

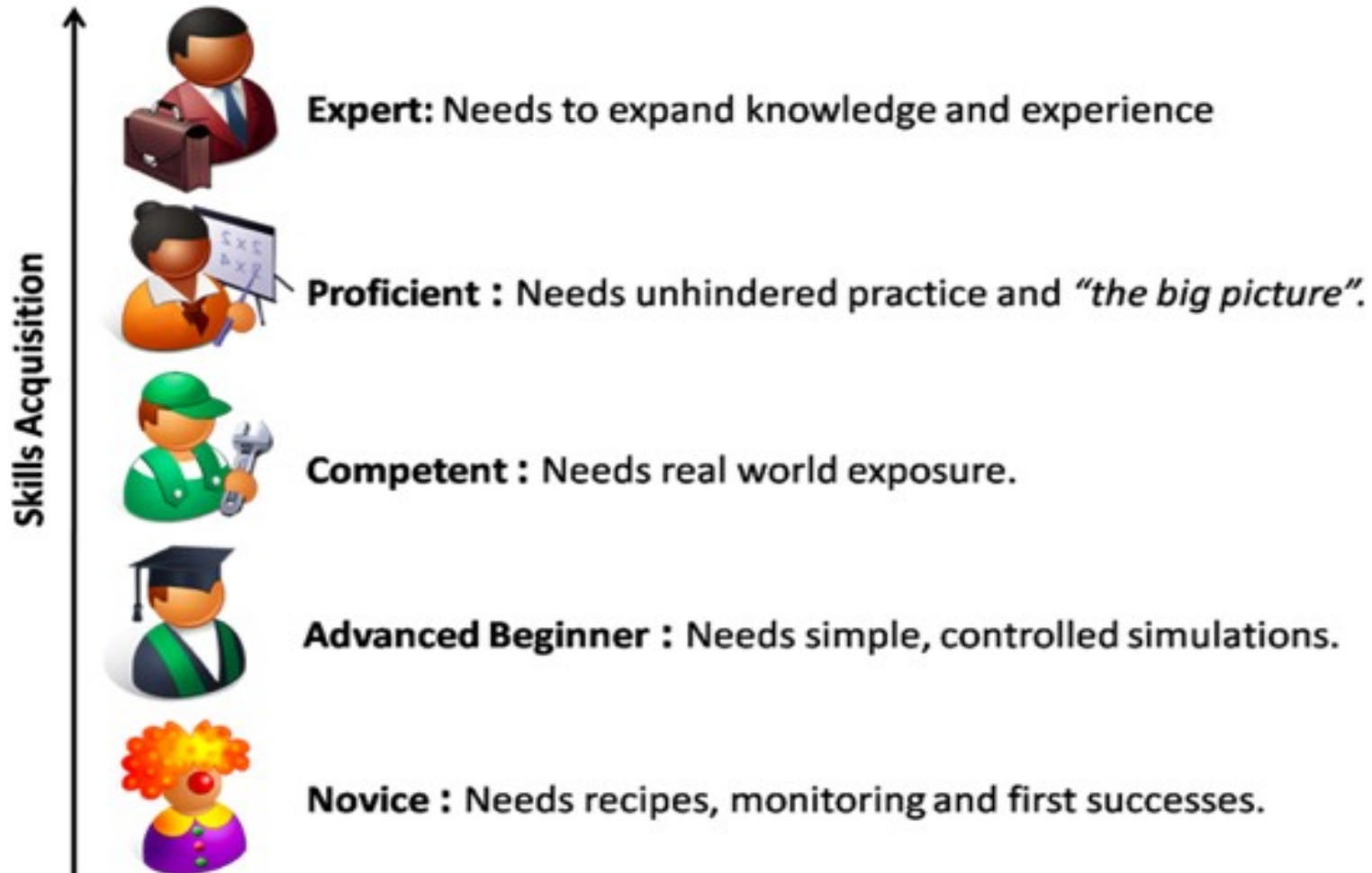
# **Geb**

## **Intro and Bootstrap**

Bob Herrmann  
Grails / Web Developer at [innocentive.com](http://innocentive.com)

# Geb

## The Dreyfus Model



# Geb

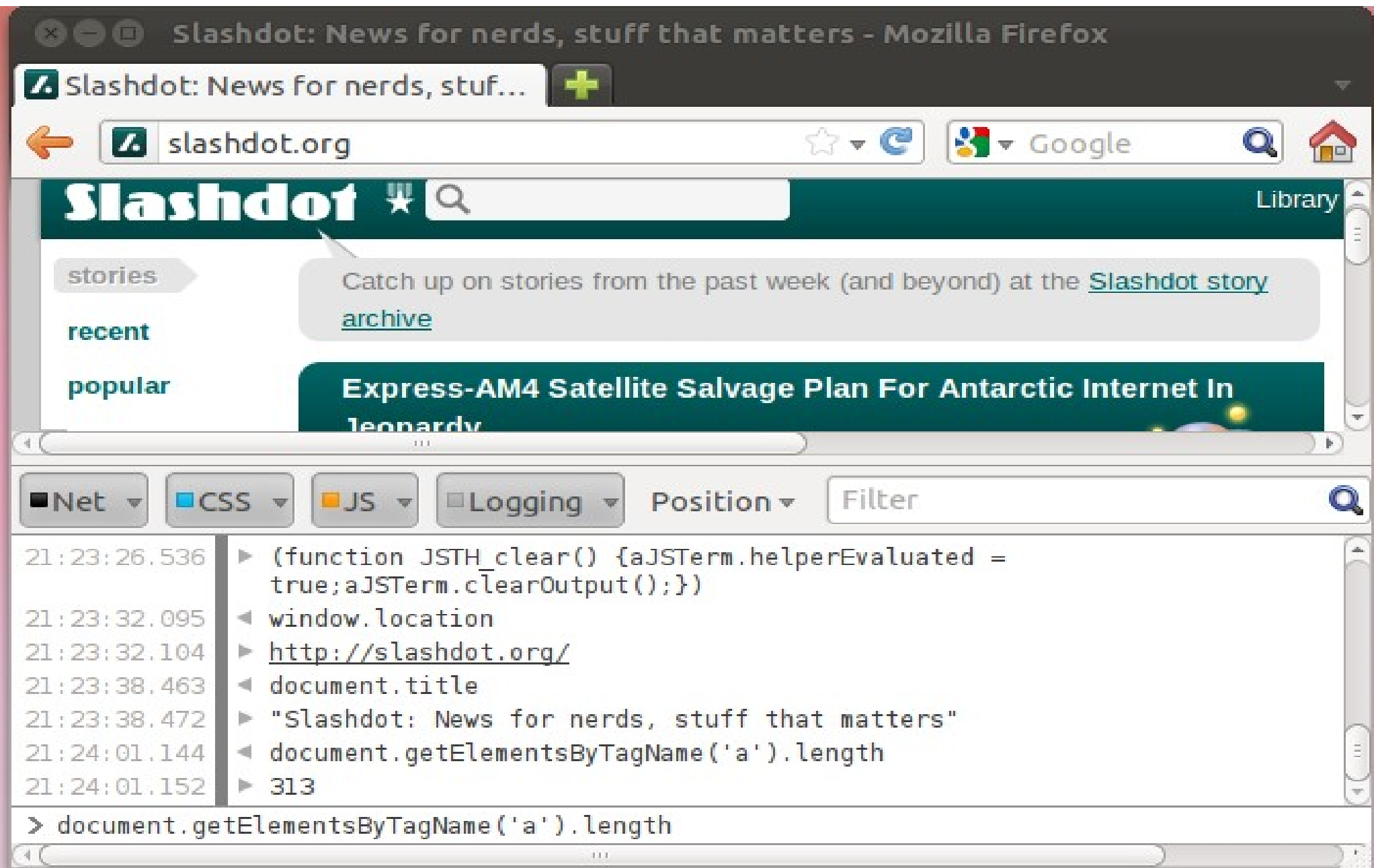
- What is the problem?
- What is the solution?
- Demo
- Extras

# Geb: What is the problem?

Testing web applications is hard

- Browsers are complicated (HTML, CSS, images, Javascript, async)
- Web page = horrid scary mess
- Parsing... Chrome/Firefox/IE/Safari

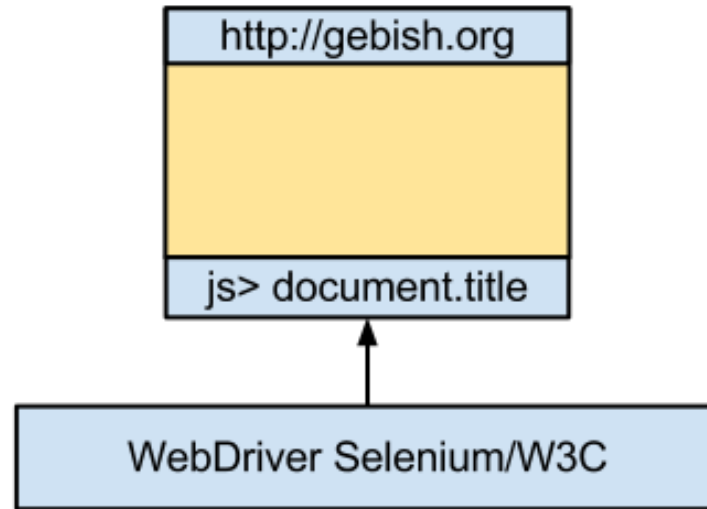
# Geb: An idea



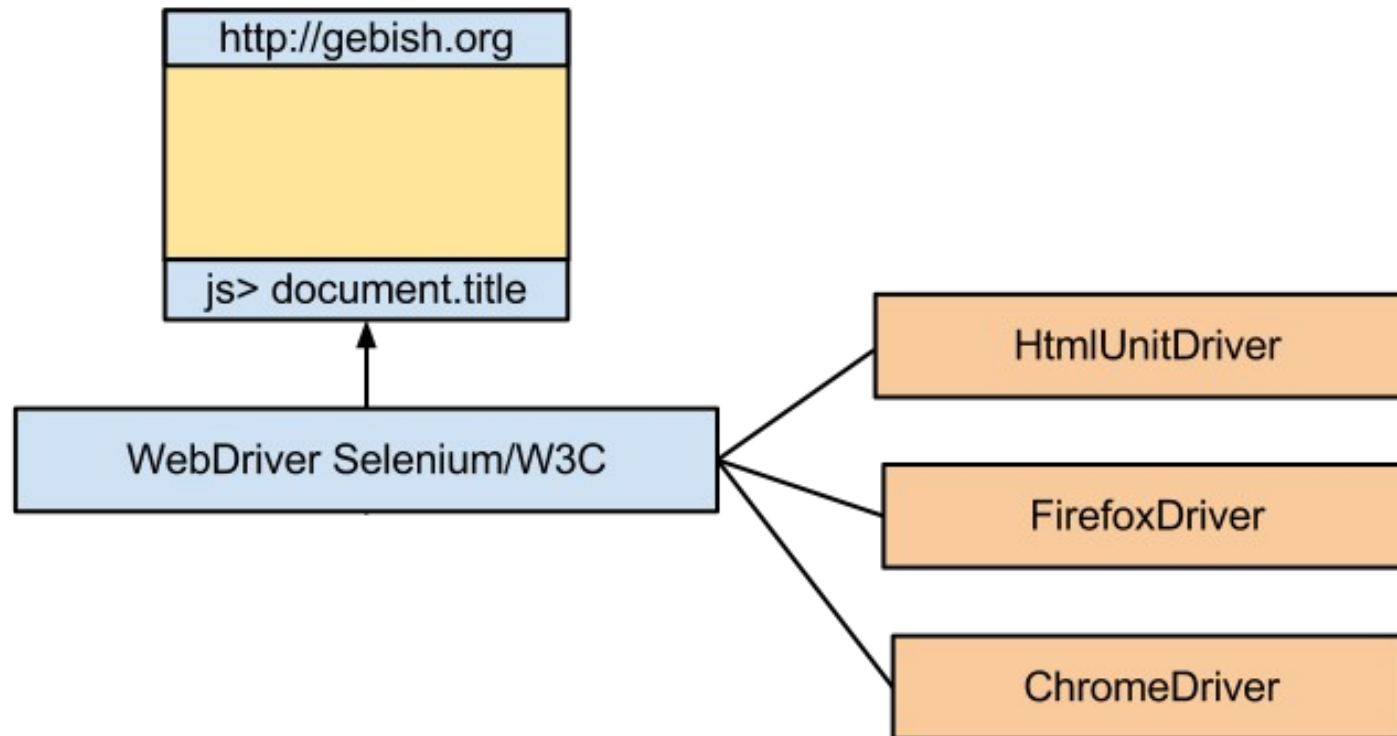
# Geb: Architecture



# Geb: Architecture

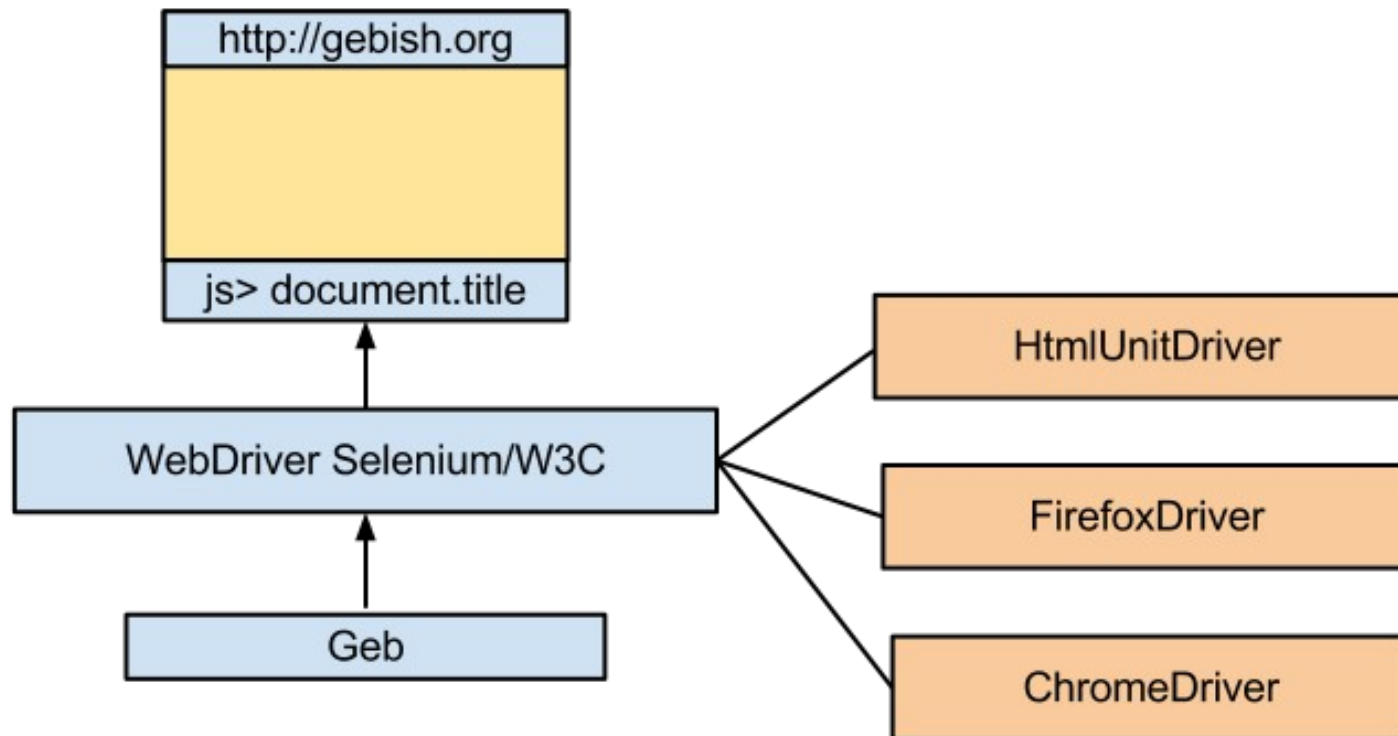


# Geb: Architecture

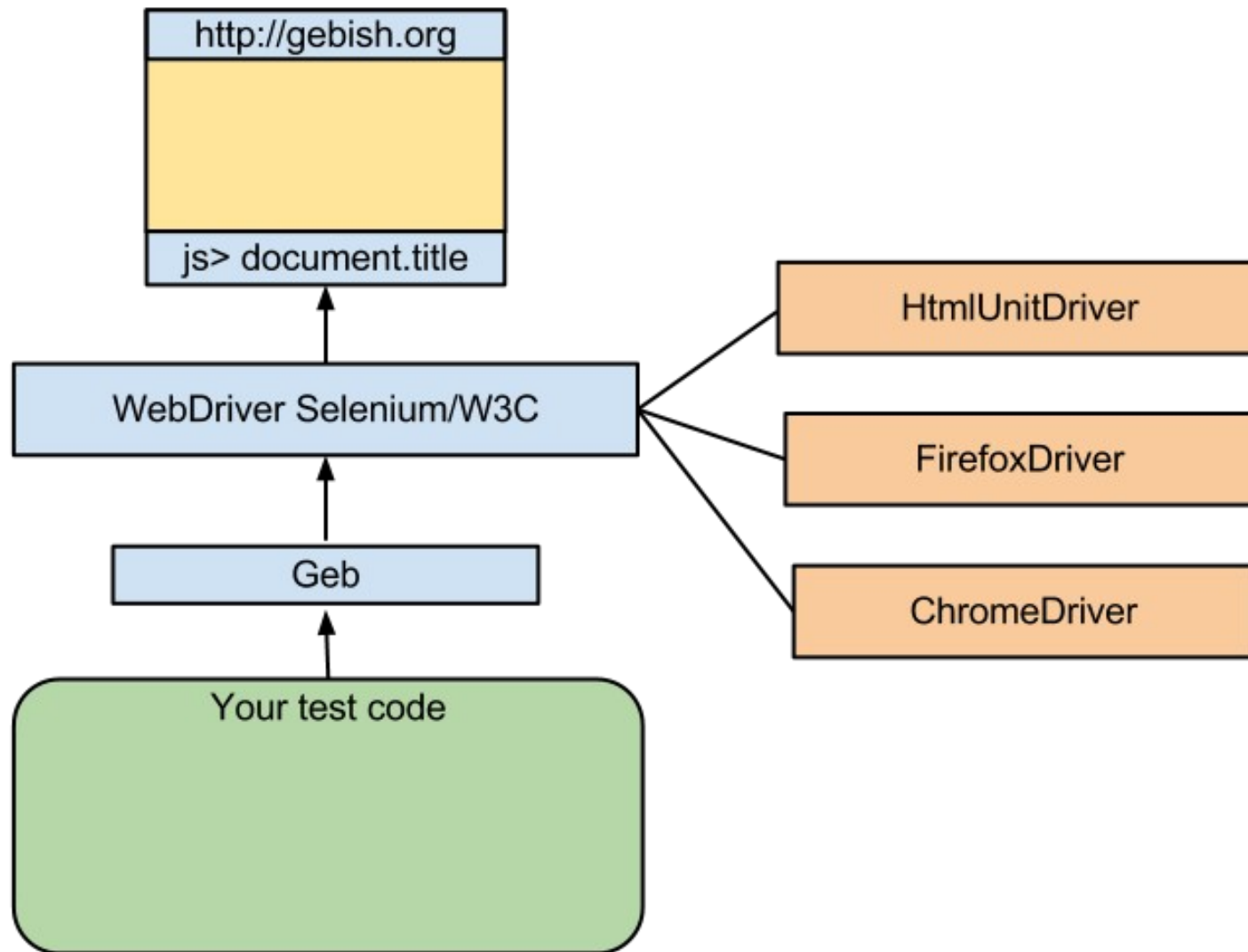




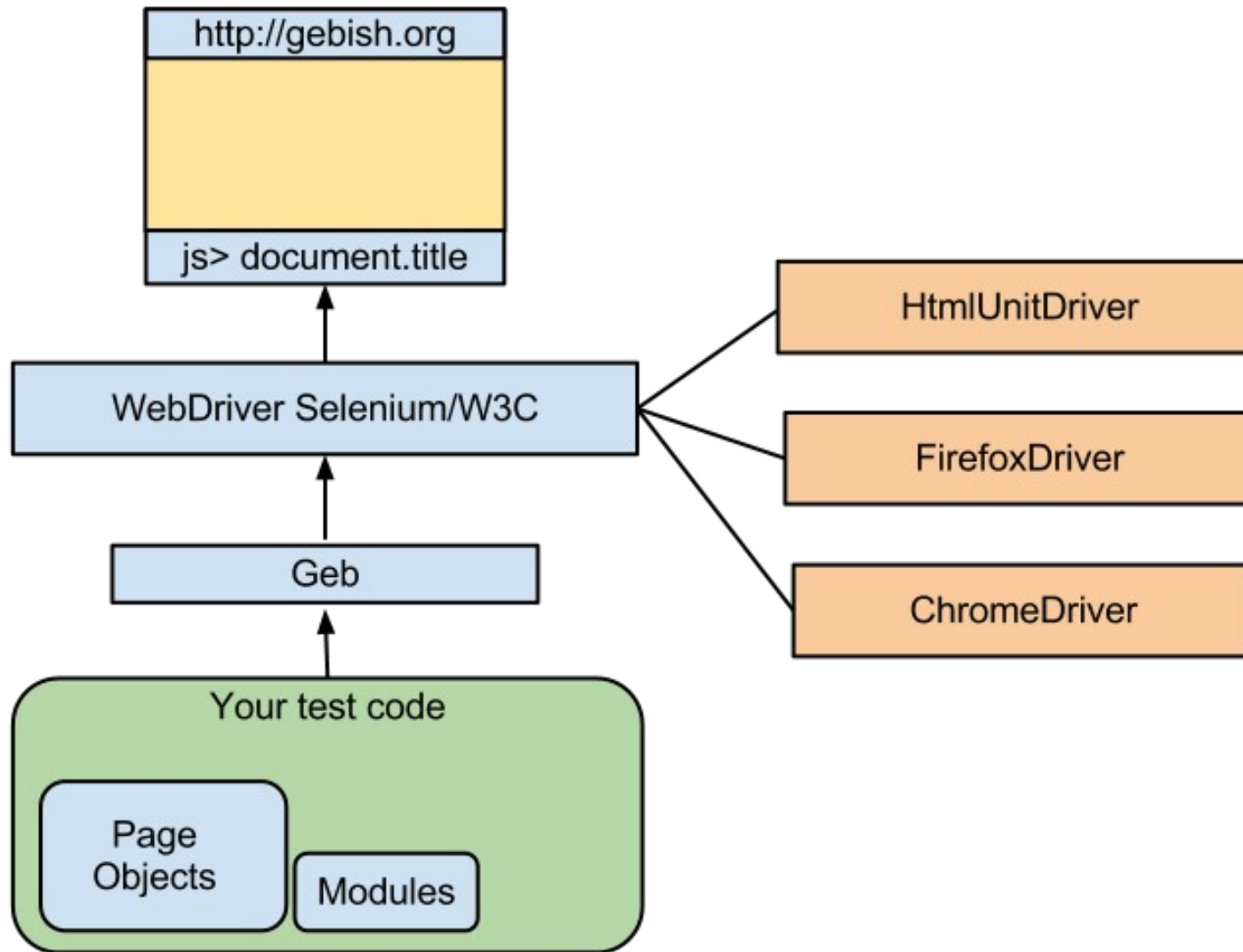
# Geb: Architecture



# Geb: Architecture



# Geb: Architecture



# Geb is...

**Browser automation (scripting, scraping, functional testing)**

**Created by Luke Daley**

**<http://gebish.org>      “The Book of Geb”**



**Active low volume mailing list**

**Pronounced with soft “G” like “Jeb”**

**Combines**

- **JQuery-like content selection**
- **Built on Selenium / WebDriver**
- **Page Object / Modules**

# Anatomy of a Geb Test



# Anatomy of a Geb Test

```
@RunWith(JUnit4)
class SimpleTest extends GebReportingTest {

    @Test
    void testSimple() {

    }

}
```

# Anatomy of a Geb Test

```
@RunWith(JUnit4)
class SimpleTest extends GebReportingTest {

    @Test
    void testSimple() {

        // Start browser at site

        // verify right page (Look at page)

        // Enter or click on stuff (click or type)

        // verify right thing happens

    }
}
```

# Anatomy of a Geb Test

```
@RunWith(JUnit4)
class SimpleTest extends GebReportingTest {

    @Test
    void testSimple() {
        go 'http://localhost:8080/s/login'

        assert $('title').text() == 'Taco Login'

        $('input[name=username]').value('bob')
        $('input[name=password]').value('taco4me')
        $('input[type=submit]').click()

        assert title == 'Welcome back'
        assert $('img').size() == 2
    }
}
```



# Anatomy of a Geb Test

```
@RunWith(JUnit4)
class SimpleTest extends GebReportingTest {

    @Test
    void testSimple() {
        go 'http://localhost:8080/s/login'

        assert $('title').text() == 'Taco Login'

        $('input[name=username]').value('bob')
        $('input[name=password]').value('taco4me')
        $('input[type=submit]').click()

        assert title == 'Welcome Back'
        assert $('img').size() == 2
    }
}
```

# Anatomy of a Geb Test

```
@RunWith(JUnit4)
class SimpleTest extends GebReportingTest {

    @Test
    void testSimple() {
        go 'http://localhost:8080/s/login'

        assert $('title').text() == 'Taco Login'

        $('input[name=username]').value('bob')
        $('input[name=password]').value('taco4me')
        $('input[type=submit]').click()

        assert title == 'Welcome Back'
        assert $('img').size() == 2
    }
}
```

# Geb's JQuery-like API

## Syntax

`$(<<css selector>>, <<index or range>>, <<attribute / text matchers>>)`

## Examples

`$('#myId')`

`$("a", text: "All about Geb")`

`$("div.some-class p:first[title='something']")`

`$("h1", 2, class: "heading")`

`$("p", name: "description")`

`$("ul.things li", 2)`

`$("p", text: contains("Geb"))`

`$("input", value: ~\d{3,}-\d{3,}-\d{3,}/)`

# Geb's JQuery like selectors

You found the element, now what?

`$.click()`

`$.text()`

`$.tag()`

`$.@attribute`

`$.value()`

`$.value('something')`

# Anatomy of a Geb Test

```
@RunWith(JUnit4)
class SimpleTest extends GebReportingTest {

    @Test
    void testSimple() {
        go 'http://localhost:8080/s/login'

        assert $('title').text() == 'Taco Login'

        $('input[name=username]').value('bob')
        $('input[name=password]').value('taco4me')
        $('input[type=submit]').click()

        assert title == 'Welcome Back'
        assert $('img').size() == 2
    }
}
```

# Anatomy of a Geb Test

```
@RunWith(JUnit4)
class SimpleTest extends GebReportingTest {

    @Test
    void testSimple() {
        go 'http://localhost:8080/s/login'

        assert $('title').text() == 'Taco Login'

        $('input[name=username]').value('bob')
        $('input[name=password]').value('taco4me')
        $('input[type=submit]').click()

        assert title == 'Welcome Back'
        assert $('img').size() == 2
    }
}
```

# Anatomy of a Geb Test

```
@RunWith(JUnit4)
class SimpleTest extends GebReportingTest {

    @Test
    void testSimple() {
        go 'http://localhost:8080/s/login'

        assert $('title').text() == 'Taco Login'

        $('input[name=username]').value('bob')
        $('input[name=password]').value('taco4me')
        $('input[type=submit]').click()

        assert title == 'Welcome Back'
        assert $('img').size() == 2
    }
}
```

# Anatomy of a Geb Test

## Demo



# Simple-Geb Quick Start

```
$ wget https://github.com/bherrmann7/simple-geb/zipball/master
```

```
$ # or use...
```

```
$ git clone git://github.com/bherrmann7/simple-geb.git
```

```
$ ./gradlew test # (or gradlew.bat)
```

```
$ # create IDE project
```

```
$ ./gradlew idea # (or gradlew eclipse) - can use free idea
```

```
$ # when running test add -Dgeb.build.reports=/tmp/
```

# Geb Dangers

- `go 'cnn.com'; $('a').size()`
- Wobbly – network, server responses
- Browser strangeness
- Development time sync
- tests consume wall clock time
- Flash / Applets / third party controls
- Ramp up slowly
- Becareful about what you get good at

# Advanced Geb

- WebDriver spec <https://dvcs.w3.org/hg/webdriver/shortlog>
- Environments -Dgeb.env=stage
- Ajax
  - `waitFor {}` // use default
  - `waitFor(10) {}` // up to 10 seconds, using the default retry interval
  - `waitFor(10, 0.5) {}` // up to 10 seconds, half a second in between retries
  - `waitFor("quick") {}` // use the preset “quick” as the wait settings
- `js.execute()`
  - `js.aVar = 7`
  - `js."window.title = 'boo ya'"`
  - `js.alert('hi Ho')`

# Advanced Geb

- Page Objects
- Modules

# Geb

Experiment

Play

Have Fun

with **Geb!**

<http://jadrn.com/geb>