

Arquitectura de Software (AS)



1. Que hace un arquitecto de software?
2. Que es AS?
3. Como definir una AS
4. Estilos de arquitectura

Que Hace un Arquitecto de Software

- ▶ Ingeniería de requerimientos
- ▶ Maneja equipos de trabajo
- ▶ Es un líder técnico
- ▶ Diseña
- ▶ Guía la implementación
- ▶ Punto de contacto con el negocio

Habilidades de un Arquitecto de Software

- ▶ Conoce el dominio del problema
- ▶ Gerencia técnica y de desarrollo
- ▶ Habilidades en diseño e implementación
- ▶ Presentación oral y escrita
- ▶ Negociación
- ▶ Liderazgo

Que es arquitectura de software ?

- ▶ “The software architecture of a program or computing system is the structure or structures of the system, which comprise the software elements, the externally visible properties of those elements, and the relationships among them.”

Len Bass

Que es arquitectura de software ?

- ▶ “Effective Software Architects understand not only technology but also the business domain of a problem space. Without business domain knowledge, it is difficult to understand the business problem, goals, and requirements, and therefore difficult to design an effective architecture to meet the requirements of the business”

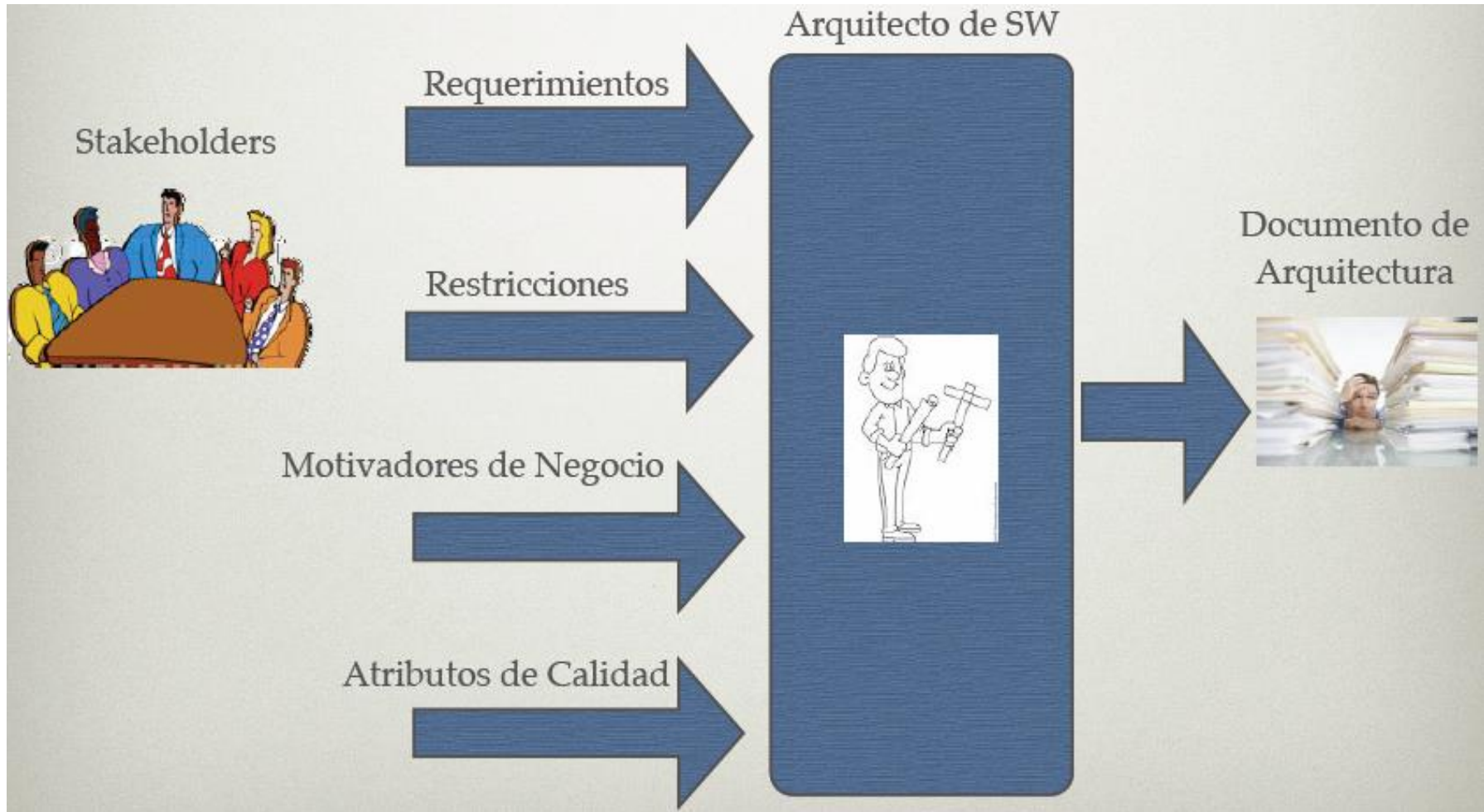
Mark Richards

Que es arquitectura de software ?

- ▶ “In the context of business enterprise applications development, an architect must act as a bridge between the business and technology communities of an organization, representing the interest of each party to the other, often mediating between the two, but allowing the business to drive.”

Dave Muirhead

Que es arquitectura de software ?



Como definir una buena AS

1. Aspectos

✓ Comprensibilidad.

- La arquitectura de software simplifica nuestra capacidad para comprender grandes sistemas de software,

✓ Reuso.

- Descripciones arquitectónicas contribuyen al reuso en múltiples niveles.
 - Actualmente se utilizan en librerías de componentes

Como definir una buena AS

Aspectos

✓ Construcción.

- Una descripción arquitectónica ofrece un blueprint para desarrollo indicando los principales componentes y dependencias entre ellos

✓ Evolución.

- Las arquitecturas de software pueden exponerlas dimensiones sobre las cuales se espera se desarrolle.

✓ Análisis.

- Las descripciones arquitectónicas ofrecen nuevas oportunidades para el análisis incluyendo la validación de consistencia, cumplimiento de limitaciones impuestas por el estilo arquitectónico

Como definir una buena AS

1. Aspectos

✓ Administración.

- La experiencia ha mostrado que los proyectos exitosos ven en el mejoramiento de una arquitectura de software la clave para el desarrollo de software a nivel industrial.

Metodología para iniciar la selección de una AS básica

1. Desarrollar un modelo mental de la aplicación a un nivel alto.

- Como si fuera una aplicación pequeña...

... “ejemplo aplicación de finanzas personales funciona al recibir o pagar dinero, en cualquier orden, controlado mediante una interfaz de usuario”.

Metodología para iniciar la selección de una AS básica

2. Desglosar en las componentes requeridas.
 - Buscar una cohesión alta y un acoplamiento bajo.
 - Ejemplo: aplicación de finanzas personales se desglosa en activos, proveedores e interfaz
3. Repetir este proceso para los componentes.

Metas de diseño

➤ Extensión

- Facilitar la adición de nuevas características

➤ Cambio

- Facilitar los cambios de requerimientos

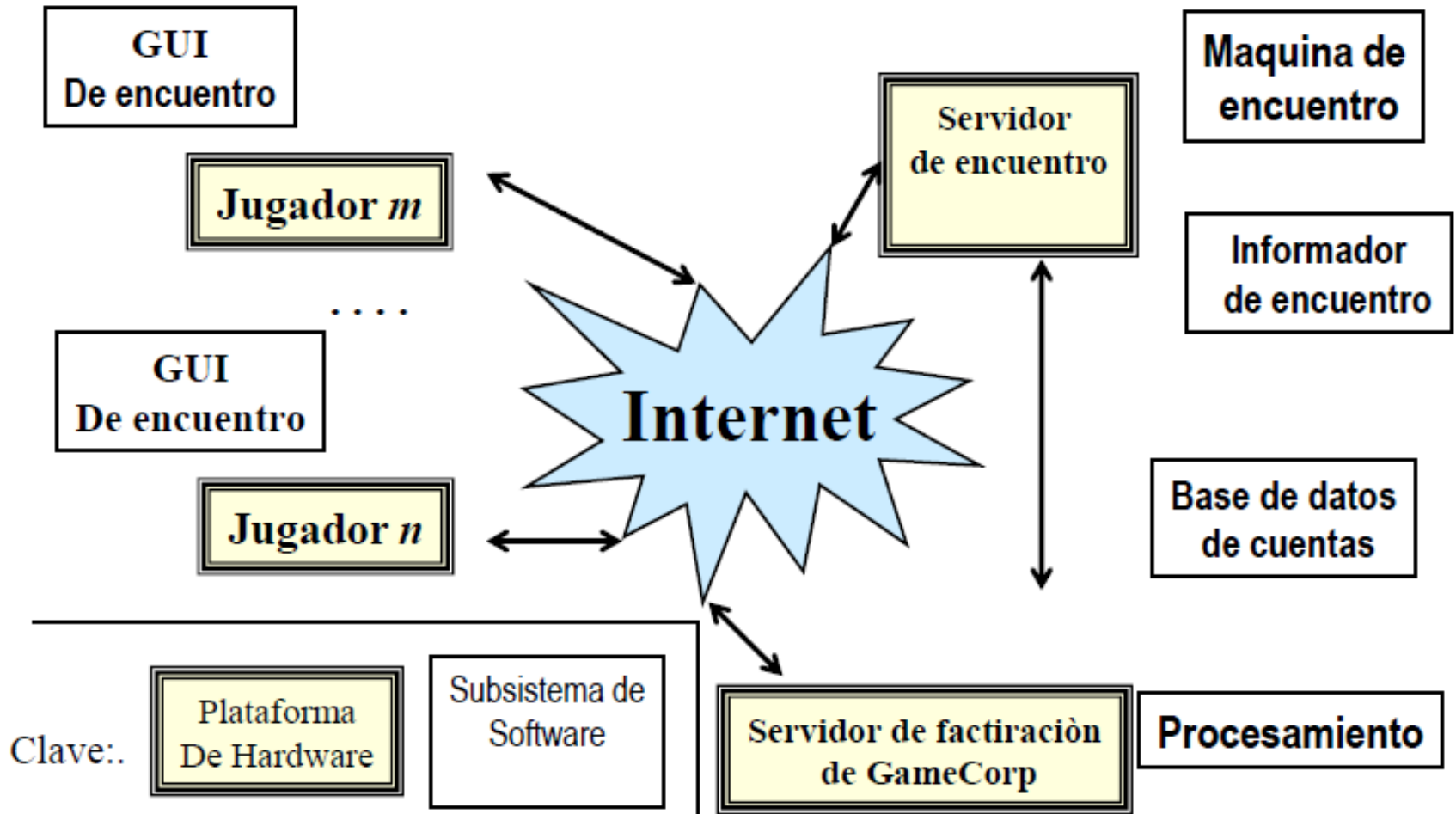
➤ Sencillez

- Hacerlo de fácil comprensión
- Hacerlo de fácil implementación

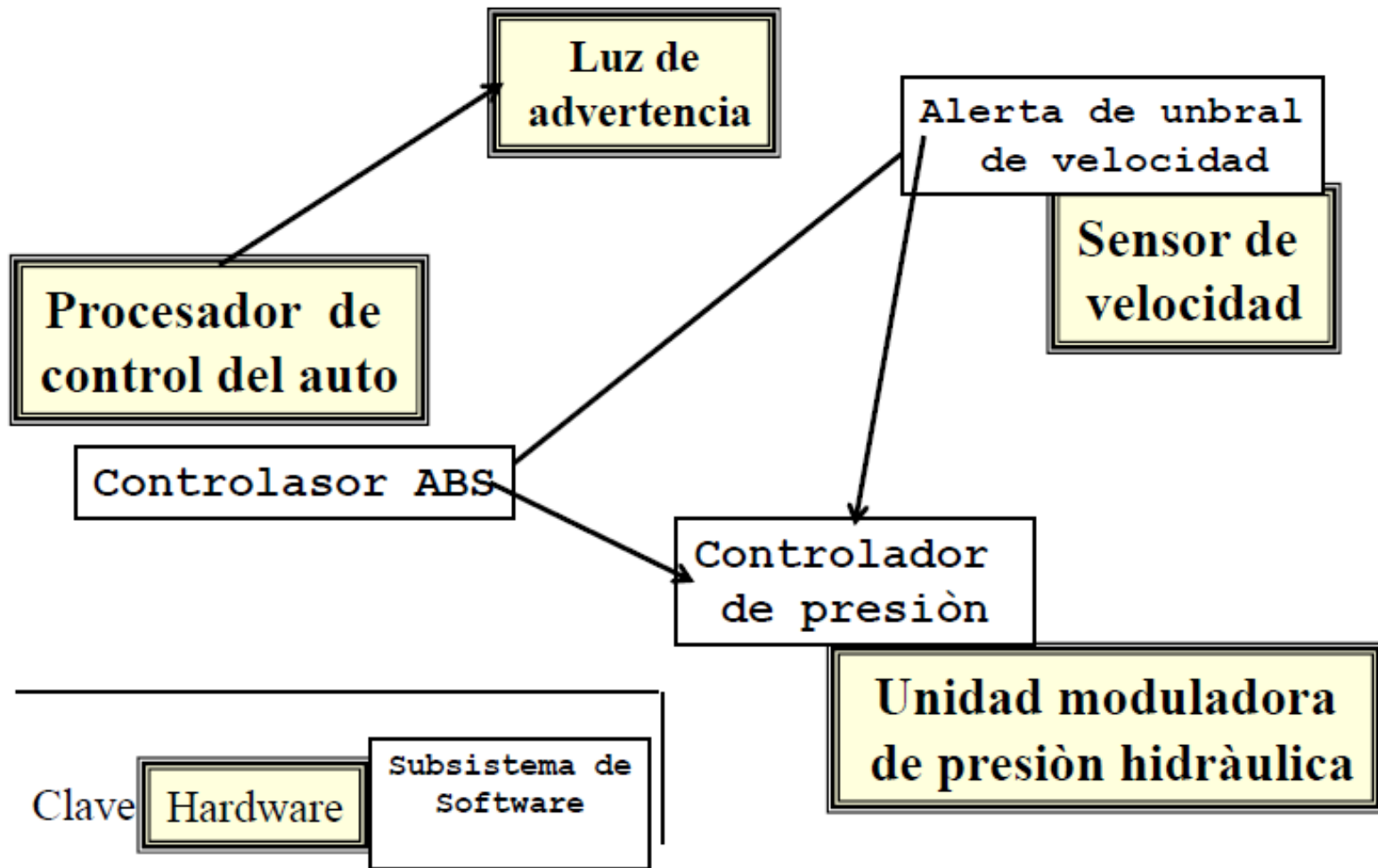
➤ Eficiencia

- Lograr alta velocidad: ejecución y/o compilación
- Lograr un tamaño pequeño: tiempo en ejecutar y/o tamaño de código

Ejemplo 1: Juego basado en internet



Ejemplo 1: Frenos de seguridad



Una propuesta metodológica

Preparación

- Análisis
 - Casos de Uso
 - Mundo del Problema
 - Requerimientos No funcionales
- **Arquitectura de software**

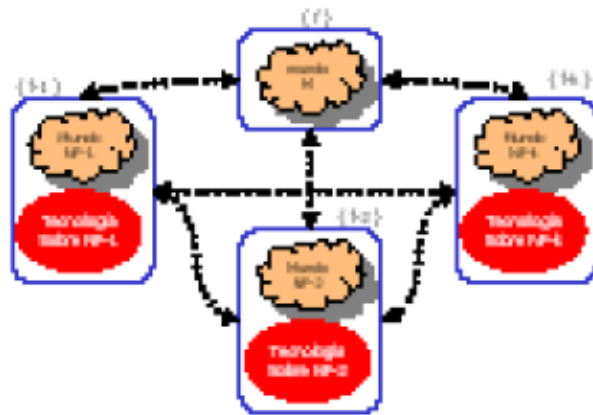
Una propuesta metodológica

Aproximación funcional

- Los requerimientos funcionales son la guía para construir el modelo
 - Qué elementos aparecen referenciados en los RF y como deben interactuar entre ellos?
- Se corre riesgo de definir arquitectura guiada por el problema
- Se construye diagramas de clase y diagrama de secuencia a la vez

Una propuesta metodológica

Definir arquitectura



Transformación
inicial



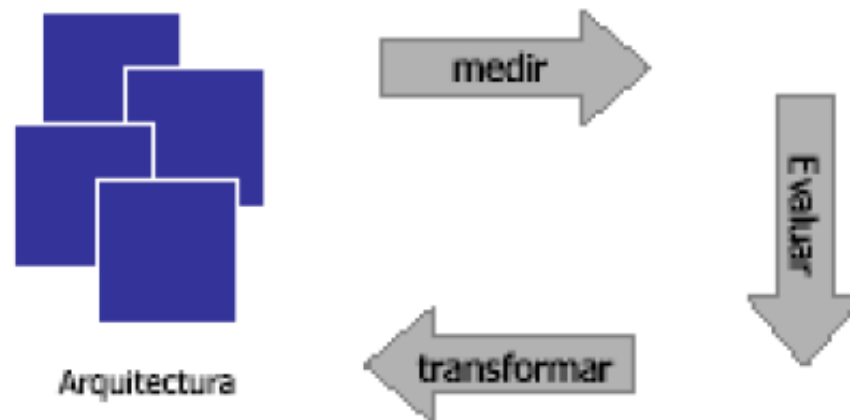
Primer borrador
de la
arquitectura

- Conjunto de cajas grises
- Debe tener en cuenta RNF, MN y RF
- Debe ser completa
- Se definen sobre este borrador los principales “protocolos”

Una propuesta metodológica

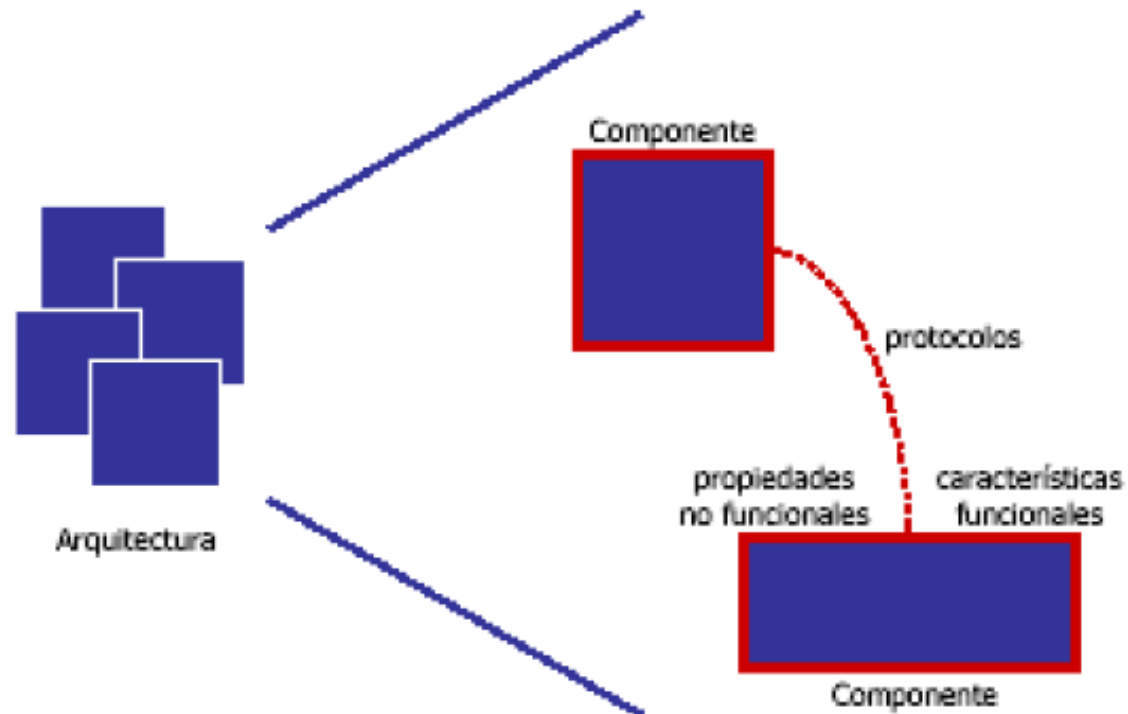
Refinar Arquitectura

- Ciclo de medición y refinación
- Aunque sea abstracta se debe poder explicar



Una propuesta metodológica

Elementos de la Arquitectura



Estilos de AS

- Que es la Arquitectura ?
 - Define las estructuras de alto nivel del sistema, en términos de:
 - Componentes de software
 - Las propiedades visibles de estos componentes (comportamiento y características)
 - Relaciones entre ellos (conectores, protocolos, restricciones,...)

Estilos de AS

Estilos de arquitectura de software (Shaw & Garlan)

- Arquitecturas de flujo de datos

- Batch sequential
- Pipes and Filters

- Componentes independientes

- Procesos de comunicación paralela
- Sistemas Cliente-servidor
- Sistemas basados en eventos

- Maquinas virtuales

- Interpretadores
- Sistemas basados en reglas

- Arquitecturas de repositorio

- Bases de datos
- Sistemas hipertexto
- Blackboards

- Arquitecturas de capas

Estilos de AS

Estilos de Flujo de Datos

El sistema tubería-filtros se percibe como una serie de transformaciones sobre sucesivas piezas de los datos de entrada. Los datos entran al sistema y fluyen a través de los componentes.



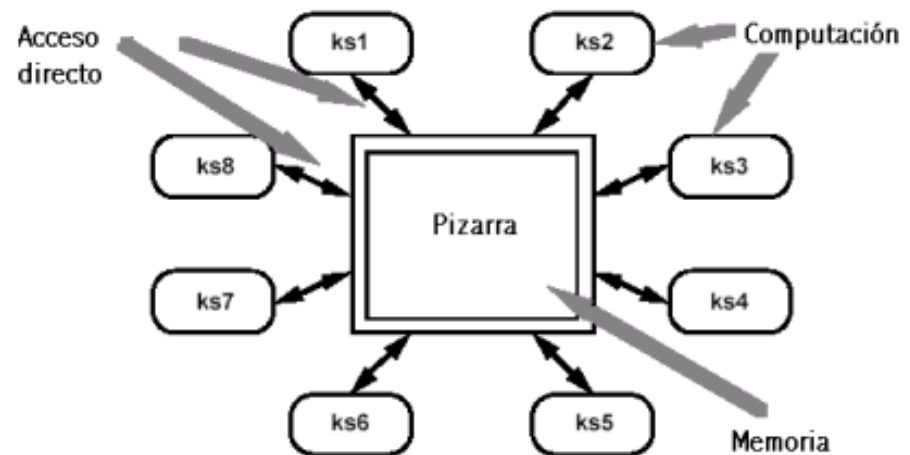
Compiladores, OS, SAX (XML), workflows, entre otros.

Estilos de AS

Estilos Centrados en Datos

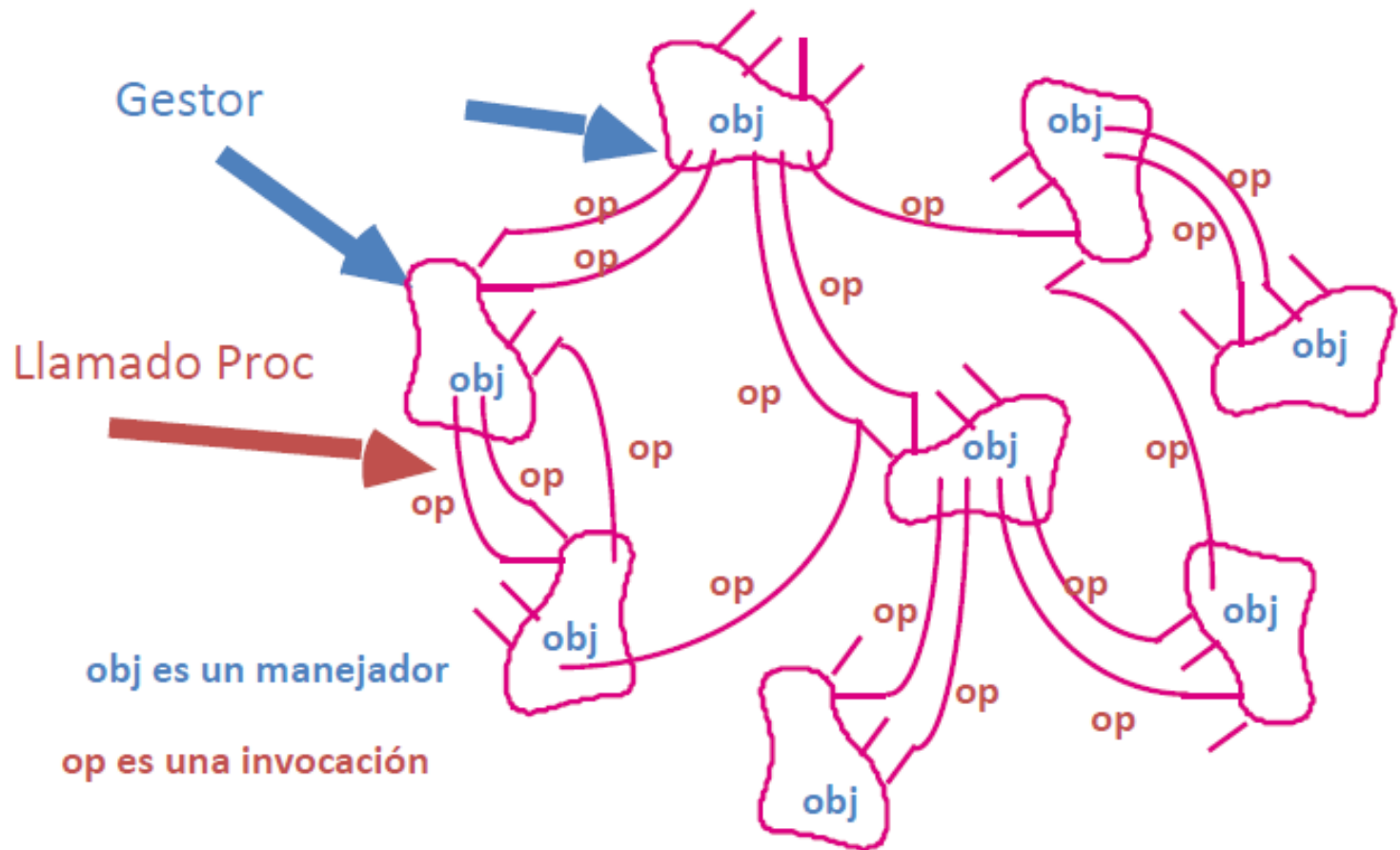
En la Arquitectura de Pizarra o Repositorio hay dos componentes principales: una estructura de datos que representa el estado actual y una colección de componentes independientes que operan sobre él (Mary Shaw y David Garlan).

Procesos en lotes BDs
Ambientes de Programación
Motores de búsqueda
Entre otros.



Estilos de AS

Llamado-y-Retorno: Orientado a objetos



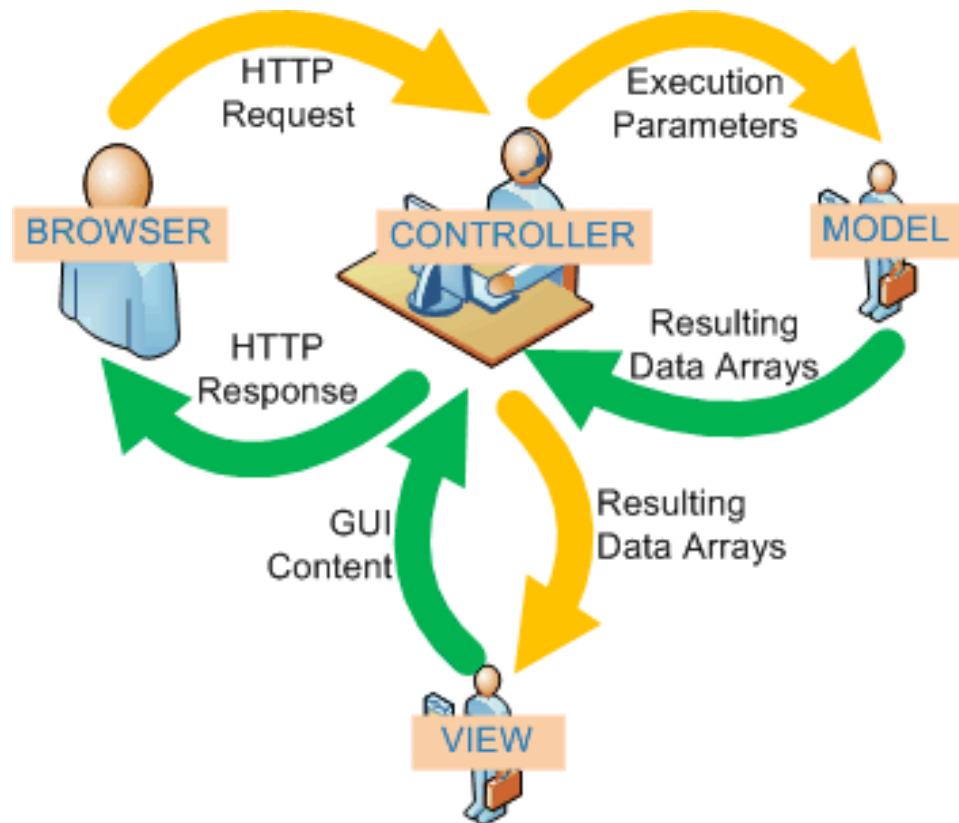
Estilos de AS

Sistemas Llamada retorno

- Objetos
 - Encapsular representaciones
 - Proveer interfaces de acceso a servicios
- Conectores
 - Llamado-retorno; servicio de invocación
- Razonamiento
 - La corrección de un componente depende de la corrección de servicios que invoca .

Estilos de AS

- ▶ Sistema Llamado y Retorno
 - Model-View-Controller (MVC)



Estilos de AS

Cliente Servidor 3-Capas

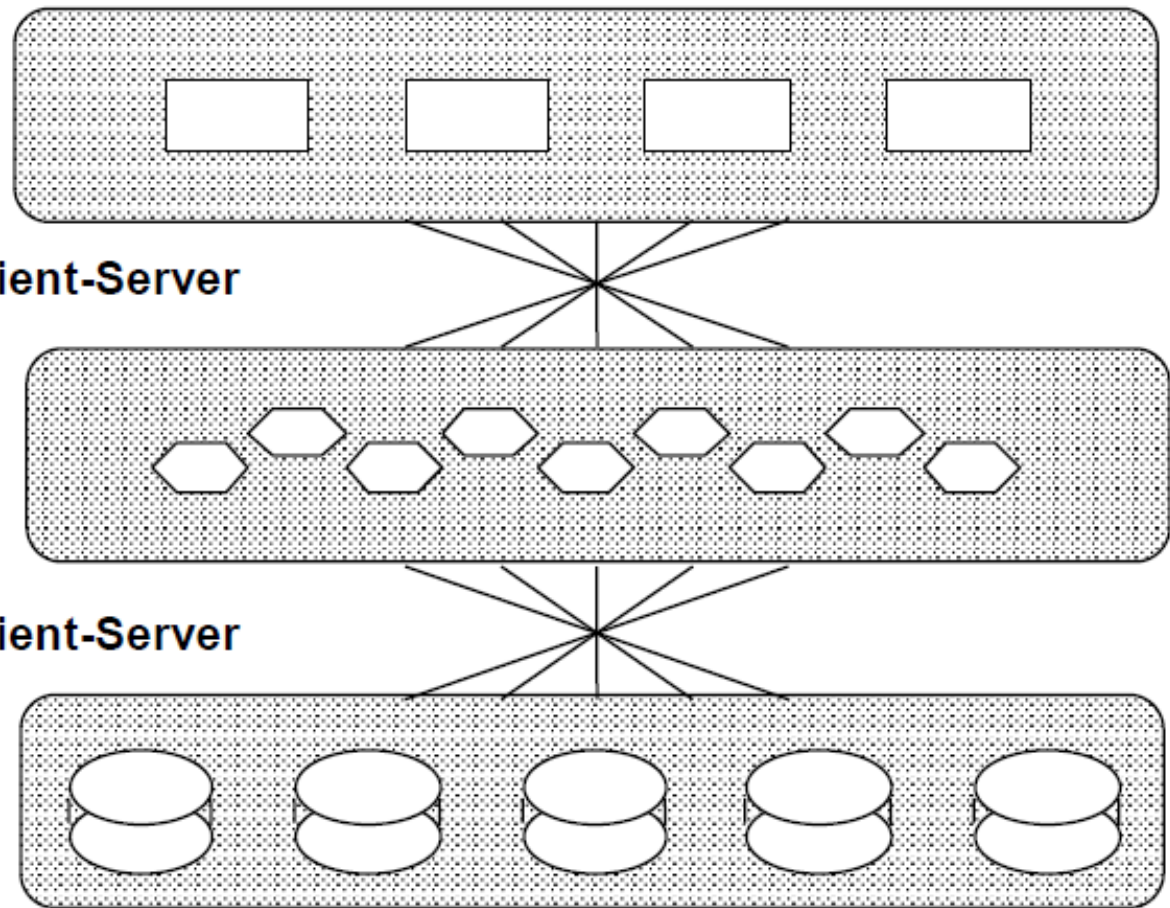
Interfaz
Usuarios

Client-Server

Lógica de
negocio

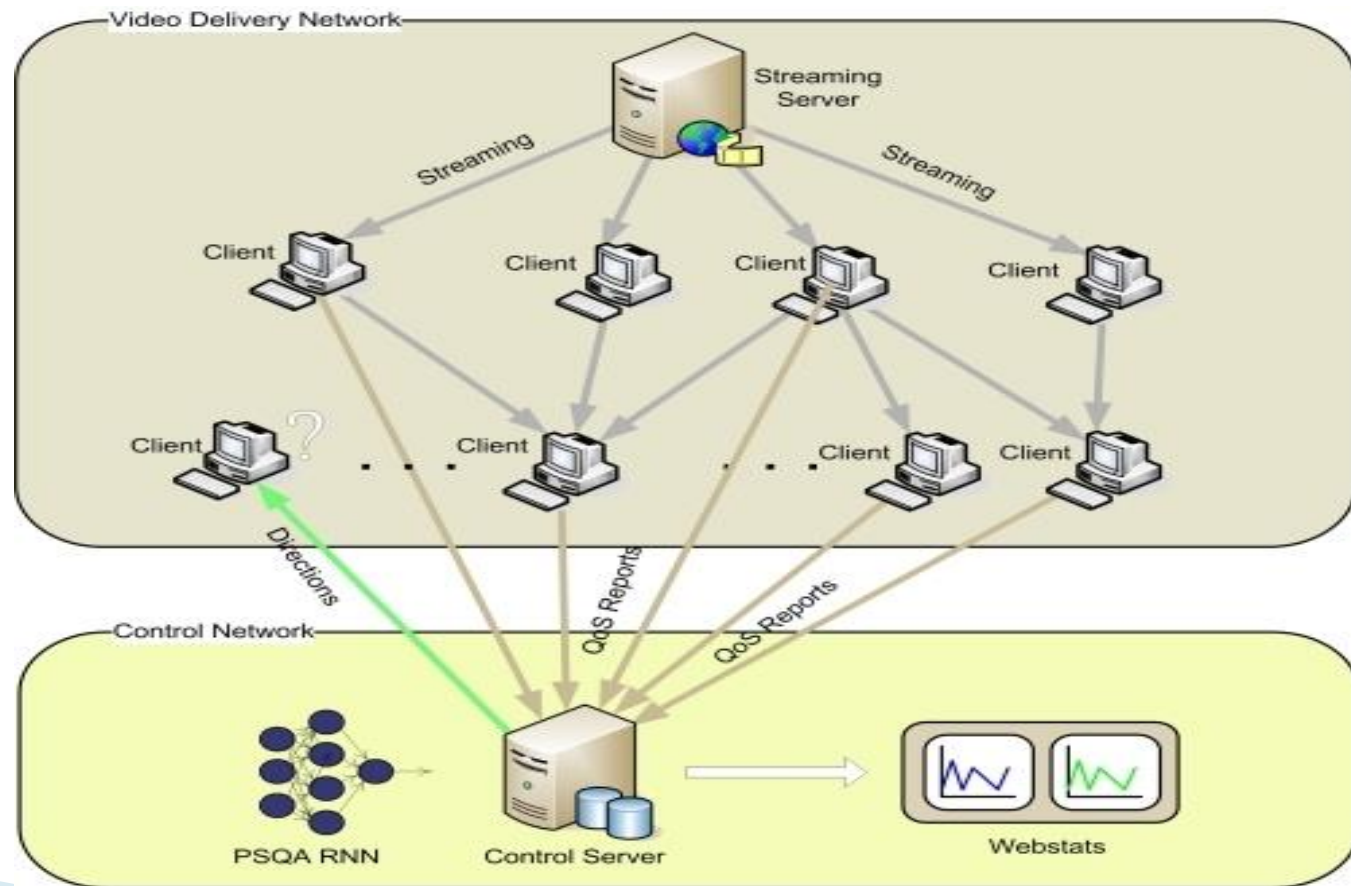
Client-Server

Persistencia



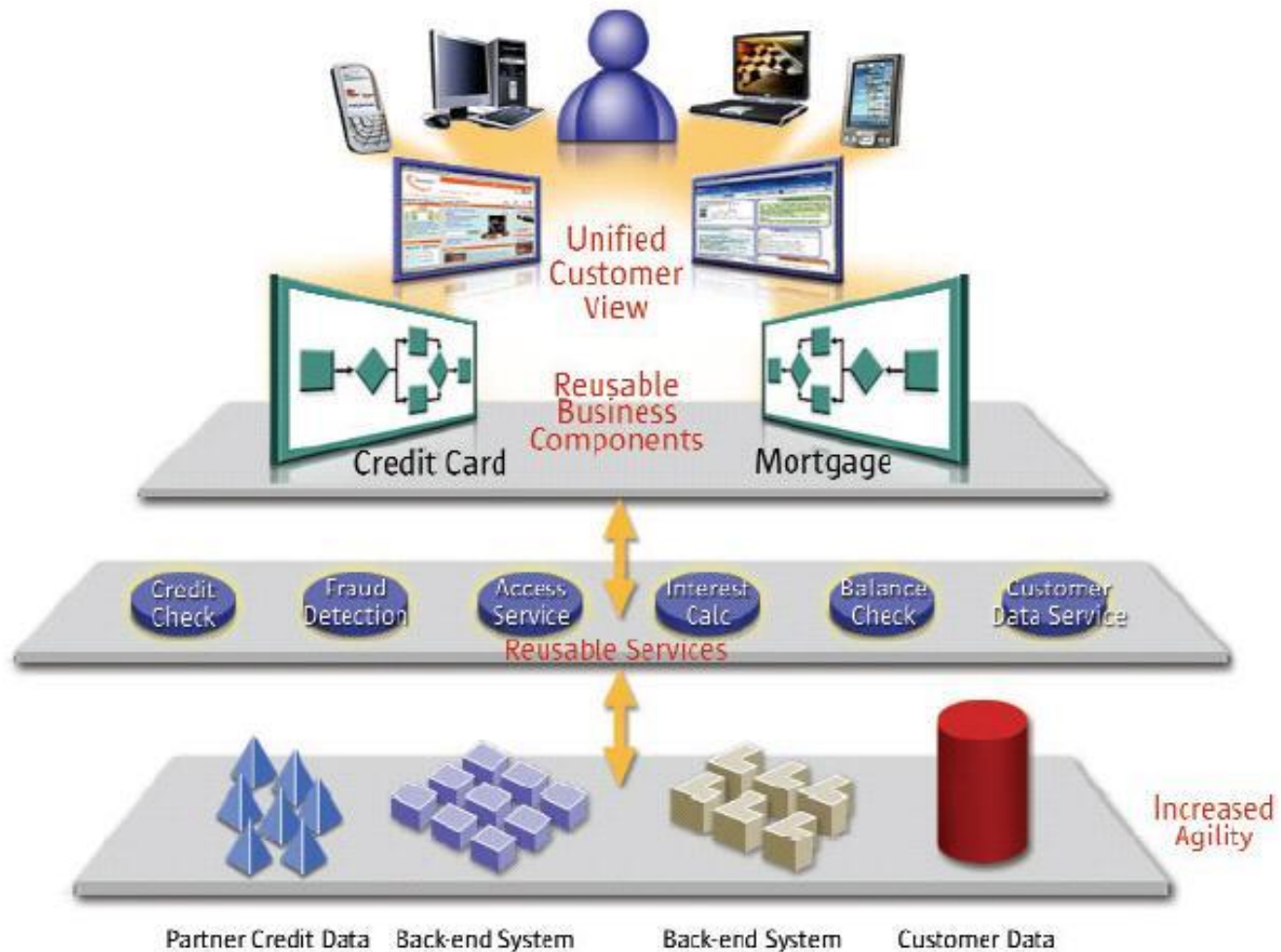
Estilos de AS

► Estilos P2P



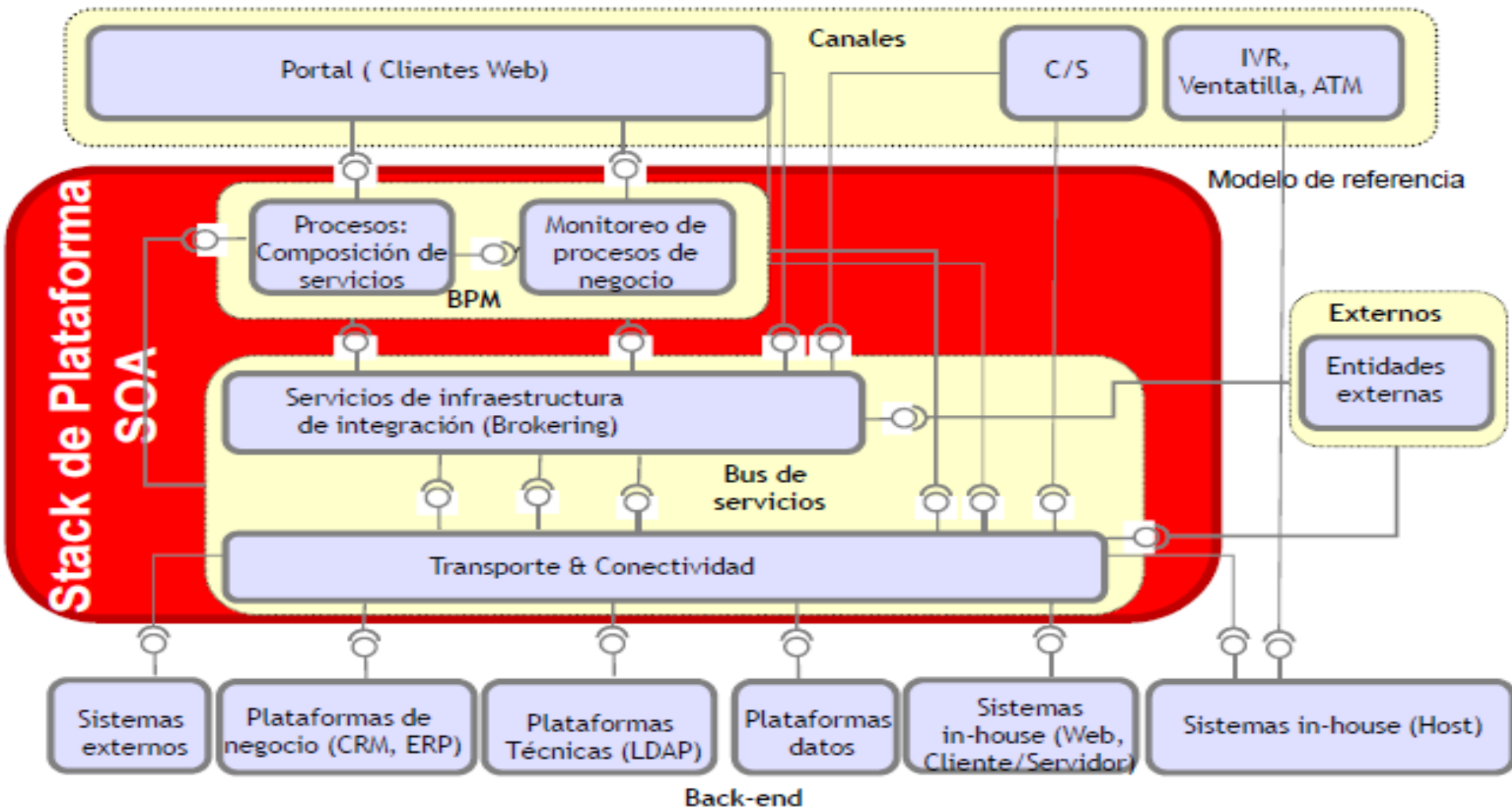
Estilos de AS

► Estilos P2P – SOA



Estilos de AS

► Estilos P2P – SOA



Estilos de AS

Otros

- **A menudo los estilos se usan en combinación**
 - Ejemplo: Cada capa puede tener un estilo diferente internamente
 - Ejemplo: un componente puede tener una subestructura definida en un diferente estilo al que la rodea
- **Muchos estilos se atan de cerca a los dominios específicos -- a menudo especializando un estilo más genérico**
 - Algunas veces llamados **frameworks de integración de componentes**
 - Sistemas de N-capas
 - Pila protocolo OSI
 - Sistemas de instrumentación
- **En muchos casos éstos se especializan a una familia de producto particular,**
 - Algunas veces llamados **arquitectura de línea de producto**

SII – Preguntas ?



SII – Preguntas ?



SII – Preguntas ?

