

# Arquitectura de Software

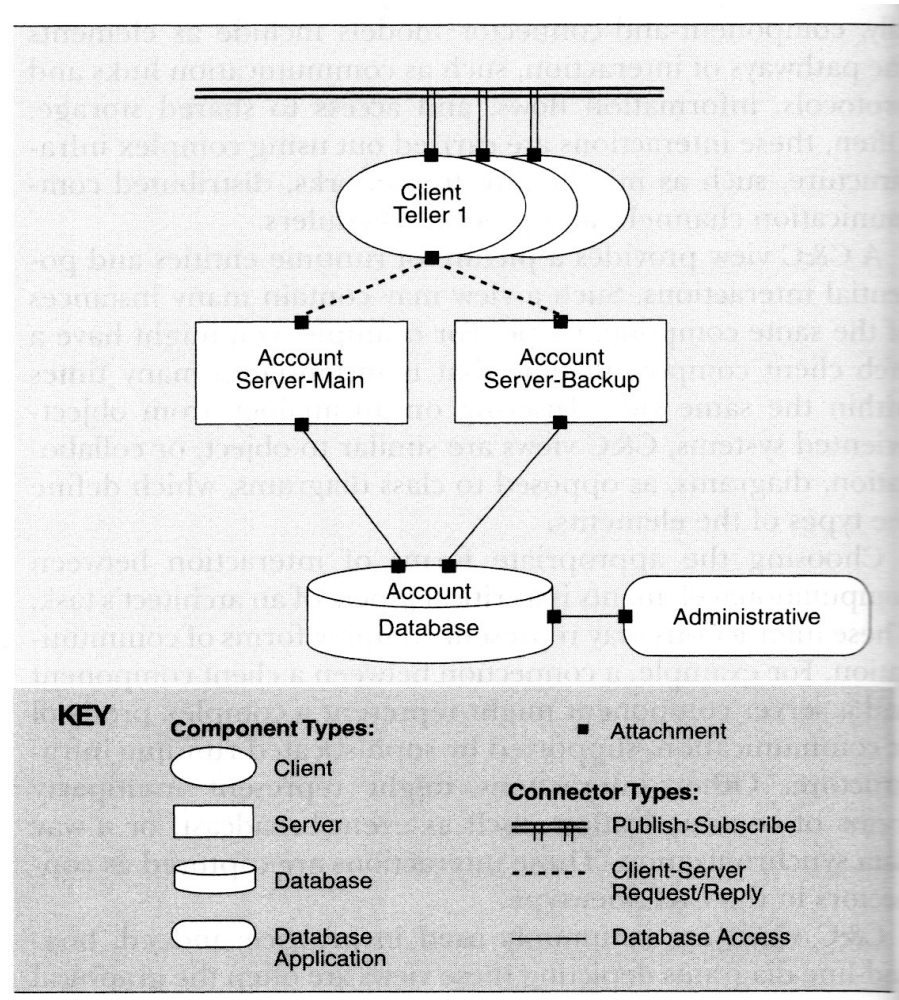
## Documentación de Arquitecturas Parte II

## Component-and-Connector Viewtype

|             |   |
|-------------|---|
| Elementos   | Components<br>Connectors  |
| Relaciones  | Attachment (puertos y conectores)   |
| Propiedades | Nombre<br>Tipo  |
| Topología   | No tiene restricciones  |
| Usos        | <ul style="list-style-type: none"><li>•Razonar sobre atributos de calidad del sistema durante su ejecución</li><li>•Identificar los principales componentes en ejecución</li><li>•Identificar controlflow y dataflow</li><li>•Identificar paralelismo</li></ul> |
| Notaciones  | <ul style="list-style-type: none"><li>•Informales</li><li>•ADLs</li><li>•UML</li></ul>  |

# Conceptos Básicos

Ejemplo de notación informal para repesentar un estilo C&C

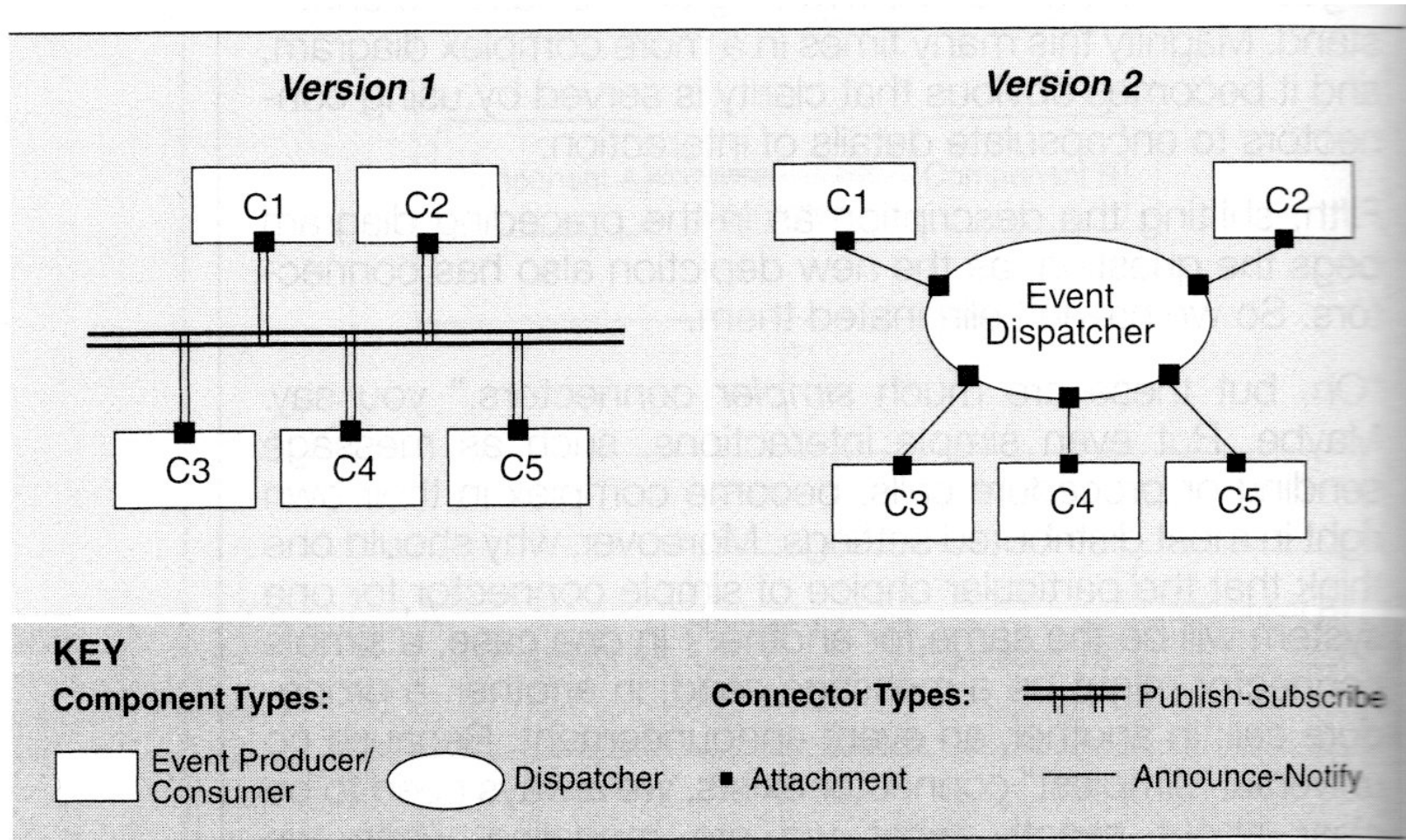


Tomado de "Documenting Software Architectures" pag 104



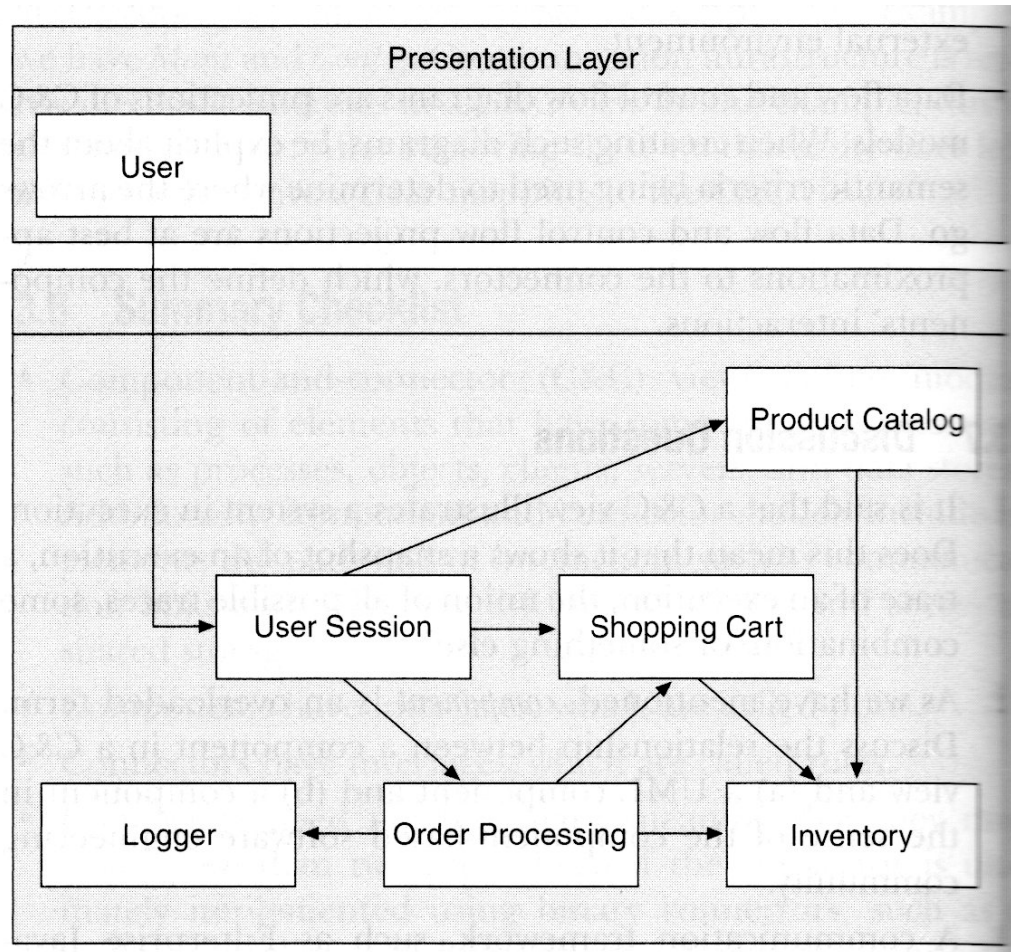
# Conceptos Básicos

Ejemplo de notación informal para repesentar un estilo C&C



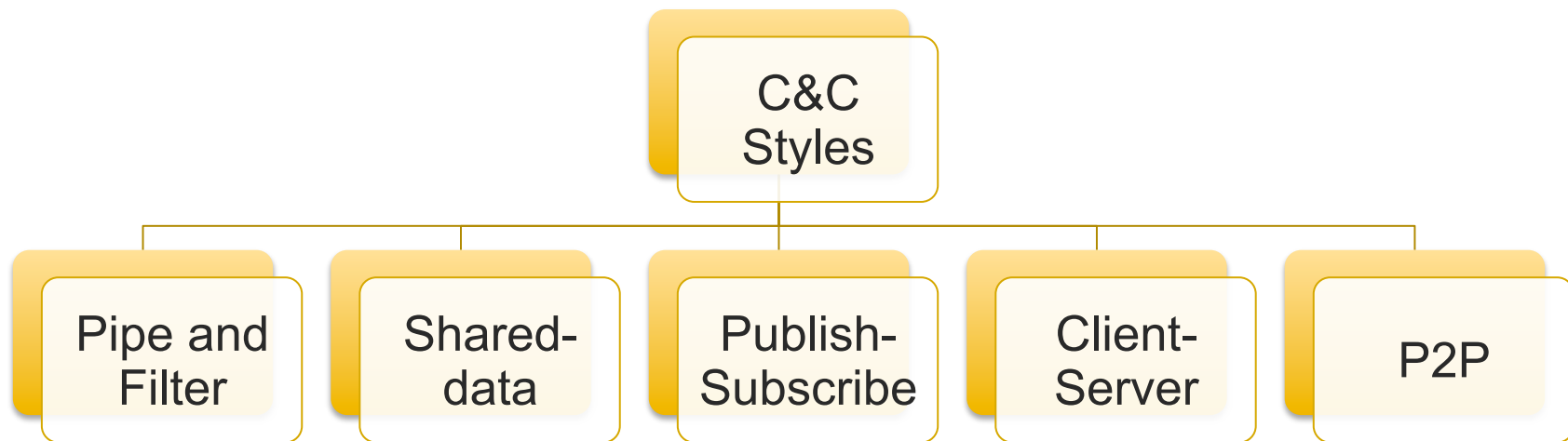
# Conceptos Básicos

Ejemplo de notación informal para repesentar un estilo C&C. Correcto?



Tomado de "Documenting Software Architectures" pag 122

# Conceptos Básicos



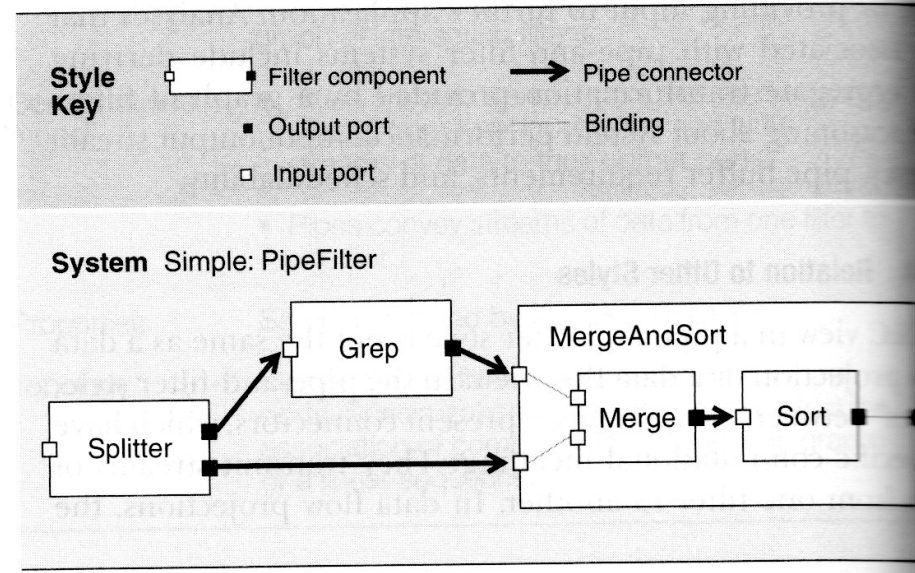
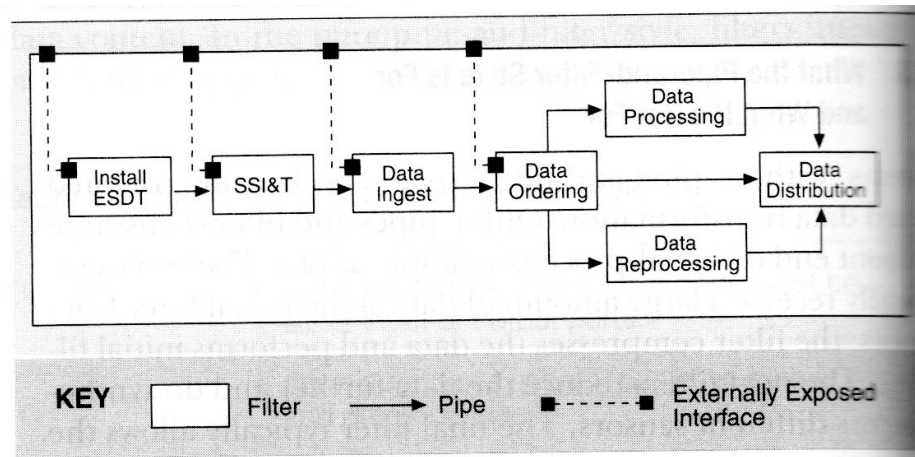


# Conceptos Básicos

## Pipe-and-Filter Style

|             |  |
|-------------|--|
| Elementos   | pipes, filtros   |
| Relaciones  | Attachment   |
| Propiedades |  |
| Modelo      | Filtros transformar datos<br>Pipes transportar datos     |
| Topología   | Salida de un pipe se conecta con la entrada de otro pipe |
| Notación    | Informales<br>UML  |

# Conceptos Básicos



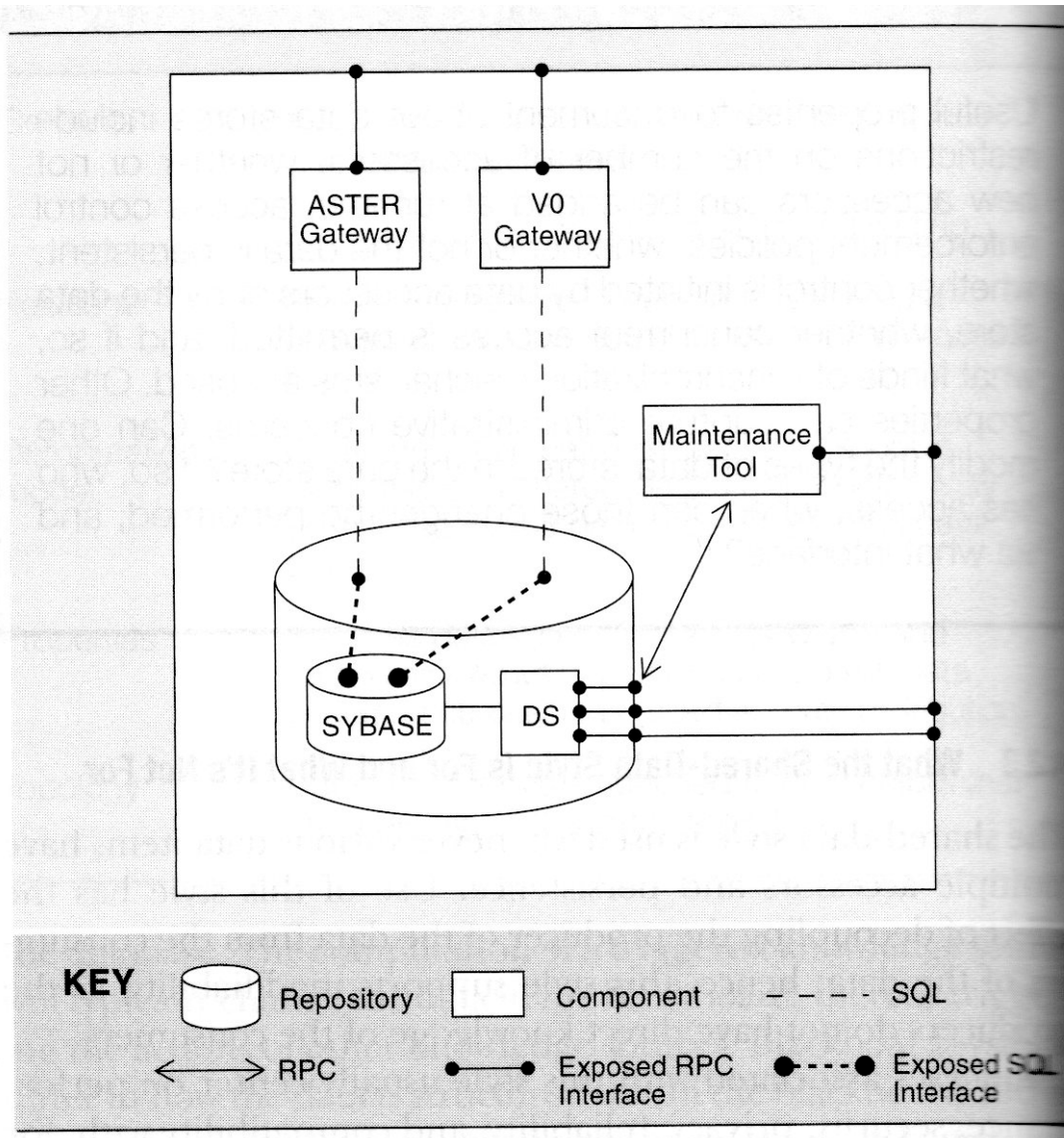
Tomado de "Documenting Software Architectures" pag 128



# Conceptos Básicos

| Shared-Data Style |  |
|-------------------|--|
| Elementos         | Componentes<br>Conectores  |
| Relaciones        | Attachment   |
| Propiedades       |  |
| Modelo            | Comunicación entre los componentes es mediada<br>por un depósito compartido de datos |
| Topología         |  |
| Notación          | Informales<br>UML  |

# Conceptos Básicos



Tomado de "Documenting Software Architectures" pag 132

# Conceptos Básicos

| Publish-Subscribe Style |  |
|-------------------------|--|
| Elementos               | Componentes<br>Conectores  |
| Relaciones              | Attachment   |
| Propiedades             |  |
| Modelo                  | Sistema de componentes independientes que anuncian eventos y reaccionan a otros eventos anunciados |
| Topología               |  |
| Notación                | Informales<br>UML  |

# Conceptos Básicos

Ejemplo de notación informal para represenar un estilo publish-subscribe

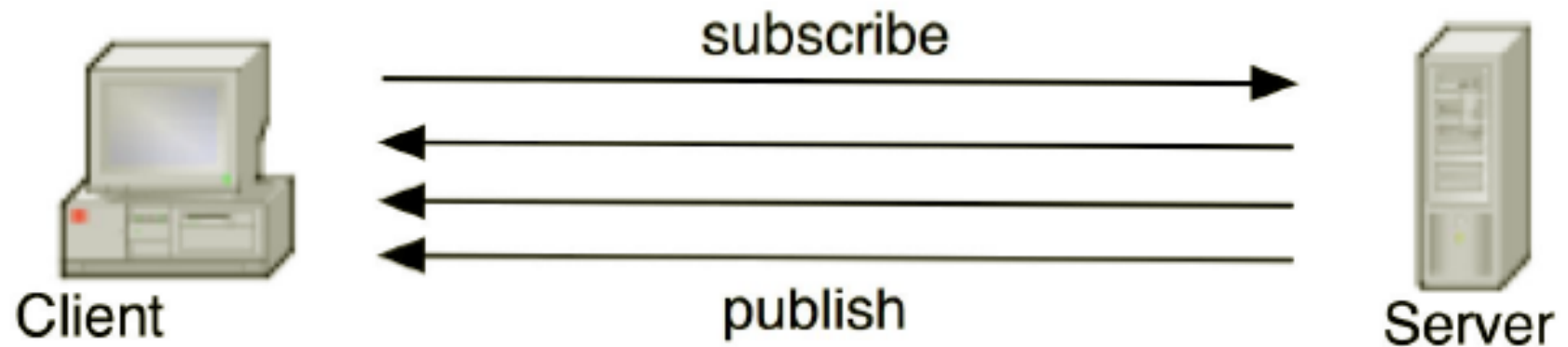


Imagen tomada de: "Service Oriented Architectures "- Phillip J. Windley



# Conceptos Básicos

## Client-Server Style

|             |  |
|-------------|--|
| Elementos   | Componentes (cliente, servidores)<br>Conectores (request/reply)                                  |
| Relaciones  | Attachment   |
| Propiedades |  |
| Modelo      | Los clientes inician actividades, solicitan servicios de los servidores y esperan las respuestas |
| Topología   |  |
| Notación    | Informales<br>UML  |

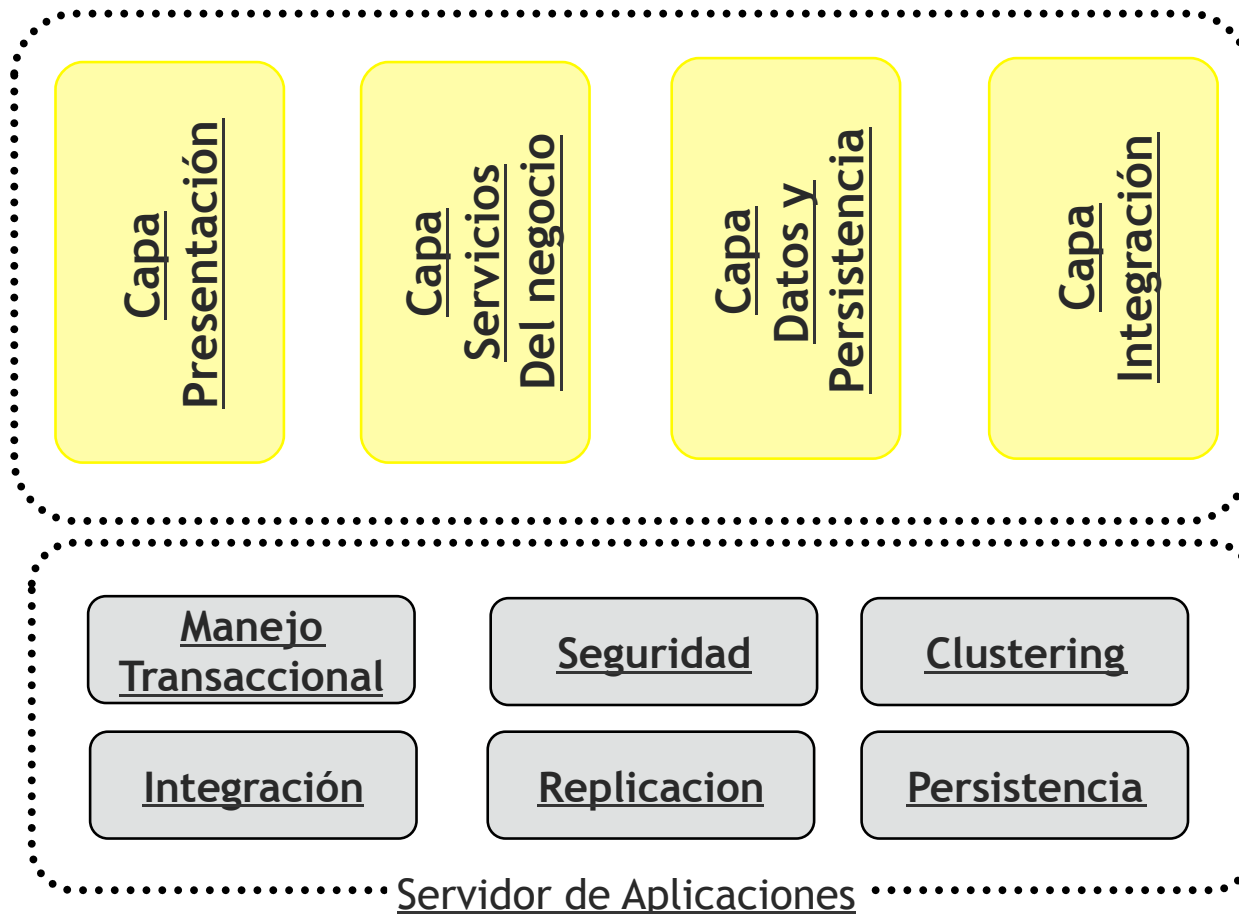


# Conceptos Básicos

| N-Tier Style |   |
|--------------|---|
| Elementos    | Componentes (cliente, servidores, presentación)<br>Conectores (request/reply, publish/subscribe)                          |
| Relaciones   | Attachment  |
| Propiedades  |   |
| Modelo       | Los clientes se conectan a componentes de negocio y solicitan servicios. La persistencia está separada de la presentación |
| Topología    |   |
| Notación     | Informales<br>UML   |

# Conceptos Básicos

Ejemplo de notación informal para represenar un estilo N-Tier



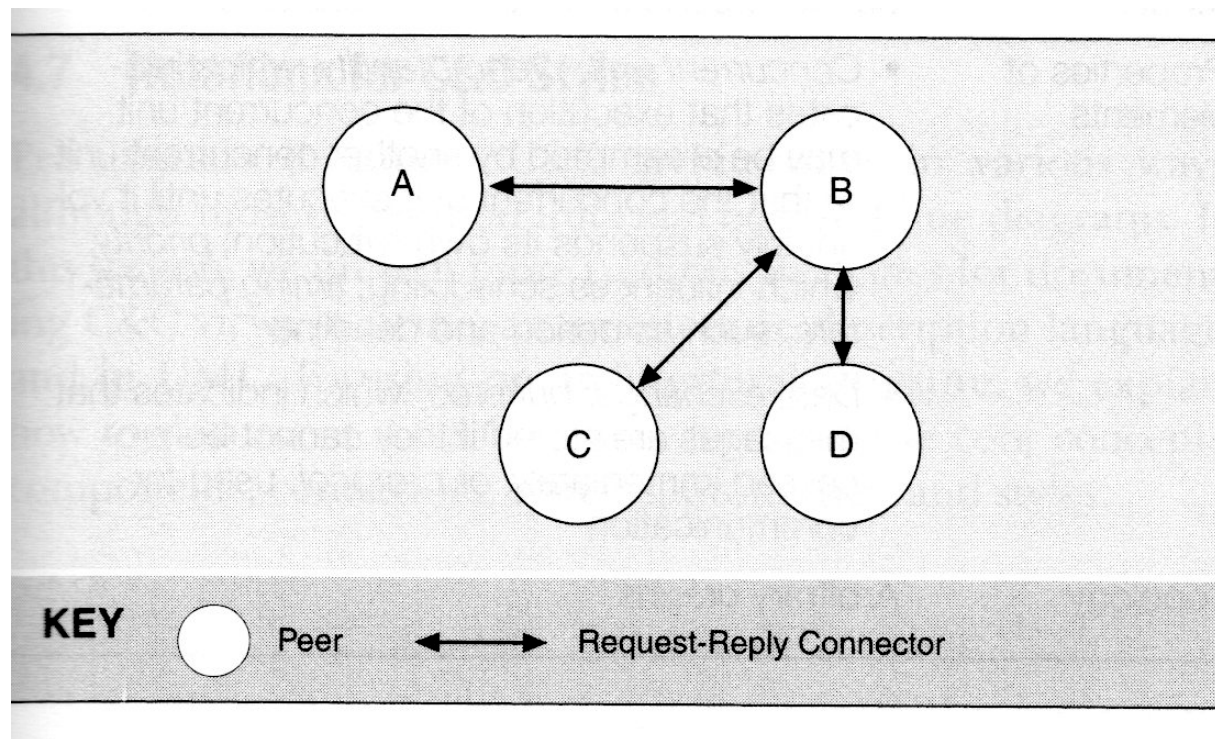
# Conceptos Básicos

| Peer-toPeer Style |  |
|-------------------|--|
| Elementos         | Componentes (peers)<br>Conectores  |
| Relaciones        | Attachment   |
| Propiedades       |  |
| Modelo            | El sistema se basa en peers que cooperan entre sí, requiriendo servicios entre ellos |
| Topología         |  |
| Notación          | Informales   |



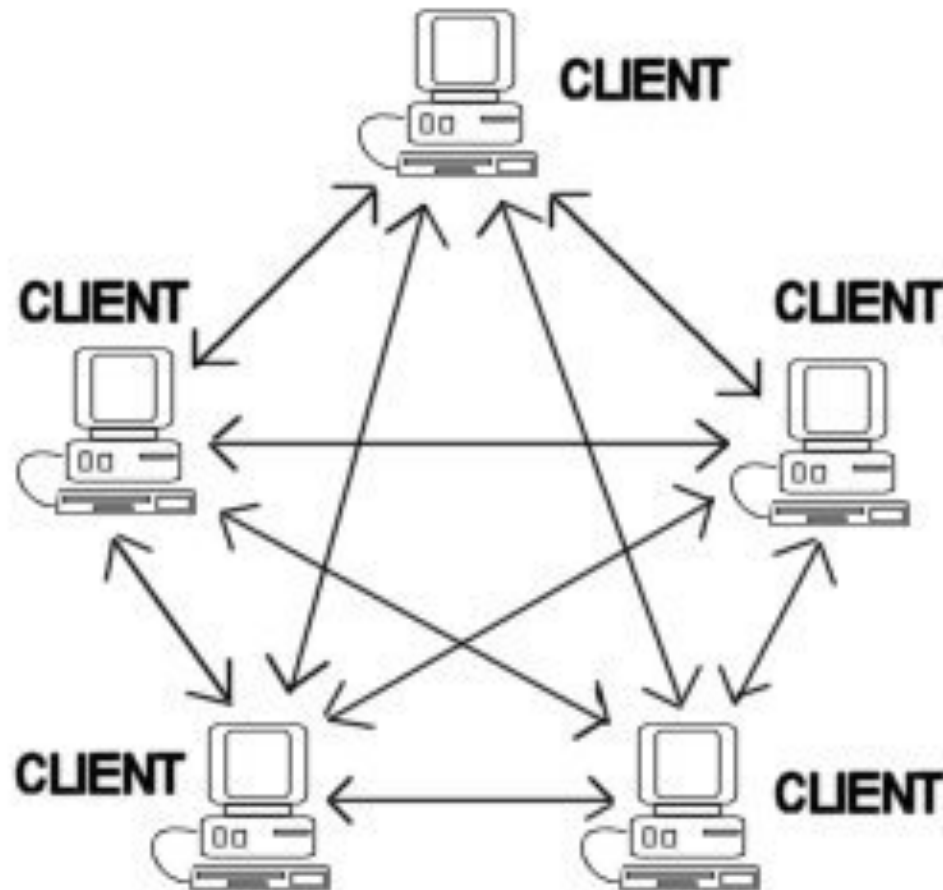
# Conceptos Básicos

Ejemplo de notación informal para represenar un estilo peer-to-peer



# Conceptos Básicos

Ejemplo de notación informal para represenar un estilo peer-to-peer



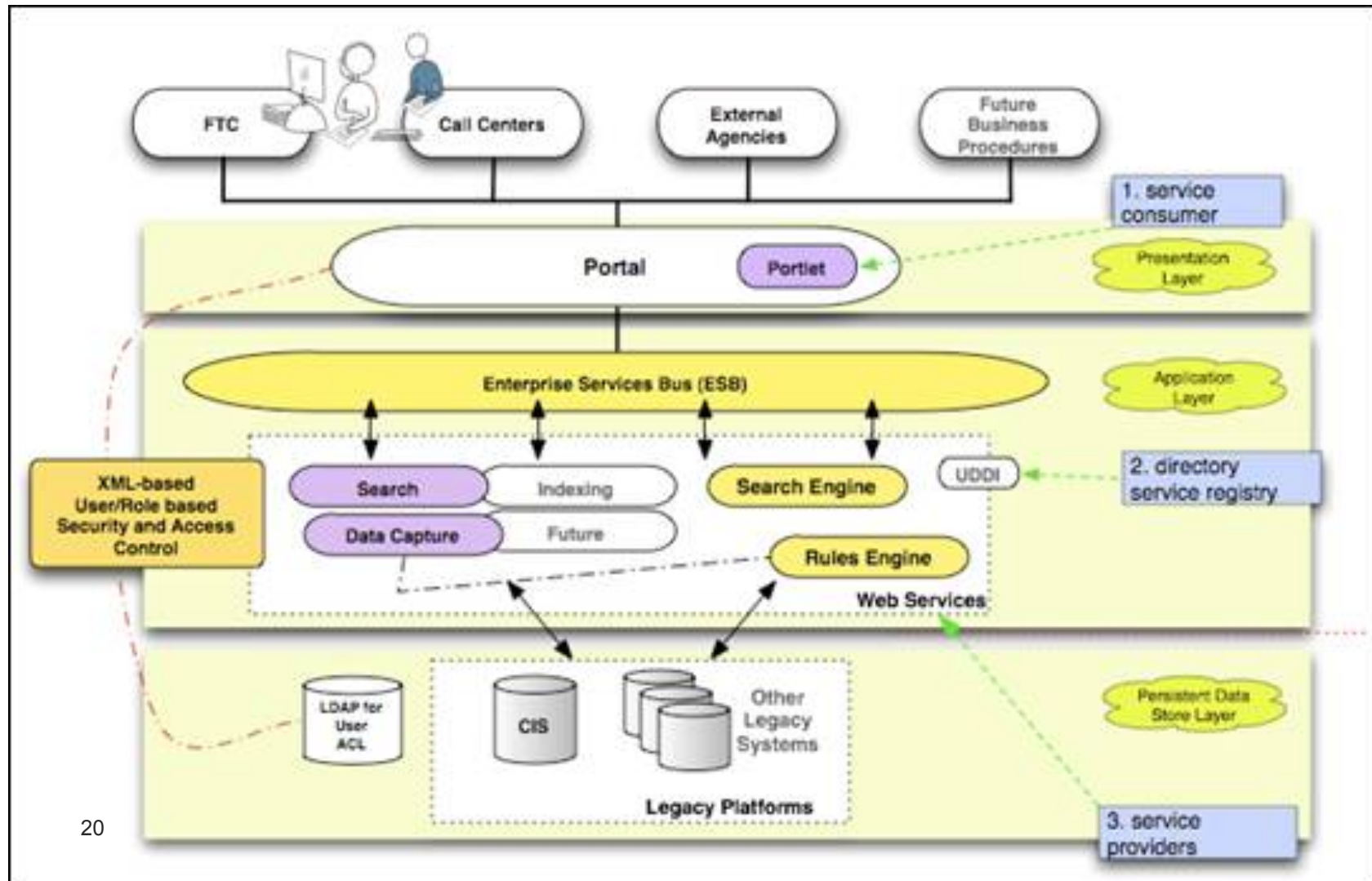
# Conceptos Básicos

## Service-Oriented Architecture (SOA) Style

|             |  |
|-------------|--|
| Elementos   | Componentes (cliente, servidores, presentación, servicios, bus)<br>Conectores (request/reply, publish/subscribe) |
| Relaciones  | Attachment   |
| Propiedades |  |
| Modelo      | Orquestación de servicios  |
| Topología   |  |
| Notación    | Informales<br>UML  |

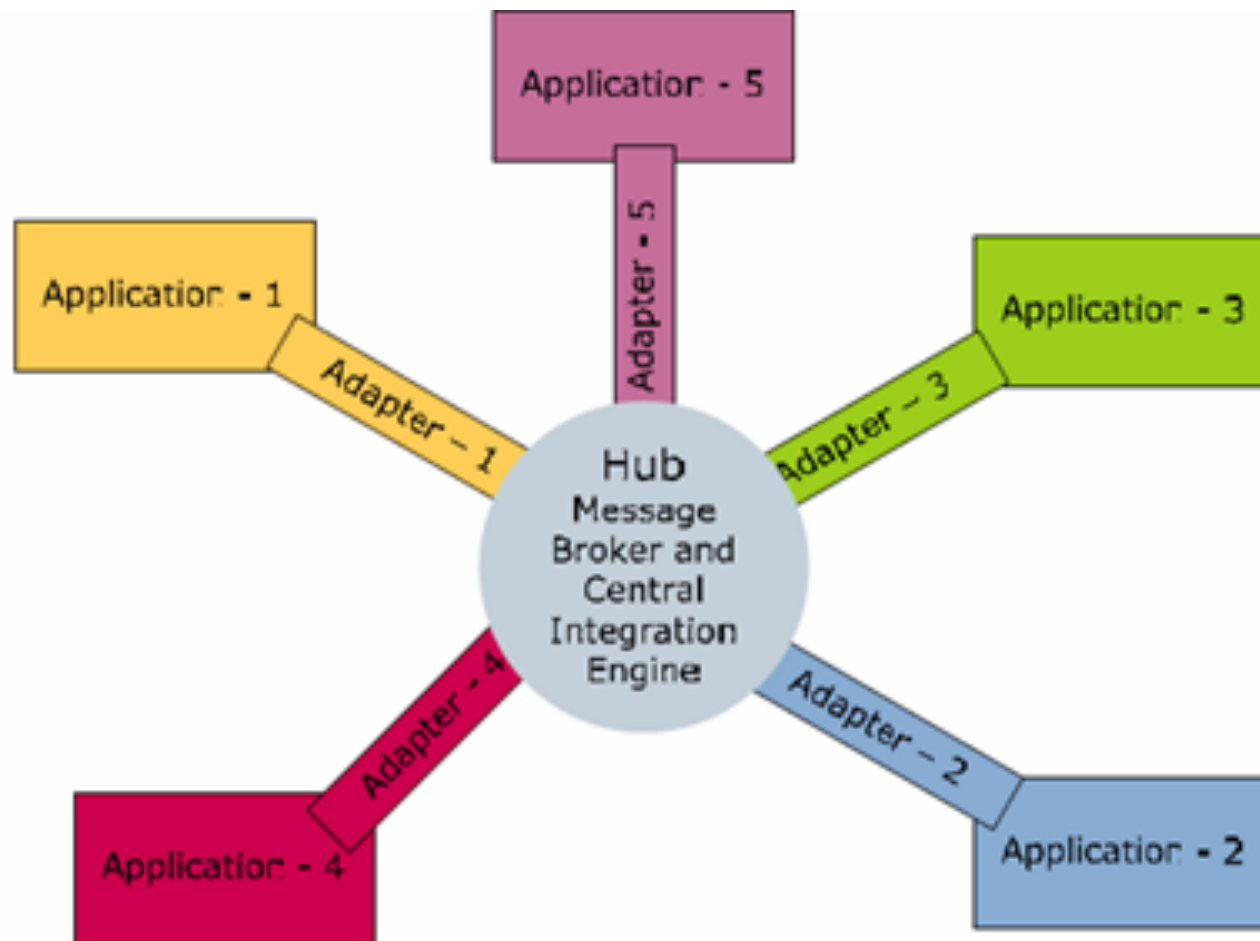
# Conceptos Básicos

Ejemplo de notación informal para represenar un estilo SOA

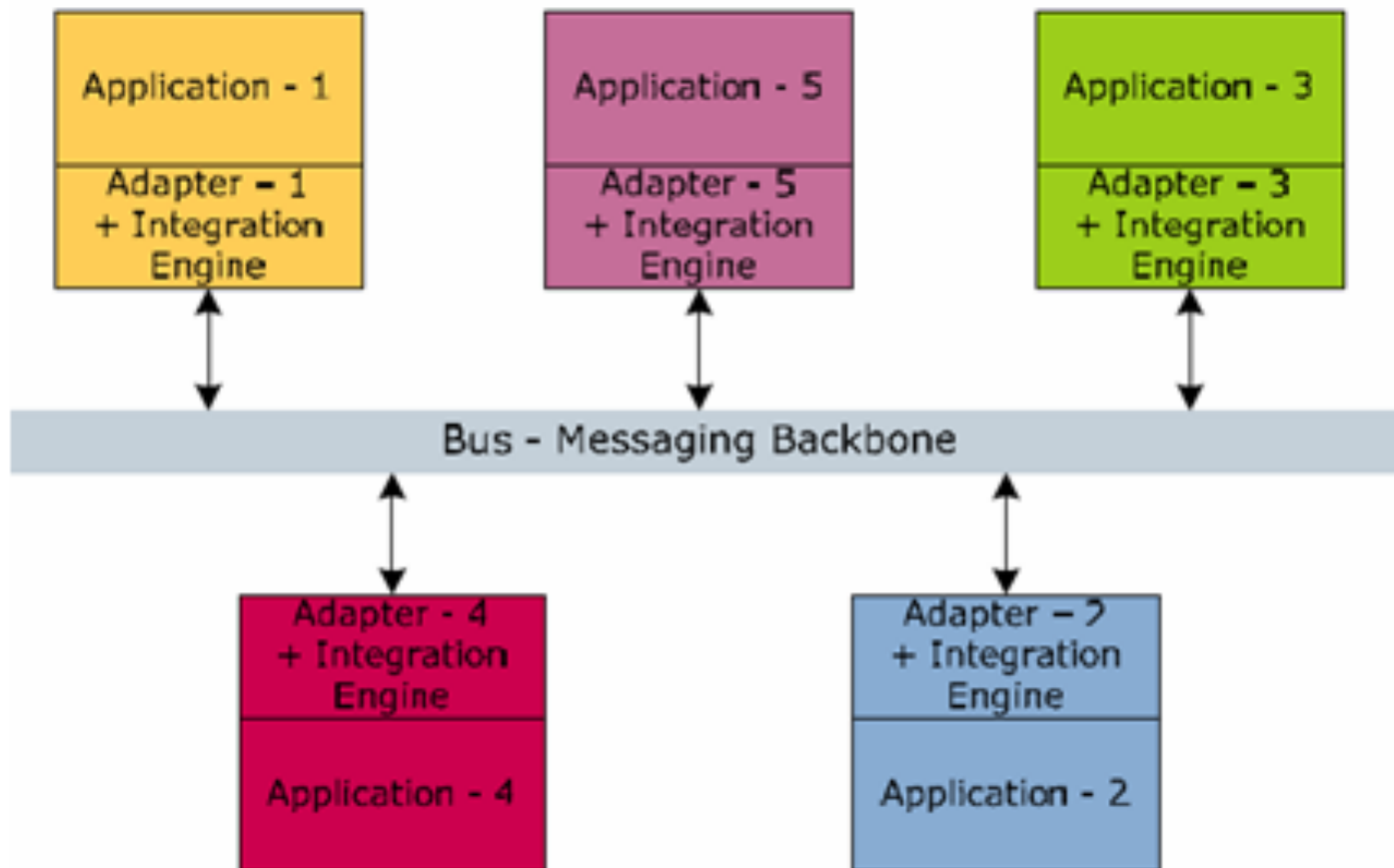




## Estilo Hub and Spoke



## Estilo Message Bus



# Conceptos Básicos

## Allocation Viewtype

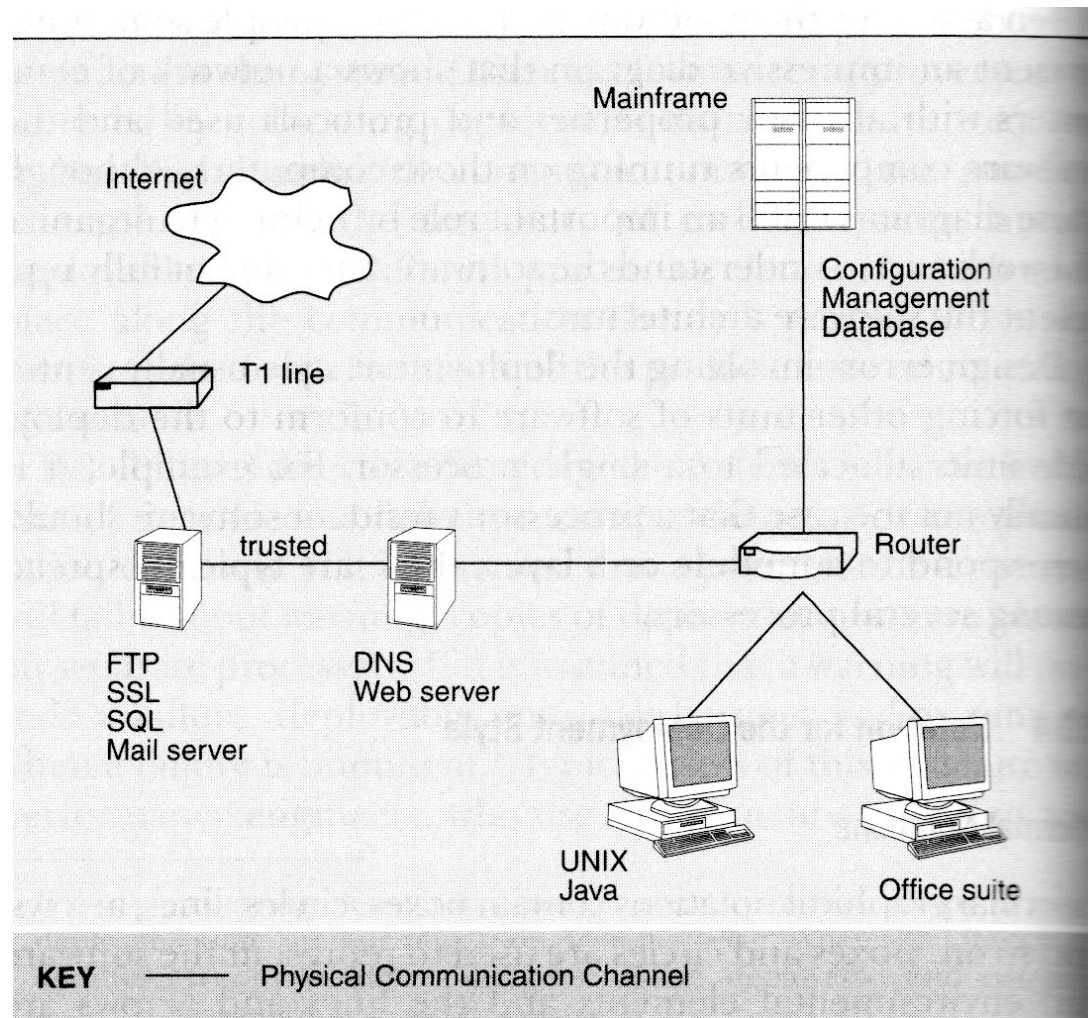
|             |                            |
|-------------|----------------------------|
| Elementos   | Software<br>Hardware       |
| Relaciones  | Allocated-to               |
| Propiedades | Nombre                     |
| Topología   | No tiene restricciones     |
| Usos        | •Específico de cada estilo |
| Notaciones  | •Informales<br>•UML        |

# Conceptos Básicos

| Deployment Style |   |
|------------------|---|
| Elementos        | Elementos de software<br>Elementos del ambiente |
| Relaciones       | Allocated-to                                    |
| Propiedades      | Requerido<br>Provisto                           |
| Modelo           | Orquestación de servicios                       |
| Topología        |   |
| Notación         | Informales<br>UML                               |

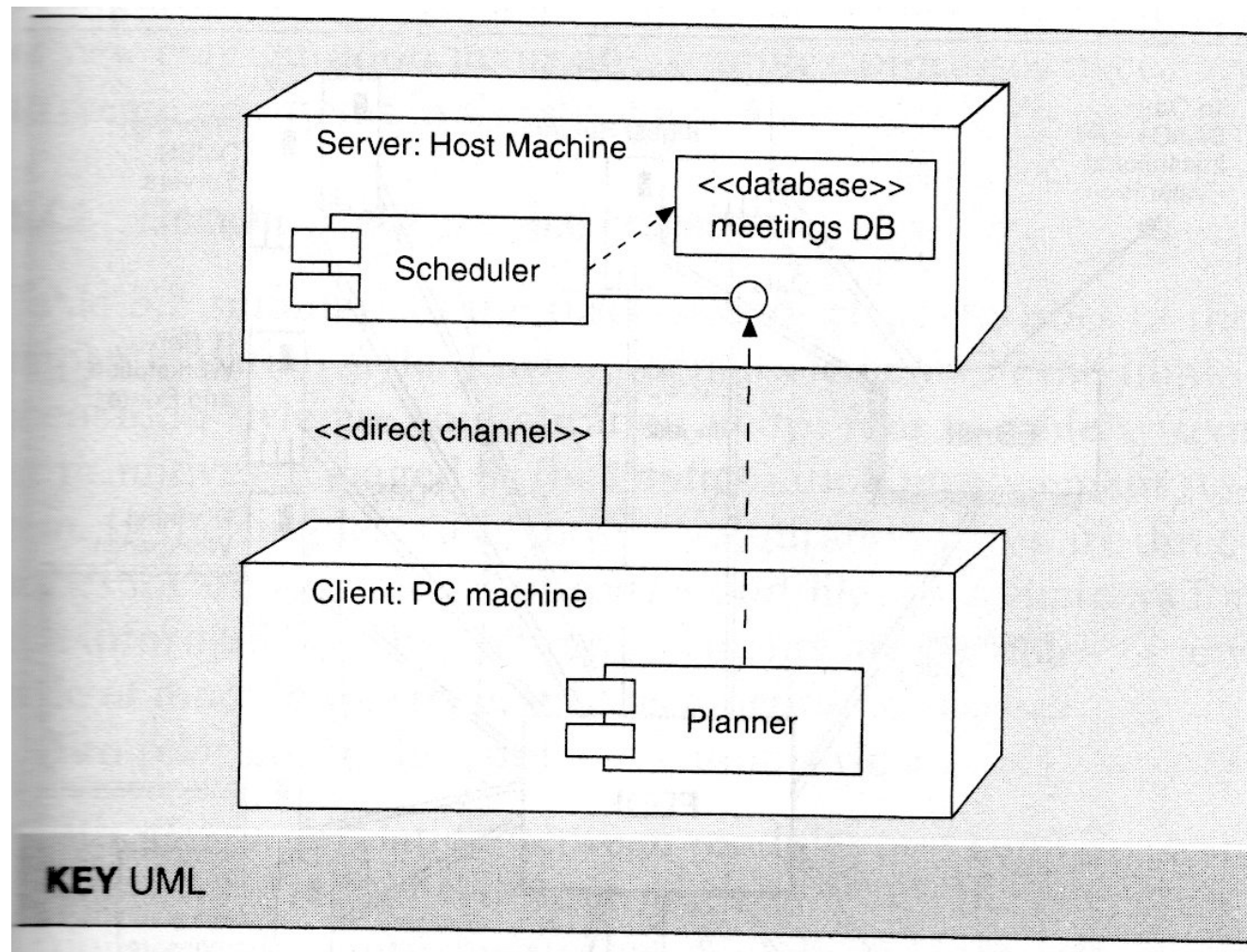


# Conceptos Básicos



Tomado de "Documenting Software Architectures" pag 174

# Conceptos Básicos



- ❑ [1] Rozanski N, Woods E. “Software Systems Architecture” Addison-Wesley. 2005
- ❑ [2] Clements, Paul et al. “Documenting Software Architectures: views and Beyond”. Addison-Wesley. 2002
- ❑ [3] Len Bass et Al. “Software Architecture in practice” Second Edition, Addison Wesley, 2007