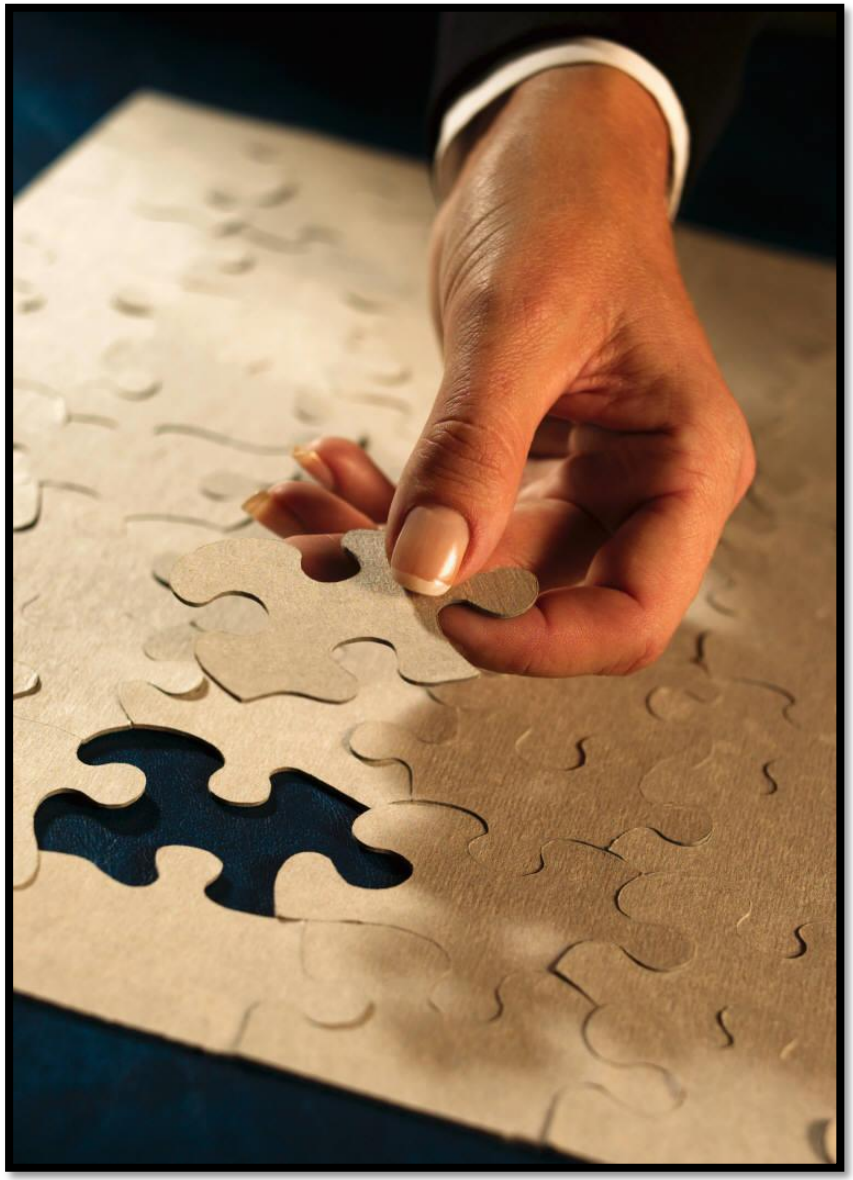
INTRODUCCIÓN A LOS SISTEMAS DISTRIBUIDOS

Logro del curso

- **Al finalizar el curso, el estudiante diseña e implementa de forma innovadora un sistema distribuido bajo los principios de Arquitectura Orientada a Servicios para satisfacer las necesidades y oportunidades de un proyecto.**



Agenda

- Silabo - Evaluación
- Asistencia
- Foros
- Blogs
- Grupos
- Trabajos
- Sistemas Distribuidos
- Retos
- Transparencias
- Tipos de SD
- Ventajas

SILABO

Sistema de Evaluación

$$15\%\,(PC1) + 20\%\,(TP1) + 5\%\,(PA1) + 25\%\,(TF1) + 5\%\,(TA1) + 5\%\,(TA2) + 5\%\,(TA3) + 20\%\,(EB1)$$

PC1 = 10 puntos de teoría + 10 puntos de práctica

TP1 (Trabajo Parcial) = 20 puntos según criterios publicados

TA1, TA2, TA3 *(Tareas Académicas) = Actividades, Foros, Tareas, Blogs, Aula Virtual*

PA1 *(Participación) = Pensamiento Innovador,* ***Rúbrica WASC***

TF1 (Trabajo Final) = 20 puntos según **Rúbrica ABET**

Se evalúa en la sustentación final

EB1*: Evaluación Final = 20 puntos todo teoría*

Grupos

Indicaciones:

- Formados en la primera sesión
- Máximo 5 integrantes (1 facilitador)
- Nombre corto como identificación
- Todos con cuenta de correo GMAIL

Grupos:

1. .Netos ()
2. Mega Red ()
3. Destructores ()
4. Zancudos ()
5. Naval Force ()

Grupo "DSDichados"

1. Héctor Saira (hectorsaira@gmail.com)
2. José Díaz (jamdiazdiaz@ gmail.com)
3. Lennon Shimokawa (lshimokawa@ gmail.com)
4. Carlos Flores Orihuela (flores.orihuela1@ gmail.com)
5. Juan Pérez (jperez@ gmail.com *Scrum Master

Trabajo Ciclo 2022-1-A

Indicaciones:

- Tema **"Solución digital innovadora que de soporte a los negocios de Restaurantes" presencial, delivery, menús, nuevos modelos de negocio, etc. aquí algunos ejemplos:** https://www.diegocoquillat.com/5-ejemplos-de-modelos-de-negocio-disruptivos-en-restaurantes/
- Puede desarrollarse con cualquier lenguaje de programación
- Se debe implementar todo el Front de la aplicación
- La Arquitectura debe contener funcionalidades con al menos 2 servicios REST o 2 Microservicios + 1 nodo de mensajería
- Se Aplicará 2 rúbricas para su evaluación WASC y ABET, cada una con una nota sobre 20.

DESARROLLO PARA SISTEMAS DISTRIBUIDOS

CERN: https://youtu.be/BH43j4seRwo

CERN

- La Web comenzó en el grupo de "adquisición y control de datos" en la Organización Europea para la Investigación Nuclear (CERN), en Ginebra, Suiza. Comenzó con un programador de computadoras que tenía una idea inteligente para un nuevo proyecto de software.
- En diciembre de 1990, para facilitar el intercambio de conocimientos, Tim Berners-Lee comenzó un proyecto de software sin fines de lucro que llamó "WorldWideWeb".
- Después de trabajar diligentemente en su proyecto durante aproximadamente un año, Berners-Lee lo había inventado e implementado.

Entonces que necesidades hace que aparezcan los sistemas distribuidos?

- Mayor velocidad de procesamiento - Hardware
- Mayor capacidad de almacenamiento de datos
- Compartir información

- Entonces que se distribuye?

- Los dispositivos
- Los servidores
- Lógica de procesos
- Datos
- El control (S.O.)
- Las aplicaciones

OTRA NECESIDAD?

Economía Social de Mercado → Economía Mundial de Mercado

La economía global https://youtu.be/tKZ8EsCeS_g

Digitalización retos: https://youtu.be/mWn-dT5hfUI

Comparación económica entre Alemania y Francia
https://youtu.be/n4XBa4EWdZw

Digitalización en Alemania
https://youtu.be/QYGIgortyn4

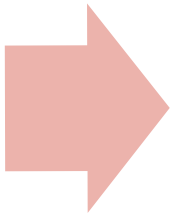
Industria 4.0 https://youtu.be/yxitQo7rn-I

https://www.bbvaopenmind.com/articulos/la-nueva-economia-y-politica-de-la-globalizacion/

Definición: Sistema Distribuido

Colección de computadoras independientes que dan al usuario la impresión de ser un único sistema coherente.

- ***Tanenbaum***



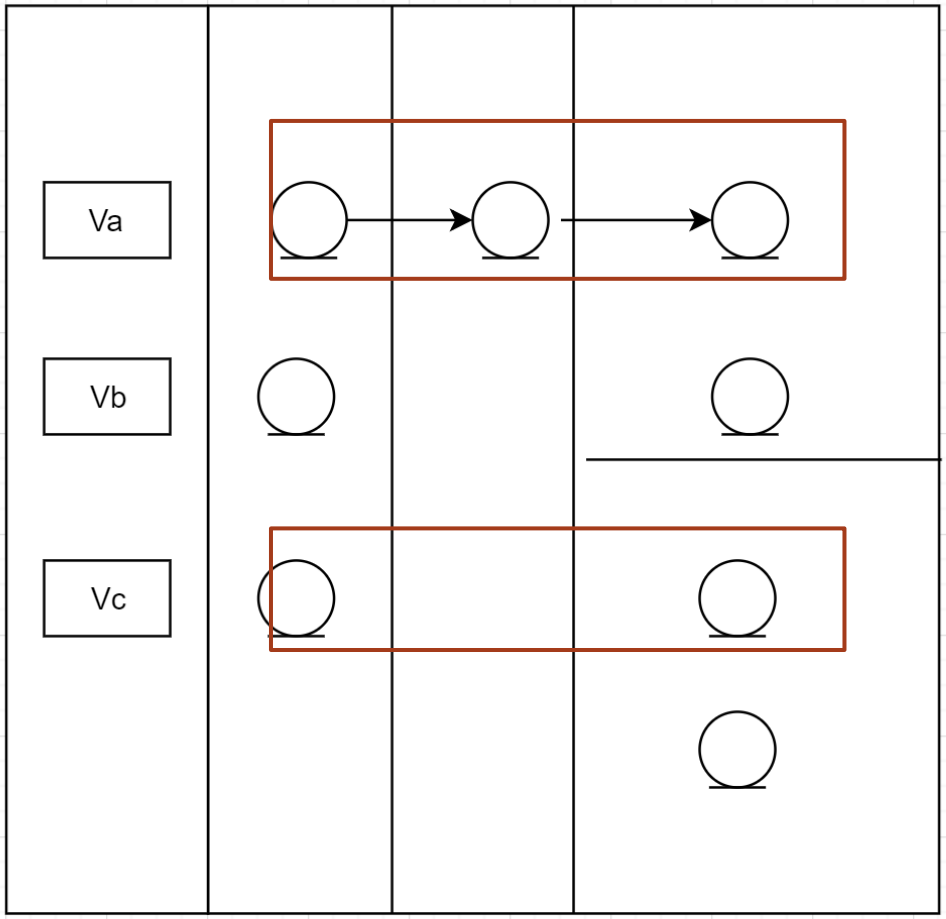
Un sistema distribuido es aquel en el que los componentes localizados en computadores, conectados en red, comunican y coordinan sus acciones únicamente mediante el paso de mensajes

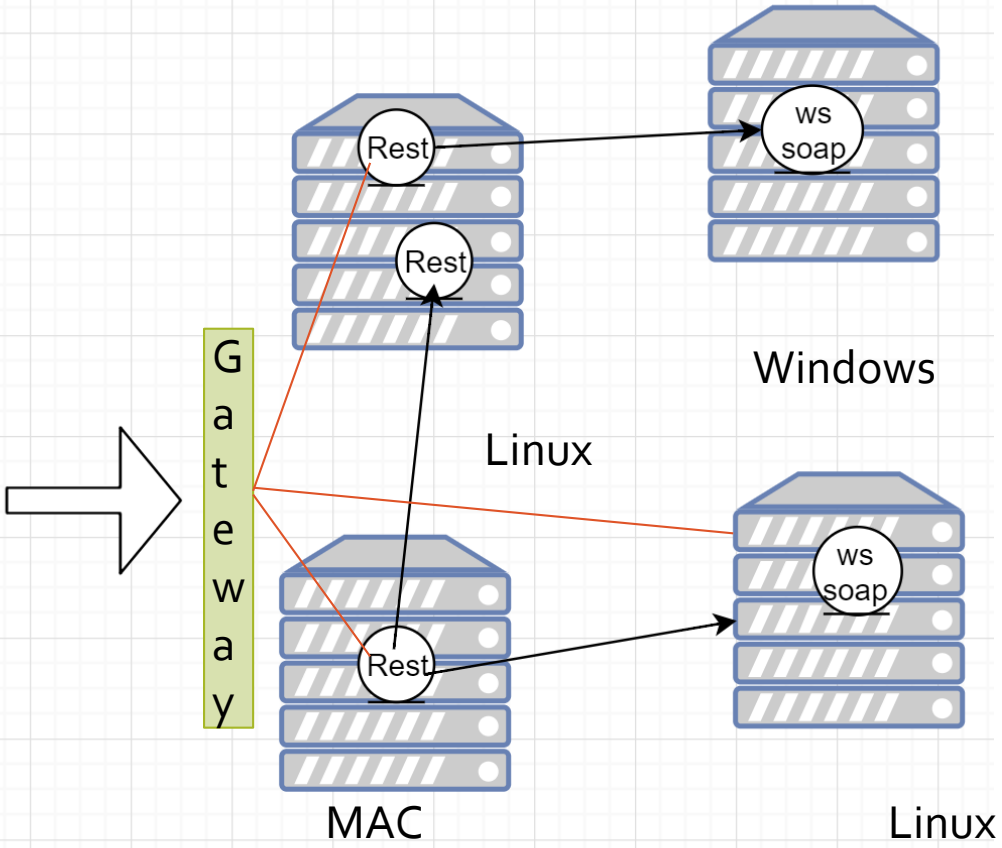
- ***Coulouris***

Ejemplos

Amazon
ViaBCP

..... un enfoque que acepte la heterogeneidad y conduzca a la descentralización





DOMINIO MONOLITICA N-LAYER

DOMAIN-BASED N-TIER

SERVICE ORIENTED FISICO

Reutilizable

Disponibilidad

Libertad de Crecimiento

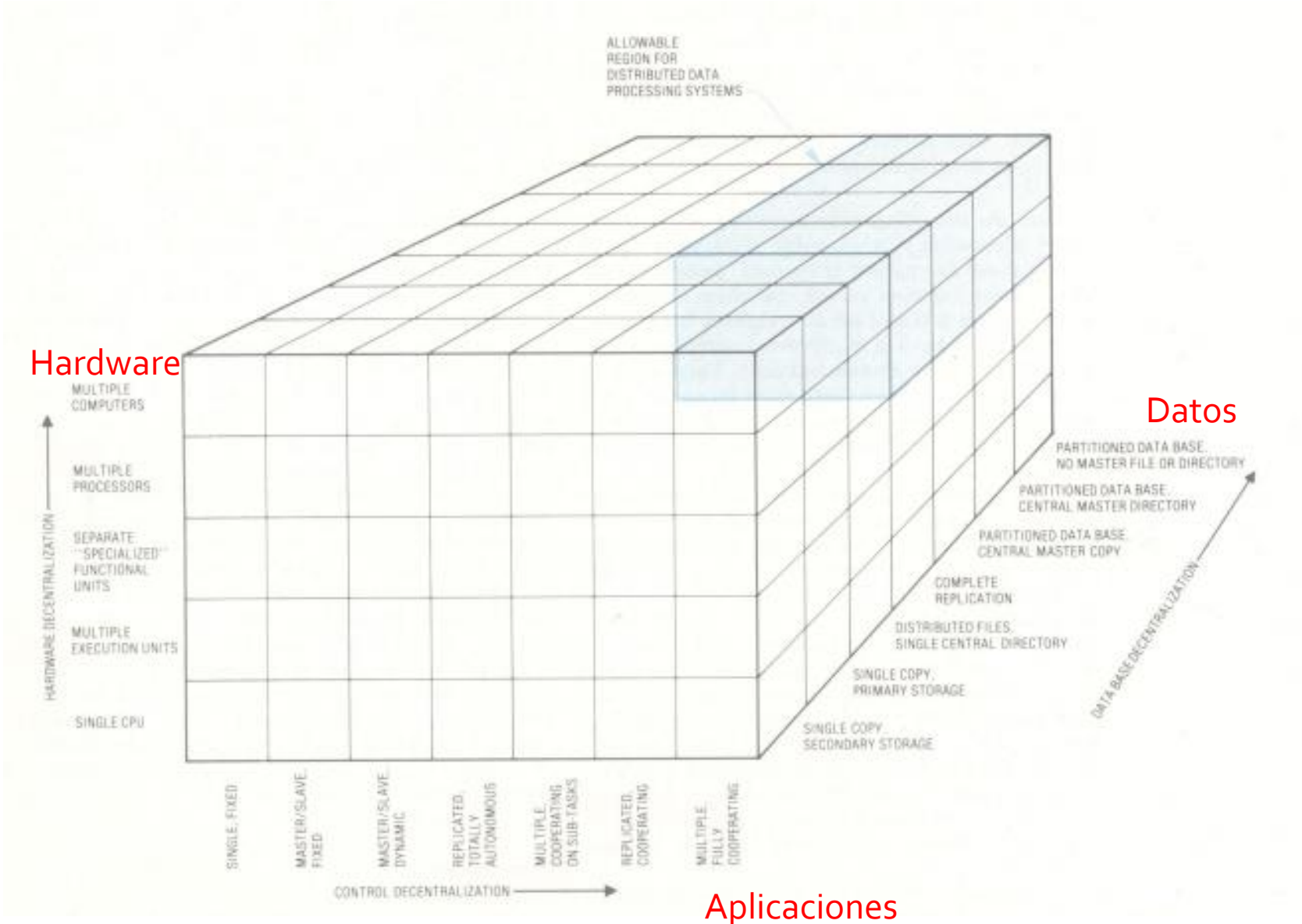
> Latencia

Independencia, plataforma y tecnología

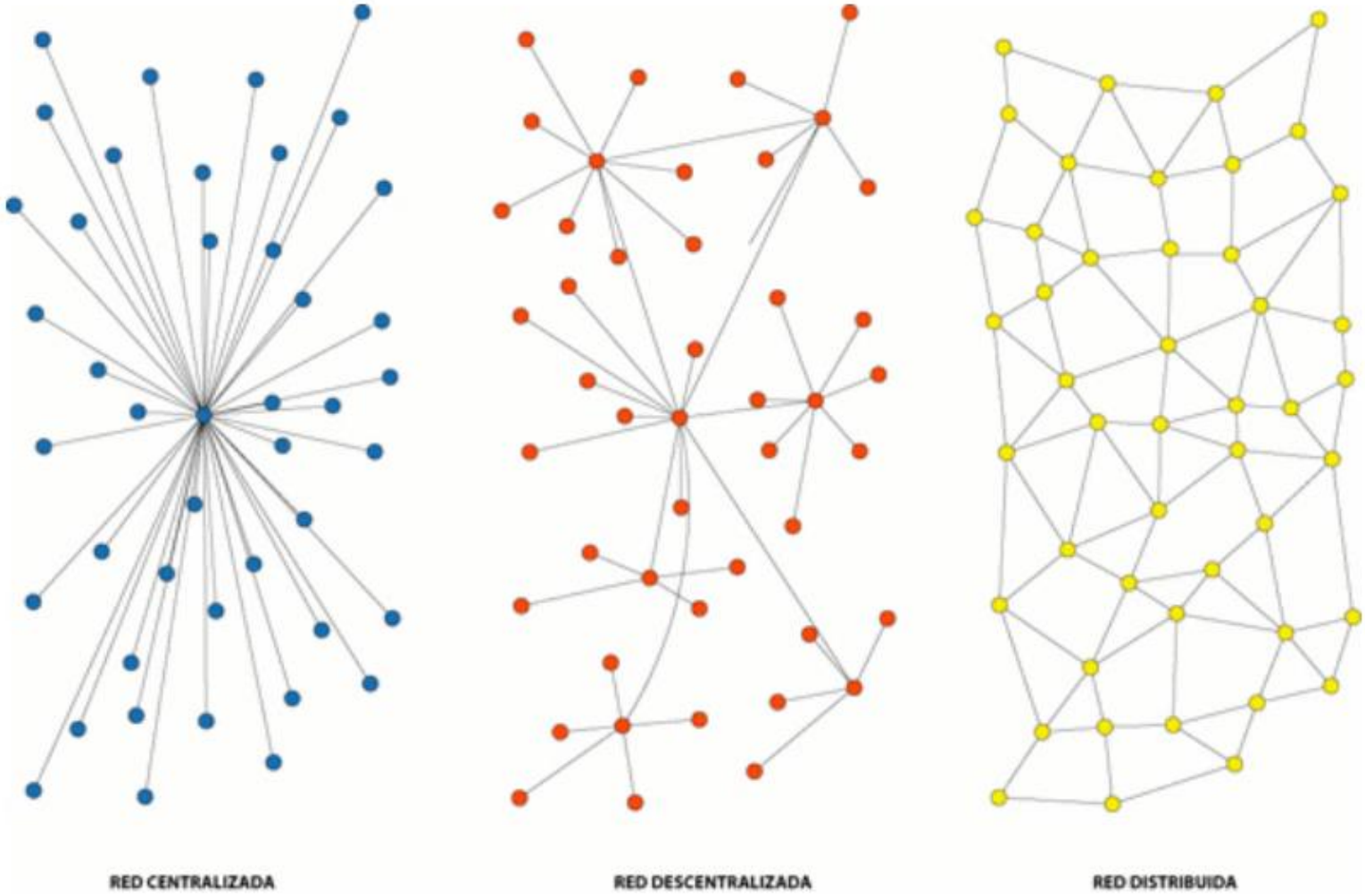
> ROI, < Costo Tx

Modelo de Phillip Enslow - Georgia Institute of Technology

Un sistema se considera distribuido si **las tres categorías (hardware distribuido, control distribuido (sw) y datos distribuidos): descentralizadas**.



“Bajo toda arquitectura de información se esconde una estructura de poder”



"La configuración básica de la red era simple, evitar cualquier nodo central y construir una red distribuida en distintos nodos interconectados“. Paul Barand.

3 comportamientos – colaboración – humanos organizan – muchos centros – clusters- tendencias Independencia de la voluntad- gran capacidad de organización – no ciencias sociales.

Motivaciones, para utilizar sistemas distribuidos

Compartición de recursos: extender las funcionalidades con la posibilidad de acceder a recursos remotos (en otros nodos).

Mejor rendimiento: combinar o paralelizar tareas para sumar capacidades y reducir "cuellos de botella".

Menor costo: tener más nodos con capacidades y tecnologías modernas resulta menos costoso que implantar grandes computadoras con hardware y soporte usualmente más especializado.

Mayor flexibilidad: reemplazar o evolucionar por partes un sistema.

Mayor disponibilidad: tener más tiempo el sistema atendiendo a los usuarios a pesar de fallas o actividades "internas".

TIPOS DE SISTEMAS DISTRIBUIDOS

- **Sistemas distribuidos computacionales:** distribuyen **recursos computacionales**. Algunas soluciones:
  - **Cluster computing** (nodos homogéneos discretos y cercanos con alto grado de control)
  - **Grid computing** (nodos heterogéneos, con poco grado de control e incluso distantes que aportan capacidades relativas para problemas complejos
  - **Cloud computing** (nodos usualmente virtualizados con control elástico para ser utilizados a demanda para múltiples capacidades).
- **Sistemas distribuidos informacionales:** distribuyen **lógica e información** que son accedidos por nodos consumidores y proveedores. Existen varios modelos y **tecnologías de integración** para las invocaciones, como: RPC, RMI, DCOM, WS-SOAP, HTTP REST, mensajería (JMS, MSMQ, AMQP).
- **Sistemas distribuidos ubicuos:** distribuyen potencialmente casi cualquier capacidad de cualquier "cosa" a la que se le **embebed** unidades computacionales y de conectividad para aprovechar **sensores** y actuadores (**smart-things**) de foma contextual y personalizada. Algunas **tecnologías** que dan vida a estos sistemas: RFID, NFC, MQTT, XMPP, Physical Web.

CARACTERÍSTICAS DE LOS SISTEMAS DISTRIBUIDOS

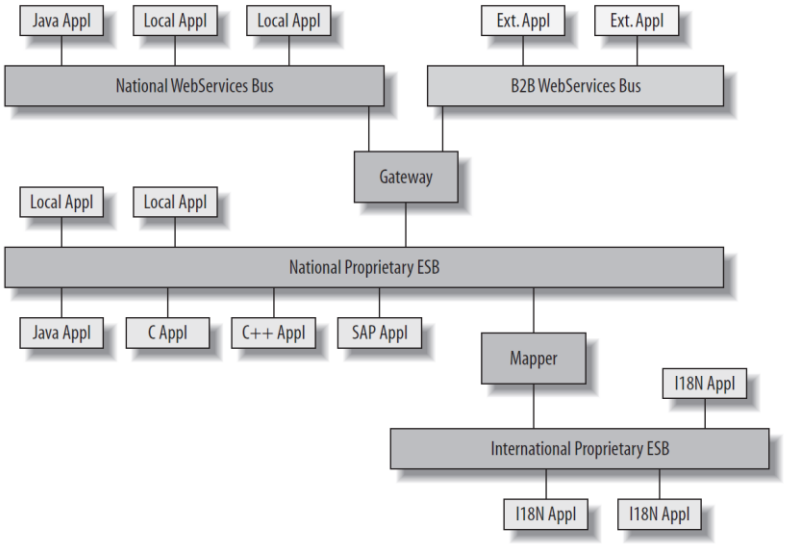
RETOS DE LOS SISTEMAS DISTRIBUIDOS

1) Heterogeneidad

- En un-SD pueden convivir muchas plataformas.
- Un SD debe poseer la capacidad de poder añadir o reemplazar los nodos de la red que pueden ser de diferente tipo heterogéneo.

- Heterogéneo:
  - Plataformas
  - Sistemas Operativos
  - Redes
  - Representación de datos
  - Protocolos
  - Lenguajes de programación

Y como puede resolverse?

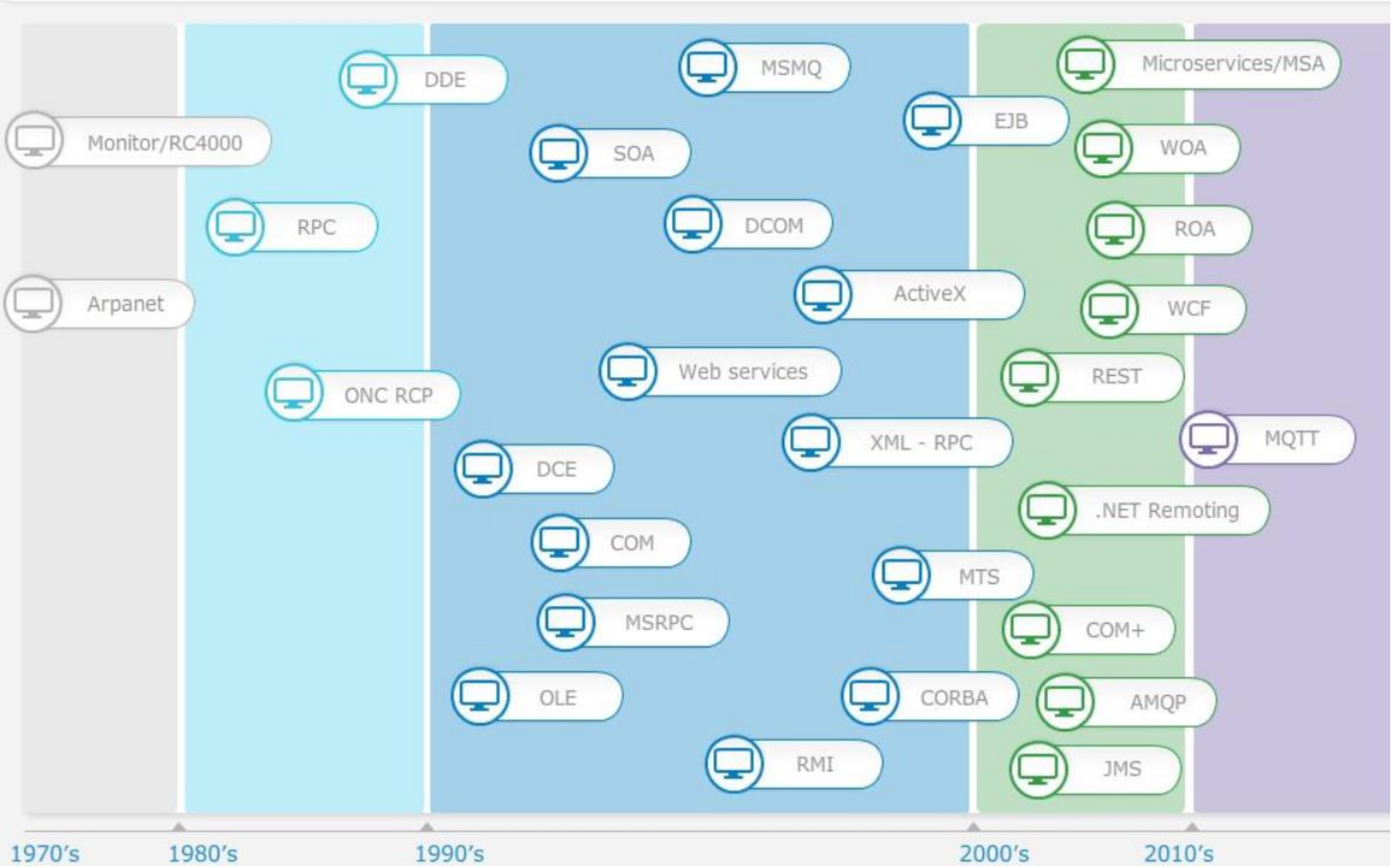


- La heterogeneidad trata de resolverse mediante un **middleware**

Características MIDDLEWARE:

- ✓ Independiza el servicio de su implantación, del sistema operativo y de los protocolos de comunicaciones.
- ✓ Permite la convivencia de distintos servicios en un mismo sistema.
- ✓ Permite la transparencia en el sistema.
- ✓ Modelo tradicional: Monitor de teleproceso o CICS, Tuxedo, Encina.
- ✓ Modelo OO: CORBA.
- ✓ BUS: SOA : WS Rest, WS SOA

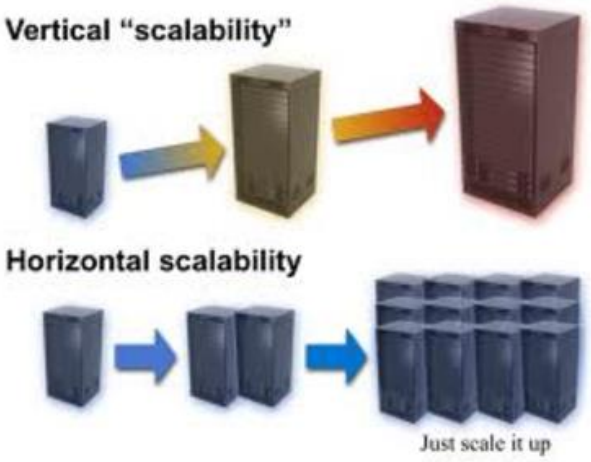
PRINCIPALES TECNOLOGÍAS Y ESTILOS ARQUITECTURALES PARA DISEÑO Y DESARROLLO DE SISTEMAS DISTRIBUIDOS A TRAVÉS DEL TIEMPO

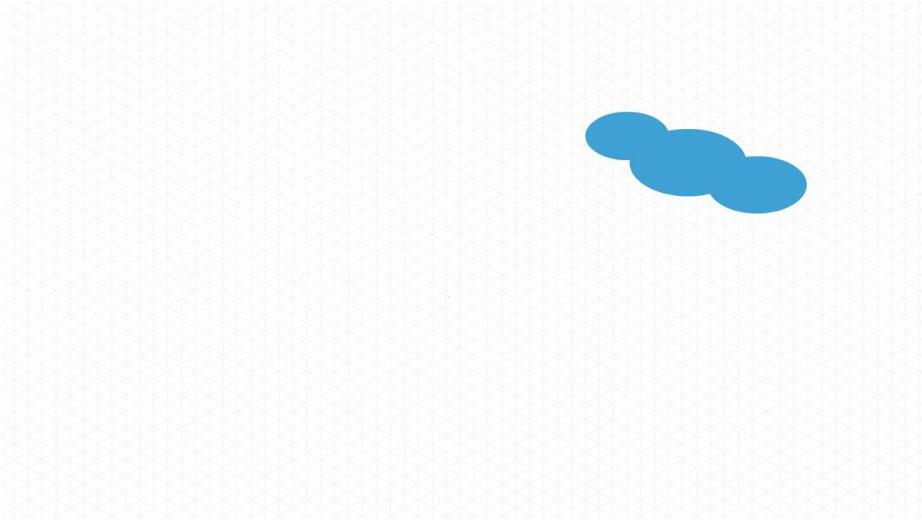


2) Escalabilidad

- El Sistema conserva su calidad/efectividad de servicio al incrementar # de recursos y el # de usuarios significativamente.

- Clusters, load balancing, cloud computing, “replicación”.





Horizontal / Load Balancing

Escalabilidad Vertical HDW

Ventajas:

- No implica un gran problema para las aplicaciones, pues todo el cambio es sobre el hardware
- Es mucho más fácil de implementar que el escalamiento horizontal.
- Puede ser una solución rápida y económica (compara con modificar el software)

Desventajas:

- El crecimiento está limitado por el hardware.
- Una falla en el servidor implica que la aplicación se detenga.
- No soporta la Alta disponibilidad.

Escalabilidad Horizontal hdw

Ventajas:

- El crecimiento es prácticamente infinito, podríamos agregar cuantos servidores sean necesarios
- Es posible combinarse con el escalamiento vertical.
- Soporta la alta disponibilidad
- Si un nodo falla, los demás sigue trabajando.
- Soporta el balanceo de cargas.

Desventajas:

- Requiere de mucho mantenimiento
- Es difícil de configurar
- Requiere características especiales las aplicaciones (si no fueron diseñadas para trabajar en cluster)
- Requiere de una infraestructura más grande.

3) Extensibilidad (openness)

- Grado en la cuál se pueden adicionar nuevos servicios de compartición de recursos con cambios mínimos en lo que existe.
- Requiere interfaces bien definidas, documentadas y publicadas, disponibles para desarrolladores
- Un factor crítico es la independencia de fabricantes concretos, debe basarse en sistemas abiertos.
- Como lograr un Sistema distribuido abierto?
  - Comunicación uniforme y publica entre procesos con interfaces públicas estándares.

- Ejm: *Plugins, APIs, Nuevos módulos*

4) Seguridad

La seguridad tiene tres frentes que defender:

Confidencialidad
Protección contra individuos no autorizados, autenticación

Integridad
Protección contra la alteración o corrupción, encriptación

Disponibilidad
Protección contra la interferencia que impide el acceso a los recursos, tokens, sessions

5) Tratamiento de fallos (Resiliencia)

Un Sistema Distribuido debe poder **detectar, ocultar y recuperarse** de fallos que puedan ocurrir en la red.

- Control de propagación de errores
- Reintentos, reenvío de mensajes, redundantes
- Recuperación frente a errores, rollback

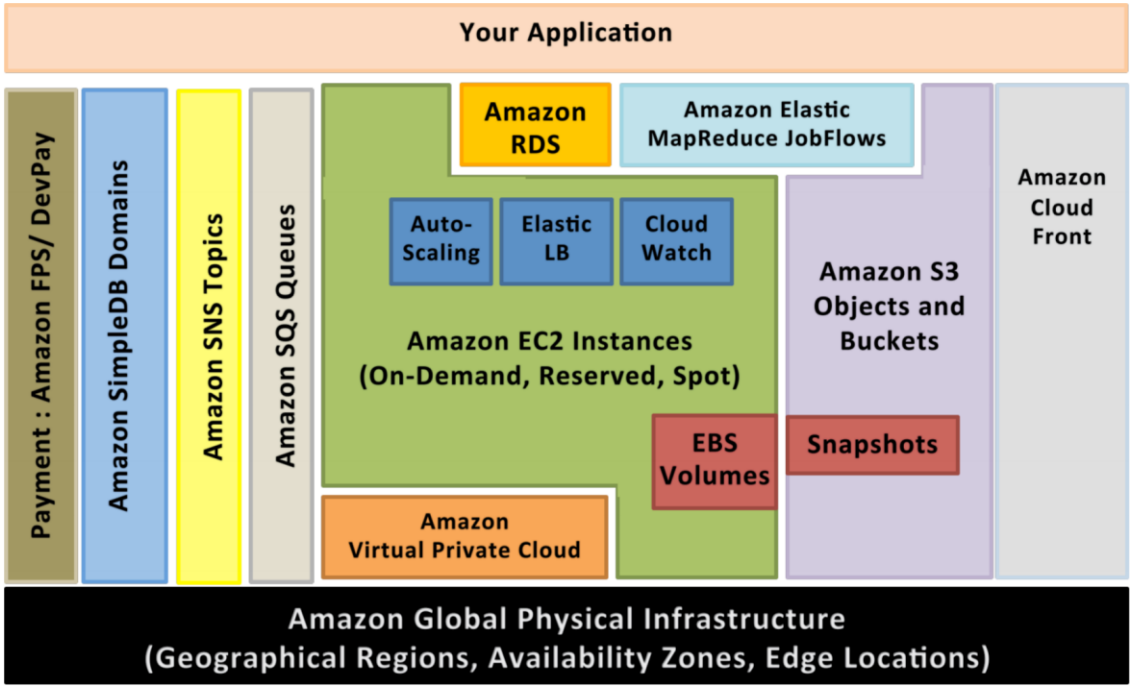
Tolerancia de fallos:
Los programas clientes de los servicios pueden diseñarse para tolerar ciertos fallos y NO dejar de dar el servicio o perder la transacción o servicio. Alta disponibilidad.

6) Concurrencia

- El sistema debe permitir múltiples usuarios utilizando simultáneamente los recursos de la red.
- Cada objeto que represente un recurso compartido en un sistema distribuido debe responsabilizarse de garantizar que opera correctamente en un entorno concurrente.
  - Recursos compartidos
  - Thread safe
  - Sincronía de operaciones

7) Elasticidad

- Funciona en ambos sentidos: ampliación o reducción de los recursos de la plataforma.
- Es instantánea y automática: No requiere nuestra intervención.
- La escalabilidad es intrínseca a la elasticidad.



8) Transparencia

- Un objetivo importante de un sistema distribuido es ocultar el hecho de que sus procesos y recursos están físicamente distribuidos a través de múltiples computadoras.
- Decimos que un sistema distribuido es **transparente** si es capaz de presentarse ante los usuarios y las aplicaciones como si se tratara de una sola computadora.

Tipos de Transparencia

- **Ubicación:** Permite acceder a los recursos sin conocer su localización. Oculta la localización física del recurso, url, path
- **Acceso:** Se accede mediante igual mecanismo a recursos locales y remotos, se ocultan detalles de formato de datos o de conectividad. Templates.
- **Migración:** Reconfiguración sin cambio en clientes, frameworks
- **Reubicación:** Oculta el que un recurso pudiera moverse a otra ubicación mientras se accede a ellos.
- **Concurrencia:** Oculta que un recurso puede ser compartido por varios usuarios, misma tabla, mismo archivo…
- **Falla:** Oculta fallos dejando que el usuario o programa de aplicación complete sus tareas a pesar de fallos HW o SW
- **Replicación**: Uso de múltiples elementos de cada recurso para aumentar fiabilidad y prestaciones sin que los usuarios necesiten su conocimiento

Ventajas DE LOS SISTEMAS DISTRIBUIDOS

Economía: mejor aprovechamiento de hardware cada vez más potente a menores costos.

Escalabilidad: las capacidades de procesamiento o almacenamiento pueden ser ampliadas incrementalmente.

Tolerancia a fallos: posibilidad de tener esquemas redundantes para mantener funciones operativas.

Mantenibilidad: facilidad para actividades parciales y por partes en el sistema.

Ejercicio Grupal:

1) Ejemplos de algún caso que no puede ser escalable
2) Describir un problema de concurrencia
3) Describir un problema de seguridad en un Sistema Web
4) Describir un caso que si contempla tratamiento de fallas

OBJETIVO DE UN SISTEMA DISTRIBUIDO

El objetivo de un sistema distribuido es **integrar** los recursos y servicios conectados por una red de comunicación.

Hoy en día, los clientes, proveedores y compañías se encuentran generalmente en diferentes localidades alejados los unos de los otros. Debido a que todos estos utilizan computadoras, las redes de información que los unen y que les permiten interactuar pueden ofrecer a las empresas mayor **competitividad**.

Estilos de Integración

RPC

ROOA

SOA

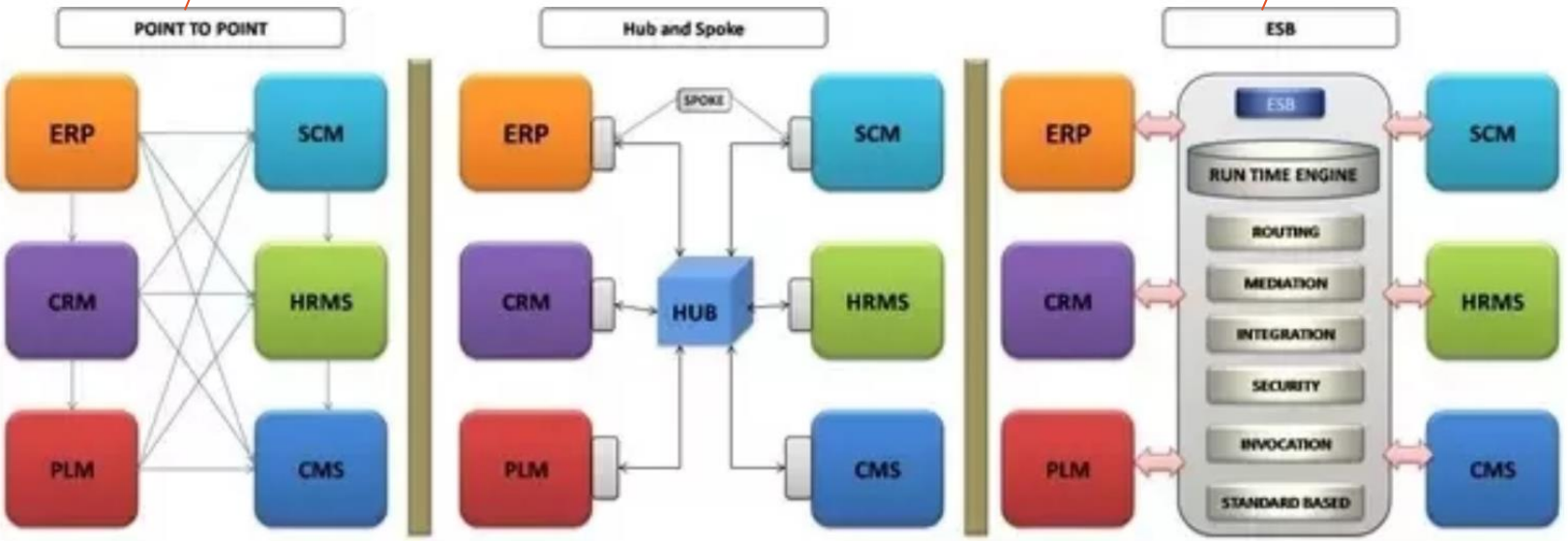
RMI, DCOM, CORBA

WS-SOAP

REST → Estilos de Invoc.

ROA → Arquitecturas

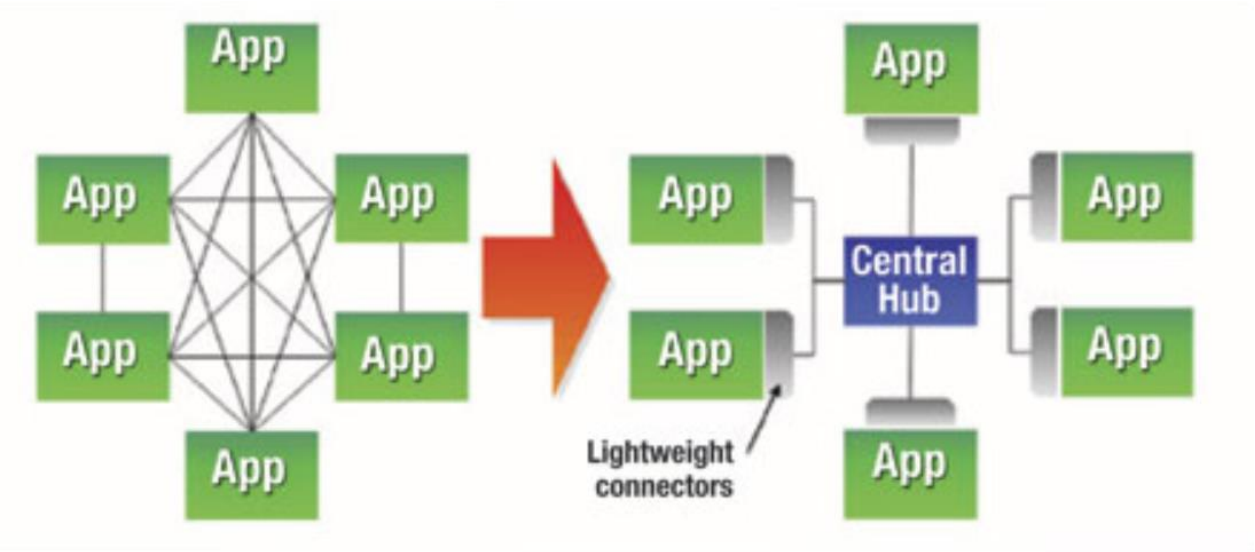
REST → Tecnologías



Point to Point

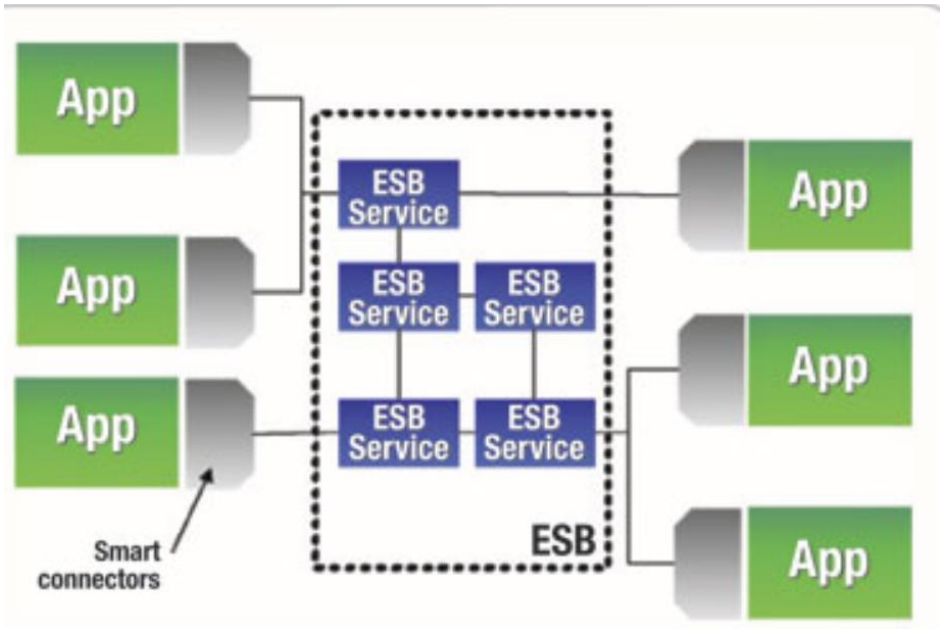
Transformar-Enrutar – (Orquestar)

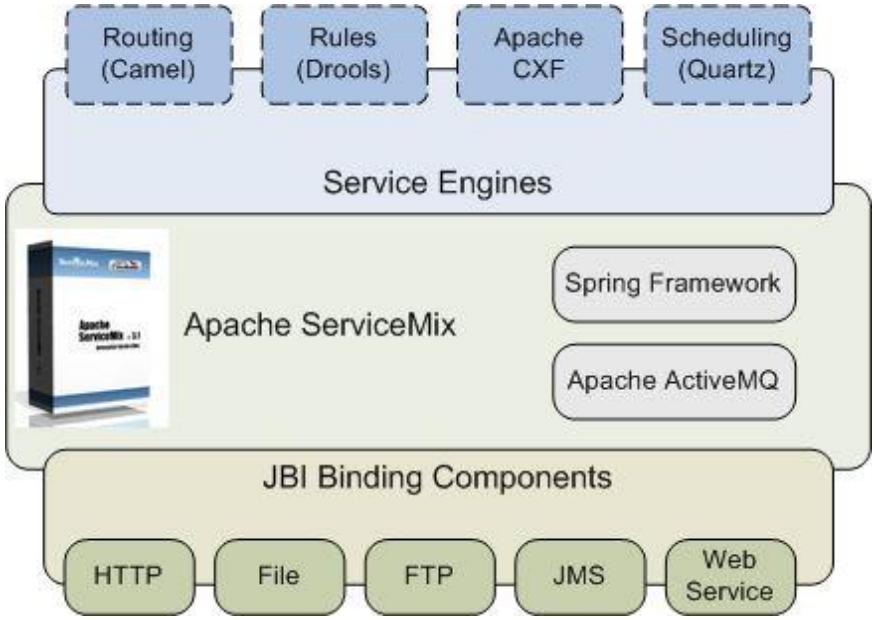
Hub and Spoke



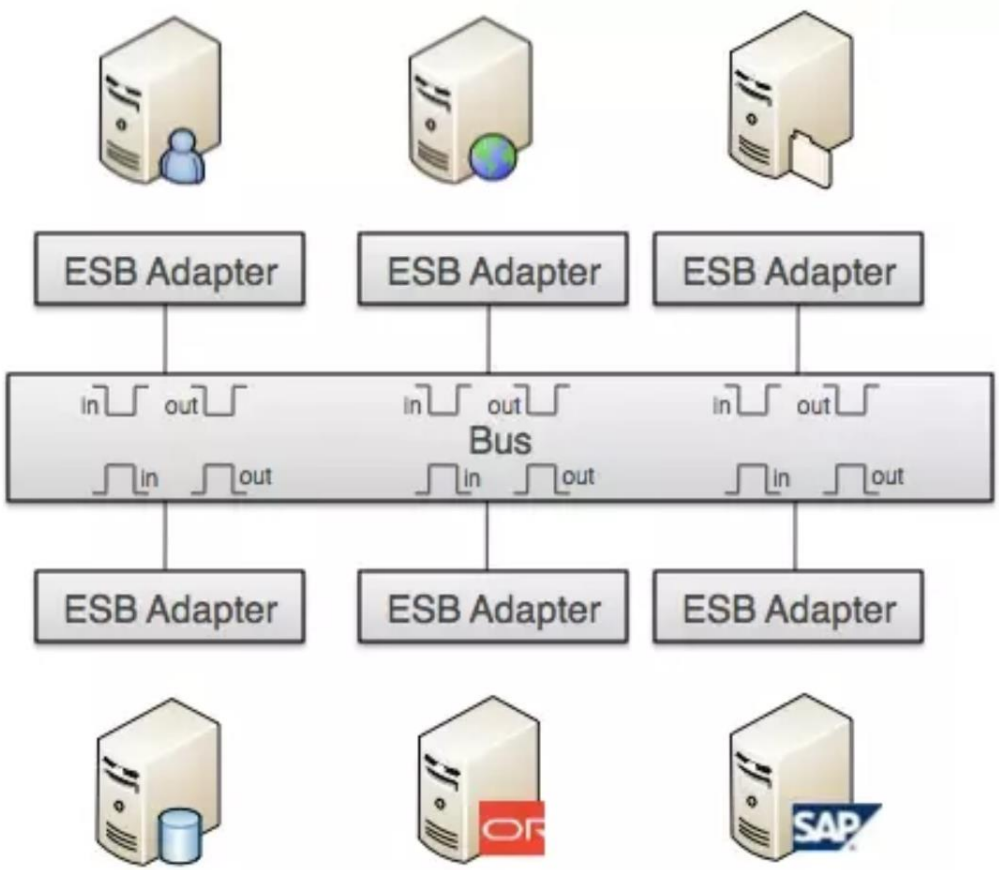
- Uno de los objetivos clave de la arquitectura de Hub and Spoke con conectores es dejar los sistemas actuales intactos y sin cambios tanto como sea posible.
- El concentrador central realiza la transformación, validación, enrutamiento y entrega de mensajes asincrónicos de mensajes.
- Hub se Cae se Cae todo
- Ej. Biztalk

ESB, Bus de Servicios





ESB

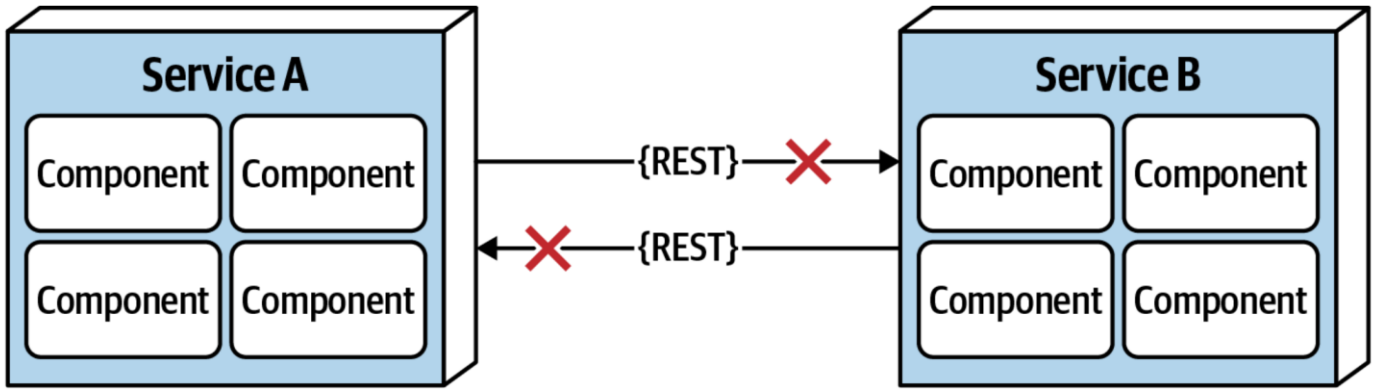


- ✓ OpenESB implementación en Java.
- ✓ Oracle ESB
- ✓ Oracle Service Bus (BEA AquaLogic Service Bus)
- ✓ Microsoft BizTalk Server
- ✓ Windows Azure Service Bus
- ✓ IBM WebSphere ESB
- ✓ IBM Integration Bus (IBM WebSphere Message Broker)
- ✓ JBoss Fuse
- ✓ Spring Integration
- ✓ Phoenix Service Bus en C#.
- ✓ Apache ServiceMix

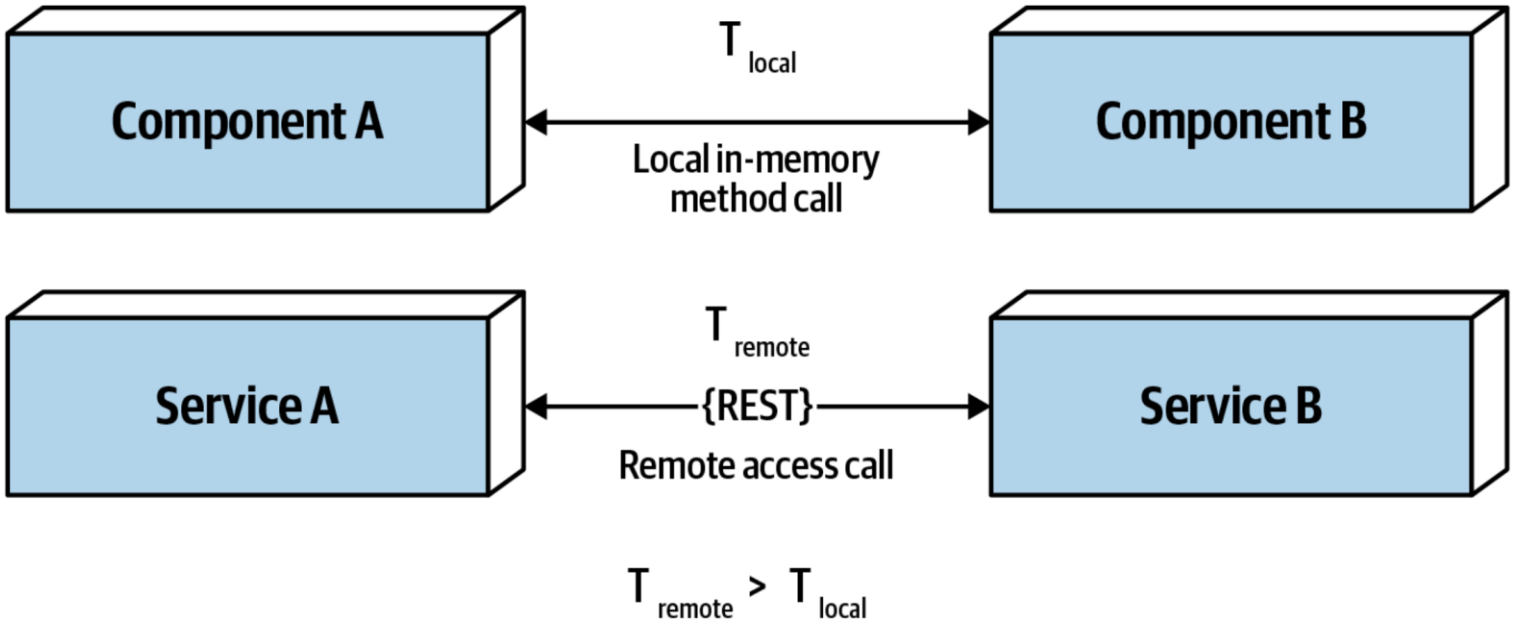
FALACIAS DE LOS SISTEMAS DISTRIBUIDOS

L. Peter Deutsch - Sun Microsystems

Falacia # 1: la red es confiable

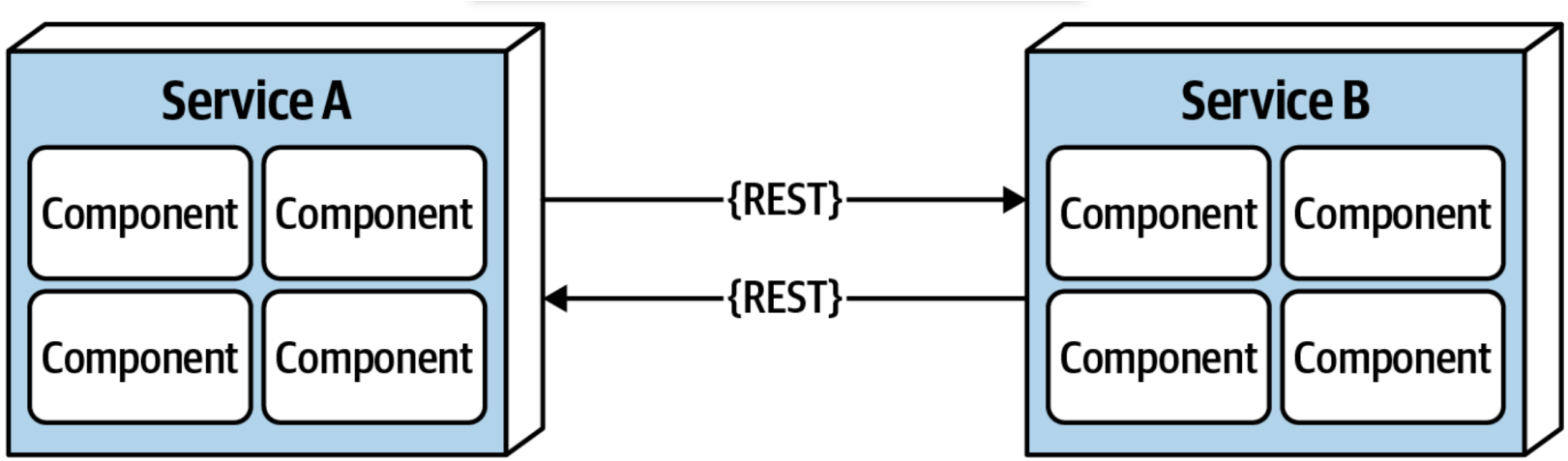


Falacia # 2: La latencia es cero

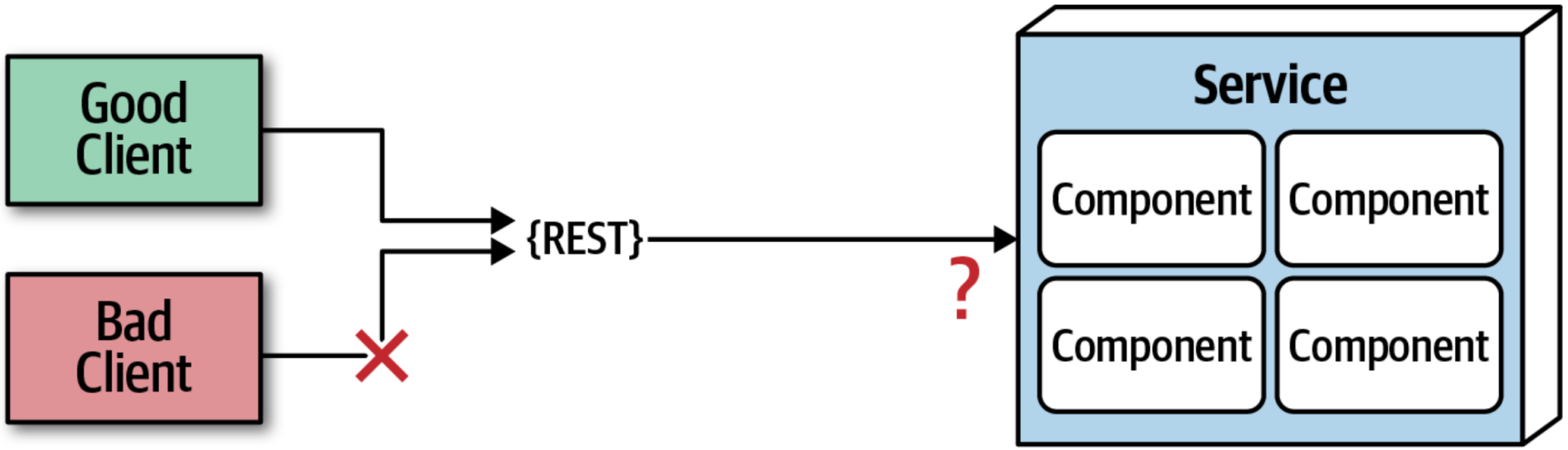


Los arquitectos deben conocer **este promedio de latencia**., es la única forma de determinar si una arquitectura distribuida es factible, particularmente cuando se consideran microservicios

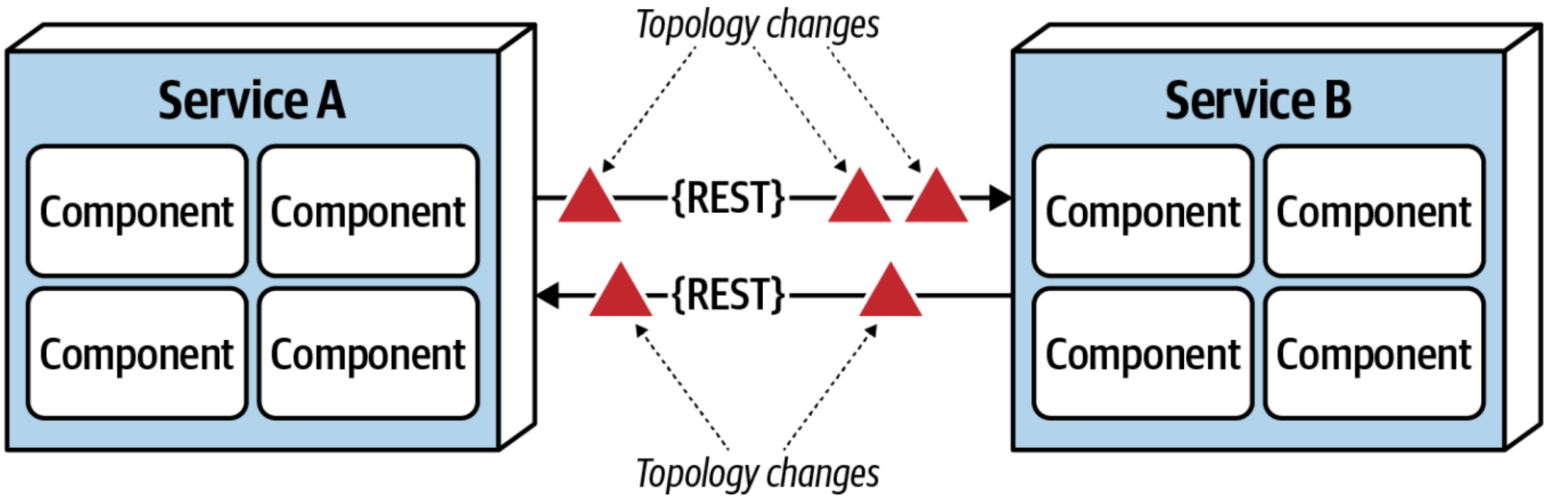
Falacia # 3: el ancho de banda es infinito



Falacia # 4: la red es segura

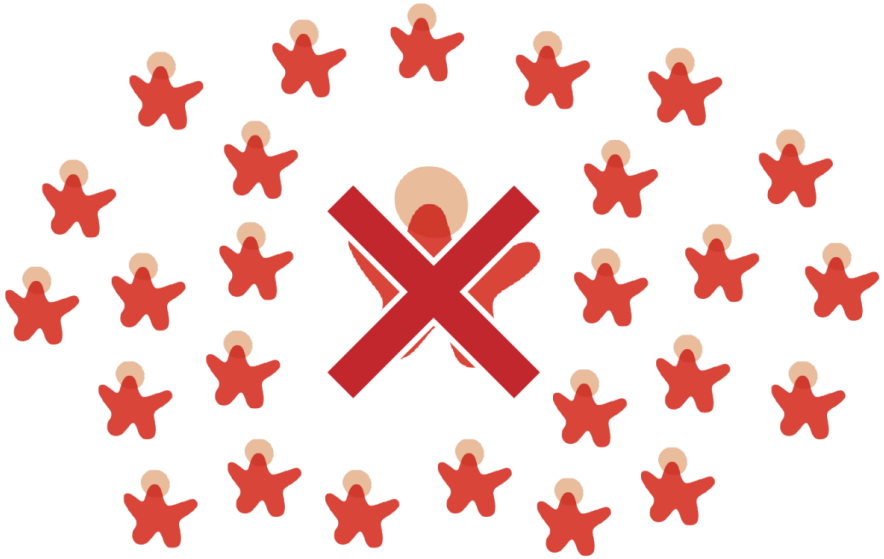


Falacia # 5: la topología nunca cambia



Falacia # 6: solo hay un administrador

- Hay muchos administradores de red, no solo uno



Falacias

8 falacias en el diseño de sistemas distribuidos

"la red es confiable"	**no!** Los mensajes sí se pierden. Existen colisiones y rechazos en la red
"la latencia es cero"	**no!** La latencia no puede ser cero (físicamente imposible). WAN degrada velocidades LAN. Sincronía aumenta los tiempos totales
"el ancho de banda es infinito"	**no!** No se puede controlar el ancho de banda de todas las redes. Mensajes pequeños tienen menos problemas
"la red es segura"	**no!** La utilización de redes aumenta la vulnerabilidad y exposición ante atacantes
"la red es homogénea"	**no!** Existen tecnologías propietarias. Interoperabilidad requiere capacidades de integración
"el costo de transporte es cero"	**no!** Existen costos monetarios y prácticos. Varias redes sacrifican rendimiento por costos
"sólo existe un administrador"	**no!** Los errores ocurren en varios dominios de responsabilidad, cada zona tiene sus prioridades y políticas
"la topología no cambia"	**no!** La evolución tecnológica es constante. A través de mecanismos de descubrimiento se acceden a recursos cambiantes

Referencias usadas y recomendadas

- Sistemas Distribuidos, Andrew Tenembaum, Segunda Edición, 2007
- Presentación de: Sorey Bibiana García Zapata
- Computación Distribuida, M. L. Liu, Ed. Pearson Adison Wesley, 2004
- Documentos producidos por UPC, Héctor Saira
- https://www.youtube.com/watch?v=Wu7h9gAdkFo
- http://www.abc.es/tecnologia/informatica-software/20141202/abci-stephen-hawking-peligros-inteligencia-201412021837.html
- https://www.youtube.com/watch?v=uY-6PcO96Bw
- https://www.youtube.com/watch?v=VTs5y1QlEtk
- https://www.ted.com/talks/jinha_lee_a_tool_that_lets_you_touch_pixels

TA1: Grupal

- IOT
- Block Change
- Microservicios
- Realidad Aumentada
- Machine Learning
- Hadoop
  - Características, que es?
  - Como Funciona?
  - Para que Sirve?
  - Que Beneficios Trae?
  - Que productos o Servicios se han desarrollado?
  - Ejemplo en Código o Video / Nacional e Internacional explicado.

Obs: Fuentes Fidedignas, libros, sites oficiales, papers
No Blogs ni Wikipedia