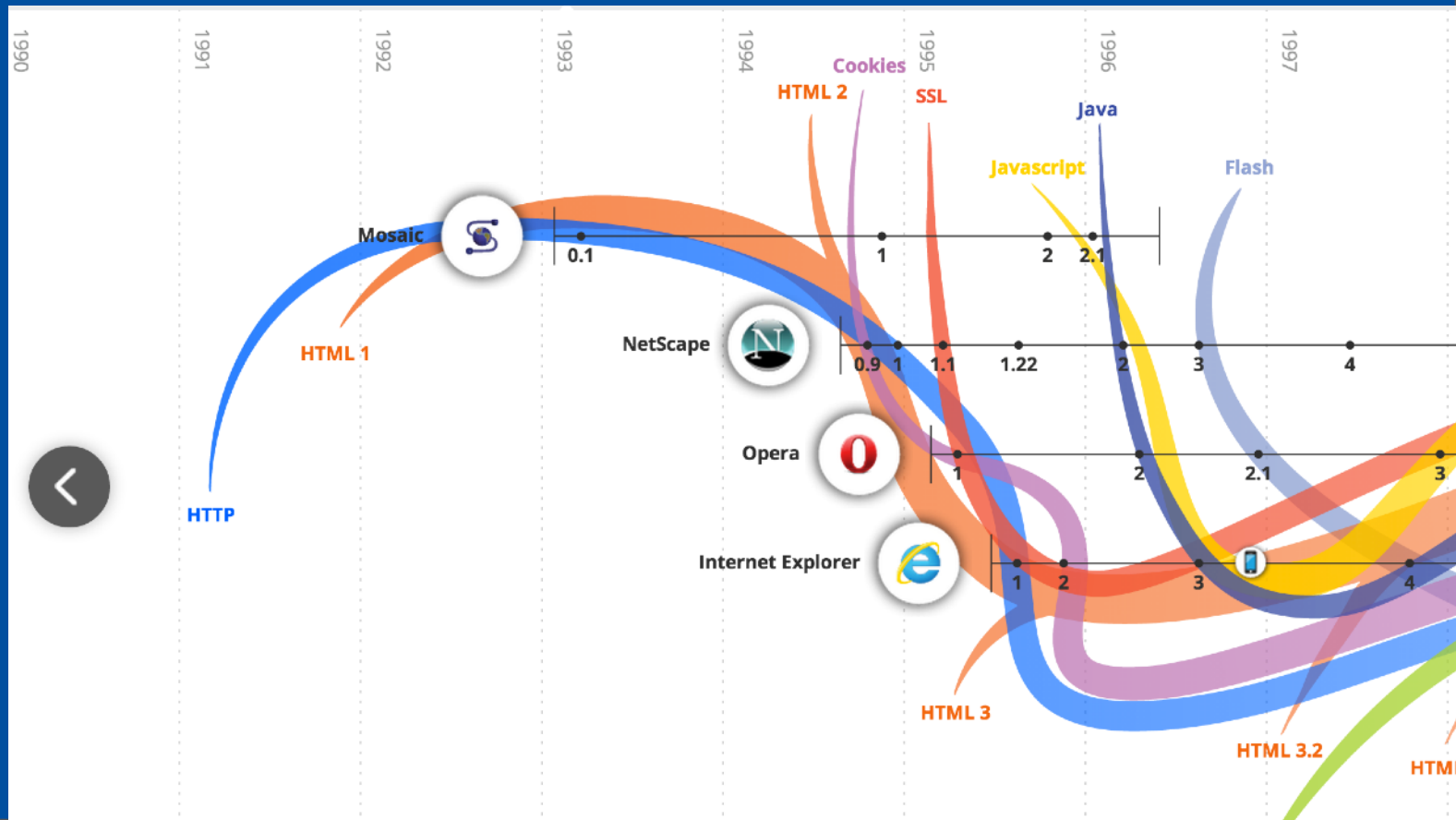
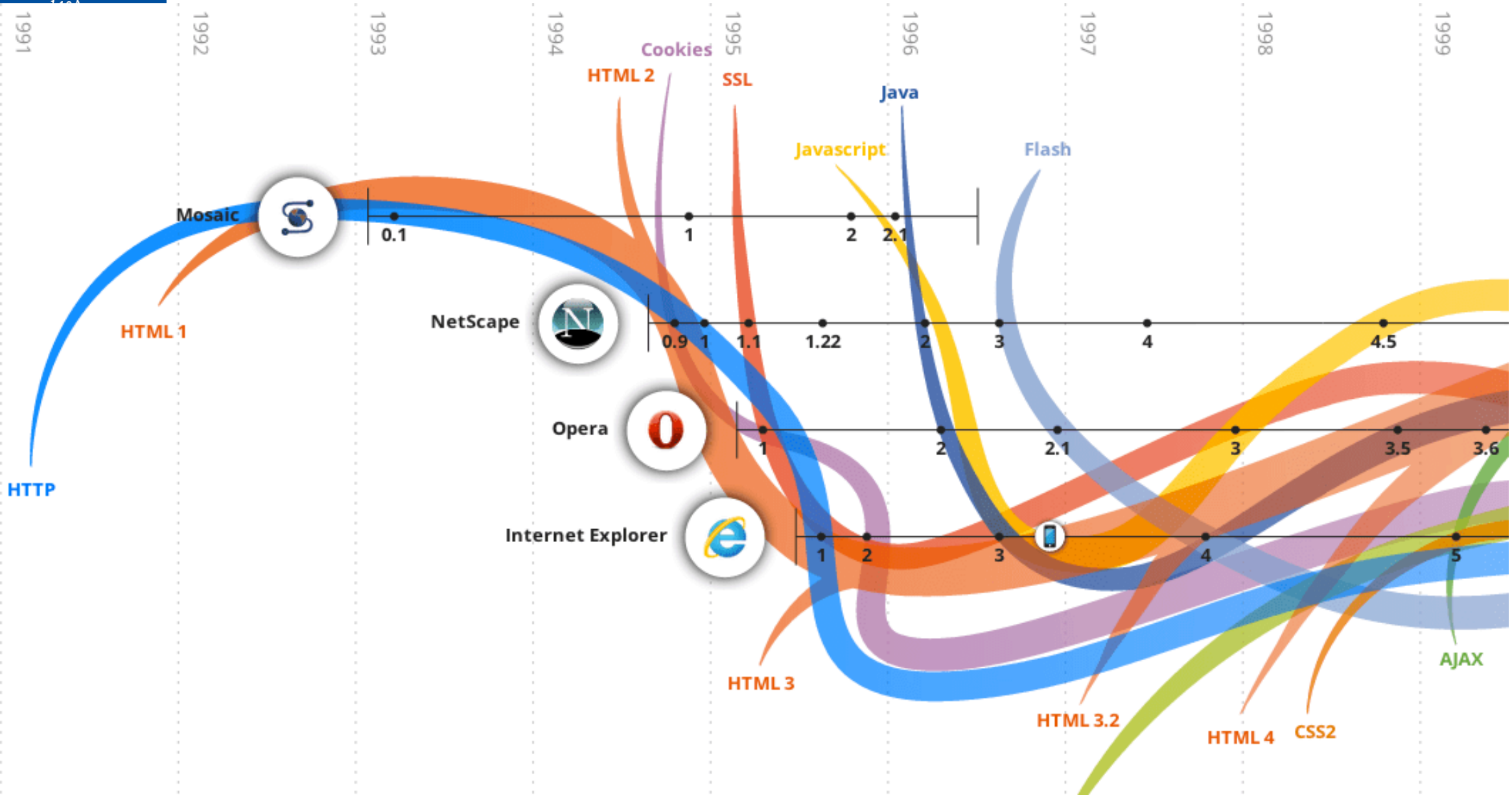
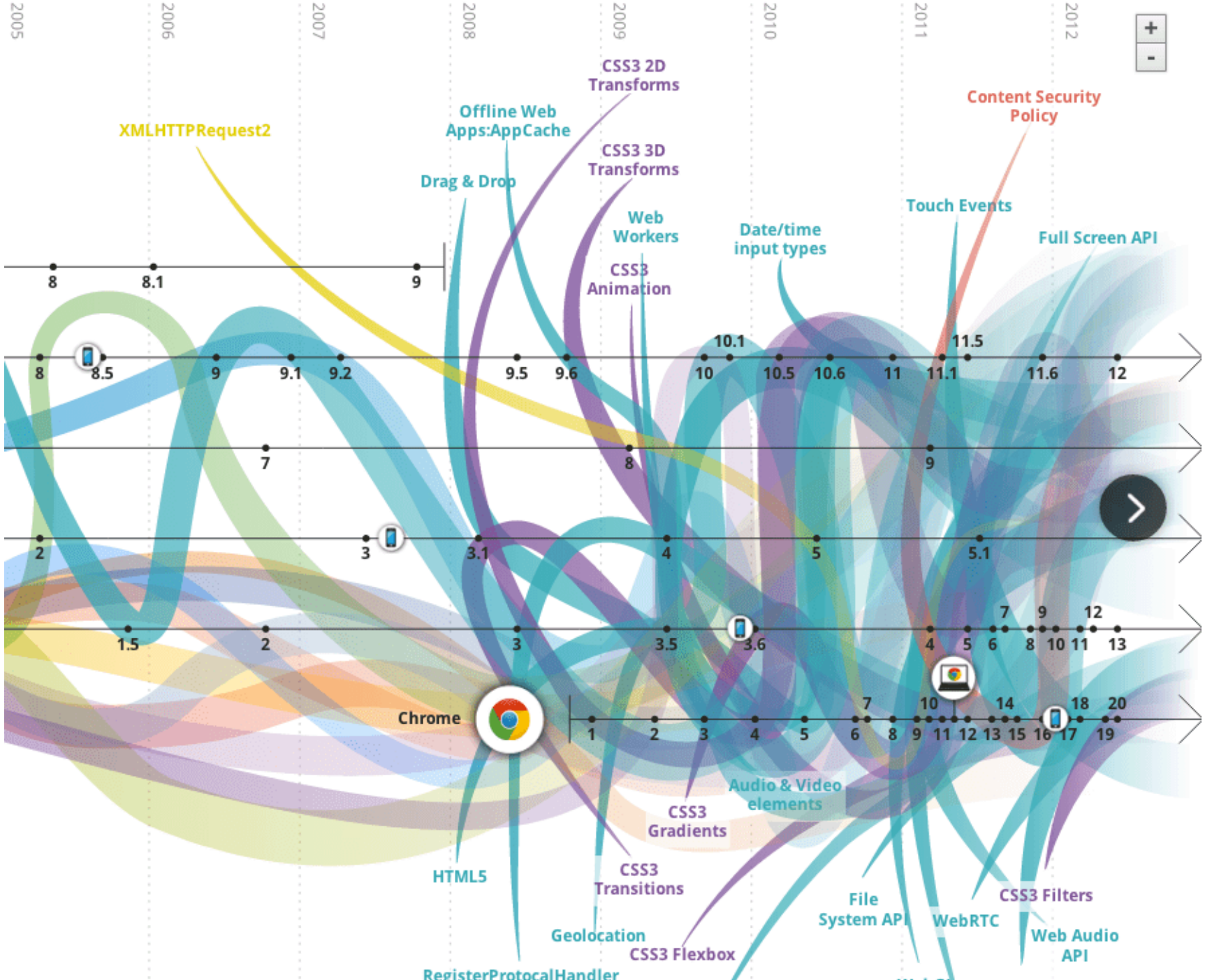
The Evolution of the web

Prof. Fabio Ciravegna
Dipartimento di Informatica
Università di Torino
fabio.ciravegna@unito.it



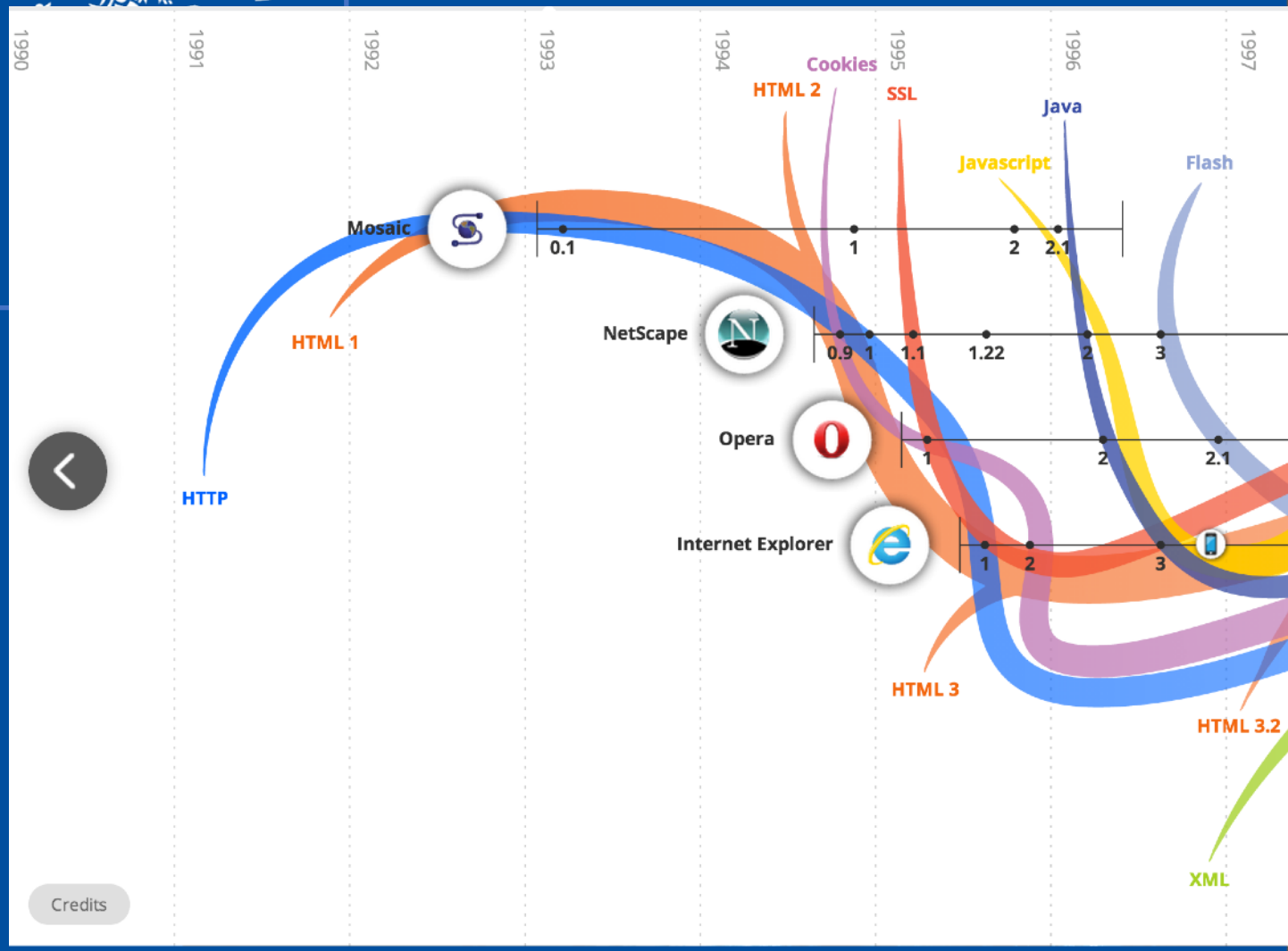


UNIVERSITÀ DI TORINO



The Http Protocol

Prof. Fabio Ciravegna
Dipartimento di Informatica
Università di Torino
fabio.ciravegna@unito.it



HTTP Protocol

- HTTP (for HyperText Transfer Protocol) is the primary method used to convey information on the World Wide Web http://en.wikipedia.org/wiki/Http_protocol
- HTTP is a protocol with the lightness and speed necessary for a distributed collaborative hypermedia information system.
- It is a generic stateless object-oriented protocol,
  - May be used for many similar tasks
    - E.g. name servers, and distributed object-oriented systems,

    by extending the commands, or "methods", used.

http://www.w3.org/Protocols/HTTP/HTTI

HTTP Protocol (ctd)

- A feature of HTTP is the negotiation of data representation, allowing systems to be built independently of the development of new advanced representations.

Connections in HTTP

- The http protocol is designed for client server architectures
- The protocol is basically stateless, a transaction consists of:
  - Connection
    - The establishment of a connection by the client to the server
  - Request
    - The sending, by the client, of a request message to the server;
  - Response
    - The sending, by the server, of a response to the client;
  - Close
    - The closing of the connection by either both parties.

we have seen request and response as parameter of any callback to the server in NodeJS

http://www.w3.org/Protocols/HTTP/HT

HTTP: GET and POST methods

- The GET method means "retrieve whatever information (…) is identified by the Request-URI".
  - e.g. Your browser requires a page (e.g. containing a form) from a server using a GET method
- The POST method is used to request that the origin server accepts the entity enclosed in the request [and acts upon it].
  - e.g. the browser POSTs the values for a form to the server

http://www.w3.org/Protocols/rfc2616/rfc26

Request Headers

- When an HTTP client sends a request, it is required to supply a request line (usually GET or POST).
- It can also send a number of other headers, all of which are optional
  - Except for Content-Length required for POS
- Here are the most common headers:
  - Accept: The MIME types the client (e.g. bro
  - Accept-Charset: The character set the client expects.
  - Accept-Encoding:
    - The types of data encodings (such as gzip) the client knows how to decode.

http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/

Request Headers (2)

- Accept-Language
  - The language the client is expecting, in case the server has versions in more than one language.
- Authorization, Authorization info,
  - Usually in response to a WWW-Authenticate header from the server.
- Content-Length
  - Obligatory for POST messages, how much data is attached
- Cookie
- From
  - email address of requester; only used by Web spiders and other custom clients, not by browsers
- Host
  - Host and port as listed in the *original* URL

Request Headers (3)

- If-Modified-Since
  - Only return documents newer than this, otherwise send a 304 "Not Modified" response
- Referrer
  - The URL of the page containing the link the user followed to get to current page
- User-Agent
  - Type of client (robot, browser, etc.)
  - Useful if server is returning client-specific content
  - Useful to identify the client: particularly important for Spiders
- Others:
  - UA-Pixels, UA-Color, UA-OS, UA-CPU (nonstandard headers sent by some Internet Explorer versions, indicating screen size, color depth, operating system, and cpu type used by the client's system)

Please note!

- Most of these headers are used to reduce the server bandwidth consumption
  - The four Accept* headers
  - If-modified since

Please note very often you will use the http request directly (e.g. via a direct call). In other cases you will use some forms of Web API (e.g. Twitter API). However all these APIs have equivalent information that is either set up automatically (e.g. user agent in the Twitter API) or via the parameters of the API itself

Not setting these means low marks in the assignment!

Responses Codes

http://www.w3.org/Protocols/rfc

http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/

Status Code	Associated Message	Meaning
100	Continue	Continue with partial request. (New in HTTP 1.1)
101	Switching Protocols	Server will comply with `Upgrade` header and change to different protocol. (New in HTTP 1.1)
200	OK	Everything's fine; document follows for `GET` and `POST` requests. This is the default for servlets; if you don't use `setStatus`, you'll get this.
201	Created	Server created a document; the `Location` header indicates its URL.
202	Accepted	Request is being acted upon, but processing is not completed.
203	Non-Authoritative Information	Document is being returned normally, but some of the response headers might be incorrect since a document copy is being used. (New in HTTP 1.1)
204	No Content	No new document; browser should continue to display previous document. This is a useful if the user periodically reloads a page and you can determine that the previous page is already up to date. However, this does not work for pages that are automatically reloaded via the Refresh response header or the equivalent `<META HTTP-EQUIV="Refresh" ...>` header, since returning this status code stops future reloading. JavaScript-based automatic reloading could still work in such a case, though.
205	Reset Content	No new document, but browser should reset document view. Used to force browser to clear CGI form fields. (New in HTTP 1.1)
206	Partial Content	Client sent a partial request with a `Range` header, and server has fulfilled it. (New in HTTP 1.1)
300	Multiple Choices	Document requested can be found several places; they'll be listed in the returned document. If server has a preferred choice, it should be listed in the `Location` response header.
301	Moved Permanently	Requested document is elsewhere, and the URL for it is given in the `Location` response header. Browsers should automatically follow the link to the new URL.
		Similar to 301, except that the new URL should be interpreted as a temporary replacement, not a permanent one. Note: the message was "Moved Temporarily" in HTTP 1.0, and the constant in

http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/

Responses (2)

401	Unauthorized	Client tried to access password-protected page without proper authorization. Response should include a `WWW-Authenticate` header that the browser would use to pop up a username/password dialog box, which then comes back via the `Authorization` header.
403	Forbidden	Resource is not available, regardless of authorization. Often the result of bad file or directory permissions on the server.
404	Not Found	No resource could be found at that address. This is the standard "no such page" response. **This is such a common and useful response that there is a special method for it in `HttpServletResponse`: `sendError(message)`.** The advantage of `sendError` over `setStatus` is that, with `sendError`, the server automatically generates an error page showing the error message.
405	Method Not Allowed	The request method (`GET`, `POST`, `HEAD`, `DELETE`, `PUT`, `TRACE`, etc.) was not allowed for this particular resource. (New in HTTP 1.1)
406	Not Acceptable	Resource indicated generates a MIME type incompatible with that specified by the client via its `Accept` header. (New in HTTP 1.1)
407	Proxy Authentication Required	Similar to 401, but proxy server must return a `Proxy-Authenticate` header. (New in HTTP 1.1)
408	Request Timeout	The client took too long to send the request. (New in HTTP 1.1)
409	Conflict	Usually associated with `PUT` requests; used for situations such as trying to upload an incorrect version of a file. (New in HTTP 1.1)
410	Gone	Document is gone; no forwarding address known. Differs from 404 in that the document is is known to be permanently gone in this case, not just unavailable for unknown reasons as with 404. (New in HTTP 1.1)
411	Length Required	Server cannot process request unless client sends a `Content-Length` header. (New in HTTP 1.1)
412	Precondition Failed	Some precondition specified in the request headers was false. (New in HTTP 1.1)
413	Request Entity Too Large	The requested document is bigger than the server wants to handle now. If the server thinks it can handle it later, it should include a `Retry-After` header. (New in HTTP 1.1)

UNIVERSITÀ DI TORINO

http://www.apl.jhu.edu/~hall/java/Servlet-Tutorial/

Responses (3)

414	Request URI Too Long	The URI is too long. (New in HTTP 1.1)
415	Unsupported Media Type	Request is in an unknown format. (New in HTTP 1.1)
416	Requested Range Not Satisfiable	Client included an unsatisfiable `Range` header in request. (New in HTTP 1.1)
417	Expectation Failed	Value in the `Expect` request header could not be met. (New in HTTP 1.1)
500	Internal Server Error	Generic "server is confused" message. It is often the result of CGI programs or (heaven forbid!) servlets that crash or return improperly formatted headers.
501	Not Implemented	Server doesn't support functionality to fulfill request. Used, for example, when client issues command like `PUT` that server doesn't support.
502	Bad Gateway	Used by servers that act as proxies or gateways; indicates that initial server got a bad response from the remote server.
503	Service Unavailable	Server cannot respond due to maintenance or overloading. For example, a servlet might return this header if some thread or database connection pool is currently full. Server can supply a `Retry-After` header.
504	Gateway Timeout	Used by servers that act as proxies or gateways; indicates that initial server didn't get a response from the remote server in time. (New in HTTP 1.1)
505	HTTP Version Not Supported	Server doesn't support version of HTTP indicated in request line. (New in HTTP 1.1)

Never ignore the response code of a

Why are error codes important?

https://www.wordbee.com/blog/localization-industry/10-mistranslated

Lost in translation: road sign carries email reply

https://www.theguardian.com/theguardian/2008/n

▲ The Welsh language road sign reading: 'I am out of the office at the moment', erected in Swansea. Photograph: PA

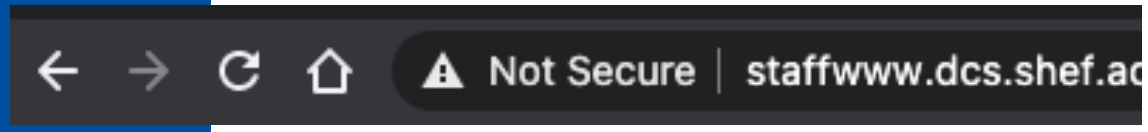
A council put up a Welsh language road sign reading "I am out of the office at the moment" when it should have said "No entry for heavy goods vehicles".

Swansea council contacted its in-house translation service when designing the bilingual sign. The seeds of confusion were sown when officials received an automated email response in Welsh from an absent translator, saying: "I am not in the office at the moment. Please send any work to be translated."

Unaware of its real meaning, officials had it printed on the sign. The council

Https

- Hypertext transfer protocol secure (HTTPS) is the secure version of HTTP
  - HTTPS is encrypted in order to increase security of data transfer
  - It is particularly important when sensitive data is sent
    - bank account information, emails, etc.
- Any websites, especially those that require login credentials, should use HTTPS
  - in the current web browsers such as Chrome, websites that do not use HTTPS are marked differently than those that are secure



Https

- HTTPS uses an encryption protocol to encrypt communications
  - Transport Layer Security (TLS) - Formerly known as Secure Sockets Layer (SSL)
- It secures communications by using an **asymmetric public key infrastructure**.
  - It uses two different keys to encrypt communications between two parties:
    - The private key
      - controlled by the owner - it is kept private. Used to decrypt information encrypted by the public key
    - The public key
      - available to everyone who wants to interact with the server securely
      - Information that's encrypted by the public key can only be decrypted by the private key

SSL Certificates

- SSL Certificates are small data files that digitally bind a cryptographic key to an organisation's identity
  - they bind together:
    - A domain name, server name or hostname.
    - An organisational identity (i.e. company name) and location
  - The certificate contain
    - the keys pair
    - the "subject," i.e. the identity of the certificate/website owner
    - the certificate is verified by an external authority
- You can get your free certificate from letsencrypt.org
  - a non profit authority

http://www.theguardian.com/technology/2015/feb/18/what-is-http2-speed-up-web-browsing-desktop-mobile

Internet

What is HTTP/2 and is it going to speed up the web?

Biggest change to how the web works since 1999 should make browsing on desktop and mobile faster

Samuel Gibbs

@SamuelGibbs

Wednesday 18 February 2015 15.10 GMT

The internet is set to get quicker as the biggest change to the protocols that run the web since 1999 arrives with HTTP/2. Photograph: Alamy

HTTP/2

http://en.wikipedia.org/wiki/HTTP/2

- HTTP/2 keeps most of HTTP 1.1's high level syntax,
  - Methods, status codes, header fields, and URIs.
- The element that is modified is
  - How data is framed and transported between the client and the server.
- Websites that are efficient minimise the number of requests required to render an entire page by minifying
  - reducing the amount of code and packing smaller pieces of code into bundles,
  - (without reducing its ability to function) resources such as images and scripts.

HTTP/2 ctd

http://en.wikipedia.org/wiki/HTTP/2

- However, minification is not necessarily convenient nor efficient,
  - it may still require separate HTTP connections to get the page and the minified resources.
- HTTP/2 allows the server to "push" content
  - to respond with data for more queries than the client requested.
  - It allows servers to supply data it knows web browser will need to render a web page, without waiting for the browser to examine the first response, and without the overhead of an additional request cycle
- Additional performance improvements come from
  - multiplexing of requests and responses to avoid the head-of-line blocking problem in HTTP 1
  - header compression, and prioritization of requests

Going beyond the limitations of http

- the protocol has been fantastic and brought the Web from a text only document linkage to a very sophisticated environment
  - However, its limitations have been a main issue over the past few year, especially:
    - memoryless
    - client-initiated
- Today we will see how we go beyond that

What you should remember

- How a connection is created with the http protocol
  - connection, request, response, closure
  - as this is relevant to most APIs you will ever use
- The main error codes
  - never ignore the error code
  - remember the guy who ignored an away message and printed it in Welsh

UNIVERSITÀ
DI TORINO

Questions?