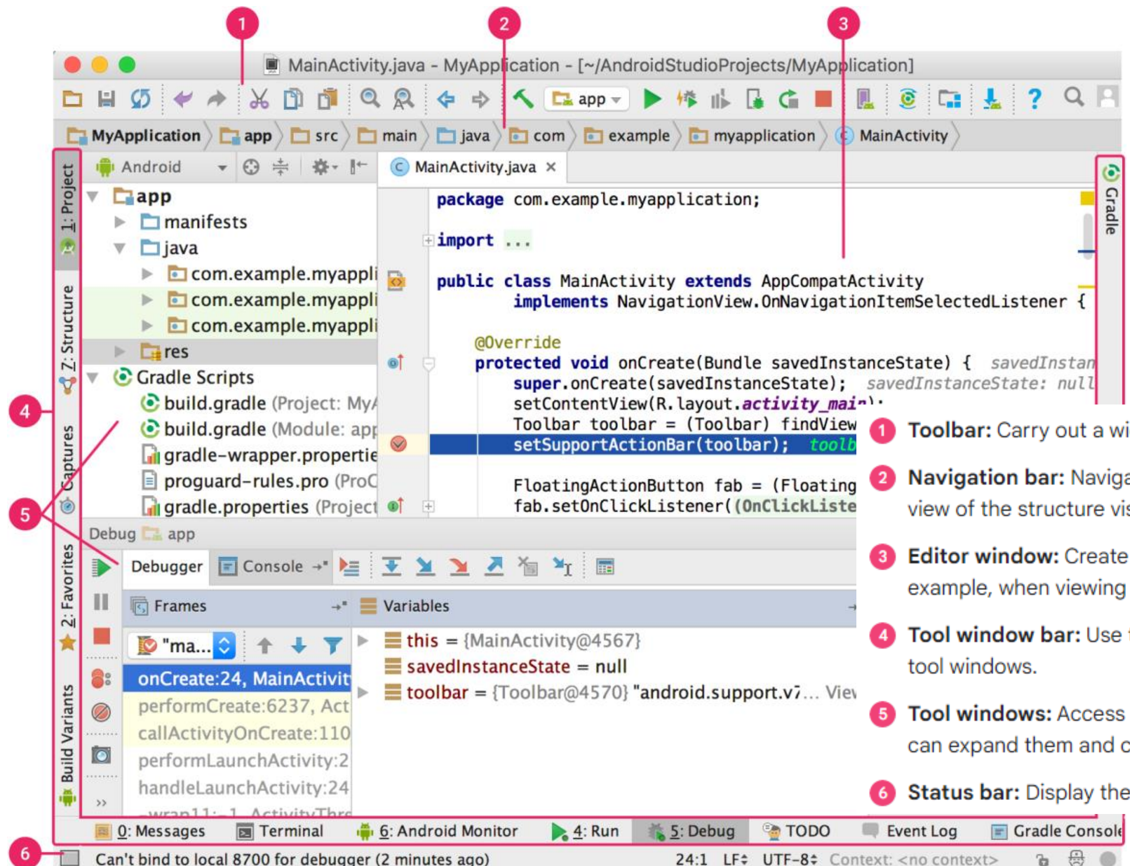
Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

Mobile Application Development

Introduction of Android Studio

U. V. Patel College of Engineering, Ganpat University

Anatomy of Android Application



```
package com.example.myapplication;

import ...

public class MainActivity extends AppCompatActivity
        implements NavigationView.OnNavigationItemSelectedListener {

    @Override
    protected void onCreate(Bundle savedInstanceState) {  savedInstan
        super.onCreate(savedInstanceState);  savedInstanceState: null
        setContentView(R.layout.activity_mai
        Toolbar toolbar = (Toolbar) findView
        setSupportActionBar(toolbar);  toolb

        FloatingActionButton fab = (Floating
        fab.setOnClickListener((OnClickListe
```

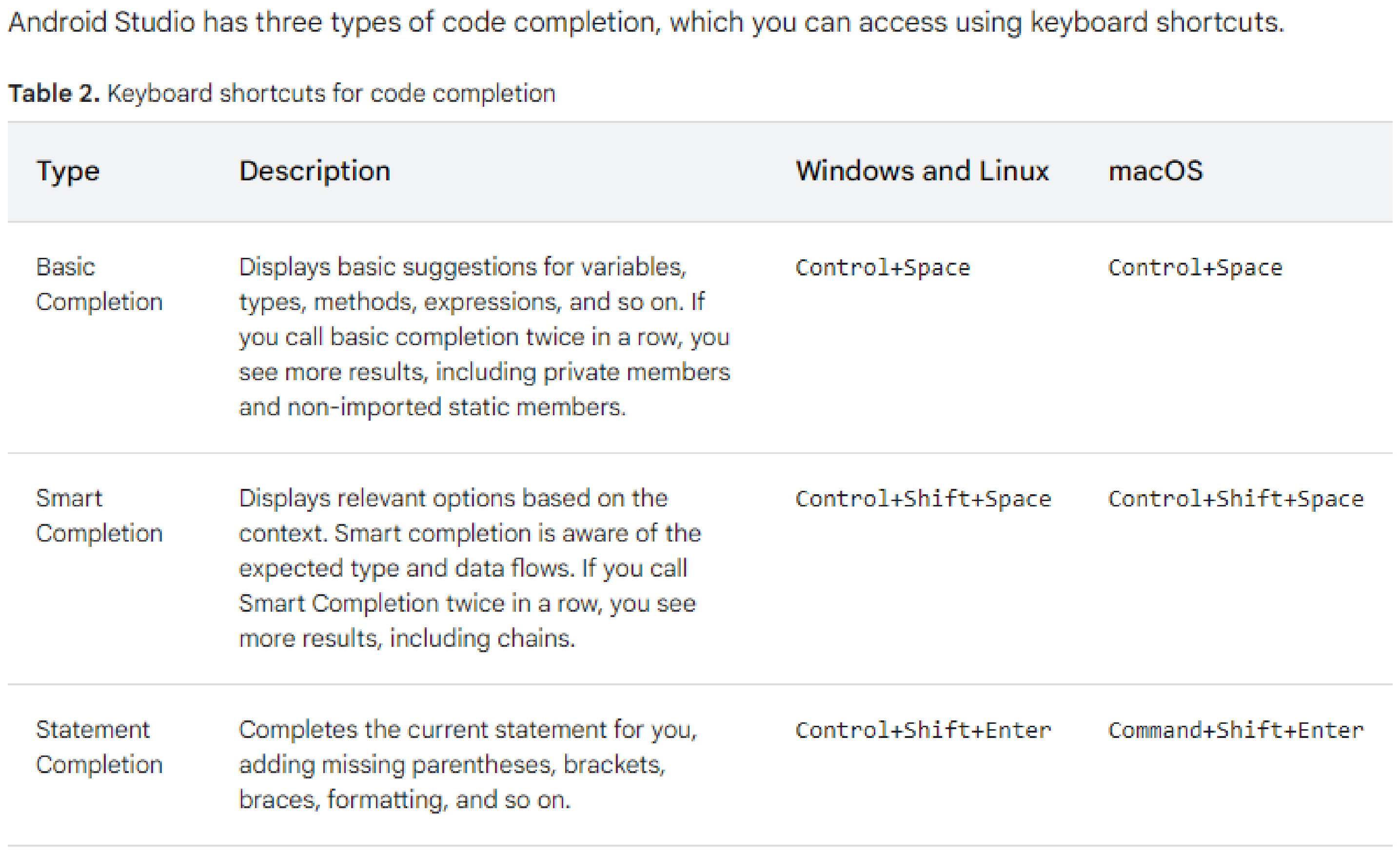
```
onCreate:24, MainActivit
performCreate:6237, Act
callActivityOnCreate:110
performLaunchActivity:2
handleLaunchActivity:24
```

```
this = {MainActivity@4567}
savedInstanceState = null
toolbar = {Toolbar@4570} "android.support.v7... Viev
```

```
Can't bind to local 8700 for debugger (2 minutes ago)    24:1  LF  UTF-8  Context: <no context>
```

1. **Toolbar:** Carry out a wide range of actions, including running your app and launching Android tools.
2. **Navigation bar:** Navigate through your project and open files for editing. It provides a more compact view of the structure visible in the **Project** window.
3. **Editor window:** Create and modify code. Depending on the current file type, the editor can change. For example, when viewing a layout file, the editor displays the Layout Editor.
4. **Tool window bar:** Use the buttons on the outside of the IDE window to expand or collapse individual tool windows.
5. **Tool windows:** Access specific tasks like project management, search, version control, and more. You can expand them and collapse them.
6. **Status bar:** Display the status of your project and the IDE itself, as well as any warnings or messages.

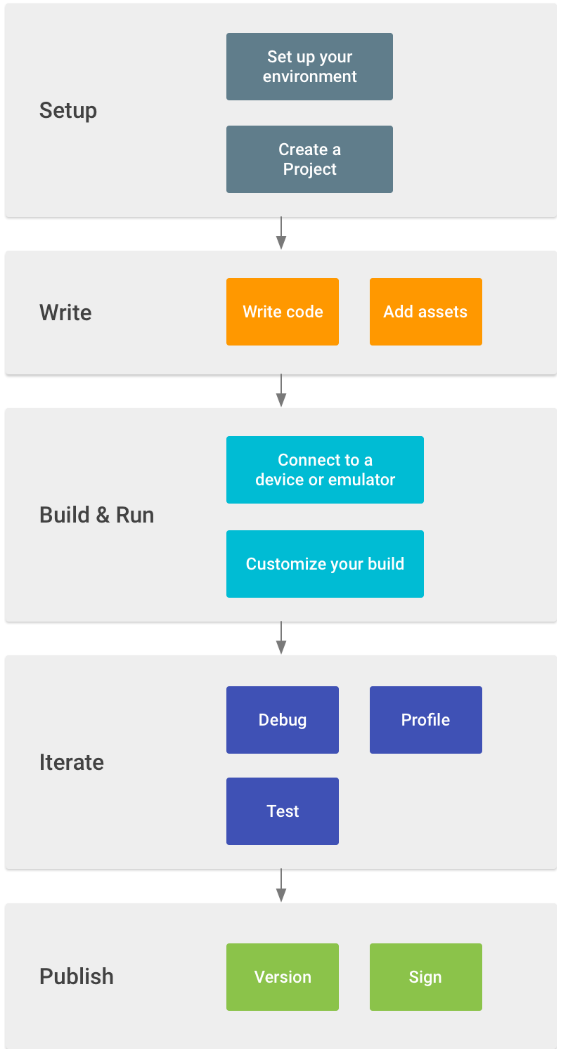
Code completion



Android Studio has three types of code completion, which you can access using keyboard shortcuts.

Table 2. Keyboard shortcuts for code completion

Type	Description	Windows and Linux	macOS
Basic Completion	Displays basic suggestions for variables, types, methods, expressions, and so on. If you call basic completion twice in a row, you see more results, including private members and non-imported static members.	`Control+Space`	`Control+Space`
Smart Completion	Displays relevant options based on the context. Smart completion is aware of the expected type and data flows. If you call Smart Completion twice in a row, you see more results, including chains.	`Control+Shift+Space`	`Control+Shift+Space`
Statement Completion	Completes the current statement for you, adding missing parentheses, brackets, braces, formatting, and so on.	`Control+Shift+Enter`	`Command+Shift+Enter`



Developer workflow

- For more information, refer: https://developer.android.com/studio/projects

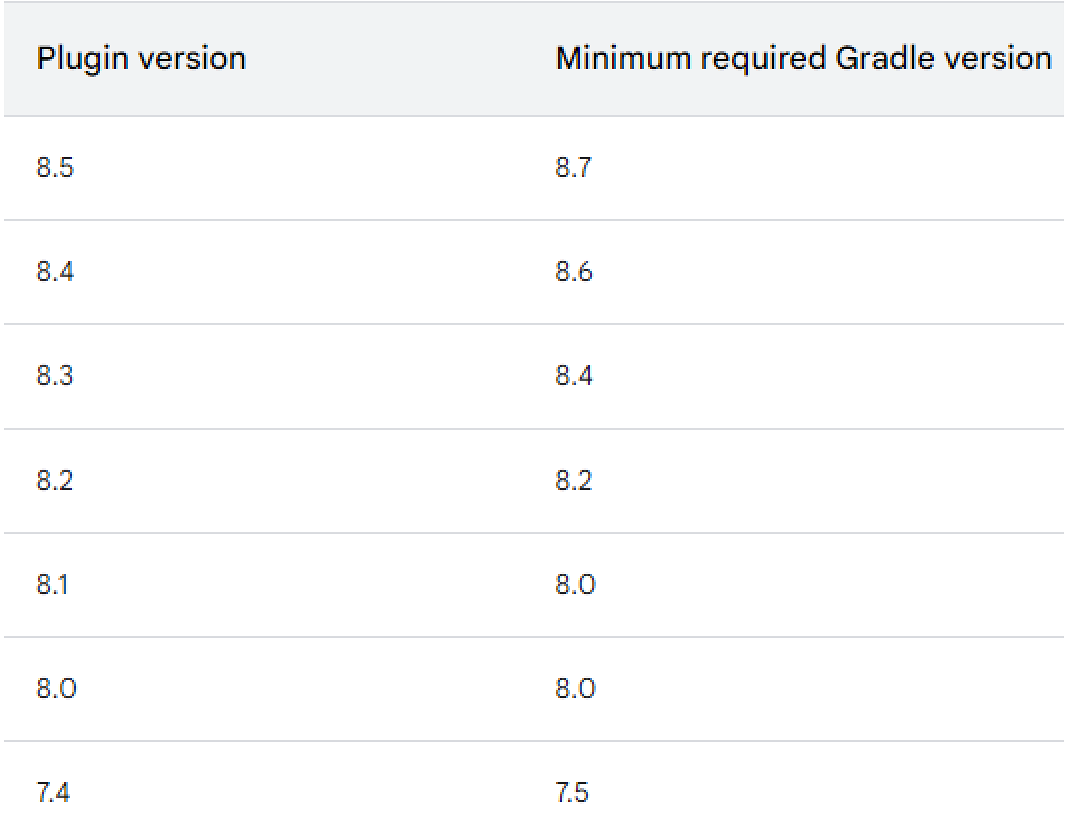
Android Gradle plugin (AGP)

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

- The Android Studio build system is based on Gradle
- The Android Gradle plugin adds several features that are specific to build Android apps.
- Although the Android Gradle plugin (AGP) is typically updated in lock-step with Android Studio
- The plugin (and the rest of the Gradle system) can run independent of Android Studio and be updated separately.
- For more information, refer: https://developer.android.com/build/releases/gradle-plugin

Gradle Version

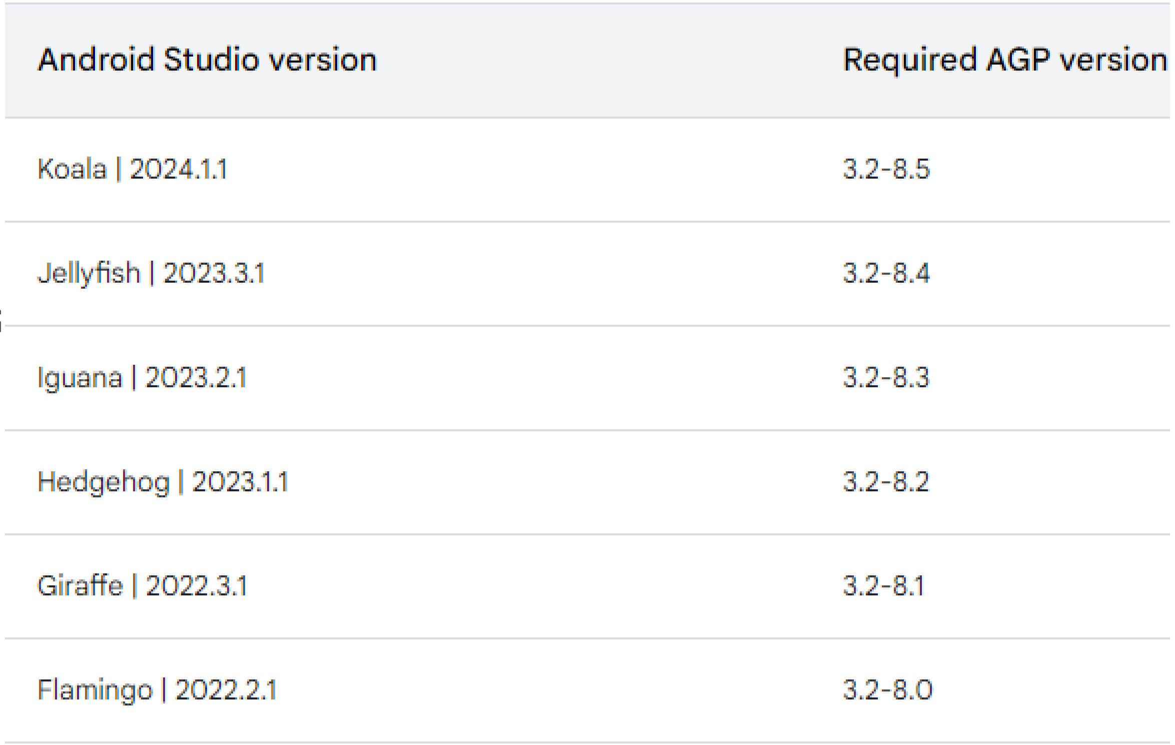
- The following table lists which version of Gradle is required for each version of the Android Gradle plugin
- For the best performance, you should use the latest possible version of both Gradle and the plugin.



Plugin version	Minimum required Gradle version
8.5	8.7
8.4	8.6
8.3	8.4
8.2	8.2
8.1	8.0
8.0	8.0
7.4	7.5

Android Gradle plugin and Android Studio compatibility

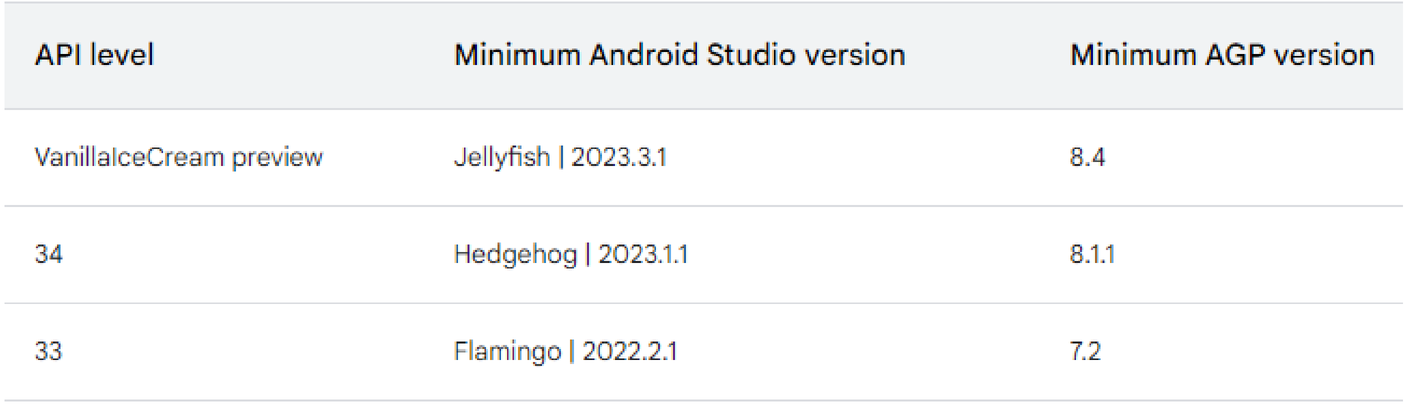
- The Android Studio build system is based on Gradle, and the Android Gradle plugin (AGP) adds several features that are specific to building Android apps.
- The following table lists which version of AGP is required for each version of Android Studio.



Android Studio version	Required AGP version
Koala | 2024.1.1	3.2-8.5
Jellyfish | 2023.3.1	3.2-8.4
Iguana | 2023.2.1	3.2-8.3
Hedgehog | 2023.1.1	3.2-8.2
Giraffe | 2022.3.1	3.2-8.1
Flamingo | 2022.2.1	3.2-8.0

Minimum versions of tools for Android API level

- There are minimum versions of Android Studio
- AGP that support a specific API level.
- Using lower versions of Android Studio or AGP than required by your project's **targetSdk** or **compileSdk** could lead to unexpected issues.
- It is recommend using the latest preview version of Android Studio and AGP to work on projects that target preview versions of the Android OS.



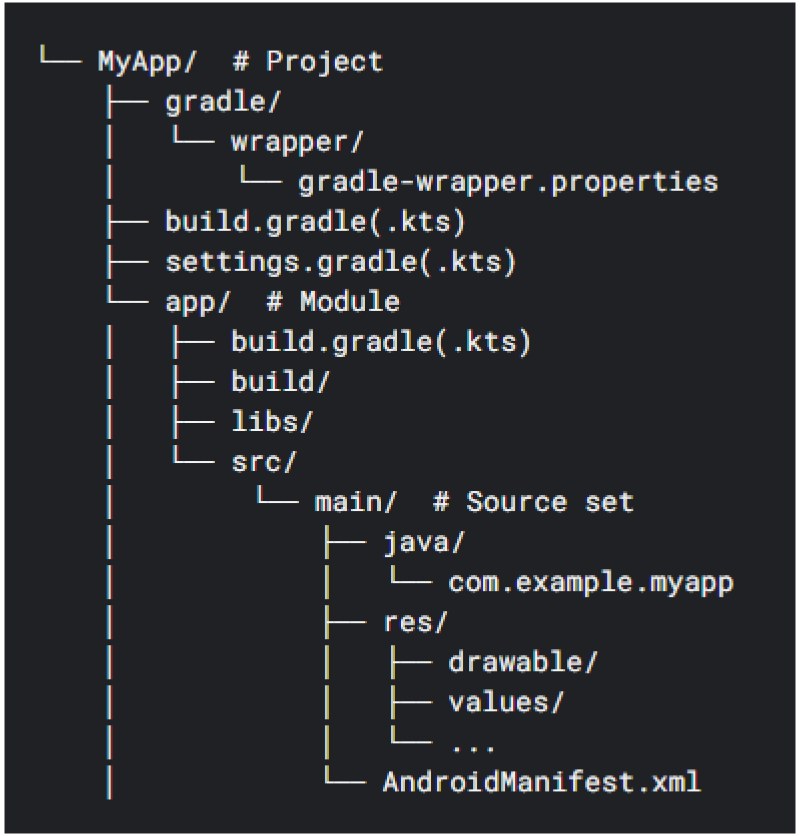
API level	Minimum Android Studio version	Minimum AGP version
VanillaIceCream preview	Jellyfish | 2023.3.1	8.4
34	Hedgehog | 2023.1.1	8.1.1
33	Flamingo | 2022.2.1	7.2

Build configuration files

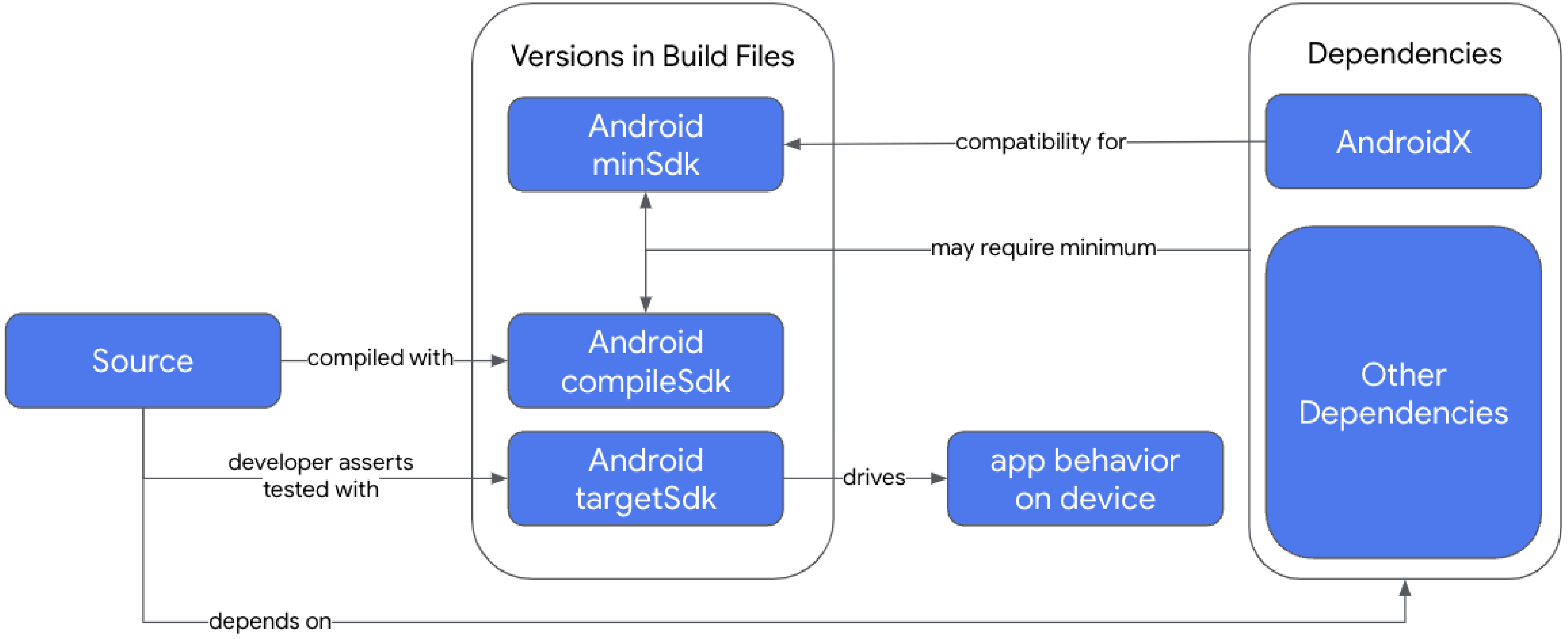
- Creating custom build configurations requires you to make changes to one or more build configuration files.
- These plain-text files use Domain Specific Language (DSL) to describe and manipulate the build logic using Kotlin script, which is a flavor of the Kotlin language.
- You can also use Groovy, which is a dynamic language for the Java Virtual Machine (JVM), to configure your builds.
- When starting a new project, Android Studio automatically creates some of these files for you and populates them based on sensible defaults. The project file structure has the following layout:

Build configuration files

- For more information, refer: https://developer.android.com/build


```
└── MyApp/  # Project
    ├── gradle/
    │   └── wrapper/
    │       └── gradle-wrapper.properties
    ├── build.gradle(.kts)
    ├── settings.gradle(.kts)
    └── app/  # Module
    │   ├── build.gradle(.kts)
    │   ├── build/
    │   ├── libs/
    │   └── src/
    │        └── main/  # Source set
    │            ├── java/
    │            │   └── com.example.myapp
    │            ├── res/
    │            │   ├── drawable/
    │            │   ├── values/
    │            │   └── ...
    │            └── AndroidManifest.xml
```

Android SDK



Versions in Build Files
Dependencies
Android minSdk
AndroidX
compatibility for
may require minimum
Source
compiled with
Android compileSdk
Other Dependencies
developer asserts tested with
Android targetSdk
drives
app behavior on device
depends on