

A LONGITUDINAL ANALYSIS OF BLOATED JAVA DEPENDENCIES



César Soto Valero, Thomas Durieux, Benoit Baudry

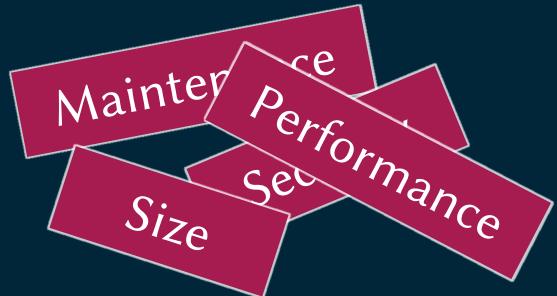


SOFTWARE BLOAT



Code that is packaged in an application
but that is not necessary for building
and running the application.

```
vivek@centos7-box:~$ ls -ld /banking-data/foo
drwxr-xr-x 2 vivek staff 128 Nov 21 18:22 foo
vivek@MacBook-Pro:~$ rm -rf banking-data/
vivek@MacBook-Pro:~$ ls -ld /banking-data/foo
drwxr-xr-x 2 vivek staff 128 Nov 21 18:22 foo
vivek@MacBook-Pro:~$ rm -rf foo
rm: cannot remove '/foo': No such file or directory
vivek@MacBook-Pro:~$ rm -rfv foo
foo/resolv.conf
foo/resolv.com
foo
vivek@MacBook-Pro:~$ ls -ld /banking-data/foo
ls: /banking-data/foo: No such file or directory
ls: foo: No such file or directory
vivek@MacBook-Pro:~$
```



QUESTION



Will this piece of bloated code be needed in the future?

BLOATED DEPENDENCIES OVER TIME

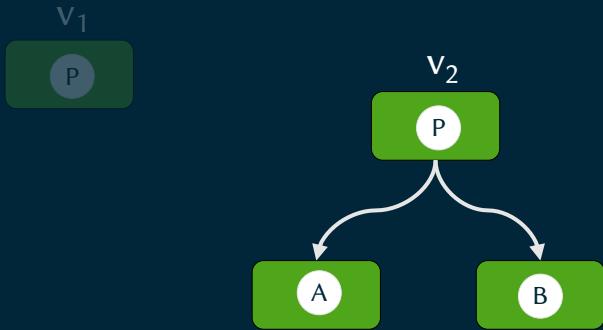


v_1

P



BLOATED DEPENDENCIES OVER TIME

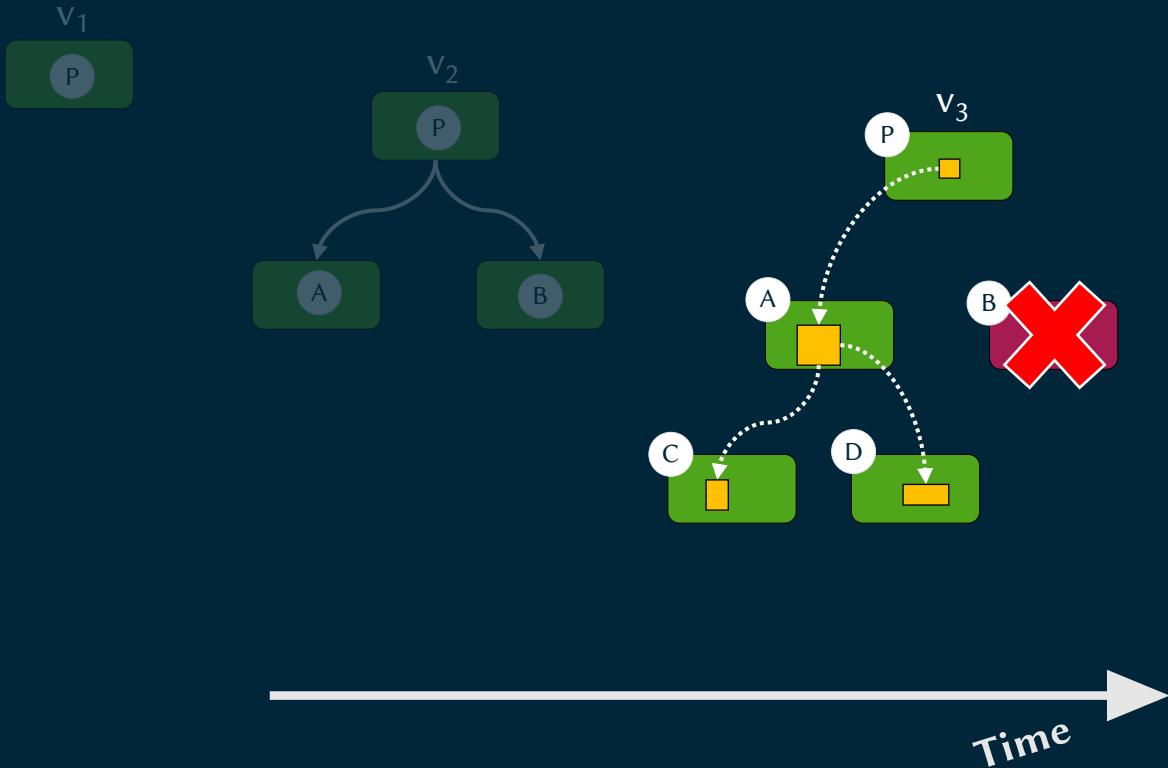


pom.xml

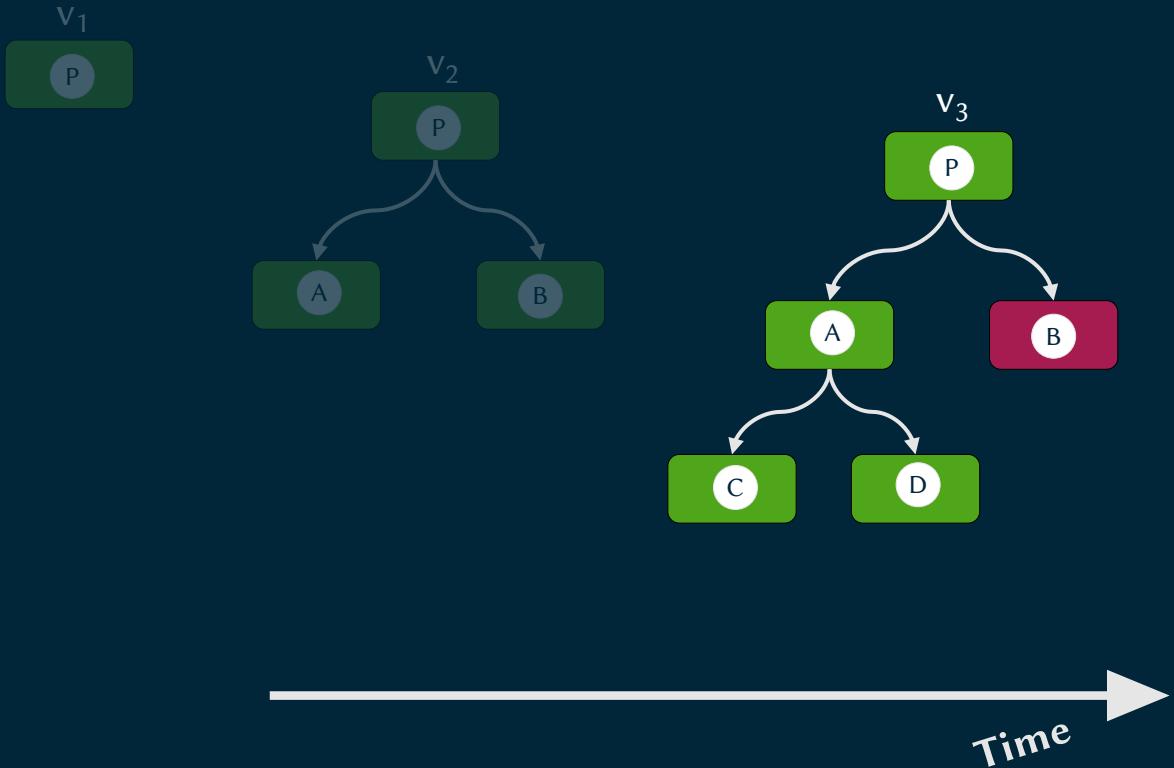
```
<dependencies>
  <dependency>
    <groupId>org.A</groupId>
    <artifactId>A</artifactId>
  </dependency>
  <dependency>
    <groupId>org.B</groupId>
    <artifactId>B</artifactId>
  </dependency>
</dependencies>
```

Time

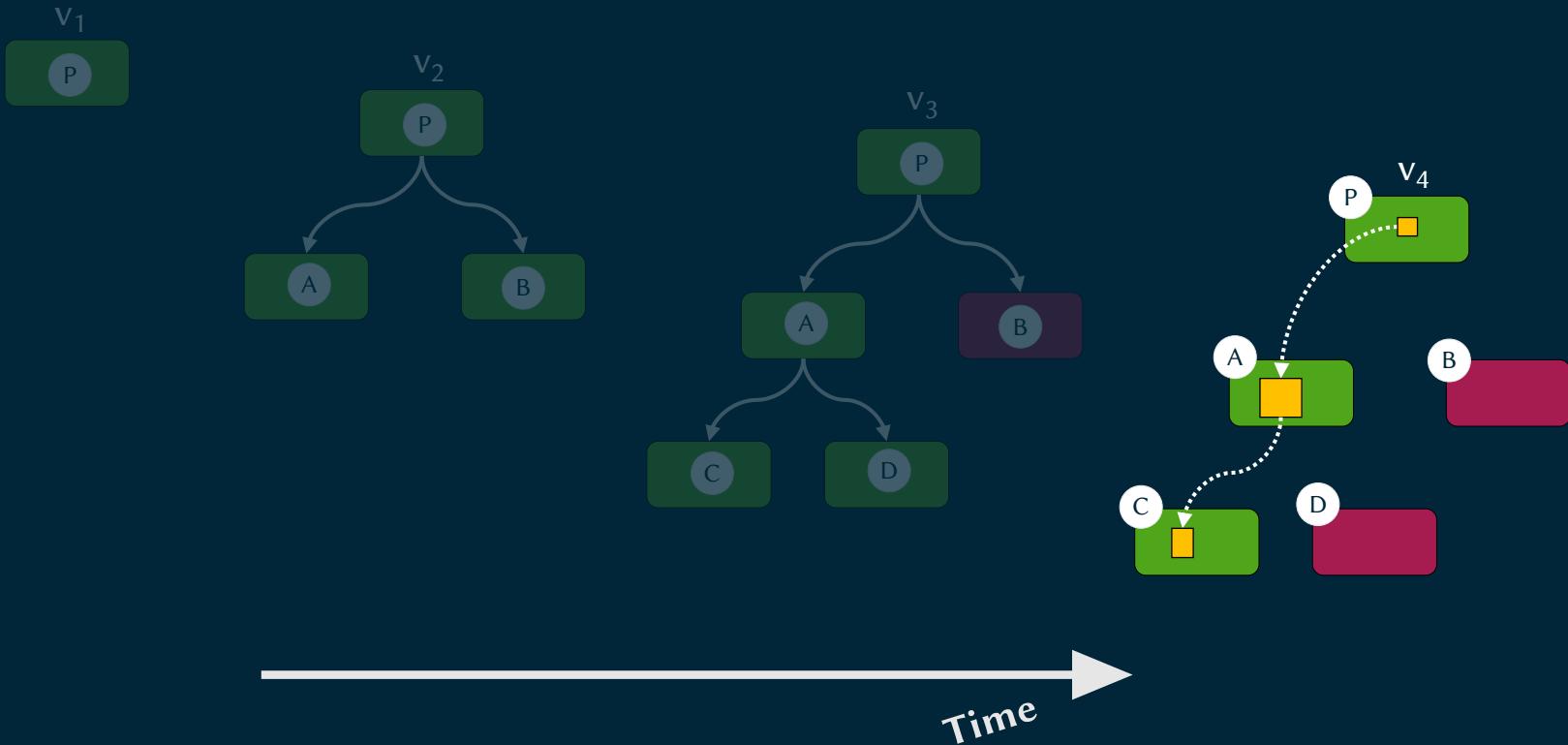
BLOATED DEPENDENCIES OVER TIME



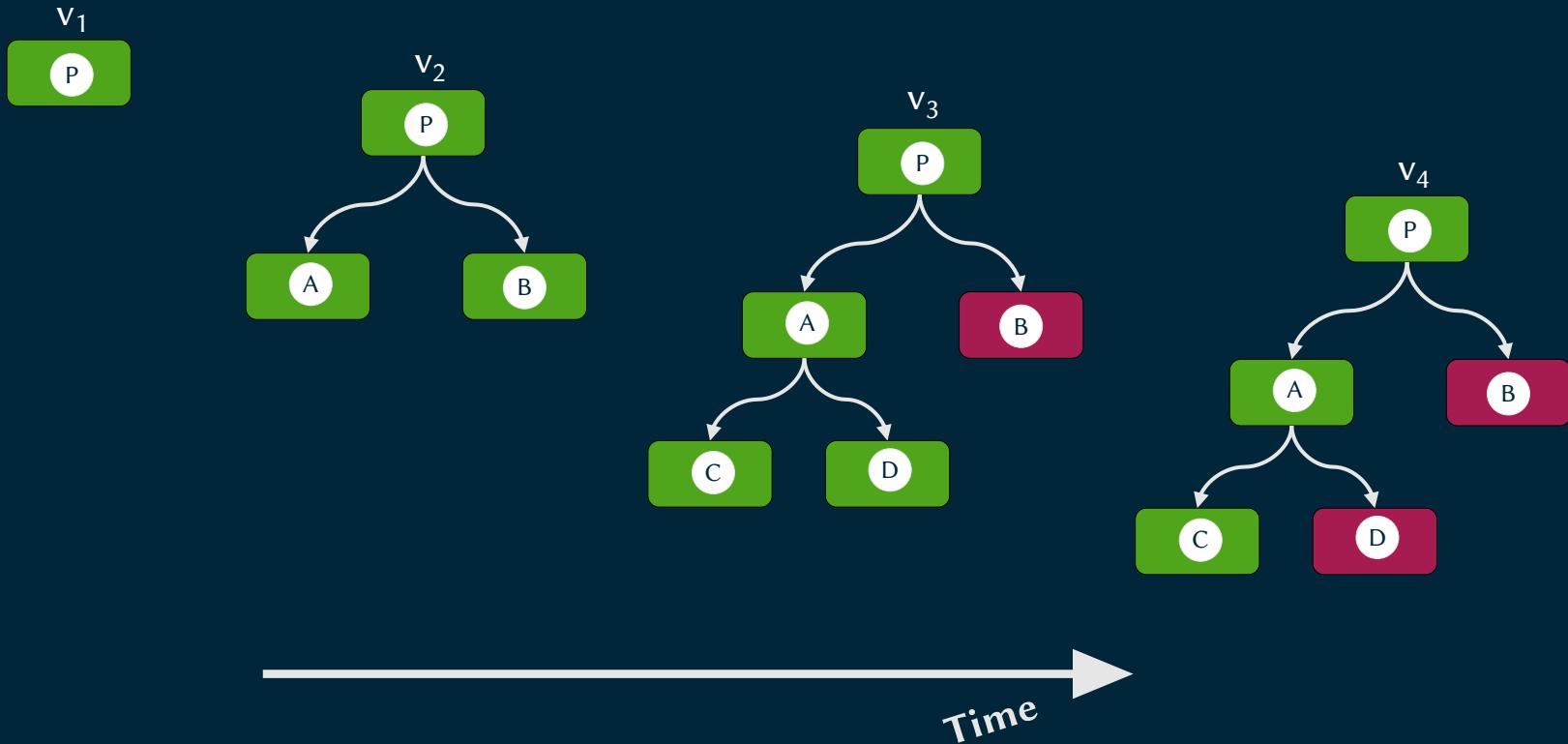
BLOATED DEPENDENCIES OVER TIME



BLOATED DEPENDENCIES OVER TIME



BLOATED DEPENDENCIES OVER TIME



RESEARCH QUESTIONS



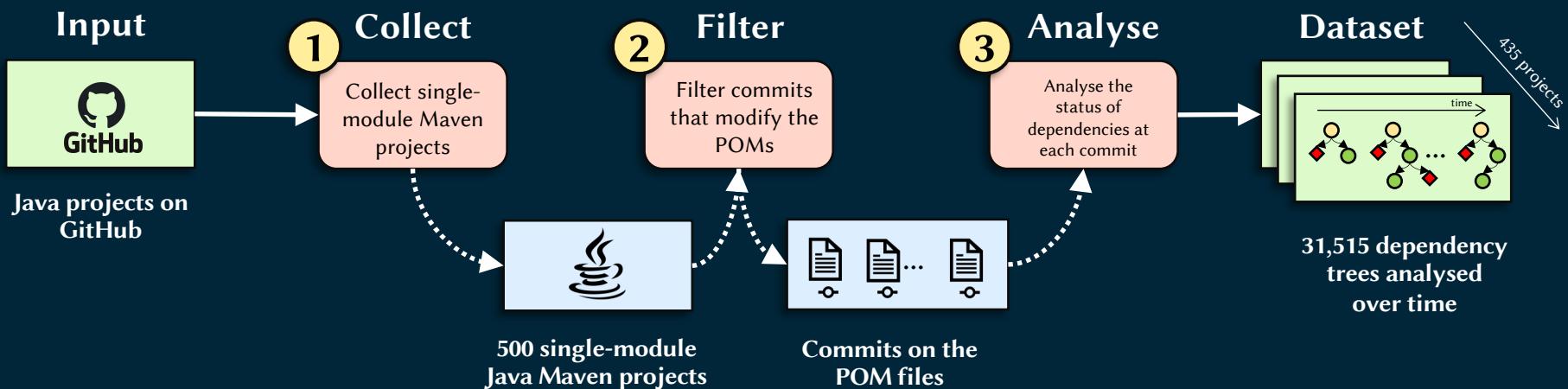
- **RQ1:** How does the amount of bloated dependencies evolve across releases?
- **RQ2:** Do bloated dependencies stay bloated across time?
- **RQ3:** Do developers maintain dependencies that are bloated?
- **RQ4:** What development practices change the usage status of dependencies?

RESEARCH QUESTIONS



- RQ2: Do bloated dependencies stay bloated across time?
- RQ3: Do developers maintain dependencies that are bloated?

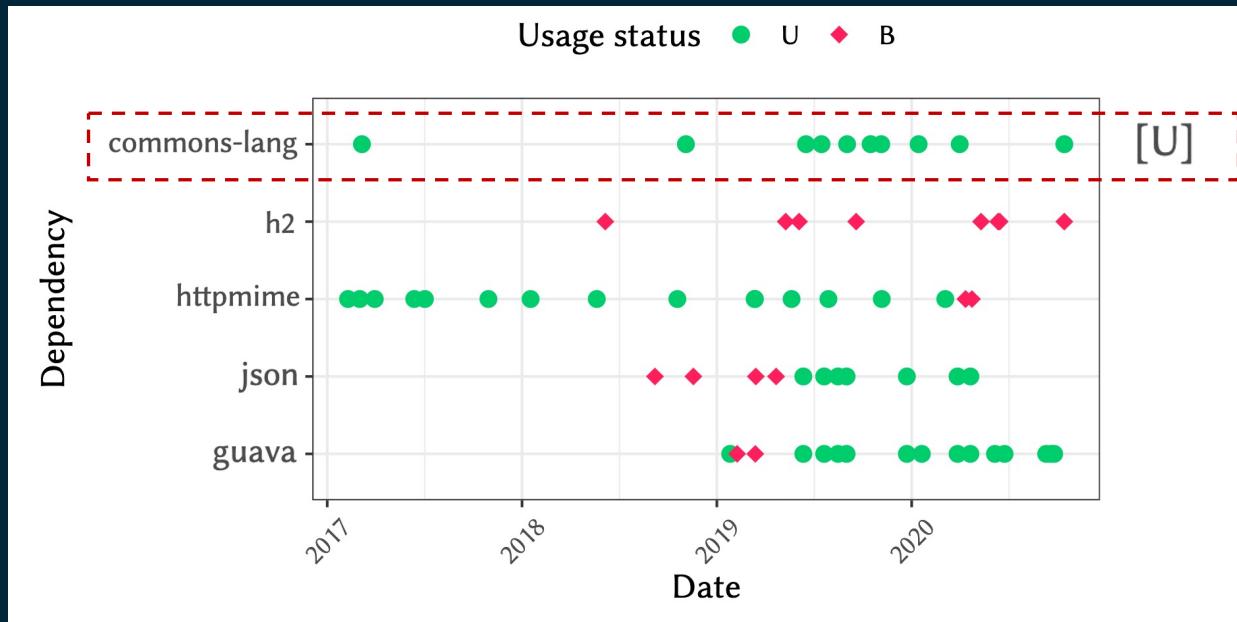
DATA COLLECTION PIPELINE



RESEARCH QUESTION #2



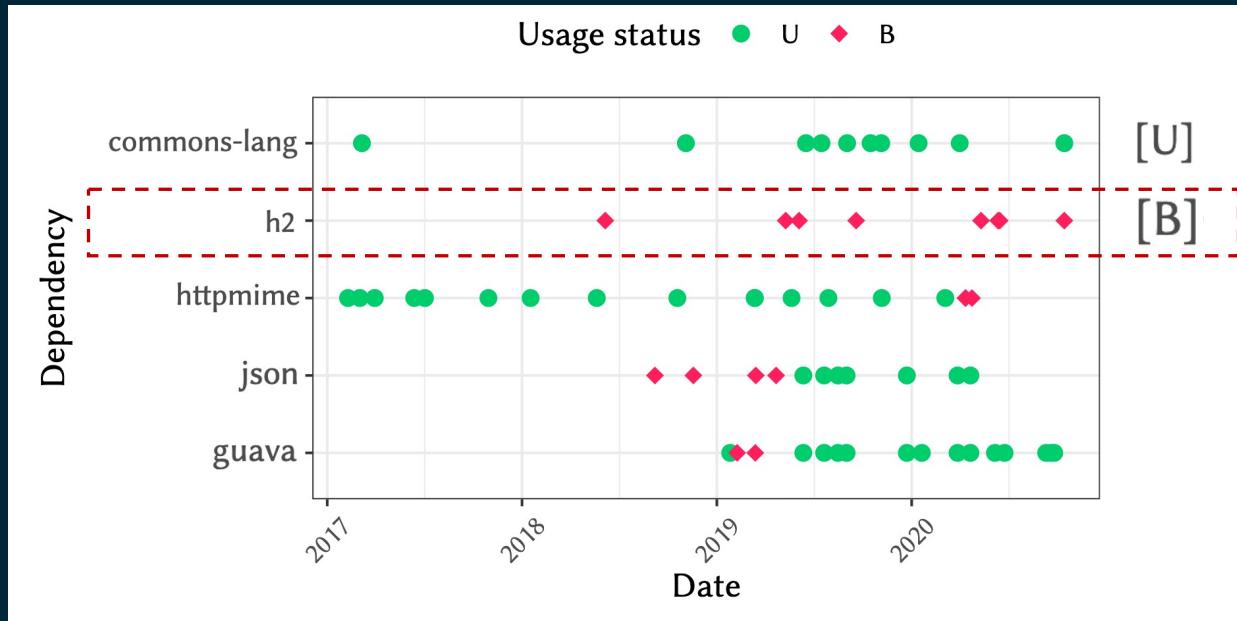
RQ2: Do bloated dependencies stay bloated across time?



RESEARCH QUESTION #2



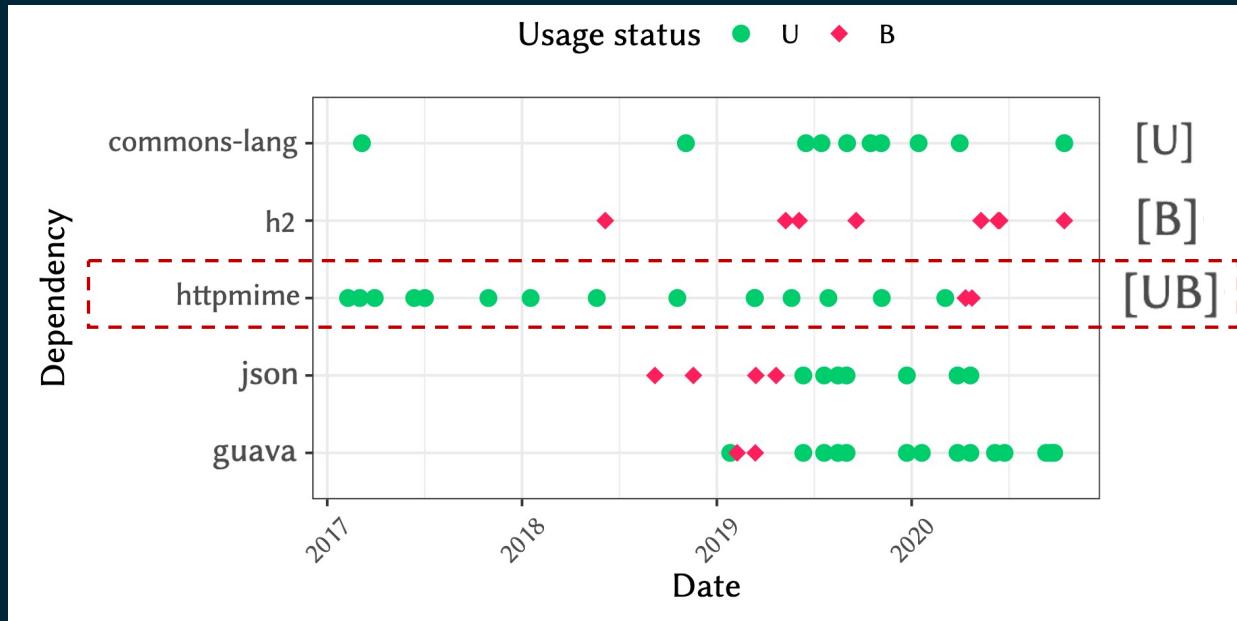
RQ2: Do bloated dependencies stay bloated across time?



RESEARCH QUESTION #2



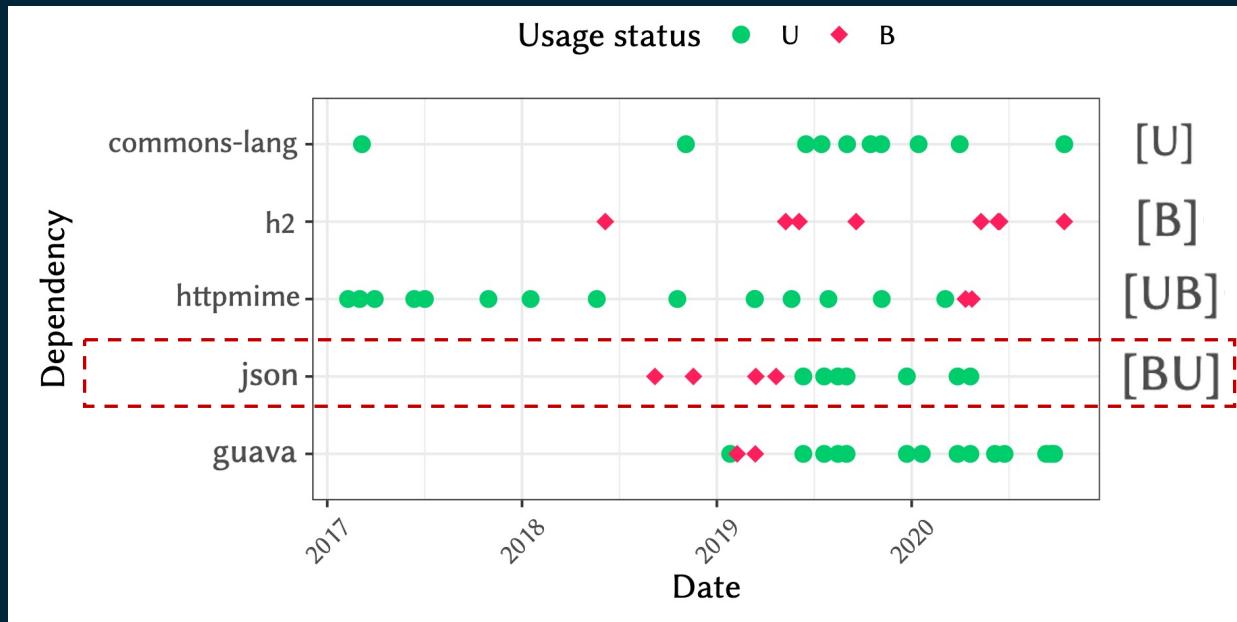
RQ2: Do bloated dependencies stay bloated across time?



RESEARCH QUESTION #2



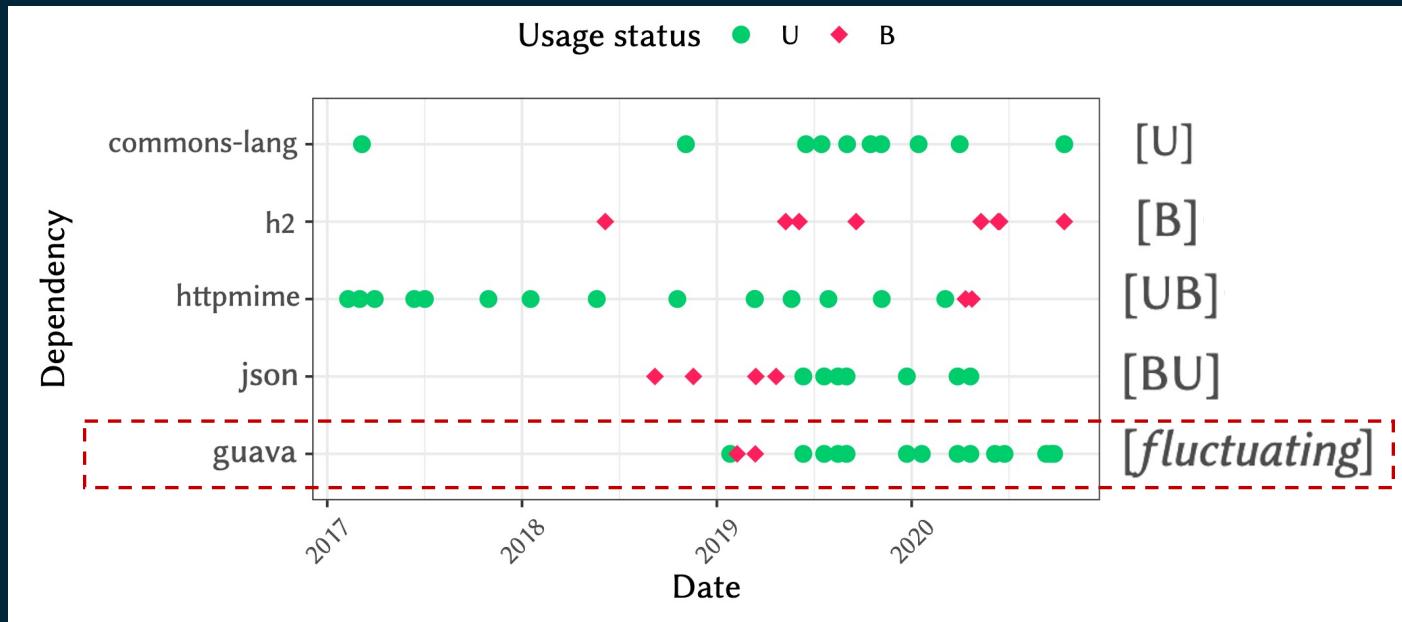
RQ2: Do bloated dependencies stay bloated across time?



RESEARCH QUESTION #2



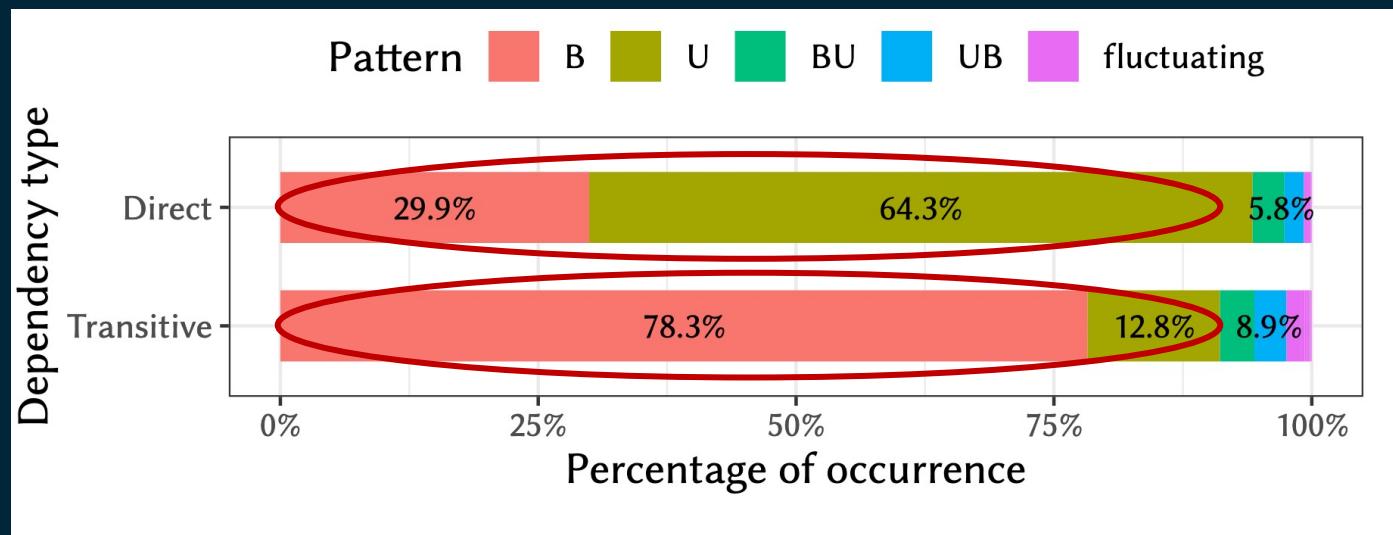
RQ2: Do bloated dependencies stay bloated across time?



RESEARCH QUESTION #2



RQ2: Do bloated dependencies stay bloated across time?



RESEARCH QUESTION #2



89.2% of bloated direct, and 93.3% of the transitive bloated dependencies remain bloated in all subsequent versions of the studied projects.

RESEARCH QUESTION #3



RQ3: Do developers maintain dependencies that are bloated?

```
118 118      <dependency>
119 119          <groupId>commons-io</groupId>
120 120          <artifactId>commons-io</artifactId>
121 -          <version>2.6</version>
121 +          <version>2.7</version>
122 122      </dependency>
```



RESEARCH QUESTION #3



RQ3: Do developers maintain dependencies that are bloated?

com.google.guava:guava

⚠ Open GitHub opened this alert on 30 Mar

⚠ Bump guava from 26.0-jre to 29.0-jre in /core [dependencies](#)
#17 opened on 1 Apr by dependabot · bot

A screenshot of a GitHub dependency alert for the com.google.guava:guava project. It shows an open alert for a dependency update from version 26.0-jre to 29.0-jre. The alert was opened on March 30th by Dependabot. A link to the dependencies section is provided.

Dependabot alerts

⚠ 5 Open ✓ 0 Closed

⚠ com.google.guava:guava
by GitHub core/pom.xml #17

⚠ junit:junit
by GitHub offline/pom.xml #15

Dismiss all ▾

A fix has already been started
No bandwidth to fix this
Risk is tolerable to this project
Vulnerable code is not actually used

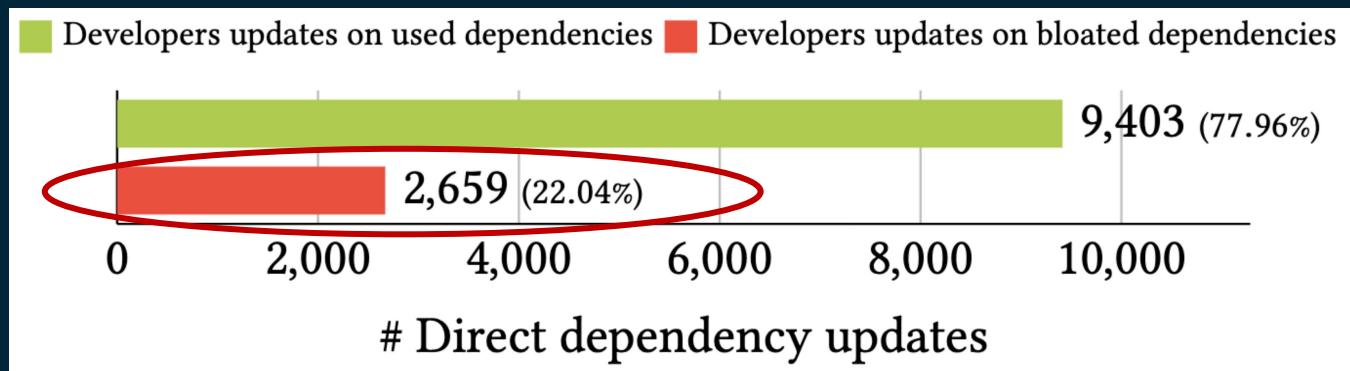
Manage repository vulnerability settings
Manage account notification settings

A screenshot of the Dependabot alerts interface. It shows five open alerts. The first alert is for com.google.guava:guava, with a link to core/pom.xml and issue #17. The second alert is for junit:junit, with a link to offline/pom.xml and issue #15. A dropdown menu is open, showing options related to fixing the alert: "A fix has already been started", "No bandwidth to fix this", "Risk is tolerable to this project", and "Vulnerable code is not actually used". The last two options are part of a redacted area. At the bottom of the dropdown are links to "Manage repository vulnerability settings" and "Manage account notification settings".

RESEARCH QUESTION #3



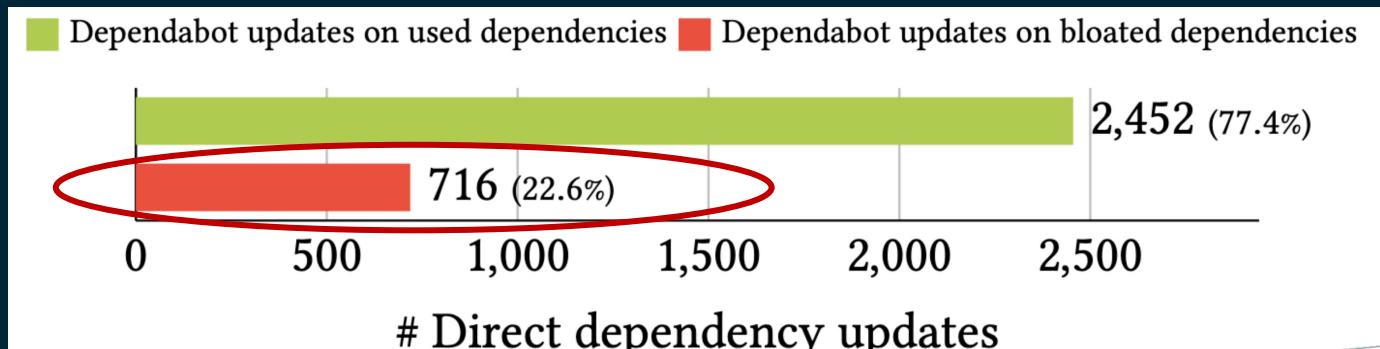
RQ3: Do developers maintain dependencies that are bloated?



RESEARCH QUESTION #3



RQ3: Do developers maintain dependencies that are bloated?



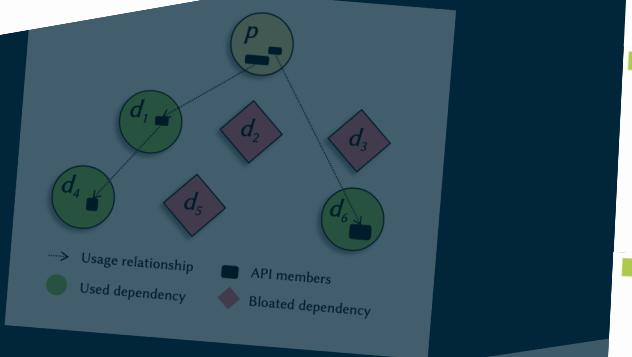
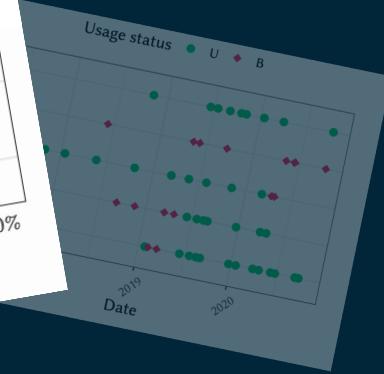
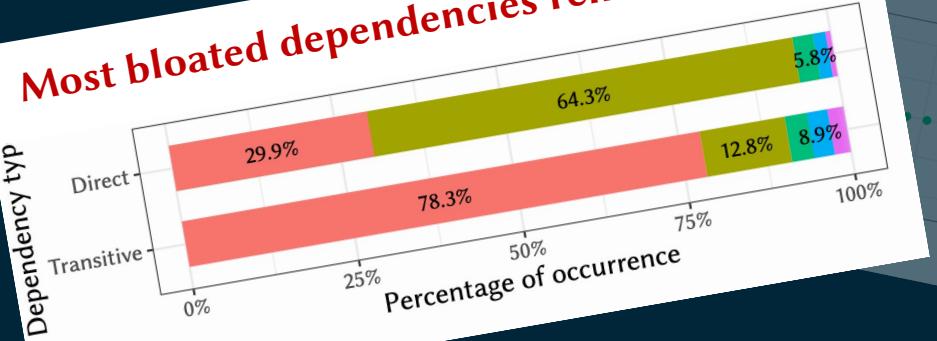
Dependabot

RESEARCH QUESTION #3

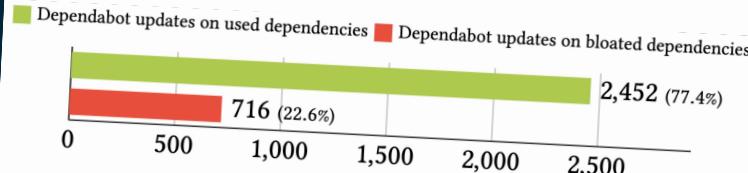
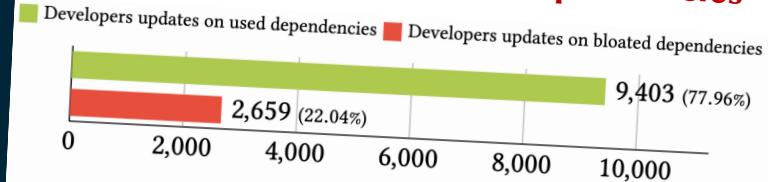


22% of dependency updates performed by developers and bots are made on bloated dependencies.

Most bloated dependencies remain bloated



Developers update bloated dependencies



<dependency>
 <groupId>commons-io</groupId>
 <artifactId>commons-io</artifactId>
 <version>2.6</version>
 <version>2.7</version>
</dependency>