Super fast end-to-end tests

Lars Thorup
ZeaLake Software Consulting

August, 2020

ZeaLake
software consulting

Lars Thorup

- Software developer/architect
  - JavaScript, Python, SQL, C#
  - Continuous Delivery
- Coach
  - Agile engineering practices and tools
- @larsthorup

ZeaLake
software consulting

Agenda

- What's so bad about end-to-end tests?
- Why are unit tests not sufficient?
- Getting the best of both worlds
- Demo!

What's so bad about end-to-end tests?

* Thanks to Randall Munroe: xkcd.com/303/

ZeaLake software consulting

What's so bad about end-to-end tests?

- Not fast
  - just a few / minute
- Not precise
  - each test covers a lot of code
- Not simple
  - multiple processes, persistent state
- Not robust
  - fail sporadically

Unit tests have many nice properties

- Fast
- Precise
- Simple
- Robust

ZeaLake
software consulting

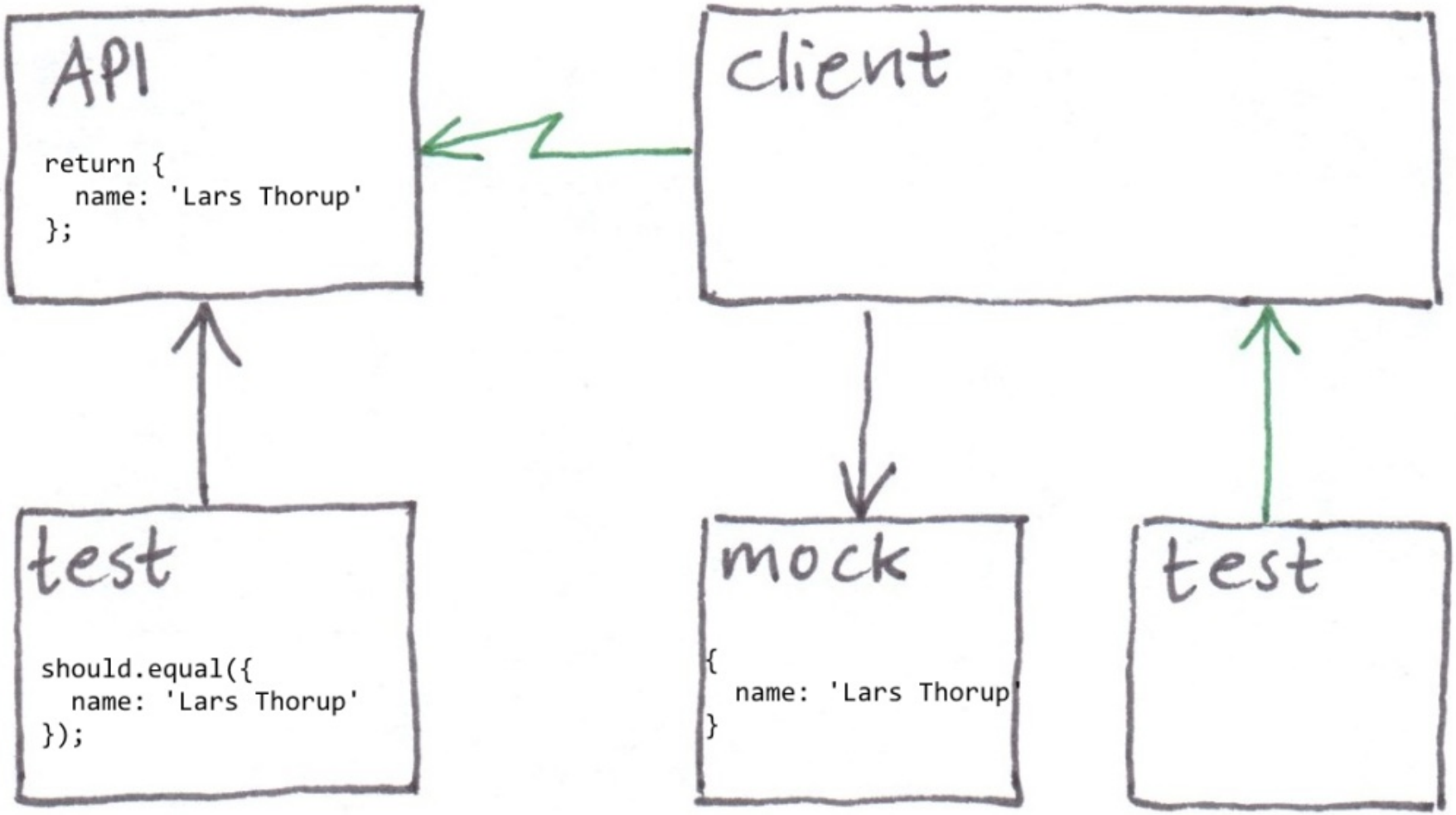
Why are unit tests not sufficient?

- To test code in isolation...
- ...dependencies are mocked
- So what happens when a dependency changes its interface?
- The application may break, while all unit tests pass!
- Because...
- ...mocks lie to you!

ZeaLake
software consulting

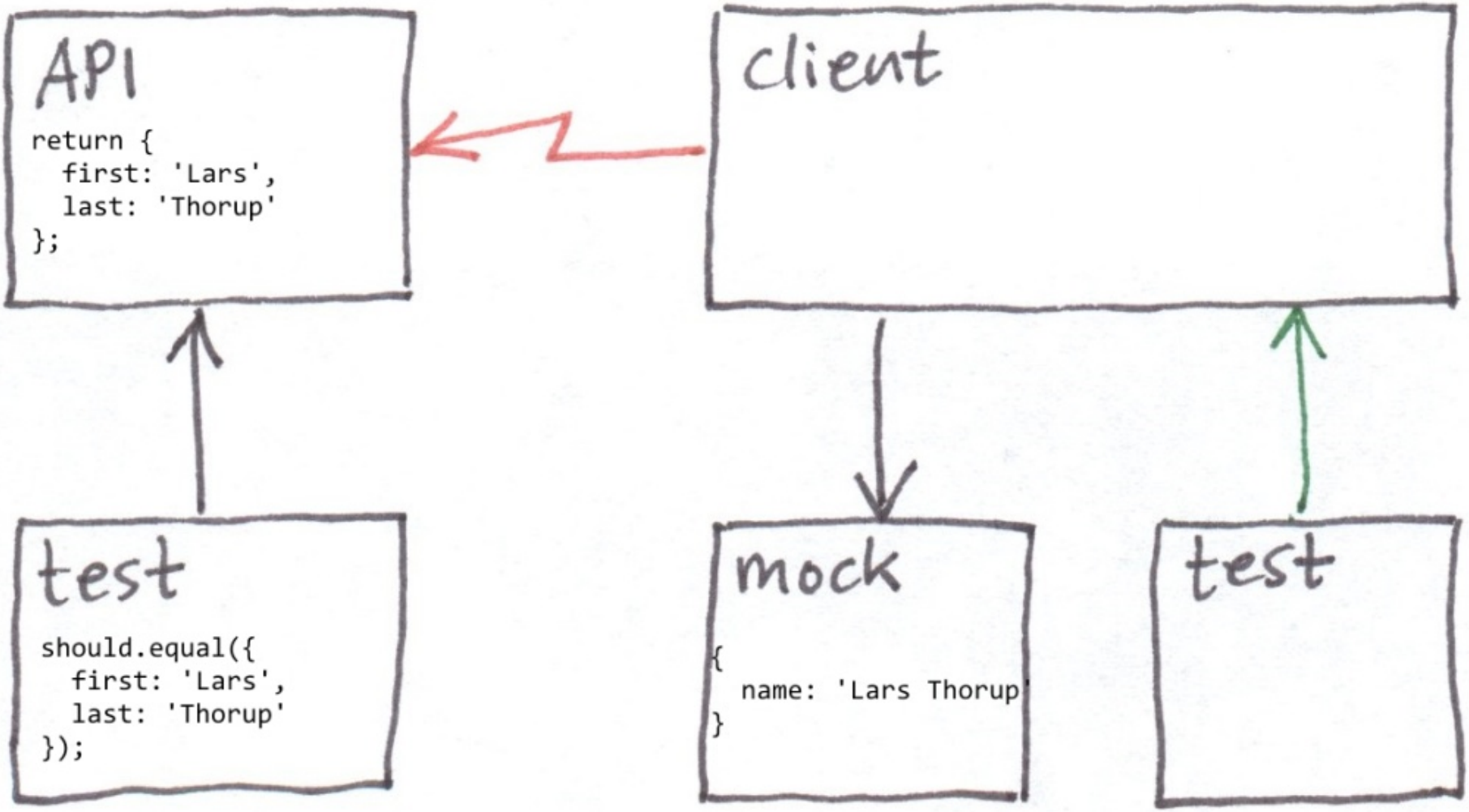
The problem with unit tests: mocks lie to you!

- All is well for now:



The problem with unit tests: mocks lie to you!

- When we change the interface - tests still pass 😱

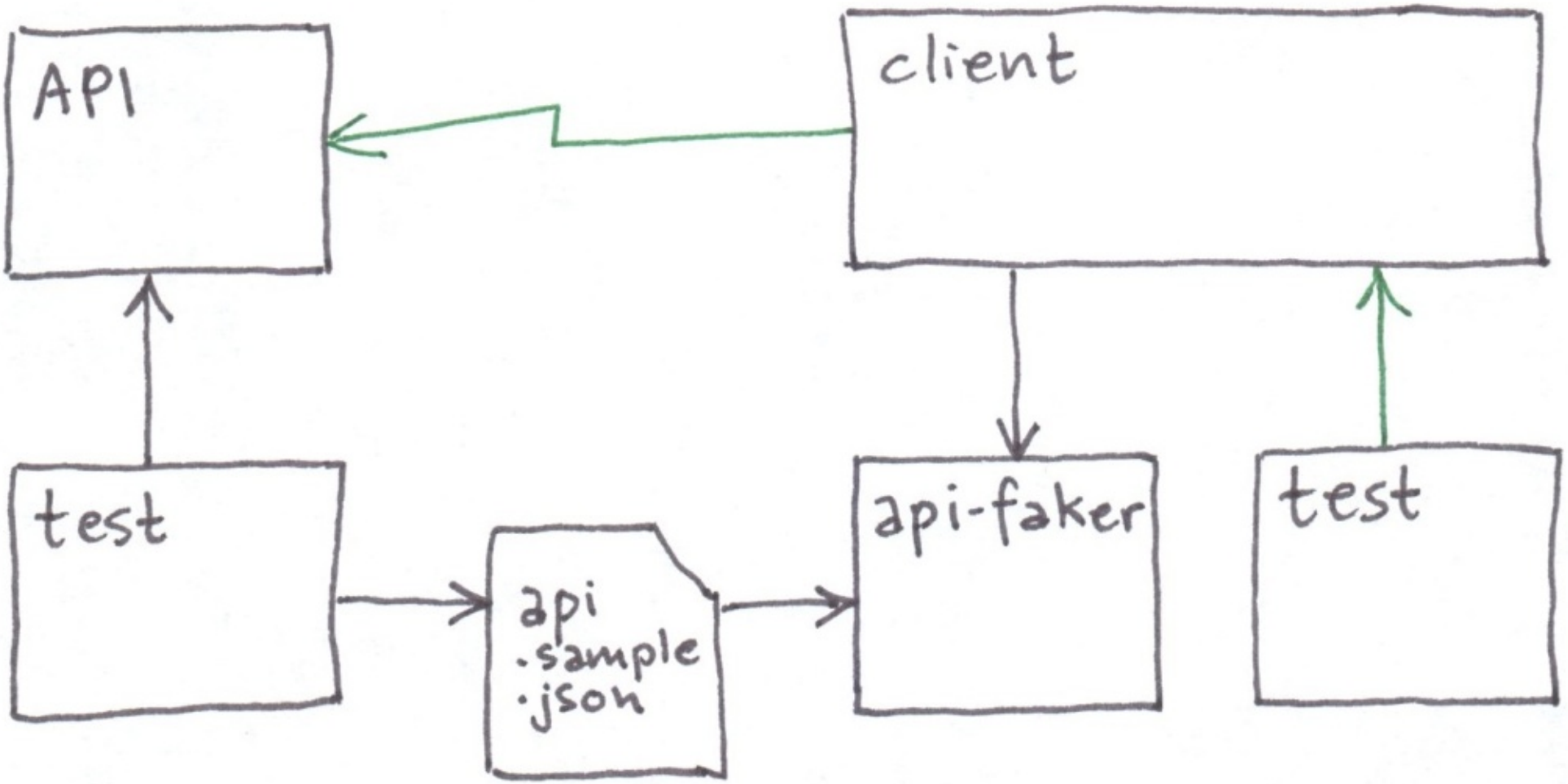


ZeaLake
software consulting

The problem with unit tests: mocks lie to you!

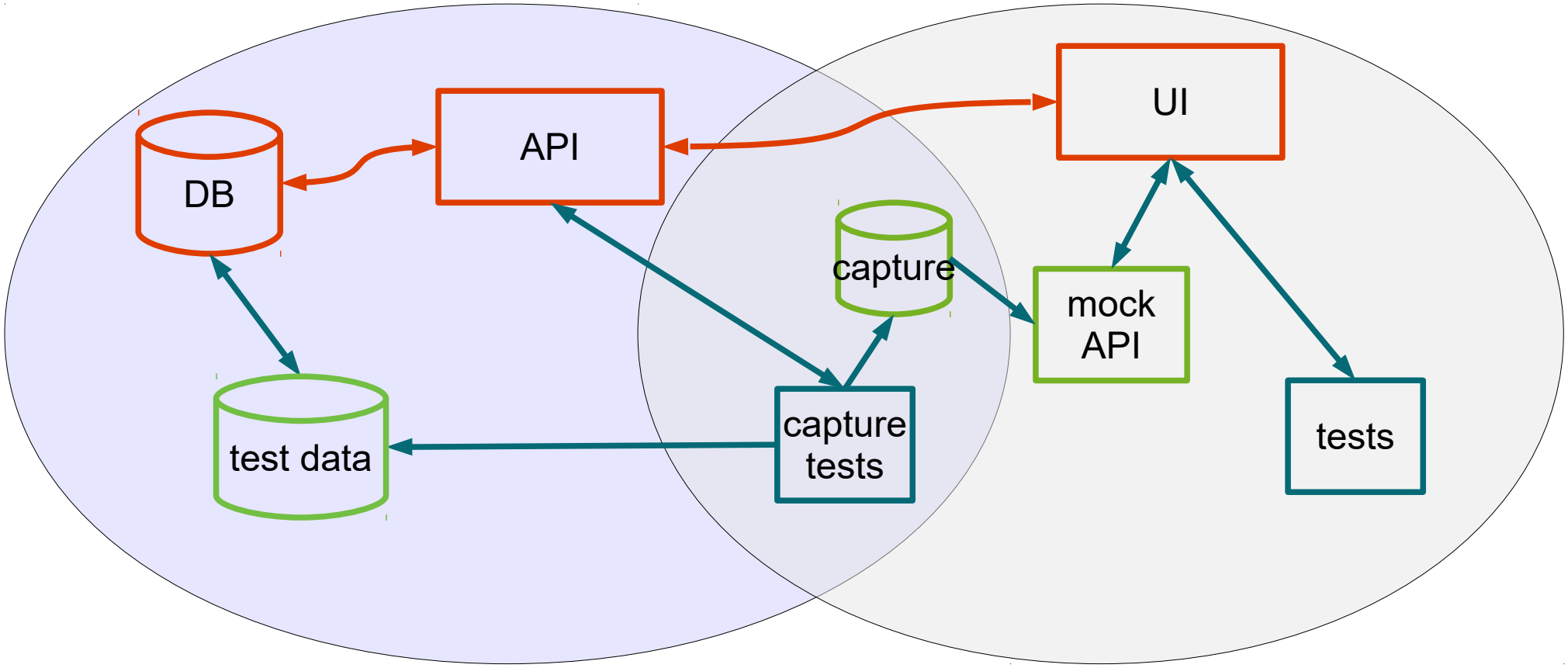
- How can mocks lie to us?
- ...because we hand-write them
- ...we copy assumptions that can change
- What if we could automatically generate correct mocks?

Getting the best of both worlds



ZeaLake
software consulting

Auto-mocking



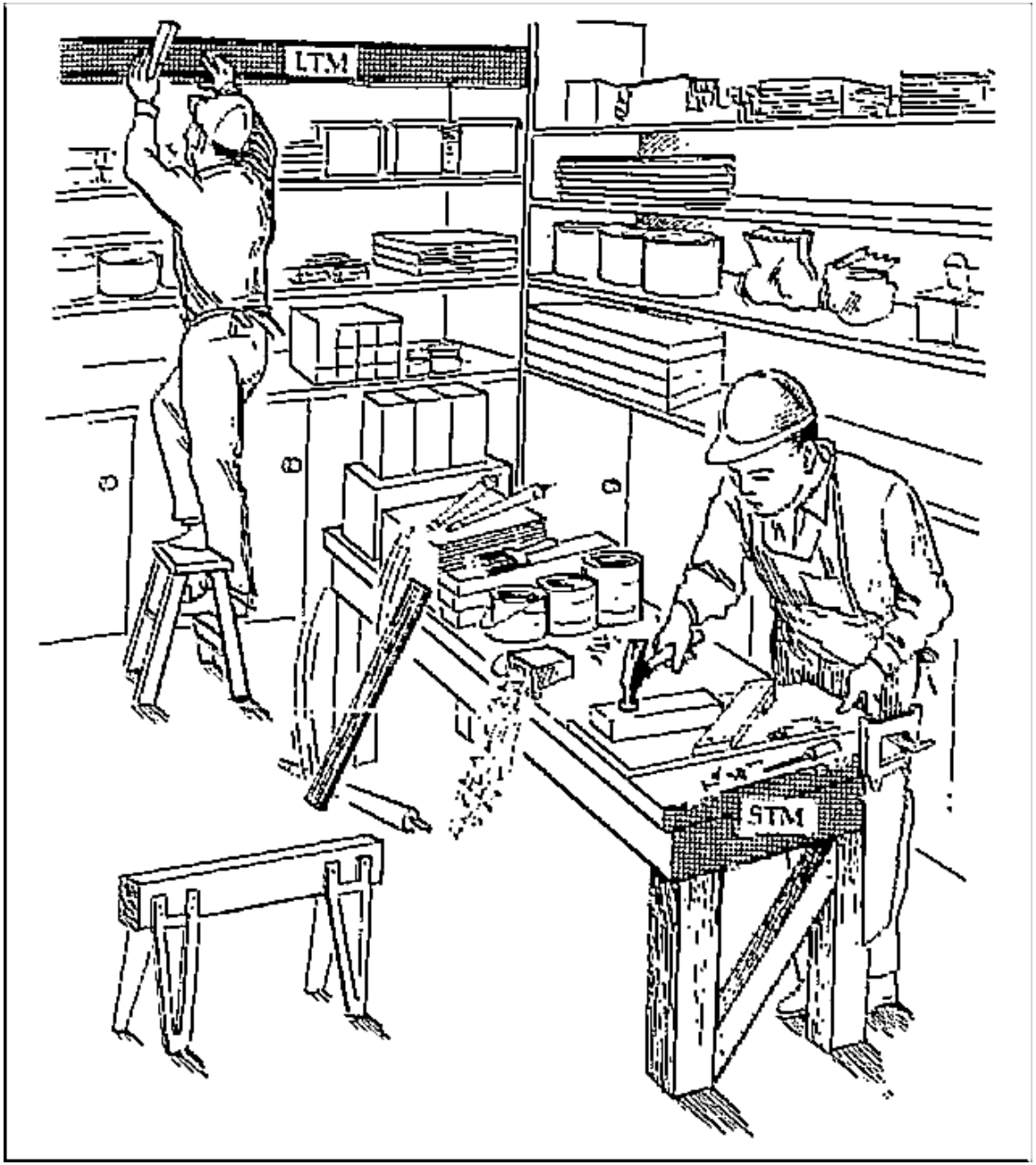
ZeaLake
software consulting

How fast can auto-mocking be?

- 1553 tests in 4 seconds

```
$ tz mocha
```

Okay, so how does this work?



ZeaLake
software consulting

Tools

- Polly.js
  - https://netflix.github.io/pollyjs/
- Hoverfly (Go, Python, Java)
  - https://docs.hoverfly.io/
- Others
  - VCR.py - https://vcrpy.readthedocs.io/
  - RubyVCR - https://relishapp.com/vcr/vcr/

Project experience

- Triggerz, 2016-2019
  - Node.js, React, Mocha
  - 1553 end-to-end unit tests, 4 seconds, 90% coverage
- Nets, 2019

  - Java, Angular, JUnit, Hoverfly
- BASE life science, 2020

  - Python, React, Jest, Polly.js
  - 98 end-to-end scenario tests, 45 seconds, 94% coverage

ZeaLake
software consulting

Sample code, Hoverfly

- Capture

```
Hoverfly hoverfly = new Hoverfly(localConfigs(), HoverflyMode.CAPTURE);
hoverfly.start();

// ... do HTTP requests

hoverfly.exportSimulation("/some/path");
hoverfly.close();
```

- Simulate

```
Hoverfly hoverfly = new Hoverfly(localConfigs(), HoverflyMode.SIMULATE);
hoverfly.start();
hoverfly.simulate(SimulationSource.file("/some/path")));

// ... do HTTP requests

hoverfly.close();
```

ZeaLake
software consulting

Sample code, Polly.js

- Setup

```
const baseOptions = {
  adapters: ['node-http'],
  persister: 'fs',
  persisterOptions: {fs: {recordingsDir}},
}
```

- Record

```
const options: PollyConfig = {
  ...baseOptions,
  mode: MODES.RECORD,
}
const polly = new Polly('MetricView', options)

// ... do HTTP requests

polly.stop()
```

- Replay

```
const options: PollyConfig = {
  ...baseOptions,
  mode: MODES.REPLAY,
  recordIfMissing: false
}
const polly = new Polly('MetricView', options)

// ... do HTTP requests

polly.stop()
```

ZeaLake
software consulting

Sample capture


```
{
  "log": {
    "_recordingName": "MetricOverview",
    "entries": [
      {
        "_id": "93e02b848a38c42434806d0e19abe8bd",
        "request": {
          "bodySize": 0,
          "cookies": [],
          "headers": [ ···
          ],
          "headersSize": 284,
          "httpVersion": "HTTP/1.1",
          "method": "GET",
          "queryString": [],
          "url": "http://localhost:3030/version"
        },
        "response": {
          "bodySize": 43,
          "content": {
            "mimeType": "application/json",
            "size": 43,
            "text": "{\"BASE-Migration\":\"2.2.1-dev+sha.6f5c8348\"}"
          },
          "cookies": [],
          "headers": [ ···
          ],
          "headersSize": 108,
          "httpVersion": "HTTP/1.1",
          "redirectURL": "",
          "status": 200,
          "statusText": "OK"
        }
      },
```


ZeaLake
software consulting

What about?

- Standard test data?
  - yes, you need that!
- Web sockets?
  - have not tried with this technique...
- External services?
  - very well suited for this!
- Stateful exchanges?
  - avoid 1) fetch A, 2) update A, 3) fetch A again
  - instead 1a) fetch A, 1b) update A, 2a) update B, 2b) fetch B
- Auto-capture instead of writing capture tests?
  - might work?

Questions!

```
$ tz mocha
```

ZeaLake
software consulting