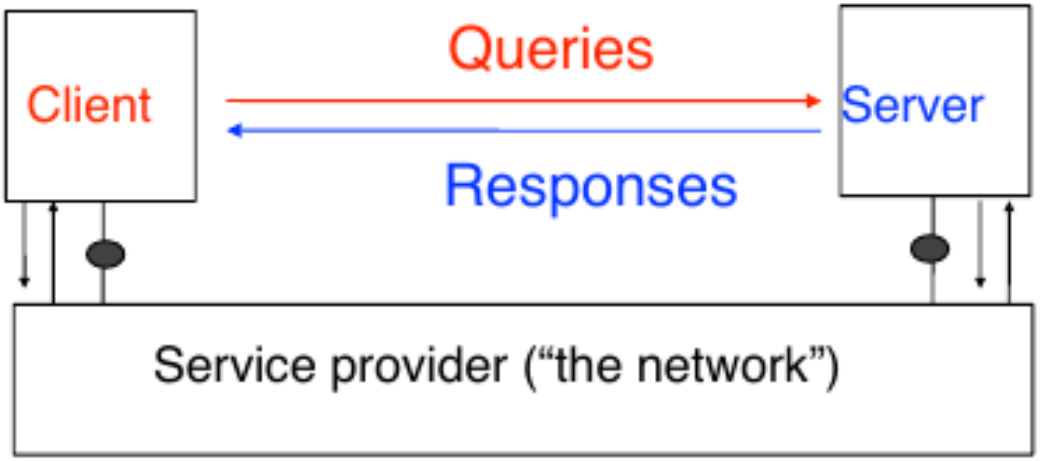
Internet, Principes et Protocoles (IPP)

Application Layer

- Highest layer, allowing hosts to exchange different kind of information (sending emails, ssh, serving websites, REST APIs,...)
- Relying on the lower protocols for the transmit ion and management of the information.
- Often using on the client-server model

Client-Server Model



Client
- interacts with server through transport layer
- sends queries or commands

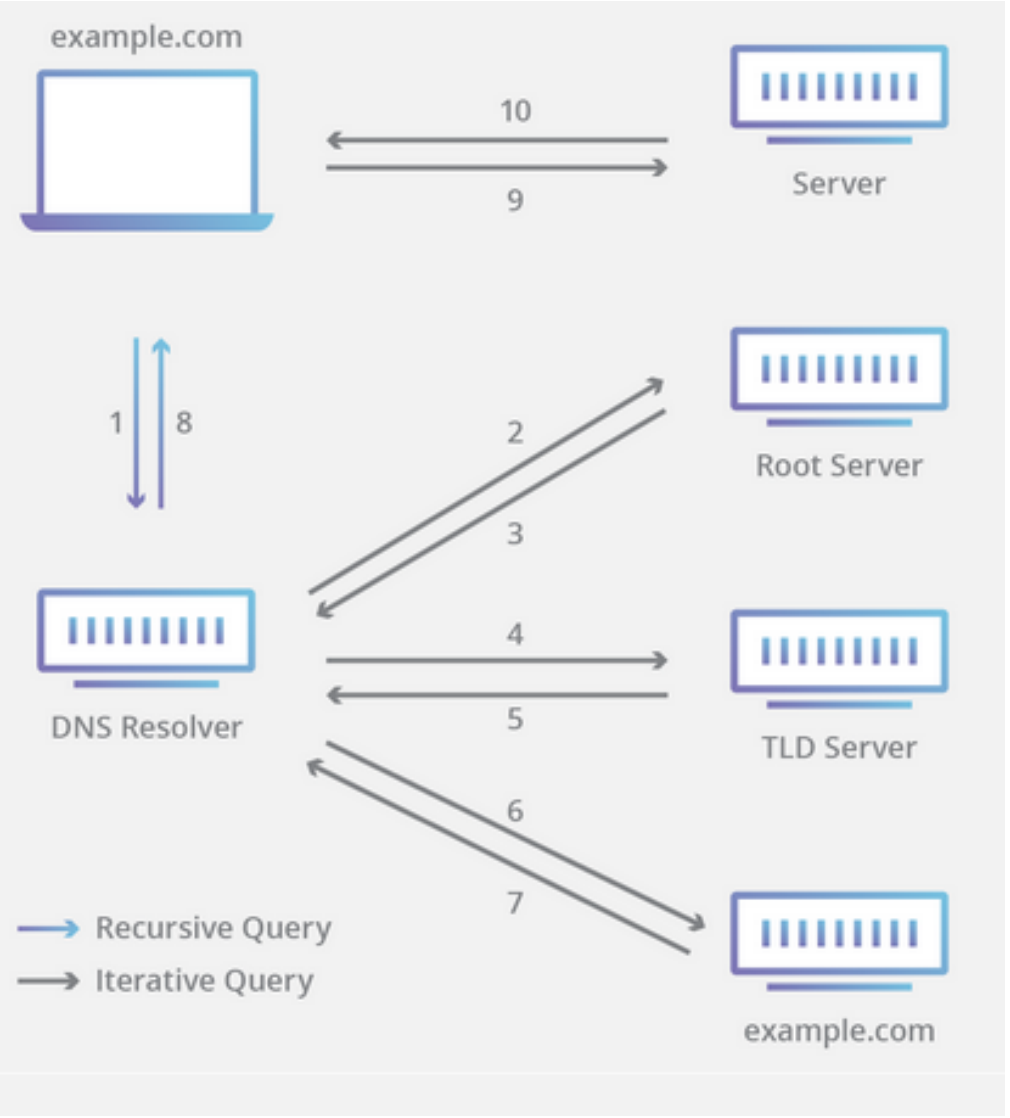
Server
- Answers the queries received from clients
- Executes the commands from clients
- Many clients can use the same server

Example : email, www, ...

DNS In Practice

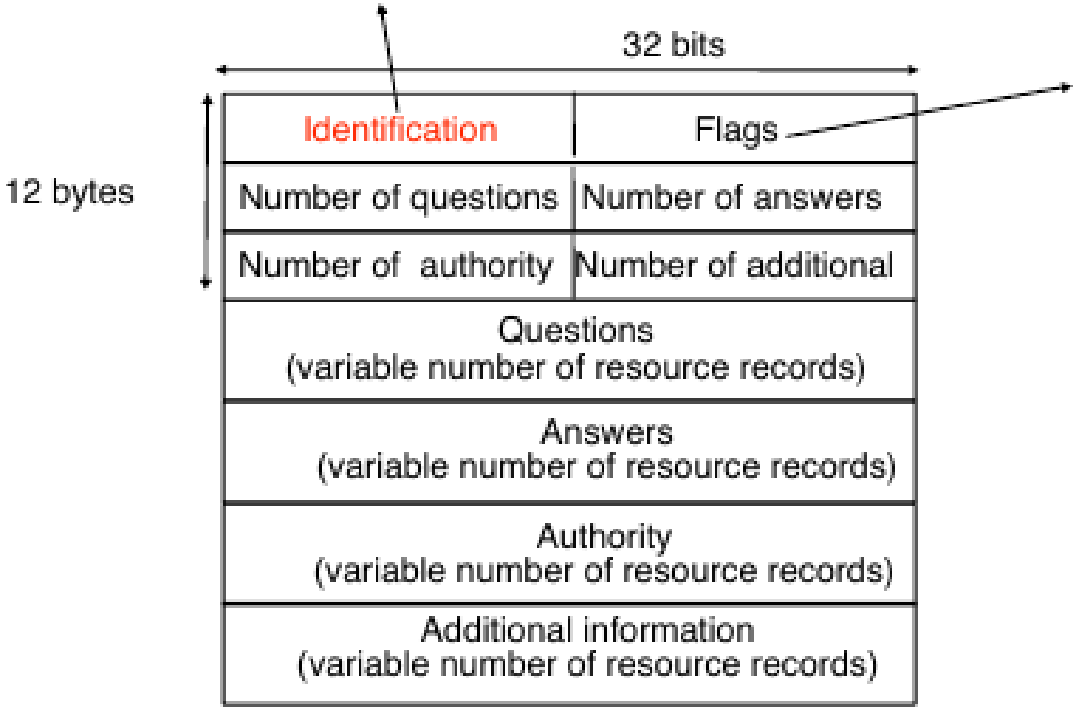
The 8 steps in an iterative DNS lookup:

- A user types ‘example.com’ into a web browser which sends the query to a DNS recursive resolver.
- The resolver then queries a DNS root nameserver.
- The root server then responds to the resolver with the address of a TLD DNS server (such as .com or .net), which stores the information for its domains.
- The resolver then makes a request to the .com TLD.
- The TLD server then responds with the IP address of the domain’s nameserver, example.com.
- The recursive resolver sends a query to the domain’s nameserver.
- The domain nameserver answers with the IP address for example.com
- The DNS resolver then responds to the web browser with the IP address



DNS Packet

Each DNS request contains a number that will be returned in the response by the server to allow the client to match the request.

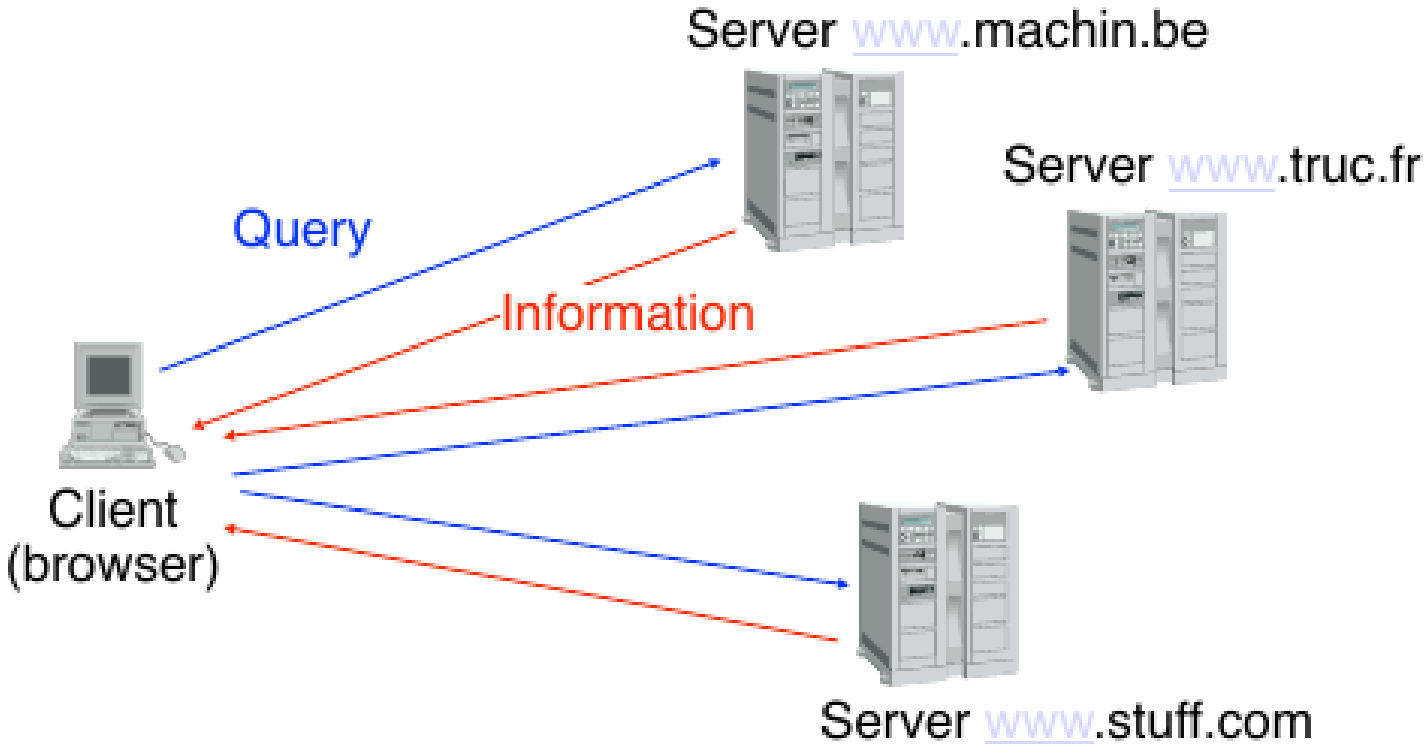


Question/Response
Recursive question or not
Authoritative answer or not
Possible error

World Wide Web

Goals

Allow browsers to browse hypertext documents stored on multiple servers



World Wide Web

The five key elements of www

1. An addressing scheme that allows to identify any document stored on a server
   URL : Uniform Resource Locator
2. An hypertext language that allows to easily write documents with hypertext links
   HTML : HyperText Markup Language
3. An efficient and lightweight application-level protocol to exchange documents
   HTTP : HyperText Transfer Protocol
4. Servers
5. Clients (browsers)

World Wide Web - URL

Uniform Resource Locator (URL)

generic syntax : <protocol>://<document>

- protocol used to retrieve document from server
  - http is the most common one but others are frequently used
- document indicates the server and the location of the document

<user>:<password>@<server>:*<port>*/<path>

- <user> : optional username
- <password> : optional password
- <machine> : hostname or IP address of the server that hosts the document
- <port> : optional port number
- <path> : document location on server

examples

- http://www.info.ucl.ac.be
- http://alice:secret@inl.info.ucl.ac.be:80/index.html

World Wide Web - HTML

HyperText Markup Langage

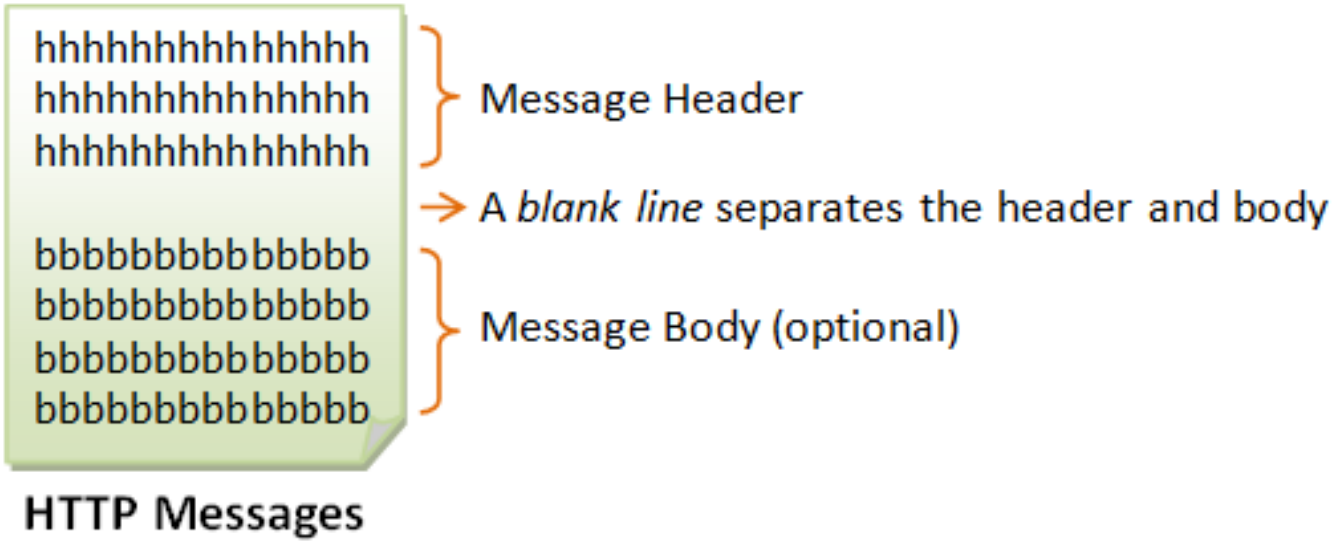
Language used to encode documents on the web

Keywords

```
<HTML>...</HTML>
<HEAD>...</HEAD>
<BODY>...</BODY>
<TITLE>..</TITLE>
<B>...</B>
<I>...</I>
<H1>...</H1>
<P>
<HR>
<UL>...</UL>
<OL>...</OL>
<IMG SRC="URL">
<A HREF="URL">text anchor</A>
```

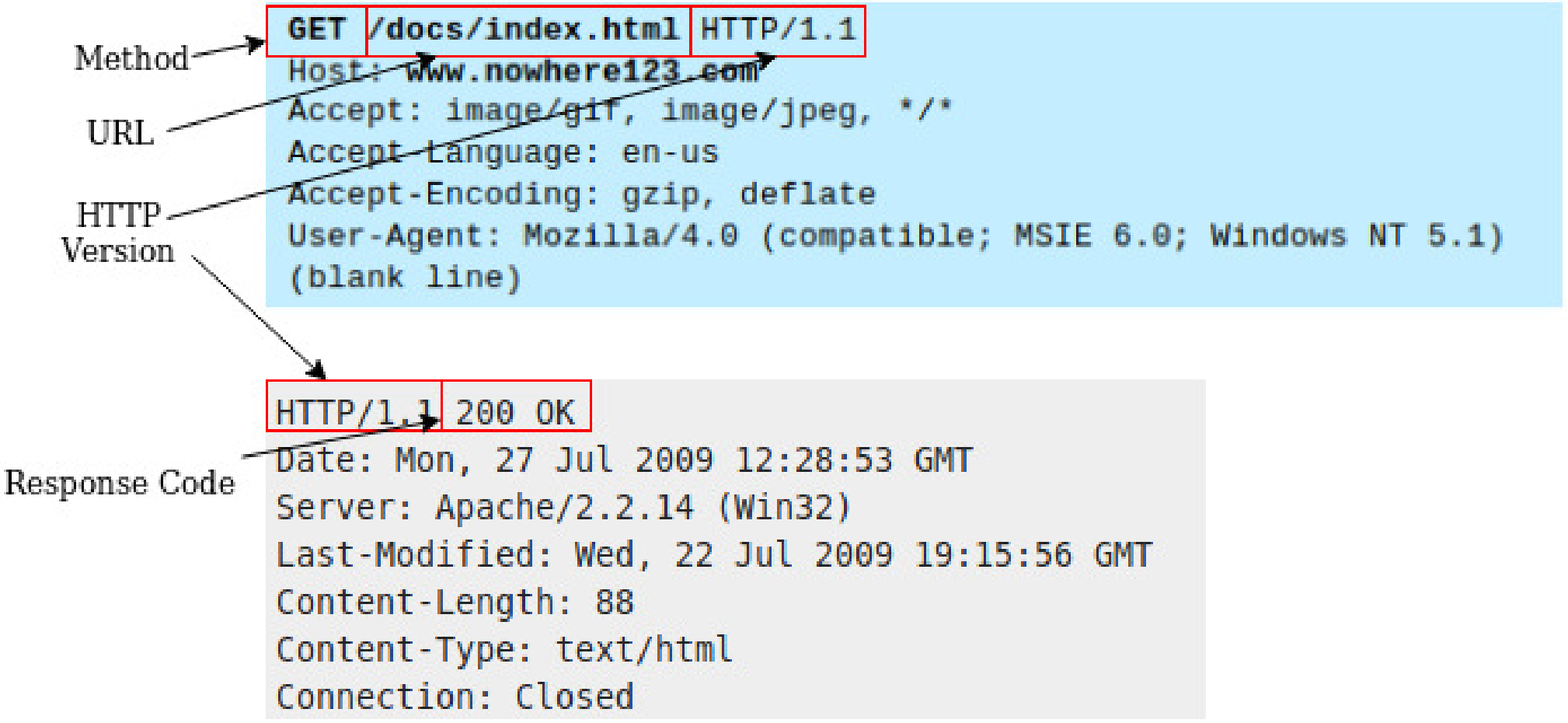
World Wide Web – HTTP

- HyperText Transfer Protocol
- Relies on TCP (default port: 80)
- Client sends request, server sends response
- APIs are usually using this protocol



HTTP

HTTP request and response example



HTTP Methods

- GET: A client can use the GET request to get a web resource from the server.
- HEAD: A client can use the HEAD request to get the header that a GET request would have obtained.
- POST: Used to post data up to the web server.
- PUT: Ask the server to store the data.
- DELETE: Ask the server to delete the data.
- TRACE: Ask the server to return a diagnostic trace of the actions it takes.
- OPTIONS: Ask the server to return the list of request methods it supports.
- ...

POST vs GET

- GET: *GET /test/demo_form.php?name1=value1&name2=value2*
  - *Can be bookmarked*
  - *remain in browser history*
  - *Can be cached*
  - *Content length limited*
    - → *Should not be used for sensitive data*

- POST:
  - *Are not cached*
  - *do not remain in browser history*
  - *Content length unlimited*

```
POST /test HTTP/1.1
Host: foo.example
Content-Type: application/x-www-form-urle
Content-Length: 27

field1=value1&field2=value2
```

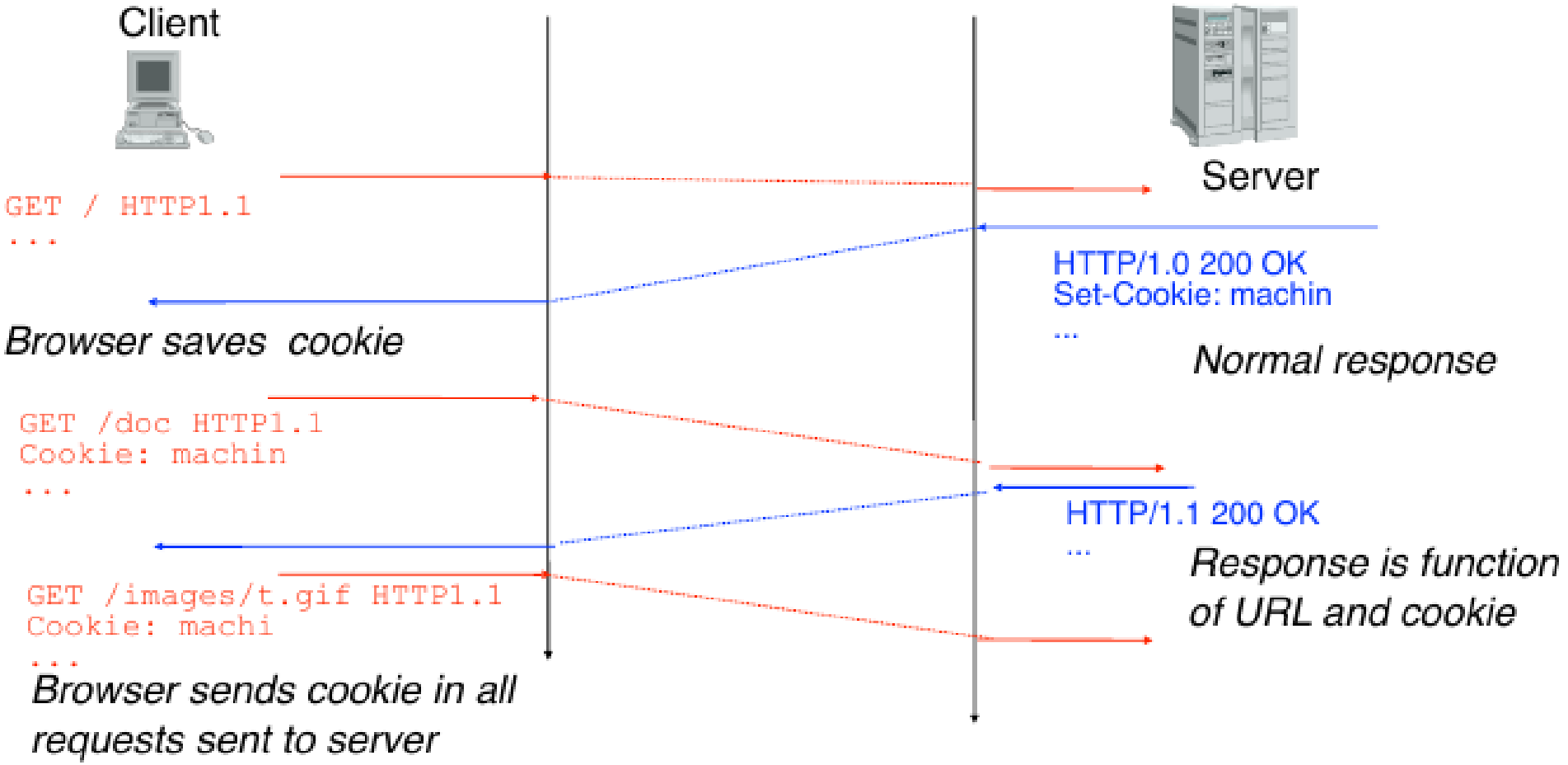
HTTP Response Codes

HTTP code	Meaning
200	OK
4xx	Bad request (client's fault)
5xx	Failed request (server's fault)
401	Unauthorized request
404	Resource not found
500	Internal error (bug)
503	Server overloaded

HTTP Misc headers

- Authorization: performs access control
- Referer: Indicates the URL visited by the clien before this request
- User-Agent: indicates the program used by the client
- Cookies (or super-cookies)

Cookies


Client
Server
GET / HTTP1.1
...
HTTP/1.0 200 OK
Set-Cookie: machin
...
Browser saves cookie
Normal response
GET /doc HTTP1.1
Cookie: machin
...
HTTP/1.1 200 OK
...
Response is function of URL and cookie
GET /images/t.gif HTTP1.1
Cookie: machi
...
Browser sends cookie in all requests sent to server

Cookies

- Set-Coockie: creates the coockie
- Expires and Max-Age: determine the life of the coockie. If not set, deleted when the client is closed
- Http-Only: the coockie will not be accessible by JS code
- Same-Site: if set to Strict, the client will only send the coockie to the server that created it. If set to Lax, the coockies will also be sent to subrequests (in order to load scripts, images,..)

Others

- Cross-Site Scripting

```
1 | <img src="https://bank.example.com/withdraw?account=bob&amount=1000000&for=mallory">
```

Now, if you are logged into your bank account and your cookies are still valid (and there is no other validation), you will transfer money as soon as you load the HTML that contains this image.

- HTTP 1.1 a single persistent TCP connection for several requests and responses
Coockies and security

REST APIs

- HTTP Methods
  - GET – Retrieve
  - POST – Create
  - DELETE – delete
  - PUT – Update
- Use plurals in the URLs
  - /v1/t-shirts (the set of T-shirts)
  - /v1/t-shirts/007 (the element of ID 007 in the T-shirts set)
- But do use non-incremental IDs (crf. Security class)
- Implement rate-limiting and SSL/TLS
- Provide documentation

World Wide Web

- If all goes on HTTP, anyone on the route or on the network could see the traffic.
  - Not goo to connect to a bank, voting, or other sensitive content

    → *Demo and Demo of HTTP requests*
- We need to make sure only the intended recipients can see the message, and that we are communicating with the right server

  → SSL/TLS protocol

World Wide Web

- SSL/TLS are cryptographic protocols designed to provide security on the network.
- It sits between the application layer and the transport layer. This means that application layer protocol can use it, but don’t have to. Yet, it is not a transport layer protocol.
- SSL = Secure Socket Layer protocol
- TLS = Transport Layer Security protocol

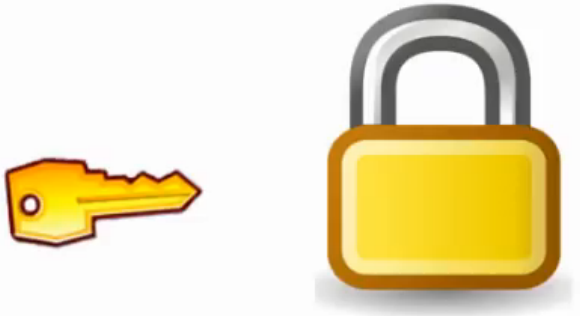
SSL / TLS

- SSL is an old protocol that has been replaced by TLS. The current TLS version is 1.3
  - It has been replaced mainly because of cryptographic enhancements.
  - SSL is INSECURE, TLS security depends on version
- Based on
  - Certificates
  - Certificate Authorities
  - Diffie-Hellman Algorythm

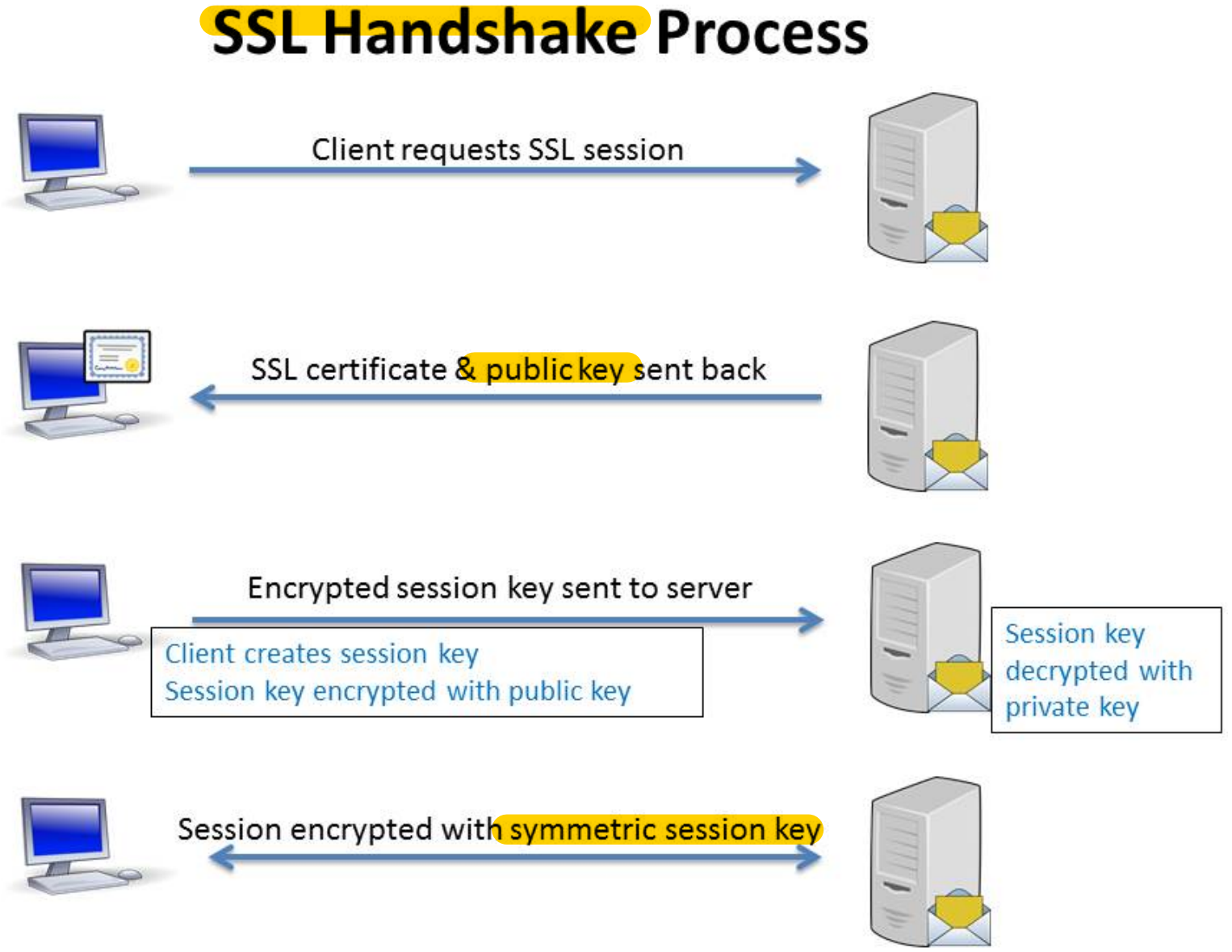
TLS

SSL/TLS encryption & authentication process

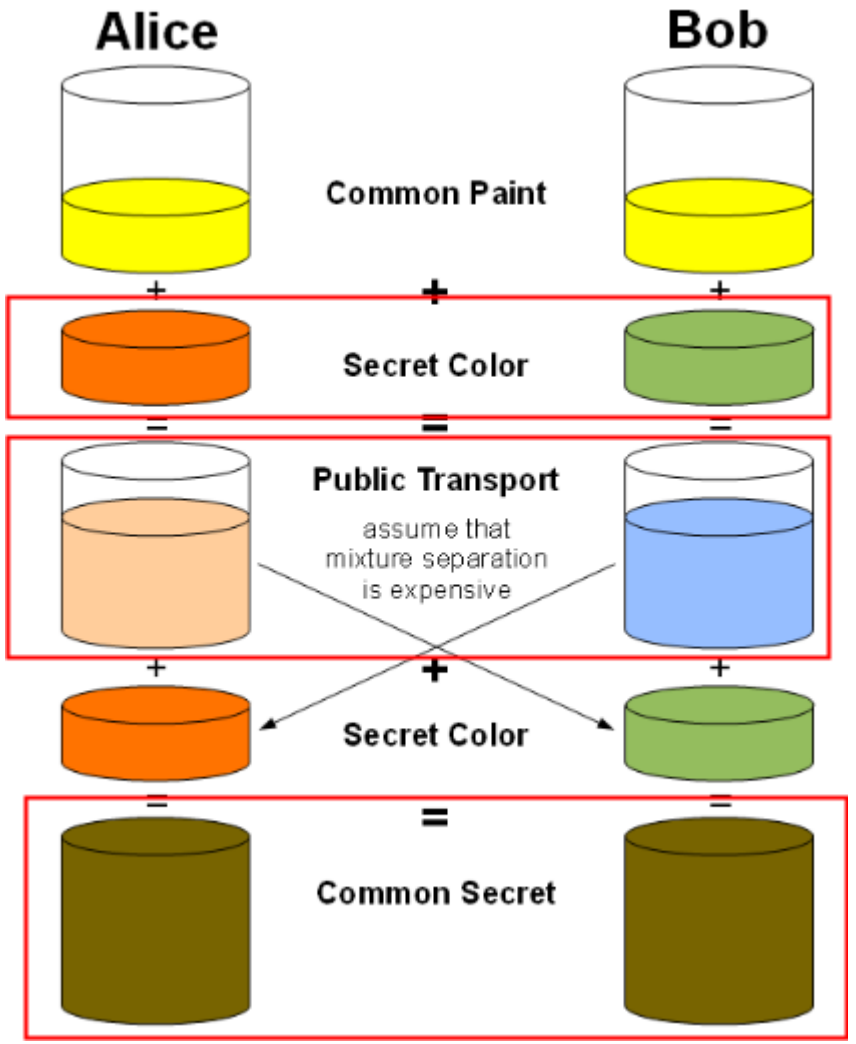
How does https works under the hood



TLS


SSL Handshake Process
Client requests SSL session
SSL certificate & public key sent back
Encrypted session key sent to server
Client creates session key
Session key encrypted with public key
Session key decrypted with private key
Session encrypted with symmetric session key

Diffie - Hellman

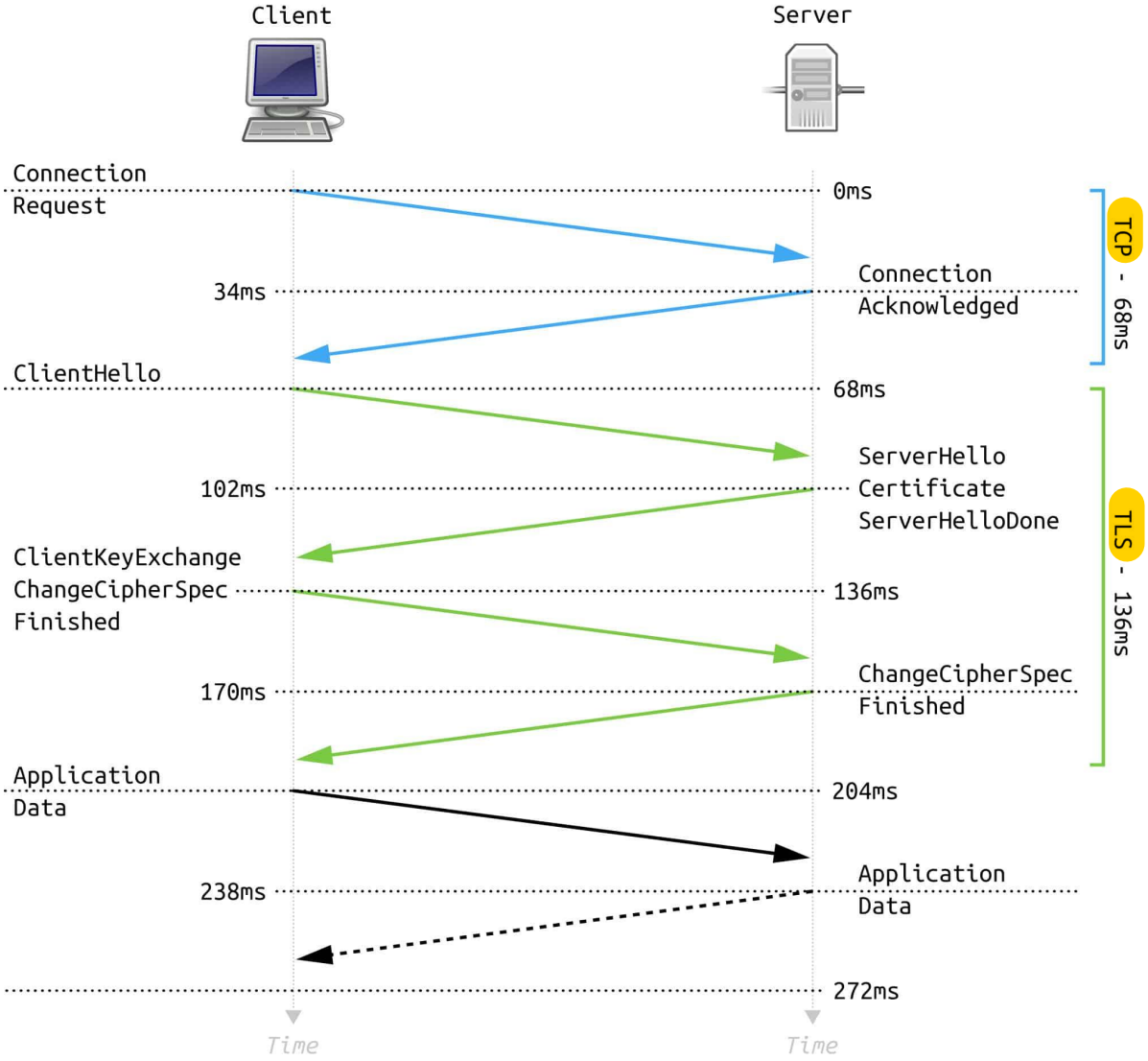


All Together

1. TCP Handshake
(on this image, simplyfied)

2. TLS Key Establishment

3. Application layer data exchanged



Question

- What happens when you type https://vinci.be/index.html in your browser? (uniquement la couche application)

Question

World Wide Web