

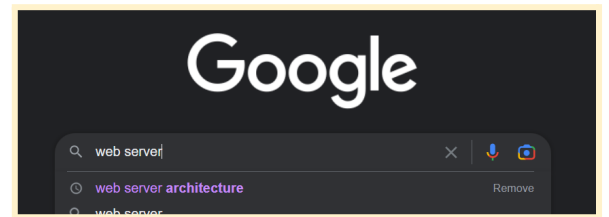
## Lecture 2

### Basic aspects of Web Applications

: Web server architecture

ايه ال web server :

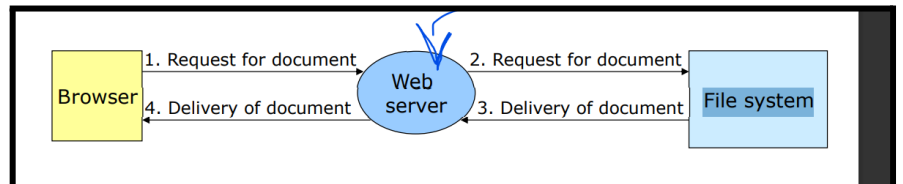
هو عبارة عن software بيخلي ال resources available من خلال an interface بواسطة HTTP  
لما انت search عن اي حاجة



ال h /Web servers that deliver static content only/

ال web server بيتعامل مع ال static content بس

ال request ده بيروح الاول علي ال web server و بعد كده ال web server بيبيع امر لل File system  
و لما يرد عليه بيبيع الرد لل browser

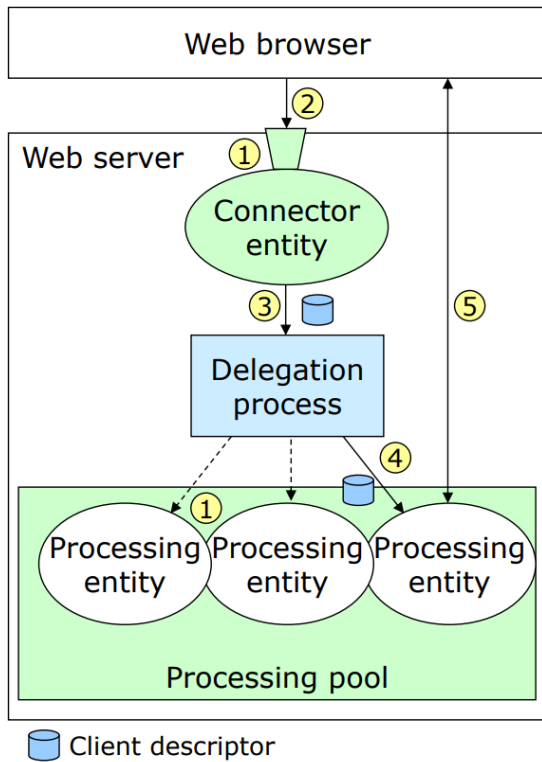


بما ان انت بتعامل مع low latency time /وقت استجابة منخفض و handle high load

فال current web servers بقت بتبني بطريقة ال hierarchical way الطريقة دي بتسمح بال

parallel request processing / انت تعمل اكتر من Request ف نفس الوقت

طيب دلوقتي هنشرح الجزء الخاص كيف ال web server بيبدأ بيستلم الطلبات و ينفذها .  
\_ عندنا الرسمه دي .



– طيب اول حاجات ال web server بيبدأ يجمع كل الطلبات الي جياته داخل حاجه اسمها ال Connector entity .

– وبعدين ال Connector entity بيبدأ يعمل حاجات تانيه وهي ال client descriptor ( الي هي الحاجه الزرقه الي جنب الرقم 3 ) ودي اللي بتكون آخر نقطة وصل بين ال web server و ال web browser .

– بعدين بيبدأ ال delegation process هو الي بقول ارد على مين الاول من الطلبات الي جياته . وبعد ما يحدد ده بيبدأ بيعت ال client descriptor ل Processing pool و الي هو يحتوي أصلاً

على مجموعه من ال Processing entity الي كل واحده منهم بتبدأ تنفذ طلب معين .

– و ال Communication الي بيحصل بين ال web browser و ال web Server بيكون عن طريق ال HTTP

## Server-side logic

طيب دلوقتي احنا عايزين نتكلم عن نوع ثاني من الـ web والي بيكون فيه **dynamic** طيب هو يعني ايه **dynamic** يعني يكون فيه **function** و حركة وكود مش حابه ثابتة كده . والي هي بتعتمد علي

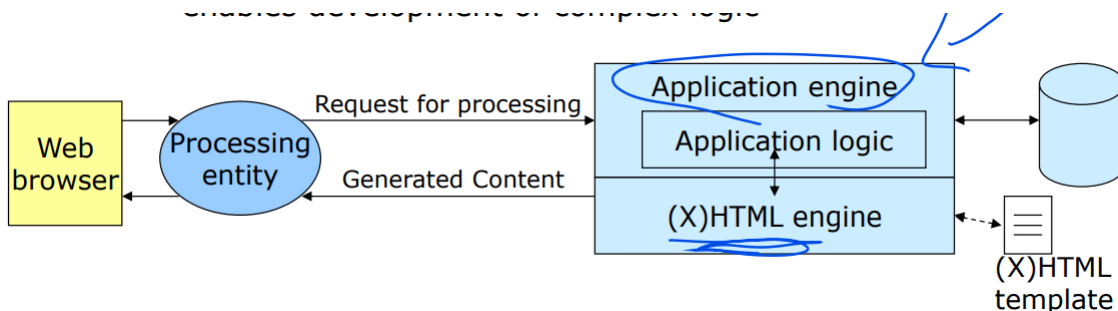
Web applications are based on **dynamic content** which is generated by application logic based on e.g.:

1- Common Gateway Interface (CGI) Scripts

2- Java Servlet Container

3- Application Server that provides further services such as **transaction, security or directory services** and thus **enables development of complex logic**

عشان كده بنستخدم مع الـ dynamic الـ **development of complex logic**

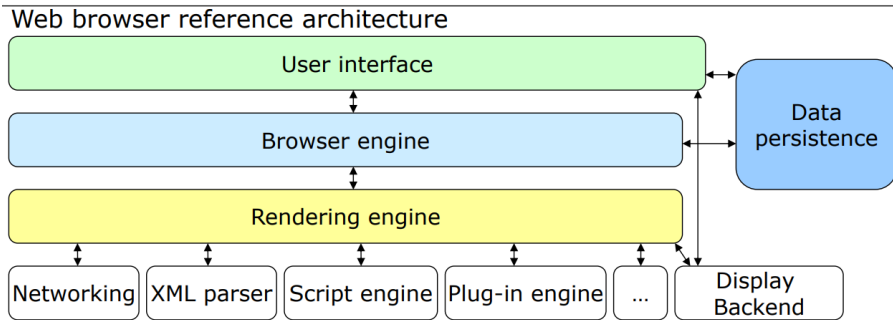


هنا الـ **application engine** بيبدأ يستخدم الـ **(X)HTML engine** علشان يعمل دمج بين الـ **dynamic content** و الـ **(X)HTML templates** وفي الآخر يطلع الناتج واللي هيكون عبارة عن **documents** تتبع في الآخر للـ **Client** الي هو الـ **web browser**

The **application engine** may use an **(X)HTML engine** to **combine dynamic content** with **(X)HTML templates** to generate **resulting documents** that finally are delivered to the **web browser**.

الـ web browser كان زمان عبارة عنه بيعت requests علشان تحصل علي بيانات معينه و خلاص انما دلوقتي هو اتوسع و بقا يضم حاجات تانيه زي الـ

### XML support, script engines and plug-in engines



الـ User interface: هو عبارة عن Front-end for يعرض الصفحة الي جياك

الـ Browser engine: عبارة عن Embedded component بيوفرلك a high level interface ليه  
؟؟ علشان تعمل

### querying and using the rendering engine

الـ Rendering engine: بيعمل parsing and layout للـ (X)HTML documents

مع الـ other languages زي الـ CSS /

الـ Networking: بت Realises الـ HTTP communication مع الـ server

الـ XML parser: بيعمل Parses / تحليل للـ XML content

الـ Script engine: بينفذ اي scripts embedded in (X)HTML pages

الـ Display backend: بيوفر

drawing and windowing primitives, user interface widgets and fonts (e.g. GNU Image Manipulation Program Toolkit (GTK+))

الـ Data persistence: بيخزن الـ associated data زي الـ (.cache pages, cookies etc)

الـ Plug-in engine: هي Dynamic extension point for plug-ins

زي ال Web feed reader plug-in

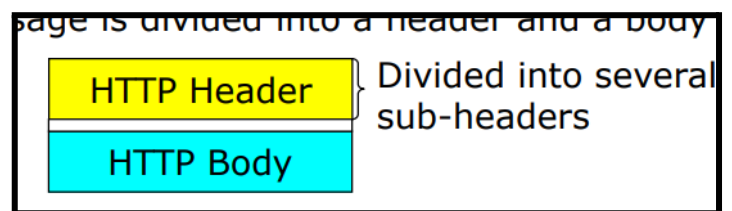
## HTTP / Hypertext Transfer Protocol

بي realises communication on top of TCP عن طريق ال

the exchange of messages in a request-response manner / تبادل الرسائل بطريقة

الطلب والرد

و ال messages بتبقي متقسمة ل a header and a body



\_ The header specifies the operation that should be performed on the addressed resource and includes parameters (passed as key-value pairs)

\_ هو بيبدأ يحدد ايه هي العملية التي هيتنفذ على ال the addressed resource

- و ال Resources بتبقي مترتبة حسب اللي عمل REQUEST الاول .. مين اللي برتبهم؟؟

ال (Uniform Resource Identifiers (URI

طيب هو ايه الفرق بين ال http 0 و ال http 1.1 :

ال http 0 لما كان بيبقى عايز حاجه من ال server كان اول حاجه يعملها يفتح connection معاه و بعدين يعمل request وبعد ما يستقبل ال response بيبدأ يقفل ال connection الي كان بينهم

\_ انما ال HTTP 1 بيبعت request و يرد عليا response و ممكن كرري العملية دي اكثر من مره  
انما بعد ماخلص اقفل ال connection ده .

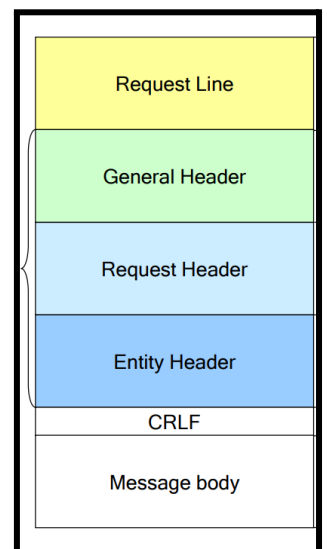
عشان الـ HTTP 1.1 بيوفر الـ a persistent connection / اتصال مستمر اللي makes communication more efficient

ـ طيب انا دلوقتي عايز ابعت اكثر من request في نفس الوقت طيب انا ممكن احل المشكلة دي ازاي

ـ ده هيكون عن طريق الـ HTTP 1.1 pipelining

- الـ HTTP 1.1 pipelining بيمكنك تبعت اكثر من request من غير ما تستنى الرد علي اي request قديم
- و الرد / response بقي مترتب و هو راجع علي حسب الترتيب اللي اتبعت بيه كل request
- و الـ HTTP 1.1 pipelining مش متوفر في كل انواع الـ browsers

الـ HTTP 1.1 request message / بتبقي متقسمة ازاي



الـ Request Line : بيبقى شايلى 3 حاجات

- method to be performed
- associated web resource
- the HTTP version

مثال عليه : GET /news/index.html HTTP/1.1

ال General Header :

ببقي فيه شوية general information زي ال the type of connection  
أو the date of request generation / الوقت اللي اتعمل فيه ال request

ال Request Header :

بيبعث information عن ال the request and the client ل server

زي ال the accepted content encoding

ال Entity Header :

بيبعث ال meta information عن ال the body content

زي ال the content length

ال CRLF :

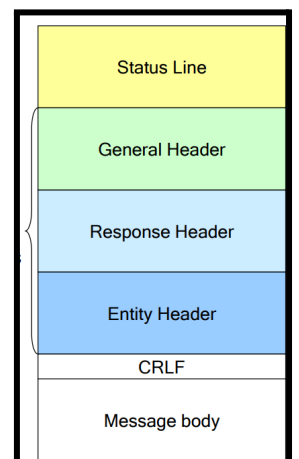
ال Content ببقي مفصول بال (CRLF new line)

ال Message body :

تعرض ال content بتاع ال request

زي ال parameters passed to the server

ال HTTP 1.1 response message / بتبقي متقسمة ازاى



الـ Status Line :

ببقي فيه حاجتين

- الـ HTTP protocol version

- الـ status code of request

مثال : HTTP/1.1 200 OK

الـ General Header :

ببقي فيه شوية general information زي الـ the type of connection

أو the date of response generation / الوقت اللي اتعمل فيه الـ response لل request

الـ Response Header :

يوفر معلومات عن الـ Response زي الـ

product data of the web server that produced the response

الـ Entity Header :

ببعت الـ meta information عن الـ the body content

زي الـ the content length

الـ HTTP 1.1 methods

- كل method بتعرف الـ resource من خلال الـ (Uniform Resource Identifier)
- و لول مفيش resource بتبعت asterisk / نجمة \*
- الـ HTTP 1.1 عبارة عن expandable set of methods / مجموعة من methods القابلة للتوسيع



HTTP 1.1 defines an expandable set of methods:

<b>GET</b>	Requests for delivery of the specified resource
<b>HEAD</b>	Identical to GET except the message body of the response is empty thus the client only receives meta information of the header; can e.g. be used to determine the type of a file or its length without the need to receive the file
<b>POST</b>	Submits the message body content to the specified resource
<b>PUT*</b>	Instructs the server to make the passed content available under the specified URL
<b>DELETE*</b>	Deletes the specified resource
<b>OPTIONS*</b>	Requests communication options from server such as e.g. the methods that are supported by the server
<b>TRACE*</b>	Echoes back the send request; can be used for diagnostic purposes
<b>CONNECT*</b>	Reserved name for use by proxy servers

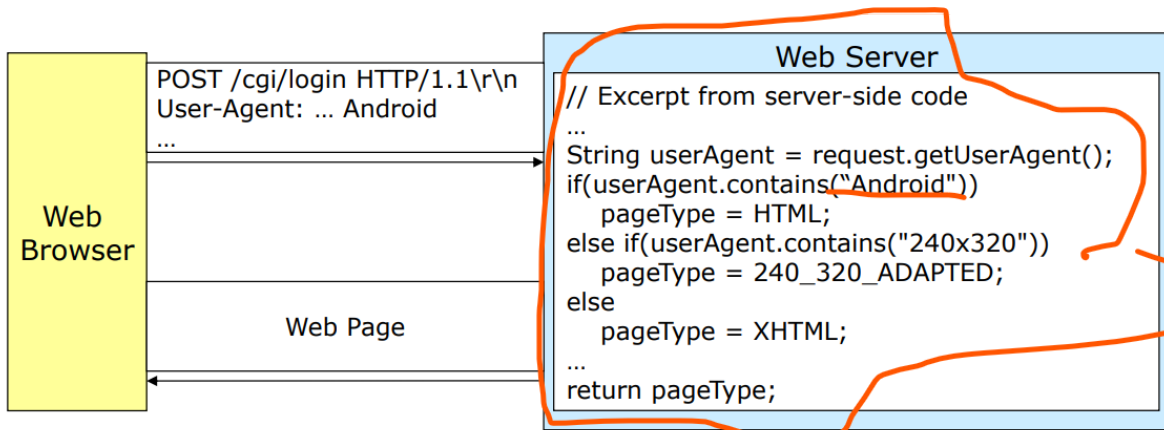
\* Not defined by HTTP 1.0

slide 16

Code	Status category	Description	Example : Explanation
<b>1xx</b>	<b>Informational</b>	Request received, continuing processing	<b>100</b> : Client should continue sending the request (e.g. extensive request)
<b>2xx</b>	<b>Success</b>	The action was successfully received, understood and accepted	<b>200</b> : Request has succeeded
<b>3xx</b>	<b>Redirection</b>	Further action must be taken in order to complete the request	<b>301</b> : Requested resource has been moved permanently to new URI
<b>4xx</b>	<b>Client error</b> (Request error)	The request contains bad syntax or cannot be fulfilled	<b>404</b> : Server has not found anything matching the Request-URI
<b>5xx</b>	<b>Server error</b>	The server failed to fulfil an apparently valid request	<b>501</b> : Server does not support the functionality required to fulfil the request

## HTTP user agent detection

في الجزء ده انت بتبدا تبعت request لل server و لكن بيكون في كود جوا ال server بيحدد انت جاي مينين بالظبط ( جاي من windows ولا android ) وده علشان يقدر يعرض المحتوي بتاعك و يحدد مقاسات الشاشة هتكون ازاي و الكلام ده .



## ال Session handling

- ده بيكون ال cookies اللي بتلاقيها في اي موقع والي بيكون الفايدة منها انها تجمع شويه معلومات عنك طول مانت في الموقع ده زي مثلا أي هي الصفحات لي دخلت عليها و ايه اللي انت بتعمله داخل الموقع ده و ايه الوقت اللي انت خدته في الموقع ده .
- فترة ال session العادية هي 20 دقيقة انما انت ممكن تغير المدة دي براحتك
- ال A session context هو اللي بيحط ال state for Web applications
- وقت ما السيشن مفتوحة ال browser ممكن يخن data عن ال user في special data structure
- عشان ال identification purpose ال web server بيعت ال session id لل web browser
- من اشهر ال technique لل session id exchange هو ال cookies

ايه هو ال cookies :

هي are small pieces of data stored on client-side

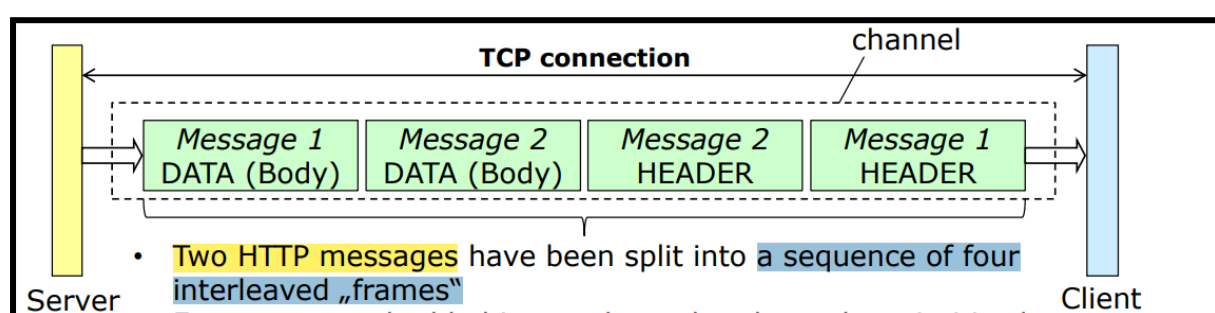
ال HTTP/2 :

ملاحظة صغيرة كده علي ال http1 وهي انه تم تحسينه في الاول من اجل التبسيط و التطبيق على نطاق واسع ولكن لم يتم تحسين البروتوكول بتاعه

\_ طيب الهدف من ال http2 وهو توفير transport for HTTP semantics وده هيكون باستخدام

ال **multiplexing and header compression**

\_ طيب هو انت ممكن تشوف ال http 2 ده لما تيجي تشغل فيديو علي اليوتيوب بتلاقيه قسم الفيديو بتاعك لمجوه من الاجزاء ودي الي بنقول عليها frames بحيث انت تشوف جزء جزء اول باول



و ال Two HTTP messages اللي هما ال request و ال response اتقسموا ل

"a sequence of four interleaved „frames"

HTML5 - Overview

