THALES

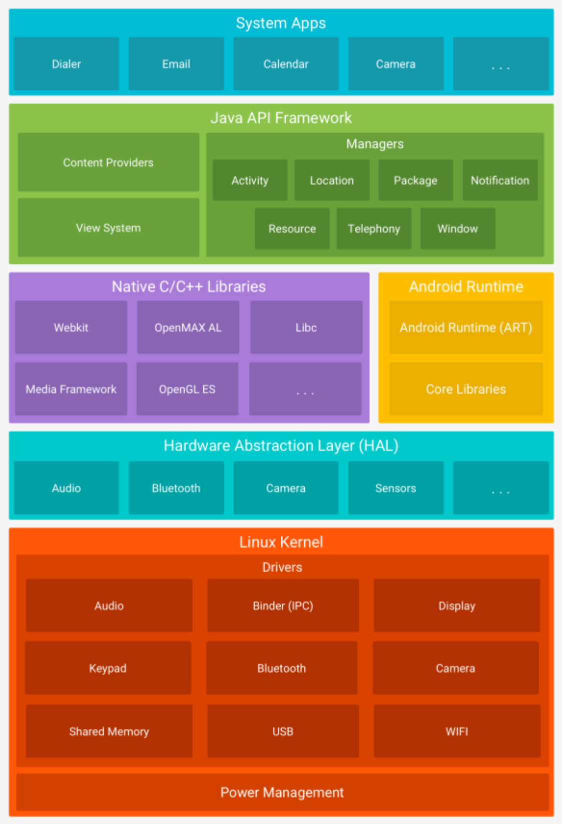
Android 101

Craig Mowder – Technical Sales
December 2019

www.thalesgroup.com

Android Layers

- Android is a modular platform with multiple layers
  - Kernel – controls underlying functions like threading and memory management
  - Hardware Abstraction – Provides standard interfaces to expose hardware capabilities (camera, Bluetooth, our modules)
  - Android Runtime
    - Since Android 5, each application has it's own Android Runtime (ART), running on separate virtual machines
    - ART replaced Dalvik, with a tech preview in KitKat 4.4. ART is backwards compatible with Dalvik, but Dalvik might not be compatible with ART.
  - C Libraries
    - Lower level libraries that run things like ART and the Hardware Abstraction Layer
    - Can be accessed directly by System apps



System Apps

Dialer | Email | Calendar | Camera | . . .

Java API Framework

Content Providers

View System

Managers

Activity | Location | Package | Notification

Resource | Telephony | Window

Native C/C++ Libraries

Webkit | OpenMAX AL | Libc

Media Framework | OpenGL ES | . . .

Android Runtime

Android Runtime (ART)

Core Libraries

Hardware Abstraction Layer (HAL)

Audio | Bluetooth | Camera | Sensors | . . .

Linux Kernel

Drivers

Audio | Binder (IPC) | Display

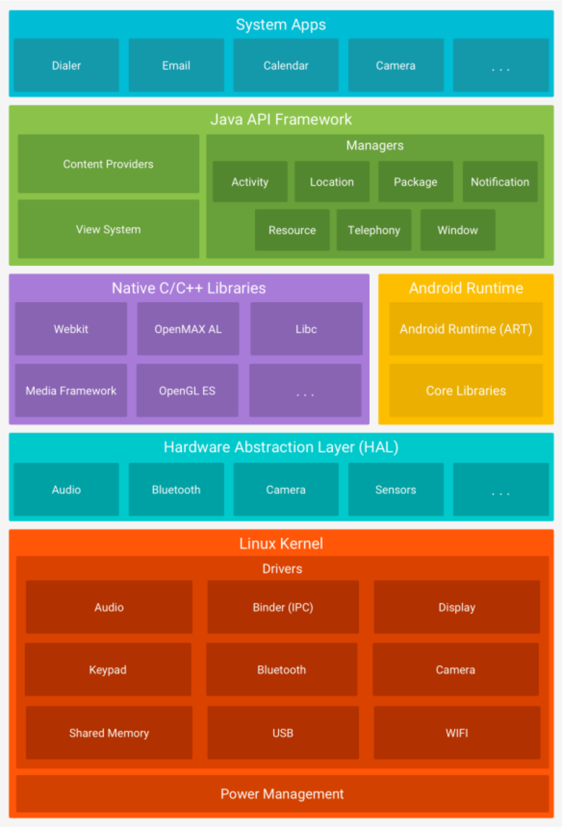
Keypad | Bluetooth | Camera

Shared Memory | USB | WIFI

Power Management

Android Layers

- Java API – higher level library with specific Java APIs for working with Android UI, resource managers, notification managers, activity managers, and content providers
- Application layer
  - End programming layer
  - Typically Java, Scala, Groovy, and Kotlin
    - All are compatible with Java API framework, Java libraries, and Java Virtual Machine.
    - In 2017, Google announced Kotlin as official language for Android Development



System Apps

Dialer | Email | Calendar | Camera | . . .

Java API Framework

Content Providers

View System

Managers

Activity | Location | Package | Notification

Resource | Telephony | Window

Native C/C++ Libraries

Webkit | OpenMAX AL | Libc

Media Framework | OpenGL ES | . . .

Android Runtime

Android Runtime (ART)

Core Libraries

Hardware Abstraction Layer (HAL)

Audio | Bluetooth | Camera | Sensors | . . .

Linux Kernel

Drivers

Audio | Binder (IPC) | Display

Keypad | Bluetooth | Camera

Shared Memory | USB | WIFI

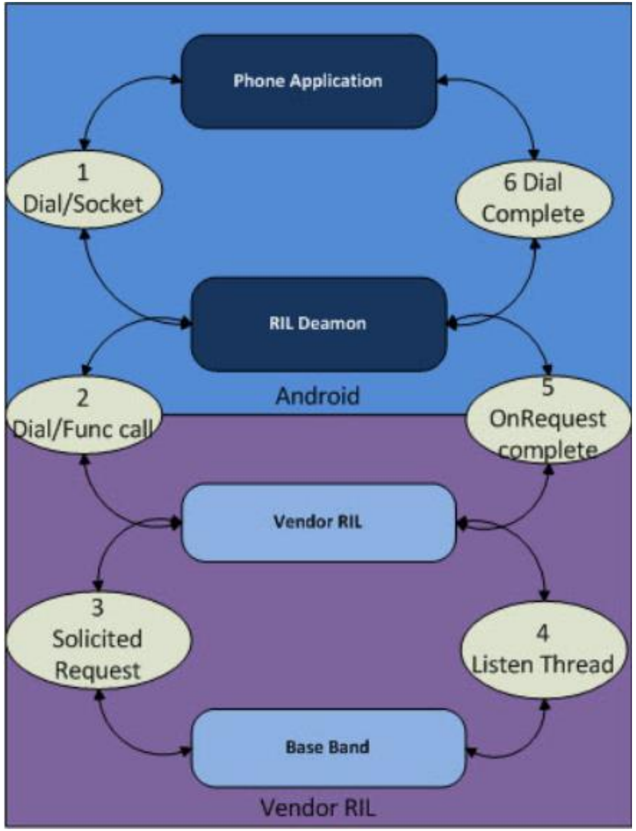
Power Management

RIL Layer

- The RIL layer is an abstraction layer that takes commands from telephony stack and pushes them to module

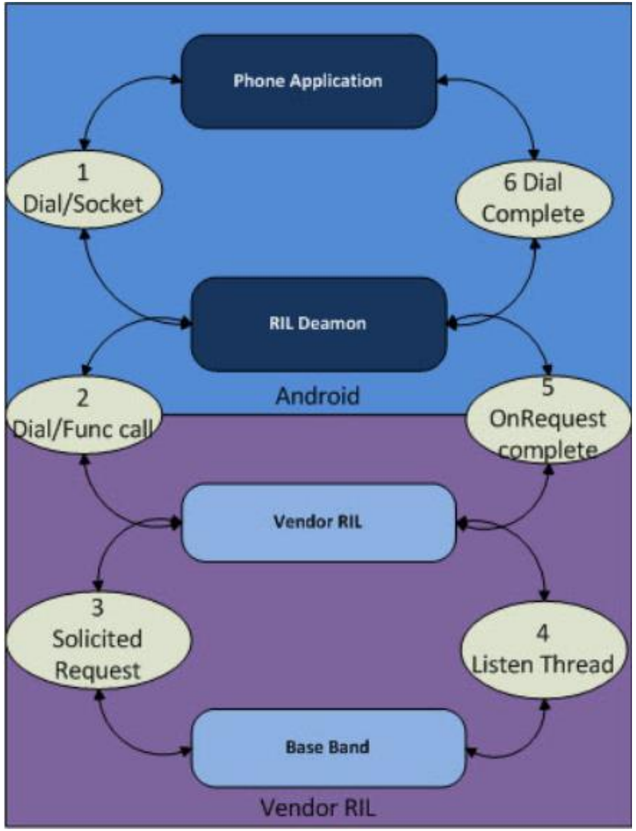
- Three parts:
  - RIL Daemon: Android facing, initializes telephony stack and vendor RIL
  - Vendor RIL: Module facing, converts standard android commands to AT commands
  - Libril: RIL libraries used by both Daemon and Vendor RIL.

- Two communication modes:
  - Solicited commands: start from telephony layer/phone application (see right)
  - Unsolicited commands: start from module (similar to starting from step 4 of the right)



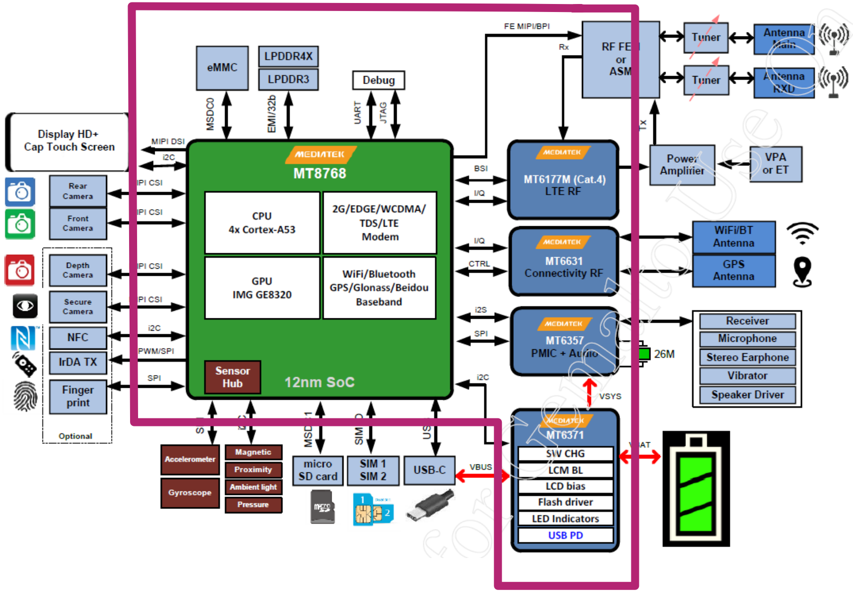
RIL Layer

- Thales provides Source code for Vendor RIL upon request
- RIL has recommended settings, but only has default values for various configurations
  - Behavior on dropped data connection
  - Behavior when no cell reception present
  - Airplane mode functionality
  - Shutdown requirements
  - Module hang recovery
- Depending on how Android is configured, there can be multiple ways of setting all of these values
- Most configurations can be illustrated by two questions
  1) Is cellular the first/only method of communication?
  2) Does data need to be transmitted real time or can it wait?
- Changes based on custom application is expected and required



System on Module - Cinterion CL31

- The CL31 will be based on a Quad Core architecture with separate CPU, GPU, Cellular, and WiFi/BT/GNSS processing
- RIL layer is still required to communicate between the Android and RF
  - Functionally no different from a software perspective than using a module with a separate Android chip
- Android can be updated without RF update/recert including:
  - Security updates
  - New releases
  - Functionality patches



Conclusion

- Android and RIL source codes and SDK will be available on request so you are not starting from scratch
- Integrating modules with Android is mostly library and configuration changes
- Thales has a dedicated R&D teams to help debug issues
- Contact your Thales Sales and TS to discuss planning and typical cellular issues for any software or hardware issues