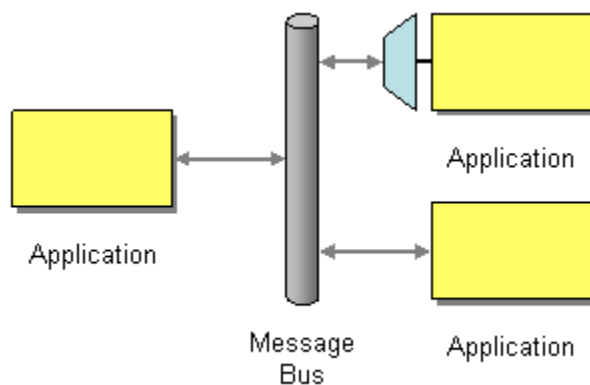


## پاسخ به سوالات مطرح شده بعد از ارائه

**Message Bus** را بیشتر توضیح دهید. به همراه مثال و کاربردها.

**Message Bus** ترکیبی از یک مدل داده مشترک ، یک مجموعه دستورالعمل مشترک و یک زیرساخت پیام رسانی است و اجازه میدهد تا ارتباط بین سیستم های مختلف از طریق مجموعه ای از رابط های اشتراکی برقرار شود. درواقع شبیه به یک گذرگاه در یک سیستم کامپیوتری، که به عنوان نقطه کانونی برای ارتباط بین پردازنده، حافظه اصلی، و لوازم جانبی سرویس میدهد.



همانطور که در اسلایدها هم گفتیم، یک برنامه وقتی که پیام تولید کرده را بخواهد در اختیار برنامه های دیگر قرار دهد، اینکار را از طریق **Message Bus** انجام می دهد. یعنی پیام خود را در **Message Bus** گذاشته و سایر برنامه ها از آن استفاده می کنند.

## Message Pattern را بیشتر توضیح دهید. به همراه مثال و کاربرد.

در زمان ارائه گفتیم که، Message اطلاعاتی است که می تواند در سیستم Messaging جابجا شود و این جابجایی از طریق Message channel صورت میگیرد.

در مورد هر کدام از Pattern ها مفصل توضیح داده شد. میتوانید به اسلایدها مراجعه کنید و یکبار دیگر توضیحات و تصاویر دقت فرمایید.

## آیا EIP در سطح معماری سیستم است یا در سطح پیاده سازی نرم افزاری یا سخت افزاری؟

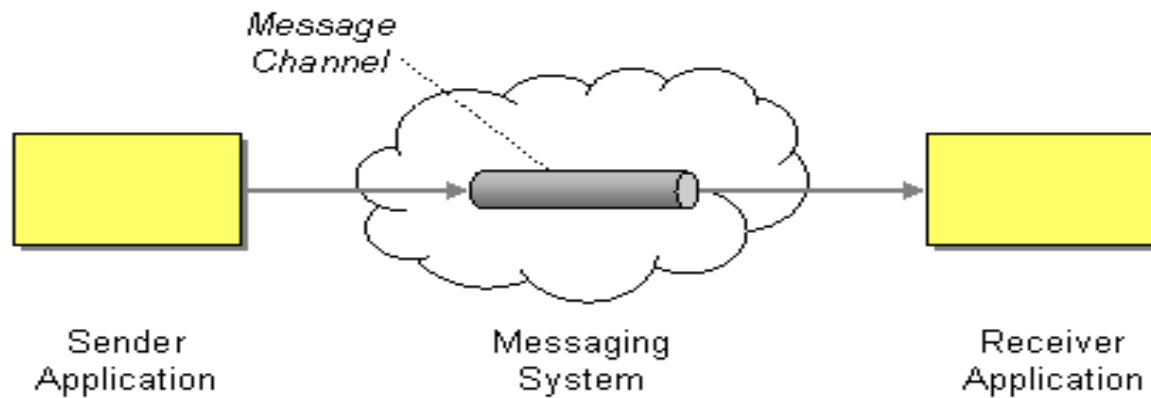
در ابتدای ارائه و در زمان معرفی EIP به این نکته تاکید کردم که EIP یک روش دیگری برای برنامه نویسی است به این ترتیب که با ارائه یکسری الگوها، استفاده از المان های قدیمی مانند if و for را کم میکند. با توجه به اینکه از برنامه نویسی در زمان پیاده سازی استفاده میکنیم، پس این روش هم باید در سطح پیاده سازی نرم افزاری کاربرد داشته باشد.

## Message Channel را توضیح دهید. به همراه مثال و کاربرد.

نمیدونم در زمان ارائه حواستون کجا بود! 😊

گفتیم که پیام از طریق Message channel منتقل می شود و Channel ورودی و خروجی دارد و پیام را به فیلتر منتقل می کند.

به شکل زیر توجه کنید:



اگر نیاز به توضیح بیشتری هست، به بخش انواع Message Channel ها در فایل ارائه مراجعه کنید.

### در صورت عدم انتقال چه اتفاقی برای پیام می افتد؟

در زمان ارائه و در بخش Message Filter گفتیم که این کامپوننت یکسری از پیام هایی را که مد نظر ما نیست را حذف میکند و فقط آن اطلاعات و پیام هایی را در کانال خروجی اش می آورد که مدنظر ماست. مثال هم زدیم و در شکل توضیح دادیم که پیامی که مدنظر ما نبود در پیام منتقل شده نیز حضور نداشت. در واقع بوسیله Message Filter حذف شد. Message Filter یک خروجی دارد در واقع Single Output می باشد. به این ترتیب که اگر تطبیق شد پیام منتقل می شود و گرنه حذف می شود.

در بخشی که در مورد انواع Message Channel ها توضیح میدادم در مورد Dead letter channel گفتیم که وقتی که در هر صورتی انتقال پیام (Delivery) انجام نشود (Fail) چند مرتبه عمل ارسال تکرار شده و در صورت عدم ارسال ، این پیام به Dead letter channel منتقل می شود تا در مراجعات بعدی بفهمیم که این پیام منتقل نشده است.

## آیا EIP برای پیام‌ها مشخصات خاصی را تعریف کرده است؟

گفتیم که Messaging کار را برای برقراری ارتباط میان نرم‌افزارهای مختلف (languages, compilers, and operating systems) آسان‌تر می‌کند چرا که تنها چیزی که باید بدانند پروتکل و فرمت پیام مشترک است. پس باید یک فرمت و استانداردهایی را رعایت کنند و همچنین توضیح دادیم که در زمان ارتباط چند برنامه به هم ممکن است فرمت اطلاعات رو عوض کنیم. مثلاً یک برنامه که XML می‌گیرد را به یک برنامه که JSON می‌گیرد وصل کنیم. در این موارد باید از Message Translator استفاده کنیم. همانطور که در زمان ارائه بر روی شکل مشاهده کردید و توضیح نیز داده شد، یک پیامی را گرفته و یک پیام با یک ترتیب دیگر را منتقل کرده است. در واقع فرمت آن را تبدیل کرده است.

## آیا EIP فقط یک معماری است یا برای هر نوع مسئله یک یا چند سیستم پیشنهاد داده میشود؟

گفتیم که EIP یکسری راه حل‌های تست شده برای برخی مشکلات هستند و میشه گفت مانند Design Pattern ها هستند.

یک روش دیگری برای برنامه نویسی است و یک روش توسعه نرم‌افزار است که با ترکیب سرویس‌هایی که وجود دارند، سرویس‌های تازه‌تر را می‌سازد.

پس برای یکسری مشکل‌های خاص، استفاده از این روش موثر می‌باشد.

پس نمیتونه یک معماری باشه، برای هر نوع مسئله ای قابل استفاده نیست یا حداقل بهترین گزینه نیست.

## ارتباط بین Service Bus و Message Bus چیست؟ آیا یکی هستند چه مفاهیم متفاوتی دارند؟

با EIP که مختصراً آشنا شدیم و در مورد Message Bus هم توضیح دادیم. احتمالاً بعد از آشنایی مختصر با ESB، این سوال، پاسخ داده شود.

ESB برگرفته شده از واژه‌های Enterprise Service Bus و بمعنای ابزاری برای یکپارچه کردن انواع برنامه‌های کاربردی که توسط چندین شرکت تهیه شده است، می‌باشد. راهکارهای مختلفی برای اجرای گذرگاه خدمات سازمانی وجود دارد که هر یک روش‌ها و معماری خود را پیشنهاد می‌دهد. مفهوم بنیادین در ESB اتصال انواع مختلف برنامه‌های سازمانی به یک گذرگاه پیام، و قادر ساختن آن برنامه‌ها به برقراری ارتباط با این گذرگاه است. این عملکرد باعث از بین رفتن وابستگی میان برنامه‌های مختلف می‌شود، و اجازه می‌دهد این برنامه‌ها بدون نیاز به داشتن اطلاع از دیگر برنامه‌های روی گذرگاه تنها با گذرگاه ارتباط برقرار کنند.

## یک شرح مختصری از اینکه چطور میتوان در یک Scale کوچک از مزیت این Pattern ها استفاده کرد.

در جواب این سوال باید عرض کنم که باتوجه به گستردگی این مفاهیم و الگوها و همانطوری که در اول ارائه هم گفتیم "قرار است بصورت خیلی کلی با مفاهیم و برخی از الگوهای EIP آشنا شویم" و عدم تجربه اینجانب در پیاده سازی با استفاده از فریمورک هایی که از EIP پشتیبانی میکنند، باید بگویم که پاسخ مناسب این سوال را نمیدانم.