

# Ingeniería de Software II

## Segundo Cuatrimestre de 2016

Clase 15: Introducción a las arquitecturas de software. Estilos arquitectónicos

Buenos Aires, 20 de Octubre de 2016

## **Analizando dibujitos...**

# File System de Google

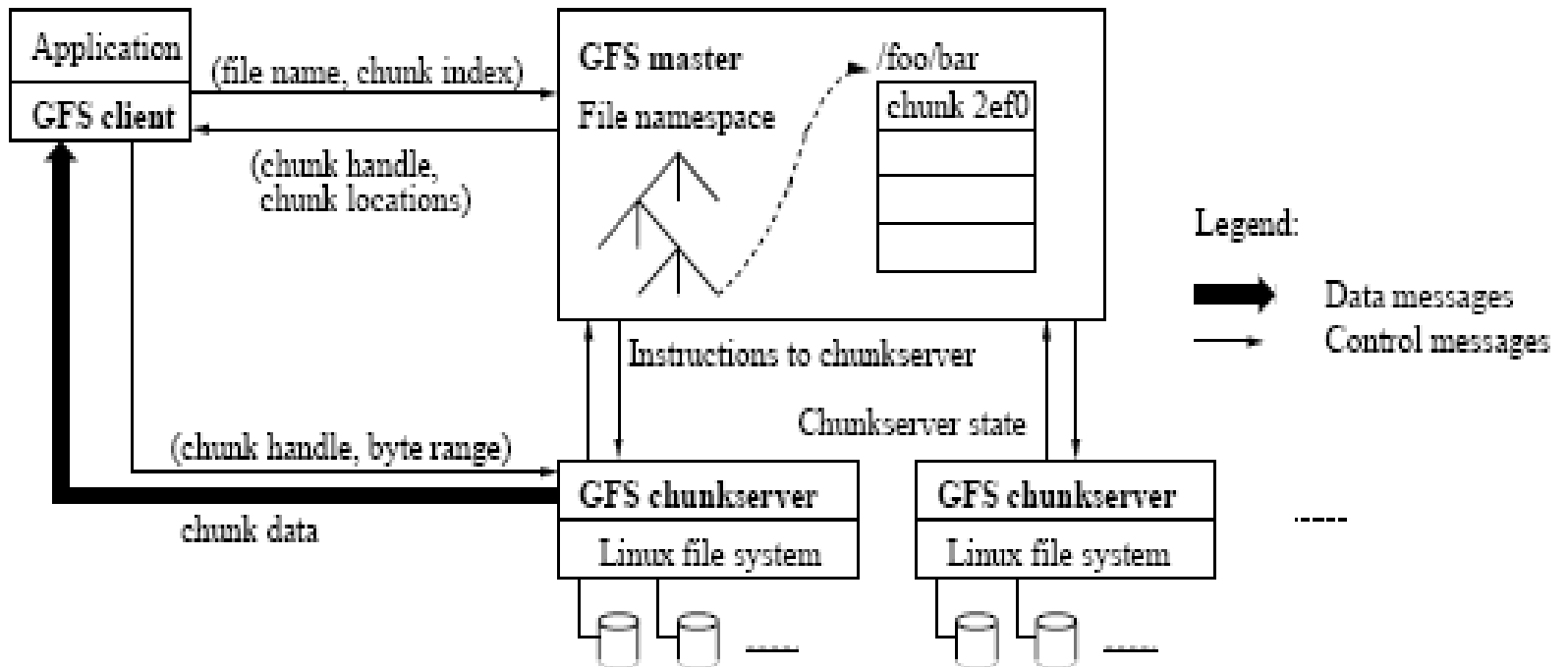
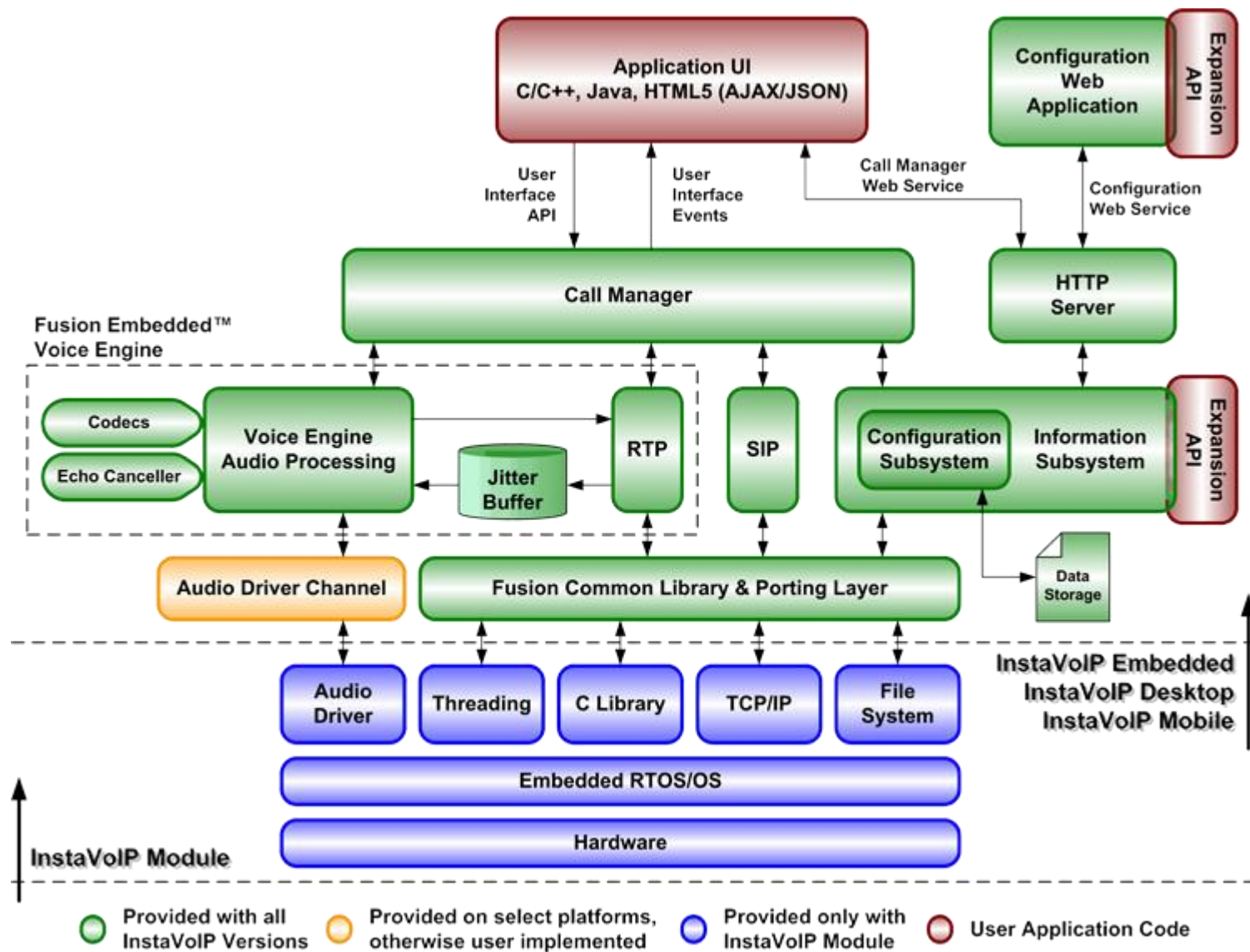


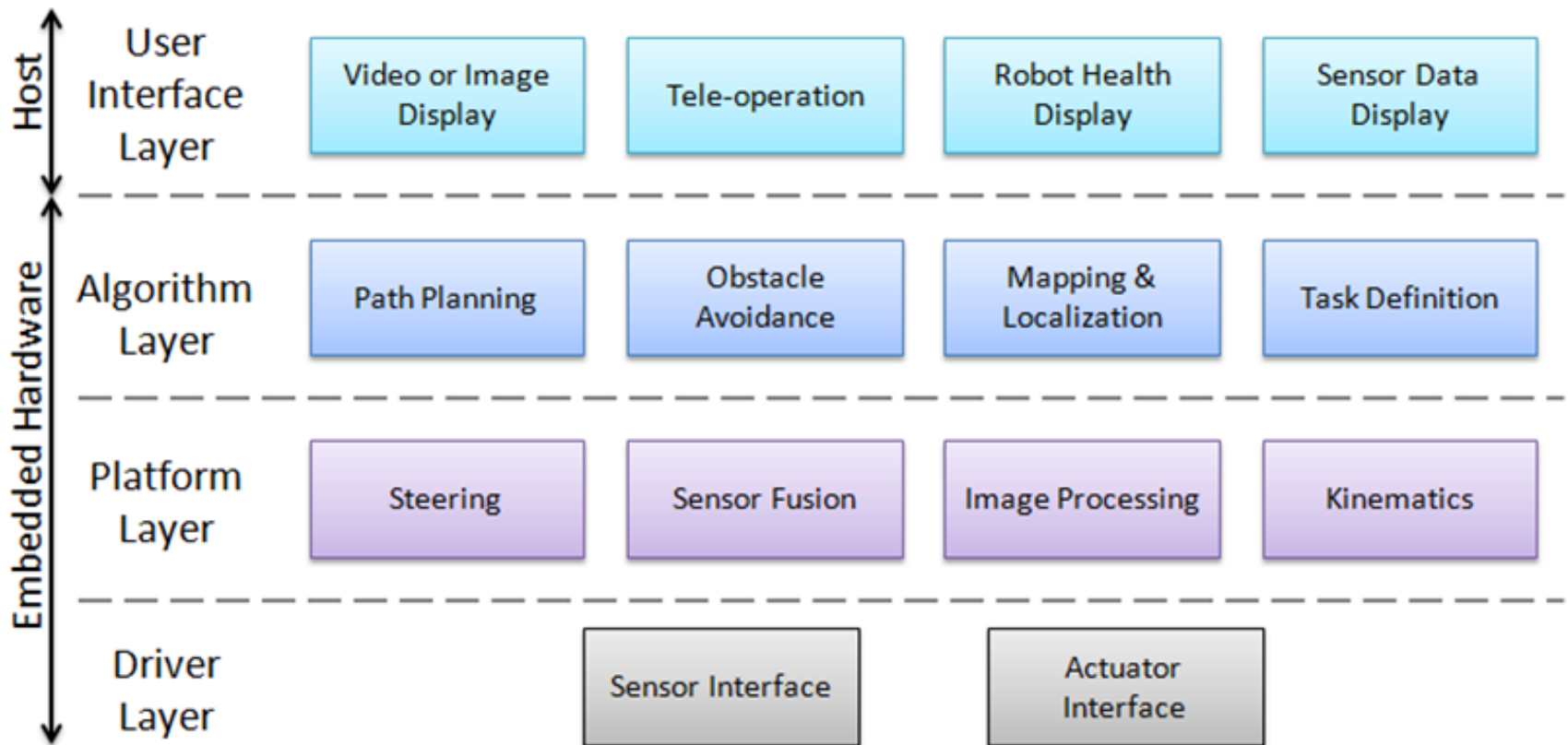
Figure 1: GFS Architecture

Source: "The Google File System" Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung

# InstaVoIP



# Referencia para?



# Robot Experimental

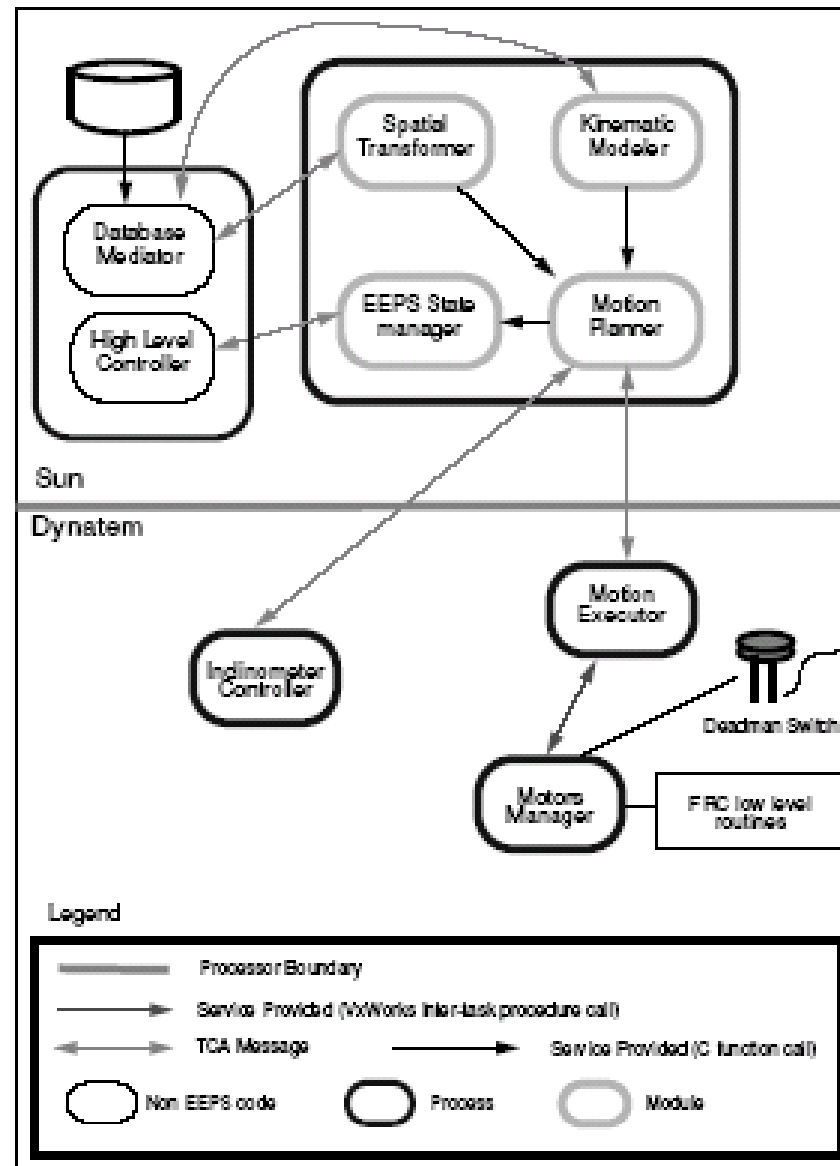
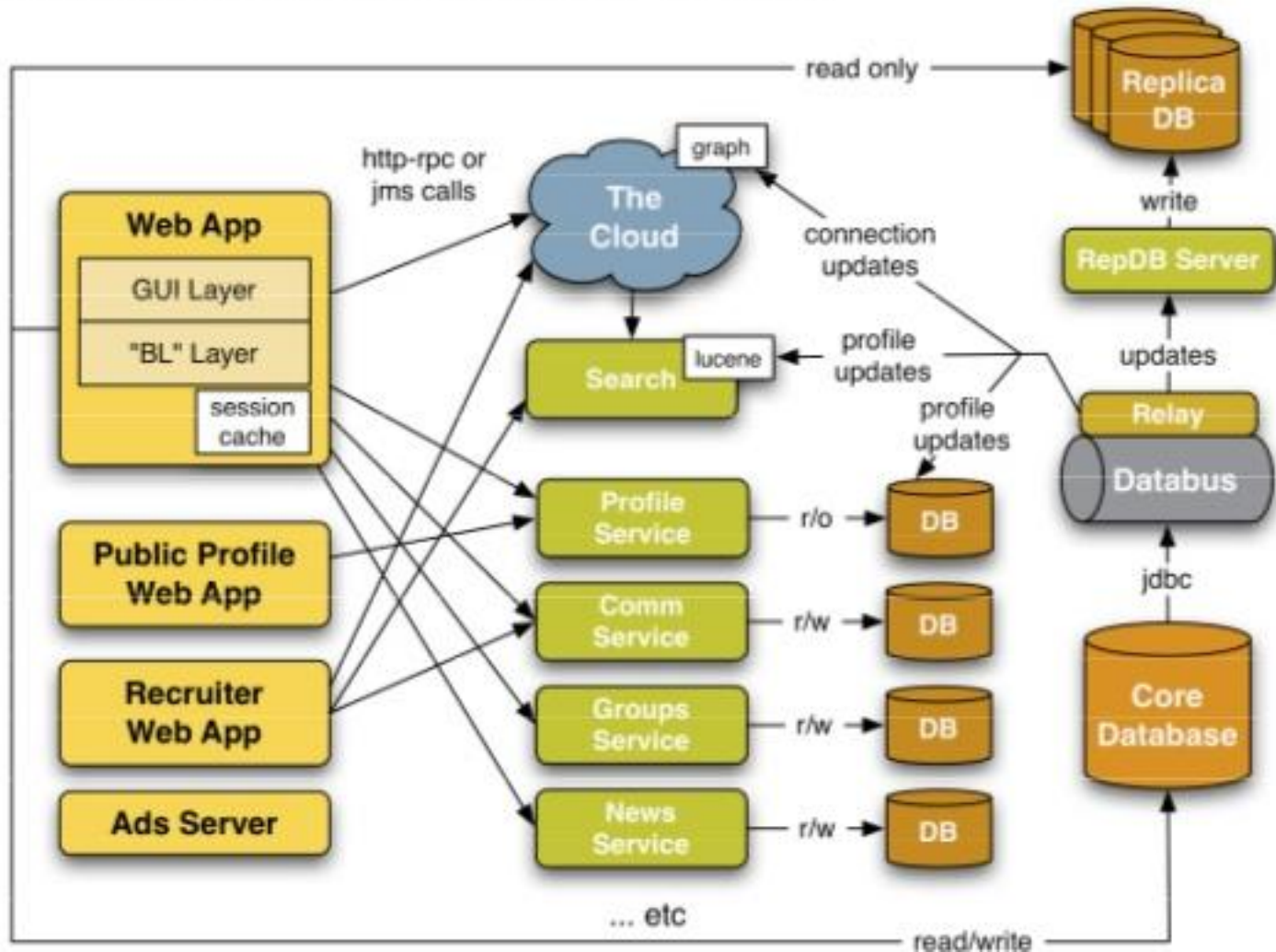
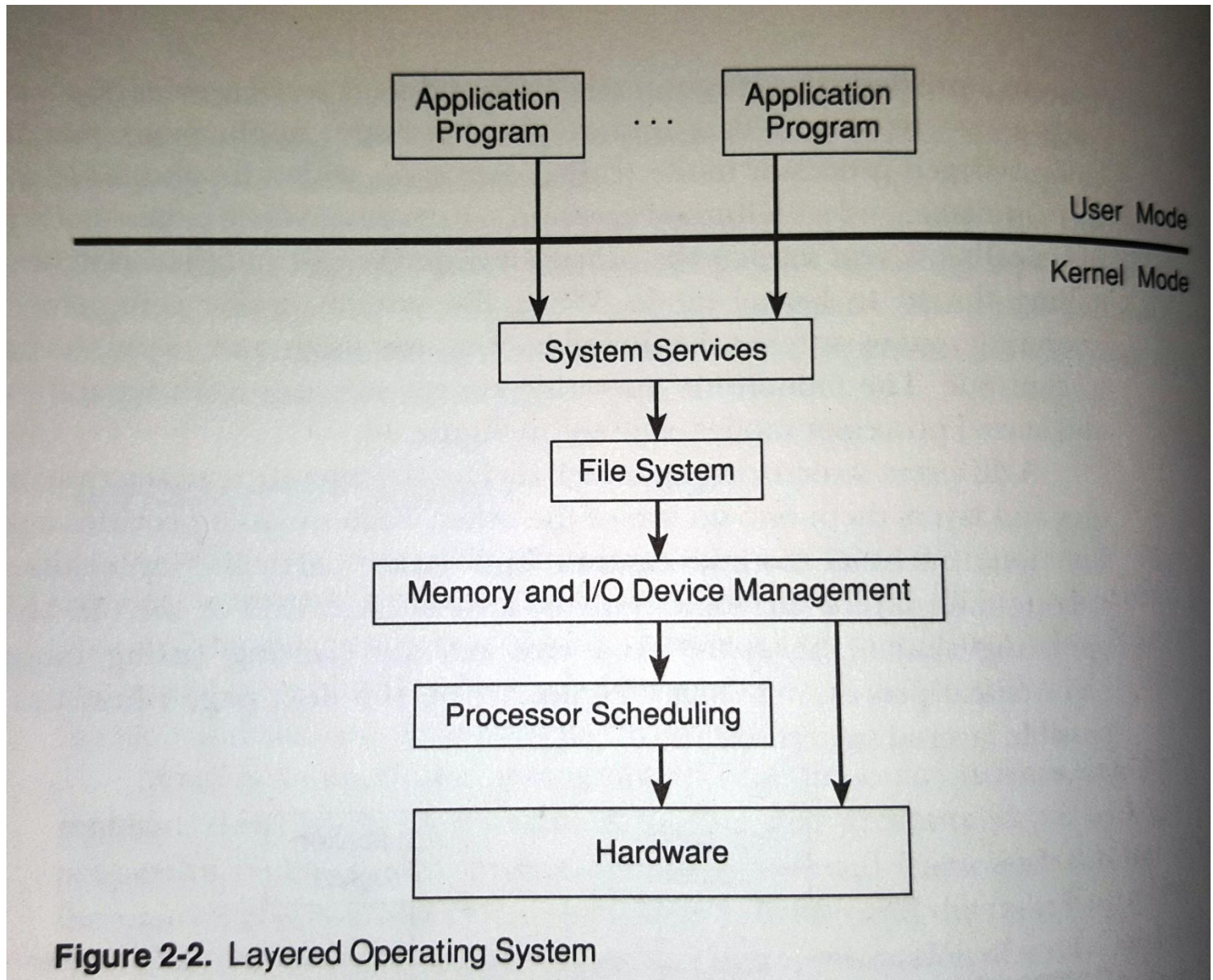


Figure 3-2: Allocation of Modules to Processes

## LinkedIn Architecture: Today

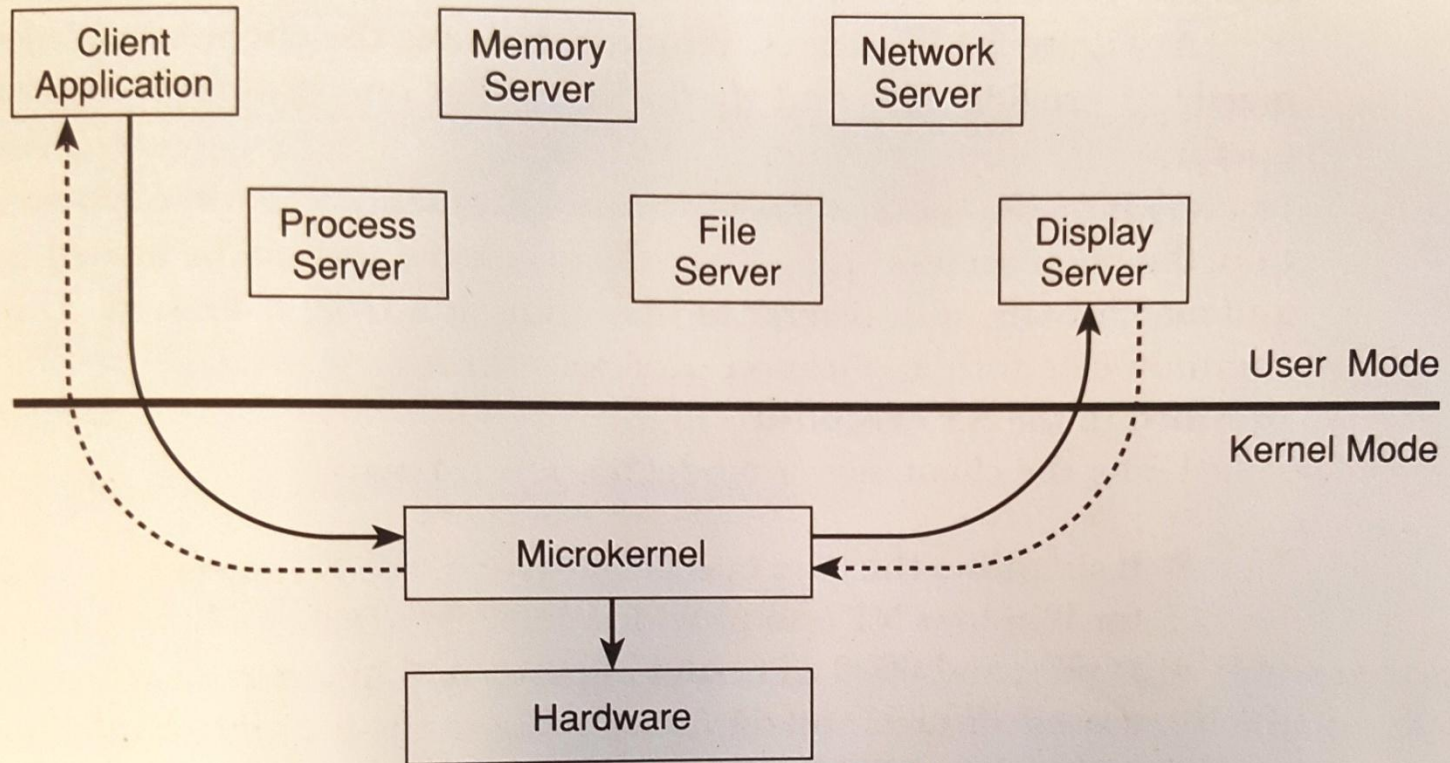


# OS - 1



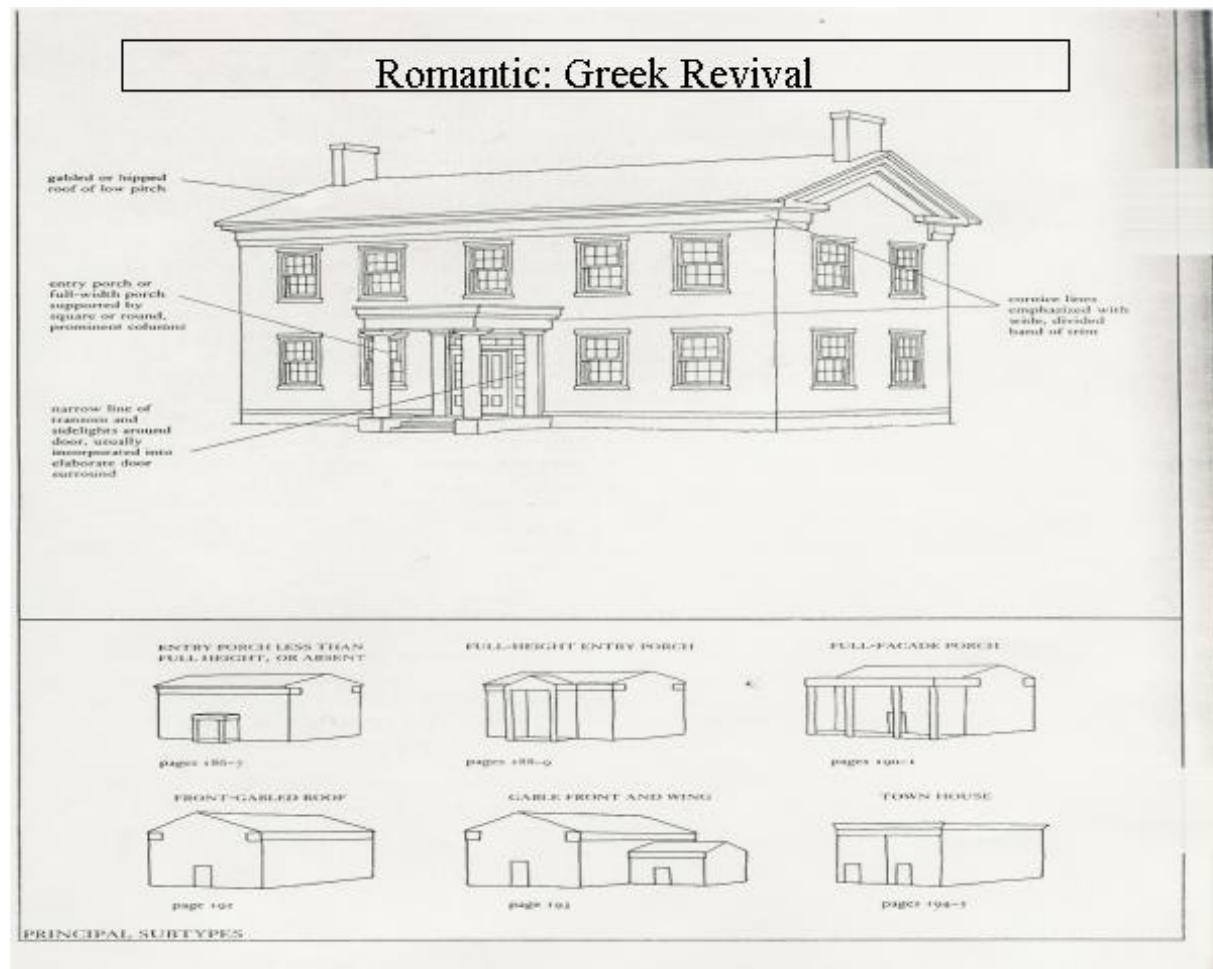


## OS - 2



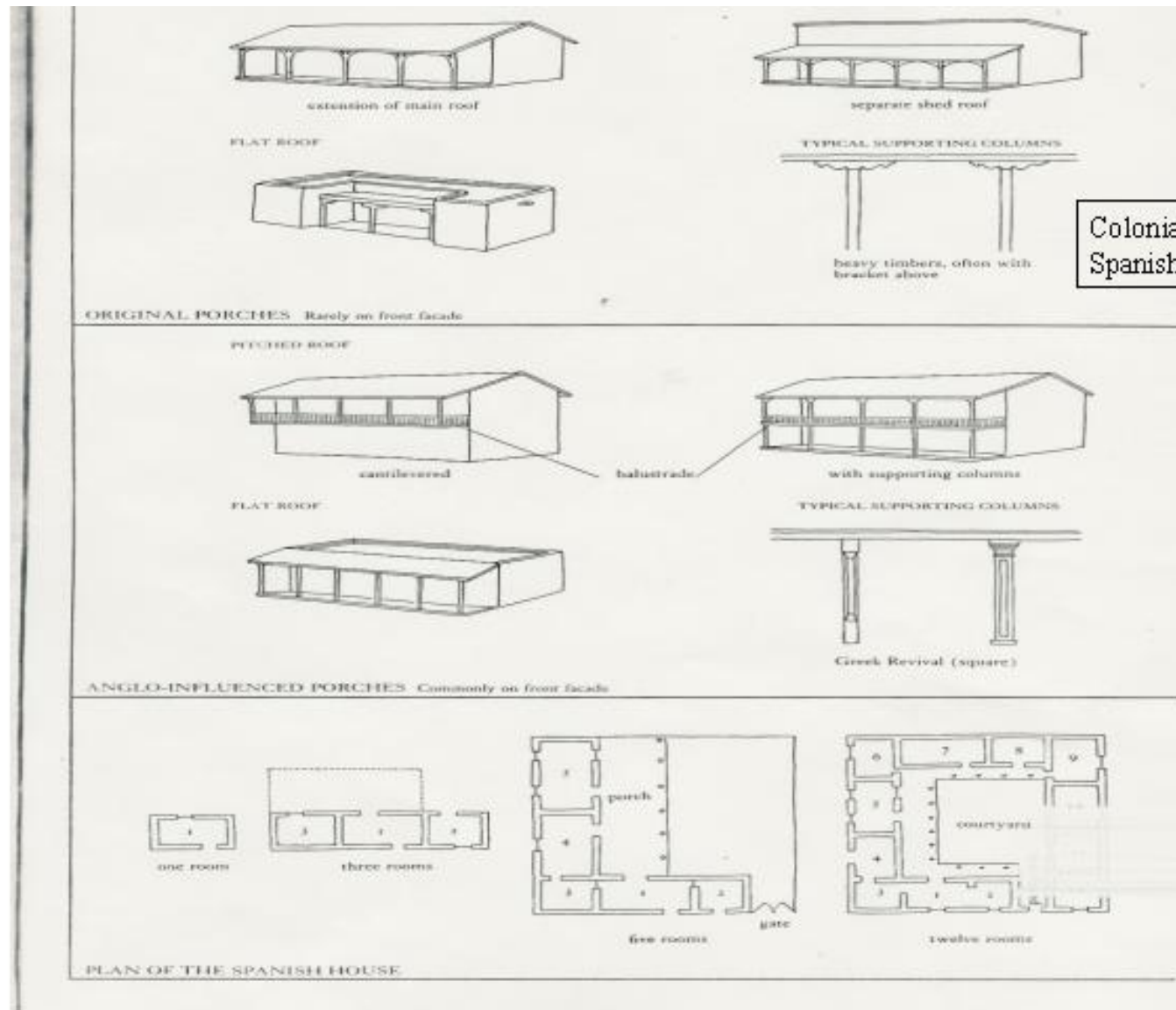
**Figure 2-3.** Client/Server Operating System

# Algunas analogías



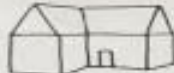

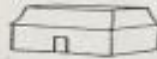


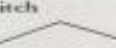
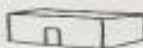





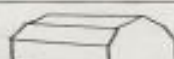





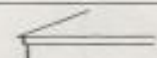
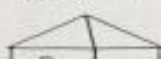

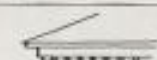

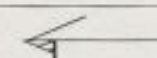


Fuente: Virginia McAlester. A Field Guide to American Houses.

# Algunas analogías (cont.)

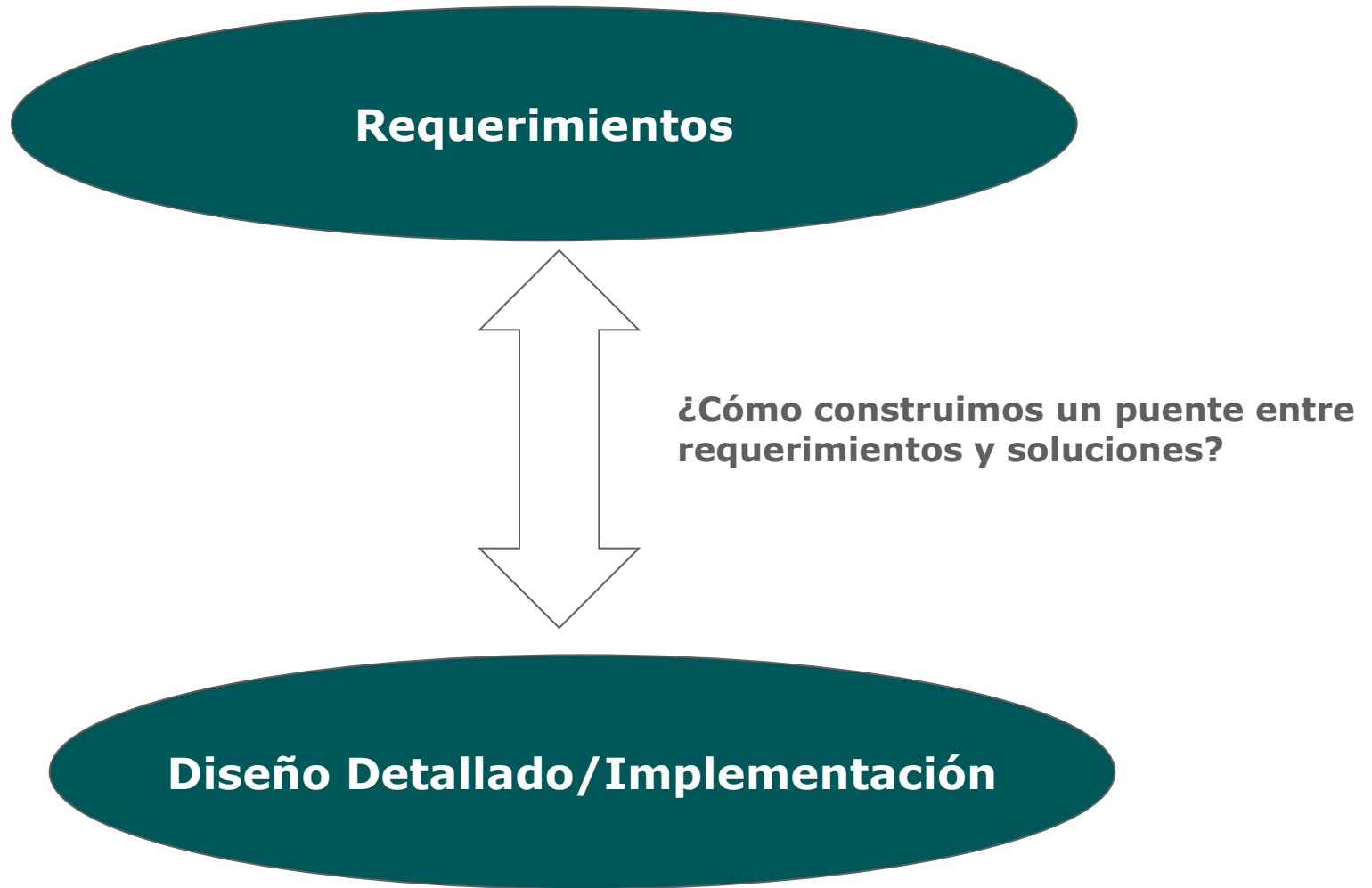


Colonial:  
Spanish Colonial

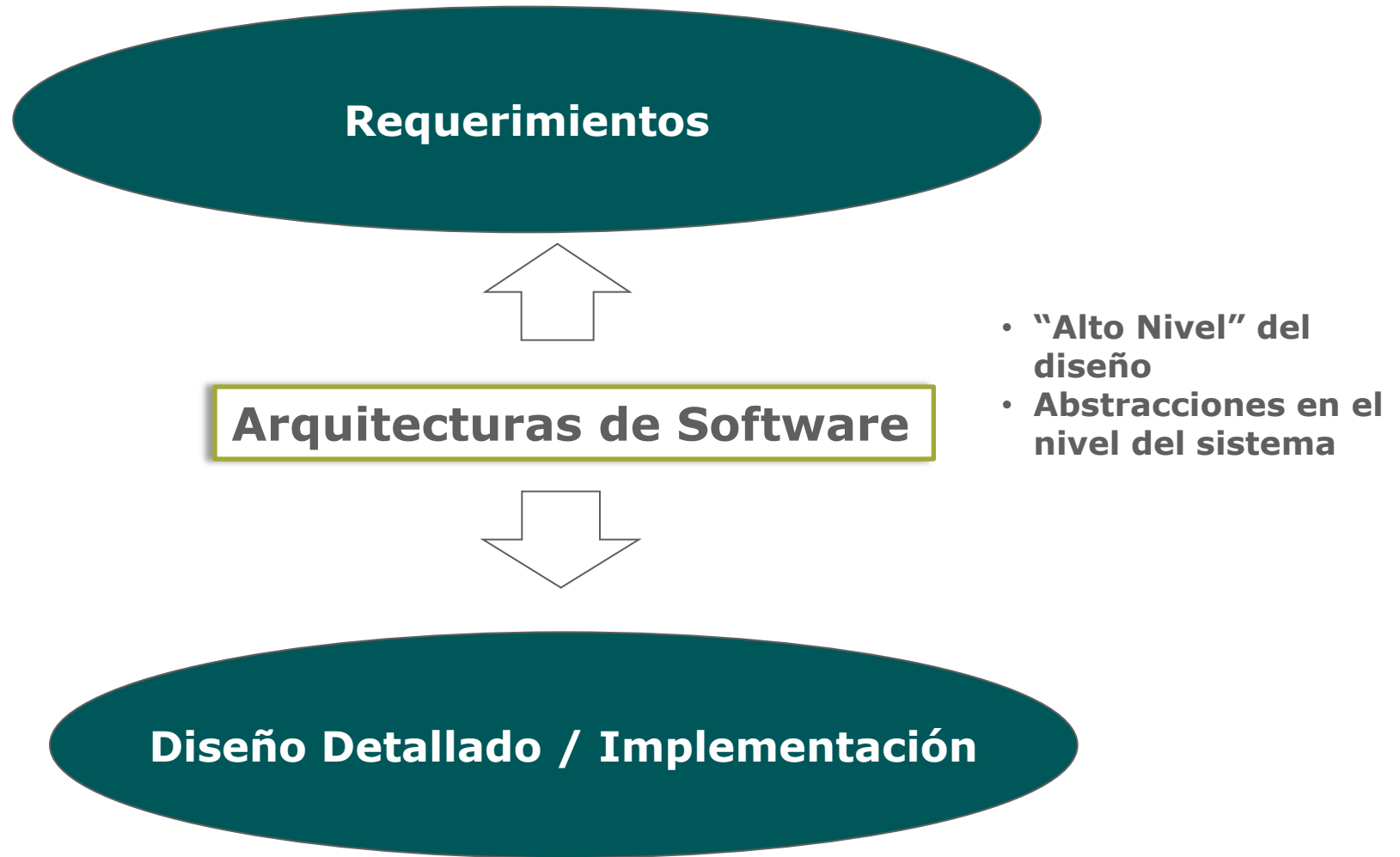
# Algunas analogías (cont.)

Roof form IF YOU SEE		TRY THESE FIRST		
	steep pitch 	Tudor, Queen Anne, Stick, Gothic Revival		Second Empire, Beaux Arts, Richardsonian Romanesque
	moderate or varied pitch 	Shingle, National Folk, Early Classical Revival		
	low pitch 	Craftsman, Spanish Eclectic, Greek Revival, Monterey		symmetrical
	steep pitch 	Gothic Revival		asymmetrical
	moderate or varied pitch 	Colonial Revival, Georgian, Adam		Georgian, Colonial Revival, Mission
	low pitch 	Italianate, Italian Renaissance, Beaux Arts		
		Dutch Colonial, Shingle, Colonial Revival, Georgian	<b>Roof-wall junction IF YOU SEE</b>	
	steep pitch 	French Eclectic, Chateausque, French Colonial		Beaux Arts, Italian Renaissance, Pueblo Revival, Spanish Eclectic, Mission, Modernistic, International, Spanish Colonial
	moderate or varied pitch 	Colonial Revival, Georgian, Adam, Early Classical Revival, Folk Victorian, Mission, Neoclassical		
	low pitch 	Italianate, Adam, Greek Revival, Italian Renaissance, Spanish Eclectic, Prairie		International, Modernistic, Spanish Eclectic, Postmedieval English
	steep pitch 	Chateausque, French Eclectic		Colonial Revival, Neoclassical, Beaux Arts, Adam, Georgian, French Eclectic, Early Classical Revival, Chateausque, Italian Renaissance
	moderate or varied pitch 	National Folk, Colonial Revival, Neoclassical, Folk Victorian, Mission		Stick, Gothic Revival
	low pitch 	Prairie, Italianate		Second Empire, Folk Victorian, Italianate

# El gran problema



## Una respuesta posible



# Arquitecturas de sistemas de software

- ▶ La arquitectura de un sistema de software:
  - ▶ Define el sistema en términos de elementos e interacción entre ellos
  - ▶ Muestra correspondencia entre requerimientos y elementos del sistema construido
  - ▶ Resuelve atributos de calidad en el nivel del sistema, como escalabilidad, flexibilidad, confiabilidad y performance
- ▶ Un paso clave para ayudar a que el desarrollo de software siga evolucionando hacia una disciplina de ingeniería



# Analogías con la ingeniería civil

- ▶ Estilos arquitectónicos: colonial, victoriano, griego
  - ▶ Paradigmas de organización de sistemas de software: pipes, layers, events, repositories
- ▶ Conocimientos específicos para un estilo en particular: cárceles, fábricas automotrices, hospitales, hoteles 5 estrellas.
  - ▶ Arquitecturas para un dominio específico, llamadas arquitecturas de referencia



# La estructura de los sistemas

- La arquitectura trata sobre la estructura de los sistemas
  - Cómo el sistema se descompone en partes
  - Cómo esas partes interactúan
- Pero esto lleva a la pregunta: ¿Qué tipos de estructuras?
  - Del código
  - Run-time
  - De deployment
  - Del entorno de desarrollo
  - Work breakdown structures
- Cada una de estas estructuras puede ser la base para una vista arquitectónica (architectural view)
  - Históricamente el foco estuvo en vistas de código

# Las definiciones más aceptadas (Bass, Clements)

- ▶ Arquitectura

La arquitectura de software de un sistema de computación es el conjunto de estructuras necesarias para razonar sobre el sistema, y comprende elementos de software, relaciones entre ellos y propiedades de ambos

- ▶ Estilo o patrón arquitectónico

- ▶ Una descripción de tipos de relaciones y elementos, junto con restricciones sobre cómo deben usarse (ej. "client server").

- ▶ Arquitectura de referencia

- ▶ Una división común de funcionalidad mapeada a elementos que cooperativamente implementan esa funcionalidad y flujos de datos entre ellos.

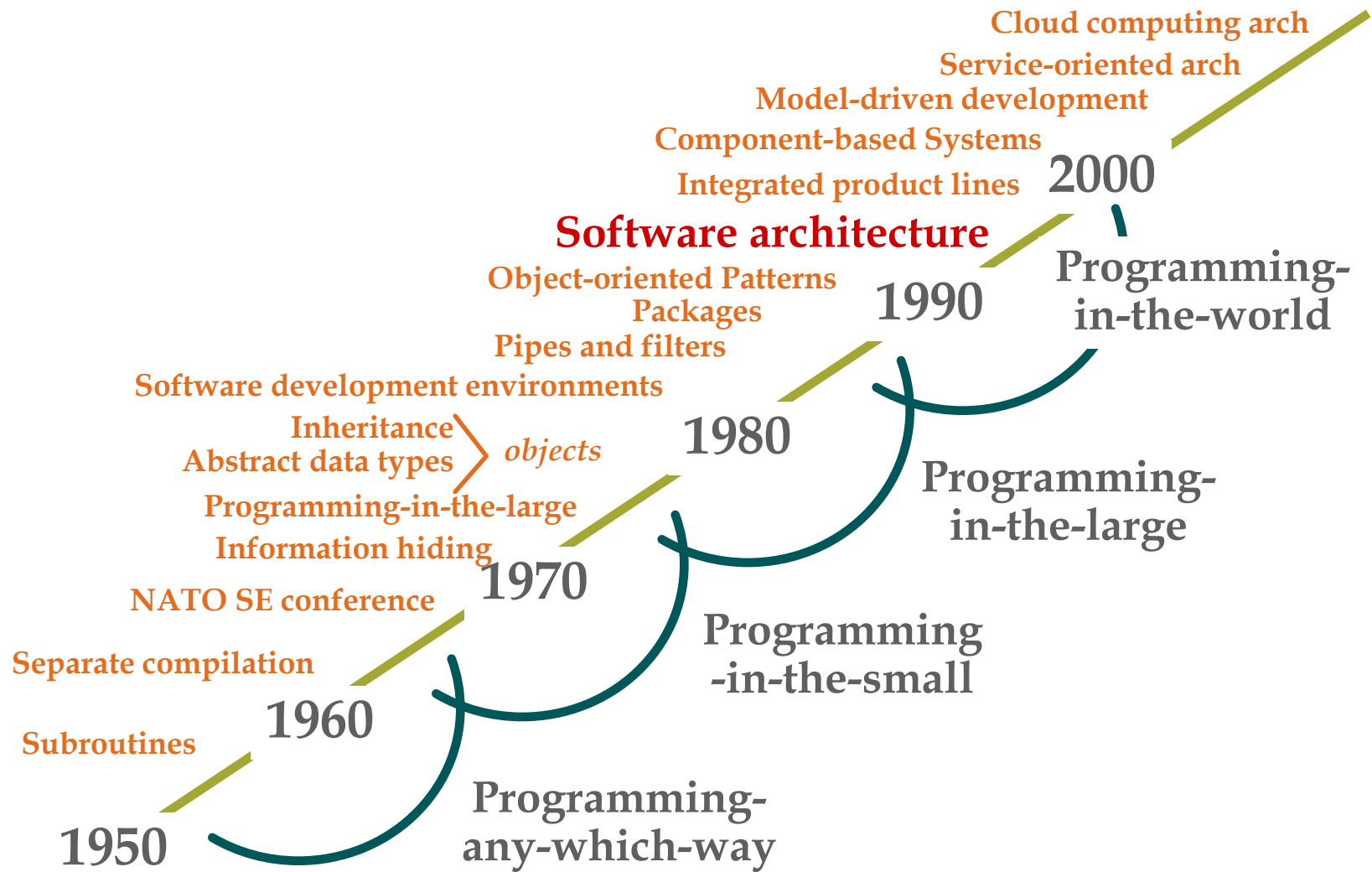
# Tres Principios Fundamentales

- ▶ La arquitectura es el conjunto de decisiones principales de diseño de un sistema de software
- ▶ Tres principios a no olvidar:
  - ▶ Toda aplicación tiene una arquitectura
  - ▶ Cada aplicación tiene al menos un arquitecto
  - ▶ La “Arquitectura” no es una fase del desarrollo
- ▶ ¿Relación entre Arquitectura y Diseño?

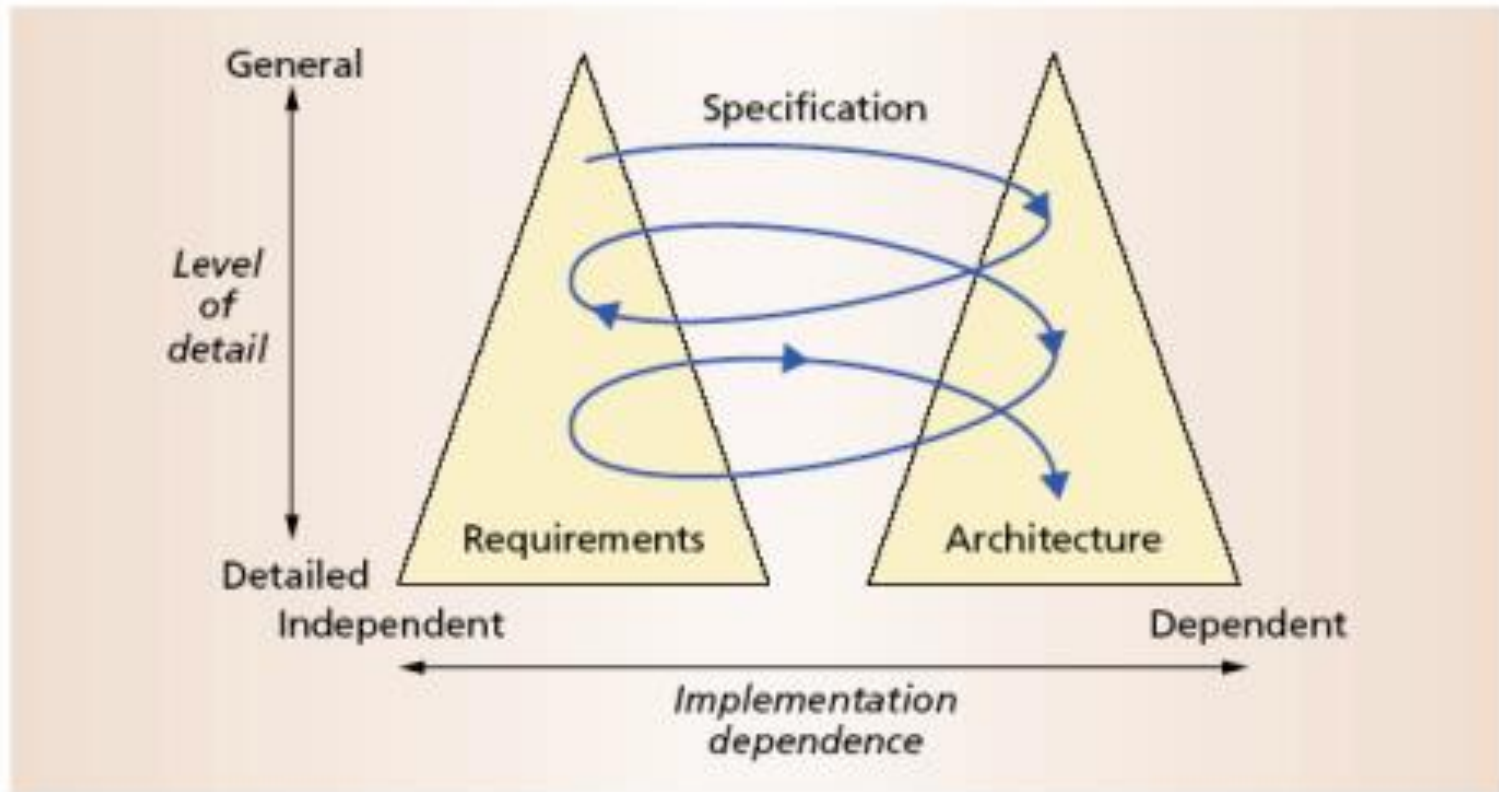
# ¿Qué hace que una arquitectura sea “buena”?

- ▶ Producto de un único arquitecto o un pequeño grupo de arquitectos con un claro líder (Brooks, Mills y otros). “Integridad conceptual”
- ▶ El equipo de arquitectura debe contar con requerimientos funcionales y atributos de calidad requeridos que sean claros
- ▶ La arquitectura debe estar documentada
- ▶ La arquitectura debe ser revisada por los “stakeholders”
- ▶ Debe ser evaluada cuantitativamente antes de que sea tarde
- ▶ Debe permitir una implementación incremental
- ▶ Módulos bien definidos basados en el ocultamiento de la información
- ▶ Interfaces claramente definidas
- ▶ No dependiente de un único producto comercial
- ▶ Usa un grupo pequeño y claro de patrones de interacción

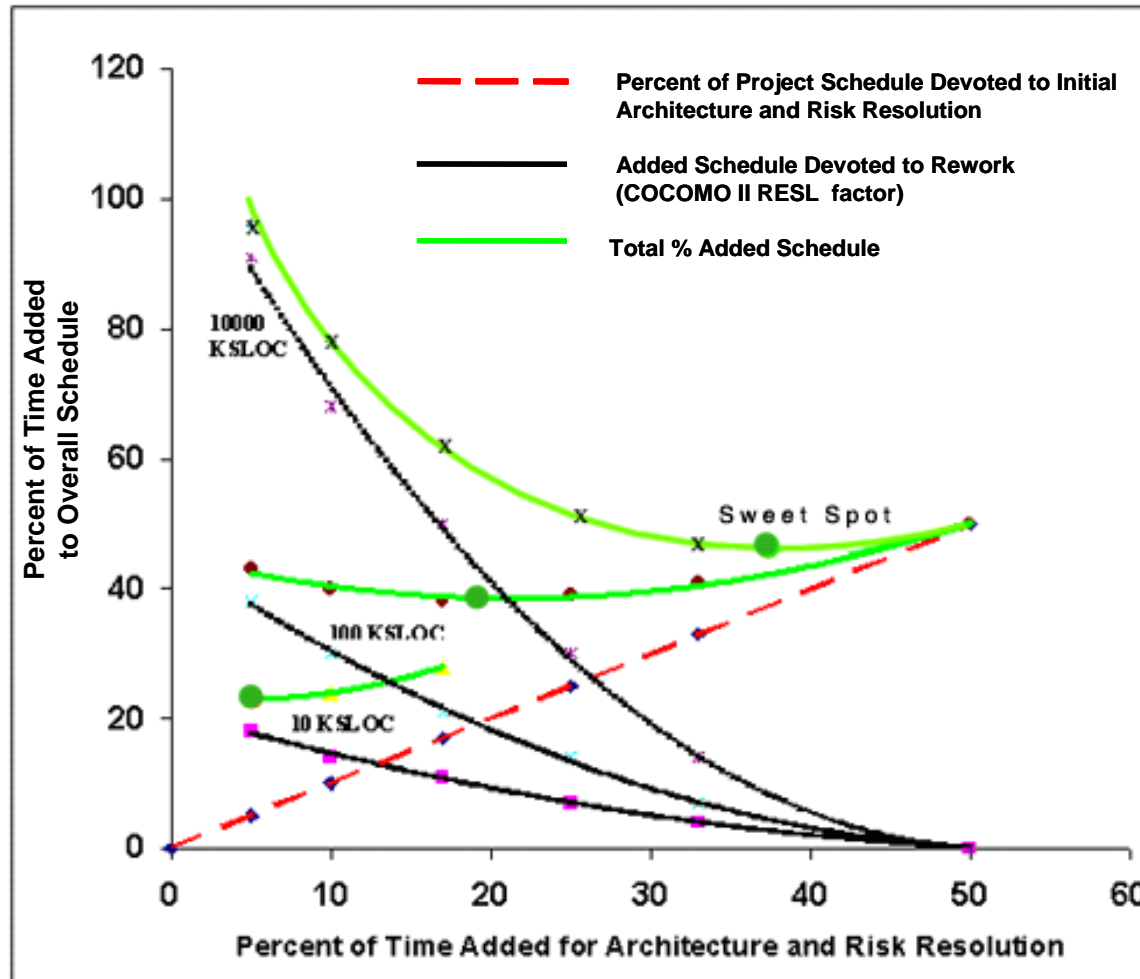
# Arquitecturas en contexto



# El Modelo “Twin Peaks”



# ¿Cuánta arquitectura?



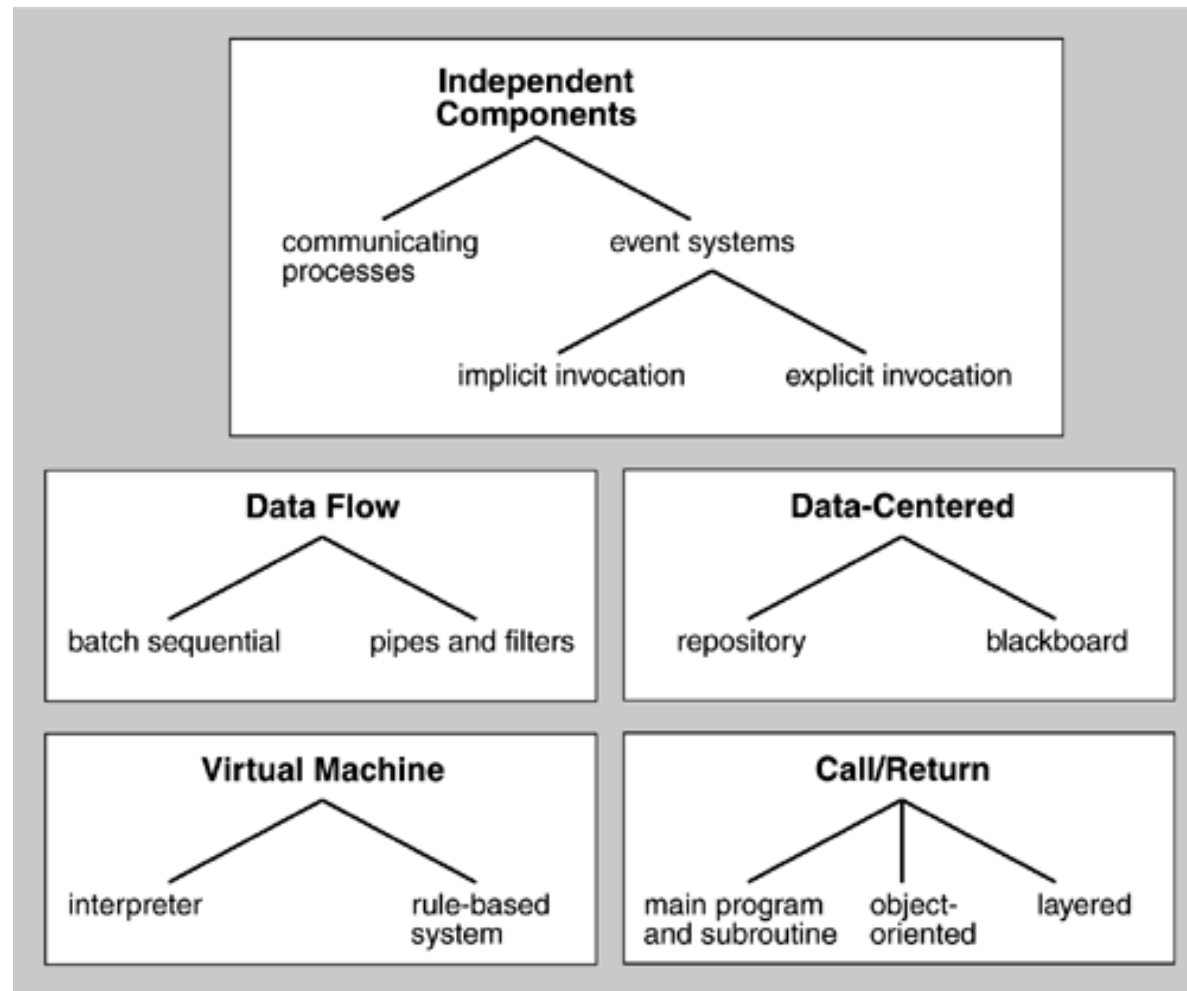
Fuente: "Using Risk to Balance Agility and Discipline: A Quantitative Analysis," Barry Boehm.

# Propiedades de los estilos arquitectónicos

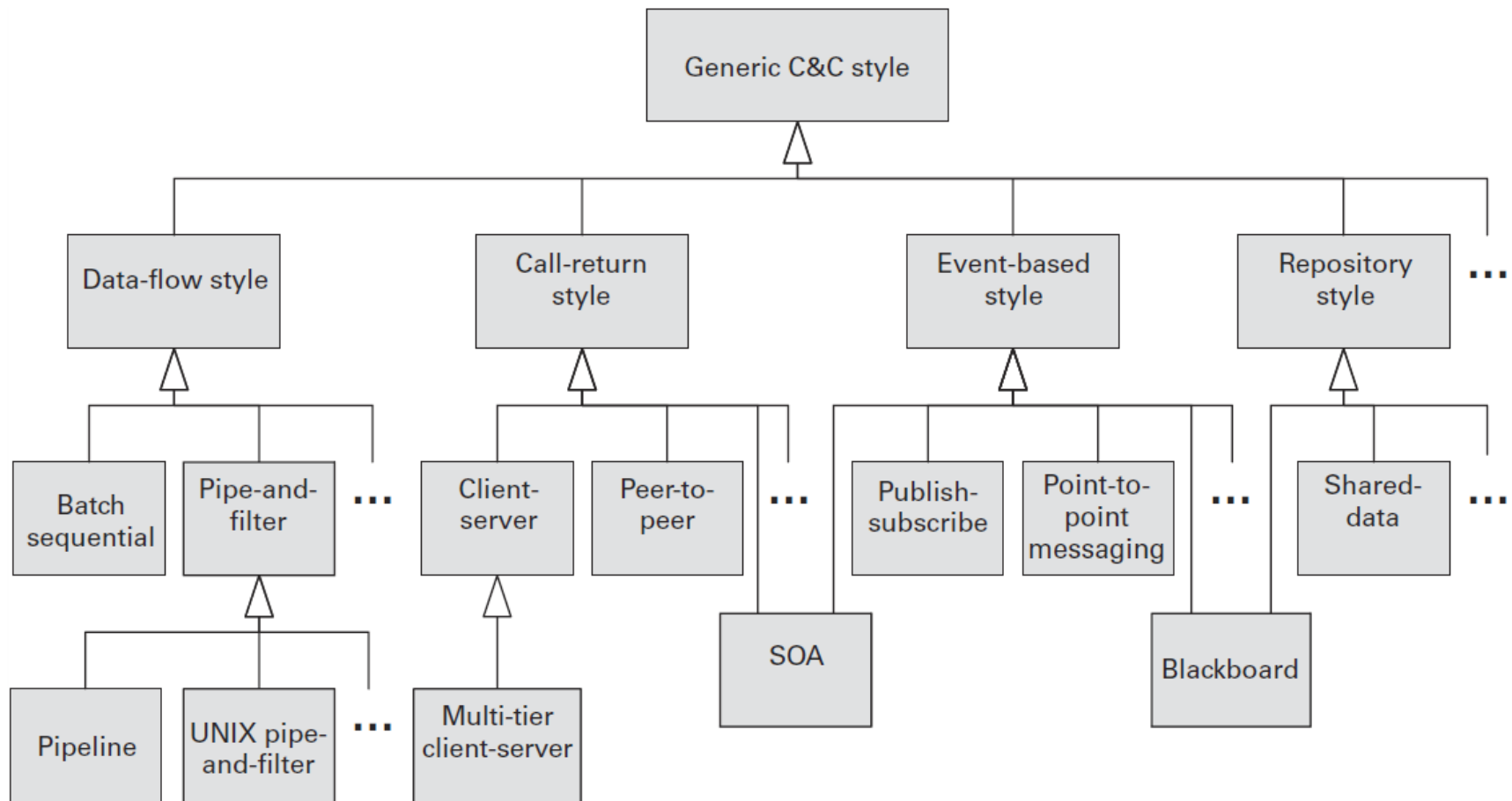
- ▶ Un vocabulario para los elementos de diseño
  - ▶ Tipos de componentes y conectores
  - ▶ Por ejemplo: clases, invocaciones, “pipes”, clientes
- ▶ Reglas de composición
  - ▶ Un estilo tiene restricciones topológicas que determinan cómo se puede hacer la composición de los elementos
  - ▶ Por ejemplo: los elementos de un “layer” se pueden comunicar sólo con los del “layer” inferior
- ▶ Semántica para esos elementos
- ▶ Idealmente, criterios para la evaluación de una arquitectura o formas de analizarla; generación de código
- ▶ Importante: un estilo arquitectónico no define la funcionalidad de un sistema. Desde ese punto de vista es algo “abstracto”.



# Taxonomía de estilos arquitectónicos



# Taxonomía de estilos arquitectónicos



# Arquitecturas Heterogéneas

- ▶ Resultan de la combinación de distintos estilos
- ▶ Por ejemplo:
  - ▶ Los componentes de un sistema “layered” pueden tener una estructura interna que use otro estilo
  - ▶ Una arquitectura hecha con J2EE probablemente resulte en una arquitectura heterogénea que incluya:
    - ▶ Layered
    - ▶ Repository
    - ▶ Independent components
    - ▶ Information hiding → Objects



**En sistemas medianos / grandes, es más probable que un estilo arquitectónico describa una parte de un sistema que al sistema completo.**

## Bibliografía

