Android Operating System Principles

Tushar B. Kute,
http://tusharkute.com

Android

- Android OS is a Linux-based mobile operating system that primarily runs on smartphones and tablets.
- The Android platform includes an operating system based upon the Linux kernel, a GUI, a web browser and end-user applications that can be downloaded.
- Although the initial demonstrations of Android featured a generic QWERTY smartphone and large VGA screen, the operating system was written to run on relatively inexpensive handsets with conventional numeric keypads.

Android: What it does?

- Manages the device's resources, like memory and storage.
- Provides a user interface (UI) for interacting with the device, including the home screen, apps, and settings.
- Enables you to download and run apps from the Google Play Store or other sources.
- Connects to the internet and other networks.
- Handles basic functions like phone calls, text messages, and multimedia playback.

Android: Features

- Open-source: The basic code is freely available, allowing manufacturers to customize it for their devices. This leads to a wide variety of Android devices with different features and user interfaces.
- Touchscreen-based: Designed for intuitive interaction using gestures like tapping, swiping, and pinching.
- Large app ecosystem: Google Play Store offers millions of apps for various purposes, from productivity and entertainment to games and social media.
- Regular updates: Google releases new versions of Android with new features and security improvements.
- Customization: You can personalize your device with different launchers, widgets, and themes.

Android: Popularity

- Android is the world's most popular mobile operating system, with over 3 billion monthly active users.
- It powers a wide range of devices, from budget-friendly smartphones to high-end flagships.

Android: Version

- As of today, February 26, 2024, the latest version is Android 14, released on October 4, 2023.

Android: History

- Early Days (2003-2005):
  - Founded in 2003, Android Inc. initially aimed to develop an operating system for digital cameras.
  - In 2004, the focus shifted to smartphones, recognizing their growing potential.
  - Google acquired Android Inc. in 2005, injecting resources and vision.

Android: History

- Building the Foundation (2005-2008):
  - The Open Handset Alliance (OHA), led by Google, formed in 2005, fostering an open-source approach for Android development.
  - The operating system was built on a modified Linux kernel, ensuring stability and flexibility.
  - Internal releases and beta versions were tested and refined within Google and the OHA.

Android: History

Android: History

Android: History

- The First Steps (2008-2010):
  - September 2008: Android 1.0 officially launched, marking the beginning of the commercial journey.
  - October 2008: The HTC Dream (T-Mobile G1) became the first commercially available Android device, featuring a touch screen and slide-out keyboard.
  - Early versions focused on core functionalities like a web browser, email, maps, and a basic app marketplace.

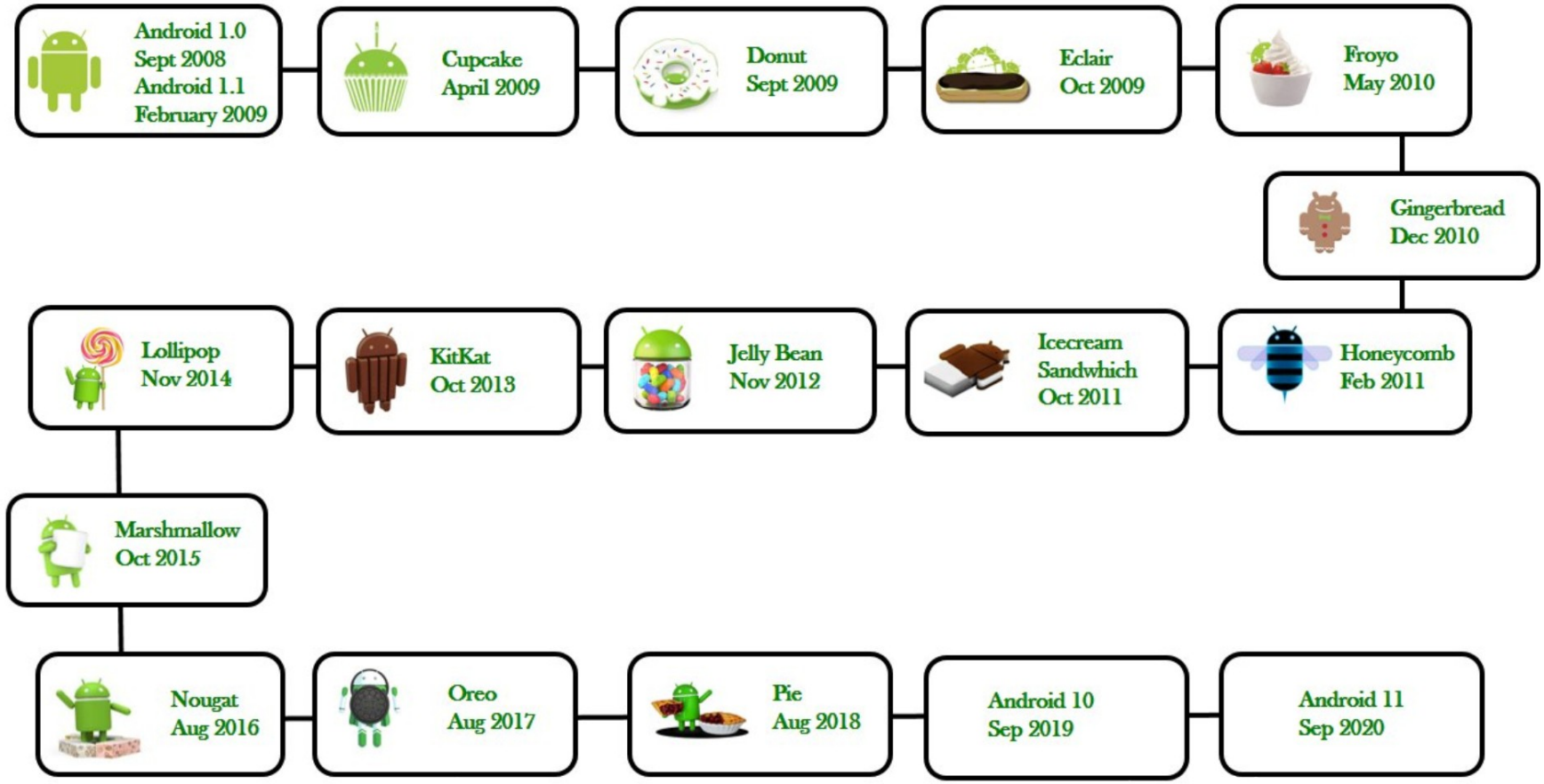
Android: History

- Rapid Evolution and Growth (2010-2014):
  - Android 2.0 (Froyo) introduced multi-touch support, enhancing user experience.
  - Android 3.0 (Honeycomb) optimized for tablets, marking a shift towards diversification.
  - Android 4.0 (Ice Cream Sandwich) unified the phone and tablet interfaces, streamlining the ecosystem.
  - Samsung, LG, and HTC emerged as major Android device manufacturers, increasing competition and innovation.
  - The Google Play Store grew exponentially, offering a vast array of apps and games.

Android: History

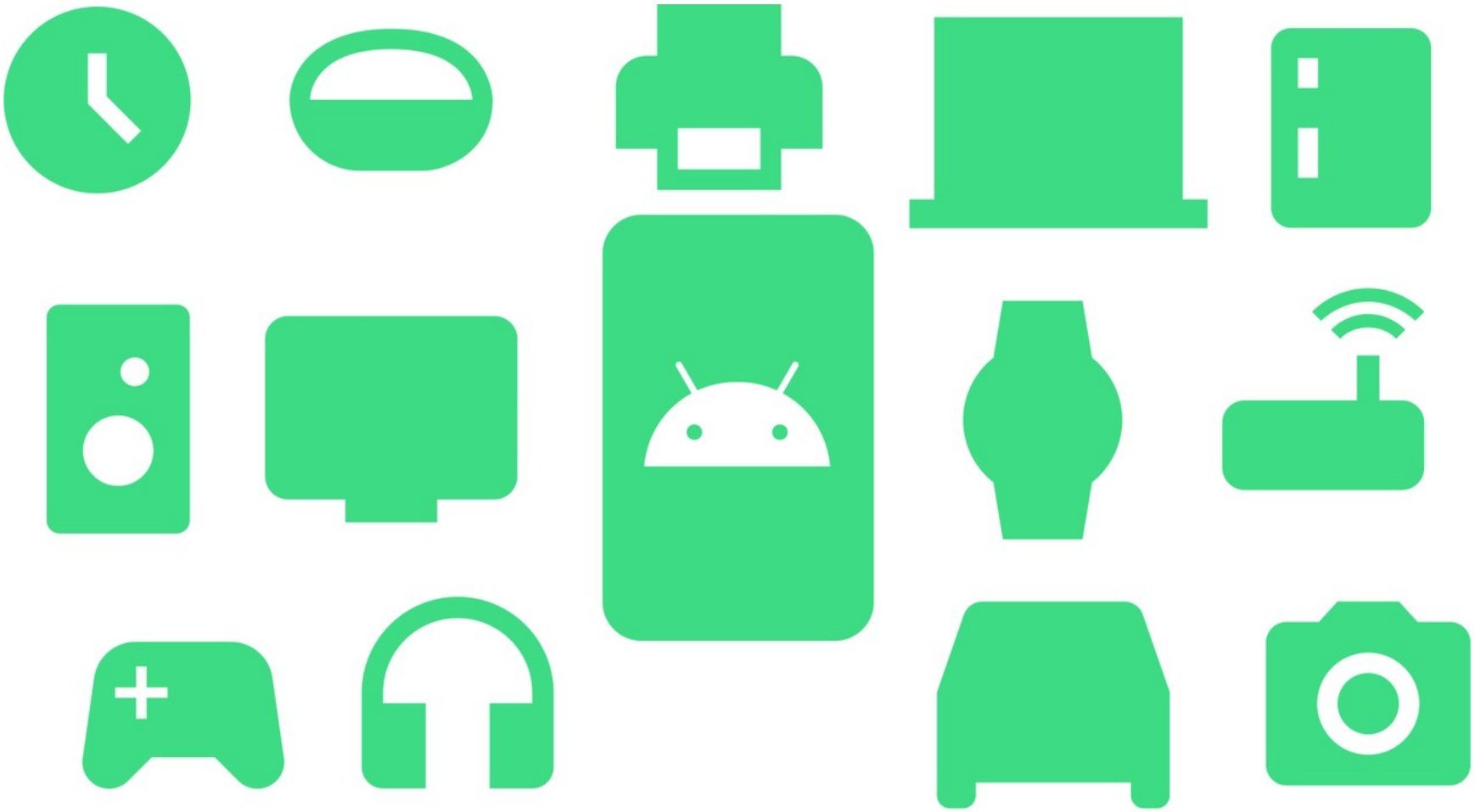
- Maturity and Refinement (2014-Present):
  - Android 5.0 (Lollipop) brought a significant design overhaul, Material Design, with a cleaner and more intuitive interface.
  - Google Pixel devices launched in 2016, showcasing Google's vision for the platform and driving hardware advancements.
  - Android versions focused on performance improvements, security enhancements, and AI integration.
  - Android 10 introduced system-wide dark mode and gesture navigation.
  - Android 12 and 13 further refine the user experience with enhanced privacy features, customization options, and improved multitasking.

Android: History



Android: History

Where do we use Android?



Hardware Requirements

- Android uses ARM for its hardware platform; later versions of Android OS support x86 and x86-64 architectures.
- Starting in 2012, device manufacturers released Android smartphones and tablets with Intel processors.
- The minimum hardware requirements of Android depend on the device's screen size and CPU type and density.
- Originally, Google required a 200 MHz processor, 32 MB of storage and 32 MB of RAM.

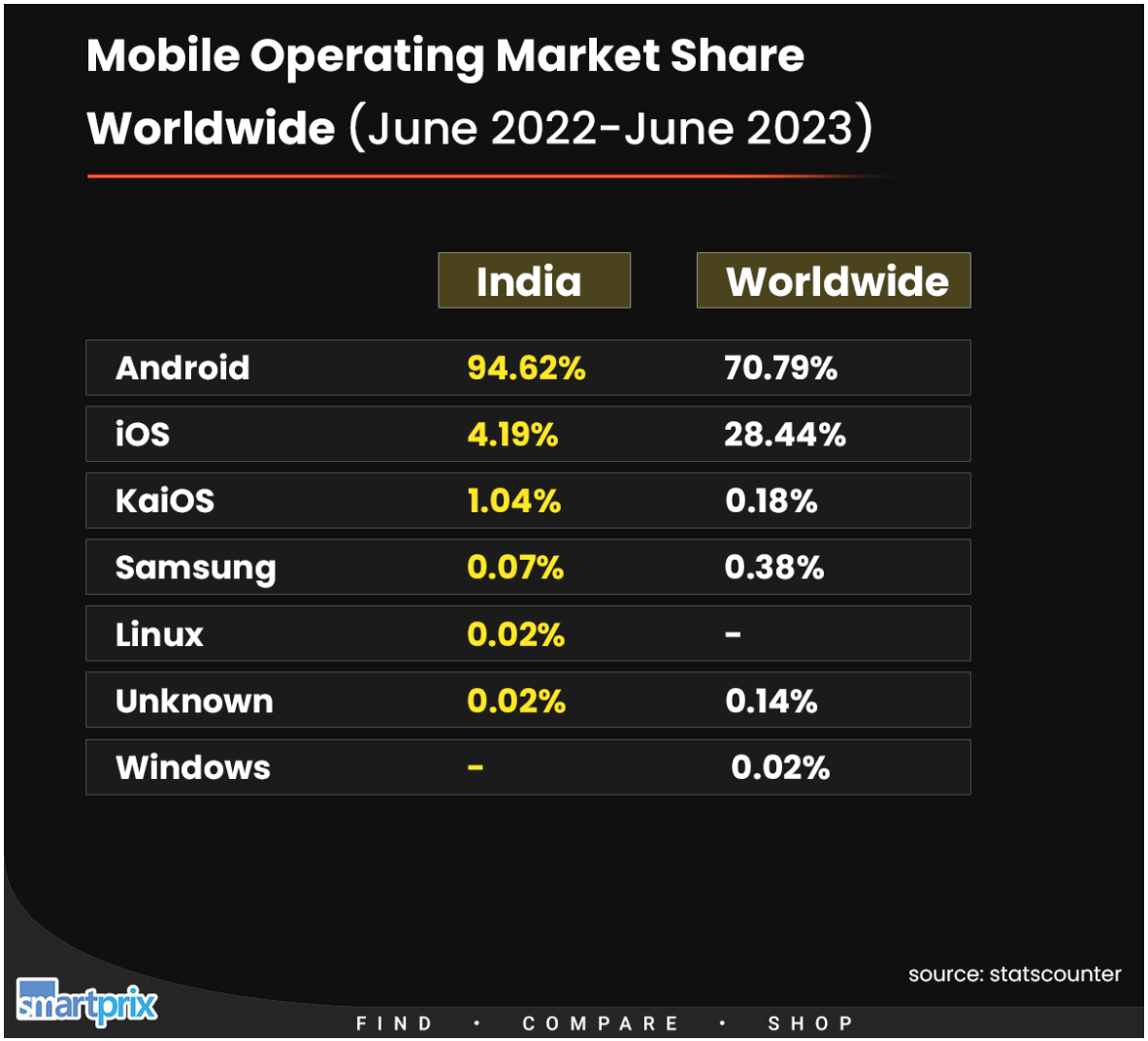
Features of Android

Feature	Description
Beautiful UI	Android OS basic screen provides a beautiful and intuitive user interface.
Connectivity	GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wi-Fi, LTE, NFC and WiMAX.
Storage	SQLite, a lightweight relational database, is used for data storage purposes.
Media support	H.263, H.264, MPEG-4 SP, AMR, AMR-WB, AAC, HE-AAC, AAC 5.1, MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF, and BMP
Messaging	SMS and MMS

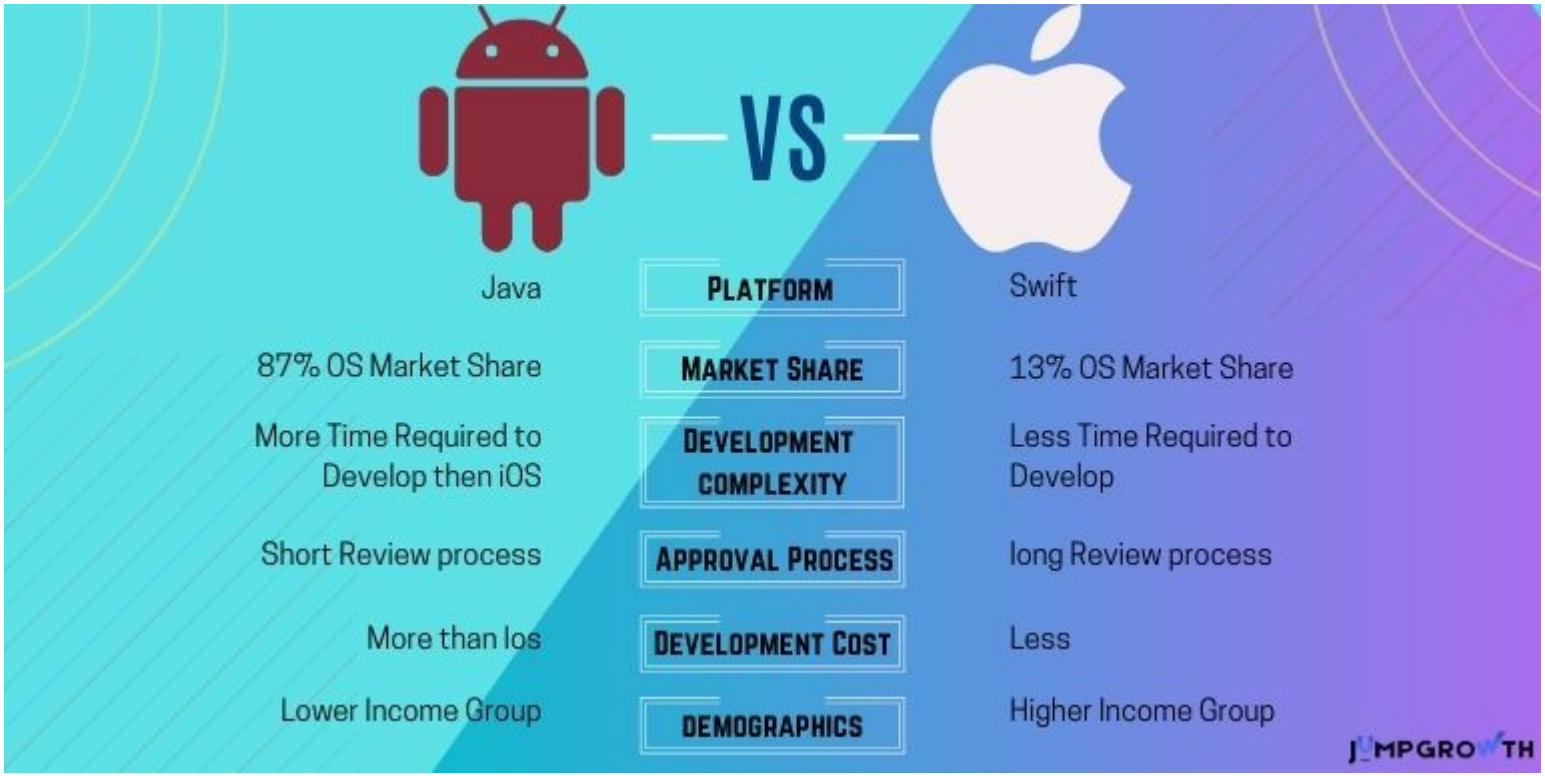
Features of Android

Web browser	Based on the open-source WebKit layout engine, coupled with Chrome's V8 JavaScript engine supporting HTML5 and CSS3.
Multi-touch	Android has native support for multi-touch which was initially made available in handsets such as the HTC Hero.
Multi-tasking	User can jump from one task to another and same time various application can run simultaneously.
Resizable widgets	Widgets are resizable, so users can expand them to show more content or shrink them to save space
Multi-Language	Supports single direction and bi-directional text.
GCM	Google Cloud Messaging (GCM) is a service that lets developers send short message data to their users on Android devices, without needing a proprietary sync solution.
Wi-Fi Direct	A technology that lets apps discover and pair directly, over a high-bandwidth peer-to-peer connection.
Android Beam	A popular NFC-based technology that lets users instantly share, just by touching two NFC-enabled phones together.

Android Market Share



Comparison


VS
Java
PLATFORM
Swift
87% OS Market Share
MARKET SHARE
13% OS Market Share
More Time Required to Develop then iOS
DEVELOPMENT COMPLEXITY
Less Time Required to Develop
Short Review process
APPROVAL PROCESS
long Review process
More than Ios
DEVELOPMENT COST
Less
Lower Income Group
DEMOGRAPHICS
Higher Income Group
JUMPGROWTH

Applications

- Traditional Devices:
  - Smartphones and Tablets: The most well-known application, powering billions of devices globally for communication, entertainment, productivity, and more.
  - Smart TVs and Streaming Devices: Android TV brings the Android experience to your living room, offering access to apps, games, and streaming services.
  - Smartwatches and Wearables: Wear OS by Google runs on many smartwatches and fitness trackers, enabling notifications, health tracking, and other functionalities.

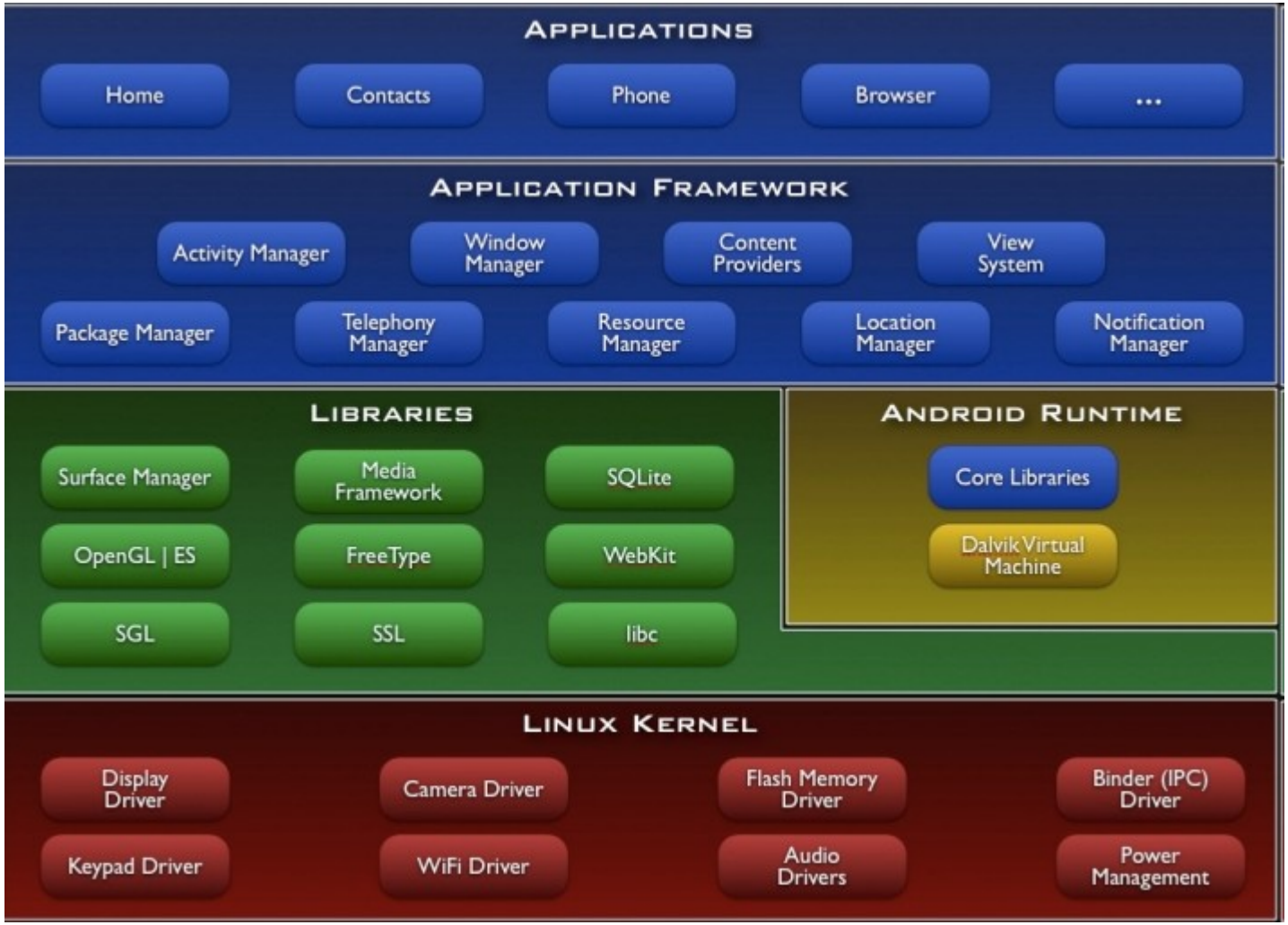
Applications

- Emerging Technologies:
  - Smart Homes and Appliances: Android Things is a lightweight version for embedded devices, controlling lights, thermostats, and other smart home components.
  - Cars and Automotive: Android Automotive OS (formerly Android Auto) integrates seamlessly with car infotainment systems, providing navigation, communication, and entertainment features.
  - Virtual and Augmented Reality: Android powers headsets like Google Daydream, offering immersive VR and AR experiences.

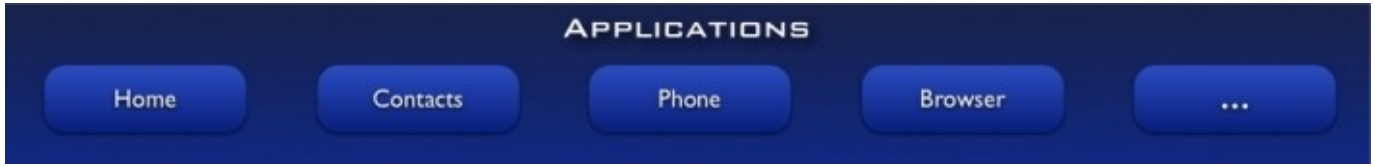
Applications

- Beyond Consumer Devices:
  - Medical Devices and Healthcare: Android is being used in medical devices for patient monitoring, data collection, and even surgical assistance.
  - Industrial Automation and Robotics: Android's reliability and customization make it suitable for controlling robots, industrial machines, and other automation systems.
  - Point-of-Sale Systems and Kiosks: Android tablets are increasingly used in retail stores and restaurants for taking orders, processing payments, and displaying information.

Android Architecture



Android Software Stack



- Android provides a set of core applications:
  - Email Client
  - SMS Program
  - Calendar
  - Maps
  - Browser
  - Contacts
  - Etc
- All applications are written using the Java language.

Android Application Framework

- Enabling and simplifying the reuse of components
  - Developers have full access to the same framework APIs used by the core applications.
  - Users are allowed to replace components.

Android Application Framework

Feature	Role
View System	Used to build an application, including lists, grids, text boxes, buttons, and embedded web browser
Content Provider	Enabling applications to access data from other applications or to share their own data
Resource Manager	Providing access to non-code resources (localized strings, graphics, and layout files)
Notification Manager	Enabling all applications to display customer alerts in the status bar
Activity Manager	Managing the lifecycle of applications and providing a common navigation backstack

Android Software Stack-Libraries

- Including a set of C/C++ libraries used by components of the Android system
- Exposed to developers through the Android application framework.

Android Software Stack-Runtime

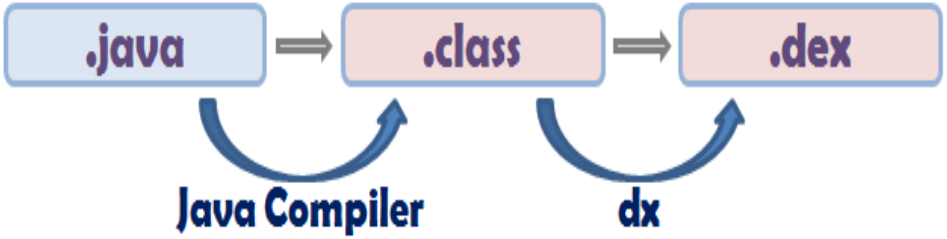
- Core Libraries
  - Providing most of the functionality available in the core libraries of the Java language
  - APIs
    - Data Structures
    - Utilities
    - File Access
    - Network Access
    - Graphics etc.

Android Software Stack-Runtime

- Dalvik Virtual Machine
  - Providing environment on which every Android application runs
  - Each Android application runs in its own process, with its own instance of the Dalvik VM.
  - Dalvik has been written such that a device can run multiple VMs efficiently.
- Register-based virtual machine

Android Software Stack-Runtime

- Dalvik Virtual Machine (Cont)
  - Executing the Dalvik Executable (.dex) format
  - .dex format is optimized for minimal memory footprint.



- Compilation
  - Relying on the Linux Kernel for:
    - Threading
    - Low-level memory management

Android Software Stack-Linux Kernel

- Relying on Linux Kernel 5.4+ for core system services
  - Memory and Process Management
  - Network Stack
  - Driver Model
  - Security
- Providing an abstraction layer between the H/W and the rest of the S/W stack

Thank you

This presentation is created using LibreOffice Impress 7.4.1.2, can be used freely as per GNU General Public License

@mitu_skillologies

@mITuSkillologies

@mitu_group

@mitu-skillologies

@MITUSkillologies

kaggle

@mituskillologies

Web Resources
https://mitu.co.in
http://tusharkute.com

@mituskillologies

contact@mitu.co.in

tushar@tusharkute.com