Create your own Device Owner
Why? How? And why Not?

Darryn Campbell, Software Architect, Zebra Technologies

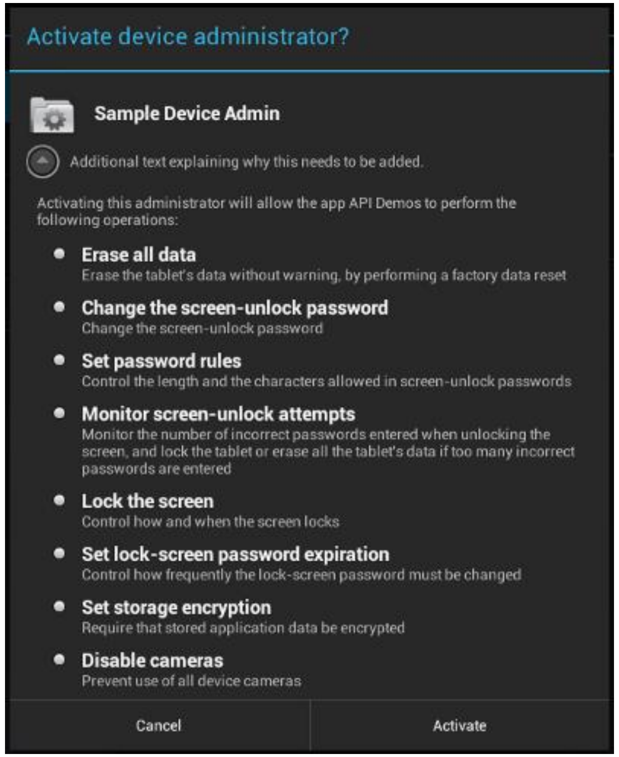
darryncampbell.co.uk @darryncampbell

DevTalk

Device admin is going away

Google is deprecating device admin and moving towards Android Enterprise

Are you already using the Device Admin APIs?? “We recommend that you begin to enroll all new company-owned devices running the major Android release after Oreo as managed devices, in preparation for the removal in the release after that.”



Activate device administrator?

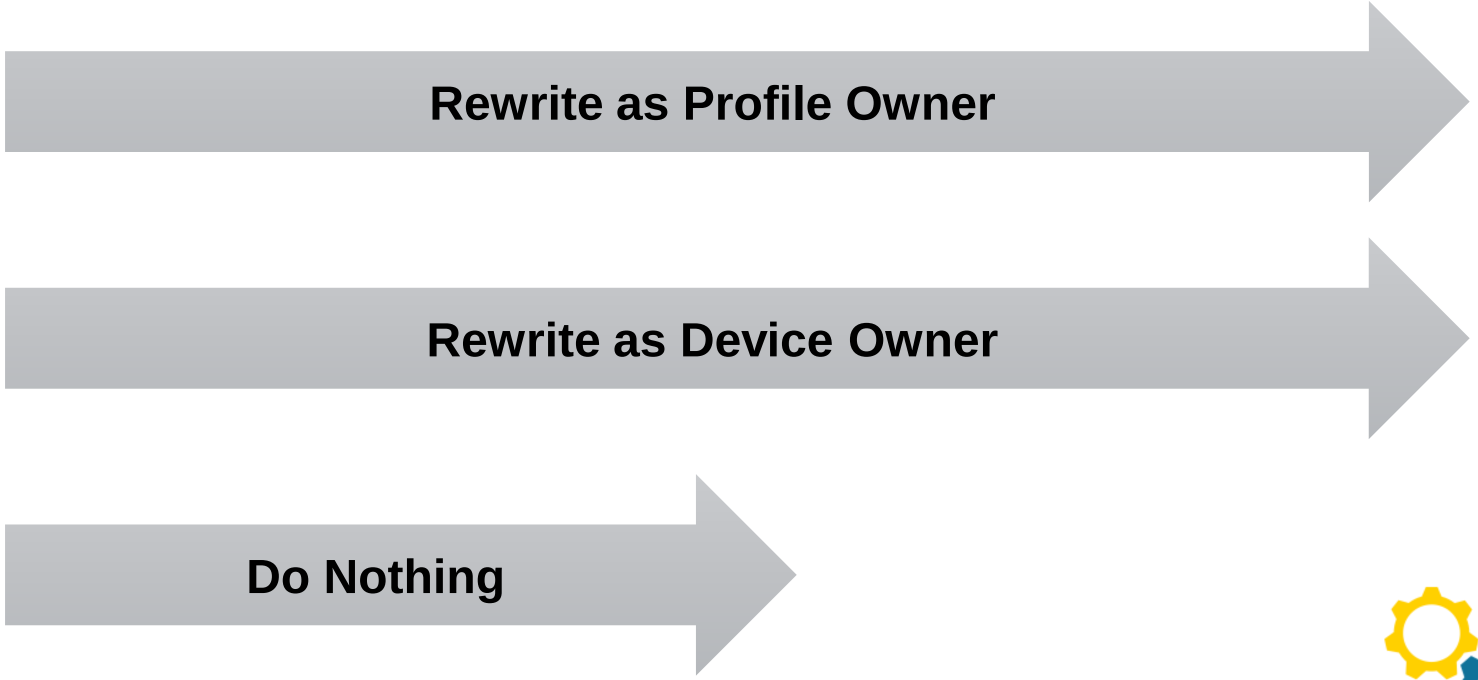
Sample Device Admin

Additional text explaining why this needs to be added.

Activating this administrator will allow the app API Demos to perform the following operations:

- **Erase all data**
  Erase the tablet's data without warning, by performing a factory data reset
- **Change the screen-unlock password**
  Change the screen-unlock password
- **Set password rules**
  Control the length and the characters allowed in screen-unlock passwords
- **Monitor screen-unlock attempts**
  Monitor the number of incorrect passwords entered when unlocking the screen, and lock the tablet or erase all the tablet's data if too many incorrect passwords are entered
- **Lock the screen**
  Control how and when the screen locks
- **Set lock-screen password expiration**
  Control how frequently the lock-screen password must be changed
- **Set storage encryption**
  Require that stored application data be encrypted
- **Disable cameras**
  Prevent use of all device cameras

Decision Time: Android Q



Rewrite as Profile Owner

Rewrite as Device Owner

Do Nothing

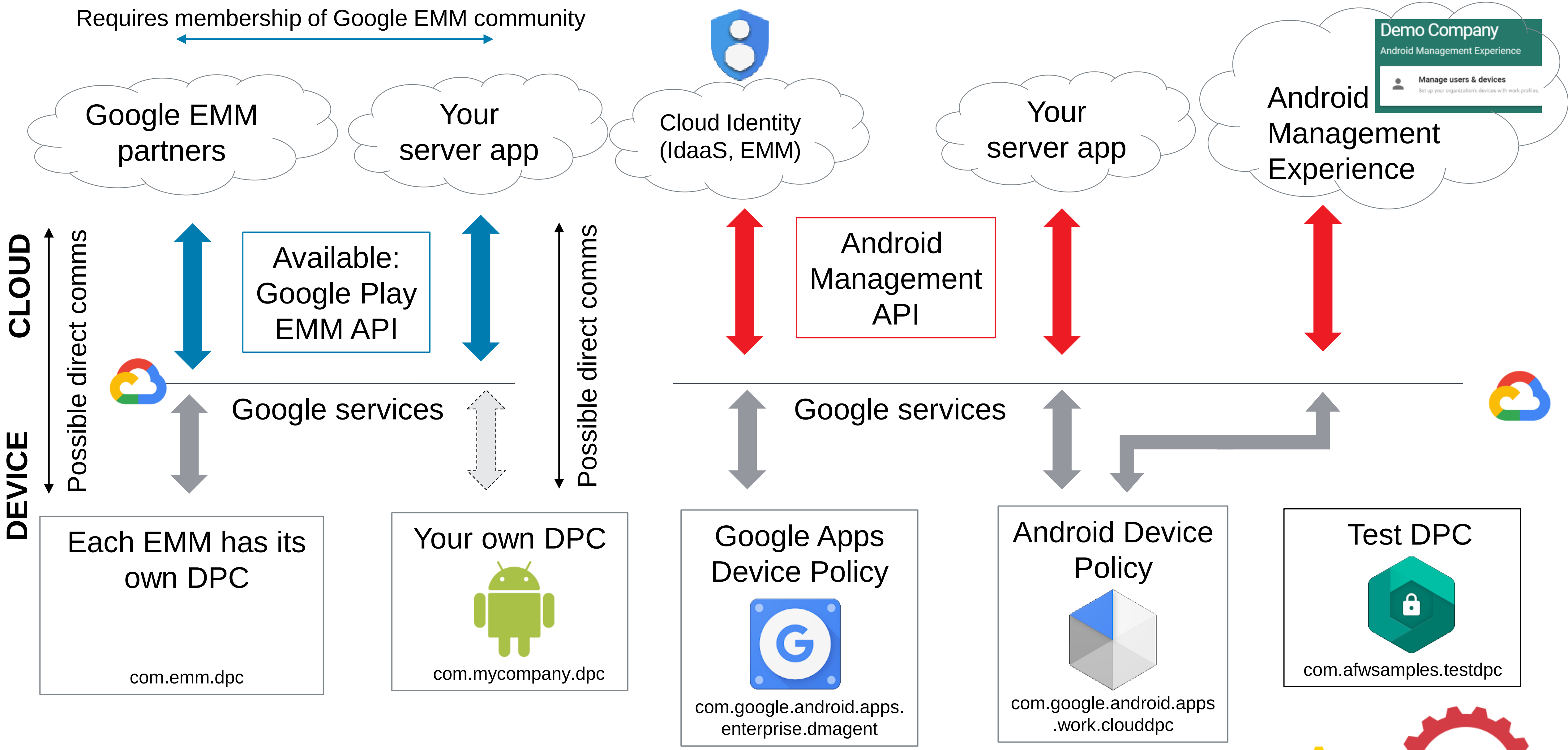
Questions to ask yourself

- Who owns the device?
  - Easiest way to determine whether to use Device owner or Profile owner
  - Single use device?
- Is my customer using an EMM?
  - If so, the administration application belongs to the EMM
  - Nothing for your to do… unless you are the EMM developer(!)
- Do I need / want to build my own EMM?
- Do my devices have network access to the cloud?
- What Accounts is my customer using?
  - GSuite? Google accounts? Managed Google Play accounts?
  - Do I need a Play account? How are apps being installed on the device?

ZEBRA

Questions you might be asking!

- What is Device owner mode? Profile owner mode?
- Do I need to use an EMM with DO / PO mode?
- How hard is it to build my own EMM?
- What is the difference between a “Managed Google Play Account”, a GSuite account and standard Google account?
- What has any of this got to do with building my own Device Owner?
- How does this all fit together? What is out there today?



Requires membership of Google EMM community

Google EMM partners

Your server app

Cloud Identity (IdaaS, EMM)

Your server app

Android Management Experience

Demo Company

Android Management Experience

Manage users & devices

Set up your organization's devices with work profiles.

CLOUD

DEVICE

Possible direct comms

Available: Google Play EMM API

Possible direct comms

Android Management API

Google services

Google services

Each EMM has its own DPC

com.emm.dpc

Your own DPC

com.mycompany.dpc

Google Apps Device Policy

com.google.android.apps.enterprise.dmagent

Android Device Policy

com.google.android.apps.work.clouddpc

Test DPC

com.afwsamples.testdpc

ZEBRA

Zebra Technologies

Overview

ZEBRA

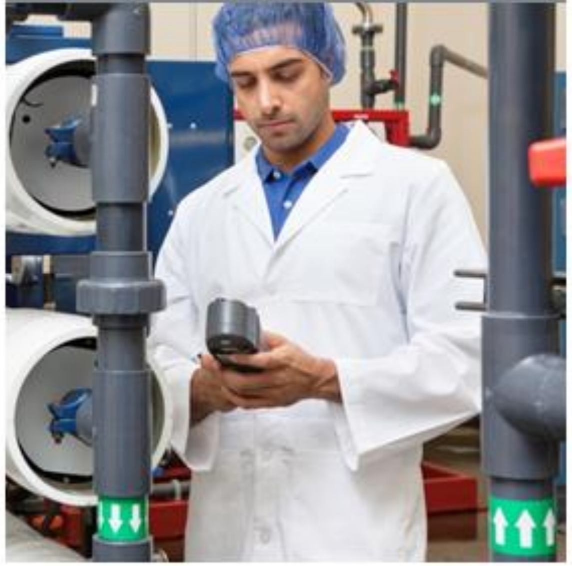
Zebra Technologies

Overview

RETAIL

T&L

MANUFACTURING



HEALTHCARE

Why create your own Device Owner?

Understanding who might do this

- Devices are owned by the business, not the end user. They have a single purpose.
  - Are you targeting apps for these devices using OEM specific management APIs?
- How does this fit in with an EMM:
  - Either: you do not want to use an EMM
  - Or: you plan on writing the Device Owner, your own EMM server component and perhaps joining the Google EMM community
    - This process is NOT just a tick in the box
    - Practically, you would be writing a direct communication channel between your server and DO
  - Need to run on premises?
    - Avoid interfacing with Google cloud APIs (privacy concerns?)

Why create your own Device Owner?

Understanding who might do this

Application management:

- Either: You are going to use the managed Play Store
  - Secure
  - User familiarity
  - Access to millions of licensed & free applications
  - Supports private application distribution
  - Requires use of Google's proprietary cloud APIs
- Or: You are going to provision applications on the device yourself.
  - Very feasible on single-use devices where OEMs provide staging tools
  - Zebra provides a tool called StageNow which enables this
  - Can be combined with other aspects of the staging process

Why create your own Device Owner?

Access to DPM APIs and the parent android.app.admin namespace classes

- Install custom certificates
- **Application management**
  - Enable system apps
  - Hide apps
  - Suspend apps
  - Block app uninstallation
- Disable camera
- **Disable screen capture**
- **Configure Wi-Fi**
  - See MAC address
- Set input methods
- Lock screen restrictions

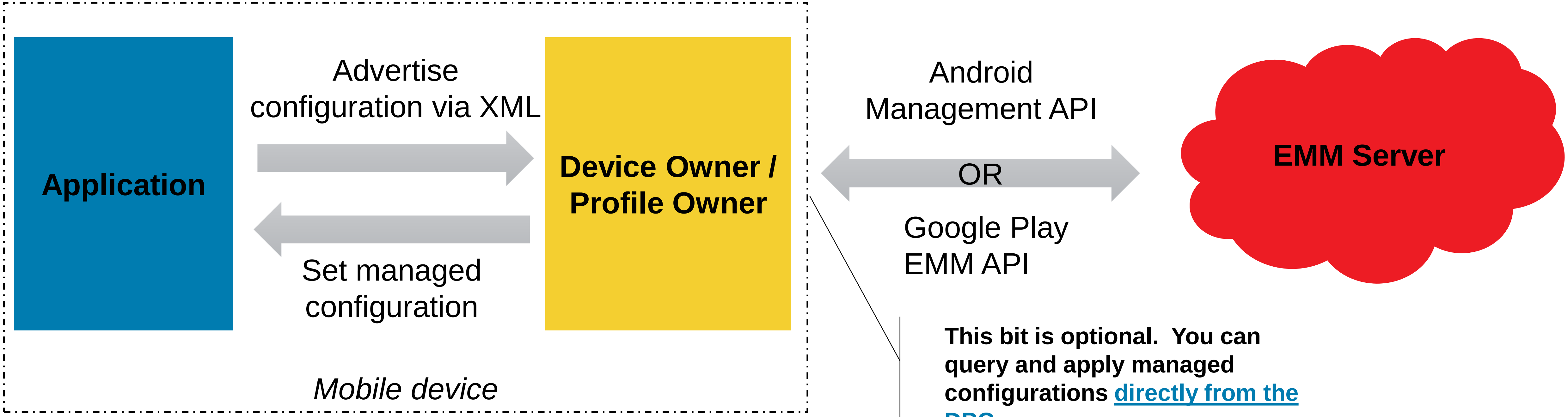
- Password constraints
- Lock the device
- Initiate a password reset
- **Lock task mode**
  - Manage whitelisted applications
  - Start / stop lock task mode
- Set networking
  - Always-on VPN
  - Global HTTP proxy
- **Single-use devices**
  - Disable status bar

  - Keyguard
  - Kiosk mode
- Install from unknown sources
- Debugging
  - Enable process logging
  - Request bug report
- **Wipe data**
- Reboot device.
- DeviceAdminService
- Much more…

Why create your own Device Owner?

Other considerations – Managed configurations

- A point about Managed configurations (formally Application Restrictions)



Application

Advertise configuration via XML

Set managed configuration

Device Owner / Profile Owner

Mobile device

Android Management API

OR

Google Play EMM API

EMM Server

This bit is optional. You can query and apply managed configurations directly from the DPC
Additionally, you could implement the server communication yourself and not go through the Google API

Why create your own Device Owner?

Other considerations – Accounts

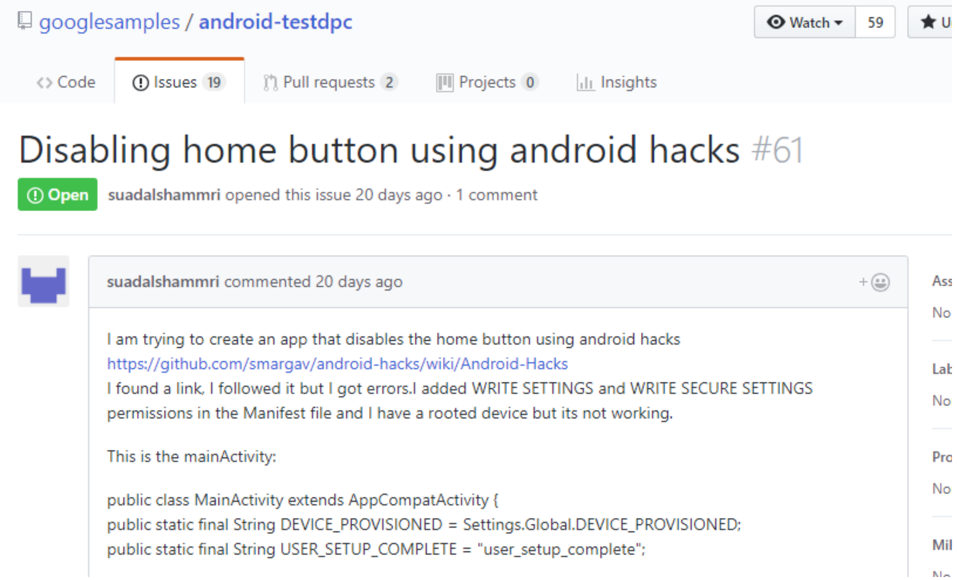
A point about accounts

- Traditional Android identity management revolved around Google accounts tied to **individuals**
  - @gmail.com
  - Gsuite accounts
- This does not work in the enterprise. Users share devices and do not own them
  - Though addressed in ‘P’ with ephemeral users, there has never been the notion of ‘logging out’ after your shift
- Managed Play accounts begin to address this:
  - Account is mapped at the device level, an enterprise owns multiple devices, not users
  - EMM is responsible for mapping devices to organization’s identity provider if required
- Account management is facilitated by the Device Owner
- You do not NEED a managed account to use a Device Owner, though it is encouraged.

Why not create your own Device Owner?

Not to circumvent system restrictions – like a sledgehammer to crack a nut(!)

- Not to circumvent restrictions - [Example](#).



Disabling home button using android hacks #61

Open suadalshammri opened this issue 20 days ago · 1 comment

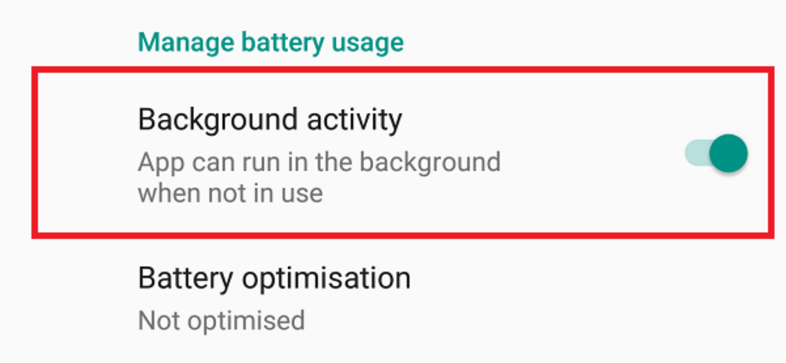
suadalshammri commented 20 days ago

I am trying to create an app that disables the home button using android hacks
https://github.com/smargav/android-hacks/wiki/Android-Hacks
I found a link, I followed it but I got errors.I added WRITE SETTINGS and WRITE SECURE SETTINGS permissions in the Manifest file and I have a rooted device but its not working.

This is the mainActivity:

```
public class MainActivity extends AppCompatActivity {
public static final String DEVICE_PROVISIONED = Settings.Global.DEVICE_PROVISIONED;
public static final String USER_SETUP_COMPLETE = "user_setup_complete";
```

- Or maybe bypassing Oreo background restrictions by using a [DeviceAdminService](#).



Background activity
App can run in the background when not in use

Battery optimisation
Not optimised

Why not create your own Device Owner?

Device Owner mode might not be a good fit for your deployment…

- You do not own the device or you are not deploying to Single Use devices
  - You might consider creating a Profile Owner in this case
- Total cost of development / maintenance might outstrip cost of EMM deployment
- You are using Android Lollipop or earlier in your deployment
- Device must be new or factory reset
  - Unlike when moving to Profile owner
  - Could be a positive: Install a dummy DO to prevent somebody else gaining privileged access
- How will your solution scale? How does that compare with an EMM scaling?
- Google have a related section on their site for "Implementation considerations for device owner mode."

Why not create your own Device Owner?

Why else?

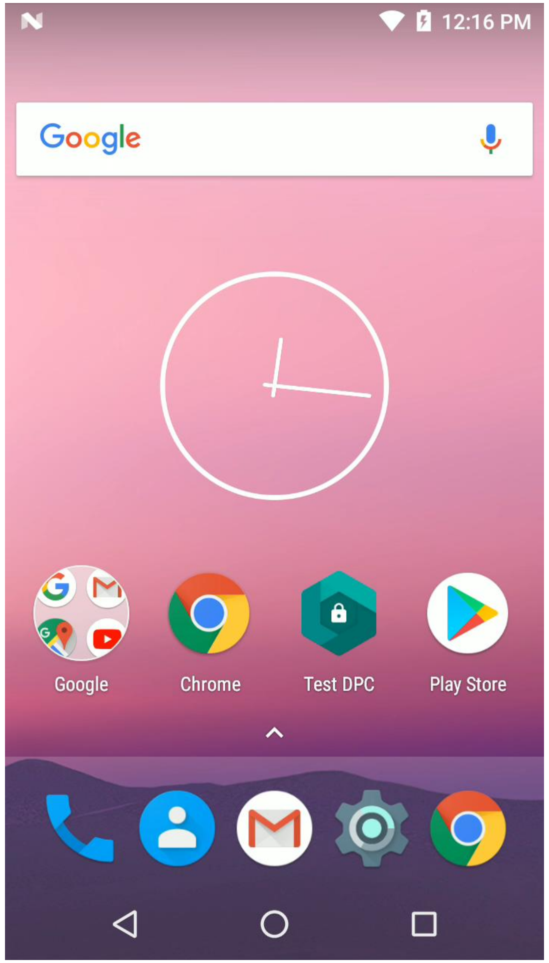
- Provisioning a device into Device Owner mode is very opinionated – will this fit with your deployment?
  - From setup wizard scan QR code **– will download the app from the Play Store**
  - From setup wizard tap NFC tag **– will download the app from the Play Store**
  - Zero touch enrolment **– will not work with a custom DO without an EMM**

- Easier solutions may exist e.g.
  - Build your own EMM with the Android Management API
  - If you need all this control, maybe consider using an EMM.
    - All EMMs provide their own device owner.

How to create your own Device Owner?

Test DPC is your friend

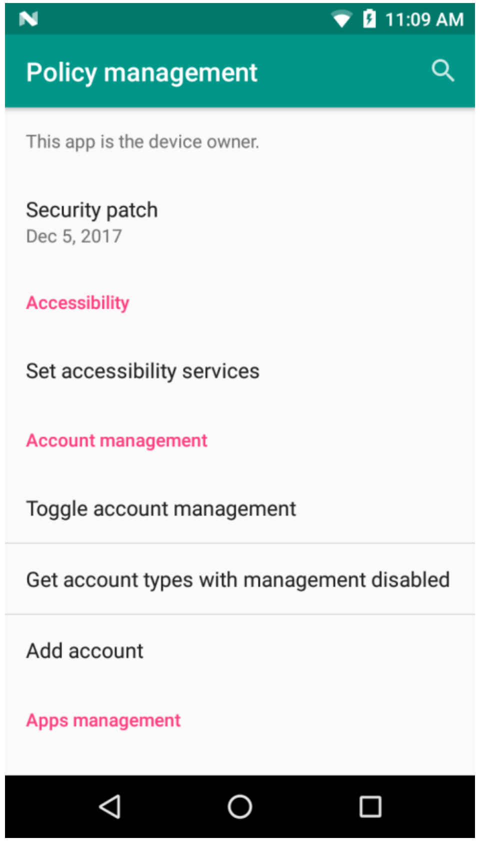
- Available in the Play Store & GitHub
- Always updated with the latest features
  - P features / code have been in the repo for over a month
- Shows the full gamut of DPC features for both PO & DO
  - Facilitates understanding & shows capabilities
- Enrol a new Android device into DO mode with afw#testdpc in the setup wizard email field
  - Or, enrol into PO mode from the App itself



How to create your own Device Owner?

Guides? Provisioning?

- No guide exists to create a Device Owner
  - There is a guide to create a profile owner.
- Understand the end to end process:
  - How will you provision the Device Owner on your device?
    - OEM solutions may be available – e.g. Zebra's StageNow
    - User driven workflow from the setup wizard in Android Enterprise
      - Suitable for BYOD
      - Requires touching every device
  - What is interacting with the Device Owner? If not an EMM.
    - How will you configure the DO on device?
- DevicePolicyManager API


11:09 AM

Policy management

This app is the device owner.

Security patch
Dec 5, 2017

Accessibility

Set accessibility services

Account management

Toggle account management

Get account types with management disabled

Add account

Apps management

How to create your own Device Owner?

Overview

- Testing:
  - Remove all other admin accounts
  - `adb shell dpm set-device-owner "com.afwsamples.testdpc/.DeviceAdminReceiver"`
    - Values for TestDPC: Package name / class name will differ depending on DO
  - OEM solutions may be available

```
<wap-provisioningdoc>
    <characteristic type="Intent" version="7.0" >
        <parm name="Action" value="EnrollDeviceOwner"/>
        <parm name="Package" value="com.afwsamples.testdpc"/>
        <parm name="Class" value="com.afwsamples.testdpc.DeviceAdminReceiver"/>
    </characteristic>
</wap-provisioningdoc>
```

More information

Resources

- Device admin mode is going away
- Deployment scenarios from Google developer site for EMM developers.
- TestDPC: GitHub, Play Store
- Android Management Experience
- Google Play EMM API
- Android Management API (beta)
- Managed Configurations
- DevicePolicyManager API and parent android.app.admin namespace

Questions?

Questions?