**Abstract**

The World Wide Web is a networked information system. Web Architecture consists of the requirements, constraints, principles, and choices that influence the design of the system and the behavior of agents within the system. When Web Architecture is followed, the large-scale effect is that of an efficient, scalable, shared information space. The organization of this document reflects the three divisions of Web architecture: identification, representation, and interaction. This document also addresses some non-technical (social) issues that play a role in building the shared information space.

This document strives to establish a reference set of requirements, constraints, principles, and design choices for Web architecture.

## 1. Introduction

The World Wide Web (or, Web) is a networked information system consisting of **agents** (programs acting on behalf of a person, entity, or process) that exchange information. Here's a simple travel scenario illustrating a common Web interaction:

* While planning a trip to Mexico, Dan reads "Oaxaca weather information: http://weather.example.com/oaxaca" in a glossy travel magazine. Dan has enough experience with the Web to recognize that http://weather.example.com/oaxaca is a URI. He can expect that the URI should allow him to access relevant weather information.
* Dan enters the URI into his browser, which downloads information. The information is provided by those responsible for weather.example.com.
* Dan's user agent displays the downloaded information. The page may include links to other information (i.e., more URIs) allowing Dan to start the cycle again.

This scenario illustrate the three architectural divisions of the Web that are discussed in this document:

1. [Identification](https://www.w3.org/TR/2003/WD-webarch-20030627/#identification). Objects in the networked information system called **resources** are identified by Uniform Resource Identifiers (URIs). The URI in the travel scenario ishttp://weather.example.com/oaxaca.
2. [Representation](https://www.w3.org/TR/2003/WD-webarch-20030627/#representations). Agents (such as servers, browsers and multimedia players) communicate resource state through a non-exclusive set of data formats, used separately or in combination (e.g., XHTML, CSS, PNG, XLink, RDF/XML, SVG, SMIL animation). In the travel scenario, Dan's user agent uses the URI to request a representation of the identified resource. In this scenario, the representation consists of XHTML with embedded weather maps in SVG.
3. [Interaction](https://www.w3.org/TR/2003/WD-webarch-20030627/#interaction). Agents exchange representations via a non-exclusive set of protocols, including HTTP, FTP, and SMTP[1](https://www.w3.org/TR/2003/WD-webarch-20030627/" \l "smtp1). In the travel scenario, Dan's browser uses HTTP to download the representation.[2](https://www.w3.org/TR/2003/WD-webarch-20030627/" \l "uris-and-protocols)

*Editor's note*: Todo: Introduce notions of client and server. Relation of client to agent and user agent. Relation of server to resource owner.

## World Wide Web Protocols

Technically the World-Wide Web hinges on three enabling protocols, the HyperText Markup Language (HTML) that specifies a simple markup language for describing hypertext pages, the Hypertext Transfer Protocol (HTTP) which is used by web browsers to communicate with web clients, and Uniform Resource Locators (URL's) which are used to specify the links between documents.

### **HyperText Markup Language**

The hypertext pages on the web are all written using the Hypertext Markup Language (HTML), a simple language consisting of a small number of tags to delineate logical constructs within the text. Unlike a procedural language such as postscript (move 1 inch to the right, 2 inches down, and create a green WWW in 15 pointer bold helvetica font), HTML deals with higher level constructs such as "headings," "lists," "images," etc. This leaves individual browsers free to format text in the most appropriate way for their particular environment; for example, the same document can be viewed on a MAC, on a PC, or on a linemode terminal, and while the content of the document remains the same, the precise way in which it is displayed will vary between the different environments.

The earliest version of HTML (subsequently labeled HTML1), was deliberately kept very simple to make the task of browser developers easier. Subsequent versions of HTML will allow more advanced features. HTML2 (approximately what most browsers support today) includes the ability to embed images in documents, layout fill-in forms, and nest lists to arbitrary depths. HTML3 (currently being defined) will allow still more advanced features such as mathematical equations, tables, and figures with captions and flow-around text.

### **Hypertext Transfer Protocol**

Although most Web browsers are able to communicate using a variety of protocols, such as FTP, Gopher and WAIS, the most common protocol in use on the Web is that designed specifically for the WWW project, the HyperText Transfer Protocol. In order to give the fast response time needed for hypertext applications, a very simple protocol which uses a single round trip between the client and the server is used.

In the first phase of a HTTP transfer the browser sends a request for a document to the server. Included in this request is the description of the document being requested, as well as a list of document types that the browser is capable of handling. The Multipurpose Internet Mail Extensions (MIME) standard is used to specify the document types that the browser can handle, typically a variety of video, audio, and image formats in addition to plain text and HTML. The browser is able to specify weights for each document type, in order to inform the server about the relative desirability of different document types.

In response to a query the server returns the document to the browser using one of the formats acceptable to the browser. If necessary the server can translate the document from the format it is stored in into a format acceptable to the browser. For example the server might have an image stored in the highly compressed JPEG image format, and if a browser capable of displaying JPEG images requests the image it would be returned in this format. However if a browser capable of displaying images only if they are in GIF format requested the same document the server would be able to translate the image and return the (larger) GIF image. This provides a way of introducing more sophisticated document formats in future but still enabling older or less advanced browser to access the same information.

In addition to the basic "GET" transaction described above the HTTP is also able to support a number of other transaction types, such as "POST" for sending the data for fill-out forms back to the server and "PUT" which might be used in the future to allow authors to save modified versions of documents back to the server.

### **Uniform Resource Locators**

The final key to the World-Wide Web is the URL's which allow the hypertext documents to point to other documents located anywhere on the web. A URL consists of 3 major components:

<protocol>://<node>/<location>

The first component specifies the protocol to be used to access the document, for example, HTTP, FTP, or Gopher, etc. The second component specifies the node on the network from which the document is to be obtained, and the third component specifies the location of the document on the remote machine. The third component of the URL is passed without modification by the browser to the server, and the interpretation of this component is performed by the server, so while a documents location is often specified as a Unix-like file specification, there is no requirement that this is how it is actually interpreted by the server.