DEVELOPING APPLICATIONS TARGETING PURPOSE-BUILT ANDROID DEVICES

Darryn Campbell – Enterprise development architect
Zebra Technologies – darryncampbell@zebra.com
14th August 2017

Introduction

Who Am I?

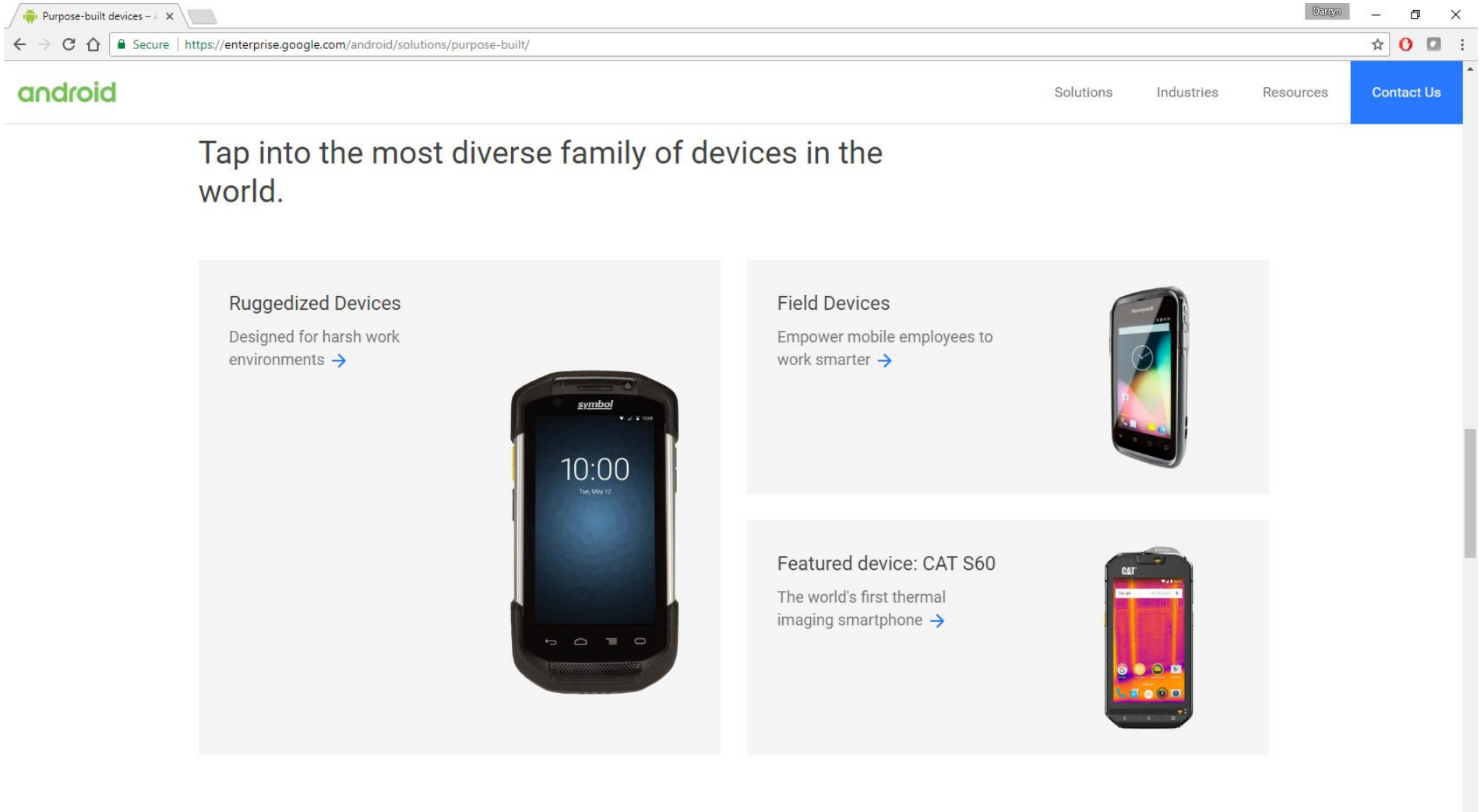
- **Darryn Campbell**
- Architect at Zebra technologies based out of the UK.
  - Manufacturers of enterprise hardware, software and services
  - Sponsor and host various developer events – come see us at Droidcon London.
- I work with developers targeting our handheld devices and printers to ensure the best possible experience
- Blog: http://darryncampbell.co.uk
- GitHub: https://github.com/darryncampbell

What are Purpose-built devices?

Examples:

Google’s focus on Enterprise

https://enterprise.google.com/android/solutions/purpose-built



What is COSU? BYOD? COPE? COMP?

What happened to Android for Work?

- Android for Work started with 5.0 (lollipop). Announced at Google I/O 2014
- Initially targeted BYOD (Bring your own device) use cases
  - Separation of 'Work' mode from personal applications
  - 'Work profile' owned by a "Profile Owner" which would be a device policy controller (DPC)
- Enhancements for COPE (Corporate owned, personally enabled)
  - Expectation that device or profile will owned by a DPC
  - DPC is acting as device owner (DO) or profile owner (PO)
- Enhancements for COSU (Corporate owned, single use) [6.0+]
  - Expectation that these devices will only have a device owner (DO)
  - Think parcel delivery, supermarket self scan or airport check-in kiosk
- 'O' has announced COMP (Corporate owned, managed profile) [8.0+]
- DO provisioning via NFC prioritized by Google (also possible via adb)

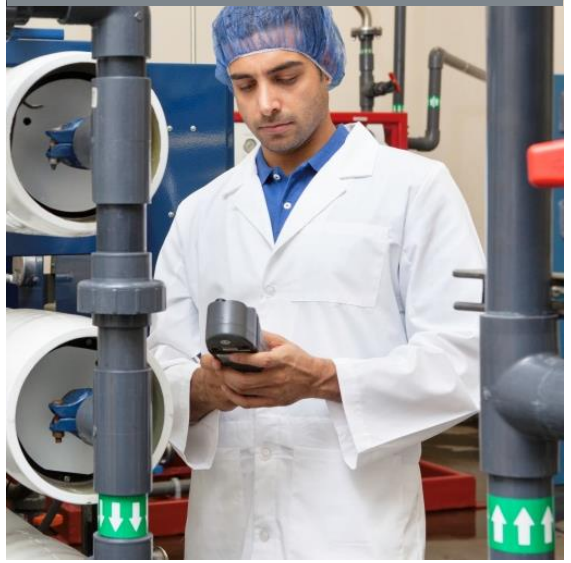
Zebra Technologies: Visibility Across Industries

Zebra makes businesses as smart and connected as the world we live in

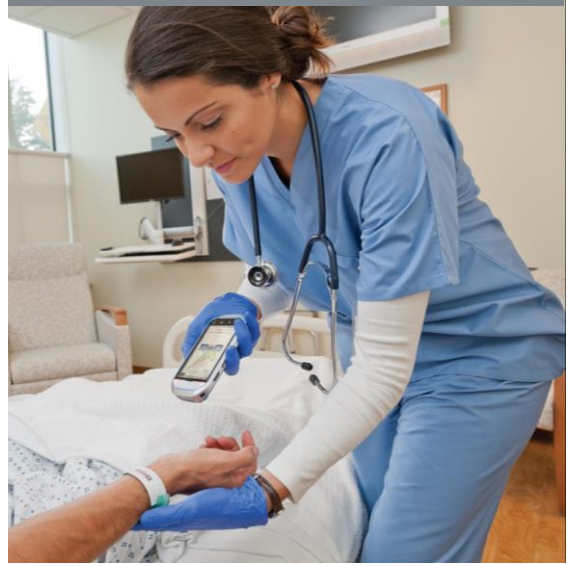
RETAIL

T&L

MANUFACTURING



HEALTHCARE



Simplify Operations

Empower Mobile Workers

Know More About Your Business and Customers

Developer Considerations

End user engagement

- Priorities of consumer applications are:
  - User engagement. More time in app = more ££
  - Providing a broadly appealing service to maximize download numbers.

- Priorities of enterprise devices users are:
  - Get their job done quickly & efficiently
  - Find the information they need to progress
  - Minimal set-up and delays (batteries, Bluetooth peripherals)

Developer Considerations

Knowledge of deployment

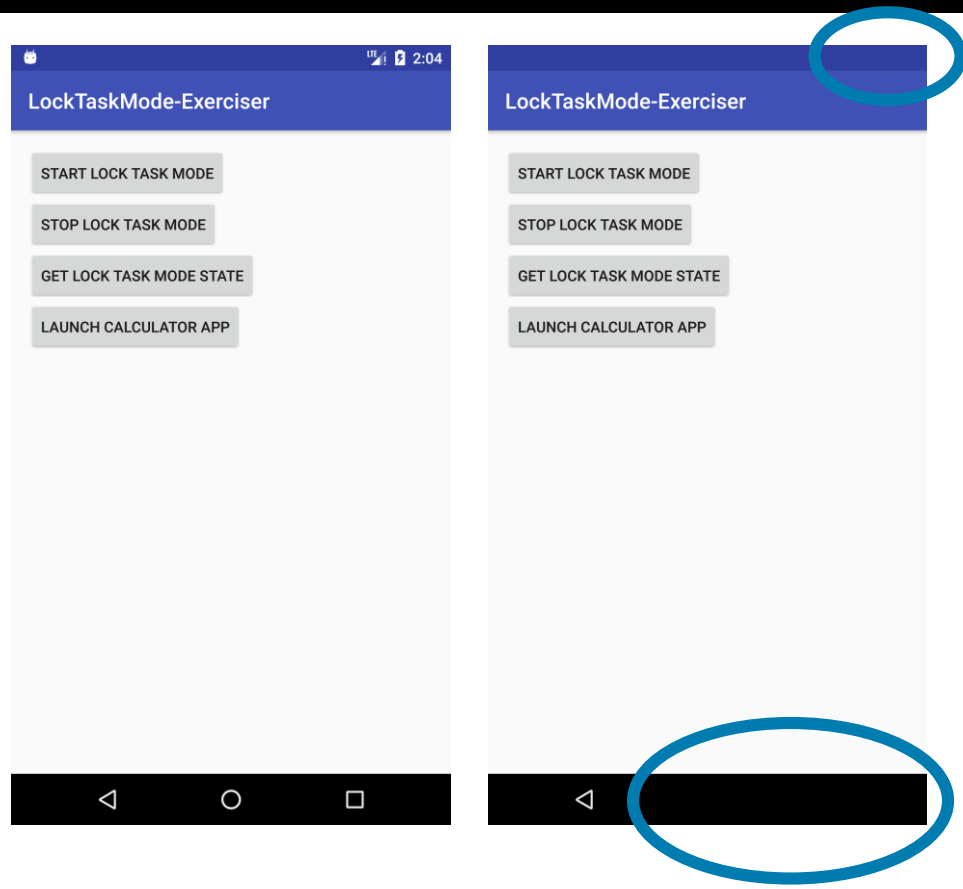
- Who will be staging & provisioning the enterprise devices?
- Is your application configurable?
  - If so, how?
    - Managed configurations / application restrictions?
    - Config file? Ini file?
    - Server based
  - How does that fit in with IT provisioning?
- Minimize (Eliminate) user configuration through UI

Developer Considerations

Device is in lock-down

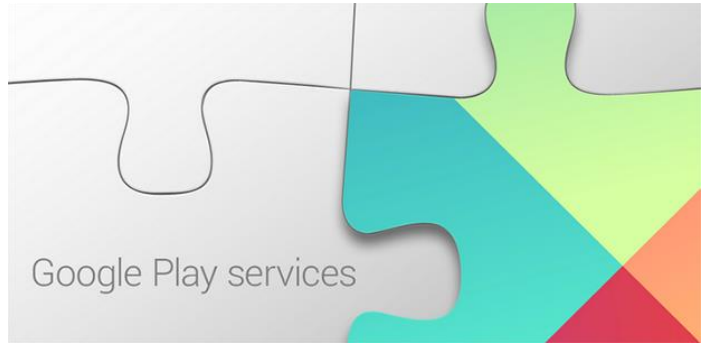
- Do not rely on the developer doing any configuration e.g.:
  - Accepting runtime permissions
  - Enabling mock locations
  - Whitelisting the app from doze mode

  Staging is usually (not always) a separate role
- Device will be locked down.
  - What are your application dependencies?
- Lock task mode vs. app pinning



Developer Considerations

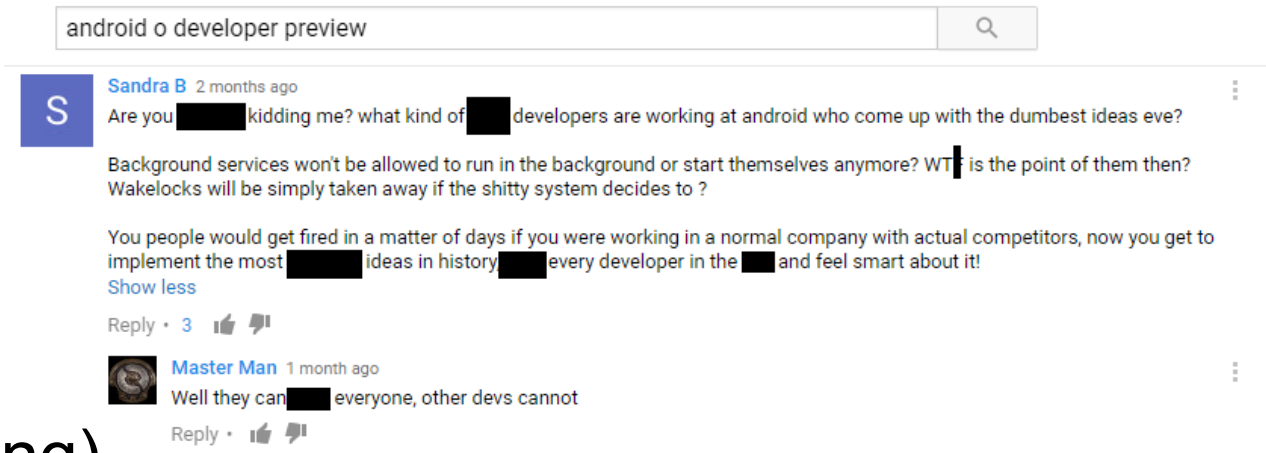
GMS (Google Mobility Services) or No GMS?



- Some devices:
  - Have full use of GMS and are usually online
  - Have GMS installed but IT may block some packages from running
  - Have GMS installed but sit behind a firewall
    - Consider the location service APIs which offer ‘ActivityRecognition’; this works offline but the “Nearby” APIs require a connection.
  - Have GMS installed but not agreed to Google’s terms of service for location
  - Do not have GMS at all and are AOSP only
    - Not affected by doze mode but how to do Maps? Push notifications?

Developer Considerations

Background processing considerations



- Background restrictions continue:
  - Job Scheduler in ‘L’
  - Doze mode in ‘M’ (& ‘N’ whilst moving)
  - “Background optimizations” in ‘O’
- Many Enterprise applications have Real time requirements:
  - My on-call physician MUST receive the push notification immediately
  - Customer requires assistance in aisle 4 NOW
- Enterprise hardware providers offer ways to circumvent this: both Zebra & Honeywell can exempt an application from doze mode.

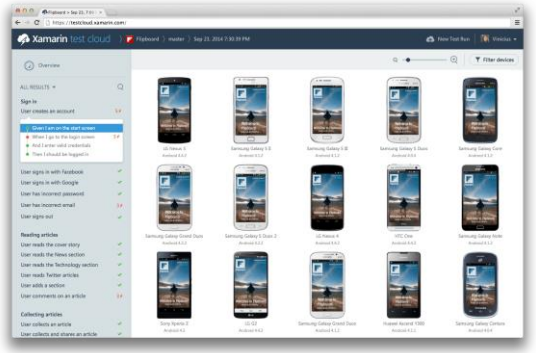
Developer Considerations

Proprietary hardware = Proprietary APIs (?)

- Enterprise hardware has barcode scanners, card readers, special screens for use with gloves, secure NFC modules etc etc.
- Typically you expect to use separate APIs. Supported languages?
- Many large companies use different hardware vendors
- ‘Wedge’ capture is a de-facto industry standard for barcodes:
  - Barcode data captured as string or HEX data.
  - Injected to the current application as key strokes
  - Very basic but will work across hardware vendors

Developer Considerations

Testing



- No emulators are available for enterprise hardware
  - How to test new devices? How to test bespoke hardware?
  - Partner programs available for test hardware
- Mocked APIs are not supported by vendors
  - You could do all the work yourself but that is not ideal:

    http://www.darryncampbell.co.uk/2017/03/01/instrumented-testing-and-the-zebra-emdk-barcode-api/
- Cloud testing services do not contain enterprise devices
  - Even if they did, still how would you run JUnit (or whatever) tests if you can't pull the barcode trigger or touch your secure NFC tag to the reader

Developer Considerations

Longevity of deployment

- Enterprise devices typically last 5+ years
- Devices will be upgradable (how much so depends on manufacturer)
  - Officially current desert + 1 for Zebra devices. Unofficially it can stretch to + 2
- OS upgrades bring security – great for IT
- OS upgrades bring new features – great for developers, meh for IT
- OS upgrades often bring breaking changes – bad for developers
- Again, different from consumer where applications are continually updated and breaking changes are rapidly fixed.

Developer Considerations

Support

- The good:
  - Enterprise organizations often have robust software testing
  - Paid support options often available
  - Support to extend beyond the lifetime of a typical Android device
- The bad:
  - Proprietary APIs and support. Limited StackOverflow and OSS
  - Smaller community than Android in general
  - Developers might be under NDA or for other reasons not want to assist in community support (too busy?)

CLOSING

In Closing

Zebra is keen to engage with current and potential developers

- Why Zebra cares? 
  - Developers have great influence over purchasing decisions
  - Good development experience makes our hardware better
  - Understand developer needs, especially the silent majority.

- Engage on Twitter:
  - @ZebraDevs
  - @ZebraEMEA
  - @darryncampbell

ZEBRA

QUESTIONS?

ZEBRA

THANK YOU