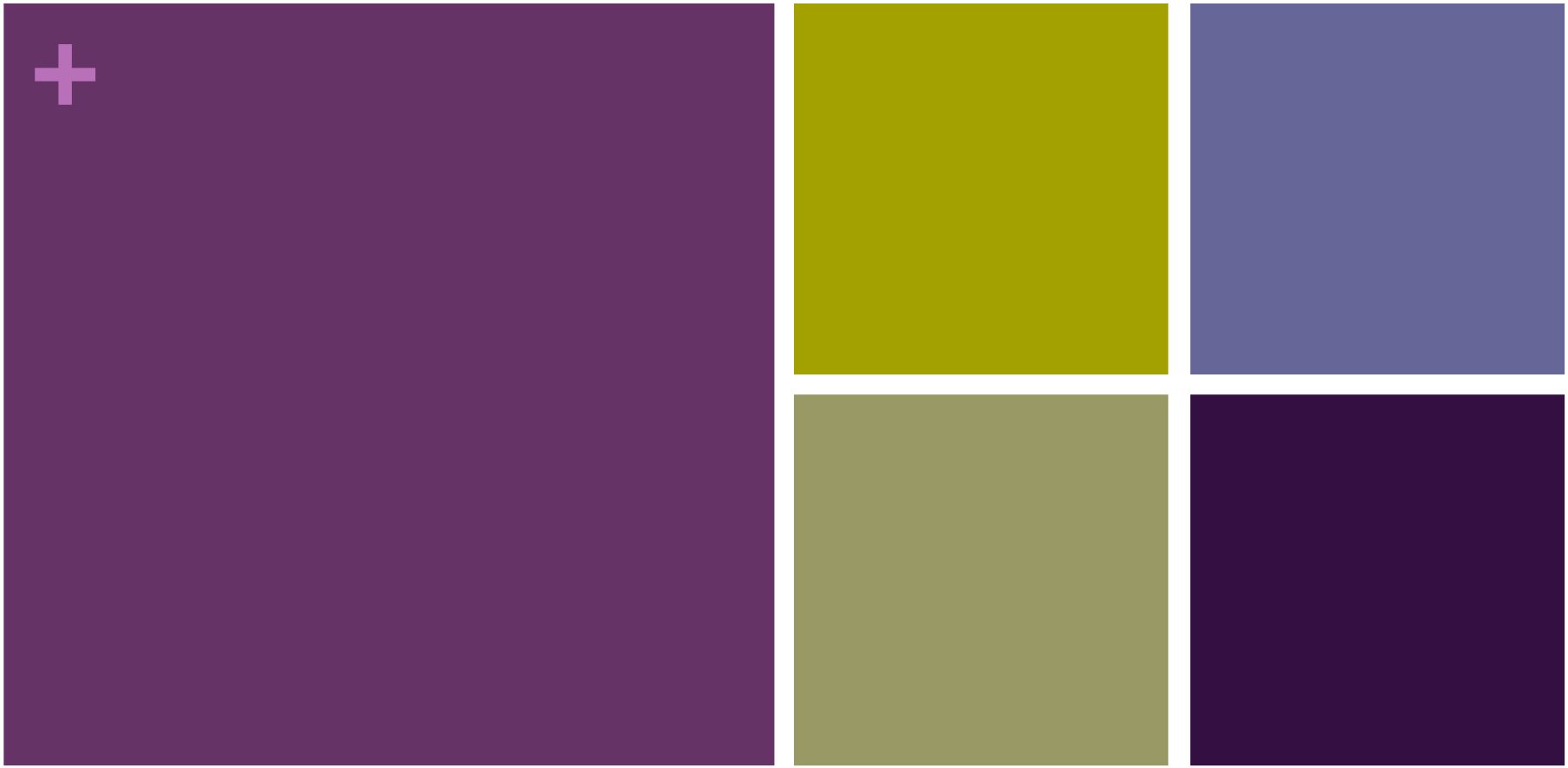
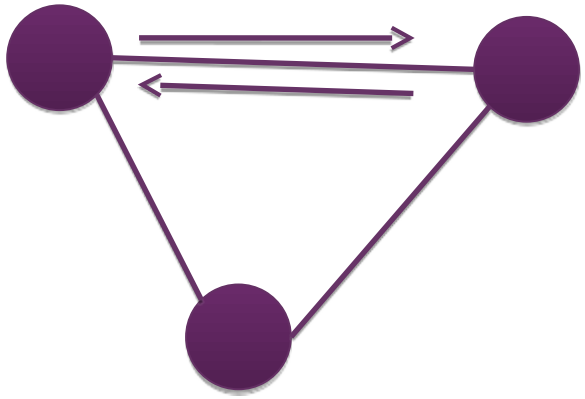


Sistemas Distribuidos
Introducción

Rodrigo Santamaría

¿Qué es un sistema distribuido?

- Es un sistema en el que los componentes hardware o software:
  - Se encuentran en computadores **unidos mediante una red**
  - Se comunican únicamente mediante **paso de mensajes**



¿Qué sistemas que conozcas crees que pueden considerarse un sistema distribuido?

Características de un SD

- Concurrencia
  - Varios componentes acceden a la vez a un recurso compartido
    - Hardware: impresoras, discos
    - Software: ficheros, bases de datos, objetos de datos
- Inexistencia de un reloj global
  - Necesidad de temporalidad para coordinación/sincronización
- Fallos independientes
  - Por aislamiento de la red (red)
  - Por parada de un computador (hardware)
  - Por terminación anormal de un programa (software)

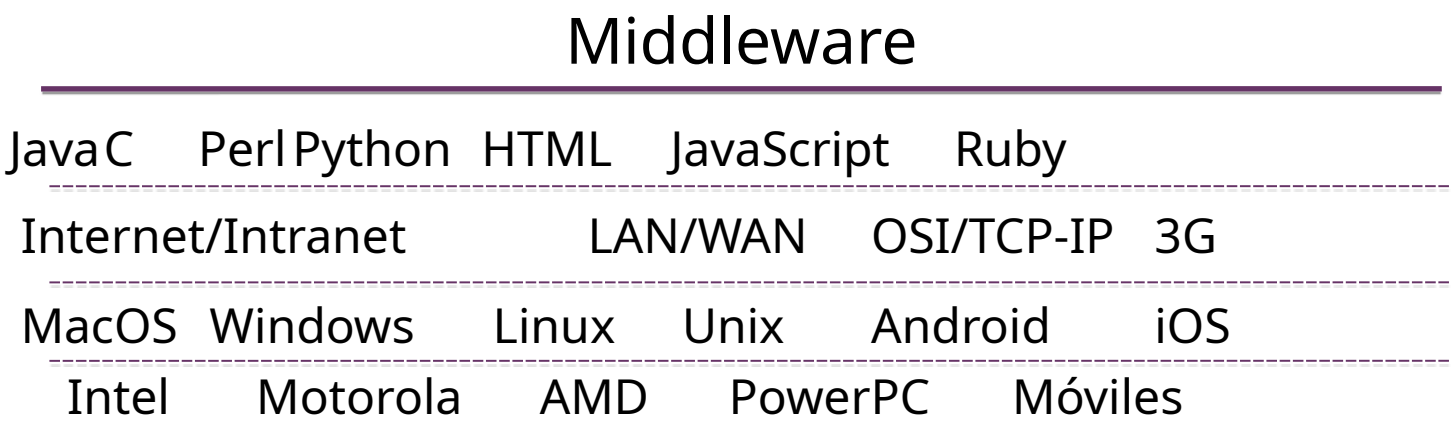
Desafíos

1. Heterogeneidad
2. Extensibilidad
3. Seguridad
4. Escalabilidad
5. Tratamiento de fallos
6. Concurrencia
7. Transparencia

1.- Heterogeneidad

- En un SD existe heterogeneidad a mucho niveles:
  - Redes → distintos protocolos de red
  - HW → distinta representación de los datos
  - SSOO → distintas llamadas al sistema
  - Lenguajes de programación → representación de estructuras de datos, acceso a métodos, etc.
  - Implementaciones → adopción de estándares

- Solución: middleware, estrato SW que enmascara la heterogeneidad subyacente



2.- Extensibilidad

- Grado en que se pueden añadir y publicar nuevos servicios para su uso por una variedad de clientes
  - Publicación de interfaces
- Por ejemplo
  - RFC (Request For Comments)
    - Propuestas de protocolos para Internet
  - WSDL (Web Service Description Language)
    - Descripción y publicación de servicios web (SOAP)

3.- Seguridad

- Comunicación por paso de mensajes
  - Estos mensajes pueden ser manipulados por terceros
- Tres desafíos principales:
  - Confidencialidad: lectura de mensajes por terceros
  - Integridad: modificación de mensajes por terceros
  - Disponibilidad: interferencia con los procedimientos de acceso a los recursos

4.- Escalabilidad

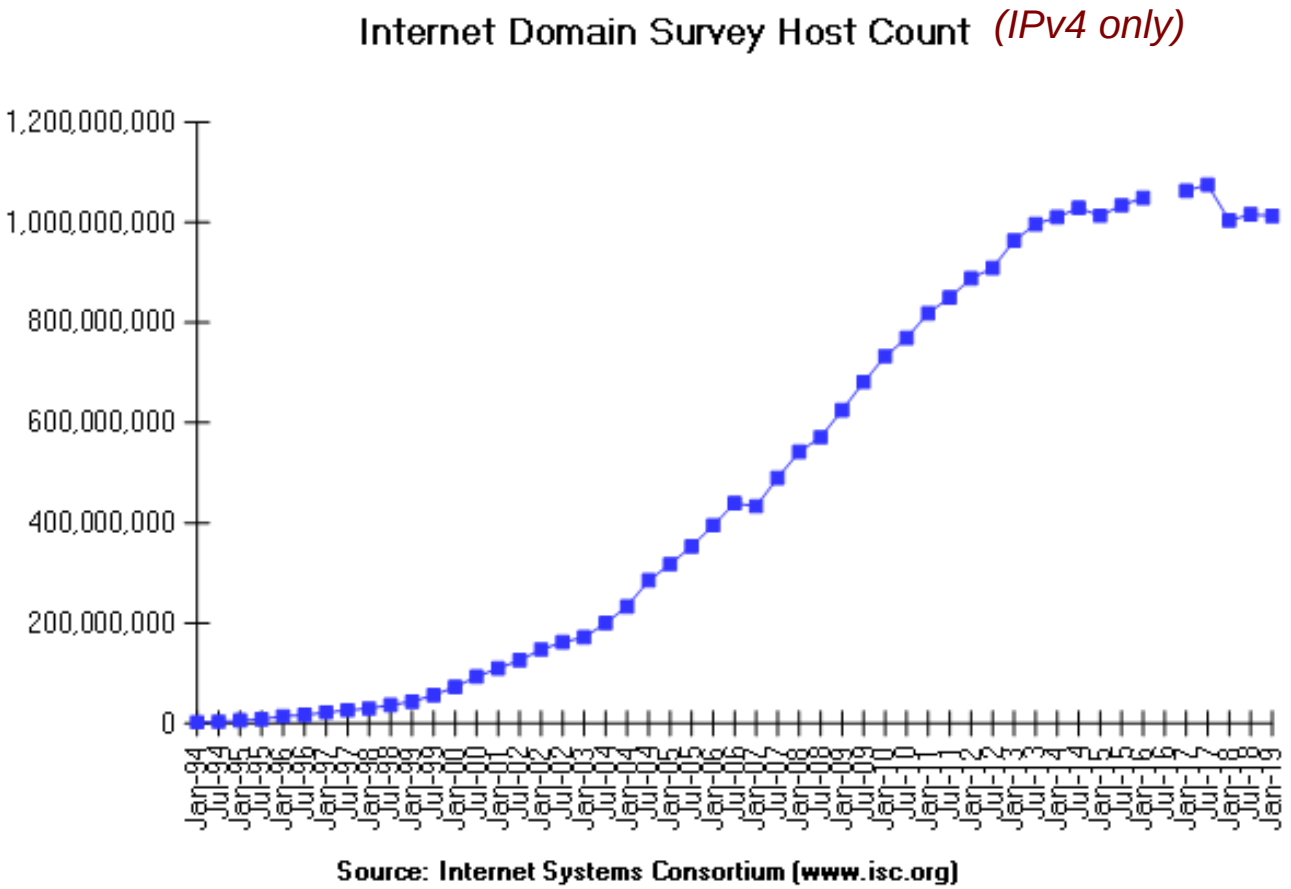
- Un sistema es escalable si conserva su efectividad ante un incremento significativo del
  - Número de recursos
  - Número de usuarios

- Puntos clave
  - Control del coste de los recursos HW
  - Prevención del desbordamiento de recursos SW
    - P. ej: paso de IPV4 (2^{34} direcciones IP) a IPV6 (2^{128} direcciones IP)*
  - Evitar cuellos de botella/pérdidas en prestaciones
    - Algoritmos descentralizados
    - Replicación
    - Uso de cachés

**En 1998 ya se había diseñado la solución al problema de escala de IPv4. ¿En qué medida es, a día de hoy, IPv6 una realidad?*

Escalabilidad

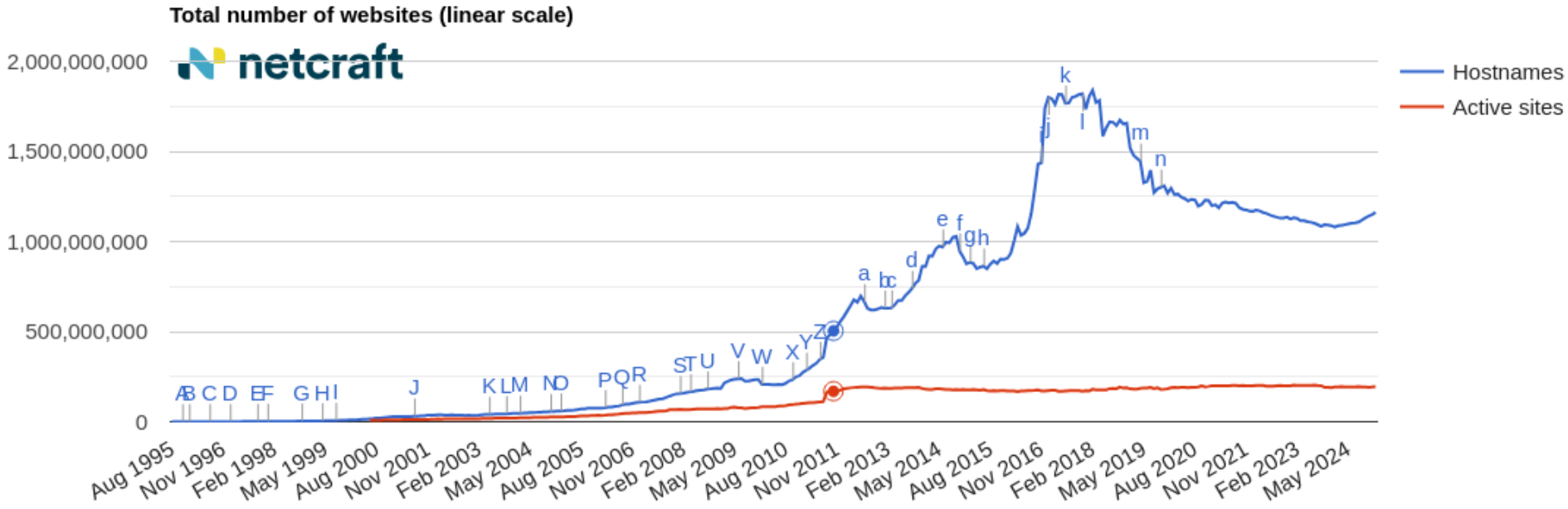
Internet



https://www.isc.org/network/survey/

Escalabilidad

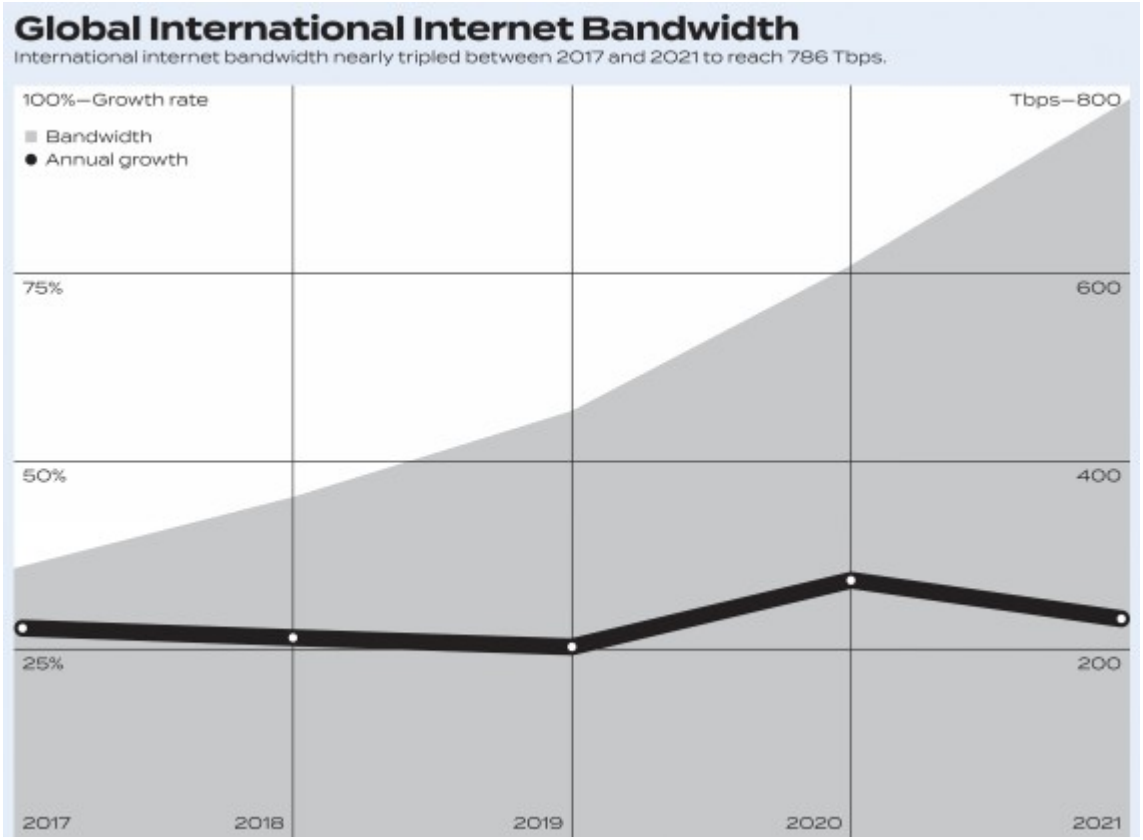
Internet



https://news.netcraft.com/archives/category/web-server-survey/

Escalabilidad

Internet



https://global-internet-map-2022.telegeography.com/

5.- Tratamiento de fallos

- En un sistema distribuido, los fallos siempre son parciales
- Detección de fallos
  - Checksum para fallos en la transmisión
  - Detección de la caída de servidores
- Enmascaramiento de fallos
  - Retransmisión de mensajes fallidos
  - Servidores proxy
- Tolerancia a fallos → Redundancia
  - Rutas alternativas entre routers
  - Sistemas de nombres duplicado
  - Replicación de ficheros
- Disponibilidad:
  - Proporción de tiempo que un sistema está operativo

6.- Concurrencia

- Cada objeto que represente un recurso compartido en un sistema distribuido debe responsabilizarse de garantizar que opera correctamente en un entorno concurrente
  - Algunos objetos deberán reimplementarse para trabajar correctamente en entornos distribuidos
    - Servidores multihilo
    - Sincronización mediante semáforos u otros mecanismos

7.- Transparencia

- Ocultación al usuario y al programador de aplicaciones de los componentes de un sistema distribuido
  - El sistema se percibe como un todo, en vez de como una colección de componentes independientes

¿Crees que se puede conseguir una transparencia perfecta? Es decir ¿crees que puede ejecutarse un método remoto como si fuera local? Razona tu respuesta

Transparencia

Tipos

- De acceso
  - Se accede mediante igual mecanismo a recursos locales y remotos
- De ubicación
  - Se accede a los recursos sin necesidad de conocer su localización
- De concurrencia
  - Varios procesos operan concurrentemente sin interferencia mutua
- De replicación
  - Uso de múltiples ejemplares de cada recurso para aumentar fiabilidad y prestaciones sin que los usuarios necesiten su conocimiento

¿Se te ocurre algún sistema distribuido que cumpla con alguno de estos tipos de transparencia?

Transparencia

Tipos (ii)

- Frente a fallos
  - Ocultación de fallos dejando que el usuario o programa de aplicación complete sus tareas a pesar de fallos HW o SW
- Movilidad
  - Reubicación de recursos y clientes en un sistema sin afectar la operación de los usuarios y programas
- Prestaciones
  - Reconfiguración del sistema para mejorar las prestaciones según varíe la carga de uso
- Escalado
  - Expansión en tamaño del sistema o aplicaciones sin cambiar la estructura subyacente o los algoritmos de aplicación

Desafíos

Contexto

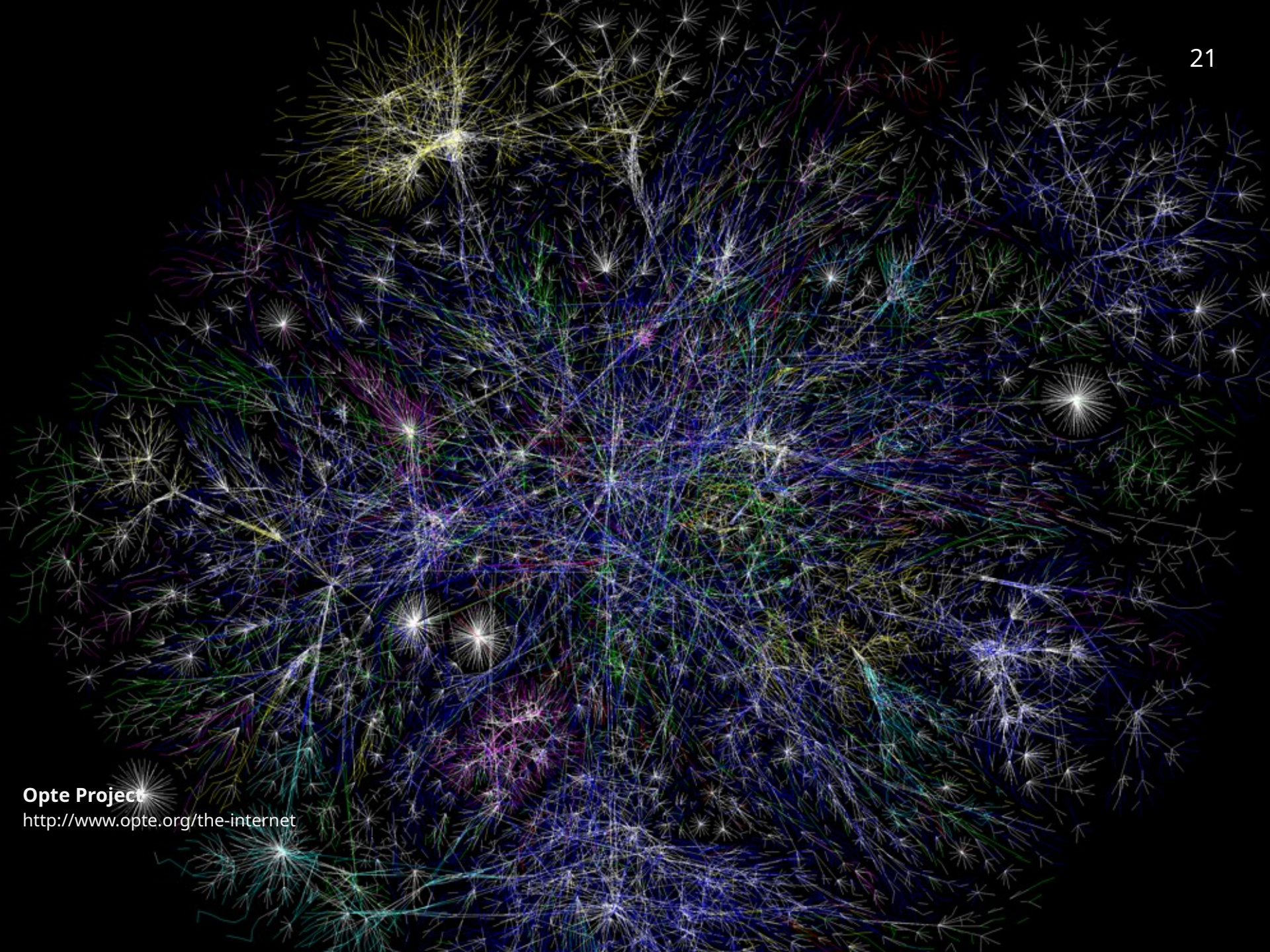
- Todos los desafíos son importantes
- Dependiendo del contexto unos van a tener más relevancia que otros
  - Transacciones comerciales → seguridad, fallos, concurrencia
  - Google, P2P, DNS → escalabilidad, fallos
- Cuando estemos estudiando un problema, a menudo:
  - Nos centraremos en un desafío
  - Dando por ‘solventados’ algunos desafíos
  - Analizando cómo repercute la solución en algunos otros desafíos

Resumen

- En un sistema distribuido (SD) los componentes están unidos mediante una **red** y se comunican mediante **paso de mensajes.**
- En un SD pueden convivir muchas plataformas, una **heterogeneidad** que trata de resolverse mediante **middleware.**
- En algunos casos es importante diseñar SDs que sean **escalables**, es decir, toleren un incremento en el número de recursos o usuarios. Especialmente importante si la escala es Internet.
- En un SD puede haber **fallos a muchos niveles** que en la medida de lo posible deben ser ocultados o tolerados.
- En un SD varios procesos pueden acceder a los mismos recursos a la vez. Esta **concurrencia** debe ser identificada y tratada adecuadamente.
- La **seguridad** y la **extensibilidad** son otros desafíos importantes, pero no nos centraremos en ellos
- Un tratamiento correcto de estos desafíos lleva a un sistema. **transparente** en el que el usuario final no percibe estos aspectos.
- Es importante definir el **contexto** en el que nos encontramos para identificar los desafíos clave, o los que vamos a necesitar enfrentar en nuestro **modelo** de SD.

Referencias

- G. Colouris, J. Dollimore, T. Kindberg and G. Blair. *Distributed Systems: Concepts and Design (5th Ed)*. Addison-Wesley, 2011
  - Capítulo 1
- Internet World Stats
  - http://www.internetworldstats.com
- Internet Systems Consortium
  - http://www.isc.org



Opte Project
http://www.opte.org/the-internet