WAP - The Model

Before we describe WAP model, first we would like you to understand how Standard Internet works.

The Internet Model

The Internet model makes it possible for a client to reach services on a large number of origin servers, each addressed by a **unique Uniform Resource Locator** (URL).

The content stored on the servers is of various formats, but HTML is the predominant. HTML provides the content developer with a means to describe the appearance of a service in a flat document structure. If more advanced features like procedural logic are needed, then scripting languages such as JavaScript or VB Script may be utilised.

The figure below shows how a WWW client request a resource stored on a web server. On the Internet standard communication protocols, like HTTP and Transmission Control Protocol/Internet Protocol (TCP/IP) are used.
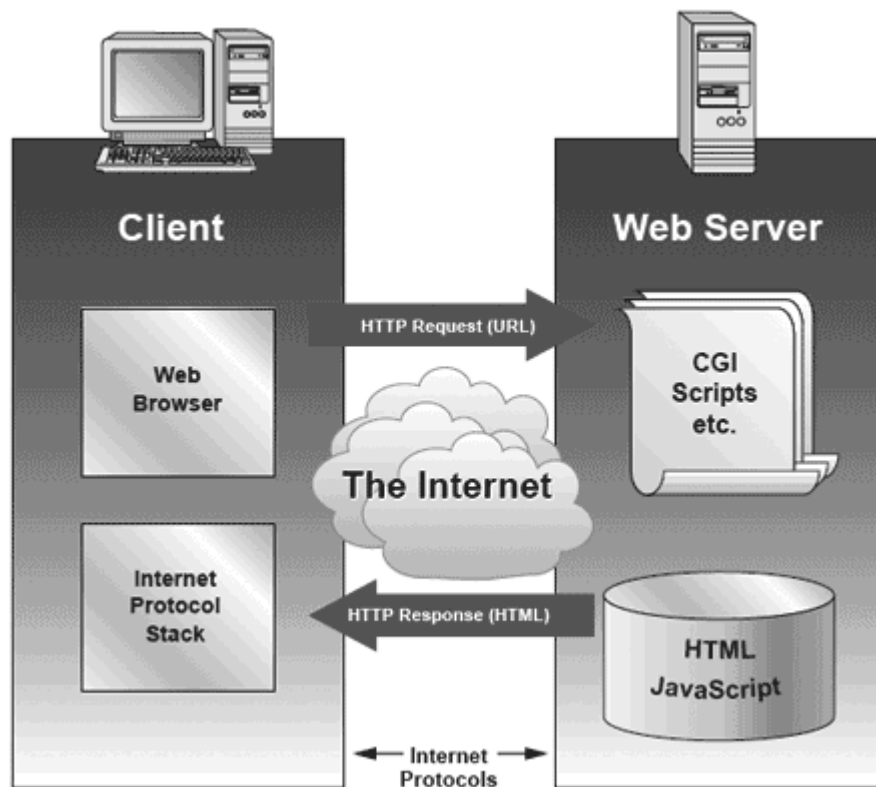
The content available at the web server may be static or dynamic. Static content is produced once and not changed or updated very often; for example, a company presentation. Dynamic content is needed when the information provided by the service changes more often; for example, timetables, news, stock quotes, and account information. Technologies such as Active Server Pages (ASP), Common Gateway Interface (CGI), and Servlets allow content to be generated dynamically.

The WAP Model

The figure below shows the WAP programming model. Note, the similarities with the Internet model. Without the WAP Gateway/Proxy, the two models would have been practically identical.

1 -- WSP Request (URL)
2 -- WSP Response (Binary WML)
3 -- HTTP Request (URL)
3 -- HTTP Response (WML)
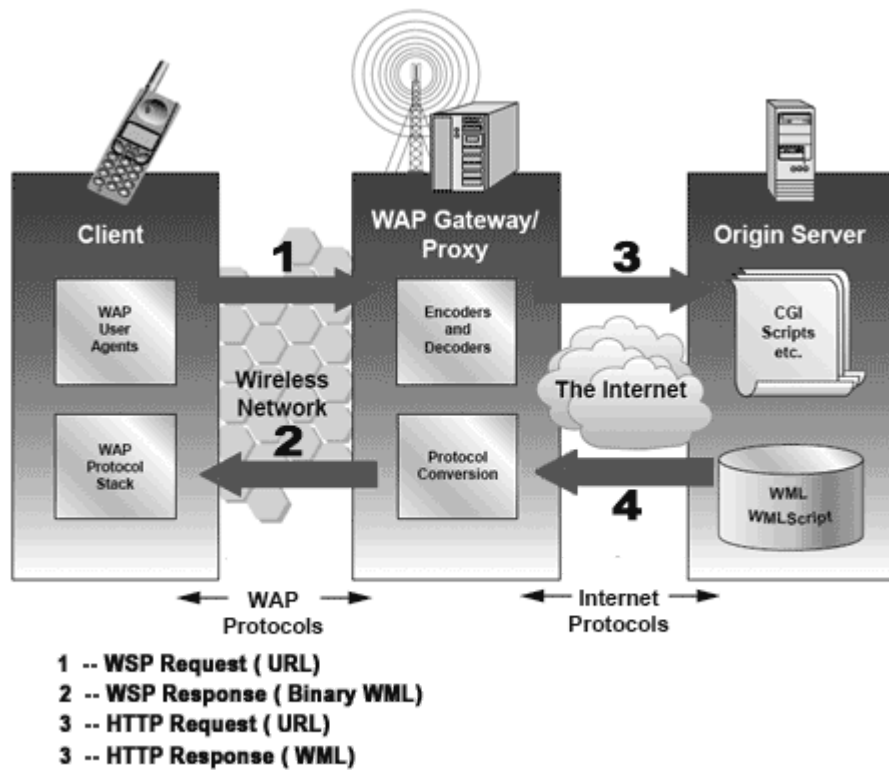
WAP Gateway/Proxy is the entity that connects the wireless domain with the Internet. You should make a note that the request that is sent from the wireless client to the WAP Gateway/Proxy uses the Wireless Session Protocol (WSP). In its essence, WSP is a binary version of HTTP.

A **markup language** − the Wireless Markup Language (WML) has been adapted to develop optimized WAP applications. In order to save valuable bandwidth in the wireless network, WML can be encoded into a compact binary format. Encoding WML is one of the tasks performed by the WAP Gateway/Proxy.

How WAP Model Works?

When it comes to actual use, WAP works as follows −

- The user selects an option on their mobile device that has a URL with Wireless Markup language (WML) content assigned to it.

- The phone sends the URL request via the phone network to a WAP gateway using the binary encoded WAP protocol.

- The gateway translates this WAP request into a conventional HTTP request for the specified URL and sends it on to the Internet.

- The appropriate Web server picks up the HTTP request.

- The server processes the request just as it would any other request. If the URL refers to a static WML file, the server delivers it. If a CGI script is requested, it is processed and the content

returned as usual.

- The Web server adds the HTTP header to the WML content and returns it to the gateway.

- The WAP gateway compiles the WML into binary form.

- The gateway then sends the WML response back to the phone.

- The phone receives the WML via the WAP protocol.

- The micro-browser processes the WML and displays the content on the screen.
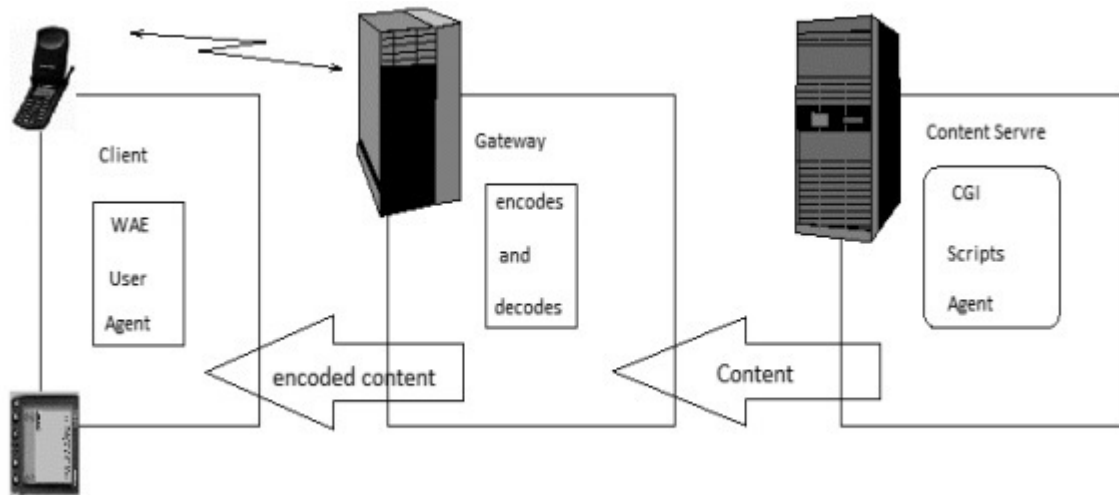
WAP - Architecture

WAP is designed in a layered fashion, so that it can be extensible, flexible, and scalable. As a result, the WAP protocol stack is divided into five layers −

Layers of WAP Protocol

Application Layer

Wireless Application Environment (WAE). This layer is of most interest to content developers because it contains among other things, device specifications, and the content development programming languages, WML, and WMLScript.

Session Layer

Wireless Session Protocol (WSP). Unlike HTTP, WSP has been designed by the WAP Forum to provide fast connection suspension and reconnection.

Transaction Layer

Wireless Transaction Protocol (WTP). The WTP runs on top of a datagram service, such as User Datagram Protocol (UDP) and is part of the standard suite of TCP/IP protocols used to provide a simplified protocol suitable for low bandwidth wireless stations.

Security Layer

Wireless Transport Layer Security (WTLS). WTLS incorporates security features that are based upon the established Transport Layer Security (TLS) protocol standard. It includes data integrity checks, privacy, service denial, and authentication services.

Transport Layer

Wireless Datagram Protocol (WDP). The WDP allows WAP to be bearer-independent by adapting the transport layer of the underlying bearer. The WDP presents a consistent data format to the higher layers of the WAP protocol stack, thereby offering the advantage of bearer independence to application developers.

Each of these layers provides a well-defined interface to the layer above it. This means that the internal workings of any layer are transparent or invisible to the layers above it. The layered architecture allows other applications and services to utilise the features provided by the WAP-stack as well. This makes it possible to use the WAP-stack for services and applications that currently are not specified by WAP.

The WAP protocol architecture is shown below alongside a typical Internet Protocol stack.
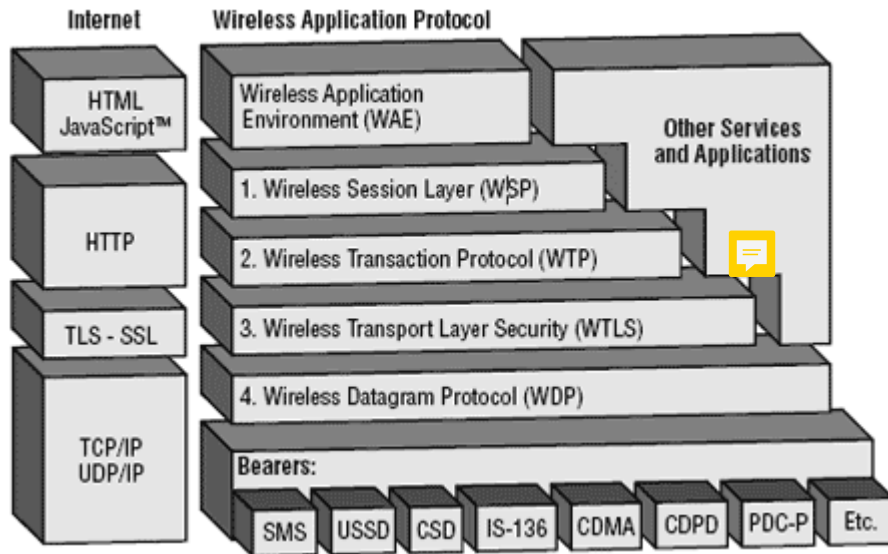
Note that the mobile network bearers in the lower part of the figure above are not part of the WAP protocol stack.

WAP - Environment

Wireless Application Environment (WAE), the uppermost layer in the WAP stack, provides an environment that enables a wide range of applications to be used on the wireless devices. We have earlier discussed about the WAP WAE programming model. In this chapter, we will focus on the various components of WAE.

Components of WAE

Addressing Model

A syntax suitable for naming resources stored on servers. WAP use the same addressing model as the one used on the Internet that is Uniform Resource Locators (URL).

Wireless Markup Language (WML)

A lightweight markup language designed to meet the constraints of a wireless environment with low bandwidth and small handheld devices. The Wireless Markup Language is WAP's analogy to HTML used on the WWW. WML is based on the Extensible Markup Language (XML).

WMLScript

A lightweight scripting language. WMLScript is based on ECMAScript, the same scripting language that JavaScript is based on. It can be used for enhancing services written in WML in the way that it to some extent adds intelligence to the services; for example, procedural logic, loops, conditional expressions, and computational functions.

Wireless Telephony Application (WTA, WTAI)

A framework and programming interface for telephony services. The Wireless Telephony Application (WTA) environment provides a means to create telephony services using WAP.

Hardware and Software Requirement

At minimum developing WAP applications requires a web server and a WAP simulator. Using simulator software while developing a WAP application is convenient as all the required software can be installed on the development PC.

Although, software simulators are good in their own right, no WAP application should go into production without testing it with actual hardware. The following list gives a quick overview of the necessary hardware and software to test and develop WAP applications −

- A web server with connection to the Internet
- A WML to develop WAP application

- A WAP simulator to test WAP application

- A WAP gateway

- A WAP phone for final testing.

Microsoft IIS or Apache on Windows or Linux can be used as the web server and Nokia WAP Toolkit version 2.0 as the WinWAP simulator.

Please have look at WAP - Useful Resources   to find out all the above components.

Configure Web Server for WAP

In the WAP architecture, the web server communicates with the WAP gateway, accepting HTTP requests and returning WML code to the gateway. The HTTP protocol mandates that each reply must include something called a Multi-Purpose Internet Mail Extensions (MIME) type.

In normal web applications, this MIME type is set to text/html, designating normal HTML code. Images on the other hand could be specified as image/gif or image/jpeg for instance. With this content type specification, the web browser knows the data type that the web server returns.

In WAP applications a new set of MIME types must be used, as shown in the following table −

File type	MIME type
WML (.wml)	text/vnd.wap.wml
WMLScript (.wmls)	text/vmd.wap.wmlscript
WBMP (.wbmp)	image/vnd.wap.wbmp

In dynamic applications, the MIME type must be set on the fly, whereas in static WAP applications, the web server must be configured appropriately.

For more information about configuring MIME types for your web server, please consult your web server documentation.