

Maestría en Ingeniería

Curso de Arquitectura de Software

Sesión 6

Fernando Barraza A.
fbarraza@javerianacali.edu.co

Sesión 6



- Objetivo: Exponer las técnicas y los sistemas dominantes más utilizadas en la implementación de las AS.
- Temas:
 - Clasificación de técnicas
 - Mecanismos de interfaz / adaptación
 - Sistemas Dominantes

Técnicas en AS



- Permiten implementar la tecnología siguiendo los modelos de AS
- Principalmente orientadas a resolver la comunicación/integración entre componentes
- Técnicas comunes:
 - Interfaces y adaptadores
 - Objetos remotos
 - Remote Scripting
 - Servicios Web
 - *Middleware*
 - Agentes

Mecanismos de Interfaz e Integración

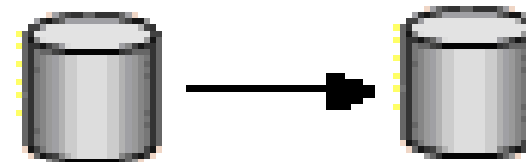
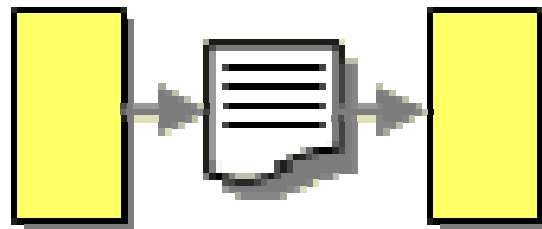


- Replicación de datos
- Interfaces y Adaptadores
 - API's
 - Wrapping
- Mediadores (*Middleware*)

Replicación de datos



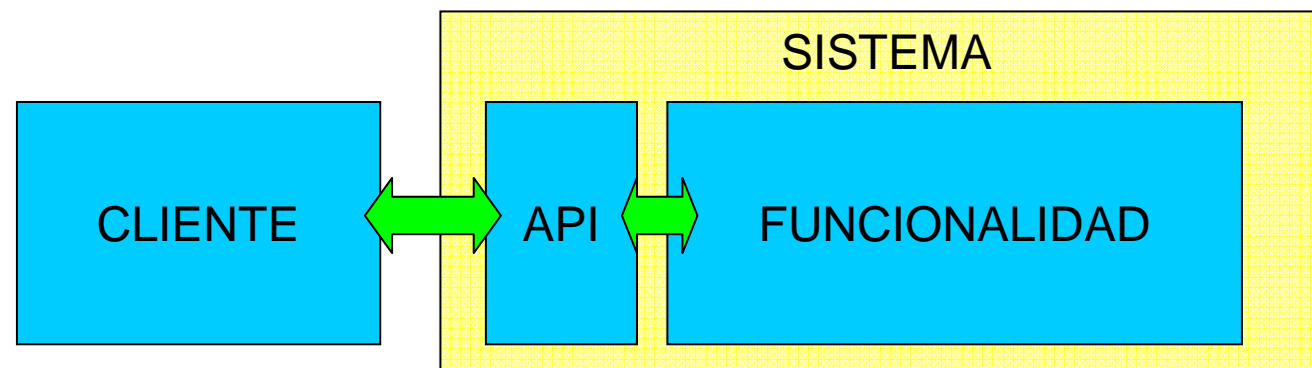
- Se basa en mover datos entre una o mas bases de datos heterogéneas
- Puede incluir servicios de transformación
- Son simples y de bajo costo
- No son recomendables si los datos están ligados a la lógica del negocio



API (Application Program Interface)



- Mecanismo más utilizado actualmente
- Pretende ocultar la complejidad de la implementación de los recursos
- Debe ser publicado y documentado para ser usable



Wrapping

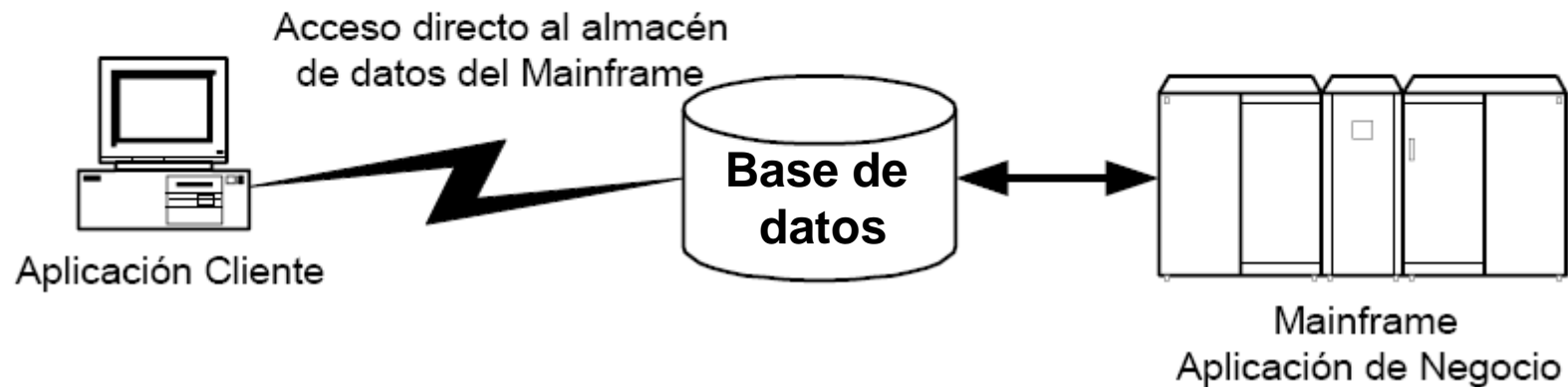


- Envoltorio virtual alrededor de la aplicación que oculta a los clientes su verdadera implementación y la muestra como una aplicación C/S
- Común en integración de aplicaciones legadas
- Se puede implementar a nivel de datos, procesos o presentación

Wrapping de BD



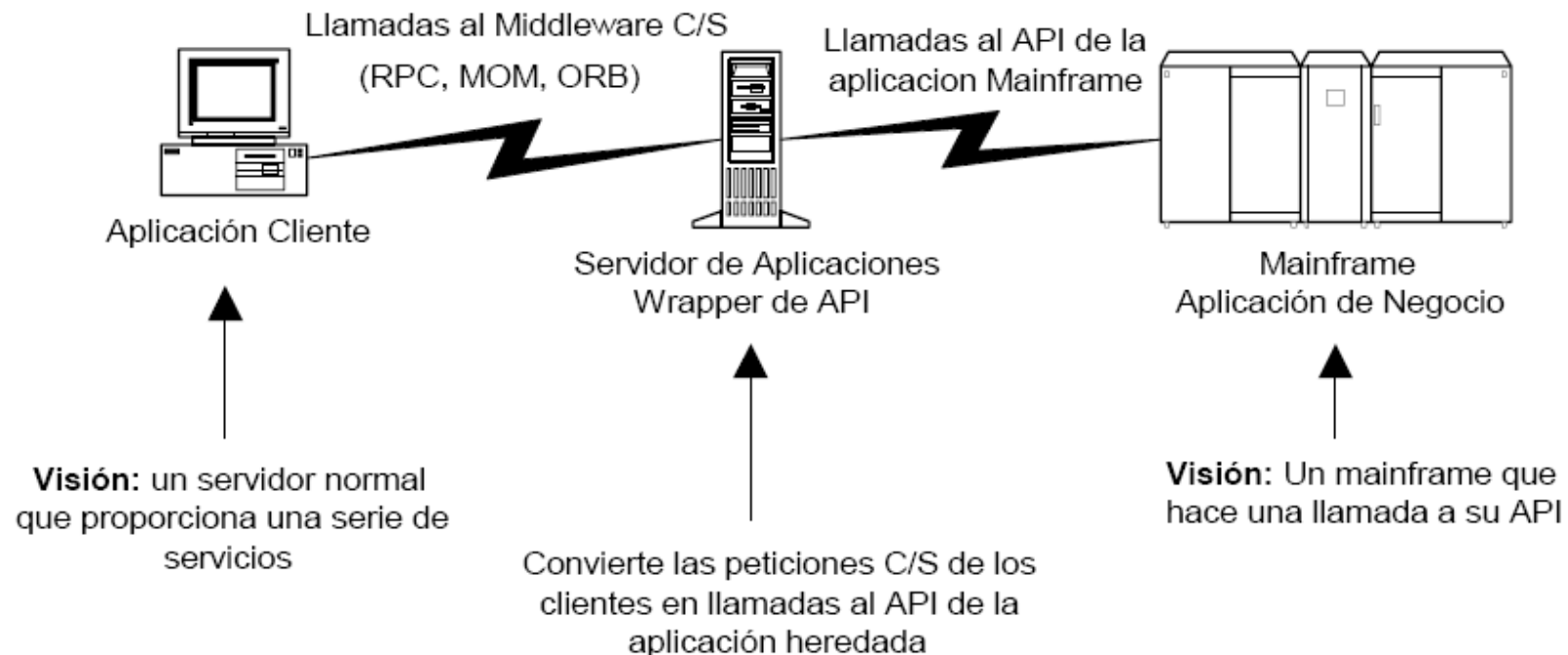
- Utiliza una base de datos intermedia como mecanismo de wrapping
- Ejemplo:
 - Acceso al almacén de datos de una aplicación heredada



Wrapping de API



- Acceso a la Funcionalidad (lógica de negocio)

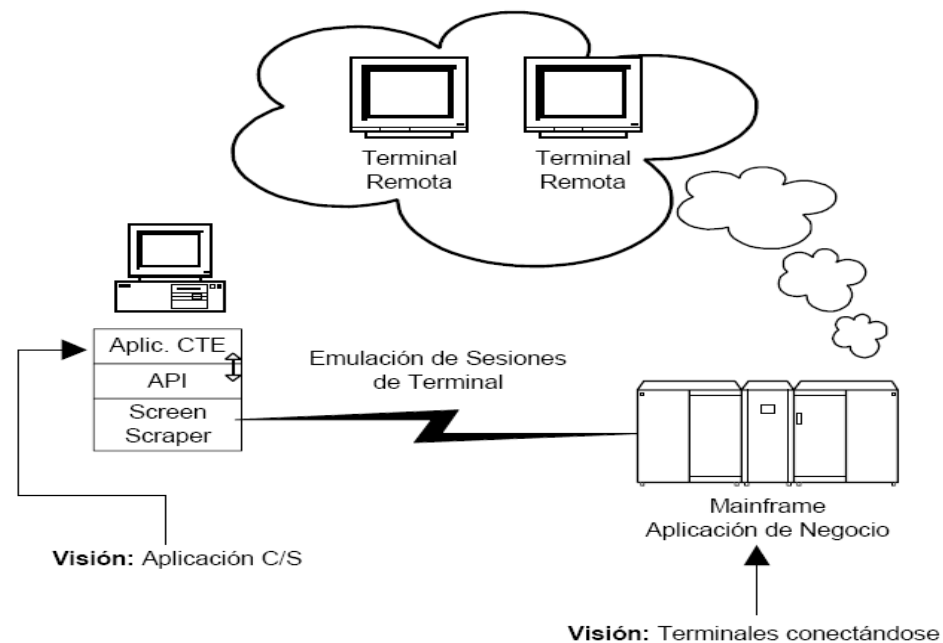


- Scripting : Acceso a la Funcionalidad cuando no hay API

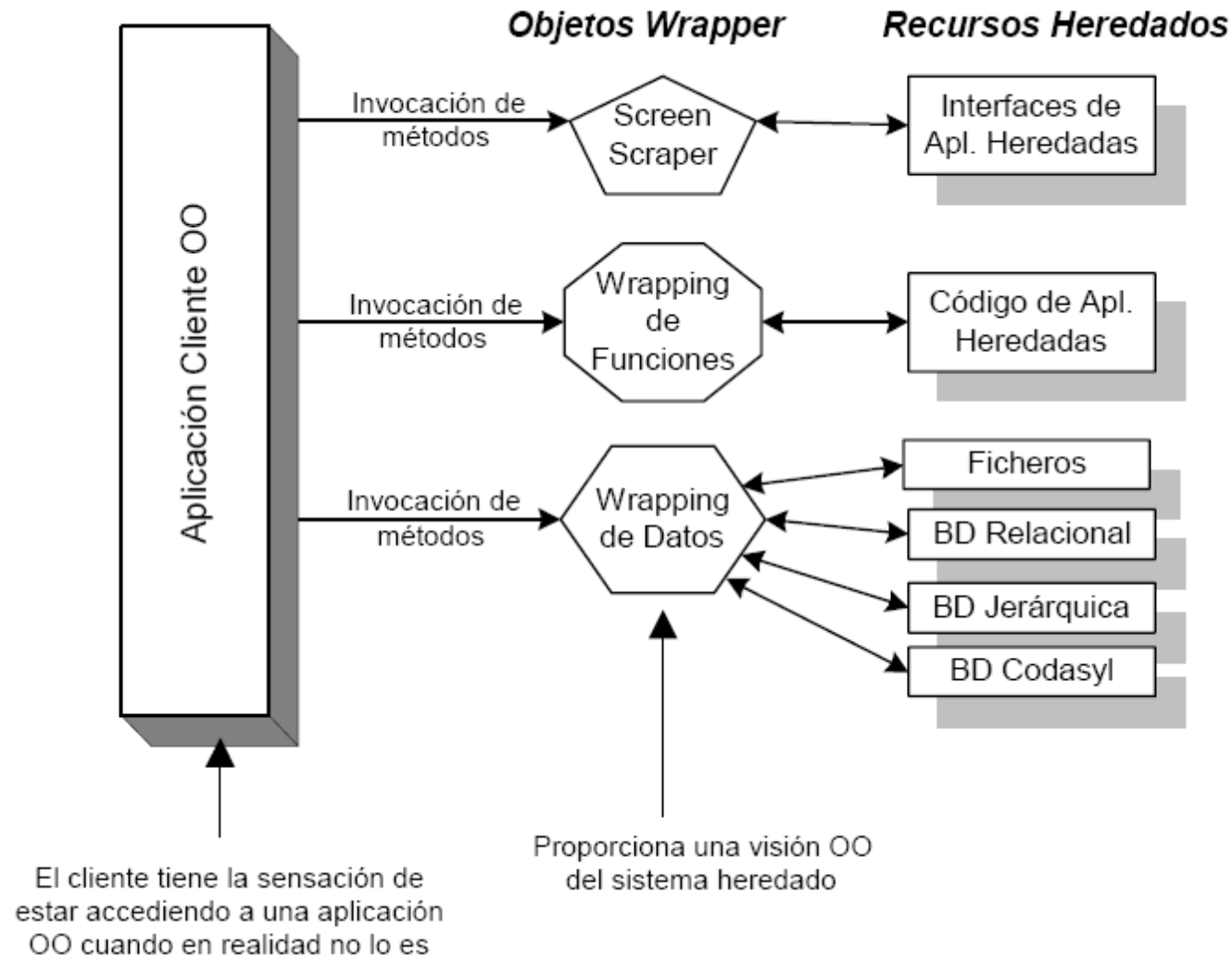
Screen Scraper



- Acceso a la Presentación de la aplicación heredada
- Software que emula a una terminal del mainframe
 - Proporciona un API para el cliente
 - Realiza la interacción con el mainframe tal y como lo haría un terminal
 - Recibe el conjunto de caracteres de la presentación y los interpreta



Object Wrapping



Middleware



- Mecanismo que permite conectar un del sistema (aplicación o base de datos) con otros componentes
- Hacen posible la integración de aplicaciones
- Implementan la arquitectura definida
- Siguen modelos y patrones de integración
- Relacionadas con el dominio del problema
- Tan simple como RMI o tan complejo con TP (Transaction Monitor)

Modelos de Arquitectura Middleware

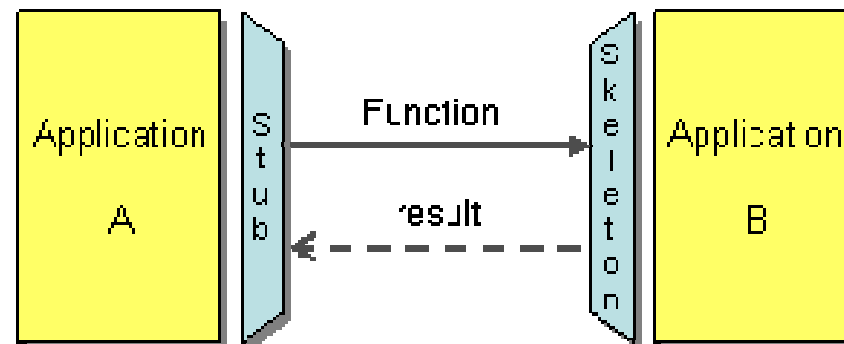


- RPC's
- Message-oriented
- Distributed Objects
- Database-oriented
- Transaction-oriented

RPC Middleware



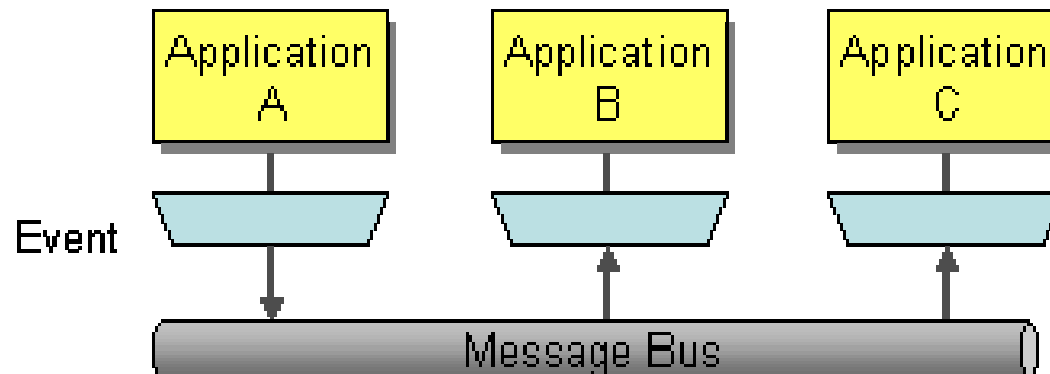
- Modelo point-to-point sincrónico
- Fácil de entender y usar
- Requieren buen ancho de banda
- Ej.: DCE de OSF (Open Software Foundation)



Message-Oriented Middleware (MOM)



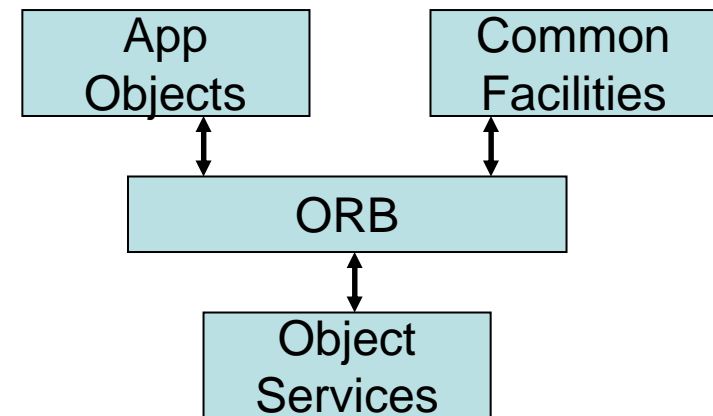
- Asincrónico y concurrente
- Garantiza entrega del mensaje
- Implementa concepto de “*broker*”
- Modelo de colas o proceso-a-proceso
- Ej.: MSMQ (Microsoft), MQSeries (IBM)



Distributed Objects Middlewares



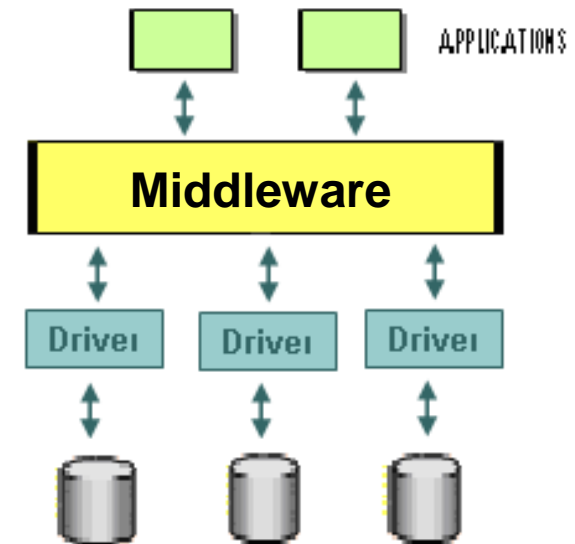
- Ideal en un modelo de computación distribuida con gran número de métodos comunes compartidos
- Método invasivo de integración
- Basado en Object Request Brokers (ORB)
- Ej.: COM+, CORBA.



Database-Oriented Middlewares



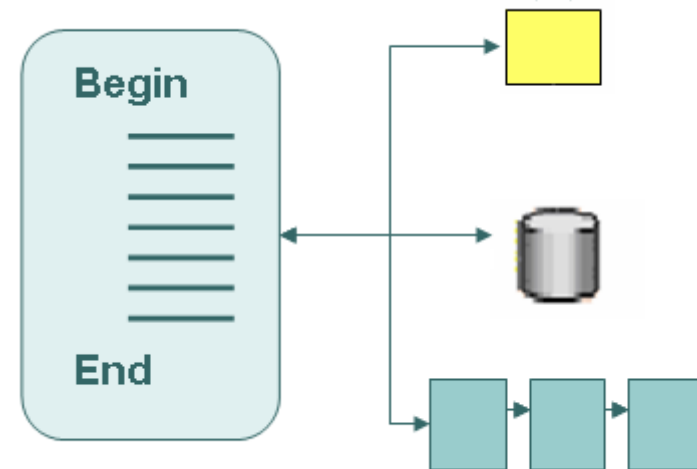
- Soporte a varias BD's
- Múltiples modelos:
 - Relacional
 - Multidimensional
 - Archivos planos
 - Orientado a objetos
- Utilizan como interfaz:
 - SQL Nativo
 - Librerías nativas en C, C++, etc.
 - Interfaces simples como JDBC y ODBC
 - Gateways como EDA/SQL, DRDA y ISO/SAG



Transaction-Oriented Middlewares



- Basado en el concepto de transacción ACID (atómica, consistente, aislada y durable)
- Diferentes mecanismos:
 - Multiplexación
 - Balanceo de Cargas
 - Tolerancia a fallos
- Implementados como TP Monitors y Application Servers



Técnicas: Objetos remotos



- Permite distribuir la ejecución de una aplicación garantizando independencia en sus componentes
- La comunicación entre objetos se hace siguiendo protocolos estándares de comunicación remota (a diferentes niveles del modelo OSI)
- RMI o CORBA
- No es fácil de implementar!

Técnicas: Remote Scripting



- Ejecución remota de programas de scripting
- Elimina complejidad de invocación de objetos remotos (Ej.: CORBA)
- Uso intensivo de estándares XML
- Diversas técnicas disponibles
 - XML-RPC (en desuso)
 - Ajax (de moda!)

AJAX (*Asynchronous JavaScript And XML*)



- **XHTML** (o HTML) y hojas de estilos en cascada (**CSS**) para el diseño que acompaña los datos
- **Document Object Model** (DOM) accedido con un lenguaje de scripting por parte del usuario
 - Especialmente implementaciones ECMAScript como **JavaScript** y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto **XMLHttpRequest** para intercambiar datos asincrónicamente con el servidor web.
 - En algunos frameworks y en algunas situaciones concretas, se usa un objeto **iframe** en lugar del **XMLHttpRequest** para realizar dichos intercambios.
- **XML** para la transferencia de vuelta al servidor
 - Cualquier formato puede funcionar, incluyendo HTML preformateado como texto plano o JSON.

Cambio de Paradigma con AJAX



Web-Based Application Model



"Dumb Client" simply renders HTML in the browser

HTML (UI content)



Server provides business logic and generates UI markup

Browser-Based Application Model



Application UI logic runs in browser.

Data (e.g. via SOAP)



Server exports application API. Performs core business logic

Técnicas: Web Services



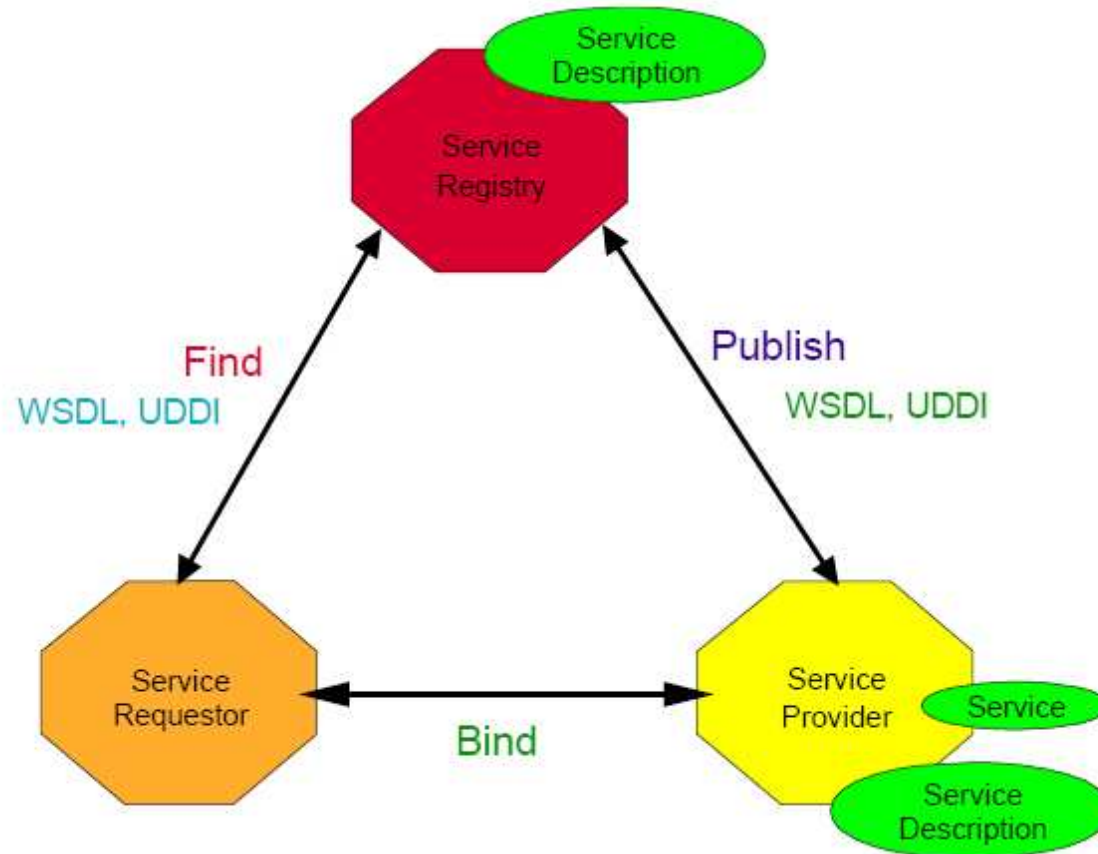
- Permite acceder a lógica encerrada en sistemas propietarios sin comprometer el sistema mismo (SOAP)
- Ideal en enfoques de integración orientados a servicios (UDDI)
- Elimina complejidad de otras tecnologías middleware tales como DCOM y CORBA
- Implementable entre plataformas J2EE y .NET
- Agrega sobre costos en el desempeño

Que es un servicio?

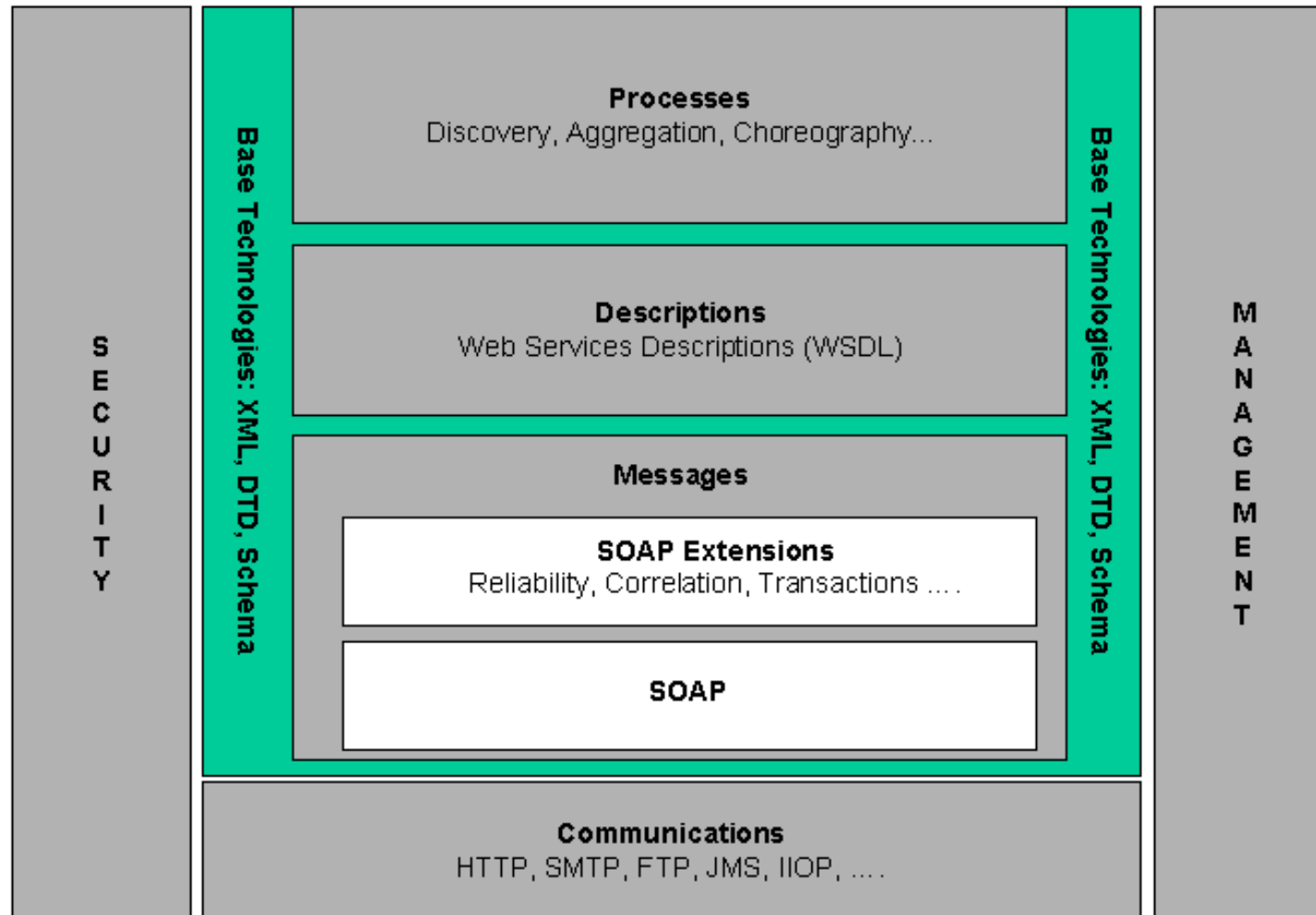


- Es una función que es invocada para cumplir tareas específicas
- Los servicios ...
 - Exponen una interfaz bien definida (API)
 - Esconden detalles de implementación
 - Se pueden invocar mediante mecanismos con estándares abiertos
- Se implementan generalmente mediante Web Services

Funcionamiento de Web Services



Arquitectura en pila de Web Services



Sistemas dominantes



- Servers
- Portales
- Workflow
- Monitores de transacciones
- Mensajería
- Servicios de Directorios
- Persistencia

Servidores Web



- Un **servidor web** es un programa que implementa el protocolo HTTP (*hypertext transfer protocol*) para transferir documentos HTML (hypertext markup language)
- El contenido HTML puede ser estático (páginas HTML) o dinámico (generado por programas CGI, servidores de aplicaciones)

Consideraciones para un Servidor Web



- Escalabilidad
- Independencia del servidor de aplicaciones
- Portabilidad entre plataformas
- Mecanismos de seguridad provistos

Servidores Web



- Apache
 - Primer servidor más utilizado en Internet (65% aproximadamente)
- Internet Information Server
 - Segundo servidor Web más utilizado en Internet
 - Primer servidor Web en entornos corporativos.
- Cherokee
 - Open-Source con mejoras a Apache

Apache

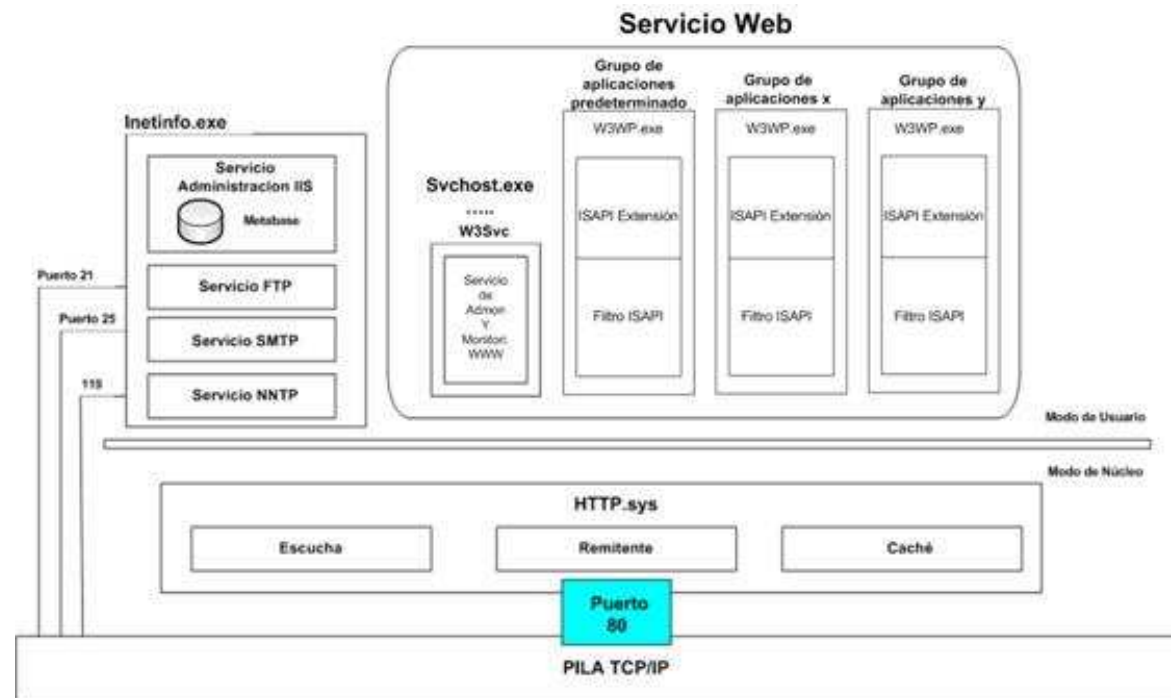


- Las principales metas de su diseño son: velocidad, simplicidad, multiplataforma y facilidad del desarrollo distribuido.
- Todo el código de fuente de Apache está escrito en C, con un total aproximado de 185.000 líneas de código.
- Es un servidor basados en procesos, utilizando la técnica pre-fork

MS IIS



- La versión 7.0 con Windows Vista, es más modular y donde los procesos se ejecutan con una cuenta específica lo que reduce la vulnerabilidad del sistema.



Servidores de Aplicaciones



- Es un programa que maneja todas las operaciones ejecutadas por las aplicaciones
- Definen un recurso específico para la lógica de integración
- Coordinan la conexión a los otros recursos
- Se ubican entre el cliente (o servidor web) y la base de datos
- Normalmente considera redundancia, monitores de alta disponibilidad, servicios para distribución de aplicaciones de alto desempeño y soporte a transacciones complejas de bases de datos.

Algunos Servidores de Aplicaciones

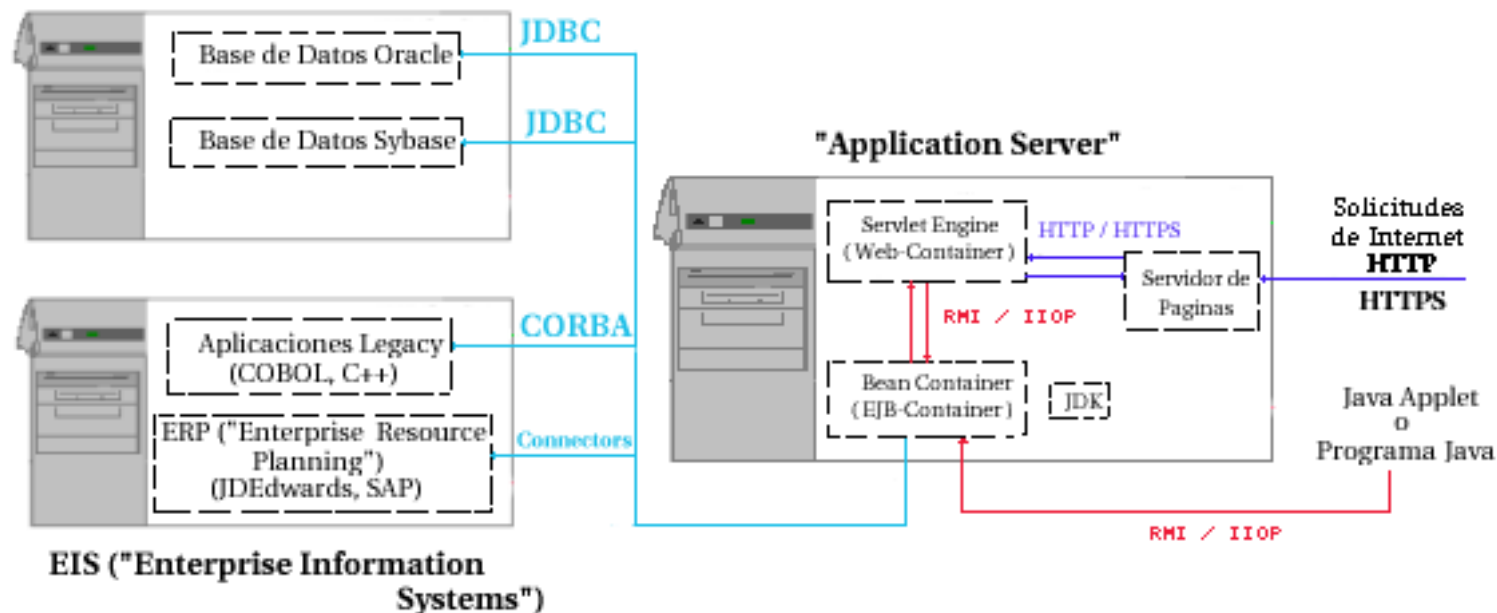


- **Tomcat:** Proyecto open-source hospedado por Apache. Es la implementación de referencia para la API Servlet
- **Jboss:** Implementación open-source de J2EE escrita en Java que se basa en la especificación de Enterprise JavaBeans para su funcionalidad
- **BEAWebLogic:** Comercial. Implementa de forma completa el estándar J2EE standard incluyendo EJB's.
- **iPlanet:** Servidor desarrollado por Sun Microsystems. Flexible y altamente configurable. Disponible con y sin capacidades J2EE.
- **IBM's Websphere:** También disponible con y sin capacidades EJB, es una extensión del servidor HTTP de Apache HTTP.
- **Otros:** Oracle Application Server, Jetty, etc.

Servidores de Aplicaciones J2EE



- Un servidor de aplicaciones que sigue el estándar J2EE consta de:
 - Un Servlet Engine: Ejecución de aplicaciones JSP's y Servlets
 - Un EJB Engine (Container): Ejecución de aplicaciones desarrolladas utilizando EJB's

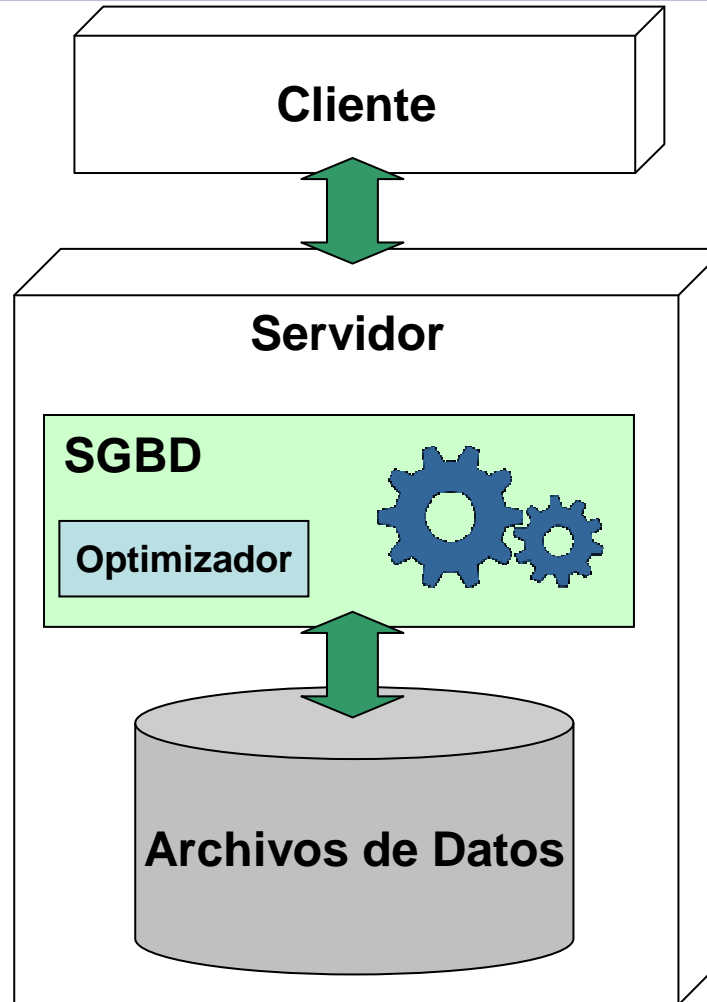


Servidores de Bases de Datos

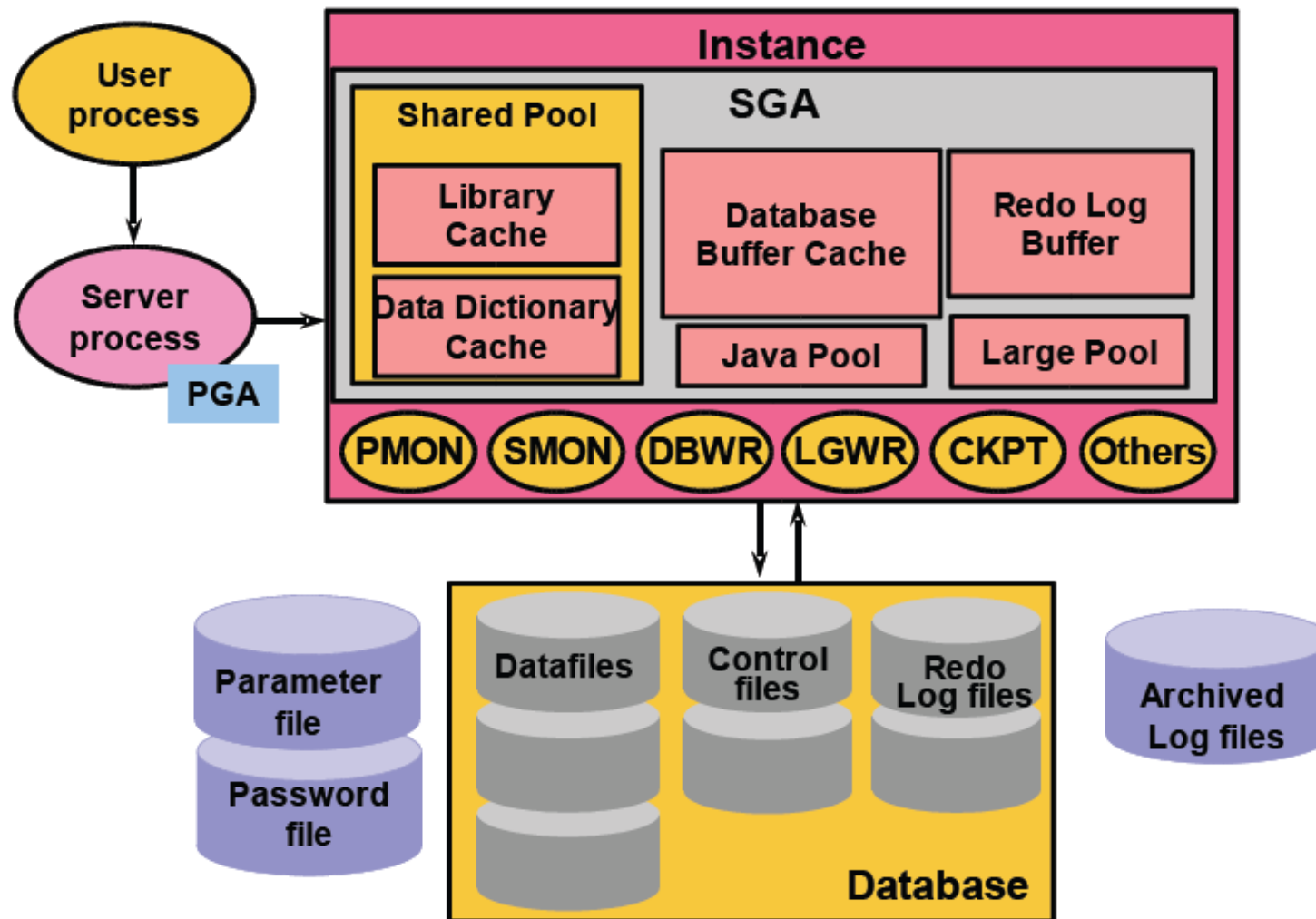


- Proveen persistencia y acceso a los datos
- Pueden ser relacionales, orientados a objetos u objeto-relacional
- Proveen acceso mediante lenguajes de consulta como SQL, OQL, etc.
- Ofrecen capacidades de gestión y monitoreo de transacciones

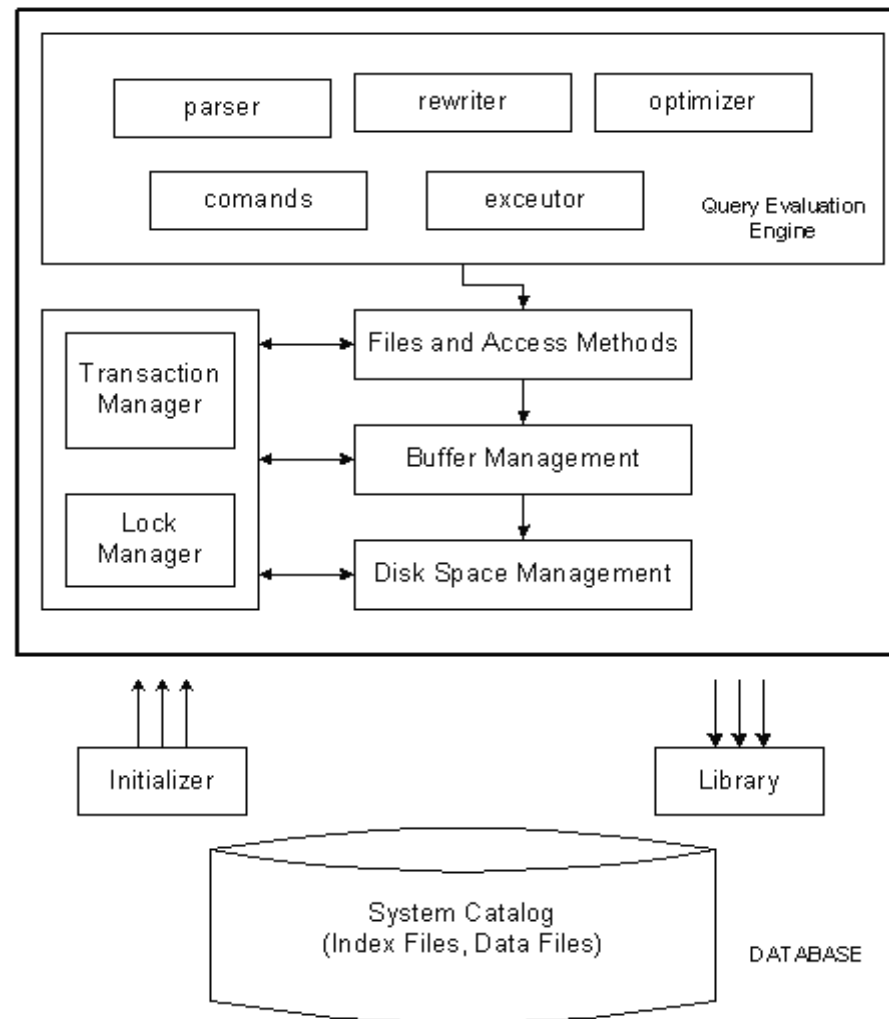
Arquitectura de Servidor de BD



Arquitectura del SGBD Oracle



Arquitectura PostgreSQL

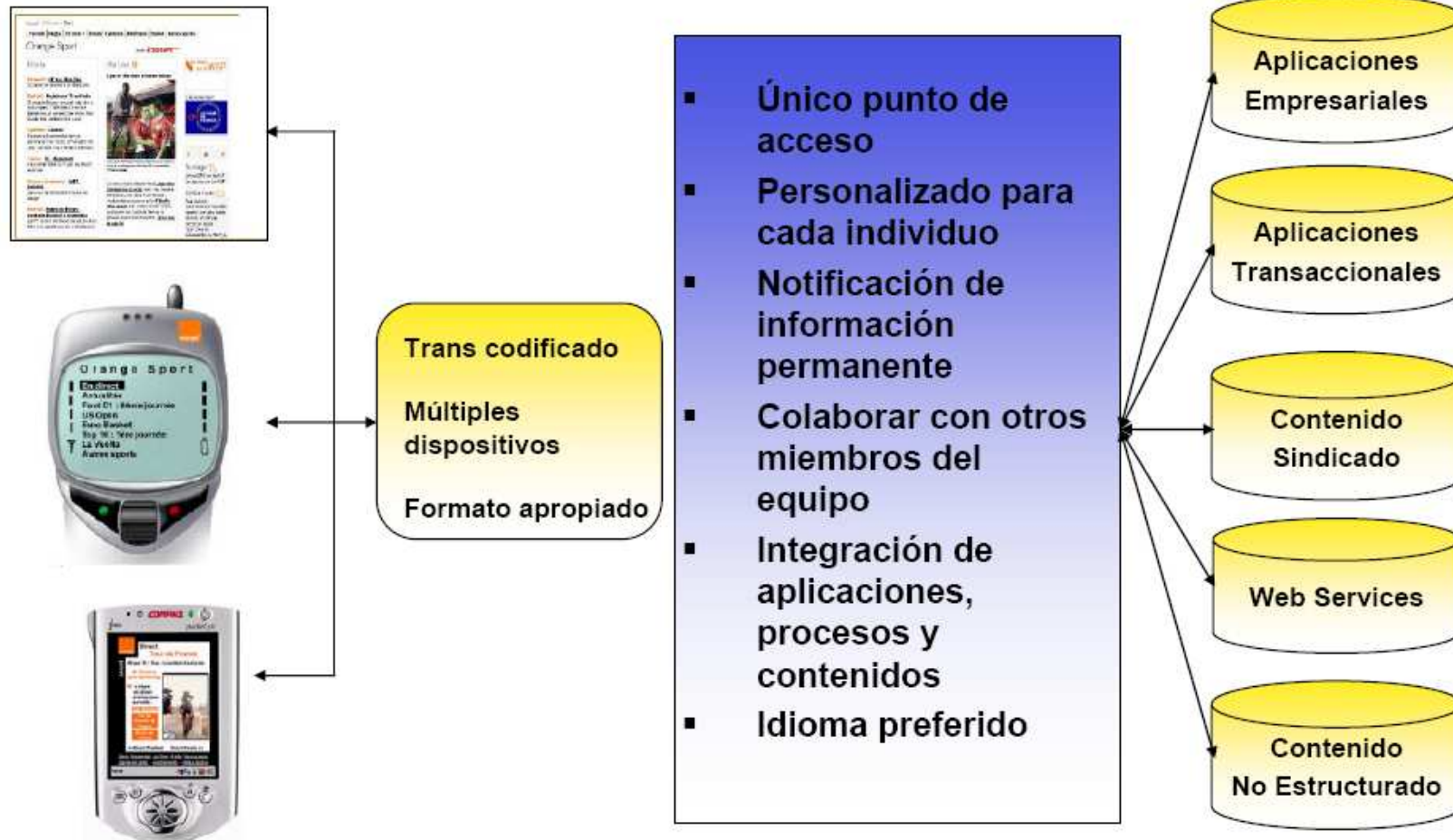


Portales



- Dan acceso unificado a todos los recursos internos de una aplicación y su contenido.
- Implementa un ambiente colaborativo y unificado de trabajo
- Provee una interfaz de usuario personalizada
- Se puede acceder a través de múltiples dispositivos

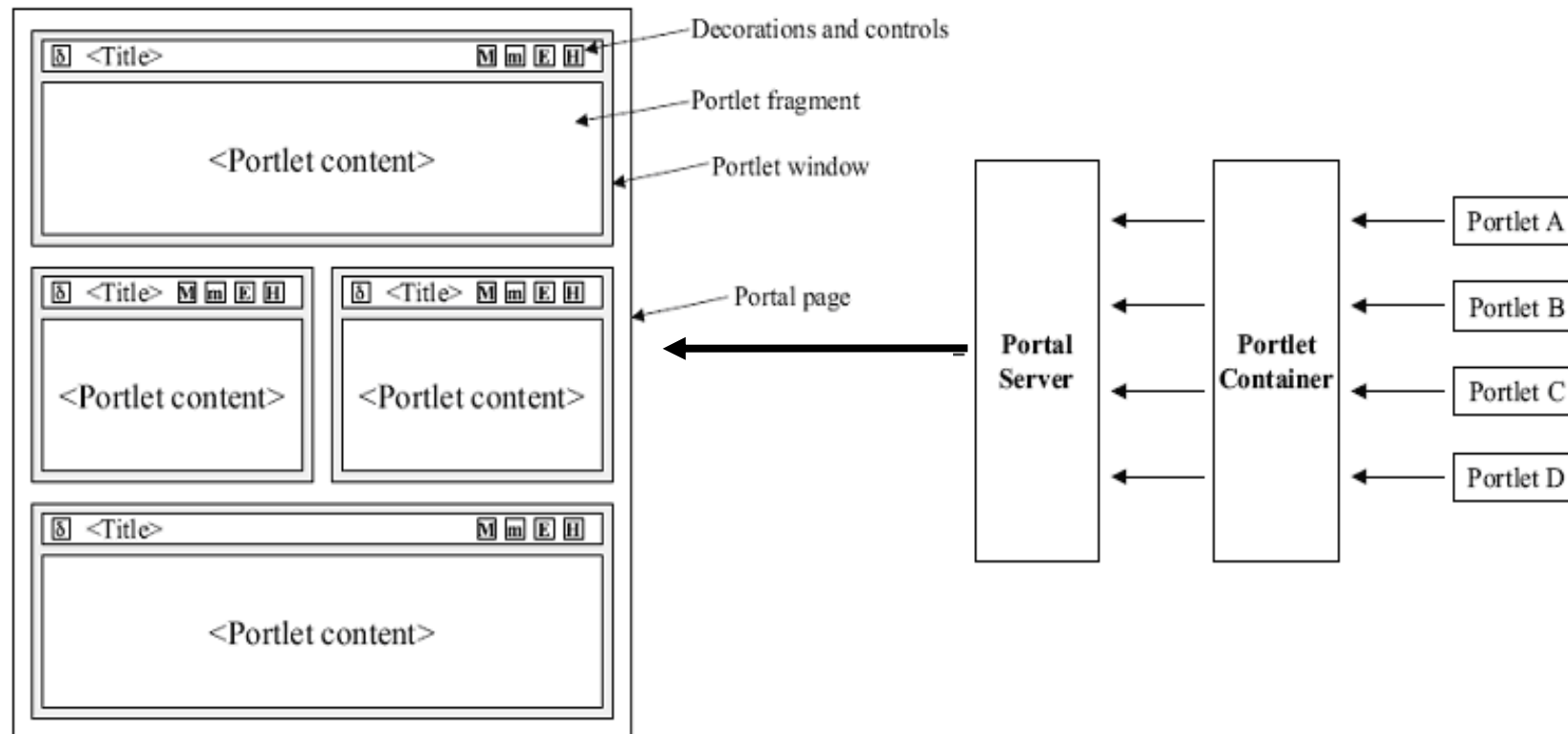
Funcionalidad de un Portal



Elementos de un Portal



Aplicaciones web distribuidas altamente modulares representadas como **portlets** que son agregadas a la página del portal



Relación entre Servlets y Portlets



PORTLET

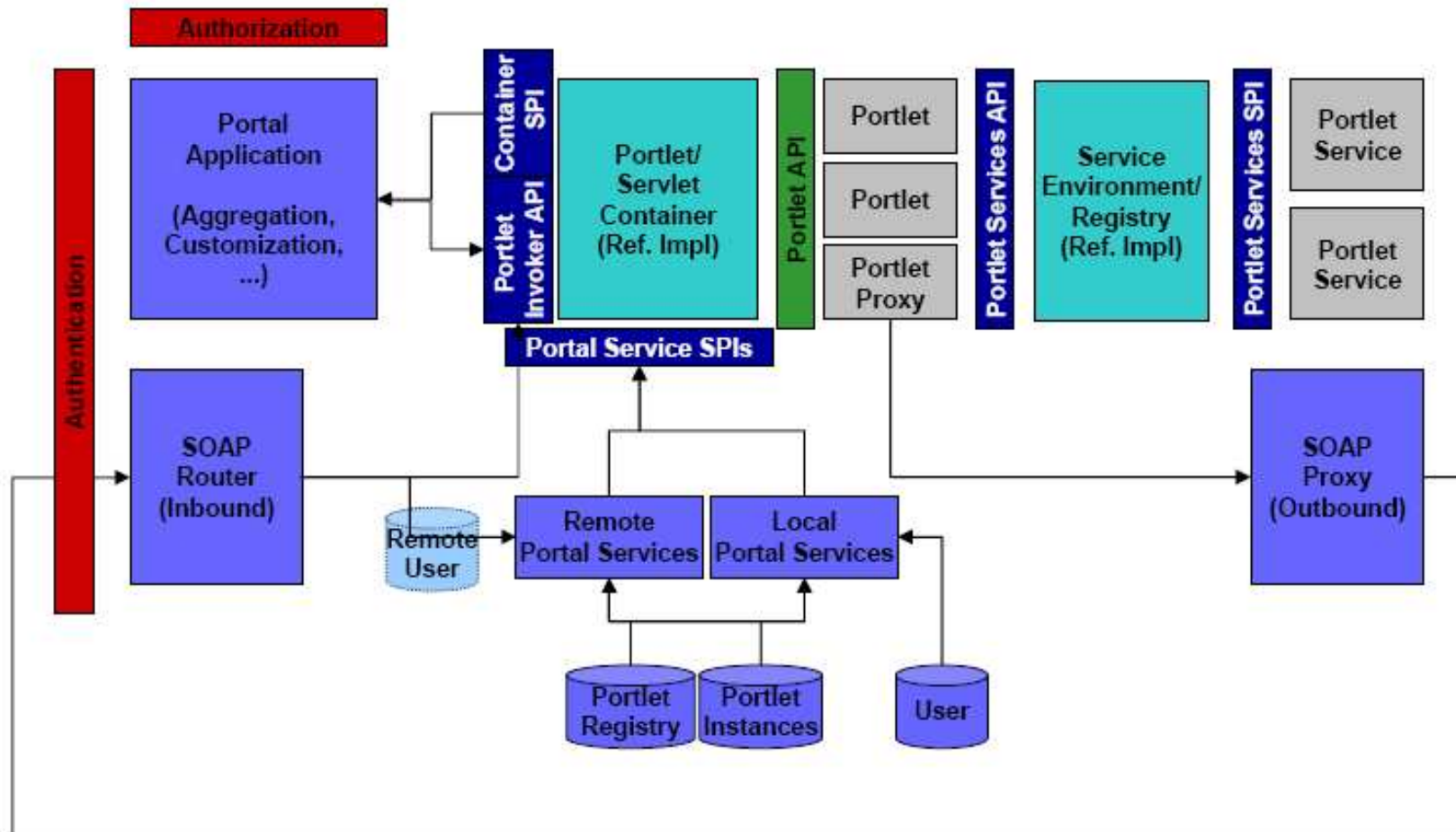
Genera fragmentos de marcado
No se enlaza directamente a un URL
Interactúa con un cliente Web a través de un portal
Pueden existir instancias múltiples en una pagina de portal
Accede a datos de configuración y perfil del usuario

SERVLET

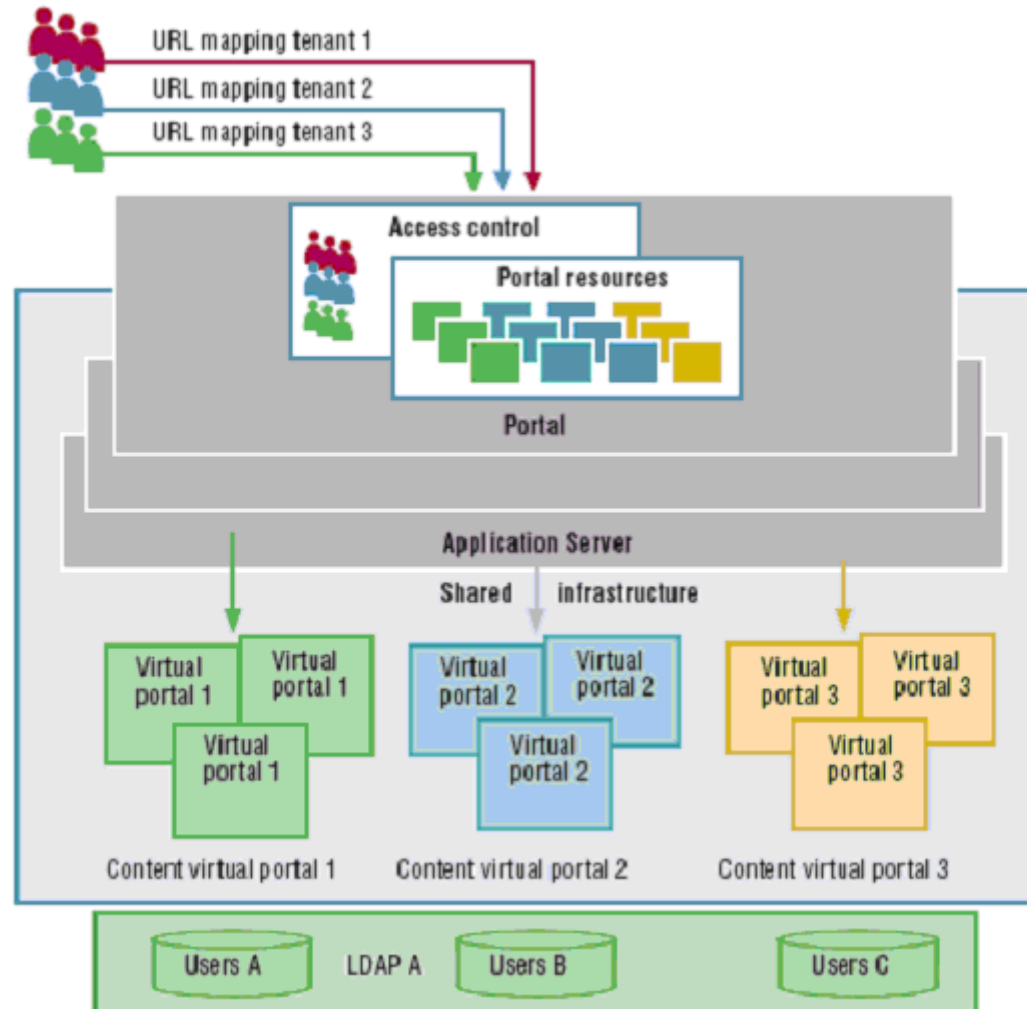
Manejado por un contenedor
Genera contenido dinámico
Usa el paradigma de request /response

Genera documentos completos
Se enlaza directamente a un URL
Interactúa con clientes Web únicamente

Arquitectura de un Portal



Implementación Multisitios (Portales Virtuales)

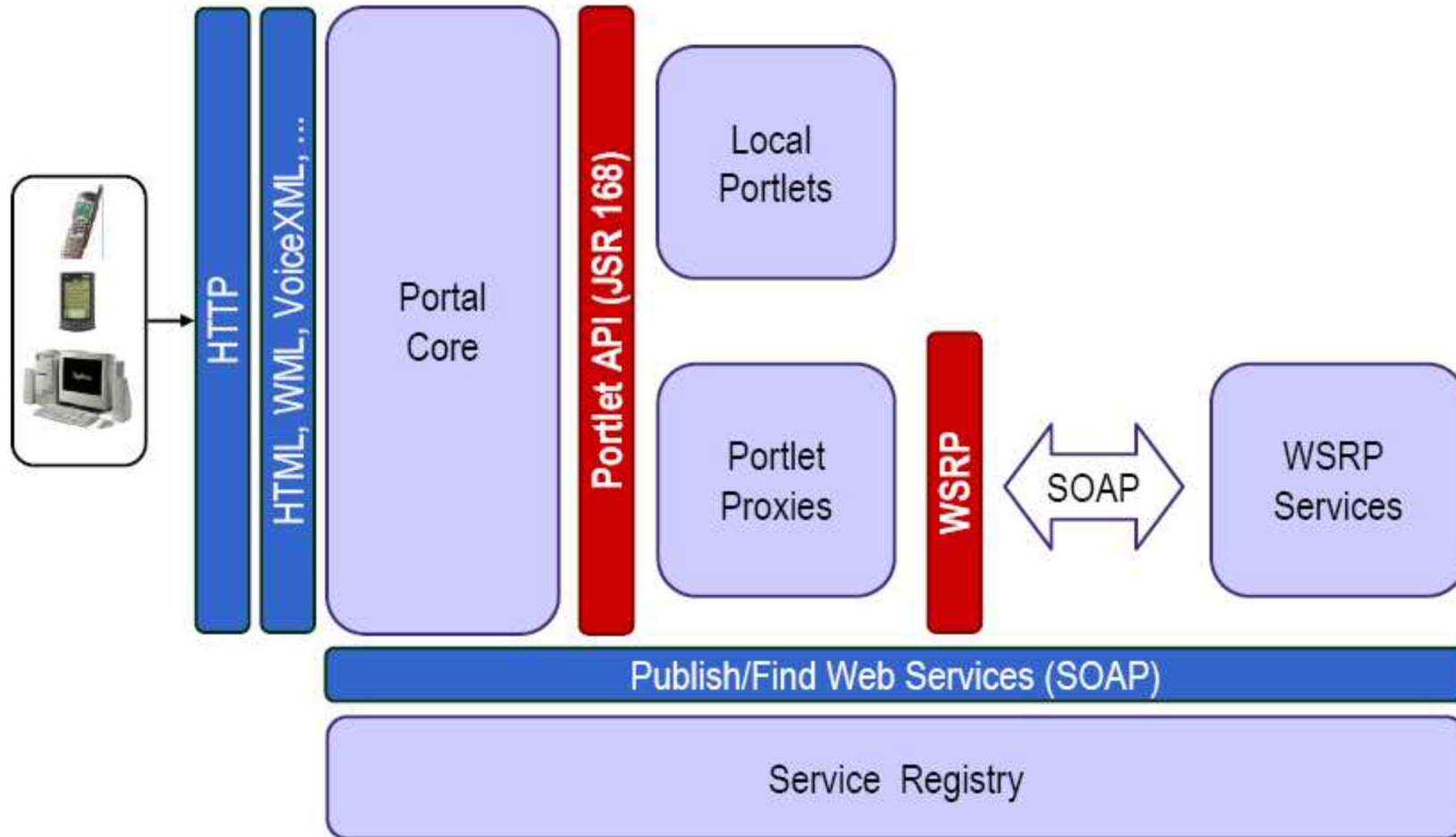


Estándares de Portales

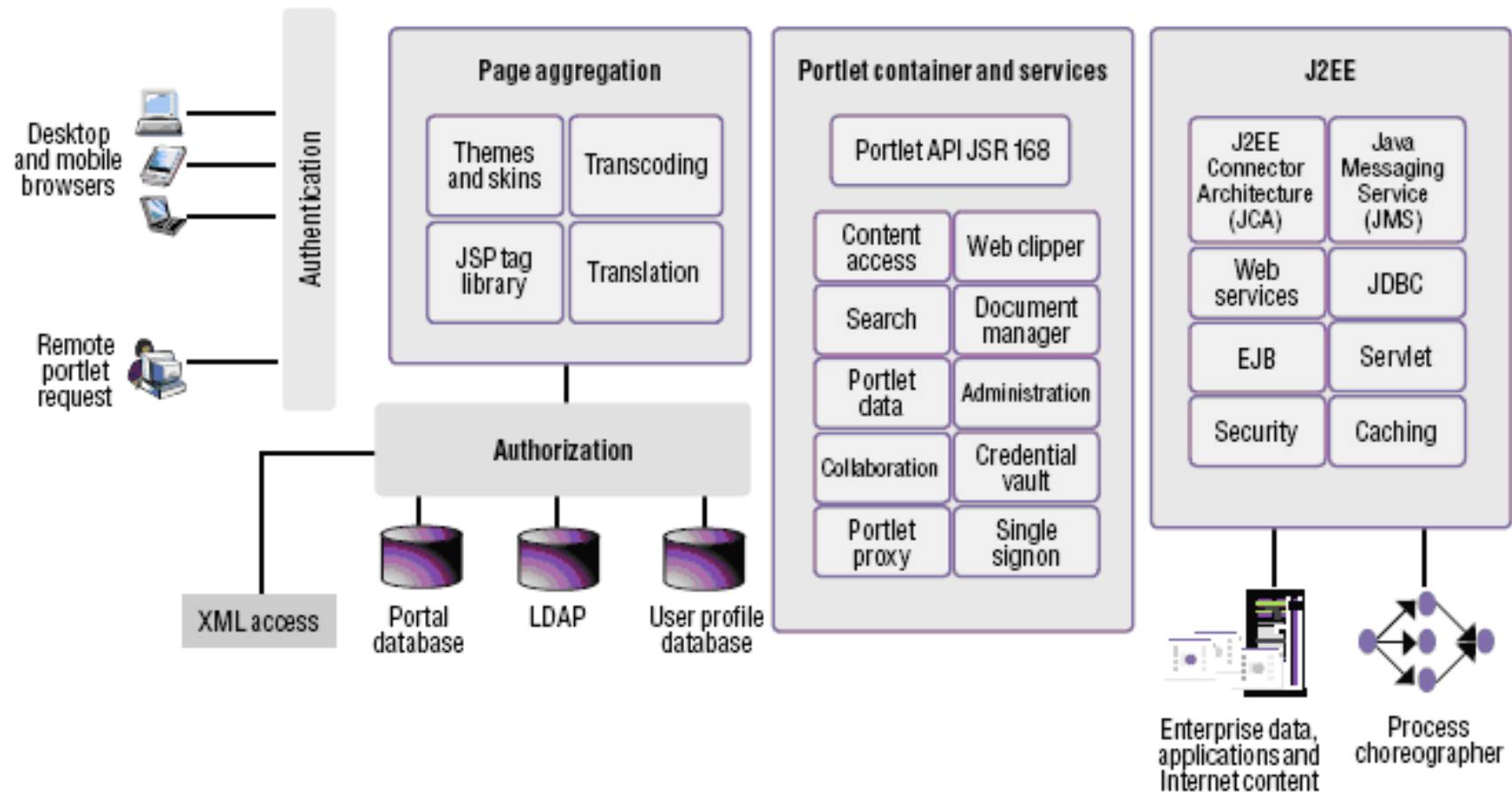


- Java Portlet API
 - Logra interoperabilidad entre portlets locales y portales
 - Estandarizado por la especificación JSR-168
- Web Services for Remote Portlets (WSRP)
 - Logra la interoperabilidad entre servicios web y portales
 - Estandarizado por el OASIS WSRP Workgroup
- Portlet API y WSRP son valiosos unidos
 - Portlets escritos en el Java Portlet API pueden ser “wrappeados” en servicios WSRP y publicados en directorios
 - Los servicios WSRP pueden ser descubiertos y ligados a los Portlets

Portlet API y WSRP



Arquitectura IBM Websphere Portal

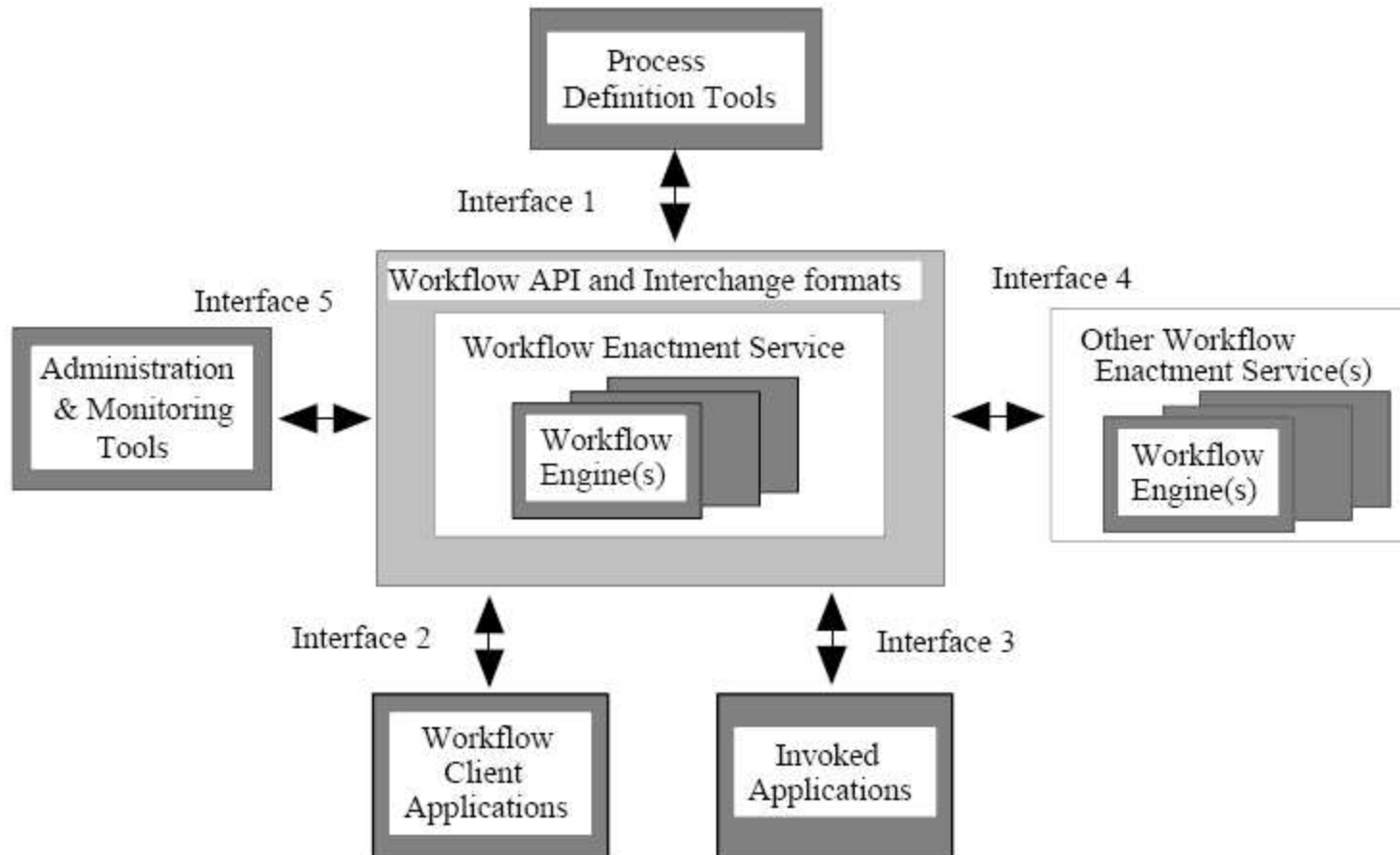


Sistemas de Workflow



- Permiten automatizar la ejecución de procesos de negocios
- Son específicos al dominio de la aplicación
- Los procesos están implementados por funcionalidades de aplicaciones de software
- Se describen por lenguajes para modelar procesos

Modelo de referencia de Arquitectura del WfMC



Workflows disponibles



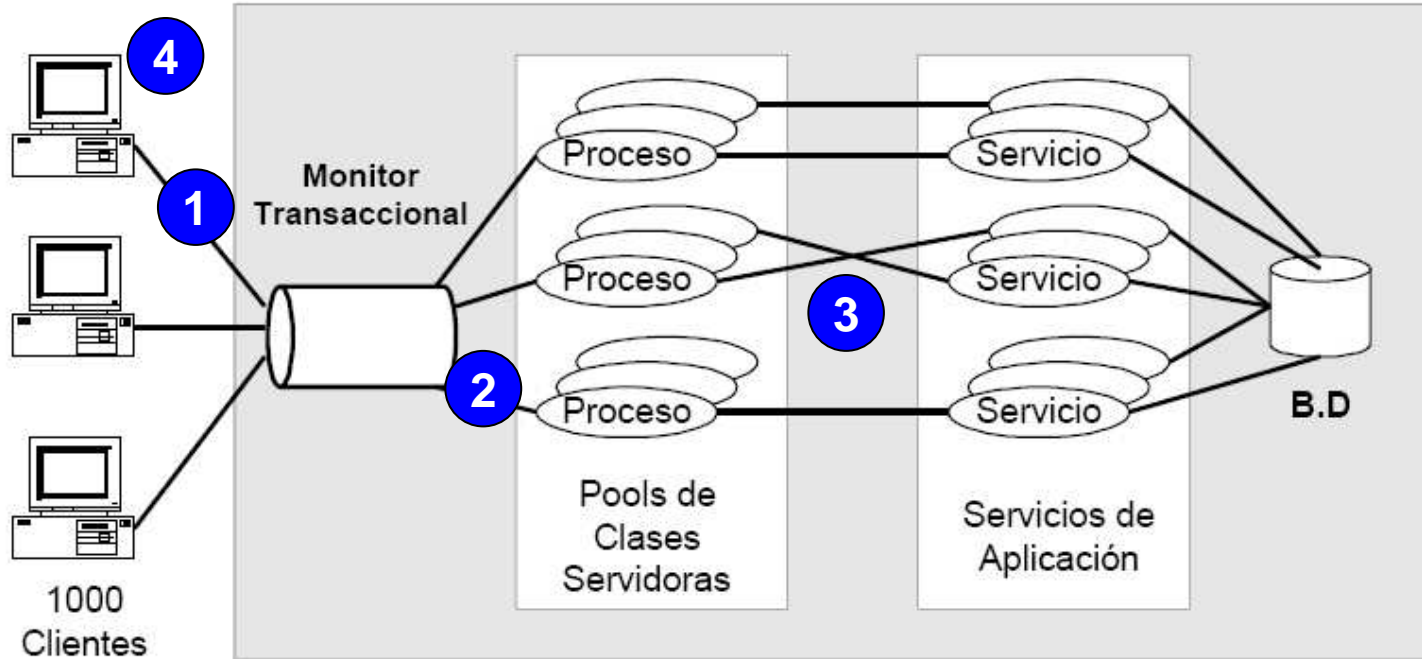
- Oracle Workflow
 - Soporta workflow de negocios mediante la integración de mensaje point-to-point, hubs de mensajes y aplicaciones distribuidas.
- Microsoft Windows Workflow Foundation (antes WinFX)
 - Soporta workflow humanos y de sistemas para aplicaciones de negocios, manejo de documentos, composición de servicios, reglas de negocio y administración de sistemas
- IBM Websphere Workflow
 - Soporta procesos largos de negocios con interacción humana y de sistemas

Monitores transaccionales



- Sistema especializado en la ejecución de transacciones
 - Gestión de procesos y recursos
 - Creación y asignación de procesos a clientes
 - Compartir, reparto y asignación de recursos
 - Balanceo de cargas
 - Gestión de transacciones para garantizar las propiedades ACID
 - Gestión de la comunicación cliente-servidor
- Se sitúa entre el cliente y la aplicación servidora transaccional

Esquema de un MT



1. El cliente realiza un petición.
2. El MT asigna un proceso servidor disponible al cliente.
3. El proceso servidor se conecta con el servicio de aplicación que proporciona los servicios requeridos por el cliente.
4. Se devuelve el resultado al cliente y el proceso servidor queda disponible para otros clientes.

Cuando usar un MT ?



- N° de usuarios → 250 usuarios
- Rendimiento → Más de 50 tps
- Disponibilidad
- Múltiples nodos
- Escalabilidad
- Interoperabilidad
- Sistemas críticos
- Sistemas heredados
- Datos no relacionales
- BD heterogéneas
- Seguridad

Sistemas de Mensajería



- Proveen paso eficiente de mensajes
- Garantizan consistencia y seguridad
- Muy utilizado en entornos corporativos y en conjunto con sistemas de workflows
- Algunos productos:
 - IBM MQSeries
 - Microsoft MQ

IBM MQSeries



- Producto IBM, líder del mercado con un 65% de participación
- Corre en IBM OS/390, Pyramid, Open VMS, AIX, NCR Unix, HP-UX, Solaris, OS/2, Windows NT, SCO Unix, Mac OS, Linux Red Hat.
- Basado en tres capas: Aplicación, Backbone de mensajes, Comunicación
- Soporte a mensajes de 100 Mb

Servicios de Directorio



- Utilizados para proveer un acceso unificado a un conjunto de información sobre una red.
- Habitualmente, almacena la información de login (usuario y contraseña) y es utilizado para autenticarse
- **LDAP** (*Lightweight Directory Access Protocol*) es un protocolo de red que permite el acceso a un servicio de directorio ordenado y distribuido

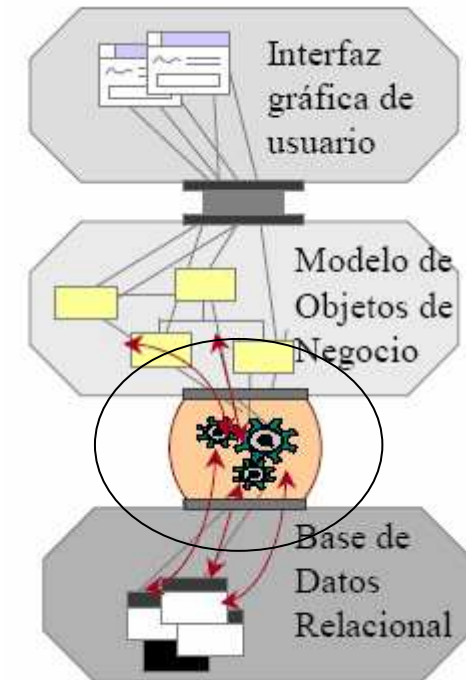
Implementaciones LDAP



- Microsoft Active Directory
- Novell Directory Services
- iPlanet
- OpenLDAP
- Red Hat Discovery Services

Persistencia

- Hacer permanente los objetos de negocio en una base de datos
- Si la base de datos es relacional, se debe utilizar un mecanismo de mapeo Objeto-Relacional
- Existen productos como Hibernate que facilitan la persistencia
- O componentes como IBATIS, JPA, etc.



Hibernate



- **www.hibernate.org** - Open Source
- Provee una capa java para interactuar con la base de datos
 - Técnicamente es un conjunto de librerías java para acceder a la BD
 - Los objetos son puestos en cache en la JVM y mapeados al modelo de objetos