

یکپارچه سازی سیستم های سازمانی (Enterprise Application Integration)

ارائه دهنده: امیر مهجوریان

اسفند ۹۲

کارگاه آموزشی شرکت ...

امیر مهجوریان

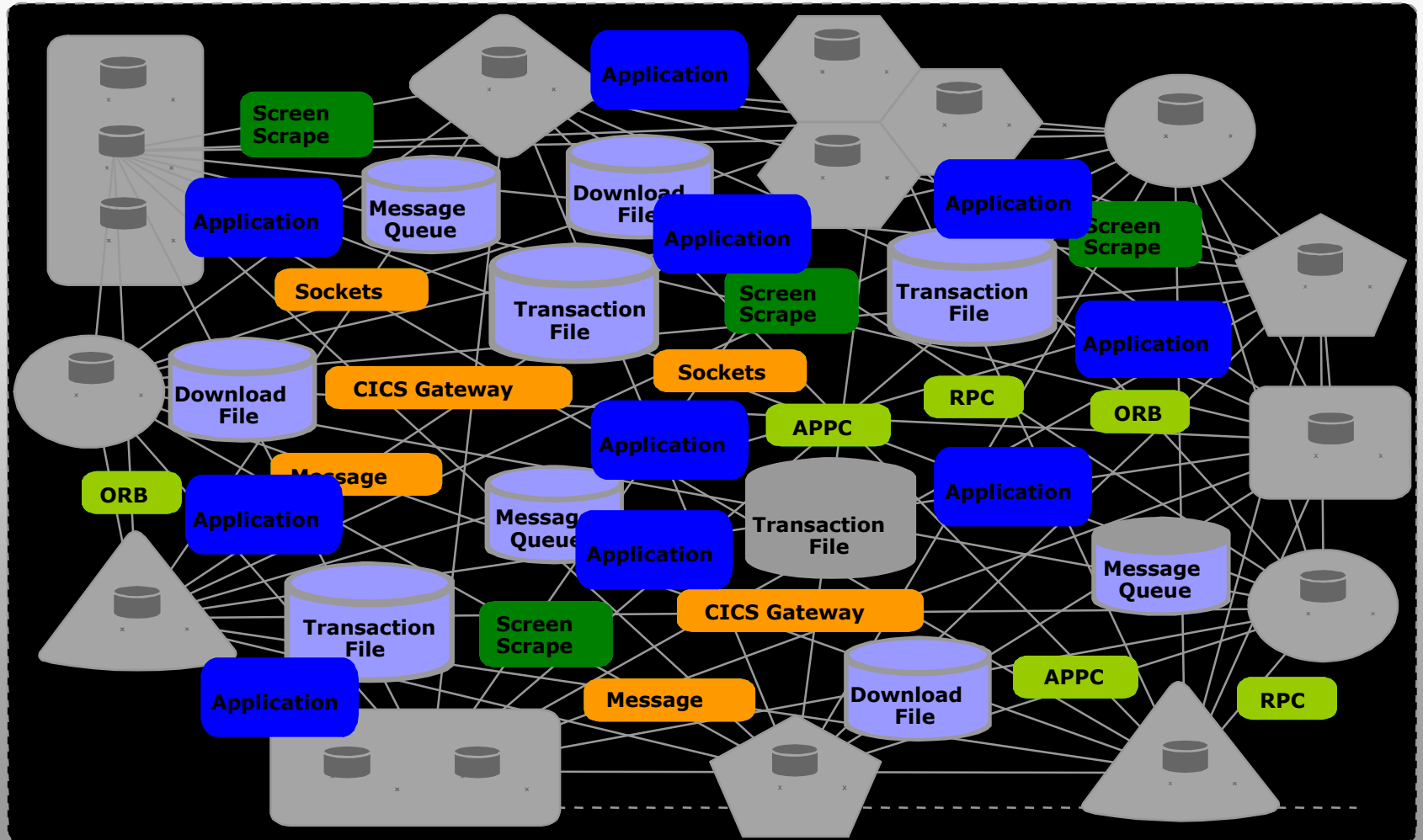
مدیر فنی آزمایشگاه معماری سازمانی سرویس گرا
مدیرعامل شرکت دانش بنیان کاریز سیستم

60% of the implementation cost of an ERP package is spent on integration

یکپارچگی

مفاهیم و تاریخچه

وضعیت فعلی سیستم های سازمانی



چالش های سیستم های اطلاعاتی موجود



یکپارچه سازی سیستم های سازمانی

- هدف آن به اشتراک گذاری کارآمد و یکپارچه داده ها، نرم افزارها و فرایندهای سازمانی در یک بستر سازمانی است
- ارتباط بین اطلاعات ، نرم افزارها و سیستم های متنوع سازمانی نوعی محیط جدید مجازی را تداعی می کند
- شامل "بستر" و "معماری" مورد نیاز برای تعامل پذیری سامانه ای سازمانی است که از حدود سال ۲۰۰۴ مورد توجه قرار گرفت
- رابطه نزدیکی با مدیریت فرایندهای حرفه (BPM) دارد، زیر BPM بر اساس یکپارچگی اجرا و مدیریت فرایندها شکل گرفته و جریان کار یک فرایند از واحدهای مختلف سازمانی و سیستم های مربوط به آنها عبور کرده و نیاز به استفاده از اطلاعات و مولفه های سیستم های متنوع سازمانی را دارد.

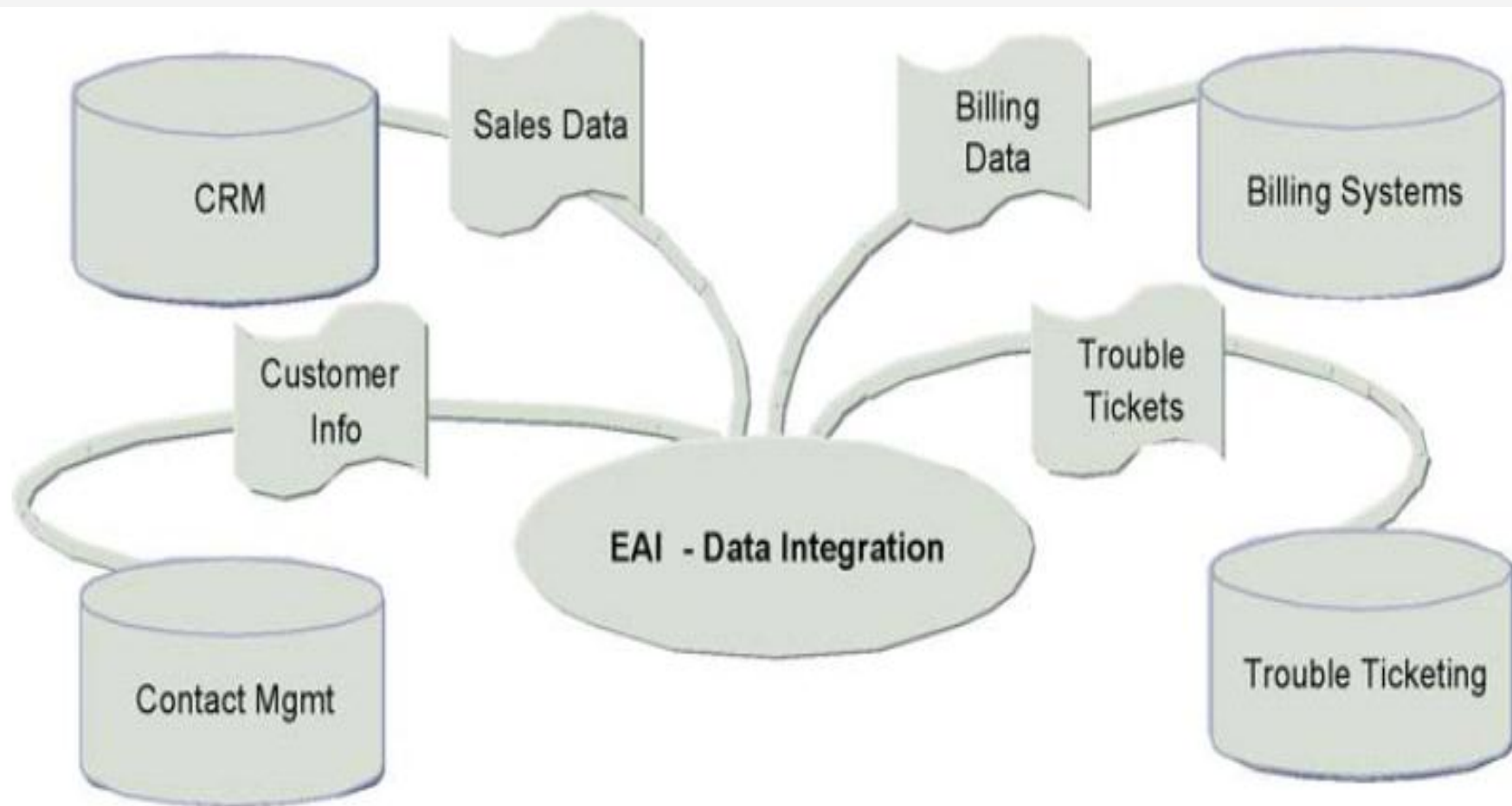
- Data Integration
- Point-to-Point Application Connections
- Portals
- Integration broker (EAI)
- SOA wit ESB

Data Integration

این رهیافت یکی از ساده ترین راه های یکپارچگی است که در ان منابع اطلاعاتی بین نرم افزارهای مختلف مبادله می شوند تا درستی و جامعیت اطلاعات نرم افزارهای مختلف با یکدیگر محقق شود. برخی مکانیزم های این رهیافت به صورت زیر است:

- وقتی نرم افزاری داده های اطلاعاتی بانک مربوط به خود را تغییر داد، بصورت اتوماتیک داده های دیگر بانک های مربوط نیز به روز رسانی شوند.
- ایجاد بانک های اطلاعاتی توزیع شده با تعیین سطوح دسترسی برای نرم افزارها

Data Integration



Point-to-Point Application Connections

- در این رهیافت هر دو نرم افزاری که قصد تعامل داشته باشند کانالی بر طبق استانداردها و پروتکل های مورد توافق دو طرف ایجاد می کنند. معمولا نرم افزار متقاضی یک درخواست اطلاعات را در قالب فرمت نرم افزار گیرنده تهیه و ارسال می کند.
- در شرایطی که تعداد نرم افزارها محدود باشد این رهیافت قابل استفاده است اما با زیاد شدن نرم افزارها و تنوع فناوری ها و پروتکل ها میزان اتصالات و تعاملات زیاد شده و مدیریت و توسعه چنین معماری سخت و پیچیده خواهد بود.

Point-to-Point Application Connections



Portals

- وظیفه پرتال ها ارائه خدمات وابسته به منابع مختلف از طریق یک واسط استاندارد و یکپارچه است.
- پرتالها باعث یکپارچگی یا حتی ارتباط سیستم های سازمانی با همدیگر نمی شوند، در عوض درگاه ارائه سرویس های این سیستم های متنوع به کاربران را یکپارچه می کنند. بنابراین پرتال ها را باید از جمله رهیافتهای یکپارچگی واسط ارائه سرویس به ذینفعان دانست.

Portals



Integration broker (EAI)

- در این رهیافت میان افزار (Middleware) واسط وظیفه یکپارچه نمودن نرم افزارهای متنوع سازمانی را به وسیله تبدیل فرمت پیامها به عهده می گیرد.
- این رهیافت حاوی تکنیک ها و تکنولوژی های مختلفی برای یکپارچه نمودن نرم افزارهای متنوع سازمانی است و در آن انواعی از پیامهای بین نرم افزاری، بین پایگاههای داده و فراخوانی مولفه ها و سرویس های سطح پائین وجود دارد.
- در این رهیافت هر نرم افزار (طرف) پیامهای خود را با فرمت و استاندارد خود ارسال می کند، وظیفه این میان افزار تشخیص فرمت و نوع پیام دریافتی و مقصد آن است تا سپس پیام را به فرمت تکنولوژی مقصد تبدیل نموده و ارسال نماید. به طور خلاصه این میان افزار، نقش دستگاه مترجم مرکزی در سمینارهای بین المللی را بازی می کند که می تواند زبان های مختلف را به هم ترجمه کند.

Integration broker (EAI)



SOA wit ESB

- تفاوت این رهیافت با سایر مواردی که گفته شد این است که در معماری سرویس گرا بجای تلاش برای متصل کردن سیستم های ذاتا ناهمجور اصل بر ساخت سیستم ها و مولفه های خوش تعریف و مستقل از فناوری/سکوی است که در بستر گذرگاه سرویس سازمان توسط سیستم ها فراخوانی می شوند.
- بطور خلاصه اگر توسعه و طراحی سیستم های سازمانی بر اساس معماری سرویس گرا باشد انگاه این سیستم ها ذاتا با هم یکپارچه بوده و اصلا دیگر نیازی به ابداع متد و روشی برای اتصال انها نیست.
- این رهیافت تمام قابلیت های رهیافتهای پیشین را داشته و قادر است یکپارچگی را در سطوح مختلف داده/فرایند/سرویس پشتیبانی کند. با گسترش این سبک معماری رهیافتهای گفته شده یکپارچگی در حال جایگزینی و فراموشی هستند

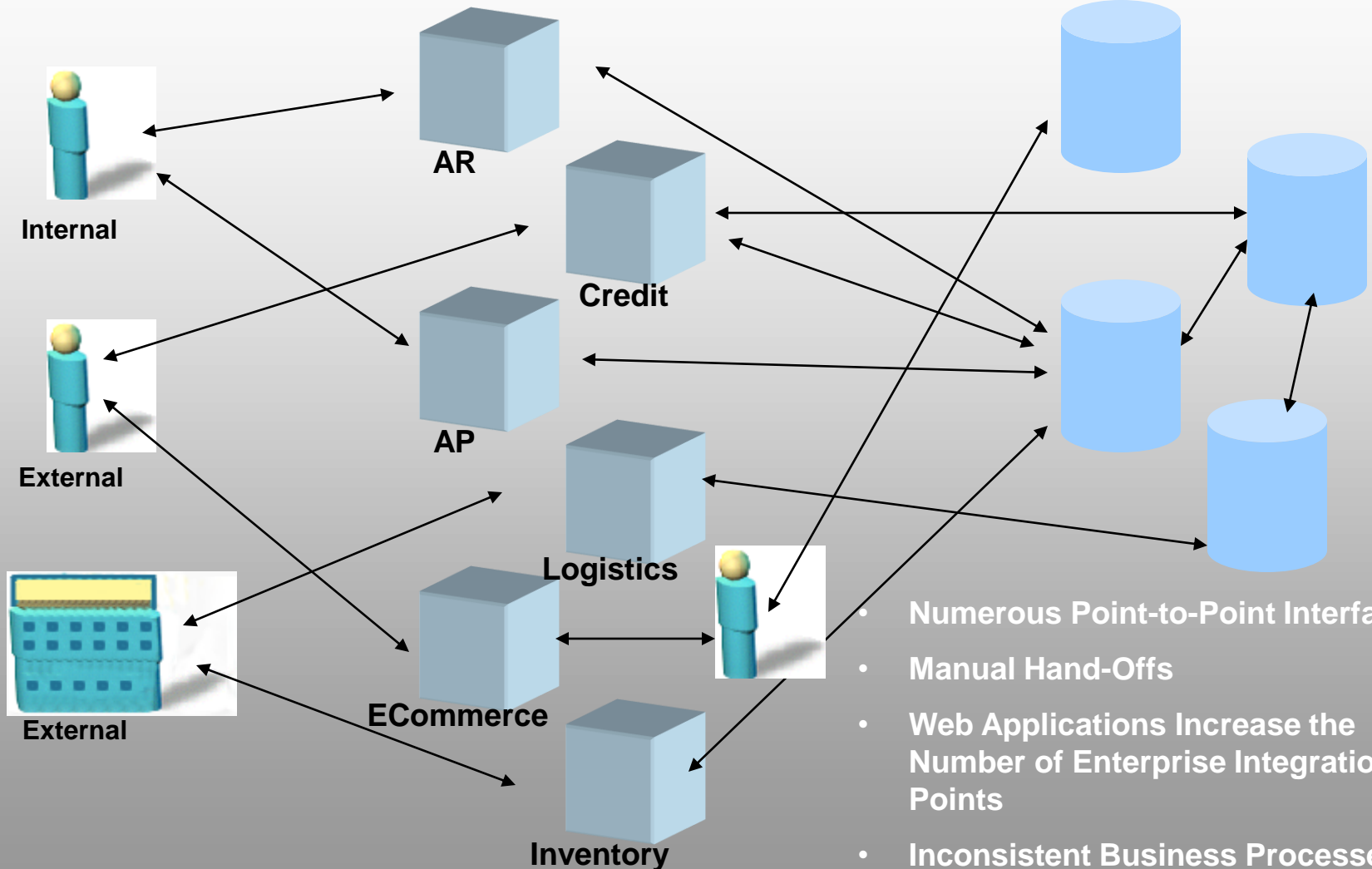
رهیافت پیشنهادی به سازمانها

- اگرچه مزیت های معماری سرویس گرا از دیگر رهیاftها بیشتر است اما از انجا که سازمانها نمی توانند در یک گام مجموعه سیستم ها و زیر ساخت های خود را تغییر داده و به عبارتی سرویس گرا شدن یک روند تکاملی و چند ساله است لذا راه حل پیشنهادی به سازمانها استفاده از معماری سرویس گرا به همراه آداپتور های یکپارچگی با سیستم های غیر سرویس گرا است.
- بدین ترتیب تا زمان سرویس گرا شدن همه سیستم های سازمان, امکان اتصال و تعامل سیستم های سرویس گرای جدید با سیستم های موروثی نیز خواهد بود

یکپارچه سازی سیستم های سازمانی

تکنیک ها و آدابورها

معماری بدون EAI

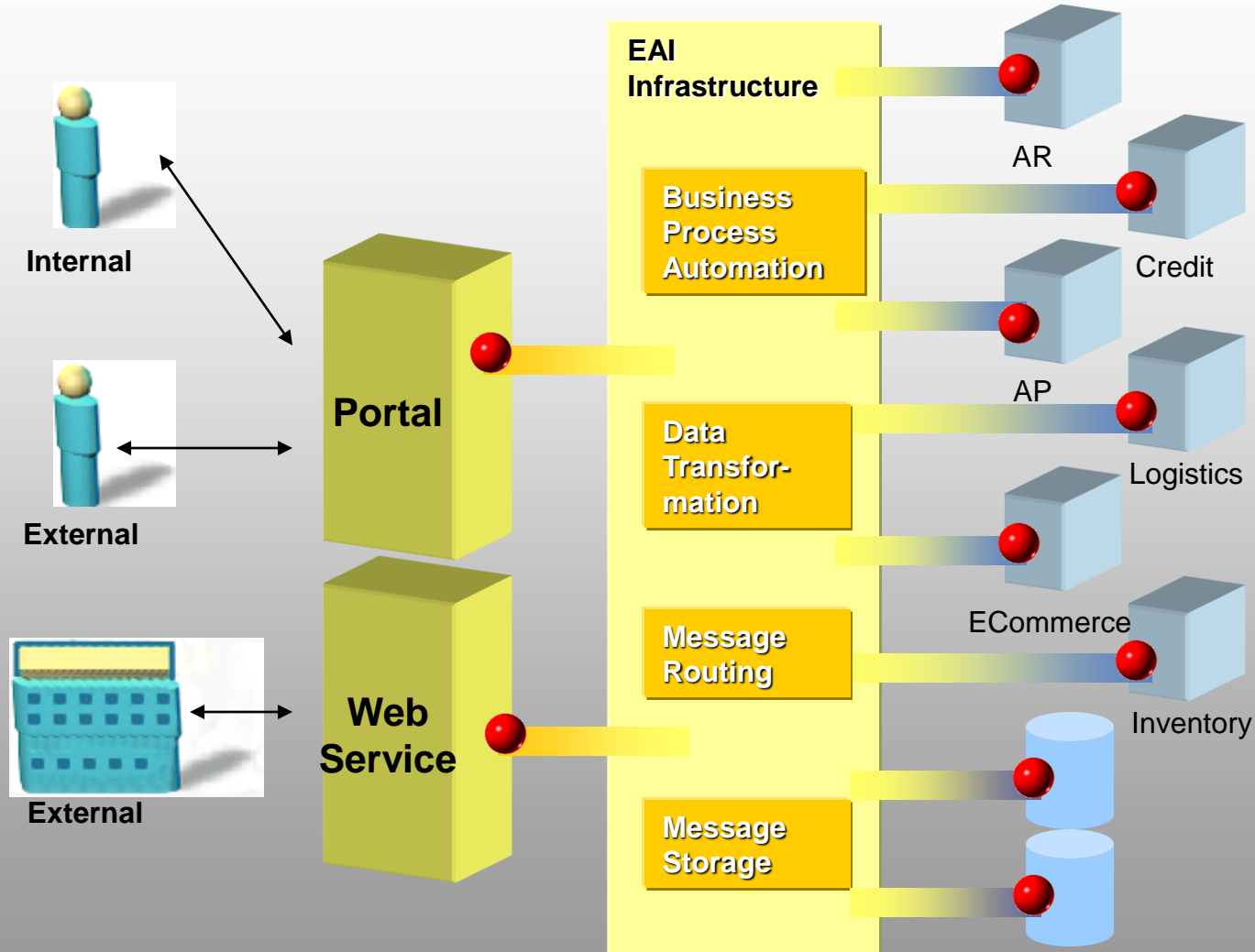


- Numerous Point-to-Point Interfaces
- Manual Hand-Offs
- Web Applications Increase the Number of Enterprise Integration Points
- Inconsistent Business Processes
- Inconsistent Qualities of Service

معماری سیستم های اطلاعاتی سازمانی وقتی EAI نباشد

- ارتباط بین سیستم ها از متد نقطه-به نقطه است
- تنظیمات دستی و متعدد
- فرایندهای حرفه ناقص و ناهماهنگ
- مدیریت و توسعه مشکل و پرهزینه

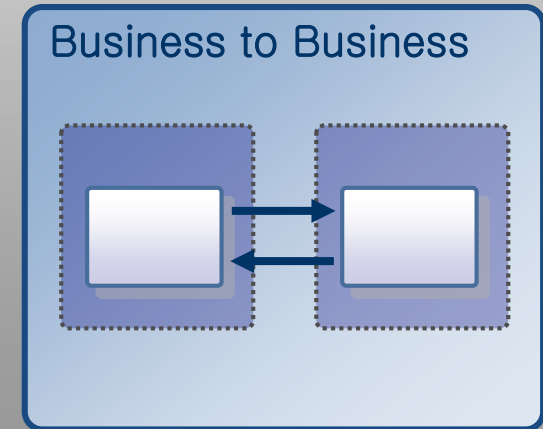
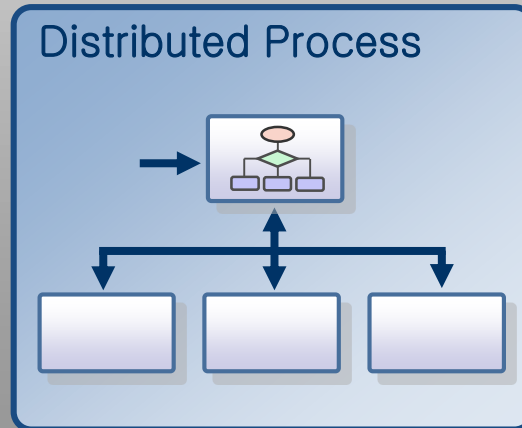
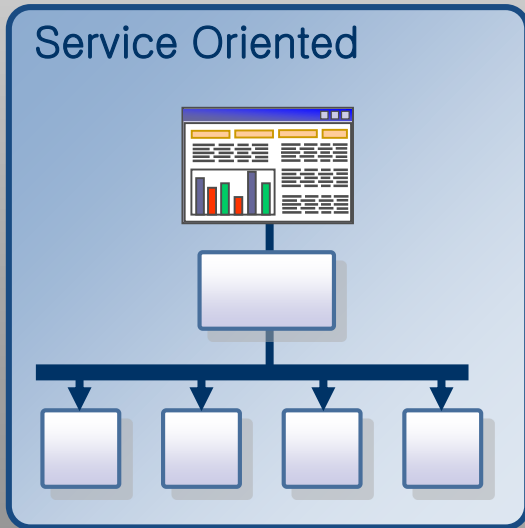
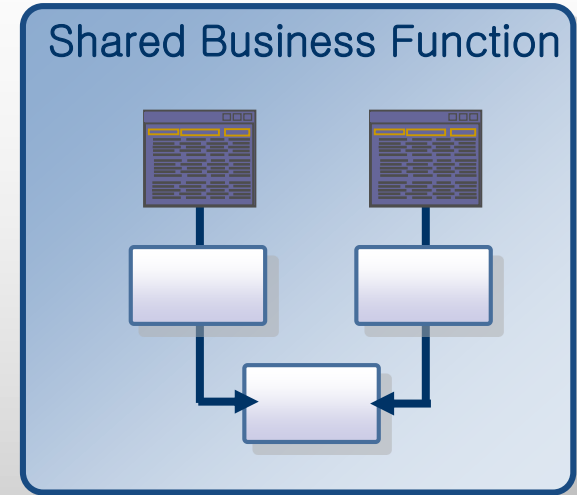
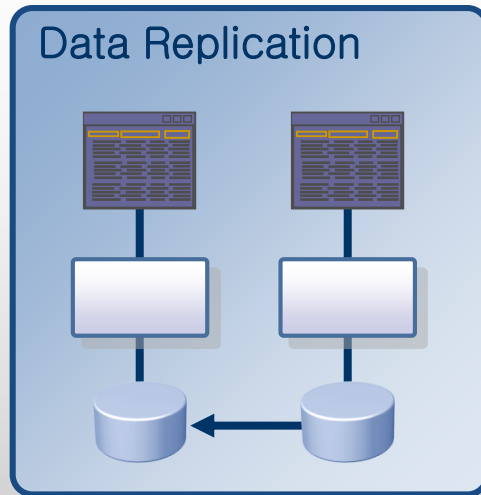
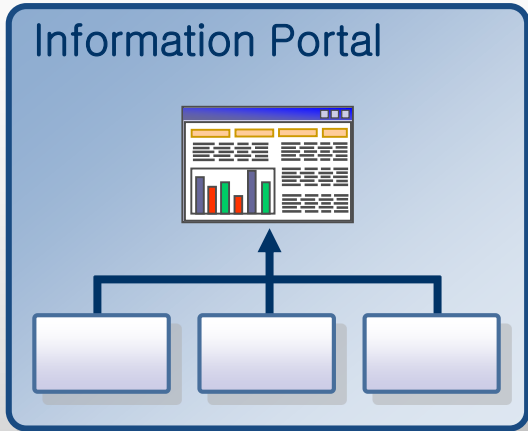
EAI معماری



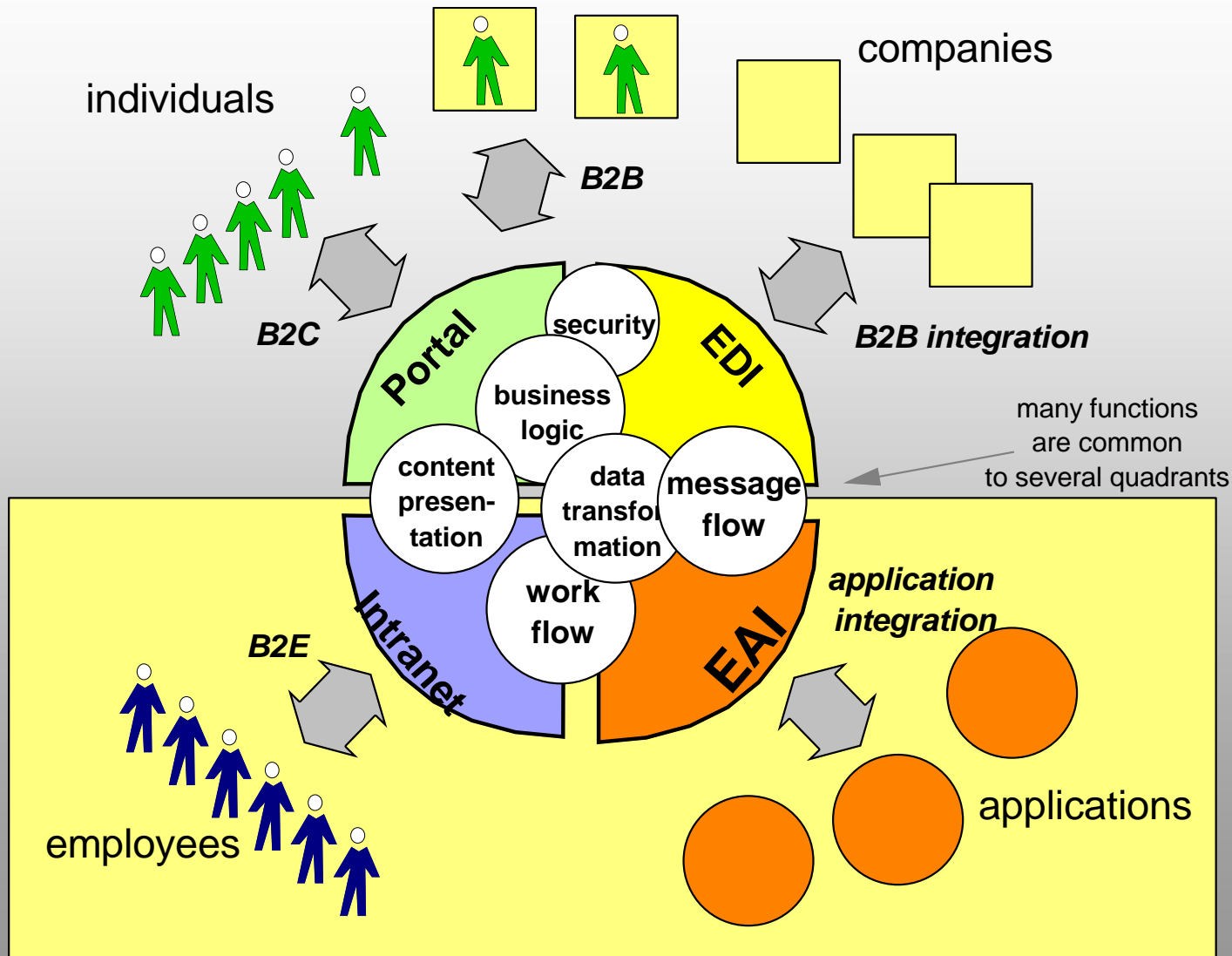
معماری سیستم های اطلاعاتی سازمانی وقتی EAI باشد

- کاهش نقاط و خطوط ارتباطی
- اجرای فرایندهای حرفه اتوماسیون شده و یکپارچه
- مدیریت و توسعه ساده و کم هزینه

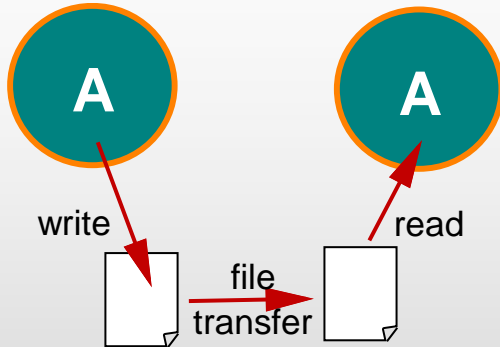
تکنیک های تعامل بین سیستم ها



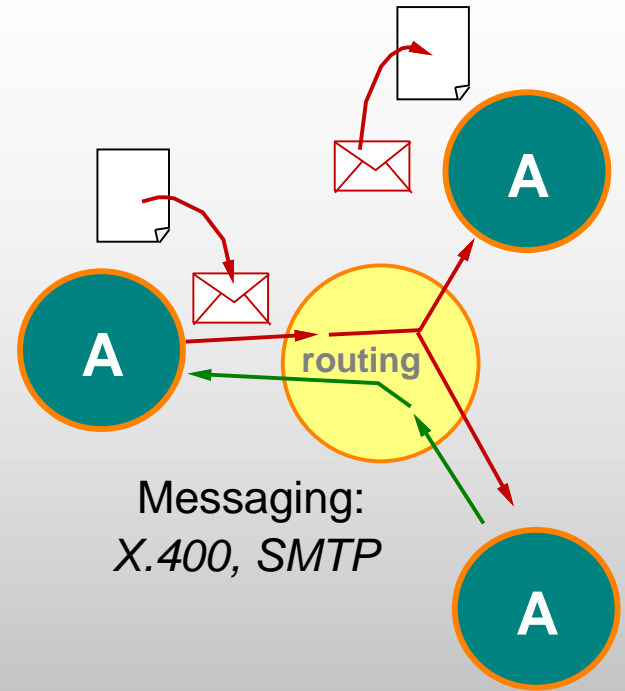
یکپارچگی در دیگر حوزه ها



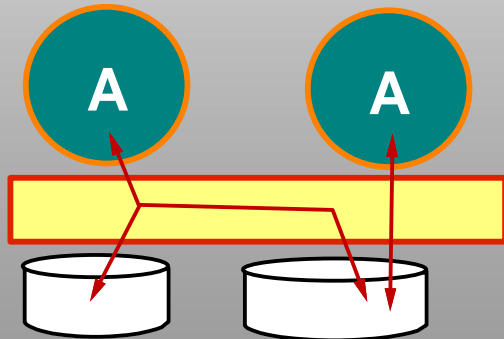
برخی تکنولوژی های تعامل بین نرم افزارها



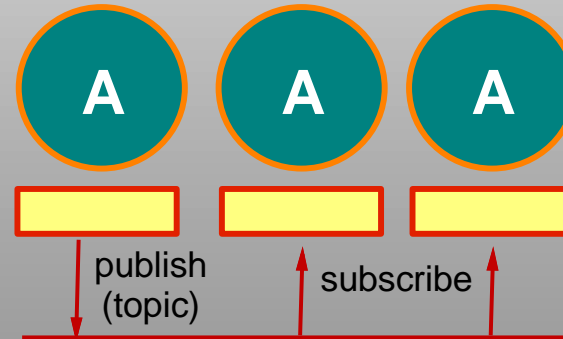
File Transfer:
*IBM RJE, FTP,
Connect:Direct, ...*



Messaging:
X.400, SMTP

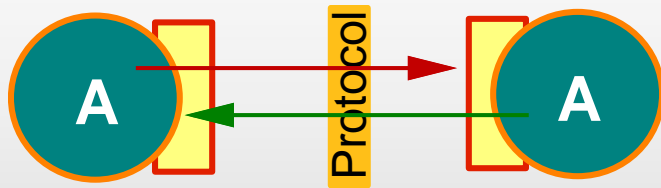


Remote Data Access:
*Remote SQL, ODBC,
JDBC, ...*

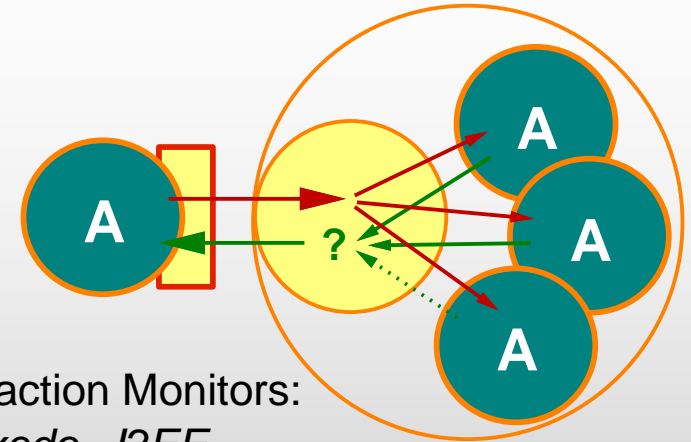


Application buses:
*TIBCO Rendez-vous, JMS,
MQ-series, MSMQ,...*

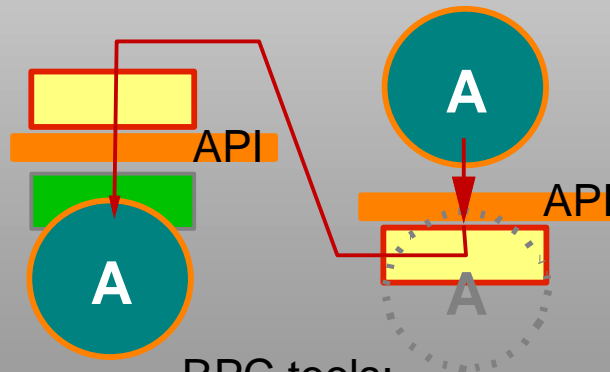
برخی تکنولوژی های تعامل بین نرم افزارها



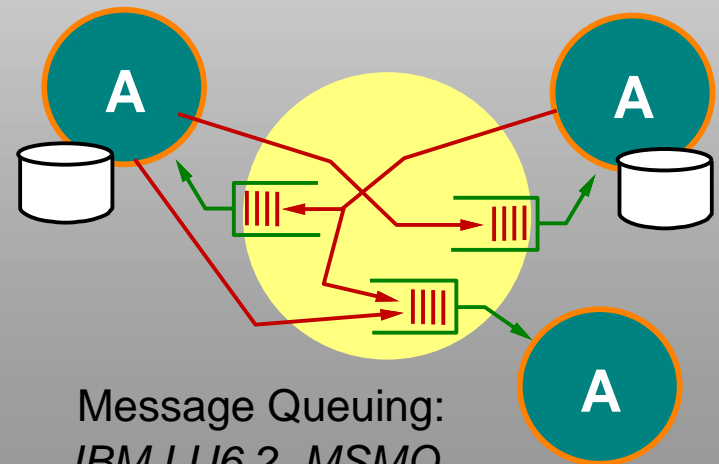
Peer-to-peer protocols:
*BSC, HDLC, X.25, ZMODEM,
 TCP, IBM LU6.2, HTTP,
 SOAP, ...*



Transaction Monitors:
Tuxedo, J2EE ...

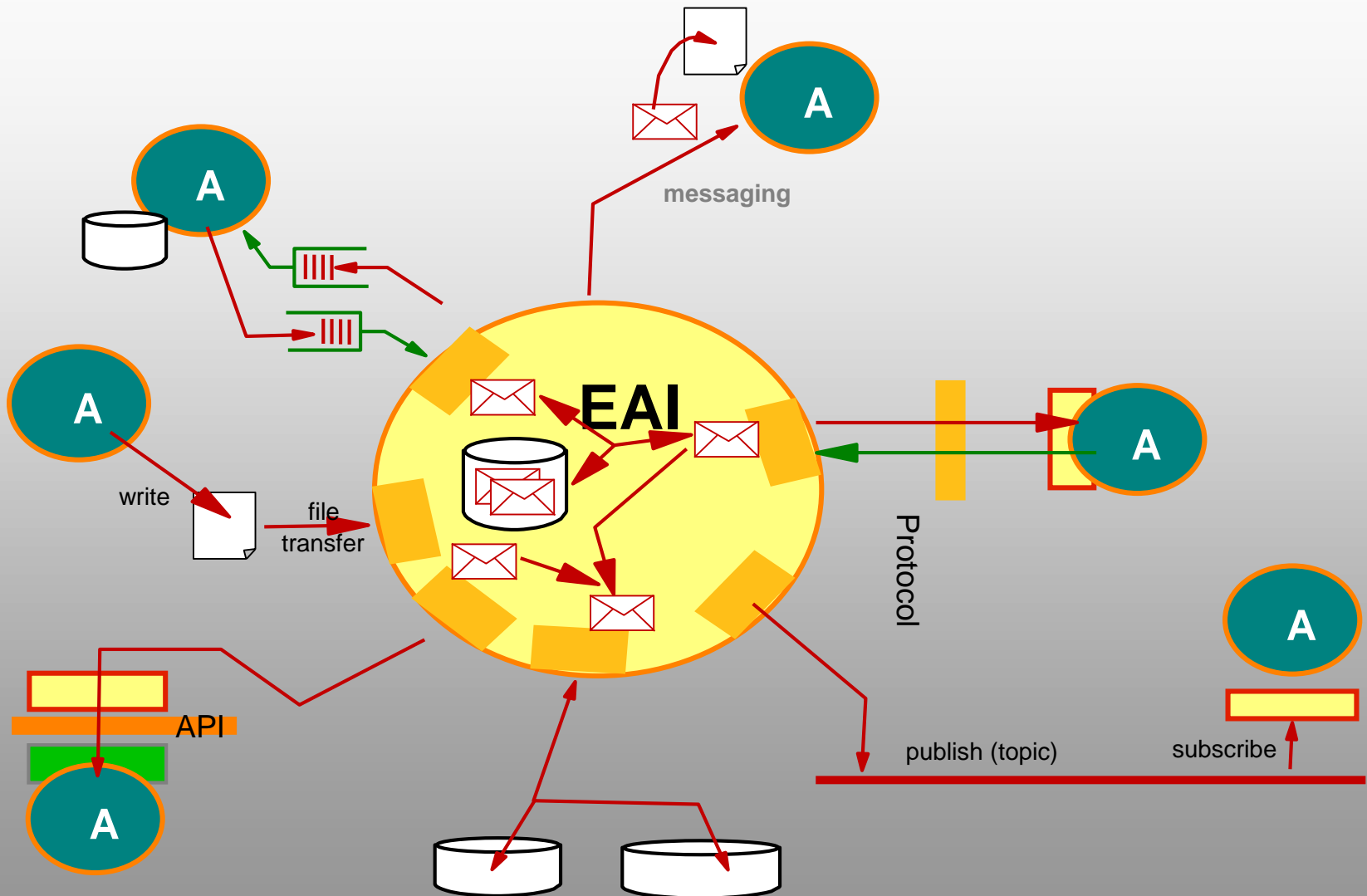


RPC tools:
*CORBA, Java RMI, DCOM,
 EJB's, ... web services (SOAP)*



Message Queuing:
*IBM LU6.2, MSMQ,
 MQ Series, JMS, ...*

EAI: پشتیبانی از تمام تکنولوژی ها و فناوری های ارتباطی

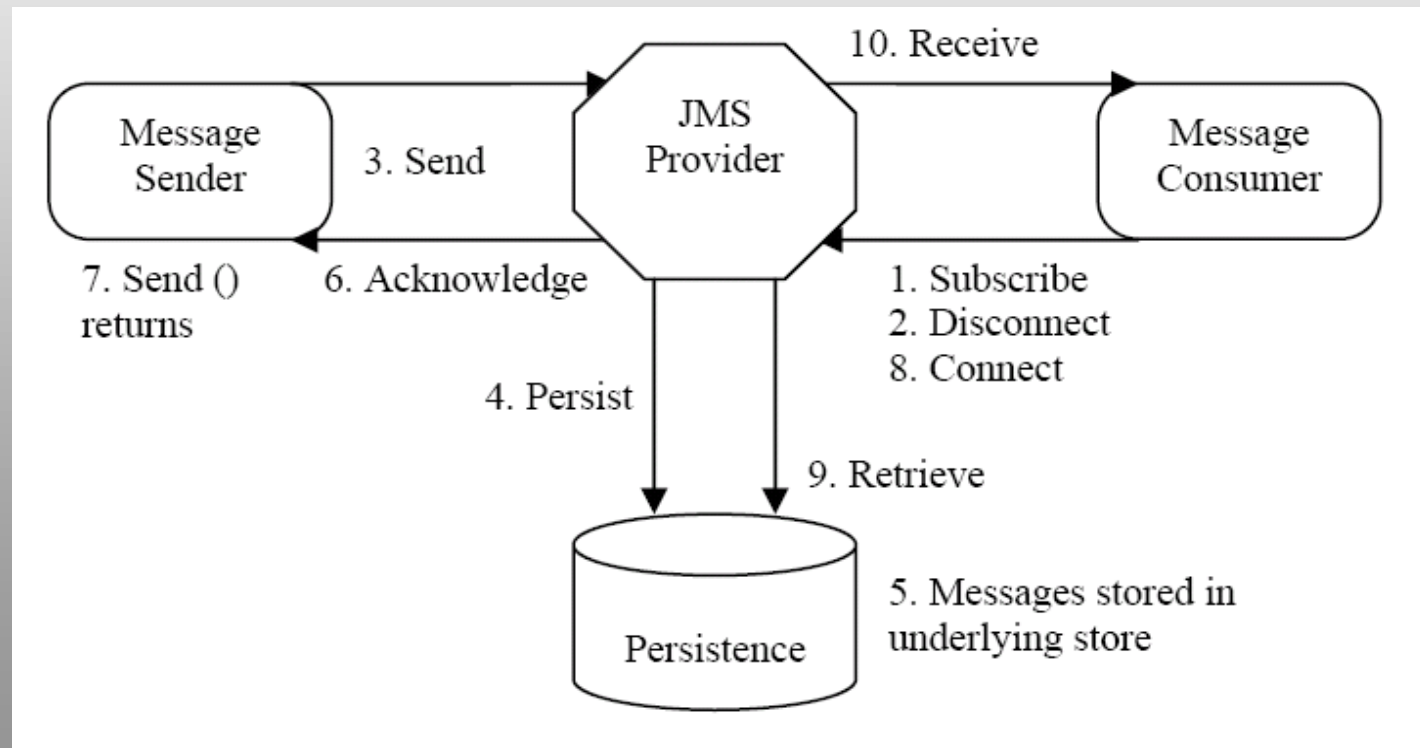


استانداردهای اتصال بین نرم افزارها

JMS, CORBA, RMI, RPC,
DCOM, WEB Service, ...

Java Message Service (JMS)

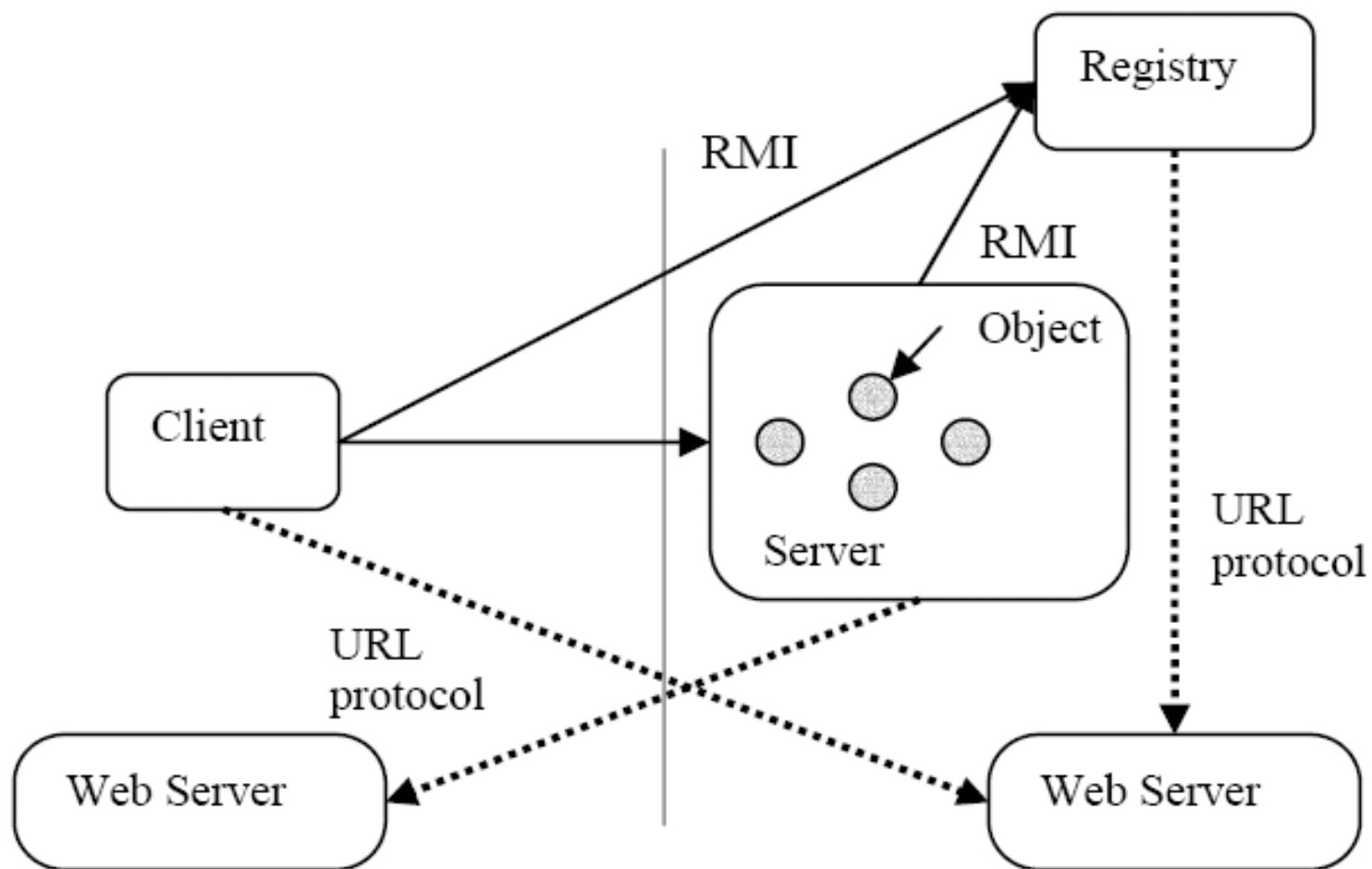
Messaging systems at enterprise level provides reliable transport of information between different applications across a variety of heterogeneous computer networks and systems. The architectural structure allows queuing of messages with guaranteed, timeliness delivery and ensuring offline applications to process messages later when they are capable of receiving them.



Remote Method Invocation (RMI)

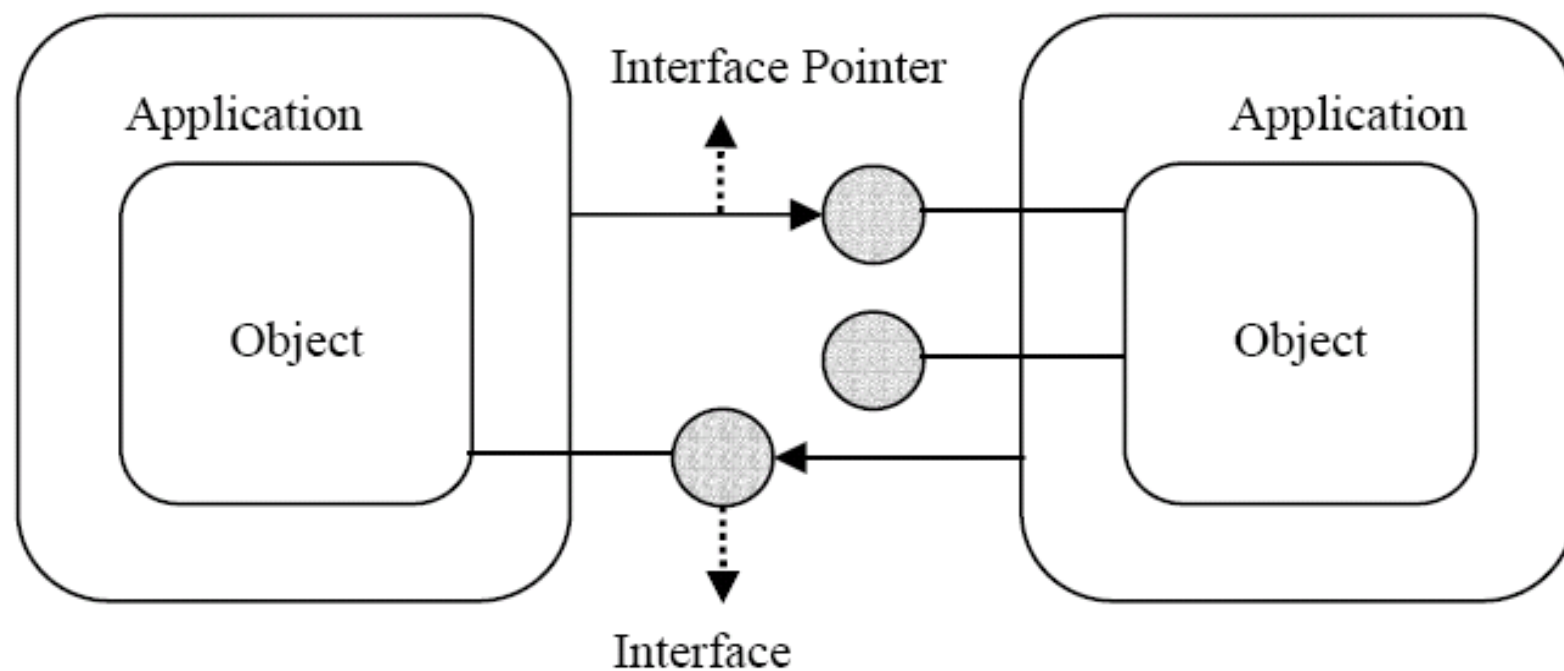
- Remote Method Invocation (RMI) is the action of invoking a method of a remote interface on a remote object. Most importantly, a method invocation on a remote object has the same syntax as a method invocation on a local object
- RMI is a lightweight due to its easy-to-use native Java model. It supports small-sized distributed object-based application development. A remote object has methods which can be invoked by another application located potentially on a different host.
- RMI is based on remote procedure call (RPC) implementation.

Remote Method Invocation (RMI)



COM

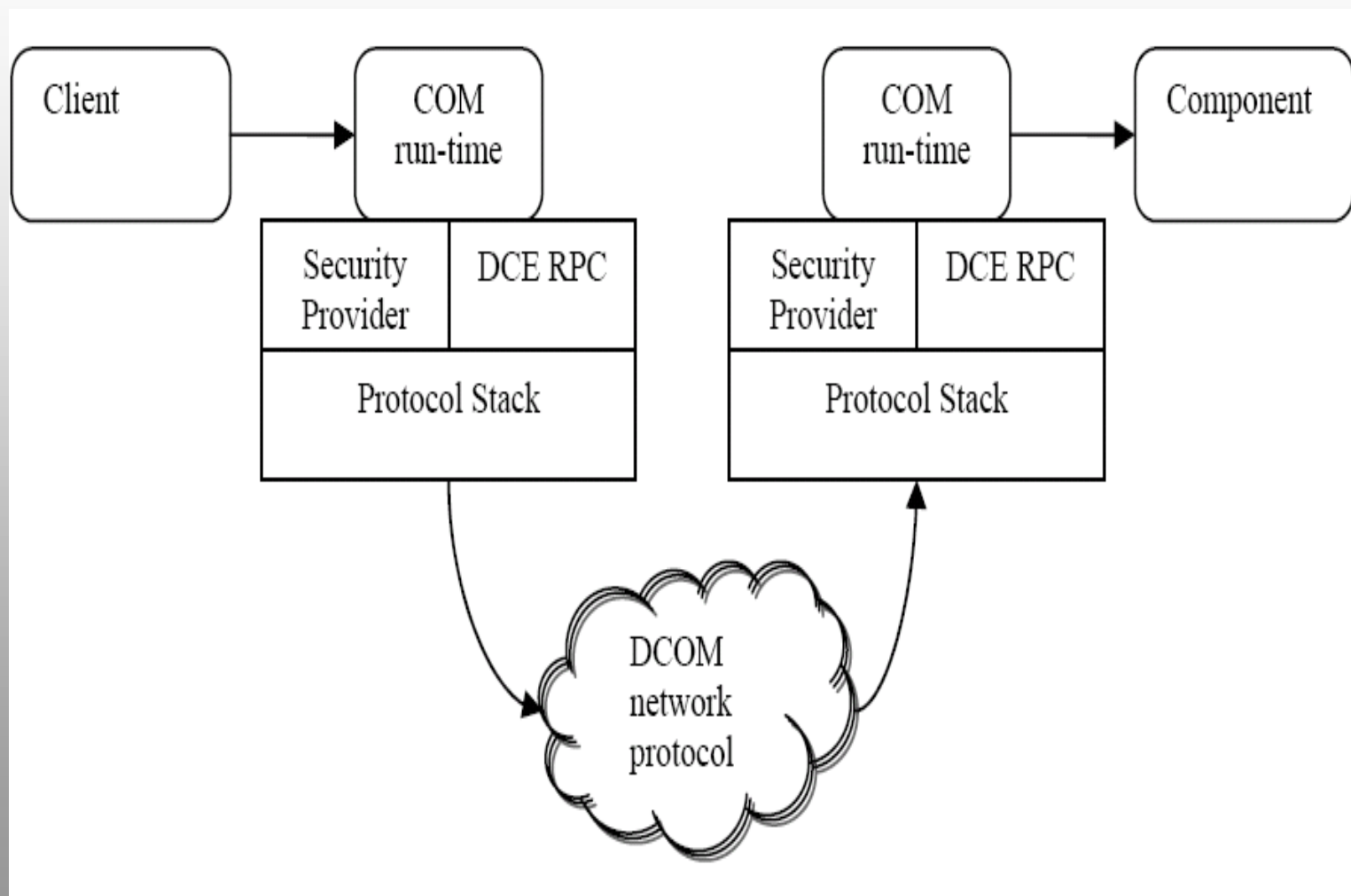
A COM object is defined as some piece of compiled code used for providing certain service. These objects is different than the objects defined in object-oriented programming,



DCOM

- Distributed Component Object Model (DCOM) extends COM so as to support communication among objects on different computers on a LAN, a WAN, and the Internet [18]. It allows the components to be used at distributed environment by handling the low-level details of network protocols.
- Architecturally DCOM provides cross-platform development. It allows the integration of platform-neutral development frameworks and virtual machine environments to build up a single distributed application.

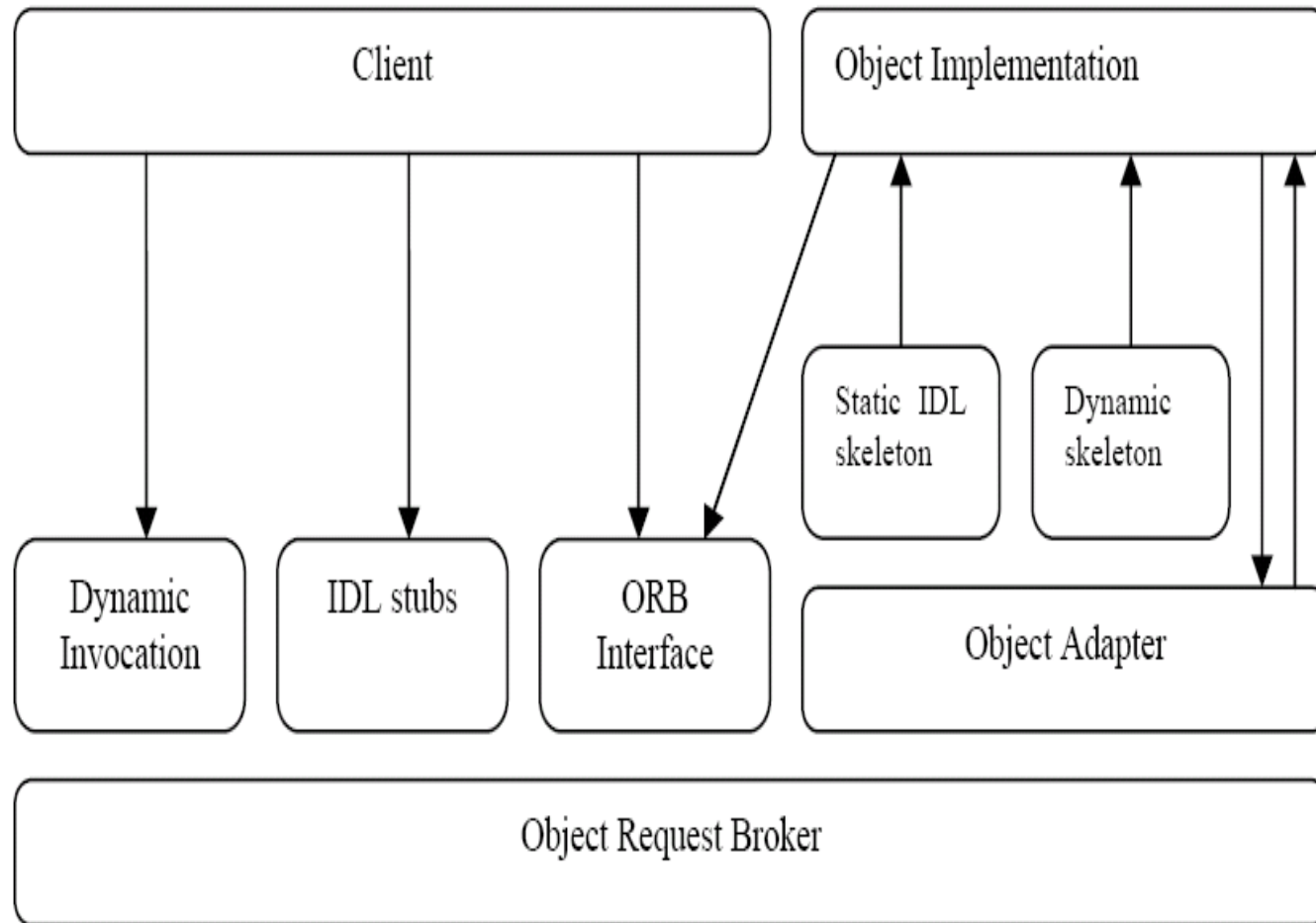
DCOM



CORBA

- Common Object Request Broker Architecture (CORBA) is an object system which provides a framework where objects can communicate with each other in a distributed manner without platform and language dependencies.
- The main component in CORBA's architecture is the Object Request Broker (ORB), which is a software component for insuring a proper communication of objects across heterogeneous environments.

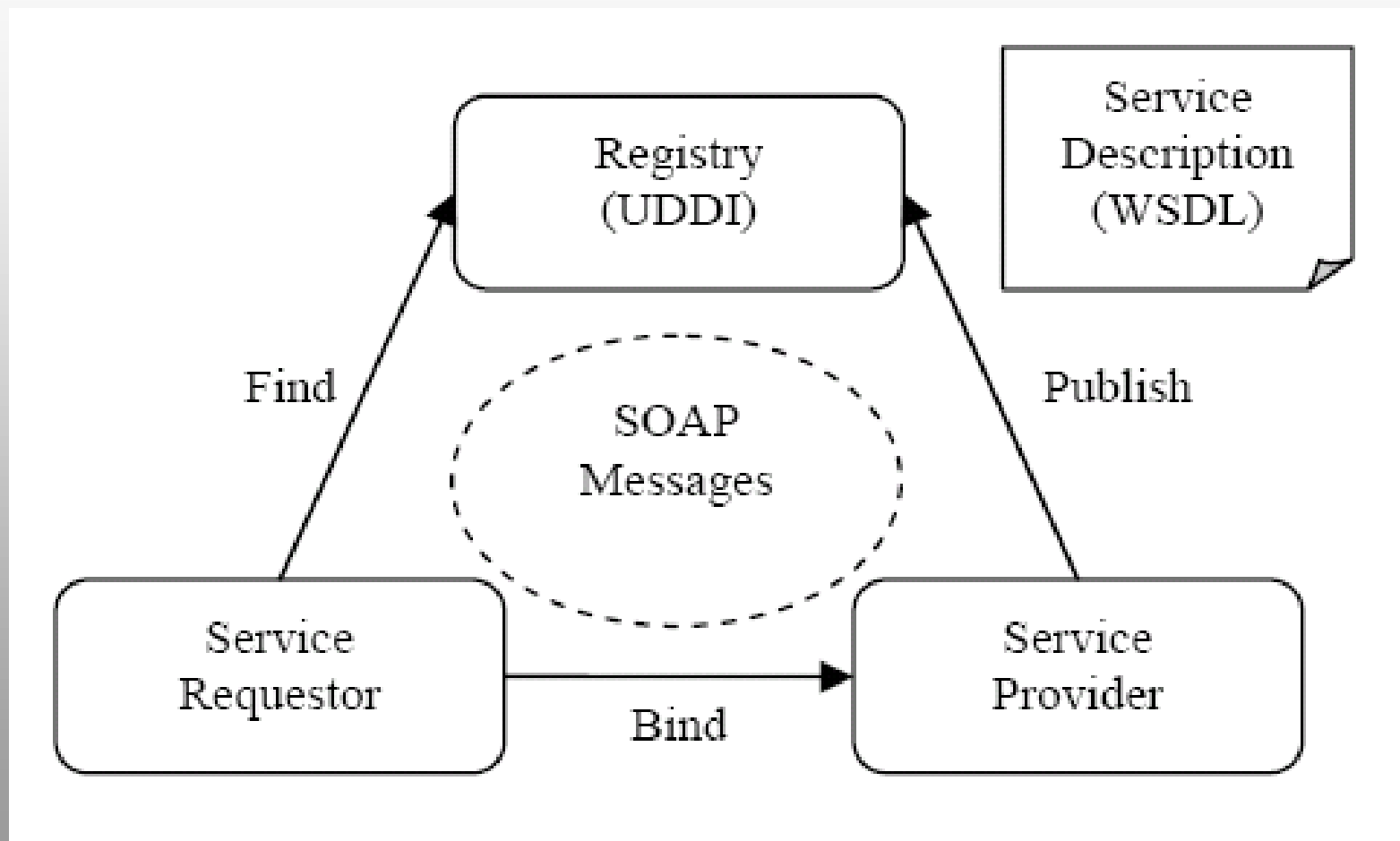
CORBA



Web Services

Web services are a set of XML-based technologies which aim to provide a standard way for communication of different applications and interoperability of heterogeneous computing environments. Web services use standard Internet technologies for messaging and data exchange, which makes them suitable for development of an application accessible in a platform-independent environment.

Web Services



	JMS	RMI	COM/DCOM	CORBA	Web services
Development Model	Object Oriented Messaging Development	Object Oriented Development	Object Oriented Component Development	Object Oriented Component Development	Component Oriented Service Development
Interface Definition Language	Java Programming Language	Java Programming Language	Microsoft Interface Definition Language	CORBA Interface Definition Language (IDL)	Web Service Definition Language (WSDL)
Coupling	Loose-Coupling	Tight-Coupling	Tight-Coupling	Tight-Coupling	Loose-Coupling
Platform Independence	Targets Java Platform	Targets Java Platform	Targets Microsoft Windows Platform	Platform Independent	Platform Independent
Interoperability, Support for Open Standards	Not Interoperable, Java Standards	Not Interoperable, Java Standards	Not Interoperable, Windows Platform Standards	Not Interoperable, CORBA Standards	Interoperable, Open Internet Standards
Location Transparency	Location Transparent through Message Provider, Vendor-specific	Location Transparent through RMI Naming Facility	Location Transparent through Interface Pointers	Location Transparent, CORBA IORs	Location Transparent, URL Addresses
Communication Mode	Asynchronous Mode, Possible to Implement Synchronous Mode	Synchronous Mode, RPC-based Development	Synchronous Mode, RPC-based Development	Synchronous Mode, RPC-based Development	Supports both Synchronous and Asynchronous Mode
Service Discovery and Registry Support	Registry of Messages by Message Provider	RMI Registry	Service Discovery through Interface Pointers, No Registry Support	Standard CORBA Services, CORBA Naming and Trading Service	Supports through UDDI
Security	J2EE-compatible security standards	J2EE-compatible security standards	Microsoft Windows-compatible security standards	CORBA Security Service	Ongoing-work for Standardization
Transactional Support	Partial Support	Supports	Supports	CORBA Object Transaction Service	Ongoing-work for Standardization

تهیه شده توسط:
امیر مهجوریان

مدیر فنی آزمایشگاه معماری سازمانی سرویس گرا
مدیر عامل شرکت دانش بنیان کاریز سیستم پویا



Mahjorian@IEAF.ir
Mahjorian@KarizSystem.ir