Construcción de Sistemas Distribuidos

“Introducción a los Sistemas Distribuidos”

Rogelio Ferreira Escutia

Contenido

1.1. Características de un sistema distribuido

Sistemas Distribuidos

Definición

- **“Un sistema Distribuido es una colección de computadoras independientes o autónomas que aparecen ante los usuarios del sistema como una única computadora”.**

  Andrew Tanenbaum

- **“Es aquel en el que los componentes de hardware y software se localizan en computadoras unidos mediante red, comunican y coordinan sus acciones sólo mediante paso de mensajes”.**

  George Coulouris

Sistemas Distribuidos - Características

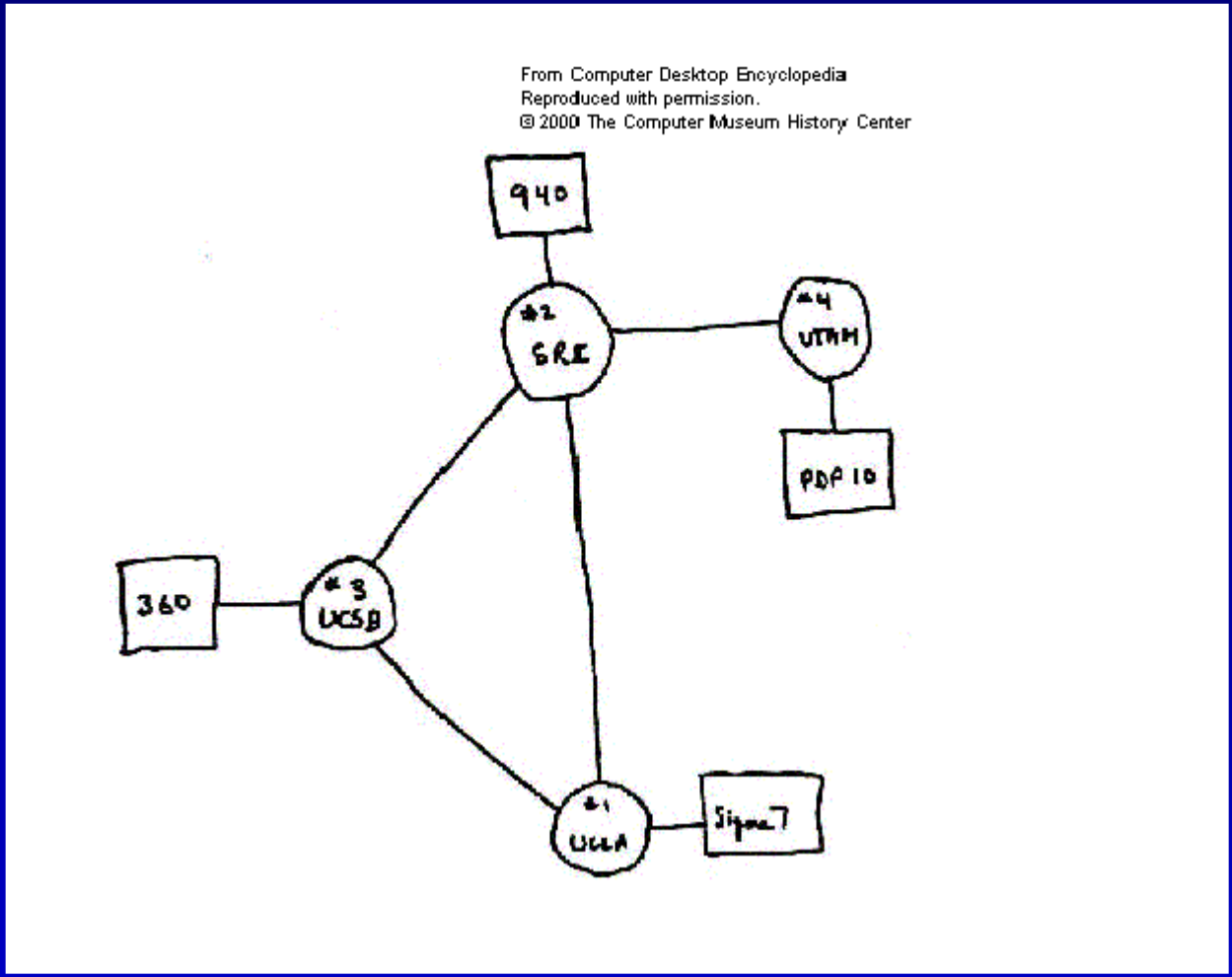
- **Un conjunto de unidades con memoria propia.**
- **Sistemas globales (locales o remotos) para sincronizar y comunicar a todos los CPU´s.**
- **Algunos CPU´s pueden dejar de comunicarse con otros, pero el sistema distribuido no puede fallar en su totalidad.**
- **En caso de existir alguna falla en algunos CPU´s, deben existir formas de recuperar la información y el sistema debe de continuar funcionando.**
- **Deben existir sistemas de protección global del sistema.**

Red Vs. SD´s

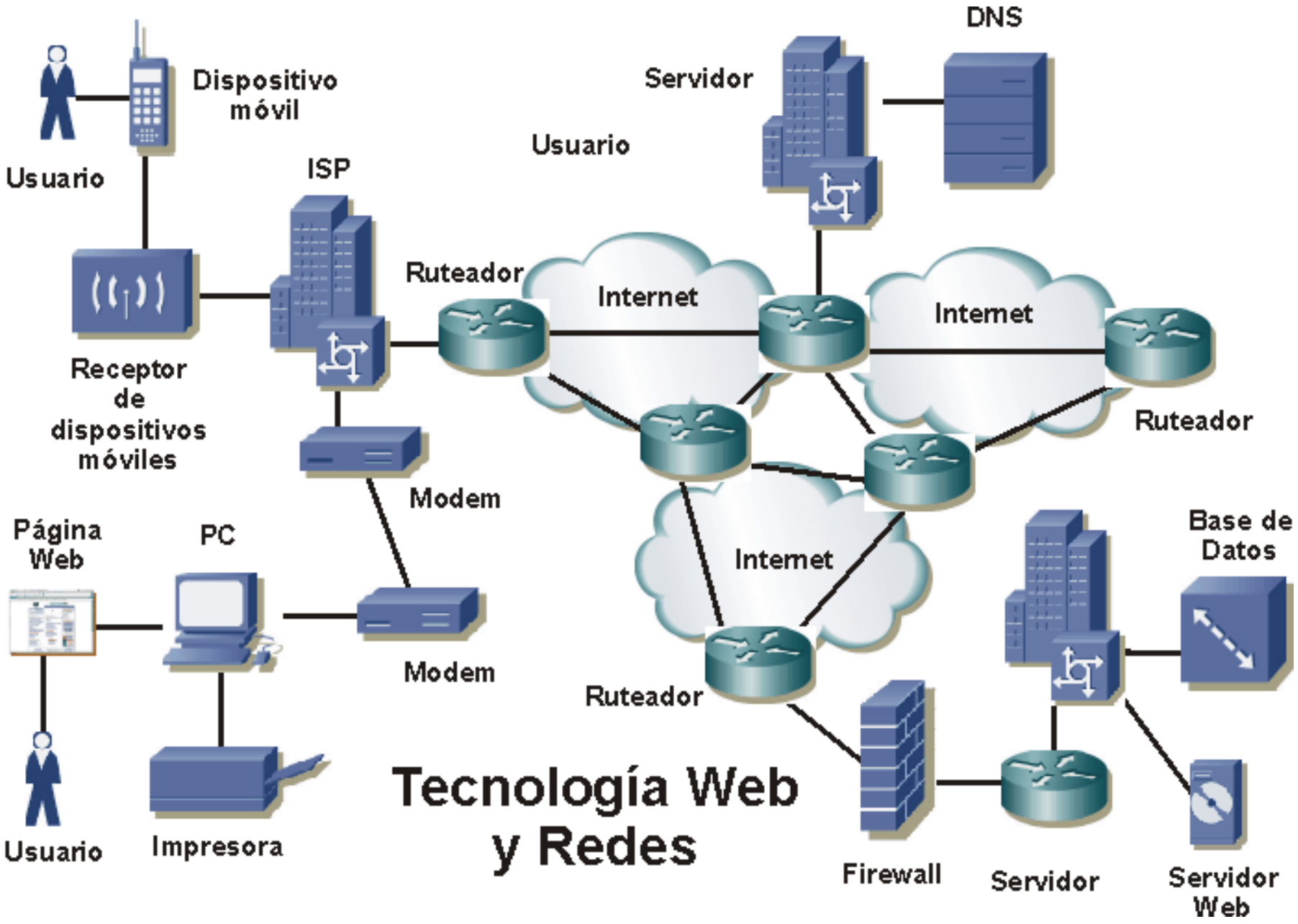
- Diferencias entre Sistemas de Red y Sistemas Distribuidos
- En una red, los procesos se ejecutan en forma local autónoma. Estos procesos deben interactuar pero tomando decisiones locales sin tomar en cuenta procesos y recursos remotos. Se comparten recursos pero sólo en forma de comunicación.
- Un sistema distribuido es un sistema expandido en toda la red, pero visto como un solo sistema. Los procesos pueden suceder en forma local o remota sin que el usuario se de cuenta. La tolerancia a fallas es más alta. Las decisiones y los recursos son administrados en forma global.

ARPANET

- **Esquema básico de Arpanet en 1969**

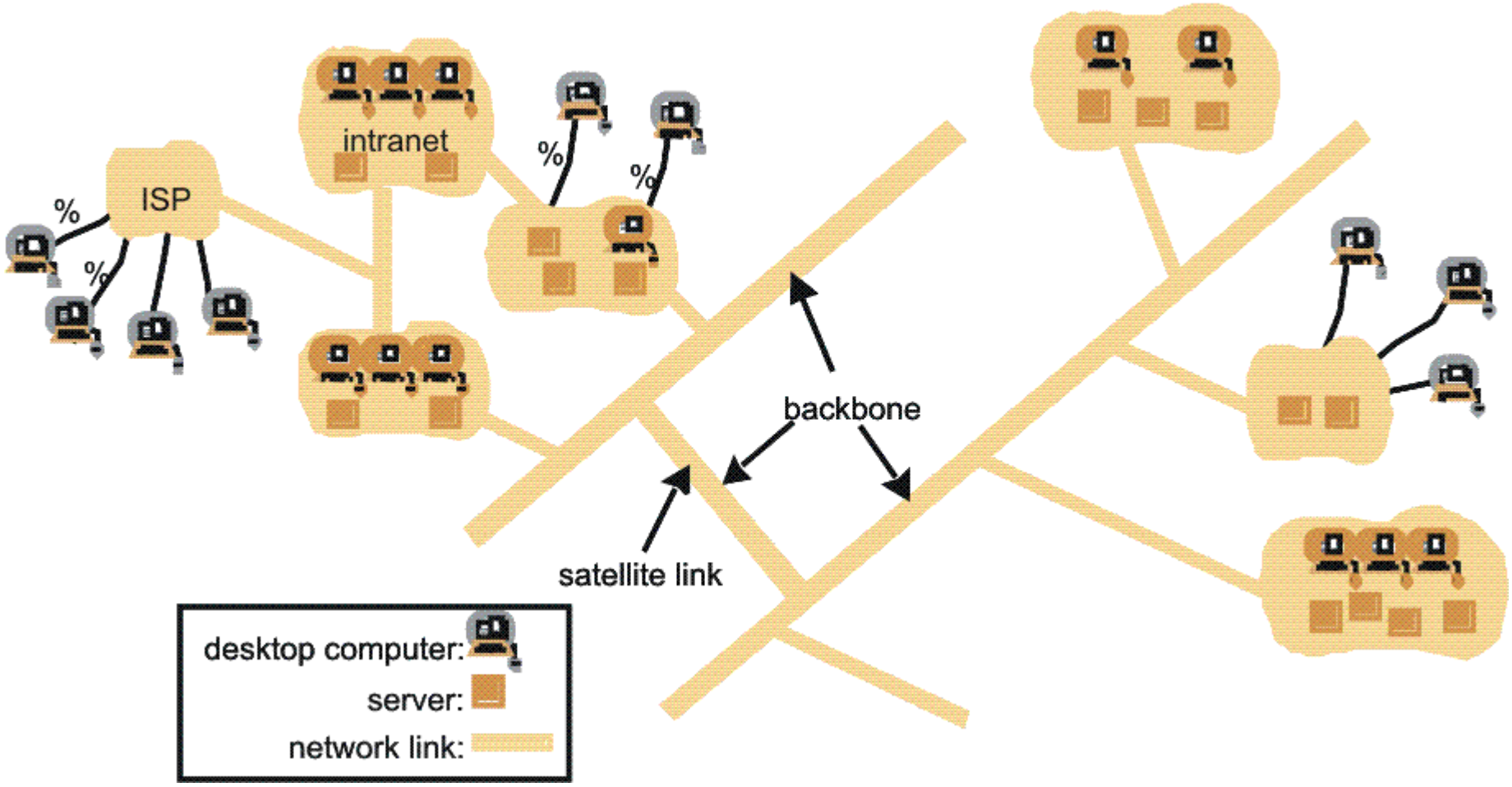


- Arpanet http://www.techweb.com/encyclopedia/ - septiembre 2002


DNS
Servidor
Usuario
Dispositivo móvil
Usuario
ISP
Ruteador
Internet
Internet
Ruteador
Receptor de dispositivos móviles
Modem
Página Web
PC
Internet
Base de Datos
Modem
Ruteador
Usuario
Impresora
Tecnología Web y Redes
Firewall
Servidor
Servidor Web

Ejemplos de SD´s

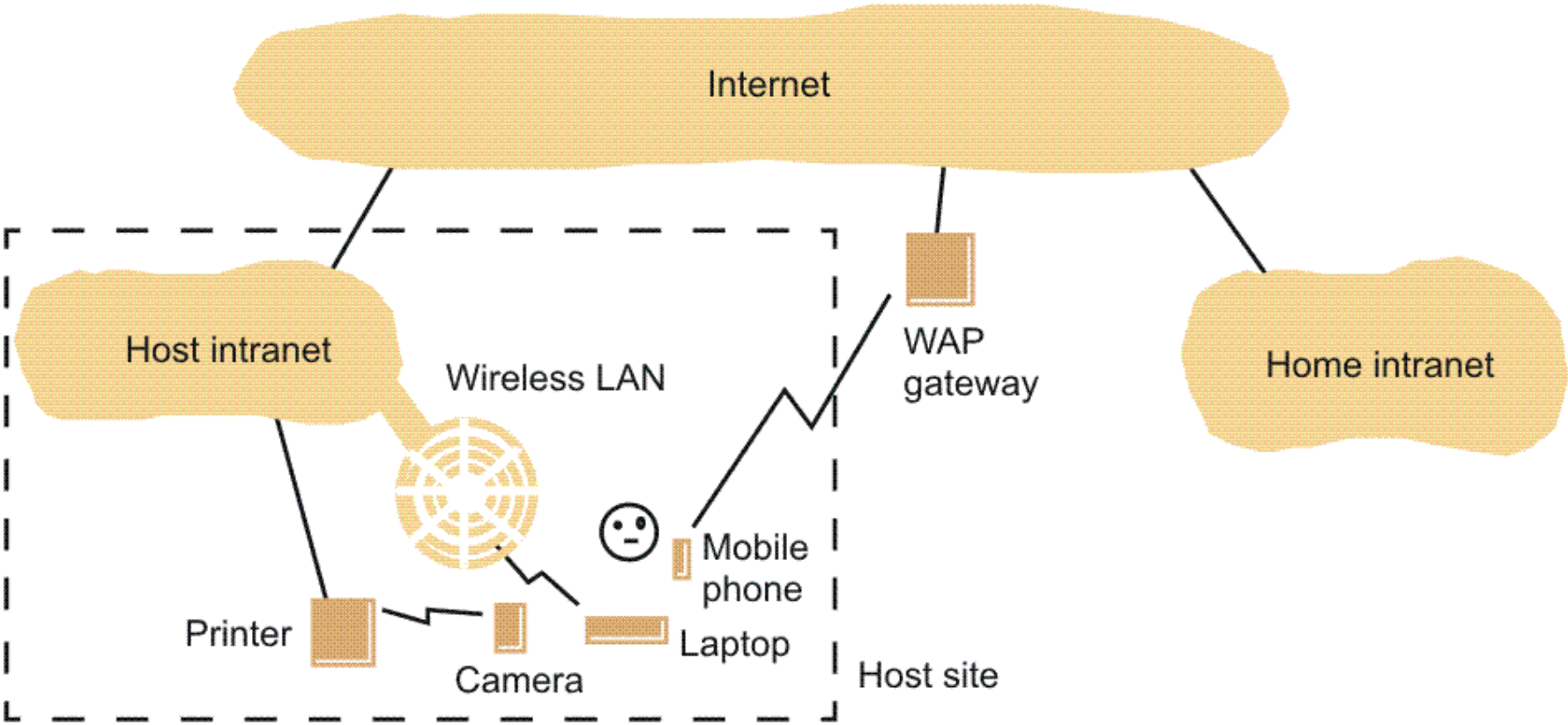
Internet



- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

Ejemplos de SD´s

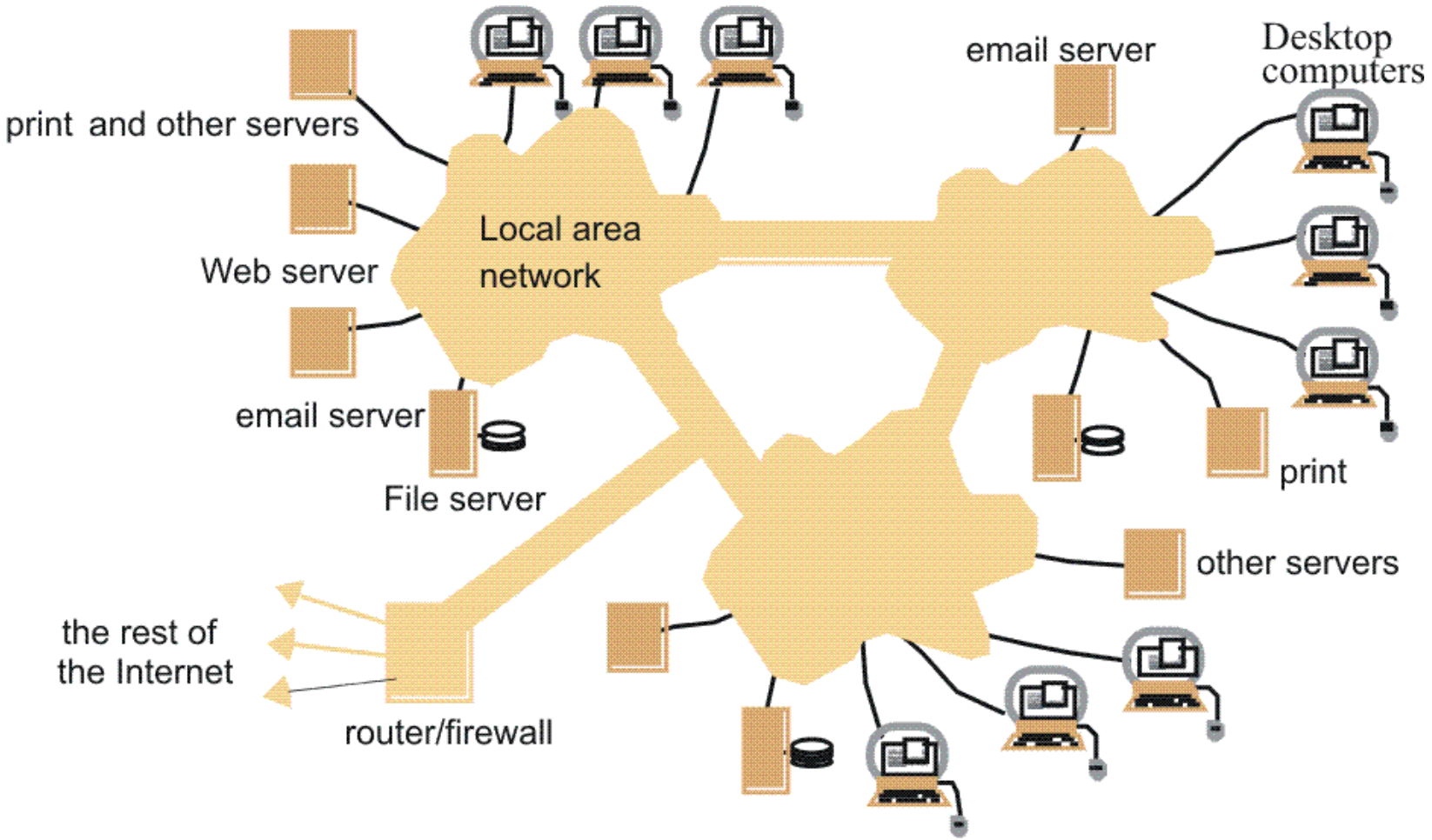
Dispositivos Móviles y SD´s



- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

Ejemplos de SD´s

Intranet Típica



- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

1.2. Objetivos de los sistemas distribuidos

Sistemas Distribuidos - Objetivos

- Compartir información generada entre diferentes estaciones de trabajo.
- Economizar el rendimiento en cuanto a respuesta de procesamiento, utilizando múltiples computadoras de rendimiento regular en vez de una sola computadora más poderosa que pueda quedar obsoleta rápidamente.
- Capacidad de expansión en cuanto a procesamiento y almacenamiento.
- Mantener un sistema disponible constantemente tolerante a fallas, en vez de mantener una sola computadora en donde se nos puede caer el sistema.
- Crear un sistema de información más confiable en forma global.

1.3. Ventajas y desventajas de los sistemas distribuidos

Sistemas Distribuidos - Ventajas

- **Compartir información entre más de un usuario en el mismo momento en que se genera.**
- **Compartir dispositivos periféricos en forma transparente.**
- **Distribución de la carga de trabajo entre las diferentes computadoras de la red.**
- **Aún cuando alguno de los nodos falle, el sistema sigue funcionando.**

Sistemas Distribuidos - Desventajas

- Son demasiado complicados en su construcción, aún en la actualidad no se ha llegado a construir un sistema distribuido totalmente eficiente.
- La tecnología de los sistemas distribuidos acaba de surgir, por lo cual no hay estándares en cuanto a software y hardware que cumplan con las características de un sistema distribuido.
- Pérdida de información a través del conjunto de redes.
- Saturación de información debido al volumen de mensajes que se pueden manejar en un sistema distribuido.
- Vulnerabilidad de la información, ya que la información puede ser accesada por un gran número de usuarios y por lo tanto no se tiene tanta seguridad.
- Existen muchos problemas debido a las fallas en cada uno de los muchos componentes e interconexiones en un sistema distribuido. Los problemas causados por la interconexión de componentes se denominan Problemas del Sistema.

1.4. Complejidad de los sistemas distribuidos

1.5. Técnicas de construcción

Construcción de SD´s - Hardware

- Clasificación de computadoras con varios CPU´s

- La clasificación más aceptada es la de Flynn (1972), para ello se propone dos características esenciales:

- Número de flujo de instrucciones.
- Número de flujo de datos.

- A partir de esta clasificación surgen 4 clasificaciones
- SISD Single Instruction Single Data – Una instrucción un dato, todas las computadoras con un procesador.
- SIMD Single Instruction Multiple Data - Una instrucción múltiples datos, máquinas paralelas.
- MIMD Multiple Instruction Multiple Data – Múltiples instrucciones múltiples datos, sistemas distribuidos.
- MISD No existen.

Construcción de SD´s - Hardware

Los MIMD a su vez se dividen en 2 categorías:

- Multiprocesadores: Un solo espacio de direcciones virtuales compartidos con varios CPU´s.
- Multicomputadoras: Computadoras con CPU´s y memorias propias.

A su vez, cada una de las anteriores se dividen en 2:

- Tecnología de Bus: Una sola red, un solo cableado.
- Tecnología de Conmutador: Diferentes tipos de cableado comunicados por conmutadores.

Construcción de SD´s - Hardware

Combinando las 2 categorías anteriores tenemos 3 divisiones:

- Multiprocesadores con Bus: Un solo bus, memoria común, sobrecarga de información.
- Multiprocesadores con conmutador: Varios procesadores comunicados entre sí por conmutadores.
- Multicomputadoras con bus: Sistemas LAN´s.

Construcción de SD´s - Hardware

- Hardware fuertemente acoplado: Retraso corto, tasa de transmisión de datos alta.
- Hardware débilmente acoplado: Retraso alto, tasa de transmisión de datos baja.
- Software débilmente acoplado en hardware débilmente acoplado: LAN en que cada usuario cuenta con su propia estación de trabajo y su propio sistema operativo.
- Software fuertemente acoplado en hardware débilmente acoplado: La red funciona como un solo sistema. Sistemas distribuidos.
- Software fuertemente acoplado en hardware fuertemente acoplado: Servidores de bases de datos.

Consejos de Construcción de SD´s

- **Duplicar la información para aumentar la disponibilidad.**
- **Usar copias locales de la información para permitir una operación autónoma.**
- **Explotar el estado local con caché.**
- **Usar tiempos de espera para revocar.**
- **Usar mecanismos estándares para llamadas remotas.**
- **Utilizar técnicas de criptografía para la autentificación y seguridad de la información.**

1.6. Requerimientos de aplicación

Aspectos de Diseño

Disponibilidad y funcionalidad

- **Disponibilidad de utilizar diferentes nodos de procesamiento y no dejar de funcionar aún cuando existan fallas.**

Transparencia

- **El sistema es transparente para el usuario.**
- **Transparencia de localización.**
- **Transparencia de réplica.**
- **Transparencia de migración.**
- **Transparente a la concurrencia.**

Seguridad

- **Contar con diferentes niveles de seguridad, tanto en aspectos físicos (disponibilidad de recursos) como de software (protección de datos con algoritmos de criptografía).**

Aspectos de Diseño

Desempeño y crecimiento modular

- **Contar con aplicaciones que puedan ser divididas en varios hilos de ejecución en paralelo y tener la capacidad de poder agregar más CPU´s también en paralelo.**

Tiempo de respuesta limitado

- **Poder ejecutar rutinas en tiempo real (tiempo mínimo de respuesta a una petición).**

Control autónomo

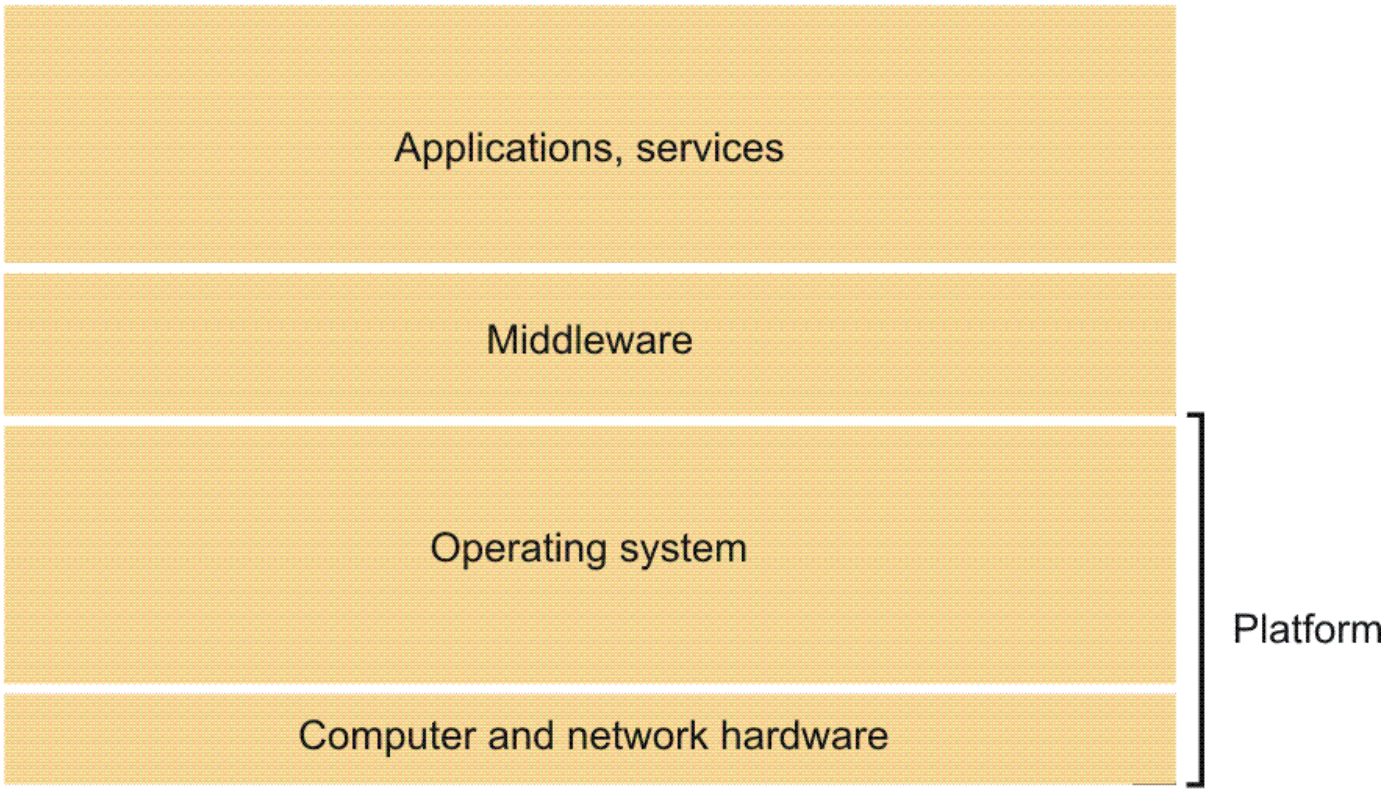
- **Capacidad de que los nodos trabajen en forma individual y a la vez interactuar con los diferentes nodos del sistema.**

Compartir recursos físicamente separados

- **Compartir dispositivos remotos (información, impresoras, etc.).**

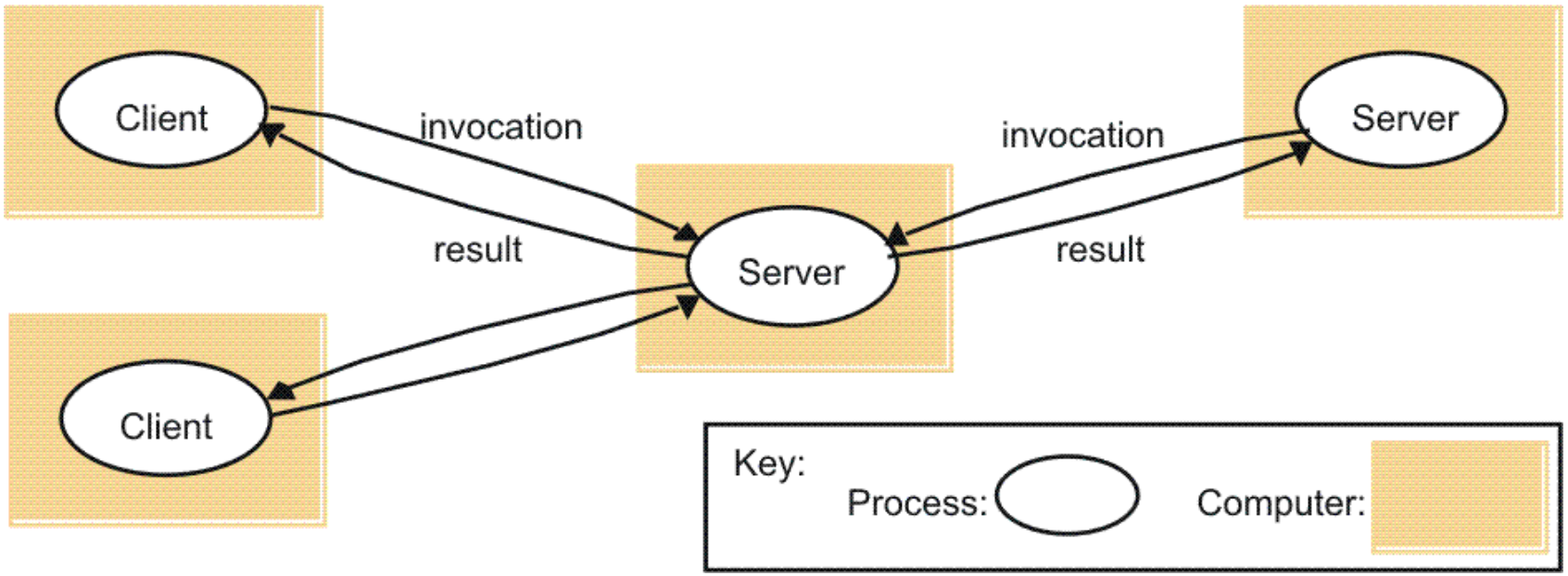
1.7. Arquitectura básica

Capas de Hardware y Software



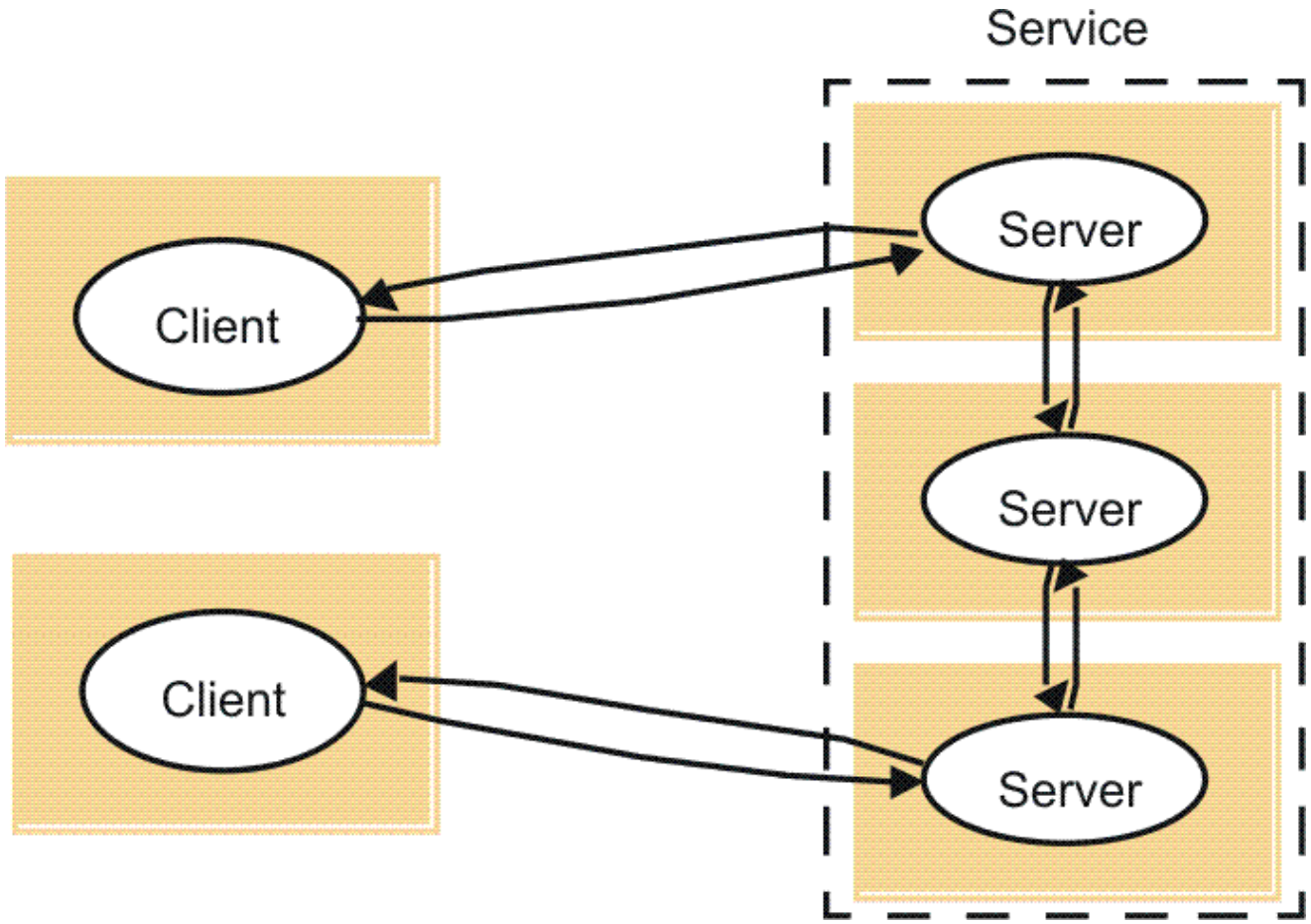
- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

Clientes y Servidores



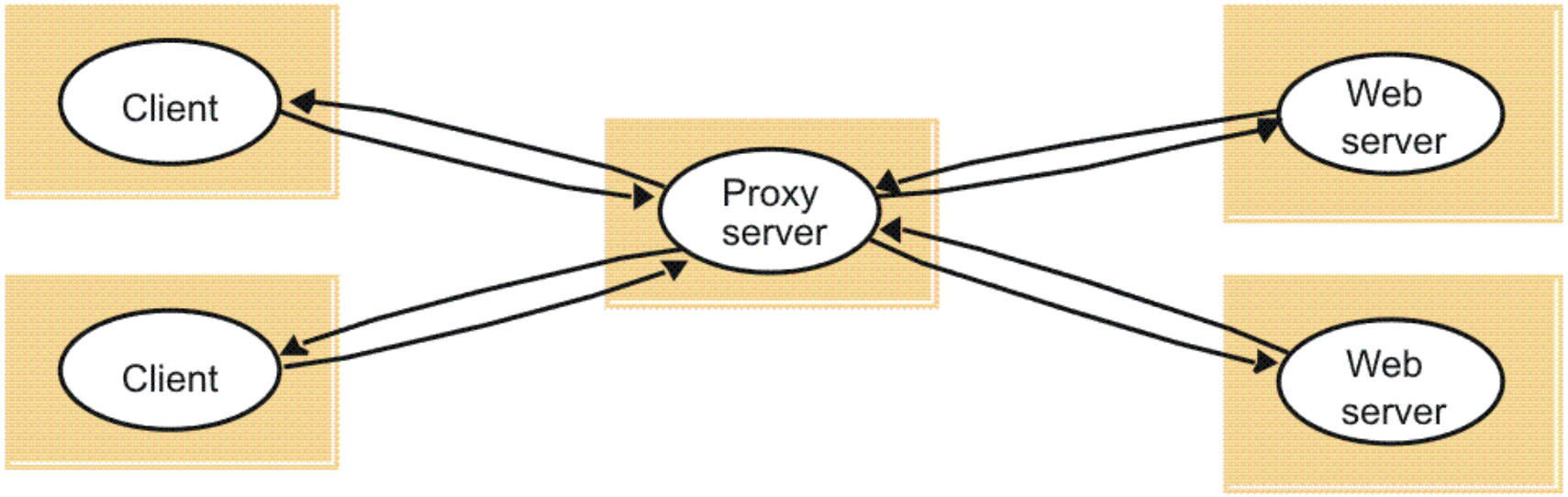
- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

Clientes y Servidores



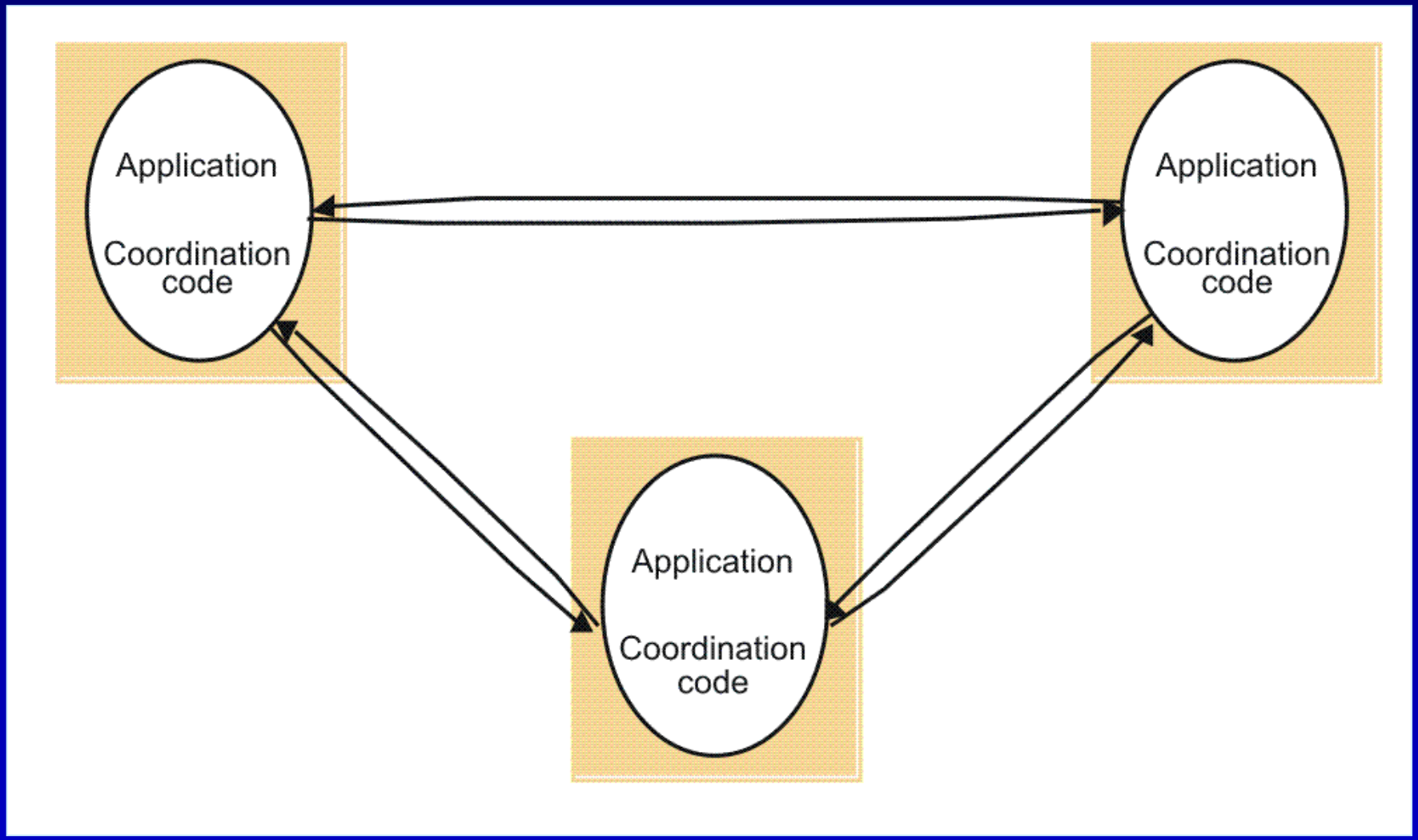
- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

Servidores Proxy Web



- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

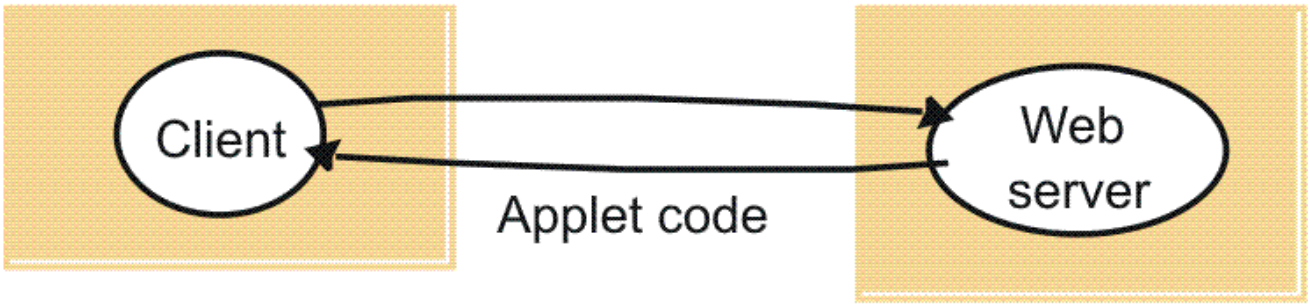
Aplicaciones Distribuidas



- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

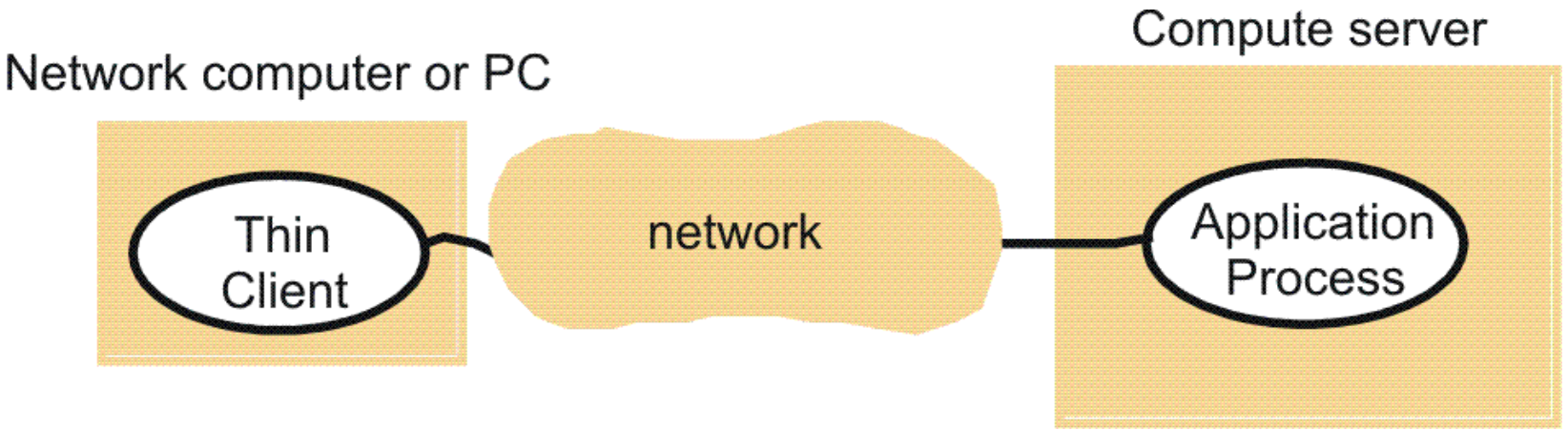
Applets

a) client request results in the downloading of applet code



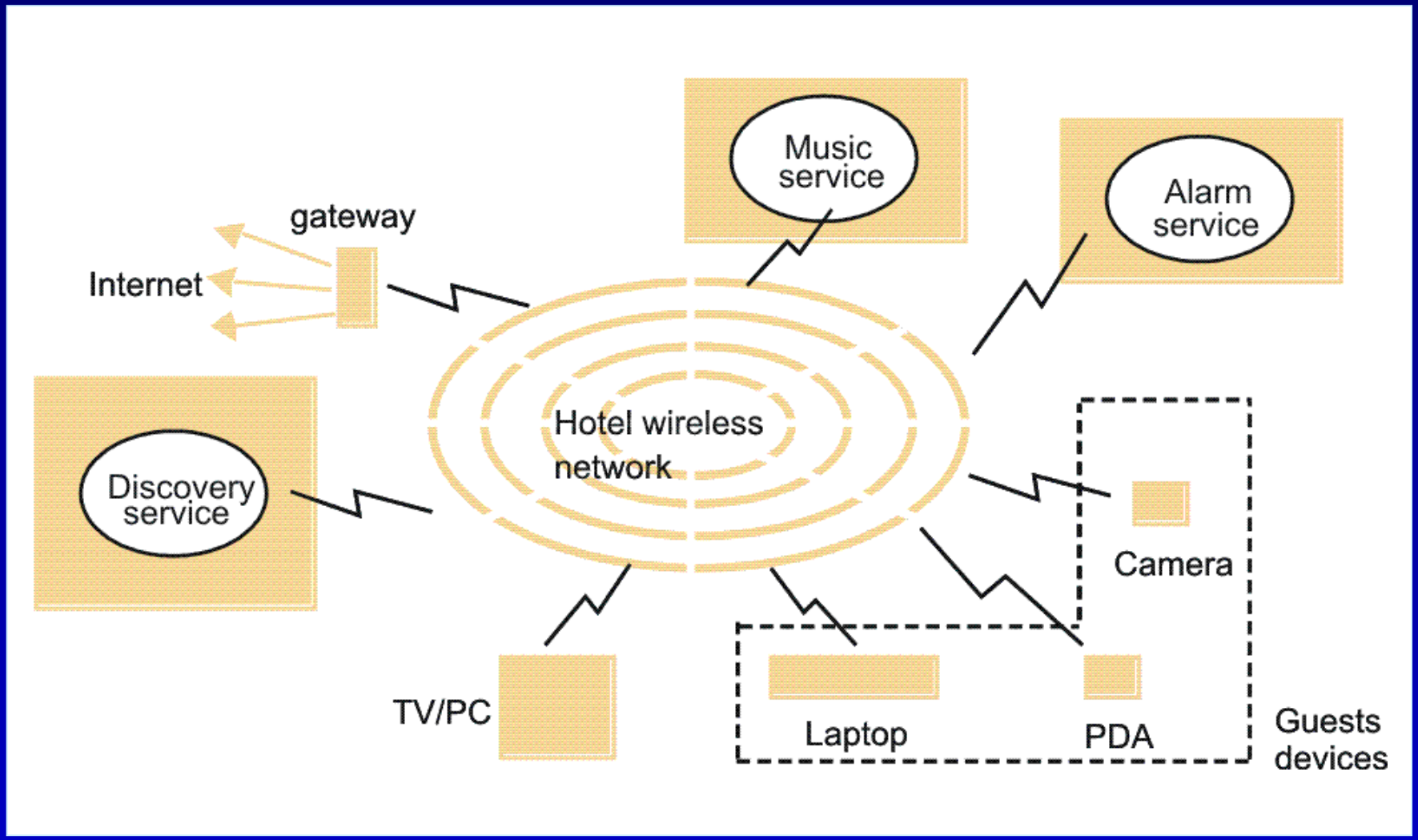
b) client interacts with the applet

Clientes Delgados



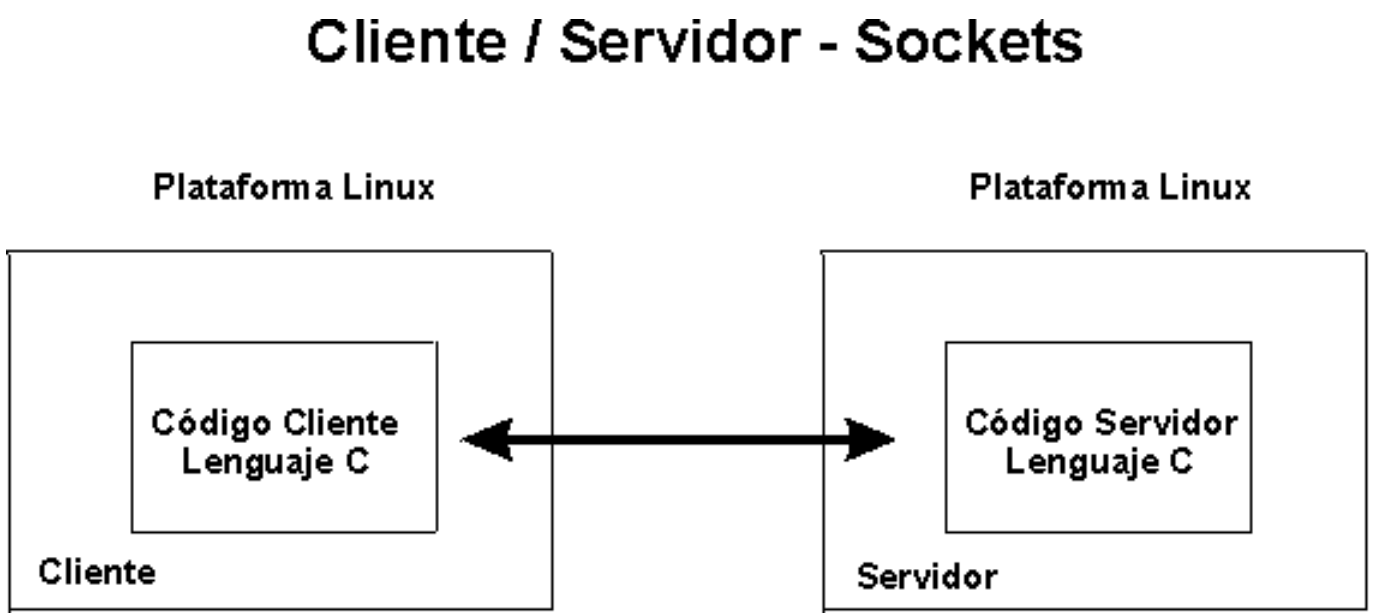
- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

Aplicaciones Inalámbricas

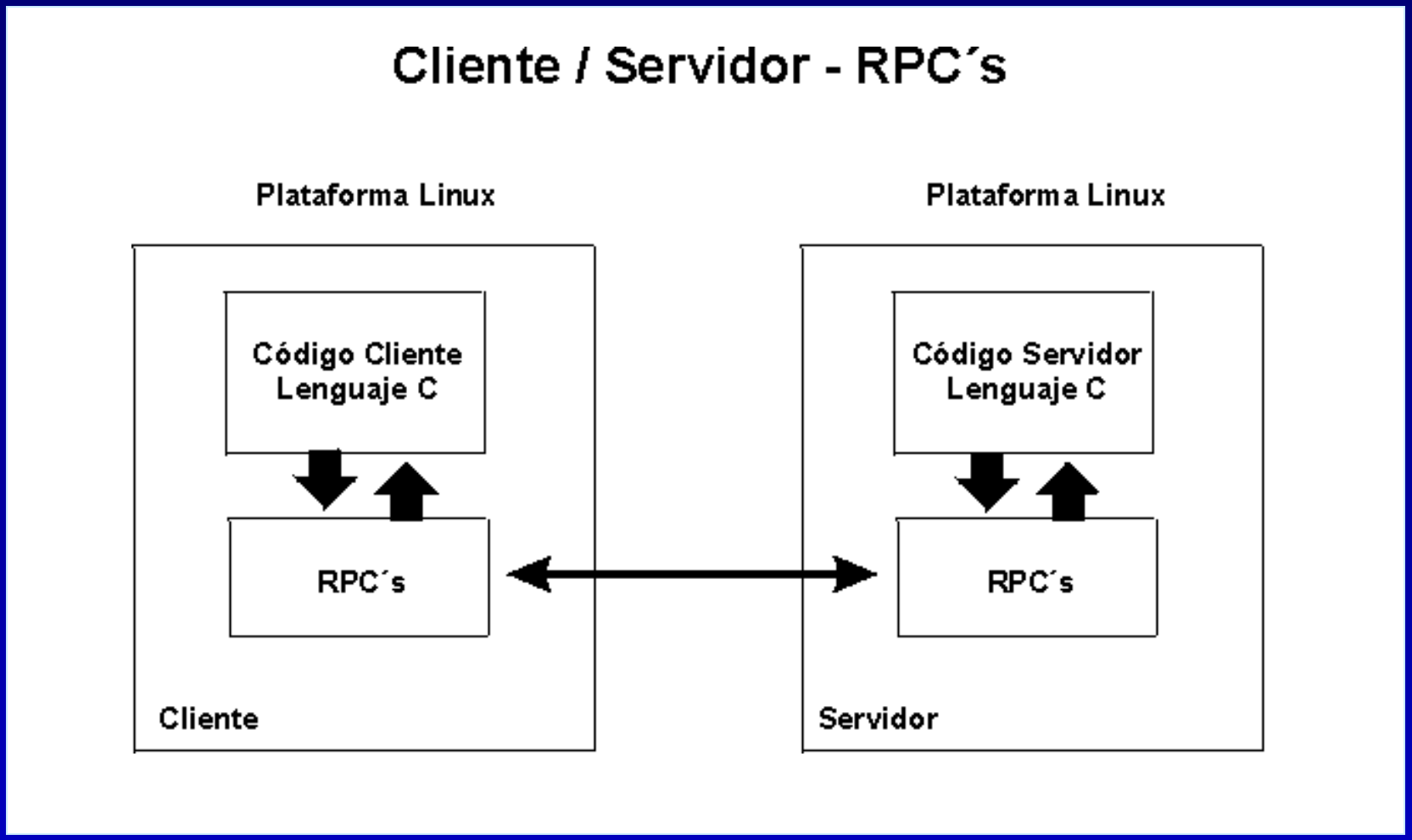


- Distributed Systems: Concepts and Design, Coulouris, Dollimore and Kindberg, Third Edition, Addison Wesley 2001

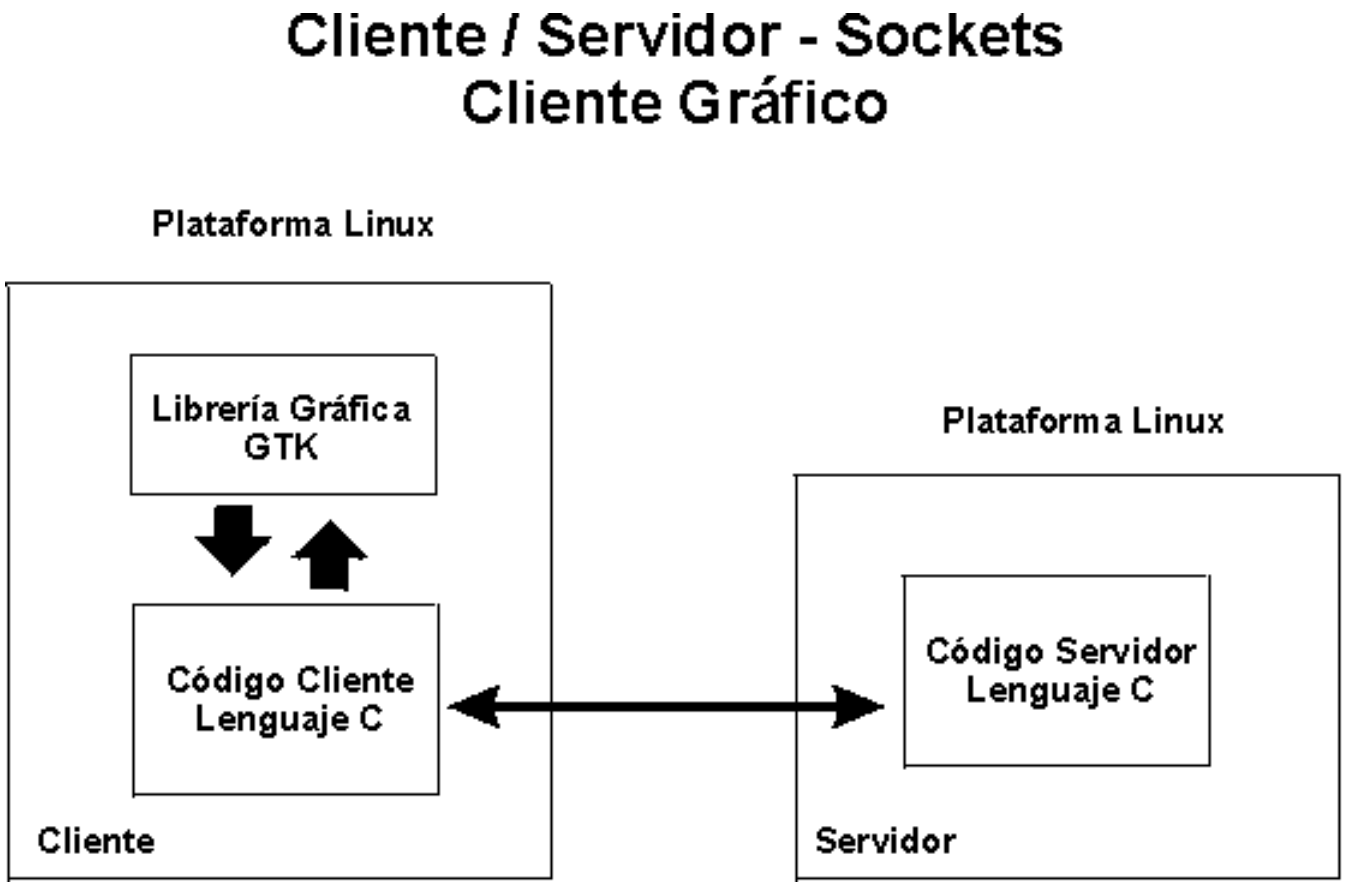
1) Sistemas Distribuidos en el ITM



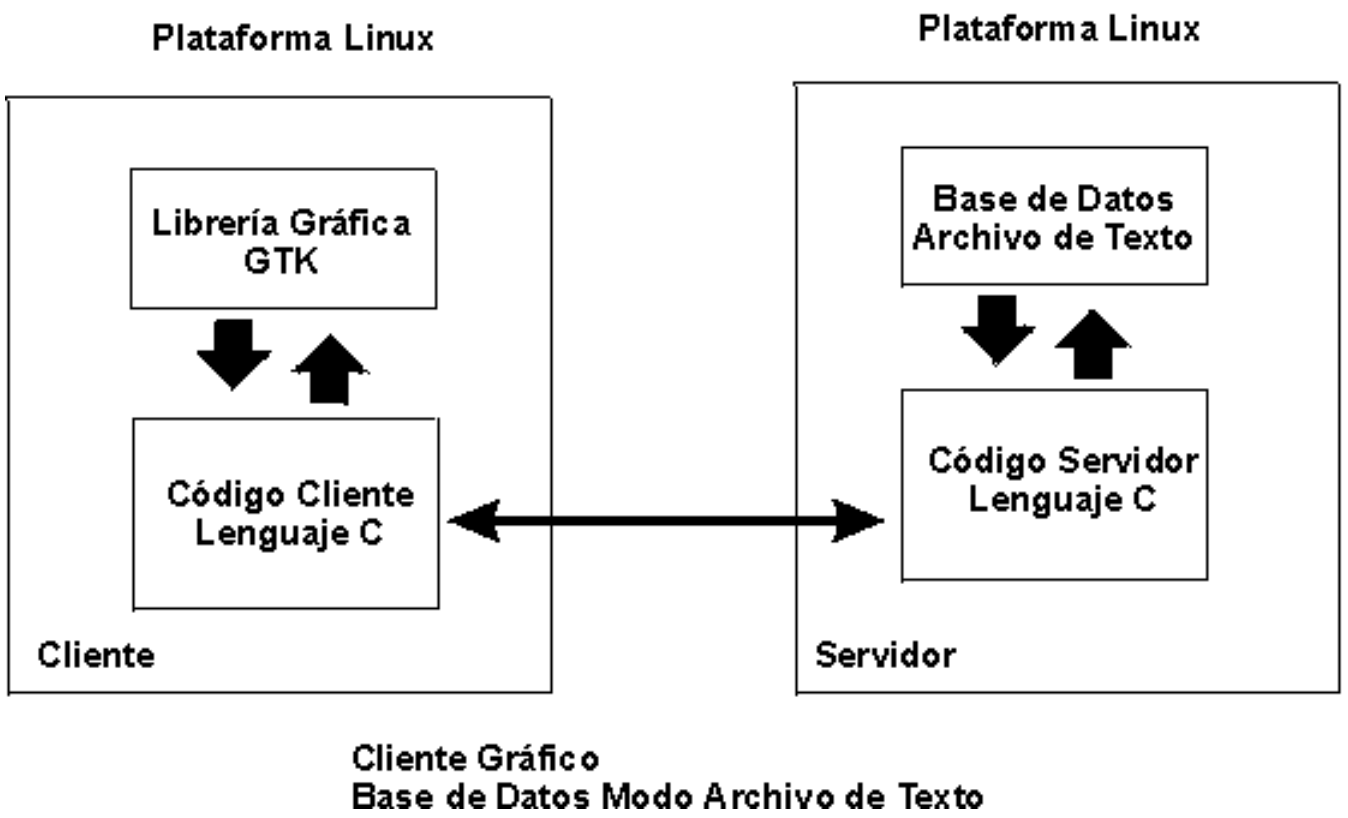
2) Sistemas Distribuidos en el ITM



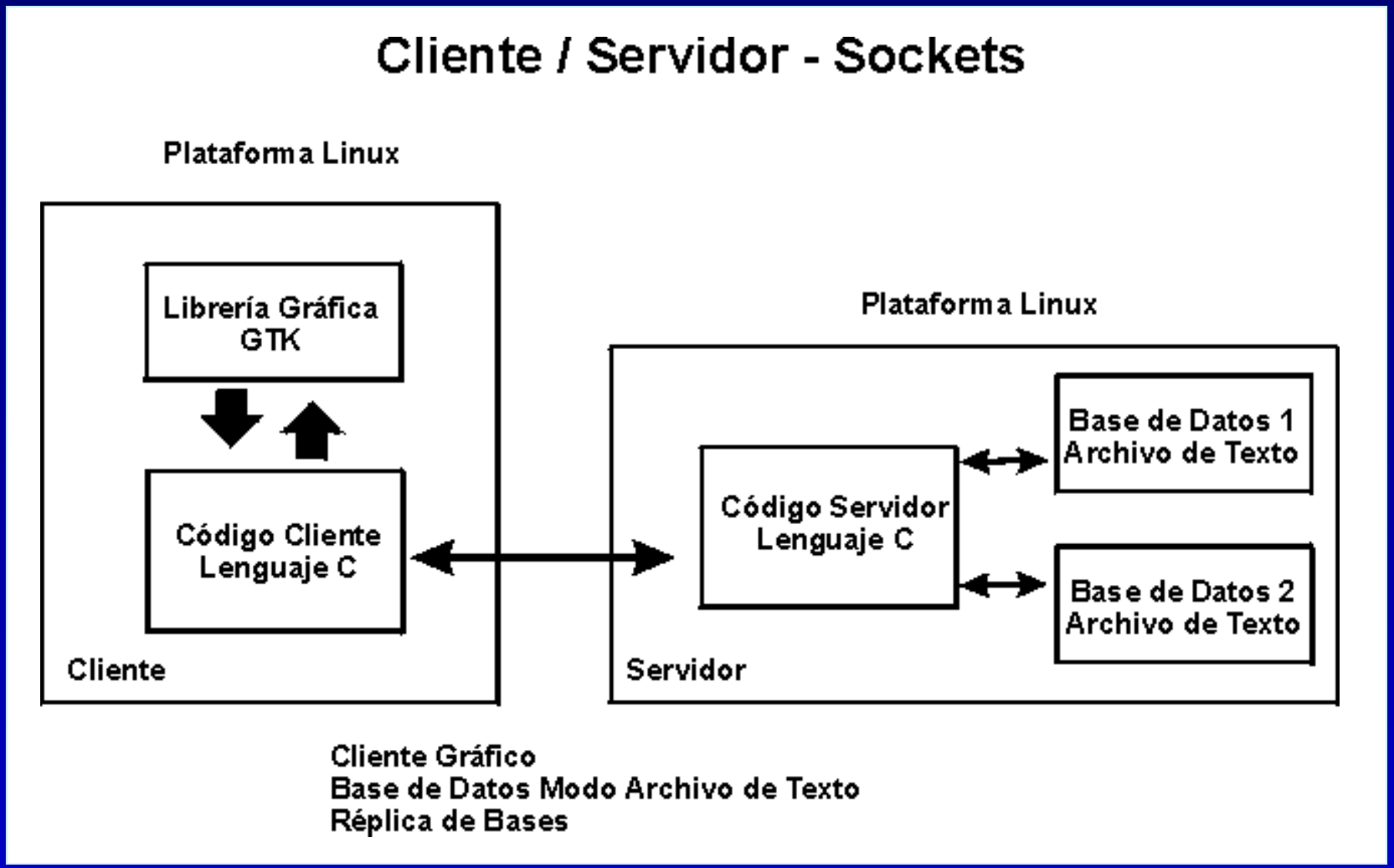
3) Sistemas Distribuidos en el ITM



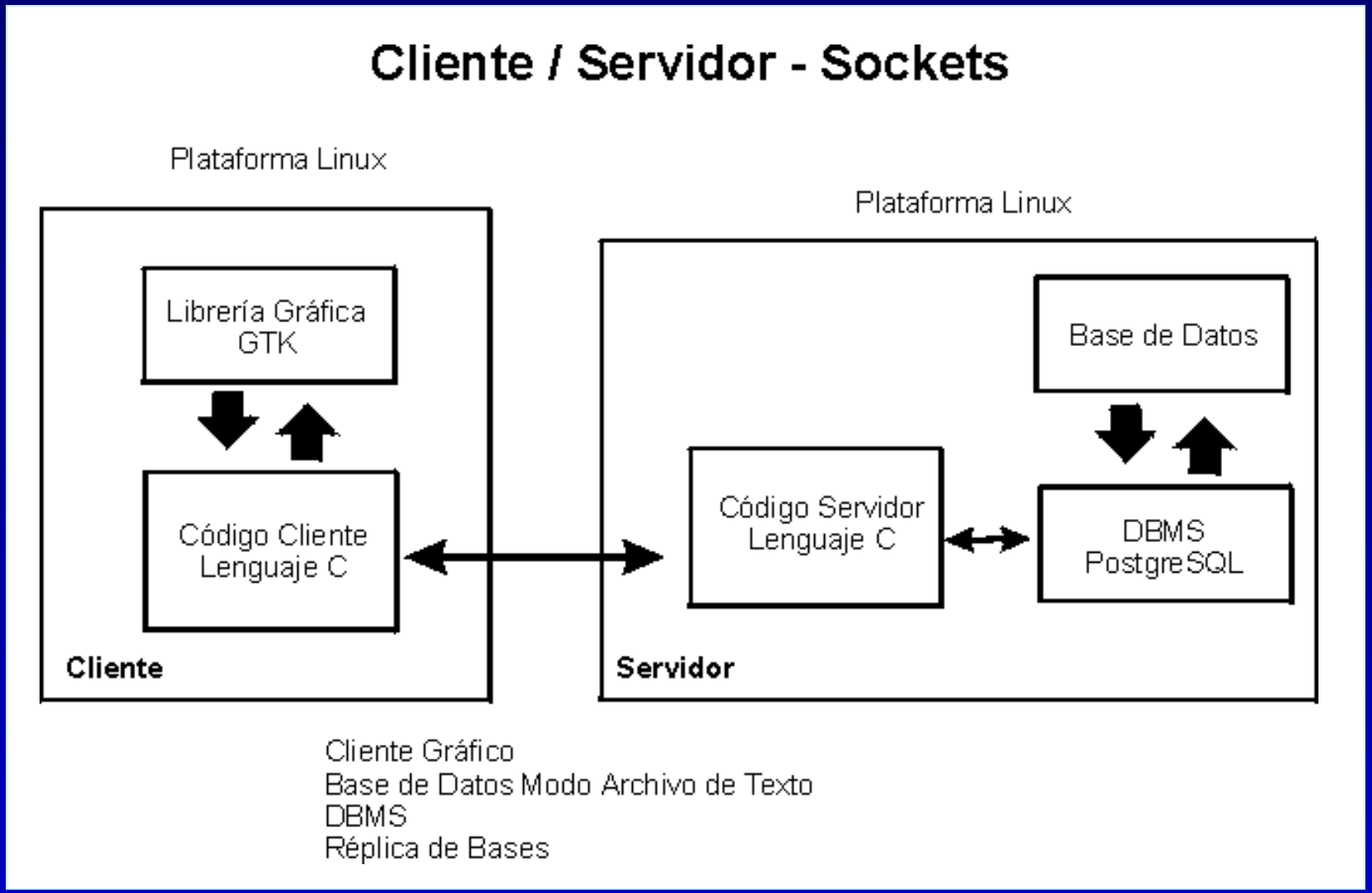
4) Sistemas Distribuidos en el ITM



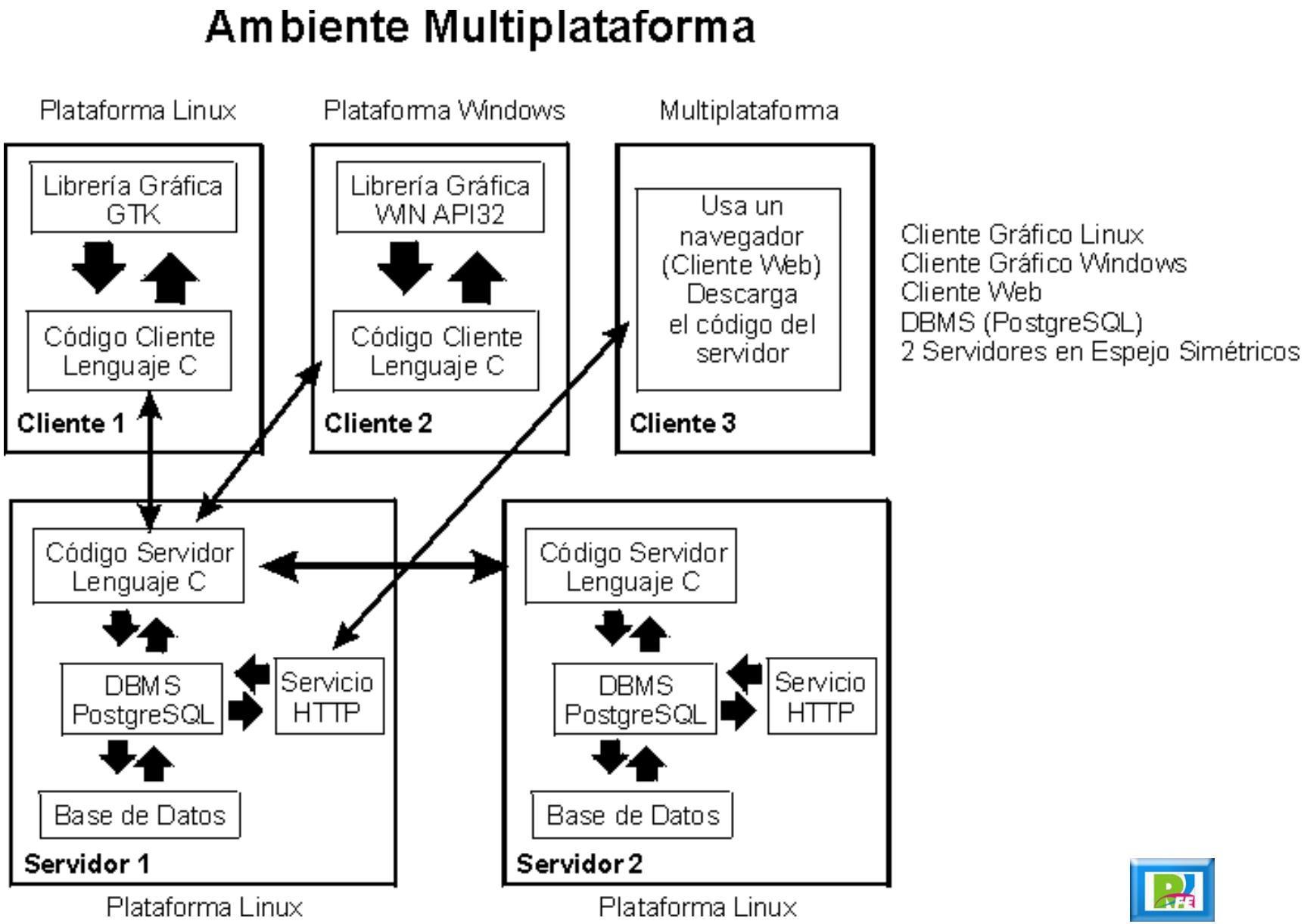
5) Sistemas Distribuidos en el ITM



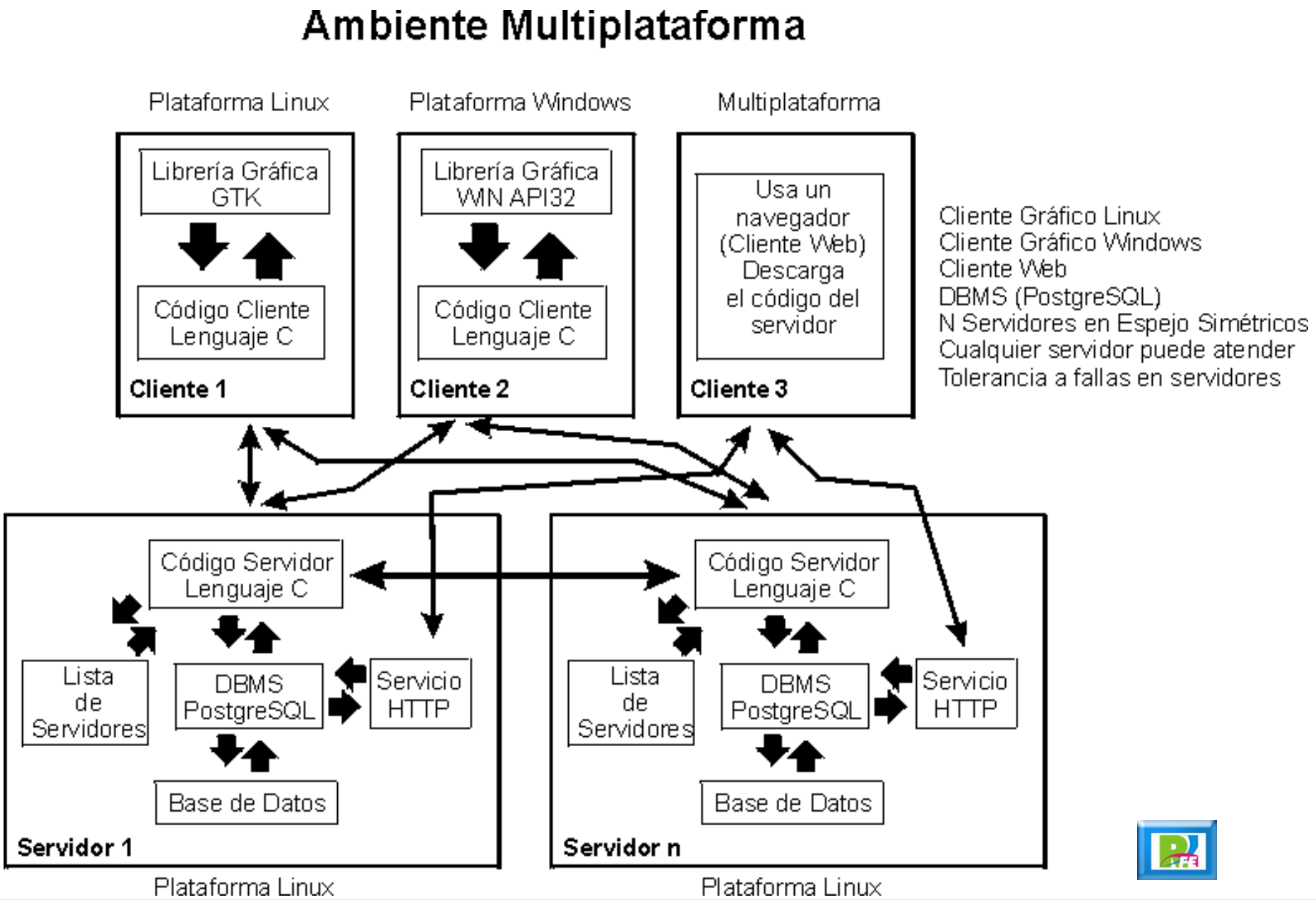
6) Sistemas Distribuidos en el ITM



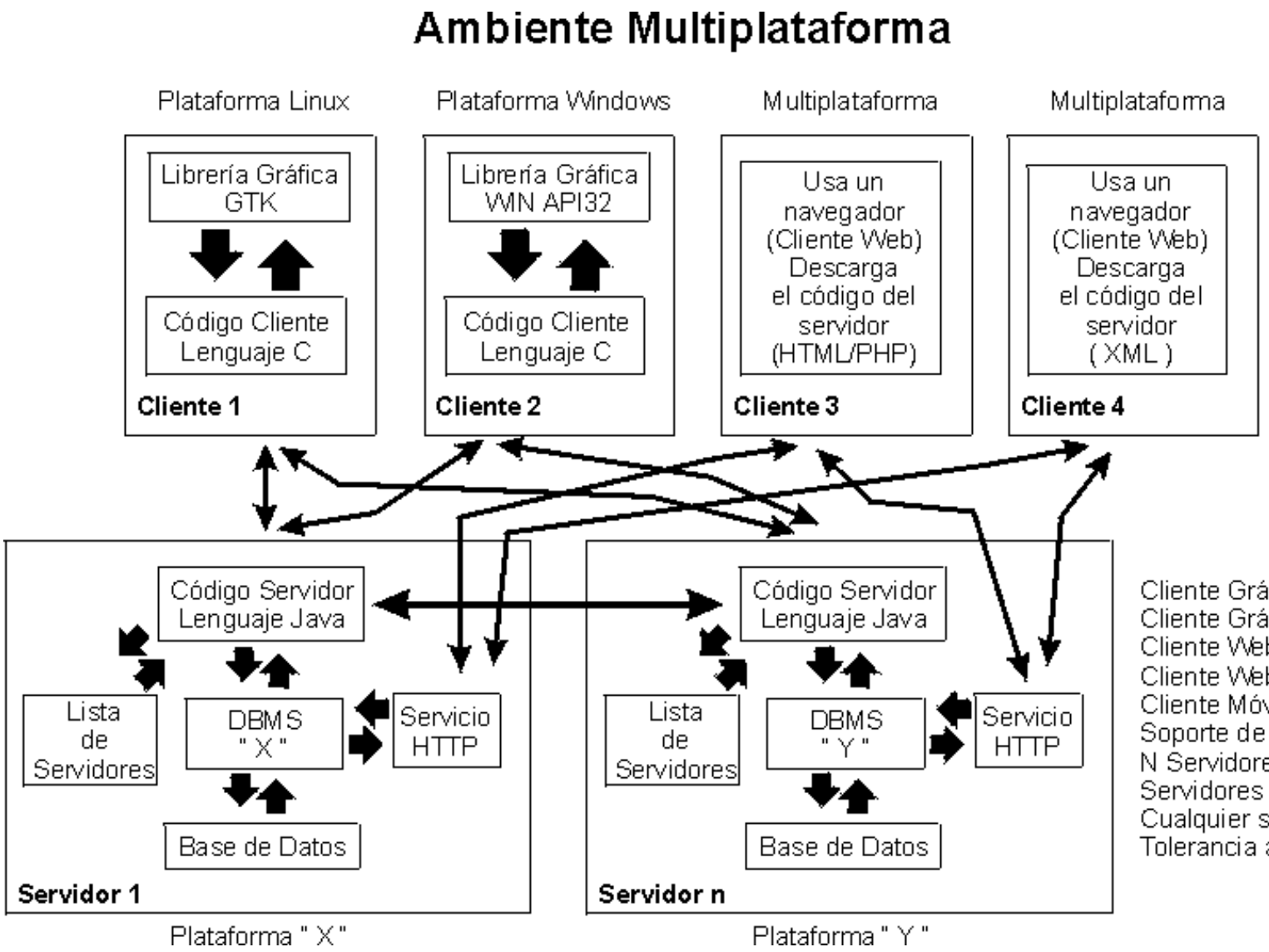
7) Sistemas Distribuidos en el ITM



8) Sistemas Distribuidos en el ITM

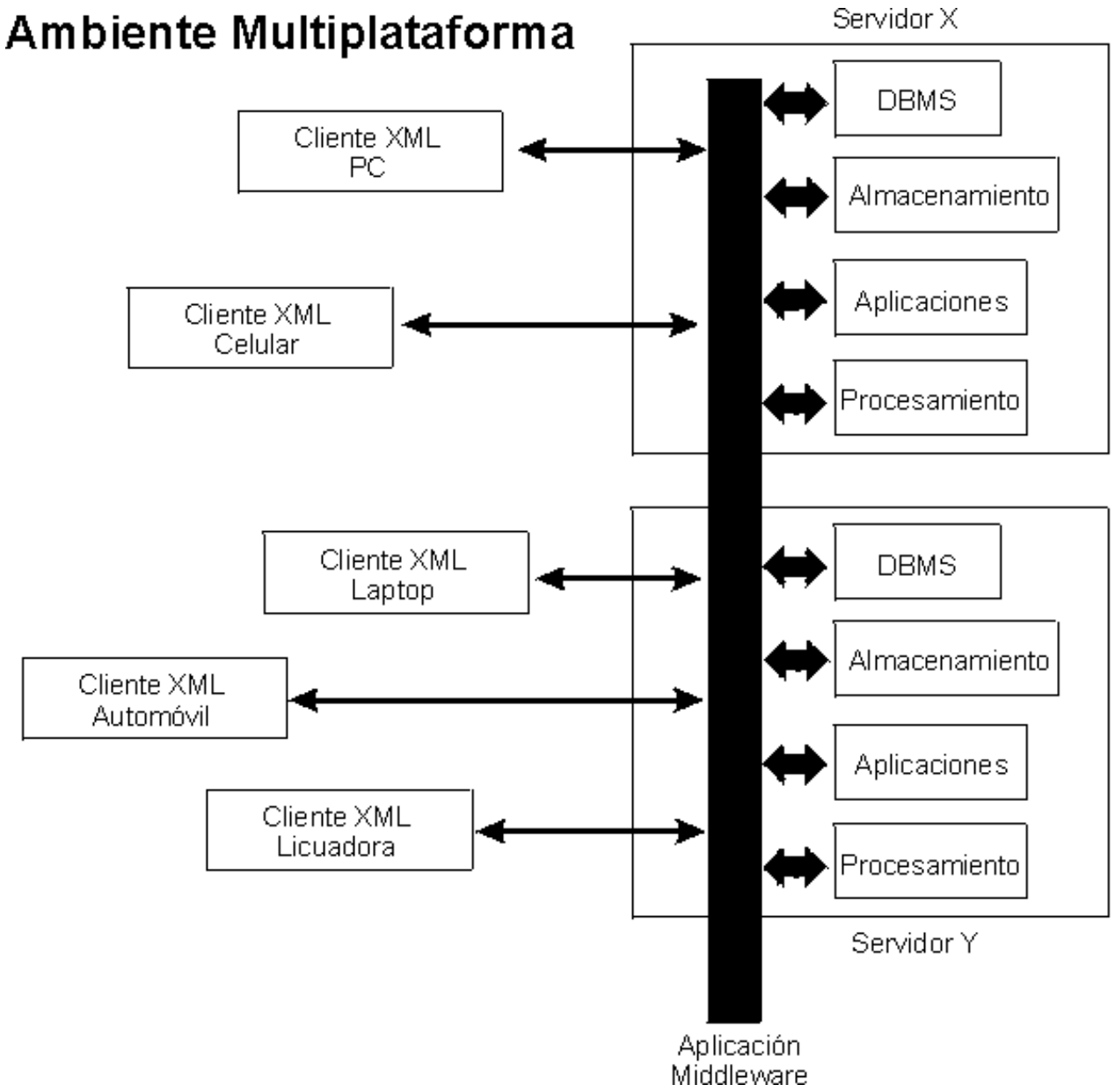


9) Sistemas Distribuidos en el ITM



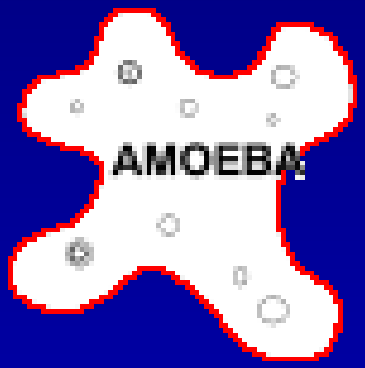
Cliente Gráfico Linux
Cliente Gráfico Windows
Cliente Web HTML / PHP
Cliente Web XML
Cliente Móvil XML
Soporte de varios DBMS
N Servidores en Espejo Simétricos
Servidores Multiplataforma
Cualquier servidor puede atender
Tolerancia a fallas en servidores

10) Sistemas Distribuidos en el ITM



1.8. Sistemas operativos distribuidos

Amoeba



Amoeba

Historia:

- **El desarrollo de Amoeba inició en 1981 en Vrije Universiteit en Amsterdam Holanda como un proyecto de Cómputo Paralelo y Distribuido.**
- **Fue diseñado inicialmente por Andrew Tanenbaum y 3 estudiantes de doctorado, Frans Kaashoek, Sape J. Mullender y Robert Van Renesse.**
- **EN 1983 se logra Amoeba 1.0 como un prototipo, pero ya tenía un nivel operacional.**

- Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Amoeba

Características:

- **Inició desde cero sin preocuparse por la compatibilidad con otros sistemas operativos.**
- **El objetivo era crear un sistema operativo distribuido transparente.**
- **En Amoeba no hay máquina origen y destino, es decir cliente servidor, todas las máquinas hacen un todo.**
- **Las máquinas no tienen propietario.**
- **Cada nuevo proceso es ejecutado por la computadora con menor carga (balanceo de carga).**
- **Amoeba esta escrito en lenguaje C.**

- Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Amoeba

Características:

- **El sistema se diseñó pensando en implementarse en un ambiente con gran número de CPU´s cada uno con gran cantidad de memoria.**
- **El sistema no se basaba en memoria compartida.**
- **Puede utilizar CPU´s 680x0, 386 ó SPARC.**
- **Amoeba esta formado por un micronúcleo que es ejecutado en todas las computadoras.**

- Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Amoeba

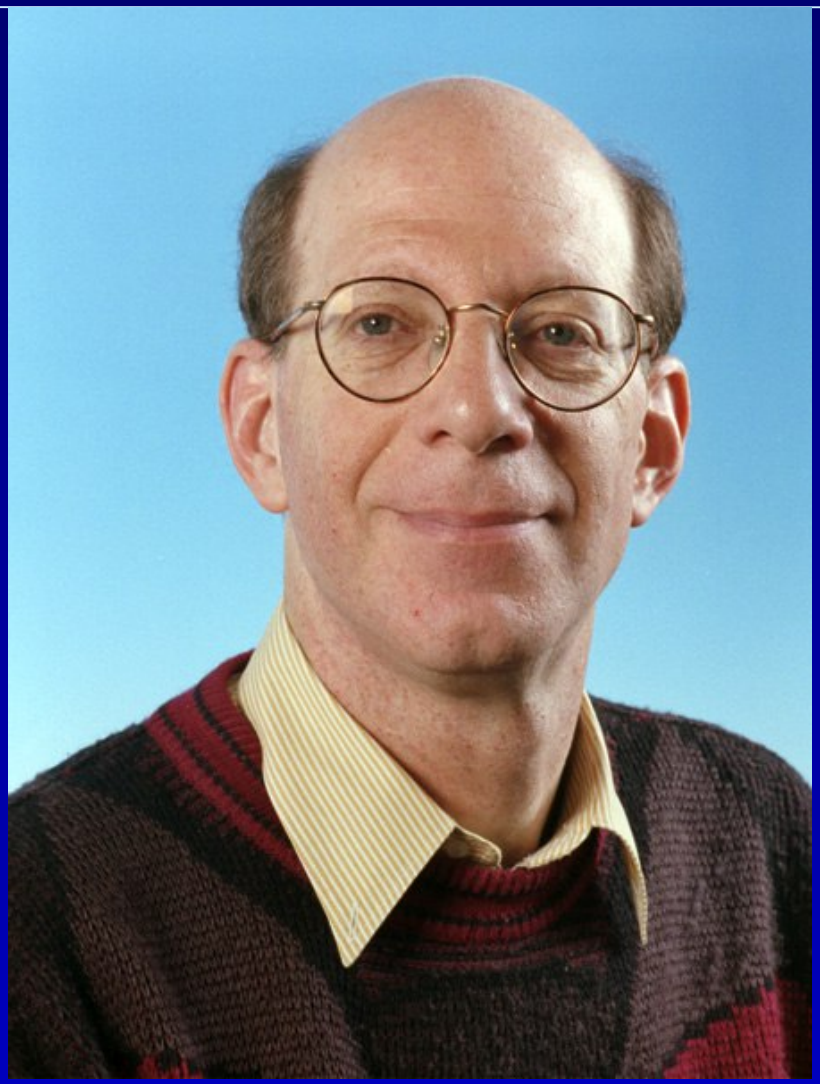
El micronúcleo se encarga de realizar las siguientes operaciones:

- **1) Controlar procesos e hilos.**
- **2) Proporcionar el soporte de la administración de memoria de bajo nivel.**
- **3) Soportar la comunicación.**
- **4) Controlar la Entrada/Salida de bajo nivel.**

- Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Amoeba

- **Andrew Tanenbaum, creador de Amoeba.**
- **Actualmente es profesor de la División de Matemáticas y Ciencias de la Computación en Vrije Universiteit, en Amsterdam Holanda.**



- Andrew Tanenbaum http://www.cs.vu.nl/~ast/photos/wi/Tanenbaum_Andy.jpg mayo 2003

Amoeba

- **Colección de 80 computadoras SPARC conectadas por medio de ethernet en Vrije Universiteit, Amsterdam Holanda, corriendo Amoeba.**

- The Amoeba Distributed Operating System http://www.cs.vu.nl/pub/amoeba/amoeba.html mayo 2003

MACH

MACH

Historia:

- **El desarrollo de MACH inició en 1984 por Richard Rashid en Carnegie Mellon University, USA, y en 1986 apareció la primera versión para una computadora con 4 cpu´s, la VAX 11/784.**
- **Debido a DARPA se aisgnaron recursos para mejorar MACH y modificaron la versión 4.1 BSD (el UNIX de la Universidad de Berkeley) para incorporar el código de MACH.**
- **Posteriormente se combinó MACH con las versiones 4.2 y 4.3 de BSD lo cual permitió una completa compatibilidad de MACH para poder correr aplicaciones UNIX.**
- **Posteriormente la OSF (Open Software Foundation) elige a MACH como su sistema operativo y lo lanza como OSF/1.**

• Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

MACH

Historia:

- **Para 1988 el núcleo de MACH 2.5 era grande y monolítico debido a la presencia de gran parte del código de UNIX BSD por lo que se decidió quitar del núcleo todo el código BSD y ponerlo en la parte del usuario, por lo que sólo quedó un micronúcleo de MACH.**

- **MACH sigue corriendo aplicaciones UNIX pero por medio de un emulador.**

• Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

MACH

Objetivos de desarrollo:

- **1) Proporcionar una base para la construcción de otros sistemas operativos (por ejemplo UNIX).**
- **2) Soporte de espacio de direcciones de gran tamaño.**
- **3) Permitir el acceso transparente a ls recursos de la red.**
- **4) Explotar el paralelismo tanto en el sistema como en las aplicaciones.**
- **5) Hacer que MACH se pueda transportar a una colección más grande de máquinas.**

- Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

MACH

Características del Micronúcleo:

- **El desarrollo del micronúcleo se realizó pensando en emular sistemas operativos como UNIX.**
- **La emulación se lleva a cabo mediante una capa de software que se ejecuta fuera del núcleo, en el espacio del usuario.**
- **Se pueden ejecutar varios emuladores al mismo tiempo, por lo que es posible ejecutar programas en 4.3BSD, UNIX System V y MS-DOS, en la misma máquina y al mismo tiempo.**

• Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

CHORUS

CHORUS

Historia:

- **Surge en 1980 en INRIA Francia y se desarrollaron sólo 4 versiones (de la 0 a la 3).**
- **Es un sistema operativo distribuido que se basa en una colección de actores. Un actor es en realidad un autómata de estado finito.**
- **Cada máquina ejecuta el mismo núcleo del sistema operativo.**
- **La versión 0 de Chorus se desarrolló en Pascal.**

- Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

CHORUS

Objetivos de desarrollo:

- **1) Emulación de UNIX de alto rendimiento.**
- **2) Uso en Sistemas Distribuidos.**
- **3) Correr aplicaciones en tiempo real.**
- **4) Integración de programas orientados a objetos.**

- Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

CHORUS

Partes de CHORUS:

1) Núcleo:

- **Se encarga de la administración de nombres, procesos, hilos, memoria y comunicación.**

2) Procesos del Núcleo:

- **Se cargan y eliminan de manera dinámica durante la ejecución del sistema.**

3) Procesos del Sistema:

- **Se ejecutan en modo usuario y junto con los procesos del núcleo forman un subsistema.**

4) Procesos del Usuario:

- **Aquí se encuentran los procesos del usuario que se encargan de llamar a los procesos del sistema.**

- Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

CHORUS

Características:

- El uso de subsistemas permite construir nuevos sistemas operativos sobre el micronúcleo de manera modular.
- Un proceso posee ciertos recursos y cuando el proceso termina se liberan sus recursos.
- Dentro de un proceso pueden existir uno o más hilos.
- Cada hilo tiene su propia pila, código y registros o datos.
- Todos los hilos de un proceso comparten el mismo espacio de direcciones.
- Los hilos son independientes entre sí.
- Los hilos de los procesos se pueden comunicar entre sí por medio de transferencia de mensajes.
- Para comunicarse se utilizan puertos.
- Cada puerto pertenece a un proceso.
- Chorus tiene un subsistema llamado Mix el cual es compatible con Unix System V.
- La versión Mix 3.2 es compatible con BSD 4.2

• Sistemas Operativos Distribuidos, Andrew S. Tanenbaum, traducción de la primera versión en inglés, Prentice Hall, Naucalpan de Juárez, México 1996 / ISBN: 84-89660-01-8

Plan 9

Plan 9

Historia:

- **A mitades de los 80´s se utilizaban grandes computadoras centralizadas conectadas a pequeñas computadoras normalmente estaciones de trabajo UNIX.**
- **UNIX es un sistema de tiempo compartido que tiene problemas con nuevos módulos que se le han integrado como los de gráficos y comunicación en red.**
- **Plan 9 se empieza a finales de los 80´s y se buscaba un sistema formado por microcomputadoras que realizaran diferentes tareas y que estuvieran conectadas a grandes servidores compartidos.**

• Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Historia:

- **Se creó un nuevo protocolo a nivel de red llamado P9 que permite a las computadoras acceder a los archivos en sitios remotos.**
- **Para 1989, Plan 9 ya era usado en diferentes partes como sistema principal, el cual ya contiene nuevos compiladores, lenguajes, librerías, sistemas de ventanas y nuevas aplicaciones.**
- **Para permitir compatibilidad con UNIX se creó un emulador que corre en una ventana, el cual permite ejecutar comandos POSIX, pero todo el sistema corre en Plan 9.**

- Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Características:

- **Los recursos tienen nombres y son accesados como archivos en orden jerárquico.**
- **Para nombrar recursos se tienen espacios locales de nombres y espacios globales de nombres donde los procesos buscan los recursos que necesitan, ya sean locales o globales.**
- **Para acceder a los recursos se crea el protocolo P9.**
- **Se tiene un número de computadoras conectadas entre sí, cada una realizando un servicio en particular.**
- **Esta formado por multiprocesadores compartidos que proveen ciclos de cómputo al sistema distribuido.**

• Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

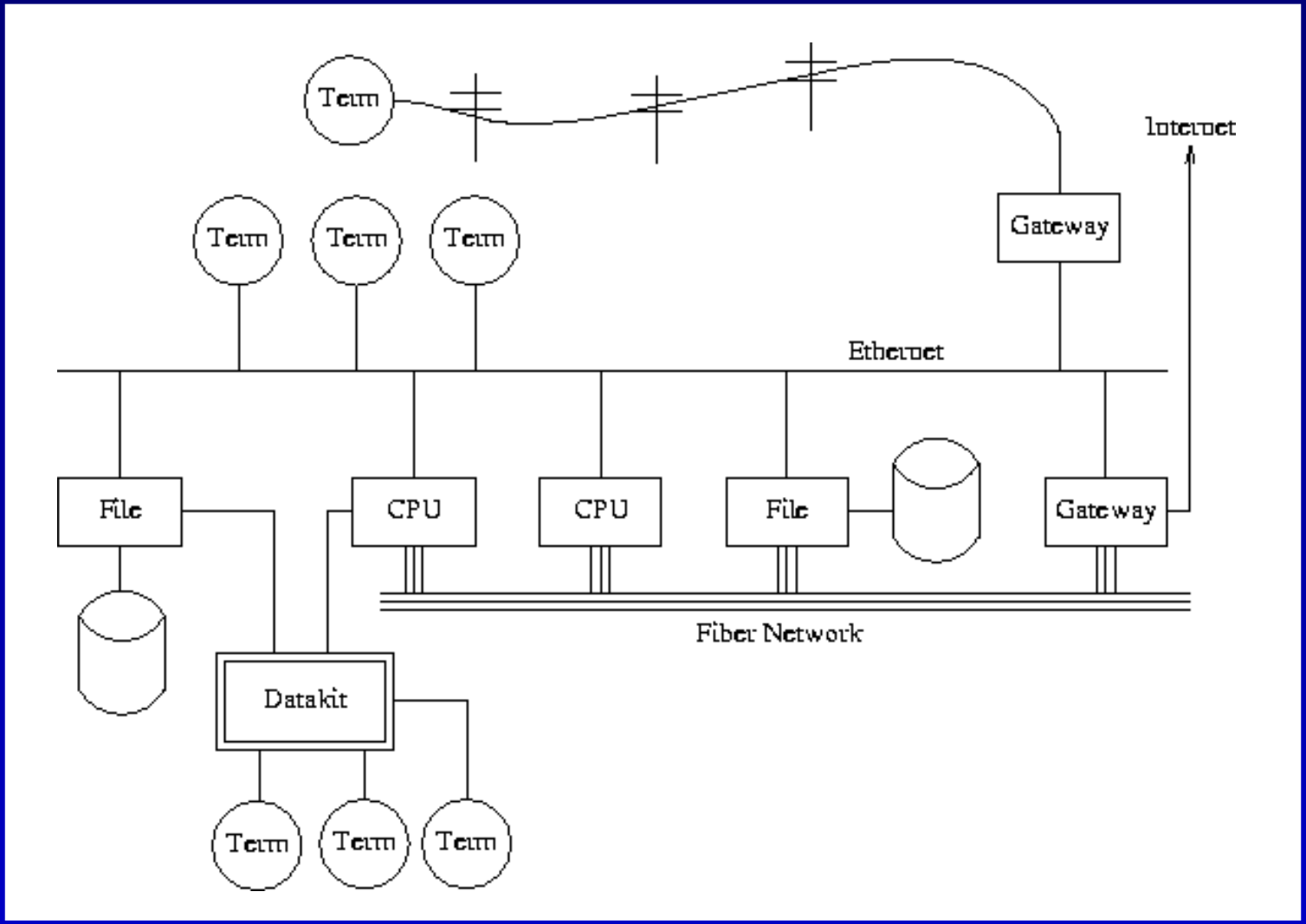
Características:

- **Algunas computadoras se dedican a almacenamiento de archivos.**
- **Estas computadoras están conectadas por una red de alto rendimiento.**
- **Los clientes o terminales del sistema (por lo general PC´s), se conectan a los servidores mediante redes de bajo rendimiento (Ethernet ó ISDN).**
- **Cuando alguien utiliza una PC como terminal de Plan 9, se crea una terminal especial (en software) para éste usuario determinado con las características de sus variables locales de entorno (indicando tipo de video) esto es para evitar configurar en forma manual cada terminal en base al hardware que tienen.**

• Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Arquitectura de Plan 9:



- Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

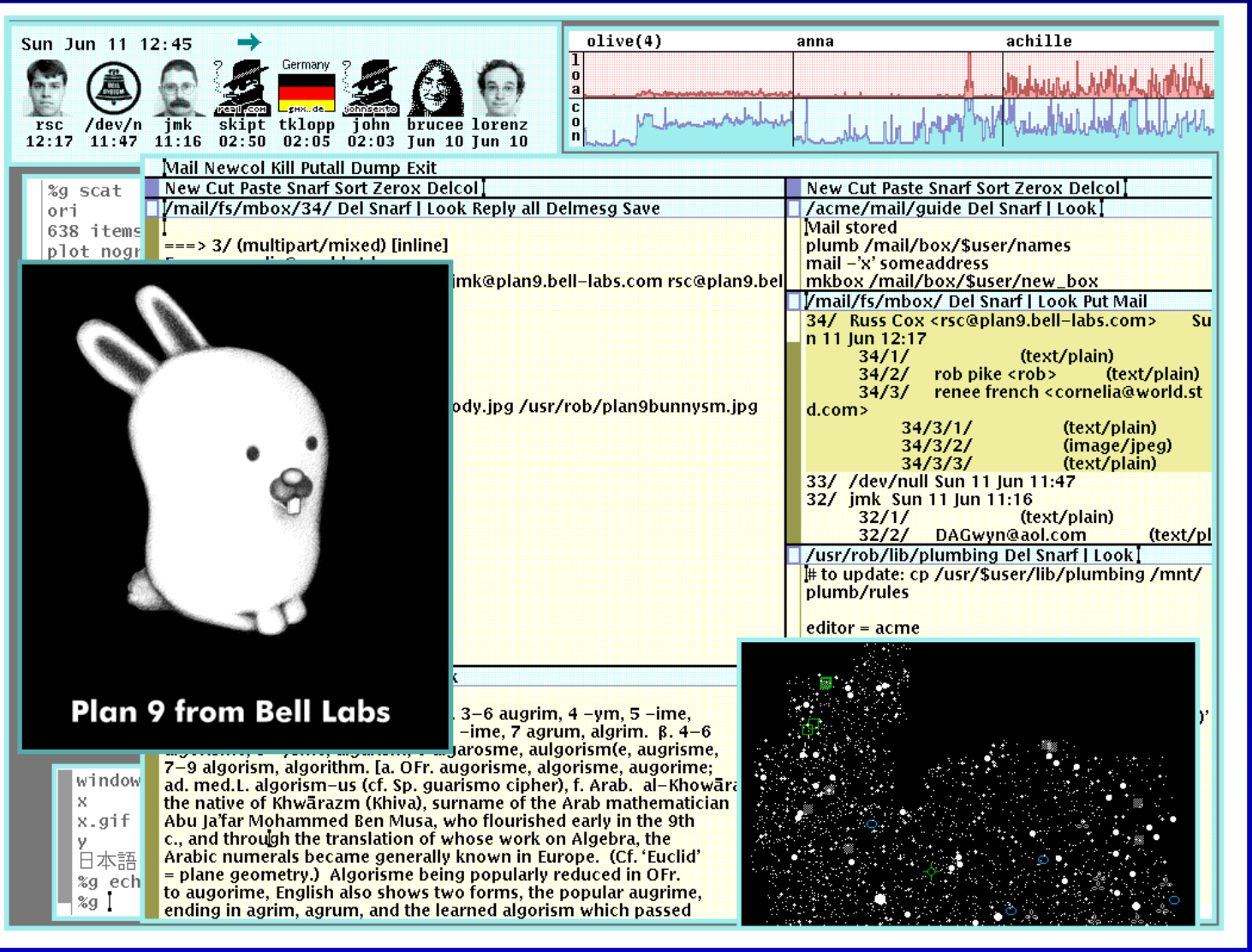
Características:

- **Plan 9 es portable en varias plataformas y puede usar microprocesadores intel en la ventana de una PC terminal y comunicarse con un servidor SPARC con cpu´s MIPS.**
- **Para procesos paralelos, se creó un lenguaje propio para realizar programación paralela que se llama ALEF.**

• Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Pantalla de Plan 9:



- Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Plan 9

Pruebas de comparación:

Prueba	Plan 9	IRIX
Context Switch	39 µs	150 µs
System Call	6 µs	36 µs
Light Fork	1300 µs	2200 µs
Pipe Latency	110 µs	200 µs
Pipe Bandwidth	11,678 Kb/s	14,545 Kb/s

- Plan 9 from Bell Labs http://www.cs.bell-labs.com/sys/doc/9.html - Mayo 2003

Introducción a los Sistemas Distribuidos