1

Overview of WAP

In 1997 the word WAP hit the headlines all over the world and everyone started looking at it as the new 'money making machine' in the telecommunications area. The arrival of WAP coincided with a period of great interest in the wireless world, both in consumer and industry markets.

WAP – the **Wireless Application Protocol** – is a communications protocol and application environment for the deployment of information resources, advanced telephony services, and Internet access from mobile devices. In this chapter, we will be examining what this statement really means. We will take a broad overview of WAP – what it is, the factors that brought about its creation, and why it is suitable for giving us the power of the Internet via mobile phones and PDAs. We will see the advantages that it has, and the limitations that will be imposed upon us as developers. We will also compare it with more traditional web development to find out what the similarities and differences are. More importantly, we will look at how WAP is structured, the types of systems that are involved and what functions they implement.

All the ideas and concepts introduced in this chapter are analyzed in more depth in later chapters of the book. In the latter part of the book, we will analyze the interaction between WAP and existing technologies, such as dynamic content generation tools and directory services. We will look at specific case studies and help you set up your own WAP solutions.

WAP and the Wireless World

In recent years, wireless telecommunications have become a common subject of technical papers. The new trend in technology is to provide users with the ability to have all they could possibly need in a device that fits in their pocket.

Smaller and smaller PDAs (Personal Data Assistants), laptop computers and mobile phones are hitting the market, incorporating brand new features designed to let the users work and access documents in whatever situation they are in. The Internet is considered with particular interest, given the fact that it is widespread and easy to access from almost anywhere in the world.

One of the latest innovations in the field – and the one that has shaken the telecommunication world from its roots – is WAP. It introduces a new way of looking at the wireless phenomenon – letting the applications 'follow' their customers and provide them with innovative services.

Mobility

Mobility is the new buzzword in the business world and over time expectations have risen about exactly what this means. In the late eighties and early nineties, *mobility* was associated with the ever-reachable salesman and his mobile phone. This concept expanded (mainly across Europe and Asia) with the advent of Global System for Mobile communications (GSM) in 1991. It is also possible to connect your laptop to a phone, whether by cable, IR port or, in the near future, the much-anticipated **Bluetooth**.

> *Bluetooth is a new technology that is designed to provide a common way to connect mobile devices, such as PDAs, laptops and mobile phones. It was developed by a consortium including Ericsson, Nokia, Intel, IBM, Toshiba, Motorola, and Palm (3Com), and its final goal is to take the place of cables and IR, providing faster connection speeds. For more information you can refer to the site* `www.bluetooth.com`*.*

Here is a definition for mobility that might work in today's business world:

> **Mobility is the ability to access information and services any time, anyhow, anywhere.**

This information might be an e-mail that my boss wrote me asking for a report, the latest sales figures for this month, or the phone number of a client I need to talk to. The services include banking applications, online shopping and checking stock quotes. What we are talking about is extending enterprise applications to incorporate the mobile client, i.e. extending the office to include any location the worker might be – at home, at a conference, traveling, and so on.

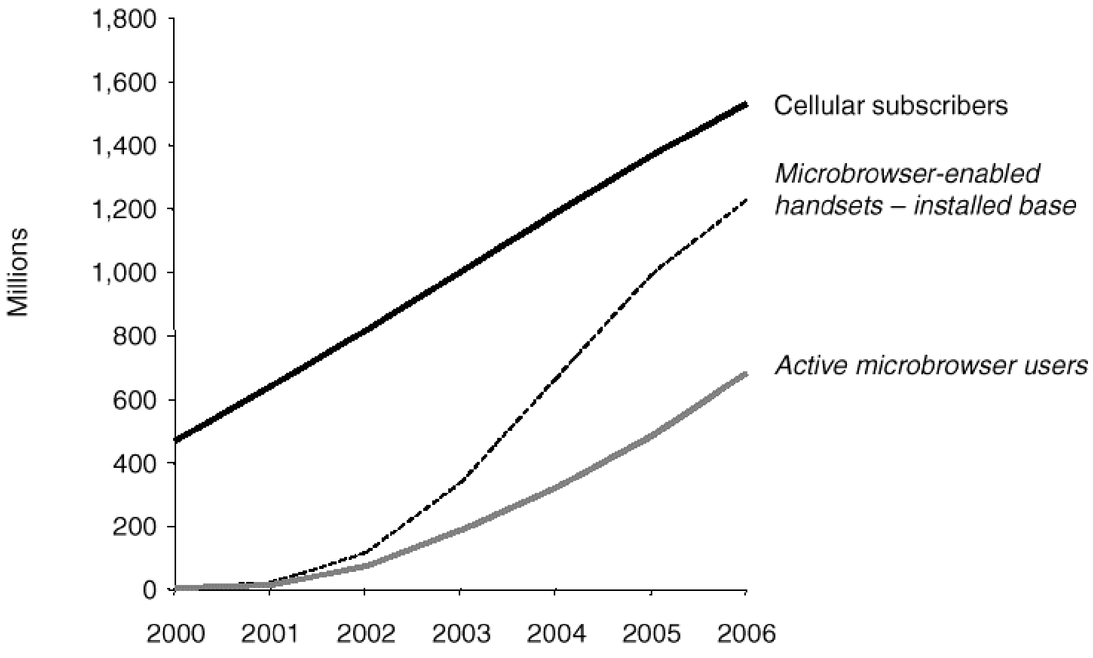
The increase in expectations of the mobile public over recent years has been driven by the rapid development of wireless technology. From mobile phones to PDAs and handheld computers, the devices being developed have become smaller, more powerful, and – as consumer demand increases – cheaper. This in turn drives the market forward. New technologies spread much faster than they did in the past, giving everyone the chance to experience new services. There is no longer a neat division between different categories of people. Technology available to businessmen is now equally available to teenagers. Although the markets for different categories of people are very different, they can all benefit from new and attractive services.

Changing the Way We Look at the Net

With the advance of the Internet, e-commerce has now grown to enormous proportions; online banking, trading and shopping have proven to be such a success that the goal of business has become the provision of services that are available from *anywhere*.

On top of this, the number of mobile phones in the world is increasing every day at an astonishing speed, with analysts forecasting that there will be more than a billion mobile phones in use within the next five years (Gartner Group) and that over half of Internet access will be through non-PCs (Meta Group). The mobile phone has become a part of daily life for many people, and together with a watch, is the only electronic device that many people carry around everywhere with them, all day long.

The graph below is from a report, *WAP Market Strategies*, from Ovum (`http://www.ovum.com`). It compares the forecasted growth of mobile phones, Internet enabled mobile phones, and Internet enabled mobile phones that are actually used to access the Internet:



Source: Ovum (WAP/A)

We have come to take for granted the availability of information, wherever we are and whatever the topic, through the computer. However a major bottleneck has been predicted for this data utopia at the beginning of this millennium. That bottleneck is the computer itself. Constant improvements have kept the price of a PC reasonably high –high enough to keep the purchase of a computer at the bottom of many people's list of priorities. Furthermore, there are still many peoples that do not feel the need for or do not feel like playing around with devices that they see as too technical. Some others may be working behind a computer all day long, and therefore refuse to sit behind one during their spare time too.

It's time that the Internet moved on from the PC. This doesn't mean the end of the PC; radio didn't kill the newspaper, TV didn't kill the radio, and the VCR didn't kill cinema – there's always room for more than one media. It simply means that there is more than one way of accessing information from the Internet, and the method you choose depends on who you are, where you are and what you want.

We now have Internet capable televisions and games consoles, and within a few years we should see the introduction of Internet enabled Hi-Fi systems. And with the advent of cheap, reliable mobile phones capable of accessing the Net, there seems to be a major opportunity for a powerful and real mobile data service.

However, the Internet – as it is now – is not well suited to the mobile phone. It is typically too complex, takes up too much bandwidth, and a web page would generally not fit onto the screen of your average mobile phone. While the third generation of wireless technology (known as 3G) should go some way to easing the bandwidth issue, there are still other problems that need to be considered, before the Internet and mobile devices can be brought together. For example, there is typically no keyboard on a mobile phone, so it is much harder for a user to enter information on their phone compared to a PC. Also, the screen is very small, and so cannot display much text at any one time and will struggle with complex graphics. It goes without saying that your average phone is nothing like as powerful as a PC!

So, if we are going to allow Internet access from a mobile phone, we first need to take into account these limitations of the client device. The Internet protocols (TCP/IP and HTTP) are far from being suitable for use with mobile phone communications. They introduce far too many overheads, requiring many messages between clients and server just to set up a connection. These overheads call for a high processing power on the client device.

Furthermore, there is a second limitation connected to the internal structure of wireless networks. This is the sustained waiting time, called **latency**. Basically, the information coming from the Internet and going to the mobile phones has to go through various elements in the mobile network, each one introducing a little delay. Also, the air interface used to transmit data to mobile telephones has a bandwidth that is very limited, nowadays reaching 9600 bit per second in a GSM network, compared to a minimum of around 28 or 56 kbps on a wired network. Thus the Internet protocols, which send many large messages, naturally result in a large latency.

These reasons motivate the need for a new set of protocols more appropriate to communication with wireless devices.

The WAP Forum – A Standard for Wireless Web Access

Back in 1995 in the US, Unwired Planet introduced **HDML** – the **Handheld Device Mark up Language** – which is a cut-down version of HTML, designed to run on wireless devices. And, in Japan, the operator NTT DoCoMo introduced a service called **i-mode** in early 1999. This has become a very popular technology, with almost 7 million users using it to access Internet services from mobile phones, which has been driven largely by the youth market.

These two technologies present us with an interesting question: what is the winning technology? Is it the one providing the best technical solution to a given problem, or is it the one that is most widely adopted? This was probably the question that was asked at Unwired Planet (now Phone.com) during 1996 and early 1997. Recall that Unwired Planet was the first company involved in the development of a new technology devised to port Internet services to wireless users. They could have kept on focusing just on the development of HDML, letting it grow in the US as NTT DoCoMo have done with I-mode in Japan. However, they chose instead to get involved the major mobile phone manufacturers in their project, reckoning that the more devices there were that supported the technology in the world market, the more they could sell their wireless Internet solutions around the world. Involving other companies, each one with a large customer base in different parts of the world, has helped to promote the newborn technology.

Thus, the **WAP Forum** was created by Phone.com, Ericsson, Nokia and Motorola. Everyone got 'infected' by the WAP virus, with network operators and device manufacturers struggling to offer the new technology to their customers, just to stay competitive. The Phone.com WAP gateway – UP.Link – is the most mounted in operator networks. Also, the Phone.com software application – UP.Browser, which allows the mobile phones that it is installed in to receive WAP data – is present in a large fraction of the WAP compliant mobile phones around the world.

With the advent of the WAP Forum, Phone.com shared its knowledge, and the partnership soon evolved into the now all-encompassing **WAP specifications** that include complementary application, session, transaction, security, and transport protocol layers. A new markup language called the Wireless Markup Language (WML) has also been created. These protocols minimize the problems outlined above associated with the use of Internet protocols for wireless data transfer. They do this by eliminating unnecessary data transfers, and using binary code to reduce the amount of data that has to be sent. Also, wireless sessions are designed to be easily suspended and resumed, without the connection overheads associated with the Internet protocols. Thus the protocols are well suited to the low bandwidth associated with wireless communications.

The WAP forum has now grown and includes over 230 members and includes carriers, handset manufacturers, software developers and other companies. Their mandate includes ensuring product interoperability and maintaining growth of the wireless market. With 90% of the handset market now being represented at the WAP Forum, along with many software companies and network operators, WAP will be the primary way of accessing the Internet.

The standardization of methods to access the Internet from mobile phones has brought many benefits to many different people. For the end users, a breadth of choice of devices, networks and applications has arisen in a competitive market, since the specifications are not biased towards any one company. The network operators have been able to extend their customer base due to the new services on offer, which are independent of the network used. For the service providers there are new functionalities, such as push technology and WTA (Wireless Telephony Applications) which are discussed later, that could increase their revenue making potential. And for the device manufacturers, there is the opportunity to devise new innovative products in a wide open market, without a huge increase in expense since WAP tries to keep the necessary processing power to a minimum.

At the time of writing, the WAP Forum is working on version 1.3 of the WAP specifications, and version 1.2 is being implemented in the wireless networks. However, at present mobile phones still only support version 1.1 of the WAP specifications. This book will focus on the aspects of the 1.1 specification that are implemented today, but we will take a quick look at some of the forthcoming features that will be important in the future.

The Business Perspective

As the new standard protocol for providing content to wireless devices, WAP has been accepted on the telecommunications market with enthusiasm from all the sides, as the growth in the stock market of some of the companies involved with WAP can confirm. The high penetration rates for mobile phones across Europe, Japan and other parts of Asia – and increasingly in the US – mean that mobile commerce has become so significant it has even given birth to new jargon terms, such as **m-commerce** and **m-business**. Many businesses were caught out by the speed of change on the Internet and the rapid rise of e-commerce, and so have jumped quickly to supply WAP services in an effort not to be left behind. There are already plans in place for **mobile advertising** – advertising content aimed at mobile devices – to finance the investments made. This is somewhat hampered by the difficulties in profiling customers.

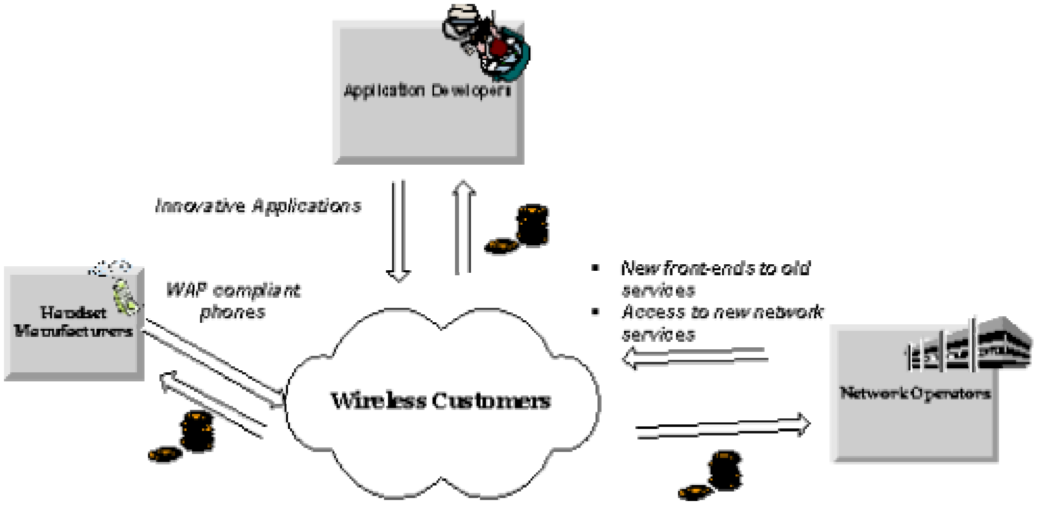
A new service, made available with WAP, is the **Location Information Service**. This can supply the position of a network subscriber to the WAP applications that use it. It is made possible by the many antennas network operators have distributed around the country to communicate with the mobile phones. The network operator always knows which antenna is receiving the signal from a given mobile phone, and of course the operator also knows where each antenna is located.

Location services provide a way of delivering location dependent information and advertising to subscribers.

For example, it will allow us to find out where the nearest bank or the nearest travel agency is. These services, even though included in the WAP 1.1 specifications, are not yet implemented in the operator networks. When available they will provide many benefits.

Together with the push technology that is part of the WAP 1.2 specifications, this feature will give the developers the ability to design marketing applications that provide advertising information to subscribers when they are close to a shop or a bank or a cinema, for example. Push technology, which is discussed in more detail later on in this chapter, can be used to send information to subscribers without them sending any request for content.

We are now ready to analyze where the money comes from in the WAP world; some of the possibilities are shown in the diagram below:



The additional data traffic in the network is the first real source of revenue, at least in the beginning when m-commerce and advertising are yet to become mainstream. The network operators are evidently the parties gaining the most advantage from this, since the more new users that the content providers attract to their WAP applications, the more the traffic will grow in the network. The other direct sources of income for network operators are the subscriptions from WAP-interested customers.

One interesting feature offered by network operators, but also sometimes by content providers, is a **WAP portal**. A WAP portal is similar to a web portal: it is a page containing many links to interesting applications, grouped by category. Since the network operator's portal will, in most cases, be the first page that a subscriber sees on their device when connecting, it is clear that the applications listed there will have an enormous advantage over those that are not.

Network operators will charge content providers and advertisers for the display of their links on the operator's portal. It is also probable that the operators will try to form partnerships with content providers who supply new and interesting services. In this way, they will monopolize their content, thus differentiating themselves from other operators, satisfying their own subscribers and attracting subscribers from other operators.

Application developers play an important role in the newborn WAP industry. In addition to providing **Value Added Services**, there is a strong demand for services that are available on the Web to be ported to WAP.

One of the major advantages to WAP is that it's markup language, WML, is based on XML. This effectively means that it should be easier to provide content in a device independent way. In practice, this is not always so, but this is a topic we'll come to time and again throughout the book. It is discussed in detail in Chapter 9.

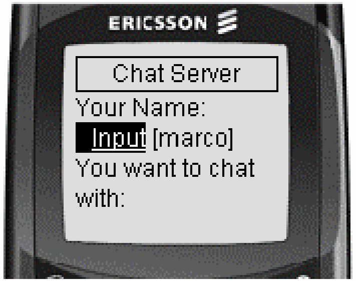
Last but not least, content providers are the ones closing the WAP circle. As we have seen they will be forced to pay to be listed on the operator's portal. They will also probably pay more than one operator, so that the service that they provide can reach a wider audience. If their service is attractive enough, they may try to share the revenues that the operators make due to the traffic that their WAP application generates on the wireless network. Incomes will also be generated for the content providers if they decide to tax the subscriber to access their content or if they include advertisements on their pages.

What is WAP Able to Do?

As with all new technologies, the expectations of WAP were very high when it was first introduced. Before the first WAP phones were available, everyone was expecting to surf the Internet from their phones just like from a normal browser, with pixel-perfect content, images and sound. The reality of course is quite different…

WAP is intended to provide a common application environment for mobile devices and its protocols *are* based on the Internet protocols. However, this does *not* mean that WAP was devised with the intent of porting the entire content of the Internet to mobile devices.

The average HTML page is now very elaborate, filled with multimedia content, frames, colors, and dynamic effects. Such pages would lose all their appeal if translated into WAP pages and presented on a display of, for example, 5 lines of 20 characters, like the one shown in the screenshot here:



Chat Server
Your Name:
Input [marco]
You want to chat
with:

Even though there are now some products available for translating HTML pages into WAP ones (and we'll discuss these in Chapter 13), the results are less than convincing. Today, WAP applications are almost all primarily developed for WAP users, bearing in mind the limitations of mobile phones. This will probably continue to be the trend for the future.

Typical applications available over WAP today include trading applications, home banking applications, shopping applications and e-mail interfaces. Many sites offer news, radio and TV listings, and some of them will help you find a restaurant or a list of cinemas for the city you're traveling to. These are a few examples of URLs to current WAP sites:

Italian Giroscopio hotel booking: `http://www.giroscopio.com/wap`
UK Entertainment Centre: `http://www.ents24.com/index.wml`
WAP portal: `http://wap.waportal.com`
WebCab.de: `http://webcab.de/i.wml`

ElectricNews.Net - tech news via WAP: `http://adlib.ie/enn/waphome.wml`

With limited or non-existent graphics, no multi-media, no complex user interaction, and no colors, what is left for us in WAP applications?

If you are willing to take a chance on building a successful WAP application, you will have to start thinking about the peculiarities of a mobile device and how people could take advantage of them. People are always on the move, so Location Information Services will have a major impact on the market. Push technology will probably supply a way to try to predict what people feel like doing and to suggest where and when they can do it without them requesting anything from your site.

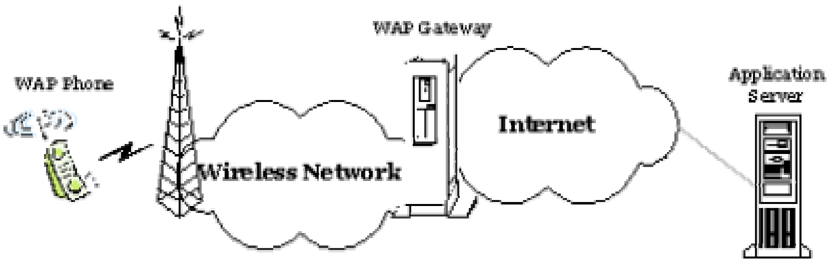
WAP Application Architecture

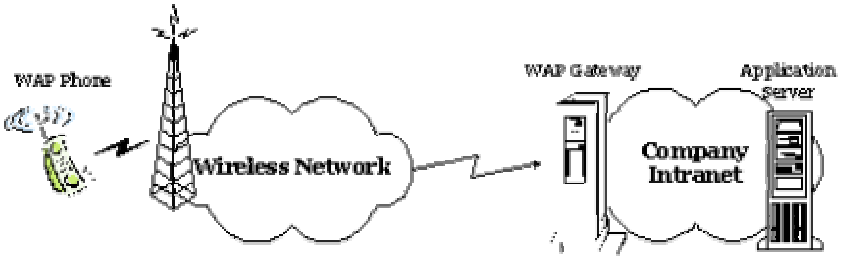
So far we have looked at what WAP is, and the reasons why it has been introduced. We will now embark on a more in depth description of its technical details.

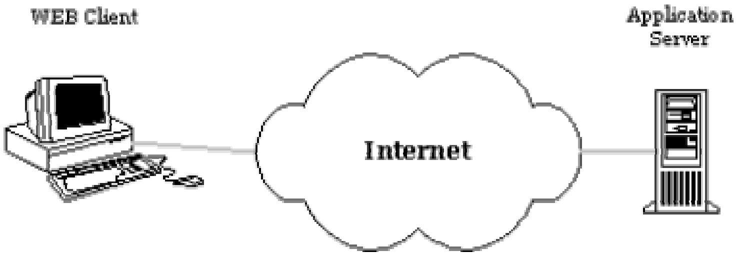
The WAP protocols were designed with the web protocols in mind. The goal of WAP was to use the underlying web structure, but to render communication between content providers and mobile devices more efficient and less time consuming than if the web protocols themselves were used. In this section we will start introducing the elements involved in mobile communications, and their role in the whole picture.

Let's start by comparing the different ways you can access information from the Internet using a WAP device. In the diagram below the main differencesbetween (a) WAP used to access the Internet, (b) WAP used to access an intranet, and (c) the Internet architectures are shown.

Since the WAP architecture has been designed to closely follow that of the Web, the client-server paradigm used by the Internet has been inherited by WAP. The main difference, however, is the presence of the WAP gateway for translating between HTTP and WAP.







c)

You will notice that the above diagrams introduce some new terms, as well as some that may already be familiar to you. Before continuing, we will take a closer look at these technical terms, which will be used in the rest of this chapter and, indeed, in the rest of the book.

- **WAP device**: This term indicates the physical device that you use to access WAP applications and content. It doesn't necessarily have to be a mobile phone – it might be a PDA or a handheld computer. More generally, it's every WAP compliant device.
- **WAP Client**: In a network environment, a client is typically the logical entity that is operated by the user and communicates with the 'server entity'. In the WAP world, the client is the entity that receives content from the Internet via a WAP Gateway. This is usually (but not necessarily) the WAP browser. Commonly, 'WAP client' and 'WAP browser' are often used interchangeably.
- **WAP Browser**: This is software running on the WAP device that interprets the WAP content arriving from the Internet and decides how to display it on the screen of the WAP device. WAP browsers are available for all WAP devices, and are frequently referred to as **microbrowsers**. There are also emulators available for some browsers, which run on PCs.
- **User Agent**: An agent is normally the software that deals with protocols, and WAP is no exception to this. The WAP client contains two different agents: the WAE User Agent and the WTA User Agent (each of which will be covered later in the chapter).
- **WAP Gateway**: This is the element that sits (logically) between the WAP device and the origin server. It acts as an 'interpreter' between the two, enabling them to communicate. It usually resides within the operator network, but you can also install your own gateway, as we will see later. Unless otherwise stated, when a gateway is discussed, we mean a gateway residing in the operator network, since this is the more common situation that one encounters.
- **Network Operator**: This is the company or organization that provides carrier services to its subscribers. As an example, the company you are paying your telephone bills to is your network operator. A network operator enables you to make calls to other phones from your telephone and, in addition, provides you with different services, such as voice mail, call diversion etc.
- **Bearer services**: These are the different ways that a mobile phone can communicate with the wireless network. To send and receive data from an application server, mobile phones have to establish some sort of connection with the WAP gateway. A bearer service is the method they use to do this. In GSM networks, for example, we either use SMS (Short Message Service) or CSD (Circuit Switched Data). With the former bearer, the gateway has to divide the information that is to be sent to the phone into a lot of little messages (just like when you send a text message to a friend using your mobile). With CSD, we communicate with the gateway using a data connection,

which is not dissimilar to the way the modem in your computer communicates with the Internet Service Provider that you have an account with.

- **Content/Origin/Application Server**: Thesethree names are used throughout the book interchangeably. They denote the element that hosts the Internet content that is sent to clients when they make a request for it. A web server is an origin server, providing HTML content (but also WAP content if properly configured).

As you can see from the diagrams above, the WAP architecture resembles closely that of the Internet.

To access an application stored on the server, the client initiates a connection with the WAP gateway, and sends a request for content. The gateway converts the requests coming from the WAP client into the format used over the Internet (HTTP), and then forwards them to the origin server. On the way back, the content is sent from the server to the gateway, which then translates it to WAP format, and then sends it to the mobile device. The gateway allows the Internet to talk to the wireless network.

The concept of connection is left deliberately vague, since the goal of WAP is to provide a protocol that is able to adapt to any type of mobile network. The connection is established between the WAP phone and the gateway by means of the bearer used. Whether we are accessing WAP services by sending data packets or SMS messages, we see the same functionality. It may, however, affect the speed of the connection and therefore affect the cost of the connection, but this is less important to the developer.

As is the case with the Internet, content servers host the content or applications, but in the case of WAP these are sent to the clients as **WML** and **WMLScript** files, rather than HTML etc. WML (Wireless Markup Language) and WMLScript are the languages used to design and write WAP content. WML has some similarities to HTML and XML, and WMLScript doesn't differ much from JavaScript. They are described later in this chapter, and in more detail in Chapters 4, 5 and 6.

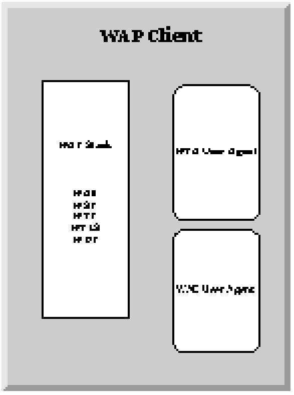
The WML and WMLScript files are sent, on request, to the WAP client via a WAP gateway, which translates the content into a form that is optimized for the narrow bandwidth radio interface. The clients contain a microbrowser that displays the received information to the user.

We'll now look at a few of the elements of the WAP architecture in a bit more detail.

WAP Client

The WAP specifications leave a great deal of autonomy to the device manufacturers. There is no WAP specification indicating what the WAP device should look like or how it should present and display the content it receives from the Internet. These kinds of decisions, together with those relating to the user interface and the internal organization of phone functionality such as the phonebook, are left to the vendor.

The only requirement for a device to be WAP compliant is that it must implement a **WAE User agent**, a **WTA User Agent** and the **WAP Stack**.



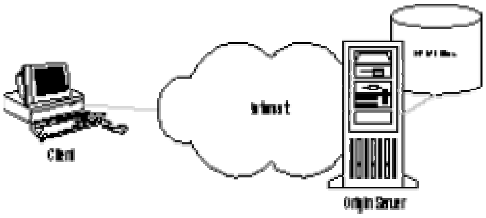
- The **WAE User Agent** (Wireless Application Environment User Agent) is the microbrowser that renders the content for display. It receives the compiled WML, WMLScript, and any images from the WAP gateway, and executes or displays them on the screen. Even if the implementation details are left to the vendor, the browser must implement all the functionality provided by WML and WMLScript. It must also manage the interaction with the user, such as text input, and error or warning messages.
- The **WTA User Agent** (Wireless Telephony Applications User Agent) receives compiled WTA files from the WTA server and executes them. The WTA User Agent includes access to the interface to the phone, and network functionality such as number dialing, call answering, phonebook organization, message management and location indication services, which we discussed earlier.
- The **WAP Stack** implementation allows the phone to connect to the WAP gateway using the WAP protocols. We'll be looking at all the WAP protocols in detail later in this chapter.

WAP Proxy, WAP Gateway or WAP Server?

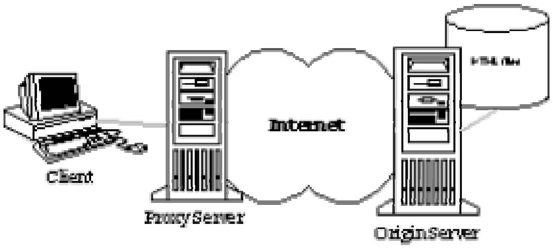
When you read articles, surf the Internet or attend conferences, you will certainly hear about WAP *gateways*, *servers* and *proxies*. These three terms are often used interchangeably and wrongly so. On the contrary, in the world of networks these three elements are quite different logically and they have different functionalities as well:

- **Content/Origin/application Server**: This is the element in the network where the information or web/WAP applications reside. (Web servers belong to this category.)
- **Proxy**: This is an intermediary element, acting both as a client and as a server in the network. It is located between clients and origin servers; the clients send requests to it and it retrieves and caches the information needed by contacting the origin servers.
- **Gateway**: This is an intermediary element usually used to connect two different types of network. It receives requests directly from the clients as if it actually were the origin server that the clients want to retrieve the information from. The clients are usually not aware that they are speaking to the gateway.

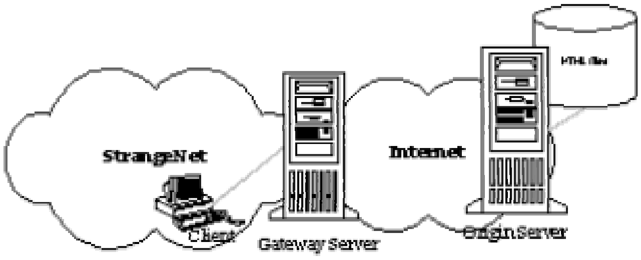
These three terms are illustrated in the diagram below where (a) and origin server has a direct connection to the Internet, (b) access to the Internet is through a proxy server, and (c) a gateway server lies between two different types of networks.



(a)



(b)



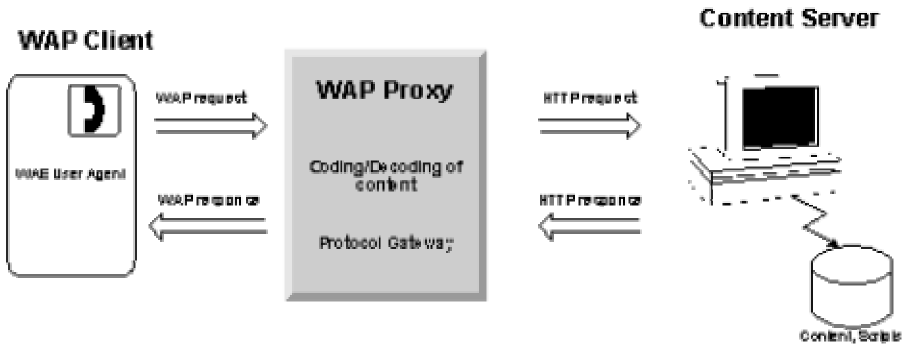
(c)

The element used in the WAP architecture, which we earlier defined as (and is commonly called) a WAP gateway, is actually a **proxy**. It is used to connect the wireless domain with the Internet one. However, it contains *protocol gateway* functionality plus *encoder/decoder* functionality.

The products at present on the market create the confusion of terms. What you are typically offered today when you search for such a WAP element is a mixture of all of the servers described above. It logically belongs to the proxy category but, as we have seen, has gateway functionality and in addition is equipped with server functionality. In other words it can run server-side scripts, Java servlets and do all the things that a standard web server can do.

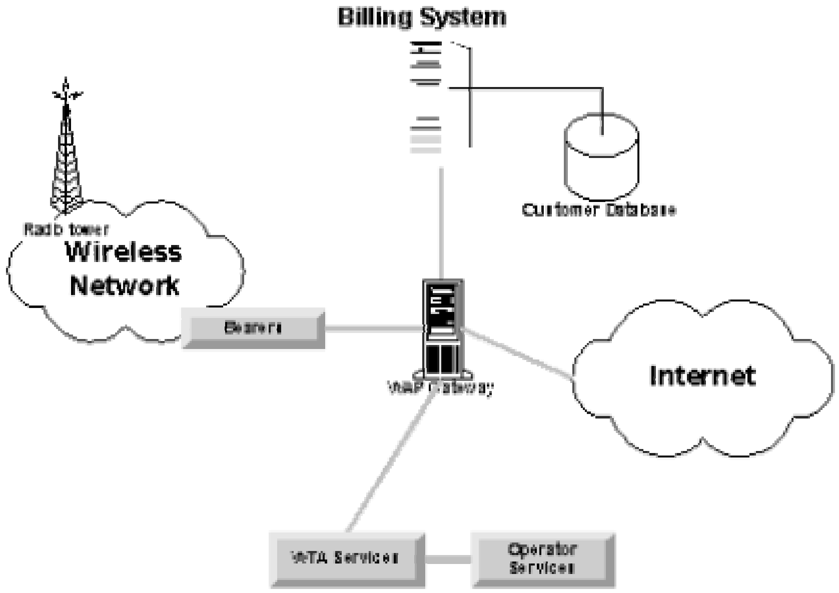
The rule to survive this confusion is generally to consider a WAP gateway and a WAP proxy as the same thing; we will try to consistently use the term 'gateway' throughout this book. Also, it's a good idea to avoid the term 'WAP server'. WAP servers are usually a WAP gateway with server functionality added. It is probably better to refer to such an element as a "combined application server and gateway".

In the diagram below we illustrate the use of a WAP proxy/gateway. We will move on to consider exactly what the gateway does in the next section.



WAP Gateway Functionality

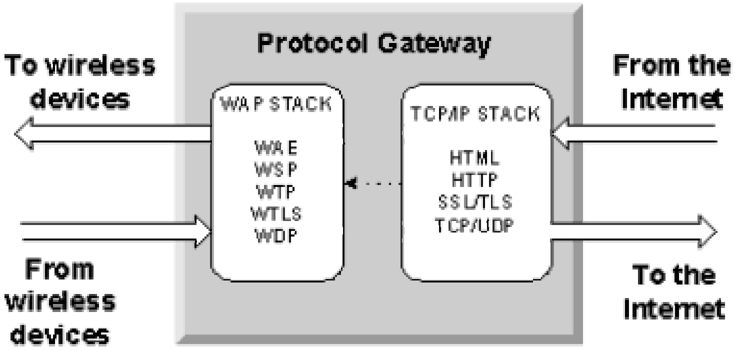
In the diagram below, a WAP gateway is shown, together with other elements in the wireless network. This highlights how the WAP Gateway has to collaborate and interface with all the other elements in order to provide a proper service:



Whenever you start a WAP session on your mobile phone, the following steps are executed. (The details of the WAP protocols are dealt with in a later section.)

- ❑ A connection is created via WSP (Wireless Session Protocol) between the mobile device and the WAP gateway, which we assume is present in the operator network.
- ❑ As you enter the address of a WAP site (by typing it or selecting a bookmark, for example), the gateway is sent a request from the device's microbrowser using WSP. WSP is the WAP protocol in charge of starting and ending the connections from the mobile devices to the WAP gateway. We will discuss it in more detail in a later section.
- ❑ The gateway translates the WSP request into an HTTP request and sends it to the appropriate origin server.
- ❑ The origin server sends back the requested information to the gateway via HTTP.
- ❑ The gateway translates and compresses the information and sends it back to the microbrowser in the mobile device.

The gateway part of the WAP proxy takes care of translating all the requests that are sent and received by the client using WSP to the protocol that the origin server is using (HTTP for example). This is illustrated in the diagram below. The content provider sends its content using HTTP to the gateway. It then forwards all the content received to the WAP devices, using the WAP protocols.

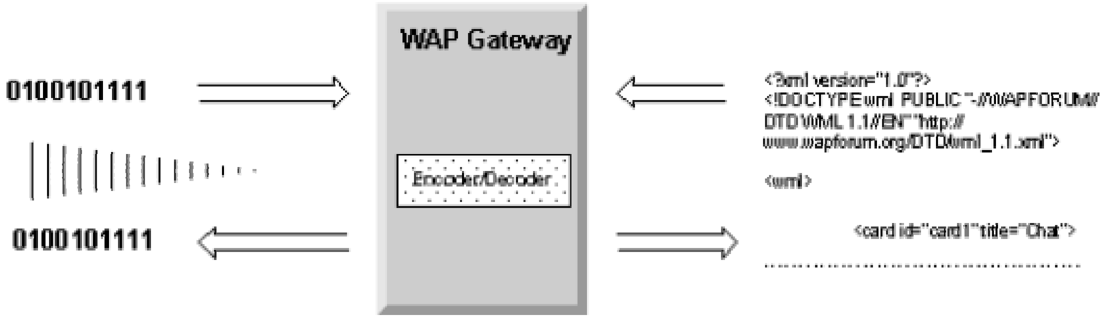


Functionally speaking, the gateway operates to some extent in a similar way to the current Internet web browsers. When you try to access an FTP or Gopher site using your web browser, you are completely shielded from the protocols and requests that your browser uses to contact the site. As far as you are concerned, both FTP and Gopher sites use the same protocol to communicate with the browser as a normal web site, since the information that is displayed on your screen is in the same format as when you access an HTML page.

The coder/decoder (CODEC) functionality within the gateway is used to convert the WML and WMLScript content going to and coming from the client into a form that is optimized for low-bandwidth networks. This is illustrated in the diagram below.

Translation of encrypted data takes place in the memory of the gateway. No unencrypted data is ever stored on a secondary storage medium since this would create crucial security problems. This idea is discussed in more detail in Chapter 3 (on gateways) and Chapter 16 (on WAP security).

Content belonging to a non-secure session is cached on the storage media of the gateway, reducing the processing time and the resources required when someone else requests the same content.


WAP Gateway
0100101111
Encoder/Decoder
0100101111
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="card1" title="Chat">

Another service that the CODEC functionality can provide is the translation of HTML or text to WML. However, this use of the gateway should be considered very carefully, since many limitations apply. HTML and WML are both based on tags and they can look very similar to each other – we can be fooled into thinking that translating HTML into WML is the easiest way of accessing Internet content from a wireless device. However, HTML has now grown into a fully-fledged language, allowing dynamic content and multimedia to be displayed. Many Internet sites around use frames, and take advantage of all the features of HTML and the use of multimedia such as sound, video and graphics. The result of the translation of such an HTML page into a WML one will therefore be quite poor. (Methods of translating existing content, with all the advantages and disadvantages this brings, and the situations in which translation is suitable are discussed in detail in Chapter 13.)

The HTML to WML translator, when present among the gateway features, is there mainly for giving us compatibility and to reassure us that if we really need something important that is stored in HTML format then we have a way to retrieve it.

The WAP gateway needs, of course, to have more functionality than just that listed above. Since it is usually the operator network element that is contacted by customers to access a service, it also has to include charging functionality. It can be connected to a billing system and a customer database for this purpose.

It also implements an interface for each of the bearers present in the wireless network of the operator. For example, if we install a gateway in a GSM network, it must implement an SMS (Short Message Service) and a CSD (Circuit Switched Data) interface.

The WAP gateway is also connected to the WTA server present in the operator network that provides the interface for accessing some of the network services the operator wants to provide.

Who Needs a Gateway?

So should you install your own gateway? There aren't too many reasons to do this, since the wireless network operator always provides the gateway. Furthermore, WAP gateways are designed for installation and use in an operator network and their use in another environment generates some difficulties, for example the adjustment of the gateway for the different bearers and handsets.

If you are involved in the design of an application that involves a high level of security, such as Intranet directory services, or involving the exchange of critical private data, you may be concerned with the security risks of a wireless connection. You will not want to be left with an insecure information exchange all the way between your hosted content and the mobile devices.

Instead, many companies wish to install their own gateway, ensuring that their content can be sent securely to the mobile phones authorized to access it, avoiding completely the Internet side of WAP. These issues will be discussed further in Chapter 3, but also see Chapter 16 for other security issues.

The installation of a fully-fledged WAP gateway will take time, effort and usually cost a lot of money, so before undertaking this action, make sure there are no alternatives. When installing your own gateway, you will also have to choose whether to restrict the number of customers that can access your service, or to install an interface for each and every type of bearer available on the wireless network market.

Another problem with setting up your own gateway is that customers wishing to use it will need to change the way their phone is configured completely, and then change it back if they are to use the original gateway provided with their phone. This will mean changing the IP address of the gateway, phone number and possibly the user name and password. One or two phones provide multiple settings. However, this is the exception rather than the rule.

If you work for an operator or if you have to implement an Intranet solution (and this last possibility is one of the only situations where you will really need your own gateway), then you can consider buying one of the software packages listed below. You will find more details on gateways in Chapter 3, but here is a brief summary of the current products available:

- **Ericsson Jambala**: product aimed at the TDMA network
- **Ericsson WAP gateway/proxy**: gateway package that will fit the GSM network
- **Nokia WAP Server**: gateway package that will fit the GSM network
- **Motorola Exchange(MIX)**: scalable gateway package that will adapt to different types of networks.
- **Phone.com UP.Link**: gateway package that fits diverse types of network, including CDMA, CDPD, GSM, iDEN, PDC, PHS and TDMA.

WAP Application Server

While on the Internet a web server is the content provider – a computer hosting the information we wish to share with the rest of the world. When considering WAP content/application servers you will see that the features that a server provides can vary greatly, depending on who you are speaking to. Out of the confusion come two definitions, which are given below:

- The WAP application/origin/content server has the exact same function as a web server and offers the same features to clients. The distinction between them is only a logical one, since the two can coexist on the same physical device, and some servers can provide both functions using the same piece of software. The only difference lies, of course, in the content that they store and send back to the clients. While the web server supports files such as HTML, JavaScript, multimedia, and all types of images, the WAP application server stores WML, WMLScript and WBMP (Wireless Bitmap) image files.
- A WAP server is usually just a WAP application server with gateway functionality added. It will provide all the services a normal origin server provides, but it will also act as a WAP gateway.

The WAP application server may, of course, also host all the technologies used to provide dynamic content. As you will see later in the book you can use XML in conjunction with XSLT, ASP, Perl scripts and Java servlets to name just a few, to dynamically generate WML content in the same way that you use them to generate HTML content on a web server. Chapters 8 to 12 will be focusing on this subject.

In order to enable a web server to host WAP applications, you merely need to add the MIME types for WAP files in the configuration settings of the server. (We will see how we can do this in the next chapter.) MIME (Multipurpose Internet Mail Extensions) is a method used to convert and transmit files over the Internet. When transmitting the files, the server attaches a header to the file defining the type of data contained in the files. The receiving client then knows what the file type is and can deal with it appropriately. Most WAP browsers accept only WAP MIME types, and sending a file with the wrong type in the header will generate an error.

WAP Internal Structure

Before we look at the details of how the WAP protocols are structured, let us first briefly examine the definitions of a **protocol** and a **layer**.

Protocols

As anyone who has done any international traveling knows, it is quite important when you travel to adapt your clothing and behavior for the place you are in. It is also important to speak a common language that allows others to understand what you are saying. The same problem arises with telecommunication networks: there are many different devices, networks and to allow them to communicate with each other, you must provide them with a common language. **Protocols** are the answer to this problem. There are a lot of different kinds, from very simple ones, to very elaborate ones, but they all have the same property in common: they allow computers to communicate with each other.

A protocol defines the type and the structure of messages that two devices have to use when they are communicating with each other.

This is what the Internet is all about: a set of common protocols (including HTTP) to let everyone speak with anyone else on the Net.

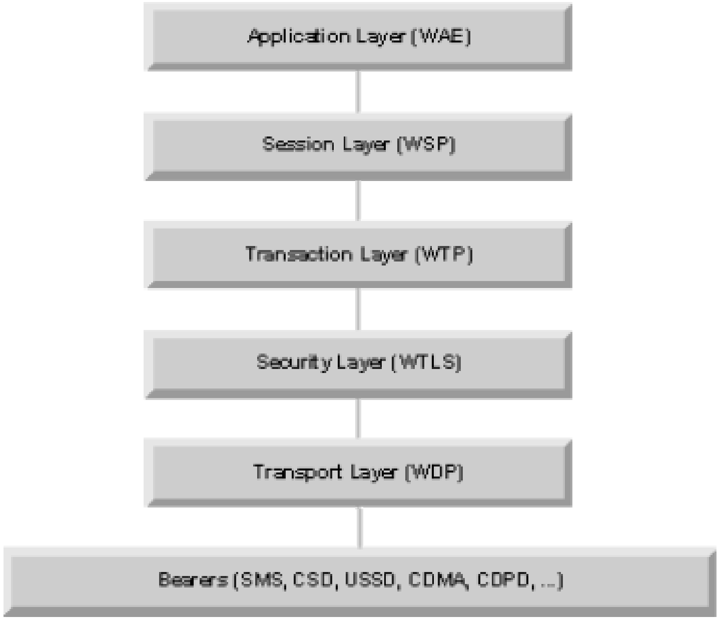
Layers

Since the protocols are functionally and logically divided into different groups of functionality, they are also physically framed into layers, each one providing a specific service to the next layer. One layer may provide methods to send bits down a physical cable; another may supply methods to establish a connection. The **protocol stack** is the set of all the layers that compose the set of protocols.

WAP Protocol Stack

In the next few pages, we will look at how the WAP protocol is structured and how the different WAP layers map into Internet protocol layers.

If your aim is to design WAP applications, you don't need to know very much about the WAP stack. The only two sections that have some relevance for developers are the ones dealing with WML and WMLScript. However, we provide details of all the WAP protocols here for completeness and for those of you who are curious about them.

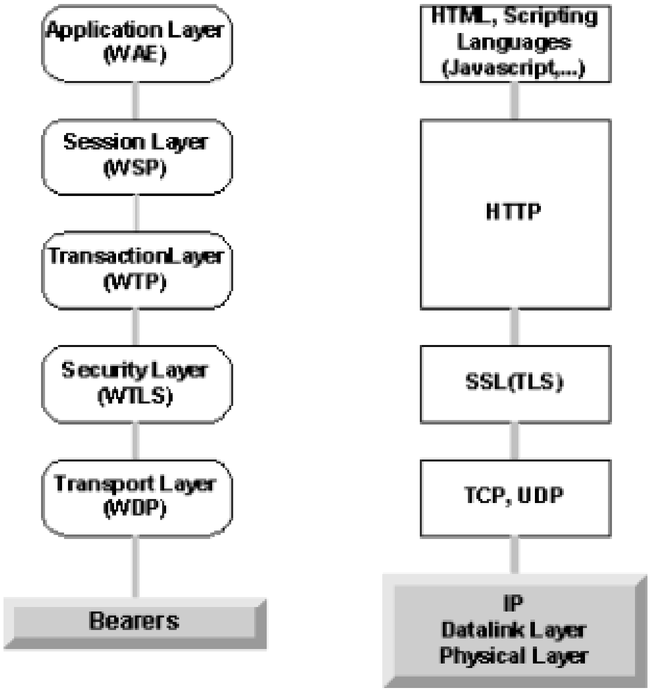


The WAP stack, illustrated in the above diagram, has 5 different layers:

- **Application Layer**: WAE (Wireless Application Environment) provides an application environment intended for the development and execution of portable applications and services.
- **Session Layer**: WSP (Wireless Session Protocol) supplies methods for the organized exchange of content between client/server applications.
- **Transaction Layer**: WTP (Wireless Transaction Protocol) provides different methods for performing transactions, to a varying degree of reliability.
- **Security Layer**: WTLS (Wireless Transport Security Layer) is an optional layer that provides, when present, authentication, privacy and secure connections between applications.
- **Transport Layer**: WDP (Wireless Datagram Protocol) is the bottom layer of the WAP stack, which shelters the upper layers from the bearer services offered by the operator.

The WAP stack was derived from, and inherited most of the characteristics of, the ISO OSI reference model [ISO7498]. The main difference between the two is the number of layers: WAP has just five layers, while the OSI model has seven of them.

For those of you not familiar with the OSI model, it is maybe useful to compare the WAP layers to the web protocol stack used in the web model. There are of course strong similarities between the two; the main differences being the compactness and lightness of the WAP protocols. As we saw earlier, the Internet protocols introduce high overheads and are not effective when used in the low bandwidth and high latency network such as the one used with mobile phones.



Both of the application layers, WAE and that of the web stack, provide a markup language and a scripting language for the development of applications. The Session Layer and the Transaction Layer in the web model are merged into the same layer, HTTP, while they are two separate entities in the WAP stack (WSP and WTP). The WAP Transport (WDP) and Security Layers (WTLS) map directly to the web TCP (or UDP)and TLS (or SSL) layers, respectively.

We'll now consider each of the WAP stack layers in turn.

Wireless Application Environment (WAE)

The application layer of WAP provides an environment that includes all the elements related to the development and execution of applications. The Wireless Application Environment (WAE) allows the developer to use specific formats and services, created and optimized for presenting content and interacting with limited capability devices. WAE consists of two different user agents located on the client side, the WAE user agent – including the microbrowser and the text message editor – and the WTA user agent. (WTA is discussed later on in this chapter, and also in more detail in Chapter 18.)

The WAE specifications say nothing about the implementation of the user agents. All the browsers, message editors, and phonebooks contained in WAP devices can vary greatly while still complying with the specifications. WAE formally specifies just the formats, such as images and text formats, that the user agents have to be compliant with. This is an important characteristic of WAP in general, as we will soon see when looking at WML.

Beginning with WAP 1.2, there will also be another scenario: it will be possible to push content towards a WAP client without any request made by the client. This will be covered later in this chapter, and also in more detail in Chapter 17.

The main building blocks of the WAE are the following:

- A lightweight markup language: WML
- A lightweight scripting language: WMLScript
- An interface to local services and advanced telephony services: WTA (not yet implemented)

We will look at WML and WMLScript in the next two sections, but these topics are covered in more detail in Chapters 4, 5 and 6.

Wireless Markup Language – WML

In the early days of the Internet, HTML was created with the intention of specifying the content to be displayed, leaving decisions as to *how* to display the content to the browsers. However, nowadays what is encapsulated in an HTML page is much more than the content. Layers, pictures, and special effects leave very little to the creativity of the browser.

With WML, WAP takes a step backwards in time to the old system. As application designers, we do not know to whom we are talking to, how big the screen of the client is, or how many keys the keyboard has. We simply know that a screen is available, and we assume that it is tiny. That's all. Keeping this in mind, we have to forget the beautiful images provided by our favorite web sites, the astonishing dynamic effects, the sounds and all the other fancy things that can be found when you browse the web.

WML has been designed to display mainly text-based pages. It is tag-based, shares elements of HTML4 and HDML2, and is defined as an XML document type. Each WML document is a single **deck**, which is made up of one or more **cards**. When the user accesses a WAP site, it sends back a deck; the user is shown the first card, reads the content, possibly can enter some information, and then moves to another card, the choice of which is dependent on the user's actions. The way in which the card is displayed is left to the browser; for example, different browsers will prompt the user for input in different ways. The browser decides how to best present the content depending on the device capabilities.

Although WML has limited capabilities when compared to HTML, it has nevertheless a wide range of features:

- **Support for Text**: When including text in a card, the programmer can use emphasis elements (such as **bold**, *Italics*, underlined, etc…), line breaks and tables. You should remember, however, that the features each browser implements may vary, and some do not support tables.
- **Support for Images**: A new format has been created for displaying images, called **WBMP** (Wireless Bitmap). Images compliant with this new standard are currently black and white. However, some browsers do not support images.
- **User Input**: Cards can contain input elements. The browser decodes input tags and then decides the best way to prompt the user for the input requested. WML specifies tags for allowing the user to submit text entries, choose among a list of options, and start a navigation or history management task (such as going to the previous card or jumping to a specified link).
- **Variables:** Variables can be included in the WML code, to keep track of hidden information and to manipulate user input.

- **Navigation and History stack**: Common navigation and history functionalities are included.
- **International Support**: The WML character set is UNICODE, which uses 16 bits to represent each character.
- **Optimization for narrow-band:** WML has been designed to adapt to the high-latency and narrow-band characteristics of wireless networks. The specifications say that connections with the origin server should be avoided unless absolutely necessary. This is accomplished by means of various technologies: variables that last longer than a single deck, cards grouped in decks, and client-side user input validation via WMLScript.

The following is an example of what WML code looks like (it is a simple 'Hello World' type example). For now we will not assume your understanding of it, as we will be studying WML in detail in Chapters 4 and 5.

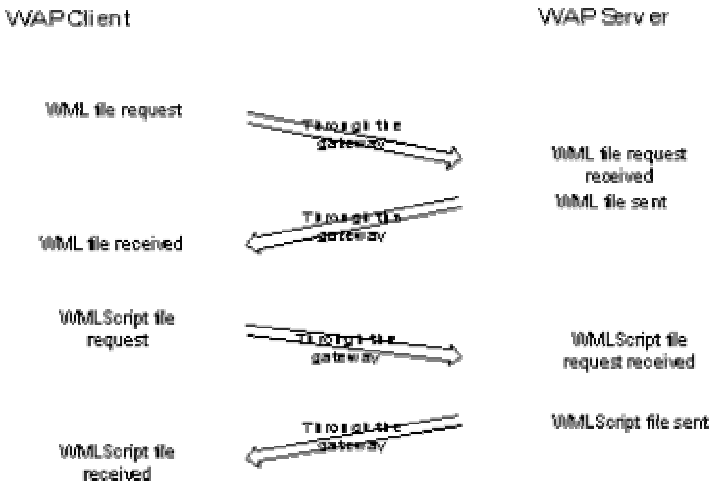
```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>
   <card id="card1" title="Hello World">
      <p>
         Hello WAP World!
      </p>
   </card>
</wml>
```

WMLScript

WMLScript is a lightweight procedural scripting language, which is based on ECMAScript, the standardized version of JavaScript. It adds intelligence to the client, providing a set of libraries for mathematical operations, string manipulation, etc., collaborates with WML, and helps prevent unnecessary connections with the server. In particular WMLScript supplies the programmer with:

- The capability of checking and validating the input submitted by the user before it is sent to the server, thus preventing the transmission of invalid data to the server.
- Access to the device facilities, such as the phonebook, calendar and list of messages (WTA)
- Methods of interacting with the user without the help of the content server, such as methods for displaying error and warning messages.

WMLScript files are separate to the WML decks from which they are called, unlike HTML where script can be embedded. Even if WML cards contain a link to WMLScript files, these are not sent to the client with the WML files, as happens with HTML. Currently, WMLScript files are sent to the WAP client only when the client tries explicitly to access functionality (typically a function) contained in one of them.



Wireless Session Layer – WSP

The Wireless Session Protocol enables services to exchange data between applications in an organized way. It includes two different protocols:

- **Connection oriented session services** – operates over the Wireless Transaction Protocol (WTP)
- **Connectionless session services** – operates directly over the Wireless Transport layer (WDP).

Session services are those functionalities that help to set up a connection between a client and a server. A service is delivered through the use of the primitives it provides. **Primitives** are defined messages that a client sends to the server to request a service facility. In WSP, for example, one of the primitives is `S-Connect`, with which we can request the creation of a connection with the server.

The **connection-oriented** session service provides facilities used to manage a session and to transmit reliable data between a client and a server. The session created can then be suspended and resumed later if the transmission of data becomes impossible. Also, once the push technology takes off, unsolicited data can be pushed from the server to the client in a confirmed or unconfirmed way. In **confirmed push** the server is notified upon reception of the data by the client, in **unconfirmed push** the server is not notified of the reception of the pushed data. Most of the facilities provided by the connection-oriented session service are confirmed, meaning that the client can send Request primitives and receive Confirm primitives and the server can send Response primitives and receive Indication primitives.

The **connectionless** session service provides only non-confirmed services; in particular only unreliable method invocation (asking the server to execute an operation and return a result) and unconfirmed push are available. In this case clients can only use the Request primitive and servers are only able to use the Indication primitive.

To start a new session, the client invokes a WSP primitive that provides some parameters, such as the server address, the client address and client headers. These can be linked to HTTP client headers and can, for example, be used by the server to retrieve the type of user agent within the WAP client (which might be both the version and type of the browser). This is useful when we want to format the output differently, depending on the client's device type. For example, one phone may have a 20 character wide display; another may have a 16 character wide display.

WSP is basically a binary form of HTTP. As previously mentioned, the binary transmission of data between a server and a client is an essential adaptation made for the narrow-bandwidth mobile network. WSP supplies all the methods defined by HTTP/1.1 and allows capability negotiation to gain a full compatibility with HTTP/1.1.

Wireless Transaction Layer – WTP

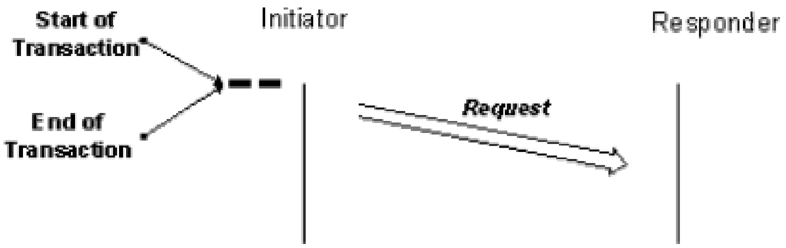
The Wireless Transaction Protocol provides services to accomplish reliable and non-reliable transactions and operates over the WDP layer or over the optional security layer WTLS. WTP, as all the other layers in WAP, is optimized to adapt to the small bandwidth of the radio interface, trying to reduce the total amount of replayed transactions between the client and server.

In particular, three different classes of transaction services are supplied to the upper layers:

- ❑ Unreliable requests
- ❑ Reliable requests
- ❑ Reliable requests with one result message

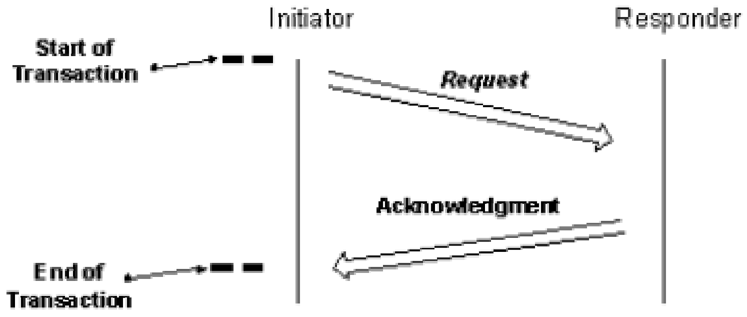
Unreliable Request

The initiator (in this case a content server) sends a request to the responder (the user agent) who does not reply with an acknowledgment. The transaction has no state and terminates once the invoked message is sent:



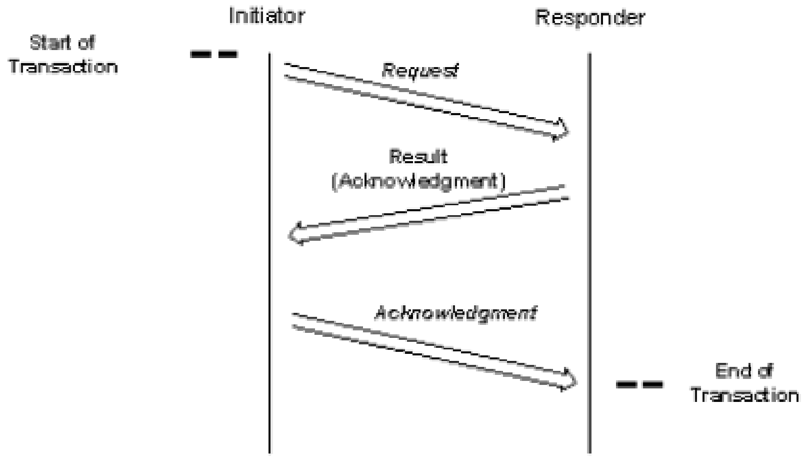
Reliable Request

The initiator sends a request to the responder who acknowledges it. The responder stores the transaction state information for some time, so that it can re-transmit the acknowledgement message if the server requests it again. The transaction ends at the initiator when the initiator receives the acknowledgement message:



Reliable Request with One Result Message

The initiator sends a request to the responder who implicitly acknowledges it with a result message. The initiator then acknowledges the result message, maintaining the transaction state information for some time after the acknowledgment has been sent, in case it fails to arrive. The transaction ends at the responder when it receives the acknowledgement message.



Wireless Transport Layer Security – WTLS

WTLS is the solution to the security issue, provided by WAP Forum. WTLS is an optional layer and is based on TLS (Transport Layer Security) v1.0, which in turn is based on SSL (Secure Sockets Layer) v3.0, which are Internet protocols. WTLS operates over the transport layer (WDP).

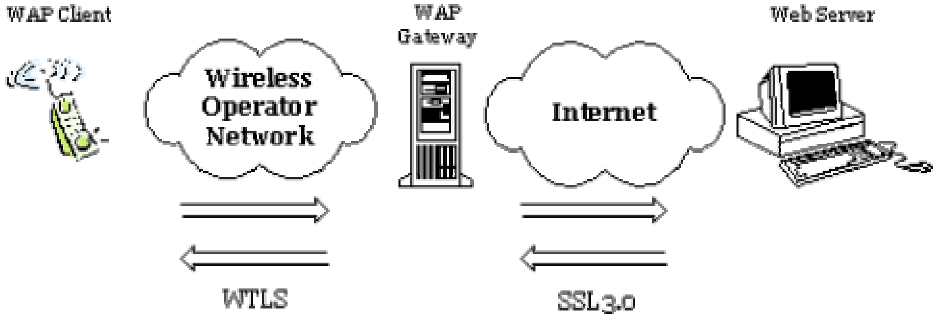
During the past few years, security over the Internet has become a big issue. E-commerce, e-banking and e-trade experienced a big evolution once SSL was standardized. By providing guaranteed privacy, confidentiality and authenticity over the TCP protocol, SSL enabled commercial solutions to expand their services.

For the companies that had to share financial or confidential data over the Net, the fact that everyone could read this data just by sneaking in during a transmission was an enormous limitation. SSL delivered methods to encrypt the data content and to make it accessible only to entrusted users. Furthermore, it gave the users the ability to check whether the content they were requesting was really coming from the origin server it was supposed to.

It should be obvious that WAP also had to adapt to this situation, by offering ways to protect, when needed, the data requested from or sent to the user. In WTLS we find the same fundamental characteristics we observed in all the previous layers in the WAP stack: it is an adaptation of an Internet protocol both to the high-latency, narrow-bandwidth air interface and to the limited memory and processing power of the WAP device. WTLS attempts to lighten the overheads associated with establishing a secure connection between two applications. It provides the same grade of security that is supplied by SSL 3.0, while reducing the transaction times. It provides services that ensure **privacy**, **server authentication, client authentication** and **data integrity**.

- **Privacy** guarantees that the data sent between the server and the client is not accessible to anyone else. No one can read the unencrypted message, although they can see the encrypted message.
- **Server authentication** ensures that the server really is who it claims to be, and that it is not an imposter.
- **Client authentication** provides a way for the origin server to limit the access to the content it provides. Just those subscribers that are recognized as trusted ones, can gain access to the site.
- **Data integrity** takes care that no one can alter the content of a message being transmitted between server and client without one of them noticing.

In the diagram that follows, we show how the WAP gateway handles secure sessions. A standard SSL session is opened between the web server and the WAP gateway and a WTLS session is initialized between the gateway and the mobile device. The encrypted content is sent through this connection from the server to the gateway, which translates it and sends it to the mobile phone.



WTLS empowers the SSL protocol by adding effective features such as datagram support, optimized handshake and dynamic key refreshing.

Today, WAP gateways are available which provide public/private key encryption with a key length up to

1024 bits. To use a secure connection the origin server has to be installed as if we were setting up a secure connection over the Internet; the gateway will take care of matching the SSL connection to a WTLS one.

The translation between SSL and WTLS takes place in the memory of the WAP gateway. It is important that unencrypted information is not stored anywhere in the gateway, since this defeats all the security measures used to protect the stored data from being seen by unauthorized people.

Even though WAP gateways are provided with many features to supply the maximum level of security, there is still a lot of concern surrounding the WAP security solution. Banks and all the companies that really have to protect their data, still prefer to host and install their own gateway, giving them the ability to send encrypted data right to the mobile phones, with no need for translation. Time will show whether WTLS will be gradually adopted as the standard or if it will just be ignored.

WTLS is an optional layer in the WAP stack. This means that *security in WAP is only available on demand and is not a built in feature of the WAP architecture*. Hence, the information traveling to and from the WAP gateway is normally not encrypted, unless we use SSL connections to communicate between the origin servers and the gateway.

Security is discussed in detail in Chapter 16.

Wireless Datagram Protocol – WDP

WDP is the bottom layer of the WAP stack and is one of the elements that makes WAP the extremely portable protocol that it is, operable on extremely different mobile networks. WDP shields the upper layers from the bearer services provided by the network, so allowing the applications a transparent transmission of data over the different bearers. Bearer services are the nitty gritty of communication between the mobile phone and the Base Stations (the antennas). They include SMS, CSD, USSD, DECT, and CDMA.

The physical layer prepares the data to be sent from the mobile device over the air interface, and sends the data using the bearer service implemented in the network that the device is operating in.

Getting a WAP Site on the Air

Now that we have discussed how the WAP protocols work, how a WAP gateway works, and which services an origin server provides, we will try to combine all these elements together. What does happen when you switch on your shiny new WAP phone and access a WAP site?

> *One main assumption made here is that the WAP phone is configured to connect to a WAP gateway and that it works properly. We will also assume, for simplification purposes, that the bearer used is not SMS (Short Message Service) or some similar bearer, but that a* data connection *is initiated between the mobile phone and the gateway (a typical bearer in GSM networks could be CSD for example). If SMS is used, the details contained in this section are still valid, with the exception that the expression 'place a call' should be replaced with 'make a connection'.*

Wireless Networks

When we want to cover an area with a wireless network, we divide the geographic region into sections called **cells**. This is the reason why wireless networks are often called **cellular networks**. Every cell contains an antenna, also called a **Base Station**, which communicates with the mobile phones.

Base Stations are grouped and controlled by a **Base Station Controller** that is attached to a **Mobile Switching Center**. The Base Station Controller has access to the fixed network as well as the entire wireless network; in this way, subscribers are able to communicate both with normal telephones (in fixed networks) and with mobile telephones (in the same or a different wireless network).

While a mobile phone is switched on, it is communicating with a Base Station. Every time someone calls your mobile phone and you are under the coverage of a given Base Station, the Base Station attempts to contact the phone and, if the operation succeeds, your phone informs you of an incoming call (by buzzing, flashing, or whatever other method). If you are moving during a conversation, driving on the highway for example, every time you exit the area covered by a Base Station, a *Hand Over* procedure is initiated to pass the connection to the Base Station that covers the area you are moving into.

In the wireless networks where WAP is implemented, a WAP gateway is installed and connected to the wireless network **LAN** (Local Area Network). A new phone number is defined and assigned to an **Access Server**. When the subscriber initiates a browsing session, a call is placed to that number.

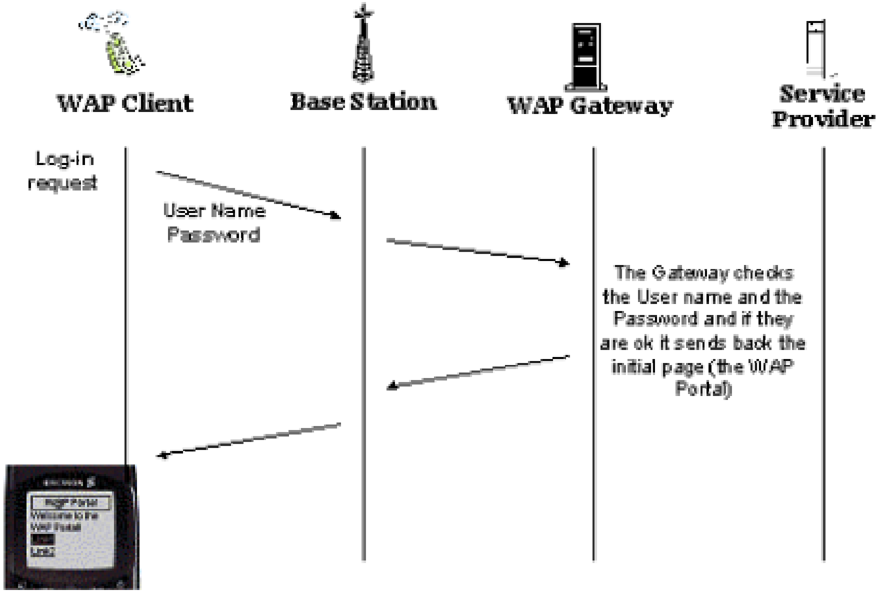
The only purpose of the Access Server is to authenticate the subscriber willing to get in contact with the gateway. It is connected to a database storing the valid subscribers numbers. Once the subscriber has been validated, they will be connected to the internal LAN, and allowed to communicate directly with the WAP gateway.

Browsing WAP Sites

The first connection to the gateway is illustrated in the figure below. The browser contained in the phone will automatically send the subscriber's details to the gateway together with a user name and password. The gateway checks them against a database as would a dialup connection to the Internet in a traditional PC session.

A WAP browser also has an associated 'home page' deck, determined by the service provider, which is loaded into the microbrowser after the user has been authenticated. This deck is a WAP portal, which was introduced earlier in the chapter. It lists available links and services from that gateway.

This process is illustrated below:


WAP Client
Base Station
WAP Gateway
Service Provider
Log-in request
User Name
Password
The Gateway checks the User name and the Password and if they are ok it sends back the initial page (the WAP Portal)

The login procedure that takes place at the gateway will cause the first, sometimes long, wait. However, during the login procedure the gateway doesn't access an external application server, since the WAP portal is stored on the gateway itself or on an application server located in the internal operator network. Once we are logged in, we can start the WAP browsing.

The latency associated with browsing via a mobile device is strongly dependent on the type of bearer used. The more advanced the bearer is, the faster the connection with the WAP gateway, and consequently the less time required to transfer data to and from the gateway. Let's take the GSM Network (the predominant implementation in Europe) as an example. GSM gives the user the choice from a diverse selection of bearers, for example SMS (Short Message System), CSD (Circuit-Switched Data) or the new (at the time of writing – still being tested) GPRS (General Packet Radio Switching).

When we run WAP over an SMS bearer, the WAP Gateway has to divide the content addressed to the WAP device into packets, each one containing at most 160 bytes. The device then has to reassemble the messages it receives to decipher the content. This procedure is very time consuming and accounts for the fact that SMS is the slowest bearer amongst the possible choices.

With CSD, a data connection is established between the WAP device and the WAP Gateway. The speed of the connection is 9600bit per second, providing a faster medium for data exchange compared to SMS.

GPRS is a new technology for data transfer within the GSM wireless network. It has been designed to be an upgrade of the GSM network, supplying more bandwidth to wireless communication. It is still under development in many countries and, while many network operators are advertising its launch in autumn 2000, the lack of terminals and possible technical problems due to its early release may postpone its introduction in the commercial market to early 2001. The main slogan related to GPRS and the one that everyone will be hearing is "always connected, always online". It is a précis of the main capabilities of GPRS (if we forget about the bandwidth for a while).

With GPRS there will be a minimal connection setup procedure which will occur when you switch on your

GPRS phone. After that, you'll be always online, ready to start receiving and sending data in less than one second. A second peculiar characteristic of GPRS is that the subscriber will be charged by the volume of data they send and receive and not, as it is now with the GSM data connection, by the time they are connected. When it enters the commercial market, GPRS will provide speeds up to 171.2 kbit per second, with an average of 30 or 40 kbit per second, which will surely be a step forward for WAP and m-commerce in general.

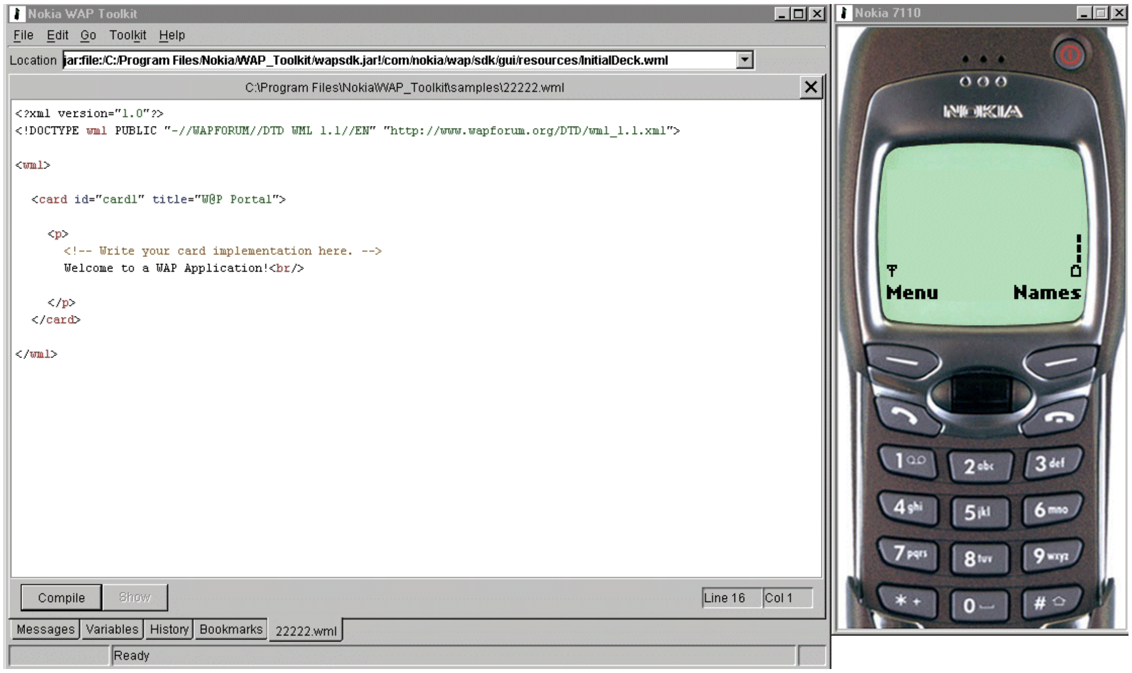
When dealing with voice or data communication in a wireless network we have to take into consideration some more factors that can limit the bandwidth and therefore increase the already high latency. The main problem connected with a wireless network is that between the Base Station and the mobile phones there are obstacles which disturb the transmission. Such obstacles include buildings, tunnels, cars and people, which absorb waves and have a negative impact on the Signal to Noise ratio (SNR). SNR is a way of measuring the amount of noise that is introduced to the signal before it is received. SNR decreases dramatically with distance.

A low SNR in a voice call will just result in strong disturbances and background noise while we are talking. However, in the case of data connections it is more critical. In WAP connections, low SNR will result in lost data, which will mean the retransmission of data that has been corrupted on its way to the mobile device from the gateway. The retransmission will, of course, increase the latency.

How Do I Get Started?

The most obvious way to get started with WAP is, of course, to buy a WAP-enabled device, such as a WAP phone or a WAP-enabled Personal Digital Assistant (PDA), and begin to look at the applications available. There are many applications on the market already, and by the time you read this book there will surely be an even larger choice.

An alternative is to use an emulator – a 'pretend' mobile device that can be used on your desktop computer. We will cover the more widely used emulators in Chapter 2.



Location jar:file:/C:/Program Files/Nokia/WAP_Toolkit/wapsdk.jar!/com/nokia/wap/sdk/gui/resources/InitialDeck.wml

C:\Program Files\Nokia\WAP_Toolkit\samples\22222.wml

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN" "http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

  <card id="card1" title="W@P Portal">

    <p>
      <!-- Write your card implementation here. -->
      Welcome to a WAP Application!<br/>

    </p>
  </card>

</wml>
```

Emulators are available from the four founding members of the WAP Forum, as well as from various third party vendors. The majority of these do not promise an exact rendition of your application; they should be used as a guide and only approximate the final result. Some even depend on the WAP specifications, rather than basing themselves on any one phone. We will cover interoperability issues throughout the book, but the current implementations of WAP among the main players are by no means consistent, making the development of applications difficult. The emulators will, however, be more than adequate to test the logic of your applications. It is also getting at least an hour's worth of 'play' on a real phone, to begin to understand the issues relating to data input on an actual device.

You can also use a WAP plug-in, which allows your standard Internet browser to display WAP content. We have included a list of resources in Appendix F. The majority of these are also useful resource centers for news, new applications and generally for keeping up with market information.

WAP Portal

A good place to start looking for applications is typically a WAP portal. We have already come across this concept a few times in this chapter. A portal provides lists of links, making it easier to navigate to different sites. Popular portals on the Internet include Yahoo.com and AltaVista. They provide links, grouped in categories, and allow you to search for useful resources and save keying in URLs. If you want some news sites or e-commerce sites for example, you simply have to look for the relevant category and follow the most appealing link.

WAP Portals supply you with two useful advantages:

- ❑ You do not have to browse and search for the sites that provide you with a certain service: you find sites already grouped by the service they provide. This, in the case of WAP, saves you a lot of time, since WAP browsing can be rather time consuming, since we have no proper keyboard or mouse available and the speed of the connection is pretty low.

- You don't have to remember or to write down the name of the site you want to visit, since you are automatically redirected by clicking on one of the links listed on the portal.

The address of the WAP portal owned by your network operator is normally specified in the settings of the WAP enabled device, so, just like on an web browser, you will be pointed there whenever you start a WAP session. This can be taken a step further to personal WAP portals. These are powered by a general portal which is customized to the user's preferences, and implements a favorites folder, screening out subjects that the user is not interested in.

WAP versus the Web

One of the main issues with the WAP technology is the wide spread confusion about its potential. When the first WAP phones were hitting the market, it was credited with magical powers, some of which were far beyond the scope of WAP itself.

Now things are clearer: WAP phones are available, people are beginning to use them and there is a better understanding of the opportunities and limitations presented by WAP. It may be said that one of the reasons for this is the similarity between WAP and the web, which results from the fact that WAP is based on the current Internet technologies. This has allowed the use of already proven methods, which have enabled the quick deployment of WAP, and also levered the experience and resources available on the Internet. It must be said, however, that there are major differences between the two. This section will examine some of these differences, and introduce some of the fundamental principles behind programming WAP applications, which we'll be examining in more detail throughout this book.

The first major difference between the web and WAP will be the services offered. While there will be common services such as mail access and reading the latest news, there are many web services that it will not be possible to port to mobile devices. Instead there will be opportunities for many new types of services such as those based on Location Information, which we saw earlier.

Another major difference between WAP and the Internet at the moment lies in the manner of browsing. When *surfing* on the Internet you are, most of the time, not concerned with the length of time you are connected, since nowadays the cost for an Internet call is quite cheap. On the other hand, a WAP user for whom the connection costs are higher will be more concerned with the issue of cost. Inputting data on a phone is relatively difficult, and so also increases the connection time. Therefore, WAP users will want fast access to services guaranteed to be useful to them.

Portals such as yahoo and excite, are present in both the web and WAP scenarios, but in the case of WAP they have an even greater importance. Why go somewhere else to search for a particular service if the service you are searching for is already listed there, just a *click* away? There will of course be situations where the WAP user will wish to enter an exact URL; in this case the user is likely to save it permanently as a bookmark for easy subsequent retrieval.

The predicted growth in mobile phone use and their predominance in modern culture brings with it further issues about WAP usability. The typical WAP user will not necessarily be a computer or web user, so applications should not depend on the computer literacy of the user. WAP applications must therefore be as simple and as user-friendly as possible.

The user interface influences the usability of any given web or WAP service, though this is of higher importance in the case of WAP. With such small screens, information needs to be displayed in a neat and clear way in order to stimulate people into using the service.

On top of all this, you must also remember that WAP user tolerance to errors will be almost non-existent. While the web users are somewhat used to "ERROR 404. Server too busy" messages and are willing to put up with them, the average WAP user will probably be much less tolerant and will stop trying to access a service if errors pop up regularly. Furthermore if error messages have not been properly programmed, the user will be unable to browse back or to perform whatever action they want, other than typing in a new address. Developers should bear in mind that they should always give the user a chance to go a step backwards or to try again if something in their application has gone wrong. The usability problem is one of the biggest issues in today's wireless world and will be discussed at length in Chapter 7.

WAP 1.2 - WTA and Push Features

The WTA and push technologies, which have been mentioned occasionally within this chapter, are two features that increase the value of WAP. Unfortunately both the technologies are for the moment not yet available. WTA was defined in WAP 1.1, but it is not yet supported by many of the mobile devices and networks. The push feature has been defined in WAP 1.2 and should be implemented in the WAP phones and operator networks of the second generation.

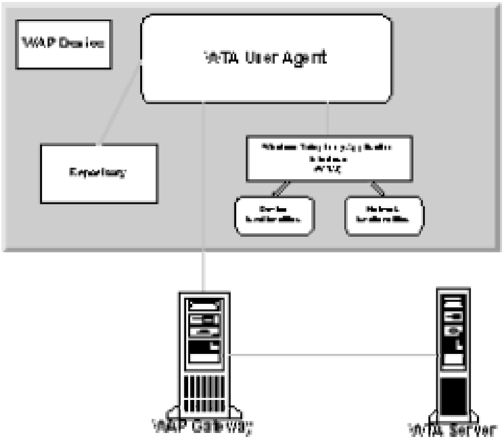
Wireless Telephony Application Interface – WTAI

WAP has been created to give us the chance to interact with the Internet and Internet-like services from our mobile phones. It is then natural to think the other way round? Can we give the Internet a way to use the mobile phone functionality? The answer to this question is supplied by the WTA framework. For now, we'll just look at a brief introduction to WTA, as it is discussed in detail in Chapter 18.

WTA is used to access the services that are present either locally on the client device or in the mobile network. An example of local services on the client might be retrieving or deleting names in the phonebook. Network services include voice mail interaction or cell location information. All of the WAP-compliant devices should support the WTA local services, so, theoretically, the WTA functions should work on every WAP compliant phone. Currently, this does not apply, since WTA is not yet fully implemented on the first generation of WAP phones. However, it should be available on the second generation.

Both local and network services are executed by an entity contained in the mobile devices called **WTA User Agent**. This is responsible for the retrieval and execution of WTA functions. All the WTA functions, both local and network, are contained in libraries stored on the **WTA Server**, a server located in the operator network. An interface called **WTAI** (Wireless Telephony Application Interface) ensures the interaction of the WTA User Agent with the device functionalities and the network services.

All these concepts are illustrated in the diagram below. When we want to access a WTA service, the WTA user agent sends a request to the WAP gateway which contains the name of the library with the function we want to use. The gateway requests the function code from the WTA server and then sends it back to the WTA user agent. The agent will now execute the code with the help of the WTAI, which is the interface in the phone to the device dependent functionalities (the phonebook for example) or the network functionalities (like placing a call).



Since the retrieval of the proper file from the WTA server is quite time consuming, a repository in the mobile client can be used to store the most commonly used WTA functions. The size of the repository has not been defined yet by the WAP Forum and can therefore vary depending on the particular WAP device.

The WTA defines a set of functions that resemble WMLScript functions; they are accessible from WMLScript code and sometimes, depending on the service, from WML code as well. If for example we want to place a call from inside WMLScript, we can call a function like this:

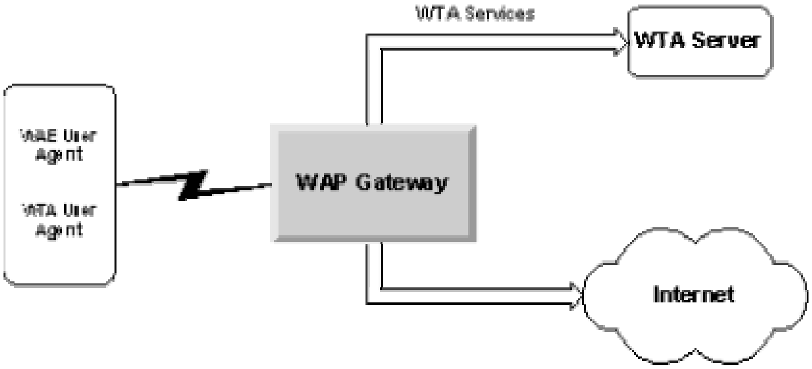
```
WTAVoiceCall.setup("+39089456789", 1);
```

Here, `WTAVoiceCall` is the library where the function `setup()` is contained. The string and the number inside the parentheses are simply the parameters we are sending to the function.

If we want to place a call from inside WML instead, we would use something like this:

```
wtai://wp/mc;+39089456789,1
```

The diagram below illustrates the use of WTA. Notice how the WAP client must request the file from the gateway, which in turn, requests it from the WTA server:



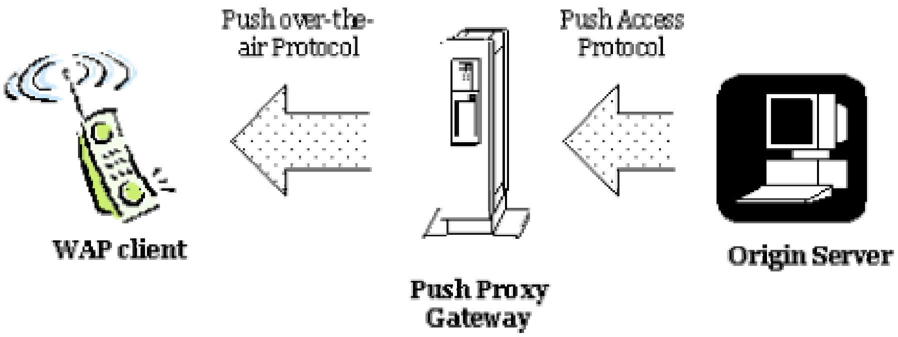
Unfortunately, WTA functions are not yet supported in the majority of the WAP mobile phones on the market now. Some phones give access to some of their internal functionality, but to fully experience the power of the WTA functions we will have to wait for the next generation of devices.

Push Technology

In November 1999 the WAP Forum released the specifications for WAP 1.2. The most important addition to this version is the launch of push technology to deliver content to WAP terminals, without the server receiving any request for it. Push will be discussed in detail in Chapter 17, so we'll just look at a brief overview here.

The potential of push technology has opened the door to an abundance of new services, and everyone in the WAP market is looking forward to being able to use it. WAP version 1.2 has introduced some problems as well, however. For example, some of the WAP mobile phones on the market today will not allow an upgrade from version 1.1, so services will need to make sure that they can cope with the first generation of WAP phones, as well as providing more powerful applications for the second.

With the advent of push technology, the WAP architecture has grown to include a new element, the **Push Proxy Gateway** (PPG), which receives the push request from the origin servers and forwards them to the WAP terminals. The PPG and the WAP gateway can coexist on the same network element.



In the new architecture there are also two new protocols involved:

- **Push Access Protocol** (PAP)
- **Push over-the-air** (OTA) **Protocol**.

PAP is used by the origin server to communicate with the Push Proxy Gateway. It uses HTTP/1.1 and the HTTP `POST` method to transmit the information to the proxy. In the specification for the Push Access Protocol, it states that PAP has been designed so that it can be tunneled through other and future protocols although HTTP is the first to be supported. **Tunneling** is the capability to send protocol messages untouched over whatever other protocol is being used. With this in mind, it is clear from the above statement that the PAP protocol has been conceived to grant future scalability and compatibility with different Internet protocols such as SMTP or FTP.

Using the PAP protocol, the origin server can perform various different operations:

- Send content to a mobile phone (through the Push Proxy Gateway).
- Read the status of a previous push operation. This allows the origin server to request, via a message following the push one, information about whether the pushed content has arrived or not.
- Be notified when the content reaches the terminal. This feature is similar to the previous one; the difference being that here the origin server sends a push request to the Push Proxy Gateway asking to be sent a notification when the actual push content is delivered.
- Cancel a previous push operation.
- Request from the Push Proxy Gateway information about the type and capabilities of a specific device.

The Push over-the-air Protocol is based on WSP and is used by the Push Proxy Gateway to send the pushed content to the mobile device.

The Push Proxy Gateway itself has logically almost the same functions as the WAP gateway:

- Acts as a protocol gateway, converting from Push Access Protocol to Push OTA Protocol.
- Resolves the addresses and locates the devices to which the content is being pushed.
- Determines what kind of capabilities a given device in the network has and sends a report to the origin server if requested. This feature is useful for the adaptation of the content for a particular type of device.
- Identifies and authenticates the origin server from where the push requests arrive.
- Encodes the content it receives in the proper format for over the air transmission.

Summary

We are facing a new and exciting era. Telecommunications are spreading and the *mobility concept,* once just related to the possibility of *speaking* to someone independently of their location, has changed to assume a wider and deeper meaning. People that ask for mobility demand access to personal data, wherever they are in the world and at any time. The same people also demand ease of use. The simpler and more user-friendly a technology is, the more chance it has of becoming a winning technology. All these factors went into the design of a way to access the Internet and Internet-like services from a mobile device.

The Wireless Application Protocol (WAP) aims to become the standard for operators, device manufacturers, and application developers to bridge the gap between the Internet and the wireless network domain. The real power of WAP technology is that it provides everyone with opportunities: device manufacturers, application developers and network operators will increase their revenues by adopting WAP, while the final users will obtain a whole new set of services that will increase their mobility.

WAP is an open specification, meaning that every member of the WAP forum can contribute to its design. The WAP architecture is derived from that of the Internet, with clients and servers communicating via a new element – the WAP gateway. It has a key role in the WAP architecture, acting as a translator between WAP devices and web servers, since the two of them are placed in different networks and therefore use different protocols.

We have seen how the different layers composing the WAP stack are related to each other, and how they relate to the TCP/IP stack used in the Internet world. We have also clarified the concepts of the WAP gateway, the WAP proxy and to some extent the WAP server.

At the end of the chapter we began talking about the development of WAP applications. In the next chapter we will discuss the different platforms available and the different development toolkits we can use to design and test WAP software.