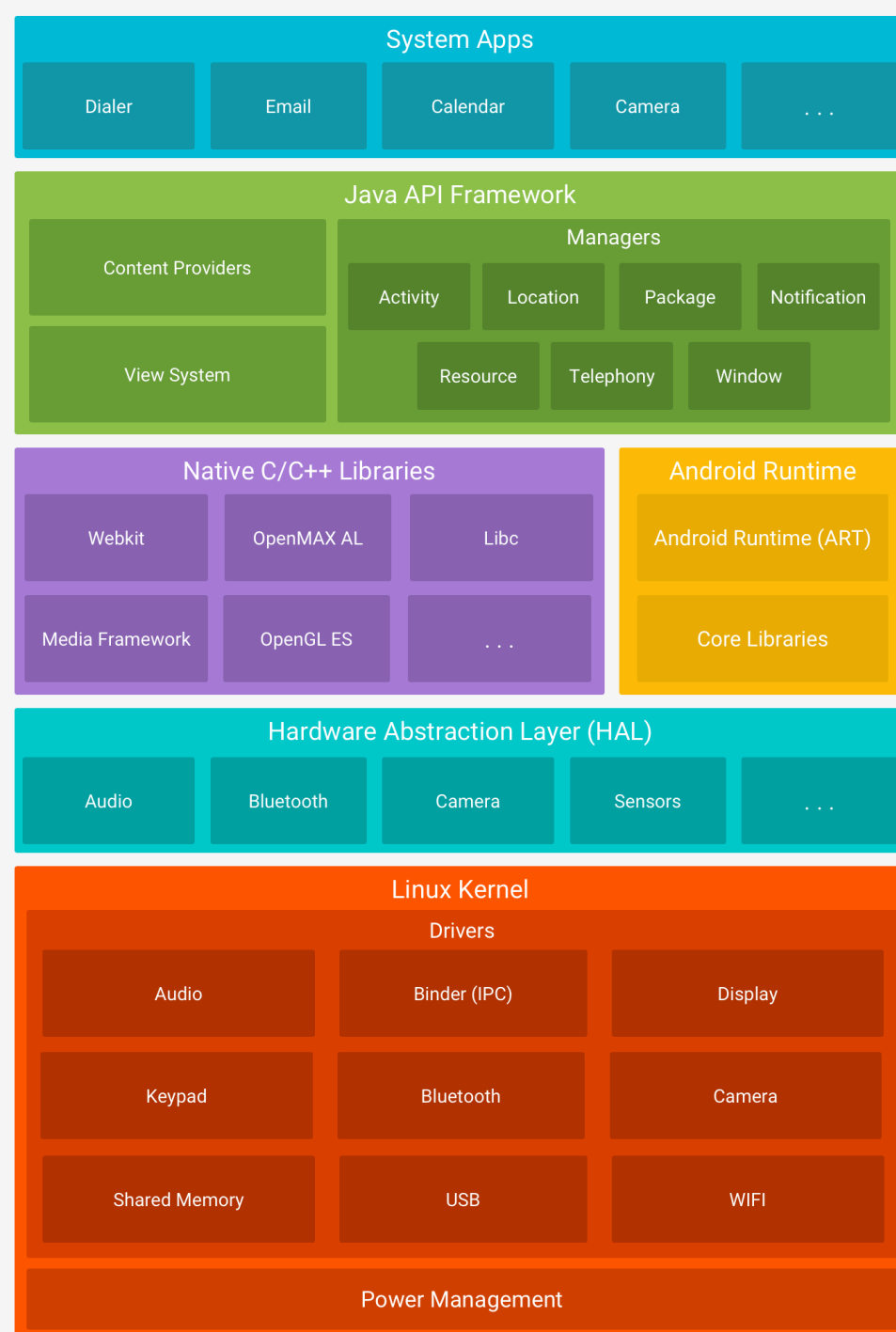


Basi Android

- Cosa è Android?
 - Android è un insieme di strumenti e librerie per sviluppare applicazioni mobili
 - è più di un SO
- Android si definisce “open”
 - Contiene tecnologie open source
 - Linux
 - Il codice di Android è open
 - Licenza Open Source Apache 2.0
 - Chi vuole usare Android non deve pagare

Componenti del SO



<https://developer.android.com/guide/platform>

Programmazione Java

Virtual Machine

- Android usa la Android RunTime (ART)
 - Non la Java Virtual Machine
 - La ART esegue bytecode compilato
 - Si compilano i .class
 - I file sono compilati al momento dell'istallazione dell'APP
 - Frammenti di codice nativo
 - Prima della V4.4 usava Dalvik
- Android non usa una Java Virtual Machine
 - Cioè in Android non viene eseguito bytecode

Tool Android

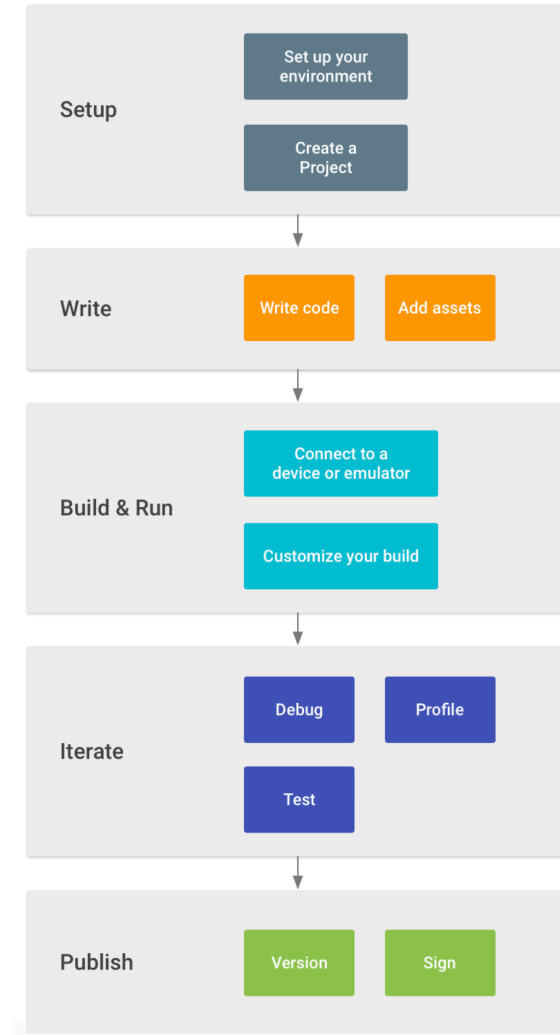
- Android Studio
 - IDE basati su IntelliJ
- Software Development KIT (SDK)
 - Tools per sviluppo applicazioni java
- Native Development Kit (NDK)
 - Sviluppo di applicazioni o parte di esse in C++

Struttura SDK






- Tools
 - Strumenti per la gestione dei progetti
 - `android`, `emulator`, etc.
- Platforms
 - Versioni di android
 - 4.x, 5.x, ... , 14 Cellulari e Tablet, ...
 - 3.2, 3.1, 3.0 Tablet
 - 2.3.3, 2.2, 2.1 Cellulari
 - Ad ogni versione corrisponde un livello di API
- Extra
 - librerie google ed altro

Developer workflow basics

- Configura l'ambiente di lavoro
 - Installa Android Studio e crea un nuovo progetto.
- Scrivi la tua app
 - Usa **Android Studio** per scrivere codice, progettare l'interfaccia e gestire risorse per diversi dispositivi.
- Compila ed esegui
 - Genera un APK eseguibile per testarlo su emulatore o dispositivo reale.
 - Personalizza la build (varianti, riduzione codice/richieste).
- Debug, profiling e test
 - Correggi bug, ottimizza le prestazioni (memoria, rete, CPU).
 - 🖐 Usa strumenti come Logcat e il profiler di Android Studio.
- Pubblica l'app
 - Crea il pacchetto finale (Android App Bundle), firmalo e pubblicalo sul Play Store.



Applicazioni Android

-  **Linguaggi:** Scritte in **Java** o **Kotlin**
-  **Formato:** Raccolte in un file **.apk** (Android Package)
-  **Sicurezza:** Ogni app opera in un ambiente **isolato** dalle altre
-  **Integrazione:** Fortemente **integrate** con il **sistema operativo**
-  **Struttura:** Composte da:
 - Componenti
 - Risorse
 - File descrittore (**AndroidManifest.xml**)

Componenti App

- **Activity** – Interfaccia utente
- **Service** – Operazioni in background
- **Content Provider** – Condivisione dati tra app
- **Broadcast Receiver** – Risposta a eventi di sistema/app

Activity

- Rappresenta una schermata dell'applicazione
- In generale una applicazione contiene varie Activity
- Quando avviamo una nuova attività la corrente è posta in background
 - ho uno stack di activity
- Le GUI si realizzano in XML o java/kotlin
- Si controlla la vita dell'activity mediante callback
 - onCreate, onPause

Service

- Servono ad eseguire operazioni in background
- Non hanno interfaccia grafica
- Servono ad operazioni come suonare la musica, gestire la rete, leggere la posizione, etc.
- Sono eseguiti indipendentemente da chi li ha creati
- Possono avere una interfaccia verso altri componenti

Content Provider

- Sono usati per memorizzare e leggere dati rendendoli accessibili alle applicazioni
- Sono l'unico modo di condividere dati fra applicazioni
- Espongono una URI pubblica per individuare univocamente i propri dati
- I dati sono esposti in forma di una singola tabella di database
- ci sono CP predefiniti per contatti, media, etc.






Broadcast Receiver

- Sono componenti che rispondono ad “annunci” inviati a tutti i componenti dal sistema
- Ex. lo schermo si spegne, la batteria e scarica, c’è una chiamata in arrivo, etc.
- Anche le app possono inviare annunci broadcast

Manifest file

- Ogni applicazione contiene il file `AndroidManifest.xml`
- Presenta l'applicazione al sistema operativo
- Descrive i componenti dell'applicazione
- Dichiarare i permessi necessari all'applicazione
- Dichiarare le API minime necessarie all'applicazione

Android Studio

- **IDE ufficiale** supportato da **Google**
- Basato su **IntelliJ IDEA**, potente e flessibile
- Include:
 -  **Editor di codice intelligente** (Java/Kotlin)
 -  **Designer grafico** per layout XML
 -  **Gestione automatica dei build** con Gradle
 -  **Strumenti di test** e debug integrati
 -  **Emulatore Android** per testare su dispositivi virtuali
- **Offre assistenza con:**
 - Autocompletamento, linting, refactoring
 - Anteprima UI in tempo reale
 - Integrazione con Git e altri VCS

Progetto Android



Creare un progetto Android

- Non basta scrivere un semplice file: serve un **ambiente di sviluppo** completo
- È necessario usare i **tool ufficiali**, come **Android Studio**



Tipologie di progetto

- Variano in base al **sistema di build** utilizzato



Sistemi di compilazione (Build System)

- **Gradle** – lo standard attuale, integrato in **Android Studio**

Struttura progetto

app/

Contiene tutto ciò che serve per costruire l'app:

- **manifests/**
 - AndroidManifest.xml
- **java/**
 - Contiene il **codice sorgente** principale
 - Sottocartelle:
 - (androidTest): test strumentali (eseguiti su dispositivo/emulatore)
 - (test): test locali (eseguiti sulla JVM)
- **res/**

Cartella delle **risorse** non di codice (layout, immagini, stringhe, temi, ecc.)

 - drawable/: immagini e forme vettoriali
 - layout/: file XML per la UI
 - mipmap/: icone dell'app
 - values/: stringhe, colori, stili, ecc.
 - xml/: configurazioni personalizzate (es. preference)

Gradle Scripts

- `build.gradle.kts` (Project) – configurazione a livello di progetto
- `build.gradle.kts` (Module :app) – configurazione specifica dell'app
- `proguard-rules.pro` – regole per l'offuscamento del codice
- `gradle.properties` – proprietà personalizzate
- `gradle-wrapper.properties` – versione di Gradle usata
- `libs.versions.toml` – catalogo delle versioni delle dipendenze
- `local.properties` – percorso locale dell'SDK Android
- `settings.gradle.kts` – impostazioni dei moduli del progetto