

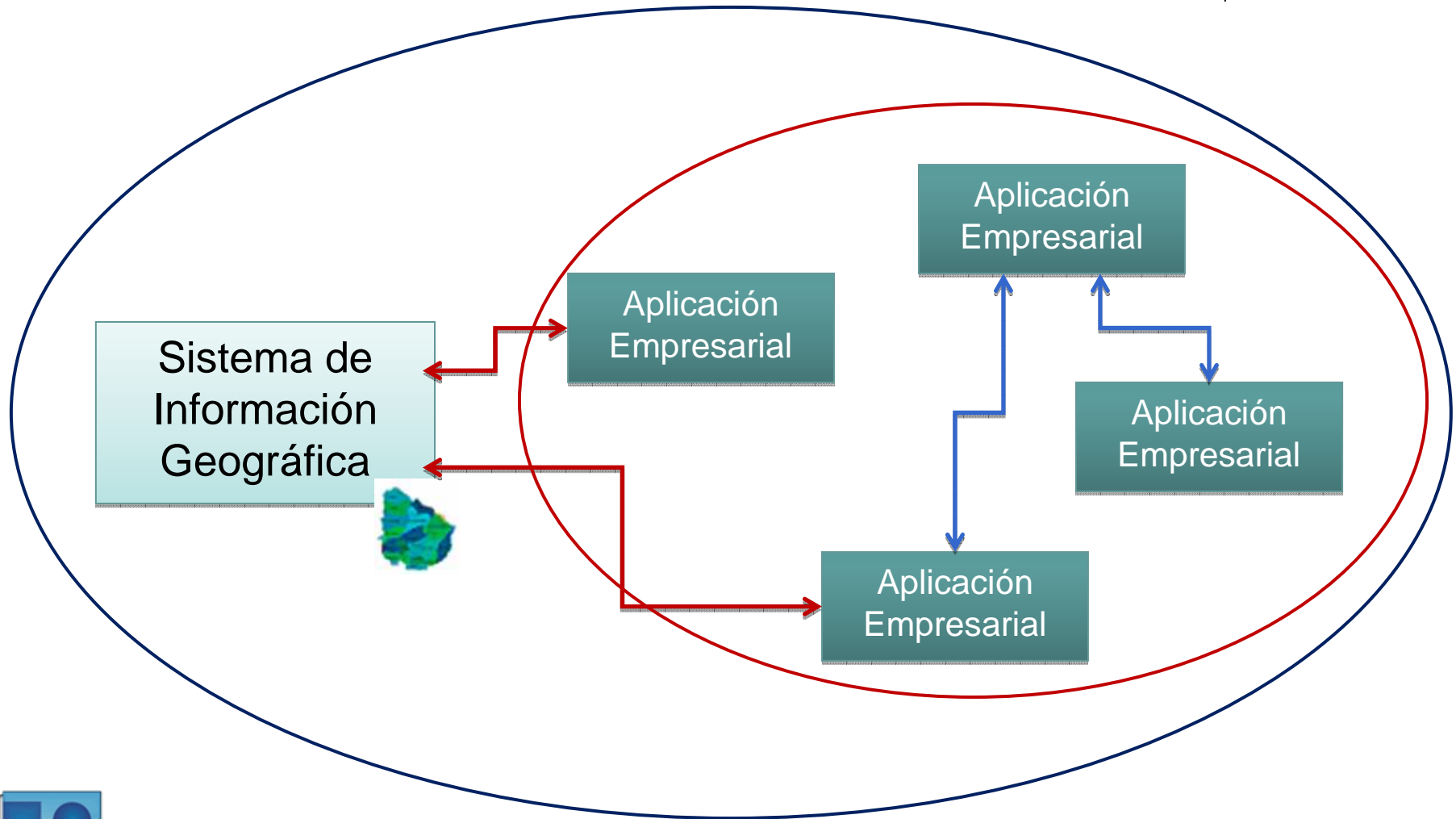
# Taller de Sistemas de Información Geográficos Empresariales



Clase 2



# Motivación



# Agenda

- ❑ Aplicaciones Empresariales
- ❑ Middleware
- ❑ Plataformas de Desarrollo Empresarial
  - .NET Framework
  - Java EE
- ❑ SOA y ESB
- ❑ Escenarios de Integración



# Taller de Sistemas de Información Geográficos Empresariales



Aplicaciones Empresariales



# Aplicaciones Empresariales

- ❑ Una **Aplicación Empresarial** es una aplicación de software desarrollada para administrar las operaciones, activos y recursos de una empresa
- ❑ Algunos ejemplos:
  - Contabilidad
  - Seguimiento de envíos
  - Servicio al cliente
  - Nómina de empleados



# Aplicaciones Empresariales

- ❑ Las aplicaciones empresariales tienen en general las siguientes características:
  - Involucran persistencia de datos
  - Se manejan grandes cantidades de datos
  - Existen varias interfaces de usuario, para distintos tipos de usuario
  - En general se deben integrar con otras aplicaciones
  - Se accede a los datos de forma concurrente



# Aplicaciones Empresariales

- ❑ El proceso de desarrollo de una aplicación empresarial involucra al menos:
  - Programadores de aplicaciones
  - Administradores de base de datos
  - Diseñadores de interfaz de usuario
  - Integradores de aplicaciones



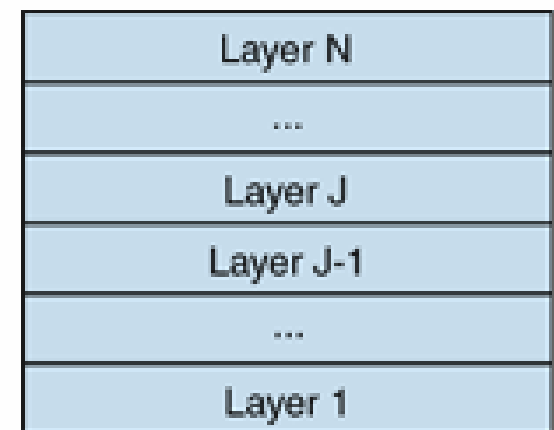
- ❑ La creación y mantenimiento de las aplicaciones presenta varias complejidades:
  - Administración
  - Mantenibilidad
  - Escalabilidad
  - Interoperabilidad
  - Seguridad
  - Confiabilidad
  - Accesibilidad y Usabilidad
  - Internacionalización





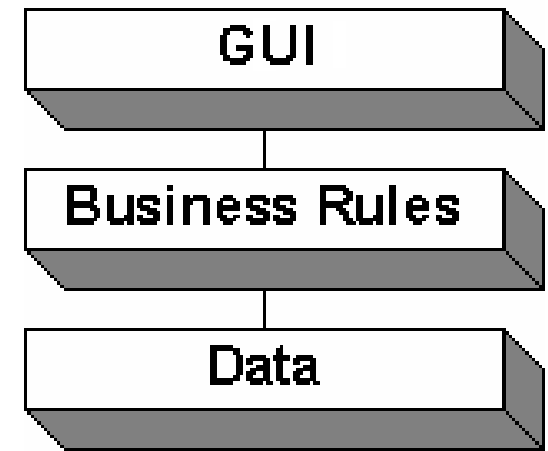
# Arquitectura en Capas (Layers)

- ❑ “Layers” es un estilo arquitectónico que comúnmente se utiliza para las Aplicaciones Empresariales
- ❑ En este esquema las capas más altas utilizan servicios definidos por las capas más bajas
- ❑ Esta división lógica entre capas de funcionalidad pueda basarse en distintas responsabilidades



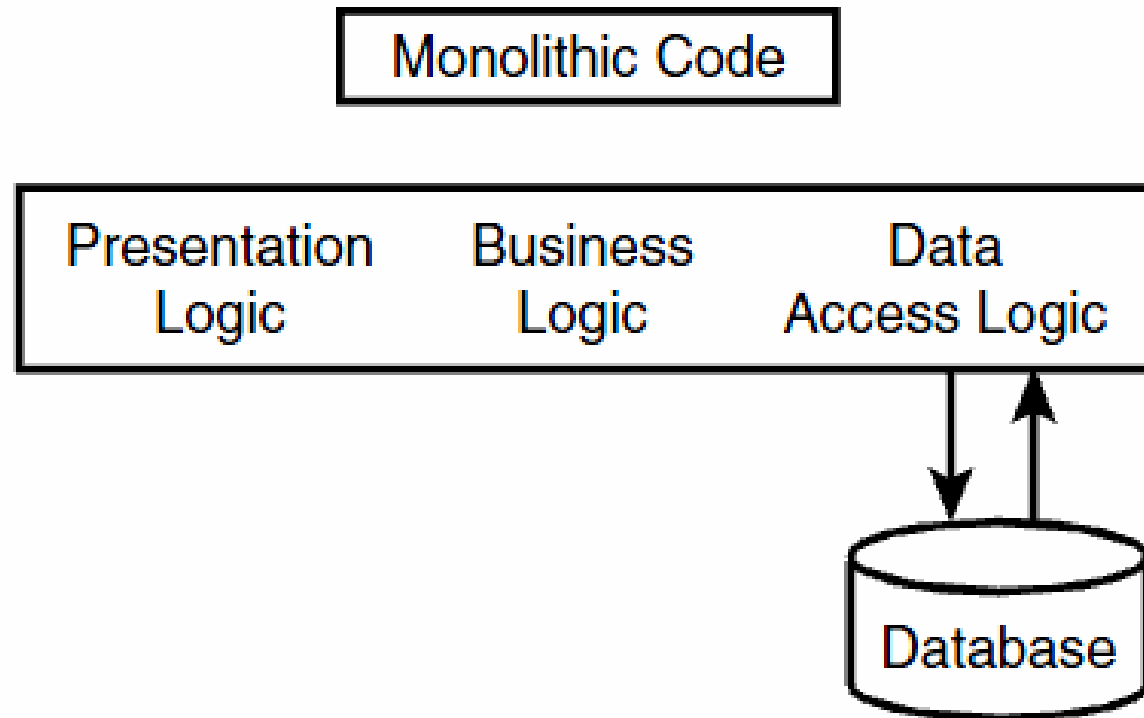
# Arquitectura en Capas (Layers)

- ❑ Tres capas comúnmente presentes en una Aplicación Empresarial son las capas de:
  - Presentación
    - Proveer servicios, presentar información, manejar pedidos de usuarios
  - Lógica de Negocio
    - Reglas que gobiernan los procesos de negocio objetivo de la aplicación
  - Datos
    - Comunicación con bases de datos, sistemas de mensajería, etc.

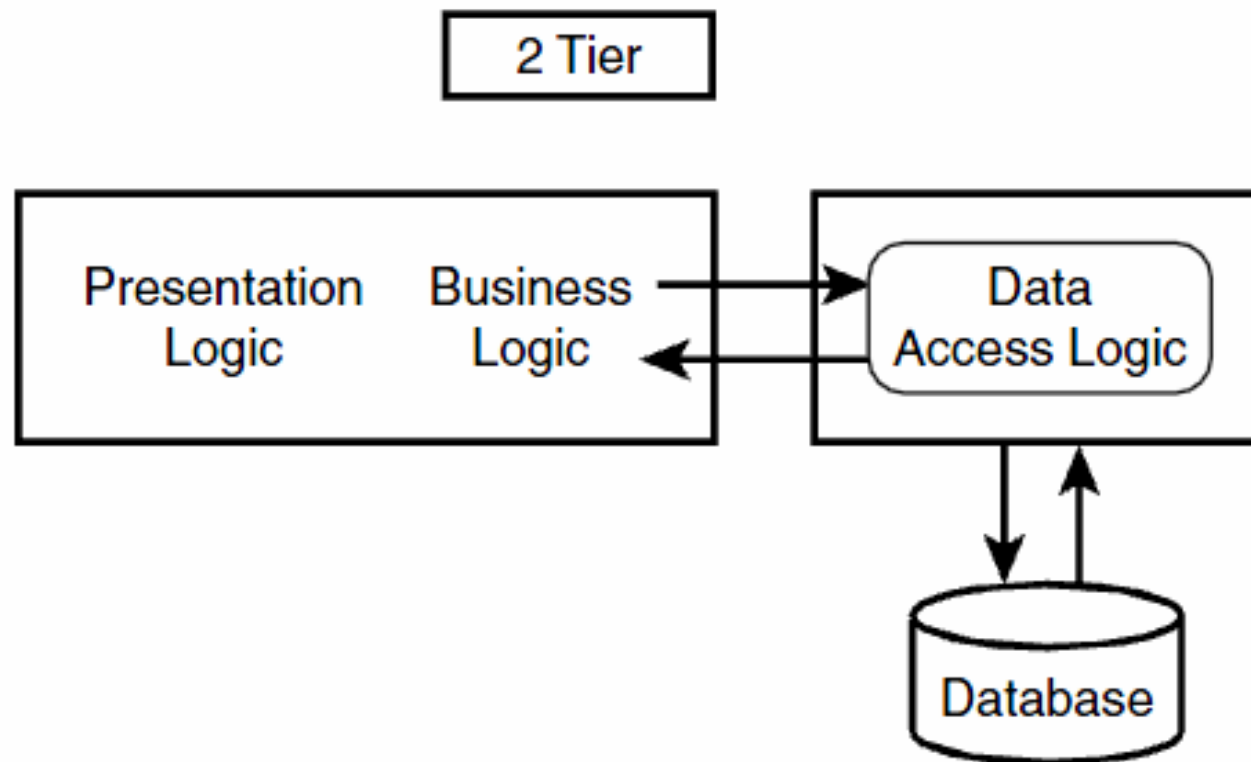


# Arquitectura Física (Tiers)

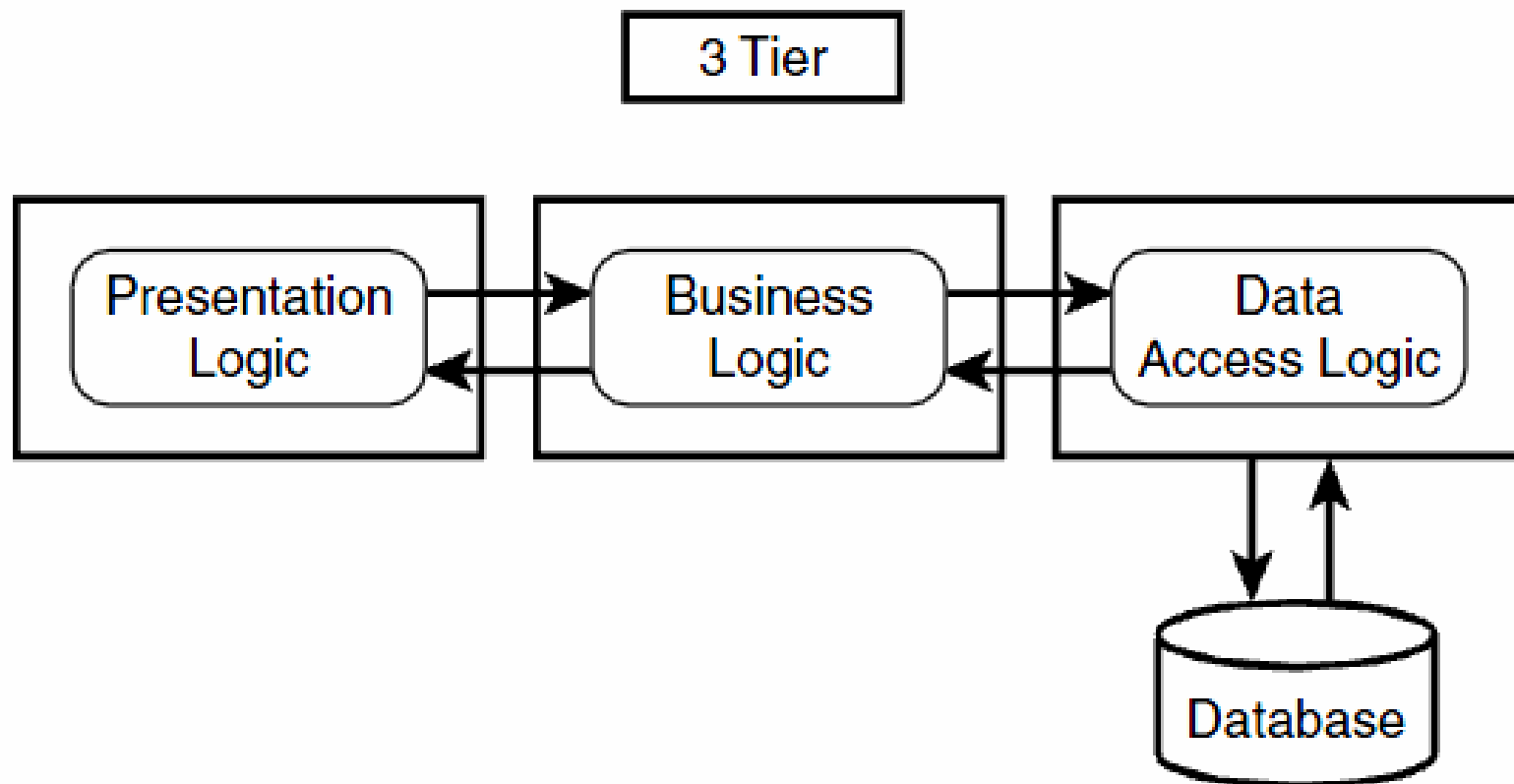
- ❑ Especifica la distribución física del sistema



# Arquitectura Física (Tiers)



# Arquitectura Física (Tiers)



- Integración de Aplicaciones Empresariales (EAI) es la tarea de hacer que aplicaciones desarrolladas de forma independiente trabajen de forma conjunta con el fin de compartir datos y procesos de negocio



- ❑ Al integrar Aplicaciones Empresariales surgen varios desafíos:
  - Las redes no son confiables
  - Las redes son lentas
  - Las aplicaciones son diferentes
    - a nivel de lenguajes de programación, formato de datos, etc
  - El cambio en las aplicaciones es inevitable



- ❑ Históricamente se han utilizado distintos enfoques para la integración:
  - Transferencia de archivos
  - Base de datos compartida
  - Invocación de procedimientos remotos
    - Comunicación sincrónica
  - Mensajería
    - Comunicación asincrónica





# Taller de Sistemas de Información Geográficos Empresariales

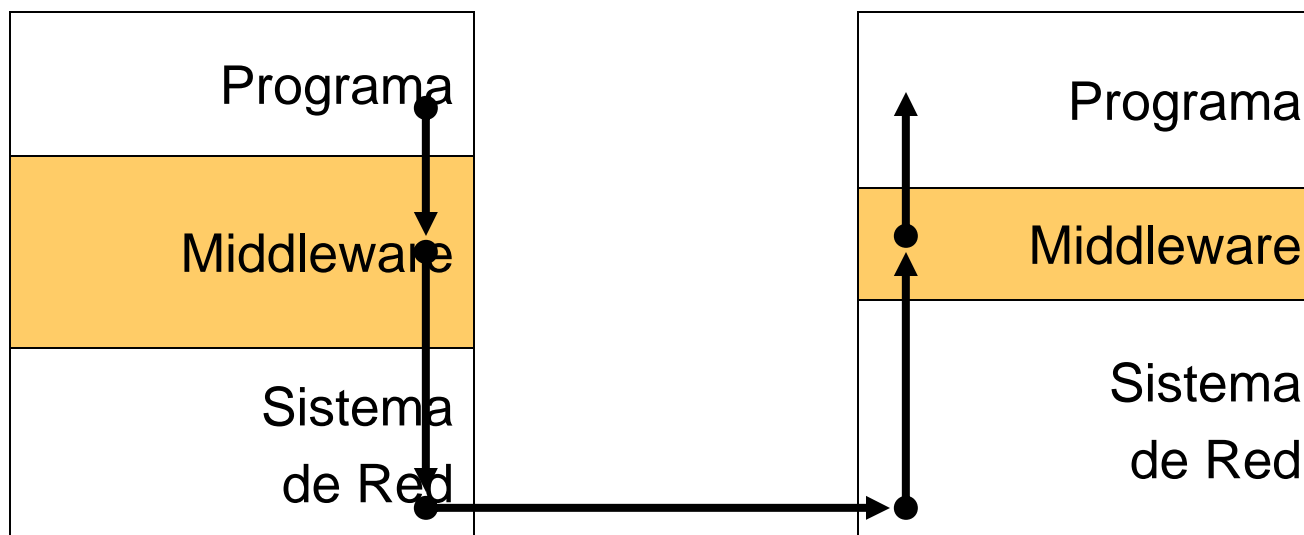


Middleware



# Middleware

- ❑ Middleware es una capa de software distribuida, situada entre el sistema operativo y las aplicaciones, diseñado para manejar la heterogeneidad y complejidad inherente a los sistemas distribuidos

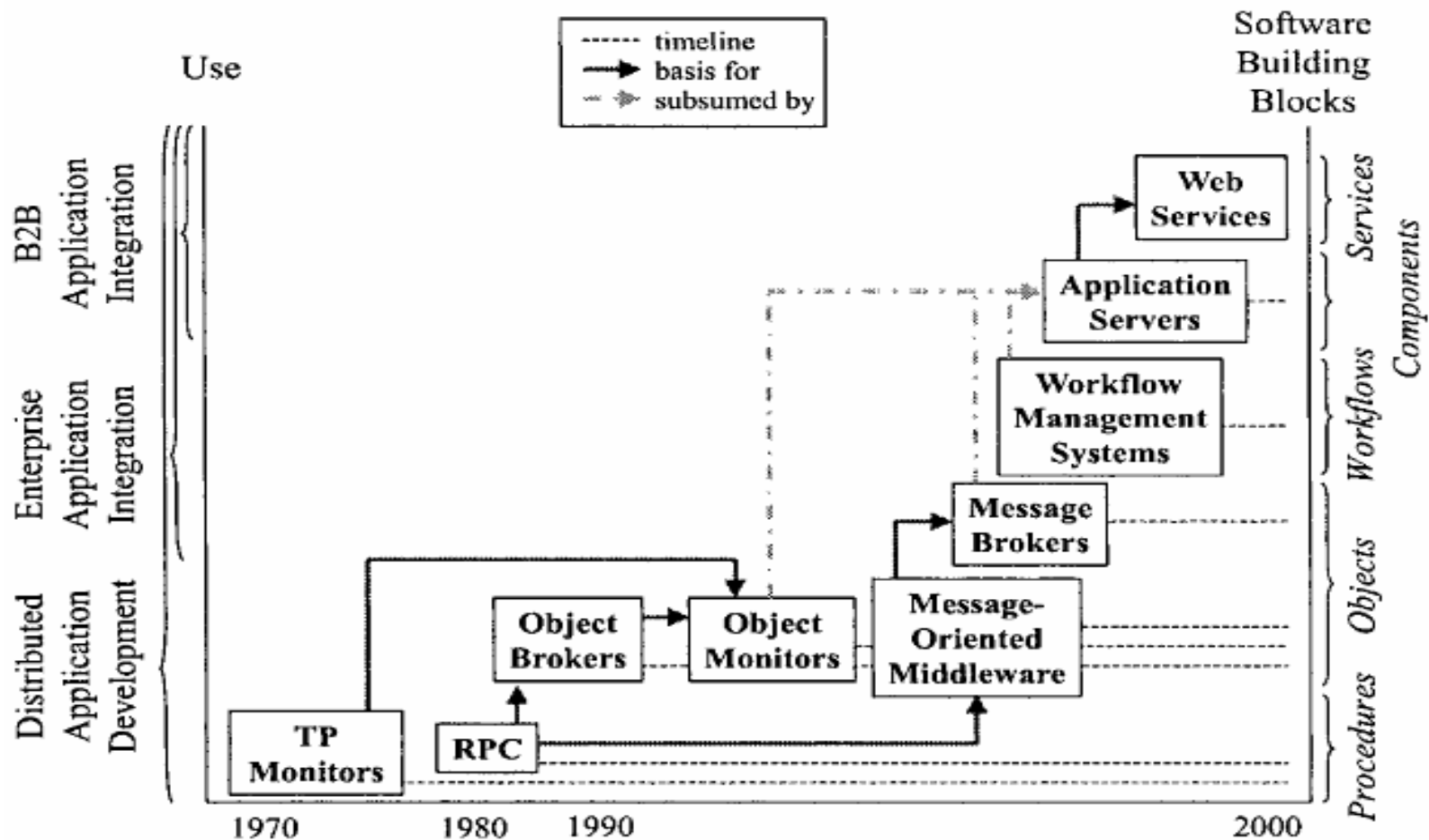


# Middleware

- ❑ El rol principal del middleware es facilitar la tarea de diseñar, programar, y administrar aplicaciones distribuidas, proveyendo un ambiente de programación distribuido simple, consistente e integrado.



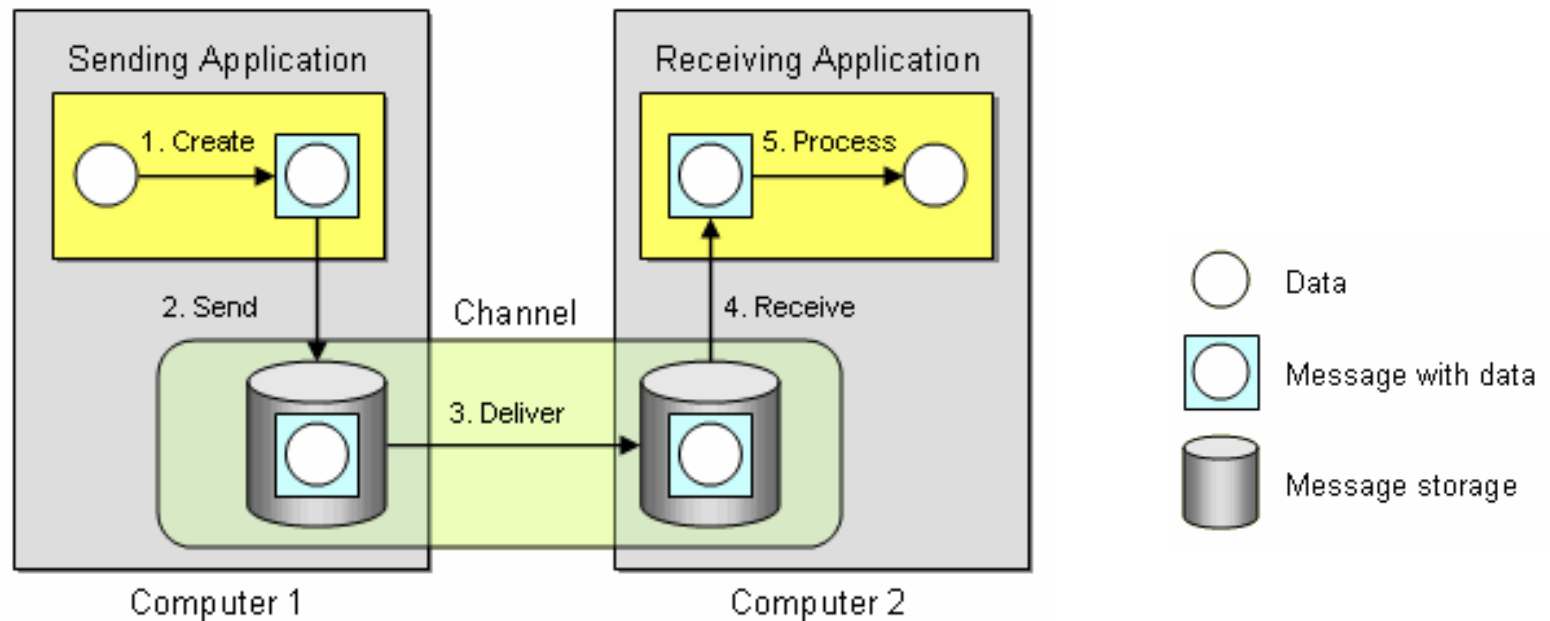
# Evolución Middleware



Semantic Management of Middleware. Ramesh Jain. Amit Sheth. Springer 2006.

# Message Oriented Middleware

- Los MOMs proveen comunicación asincrónica a través de mensajes, utilizando colas de mensajes para su almacenamiento temporal



# Message Oriented Middleware

- ❑ El principal objetivo de un MOM es transportar mensajes desde el equipo remitente al equipo receptor de una manera confiable
- ❑ Algunos Patrones de Mensajería
  - Point to point
  - Request – Response
  - Request – Callback
  - Publish - Subscribe



# Application Servers

- ❑ Los servidores de aplicaciones proveen mecanismos para manejar toda o la mayoría de las interacciones entre los componentes de una aplicación distribuida
- ❑ Proveen varias tecnologías de middleware (MOMs, etc) junto con el concepto de contenedor, que brinda un entorno de ejecución para los componentes de una aplicación



# Application Servers

- ❑ En general se puede encontrar soporte para seguridad, transacciones, administración de aplicaciones y recursos, y balanceo de carga
- ❑ Proveen una solución completa para la construcción e integración de aplicaciones empresariales



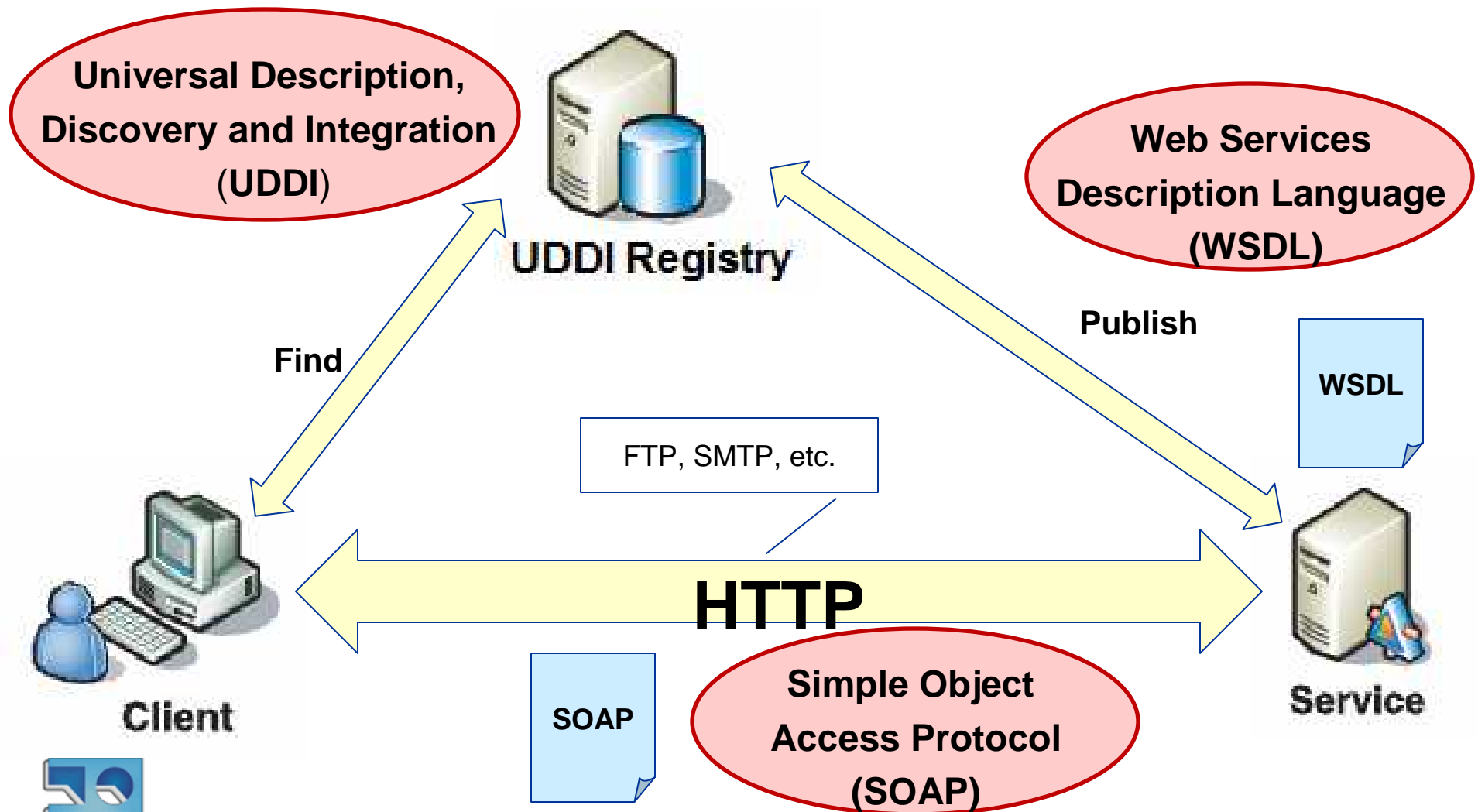


- Un Web Service es una **aplicación de software identificada por una URI**, cuyas interfaces y formas de acceso pueden ser **definidas, descritas y descubiertas como artefactos XML**, y soporta la interacción directa con otros componentes de software utilizando **mensajes basados en XML**, intercambiados a través **de protocolos basados en internet**

<http://www.w3.org/TR/ws-desc-reqs/#definitions>



# Primera Generación de WS



- ❑ Provee una forma estándar de estructurar mensajes utilizando XML
- ❑ Define mecanismos para utilizar distintos protocolos de transporte para el envío de mensajes
- ❑ Especifica un modelo de procesamiento que indica cómo se deben procesar los mensajes



# Mensaje SOAP

```
<?xml version="1.0"?>
```

```
<soap:Envelope
```

```
  xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
```

```
  soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

```
    <soap:Header>
```

```
      ...
```

```
    </soap:Header>
```

```
    <soap:Body>
```

```
      ...
```

```
      <soap:Fault> ... </soap:Fault>
```

```
    </soap:Body>
```

```
</soap:Envelope>
```



- ❑ Lenguaje basado en XML que permite describir la interfaz y otras características de un Web Service
- ❑ Un documento WSDL puede dividirse en dos partes:
  - descripción abstracta
  - descripción concreta



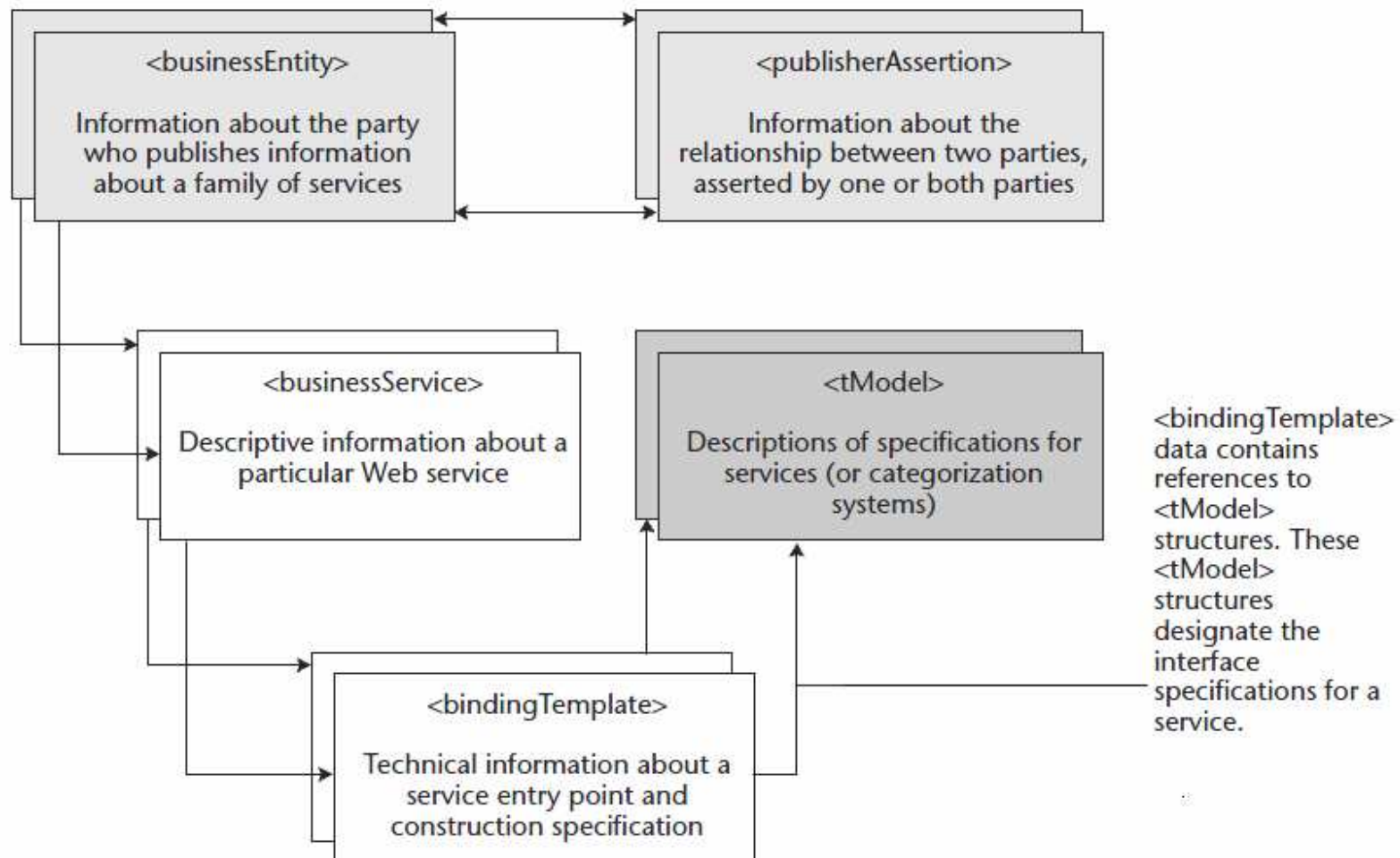
- ❑ La descripción abstracta describe de forma general la estructura de la interfaz del Web Service, que incluye operaciones, parámetros y tipos de datos abstractos
- ❑ La descripción concreta asocia a una descripción abstracta una dirección de red concreta, un protocolo de comunicación y estructuras de datos concretas



- ❑ Especificación que provee una forma estándar de publicar y descubrir Web Services
- ❑ UDDI define
  - un modelo de datos para almacenar información de servicios y negocios
  - dos interfaces para utilizar el registro UDDI
    - Inquiry
    - Publish



# UDDI – Modelo de Datos





# Segunda Generación de WS

- ❑ Surgen como forma de abordar problemáticas comunes en contextos empresariales
- ❑ Se les conoce como WS-\*
- ❑ Cada una aborda una problemática específica:
  - Seguridad, Transacciones, Mensajería, etc



- ❑ *Web Services Business Process Execution Language* es un lenguaje para “orquestrar” Web Services
- ❑ *WS-BPEL es un lenguaje de flujo basado en XML para la especificación formal de procesos de negocio y protocolos de interacción de negocio*



# Mensajería – WS-Addressing

- ❑ WS-Addressing (WS-A) provee un mecanismo estándar para direccionar mensajes y Web Services
- ❑ Define dos construcciones básicas
  - *endpoint reference*
    - Address, Reference Parameters, Metadata
  - addressing properties
    - To, From, ReplyTo, FaultTo,
    - Action, MessageID, RelatesTo
    - ReferenceParameters



# Metadata – WS-Policy

- ❑ Define un modelo abstracto, independiente del dominio, que permite describir características, requerimientos y capacidades de un Web Service
- ❑ Delega a otras especificaciones la definición de políticas particulares a un dominio.
  - WS-SecurityPolicy
  - WS-ReliableMessagingPolicy



- ❑ Transacción Atómica: WS-AtomicTransaction
  - Propiedades ACID
  - Corta Duración
  - Ambiente seguro
  - Diseñado principalmente para dar soporte a la interoperabilidad
- ❑ Actividad de Negocio: WS-BusinessActivity
  - Larga Duración
  - Se define un mecanismo de compensación



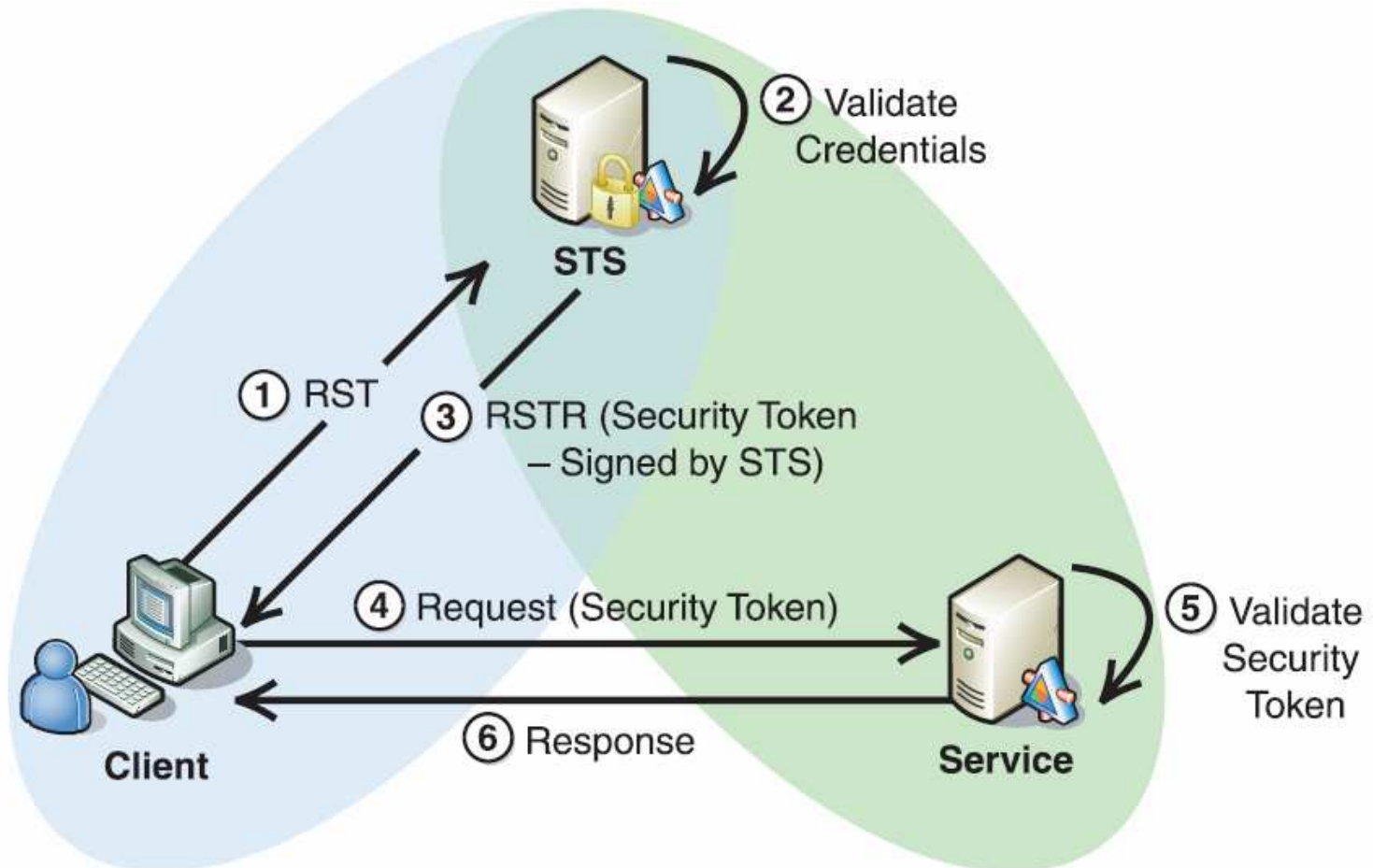
- ❑ Define un conjunto de extensiones SOAP para brindar seguridad a nivel de mensaje
- ❑ Se especifica cómo:
  - utilizar XML Signature en mensajes SOAP
  - utilizar XML Encryption en mensajes SOAP
  - incluir Tokens de Seguridad en mensajes SOAP



- ❑ WS-Trust es una especificación que extiende WS-Security y provee:
  - métodos para la expedición, renovación y validación de tokens de seguridad
  - formas para establecer, evaluar y administrar relaciones de confianza



# WS-Trust



Microsoft Corporation. Web Service Security Scenarios, Patterns, and Implementation Guidance for Web Services Enhancements (WSE) 3.0. 2005. <http://msdn.microsoft.com/en-us/library/aa480545.aspx>



# Especificaciones de WS

- ❑ Actualmente la tecnología de Web Services está basada en un gran número de especificaciones que:
  - en general, son propuestas por la industria
    - Microsoft, IBM, Oracle, etc.
  - son estandarizadas por distintas organizaciones
    - W3C, OASIS, etc.
  - son implementadas por distintos proveedores
    - Apache, JBoss, Sun, Microsoft, IBM, Oracle, etc.



# Web Services REST

- ❑ REST (REpresentational State Transfer)
  - Estilo arquitectónico para sistemas de hipermedia distribuidos
  - Todo es tratado como recursos que se identifican por URIs
  - Toma ventaja de los verbos HTTP
    - GET, POST, PUT, DELETE



# Web Services REST

- ❑ La intención de una llamada a un RESTful Service, se obtiene del verbo HTTP
  - GET (recuperar), DELETE (eliminar)...

Verbo HTTP	Significado en términos de CRUD (Create, Read, Update, Delete)
POST	Crear un nuevo recurso a partir de los datos de la solicitud.
GET	Leer un recurso.
PUT	Actualizar un recurso a partir de los datos de la solicitud.
DELETE	Eliminar un recurso.



# Web Services REST

- De este modo las URIs actúan como identificadores de recursos y los métodos HTTP como verbos que especifican operaciones sobre los mismos

Verbo HTTP / URI	Significado en términos de CRUD
POST emps	Crear un nuevo empleado a partir de los datos de la solicitud.
GET emps	Leer una lista de todos los empleados.
GET emps?id=27	Leer el empleado 27.
PUT emps	Actualizar la lista de empleados con los datos de la solicitud.
DELETE emps	Eliminar la lista de empleados.
DELETE emps?id=27	Eliminar el empleado 27.



# Taller de Sistemas de Información Geográficos Empresariales



Plataformas de Desarrollo Empresarial



# .NET Framework

- ❑ Es un Framework desarrollado por Microsoft, disponible en plataforma Windows
- ❑ Incluye
  - Una biblioteca de clases orientada al programador a fin de facilitar los problemas típicos de programación
  - Una maquina virtual que administra la ejecución de programas escritos para esta plataforma



# .NET Framework

- ❑ La biblioteca de clases provee una gran variedad de funcionalidades, entre las que se incluyen
  - Interfaz de usuario
  - Acceso a datos
  - Conectividad
  - Aplicaciones web
  - Seguridad



# .NET Framework

- ❑ Los programas escritos para el framework .NET ejecutan en un ambiente de software que administra los requerimientos de dicho programa
- ❑ Este ambiente de ejecución, se denomina Common Language Runtime





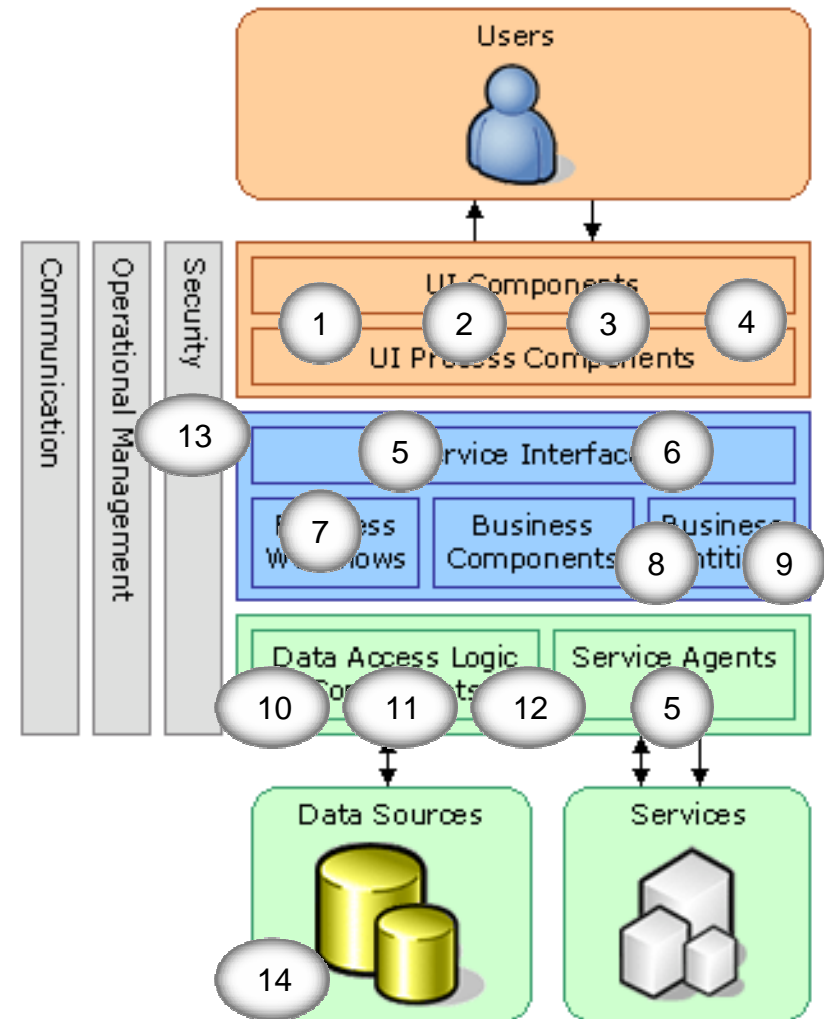
# Common Language Runtime

- ❑ Otros servicios que provee
  - Seguridad
  - Manejo de memoria
  - Control de excepciones
  - Performance



# .Net Framework

1. ASP.NET
2. AJAX
3. WPF
4. Silverlight
5. WCF
6. WS\*
7. Workflow Foundation
8. Datatypes
9. Datasets
10. ADO.NET
11. LINQ
12. ORM (NHibernate)
13. Membership
14. SQL Server



# Java Enterprise Edition

- ❑ Definición de Sun Microsystems
  - Java Platform, Enterprise Edition 5 (Java EE 5) define el estándar para el desarrollo de aplicaciones empresariales distribuidas, basadas en componentes, utilizando un modelo de múltiples capas



# Independencia del proveedor

- ❑ La plataforma promueve la construcción de sistemas independientes de la plataforma
  - Heredado de Java
- ❑ La especificación es abierta, puede ser implementada por cualquier proveedor
- ❑ Este deberá cumplir dicho estándar
  - Hay procesos de certificación
  - No implica que sólo debe soportar lo que el estándar establece

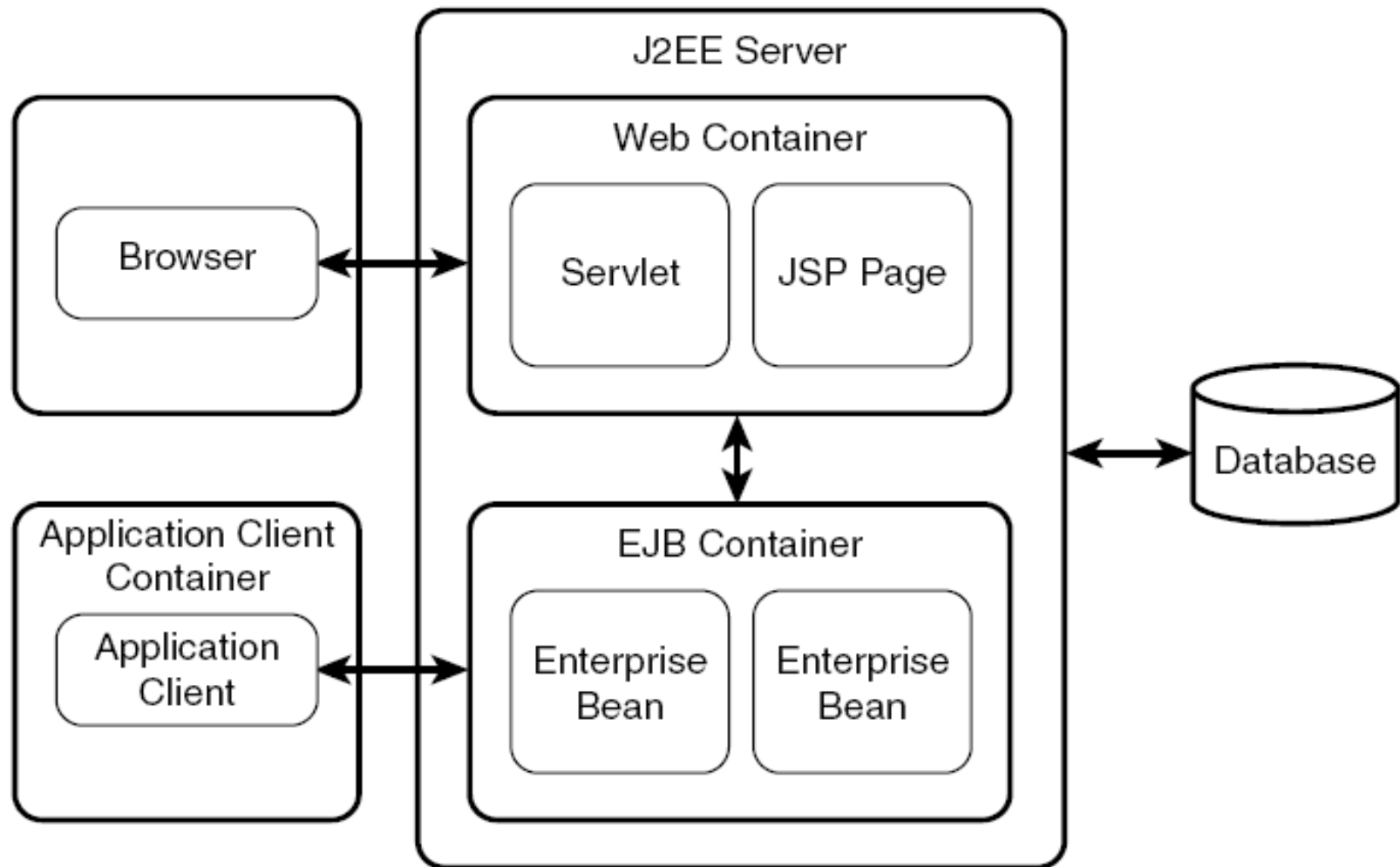


# Servidores Java EE

- ❑ Representa el ambiente en el que ejecutan los componentes Java EE
- ❑ Estos componentes se denominan componentes server-side o componentes de aplicación JEE
- ❑ Pueden tomar la forma de
  - Componentes web (JSP / Servlets / JSF)
  - Componentes de negocio (EJB)
- ❑ Estos componentes ejecutan en un runtime denominado **contenedor**



# Servidores Java EE



# Contenedores Java EE

- ❑ Los componentes web y de negocio, existen y ejecutan dentro de contenedores
- ❑ Los componentes de aplicación JEE nunca interactúan directamente entre sí
  - utilizan protocolos y métodos del contenedor para interactuar entre ellos y con servicios de la plataforma
  - este rol de intermediario le permite al contenedor inyectar servicios requeridos por los componentes



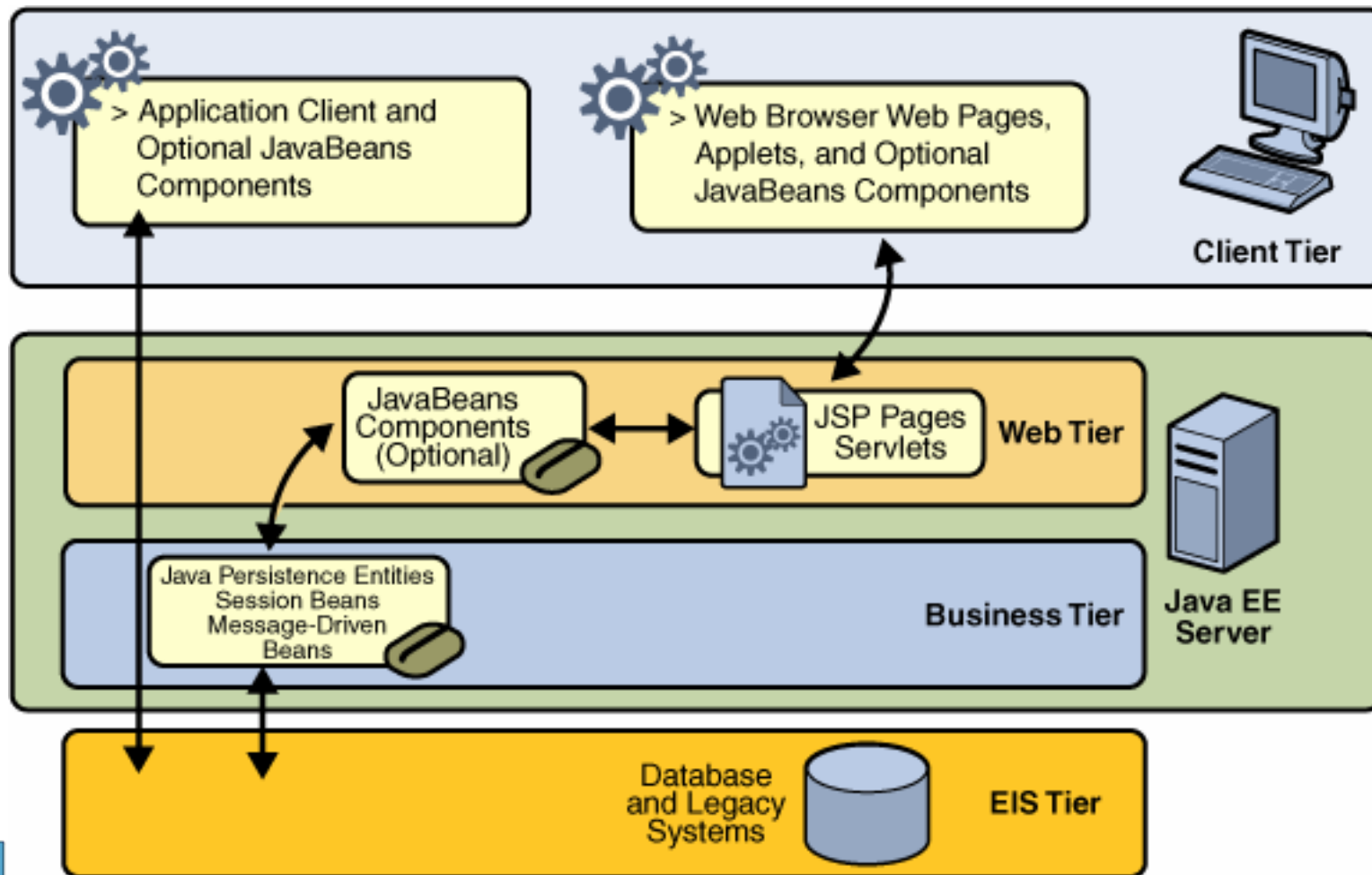
# Contenedores Java EE

- ❑ Un contenedor permite a los componentes interactuar con los servicios brindados por el servidor de aplicaciones
  - Seguridad
  - Acceso a datos
  - Transacciones
  - Acceso a recursos
  - Comunicaciones



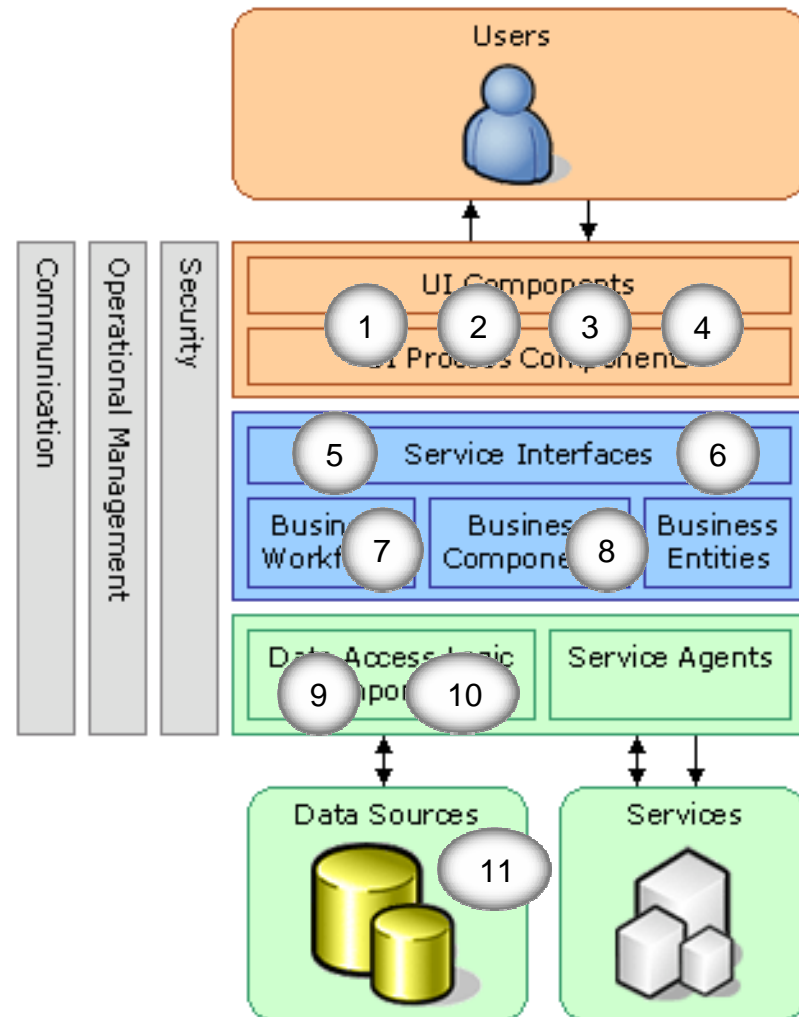


# Componentes



# Java EE

1. Java Server Faces
2. Flex
3. Granite
4. AJAX
5. JAX-WS
6. WS\*
7. jBPM
8. EJB3
9. Java Persistence API
10. ORM (Hibernate)
11. PostgreSQL



# Taller de Sistemas de Información Geográficos Empresariales



Arquitecturas Orientadas a  
Servicios y Enterprise Service  
Bus



- ❑ Computación Orientada a Servicios
  - paradigma que basa el diseño de aplicaciones en servicios para dar soporte al desarrollo ágil y flexible de aplicaciones distribuidas en ambientes heterogéneos
- ❑ Un Servicio es una entidad de cómputo que expone una funcionalidad de negocio y es:
  - autónoma
  - independiente de la plataforma
  - puede ser descripta, publicada, descubierta y combinada

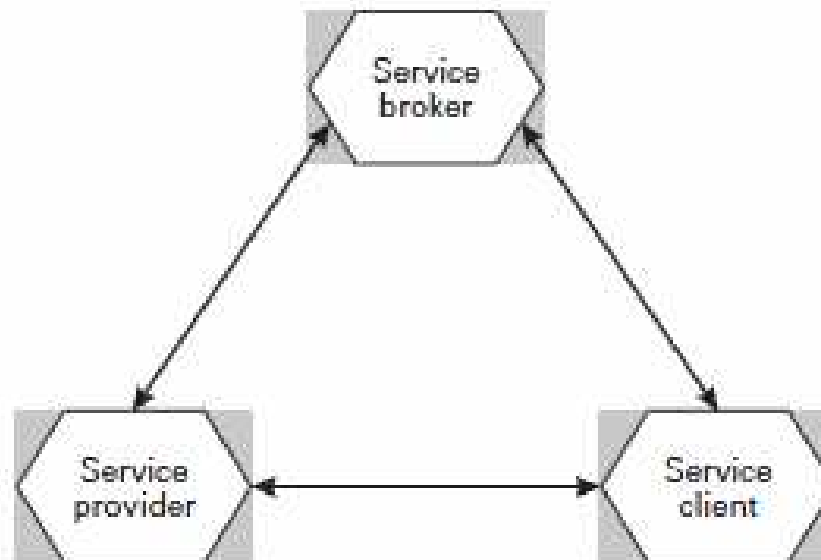


## □ Principios

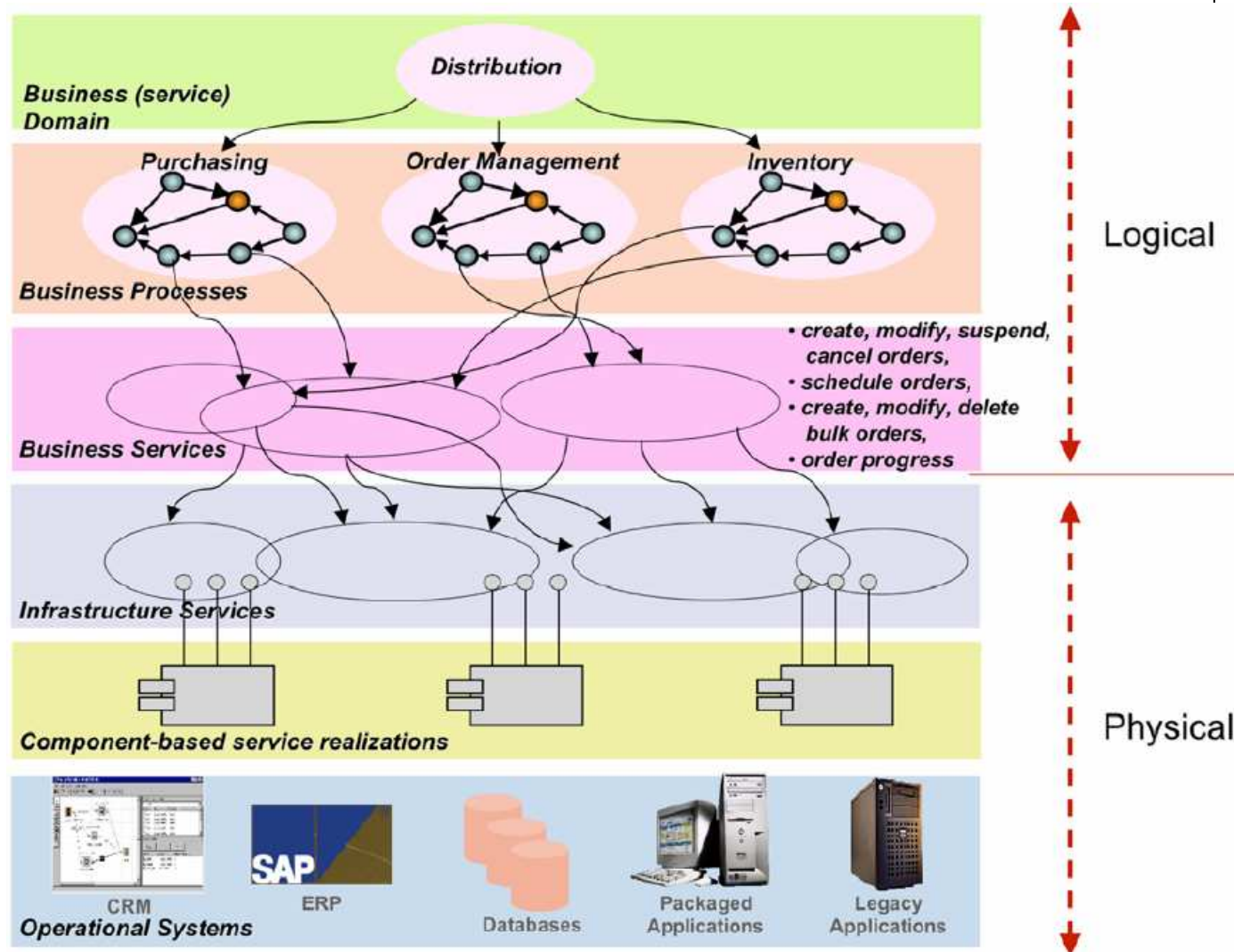
- Standardized Service Contracts
- Service Loose Coupling
- Service Abstraction
- Service Reusability
- Service Autonomy
- Service Statelessness
- Service Discoverability
- Service Composability



- Arquitectura Orientada a Servicios
  - forma lógica de diseñar sistemas de software para proveer servicios a través de interfaces públicas y descubribles



# Service Oriented Architecture



# Service Oriented Architecture

- ❑ Si bien los principios de SOA no dependen de una tecnología en particular, los Web Services se han convertido en el mecanismo preferido para su implementación
- ❑ Actualmente, la forma más común de proveer una infraestructura de integración administrable, para Web Services y SOA, es a través de un ESB



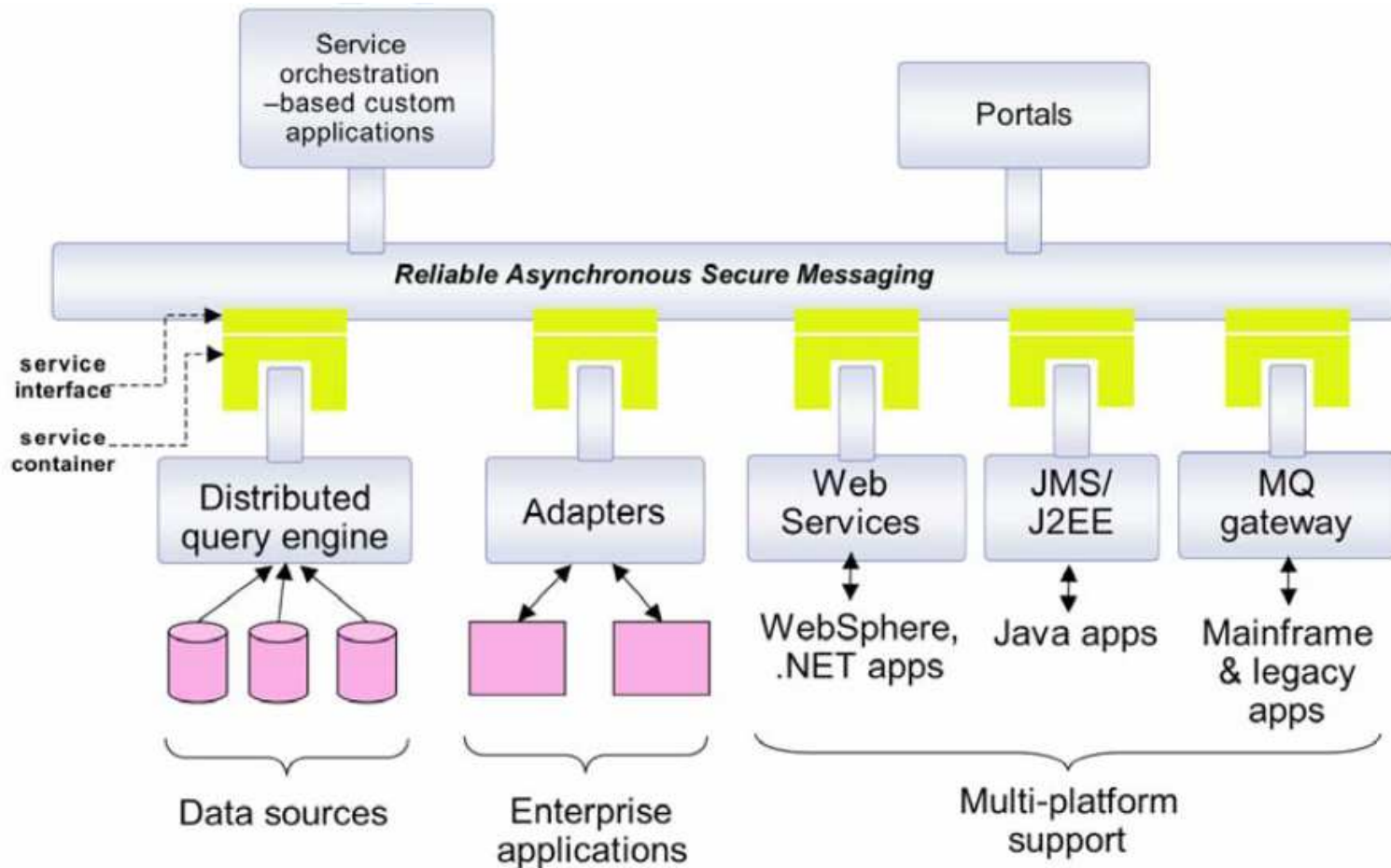


# Enterprise Service Bus (ESB)

- Un ESB es una plataforma de integración basada en estándares que combina mensajería, Web Services, transformación de datos, y ruteo inteligente, para conectar y coordinar de forma confiable la interacción de un gran número de aplicaciones a través de empresas con integridad transaccional

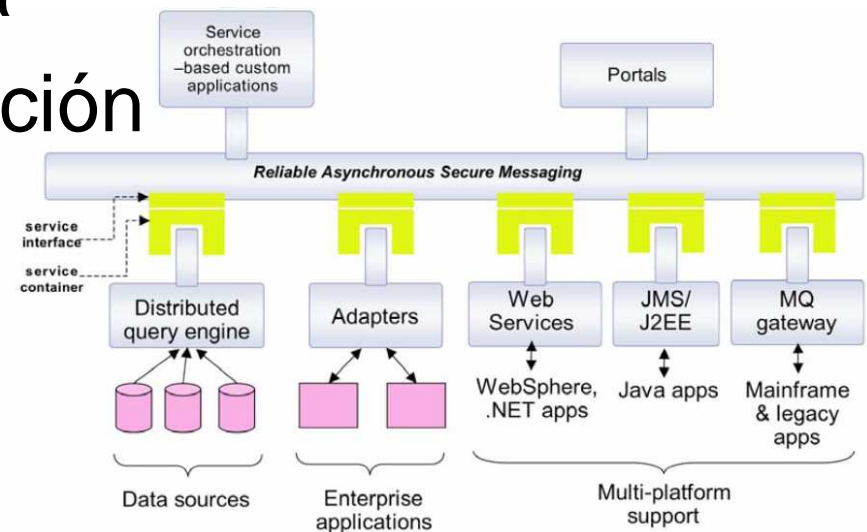


# Enterprise Service Bus



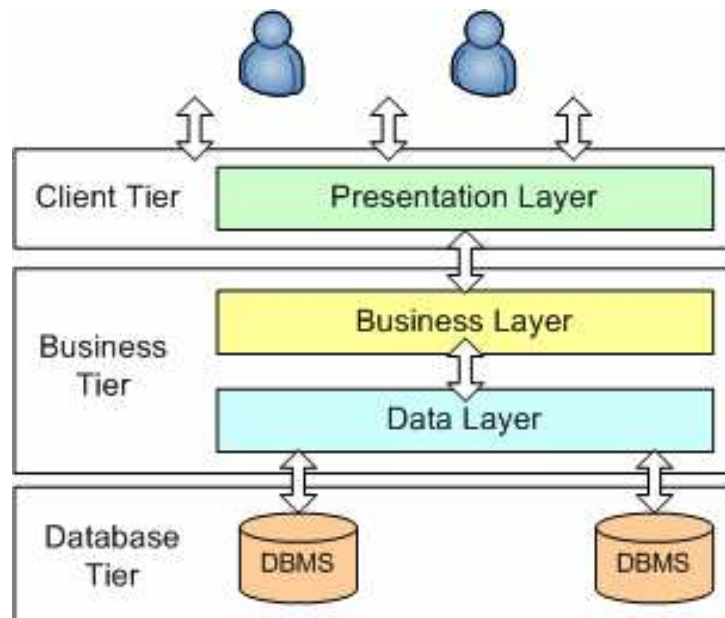
# Funcionalidades de ESB

- ❑ Conectividad / Adaptadores
- ❑ Transformación de Mensajes
- ❑ Ruteo Intermediario
- ❑ Flujos de Mediación
- ❑ Mensajería Asincrónica
- ❑ Monitoreo y Administración
- ❑ Otras...

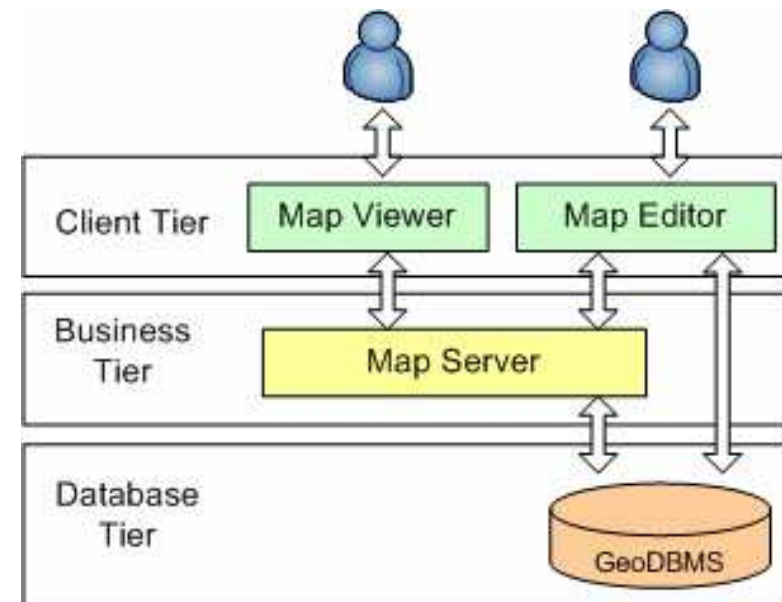


# Escenarios de Integración

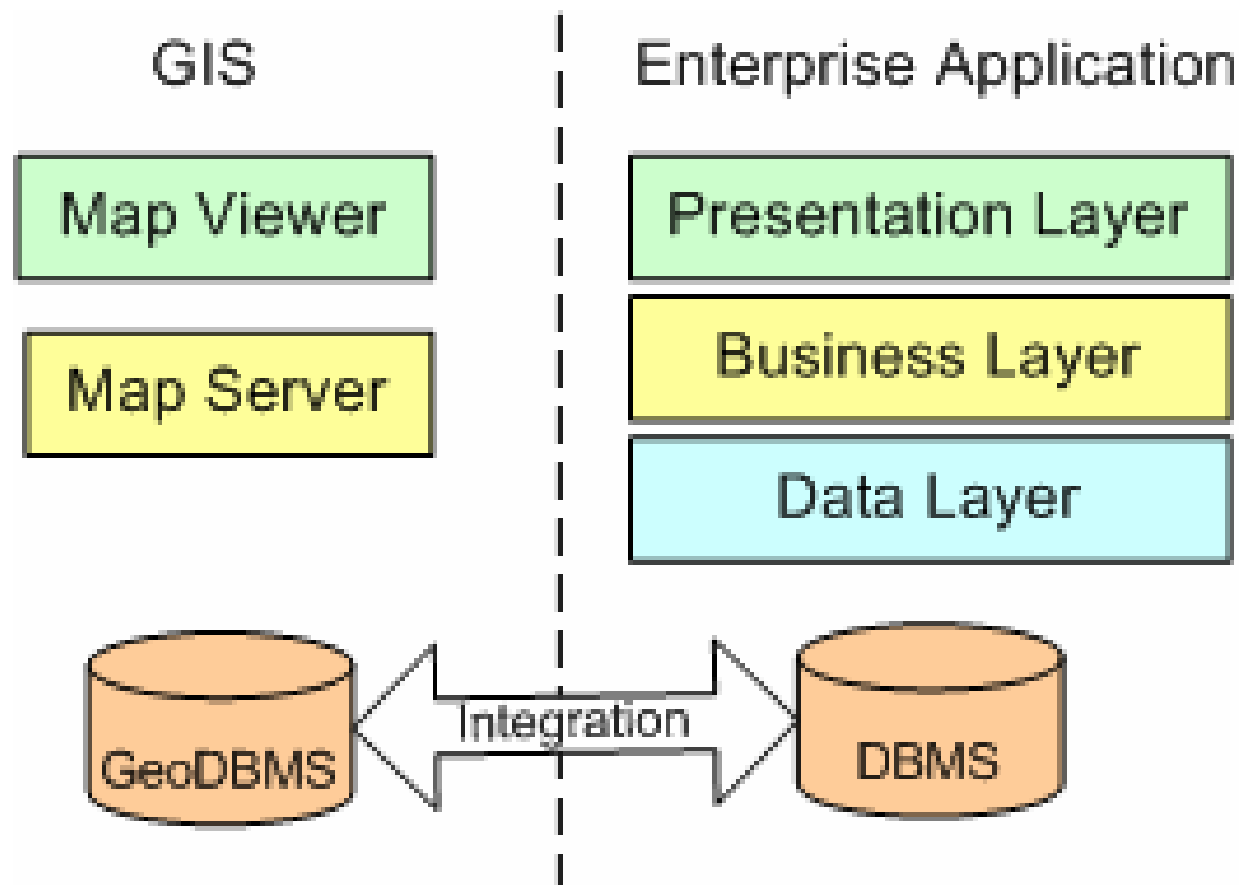
## GIS



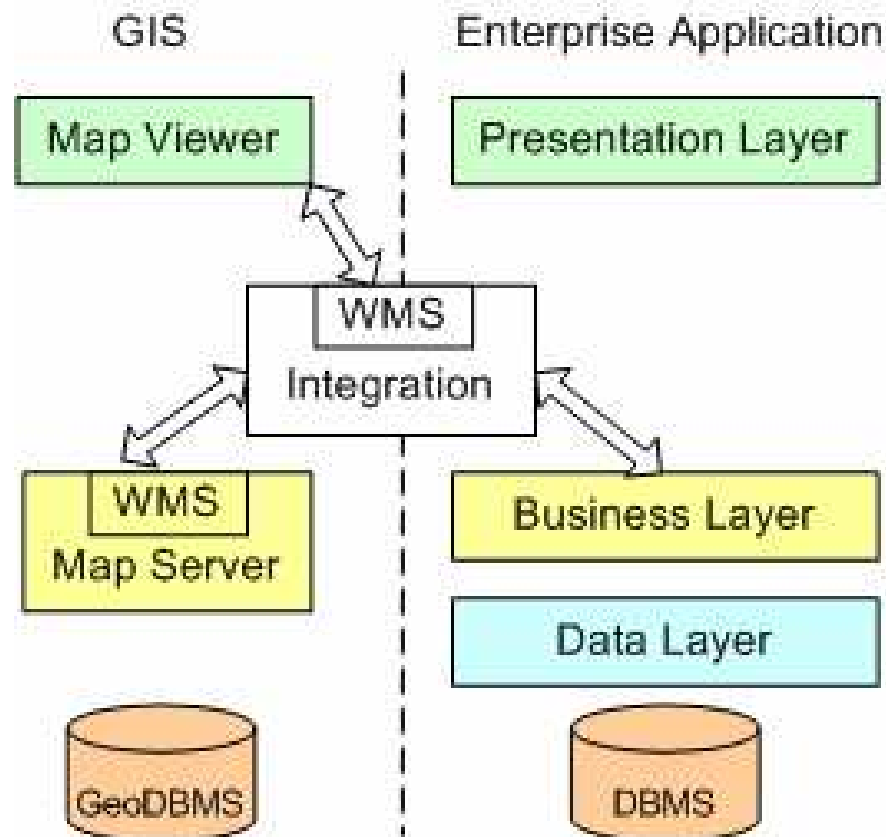
## Aplicación Empresarial



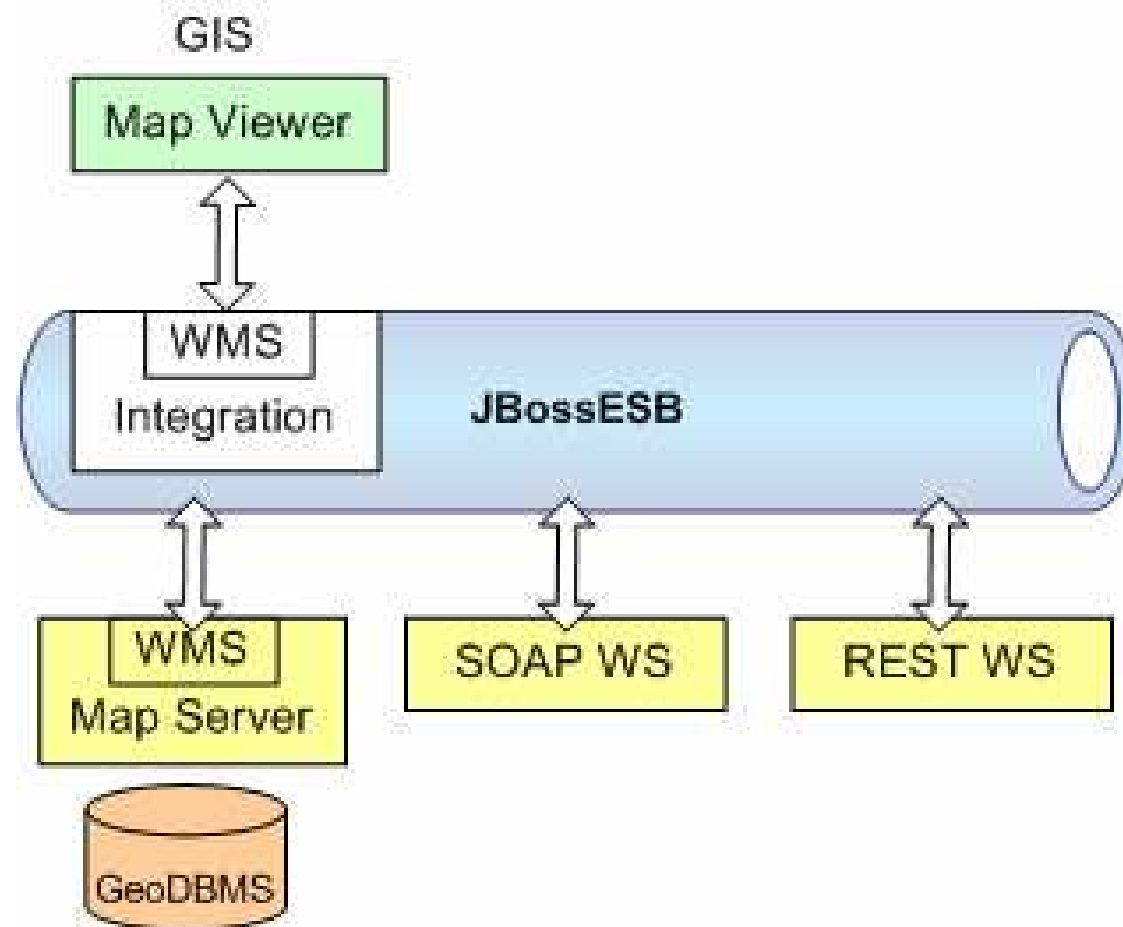
# Escenarios de Integración



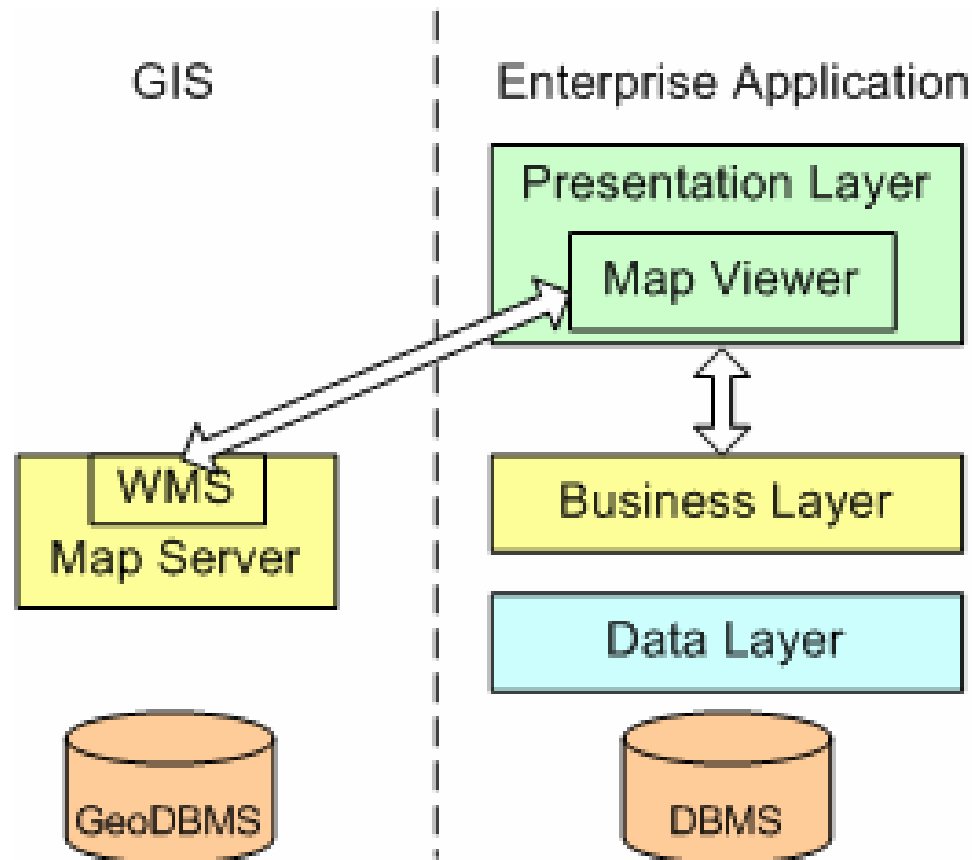
# Escenarios de Integración



# Escenarios de Integración



# Escenarios de Integración





# Referencias

- ❑ G. Hohpe and B. Woolf, Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional, October 2003.
- ❑ J. McGovern, O. Sims, A. Jain, and M. Little, Enterprise Service Oriented Architectures: Concepts, Challenges, Recommendations
- ❑ D. Chappell, Enterprise Service Bus. O'Reilly Media, Inc., July 2004.
- ❑ M. Papazoglou, Web Services: Principles and Technology, 1st ed. Prentice Hall, September 2007.
- ❑ G. Alonso, F. Casati, H. Kuno, and V. Machiraju, Web Services, 1st ed. Springer, October 2003.
- ❑ M. P. Papazoglou, P. Traverso, S. Dustdar, and F. Leymann. "Service-oriented computing: State of the art and research challenges", Computer, vol. 40, no. 11, pp. 38-45, 2007.
- ❑ The SOA Source Book



<http://www.opengroup.org/soa/source-book/intro/>