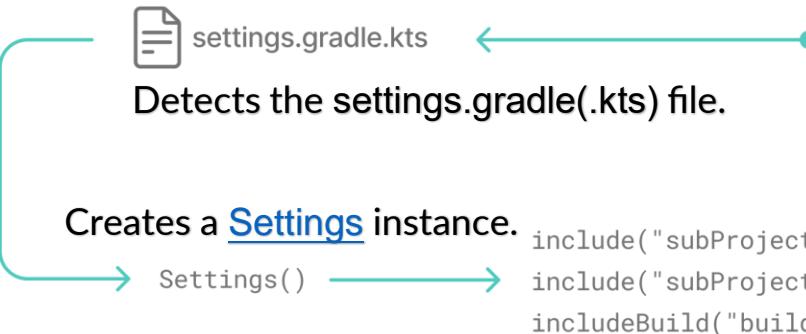


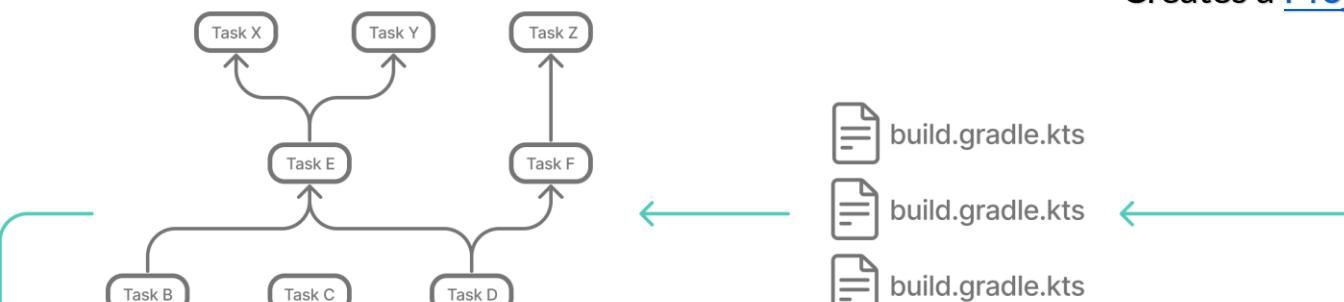
# Gradle: Build Lifecycle

1  
Initialization Phase

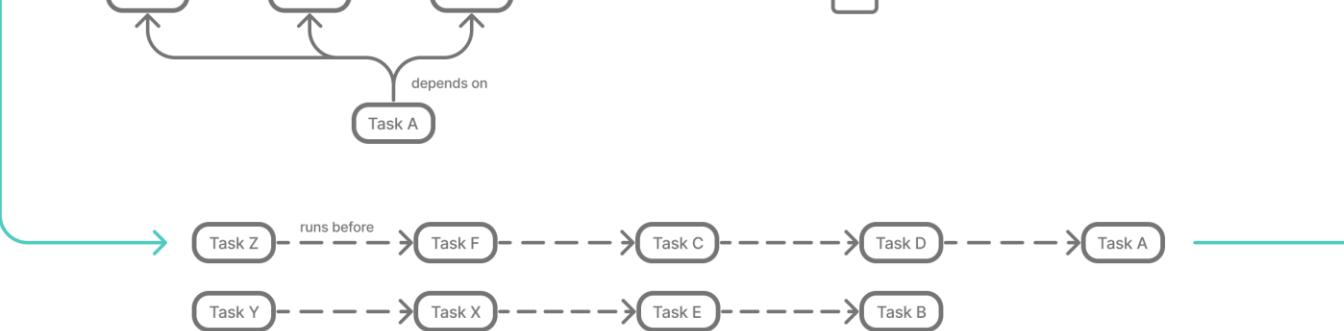


## Build Lifecycle

2  
Configuration Phase



3  
Execution Phase



# Gradle: Build Script

- ☞ Build Script의 Top-level properties and block은 모두 Project API이다.  
→ 즉, Project Object의 Method와 Properties다.

## build.gradle.kts

`println(name)` Build Script 전체에서 name property에 접근 가능하다. → Top-level Reference

`println(project.name)` Project 또한, 위의 name과 마찬가지로 Top-level Reference이다.  
→ Project Object를 return 한다. 따라서, project.name을 통해 name에 접근 가능하다.

# Gradle: Build Script – extra

```
val springVersion by extra("3.1.0.RELEASE")
val emailNotification by extra { "build@master.org" }

sourceSets.all { extra["purpose"] = null }

sourceSets {
    main {
        extra["purpose"] = "production"
    }
    test {
        extra["purpose"] = "test"
    }
    create("plugin") {
        extra["purpose"] = "production"
    }
}
```

## Extra properties extensions

- allow new properties to be added to existing domain objects
- act like maps, allowing the storage of arbitrary key/value pairs.

```
project.ext.set("myProp", "myValue")
assert project.myProp == "myValue"

project.myProp = "anotherValue"
assert project.myProp == "anotherValue"
assert project.ext.get("myProp") == "anotherValue"

project.ext["otherProp"] = "a"
assert project.otherProp == "a"
assert project.ext["otherProp"] == "a"

project.ext {
    myprop = "a"
}
assert project.myprop == "a"
assert project.ext.myprop == "a"

project.myprop = "b"
assert project.myprop == "b"
assert project.ext.myprop == "b"
```