

Geb

Intro and Bootstrap

Bob Herrmann
Grails / Web Developer at innocentive.com

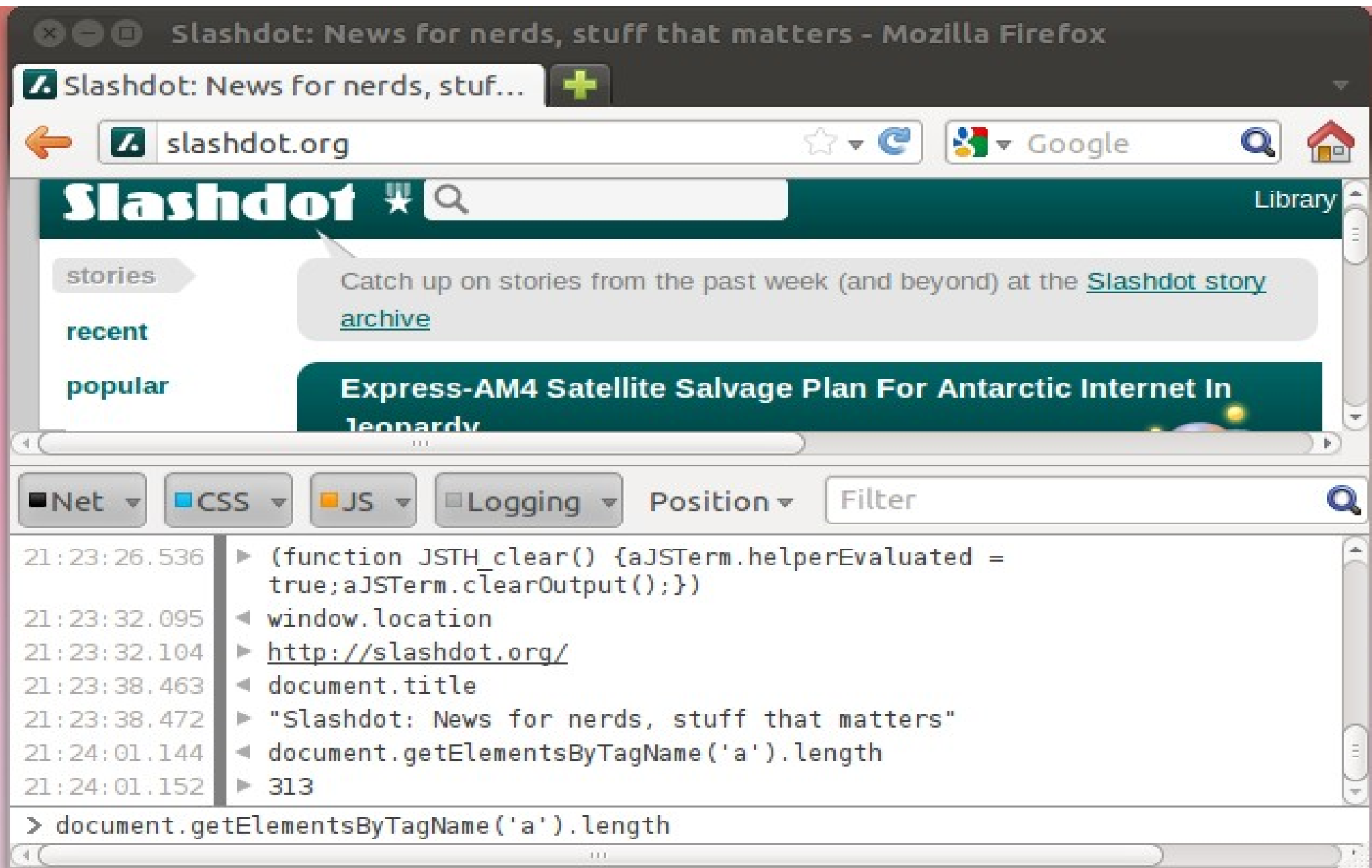
Geb

- 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)
- A web page is a horrid scary mess
- Who can understand a web page?
Chrome/Firefox/IE/Safari
aka Google/Microsoft/Apple (not you)

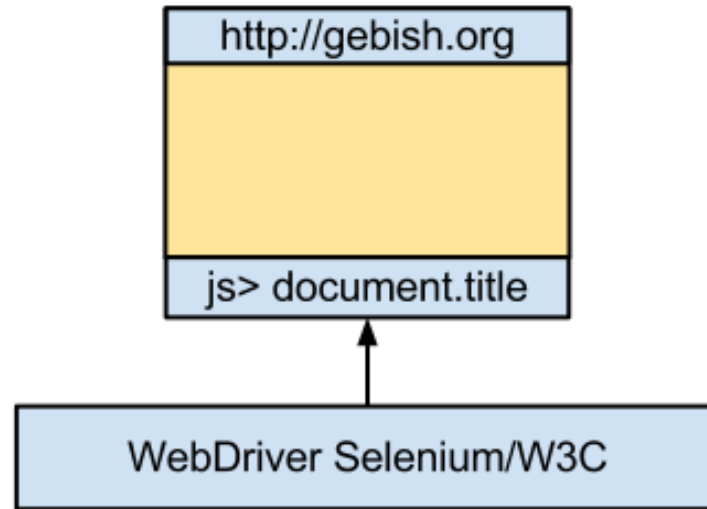
Geb: An idea



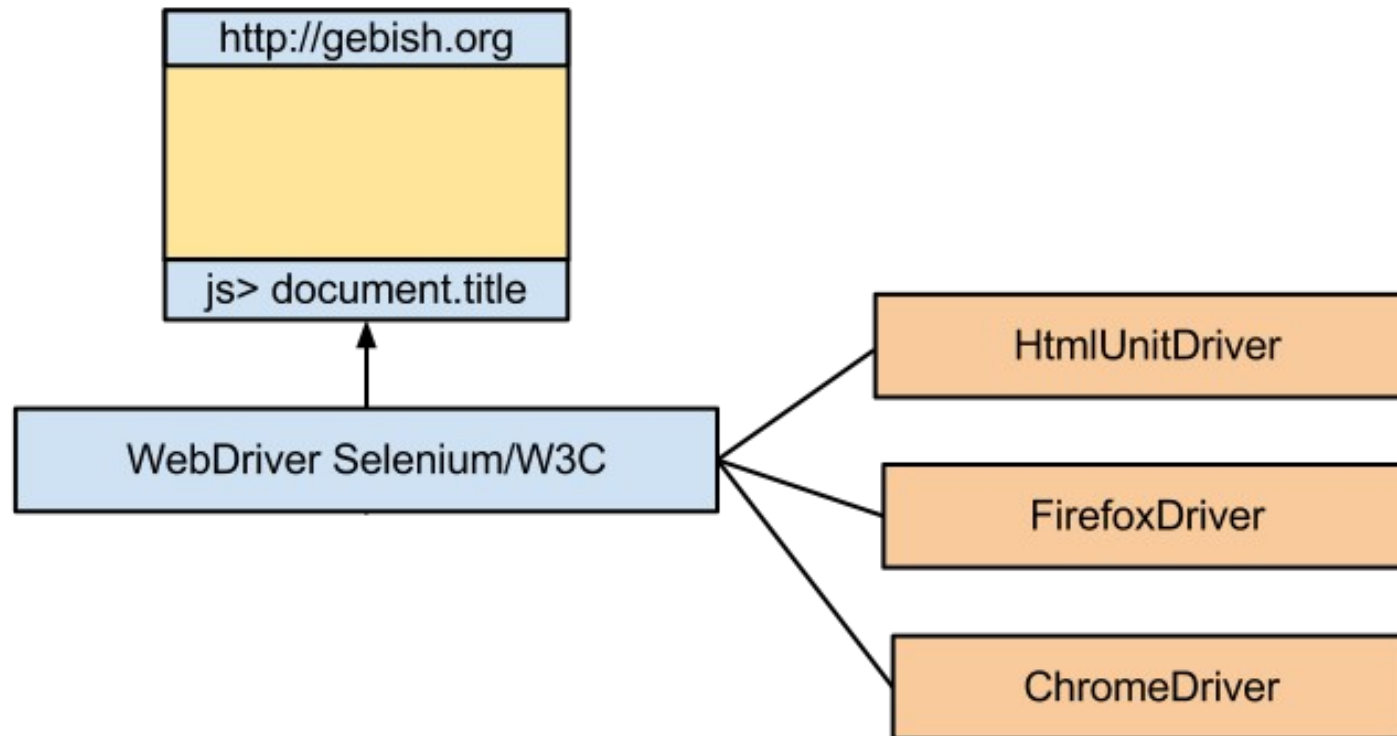
Geb: Architecture



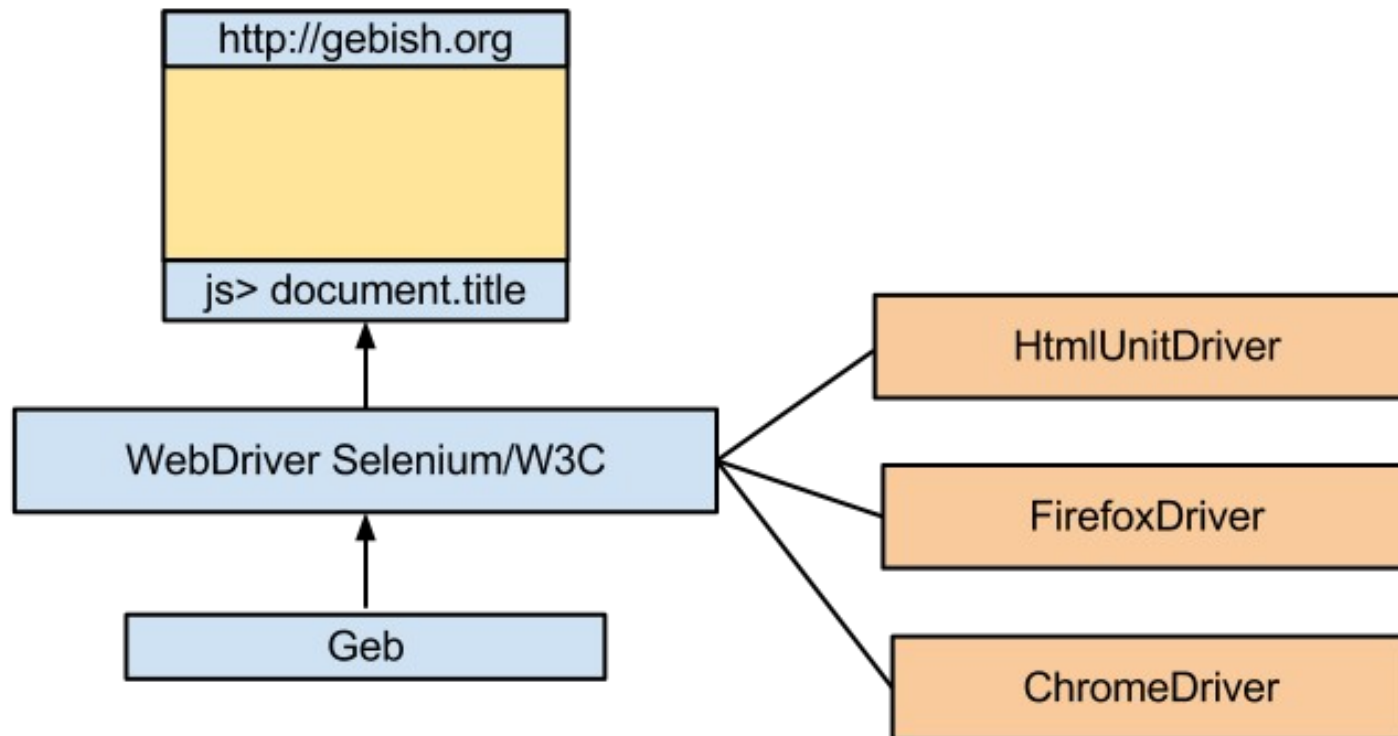
Geb: Architecture



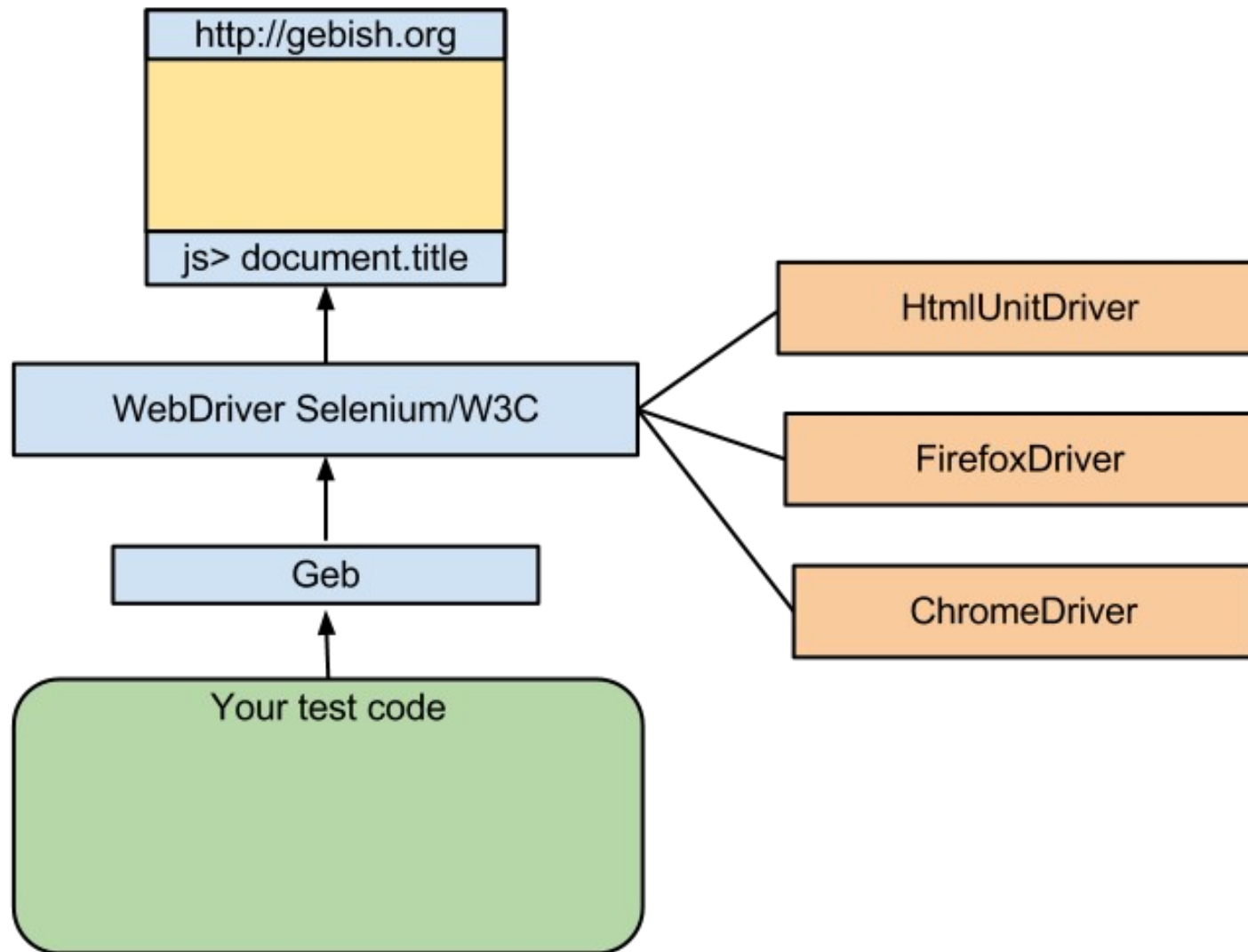
Geb: Architecture



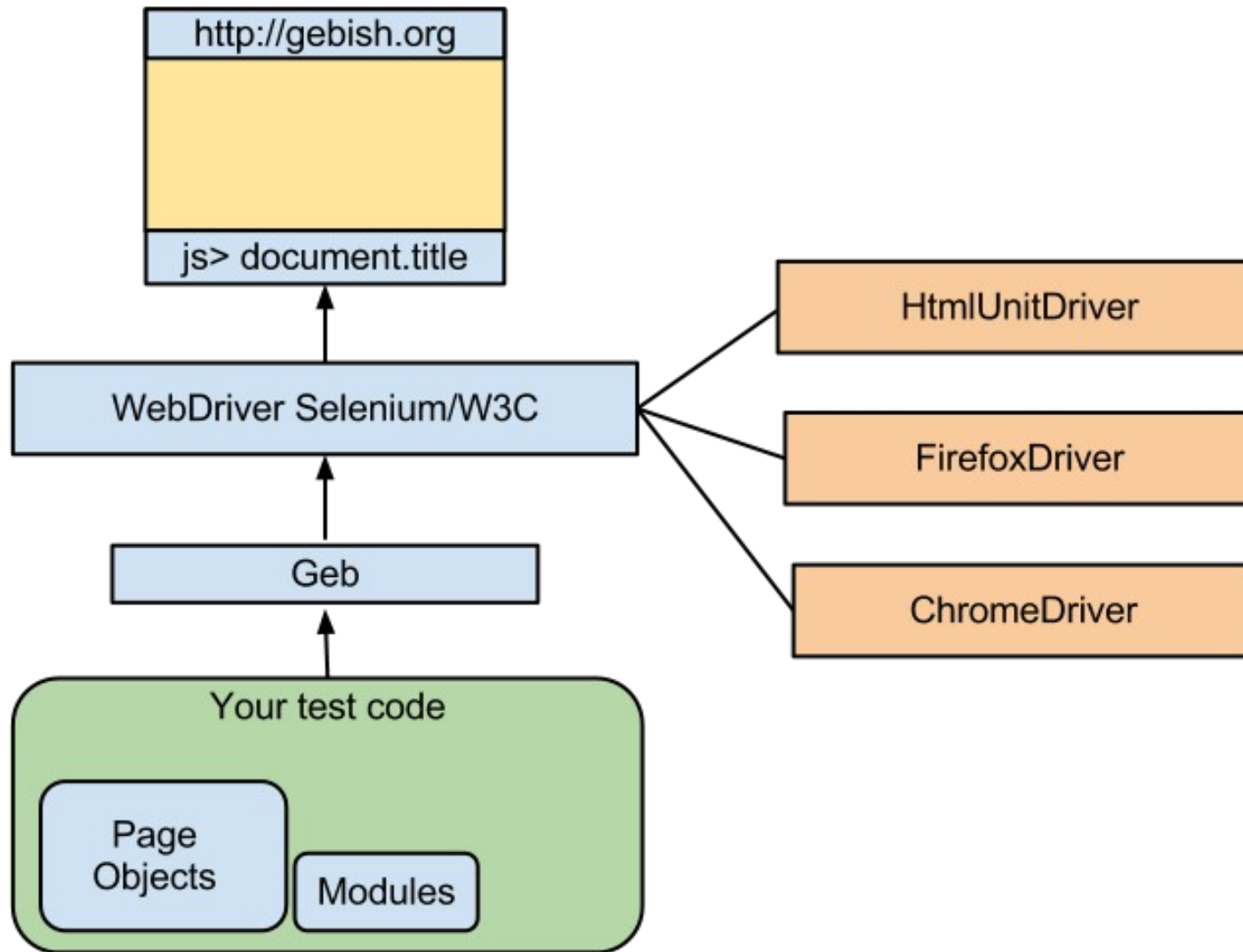
Geb: Architecture



Geb: Architecture



Geb: Architecture



Geb is...

Geb is a browser automation solution.

- **Build on Selenium / WebDriver**
- **Groovy**
- **jQuery-like content selection**
- **Page Object / Modules**
- **scripting, scraping, functional testing**
- **Junit, Spock & TestNG**

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

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

```
@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

quick getting started guide

\$ wget <https://github.com/bherrmann7/simple-geb/zipball/master>
or

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

\$ cd simple-geb

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

create IDE project

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

when running test add -Dgeb.build.reports

Geb Dangers

- `go 'cnn.com'; $('a').size()`
- Wobbly – network, server responses
- Browser strangeness
- Development enormous time sync
- Running tests consume wall clock time
- Flash / Applets / third party controls
- Mitigate dangers by on ramping slowly

More Geb

- <https://dvcs.w3.org/hg/webdriver/shortlog>
- Ajax
 - `waitFor(timeoutSeconds, intervalSeconds, Closure condition)`
 - `waitFor(timeoutSeconds, Closure condition)`
 - `waitFor(Closure condition)`
- `js.execute()`
 - `js.execute('document.title = "Bow Wow"')`
- Page Objects
- Modules