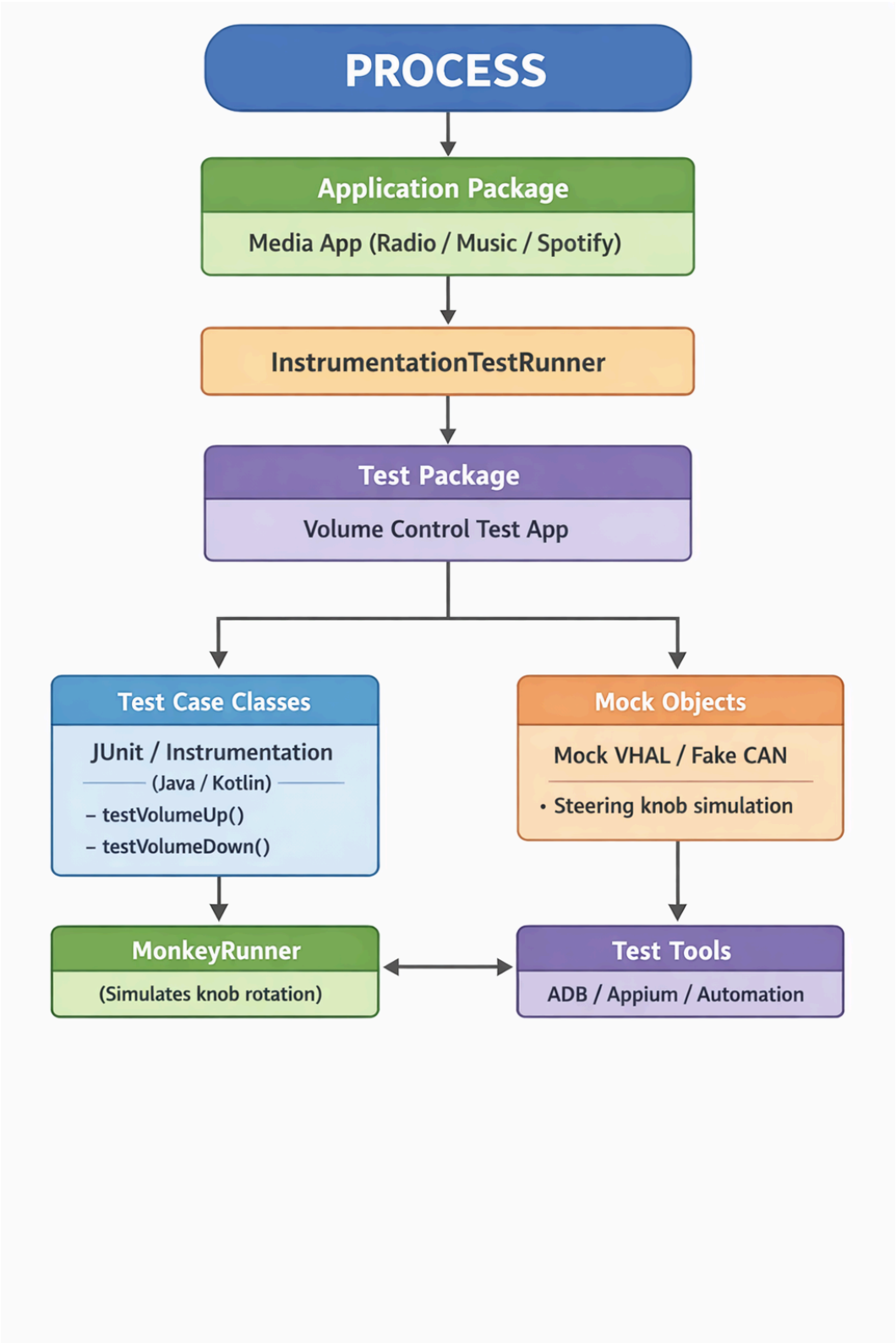
PROCESS
Application Package
Media App (Radio / Music / Spotify)
InstrumentationTestRunner
Test Package
Volume Control Test App
Test Case Classes
JUnit / Instrumentation
(Java / Kotlin)
– testVolumeUp()
– testVolumeDown()
Mock Objects
Mock VHAL / Fake CAN
• Steering knob simulation
MonkeyRunner
(Simulates knob rotation)
Test Tools
ADB / Appium / Automation

1. Application Package (Real App)

The actual media application runs on the system as a normal Android process.
It contains the business logic for handling volume, media playback, and user interactions.

---

2. InstrumentationTestRunner

This is the test execution engine provided by Android.
It launches the app under test and allows test code to:

- Access app internals
- Trigger UI actions
- Validate behavior

It acts as a **bridge between the app and the test package**.

---

3. Test Package (Test App)

A separate APK that contains all test-related code.
In this case, it includes a **Volume Control Test App** responsible for validating volume behavior.

---

4. Test Case Classes (JUnit / Instrumentation Tests)

These are written in **Java or Kotlin** using JUnit and Android instrumentation APIs.
They verify:

- Volume up functionality
- Volume down functionality
- Correct response to hardware input events

Example:

- `testVolumeUp()`
- `testVolumeDown()`

---

5. Mock Objects (Mock VHAL / Fake CAN)

Real vehicle hardware is not used during testing.
Instead, mocks simulate:

- Steering wheel knob rotation
- CAN signals
- Vehicle HAL events

This allows **safe, repeatable, and fast testing** without a real car.

---

6. MonkeyRunner (User Action Simulation)

MonkeyRunner simulates physical user actions such as:

- Turning the volume knob
- Pressing buttons
- Repeated inputs

This helps test real-world user behavior automatically.

---

7. Test Tools (ADB / Appium / Automation)

Automation tools control the full flow:

- ADB starts and stops tests

- Appium handles UI automation
- Scripts integrate tests into CI/CD
