

# Multi-Screen on FxOS



Alive Kuo  
alive@mozilla.com

# Outline

- MultiScreen Solutions for Web
  - Demo
  - Architecture
  - Challenge and known issues
  - Future work
- 

How to use second display from the mobile device?



# W3C Presentation API

```
navigator.presentation.requestSession("receiver.html");
```

- 定義網頁投影到第二螢幕的 API
- UA 自行決定 requesting page 跟 presentation page 之間如何溝通
- UA user agent is any software that retrieves, renders and facilitates end user interaction with Web content.

# W3C Presentation API (cont.)

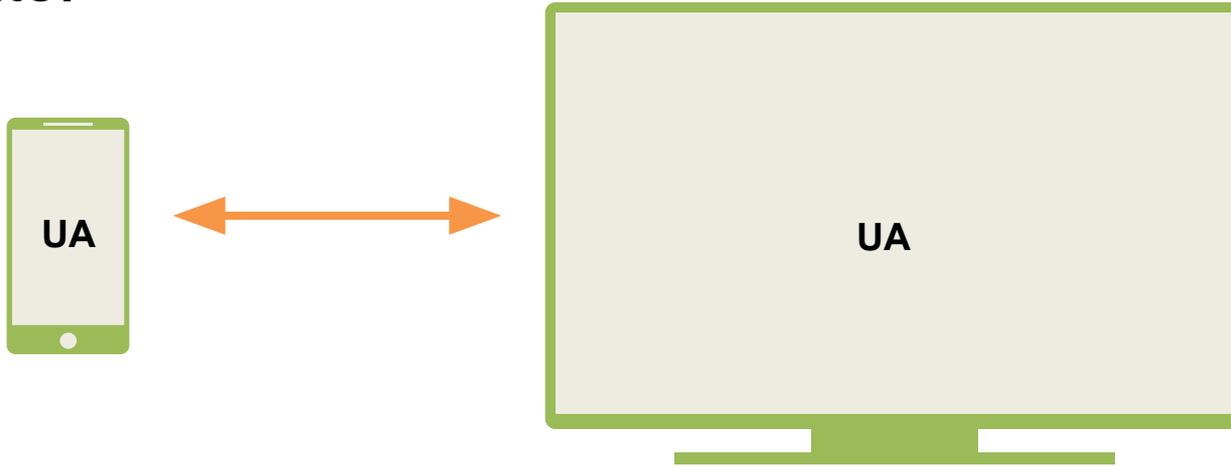
- App should have awareness of the API
  - App should do certain communication to control the remote page
  - 不實際定義訊息傳遞介面，表面上給了很大的空間，實際上可能造成各瀏覽器/各作業系統彼此不相容。
- 

# MultiScreen Modes

- Mirror
  - 所需要的後端技術「相對」簡單
  - 基本上不會牽涉到 web 前端
- Extended/非鏡射
  - 前端「至少」要做 responsive design
  - Funny!

# 2UA Extended Mode

In most implementation, the 2 UA needs to be the same operating system to communicate with each other how to cooperate.



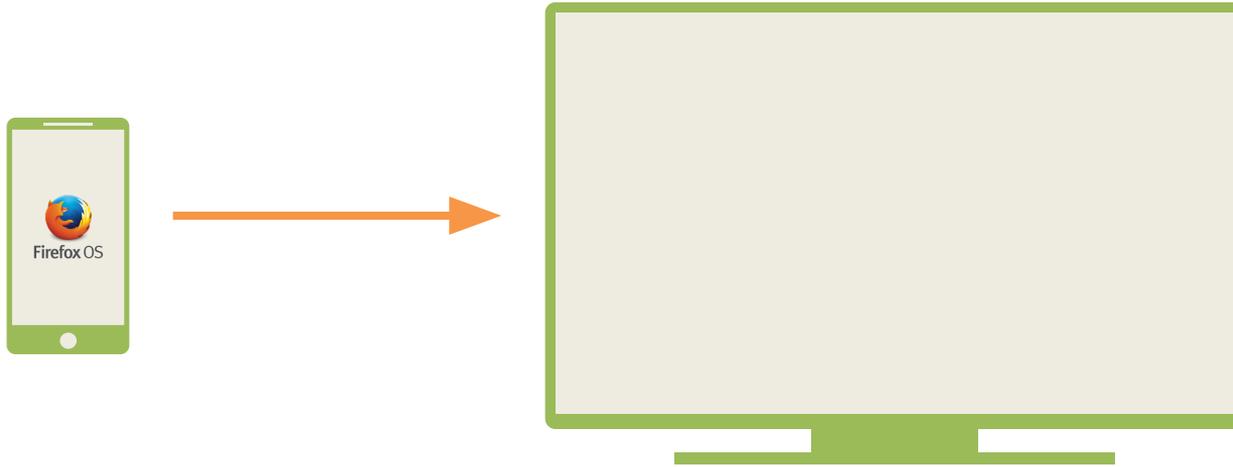
# 1UA Mirror Mode

Otherwise, 1UA for mirror mode only.



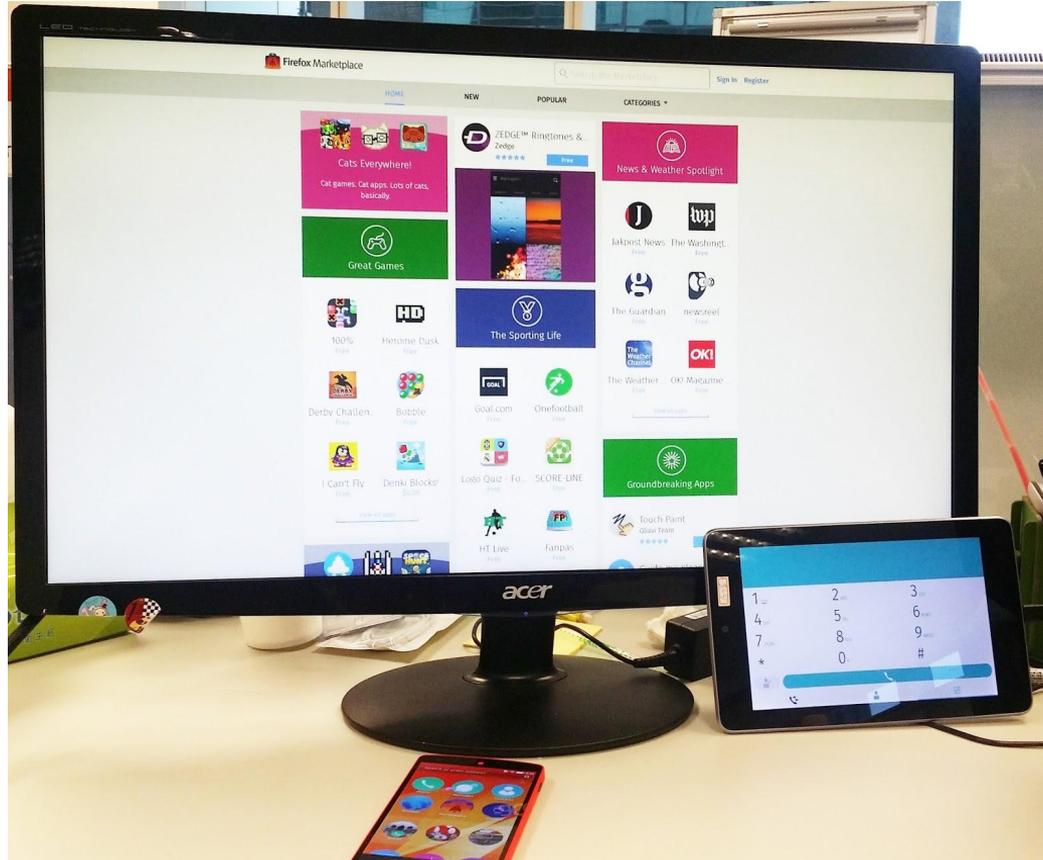
# 1 UA NonMirror Mode

- External displays doesn't run the OS of primary device(1 UA)

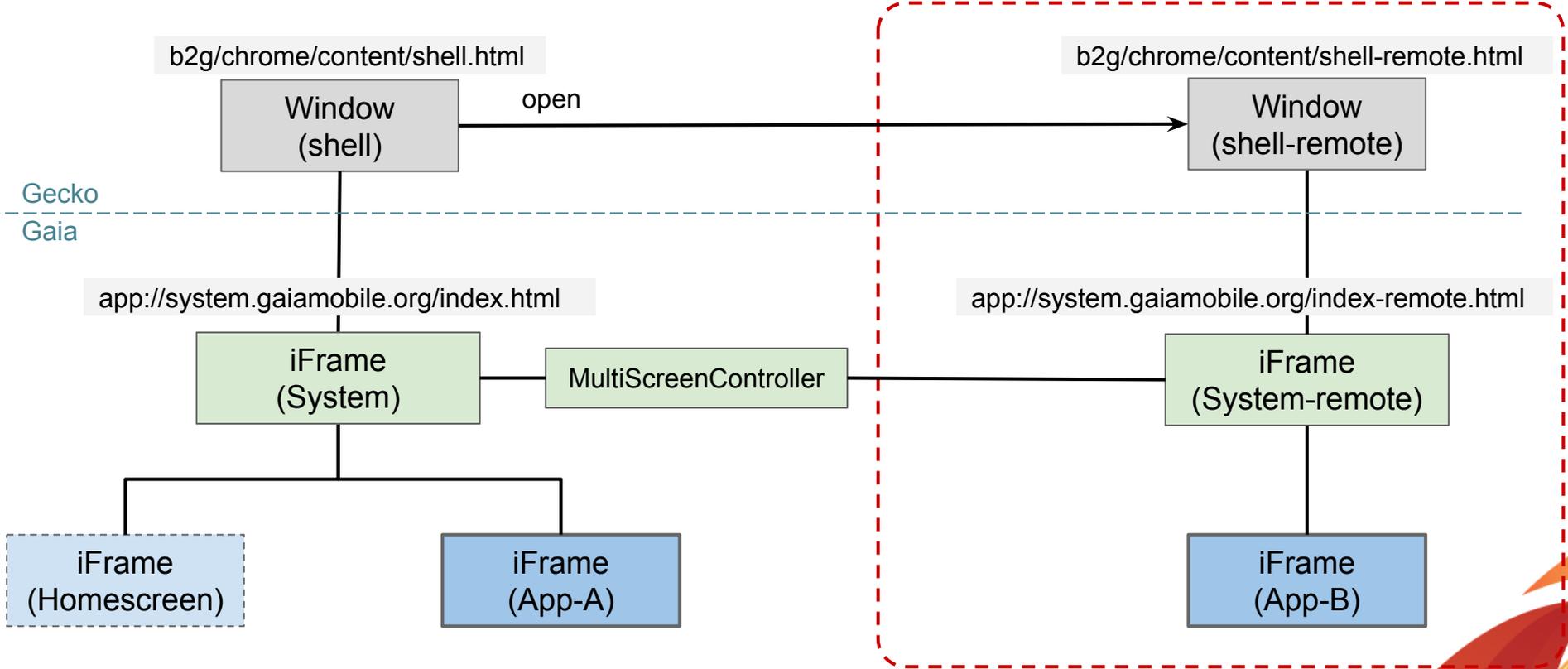




# How does this happen?



# Perspective from DOM tree



# BroadcastChannel

```
// Local system
this.bc = new BroadcastChannel('multiscreen');
this.bc.addEventListener('message', this);
this.bc.postMessage(data);

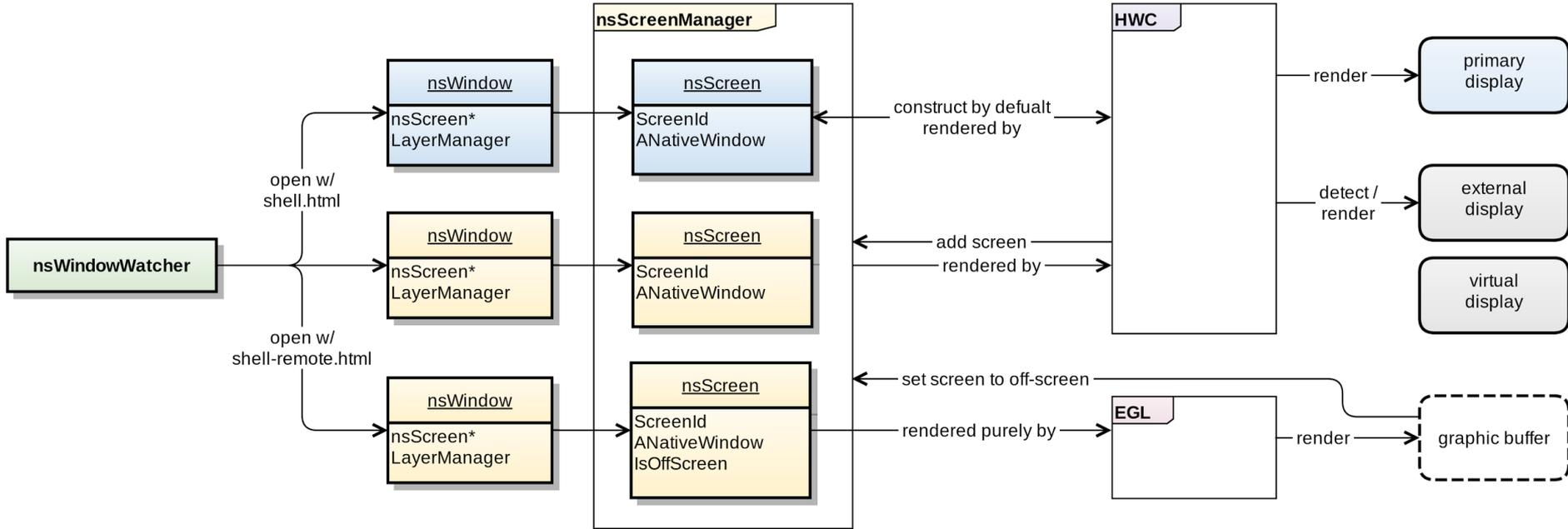
// Remote system
this.bc = new BroadcastChannel('multiscreen');
this.bc.addEventListener('message', this);
this.bc.postMessage('remote-started', this);
```

- **Now Firefox only; alternative: Shared Worker**

[https://developer.mozilla.org/en-US/docs/Web/API/Broadcast\\_Channel\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Broadcast_Channel_API)



# Connected w/ two extra displays



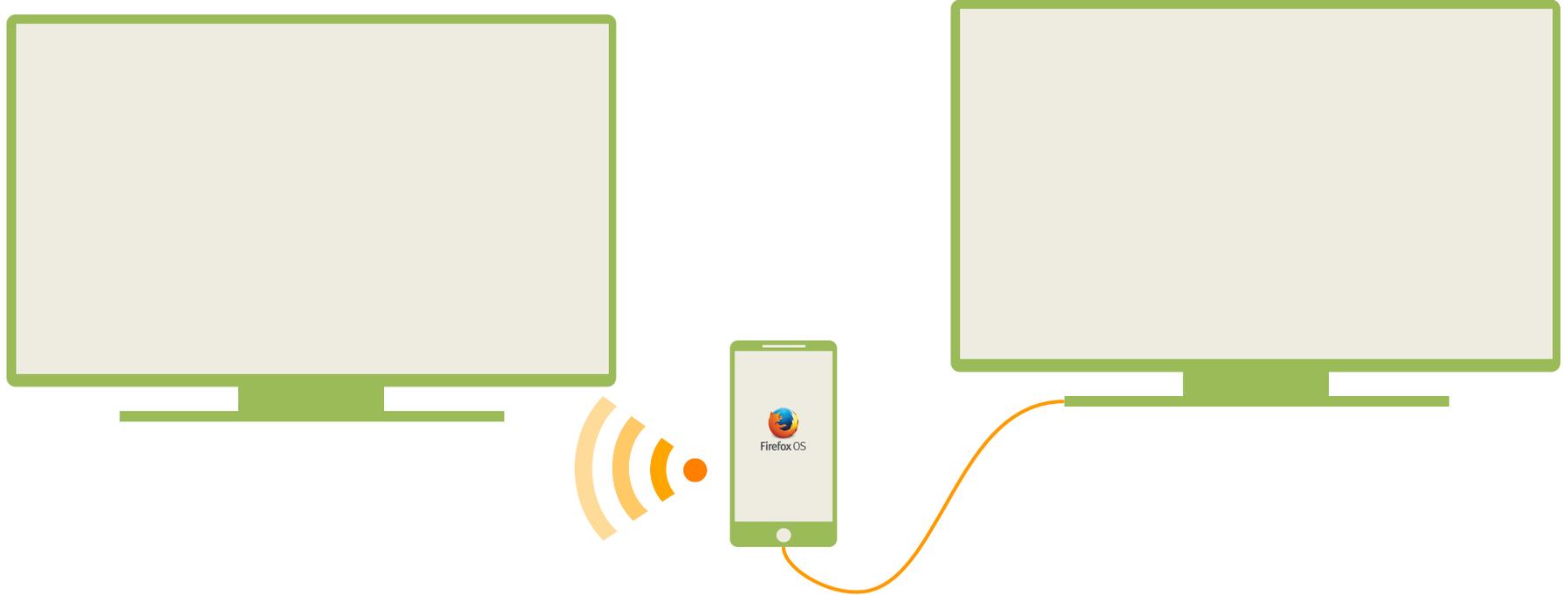
# New MultiScreen Solution

- Independent DOM tree of each screen
    - 避免 Reflow/Restyle 影響另一台螢幕
  - OS GUI 需要同時支援兩種模式
    - 桌面模式不該出現手機專屬的 UI, 使用者情境也比較不同
  - 相對其他 Solution, OS(UA) 扮演了比較重要的角色
    - 對網頁來說, 不需要使用特別的 API, 只需要做好 responsive design
    - 對於想要同時使用兩個螢幕的網頁: 可以透過 same origin 的API 彼此溝通
    - 第二螢幕不需要有另一個 UA, 自然沒有相容問題
- 

# 已知問題與挑戰

- 效能
  - 使用者體驗
  - 架構設計
- 

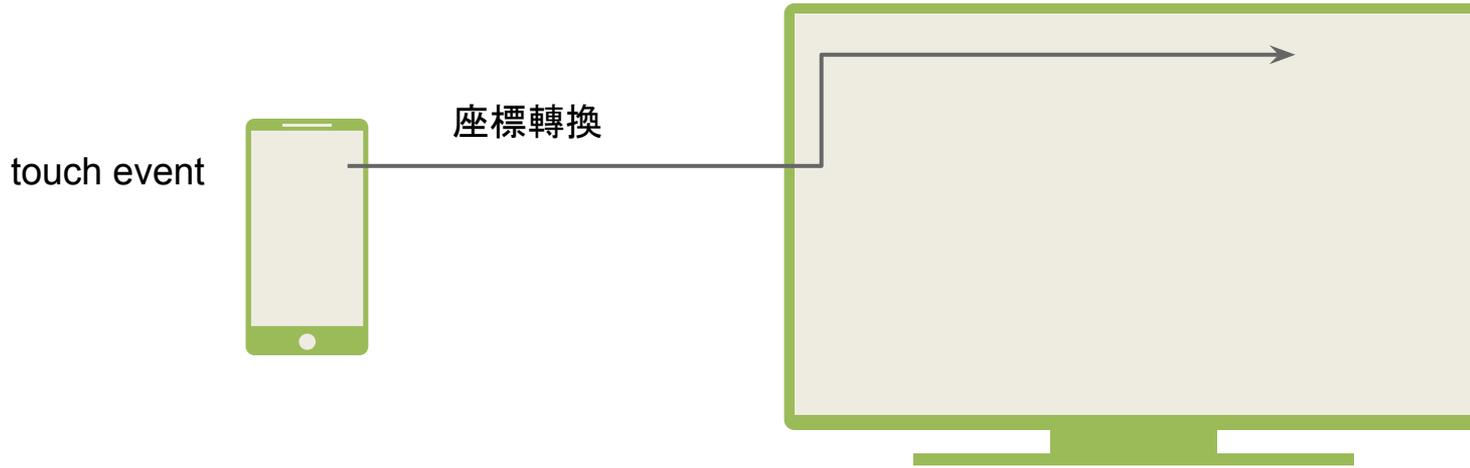
# Performance: Device capability



- Device capability determines how many apps running and how many screens at the same time.

# User control: Virtual touch pad

若touchstart與touchend之間平移距離太短  
則連續送出mousedown, mousemove, mouseup  
模擬 click event



# UX Issues

- 同時操作三個以上螢幕的方法
  - 輸入法管理：軟體鍵盤要繪製在主螢幕或者在個別螢幕上重新繪製？
  - 游標管理：無法跨螢幕移動游標，該如何處理多於一個接收端？
- 

# Prototyping V.S. Maintenance

- 為了 prototyping 快速重寫了一個小型的 desktop system
    - Single window only
    - 當往後功能越來越複雜, desktop system 會有越來越多與 mobile system 交疊的模組
    - 同時維護兩份功能相似的 code base? :(
  
  - 長久來看應該重構現行的 mobile system
- 

# Responsive design... of the operating system

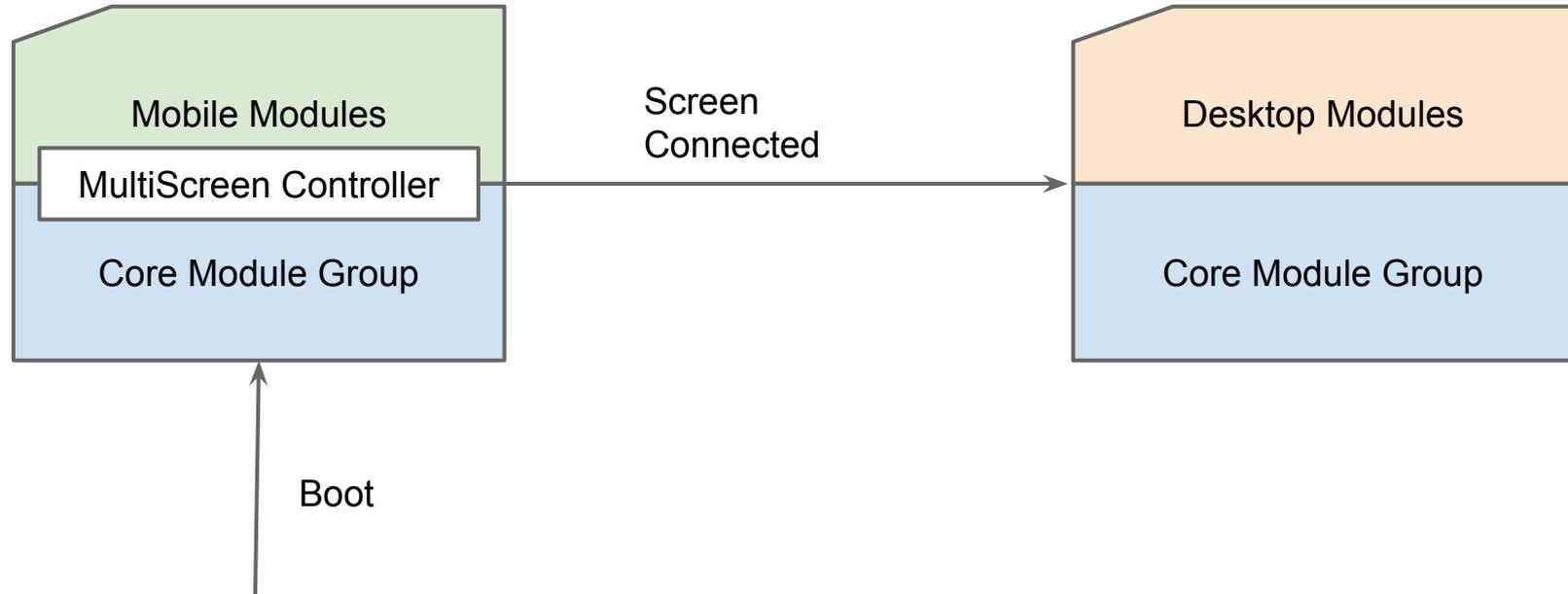
- Mobile OS V.S. Desktop OS
- B2G system is one of the most complex web applications
  - 最初沒有預期到會做  
Tablet/TV/Watch/Desktop/...
  - 牽一髮動全身，重構不易
  - 整合測試很重要！

# Mobile OS V.S. Desktop OS

- Single window management V.S. Multi window management
- 單視窗作業系統 != 簡單
  - Hierarchy, z-index management, Audio channel control, Layout management, Orientation management, Page visibility management, Memory management...
- Not only layout change by media query, but also functionality



# Ideal Architecture



# Thank You

[Gaia] <https://github.com/mozilla-b2g/gaia>

[Contact me] <http://alivedise.logdown.com/>

