

Serveis web i de transferència de fitxers

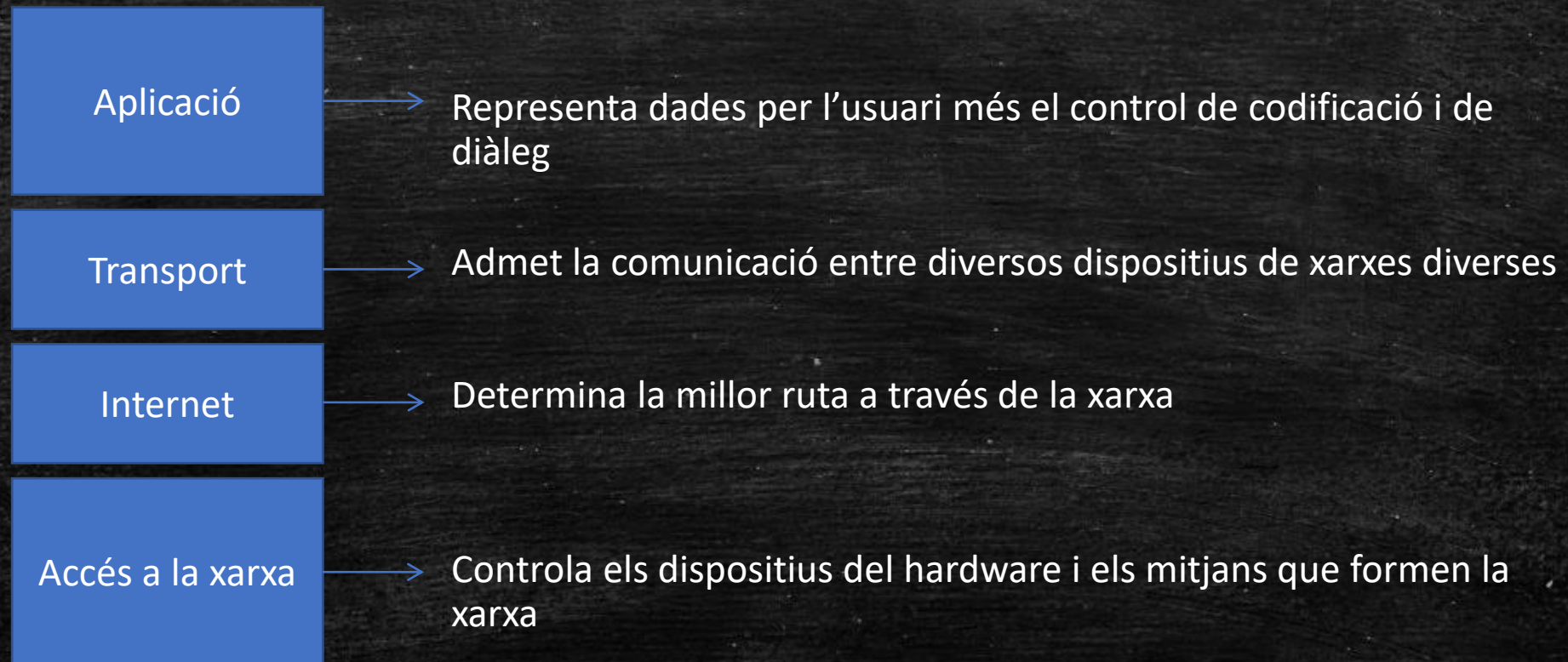
Instal·lació i administració de servidors web

Protocol HTTP

Protocol de Transferència d'Hipertext

- Protocol de capa d'aplicació
- Proporciona transferència de documents d'hipertext a la web
- Utilitza un mecanisme client/servidor al port 80 basat en TCP

Model TCP/IP

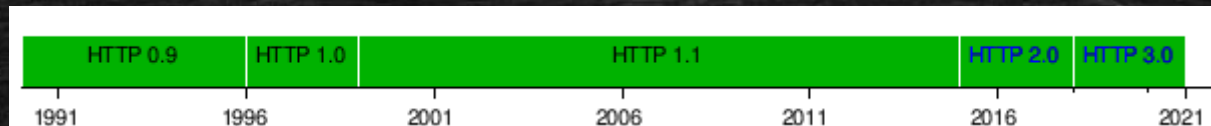


Protocol HTTP

- Mecanisme client/servidor al port 80 basat en TCP
- La connexió s'inicia i es tanca per cada petició/resposta
- Protocol sense estat i no orientat a la connexió permanent

Història del protocol HTTP

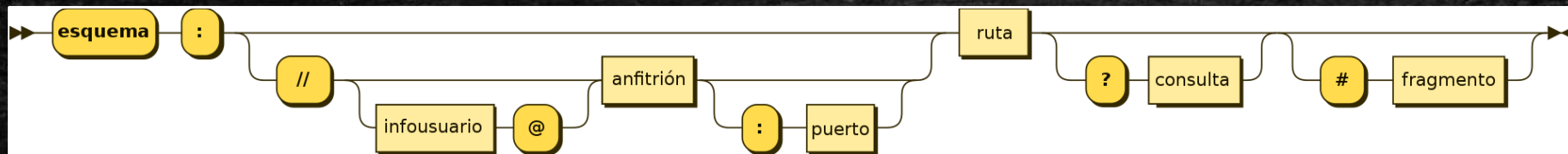
- Apareix als anys 90 per transferir documents d'hipermèdia
- Primer transportava text, actualment contingut multimèdia.



- Utilitzarem els navegadors (Firefox, Chrome...) per obtenir contingut HTTP
- També existeixen navegadors per contingut únicament de text!

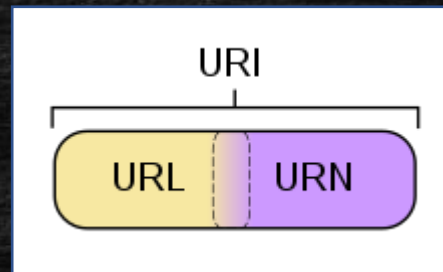
Com accedir als documents del web

- Cal un mecanisme universal: Uniform Resource Identifier (URI)
- Consta de: *schema:identifíer*
- *Esquema* descriu la sintaxi utilitzada per l'identificador (HTTP, FTP...)
- *Identifíer* permet determinar el recurs concret dins l'esquema



Com accedir als documents del web

- URI permet identificar elements globalment
- URL localitzar-los
- URN proporciona un mecanisme d'assignació de noms únics



Funcionament del servei web

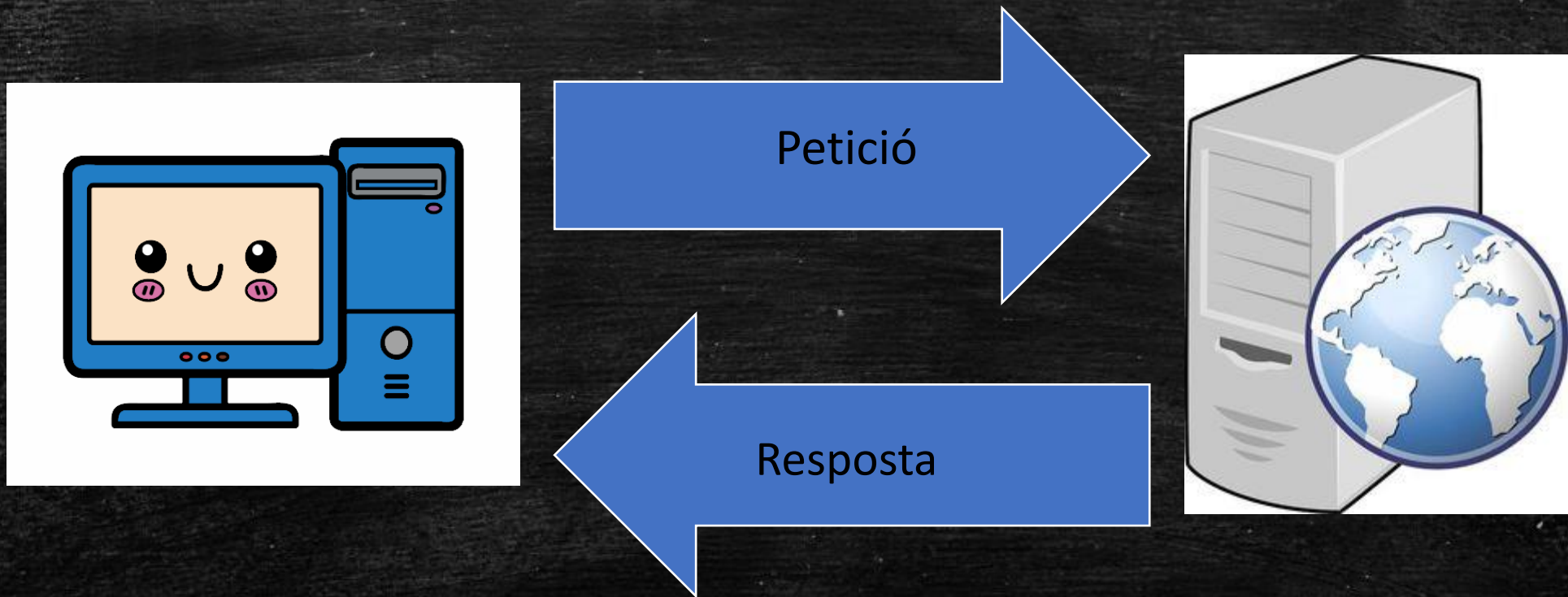
- HTTP estructura el diàleg client/servidor en un esquema de petició/resposta.
- HTTP 1.0 → la connexió s'obre i es tanca per cada petició
- HTTP 1.1 → la connexió es manté oberta un temps per realitzar més peticions
- HTTP 2.0 → incorpora millores per reduir la latència en càrrega de pàgines web
- HTTP 3.0 → encara no està activat per defecte en cap navegador

Connexions persistents

Permeten que els múltiples elements d'una pàgina web es puguin baixar sense que calgui una connexió per cada element

HTTP és un protocol sense estat. Ni client ni servidor mantenen un estat de la sessió a nivell de protocol

Descripció del diàleg petició/resposta



Connectar al servidor

- Hem d'establir una connexió amb el servidor web
- Per Telnet → IP:PORT

Petició (HTTP Request)

El missatge de petició consta de:

- Línia de petició
- Capçaleres
- Línia en blanc
- Cos de la petició

Petició → Línia de petició

mètode de la petició + document a obtenir + versió protocol HTTP + ¶

Exemple: GET /index.html HTTP/1.0

Petició → Mètodes

- HEAD: Ens permet verificar l'existència del document.
- GET: Permet obtenir el document sol·licitat.
- POST: Permet enviar informació al servidor per incorporar-la al document. Exemple: formularis, log-in...
- PUT: Permet pujar un document al servidor.
- DELETE: Permet eliminar un document del servidor (si es tenen permisos).
- TRACE: El servidor ens retorna una còpia del missatge que li ha arribat.
- OPTIONS: Sol·licitud d'informació de les opcions de transferència del servidor. Aquest contesta indicant les seves capacitats.

Petició → Capçaleres

- Permeten descriure opcions preferibles del servidor i opcions del client. El servidor ho tindrà en compte en la resposta
- Són OPCIONALS!!
- Exemple: el client pot indicar el seu sistema operatiu i el navegador
- En HTTP/1.1 (i superiors) hi ha una capçalera obligatòria: HOST
- HOST indica la URL del servidor al qual s'adreça la petició

Petició → CRLF + cos

- CRLF: salt de línia, enter
- COS: és opcional, tot i que habitualment no s'utilitza per les peticions

Esquema d'una petició

```
root@server:~# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /index.html HTTP/1.0
```


Resposta

- Els missatges tenen una estructura similar a les peticions:
 - Línia d'estatus
 - Capçaleres
 - Línia en blanc
 - Cos de la resposta

Resposta → línia d'estatus

Protocol HTTP usat + valor numèric amb el tipus de resposta + text descriptiu de l'estatus

- Exemple: HTTP/1.1 200 OK

Resposta → estatus de les respostes

- Segons el primer dígit del valor numèric d'estatus es pot sap que:
 - 1xx: informació genèrica
 - 2xx: acció amb èxit
 - 3xx: redirecció
 - 4xx: error del client
 - 5xx: error del servidor

Resposta → Capçaleres

- La resposta conté totes les capçaleres que el servidor cregui oportunes
- Exemples:
 - Tipus de servidor que utilitzem
 - Última modificació a l'arxiu
 - Data

Resposta → CRLF + Cos

- CRLF: salt de línia, línia en blanc, enter
- Cos: conté el contingut “real” de la resposta.

Resposta → Esquema d'una resposta

- Resposta per OPTIONS:

```
HTTP/1.1 204 No Content
Allow: OPTIONS, GET, HEAD, POST
Cache-Control: max-age=604800
Date: Thu, 13 Oct 2016 11:45:00
GMT
Server: EOS (lax004/2813)
```

- Resposta per GET:

```
HTTP/1.1 200 OK
Date: Mon, 14 Sep 2020 07:59:09 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Mon, 07 Sep 2020 20:01:44 GMT
ETag: "a8-5aeb015df61"
Accept-Ranges: bytes
Content-Length: 168
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<html>
    <head>
        <title>Pàgina principal</title>
    </head>
    <body>
        <h1>Pàgina principal</h1>
        <p>Servidor Apache</p>
    </body>
</html>
Connection closed by foreign host.
```


Exemple petició/resposta

```
root@server:~# telnet localhost 80
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
GET /index.html HTTP/1.0

HTTP/1.1 200 OK
Date: Mon, 14 Sep 2020 07:59:09 GMT
Server: Apache/2.4.38 (Debian)
Last-Modified: Mon, 07 Sep 2020 20:01:44 GMT
ETag: "a8-5aeb015df61"
Accept-Ranges: bytes
Content-Length: 168
Vary: Accept-Encoding
Connection: close
Content-Type: text/html

<html>
    <head>
        <title>Pàgina principal</title>
    </head>
    <body>
        <h1>Pàgina principal</h1>
        <p>Servidor Apache</p>
    </body>
</html>
Connection closed by foreign host.
root@server:~#
```