|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI  VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG  ──────── \* ───────  ĐỒ ÁN  **TỐT NGHIỆP ĐẠI HỌC**  NGÀNH HỆ THỐNG THÔNG TIN   |  |  | | --- | --- | | Sinh viên thực hiện | : **Bùi Ngọc Luân** | | Lớp | : **CNTT-TT 2.03 – K57** | | Giáo viên hướng dẫn | : **TS. Nguyễn Bình Minh** |   Hà Nội, 12-2016 |

Contents

[Chương I, 2](#_Toc469377506)

[1. Giới thiệu về Internet of Things 2](#_Toc469377507)

[1.1 IoT là gì ? 2](#_Toc469377508)

[1.2 Tiềm năng của Internet of Things 3](#_Toc469377509)

[1.3 IoT technology stack 4](#_Toc469377510)

[2. IoT Platform là gì ? 4](#_Toc469377511)

[2.1 Định nghĩa 4](#_Toc469377512)

[2.2 Tính chất 5](#_Toc469377513)

[3, Đặt vấn đề 5](#_Toc469377514)

[3.1. IoT đang phát triển rất nhanh 5](#_Toc469377515)

[3.2 Tiêu chuẩn cho IoT Platform 6](#_Toc469377516)

[3.3 Giải pháp cho việc tích hợp các IoT platforms 7](#_Toc469377517)

[3.4 Mục đích của đồ án 7](#_Toc469377518)

[4, Khảo sát một số nền tảng IoT. 7](#_Toc469377519)

[4.1. OpenHab 7](#_Toc469377520)

[4.2. Home-Assistant 8](#_Toc469377521)

[4.3. Kura 8](#_Toc469377522)

# Chương I, Chương mở đầu

## 1. Giới thiệu về Internet of Things

### 1.1 IoT là gì ?

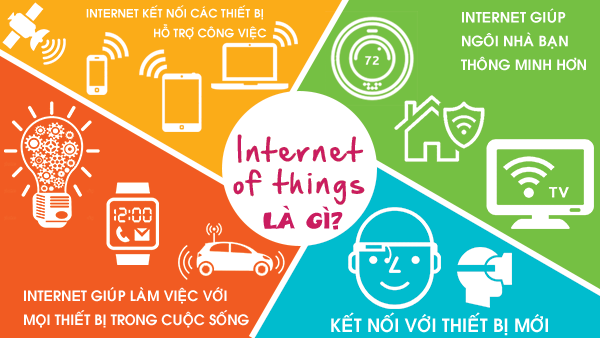
Mạng luới vạn vật kết nối Internet hoặc là mạng lưới thiết bị kết nối với Internet viết tắt là IoT (tiếng anh Internet of Things) là một kịch bản của thế giới, khi mà mỗi đồ vật, con người được cung cấp một dịnh danh của riêng mình, và tất cả có khẳ năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. IoT đã phát triển từ sự hội tụ của mạng không dây, công nghệ vi cơ điện tử và mạng Internet. Nói đơn giản là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài.

Figure 1. Internet of things là gì ( nguồn: iotvietnam.com)

Công nghệ IoT là một hệ thống mạng vật lý, mà dựa vào các thiết bị truyền động và cảm biến để thu thập dữ liệu từ thế giới vật chất. Các hệ thống lớn bao gồm nhiều thành phần trong IoT như là thành phố thông minh, nhà thông minh, giao thông thông minh và ngôi nhà thông minh. Các thiết bị IoT (cảm biến..) có thể xác định qua các hệ thống máy tính nhúng và tích hợp với cơ sở hạ tầng Internet hiện có. Theo Cisco, sẽ có khoảng 50 tỷ thiết bị IoT vào năm 2020.

### 1.2 Tiềm năng của Internet of Things

Kevin Ashton, đồng sáng lập và là giám đốc điều hành của Trung tâm Auto-ID tại Viện công nghệ Massachusetts, đã đề cập đến Internet of Things lần đầu tiên tại buổi thuyết trình ở công ty Procter & Gamble. Và đây là các Ashton đã giải thích về tiềm năng của Internet of Things:

"Ngày nay máy tính, và Internet, hầu như hoàn toàn phụ thuộc vào con người mới có thông tin. Gần như tất cả trong số khoảng 50 petabyte (1 petabyte là 1.024 terabyte) dữ liệu trên Internet lần đầu tiên được con người nắm và tạo ra bằng cách đánh máy, nhấn nút ghi âm, chụp ảnh hoặc quét mã vạch.

Vấn đề là, con người rất hạn chế về thời gian, sự chú ý và chính xác – nghĩa là con người không được tốt lắm trong việc lưu giữ dữ liệu về mọi thứ trong thế giới. Nếu chúng ta có những chiếc máy tính biết mọi thứ - sử dụng được dữ liệu chúng thu thập mà không cần sự giúp đỡ của con người – thì chúng ta sẽ có thể theo dõi và đếm mọi thứ, điều này sẽ giúp giảm rất lớn sự lãng phí, thất bại và chi phí. Chúng ta sẽ biết khi nào mọi thứ cần thay thế, sửa chữa hoặc phục hồi và liệu chúng còn có thể còn tiếp tục hoạt động hay hoạt động tốt nhất nữa không".

### 1.3 Sự bùng nổ của IoT

Khái niệm IoT được ra đời từ năm 1999, nhưng trong vòng vài năm trở lại IoT đang dẫn đầu 1 xu hướng, xu hướng kết hợp giữa các hệ thống ảo và thực thể, vạn vật kết nối Internet (IoT) và các hệ thống kết nối Internet (IoS). Xu hướng này có tác động không nhỏ tới nền kinh tế thế giới, cũng như cách mà con người đang làm việc. Một số người gọi đây là **cuộc cách mạng công nghiệp lần thứ 4**.

Song hành cùng sự phát triển của IoT luôn có sự góp mặt của các ông lớn công nghệ trong tất cả các lĩnh vực liên quan  
- Platform:

+ IBM với IBM Watson IoT

+ Amazon với AWS internet of things

+ Intel với Intel IoT Platform

+ Micosoft với Arure IoT Suite  
+ SamSung ARTIK Smart IoT Platform

+ Rất nhiều các sản phẩm IoT plarform mã nguồn mở được hỗ trợ mạnh mẽ bởi cộng đồng như OpenHab, Home-Assistant, OM2M, Kura...  
- Hardware (sensor, Kit):   
+ Apple với Apple Home Kit  
+ Samsung với SmartThings Starter Kit  
+ Google với Google Brillo

+ Intel với Intel IoTivity

+ Qualcom AllJoyn cho đến UPnP Forum, ARM mbed và nhiều tay đua mới nổi khác.

+ Cùng với đó là sự ra đời của rất vô số loại sensor được tích hợp vào trong các sản phẩm thân thuộc đời thường như TV, Tủ Lạnh, Quạt điện, Ổ cắm…..   
  
Qua những ví dụ kể trên, chúng ta có thể thấy được sự chú ý, quan tâm của thế giới trong ngành công nghiệp IoT và có cái nhìn cơ bản về sự phát triển của IoT trong vài năm trở lại đây.

### 1.4 IoT technology stack

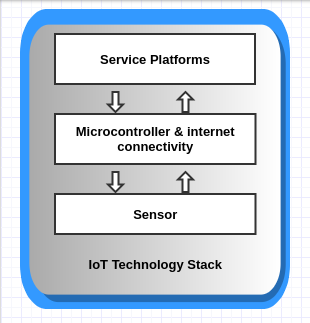
IoT Technology Stack là một mô hình gồm các lớp của một hệ thống IoT. Trong đó định nghĩa ra , chức năng hoạt động của mỗi lớp và cách lớp giao tiếp với nhau. IoT Stack gồm có 3 lớp cơ bản: Sensor, microcontrollers and internet connectivity and service platforms.

Figure 2. 3 layers of IoT Technology Stack

Lớp 1: Các sensor được nhúng vào các đối tượng vật lý để thu thập dữ liệu, điều khiển các đối tượng theo ý muốn.

Lớp 2: Cho phép lưu trữ, xử lí dữ liệu, và kết nối tới internet.

IoT cần kết nối internet để có thể truyền dữ liệu từ sensor tới cloud hoặc các trung tâm xử lí dữ liệu. Bởi vì có rất nhiều cảm biến tạo ra rất nhiều dữ liệu (10000 điểm dữ liệu /s). Nên lớp này có ý nghĩa là nơi tiền xử lí dữ liệu trước khi gửi đi để giảm thiếu khối lượng dữ liệu không cần thiết.

Lớp 3: Service Platform là lớp cung cấp cho người dùng những dịch vụ để người dùng có thể tiếp cận được tới những giá trị đã được khai thác từ việc thu thập và phân tích dữ liệu.

## 2. IoT Platform là gì ?

### 2.1 Định nghĩa

Để có thể triển khai các mô hình IoT, thì chúng ta cần một hạ tầng công nghệ mà trung tâm của nó là các nền tảng IoT. Nền tảng IoT được định nghĩa là nền tảng mà ở đó nó thực hiện các chức năng chính:

- Triển khai các ứng dụng, trình giám sát, quản lý, và kiểm soát các thiết bị đã kết nối.

- Thu thập dữ liệu từ xa từ các thiết bị kết nối.

- Kết nối độc lập và an toàn giữa các thiết bị.

- Quản lí các sensor/device.

Và IoT Platform là một thành phần tồn tại độc lập với phần cứng và các thành phần trong các lớp của IoT technology Stack [1].

### 2.2 Tính chất

Từ định nghĩa trên, ta có thể lọc ra được 8 tính chất mà IoT Platform cần phải có.

- Kết nối và chuẩn hóa: Đem các giao thức và định dạng dữ liệu khác nhau vào trong một “phần mềm” trong khi vấn đảm bảo chính xác luồng dữ liệu và tương tác tốt với tất cả thiết bị.

- Quản lí thiết bị: Liên tục chạy các bản vá lỗi và cập nhật phần mềm sao cho đảm bảo các thiết bị đã kết nối đều hoạt động tốt.

- Cơ sở dữ liệu: Thu thập và lưu trữ dữ liệu từ các thiết bị đã kết nối.

- Quản lí các hành động: cho phép thiết lập các quy tắc để có thể thực hiện các hành động thông minh dựa trên dữ liệu đã thu thập được.

- Phân tích: Thực hiện một loạt các phân tích phức tạp từ các dữ liệu đã thu thập được để có thể đưa ra những thông tin giá trị.

- Hình ảnh: Cho phép người dùng có thể có thể xem trạng thái thiết bị, các biểu đồ trực quan thông qua các biểu đồ.

- Các công cụ khác: Cho phép các lập trình viên có thể phát triển các plugin, để tạo nên hệ sinh thái phong phú và đa dạng hơn.

- Tính khả mở: Tích hợp được với các hệ thống bên thứ 3, thông qua API hoặc các SDK, gateways.

## 3, Khảo sát một số nền tảng IoT.

Sau khi xác định được mục đích của đồ án, em tiến hành khảo sát một số nền tảng IoT đang được cộng đồng mã nguồn mở phát triển.

### 3.1. OpenHAB

#### a, Giới thiệu về OpenHab

OpenHAB là một IoT Platforms mã nguồn mở dùng cho hệ thống nhà thông minh cho phép người dùng tích hợp các công nghệ khác nhau thành một hệ thống duy nhất. Openhab cho phép thiết lập các quy tắc tự động hóa trong hệ thống một cách mềm dẻo và cung cấp cho người dùng một giao diện điều khiển thống nhất.  
OpenHab định nghĩa ra một khái niệm gọi là **Item.** Một Item đơn vị nhỏ nhất của hệ thống. OpenHAB không quan tâm tới giá trị của Item đến từ các thiết bị vật lí, hay dữ liệu ảo hóa được gửi từ web hoặc các dữ liệu mô phỏng. Tất cả các thông tin liên quan tới item đều nằm ở tầng trừu trượng mà openHAB định nghĩa ra. Nghĩa là chúng ta không thể truy vấn trực tiếp tới các thông tin liên quan của thiết bị kết nối trực tiếp với item như địa chỉ IP, ID…đây có thể là một điểm trừ nhưng cũng là điểm cộng khi làm cho người sử dụng có thể dễ dàng thay đổi công nghệ ở phía dưới mà không cần thay đổi các quy tắc và giao diện.

#### b, Các tính năng của OpenHab

- Có thể chạy trên bất kì thiết bị nào có môi trường JVM (Linux, Mac, Window).

- Cho phép dễ dàng tích hợp các công nghệ (sensor, device) của các hãng khác nhau vào một sản phẩm hoàn thiện

- Có giao diện điều khiển nền web, và phiên bản mobile dành cho iOS và Android.

- Hoàn toàn là mã nguồn mở.

- Cung cấp REST API để có thể tích hợp với bên thứ 3.

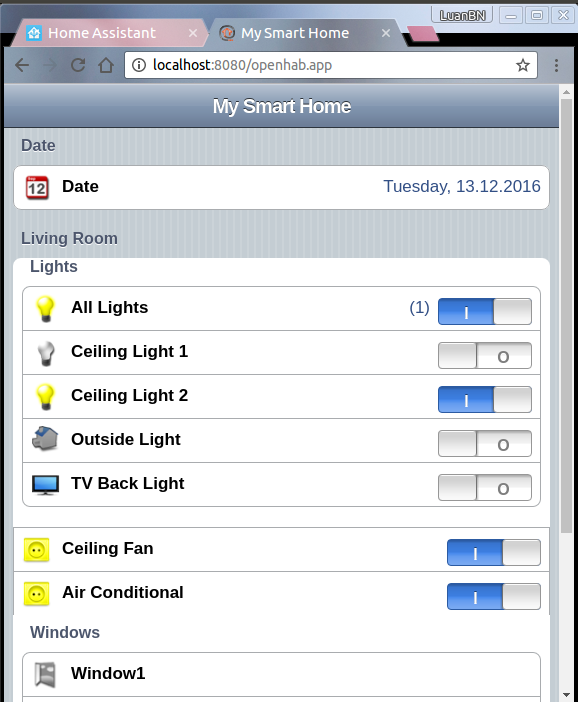


Figure 3. Giao diện điều khiển của OpenHAB

#### c, Cơ chế binding của OpenHAB

Binding là các gói tùy chọn có thể giúp mở rộng chức năng của OpenHAB.Binding cung cấp cơ chế tích hợp công nghệ nhà thông minh và các thiết bị với nhiều bundle khác nhau hỗ trợ tích hợp và giao tiếp với các mạng xã hội, nhắn tin, điện toán đám mây của các nền tảng IoT và rất nhiều các thành phần khác. Với sự hỗ trợ của cơ chế bidings, người dùng openHAB có thể dễ dàng điều khiển SamSung Smart TV hoặc điều hòa Daikin và còn nhiều nhiều các sản phẩm khác được OpenHAB hỗ trợ.  
Mỗi công nghệ hoặc thiết bị, mạng xã hội hay nền tảng điện toán đám mây tích hợp vào OpenHAB đều được hỗ trợ bởi các gói cụ thể. Các gói này là tùy chọn và có thể dễ dàng thêm vào OpenHAB khi cần.

3 Bước cài đặt:

* Download gói dữ liệu dể binding vào trong thư mục cố định.
* Thiết lập cấu hình cho binding trong file openhab.cfg.
* Thiết lập các hành động và các luật cho item vừa mới binding.

#### d, REST API của OpenHab

REST API của openhab phục vụ cho nhiều muc jdidchs khác nhau. Nó có thể được sử dụng để ketesn ôi openHAB với các hệ thống khác vì nó cho phép đọc trạng thái của các items, cũng như cho phép cập nhật trạng thái của items bằng cách gửi các lệnh tới items.

REST API của openHAB rất nhanh, vì vậy nó có thể sử dụng cho các thành phần tương tác thời gian thực với openHAB, đặc biệt là từ giao diện của bên thứ 3.

Trong REST API của openHAB cung cấp 3 giao diện chính để tương tác:

Send Command: gửi một lệnh để thực hiện một số hành động (bật/tắt đèn).

Send Status: chỉ ra trạng thái của một số item đã thay đổi.

Get Status: lấy về trạng thái hiện tại của một item.

Ví dụ:

* Truy vấn trạng thái của tất cả sensor được thiết lập trong OpenHAB  
  + Vào đường link sau bằng trình duyệt:  
   http://{host}:{port}/rest/states

Dữ liệu sẽ được trả về ở dạng xml như hình dưới:

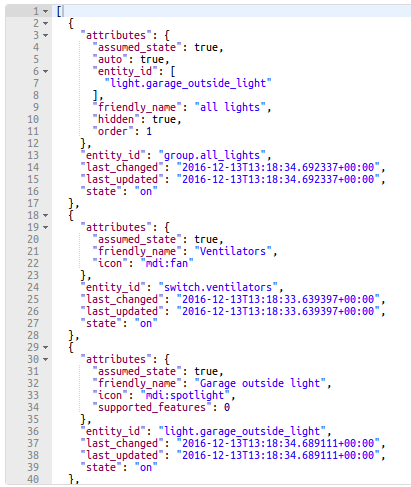


Figure 4. Trạng thái các item của OpenHAB

* Lấy trạng thái của 1 item chỉ định:

(ở đây là item Date)

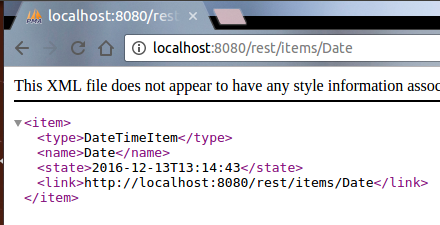


Figure 5. Trạng thái của 1 item chỉ định

* Ngoài ra, việc cài đặt và cập nhật trạng thái cho sensor được thực hiện thông qua phương thức POST và PUT
  + curl --header "Content-Type: text/plain" --request PUT --data "OFF" <http://192.168.100.21:8080/rest/items/MyLight/state>
  + curl --header "Content-Type: text/plain" --request POST --data "ON" <http://192.168.100.21:8080/rest/items/MyLight>

### 3.2. Home-Assistant

#### a, Giới thiệu về Home-Assitant

Home-Assistant là một nền tảng nhà thông minh chạy trên nền Python 3. Mục tiêu của Home-Assistant là có thể theo dõi và kiểm soát tất cả các thiết bị ở trong nhà và cung cấp cho người dùng một nền tảng điều khiển tự động.

Home-Assistant quan niệm rằng hệ thống nhà thông minh thì nên chạy tại nhà, chứ không phải chạy trong đám mây. Chính vì lí do đó, Home-Assitant hỗ trợ rất nhiều các thiết bị IoT (484 các loại thiết bị và nền tảng) đảm bảo việc triển khai hệ thống nhà thông minh diễn ra một cách dễ dàng nhất. Mỗi thiết bị được kết nối tới Home-Assistant gọi là một component. Mỗi component sẽ được config theo một cách chuyên biệt trong file config.

#### b, Các tính năng của Home-Assistant

* Cho phép chạy trên các môi trường có cài đặt Python 3.5 trở lên ( Window, Mac, Linux).
* Hỗ trợ tích hợp gần 500 các loại thiết bị và giao thức.
* Giao diện thân thiện với người dùng (có bản web và mobile).
* Hỗ trợ tính năng push notification.
* Cho phép đặt ra các luật tự động trong nhà thông minh.
* Hỗ trợ REST API và có thể giao tiếp qua MQTT để có thể dễ dàng giao tiếp với các phần mềm bên thứ 3

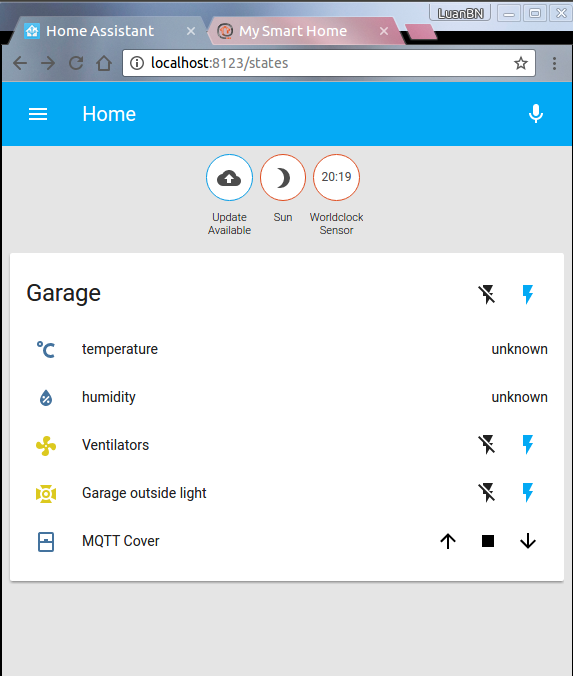
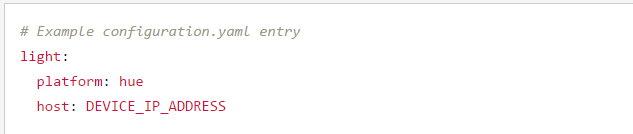


Figure 6. Giao diện điều khiển của Home-Assistant

#### c, Config trong Home-Assistant

Trong Home-Assistant, hầu hết các config đều được để ở trong file configuration.yaml. File này chưa toàn bộ các component sẽ được load lên và config riêng của các component này.

Ví dụ, để cấu hình một Philp Hue Light vào trong Openhab thì chỉ cần thêm config  


Nhưng để config một bóng đèn đi kèm trong bộ kit Z-wave thì cần phải thực hiện qua tương đối nhiều bước như config Z-wave platform trước (cần config network, cài đặt thêm driver..) sau đó mới config đến Z-wave light.

#### d, REST API của Home Assistant

Home-Assistant hỗ trợ REST API trả dữ liệu về dưới dạng JSON để người dùng có thể:

- Kiểm tra xem API có chạy không.

- Lấy các về config của các component.

- Truy vấn các hành động của người dùng.

- Lấy về trạng thái của các đối tượng đang hoạt động.

- Lấy về trạng thía của một sensor đang hoạt động

- Truy cập lịch sử trạng thái của 1 component chỉ định.

- Cập nhật hoặc cập nhật trạng thái của một component.

Ví dụ về việc truy vấn dữ liệu trong Home-Assistant

* Lấy về toàn bộ trạng thái của các component đang hoạt động.
* Lấy về trạng thái của một sensor chỉ định.

### 3.3. Eclipse Kura

#### A, Giới thiệu về Kura

Eclipse Kura là một dự án Eclipse IoT cung cấp một nền tảng cho việc xây dựng các IoT Gateway. Nó là một tập hợp các ứng dụng thông minh cho phép quản lý các gateway từ xa và cung cấp một loạt các API để có thể hỗ trợ người dùng triển khai ứng dụng IoT của riêng họ.  
(IoT gateway có vai trò tổng hợp dữ liệu của cảm biến, thông dịch giữa các giao thức của các cảm biến khác nhau, xử lý các dữ liệu này trước khi gửi đi).

#### B, Tính năng của Kura

Kura chạy trên môi trường Java Virtual Machine (JVM) và cung cấp một số chức năng chính sau:

* Quản lí từ xa:
  + Cho phép quản lý từ xa các ứng dụng IoT được cài đặt trong Kura, bao gồm các dự ản đã triển khai, nâng cấp, quản lý cấu hình và các dịch vụ đám mây.
* Networking:
  + Cung cấp các API để cấu hình cho các network có sẵn ở gateway như Ethernet, Wifi hay modem di động.
* Giiao diện quản trị web
  + Cung cấp một trang web để quản lý các gateway.

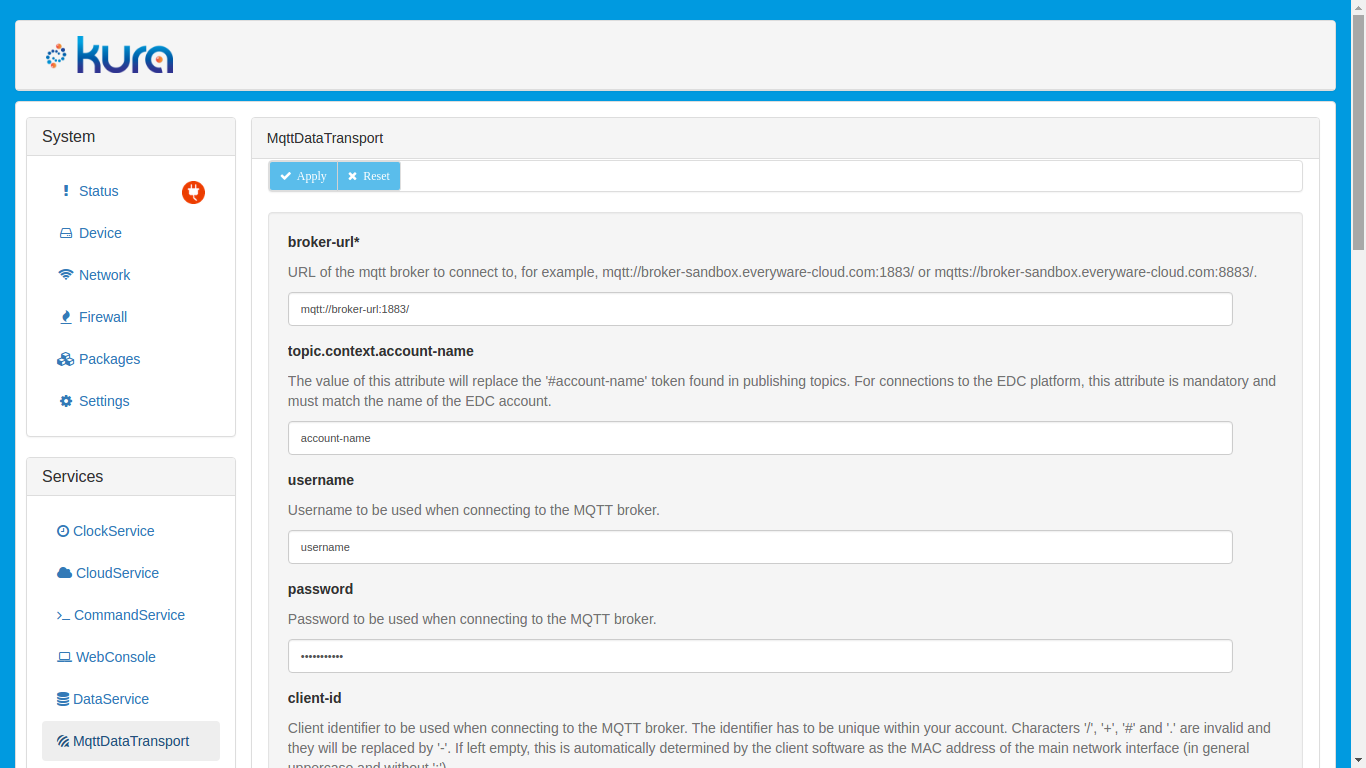
Giao diện quản lí, kết nối tới MQTT Service của Kura

Figure 7. Giao diện quản lí MQTT Service Kura

Giao diện quản lí, kết nối tới Cloud Service

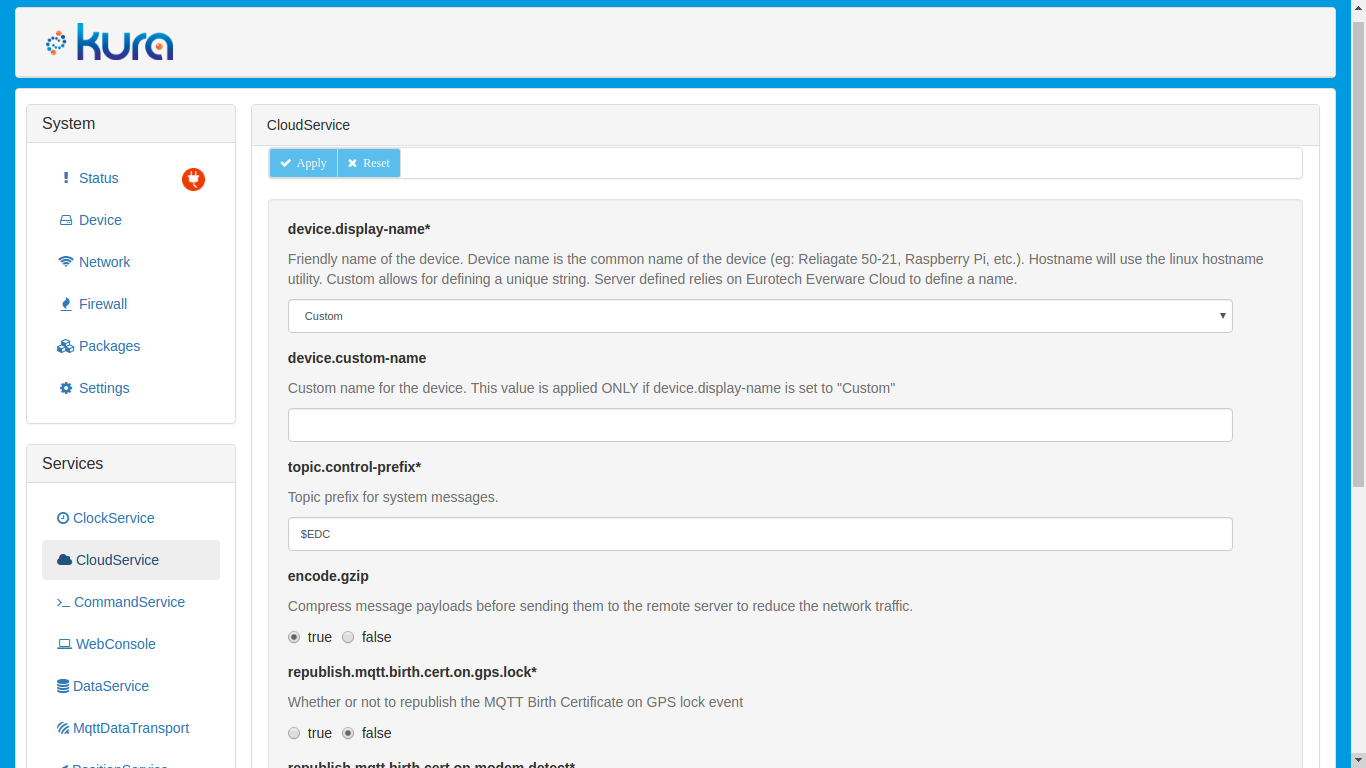


Figure 8. Giao diện quản lí Cloud Service của Kura

#### C, REST API của Kura

Kura quản lí các gateway bằng các bundle giống như openHAB. Tuy nhiên, Kura phép người đọc tự build các bundle của mình dựa trên các package mà Kura đã cho sẵn và có thể tích hợp các bundle đó vào Kura dưới hình thức là một OSGi plugin.   
Chính vì thế, bạn có thể tự xây dựng một REST API cho mình dựa trên các thư viện có sẵn của Kura.

## 4, Đặt vấn đề

### 4.1 Đưa ra bài toán

Hãy tưởng tượng bạn thuê công ty A triển khai lắp đặt nhà thông minh cho bạn. Dự án có mô hình lắp đặt các thiết bị như sau:

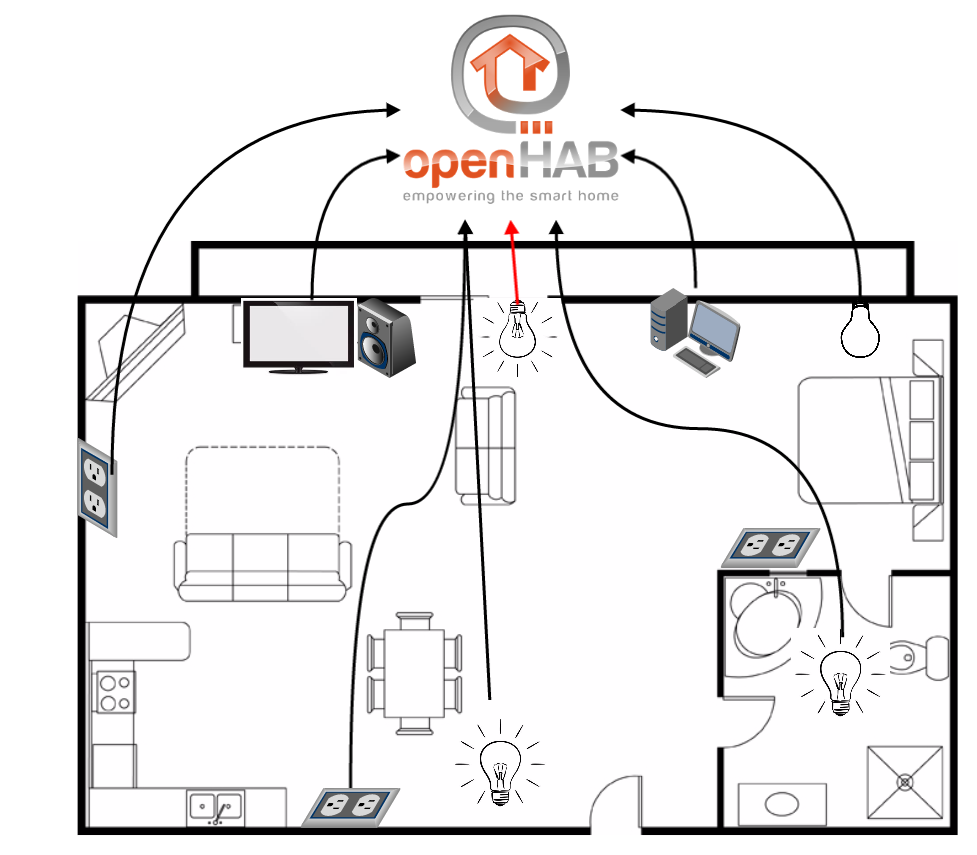


Figure 9. Nhà thông minh quản lí bởi OpenHAB

Ở đây, công ty A đã sử dụng **OpenHAB** để quản lí các thiết bị thông minh như:  
- Đèn chiếu sáng

- Nguồn điện, công tắc điện

- Tivi, loa

- Dàn máy tính

Sau hai năm, bạn tiến hành cơi nới thêm phòng ngủ và có mong muốn lắp đặt hệ thống thông minh trong căn phòng mới tích hợp với hệ thống có sẵn. Nên bạn liên hệ lại với công ty A, tuy nhiên vì lí do nào đó (giả thuyết là OpenHAB không được tiếp tục phát triển), bên A đã chuyển từ OpenHAB sang sử dụng Home-Assistant. Từ đây nảy sinh ra 2 vấn đề chính:

- Home-Assitant không hỗ trợ các sensor cũ trong hệ thống.

- OpenHAB không hỗ trợ các sensor mới.

Nên phòng ngủ của bạn đã được lắp đặt với một hệ thống hoàn toàn mới lấy Home-Assisstant là nền tảng.

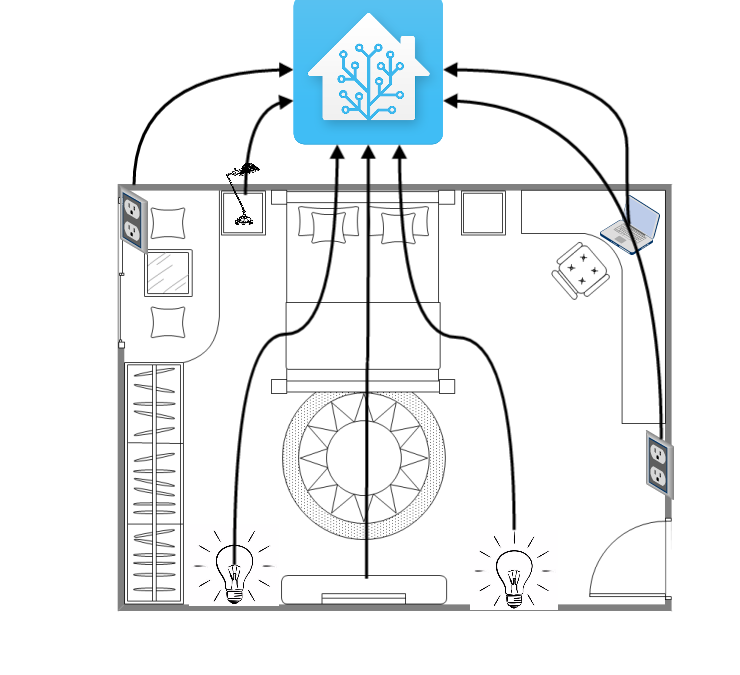


Figure 10. Hệ thống trong phòng ngủ do Home-Assitant quản lý

Vấn đề nảy sinh ở đây là trong một căn nhà mà bạn phải điều khiển thông qua 2 phần mềm khác nhau (2 điều khiển khác nhau). Dẫn đến sự khó khăn trong điều khiển và giảm chất lượng trải nghiệm của bạn khi sử dụng nhà thông minh. Và nghĩ xa hơn là trong một đến hai năm nữa bạn lại cơi nới nhà của mình, thì biết được lúc ấy IoT plarform nào sẽ được sử dụng để lắp đặt trong những căn phòng mới đó.

Và từ ví dụ ở trên, ta đặt ra câu hỏi: liệu có cách nào tích hợp nhiều IoT Platform lại với nhau để người dùng có thể dễ dàng điều khiển được căn nhà của mình thông qua một nền tảng duy nhất.

### 4.2 Tiêu chuẩn cho IoT Platform

Tính đến ngày hôm nay, đã có hơn 300 Iot Plarform và con số ấy đang tiếp tục tăng lên. Tuy nhiên, không phải mọi nền tảng đều như nhau, mỗi nền tảng đang được hình thành từ các chiến lược thị trường của các công ty khác nhau để cố gắng tận dụng hết tiềm năng của IoT. Các Startups sáng tạo, các nhà sản xuất phần cứng, thiết bị mạng đình đám, đến các công ty quản lí di động đều cạnh tranh để nền tảng IoT của họ trở thành nền tảng IoT tốt nhất trên thị trường.



Figure 11. 14 trong số hơn 300 IoT Plarforms

Có vẻ như mọi chuyện dang diễn ra rất tốt, nhưng thực tế, tại triển lãm khoa học và công nghệ CES 2016 được tổ chức tại Las Vegas (Mỹ), khi các thiết bị Internet Of Things ra mắt thì người ta đặt câu hỏi làm sao có thể kết nối và đem tất cả các thiết bị thông minh gộp chung lại và sử dụng cùng nhau.   
Mối một sản phẩm được phát triển một công ty khác nhau, trong khi vẫn chưa có một chuẩn chung nào cho tất cả các thiết bị làm cho việc tích hợp các thiết bị này với nhau rất khó khăn. Một sensor có thẻ hoạt động tốt với IoT plarform này nhưng có thể không thể tích hợp với các IoT Plarform khác. Trong khi các sản phẩm mới ra liên tục, dẫn đến sự tích hợp, thay đổi cập nhật và phát triển các sản phẩm IoT vướng phải rất nhiều khó khăn.  
Diễn giải một giả thuyết dễ hiểu cho thấy khó khăn trong việc tích hợp các hệ thống IoT là cách đây 2 năm có thể bạn sử dụng sensor và một IoT Plarform để triển khai dự án nhà thông minh của bạn. Đến hiện tại bạn xây thêm 1 phòng và muốn đồng bộ với hệ thống đã dụng sẵn thì chắc chắn sẽ gặp nhiều khó khăn. Vì sensor cũ có thể đã ngừng sản xuất, nhưng sensor với driver mới hơn lại không được IoT Platform của bạn hỗ trợ. Điều này có thể dẫn đến trường hợp bạn phải lắp đặt một hệ thống hoàn độc lập với hệ thống cũ cho căn phòng mới xây của bạn.

### 4.3 Giải pháp cho việc tích hợp các IoT platforms

Chính vì khó khăn trong việc tích hợp, thay đổi cập nhật và phát triển của các sản phẩm IoT, khi muốn tích hợp các hệ thống lại với nhau, chúng ta sẽ nghĩ đến việc triển khai các IoT plarform một cách độc lập và triển khai một phần mềm ở phía trên có thể tích hợp và điều khiển các IoT Plarform.

\*TODO // tìm một số project có định hướng ///

### 4.4 Mục đích của đồ án

Chính từ tư duy ở trên, em đã phát triển đồ án tốt nghiệp của mình với mục đích mở ra một nền tảng cho phép kết hợp tất các IoT Platforms lại với nhau. Cung cấp cho người dùng một giao diện trực quan để dễ dàng sử dụng hệ thống mà không cần quan tâm quá nhiều đến các thành phần bên dưới.

Chức năng chính của hệ thống:

- Tích hợp việc điều khiển các thiết bị do nhiều IoT Plarform vào một giao diện duy nhất.

- Cập nhật dữ liệu của các thiết bị (sensor, device..) theo thời gian thực.

- Lưu trữ dữ liệu thu thập được từ các IoT Platform

- Cung cấp các API để người dùng có thể từ đó thiết lập các rules cho mục đích điều khiển tự động.

- Có tính khả mở (có thể tích hợp thêm IoT Platform một cách dễ dàng).

## 5. Phạm vi của đồ án và các công cụ được sử dụng

### 5.1 Tầm nhìn và phạm vi của đồ án

Với mong muốn có thể tạo ra một nền tảng hỗ trợ điều khiển và tích hợp tất cả các IoT Platform. Tuy nhiên, trong phạm vi giới hạn về thời gian của đồ án, em lựa chọn tích hợp hai IoT Platform là OpenHab và Home-Assistant để có thể chứng minh tầm nhìn và mục đích đồ án của em là khả thi.

### 5.2 Python2.7/Flask

Đồ án của em được chia làm 2 phần Back-end và Front-end. Trong đó phần Back-end được em viết bằng ngôn ngữ Python2.7 và framework Flask.

[Python là một ngôn ngữ lập trình đa mục đích](https://techmaster.vn/posts/33449/ngon-ngu-lap-trinh-tot-nhat-cho-nguoi-moi-bat-dau) được tạo ra vào cuối những năm 1980s, và được đặt tên theo nhóm kịch Monty Python, nó được sử dụng bởi hàng ngàn người để làm những việc từ kiểm thử vi mạch tại hãng Intel, sử dụng trong ứng dụng Instagram, cho tới xây dựng các video game với thư viện [PyGame.](http://pygame.org/) Nó nhỏ và chặt chẽ như ngôn ngữ tiếng Anh, và có hàng trăm các thư viện của bên thứ ba (third-party).

Em lựa chọn Python vì:

* Cú pháp Python rất dễ đọc

Python có điểm chặt chẽ rất giống với ngôn ngữ tiếng Anh, sử dụng những từ như 'not' và 'in' nên khi bạn đọc một chương trình, script, hoặc khi đọc to cho người khác nghe mà không cảm thấy giống như bạn đang nói một thứ ngôn ngữ bí mật nào đó. Điều này cũng được hỗ trợ bởi các quy tắc chấm phẩy câu rất nghiêm ngặt của Python, có nghĩa là lập trình viên không có những dấu ngoặc nhọn ({}) trong code của mình.

* Các thư viện phong phú

Python đã tồn tại khoảng hơn 20 năm, vì vậy có rất nhiều code viết bằng Python được xây dựng qua nhiều thập kỷ, và là một ngôn ngữ mã nguồn mở, rất nhiều trong số này được phát hành cho người khác sử dụng. Hầu như tất cả chúng được tập hợp lại trên trang web [https://pypi.python.org](https://pypi.python.org/), bạn phát âm nó là "pie-pee-eye", hoặc còn được gọi bằng một cái tên phổ biến hơn là "the CheeseShop". Lập trình viên có thể cài đặt phần mềm này lên hệ thống của mình để sử dụng bởi các dự án của riêng. Ví dụ, nếu muốn sử dụng Python để xây dựng những script với các đối số dòng lệnh, lập trình viên nên cài đặt thư viện "click" và sau đó import nó vào trong các script của mình rồi sử dụng nó. Có những thư viện sử dụng được cho khá nhiều trường hợp từ thao tác với hình ảnh, cho tới tính toán khoa học, và tự động hóa máy chủ.

* Python có một cộng đồng sử dụng lớn

Python có nhiều nhóm người sử dụng ở khắp mọi nơi, thường được gọi là các PUG, và họ tiến hành những cuộc hội thảo lớn trên tất cả mọi châu lục ngoại trừ Nam Cực. PyCon NA, hội nghị về Python lớn nhất ở Bắc Mỹ, đã bán ra 2.500 vé trong năm nay. Hội nghị này phản ánh cam kết đa dạng hóa của Python, vì có trên 30% diễn giả là phụ nữ. PyCon NA 2013 cũng bắt đầu một xu hướng của việc đưa ra workshop gọi là "Young Coder", nơi mà những người tham dự đã dạy Python cho trẻ em từ 9 đến 16 tuổi trong vòng một ngày, để cho chúng làm quen với ngôn ngữ này và cuối cùng giúp chúng hack và mod một số trò game trên con Raspberry Pi mà chúng được nhận. Việc trở thành một phần của một cộng đồng tích cực như vậy sẽ luôn tạo ra rất nhiều động lực cho những người lập trình.

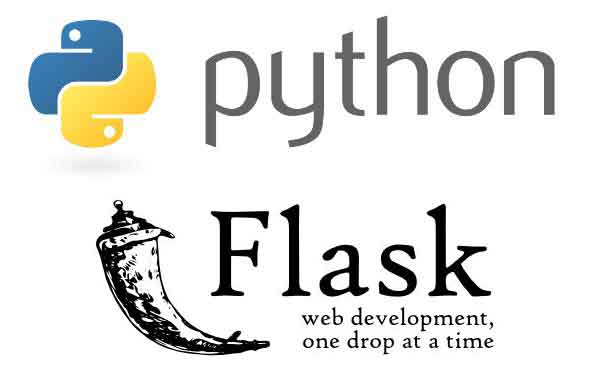


Figure 12. Python/Flask

Khi sử dụng Python, em biết có rất nhiều Python Framework như Django, web2py, CubicWeb…tuy nhiên em chọn Flask để viết back-end vì thấy nó phù hợp với mục đích bài toán mà mình đang làm: đơn giản, dễ học trong thời gian ngắn. Đơn giản, nhưng Flask lại rất hữu dụng vì:

- Bên trong Flask có hỗ trợ ORM, routing đầy đủ.

- Flask hỗ trợ mở rộng các tính năng qua extension.

- Các Extension của Flask rất nhiều, hỗ trợ từ xác nhận, xử lí tải file,….

- Flask và các Extension được cập nhật liên tục.

### 5.3 Angular2

**Angular 2**là một **javascript framework**(nôm na là 1 thư viện javascript được đóng gói và xây dựng nhằm dễ dàng tái sử dụng và xây dựng các ứng dụng có quy mô lớn cần yếu tố tổ chức và quy chuẩn). AngularJS 1.x được khai sinh từ năm 2009 với sự đỡ đầu và phát triển của Google, vì thế mà nó ngày một khẳng định được xu thế một cách nhanh chóng hơn các js framework cùng thời. Với bản chất là mã nguồn mở đúng nghĩa, Angular Js được đông đảo các lập trình viên đón nhận. Và sau một thời gian phát triển bấy lâu, đội ngũ Angular Team đã cho ra mắt Angular 2 kết hợp với TypeScript từ Microsoft để trở nên hoàn thiện hơn về cơ cấu tổ chức ứng dụng, cũng như tốc độ xử lý và hiệu năng khi sử dụng. Chính vì thế, nếu ai đã có kiến thức cơ bản về những mô hình trên, sẽ cảm thấy Angular 2 thật dễ dàng tiếp cận.



Figure 13. Angular2 + Flask

Angular 2 thích hợp xây dựng ứng dụng theo mô hình SPA (Single Page Application). Mô hình ứng dụng một trang duy nhất, các phân bố dữ liệu đều được truyền nhận âm thầm với kỹ thuật ajax kết hợp API tương tác với Web API. Chính vì tính tiện dụng này mà Angular 2 thường được ưu tiện lựa chọn cho các mô hình web application chuyên về front-end.

### 5.4 Docker

**Docker** là một Open Platform để xây dựng, vận chuyển và chạy các ứng dụng phân tán(Build-Ship-Run). Ban đầu viết bằng Python, hiện tại đã chuyển sang Go-lang.

Docker đưa ra một giải pháp mới cho vấn đề ảo hóa, thay vì tạo ra các máy ảo con chạy độc lập kiểu hypervisors (tạo phần cứng ảo và cài đặt hệ điều hành lên đó), các ứng dụng sẽ được đóng gói lại thành các Container riêng lẻ. Các Container này chạy chung trên nhân hệ điều hành qua LXC (Linux Containers), chia sẻ chung tài nguyên của máy mẹ, do đó, hoạt động nhẹ và nhanh hơn các máy ảo dạng hypervisors.

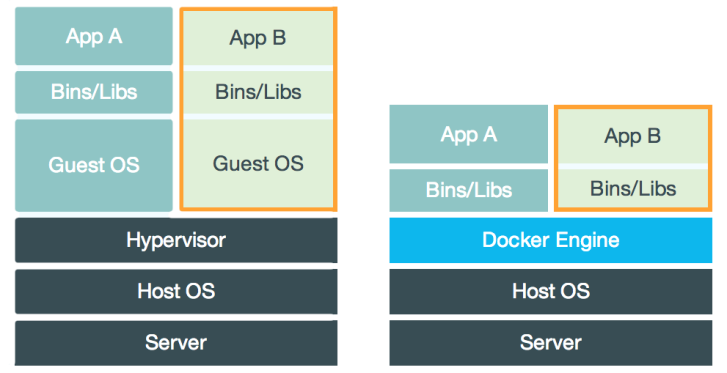


Figure 14. So sánh docker với virtual machine

Điểm khác biệt chính là các containers sử dụng chung kernel với Host OS nên các thao tác bật, tắt rất nhẹ nhàng, nhanh chóng.

**- Ưu điểm**: nhanh, nhẹ, có thể chia sẻ dễ dàng qua DockerHub.

- **Nhược điểm**: còn mới, cập nhật thay đổi thường xuyên.

## 6. Tóm tắt chương

Qua chương mở đầu của đồ án, em mong muốn nhưng ai đang đọc quyển đồ án này hiểu và năm được khái niệm về Internet of Things. Đồng thời hiểu được IoT Platform là gì, các chức năng chính của IoT Platform cũng như cách người ta sử dụng IoT Platform để dùng trong các bài toán thực tế. Trong chương này, em còn đưa ra vấn đề về sự bùng nổ quá nhanh của IoT mà không đặt ra các tiêu chuẩn chung kịp thời. Dẫn đến việc các hệ thống tích hợp vào nhau gặp rất nhiều khó khăn. Điều này cũng là bài toán đặt tiền đề cho đồ án của em. Về chi tiết của đồ án, xin mời mọi người đọc tiếp chương sau.

# Chương II, Các kết quả đạt được

Dựa theo mục tiêu đặt ra ban đầu của đồ án là tạo ra một nền tảng có khả năng tích hợp tất cả các IoT Platform lại với nhau. Trong phạm vi đồ án, em đã phát triển một nền tảng theo đúng mục đích ban đầu, và lựa chọn 2 IoT Platform là OpenHab và Home-Assistant như 2 nền tảng thử nghiệm để chứng minh nền tảng của em có thể tích hợp các IoT Platform lại với nhau.

## 1, Kiến trúc hệ thống

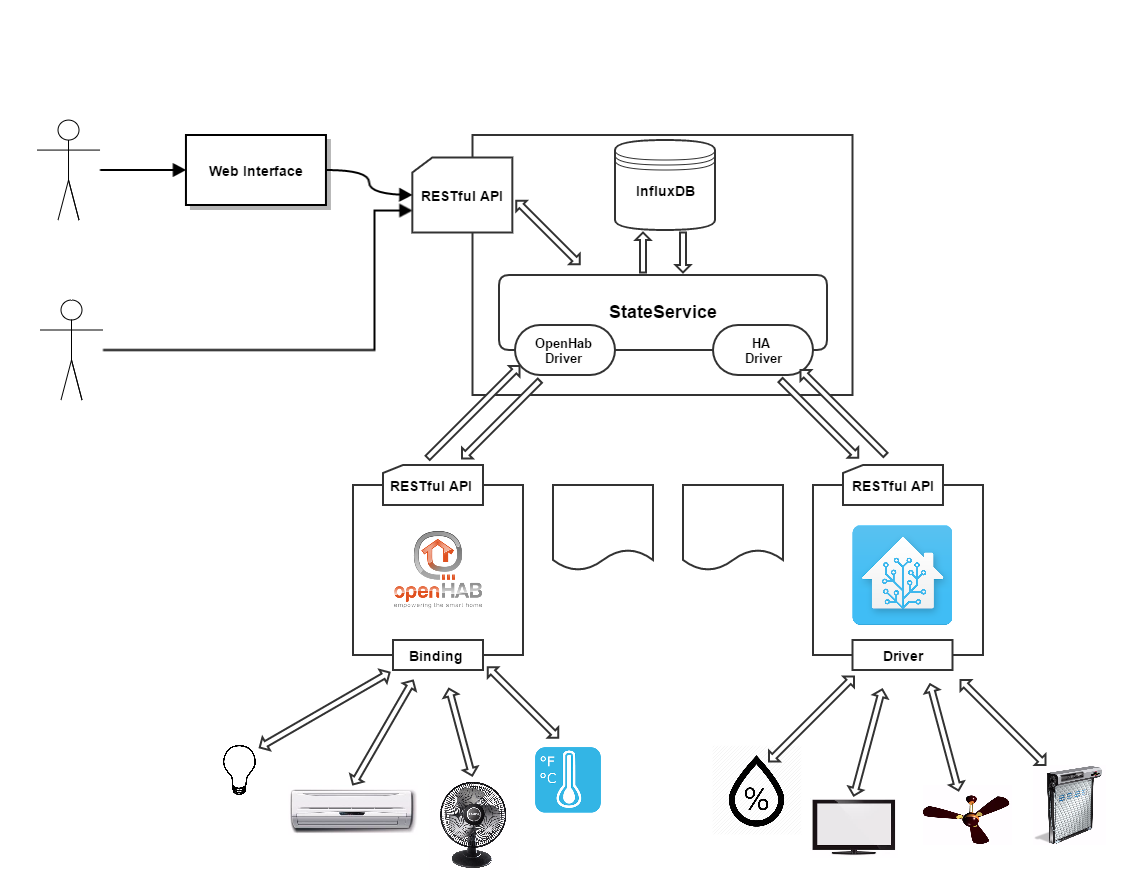


Figure . Kiến trúc hệ thống

Hệ thống được chia thành 3 thành phần chính:

Phần 1: