

Surviving Dependency Hell

With Maven

Ray Tsang

Developer Advocate
@ Google

Java Champion
Spring Cloud GCP
JDeferred

Contributors to JLBP.dev

@saturnism | saturnism.me



Robert Scholte

Java Champion

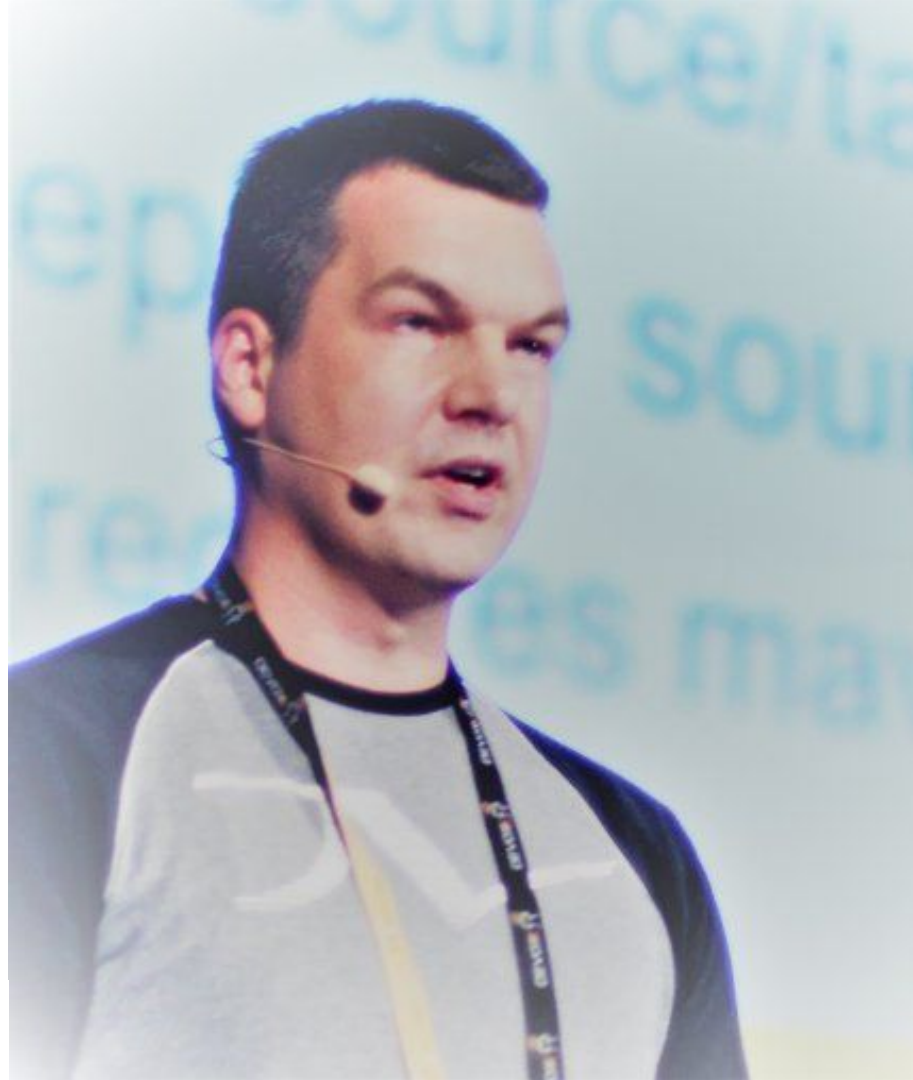
Founder, Source Ground

Apache Maven PMC Chair

Member Expert Group

JSR 376 - Jigsaw

@rfscholte



Andres Almiray

Seasoned Sourceror

@ Oracle

Java Champion

Hackergarten

JDeferred & many others!

@aalmiray | andresalmiray.com



Let's see the code

NoSuchMethodError

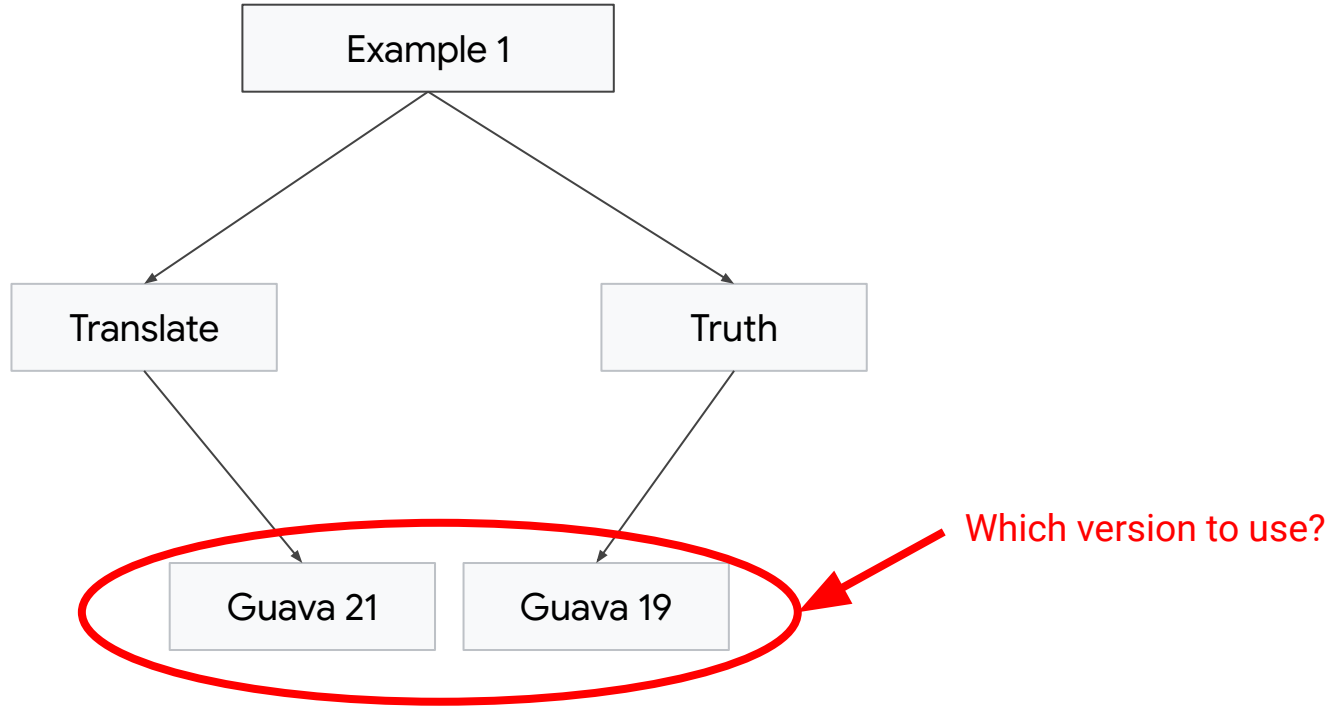
NoSuchFieldError

NoClassDefFoundError

```
mvn dependency:tree
```

```
mvn dependency:tree -Dverbose=true
```

```
mvn dependency:tree -Dverbose=true -Dincludes=...
```



Any dependency there can be only ONE version

Classpath - First class wins

Maven - Nearest wins

Maven **does not understand** Semver
Nor compatibility

Use Maven Enforcer

Convergence vs Upper bound

<http://maven.apache.org/enforcer/enforcer-rules/dependencyConvergence.html>

(See Code!)

Ensuring only ONE version
of the dependency in tree

Exclusions, or Dependency Management
(See Code)

Upper bound if higher version is backwards compatible

[JLBP-7] Make breaking transitions easy

[JLBP-10] Maintain API stability as long as needed for consumers

(Guava 21 and up are backwards compatible)

If upper version is breaking lower version...

Or, system classpath has an incompatible version
(Hadoooooooooop)

Shade + Relocation :(

Classloaders

OSGi

GA libraries don't depend on non-GA APIs

alpha, beta, RC, 0.xx, @UnstableApi, @Beta, @Internal

[JLBP-4] Avoid dependencies on unstable libraries and features

Major release, breaking changes

Use new Group ID or Artifact ID - different coordinate!

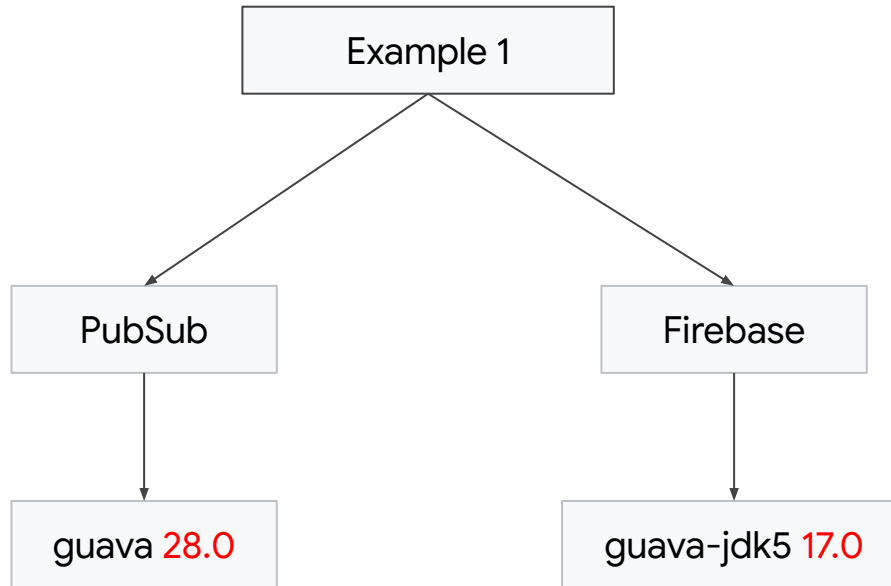
AND Use a new package name

[\[JLBP-6\] Rename artifacts and packages together](#)

How do we change versions in transitive dependencies?

Maven Dependency Management

Let's see Example 3e



Ban Duplicate Classes

<https://www.mojohaus.org/extra-enforcer-rules/banDuplicateClasses.html>

2 artifacts should not have overlapping classes

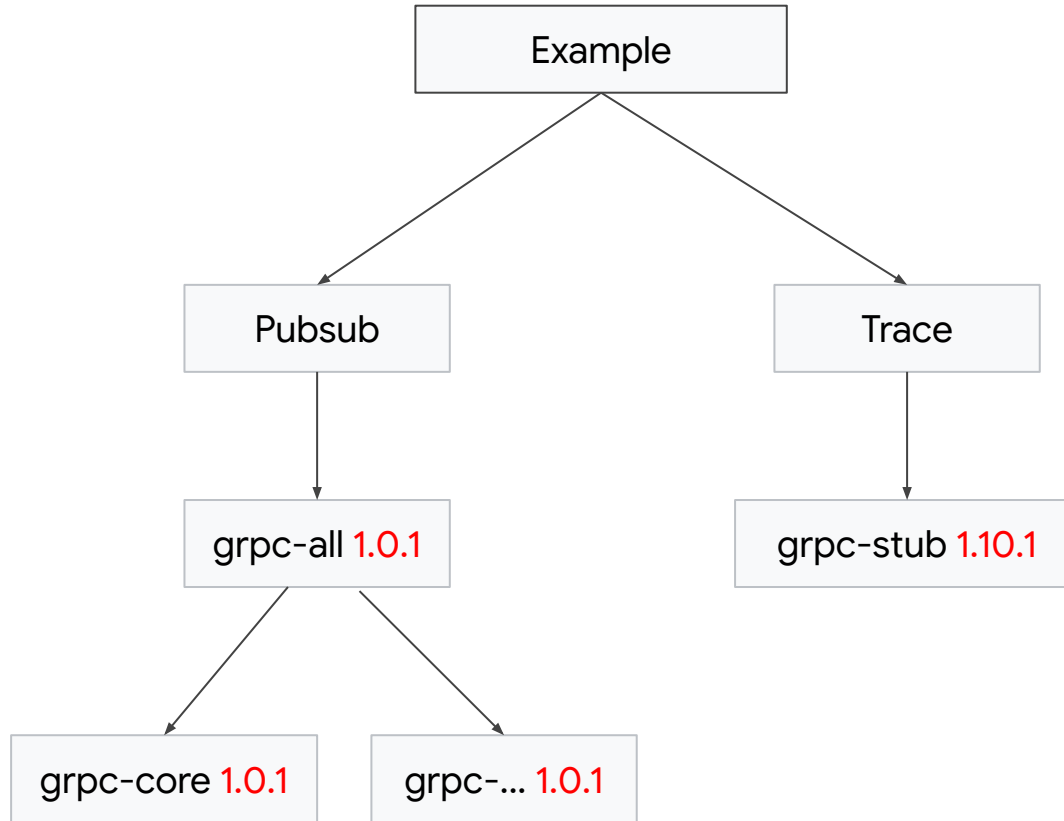
[JLBP-5] Avoid dependencies that overlap classes with other dependencies

[JLBP-19] Place each package in only one module

[JLBP-11] Keep dependencies up to date!

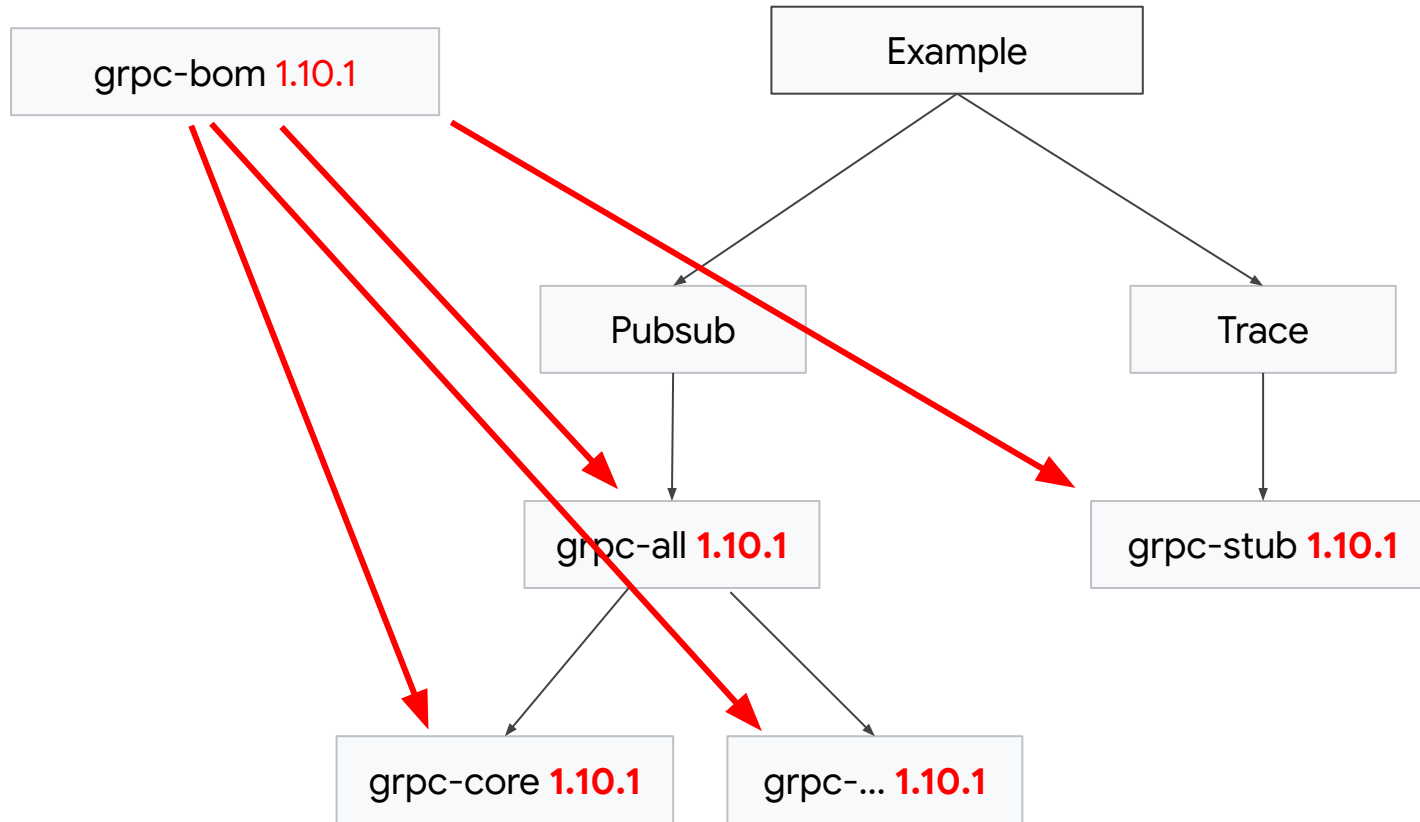
Let's fix all of these

And see the next example 3c



Version Misalignment

How do we change versions in transitive dependencies?



Multi-module projects Produce a BOM!

[JLBP-15] Publish a BOM for multi-module projects

See an example!

Power → Responsibility

BOM should not overreach!

ONLY modules within the multi-project project

gRPC also uses another method

Strict Version Ranges for inter-module dependencies,
e.g. [1.10.1]

Linkage Checker

Static Analysis to identify missing links

<https://github.com/GoogleCloudPlatform/cloud-opensource-java/tree/master/enforcer-rules>

Bigger issues with Dependency Management...

What you build & test with
Is not what consumers get!

Flatten the POM

<https://github.com/mojohaus/flatten-maven-plugin/pull/93>

Bonus

`mvn dependency:analyze` - find dependencies you don't need

Maven 3.6.3 - fixes our exclusion fix in Maven 3.6.2

Enforcer Rules

Enforcer Version Convergence or Upperbound

Ban Duplicate Classes

Linkage Checker

Manage Transitive Versions

`<dependencyManagement/>`

BOM for Multi-Module

Import BOM

Maven Flatten Plugin

Thanks!

Andres Almiray @aalmiray

Ray Tsang @saturnism

Maven used by 60% to 80% of Java projects

92 projects, 50+ plugins, lots of libraries - Contribution Appreciated!

Visit the best practices at [JLBP.dev](https://jlbp.dev)