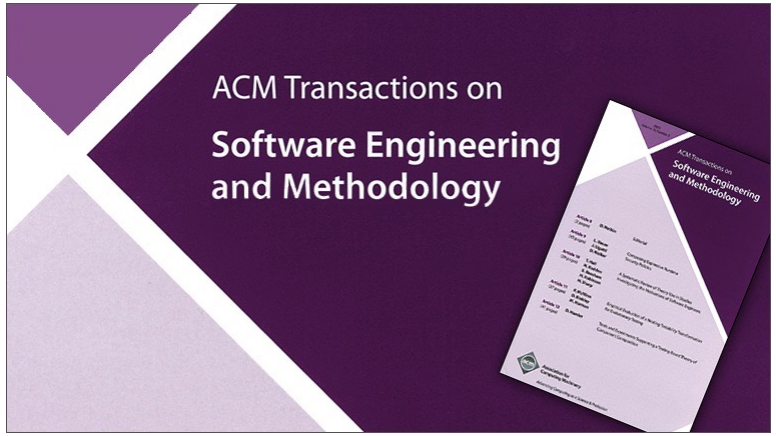
Client-Specific Upgrade Compatibility Checking via Knowledge-Guided Discovery

Chenguang Zhu[1], Mengshi Zhang[2], Xiuheng Wu[2], Xiufeng Xu[2], **Yi Li**[3]

1. The University of Texas at Austin, USA
2. Meta Platforms, USA
3. Nanyang Technological University, Singapore



ICSE 2023

May 18, 2023

To upgrade, or not to upgrade

50%

A recent study[1] shows **50%** of the 408 studied open-source Java projects break after an upgrade to library

40%

40% of the breakages were runtime test failures (not caught by compilers)

82%

In another study[2], **82%** of the developers of the studied systems keep outdated dependencies, leaving system open to zero-day attacks

[1] Alex Gyori, Owolabi Legunsen, Farah Hariri, and Darko Marinov. Evaluating regression test selection opportunities in a very large open-source ecosystem. In ISSRE'18.
[2] Raula Gaikovina Kula, Daniel M. German, Ali Ouni, Takashi Ishio, and Katsuro Inoue. Do developers update their library dependencies? In Empirical Software Engineering'18

Client-Specific Library Upgrade Incompatibility

- Will a library upgrade break a specific client (resulting in different behaviors)?

Client1:

```
if (lib(s1, s2).length() > 0) {
  …
}
```

Incompatible for Client1 (may throw NullPointerException)

V0

```
lib(s1, s2) {
    if (s1 ∩ s2 == ∅) return “”;
    else return …
}
```

Upgrade

V1

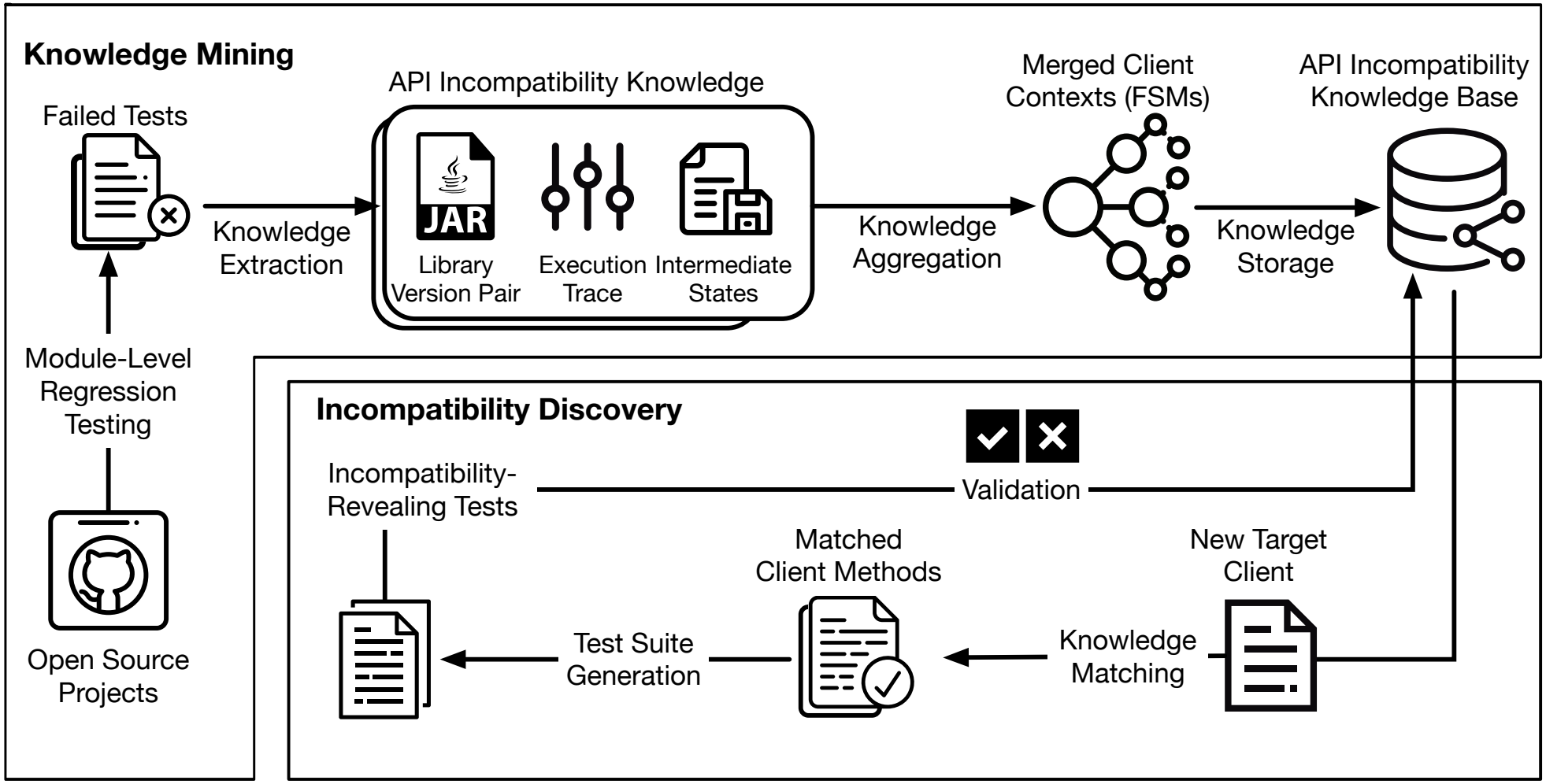
```
--- if (s1 ∩ s2 == ∅) return “”;
+++ if (s1 ∩ s2 == ∅) return null;
```

Client2:

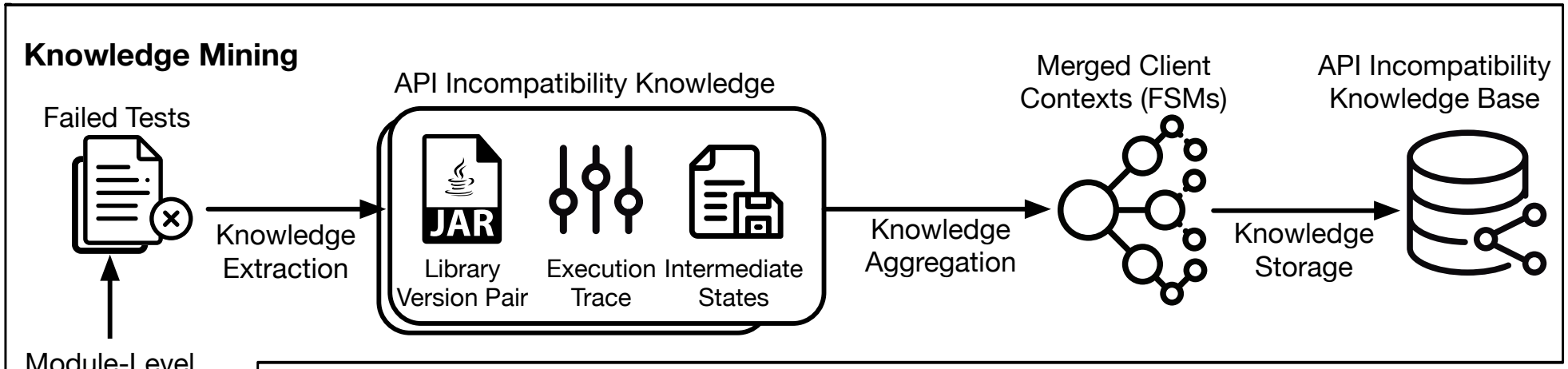
```
lib(
  x.concat("abc"),
  y.concat("abc")
);
```

Compatible for Client2

Our Solution: Client-Specific Compatibility Checking (CompCheck)



Our Solution: Client-Specific Compatibility Checking (CompCheck)



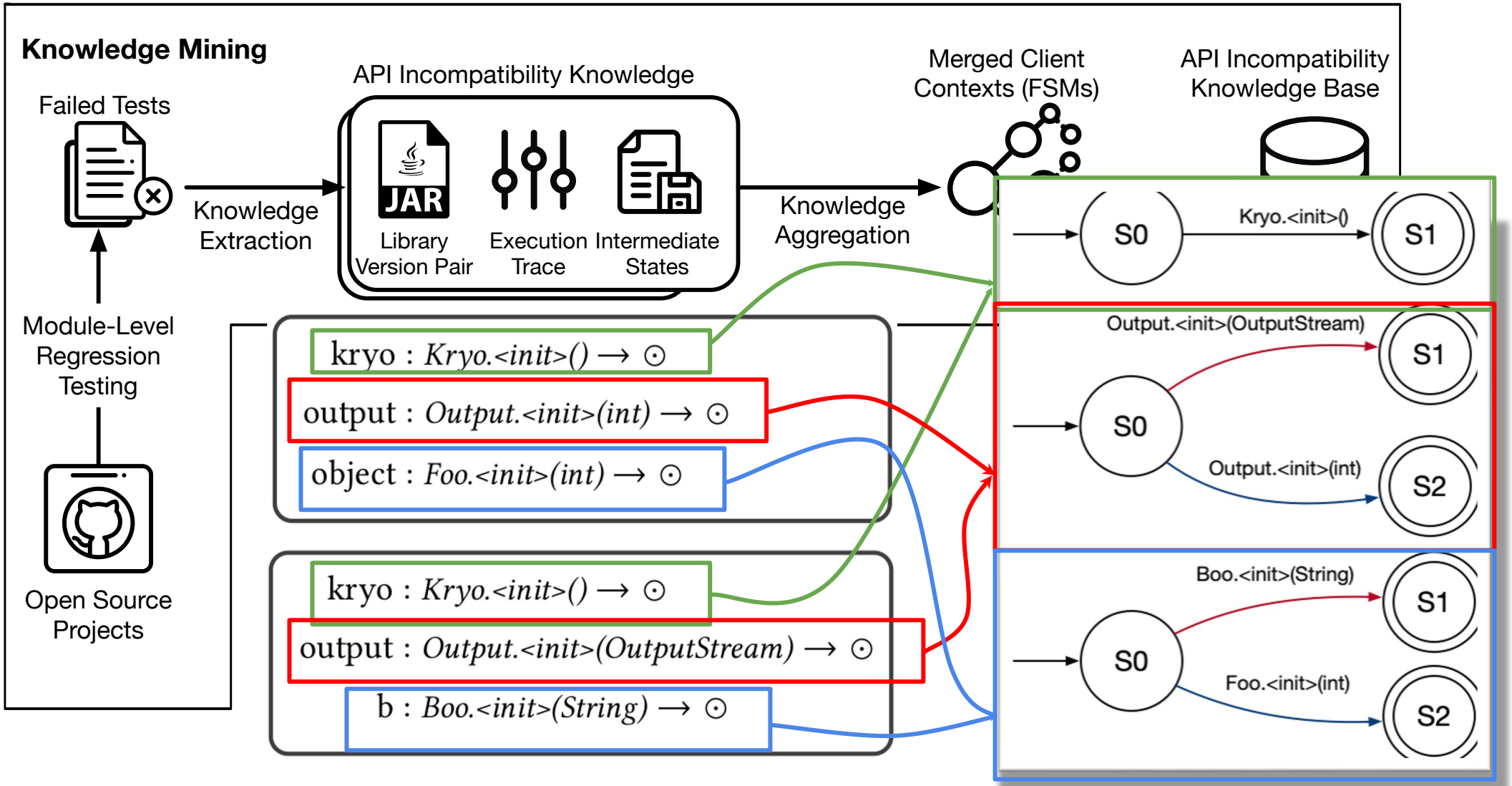

```
1  public void testKryo() {
2    Kryo kryo = new Kryo();
3    ByteArrayOutputStream bytes = new
           ByteArrayOutputStream();
4    Output output = new Output(bytes);
5    Boo b = new Boo("hello");
6    kryo.writeObject(output, b);
7    output.close();
8    ...
9  }
```

kryo : *Kryo.<init>()* → ⊙

output : *Output.<init>(OutputStream)* → ⊙

b : *Boo.<init>(String)* → ⊙

Our Solution: Client-Specific Compatibility Checking (CompCheck)

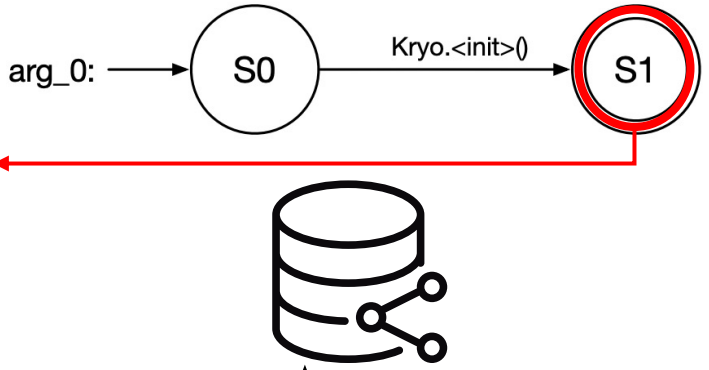


Our Solution: Client-Specific Compatibility Checking (CompCheck)

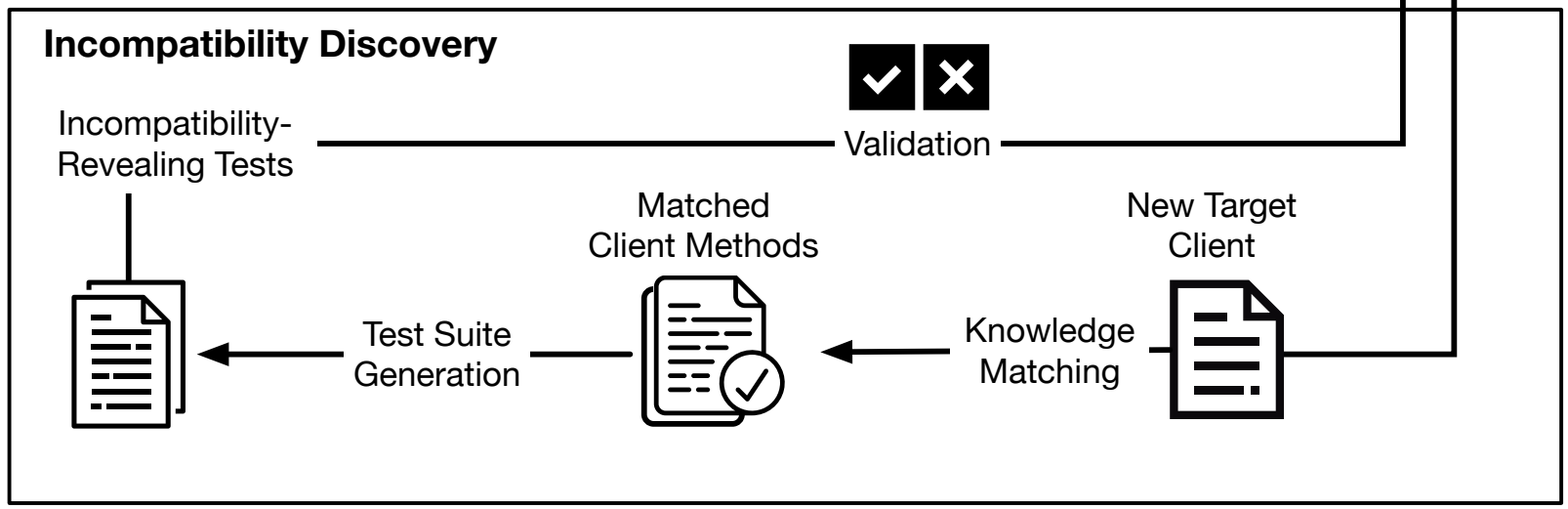
```
1  public static byte[] serialize(Kryo kryo, Object o) {
2      ...
3      Output output = new Output(4096);
4      kryo.writeObject(output, o);
5      output.flush();
6      return output.getBuffer();
7  }
```

A new client method with matching context

```
1  @Test
2  public void test0() {
3      Kryo kryo = loadObj(arg_0_S1);
4      Boo b = loadObj(arg_2_S1);
5      serialize(kryo, b);
6  }
```



New test generated (by reusing stored states)



Evaluation Subjects

- 24 Backward Incompatible APIs from 8 popular libraries

protostuff

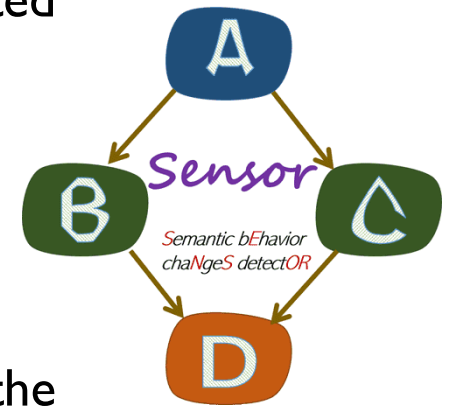
args4j

• • •

- 35 high-starred Java client projects on GitHub, having 202 call sites in total

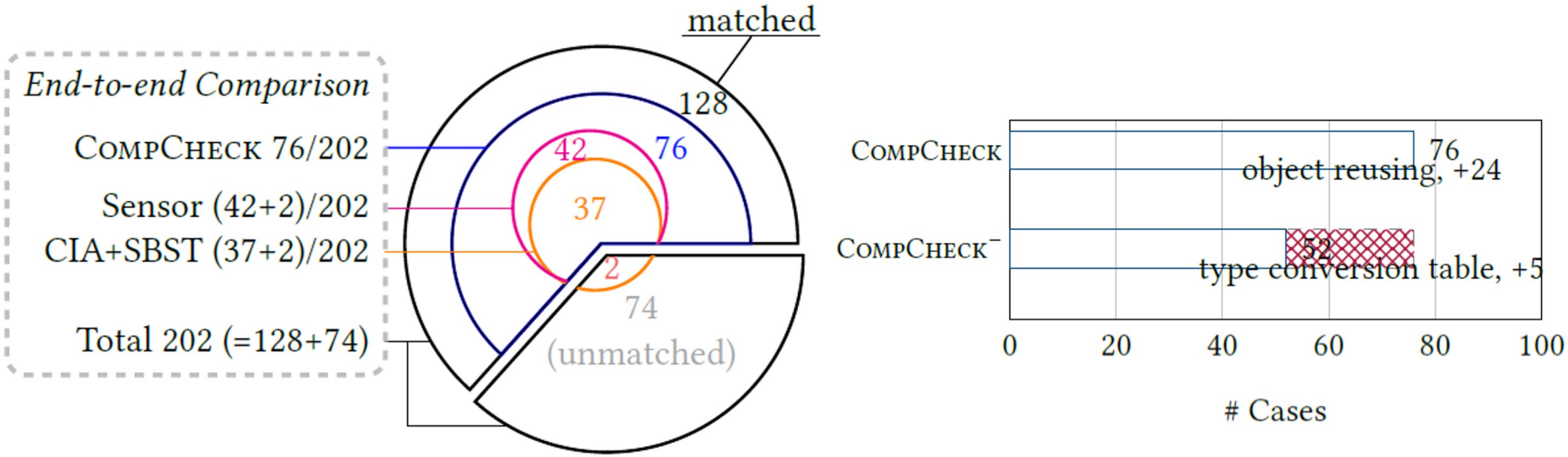
Evaluation Baselines

- Baselines
  - **Sensor**[1]: generating tests to reveal library dependency conflicts
  - **CIA+SBST**: Uses change impact analysis (CIA) to find the call sites affected by the library upgrade, then perform search-based test generation
  - **CompCheck--**: Disable object reusing of CompCheck
- Comparison Goals
  - **CompCheck vs Sensor vs CIA+SBST**: An *end-to-end* comparison on the effectiveness of incompatibility discovery
  - **CompCheck vs CompCheck--**: Measure the benefit of object reusing



[1] Ying Wang, Rongxin Wu, Chao Wang, Ming Wen, Yepang Liu, Shing-Chi Cheung, Hai Yu, Chang Xu, and Zhi-liang Zhu. Will Dependency Conflicts Affect My Program's Semantics? TSE 2021.

Evaluation Results: number of incompatibilities found through generated tests



1. CompCheck is effective in discovering incompatibility issues. It revealed 72.7% more issues than Sensor and 94.9% more issues than CIA+SBST

2. Object reusing significantly contributes to the overall effectiveness

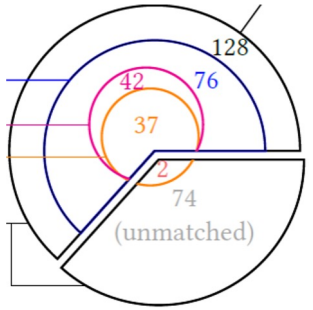
Contribution and Summary

yi_li@ntu.edu.sg
@liyistc

Problem Highlight

Many library upgrades have client-specific compatibility issues, which needs to be analyzed case by base

Evaluation



Revealed 72.7% more issues with incompatibility tests, than existing techniques

Tool: CompCheck

Source code and experiment data publicly available

https://sites.google.com/view/compcheck

Dataset: CompSuite

A newly compiled dataset for library behavioral incompatibilities

https://github.com/compsuite-team/compsuite