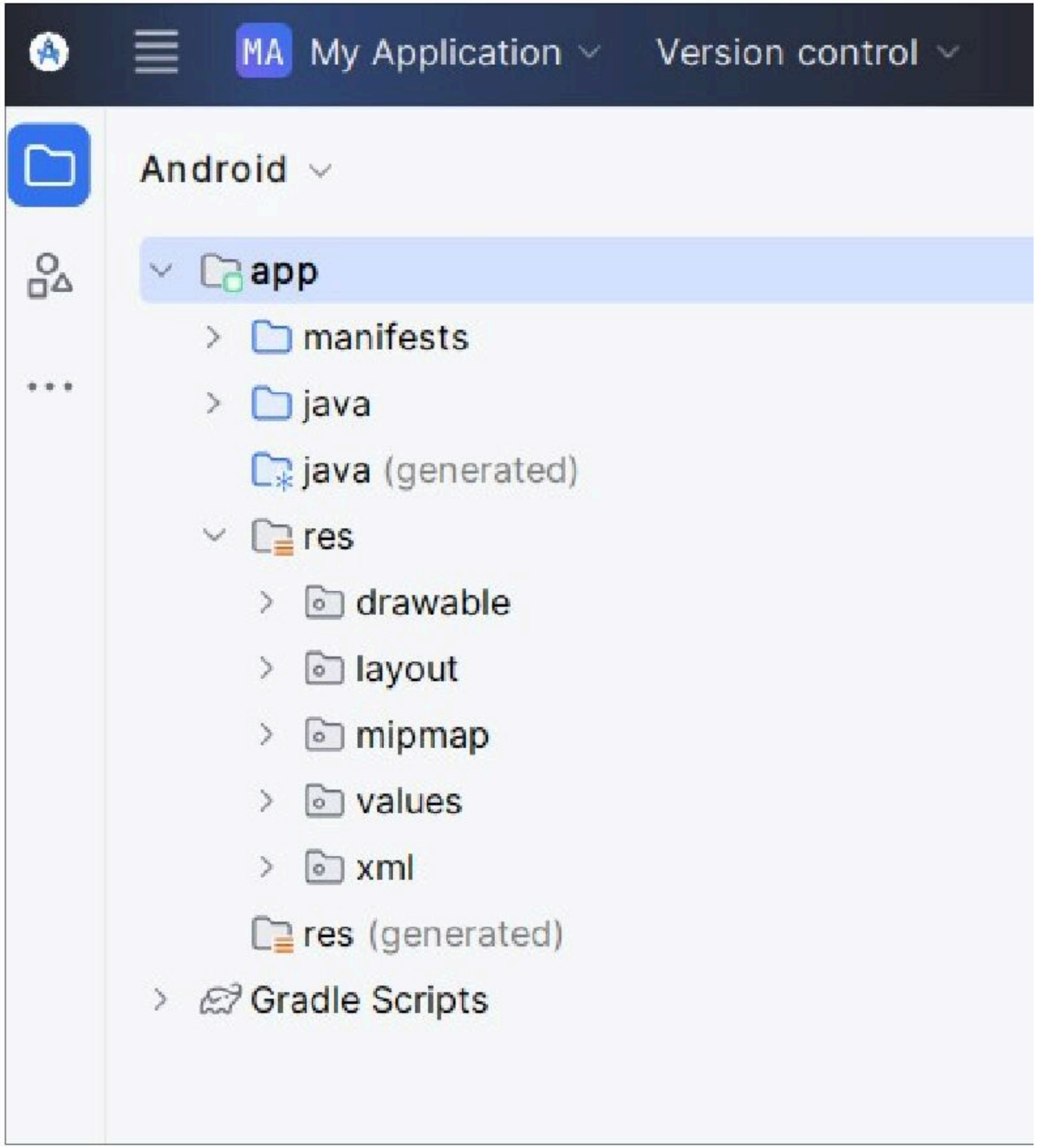
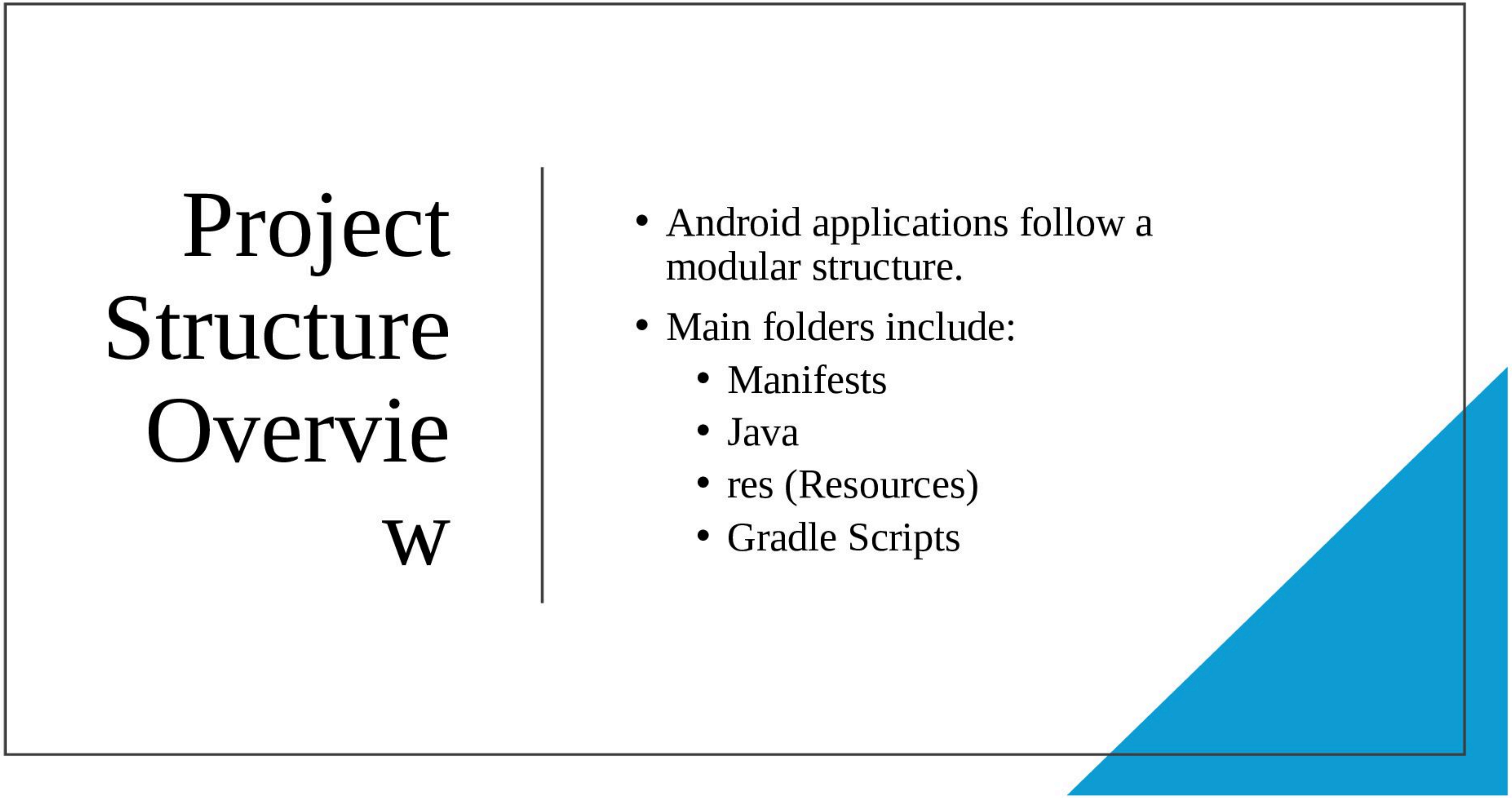
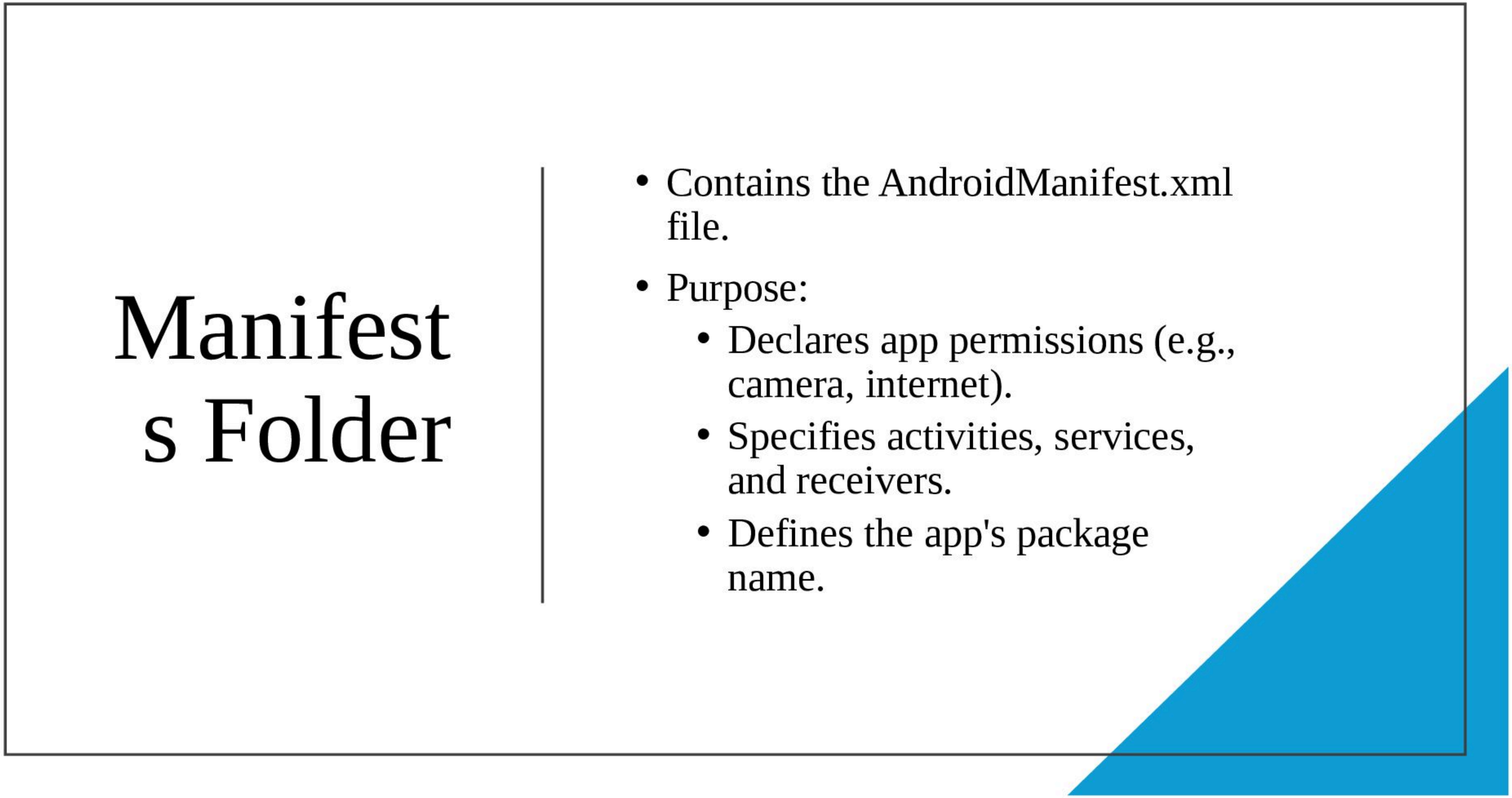
Chapter 2: ANTOMY OF ANDROID APPLICATIO N





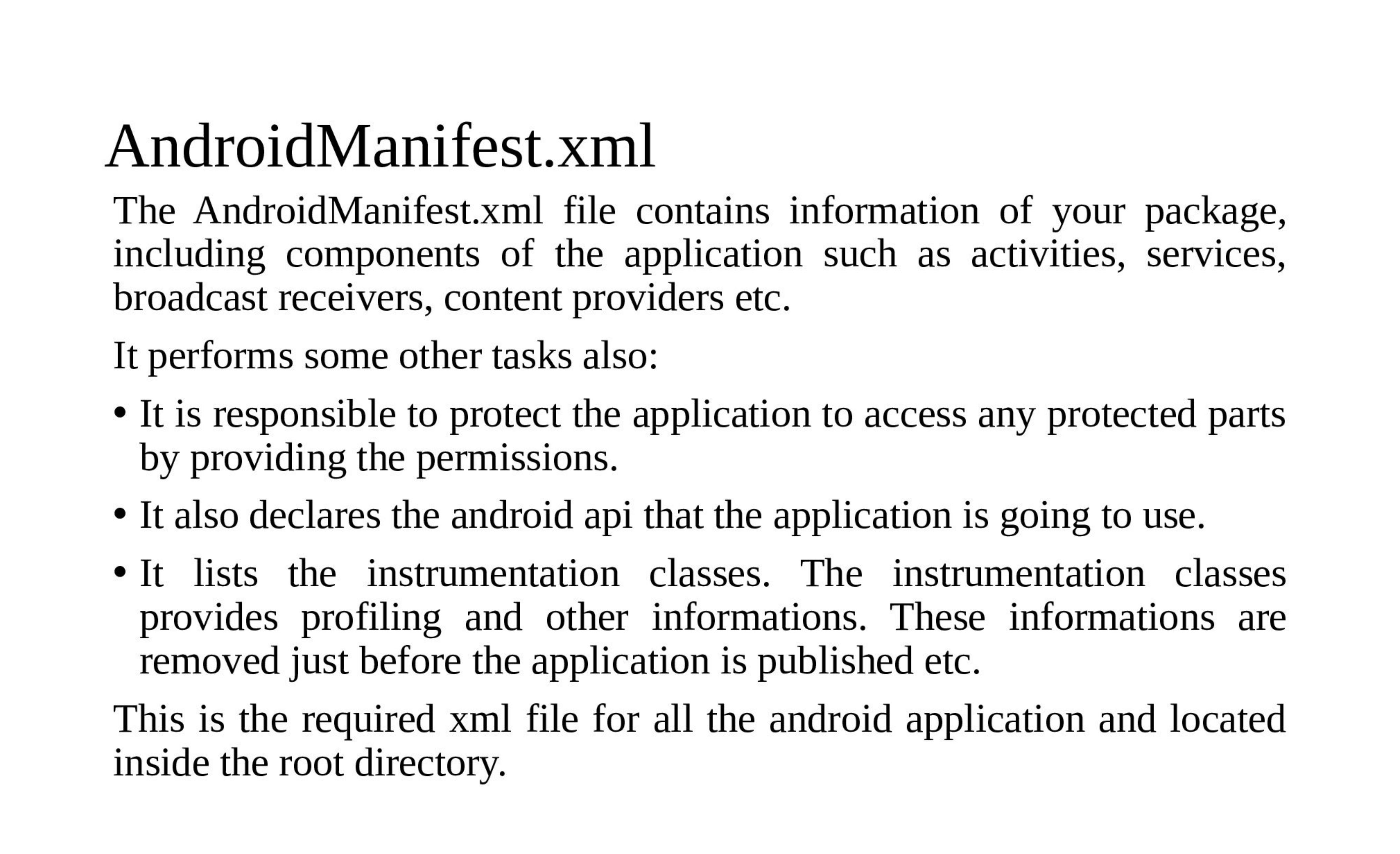
Project Structure Overview

- Android applications follow a modular structure.
- Main folders include:
  - Manifests
  - Java
  - res (Resources)
  - Gradle Scripts



Manifests Folder

- Contains the AndroidManifest.xml file.
- Purpose:
  - Declares app permissions (e.g., camera, internet).
  - Specifies activities, services, and receivers.
  - Defines the app's package name.



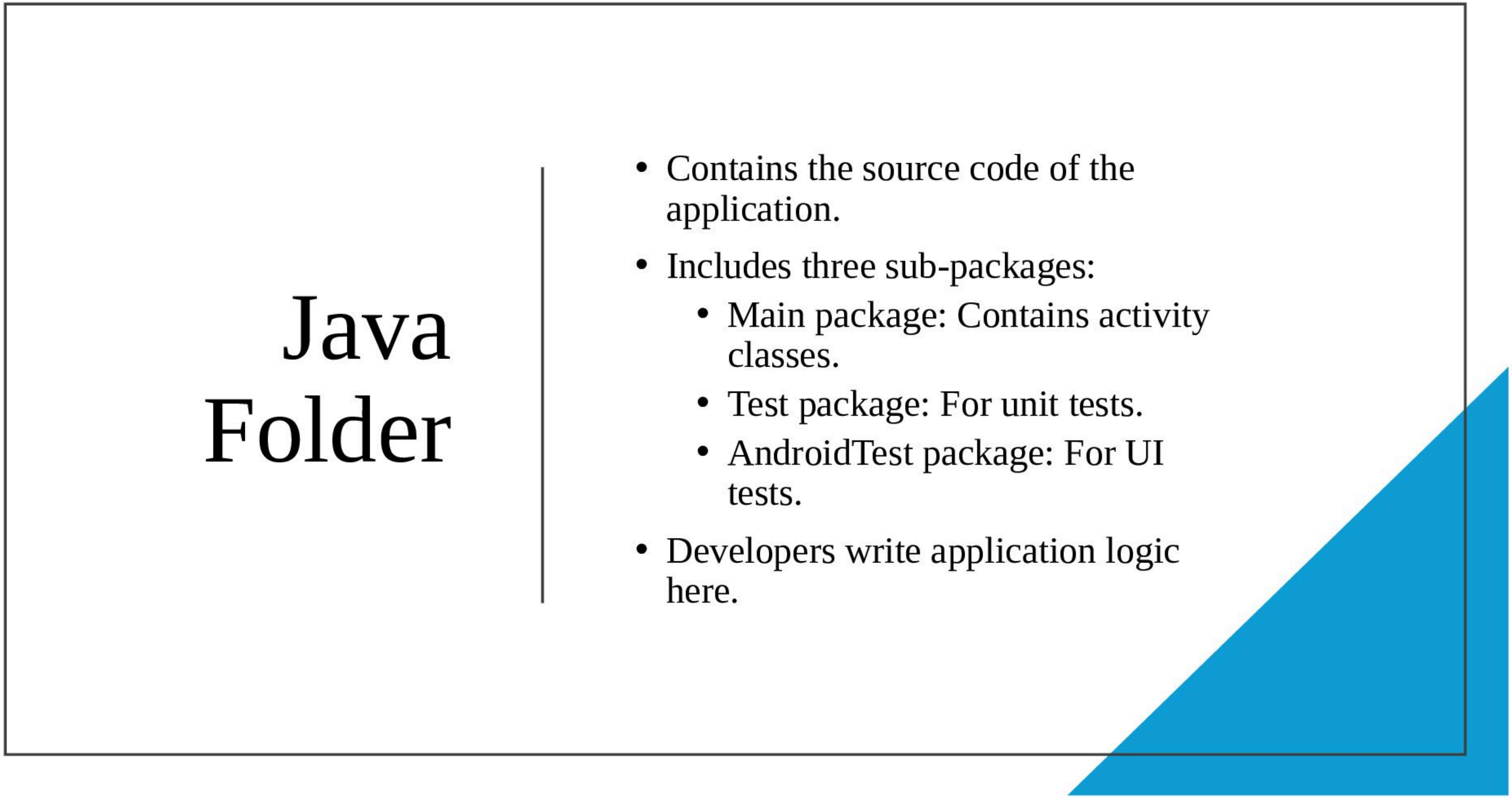
AndroidManifest.xml

The AndroidManifest.xml file contains information of your package, including components of the application such as activities, services, broadcast receivers, content providers etc.

It performs some other tasks also:

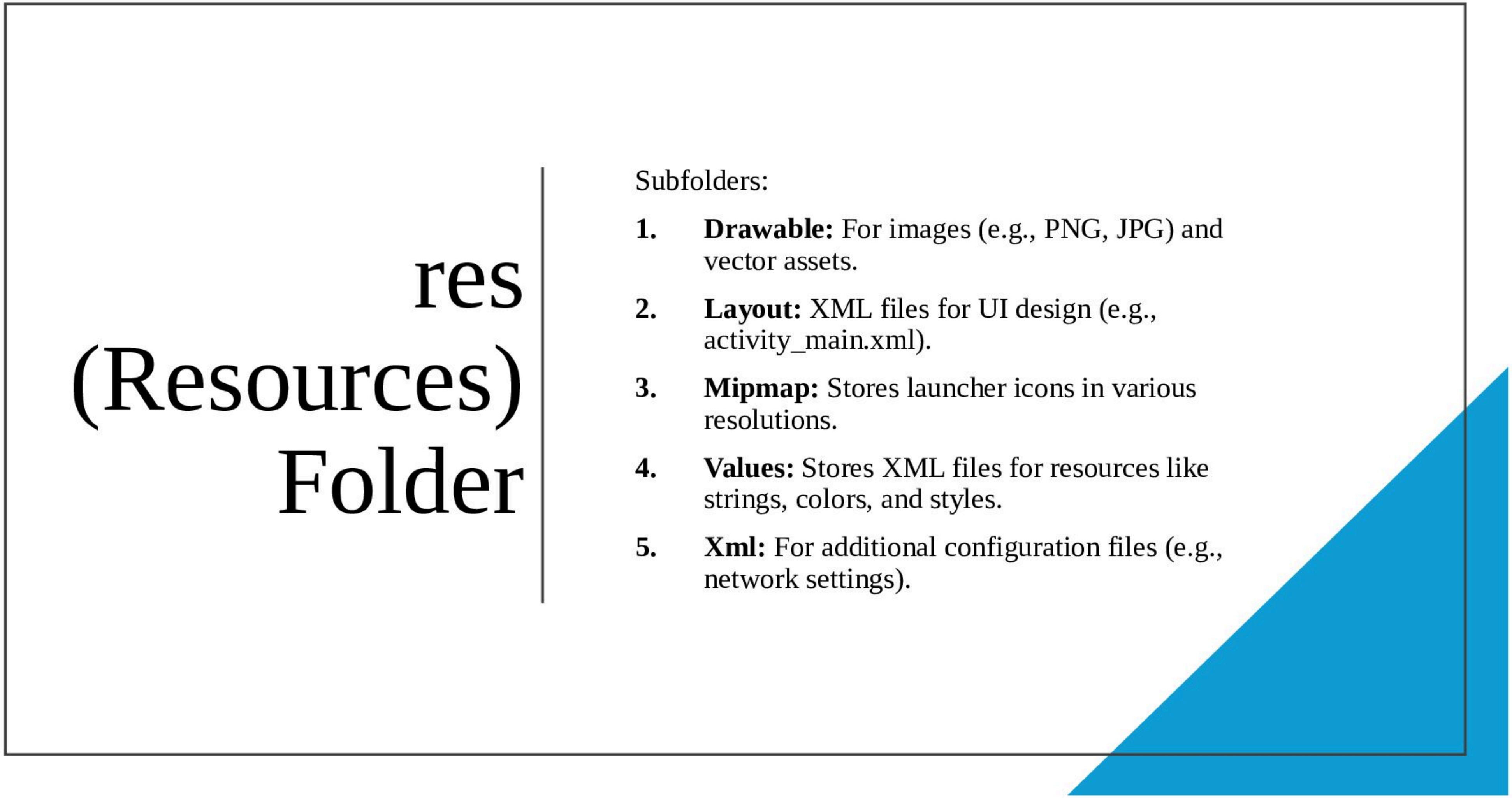
- It is responsible to protect the application to access any protected parts by providing the permissions.
- It also declares the android api that the application is going to use.
- It lists the instrumentation classes. The instrumentation classes provides profiling and other informations. These informations are removed just before the application is published etc.

This is the required xml file for all the android application and located inside the root directory.



Java Folder

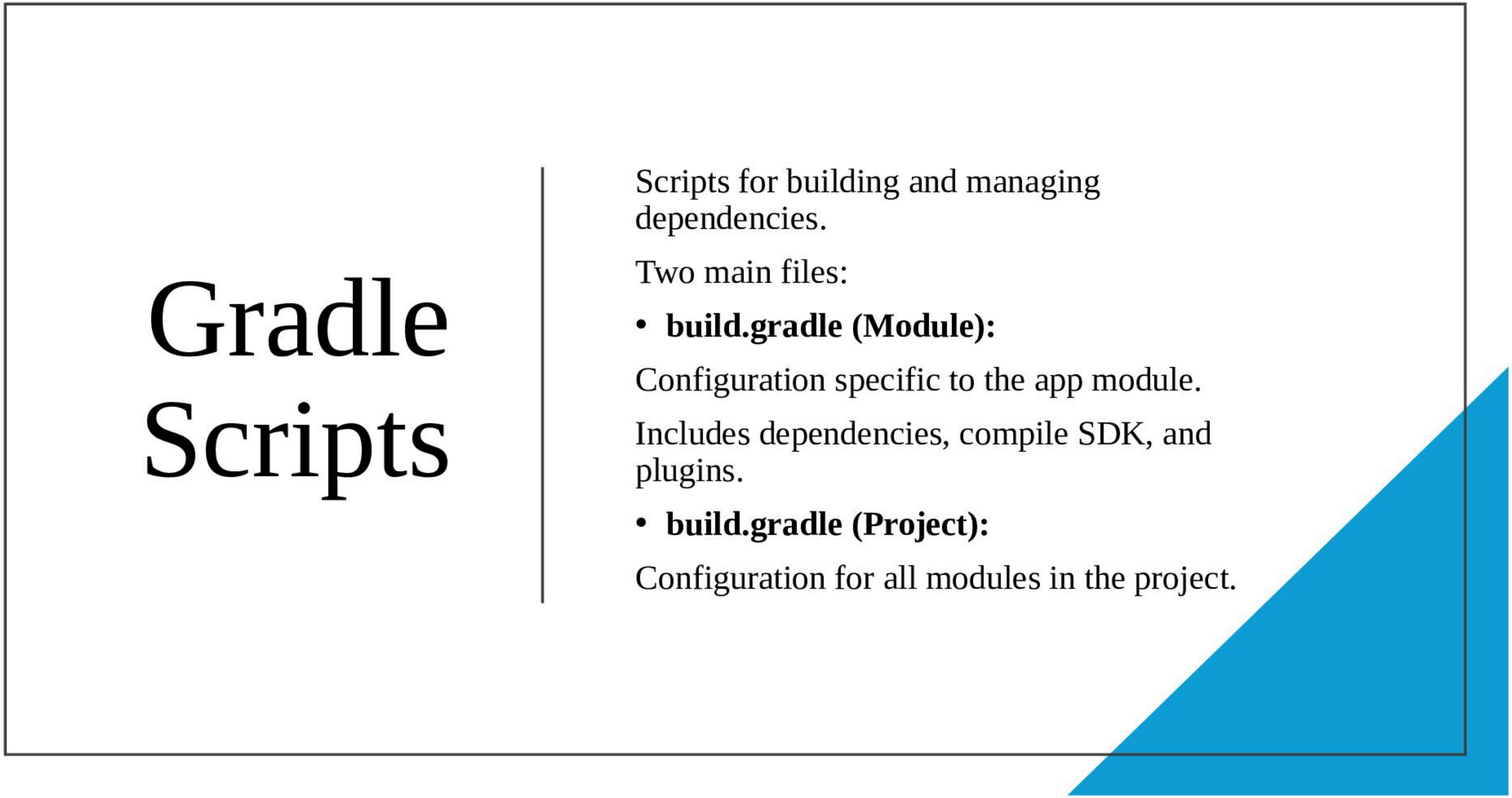
- Contains the source code of the application.
- Includes three sub-packages:
  - Main package: Contains activity classes.
  - Test package: For unit tests.
  - AndroidTest package: For UI tests.
- Developers write application logic here.



res (Resources) Folder

Subfolders:

1. **Drawable:** For images (e.g., PNG, JPG) and vector assets.
2. **Layout:** XML files for UI design (e.g., activity_main.xml).
3. **Mipmap:** Stores launcher icons in various resolutions.
4. **Values:** Stores XML files for resources like strings, colors, and styles.
5. **Xml:** For additional configuration files (e.g., network settings).



Gradle Scripts

Scripts for building and managing dependencies.

Two main files:

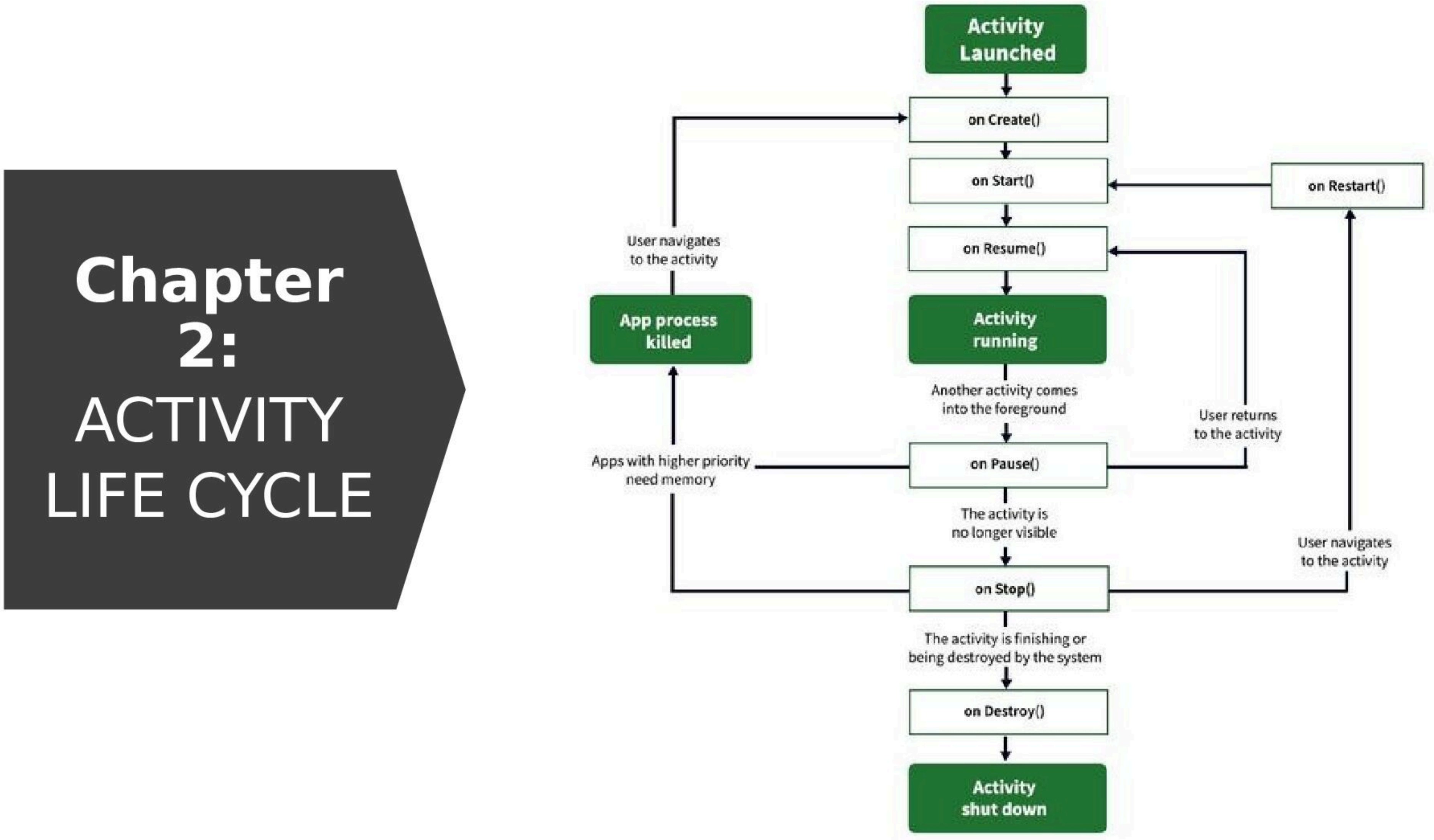
- **build.gradle (Module):**

Configuration specific to the app module.

Includes dependencies, compile SDK, and plugins.

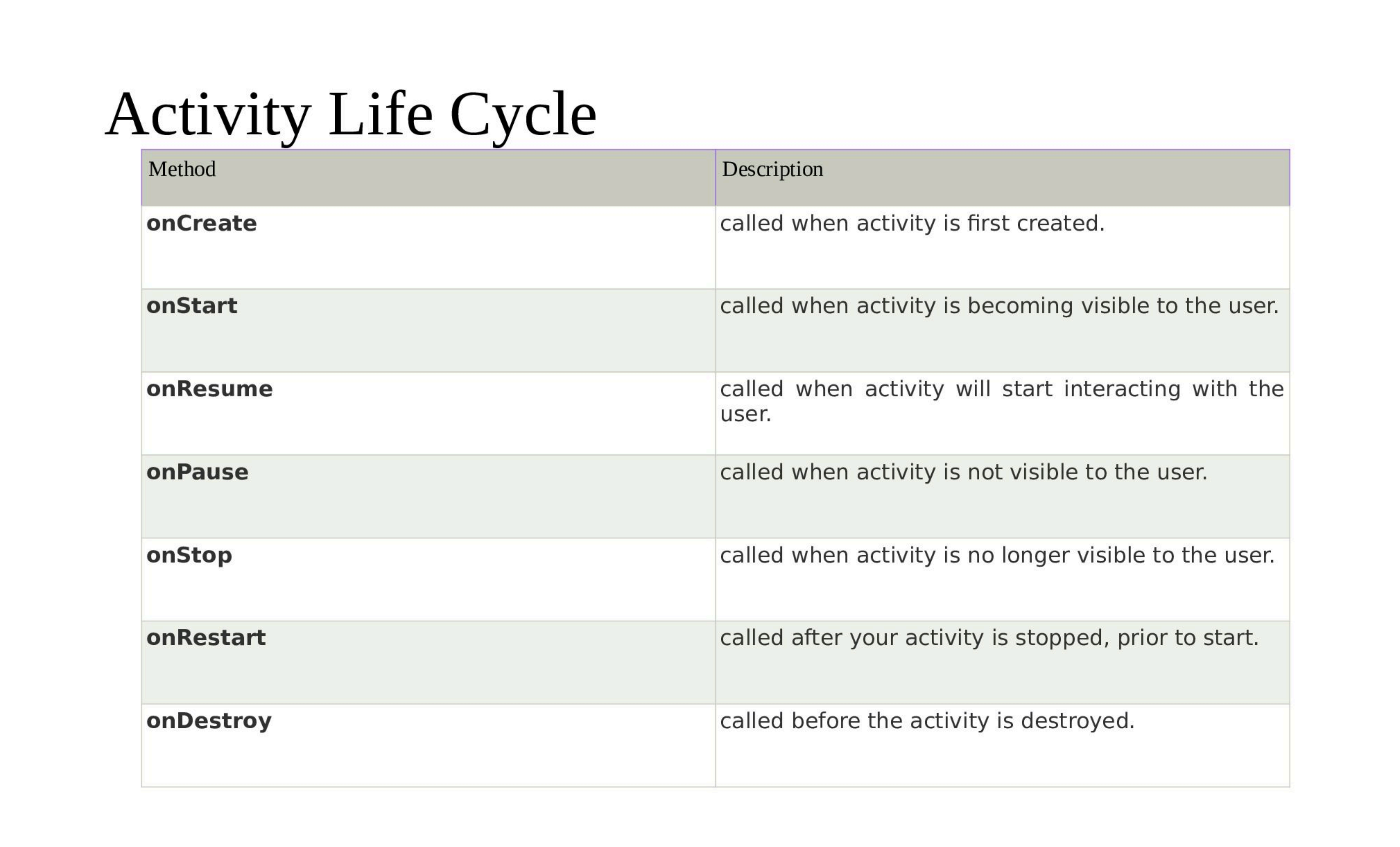
- **build.gradle (Project):**

Configuration for all modules in the project.



Chapter 2: ACTIVITY LIFE CYCLE

Activity Launched
on Create()
on Start()
on Restart()
on Resume()
User navigates to the activity
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
on Pause()
The activity is no longer visible
User navigates to the activity
on Stop()
The activity is finishing or being destroyed by the system
on Destroy()
Activity shut down



Activity Life Cycle

Method	Description
onCreate	called when activity is first created.
onStart	called when activity is becoming visible to the user.
onResume	called when activity will start interacting with the user.
onPause	called when activity is not visible to the user.
onStop	called when activity is no longer visible to the user.
onRestart	called after your activity is stopped, prior to start.
onDestroy	called before the activity is destroyed.