Agenda

- **Selenium**
  - **Introduction**
  - **Selenium Core**
  - **Different ways of using Selenium**
    - **Selenium Core (FIT style tables)**
    - **Selenium Remote Control(RC)**
    - **Selenium DIE**
    - **Selenium Grid**
- Other test tools
  - HtmlUnit
  - jWebUnit
  - JSFUnit
- White-box test (Unit test)
  - Define test objects, design test cases
- Black-box test (funktion test)
  - Define test objects, design test cases

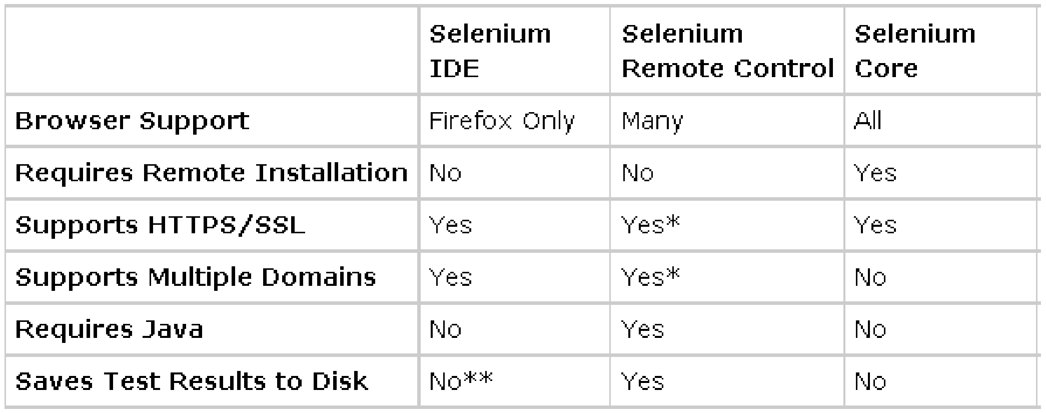
TODO:

What's Selenium?

- **Automated black-box web sites testing tool.**
- **It is created by ThoughtWorks, based on JavaScript.**
- **Used for acceptance / functional testing.**
- **Selenium tests run directly in a browser, just like real users do.**
- **Running on multiple browsers, multiple Operating Systems.**
- **Ajax testing support:**
  - **The problem: asynchronous and dynamic.**
  - **Solution:**
    - Action: waitFor…, waitNotFor…

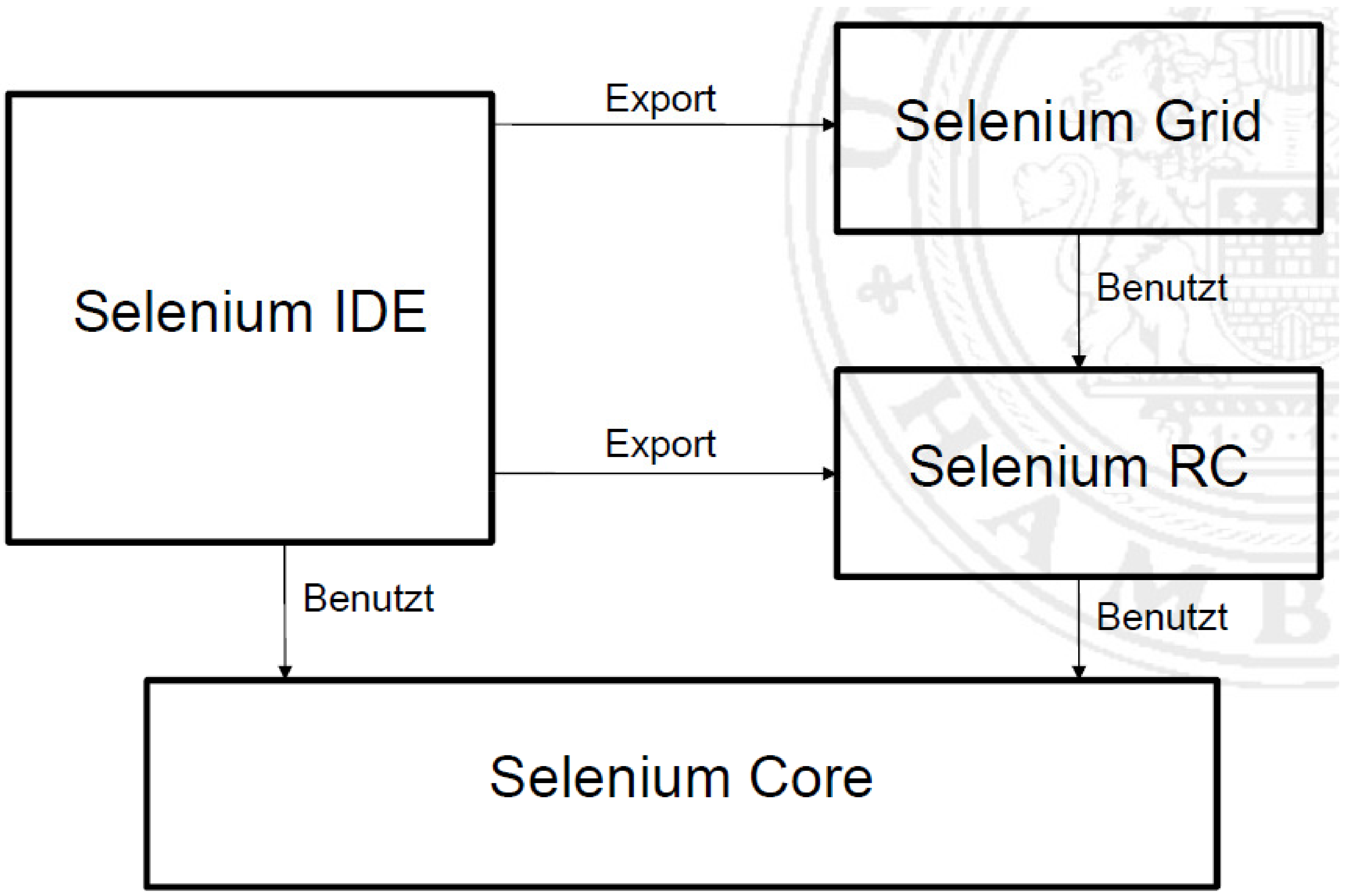
Different ways to use Selenium

- Test runner mode (Bot mode): Selenium Core
  - HTML tables
- Driven mode: Selenium Remote Control (RC)
  - Test cases in languages of choice (Java, C#, Python, Perl, Ruby…)
- Record mode: Selenium IDE
  - Firefox plug-in
- Selenium Grid –*TODO:*



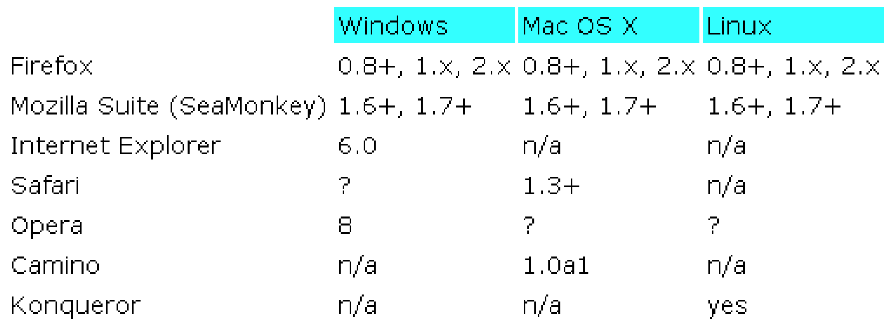
	Selenium IDE	Selenium Remote Control	Selenium Core
Browser Support	Firefox Only	Many	All
Requires Remote Installation	No	No	Yes
Supports HTTPS/SSL	Yes	Yes*	Yes
Supports Multiple Domains	Yes	Yes*	No
Requires Java	No	Yes	No
Saves Test Results to Disk	No**	Yes	No

– Selenium website


Selenium IDE
Export
Selenium Grid
Benutzt
Export
Selenium RC
Benutzt
Benutzt
Selenium Core

Selenium Core: test runner mode

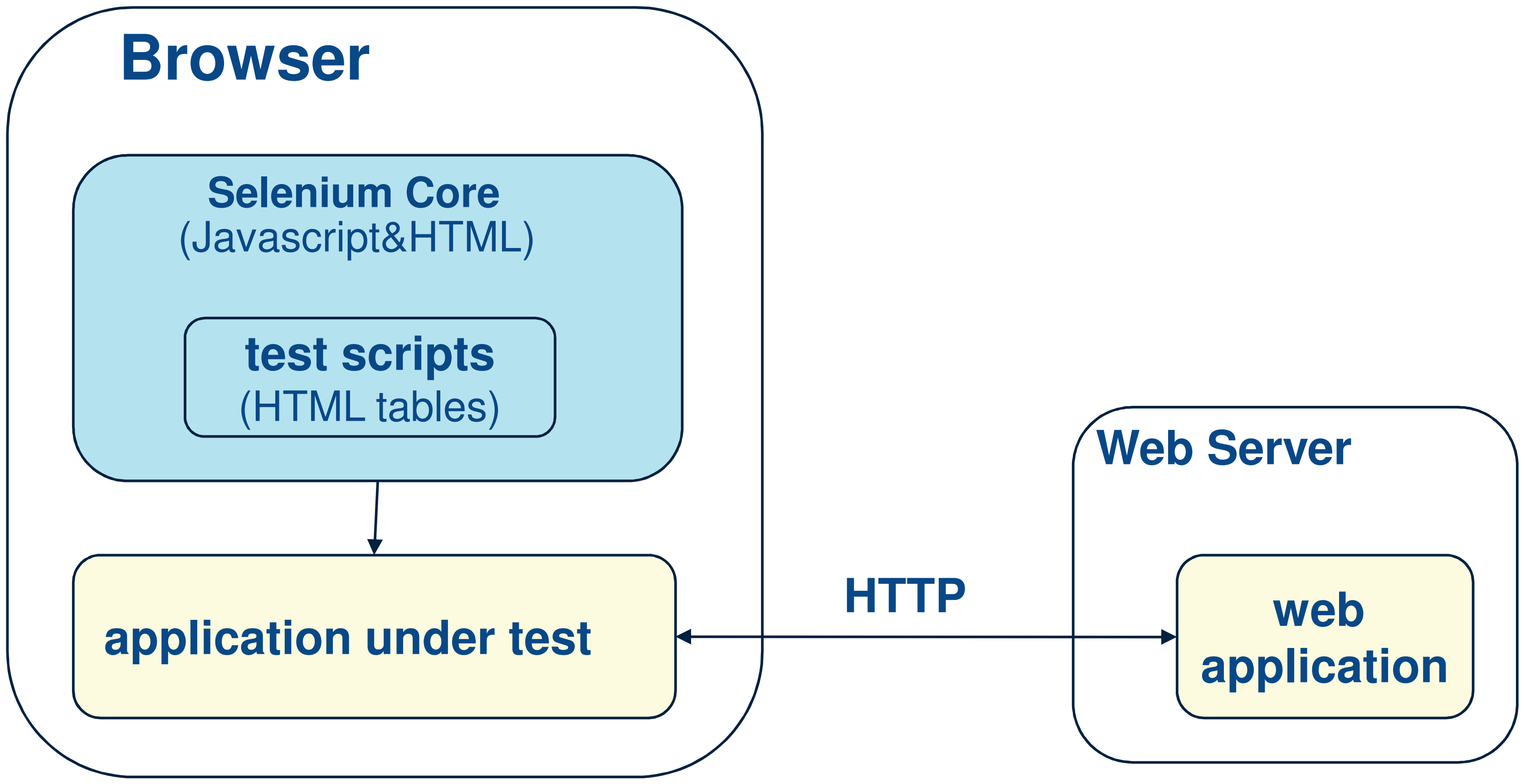
- **The core of Selenium (browser bot) is written in pure DHTML.**
  - **Uses JavaScript and Iframes to embed a test automation engine in the browser.**
    - **is responsible for executing commands received from test scripts**
    - **allows test scripts to run inside supported browsers.**
  - **It can be installed in any web/app server and should work with any JavaScript-enabled browser.**
  - **Controls AUT (application under test) in other frame.**
- **Supported Platforms:**



	Windows	Mac OS X	Linux
Firefox	0.8+, 1.x, 2.x	0.8+, 1.x, 2.x	0.8+, 1.x, 2.x
Mozilla Suite (SeaMonkey)	1.6+, 1.7+	1.6+, 1.7+	1.6+, 1.7+
Internet Explorer	6.0	n/a	n/a
Safari	?	1.3+	n/a
Opera	8	?	?
Camino	n/a	1.0a1	n/a
Konqueror	n/a	n/a	yes

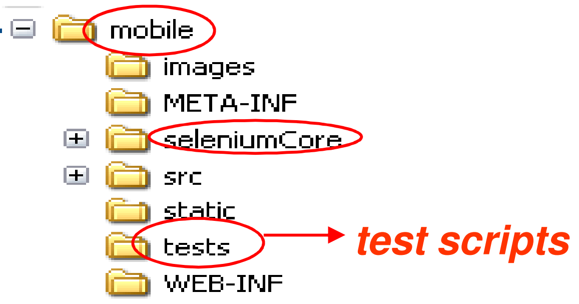
– Selenium website

Selenium Core - Architecture



Selenium Core - Restriction

- **Browser bot uses JavaScripts to emulate user actions, so test runner scripts are deployed on the same server (host & port) as the application under test.**
  - **For using must copy Selenium Core and test scripts (written in HTML) directly into your web application web server, because Selenium core cannot script from one domain to another (because of JavaScript security policies).**
  - **Selenium Core allows the tests to run in any supported browser on the client-side in order to:**
    - test if it works correctly on different browsers and operating systems.
    - perform functionality and acceptance testing.

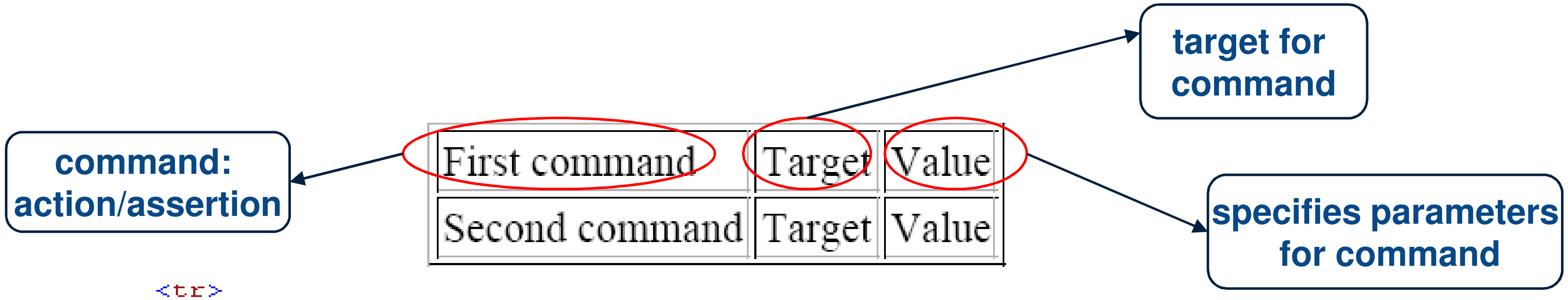


- **Selenium-Core also cannot go from an insecure (http) page to and secure (https) page.**

A test script:

TestCase:web site login

- Written in HTML using a simple table layout

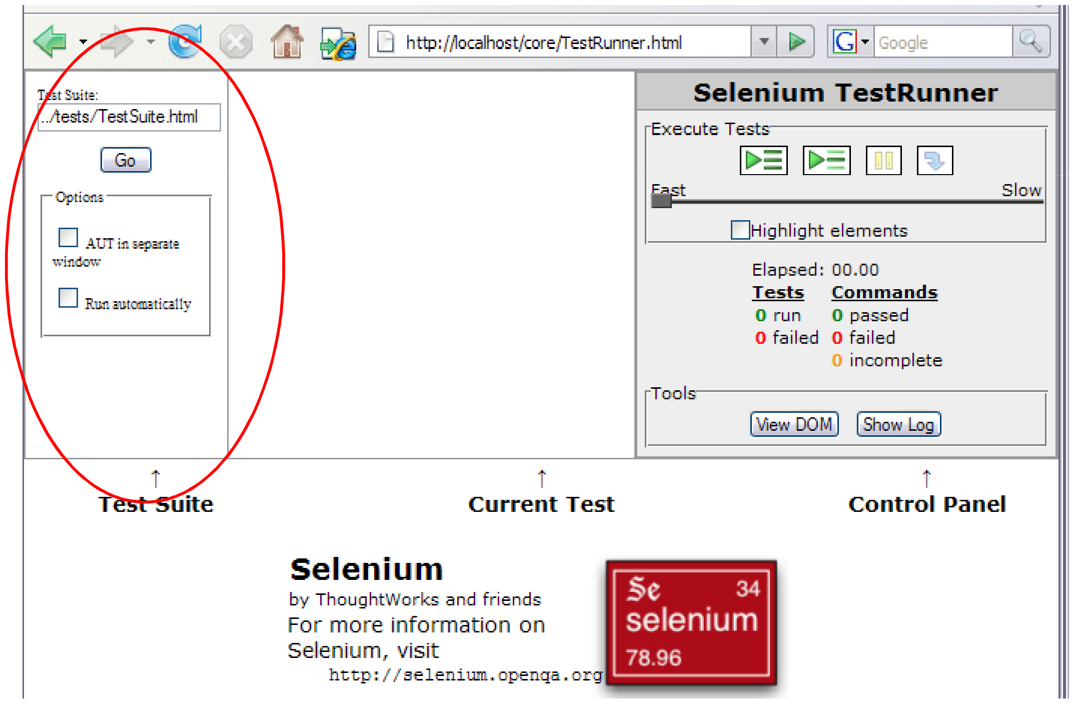


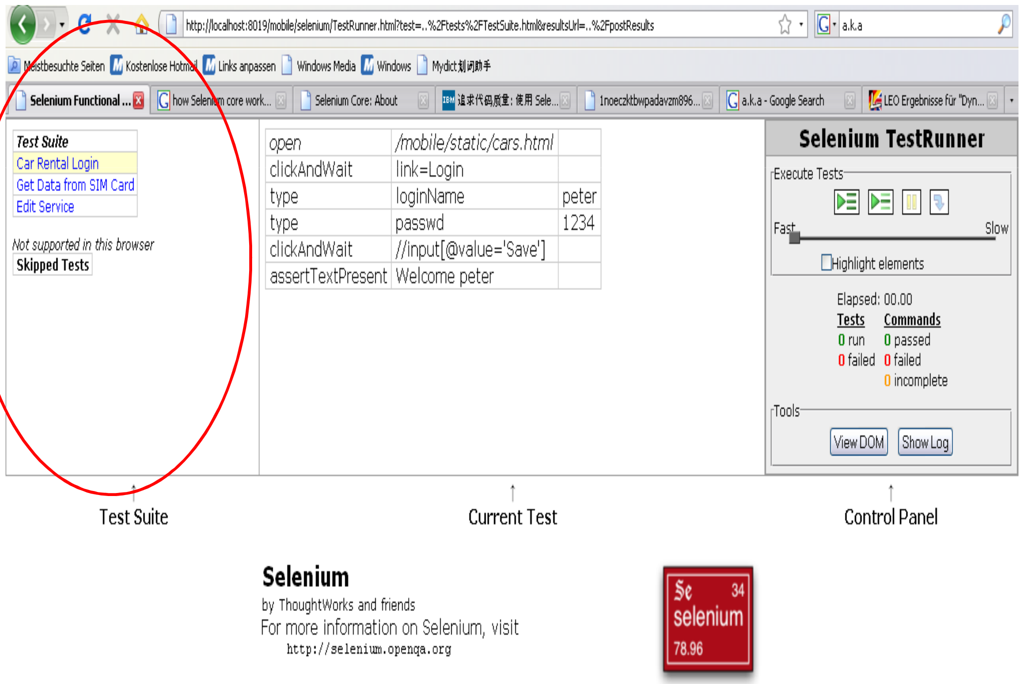
First command	Target	Value
Second command	Target	Value

```
<tr>
    <td>open</td><td>/mobile/static/cars.html</td><td></td>
</tr>

<tr>
    <td>clickAndWait</td><td>link=Login</td><td></td>
</tr>

<tr>
    <td>type</td><td>loginName</td><td>peter</td>
</tr>

<tr>
    <td>type</td><td>passwd</td><td>1234</td>
</tr>

<tr>
    <td>clickAndWait</td><td>//input[@value='Save']</td><td></td>
</tr>

<tr>
    <td>assertTextPresent</td><td>Welcome  peter</td><td></td>
</tr>
```

TestRunner and TestSuites

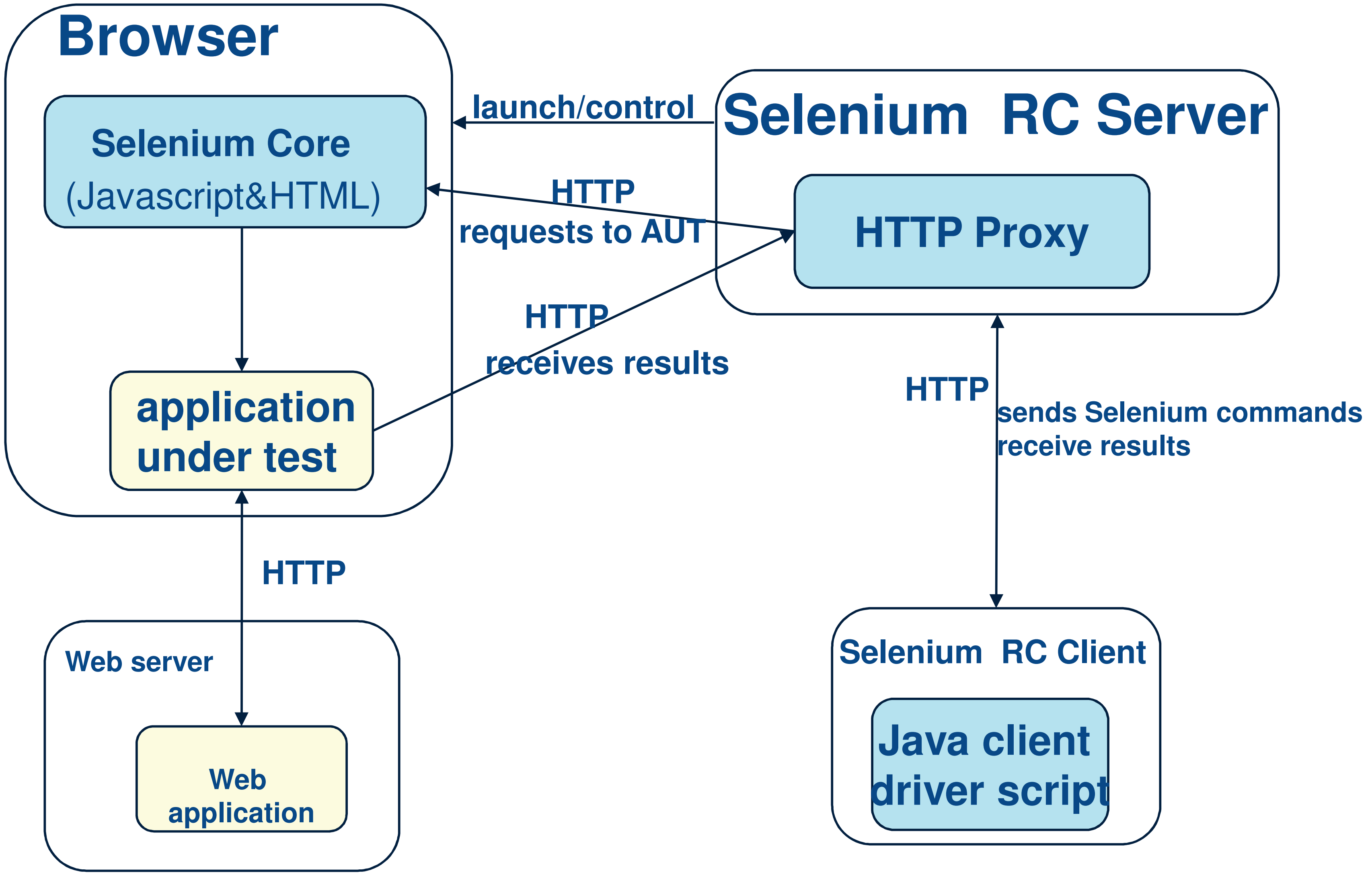
- **TestRunner.html: framework of Selenium Core, allows to select test suites to run in Selenium.**
- **Test suites**
  - **Used to group a set of test cases with similar functionality together so that they can be run in sequential order.**
  - **Use simple HTML tables, the default test suite executed by Selenium is named TestSuite.html.**





open	/mobile/static/cars.html	
clickAndWait	link=Login	
type	loginName	peter
type	passwd	1234
clickAndWait	//input[@value='Save']	
assertTextPresent	Welcome peter	

Selenium Remote Control - Architecture



Driven mode: Selenium Remote Control

Selenium Server

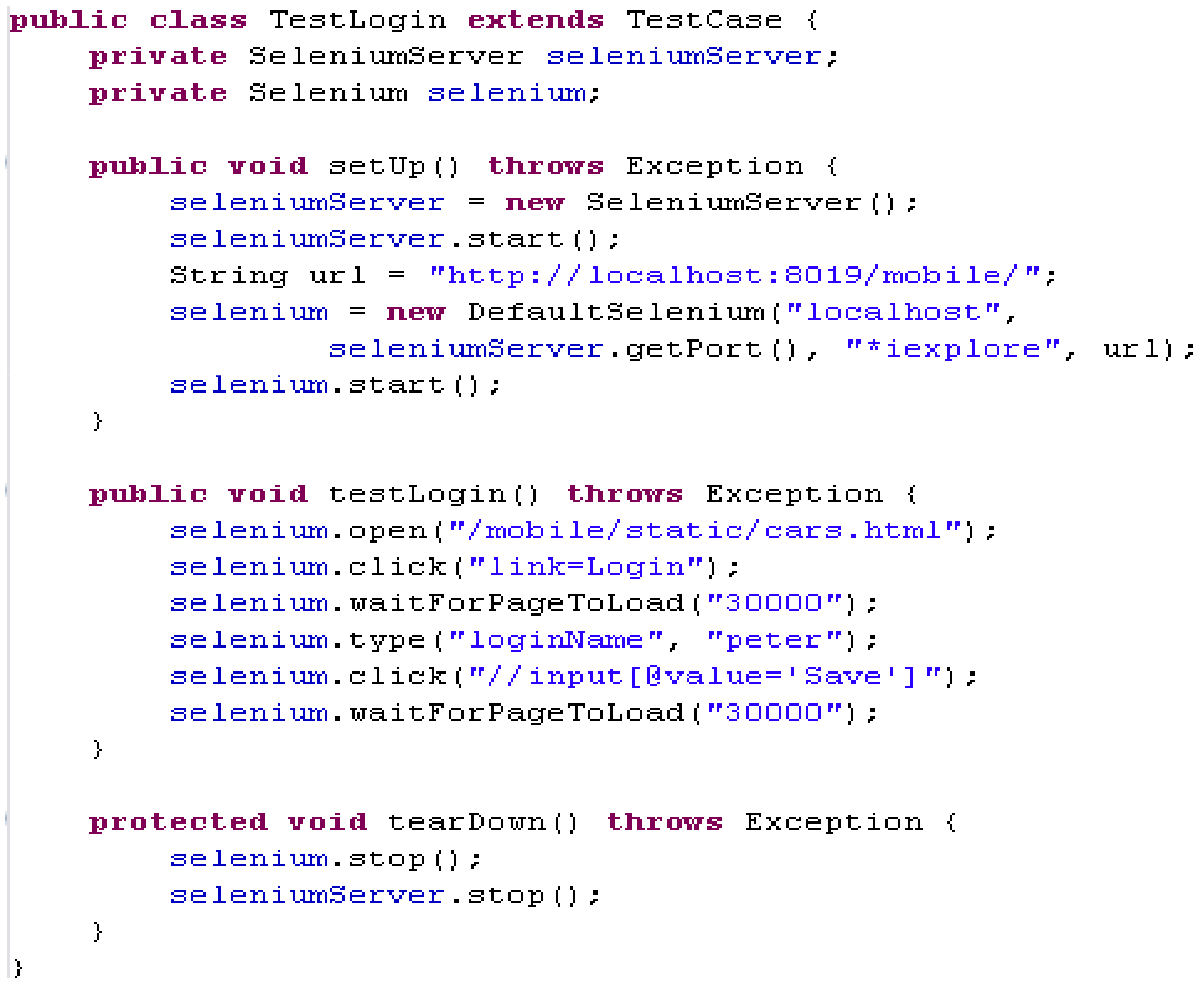
- A java program (built atop of Jetty): is a lightweight process that can be programmatically started and stopped within a test.
- Lets automatically start/stop/control any supported browser to perform the actions of a real user, it communicates directly with the browser using AJAX and works by using Selenium Core.
- Promotes itself as proxy for all requests.
- Launch Selenium core with start page.

Selenium client driver

- Written in one of the supported programming languages (Java, Ruby, Python…)
  - Control Selenium Server: starts browser, loading AUT, test interaction, ends browser session.
- Runs in a separate process outside of the browser.

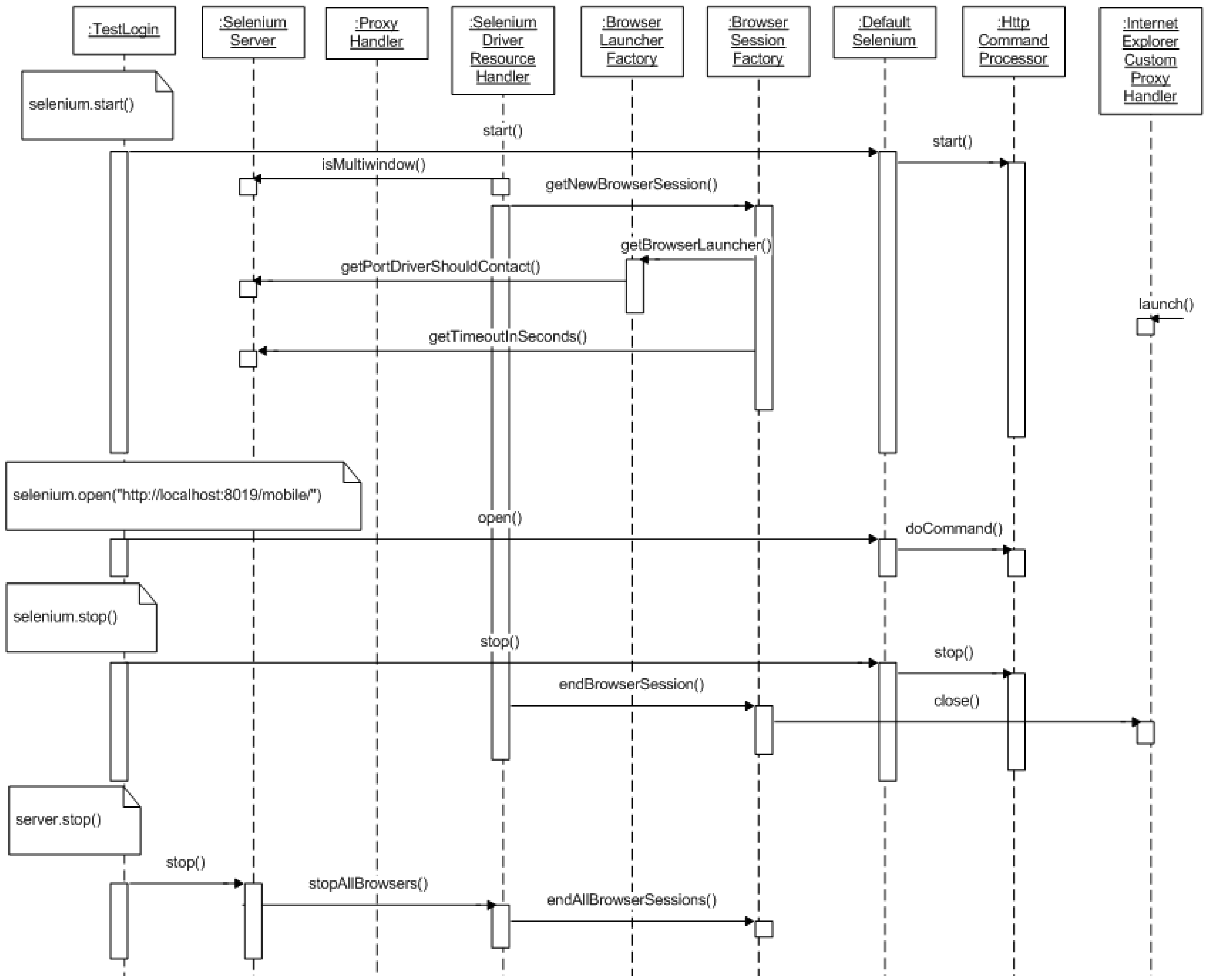
A client driver script

Test Case: web site login with IE

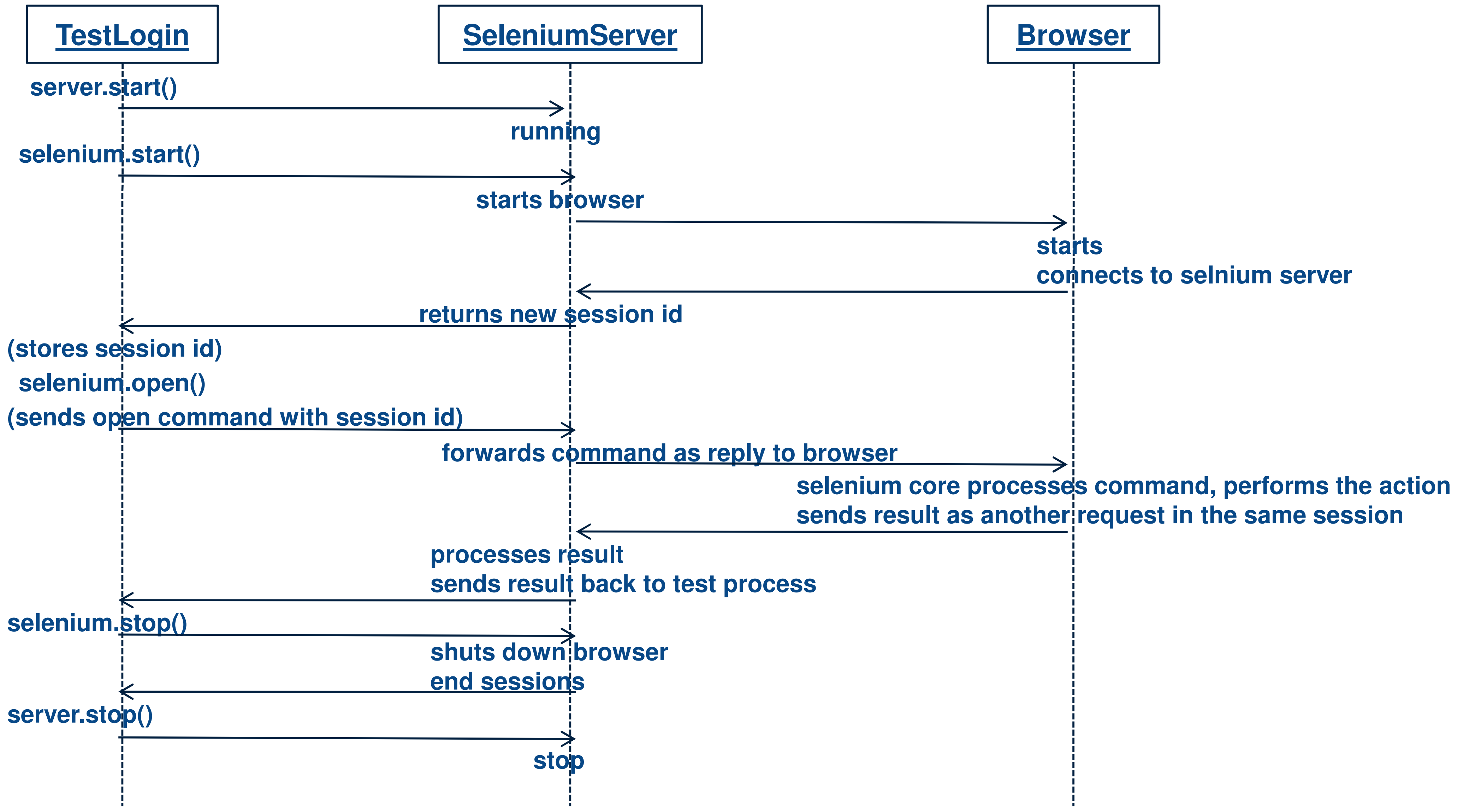


```
public class TestLogin extends TestCase {
    private SeleniumServer seleniumServer;
    private Selenium selenium;

    public void setUp() throws Exception {
        seleniumServer = new SeleniumServer();
        seleniumServer.start();
        String url = "http://localhost:8019/mobile/";
        selenium = new DefaultSelenium("localhost",
                seleniumServer.getPort(), "*iexplore", url);
        selenium.start();
    }

    public void testLogin() throws Exception {
        selenium.open("/mobile/static/cars.html");
        selenium.click("link=Login");
        selenium.waitForPageToLoad("30000");
        selenium.type("loginName", "peter");
        selenium.click("//input[@value='Save']");
        selenium.waitForPageToLoad("30000");
    }

    protected void tearDown() throws Exception {
        selenium.stop();
        seleniumServer.stop();
    }
}
```

Sequence diagram

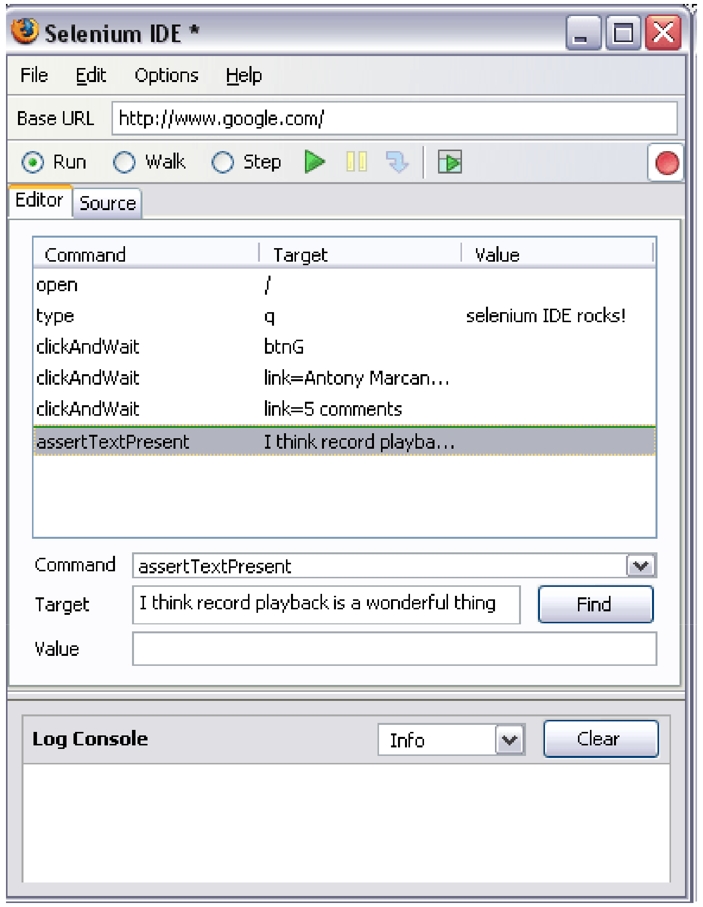


A simple version


TestLogin
SeleniumServer
Browser
server.start()
running
selenium.start()
starts browser
starts
connects to selnium server
returns new session id
(stores session id)
selenium.open()
(sends open command with session id)
forwards command as reply to browser
selenium core processes command, performs the action
sends result as another request in the same session
processes result
sends result back to test process
selenium.stop()
shuts down browser
end sessions
server.stop()
stop

Selenium IDE

- Selenium IDE is an integrated development for Selenium tests.
- Implemented as a Firefox extension.
- Includes Selenium Core.
- It is used to record and playback tests in the browser.
- Import/export various formats (HTML tables is default, like Selenium).
- Possible to edit scripts by hand.



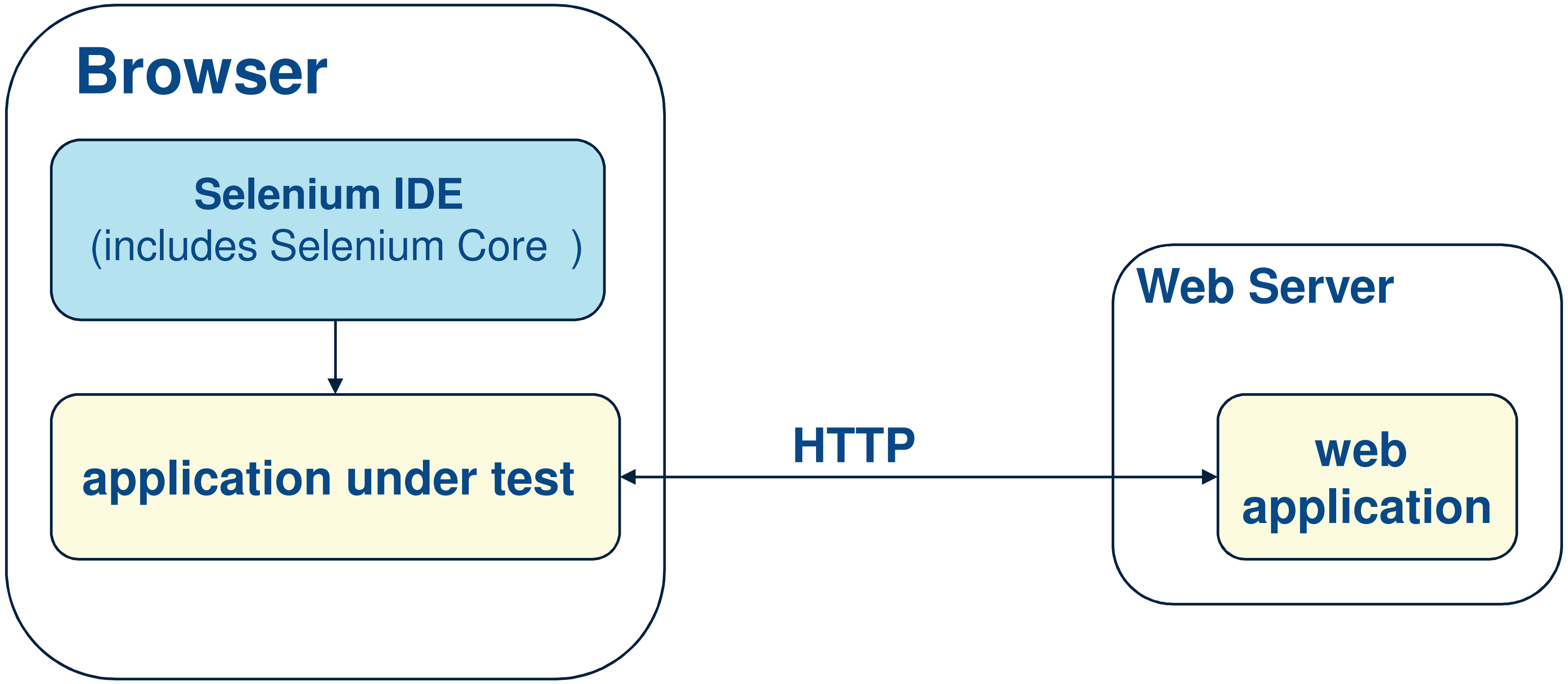
Base URL: http://www.google.com/

Command	Target	Value
open	/	
type	q	selenium IDE rocks!
clickAndWait	btnG	
clickAndWait	link=Antony Marcan...	
clickAndWait	link=5 comments	
assertTextPresent	I think record playba...	

Command: assertTextPresent
Target: I think record playback is a wonderful thing
Value:

– Selenium website

Selenium IDE - Architecture



Selenium commands and locators

- **Commands**
  - Action: simulates the user's interaction with the Web application.
  - Accessors : examine the state of the application and store the results in variables.
  - Assertions: verify the expected outcome of a command.
- **Element Locators: tell Selenium which HTML element a command refers to**
  - "id" attribute, e.g. id=userName
  - "name" attribute, e.g. name=userName
  - dom (JavaScript expression),

    e.g. dom=document.forms['icecreamForm'].flavorDropdown
  - xpath, e.g. xpath=//a[contains(@href,'#id')]/@class
  - link, e.g. link=Click Here To Continue
  - css selector, e.g. css=a[href="#id"]

Agenda

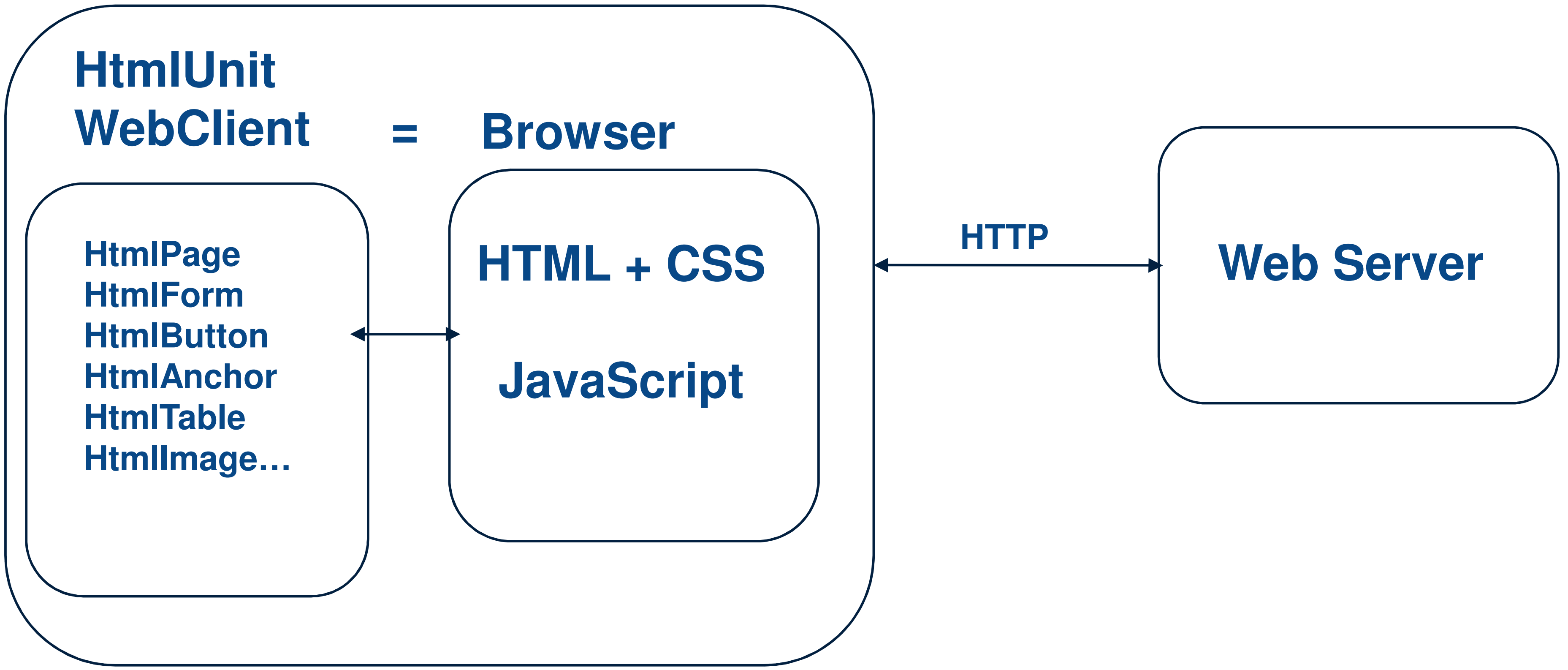
- Selenium
  - Introduction
  - Selenium Core
  - Different ways of using Selenium
    - Selenium Core (FIT style tables)
    - Selenium Remote Control(RC)
    - Selenium DIE
    - Selenium Grid
- Other test tools
  - HtmlUnit
  - jWebUnit
  - JSFUnit
- White-box test (Unit test)
  - Define test objects, design test cases
- Black-box test (funktion test)
  - Define test objects, design test cases

TODO:

HtmlUnit – short description

- Automated black-box testing tool for acceptance and system test.
- It simulates a browser to perform testing of web sites.
- It can be used to:
  - Visit a specified url with the simulated web browser.
  - Test table/frame/form elements from a web page, submit the form by clicking the button.
  - Execute the specified JavaScript within the page...

Architecture



HtmlUnit vs. HttpUnit

- Similar in concept to HttpUnit but is very different in implementation.
  - HttpUnit models the http protocol so you deal with request and response objects.

```
//get application home
WebConversation wc = new WebConversation();
String url = "http://localhost:8019/mobile/";
WebRequest request = new GetMethodWebRequest(url);
WebResponse resp = wc.getResponse(request);
```

  - HtmlUnit models the returned document so that you deal with pages, forms, tables these html elements, it has good JavaScript support and is able to work even with quite complex AJAX libraries.

```
//get application home
WebClient webClient = new WebClient(BrowserVersion.INTERNET_EXPLORER_7_0);
URL url = new URL("http://localhost:8019/mobile/");
HtmlPage homePage = (HtmlPage)webClient.getPage(url);
```

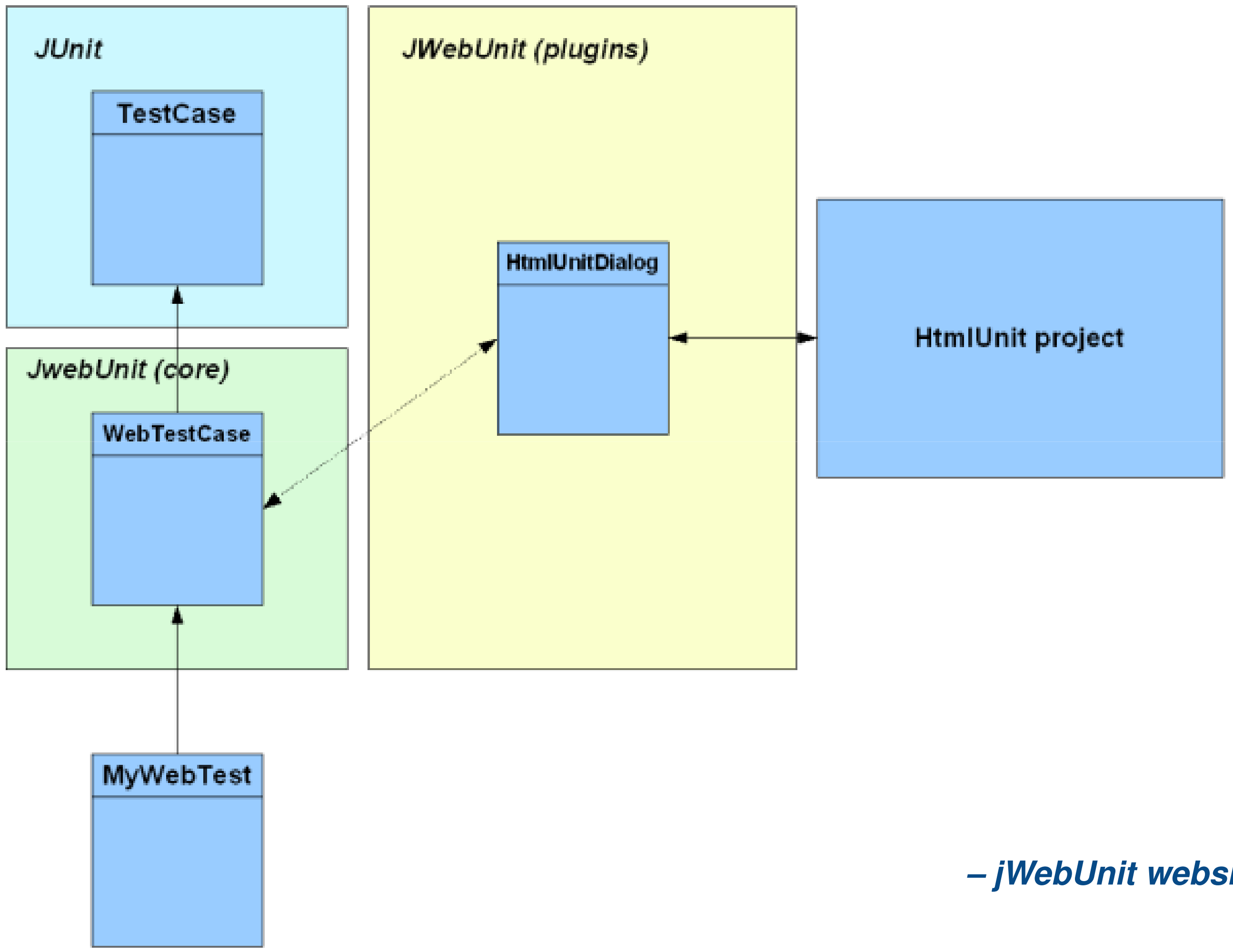
Features

- Support for the HTTP and HTTPS protocols.
- Support for cookies
- Ability to specify whether failing responses from the server should throw exceptions or should be returned as pages of the appropriate type (based on content type)
- Support for submit methods POST and GET (as well as HEAD, DELETE, ...)
- Ability to customize the request headers being sent to the server
- Support for HTML responses
  - Wrapper for HTML pages that provides easy access to all information contained inside them
  - Support for submitting forms
  - Support for clicking links
  - Support for walking the DOM model of the HTML document
- Proxy server support
- Good support for JavaScript

jWebUnit – short description

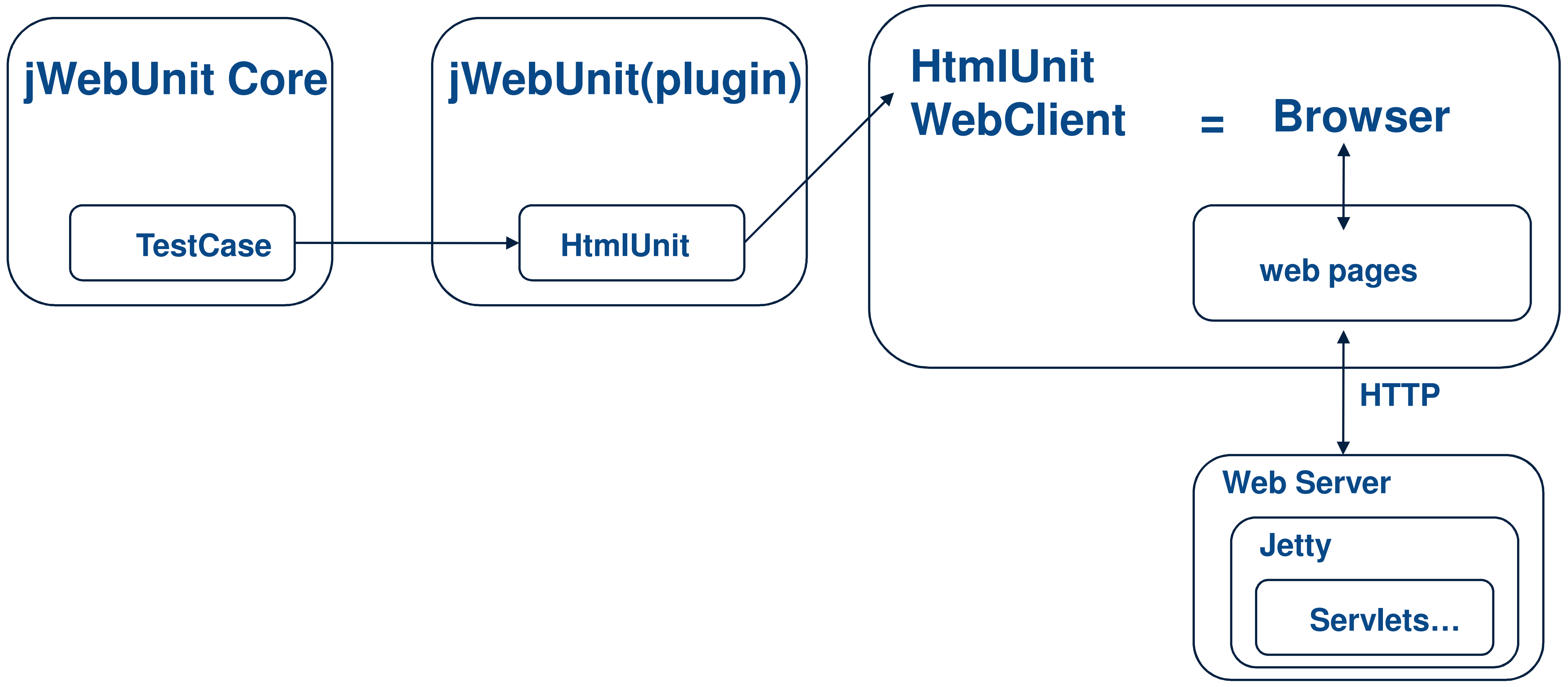
- Automated black-box testing tool that facilitates creation of acceptance tests for web pages.
- It is built on top of HtmlUnit, provides a set of test assertions and application navigation methods which simplify the navigating processes.
- It uses WebClient from HtmlUnit to simulate a browser for testing.
- It can be used to:
  - Navigating via links.
  - Form entry and submission.
  - Validation of table contents…

jWebUnit - Architecture



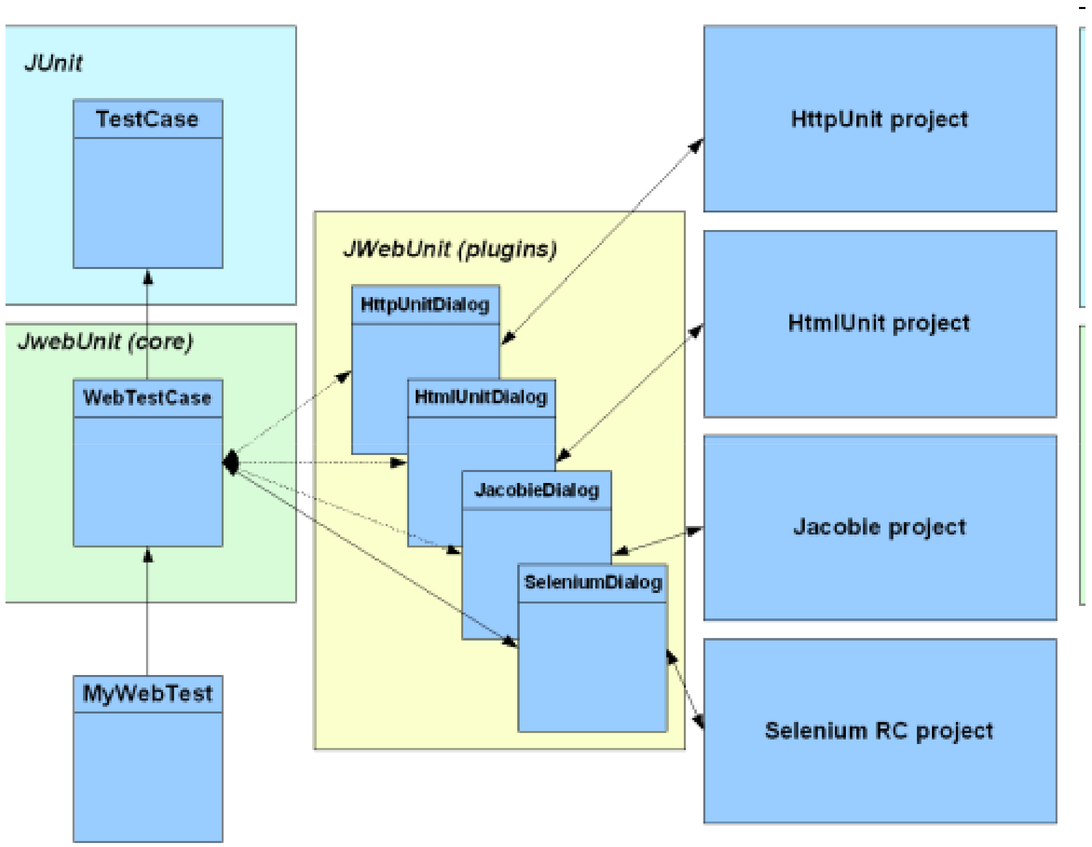
– jWebUnit website

jWebUnit - Architecture



jWebUnit 2.x - preview

- It is the develop version of jWebUnit.



– jWebUnit website

JSFUnit

References

- Selenium-RC, How it works
- Selenium RC: Automated Testing of Modern Web Applications, Thomas Herchenröder, 1&1 Internet AG
- Testen vereinfachen mit Selenium, Michael Kain, Thomas Lieder