



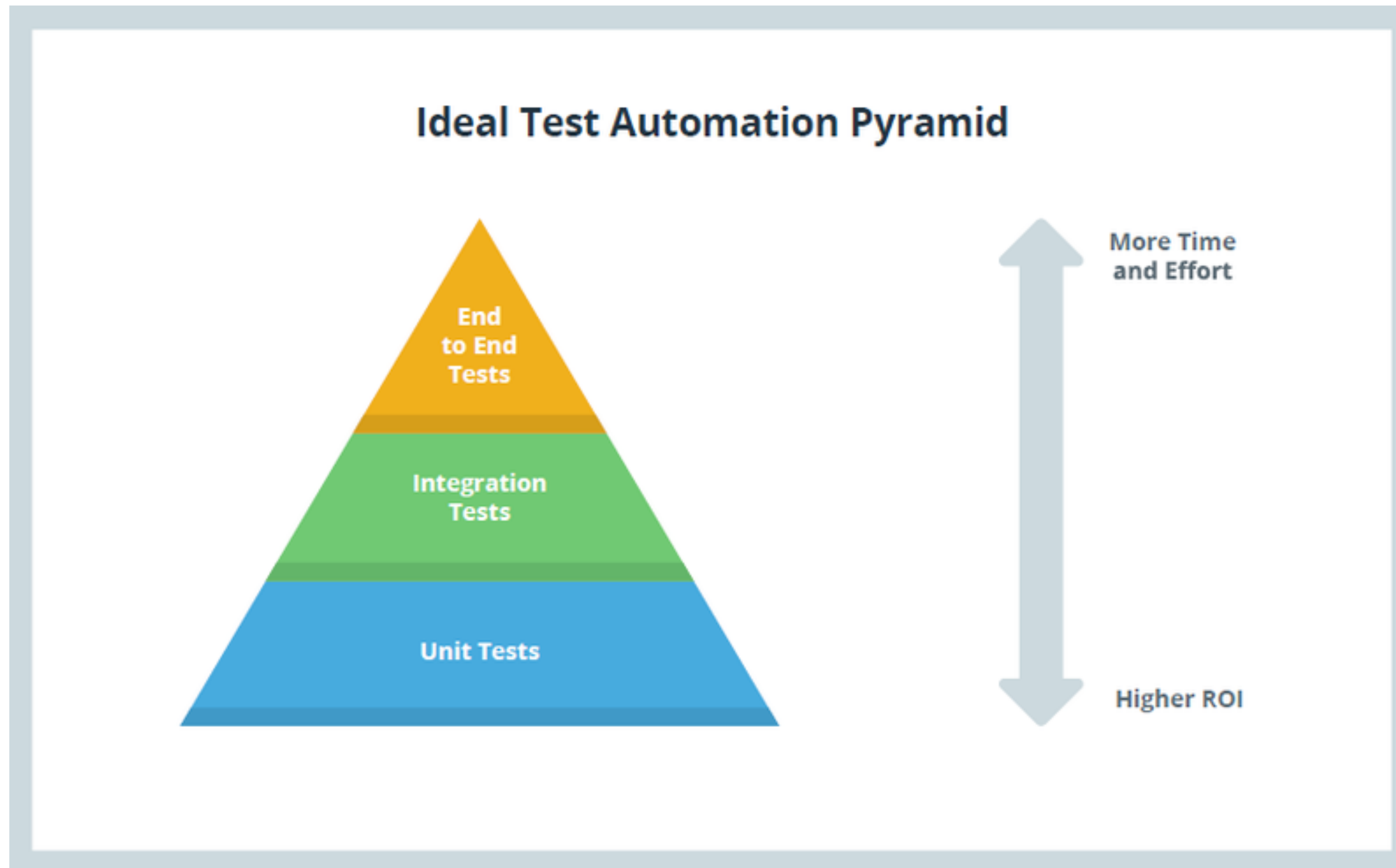
E2E Testing and You

IVAN GRGINOV

WHY SHOULD YOU WRITE TESTS?

TESTING FROM A DEV'S STANDPOINT

- Unit testing
- Integration testing
- Regression / visual regression testing
- E2E (end-to-end) testing == Black box testing
- ...



[HTTPS://BUGWOLF.COM/BLOG/THE-DEFINITIVE-GUIDE-TO-USER-ACCEPTANCE-TESTING-UAT](https://bugwolf.com/blog/the-definitive-guide-to-user-acceptance-testing-uat)

E2E TESTING

WHAT IS PUPPETEER?

Awesome is what it is...

```
const puppeteer = require('puppeteer');

(async () => {
  const browser = await puppeteer.launch();
  const page = await browser.newPage();
  await page.goto('https://example.com');
  await page.screenshot({path: 'example.png'});

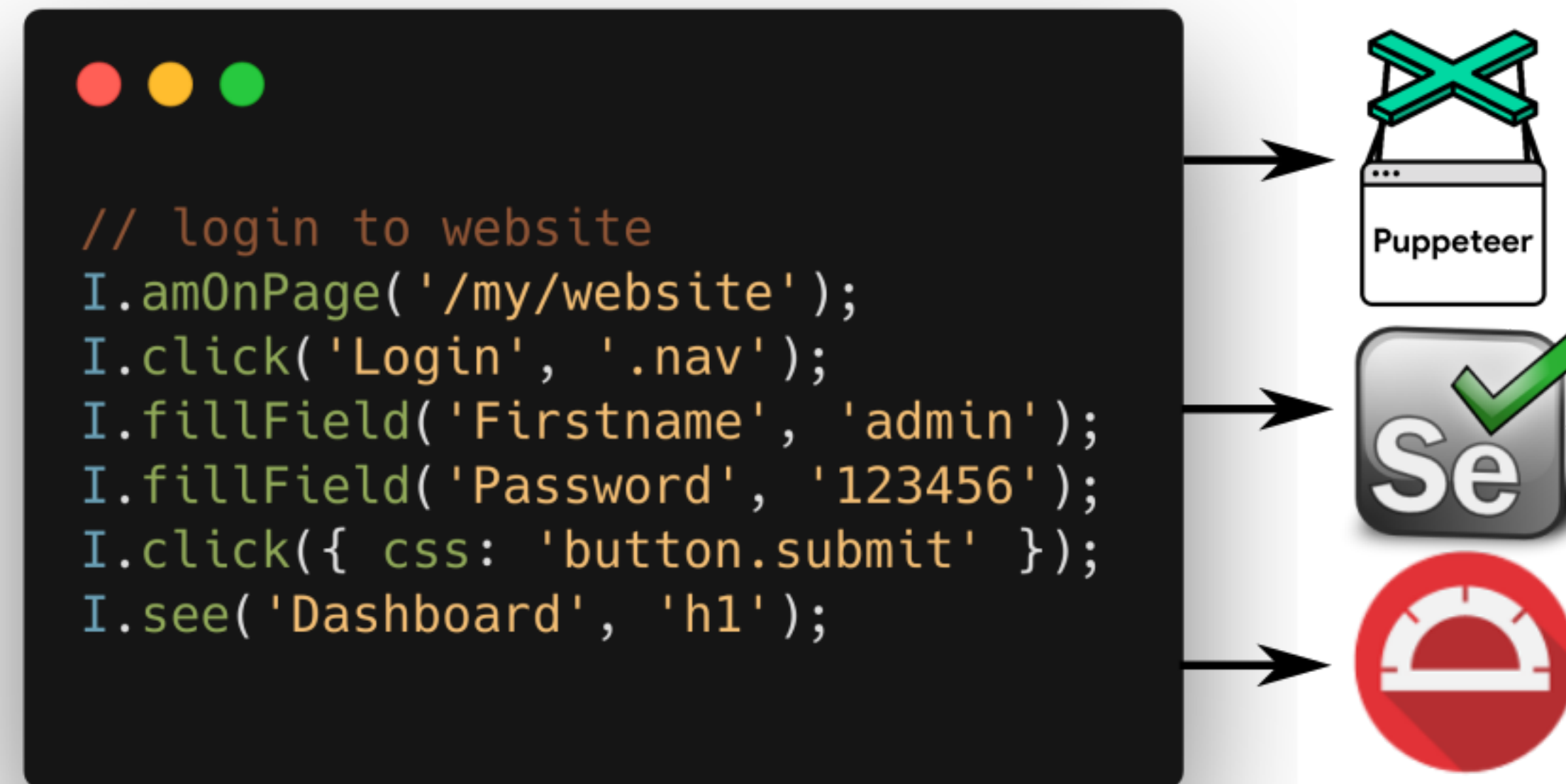
  await browser.close();
})();
```

JEST + PUPPETEER

JEST + PUPPETEER PROBLEMS

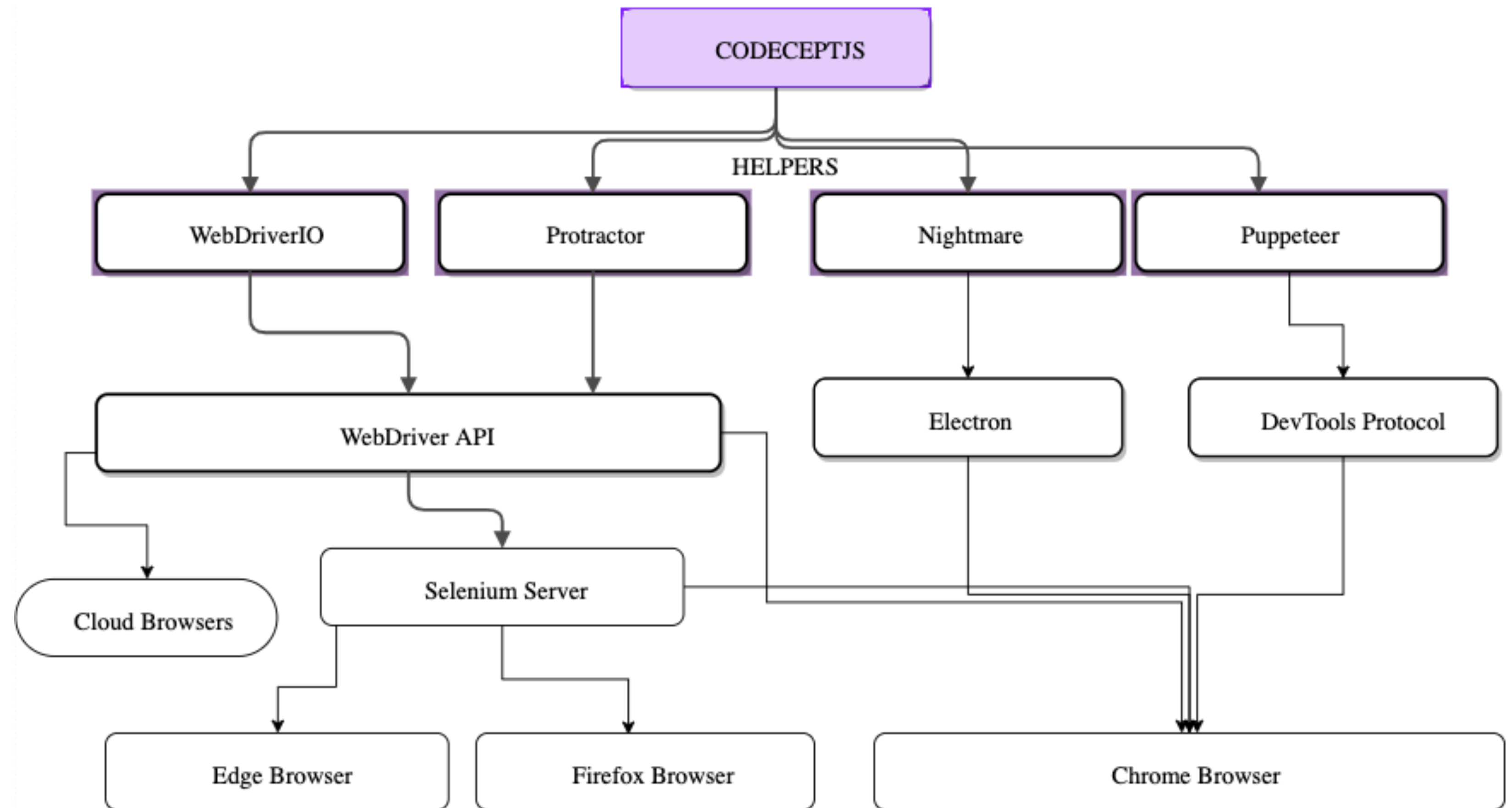
- Async / Await
- Managing Puppeteer
- Stability
- Chrome only
- Puppeteer not designed for testing

WHAT IS CODECEPTJS?



UNIFIED SYNTAX, MULTIPLE ENVIRONMENTS

- Selenium's WebDriver
- Puppeteer
- Protractor (AngularJS / Angular)
- Appium (Native mobile)
- Detox (Native mobile / React native)
- Nightmare
- TestCafe



EXAMPLE

[HTTPS://GITHUB.COM/IVANGRGINOVINF/CODECEPTJS-EXAMPLE](https://github.com/IVANGRGINOVINF/CODECEPTJS-EXAMPLE)

BROWSER AUTOMATION

```
Feature('Some feature');

Scenario('This thing we want to automate', (I) => {
  // Do something that you used to do manually but is cumbersome
  // (like for example filling a long form)
}).tag('@manual');
```

```
npx codeceptjs run --grep "This thing we want to automate" --steps
```

TIPS

USE PROPER SELECTORS

Selector	Recommended	Notes
<code>cy.get('button').click()</code>	⚠ Never	Worst - too generic, no context.
<code>cy.get('.btn.btn-large').click()</code>	⚠ Never	Bad. Coupled to styling. Highly subject to change.
<code>cy.get('#main').click()</code>	⚠ Sparingly	Better. But still coupled to styling or JS event listeners.
<code>cy.get('[name=submission]').click()</code>	⚠ Sparingly	Coupled to the <code>name</code> attribute which has HTML semantics.
<code>cy.contains('Submit').click()</code>	✅ Depends	Much better. But still coupled to text content that may change.
<code>cy.get('[data-cy=submit]').click()</code>	✅ Always	Best. Insulated from all changes.

[HTTPS://DOCS.CYPRESS.IO/GUIDES/REFERENCES/BEST-PRACTICES.HTML](https://docs.cypress.io/guides/references/best-practices.html)

USE CUSTOM LOCATORS

[HTTPS://CODECEPT.IO/LOCATORS](https://codecept.io/locators)

```
customLocator: {  
  enabled: true,  
  attribute: 'data-test',  
},
```

```
fields: {  
  bidderId: '[data-test=bidder_id]',  
  password: '[data-test=password]',  
  placeBid: '[data-test=place_bid]',  
},
```

==

```
fields: {  
  bidderId: '$bidder_id',  
  password: '$password',  
  placeBid: '$place_bid',  
},
```


WAITFORACTION & PRESSKEYDELAY === 🎯

```
helpers: {  
  Puppeteer: {  
    url: basicAuthRootUrl,  
    show: false,  
    windowSize: '1920x1080',  
    waitForAction: 0,  
    pressKeyDelay: 0,  
  },  
},
```

RESOURCES

- <https://github.com/IvanGrginovInf/codeceptjs-example>
- <https://codecept.io/basics>
- <https://codecept.io/quickstart>
- <https://docs.cypress.io/guides/references/best-practices.html>
- <https://www.monterail.com/blog/end-to-end-testing-with-codeceptjs>



Thank you! Questions?

IVAN GRGINOV