Taming Java dependencies in 70+ GCP client libraries

Stephanie Wang
http://stephaniewang526.github.io/

Stephanie Wang

Developer Programs Engineer
Google Cloud Platform

Java BigQuery client libraries

@StephWangBuilds | stephaniewang526.github.io

What are client libraries?

- Google Cloud Client Libraries are our **latest** and **recommended** client libraries for calling Google Cloud APIs.
- They provide an optimized developer experience by using each supported language's natural conventions and styles.
  - Provide idiomatic, generated or hand-written code in each language, making the Cloud API simple and intuitive to use.
  - Handle all the low-level details of communication with the server.
  - https://cloud.google.com/apis/docs/client-libraries-explained

The Real Java Maintainers of 70+ GCP Client Libraries

When you have 70+ Java client libraries...

- Dependency conflicts are hard to reconcile!

[2/20/2020] UpperBound dependency error when updating GCS in BigQuery:

```
Failed while enforcing RequireUpperBoundDeps. The error(s) are [
Require upper bound dependencies error for
com.google.protobuf:protobuf-java-util:3.11.3 paths to dependency
are:
+-com.google.cloud:google-cloud-bigquery:1.107.1-SNAPSHOT
  +-com.google.cloud:google-cloud-core:1.92.5
    +-com.google.protobuf:protobuf-java-util:3.11.3
and
+-com.google.cloud:google-cloud-bigquery:1.107.1-SNAPSHOT
  +-com.google.cloud:google-cloud-storage:1.104.0
    +-com.google.protobuf:protobuf-java-util:3.11.4
and
+-com.google.cloud:google-cloud-bigquery:1.107.1-SNAPSHOT
  +-com.google.cloud:google-cloud-core:1.92.5
    +-com.google.protobuf:protobuf-java-util:3.11.3
,
Require upper bound dependencies error for
io.opencensus:opencensus-api:0.24.0 paths to dependency are:
```

```
Require upper bound dependencies error for
io.opencensus:opencensus-api:0.24.0 paths to dependency are:
+-com.google.cloud:google-cloud-bigquery:1.107.1-SNAPSHOT
  +-com.google.cloud:google-cloud-core-http:1.92.5
    +-io.opencensus:opencensus-api:0.24.0
and
+-com.google.cloud:google-cloud-bigquery:1.107.1-SNAPSHOT
  +-com.google.cloud:google-cloud-storage:1.104.0
    +-io.opencensus:opencensus-api:0.25.0
and
+-com.google.cloud:google-cloud-bigquery:1.107.1-SNAPSHOT
  +-com.google.http-client:google-http-client:1.34.2
    +-io.opencensus:opencensus-api:0.24.0
and
+-com.google.cloud:google-cloud-bigquery:1.107.1-SNAPSHOT
  +-com.google.api:gax:1.53.1
    +-io.opencensus:opencensus-api:0.24.0
and
+-com.google.cloud:google-cloud-bigquery:1.107.1-SNAPSHOT
  +-com.google.cloud:google-cloud-core-http:1.92.5
    +-io.opencensus:opencensus-contrib-http-util:0.24.0
      +-io.opencensus:opencensus-api:0.24.0
]
```

When you have 70+ Java client libraries...

- Excessive dependency update PRs
  - 104 dependency update PRs in BigQ day)!
  -

stephwang@, BigQuery Java client libraries maintainer, circa Feb 2020.

104 issues or view all results on GitHub

Sort: Best match

googleapis/java-bigquerydatatransfer #100
deps: update core dependencies
This PR contains the following updates: | Package | Update | Change | | --- | --- | --- | | io.grpc:grpc-bom | patch | 1.28.0 -> 1.28.1 | | com.google.api:gax-grpc | minor | 1.54.0 -> 1.55.0 | | ...
cla: yes
renovate-bot opened on Mar 27 2 comments
`googleapis:master ← renovate-bot:renovate/core-dependencies`

googleapis/java-bigquery #218
deps: update core dependencies
This PR contains the following updates: | Package | Update | Change | | --- | --- | --- | | io.grpc:grpc-bom | minor | 1.27.2 -> 1.28.0 | | com.google.cloud:google-cloud-core | patch | 1.93.1 -> ...
cla: yes kokoro:run
renovate-bot opened on Mar 10 2 comments
`googleapis:master ← renovate-bot:renovate/core-dependencies`

googleapis/java-bigquerystorage #98
deps: update core dependencies
This PR contains the following updates: | Package | Update | Change | | --- | --- | --- | | io.grpc:grpc-bom | minor | 1.27.2 -> 1.28.0 | | com.google.cloud:google-cloud-core | patch | 1.93.1 -> ...
cla: yes kokoro:run

Understand our problems

- Goals:
  - Reduce number of dependency update PRs.
  - Reduce UpperBound dependency errors during dependency updates.
- Translated goals:
  - Stop managing common dependencies individually in each client library.
  - Use consistent versions of the same dependency.
- Ideally, we should be able to...
  - Bundle up all the common dependencies to manage their versions at one place.
  - Pull the commonly managed dependency version easily into each client library.

google-cloud-shared-dependencies BOM to the rescue!

- What is a BOM?
  - A special POM file that Maven lets us define the versions of our dependencies or transitive dependencies.
  - It is in this POM that we declare the versions and scope of the dependencies.
  - A centralized place to mention all the dependency details.
- Best practice! https://jlbp.dev/JLBP-15
  - Publish a BOM for multi-module projects
  - Importing a BOM means dependency version from the BOM will be used.

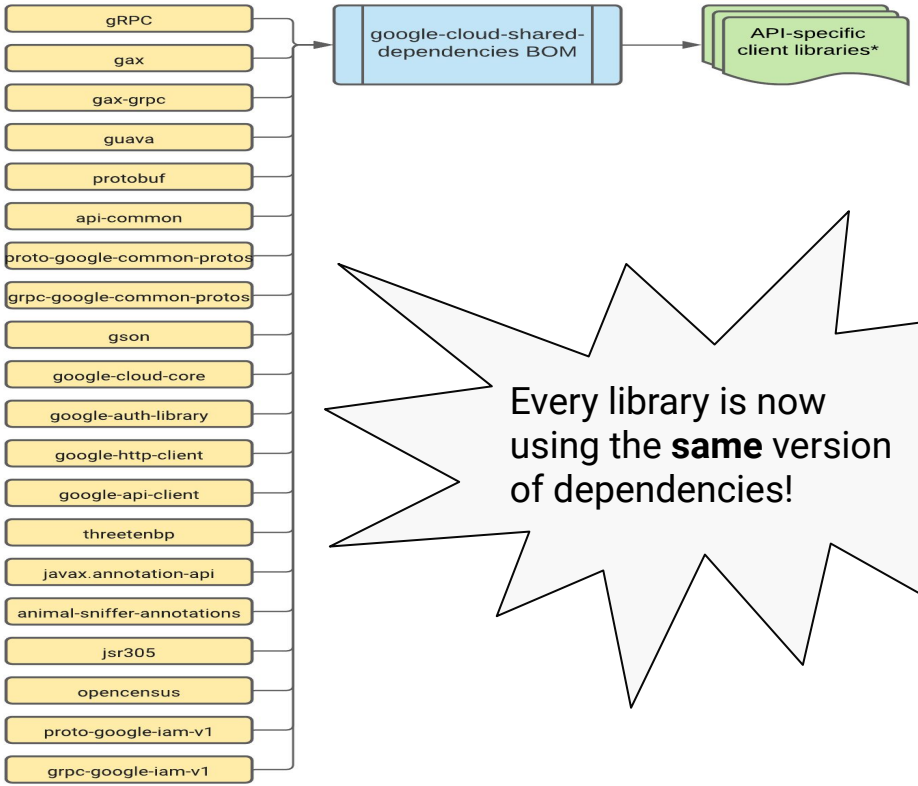
google-cloud-shared-dependencies BOM

What we did:

- Bundled up all the common depend
- Each client is importing this BOM.
- Full list of dependencies

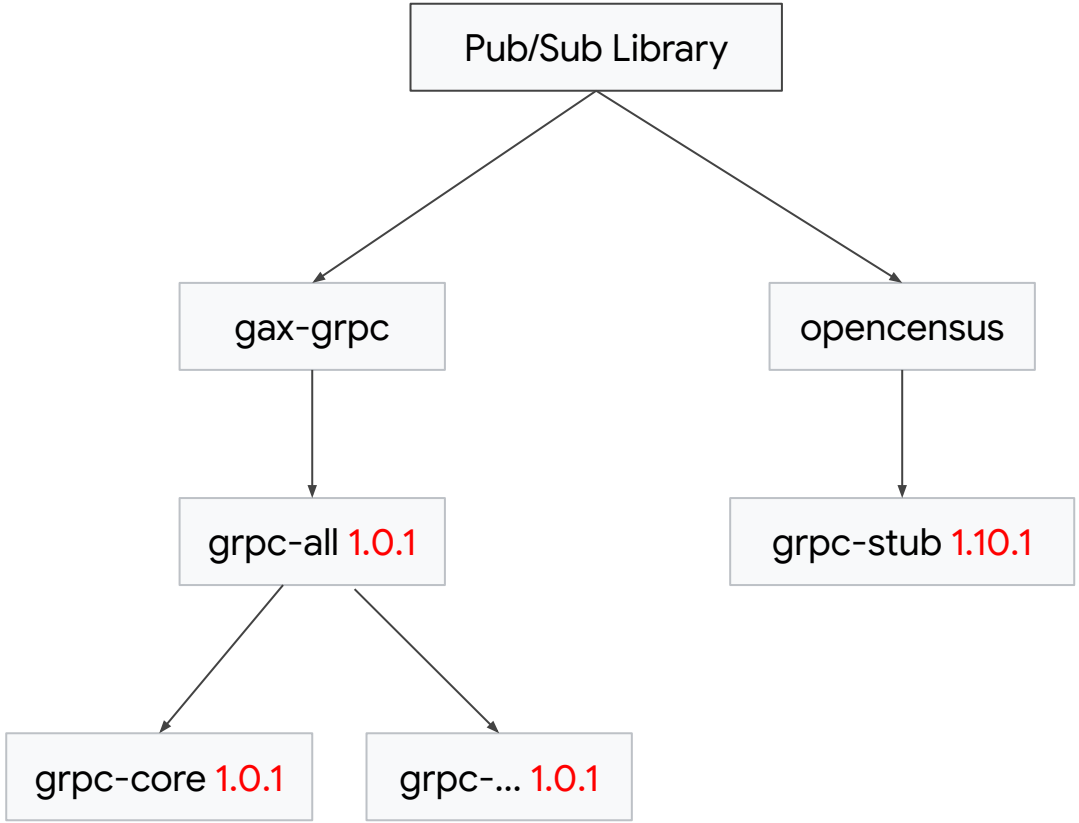
Results:

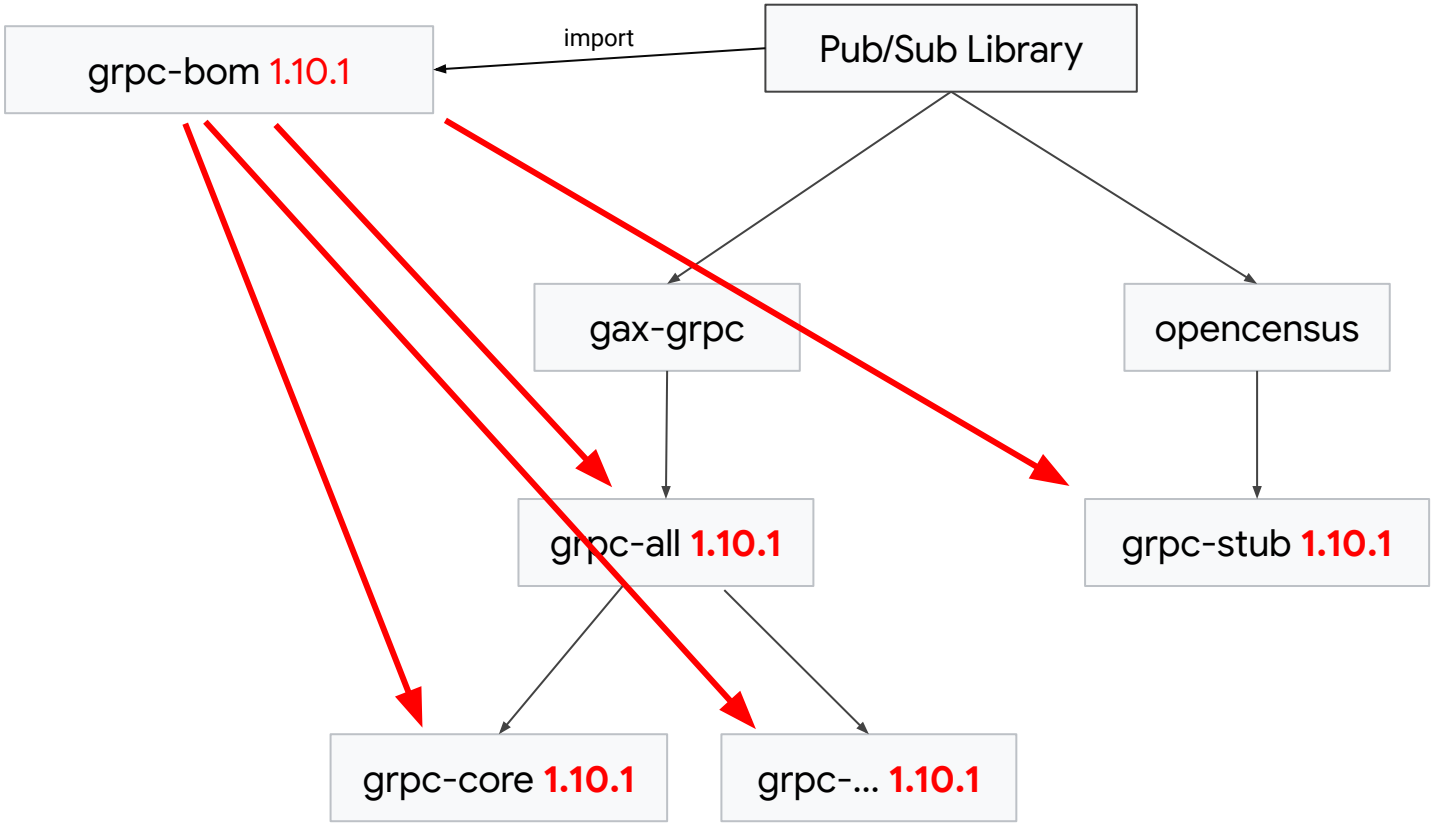
- No more UpperBound dependency
  same dependency.
- Number of dependency update PRs

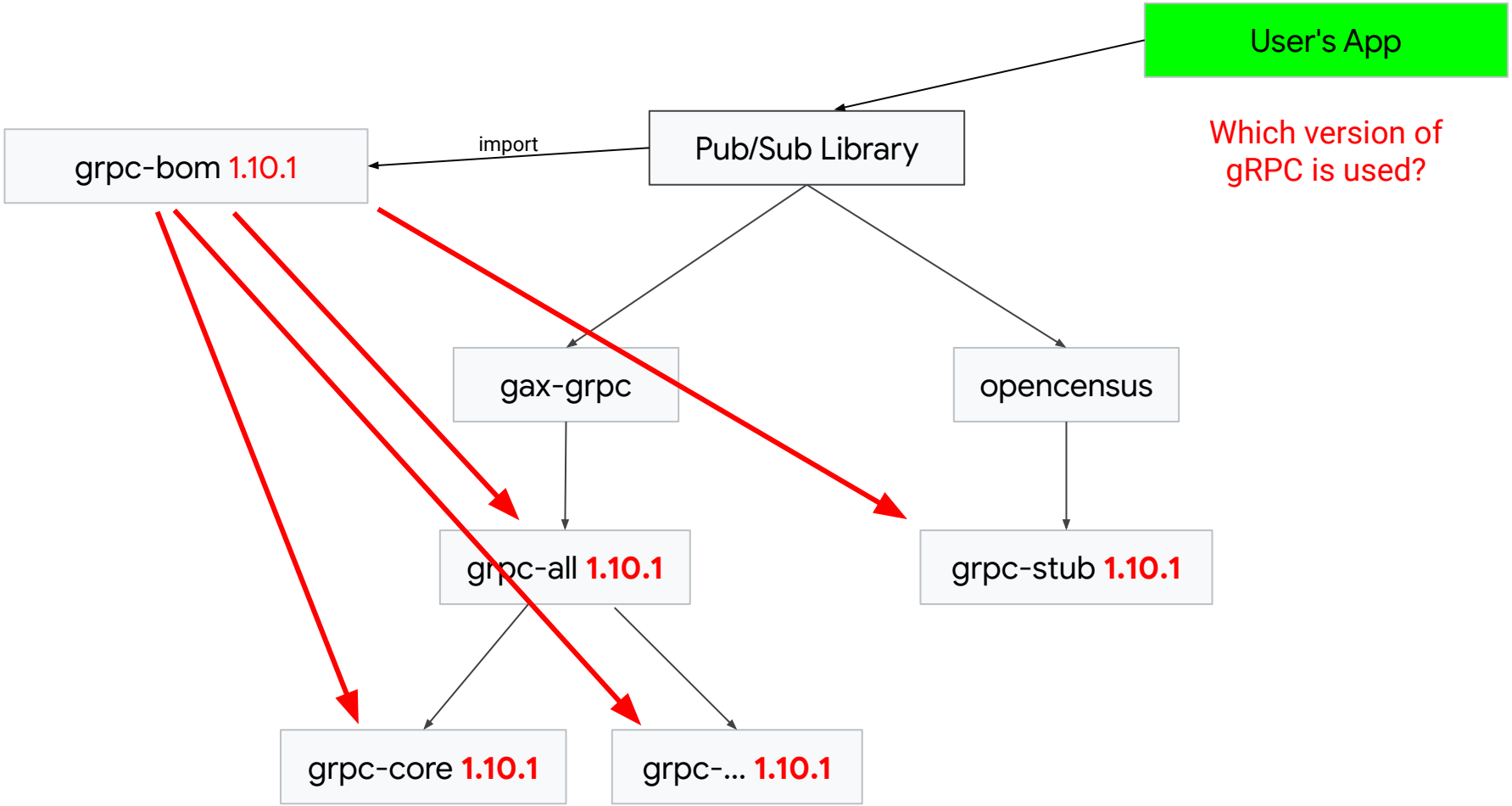


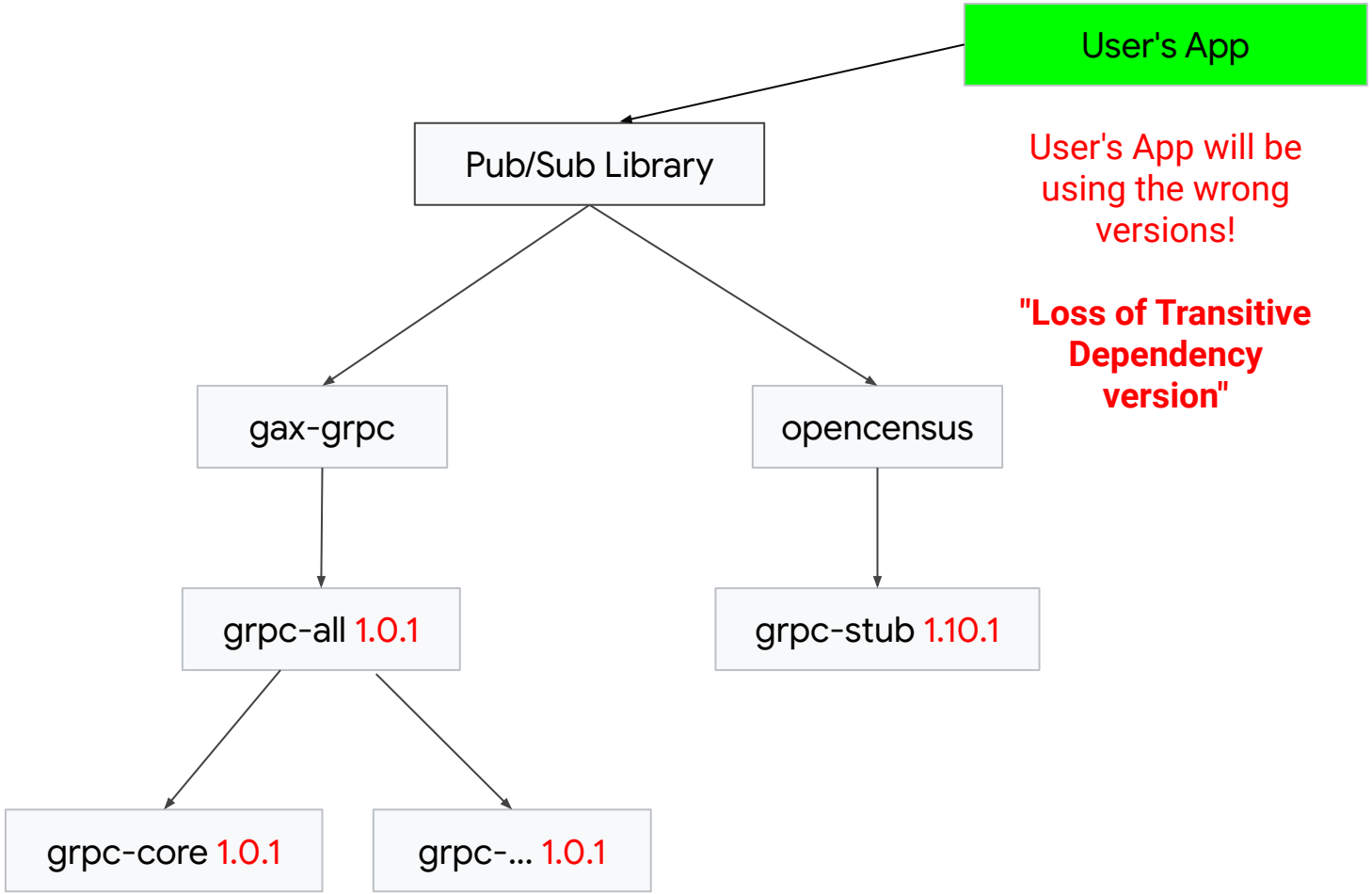
BOM is just part of the solution

But it also introduces new problems


Pub/Sub Library
gax-grpc
opencensus
grpc-all 1.0.1
grpc-stub 1.10.1
grpc-core 1.0.1
grpc-... 1.0.1

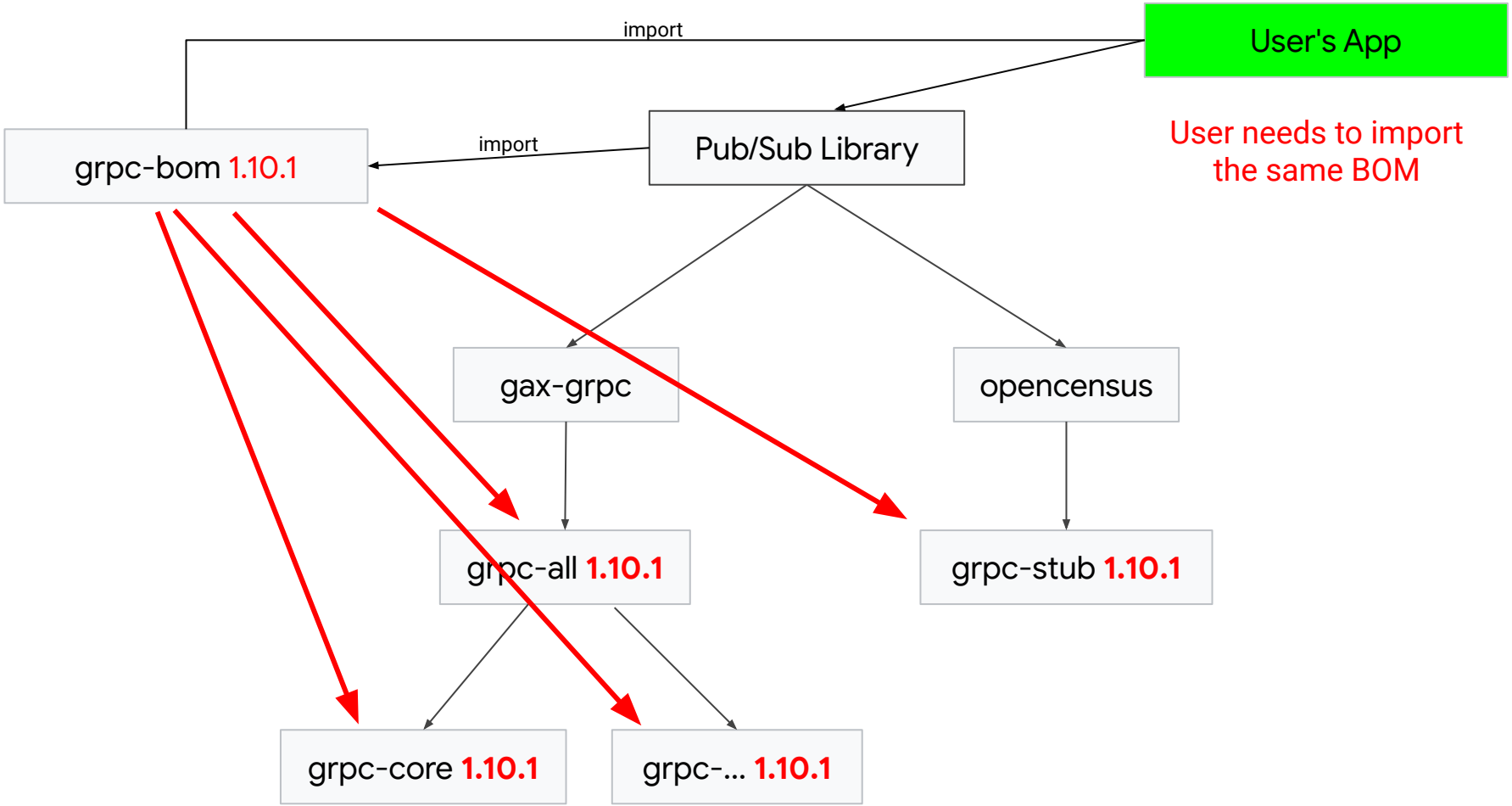

grpc-bom 1.10.1
import
Pub/Sub Library
gax-grpc
opencensus
grpc-all 1.10.1
grpc-stub 1.10.1
grpc-core 1.10.1
grpc-... 1.10.1


User's App
Which version of gRPC is used?
import
grpc-bom 1.10.1
Pub/Sub Library
gax-grpc
opencensus
grpc-all 1.10.1
grpc-stub 1.10.1
grpc-core 1.10.1
grpc-... 1.10.1



User's App will be using the wrong versions!

"Loss of Transitive Dependency version"


import
User's App
import
grpc-bom 1.10.1
Pub/Sub Library
User needs to import
the same BOM
gax-grpc
opencensus
grpc-all 1.10.1
grpc-stub 1.10.1
grpc-core 1.10.1
grpc-... 1.10.1

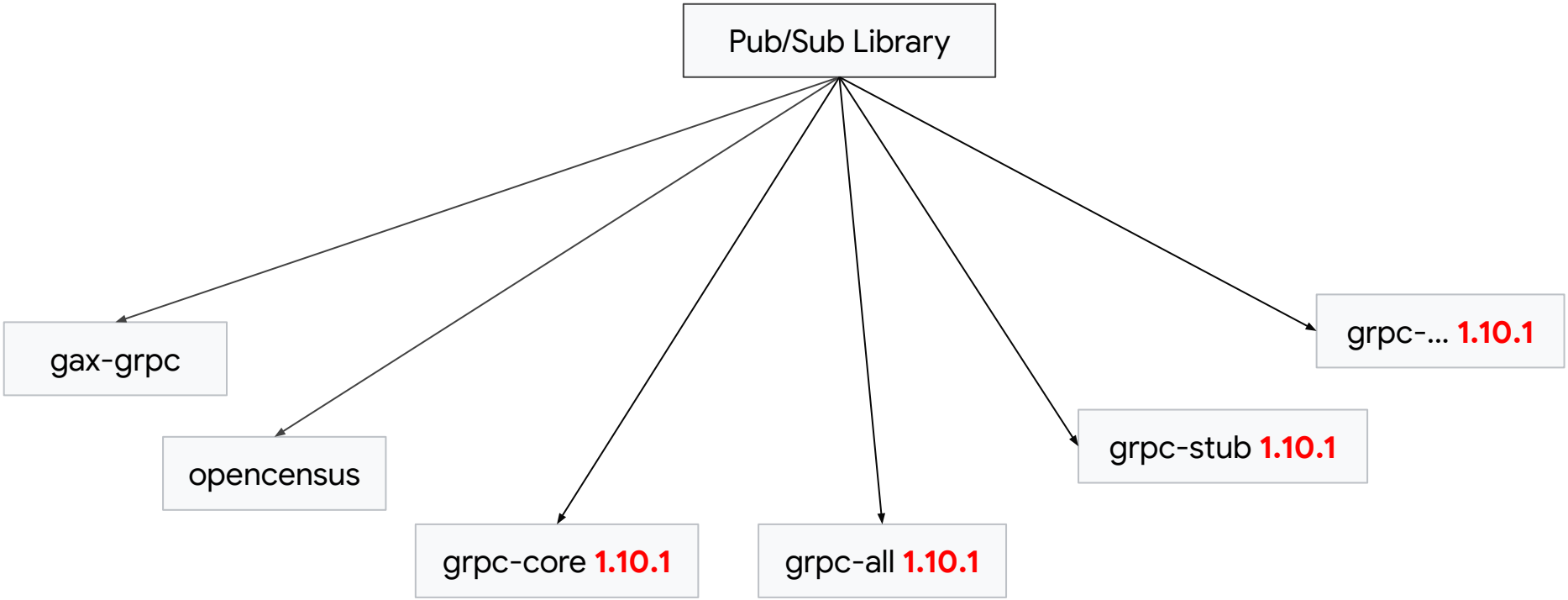
Or… Lock the Versions

Maven Flatten Plugin

Encode the version specified by the BOM into the pom.xml as it's published to Maven Central

pom.xml → .flattened-pom.xml → Maven Central

Example: google-cloud-bigquery pom.xml (flattened pom)


Pub/Sub Library
gax-grpc
opencensus
grpc-core 1.10.1
grpc-all 1.10.1
grpc-stub 1.10.1
grpc-... 1.10.1

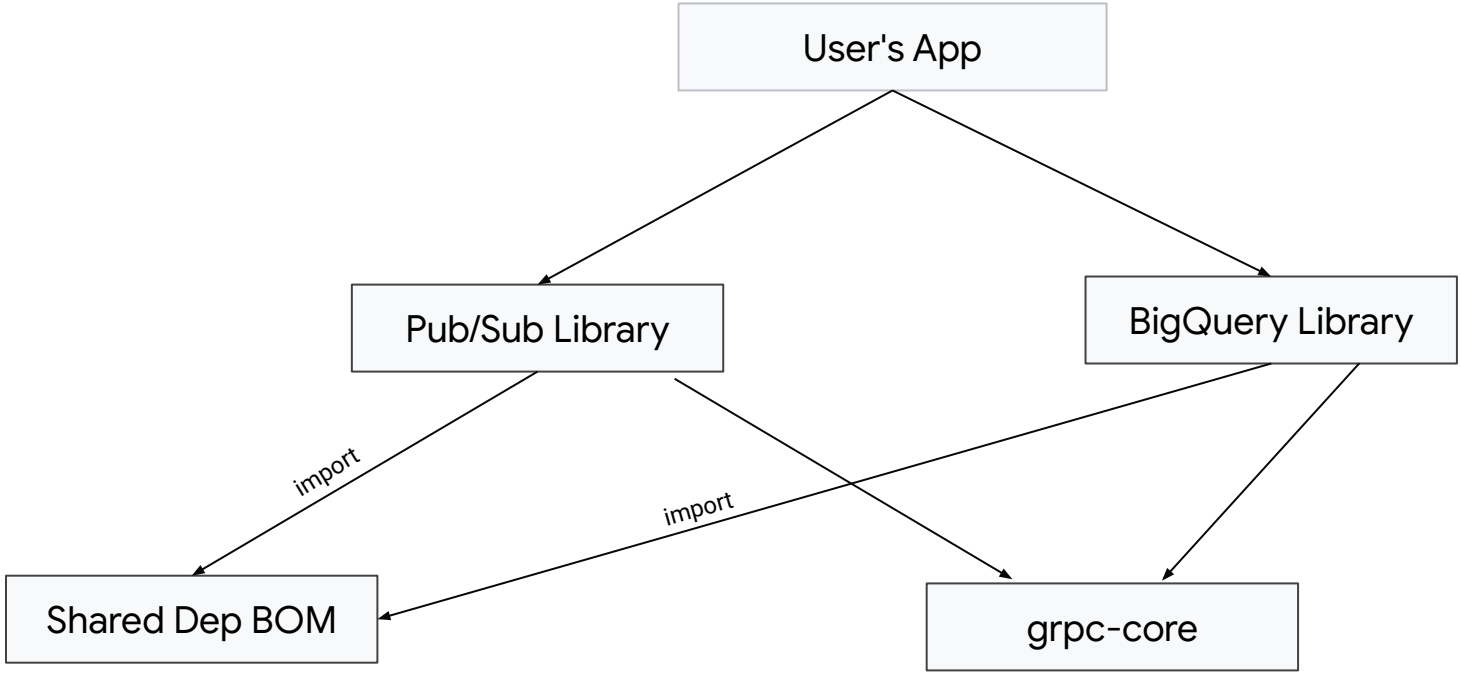
Transformation can have issues

Check for correctness of the transformed flattened pom.xml

- Check for resolution order is the same
- Check for the runtime dependencies are the same

Check is pre-submit (Link)

```
47    msg "Generating dependency list using original pom..."
48    mvn dependency:list -f pom.xml -DincludeScope=runtime -Dsort=true | grep '\[INFO]    .*:.*:.*:.*:.*' | sed -e s/\\s--\\smodule.*// >.org-list.txt
49
50    # Output dep list generated using the flattened pom (only 'compile' and 'runtime' scopes)
51    msg "Generating dependency list using flattened pom..."
52    mvn dependency:list -f .flattened-pom.xml -DincludeScope=runtime -Dsort=true | grep '\[INFO]    .*:.*:.*:.*:.*' >.new-list.txt
53
54    # Compare two dependency lists
55    msg "Comparing dependency lists..."
56    diff .org-list.txt .new-list.txt >.diff.txt
```


User's App
Pub/Sub Library
BigQuery Library
import
import
Shared Dep BOM
grpc-core

Case study: Address security vulnerability

- CVE introduced due to commons-codec 1.11

> commons-codec:commons-codec:1.11 - sonatype-2012-0050
> The Apache commons-codec package contains an Improper Input Validation vulnerability. The decode() method in the Base32 and Base64 classes fails to reject malformed Base32 and Base64 encoded strings and consequently decode them into arbitrary values. A remote attacker can leverage this vulnerability to potentially tunnel additional information via seemingly legitimate Base32 or Base64 encoded strings.

- Dependency tree:

```
[INFO] com.google.cloud:google-cloud-bigquery:jar:1.126.4-SNAPSHOT
[INFO] \- com.google.http-client:google-http-client:jar:1.38.0:compile
[INFO]    \- org.apache.httpcomponents:httpclient:jar:4.5.13:compile
[INFO]       \- commons-codec:commons-codec:jar:1.11:compile
```

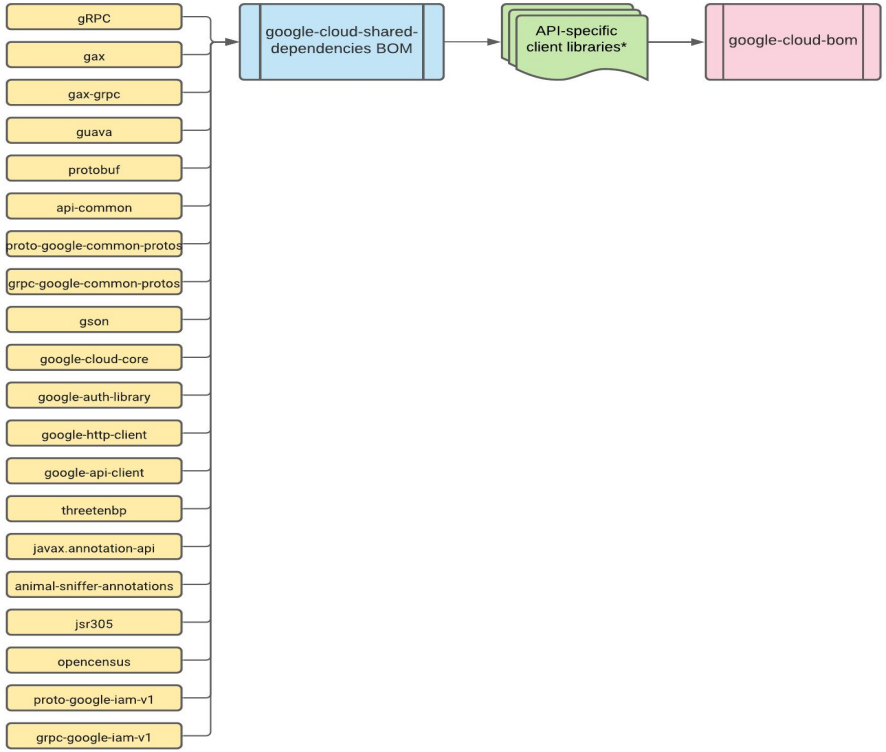
- Not gonna work: https://github.com/googleapis/google-http-java-client/pull/1221
- What works! https://github.com/googleapis/java-shared-dependencies/pull/251

```
[INFO] com.google.cloud:google-cloud-bigquery:jar:1.126.6
[INFO] \- com.google.http-client:google-http-client:jar:1.38.0:compile
[INFO]    \- org.apache.httpcomponents:httpclient:jar:4.5.13:compile
[INFO]       \- commons-codec:commons-codec:jar:1.15:compile
```

What does this mean for library consumers?

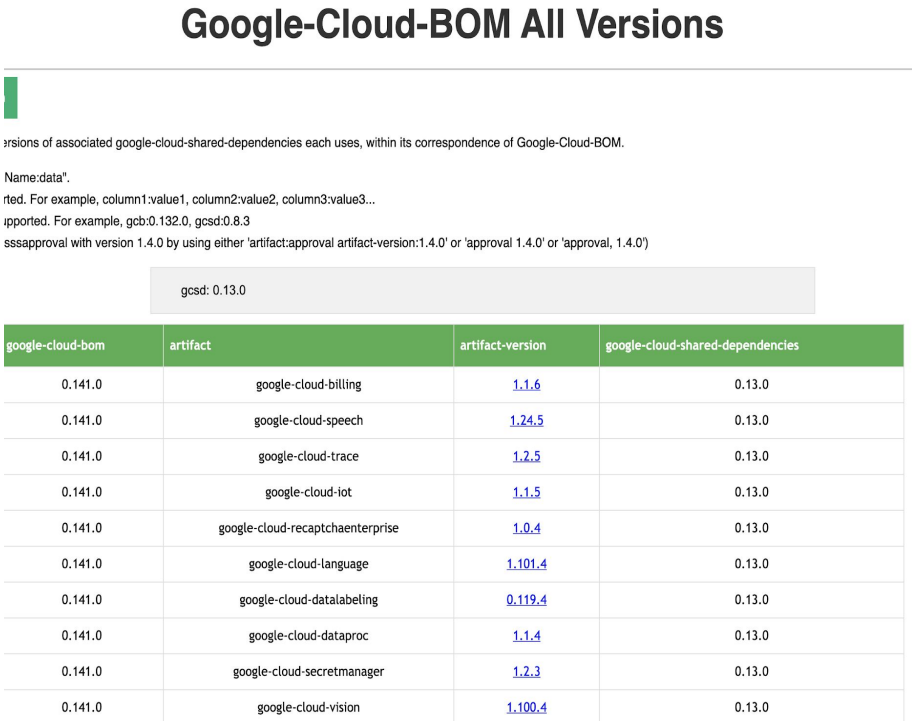
How to achieve client library compatibility?

- If a customer (library consumer) wants to use BigQuery DataTransfer and PubSub Java client libraries to schedule data transfer jobs with realtime PubSub notifications, how do they ensure that the two client libraries are compatible?
  - Solution **1**: Use google-cloud-bom.



How to achieve client library compatibility?

- If a customer (library consumer) wants to use BigQuery DataTransfer and PubSub Java client libraries to schedule data transfer jobs with realtime PubSub notifications, how do they ensure that the two client libraries are compatible?
  - Solution **2**: Use the [google-cloud-bom dashboard] to find out what versions of the client libraries are compatible.
    - If the customer cannot use BOM to manager versions.



Google-Cloud-BOM All Versions

ersions of associated google-cloud-shared-dependencies each uses, within its correspondence of Google-Cloud-BOM.

Name:data".
rted. For example, column1:value1, column2:value2, column3:value3...
ipported. For example, gcb:0.132.0, gcsd:0.8.3
sssapproval with version 1.4.0 by using either 'artifact:approval artifact-version:1.4.0' or 'approval 1.4.0' or 'approval, 1.4.0')

gcsd: 0.13.0

google-cloud-bom	artifact	artifact-version	google-cloud-shared-dependencies
0.141.0	google-cloud-billing	1.1.6	0.13.0
0.141.0	google-cloud-speech	1.24.5	0.13.0
0.141.0	google-cloud-trace	1.2.5	0.13.0
0.141.0	google-cloud-iot	1.1.5	0.13.0
0.141.0	google-cloud-recaptchaenterprise	1.0.4	0.13.0
0.141.0	google-cloud-language	1.101.4	0.13.0
0.141.0	google-cloud-datalabeling	0.119.4	0.13.0
0.141.0	google-cloud-dataproc	1.1.4	0.13.0
0.141.0	google-cloud-secretmanager	1.2.3	0.13.0
0.141.0	google-cloud-vision	1.100.4	0.13.0

Case study: GCP customer Data Fusion

- The customer would like to upgrade to at minimum v1.124.5 to get access to a number of new features released in the BigQuery client library.
- However, upgrading only the BigQuery client caused dependency conflicts with other older versions of GCP client libraries (source):

```
<flogger.system.backend.version>0.3.1</flogger.system.backend.version>
<gcs.connector.version>hadoop2-2.0.0</gcs.connector.version>
<google.cloud.bigtable.version>1.11.0</google.cloud.bigtable.version>
<google.cloud.bigquery.version>1.92.0</google.cloud.bigquery.version>
<google.cloud.pubsub.version>1.92.0</google.cloud.pubsub.version>
<google.cloud.spanner.version>1.37.0</google.cloud.spanner.version>
<google.cloud.speech.version>1.20.0</google.cloud.speech.version>
<google.cloud.storage.version>1.92.0</google.cloud.storage.version>
<google.cloud.datastore.version>1.92.0</google.cloud.datastore.version>
<google.protobuf.java.version>3.4.0</google.protobuf.java.version>
<google.tink.version>1.3.0-rc3</google.tink.version>
<guava.version>27.0.1-jre</guava.version>
```

Case study: GCP customer Data Fusion

- BOM strategy was recommended **but the customer prefers managing their own dependencies and remain flexible in version management**.
- How do we help them to find compatible client libraries that will 100% introduce no dependency conflicts?
  - Use the google-cloud-bom dashboard to:
    1. Identify the version of google-cloud-shared-dependencies BOM used in **google-cloud-bigquery v1.124.5**;
    2. Locate the versions of the other client libraries that use the same version of google-cloud-shared-dependencies BOM:
       Bigtable: 1.18.0
       Pubsub: 1.109.0
       Spanner: 3.0.4
       Speech: 1.24.7
       Storage: 1.113.4
       Datastore: 1.105.1

Debugging Maven plugin in IntelliJ on Medium

- Debugging Maven plugin in IntelliJ

Thank you