Paris JUG
BJPC

Sponsors
Platinium

www.parisjug.org

arolla
Mastering Software Development
carbon it
MIRAKL
Cosium
Business Suite Software
SOFTEAM Cadextan
AViSTO
SOFTWARE SOLUTIONS
zenika

Sponsors
Gold

Codenvy
IPPON
Digital . Technologies . Hosting
VISEO

ParisJUG

Extremely fast builds with Gradle 3

Cédric Champeau

2016

Cédric Champeau (@CedricChampeau), Gradle

Gradle

WHO AM I

```
speaker {
    name 'Cédric Champeau'
    company 'Gradle Inc'
    oss 'Apache Groovy committer',
    successes 'Static type checker',
                      'Static compilation'
                      'Traits',
                      'Markup template engine',
                      'DSLs'
        failures Stream.of(bugs),
        twitter '@CedricChampeau',
        github 'melix',
        extraDescription '''Groovy in Action 2 co-author
Misc OSS contribs (Gradle plugins, deck2pdf, jlangdetect, ...)'''
}
```

Happy

AGENDA

- The Gradle Daemon
- Profiling
- Incremental builds
- Incremental compilation
- Continuous builds
- Composite builds
- Task output cache

WHAT IS GRADLE?

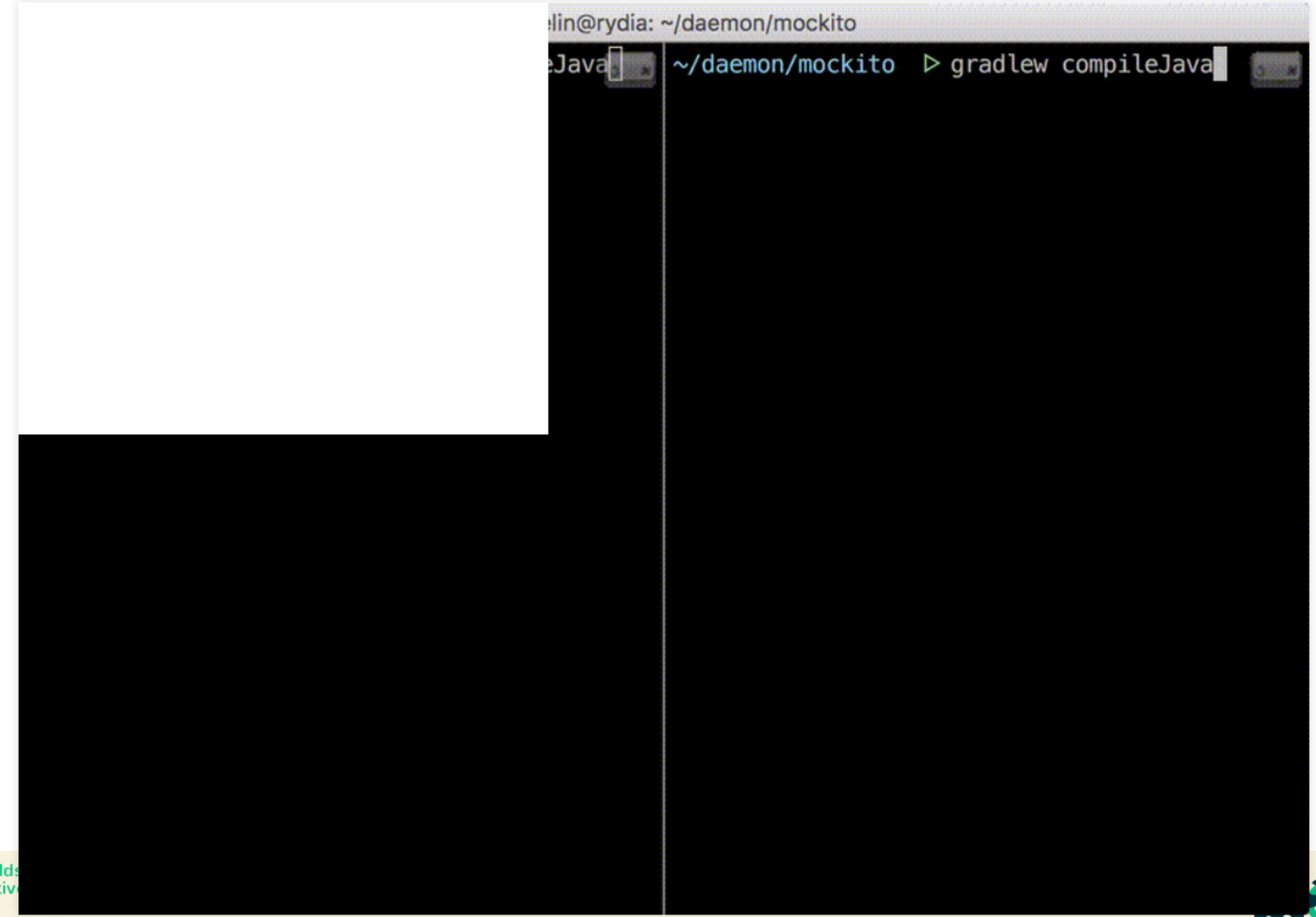
- A build tool
- Cloud Services

THE GRADLE DAEMON

- Long-lived background process
- Listens and executes build actions
- Faster startup / execution
- Enabled by default since 3.0

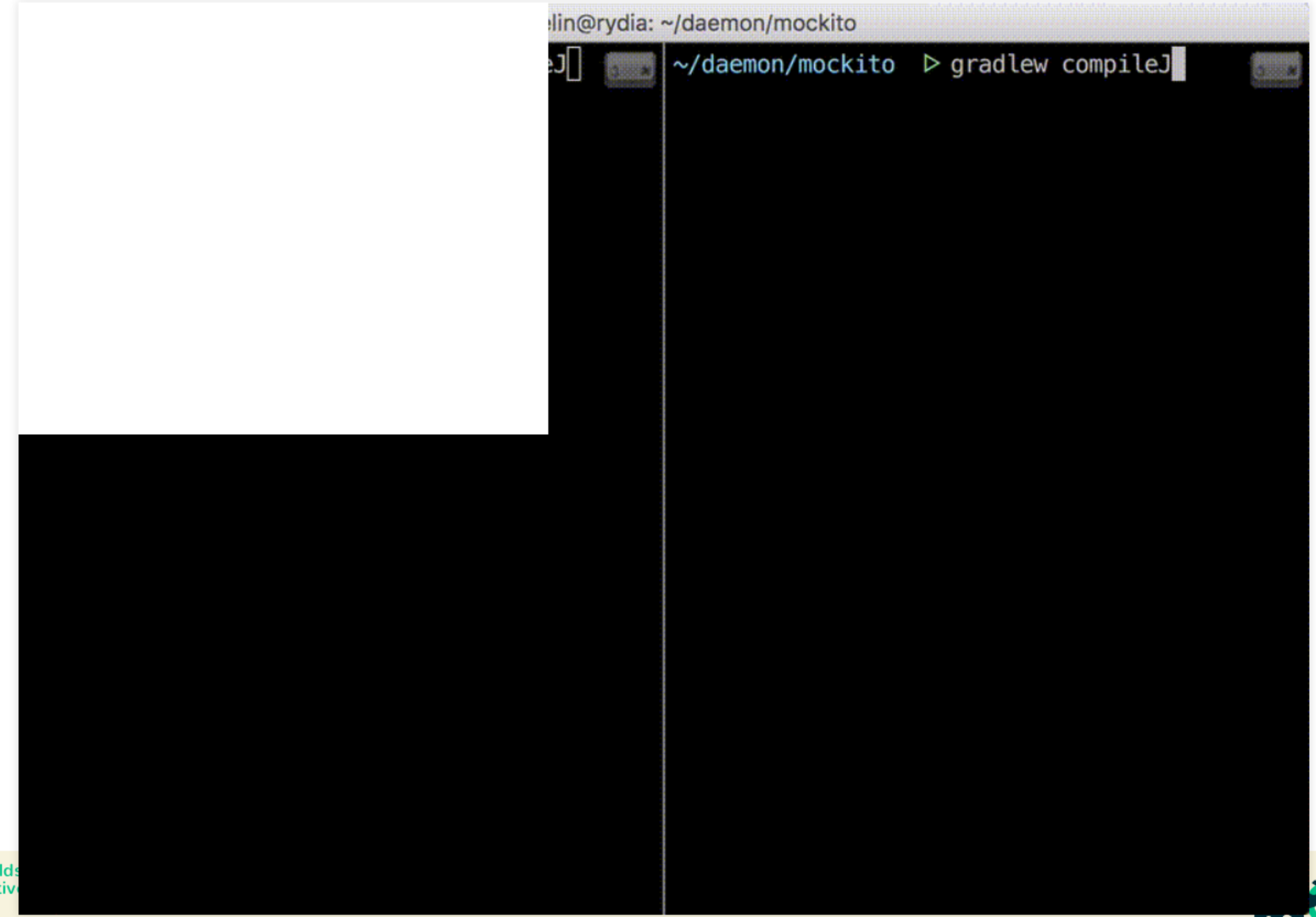
THE GRADLE DAEMON

COLD DAEMON



Gradle

WARM DAEMON



Gradle

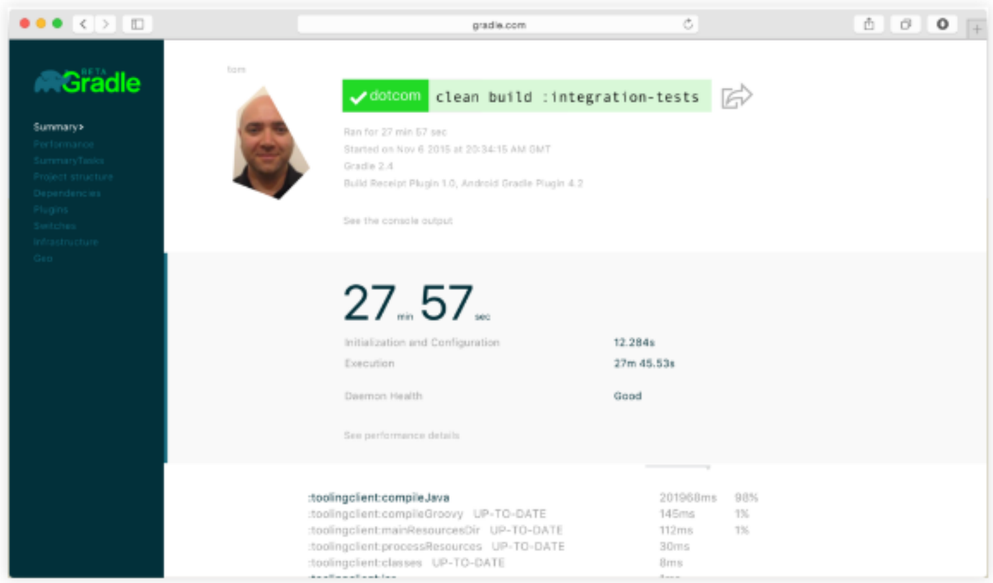
PROFILING A BUILD

`gradle --profile`

- Generates a report in the build directory
- Limited insight
- Not easily shareable
- But can already give precious information

BUILD SCANS

- First member of the Cloud Services family
- Insights into your build
- View and share via URL
- Debug, optimize and refine
- Analyze all of your builds
- Available for free



BUILD SCAN DEMO

https://scans.gradle.com/s/sample/groovy

CREATING A BUILD SCAN

APPLY THE BUILD SCAN PLUGIN

```
plugins {
    id 'com.gradle.build-scan' version '1.2'
}

buildScan {
    licenseAgreementUrl = 'https://gradle.com/terms-of-service'
    licenseAgree = 'yes'
}
```

RUN THE BUILD

```
./gradlew -Dscan build
```

CLICK ON THE LINK

```
:jrubyPrepareGems
:asciidoctor UP-TO-DATE

BUILD SUCCESSFUL

Total time: 1.132 secs

Publishing build information...
https://gradle.com/s/v2f5knnujnsx2
```

INCREMENTAL BUILDS

- Gradle is meant for incremental builds
- `clean` is a waste of time
- Prepare your builds for incrementalness

EXAMPLE: BUILDING A SHADED JAR

```
task shadedJar(type: ShadedJar) {
   jarFile = file("$buildDir/libs/shaded.jar")
   classpath = configurations.runtime
   mapping = ['org.apache': 'shaded.org.apache']
}
```

- What are the task inputs?
- What are the task outputs?
- What if one of them changes?

DECLARING INPUTS

```
@InputFiles
FileCollection getClasspath() { ... }

@Input
Map<String, String> getMapping() { ... }
```

DECLARING OUTPUTS

```
@OutputFile
File getJarFile() { ... }
```

INCREMENTAL COMPILATION

- Given a set of source files
- Only compile the files which have changed…
- and their dependencies
- Language specific

GRADLE HAS SUPPORT FOR INCREMENTAL COMPILATION OF JAVA

```
compileJava {
    //enable incremental compilation
    options.incremental = true
}
```

CONTINUOUS BUILDS

- Gradle watches for changes in task inputs
- Re-executes tasks as changes occur
- Enabled with `-t`

```
gradle -t asciidoctor
```

COMPOSITE BUILDS

- Compose various projects as if there were one
  - Each project can live in its own repository
  - Each project has its own Gradle build
  - Composition unites them through dependency resolution
- Split monolithic projects
  - For large multiproject builds, allows splitting them into several pieces
  - Each piece can be versioned independently
  - Developers can choose what subprojects they care about

COMPOSITE BUILDS DEMO

TASK OUTPUT CACHE

- Avoid doing work even after clean
- Share binaries between projects on a single machine
- Share binaries between projects on a network
- Backend agnostic

TASK OUTPUT CACHE USE CASES

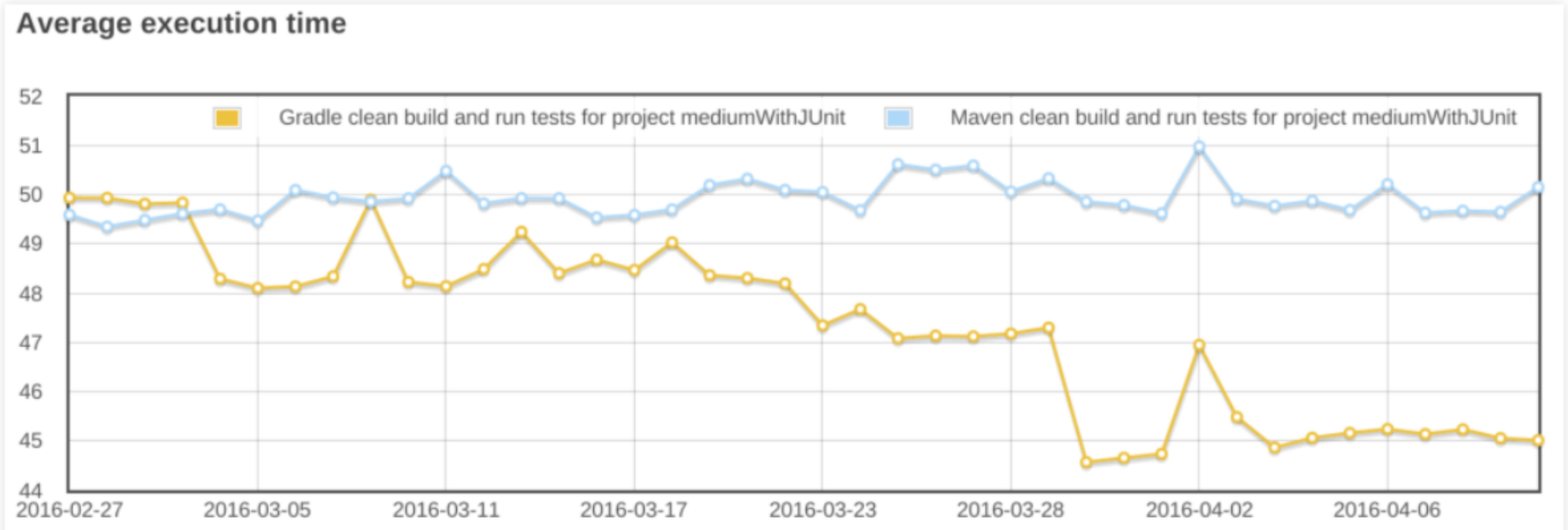
- Long compile tasks
- Bisecting
- Continuous integration
- Green earth

TASK OUTPUT CACHE DEMO

With composite builds!

PERFORMANCE GUIDE

https://gradle.github.io/performance-guide/



THANK YOU!

- Slides and code : https://github.com/melix/breizhjug-fast-builds
- Gradle documentation : http://gradle.org/documentation/
- Follow me: @CedricChampeau

Learn more at www.gradle.org