**Lifecycle Phases in Gradle**  
1. **Initialization**  
 *init.gradle + <other>.gradle* First phase and Runs before the script run starts  
 The .gradle files are looked for in user-home-directory/.gradle/init.d/  
 Files in init.d are run in file name text order  
  
 Setup info like  
 enterprise wide configurations fpr something like  
 where to find customised plugins  
 by having properties based on current environment  
 local machine versus CI server etc  
 *settings.gradle*  
 Runs at the end of initialization phase  
  
2. **Configuration**  
 *build.gradle*  
 In configuration phase, build.gradle iss converted into Script Interface object that delegates to Project interface object. Project object of the folder in which build.gradle exists.  
 So build.gradle can access methods and properties from Script Interface  
 And Script Interface has reference to Project object  
3**. Execution** Performs tasks/actions in *build.gradle*

**Defining a custom property** to hold value or closure. This can be used in scripts later.

A computer code with text

Description automatically generated

.ext allows define user properties. In this case, the user custom defined value/Closure assigned to gradle **ext** property *timestamp* can be accessed directly using $gradle.timestamp(). timestamp in this example holding a Closure gradle.ext.timestamp is defined in init.gradle and accessed in build.gradle as gradle.timestamp().

**Initialization Phase when init.gradle or other \*.gradle files that can be found in user-home/.gradle/init.d are run**

At this time, the script does not know how many projects exist and what they are. Hence the delegate object would be limited to the Global Gradle object.

A white flower with black text

Description automatically generated

**Initialization Phase when settings.gradle is run**settings.gradle has information about all the projects and their locations. Hence the delegate object would be Settings object that has a collection of Projects each containing build.gradle in them.

A diagram of a setting

Description automatically generated

**Configuration Phase when build.gradle is run**build.gradle contains inside a project folder. Hence the delegate would be current Project object.

A diagram of a script

Description automatically generated

**Summary**A diagram of a process

Description automatically generated

Project object also has reference to Gradle object  
Settings object also has reference to Gradle object

**Main Gradle Lifecycle objects**

A screenshot of a computer

Description automatically generated  
  
DSL References  
<https://docs.gradle.org/current/dsl/index.html>  
<https://docs.gradle.org/current/dsl/org.gradle.api.Project.html>  
<https://docs.gradle.org/current/dsl/org.gradle.api.invocation.Gradle.html>  
Java API References  
<https://docs.gradle.org/current/javadoc/org/gradle/api/package-summary.html>  
<https://docs.gradle.org/current/javadoc/org/gradle/api/Project.html>  
<https://docs.gradle.org/current/javadoc/org/gradle/api/invocation/Gradle.html>

**Gradle in scripts makes it easy by assuming the obvious delegate object and allows to access the delegate object properties directly.**

1. build.gradle delegating to Project object can refer to project object properties directly and need not mention “project.”

project.gradle.gradleVersion can be accessed directly as gradle.gradleVersion because project is the obvious delegate object and Gradle makes it easy by assuming properties not found in the script to be available in delegate object.

1. settings.gradle delegating to Settings object can refer to settings object properties directly and need not mention “settings.”
2. init.gradle delegating to Gradle object can refer to gradle object properties directly and need not mention “gradle.”

**Properties**

1. **Built-in Properties**

A screenshot of a computer program

Description automatically generated

1. **Key-Value Pairs**

A screenshot of a computer program

Description automatically generated

1. **Extension Properties**

Properties that can be attached to Domain objects like Project object.

A yellow and orange sun with red text

Description automatically generated

<https://docs.gradle.org/current/dsl/org.gradle.api.plugins.ExtraPropertiesExtension.html>

A screenshot of a computer

Description automatically generated

A diagram of a project

Description automatically generated

Two phases in which build.gradle comes into play

A black line drawing of a couple of objects

Description automatically generated with medium confidence

1. Project maintains a collection of zero or more tasks
2. Task Container maintains information about tasks and can lookup using task names
3. Each Task maintains a collection of actions.
4. Each action is either an Action interface or Closure object.

A diagram of a shopping bag

Description automatically generated

**Adding actions/closures to a Task**  
A diagram of a chain actions

Description automatically generated

1. hello task would be created in the project during Configuration phase
2. hello task would be executed during the Execution phase
3. doLast and doFirst methods on Task allow add actions to the task either at the end or beginning of the actions collection maintained in the Task object

A computer screen with text and arrows

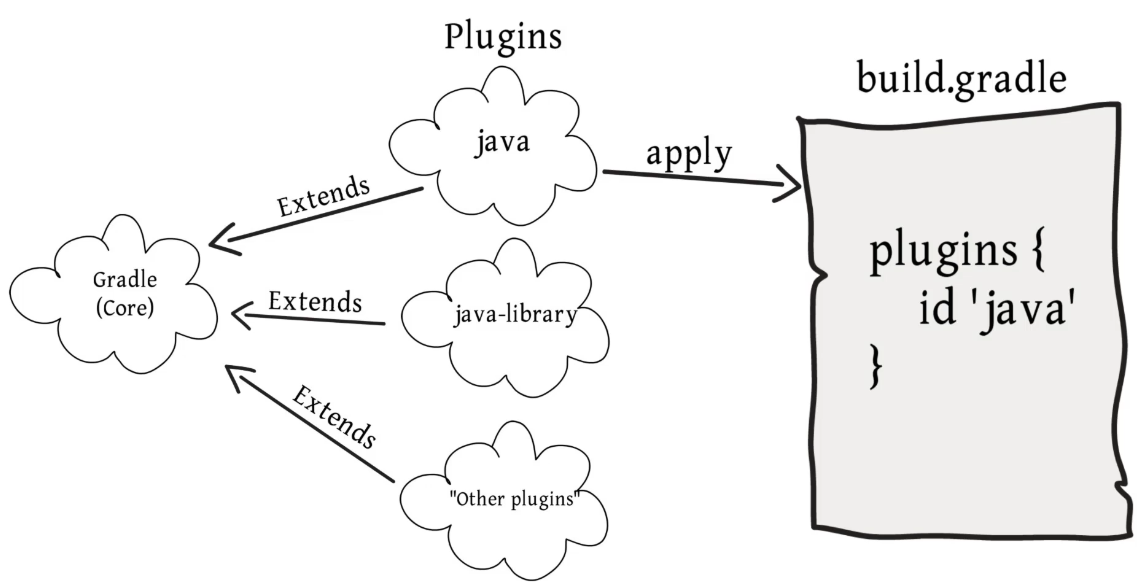
Description automatically generated with medium confidence

A diagram of a process

Description automatically generated

Errors out runtime if cyclic dependency is detected. Gradle does DFS – Depth First Search  
A diagram of a cloud system

Description automatically generated



A diagram of a computer program

Description automatically generated

**Analyzing library dependencies**  
A screenshot of a computer program

Description automatically generated

Run help group -> **dependencies** gradle task to find out transitive dependencies in the project and fix any duplicates or version conflicts

A screenshot of a computer program

Description automatically generated

A screenshot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

<https://docs.gradle.org/current/userguide/java_library_plugin.html#sec:java_library_configurations_graph>.

[https://abhiappmobiledeveloper.medium.com/difference-between-implementation-api-compile-and-runtimeonly-in-gradle-dependency-55b70215d245A table with black text

Description automatically generated](https://abhiappmobiledeveloper.medium.com/difference-between-implementation-api-compile-and-runtimeonly-in-gradle-dependency-55b70215d245)

<https://docs.gradle.org/current/userguide/java_library_plugin.html#sec:java_library_configurations_graph>  
Refer to Table 1 in the link page to know the meaning all the possible configuration scopes like api, implementation, compileOnly, runtimeOnly etc.­­

**Changing the name of the default build.gradle file in subprojects**

Following used in the parent settings.gradle file will allow every subproject to have gradle file matching the name of the subproject name. This will avoid the confusion when we have multiple build.gradle files open.

A computer screen shot of a black background

Description automatically generated

A screenshot of a computer

Description automatically generated

If very specific gradle wrapper version is required

A close-up of a text

Description automatically generated

And then run the ‘wrapper’ task under ‘build setup’ group. This would update the gradle/wrapper folder with appropriate gradle-wrapper.zip and gradle-wrapper.properties.

Executing tasks from CLI

A screen shot of a computer

Description automatically generated

