Mobile Software Licensing

Ivana Dusparic and Jim Dowling

3rd July 2003

Abstract

Licensing models and license management systems are going through major changes because the traditional shrink-wrapped software way of licensing does not meet the needs of the modern software market. In particular the emergence of new platforms, such as mobile computing and the assumption by software vendors of clients possessing network connected software, will revolutionize how software is licensed and managed. Software license systems are moving towards more flexibility and supporting several licensing models. Usage of the software can be tracked by licensing servers, and new pay-per-use models are being introduced. This paper reviews current situation in the mobile software applications, analyzes current licensing systems and models used in mobile computing and issues associated with them, and introduces possible future developments in the field.

1 Introduction

Software licensing is an area that is undergoing significant structural changes in recent years, centred primarily around the notion of software delivered as a service rather than as a physical package. Vendors are now shifting from traditional shrink-wrapped package software delivery to licensed software, especially electronic software/license distribution and management [23]. Traditionally, software licenses have been sold on a perpetual basis, where customers were purchasing 'lifetime' licenses. This model, even though widely accepted because of its simplicity, has resulted in a general perception that, once a package was purchased, the customer owns the software, rather than only having the right-to-use it under certain conditions specified in the license agreement. In general, the customer does not have the right to modify, copy or distribute the software. After the termination of the license agreement, the customer is no longer legally allowed to use the product [23]. This type of license, since it covers unlimited usage of the product, is generally expensive for customers and fails to meet needs of the modern software market, particularly in the developing market of software for mobile devices, due to inflexibility and cost.

Growth in the level of network connectivity of desktops and in the adoption of mobile devices has changed the requirements for methods of delivery and licensing of software. With an estimated 1.05 billion mobile phones in use at the end of 2002 and 13.1 PDAs sold in the year 2001 alone [30], mobile devices represent increasingly important platform for software applications, and these applications require effective licensing systems.

2 State of the Art in Software Applications for Mobile Devices

Software developed for mobile devices can roughly be divided in three categories that differ in their main purpose, target user, price, and consequently in ways the application should be licensed. The areas are: platform software, and two kinds of application software–enterprise solutions, developed for large companies for business activities, and personal software, developed for individual consumers for entertainment or personal productivity.

- Platform Software–There are currently a few operating systems (OS) competing for the mobile device market. Symbian OS [16] is currently the most popular OS on mobile phones (Nokia, SonyEricsson, Motorola, and Siemens) together with Palm OS [13] developed for handhelds. Microsoft developed Windows CE [18]and Microsoft SmartPhone [10] operating system for smartphones and pocket PCs. Other platform software includes J2ME (Java 2 Micro Edition), user interfaces such are UIQ (SonyEricsson) and Series 6.0 (Nokia) and other middleware software.

- Enterprise Applications–They are developed for companies that have workers in the field who need to be connected with each other and with the central servers and databases. Examples of target workforce include medical teams, pharmaceutical teams, sales representatives, insurance agents, delivery, law enforcement, government agencies etc. Such applications are mostly customized for specific requirements, but there are some off the shell applications as well that evolved from customized solutions. Such an example is AvantGo Mobile Delivery package initially developed for McKesson Corporation, a healthcare supplier, as Closed Loop Delivery System [32], but is today available as an off the shelf product [27]. These enterprise applications can be user applications installed on mobile devices, or middleware used to mediate between applications and servers, or combination of both. Oracle, IBM, and Microsoft are also introducing mobile device editions of the clients for their enterprise solutions (SmartPhones, Palms, PDAs, Pocket PCs).

- Consumer Applications–are generally developed for personal use. Consumer applications can be broken down further into entertainment software, personal productivity, business applications, content tools etc. Currently the most popular entertainment software is games. They can be single player or multiple player games where gamers are connected on the network. Networked games on mobile devices can be played against other mobile users via GPRS, Bluetooth or infrared connections. Personal productivity software includes internet and e-mail applications, organizers, reference applications/dictionaries etc. Business mobile software includes applications that help small businesses conduct their operations, particularly individuals who conduct their business away from the central office–doctors, veterinarians, drivers, couriers etc.

Handago compiled a statistics for mobile software sold in the year 2002 to show that games and personal productivity take up the largest share of the mobile software sales (see Figure 1.)

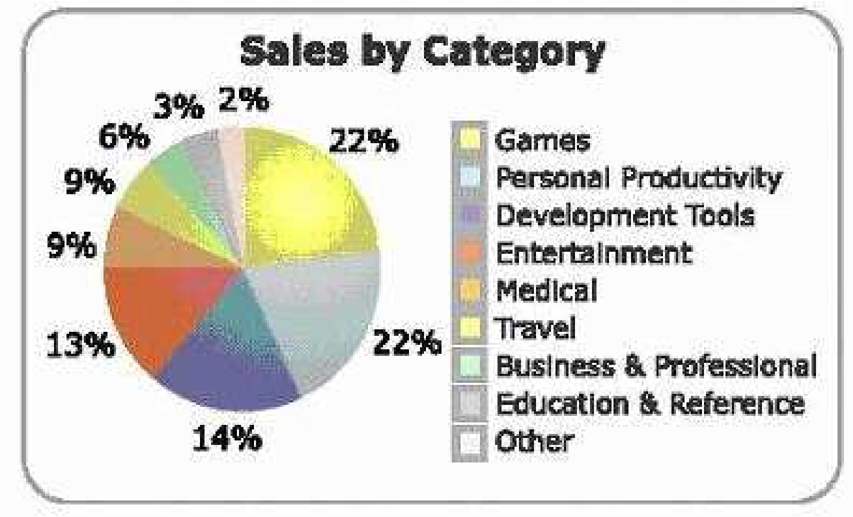

Figure 1: Handago statistics 2002[25]

22% of all mobile software sold in year 2002 was games, and another 22% were personal productivity tools - browsers, e-mail clients, reference and educational tools etc [25]. Significant growth in mobile applications market is expected to continue in the future, in particular development and usage of mobile games. According to UK telecoms consultancy BWCS, by 2007 there will be more than 200 million mobile subscribers regularly playing games on their phones [30]. However, the estimated growth will keep up only if software applications, and the way they are distributed and licensed, is adjusted to suit the end users needs, proves easy to use, and is affordable to wider audience.

3 Licensing Requirements for Mobile Applications

Mobile applications today are licensed in several ways, some of them adopted from the licensing models for PCs and some of them introduced when distributed software applications have been developed. These models have to meet requirements of software vendors, customers, and end users. In the case of the mobile software industry the mobile network operator, mobile device manufacturer, and billing provider also have to be included in the process of software licensing.

Vendors are interested in restricting piracy and unlicensed use of their software but without imposing unnecessary restrictions that would make their software inflexible for the honest customers and have them turn to competitors' products. Another goal of vendors is also to gather statistics on software usage [28].

Enterprise customers do not want software they paid for to have any unreasonable terms of usage or difficult to use complex licensing models and implementations. Customers would also like gather statistics on software usage to determine whether their current licensing model is leaving them over-licensed or under-licensed. They do not want to pay for more software licenses or longer licensing terms than they actually need for their business. They need to audit, control and restrict usage of their licensed software and to be able to exceed licensing terms if need be and then settle with the vendor later. Customers want to try out software for free and decide does it meet their needs before purchasing the license [28].

End-users of the software would like to be shielded from the underlying license agreement. They want to use the software at any time they need to without worrying about license availability and compliance with license terms while performing their activities [28].

Handset device manufacturers and mobile operator often act as providers of content and software and as such are involved in the licensing processes. Since delivery and licensing of the software is done over the Internet to PCs or mobile networks to mobile devices, payment for content and software should not differ and should be made simple for end users. Network operators can be providers of the billing system, or a third party's billing system can be used [26].

Depending on type of mobile software, the requirements of the architecture and features of licensing system differ.

3.1 Platform Software

Platform software licensing is relatively straight-forward and customers/end users are shielded from it. The handset manufacturer pays the OS vendor for every device using the given operating system, and the price is included in the overall cost of the device.

3.2 Enterprise Applications

The architecture of the licensing system for the enterprise applications is quite complex to develop and manage. Enterprise applications licenses are usually volume licenses that include dozens or even hundreds individual licenses that need to be managed and flexible to support various licensing models. Vendors of enterprise applications would most often like to host license servers themselves rather than at third party servers. Large companies using enterprise applications would also like to have a licensing server installed locally that would be connected to the vendor's server and would manage and distribute the licenses within a company. Enterprises use several applications from different vendors with different licensing terms, and they would like to be able to manage and track usage of these licenses through the same system. This can be accomplished if vendors/applications implement licensing according to the standards using the same APIs. The Open Group specifies XSLM (X-Open Software Licensed use and Management) as a technical standard for implementing software licenses [20]. It defines centralized license use and management system introduced to solve interoperability problems among different license management systems, license certificate formats and operating environments [24]. Licensing terms and conditions are stored in an external certificate separated from the program logic allowing vendors to change licensing terms without changing the application. Certificate files are tamper-proof and encrypted and as such suitable for electronic transmission. XSLM license servers can administer several applications across platforms and different locations in the enterprise [20]. No actual license management system has yet been fully developed to meet this standard but several companies (Isogon, IBM) released solutions partially implementing XSLM and are planning to fully implement it in the future [8].

3.3 Consumer applications

Consumer applications for mobile devices are small applications developed by individual customers or small businesses. It is often not feasible or even possible for vendors of these applications to distribute software and host licensing systems themselves, but they do it through third party systems. These third party software/license distributors provide access to various applications developed by different vendors. Licensing for consumer applications is less demanding than licensing for enterprise applications, but still has to be flexible, easy to use and affordable to end users. It has to support various licensing models to meet the needs of consumers who need to use the application on the regular basis as well as those who need to use it for a shorter amount of time.

4 Licensing Models

Current licensing models for PC software will be the basis for licensing models for network addressable mobile devices. Those models are pay-per-use, named-user, concurrent user, node-locked, time limited, metered use, package suits, mobile licenses, various value-based pricing models (number of processors, number of employees, total revenue of the company etc), as well as combination of these models [5, 33].

- *Shrink-wrapped package.* License agreement is accepted by unwrapping the product or digitally by checking the "accept" checkbox. The customer has unlimited lifetime usage of the software under the terms in the license agreement, usually stating the number of devices software can be installed on.

- *Named user (personal license).* The product may be used on any device on the network by any of the users listed as network licensed users. The vendor specifies the number of licensed users based on purchase conditions and the customer's system administrator names the licensed users [5].

- *Concurrent user(floating license).* The product can be used by a maximum n users on the network at any time, regardless of the network node or username [5] .

- *Node-locked.* Allows unlimited use of the software product only on the certain network node [5].

- *Time-limited.* Used for evaluation (demo) or lease purposes. Functionality provided by the demo license may be less than the one of standard product, while the lease one has full functionality but is time-limited. It is limited by the date of expiration (expires n days after the installation), or after the product/feature has been used n times [5].

- *Enable/disable model.* Features of the product are combined to provide the product most suitable for required functionality. Features can be enabled and disabled. Commonly a single software binary would have three predefined packages to pick from– basic (or light) version, standard version and professional version [5].
- *Package suite.* Packaging independent software modules into a single package. Packages are defined by vendors in the license and not in the software itself [5].
- *Pay-per-use.* Currently most popular with ASPs(application service providers) but very desirable for wider use. Vendors monitor software usage and patterns and bill the customers based on this data [5].
- *Subscription based.* Company or an individual is subscribed, by paying ahead, for use of the software (or access to the content) for certain amount of time, usually a month or a year.
- *Value-based licensing.* Currently most popular example of value pricing is per number of CPUs in PC. However, depending on the type of software, these models could expand to account for the amount of RAM, total revenue of the company using it, total number of employees in the company etc. Recently the problem came up with the licensing per CPU due to new Intel's technology introducing multiple virtual processors within a single physical processor. The question is whether customers should now be charged for the license by the number of virtual or physical processors [33].
- *Open source software.* This model differs from the above listed ones in such a way that source of this software is freely open and available for anyone to see or modify (in the above commercial licensing models source code is not visible to the customer and only vendor's developers can access and modify it). Certain license models, such as GNU, require that the modified application must be available under same license as original.

Some of these models are desirable as licensing models for both enterprise and consumer applications, while some make sense only on enterprise software. Enterprise applications should be able to support all the network models of licensing– named user, concurrent user, and node-locked, because it is likely that users of individual licenses within a company will be connected by the network and connected to the central licensing server. Usage-based licenses (subscription, pay per use) as well as demo/free trial licenses should be enabled on both enterprise applications and consumer applications. It is important to support various licensing models in order to make the product useful for wider range of customers– those who need to use the application on regular basis for longer period of time, as well as those who need to use it only a few times, or only over a certain shorter period of time.

5 Licensing Models in the Mobile Industry by Type of Software

Licensing models for enterprise applications and consumer application are quite different. They differ depending on who is the provider of the application–whether it is obtained through device manufacturer or network operator, depending on what technique is used to verify the identity of the device/user on the network. Enterprise software licensing systems are generally obtained directly from software vendor and need to support various licensing models, and to have ways to manage licenses at customer's server as well. Enterprise solutions developed by companies such are AvantGo or Extended Systems are sold as volume packages that make certain number of licenses available to the customer. In order for customers to evaluate the software these companies provide 30 or 60 days free trial for certain number of devices [4, 1]. Extended Systems solution XTNDConnect Server requires every mobile device to check the license with the server at the begining of application use. The license file is present on every mobile device. The license is specific to a certain device, and the only way it can be made available to other devices is after it has not been used for 14 days. The server assumes the device that didn't use the license for 14 days won't be using it anymore, and makes the license available for use on other devices [4]. This solution, although it protects vendors from unlicensed use of the software, tends to lack the flexibility customers might require. Another enterprise solution, Oracle database clients for mobile devices, uses named user licensing model. This is the same model used on database clients on desktops.

Consumer software applications for mobile devices are largely licensed based on identification of the hardware (IMEI code–International Mobile Equipment Identity [11]) or network identification of the user–(MSISDN number–Mobile Station ISDN number [11]). For example, Wireless Game Club [19] allows users to download the software only after registration that includes the IMEI number of the device user will be downloading software to. A license code is generated using this number and the software is licensed and can be used only on that device. Similar restrictions apply to the Vodafone live service [17] and Selatra wireless lifestyle services [15]. Once a user pays for the game and downloads it to a device, the software can only be used, although for an unlimited time, on that device. This is a per-download licensing model. Payment for the software is one-time fee that can, in the case where the software provider is a network provider, be added to a monthly bill or subtracted from a prepaid account, or in the case of other software providers, can be billed by reverse SMS billing or a credit card. Applications sold over Nokia Software Market require either a phone number or IMEI code, depending on the licensing model software developers require. Software licensed by phone number is delivered over the air (OTA) and cannot be returned/refunded. Software licensed by IMEI code is downloaded to a PC from the website and transferred to a device via cable, infrared or Bluetooth. IMEI code licensed software can be returned/refunded [12]. Applications payment is done by credit card prior to the download, and licenses the user for unlimited usage of the software on a specified device. FoneArcade [6] requires users to provide their phone number prior to game download. Downloads can be paid for by credit card or by a text message sent to a premium number where text messages cost the standard rate plus the cost of the game. A single premium SMS costs Euro 1.50. Users are required to send one, two, or three messages depending on the cost of the game they are purchasing and the games are downloaded using WAP on successful delivery of the SMS messages. PhoneGames4U provides downloads of the games for mobile phone that are paid by credit card, and does not require any mobile phone authentication (IMEI, MSISDN etc). Once a download is purchased, users are sent a URL from where to download their game. The license is not tied to either IMEI or mobile phone number [14]. Bango.net provides both software from independent vendors and content from content providers to mobile users through their bango number systems. Mobile users access bango numbers instead of the usual URLs to access the content. Users pay for content/software through their phone bill, txt trigger (SMS sent to a premium SMS number), credit card, telephone, or using Vodafone m-pay [2].

License protection methods used in software licensing can also be incorporated with the protection of copyrighted digital content distributed to mobile devices–digital license management (DRM). DRM software controls the way digital content is viewed and forwarded to other devices–it can disable content from being viewed after a certain period expires or after it is viewed certain number of times, and it can disable forwarding of the content. LockStream recently released their version of DRM software that is the first one to support digital rights management not only for content but also games for mobile devices [21]. The Open Mobile Alliance adopted Open Digital Rights Language (ODRL)–standard language for protecting copyrighted content on mobile devices. ODRL enables tracking and limiting how many times content can be consumed, for how long, how many times it can be forwarded, and if it can be forwarded at all [22]. The same concept applies to licensing of the software on mobile devices–software has to be able to control or track number of times software is used and disable unauthorized forwarding of the software.

6 License Validation

License validation is directly dependant on the licensing model implemented. Traditionally, shrink-wrapped software packages were distributed together with the license key/serial number that user would have to enter during the installation of software. There was nothing to prevent software and the same serial number to be used on several machines. Validation software checks the serial number against the algorithm used for generating valid serial numbers and based on the results allows or disables the installation of the software. Another technique is for the serial number provided to be combined with the hardware serial number to become installation ID. The installation ID by itself is not enough to enable the software to run– the software has to be activated either online or over the telephone. During activation it is checked whether the given serial number is used in combination with various hardware serial numbers to ensure the software was installed only a single machine, in a case of a single license.

Similar to the traditional models, mobile device software can also be delivered with all the licensing information included, or might require additional validation and activation phase. An application can be downloaded to a device either via WAP directly from the Internet or first downloaded to a PC and then transferred to a device by Bluetooth or infrared. A license file can be included with the download/application and no additional activation is needed. If the license file is not included with the download, a serial number for activation can be sent to user via SMS or email. The SMS option is the more obvious method for mobile license validation, however it is not secure enough without additional encryption. Mobile operators do not secure SMS channels or encrypt SMS messages, but some independent software vendors did develop their own SMS encryption software. Silicon Village developed service-independent SMS encryption solution - Fortress SMS. The application currently runs only on Nokia's Series 60 and the Symbian platform and it encrypts and decrypts SMS messages on the handset, without using a mobile operator as an intermediary [29]. By implementing solutions similar to this one, SMS can become a secure channel for transmitting sensitive information, including software licenses. Currently if serial numbers are not included with the application, they are either sent to the user by email, or are generated prior to or after the download and the user reads them from the website. This can be once-off validation of the license file/serial number, so when the application is started in the future no additional validation needs to be done. Some applications require subsequent validation, each time the application is used the license file/serial is checked with the licensing server and usage is granted or denied.

Another issue associated with license files and license delivery is security. Licensing information has to be delivered to the end user over the network and stored on the mobile device–both delivery and storage raise security concerns. The license has to be delivered in such a way that it cannot be intercepted and misused. The server/vendor has to validate the return information that the user actually did receive the license. Also once licensing information is stored on the device, unauthorized alterations to the file should be disabled. Licenses delivered over the WAP are encrypted in security layer of the protocol (Wireless Transport Layer Security) [7] so this is a safe way of delivering the licensing information. Licenses downloaded to a PC over the Internet are also encrypted using SSL. Another channel that can be used to deliver licensing information to customers is SMS messaging, however text messages are not encrypted and can be easily intercepted.

A licensing system also has to ensure that no unauthorized modifications are done to the licensing file once it is stored on the mobile device. One way to protect the file is to encrypt the information in it that would be decrypted while performing license check. A problem with this is that U.S. government treats high encryption technologies as protected ones and has various export restrictions on them, and neither vendors nor customers want additional export problems with licensed/purchased applications. This problem can be solved using MAC (Message Authentication Code) checksum of the information in the license file that would be checked every time information from the license file is read and detect if changes have been done to a file [31]. The contents of the file are passed through an algorithm to generate the MAC checksum. If the checksum of the content of the file does not match the original checksum generated with the original file, the license is denied. Another issue related to updating license file is non-repudability of software usage. The licensing mechanism should ensure that software usage recorded in a license file cannot be later repudiated by the user.

Serial codes for mobile software licenses are currently being generated using IMEI or MSISDN as device IDs. Identification of the device/user on the network is an open issue in mobile software licensing. Should the license owner be a device or user? Although currently licenses are issued and fixed to a device (by a hardware id), or to user (by the mobile phone number), these models are inflexible in cases when users change the handsets or phone numbers. These situations are not rare–mobile users upgrade mobile devices very often, every year or two, because mobile device industry is advancing really fast. Users cannot transfer and use their old software licensed by the old device's IMEI on the new device, and all the software has to be purchased again. A similar situation is with the software licensed by phone number. Mobile users often switch from one mobile network to another because of the difference of cost or availability of services on the network. Once user changes the network, all the software licensed by the mobile phone number becomes unusable to that customer. Future licensing systems for mobile devices should solve the problem of the license ownership, license transfer between users and devices, and device/user identification on the network.

An issue tightly connected with license delivery and management is the processing the payment for these licenses. Payment methods have to fit well in the notion of the mobile computing. Currently there are several ways payment for the software for mobile devices is done. Credit card payment was inherited from e-commerce solutions because mobile applications are in majority of situations downloaded from the web. The user registers for the download, enters credit card details, and proceeds to download the application either directly to a mobile device or to a PC. This method of payment involves the customer and the software publisher or a third party distributing the software, without involving the mobile operator. Payment for the mobile software can also be done by debiting the price of the software from the customer's pre-paid amount with network operator or by adding the price to the customer's monthly bill. SMS can also be used as means of mobile payment. One way is by reverse SMS billing, where recipient (customer) pays for the charges of SMS sent by the software vendor, where charges are equal to the price of the software. Another way is premium SMS messaging where customer sends SMS to a certain number vendor/distributor specifies where the price of sending that SMS is equal to the price of the software. These methods should be available for both first time purchase of the software as well as license renewal. The license manager should enable users to renew a license after it is expired, possibly with smaller renewal charges than the initial purchase. Payment for the enterprise applications is different because it involves larger sums of money and is done depending on the often customized agreement between enterprise and vendor with money transfer usually by invoice payment.

7 Possible Solutions for new Licensing Systems

Most of the actors in the industry agree that if growth of mobile applications use is to remain steady, changes in the current licensing models are needed. Licensing should be more pay-per-use oriented, either prepaid or postpaid, similar to the system used to charge mobile phone customers for telecom services (phone calls, SMS, wireless access to the Internet etc.) In this case, download or installation of the software would not be charged for, but customers would be charged every time they use the software. This way ensures customers pay only for software they use and don't waste money on the unlimited licenses they used only a few times and do not require anymore. Pay per use software billing should be more profitable for vendors as well, because on the applications customers keep using, with regular charges for the software usage, they could potentially earn larger revenues than those they would obtain by just charging for the downloads. Introducing pay per use model should also increase the number of customers using mobile software applications. Initial step of starting to use the application is a lot easier, less costly, and less risky. Customers are reluctant to pay 5, 10, or 20 EURO for the application for which they are not sure if it will suit their needs. With pay per use licensing model they can download the application for free and pay 5 or 10 cents per time they ran the application, or be charged per minute/hour of using it. This way, if they do not want to continue using the application, they are not at loss. Another benefit of pay per use model for software providers (independent vendors, network operators, device manufactures) is its flexibility. Software providers can charge per use of the service (time application is run), per use of feature of the application (in case of the games this can be per every new level user achieves), or subscriptions, with or without additional pay per usage. Pay per use licensing model has a marketing value as well, because it enables providers to gain statistics on software usage /time and length of software usage, gather personalized data on their subscribers so they can improve the products and promote new ones to the customers in the areas of their interest. This model of software licensing will in the future be incorporated with existing billing systems. Currently charges for software downloads can be added to users mobile phone bill, or subtracted from the prepaid amount, or in the case software isn't distributed by network operator, billing can be done via reverse SMS billing or by credit card. However, research shows that customers feel more comfortable with paying over the mobile phone than sending credit card details. Research firm Frost & Sullivan, has found that consumers feel it is more secure to be adding purchases to their mobile phone bills than disclosing their credit card details [30]. This is possibly due to the fact that mobile billing is a new concept and there was no fraud associated with it yet, while credit card fraud is something consumers are often faced with.

Another condition for success of any new licensing model for mobile devices is possibility of easy integration with existing licensing systems on both vendors' and customers' side. On the customers' side it has to be addressable by asset management systems to analyze purchased licenses against their actual usage to determine if the company is over-licensed or there is a shortage of licenses. On the vendors' side software and licenses should be distributed through existing distribution channels, billed in familiar way, integrated with customer relation management systems.

In order to implement a value-based pay per use licensing and billing system, an architecture is required that would track the usage of the application on the mobile device (client) and to report the usage to the server. Client should collect data on usage both when the device is connected to the network and when it is disconnected, and then report the usage data on the first following connection.

Echovox developed a mobile commerce technology for pay-per-use applications on Microsoft's Smartphone platform. The technology, called SmartPay is based on specific m-payment APIs which developers integrate into their applications and enables developers to bill their application usage per use. Customers download the application and are able to test it a certain number of times without any fee. After the user uses up all the free trials, the application prompts the user for payment. If the user accepts the transaction, the application sends info to the server via SMS, web services or telephony, the server generates the unlock sequence (license) and sends it back to the device. Transaction costs are added to the customer's mobile phone bill. Echovox however does not provide any information on their website if any software companies have implemented this solution yet. A big limitation of this system is that it is developed only for Microsoft's Smartphone platform. Also, from the information provided it seems that the application bills only for unlocking certain products/features, and not actually for the time the application is used or number of times the application is ran, therefore not truly implementing pay per use model. It is also unclear which programming language(s) this API can be integrated with [3].

Tira Wireless has launched the initiative for developing an additional Java API that would be used to implement pay per use billing (Java Specification Request JSR 190) Java is currently the most popular language used to develop the games for mobile platforms, in particular for the Symbian OS. This API would enable the pay per use licensing model implementation for Java mobile applications. It will standardize the tracking of application events on a mobile device and the submission of these event records to an event-tracking server via a standard protocol (e.g. HTTP). A final release was scheduled for spring 2003 but is not out yet [9]. The real value of such an API won't be known until it is released and a few applications test and implement it, but some already argue that it won't provide a solution for pay per use licensing model industry needs. Since it is only aimed at Java applications and is an optional API, it might be inflexible and cause further fragmentation of market, rather than introduce a licensing standard. Also, an argument against this solution is that the management of the billing event records needs to be outside of the application itself, ideally integrated with license management. Concern exists that the potential need to do code injection will complicate code signing and application deployment. Another issue raised is if there is need at all for such an additional API when standardized web services can be used to perform event tracking and report to the server. A counter argument is that this API is using a light weight protocol so there is less traffic between the client and server, which is reflected in less expensive and faster transactions [9].

8 Conclusion

Licensing systems used today in distributed computing have several flaws that future developments are aiming to solve. Issues to be addressed include license management for enterprise applications, support for pay per use licensing systems, license delivery, security, support for multiple license models and methods of payment for mobile applications. Future licensing systems will be capable of tracking the usage of software and charge users based on the amount of time they've used the application–either prior to the actual application usage (pre-paid, subscription model) or after the application use, and will be integrated with mobile billing technologies. Licensing systems will be developed separately from the software application, probably not by the same vendors either, but will be incorporated with vendors' applications and with billing system (that might or might not include the network operator). These licensing systems are meant to slash the unauthorized use of the software (intentional or unintentional resulting from misunderstanding of complicated licensing terms), and make the applications available to a wider range of the audience by introducing flexible licensing models that meet the different needs of customers. Whether or not the legal systems of many countries will allow such restrictions on the access and use of software is a question that has yet to be answered.

References

[1] AvantGo. http://www.avantgo.com.

[2] Bango.net. http://www.bango.net.

[3] EchoVox SmartPAY. http://www.echovox.com/SmartPAY.html.

[4] Extended Systems. http://www.extendedsystems.com/.

[5] FLEXlm License Models. http://www.macrovision.com/solutions/esd/flexlm/lmmodels.shtml.

[6] Fone Arcade. http://www.fonearcade.com/.

[7] How secure is WAP with SSL and WTSL? *123 WAP Info.* http://www.123wapinfo.com/Faqs/Security/Default.asp?4.

[8] Isogon License Management. http://www.isogon.com/license_management.htm.

[9] Java Specification Request JSR 190. Event Tracking API for J2ME. http://www.jcp.org/en/jsr/detail?id=190.

[10] Microsoft SmartPhone. http://www.microsoft.com/windowsmobile/products/smartphone/default.mspx.

[11] Mobile phone glossary of terms. http://ucables.com/docs/mobilephones/gsm/glossary.htm.

[12] Nokia Software Market. http://www.softwaremarket.nokia.com.

[13] PalmSource Palm OS. http://www.palmsource.com.

[14] PhoneGames4U. http://www.phonegames4u.com.

[15] Selatra. http://www.selatra/com.

[16] Symbian OS. http://www.symbian.com.

[17] Vodafone Live. http://www.vodafone.ie/live/.

[18] Windows CE Microsoft. http://www.microsoft.com/windows/embedded/ce.net/default.asp.

[19] Wireless Game Club. http://www.worldcup.com/wgc/.

[20] XSLM Resource Center. http://www.xslm.org.

[21] LockStream Ships First Digital Rights Management Software Designed to Support Games on Wireless. *Business Wire*, March 27, 2003. http://www.hostingtech.com/news/2003/3/28/St_Nitf_LockStream_Ships_First_Digital _b0327035.0sw.html.

[22] Nathan Cochrane. Sydney firm to protect 3G content. *Sydney Morning Herald*, April 15, 2003. http://www.smh.com.au/articles/2003/04/14/1050172523035.html.

[23] KPMG Consulting. Making the case: Transformation to Digital Licensing. 2002. http://www.kpmgconsulting.com/library/pdfs/19.C2364_Dig_Lic_WP_Rev2.pdf.

[24] The Open Group. Systems Management: Software License Use Management (XSLM). March 1999. http://www.opengroup.org/publications/catalog/c806.htm.

[25] Handago. The Handago Yardstick 2002. 2002. http://www.handango.com/pdf/ HandangoYardstick_2002.pdf.

[26] Allen Lau. Value-Based Billing for wireless Java apps. *Wireless Future Magazine*, january/February 2003.

[27] Mathew G. Nelson. Easy to Track Deliveries. *Information Week*, Sept 11, 2002. http://www.informationweek.com/story/IWK20010907S0003.

[28] Clarke Niall. Distributed Software Licensing Framework based on Web Services and SOAP. Master's thesis, Trinity College Dublin, May 2002.

[29] Ilya Poropudas. Fortress offers handset-based SMS encryption. June 09, 2002. http://www.mobile.commerce.net/story.php?story_id=3188&s=5.

[30] Electronic Publishing Services. EPS Statistics File–mobile. 2002. http://www.epsltd.com/IndustryInfo/Statistics/Mobile.htm#UMP.

[31] Richard Mirabella. GLOBETrotter Software. Some Encryption Issues. 2002-2003. http://www.macrovision.com/solutions/esd/articles/art2.htm.

[32] Vince Stanford. Pervasive Computing Goes to Work: Interfacing the Enterprise. *Pervasive Computing*, 2002. http://www.ee.oulu.fi/s̃kidi/teaching/mobile_and_ubiquitous_multimedia_2002/pervasive _computing_goes_to_work.pdf.

[33] Mary Welch. Saugatuck Technology. Value Pricing: A Viable Software Pricing Metric? July 18, 2002. http://www.saugatech.com.