

La distribution de contenu dans l'Internet

1 – Les technologies du Web

Christophe Deleuze

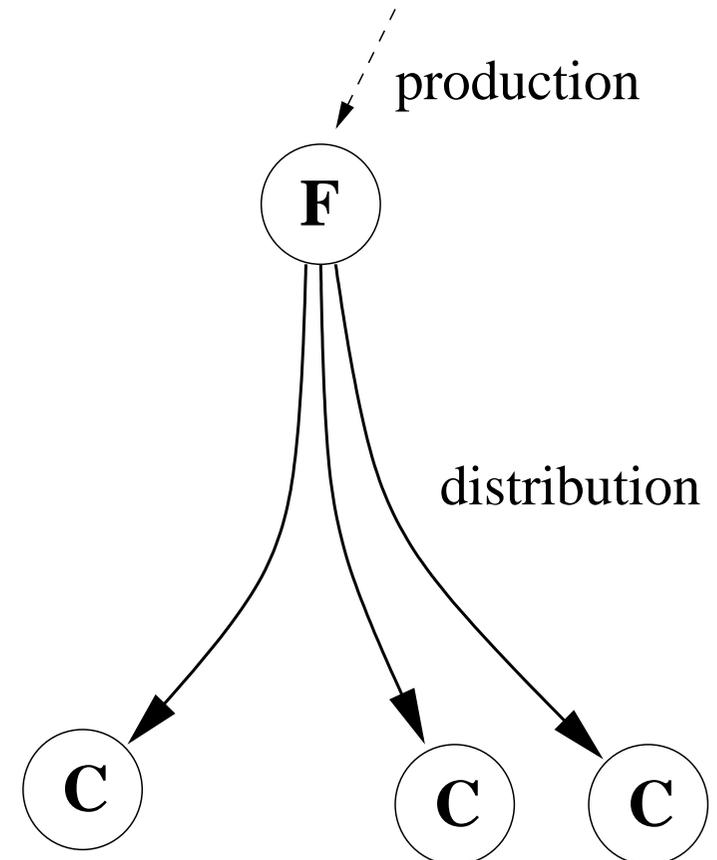
`Christophe.Deleuze@free.fr`

ENST Paris

janvier 2004

La distribution de contenu...

- contenu
 - ✓ produit professionnellement
 - ✓ livres
 - ✓ film, musique
 - ✓ ...
- production
- distribution



...dans l'Internet

- contenu numérique
- “convergence numérique”
 - ✓ '80-'90 RNIS-large bande
 - ✓ '00 web

1. Les technologies du web (2h15)
2. L'usage des proxies dans le web (2h)
3. Les réseaux d'acheminement de contenu (CDN) (3h)
4. Réseaux pair à pair, nommage, ... (1h30)

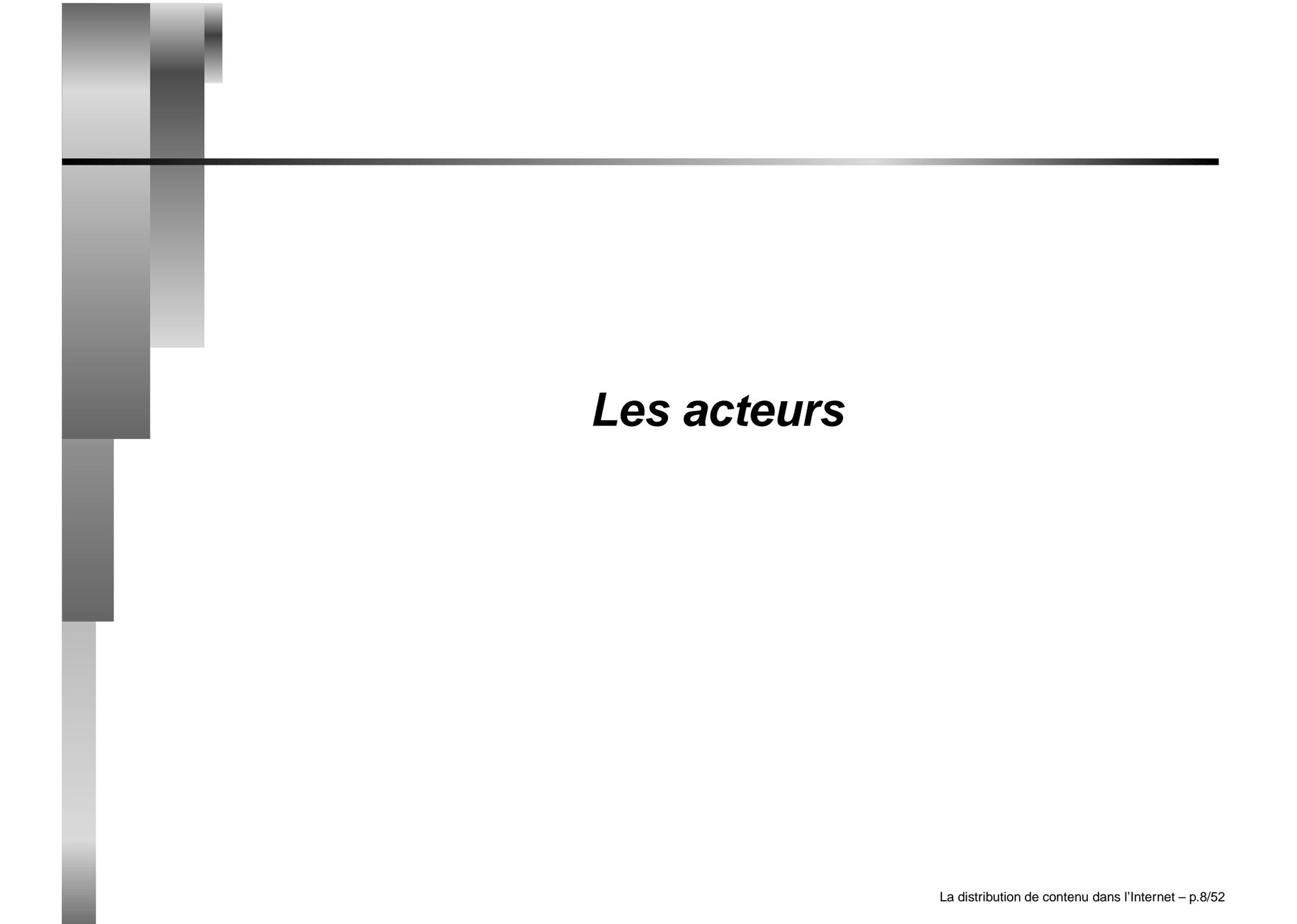
L'Internet avant le web

- réseau à commutation de paquets
- TCP/IP
- applications
 - ✓ courrier électronique
 - ✓ telnet
 - ✓ FTP (44 % trafic en '91)
 - ✓archie
 - ✓ gopher
 - ✓ ...

- 1989/91 CERN
- Tim Berners-Lee
- outil de travail collaboratif
- hypertexte
- '93 Mosaic, 1^{er} browser graphique
- fondation Netscape 1 an plus tard
- unification interface
 - ✓ web
 - ✓ email
 - ✓ forums (news)
 - ✓ FTP
- unification applications (HTTP/XML)

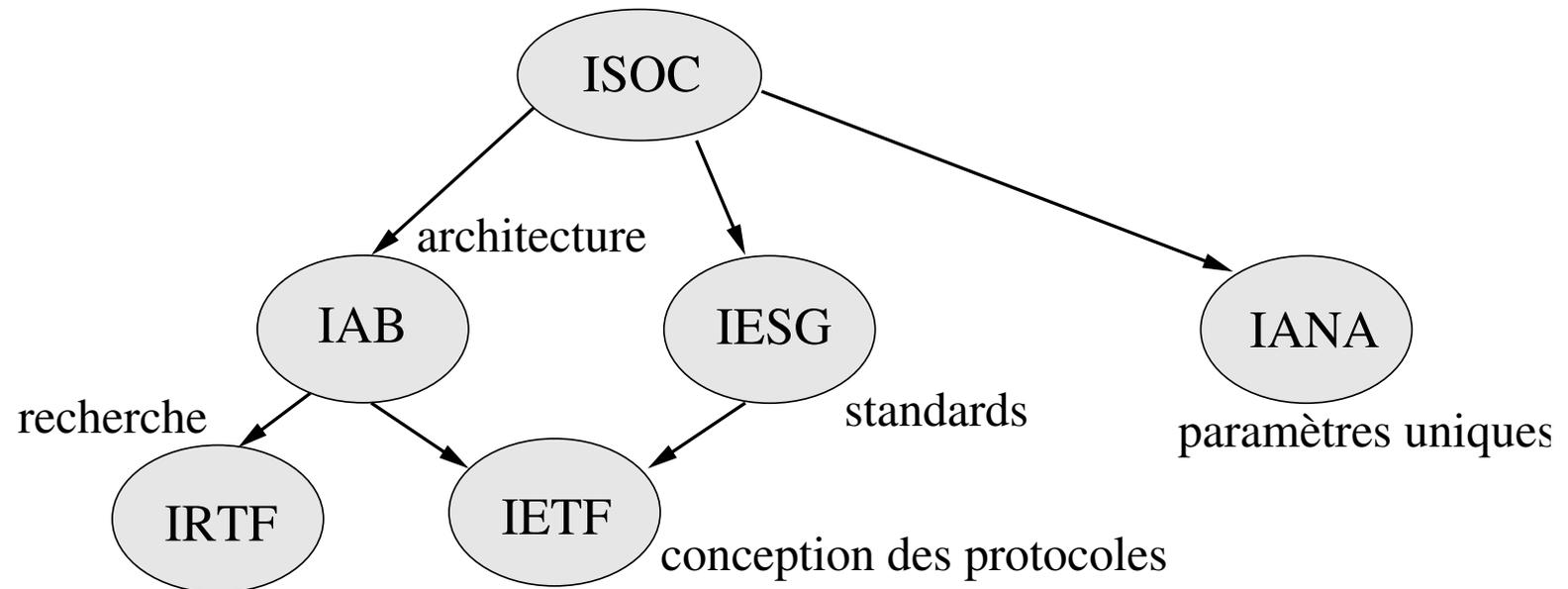
Les technologies du Web

- HTML/XHTML
- XML/XSLT/X...
- HTTP
- SSL/TLS
- CGI, PHP, Javascript, JSP, ASP ...
- Java, Perl
- Cookies
- ...



Les acteurs

Acteurs : l'IETF



- Internet Society
- Internet Assigned Numbers Authority
- Internet Architecture Board
- Internet Engineering Steering Group
- Internet Engineering Task Force
- Internet Research Task Force

- ouverte à tous les individus
- travail en *Working Groups* groupés en 8 domaines (*areas*)
- discussions par *mailing-lists*
- réunions (*IETF meetings*) 3 fois par an
- propriété intellectuelle
 - ✓ pas de confidentialité
 - ✓ indication des droits
 - ✓ si possible, “termes raisonnables et non discriminatoires”

Les documents de l'IETF

- *Internet Drafts (ID)*
- *Request For Comments (RFC)*
 - ✓ *Informational*
 - ✓ *Experimental*
 - ✓ *Historic*
 - ✓ *Standards Track (STD)*
 - ➡ *Proposed Standard*
 - ➡ *Draft Standard*
 - ➡ *Internet Standard*
 - ✓ *For Your Information (FYI)*
 - ✓ *Best Current Practice (BCP)*

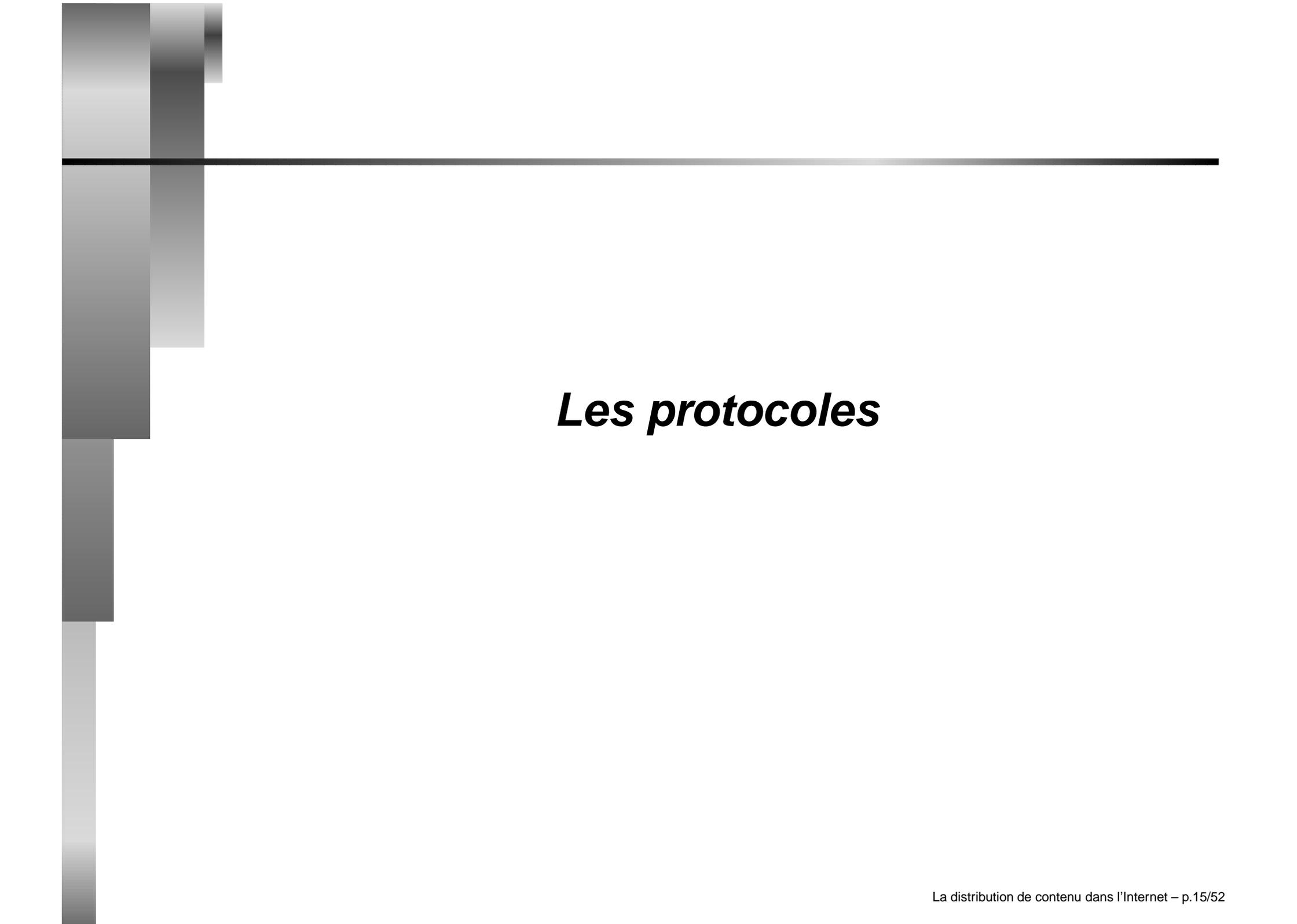
Les standards Internet

- *The Internet Standards Process* - rfc2026
- une spécification
 - ✓ stable et bien comprise
 - ✓ techniquement satisfaisante
 - ✓ plusieurs implémentations indépendantes et interopérables
 - ✓ soutien public significatif
 - ✓ utile dans (une partie de) l'Internet

“rough consensus and running code”

- *World Wide Web Consortium*
- fondé en 1994 (Tim Berners-Lee)
- *“promoting interoperability and encouraging an open forum for discussion...”*
- *“... leading the technical evolution of the web”*
- *recommandations*
 - ✓ working draft
 - ✓ last call WD
 - ✓ candidate recommandation
 - ✓ proposed recommandation
 - ✓ recommandation

- les industriels
 - ✓ prennent de l'avance
 - ☞ HTML (balises non standard)
 - ☞ cookies ...
 - ✓ compliquent la standardisation...
- les universitaires
 - ✓ cache Squid (NLNR)
 - ✓ NCSA
- la “communauté”
 - ✓ serveur Apache
 - ✓ Perl



Les protocoles

Uniform Resource Identifier – URI

- uniform
 - ✓ ≠ types dans le m^{ême} contexte
 - ✓ conventions syntaxiques
 - ✓ nouveaux types
- resource
 - ✓ “anything that has identity”
 - ✓ correspondance conceptuelle avec une entité
 - ☞ doc. électronique, image, service etc
 - ✓ pas forcément “network retrievable”
- identifier
 - ✓ objet référence à qqe chose qui a une identité
 - ✓ séquence de caractères à syntaxe contrainte

Uniform Resource Locator

`<scheme> : <scheme-specific-part>`

- `scheme` : 'a' - 'z', '0' - '9', '+', '.', '-'
- `encodage` : ASCII ou %xx
 - ✓ non ascii
 - ✓ non sur (espace ...)
 - ✓ réservé

les *scheme* Internet :

- `<scheme>://<user>:<password>@<host>:<port>/<url-path>`
- `ftp://<cwd1>/<cmd2>...<name>;type=<typecode>`
- `http://<host>:<port>/<path>?<searchpath>`
- **news : pas des "locators" !**
 - ✓ `news:<newsgroupname>`
 - ✓ `news:<message-id>`
- `nntp://<host>:<port>/<newsgroupname>/<article-number>`
- `mailto` : ne désigne pas un objet
- `gopher`, `gopher+`, `telnet`, `wais` ...

- *...distributed collaborative hypermedia information systems*
- requête/réponse
 - ✓ requête
 - ➡ méthode + URI + version protocole
 - ➡ message type MIME (mod. requête + info sur le client [+ contenu])
 - ✓ réponse
 - ➡ ligne de status
 - ➡ message type MIME (info sur le serveur + information sur l'entité (en-tête) [+ corps de l'entité])

Histoire de HTTP

- version originale 0.9, transfert brut de données
- '96 : HTTP-WG (IETF) → rfc1945 HTTP/1.0 (Informational)...
- '98 : HTTP/1.1 (rfc2068), '99 rfc2616 (176 pages)
 - ✓ 1.1 après 4 ans de débats
 - ✓ compatibilité avec 0.9 et “famille 1.0”
 - ✓ compatibilité avec prétendues implémentations 1.1
- très vaste
- complexe

OPTIONS capacités du serveur/contraintes associées à une ressource

* **GET** récupère une entité

* **HEAD** GET sans le corps de l'entité

POST associe un objet à une entité

PUT ajoute une entité

DELETE retire une entité

TRACE équivalent de `ping -R`

CONNECT établir un "tunnel"

1xx Information

2xx Succès

3xx Action nécessaire

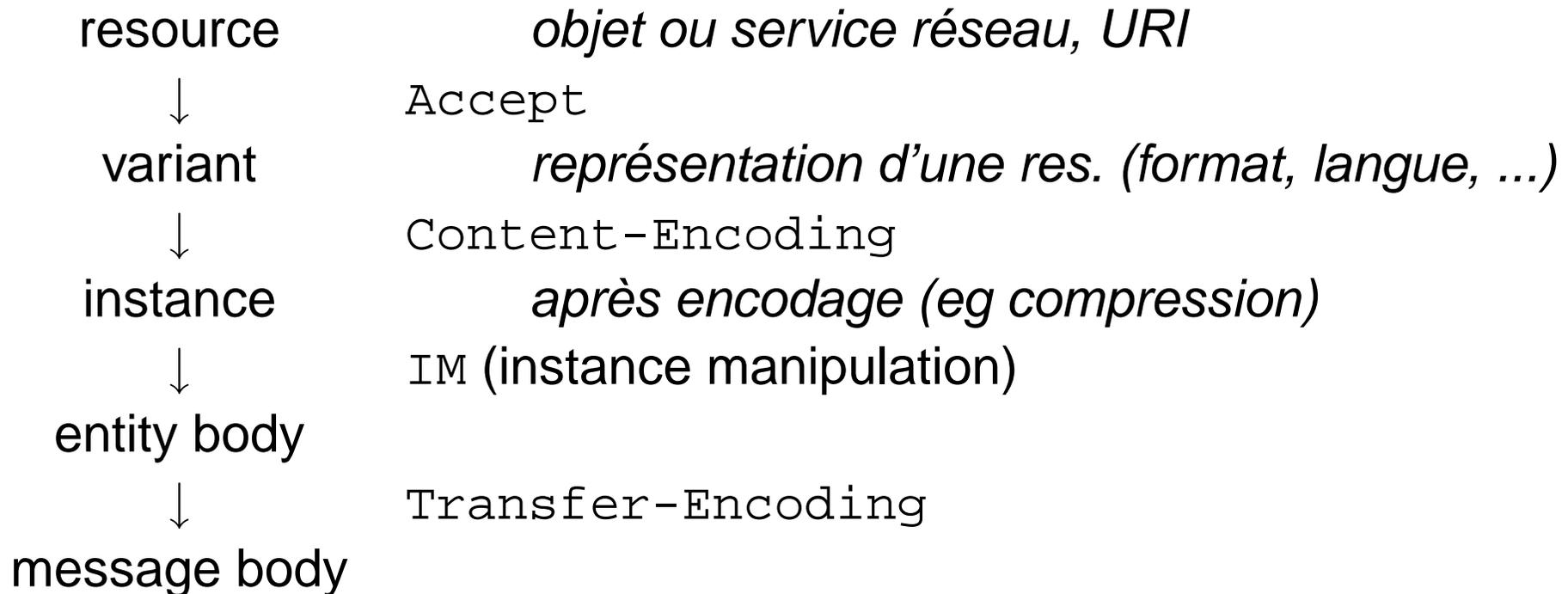
4xx Erreur client

5xx Erreur serveur

pour la partie xx

- aucune structure
- client ne comprend pas forcément
 - ✓ extensions possibles
 - ✓ alors, ne cache pas

Ressource, instance, entité etc



HTTP : exemple

⇒ GET /%7Edeleuze/ HTTP/1.0
User-Agent: Wget/1.5.3
Host: rp.lip6.fr:80
Accept: */*

⇐ HTTP/1.0 200 OK
Date: Thu, 20 Dec 2001 10:07:24 GMT
Server: Apache/1.3.19 (Unix) mod_ssl/2.8.1 OpenSSL/0.9.5a PHP/4.0.5
Last-Modified: Mon, 09 Jul 2001 11:20:30 GMT
ETag: "c11b2-520-3b49937e"
Accept-Ranges: bytes
Content-Length: 1312
Content-Type: text/html
X-Cache: MISS from firebolt.local
Connection: close

Data (1312 bytes)

HTTP : Virtual hosts

- pourquoi le champ `Host` ?
- HTTP/1.0 ...
 - ✓ `GET path` vers un serveur
 - ✓ `GET URI` vers un proxy
- web hosting : plusieurs (centaines d') IP
 - ✓ les adr. IP sont précieuses !
- ajout du champ `Host`
- la spec. me semble confuse...

HTTP : négociation de contenu

Plusieurs représentations pour le même document

- négo. menée par le serveur
 - ✓ Accept, -Charset, -Encoding, -Language
 - ✓ *quality values* (qvalue) 0.0 à 1.0
- négo. menée par l'agent
 - ✓ serveur donne liste des rep. (chacune son URI)
 - ✓ choix par l'utilisateur (ou automatique)
 - ✓ – 2 requêtes
- négo. transparente
 - ✓ cas d'un proxy

- négociation de protocole

⇐ Upgrade: HTTP/2.0 HTTP/3.0

⇒ GET http://... HTTP/1.1

Host: ...

Upgrade: HTTP/2.0

Connection: Upgrade

⇐ HTTP/1.1 101 Switching Protocols

Upgrade: HTTP/2.0

Connection: Upgrade

suite en HTTP/2.0

Authentication HTTP

- schéma général

⇐ 401 Unauthorized

WWW-Authenticate: *scheme params*

⇒ Authorization: *scheme params*

- **Authentication basique (HTTP/1.0)**

⇐ WWW-Authenticate: Basic realm="WallyWorld"

⇒ Authorization: Basic QW...h2 (user:pass en base64)

très peu sur !

Digest Access Authentication

- cryptographie faible

⇐ WWW-Authenticate: Digest nonce=*challenge*

⇒ Authorization: Digest username=... nonce=... response=*resp*

resp : digest (MD5) de user:pass:nonce:method:URI
codé en base64 (32 octets)

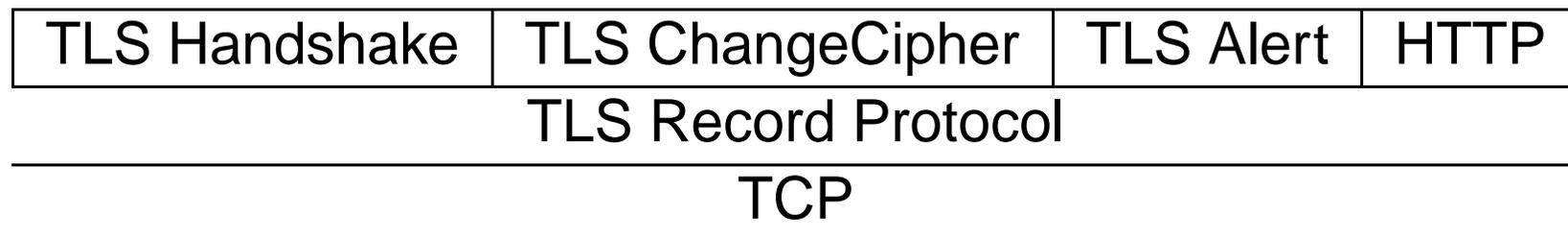
- ex. de *nonce* Base64 de
timestampH(timestamp" : "ETag" : "clef")
- optimisations...

⇐ WWW-Authenticate: Digest stale=TRUE nonce=...

⇐ Authorization-Info: nextnonce=...

Transport Layer Security : STD IETF basé sur SSL 3.0 (Netscape)

- confidentialité (crypto symétrique)
- intégrité (hash crypto)
- authentification (crypto asymétrique)
- négo. sure du secret partagé



- méthode “à la SSL”
`https://...` (port défaut 443)
 - méthode recommandée Upgrade: TLS/1.0
 - ✓ si obligatoire serveur
 - ⇐ 426 Upgrade required
 - ✓ si obligatoire client
- ⇒ `OPTIONS * HTTP/1.1`
`Upgrade: TLS/1.0 ...`

Encodage différentiel

Delta-encoding

⇒ If-None-Match: *etag*
A-IM: *delta*

⇐ 226 IM Used
IM: *delta*
ETag: ...

(caches HTTP/1.0)

- serveur peut garder vieilles instances...
- client aussi !

⇒ If-None-Match: *tag1 tag2*
⇐ IM: ...
Delta-Base: *tag*

- mais lesquelles ?

⇐ Cache-Control: *retain[=n]*

(documents expirés)

Performances : connexions persistantes

- HTTP initial
 - ✓ TCP open, req HTTP, rep HTTP, TCP close
 - ✓ performances lamentables
 - ✓ TCP open : 1 RTT
 - ✓ TCP slow-start
 - ☞ req HTTP > MSS \Rightarrow +1 RTT
 - ☞ rep HTTP > 2MSS \Rightarrow +1⁺ RTT
- connexions persistantes
 - ✓ HTTP/1.0 `Keep-Alive` mais bug si plus d'un proxy
 - ✓ HTTP/1.1 perst. par défaut
 - ✓ – de paquets, – de délais
- T/TCP (TCP for transactions) ?

Performances : pipelining

- autre cause de délai : SEND AND WAIT
- petits objets inclus \Rightarrow 1 RTT par objet
- solutions ...
 - ✓ envisagée : 1^+ objet(s) / req.
 - ✓ utilisée : connexions parallèles
 - ✓ standardisée : *(buffered) pipelining*
- pb : algo TCP Nagle
 - ✓ sol : désactiver (TCP_NO_DELAY)
- pour revalidation :
 - ✓ buff. pipelined 1.1 \Rightarrow $3\times$ moins pqt que pipelined 1.1
 - ✓ buff. pipelined 1.1 \Rightarrow $10\times$ moins pqt que 1.0
 - ✓ taille moy. pqt $\times 2$

Performances : morale

- résultats
 - ✓ 1.1 avec pipeline meilleur que 1.0 (m si conn. //)
 - ✓ nb paquets $\div 2$
 - ✓ délai significatif
- surtout pour revalidation
- morale
 - ✓ prévision de l'évolution des besoins
 - ✓ limites de l'abstraction en couches ?

- HTTP est **sans état**

 - ✓ session : remplissage de panier...

- encodage dans l'URL

 - `http://shop.example.com/acheter?panier=Lessive`

- Cookies (IETF '00, Netscape)

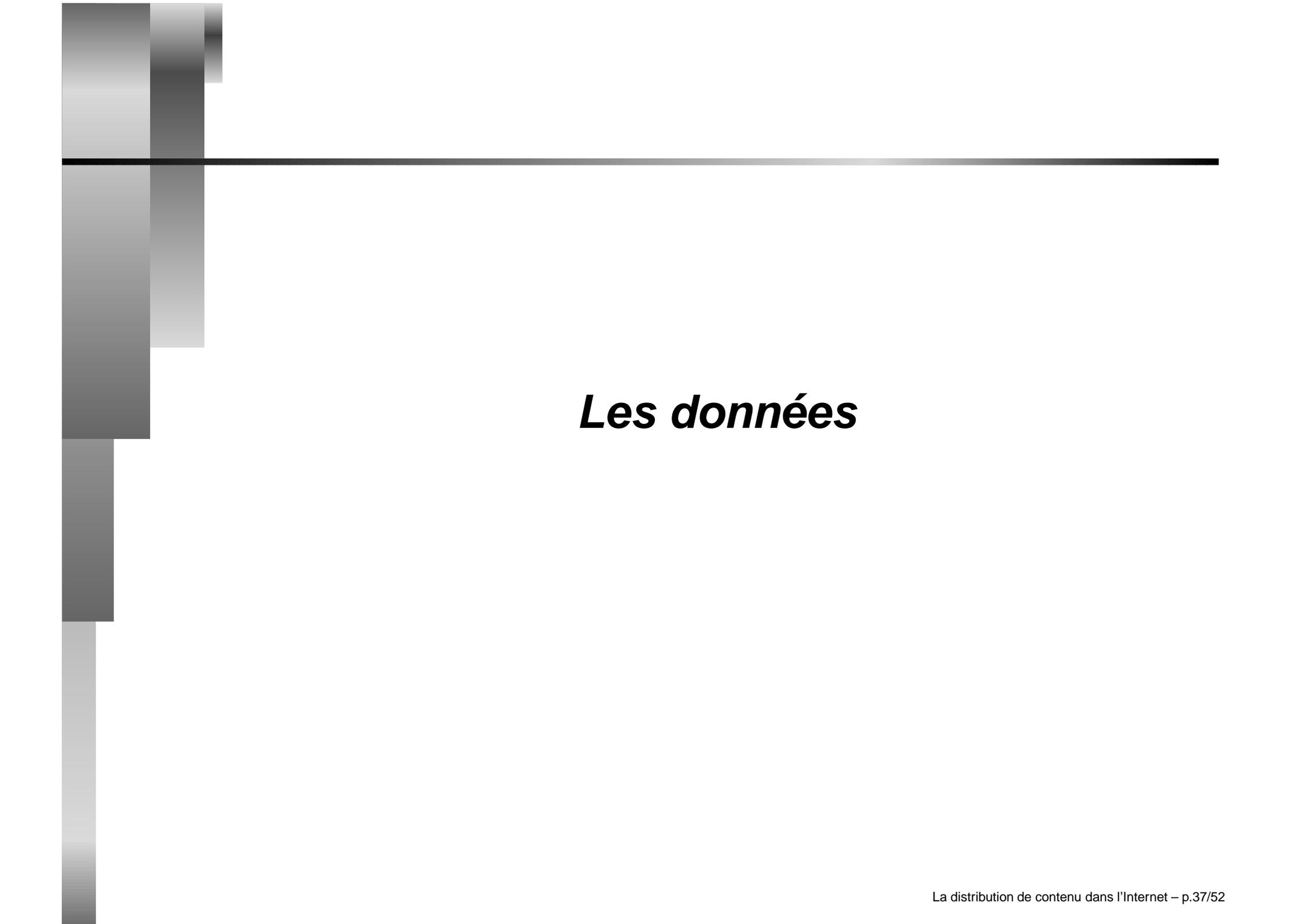
 - ⇐ `Set-Cookie2: panier=Lessive TTL, portée...`

 - ⇒ `POST /acheter HTTP/1.1`

 - `Host: shop.example.com`

 - `Cookie: panier=Lessive ...`

- en général, plutôt une clé de BD dans le serveur



Les données

1.0 web confidentiel. Très limité

2.0 '96 rfc1866 IETF HTML-WG (fermé en '98)

- *active maps*, formulaires

x.x *Netscape extension tags*

3.0 trop complexe, pas implémentée ➡ abandonnée

3.2 '96 W3C, bcp plus simple que 3.0

- tableaux
- applets
- texte autour des images
- indices/exposants
- compatible 2.0

4.0 '97 corrigée '98 – reprend les idées de 3.0

4.01 déc '99

XHTML 1.0 '00 reformulation de HTML 4 en XML 1.0

- strict
- transitional
- frameset

XHTML Basic 12/00

Modularization of XHTML 04/01

XHTML 1.1 05/01

MathML 2.0 10/03

...

ISO HTML 05/00 sous-ensemble de HTML 4

drafts

- Modularization of XHTML in XML Schema
- XHTML + MathML + SVG profile
- XHTML 2.0
- XFrames

related work

- XML
- feuilles de styles (CSS en HTML, XSL pour XML)
- DOM (Document Object Model)
- MathML
- ...

Javascript

- créé par Netscape
- langage de script compact et orienté objet
- version serveur (*Livewire Javascript*)
- script inséré dans une balise <SCRIPT>
- ...

Script côté serveur : CGI

Common Gateway Interface

- une URL contenant `cgi-bin` représente un script à exécuter
- l'objet de la requête est passé sur `stdin`
- variables d'environnement
- script génère une page HTML sur `stdout`

```
POST www.cgiexample.com/cgi-bin/annuaire
```

```
name=Martin&firstname=Laurent
```

ou

```
GET www.cgiexample.com/cgi-bin/annuaire?name=Martin&firstname=La
```

Server Side Includes

- '95 NCSA
- directives SSI dans commentaires HTML
 - ✓ `include`
 - ✓ `echo (variable)`
 - ✓ `fsize, flastmod`
 - ✓ `exec` sur `/bin/sh` ou `CGI`
- documents de type `text/x-server-parsed-html`
 - ✓ extension par défaut `.shtml`

PHP: Hypertext Preprocessor

- projet de la *Apache Software Foundation*
- multi-OS/multiserveur
- paradigme procédural ou objet
- interface avec bases de données
- génère HTML, images, PDF, Flash ...
- également
 - ✓ command line scripting
 - ✓ clients graphiques (PHP-GTK)

- créé en '96
- interpréteur Perl intégré au serveur web Apache
- 100 × plus rapide que CGI
- bcp plus flexible
- ...
- aussi `mod_caml`

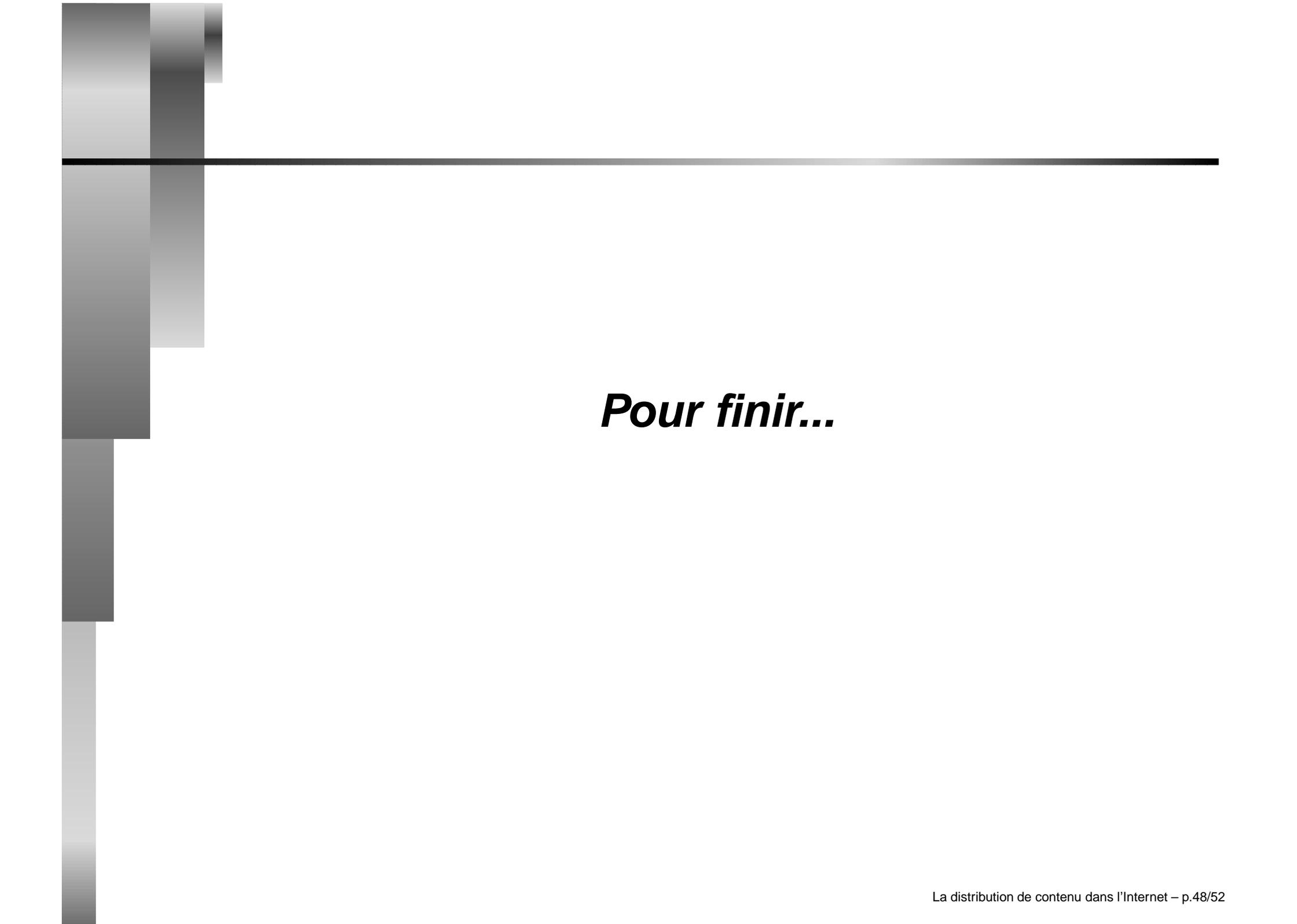
introduit en '95 (Java2 '98)

- indépendant du système (JVM)
- *applet*
- *servlet*
 - ✓ JDBC API, HTTP, ...
- *scriptlet (JavaServer Pages – JSP)*

Practical Extraction and Report Language '86

- admin. système tout OS (Unix en condensé)
- web dynamique (CGI puis mod_perl)
- Comprehensive Perl Archive Network (CPAN)
- “*write only language*”

machine virtuelle unifiée : *parrot*



Pour finir...

Le web comme unification

- passerelles (proxies) vers autres protocoles/services
 - interface vers base de données
 - *browser* logiciel de communication unifié
 - applications réparties
 - ✓ XML comme format de données
 - ✓ HTTP comme protocole de transport
- ▣► “services web”

Services Web : SOAP

services web : utiliser le web pour la communication entre applications distantes

RPC (Remote Procedure Call) : Unix, C

DCOM (Distributed Component Object Model) : Windows, *

RMI (Remote Method Invocation – Java) : *, Java

CORBA *, *

SOAP (Simple Object Access Protocol, prev. XML-RPC) : *, *

1. encodage objet/méthode en XML
2. transfert sur HTTP (ou autres)
3. décodage
4. exécution
5. encodage XML du résultat ...

- SOAP : est-ce bien raisonnable ?
- “everything over HTTP”
 - ✓ *stream* sur HTTP ! (“ça passe les firewalls”)
- interfaces de configuration non automatisables
- ...
- HTCPCP (rfc2324)

Le problème de la distribution

- acheminer de 1 vers beaucoup
- croissance exponentielle
- scalabilité
 - ✓ multicast
 - ✓ cache
 - ☞ client
 - ☞ proxy