Mobile OS - Features, Concepts and Challenges for Enterprise Environments

Thomas Renner
SNET Project
Technische Universität Berlin
thomas.renner@campus.tu-berlin.de

Abstract—**The impact of mobile devices in our life is increasing continuously. Especially the integration of mobile devices into an enterprise environment is a hot topic these days. The following paper should give an overview about the state of the art of features, concepts and challenges of mobile operating systems to integrate them into an enterprise environment. Therefore it is necessary to show what kind of policies are needed for using a mobile device in an enterprise environment and to analyse how currently available mobile operating systems and their specific policies and security mechanisms can fulfill these requirements.**

I. Introduction

The worldwide sales of mobile devices, especially for smartphones, grew by an increasing rate over the last years. Gartner says that 304 millions of mobile device units where sold in the year 2010, which is an increase of 72,1 percent in comparison to the year 2009 [1]. The increasing popularity and capability of mobile devices and the confides of organisation to integrate them into their business processes represents an attractive target for criminals to attack [2]. As a consequence, organisations need to implement policies to manage the risk of using mobile devices in an enterprise environment, especially when the data that the mobile devices are handing is sensitive and confidential. The following paper should give an overview about features, concepts and challenges to ensure specific policies for a safe integration of mobile devices into an enterprise environment.

II. Enterprise Policies

Campbell defines Policies as ”guidelines that regulate organisational action. They control the conduct of people and the activities of systems” [3]. This regulation is necessary to specific how employees or applications have to operate in situations to avoid the exposure of private or confidential information due to unintended handling of a device or software.

An overview about typical policies and challenges for managing mobile devices in an enterprise environment is illustrated in table I, which is derived from the IT-Governances framework COBIT [4].

Challenges	Policies
A lost or stolen mobile device	Implement a central management console for device remote control - i.e., location tracking, data wipe-out, password/PIN change or user strong authentication
Enforcing the enterprise policy for standard devices	Gain visibility of all devices connected to the infrastructure
Providing support for various devices	Turn to cross-platform centrally managed mobile device managers
Controlling data flow on multiple devices	Secure the systems that are accessed with authorization, encryption and privileges control
Preventing data from being synchronized onto mobile devices in an unauthorized way	Monitor and restrict data transfers to hand held or removable storage devices and media from a single, centralized console

TABLE I
Examples of policies and challenges

To enforce policies for mobile devices in an enterprise environment, is a complex, but also a required task for organisations. It can not be compared with the enforcement of other usual items in the IT world, because of the property of high portability of the mobile devices. The property portability signifies, that the device is used anytime and -place and can easily get lost or stolen, which describes a bad scenario, if it for example stores sensitive corporate data locally [5]. But the threat of losing a device is not the only risk of using a mobile device in an enterprise environment. Due to the fact that it is very portable and is used as a mobile interface to enterprise communication backends, mobile devices communicate via wireless networks, which are less secure than wired networks [6]. Because of that, it is necessary to develop convenient policies to realize such a secure data exchange, which is not an easy task, because many vendors develop products primarily for the consumer market [5].

An additional challenge is, to develop a way to enforce convenient policies, so that owners of mobile devices can use them in both private and business environments. This means for organisations, that there is a mixed enterprise and private environment and no explicit enterprise environment [2]. The problem of these mixed environment is, that an enterprise environment needs very strict polices to ensure a high security standard for using a mobile device for business purposes. On the other hand, in a private environment the user does not want to be restricted by using the device. For Example, he wants to install 3rd party applications, which could be forbidden because of the enterprise policies. To forbid the private use of a mobile device completely is no good solution, because this could lead to the situation that the employee does not use the device and leaves it at home. The Advantages of using a mobile device in an enterprise would be unused, because of no acceptance of the employee.

General threats and risks of using a mobile device in an enterprise environment can be summarized as following:

- Mixed private and sensitive corporate data stored on the device
- Sensitive data stored locally on the device can be stolen, i. e. though stealing the device
- Sensitive data exchange, i. e. e-mail, contacts and calendar synchronization, can be read by 3rd persons through using insecure technology
- Installing 3rd party code for private usage by the employee, i. e. due using no or insufficient policy settings.

To control the shown challenges and develop suitable policies, it is necessary to manage the mobile operating system, which provides security mechanism and the ability to set security settings on the mobile device via policies [5]. Therefore the next chapters will give an overview of the different available mobile operating systems and their specific security policies.

III. Mobile Operating Systems

An operating system is a system software, that is designed to operate and control the computer hardware [7]. The operating system operates on a mobile device and is called mobile operating system. An overview about different mobile operating systems, that are currently available on the market is shown in Fig. 1 [8].

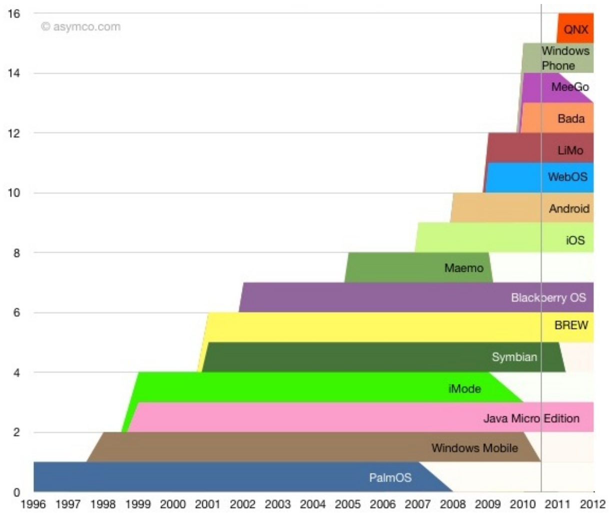

Fig. 1. Mobile Operating Systems Overview

In total it shows up 16 different mobile operation systems, whereof twelve are actual on the market or will be released in an expected time. An exception is the operation system MeeGoo of the companies Nokia and IBM, which future is unknown after Nokia announced that they will use the operating system Windows Phone for their new devices [9]. The graphic also illustrates a high fragmentation of the mobile operating systems, which also shows a challenge of developing standard policies for organisations, because employees are able to choose their own mobile devices, so that the management has to know the specific properties of each operating system. An approach to solve these problem could be a limitation of the number of allowed mobile devices or operating systems in an organisation, what on the other side means a lose of freedom for the employee and can lead to a lower acceptance for using a mobile device in an enterprise environment.

Because of the high number and distribution of varying mobile operating systems, it is not possible to analyse each system within this paper. In reason to that, it is necessary to limit their number and analyse only the most important once, which are derived from the mobile operating market distribution of the last four years shown in Fig. II.

Year	Symbian	Android	BlackBerry OS	iOS	Microsoft	other OS
2010 [1]	37.6%	22.7%	16.0%	15.7%	4.2%	3.8%
2009 [10]	46.9%	3.9%	19.9%	14.4%	8.7%	6.1%
2008 [11]	52.4%	0.5%	16.6%	8.2%	11.8%	10.5%
2007 [11]	63.5%	N/A	9.6%	2.7%	12.0%	12.1%

TABLE II
Market Share of Mobile Operating Systems

The collective notion Microsoft includes the mobile operating systems of the same-named company and is found in the fact, that the operating system Windows Mobile was replaced by Windows Phone in the year 2010. Furthermore Nokia announced, that they will use Windows Phone in the future for their new devices [9]. This announcement of Nokia, the developer of Symbian, can be interpreted as a stop for a further development of the mobile operating system and a lose of relevance for Symbian in the future, wherefore Symbian will not take into consideration in the paper.

Despite the fact that the operating systems of Microsoft lost market share in the last years, it will be part of the analyse in this paper. The reason for that is the cooperation between Microsoft and Nokia [9], the largest manufacturer for mobile devices [1], which can be rated as a high potential for a market share growth in the future. Furthermore Gartner predicts that Windows Phone will be the mobile operating system with the second most market-share behind Android in the year 2015 [12].

A. Mobile OS: Android

Android is an open source operating system for mobile devices developed by Google and the Open Handset Alliance [13]. With 22,7% it is the second most used operating system for mobile devices worldwide behind Symbian [1]. The system architecture consists of [14]:

- a modified **Linux Kernel**

- open source **Libraries** coded in C and C++
- the **Android Runtime**, which considers core libraries that disposals the most core functions of Java. As virtual machine it uses Dalvin, which enables to execute Java applications.
- an **Application Framework**, which disposals services and libraries coded in Java for the application development
- and the **Applications**, which operate on it

In an execution environment, local code is executed with full permission and has access to important system resources. On the other hand, application code is executed inside a restricted areas called a sandbox. This restrictions affects some specified operations such as: local file system access or invoking applications on the local system [15]. Sandboxing enforces fixed security policies for the execution of an application. Isolation an application into a sandbox brings more security and stability, because applications have only access to the core operating system in controlled and required areas. The goals of application sandboxing are to protect applications from each other and to protect the underlying operating system from malicious applications [5].

Android, iOS [16] and Windows Phone [17] use the same model of application sandboxing, which is shown in Fig. 2. Each application owns a unique identity and any data, process or permission belongs to it. For example, the data assigned to one application identity has no access to any other data of another applications identity. This sandboxing model will discussed closer in the following by using the example of Android, to underly the understatement for it.

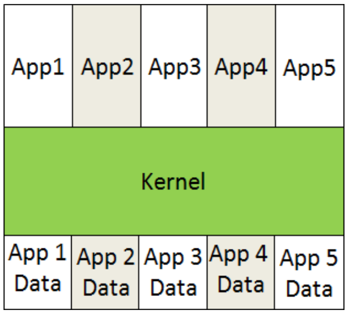

Fig. 2. Android, iOS, Windows Phone Sandboxing Modell

Like previously mentioned in the Android system architecture overview, is Android mainly based up on a Linux kernel and Java. This combination brings up some secure features, like efficient shared memory management, preemptive multitasking, Unix user identifiers (UIDs) and file permissions with the type safe concept of Java. Every Android application runs in a separate process under a unique UID with distinct permissions, which means that applications can typically not read or write each other's data or code. The kernel sandboxes applications from each, so that resource and data must be share explicitly. To make a resource share between applications possible, the permissions which are required must be declare statically at the time the application is installed. The Android system prompts the user for consent at this time, a mechanism for granting permission dynamically at runtime is not possible and would lead to an increase of security transparency [18]. Currently there is a high fragmentation of the different available versions of Android, seen in Fig. 3 [19]. The reason for that high platform version distribution is, that Android is an open source operating system and runs on many different mobile devices, which differ each other in size, form and other technical conditions, like number and function of hardware buttons [20]. To give various manufactures the possibility to conform Android to these factors, it is possible to change the native user interface of the mobile operating system [21]. The disadvantage of these possibility is, that manufactures have to conform the customize mobile operating system again, if an Android update is available. These leads to high costs for the manufacturer, which they probably try to avoid and stay focus on developing new mobile devices. Therefore many mobile devices run with an outdated platform version [22], which also effects the provided policies. Especially if an organization wants to support more than one Android device, these high fragmentation has impacts to the enforcement of policies in an enterprise environment. The reason therefore is, that every version comes with a different API Level and provides for example a different number of polices, which will be discussed closer in the next chapter.

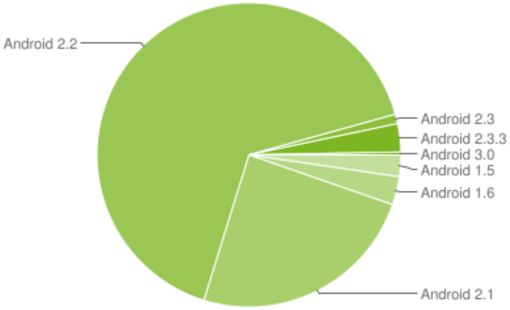

Fig. 3. Android Versions Distribution

B. Mobile OS: BlackBerry OS

Blackberry OS is the proprietary mobile operating system, developed by the Canadian company Research in Motion and is used for Blackberry devices only. Instead of all the other regarded mobile operating systems, it is mainly developed for business usage. Gartner says that it is one of the most popular mobile operating system today with 16,0% market share, but they also predict a decreasing relevance in the future [12].

BlackBerry OS uses an older model for application sandboxing, which can be seen in Fig. 4. It uses different trust roles for assignments and applications have full access to the complete device and data. It is also requiring to sign an application via Certificate Authorities (CA) or generated (self-signed) certificate to run code on the device [23]. Furthermore the signature provides information about the privileges for an application, which is necessary because applications have

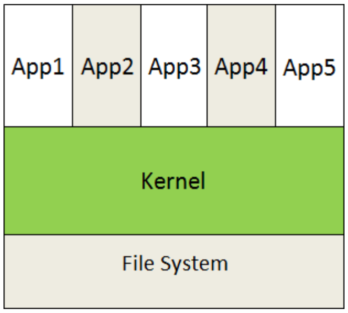

Fig. 4. BlackBerry OS Sandboxing Modell

full access on BlackBerry devices, because of its sandboxing model [5].

C. Mobile OS: iOS

The proprietary mobile operating system iOS is only used for Apple devices like the iPhone and is a further development of the operating system Mac OSX. Its market share grew continuously over the last year to 15,7%.

The system architecture is identical to the MacOSX architecture and consists of the following components [24]:

- **Core OS:** The kernel of the operating system
- **Core Services:** Fundamental system-services, which are subdivided in different frameworks and based on C and Objective C. For example offers the CFNetwork Framework the functionality to work with known network protocols.
- **Media:** Considers the high-level frameworks, which are responsible for using graphic-, audio- and videotechnologies.
- **Coca Touch:** Includes the UIKIT, which is an Objective-C based framework and provides a number of functionalities, which are necessary for the development of an iOS Application like the User Interface Management

Like in the Android section mentioned, iOS uses a similar sandboxing model [16]. Furthermore applications must be signed with an issued certificate. This ensures that application have not been manipulated and ensures the runtime to check if an application has not become untrusted since it was last used. Uneven Android applications, iOS applications can only be signed with an official certification [25].

D. Windows Phone

Windows Phone is a successor of the operating system Windows Mobile of the software developer Microsoft. By comparison to the other discussed mobile operating systems, the market share is low with only 4,2%. But like in the previously chapter mentioned can the cooperation between Nokia and Microsoft be rated as a high potential for a market share growth [9].

Windows Phone uses technologies and tools, which are also used in the station based application development, like the development environment Visual Studio and the Frameworks Silverlight, XNA and .NET Compact. Furthermore Windows Phone considers a complete integration with the Microsoft Services Windows Live, Zune, Xbox Live and Bing [26]. For sandboxing Windows Phone uses the same model like Android and iOS [17]. Furthermore 3rd party applications can only be signed with an official certification, like iOS Applications [17].

The following chapter will show the currently available policies of the discussed mobile operating system. Afterwards there will be a summarized comparison of an all mobile operating system policies.

IV. Mobile Operating System Policies

Like in second chapter mentioned, Policies do ”control the conduct of people and the activities of systems” [3] and are necessary to specific how employees or applications have to operate in situations to avoid the exposure of private or confidential information due to unintended handling of a device or software. There is different between native policies, policies that are provided by the operating system and policies that are only work with a server solution. The following subchapters will show up the policies of each specific mobile operating system. Afterwards there will be a summarized comparison of all offered mobile operating system policies.

A. Android Policies

Since version 2.2 Android offers an Device Administration API, that provides device administration and control features at the system level. With this API, developers can write device management applications that enforces policies. These policies could be hard-coded into the application, or fetched dynamicly from a third party server. The API supports following native policies, which are segmented into needed platform version for support [27].

Android 2.2 or higher:

- Password enabled
- Minimum password length
- Alphanumeric password required
- Maximum failed password attempts
- Maximum inactivity time lock
- Prompt user to set a new password
- Lock device immediately
- Wipe the device’s data (restore the device to its factory defaults)

Android 3.0 or higher:

- Complex password required
- Minimum letters required in password
- Minimum lowercase letters required in password
- Minimum non-letter characters required in password
- Minimum numerical digits required in password
- Minimum symbols required in password
- Password expiration timeout
- Password history restriction
- Maximum failed password attempts
- Maximum inactivity time lock
- Require storage encryption

The Device Administration API contains: ”If a device attempts to connect to a server that requires policies not supported in the Device Administration API, the connection will not be allowed" [27]. This means, if a device does not support the required policies, there is no way to ensure the policies on the device. This is a huge disadvantage, because of the high platform version fragmentation and the fact, that only 0,6% of all Android devices run on version 3.0 or higher in June 2011 [19]. As shown in the polices list, this implies that there is no complex password security and storage encryption policy for most of the Android devices by default. But especially the missing storage encryption is a significant need for using a mobile device in an enterprise environment, because if the device is get stolen or lost, sensitive locally saved data are saved in clear text and easy accessible for 3rd persons.

Android also provides an official application for using a mobile device in an enterprise environment. This application, named Device Policy, is for business and education customers only, but contains only Android version 2.2 supported policies [28]. But in the android market can also be finde 3rd party solutions like: Afaria, Good for Enterprise or WaveSecure [29].

B. Blackberry OS Policies

As a classic business mobile device manufacturer, Blackberry provides over 400 policies for their mobile operating system, which can be used to control specific mobile IT policies in an enterprise environment [30]. Otherwise to Android they can only be enforced in combination with their BlackBerry Enterprise Server solution, which architecture is described closer in the fifth chapter.

The over 400 policies provided by BlackBerry offer everything that is necessary to use a mobile device securely in an enterprise environment and can be categorized into the following groups [31]:

- **Group IT Policies**, simplifies the creation and modification of group policies to ensure the data security and access in an organisation.
- **Default IT Policy**, to ensure a minimum level of security the BlackBerry OS uses a customizable base IT policy set. Administrators can create and modify policies of users or groups to meet the security needs of the organisation by using the BackBerry Enterprise Server.
- **Over the Air Enforcement**, all policy settings are synchronized and assigned to the BlackBerry device. So Administrators can easily change policies without requiring the users acceptance or changes on the device itself. As well, policies carry unique digital signatures to ensure that only the BlackBerry Enterprise Server can send updates to a BlackBerry device.
- **Malware Control**, the BlackBerry Enterprise Server comes with 19 application policies, that allow the administrator to limit the resources and user data available to a given application. For example, limitation can be imposed on internal or external domains, the phone, Bluetooth and user data such as email and Personal Information Management (PIM). Because limitations can all be specified on a single application, administrators can also grant elevated permissions to trusted applications, like a Customer Relationship Management (CRM) application.
- **Comprehensive Control over the Entire BlackBerry Enterprise Solution**, gives administrators the capabilities to: Forcing password use, password complexity and time-outs, Application availability, policy change notification and many more options to control the usage of a mobile device in an enterprise environment.

BlackBerry provides a very considerable policy solution with the disadvantage, that the Blackberry Enterprise Server is necessary to enforce them. This means for an organisation, that they have to use Research in Motion devices and can not support a multiple solution with using different mobile devices running on different mobile operating systems.

C. iOS Policies

The iOS provides some policy protection that can be delivered and enforced over the air or locally. This enforcement can be controlled and configured by using 3rd party Mobile Device Management solutions, wherefore Apple provides a separate API [32].

After an iOS device is registered with the Mobile Device Management Server, the device can be control dynamically. Therefore the Server sends XML configuration profiles to the device, which enables a secure use in an enterprise environment. The configuration profile contains: device security policies and restrictions, VPN configuration information, Wi-Fi settings, email and calendar accounts, and authentication informations, that permit to work with the enterprise system. Supported device security policies and restrictions are shown in the flowing lists.

Policies [32]:

- Require passcode
- Allow simple value
- Require alphanumeric value
- Passcode length
- Number of complex characters
- Maximum passcode age
- Time before auto-lock
- Number of unique passcodes before reuse
- Grace period for device lock
- Number of failed attempts before wipe
- Control Configuration Profile removal by user

Restrictions [32]:

- Application installation
- Camera
- Screen capture
- Automatic sync of mail accounts while roaming
- Voice dialing when locked
- In-application purchasing
- Require encrypted backups to iTunes
- Explicit music and podcasts in iTunes
- Safari security preferences
- YouTube
- iTunes Store

- App Store
- Safari

Additionally the Mobile Device Management server can operate some **actions** on the iOS device, like:

- Remote Wipe: A remotely delete of all data on the device, restoring to factory settings
- Remote lock: The Server locks the device and requires the device passcode to unlock it
- Clear passcode: Enables the user to create a new password, if he forgot the old one.
- Configuration and Provisioning Profiles: Give the ability to add and remove configuration profiles remotely

An advantage of iOS is, that the it operates only on a few Apple devices, which means that there is no high platform version distribution like on Android. Unlike in Android, mobile operating system updates effect every iOS device.

D. Windows Phone Policies

The Windows Phone operating system was primarily designed for the customer and not the enterprise market and provides no native policy settings, like Android or iOS. But it provides support for ActiveSync policies, which are policy settings used by an Microsoft Exchange Server, also a Microsoft product [33]. The Windows Phone successor Windows Mobile supported all Exchange ActiveSync policies, whereby Windows Phone supports only some basic policies like [33].

- Password Required
- Minimum Password Length
- Idle Timeout Frequency Value
- Allow Simple Password
- Password Expiration
- Password History

How policy enforcement over an Exchange Server works is discussed in the chapter 5. But some Exchange ActiveSync policy settings are also supported by other mobile operating systems and can be seen in the following subsection, which provides a comparison of the mobile devices and their supported policies.

E. Mobile Operating System Polices Comparison

By comparing the policies of the different mobile operating systems to each other, it is noticeable that the BlackBerry OS and iOs provides a larger number than the other two, especially the BlackBerry OS with more than 400. This could justified because both are the oldest available on the market and especially the BlackBerryOS is developed for business cases. The largest number of policies doesn't mean the best solution for an enterprise environment shows an scenario of the Deutsche Bank. ”We found enterprise email on iPhone was a fantastic experience as it was easier and faster to access data than on the Blackberry,” said Chris Whitmore, an analyst at Deutsche Bank. ”It was also great to only have to carry one device for personal and corporate email access” [34]. This shows that beside the number of policies the infrastructure quality, acceptance by the employee and mixed private and business environment plays a role by choosing the right solution.

Also it is noticeable that Windows Phone and BlackBerry OS do not provide any native policies. The policy enforcement on these mobile operating systems can only be accomplished by a offered server solution, BlackBerry Enterprise Server or Exchange ActiveSync. Android and iOS however provide a number of native policies, which means that organisations for example can develop their own mobile device management solution to control multiple mobile devices of different mobile operating systems.

A deeper look on the policies of Windows Phone also confirmed the statement that the mobile operating system is primarily designed for the customer market [33]. It offers the fewest number of policies of all mobile devices. So it is necessary that Microsoft releases an update to improve their mobile operating system to establish a good solution for using it in an enterprise environment. In my opinion this is important and a big opportunity for Microsoft, because they are a big player in the fixed enterprise environment with their established products.

Android provides a smaller number of policies in comparison to iOS and BlackBerryOS, but they provide the important one's like a remotely wipe or password usage. But a big problem for Android in the future will be the high distribution of their platform versions, so that updates that effect policies will be not available or with a long delay for Android devices.

An overview about which policies are supported by the different mobile operating system can be found in table III [35]. The short form EAS stands for Exchange ActiveSync and BES for Blackberry Enterprise Server. A closer look behind these two technologies is part of the next chapter, which discusses the policy enforcement for mobile operating systems.

Policy	**Android 2.2 or higher**	**Android 3.0 or higher**	**BlackBerry OS**	**iOS**	**Windows Phone**
Enforce Password	yes	yes	BES	yes	EAS
Complex Password	no	yes	BES	yes	no
Remote Lock	yes	yes	BES	yes	EAS
Remote Wipe	yes	yes	BES	yes	EAS
Storage Encryption	no	yes	BES	yes	no
Restriction	no	yes	BES	yes	no
Manage over Air	yes	yes	BES	yes	EAS
EAS number support	9	13	noone	14	7

TABLE III
MOBILE OPERATING SYSTEM POLICIES

V. Policy Enforcement

The following chapter describes mechanisms and instruments to ensure policies, which have been introduced in the previous chapter.

A. Exchange ActiveSync

Exchange ActiveSync is a Microsoft product and a ”proprietary protocol that uses Extensible Markup Language (XML) over Hypertext Transfer Protocol (HTTP) to synchronize Microsoft Exchange data to clients” [36]. It's main function is it to provide a secure synchronization for e-mails, contacts and calendars, but it can also be used to monitor mobile devices. Therefore the administrator can set up device policies, which are enforced and synchronized over a network on the device [36]. Like in Table III shown, do almost all discussed mobile operating systems support some of the Exchange ActiveSync Policies.

B. BlackBerry Enterprise Server

BlackBerryOS does not support any Exchange ActiveSync Policies and uses his BlackBerry Enterprise Server solution for enforcing the over 400 policies on their devices. The following paragraph will describe how the data exchange between the mobile device and BlackBerry Server works, which is necessary to enforce and synchronize the policies of an network connection. The data exchange of the BlackBerry Enterprise Solution complies the strong requirements of government authority and is certified by the NATO [37]. BlackBerry offers two transport encryption options, Advanced Encryption Standard (AES) and Triple Data Encryption Standard (Triple DES), for data exchange between the BlackBerry Enterprise Server and the BlackBerry device. The private encryption keys, who are necessary for the encryption, are generated in a secure, two-way authentic environment and are assigned to the Blackberry device. Each secret key is stored on the users secure enterprise account and on the BlackBerry device. Corporate data, like e-mails, sent to the BlackBerry device is encrypted by the BlackBerry Enterprise Server using the private key. The encrypted data will be transfered securely across the network to the device, where it is decrypted with the other stored private key. Data remains encrypted in transit and is never decrypted outside of the enterprise environment [23]. An overview about how the data exchange between the BlackBerry Enterprise Server and the BlackBerry device works can be seen in Fig. 5.

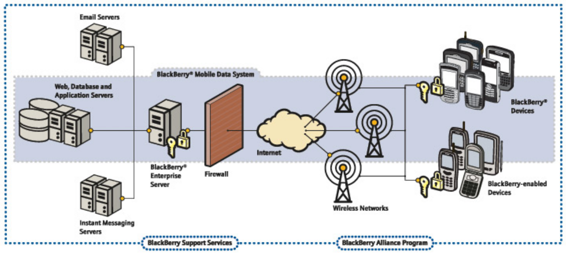
Email Servers
BlackBerry® Mobile Data System
Web, Database and Application Servers
BlackBerry® Enterprise Server
Firewall
Internet
Wireless Networks
BlackBerry® Devices
BlackBerry-enabled Devices
Instant Messaging Servers
BlackBerry Support Services
BlackBerry Alliance Program

Fig. 5. BlackBerry Enterprise Server and Device Data Exchange Model

C. Apple Mobile Device Management Server

In contrast to the BlackBerry solution, Apple provides an API for developer to create an own solution for monitoring mobile devices and ensure policies. The basic structure of a 3rd party mobile device management server shows Fig. 6 [32].

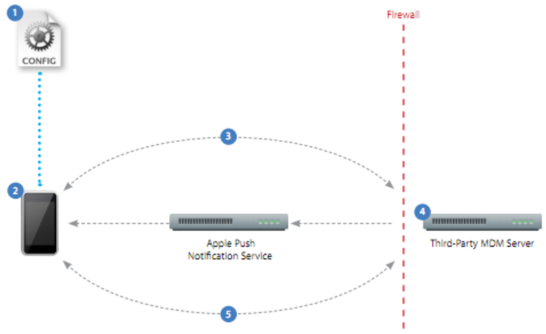

Fig. 6. Figure Caption

1) A Configuration Profile containing mobile device management server information is sent to the device. The user is presented with information about what will be managed and queried by the server.
2) The user installs the profile, so that the device can be managed.
3) Device enrollment takes place as the profile is installed. The server validates the device and allows access.
4) The server sends a push notification prompting the device to check in for tasks or queries.
5) The device connects directly to the server over HTTPS. The server sends commands or requests information

Configuration profiles can be signed and encrypted. Signing a configuration profile ensures that the settings it enforces cannot be changed. Encrypting a configuration profile protects the profiles contents and permits installation only on the device for which it was created. [25].

For an protected exchange of corporate data the iOS provides technologies like: VPN (IPsec, L2TP and PPP), SLL/TLS and WPA/WP2 [25].

D. Visualization

Visualization means the ability to run multiple instances of a operating system on a system, like a mobile device, by using one modified system kernel. Every instance has it's own environment with a specified file system and applications and processes that are assigned to one explicit instance. The main benefit of using visualization is, that malicious or corrupt applications are isolated and have not effect to the other operating system instance [38].

But it also opens new security capabilities for using an mobile device in an enterprise environment. Visualization for example could handle the problem of the mixed private and business environment usage, by running one instance for enterprise purposes with high policies and one instance for private purposes with no or low policies. Both visualized instances run isolated from each and the user can switch between them, without using a second mobile device.

Currently no mobile operating system does support visualization by default, but becasue android is an open source operating system, there are projects enabling it, like L4Android. Therefore the developers implement a microkernel instead of the modified linux kernel, which only provides the necessary functionalities of a kernel. [39]

The disadvantage of using the modified Linux kernel, with about 14 million lines of code, as a monolithic kernel at the lowest layer in the software stack, is that it contains i.a. many device drivers, like Keypad or Camera Drivers. This complies not the requirements of a Trusted Computering Base (TCB), which is the set of all components that are critical to it's security. The idea of a TCB is to handle only components that are very critical to the system security and to keep the size of the TCB as low as possible. The reduction to a small TCB leads to a lose of complexity and higher security. Components like the Audio Driver don't belong to the TCB, because, no matter how insecure they are, they are no threat to the real system security [40]. Using this microkernel could be a good solution for organisations who think about using Android in an enterprise environment. The disadvantage of this solution is, that every mobile device needs to be updated with these modificated software.

E. Application Signing

Application Signing "ensures the integrity of the code downloaded from the Internet. It enables the platform to verify that the code has not been modified since it was signed by its creator" [15].

Android applications are signed with a certificate whose private key is held by their developer. The certificate does not need to be signed by a certificate authority, which means it is allowable and typical to use self-signed certificates. So companies have the opportunity to use their own certificate for their policy enforcement application. Ways to distribute the application to the employees are the official Android market or non-marketing installation via flash drive, email or a website.

Apple's iOS and Windows Phone applications must also be signed, but with an issued certificate. This ensures that application haven't be manipulated and ensures the runtime to check if an application hasn't become untrusted since it was last used . Uneven Android applications, iOS and Windows Phone applications can official only be distributed over their specifically application market [25] [17]. The disadvantage of this solution is, that updates on an self created policy application can not be available in real time and must checked first by apple.

BlackBerry applications require developers to sign and register their applications with Research In Motion. This adds protection by providing a greater degree of control and predictability to the loading and behavior of applications on BlackBerry devices. It is also requiring to sign an application via Certificate Authorities (CA) or generated (self-signed) certificate to run code on the device [23]. Furthermore the signature provides information about the privileges for an application, which is necessary because applications have full access on BlackBerry devices in reason to the traditional sandboxing model [5]. Because policy enforcement is only possible by using the BlackBerry Enterprise Server, the strict application signing solution has no impact on it.

VI. Conclusion

Every mobile operating system supports different policies and policies enforcement, so it is necessary for a organisation to choose the mobile operating systems which they want to deploy. This is not an easy task, like the example of the Deutsch Bank [34] shows, because it is important for the user to use the device in an enterprise and a private environment. In addition to that, there is a high mobile device distribution, which means that every employee has different device preferences. The high device and platform fragmentation and mixed environment leads to an assumption, that a multiple solution like using an Exchange ActiveSync is the best way to use policies in an enterprise environment. But like in the fourth chapter mentioned, not every mobile operating system does support ActveSync in the same volume. Furthermore is the right solution in my opinion a question of organisation size. It won't be the best way for a small organisation to use a complex solution like Exchange ActiveSync. Smaller organisation are probably better advised to use a complete solution like an 3rd Party iOS Mobile Device Management Server, even if they are bound on specific mobile devices. Like in the policies comparison in the fourth chapter mentioned, Android 2.2 offers not so many policies like the BlackBerry OS or iOS. If Android wants to take the advantage of their high market share in the future, they have to find a solution for the high platform fragmentation problem to enforce their policy improvement, which comes with Android 3.0. The iOS has the advantage, that they offer only a very specific number of mobile devices, which provides that there is always a stable version of the device. In addition, iOS provides just as the BlackBerry OS a lot of policies and an elaborate policy enforcement. To support Windows Phone in an enterprise environment can not be recommended, because like in the paper shown, are there not many supported policies. The reason for that is, that Windows Phone is primarily developed for the customer market, but they promised to develop an update, which brings up some more possibilities for using an Windows Phone device in an enterprise environment.

References

[1] Gartner, "Gartner says worldwide mobile device sales to end users reached 1.6 billion units in 2010; smartphone sales grew 72 percent in 2010," Febuary 2011. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1543014

[2] M. Landman, "Managing smart phone security risks," in *2010 Information Security Curriculum Development Conference*, ser. InfoSecCD '10. New York, NY, USA: ACM, 2010, pp. 145–155. [Online]. Available: http://doi.acm.org/10.1145/1940941.1940971

[3] N. J. Campbell, *Writing Effective Policies and Procedures: A Step-By-Step Resource for Clear Communication*. AMACOM, 1998.

[4] ISACA, “Managing mobile devices and relevant framework process,” 2010. [Online]. Available: http://www.isaca.org/Knowledge-Center/Research/Documents/SecureMobileDevice-Chart-21July2010-Research.pdf

[5] H. Dwivedi, C. Clark, and D. Thiel, *Mobile Application Security*. McGraw-Hill Osborne Media, 2010.

[6] ISACA, “Securing mobile devices,” 2010. [Online]. Available: http://www.isaca.org/Knowledge-Center/Research/Documents/SecureMobileDevices-Wht-Paper-20July2010-Research.pdf

[7] D. Abts, *Grundkurs Wirtschaftsinformatik*. Vieweg Friedr. + Sohn Ver, 2004.

[8] H. Dediu, “The lives and deaths of mobile platforms,” Febuary 2011. [Online]. Available: http://www.asymco.com/2011/02/19/the-lives-and-deaths-of-mobile-platforms/

[9] Golem, “Nokia setzt auf windows phone 7,” Febuary 2011. [Online]. Available: http://www.golem.de/1102/81342.html

[10] Gartner, “Gartner says worldwide mobile phone sales to end users grew 8 per cent in fourth quarter 2009; market remained flat in 2009,” Febuary 2010. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1306513

[11] Gartner, “Gartner says worldwide smartphone sales reached its lowest growth rate with 3.7 per cent increase in fourth quarter of 2008,” Febuary 2009. [Online]. Available: http://www.gartner.com/it/page.jsp?id=910112

[12] Gartner, “Gartner says android to command nearly half of worldwide smartphone operating system market by year-end 2012,” April 2011. [Online]. Available: http://www.gartner.com/it/page.jsp?id=1622614

[13] R. Meier, *Professional Android 2 Application Development (Wrox Programmer to Programmer)*. Wrox, 2010.

[14] Android, “What is android.” [Online]. Available: http://developer.android.com/guide/basics/what-is-android.html

[15] D. Chadwick and B. Preneel, *Communications and Multimedia Security: 8th IFIP TC-6 TC-11 Conference on Communications and Multimedia Security, Sept. 15-18, 2004, Windermere, The ... in Information and Communication Technology)*. Springer, 2005.

[16] Apple, “The application runtime environment.” [Online]. Available: http://developer.apple.com/library/ios/#documentation/iphone/conceptual/iphoneosprogrammingguide/RuntimeEnvironment/RuntimeEnvironment.html

[17] Microsoft, “Security for windows phone.” [Online]. Available: http://msdn.microsoft.com/en-us/library/ff402533%28v=VS.92%29.aspx

[18] Android, “Security and permissions.” [Online]. Available: http://developer.android.com/guide/topics/security/security.html

[19] Android, “Platform versions.” [Online]. Available: http://developer.android.com/resources/dashboard/platform-versions.html

[20] J. Mattson, “Casting a wide net: how to target all android devices,” May 2010. [Online]. Available: http://www.google.com/intl/de-DE/events/io/2010/sessions/casting-wide-net-android-devices.html

[21] Android, “Android 2.3 compatibility definition.” [Online]. Available: http://static.googleusercontent.com/external_content/untrusted_dlcp/source.android.com/de//compatibility/android-2.3.3-cdd.pdf

[22] heise, “Eine fragmentierte android-landschaft,” April 2010. [Online]. Available: http://www.heise.de/mobil/meldung/Eine-fragmentierte-Android-Landschaft-977538.html

[23] BlackBerry, “Blackberry security.” [Online]. Available: http://uk.blackberry.com/ataglance/security/

[24] Apple, “ios overview.” [Online]. Available: http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/URL_iPhone_OS_Overview/

[25] Apple, “iphone in business security overview.” [Online]. Available: http://images.apple.com/iphone/business/docs/iPhone_Security.pdf

[26] msdn, “Application platform overview for windows phone,” 2011. [Online]. Available: http://msdn.microsoft.com/en-us/library/ff402531(v=vs.92).aspx

[27] Android, “Device administration.” [Online]. Available: http://developer.android.com/guide/topics/admin/device-admin.html

[28] Android, “Google apps for enterprise: Device policy fr android: bersicht fr nutzer.” [Online]. Available: http://www.google.com/support/mobile/bin/answer.py?answer=190930x

[29] S. L. B. Reimold, “Android security - device management and security.” [Online]. Available: http://sigs.de/download/oop_2011/downloads/files/Mi8-4_Reimold_Linzner_Update.pdf

[30] BlackBerry, “Blackberry enterprise server - policy reference guide.” [Online]. Available: http://na.blackberry.com/eng/deliverables/1417/BlackBerry_Enterprise_Server_Policy_Reference_Guide.pdf

[31] BlackBerry, “It policy.” [Online]. Available: http://uk.blackberry.com/ataglance/security/it_policy.jsp

[32] Apple, “iphone in business mobile device management.” [Online]. Available: http://images.apple.com/iphone/business/docs/iPhone_MDM.pdf

[33] H. Walther, “Exchange activesync considerations when using windows phone 7 clients.” [Online]. Available: http://social.technet.microsoft.com/wiki/contents/articles/exchange-activesync-considerations-when-using-windows-phone-7-\linebreakclients.aspx

[34] K. Asharya, “Iphone use in business boosted by security apps.” [Online]. Available: http://www.mobiledia.com/news/80681.html

[35] G. Gruman, “Mobile management: How iphone, android, windows phone 7, and the rest stack up.” [Online]. Available: http://computerworld.com.edgesuite.net/insider/InfoWorld-MobileManage-DeepDive.pdf

[36] C. Edge, *Enterprise iPhone and iPad Administrator’s Guide (Books for Professionals by Professionals)*. Apress, 2010.

[37] BlackBerry, “Approvals and certifications.” [Online]. Available: http://uk.blackberry.com/ataglance/security/certifications.jsp

[38] H. Muehe, *Virtualisierung - Geschichte, Techniken und Anwendungsfllle (German Edition)*. GRIN Verlag, 2008.

[39] M. Lange and S. Liebergeld, “Taming the robot: Sandboxing androi.” [Online]. Available: http://www.isti.tu-berlin.de/fileadmin/fg214/mlange/taming_the_robot_droidcon11_talk.pdf

[40] A. S. Tanenbaum, *Moderne Betriebssysteme*. Pearson Studium; Auflage: 3. aktualisierte Auflage, 2009.