

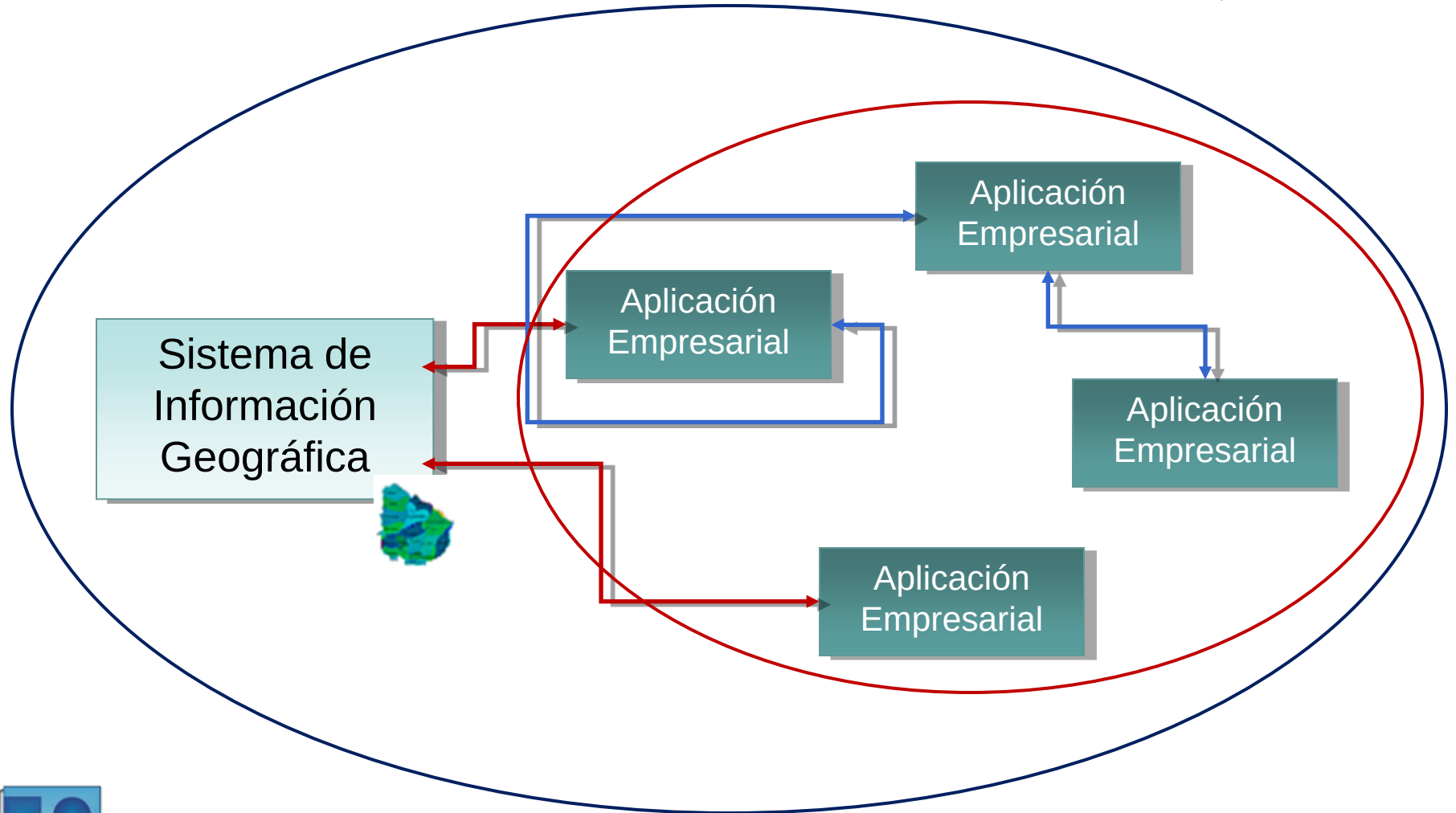
Taller de Sistemas de Información Geográficos Empresariales



Aplicaciones Empresariales y GIS



Motivación



Agenda

- ☐ Aplicaciones Empresariales
- ☐ Middleware
- ☐ SOA y ESB
- ☐ Arquitectura y Tecnologías GIS
- ☐ Alternativas de Integración



Taller de Sistemas de Información Geográficos Empresariales



Aplicaciones Empresariales



- ❑ Una **Aplicación Empresarial** es una aplicación de software desarrollada para administrar las operaciones, activos y recursos de una empresa
- ❑ Ejemplos: Contabilidad, Seguimiento de envíos, Servicio al cliente, Nómina de empleados
- ❑ Características: persistencia de datos, grandes volúmenes de datos, varias Interfaces de Usuario, integración con otras aplicaciones, acceso concurrente a datos.

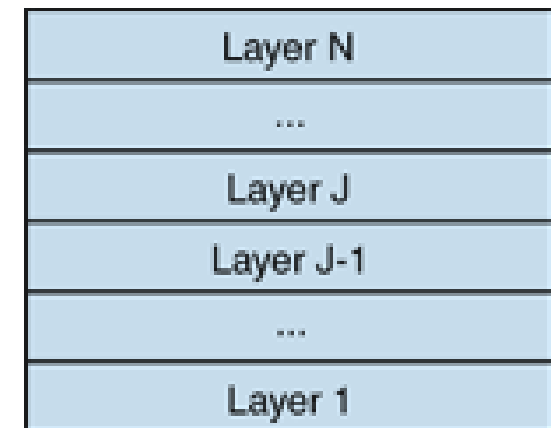


- ❑ El proceso de desarrollo de una aplicación empresarial involucra al menos:
 - Programadores de aplicaciones
 - Administradores de base de datos
 - Diseñadores de interfaz de usuario
 - Integradores de aplicaciones
- ❑ Complejidades:
 - ❑ Administración, Mantenibilidad, Escalabilidad, Interoperabilidad, Seguridad, Confiabilidad, Accesibilidad y Usabilidad, Internacionalización

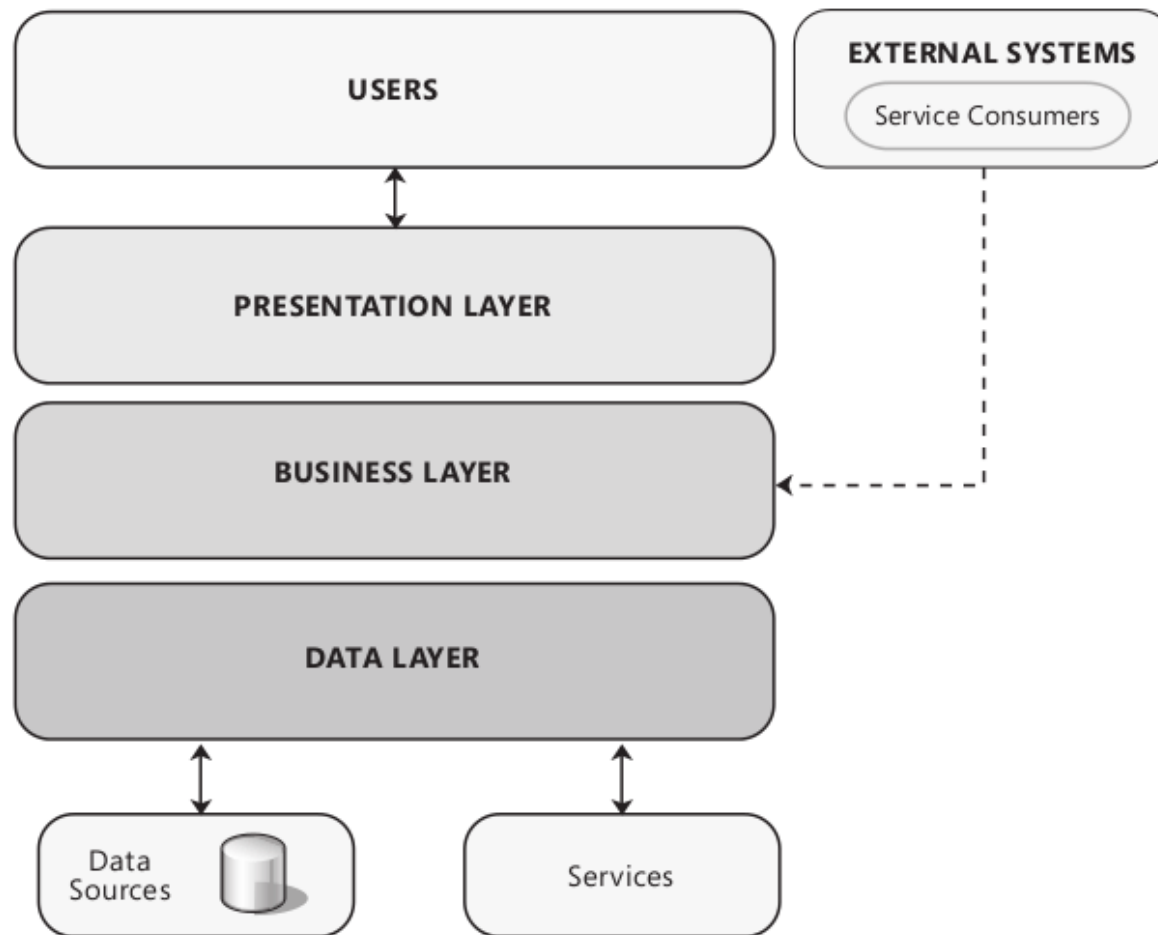


Arquitectura en Capas (Layers)

- ❑ “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)



Arquitectura Física (Tiers)

- ❑ Las capas antes presentadas pueden estar ubicadas en la misma locación física (tier) o en diferentes locaciones físicas
- ❑ Si se encuentran en locaciones físicas diferentes, existen fronteras físicas que deben ser tomadas en cuenta en el diseño



Deployment No Distribuido

- ❑ Este enfoque minimiza el número de servidores requeridos
- ❑ Minimiza el impacto en performance inherente a la comunicación entre capas de diferentes lugares físicos
- ❑ Sin embargo, compartir el mismo hardware, puede impactar la performance, por ejemplo, al acceder a recursos compartidos



Deployment Distribuido

- ❑ Este enfoque permite configurar el hardware según las necesidades de cada capa
- ❑ Esto permite ajustar las necesidades de escalabilidad según cada capa de la aplicación
- ❑ Sin embargo, el uso de componentes distribuidos, impacta la performance a la hora de realizar llamadas remotas entre diferentes locaciones físicas



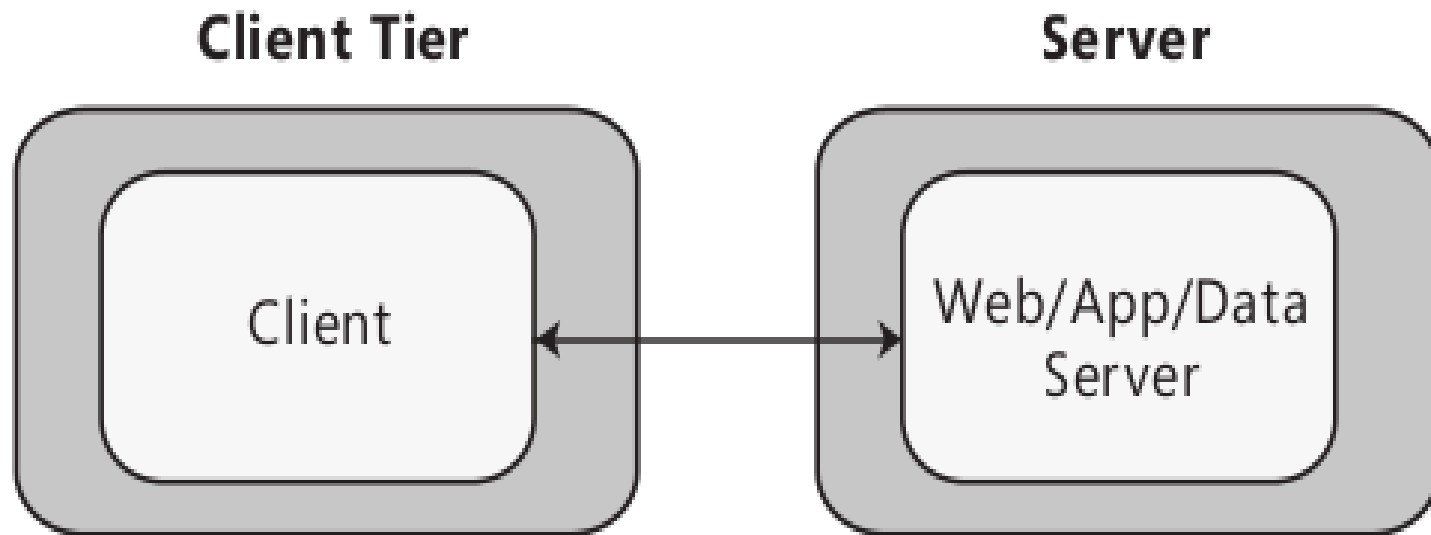
Deployment Distribuido (N-Tiers)

- ☐ Cliente / Servidor
- ☐ 2-Tier
- ☐ 3-Tier
- ☐ N-Tier



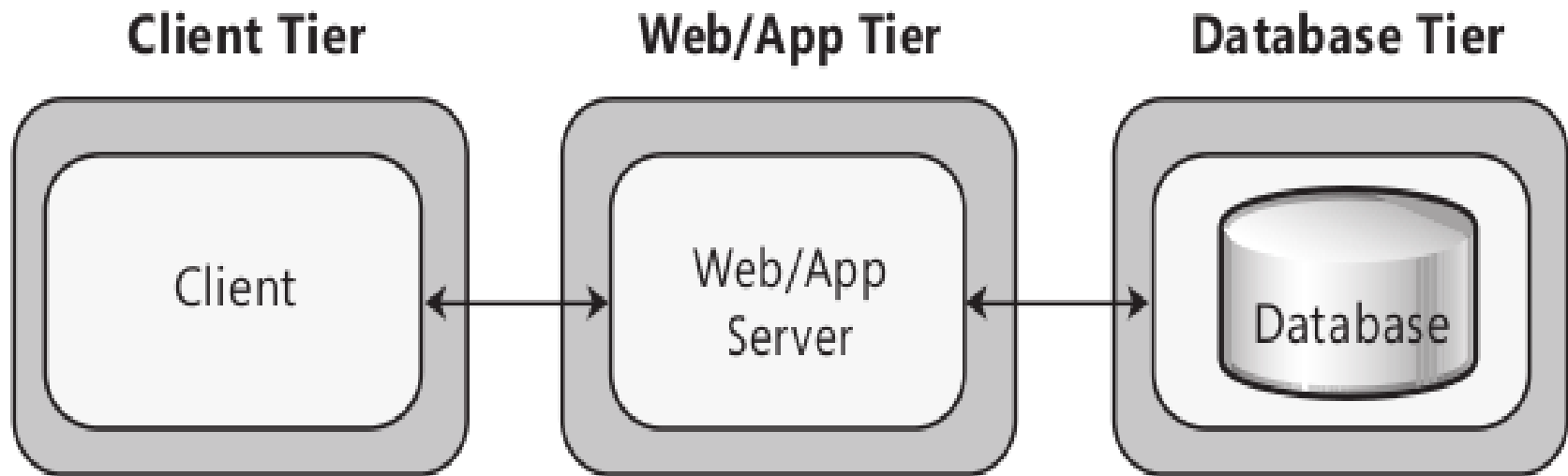
Deployment Distribuido (N-Tiers)

Cliente / Servidor



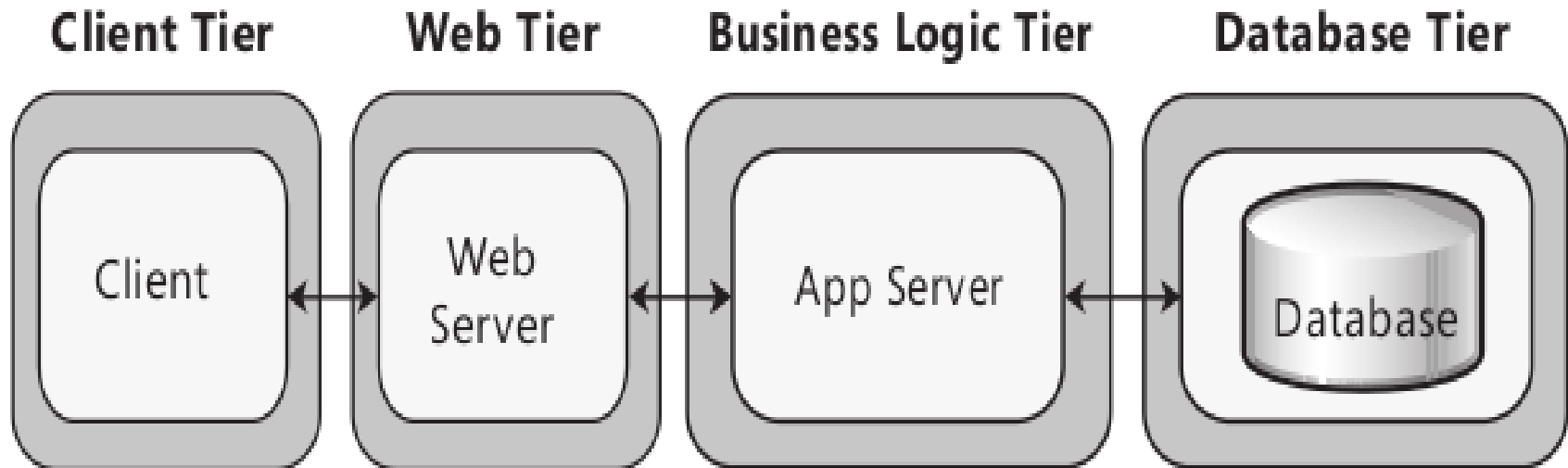
Deployment Distribuido (N-Tiers)

3-Tier



Deployment Distribuido (N-Tiers)

4-Tier



Integración de Aplicaciones

- 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
 - A nivel de Interfaz de Usuario

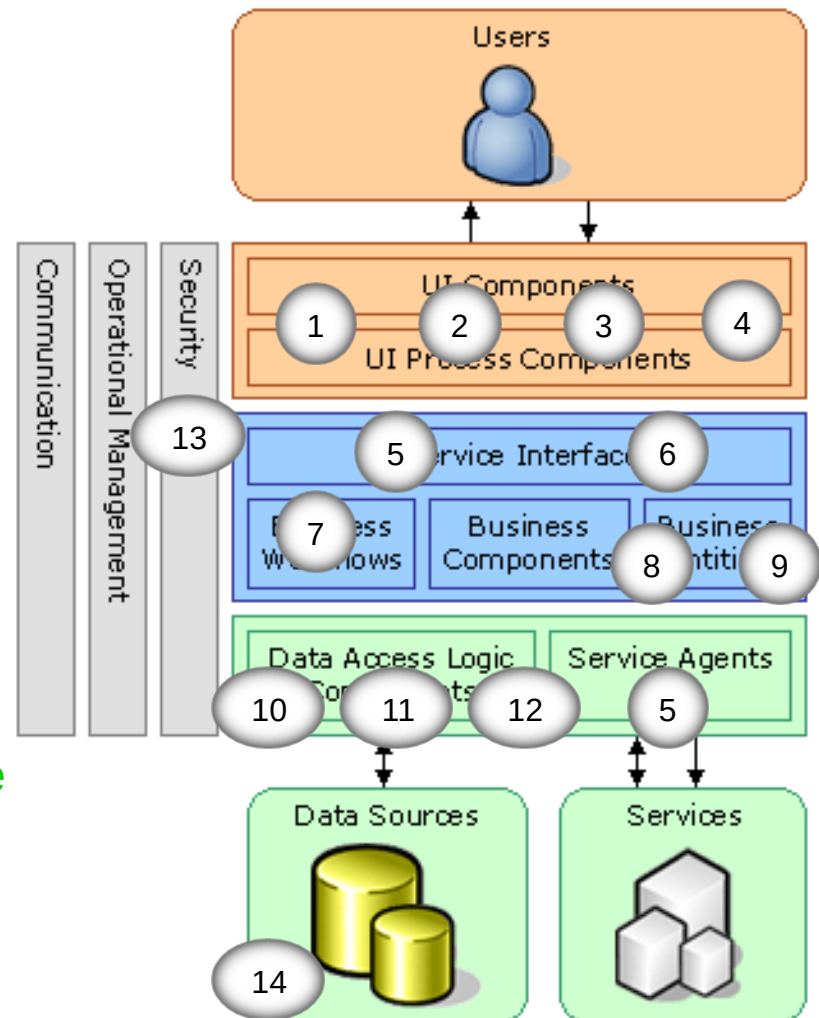


- ☐ Existen plataformas que facilitan el desarrollo e integración de aplicaciones empresariales
- ☐ Las mismas brindan soluciones a varios de los problemas presentados
- ☐ Permiten que el desarrollador se concentre en los aspectos relevantes para el negocio
- ☐ .Net Framework
- ☐ Jakarta Enterprise Edition (JEE)



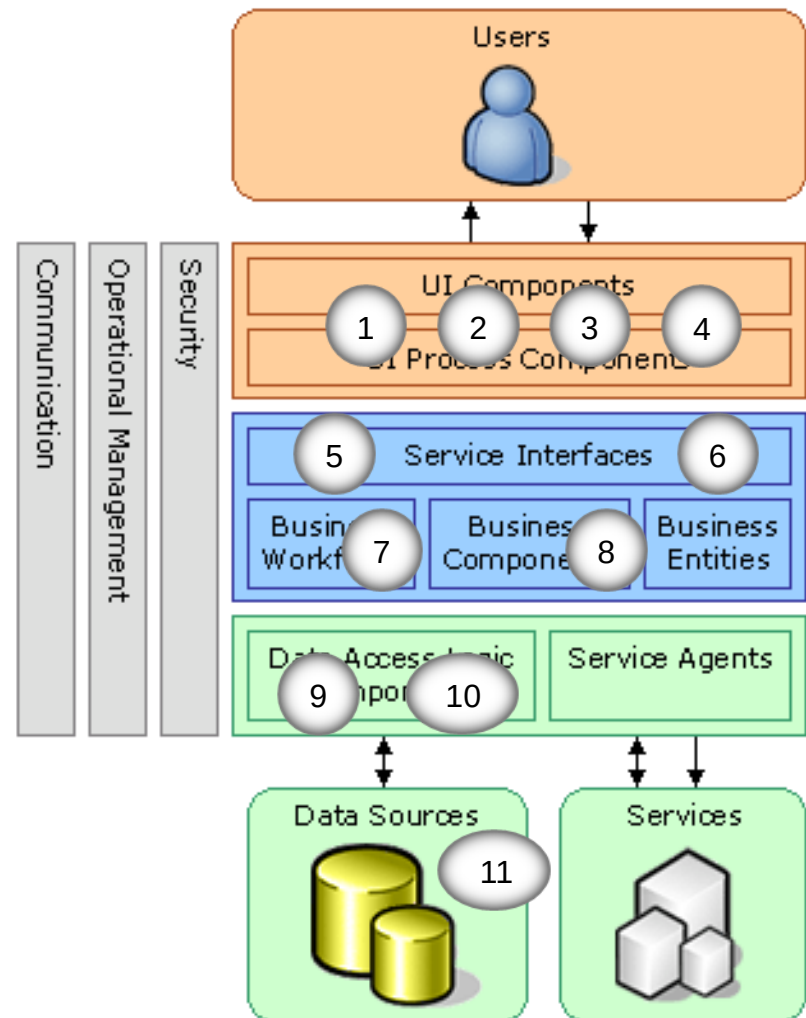
.Net Framework

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



Jakarta 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. Hibernate
11. PostgreSQL



<https://jakarta.ee/>



Taller de Sistemas de Información Geográficos Empresariales

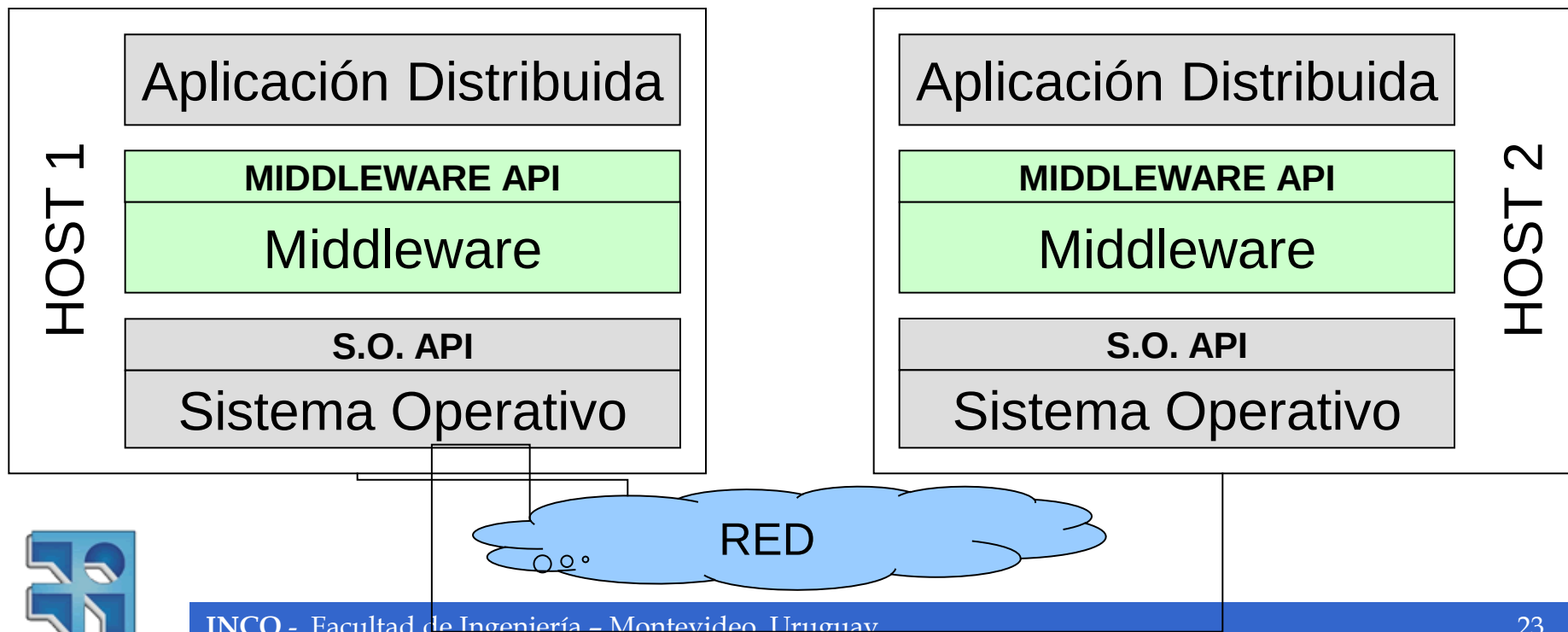


Middleware



Middleware

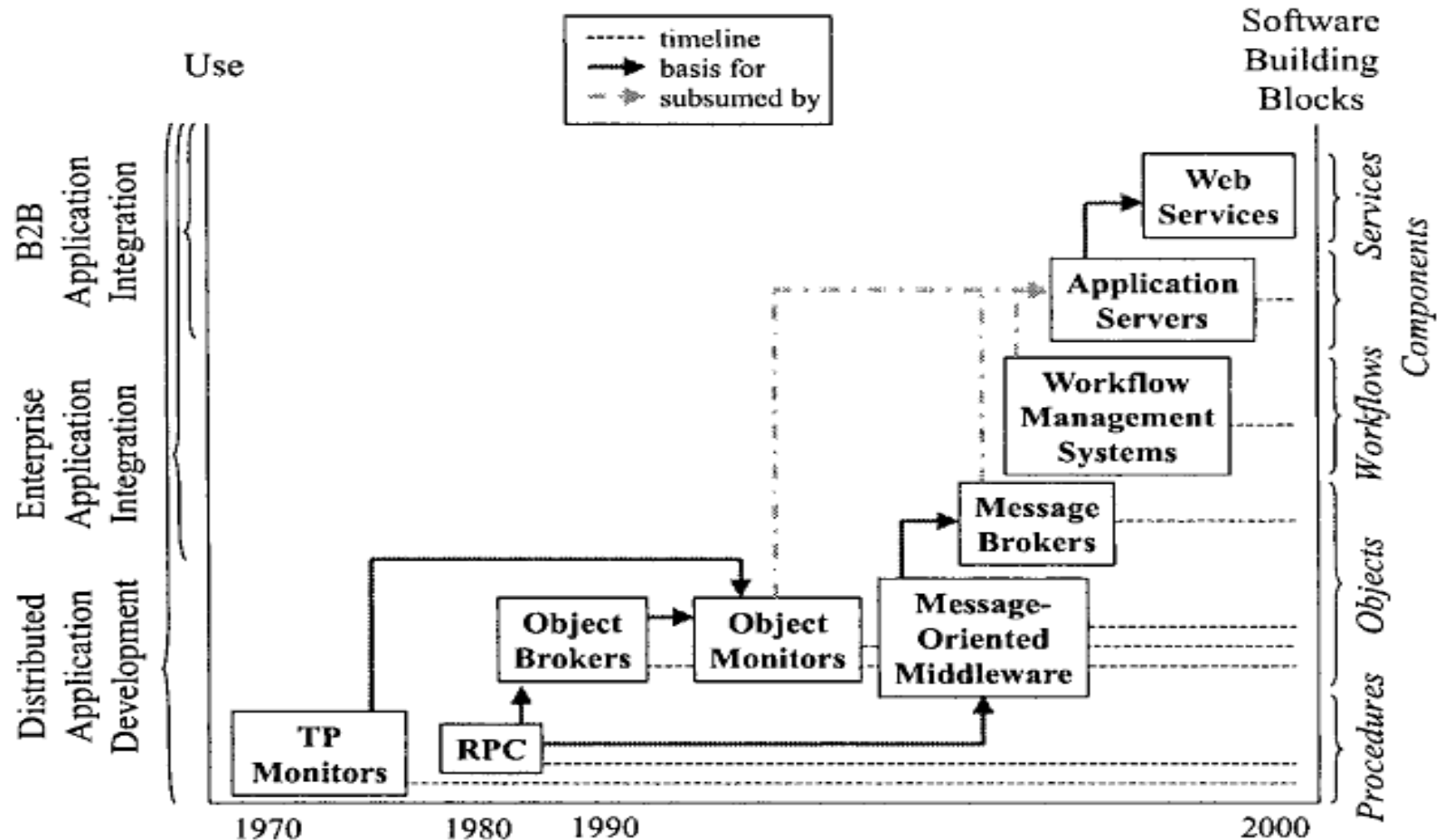
- ❑ Middleware es una capa de software distribuida, situada entre el sistema operativo y las aplicaciones, diseñado para manejar la heterogeneidad y complejidad inherente a los sistemas distribuidos



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



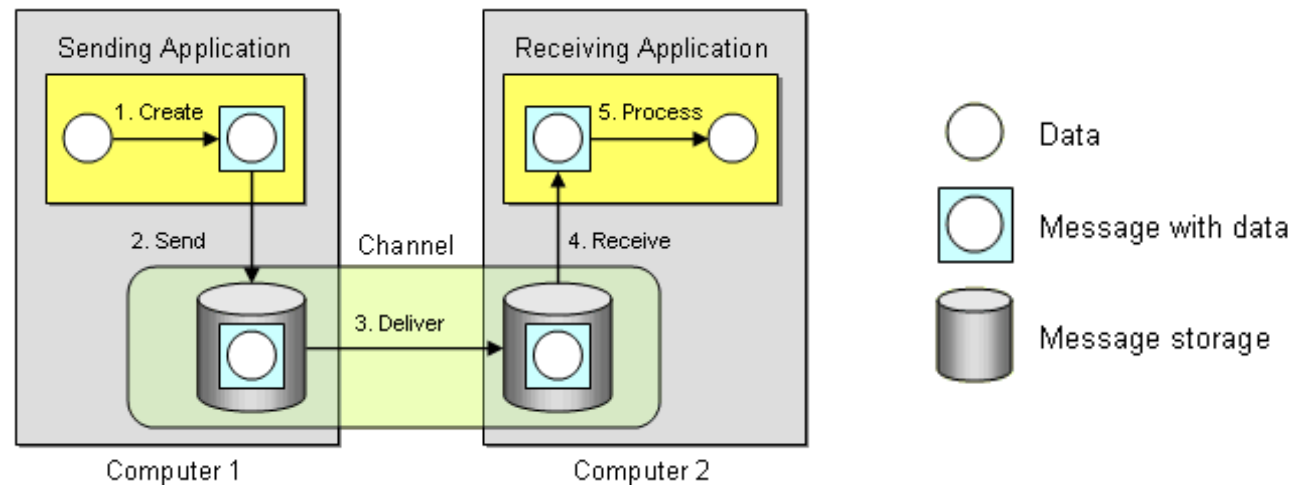
Evolución Middleware



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



G. Hohpe and B. Woolf, Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional, October 2003.



- ❑ 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

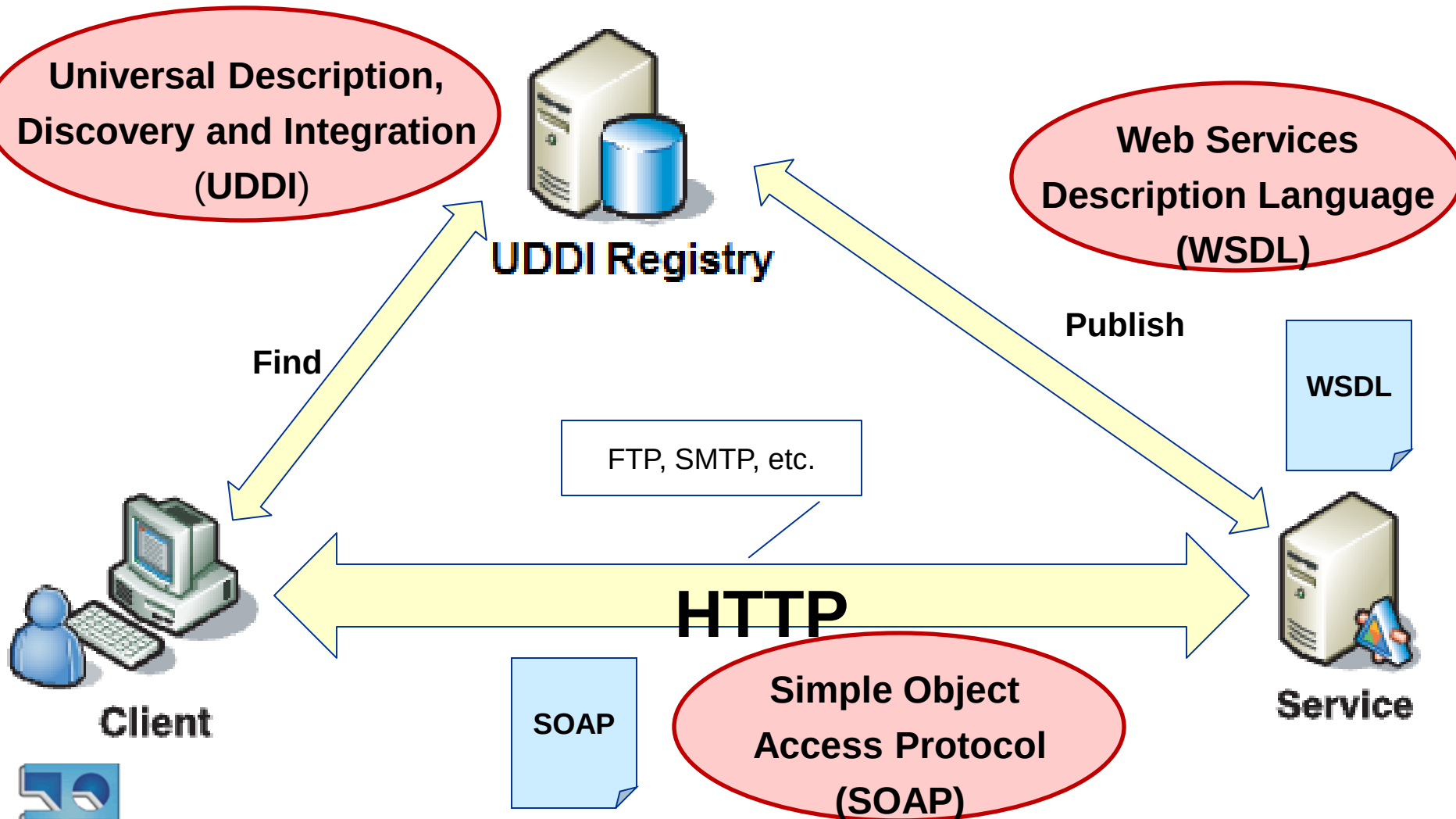


- Un Web Service es una **aplicación de software identificada por una URI**, cuyas interfaces y formas de acceso pueden ser **definidas, descritas y descubiertas como artefactos XML**, y soporta la interacción directa con otros componentes de software utilizando **mensajes basados en XML**, intercambiados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



- ❑ 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
 - interfaces para utilizar el registro UDDI
 - Inquiry
 - Publish



Segunda Generación de WS

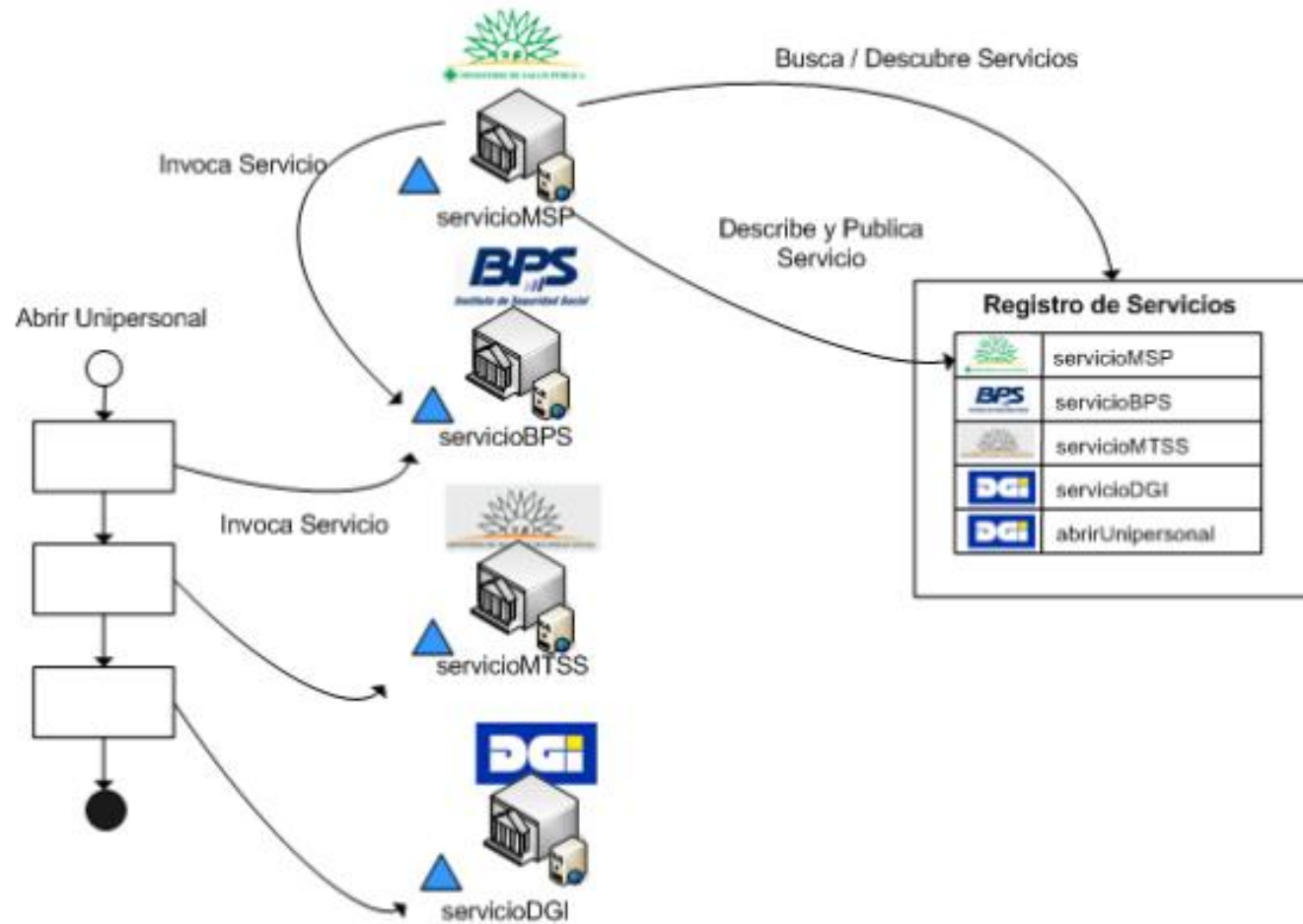
- ❑ Surgen como forma de abordar problemáticas comunes en contextos empresariales
- ❑ Se les conoce como WS-*
- ❑ Cada una aborda una problemática específica:
 - Seguridad, Transacciones, Mensajería, etc



- ❑ *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*



WS-BPEL



- ❑ 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



- ❑ 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



□ Alternativas

- Seguridad a nivel de transporte
 - A través de HTTPS
 - Tecnología madura y existencia de expertos
- Seguridad a nivel de mensaje SOAP
 - WS-Security



- ❑ 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



- ❑ 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.



□ 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.



- 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.



Web Services REST

Encabezados de la petición

```
GET /sqlrest/INVOICE/9999/ HTTP/1.1
Host: www.thomas-bayer.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; es-AR; rv:1.9.2.3)
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: es,en;q=0.8,es-ar;q=0.5,en-us;q=0.3
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 115
Connection: keep-alive
```

Encabezados de la respuesta

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: application/xml
Transfer-Encoding: chunked
Date: Wed, 07 Apr 2010 01:30:00 GMT
```



Taller de Sistemas de Información Geográficos Empresariales



SOA y ESB



❑ 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 descrita, publicada, descubierta y combinada



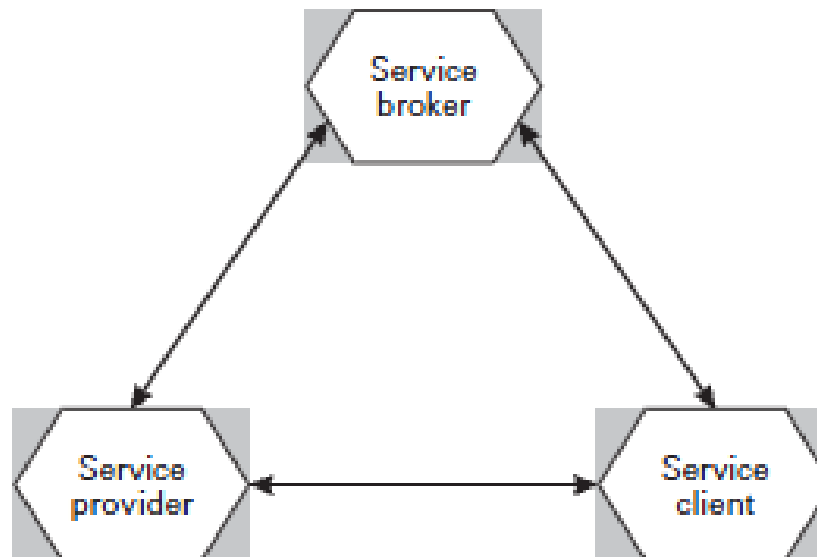
□ Principios

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

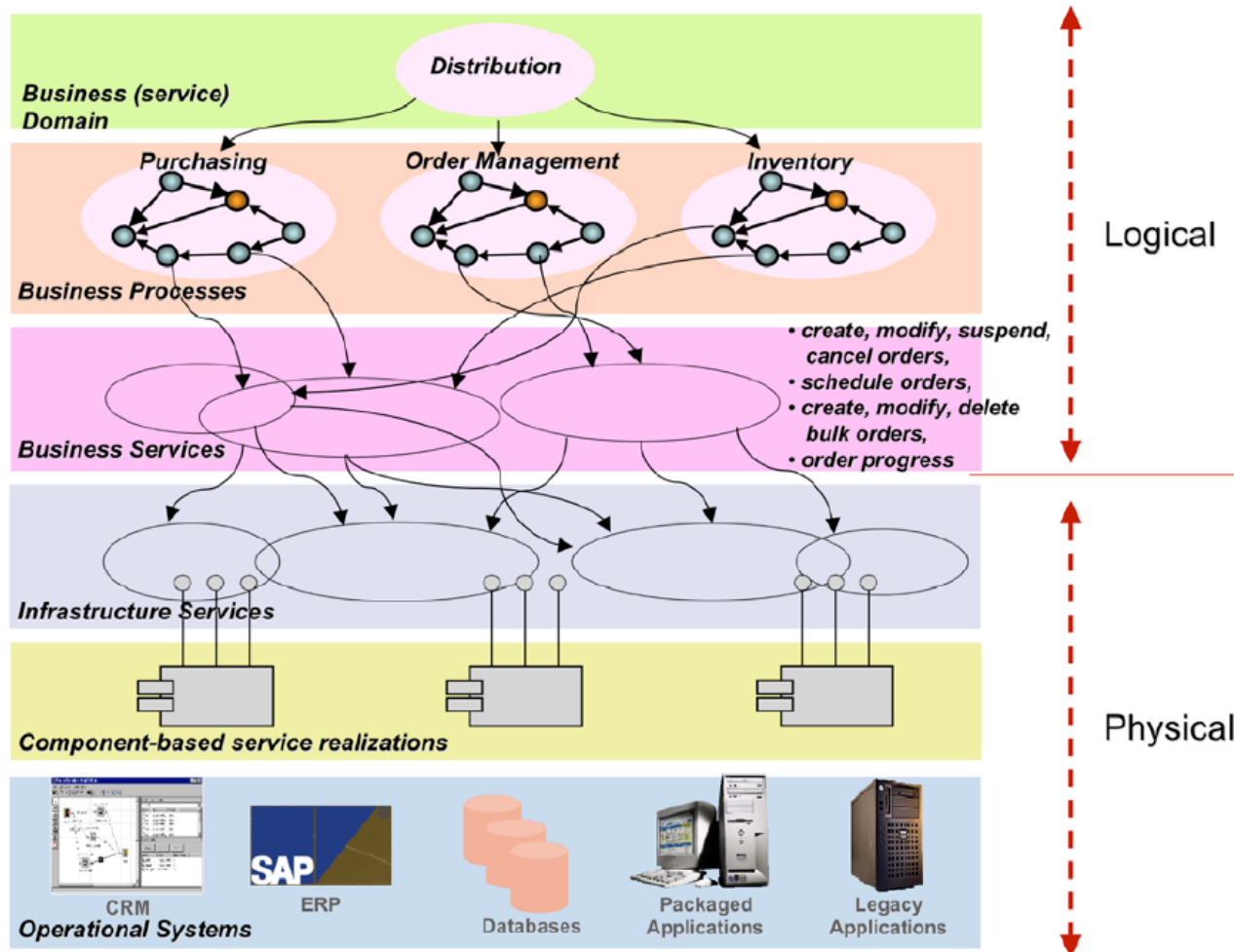


□ Arquitectura Orientada a Servicios

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



Service Oriented Architecture

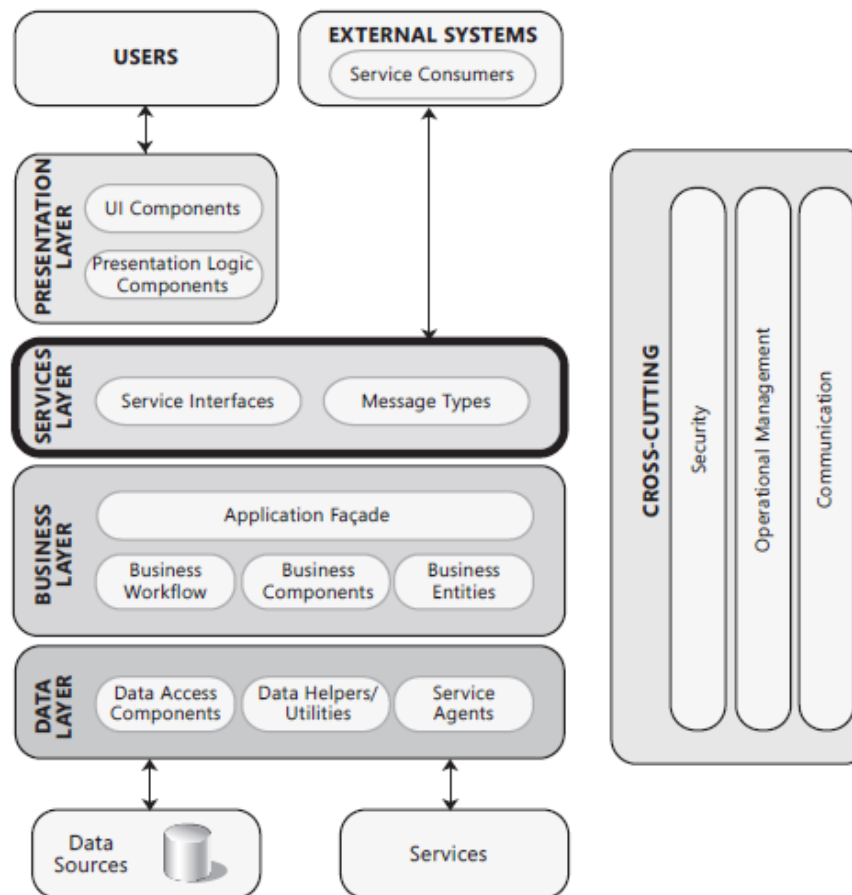


M. Papazoglou, Web Services: Principles and Technology, 1st ed. Prentice Hall, September 2007.

- ❑ 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



Capa de Servicios

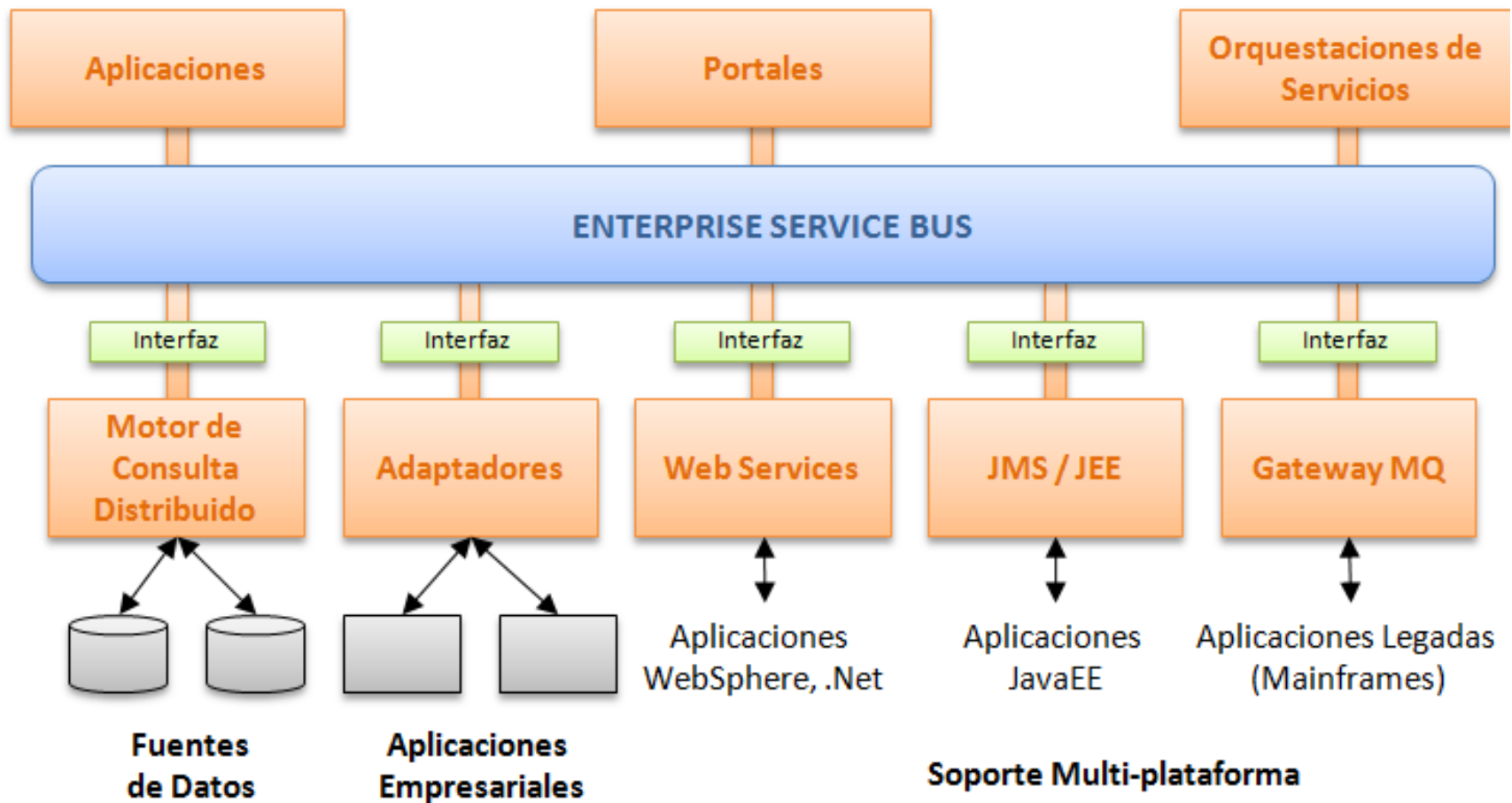


Enterprise Service Bus (ESB)

- 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

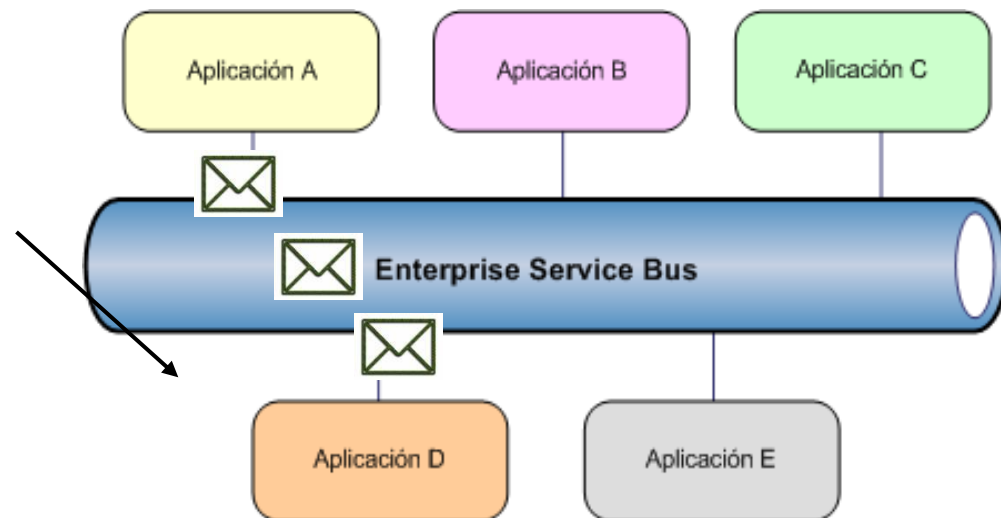


Enterprise Service Bus (ESB)

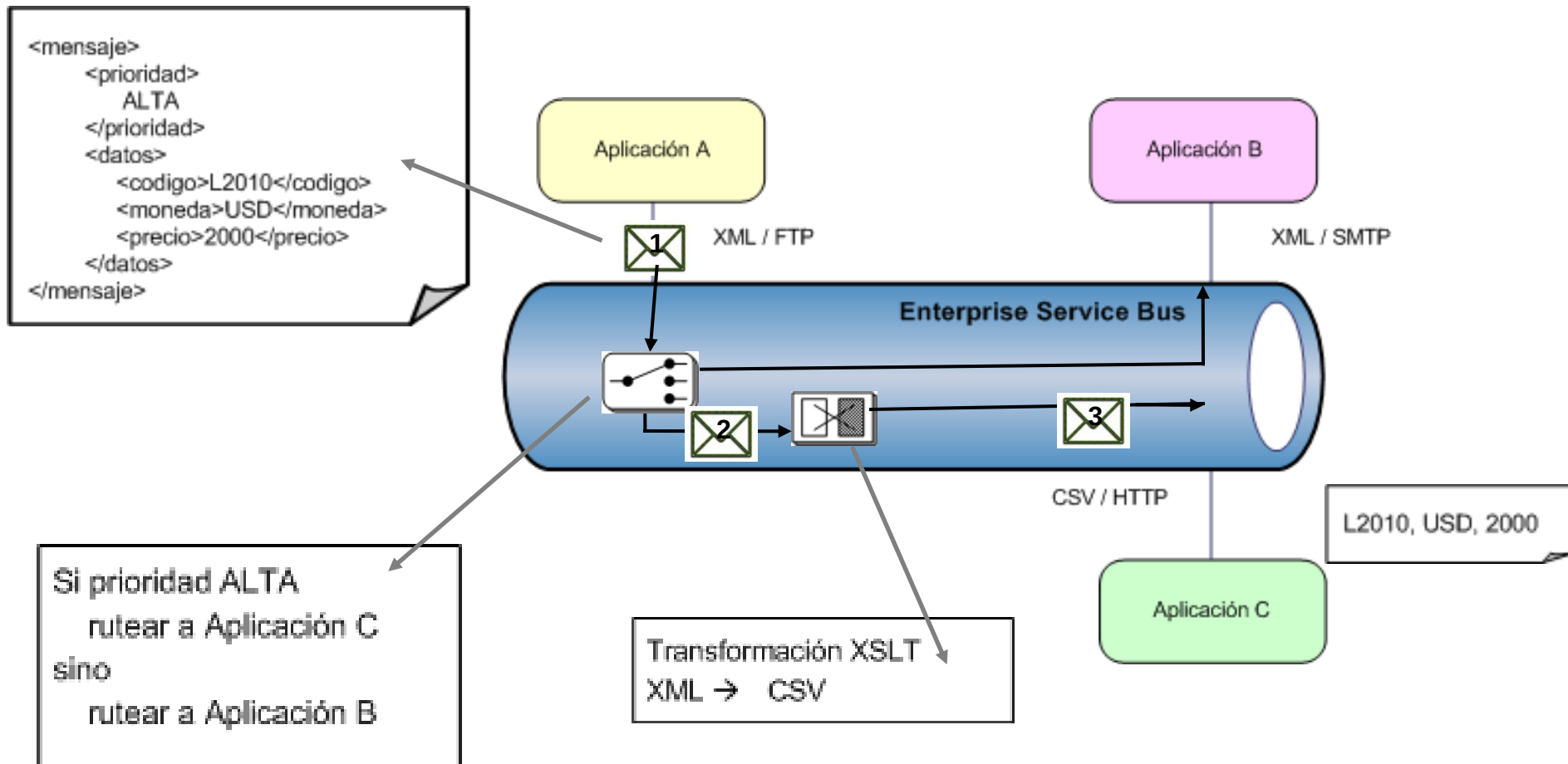


Enterprise Service Bus (ESB)

- ❑ En lugar de interactuar directamente las aplicaciones se comunican enviando mensajes a través del ESB
- ❑ Los mensajes que fluyen a través del ESB son en general mensajes XML



Enterprise Service Bus (ESB)



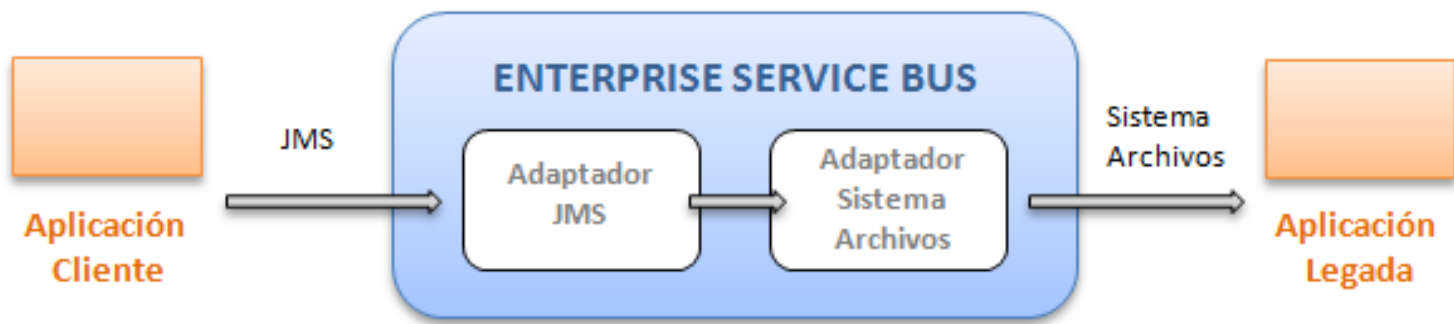
Funcionalidades de ESB

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



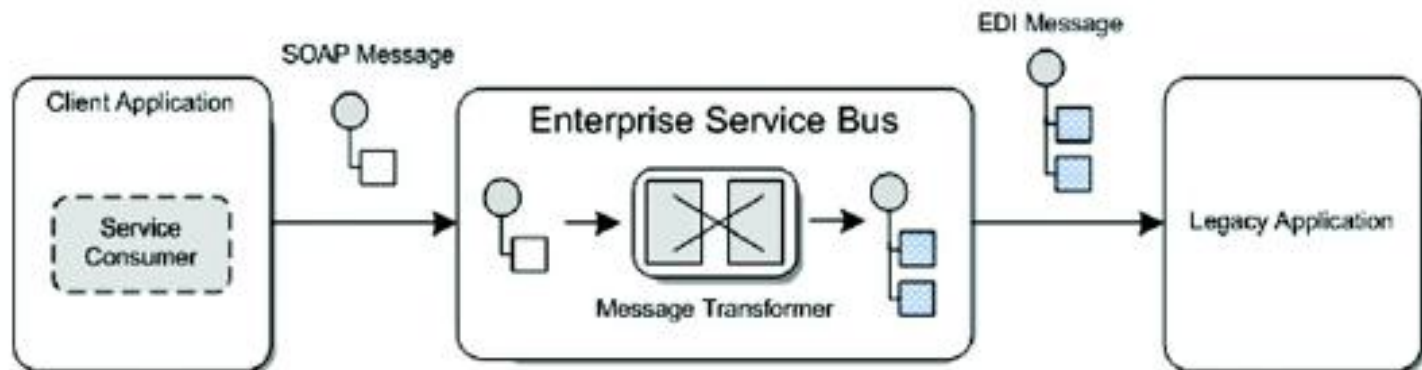
Conectividad y Adaptadores

- ❑ Permiten satisfacer un requerimiento de integración común conocido como conversión de protocolo (protocol switch).
- ❑ Este requerimiento se da cuando dos aplicaciones que necesitan integrarse no manejan un protocolo de comunicación común.



Transformación de Mensajes

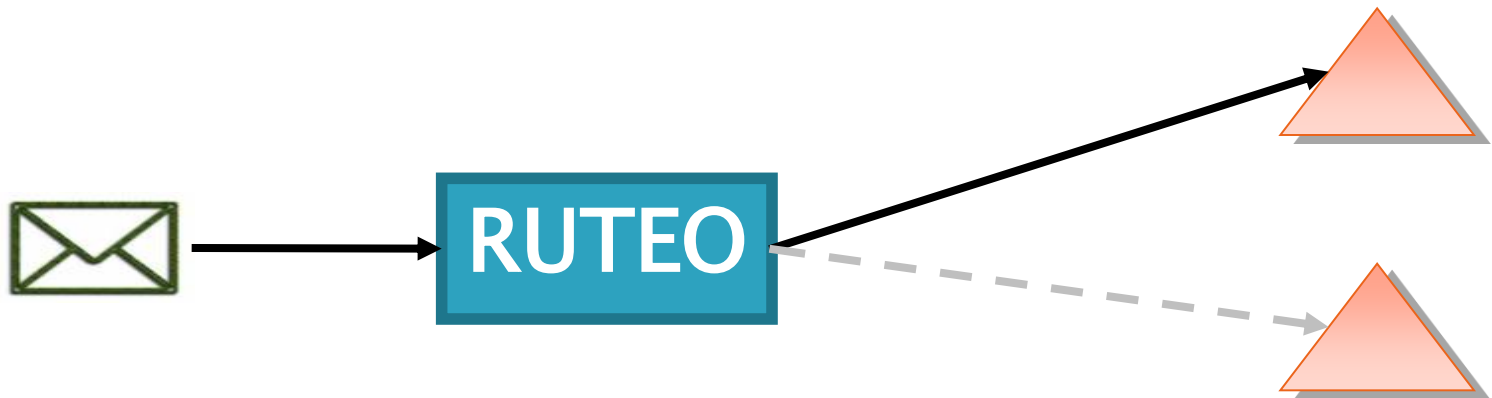
- ❑ Los ESBs también incluyen capacidades de transformación de mensajes.
- ❑ Estas capacidades posibilitan, por ejemplo, que aplicaciones que utilizan distintos formatos o modelos de datos puedan comunicarse.



(Rademakers and Dirksen 2008)



- ❑ El ruteo inteligente permite determinar dinámicamente el destino de un mensaje (seleccionando de varios destinos posibles)



- ❑ Basado en contenido, contexto, balanceo de carga, etc



- ❑ Los ESB incluyen funcionalidades de monitoreo que proveen valores para distintas métricas
 - tiempos de respuesta,
 - cantidad de mensajes procesados por servicio,
 - errores en la invocación de servicios, etc
- ❑ Esto permite detectar:
 - cuellos de botella,
 - incumplimiento de requerimientos de calidad de servicio, etc



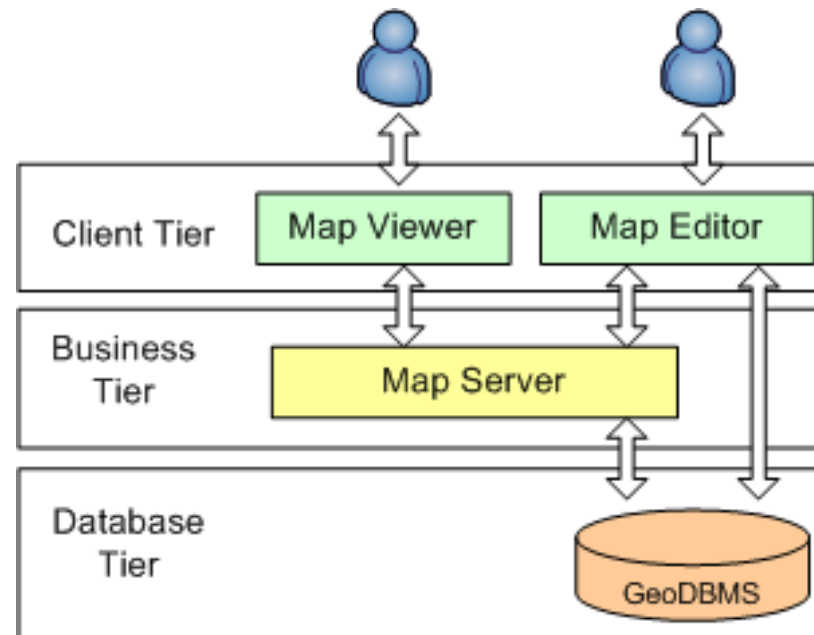
Taller de Sistemas de Información Geográficos Empresariales



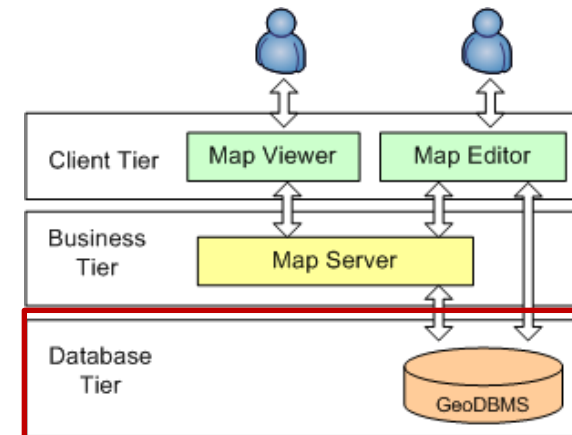
Arquitectura y Tecnologías GIS



- ❑ En las últimas décadas el Software GIS ha pasado de ser un software “standalone” de escritorio, a tener la típica arquitectura en capas de las aplicaciones empresariales



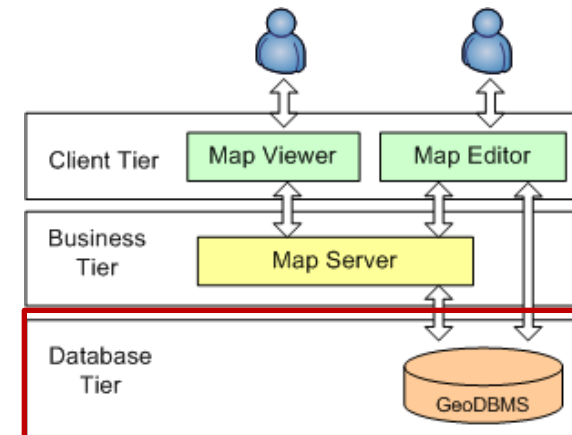
- ❑ Un GeoDBMS da soporte al almacenamiento y consulta de objetos geográficos.
- ❑ El enfoque común es extender un DBMS con elementos geográficos:
 - Tipos
 - Funciones
 - Metadatos



SELECT r.nombre, **ST_LENGTH(r.geom) AS longitud**
FROM Calles r WHERE r.codigo=223;

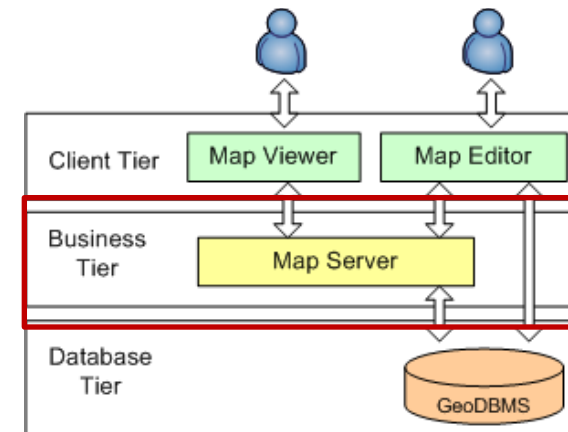


- ❑ Las extensiones geográficas a un DBMS deben cumplir con el estándar Open Geospatial Consortium (OGC) conocido como:
 - Simple Features Standard, SQL Option
- ❑ Algunas extensiones SFS son:
 - Oracle Spatial
 - PostGIS
 - MySQL Spatial Extensions
 - ESRI ArcSDE



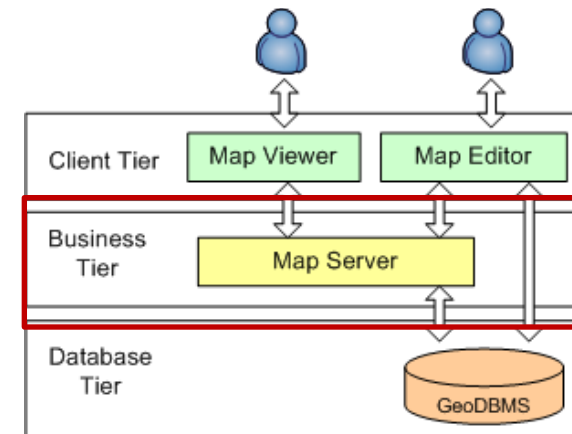
Map Server

- ❑ Un Servidor de Mapas es un componente de SW que generalmente se ejecuta sobre un Servidor Web o Servidor de Aplicaciones
- ❑ Su objetivo principal es proveer a clientes de Internet mapas generados dinámicamente (a partir de datos vectoriales o raster)



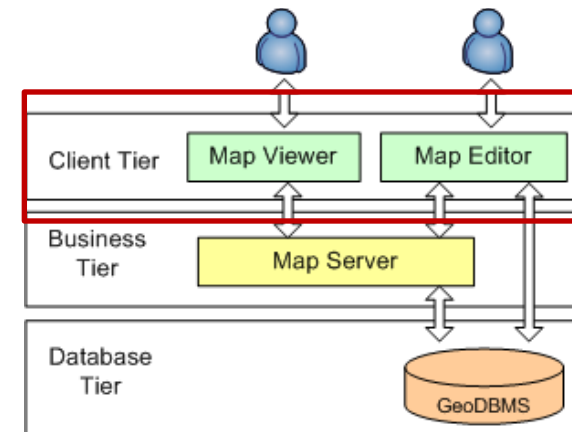
Map Server

- ❑ La tendencia actual es que los servidores de mapas implementen estándares de Web Services geográficos (de la OGC):
 - Web Map Service (WMS)
 - proveen mapas sólo lectura
 - Web Feature Service (WFS)
 - acceso y manipulación de objetos geográficos (protocolos basados en XML)



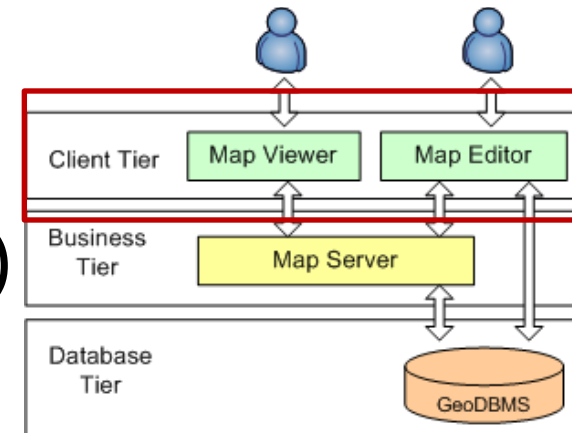
Map Viewer

- ❑ Un Map Viewer es una aplicación cliente que despliega un mapa en una interfaz de usuario
 - en general son aplicaciones Web-based que obtienen mapas desde Map Servers (ej. via WMS)
- ❑ Ejemplos
 - Google Maps
 - OpenLayers



Map Editor

- ❑ Un Map Editor permite además crear mapas, modificar objetos, realizar geo-procesamiento avanzado
- ❑ Son generalmente aplicaciones desktop pero hay alternativas Web-based
- ❑ Se pueden comunicar con Map Servers (vía WMS o WFS) o con geo-databases (vía SQL)
- ❑ Ejemplos: gvSIG, QGIS



Taller de Sistemas de Información Geográficos Empresariales

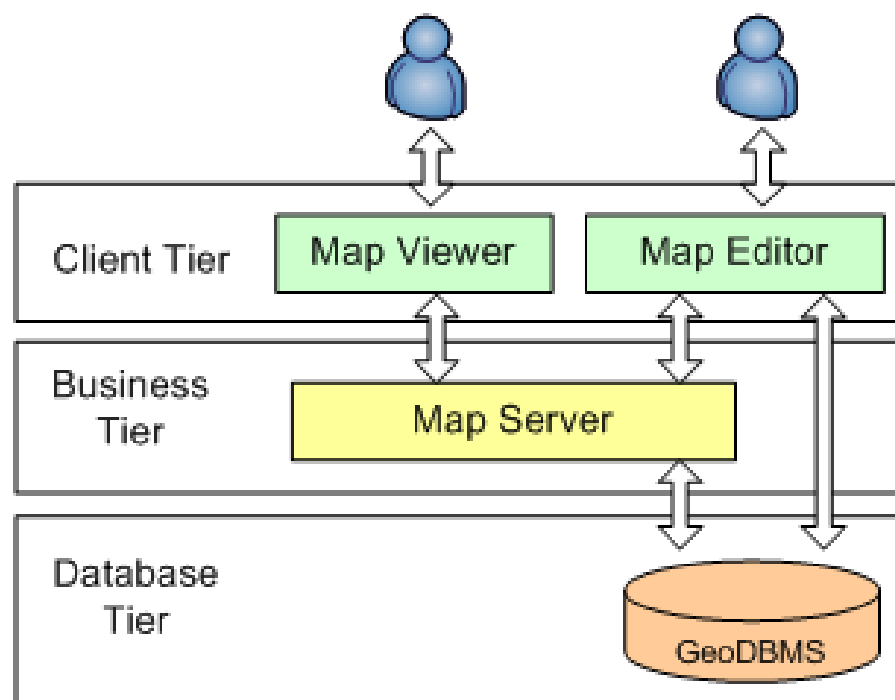


Alternativas de Integración

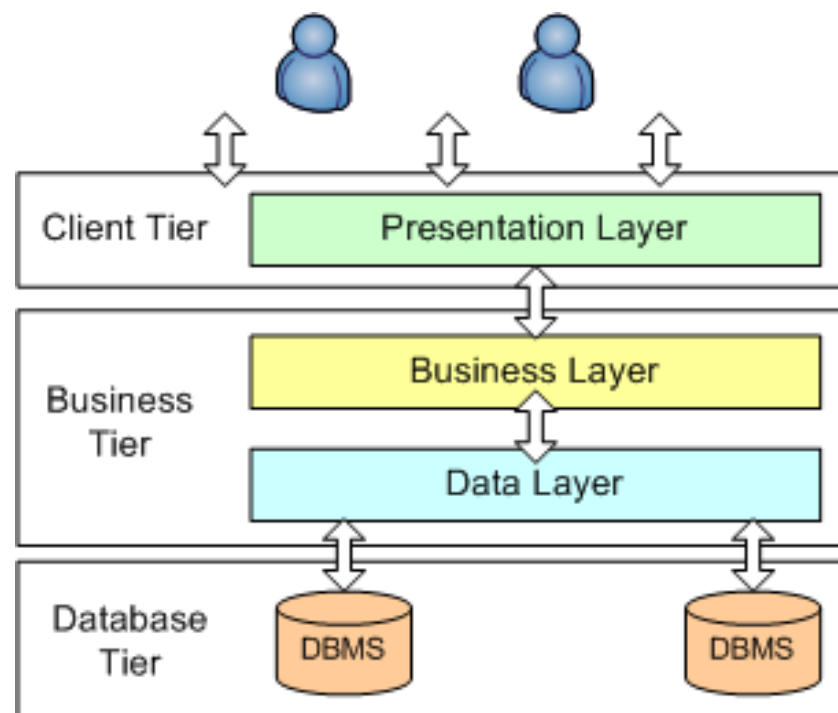


Escenarios de Integración

GIS



Aplicación Empresarial

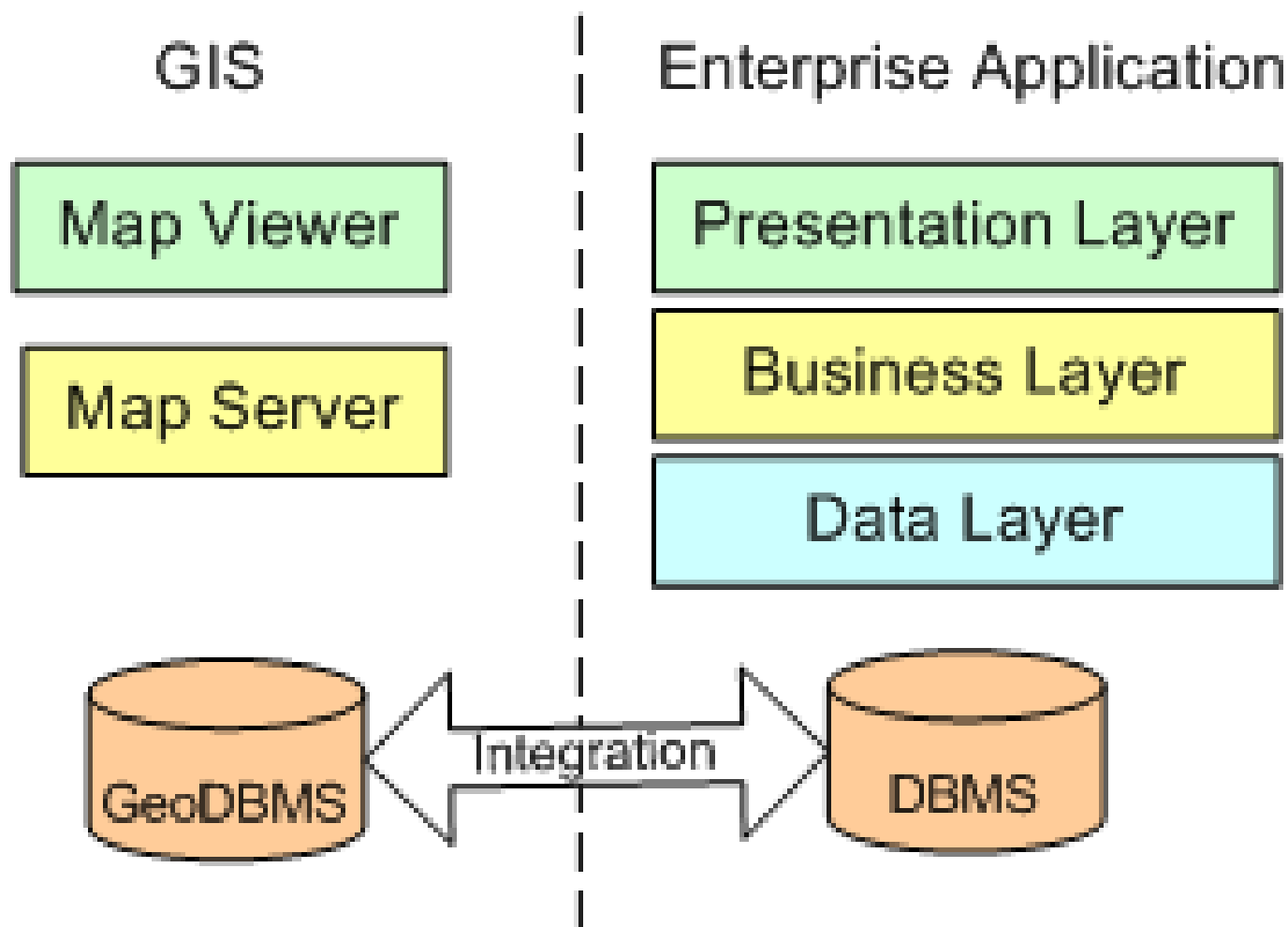


Alternativas de Integración

- ☐ Integración a nivel de Base de Datos
- ☐ Integración a nivel de Servicios
- ☐ Integración a nivel de Interfaz de Usuario



Integración a nivel de BD



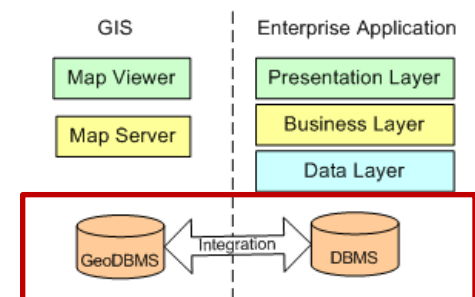
Integración a nivel de BD

❑ Utilizar un database-link

- Es posible vía las capacidades de algunos DBMS o productos de terceros (ej. dblink)
- Este mecanismo permite acceder a una BD externa a través de vistas

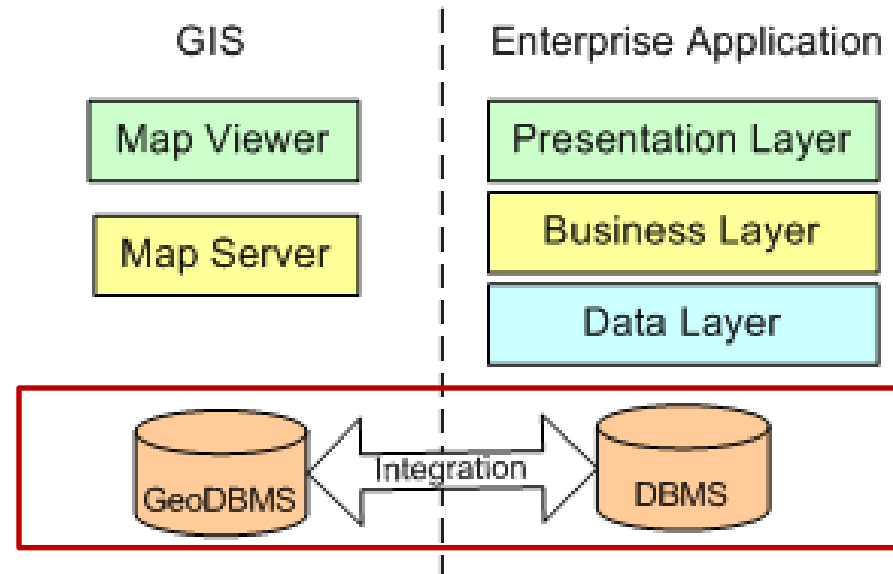
❑ Utilizar mecanismos de ETL

- Se deben implementar mecanismos para extraer información de la BD empresarial y almacenarla en la geo-DB
- Definir periodicidad de carga



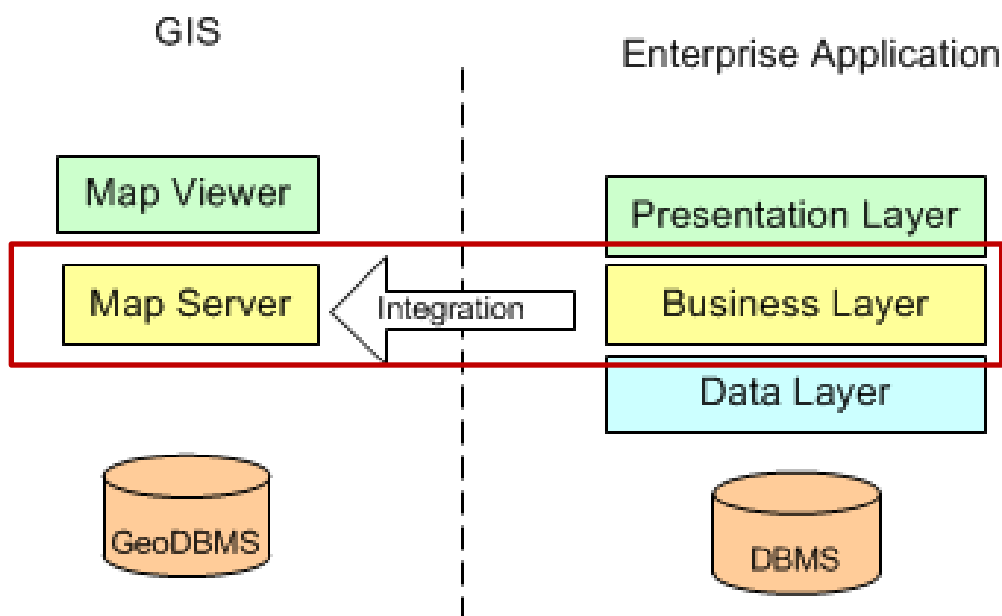
Integración a nivel de BD

- ❑ ¿database-link vs ETL?
- ❑ ¿es siempre posible este enfoque?



Integración a nivel de Servicios

- ❑ Los servicios geográficos tienen interfaces bien conocidas, por lo que invocarlos desde aplicaciones empresariales no presenta mayores desafíos

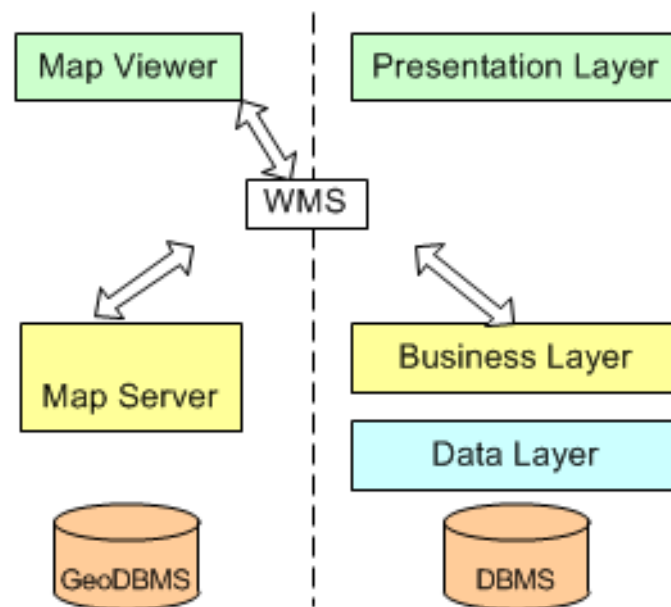


- ❑ El principal desafío es proveer datos provenientes de aplicaciones de negocio a través de interfaces geográficas
- ❑ La problemática se da porque:
 - Muchos clientes (ej. Map Viewers) están diseñados para consumir sólo WS Geográficos
 - Los Map Servers proveen poco soporte para consumir servicios externos (ej. WS SOAP)
- ❑ ¿Qué alternativas hay?



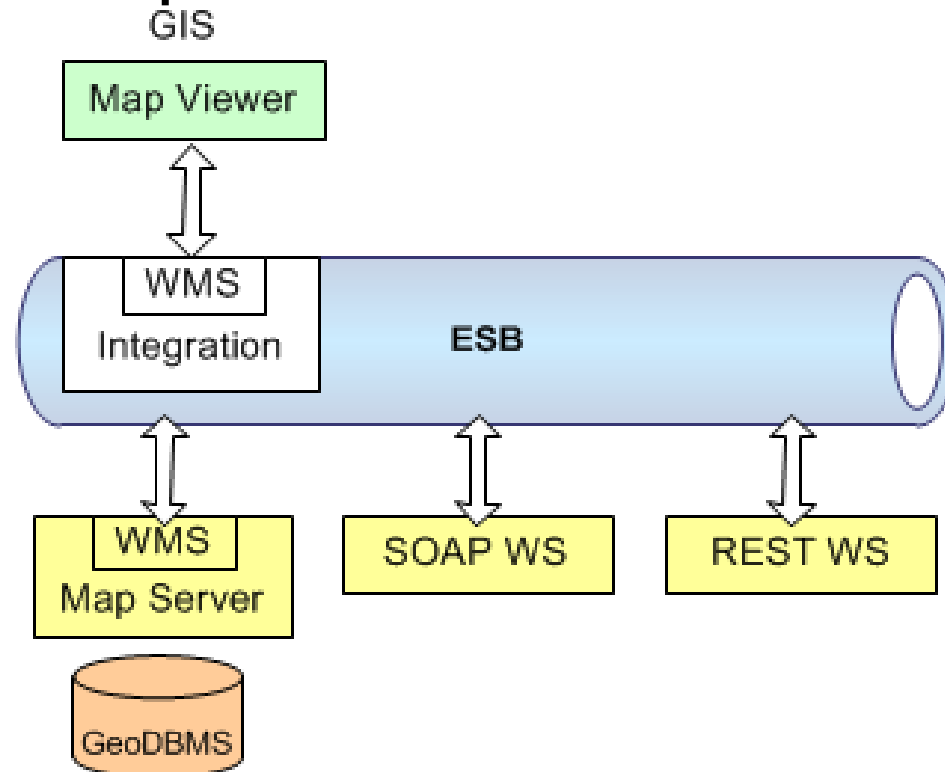
Integración a nivel de Servicios

- ❑ Un posible enfoque es desarrollar un componente que consuma ambos servicios, integre los datos y los provea a través de una interfaz geográfica
- ❑ La obtención de datos y la integración es transparente tanto para el cliente como para el Map Server



Integración a nivel de Servicios

- Una alternativa para implementar el componente de integración es aprovechar los mecanismos que brindan los ESBs



- ❑ Otra posible problemática de integración es cómo utilizar un WS Geográfico en un proceso WS-BPEL
- ❑ Habría que brindar una interfaz SOAP para el WS Geográfico
 - Para esto existe una recomendación del OGC
- ❑ Se podría también utilizar los mecanismos de los ESBs para implementar esta conversión

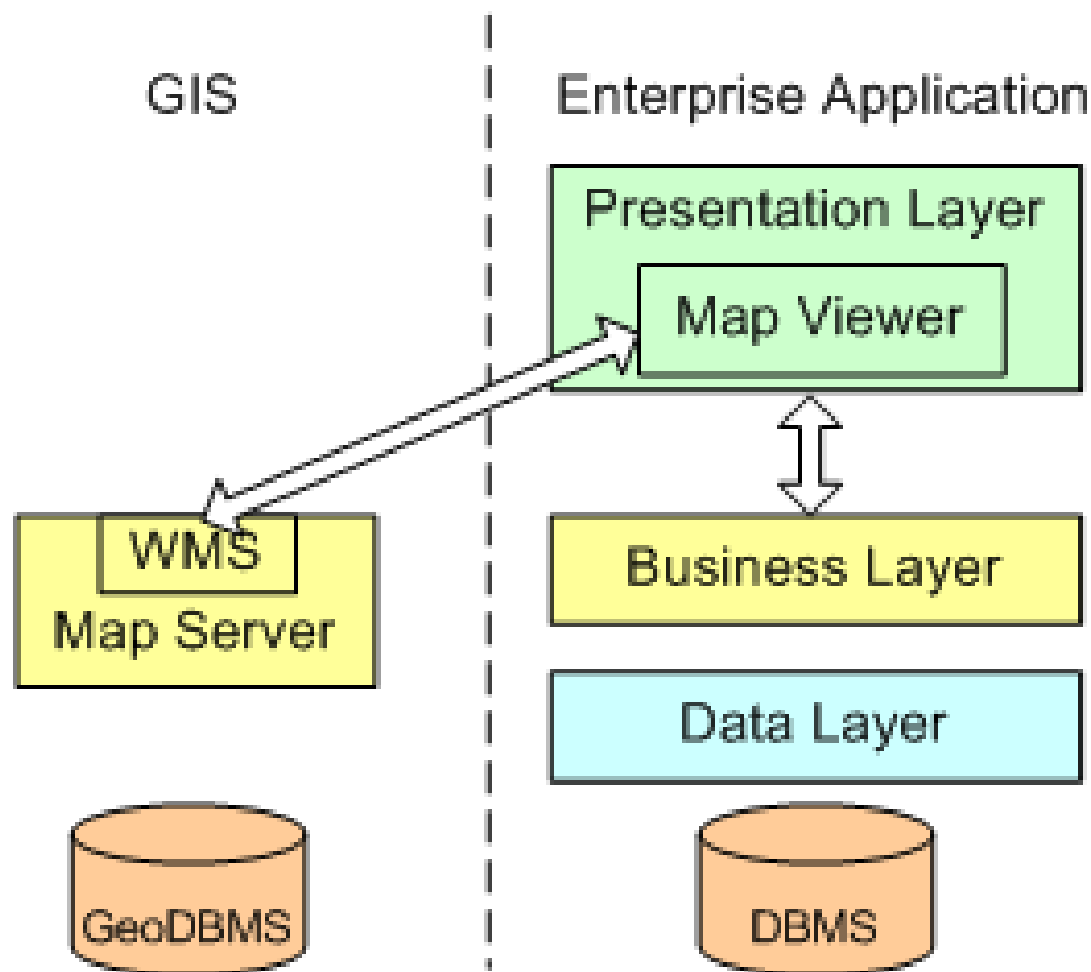


Integración a nivel de UI

- ❑ Las organizaciones están notando los beneficios de agregar mapas a sus aplicaciones existentes
- ❑ Una posible alternativa es agregar un Map Viewer a la UI de las aplicaciones empresariales



Escenarios de Integración



Integración a nivel de UI

- ❑ Dada la heterogeneidad tecnológica, muchas veces la mejor opción es utilizar un viewer basado en javascript (ej. OpenLayers)
- ❑ Si una organización no tiene su propio dataset geográfico, podría utilizar mapas de propósito general (ej, google maps)



Integración a nivel de UI

- ❑ Otro posible escenario es enriquecer los datos brindado por un Map Viewer con datos de Negocio
- ❑ En este caso, se puede utilizar la integración a nivel de servicios descripta previamente
- ❑ Si esto no es posible, la integración se debe hacer a nivel del cliente
 - Qué problemáticas presenta esta solución?



- ❑ Varias tecnologías empresariales están siendo extendidas para tener soporte geográfico
- ❑ Algunos Ejemplos:
 - Soporte en BD (postgis, etc)
 - Hibernate Spatial
 - GeoMajas
 - GeoFaces



Actividad General

- ❑ 3 conceptos importantes de la clase



- ❑ 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/>

Referencias

- ❑ Rienzi, B., Sosa, R., Foti, P., González, L.: Benefits and challenges of using geographic information systems to enhance social security services. (2010).
- ❑ Bruno Riezi, Laura González: Towards an ESB-Based Enterprise Integration Platform for Geospatial Web Services. (2013).

