



Học viện
Công nghệ Bưu chính Viễn thông

Khoa Công nghệ thông tin 1

Thiết kế hệ thống IoT

Giảng viên: TS. Lê Văn Vịnh

Email: vinhlv@ptit.edu.vn

Chương 2: Thiết kế hệ thống IoT

1. Thiết kế vật lý của hệ thống IoT
2. Các giao thức truyền thông IoT
3. Thiết kế logic của hệ thống IoT
4. Thiết kế hệ thống IoT

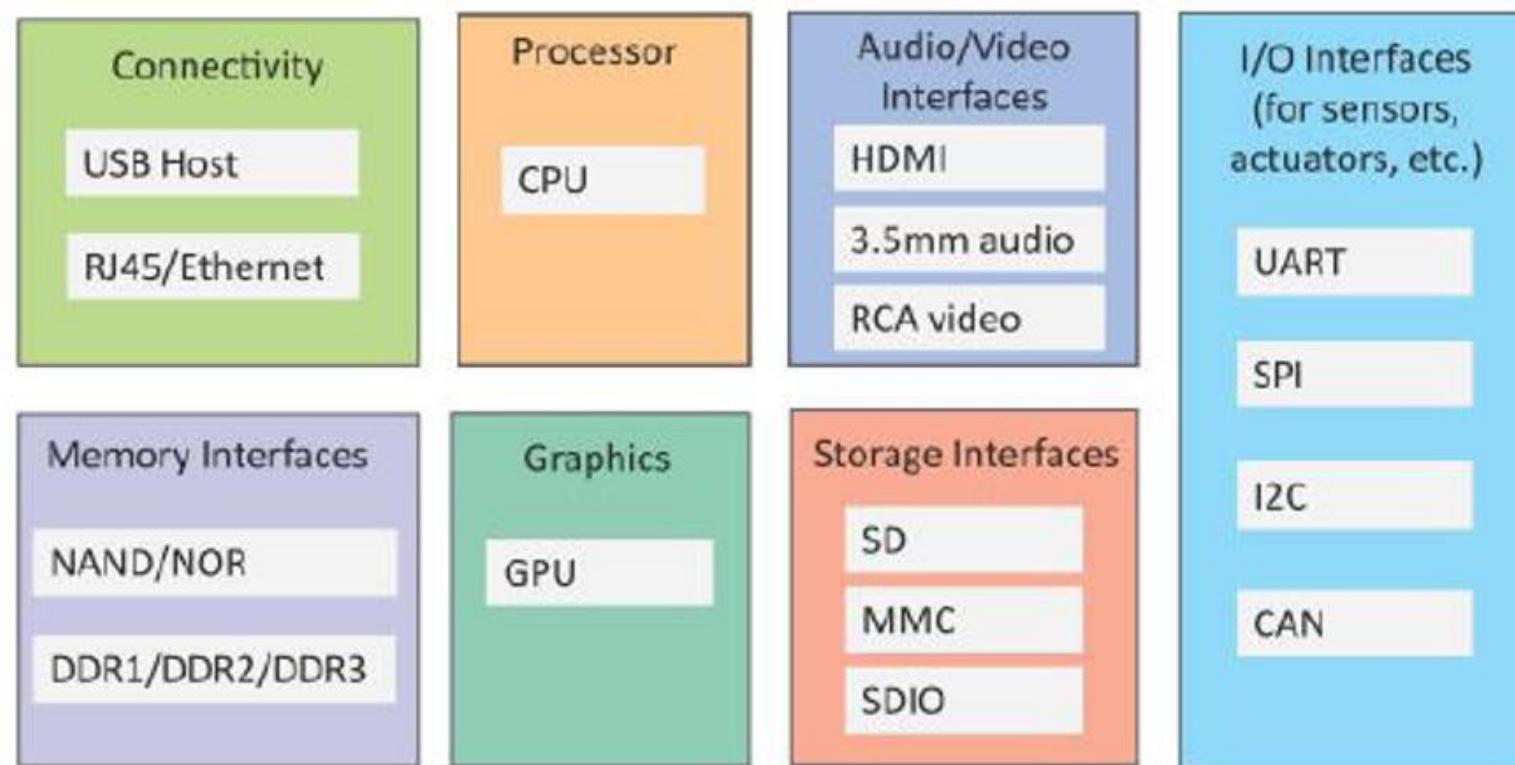
Thiết kế vật lý của IoT

Từ “vạn vật” trong IoT thường được biết tới là các thiết bị IoT, chúng có định danh duy nhất và có thể thực hiện các công việc cảm nhận (sensing), chấp hành (actuator) và giám sát (monitoring).

Các thiết bị IoT có thể

- Gửi dữ liệu đến các máy chủ tập trung hoặc phần back-end của ứng dụng dựa trên đám mây để xử lý dữ liệu.
- Thực hiện một số tác vụ cục bộ và các tác vụ khác trong cơ sở hạ tầng IoT, dựa trên các hạn chế về thời gian và không gian.
- Trao đổi dữ liệu với các thiết bị khác, các ứng dụng khác được kết nối cùng, một cách trực tiếp hoặc gián tiếp.
- Thu thập dữ liệu từ các quá trình xử lý dữ liệu cục bộ hoặc từ thiết bị khác.

Thiết kế vật lý của IoT



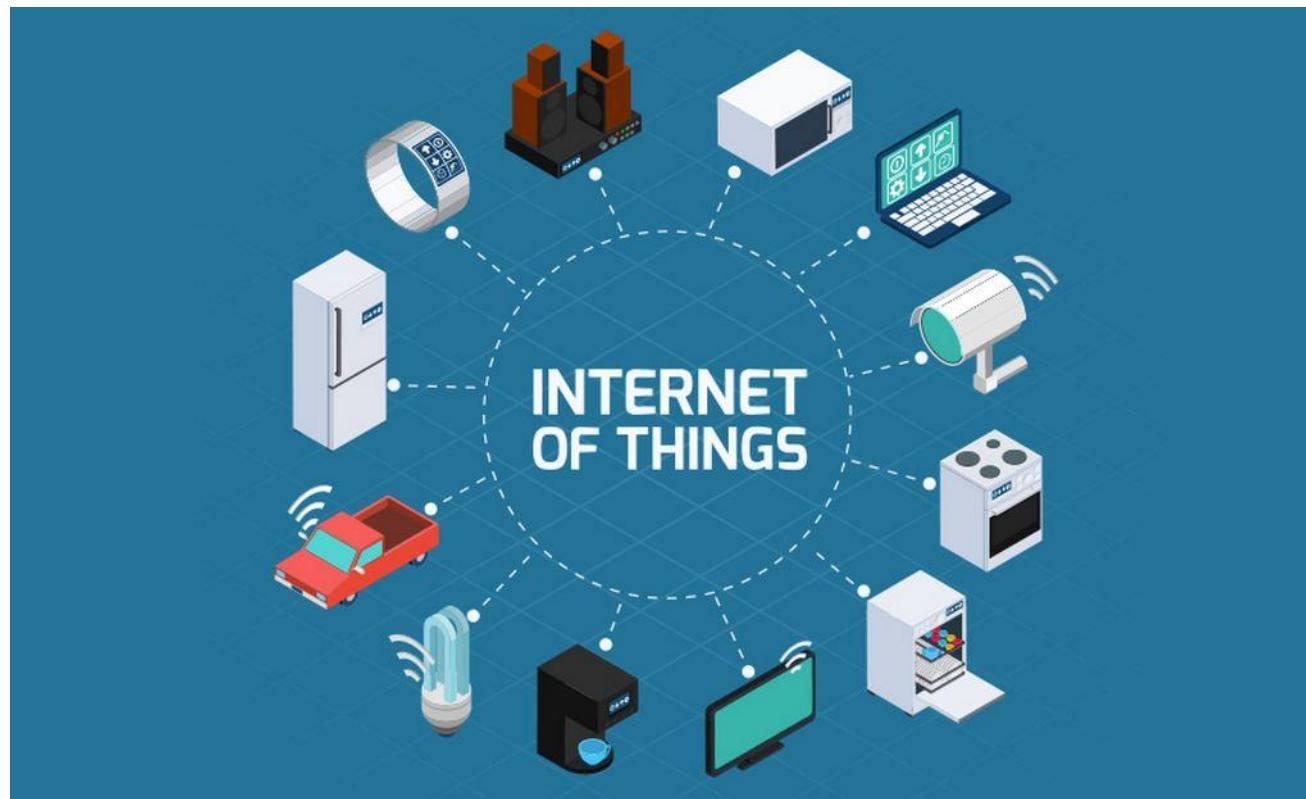
Sơ đồ khái niệm về thiết kế vật lý của một thiết bị IoT

Thiết kế vật lý của IoT

- ❖ Một thiết bị IoT có thể bao gồm một số giao diện để kết nối với các thiết bị khác, cả có dây và không dây
 - Các giao diện I/O cho các cảm biến
 - Các giao diện cho việc kết nối Internet (internet connectivity)
 - Các giao diện với thiết bị lưu trữ và bộ nhớ (memory & storage)
 - Các giao diện âm thanh và hình ảnh (audio & video)

Thiết kế vật lý của IoT

Các loại thiết bị IoT khác nhau



Thiết kế vật lý của IoT

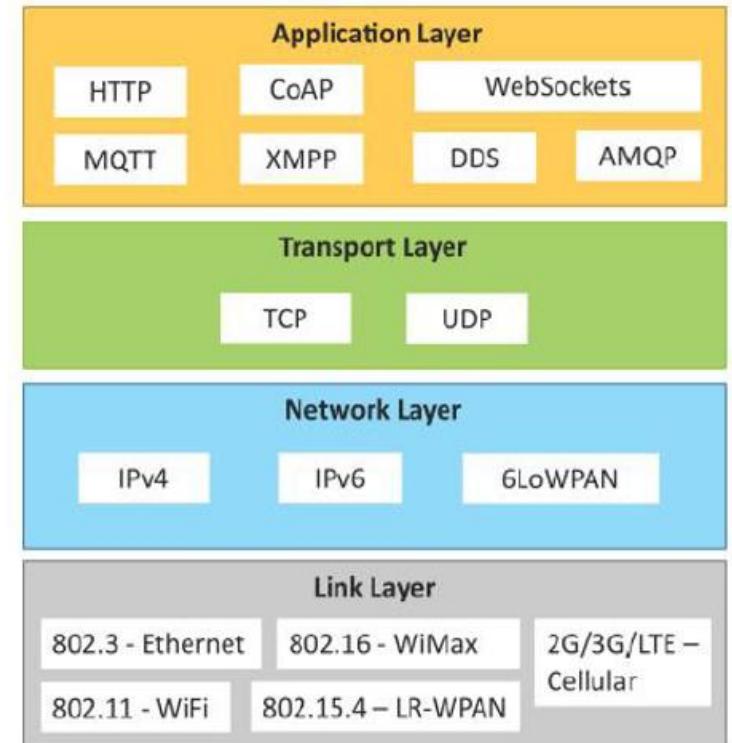
Các loại thiết bị IoT khác nhau

- Các thiết bị gia dụng
- Điện thoại thông minh & máy tính
- Các thiết bị điện tử cầm tay (hoặc đeo được-Wearable)
- Ô tô
- Các hệ thống năng lượng
- Hệ thống thanh toán bán lẻ
- Các loại máy in
- Các máy móc công nghiệp, nông nghiệp
- Hệ thống chăm sóc sức khỏe
- Hệ thống giám sát
- *Phụ thuộc vào sự sáng tạo của chúng ta !*

Thiết kế vật lý của IoT

Các giao thức IoT

- Link Layer
 - 802.3 – Ethernet
 - 802.11 – WiFi
 - 802.16 – WiMax
 - 802.15.4 – LR-WPAN
 - 2G/3G/4G
- Network/Internet Layer
 - IPv4
 - IPv6
 - 6LoWPAN
- Transport Layer
 - TCP
 - UDP
- Application Layer
 - HTTP
 - CoAP
 - WebSocket
 - MQTT
 - XMPP
 - DDS
 - AMQP



Các giao thức IoT

Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp mạng (Network/Internet Layer)
 - **IPv4** phổ biến nhất sử dụng để định danh các thiết bị trong mạng IoT. Sử dụng lược đồ địa chỉ phân cấp với 32 bit ($2^{32} \sim 4.294.967.296$ địa chỉ). IP truyền các gói tin (packets)
 - **IPv6** dùng 128 bit $\rightarrow 3,4 \times 10^{38}$ địa chỉ
 - **6LoWPAN** – (IPv6 over Low power Wireless Personal Area Networks). Dùng cho các thiết bị định danh bởi IPv6 có công suất nhỏ. Tần số vô tuyến 2,4GHz, tốc độ truyền dữ liệu 250Kb/s.

Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp giao vận (Transport Layer)
 - **TCP** (Transmission Control Protocol) giao thức phổ biến nhất, sử dụng bởi Web browser (lớp ứng dụng HTTP, HTTPS), các chương trình gửi thư điện tử (giao thức lớp ứng dụng SMTP), truyền file (FTP). Có khả năng phát hiện lỗi, chống tắc nghẽn, điều khiển luồng dữ liệu.
 - **UDP** (User Datagram Protocol) – gửi những dữ liệu ngắn **datagram**. UDP không cung cấp sự tin cậy, thứ tự truyền nhận; UDP nhanh và hiệu quả hơn đối với các mục tiêu như kích thước nhỏ và yêu cầu khắt khe về thời gian. Bản chất không trạng thái nên hữu dụng đối với việc trả lời các truy vấn nhỏ với số lượng lớn yêu cầu. Những ứng dụng phổ biến sử dụng UDP: DNS (Domain Name System), streaming media, Voice over IP, Trivial File Transfer Protocol (TFTP) và game trực tuyến.

Thiết kế vật lý của IoT

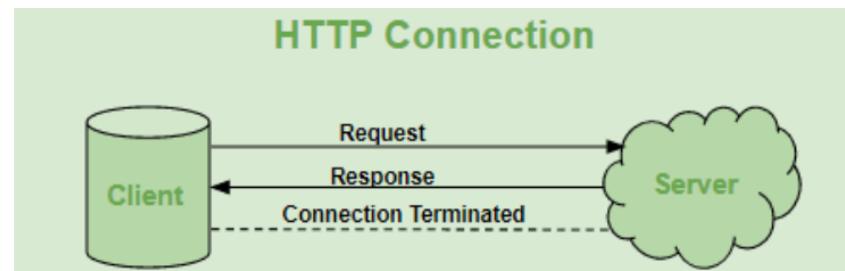
Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - **HTTP** – Hypertext Transfer Protocol
 - **CoAP** – Constrained Application Protocol
 - **WebSocket**
 - **MQTT** - Message Queue Telemetry Transport
 - **XMPP** – Extensible Messaging & Presence Protocol
 - **DDS** – Data Distribution Service
 - **AMQP** – Advanced Message Queuing Protocol

Thiết kế vật lý của IoT

Các giao thức IoT

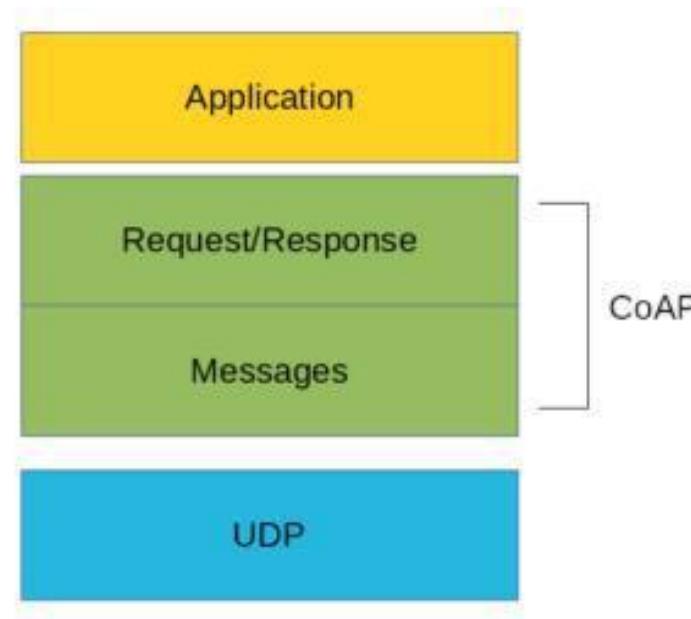
- Lớp Ứng dụng (Application Layer)
 - **HTTP** – Giao thức siêu văn bản (Hypertext Transfer Protocol)
 - Là giao thức lớp ứng dụng hình thành nên **World Wide Web** (WWW)
 - Hoạt động theo mô hình **Request-Response** trong đó client gửi request tới server sử dụng các lệnh HTTP . VD: GET, PUT, POST, DELETE, TRACE...
 - HTTP là giao thức **phi trạng thái** (stateless) và các request độc lập với nhau.
 - Giao thức HTTP sử dụng URI để định danh tài nguyên HTTP
 - HTTP client có thể là Browser hoặc App chạy trên máy khách
 - Được mô tả trong chuẩn RFC 2616



Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - CoAP – Giao thức ứng dụng bị ràng buộc (Constrained Application Protocol)



Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - CoAP – Giao thức ứng dụng bị ràng buộc (Constrained Application Protocol)
 - Được dùng cho các ứng dụng M2M (Machine to Machine)
 - Sử dụng mô hình Request-Response và là giao thức truyền web
 - Chạy trên đỉnh UDP thay vì TCP
 - Truyền các datagram theo phương thức phi kết nối giữa server và client
 - CoAP được thiết kế để giao tiếp dễ dàng với HTTP
 - Hỗ trợ các phương thức như GET, POST, PUT và DELETE
 - Được thiết kế để sử dụng trong các thiết bị internet hạn chế tài nguyên, chẳng hạn như các nút mạng cảm biến không dây.
 - Định nghĩa trong RFC 7252

Thiết kế vật lý của IoT

Các giao thức IoT

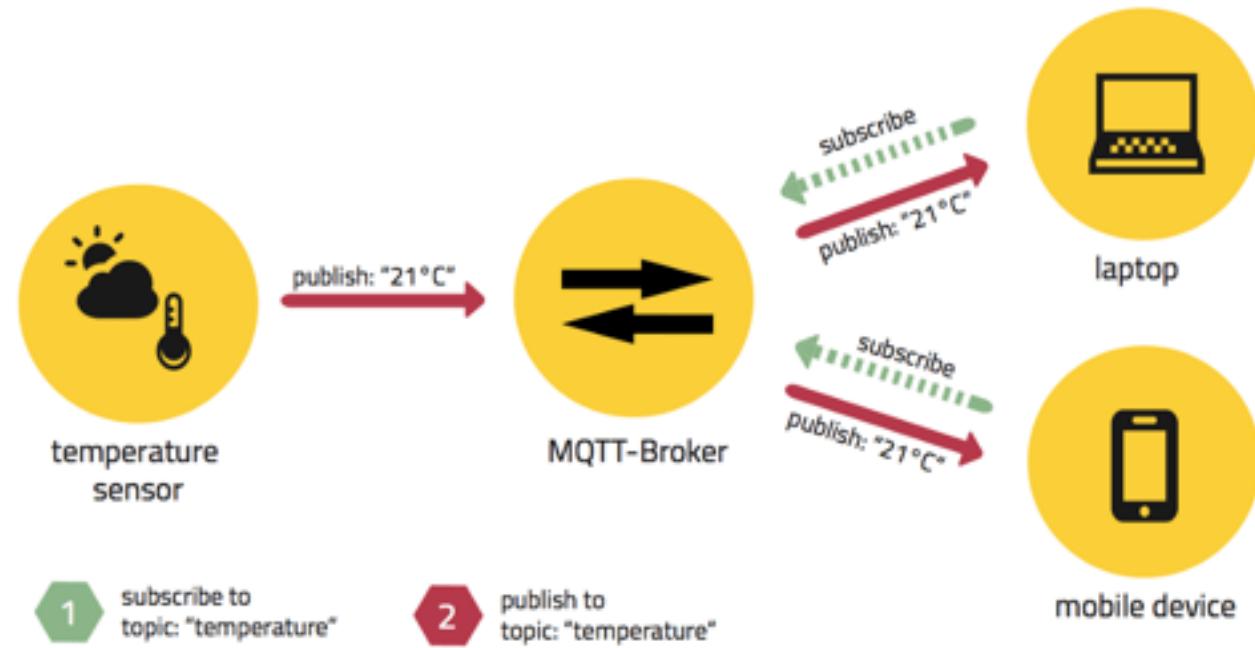
- Lớp Ứng dụng (Application Layer)
 - **WebSocket**
 - Truyền thông tin song công full-duplex trên một **kết nối TCP** duy nhất được **duy trì** để gửi bản tin giữa client và server.
 - Client có thể là browser, ứng dụng di động hoặc các thiết bị IoT.
 - Websocket **nhanh hơn** và cải thiện hiệu suất ứng dụng
 - Được dùng trong các ứng dụng **web thời gian thực, chat, game online**



1.4 Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - MQTT (Message Queue Telemetry Transport)



Thiết kế vật lý của IoT

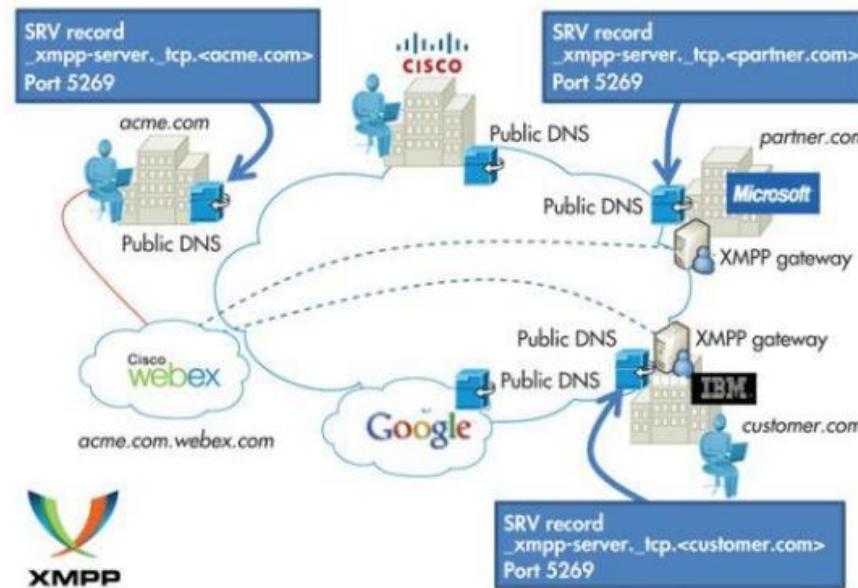
Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - **MQTT** (Message Queue Telemetry Transport)
 - MQTT là một giao thức **nhắn tin** theo tiêu chuẩn OASIS cho IoT.
 - Nó được sử dụng như một phương tiện truyền tải tin nhắn dạng xuất bản (**Publish**)/đăng ký (**Subscrible**) rất nhẹ.
 - MQTT rất thích hợp để kết nối các thiết bị từ xa với **băng thông mạng thấp** và các thiết bị **giới hạn bộ nhớ và khả năng xử lý**.
 - Một MQTT broker là một máy chủ (server) nhận tất cả các tin nhắn (message) từ các client và sau đó định tuyến (route) các tin nhắn này đến các client phù hợp.
 - Một MQTT client là bất kỳ thiết bị nào (từ vi điều khiển đến một máy chủ chính thức) chạy thư viện MQTT và kết nối tới MQTT broker qua mạng.
 - Quá trình hoạt động của MQTT được chia thành **4 giai đoạn**. Đó là **kết nối, xác thực, giao tiếp và kết thúc**.

Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - XMPP – Giao thức nhắn tin mở rộng và thông tin hiện diện trực tuyến (Extensible Messaging & Presence Protocol)



Thiết kế vật lý của IoT

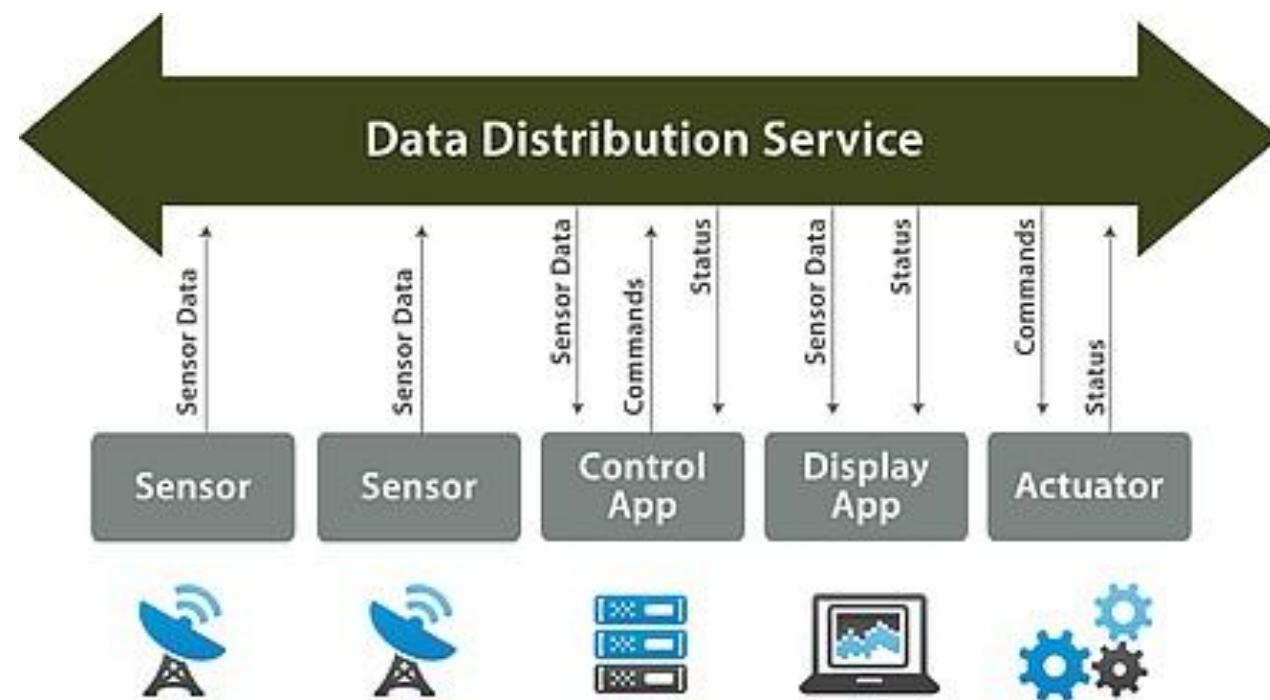
Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - XMPP – Extensible Messaging & Presence Protocol
 - XMPP (trước đây gọi là “Jabber”) là giao thức truyền thông dùng cho định hướng **tin nhắn** trung gian dựa trên ngôn ngữ **XML**.
 - XMPP là mô hình phân quyền **client-server** phi tập trung
 - XMPP cho phép gửi 1 đoạn XML từ một thực thể trên mạng tới một thực thể khác theo thời gian thực (**Real Time**). Trong hệ thống IoT XMPP cho phép trao đổi thời gian thực giữa các thiết bị IoT.
 - XMPP hỗ trợ truyền thông cả **client to server** và **server to server**
 - Ứng dụng trong gửi **tin nhắn**, **dữ liệu trực tuyến**, **cung cấp dữ liệu**, **game**, **chat đa thành viên** và hỗ trợ cuộc **gọi thoại/video**.
 - Được định nghĩa trong **RFC 6120**

Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - DDS – Data Distribution Service



Thiết kế vật lý của IoT

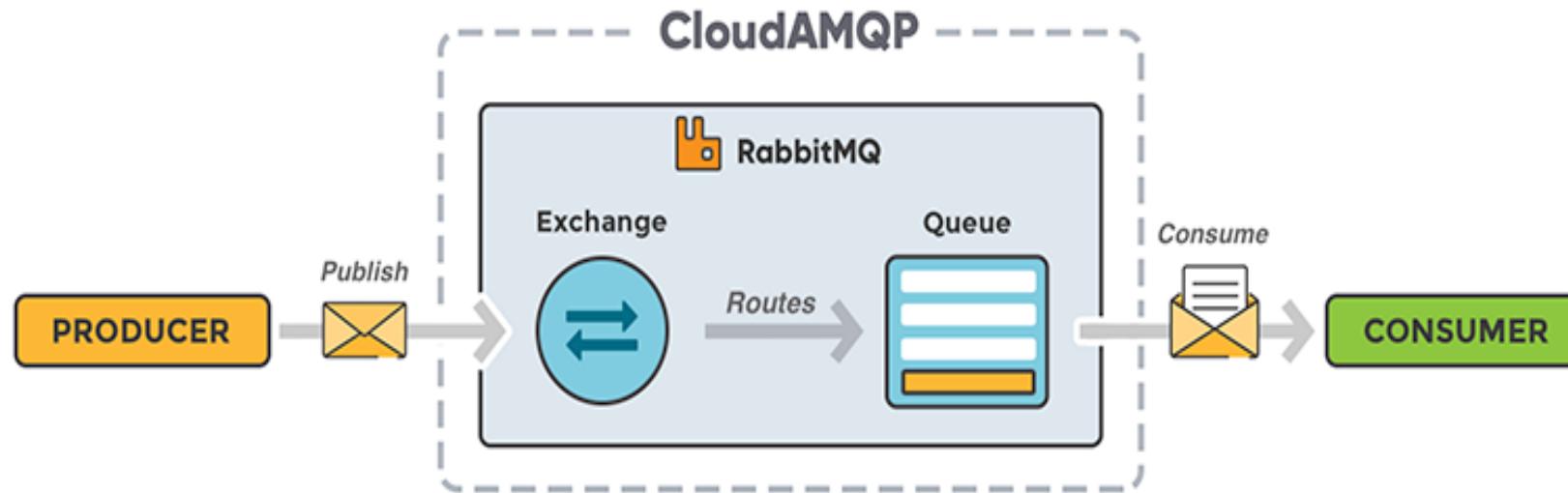
Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - DDS – Dịch vụ phân bổ dữ liệu (Data Distribution Service)
 - Là một phần mềm trung gian (middleware) dùng cho việc trao đổi thông tin giữa thiết bị tới thiết bị (**device to device**), máy tới máy (**machine to machine**).
 - DDS gửi và nhận dữ liệu, sự kiện, và thông tin lệnh **trên UDP** nhưng cũng có thể chạy trên các giao thức truyền tải khác như IP Multicast, TCP / IP, bộ nhớ chia sẻ ...
 - DDS sử dụng mô hình **Publish-Subscribe** trong đó Publisher (ex: thiết bị tạo ra dữ liệu) tạo các topic với các Subscribe (ex: thiết bị muốn sử dụng dữ liệu).
 - **Publisher** là một đối tượng tương ứng cho việc **phân phối dữ liệu** còn **Subscribe** là đối tượng tương ứng cho việc **nhận dữ liệu** được publish.
 - Có khả năng điều khiển chất lượng dịch vụ (**QoS**) và khả năng tự dò tìm

Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - AMQP – Advanced Message Queuing Protocol

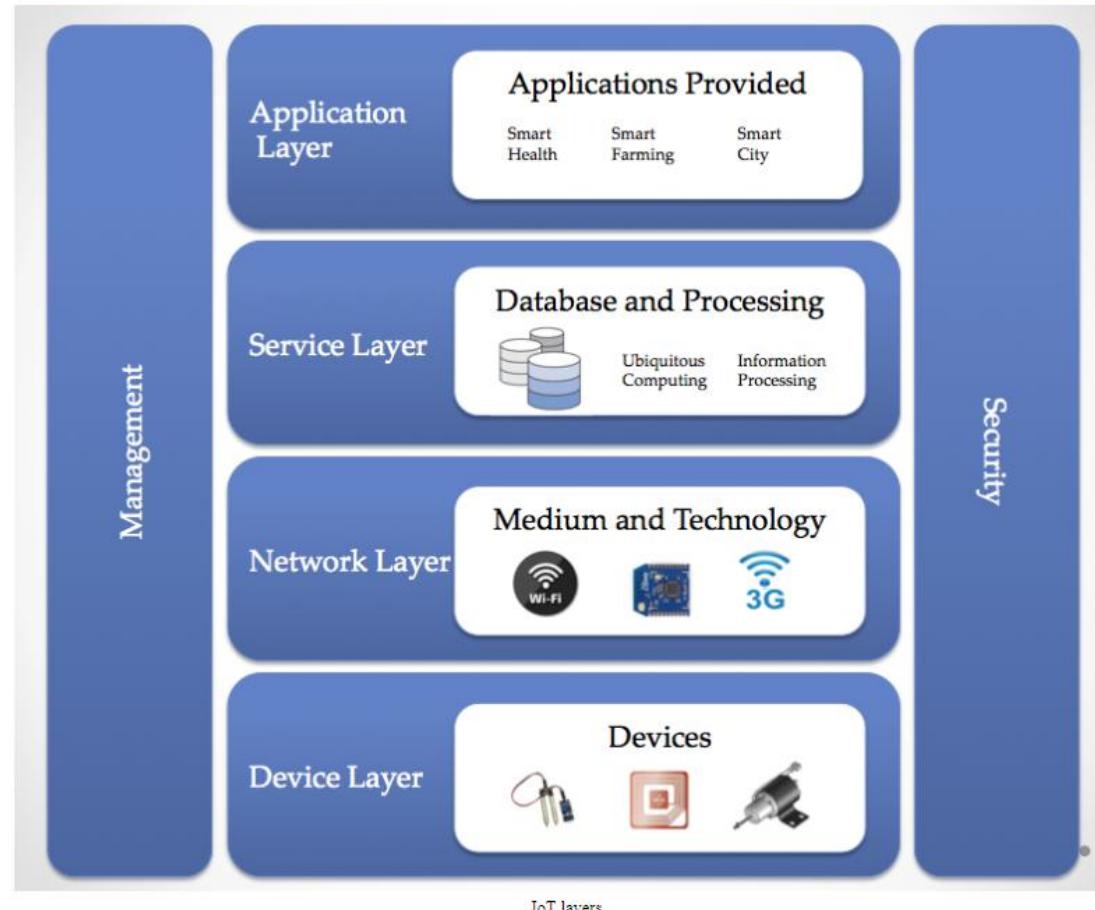


Thiết kế vật lý của IoT

Các giao thức IoT

- Lớp Ứng dụng (Application Layer)
 - **AMQP** – (Advanced Message Queuing Protocol)
 - Là một giao thức mở lớp ứng dụng, dùng để trao đổi các **bản tin**
 - AMQP hỗ trợ truyền thông theo các mô hình **Publisher-Subscribe**, **Point to Point**. Có khả năng **định tuyến và xếp hàng**.
 - Các AMQP brokers nhận các bản tin từ Publisher (ex: các thiết bị hoặc ứng dụng tạo ra dữ liệu) sau đó định tuyến chúng tới Consumer (các ứng dụng sẽ xử lý dữ liệu nhận được) thông qua các kết nối.
 - Các Publisher xuất bản các bản tin tới các Exchange sau đó các Exchange sẽ phân phối các bản copy của dữ liệu tới các hàng đợi.
 - Các bản tin cũng được chuyển đi bởi Broker tới Consumer đã được mô tả với hàng đợi. Hoặc các Consumer có thể lấy các bản tin từ hàng đợi.

Thiết kế logic của IoT



Thiết kế logic của IoT

- ❖ Thiết kế logic của một hệ thống IoT đề cập đến sự thể hiện trừu tượng của các thực thể và quy trình mà không đi sâu vào các đặc điểm cụ thể ở cấp độ thấp của việc triển khai.
- ❖ Hệ thống IoT bao gồm một số khối chức năng cung cấp cho hệ thống các khả năng nhận dạng (**identification**), cảm biến (**sensing**), hành động (**actuation**), trao đổi thông tin (**communication**) và quản lý (**management**).

Thiết kế logic của IoT

❖ Lớp Thiết bị (**Device Layer**)

- Một hệ thống IoT bao gồm các thiết bị cung cấp khả năng cảm nhận, chấp hành, giám sát và điều khiển. VD: các cảm biến, các động cơ, relay, đèn, quạt, máy lạnh,..vv

❖ Lớp Trao đổi thông tin (**Communication Layer**)

- Lớp này điều khiển việc trao đổi thông tin của hệ thống IoT.

❖ Lớp Dịch vụ (**Service Layer**)

- Lớp này cung cấp các dịch vụ cho hệ thống IoT: giám sát, điều khiển, xuất bản dữ liệu, tự động dò tìm,...

Thiết kế logic của IoT

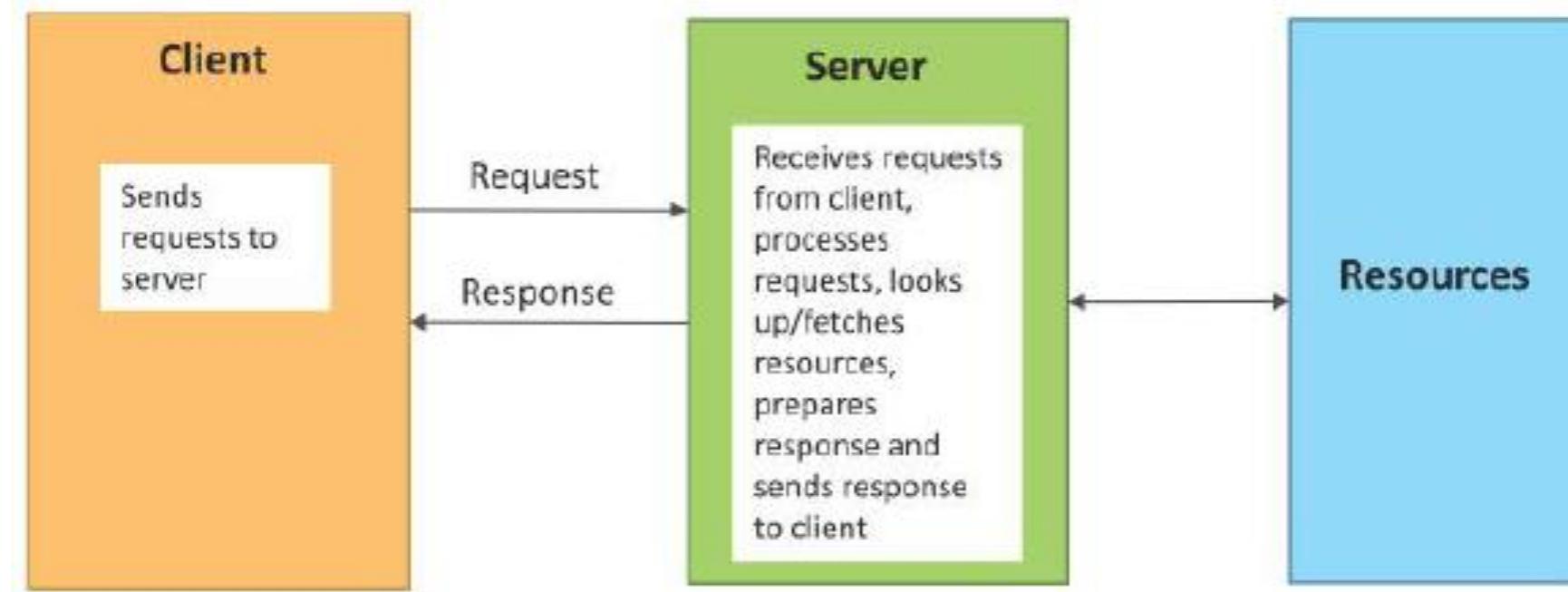
- ❖ Lớp Ứng dụng (**Application Layer**)
 - Lớp này cung cấp các giao diện để người dùng có thể sử dụng nhằm điều khiển và giám sát rất nhiều khía cạnh khác nhau của hệ thống IoT. Lớp ứng dụng cũng cho phép người dùng xem xét trạng thái của hệ thống, phân tích và xử lý dữ liệu.
- ❖ Chức năng Quản lý (**Management**)
 - Cung cấp các chức năng quản lý cho hệ thống IoT
- ❖ Chức năng Bảo mật (**Security**)
 - Cung cấp khả năng bảo mật cho hệ thống IoT. VD: xác thực, ủy quyền, tích hợp nội dung và bản tin, bảo mật dữ liệu,...

Một số mô hình trao đổi thông tin IoT

- ❖ Mô hình Request – Response
- ❖ Mô hình Publish – Subscribe
- ❖ Mô hình Push – Pull
- ❖ Mô hình Exclusive Pair
- ❖ Mô hình REST APIs
- ❖ Mô hình WebSocket APIs

Một số mô hình trao đổi thông tin IoT

❖ Mô hình Request – Response



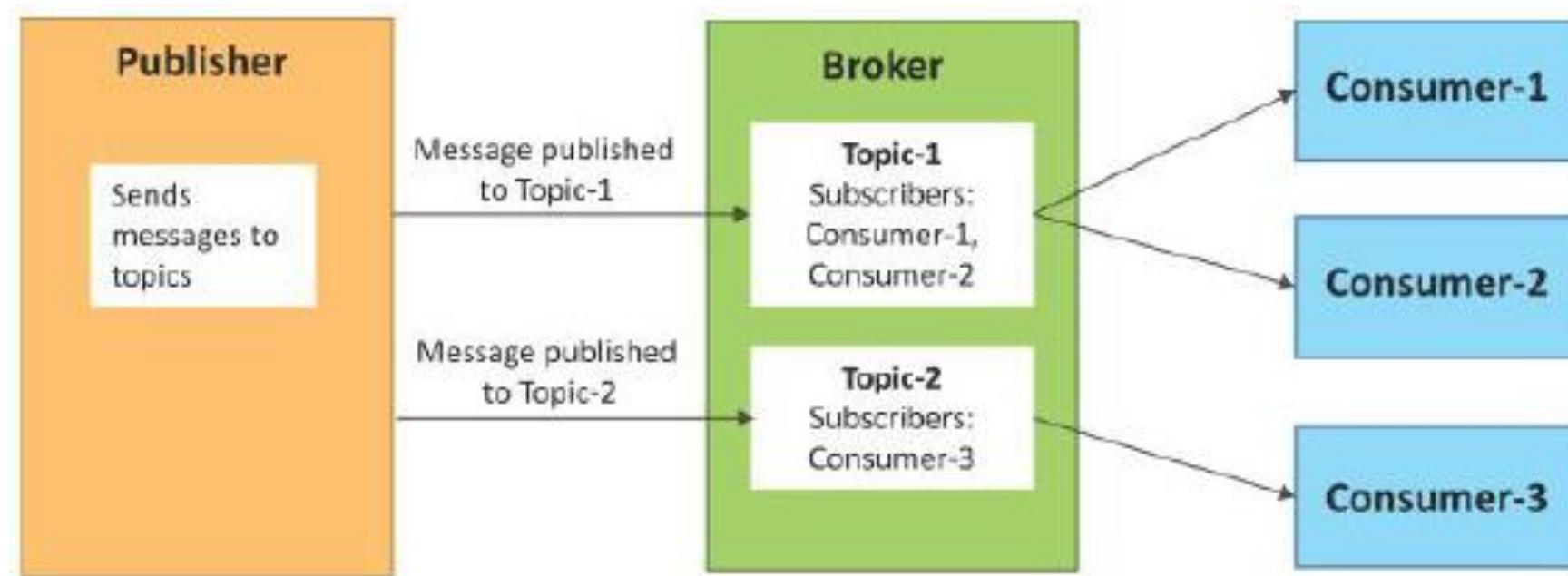
Một số mô hình trao đổi thông tin IoT

❖ Mô hình Request – Response

- Request-Response là một mô hình trao đổi thông tin trong đó (máy khách) client gửi yêu cầu (request) tới (máy chủ) server và server phản hồi lại những yêu cầu đó.
- Khi server nhận được một yêu cầu nó sẽ quyết định cách thức phản hồi, nhận dữ liệu, truy xuất tài nguyên, chuẩn bị phản hồi, và sau đó gửi phản hồi tới client.

Một số mô hình trao đổi thông tin IoT

❖ Mô hình Publish-Subscribe



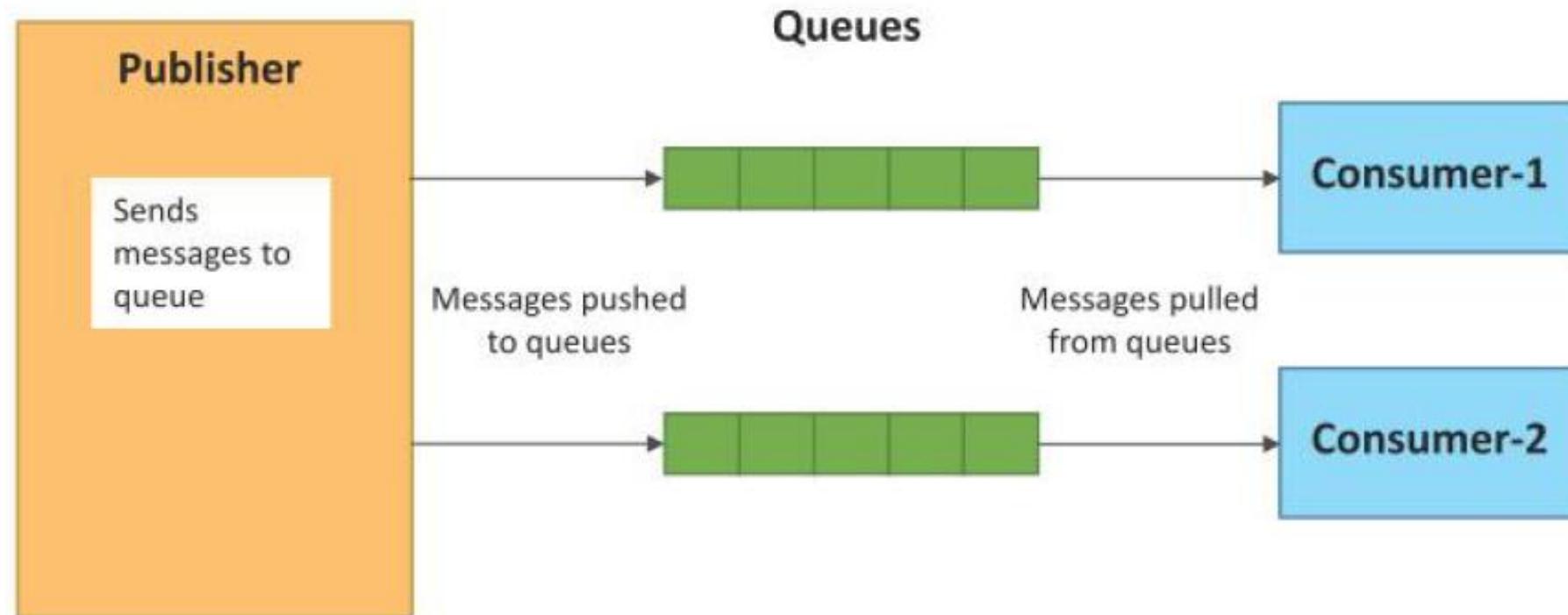
Một số mô hình trao đổi thông tin IoT

❖ Mô hình Publish-Subscribe

- Mô hình Publish-Subscribe là một mô hình trao đổi thông tin trong đó liên quan đến Publisher (người xuất bản), Broker (người môi giới) và Consumer (người tiêu dùng).
- Publisher là nguồn dữ liệu, từ đây dữ liệu được gửi tới các Topic. Các Topic được quản lý bởi các Broker. Publisher không biết về Consumer.
- Consumer đăng ký (subscribe) với các topic được quản lý bởi Broker.
- Khi Broker nhận dữ liệu cho một Topic từ Publisher, nó gửi dữ liệu này tới tất cả các Consumer đã đăng ký.

Một số mô hình trao đổi thông tin IoT

❖ Mô hình Push-Pull



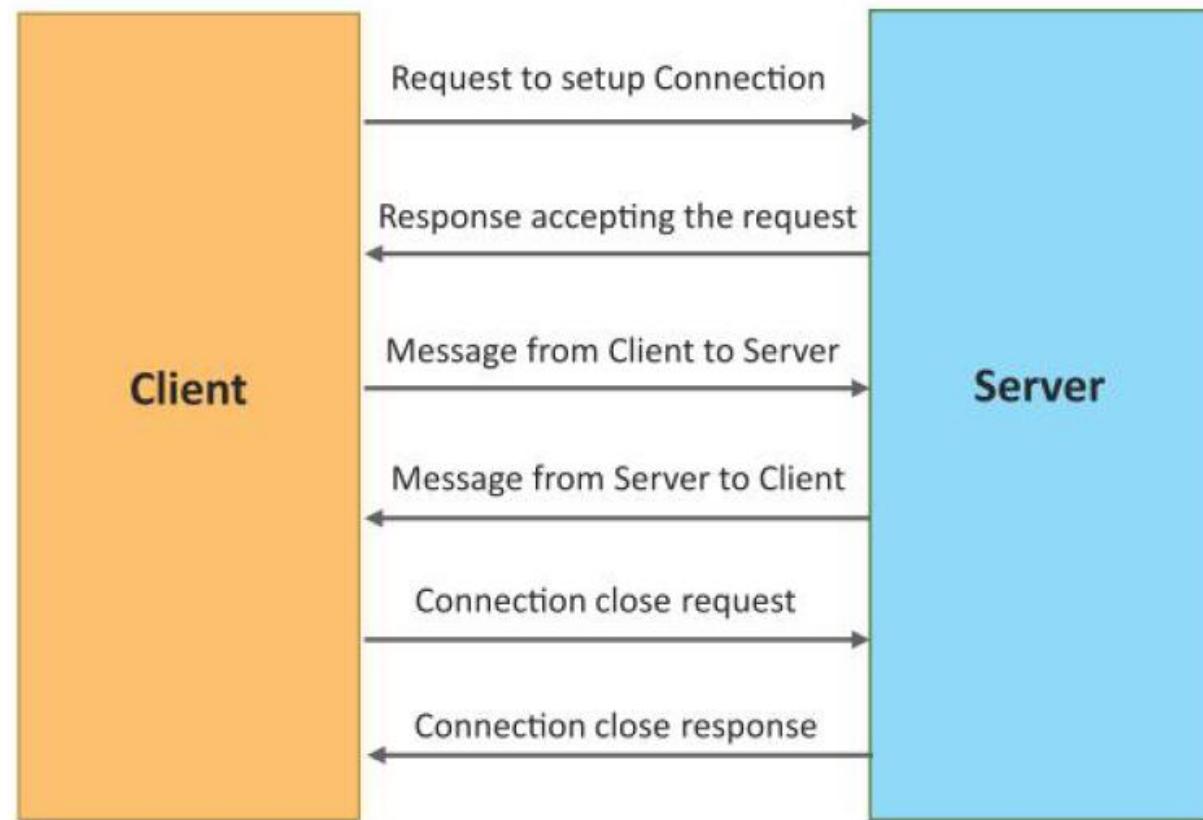
Một số mô hình trao đổi thông tin IoT

❖ Mô hình Push-Pull

- Mô hình Push-Pull là một mô hình trao đổi thông tin trong đó các nhà sản xuất dữ liệu (Data Producers) đẩy (Push) dữ liệu tới hàng đợi và Consumer sẽ kéo (Pull) dữ liệu về từ hàng đợi. Các nhà sản xuất dữ liệu (Producers) không cần biết về Consumer.
- Hàng đợi giúp phân tách thông điệp giữa các nhà sản xuất dữ liệu Producer và người tiêu dùng Consumer.
- Hàng đợi (Queue) cũng hoạt động như một bộ đệm và nó trợ giúp trong trường hợp khi có sai lệch giữa tốc độ đẩy dữ liệu lên của Publisher và kéo dữ liệu về của Consumer.

Một số mô hình trao đổi thông tin IoT

❖ Mô hình Exclusive Pair



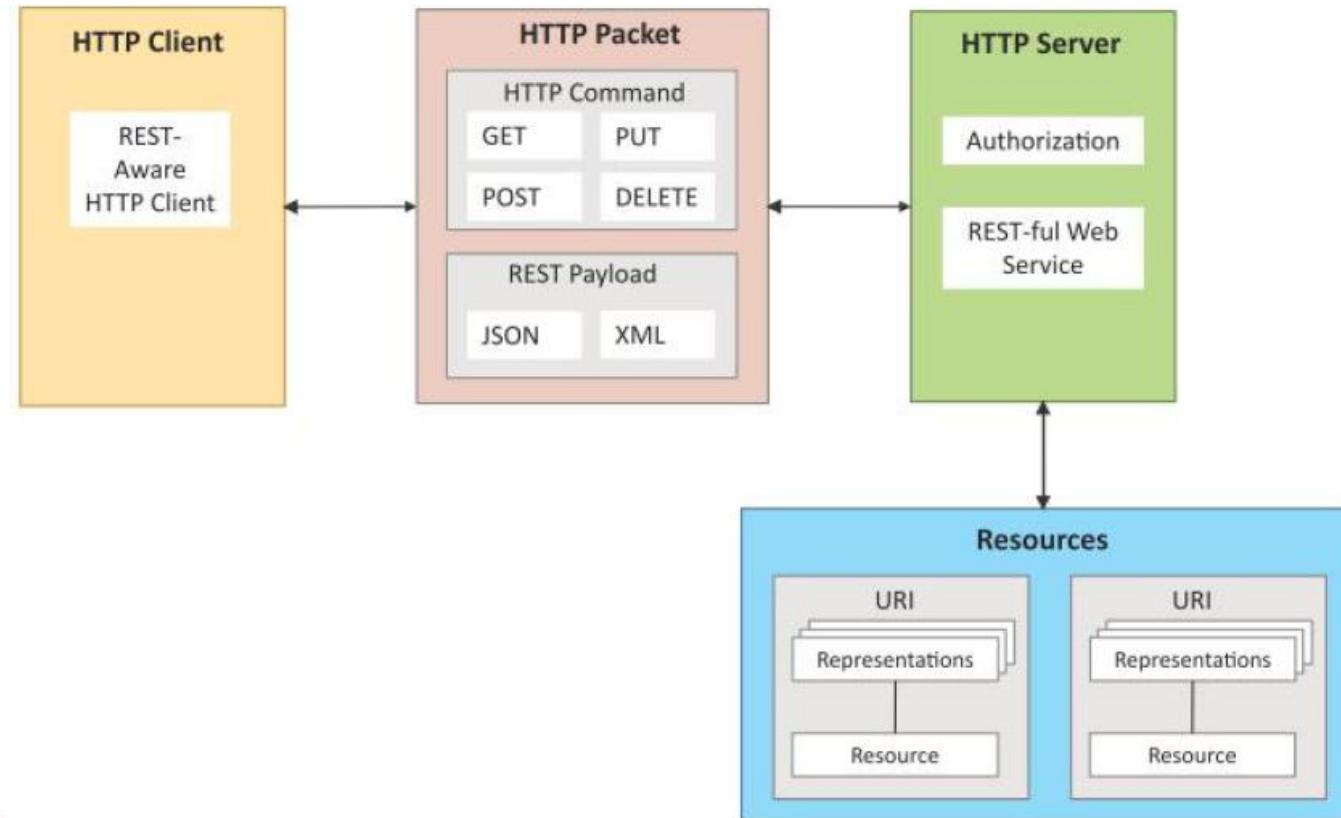
Một số mô hình trao đổi thông tin IoT

❖ Mô hình Exclusive Pair

- Mô hình Exclusive Pair là một mô hình trao đổi thông tin hai hướng, song công, sử dụng kết nối duy trì liên tục giữa máy chủ và máy khách.
- Khi một kết nối được thiết lập, nó sẽ duy trì liên tục cho tới khi máy khách gửi tới máy chủ một yêu cầu đóng kết nối này lại.
- Máy chủ và máy khách có thể gửi tin cho nhau khi kết nối được thiết lập.

Một số mô hình trao đổi thông tin IoT

❖ Mô hình REST APIs



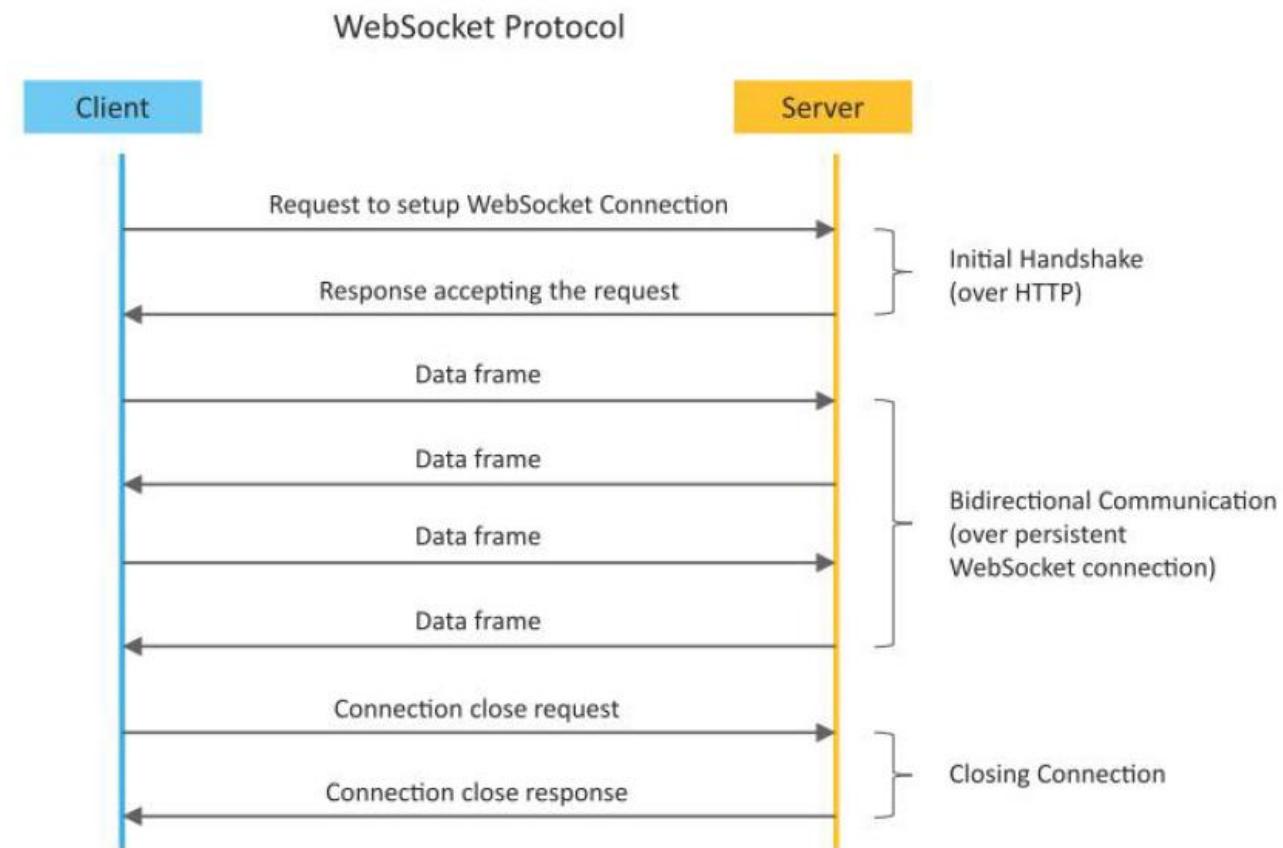
Một số mô hình trao đổi thông tin IoT

❖ Mô hình REST APIs

- REST- (Representational State Transfer - Chuyển giao trạng thái đại diện) là một tập hợp các nguyên tắc kiến trúc mà dựa vào đó chúng ta có thể thiết kế các dịch vụ web và Web API tập trung vào tài nguyên của hệ thống và cách giải quyết và chuyển trạng thái tài nguyên.
- REST APIs hoạt động theo mô hình trao đổi thông tin Request – Response.
- Các ràng buộc của kiến trúc REST áp dụng đối với các thành phần, các trình kết nối và các phần tử dữ liệu trong một hệ thống siêu phương tiện phân tán.

Một số mô hình trao đổi thông tin IoT

❖ Mô hình WebSocket APIs



Một số mô hình trao đổi thông tin IoT

❖ Mô hình WebSocket APIs

- WebSocket APIs cho phép truyền thông song công, hai hướng giữa các client và các server.
- Mô hình này tuân theo mô hình trao đổi thông tin Exclusive Pair

Các cấp độ phát triển IoT

Một hệ thống IoT thường bao gồm các thành phần sau:

- **Device** (thiết bị): Một thiết bị IoT cho phép định danh, nhận biết từ xa, chấp hành và giám sát từ xa. Có rất nhiều ví dụ đã biết ...
- **Resource** (tài nguyên): Tài nguyên là các thành phần phần mềm trên thiết bị IoT để truy cập, xử lý và lưu trữ thông tin cảm biến hoặc điều khiển các cơ cấu chấp hành được kết nối với thiết bị. Tài nguyên cũng bao gồm các thành phần phần mềm cho phép truy cập mạng cho thiết bị.

Các cấp độ phát triển IoT

Một hệ thống IoT thường bao gồm các thành phần sau:

- **Controller service** (bộ điều khiển dịch vụ): Dịch vụ bộ điều khiển gửi dữ liệu từ thiết bị đến dịch vụ web và nhận lệnh từ ứng dụng (qua các dịch vụ web) để bật các thành phần thiết bị cho phép truy cập mạng cho thiết bị.
- **Database** (cơ sở dữ liệu): cơ sở dữ liệu có thể là csdl cục bộ hoặc trên đám mây. Cơ sở dữ liệu lưu trữ các dữ liệu được tạo ra bởi các thiết bị IoT.

Các cấp độ phát triển IoT

Một hệ thống IoT thường bao gồm các thành phần sau:

- **Web service** (dịch vụ web): Các dịch vụ Web đóng vai trò như một liên kết giữa các thiết bị IoT, các ứng dụng, cơ sở dữ liệu và các thành phần phân tích. Dịch vụ web có thể được triển khai bằng cách sử dụng các nguyên tắc HTTP và REST (dịch vụ REST) hoặc sử dụng giao thức WebSocket (Dịch vụ WebSocket).
- **Analysis Component** (các thành phần phân tích): Các thành phần phân tích chịu trách nhiệm phân tích dữ liệu IoT và tạo ra kết quả ở dạng dễ hiểu cho người dùng.

Các cấp độ phát triển IoT

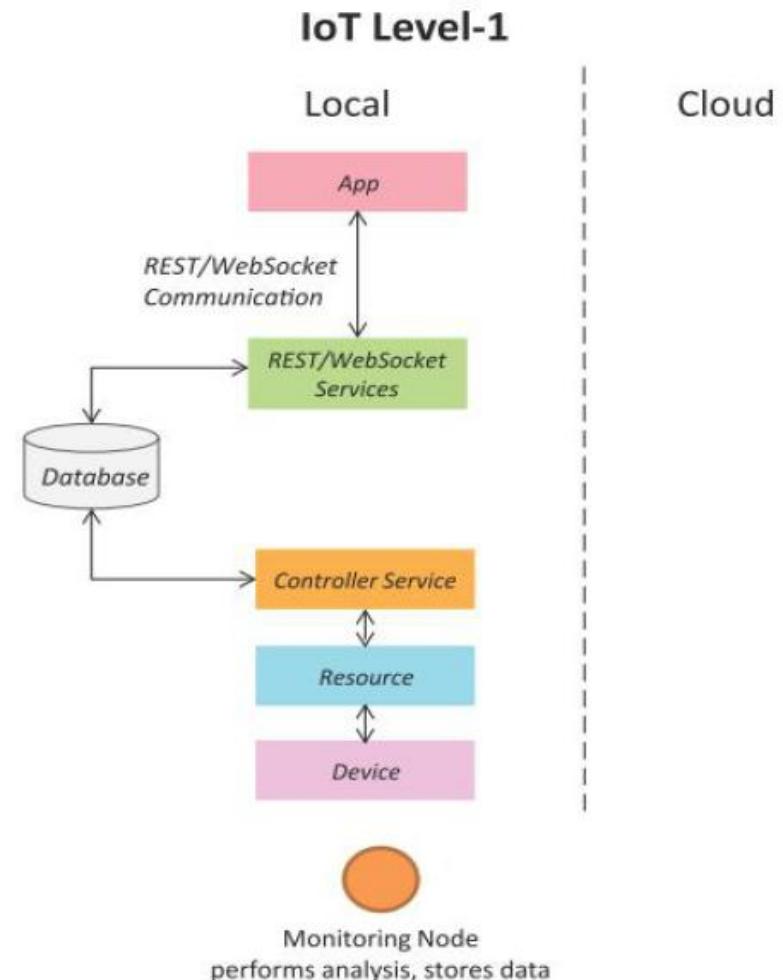
Một hệ thống IoT thường bao gồm các thành phần sau:

- **Application** (ứng dụng): Các ứng dụng IoT cung cấp một giao diện mà người dùng có thể sử dụng để điều khiển và giám sát các khía cạnh khác nhau của hệ thống IoT. Các ứng dụng cũng cho phép người dùng xem trạng thái hệ thống và xem dữ liệu đã xử lý.

Các cấp độ phát triển IoT

IoT cấp độ 1

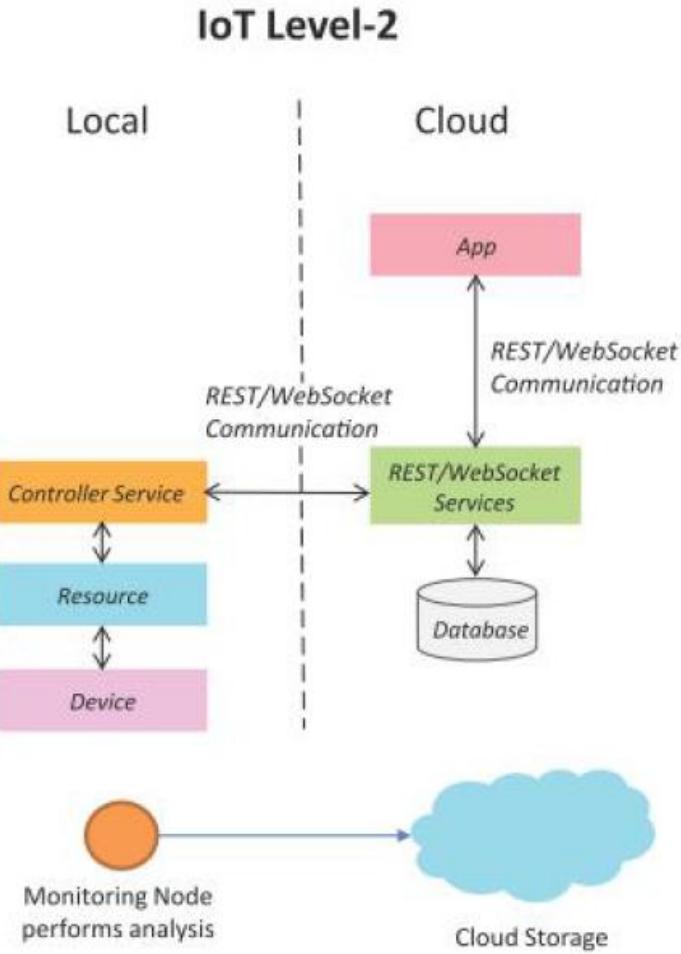
- Hệ thống IoT cấp độ 1 có một nút/thiết bị duy nhất thực hiện nhiệm vụ cảm biến và/hoặc hoạt động, lưu trữ dữ liệu, thực hiện phân tích và lưu trữ ứng dụng.
- Hệ thống IoT cấp độ 1 phù hợp cho mô hình IoT chi phí thấp và các giải pháp có độ phức tạp thấp, trong đó dữ liệu liên quan không lớn và các yêu cầu phân tích không chuyên sâu về mặt tính toán.



Các cấp độ phát triển IoT

IoT cấp độ 2

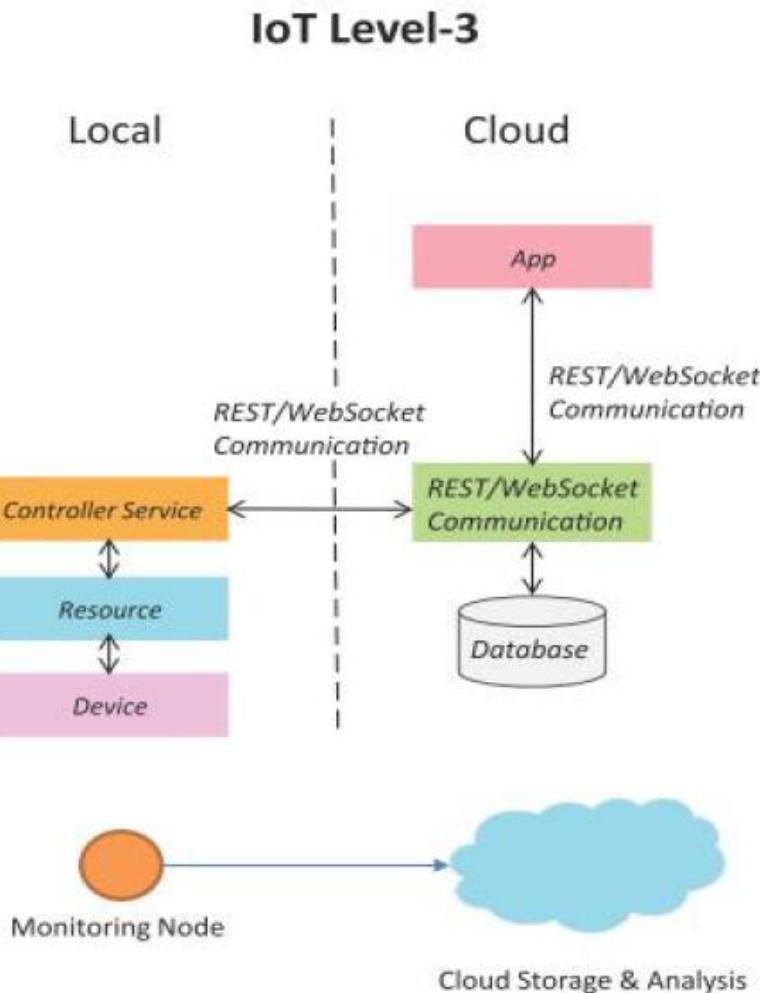
- Hệ thống IoT cấp độ 2 có một nút duy nhất thực hiện cảm biến và/hoặc hoạt động và phân tích cục bộ.
- Dữ liệu được lưu trữ trên đám mây và ứng dụng thường dựa trên đám mây.
- Hệ thống IoT cấp độ 2 phù hợp với các giải pháp có dữ liệu liên quan lớn, tuy nhiên, yêu cầu phân tích chính không chuyên sâu về mặt tính toán và có thể được thực hiện tại chỗ.



Các cấp độ phát triển IoT

IoT cấp độ 3

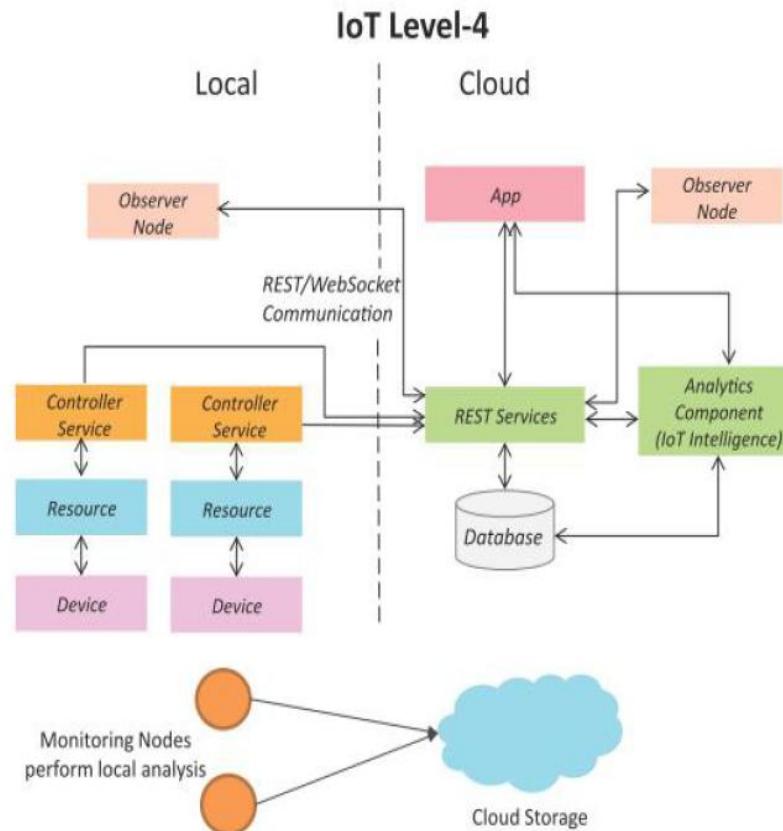
- Hệ thống IoT cấp độ 3 có một nút duy nhất. Dữ liệu được lưu trữ và phân tích trên đám mây và ứng dụng dựa trên đám mây.
- Hệ thống IoT cấp độ 3 phù hợp với các giải pháp mà dữ liệu liên quan lớn và các yêu cầu phân tích chuyên sâu về tính toán.



Các cấp độ phát triển IoT

IoT cấp độ 4

- Hệ thống IoT cấp 4 có nhiều nút thực hiện phân tích cục bộ. Dữ liệu được lưu trữ trên đám mây và ứng dụng dựa trên đám mây.
- Cấp độ 4 chứa các nút quan sát cục bộ và dựa trên đám mây, chúng có thể đăng ký và nhận thông tin được thu thập trên đám mây từ các thiết bị IoT.
- Hệ thống IoT cấp 4 phù hợp với các giải pháp yêu cầu nhiều nút, dữ liệu liên quan lớn và các yêu cầu phân tích chuyên sâu về tính toán.

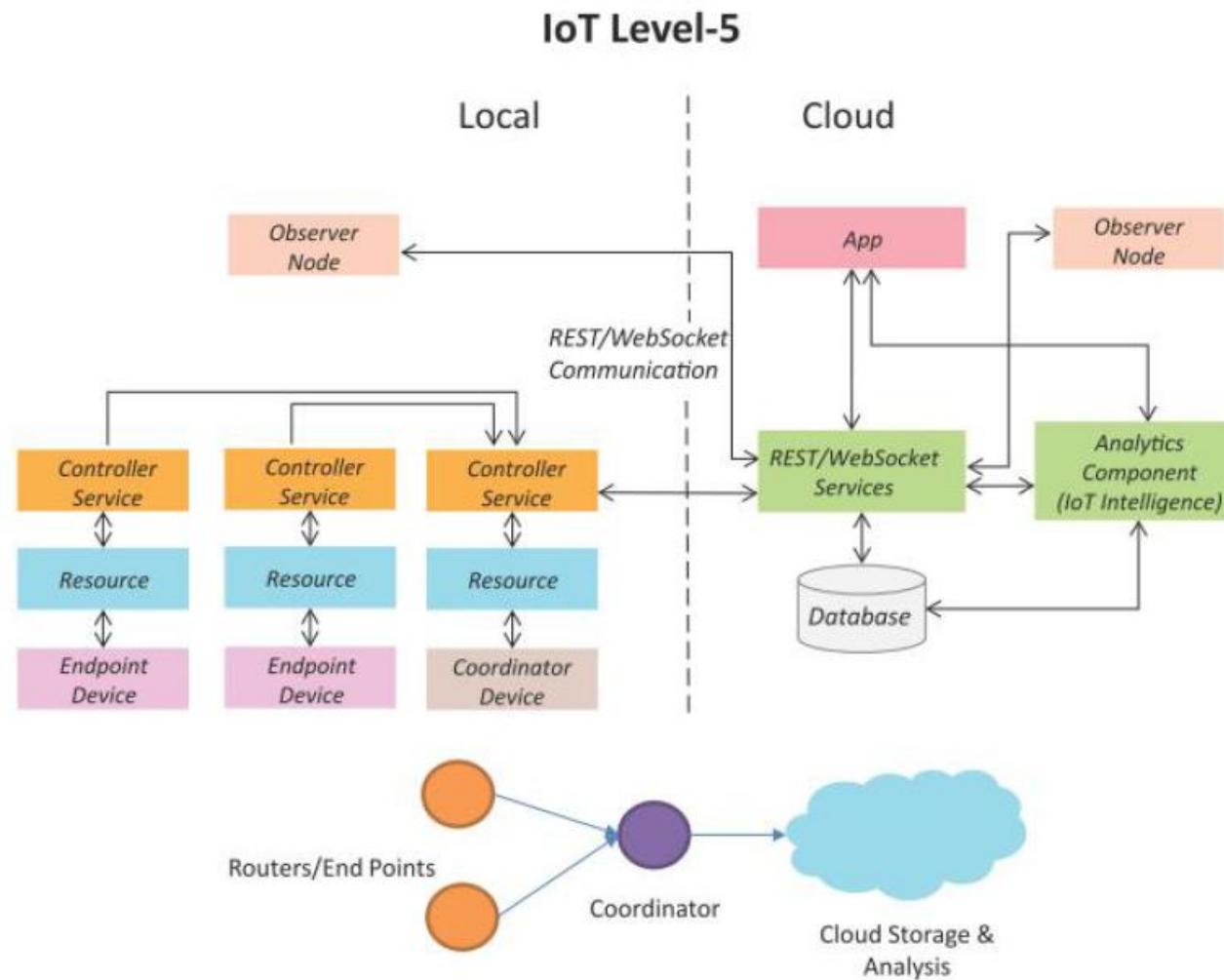


Các cấp độ phát triển IoT

IoT cấp độ 5

- Hệ thống IoT cấp độ 5 có nhiều node đầu cuối và một node điều phối (coordinator).
- Các node đầu cuối thực hiện việc cảm nhận và chấp hành.
- Các node điều phối thực hiện việc thu thập dữ liệu từ các node đầu cuối và gửi dữ liệu lên đám mây.
- Dữ liệu được lưu trữ và phân tích trên đám mây và các ứng dụng dựa trên đám mây.
- IoT cấp độ 5 phù hợp cho các giải pháp dựa trên mạng cảm biến vô tuyến, trong đó dữ liệu có liên quan là lớn và có các yêu cầu phân tích và tính toán chuyên sâu.

Các cấp độ phát triển IoT

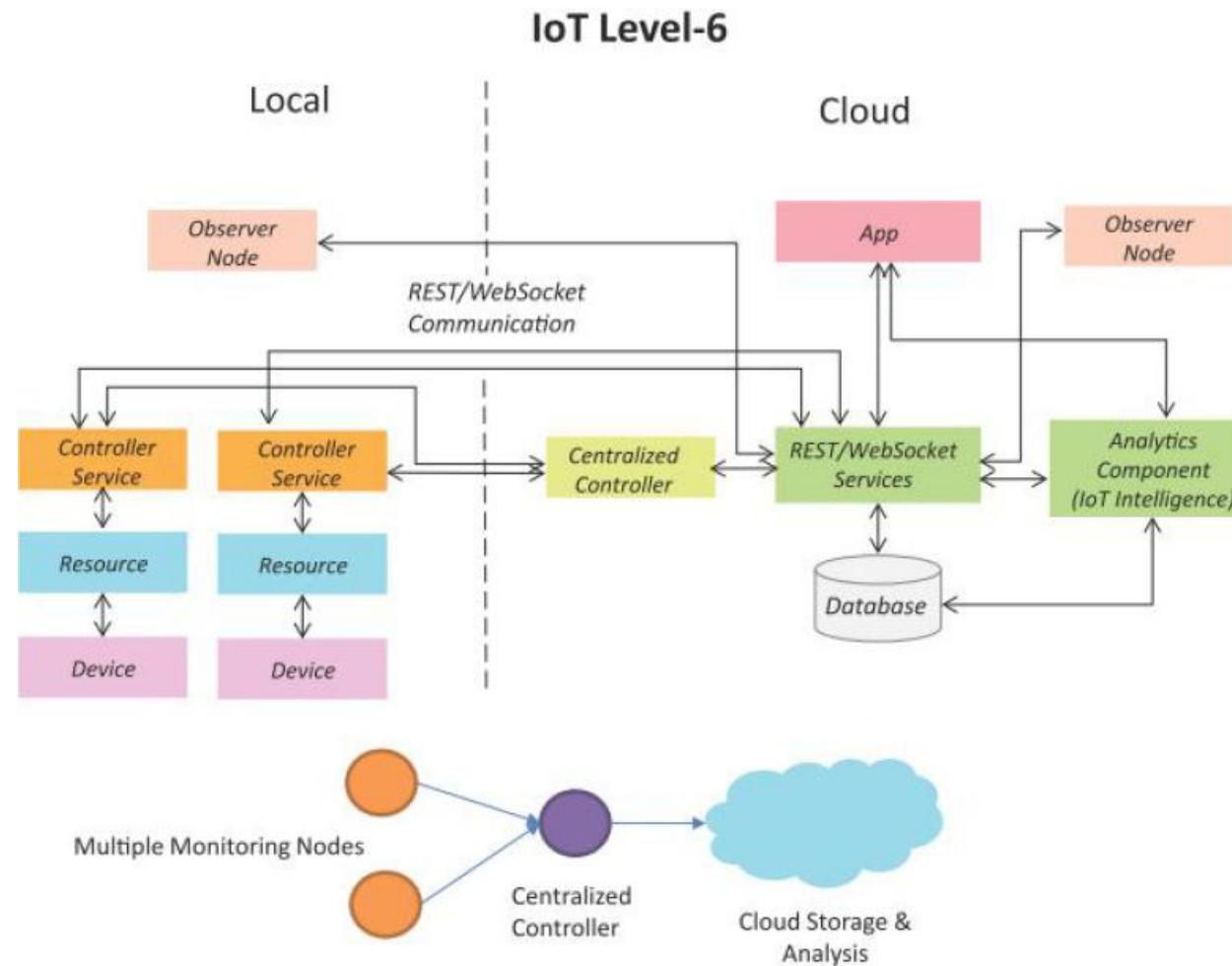


Các cấp độ phát triển IoT

IoT cấp độ 6

- Hệ thống IoT cấp độ 6 có nhiều node đầu cuối độc lập thực hiện việc cảm nhận, chấp hành và gửi dữ liệu lên đám mây.
- Dữ liệu được lưu trữ trên đám mây và ứng dụng dựa trên đám mây.
- Các thành phần trong hệ thống IoT làm nhiệm vụ phân tích sẽ phân tích dữ liệu và lưu trữ kết quả trên cơ sở dữ liệu đám mây.
- Các kết quả thường là trực quan với các ứng dụng của đám mây.
- Bộ điều khiển tập trung (centralized controller) được biết trạng thái của tất cả các node đầu cuối và có thể gửi các lệnh tới các node này.

Các cấp độ phát triển IoT





Thank you!