

On the reaction to deprecation of 25,357 clients of 4+1 popular Java APIs

Anand Ashok Sawant, Romain Robbes and Alberto Bacchelli

Deprecation is an attribute applied to a computer software feature, characteristic, or practice to indicate that it should be avoided (often because it is being superseded).

SMALLTALK-80

THE LANGUAGE AND ITS IMPLEMENTATION

Gold





github
SOCIAL CODING

Selected APIs



On the reaction to
deprecation
of **25,357** clients of **4+1**
popular Java APIs



Research Questions

The background of the slide is a dark gray gradient. It is decorated with numerous 3D question marks of varying sizes and orientations. Most of the question marks are black, but one question mark in the center-left area is a distinct blue color. The question marks are scattered across the entire slide, creating a thematic visual for the 'Research Questions' title.

RQ0: What API versions do clients use?

RQ1: How does API method deprecation affect clients?

RQ2: What is the scale of reaction in affected clients?

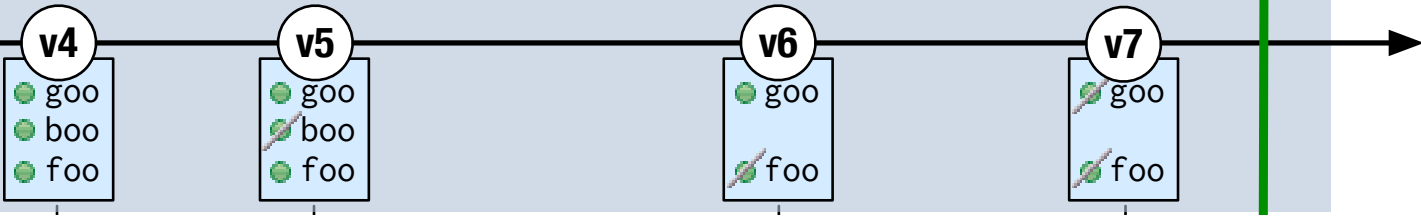
RQ3: What proportion of deprecations does affect clients?

RQ4: What is the time-frame of reaction in affected clients?

RQ5: Do affected clients react similarly?

Example API

- released versions,
- declared methods,
- ~~●~~ & deprecated methods



Results

RQ 0

RQ 1

RQ 2

RQ 3

RQ 4

RQ 5

Results

RQ 0

RQ 1

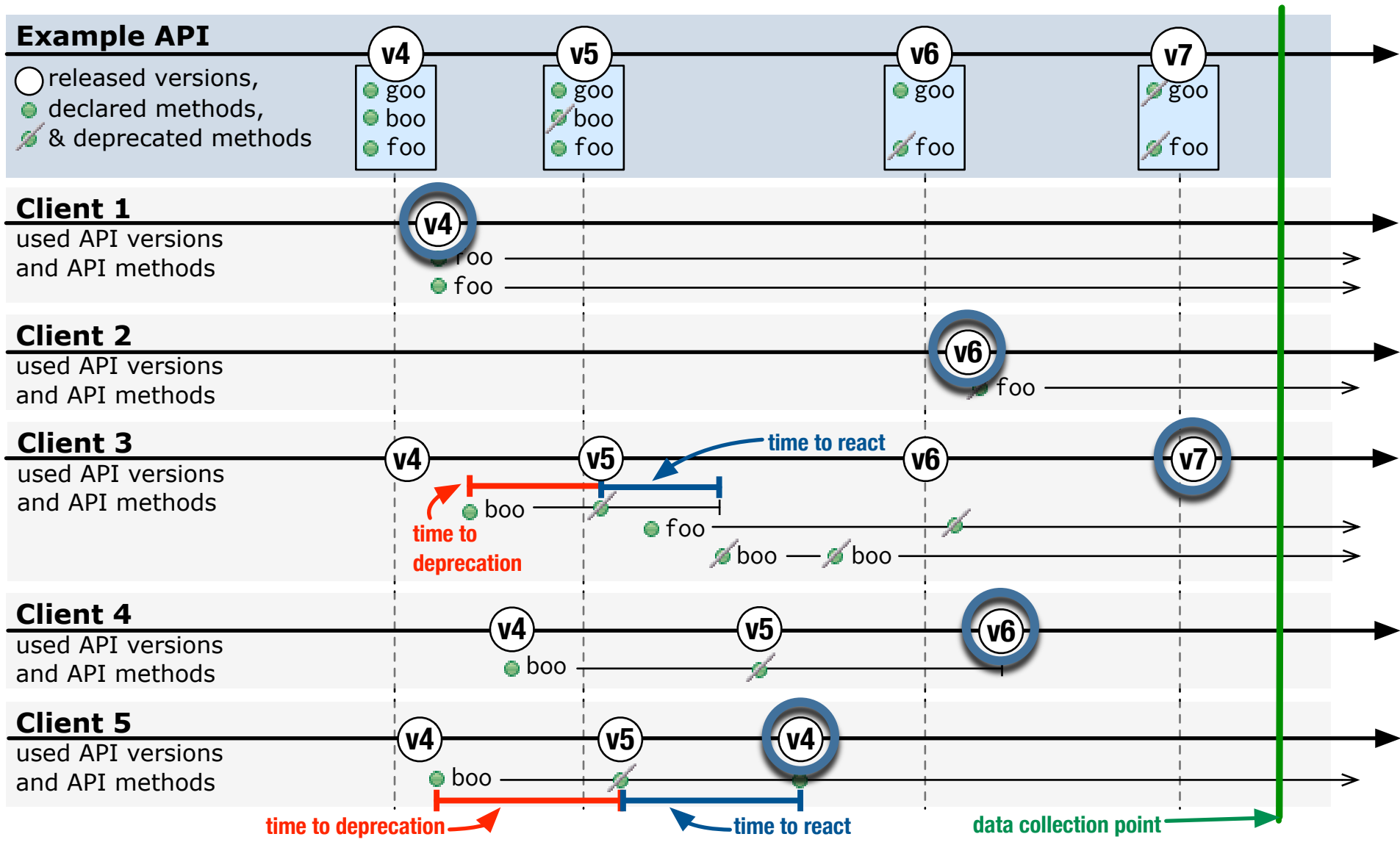
RQ 2

RQ 3

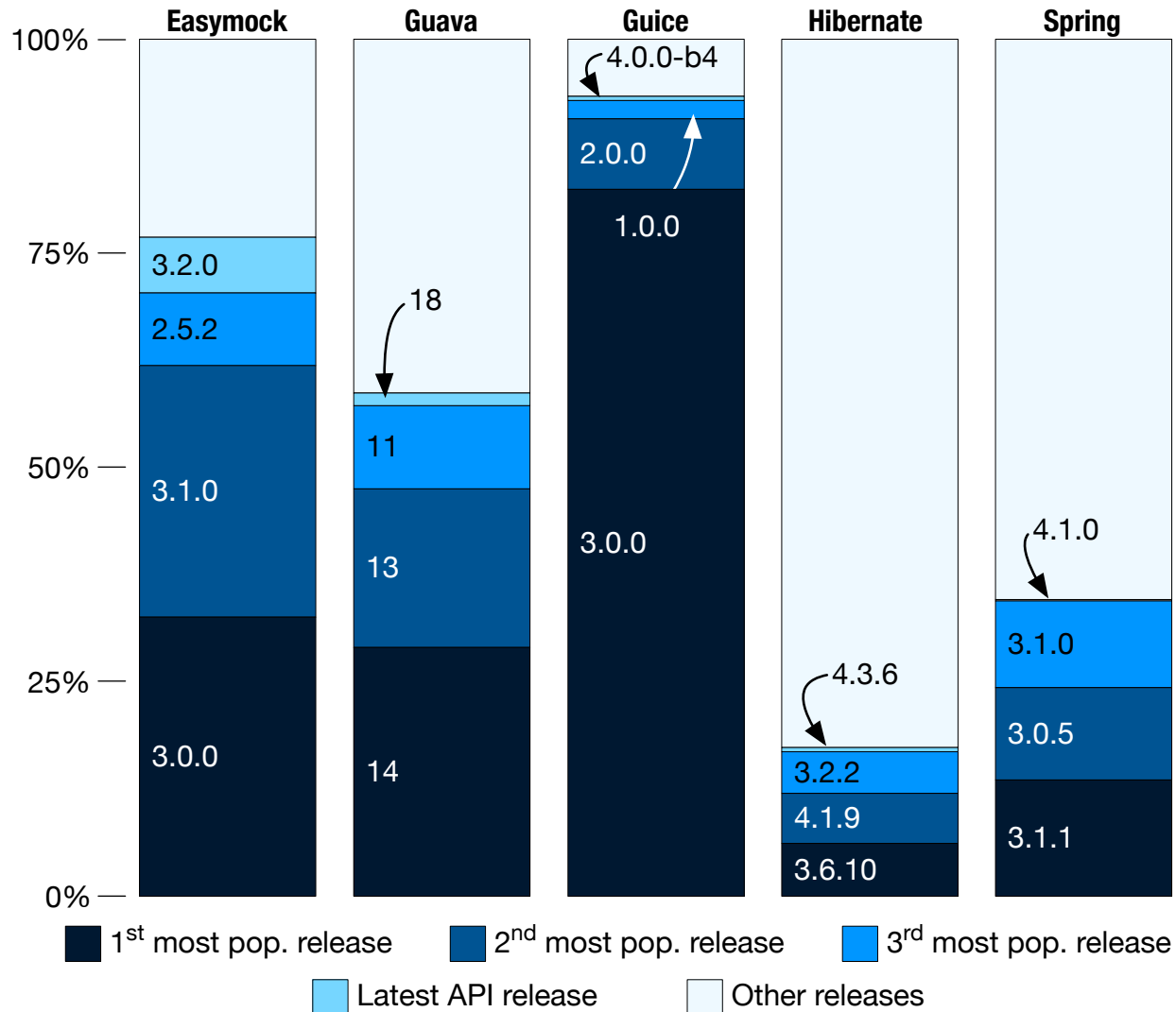
RQ 4

RQ 5

RQ0: What API versions do clients use?

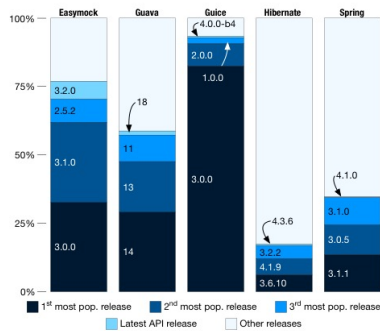


RQ0: What API versions do clients use?



Results

RQ0: What API versions do clients use?



RQ 1

RQ 2

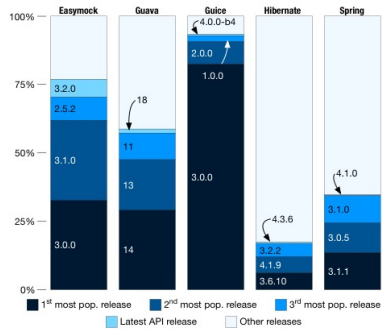
RQ 3

RQ 4

RQ 5

Results

RQ0: What API versions do clients use?



RQ 1

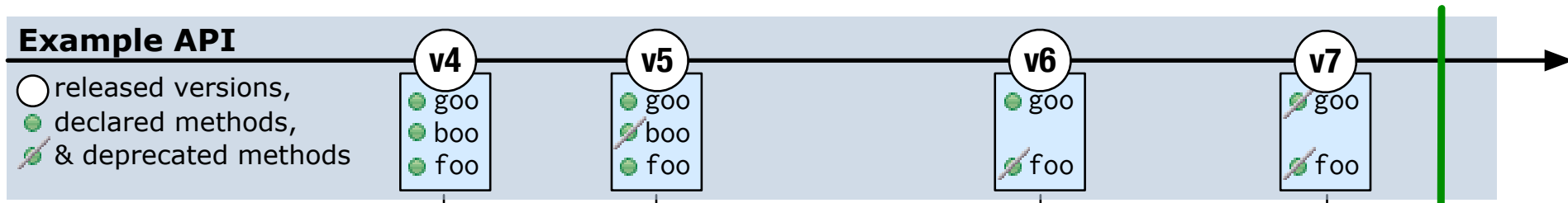
RQ 2

RQ 3

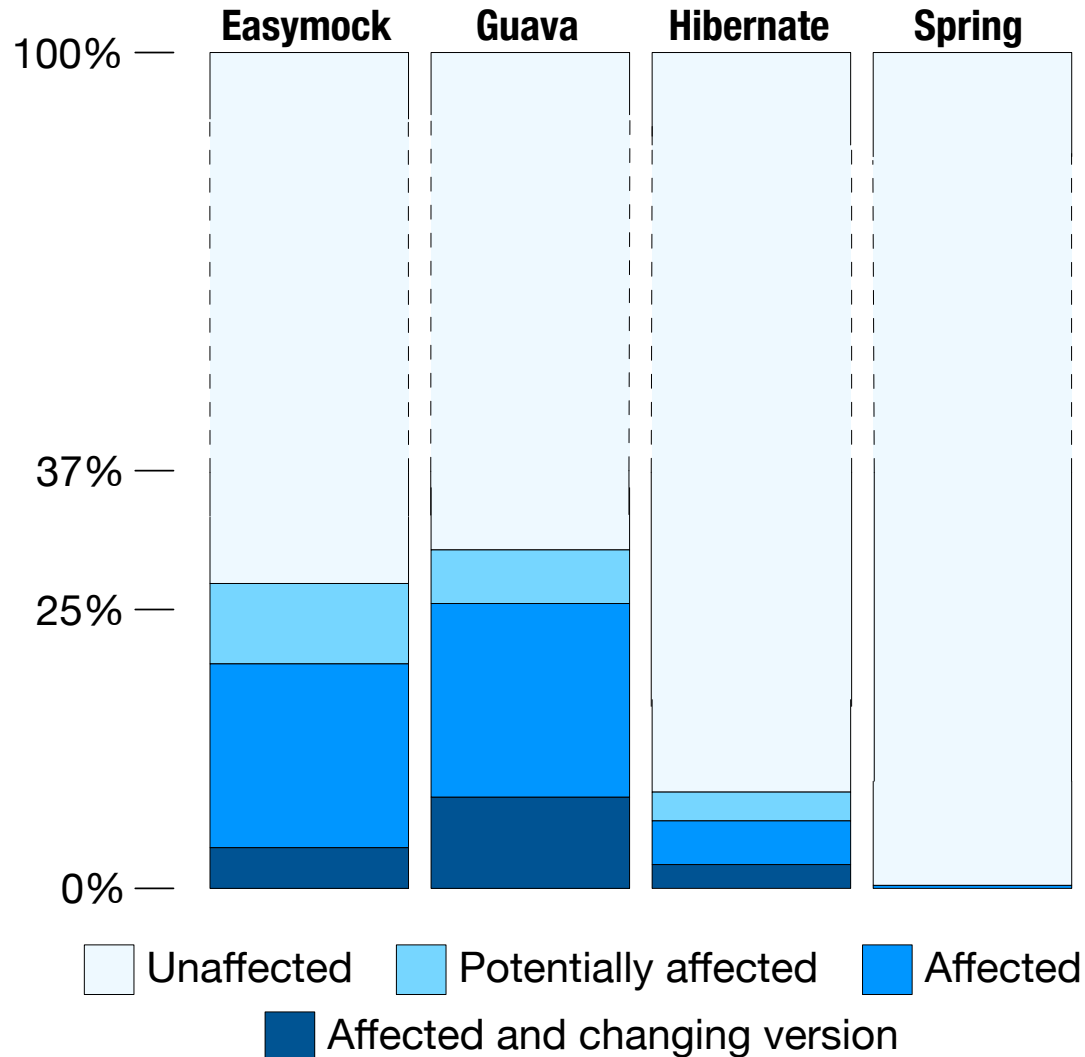
RQ 4

RQ 5

RQ1: How does API method deprecation affect clients?

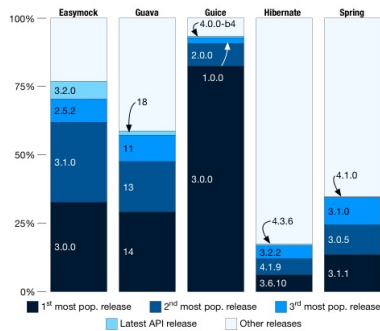


RQ1: How does API method deprecation affect clients?

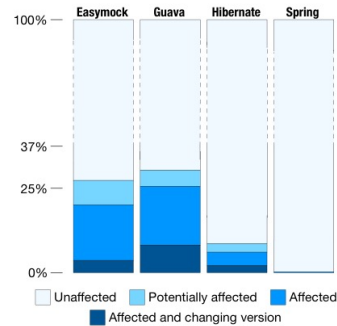


Results

RQ0: What API versions do clients use?



RQ1: How does API method deprecation affect clients?



RQ 2

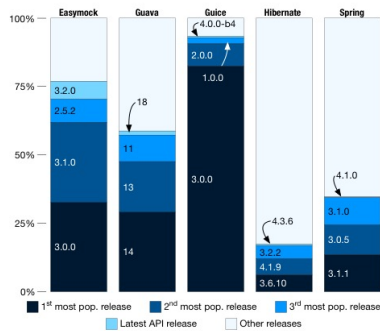
RQ 3

RQ 4

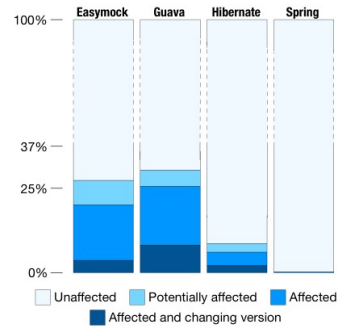
RQ 5

Results

RQ0: What API versions do clients use?



RQ1: How does API method deprecation affect clients?



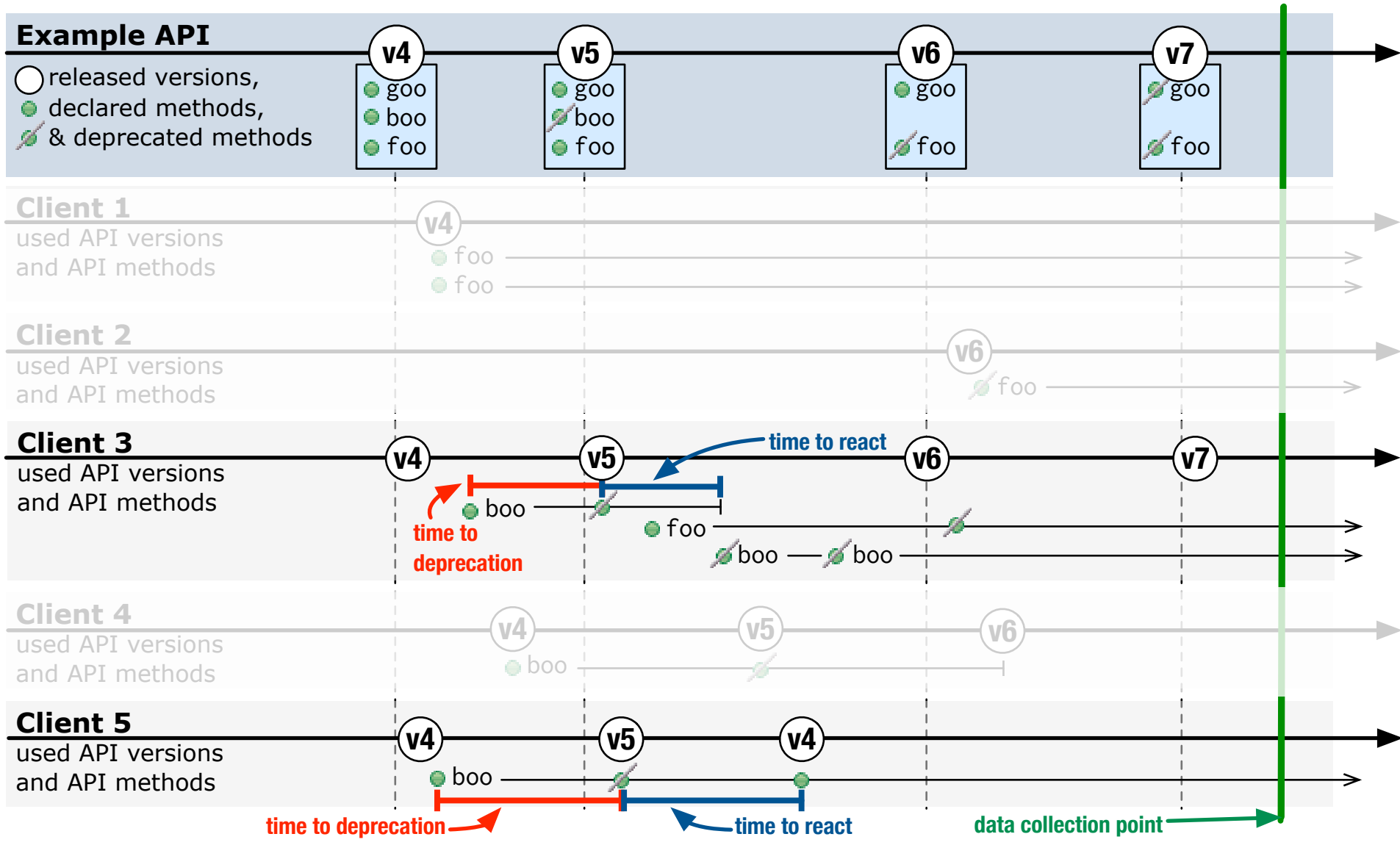
RQ 2

RQ 3

RQ 4

RQ 5

RQ2: What is the scale of reaction in affected clients?



RQ2: What is the scale of reaction in affected clients?

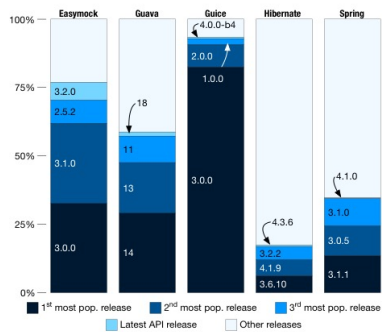
API	Affected clients that react	Median	Max
Easymock	17	11	109
Guava	161	3	283
Hibernate	40	5	59
Spring	10	31	131

RQ2: What is the scale of reaction in affected clients?

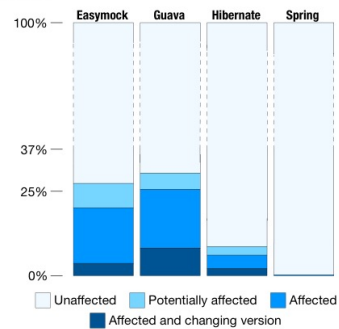
API	Potentially affected clients	Median	Max
Easymock	178	55	4,464
Guava	917	12	8,568
Hibernate	521	15	17,471
Spring	41	3	205

Results

RQ0: What API versions do clients use?



RQ1: How does API method deprecation affect clients?



RQ2: What is the scale of reaction in affected clients?

API	Affected clients that react	Median	Max
Easymock	17	11	109
Guava	161	3	283
Hibernate	40	5	59
Spring	10	31	131

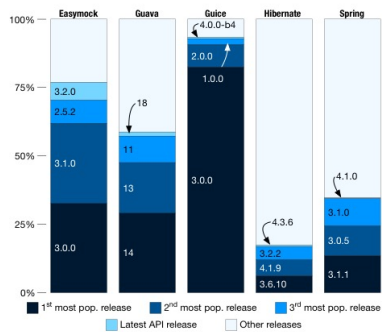
RQ 3

RQ 4

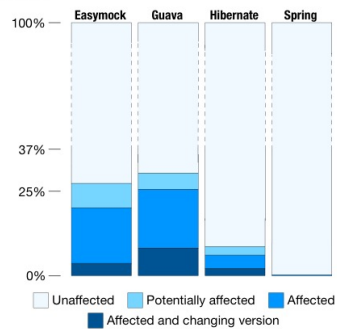
RQ 5

Results

RQ0: What API versions do clients use?



RQ1: How does API method deprecation affect clients?



RQ2: What is the scale of reaction in affected clients?

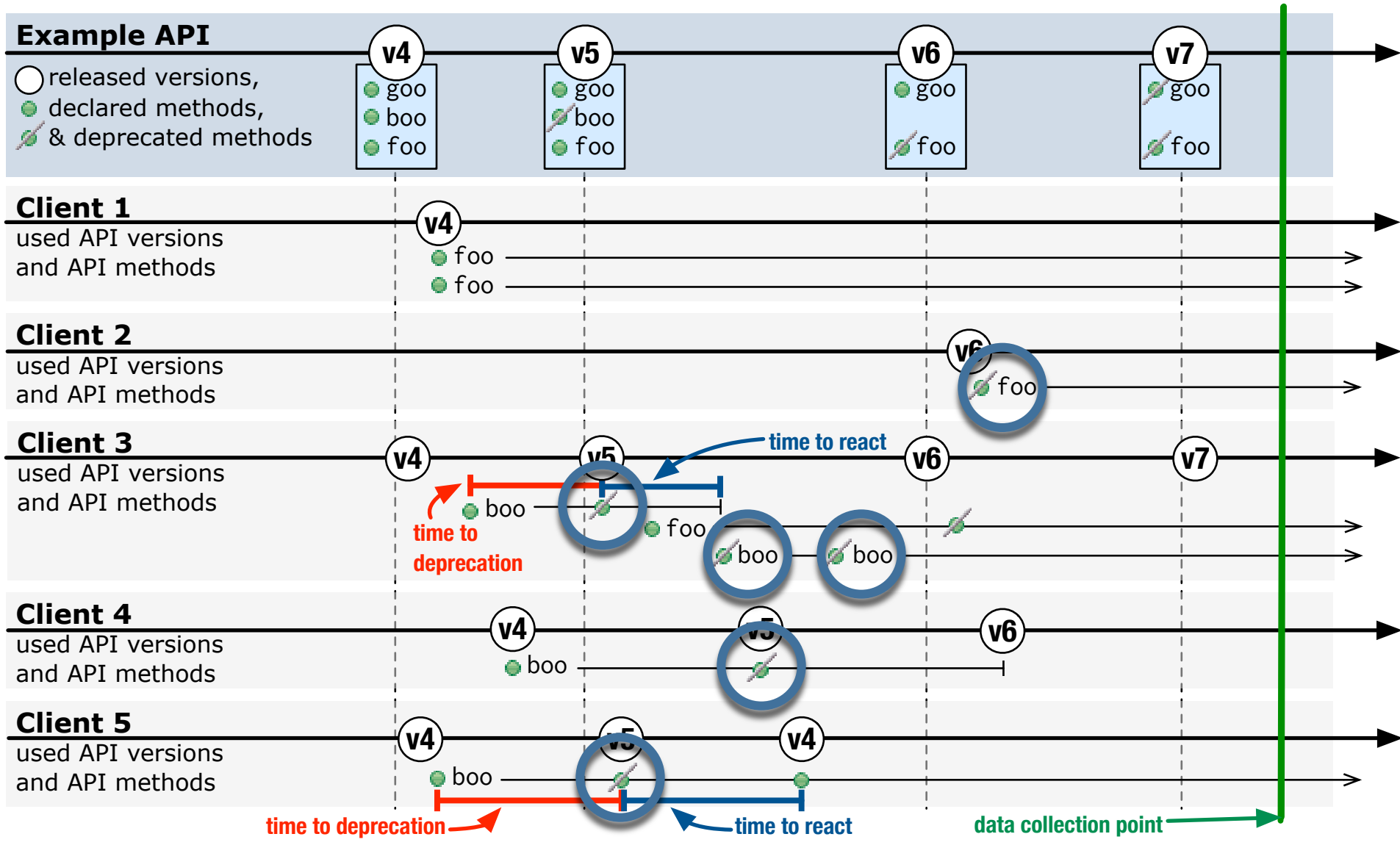
API	Affected clients that react	Median	Max
Easymock	17	11	109
Guava	161	3	283
Hibernate	40	5	59
Spring	10	31	131

RQ 3

RQ 4

RQ 5

RQ3: What proportion of deprecations does affect clients?



RQ3: What proportion of deprecations does affect clients?

EASYMOCK

13%

 **spring**

11%

 guava-libraries

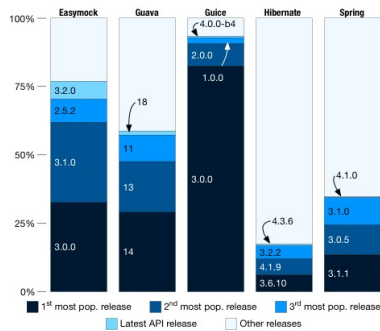
7%

 **HIBERNATE**

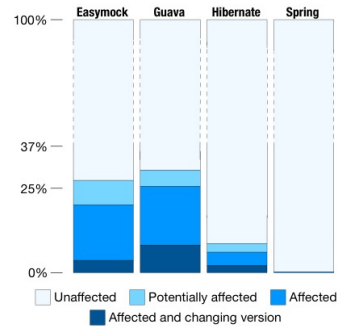
6%

Results

RQ0: What API versions do clients use?



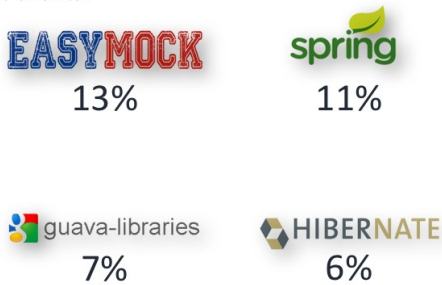
RQ1: How does API method deprecation affect clients?



RQ2: What is the scale of reaction in affected clients?

API	Affected clients that react	Median	Max
Easymock	17	11	109
Guava	161	3	283
Hibernate	40	5	59
Spring	10	31	131

RQ3: What proportion of deprecations does affect clients?

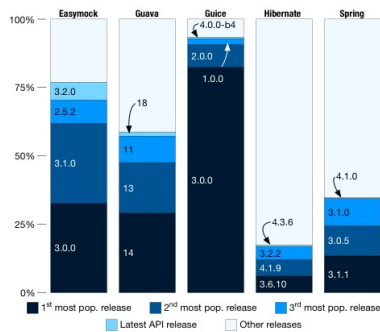


RQ 4

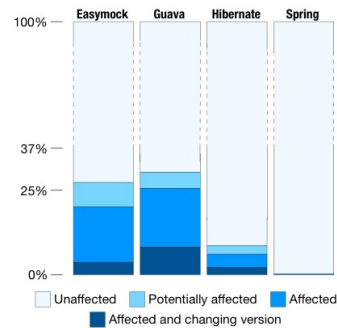
RQ 5

Results

RQ0: What API versions do clients use?



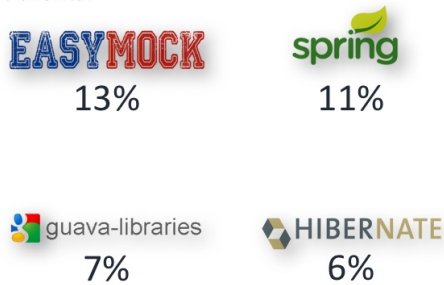
RQ1: How does API method deprecation affect clients?



RQ2: What is the scale of reaction in affected clients?

API	Affected clients that react	Median	Max
Easymock	17	11	109
Guava	161	3	283
Hibernate	40	5	59
Spring	10	31	131

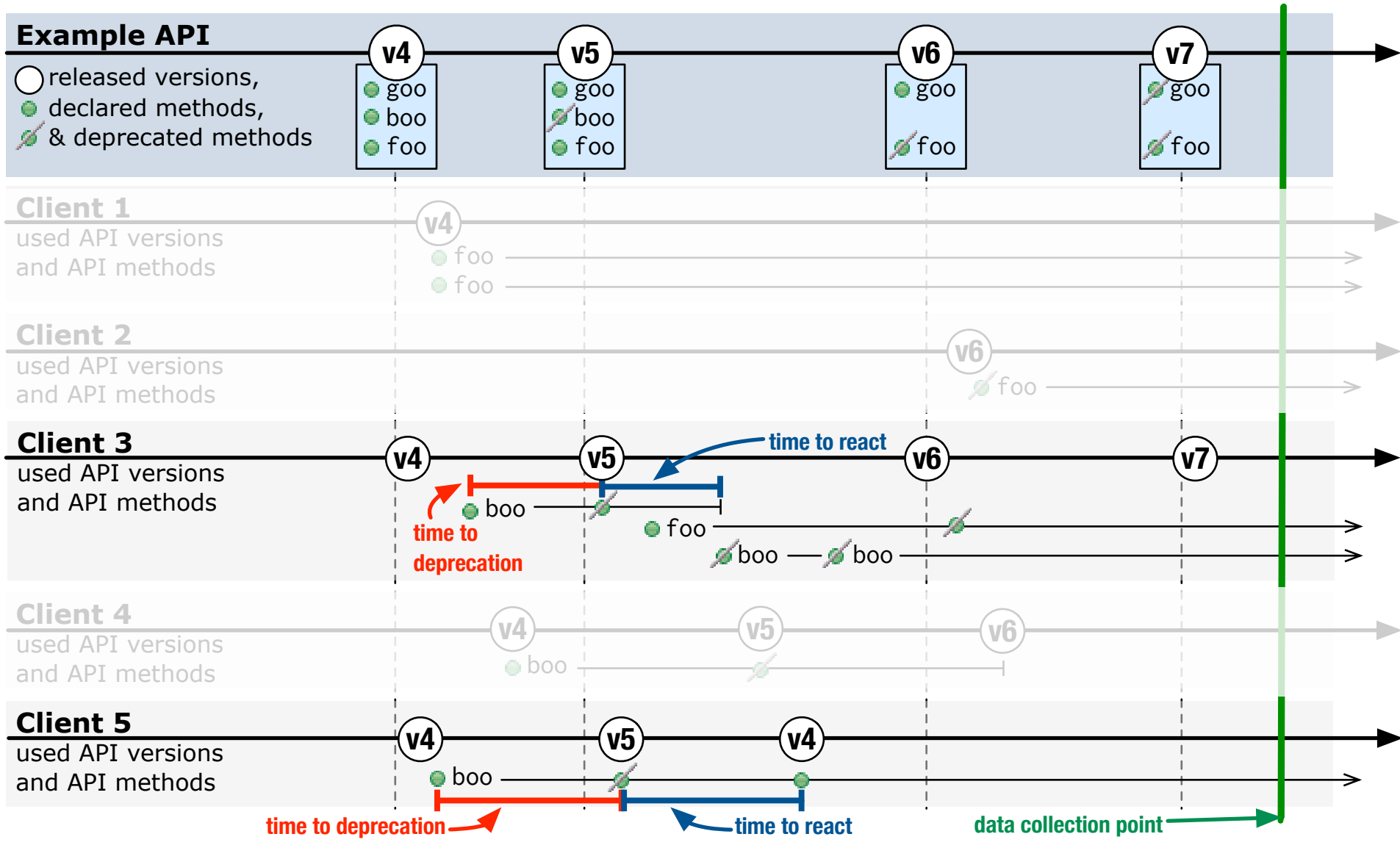
RQ3: What proportion of deprecations does affect clients?



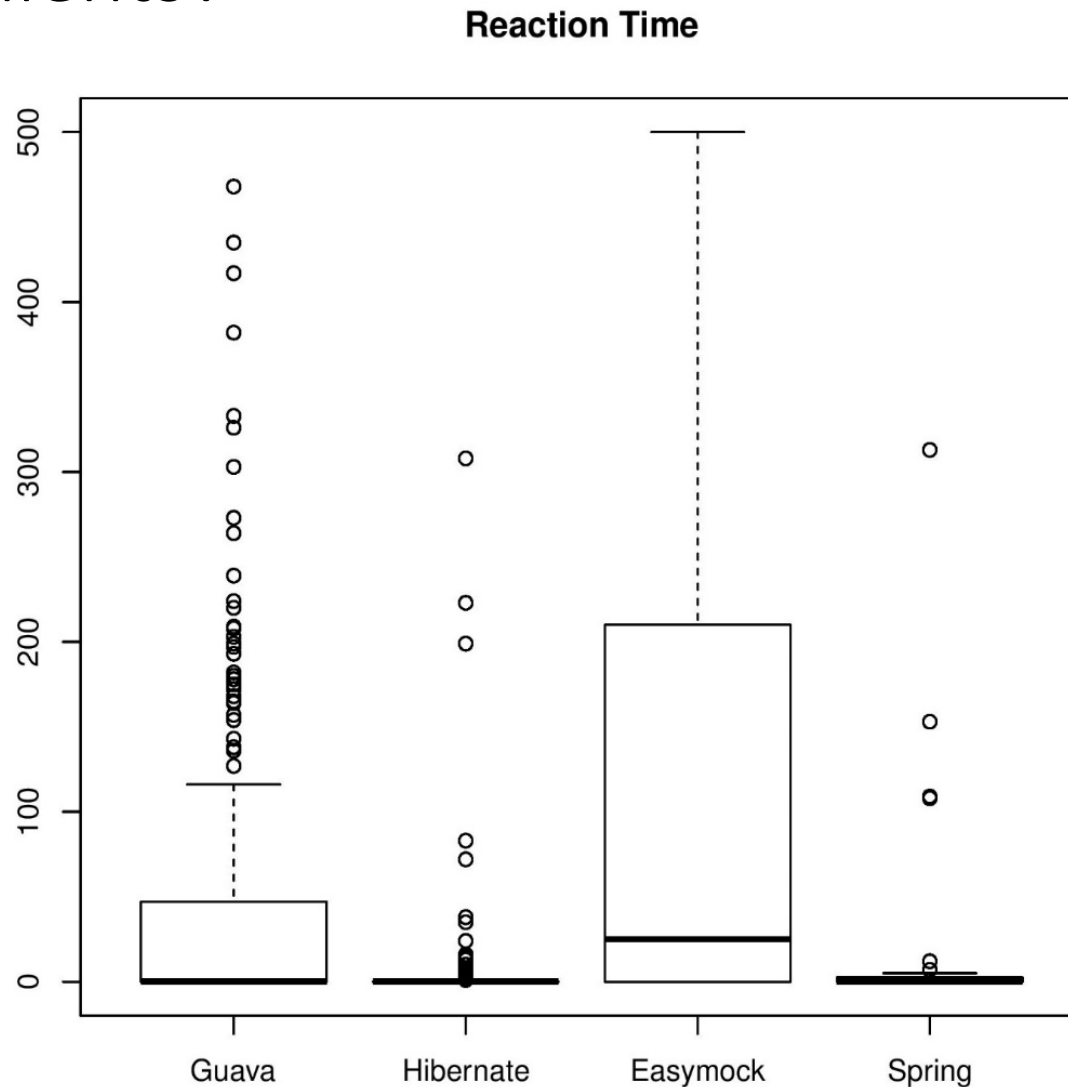
RQ 4

RQ 5

RQ4: What is the time-frame of reaction in affected clients?

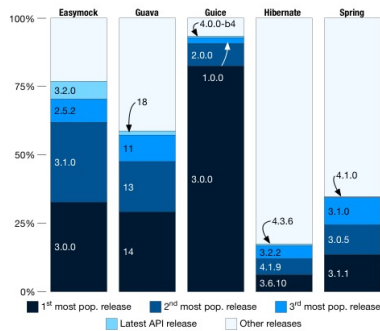


RQ4: What is the time-frame of reaction in affected clients?

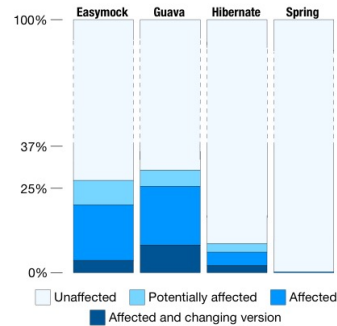


Results

RQ0: What API versions do clients use?



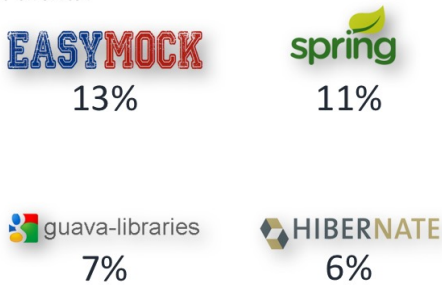
RQ1: How does API method deprecation affect clients?



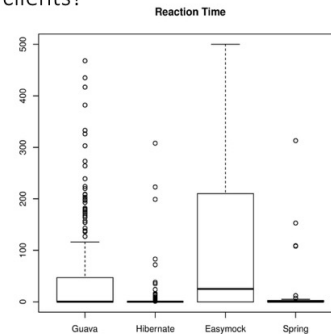
RQ2: What is the scale of reaction in affected clients?

API	Affected clients that react	Median	Max
Easymock	17	11	109
Guava	161	3	283
Hibernate	40	5	59
Spring	10	31	131

RQ3: What proportion of deprecations does affect clients?



RQ4: What is the time-frame of reaction in affected clients?



RQ 5

RQ5: Do all clients react similarly?

apply

```
@Deprecated  
public boolean apply(T input)
```

Deprecated. *Provided only to satisfy the Predicate interface; use `mightContain(T)` instead.*

Description copied from interface: Predicate

Returns the result of applying this predicate to input. This method is *generally expected*, but not absolutely required, to have the following properties:

- Its execution does not cause any observable side effects.
- The computation is *consistent with equals*; that is, `Objects.equal(a, b)` implies that `predicate.apply(a) == predicate.apply(b)`.

Specified by:

`apply` in interface `Predicate<T>`

getTargetType

```
@Deprecated  
public static Type getTargetType(MethodParameter methodParam)
```

Deprecated. *as of Spring 4.0, use `MethodParameter.getGenericParameterType()`*

Determine the target type for the given parameter specification.

Parameters:

`methodParam` - the method parameter specification

Returns:

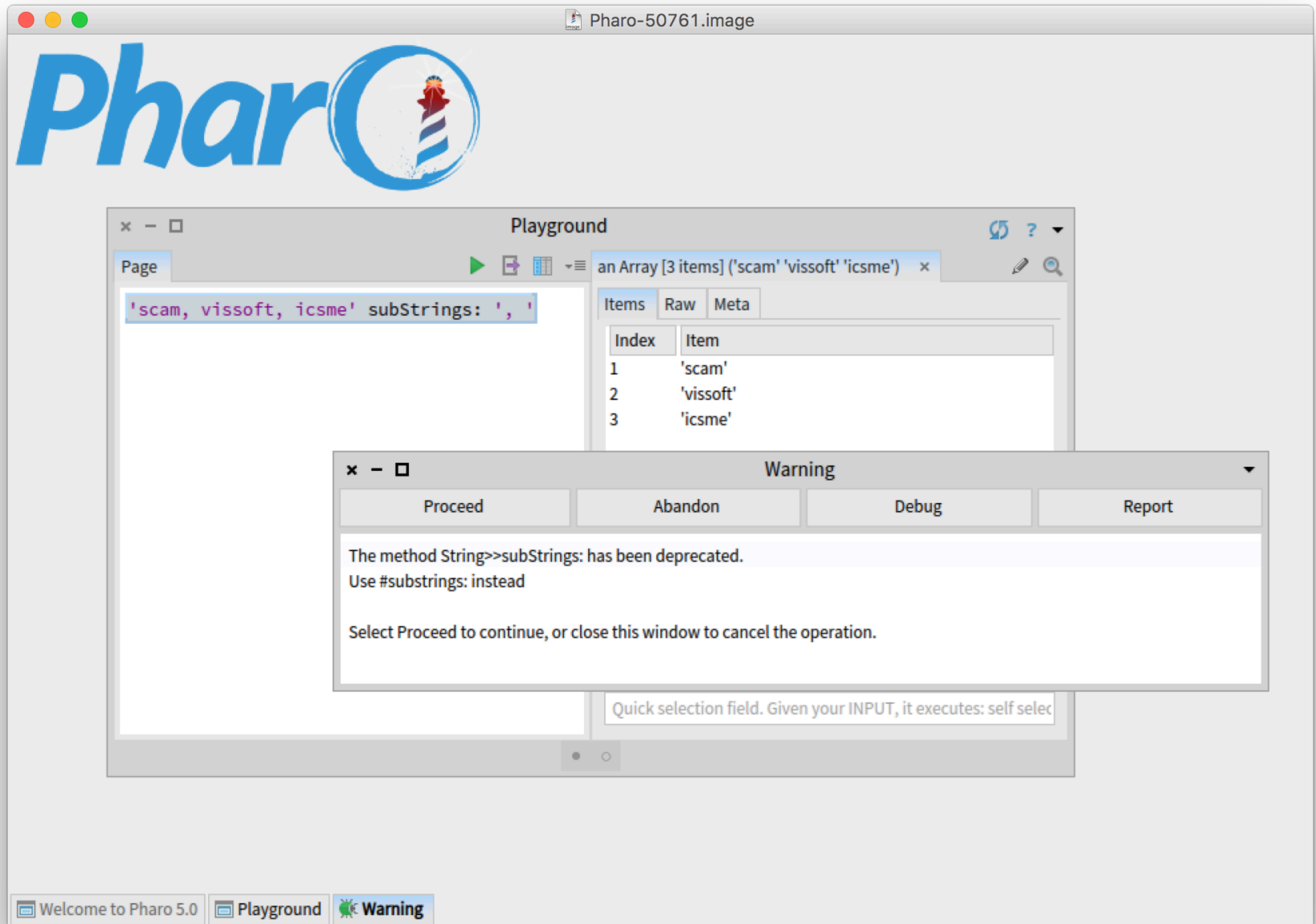
the corresponding generic parameter type

TAKE AWAY

Similar to Smalltalk



Deprecation in SmallTalk



Deprecation in Java

```
package org.serg;

import java.util.Date;

public class DateUser {
    public static void main(String args[]) {
        Date date = new Date();
        int year = date.getYear();
    }
}
```

Similar to Smalltalk

Not many API clients change their version of the API

Few clients react to deprecation

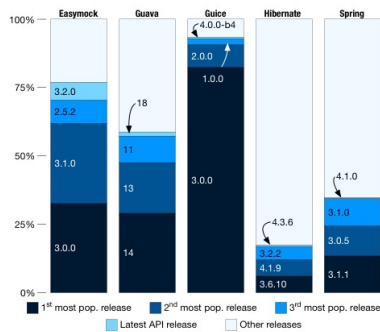
APIs document deprecated changes well

Clients tend to delete references to deprecated entities and not replace them

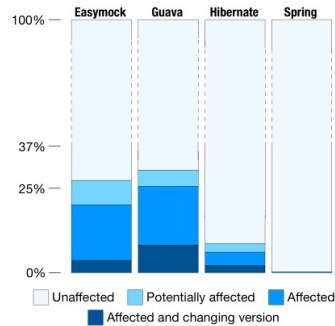
Smalltalk clients actually replace deprecated calls

Results

RQ0: What API versions do clients use?



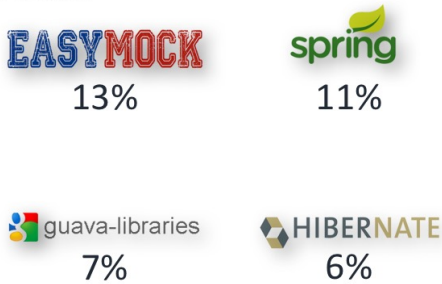
RQ1: How does API method deprecation affect clients?



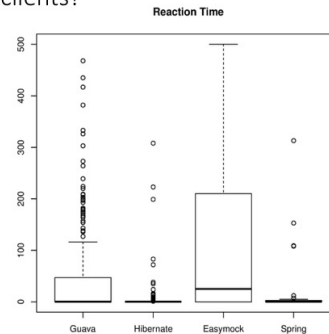
RQ2: What is the scale of reaction in affected clients?

API	Affected clients that react	Median	Max
Easymock	17	11	109
Guava	161	3	283
Hibernate	40	5	59
Spring	10	31	131

RQ3: What proportion of deprecations does affect clients?



RQ4: What is the time-frame of reaction in affected clients?



RQ5: Do all clients react similarly?

```

@Deprecated
public boolean apply(T input)
    @Deprecated. Provided only to satisfy the Predicate interface; use rightContains(T) instead.
    Description copied from interface: Predicate
    Returns the result of applying this predicate to input. This method is generally expected, but not absolutely required, to have the following properties:
    • Its execution does not cause any observable side effects
    • The computation is considered with equals: that is, Objects.equals(a, b) implies that predicate.apply(a) == predicate.apply(b).
    Specified by:
    apply in interface Predicate<T>

getTargetType
    @Deprecated
    public static Type getTargetType(MethodParameter methodParam)
    @Deprecated. as of Spring 4.0, use MethodParameter.getGenericParameterType()
    Determine the target type for the given parameter specification.
    Parameters:
    methodParam - the method parameter specification
    Returns:
    the corresponding generic parameter type
    
```