Mobile Application Management

Technical White Paper

omnissa™

Contents

Introduction

Many enterprises deploy Mobile Device Management (MDM), or Mobile Application Management (MAM), or both.

MDM enables the enterprise to enroll, track, monitor, lock, encrypt, wipe data, and enforce security policies on mobile devices. As such, MDM might be inappropriate for devices that can access both personal and enterprise data. Employees who bring their own devices to work expect that their personal data and applications remain private.

MAM, in principle, seeks to support the same controls as MDM, but limited in scope to enterprise data and applications. In practice, the controls offered by MAM and MDM solutions aren't exactly equivalent. Analysis is necessary to understand the differences, and how they meet requirements for enterprise mobility.

Background

The work of the modern enterprise relies upon access to data and services. This applies to all enterprises, whether commercial, charitable, governmental, utilities, or in other fields. Access will be required by staff who are

- Stationary at their desks, tills, or other workstations.
- Moving around the office, branch, store, or other enterprise premises.
- Visiting customers, clients, or prospects; or installing or maintaining equipment; or otherwise away from enterprise premises.

To cover all of these cases, the enterprise requires that access to its data and services is made mobile.

Mobile workers will require access to some or all of the following types of enterprise data as end users.

- Email, contacts, calendar, and other personal information management (PIM) data.
- Instant messaging.
- Intranet content.
- Web applications that run in a browser.
- Services utilized via custom client applications.

Access can be provided by mobile applications running on a smartphone, tablet, or similar device. Separate applications might be needed, for example to send and receive different types of data, to access services, or to view particular file formats. Whatever the reason, end users will expect that all their applications work together in a seamless manner, as a suite, to mobilize the data that they need to do their jobs.

The enterprise will require that mobilized data is protected against accidental leakage, and against deliberate attack.

Requirements

Requirements for enterprise data mobilization and protection can be categorized as follows.

Protected Data at Rest

- Requirements for protection of data at rest on the mobile device may include the following.
- Authentication of the user at the device by an unlock interaction, such as entering a passcode.
- Inactivity monitoring that implements an idle time-out, after which authentication of the user at the device is considered to have expired.
- Encryption of data at rest on the mobile device.
- Remote data management commands, for example to wipe the enterprise data on the device.
- Specification and handling of enterprise compliance policies such as
    - Which applications are allowed on the device, and which versions.
    - Whether the camera can be used.
    - Whether rooting, for Android, or jailbreak, for iOS, is allowed. This implies a requirement for device compromise detection.
    - Which operating systems and versions are allowed.
- Applying an offline access limit after which contact with the enterprise is required, for applications that store data locally.
- Preventing leakage of on-screen data by screen capture or exposure in the task switcher. Other data leakage prevention requirements are described in the Protected Data in Motion section.

Protected Data in Transit

Requirements for protected data in transit between the mobile device or application and the host of the data or service may include the following.

- Host access route from mobile applications to enterprise data and services.
- Authentication of the end user at the host.
- Storage and use of electronic certificates for private infrastructure access, end user authentication, or other purposes.

Protected Data in Motion

Requirements for protection of data in motion between mobile applications may include the following.

- Encrypting data during inter-application communication.
- Encrypting data in cut-copy-paste interactions.
- Limiting which applications can communicate with enterprise applications.

These are all **data leakage prevention** requirements, as is screen capture prevention, which was previously mentioned.

Usability

Specific usability requirements may include the following.

- Loading the **application configuration** with server addresses and other values specific to the enterprise, so that end users do not have to do this by hand. AppConfig is a standard for this type of configuration.

Usability also applies in general across all the preceding requirements.

Workers in the modern enterprise are often resourceful, tenacious, and creative people. They will find ways to mobilize the enterprise data on which their success depends, regardless of whether an enterprise mobility management solution is in place. Also, if the solution in place is too limited, too difficult to use, or too intrusive, the intended end users will bypass it.

A similar analysis can be applied to the people that operate and administer the solution. Their user interface must also be easy to use, either natively or by integration into an administrative platform. They may also require that there is a self-service console through which end users can manage their own day-to-day tasks.

Summary

The following are all required for mobile access to enterprise data and services.

- Protected data at rest.
- Protected data in transit.
- Protected data in motion.
- Usability.

Solutions

Solutions for mobile access to enterprise data and services will address the requirements in the preceding section. The requirements may be addressed in different ways and there are different types of solution.

Types of Solution

The following are some types of solution for mobile access to enterprise data and services.

- **Mobile Device Management (MDM)** solution, in which a management console takes control of the end-user mobile device.
- **Mobile Application Management (MAM)** solution, in which a management console takes control of selected applications on the end-user mobile device.
- **Non-integrated** solution, in which there is no management console and the enterprise does not take control of the device nor of any mobile applications.

Some products support more than one of the above types of solution. An enterprise can deploy multiple solutions, and multiple types of solution.

Notes:

- MDM solutions require privileged access to the device, either out of the box, or after market and granted by the end user. MAM and non-integrated solutions typically require fewer or no privileges.
- MDM and MAM solutions in general include a runtime library that can be integrated with a mobile application at build time. The library might be packaged as a software development kit (SDK), for example. Integration of the library is not necessarily mandatory for all applications in scope of the solution.
- MDM solutions that do not have an SDK may be called Pure MDM.
- Some devices support an enterprise partition, to which selected applications can be assigned. Applications assigned to an enterprise partition will be isolated from other applications on the device. The Android Enterprise work profile is an example of an enterprise partition.

Some typical features of an enterprise partition are as follows.

- The partition has a separate unlock interaction. If the user has unlocked the device, but has not unlocked the partition, then none of the applications in the partition can run.
- Applications inside the partition cannot utilize inter-application communication, such as Android Content Provider, with applications outside the partition.
- If an application inside the partition writes credentials to a shareable location, these credentials cannot be read by an application outside the partition.
- Applications inside the partition have access to a virtual private network.
- The enterprise can select which applications get installed to the partition.
- The contents of only the partition can be erased using remote data management.

In summary, an enterprise partition can be a solution for some unlock, data management, compliance, and data leakage requirements.

Solutions that create an enterprise partition are within the MDM type. This is because partition creation will require MDM privileges on the device.

- A solution could be based on a mobile application that communicates with an Internet service that can be configured by an enterprise administrator. For the purposes of this discussion, that counts as a MAM solution.

Solutions of each of these types will offer capabilities that meet some or all of the requirements, but may involve the end user or the enterprise in administrative tasks such as enrollment.

Enrollment

MDM and MAM solutions require enrollment as a first step, also called onboarding. Enrollment is the establishment of a connection with the enterprise's management console. Depending on the solution, the connection is either between the device and the management console, or between the application and the management console, or both.

Some common enrollment mechanisms are as follows.

- Pre-enrollment, also known as *out of box*, in which the mobile device has been allocated to the enterprise at some point in the device's manufacture or packaging.
- Entry of enrollment credentials in a user interface that is part of the operating system.
- Entry of enrollment credentials in a dedicated MDM or MAM endpoint application, sometimes referred to as an agent or device administrator.
- Entry of enrollment credentials in an enterprise application that has integrated the SDK of the MDM or MAM solution. The application could be an email client, for example.
- Facilitated enrollment by delegation to an application on the device that has already been enrolled using another mechanism.

Not all products support all mechanisms.

Capabilities Comparison Table

The following table compares the typical capabilities of the different types of solution in terms of how they meet requirements for mobile access to enterprise data and services. The requirements themselves are discussed in the introduction, in the Requirements section.

REQUIREMENT	PURE MDM	MDM OR MAM WITH SDK	NON- INTEGRATED
Unlock *	The enterprise can • Require a device level passcode. • Require a passcode for an enterprise partition. • Specify complexity rules and allowed types for the passcode.	The enterprise can • Require an application-level passcode. • Specify complexity rules and allowed types for the passcode. The same passcode can be shared by a suite of applications.	The enterprise cannot configure unlock.
Inactivity monitoring	The enterprise can restrict the length of the sleep time-out at the device level.	The enterprise can impose an inactivity timer: • On one application. • That is shared by a suite of applications.	The enterprise cannot configure inactivity monitoring.
Encryption of data at rest	The enterprise can force encryption of the whole device, or of the enterprise partition.	Encryption of application data is facilitated by the SDK.	Encryption of data at rest isn't facilitated nor configurable by the enterprise.
Data Management Commands	The enterprise can • Wipe all data from the device. • Uninstall selected applications.	The enterprise can wipe data from enterprise applications.	The enterprise cannot send data management commands.

REQUIREMENT	PURE MDM	MDM OR MAM WITH SDK	NON- INTEGRATED
Compliance	The enterprise can • Force or block installation of selected applications. • Force or block operating system updates and security patches. • Block use of the camera, and other device features. Compliance can be enforced on the whole device, or only in the enterprise partition.	The enterprise can • Facilitate installation of recommended applications. • Block enterprise applications if operating system or security patch level doesn't comply. • Block enterprise applications if location services are switched off. Current compliance restrictions are available from the SDK.	The enterprise cannot configure compliance.
Device Compromise Detection	An agent application, if installed, will detect compromise.	Any SDK application will detect compromise.	The enterprise cannot detect device compromise.
Offline access limit	The enterprise can track last connection time of the device.	The enterprise can configure a limit, which is then imposed on applications by the SDK.	The enterprise cannot limit offline access.
Host Access	The enterprise can set up a virtual private network (VPN) at the device level, or in the enterprise partition. The VPN would be configured by a VPN client built in to the device operating system, or by a custom VPN app. The enterprise can specify which mobile applications have access to the VPN.	A proprietary proxy endpoint can be set up at the enterprise. The SDK then enables application tunnel connections to enterprise hosts via the endpoint. The enterprise can specify what tunnel connections are allowed.	Enterprise servers that host data or services must be open to the public internet.

REQUIREMENT	PURE MDM	MDM OR MAM WITH SDK	NON- INTEGRATED
Authentication of the end user at the host **	The enterprise can allow and configure any of the following: • Login credentials. • Proxy authentication by custom VPN. • Certificate-based authentication from the device store.	The enterprise can allow and configure any of the following: • Login credentials. • Proxy authentication by tunnel. • Certificate-based authentication for each application.	The enterprise must allow login credentials.
Electronic Certificate Storage ***	The enterprise can manage the device store, and can utilize electronic certificates signed by a private authority.	The enterprise can • Manage certificates sent to each application, which can include electronic certificates signed by a private authority. • Share certificates between a suite of applications.	The enterprise cannot control electronic certificates.
Data Leakage Prevention	The enterprise can • Create an enterprise partition. • Block device feature usage	The enterprise can configure the following for implementation by the SDK • Encryption of data during inter-application communication. • Encryption of data in cut-copy-paste interactions. • Limits to which applications can communicate with enterprise applications.	The enterprise cannot configure data leakage prevention.

REQUIREMENT	PURE MDM	MDM OR MAM WITH SDK	NON-INTEGRATED
Usability	Enrollment is required. A high level of privilege must be granted to the enterprise, which may be seen as intrusive by end users. Devices might issue repeated warnings about certificate installation, network traffic interception, and other intrusions. Administrators with more control over end user devices therefore have more responsibility. After enrollment and granting of privileges, the need for re- authentication may be reduced or waived.	Enrollment is required. A lower level of privilege, or none, need be granted to enterprise applications. Administrators with less control therefore have less responsibility. Authentication sessions and the inactivity timer are shared between applications.	No enrollment. A lower level of privilege, or none, need be granted
Application Configuration	Standard AppConfig, so the end user doesn't have to enter, for example, a server address.	SDK provides AppConfig compatibility.	The end user must enter any settings by hand.

Table 1: Comparison of Meeting Enterprise Mobility Requirements by Type of Solution

Notes on the capabilities comparison table:

*Unlock **

Under any solution, a mobile application might be able to

- Detect whether a device passcode or similar security has been set.
- Block the application user interface if a device passcode hasn't been set.
- Force the user to repeat their device-level unlock interaction.

The application would depend on support from the mobile device operating system for any of these capabilities. Support could differ between different versions of the operating system.

In a non-integrated solution, use of those capabilities is at the option of the application developer and isn't configurable by the enterprise.

*Authentication of the end user at the host ***

Under any solution, a mobile application could

- Accept login credentials entered by the end user.

- Utilize credentials stored in the browser.

Either of these might be undesirable from a security point of view.

Electronic Certificate Storage ***

Any application can add electronic certificates to the device certificate store. The device operating system might allow

- Direct addition by any application.
- Indirect addition through a native application with privileges, such as iOS Safari.

The operating system might warn the end user at the time of adding the certificate, and subsequently, for example, every time the device is switched on.

Capabilities Comparison Diagrams

The following diagrams illustrate some differences between MDM, MAM, and non- integrated solution types.

Applications and Configuration

The following diagram illustrates some differences between the MDM and MAM solution types' capabilities for application installation and configuration.

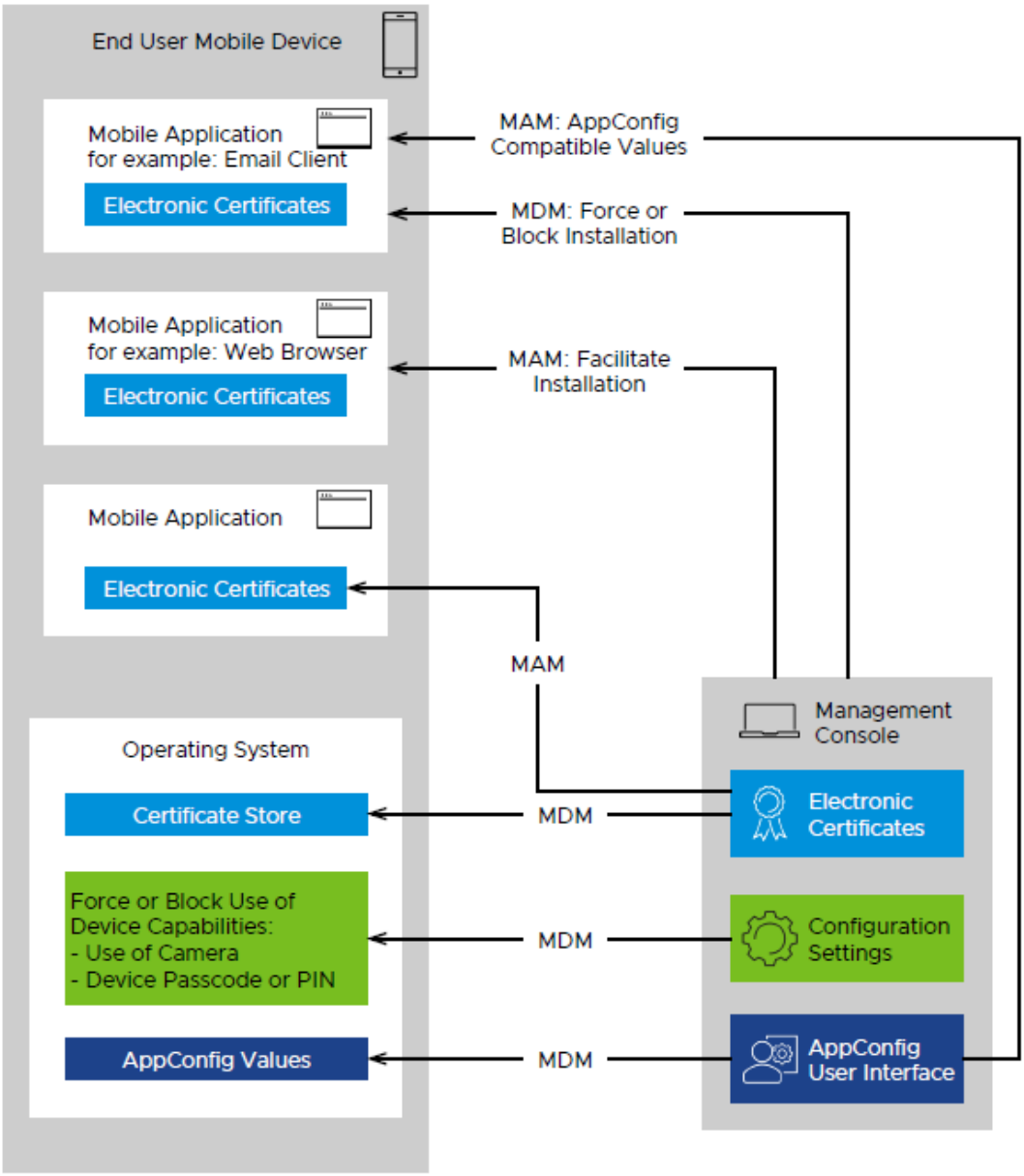

Figure 1: Differences in Application Installation and Configuration Between Mobile Application Management and Mobile Device Management

Host Access

The following diagram illustrates some differences between the MDM, MAM, and non- integrated solution types' capabilities for host access.

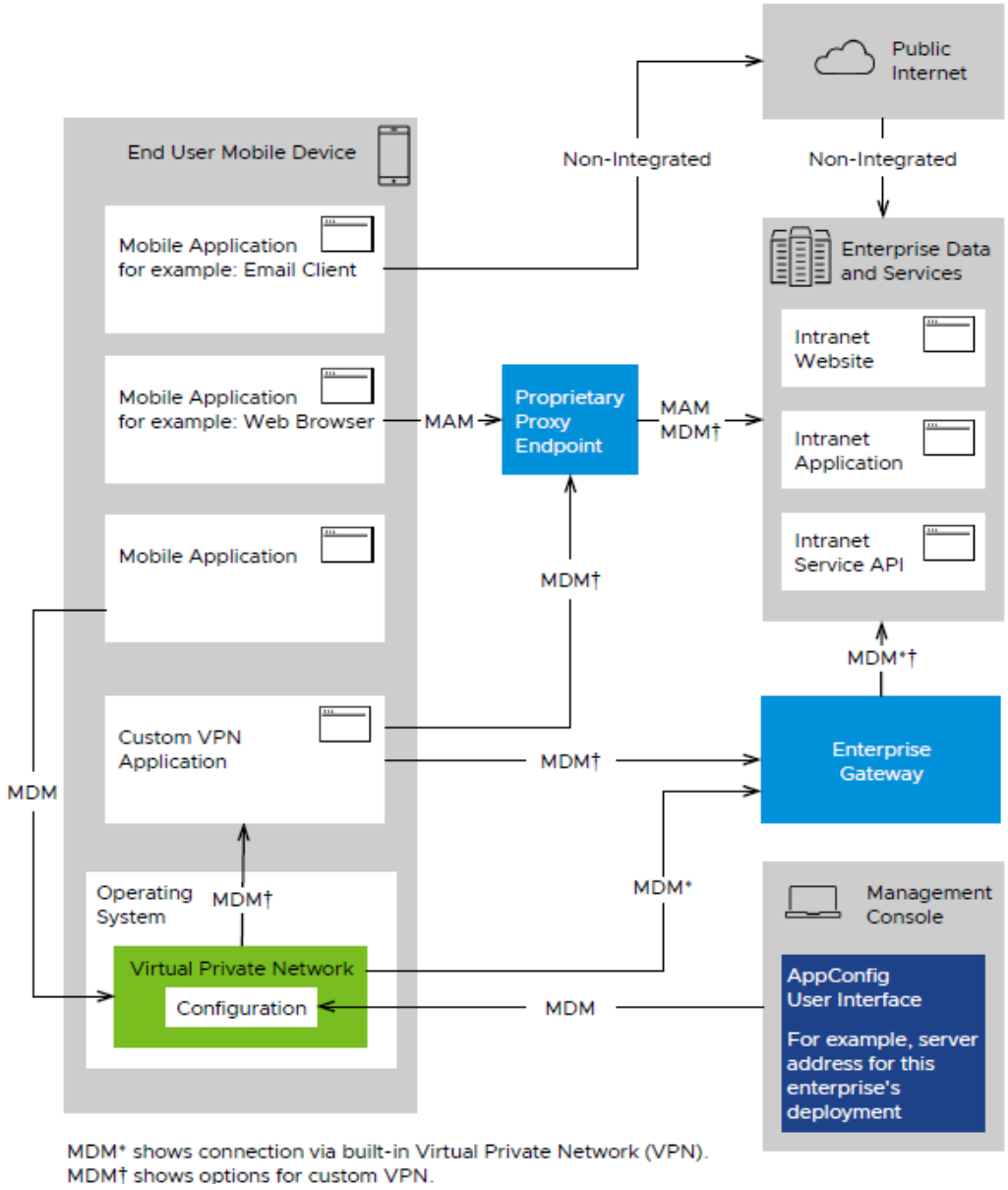

Figure 2: Differences in Host Access Capabilities Between Mobile Application Management, Mobile Device Management, and Non-Integrated Solutions

Omnissa Workspace ONE Platform

The Omnissa Workspace ONE platform is a solution for the mobilization of enterprise data and services. Workspace ONE has both Mobile Device Management (MDM) and Mobile Application Management (MAM) facilities.

Workspace ONE Components

The components of Workspace ONE are as follows.

UEM

The Workspace ONE Unified Endpoint Manager (UEM) is the management console, for both MDM and MAM enrollment. Each enterprise has its own instance of the UEM and sets up their own policies and configurations.

The UEM provides either a Managed (MDM) or Unmanaged (MAM) SDK profile to each mobile endpoint as part of enrollment. The selection of Managed or Unmanaged is made according to rules that are configurable by the enterprise and can be different for different end users and devices.

In the typical case, the first enterprise application to enroll is the Hub, as follows.

Hub

The Workspace ONE Intelligent Hub is a mobile application that serves as the main endpoint for MDM and MAM on the device. When Hub enrolls with the UEM, it receives either a Managed or Unmanaged SDK profile.

Hub serves as an anchor application, as follows.

- Hub shows the enterprise catalog and installs other applications from it.
- Hub facilitates enrollment of other applications, by providing credentials and the address of the UEM server. In some cases, enrollment of subsequent applications won't require the end user to enter credentials.

In the Managed case, on an Android device, Hub runs with some device administrator privileges. Those privileges are used to manage configuration of, for example

- Device-level compliance.
- Device-level Virtual Private Network (VPN).
- Device-level electronic certificate store.
- AppConfig values.
- Enterprise partition.

Hub is a replacement for the legacy Workspace ONE mobile application.

Tunnel App

The Omnissa Workspace ONE Tunnel App is a custom VPN provider and can be configured by a Managed profile. The Tunnel App connects the Omnissa Unified Access Gateway.

SDK

The Workspace ONE mobile software development kit (SDK) is a software library for integration of mobile applications with the Workspace ONE platform.

The SDK includes the following.

- User interfaces for common Workspace ONE tasks such as
    - UEM enrollment.
    - Application unlock.
- Informative programming interfaces for access to settings such as
    - Compliance.
    - Data leakage prevention.
    - Application configuration.
- Access to enterprise servers that host data and services via the SDK Tunnel.
- Implementation of
    - Device compromise detection.
    - Encryption of data at rest.
    - Policy enforcement.

See also the more detailed list under Software Development Kit Features, below.

The Workspace ONE Productivity Apps and the Hub are built with the SDK. The SDK is available for Android and iOS.

Omnissa Workspace ONE Productivity Apps

The Workspace ONE Productivity Apps are a suite of mobile applications for access to enterprise data and services. The suite includes the following.

- Omnissa Workspace ONE® Boxer for email, contacts, calendar, and other personal information management (PIM) data.
- Omnissa Workspace ONE® Web for intranet browsing and web applications.
- Omnissa Workspace ONE® Content for secure access to content repositories and files.
- Omnissa Workspace ONE® Send for securely sharing documents between Workspace ONE and Office 365 applications.

Applications in the suite make full use of the Workspace ONE mobile SDK and MAM integration. The suite is available for Android and for iOS.

Customer and Partner Apps

A mobile application that has integrated the Workspace ONE SDK, but which isn't one of the Workspace ONE Productivity Apps, is in one of the following categories:

- *Customer App*, if written by a developer team working at the same enterprise that has deployed the UEM.
- *Partner App*, if written by an independent software vendor (ISV), system integrator, or other third party.

Customer App and Partner App developers can make use of Workspace ONE SDK features in their applications.

Applications in either category mustn't be installed before Hub has enrolled.

Non-Integrated Apps

Mobile applications that haven't integrated the SDK can still be part of a Workspace ONE solution.

Non-integrated applications can, for example

- Be presented to the user and installed by Hub.
- Be installed to the enterprise partition, if in use.
- Connect to enterprise servers via a device-level VPN.
- Read AppConfig values, if an MDM profile is in effect.

Workspace ONE Administrative Features

At the point of enrollment, the UEM selects a profile based on a number of criteria, including the following.

- Organization group membership of enrolling end user.
- Operating system of the enrolling device.
- Which application is enrolling.

The profile specifies the following:

- Management type, either Managed (MDM) or Unmanaged (MAM).
- Applications that can and cannot be installed, in terms of
    - Mandatory applications.
    - Prohibited applications.
    - Recommended applications.
    - Allowed application versions.
- How the end user can be authenticated at the device, and whether they can use biometric authentication.
- Compliance restrictions, which can include the following:
    - Which versions of the mobile operating system, Android or iOS, are allowed.
    - The minimum allowed security patch level, if Android.
    - Whether running on a compromised device is allowed.
- Offline access time-out duration.
- Data leakage prevention (DLP) measures.
- How connection should be made to servers that host enterprise data and services, sometimes referred to as *tunnel configuration*.
- How the end user can be authenticated at the enterprise host.
- Electronic certificates.
- The UEM can store certificates for each end user. The UEM can also generate some types of certificate on demand.
- Branding of the SDK user interface.
- Logging detail level.

The SDK or Hub implements the profile on the mobile device.

Workspace ONE Software Development Kit Features

The Workspace ONE mobile SDK is a software library for integration of mobile applications with the Workspace ONE platform. Its features are available under either type of profile, Managed (MDM) and Unmanaged (MAM), and are as follows.

Enroll

The SDK includes a user interface for UEM enrollment. The SDK can enroll using any of the following mechanisms.

- One-time token, obtained out-of-band from the UEM, for example by email or read from the UEM self-service user interface.
- Security Assertion Markup Language (SAML) assertion.
- Lightweight Directory Access Protocol (LDAP) user name and password, sometimes referred to as domain credentials.
- Delegation to Hub, sometimes referred to as single sign-on (SSO).

Whichever mechanism is used, the outcome is that the application has a relationship with the enterprise, and the SDK receives a profile selected by the UEM.

Apply Profile

Depending on the profile, the SDK will do some or all of the following.

- Check application entitlement.

  The UEM can specify which applications the user is allowed to have. The UEM can also specify a range of allowed versions for each application. The SDK checks that the current application and version is allowed. If the current application isn't allowed then enrollment fails at this point.
- Store branding resources.
  The UEM can send brand resources, such as logo images, for customization of the SDK user interface. The resources are also available for the application via a programming interface in the SDK.
- Set up unlock credentials.
  The profile can specify one of the following.
  - The user must set a password, personal identification number (PIN), or other passcode. The specification includes passcode complexity requirements, which the SDK enforces.

    There might be another Workspace ONE mobile application on the device that has already set up a passcode, and that can share credentials with the just-enrolled application. In this case a passcode-sharing relationship is set up, instead of a new passcode.
  - The user unlocks by entering their user name and password.
  - The user doesn't need to unlock the application. This option could be selected in the case that a device-level unlock is already enforced, by MDM.
- Set up data-at-rest encryption.
  After unlock credentials have been set up, encryption of data at rest can be set up. See the Workspace ONE Encryption of Data at Rest section for a description.

- Store electronic certificates.
  The UEM can send one or more electronic certificates to the application. These are received and stored by the SDK as follows.
    - On Android, certificates can be stored in the device certificate store, by the Hub, if the profile type is MDM. Other applications store certificates in their own encrypted data areas.
    - On iOS, certificates can be stored in a shared keychain, if the application is a member of an access group, or in the application's own keychain otherwise.

      See also Workspace ONE Enterprise Server Connections, for how certificates are used.
- Check Compliance.
  The SDK checks that the application meets the compliance conditions sent by the UEM. For a list of supported conditions, see the Workspace ONE Administrative Features section, under Compliance restrictions.
  The profile also specifies the action to take in case the device or application doesn't comply: either to lock the user interface or to wipe the application data.
- Store application configuration values from the UEM, either AppConfig values or UEM Custom Settings values.
  After the profile has been applied, the SDK handles the ongoing protection of data at rest and some other maintenance, and enables use of the SDK Tunnel for connection to enterprise servers. See the following for details.

Protect Data at Rest

Going forward, the SDK will protect data at rest by doing some or all of the following.

- Displaying its unlock user interface when authentication is required.

The user interface offers biometric unlock, if it has been set up on the device and is allowed in the profile from the UEM.

- Monitoring user inactivity.

The SDK locks the user interface when the user hasn't been active in the application for an extended period. Locking is managed by a timer that is restarted whenever user activity is detected. The duration of the timer is specified by the UEM.

In some cases, the inactivity timer is shared:

- All Workspace ONE applications on an Android device share one timer.
- Workspace ONE applications that can access the same shared keychain on an iOS device share one timer. In practice, this means only the Workspace ONE Productivity Apps.

Activity in an application that shares its inactivity timer will restart the timer for all other applications that share the same timer.

Note that inactivity monitoring by the SDK doesn't apply in the case that the user doesn't have to authenticate at the application.

- Showing a background guard screen.

  The SDK superimposes a guard screen on the user interface when the application moves to the background. This prevents any application data that was on-screen at the time of transition from being exposed in the device's task switcher.

The SDK guard screen can be deactivated in case the application implements its own guard screen.

- Preventing data leakage.

  The SDK applies any data leakage prevention (DLP) measures specified by the UEM. These can include the following:

  - Encrypts data being written to the clipboard by a cut or copy action.
  - Decrypts data being read from the clipboard by a paste action.
  - Prevents screen capture, on Android.

    (The term pasteboard is sometimes used instead of clipboard.)

  The encryption and decryption of clipboard data depends on a key being shared:

  - All Workspace ONE applications on an Android device can share one clipboard key.
  - Workspace ONE applications that can access the same shared keychain on an iOS device share one clipboard key. In practice, this means only the Workspace ONE Productivity Apps.

    The UEM can specify a list of allowed applications for Open-In types of interaction. The SDK makes this list available so that it can be enforced by the application.

- Handling data management commands from the console.

  The UEM can send data management commands, such as enterprise wipe, which is an instruction to delete the stored data for the application. The SDK receives and executes these commands.

Maintenance

Going forward, the SDK will handle the following maintenance tasks as needed.

- Applying profile updates.

  The enterprise can from time to time change policy settings, such as password complexity requirements. Changes are made in the UEM, and then received by the SDK instance in an application.

  The SDK handles changes by, for example

  - Forcing the user to change their unlock passcode to meet new complexity requirements.
  - Locking the application user interface if the device isn't running an allowed operating system according to new compliance rules.

  When the SDK detects a compliance violation, it will attempt to notify the UEM. If it succeeds, then the UEM can restrict the end user's access to enterprise services, or take some other server-side actions. Whether notification succeeds or not, the SDK takes a compliance action at the device. The action will have been specified in the UEM profile: either to lock the user interface or to wipe the application data.

- Updating data-at-rest encryption.

  This is necessary when, for example, the end user changes their passcode. See Workspace ONE Encryption of Data at Rest for a description.

Workspace ONE Enterprise Server Connections

Applications in a Workspace ONE deployment can connect to enterprise servers by the following routes.

- Device virtual private network (VPN) if the profile is Managed. This includes
  - Operating system built-in VPN, which connects to a generic enterprise gateway appliance.
  - Custom VPN provided by a privileged mobile application, such as the Workspace ONE Tunnel App, which connects to a proprietary proxy endpoint.

  Use of either type of device VPN doesn't require SDK integration.
- **Workspace ONE SDK Tunnel**, whether the profile is Managed or Unmanaged.

  The SDK Tunnel connection is made by the SDK from within the application to a proprietary proxy endpoint.

Both the Workspace ONE Tunnel App and the SDK Tunnel connect via an endpoint in the Unified Access Gateway (UAG). The UAG is deployed at the same enterprise as the UEM.

Electronic certificates are typically used when making a mobile connection to an enterprise server. For example, certificates are used when establishing a Secure Socket Layer or Transport Layer Security (SSL/TLS) connection. An enterprise infrastructure may use a private certification authority (CA) to sign certificates for SSL/TLS. Use of a private CA in the enterprise infrastructure is supported by both the Tunnel App and the SDK Tunnel. In either case, the required client certificates are sent by the UEM during or after enrollment.

Mobile applications that can connect to an enterprise server will then need to be authenticated. The Workspace ONE SDK has integrated authentication, which supports the following.

- Automatic presentation of stored login credentials.
- Automatic response to certificate-based authentication challenges.

Workspace ONE Profile Comparison Table

The following table compares the capabilities of the different UEM profiles in terms of how they meet requirements for mobile access to enterprise data and services. The requirements themselves are discussed in the introduction, in the Requirements section.

Requirement	Workspace ONE Managed Profile Only	Workspace ONE Managed or Unmanaged Profile

Unlock	The enterprise can
	• Require a device-level passcode.
	• Require a passcode for an enterprise partition.
	• Specify complexity rules and allowed types for the passcode.
	• Allow or block biometric unlock of the device.
		The enterprise can
		• Require application unlock by the following:
		  ○ Domain user name and password.
		  ○ Passcode of specified complexity.
		  ○ Default; that is, rely on device unlock.
		• Allow or block biometric unlock of applications.
		The same passcode can be shared by the *application suite.

Inactivity Monitoring	The enterprise can restrict the length of the sleep time-out at the device level.	The enterprise can impose an inactivity timer on applications. The timer is shared by the *application suite.
Encryption of data at rest	The enterprise can force encryption of the whole device, or of the enterprise partition.	Encryption of application data is facilitated by the SDK. See Workspace ONE Encryption of Data at Rest for a description.
Data Management Commands	The enterprise can • Wipe all data from the device, a device wipe. • Uninstall selected applications.	The enterprise can • Remove all credentials and resources obtained as part of enrollment, an *enterprise wipe*. • Remove entitlement to an application, referred to as *unassigning*, which will result in that application being wiped. **Note**: Entitlement to Hub cannot be removed.

Requirement	Workspace ONE Managed Profile Only	Workspace ONE Managed or Unmanaged Profile
Compliance	The enterprise can Force or block installation of specified applications. Force or block operating system updates and security patches. Block access if the device doesn't comply with UEM restrictions on operating system version and security patch level. Block use of the camera, and other device features. Enforce compliance at the device level, or in the enterprise partition.	The enterprise can Facilitate installation of recommended applications. Block enterprise applications if the device doesn't comply with UEM restrictions, such as the following: Operating system version or security patch level. Running on a compromised device. Some profile restrictions are available from an SDK programming interface so that they can be implemented by the application, including: Location services being switched on.
Device Compromise Detection	The enterprise can detect device compromise, based on a dynamic set of conditions. A mobile application has to be present to execute the detection process.	Every SDK application can detect device compromise, based on a dynamic set of conditions.
Offline access limit	The enterprise can track the last seen time of the device. The UEM can implement restrictions and enforcement on the server side if the offline access limit is exceeded.	The enterprise can track the last connection time of each enterprise application. If the offline access limit is exceeded, the SDK locks the application user interface.
Host Access	The enterprise can set up a virtual private network (VPN) at the device level, or in the enterprise partition. The VPN can be configured by • The VPN client built in to the device operating system, connecting to any supported enterprise gateway. • Workspace ONE Tunnel custom VPN application, connecting to Unified Access Gateway at the enterprise. The enterprise can limit which mobile applications have access to the VPN.	SDK applications can use the Workspace ONE SDK Tunnel to connect via the Unified Access Gateway at the enterprise. The enterprise can specify what tunnel connections are allowed.

Requirement	Workspace ONE Managed Profile Only	Workspace ONE Managed or Unmanaged Profile
Authentication of the end user at the host	Enterprise end users can be authenticated by • Login credentials entered on demand. • Proxy authentication by custom VPN. • Certificate-based authentication from the device store.	Enterprise end users can be authenticated by • Login credentials entered on demand and stored for *integrated authentication*. • Proxy authentication by tunnel endpoint in UAG. • Certificate-based authentication for each application.
Electronic Certificate Storage	The enterprise can manage the device store, and can utilize electronic certificates signed by a private authority.	The enterprise can • Manage certificates sent to each application by the UEM, including electronic certificates signed by a private authority. • Share certificates between the Workspace ONE Productivity Apps for iOS.
Data Leakage Prevention	The enterprise can • Set up an enterprise partition into which the Workspace ONE Productivity Apps, Customer and Partner Apps, and non-integrated apps, are all installed. • Block device feature usage.	The SDK can • Encrypt data during inter-application communication. • Encrypt data in cut-copy- paste interactions. • Block screen capture, for Android. • Provide a list from the UEM of the applications that are allowed to communicate with enterprise applications.
Usability	Enrollment is required. A high level of privilege must be granted to the enterprise, which may be seen as intrusive by end users. Devices might issue repeated warnings about certificate installation, network traffic interception, and other intrusions. UEM Administrators with more control over end-user devices therefore have more responsibility. After enrollment and granting of privileges, the enterprise may reduce or waive the need for re- authentication.	Enrollment is required. A lower level of privilege, or none, need be granted to enterprise applications. Administrators with less control therefore have less responsibility. Authentication sessions and the inactivity timer can be shared between applications.

Requirement	Workspace ONE Managed Profile Only	Workspace ONE Managed or Unmanaged Profile
Application Configuration	AppConfig support • AppConfig standard user interface in the UEM. • AppConfig values available to mobile applications through standard programming interface.	AppConfig compatibility • AppConfig standard user interface in the UEM. • AppConfig compatible values available through the SDK programming interface, for Android. Workspace ONE Custom Settings • User interface in the UEM. • Values available through the SDK programming interface.

Table 2: Comparison of Enterprise Mobility Management Capabilities Between Different UEM Profiles

Notes on the profile comparison table:

* *Application Suite.*

In a Workspace ONE solution, the scope of application suite sharing is as follows:

All enterprise applications, for Android.

Workspace ONE Productivity Apps, for iOS.

Workspace ONE Component Diagram

The following diagram shows possible components of a Workspace ONE solution and illustrates some differences between Managed and Unmanaged profiles.

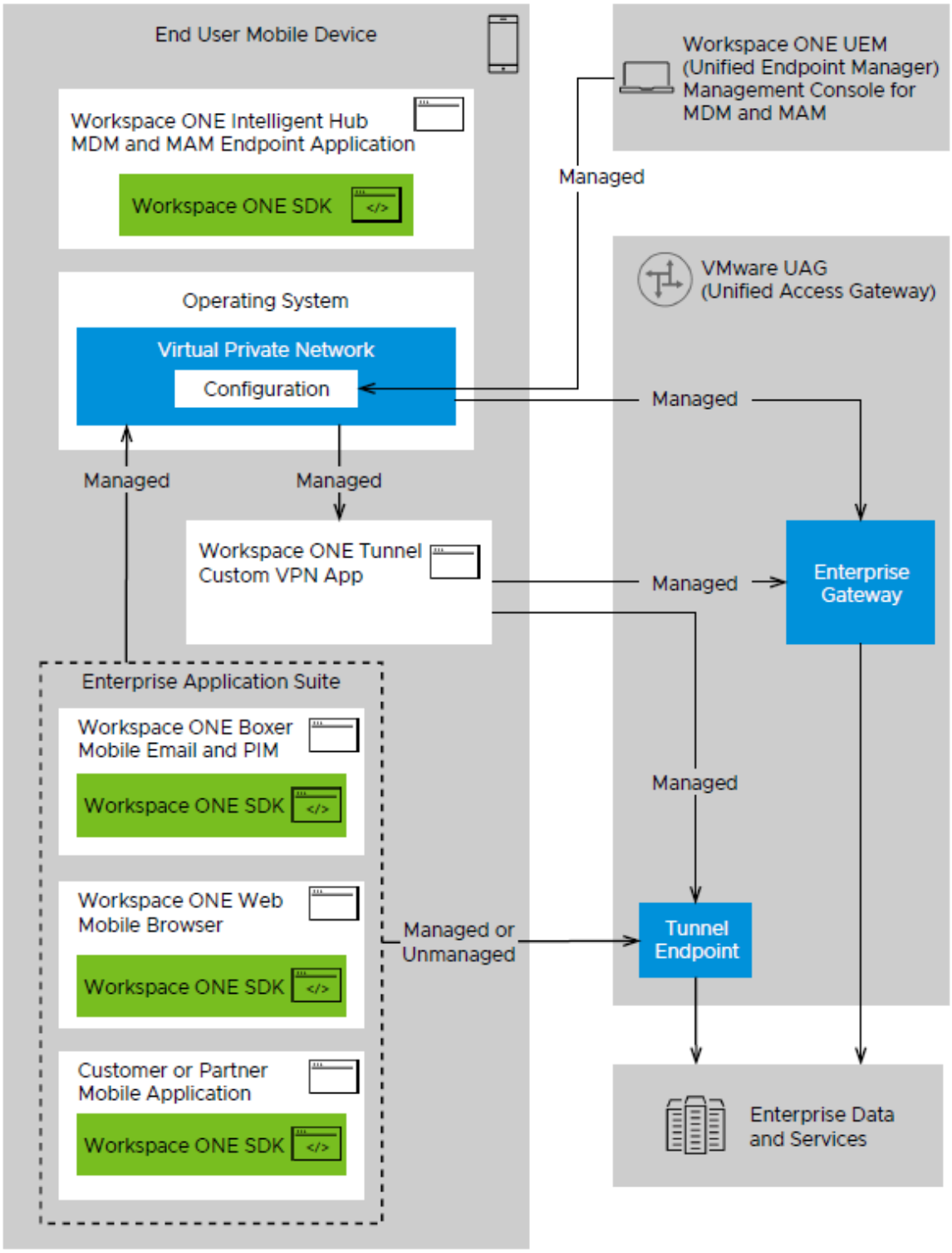

Figure 3: Components of a Workspace ONE Solution and Differences Between Managed and Unmanaged Profiles

Use Cases

The following use cases describe some typical solutions based on the Workspace ONE platform.

SDK Only Without MDM (End-User Device)

For some organizations with Bring Your Own Device (BYOD) programs, requiring end users to install device management may be against corporate policy. In these cases, organizations must rely completely on the Workspace ONE SDK to provide a management layer around corporate applications and resources. Corporate applications embed the SDK so that management controls and core enterprise capabilities become available. Following is a list of common use cases and how they are achieved in such a scenario:

- Enterprise wipe – The Workspace ONE SDK wipes the data inside SDK integrated apps.
- Passcode – The Workspace ONE SDK shows an in-app passcode which is also used to encrypt data in the app.
- Tunneling – Handled inside each application via the SDK tunnel.
- Device encryption – Handled by the SDK with app-level encryption.
- Single sign-on – SDK is used to perform certificate, basic, or NTLM through the integrated authentication feature.
- Compliance – Available through SDK compliance engine.
- Offline compliance – Enforced through the SDK compliance engine; the SDK stores an offline version of the compliance rules and actions and enforces them even if there is no connectivity to the admin console.

MDM Supplemented with SDK (End-User Device)

Organizations either with or without BYOD often choose to integrate the Workspace ONE SDK into their in-house mobile applications, in addition to enrolling devices into device management. In such cases, SDK capabilities like SDK passcode and encryption, which are already being handled by MDM, are disabled, and a subset of SDK functionality such as compromised protection or offline compliance policies that are not available out of the box through pure MDM are enabled. This hybrid approach to combine SDK with MDM facilitates the Workspace ONE adaptive management paradigm, enabling users to first enroll without MDM, and then eventually step up into MDM management at a later time in order to access additional apps and resources that are available only when the device is MDM-managed.

Following are some examples of common use cases in this situation and how they are addressed:

- Enterprise wipe – Handled via MDM; apps are un-installed, and device configuration profiles are removed.
- Passcode – Handled via device passcode, and complexity is defined through MDM policy.
- Tunneling – Handled via MDM per-app VPN.
- Device encryption – Handled via device encryption enforced by MDM.
- Single sign-on – Depending on the authentication type required, the device OS may not support SSO natively out of the box through MDM. In these cases, the SDK is often utilized to provide certificate authentication or SSO to basic / NTLM-authenticated web services.
- Compliance – Available through Workspace ONE UEM compliance engine.
- Offline compliance – Enforced through the SDK compliance engine, the SDK stores an offline version of the compliance rules and actions and enforces them even if there is no connectivity to the admin console.

MDM Supplemented with SDK (Shared Device / Line of Business)

For organizations deploying corporate-owned shared devices, the Intelligent Hub's check- in / check-out feature is utilized to manage which resources must be installed and removed from the device. A common example of this is certain apps are removed, installed, or hidden based on the current checked-out user. Many organizations opt to use the hide-app feature, rather than remove and re-install apps during every shift change, which can be taxing on the network infrastructure and slow down employee productivity. In these scenarios, the Workspace ONE SDK is integrated into apps on the device to ensure the app is aware of the user change, and the correct data is shown to each user, along with providing single sign-on to the SDK-integrated apps.

Workspace ONE Encryption of Data at Rest

The Workspace ONE mobile SDK encrypts data at rest on the device.

Terminology

A number of cryptographic terms are used to describe how the Workspace ONE mobile SDK protects data at rest. Introductory definitions are given here. Except where noted, these are common terms and full definitions may be found elsewhere, in general online and published literature.

Terminology	Definitions
Symmetric and Asymmetric Encryption	Symmetric encryption utilizes the same key to encrypt and decrypt data. Asymmetric encryption utilizes two dependent keys, referred to here as a key pair. Public Key Cryptography is a type of asymmetric cryptography. There are a number of standard ciphers for symmetric and asymmetric encryption.
Cryptographic Hashing	Applying a cryptographic hash process to an input value gives an output result that is predictable and repeatable, but from which the input cannot be recovered. It is a one- way process. There are a number of standard hashing functions. Cryptographic hashing can be used, for example, to check that a supplied passcode matches a passcode that was set earlier. The passcode value itself might be too sensitive to store, but a hash of the value may be safe to store because the passcode cannot be recovered from it. The check is then made by generating a hash of the supplied passcode. If the hash of the supplied value matches the stored hash, then the check passes. A hashing function can also impose a proof-of-work type of delay in order to stymie a brute force attack on a passcode or other secret.

Terminology	Definitions
Salt Values	A cryptographic hash process on its own may leave its input vulnerable to what is known as a *rainbow table* attack. Rainbow tables are pre-generated lists of common input values, for example, passwords, and their output hashes. The table can help an attacker bypass the proof-of-work defense in cryptographic hashing. To defend against this type of attack, an extra input value is appended to the secret input prior to the hash process. This extra input is referred to as a salt value.
Key Derivation	An encryption key can be generated from other data, such as a passcode, by using a key derivation (KD) function. There are a number of standard KD functions, some based on repeated cryptographic hashing. A KD function can also impose a proof-of-work type of delay in order to stymie a brute force attack on a passcode.
Secure Random Number Generation	The operating systems in scope of this product each provide a secure random number generator (RNG) that is suitable for cryptography.
Key Hierarchy	In principle, only a single encryption key is necessary to protect a set of application data. In practice however, a single key doesn't support some common requirements, such as the following: • Secure recovery from a forgotten passcode. • Secure shared passcode. • Changing passcode without re-encrypting the whole set of data. • Use of biometric authentication. Workspace ONE mobile SDK protection of data at rest therefore makes use of a system of multiple keys and other cryptographic resources. This type of approach is referred to as a key hierarchy.
Run-Time Memory	This term is used here to mean a location that holds values only until the application terminates. Values in run-time memory aren't stored persistently. Keys and other cryptographic resources that are in run-time memory may be protected further, for example by obfuscation.

Terminology	Definitions
Application Storage	This term is used here to mean a persistent storage location that is only accessible to one application. For example, the default data area for an Android or iOS application is accessible only to that application.
Shared Storage	This term is used here to mean a persistent storage location that is accessible to multiple applications. For example, a shared iOS keychain group is accessible to multiple applications.

Passcode-Based Encryption

The Workspace ONE mobile SDK implements passcode-based encryption. The cryptographic processing is as follows.

Notes on the Descriptions

- These descriptions are simplified in some areas, for brevity. Full and authoritative descriptions are available on request.
- Unless otherwise stated, key means a key for symmetric encryption.
- Unless otherwise stated, random numbers are 32 bytes, that is, 256 bits.
- Symmetric encryption uses an AES Key Wrap cipher that conforms to RFC 3394.
- Asymmetric encryption uses an RSA cipher with PKCS1 padding.
- Key derivation functions are platform-specific, as follows.
    - PBKDF2 HMAC with SHA256 digest and 20,000 iterations for Android.
    - PBKDF2 HMAC with SHA256 digest and 10,000 iterations for iOS.
- Shared storage implementation depends on the mobile platform, as follows:
    - For Android, shared storage is implemented by a Content Provider in the SDK.
    - For iOS, shared storage is implemented by a shared keychain access group. An iOS keychain access group can only include applications from the same application vendor, that is, signed by the same developer team.
- Hardware security module (HSM) implementation depends on the mobile platform, as follows:
    - For Android, HSM is the Android Keystore.
    - For iOS, HSM is the Secure Enclave Processor.

HSM here means a component that can execute cryptographic operations using keys that it stores without giving access to the key values themselves.

Setting Up Passcode-Based Encryption

To set up passcode-based encryption, the following steps occur.

1. The user sets an application unlock password, PIN, or other credential, referred to here as the *Passcode*.
2. A random value is generated, the *Passcode Salt*.
3. The Passcode and Passcode Salt are passed through a key derivation function to generate a key, the Passcode Key.
4. A random key is generated, the *Key Encryption Key (KEK)*.
5. A copy of the KEK encrypted by the Passcode Key is placed in the shared storage.
6. A random key, the *Data Key*, is generated.
7. Depending on the device operating system, a *Master Key* is generated:
    - If Android, the KEK and a new random salt value are passed through a key derivation function to generate the Master Key. The salt value, referred to as the Application Salt, is placed in application storage.
    - If iOS, the Master Key is the same as the KEK.
8. A copy of the Data Key encrypted by the Master Key is placed in application storage.

The Data Key is then used to encrypt the general application data at rest on the device. Note that unencrypted key values aren't stored persistently.

Starting Access to Encrypted Data

To access data encrypted by the Data Key, the following steps occur.

1. The user provides the unlock credential, referred to here as the Passcode.
2. Passcode Salt is read from shared storage.
3. The Passcode and Passcode Salt are passed through a key derivation function to generate a key, the Passcode Key.
4. The KEK encrypted by the Passcode Key is read from shared storage and decrypted.
5. Depending on the device operating system, the Master Key is recovered:
    - If Android, the Application Salt read from application storage and the KEK are passed through a key derivation function to re-generate the Master Key.
    - If iOS, the Master Key is the same as the KEK.
6. The Data Key encrypted by the Master Key is read from application storage and decrypted.

The Data Key is then used to decrypt the general application data at rest on the device.

Changing Passcode

To change the passcode credential value, the following steps occur.

1. All the steps necessary for Starting Access to Encrypted Data occur. This means that the KEK is available.
2. The user supplies the new Passcode value.
3. A new random Passcode Salt is generated, if iOS.
4. The new Passcode and Passcode Salt are passed through a key derivation function to generate a new Passcode Key.
5. A copy of the KEK encrypted by the new Passcode Key is placed in the shared storage. The KEK encrypted by the old Passcode Key is discarded.

The new passcode must now be used for future access to encrypted data.

Setting Up Recovery from Passcode Loss for Android

After the KEK is first generated, see Setting up Passcode-Based Encryption, the following steps occur.

1. A random value, the Escrow Key, is generated.
2. A copy of the KEK encrypted by the Escrow Key is placed in the application storage.
3. The Escrow Key is sent to the management console, which stores it.

In case the user forgets their passcode, they must be authenticated in some other way, and then be given access to the Escrow Key. That key can then be used to recover the KEK, and hence the Data Key. After recovery, the user is required to set a new passcode by following the steps under Changing Passcode.

Setting Up Recovery from Passcode Loss for iOS

After the KEK is first generated, see Setting up Passcode-Based Encryption, the following steps occur.

1. A random value, the Escrow Key, is generated in the hardware security module (HSM).
2. A copy of the KEK is encrypted by the Escrow Key in the HSM.
3. The encrypted KEK is sent to the management console, which stores it.

In case the user forgets their passcode, they must be authenticated in some other way, and then be given access to the encrypted KEK. The KEK can be decrypted in the HSM, and then be used to recover the Data Key. After recovery, the user would be required to set a new passcode by following the steps under Changing Passcode.

Passcode-Based Encryption Diagrams

The following diagrams show how the keys and other resources involved in Workspace ONE passcode-based encryption are stored and protected.

Passcode-Based Encryption Class Diagram

The following diagram represents the encryption processes as a Unified Modeling Language (UML) class diagram.

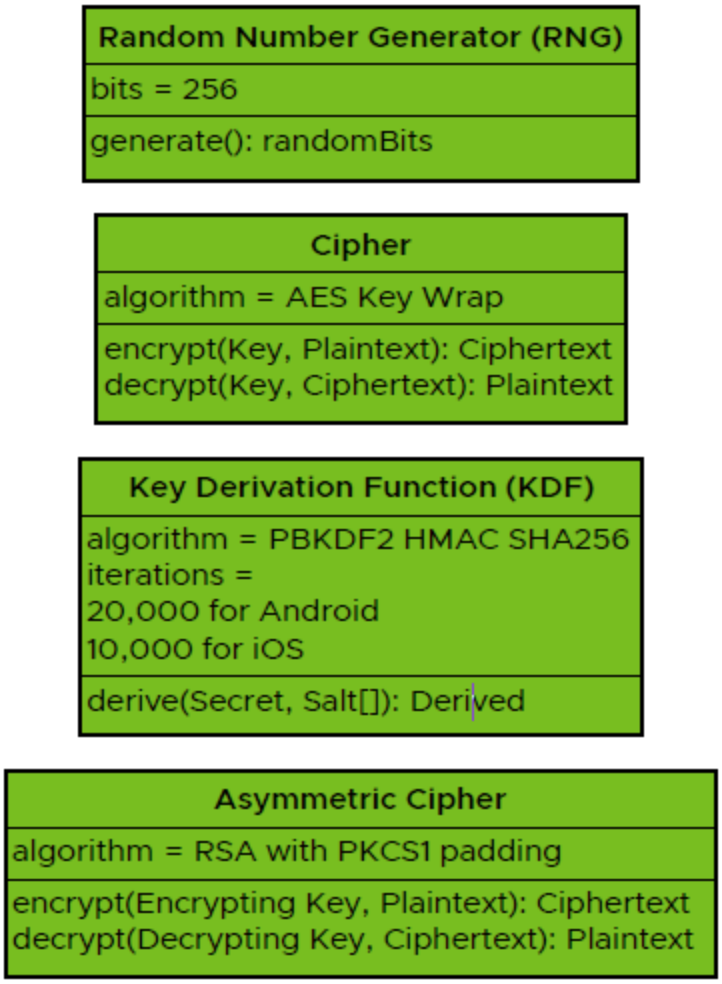

Figure 4: Workspace ONE Passcode-Based Encryption Class Diagram

Passcode-Based Encryption Deployment Diagram

The following diagram represents the storage and protection of the base cryptographic resources involved in Workspace ONE passcode-based encryption as a UML deployment diagram.

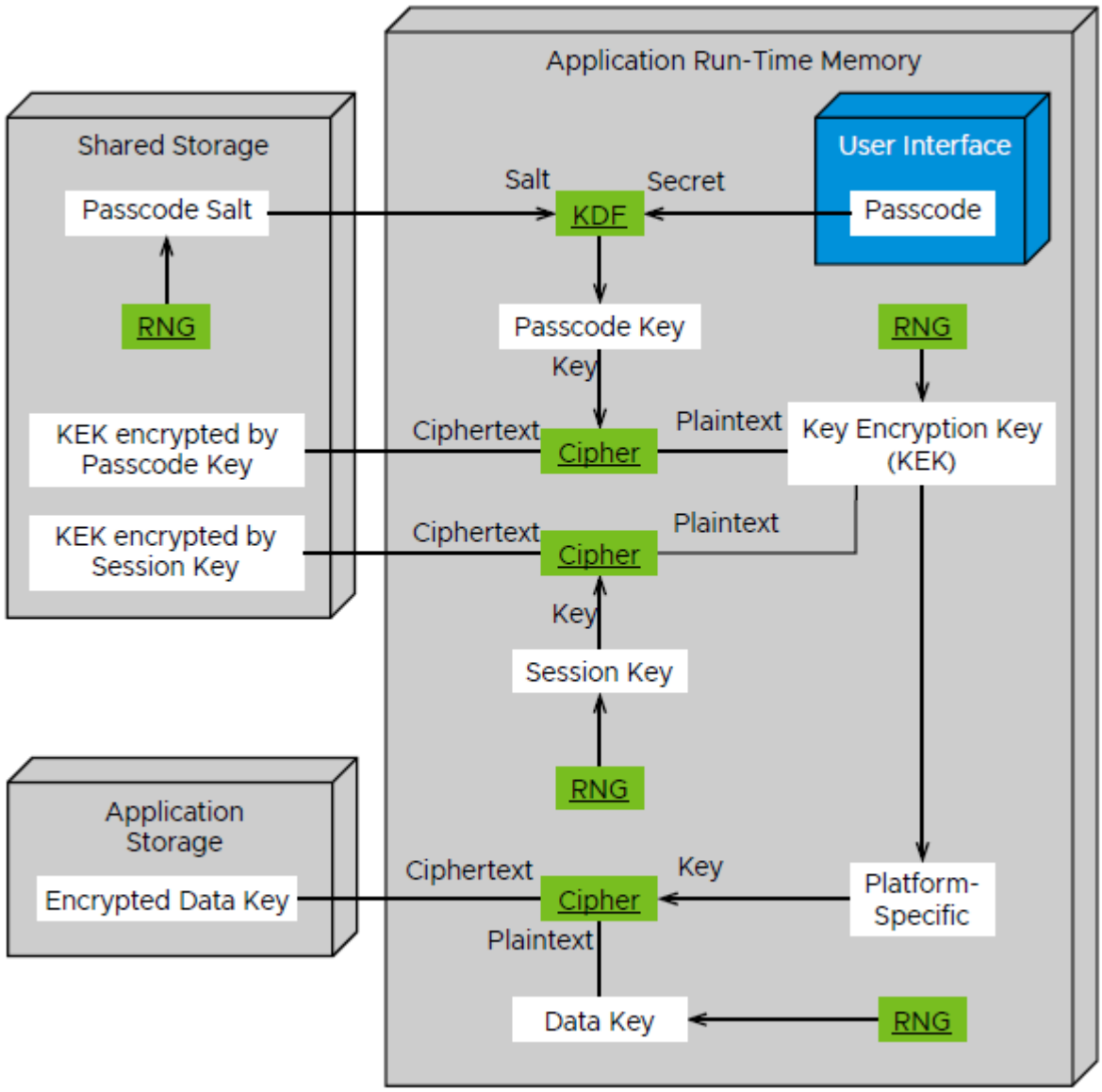

Figure 5: Workspace ONE Passcode-Based Encryption Deployment Diagram

Escrow for Android Deployment Diagram

The following diagram represents the storage and protection of the cryptographic resources involved in Workspace ONE key escrow for Android as a UML deployment diagram.

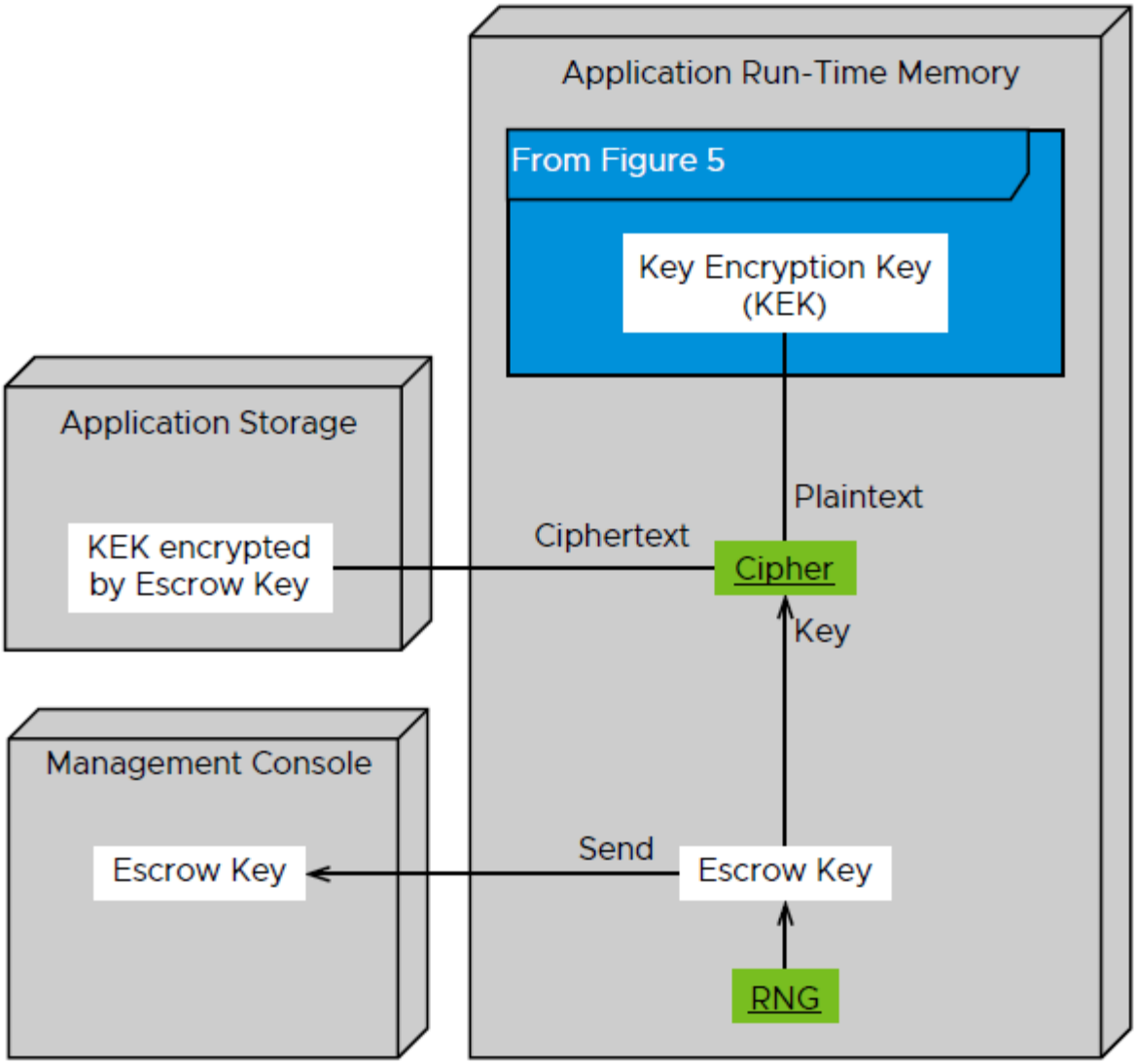

Figure 6: Workspace ONE Escrow for Android Deployment Diagram

Escrow for iOS Deployment Diagram

The following diagram represents the storage and protection of the cryptographic resources involved in Workspace ONE key escrow for iOS as a UML deployment diagram.

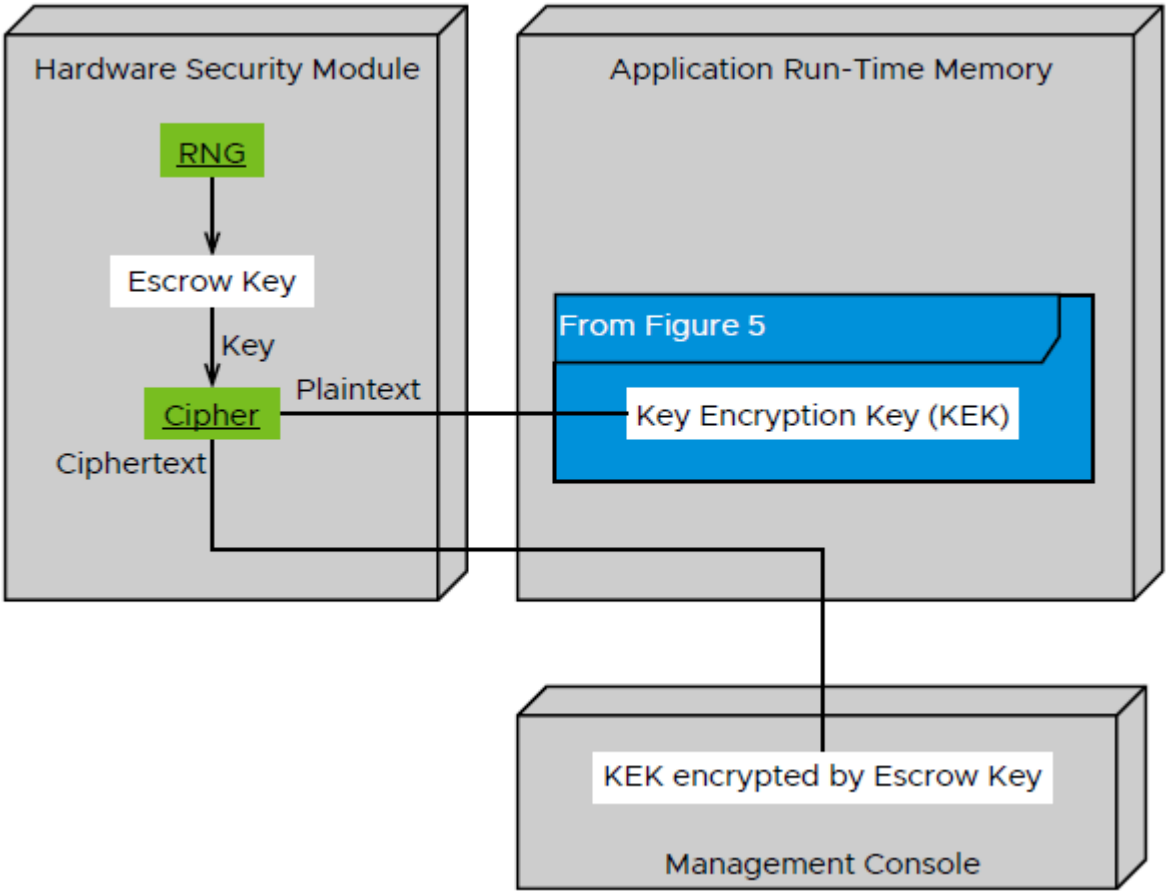

Figure 7: Workspace ONE Escrow for iOS Deployment Diagram

Passcode Sharing

The Workspace ONE mobile SDK implements sharing of the credential for passcode-based encryption. The cryptographic processing for sharing depends on the device operating system.

Setting Up Sharing for Android

After the KEK is first generated, see Setting up Passcode-Based Encryption, the following steps occur to set up passcode sharing for Android.

1. A random value, the Session Key, is generated.
2. A session expiry time is generated as an offset from generation time into the future.
3. The session expiry time, and a copy of the KEK encrypted by the Session Key, are placed in the shared storage.
4. The Session Key and expiry time is made available to other applications by a Content Provider in the SDK.
5. In addition, a random value, the Ephemeral Key is generated in the hardware security module (HSM). A copy of the Session Key encrypted by the Ephemeral Key is placed in the application storage.

Notes:

- The Ephemeral Key enables quick recovery of the Session Key in case the application crashes or is unloaded from memory by the operating system.
- The Ephemeral Key is deleted when the SDK determines that the device has been power cycled since the application last launched.

Setting Up sharing for iOS

After the KEK is first generated, see Setting up Passcode-Based Encryption, the following steps occur to set up passcode sharing for iOS.

1. A random value, the Session Key, is generated.
2. A session expiry time is generated as an offset from generation time into the future.
3. The session expiry time, and a copy of the KEK encrypted by the Session Key, are placed in the shared storage.
4. A random key pair for asymmetric encryption, the *Application Public Key* and *Application Private Key*, is generated.
5. A copy of each key in the pair encrypted by the KEK is placed in the application storage.
6. A copy of the Application Public Key (AUK) is placed in the shared storage.
7. If any other applications have already placed their own AUK values in the shared storage, then this application places in shared storage a copy of the Session Key encrypted by each other AUK value.

Notes:

When an iOS application starts, it won't have its Application Private Key in memory. It must first obtain the KEK, for example by following the steps under Starting Access to Encrypted Data.

The multiple encryptions of the Session Key enable other applications that still have their Application Private Key in memory to decrypt one of the encrypted Session Key values in shared storage. From the Session Key, the other

application can recover the KEK, and hence its Data Key.In case the end user selects a manual lock option, or if the session expiry time is passed, the SDK in the current application removes the KEK encrypted by the Session Key from shared storage. This forces re-authentication next time, for example by following the steps under Starting Access to Encrypted Data.

Session for Android Deployment Diagram

The following diagram represents the storage and protection of the cryptographic resources involved in Workspace ONE passcode sharing for Android as a UML deployment diagram.

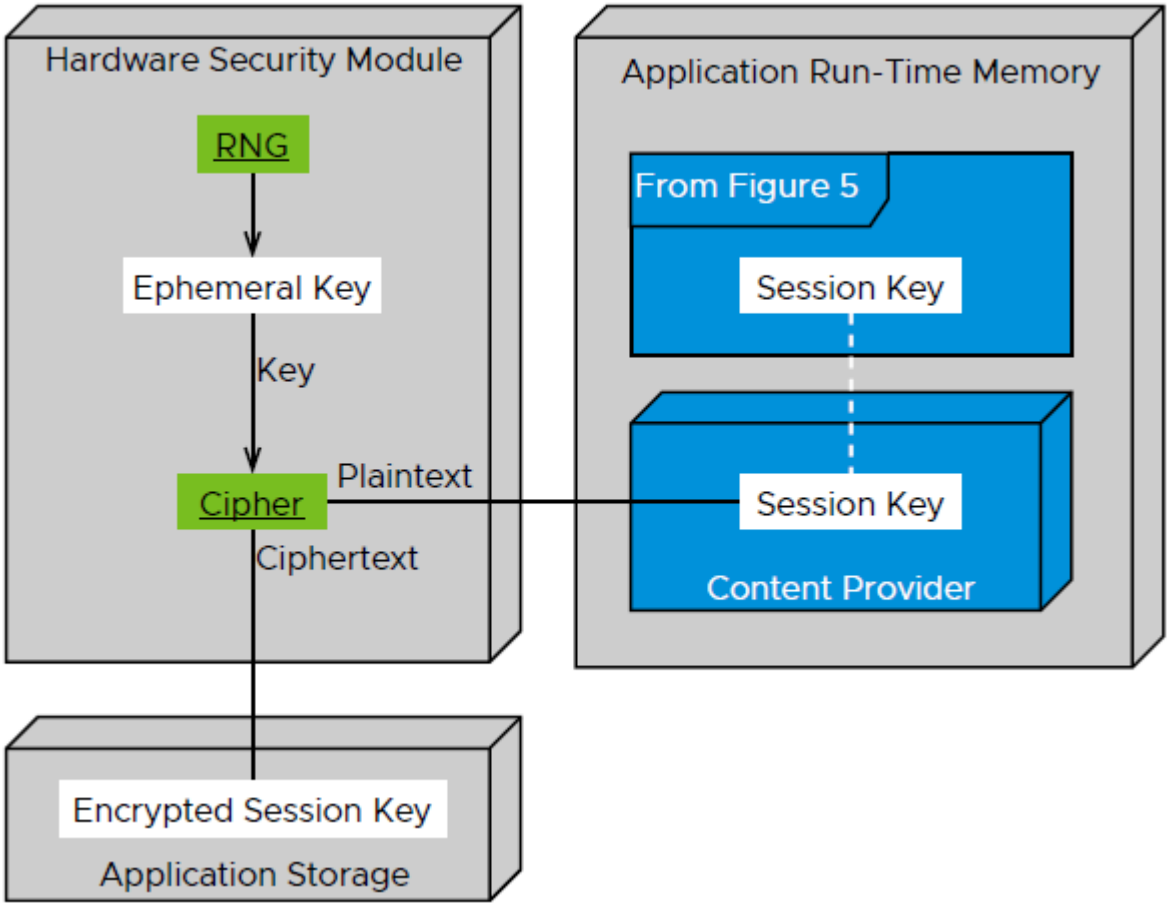

Figure 8: Workspace ONE Session for Android Deployment Diagram

Session for iOS Deployment Diagram

The following diagram represents the storage and protection of the cryptographic resources involved in Workspace ONE passcode sharing for iOS as a UML deployment diagram.

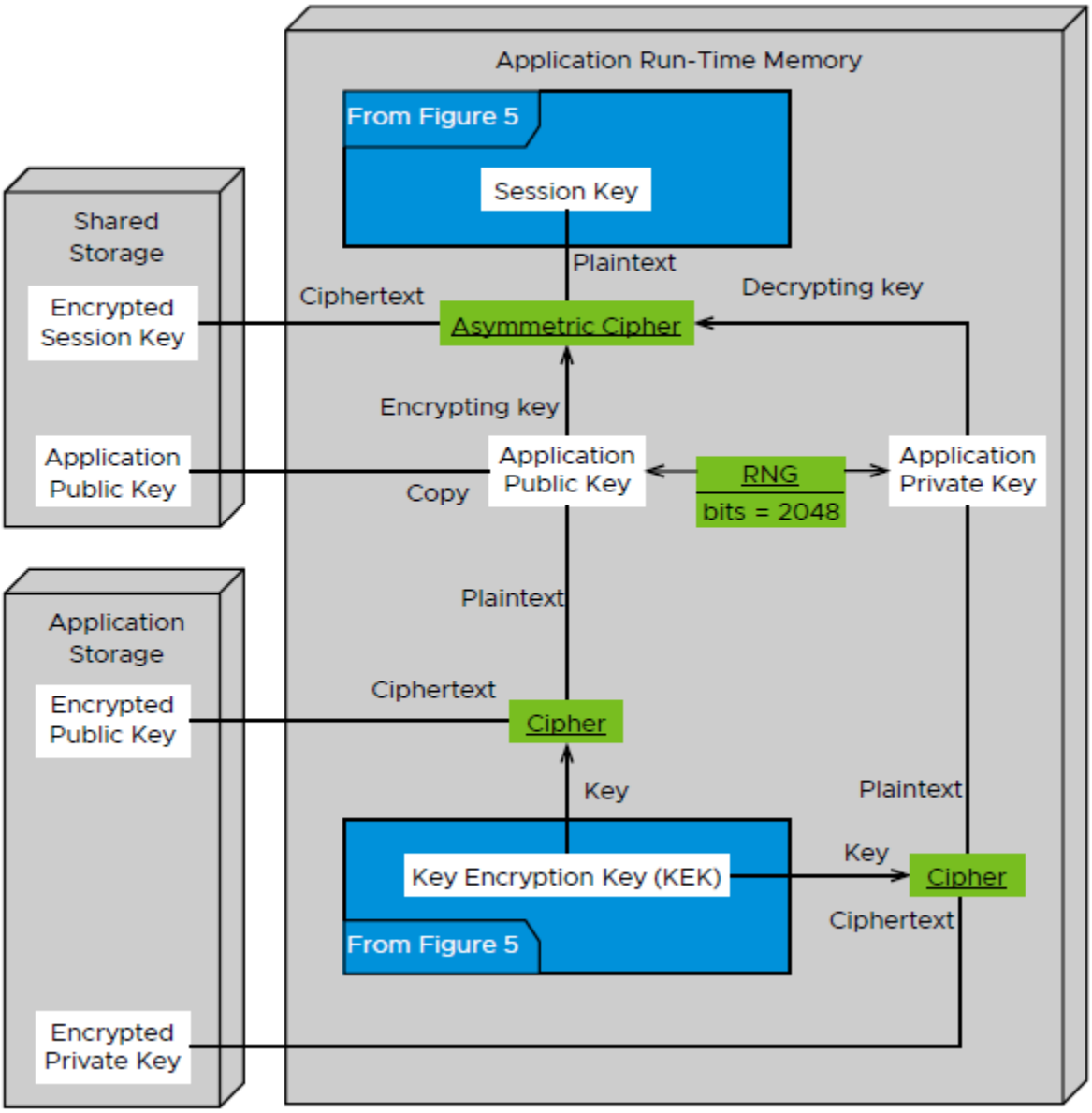

Figure 9: Workspace ONE Session for iOS Deployment Diagram

Conclusion

The work of the modern enterprise relies upon mobile access to data and services. Mobilizing enterprise data requires

- Protected data at rest, in applications on mobile devices.
- Protected data in transit between mobile applications and host servers.
- Protected data in motion, between applications on the same device.
- Usability, by end users and by system administrators.

Enterprises can choose one or more of the following:

- Solutions that integrate Mobile Device Management (MDM).
- Solutions that integrate Mobile Application Management (MAM).
- Non-integrated solutions.

Each type of solution meets enterprise requirements for mobility to a greater or lesser extent, and in a different way.

Omnissa Workspace ONE is a suitable platform for all these types of solutions.

Document Information

Revision History

Date	Description
17th March 2025	Rebranding changes