Virtualization Techniques for Mobile Devices

David Jaramillo[1], Borko Furht[2], Ankur Agarwal[3]

Abstract – *In current mobile system environment there is a huge gap between the personal smart phone and the enterprise smart phone due to the issues related to security, enterprise policies and freedom of use. In the current environment, data-plans on mobile systems have become so prevalent that the rate of adaptation of data plan for the current customers has far outpaced the ability to add new consumers for the mobile service providers. Most of the enterprises require/provide the access of emails and other official information on smart platforms. This presents a big challenge for enterprise in securing their systems. Therefore due to the security issues and policies imposed by the enterprise in using the same device for dual purpose (personal and enterprise), the consumers often lose their individual freedom and convenience at the cost of security. To address this challenge, few solutions have been presented. One effective way is to partition the mobile device such that the enterprise system access and its information are completely separated from the personal information. This paper discusses and presents such approaches for mobile information partition in order to create a secured and secluded environment for enterprise information while allowing the user access to their personal information.*

Keywords: *mobile; container; virtualization; hybrid; security*

I. Introduction

Ever since mobile devices started connecting through wireless data networks, it became important to be concerned about the information that was stored on the device. At first, the devices were not very powerful and did not have much function except basic email, calendar and contacts. Much of this information was of personal nature. Due to the advances in the semiconductor technology, these devices started offering the computation power comparable to an older generation computer. Enterprises immediately embraced these platforms in order to enhance the overall productivity and communication among the team members. However, it is extremely important to provide a secure way of enterprise information access to a mobile device in such a manner which would comply with their policies. This would require an enterprise level device management and policy enforcement. Today, there is a diverse number of mobile operating systems such as Android, BlackBerry, Brew, iOS, PalmOS, Symbian and Windows Mobile among others, that are available currently. Such a wide range of platforms posed further challenge for enterprises to manage secured enterprise access to their systems. As a result, enterprises started to focus on specific platforms and eventually started to invest in Mobile Device Management Systems – MDMs. This type of solution was fine and worked especially well for the corporate owned/managed devices, but this only addressed the needs of a smaller part of the enterprise that wanted to be connected to the enterprise resources. Some enterprises would board these users onto their systems but would have to conform to the security policies set for the enterprise users, which would render their personally owned device much less friendly to a personal user. This group of users is now called the Bring Your Own Device (BYOD) community and it has quickly realized that the industry needs to provide solutions that give both the enterprise and the personal community both the security for the enterprise and freedom for the personal user.

In this section, the architectural details and work flows of such virtualization techniques and their system architecture for BlackBerry Balance from RIM, VMware Mobile Virtualization Platform and Divide from Enterproid will be presented. Furthermore, we will also provide a comparative analysis of various approaches and would propose a solution composed from the best practices from these architectures.

II. Mobile Virtualization Fundamentals

Enterprises are being faced with supporting employee owned devices since they no longer can afford to incur the cost of paying for devices and phone/data services. Employees are buying their own devices that have enterprise capability and want to connect to the enterprise

Manuscript received January 2007, revised January 2007

network so that they can do their work with greater flexibility and freedom. However, the employees also don't want to give up user experience and freedom at the cost of complex IT security policies. There are a number of components such as device choice, security models, liability, user experience, privacy and economics that need to be considered in delivering a BYOD offering that works well for the user and the enterprise [1]. Allowing BYOD devices in the enterprise is more than just saying "yes" to employee preferences, but it means to put policies in place that govern how devices will be used and how they will be managed.

In order to achieve adoption and sustainability in the enterprise, mobile virtualization is rapidly becoming a very attractive choice because it addresses the flexibly while preserving the privacy concerns from the user and it also delivers the security requirements for the enterprise [2]. The other side of the ecosystem, the device makers and carriers will benefit from mobile virtualization because they are able to more easily replicate the features found in the various devices and also deliver more features at a lower cost [3].

II.1. Mobile Device Security

In a BYOD environment, mobile device security is critical because you want to protect the user's device from being compromised from internet attacks which could lead to loss of personal and enterprise data. Various forms of protection are built in as part of the Operating System like device encryption, password policies, remote lock, remote selective/full wipe and more, [4] where is additional security is supplemented via third party applications that take advantage of Device Administration API's that are built into the Operating System to provide protection against malware, application scanning upon installation and more [5].

II.2. Virtualization via Device Management Policies

Mobile Separation can be achieved through the use of IT Security Policies that are managed by a Mobile Device Management (MDM) system. This type of approach is done by a server based management approach that lets the IT managers enforce policies as across the user base which can be applied to the whole community or on a group basis so that it can be customized to the level of security required per group. For example, executives can have a more strict set of policies to include device encryption and a general group that allows for limited email, calendar and contacts support. There are many MDM's available now in the market and most of them support the major Operating Systems like iOS, Android, RIM and Windows. Some examples of these that provide a wide range of security policies that manage separation of applications and data are BlackBerry Enterprise Server, Mobile Iron and Good Technology amongst several more [6].

II.2.1. BlackBerry Balance

Research in Motion (RIM) specifically took advantage of this marketing opportunity by focusing on the enterprise market and developed the BlackBerry smartphone along with a server component called the BlackBerry Enterprise Server (BES). RIM incorporated algorithms for managing the security of the device, applications and data in a secure and reliable manner. RIM further extended their technology and allowed the consumers to manage both, the personal and enterprise data & applications by introducing BlackBerry Balance technology solution. BlackBerry Balance extended its device security policies by managing them from the server and maintained a separation of enterprise applications and data from the rest of the system. The security layer clearly identified and elegantly separated the secured applications installed on the device with applications marked as unsecured. It further obstructed the user to exchange data between secured and unsecured applications, like cut and pasting, thereby preventing any information leakage [7].

Essentially, this solution allows work-related data to be stored in such a way that it is not accessible to applications that the user might install for personal use, essentially it allows for the classification of enterprise and personal data and applications. For example, in the case of social network applications like Twitter or Facebook, BlackBerry Balance restricts the access to enterprise data from within social networking applications. From a personal use perspective, access to the personal phone is permitted when the BlackBerry smartphone is locked. An important aspect for the enterprise is the ability to manage the devices, especially when people leave the company. For this, the IT admin has the ability to either delete all device data or delete only the enterprise data and disassociate the device from the BlackBerry Enterprise Server [8].

BlackBerry Balance is included in RIM's latest version of BlackBerry Enterprise Server 5.0.3 and BlackBerry Enterprise Server Express 5.0.3 or later, which can be used for company owned or employee owned smartphones running BlackBerry OS 6, 7 and the latest version 10 [8].

In summary, BlackBerry Balance leverages specific IT policies along with features built into the BlackBerry device operating system to provide the data and application separation for business and personal purposes. This gives the flexibility for the enterprise to support both personal and corporate owned devices by keeping

enterprise data secure and at the same time provide the user with personal freedom of their device.

II.2.2. Mobile Iron Virtual Platform

The MobileIron Virtual Smartphone Platform allows companies to manage multiple operating systems at a granular level, support corporate and employee-liable devices, enforce cost control, and create a private enterprise application storefront for employees [1]. By creating this storefront, Mobile Iron provides an effective private online app delivery system that is controlled via policies that controls who can download or run the app. IT Administrator can also create and apply rules for application security, which can define the applications as required, allowed or disallowed [9]. Through this policy control, Mobile Iron achieves the control and separation as to what applications can be installed or run on the mobile device.

II.3. Virtualization via Hypervisors

Virtualization via hypervisors gives the ability to run two or more instances of an operating system on the same phone, thereby giving the ability to run personal apps and services on one OS and the business services on the more secure OS. Basically there are two types of virtualization approaches for this:

Type 1 – Bare metal virtualization: Hypervisor that runs at the host mobile hardware level and has direct access to the hardware resources. This type of hypervisor can host multiple operating systems. Because there is direct control to the hardware, performance of each of the operating systems can be optimized. Furthermore, since each operating system is completely isolated from each other, this provides the best isolation and security from one another [10].

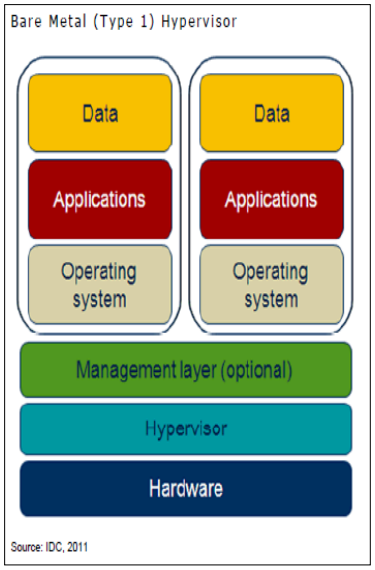

Figure 1 – Type 1 Hypervisors

Type 2 – Hosted virtualization: Hypervisor that runs within the host mobile operating system environment, just above the hosted OS where the second software level runs and the guest operating systems run at the third level above the type 2 hypervisor. Installation of the hypervisor is done on top of the guest OS because it is just like any other application. Performance of the guest OS is heavily dependent on the host OS. Furthermore, any compromise of the host OS will render the guest OS inoperative as well [10].

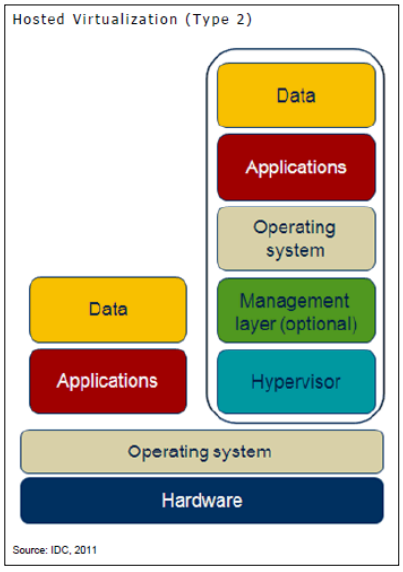

Figure 2 – Type Hypervisor

Hypervisor technologies have a downside in that they require working directly with OEMs which takes longer and there are fewer smartphones today that can support hardware level virtualization [11], however, this will change overtime as the ARM Cortex-A7, A15 and similar processors are incorporated into more mobile smartphones / tablets and also as standards like Virtualization Management Object (VirMO) proposed by Red Bend in the Open Mobile Alliance Device Management (OMA DM) Working Group [12].

II.3.1. KVM on ARM

Kernel based Virtual Machine (KVM) is a virtualization infrastructure for the Linux kernel that supports native virtualizations on processors with hardware virtualization extensions. KVM/ARM is a KVM based virtualization solution for ARM based devices that can run virtual machines with nearly unmodified operating systems. Since ARM is not virtualizable, KVM/ARM uses a lightweight para-virtualization via a script-based method to automatically modify the source code of an operating system kernel to allow it to run in a virtual machine. This lightweight paravirtualization is architecture specific but operating system independent [13]. These changes in the guest OS

kernel are made so that it can take care of sensitive non-privileged instructions doing the trap and emulate methods which are then handled by an interrupt handler which then emulates the appropriate functionality [14].

II.3.2. Xen Hypervisor on ARM

Xen is an open-source hypervisor that allows for multiple operating systems to safely share the hardware via resource management without sacrificing performance or functionality [15]. A basic Xen configuration is a hypervisor that consists of a small layer on top of the physical hardware. It implements virtual resources such as vMemory, vCPU, event channels and shared memory, and it controls the assignment of I/O devices to VMs. The user domains – DomUs are started by the Dom0 and they can run any paravirtualized operating systems like Linux and others. The Guest OSs have minimal changes where privileged operations are changed to calls to the hypervisor [16].

Xen has been ported to the ARM architecture [17] used for secure mobile phones supporting enhanced security features for mobile devices with mandatory access control through access control models and a secure boot process which is designed to detect any alternations of the VMM during the bootstrap process. Samsung has taken a keen interest in this project and has developed a version that supports ARMv5, ARMv6 and ARMv7 processors, the later one supporting the new virtualization extensions [18].

II.3.3. OKL4 Microvisor

Open Kernel Labs (OK Labs) is a provider of virtualization software for mobile devices, consumer electronics and embedded systems and its leading offering is the OKL4 Microvisor which has been embedded into more than 1.2 billion devices, including almost all CDMA phones because of the strong partnership with Qualcom [19]. The OKL4 Microvisor is a type 1 hypervisor that can be either built into the device at the OEM level or applied after the fact via OK Lab's Virtualization Over the Air (VOTA) process, which is similar in concept to Over the Air (OTA) firmware updates [20].

II.3.4. Motorola Evoke AQ4

The Motorola Evoke was the world's first mobile phone that uses hypervisor virtualization, implemented using OKL4 as the core virtualization technology. The requirements for the phone were such that only could be met by a design based on virtualization. This was accomplished by using a phone with a specific price point based on a single core design using an ARM9 core, a user interface running on Linux (OS), the baseband stack running outside of Linux, and components from BREW UI framework were re-used.

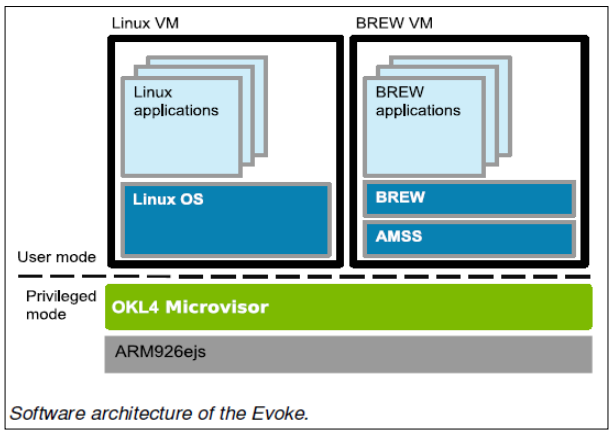

Figure 3 - Motorola Evoke AQ4 Architecture

The software architecture for the Evoke is shown in the diagram below, where there are two virtual machines, running on top of the OKL4 Microvisor which interact via the OKL4 message-passing IPC as well through shared memory. The complete Linux system is running de-privileged and the AMSS/BREW baseband stack/OS are both running in user mode. Due to the high performance of the OKL4 Microvisor, the virtualization overhead was kept to almost unnoticeable levels which in many respects, it was better than what was achievable with running native Linux.

In conclusion, Motorola produced an attractive device with a snappy user interface that was even more responsive than other non-virtualized phones, even those which were based on more powerful ARM11 processors [21].

II.3.5. VMWare Mobile Virtualization

VMware is committed to bringing virtualization to the mobile handset and two manufacturers – LG and Samsung have announced their support for this solution on Google Android [22]. At VMworld 2011, VMWare announced VMware Horizon Mobile Manager, previously known as VMware Mobile Virtualization Platform (MVP) which allows Android phones use virtual machine technology to run a second instance of Android, very much same way virtualization works on servers and desktops. This solution basically has two separate phones running on one device, and can switch from the personal one to the corporate one by clicking a “work phone” icon. By isolating the employee’s work environment from their personal environment and providing IT managers a Web-based management console to control what employees can do on the work portion of the phone, the user can have a more relaxed personal experience while the enterprise can have a manageable and secure environment [23].

Having two environments on the same device doesn’t

really make things more complex for the user because common functions such as receiving a call are active regardless of which environment is active. VMware says performance impact will be minimal and that the offering will work on both single and dual-core processors. Google Android was picked for the development of this platform largely due to the flexibility of it being open source. Initially, LG signed on to the project last December and now recently, Samsung has joined in giving a wider variety of devices such as the Galaxy S II phones and Galaxy tablets. In the future, more devices and other manufacturers are to be supported according to press announcements [24].

Figure 4 - MVP personal/enterprise phone screenshots

It is possible today to do many work activities on Android phones and tablets, but the real issue how to ensure that the enterprise doesn't get affected by malware, viruses or loosing data when the device is lost. This is why VMware's Horizon Mobile is very attractive because the enterprise phone is not affected by any malicious software and can be managed by IT administrators. If the device is lost, it can be remotely locked or wiped. Furthermore, the virtual phone can be provisioned with standardized or custom templates and also push out application updates over the air. User policies can be defined that can restrict what functions/features can be used and configure security features such as device lockup timeouts and passwords. From the employee's perspective, they are happy because they can now use their favorite personal device to do their work as well as use their favorite personal applications without having to carry two different devices. [25].

Horizon Mobile Manager (HMM) is a web-based device management system that allows IT administrators to comprehensively manage the lifecycle of the work phone from creating it to wiping it. HMM has a key set of capabilities that make up the system that allows it to be managed. Key features implemented in the system are: (1) Templates to create a work phone which can be created to address the needs of different employees. Some employees like managers can have some extra applications that are required for their day to day work or executives may have more restrictive policies. (2) Policy management engine to define what the user can and cannot do in the work phone. This is a key feature because the enterprise may want to restrict particular features on the phone whenever the enterprise phone is active. One example is the ability of copying data across applications between the personal and work side, thereby preventing data leakage. (3) Provision the work phone over-the-air which allows the device to be remotely provisioned thereby not requiring a manual install on the device. Furthermore, the IT administrator can make adjustments, add/remove applications, push down new templates without having to reload the device. (4) Review health of the deployment and vital stats from a dashboard. (5) Application management with the ability to push applications over-the-air to the work phone from the application catalog. Allows adding, removing of apps and dynamically pushing updates to the device. Multiple application versions are allowed. (6) Lock or wipe to de-provision the work phone which allows the ability to lock or wipe a device when the device is lost or sometimes the user might want to just reload the enterprise side [23].

On the device side, VMware Horizon Mobile Platform is built around a type 2 mobile hypervisor that is based on a lightweight paravirtualization technique for ARMv7 cores that is aimed at minimizing the total system complexity. A series of device and platform virtualization approaches for storage; networking and telephony are implemented which are key in enabling the performance, reliability and security of the system. Finally, this hypervisor is applied to the virtualization of the Android operating system, allowing it to run both the guest and host environments [26].

VHMM differentiates from previous approaches of system virtualization on the ARMv4-7 architectures that have entailed some form of core paravirtualization like Xen on Arm [16] by employing a distinct shallow paravirtualization approach that requires only the identification and replacement of sensitive instructions [26].

II.3.6. Red Bend – vLogix Mobile

Red Bend acquired VirtualLogix, a provider of mobile device virtualization solutions which it delivered to semiconductor vendors, OEMs, ODMs' service providers, and systems integrators. Over 1 million devices shipped with VirtualLogix's type 1 hypervisor technology embedded on devices like the Acer beTouch E110, E120, and E130 models; HTC Tianyi; KTouch W606; and CoolPad Yulong W711. Red Bend adopted this technology to fit into a broader portfolio and also saw it could use its strengths is in its ability to partition secure software domains that can be managed separately [10].

This offering supports processors based on the ARM

Cotex-A15 and Cortex-A7 cores in single and multi-core configurations. Red Bend is enabling device manufacturers to take advantage of this latest family of cores without the need to modify an existing high-level operating system (HLOS). vLogix Mobile performs management of the chipset's multi-core architecture in the Virtualizer, allowing the HLOS to remain as is. Users of this technology benefit by not having to redesign, redevelop and revalidate existing software to support new OS configurations [12].

The solution is comprised of the Virtualizer which is a type-1 hypervisor that runs together with suite of software modules that are configurable according to the desired deployment. The Virtualizer runs on the host hardware, also known as a Bare Metal Hyper, and schedules access to the shared hardware services like file systems, serial lines and network interfaces amongst the virtualized operating environments. System resources like RAM and persistent storage like flash memory are partitioned and allocated according to the performance demands of each of the operating domains or virtual machines. Hardware resources like CPU, clock and memory management unit are also virtualized for each of the guest operating systems and the access to the actual hardware is allocated by the Virtualizer [12].

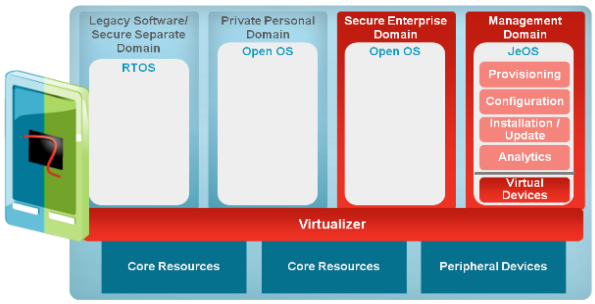

Figure 5 - Red Bend vLogix Mobile

Virtual devices are provided for each of the guest operating systems that access system resources like screen, 2D/3D hardware acceleration, multimedia acceleration, Wi-Fi, GPS and other input/output devices. This provides a secure separation in each of the OS/domains to the features of each device without the physical access to the device itself. The Domains are virtual machines that are designed to run virtualized images of Android where IT administrators can deploy over the air a secure enterprise environment that can run corporate applications and network access on a consumer owned device [12].

Red Bend's solutions is simple to use, where the user clicks on an icon on their home screen that takes them to work image that is running their business apps and easily switch back to their personal side by clicking on the personal home icon. IT administrators can control what applications are installed on the enterprise side and also excluded services like Android Market/Google Play [27]. Notifications in the business partition will appear on the user's notification bar where they can then go back to the home screen and switch to the business partition. The difference between Type 1 and Type 2 hypervisors can be visualized by noting the change from one platform to another when one is done at the phone lock screen where the other one is done at the application level, thereby providing an increased hardware level security and better performance [28].

II.3.7. Cells

Cells is a virtualization architecture for enabling multiples smartphones that run simultaneously on the same physical cell phone in an isolated manner. This architecture follows a usage model where there is one foreground virtual phone and multiple background virtual phones. A device namespace mechanism and device proxies are integrated with a lightweight operating system virtualization to multiplex phone hardware across multiple virtual phones while providing native hardware device performance. The platform include a fully accelerated 3D graphics, complete power management features, and full telephony functionality with separately assignable telephone numbers and caller ID support. A prototype was implemented of Cells that supports multiple Android virtual phones on the same phone. Performance results demonstrated that Cells imposed only modest runtime and memory overhead, worked seamlessly across multiple hardware android smartphone devices and transparently runs Android applications at native speed without any modifications [29].

II.3.8. Cellrox

Cellrox is an Android based technology that has roots on the Cells project [30] and it is not considered to be a complete OS virtualization because the virtualization is limited to the user space, creating multiple personas that share a common Linux kernel. Since the kernel is shared, all of the personas have to be a similar version of Android. Cellrox provides the same look and feel as the original operating system in order to provide a common user experience between the personal and work personas. Mulit-tasking between the various personas works much the same way as how multi-tasking works between different Android apps where only one persona is in the foreground at any given time and the apps running in other personas run just like any other background app.

Cellrox has the ability to support more than two personas which gives the ability to have a multi-tier level of locking down applications. For example, the personal

persona can be very relaxed, while a work persona has semi-restricted environment with standard security policies for the enterprise and third persona could be a completely locked down environment for highly confidential applications. One of the outstanding features that Cellrox features is the ability to have shortcuts of the applications from the different personas such that the user doesn't have to jump between the different personas to find the application which is the case in most of the separation technologies [31].

Usability is a key component in Cellrox. Only one persona is in the foreground while the others run in the background. The user can easily switch between the personas by using a custom key-combination to cycle through the personas or by swiping up and down on the home screen of a persona and each persona has an application icon that can be launched to see a complete list of available personas to select from. For security purposes, the system can be configured such that a no-auto-switch option will prevent background personas from being switched to the foreground without explicit user consent, preventing a background persona from appearing unexpectedly. An auto-lock feature can be enabled that will require the user to unlock the persona using a pass code or gesture whenever a persona switches from the background to the foreground. CellRox's technology was evaluated through an experimental study at Columbia University and the results demonstrated performance benefits in the their approach suggesting no noticeable performance difference between the operation of the mobile device in a persona compared to the native device operation [32].

II.4. Mobile Separation via Containers

Another approach to providing separation on a mobile device is via Application Containers. This is achieved by using a solution more like Unix method of multiple users where you have one box with multiple users logged in and each user has their own experience. Each user has its own experience and all users run concurrently with one kernel and one operating system [37]. Applying this logic, one user is the personal side and the second user is the enterprise side.

II.4.1. Good Dynamics Technology

Good Technology provides two technologies that address BYOD. Good for Enterprise, the first of the two, is a secure e-mail, mobile device management and a Intranet-Internet proxy server solution targeted for enterprises. Second is Good Dynamics platform that brings necessary tools, infrastructure, and APIs to developers, allowing them to provide secure applications across devices and operating systems. This protection is delivered by containerizing data at the application level which is accomplished by wrapping a layer of protection around the enterprise deployed apps, which separates the corporate data from the employee's private information and consumer applications. Through this containerized approach, Good Dynamics establishes a secure application environment that minimizes the possibility of data loss. The containerized applications provide the employee the freedom to access enterprise data in a safe and secure manner while being able to switch back and forth with their personal applications without compromising company information. This container-based method applies secure and encrypted transmission of data end to end, from the enterprise servers behind the firewall all the way to the mobile device. Good Dynamics architecture requires that every application use the Good Dynamics APIs and be compiled/linked with its SDK in order for it to run within its secured container. This limits the number of commercial applications that can within this container [33].

II.4.2. Divide by Enterproid

Enterproid introduced a mobile virtualization solution late 2011 called the Divide Platform which gives users a way to use their smartphones for both work and personal life. This solution which runs on Google's Android devices 2.2 or later and Apple's iOS devices like iPhone and iPad is designed to provide multiple profile support, a set of productivity apps, as well as a personal and enterprise cloud management system. The Divide system truly blends in well providing the end user true separation from the work environment without compromising personal freedom [34].

Figure 6 – Divide - Personal to Enterprise switching

Divide functions as a container application, with security policies and management features applied solely around that application, leaving the rest of the user's device untouched. Organizations can then in turn manage their own applications within the container. The management features are essentially the same as with most MDM solutions: password policy, encryption, data isolation, clipboard restrictions, remote wipe, screen locking and more [35]

Virtualization is done by having separate secure

profiles for work and personal environments. The personal side enjoys the freedom of all the functions, features and applications including access to the Android Market, where as the work side is managed by the IT administrators with a separate more strict profile suited for the appropriate enterprise. Switching between the two environments is done by a simple double tap of the Home key or can also be done via application icons on each of the sides or by going to the notification/alerts bar. Separation of applications is essential in a virtualization system because you want to avoid data leakage from enterprise into personal applications. All applications in the container side benefit from an encrypted 256 bit storage. Furthermore, encryption is not dependent on the OS, hence not compromised immediately on rooted or jailbroken devices – all the encryption is built within the application. Another example of separation is to limit the potential for leaking enterprise information into personal social networks by limiting the transfer of information between the personal side and enterprise container by restrict the ability to cut/paste information between the to environments. Additionally, Divide provides a rich set of native-Android office applications like mail, calendar, tasks and contacts. The mail application provides threaded email conversations which can also be searched both on the device and on the server. The enterprise mail, calendar, contacts are configured and delivered via the standard Active-Sync API's which provides the ability to connect to various email backend systems like Google, Yahoo, Microsoft, IBM Lotus Notes, etc., All of which are fully synced via 3G or Wi-Fi networks, thereby providing ultimate flexibility where ever you are at. Most importantly, because all these applications run in the separate work environment, everything is fully encrypted and compliant to the policies defined by your IT administrator [36].

Another set of features that furthers capabilities of the system is a cloud based management portal that allows the user and the IT administrator to manage their devices. The user portal, also referred to "My Divide", provides the user of the smartphone a number of features to control their device, such as they typical device wipe, reset both device and divide password, lock device, but it also provides features like just wiping enterprise data, audio beacon to locate device, device location, push a URL do the device browser and more. The portal also provides additional tabs that give the user the ability to view their network usage, the applications installed as well as a detailed list of the state of the device components like the phone, Wi-Fi, battery, network, audio, location and more.

The IT admin portal, also called "Divide Manager", provides some similar functions like location, device or enterprise wipe, password reset but it also where the system is configured by defining groups of users, device policies and enterprise applications. The device policies allow for configuring the typical mobile device management settings, but what is different here is that these settings are for managing the enterprise partition. Here you can set the device password quality, length, expiration, history and lockup timeouts.

Security features allow for controlling clipboard sharing, what happens if a user removes their SIM card and even checks to see if the device is rooted and gives options on the actions to take if this condition takes place. Another key piece of the system is the ability to manage applications in the enterprise partition. The admin is able to upload specific applications that will get pushed to the device upon installation of Divide. Furthermore, the system allows for the ability to control what apps are allowed or not allowed to be installed in the enterprise partition. System performance and battery life are well maintained because these are managed by the operating system and does not require direct access to the hardware [37].

Divide differentiates itself from other solutions is that it runs as an applications and it does not require any cooperation with the phone OEM. The install does not require any low-level drivers and uses the standard Android procedures for installing applications. It is a light weight solution that shares much of the device resources and significantly reduces the device overhead required by virtualization and also delivers 256 bit encryption for data [28].

Overall, Divide by Enterproid provides a very promising system for enterprises to use, especially for their BYOD community. It is very functional and provides a wide range of security options and features as well as giving the end user the freedom of their device on the personal side [38].

II.4.3. TrustDroid

TrustDroid is a practical and lightweight domain isolation solution that runs on the Android OS. It provides application and data isolation by controlling the main communication channels in Android, mainly IPC (Inter-Process Communication), files, databases and, socket connections. This solution is lightweight because it has a low computational overhead and does not require duplication of Android's middleware and kernel like other Virtualized solutions approaches. It also organizes applications along with their data into logical parallel domains. Figure 7 illustrates different methods for achieving isolation. TrustDroid is shown in (a) where it extends Android's middle ware and kernel with mandatory

access control. OS-level virtualization is shown in (b) and this is typically seen in Application level containers. Hypervisor based technologies is shown in (c). The areas designated in black are the trust computing base (TCB) which is responsible for the security enforcement on the platform and also trusted by the enterprise. TrustDroid has the largest TCB, however, it is one of the most lightweight because it doesn't duplicate any portion of the operating system stack and provides good isolation.

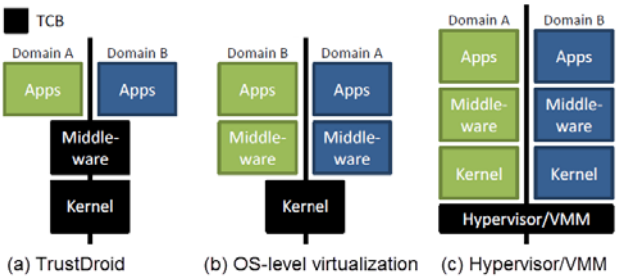

Figure 7 – Approaches to isolation

At runtime, all application communications are monitored, as well as access to common shared databases, file-system, networking, and denies any data exchange or application communication between different domains. TrustDroid adds a negligible runtime overhead and compared to other virtualization approaches, only minimally affects performance and battery life [39].

II.5. Multi-user – Android 4.2

Google in its latest version of Android 4.2 OS the option to have multi-user support which gives the device owner provide separation across users [40]. This becomes a very interesting development because it gives the device owner the ability to have its own version of separation – one user for personal use and a second user for enterprise use. It will be interesting to see if enterprises will be leverage this functionality in such a way to accommodate and attract the BYOD community to use their personal device for enterprise use as well in this type of environment. In order to achieve this, it will be necessary to see if the enterprise will be able to apply its device management software and policies on the enterprise user while allowing the personal freedom on the personal side.

III. Comparative Analysis

A way to help analyze which system is most appropriate for your environment is to do a comparative analysis of the various solutions available. Here we compare four solutions: Divide from Enterproid, Horizon Mobile from VMWare, BlackBerry Balance from RIM and Good Dynamics from Good Technology.

In order to analyze and understand these various solutions, we have to look at them by each category and the corresponding functions.

III.1. General Platform Support

Each of the solutions provides a Self Service Portal where users can manage and view their device which really helps minimize the end user support costs. Equally important is the management/admin system for the solutions where the IT Administrator can manage the system from any location via a standard browser. The solutions start to differentiate from each other when it comes to what platforms/OS they support. Both BlackBerry and VMWare are platform OS/device specific where as Enterproid and Good Technology support across platform solutions on both Google Android and Apple iOS.

III.2. Device Inventory

The ability to identify the devices being used by user communities reveals a very useful set of analytics for the enterprise. Being able to make decisions based on OS versions or device adoption can save a lot of money and deployment time for applications. All the solutions evaluated do very well at being able to collect device information like device model, manufacturer, carrier, cpu, memory and more. GPS and Location Based Services provide the ability to collect and record device location.

III.3. Management Actions

Each of the solutions provide a complete set of device management functions that enable the system to have device or selective wipe, remote lock, device locate, deny email access or user functions like password reset, message sending.

III.4. Security and Policy Management

A key component in each of these systems is the ability to ensure proper device security which is enforced via a set of key device policies. Key security features like requiring device lockup timeouts, minimum password lengths and complexity help ensure that the device is only accessed by the intended person which is also well done by various solutions. Jailbreaking/rooting the device exposes unwanted device access which lends itself to being infected by malware or installing applications that can leak information about the user or the enterprise. Both Enterproid and Good Technology provide very good detection and prevention for this kind of exposure.

One function that is not available across the surveyed solutions is the capability to interface with other unified management systems, this being a key feature to help distribute and deploy standard security policies across servers, desktops, tablets and smartphones in an

TABLE I
Comparative Analysis of Mobile Virtualization Solutions

Category	Functionality	Enterproid Divide	Horizon Mobile VMWare	BB10 Balance	Good Technology
General Platform Requirements	Self Service Portal	✓	✓	✓	✓
	Management/Admin Portal	✓	✓	✓	✓
	iOS - iPhone, iPad, iPod Touch Support	×	×	✓	✓
	Android 2.2 + Smartphone and Tablet Support	✓	✓	✓	✓
	BlackBerry OS Support	×	×	✓	×
Device Inventory	Model, manufacturer, carrier, name, user	✓	✓	✓	✓
	Memory, external storage, battery info, serial #	✓	✓	✓	✓
	Device location (GPS)	✓	✓	✓	✓
Management Actions	Device wipe, selective wipe, remote lock	✓	✓	✓	✓
	Reset pwd, send user msg, deny email access	✓	✓	✓	✓
Security and Policy Management	Password policies - 8 alphanumeric password	✓	✓	✓	✓
	Lockup timeout - 30 min support	✓	✓*	✓	✓
	Jailbreak/Root detection / actions	✓	×	×	✓
	Device restrictions (cloud backups, etc)	×	×	×	✓
	Unified Mgmt, incl. traditional endpoints	×	×	×	×
Enterprise Access	Email, calendar, contacts, instant messaging	✓	✓	✓	✓
	VPN separation	×*	×*	✓	✓
Application Management	Apps are pushed and installed on device	✓	✓	✓	✓
	Prevent access to external app store	✓	✓	×	×
	Recommend apps / enterprise app store	×	×	×	✓
	View installed apps	✓	✓	✓	✓
Data Leakage Protection	Prevent Cut/Paste from Enterprise / Personal	✓	✓	✓	✓
	Prevent/manage email attachment export	✓	×	×	✓
	Integration with Enterprise App containers	×	×	×	✓

* Some support available

enterprise.

III.5. Enterprise Access

Email, calendar, contacts and instant messaging are key productivity applications for the enterprise, all of which are well supported in each of these mobile device virtualization technologies. As to VPN separation, this is much more complicated based on the virtualization technology. As to VPN access, both BlackBerry and Good Technology provide a built in solution and since both of these provide device level security and policies, they don't have to separate out the network access. In the case of Enterproid and VMWare, VPN has to be filtered at the container level, thereby requiring integration with 3rd party VPN solutions and making the options much more limited for enterprise deployments. The key here is that the data and access to these are separated and protected.

III.6. Application Management

In the enterprise, applications are key to providing information to their users, which make application delivery and management a key component in these solutions. Furthermore, in order to maintain device integrity, access to general application stores needs to be restricted and at the same time, an enterprise application store or delivery mechanism needs to be available. Device policies typically will deliver a core set of applications to the device as defined by the enterprise. Another function that is important is to have the ability to keep an inventory of the applications that are installed on the device per user. The solutions in this space do offer the ability to deliver a core set of applications to the device when installed but they are still very immature in the concept of having an enterprise appstore but they do have the foundations to integrating to existing ones.

III.7. Data Leakage Protection

From an enterprise perspective, this is probably one of the more important features / functions in these systems which are to prevent the leakage of data from the enterprise to the personal side. Many users use their personal email to transfer work related information. Here, a very clear distinction is made between what is work and what is personal. Several restrictions are imposed to help limit the leakage of data. One is by restricting the ability to cut/paste data between personal and work. Second, is to prevent attachments in emails from being detached in a non-secure location and using application viewers to securely view the documents.

Finally, the more critical function is the ability to attach or associate an application with the container such that all the data stored by that application is placed within the secure container. Both Enterproid and Good Technology have the ability to integrate an application into the container, however, Good Technology requires that the application be recompiled with their special SDK libraries, whereas Enterproid has a simple binding process where you can upload the application and it will add specific hooks to the executable that will associate it with the container.

From the above comparative analysis, each of the solutions are very good at providing the traditional Service and Management portals, device inventory and information, email/calendar/contacts, management features like device wipe and password management and data leakage protection. However, the different solutions start to differ in function and capabilities when it comes to platforms, application delivery and data protection.

III.8. Comparative Results

The BYOD phenomenon is real and it is affecting every enterprise. Some allow those users on their corporate network with enterprise policies but users only remain on for a short time because they don't want to give up their personal flexibility. They find corporate policies too restrictive loose too much personal freedom. As such, enterprises are losing many users and wasting valuable resources in the process. They are quickly realizing that they need to invest in device virtualization technologies that will allow them to maintain their security integrity as well as keep the personal freedom and flexibility of the user because it is becoming very clear from various BYOD surveys that employees are willing to pay for their device and usage and enterprises are willing to provide the necessary infrastructure for them to access their networks. Due to this, many enterprises are evaluating solutions such as the ones discuss here.

By far the most well managed system is that of BlackBerry Balance because it provides proven device level security and data integrity as well as very good integration with the device and device management system. Compared to the other systems, it is restrictive due to the device lockup policy being device wide. The BlackBerry Balance system would be much better for personal use if it provided a way to containerize all the enterprise apps and data such that only that section would require the device lockup and timeout. Everything else would be fine because any wipes would be that of only enterprise data and applications, thereby leaving the personal information intact. The one obvious downside to this solution is that it is BlackBerry centric and it only works for BlackBerry devices. Given the large trend and influx of iPhone and Android devices, this solution is limited and doesn't solve the diverse devices that users have.

As to device virtualization via the use of a hypervisor, VMWare really has something very interesting and powerful with its VMware Horizon Mobile platform. It provides the most flexibility and gives the ultimate separation; however, it has some limiting factors that prevent it from wide adoption at the moment. One limitation is that the solution requires that the device manufacturers to incorporate the hypervisor into the device image for security purposes and on top of that, enterprises need the ability to ensure that the hypervisor has not been compromised in any way. As such, some types of security applications like malware detection software and device management agents will need to reside on the personal side in order to ensure that the device has not been compromised.

Lastly, Divide from Enterproid provides a very attracting solution for the user community because it provides the best flexibility out of the three solutions. The product is really addressing the sweet spot in the market by addressing most of the security requirements as well as addressing the personal requirements from the employees. Many enterprises are looking at this solution because it provides the flexibility to incorporate their own corporate policies, enterprise applications and device management.

IV. New Mobile Virtualization Architecture – Hybrid Containers

An agile, lightweight, but secure extensible hybrid application container and deployment mechanism for mobile devices has the potential to provide improved cross platform support, reduced development lifecycle time and a consistent application security model within large organizations. Such an architecture can provide a set of security and mobile device management interfaces to lightweight applications written in a high level markup and scripting language and also how these applications can be provisioned via push or pull mechanisms to an enterprise user's device. The use of such a container for hybrid applications widens the potential support and development resource within an enterprise possessing readily available skill sets, thus permitting mission critical applications to be developed in a much shorter time frame [41].

IV.1. Secure Hybrid Mobile Application Container

The hybrid container builds upon the ideas of solutions such as Apache Cordova (PhoneGap) [42] but extends this idea such that the container shell application becomes home to not just one single use application but multiple applications, These are presented to the user as an

extensible virtual desktop within one application running on the device, also referred to as the Hybrid Mobile Application Container. Additional hybrid mobile applications can be dynamically downloaded and installed within, their data and services managed and encapsulated within the application space of the container. This hybrid mobile application (HMA) is created to run on a mobile device where the application logic is primarily written in JavaScript and the user interface components are defined using HTML5 and CSS3.

The HTML and associated JavaScript application logic are contained within a native code wrapper which allows what would otherwise be a web application to be installed as an application on the mobile devices desktop. The native wrapper exposes device level function to the scripted portion of the application through a set of plug-ins written in the native language of the particular mobile platform, each plug-in has an associated JavaScript API. The plug-ins can expose native function that would otherwise be inaccessible using standard HTML5, for example access to the device calendar, phonebook or in built hardware such as the camera.

The interaction between the JavaScript API and the native code is implemented on each mobile platform using methods made available through the platform SDK. Calls from JavaScript to native code exploit the ability to intercept navigation events and examine the URI scheme, a URI scheme can be used to indicate that the URI is an encoded method call and should not be passed on to the embedded browser to handle, the hybrid application shell can then process the encoded URI and call native methods within the application shell. Conversely, function calls can be made to the embedded browser code by injecting JavaScript at runtime into the currently loaded web page.

IV.2. Benefits of Hybrid Application Development

There are several advantages to using a hybrid development approach when creating mobile applications. Using HTML5 and JavaScript increases the portability of the application; the UI and application logic can be reused across multiple platforms with minor changes to presentation format performed using CSS [43]. Another major advantage is the abundance of HTML and JavaScript development resource in comparison to native language knowledge of Objective-C and Java.

IV.3 Hybrid Mobile Application Container Requirements

Increasingly large organizations are allowing employees to use their own mobile devices for work purposes; this has created a set of Mobile Device Management (MDM) challenges for the IT teams within these organizations. A major challenge is to allow the employee freedom to use the phone for personal tasks unencumbered by lengthy passwords and IT Management enforced policy, yet still protects enterprise data and applications. In order to do this some type of separation mechanism needs to be introduced in order to ensure that there are no data leakages or security intrusions. Protection is delivered in the form of a virtual container that applications can run inside, their data is containerizing at the application level, this is accomplished by wrapping a layer of protection around the enterprise deployed apps, which separates the corporate data from the employee's private information and consumer applications. Further security is provided by some solutions by using specific SDKs that intercept all data related APIs and store the data in secure/encrypted file stores. An additional security measure is to monitor the application using Mobile Device Management and provide functions that give an IT administrator the power to do corporate wipes of the enterprise container contents. A time bomb mechanism maybe employed that will self wipe the container and data if the device has not checked in within a period of time defined by a device security policy. Other solutions like micro-containers have been done which use cloud based services to provide mobile service-containers for hosting service-based applications [44].

A few steps/requirements are needed in order to create a functioning Hybrid Application Containers which include Hybrid Application Creation, Application Deployment and Update Management, Mobile Appstore, Hybrid Container API and Hybrid Container Security.

IV.3.1 Hybrid Application Creation

Creation of a hybrid application should not require a developer to possess platform specific skills but rather have a grasp of common web development methods including knowledge in HTML5, CSS3 and JavaScript. The container itself consists of a native application code shell and provides all the necessary services and APIs that an application team would need to build applications. This allows the creators of hybrid applications to focus on the presentation and business logic in a platform independent manner without having to learn new high-level languages such as Objective-C or Java. Web development skills are more widely available in an organization; as a result the pool of developers who can contribute custom applications will in most cases be substantially larger than the set of developers who possess knowledge of the high level languages used to create applications on platforms such as iOS and Android.

IV.3.2 Application deployment and updates

Application deployment as well as application updates are essential in a container environment and they need to be simple and secure. Application deployment can be performed in two ways. The first by pushing the application(s) during initial installation of the hybrid container, the second deployment is via user requested

installation from an app store. The configuration of the applications to be installed is controlled by the backend system that manages or communicates with the hybrid container. In the case of application updates, there are two types that applications are sensitive to. One is the core native shell where plug-ins reside and the second is the internal HTML/CSS code. Depending on the complexity of the application, there may be few updates to the shell and frequent changes to the UI. In the case of updating the shell, this requires a complete re-installation of the application including both the shell plug-in as well as the core UI files. Careful consideration needs to be taken to make sure that any files or settings from previous installations do not conflict or cause adverse effects with the new installation.

IV.3.3 Appstore model

Hybrid applications by nature are very dynamic and are continuously being updated, new applications may be released that users will want to install after the initial setup of the hybrid container. In order to address this scenario, the hybrid container can be architected to connect to an application store backend that will allow the user to select from a catalog of applications that they are entitled to download and install.

The enterprise can also have a managed set of required applications that would be pushed to the hybrid container ensuring that users always have a set of pre-defined applications.

IV.3.4 Hybrid Container API

The container itself must be written in the native language used by each targeted mobile platform. It will provide a consistent set of APIs to application developers; it will also contain the mechanisms to allow hybrid applications to be installed dynamically within the container.

IV.3.4.1 Security and Storage

Security is of utmost importance within a large organization, data may have varying levels of sensitivity but the determination of what is sensitive should not be left purely down to the individual application developer. The best policy for application development is therefore to provide secure storage through strong encryption for all application data. Allowing the container to handle this behind a simple storage API reduces the burden on the application developer and allows IT management to be confident that standards and policy are being adhered to.

Due to the nature of the hybrid container application, data separation must be enforced via strict name spacing to prevent one application reading another's data. It may however be useful to allow sharing of data in certain circumstances e.g. allowing an instant messaging application access to the database of a contacts application.

IV.3.4.2 Enterprise Network Access

Providing access to data and services that are located within an organizations internal network from a mobile device can be problematic and a source of much debate amongst security teams. The option to provide device level Virtual Private Network (VPN) exists but could expose the organization to attacks by malware that may have been unintentionally installed on an employee's personal device. A second option is to provide access on an application-by-application basis often through a secure reverse proxy mechanism. This is much more secure but can be a lot of effort for the IT management part of the organization. The ideal option is to allow the container to manage access providing a common interface for applications that reside within it.

IV.3.4.3 Notifications

Applications in the most part act in a request response manner; however there are occasions where it is necessary to push information to an application. In a mobile context this usually takes the form of a „push notification" sent from a messaging provider service usually run by the creator of that particular mobile operating system. Apple provides the Apple Push Notification Service (APNS) whilst Google provide the Google Cloud Messaging (GCM) service. It is necessary to abstract the use of these messaging providers away from application developers both for client and server side development [45].

IV.3.4.4 Email/Calendar/Contacts

Common applications such as email, calendar and contacts could be provided with the container rather than on an organization-by-organization basis. This function is required to reside within the container as the information used is of a highly sensitive nature. The protocols used for these applications are well documented and standardized so implementing them as common components is justifiable.

IV.3.4.5 Application Updates

An update mechanism must be present in the container logic providing two distinct functions. Firstly the ability to check for code updates to the container itself, these may be frequent as the demand for new native plug-in components is driven by the arrival of different applications. Secondly to check for new versions of previously installed applications within the container. Both methods of update will require communication with a server based management component.

IV.3.5. Hybrid Container Security

The container should provide an authentication mechanism that secures access to the applications and data residing within. The credentials used by the container will be determined at install time and may be governed by a policy which enforces a password change at a time interval that is configurable.

IV.4. Hybrid Application Container Creation

The decision was made to use the open source Apache Cordova project rather than create the hybrid framework of the container from scratch, Cordova already has a clean extensible plug-in architecture and a large community of developers creating plug-ins to offer additional functionality on top of the base set.

When a Cordova application starts up the native shell code loads the HTML/JavaScript UI application logic from a fixed www root folder within the application file system. This root folder is read only once the native application is installed so it is necessary to subclass the Cordova class that loads from the root folder, this allows the UI application logic to be loaded from a different writable folder that we can install downloaded hybrid applications within.

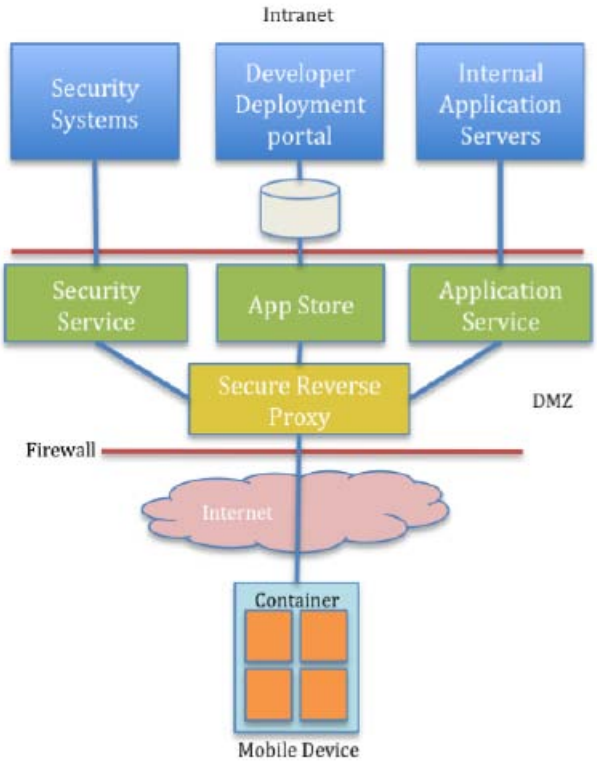

Figure 8 - Hybrid container deployment architecture

IV.4.1. System architecture

Figure 8 shows a typical deployment configuration for the container architecture and required backend services. A mobile client running the container connects over the Internet to a secure reverse proxy that can perform device authentication and authorization to establish a secure session. Within this secure session the device can contact a variety of services. These services are located behind a firewall in a demilitarized zone (DMZ).

The App Store service allows a user to browse and download new applications to the container; it also provides version management services to the container. Individual applications will need specific backend services made available to them, these can all be accessed via an application service located inside the DMZ. Requests to the service will be distributed to many pre-existing service APIs located within the organizations intranet.

A Security service also resides within the DMZ which provides applications with the service APIs that allow various authentication and authorization tasks to be performed. This service layer also accesses a variety of pre-existing intranet based applications. These can include employee directory servers, Mobile Device Management (MDM) solutions or authentication services. Figure 8 demonstrates system architecture for a hybrid container deployment.

IV.4.2. Hybrid Container Components

The following are the components that make up the Hybrid Container.

IV.4.2.1 Container Application

As illustrated in Figure 9 the container application is made from a variety of layered components, the application itself is written in the native language of the mobile device i.e. Objective-C for iOS, Java for Android. The container contains bootstrap code to load the initial Application Desktop user interface web application in a Web View when the application starts. An application executing in the Web View can call native code in the form of plug-ins, each plug-in has a JavaScript API and a corresponding native language class implementation. The desktop contains icons which when selected by the user trigger a call to the Application Manager (AM) which then loads the selected application into the Web View components.

IV.4.2.2 Plug-ins

A base set of plug-ins are included with the container, however more may be added over time as the container receives updates. Plug-ins are written to conform to the Apache Cordova interface and many of the standard Cordova plug-ins are included. Each plug-in has a corresponding JavaScript API so that it may be called from a hybrid application.

IV.4.2.2.1 Security

This set of plug-ins provides authentication and authorization components used when accessing services hosted on the organization's intranet, this includes a filtered VPN service.

IV.4.2.2.2 Device Hardware

Access to hardware-based functions of the device such as GPS, Camera and Microphone can be obtained and interacted with through this set of APIs.

IV.4.2.2.3 Messaging

The messaging plug-in set has responsibility for processing incoming notifications and ensuring that the

message is routed to the correct hybrid application. The AM is called if the application is not currently running and an appropriate alert can be shown allowing the user to switch to the application that the push notification targeted.

IV.4.2.2.4 Storage

In order to maintain security standards with regards to storage any write operations performed by a hybrid application should go through the storage API this allows the data to be encrypted before it is written. If the application requires access to a database then an application specific SQLlite instance is created. The entire database is encrypted using the container password created when the container is first installed.

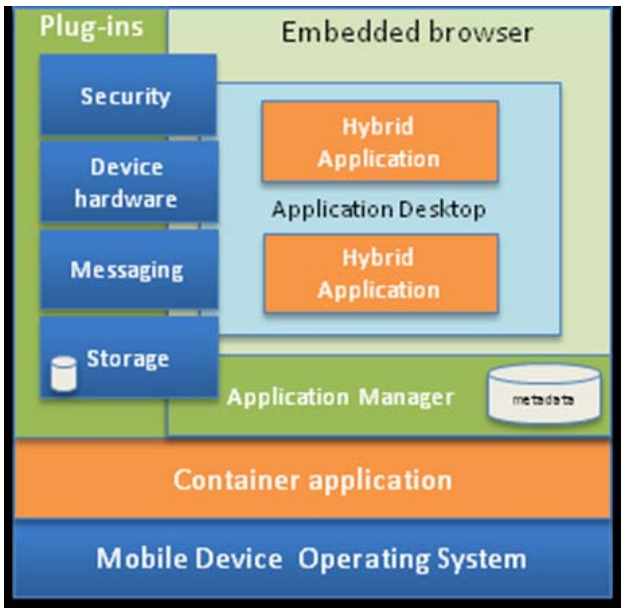

Figure 9 - Hybrid mobile application container

IV.4.2.3 Application Manager

The Application Manager shown in Figure 9 has several responsibilities including loading the desktop on the first instantiation of the container. When applications other than the desktop are started the container stores the state of the current application then reads the new applications descriptor. Any plugins required by the application are instantiated and the web resources loaded into the web view component, if a previous state was stored the state is reloaded before the application displays. The AM handles the installation of new applications within the container and also periodic version checks for both installed applications and the container itself. Version checks are performed by making an HTTP request to a server based version component within the appstore, applications can be updated dynamically and restarted within the container. In order to update the container itself the full application must be reinstalled.

Once an application has been developed the application resource files are added to an archive file and zip compressed. Along with the resources in the archive are an application descriptor and a desktop icon used to display the application within the container.

The application descriptor contains metadata such as the application name, version, the plug-ins used by the application. Installed application details are maintained within a local database by the container and queried at various points in the applications execution cycle.

IV.5 Mobile Hybrid Container Results

This hybrid mobile application container solution has shown itself to be very effective delivering increased application deployments, application management and improved application security, thereby, delivering an improved enterprise mobile ecosystem.

V. Conclusion

The explosion of mobile devices in the consumer space has been quite a disruptor for the mobile enterprise where corporate managed platforms are well defined, well contained and fairly secure. However, now with the emergence of the consumer mobile devices from Apple iOS and Google Android, the enterprise has been forced to make functional tradeoffs in order to maintain platform and data security. Due to the corporate security decisions, the end user gives up a significant amount of personal freedom and ease of use of their device. Enterprise security requirements like password complexity, device encryption, network restrictions and other techniques that restrict the access to information on the mobile device tends to drive users away and/or encourages users to find other less secure alternatives that will eventually compromise enterprise data and access. A number of mobile virtualization technologies have been presented here, each of them delivering specific features that focus on ensuring the security of the enterprise container and applications. Application level containers, type 1 and 2 hypervisors and the new hybrid application container architecture all lack the organic integration of enterprise & personal personas found in solutions like the emerging BlackBerry 10 Balance platform. Hypervisor technologies show promise; however, it is specific to the Android platform, has higher end hardware requirements and not likely to be seen on the iOS platform anytime soon.

In summary, this paper introduces details and analyzes via comparison several mobile virtualization technologies available on the market today. With this information, decision makers can develop a strategy for integrating the growing BYOD community into their enterprise mobile ecosystem.

References

[1] "Building "Bring-Your-Own-Device" (BYOD) Strategies." BYOD Strategies White Paper. Mobile Iron, 2011. Web. 27 Nov. 2011. http://info.mobileiron.com/BYOD_1.html

[2] "Are you ready for BYOD? Here are seven questions you should answer as you roll out new mobile capabilies" IBM Software – Thought Leadership White Paper, December 2011.

[3] Kharif, Olga. "Virtualization Goes Mobile." Businessweek - Business News, Stock Market & Financial Advice. N.p., 22 Apr. 2008. Web. 22 Sept. 2012. http://www.businessweek.com/stories /2008-04-22/virtualization-goes-mobilebusinessweek-business-news-stock-market-and-financial-advice

[4] Gruman, Galen. Mobile and BYOD Deep Dive. Rep. N.p.: InfoWorld, November 2011. http://www.infoworld.com/d/mobile-technology/download-the-byod-and-mobile-strategy-deep-dive-179850

[5] "Device Administration" Google Android SDK, October 20, 2012. http://developer.android.com/guide/topics/admin/device-admin.html

[6] Gruman, Galen. Mobile Device Management Deep Dive. N.p.: InfoWorld, March 2011. http://www.infoworld.com/d/mobilize/ mobile-management-infoworlds-expert-guide-371-0

[7] "Balance Technology." BlackBerry - BlackBerry Balance Technology Separates Personal from Business Information. RIM, n.d. Web. 14 Feb. 2012. http://us.blackberry.com/business/ software/blackberry-balance.html

[8] Halevy, Ronen. "RIM Finally Details Features of BlackBerry Balance." BerryReview, 3 May 2011. Web. 14 Feb. 2012. http://www.berryreview.com/2011/05/03/rim-finally-details-features-of-blackberry-balance

[9] Cox, John. "MobileIron Extends Control over Enterprise IOS Apps." Network World. N.p., 9 Dec. 2010. Web. 02 Nov. 2012. http://www.networkworld.com/news/2010/120810-mobileiron-4.html

[10] Crook, Stacy K., and Ian Song. "Mobile Virtualization Technology Assessment." IDC, May 2011. Web. 02 Oct. 2012. http://www.idc.com/getdoc.jsp?containerId=228088

[12] Red Bend Software. Mobile Virtualization, 2011. http://www.redbend.com

[13] Christoffer Dall, Jason Nieh. "KVM for ARM." Proceedings of the 12th Annual Linux Symposium, Ottawa, Canada, July 13-16, 2010.

[14] Rahul Ramasubramanian. Exploring Virtualization Platforms for ARM based Mobile Android Devices, Master Thesis, North Carolina State University, 2011. http://www.lib.ncsu.edu/resolver/1840.16/6998

[15] Paul Barham , Boris Dragovic , Keir Fraser , Steven Hand , Tim Harris , Alex Ho , Rolf Neugebauer , Ian Pratt , Andrew Warfield, Xen and the art of virtualization, Proceedings of the nineteenth ACM symposium on Operating systems principles, October 19-22, 2003, Landing, NY, USA

[16] Reiner Sailer, Trent Jaeger, Enriquillo Valdez , Ramon Caceres , Ronald Perez , Stefan Berger , John Linwood Griffin , Leendert van Doorn, Building a MAC-Based Security Architecture for the Xen Open-Source Hypervisor, Proceedings of the 21st Annual Computer Security Applications Conference, p.276-285, December 05-09, 2005

[17] Xen ARM Project, http://wiki.xen.org/wiki/Xen_ARM_(PV)

[18] Morgan, Timothy P. "Xen Hypervisor Ported to ARM Chips." Xen Hypervisor Ported to ARM Chips. The Register, 11 Nov. 2011. Web. 11 Dec. 2011. http://www.theregister.co.uk/2011/11/30/ xen_kvm_hypervisor_for_ arm_chips

[20] Open Kernel Labs. OKL4 Microvisor, Mar. 2011. http://www.ok-labs.com/products/okl4-microvisor.

[21] Gernot Heiser. The Motorola Evoke AQ4 – A Case Study in Mobile Virtualization. Open Kernel Labs, 2009 – okl4.net. http://www.ok-labs.com/_assets/image_library/evoke.pdf

[22] Thomas, Keir. "Virtualization Boosts LG Android Phones | PCWorld Business Center." Reviews and News on Tech Products, Software and Downloads | PCWorld. N.p., 7 Dec. 2010. Web. 22 Sept. 2012. http://www.pcworld.com/businesscenter/article/ 212764/virtualization_boosts_lg_android_phones.html

[23] VMTN. "VMware End-User Computing Blog." VMware End-User Computing Blog. N.p., 30 Aug. 2011. Web. 14 Feb. 2012. http://blogs.vmware.com/euc/2011/08/vmworld-2011-announcing-vmware-horizon-mobile-manager.html

[24] Ziegler, Chris. "Work, Play on a Single Phone: LG Teams up with VMware to Deploy Android Handsets with Virtualization." Engadget. N.p., 7 Dec. 2010. Web. 22 June 2011. http://www.engadget.com/2010/12/07/work-play-on-a-single-phone-lg-teams-up-with-vmware-to-deploy

[25] Whittle, Sally. "VMware Foresees Mobile Virtualization in 2010 | Business Tech - CNET News." Technology News - CNET News. N.p., 21 May 2009. Web. 22 Sept. 2012. http://news.cnet.com/8301-1001_3-10246338-92.html

[26] Ken Barr, Prashanth Bungale, Stephen Deasy, Viktor Gyuris, Perry Hung, Craig Newell, Harvey Tuch, and Bruno Zoppis. 2010. The VMware mobile virtualization platform: is that a hypervisor in your pocket?. SIGOPS Oper. Syst. Rev. 44, 4 (December 2010), 124-135.

[27] Gohring, Nancy. "Red Bend Working on Mobile Virtualization." CIO. N.p., 12 Oct. 2011. Web. 11 Apr. 2012. http://www.cio.com/article/691671/Red_Bend_Working_on_Mobile_Virtualization

[28] Brandon, John. "Is Mobile Virtualization Ready for Your Business?" CIO. N.p., 15 Mar. 2012. Web. 11 Apr. 2012. http://www.cio.com/article/702299/Is_Mobile_Virtualization_Ready_for_Your_Business_

[29] Andrus, Jeremy, Christoffer Dall, Alexander Van't Hof, Oren Laadan, and Jason Nieh. "Cells: A Virtual Mobile Smartphone Architecture." Proceedings of the 23rd ACM Symposium on Operating Systems Principles (SOSP 2011). Portugal, Cascais. 173-87. Web.

[30] "Cells: Lightweight Virtual Smartphones." Cells: Lightweight Virtual Smartphones. Columbia University Department of Computer Science, n.d. Web. 23 May 2012. http://systems.cs.columbia.edu/projects/cells

[31] Madden, Jack. "Cellrox Offers Multiple "personas" for Android Phones. Is This How We Wish VMware Horizon Mobile Worked?" Brianmadden.com. N.p., 22 Sept. 2012. Web. 29 Sept. 2012. http://www.brianmadden.com/blogs/jackmadden/archive

/2012/09/21/cellrox-offers-multiple-personas-for-android-phones-is-this-how-we-wish-vmware-horizon-mobile-worked.aspx

[32] "The ThinVisor Mobile Device Virtualization Architecture." Cellrox, Nov 2011. http://www.cellrox.com/wp-content/uploads/2012/02/Cellrox-ThinVisor-Architecture.pdf

[33] "Balancing Security and Speed: Developing Mobile Apps for Enterprise" Good Dynamics White Paper. Good Technology, 2011. Web. 27 Nov. 2011. http://media.www1.good.com/documents/good_dynamics_wp.pdf

[34] Dolcourt, Jessica. "Divide for Android Takes on BlackBerry, Sprint ID." CNET. N.p., 28 Feb. 2011. Web. 22 June 2011. http://www.cnet.com/8301-17918_1-20036669-85.html

[35] Madden, Jack. "BYOD Smackdown 2012: Enterproid Divide Creates a Secure Work Persona on Personal Devices." ConsumerizeIT. N.p., 16 Feb. 2012. Web. 10 Oct. 2012. http://www.consumerizeit.com/blogs/consumerization/archive/2012/02/16/byod-smackdown-2012-enterproid-divide-creates-a-secure-work-persona-on-personal-devices.aspx

[36] Hazard, John. "Enterproid Divides Work and Personal on Android Devices, Fires at BlackBerry." ZDNet. Between the Lines, 28 Feb. 2011. Web. 22 June 2011. http://www.zdnet.com/blog/btl/enterproid-divides-work-and-personal-on-android-devices-fires-at-blackberry/45407

[37] Mache Creeger. ACM CTO Roundtable on Mobile Devices in the Enterprise. August 3, 2011. http://queue.acm.org/detail.cfm?id=2016038

[38] "Implementing Your BYOD Mobility Strategy." The Divide Platform Enables BYOD Mobility. Enterproid, 2012. Web. 14 Oct. 2012. http://www.divide.com

[39] Sven Bugiel , Lucas Davi , Alexandra Dmitrienko , Stephan Heuser, Ahmad-Reza Sadeghi , Bhargava Shastry, Practical and lightweight domain isolation on Android, Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices, October 17-17, 2011, Chicago, Illinois, USA

[40] Cabebe, Jaymar. "Android 4.2 Adds Multiple Users and Panoramic Photos, Copies Swype and AirPlay." CNET. N.p., 29 Oct. 2012. Web. 02 Nov. 2012. http://reviews.cnet.com/8301-19736_7-57542145-251/android-4.2-adds-multiple-users-and-panoramic-photos-copies-swype-and-airplay

[41] David Jaramillo, Robert Smart, Ankur Agarwal, Borko Furht, "A Secure Extensible Container for Hybrid Mobile Applications" IEEE Southeast Con 2013, (ID: 200)

[42] PhoneGap. http://phonegap.com, 2012.

[43] V. G. Sarah Allen and L. Lundrigan. "Pro Smartphone Cross-Platform Development" Apress, 2010.

[44] Omezzine, A.; Sami Yangui; Bellamine, N.; Tata, S.; , "Mobile Service Micro-containers for Cloud Environments," Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2012 IEEE 21st International Workshop on, vol., no., pp.154-160, 25-27 June 2012.

[45] David Jaramillo, Richard Newhook, Robert Smart, "Cross-Platform, Secure Message Delivery for Mobile Devices", IEEE Southeast Con 2013, (ID: 193)

AUTHORS' INFORMATION

[1]PhD Candidate at Florida Atlantic University
Boca Raton, Florida, USA

[2]Professor and Chairman of the Department of Computer & Electrical Engineering at Florida Atlantic University, Boca Raton, Florida, USA.

[3]Associate Professor at the Department of Computer & Electrical Engineering at Florida Atlantic University, Boca Raton, Florida, USA.

David Jaramillo is a Senior Technical Staff Member at IBM CIO Labs and is the Chief Architect for Enterprise Mobile Innovations. He is currently a PhD candidate in Computer Engineering at Florida Atlantic University where he also received his M.S. degree in Computer Science in 1990 and his B.S. degree in Mathematics and Computers in 1988. He joined IBM in 1992 to work on the IBM OS/2 Operating System, followed by working in IBM Embedded ViaVoice group where he delivered several Embedded Speech releases to major customers and produced the IBM Embedded Multimodal Speech Browser. Since then, he has been a member of the CIO Labs leading and transforming the way for Mobile Computing in the Enterprise. David is a Master Inventor at IBM with 26 patents issued in the US, 9 patents abroad and a number of published technical papers. He is also a member and contributor to the Institute of Electrical and Electronics Engineers.

Borko Furht is chairman and professor at in the Department of Computer & Electrical Engineering and Computer Science at Florida Atlantic University (FAU) in Boca Raton, Florida. He is also Director of the NSF-sponsored Industry/University Cooperative Research Center for Advanced Knowledge Enablement. Professor Furht received his Ph.D. degree in electrical and computer engineering from the University of Belgrade. His current research is in multimedia systems, video coding and compression, 3D video and image systems, video databases, wireless multimedia, and Internet computing. He has been Principal Investigator and Co-PI of several multiyear, multimillion dollar projects – on Coastline Security Technologies, funded by the Department of Navy, I/U CRC Center funded by NSF, One Pass to Production funded by Motorola, NSF PIRE project on Global Living Laboratory for Cyber Infrastructure Application Enablement, and High-Performance Computing grant from NSF. He is the author of numerous books and articles in the areas of multimedia, Internet engineering, computer architecture, real-time computing, and operating systems.

Ankur Agarwal is an associate professor of the Department of Computer & Electrical Engineering and Computer Science at Florida Atlantic University (FAU) in Boca Raton, Florida. He is Co-PI of Research Projects and also Assistant Director of Center of Systems Integration at FAU. Currently, he is President of MedSoftSys, Inc – a software development company engaged in developing cloud based enterprise applications and solutions. Professor Agarwal received his Ph.D. degree in Computer Engineering at FAU in 2006 with his dissertation on QoS Driven Design of NOC for Embedded Systems. His main areas of research include Medical Informatics System Design, Embedded & FPGA Design, Medical Device Design, Network on Chip & Multiprocessor Architectures and Real Time Systems & Concurrency Modeling. He has published more than forty papers (journals, conference papers and book chapters).