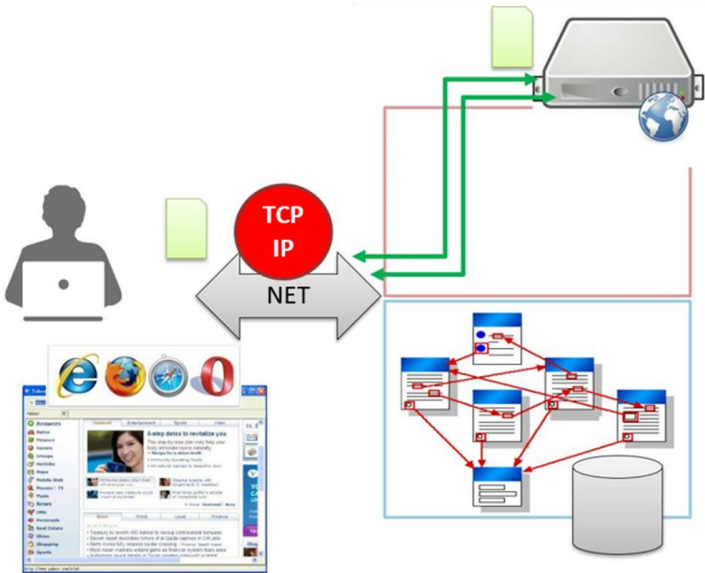




Front-end and Web Service Client

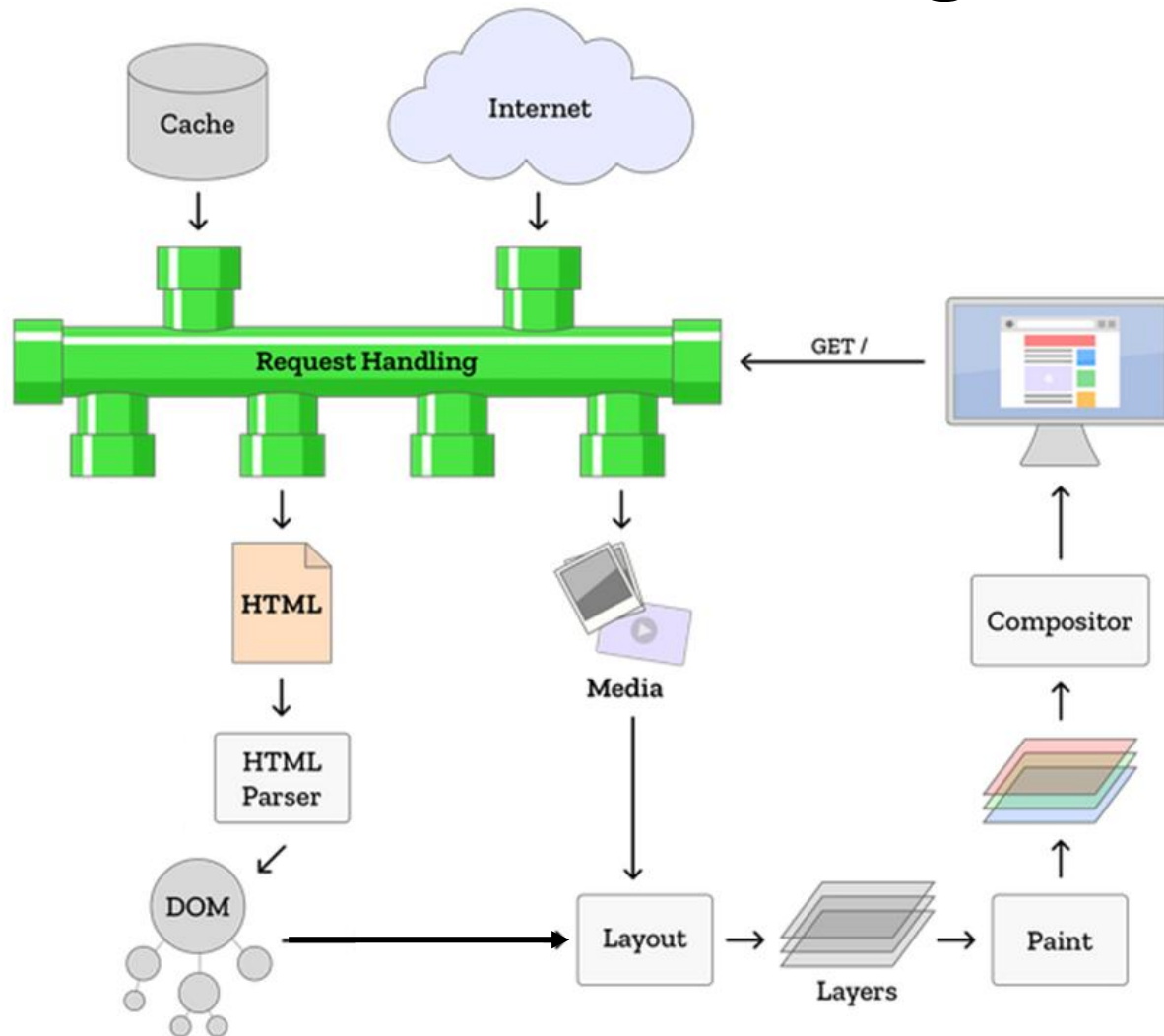
A Web Service Client in a Web Browser



- When the browser offers more and more features ...
- ... to become the most popular UI for Web and Web service client.
- That's what a front-end is !

First Web Browser generation

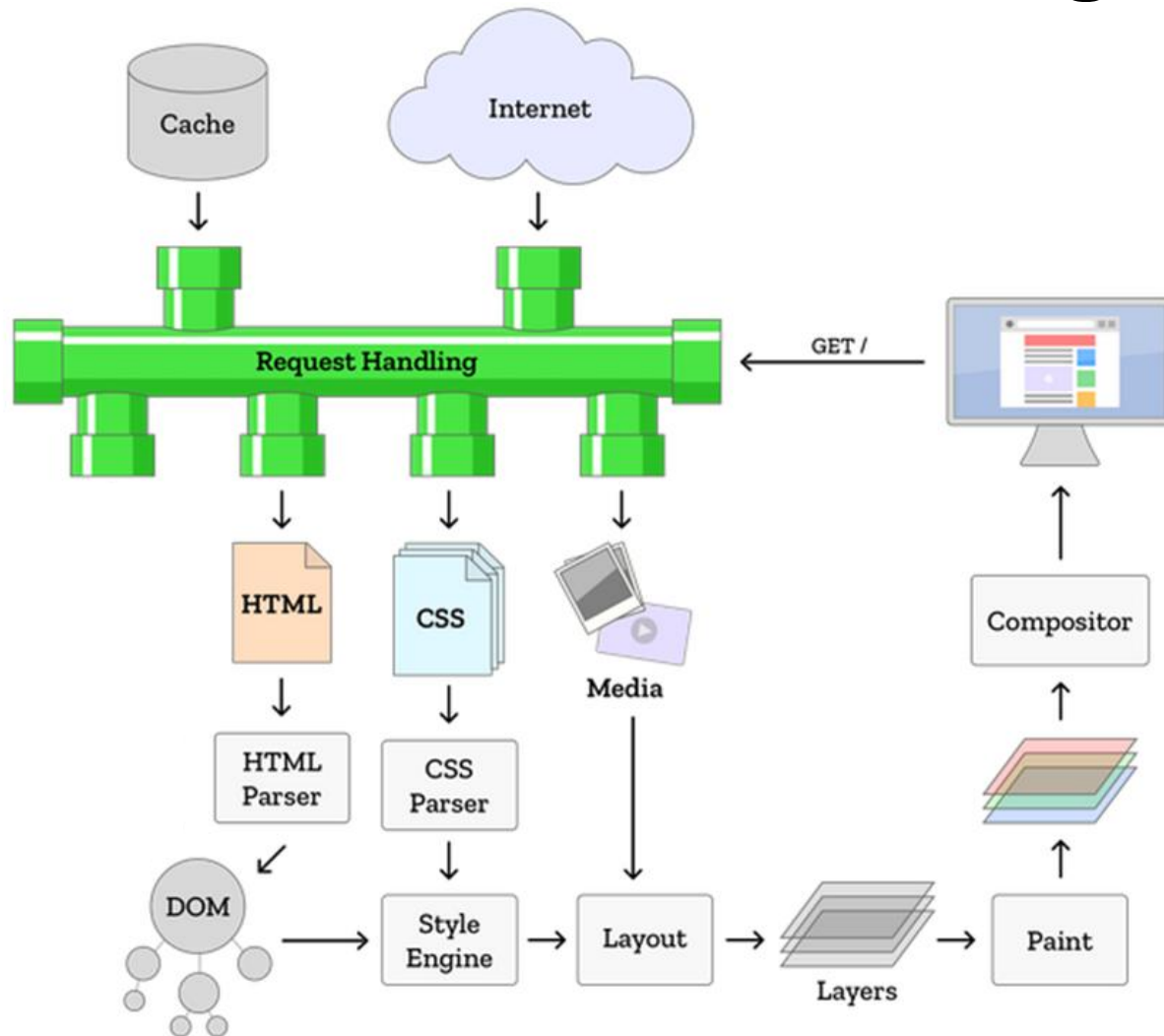
- HTML (HyperText Markup Language) describes the **content** of the page. (+ media)



HTML Overview

- Markup Language: `<tag attribute=attributeValue> foo </tag>` or `<tag attribute=attributeValue />`
- Architecture of a HTML page:
 - `<html lang="fr">` → Opening tag (with language)
 - `<head></head>` → Hidden. Used for configuration (title, SEO, CSS loading, ...)
 - `<body></body>` → Content of the page
 - `</html>` → Closing tag
- 2 types of tag:
 1. Block elements (e.g. `<div> foo </div>`) takes all the available width, and can be resized.
 2. Inline elements (e.g. ` bar `) takes only the width they need, and cannot be resized.
- Comments in HTML: `<!-- -->`
- List of all HTML tags:
<https://www.w3schools.com/TAGS/default.ASP>

Second Web Browser generation



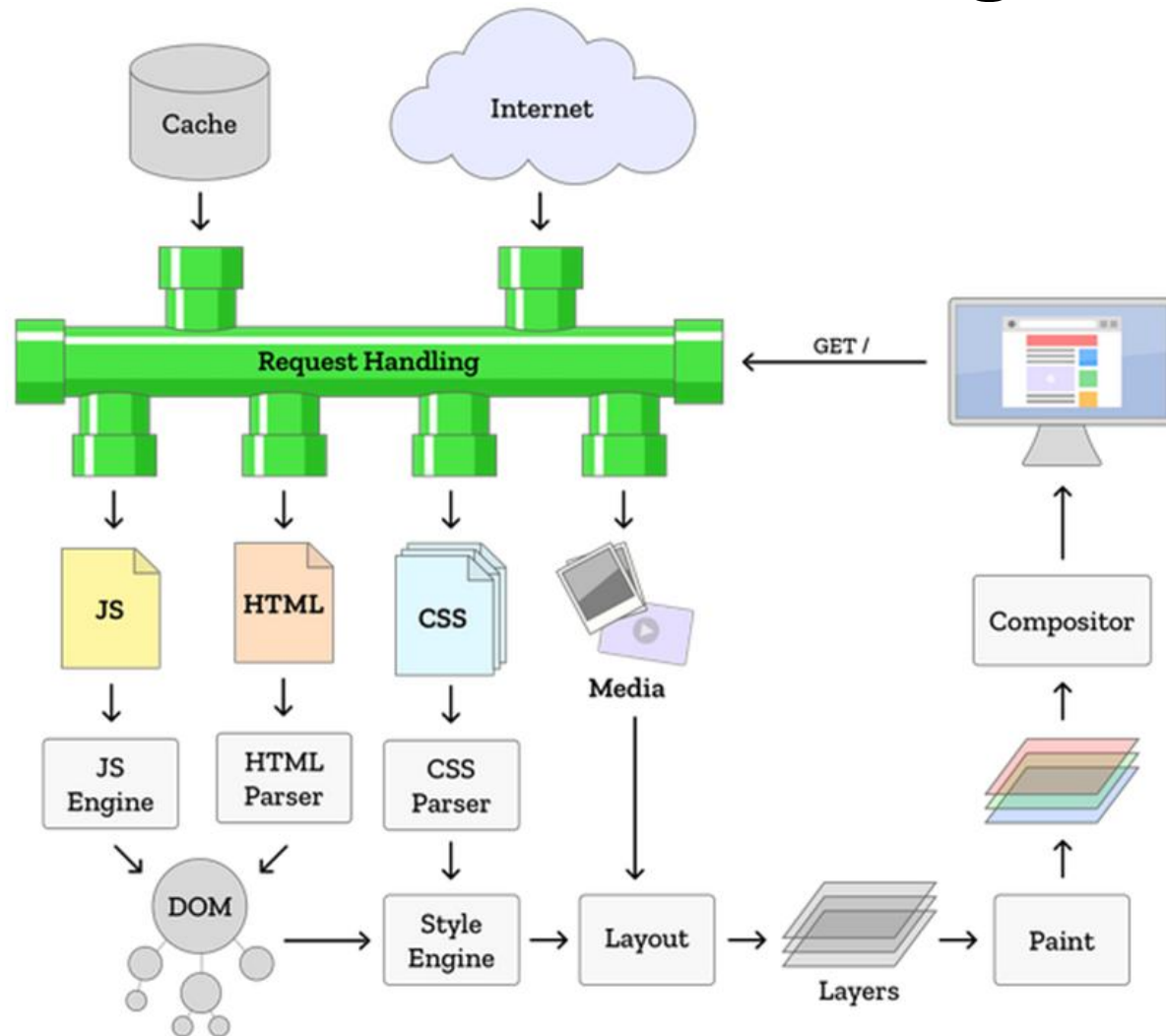
- HTML
- CSS (Cascading Style Sheets) defines the **style** and **positioning** of the content.

CSS Overview

- 3 main levels of selection:
 1. `#foo` → The only element with an attribute **id** whose value is "foo"
 2. `.bar` → All the elements with an attribute **class** containing "bar"
 3. `img` → All `` elements
- Cascading:
 - Rules can be overwritten.
 - In case of multiple rules targeting the same element(s), they are merged and all apply.
 - If the same CSS rule exists in multiple selectors, the kept value is chosen by the precision of the selector (id > class > tag > order (LIFO))
- Syntax:

<ul style="list-style-type: none">• <code>Selector1, selector2 {</code> <code>action1: value1;</code> <code>action2: value2;</code> <code>}</code>	→	<code>#foo, .bar {</code> <code>font-size: 15px;</code> <code>margin-right: 10px;</code> <code>}</code>
--	---	--

Third Web Browser generation



- HTML
- CSS
- JS (Javascript) manages the **behaviour** of the content.

JS Overview

- Included via `<script>` tag
- Programming Language:
 - Arrays, dictionaries, loops, “objects”, functions, variables, ...
 - Used mainly by browsers but also on the OS (NodeJs for instance)
- Fields of application:
 - DOM (Document Object Model) → Javascript can retrieve and modify DOM elements’ content and style (e.g. `document.getElementById(“foo”).style.color = “#f00”;`)
 - External communications → Javascript can interact with servers to retrieve data without having to refresh the page (<https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>)
- Different JS versions:
 - Whenever an addition or a modification is made to javascript, browsers have to implement it, and that’s not always the case
 - Compiler (e.g babel) are used to transform a js code into another one, compatible with all browsers*
 - <https://caniuse.com/> to know how well a version is implemented in the different browsers

Web service invocation in Javascript

- The [Fetch API](#) is not available on Internet Explorer, so we will use [XMLHttpRequest](#)
- SOAP servers furnish a description, in a WSDL (Web Service Description Language) file
- Many languages / frameworks contain a tool to convert WSDL in js, to be used to communicate with the server (for instance with [Apache](#), [Python](#), ...)