

To Mock Or Not To Mock?

An Empirical Study on Mocking Practices

Davide Spadini, Mauricio Aniche,
Mael Bruntink, Alberto Bacchelli



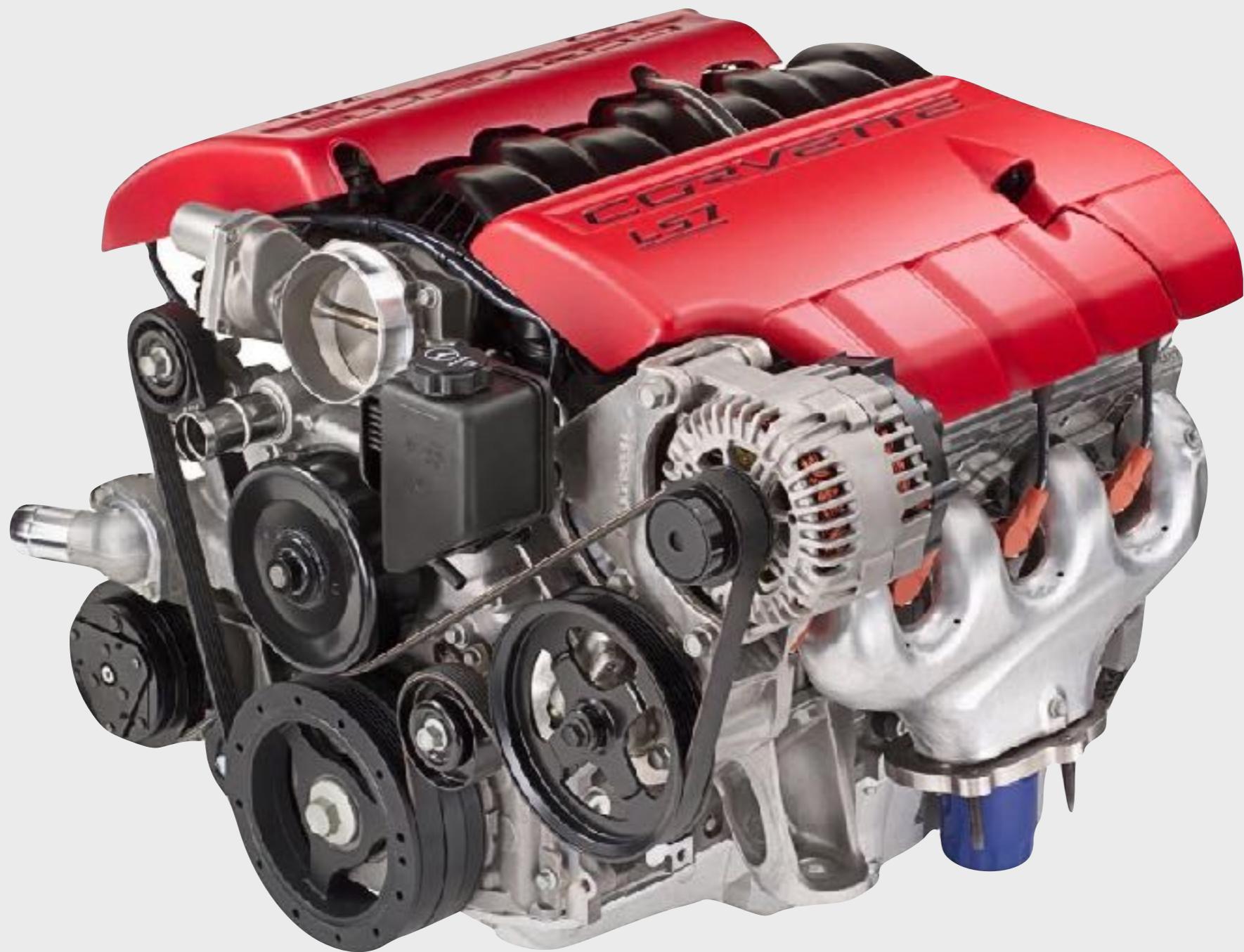
@DavideSpadini



European Commission



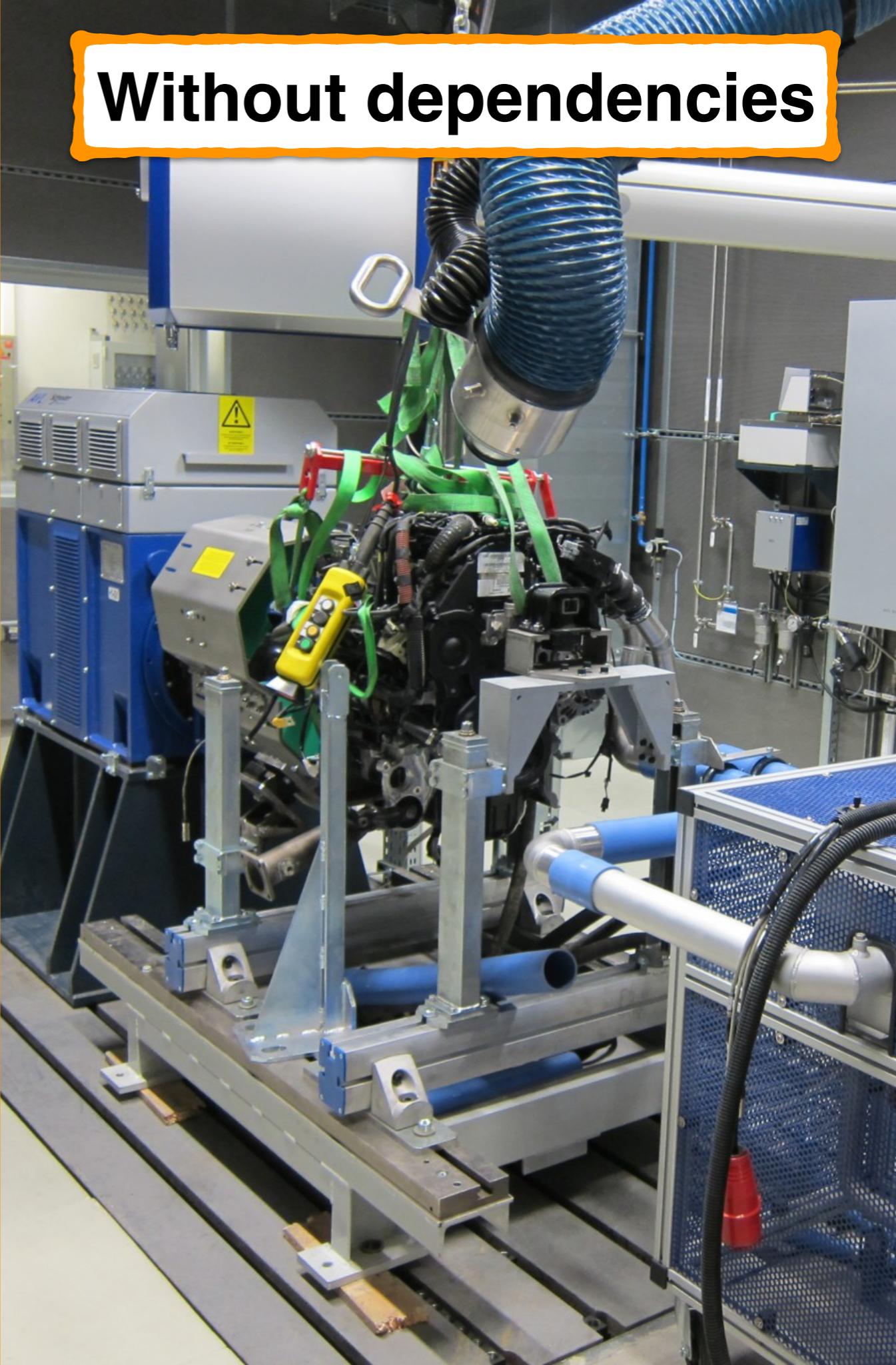
How do I test this?



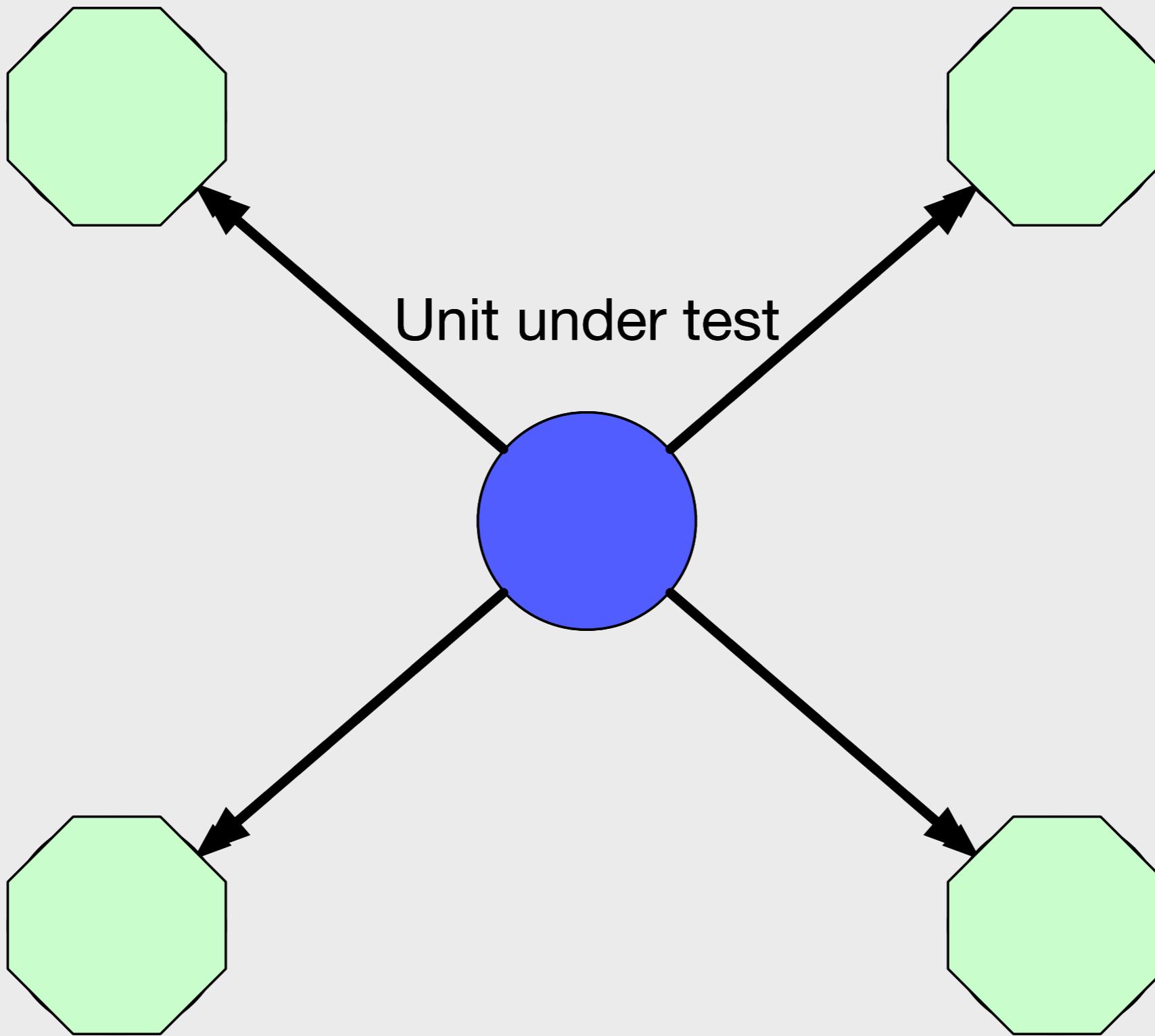
With dependencies



Without dependencies



Mock



```
1 // 1: Mocking LinkedList  
2 LinkedList mockObj = mock(LinkedList.class);  
3  
4 // 2: Instructing the mock object behaviour  
5 when(mockObj.get(0)).thenReturn("first");  
6 when(mockObj.get(1))  
7     .thenThrow(new RuntimeException());  
8  
9 // 3: Invoking methods in the mock  
10 System.out.println(mockObj.get(0));  
11 System.out.println(mockObj.get(1));
```

With dependencies



With dependencies



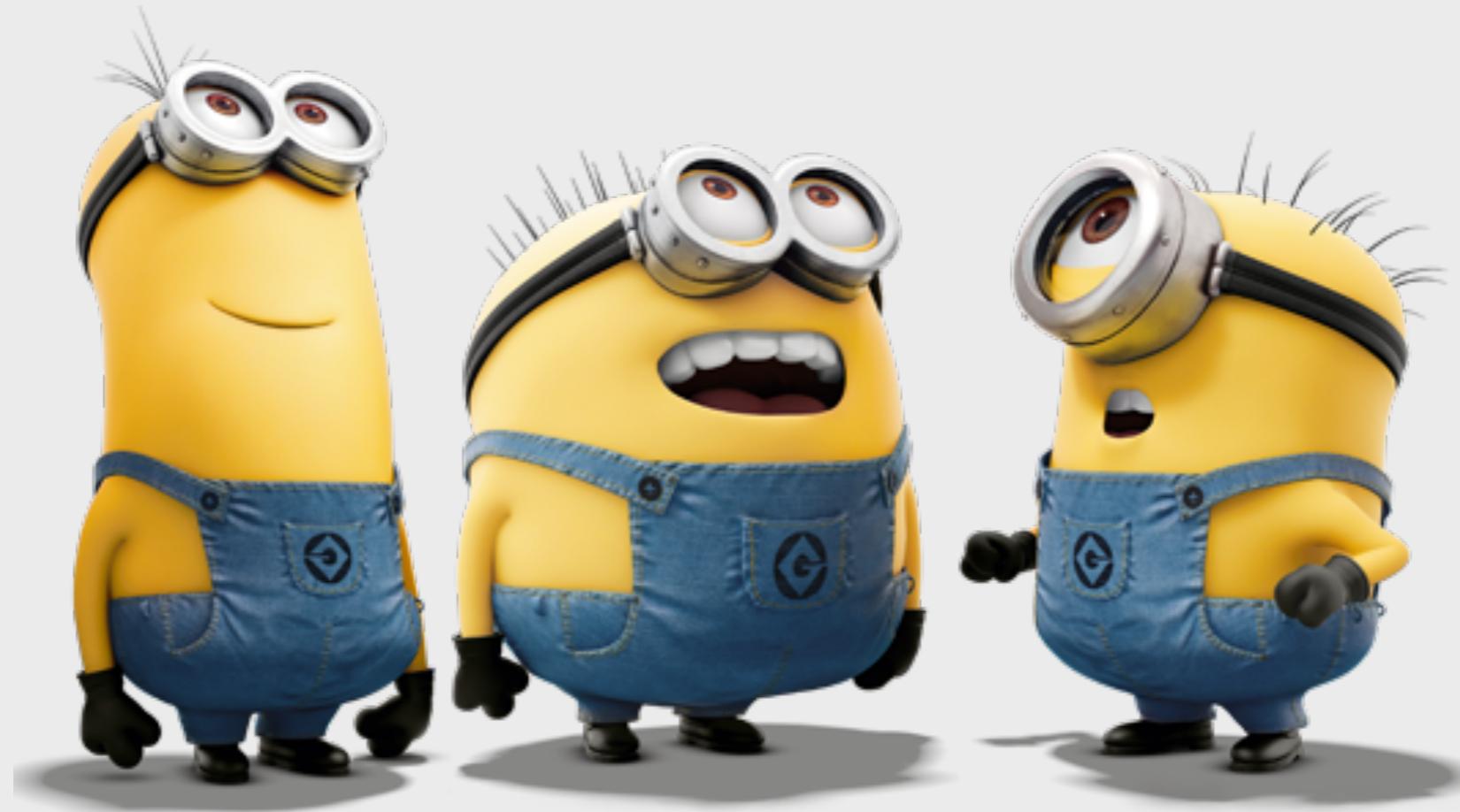
With dependencies

The image is a collage of mathematical concepts and formulas. It includes:

- A graph of a function $y = \sin x$ with its derivative $y' = \cos x$.
- Trigonometric identities: $\sin^2 x + \cos^2 x = 1$, $\sin 2x = 2 \sin x \cos x$, $\cos 2x = \cos^2 x - \sin^2 x$.
- Pythagorean theorem: $a^2 + b^2 = c^2$.
- Area formulas: $\text{Area} = \frac{1}{2}ab \sin C$, $\text{Area} = \frac{1}{2}r^2 \theta$.
- Probability: $P(A|B) = \frac{P(A \cap B)}{P(B)}$.
- Calculus: $\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n f(x_i)$, $\int_a^b f(x) dx$.
- Algebra: $x^2 + 3xR + R^2 = 3x^2$.
- Geometry: A pyramid with base side length a and height b .
- Physics: $E = mc^2$.
- Trigonometric ratios: $\sin \theta = \frac{\text{opposite}}{\text{hypotenuse}}$, $\cos \theta = \frac{\text{adjacent}}{\text{hypotenuse}}$, $\tan \theta = \frac{\sin \theta}{\cos \theta}$.
- Series: $\sum n^k$.
- Logarithms: $\log p(A) = \sum p(\omega) \log \sin 2x$.
- Equations: $a^3 + b^3 = (a+b)(a^2 - ab + b^2)$, $(a+b)^2 = a^2 + 2ab + b^2$.
- Trigonometric identities: $\sin(\pi - x) = \sin x$, $\cos(\pi - x) = -\cos x$.
- Integration: $F(t) = \int f(t) \cdot e^{-it} dt$.

Without dependencies

- Faster
- Easier

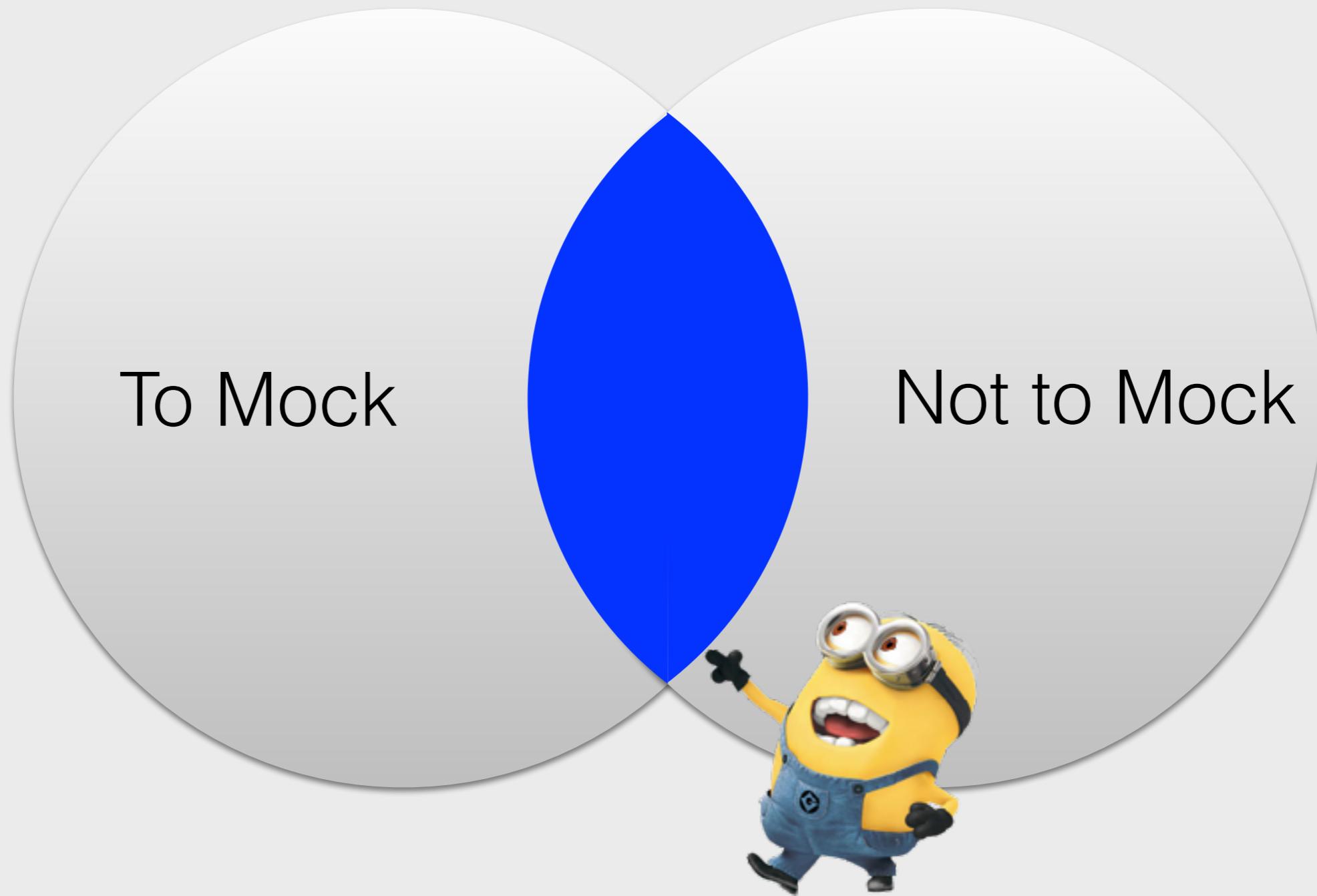


Without dependencies

You lose in reality, after all, it is just a simulation!



So, what should we do?



Research Questions

RQ1: **What** test dependencies do developers mock?

RQ2: **Why** do developers decide to (not) mock specific dependencies?

RQ3: Which are the main **challenges** experienced with testing using mocks?

Methodology



Opensource

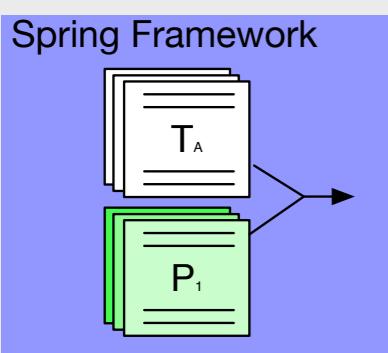
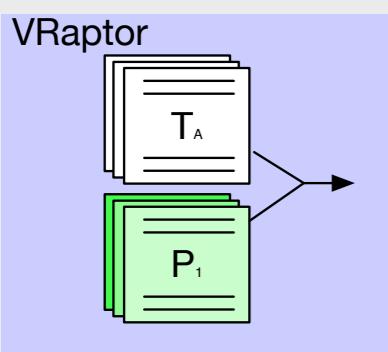
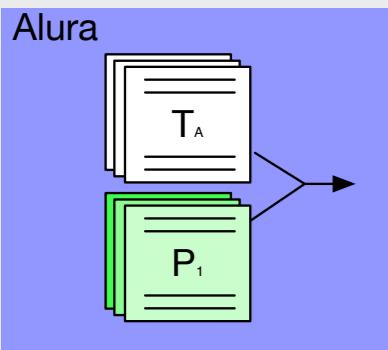
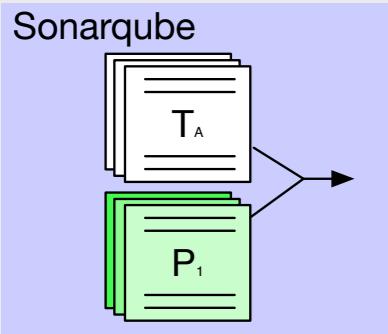
vraptor sonarqube



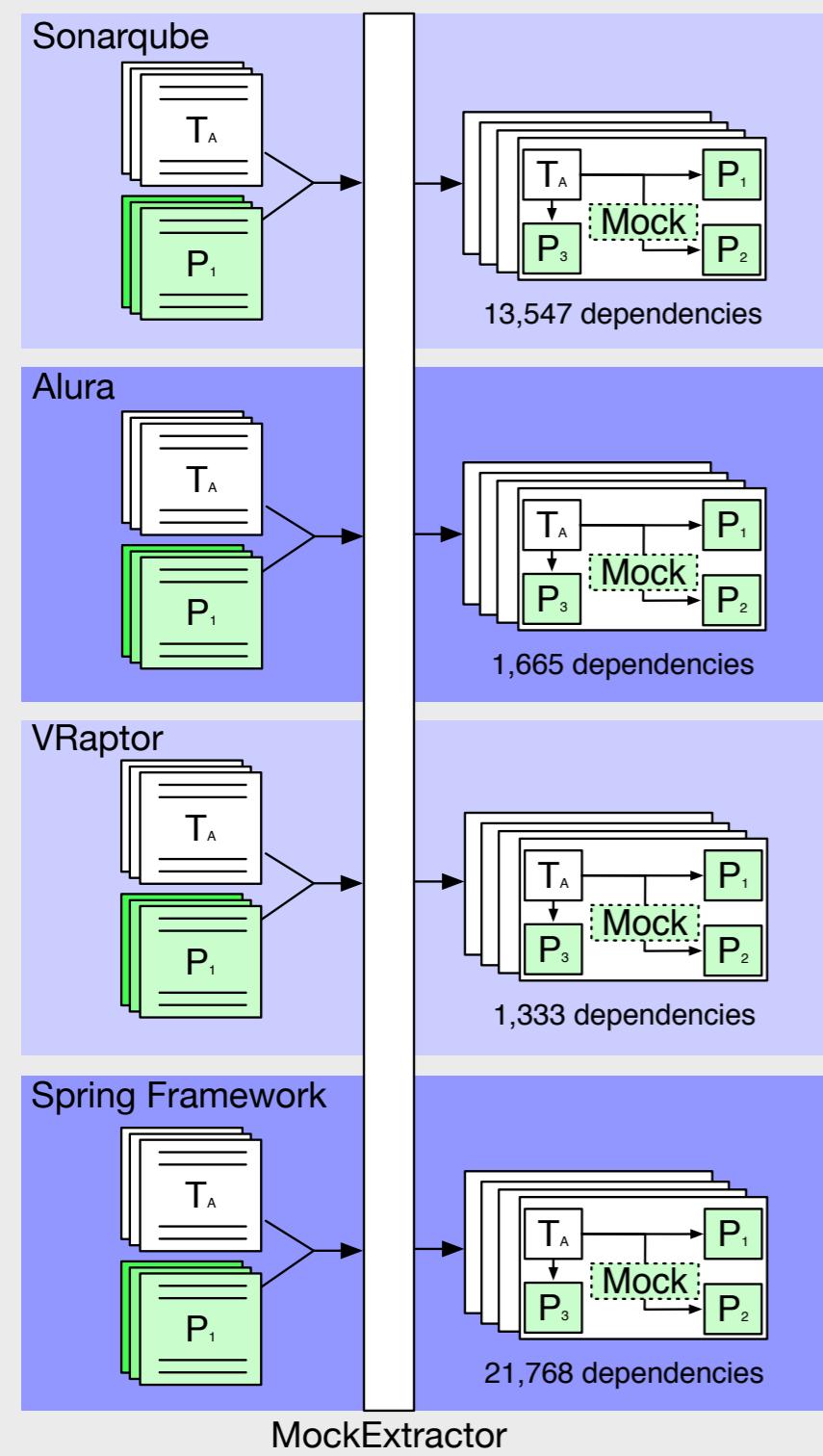
Industrial

alura

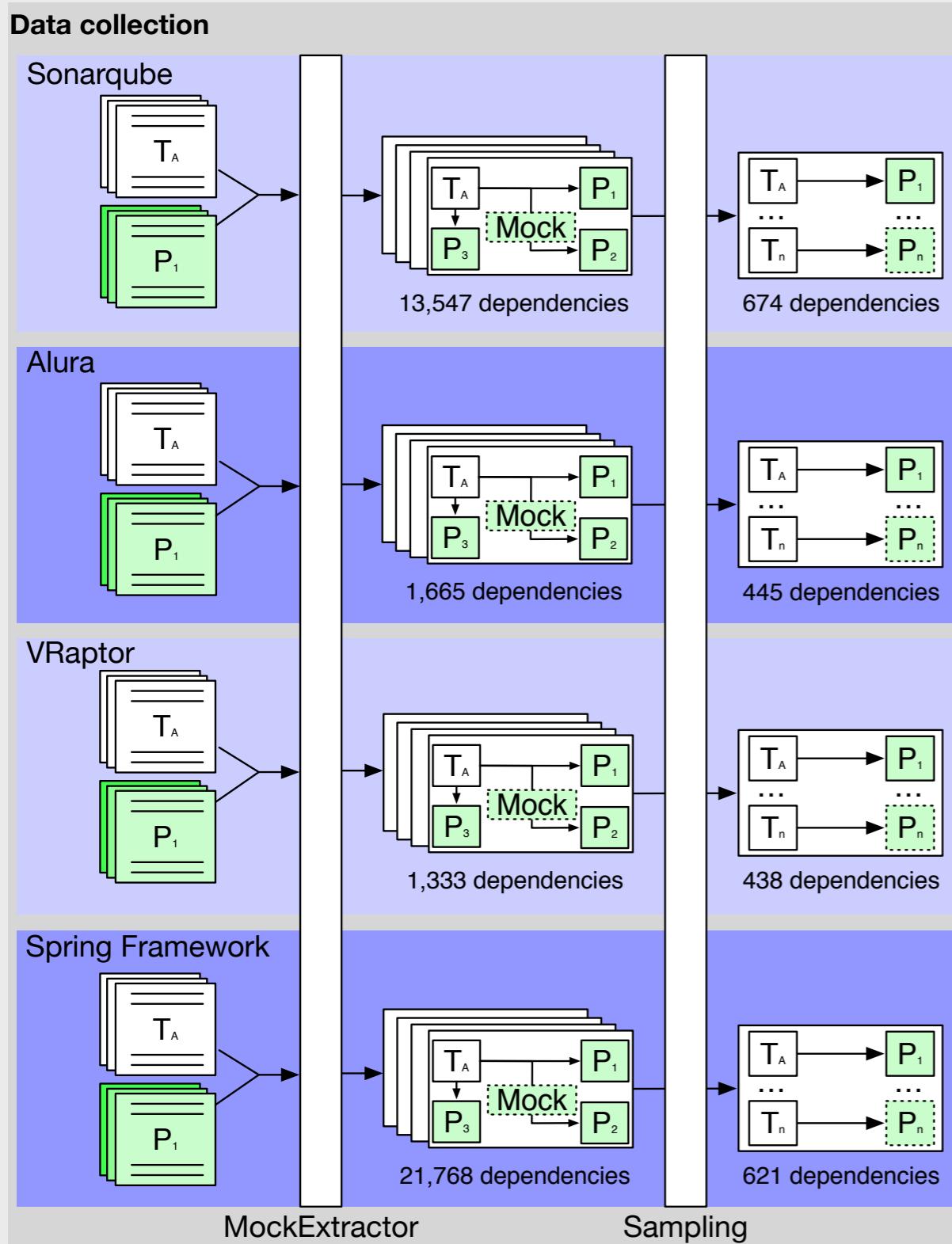
Methodology



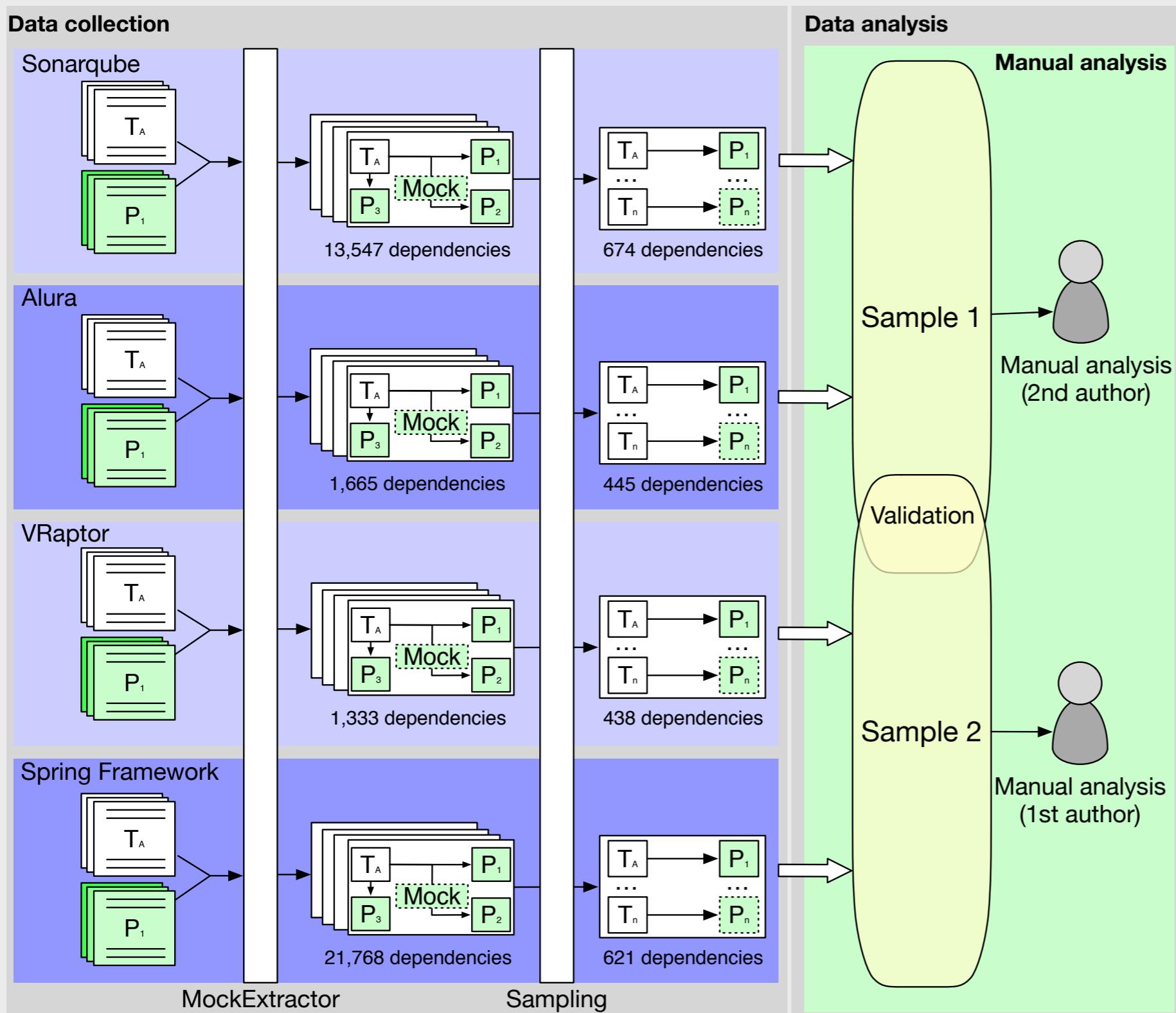
Methodology



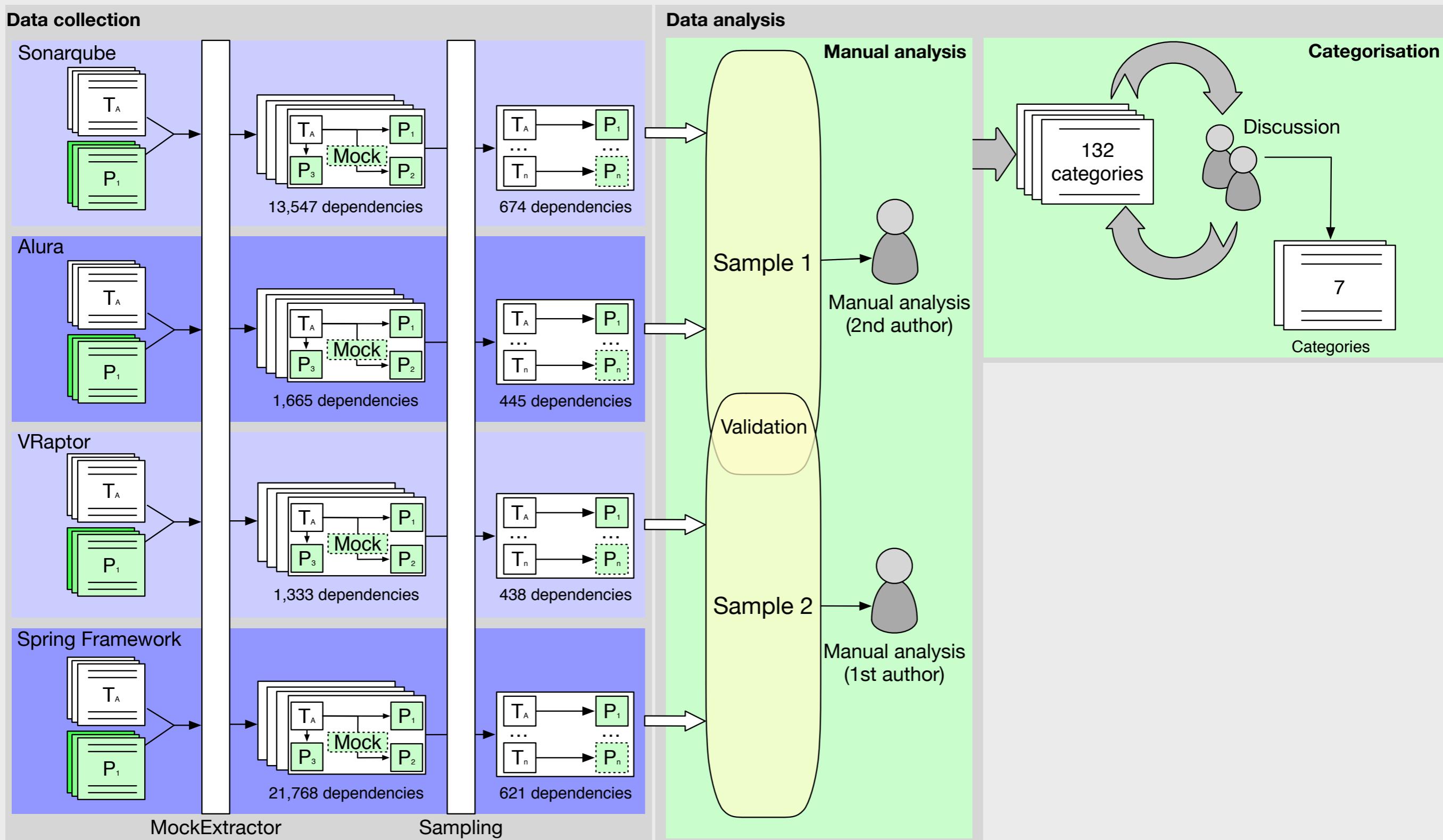
Methodology



Methodology



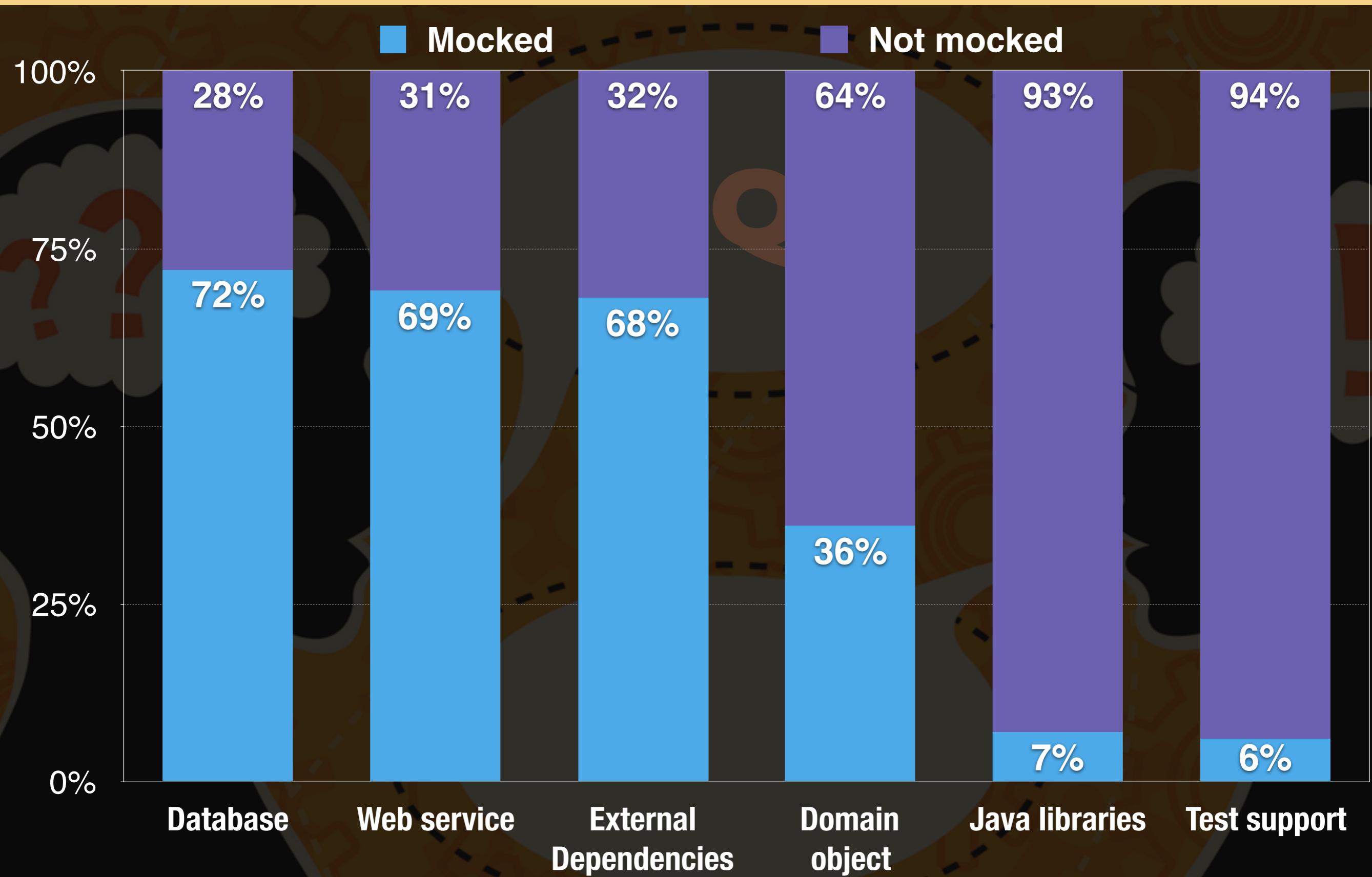
Methodology



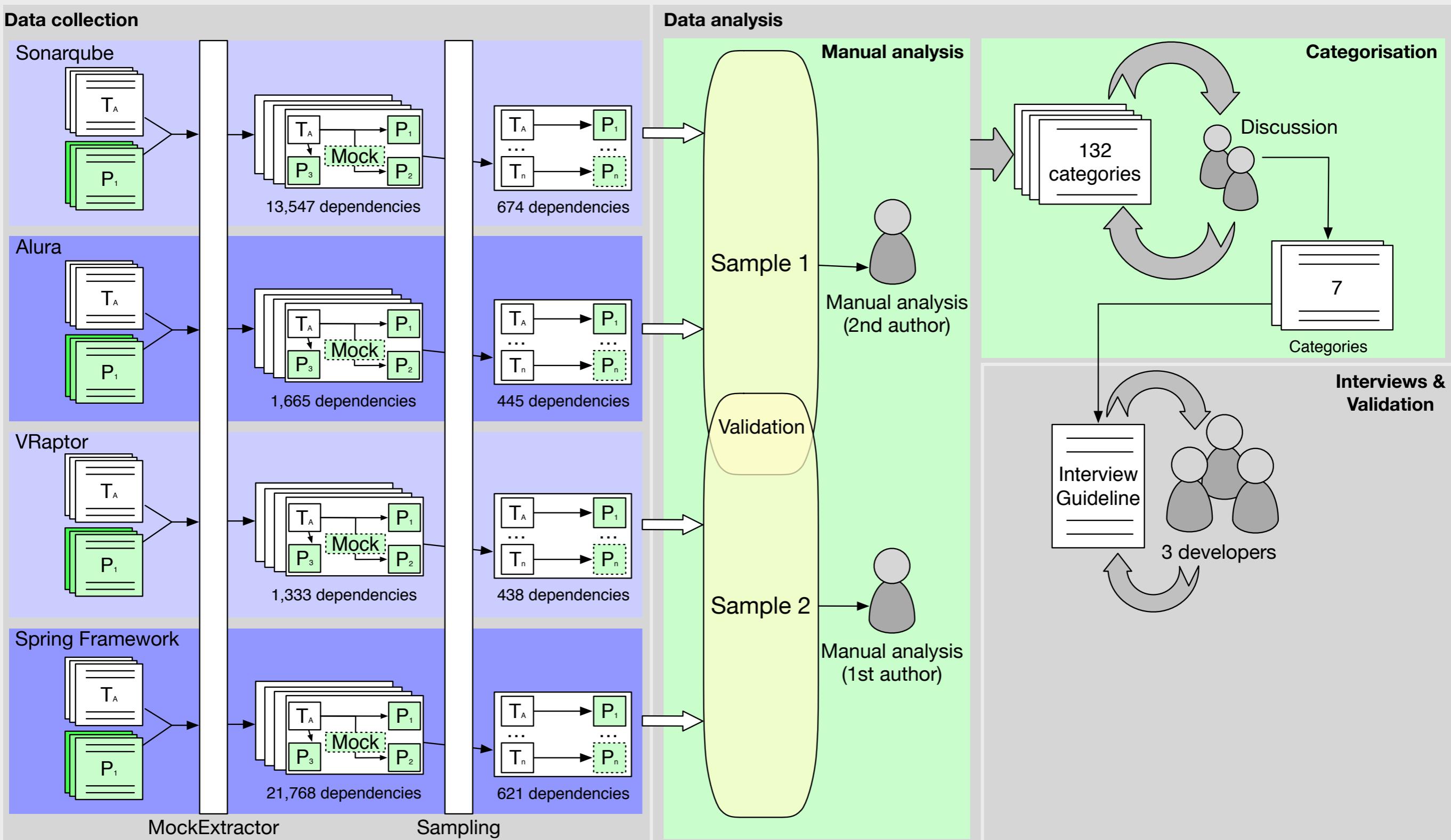
RQ1: **What** test dependencies do developers mock?



RQ1: What test dependencies do developers mock?



Methodology



RQ2: **Why** do developers decide to (not) mock a dependency?



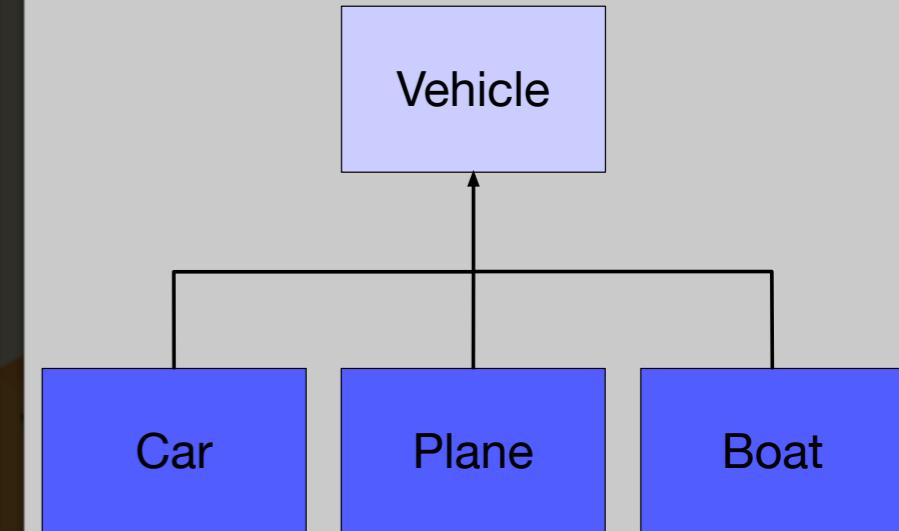
RQ2: Why do developers decide to (not) mock a dependency?



Concrete implementation is not easy

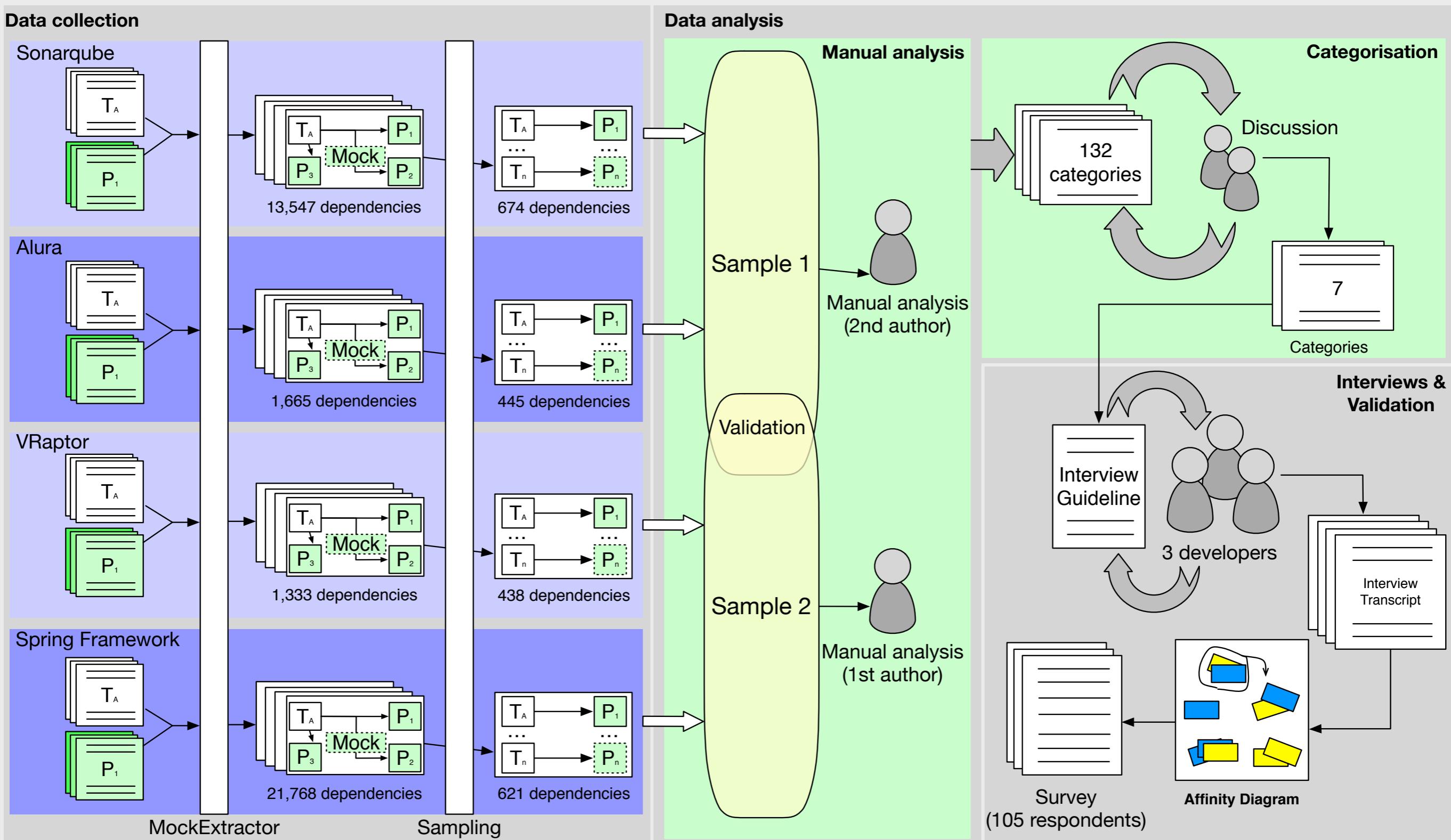


Not when the focus of the test is the integration



Mock interfaces rather than a specific implementation

Methodology



RQ3: Which are the main **challenges** experienced with testing using mocks?



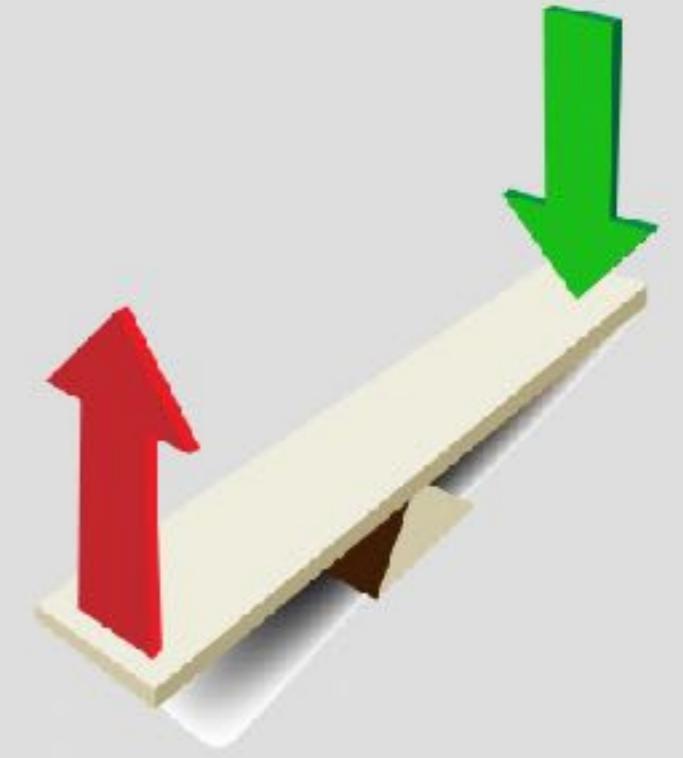
RQ3: Which are the main **challenges** experienced with testing using mocks?



Dealing with
coupling



Legacy systems



Production quality
code vs
mocks

Discussion



Mockito developer

- He agreed on all the findings
- How Mockito can help?
- What Mockito is supposed to do more?

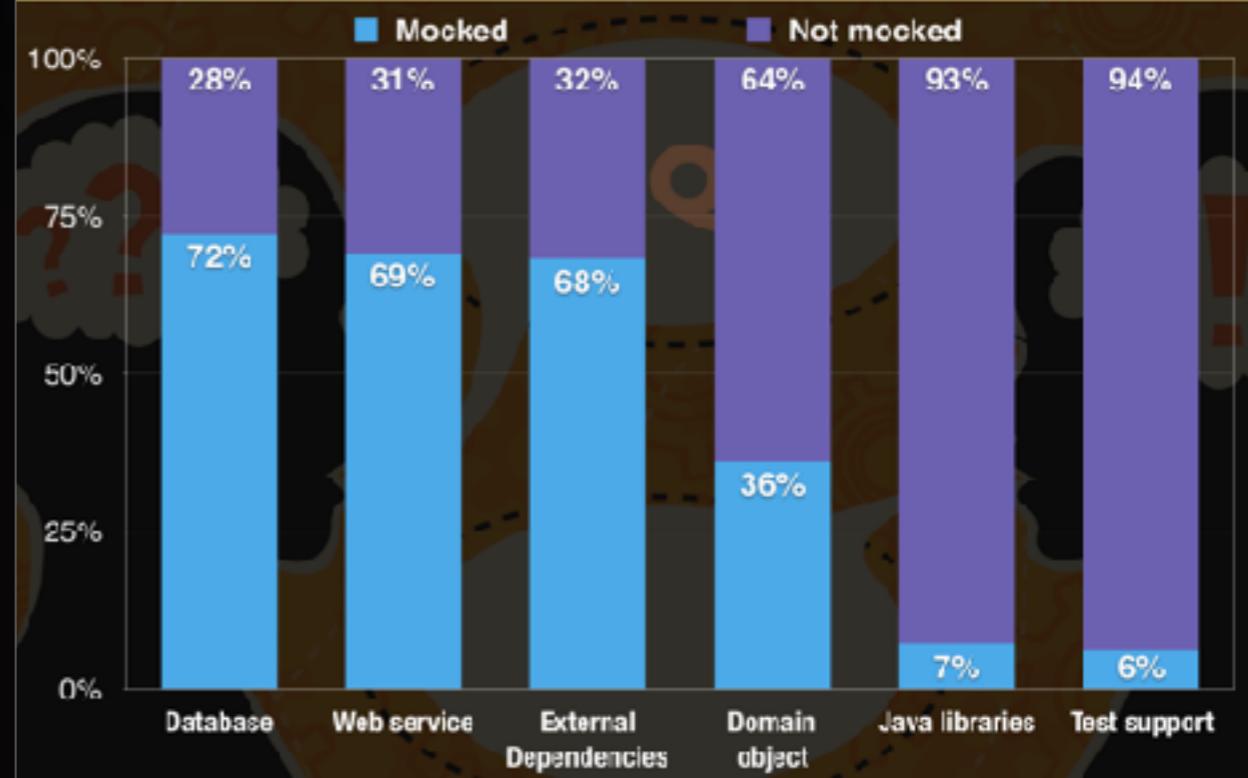


Open questions

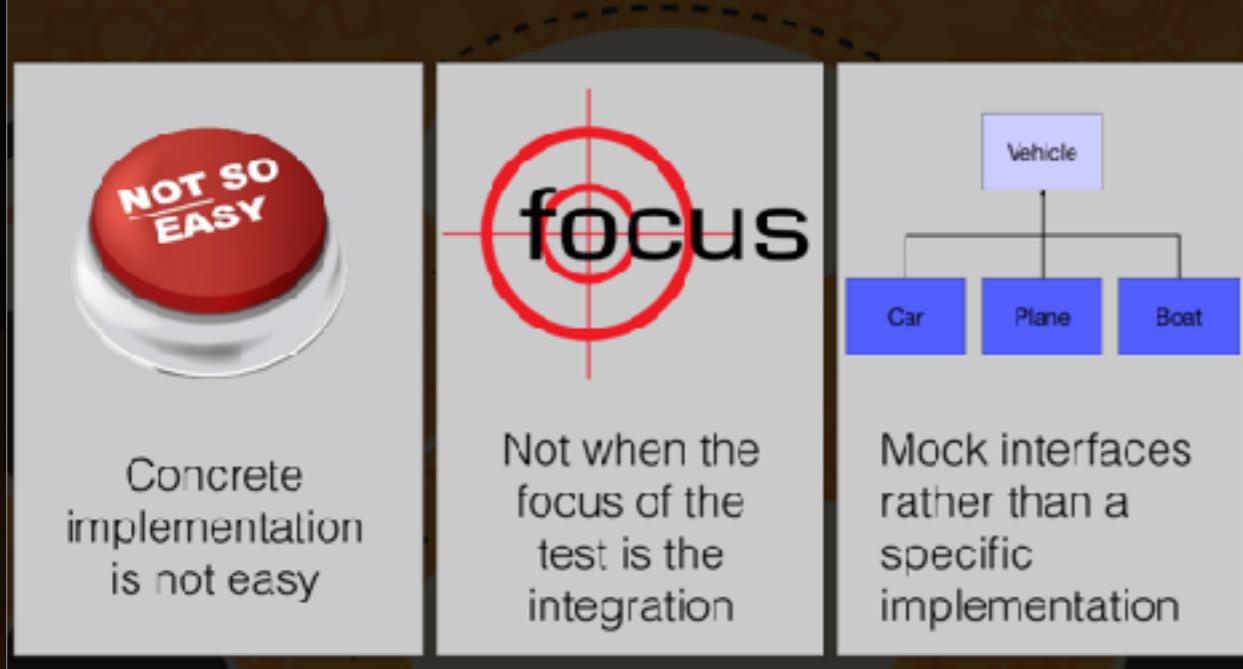
- Software quality vs mocks
- Are slow tests more mocked?
- How faster are tests with mocks?



RQ1: What test dependencies do developers mock?



RQ2: Why do developers decide to (not) mock a dependency?



RQ3: Which are the main challenges experienced with testing using mocks?

