

Chương 2. Cấu trúc Hệ thống

2.1 Mô hình phần cứng

Theo Hình 1.1: Cấu trúc cơ bản của Indoor Positioning System, ta cần các thiết bị có nhiệm vụ thu, phát sóng Bluetooth Low Energy, ngoài ra thiết bị thu sóng phải có thêm tính năng kết nối mạng (WiFi hoặc dây Ethernet – để tăng tính linh động cho thiết bị ta sẽ sử dụng WiFi) để kết nối, gửi dữ liệu lên server. Như vậy, ta sẽ chọn các thiết bị sau:

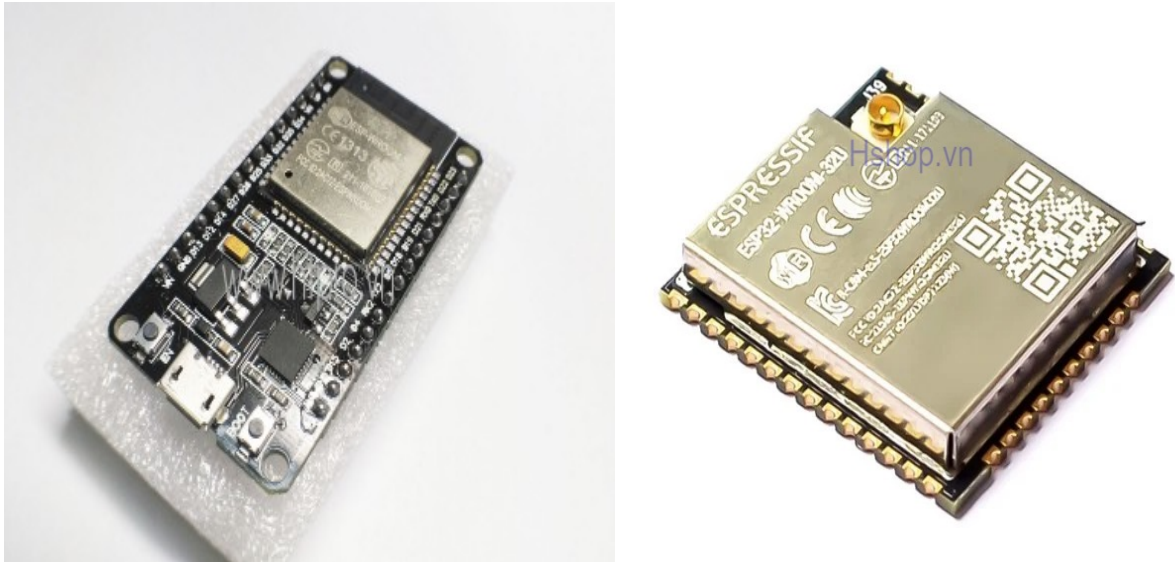
2.1.1 Kit RF Thu Phát Wifi BLE ESP32 NodeMCU LuaNode32:

ESP32 là chip kết hợp WiFi và Bluetooth băng tần 2.4GHz được sản xuất trên tiến trình ultra-low-power 40nm của TSMC, được phát triển bởi ESPRESSIF SYSTEMS (SHANGHAI) CO., LTD.. CPU của ESP32 là Xtensa® single-/dual-core 32-bit LX6 microprocessor, 448 KB ROM, 520KB SRAM, xung nhịp cơ bản 8MHz, thạch anh ngoài 2~60MHz (40MHz cho chức năng WiFi/Bluetooth).

Kit RF thu phát Wifi BLE ESP32 NodeMCU LuaNode32 được phát triển trên nền module trung tâm là ESP32 với công nghệ Wifi, BLE và nhân ARM SoC tích hợp mới nhất hiện nay, với ưu điểm là cách sử dụng dễ dàng, ra chân đầy đủ, tích hợp mạch nạp và giao tiếp UART CP2102, Kit Wifi BLE ESP32 NodeMCU LuaNode32 là sự lựa chọn hàng đầu trong các nghiên cứu, ứng dụng về Wifi, BLE, IoT và điều khiển, thu thập dữ liệu qua mạng.

Tuy nhiên, ESP32 NodeMCU LuaNode32 sử dụng Antenna Onboard, đây là loại antenna có hướng, loại antenna này nếu thu nhận sóng khác hướng đến thì sẽ làm cường độ tín hiệu bị giảm đi đáng kể dẫn đến khả năng ước lượng khoảng cách thiết bị. Vì vậy thay vì sử dụng Antenna Onboard, ta sẽ sử dụng Antenna IpeX là loại antenna lưỡng cực được cắm ngoài để hạn chế tối thiểu sự

chênh lệch giữa các hướng đến của tín hiệu. Vậy ta sẽ sử dụng module ESP32 ESP-WROOM-32U.



Hình 2.1: Kit ESP32 NodeMCU LuaNode32 (trái)

ESP32 ESP-WROOM-32U (phải)

Khi sử dụng ESP32 ESP-WROOM-32U, để tiết kiệm công sức đặt Kit phát triển (hiện tại Việt Nam có/ít kinh doanh) hay thiết kế mạch phần cứng tích hợp, ta có thể sử dụng Kit ESP32 NodeMCU LuaNode32 (khá phổ biến tại Việt Nam) và thay module hiện có của kit bằng module ESP32 ESP-WROOM-32U.

2.1.2 Beacon

Beacon là các thiết bị phát các tín hiệu Bluetooth Low Energy liên tục để ESP32 có thể đọc được RSSI của các tín hiệu này, từ đó gửi lên server để tính toán và ước lượng. Một chu kỳ phát tín hiệu của Beacon sẽ giao động từ 100-5000ms. Khi chu kỳ gửi càng ngắn (tức tần suất phát tín hiệu càng tăng) thì lượng dữ liệu được đưa lên server càng nhiều, dẫn đến đáp ứng về ước lượng càng nhanh và chính xác hơn. Từ đó ta có thể xây dựng một hệ thống với thời gian thực (real time).

Hiện nay trên thị trường Việt Nam có rất nhiều loại Beacon đa dạng về mẫu mã chất lượng, giá thành,....



Hình 2.2: Cấu trúc của E9 Dear Beacon

E9 Dear Beacon là một kiểu beacon sử dụng Bluetooth Low Energy, dựa trên chip BLE thế hệ mới – Nordic nRF52. Beacon này hỗ trợ cả công nghệ iBeacon(BLE của Apple) và Eddystone(BLE của Google), có thể tùy chỉnh cấu hình thông qua App trên Smart Phone.

Dear Beacon được thiết kế cho các ứng dụng trong môi trường ẩm ướt, bởi vì nó đạt tiêu chuẩn chống nước IP67. Ngoài nó, Dear Beacon còn được trang bị thêm cảm biến gia tốc (accelerometer sensor) nhằm thực hiện nhiều loại ứng dụng như, theo dõi tài sản, quảng cáo tại danh lam thắng cảnh,....

Bên cạnh đó, có thể lắp đặt Dear Beacon với ốc vít tại các khu vực công cộng và dễ dàng được tháo lắp bởi thiết kế xoay vò.

E9 Dear Beacon được trang bị viên pin 1000mAh có thể hoạt động liên tục 3 năm trong điều kiện Công suất phát 0dBm và chu kỳ 900ms). Tầm hoạt động có thể đạt đến 100 mét trong điều kiện lý tưởng và có thể tùy chỉnh công suất phát.

2.2 Server trung tâm

Các thiết bị phần cứng chỉ đóng vai trò thu thập dữ liệu để gửi lên server, mọi thao tác tính toán, lưu trữ đều được thực hiện một cách tự động trên server. Vì vậy ta phải chọn cấu hình của Server phù hợp với số lượng công việc mà Server phải giải quyết, cũng như lựa chọn phương thức liên lạc giữa server và thiết bị một cách phù hợp.

2.2.1 Microsoft Azure Compute Engine

Với sự phát triển với tốc độ vũ bão của khoa học công nghệ như hiện nay thì việc ứng dụng Cloud Computing trở nên phổ biến và được nhiều doanh nghiệp cũng như các ứng dụng khác trong đời sống xã hội sử dụng là điều cần thiết.

Ưu điểm nổi bật nhất mà Cloud Computing mang đến là việc sử dụng và khai thác các tài nguyên trên máy tính hiệu quả hơn rất nhiều, điều này giúp tiết kiệm chi phí đầu tư ban đầu và giúp cho người dùng không còn phải quá lệ thuộc vào các cơ sở hạ tầng. Không quá khi cho rằng Cloud Computing đang dần trở thành xu hướng của thời đại.

Với xu hướng mới của thời đại, các ông lớn trong ngành công nghệ luôn tìm cách xây dựng và phát triển một cách độc đáo và mãnh mẽ chẳng hạn như Google có Google Cloud Platform, IBM có IBM Cloud, và đương nhiên Microsoft cùng với Microsoft Azure không phải là ngoại lệ.

Microsoft Azure là một nền tảng ứng dụng cho công nghệ điện toán đám mây. Ta có thể sử dụng nền tảng này theo nhiều cách khác nhau. Chẳng hạn, ta có thể sử dụng Microsoft Azure để xây dựng các ứng dụng web để chạy hoặc lưu trữ dữ liệu trong Microsoft Azure DataCenters. Ngoài ra, chúng ta có thể dùng Microsoft Azure để tạo các máy ảo cho phát triển và kiểm thử phần mềm hoặc chạy SharePoints hay các nền tảng ứng dụng khác.

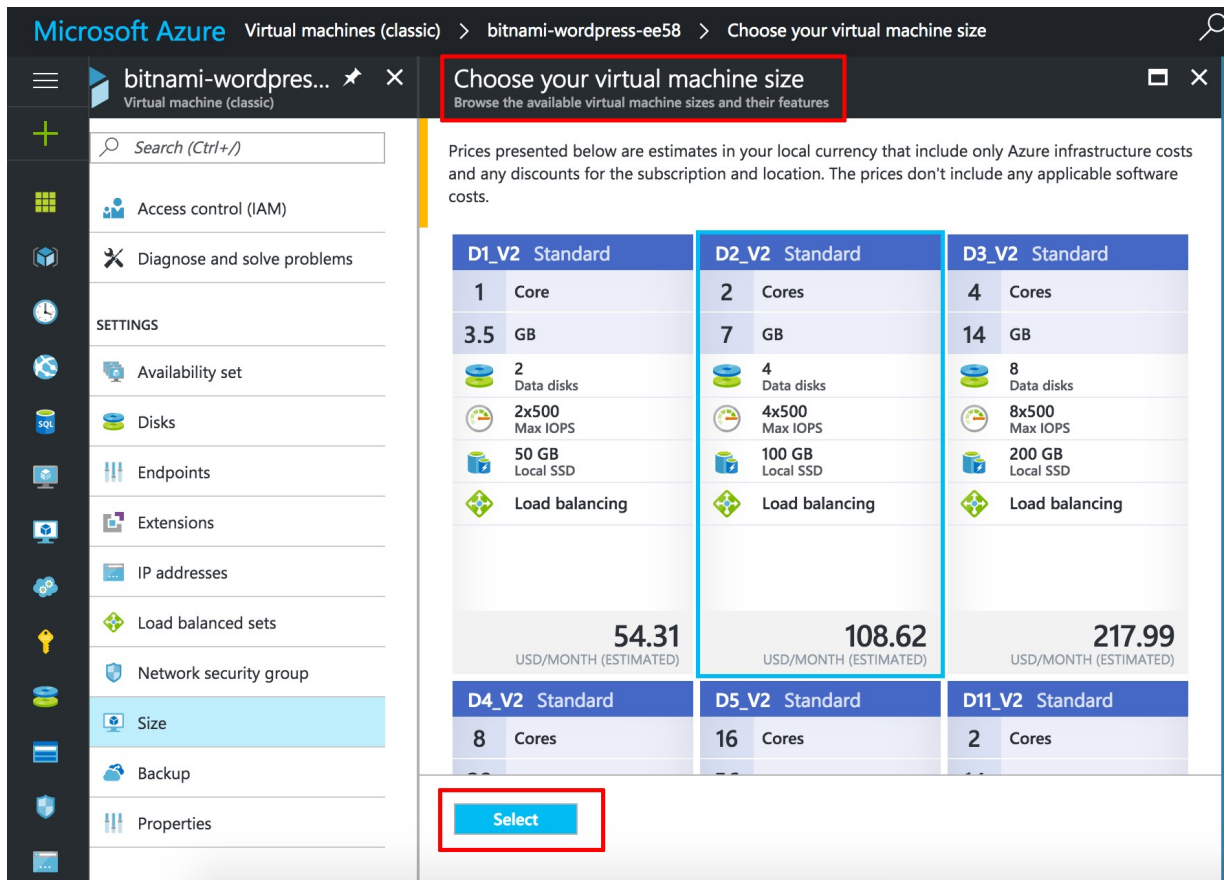
Virtual Private Server (VPS) của Microsoft Azure là một nền tảng mạnh mẽ, hỗ trợ nhiều tính năng, đáp ứng đủ điều kiện để chúng ta có thể thiết lập một server Indoor Positioning System. Một trong những tính năng đặc biệt của VPS là cho phép ta lựa chọn các thông số cấu hình phù hợp với yêu cầu của mình. Ta sẽ xây dựng một **Virtual Machine (VM)** làm VPS với cấu hình Intel Xeon E3-2 Core, Ram 4GB, IOPS 4000, chạy Hệ điều hành (OS) Ubuntu Server 18.4 LTS. VM này sẽ có nhiệm vụ thực hiện các phép tính toán trên dữ liệu mà các thiết bị phần cứng đưa lên, lưu trữ dữ liệu lên Database, thực hiện các Truy vấn (Command) của người dùng User.

Các ngôn ngữ sử dụng Python cho tính toán, JavaScript (Node.js) để thực hiện các truy vấn của User.

The logo for Microsoft Azure Virtual Machines. It features the text "Microsoft Azure" in a dark blue font above the text "Virtual Machines" in a larger, bold, dark blue font. To the right of the text is a large, stylized blue geometric shape composed of many triangles, resembling a crystal or a modern building facade.

Microsoft Azure
Virtual Machines

Hình 2.3: Microsoft Azure Virtual Machines



Hình 2.4: Tùy chọn cấu hình cho VM

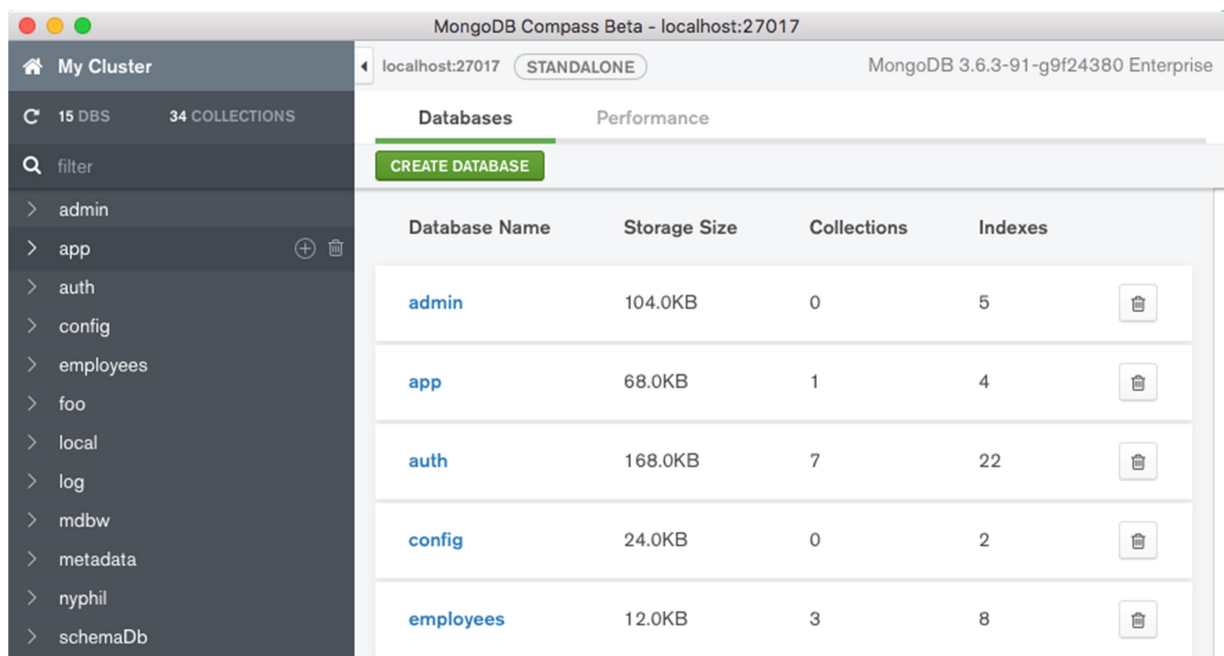
2.2.2 Cơ sở dữ liệu

Cơ sở dữ liệu (Database) là một tập hợp các dữ liệu có tổ chức, thường được lưu trữ và truy cập điện tử từ hệ thống máy tính. Khi cơ sở dữ liệu phức tạp hơn, chúng thường được phát triển bằng cách sử dụng các kỹ thuật thiết kế và mô hình hóa chính thức.

Hệ quản lý cơ sở dữ liệu (Database Management System - DBMS) là phần mềm tương tác với người dùng cuối, ứng dụng và chính cơ sở dữ liệu để thu thập và phân tích dữ liệu. Phần mềm DBMS bao gồm các tiện ích cốt lõi được cung cấp để quản trị cơ sở dữ liệu. Tổng cộng của cơ sở dữ liệu, DBMS và các ứng dụng liên quan có thể được gọi là "hệ thống cơ sở dữ liệu". Thông thường

thuật ngữ "cơ sở dữ liệu" cũng được sử dụng để nói đến bất kỳ DBMS, hệ thống cơ sở dữ liệu hoặc ứng dụng nào được liên kết với cơ sở dữ liệu.

MongoDB là một cơ sở dữ liệu hướng tài liệu mã nguồn mở phổ biến được phát triển bởi 10gen, sau này được gọi là MongoDB Inc. Trong trường hợp này, các tài liệu được tạo và lưu trữ trong các tệp BSON, định dạng Binary JSON (JavaScript Object Notation), vì vậy tất cả các loại dữ liệu JS là được hỗ trợ. Trong trường hợp đó, MongoDB thường được áp dụng cho các dự án Node.js. Ngoài ra, JSON cho phép chuyển dữ liệu giữa các máy chủ và ứng dụng web bằng cách sử dụng định dạng có thể đọc được. Nó cũng là một lựa chọn tốt hơn, khi nói đến dung lượng và tốc độ lưu trữ, vì nó mang lại hiệu quả và độ tin cậy cao hơn.



The screenshot shows the MongoDB Compass Beta interface. The title bar reads "MongoDB Compass Beta - localhost:27017". The interface includes a sidebar on the left with a "My Cluster" section showing "15 DBS" and "34 COLLECTIONS". Below this is a search bar labeled "filter" and a list of databases: admin, app, auth, config, employees, foo, local, log, mdbw, metadata, nyphil, and schemaDb. The main panel displays the "Databases" tab for the "localhost:27017" cluster, which is in "STANDALONE" mode. A "CREATE DATABASE" button is visible. Below this is a table listing the databases and their properties.

Database Name	Storage Size	Collections	Indexes
admin	104.0KB	0	5
app	68.0KB	1	4
auth	168.0KB	7	22
config	24.0KB	0	2
employees	12.0KB	3	8

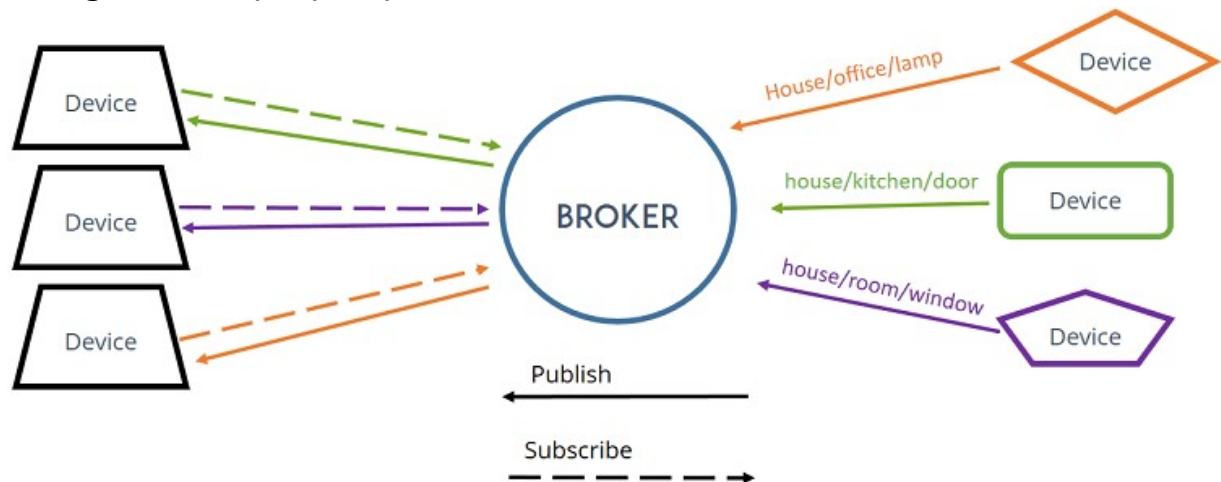
Hình 2.5: MongoDB

2.3 Giao thức liên lạc MQTT:

Message Queuing Telemetry Transport (MQTT) là một giao thức truyền thông điệp (message) theo mô hình publish/subscribe (xuất bản – theo dõi), sử dụng băng thông thấp, độ tin cậy cao và có khả năng hoạt động trong điều kiện đường truyền không ổn định.

MQTT là giao thức gọn nhẹ được thiết kế chủ yếu để kết nối các thiết bị bị hạn chế nguồn trên các mạng băng thông thấp. Mặc dù nó đã tồn tại trong hơn một thập kỷ nhưng chỉ khi có sự ra đời của Machine to Machine (M2M) và Internet of Things (IoT) mới làm cho nó trở thành một giao thức phổ biến.

MQTT sử dụng mẫu pub/sub để kết nối các bên với nhau. MQTT làm điều đó bằng cách tách người gửi (nhà xuất bản) với người nhận (người đăng ký). Nhà xuất bản gửi tin nhắn đến một chủ đề trung tâm, cái mà có nhiều người đăng ký đang chờ nhận tin nhắn. Nhà xuất bản và người đăng ký tự chủ, có nghĩa là họ không cần biết sự hiện diện của nhau.



Hình 2.6: Tổng quan giao thức MQTT

Các thuật ngữ trong MQTT:

Client: Bất kỳ nhà xuất bản hoặc người đăng ký nào kết nối với nhà môi giới tập trung qua mạng đều được coi là khách hàng. Cả nhà xuất bản và người đăng ký

đều được gọi là khách hàng vì họ kết nối với dịch vụ tập trung, khách hàng có thể liên tục hoặc tạm thời. Khách hàng liên tục duy trì một phiên làm việc với nhà môi giới trong khi khách hàng tạm thời không được nhà môi giới theo dõi. Khách hàng thường kết nối với nhà môi giới thông qua thư viện và SDK. Có rất nhiều thư viện có sẵn cho C, C ++, Go, Java, C #, PHP, Python, Node.js và Arduino.

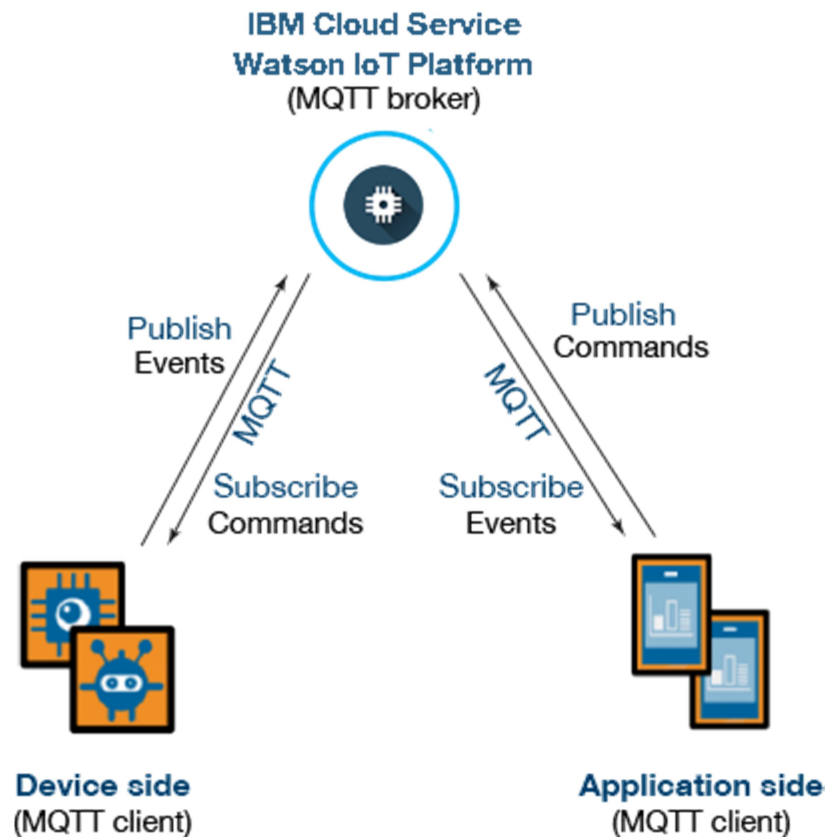
Broker: Người môi giới là phần mềm nhận tất cả các tin nhắn từ các khách hàng xuất bản và gửi chúng đến các khách hàng đăng ký. Nó giữ kết nối với các khách hàng liên tục. Tùy thuộc vào người triển khai để quyết định cách tạo lớp môi giới có thể mở rộng. Một số triển khai thương mại của các nhà môi giới MQTT bao gồm HiveMQ, Xively, AWS IoT và Loop.

Topic: Một chủ đề trong MQTT là điểm cuối mà khách hàng kết nối. Nó hoạt động như nơi phân phối trung tâm để xuất bản và đăng ký tin nhắn. Trong MQTT, một chủ đề là một vị trí nổi tiếng cho nhà xuất bản và người đăng ký. Nó được tạo ra khi chúng ta thiết lập kết nối với nhà môi giới. Chủ đề là các chuỗi phân cấp đơn giản, được mã hóa bằng UTF-8, được phân cách bằng dấu gạch chéo. Người đăng ký có thể chọn đăng ký một chủ đề cụ thể hoặc tất cả các chủ đề phụ thông qua các ký tự đại diện.

Connection: MQTT có thể được sử dụng bởi các máy khách dựa trên TCP/IP. Cổng tiêu chuẩn được giới thiệu bởi các công ty môi giới năm 1883, không phải là một cổng an toàn. Những nhà môi giới hỗ trợ TLS/SSL thường sử dụng cổng 8883. Để liên lạc an toàn, khách hàng và nhà môi giới dựa vào chứng chỉ kỹ thuật số.

Hiện nay có rất nhiều nhà cung cấp cho phép chúng ta xây dựng từ những server MQTT mức độ đơn giản cho đến phức tạp. Tùy vào cấu trúc của hệ thống và số lượng các kết nối, tần suất tin nhắn mà nhà cung cấp đề xuất cấu hình và mức giá khác nhau. Các nhà cung cấp điển hình như Microsoft Azure, Google Cloud, IBM Watson,....

Trong đề tài luận văn này chúng ta sẽ sử dụng MQTT server của IBM Watson vì ưu điểm là hệ thống băng thông lớn, giới hạn số lượng tin nhắn trong một ngày lớn (400.000 tin/ ngày so với 8000 tin/ngày của Microsoft Azure - ở gói Free). Giả sử ta có 1 thiết bị, 1 giây gửi 1 tin thì tổng cộng 1 giờ sẽ có 3600 tin và 1 ngày có 86400 tin, lớn hơn giới hạn 8000 tin/ngày của Microsoft Azure.



Hình 2.7: IBM Watson MQTT