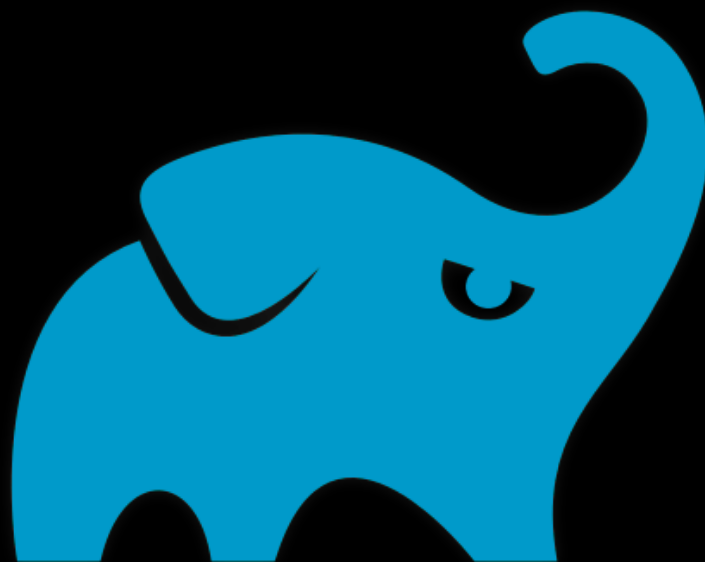




Gradle Mash 2019



+

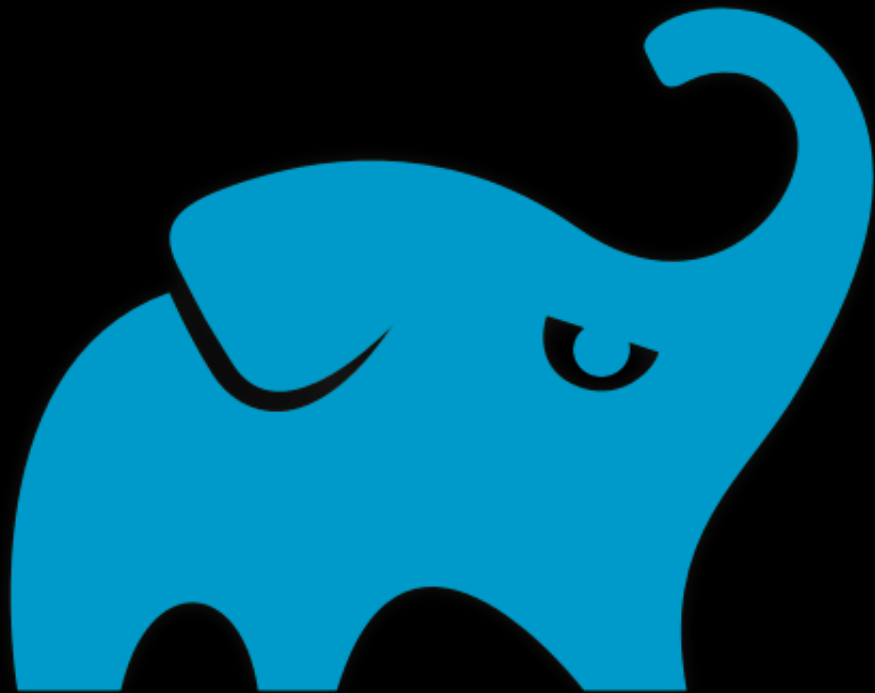
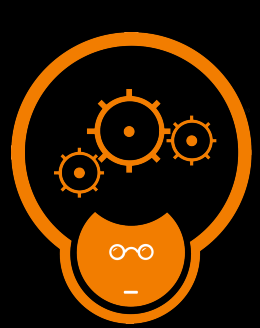


Gradle + Ketchup

Presenter: David Lucas

L
S
E





+



Gradle + Kotlin

Presenter: David Lucas

L
S
E





Who am I ?

- Over 25 years in software industry
- Working with Java since 1998
- IntelliJ / JetBrains introduced me to Kotlin
- Google made me realize ...

Kotlin is now the new Java

- I am a Kotlin Enthusiast
- Extensive production deployments
- Experience with Make, Ant, Maven, and new to Gradle



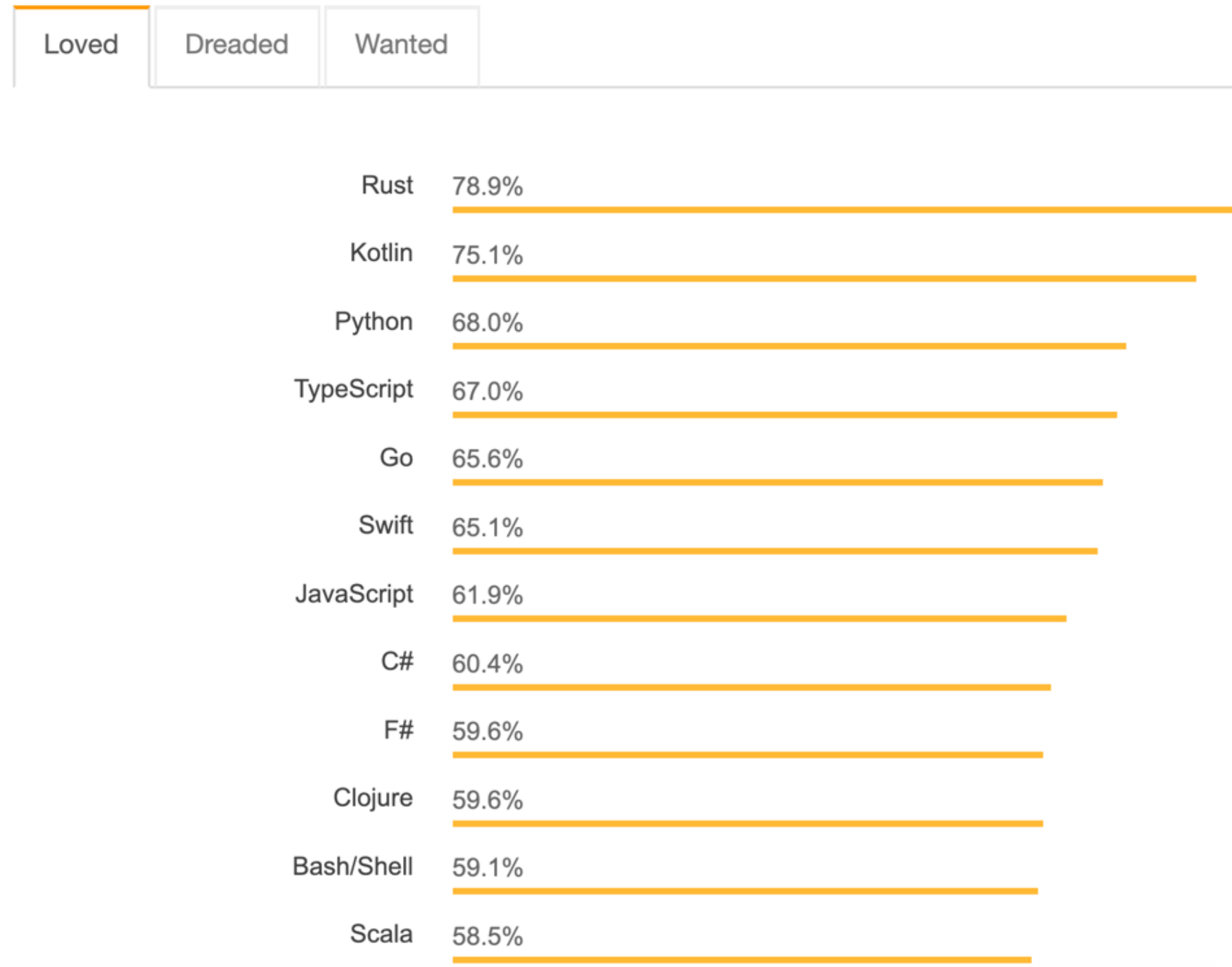
David Lucas
Lucas Software Engineering, Inc.
www.lse.com
ddlucas@lse.com
[@DavidDLucas](https://twitter.com/DavidDLucas)

Goals

- Introduce Gradle
- Introduce Kotlin DSL
- Review Performance Features
- DEMOs

Kotlin Stats

Most Loved, Dreaded, and Wanted Languages



2018

MAKE

In the beginning...
there was Make...
and build tools seemed to be

targets: dependences
commands

[sad, I could not find an image for Make]

Note: Ask me later about evil white space.



tasks, mostly Java Centric



Maven

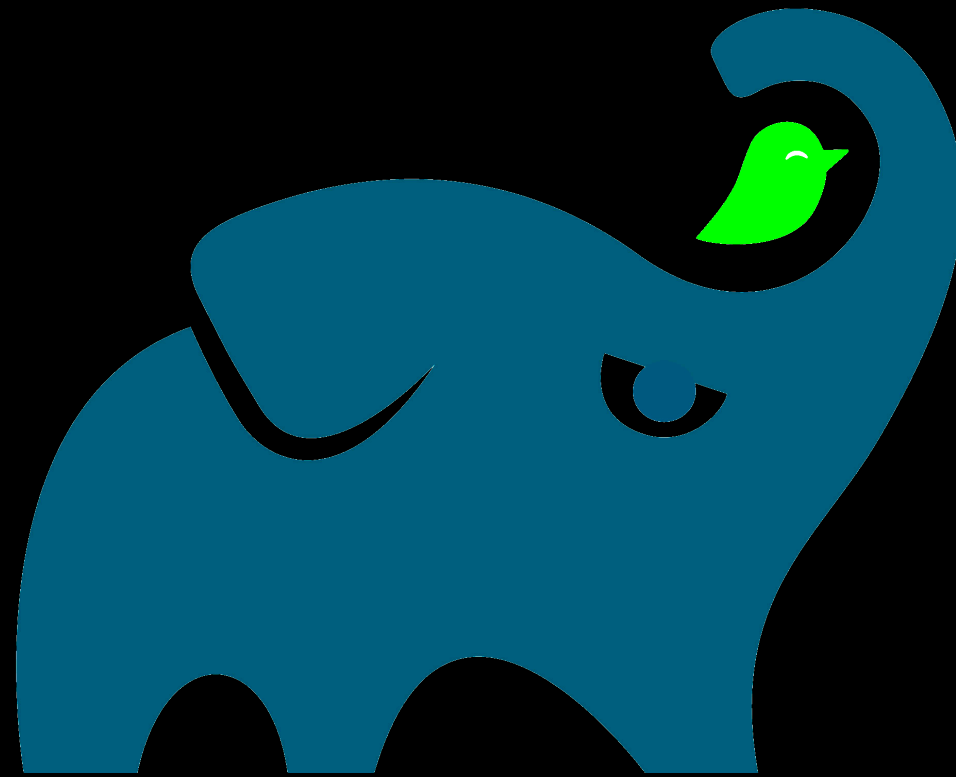
dependencies, lifecycle, plugins,
Java Centric

[rigid]



dependencies, tasks,
Java Centric

[ant + ivy]



Gradle

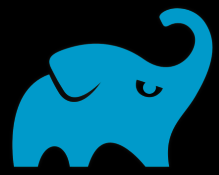
dependencies, tasks,
plugins, cache, multi-platform
targets

[flexible]



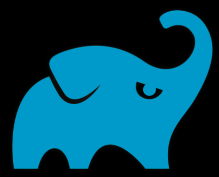
Gradle Intro

- Created in 2007 as an alternative to ant / maven
- Gradle Inc. founded by Hans Dockter
- Goal: allow build authors to write fully declarative models
- Build by convention, avoid XML
- JVM Based Command Line Tool
- Google selected Gradle as Official Build Tool for Android Development



Gradle Intro

- Ecosystem:
 - JVM: Java, Kotlin, Groovy, Scala, Clojure...
 - Native: C, C++ Swift, Rust, ...
 - Mobile: Android, iOS
 - Other: Go, Python, JavaScript, ...



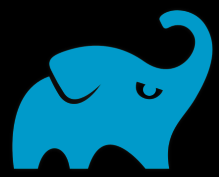
Gradle Intro

- Write Groovy or Kotlin “build scripts”
- Leverage Gradle Tasks and Plugins
- Command line executes tasks
- Tasks are made up of
 - dependencies
 - configuration
 - code
- Code is used to create tasks or plugins



Gradle Intro

- Plugins
 - Core (java, jacoco, maven-publish, ...)
 - Community (kotlin, android, goLang, ...)
 - Contribute Reusable Tasks
 - Configurable Model
 - Configure via Build Script



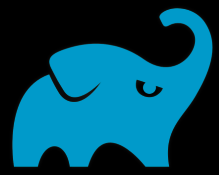
Gradle Features 5.x

- Requires JDK 8 or higher
- Support Groovy 2.5
- Support Kotlin 1.3.11
- Support JUnit 5.x
- Support JDK 11.x
- Support found in multiple IDEs



Gradle Features 5.x

- Improved Incremental Compilation
- Improved Dependency Management
- Signing artifacts
 - Maven Publish
 - IVY Publish
- Properties (Global or By Project)
- Config Avoidance
- Custom Command Line Arguments



Gradle Features 5.x

- Fail Fast Testing

`gradle --fail-fast test` # since 4.6

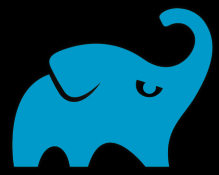
stops on first failure

runs failure first

- Parallel

- Worker API

- Interactive gradle init



Gradle Features 5.x

Process Resource Improvements

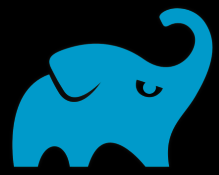
Process Type	Gradle 4.x default heap	Gradle 5.0 default heap
Command-line client	1 GB	64 MB
Gradle Daemon	1 GB	512 MB
Worker processes	1/4 of physical memory	512 MB



Gradle Features 5.x

- Task Timeouts

```
val helloTask by tasks.registering(Test::class) {  
    group = "greeting"  
    description = "hello greeting"  
    doLast {  
        println("Hello World!")  
    }  
    timeout.set(Duration.ofMinutes(5))  
}
```



Gradle Features 5.x

- Test Grouping

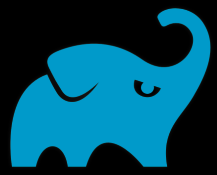
```
tasks.test {  
    useJUnitPlatform {  
        excludeTags("slow")  
        includeEngines("junit-vintage")  
        failFast = true  
    }  
}
```



Gradle Features 5.x

- Dependency Constraints

```
dependencies {  
    implementation("org.apache.httpcomponents:httpclient")  
    constraints {  
        add("implementation",  
            "org.apache.httpcomponents:httpclient:4.5.3")  
        {  
            because("previous versions bug impacts application")  
        }  
        add("implementation", "commons-codec:commons-codec:1.11")  
        {  
            because("version 1.9, pulled from httpclient, has bugs")  
        }  
    }  
}
```



Gradle Features 5.x

- Dependency Version Alignment

```
open class JacksonAlignmentRule: ComponentMetadataRule {  
    override fun execute(ctx: ComponentMetadataContext) {  
        ctx.details.run {  
            if (id.group.startsWith("com.fasterxml.jackson")) {  
                belongsTo("com.fasterxml.jackson:jackson-platform:${id.version}")  
            }  
        }  
    }  
}
```



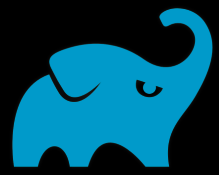
```
dependencies {  
    components.all(JacksonAlignmentRule::class.java)  
}
```



Gradle Features 5.x

- Bill Of Materials (BOM) Support

```
dependencies {  
    // import a BOM  
    implementation(  
        platform("org.springframework.boot:spring-boot-  
dependencies:1.5.8.RELEASE")  
    )  
    // define dependencies without versions  
    implementation("com.google.code.gson:gson")  
    implementation("dom4j:dom4j")  
}
```

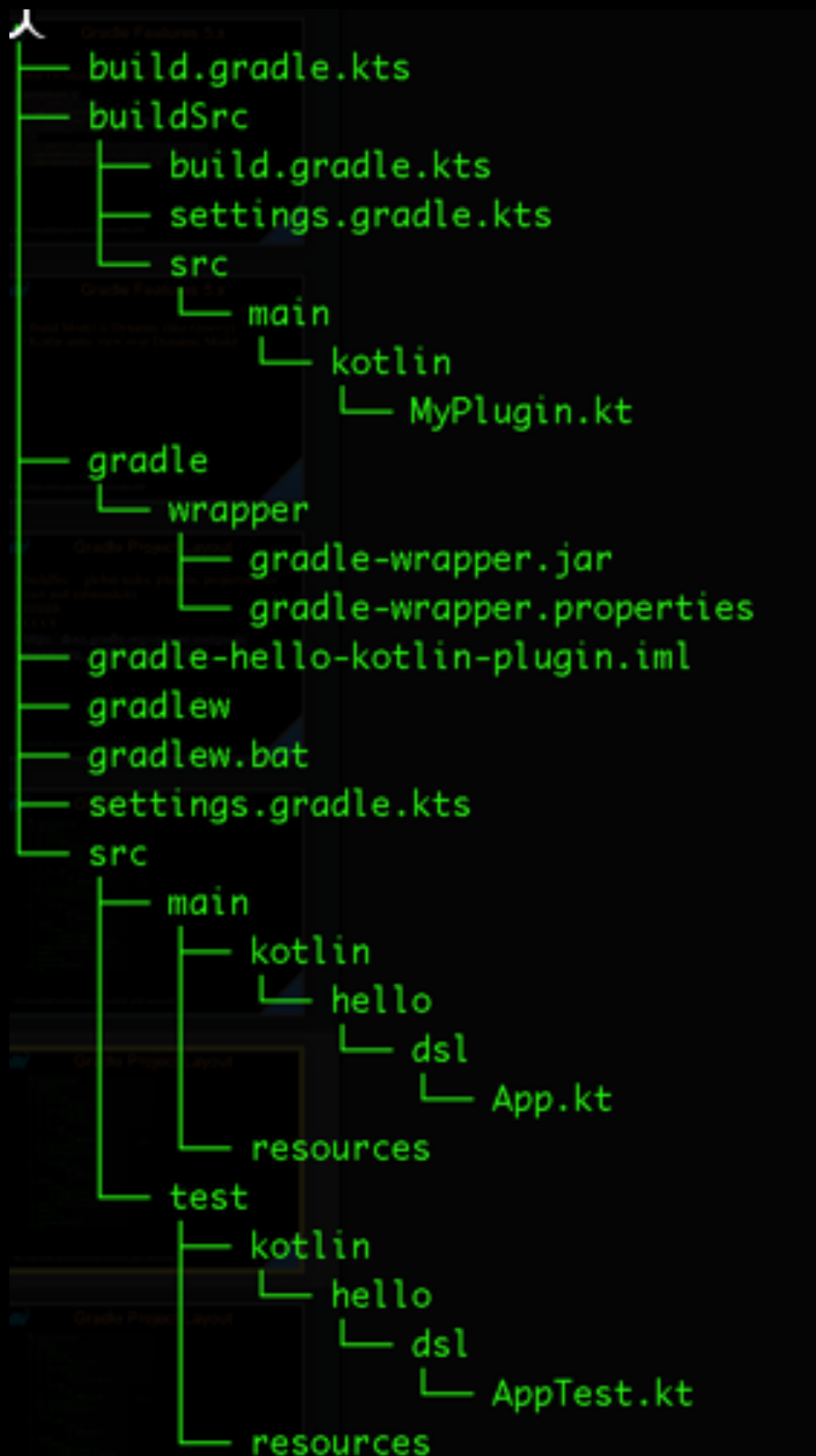


Gradle Features 5.x

- Build Model is Dynamic (like Groovy)
- Kotlin static view over Dynamic Model



Gradle Project Layout



Kotlin Project

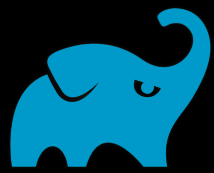
- https://docs.gradle.org/current/userguide/organizing_gradle_projects.html



Gradle Project Layout



Root Kotlin DSL Scripts

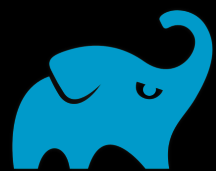


Gradle Project Layout

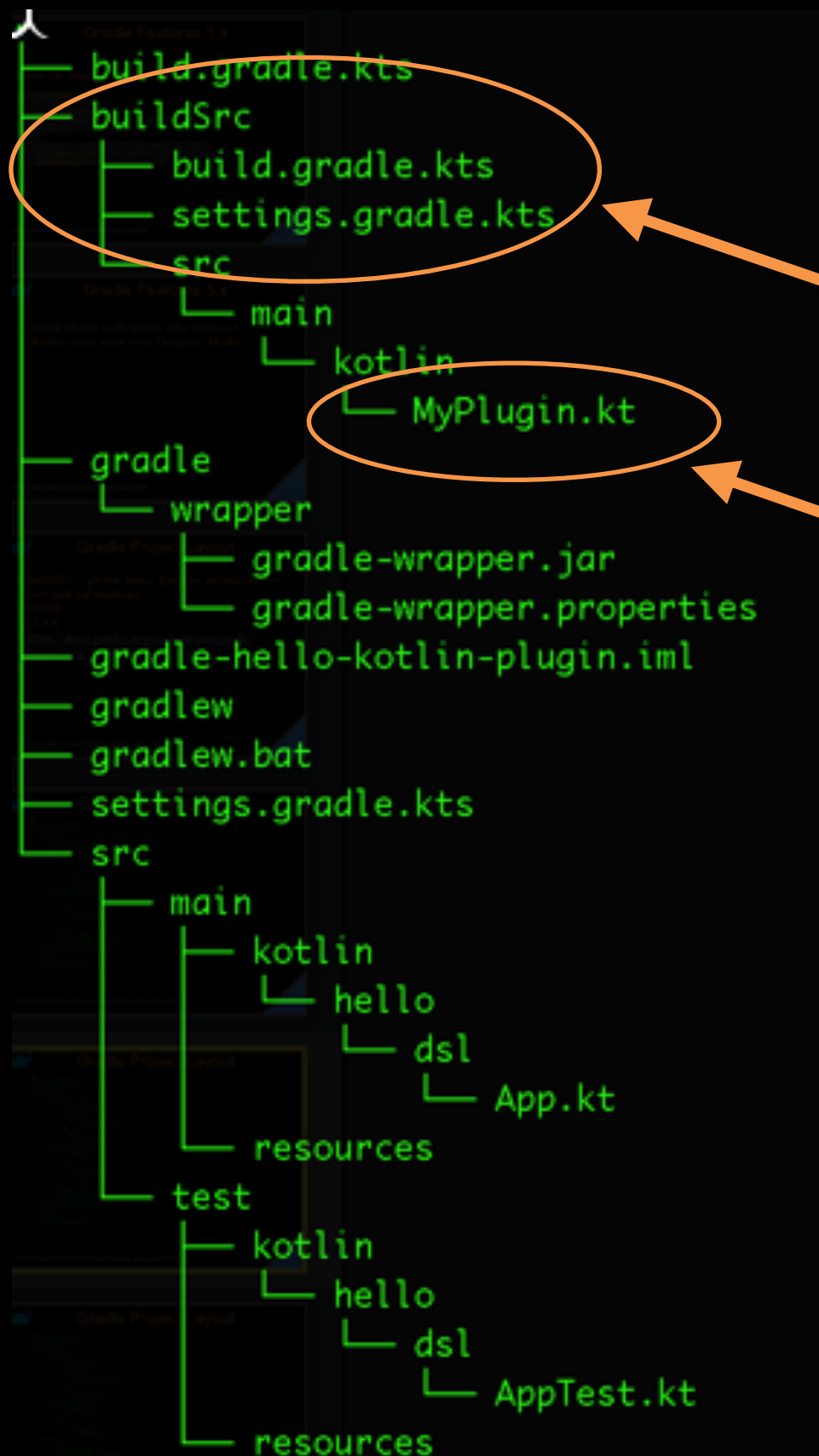


Project Object

Settings Object

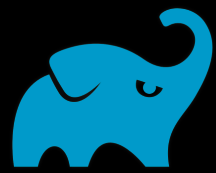


Gradle Project Layout

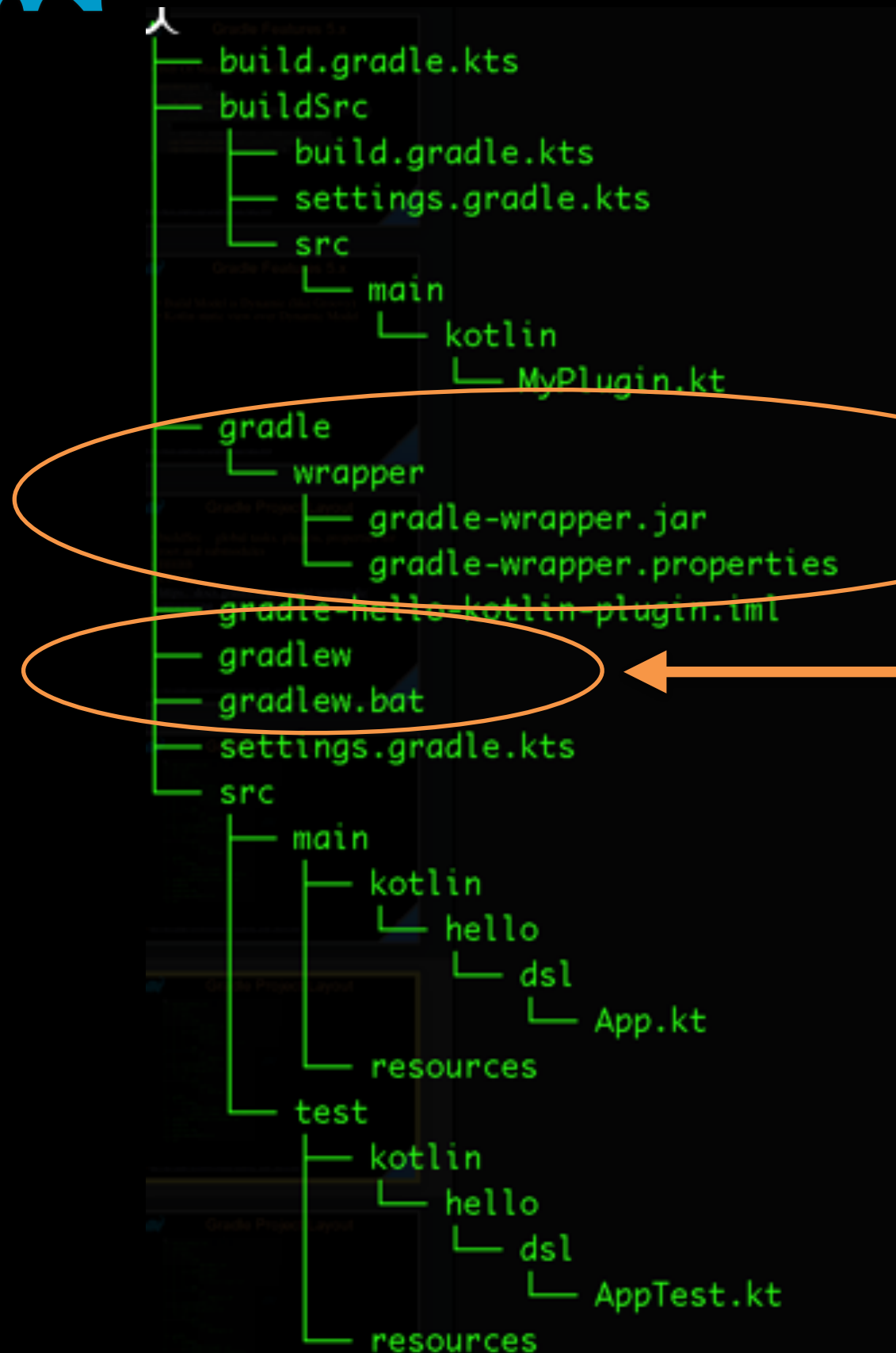


Global Kotlin DSL Scripts

Global Plugin Source



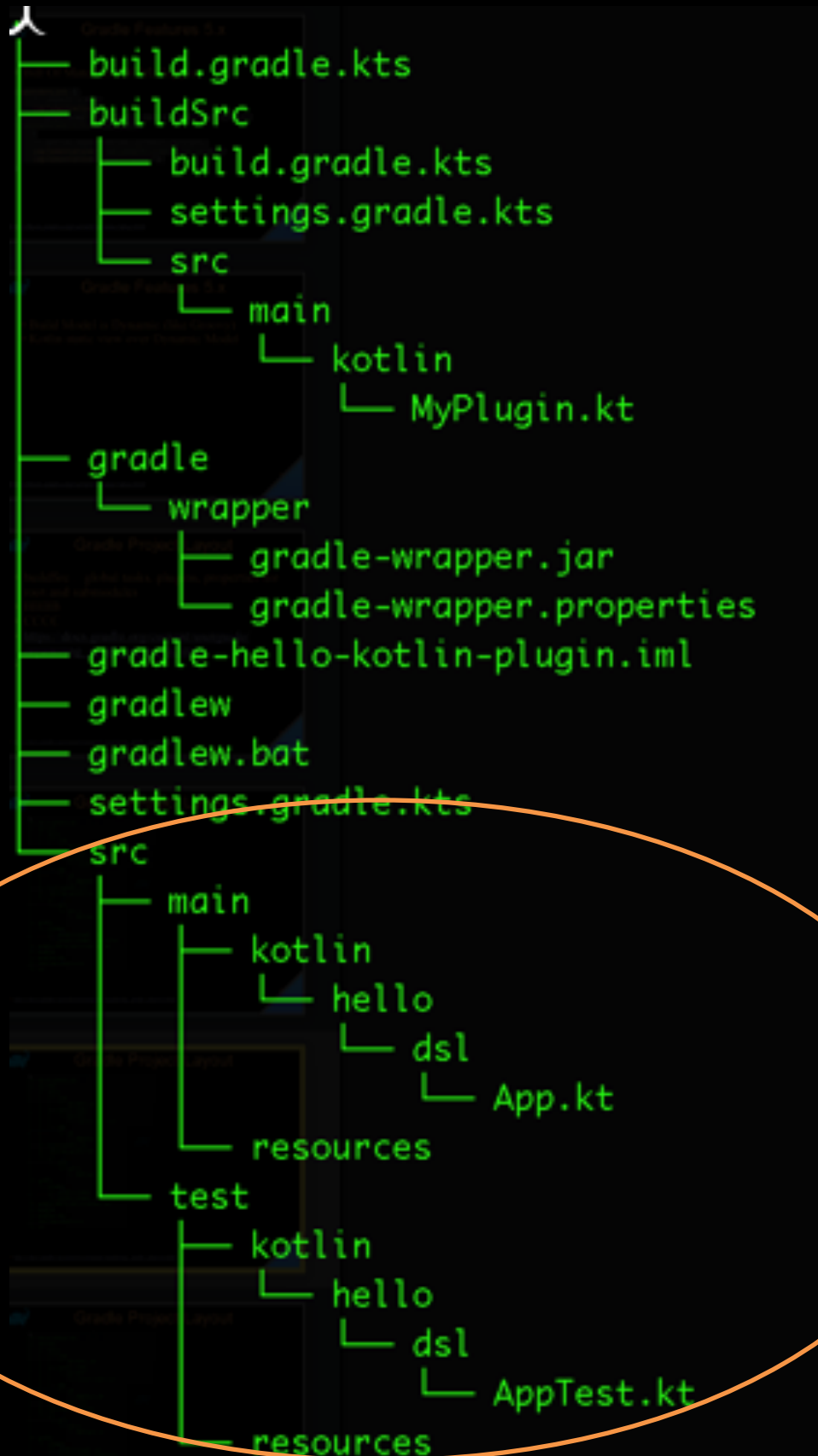
Gradle Project Layout



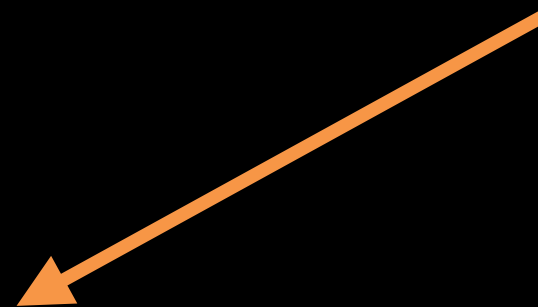
Gradle Wrapper

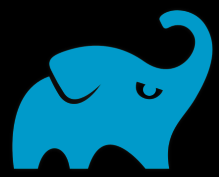


Gradle Project Layout



Project Source





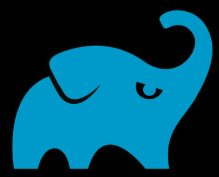
Gradle Upgrading to 5.x

- Follow migration guide
- Minimum of JDK 1.8
- `gradle help --scan` # report shows deprecation
- Upgrade wrappers to latest
`gradle wrapper --gradle-version 5.1`



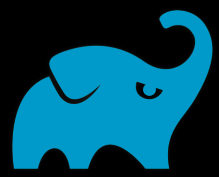
Gradle Upgrading to 5.x

- Plugins most likely need upgraded
 - Shadow plugin at least 2.x or higher
 - Some plugins are now part of Core
 - Checkstyle Plugin now uses 8.12 by default
 - CodeNarc Plugin now uses 1.2.1 by default.
 - JaCoCo Plugin now uses 0.8.2 by default.
 - PMD Plugin now uses 6.8.0 by default.



Gradle Upgrading to 5.x

- Groovy now 2.5.4.
- Ant now 1.9.13.
- Ivy has been upgraded from 2.2.0 to 2.3.0.
- Maven/Ivy Repository Access
 - AWS SDK now 1.11.407.
 - Google Cloud Storage JSON API Client now v1-rev136-1.25.0.
 - Maven Wagon now 3.0.0.



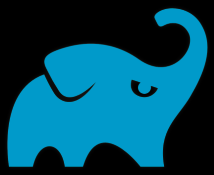
Gradle Upgrading to 5.x

- BND library used by the OSGi Plugin now 4.0.0.
- JUnit Platform now 1.3.1.
- SLF4J now 1.7.25.



Gradle Command Line

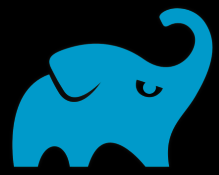
- `gradle --help`
- `gradle init help`
- `gradle tasks --all`
- `gradle clean build`
- `gradle test deploy`
- `gradle dist --exclude-task test`
- `gradle test --rerun-tasks`
- `gradle projects`
- `gradle dependencies`
- `gradle properties`



Gradle Groovy DSL



- Dynamic
- By-passes boilerplate code (get/set)
- <https://docs.gradle.org/current/dsl/index.html>



Gradle Kotlin DSL



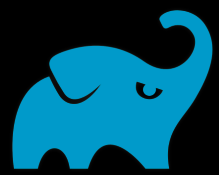
- Kotlin DSL 1.1
- Type Safety
 - Improved Build Logic
 - Improved Troubleshooting Build Issues
- File Extensions
 - .kt = kotlin compiler code
 - .kts = kotlin scripting
 - .gradle.kts = Gradle Kotlin DSL script
 - .gradle = Gradle Groovy DSL script



Gradle Kotlin DSL



- Improved experience with tooling (IDE)
 - Static Typed Language
 - Code Completion
 - Error Highlighting
 - Quick Documentation
 - Refactoring



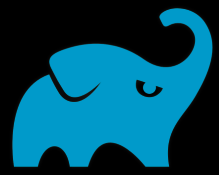
Gradle Kotlin DSL IDE Support



Build import	Syntax highlighting 1	Semantic editor 2	
IntelliJ IDEA	✓	✓	✓
Android Studio	✓	✓	✓
Eclipse IDE	✓	✓	✗
CLion	✓	✓	✗
Apache Netbeans	✓	✓	✗
Visual Studio Code (LSP)	✓	✓	✗
Visual Studio	✓	✗	✗

1 Kotlin syntax highlighting in DSL scripts

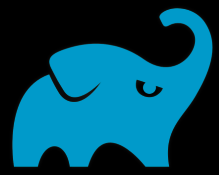
2 Code completion, jump to source, documentation, refactoring in DSL



Gradle Kotlin DSL



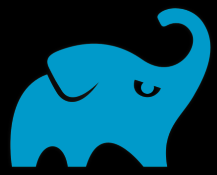
- Less Runtime Surprises
- Faster feedback loop
- Dynamism at runtime has a cost
- Extensions to Gradle API for Kotlin
 - `KClass<T>`
 - `reified<T>()`
 - Delegated properties
- Kotlin was the missing link. — Hans Dockter



Gradle Kotlin DSL



- https://docs.gradle.org/current/userguide/kotlin_dsl.html
- <https://blog.gradle.org/kotlin-meets-gradle>
- <https://try.kotlinlang.org>
- <https://kotlinlang.org/docs/tutorials/koans.html>
- <https://kotlinlang.org/docs/tutorials/edu-tools-learner.html>



Groovy to Kotlin DSL



- Convert single quote to double quote
- Move plugins into plugins block
- Methods: replace space with parenthesis
- Make implicit conversions explicit
`maven { url = uri("http://....") }`
- Use `withType` for mapping into project tasks

```
tasks.withType<JavaCompile> {  
    options.encoding = "UTF-8"  
}
```

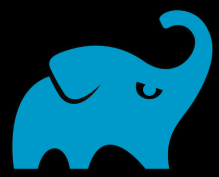
- <https://guides.gradle.org/migrating-build-logic-from-groovy-to-kotlin/>



Groovy to Kotlin DSL

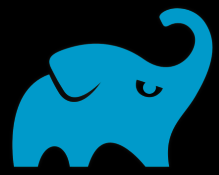


Description	Before	After
Replace all ' with "	kotlin-android'	"kotlin-android"
Replace "def " with "val "	def appcompat = "1.0.0"	val appcompat = "1.0.0"
Convert plugins	apply plugin: "kotlin-kapt"	apply(plugin = "kotlin-kapt")
Add () to dependencies	implementation ":epoxy"	implementation(":epoxy")
Convert Maven	maven { url " https://jitpack.io " }	maven(" https://jitpack.io ")
Convert Sdk Version	compileSdkVersion 28	compileSdkVersion(28)
Convert Version Code	versionCode 4	versionCode = 4
Convert Build Types	debuggable true	isDebuggable = true
Convert proguardFiles	proguardFiles getDef..., "..."	setProguardFiles(listOf(getDef..., "..."))
Convert sourceCompatibility	sourceCompatibility = "1.8"	sourceCompatibility = JavaVersion.VERSION_1_8
Convert AndroidExtensions	androidExtensions { experimental = true }	androidExtensions { isExperimental = true }
Convert include	include ":app", ":diffutils"	include(":app", ":diffutils")
Convert signingConfigs	signingConfigs { debug { ... } }	signingConfigs { register("debug") { ... } }
Convert buildTypes	buildTypes { debug { ... } }	buildTypes { named("debug") { ... } }



Gradle From Scratch: DEMO

- <https://gradle.org/install>
- add to Path
- mkdir hello-world
- cd hello-world
- git init help



Gradle Plugins: DEMO

- Plugins can be loaded from jars or classpath
- buildSrc
 - root gradle source code folder
 - properties, tasks, and plugins defined here
 - visible to all submodules

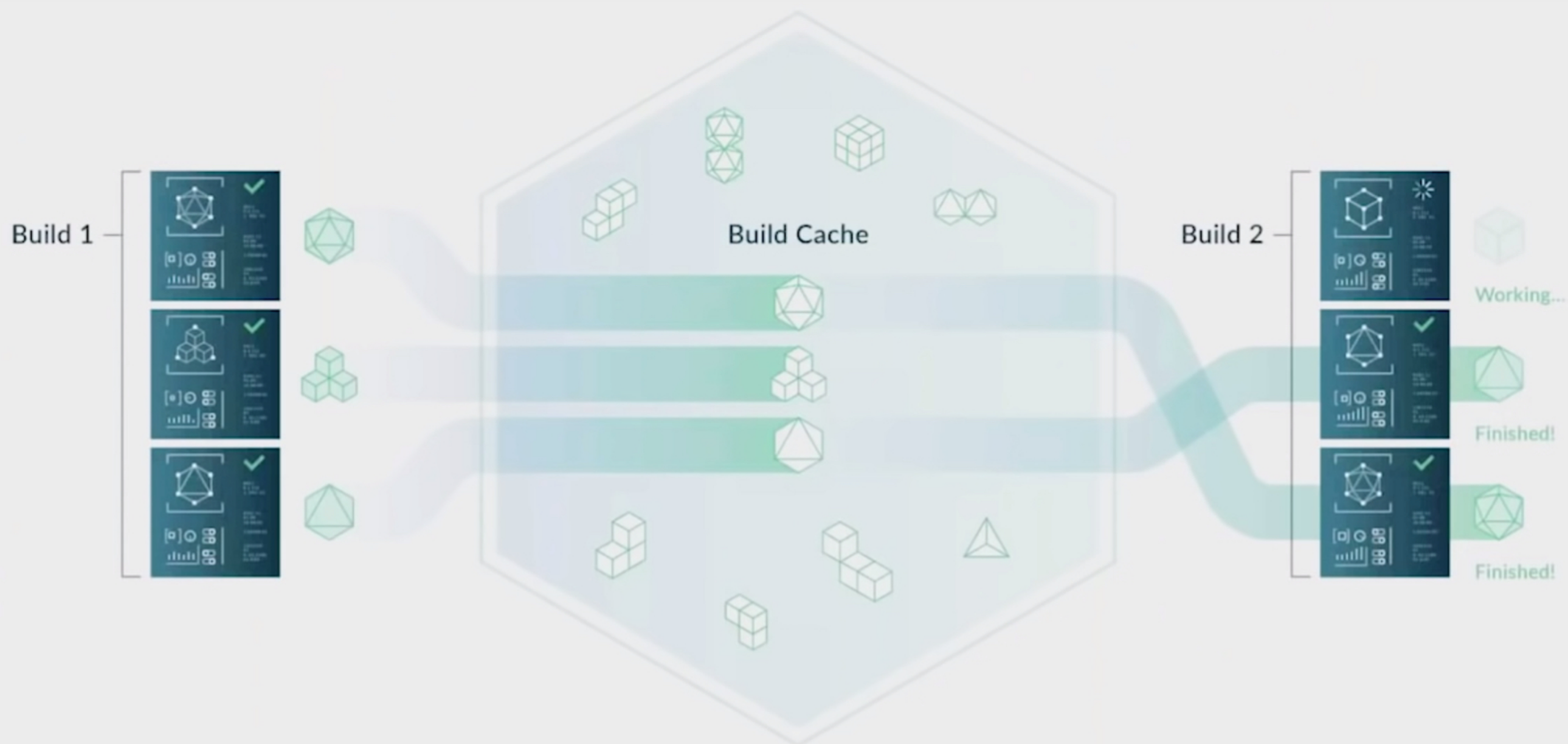


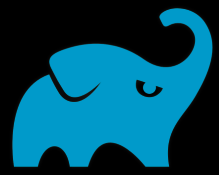
Gradle From Maven: DEMO

- <https://gradle.org/maven-vs-gradle>
- `gradle init` # converts maven to groovy dsl
- plugins may be missing, will need to check
- `./gradlew build` # compiles, tests, and packages



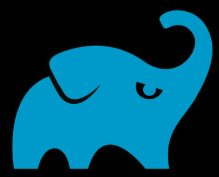
Gradle Build Cache





Gradle Build Cache

- Larger the build, the bigger potential of improvement
- Cache improves incremental builds
- Cache of inputs and outputs of Tasks
- Build Cache
 - Basic (free) is File Based Cache
 - Enterprise (pay) is Remote Server Cache w/extras
 - Plugin speed will improve, like Docker
 - Enterprise tracks performance history







Gradle Build Cache

- File Cache (local) on by default
- Shared Cache (remote) uses Hazelcast
- Plugins are opting in to cache (Checkstyle, etc)
- https://docs.gradle.org/current/userguide/build_cache.html
- P.S. Guess what is coming to Maven?
https://gradle.org/training/maven-builds-with-gradle-enterprise/?time_value=maven-builds-with-gradle-enterprise-2019-01-23





Gradle Build Scans


 Build Scan


✓ commons-lang3 buildEnvironment Jan 8, 2019 10:10:33 PM EST   


Summary


 Console log


 Timeline


 Performance

 Projects

 Dependencies


 Plugins

 Switches

 Infrastructure

Started today at 10:10:33 PM EST, finished today at 10:10:41 PM EST
Gradle 5.1, Build scan plugin 2.1

[Explore console log](#)

 1 task executed in 1 project in 7.952s

Initialization & configuration

:buildEnvironment0.012s

[Explore timeline](#)

8 sec total build time

Initialization & configuration7.919s

Execution0.033s

[Explore performance](#)

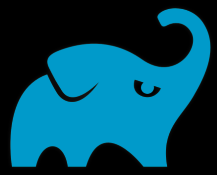
1 project: commons-lang3

50 L S E

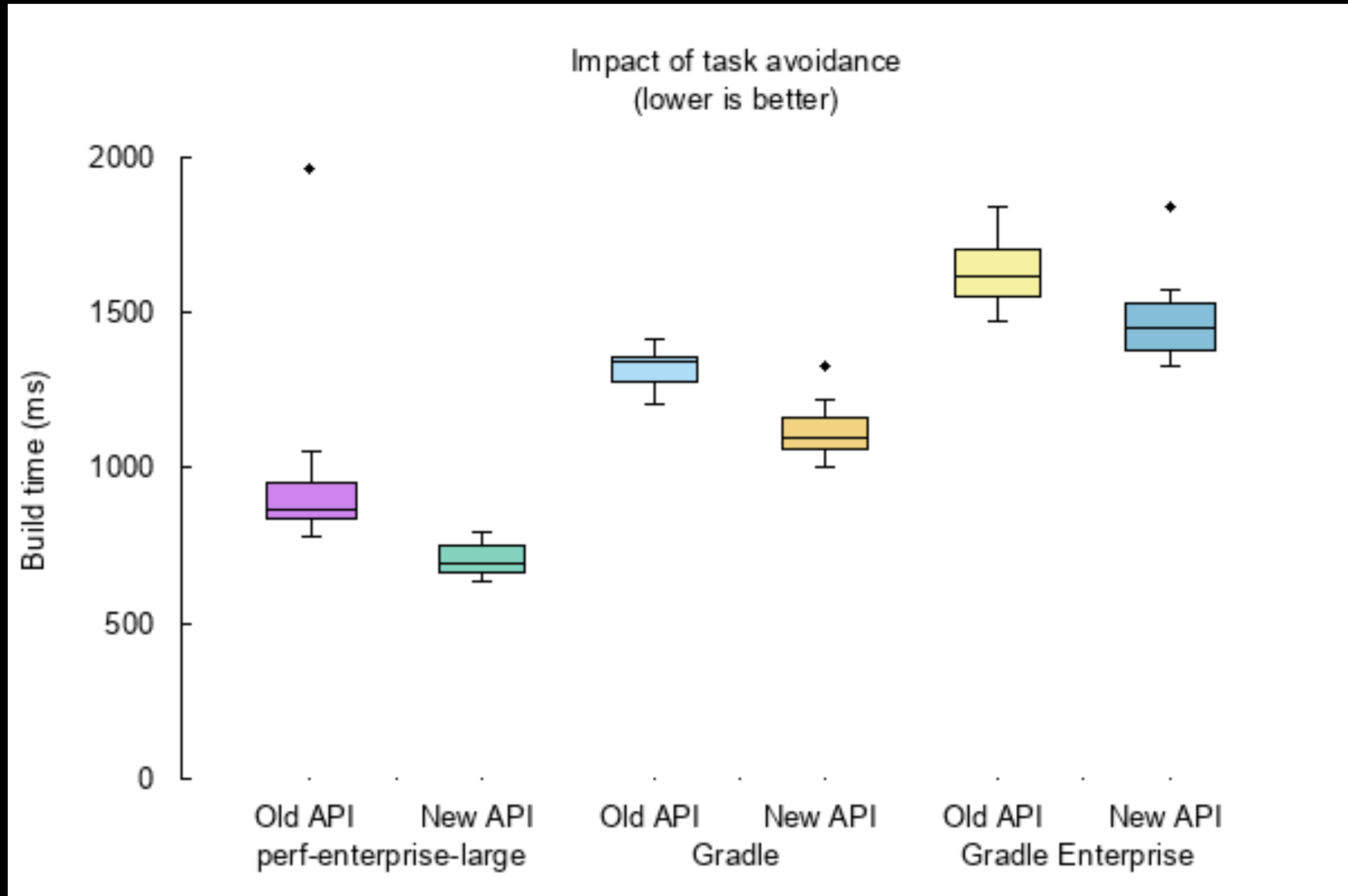


Gradle Build Scans

- <https://scans.gradle.com>
- Cloud based performance report
- Example:
<https://scans.gradle.com/s/n4mvpcjm4ms4k/>



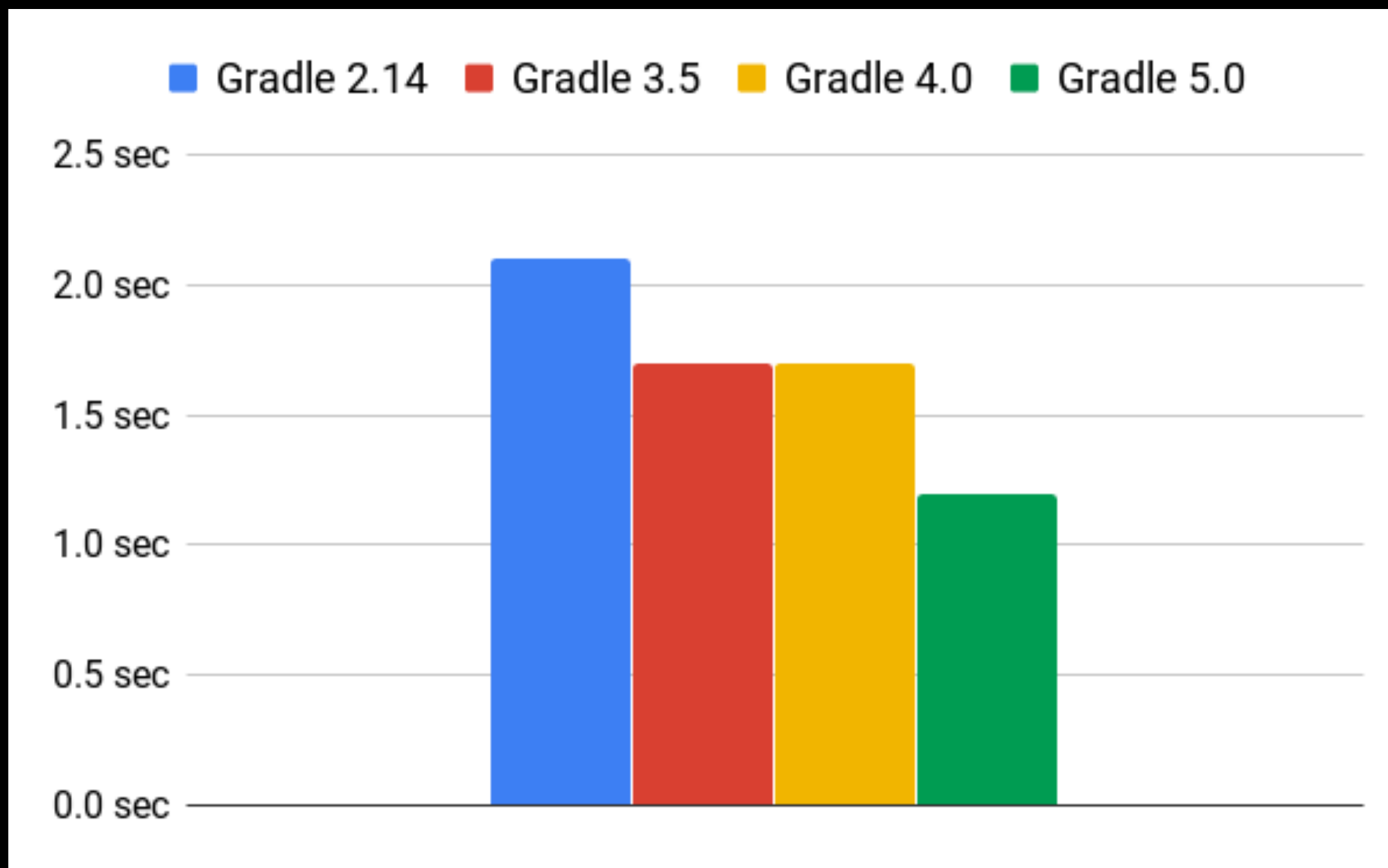
Gradle Performance



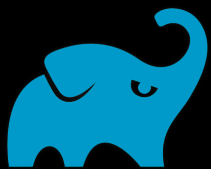
Configuration Avoidance



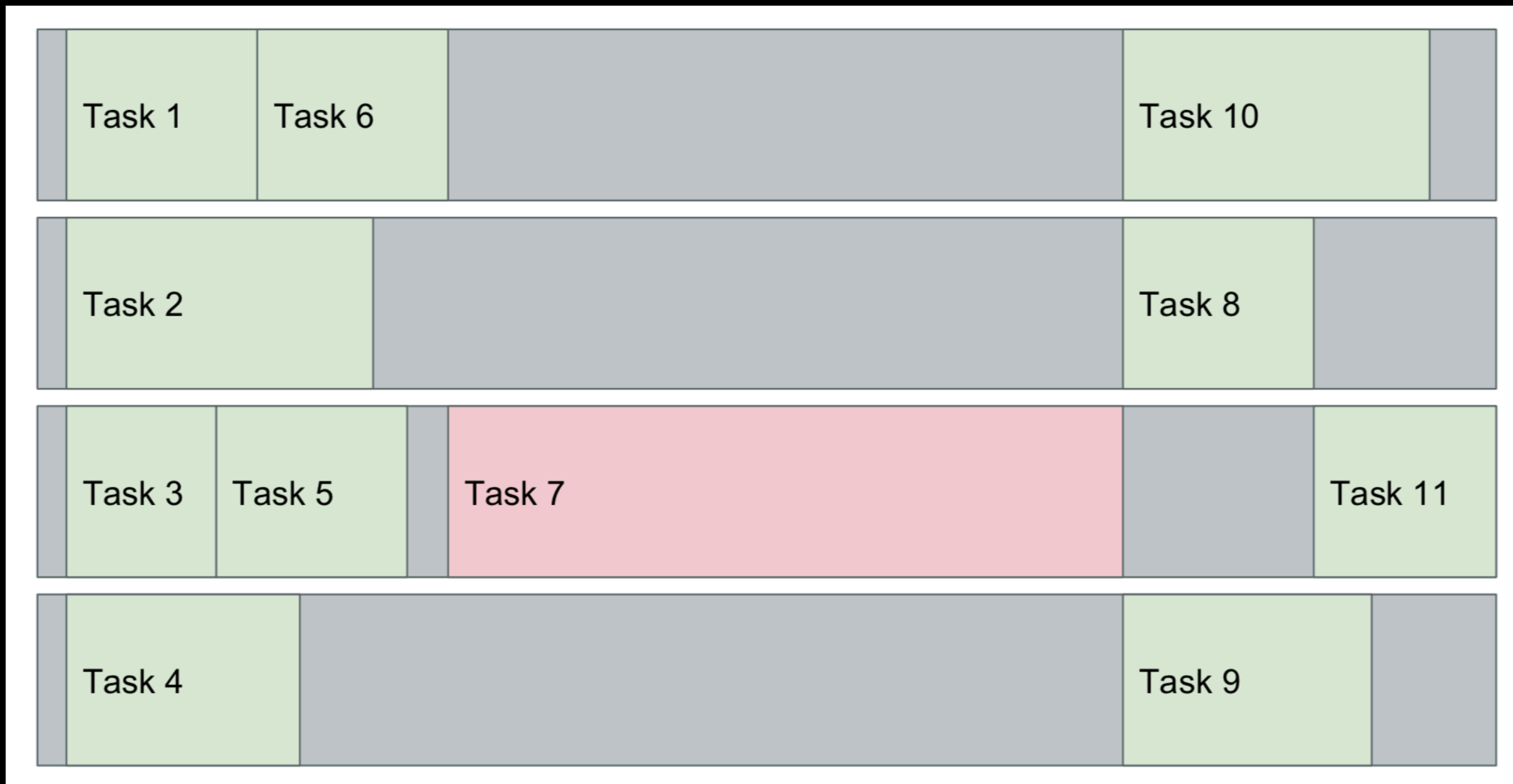
Gradle Performance



Incremental Compile 1000
module Java Project



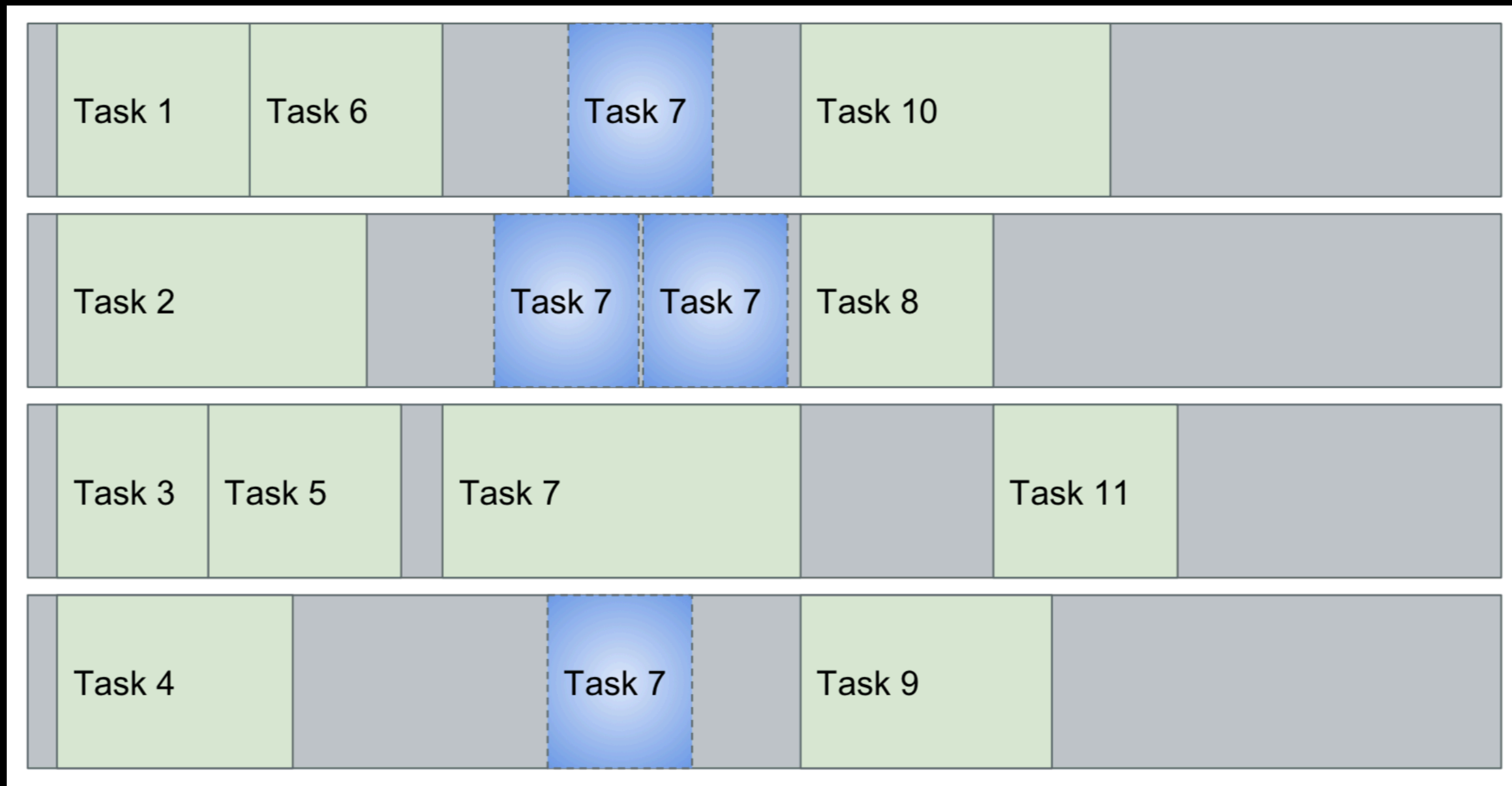
Gradle Performance



Without Worker API



Gradle Performance

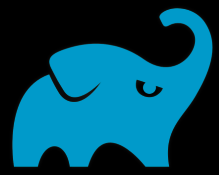


With Worker API



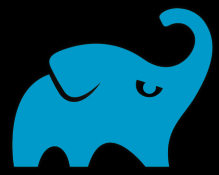
Gradle Performance: DEMO

- Build Cache
- Profile
- Scans
- <https://gradle.org/whats-new/gradle-5/#new-gradle-invocation-options>



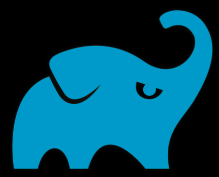
Gradle Summary

- Time is money, waiting is deferred feedback
- Multi-platform support
- Extensible
- Flexible
- Open Source
- I get to use Kotlin more
- Enterprise Edition provides even more



Gradle Resources

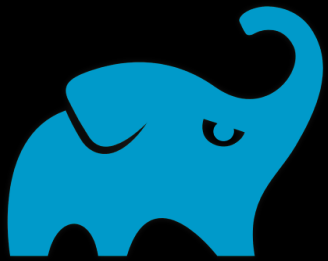
- <http://gradle.org>
- <http://gradle.org/guides>
- <https://gradle.org/training>
- <https://gradle.org/maven-vs-gradle/>
- <https://bmuschko.com/blog/jenkins-build-pipeline/>
- <https://container-solutions.com/how-to-build-docker-images-with-gradle/>



Gradle Kotlin DSL Resources



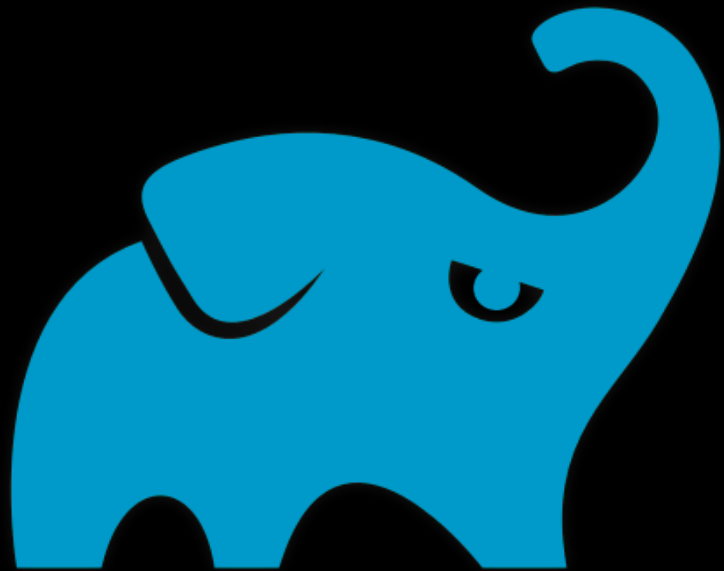
- https://docs.gradle.org/current/userguide/kotlin_dsl.html
- <https://blog.gradle.org/kotlin-meets-gradle>
- <https://try.kotlinlang.org>
- <https://kotlinlang.org/docs/tutorials/koans.html>
- <https://kotlinlang.org/docs/tutorials/edu-tools-learner.html>



Questions ?

<https://github.com/lseinc/gradle-ketchup-cm19.git>

Don't forget to fill
out the survey!



Thank You !

<https://github.com/lseinc/gradle-ketchup-cm19.git>



David Lucas
Lucas Software Engineering, Inc.
www.lse.com
ddlucas@lse.com
[@DavidDLucas](https://twitter.com/DavidDLucas)

L
S
E

