

What is node.js ?

A new JS framework like jQuery, ExtJS, Yui.... ?

John
2013/12/15

認識 node.js

- node.js 的角色並不是瀏覽器的腳本語言，所以它並不是像jQuery,ExtJS 的前端工具
- 它是脫離於瀏覽器的運行環境，採用 Google V8 引擎來解釋和執行JavaScript代碼
- 簡單來說它是可拿來做服務器端的應用的，甚至可以拿來當做桌面端應用程序
- node.js只是另一種應用？
 - no，它可是能運用在很多範圍



node.js 特性

- 採用事件驅動(event driven)
- 非同步機制(non blocking)
- 高效處理並發連接數
- 非常適合當作中間層的即時應用系統
- 前後端語言統一

node.js 可以做

- node.js as webserver
- node.js as TCP Server
- node.js as CLI Tools
- node.js as GUI Tools
 - with HTML5

node.js 適合做

- 即時應用系統
 - Chat
 - 公告系統(廣告)
- 利用消息機制跨系統協同合作
- 需長時間運算的應用
- 結合 HTML5 撰寫 desktop app
- 視訊會議 (使用 WebRTC - HTML5 的新規格)
- Game server

node.js

- 不適合拿來做需要高CPU使用量的工作
 - 個人想法: 拿來當作中間層，將工作轉派給適合的系統、語言執行
- 它不是拿來取代原有技術的(儘管可以)

誰在使用 node.js

- google
- amazon
- linkin
- ebay
- Yahoo
- Microsoft
- 淘宝网
- 小米
- 搜狐
- Foxconn
-

DEMO - Hello world

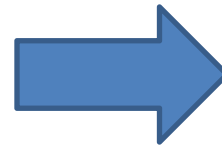
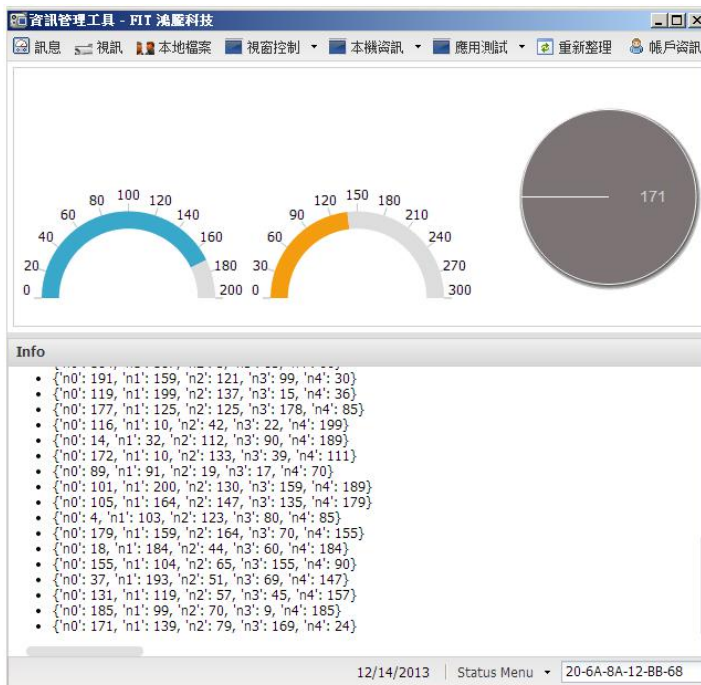
node.js 本身就可當做服務器應用

```
var http = require("http");
```

```
http.createServer(function(request, response) {  
  response.writeHead(200, {"Content-Type":  
    "text/plain"});  
  response.write("Hello World");  
  response.end();  
}).listen(8888);
```


DEMO

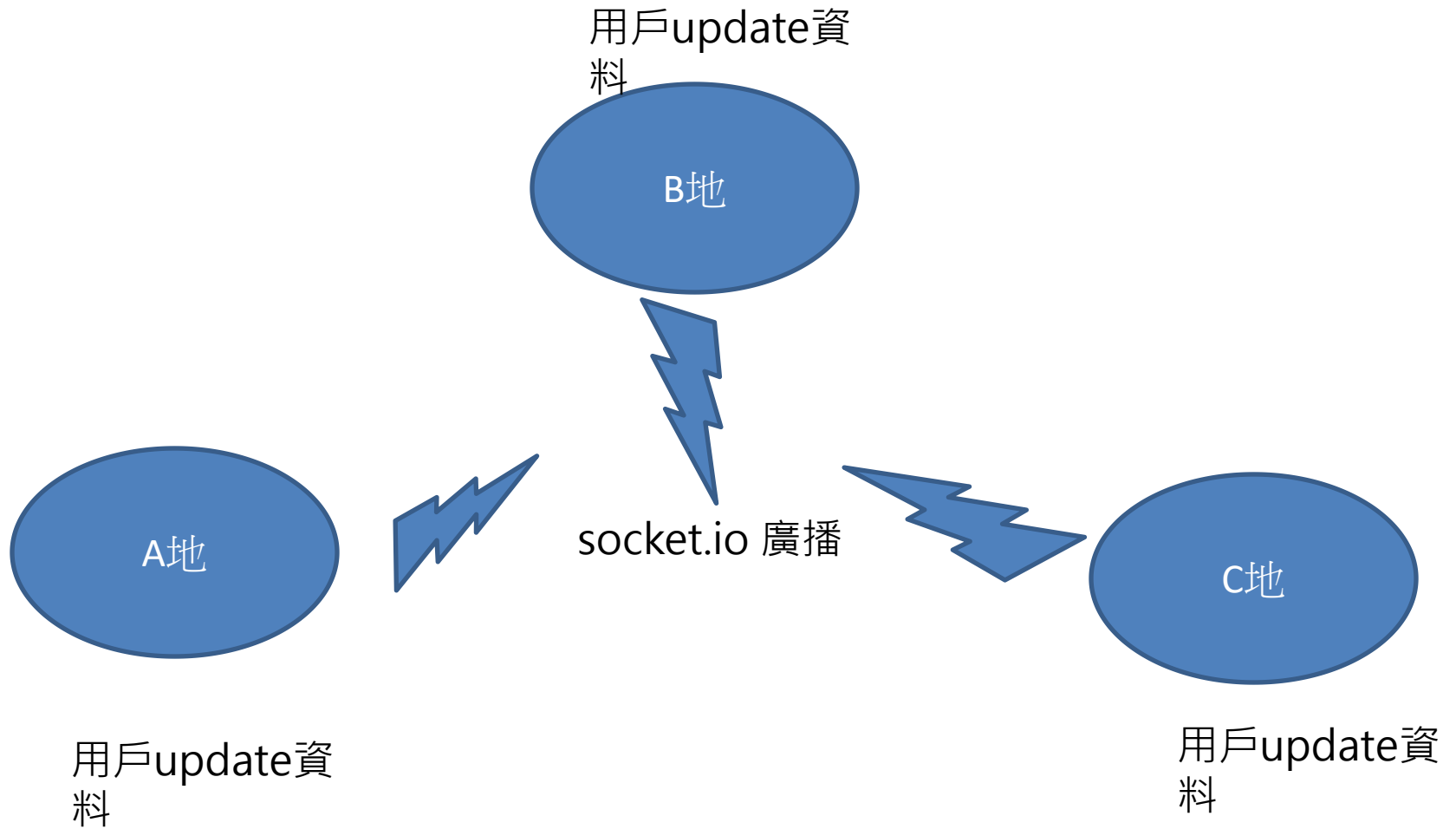
- node.js + express = MVC Web
- node.js + socket.io = realtime service
- node.js + webkit = desktop app



This is desktop application, not web, but use same skills (html, js, css)

可能的應用

資訊同步?



Example : 線上 PPT

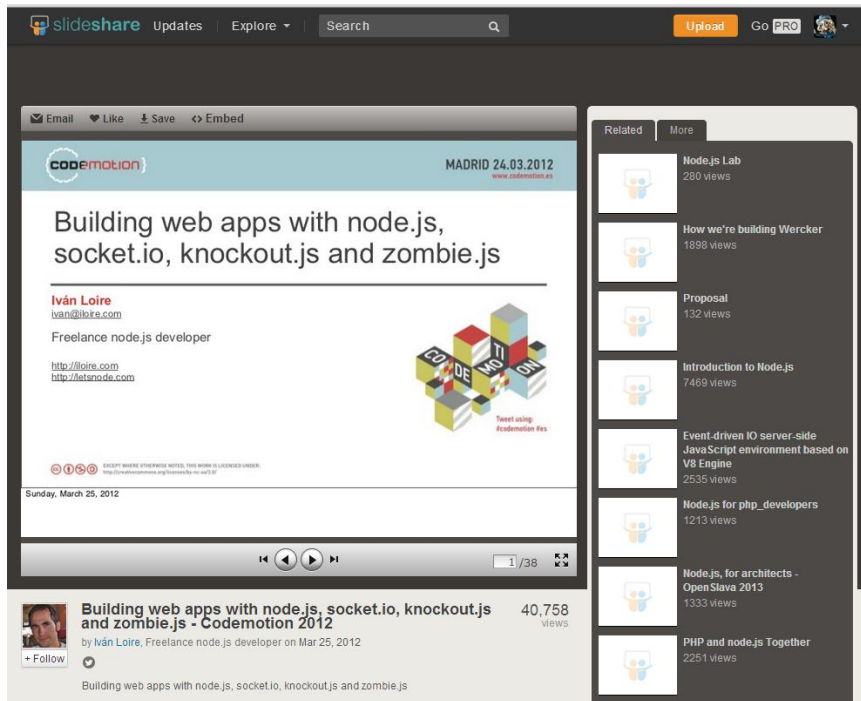
假設要做一個讓用戶上傳office 文件，然後直接在 Web 上觀看的功能
(例如 slideshare, 百度 這類可線上瀏覽ppt 的系統)

線上轉檔需要至少2道程序

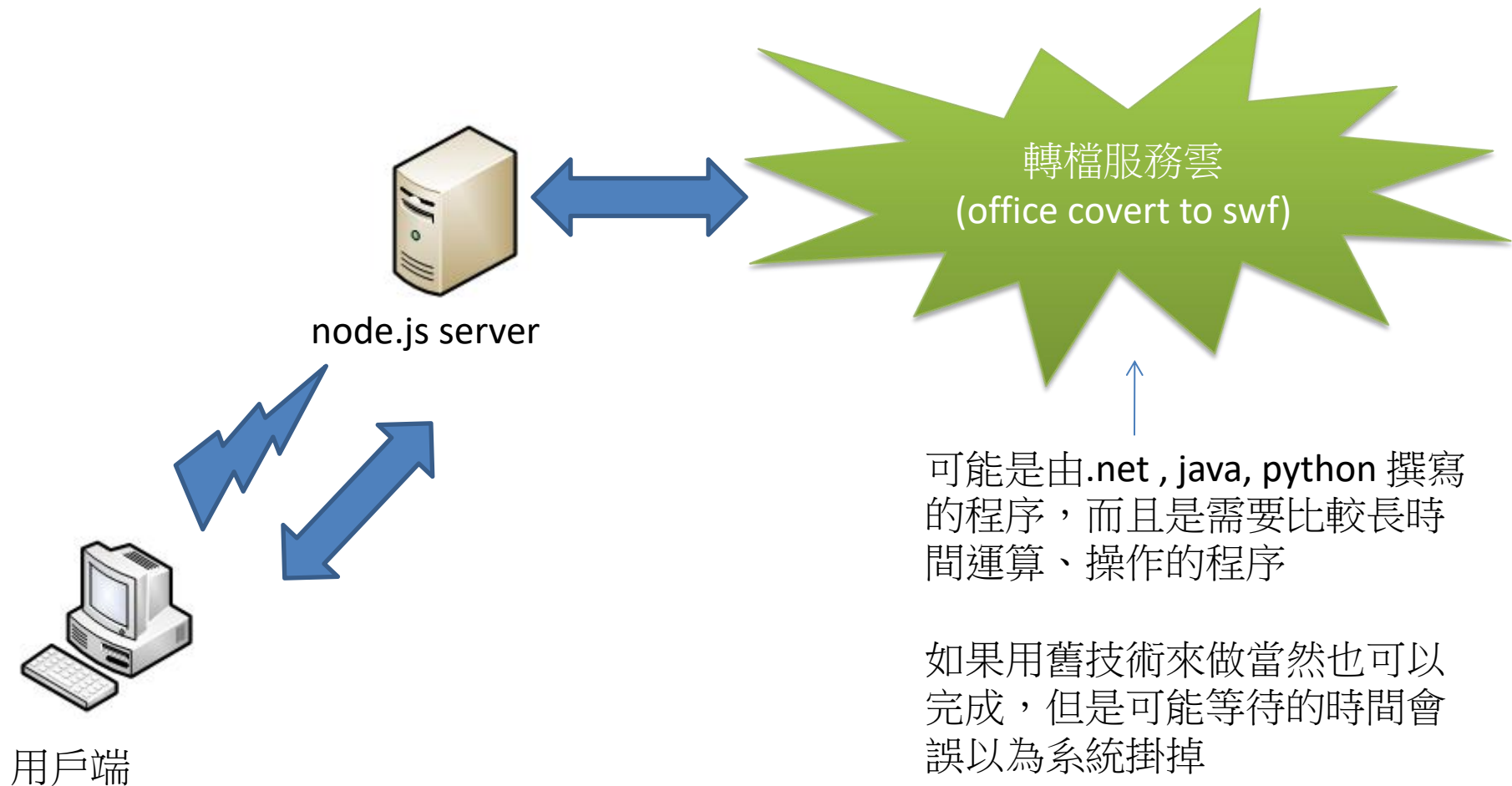
1: 將office 文件轉成 PDF

2: 將PDF 轉成 SWF，視檔案大小，轉檔可能需要長時間進行

3: 轉檔完成後，用戶端看到類似以下結果



Example : 線上 PPT



轉檔 demo code

```
def convert_to_swf( formatType = 32):
    print os.path.dirname(os.path.abspath(__file__))
    print 'starting convert ppt to pdf..'
    filename = 'redmine.ppt'
    powerpoint = win32com.client.Dispatch("Powerpoint.Application")
    powerpoint.Visible = 1
    newname = os.path.splitext(filename)[0] + ".pdf"
    current_path = os.path.dirname(os.path.abspath(__file__))
    deck = powerpoint.Presentations.Open(current_path + '\\ ' + filename)
    deck.SaveAs(current_path + '\\ ' + newname, formatType)
    deck.Close()

    file_path = current_path + '\\ ' + newname #u'%s/%s' % (settings.PROJECT_PATH,unicode(ori
    output_path = os.path.splitext(filename)[0] + '.swf'

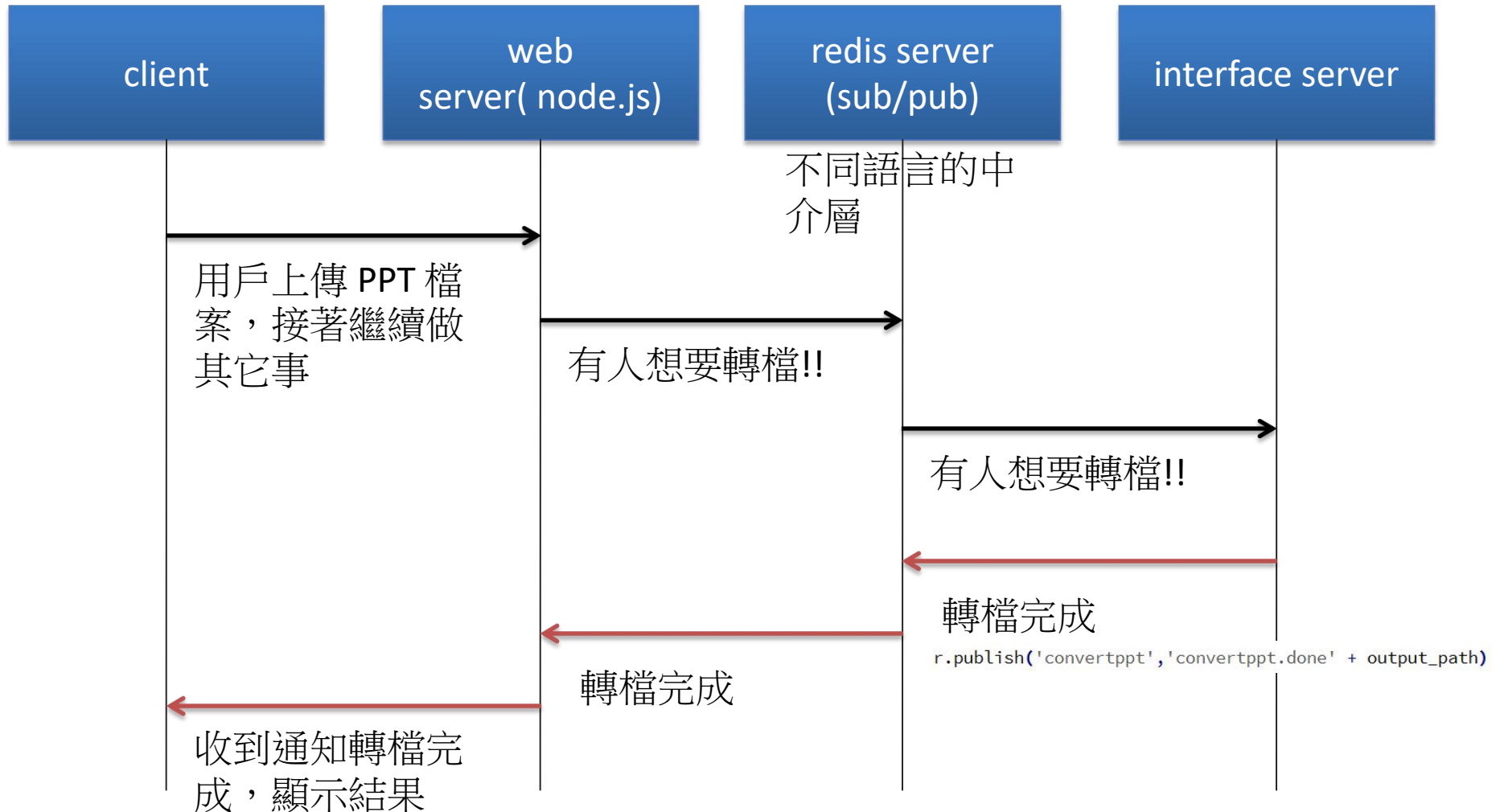
    try:
        print 'start convert to swf'
        exe_name = 'e:/SWFTools/pdf2swf.exe'
        code = subprocess.Popen([exe_name, file_path, '-o', output_path], stdout=subprocess.PIPE)

        #檢查檔案是否已建立完成
        is_exist = False
        while not is_exist :
            is_exist = os.path.exists(output_path)
            print 'cheking...'
        print 'done'

        r.publish('convertppt','convertppt.done' + output_path)

    except Exception:
        pass
```

Example : 線上 PPT

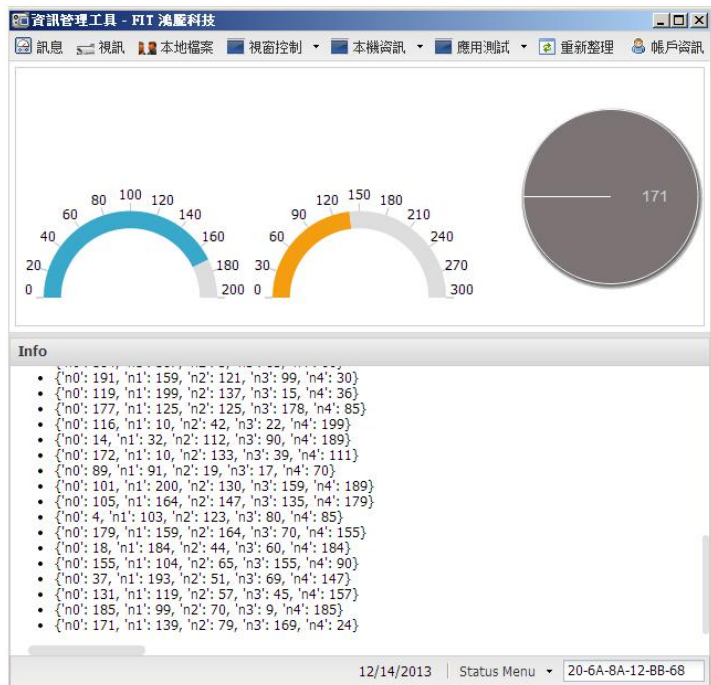


node.js + HTML5 – WebRTC = 視訊會議

WebRTC 是 HTML5 制定的新規格，目前 google,firefox,opera 都已支援



可能的應用



node.js 可以做 desktop 的應用程式!!

這個想像空間就滿大的!為什麼?

因為大家一直為IE所苦，討論的結果裝chrome似乎也不是解決之道...

而node.js + webkit 的應用等同於基於chromium
打造我們自己的瀏覽器!!

個人想法是，利用這樣一個桌面程序，除了存取local資源，儘可能保持 desktop app 的簡單性、不要太頻煩的更新(因為桌面程序有個麻煩點就是程式更新的處理)，大多數內容還是直接連到WEB網頁。

利用這樣又是desktop app 又是browser 的特性，新開發的網站就可以由這個自行開發的app 來開啟，而舊的系統，當然就可以由程式控制用IE來開啟。

持續更新....

node.js這樣的技術可以應用的面我覺得相當的多，希望大家不要侷限在某一些點上，不要用目前用不到的理由就不去認識及思考!

ex: node.js 搭配行動裝置，Web, App，視訊、工廠生產數據即時看板、家電操控、搭配WebRTC、Raspberry 遠端巡場都是可能的應用!!
(據我所知，已有node.js應用在工廠的、銀行的.....等系統)

此外請大家了解這不是要推翻原本就有的機制、程式，而是可以利用這樣簡單的機制來達成系統與系統間的協同運作，希望大家儘可能的多方面思考這個技術可以使用的狀況!