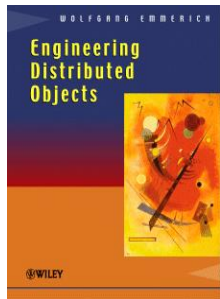


# Capítulo I – Introdução aos Sistemas Distribuídos

**From: Wolfgang Emmerich**

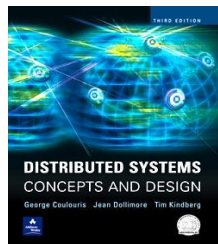
**Engineering Distributed Objects**

John Wiley & Sons, Ltd 2000



**From: Coulouris, Dollimore and Kindberg**  
**Distributed Systems: Concepts and Design**

Edition 4 © Addison-Wesley 2005



Paula Prata,

Departamento de Informática da UBI

<http://www.di.ubi.pt/~pprata>

## O que é um sistema distribuído?

Conjunto de computadores ligados em rede, com software que permita a partilha de recursos e a coordenação de actividades, oferecendo idealmente um sistema integrado.

### Características:

- Comunicação através de mensagens
- Concorrência
- Partilha de recursos
- Sistema Assíncrono
- Falhas Independentes
- Heterogeneidade

## O que é um sistema distribuído?

Características:

### - Comunicação através de mensagens

Os componentes do sistema comunicam através de mensagens

(não existem variáveis globais partilhadas)

modelos de programação: cliente/servidor, modelo baseado em objectos, ...

### Partilha de recursos

impressoras, discos, ferramentas para trabalho cooperativo, bases de dados, ...

A partilha de recursos levanta questões de segurança

Gestores de recursos controlam o acesso a recursos partilhados

### - Concorrência: os vários utilizadores usam o sistema em simultâneo

(é necessário coordenar o acesso aos recursos partilhados: hw, sw, dados)

# O que é um sistema distribuído?

## Características:

### - Sistema Assíncrono

- não existe um relógio global
- diferentes velocidades de processamento
- não existe um limite para o tempo de comunicação

### - Falhas Independentes

- falhas na rede (perda de mensagens, duplicação, reordenação)
- falhas em unidades de processamento

=> a falha de um componente não impede necessariamente os outros de funcionar

## O que é um sistema distribuído?

Características:

### - Heterogeneidade

Um sistema distribuído pode possuir:

- . diferentes tipos de rede
- . diferentes tipos de hardware  
(diferentes representações de dados, diferente código máquina)
- . diferentes sistemas operativos  
(diferentes interfaces para os protocolos de comunicação)
- . diferentes linguagens de programação  
(diferentes representações de estruturas de dados como arrays ou registos,...)

Para tentar resolver o problema da heterogeneidade define-se uma camada de software intermédia: *middleware*

# The Eight Fallacies of Distributed Computing

“Essentially everyone, when they first build a distributed application, makes the following eight assumptions.

All prove to be false in the long run and all cause big trouble and painful learning experiences.”

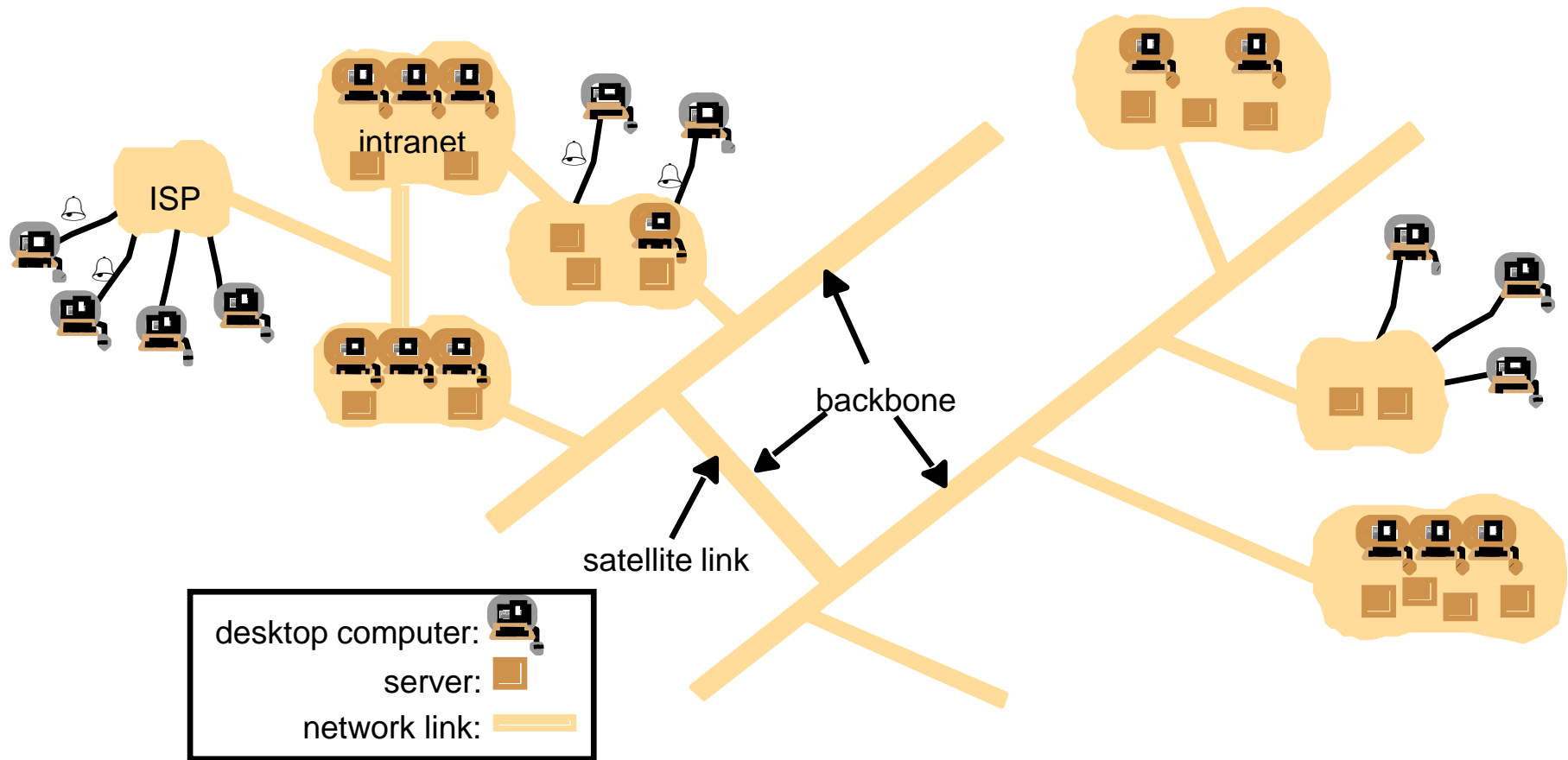
Peter Deutsch

first published the "8 Fallacies of Networking" internally while working at Sun Labs in 1991-92.

1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

*In January 1991 the first Web servers outside CERN itself were switched on.*

# A Internet: infraestrutura de redes(s) que liga milhões de computadores



O desenho e construção dos mecanismos de comunicação da Internet (protocolos Internet) permitiu que um programa em execução num qualquer ponto da rede possa enviar mensagens a programas em qualquer outro lugar.

World Wide Web (Web) [Berners-Lee 89] : Modelo de partilha de informação construído no topo da internet. Conjunto de servidores web acedidos através da internet a partir de web browsers.

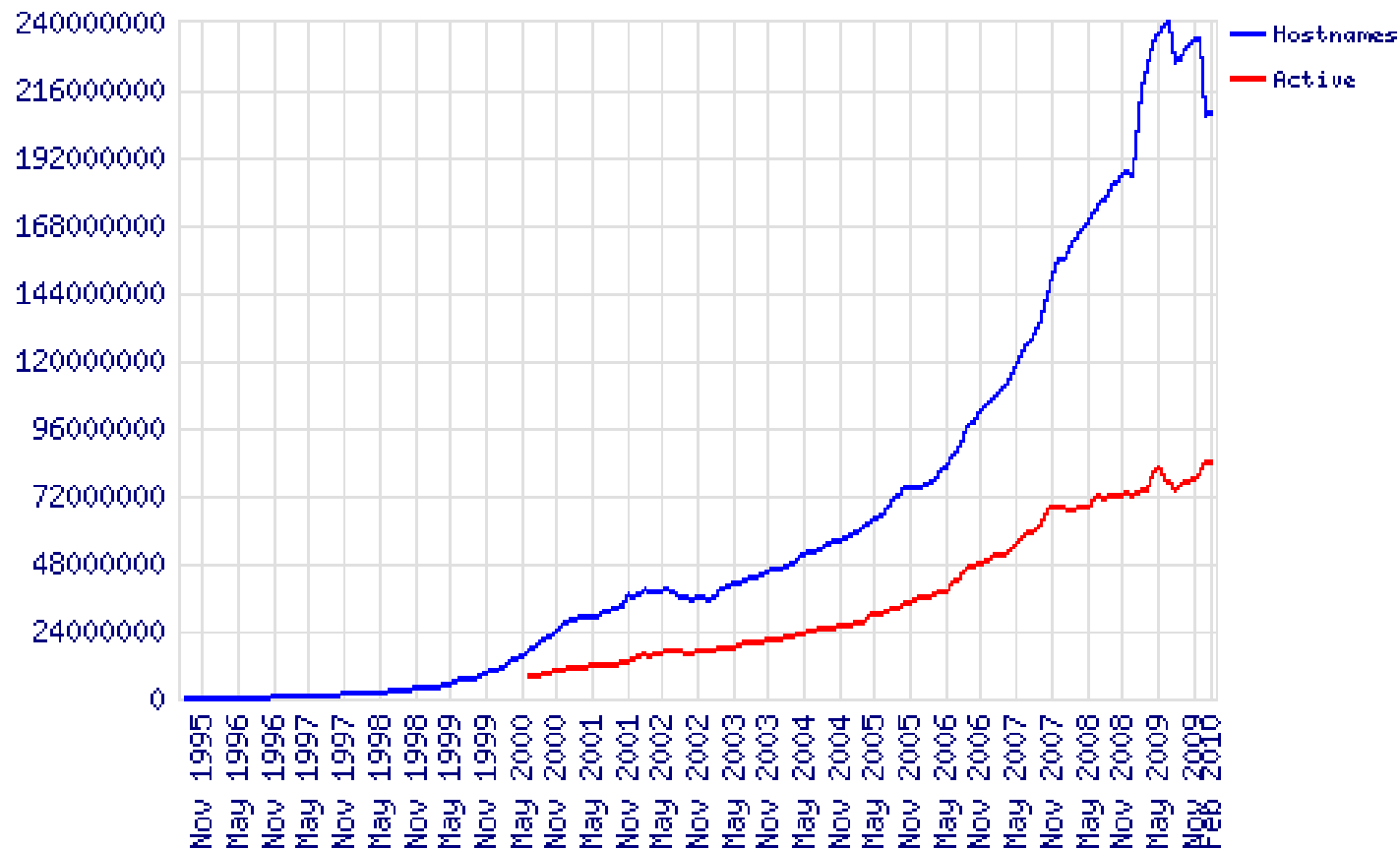
### Computadores na internet

<i>Date</i>	<i>Computers</i>	<i>Web servers</i>
1979, Dec.	188	0
1989, July	130,000	0
1999, July	56,218,000	5,560,866

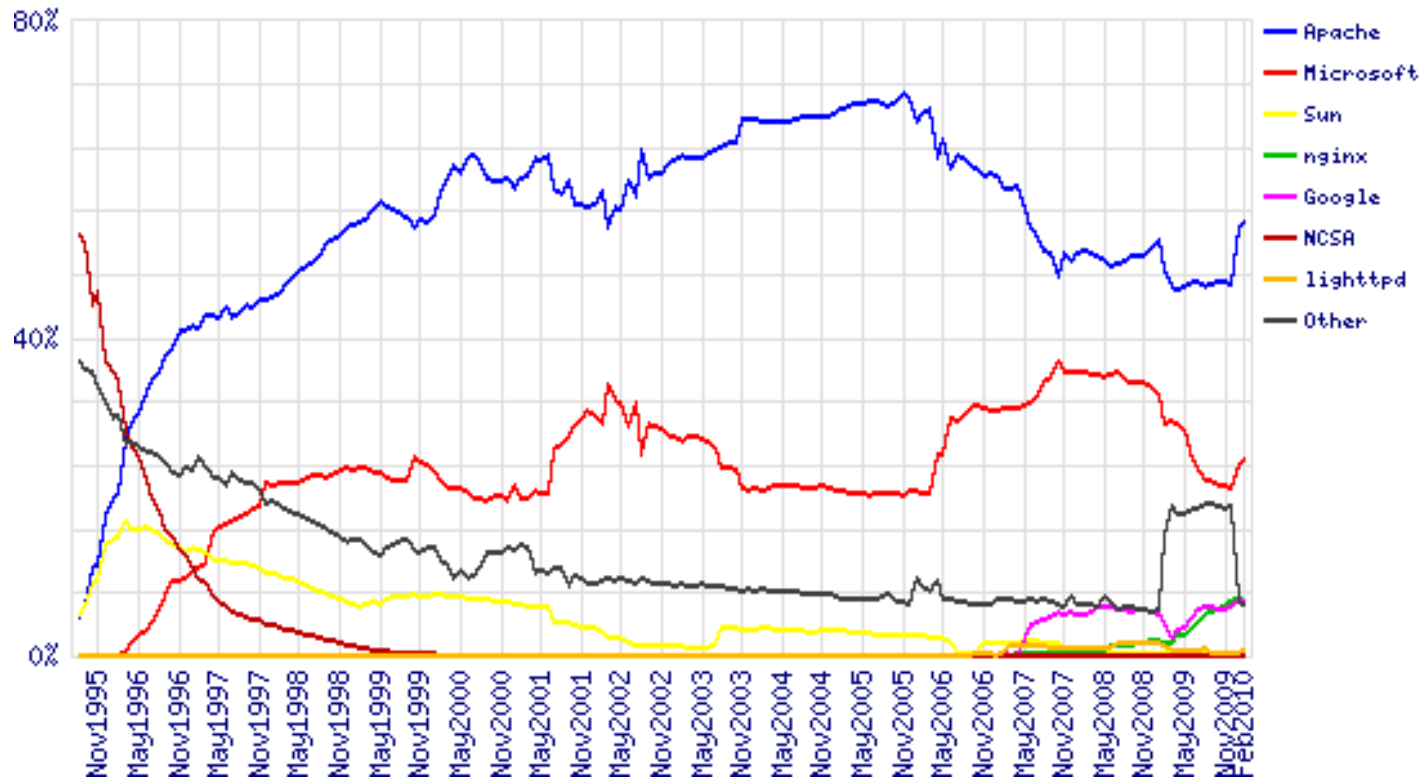


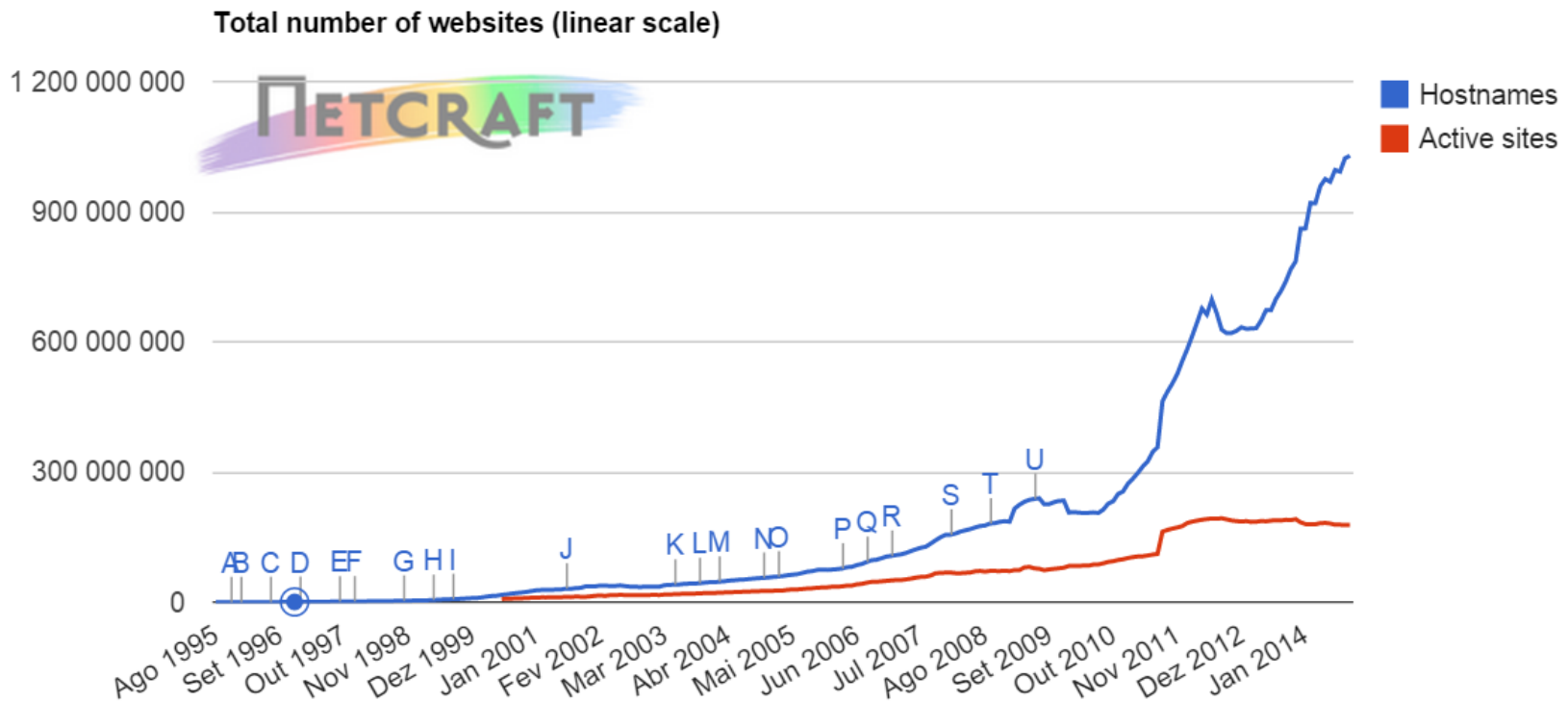
## Exemplos de sistemas distribuídos: aplicações web

### Total Sites Across All Domains August 1995 - February 2010



# Market Share for Top Servers Across All Domains August 1995 - February 2010



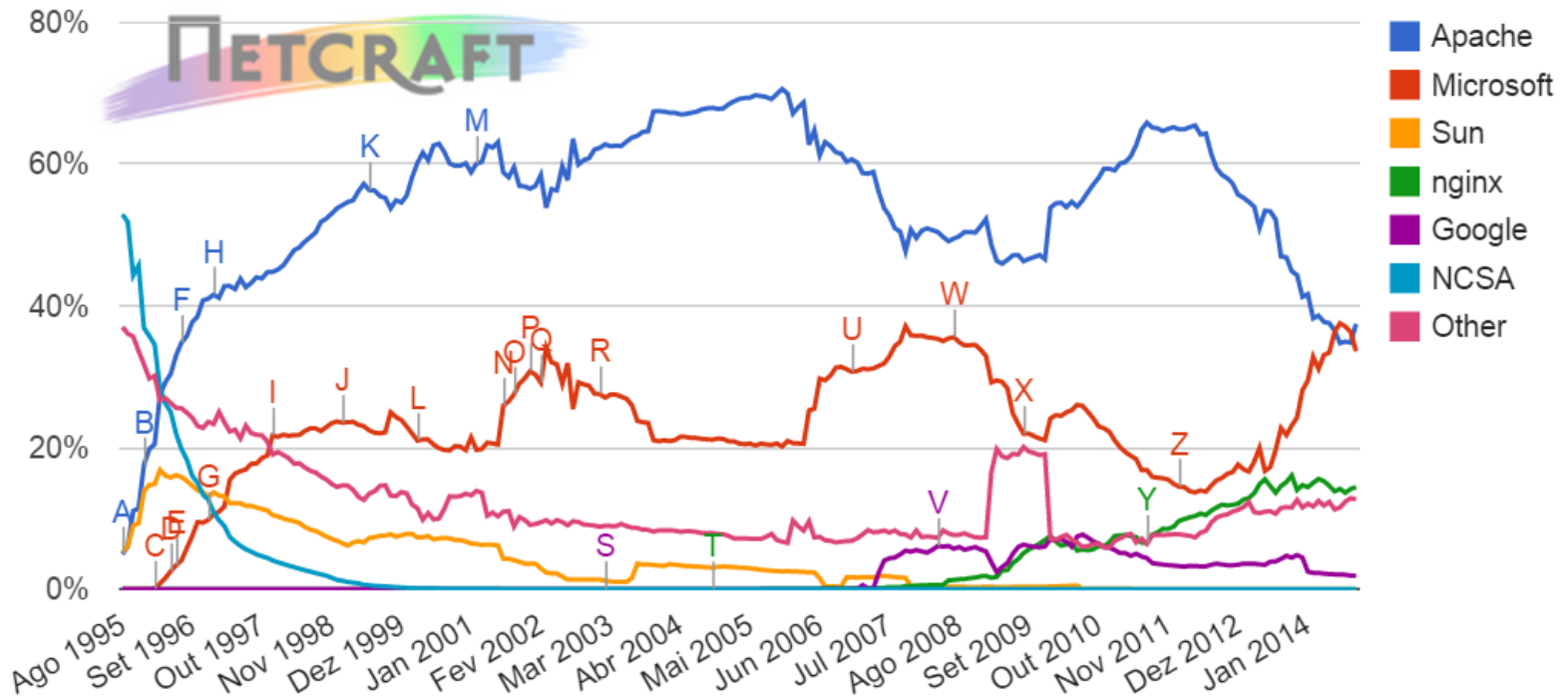


Outubro 2014 – Hostnames: 1 028 932 208, Active Sites: 178 174 498

From

<http://news.netcraft.com/archives/2014/10/10/october-2014-web-server-survey.html>

Web server developers: Market share of all sites



From

<http://news.netcraft.com/archives/2014/10/10/october-2014-web-server-survey.html>

<b>Developer</b>	<b>September 2014</b>	<b>Percent</b>	<b>October 2014</b>	<b>Percent</b>	<b>Change</b>
Apache	355,925,985	34.79%	385,354,994	37.45%	2.66
Microsoft	371,406,909	36.31%	345,485,419	33.58%	-2.73
nginx	144,717,670	14.15%	148,330,190	14.42%	0.27
Google	19,499,154	1.91%	19,431,026	1.89%	-0.02

From

<http://news.netcraft.com/archives/2014/10/10/october-2014-web-server-survey.html>

<b>Developer</b>	<b>December 2014</b>	<b>Percent</b>	<b>January 2015</b>	<b>Percent</b>	<b>Change</b>
Apache	358,159,405	39.11%	348,460,753	39.74%	0.63
Microsoft	272,967,294	29.81%	241,276,347	27.52%	-2.29
nginx	132,467,763	14.47%	128,083,920	14.61%	0.14
Google	20,011,260	2.19%	20,209,649	2.30%	0.12

From

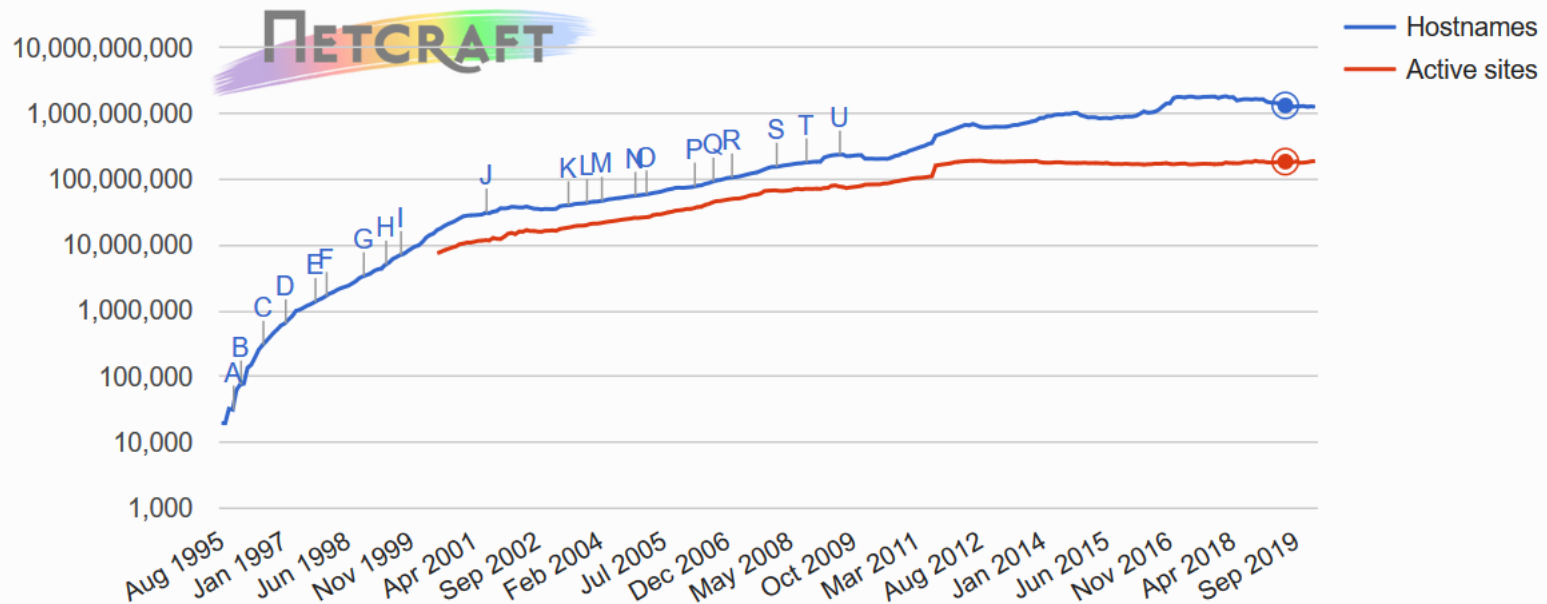
<http://news.netcraft.com/archives/2014/10/10/october-2014-web-server-survey.html>

Developer	December 2015	Percent	January 2016	Percent	Change
Apache	320,676,759	35.59%	304,271,061	33.56%	-2.03
Microsoft	239,927,013	26.63%	262,471,886	28.95%	2.32
nginx	157,001,018	17.43%	141,443,630	15.60%	-1.82
Google	20,362,678	2.26%	20,799,087	2.29%	0.03

From

<http://news.netcraft.com/>

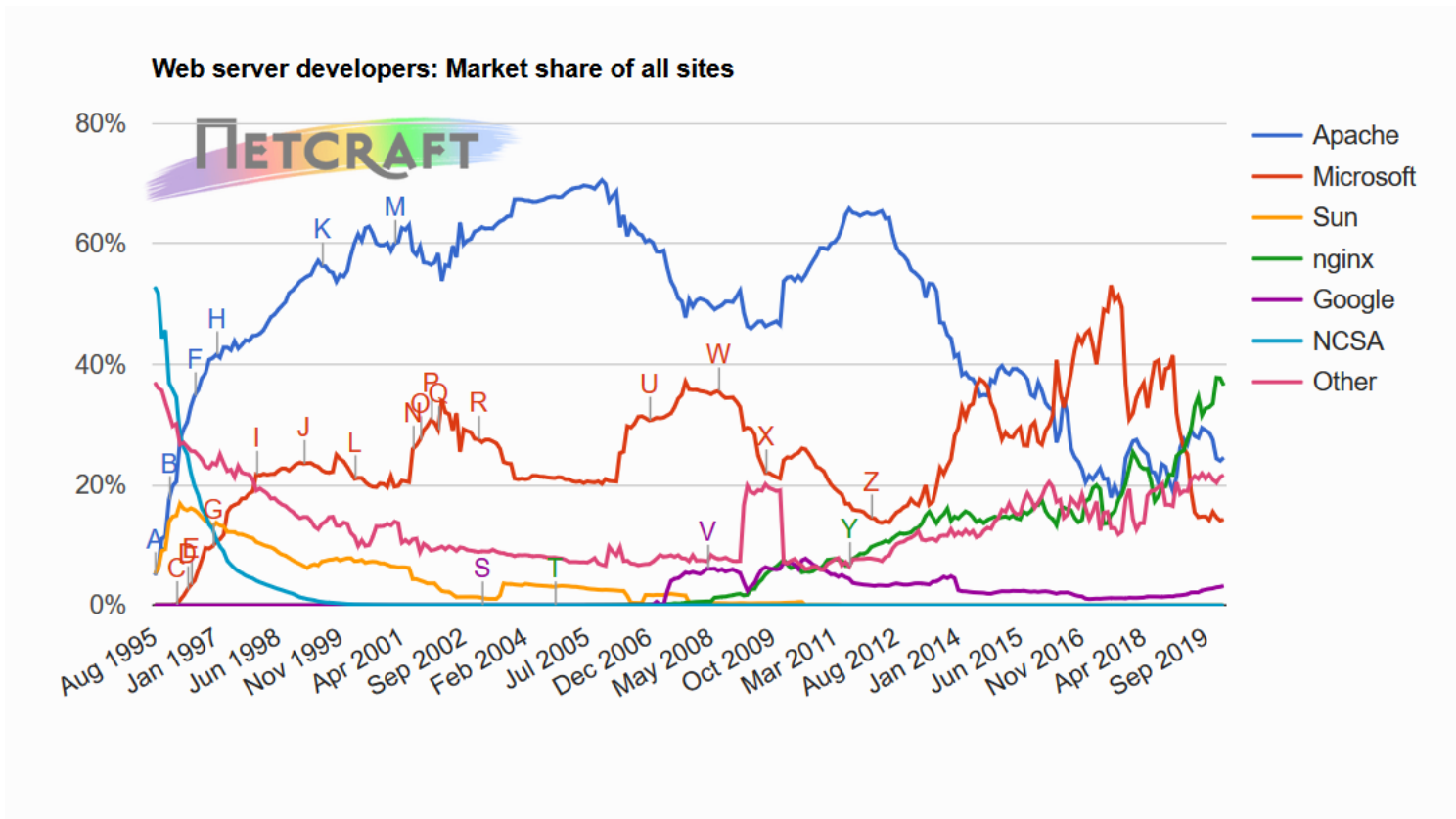
Total number of websites (logarithmic scale)



From

<http://news.netcraft.com/>





From

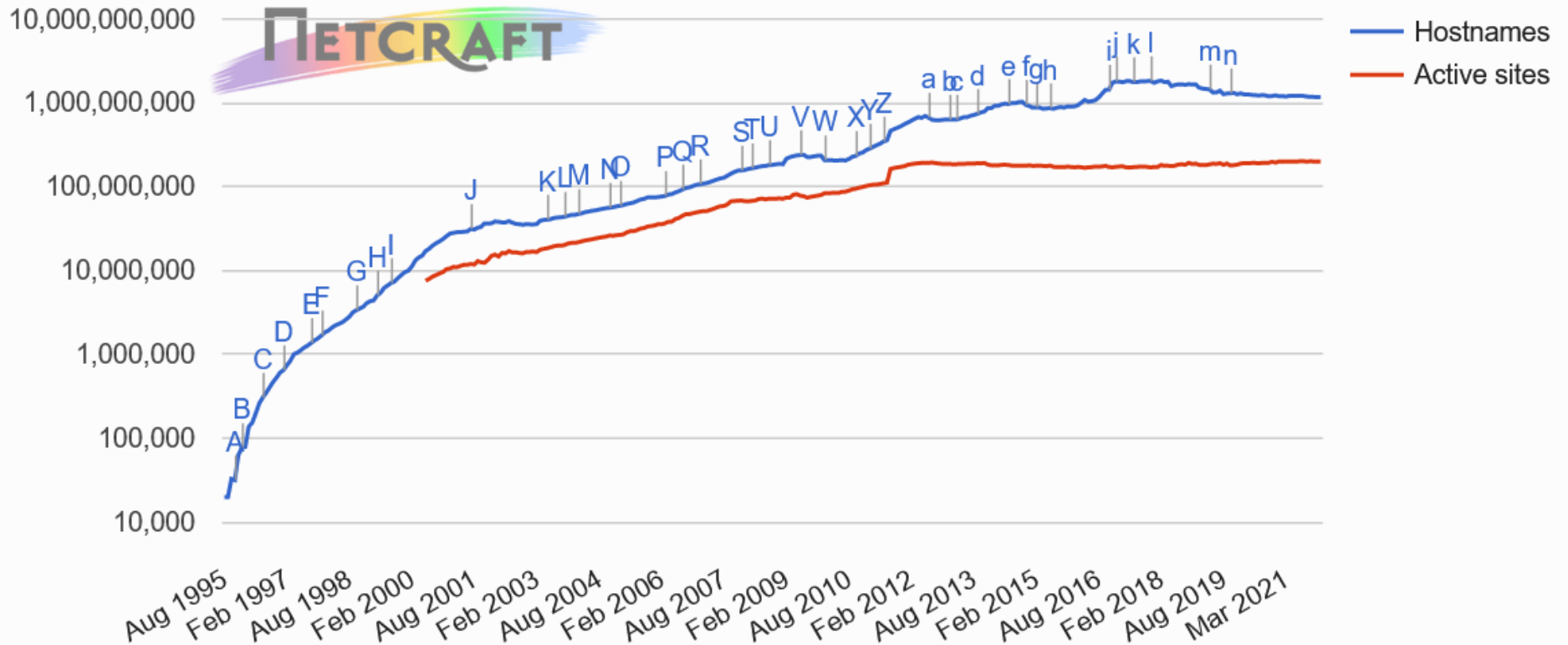
<http://news.netcraft.com/>

<b>Developer</b>	<b>January 2020</b>	<b>Percent</b>	<b>February 2020</b>	<b>Percent</b>	<b>Change</b>
nginx	488,628,547	37.70%	459,966,569	36.48%	-1.22
Apache	310,833,084	23.98%	309,061,300	24.51%	0.53
Microsoft	181,873,181	14.03%	179,225,073	14.21%	0.18
Google	39,081,956	3.02%	40,120,733	3.18%	0.17

From

<http://news.netcraft.com/>

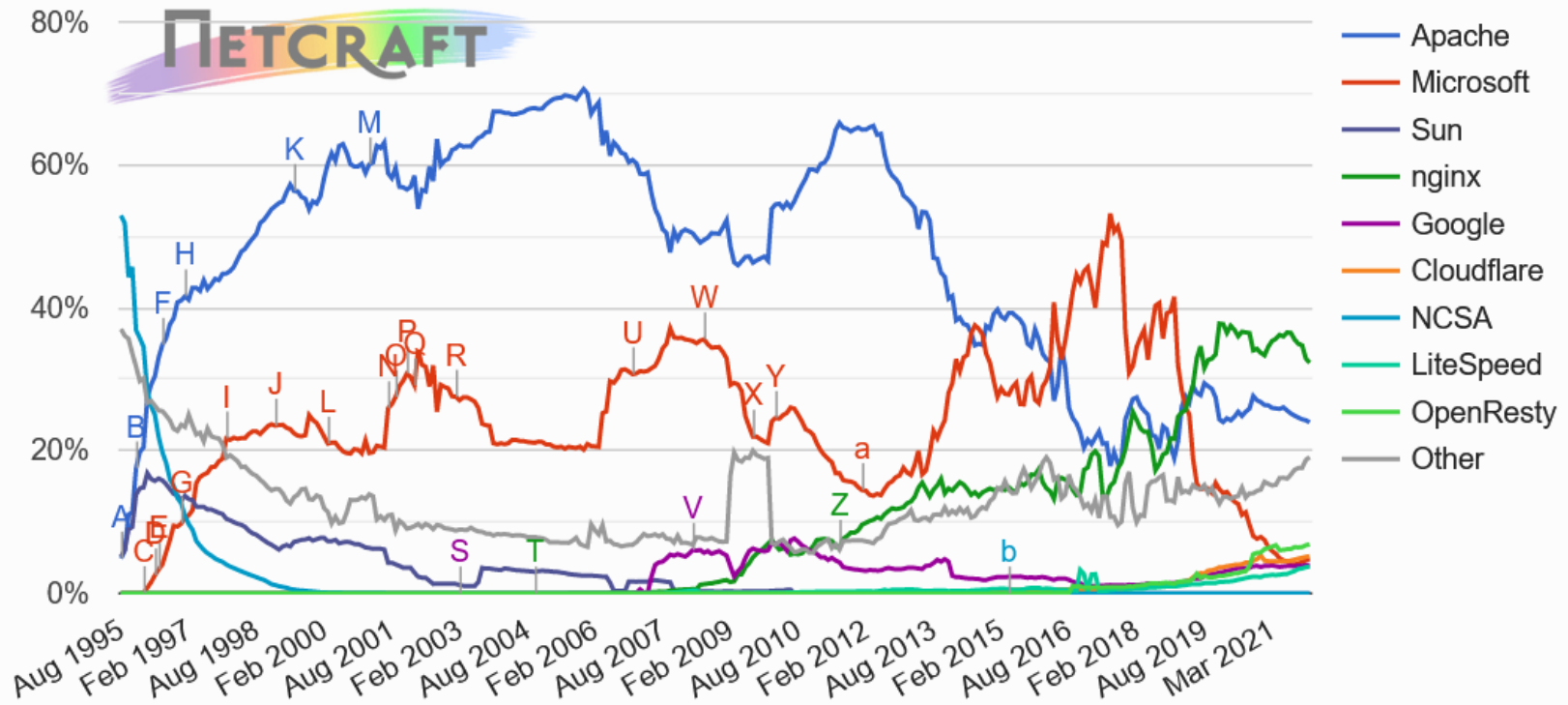
Total number of websites (logarithmic scale)



From

<http://news.netcraft.com/>

### Web server developers: Market share of all sites



From

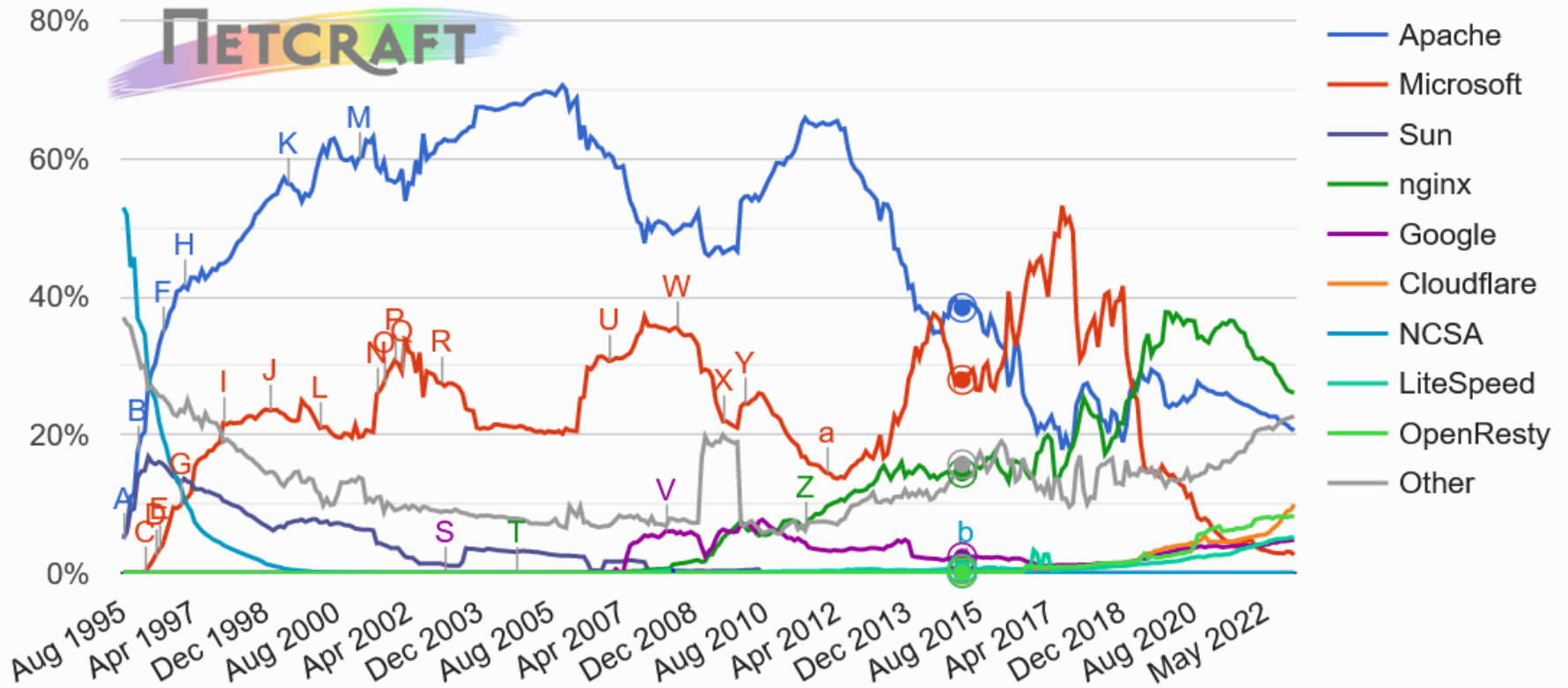
<http://news.netcraft.com/>

<b>Developer</b>	<b>December 2021</b>	<b>Percent</b>	<b>January 2022</b>	<b>Percent</b>	<b>Change</b>
nginx	384,347,394	32.88%	377,019,054	32.29%	-0.60
Apache	283,409,491	24.25%	279,709,815	23.95%	-0.29
OpenResty	78,902,138	6.75%	80,238,470	6.87%	0.12
Cloudflare	59,904,450	5.13%	60,881,028	5.21%	0.09

From

<http://news.netcraft.com/>

### Web server developers: Market share of all sites



From

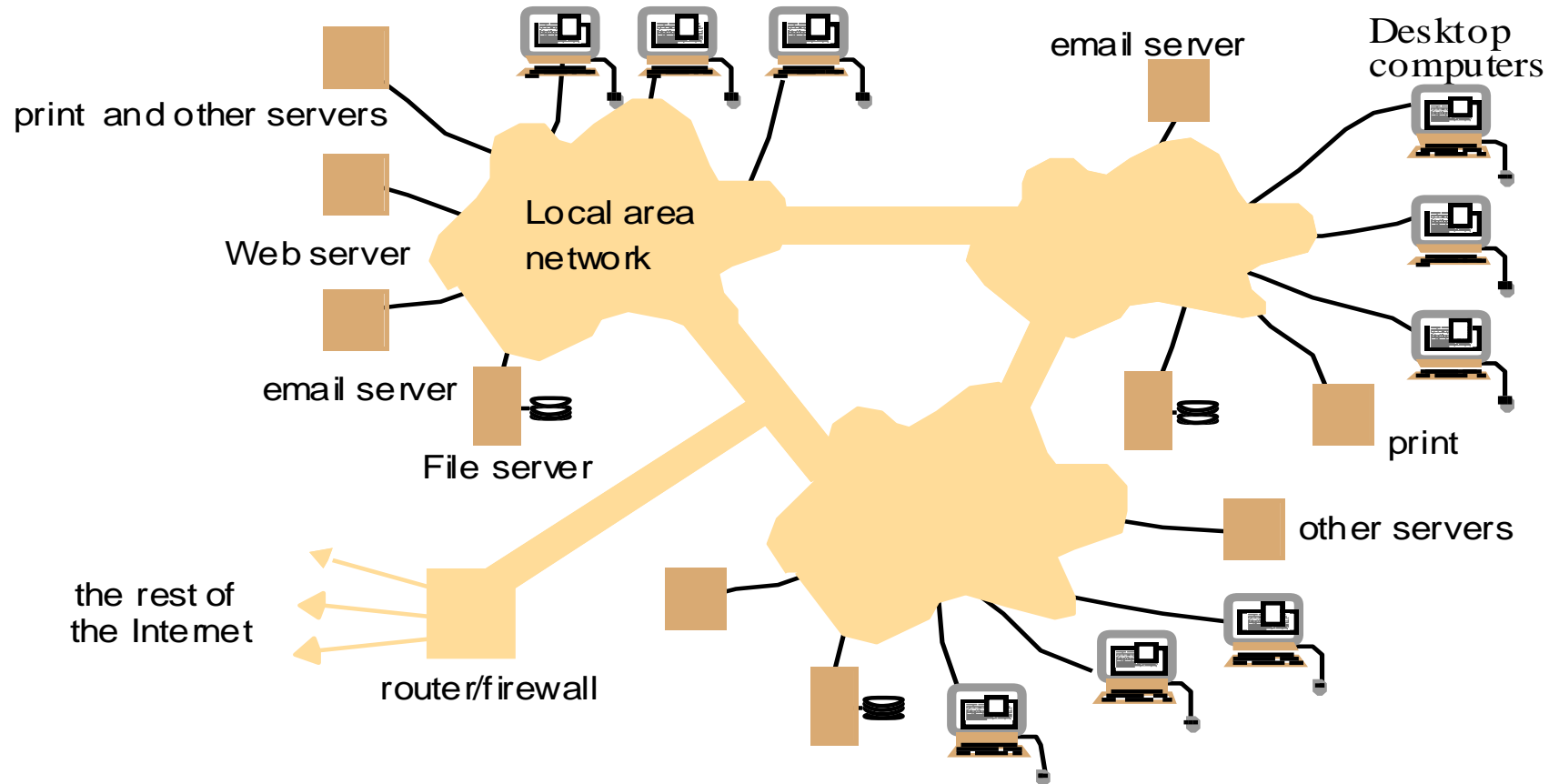
<http://news.netcraft.com/>

<b>Developer</b>	<b>December 2022</b>	<b>Percent</b>	<b>January 2023</b>	<b>Percent</b>	<b>Change</b>
nginx	295,366,783	26.25%	295,678,304	26.11%	-0.13
Apache	235,541,408	20.93%	233,636,177	20.63%	-0.30
Cloudflare	102,829,746	9.14%	112,159,331	9.91%	0.77
OpenResty	92,711,293	8.24%	92,291,824	8.15%	-0.09

From

<http://news.netcraft.com/>

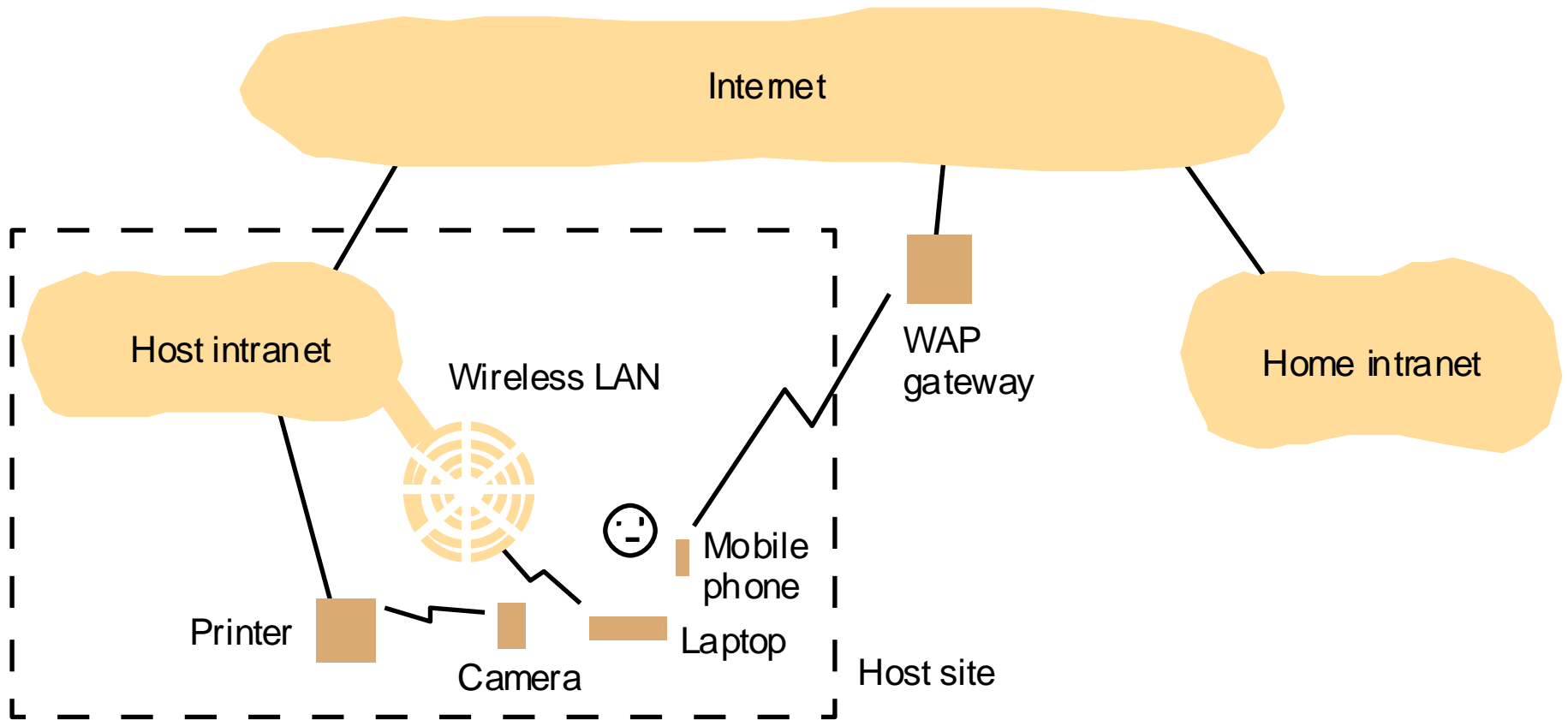
# Intranets





# Computação Móvel e Ubíqua

Integração de equipamento portátil em sistemas distribuídos



## Computação Móvel e Ubíqua

- . Computadores portáteis
- . Telefones móveis
- . Máquinas fotográficas digitais
- . PDA's - Personal Digital Assistants
- . Pagers
- ...
- . “wearable devices”, relógios com processador, ...
- . Sistemas embebidos (= computadores dentro de um produto)  
electrodomésticos, carros, ...

=> O termo computação Ubíqua pretende designar sistemas cuja utilização está de tal forma integrada na funcionalidade do produto que é transparente para o utilizador.

=> Maiores restrições em termos de: custo, tamanho, potência/autonomia

## Outros Sistemas Distribuídos:

- . Correio electrónico
- . Sistemas de ficheiros distribuídos
- . FTP, Telnet
- . Chat
- . I.M.
- ...

### **Aplicações críticas:**

- . Reserva de bilhetes em companhias de transportes
- . Comércio electrónico
- . Máquinas Multibanco
- ...

=> Exigem fiabilidade e segurança

## Desafios na implementação de sistemas distribuídos:

- Escalabilidade
- Abertura
- Tolerância a Falhas
- Segurança
- Transparência

### Escalabilidade

Capacidade de o sistema se manter a funcionar de forma correcta e à velocidade desejada independentemente do número de utilizadores.

É necessário:

- desenhar o software de forma a que o aumento de utilizadores não exija grandes alterações
- evitar algoritmos e estruturas de dados centralizadas (replicação de dados se necessário)

## Desafios na implementação de sistemas distribuídos:

### Escalabilidade (cont)

- controlar o aumento de custos devido à disponibilização de mais recursos
- controlar a perda de performance

(replicação de serviços)

- evitar o transbordo de certos limites de recursos (ex. Endereço IP com 32 bits, insuficiente)

### Sistema Aberto (“Openness”)

Capacidade de o sistema ser extensível, quer em hardware quer em software

- novos componentes devem poder ser adicionados sem por em causa o funcionamento dos já existentes, e poder comunicar com eles

## Desafios na implementação de sistemas distribuídos:

### Sistema Aberto (cont)

Para isso é importante que:

- sejam conhecidas as interfaces dos novos componentes através da publicação da sua documentação
- utilizar protocolos e formatos standard

Exemplo de publicação de interfaces:

Request For Comment (RFCs) [www.ietf.org](http://www.ietf.org)

Contém as especificações dos protocolos internet desde o início dos anos 80

### Tolerância a Falhas

Tolerar uma falha significa conter os seus efeitos de forma a que o sistema continue a funcionar.

## Desafios na implementação de sistemas distribuídos:

### Tolerância a Falhas (cont)

Para isso é necessário:

#### 1) Detecção da falha

Ex.lo: Dados corrompidos (mensagens ou ficheiros) podem ser detectados através de somas de verificação.

#### 2) Localização da falha

Se não houve resposta a um pedido, o que significa?

- falha na rede
- falha no nó destino

como distinguir ?

## Desafios na implementação de sistemas distribuídos:

### Tolerância a Falhas (cont)

#### 3) Mascarar / Tolerar a falha

Algumas falhas podem ser ocultadas do utilizador se se utilizar redundância suficiente:

- quando uma mensagem não chega, pode ser retransmitida
- um ficheiro pode ser escrito em duplicado (um em cada disco)
- entre cada dois “routers” da internet, devem sempre existir dois percursos
- uma base de dados pode ser replicada em vários servidores



## Desafios na implementação de sistemas distribuídos:

### Segurança

Manter recursos computacionais seguros significa:

- Manter o nível de confidencialidade exigido pelos utilizadores  
(proteção contra acessos não autorizados)
- Garantir a integridade dos dados  
(proteção contra alteração ou corrupção de dados ou programas)
- Manter a disponibilidade do sistema  
(proteção contra interferências aos meios de acesso aos recursos)

Alguns problemas por resolver:

- Ataques do tipo “negação de serviço” (denial of service)
- Segurança do código móvel

## Desafios na implementação de sistemas distribuídos:

### Transparência

O sistema deve ser visto como um todo e não como uma coleção de componentes distribuídos.

No standard de “Open Distributed Processing (ODP) foram definidos os seguintes tipos de transparência:

- Acesso
- Localização
- Concorrência
- Replicação
- Falhas
- Migração
- Desempenho
- Escalabilidade

Mais Importantes:

Acesso e Localização em conjunto são referidos como transparência de rede

## Desafios na implementação de sistemas distribuídos:

### Transparência de acesso:

- permite que o acesso a recursos locais e a recursos remotos seja feito através das mesmas operações (i.é, usando a mesma interface).

(ex. WinSCP vs FTP)

### Transparência de localização:

- permite que os recursos possam ser acedidos sem o conhecimento da sua localização.

(ex. Programas de correio electrónico)

### Transparência de concorrência:

- permite que os vários clientes de um componente não necessitem de ter em conta o acesso concorrente ao componente.

## Desafios na implementação de sistemas distribuídos:

### Transparência de replicação:

- permite que os clientes de um componente não se apercebam se existe replicação e estão a usar uma réplica e não o original;

A utilização de várias instâncias de um componente pode ocorrer por:

razões de desempenho ou de fiabilidade;

Os utilizadores do componente não necessitem de saber que o componente possa ser replicado.

### Transparência de Falhas:

- permite que o sistema funcione na presença de falhas de hardware ou software sem que utilizadores saibam como as falhas foram ultrapassadas.

(por ex. um sistema de e-mail pode retransmitir uma mensagem até que a mesma seja entregue com sucesso)

## Desafios na implementação de sistemas distribuídos:

### Transparência de Migração:

- permite que um recurso possa mudar de localização sem que isso afete a sua utilização.

(ex. Telemóveis em movimento)

### Transparência de Desempenho:

- permite que o sistema seja reconfigurado para melhorar o seu desempenho sem que os utilizadores se apercebam.

### Transparência de Escalabilidade:

- permite que o sistema seja expandido sem que os utilizadores se apercebam de como isso foi conseguido.