

Zebra Developers

Build Your Edge

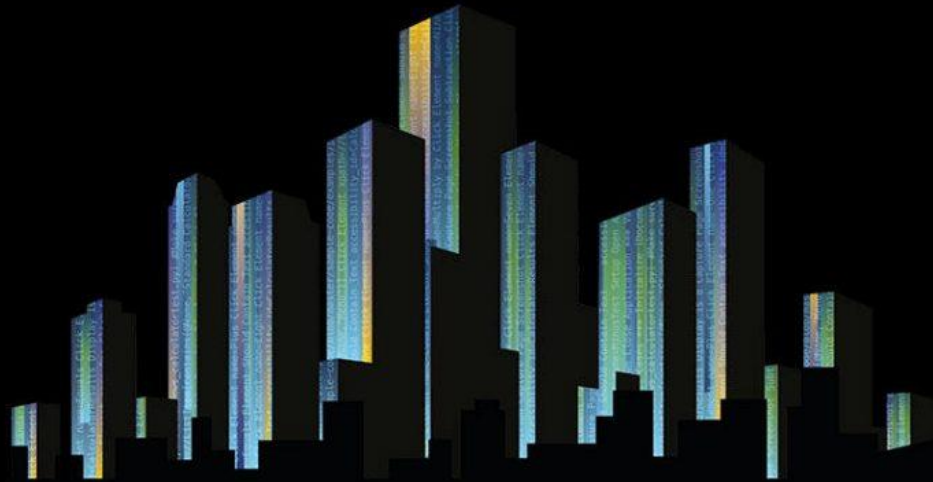




EMM (MDM) and Application Management



Zebra
DevCon 2021
Connect | Learn | Build



Darryn Campbell

SW Architect, Zebra Technologies
@darryncampbell

November 3rd – 5th, 2021

EMM (MDM) and Application Management

Agenda

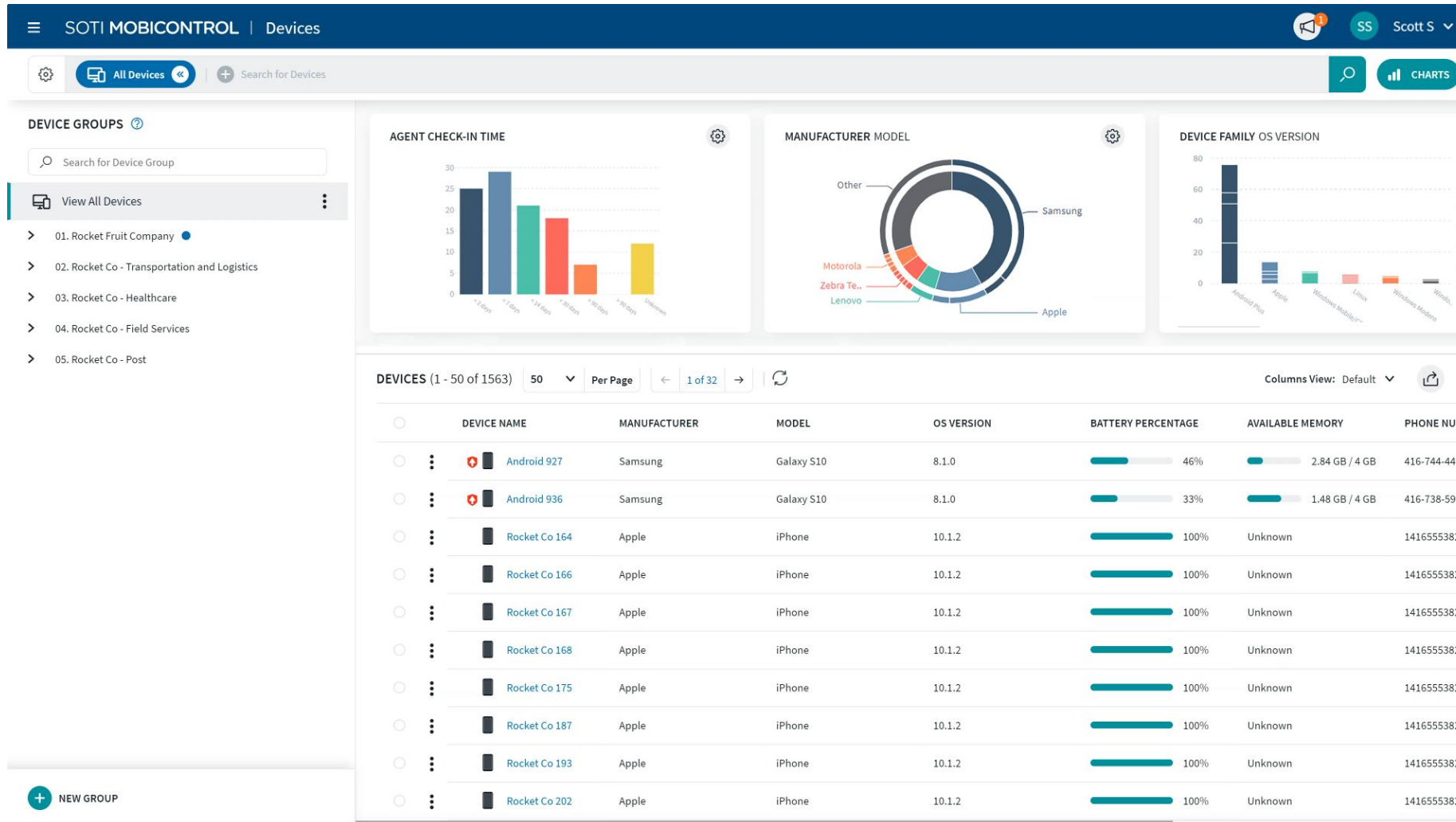
- Overview of EMM landscape
 - Terminology & overall setup
 - EMM options
- Developer considerations when your app is controlled by an EMM
 - Configuration
 - OEMConfig
 - How to share a file with a specific app with Scoped Storage Manager

EMM (MDM) and Application Management

Overview of EMM landscape

EMM (MDM) and Application Management

Terminology & Overall setup



EMM (MDM) and Application Management

Terminology & Overall setup

- EMM: Enterprise Mobility Management | MDM: Mobile Device management
- MAM: Mobile Application Management
- UEM: Unified Endpoint management: EMM + MAM
- Work Profile: Designed for using personal devices in enterprise. Formally BYOD or BYOP.
- Fully Managed device: Category of devices which are company-owned and company-managed. Typically smartphone form factor. More restrictive than the Work profile from a user point of view. Formally COPE.
- Dedicated devices: Category which defines Zebra devices. Google define this as devices which fulfil a single use case but that can be misleading – more accurately they are multi-purpose devices which can excel at enabling employees to perform a single task. Formally COSU.
- See also Google's defined feature list and solution sets:
<https://developers.google.com/android/work/requirements>

EMM (MDM) and Application Management

EMM Options - Demo



- Android Enterprise Recommended → Find Solutions
 - https://www.android.com/intl/en_uk/enterprise/solutions-finder/

- Designed for both dedicated devices and BYOD devices
- Buckets 1, 2 and 3 are Google products (this website is continually updating)
- Majority of Zebra customers will adopt an EMM from the fourth bucket, "AER EMM Provider"

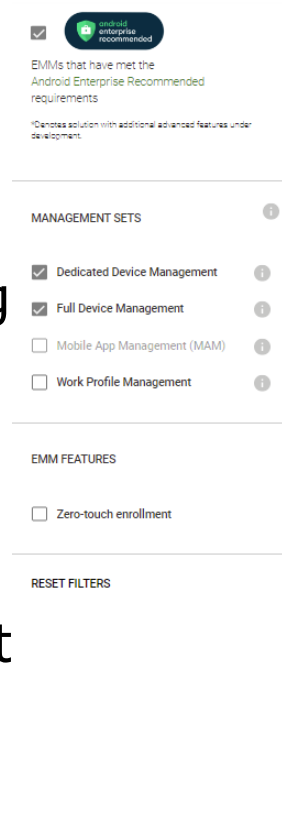
See next steps below by selecting a column.	Google Fundamental Endpoint Management	Android Enterprise Essentials	Google Advanced Endpoint Management	Android Enterprise Recommended EMM Provider
	Recommended			
• View device listing ⓘ	✓	✓	✓	✓
• Enforce PIN or passcode ⓘ	✓	✓	✓	✓
• Wipe data remotely ⓘ	✓	✓	✓	✓
• Distribute apps ⓘ	✓		✓	✓
Remote deployment and inventory management ⓘ		✓	✓	✓
Manage lost devices ⓘ			✓	✓
Separate work and personal data ⓘ			✓	✓
Configure settings and networks ⓘ			✓	✓

EMM (MDM) and Application Management

EMM Options - Demo



- Android Enterprise Recommended EMM Provider → Browse Providers
 - <https://androidenterprisepartners.withgoogle.com/emm/>
- Also shows EMMs not part of AER (a Google certification program)
- Some filters exist but recommend additional research before selecting an EMM, e.g. AOSP support is not a filterable option.
- How the EMM agent is deployed and the *potential* capabilities of that agent will be largely consistent across EMMs.



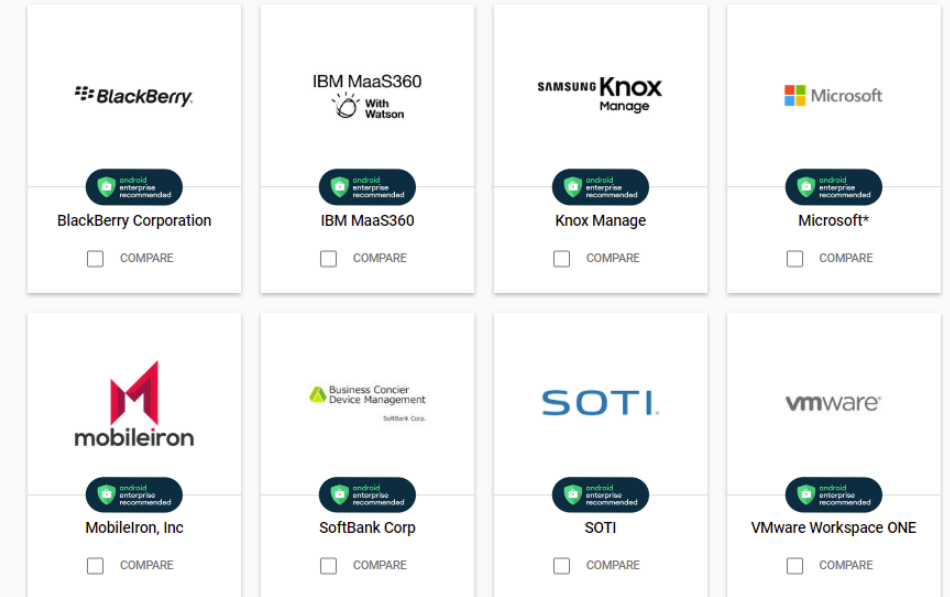
EMMs

Validated mobility management solutions that support Android Enterprise advanced and standard features.

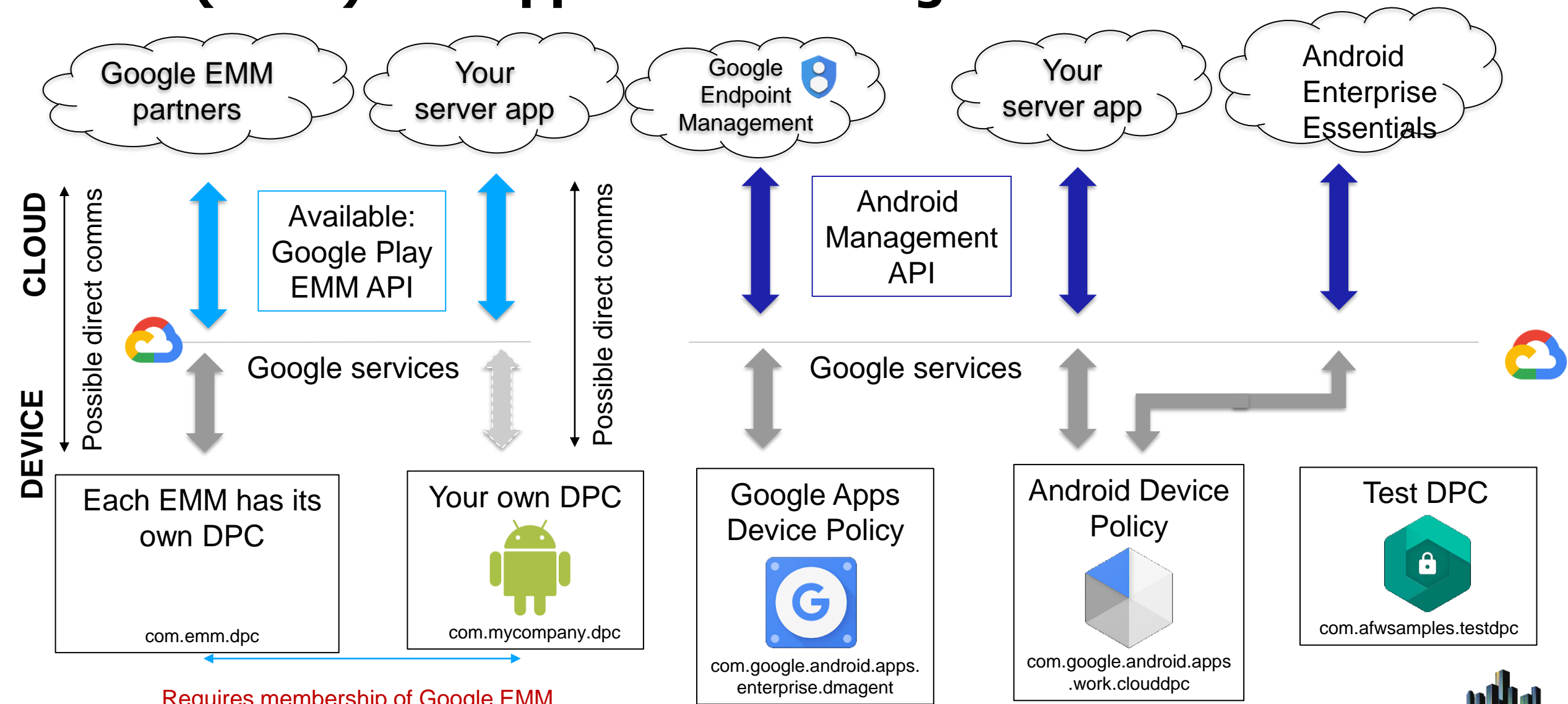
Search EMMs

FILTERING 8 OF 8

SORT BY RELEVANCY



EMM (MDM) and Application Management



Requires membership of Google EMM community. Not open to new members.

EMM (MDM) and Application Management

Developer considerations when your app
is controlled by an EMM

EMM (MDM) and Application Management Configuration

- Consider **how** you configure your app today:
 - How do you expose application configuration?
 - Would you still want the app to be configured under an EMM?
- Consider **what** you configure in your app today:
 - Would the configuration be different under EMM?
 - Are there any features of your app you might want to disable when running under EMM?
 - Does my app collect data that an Enterprise device administrator might consider proprietary?

EMM (MDM) and Application Management

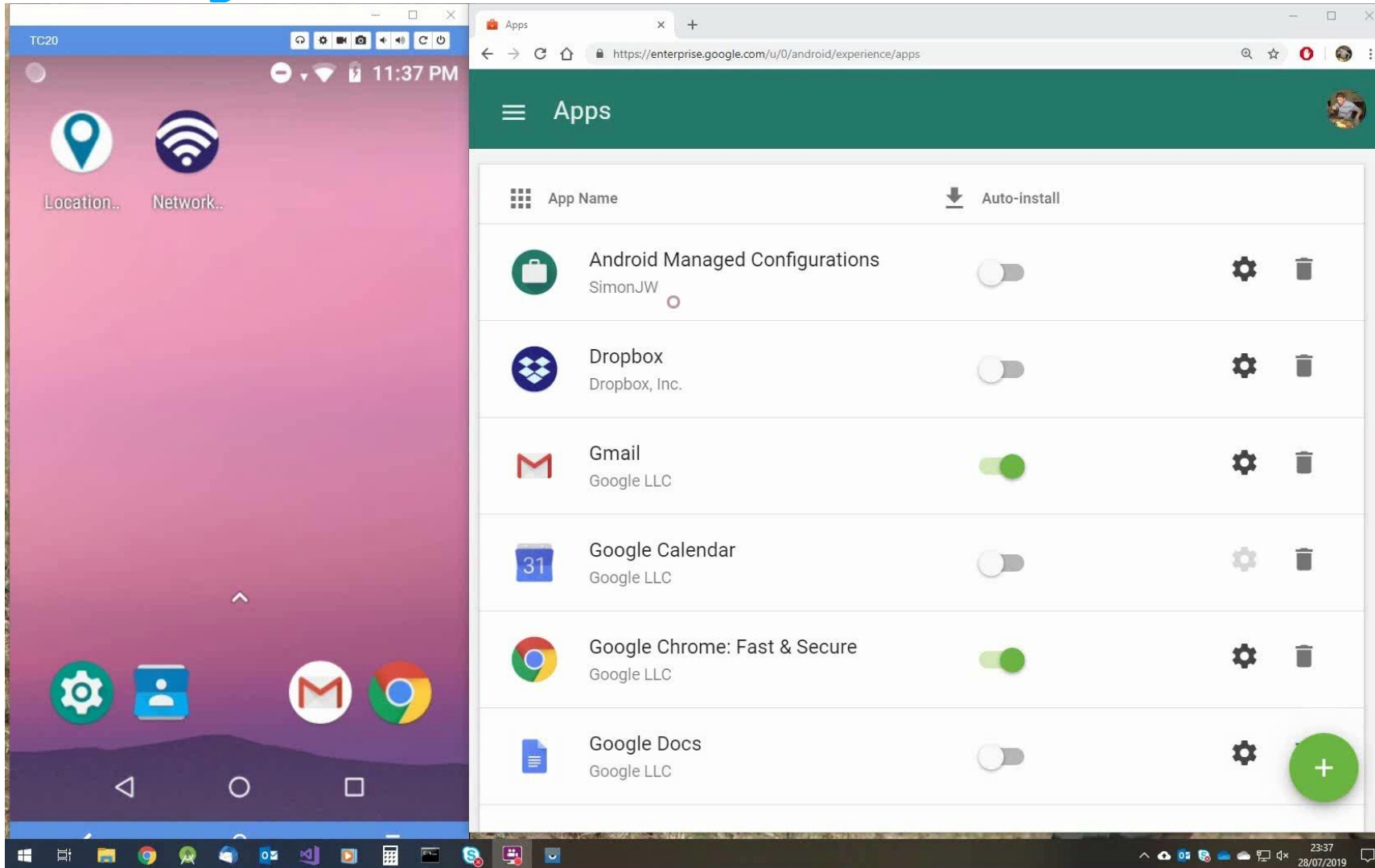
Managed Configurations

1. Define your application configurations (application restrictions) in XML
2. Supported types: Bool, String, Integer, Choice, Bundle
3. Check state of managed configs:
 - OnResume()
 - Listen for broadcast
4. Configure the application as appropriate

```
<resources>
  <!-- Bool restriction -->
  <string name="title_can_say_hello">Can say hello</string>
  <string name="desc_can_say_hello">Whether app can say Hello</string>
  <bool name="default_can_say_hello">>false</bool>
</resources>
```

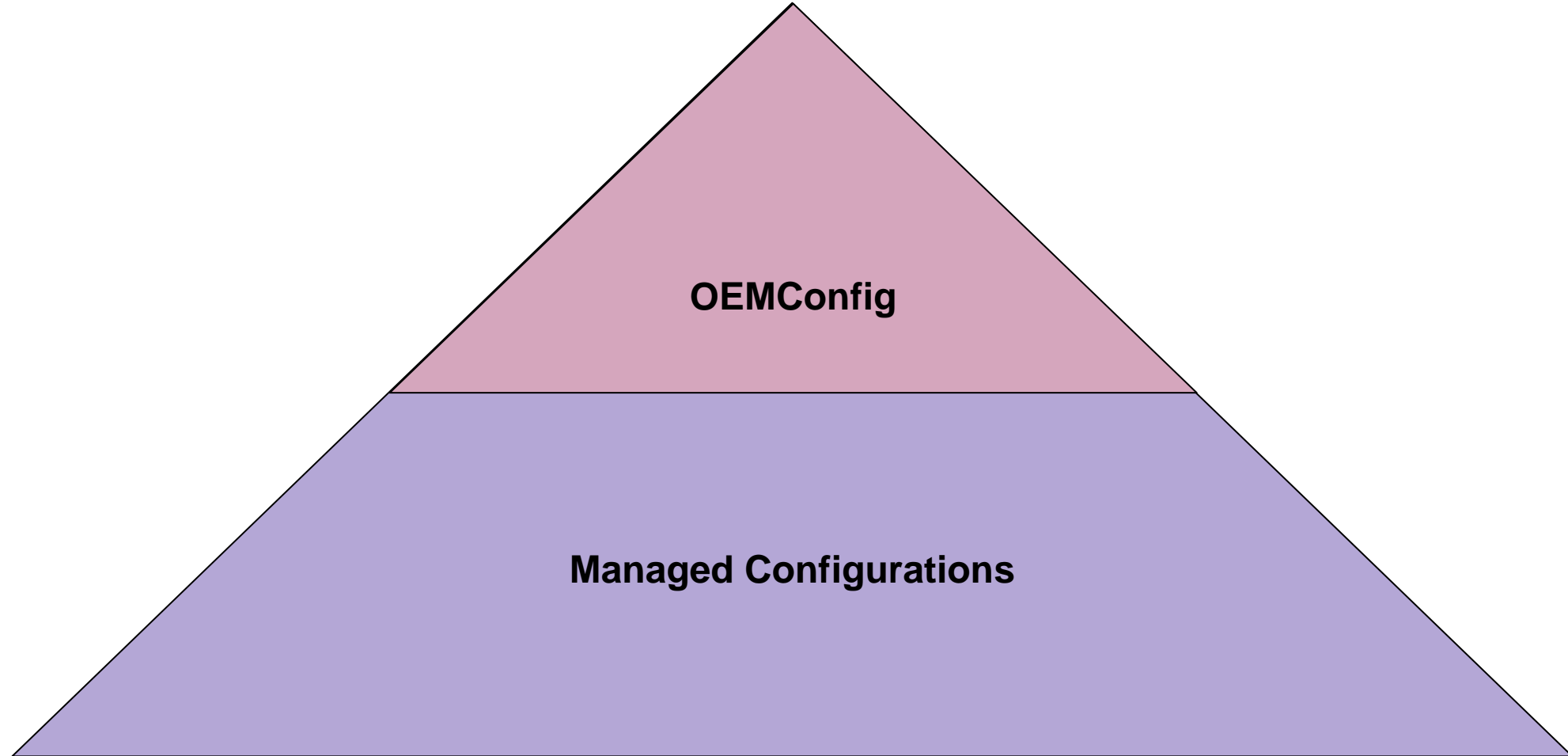
EMM (MDM) and Application Management

Managed Configurations



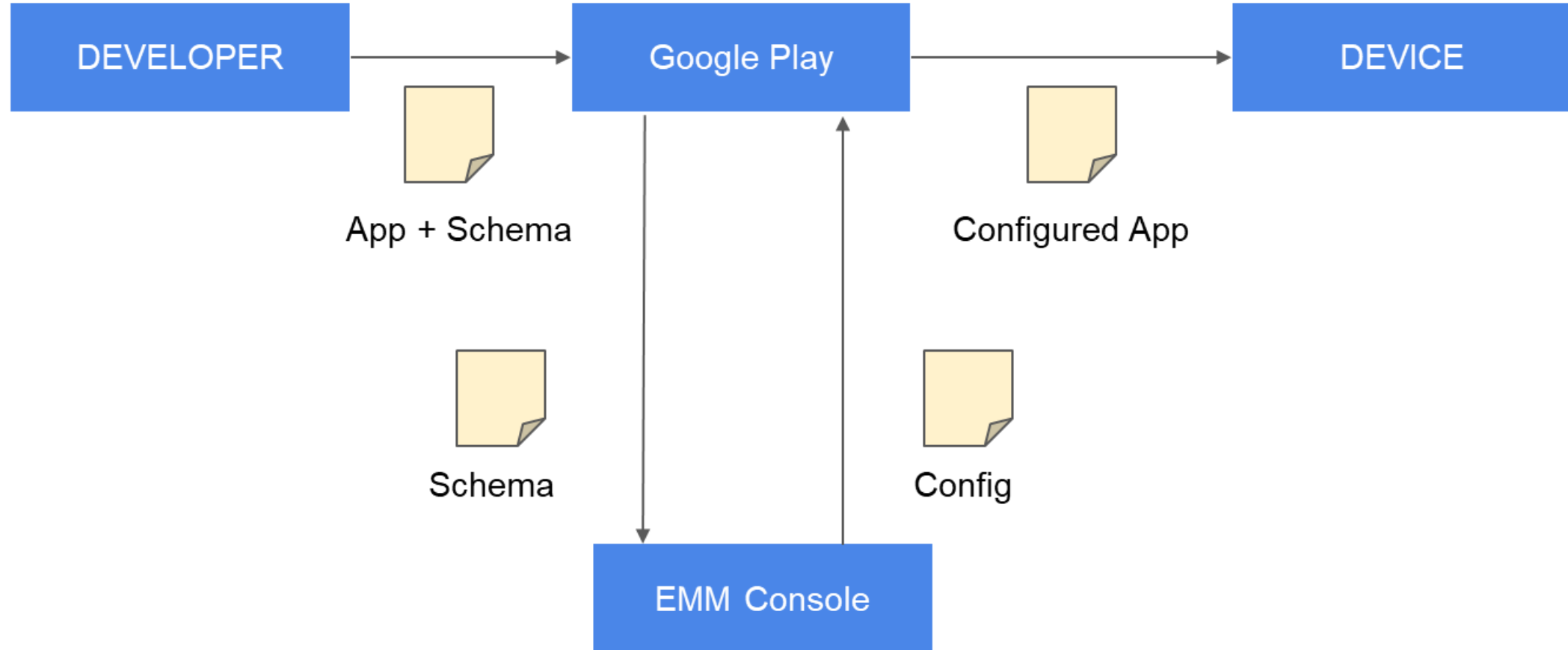
EMM (MDM) and Application Management

OEMConfig – Similarity between Managed config and OEMConfig



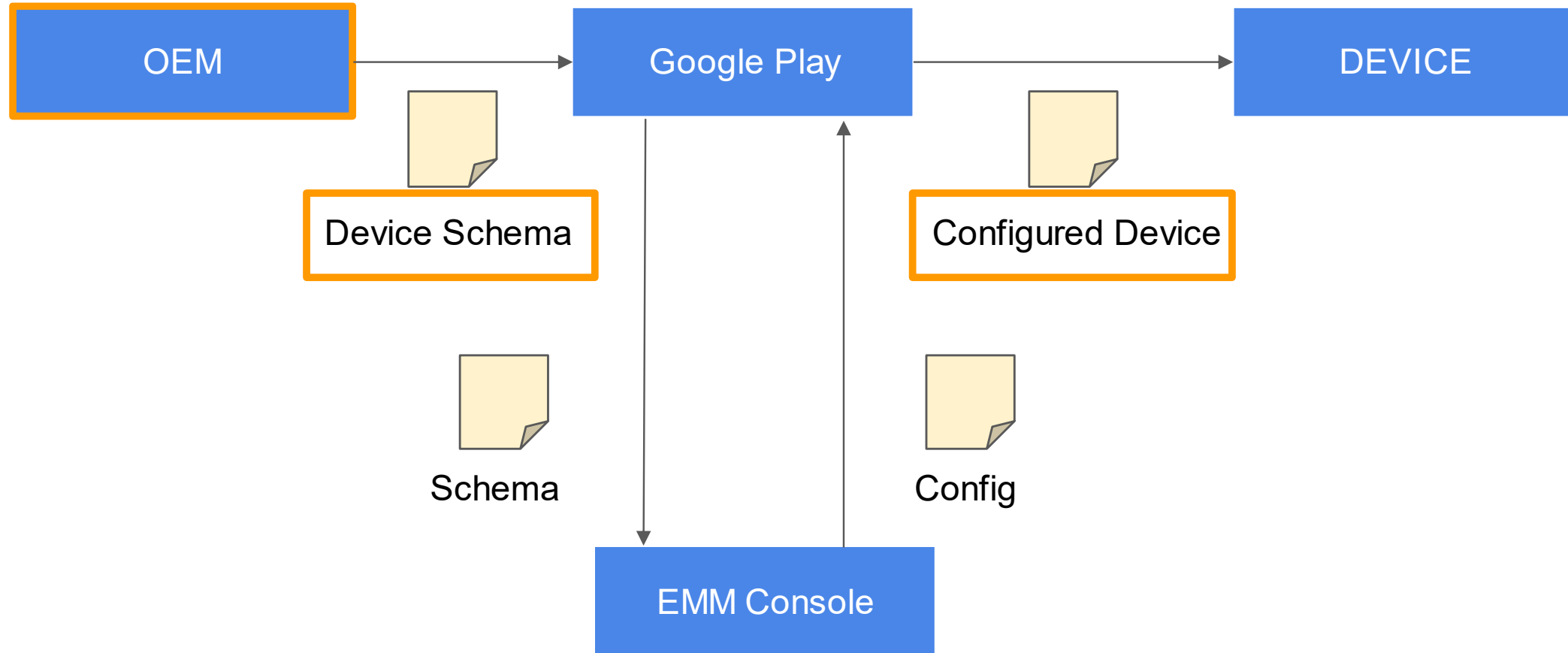
EMM (MDM) and Application Management

OEMConfig – Similarity between Managed config and OEMConfig



EMM (MDM) and Application Management

OEMConfig – Similarity between Managed config and OEMConfig



EMM (MDM) and Application Management

OEMConfig

- Zebra's OEMConfig application is available on the Play Store:
 - <https://play.google.com/store/apps/details?id=com.zebra.oemconfig.common>
- Aim is to expose the full capabilities of Mobility eXtensions in an **industry standard** way
 - Can take full advantage of Zebra's Android enhancements for enterprise without specifically targeting Zebra devices or code changes.
- More information available on techdocs: <https://techdocs.zebra.com/oemconfig/about/>
- Compare with the documentation for MX in general: <https://techdocs.zebra.com/mx/>. There will be close parity.
- Options will also be visible from your EMM console. EMMs will have documentation on how to access OEMConfig from their platform, e.g. [Intune](#), [SOTI](#), [WizyyEMM](#)



EMM (MDM) and Application Management

Other development considerations

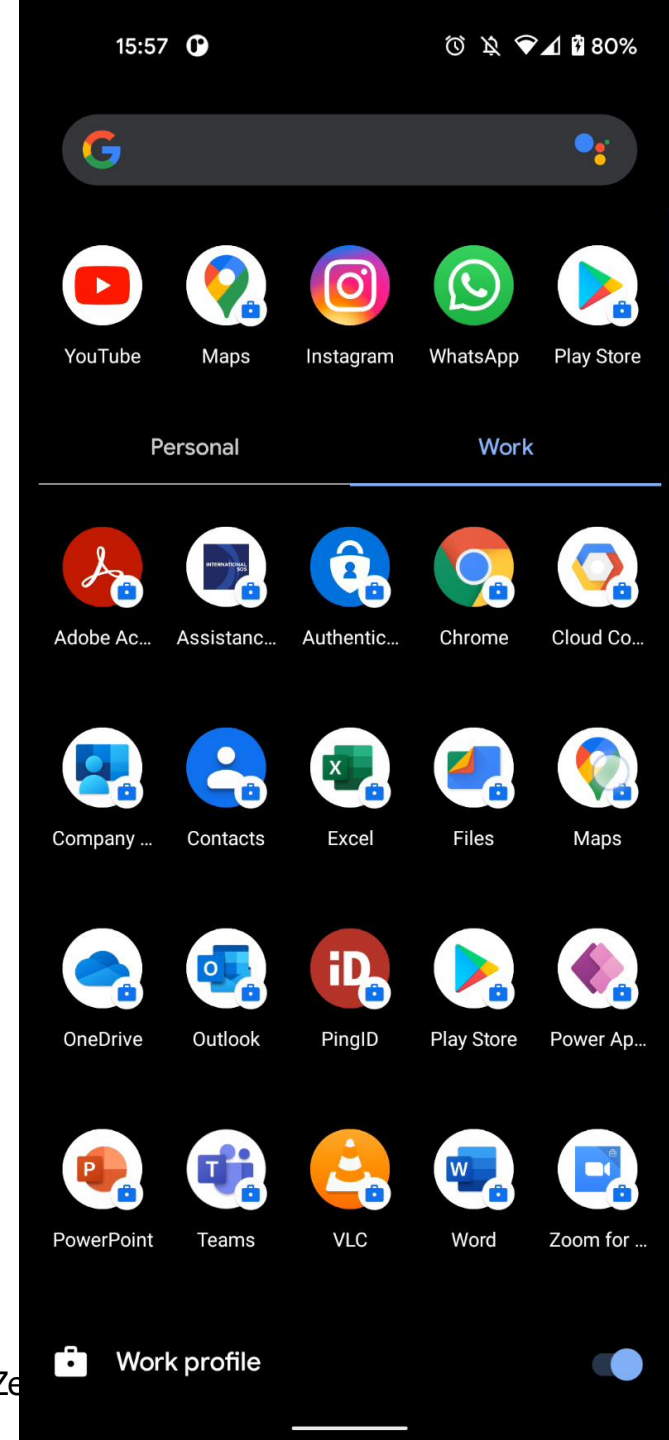
- The behaviour of the device may be locked down in unexpected ways
 - See [Device Policy Manager](#) for a full capabilities exposed to EMMs →
 - Examples:
 - [Disable account sign-in](#)
 - Disable camera
 - Disable backup
 - Disable location
 - & many more
- GMS services might not be available
- Settings screens might be restricted
- Notifications might be invisible

Developer Guides	
Summary	
Constants	
Public methods	
Inherited methods	
Constants	
void	addPermissionPreferredActivity(ComponentName admin, IntentFilter filter, ComponentName activity) Called by a profile owner or device owner to set a default activity that the system selects to handle intents that match the given IntentFilter.
void	addUserRestriction(ComponentName admin, String key) Called by a profile or device owner to set a user restriction specified by the key.
boolean	bindDeviceAdminServiceAdmin(ComponentName admin, Intent serviceIntent, ServiceConnection conn, int flags, IUserDeviceTargeter) Called by a device owner to bind to a service from a secondary managed user or profile.
boolean	canDeleteGuestSessionPermissions() Returns true if the caller is running on a device where the admin can grant permissions related to device sessions.
boolean	canObtainUiccSignalEnabled() Returns whether enabling or disabling UICC data signaling is supported on the device.
void	clearAppGrowthData(ComponentName admin, String packageName, IAppGrowthManager, IAppGrowthManager.Stub listener) Called by the device owner or profile owner to clear application user data of a given package.
void	clearCrossProfileIntentFilters(ComponentName admin) Called by a profile owner of a managed profile to remove the cross-profile intent filters that go from the managed profile to the parent, or from the parent to the managed profile.
void	clearDeviceOwnerApp(String packageName) This method was deprecated in API level 30. This method is expected to be used for testing purposes only. The device owner will lose control of the device and its data after calling it. In order to prevent any sensitive data that remains on the device, the device owner factory needs the device instead of calling this method. See wipeData().
void	clearPackagePermissionPreferredActivities(ComponentName admin, String packageName) Called by a profile owner or device owner to remove all permission handlers (permissions associated with the given package) that were set by addPermissionPreferredActivity(ComponentName, IntentFilter, ComponentName).
void	clearProfileOwner(ComponentName admin) This method was deprecated in API level 30. This method is expected to be used for testing purposes only. The profile owner will lose control of the device and its data after calling it. In order to prevent any sensitive data that remains on the device, the device owner factory needs the device instead of calling this method. See wipeData().
boolean	clearUserPassword(ComponentName admin) Called by a profile or device owner to remove the current password restriction.
void	clearUserRestriction(ComponentName admin, String key) Called by a profile or device owner to clear a user restriction specified by the key.
boolean	createAdminSupportScreen(String restriction) Called by any app to display a support dialog when a feature was disabled by an admin.
boolean	createAdminSupportScreen(ComponentName admin, String name, ComponentName profileOwner, PermissionObserver adminExtra, int flags) Called by a device owner to create a user with the specified name and a given component of the calling package as profile owner.
int	enableSystemApp(ComponentName admin, boolean intent) Re-enables a system app by intent that was disabled by default when the user was installed.
void	enableSystemApp(ComponentName admin, String packageName) Re-enables a system app that was disabled by default when the user was installed.
boolean	generateKeyPair(ComponentName admin, String algorithm, KeyGenParametersSpec keySpec, int subKeyTypeFlags) This API can be called by the following to generate a new physical public key pair: <ul style="list-style-type: none"> • Device owner • Profile owner • Delegated software installer • Credential management app If the device supports key generation via secure hardware, this method is useful for creating a key in KeyChain that never left the secure hardware.
String[]	getAccountTypesWithManagementDisabled() Gets the array of accounts for which account management is disabled by the profile owner or device owner.
List<ComponentName>	getActiveAdmins() Returns a list of all currently active device administrators' component names.
String	getAffiliationId(ComponentName admin) Returns the affiliation id previously set via setAffiliationId(ComponentName, Set), or an empty set if none has been set.
String	getDevicePolicyLockdownWithIntent(ComponentName admin) Called by device or profile owners to query the set of packages that are allowed to access the network directly when airplane mode is in lockdown mode but not connected.

EMM (MDM) and Application Management

Other development considerations

- The behaviour of the device may be locked down in unexpected ways
 - See [Device Policy Manager](#) for a full capabilities exposed to EMMs →
 - Examples:
 - [Disable account sign-in](#)
 - Disable camera
 - Disable backup
 - Disable location
 - & many more
- GMS services might not be available
- Settings screens might be restricted
- Notifications might be invisible



EMM (MDM) and Application Management

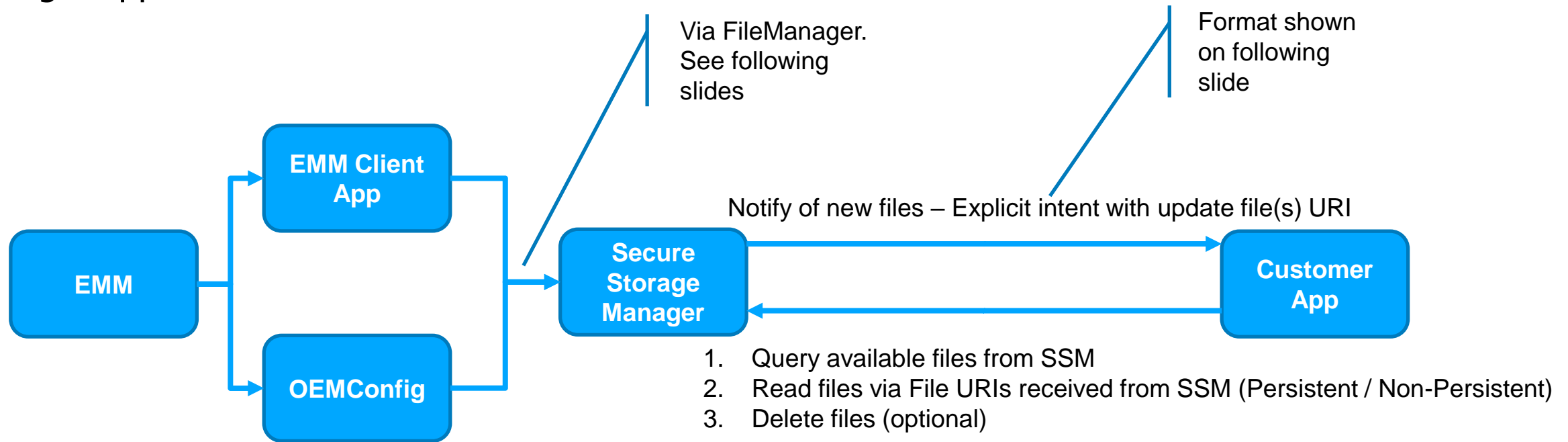
How to share a file with a specific app using SSM

- Before Scoped Storage: EMMs would download files required by applications, e.g. configuration files, and place them in a world-writeable location. The application would then read the file.
- With Scoped Storage: It is no longer possible for an EMM to place a file into a world-writeable location (touch-free) since these locations have been removed.
- Problem: How to meet the use case of an EMM sharing files with applications?
- The solution: Share files from the EMM client app to the customer app using [Android File Provider](#)

EMM (MDM) and Application Management

How to share a file with a specific app using SSM

The EMM app will make use of Zebra's Secure Storage Manager to exchange data securely to the target application



EMM (MDM) and Application Management

How to share a file with a specific app using SSM

Format of explicit intent sent from Secure Storage manager to customer app

Extra Key	Description	Example
filename	Filename or relative file path	Config.json or /config/Config.json
URI	content://com.zebra.securestoragemanager/config/Config.json	
isdir	Whether the file uri points to a directory or file.	True or false
crc	Check sum of the file. This can be checked by applications to see whether they already process this file or not.	

EMM (MDM) and Application Management

How to share a file with a specific app using SSM

How to specify the file to share from your EMM

Existing File Manager CSP has been enhanced. Can be defined via:

- OEMConfig (for EMM) or StageNow

New Action: *Deploy file for an application*

Properties to define: *Target application package, Target application signature.*

For more information on defining signatures I suggest <https://github.com/darryncampbell/MX-SignatureAuthentication-Demo>

Additional options: Whether the file should persist across enterprise reset

Define file as: On a remote server, on the device (subject to scoped storage restrictions), embedded

EMM (MDM) and Application Management

How to share a file with a specific app using SSM

How to specify the file to share from StageNow

Create New Setting

☐ Save Setting for Re-use ?

File Action: ?
Deploy file for an application ▼

Target Application File Definition: ?
com.symbol.datawedge/datawedge.db %

Target Application Signature: ?
%

Persist The File: ?

Do not persist

Persist

Share File Among Profiles: ?

Do not share among profiles

Share among profiles

Source Access Method: ?

File on a Remote Server

File in the Device File System

File embedded in XML

Source File URI: ?
.....

EMM (MDM) and Application Management

Testing your app: Test DPC

- Test DPC is available in the [Play Store](#) and on [github](#)
- Set to Device Owner as follows:

```
adb shell  
dpm set-device-owner  
"com.afwsamples.testdpc/.DeviceAdminReceiver"
```
- Be mindful of the difference between work profile and device owner
- Test DPC exposes (pretty much) the full capabilities that are available to an EMM, but runs entirely on-device.



Test DPC

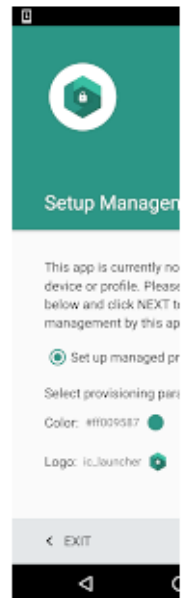
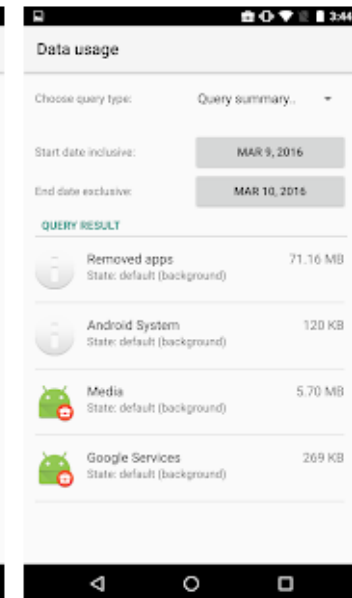
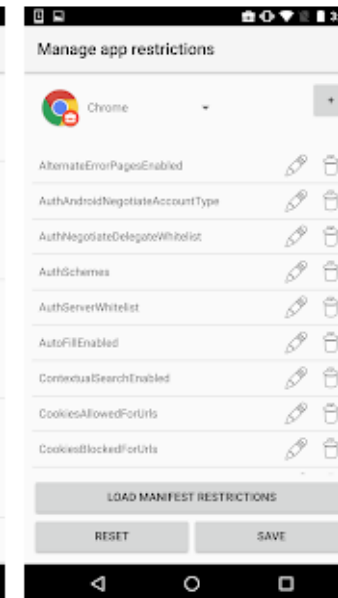
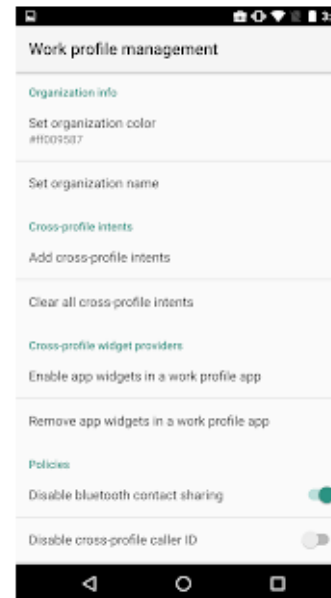
Sample developer Libraries & Demo

★★★★★ 910

PEGI 3

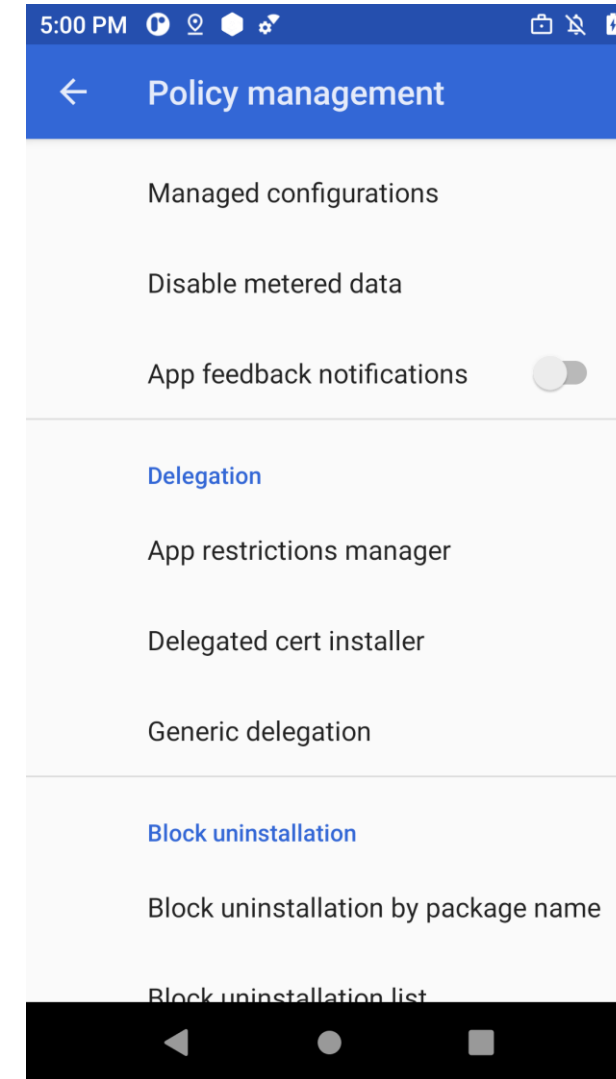
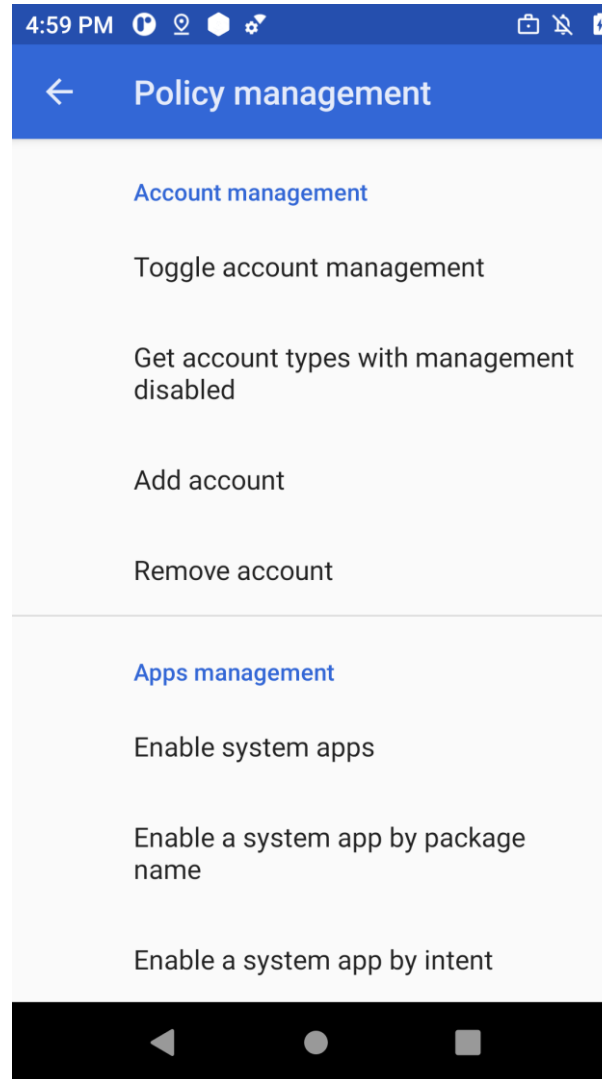
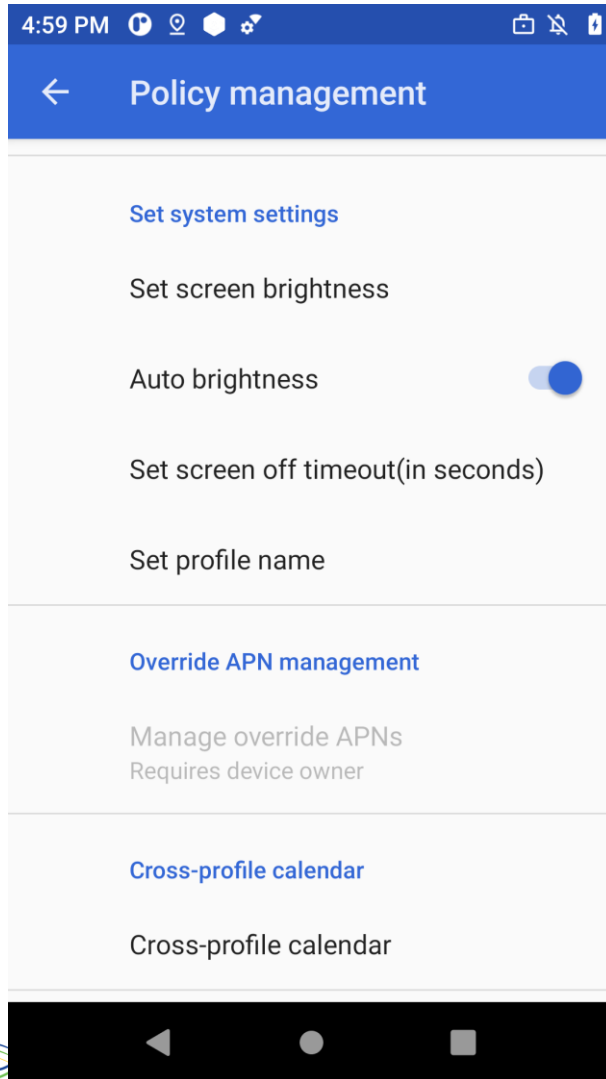
This app is available for all of your devices

Installed



EMM (MDM) and Application Management

Testing your app: Test DPC

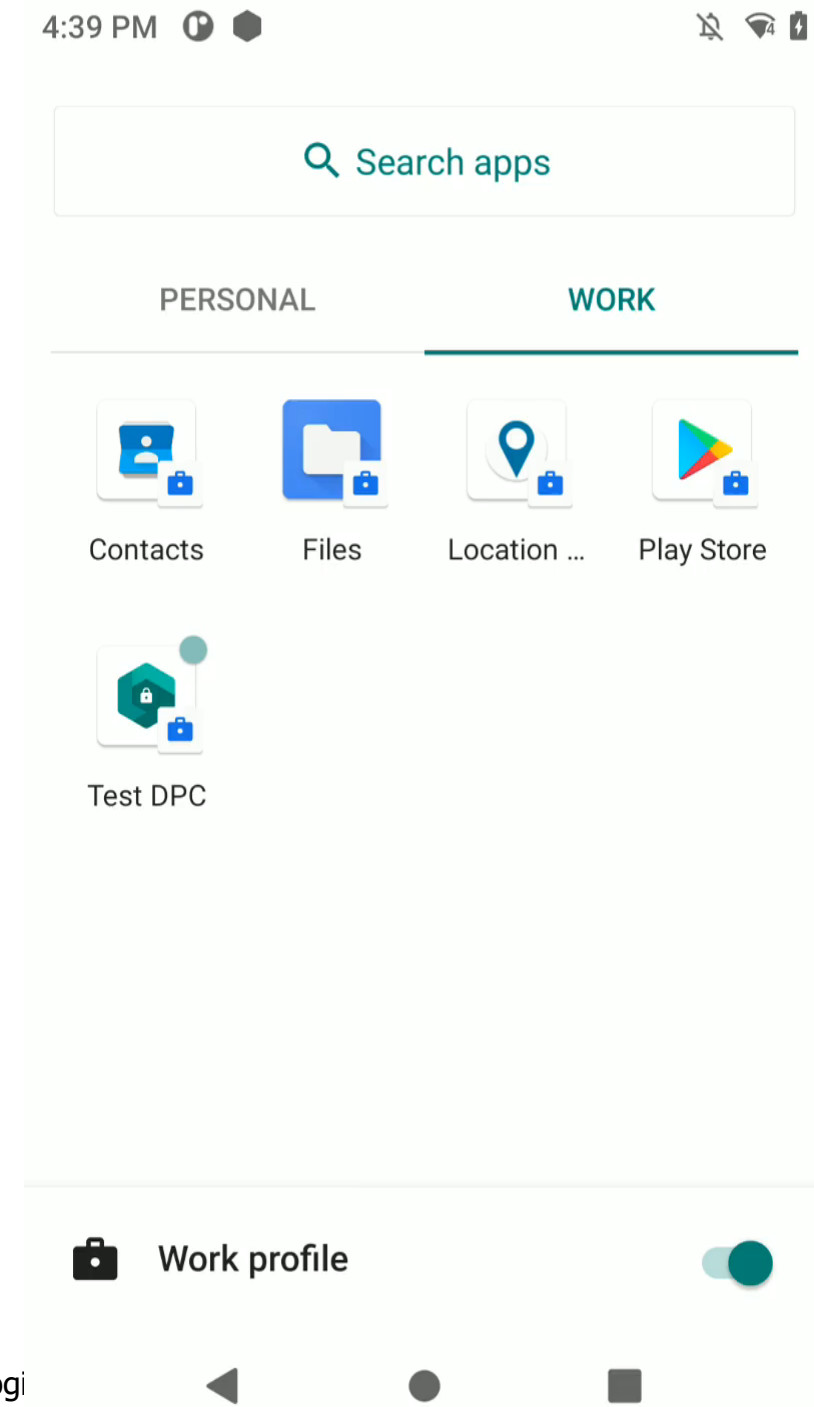


EMM (MDM) and App Management

Testing your app: Test DPC

Example:

- **Video shows an application managed by the Work Profile but the principle is the same for a fully managed device.**
- Application installed into Work Profile which accesses location
- Will request location permission when launched
- In work profile, expect location permission to be pre-granted
- Use Test DPC to specify the pre-approved runtime permissions



EMM (MDM) and Application Management

Resources

- Android Enterprise Recommended: https://www.android.com/intl/en_uk/enterprise/

EMM (MDM) and Application Management

Questions?



<https://developer.zebra.com>



Zebra Developers Community – LinkedIn Group



@ZebraDevs



<https://github.com/ZebraDevs>

Thank You



Zebra
DevCon 2021
Connect | Learn | Build

