IT@Intel White Paper

Intel IT
IT Best Practices
Native Email Solution
February 2014

Enabling Native Email, Calendar, and Contacts on Android* Devices

Our native email solution enables employees to access their email, calendar, and contacts using the native applications on their Android devices.

Paul Donohue
Mobility Engineer, Intel IT

Rob Evered
Senior Information Security Specialist, Intel IT

Derek Harkin
Mobility Engineer, Intel IT

Aideen Prendergast
Project Manager, Intel IT

Emer Roche
Mobility Engineer, Intel IT

Executive Overview

As Intel's bring-your-own-device (BYOD) program grows and Android* devices become more popular, Intel IT wants to remove barriers to employees using these devices while also keeping the enterprise secure. We are accomplishing this goal by delivering a native email solution that enables employees to access their email, calendar, and contacts using the native applications on their Android devices. Our solution secures these devices to a high enough standard that employees can also use enterprise applications that require access to corporate data.

Adding support for native Android applications requires us to address three main challenges:

- **Security.** Prior to the release of version 4.0 in late 2011, Android did not have the encryption or two-factor authentication capabilities we require for enterprise access.
- **Fragmentation.** Device manufacturers may customize the Android OS, resulting in hundreds of variations.
- **Scalability.** Due to security risk and fragmentation implications, along with the increase of Android OS/device combinations, it was difficult to scale support for registering and configuring the devices.

Our native email solution, which encompasses email, calendar, and contacts, overcomes these challenges by doing the following:

- **Mitigating security risks.** We evaluated Android 4.0 and higher against a set of threats, vulnerabilities, and consequences and then developed mitigations that limited risk as much as possible. We also required our mobile device management (MDM) solution supplier to include specific certificate provisioning features to help reduce risk.
- **Accommodating multiple devices from different manufacturers.** Employees load the solution in two phases. First, they register their device with the MDM solution. Second, they manually configure their email account. The number of manual steps employees take depends on the type of device they are using—a result of Android OS fragmentation.
- **Enabling a majority of Android devices in use at Intel.** A majority of employees use an Android device from one of three categories: Intel® architecture-based devices, Android Open Source Project (AOSP) devices, and devices from a specific third-party supplier.
- **Assisting employees with setup.** Written instructions complement generic automated instructions that guide employees through the device registration and email account configuration process. This helps limit the number of incident tickets submitted to our IT help desk.

A survey conducted in mid-2013 of a sample of the first 6,000 employees using the native email solution revealed that 90 percent find it acceptable and 81 percent prefer working in the native applications to working in the secure container used at Intel since 2005. Besides satisfying our employees and allowing them to be more productive, we also realize a cost savings because MDM licenses are less expensive than secure container licenses.

Contents

IT@INTEL

The IT@Intel program connects IT professionals around the world with their peers inside our organization–sharing lessons learned, methods, and strategies. Our goal is simple: share Intel IT best practices that create business value and make IT a competitive advantage. Visit us today at www.intel.com/IT or contact your local Intel representative if you'd like to learn more.

BACKGROUND

At the end of 2011, there were about 6,000 Google Android* OS-based devices under management by Intel IT that used a secure container solution for email, calendar, and contacts only. Two years later, the number had nearly quadrupled. As part of our ongoing goal of providing an optimal user experience to employees, we wanted to enable this growing group to access their email, calendar, and contacts using the native applications on their Android devices—the way they wanted to—while maximizing information security and IT efficiency. This initiative would also provide baseline security for business applications beyond email, calendar, and contacts.

Providing an Android native email solution was a key initiative to support choice in IT consumerization. We wanted to support employees' freedom to choose without increasing Intel's risk. Our employees wanted to be able to quickly send an email, schedule a meeting, or join a conference call from their device of choice, including Android devices. The native email solution made this possible. It also provided a baseline of enterprise security on the device for business applications beyond email, a comparable native email solution to that delivered on devices that run a different OS, and a lower cost of IT management for Android devices.

Benefits of Securing Android Devices

After extensive analysis, we found that securing Android devices had many benefits:

- Delivering a native application experience allowed us to significantly reduce licensing expenditures on the secure container solution, or at least to repurpose the licenses for other less secure or less mature platforms.
- Employees using Android devices with the secure container solution couldn't experience the full range of enterprise services for email, calendar, and contacts. This limitation negatively impacted productivity, efficiency, and job satisfaction. However, enabling employees to see personal and corporate calendars using the native applications on their Android device helped with their work-life balance. They could more easily note availability of Intel colleagues to schedule events, which was a timesaving advantage for a global company with employees working across multiple time zones.
- Internally, the use of Intel® architecture-based Android devices was growing faster than the use of any other device/OS combination. To take advantage of reduced licensing expenditures, we wanted to provide the widest possible range of enterprise services to the increasing number of employees registering these devices with our BYOD program.
- Application development teams needed to be able to test on Intel architecture-based Android devices. We did not want a secure container solution to limit their ability to test applications in the native environment.
- Employees like the open source Android platform because it allows them to be innovative and creative. By limiting Android capabilities within a third-party secure container, we were hindering employees' ability to be innovative and creative on their platform of choice.
- Many employees who interfaced with wearable computing devices preferred to do so through the Android platform. For example, they wanted to receive new email or calendar notifications on smart watches, which would require the watch's native application to access email.

Challenges

Before we could create a supported, scalable, enterprise-level solution, we needed to fully understand the complex Android ecosystem and its challenges. We explored the challenges from three perspectives: security, fragmentation, and scalability.

SECURITY

In 2011, most Android devices at Intel were running version 2.3. Intel IT considered this OS to be an insecure platform for our environment, and it did not meet minimum requirements to enable native email, calendar, and contacts. Specifically, this version could neither protect data at rest (using encryption) nor support two-factor authentication for access to services within our enterprise.

To make it possible for employees to use their Android devices, we had to deliver email, calendar, and contacts to these devices through a secure, encrypted container. This compromise was the only way we could balance risk and reward; this solution delivered part of what employees wanted, but it was less than ideal for a number of reasons:

- Licensing for third-party secure containers was expensive.
- There was still some risk involved because we were using a secure container on an insecure platform.
- Secure containers did not allow the devices to run any applications that required corporate data access, so employees were limited in what they could do compared to what was available on mobile devices running a different OS that did not require secure containers.
- Gesture control, voice input, and other technologies that depended on device sensors did not work with the secure containers.
- Some secure containers allowed syncing to the corporate phone book, but that exposed more data to the unsecured Android platform, which increased risk.
- Managing a variety of business applications through secure containers was expensive and difficult, and it was not scalable.

The release of Android 4.0 in late 2011 included the additional enterprise security features we needed to deliver secure and cost-effective Android device management using native applications for email, calendar, and contacts, as our employees had been requesting. These new features also provided the foundation for securely enabling other Android business applications on managed devices. Managing the full device was more efficient and cost-effective than providing separate containers for multiple applications.

FRAGMENTATION

Fragmentation of the Android OS was another factor that prevented us from supporting native applications prior to version 4.0. Whenever a new Android OS version was released, each device manufacturer had the ability to modify the OS. This meant we had to manage more than 1,000 combinations of Android OS versions and devices. This amount of fragmentation was time-consuming and labor-intensive to support. We needed to create a solution that was scalable and manageable with minimal intervention by the IT help desk.

The following are other examples of the fragmentation factors we considered when developing the solution to deliver native email:

- Android 4.1.1 and Android 4.4.0 created some certificate-related problems for us pertaining to installation and interrupted email syncing.
- Different OS update release dates based on device model, country, and carrier made it impossible to test at the OS level alone.
- The steps required to configure the native email solution varied by OS/device combination.
- Device performance with the native email solution varied with the model, age, applications installed, and data usage.

SCALABILITY

The security and fragmentation complexities, along with the increase of Android OS/device combinations, made it challenging to create a solution that we could manage efficiently. We encountered three major issues:

- Device eligibility. Determining the eligibility of each Android device was difficult. We needed to assess each model/OS version combination to determine what it needed to meet the minimum security and technical requirements. In contrast, for mobile devices running a different OS, we could simply determine eligibility at the OS level, which allowed for complete automation of the device provisioning process.
- User instructions for manual processes. Because the registration process required employees to take manual steps, we needed to develop instructions and training material to support them.
- Scalability of support. Creating and maintaining more than 1,000 versions of instructions for each OS/device combination was not feasible. We needed a solution that was as generic as possible yet still detailed enough to enable employees to independently complete the configuration of their devices while helping to prevent our IT help desk from being overburdened with requests.

Table 1. Eligibility Requirements for Android* Devices. As part of the security risk assessment portion of the native email solution, we identify whether an employee's Android device meets these minimum qualifications.

Feature	Eligibility Requirement
Operating System	Must have Android* 4.0 or higher, including enforceable native encryption
Device Management	Must support Intel's MDM solution
Push Email	Must support two-factor authentication (username/password and certificates)
Encrypted Email	Must not be accessible
Malicious Code Defense	Must have additional malware controls in place and available
Rooted Devices	Must be blocked
Developer Mode	Must be blocked

ENABLING THE ANDROID NATIVE EMAIL SOLUTION

To address the challenges of delivering native email, calendar, and contacts to Android devices, we created a comprehensive enterprise-level solution that is supportable, financially accountable, and—perhaps most importantly due to the increasing popularity of Android devices in the enterprise—secure.

Our native email solution consists of four parts: a comprehensive security risk assessment of the Android device, a solution architecture that accommodates multiple devices from different manufacturers, in-depth analysis of our Android ecosystem to determine which devices fit the solution, and in-depth user training.

1. Assessing Security Risk

Before approving Android devices for the native email solution, we conduct a comprehensive security risk assessment of the OS. (Devices running versions prior to Android 4.0 are not eligible; see Table 1 for more device eligibility requirements.) This allows our native email solution to keep security risk at an acceptable level.

In order for an Android device to run the native email solution at Intel, we secure it to a specific standard as defined by our security trust model.[1] Internal applications other than email also require this standard, and we consider email as the foundation to establish it. After employees securely configure their device for email, they can access or download any applications that meet the standard. We evaluate Android 4.0 and higher versions against a set of threats, vulnerabilities, and consequences, ranging from stolen hardware to legal exposure to remote deletion capabilities. When we find high or medium risks, we develop mitigations to make the residual risks as low as possible.

WORKING WITH OUR MOBILE DEVICE MANAGEMENT (MDM) SUPPLIER

Our MDM supplier helps us further reduce any residual risks that exist with Android 4.0 and higher. We collaborate with the supplier to make sure it can do the following:

- Provision certificates to each device as employees configure their email.
- Provide reporting capabilities that can identify and notify employees of impending biannual certificate renewal to avoid disruption of service.
- Identify whether the device is rooted or has developer mode enabled.

BLOCKING ROOT ACCESS AND DETECTING DEVELOPER MODE

We block access to any device that has been rooted or is running in developer mode. Rooting (also known as jailbreaking) gives attackers privileged control (known as root access) within Android's subsystem. Rooting is often performed with the goal of overcoming limitations that carriers and hardware manufacturers put on some devices. It gives attackers the ability to alter or replace system applications and settings, run specialized applications that require administrator-level permissions, or perform other operations that are otherwise inaccessible. On Android devices, rooting also enables the complete removal and replacement of the device's OS.

[1] For a complete explanation of the security trust model and how an Android device is assessed, see these IT@Intel white papers: "Android* Devices in a BYOD Environment," "Maintaining Information Security While Allowing Personal Hand-Held Devices in the Enterprise," and "Granular Trust Model Improves Enterprise Security."

In general, rooting enables full access to all applications and application data. Attackers who change the permissions on an Android device to grant root access to applications can increase the security exposure to malicious applications and potential application flaws.

We also detect whether devices are running in developer mode to reduce security exposure to malicious applications. The developer mode setting on an Android device allows the device to connect using a toolkit called the Android Debug Bridge (ADB). In developer mode, attackers can bypass the full volume encryption by simply connecting to the device via USB and ADB. It is also possible for an attacker to circumvent the device's PIN lock if it is in developer mode. If an employee's Android device is in developer mode, we ask that it be switched off unless the employee is using it for a job-related application.

We select our MDM solution supplier based on its ability to collect information about each device. We also analyze our risk exposure and identify devices that are rooted or running in developer mode. Depending on the level of risk revealed by the MDM, we may need to take further action—such as communication, training, or removing access.

2. Configuring the Solution

Employees load the native email solution in two phases (Figure 1). In phase 1, employees register their Android device with our MDM solution and get the security risk assessment. In phase 2, employees manually configure their email account.

After employees register their devices, our MDM solution secures the device by conducting the security risk assessment detailed previously, implementing the necessary controls to comply with the security policy, and enabling delivery of public key infrastructure (PKI) certificates.

After the MDM solution delivers the certificates, the employee manually configures the email account settings on the device. The device then accesses the employee's Microsoft Exchange* mailbox through a web application firewall, authenticates the account on the RADIUS server, and then connects to the mailbox.

3. Selecting Which Android Devices Fit the Solution

The Android platform is impacted by fragmentation with differences by manufacturer, OS version, model, country and carrier. These factors can lead to inconsistency across Android devices in several ways: performance, frequency of update release dates, and customization of OS updates. Therefore, no one-size-fits-all native email solution is applicable. Our native email solution works for a majority of the Android devices used at Intel. We identified this majority by segmenting Android devices according to OS/manufacturer/model groups and then forecasted which groups will see the most increase in the number of devices. The three groups we prioritized are: Intel architecture-based devices, Android Open Source Project (AOSP) devices, and devices from a specific third-party supplier.

Intel architecture and AOSP devices use the same native email solution. Devices from the third-party supplier require a different certificate provisioning and email account configuration process. However, additional APIs in the third-party supplier's device do allow for a more streamlined setup process compared to Intel architecture and AOSP devices.

By focusing on these three groups, our native email solution is usable by a majority of employees using Android devices.

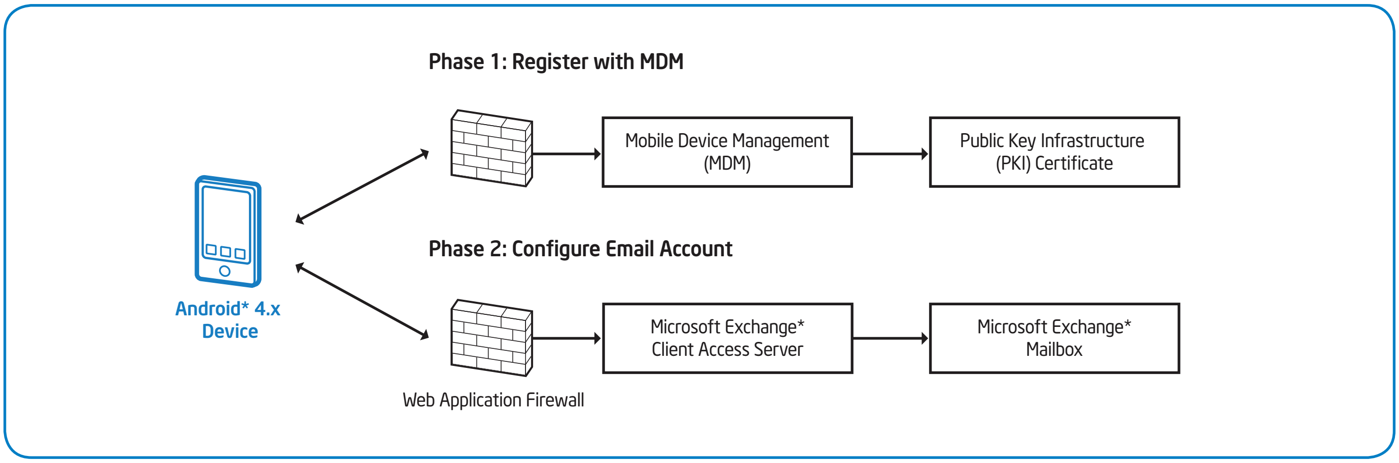
Phase 1: Register with MDM
Mobile Device Management (MDM)
Public Key Infrastructure (PKI) Certificate
Phase 2: Configure Email Account
Android* 4.x Device
Microsoft Exchange* Client Access Server
Microsoft Exchange* Mailbox
Web Application Firewall

Figure 1. Native email solution architecture. Employees register their Android* devices with the mobile device management (MDM) solution to get a security risk assessment, and then they configure their email accounts.

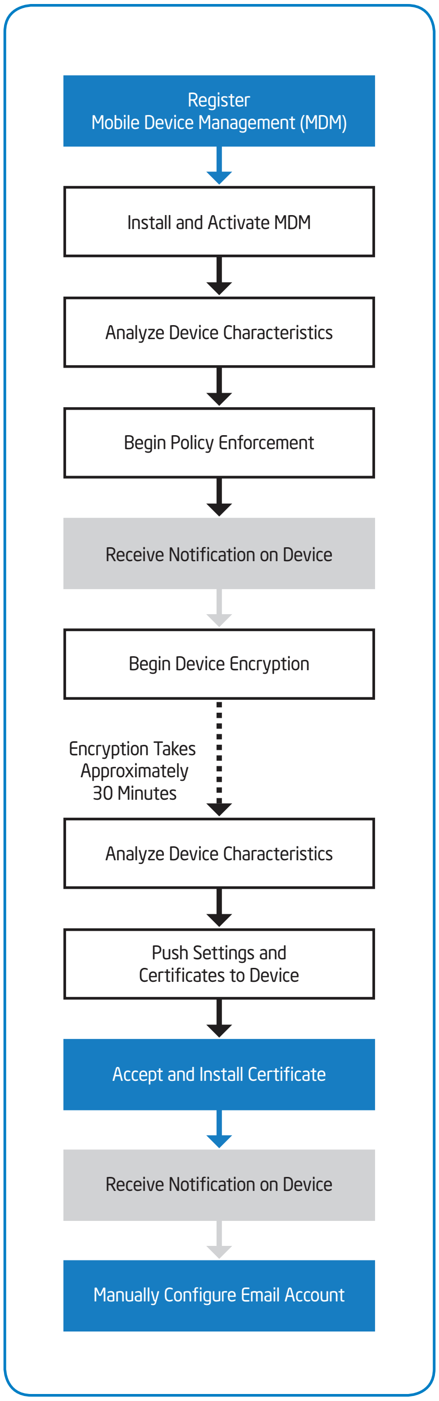
Register
Mobile Device Management (MDM)
Install and Activate MDM
Analyze Device Characteristics
Begin Policy Enforcement
Receive Notification on Device
Begin Device Encryption
Encryption Takes Approximately 30 Minutes
Analyze Device Characteristics
Push Settings and Certificates to Device
Accept and Install Certificate
Receive Notification on Device
Manually Configure Email Account

Figure 2. Email solution setup process for Android* devices. Employees take manual steps to register their devices, accept certificates, and configure their email accounts.

4. Training Users

Because employees take manual steps to configure their Android device, we have made the process illustrated in Figure 2 as intuitive as possible. Automated prompts guide employees through part of the process of registering their device with the MDM solution and configuring their email account. But the prompts need to remain generic to accommodate multiple devices from different manufacturers. Employees can download more comprehensive written instructions for their OS/device combination from our internal mobile devices website if the generic instructions don't fulfill their needs.

RESULTS

About 6,000 Android devices under management use the native email solution, and we expect that number to increase since there are Android devices still being managed with the secure container solution and the popularity of Android devices is rising. According to a mid-2013 survey of employees using the native email solution, 90 percent found it acceptable, and 81 percent preferred working with the native email solution to the secure container user experience on their Android devices.

Since our MDM licenses cost less than secure container licenses, we are realizing a large cost benefit. That benefit is compounded by the fact that since those 6,000 Android devices were moved to the native email solution, the secure container licenses were available for use. This enabled us to repurpose approximately 3,000 secure container licenses to less secure Android devices that are not yet eligible for the native email solution.

By enabling the native email experience on Android devices, we also secure them to natively access other services and applications, eliminating more expensive secure containers and improving the user experience in the process. The 6,000 Android devices running the native email solution are also eligible for the following:

- A proprietary application for conference calls
- Instant messaging
- Business applications used by factory, sales and marketing, and other Intel business units

As we continue to develop user training for the solution, we expect requests for instructions and general questions about the setup process to decrease. Incident tickets relating to Android device registration at the IT help desk are already within an acceptable range compared to devices running a different OS.

We anticipate that future versions of the Android OS will show improved certificate management and email provisioning to simplify the device setup process. In collaboration with our MDM supplier, we will continue to remove usability barriers such as the complex setup process and six-digit password requirements. In fact, as dual-persona[2] solutions mature, employees will be able to use the personal environment on their devices without being impacted by corporate device management policies such as encryption or setting up a six-digit password. Until then, we continue to educate the IT help desk on how best to resolve these issues with employees.

[2] Dual-persona solutions enable us to manage separate work and personal environments on an employee's device in the BYOD program.

CONCLUSION

To support employees' freedom to use their devices of choice while we work to keep the enterprise secure, we developed a solution that allows employees to access their email, calendar, and contacts using the native applications on their Android devices. The native email solution addresses Android enterprise security, fragmentation, and scalability challenges that limit the use of the Android OS as a secure enterprise platform. Our solution secures the Android device to a high enough standard to enable other enterprise applications and services to improve the employee user experience.

Prior to the deployment of this solution, we managed email and other services on Android devices through secure containers. Working in the devices' native environment saves us the expense of deploying the secure container, improves the user experience, and gives employees the freedom to use Android devices at work more productively.

The initial solution works for a majority of the Android devices used at Intel. We will continue to expand the solution to include other Android devices as more manufacturers make advancements in certificate management and email provisioning functionality. These improvements will also help reduce the number of manual steps required to configure the native email solution, which will in turn reduce the number of support calls to the IT help desk. Our goal is to efficiently support as many employees as possible by providing access to a greater range of devices through our BYOD program.

FOR MORE INFORMATION

Visit www.intel.com/IT to find content on related topics:

- "Android* Devices in a BYOD Environment"
- "Maintaining Information Security While Allowing Personal Hand-Held Devices in the Enterprise"
- "Granular Trust Model Improves Enterprise Security"

ACRONYMS

ADB	Android Debug Bridge
AOSP	Android Open Source Project
BYOD	bring your own device
MDM	mobile device management
PKI	public key infrastructure

For more information on Intel IT best practices, visit www.intel.com/IT.

THE INFORMATION PROVIDED IN THIS PAPER IS INTENDED TO BE GENERAL IN NATURE AND IS NOT SPECIFIC GUIDANCE. RECOMMENDATIONS (INCLUDING POTENTIAL COST SAVINGS) ARE BASED UPON INTEL'S EXPERIENCE AND ARE ESTIMATES ONLY. INTEL DOES NOT GUARANTEE OR WARRANT OTHERS WILL OBTAIN SIMILAR RESULTS.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

Printed in USA ♻ Please Recycle 0214/JGLU/KC/PDF 329834-001US