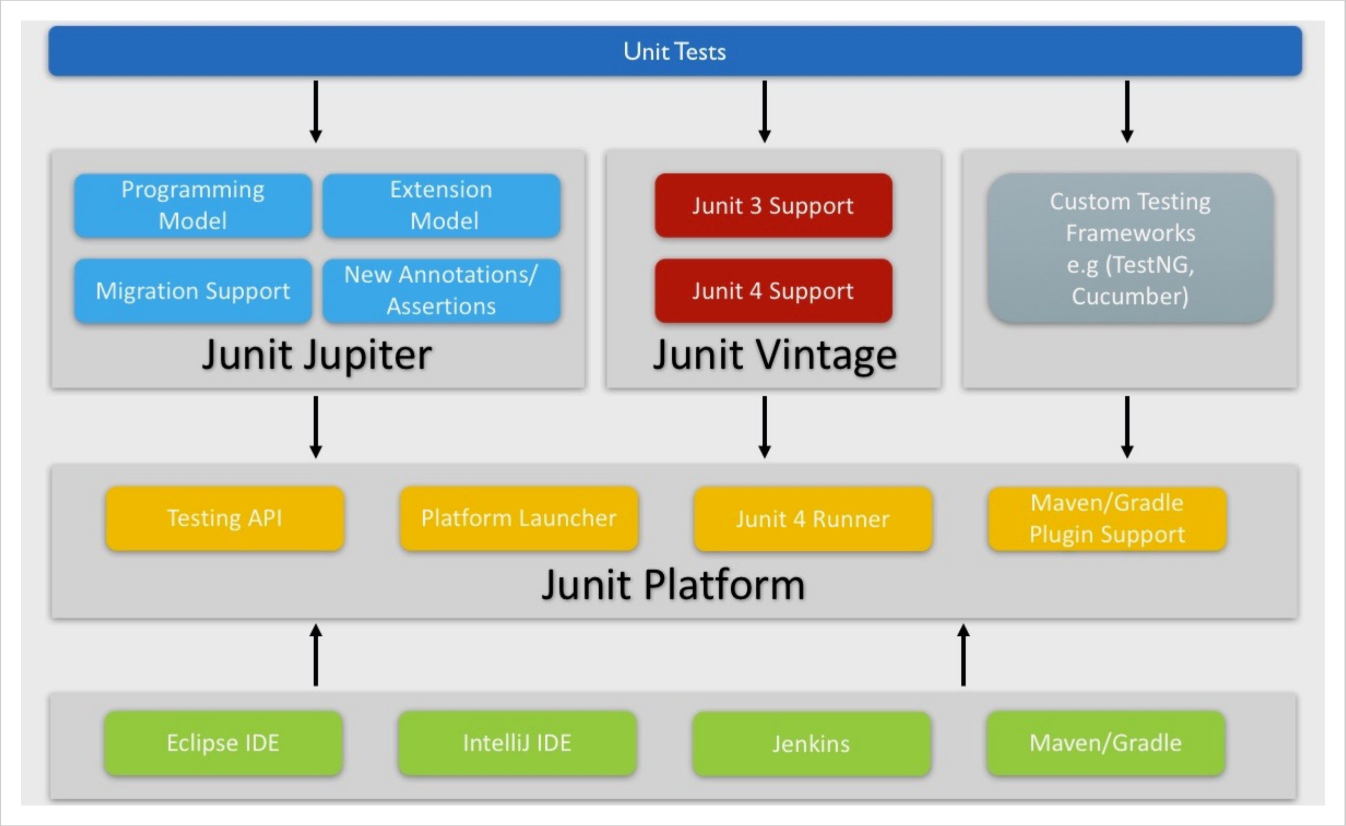
Junit 5 Architecture

This lesson describes Junit 5 architecture, along with their core functionality.



Junit 5 Architecture has three main components -

- **Junit Platform**

  It provides a core foundation to help launching testing frameworks on JVM. It acts as an interface between JUnit and its clients such as build tools (`Maven and Gradle`) and IDE's (`Eclipse and IntelliJ`). It introduces the concept of a `Launcher` which external tools use to discover, filter, and execute tests.

It also provides the `TestEngine API` for developing a testing framework that runs on the JUnit platform. Using `TestEngine API`, 3rd party testing libraries such as Spock, Cucumber, and FitNesse can directly plug in and provide their custom TestEngine.

- **Junit Jupiter**

It provides a new programming model and extension model for writing tests and extensions in Junit 5. It has a whole new annotation to write test cases in Junit 5. Some of the annotations are `@BeforeEach`, `@AfterEach`, `@AfterAll`, `@BeforeAll` etc. It implements TestEngine API provided by Junit Platform so that Junit 5 test can be run.

- **Junit Vintage**

  The term `Vintage` basically means **classic**. Thus, this sub-project provides extensive support for writing test cases in JUnit 4 and JUnit 3. Thus, backward compatibility is been provided by this project.

JUnit 5 Architecture

Quiz on JUnit 5 Architecture

Q

JUnit 4 test cases are run on JUnit Vintage Engine.

COMPLETED 0%

1 of 1 ‹ ✓

Quiz on JUnit 5 Architecture

Q

JUnit 5 test cases are run on JUnit Jupiter Engine.

COMPLETED 0%

1 of 1 ‹ ✓

In the next lesson, we will learn to create our first JUnit 5 test case.