Lecture 3 – Androind and iOS Architecture

Mr. Yousif Garabet Arshak
Computer Science Department
University of Zakho
yousif.arshak@uoz.edu.krd

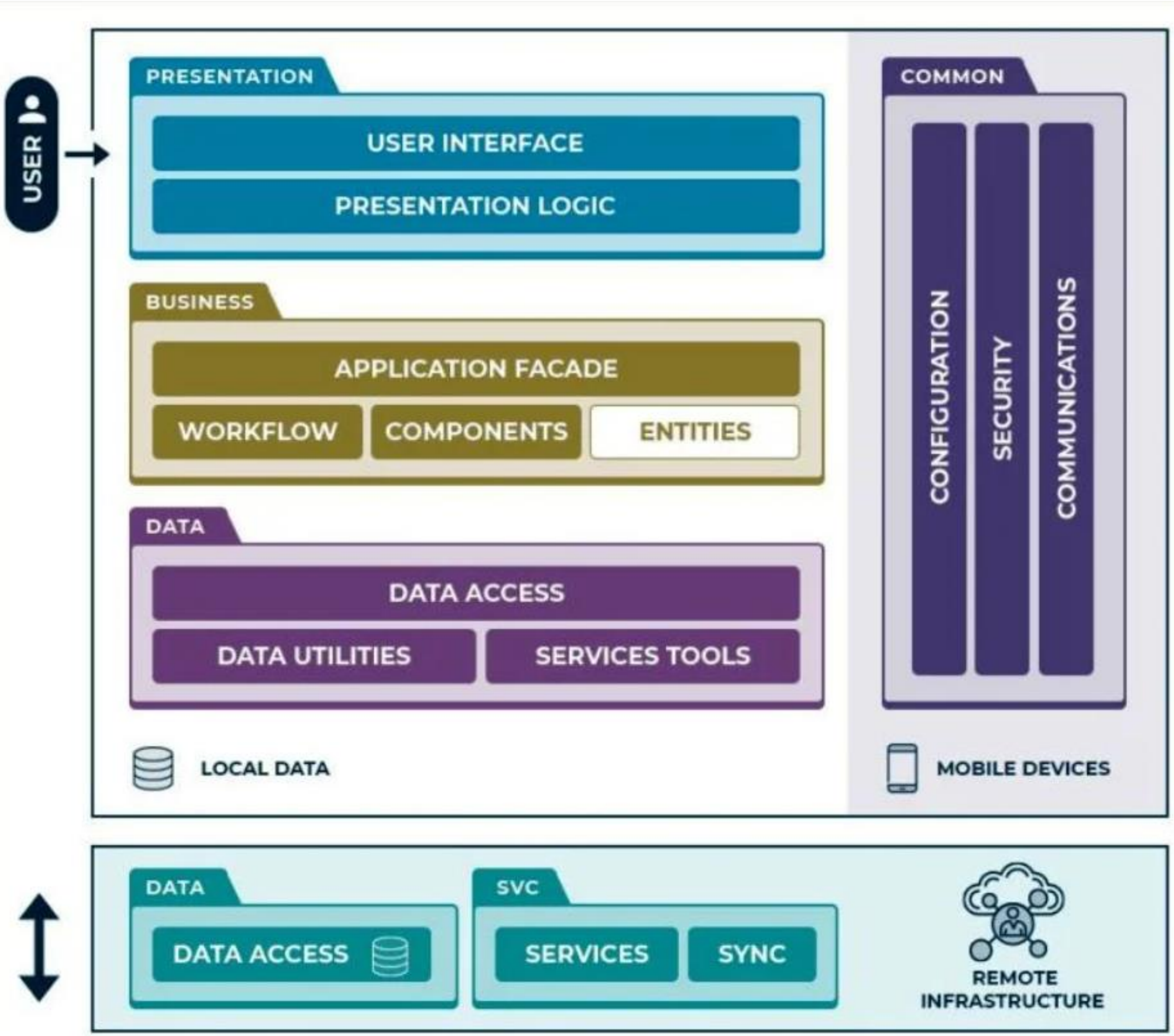
Outlines

- What is Mobile Application Architecture?
- Need to Have a Mobile App Architecture
- How does iOS differ from Android in its architecture?
- Selecting A Perfect Mobile App Architecture: Android vs. iOS

What is Mobile Application Architecture?

- Mobile application architecture is a set of techniques and patterns that are required to develop completely functional mobile app with industry standards and vendor requirements.
- Mobile application architecture design generally consists of multiple layers, including:
  - **Presentation Layer –** This layer has UI components as well as the components processing them.
  - **Business Layer –** This layer composed of business entities, business workflow and components.
  - **Data layer –** This layer contains data access components, data utilities, and service agents.

Mobile Application Architecture



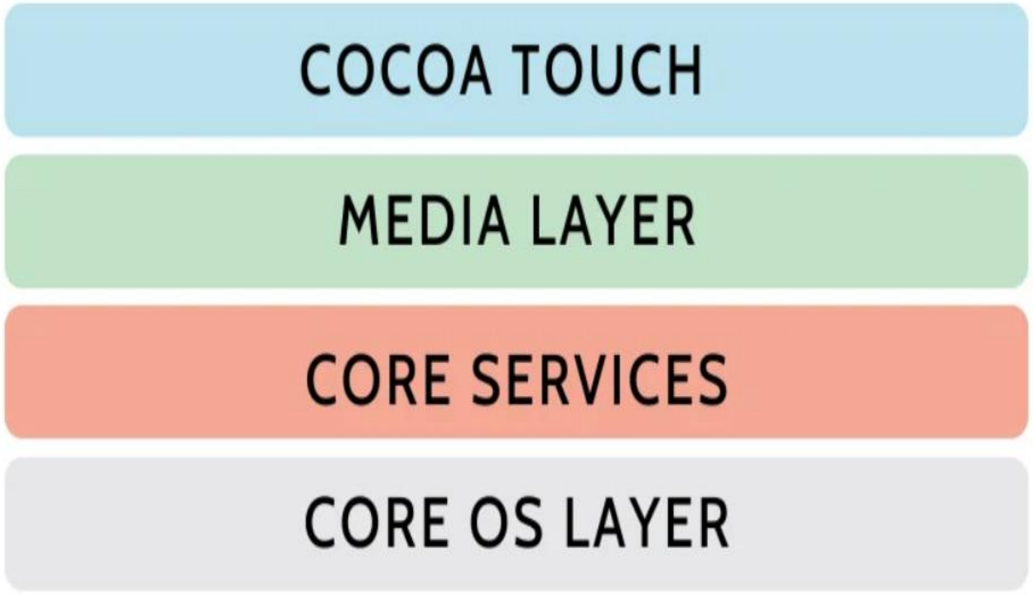
Need to Have a Mobile App Architecture

- Global app store consumer spend is estimated to reach $120 billion by the end of 2019, according to mobile data and analytics provider App Annie.
  - Mobile Gaming Goes Global
  - Mobile Is Non-Negotiable for Gen Z
  - Video Is King
  - Monetization Impact
- Gaming apps account for almost 74% of the global consumer spending while non-gaming apps account for the rest. Android and iOS being the two most prevalent OS, account for very different ecosystems and app architecture.

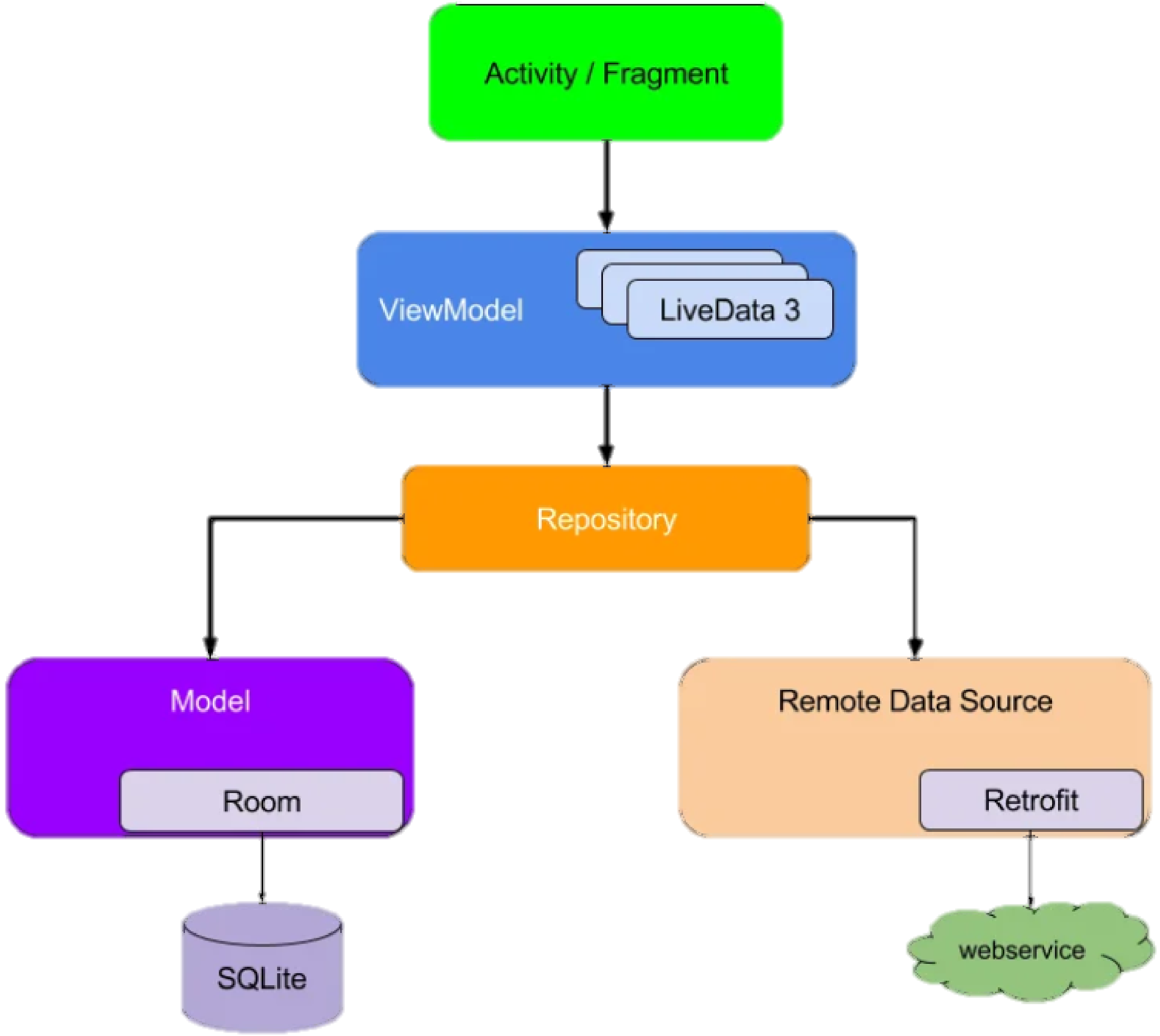
- Android and iOS development are difficult in terms of demographics (age, profession, location of users), development process, market, audience, tools used, media engagement diversity (contrary to popular belief, iOS users engage with all types of contents and media more than Android users do).
- An increasing number of people are involved in m-commerce (mobile shopping). Besides, implementation of most prevalent monetization strategies like paid apps, ads and IAP’s, iOS has a clear advantage over Android audience. Programming languages, development tools and cost of development are additional factors that give a certain advantage to a particular platform.

How does iOS differ from Android in its architecture

- Both iOS and Android architecture are similar in principle but differ in execution. Respective architecture clarifies how their apps function. Largely, Android architecture is perceived to be open as compared to iOS. Android adopts a Linux kernel, whereas iOS opts a BSD-derived kernel called Darwin. Both Android, as well as iOS, are Unix based, start with a kernel, controlling hardware at its core, along with timing, file system, drivers, interrupts and power management.



iOS Architecture



Android Architecture

- iOS architecture seems more customized as compared to Android architecture as it is programmed for security. Apple creates all the libraries from scratch and makes use of personalized BSD based kernel. On the other hand, Android users a lot of open-source software. As widely perceived it might not be as useful for the mobile phones as for the laptops with every part in it built by the same manufacturer or particularly useful while building your computer, making use of the shelf parts and using a custom case to put it in.

Memory Management

Android

- Android applications usually are limited to 16 MB of heap.
- All Android apps are written in Java. Java, unlike other programming languages, does not require any coding to allocate or de-allocate memory automatic garbage collection the garbage collector
- However, the garbage collector can sometimes cause performance issues if memory allocation is not handled carefully. The Android SDK provides allocation tracker, a tool to avoid the frequent garbage collection.

IOS

- iPhone has no garbage collection, developer has to clean up the variables after use them, otherwise the program will leak memory. Though NSObject class has accounting stuff help to keeps the track of how many other objects are currently using the object,
- When create or copy an object, its retain count is 1. Thereafter other objects may express an ownership interest in your object, which increments it’s retain count. The owners of an object may also relinquish their ownership interest in it, which decrements the retain count.
- When the retain count becomes zero, the object is deallocated (destroyed).

Power Management

Android

- CPU shouldn't consume power if no applications or services require power. Android requires that applications and services request CPU resources with "wake locks" through the Android application framework and native Linux libraries.
- If there are no active wake locks, Android will shut down the CPU. The Android Framework exposes power management to services and applications through the PowerManager class. User space native libraries should never call into Android Power Management directly. Bypassing the power management policy in the Android runtime will destabilize the system.
- All calls into Power Management should go through the Android runtime PowerManager APIs.

IOS

- iPhone does not have the power management toolkit as Mac OS does. Instead, this function is embedded into the core layer,
- which intelligently powers up planes of devices as the system goes into standby or to sleep. The most power hungry systems in iPhone from
- When put an iPhone into sleep, it will disconnect from network, turn off the Wi-Fi and screen light

Selecting A Perfect Mobile App Architecture: Android vs. iOS

- Both Android, as well as iOS operating systems, offer respective advantages and disadvantages, which forces developers to narrow down their focus towards the target market. As app become increasingly complex, the development teams are super charged to deliver faster results in the face of changing requirements and increasing competition. This calls for the requirement to have good mobile application architecture practices in projects. As the decision to adopt a particular architecture also depends upon organizational and functional requirements, follow our developers in case you wish to create a mobile application with trending architecture approach!

Any Questions?