



# 借助 WebDriver 执行 自动化 Web 测试

Yugang Fan  
英特尔

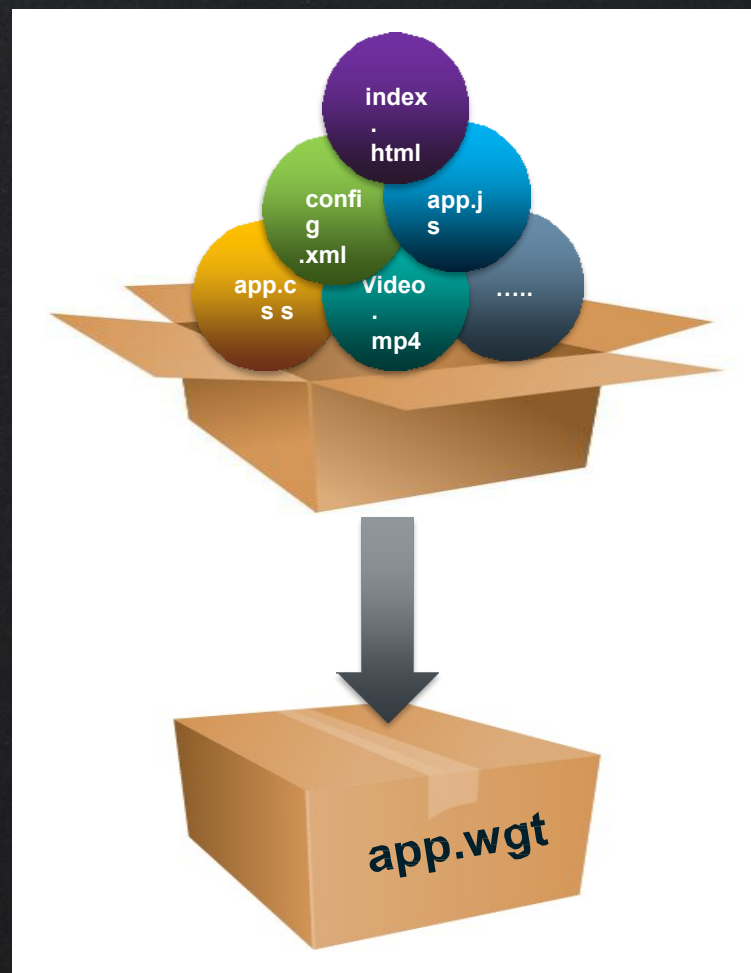


# 议程

- 背景
- 挑战
- WebDriver
- BDD
- 行为驱动测试
- 架构
- 示例
- 基于 WebDriver 的行为驱动测试
- 总结
- 参考资料

# 背景

- Web 应用是指任何在 **web** 浏览器中运行的软件。它通过支持浏览器的编程语言（例如 JavaScript、HTML 和 CSS 三者的结合）创建而成并且依赖于 web 浏览器交付应用
- 对于 web 运行时，**web** 应用是一种使用 **JavaScript**、**HTML** 和 **CSS** 等语言编写的打包应用。
- Crosswalk 是 Tizen 上的默认 web 运行时。





# 背景



# 挑战

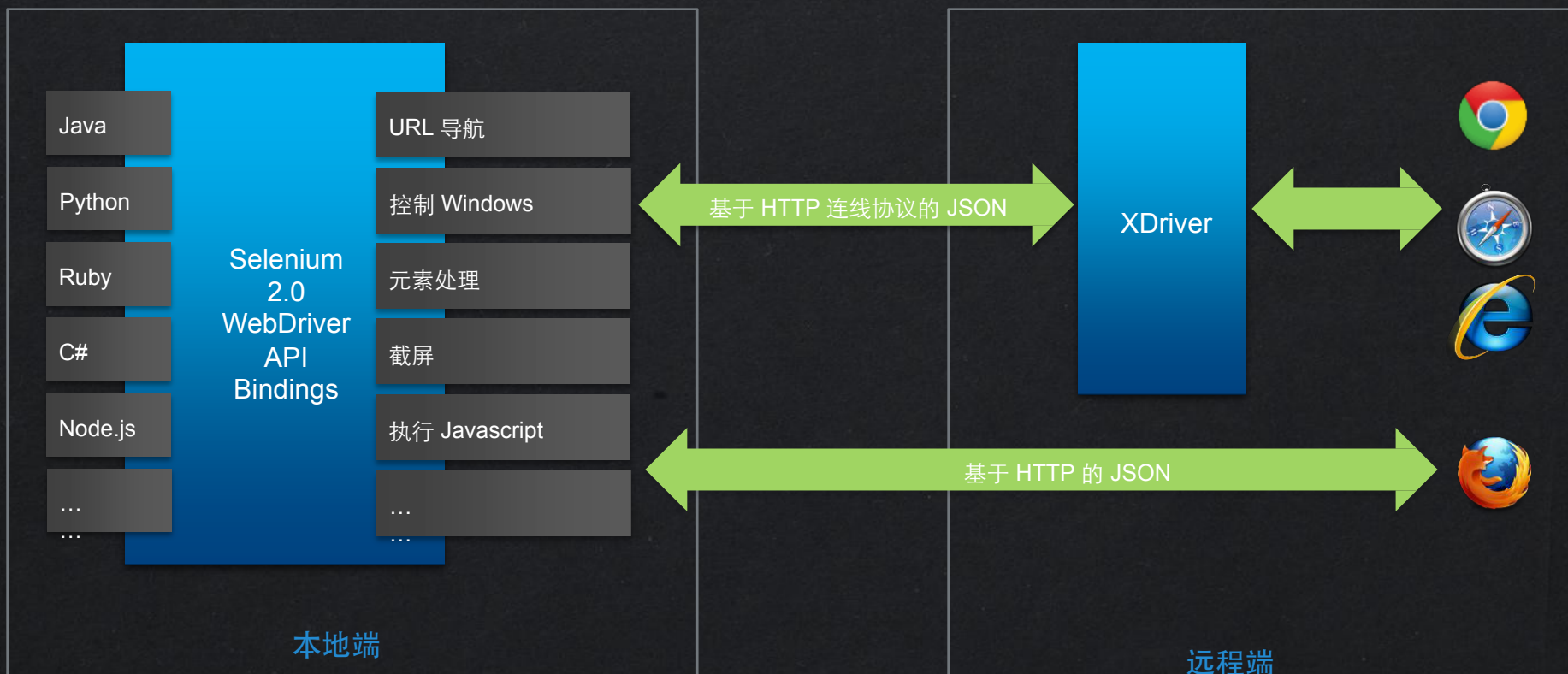
- 难以在 web 应用中应用图像识别或平台本地 UI 自动化解决方案
- Web UI 布局和元素的多样性（multi-formity）需要更为便捷的测试设计和开发方法
- 现有的浏览器测试自动化技术和工具在 web 应用或 web 运行时方面仍存有空白

# WebDriver

- <http://www.w3.org/TR/webdriver/>: WebDriver API 由连线协议和一套接口进行定义，可发现和操作页面上的 **DOM** 元素，并控制单独控制流程中包含浏览器的行为
- 多数常用 WebDriver 规格参数为 **selenium2.0**: Chrome、Firefox、IE 和 Opera .....
- 越来越多的框架采用 WebDriver 实施 Web UI 自动化，例如 Watir WebDriver 和 Appium



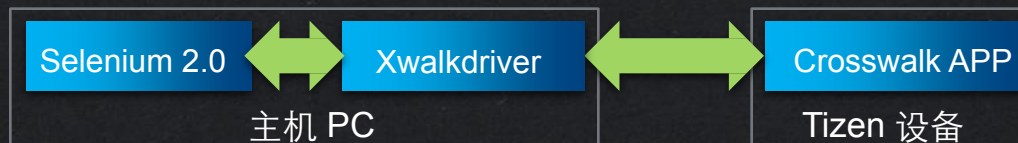
# 浏览器中的 WebDriver 支持



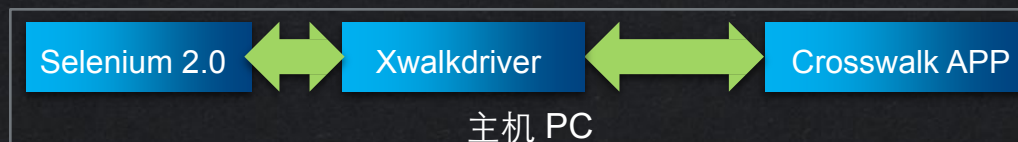
# Web 运行时中的 WebDriver 支持

How to enable Crosswalk  
WebDriver on Tizen?

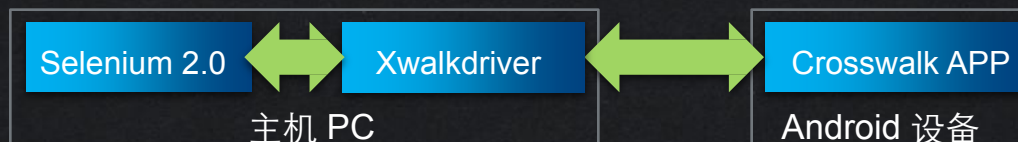
>> <https://crosswalk-project.org/#wiki>



基于 Tizen 的 Crosswalk



基于 Ubuntu 的 Crosswalk



基于 Android 的 Crosswalk



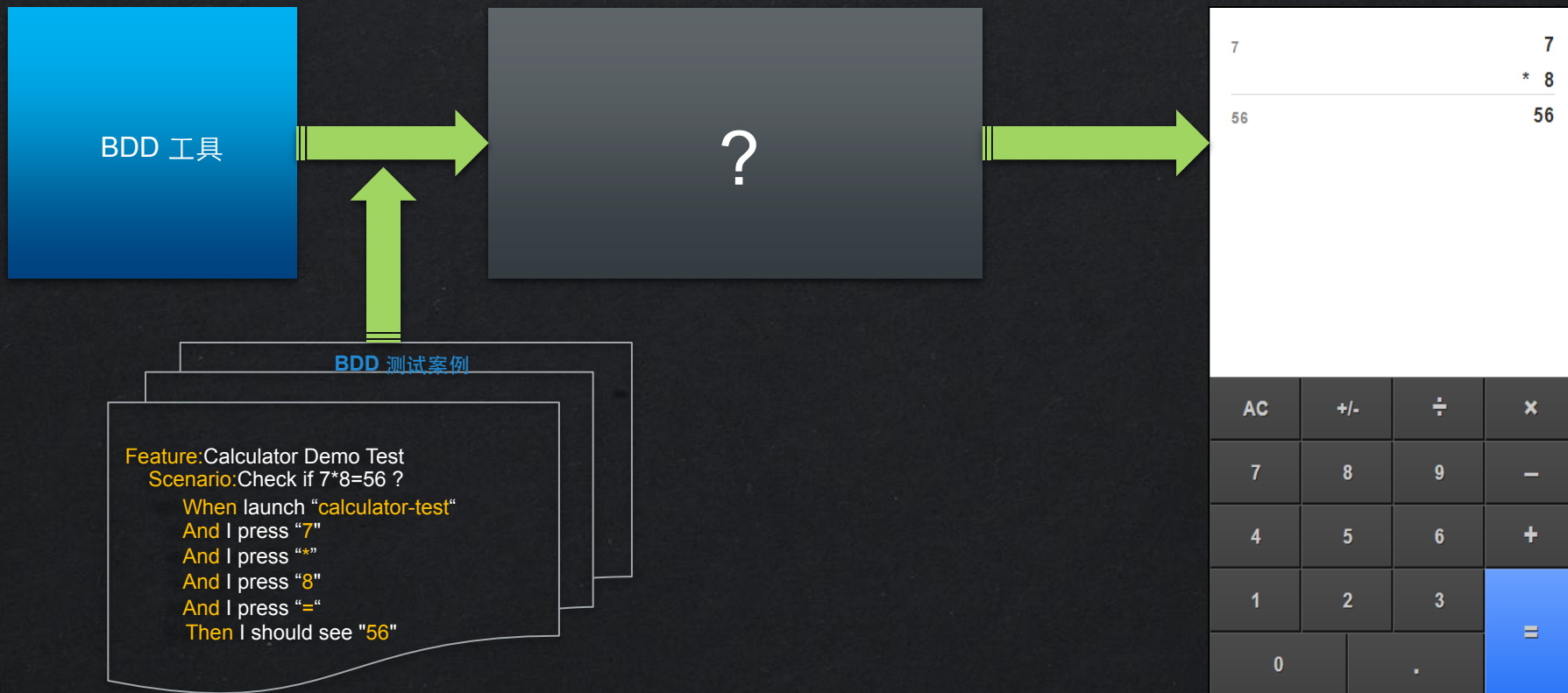
# BDD

## 行为驱动开发

- 灵活的软件开发技术支持开发人员、QA 和非技术人员以及业务人员在软件项目中互相协作。
- 我们的解决方案遵循 BDD 原则，支持写入通用语言风格的用户测试，从而验证 web 应用和 web 平台，提供便捷、高效的方式进行开发、管理和执行测试案例，并且易于与高级别测试框架进行整合。

# 基于 BDD 的测试

应用



# 架构



- “行为” — 采用基于 BDD 的开源 Python 作为默认的 BDD 工具进行应用测试自动化
- “ATIP” (Python 中的应用测试) — 开发一个“行为”约束 (binding) 库，作为应用与“行为”之间的联系，并采用 WebDriver 实施面向 web 应用的具体 BDD 措施



# 示例

## BDD test cases \*.feature file

```
Feature: Web Storage Test
Scenario: storage load page times
When launch "w3c-webstorage-app" And I go to "test.html"
And I press "clear_button" And I reload
Then I should see "viewed this page 1 time(s)." And I reload
And I reload And I reload And I reload
Then I should see "viewed this page 5 time(s)."
```

## ATIP steps

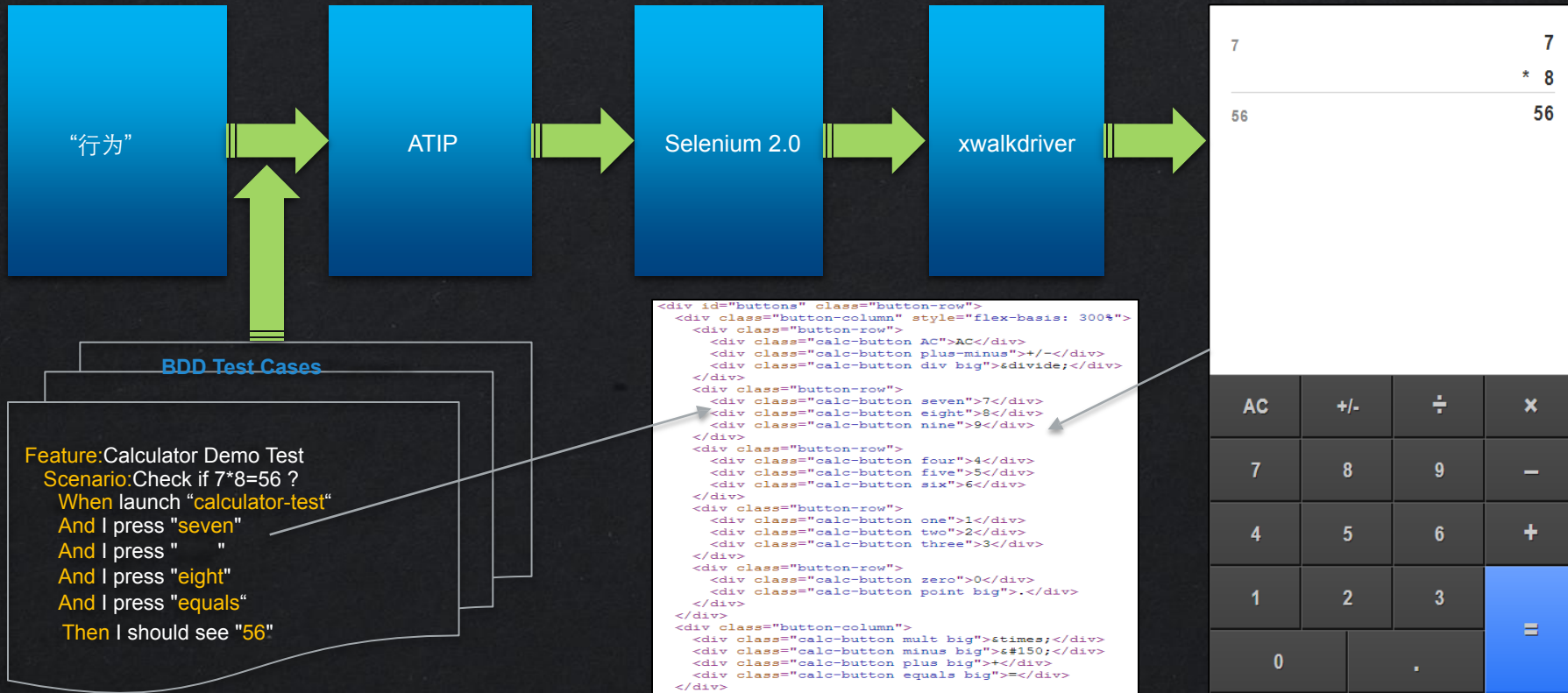
```
@step(u'I go to "{url}"') def
i_visit_url(context, url):
    assert context.app.switch_url(url,
    True)
@step(u'I reload') def
reload(context):
    assert
    context.app.reload()
@step(u'I press "{key}"') def
i_press(context, key):
    assert
    context.app.press_element_by_key(key
    .....
    .....)
```

## ATIP low level implementations

```
def switch_url(self, url,
    with_prefix=True): if
    with_prefix:
    url = urljoin(self.url_prefix, url)
    try:
    self.__driver.get(url)
    except Exception as e:
    print "Failed to visit %s:%s" %
    (url, e) return False
    return True
.....
.....
```

Selenium2.0 WebDriver API

# 基于 WebDriver 的行为驱动测试



# 总结

- 采用 Crosswalk web 测试自动化和实例应用测试自动化：  
<https://github.com/crosswalk-project/crosswalk-test-suite>
- 轻松扩展至任何 web 应用、浏览器、以及带有 WebDriver 支持的 web 运行时



# 后续措施

- 在 ATIP 中支持更多的预定义场景
- ATIP 中的本地和混合应用支持
- 在 ATIP 中提供多种应用支持
- 支持 BDD 措施以及 Python API
- 根据 W3C WebDriver 规格更新进行调整

# 参考资料

- <https://github.com/crosswalk-project/crosswalk-test-suite/wiki/WebDriver-Based-Behavior-Driven-Test>
- <https://github.com/crosswalk-project/crosswalk-test-suite>
- <https://github.com/crosswalk-project/crosswalk-test-suite/tree/master/tools/atip>
- <http://docs.seleniumhq.org/projects/>
- <http://pythonhosted.org/behavior/>



TIZE <sup>TM</sup> N

开发者  
峰会  
2014



上海

fil!:ffi}e.rat9fil!:mle.rat9fil!:ml  
TIZEN7f :M d\$ti( ..t5\$)