The ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE’21)

A Large-Scale Empirical Study on Library Migrations: Prevalence, Trends, and Rationales

Hao He, Runzhi He, Haiqiao Gu, Minghui Zhou

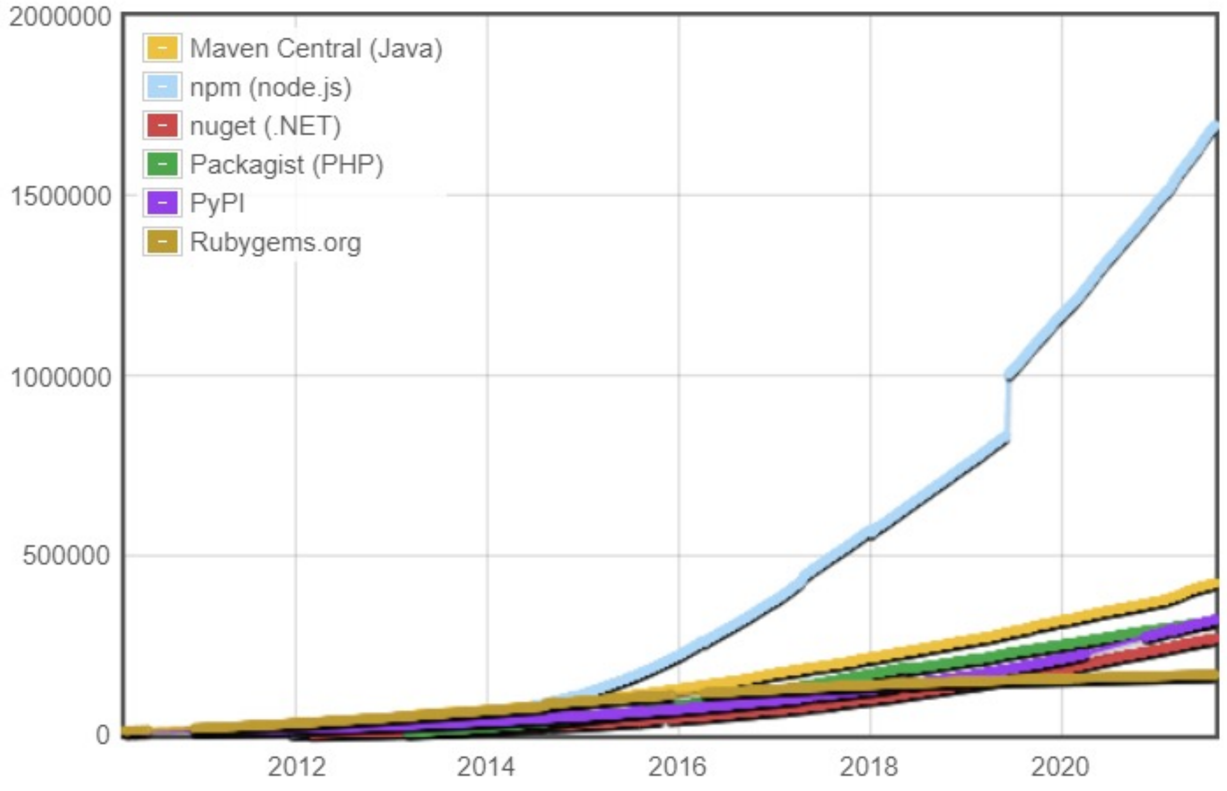
School of Computer Science and Technology, Peking University

Presenter: Hao He (何昊), Ph.D. Candidate

北京大學
PEKING UNIVERSITY

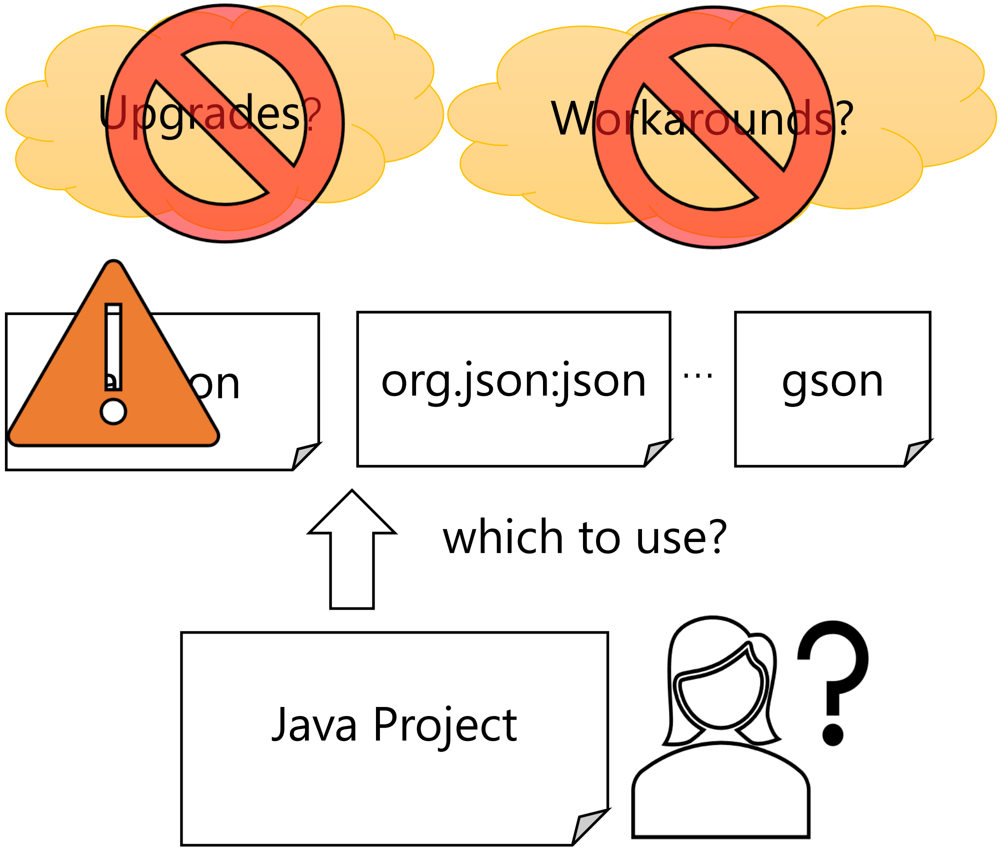
Background: Modern software development are powered by huge and rapidly growing library ecosystems

Library Count (by Ecosystem)



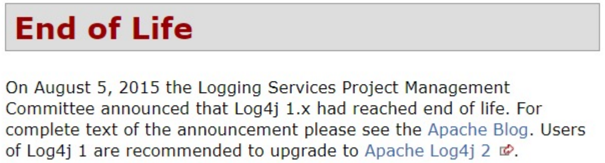
(http://www.modulecounts.com/)

Background: Choosing libraries to use can be hard, and they frequently cause problems after adoption



- Security vulnerabilities
- Lack of maintenance
- License incompatibilities
- Misalignment with requirements

cve.mitre.org
Common Vulnerabilities and Exposures
It takes a community.



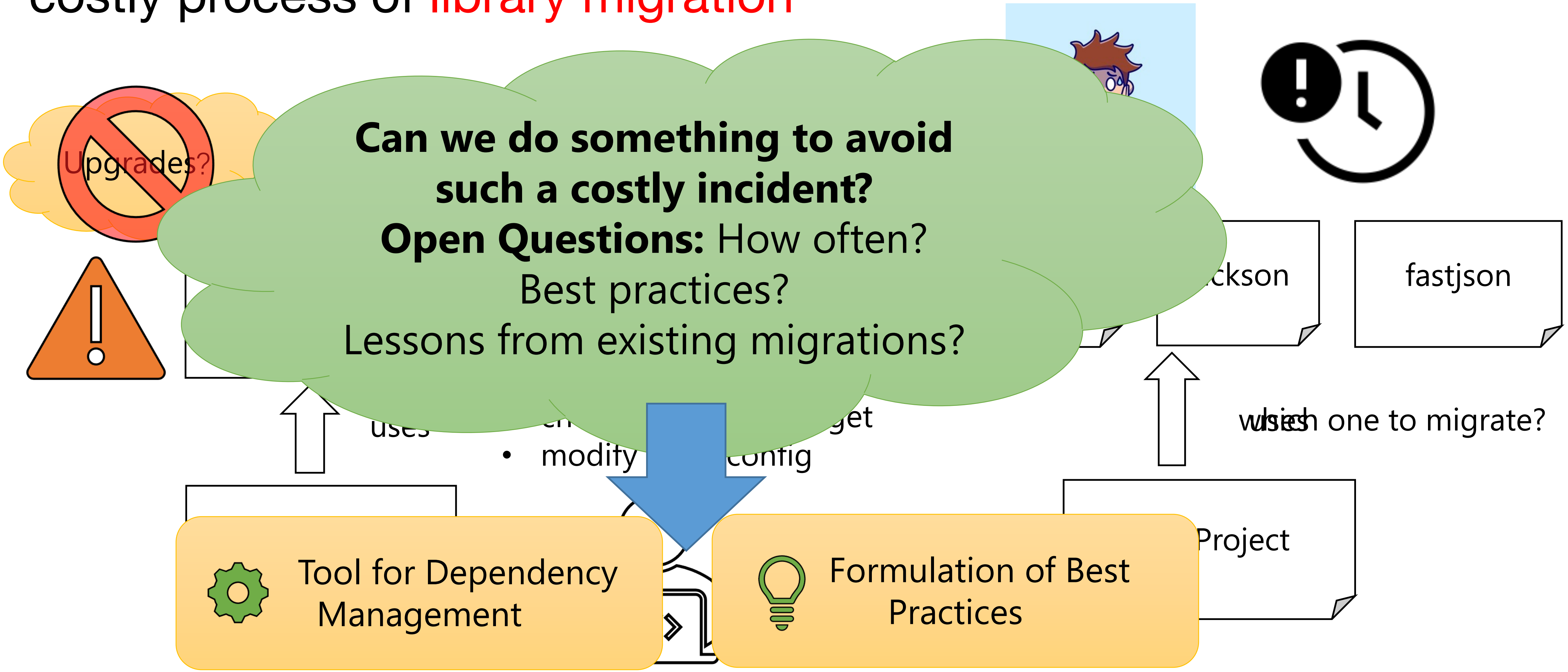
End of Life

On August 5, 2015 the Logging Services Project Management Committee announced that Log4j 1.x had reached end of life. For complete text of the announcement please see the Apache Blog. Users of Log4j 1 are recommended to upgrade to Apache Log4j 2.

GPL 3 Free Software
Free as in Freedom

Scalability?
Performance?
Feature?
Security?

Background: In such scenarios, projects underwent a costly process of library migration



Can we do something to avoid such a costly incident?

Open Questions: How often?

Best practices?

Lessons from existing migrations?

Tool for Dependency Management

Formulation of Best Practices

Methodology: Research Questions

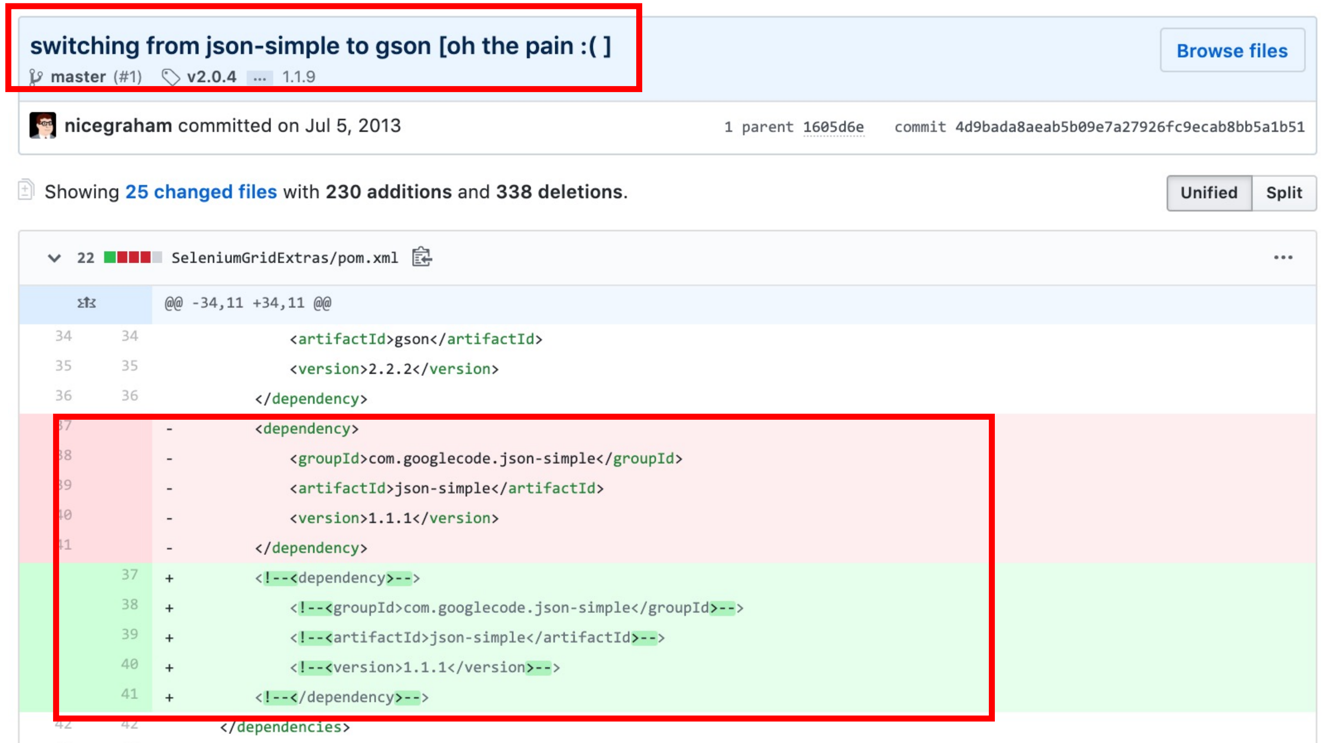
- **RQ1: (Prevalence)** How prevalent are library migrations?
  - **RQ1.1:** How frequently do projects remove a library?
  - **RQ1.2:** How frequently do projects migrate a library?

- **RQ2: (Trends)** How do migrations happen between libraries?

- **RQ3: (Rationales)** What are the frequently mentioned reasons by developers when they conduct a library migration?

Methodology: Important Operationalizations

- A commit is a **migration commit** if its commit message indicates a migration
- $\langle l_1, l_2 \rangle$ is a **migration rule** if there exists a migration commit that conducted the migration from l_1 to l_2



switching from json-simple to gson [oh the pain :( ]

master (#1) v2.0.4 ... 1.1.9

nicegraham committed on Jul 5, 2013 1 parent 1605d6e commit 4d9bada8aeab5b09e7a27926fc9ecab8bb5a1b51

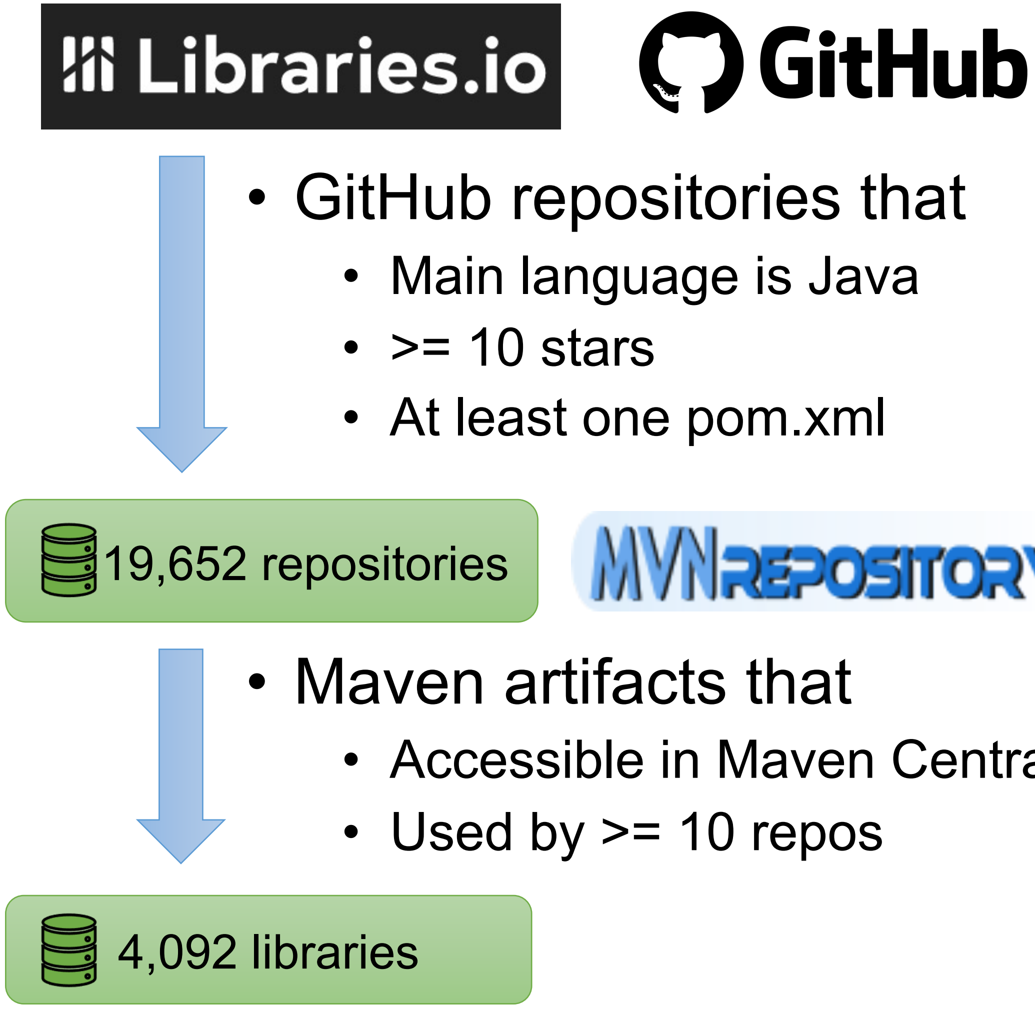
Showing 25 changed files with 230 additions and 338 deletions.

22 SeleniumGridExtras/pom.xml

```
@@ -34,11 +34,11 @@
                <artifactId>gson</artifactId>
                <version>2.2.2</version>
            </dependency>
-           <dependency>
-               <groupId>com.googlecode.json-simple</groupId>
-               <artifactId>json-simple</artifactId>
-               <version>1.1.1</version>
-           </dependency>
+           <!--<dependency>-->
+               <!--<groupId>com.googlecode.json-simple</groupId>-->
+               <!--<artifactId>json-simple</artifactId>-->
+               <!--<version>1.1.1</version>-->
+           <!--</dependency>-->
        </dependencies>
```

- Migration Commit:
  4d9bada8
- Migration Rule:
  <json-simple, gson>

Methodology: Data Collection



Libraries.io

GitHub

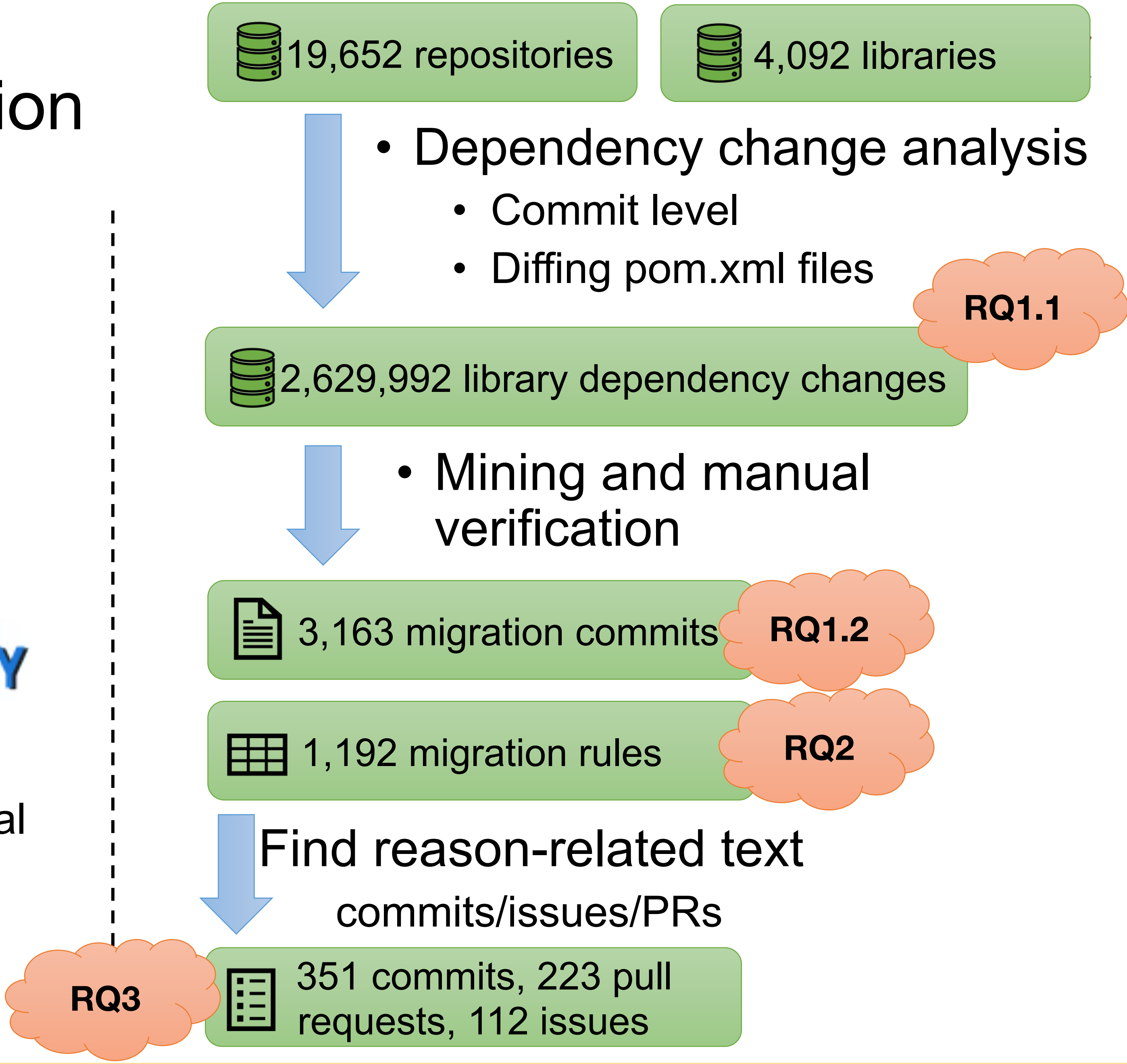
- GitHub repositories that
  - Main language is Java
  - >= 10 stars
  - At least one pom.xml

19,652 repositories

MVNREPOSITORY

- Maven artifacts that
  - Accessible in Maven Central
  - Used by >= 10 repos

4,092 libraries



19,652 repositories

4,092 libraries

- Dependency change analysis
  - Commit level
  - Diffing pom.xml files

2,629,992 library dependency changes

RQ1.1

- Mining and manual verification

3,163 migration commits

RQ1.2

1,192 migration rules

RQ2

Find reason-related text

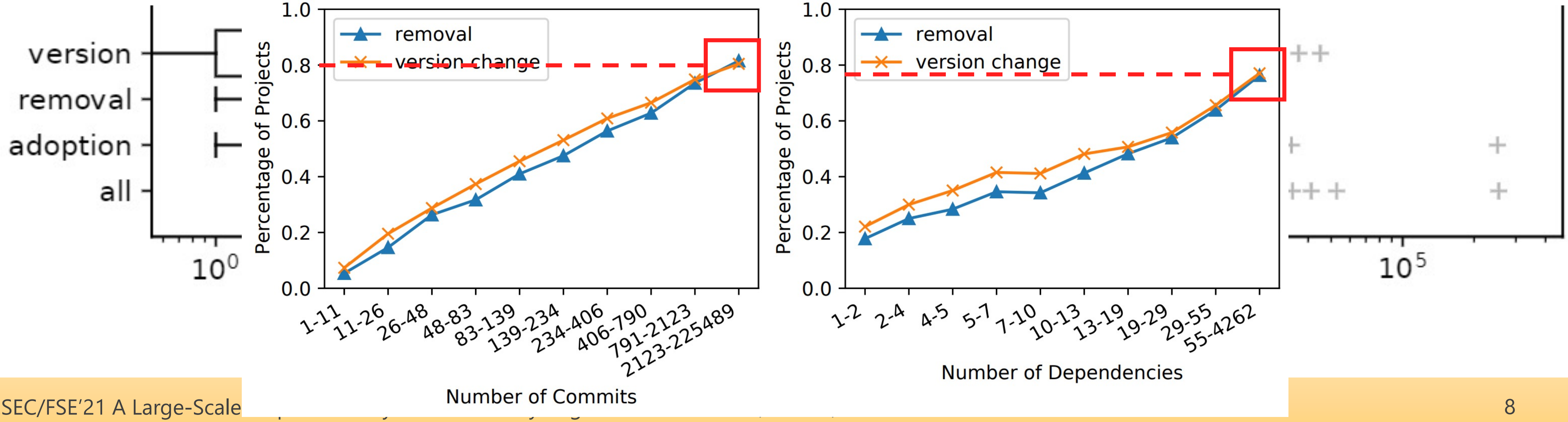
commits/issues/PRs

351 commits, 223 pull requests, 112 issues

RQ3

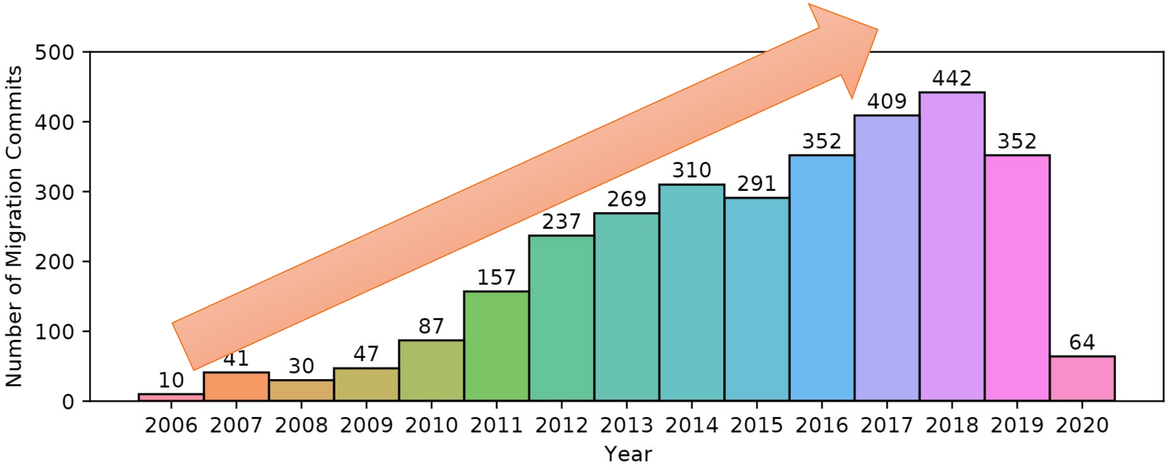
RQ1.1: How frequently do projects remove a library?

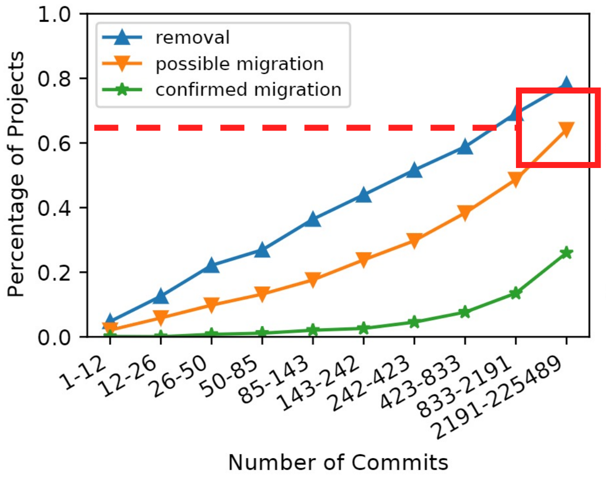
- **Methodology:** Count # of library removals w.r.t. library adoptions and upgrades
- **Main Results:**
  - Among all repositories, 44% have at least one library removal
  - For those repositories, library removals are as frequent as library upgrades
  - Repositories with longer histories and more dependencies are more likely to have library removals, the ratio can be up to 80%

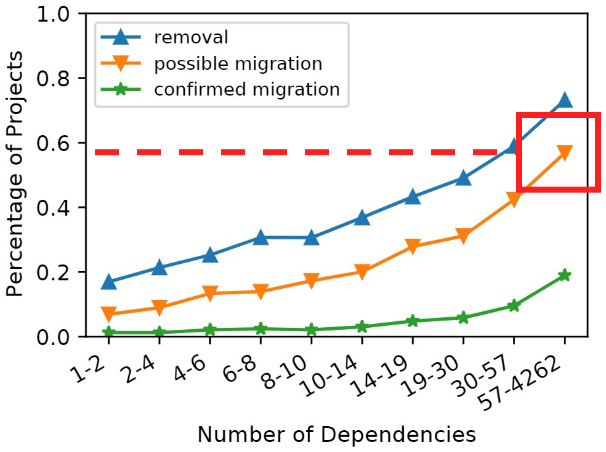


RQ1.2: How frequently do projects migrate a library?

- **Methodology:**
  - Use upper-bound/lower-bound to estimate library migrations, count # of library migrations
- **Main Results:**
  - Among all repositories, 9% to 29% have undergone at least one library migration
  - # of library migrations are increasing over the year
  - Repositories with longer histories and more dependencies are more likely to have library migrations, the ratio can be up to 60%

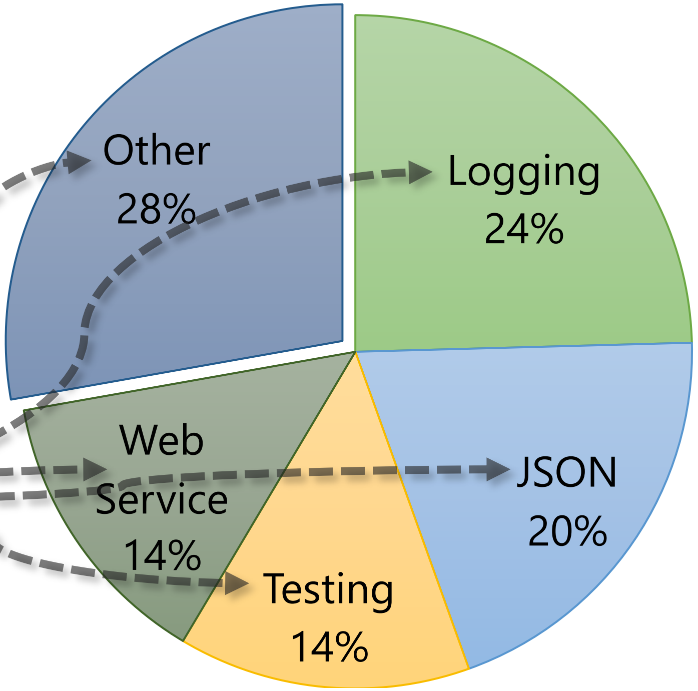


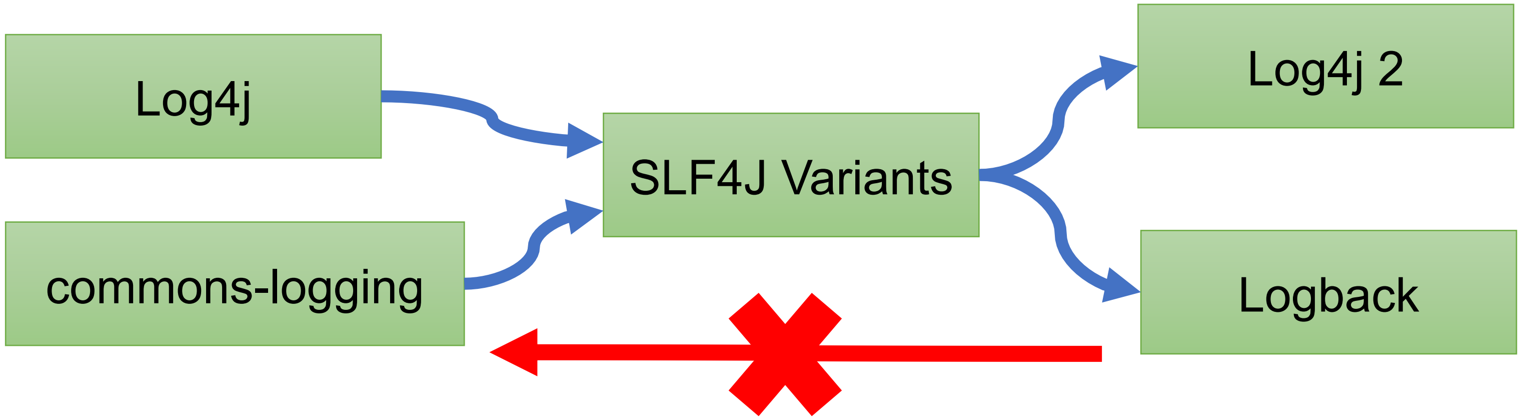




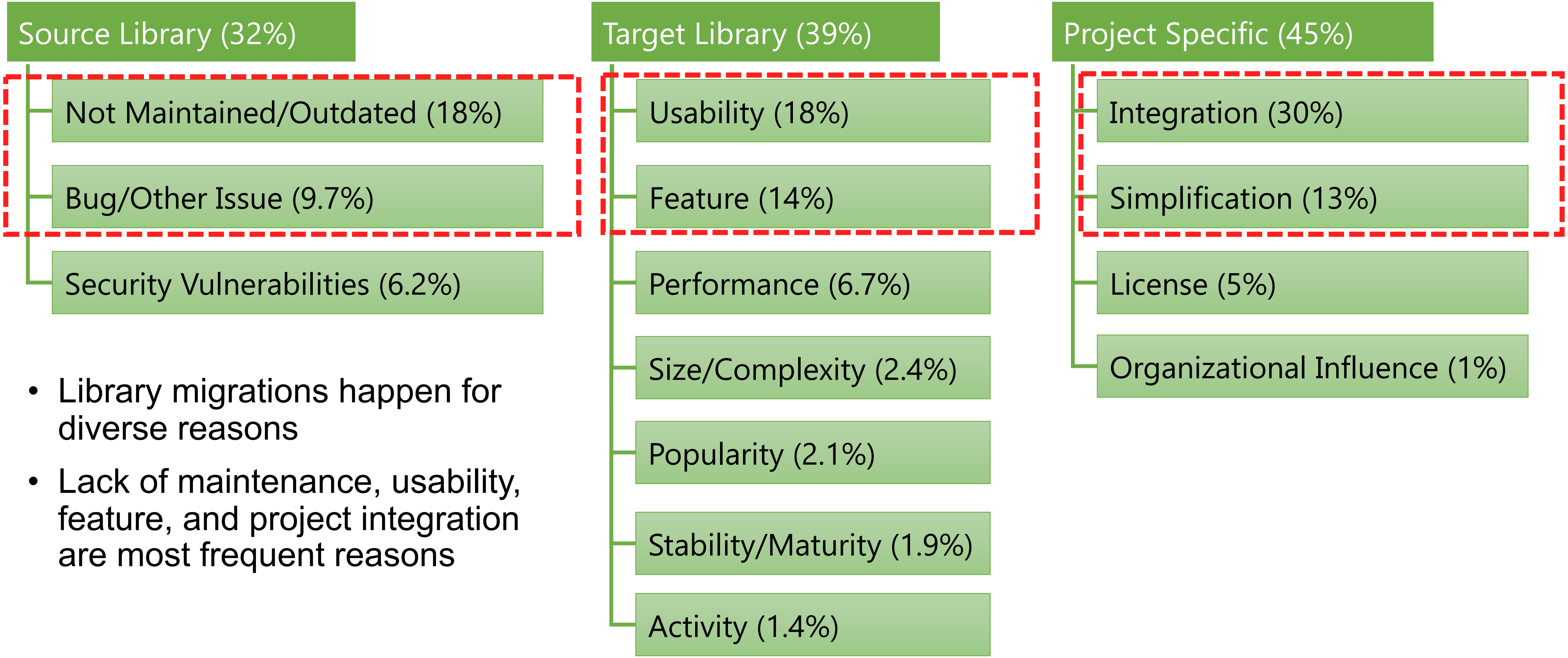
RQ2: How do migrations happen between libraries?

- **Methodology:**
  - Analyze distribution of application domain
  - Analyze migration graph
- **Main Results:**
  - Library migrations happen among diverse domains
  - Most migrations happen among several domains
  - Within same domain, library migrations generally follow a unidirectional trend





RQ3: (Rationale) What are the frequently mentioned reasons when developers conduct a library migration?



Source Library (32%)
- Not Maintained/Outdated (18%)
- Bug/Other Issue (9.7%)
- Security Vulnerabilities (6.2%)

Target Library (39%)
- Usability (18%)
- Feature (14%)
- Performance (6.7%)
- Size/Complexity (2.4%)
- Popularity (2.1%)
- Stability/Maturity (1.9%)
- Activity (1.4%)

Project Specific (45%)
- Integration (30%)
- Simplification (13%)
- License (5%)
- Organizational Influence (1%)

- Library migrations happen for diverse reasons
- Lack of maintenance, usability, feature, and project integration are most frequent reasons

Implications

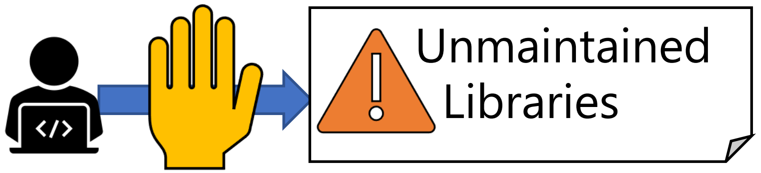
We need proper practice and tool support to ease library adoption and (potentially) avoid library migrations!

Best Practice

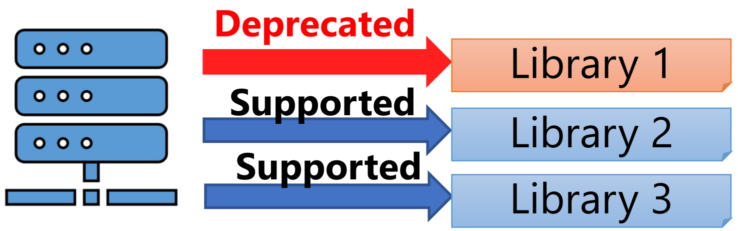
- Tool to show "declining" libraries:
  - Historical removals
  - Historical migrations

- Libraries should document
  - Co-usage practices
  - Common integration pitfalls

Unmaintained Libraries



- Do not use unmaintained libraries



- "Deprecation" mechanism for unmaintained libraries in package hosting platforms

Library Selection

Usability

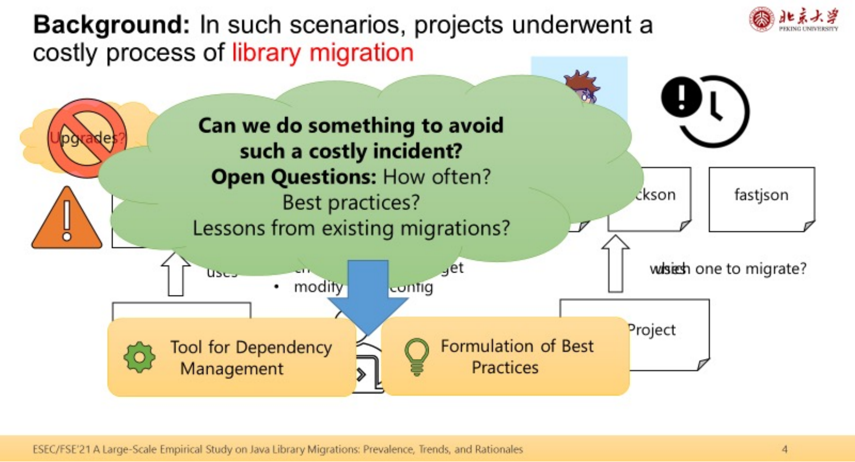
Feature

Performance

License

Summary

We need proper practice and tool support to ease library adoption and (potentially) avoid library migrations!



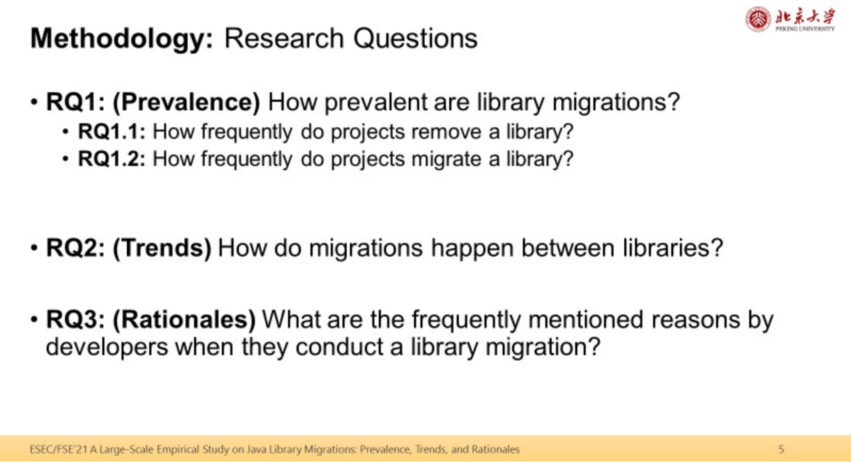
Background: In such scenarios, projects underwent a costly process of library migration

Can we do something to avoid such a costly incident?
Open Questions: How often?
Best practices?
Lessons from existing migrations?

Tool for Dependency Management

Formulation of Best Practices

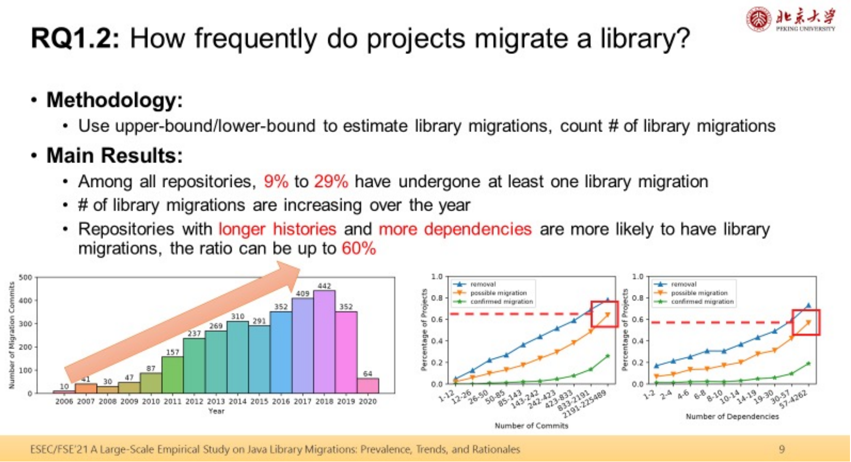
ESEC/FSE'21 A Large-Scale Empirical Study on Java Library Migrations: Prevalence, Trends, and Rationales 4



Methodology: Research Questions

- **RQ1: (Prevalence)** How prevalent are library migrations?
  - **RQ1.1:** How frequently do projects remove a library?
  - **RQ1.2:** How frequently do projects migrate a library?
- **RQ2: (Trends)** How do migrations happen between libraries?
- **RQ3: (Rationales)** What are the frequently mentioned reasons by developers when they conduct a library migration?

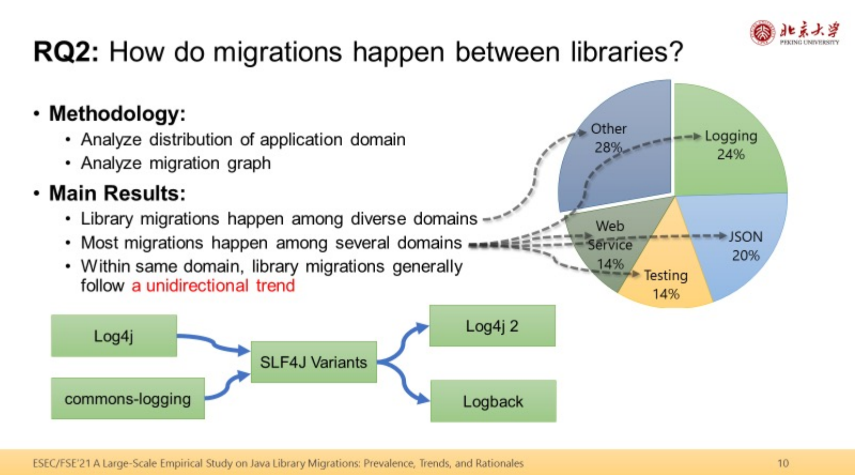
ESEC/FSE'21 A Large-Scale Empirical Study on Java Library Migrations: Prevalence, Trends, and Rationales 5



RQ1.2: How frequently do projects migrate a library?

- **Methodology:**
  - Use upper-bound/lower-bound to estimate library migrations, count # of library migrations
- **Main Results:**
  - Among all repositories, 9% to 29% have undergone at least one library migration
  - # of library migrations are increasing over the year
  - Repositories with longer histories and more dependencies are more likely to have library migrations, the ratio can be up to 60%

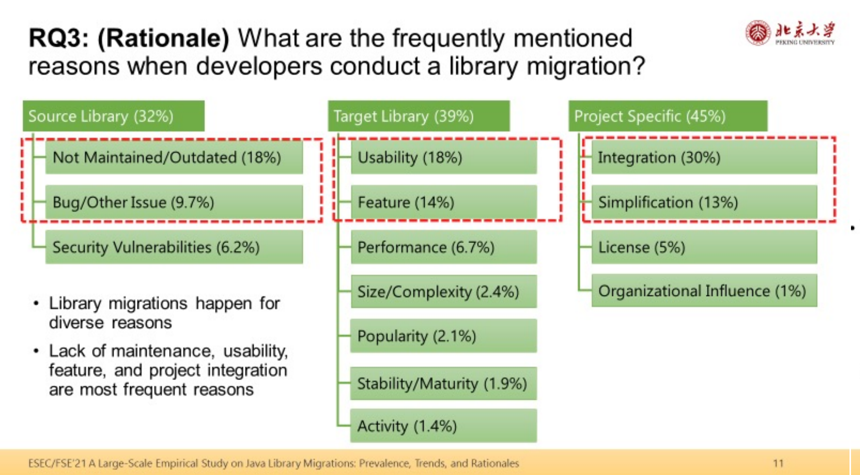
ESEC/FSE'21 A Large-Scale Empirical Study on Java Library Migrations: Prevalence, Trends, and Rationales 9



RQ2: How do migrations happen between libraries?

- **Methodology:**
  - Analyze distribution of application domain
  - Analyze migration graph
- **Main Results:**
  - Library migrations happen among diverse domains
  - Most migrations happen among several domains
  - Within same domain, library migrations generally follow a unidirectional trend

ESEC/FSE'21 A Large-Scale Empirical Study on Java Library Migrations: Prevalence, Trends, and Rationales 10



RQ3: (Rationale) What are the frequently mentioned reasons when developers conduct a library migration?

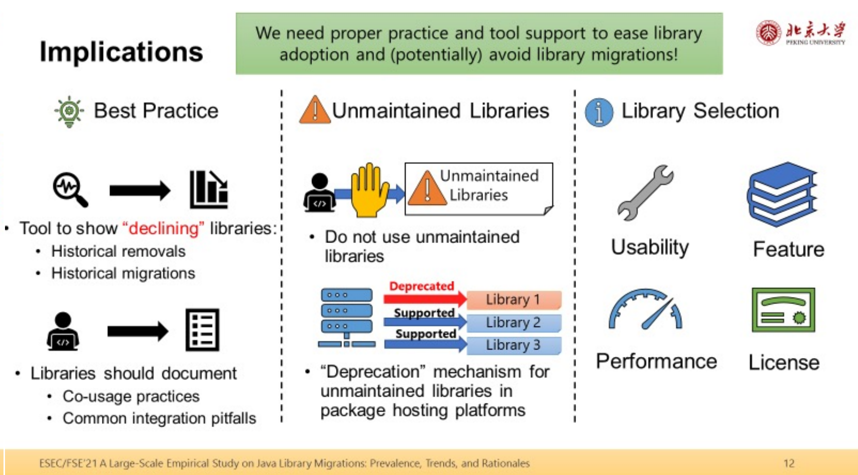
Source Library (32%)
- Not Maintained/Outdated (18%)
- Bug/Other Issue (9.7%)
- Security Vulnerabilities (6.2%)

Target Library (39%)
- Usability (18%)
- Feature (14%)
- Performance (6.7%)
- Size/Complexity (2.4%)
- Popularity (2.1%)
- Stability/Maturity (1.9%)
- Activity (1.4%)

Project Specific (45%)
- Integration (30%)
- Simplification (13%)
- License (5%)
- Organizational Influence (1%)

- Library migrations happen for diverse reasons
- Lack of maintenance, usability, feature, and project integration are most frequent reasons

ESEC/FSE'21 A Large-Scale Empirical Study on Java Library Migrations: Prevalence, Trends, and Rationales 11



Implications

We need proper practice and tool support to ease library adoption and (potentially) avoid library migrations!

Best Practice
- Tool to show "declining" libraries:
  - Historical removals
  - Historical migrations
- Libraries should document
  - Co-usage practices
  - Common integration pitfalls

Unmaintained Libraries
- Do not use unmaintained libraries
- "Deprecation" mechanism for unmaintained libraries in package hosting platforms

Library Selection

ESEC/FSE'21 A Large-Scale Empirical Study on Java Library Migrations: Prevalence, Trends, and Rationales 12

Hao He heh@pku.edu.cn Runzhi He rzhe@pku.edu.cn
Haiqiao Gu ghq@stu.pku.edu.cn Minghui Zhou zhmh@pku.edu.cn