



Presentado por **Carlos Icaza**
www.consultec-ti.com



Fundamentos

Objetivos

- Aprender los fundamentos de Gradle.
- Instalar y configurar Gradle.
- Usar Gradle desde la consola y desde Eclipse.
- Construir automáticamente aplicaciones Java usando Gradle.

¿Qué es Gradle?



Gradle, es una herramienta que permite la automatización de compilación de código abierto, la cual se encuentra centrada en la flexibilidad y el rendimiento.

Los scripts de compilación de Gradle se escriben utilizando Groovy o Kotlin DSL (Domain Specific Language).

¿Qué es Gradle?



Gradle tiene una gran flexibilidad y nos deja hacer usos otros lenguajes y no solo de Java, también cuenta con un sistema de gestión de dependencias muy estable.

Gradle es altamente personalizable y rápido, ya que completa las tareas de forma rápida y precisa.

Además es el sistema de compilación oficial para Android y cuenta con soporte para diversas tecnologías y lenguajes.

Documentación

Manual de Usuario

<https://docs.gradle.org/current/userguide/userguide.html>

Referencia del DSL

<https://docs.gradle.org/current/dsl/>

Guías Generales

<https://gradle.org/guides/>

Plugins

Gradle Plugins

<https://plugins.gradle.org/>

Características de Gradle

Depuración colaborativa: Permite compartir los resultados de la compilación para resolver en equipo de forma eficiente posibles problemas que aparezcan.

Construcción incremental: Valida en el proceso de compilación si la entrada, salida o implementación de una tarea ha cambiado, en caso de no existir algún cambio la considera actualizada y no se ejecuta.

Características de Gradle

Dependencias transitivas: Es uno de los principales beneficios que ofrece al utilizar la gestión de dependencias ya que se encarga de descargar y administrar las dependencias transitivas.

Soporte a Groovy y Scala incorporado: Compatibilidad con los proyectos de Groovy, permitiendo trabajar con código Groovy o código Scala e inclusive desarrollar código mixto Java y Groovy o Java y Scala.

Características de Gradle

Compilación incremental para Java: En caso de que el código fuente o la ruta de clase cambien, Gradle cuenta con la capacidad para detectar todas las clases que se vean afectadas por dicho cambio y procederá a recompilarlas.

Publicar en repositorios Ivy y Maven: Permite publicar Artifacts en repositorios Ivy con diseños de directorios completamente personalizables. De igual modo, sucede con Maven en Bintray o Maven Central.

Características de Gradle

Lee el formato POM: Es compatible con el formato de metadatos POM, por lo que es posible recuperar dependencias de cualquier repositorio compatible con Maven.

Soporta el principio **Convención sobre Configuración**

Características Avanzadas de Gradle

- Ejecución de pruebas unitarias en paralelo
- Gestión de dependencias
- Compatibilidad para compilaciones incrementales
- Tareas dinámicas
- Gradle Daemon
- Groovy DSL
- Kotlin DSL

¿Groovy?

El script de compilación de Gradle está escrito en Groovy.

Todo el diseño de Gradle está orientado a ser utilizado como un lenguaje, no como un framework rígido.

Groovy te permite escribir tu propio script con algunas abstracciones. Toda la API de Gradle está diseñada en lenguaje Groovy.

¿Kotlin?

Groovy es el idioma predeterminado para escribir scripts de Gradle.

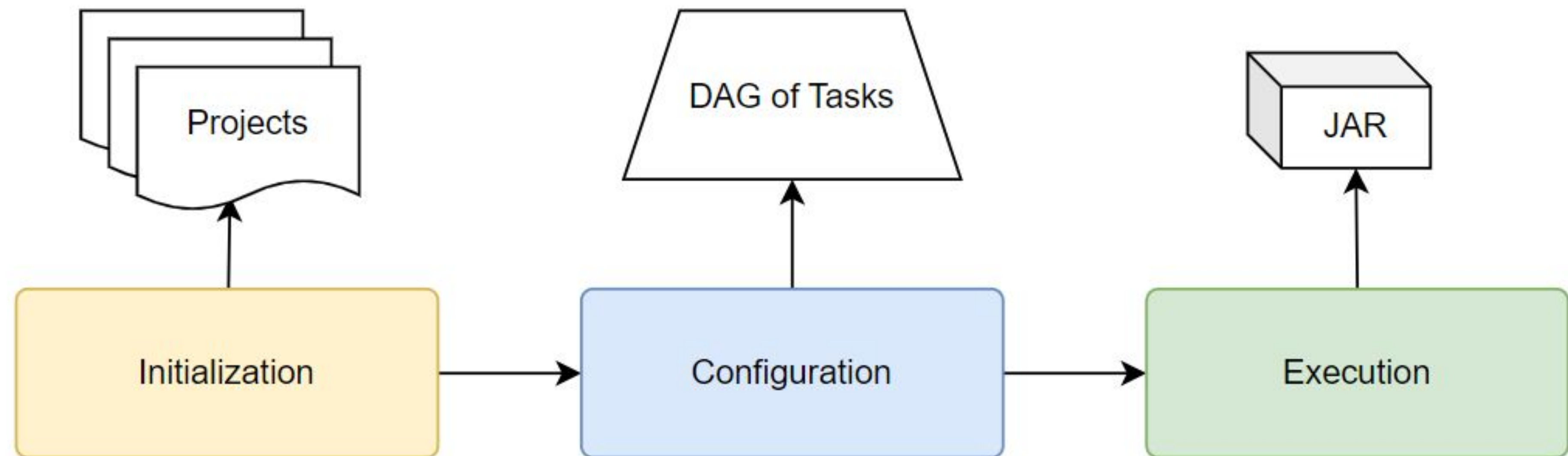
Sin embargo, desde la versión 5.0 de Gradle, también podemos escribir estos scripts en Kotlin.

Características de Maven

Convención sobre configuración

Maven intenta evitar tanta configuración como sea posible, eligiendo valores predeterminados y suministrando plantillas de proyecto.

Ciclo de vida de compilación



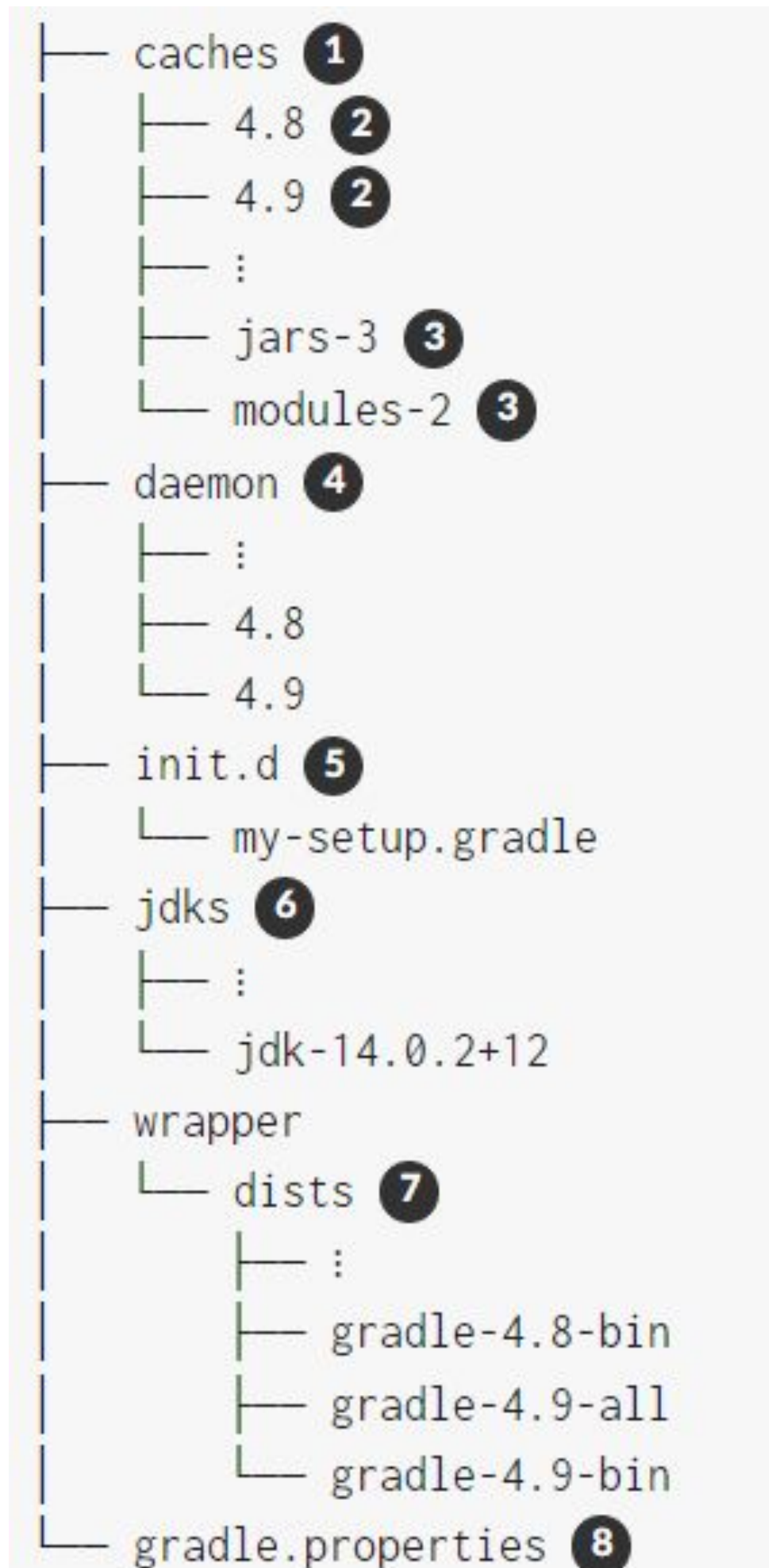
Ciclo de vida de compilación

- **Fase de inicialización**
 - Configurar entorno (init.gradle, gradle.properties)
 - Encontrar proyectos y build scripts (settings.gradle)
- **Fase de configuración**
 - Evaluar todos los build scripts
 - Construir modelo de objetos (Gradle -> Project -> Task, etc.)
 - Crear gráfico de ejecución de tareas (DAG)
- **Fase de ejecución**
 - Ejecutar las tareas

Archivos de configuración

Gradle user home directory

Ubicación: ~/.gradle



1. Global cache directory
2. Version-specific caches
3. Shared caches
4. Registry and logs of the Gradle Daemon
5. Global initialization scripts
6. JDKs downloaded by the toolchain support
7. Distributions downloaded by the Gradle Wrapper
8. Global Gradle configuration properties

Archivos de configuración

Project root directory

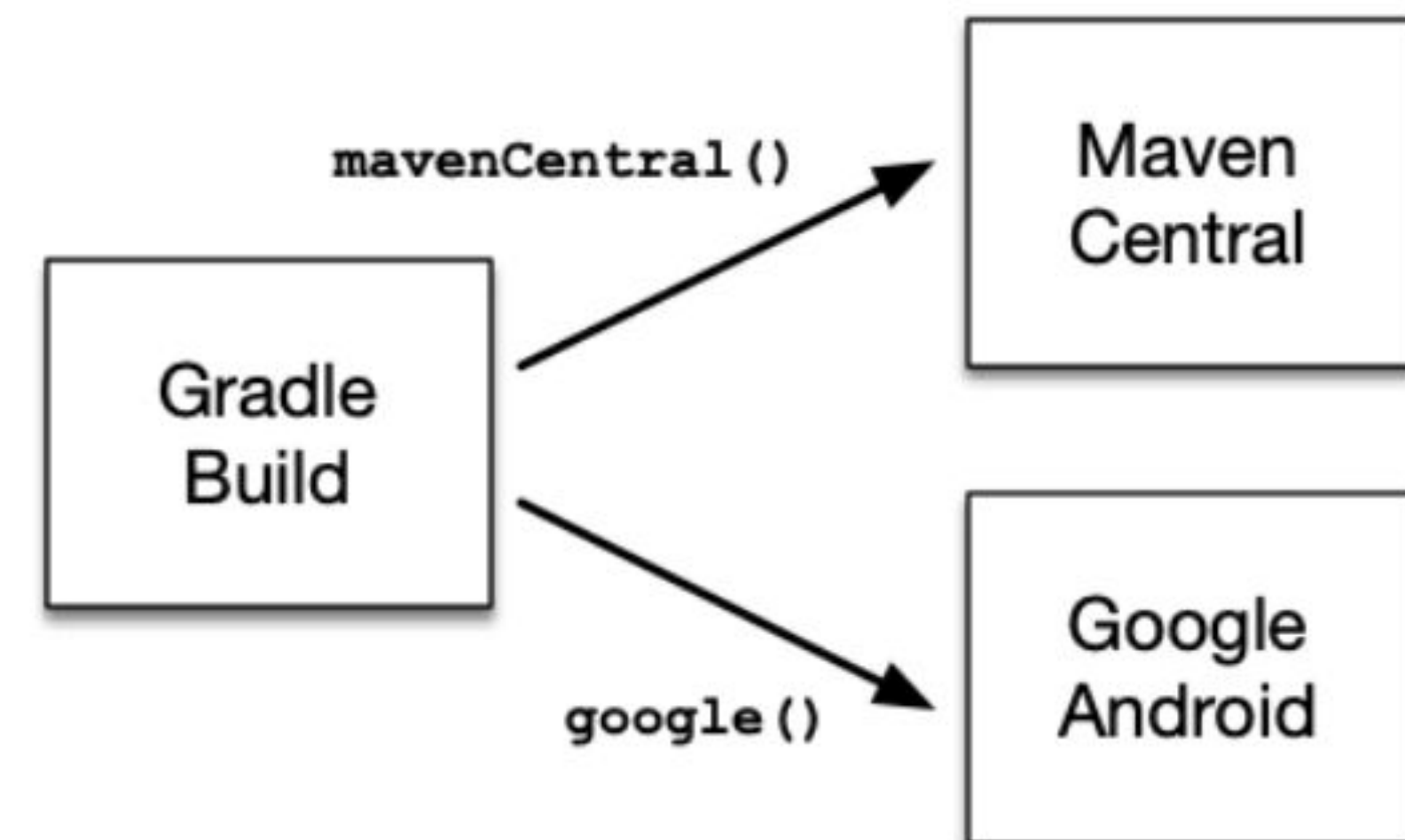
```
| .gradle 1  
| | 4.8 2  
| | 4.9 2  
| | :  
| build 3  
| gradle  
| | wrapper 4  
| gradle.properties 5  
| gradlew 6  
| gradlew.bat 6  
| settings.gradle or settings.gradle.kts 7  
| subproject-one 8  
| | build.gradle or build.gradle.kts 9  
| subproject-two 8  
| | build.gradle or build.gradle.kts 9  
| :  
|
```

1. Project-specific cache directory generated by Gradle
2. Version-specific caches
3. The build directory of this project into which Gradle generates all build artifacts.
4. Contains the JAR file and configuration of the Gradle Wrapper
5. Project-specific Gradle configuration properties
6. Scripts for executing builds using the Gradle Wrapper
7. The project's settings file where the list of subprojects is defined
8. Usually a project is organized into one or multiple subprojects
9. Each subproject has its own Gradle build script

Repositorios Gradle

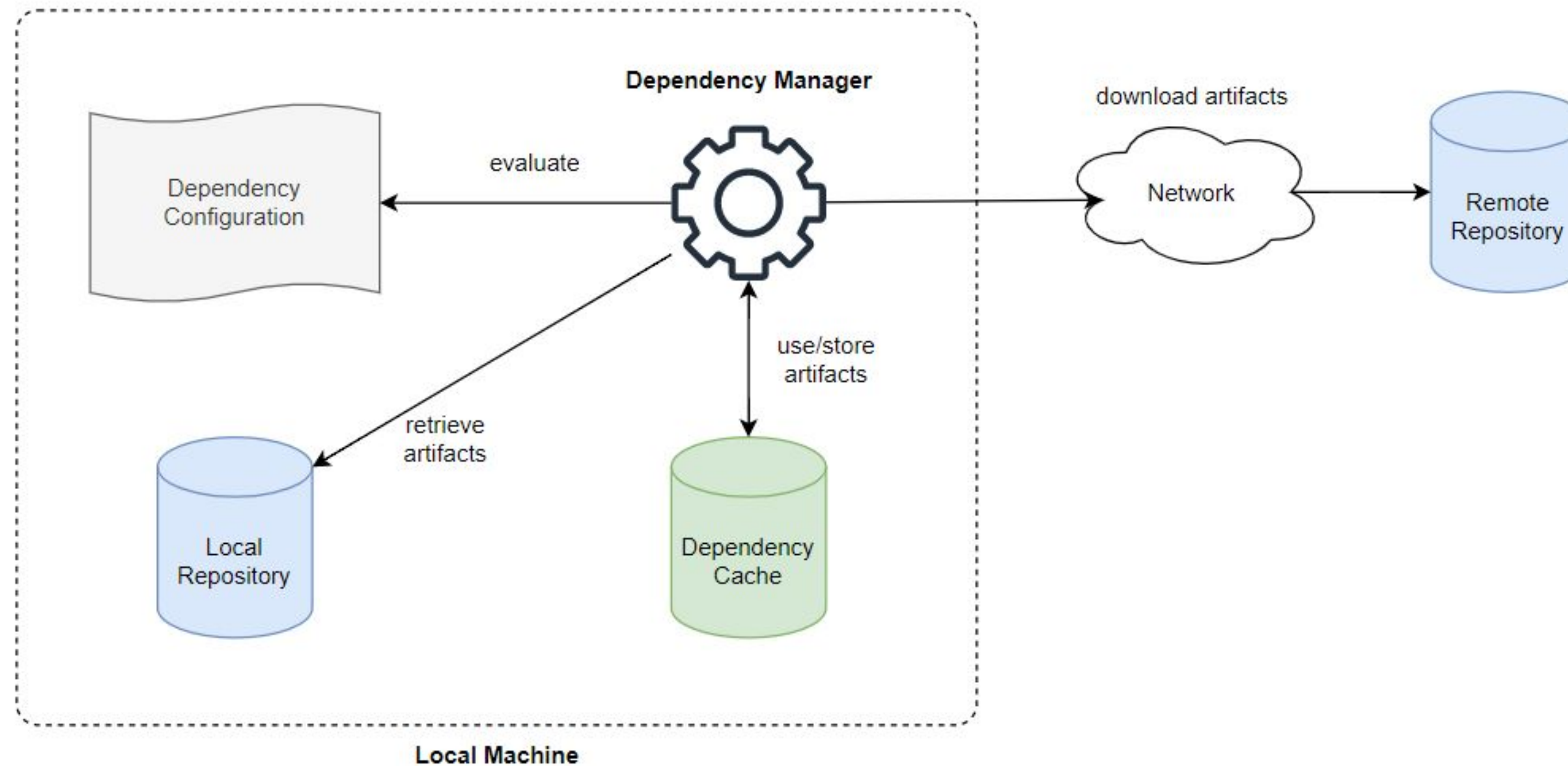
Gradle puede resolver dependencias de uno o varios repositorios basados en Maven o Ivy.

```
repositories {  
    mavenLocal()  
    mavenCentral()  
    google()  
    maven{  
        url "https://repo.spring.io/release"  
    }  
}
```



Caché de Dependencias

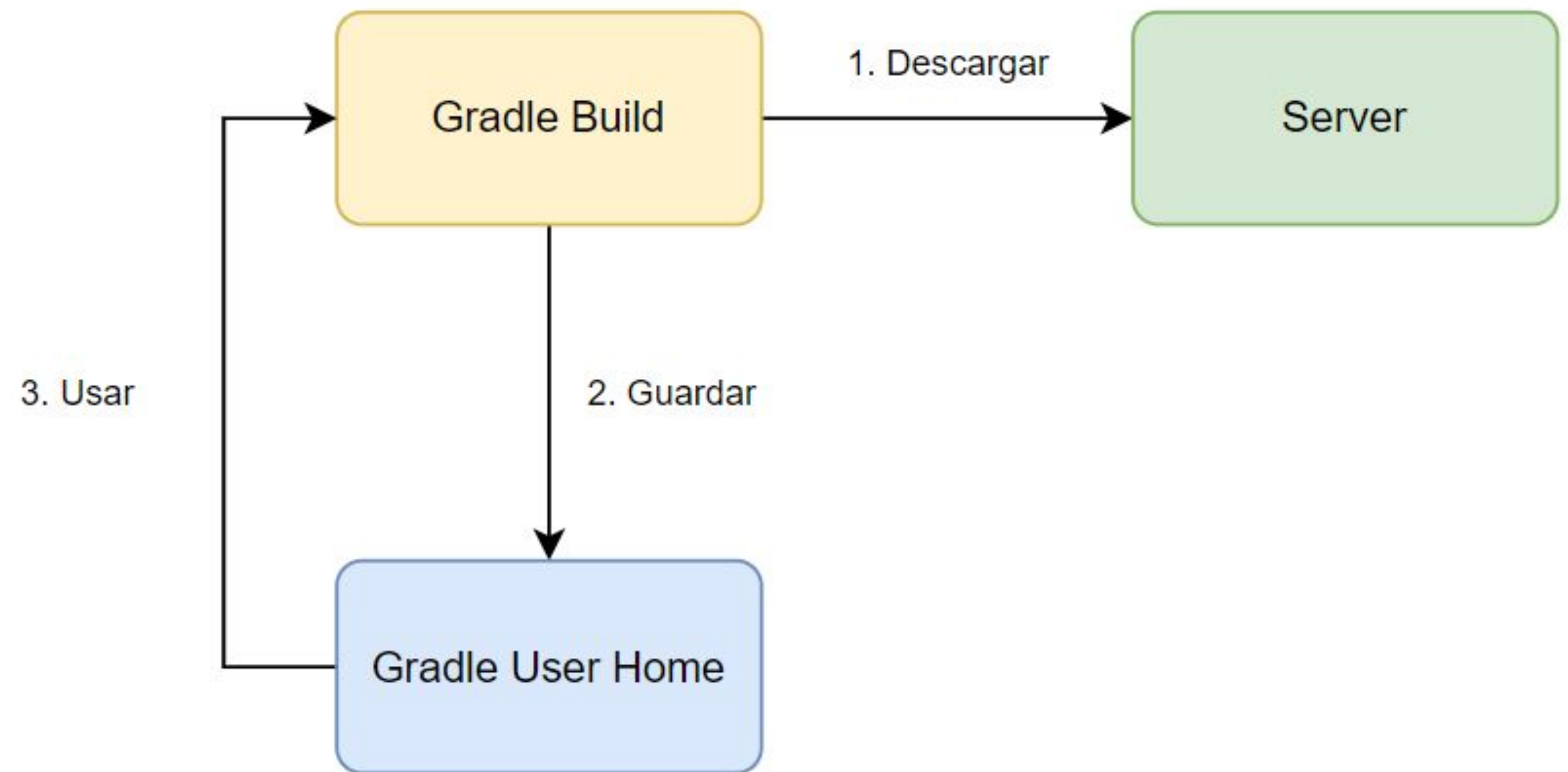
Ubicación: `~/ .gradle/caches/...`



Gradle Wrapper

Estandariza un proyecto en una versión determinada de Gradle, lo que genera compilaciones más confiables y sólidas.

Aprovisionar una nueva versión de Gradle para diferentes usuarios y entornos de ejecución (por ejemplo, IDE o servidores de integración continua) es tan simple como cambiar la definición de Wrapper.



Prácticas





Gracias

¡Nos vemos pronto!