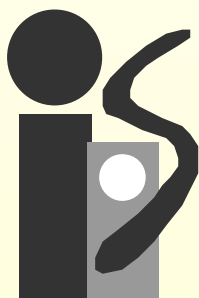
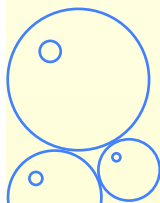


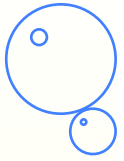
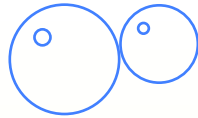
# Web Semántica: Orígenes y Futuro



J.A. Bañares Bañares

Departamento de Informática e Ingeniería de Sistemas  
C.P.S. Universidad de Zaragoza

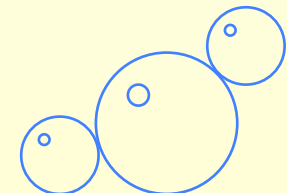
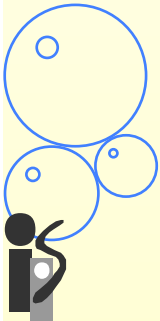


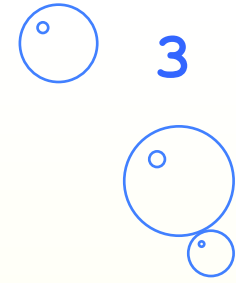
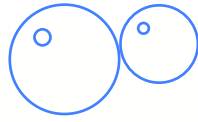


# Índice

---

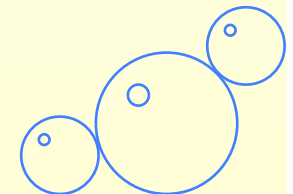
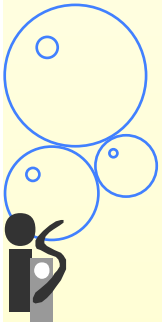
- **Los orígenes**
  - De los documentos hiperenlazados al comercio electrónico (B2C y B2B)
- **Servicios Web en la Actualidad:**
  - Framework de Servicios Web
  - Arquitecturas Orientadas a Servicios
  - Computación Orientada a Servicio
  - Procesos de Negocio
  - Web 2.0
- **Servicios Web el Futuro**
  - Web 3.0. The Semantic Web Service Architecture.
  - ...

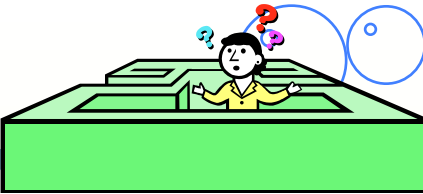




**De la Web al comercio electrónico**

# LOS ORÍGENES





Interoperability Issues

Basic Profile  
1.1  
W3C  
Final Specification

Attachments Profile  
1.0  
W3C  
Final Specification

Single SOAP Binding Profile  
1.0  
W3C  
Final Specification

Basic Security Profile  
1.0  
W3C  
Working Draft

REL Token Profile  
1.0  
W3C  
Working Draft

SAML Token Profile  
1.0  
W3C  
Working Draft

Confession Chain  
Attaches Mechanism (CCAM)  
1.0  
W3C  
Final Specification

Reliable Anonymous  
Messaging Profile (RAMP)  
1.0  
W3C  
Final Specification

Standards Bodies

W3C - The World Wide Web Consortium  
IETF - Internet Engineering Task Force  
ISO - International Organization for Standardization  
ITU - International Telecommunication Union  
IEEE - Institute of Electrical and Electronics Engineers  
ANSI - American National Standards Institute  
NIST - National Institute of Standards and Technology  
EN - European Norm  
BS - British Standard  
DIN - Deutscher Normenausschuss

Business Process Specifications

Business Process  
Execution Language  
for Web Services  
(BPEL4WS)  
1.1  
W3C  
Final Specification

Business Process  
Management  
(BPM)  
1.0  
W3C  
Final Draft

WS-Choreography  
Model Overview  
1.0  
W3C  
Working Draft

Web Service  
Choreography  
Description Language  
(WS-CDL)  
1.0  
W3C  
Working Draft

Web Service  
Choreography  
Interfaces  
(WS-SCI)  
1.0  
W3C  
Final Draft

Management Specifications

Web Services  
Management  
Framework  
(WS-MF)  
1.0  
W3C  
Final Draft

WS-Management  
1.0  
W3C  
Final Draft

WS-Events  
1.0  
W3C  
Working Draft

Web Services  
Management  
(WS-MS)  
1.0  
W3C  
Final Draft

Management  
Using  
Web Services  
(MS-MUW)  
1.0  
W3C  
Final Draft

Metadata Specifications

WS-Policy  
1.0  
W3C  
Final Specification

WS-PolicyAssertions  
1.0  
W3C  
Final Specification

WS-PolicyAttachment  
1.0  
W3C  
Final Specification

WS-Recovery  
1.0  
W3C  
Final Specification

WS-MetadataExchange  
(WSDL-MEX)  
1.0  
W3C  
Final Specification

Universal  
Description,  
Discovery  
and  
Integration  
(UDDI)  
1.0  
W3C  
Final Specification

Web Service  
Description  
Language  
(WSDL)  
1.0  
W3C  
Final Specification

Reliability Specifications

WS-ReliableMessaging  
(WS-RM)  
1.0  
W3C  
Final Specification

WS-Reliability  
1.0  
W3C  
Final Specification

Security Specifications

WS-Security  
1.0  
W3C  
Final Specification

WS-Security  
SOAP  
Message  
Security  
(WS-SM)  
1.0  
W3C  
Final Specification

WS-Security  
Keystore  
Binding  
(WS-Ke)  
1.0  
W3C  
Final Specification

WS-Security  
SAML  
Token  
Profile  
(WS-SSTP)  
1.0  
W3C  
Final Specification

WS-Security, X.509  
Certificates  
Token  
Profile  
(WS-SSTP-CP)  
1.0  
W3C  
Final Specification

WS-Security  
Header  
(WS-SEH)  
1.0  
W3C  
Final Specification

WS-Security  
Policy  
(WS-SP)  
1.0  
W3C  
Final Specification

WS-Trust  
1.0  
W3C  
Final Specification

WS-Federation  
1.0  
W3C  
Final Specification

WS-Security, X.509  
Certificates  
Token  
Profile  
(WS-SSTP-CP)  
1.0  
W3C  
Final Specification

Transaction Specifications

WS-BaseFaults  
(WS-BF)  
1.0  
W3C  
Final Specification

WS-AtomicTransaction  
(WS-AT)  
1.0  
W3C  
Final Specification

WS-Coordination  
(WS-CO)  
1.0  
W3C  
Final Specification

WS-Composite  
Application  
Framework  
(WS-CAF)  
1.0  
W3C  
Final Specification

WS-Context  
(WS-CT)  
1.0  
W3C  
Final Specification

WS-Coordination  
Framework  
(WS-COF)  
1.0  
W3C  
Final Specification

WS-Transaction  
Management  
(WS-TM)  
1.0  
W3C  
Final Specification

Resource Specifications

Web Services  
Resource  
Framework  
(WS-RF)  
1.0  
W3C  
Final Specification

WS-BaseFaults  
(WS-BF)  
1.0  
W3C  
Final Specification

WS-ServiceGroup  
(WS-SG)  
1.0  
W3C  
Final Specification

WS-ResourceProperties  
(WS-RP)  
1.0  
W3C  
Final Specification

WS-ResourceLifecycle  
(WS-RL)  
1.0  
W3C  
Final Specification

WS-Transfer  
(WS-T)  
1.0  
W3C  
Final Specification

Resource  
Representation  
SOAP  
Header  
Block (RRS-HB)  
1.0  
W3C  
Final Specification

Messaging Specifications

WS-Notification  
1.0  
W3C  
Working Draft

WS-Addressing  
1.0  
W3C  
Final Specification

WS-Eventing  
1.0  
W3C  
Final Specification

WS-BaseNotification  
1.0  
W3C  
Final Specification

WS-Topics  
1.0  
W3C  
Working Draft

WS-Eventing  
1.0  
W3C  
Final Specification

WS-Topics  
1.0  
W3C  
Working Draft

WS-Eventing  
1.0  
W3C  
Final Specification

SOAP

SOAP  
1.1  
W3C  
Recommendation

SOAP  
1.2  
W3C  
Draft

SOAP  
Message  
Transport  
Optimization  
Extension  
(MTOM)  
1.0  
W3C  
Final Specification

SOAP  
1.2  
W3C  
Draft

XML Specifications

XML  
1.0  
W3C  
Recommendation

XML  
1.0  
W3C  
Recommendation

Namespaces  
in XML  
1.0  
W3C  
Recommendation

XML Information Set  
1.0  
W3C  
Recommendation

XML Information Set  
1.0  
W3C  
Recommendation

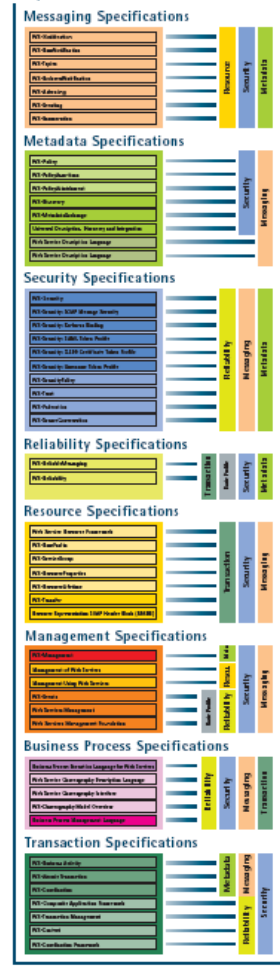
XML Schemas  
1.0  
W3C  
Recommendation

XML Schemas  
1.0  
W3C  
Recommendation

XML Schema  
1.0  
W3C  
Recommendation

XML Schema  
1.0  
W3C  
Recommendation

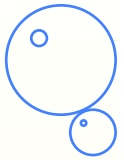
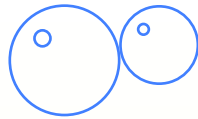
Dependencies



innoQ Deutschland GmbH  
Hallestraße 17  
D-41899 Ratingen  
Telefon +49 (0) 21 02-77 162-300  
Telefax +49 (0) 21 02-77 16-01  
info@innoq.com - www.innoq.com

innoQ Schweiz GmbH  
Gewerbestrasse 11  
CH-4130 Châten  
Telefon +41 (0) 41-743 01 19  
Telefax +41 (0) 41-743 01 19





## Los Orígenes: PreWeb

---

1969

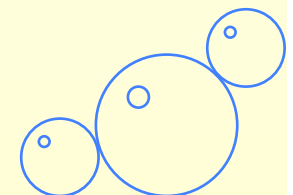
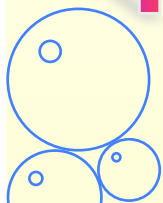
### La Advanced Research Projects Agency (ARPA)

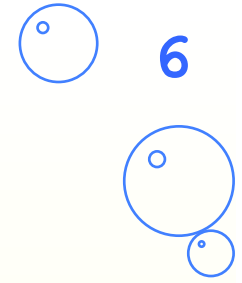
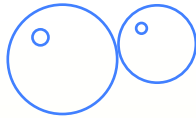
- Conecta computadores de 4 universidades americanas en una pequeña red denominada ARPANET)
- Surgen primeras organizaciones de estandarización para gobernar esta red. Desarrollan estándares como TCP (Transmission Control Protocol). El IP (Internet Protocol). *TCP/IP es la tecnología que define Internet.*

- **Primeros estándares para intercambiar información en Internet (protocolos de aplicación).**

- El protocolo *telnet* y el protocolo de correo electrónico *Simple Mail Transfer Protocol (SMTP)*.
- En 1973 se publica File Transfer Protocol (**FTP**) que soporta la transferencia de ficheros.
  - FTP soportó el primer sistemas de información distribuida tipo Web y **Goopher** supuso el primer protocolo de aplicación que permitía un sencillo sistema cliente servidor y un GUI sencillo.

1973





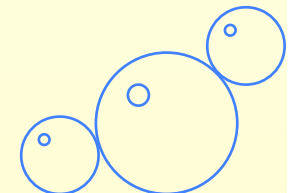
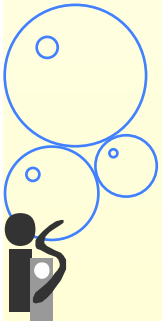
## La Web

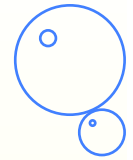
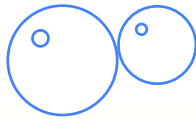
---

- **El núcleo de las tecnologías Web**

- HTTP, HTML, servidores Web y navegadores Web
- HTTP: Hyper Text Transport Protocol
  - Protocolo genérico, sin estado que gobierna la transferencia de ficheros por la red.
  - Desarrollado en el Laboratorio de física de partículas europeo (CERN) por Tim Berners Lee
  - HTTP se desarrolló para soportar hipertextos
    - En particular soporta **HTML** (Hyper Text Markup Language)

1990

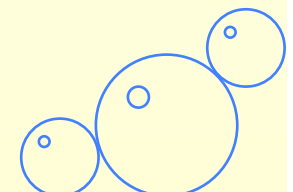
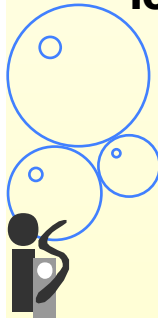
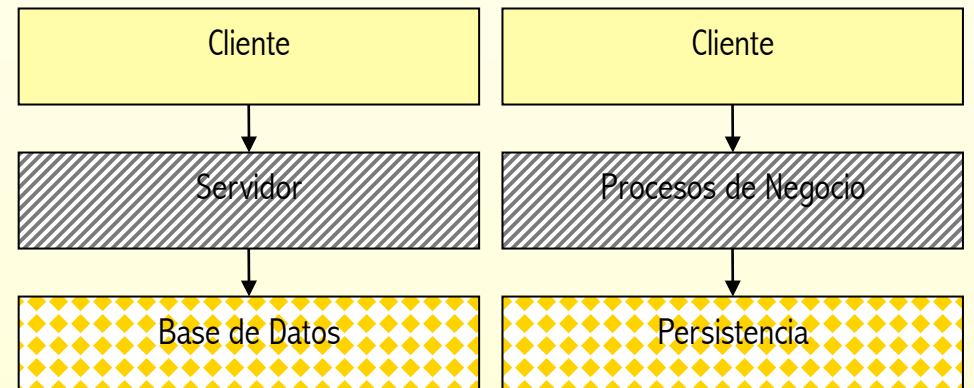
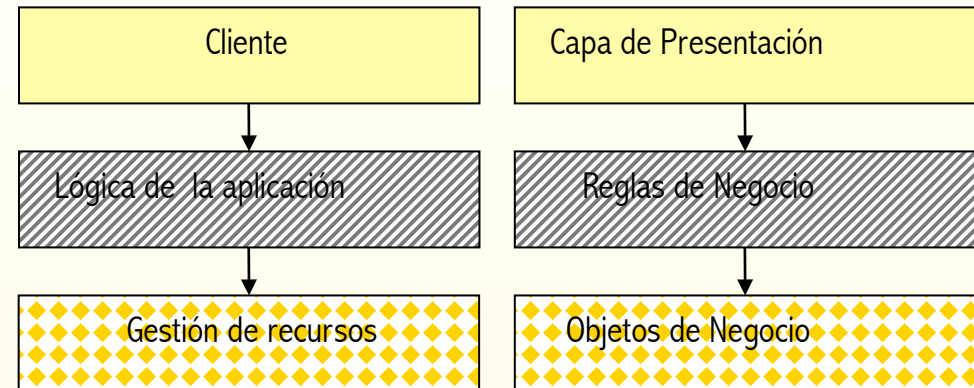


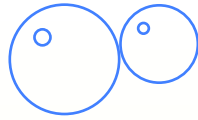


# Capas y Niveles (Layers & tiers)

## Una historia paralela

- **Cliente:** cualquier usuario o programa que quiere realizar una operación sobre el sistema. El cliente a través de la capa de presentación
- La lógica de la aplicación determina lo que hace realmente el sistema.
- La gestión de recursos trata con la organización (almacenamiento, indexación, y recuperación) de los datos necesarios para soportar la lógica de la aplicación.



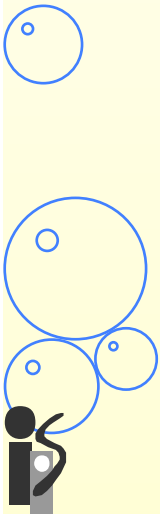
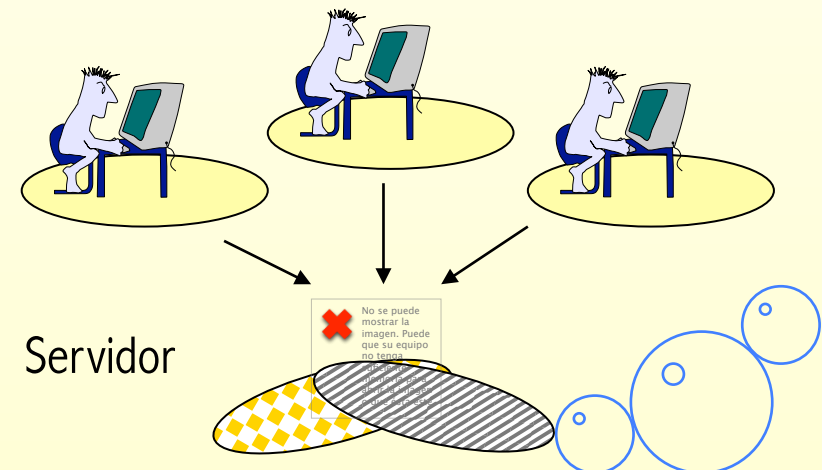


## Primer uso de la “tecnología Web”

---

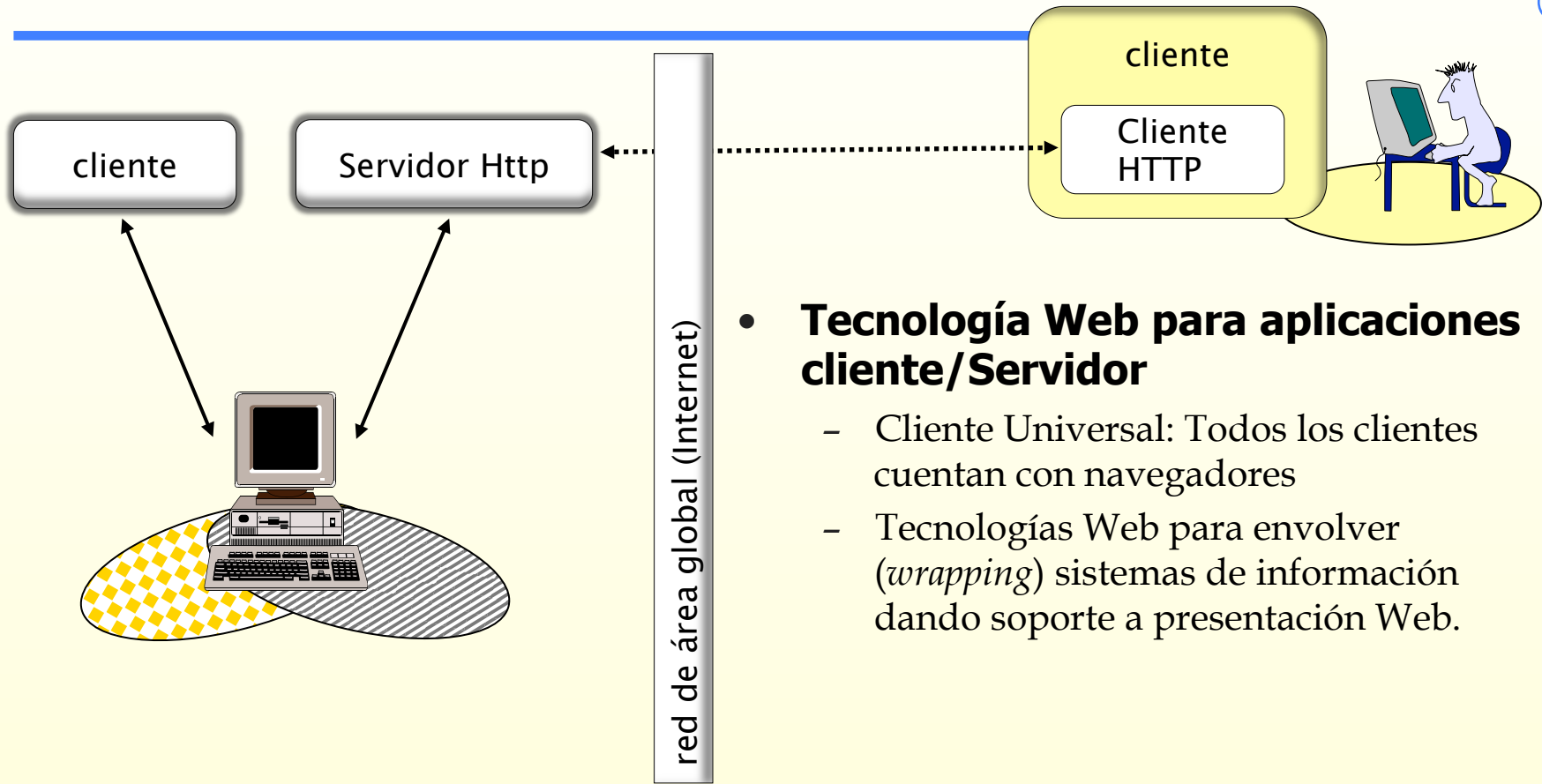
- **Clientes remotos**
  - Rápidamente se vio el valor de mostrar la capa de presentación como documentos HTML
- **Apertura de los sistemas más allá de los límites de una empresa**
  - El ideal de un banco sería instalar un cajero automático en cada hogar.  
*Bussines-to-consumer (B2C)*

Aplicaciones Cliente/Servidor





# Cientes Web

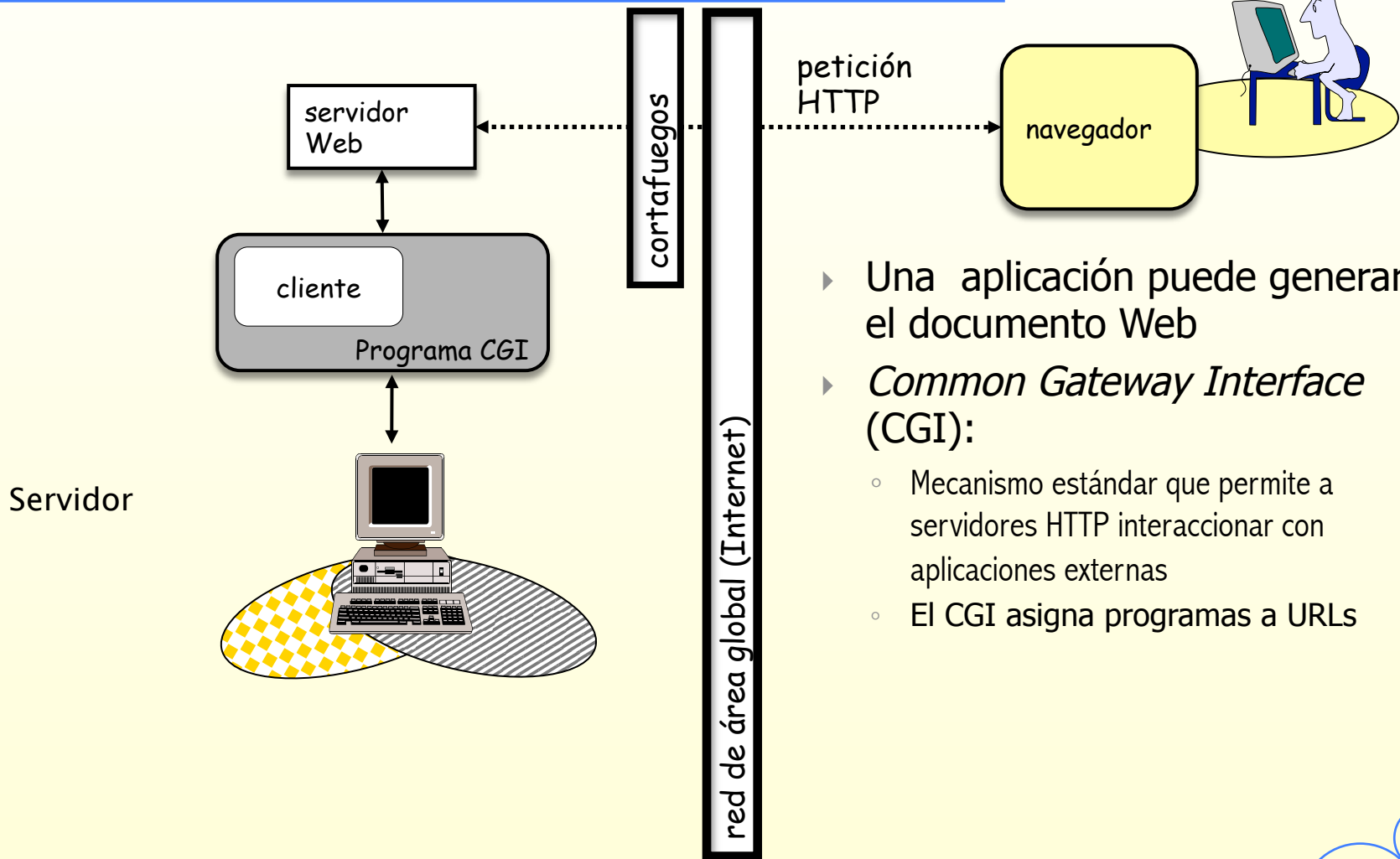


- **Tecnología Web para aplicaciones cliente/Servidor**

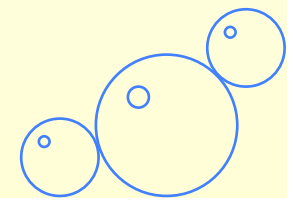
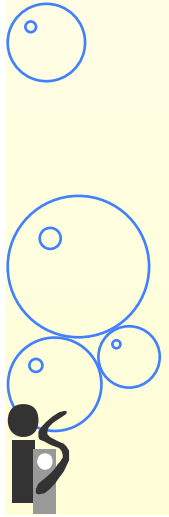
- Cliente Universal: Todos los clientes cuentan con navegadores
- Tecnologías Web para envolver (*wrapping*) sistemas de información dando soporte a presentación Web.

vidor

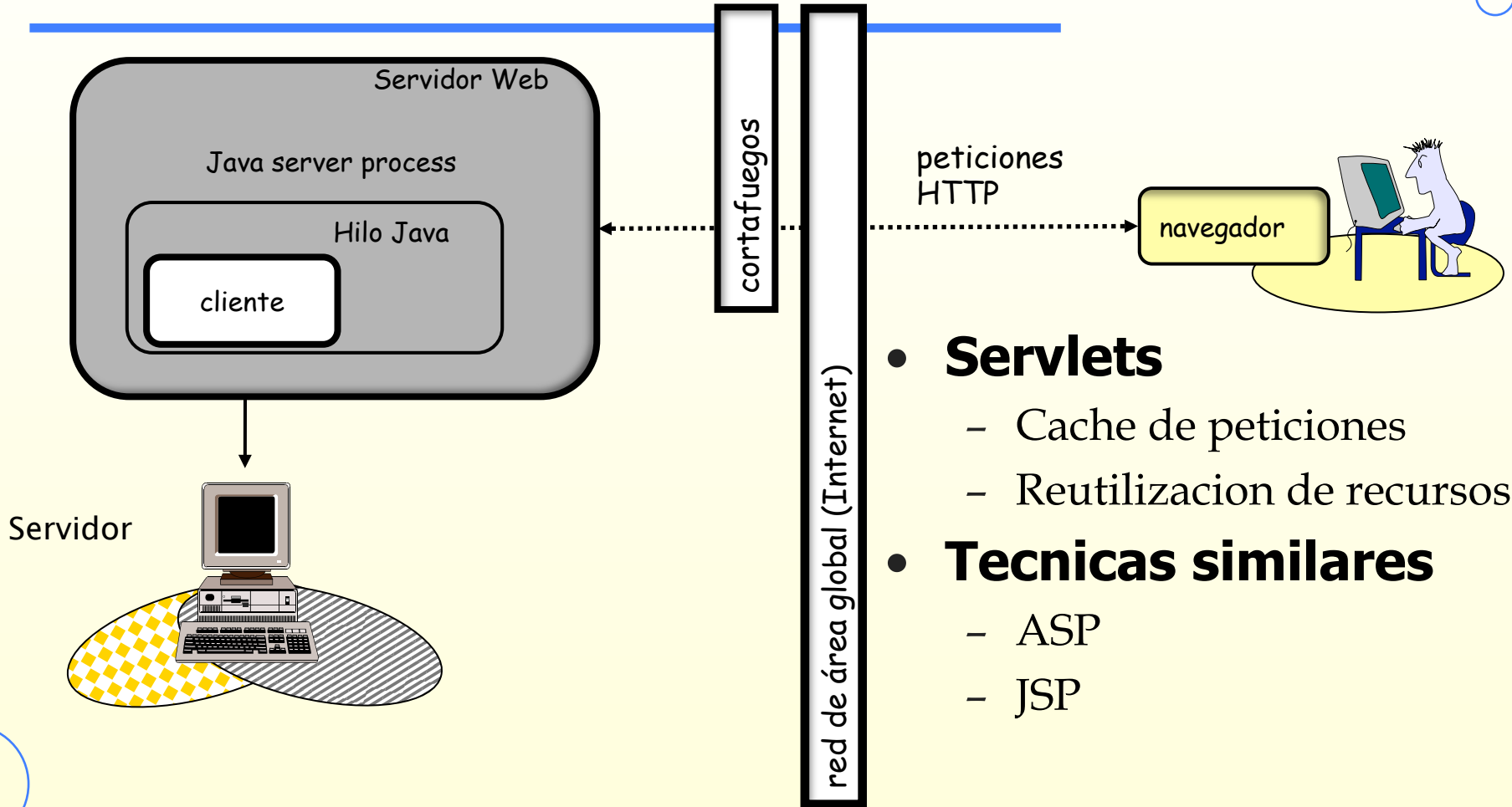
# Programas CGI



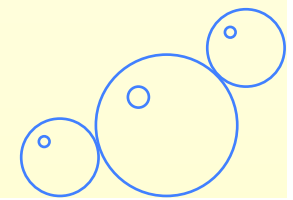
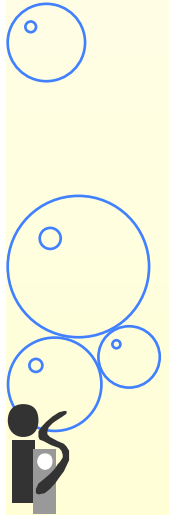
- ▶ Una aplicación puede generar el documento Web
- ▶ *Common Gateway Interface (CGI)*:
  - Mecanismo estándar que permite a servidores HTTP interactuar con aplicaciones externas
  - El CGI asigna programas a URLs



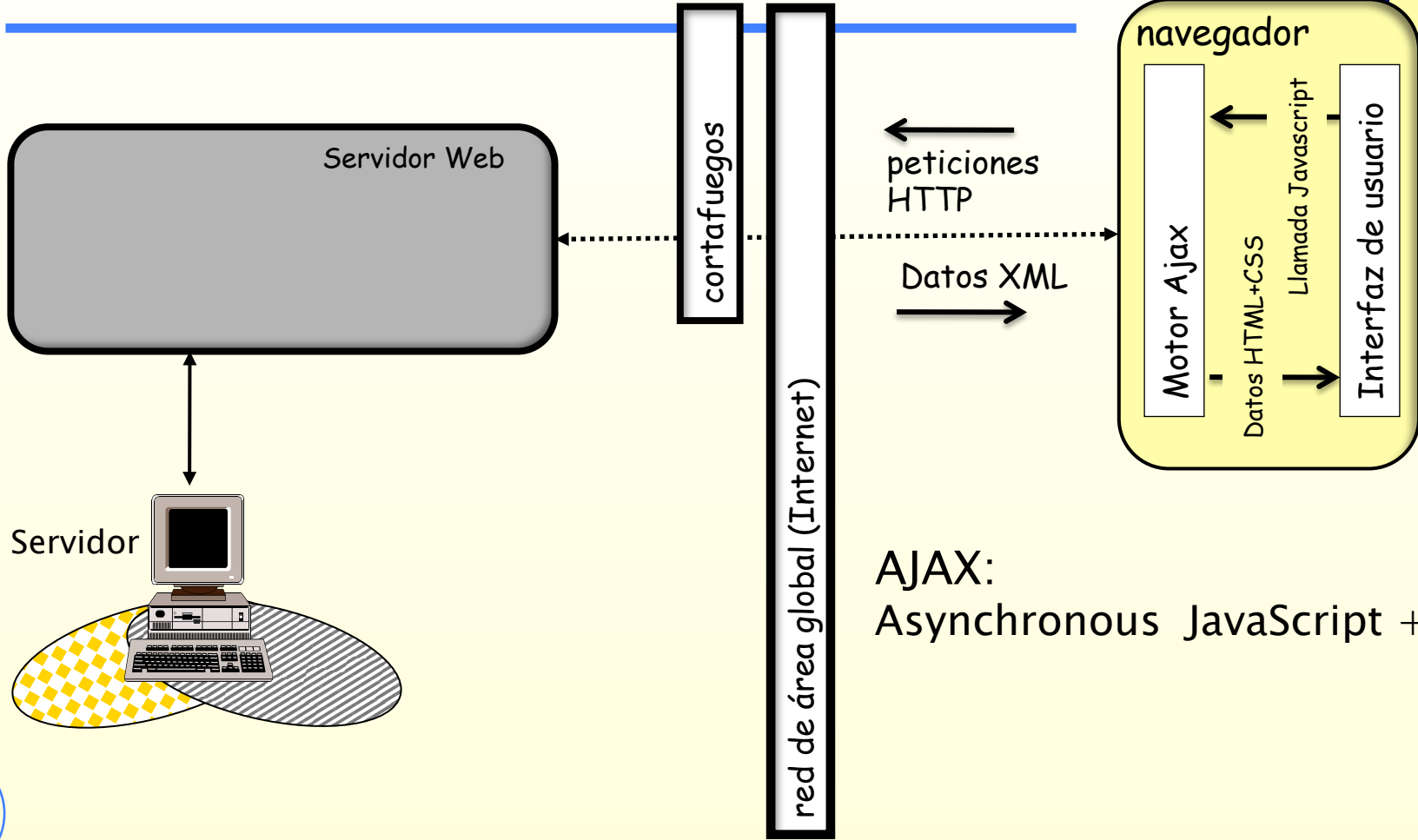
# Servlets



- **Servlets**
  - Cache de peticiones
  - Reutilizacion de recursos
- **Tecnicas similares**
  - ASP
  - JSP

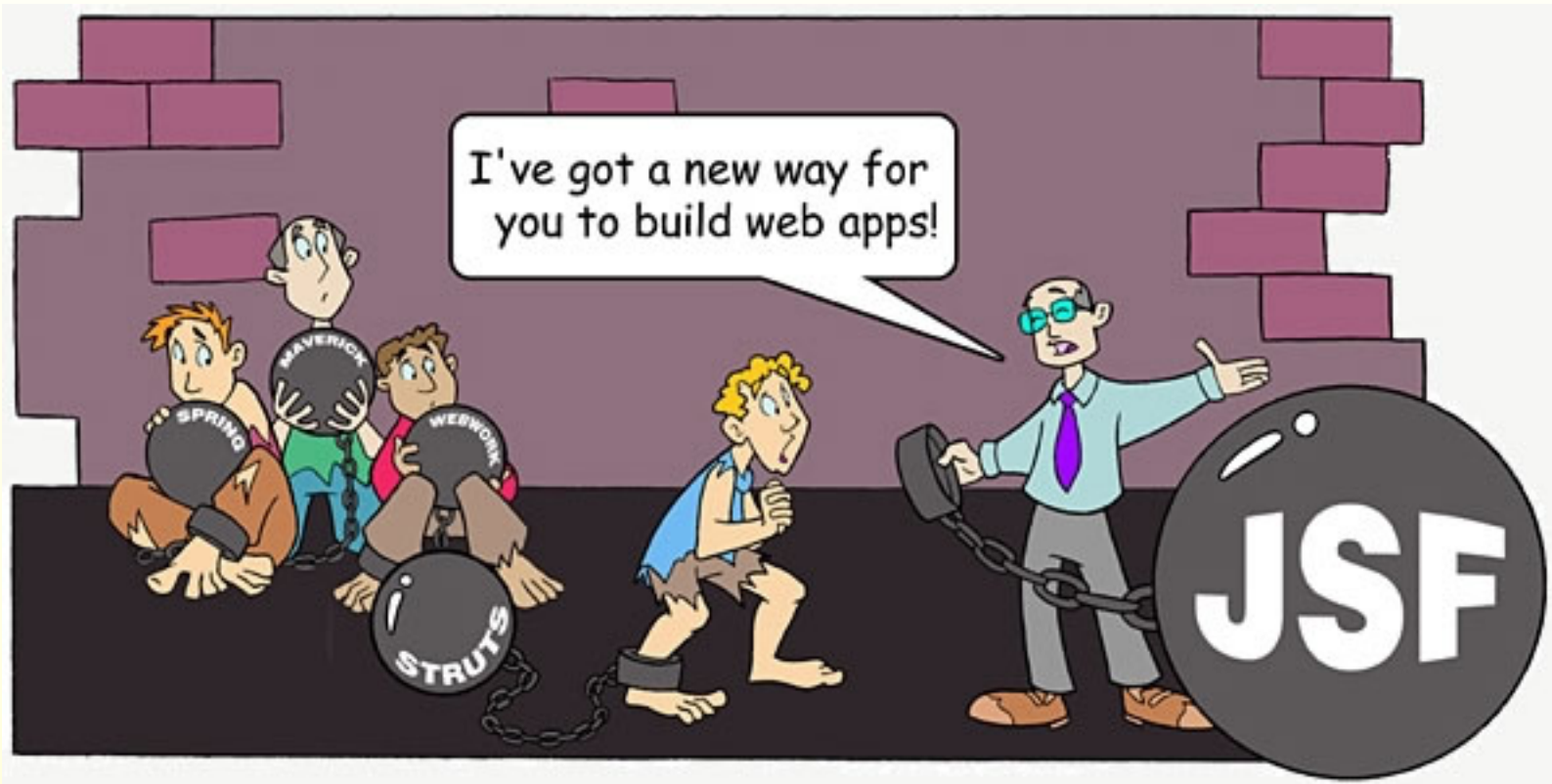


# Y más...



AJAX:  
Asynchronous JavaScript + XML

## Tecnologías para la capa de presentación

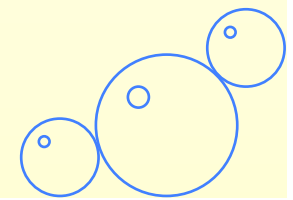
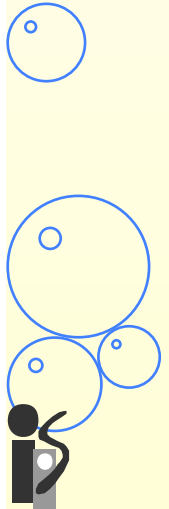
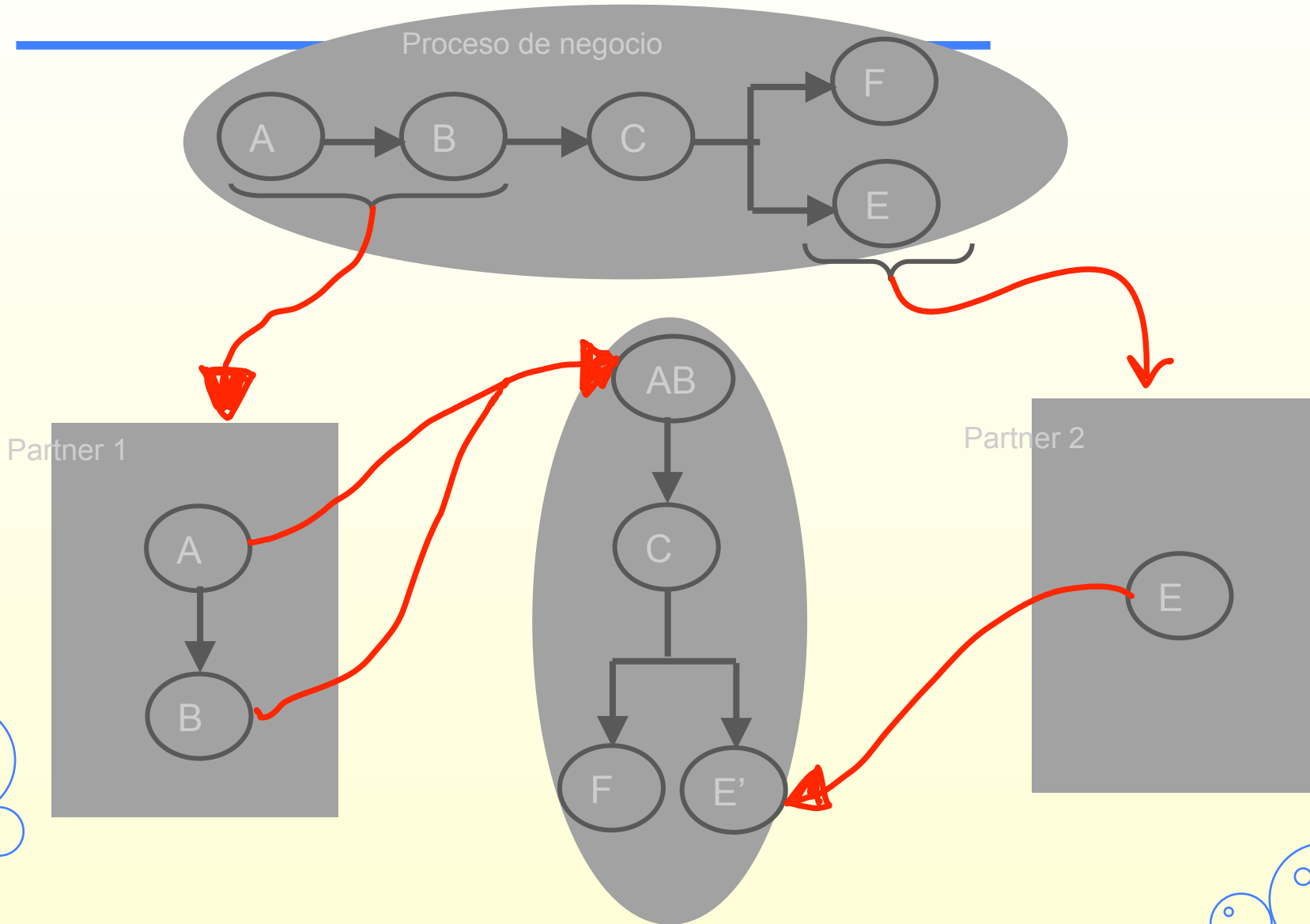


theserverside.com

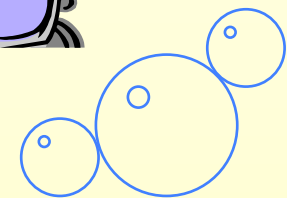
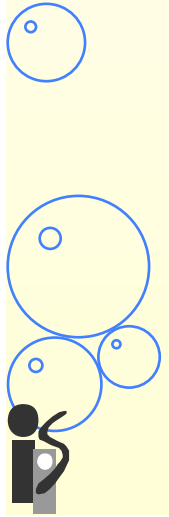
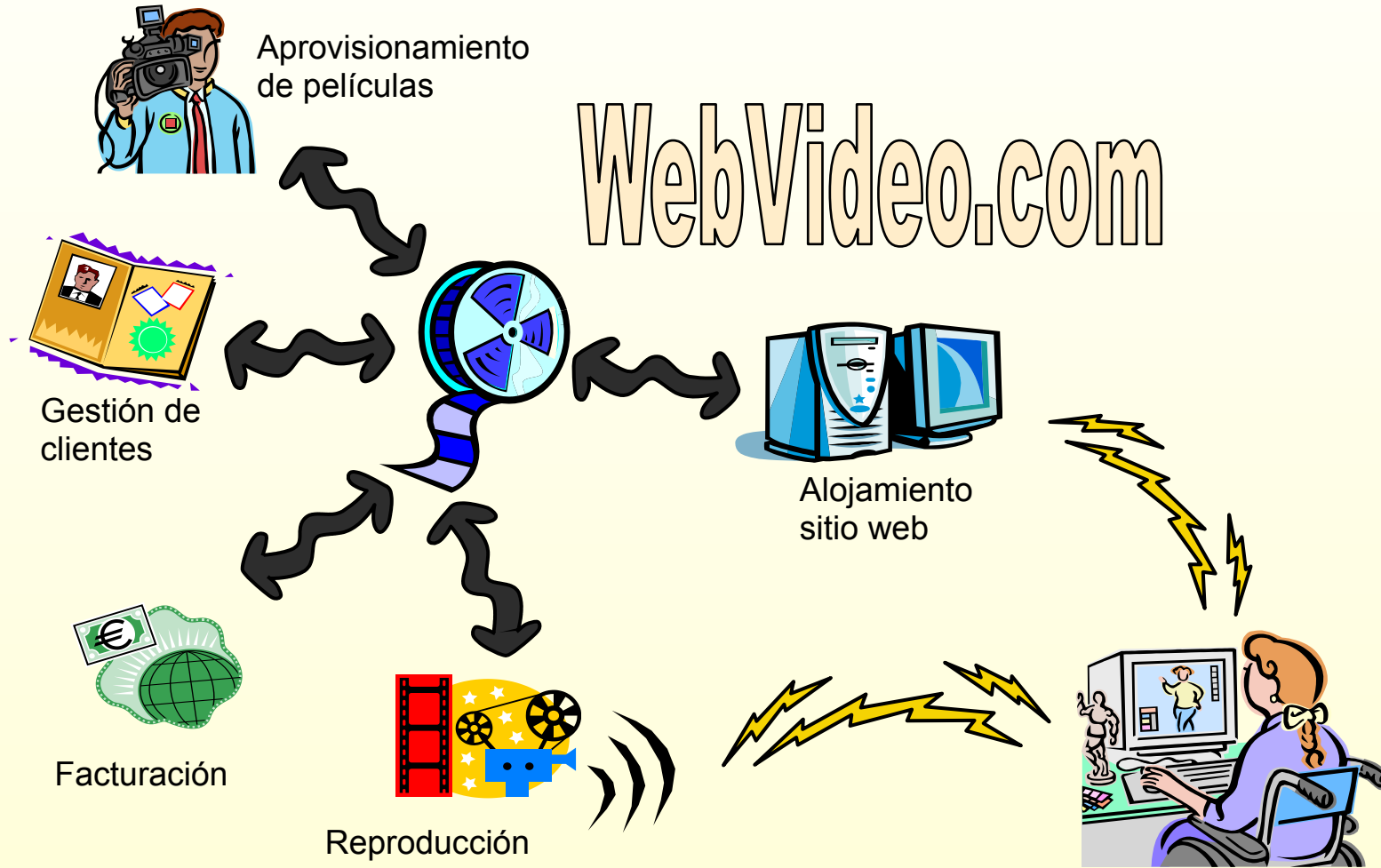
<http://www.theserverside.com/cartoons/TalesFromTheServerSide.tss>



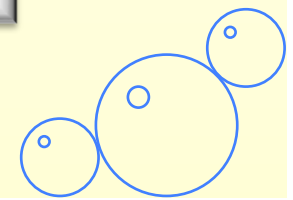
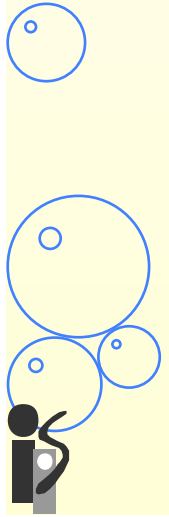
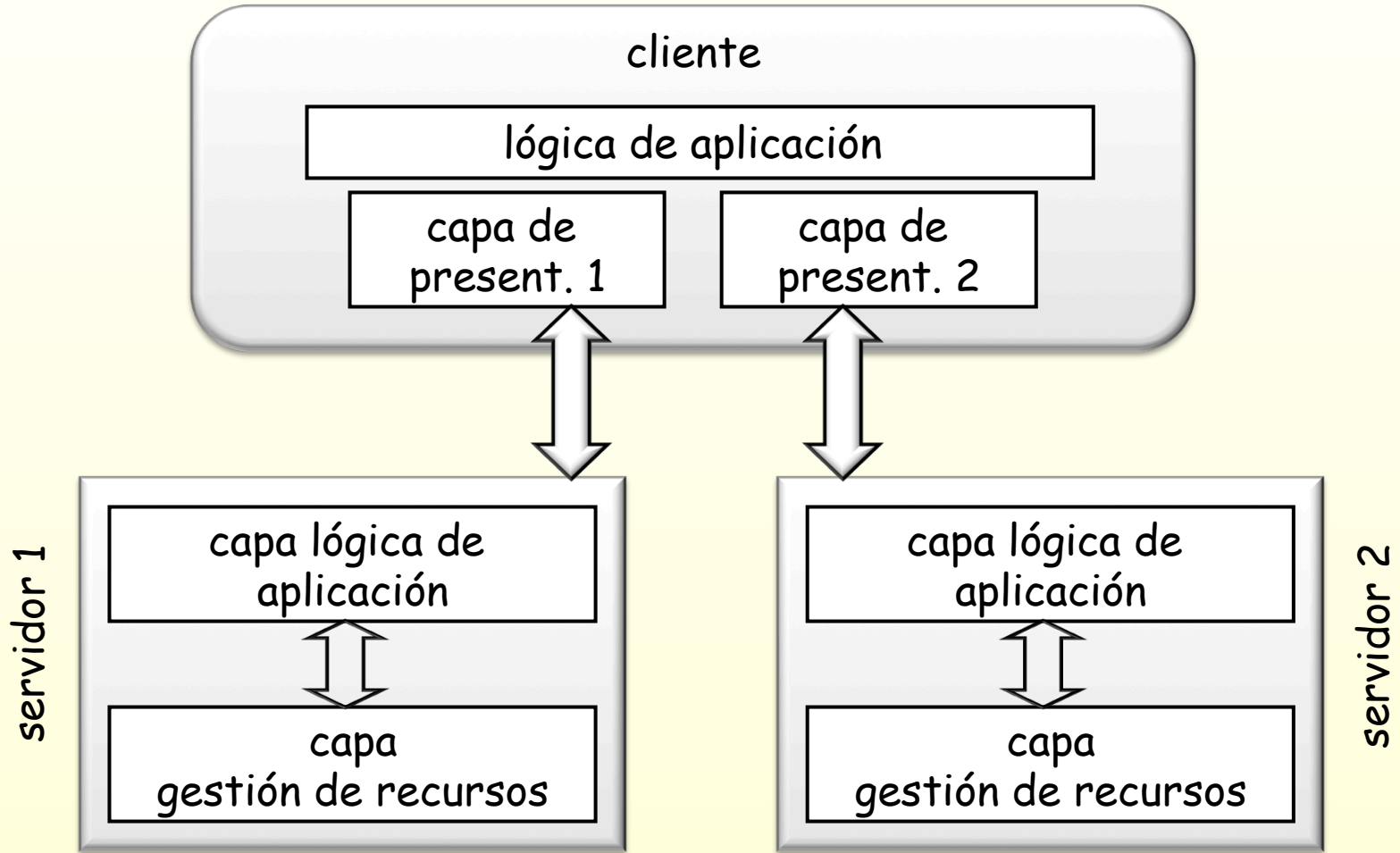
# Procesos de negocio: Del B2C al B2B



# Un vídeo club en tiempo real

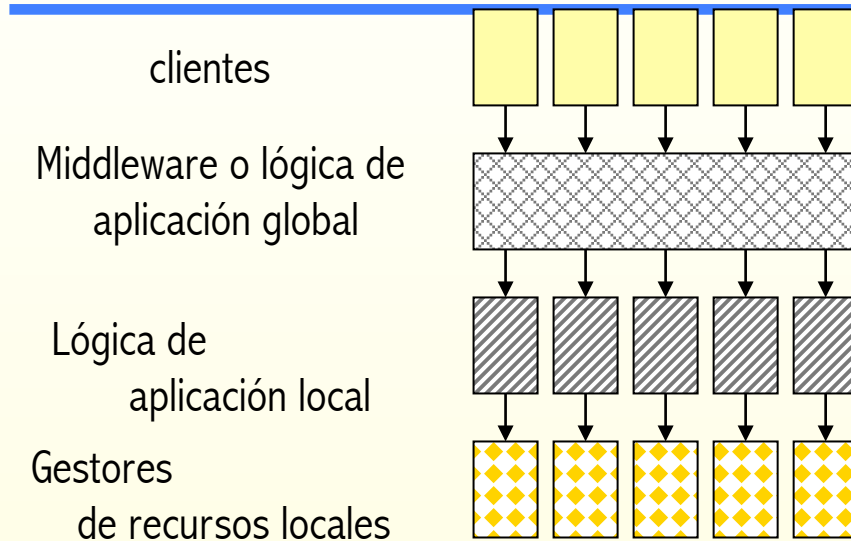


# Integrando capas

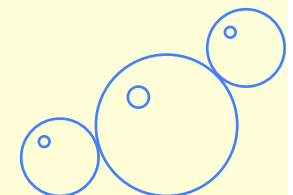
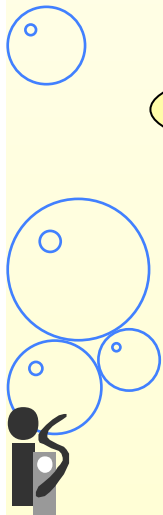
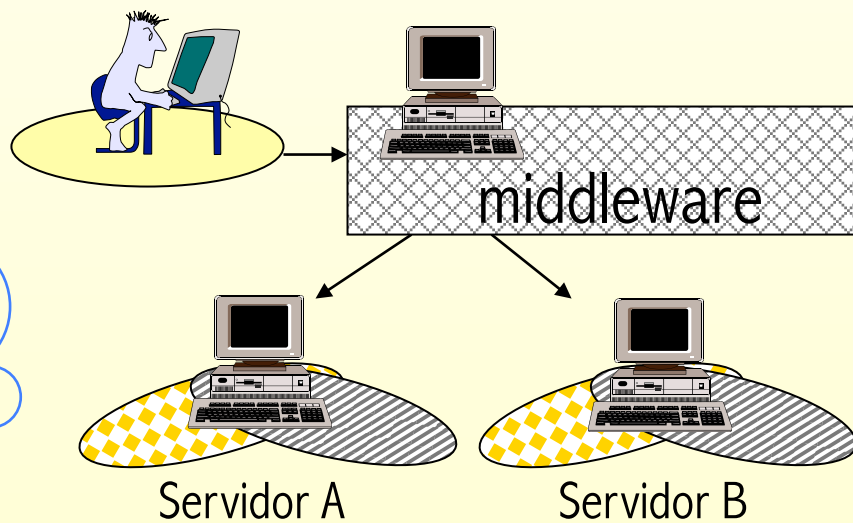


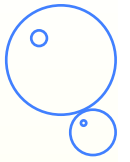
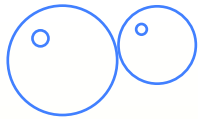


# Middleware

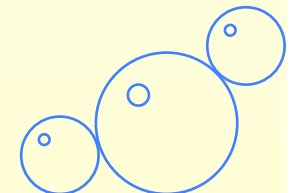
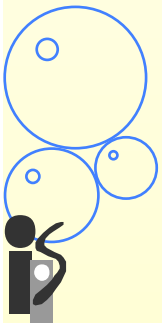
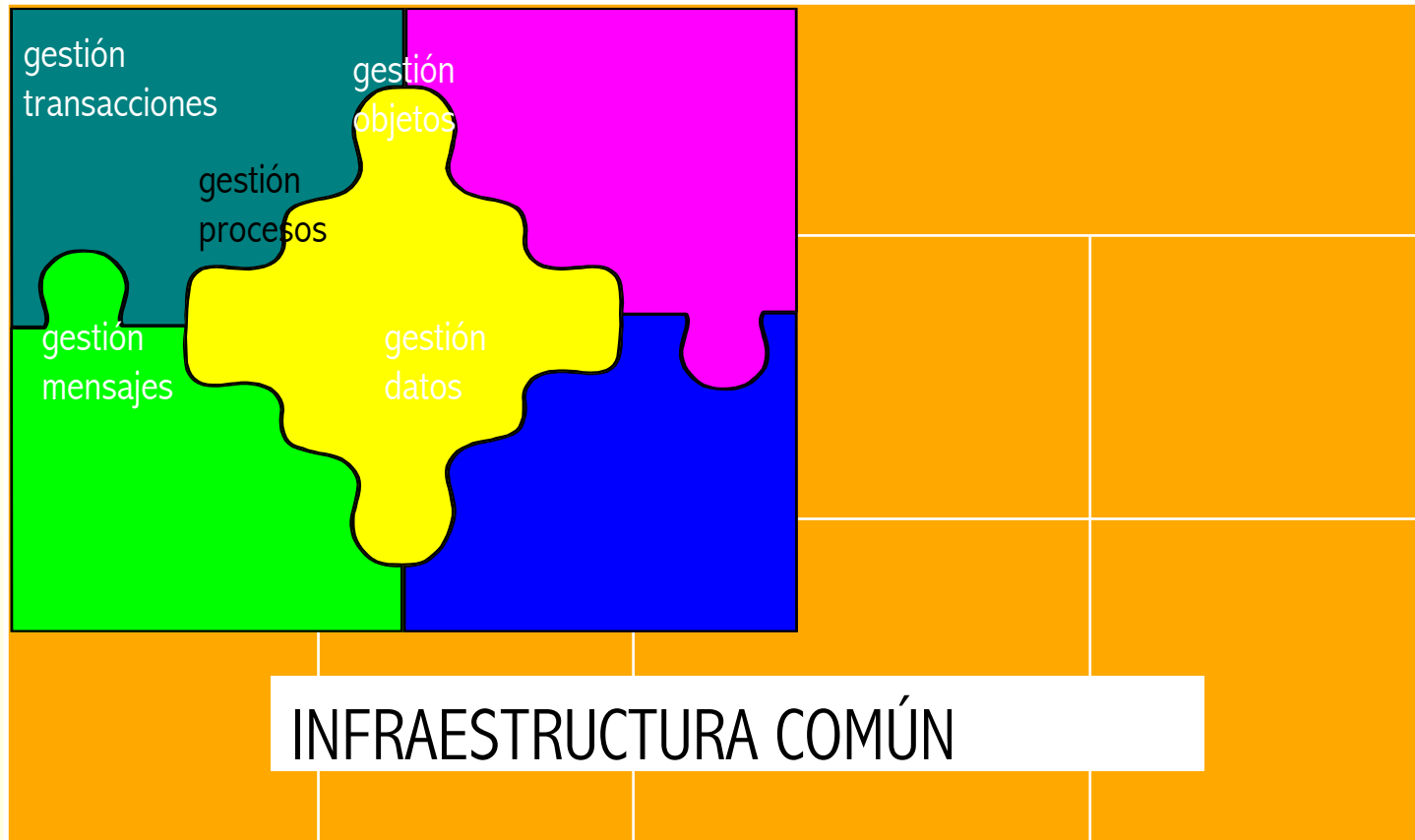


- **Un Middleware es simplemente un nivel de indirección entre los clientes y el resto de capas del sistema.**
- **El middleware es infraestructura que:**
  - Simplifica el diseño de los clientes reduciendo el número de interfaces,
  - Se ofrece acceso transparente a los sistemas subyacentes,
  - El middleware actúa como la plataforma para integrar la funcionalidad de los sistemas y soporta la lógica de aplicación de alto nivel, y
  - Se preocupa de localizar los recursos, acceder a estos, y recoger los resultados.





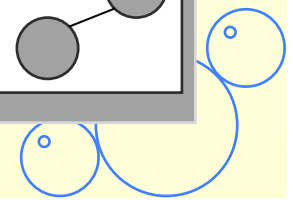
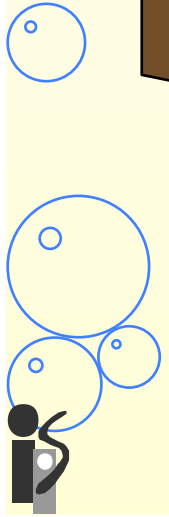
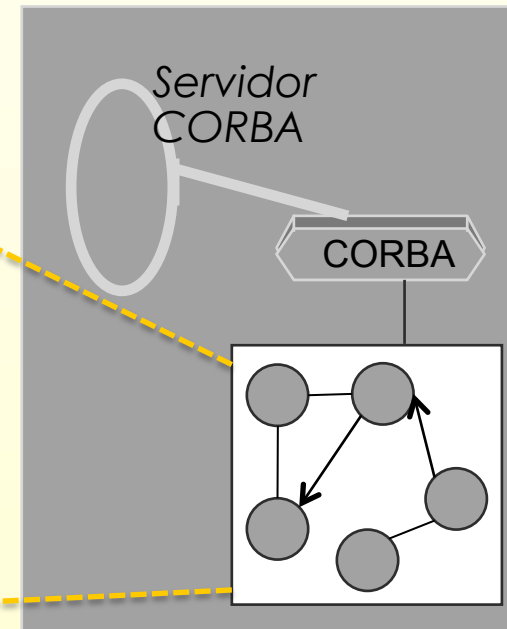
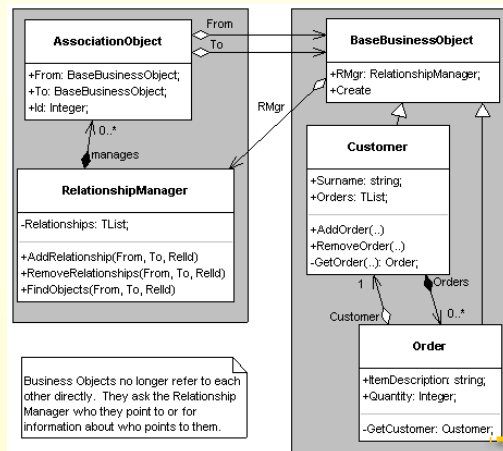
# Middleware “Ideal” para definir procesos de negocio

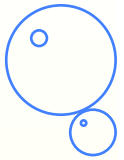
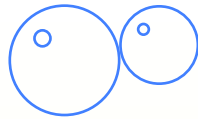


# Problemas del Middleware tradicional

- **CORBA/Middleware tradicional**

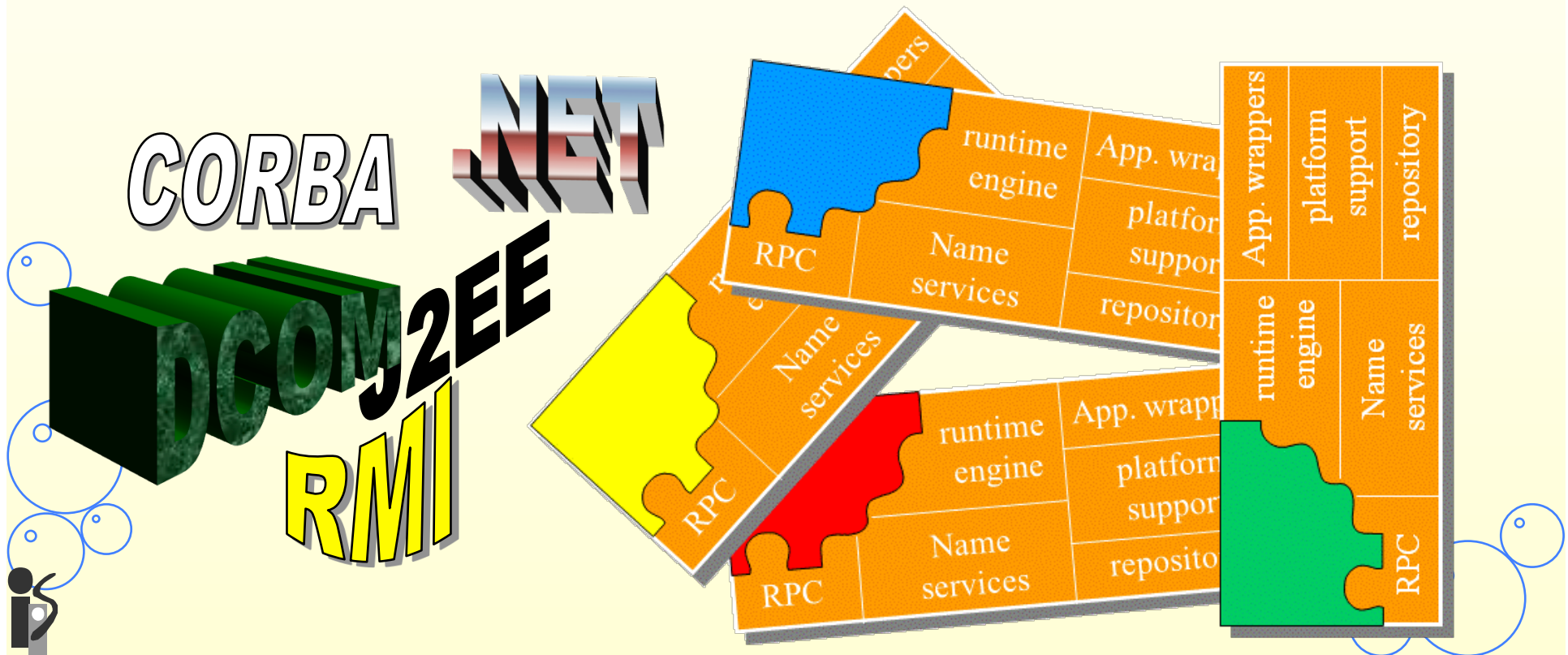
- El programador debe conocer el modelo computacional y el diseño utilizado por los proveedores de servicios
- Uso para red de área local

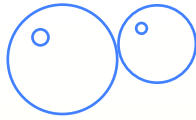




## Middleware incompatibles

- **En interacciones entre organizaciones no hay un lugar obvio donde colocar el middleware**
  - La adopción de la misma solución supone que todos acuerdan utilizar una determinada plataforma middleware.

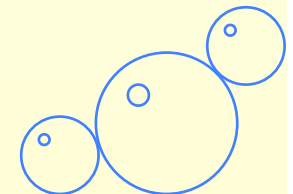
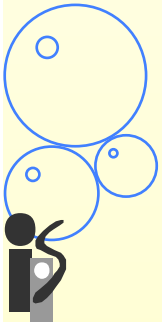


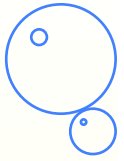
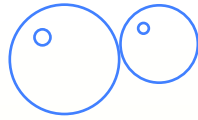


---

**Infraestructura actual para el comercio electrónico**  
**SOC, Web Services, SOA**  
**REST**  
**Web 2.0**

# SERVICIOS WEB

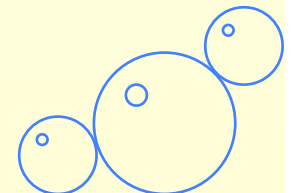
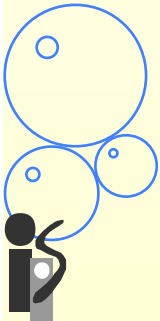




## Requisitos para el e-commerce

---

- **Abstracciones de programación**
  - Funcionalidad adicional introduciendo nuevas capas
  - Middleware, que abstrae de aspectos de distribución, prestaciones, ciclo de vida y la gestión de recursos, la persistencia, etc.



## Requisitos para el e-commerce

---

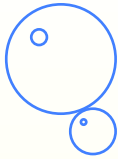
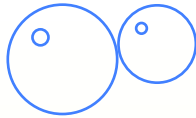
- **Infraestructura de soporte a las abstracciones**

*En el desarrollo de software, un **framework** es **una estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado**. Típicamente, un framework puede incluir soporte de programas, librerías y un lenguaje de scripting entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto.*

*Un framework **representa una arquitectura de software** que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.*

(WIKIPEDIA <http://es.wikipedia.org/wiki/Framework>)

*framework = **infraestructura***



## Web Services Framework

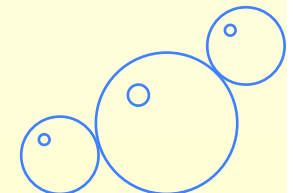
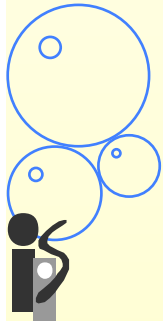
---

***Today, the principal use of the World Wide Web is for interactive access to documents and applications. In almost all cases, such access is by human users, typically working through Web browsers, audio players, or other interactive front-end systems. The Web can grow significantly in power and scope if it is extended to support communication between applications, from one program to another.”***

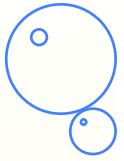
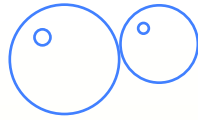
***IBM “Web Services Framework” for W3C Workshop on Web Services 11-12 April 2001***

<http://www.w3.org/2001/03/WSWS-popa/paper51>

*Web Service framework = soporta la comunicación  
entre aplicaciones*







## ¿Qué tipo de Framework....?

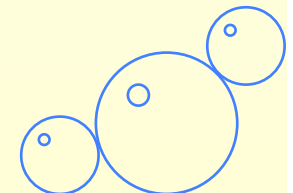
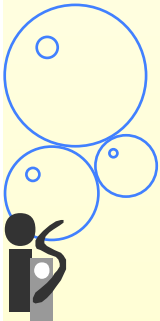
... our primary goal is to **connect applications** on a worldwide basis. Such applications will necessarily be **built in a variety of programming languages, using a range of operating systems, database, and middleware technologies.** **The interoperability we seek can only be achieved when based on standard data formats and protocols, not APIs.**

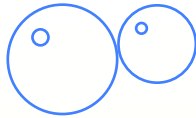


NO ES UNA INFRAESTRUCTURA CON APIs definidas!!!



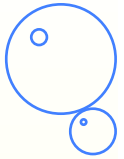
SE BASA EN FORMATOS DE DATOS Y PROTOCOLOS  
ESTÁNDAR = INFRAESTRUCTURA MINIMA DE COMUNICACIÓN





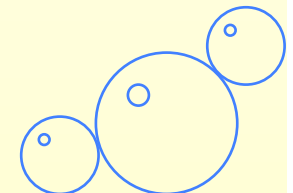
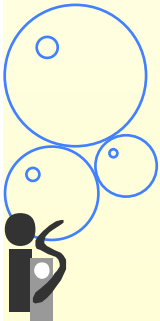
## SOC:Comunicación en Servicios

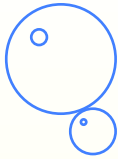
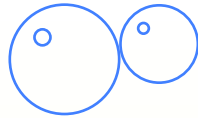
---



*Services are defined as exchange of messages between participants. This separation of participants in a exchange is a key to decoupling applications. **Service-oriented systems hide the internal abstractions that provides the service such as classes, objects, methods, or remote procedures.** By avoiding any knowledge of the internal structure, it is possible to incorporate any software component or application that can be "wrapped" in message handling code that allows it to adhere to the formal service definition*

**Web Services Architecture**  
**W3C Working Group Note 11 February 2004**  
<http://www.w3.org/TR/ws-arch/wsa.pdf>

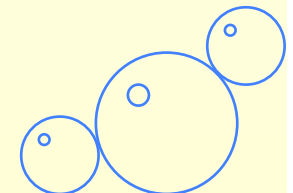
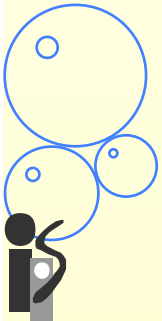


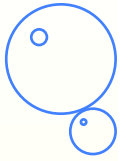
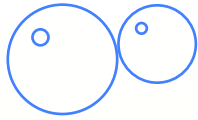


## Middleware como infraestructura

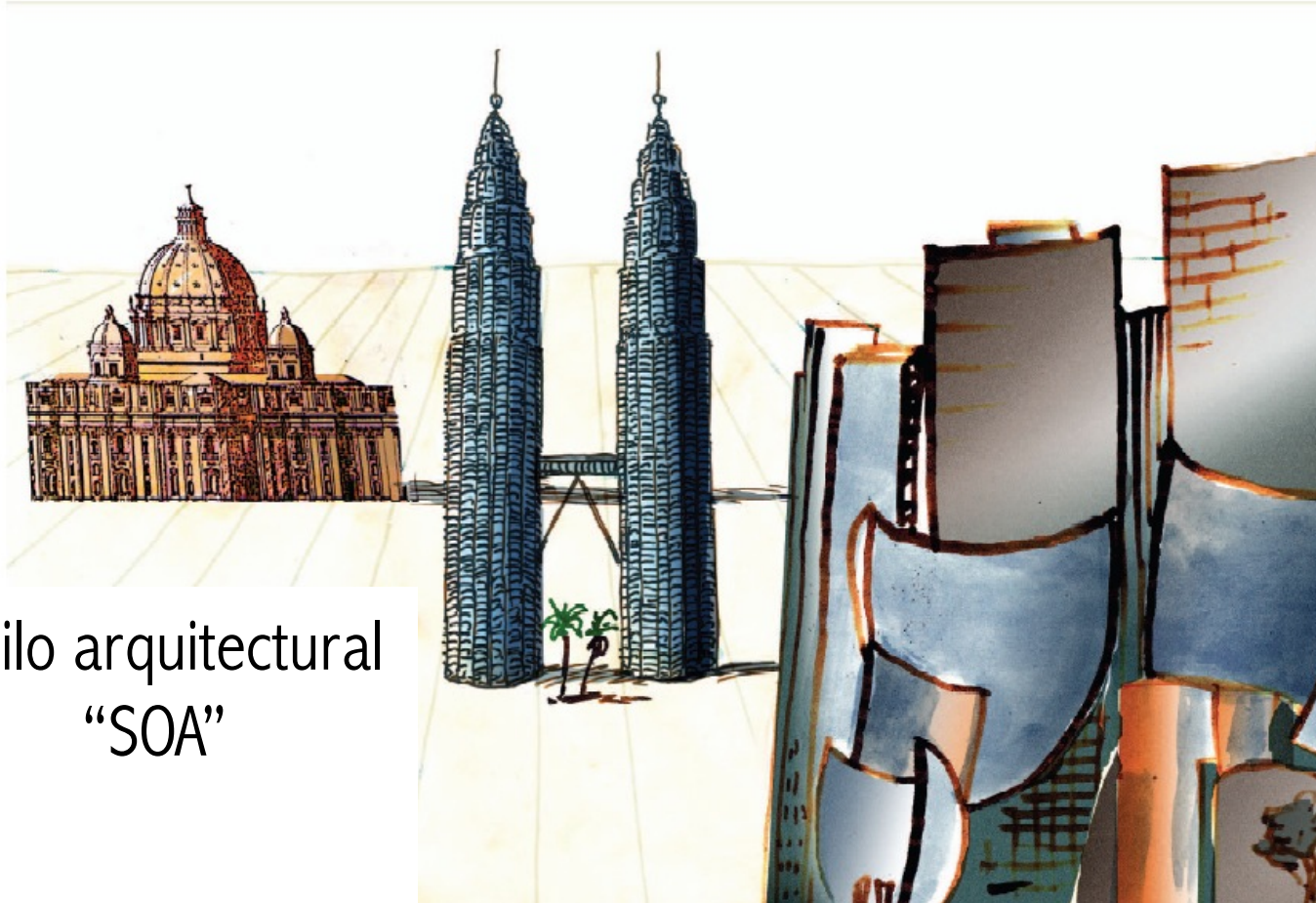
---

- **Java (EJB, RMI, CORBA, etc.), .NET, son infraestructuras middleware. Capa software ejecutable que me permite abstraernos de aspectos cotidianos en la programación de sistemas distribuidos**
  - Primitivas de comunicación basada en RPC, RMI, ...
  - Soporte a transacciones
  - Gestión del ciclo de vida de los objetos/Procesos
  - Nos facilitan la definición de la lógica de negocio
  - ...
- **¡Son plataformas ejecutables con un modelo de programación concreto!**

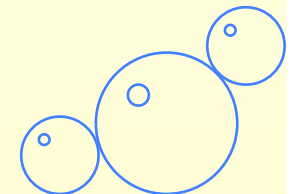
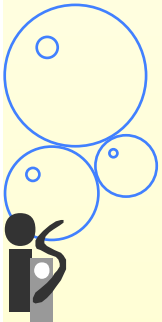


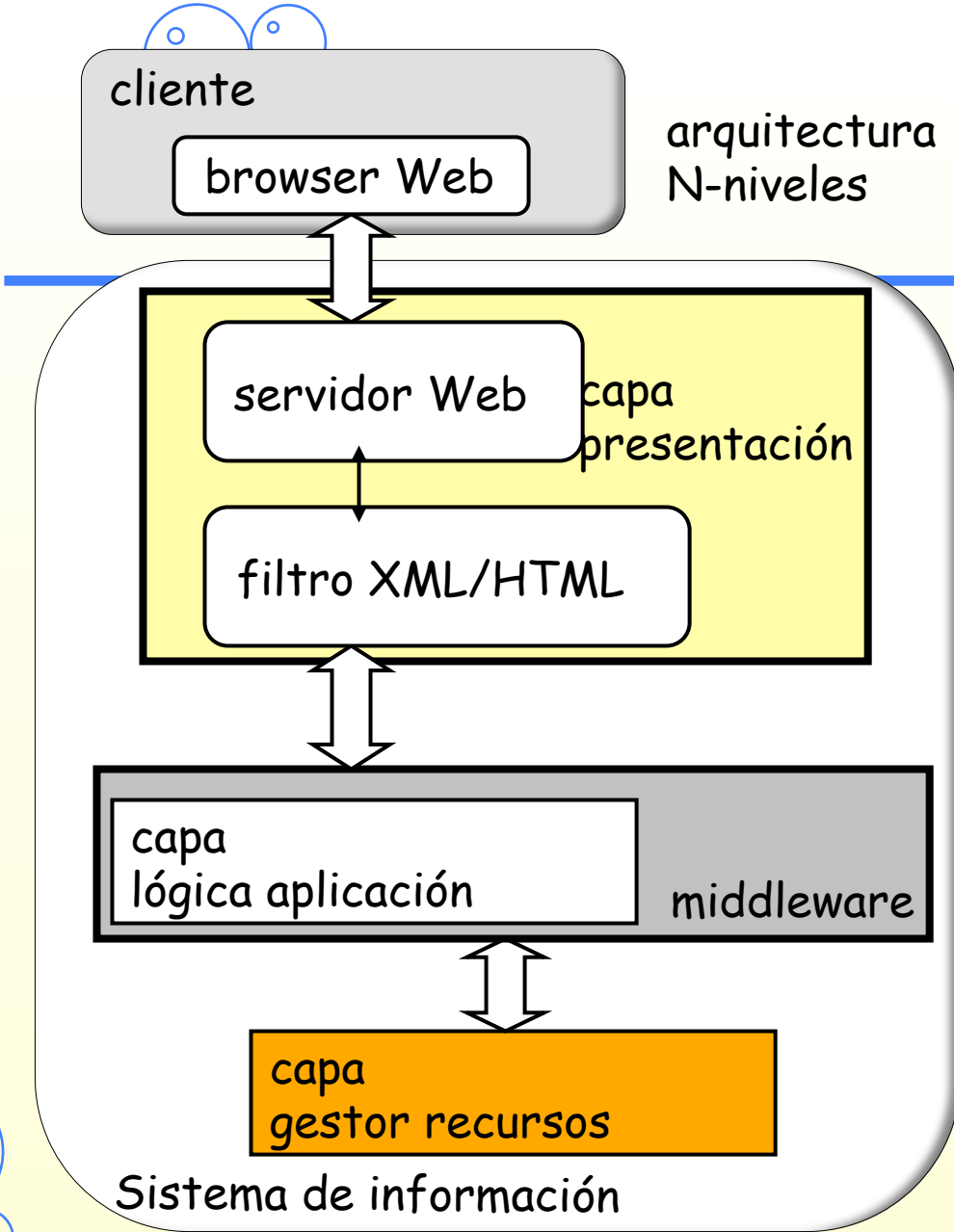


## Arquitectura Orientada a Servicios

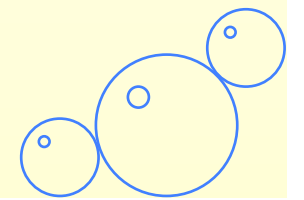
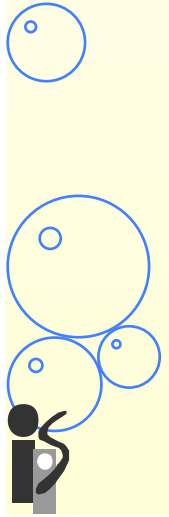


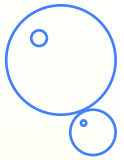
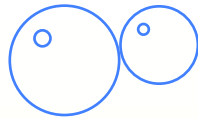
Estilo arquitectural  
"SOA"





## Servidores de Aplicaciones: Middleware con capa presentación Web



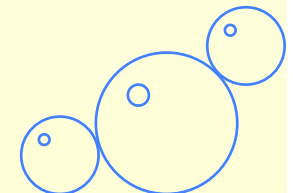


## Arquitectura Orientada a Sevicios (SOA)

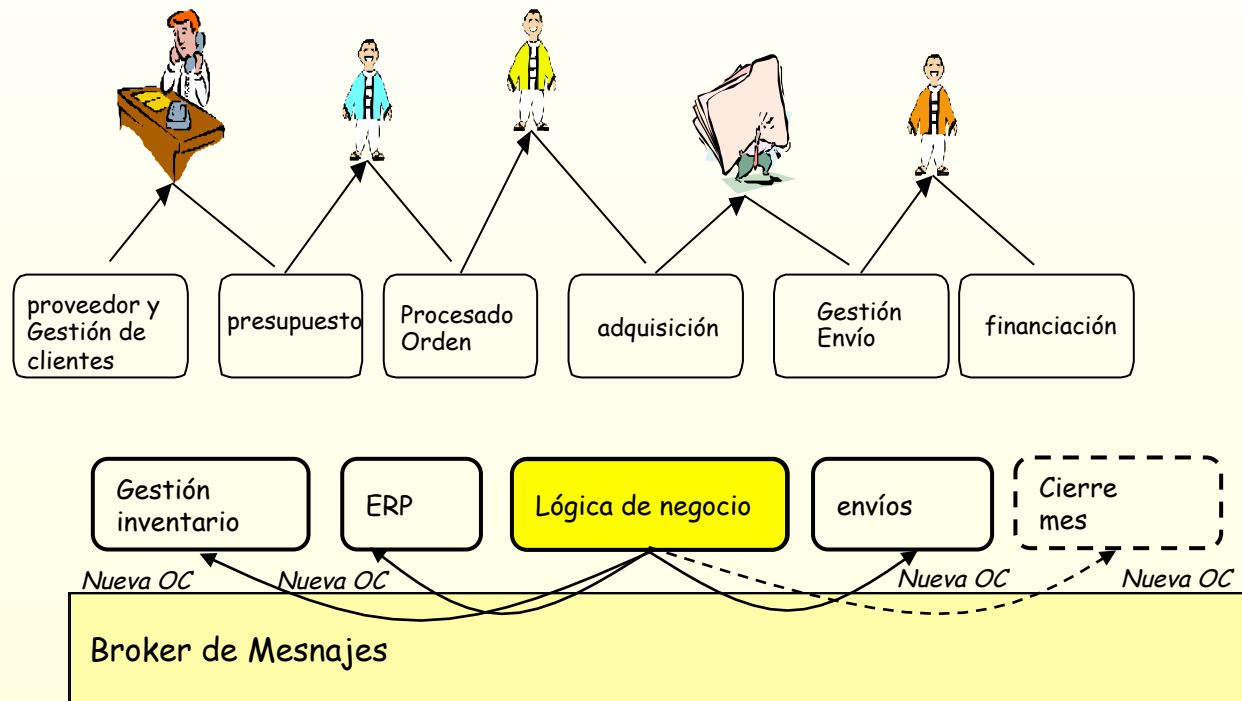
---

"A service oriented architecture is an **approach to design and integrate** software in a modular method where each module is precisely a **'loosely coupled service'** that is **accessible over a network** and has the capability of being **dynamically integrated** with other services at run time. A service must present a **standard Interface** (be it WSDL today) for its functionality and invocation methods while **the real implementation of the service is not a concern of an SOA.**"

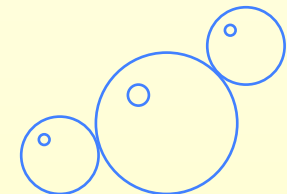
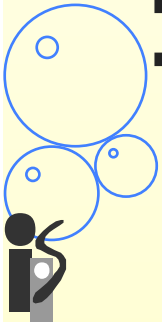
*Rajesh Dawar*



# Desacoplo entre aplicaciones

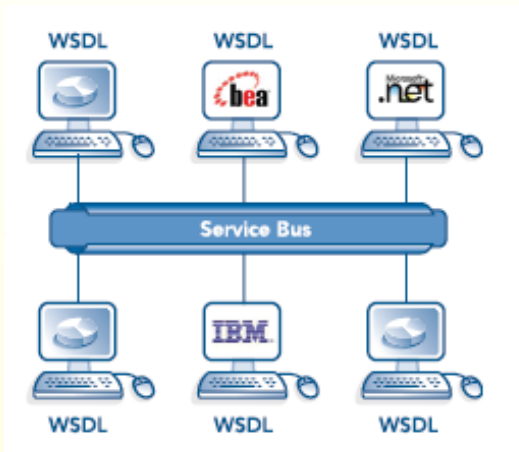


- **Flexibilidad a la hora de dirigir los mensajes**
- **Comunicación Asíncrona**
- **Las aplicaciones pueden interaccionar con nuevos sistemas sin cambios**

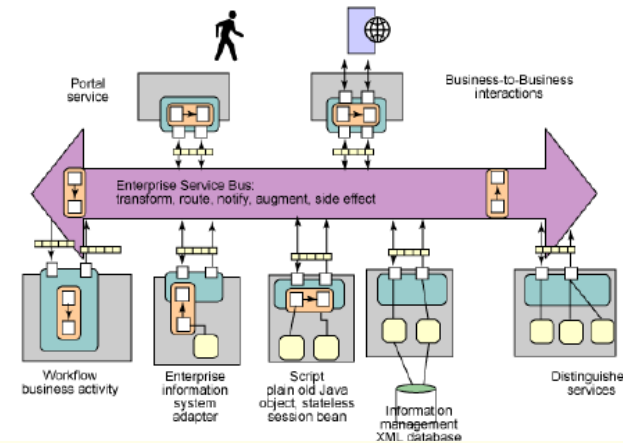


# Productos/Especificaciones SOA

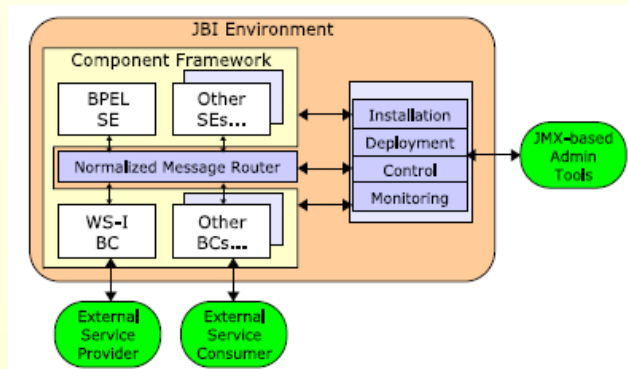
## TIBCO: Enterprise Server Bus



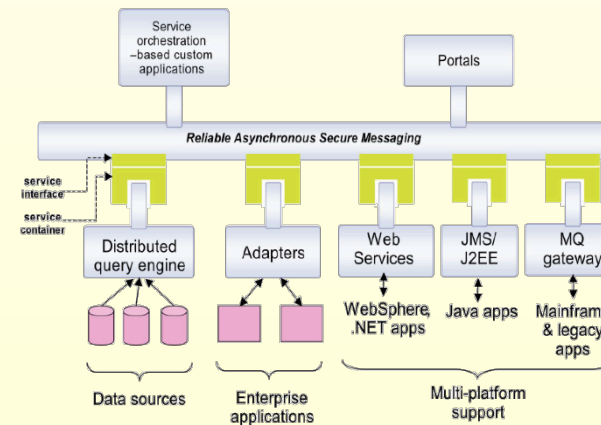
## IBM: Enterprise Server Bus



## SUN: Java™ Business Integration (JBI) 1.0



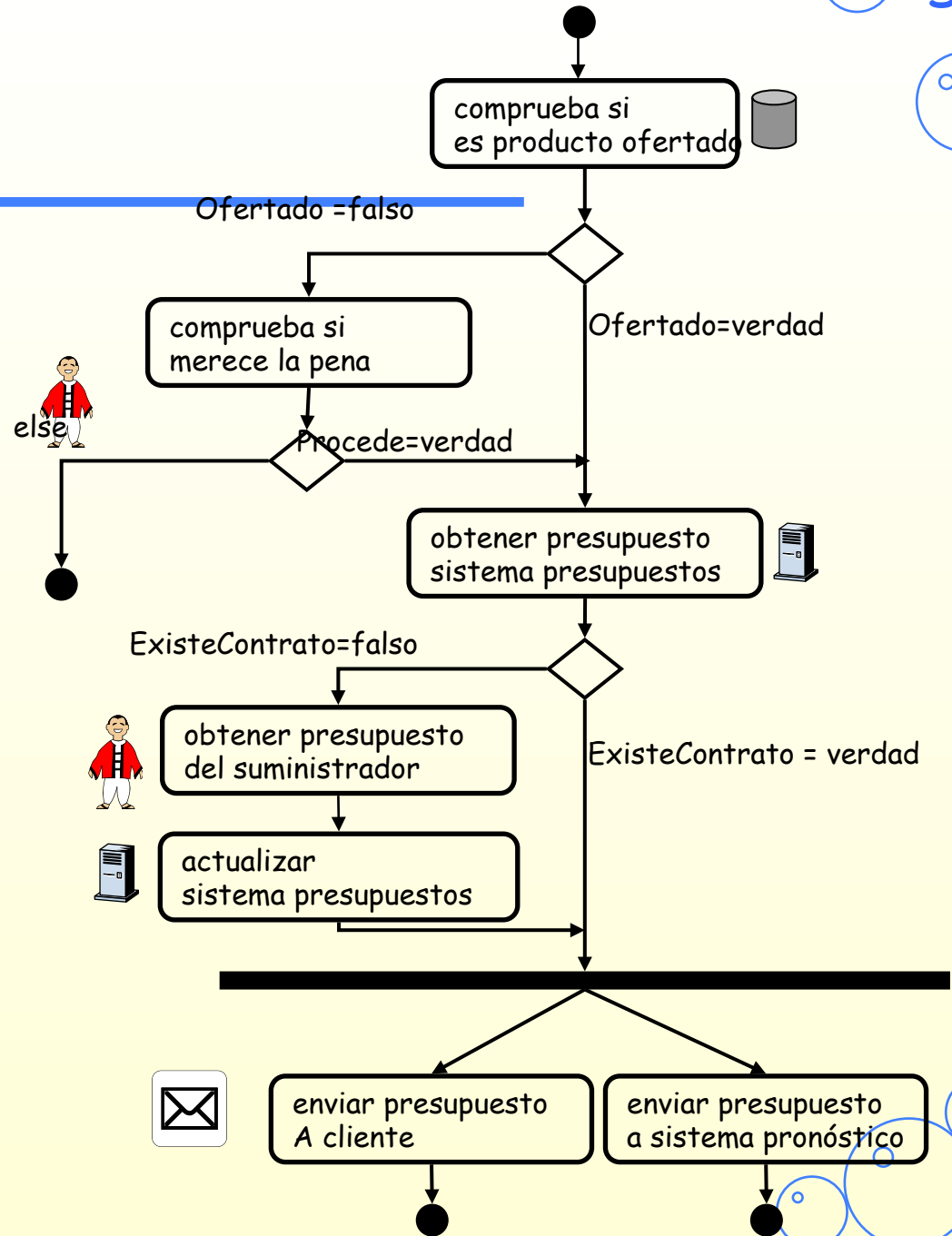
## Service-Oriented Computing: A Research Roadmap



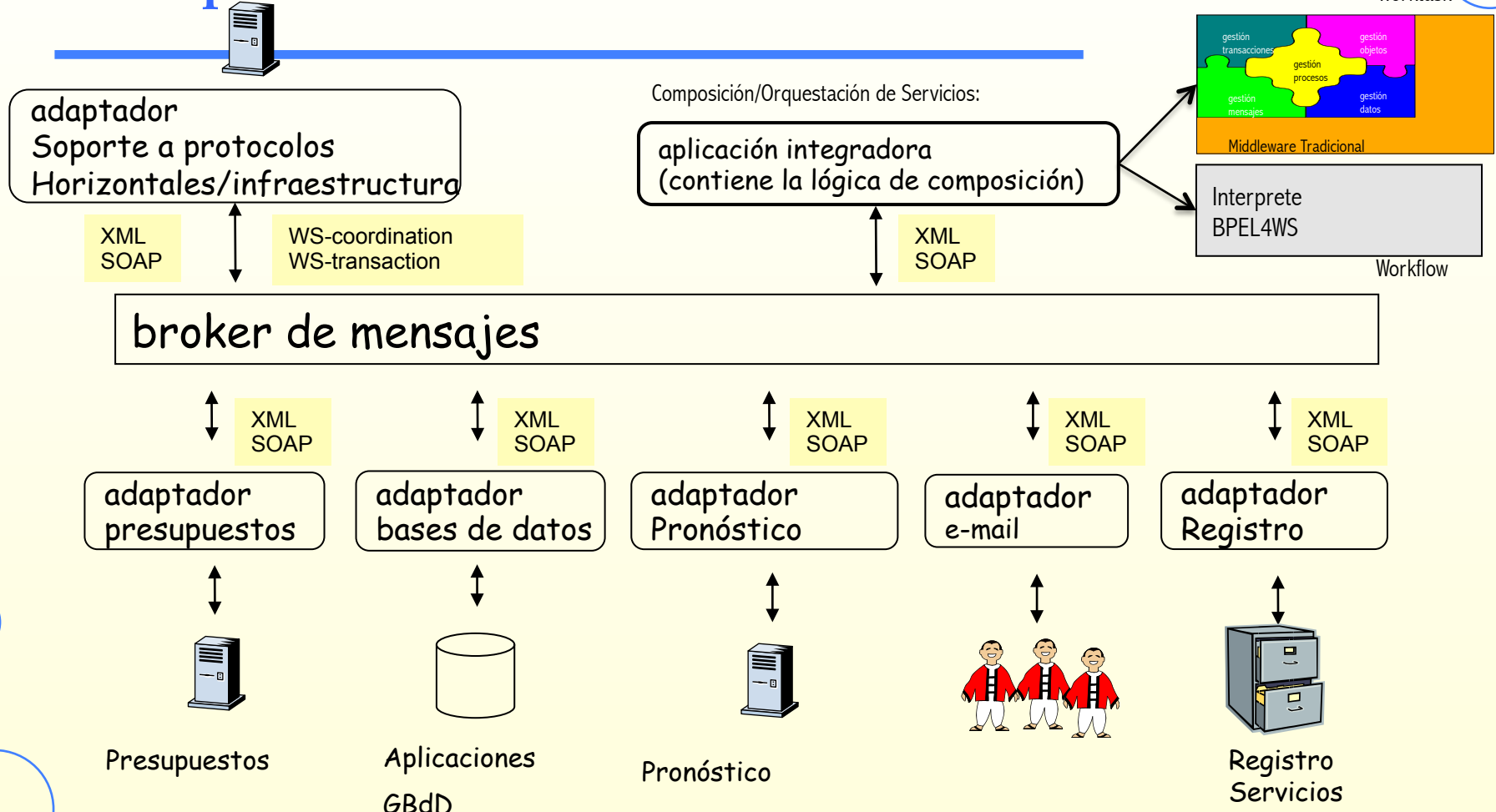


# Lógica de negocio

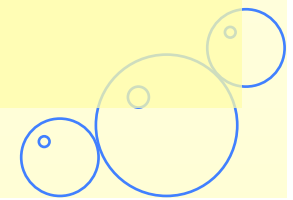
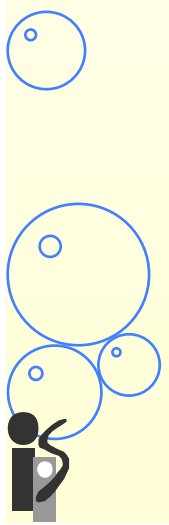
- **Sistemas de Gestión de Workflows**
  - Interacciones con servicios
  - Soporte al ciclo de vida
  - Soporte a fallos
- **BPEL4WS** (*Business Process Execution Language for Web Services*)

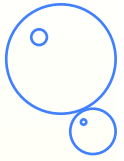
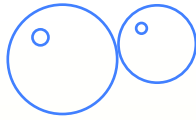


# Enterprise Server Bus



WSDL/WSCDL/WSCI  
 UDDI  
 RDF, OWL-S



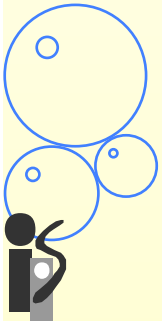


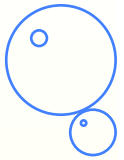
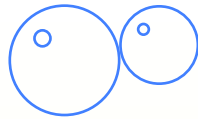
# REST

---

- **La alternativa a la torre de babel de los WS-\***
- **Simplemente identifica las claves del éxito de la Web y define un estilo arquitectural**
  - Representational State Transfer (REST)," from *Architectural Styles and the Design of network-Based Software Architectures*, Doctoral Dissertation, by Roy Thomas Fielding.

**¡La mayoría de los proveedores de servicios hoy en día utilizan el estilo REST**

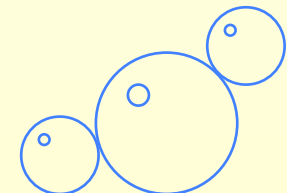
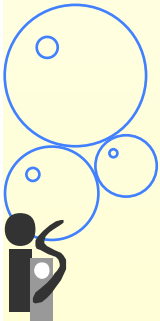


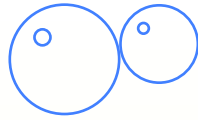


## REST identifica las claves de la Web

---

- **Intercambio de documentos identificados por URIs**  
*(Uniform Resource Identifier).*
- **HTTP se basa en el modelo cliente/servidor:**
  - Un cliente envía una petición a un servidor que consta de:
    - un método, una URI, una versión de protocolo y un mensaje al estilo MIME.
  - El servidor devuelve una línea de estado (indicando protocolo y éxito o error), y un mensaje al estilo MIME conteniendo el **documento** y cierra la conexión.
- **INTERFAZ UNIVERSAL:**
  - Los **métodos** son:
    - **GET** (recupera documento), **POST** (adjunta información al recurso), **PUT** (almacena información), **DELETE** (borra el recurso indicado).

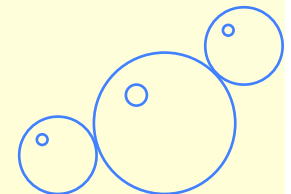
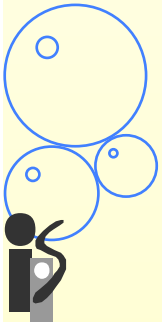


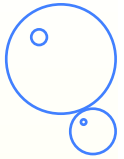
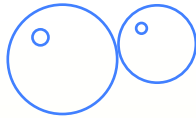


## Invocación de un SW

---

[http://webservices.amazon.com/onca/xml?  
Service=AWSECommerceService&AWSAccessKeyId  
=1JFWX63WKHTWX34G4KG2&Operation=ItemSearch  
&Keywords=Tim%20Berners&SearchIndex=Books](http://webservices.amazon.com/onca/xml?Service=AWSECommerceService&AWSAccessKeyId=1JFWX63WKHTWX34G4KG2&Operation=ItemSearch&Keywords=Tim%20Berners&SearchIndex=Books)

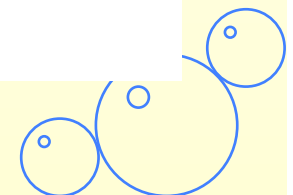
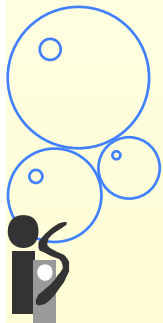
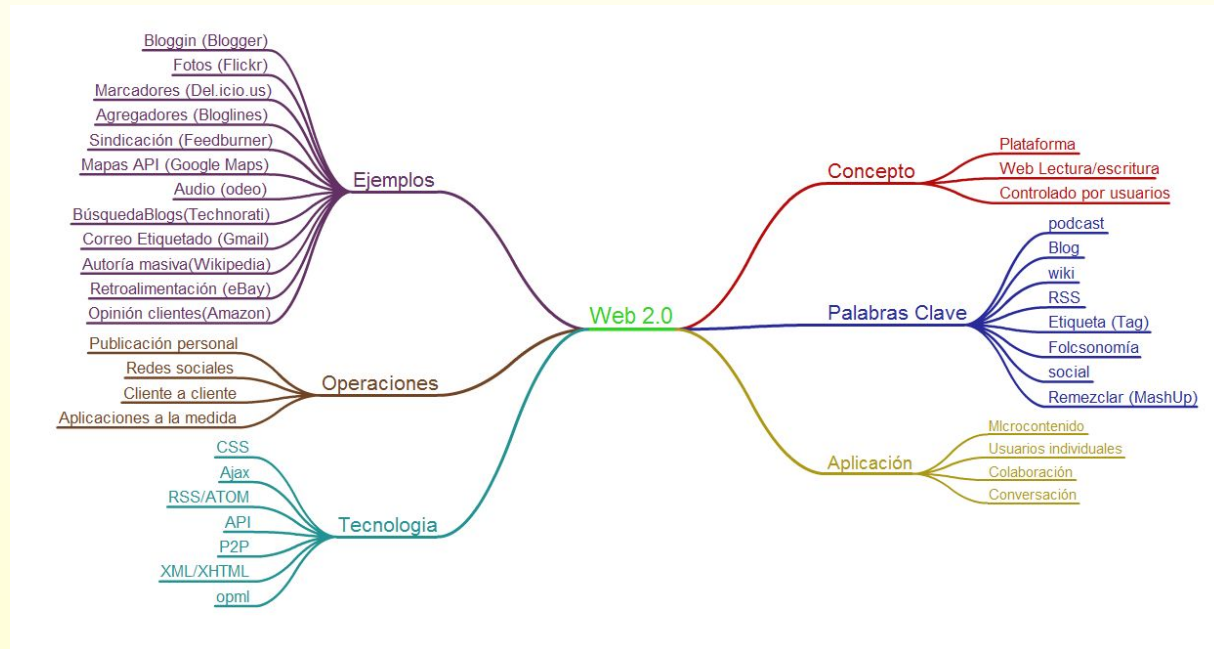


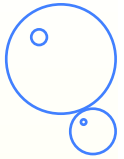
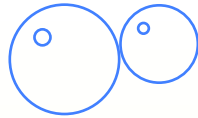


# Web 2.0

## • <definición>Wikipedia</definición>

El término Web 2.0 fue acuñado por O'Neill Media en 2004 para referirse a una segunda generación de Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs, los wikis o las folcsonomías, que fomentan la colaboración y el intercambio ágil de información entre los usuarios



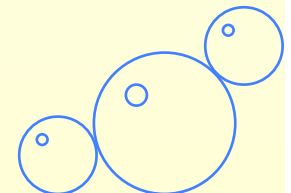
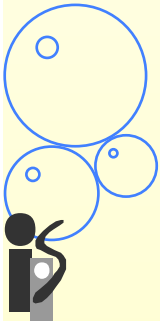


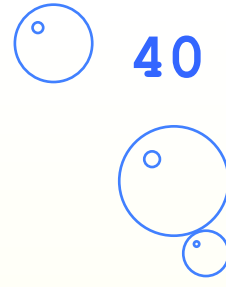
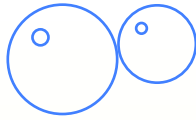
## Tim Berners-Lee opina

---

- **¡No hay un significado preciso de la Web 2.0!**
- **Cuando se le pregunta si *“es justo decir que la diferencia entre la Web 1.0 y la Web 2.0 es que la primera conecta computadores y la segunda conecta gente”*, Berners-Lee lo niega con rotundidad: *“La Web 1.0 conecta gente”*.**

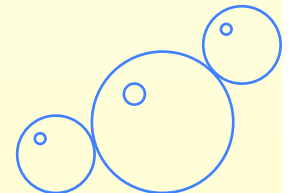
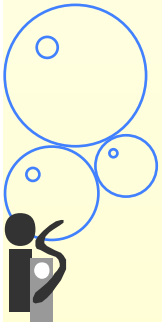
Tim Berners-Lee on Web 2.0: "nobody even knows what it means"



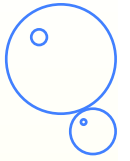
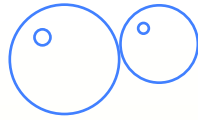


## Web 3.0: La Web semántica y mas allá ...

# EL FUTURO





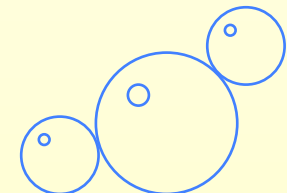
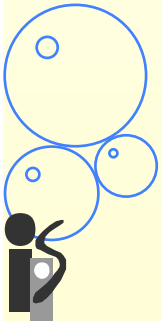


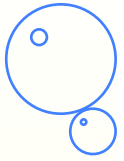
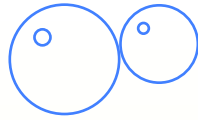
## La Vision

---

*Tengo un visión de la Web en la que los computadores serán capaces de analizar todos los datos de la Web – El contenido, los enlaces, y todas las transacciones entre la gente y las computadoras. La Web Semántica que haría esto posible no ha surgido todavía, pero cuando lo haga, el día a día del comercio, la burocracia y nuestras vida será manejado por maquinas que dialogan entre ellas. La gente de los agentes inteligentes materializara esta visión.*

**Berners-Lee, Tim; Fischetti, Mark (1999). *Weavingthe Web*. HarperSanFrancisco, chapter 12. ISBN 9780062515872.**



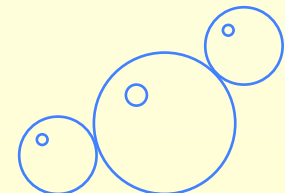
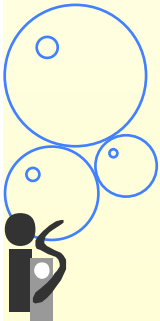


## Inteligencia Artificial

---

*Proyectando las tendencias actuales en el futuro, pienso que habrá un nuevo énfasis sobre sistemas autónomos- robots y softbots. Softbots son agentes software que navegan por Internet, buscando información que pueda ser interesante para sus usuarios. La presión para mejorar las capacidades de los robots y los agentes software motivarán y guiarán la investigación en IA durante los próximos años."*

*(Artificial Intelligence, A new Synthesis. Nilsson 1998)*



Agentes Web que conocen, aprenden y razonan como los humanos

Artificial Intelligence      Intelligent Agents

Personal Assistants

Ontologies      3      Semantic Webiste

**La web Semántica**  
**Conecta Conocimiento**

Semantic Search

Thesaurus & Taxonomies      Knowledge Bases      Semantic Desktop      Bots

Autonomic Intellectual Property      Smart Markets      Semantic Agents

Natural Language      4      Semantic Communities

Blogets      **La web ubicua**  
**Conecta Inteligencia**

Semantic Blog      Semantic Wiki      Semantic Social Network

Semantic email

Enterprise Portals

Search Engines      Web sites

Content Portals      1      PIMS

**La web**  
**Conecta Información**

Databases      "Push Publish & Subscribe"      P2P file sharing

File Servers

Mash-up      Wiki      Multi-user Gaming      Community Portals

Blogs      RSS      2      Marketplaces      Social Bookmarking

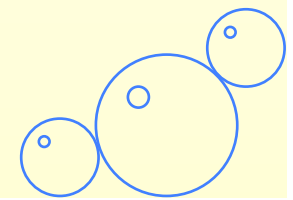
**La web Social**  
**Conecta Gente**

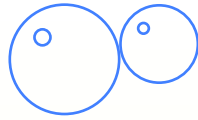
Email      Conferencing      Social network

Instant messages

Incrementar conectividad Social

Project10X's Semantic Wave 2008 Report: Mills Davis, Managing Director, Project 10X  
www.project10x.com

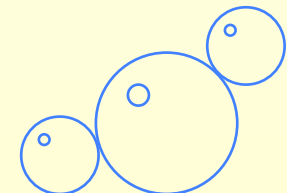
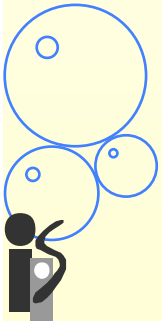


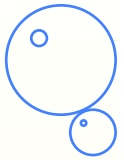
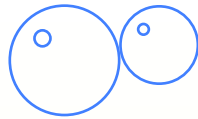


## Se tiende a pensar en los servicios Web como ...

Extensiones a las soluciones para integrar aplicaciones y como un middleware convencional que facilita la integración de sistemas

Un vehículo para comercio electrónico y sistemas de información globales

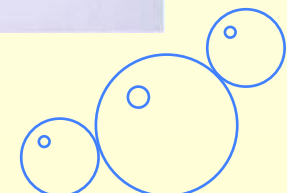
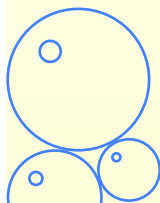




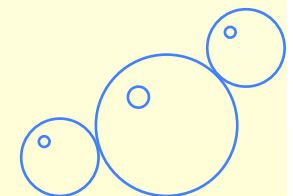
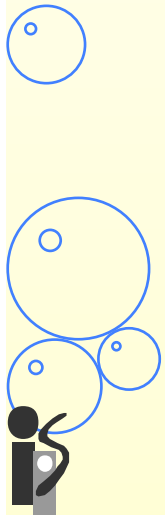
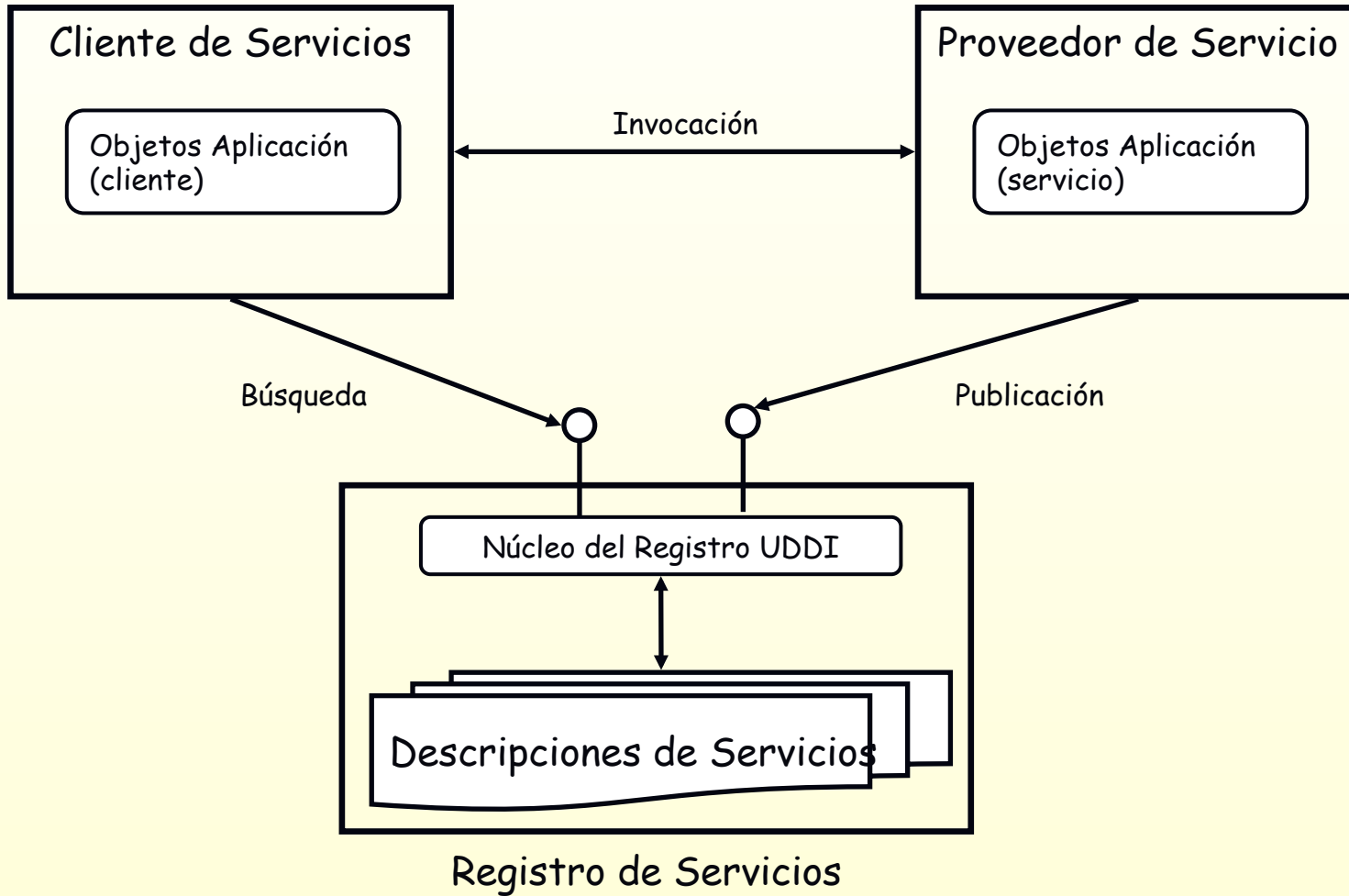
## Las ciencias artificiales ...

---

Lo más relevante del software es que no está sujeto a las leyes de la naturaleza y esto nos lleva a pensar que podemos ignorar el sentido común

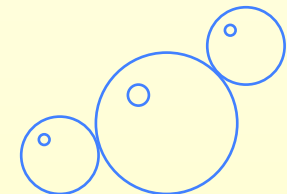
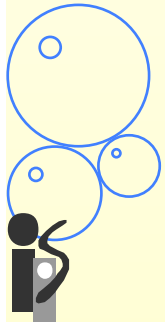
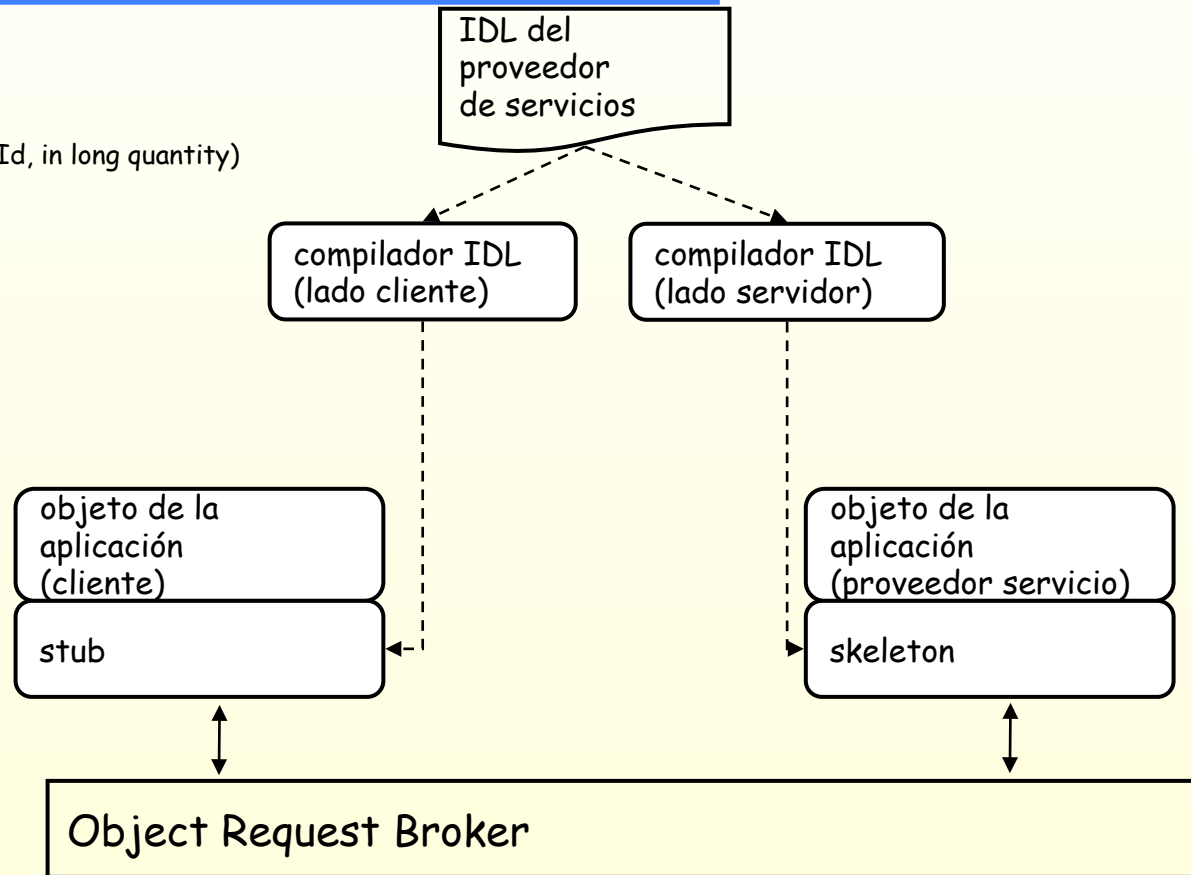


# Problema de interoperabilidad sin resolver



# CORBA

```
interface Purchasing {  
    float getQuote ( in long productId);  
    float purchaseGoods (in long productId, in long quantity)  
}
```



# Middleware de Servicios Web

```
<operation name="orderGoods">  
<input message = "OrderMsg"/>  
</operation>
```

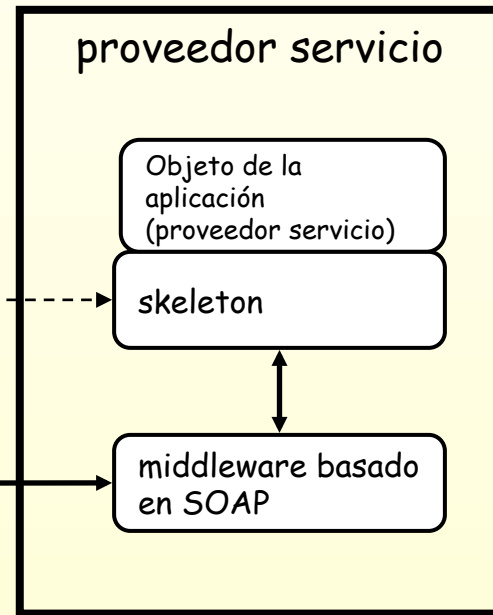
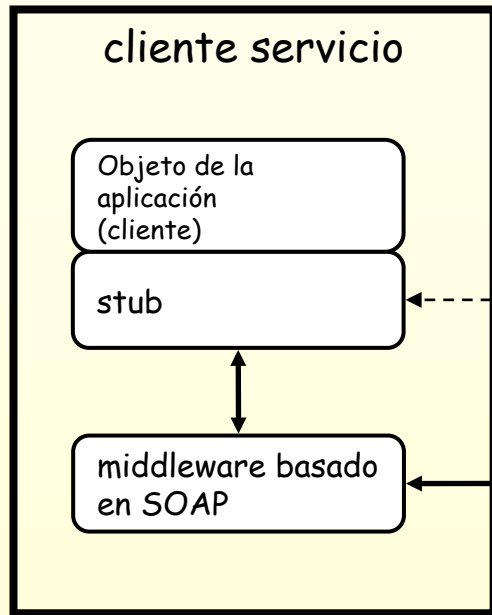
WSDL del proveedor del servicio

Diferencias:

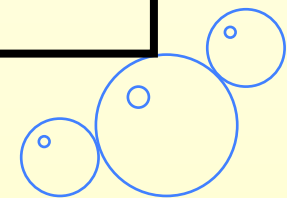
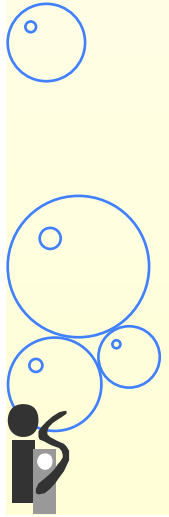
- Ausencia de un middleware común ejecutable.
- Ocultación de los detalles de implementación

compilador WSDL (lado cliente)

compilador WSDL (lado servidor)

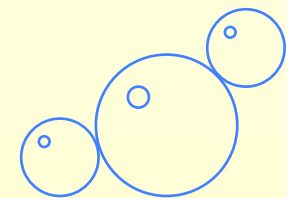
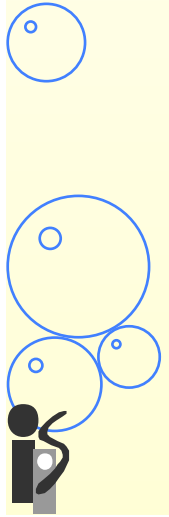
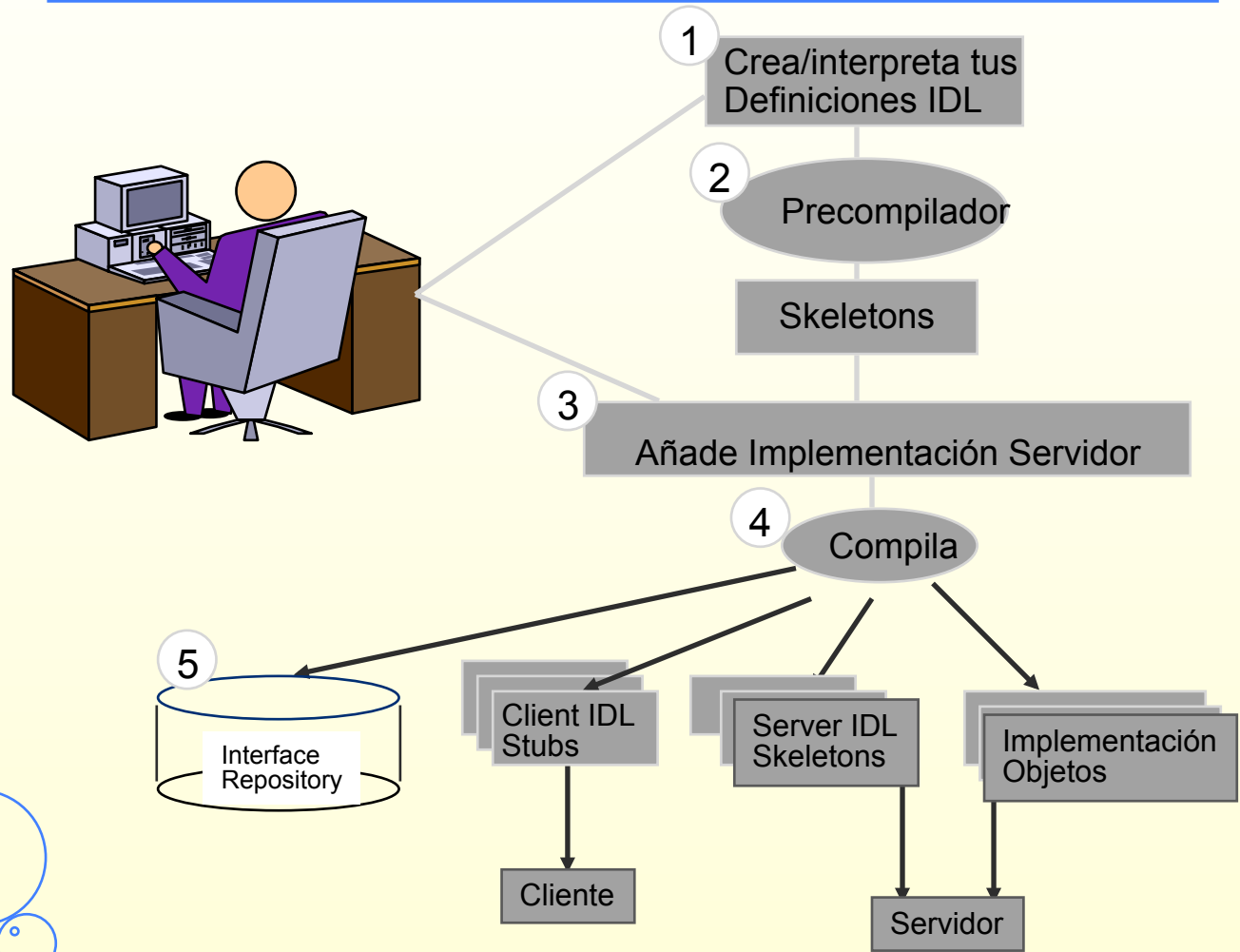


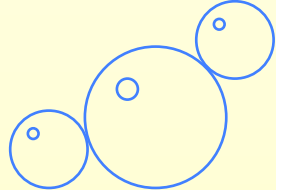
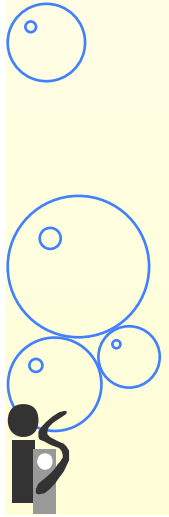
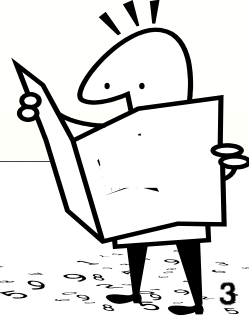
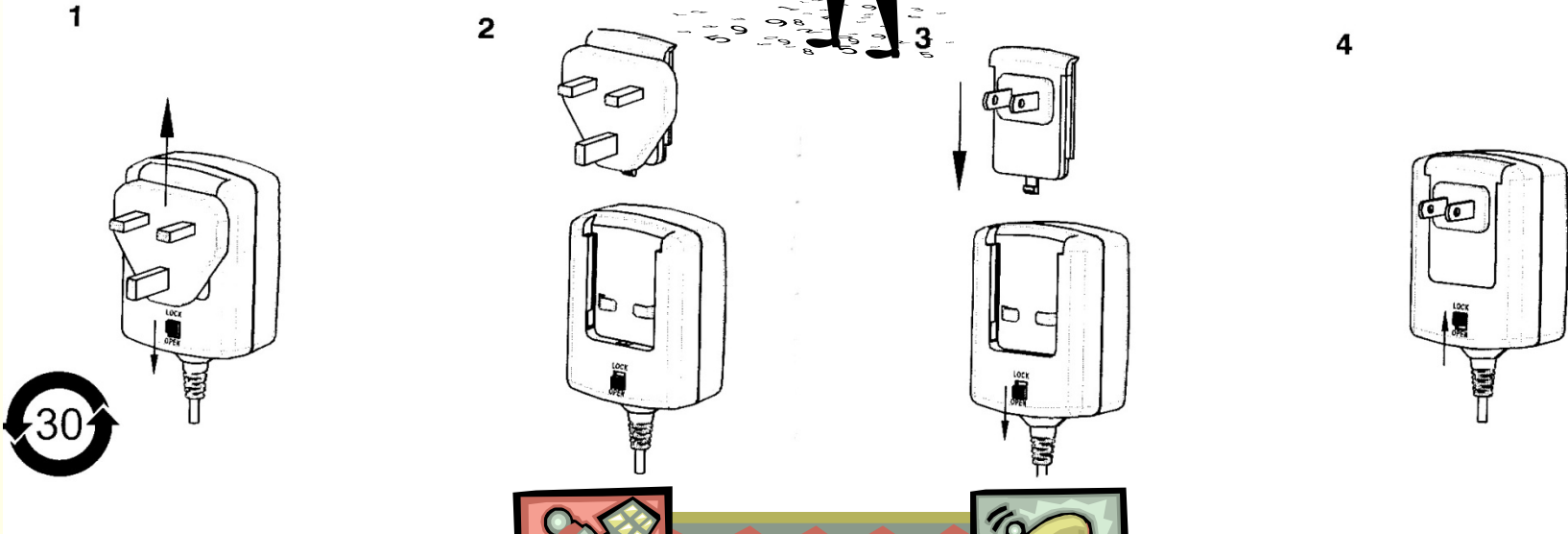
mensajes SOAP

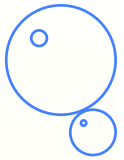
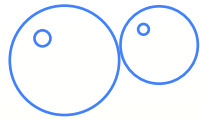




# Desarrollo con Middleware tradicionales y de Servicios Web







## Los Servicios Web hoy ...

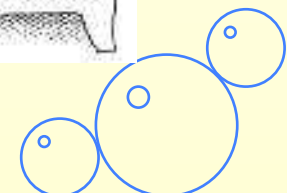
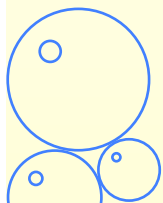
---

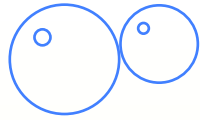
- **Desde el punto de vista del programador**



A BRIEF HISTORY OF ARTIFICIAL INTELLIGENCE

[http://www.atariarchives.org/deli/artificial\\_intelligence.php](http://www.atariarchives.org/deli/artificial_intelligence.php)

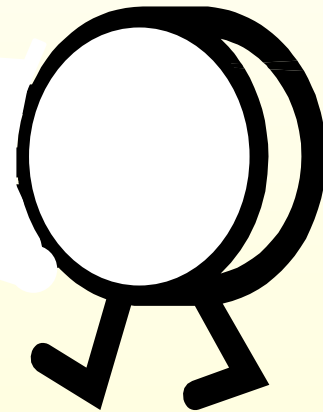
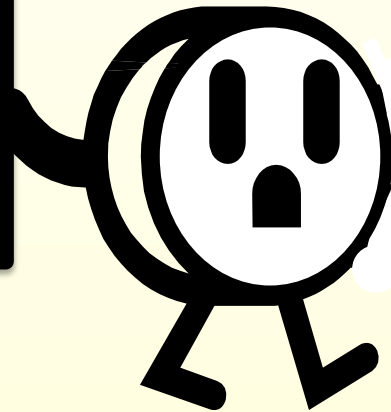




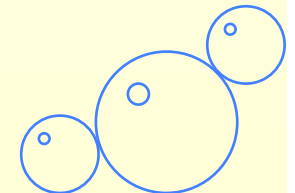
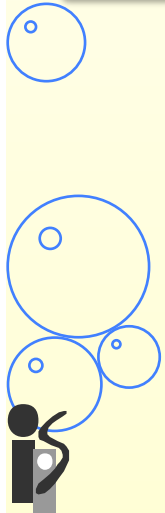
# La Interoperabilidad en el Futuro

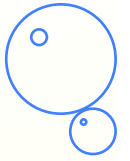
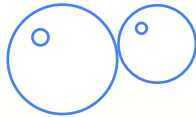
---

Componente Software



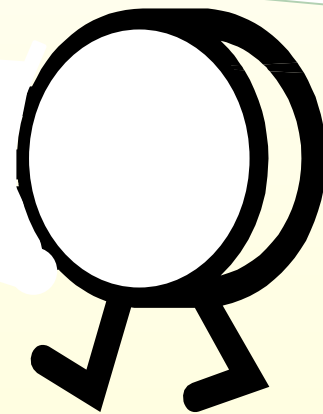
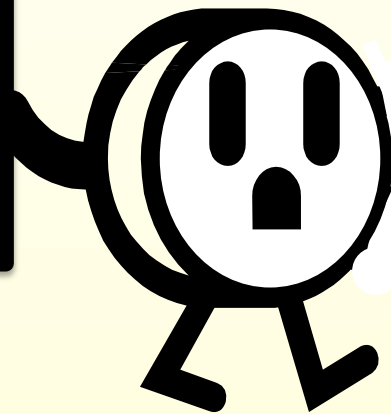
Componente Software



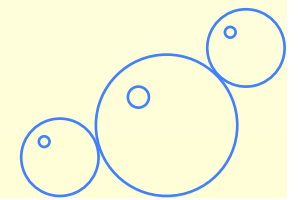
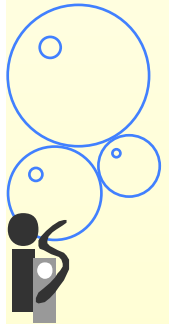
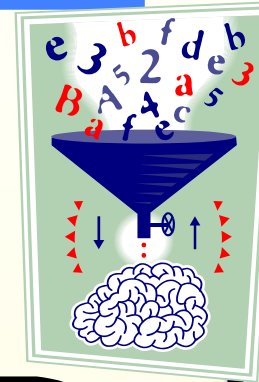


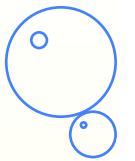
# La Interoperabilidad en el Futuro

Componente Software



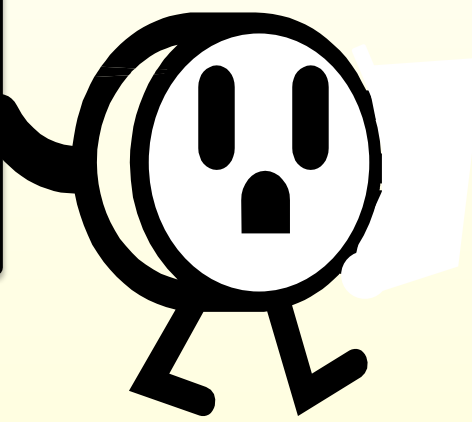
Componente Software



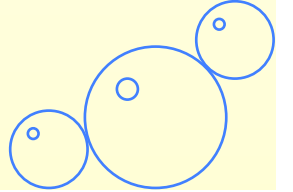
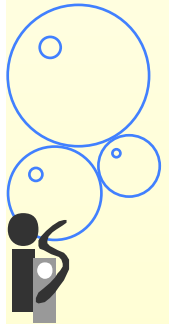
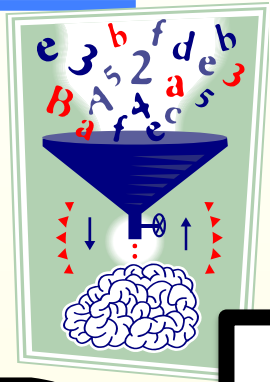


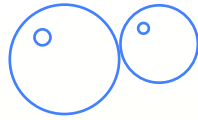
# La Interoperabilidad en el Futuro

Componente Software



Componente Software

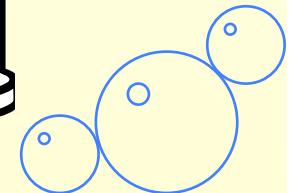
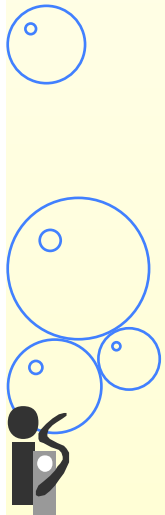


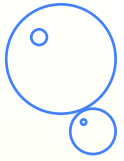
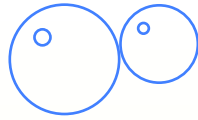


## Agentes

---

- Los agentes son:  
**Entidades autónomas**
- Los agentes tienen:  
**Inteligencia**
- Los agentes no actúan solos, sino como:  
**Sistemas Multi-Agentes**

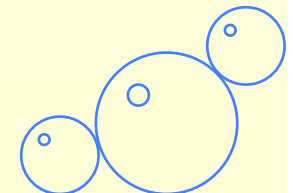
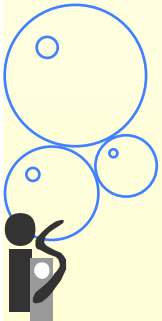




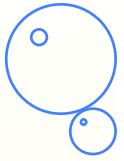
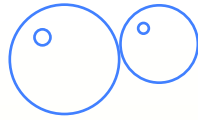
## Representación del conocimiento

---

- **RDF: *ResourceDescription Framework.***
  - Representación estructurada del conocimiento
  - Tripletas objeto, atributo, valor
  - Herramientas
    - Repositorios de tripletas
    - Lenguaje de Consulta: SPARQL: *SPARQL Protocol and RDF Query Language.* (Estandar 15 de enero de 2008).
    - Generación a datos RDF: GRDDL: *Gleaning Resource Descriptions from Dialect soft Languages (de XML/XHTML a RDF)*



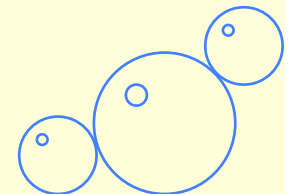
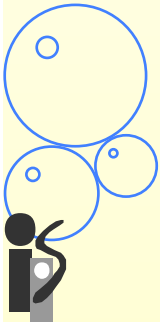


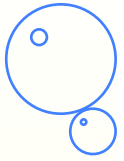
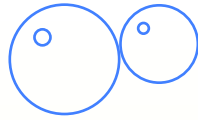


## Lenguajes del Web Semántico

---

- **SKOS: Simple Knowledge Organization System**
  - Tesoros, clasificaciones, taxonomías a RDF
- **OWL**
  - Ontologías. Representaciones semánticas. Define relaciones, conceptos, ...
- **Rules**
  - Conocimiento declarativo.
  - RIF (Rule Interchange Format)

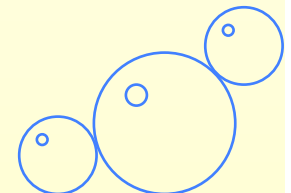
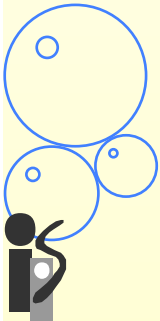


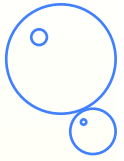
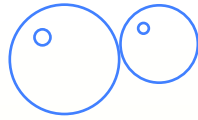


## Web Semantic ServiceArchitecture

---

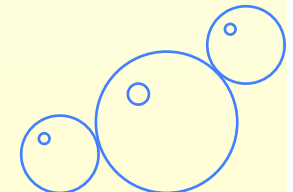
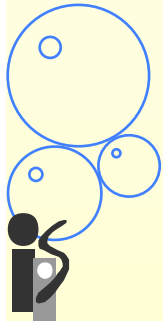
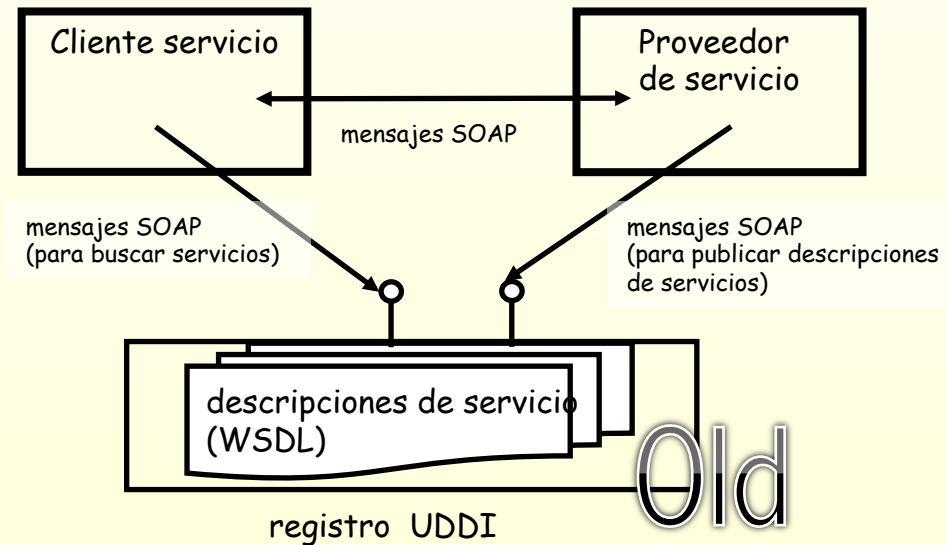
- **La contribución de los sistemas multi agentes**
  - Describen las fases y los protocolos de interacción entre agentes/ entidades
- **Estándar Arquitectura de agentes**
  - FIPA (The Foundation of Intelligent Physical Agents)
  - Arquitectura, lenguaje de comunicación, protocolos

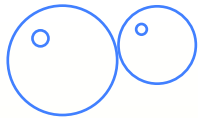




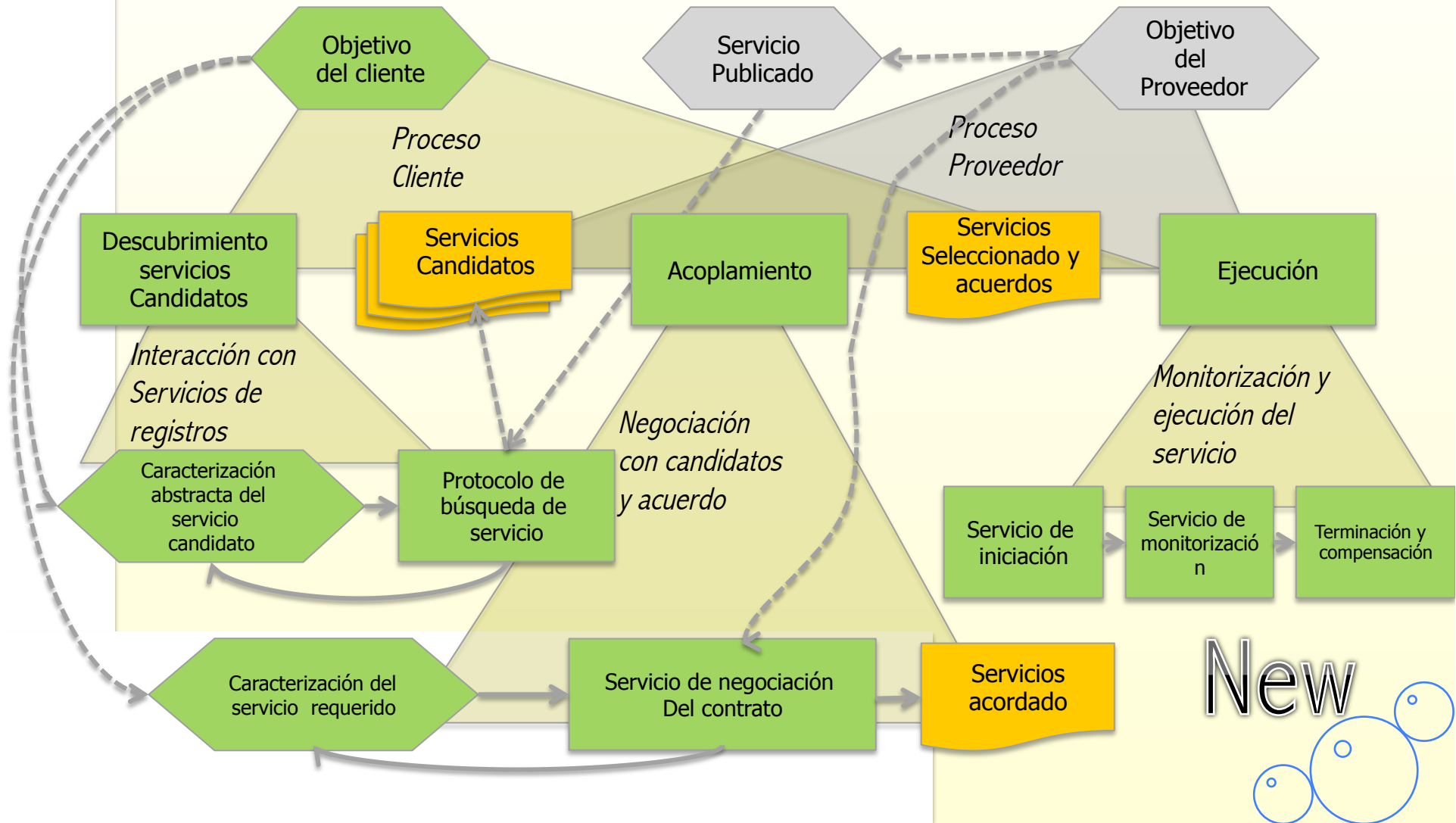
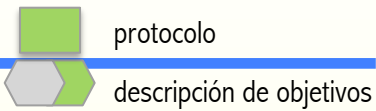
# Web Semantic ServiceArchitecture

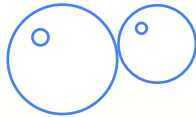
- **La contribución de los sistemas multi-agentes**
  - Trasladan las fases y los protocolos de interacción entre agentes/ entidades a la Web





# Web Semantic ServiceArchitecture

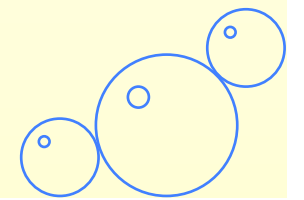
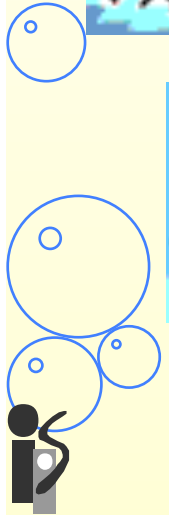




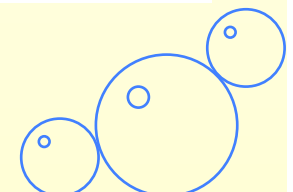
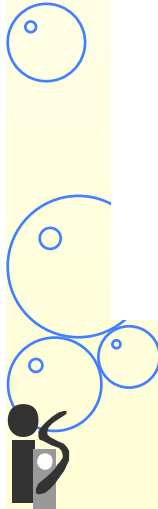
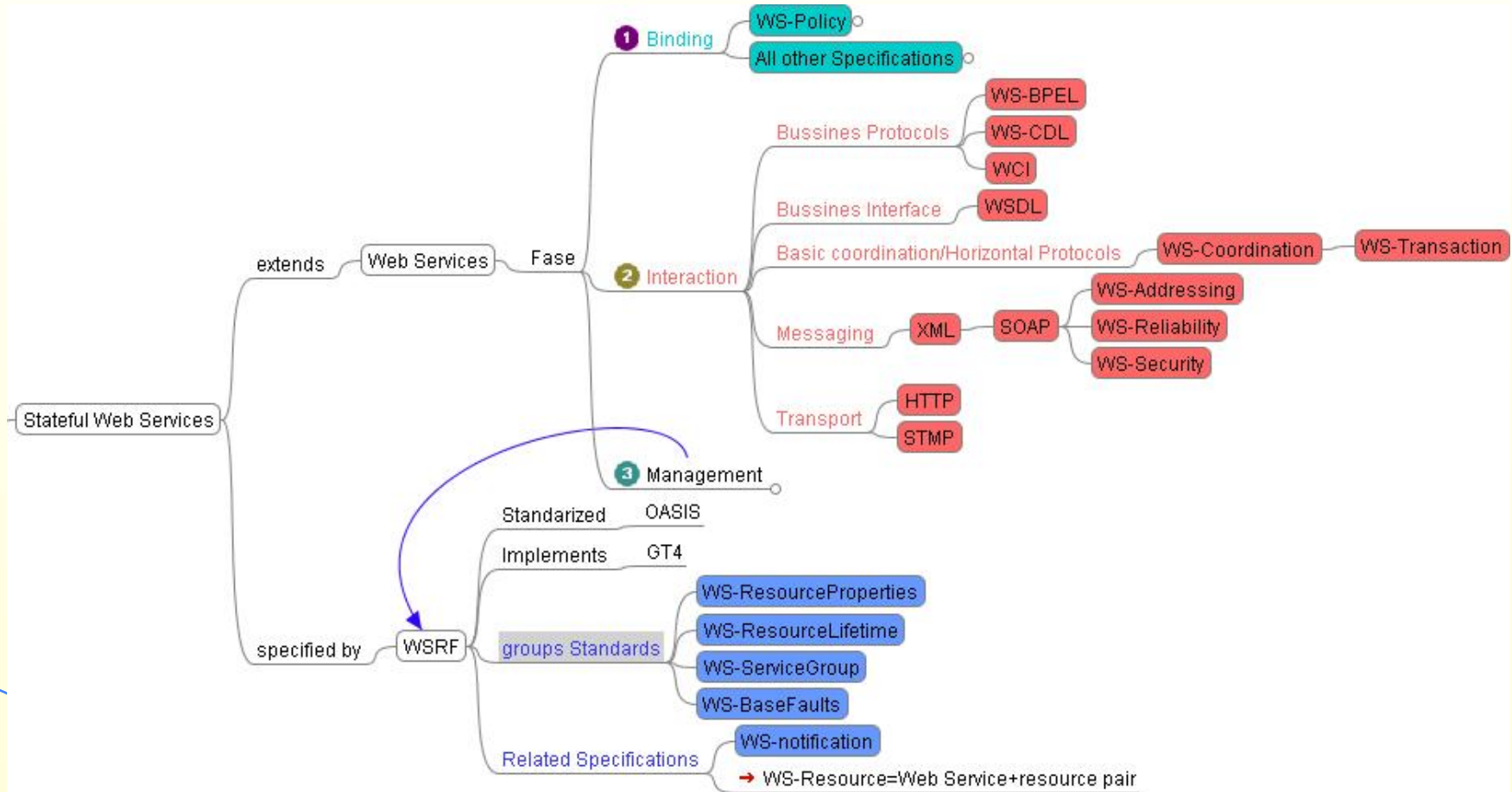
# Grid Computing/WSRF

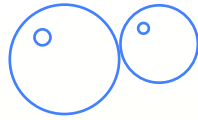
---

- **Analogía con el suministro eléctrico**



# Grid Computing

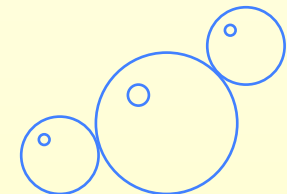
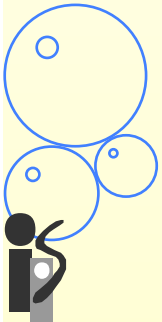




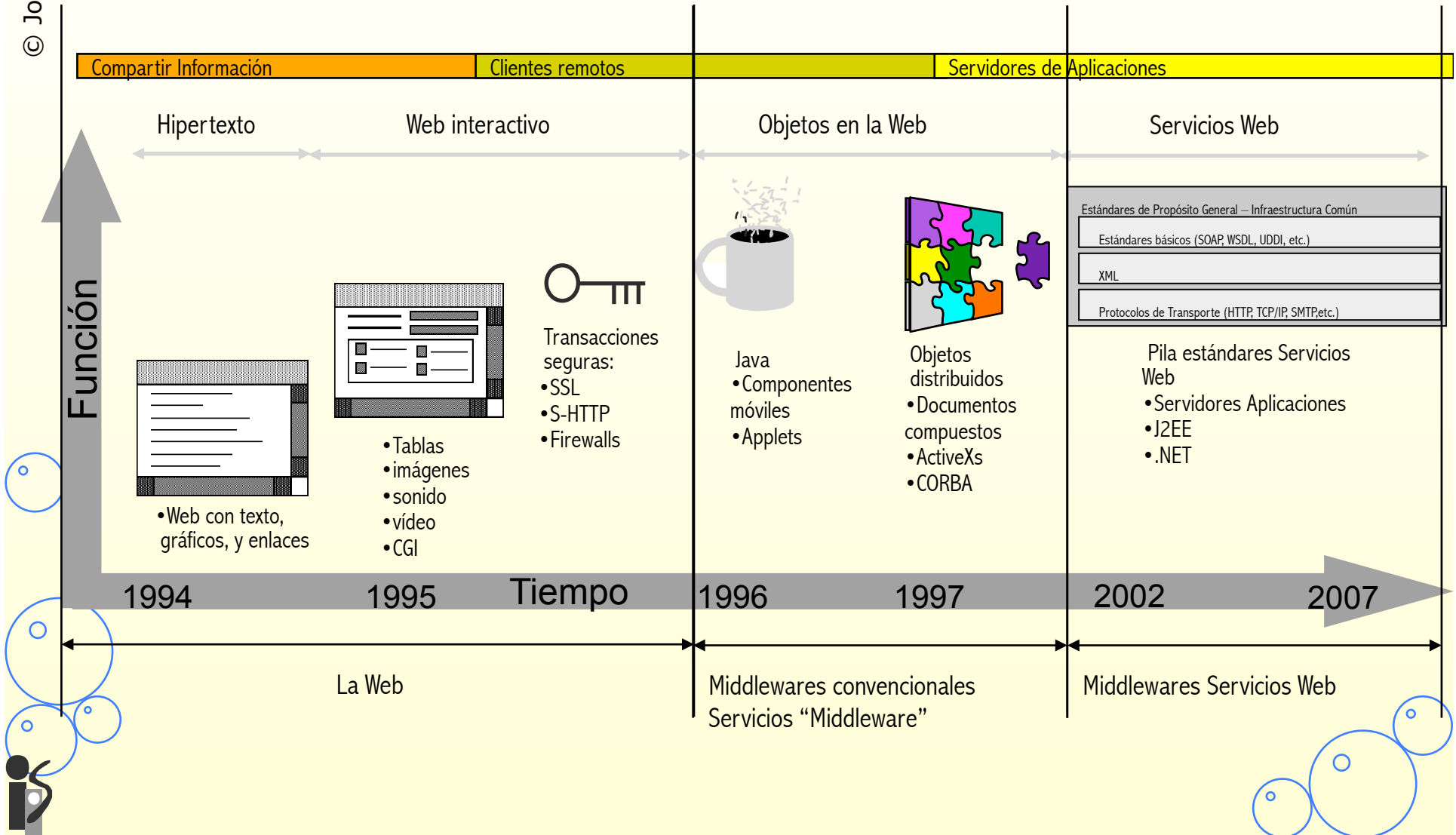
## Autonomic Computing

---

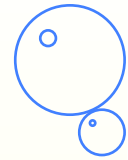
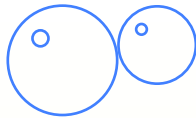
- **En un sistema autogestionado, el programador juega un nuevo papel; No controla el sistema directamente. En su lugar define políticas y reglas para que los procesos se auto gestionen:**
  - **Auto-configuración**
  - **Auto-reparación**
  - **Auto-optimización**
  - **Auto-protección**



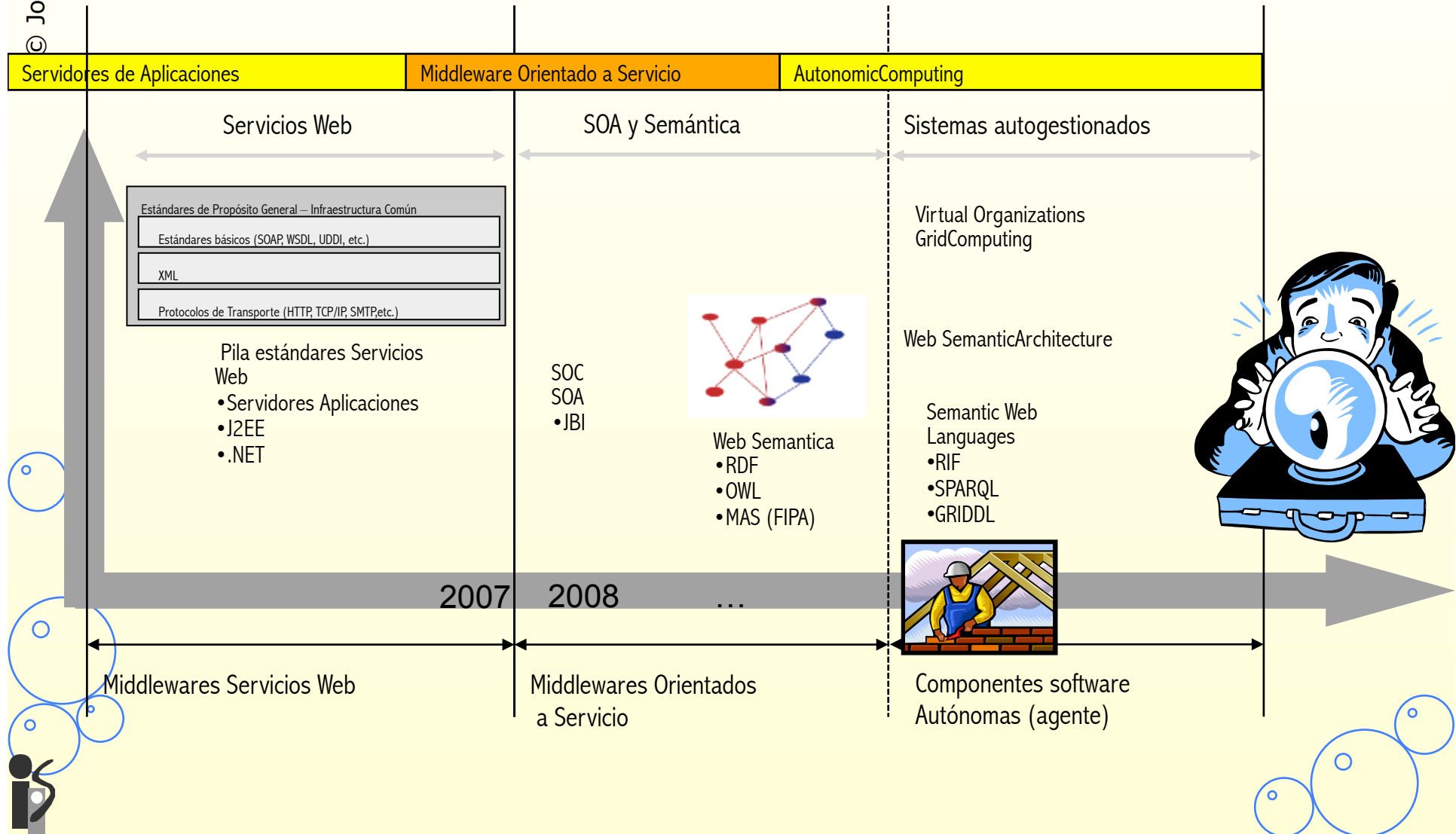
# Evolución Tecnologías Web

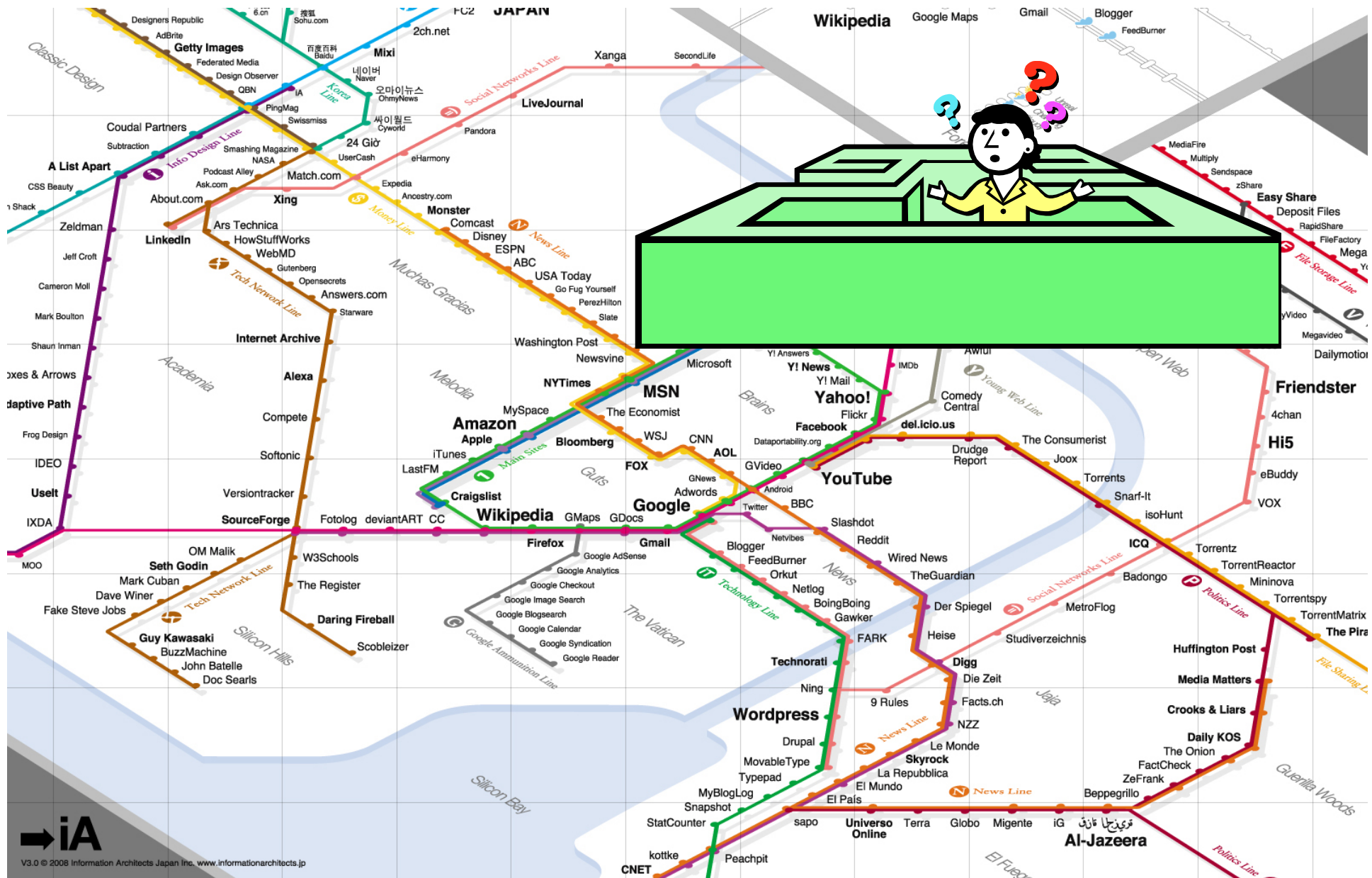






# Evolución Tecnologías Web

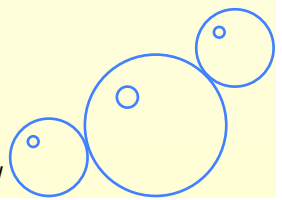
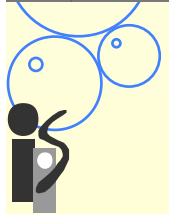


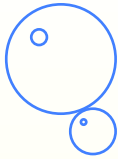
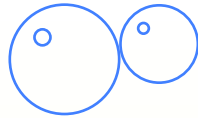


iA

V3.0 © 2008 Information Architects Japan Inc. www.informationarchitects.jp

iAInformation Architects Japan Web Trend Map  
<http://informationarchitects.jp/ia-trendmap-2007v2/>





## Referencias

---

- **Alonso · Casati · Kuno · Machiraju Web Services Concepts, Architectures and Applications. Springer Verlag 2004.**
- **Hamid R. Motahari Nezhad, Boualem Benatallah, Fabio Casati, and Farouk Toumani. Web Services Interoperability Specifications. *Computer*, 39(5):24--32, 2006.**
- **Mark H. Burstein, Christoph Bussler, Michal Zaremba, Timothy W. Finin, Michael N. Huhns, Massimo Paolucci, Amit P. Sheth, Stuart K. Williams: A Semantic Web Services Architecture. *IEEE Internet Computing*, 9(5): 72-81, 2005.**
- **Michael Stal. Using Architectural Patterns and Blueprints for Service-Oriented Architecture. *IEEE Software*. 23(2): 54-61, 2006.**
- **Curbera, F., Khalaf, R., Mukhi, N., Tai, S., and Weerawarana, S. 2003. The next step in Web services. *Commun. ACM* 46, 10 (Oct. 2003), 29-34.**
- **R. Khalaf, A. Keller, F. Leymann. Business processes for Web Services: Principles and applications. IBM SYSTEMS JOURNAL, VOL 45, NO 2, 2006**
- **Mike P. Papazoglou. Service -Oriented Computing: Concepts, Characteristics and Directions. Proceedings of the Fourth International Conference on Web Information Systems Engineering (WISE'03)**
- **Presentación del 2008-03-07, State of the Semantic Web, Karl Dubost and Ivan Herman, INTAP Semantic Web Conference 2008, in Tokyo, Japan (Disponible en <http://www.w3.org/2008/Talks/0307-Tokyo-IH/>)**
- **Project10X's Semantic Wave 2008 Report: Mills Davis, Managing Director, Project 10X. [www.project10x.com](http://www.project10x.com)**

