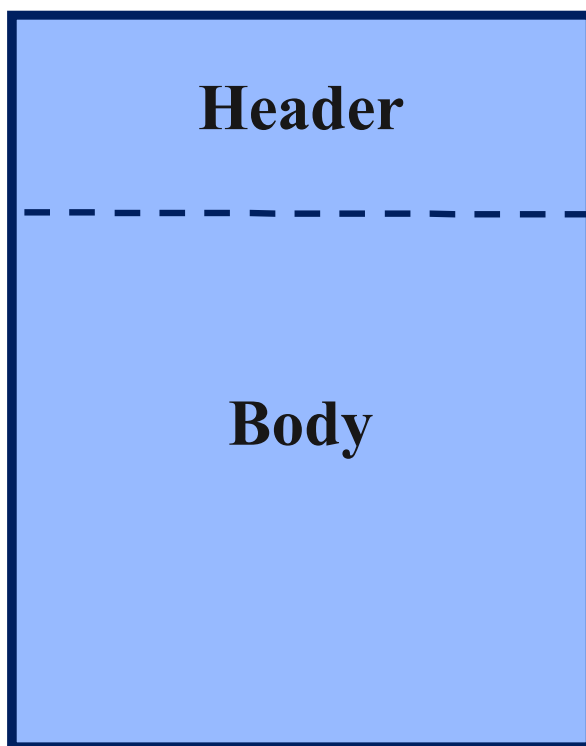
- در مدل پیام رسانی انتشار و اشتراک، ناشر ممکن است هیچ توجهی به این مساله نکند که آیا همه در حال گوش دادن هستند یا حتی هیچ کس گوش نمی دهد.
- به عنوان مثال ناشری را در نظر بگیرید که قیمت های سهام را پخش میکند. اگر هیچ مشترک خاصی در حال گوش دادن نباشد و مبلغ زیادی را از دست بدهد، باز هم هیچ اهمیتی نمیدهد.

عناصر سیستم پیام رسانی

پیام ها



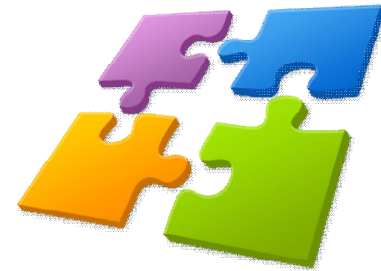
- پیام ها، داده ها را برای انتقال، کپسوله می کنند.
- هر پیام شامل دو بخش است:
 - سرصفحه
 - بدنه



عناصر سیستم پیام رسانی

پیام ها

سرصفحه



- اطلاعات سرصفحه توسط سیستم پیام رسانی مورد استفاده قرار میگیرد و شامل اطلاعاتی در مورد مبدا و هویت پیام و اطلاعات کنترلی و ... است.

- هر سر صفحه تعدادی ویژگی (Attribute) دارد. برخی از ویژگی ها عبارتند از:

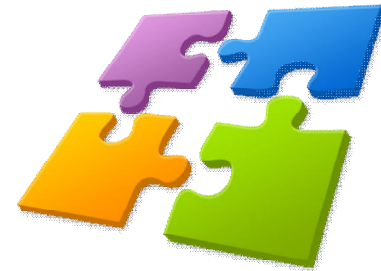
۱. ID پیام / ID همبستگی

- برای شناسایی یک درخواست خاص، یا پاسخ دادن به پیام
- قبل از اینکه پیام درخواست در صف قرار گیرد، ID ها ذخیره میشود و از آن در عملیات بعدی برای پیام پاسخ استفاده میشود. برنامه ای که پیام درخواست را دریافت میکند این اطلاعات را در پیام پاسخ کپی میکند.
- این کار باعث میشود، برنامه مبدا (که پاسخ را دریافت میکند) به سیستم پیام رسانی دستور دهد، به جای دریافت اولین پیام در صف، در جستجوی پیام خاصی در صف باشد.

عناصر سیستم پیام رسانی

پیام ها

سرصفحه ...



۲. پایدار / ناپایدار

- پیام های پایدار همیشه به مقصد می رسند، حتی هنگامی که سیستم از کار بیفتد.
- این پیام ها روی دیسک ذخیره میشوند.
- میتوان یک پیام خاص، یا کلیه پیام های یک صف خاص را پایدار کرد.

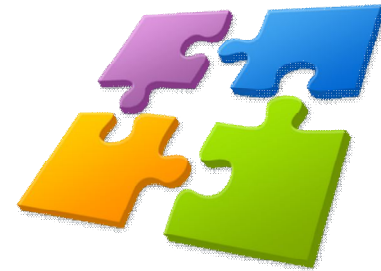
۳. آدرس برگشت

- آدرس برگشت برای پیام های درخواست/پاسخ اهمیت دارد.
- برنامه نویس باید به برنامه سرور بگوید که پیام پاسخ را کجا بفرستد.
- کلاینت ها و سرور رابطه یک به چند دارند و معمولاً برنامه سرور نمیتواند از روی داده کاربر تشخیص دهد که پیام درخواست از سمت کدام برنامه آمده است. بنابراین در سرصفحه پیام، یک آدرس Reply-to-Queue و Reply-to-Queue-Manager ایجاد میکند.

عناصر سیستم پیام رسانی

پیام ها

سرصفحه ...



۴. اولویت

- میتوان به هر پیام یک اولویت داد و با پیاده سازی کانال به صورت یک صف اولویت، ابتدا پیام هایی را ارسال کرد که اولویت بالاتری دارند.

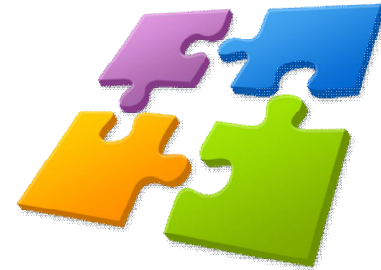
۵. اطلاعات گروه بندی / بخش بندی

- پیام ها میتوانند بخش بندی یا گروه بندی شوند.
- بخش بندی-روش اول:
- بخش بندی میتواند برای برنامه نویس برنامه کاربردی، شفاف شود. یعنی مدیر صف در صورت داشتن مجوز، پیام بزرگ را هنگامی که در یک صف نمی گنجد، بخش بندی میکند.
- در انتهای دریافت، برنامه کاربردی این حق را دارد که کل پیام را یک جا یا هر بخش را جداگانه دریافت کند. که این مساله به بافری بستگی دارد که در دسترس برنامه کاربردی است.

عناصر سیستم پیام رسانی

پیام ها

سرصفحه ...



۵. اطلاعات گروه بندی / بخش بندی

- بخش بندی - روش دوم:

- کنترل در اختیار برنامه نویس است.
- بنابراین برنامه نویس میتواند پیام را طبق مرزهای منطقی یا اندازه بافری که در دسترس برنامه است، تقسیم بندی کند.
- برنامه نویس هر بخش را به عنوان یک پیام فیزیکی مجزا قرار میدهد بنابراین چند پیام فیزیکی، یک پیام منطقی را می سازند.
- مدیر صف تضمین می کند که ترتیب بخشها حفظ شود.

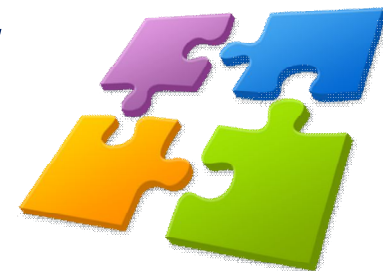
- گروه بندی:

- برای کاهش ترافیک شبکه میتوان چند پیام کوچک را گروه بندی کرد و یک پیام فیزیکی بزرگتر ساخت. سپس این پیام به مقصد ارسال و در آنجا تجزیه میشود.
- گروه بندی پیام تضمین میکند که ترتیب ارسال پیام ها حفظ میشود.

عناصر سیستم پیام رسانی

پیام ها

سرصفحه ...



۶. تاریخ و زمان

۷. عمر پیام

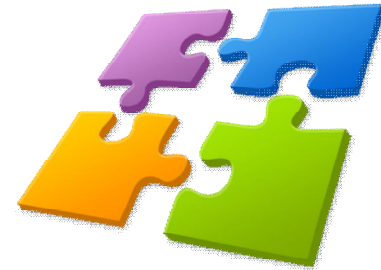
- میتوان برای هر پیام یک تاریخ انقضا تعیین کرد. هنگامی که این تاریخ به پایان میرسد، پیام رها خواهد شد.
- در این جا، Daemon وجود ندارد که صف را برای پیام های منقضی شده بررسی کند.
- پیامهای منقضی شده ممکن است هفته ها در یک صف بمانند تا زمانی که برنامه ای تلاش کند، آنها را بخواند.

۸. نسخه

عناصر سیستم پیام رسانی

پیام ها

بدنه

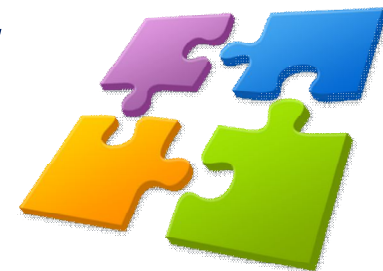


- بدنه شامل داده واقعی است که توسط گیرنده مورد استفاده قرار می گیرد و پردازش می شود.
- داده موجود در بدنه می تواند انواع مختلفی داشته باشد:
- داده ممکن است یک **پیام دستوری** باشد که برای احضار یک رویه در برنامه کاربردی گیرنده استفاده میشود.
- داده ممکن است یک **سند** باشد که برای انتقال داده از یک برنامه کاربردی به برنامه کاربردی دیگر استفاده می شود.
- داده ممکن است یک **رویداد** باشد که برنامه کاربردی گیرنده را از وقوع رویدادی در برنامه کاربردی فرستنده آگاه میکند.

عناصر سیستم پیام رسانی

پیام ها

بدنه ...

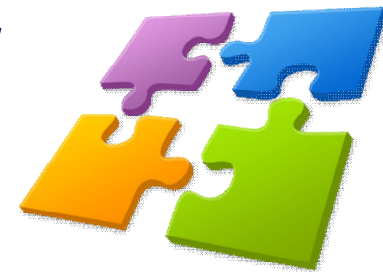


- JMS: یک رابط برنامه نویسی استاندارد که میتواند برای دسترسی به سیستم های پیام رسانی مورد استفاده قرار گیرد.
- در JMS یک پیام توسط Type Message نشان داده میشود که شامل چندین زیر نوع است.
- در همه زیرنوع ها ساختار سرصفحه یکسان است و بدنه انواع متفاوتی دارد.

عناصر سیستم پیام رسانی

پیام ها

بدنه ...

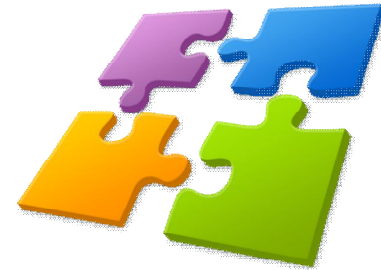


- انواع پیام هایی که توسط JMS پشتیبانی میشود:
- **پیام متنی:**
 - متداولترین نوع پیام
 - بدنه یک String است مانند متن لفظی یا یک سند XML
- **پیام بایتی:**
 - ساده ترین و عمومی ترین نوع پیام
 - بدنه پیام، آرایه ای از بایت ها است.
- **پیام شی:**
 - بدنه پیام، یک شی واحد Java است.
 - این شی باید رابط قابل سریالیزه شدن را پیاده سازی کند.

عناصر سیستم پیام رسانی

پیام ها

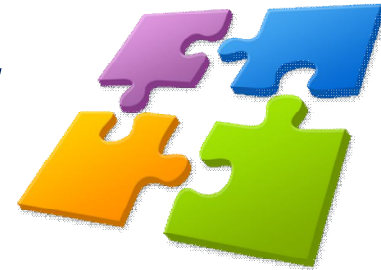
بدنه ...



- انواع پیام هایی که توسط JMS پشتیبانی میشود:
- پیام جریان:
- بدنه جریانی از متغیرهای ابتدایی جاوا است مانند Char, int و long
- پیام نقشه:
- بدنه مانند Java.util.Map رفتار میکند که در آن کلیدها اشیای String هستند.

عناصر سیستم پیام رسانی

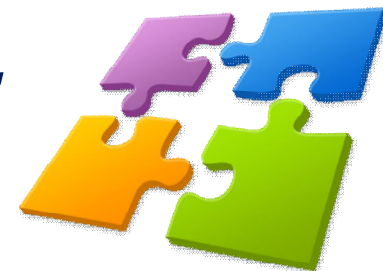
نقاط پایانی



- یک سیستم پیام رسانی مانند سرور عمل میکند و برنامه های فرستنده و گیرنده به عنوان کلاینت برای سیستم پیام رسانی عمل میکند.
- سیستم پیام رسانی، یک رابط برنامه نویسی کاربردی کلاینت را فراهم میکند که کلاینت ها با استفاده از این رابط می توانند با سیستم پیام رسانی در تعامل باشند.
- این رابط ویژه برنامه خاصی نیست، بنابراین هر کلاینت باید شامل مجموعه کدی باشد که از این رابط برنامه نویسی کاربردی برای اتصال به سیستم پیام رسانی به منظور تبادل پیام با سایر برنامه های کاربردی استفاده کند. پس:
- "نقطه پایانی پیام، شامل مجموعه ای کد است که برای اتصال به سیستم پیام رسانی و ارسال یا دریافت پیام استفاده میشود."

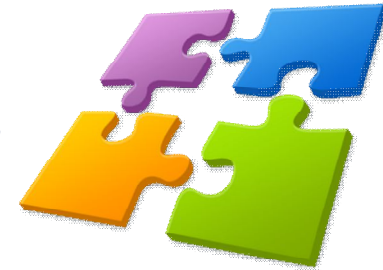
عناصر سیستم پیام رسانی

نقاط پایانی ...



- برنامه کاربردی زمانی از این بخش کد استفاده میکند که نیاز به ارسال یا دریافت پیام دارد.
- نقطه پایانی می تواند تنها برای ارسال یا دریافت پیام مورد استفاده قرار گیرد ولی نه برای هر دو.
- انواع نقاط پایانی:
 - زیرنوع اول که برای ارسال پیام در مدل نقطه به نقطه استفاده میشود.
 - زیرنوع دوم که برای ارسال پیام در مدل انتشار و اشتراک استفاده میشود.

مزایا و معایب پیام رسانی



• مزایا:

- مقیاس پذیرترین روش برای تسهیم داده و تسهیم عملکرد (علت این مقیاس پذیری: ماهیت ناهمگام)
- قابلیت اعتماد (به دلیل پایداری داده مورد تبادل در هر دو سمت شبکه)

• معایب:

- نرم افزار پیام رسانی ناهمگام، از نظر مالی گران تر از میان افزار مبتنی بر ORB است.
- منحنی یادگیری به محیط پیام رسانی ناهمگام وابسته است.
- هزینه شبیه سازی تعامل همگام بین دو برنامه کاربردی بالا است.



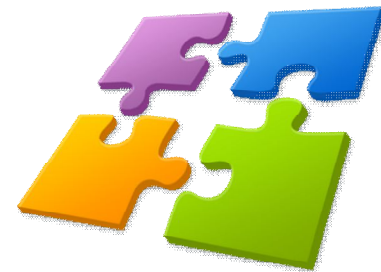
یکپارچه سازی کاربردهای سازمانی

فصل ششم: مروری بر وب سرویس ها

بتول لکزائی

b_lakzaei@yahoo.com

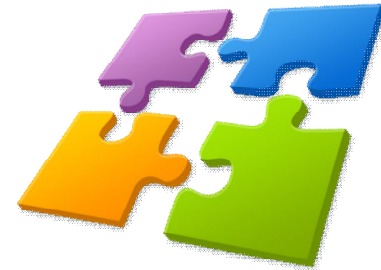
مروری بر فصلهای ۲ تا ۵



✓ فصل ۲: سه روش تسهیم داده:

- تسهیم داده مبتنی بر فایل
- تسهیم داده با استفاده از پایگاه داده مشترک
- تسهیم داده با استفاده از برنامه نویسی سوکت
- مناسب تسهیم بالادرنگ
- سوکت ها برای تسهیم عملکرد نیز بنیادی هستند.
- سوکت ها، صرف نظر از رویکرد یکپارچه سازی، همیشه در پس زمینه حضور دارند.

مروری بر فصلهای ۲ تا ۵ ...



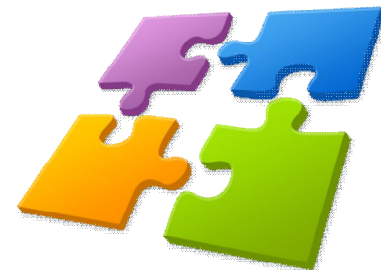
✓ فصل ۳: تسهیم عملکرد: فراخوانی رویه راه دور (RPC):

- برخی ویژگی ها و ایده های حیاتی را معرفی کرد و برای اولین بار، مراحل اصلی و ضروری تسهیم عملکرد بین برنامه های کاربردی را مطرح کرد.

✓ فصل ۴: تسهیم عملکرد: اشیای توزیع شده:

- با معرفی رابط درخواست شی (ORB)، مساله کمبود استفاده مجدد از کد در روش RPC را حل کرد.
- با معرفی زبان تعریف رابط (IDL)، مفهوم عدم وابستگی به زبان را معرفی کرد.

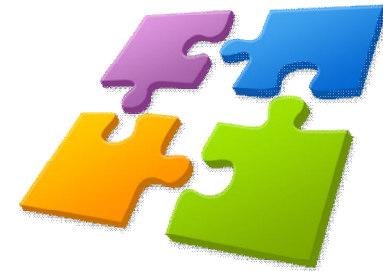
مروری بر فصلهای ۲ تا ۵ ...



✓ فصل ۵: تسهیم عملکرد: پیام رسانی ناهمگام:

- دو مشکل اصلی در روشهای RPC و اشیای توزیع شده، را حل کرد:
- مساله مقیاس پذیری
- عدم قابلیت اطمینان

مساله عدم تجانس



معرفی این فناوری های جدید و مختلف برای یکپارچه سازی برنامه های کاربردی، منجر به مساله عدم تجانس در سازمان های بزرگ میشود. زیرا در یک سازمان بزرگ یا یک سیستم بین سازمانی شامل یک سازمان و شرکای آن، برای یکپارچه سازی برنامه های کاربردی عموماً بیش از یک فناوری استفاده میشود. بنابراین تحمیل استانداردهای سازمانی در این زمینه واقعا غیرممکن است.

انواع عدم تجانس در یک سازمان:

تنوع فرمت های داده

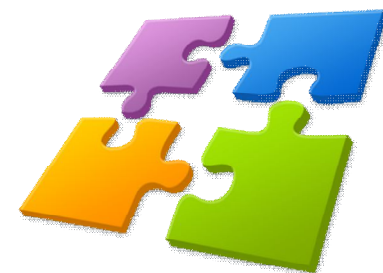
عدم تجانس میان افزار

عدم تجانس همگامی

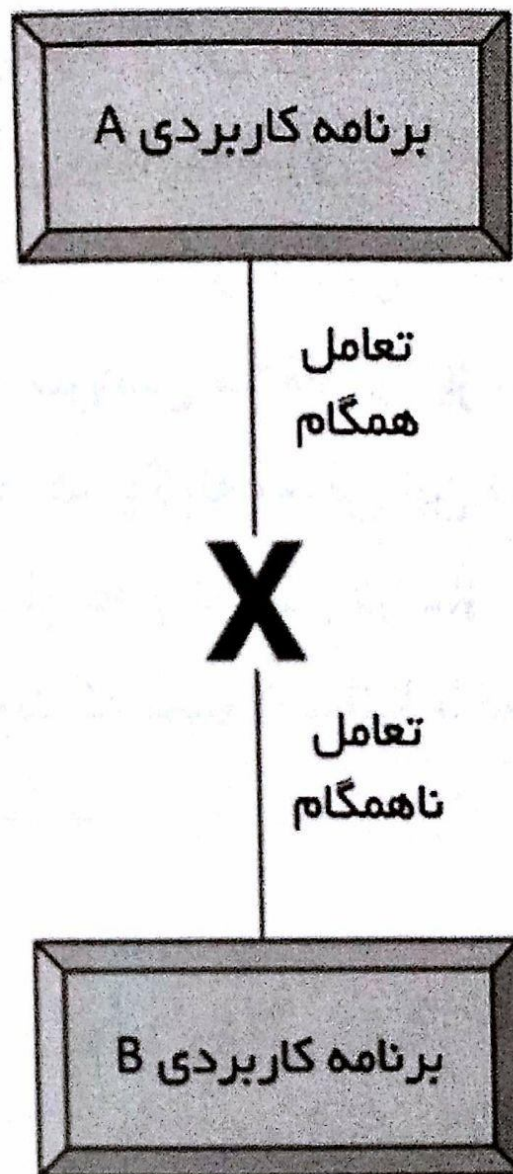
عدم تطابق نسخه ها

تنوع معرفی رابط ها

عدم تجانس پروتکل



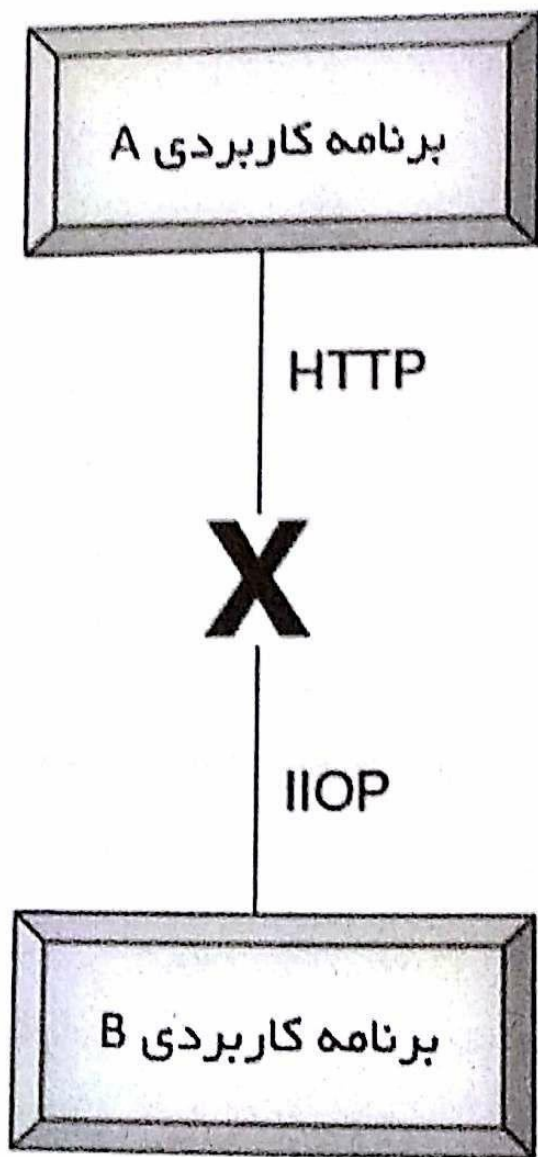
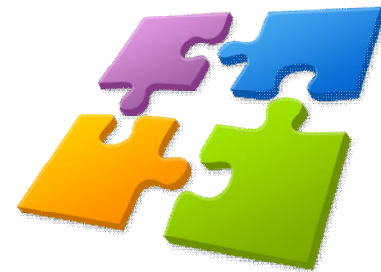
مساله عدم تجانس ...



۱. عدم تجانس همگامی:

- تقریباً همیشه نیاز به پشتیبانی تعاملات همگام و ناهمگام بین برنامه های کاربردی وجود دارد.
- در بسیاری از مواقع، سبک های تعامل مورد استفاده در دو برنامه کاربردی مختلف، با هم مطابقت ندارد. بنابراین این دو برنامه نمیتوانند با یکدیگر تعامل برقرار کنند.

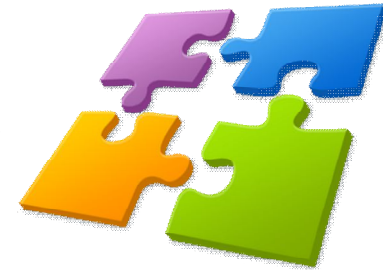
مساله عدم تجانس ...



۲. عدم تجانس پروتکل:

- برای دسترسی به سرویسهای برنامه های کاربردی مختلف، از پروتکل های انتقال مختلف و گاهی ناسازگار استفاده میشود.
- به عنوان مثال برنامه A می خواهد از طریق پروتکل HTTP با برنامه B ارتباط برقرار کند. اما پروتکل مناسب برای برنامه B، پروتکل IIOP است.
- در چنین مواردی تبدیل پروتکل نیاز است.

مساله عدم تجانس ...



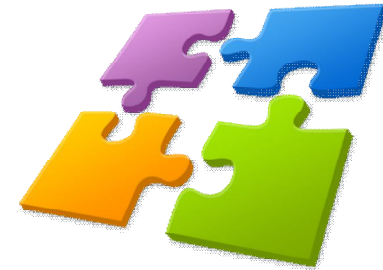
۳. عدم تجانس میان افزار:

- در یک سازمان بزرگ، معمولاً بیش از یک نوع میان افزار استفاده میشود.
- متداولترین نوع میان افزارها، سرورهای برنامه کاربردی و میان افزار پیام گرا (MOM) هستند.

۴. تنوع معرفی رابط ها:

- هنگامی که روشهای معرفی رابط های سرویس و استفاده از آنها برای احضار سرویس ها متفاوت باشند، این مساله پیش می آید.
- به عنوان مثال، روش معرفی رابطها در CORBA و Java RMI با هم متفاوت است.

مساله عدم تجانس ...



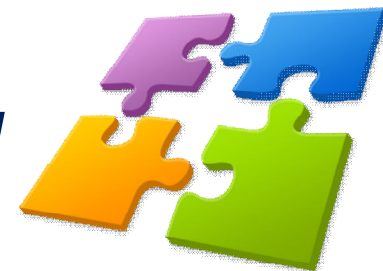
۵. تنوع فرمت های داده:

- این مساله هنگامی به وجود می آید که در فرمت داده مورد تبادل، تنوع وجود داشته باشد.
- در اکثر مواقع، داده به میان افزار مورد استفاده بستگی دارد.
- اگر دو برنامه کاربردی از فرمت های مختلف داده پشتیبانی کنند نیز ممکن است مساله تنوع فرمت های داده، پیش آید.

۶. عدم تطابق نسخه ها:

- به محض اینکه یک نسخه جدید از برنامه سرور ارائه میشود، باید برنامه های کلاینت اصلاح شوند.
- برای حل این مساله باید متدهایی یافت شوند که به سرویسها اجازه دهند بدون قطع نسخه های قبلی برنامه های کلاینت؛ توسعه پیدا کنند (به عنوان مثال با افزودن پارامترهای بیشتر)

استانداردهای وب سرویس

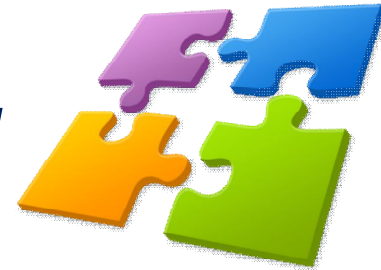


- با توسعه استانداردها و توسعه بیشتر فناوری، تا حدودی به این تنوع و توسعه پذیری سرویسها، رسیدگی شده است.

• استانداردها:

- مجموعه ای از مشخصات، قوانین و رهنمودها که توسط شرکای پیشگام بازار، تنظیم و پذیرفته شده اند و به جزئیات پیاده سازی وابسته نیستند.
- استانداردها، مبنایی برای تشابه ایجاد می کنند و با تعامل پذیری چندمحیطی، پذیرش گسترده ای را امکان پذیر میکنند.

استانداردهای وب سرویس ...

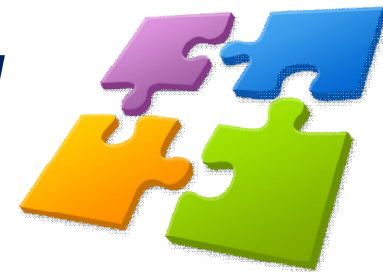


- مثالهایی از این استانداردها:

- XML: زبان ارتباطات مشترک
- SOAP: فرمت مشترک برای تبادل پیام ها
- WSDL: فرمت مشخصات سرویس مشترک
- UDDI: ابزار مشترک برای جستجوی سرویس

- این استانداردها به عنوان وب سرویس شناخته میشوند.

استانداردهای وب سرویس ... XML

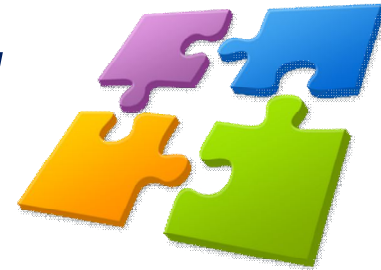


• XML یک فرمت استاندارد و محبوب مستقل از میان افزار برای تبادل داده و اسناد است.

• مزایا و ویژگی های XML:

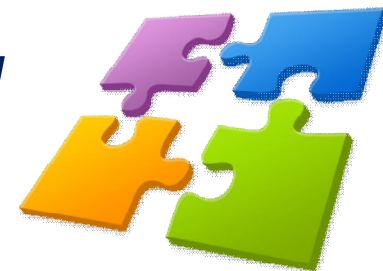
- به هیچ فناوری خاص یا استاندارد میان افزاری محدود نمیشود و امروزه بیشتر به عنوان فرمت موردی (Ad-hoc Format) برای پردازش داده در پلت فرم های مختلف و بسیار ناسازگار میان افزار استفاده می شود.
- رایگان است و با تعداد زیادی از ابزارها روی بسیاری از پلت فرم های مختلف قابل استفاده است.
- ساختار داده را هنگام انتقال حفظ میکند.
- بسیار انعطاف پذیر است و این انعطاف پذیری آن را به عنوان مناسب ترین استاندارد برای حل مسائل عدم تجانس میان افزار و برنامه های کاربردی قرار میدهد.
- مسله فرمت داده را حل میکند.

استانداردهای وب سرویس ... XML



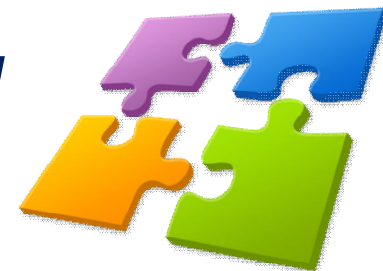
- XML مهم ترین استاندارد است که وب سرویس ها بر اساس آن ساخته میشوند.
- اسناد XML به عنوان ابزاری برای انتقال اطلاعات بین ارائه دهنده سرویس و مصرف کننده سرویس استفاده میشود.
- XML مبنایی است برای سایر استانداردهای وب سرویس از جمله SOAP، WSDL و UDDI

استانداردهای وب سرویس ... SOAP



- XML به تنهایی برای برقراری صحیح ارتباطات بین سرور و کلاینت کافی نیست.
- برای ارتباطات اثربخش طرفین باید بتوانند طبق یک فرمت توافقی پیام ها را مبادله کنند.
- SOAP چنین پروتکلی است که یک فرمت پیام مشترک را برای سرویس ها ارائه میدهد.
- SOAP، یک فرمت پیام رسانی متنی است از فرمت رمزگذاری داده مبتنی بر XML استفاده می کند.
- SOAP = پروتکل ساده دسترسی به شی

استانداردهای وب سرویس SOAP

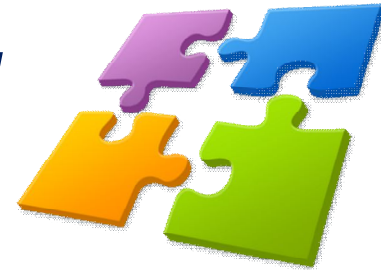


• ویژگی های SOAP:

- مستقل از زبان برنامه نویسی و پلت فرم عملیاتی است و به هیچ فناوری خاصی در نقاط انتهایی نیاز ندارد.
- فرمت متنی آن به گونه ای است که آن را تبدیل به یک پروتکل موافق با دیواره آتش می کند.
- این پروتکل در ابتدا فقط برای کار با HTTP طراحی شد.

- پیام SOAP یک سند XML است که شامل عنصر بدنه و هدر اختیاری است.
- عنصر بدنه پیام واقعی را حمل میکند که توسط گیرنده استفاده میشود.
- عنصر هدر جهت ویژگی های پیشرفته برای پردازشگرهای میانی استفاده میشود.

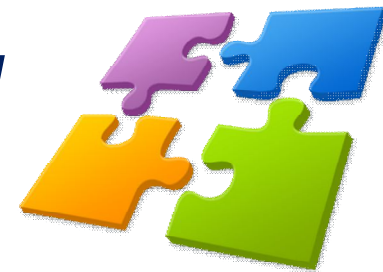
استانداردهای وب سرویس ... WSDL



• WSDL = زبان توصیف وب سرویس ها

• WSDL مبتنی بر XML است و برای تعریف رابط و سایر مشخصات وب سرویس مورد استفاده قرار میگیرد.

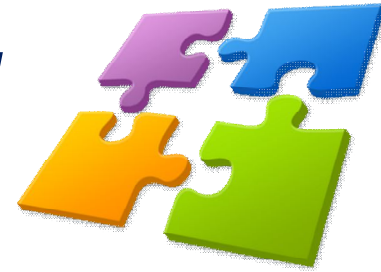
استانداردهای وب سرویس WSDL



• مزایا و ویژگی های WSDL

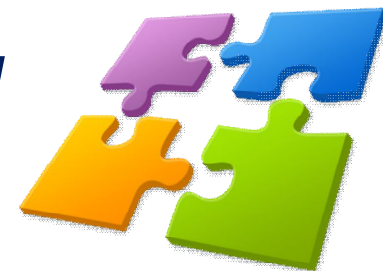
- مستقل از زبان برنامه نویسی و فناوری های میان افزار است. زیرا مبتنی بر XML است. در نتیجه برای توصیف هر نوع سرویسی مناسب است.
- روش تعیین پروتکل ارتباطات را برای احضار سرویس ارائه میدهد. بنابراین سرویس آزاد است که هر پروتکلی را که راحت میتواند پیاده سازی کند، برگزیند.
- روشی برای تعیین فرمت پیام برای برقراری ارتباط با یک سرویس معین را ارائه میدهد. بنابراین سرویس در انتخاب فرمت پیام، آزاد است.
- وسعت عمل زیادی به ارائه دهنده سرویس جهت تعیین نوع عملیات سرویسی عرضه شده، ارائه میدهد. (عملیات همگام / ناهمگام)
- متدی دارد برای تعیین نقطه نهایی سرویس.
- نقطه انتهایی سرویس آدرس شبکه ای است که سرویس برای احضار در آن قرار دارد.

استانداردهای وب سرویس ... رجیستری UDDI



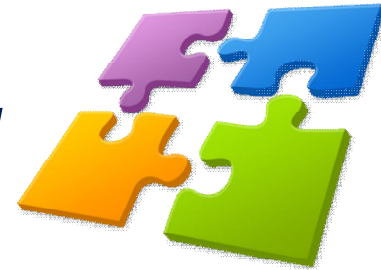
- در یک سازمان بزرگ ممکن است منابع توسعه دهنده از لحاظ جغرافیایی پراکنده باشند و سرور و کلاینت دور از هم قرار گرفته باشند.
- یک سازمان بزرگ به یک مکان مرکزی نیاز دارد که در آن ارائه دهنده سرویس بتواند سرویس های خود را با استفاده از زبان توصیف وب سرویس ها، منتشر کند و مصرف کنندگان سرویس بتوانند سرویس های موجود را بیابند. چنین مکان مرکزی، رجیستری نامیده میشود.
- رجیستری شبیه فهرست برگه های کتابخانه است که برای ثبت ورود کتابهای جدید و سایر رسانه ها و همچنین جست و جوی آنها مورد استفاده قرار میگیرد.

استانداردهای وب سرویس ... رجیستری UDDI ...



- UDDI = مشخصه توسف، جستجو و یکپارچه سازی فراگیر
- UDDI روشی استاندارد برای ثبت، حذف و یکپارچه سازی سرویس ها ارائه میدهد.
- ابتدا ارائه دهنده سرویس یک سرویس را با استفاده از رجیستری UDDI ثبت میکند.
- مصرف کننده سرویس، سرویس مورد نظر را در UDDI جست و جو میکند و پس از یافتن سرویس مورد نظر، برای استفاده از این سرویس مستقیما به ارائه دهنده سرویس متصل میشود.

استانداردهای وب سرویس ...



- برخی از مسائل عدم تجانس که استانداردهای وب سرویس نمیتوانند آنها را حل کنند:
 - عدم تطابق پروتکل
 - عدم تطابق فرمت پیام



یکپارچه سازی کاربردهای سازمانی

فصل هفتم: گذرگاه سرویس سازمان

بتول لکزائی

b_lakzaei@yahoo.com

گذرگاه سرویس سازمان (ESB)



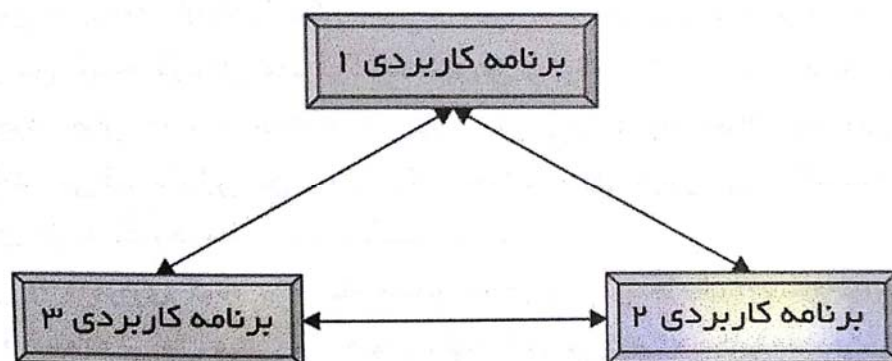
- گذرگاه سرویس سازمان، روشی جامع و مقیاس پذیر است برای اتصال تعداد زیادی برنامه کاربردی، بدون اینکه نیاز باشد هر جفت از این برنامه های کاربردی به طور مستقیم به هم متصل شوند.
- اتصال مستقیم، اتصال نقطه به نقطه نامیده میشود.
- حتی در وب سرویس ها، اتصال بین برنامه ها به صورت اتصال نقطه به نقطه است.
- رویکرد اتصال نقطه به نقطه، مقیاس پذیری خوبی ندارد، زیرا تعداد برنامه های کاربردی درگیر، افزایش می یابد.
- استانداردهای وب سرویس به عنوان راه حلی برای مساله عدم تجانس مطرح شد.
- این استانداردها دو مساله تبدیل پروتکل و تطبیق داده را پوشش نمی دهند.

نقش گذرگاه سرویس سازمان در مسیریابی و اتصال مقیاس پذیر



- وب سرویس ها: به عنوان راه حلی برای مساله عدم تجانس
- اتصال در وب سرویس ها به صورت نقطه به نقطه است، بنابراین وب سرویسها برای یکپارچه سازی یک سازمان بزرگ با تعداد زیادی برنامه کاربردی مناسب نیست.
- تعداد اتصالات مورد نیاز در رویکرد نقطه به نقطه: $N(N-1)/2$

- مثال ۱: یکپارچه سازی سازمانی با ۳ برنامه کاربردی - اتصال نقطه به نقطه
- تعداد اتصالات مورد نیاز: ۳

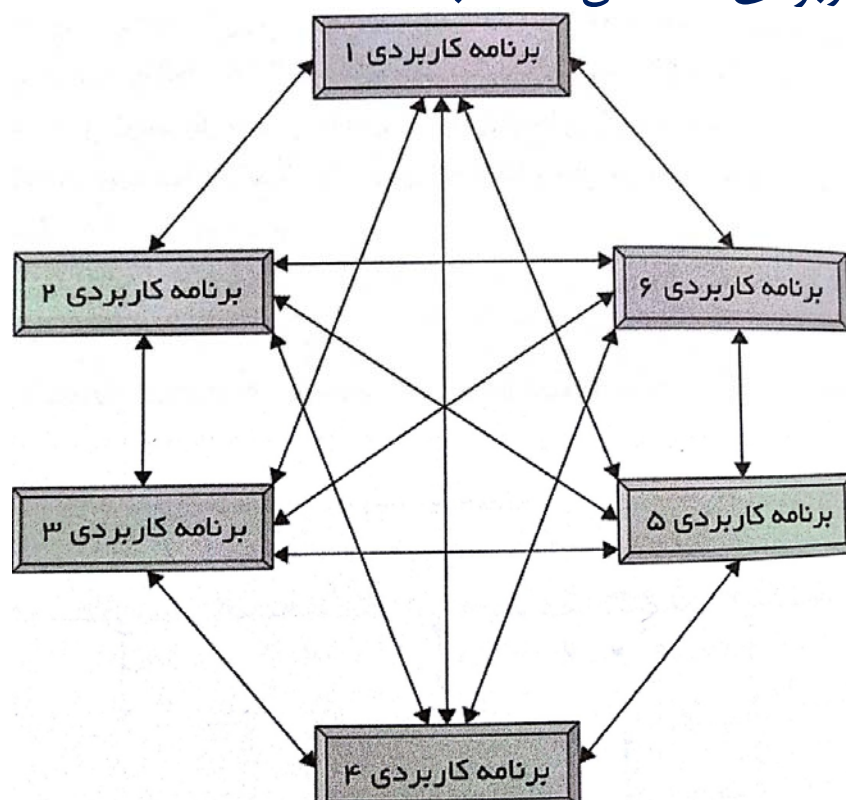




نقش گذرگاه سرویس سازمان در مسیریابی و اتصال مقیاس پذیر ...

• مثال ۲: یکپارچه سازی سازمانی با ۶ برنامه کاربردی - اتصال نقطه به نقطه

• تعداد اتصالات مورد نیاز: ۱۵

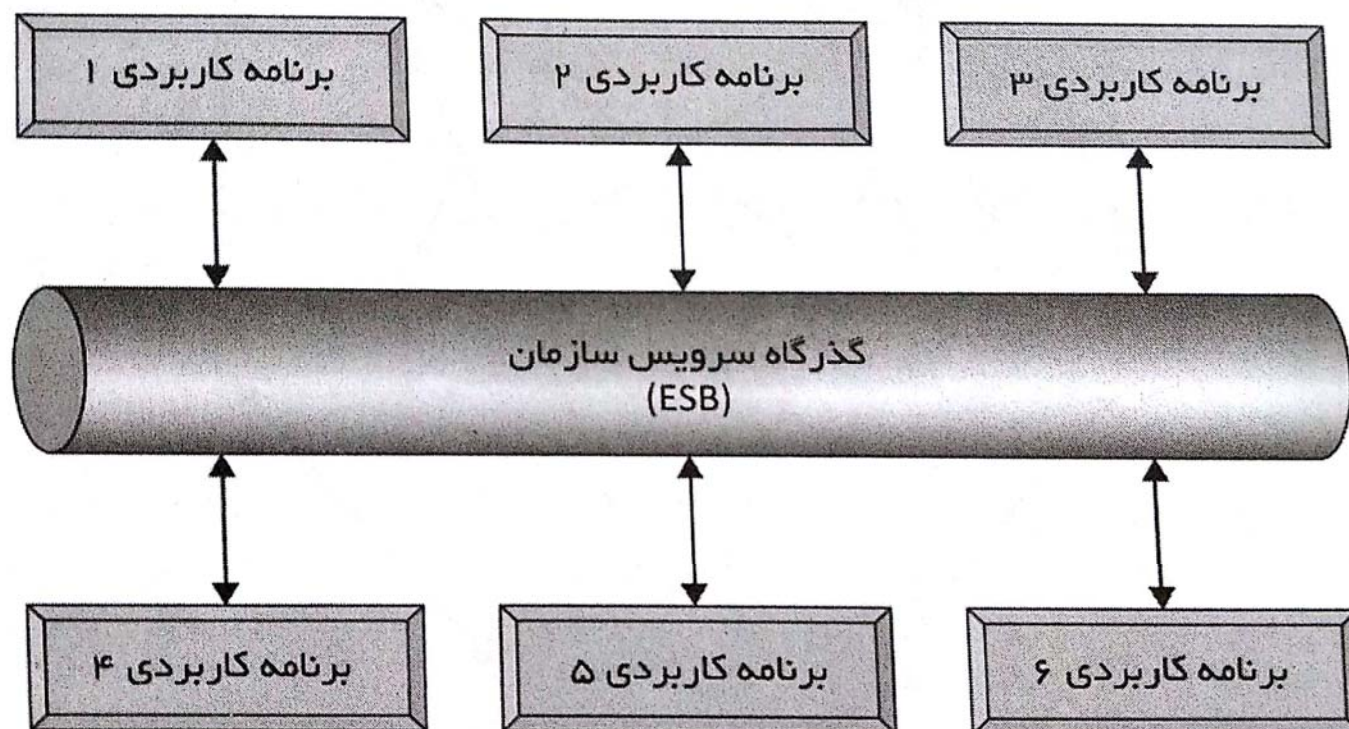


نقش گذرگاه سرویس سازمان در مسیریابی و اتصال مقیاس پذیر ...



• مثال ۳: یکپارچه سازی سازمانی با ۶ برنامه کاربردی - استفاده از گذرگاه سرویس سازمان

• تعداد اتصالات = ۶



نقش گذرگاه سرویس سازمان در مسیر یابی و اتصال مقیاس پذیر ...



- گذرگاه سرویس سازمان، راه حلی عالی برای مساله مقیاس پذیری یکپارچه سازی ارائه میدهد. و برای سازمان های متوسط تا بزرگ که نیاز به یکپارچه سازی تعداد زیادی برنامه کاربردی دارند، مناسب است.
- در گذرگاه سرویس سازمان، برنامه ها به طور مستقیم با یکدیگر تعامل ندارند، بلکه برنامه ها به گذرگاه متصل می شوند و این گذرگاه ابزاری را برای اتصالات بین برنامه های کاربردی فراهم میکند.

نقش گذرگاه سرویس سازمان در مسیر یابی و اتصال مقیاس پذیر ...



مزایای گذرگاه سرویس سازمان:

۱. کاهش تعداد اتصالات مورد نیاز:

- تعداد اتصالات مورد نیاز برای یکپارچه سازی N برنامه کاربردی: N

۲. سهولت نگهداری و ارتقا:

- اضافه کردن و حذف کردن برنامه های کاربردی از ساختار یکپارچه سازی، آسان است.
- برای افزودن یک برنامه جدید، تنها یک اتصال کافی است و لازم نیست هیچ تغییری در هیچ یک از برنامه های کاربردی موجود ایجاد شود.
- در رویکرد نقطه به نقطه، افزودن یک برنامه جدید، نیازمند ایجاد اتصال به هر یک از برنامه های موجود است که ممکن است منجر به تغییر در کلیه برنامه های موجود شود.

نقش گذرگاه سرویس سازمان در مسیر یابی و اتصال مقیاس پذیر ...



مزایای گذرگاه سرویس سازمان:

۳. چابکی بیشتر

- در صورتی که ارائه دهنده اولیه سرویس، به دلایلی دیگر، قادر به عرضه سرویس نباشد، گذرگاه سرویس، امکان جایگزینی آسان ارائه دهنده سرویس با یک ارائه دهنده معادل را امکانپذیر می سازد.
- دلیل این توانایی این است که با ساختار گذرگاه سرویس نیازی به کدنویسی آدرس شبکه نقطه انتهایی سرویس به برنامه کاربردی کلاینت وجود ندارد. زیرا این گذرگاه امکان جستجوی آدرس نقطه انتهایی سرویس را بر اساس محتوا و مفهوم درخواست سرویسی که از برنامه کاربردی کلاینت دریافت میکند، فراهم می آورد.
- این قابلیت یکی از سه قابلیت کلیدی گذرگاه سرویس است و به این معنا است که کلاینت نمیداند سرور کیست و سرور از هویت کلاینت بی اطلاع است.

انواع گذرگاه سرویس سازمان



گذرگاه سرویس سازمان
بر اساس سیستم پیام
دهی ناهمگام

گذرگاه سرویس سازمان
بر اساس فراخوانی رویه
راه دور

انواع گذرگاه سرویس سازمان ...



- گذرگاه سرویس سازمان بر اساس فراخوانی رویه راه دور:
- مزایا:
 - راه اندازی آسان
 - نسبتاً ارزان
- معایب:
 - مقیاس پذیری پایین در صورت اهمیت میزان تراکنش ها
 - عدم توانایی برای مدیریت مجموعه متنوعی از برنامه های کاربردی
- این نوع، هنگامی استفاده میشود که حجم کمتری از تراکنشها پیش بینی میشود.
- این نوع، معمولاً فقط برای کار با وب سرویسها، XML، و Java RMI طراحی شده است.

انواع گذرگاه سرویس سازمان ...

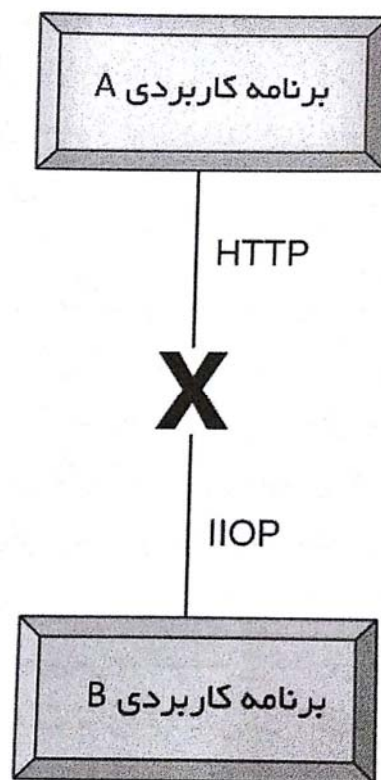


- گذرگاه سرویس سازمان بر اساس سیستم پیام دهی ناهمگام
- معایب (نسبت به نوع اول):
 - راه اندازی پیچیده
 - نسبتا گران
- مزایا (نسبت به نوع اول):
 - ارائه راه حلی مقیاس پذیر بر اساس حجم تراکنش ها (بنابراین میتواند از تراکنشهای بیشتری پشتیبانی کند).
 - قابلیت استفاده برای مجموعه متنوعی از برنامه های کاربردی
 - تضمین تحویل پیام

نقش گذرگاه سرویس سازمان در تبدیل پروتکل



- یکی از مسائل عدم تجانس، مساله عدم تطابق پروتکل ارتباطی بین برنامه های کاربردی است.



نقش گذرگاه سرویس سازمان در تبدیل پروتکل ...



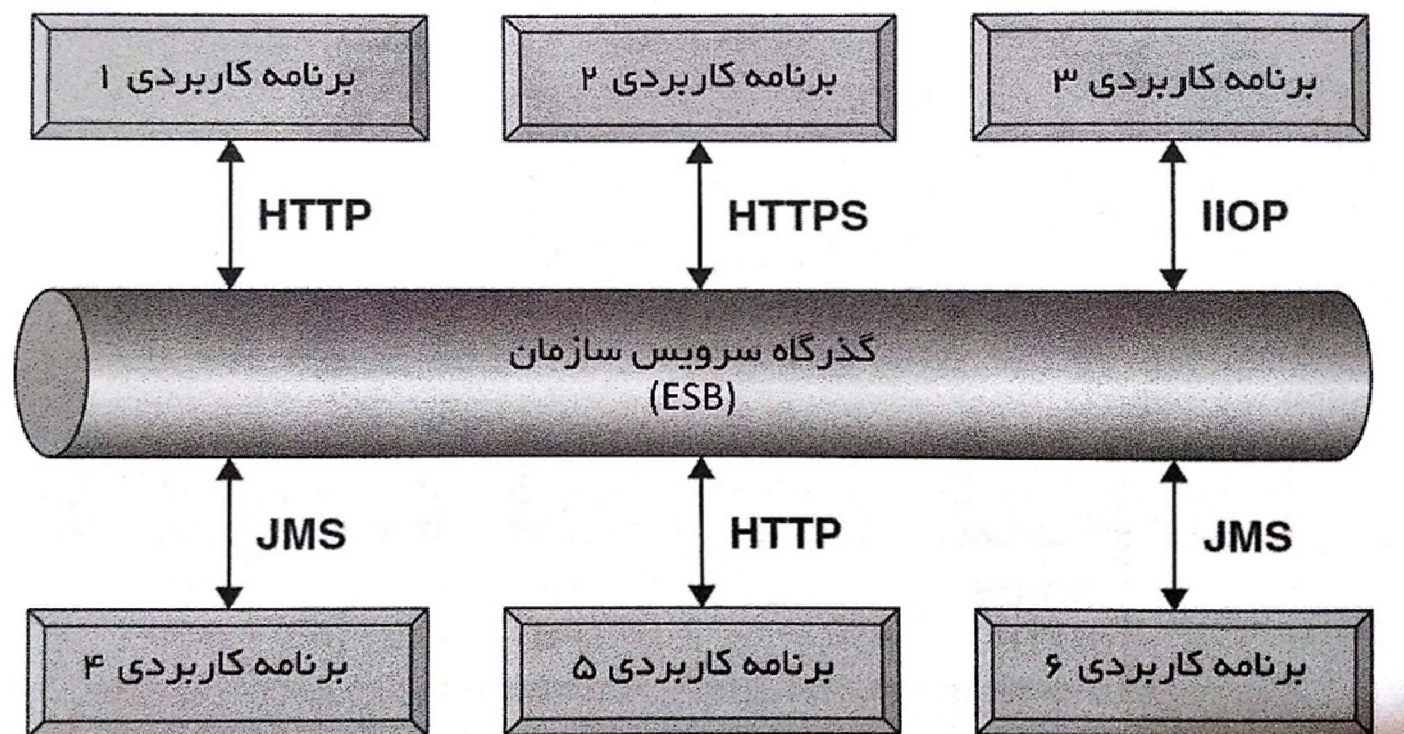
• راه حل ایده آل:

- استانداردسازی پروتکل ارتباطات، به گونه ای که همه برنامه های کاربردی سازمان از یک پروتکل ارتباطی استفاده کنند.
- در دنیای واقعی پیاده سازی چنین استandاری دشوار و حتی غیر ممکن است به دلیل:
 - کمبود منابع توسعه دهنده و محدودیت های زمانی
 - کیفیت و امنیت نیازمندی های سرویس میتواند دچار مشکل شود.

• راه حل کاربردی:

- استفاده از گذرگاه سرویس سازمان که امکان تبدیل پروتکل را فراهم میکند.

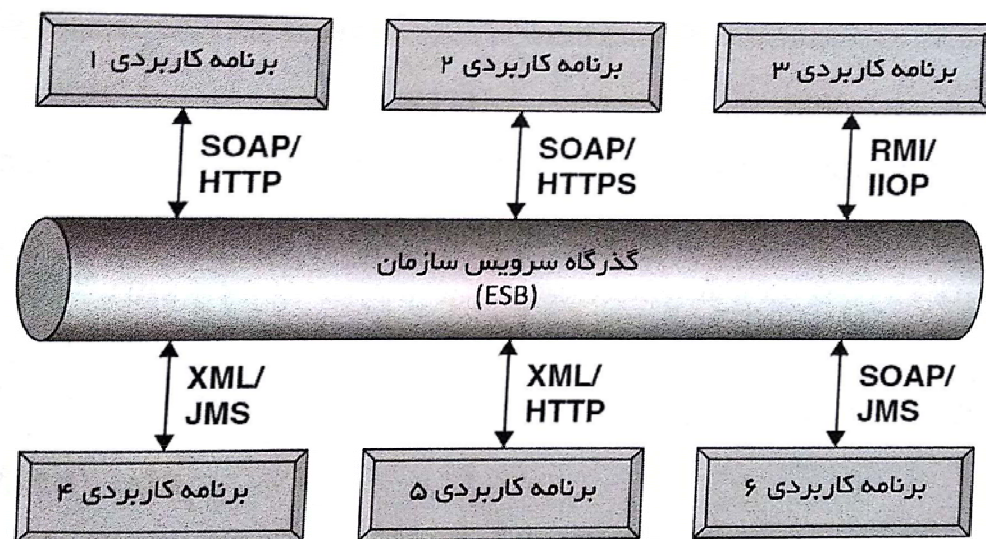
نقش گذرگاه سرویس سازمان در تبدیل پروتکل...



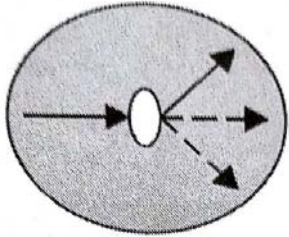
نقش گذرگاه سرویس سازمان در تبدیل داده/پیام



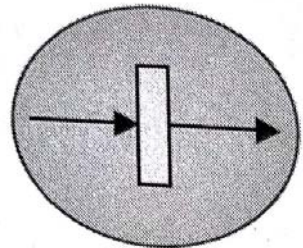
- یکی از مسائل عدم تجانس، مساله عدم تطابق فرمت داده/پیام بین برنامه های کاربردی است.
- راه حل: استفاده از گذرگاه سرویس سازمان که قابلیت تبدیل فرمت های مختلف داده را فراهم میکند.



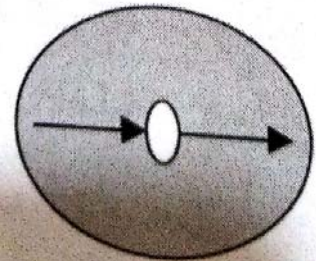
مسیریابی



سوئیچ پروتکل



تبدیل داده



عملکردهای کلیدی گذرگاه سرویس سازمان



سه عملکرد کلیدی EBS:

- مسیریابی مبتنی بر محتوا و مفهوم
- تبدیل یا سوئیچ پروتکل
- تبدیل داده یا پیام



عملکردهای کلیدی گذرگاه سرویس سازمان...

- با ترکیب این سه عملکرد در هسته EBS، این گذرگاه میتواند برخی از مجازی سازی ها را ارائه دهد:

مجازی سازی مکان و هویت

پروتکل تعامل

رابط



عملکردهای کلیدی گذرگاه سرویس سازمان...

- مجازی سازی مکان و هویت:

- لازم نیست برنامه کلاینت، آدرس یا مکان برنامه سرور را بداند، و سرور لازم نیست هویت کلاینت را بداند.
- درخواست سرویس، با هر تعداد ارائه دهنده سرویس می تواند تامین شود. این کار باعث میشود ارائه دهنده سرویس بدون قطع سیستم، از ساختار یکپارچه حذف شود یا به آن اضافه شود، بنابراین سرویسی لاینقطع به مصرف کننده سرویس ارائه می شود.



عملکردهای کلیدی گذرگاه سرویس سازمان...

• پروتکل تعامل:

- لازم نیست سرور و کلاینت، از پروتکل ارتباطات یا سبک تعامل یکسانی استفاده کنند.

• رابط:

- لازم نیست کلاینت با رابط عرضه شده توسط سرور، تطابق کامل داشته باشد.
- گذرگاه سرویس سازمان با تبدیل پیام درخواست به شکل مورد انتظار سرور، این اختلاف را حل می کند.



عملکردهای کلیدی گذرگاه سرویس سازمان...

- این مجازی سازی باعث می شود، گذرگاه سرویس سازمان پیاده سازی شفافیت از سرور به کلاینت هم در زمان توسعه و هم در زمان استقرار ارائه دهد.
- گذرگاه سرویس سازمان این مسئولیت را بر عهده میگیرد که درخواست سرویس را به سرور مناسب تحویل دهد و سرور بدون آگاهی از مبدا درخواست سرویس، به آن پاسخ می دهد.
- گذرگاه سرویس به خودی خود هم برای سرور و هم برای کلاینت شفاف است:
- منطق برنامه کاربردی می تواند بدون اطلاع از اینکه اتصال مستقیم است یا از گذرگاه سرویس استفاده میشود، سرویسی را مصرف کند یا ارائه دهد. بنابراین تصمیم گیری درباره به کارگیری گذرگاه سرویس یا عدم به کار گیری آن در زمان استقرار صورت می گیرد، زیرا هیچ تغییری در کد برنامه لازم نیست.



عملکردهای کلیدی گذرگاه سرویس سازمان...

• انواع تعاملات مورد پشتیبانی توسط گذرگاه سرویس، بین سرور و کلاینت:

• عملیات درخواست و پاسخ همگام

• تعامل ناهمگام

• انتشار و اشتراک

• گذرگاه سرویس سازمان میتواند این تعاملات را تبدیل کند.



عملکردهای کلیدی گذرگاه سرویس سازمان...

- نیازمندی های کیفیت سرویس (QOS)
- نیازمندی های غیر کارکردی که توسط شرکای سرویس مشخص میشوند و گذرگاه سرویس سازمان، سرویسها را برای پیاده سازی این نیازمندی ها ارائه می دهد.

QOS

عملکرد و قابلیت اعتماد

سرویس های امنیتی



عملکردهای کلیدی گذرگاه سرویس سازمان...

عملکرد و قابلیت اعتماد:

- تعیین زمان پاسخ به سرویس
- زمان پاسخ به سرویس از ۵۰ میلیون ثانیه تجاوز نکند.
- ارائه دهنده سرویس ۹۹/۹۹۹ درصد مواقع فعال باشد.

سرویس های امنیتی:

- امنیت به طور کلی مساله ای مهم برای رایانش توزیع شده است. اما هنگامی که سرویس های شخص ثالث توسط سیستم شما مصرف شوند یا هنگامی که سیستم شما به اشخاص ثالث خارجی سرویس ارائه دهد، اهمیت ویژه ای دارند.



عملکردهای کلیدی گذرگاه سرویس سازمان...

سرویس های امنیتی...

- گذرگاه های سرویس سازمان که سرویسهای امنیتی را عرضه می کنند، مستقیماً خودشان امنیت را ارائه نمی دهند. بلکه به سادگی چارچوبی برای برای اتصال نرم افزار امنیت ارائه می دهند و قابلیت هایی برای کمک به هدایت گذرگاه سرویس سازمان در شبکه بدون مسدود شدن توسط فایروال یا هر نوع برنامه امنیتی دیگر، فراهم می کنند.

• Tivoli Suites: مثالی از نرم افزار امنیت که میتواند با گذرگاه سرویس سازمان کار کند.



عملکردهای کلیدی گذرگاه سرویس سازمان...

- برخی سرویس های امنیتی که توسط گذرگاه سرویس سازمان ارائه میشوند:
- رمزگذاری داده برای تضمین حریم خصوصی داده ها
- مجوز دادن به درخواست های سرویس. آیا کاربر سرویس همان کسی است که ادعا میکند؟
- صحت داده ها. آیا داده ها حقیقی است؟
- سرویس ممیزی. ممیزی خودکار تعاملات سرویس به دلایل قراردادی/قانونی یا برای اهداف حسابداری

ویژگی های اختیاری گذرگاه سرویس سازمان

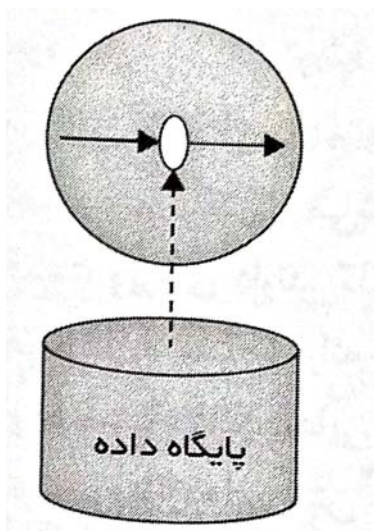


۱. غنی سازی داده:

- گذرگاه سرویس سازمان، ظرفیت پیام دریافت شده از منبع خارجی را تکمیل میکند تا نیازمندی های سرور را تطبیق دهد.

• مثال: دریافت درخواستی که شامل آدرس بدون نام کشور است.

- سرور باید آدرس را طوری کامل کند که نام کشور را در برگیرد تا بتواند درخواست را پردازش کند.
- گذرگاه سرویس میتواند پیام ظرفیت دریافتی را با نام کشوری که از یک منبع داده، یا با یک نام پیش فرض کامل کند.



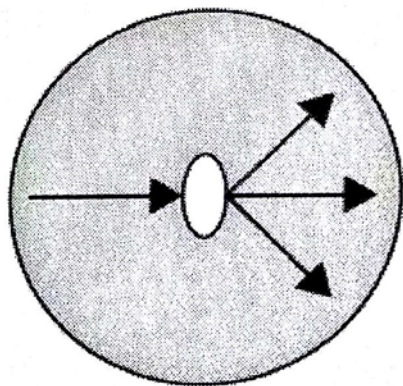


ویژگی های اختیاری گذرگاه سرویس سازمان...

۲. توزیع:

- چند برنامه کاربردی کلاینت میتوانند نوع خاصی از پیام را به اشتراک بگذارند و گذرگاه سرویس سازمان، آن پیام را بین تمام آن برنامه های کاربردی توزیع میکند.

- مثال: درخواست قیمت سهام

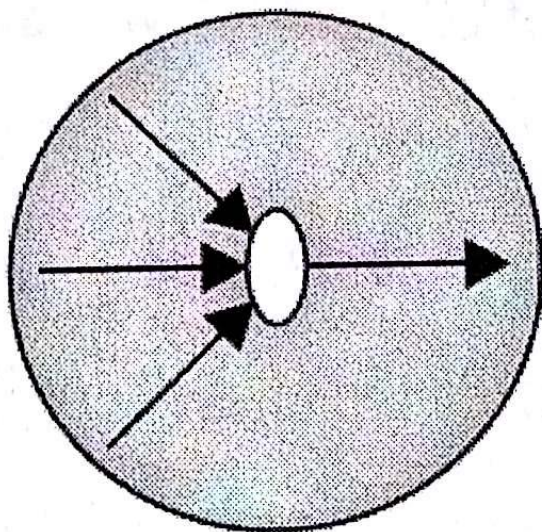




ویژگی های اختیاری گذرگاه سرویس سازمان...

۳. همبستگی:

- امکان استنتاج رویدادی پیچیده از چند پیام یا از جریانی از رویدادها.
- این امکان به قوانین تشخیص الگو و قوانین شکل دهی پیام خروجی پیچیده تکیه دارد.

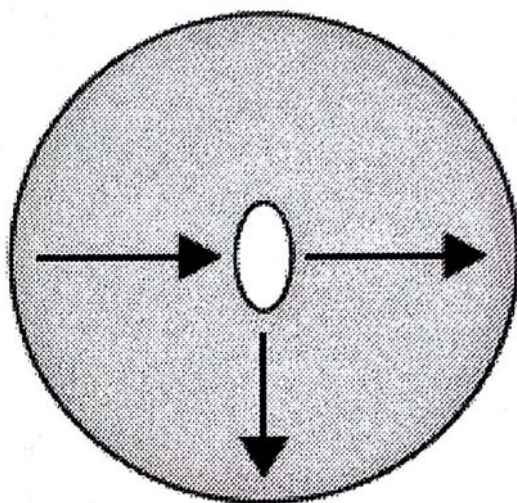




ویژگی های اختیاری گذرگاه سرویس سازمان...

۴. پایش:

- امکانی برای مشاهده پیام ها هنگام عبور از واسط بدون اینکه تغییری در پیام ها ایجاد شود.



• کاربرد:

- سنجش کاربرد برای صورتحساب کاربر بعدی
- ثبت رویدادهای سطح کسب و کار (مانند خرید بیشتر از مبلغی خاص)
- ثبت پیام ها جهت ممیزی یا برای داده کاوی بعدی



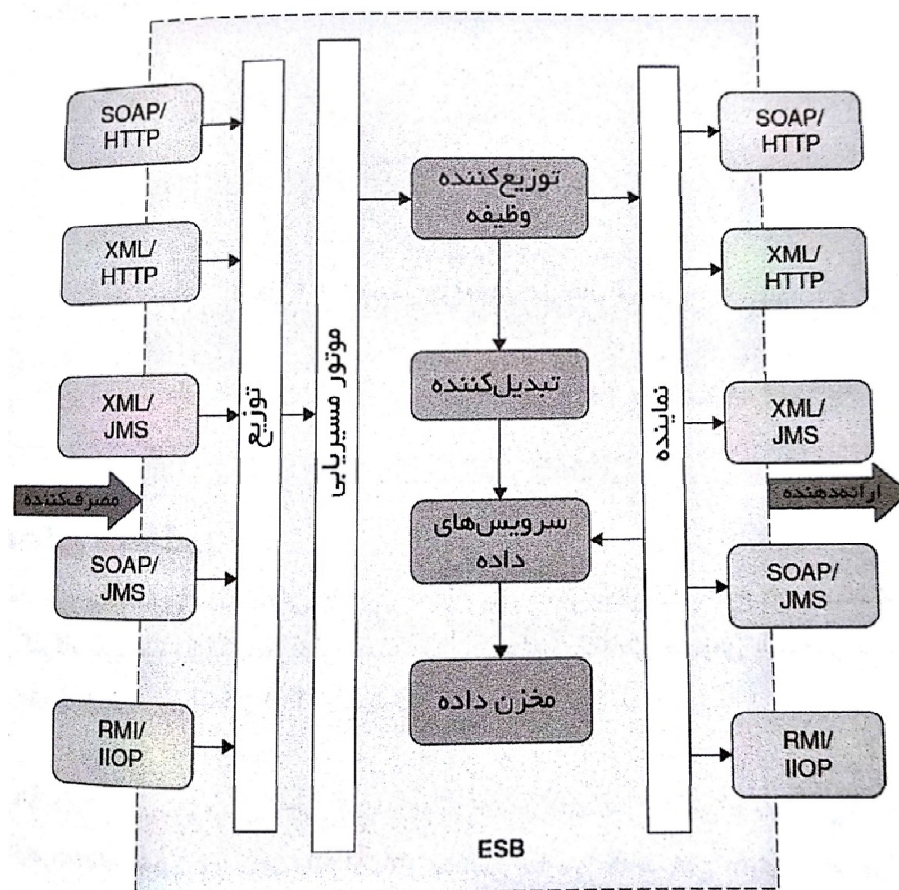
مولفه های منطقی گذرگاه های سازمان



- به منظور پیاده سازی عملکردهای کلیدی و ویژگی های اختیاری گذرگاه سرویس سازمان، تعدادی مولفه در ساختار گذرگاه مورد نیاز است:



مولفه های منطقی گذرگاه سرویس سازمان...





مولفه های منطقی گذرگاه سرویس سازمان...

۱. آداپتورها:

- مهم ترین مولفه گذرگاه سرویس سازمان
- در پیرامون گذرگاه سرویس سازمان قرار میگیرند.
- کلیه درخواست هایی که وارد و خارج میشوند از این آداپتورها استفاده میکنند.
- هر آداپتور نیاز کلاینت یا سرور خاصی را برآورده میکند و فقط ترکیب خاصی از پروتکل ارتباطی و فرمت پیام را می پذیرد.



مولفه های منطقی گذرگاه سرویس سازمان...

۱. آداپتورها ...

- برخی فرمت های رایج:
 - SOAP/HTTP, XML/HTTP, RMI/IIOP, XML/JMS, SOAP/JMS
- آداپتورها با انتزاعی کردن جزئیات پروتکل ها و سازوکارهای نقطه ورودی و خروجی، دیدگاهی یکسان به مولفه های EBS ارائه می دهند. این کار باعث میشود مولفه ها، ورودی را از هر پروتکلی دریافت کنند و خروجی را به هر پروتکلی ارسال کنند.



مولفه های منطقی گذرگاه سرویس سازمان...

۲. توزیع کننده:

- مولفه توزیع کننده به عنوان نقطه ورودی متمرکز عمل می کند.
- مسئول بازیابی ورودی از آداپتورها و سوق دادن آنها به سمت وظایف مسیریابی صحیح، تبدیل و غنی سازی است.
- توزیع کننده درخواست را به مدیر درخواست ارسال میکند و همراه با مدیر درخواست، قابلیت مسیریابی مبتنی بر محتوا را برای EBS فراهم میکند.



مولفه های منطقی گذرگاه سرویس سازمان...

۳. مدیر درخواست

- هر سرویس یک مدیر درخواست ویژه دارد.
- موتور مسیریابی پارامترهای ویژه سرویس را از مدیر درخواست دریافت میکند و سپس مدیر درخواست، درخواست را برای اجرای صحیح وظیفه به موتور مسیریابی تحویل میدهد.

۴. موتور قوانین و مسیریابی

- مسئول اجرای وظایف تبدیل و غنی سازی و مسیریابی به سمت نماینده های سرویس مناسب است.



مولفه های منطقی گذرگاه سرویس سازمان...

۵. نماینده های سرویس

- نماینده ها، مولفه های ویژه نقطه پایانی سرور هستند.
- با استفاده از آداپتورها با سرور ارتباط برقرار میکنند.

۶. موتور تبدیل

- داده / پیام دریافتی را به فرمتی تبدیل می کند که با فرمت مورد نیاز سرور سازگار باشد.
- این مولفه یکی از عناصر کلیدی گذرگاه سرویس سازمان است که یکپارچه سازی بین رابط های سرویس مجزا را تسهیل میکند.



مولفه های منطقی گذرگاه سرویس سازمان...

۷. مولفه غنی سازی

- باعث می شود، EBS ظرفیت پیام را از یک منبع خارجی تکمیل کند تا با نیازمندی های سرور مطابقت داشته باشد.
- این منبع خارجی ممکن است یک پایگاه داده باشد.
- مثال: یک آدرس بدون نام کشور



مولفه های منطقی گذرگاه سرویس سازمان...

۸. مولفه واقعہ نگاری:

- پشتیبانی از واقعہ نگاری مولفه های گذرگاه سرویس سازمان

۹. مولفه مدیریت استثنا:

- مدیریت کلیه استثنای تولید شده توسط مولفه های مختلف گذرگاه سرویس سازمان

پیکربندی های استقرار



- عامل موثر در استقرار گذرگاه های سرویس سازمان:

- اندازه سازمان

- یک سازمان کوچک: یک گذرگاه سرویس سازمان واحد با یک رجیستری واحد ضمیمه شده به آن
- سازمان های بزرگتر: الگوی پیچیده تری شامل چند گذرگاه سرویس سازمان با تعدادی رجیستری

- عامل موثر در تعیین نوع گذرگاه مورد نیاز در هر الگو:

- نیازمندی های اتصال خاص هر سازمان

- به عنوان مثال سازمانی که نیاز به اتصال با شرکای شخص ثالث و فروشندگان دارد، ممکن است نیاز به گذرگاه سرویس سازمانی داشته باشد که به طور ویژه برای مدیریت موقعیت های امنیتی برای اتصال به اشخاص ثالث طراحی شده است. (این گذرگاه ها به عنوان دروازه شناخته میشود.)

پیکربندی های استقرار ...



• پیکربندی های استقرار:

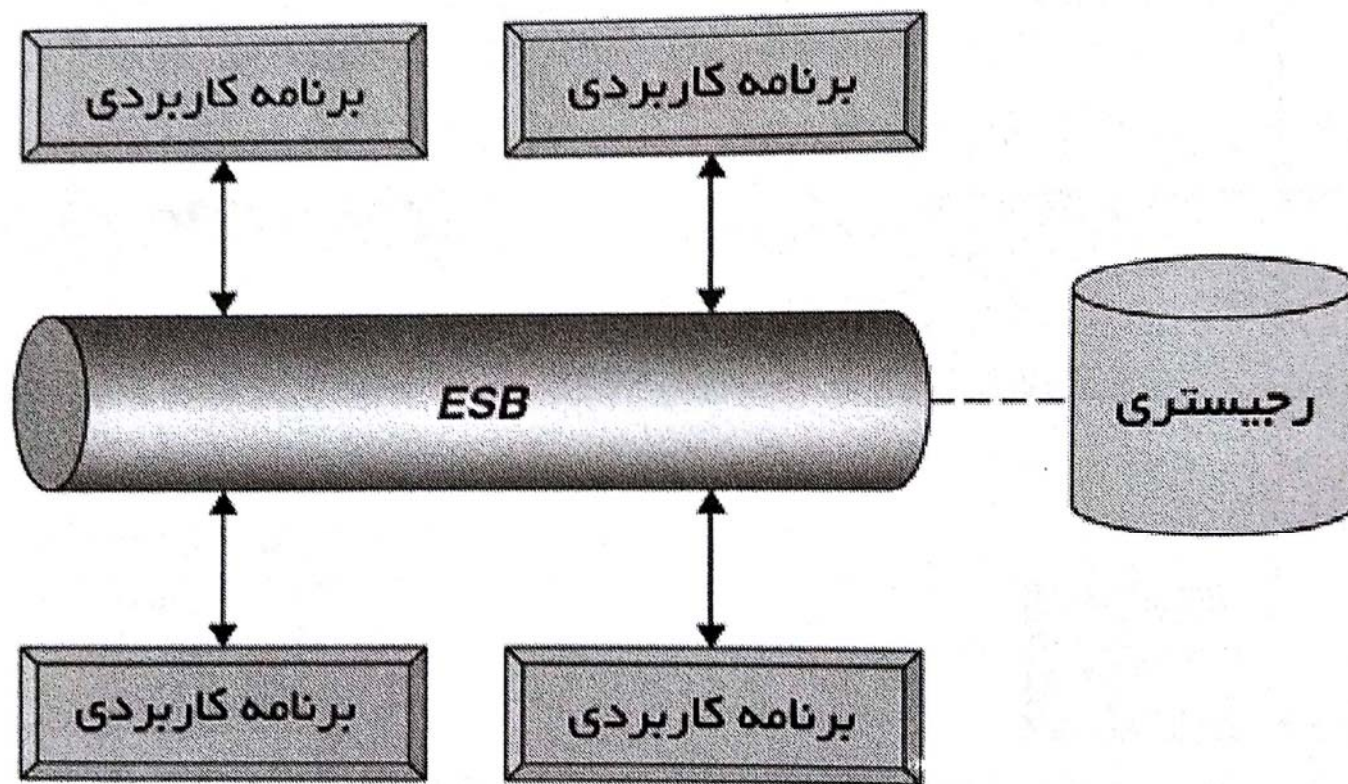
۱. گذرگاه سرویس سازمان سراسری
۲. گذرگاه سرویس سازمان با اتصال مستقیم
۳. گذرگاه سرویس سازمان متحد
۴. گذرگاه سرویس سازمان واسطه

پیکربندی های استقرار گذرگاه سرویس سازمان سراسری



- برای کل سازمان یک گذرگاه سرویس سازمان واحد وجود دارد.
- این گذرگاه یک رجیستری واحد را به کار میگیرد.
- برای سازمان کوچک یا یک خط واحد از کسب و کار مناسب است.
- کلیه سرویس ها در کلیه شرکا قرار دارد.
- نیازی به اتصال با شرکا یا فروشندگان خارجی شخص ثالث نیست، بنابراین نیازمندی های امنیتی حداقل هستند.

پیکربندی های استقرار گذرگاه سرویس سازمان سراسری...



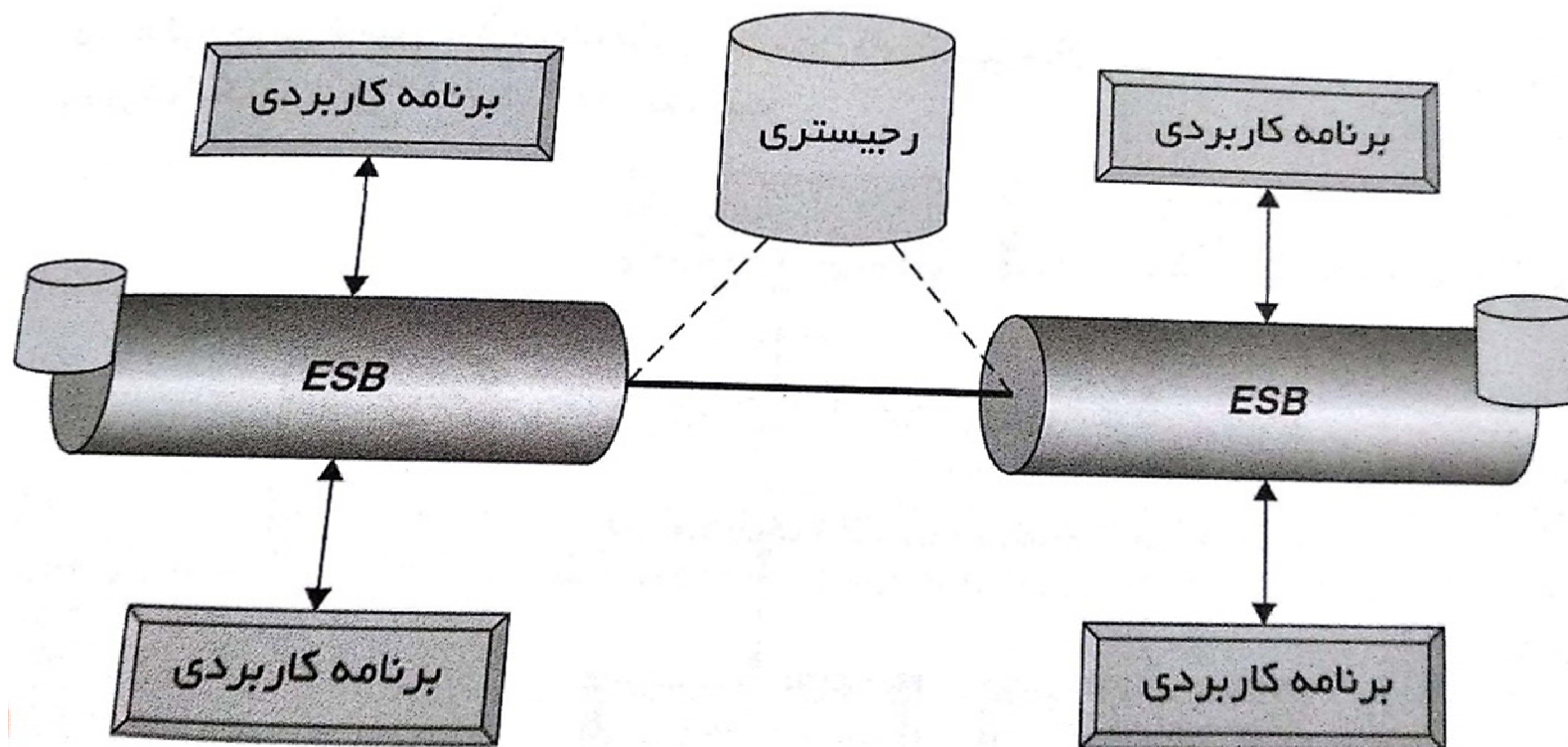


پیکربندی های استقرار گذرگاه سرویس سازمان با اتصال مستقیم

- بیش از یک گذرگاه سرویس سازمان با اتصال مستقیم
- یک رجیستری مشترک برای قابلیت رویت سرویس ها در سراسر سازمان
- به عنوان مثال این الگو میتواند برای اتصال مجموعه ای از برنامه های کاربردی (مانند برنامه های SAP) با بقیه سازمان به کار گرفته میشود.
- چنین برنامه های کاربردی بسته بندی شده معمولا EBS مخصوص به خود را دارند که ماژولهای مختلفی (مانند CRM و ERP) را به یکدیگر متصل میکنند.
- به عنوان مثال ماژولهای SAP توسط EBS مخصوص خود به نام NetWeaver به هم متصل میشوند.



پیکربندی های استقرار گذرگاه سرویس سازمان با اتصال مستقیم ...

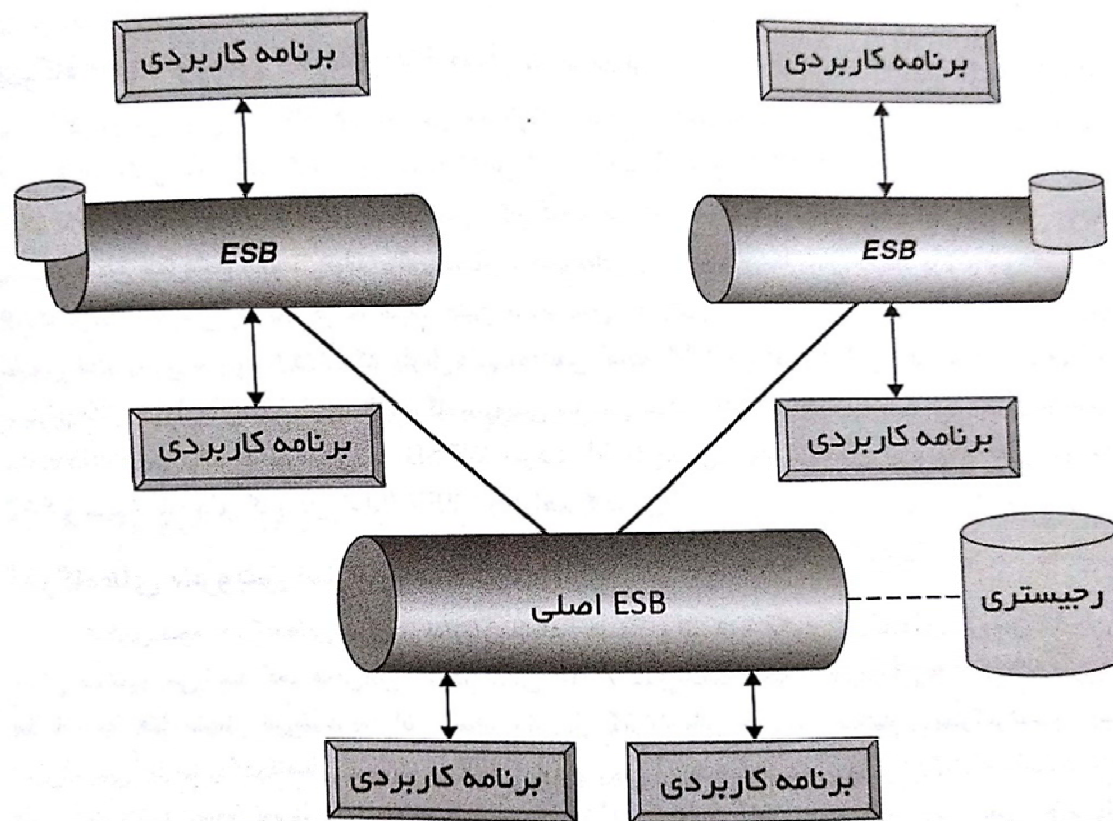


پیکربندی های استقرار گذرگاه سرویس سازمان متحد



- مشابه الگوی گذرگاه سرویس سازمان با اتصال مستقیم
- چند گذرگاه سرویس سازمان به هم متصل میشوند.
- یکی از گذرگاه ها، رابطه اصلی/فرعی با سایر گذرگاه ها دارد و روی آنچه در دسترس سایر مشارکت کنندگان ست کنترل دارد.
- قابل استفاده برای واحدهای نسبتاً خودگردان در سازمان که میخواهند سرویس های خود را با واحدهای دیگر به اشتراک بگذارند.

پیکربندی های استقرار گذرگاه سرویس سازمان متحد ...

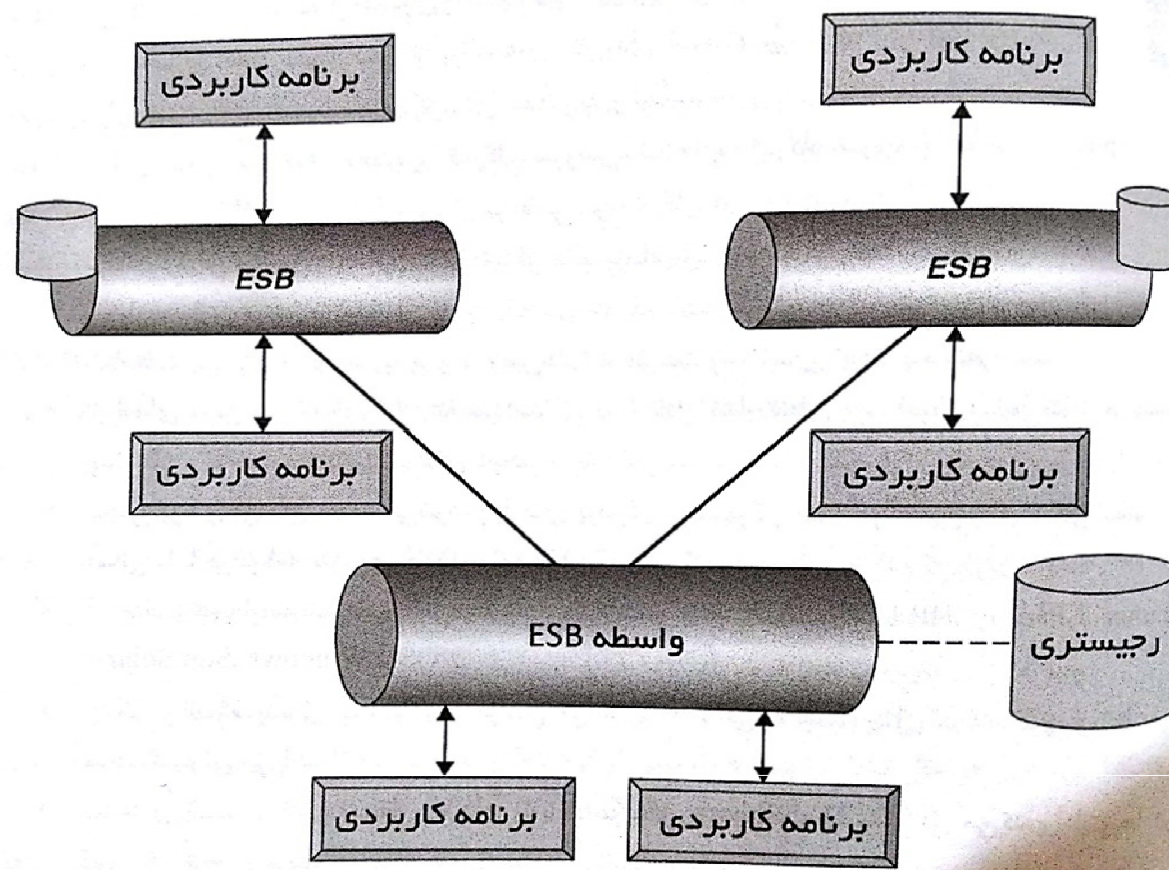


پیکربندی های استقرار گذرگاه سرویس سازمان واسطه



- استفاده از واسط برای وساطت بین گذرگاه های سرویس سازمان متصل بهم
- هر گذرگاه، رجیستری خود را دارد و کنترل میکند که چه نوع تعاملاتی خارج از قلمرو آن مجاز هستند.
- قابل استفاده توسط واحدهای خودگردانی که سرویس های خود را توسعه میدهند و مستقر می کنند اما میخواهند گروهی منتخب از سرویس ها را در دسترس بقیه سازمان قرار دهند.

پیکربندی های استقرار گذرگاه سرویس سازمان واسطه ...



انواع گذرگاه های سرویس سازمان



- انواع گذرگاه های سرویس سازمان
- EBS مبتنی بر سرور برنامه کاربردی
- EBS مبتنی بر سیستم پیام رسانی
- EBS مبتنی بر سخت افزار

انواع گذرگاه های سرویس سازمان

EBS مبتنی بر سرور برنامه کاربردی



- از سرورهای برنامه کاربردی به عنوان ستون فقرات استفاده میکنند.
- علاوه بر کارکرد رایج همگام، از پیام رسانی ناهمگام نیز پشتیبانی می کنند.

• مزیت:

- هزینه نسبتاً پایین
- راه اندازی آسان در زمان استقرار



انواع گذرگاه های سرویس سازمان EBS مبتنی بر سرور برنامه کاربردی ...

- معایب
- قدرت این محصولات در سروکار داشتن با جاوا و XML است. بنابراین اگر مجموعه متنوع تری از برنامه های کاربردی نیاز به یکپارچه سازی داشته باشند، چالش هایی را به وجود می آورند.
- معمولاً برای یکپارچه سازی تعداد نسبتاً کمی از برنامه های کاربردی استفاده می شوند زیرا این نوع برای تعداد زیادی برنامه کاربردی، مقیاس پذیر نیست.

انواع گذرگاه های سرویس سازمان

EBS مبتنی بر سرور برنامه کاربردی ...



- مثال: گذرگاه سرویس سازمان webSphere (WESB) از شرکت IBM که مبتنی بر سرور برنامه کاربردی webSphere (WAS) است.
- ویژگی ها
- یکپارچه سازی مبتنی بر استانداردها که باعث میشود با کاهش تعداد و پیچیدگی رابطها، تعاملات بین برنامه های کاربردی و سرویس ها را به سرعت و به آسانی ایجاد و مستقر کنید.
- ابزارهای سهل استفاده که به حداقل مهارت برنامه نویسی نیاز دارند و نصب، پیکربندی، ساخت و مدیریت ساده ای دارند.
- به طور پویا پیکربندی مجدد میشوند تا بارهای پردازشی کسب و کار متغیر را تامین کند و امکان تعاملات ساده را با هر برنامه کاربردی JMS و HTTP فراهم کند.

انواع گذرگاه های سرویس سازمان

EBS مبتنی بر سرور برنامه کاربردی ...



- ویژگی ها...
- پشتیبانی از تعداد زیادی سیستم عامل: AIX، HP Unix، I Family، Linux، Sun Solaris، Windows، و z/OS
- چابکی و انعطاف پذیری کسب و کار را افزایش میدهد و به آسانی به سمت مدل گذرگاه سرویس سازمان متحد، توسعه می یابد.
- صدها فروشنده نرم افزار مستقل را با آداپتورهای WebSphere پشتیبانی میکند.

انواع گذرگاه های سرویس سازمان

EBS مبتنی بر سیستم پیام رسانی



- ستون فقرات یک سیستم پیام رسانی مانند WebSphere MQ شرکت IBM است.
- از هر دو نوع پیام رسانی پشتیبانی میکند: همگام و ناهمگام

مزایا نسبت به دو نوع دیگر:

- مقیاس پذیری بالاتر و حمایت از حجم زیادی از تراکنش ها
- ارائه متنوع ترین مجموعه از برنامه های کاربردی از جمله برنامه های جاوا، C، C++، COBOL
- تضمین تحویل پیام ها

معایب:

- راه اندازی دشوار
- هزینه بیشتر

انواع گذرگاه های سرویس سازمان

EBS مبتنی بر سیستم پیام رسانی ...



- مثال: واسط پیام WebShpere شرکت IBM (WMB) که ستون فقرات آن IBM WebShpere MQ است.
- ویژگی ها:
- متنوع ترین مجموعه از برنامه های کاربردی را یکپارچه میکند.
- از جمله برنامه J2EE، برنامه های c/c++، برنامه های CRM، و برنامه های مین فریم COBOL
- میتواند حجم زیادی از تراکنش ها را مدیریت کند.
- پیام های در حال تبادل را بین هر ترکیبی از فرمت های مختلف پیام، تایید و تبدیل میکند، از جمله وب سرویس ها، XML و فرمت های غیر XML
- پیام ها را براساس قوانین کسب و کار مسیریابی میکند تا محتوای اطلاعات و فرآیندهای کسب و کار را تطبیق دهد.
- بدون نیاز به برنامه نویسی مجدد برنامه های کاربردی نقطه پایانی، الگوهای توزیع اطلاعات را برای پیکربندی مجدد فراهم میکند.
- از مدل امنیتی قدرتمندی پشتیبانی میکند.
- از مجازی سازی سرویس ها با استفاده از رجیستری و سرویس های Web Shpere پشتیبانی میکند.

انواع گذرگاه های سرویس سازمان

EBS مبتنی بر سیستم پیام رسانی ...



- ویژگی ها...
- بین درخواست های وب سرویس ها و ارائه دهندگان وساطت می کند (امکان مسیریابی، تبدیل و واقعه نگاری را فراهم میکند).
- با آخرین پیاده سازی استانداردها کار میکند.
- در سیستم عاملهای مختلفی قابل استفاده است از جمله:

IBM z/OS, IBM AIX, Linux, Solaris, HP/UX, Microsoft Windows Server

انواع گذرگاه های سرویس سازمان EBS مبتنی بر سخت افزار



- این نوع، برای انجام بیشتر کار پردازش، به سخت افزار تکیه دارد.

- مزیت:

- راه اندازی آسان

- امنیت بالا

- پردازش کارا

انواع گذرگاه های سرویس سازمان

EBS مبتنی بر سیستم پیام رسانی ...



• مثال: Web Shpere Data Power که ویژگی های زیر را دارد:

- فرمت های مختلف پیام را تبدیل میکند از جمله XML, Legacy, binary
- امنیت و مسیر یابی پیام، اتصال MQ/HTTP/FTP و وساطت انتقال را فراهم میکند.
- تبدیلات مستقل از انتقال بین پیام های مختلف را فراهم میکند.
- امنیت و نظارت متمرکز و مبتنی بر استانداردها را برای معماری سرویس گرا فراهم می کند.
- تعامل بین چند برنامه کاربردی نامتجانس را ممکن میکند از جمله اتصال محلی به رجیستری ها و مخازن و دسترسی مستقیم به پایگاه داده.

