Developing Web Applications For PDAs and Cell Phones

A standard named the Mobile Station Execution Environment, MexE, has been developed which sets levels of standards for mobile devices. These various levels are called classmarks and there is currently three defined classmarks in the standard. This could be expanded at a later time. The current classmarks include, classmark 1 which is the Wireless Application Protocol, WAP which is the most basic layer. The second classmark is built off of classmark 1 and includes PersonalJava and JavaPhone. The third is the Java 2 Mobile Edition, J2ME, which includes CLDC and MIDP.

Classmark #1: Wireless Application Protocol

The Wireless Application Protocol, WAP, is a set of layered protocols which sets a standard for wireless communication. The WAP standard mimics the protocols which are used for the internet or more specifically the web, however, WAP protocols are designed to address the issues that arise in a wireless environment.

WAP addresses many issues which do not exist in a wired environment yet are of major concern for the successful adoption of wireless applications for mobile devices. Firstly, wireless communication in itself presents us with the problem of intermittent network connectivity, as the user can potentially be traveling and thus moving in and out of connectivity. Latency also becomes an issue, as wireless data transfer rates cannot match that of the wired equivalent. WAP takes steps to increase responsiveness and reduce network flow to hopefully provide fast and responsive services. Mobile devices also tend to be small. Inherently, they generally have small screens and input interface that pale in comparison to their desktop brethren. As data is broadcast in open air there is always the potential that a third party could also be intercepting your broadcasts and

inherently some information, which depending on the type of transaction, may be quite important. Mobile devices also run off batteries and therefore have a limited power supply. Processors in mobile devices also do not match the speed and processing power of regular computers. WAP therefore has the challenge of addressing all of these issues.

WAP is built around the idea of having a gateway server which mobile devices will connect to. The WAP gateway acts as a middleman to connect the mobile device to the rest of the web. The WAP gateway is used to address several of the above issues. One such function of the gateway is to receive encrypted data from a mobile device where it is decrypted and then re-encrypted using a web encryption standard, such as SSL. This happens in reverse when data flows towards the mobile device. This is done because the WAP encryption protocol is lightweight to reduce processing demands on the mobile device. By having a more robust encryption protocol for transmitting over the rest of the web privacy of messages is increased. Unfortunately by doing this the gateway becomes the security hole in the system. Therefore great care must be taken to ensure that WAP gateways are not compromised. The gateway also converts all data to send to binary, as appose to sending text, to reduce network load.

The Wireless Application Environment (WAE), Wireless Session Protocol (WSP), Wireless Transaction Protocol (WTP), Wireless Transport Layer Security (WTLS), and the Wireless Datagram Protocol (WDP) make up the set of protocols which WAP is based on. The Wireless Application Environment (WAE) can be broken down into its own set of protocols including the Wireless Markup Language (WML), WMLScript, and the Wireless Telephony Application (WTAI/WTA).

WML, Wireless Markup Language, was created to replace HTML for WAP. WML is XML based but unlike HTML, WML follows the strict formatting rules governing XML. When using HTML a site is generally a collection of HTML pages where each viewable page is split up into its own file. This is not the case in WML sites are contained in one file and the collection of all the pages is referred to as a "deck",

where each page which a user views is called a "card". Cards are much smaller than your average webpage as the viewable area on most mobile devices is quite small.

WMLScript, is the JavaScript equivalent for WAP however it does have some important differences. WMLScript is has less functionality than JavaScript. WMLScript is compiled into byte code at the WAP gateway to reduce the load on the mobile devices processor as well as reducing network flow. WMLScript is not embedded into your WML but is found in a ".wmls" file.

The Wireless Session Protocol, WSP, is a session level protocol which is the HTTP equivalent for WAP. WSP has been tweaked to get more out the available bandwidth which mobile devices use. One of the ways it does this is to transmit data in a binary form rather than text which greatly reduces the size of data to be transmitted.

The Wireless Transaction Protocol, WTP, is a transport level protocol which is quite similar to TCP. WTP provides both capabilities for reliable and unreliable data transport depending on the needs of the application. Its main goal is to achieve low latency over the wireless network.

The Wireless Transaction Layer Security is a security layer protocol which mirrors the functionality of SSL, Secure Socket Layer. WTLS provides authentication and encryption functionality which is used only when needed by applications. WTLS is not as secure as WTLS due to the smaller key size that is used in its encryption algorithm.

The Wireless Datagram Protocol is the lowest protocol level in WAP and it works on the same level as UDP, User Datagram Protocol. Protocol layers under WDP are not defined in the WAP specification as these protocols are generally device dependent. Point-to-Point Protocol (PPP), Short Messaging Service (SMS), and General Packet Radio System (GPRS) and some of the most predominantly used protocols under WDP.

Current WAP protocols have attempted to achieve the objectives of security, privacy and usability, however, compromises have been made and the system is still far from perfect.

Classmark #2: JavaPhone

Java Phone is an API that allows for the development of Java applications for cellular telephones. It is written by Sun Microsystems. Packages are also available for Internet screen phones.

Applications developed with Java Phone are executed in the PersonalJava or EmbeddedJava application environment. These two environments act as the Java virtual machine for the cellphones. This allows for compatibility across a wide variety of cell phone hardware and real time operating systems. PersonalJava is currently being phased out by Sun Microsystems

Using JavaPhone does not require the learning of any new languages as it is a extension of the Java language. Applications can be developed on a computer and uploaded to the cell phone or the can be built into the cell phone itself. All cell phones that use JavaPhone require at least one program to be built in. This program is the software loader and is implemented with serial communications for downloading programs and installing (identifying the entry point) them on the cell phone.

JavaPhone components include datagram messaging allowing for independent addressing and delivery of messages to other wireless devices. Also provided is a serial communication package. This package includes a Secure Socket Layer API to provide secure communication over TCP/IP sockets.

The rest of the Java Phone API is directly used for application development on a cellphone. The packages include calendar information, power management, power monitoring, address book, user profile packages and more.

Classmark #3: J2ME, CLDC & MIDP

Since its beginnings, JAVA has been ported onto more systems then any other language before it. It is used on items such as mobile phones to full-blown transactions servers. All this success is due to one small well-put idea, the java virtual machine standard.

In the past few years processing power has been getting significantly cheaper. Therefore JAVA had been getting significantly more popular. Since a language like C or C++ produce much more efficient native code, it makes them somewhat competitive with java because they run faster. Even though this is true, the other languages are extremely limited to cross platform portability. Sacrificing some of this efficiency java is capable of being one of the most portable languages today. This means that it is compiled once to java byte code, and then the platform dependant virtual machine on operating systems does the rest,

Lately a new type of standard has been developed, the J2ME, Java 2 Micro Edition. It is a small sibling to the J2EE, Java 2 standard edition. The J2EE JVM equivalent for the J2ME is the K-Virtual Machine

Because of personal devices hard ware limitation, such as memory, user interface, and especially processing power. The J2ME has been designed with these limitations in mind, there for the KVM is much smaller then JVM. Such devices are connectable Consumer Devices, Cellular phones, Personal digital assistants.

J2ME has to primary components, configurations and profiles. Configurations configure the hardware aspect of devices and their capabilities. Profiles add APIs onto a configuration that provides and extends the device capabilities. The Connected Limited Device Configuration (CLDC) in J2ME specifies an environment for Mobile phones, and personal data organizers. Devices with limited memory, 128 kb - 512 kb, intermittent, low bandwidth network link, constrained UI's, small screens and most often wireless. It makes use of a JVM designed for resource constrained devices. The Mobile Information Device Profile (MIDP) also specifies an environment for Mobile phones and personal data organizers. It addresses issues such as user interface, Store and Manage persistent local data, networking, application model. The MIDP runs on top of the CLDC. Developed by an expert group made up of 20 companies representing the wireless industry.

J2ME software ranges from items such as Active Desktop where you can view your home computers desktop remotely. Web browsers, such as Web Viewer, are also very popular to make games with. One game is the once popular Street Fighter, which now people are able to play on their mobile device.

The J2ME tools are all available now and what is best is that most of them are free. and they are all you need to create mobile software. J2ME program is called a MIDlet and when creating a MIDlet with java, key property is to extend you class with the MIDlet object.

References for more information on F2ME can be found at

http://java.sun.com/j2me/
http://java.sun.com/products/cldc/
http://java.sun.com/products/midp/

Smart Cards

Smart cards have been in use for quite a few years. They serve a great variety of purposes and are used in dozens of different kinds of devices. Among these devices are cell phones and PDA's which are the focus of this discussion.

The smart cards contained in cell phones and PDAs are extremely small, generally about as big as a thumb. This is in comparison to, say, the smart cards used in satellite receivers, or used as credit cards, which are the size of...well...a credit card. There are also various sizes in between.

So what are smart cards? Smart cards are essentially microcomputer devices capable of handing data, and running Operating Systems. They are designed to be virtually tamper-proof which is important when considering that they can be used for banking, and communications. Obviously, the infiltration of these devices would be potentially disastrous. Smart cards use what are called Application Protocol Data Units (APDU) which allow the cards to connect to host devices. These connections are made through the use of PIN codes and cryptographic keys.

Smart cards come in all forms and sizes as I mentioned earlier. There are several different types of cards, but the main focus here will be on the SIM card and the WIM card.

SIM cards are most commonly found in communication devices like cell phones and PDAs. SIM stands for Subscriber Identity Module. This means that a SIM card is sort of like a fingerprint for you electronic device in that it identifies it uniquely. No two SIM cards are the same. SIM cards are virtually a microcomputer consisting of a microprocessor, ROM, EEPROM memory, volatile RAM, and a serial I/O interface. SIM cards accept software. This software is usually an operating system, a file system

and various applications. A SIM card cannot operate by itself. It has no internal power supply and therefore relies on its host device for power.

The SIM toolkit is an ETSI/SMG standard for Value Added Services and e-commerce using cell phones to do the transactions. The SIM toolkit allows cell phone and PDA users to check their bank account balances and pay bills among other things. The SIM toolkit is programmed into a SIM card. It allows the card to run a handset interface , maintain information exchange between the end user and the network and access or control access to the network. The SIM card initiates commands independently of the phone/PDA and the network. The SIM toolkit was first standardized in 1996.

WIM, or WAP Identity Module, cards are basically security cards. They operate in conjunction with a SIM card, keeping transactions between the phone/PDA and the network secure on both ends. WIM was introduced with WAP protocol 1.2. It provides end-to-end security for Wireless Application Protocol applications, which are a great improvement over the limitations of 1.1. A WIM card allows the end user to store security certificates and digital signatures to ensure safe and secure transactions from the phone/PDA to the wireless network.