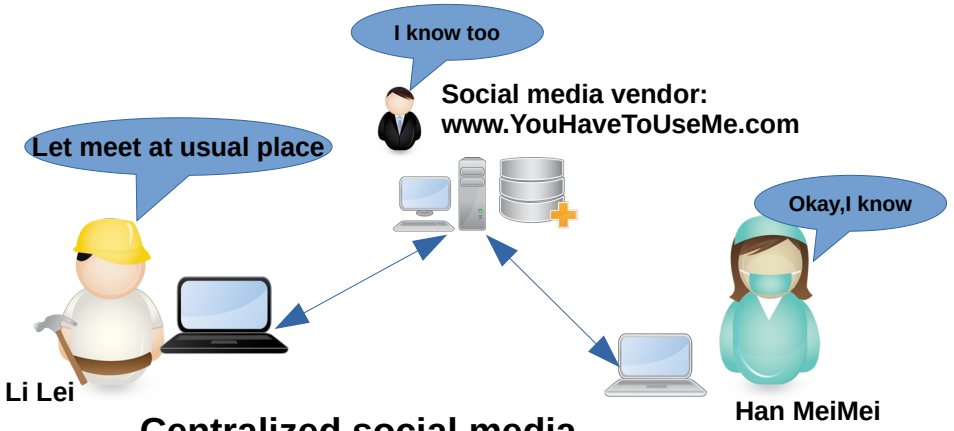
Decentralized social media design training

episode 1: training intro.

課程目的

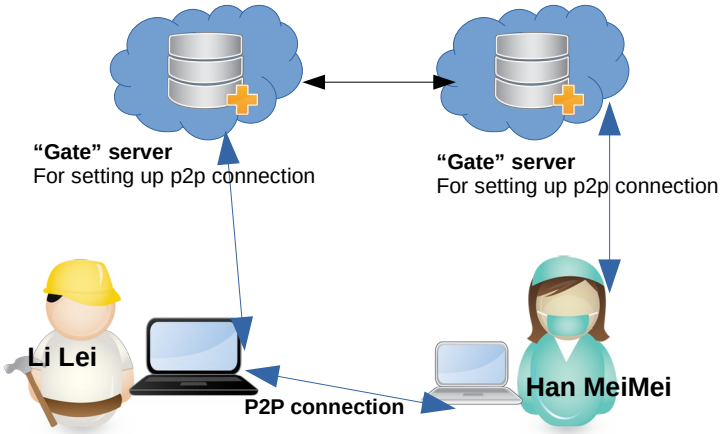
- 嘗試建立一種去中心化的社交媒體(信息，視頻，博客推送，相冊，等等）使得網絡上的peers（個人設備）能直接互相通信，就像我們個人在現實生活中的相互交際一樣，讓網絡的交際回歸正常的人際交際方式。



Centralized social media

All user data goes thru social media vendor's data center,vendor gets access to private data

VS



decentralized social media

No social media vendor is required!
Peer to peer secure connection
No one can easily collect our private data

- 完成這套(免費)課程你將會掌握以下技能：

1. 各種網站設計（前端/後端/數據庫，網站商城，視頻聊天，博客，等等)
2. 深度學習，IOT與網站的結合

課程預設

- 課程介紹
- 第一部分：前端/後端(包括建立一個自己的後端框架)/數據庫入門
- 第二部分：peer to peer信息/語音/視頻聊天系統
- 第三部分：博客/簡文推送和跟隨系統
- 第四部分：個人相冊
- 第五部分：基本的視頻點播系統
- 第六部分：基本的網站與深度學習NLP的結合
- 第七部分：簡單的IOT(internet of things,物連網)：通過網絡用手機遠程控制家裡的開發板，攝像頭等等
- 后续课程：嵌入式，FPGA相关的设计

网站已建和待建模块介绍

- https://decensormedia.org/

關於我

- 伍成和，草根，畢業本應當中學物理老師的，應大學裏喜歡微電子做起了工程師。這幾年因为政治言论沦为土共(CCP)“猎狐行动”的目标，在韩国济洲岛寻求政治庇护和他国工作签证。
- Linkedin : https://www.linkedin.com/in/brianwchh
- Twitter: @brianwchh
- Parler: @brianwchh
- Github: https://github.com/brianwchh
- 13年的工作經歷，做過的項目有：無線通信算法及FPGA實現，機器視覺(computer vision)，深度學習，前期主要是專注於嵌入式平台,主要用的語言是：verilog HDL, C++/C, python，最近開始做網站建設相關的設計。
- 最近會做的視頻課程包括：

  1. 去中心化的網站建設，即本課程 （免費）

  2. 我的創業項目：3D sensor的FPGA實現： https://youtu.be/bsx7o-PnaLA （會移至收費平台繼續）

Today’s agenda

- 提防幾種祕密害人的伎倆
- 網站的短（時）連接和長（時）連接
- 前端框架和演進歷史
- 後端語言的江湖
- 後端request-response的lifecycle
- 後端框架
- 我們要能夠自己寫一個框架
- DeCensorMedia的系統框圖
- Micro data center是以後的創業方向
- 未來手機是一個remote controller
- 前端和後端加密市場

提防幾種祕密害人的伎倆

- 信息可以救人！親身經歷，說出來是爲了讓別人不要傻傻地遭受這樣的悲劇。這幾種沒底線的手段都是及其隱蔽: 化學氣體，超聲波.....

- 三種隱蔽的方式會引起失眠。如果你覺得自己的言論等行爲得罪過這世界上最大的黑幫，出現了奇怪的失眠，應考慮換個地方，並且留意以下。我在2017年9月份出現了失眠，求遍醫院都無法醫治，只能長期靠安眠藥。直到2018年底在新加坡租住的房間發現窗簾上慢慢散發出來的不易察覺的氣味，才知道自己被盯上了！聯想起2017年五月份在深圳咖啡廳喝完咖啡經常出現頭暈心跳加快，還以爲是身體出問題了，打點滴全身裏面瘙癢！隔日上個臺階心跳像是長跑了一樣。現在把三年來的一些經驗和有必要的人分享一下，也請互相提醒！萬一碰到類似的事情，知道有這麼骯髒的伎倆，至少可以提防下，不至於深入圈套，不明不白死去！

  1. 化學氣體，對心髒不好。很難取證，它們即使明目張膽的在你樓下或附近排放，有一陣沒一陣的，你也拿它們沒轍。你也要提防下水道進來的氣體。最近又一個跟華爲有關的外國人心髒病死於酒店。當你連日感覺不到困意時，自己就要留心下房間裏的空氣。後來在濟周島想到了一個簡單的應對辦法，就是跟潛水艇一樣，做個塑料薄膜的3×3×3=27m^3大容器，在窗外把幹淨的空氣先用風機吹入容器內。人呼出的氣體還是含有大量的氧氣，可循環利用多次。雖然麻煩，但睡個安穩的覺比睡眠質量不好，甚至失眠和心髒不適要好！恨自己沒早點想到這方法，不然在新加坡也無需被迫多次搬家！

提防幾種祕密害人的伎倆(2)

2. 超聲波(>20,000Hz)/次聲波(<20Hz)。這種聲音聽不到，但卻能引起失眠和身體器官損壞。這個我無意中錄下了好幾個晚上的異常的聲音，麥克風是可以聽到這些聲波的，做頻譜分析可以看到，只是我們人耳朵的結果決定了我們能聽到聲音的最低是20Hz，最高是20,000Hz。如果數字濾波器做的不好，在做模擬(類比)信號到數位信號的時候，就會出現頻譜搬移到可以聽到的部位，這也是爲啥我們可以聽到這些超聲波。

以下是我上傳到youtube上的錄到的聲音：https://youtu.be/L_dbZssEKas?t=770 把音量開到最大可以聽的很清楚。

這個是美國在古巴駐使館官員的對超聲波的描述：說是像開車時窗戶打開時風吹的聲音一樣，和我錄到的很類似，而且我錄的有嗡嗡的刺耳的聲音。
https://www.nytimes.com/2019/01/04/science/sonic-attack-cuba-crickets.html

> "It could be like a low-tone motor, or metal scraping, or like driving in a car with the back window open" said Dr. Smith, director of the Center for Brain Injury and Repair at the University of Pennsylvania.

應對方法：可以買測超聲波的儀器，住人多的居民區。

提防幾種祕密害人的伎倆(3)

- 去醫院要小心。它們折騰你，下藥，最好偷偷下手的地方恐怕是醫院了。莫要莫名奇妙癌症了。

網站的短連接和長連接

- Request-response 的http/https 短時間的連接。網站必須要在接收到用戶的requst之後盡快處理完數據，然後作爲response返回給用戶的瀏覽器。所以這種短時間的連接不適合做復雜的耗時的運算，比如一些耗時（如一分鍾）的深度學習的處理。這種耗時的任務必須做成類似“異步”的，應快速返回一個唯一查詢碼給客戶，然後客戶過一段時間之後再憑此查詢碼去後臺數據庫要計算的結果。
- 長連接。即webSocket，客戶瀏覽器和後端服務器建立一種長時間的連接，有別於http/https的request和response的連接模式。信息聊天，視頻/語音聊天，網頁遊戲等等都是這種長連接。

几个基本概念

- Front-end (app) 前端(应用程序）：就是和网站后台打交道的程序，目前大致有2种：

  1. 浏览器。如不涉及复杂的耗时耗资源的算法，但开发时间较快，一套程序可以支持多个平台。目前浏览器可以取代很多手机APP。開發语言：html(好似建筑材料），css（指导建筑材料如何建构及建构成怎样的房子）, javascript（主要处理网页点击事件）.

  2. 各个商家定制开发的电脑/手机APP，如line等手机应用程序。程序运行速度快，开发时间长！开发语言，android支持java，IOS（苹果）支持swift！

前端框架及演进历史

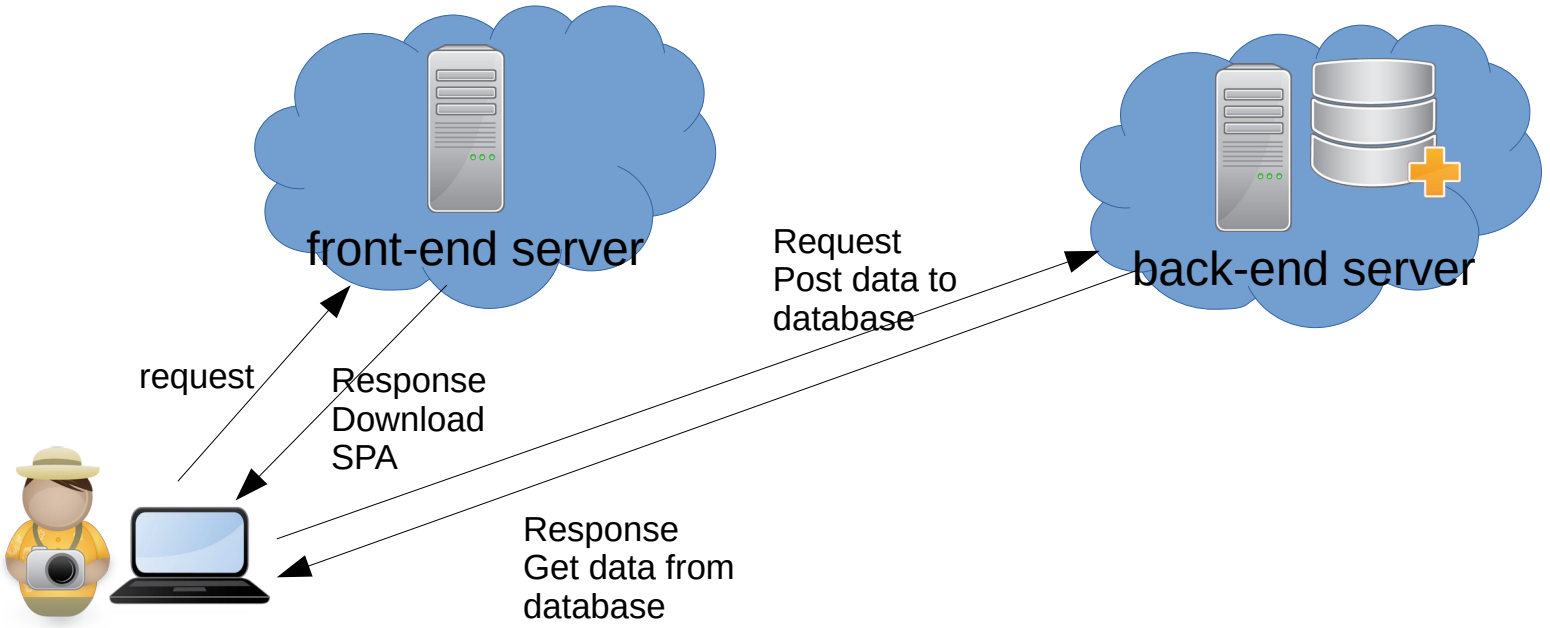
- 每个新技术的出现都是因为有历史痛点。在不是很久很久以前，网站的模式是这样的：网站服务器在收到访问/更新/修改某页面request之后，渲染(render)出此网页，然后作为response返回给用户

- 以上网站框架仍然在用，其不足之处是每次点击页面，本只需更新一小部分网页内容，它都会要重新从服务器下载整个页面！因此更新速度慢！在大量用户的情况下，这种网页更新给服务器增加不少负担，最重要的是，用户体实时体验不佳！

- 于是出现了前后端分离的SPA（single page application)框架(angular, react, vue...)。其主要就是解决上述问题，主要思路就是在服务器上分为前端服务器(front-end server) 和 后端服务器(server)。在前端app访问前端服务器时，它将程序打包传给前端app，其本质就是单个页面的javascript程式，只是页面的各个模块是根据用户的点击，在用户的浏览器上动态渲染出網頁，这种方法也叫做前端(app)页面渲染。这带来的好處是每次頁面更新，只需通過API向後端服務器取database必要的數據就可以了。

前端框架及演进历史(2)

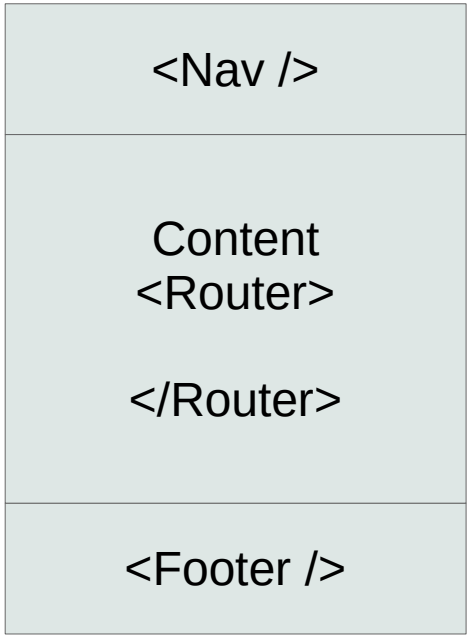
- 前端瀏覽器渲染SPA框圖如下



Front-end browser render.
In real project,we hide
back-end sever behind
nginx server via proxy

前端框架及演进历史(3)

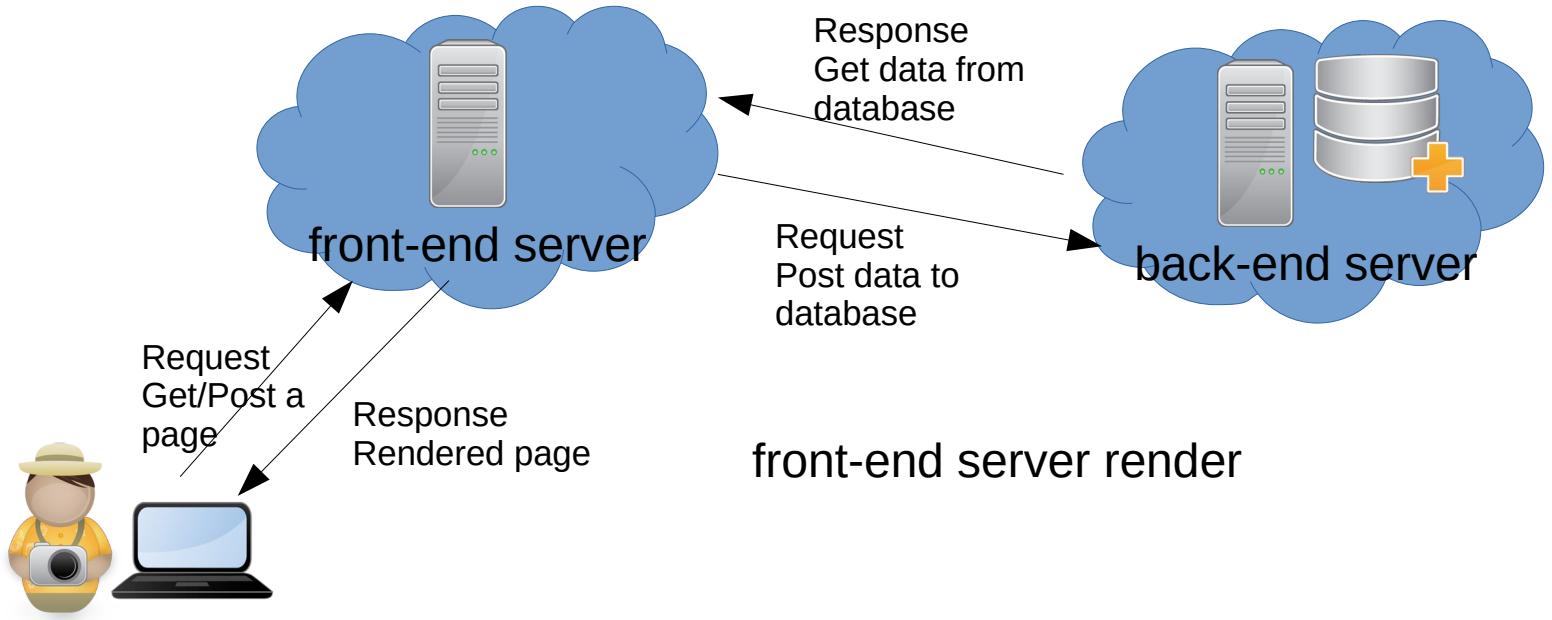
- 點擊網站的home或about按鈕，更新content部分的內容，而Nav和Footer不必更新




```
function App() {
  return (
    <div className="App">
      {/* header nav bar */}
      <Nav />

      {/* content */}

      <Router>
        <Switch>
          <Route path="/aboutus">
            <AboutUsInfo />
          </Route>

          <Route path="/">
            <Home />
          </Route>

        </Switch>
      </Router>

      {/* Footer  */}
      <Footer />

    </div>
  );
}

export default App;
```

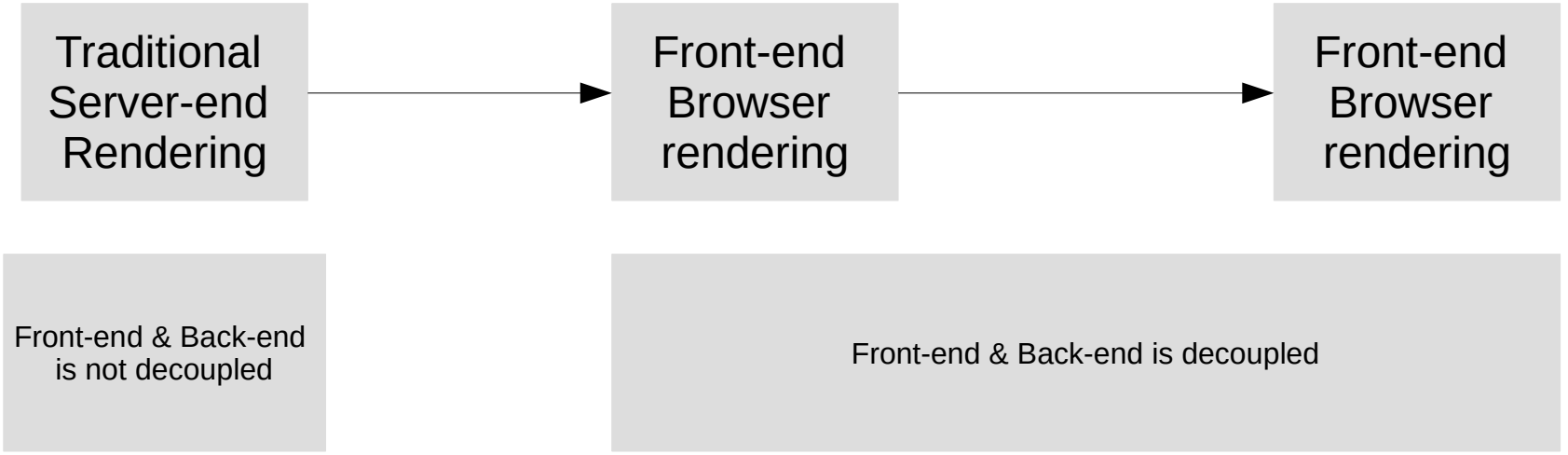
前端框架及演进历史(4)

- Front-end 瀏覽器render的不足。從客戶的角度來講，他們除了喜歡好的體驗感之外，自然都希望自己的網站在search engine的排名中比較靠前，這是SEO的範疇，然而似乎search引擎不會render javascript的代碼生成網頁。
- 於是又有了SPA front-end server render的框架,如居於react框架的Next服務器端的渲染。其框圖如下:



前端框架及演进历史(5)

- 小結:



- 目前框架未同時滿足兩類用戶:

  1. 真實的用戶的體驗需求

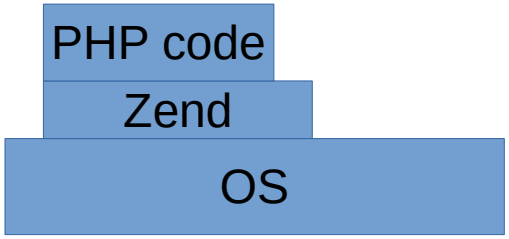
  2. search engine的網站爬蟲的排名需求

我們的前端解決方案

- 在這個項目裏面，我們會綜合使用SPA的前端瀏覽器端渲染，和前端服務器端渲染兩個框架。做到“給鬼看的內容，滿足鬼的要求；給人看的內容滿足人的要求”，需要做SEO的頁面優先用前端服務器渲染，追求瀏覽體驗的頁面優先用瀏覽器端渲染。

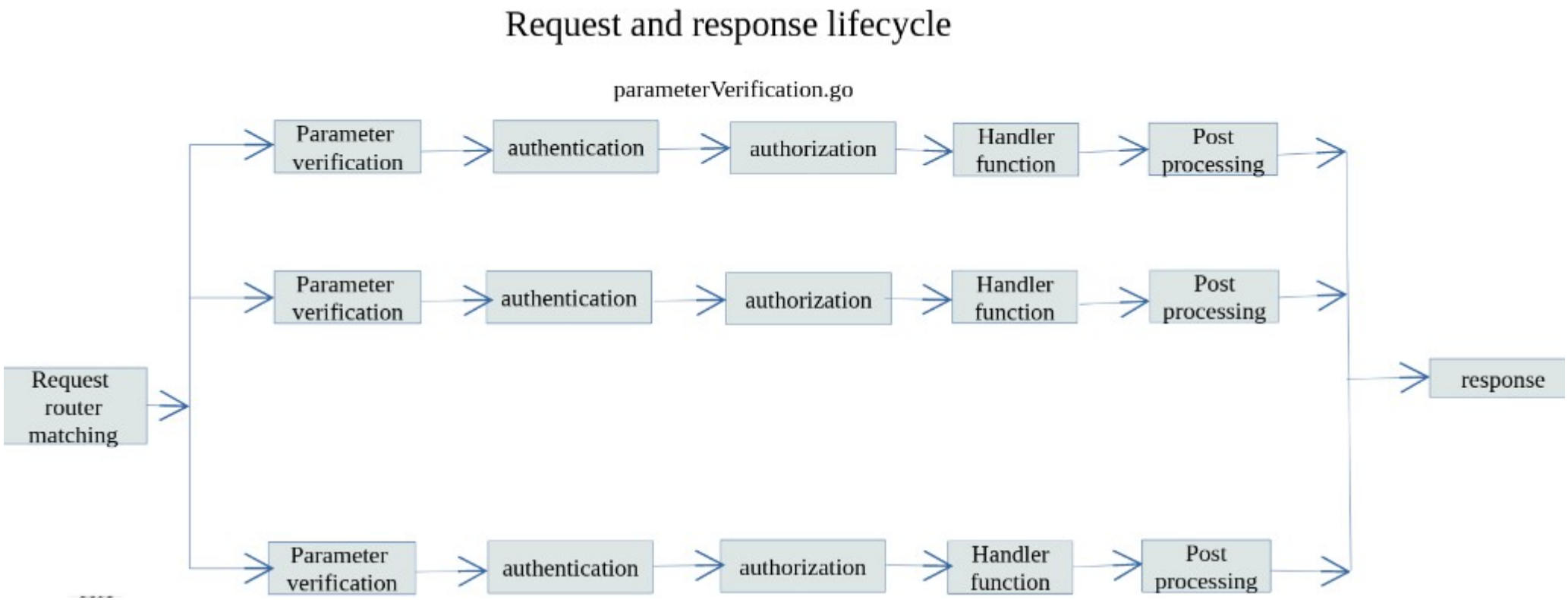
後端語言的江湖

- 從C++到PHP,Java,Javascript,再到golang
- 也是在不是很久很就以前，後端還是C++的天下，C++編譯生成的可執行文件直接可以在linux或者windows的OS上執行，加載速度，甚至運行速度都快。當然在golang之前，很多大型的高性能的網站還是得用C++來做，因爲只有直接運行在OS上的程序才能發揮硬件最大的性能。
- C++不足之處是設計難度比較大，需要人爲做細致的內存釋放，否則會出現程序崩潰，而且設計周期比較長。
- 之後，用非專業容易理解的話說，有人在OS上設計了引擎(engine)，然後針對這種引擎又設計了一種高級語言，這個引擎就是對這種語言進行“解釋”然後運行。比如PHP的引擎是Zend，javascript的後端引擎是nodejs，而nodejs本身就是是根據google瀏覽器的引擎而寫的C++應用程序，python/java也是如此，需要引擎來解釋和運行。



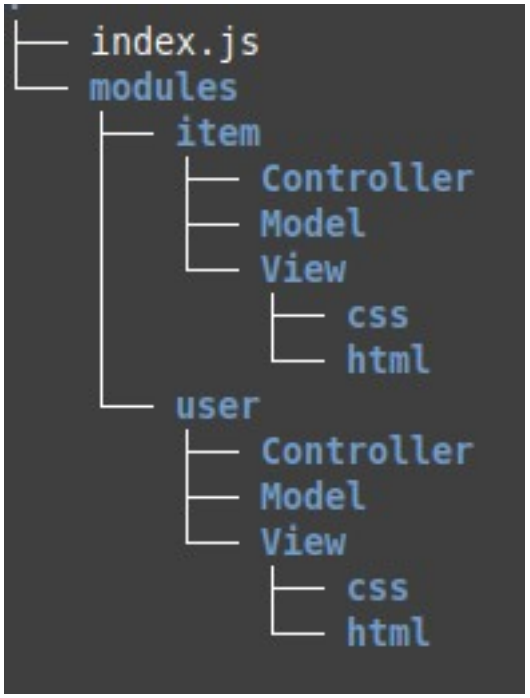
- 這些高級語言的好處是，engine有專門的機構設計了，可以i跨平臺。設計網站的速度也加快了很多。在性能和設計效率上取折中，很多無需追求高性能的網站用這些高級語言設計更有優勢，性能不足，可以用多買一些機房服務器來彌補。
- golang語言比較類似於C語言，但比C語言好設計，無需寫makefile，有垃圾內存回收功能，程序和C/C++一樣直接運行在os上。設計效率也比C++快。可以用於大型高性能的網站，也可以設計微服務器。

後端request-response life cycle



後端框架

- 常用的舊框架有MCV，以及由此改進的框架。所謂框架，更多的是一種team成員一起遵循的一種寫代碼和放置文件的“套路”，方便分工合作和代碼維護。如下圖所演示的MCV（只是示意圖）



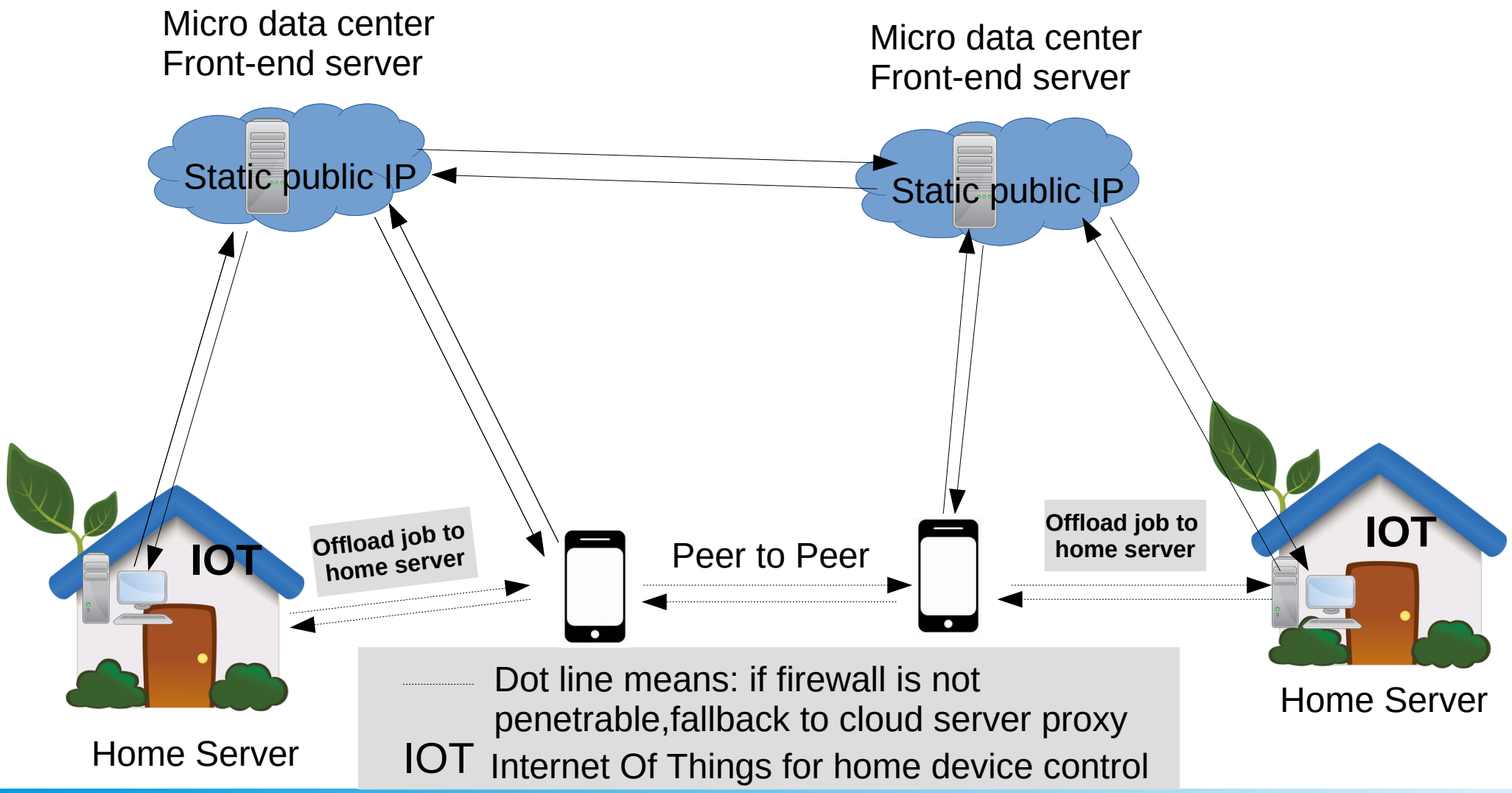
- Controller 文件夾下包含的就是路由文件，比如當前端服務器要向後端服務器讀取item list，get https://backend-server:port/api/item/，request會進入modules/item/Controller裏面的某個路由文件，路由文件將此request交給對應的handler函數處理。你可以將handler函數文件放在Controller文件夾下，也可以放在View文件夾下。
- Model 是跟數據庫相關的表格和讀取數據庫後的邏輯處理，handler函數會調用這裏面的操作數據庫的函數。
- View。render頁面相關的處理。html文件夾裏是和item相關的各種template文件，比如爲了前端能顯示itemlist, 我們會寫一個itemlist.html文件放置其中，對應css文件夾下有一個itemlist.css文件。我們需從數據庫中把各個item的名稱，圖片地址，價格，等等信息傳遞如itemplist.html中的變量中，然後用response返回給前端。

我們要能夠自己寫一個框架

- 不管哪種框架，我們只要抓住本質，就不容易被他們的套路繞暈了。而且不是所有的框架都滿足我們自己的項目需求。我們也不需完全去贊同別人的套路，比如controller這個概念就令人費解！當我們抓住request-response這條lifecycle,我們或許會覺得用router去代替它更容易了解，其本質就是將對應的request和handler_function連接起來的橋樑。當我們抓住resquest-response lifecycle這條主線，我們也不會被middleware/plugin這些概念搞的一頭霧水，我們可以自己任意寫一個plugin插入到對應的lifecycle中。我們完全可以自己寫一個自己的簡單的框架。
- 在這個項目裏，我們會嘗試自己用golang寫一個框架。
- github: https://github.com/brianwchh/buildGoFrameWork.git

我們要能夠自己寫一個框架(2)

DeCensorMedia系統框圖



微機房會成創業熱點

- 如果我們團結努力支持同一種協議，機房不再是大資本的市場，每個人都有機會參與
- 何爲機房？一個價格稍貴的public static IP + 服務器，何爲服務器？就是不需要顯示器的規格稍高的電腦。可以幾個人合夥買或者租，作爲私人使用，非商業級別，放在一個人家裏即可實現外網的訪問和轉發等功能的微型雲端機房。
- 機房最終要的品質是信任！社區裏的機房，或是相識之人的機房，或許會比遠在天邊的機房更讓人有信任感覺！人之常情！
- 做FPGA的朋友可以往微機房設備方向發展。

未來手機是一個remote controller

- 手機就如家庭IOT(Internet Of Things)的一個remote controller.
- 很多耗時好資源的應用可以offload到home server去處理。
- 人們無需再將自己的語音upload到別人的服務器做翻譯等，免費的服務不是免費的，我們的隱私更值錢。我們可以將自己的語音送到自己家庭服務器做翻譯，然後返回。

前端和後端加密市場

- 世上沒有一臺絕對安全的電腦。安全是和自由一樣，不是免費的!
- 我想在有選擇的情況下，沒有人願意自己成爲深度學習訓練中的一個數據點！但這麼些年來，大家在各種社交平臺上”裸聊“了這麼久，也沒覺得被冒犯了！正如只要一輩子沒察覺到你洗澡被人偷看了，就感受不到憤怒一樣。
- 再者從自己公司的服務器免費私下不知不覺地拿（盜？偷？有證據你告我？）數據和到別人家裏去偷，犯罪被抓的幾率和懲罰程度是不一樣的。
- 不同的個體有不同的安全規格需求，有市場有別的活路，自然會有一大堆程序猿會從大公司的樹上跳下來，直立行走，然後變成.....！(引用中國大陸歷史書上的進化論：））， 爲你服務的數據安全專家。
- 這前後端認證的加密市場會很大，或許還有嚴格到要用USB Key pair的，動帶密碼的.....