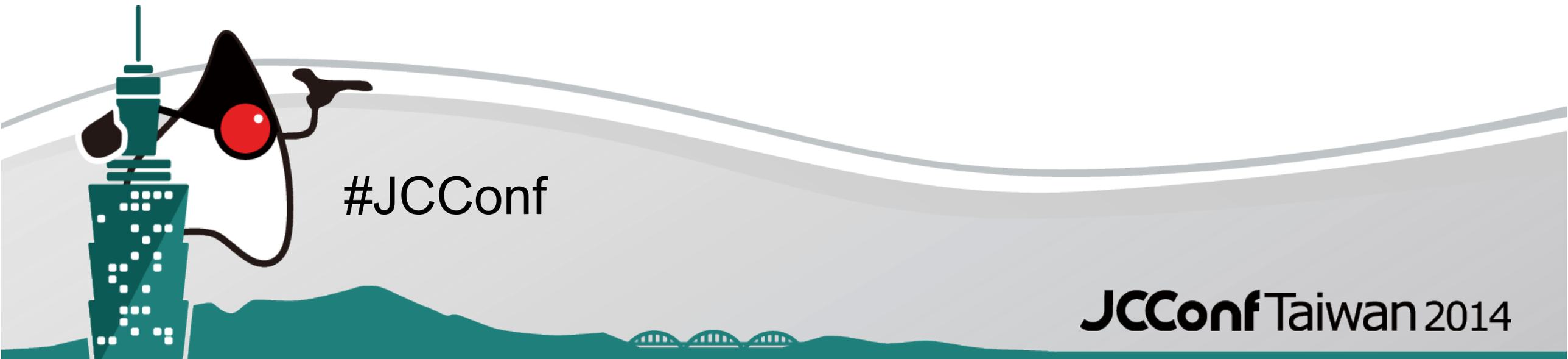
Gradle 起步走: 以 CLI Application 為例

陸振恩 (popcorny)
popcorny@cacafly.com



Outline

- What is Gradle
- Basic Concept
- Build Simple CLI Application by Gradle

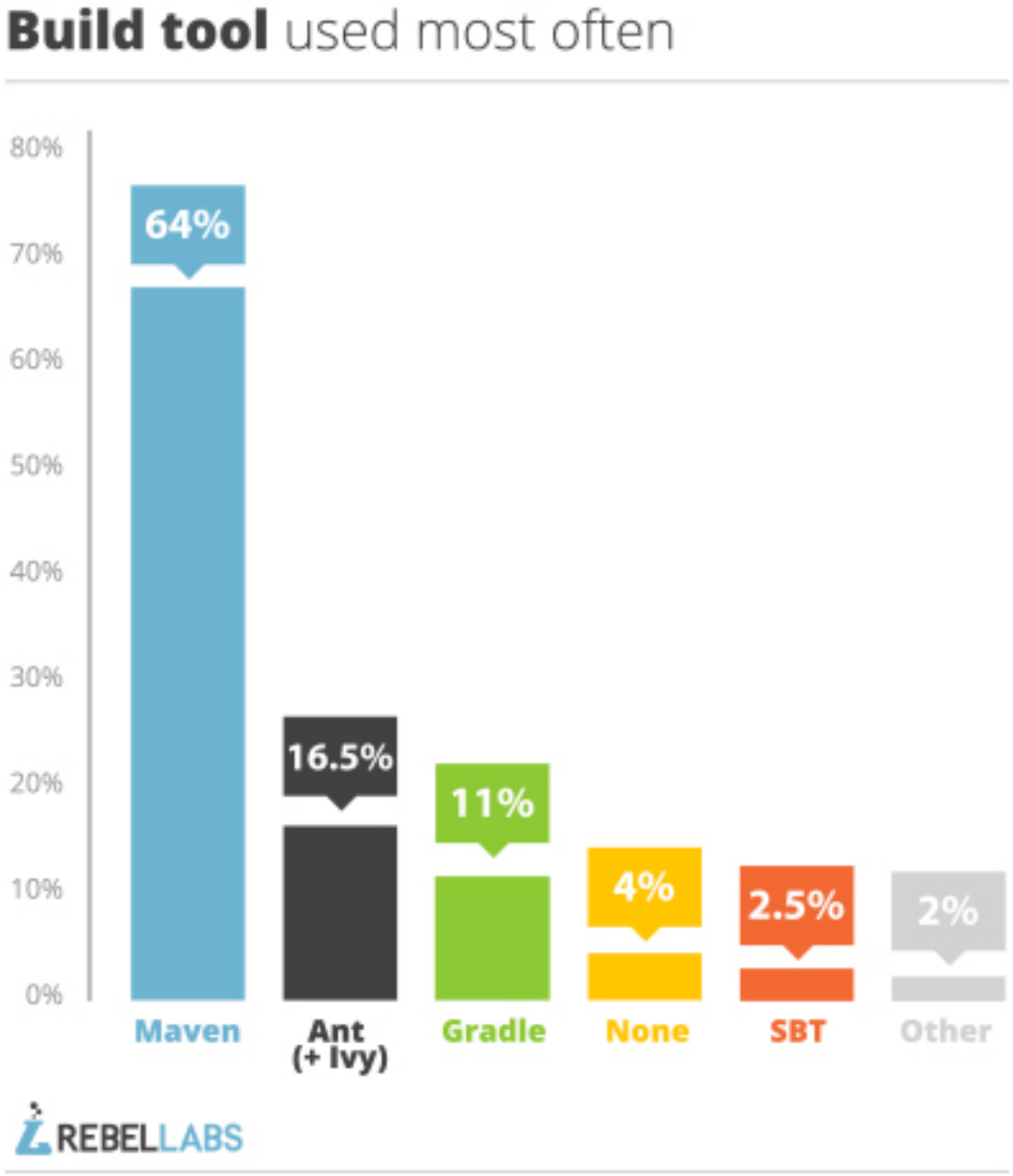
What is Gradle?

#JCConf

JCConf Taiwan 2014

鬼島?

Gradle is a Build Tools



source:
http://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-for-2014/7/

Which build tool would you like to learn more about?

REBELLABS

source :
http://zeroturnaround.com/rebellabs/java-tools-and-technologies-landscape-for-2014/7/

- Features
  - '*make*' in the java ecosystem
  - Pure java
  - Target dependency
- Basic Components
  - Project *(build.xml)*
  - Target *(eg. build, clean, deploy, …)*
  - Task *(javac, jar, copy, …)*

- Features
  - Project Object Model
  - Common Build Lifecycle
  - Convention over Configuration
  - Dependency Management
- Basic Components
  - Project *(pom.xml)*
  - Lifecycle, Phase, Goal
  - Plugin


```
[popcorny@MBP /tmp]$ tree myapp/
myapp/
├── pom.xml
└── src
    ├── main
    │   └── java
    │       └── tw
    │           └── jcconf
    │               └── App.java
    └── test
        └── java
            └── tw
                └── jcconf
                    └── AppTest.java
```

- Features
  - **Groovy-based DSL (Domain Specific Language)**
    - Not XML configuration any more
  - **Dependency Management**
    - Integrate Maven repository
  - **Task Dependency**
    - Like target dependency in ant
  - **Plugins provide the out-of-the-box tasks**
    - Like lifecycle in maven
- Basic Components
  - Project *(build.gradle)*
  - Task
  - Plugin

Gradle起步走

gradle init --type=java-library

gradle project

```
[popcorny@MBP /tmp/gradle-test]$ tree
.
├── build.gradle
├── gradle
│   └── wrapper
│       ├── gradle-wrapper.jar
│       └── gradle-wrapper.properties
├── gradlew
├── gradlew.bat
├── settings.gradle
└── src
    ├── main
    │   └── java
    │       └── Library.java
    └── test
        └── java
            └── LibraryTest.java
```

build script

gradle wrapper

multi-project settings

source code

build.gradle

```
apply plugin: 'java'

repositories {
    mavenCentral()
}

dependencies {
    compile 'org.slf4j:slf4j-api:1.7.5'
    testCompile 'junit:junit:4.11'
}
```

Much Simpler Than Maven!!

Useful Tasks

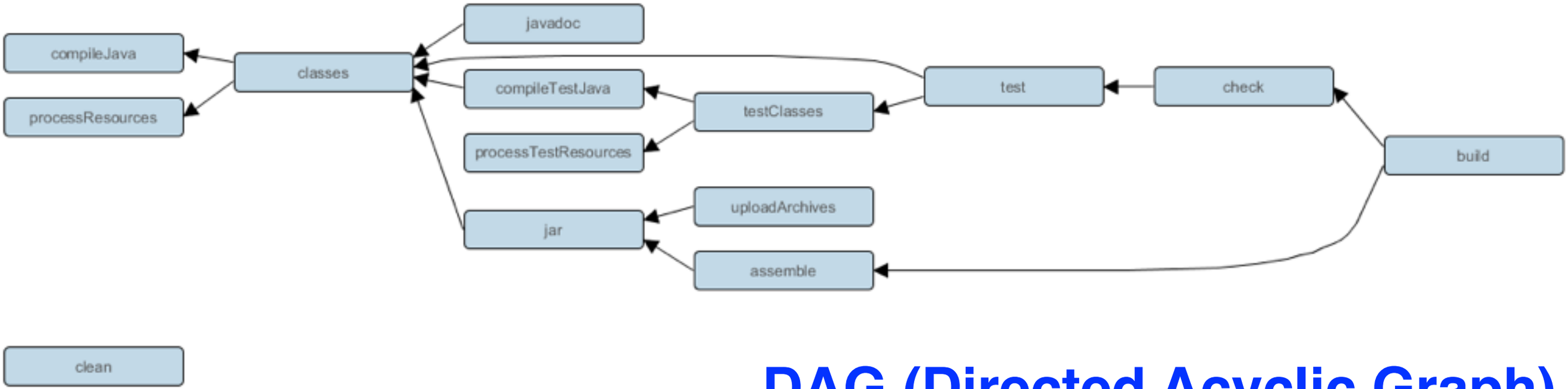
[Default]

- **gradle tasks**
  - List the tasks
- **gradle dependencies**
  - Display the library dependencies.

[Java Plugin]

- **gradle build**
  - Build the project
- **gradle test**
  - Test the project
- **gradle jar**
  - Generate the jar file
- **gradle clean**
  - Clean the project.

Task Dependency (Java Plugin)



DAG (Directed Acyclic Graph)

Custom Tasks

```
task hello << {
    println "hello"
}

task hello2(dependsOn: 'hello') << {
    println "hello2"
}

task hello3(type: Exec) {
    commandLine '/bin/echo', 'hello3'
}
```

Dependency Management

- Simple dependency

```
compile 'org.slf4j:slf4j-api:1.7.5'
testCompile 'junit:junit:4.11'
```

- Dynamic version

```
compile 'org.slf4j:slf4j-api:1.7.+'
```

- Changing module

```
compile 'tw.jcconf:myapp:1.0-SNAPSHOT'
```

- Usually, we can find libraries from http://mvnrepository.com/

Multi Project

- settings.gradle

```
include 'project1','project2'
```

- Run tasks in subproject

```
> gradle project1:build

> cd project1
project1> gradle build
```

- Dependency

```
compile project(':project1')
```

IDE Integration

Gradle IDE plugin

or

IDE Gradle plugin

IDE Integration (cont.)

- Gradle IDE plugin

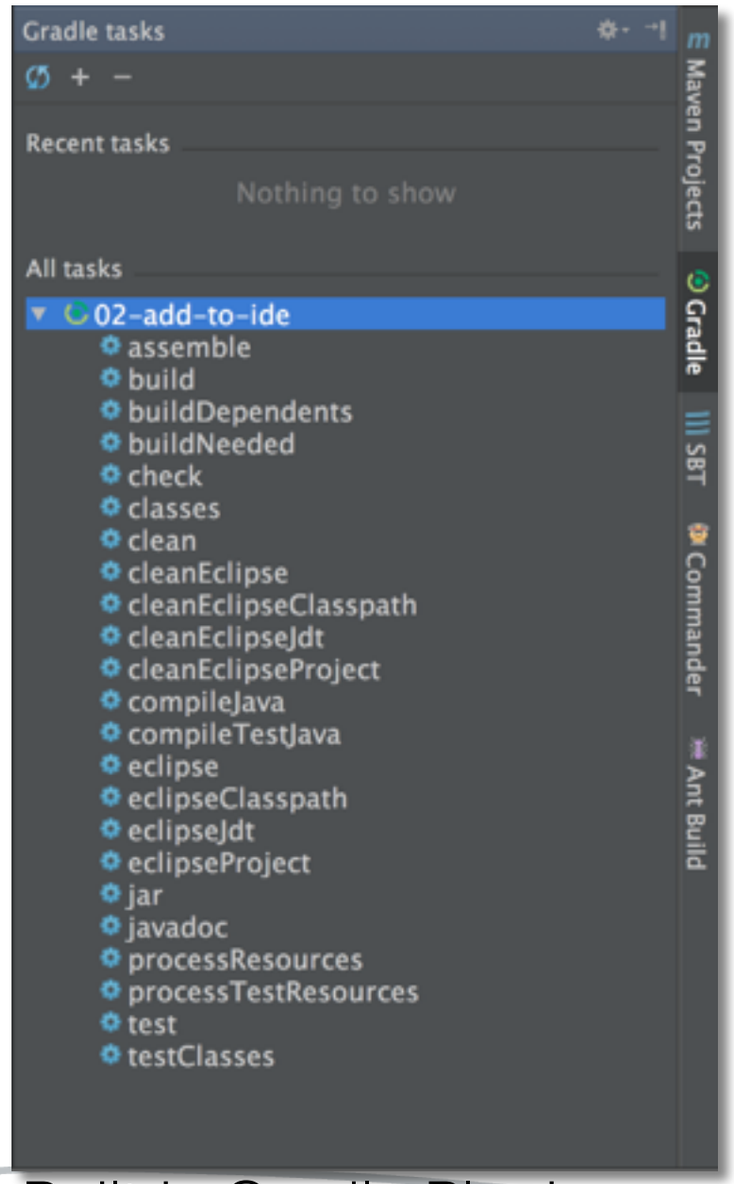
```
apply plugin:’eclipse’
apply plugin:’idea’
```

- Tasks
  - gradle eclipse
  - gradle idea
- Create the project file and put the jars in the IDE classpath.

IDE Integration (cont.)

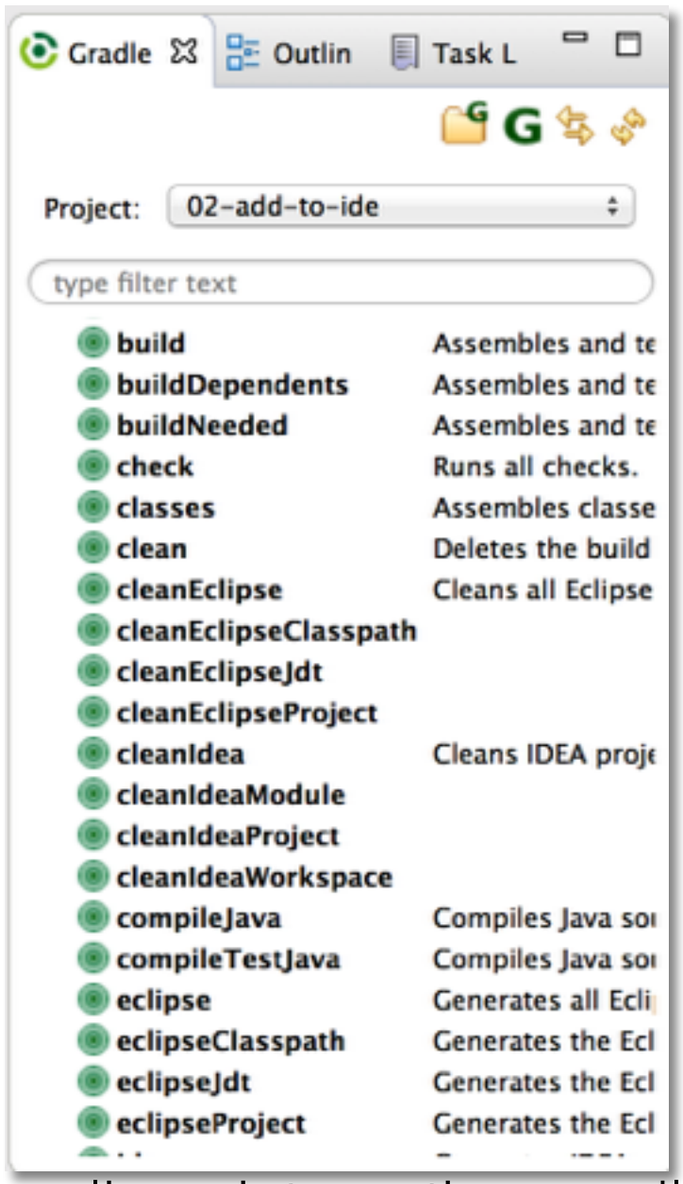
- IDE Gradle plugin or native support
- Gradle Daemon

Intellij



Built-in Gradle Plugin

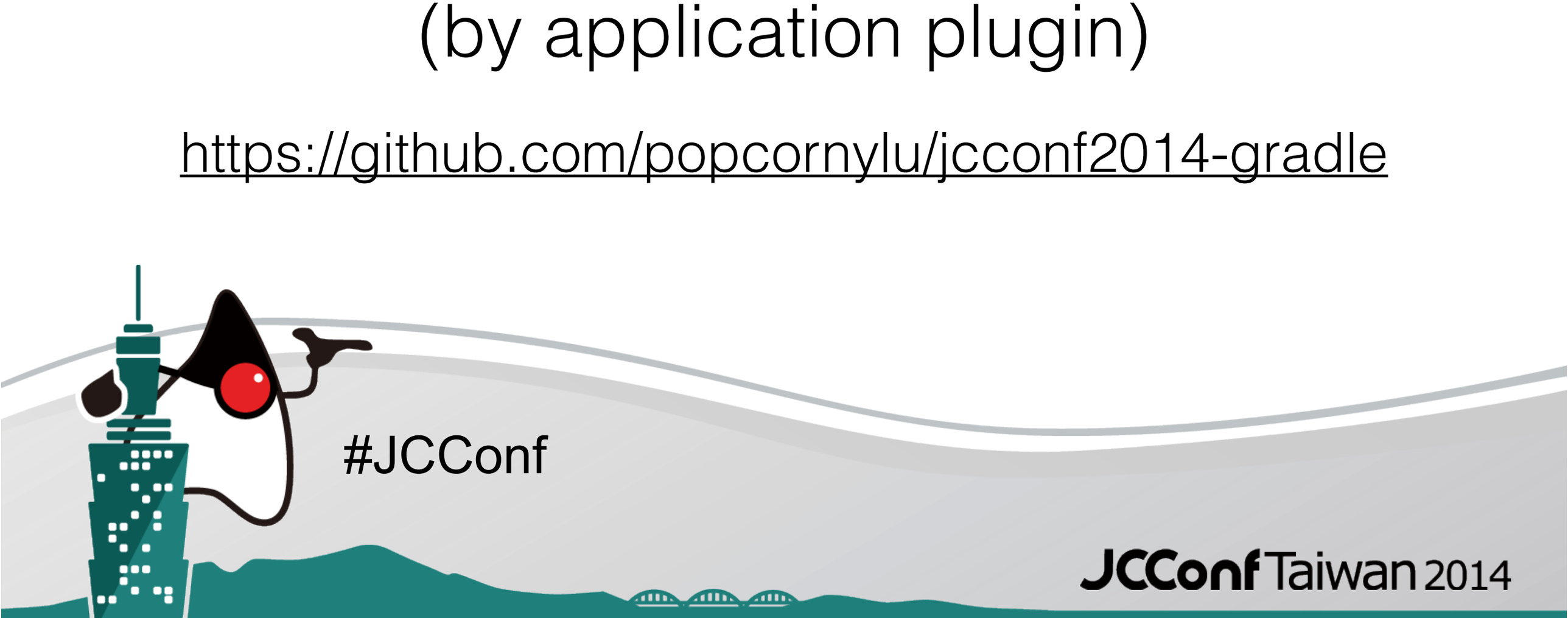
Eclipse



eclipse-integration-gradle

Create a Simple CLI Application by Gradle (by application plugin)

https://github.com/popcornylu/jcconf2014-gradle



```
> java -cp a.jar:b.jar:c.jar tw.jcconf.MyHello
```

https://github.com/popcornylu/jcconf2014-gradle

```
> java -jar myhello.jar
```

https://github.com/popcornylu/jcconf2014-gradle

```
> myhello
```

https://github.com/popcornylu/jcconf2014-gradle

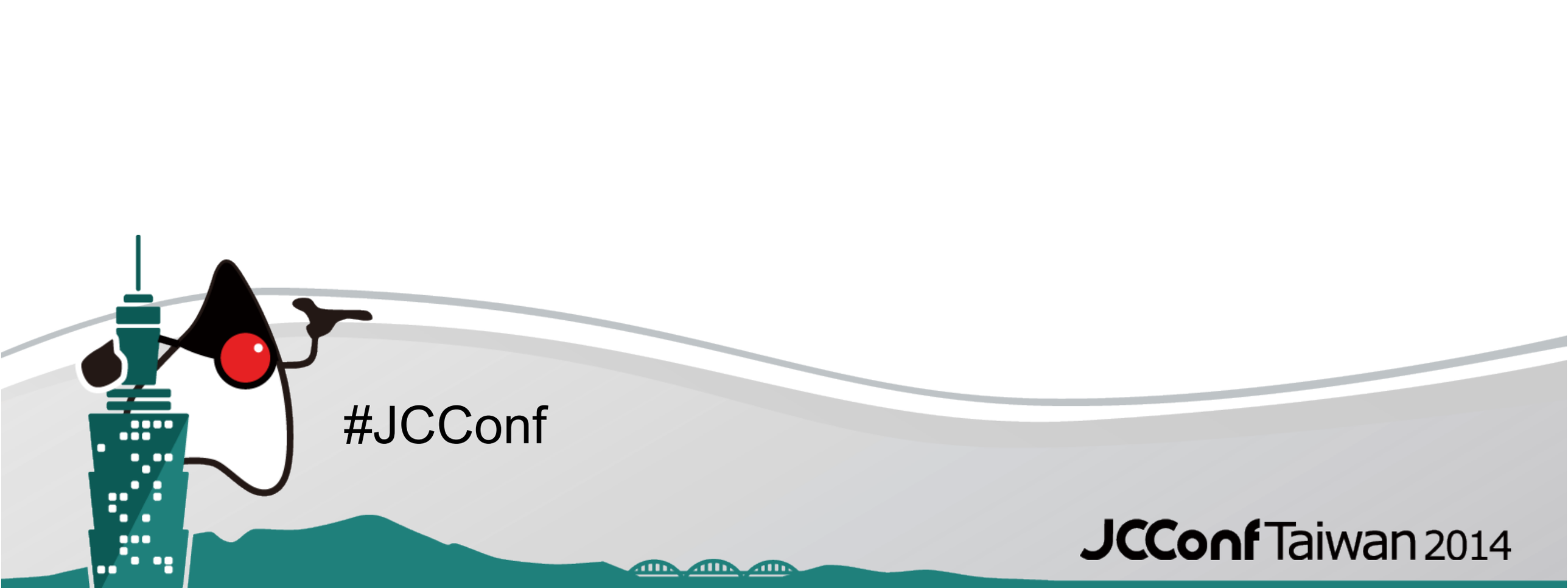
Application Plugin

- Tasks
  - gradle run
  - gradle installApp
  - gradle distZip
  - gradle distTar

```
> tree build/install/myhello/
build/install/myhello/
├── bin
│   ├── myhello
│   └── myhello.bat
└── lib
    ├── commons-cli-1.2.jar
    ├── myhello.jar
    └── slf4j-api-1.7.5.jar
```

https://github.com/popcornylu/jcconf2014-gradle

Demo



Recap

- Gradle exploit advantages from **ant** and **maven**
  - Flexible like Ant
  - Convention over configuration
  - Dependency management
  - Out-of-the-box tasks

- Use DSL instead of XML
- Good IDE integration

Reference

- Gradle User Guide
  http://www.gradle.org/docs/current/userguide/userguide_single.html

- Gradle Tutorial in CodeData (by qrtt1)
  http://www.codedata.com.tw/java/understanding-gradle-1-ant/

- Hibernate. Why Gradle?
  https://developer.jboss.org/wiki/GradleWhy

- Java Build Tools: Ant vs Maven vs Gradle
  http://technologyconversations.com/2014/06/18/build-tools/

Thanks