



# FACULTAD DE CIENCIA Y TECNOLOGÍA INGENIERÍA EN SISTEMAS DE INFORMACION SEDE ORO VERDE

Cátedra: Proyecto

Alumno: Emanuel Goette

# “Implementación de Arquitecturas de Software Orientadas a Servicios en la Web 2.0, basada en herramientas Open Source”

# Arquitecturas de Software Orientadas a Servicios

Sistema Ventas



Sistema Pagos



Sistema Stock y  
Producción



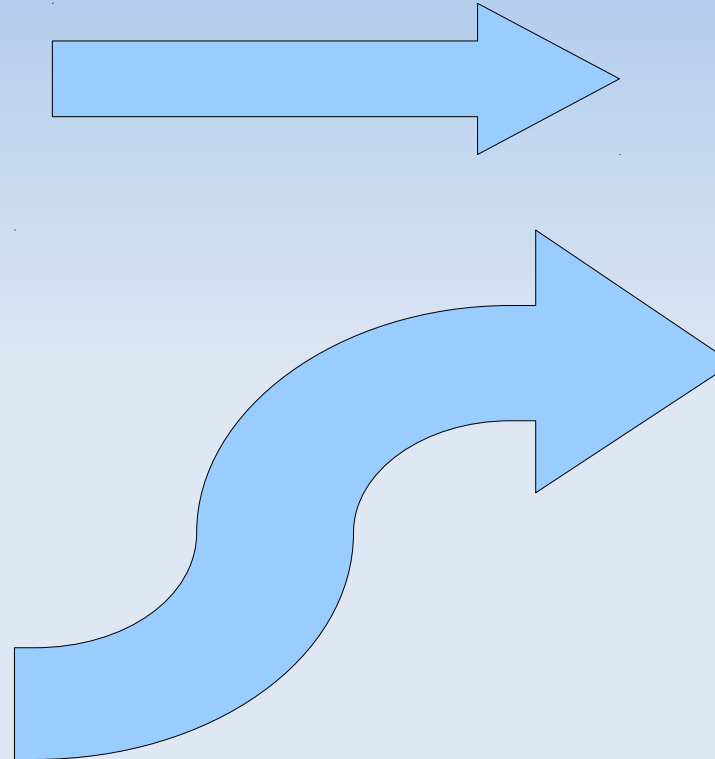
Sistema RRHH

# Arquitecturas de Software Orientadas a Servicios

Sistema Ventas



Sistema Pagos



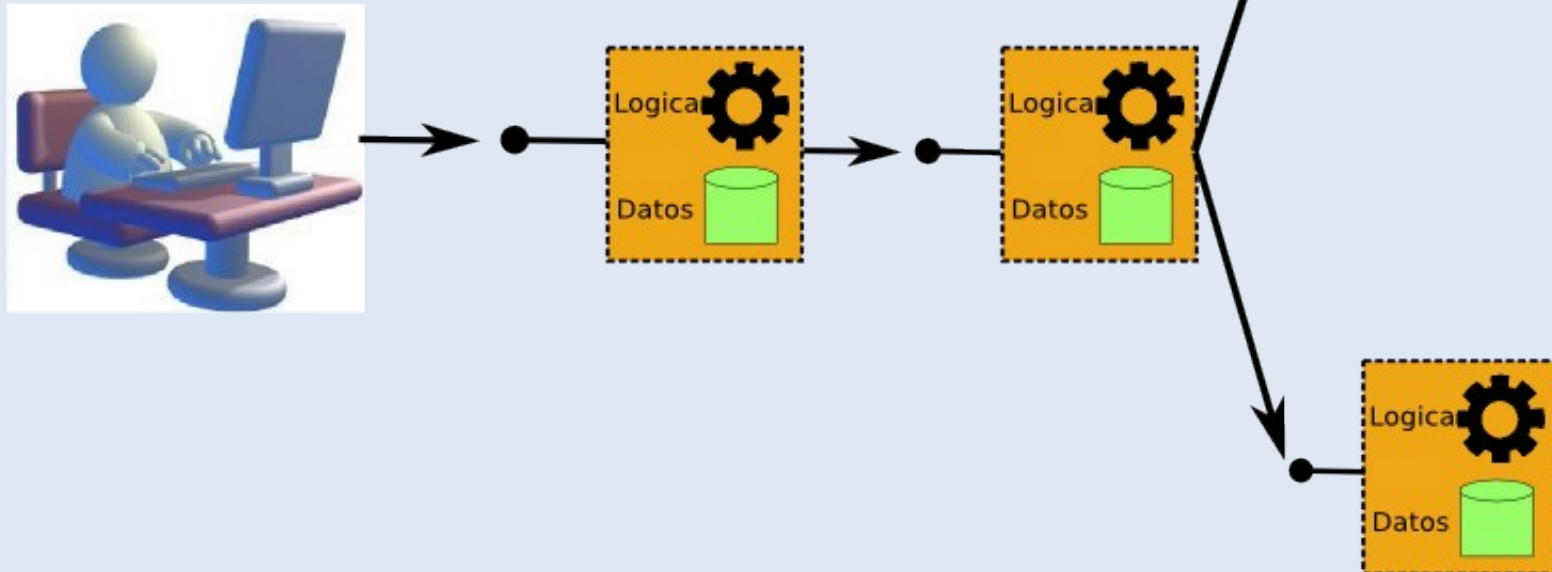
Sistema Stock y  
Producción



Sistema RRHH

# Arquitecturas de Software Orientadas a Servicios

- Interacción de sistemas
- El bajo acoplamiento
- Alta reutilización



# Arquitecturas de Software Orientadas a Servicios

“La Arquitectura Orientada a Servicios (en inglés Service Oriented Architecture), es un concepto de arquitectura de software que define la utilización de servicios para dar soporte a los requisitos del negocio.

Permite la creación de sistemas altamente escalables que reflejan el negocio de la organización, a su vez brinda una forma bien definida de exposición e invocación de servicios, lo cual facilita la interacción entre diferentes sistemas propios o de terceros.”

Wikipedia

# Arquitecturas de Software Orientadas a Servicios

La arquitectura SOA define :

- Los servicios.
- Como localizar un servicio.
- Como conseguir que se comuniquen diferentes servicios.
- Como encajar diferentes servicios para que funcionen como un sistema.

SOA = Servicios + Gobierno SOA

# Gobierno SOA

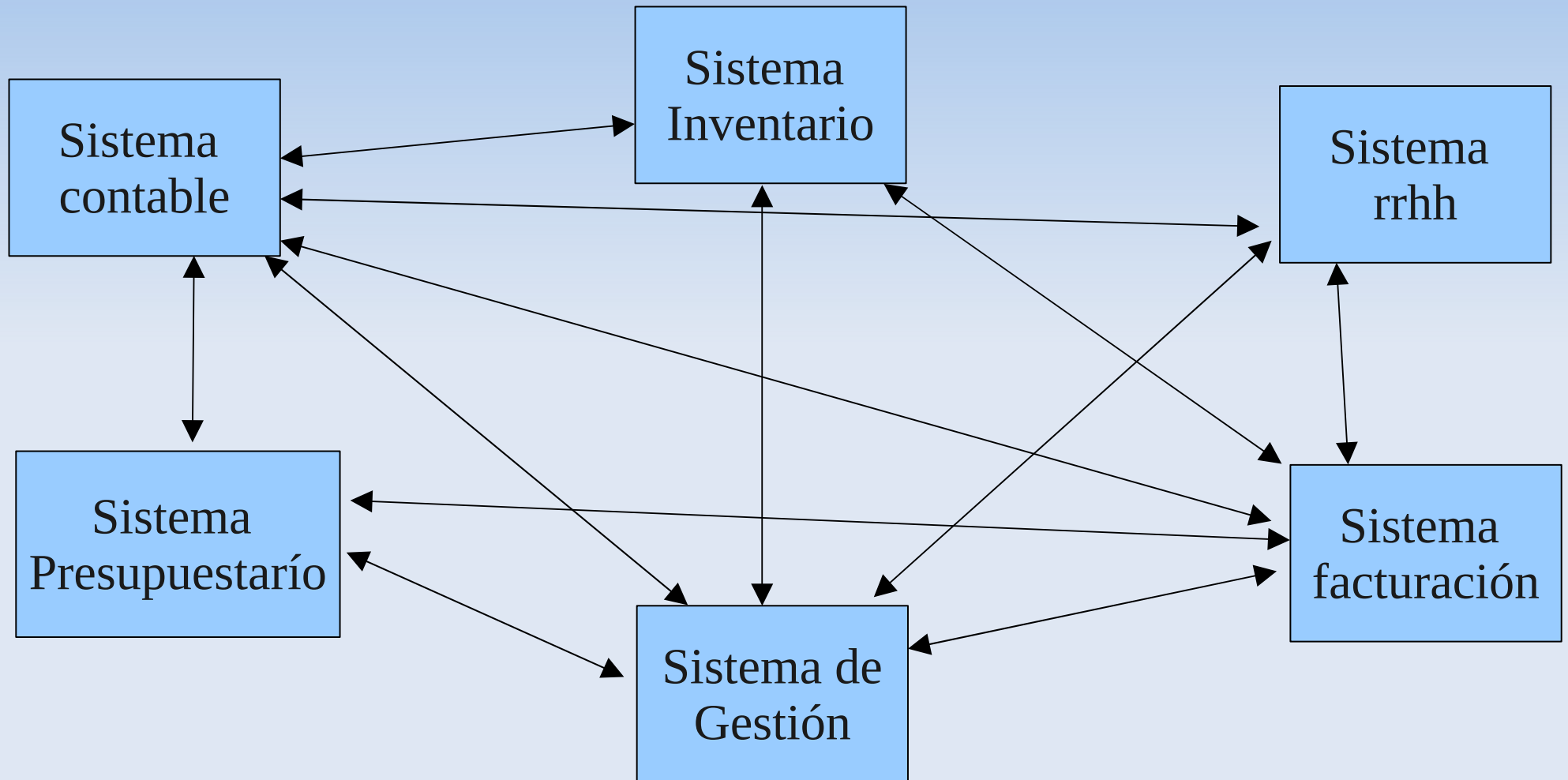
- El gobierno SOA es el conjunto de roles, políticas y procedimientos que sirven de guía para la adopción de la SOA. Al implementar los componentes tecnológicos de gobierno, está creando la infraestructura para soportar y aplicar estos roles, políticas y procedimientos en toda su SOA.



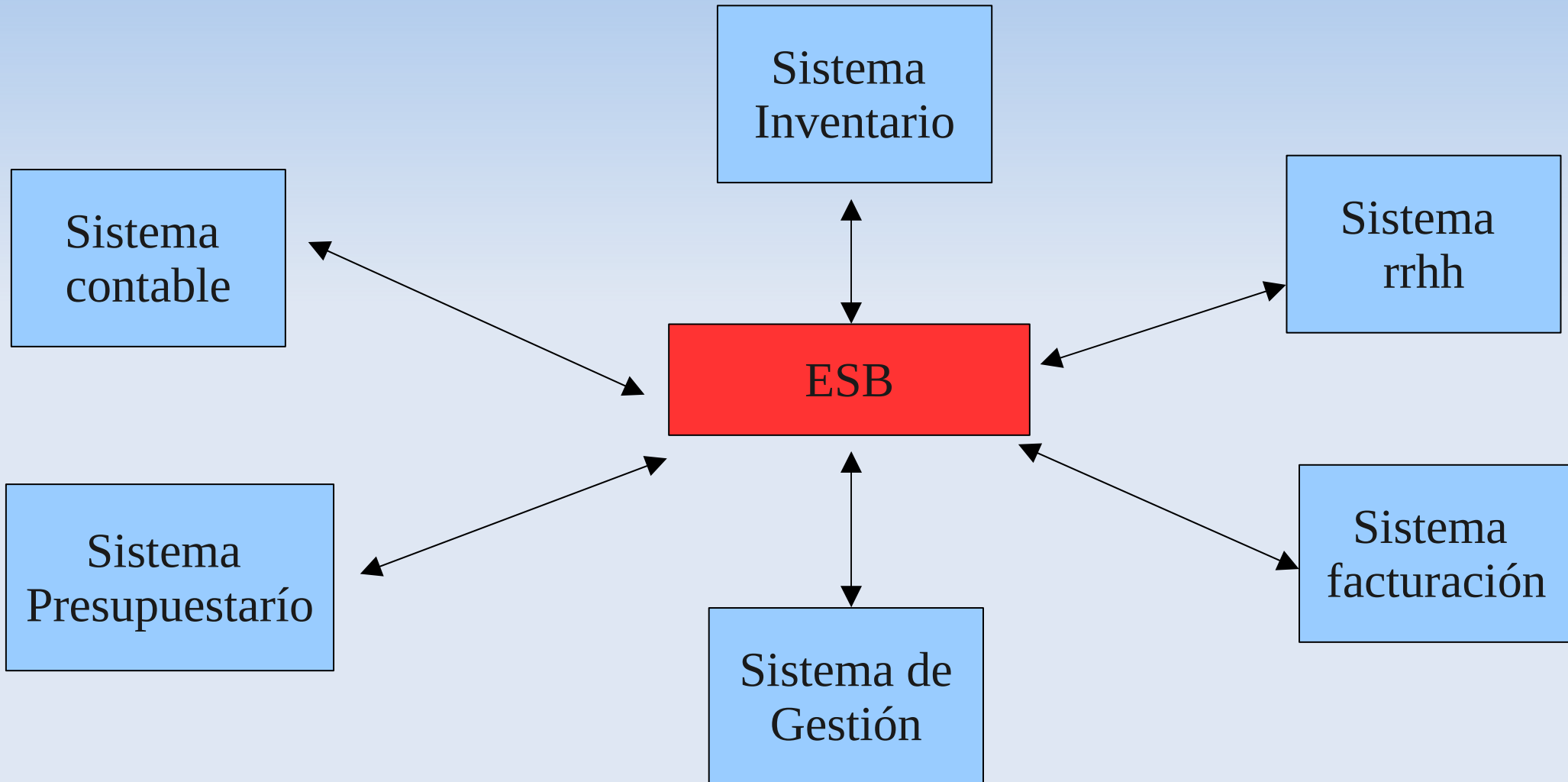
# Gobierno SOA

- Registro de servicios
- Bus de servicios o ESB (Enterprise Service Bus)
- Orquestación y coreografía
- CEP (Complex Event Processing) y Procesador de flujos de eventos o ESP(Event Stream Processing)
- EAI (Enterprise Application Integration)

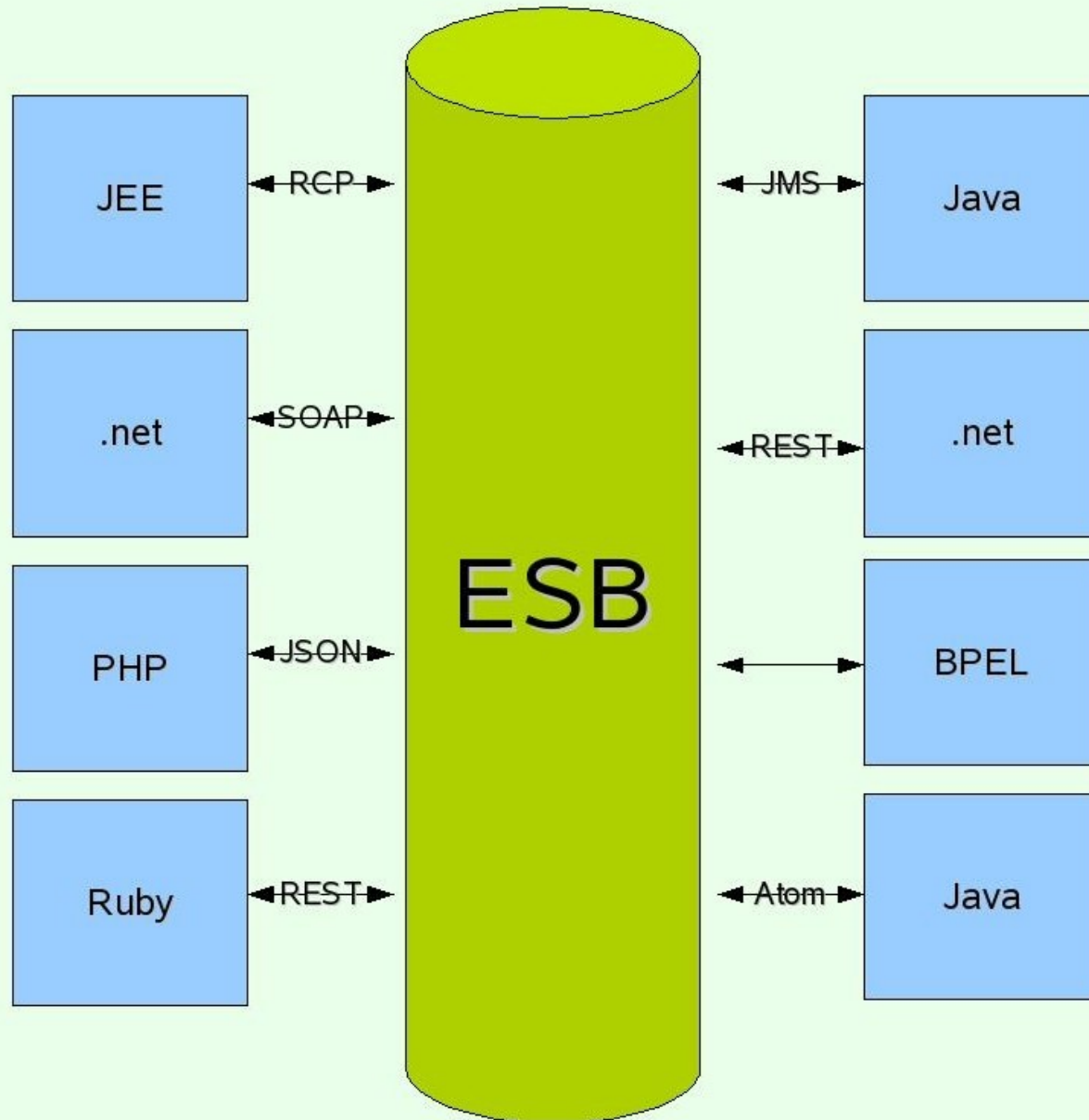
# ESB



# ESB



# ESB



# SCA

- Podríamos concebir una aplicación como un conjunto de componentes de software interrelacionados. Todos estos componentes están contruidos bajo las mismas o diferentes tecnologías. Estos componentes pueden correr sobre la misma máquina en un mismo sistema operativo y sobre una misma plataforma o en diferentes procesos, diferentes máquinas con diferentes plataformas y sistemas operativos. Sin embargo si una aplicación es organizada de esta manera se requiere una forma de crear los componentes y un mecanismo para describir cómo los componentes trabajan juntos.

# Requerimientos no funcionales de una Web 2.0

- Capacidad de escalar



# Requerimientos no funcionales de una Web 2.0

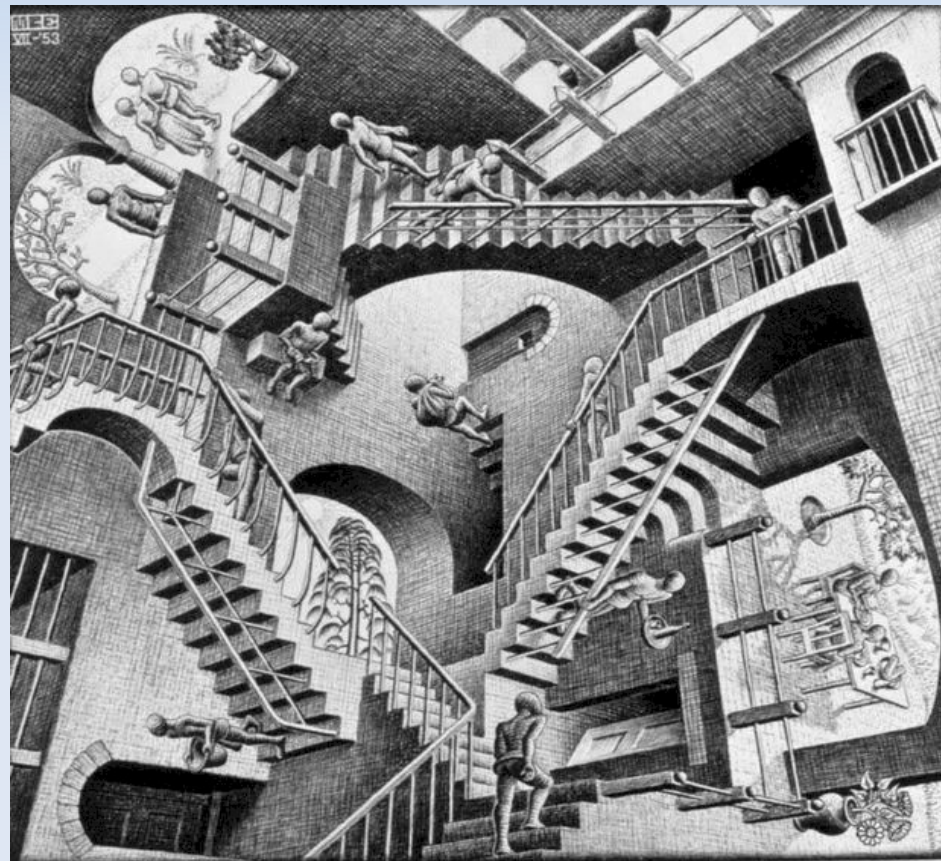
- Performance





# Requerimientos no funcionales de una Web 2.0

- Mantenibilidad






# Requerimientos no funcionales de una Web 2.0

- Comunicación con otras paginas



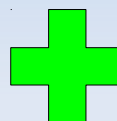


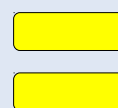


# Requerimientos no funcionales de una Web 2.0

- Soporte los navegadores más usados
  - Facilidad de uso
  - Etc..
- 



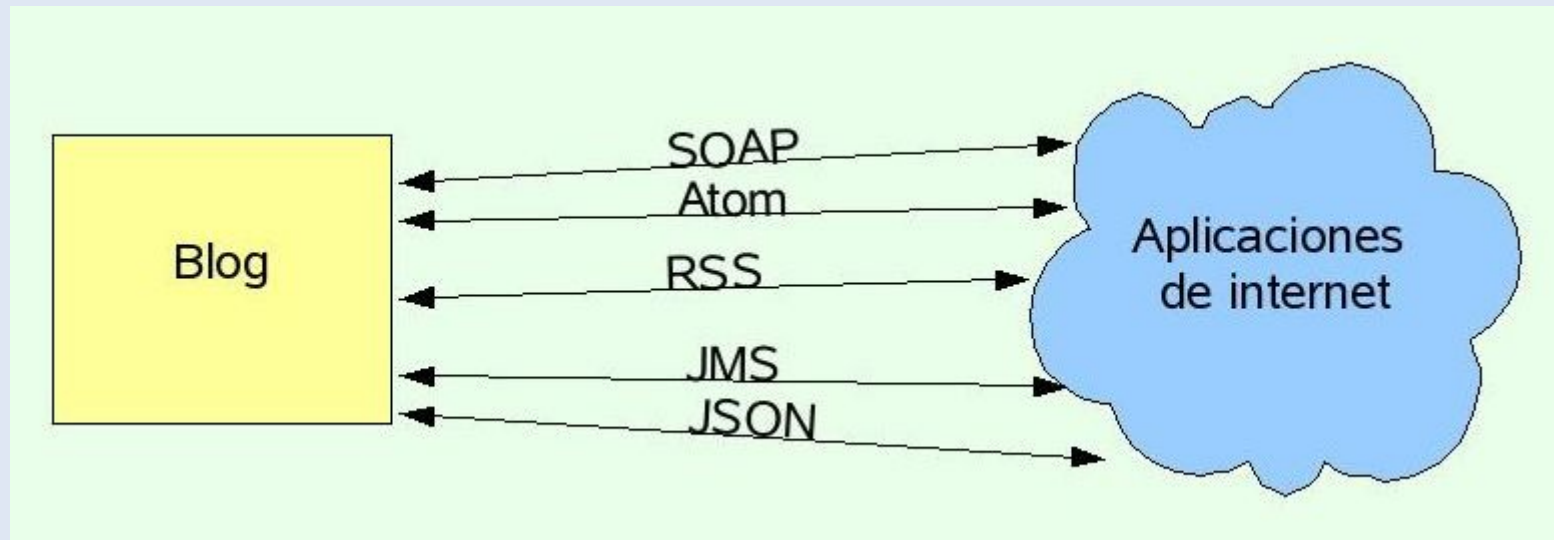
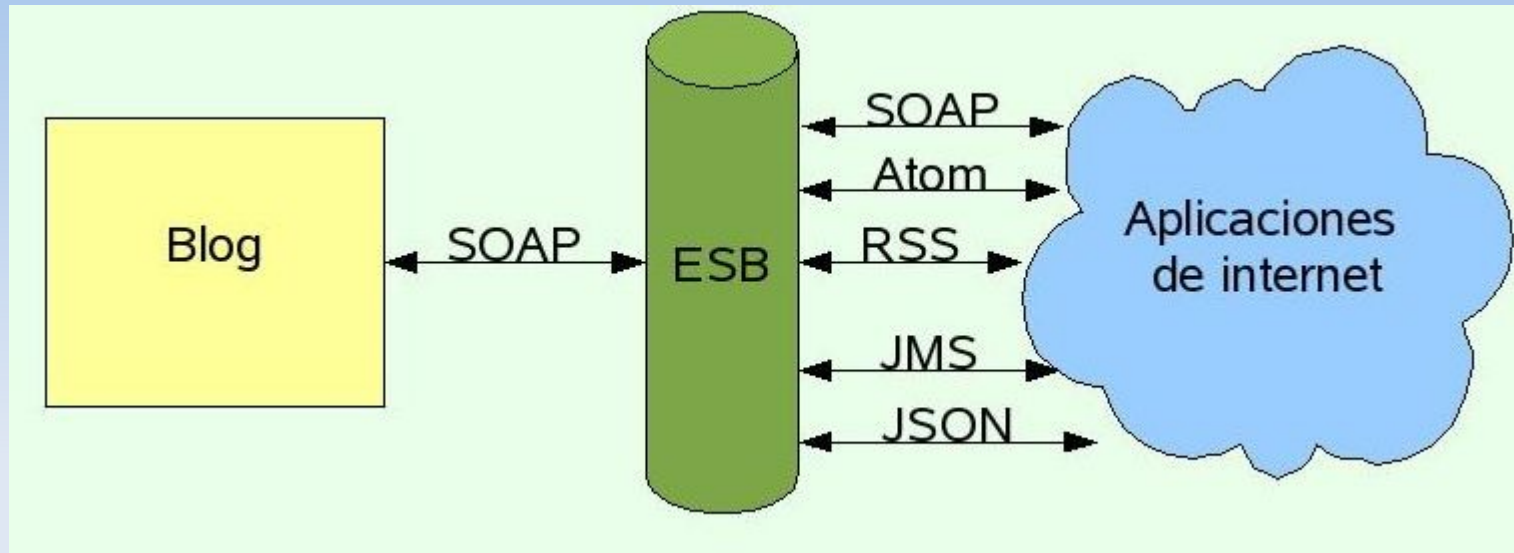
# SOA vs Web 2.0

- Mantenibilidad 
- Comunicación con otras paginas 
- Capacidad de escalar 
- Performance 
- Soporte los navegadores más usados 
- Facilidad de uso 

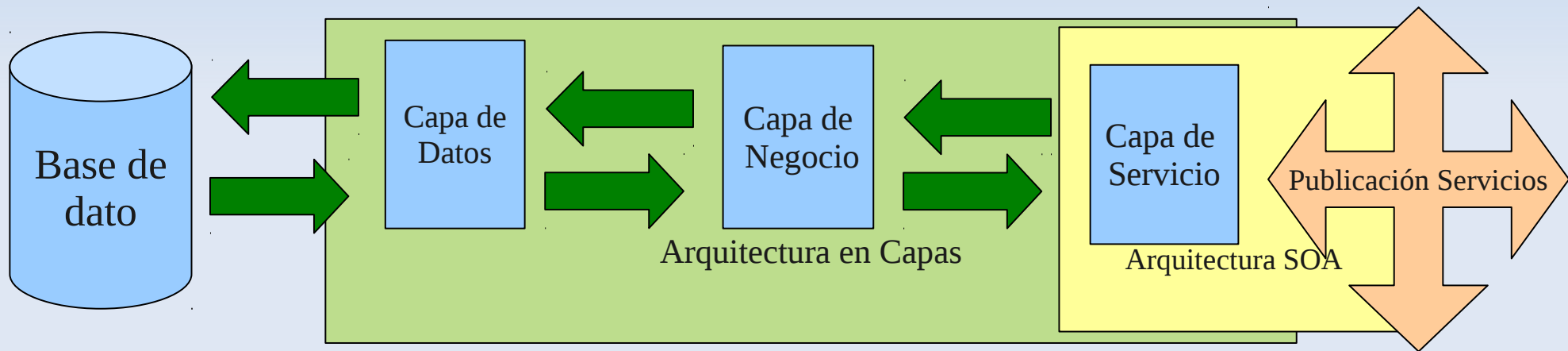
# WebBlog

- Que plataforma utilizar?
- Java!

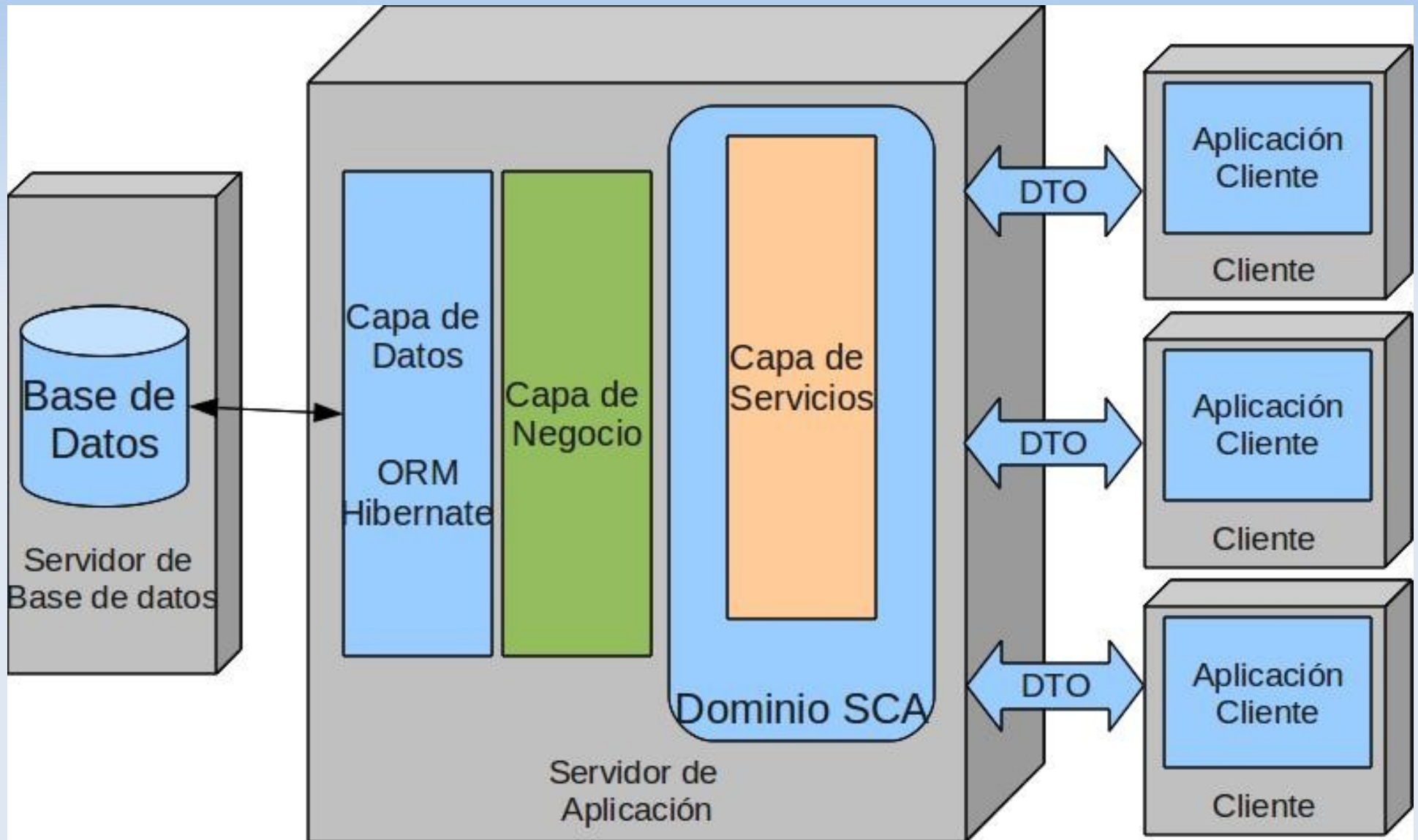
# WebBlog: ESB vs SCA



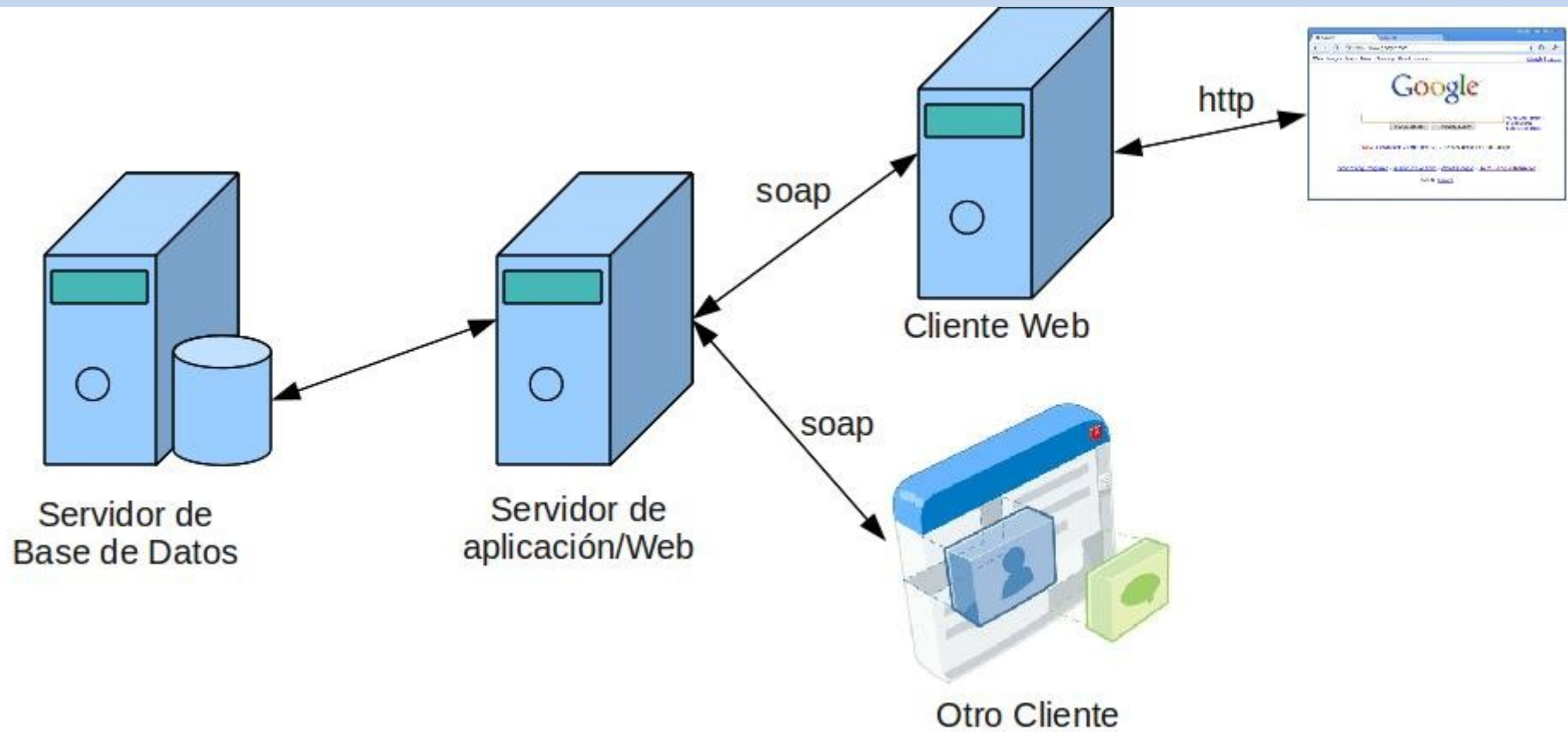
# WebBlog: Arquitectura



# WebBlog: Arquitectura

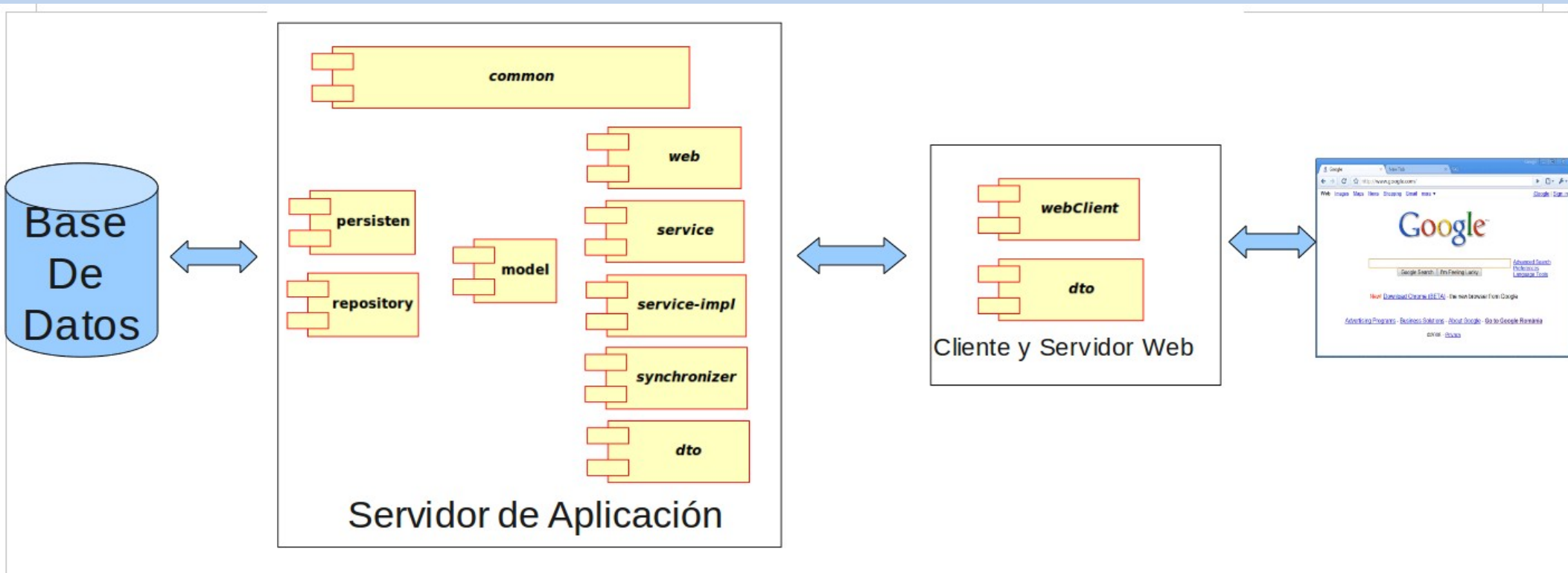


# WebBlog: Arquitectura

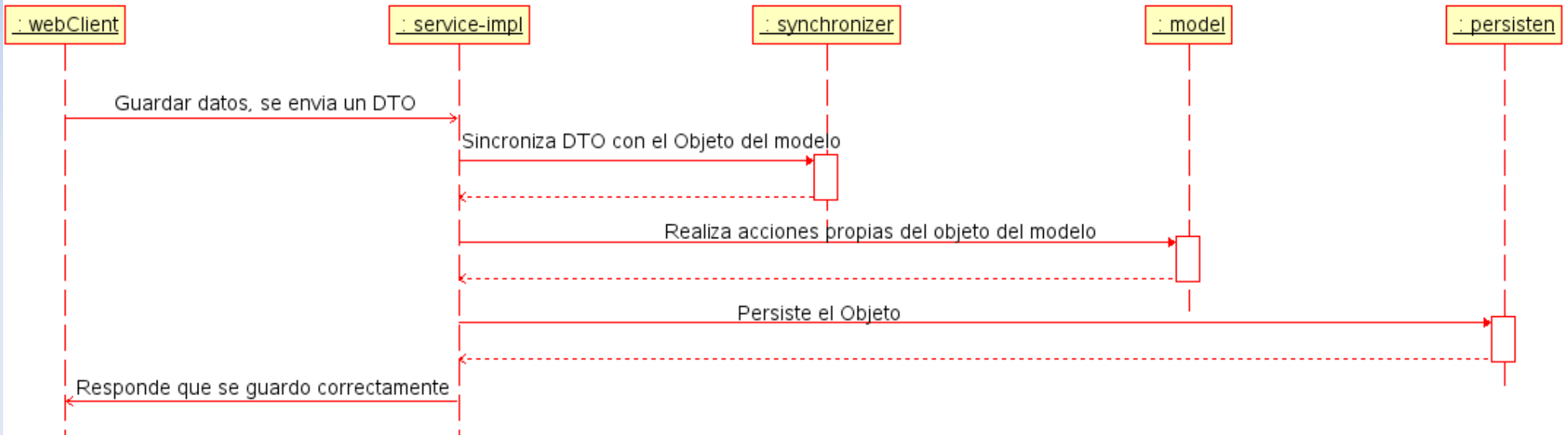




# WebBlog: Arquitectura



# WebBlog: Arquitectura



**DEMO**

# Conclusión

- Se pueden hacer aplicaciones web 2.0 con arquitectura SOA Open Source
- SOA beneficia el desarrollo Web 2.0
- Si se utiliza SOA para el desarrollo Web 2.0 es necesario analizar y medir la performance.

# Futuro: Web-oriented architecture

