



L&T Technology Services





Learning Report

ANDROID FRAMEWORKS



Document History

Ver. Rel. No.	Release Date	Prepared. By	Reviewed By
1	24/10/2020	Akshata Shet	

Learning Report	2
Overview of Layered Architecture	5
Overview of Protocol Stack	5
Middleware Stack	5
Layers Architecture Pattern	5
Android layered architecture	5
Overview of the Android Linux Kernel	6
Linux Kernel	6
primary and secondary storage mechanisms	6
Core Kernel IPC & Processing Mechanisms	6
Android linux Extensions	6
Infrastructure Middleware	7
HAL	7
Runtime execution environment	7
Core and native libraries	7
Android Common Service and Apps	7
Android's activity framework	7
service framework and package apps	7

MODEL-1

INTRODUCTION

Overview of Layered Architecture

Overview of Protocol Stack

Protocol Stack enables end to end communication by specifying how data should be packetized, addressed, transmitted, routed & received. Individual protocols within a suite are often designed with a single purpose in mind. This modularization simplifies design and evaluation.

Datalink and Physical layer: Handles interaction with the hardware.

Transport and Network layer: Exchanges packet across hosts & rout.

Presentation and Session layer: Implements & interacts with application.

Application layer: An application layer is an abstraction layer that specifies the shared communications protocols and interface methods used by hosts in a communications network.

Middleware Stack

Middleware provides services beyond the operating system and protocol stacks to enable components in a distributed system to communicate and manage data.

Examples: Includes operating systems like Linux and Windows. As well as various host infrastructure middleware, such as Java virtual machine, and the adaptive communication environment or ACE.

Layers Architecture Pattern

An architectural pattern is a structural organization schema for software systems that provides a set of predefined subsystems, specifies their responsibilities, and includes the rules and guidelines for organizing the relationships between these roles.

Android layered architecture

Android's architecture is structured in accordance with multiple layers.

Example: The Android Linux kernel layer is used to control hardware and manage various system resources such as the network, file system, threads and processes, cores, and so on. In a complex system like Android, layering is applied for several reasons.

- Helps to enhance systematic software reuse.
- libc provides a common API for accessing OS kernel capabilities.
- Use of virtual machines.
- Enable plug and play replacement of certain layer implementation.
- Reduce complexity of APIs.
- Enables use of popular protocols, APIs, programming languages.

Overview of the Android Linux Kernel

Linux Kernel

primary and secondary storage mechanisms

primary: RAM

Secondary: flash

Android linux executes in kernel space RAM, all android apps execute in user space RAM

Core Kernel IPC & Processing Mechanisms

Local and remote IPC-Local and remote IPC mechanisms are used to mediate interactions between apps and system services.

Example: TCP-IP model

Android linux kernel processes and Threads:Processes and threads are used to encapsulate app instructions and data efficiently, reliably and securely on one or more processor cores.

Android linux Extensions

- Memory management: Android platform tries to use all of the available memory at all times. For example, the system keeps apps in memory after they've been closed so the user can quickly switch back to them.
- Anonymous shared memory
- Power management and local IPC: Local and remote IPC mechanisms are used to mediate interactions between apps and system services.

Example: applications running on a phone or a tablet are typically used to access resources that reside across a TCPIP network. Running in cloud services in cloud environments.

Infrastructure Middleware

HAL

HAL helps to separate concerns in the Android system architecture. In particular, it separates the Android platform logic from the hardware logic. The apps and system services use the HAL to interact with the kernel device drivers on their behalf.

Runtime execution environment

Dalvik virtual machine or DVM was used to interpret so-called Dalvik bytecode.

Core and native libraries

One of the key classes provided as part of the core Java libraries in Android is the Java thread class.

- OpenGL -2D or 3D
- Vector graphics is often used for gaming apps on Android
- SQLite is a relational database engine that performs so-called CRUD or create, read, update and delete operations,
- Secure socket layer or SSL is used to ensure confidentiality and
- Integrity for web interactions.

Android Common Service and Apps

Android's activity framework

-Initiate the call backs: onCreate(), onStart(), onStop(), onDestroy() and so on.

For example a GUI component can register to listen for events. And when a user clicks a button, its onClick() method is automatically called

service framework and package apps

Android's Application Framework layer contains system services that provide
For example these services expose hardware and Linux OS kernel capabilities to apps.

Android's apps are largely written in Java AOSP, release, which you can download at the link at the bottom of the slide. Other popular apps, however, are not available in

open-source form, and these include things like Gmail, Navigation, YouTube, Google Wallet, Google Docs, and so on. Which are also very commonly used on Android devices, either tablets, smartphones or, even in some cases, wearables.