UT6 TFU – Grupo 1

Tema: Protocolos de intercambio de mensajes y MQTT

Introducción

Para los sistemas informáticos, la capacidad de interactuar confiablemente con otros sistemas siempre ha sido un aspecto relevante a tomar en cuenta en su diseño. En las últimas décadas, se ha vuelto frecuente el uso de arquitecturas distribuidas, c alejándose de los sistemas monolíticos de antaño. Estas arquitecturas traen consigo varios beneficios, como separación de responsabilidades, desacoplamiento, redundancia, tolerancia a fallos y alta disponibilidad, pero surge recurrentemente el problema de asegurar una comunicación eficaz, eficiente y confiable entre los distintos componentes de una solución informática. Además, distintos escenarios y problemas a resolver imponen distintas restricciones en cuanto al orden de trabajo de distintos componentes de software, la forma de comunicarse pasarse información entre sí, esperar por alguna señal o repartirse la carga de trabajo.

El objetivo de este informe es explorar los protocolos de encolamiento de mensajes, que surgen como una forma estandarizada de comunicar distintos componentes de software entre sí mediante mensajes asíncronos que permiten distribuir cargas de trabajo pesadas y desacoplar procesos de uso intensivo de recursos (AWS, s.f-b). Dentro de este grupo, existen distintos protocolos que responden a distintas características de un contexto, como el ancho de banda disponible, lógica de distribución de mensajes enviados, confiabilidad de la transmisión de mensajes o plataforma de ejecución de los clientes. Algunas de las opciones que existen hoy en día son MQTT, Kafka, IBM MQ y AMQP (Cleo, s.f). Para permitirnos profundizar en detalles de la implementación, nos enfocaremos en el protocolo MQTT, aunque eventualmente podremos hacer alusión a otros protocolos de la misma familia para destacar diferencias importantes.

¿Qué es MQTT?

El protocolo MQTT (*Message Queues Telemetry Transport*) es un protocolo ligero de tipo Publicación-Suscripción, para la comunicación máquina a máquina en red para el intercambio de mensajes mediante una cola. Fue inicialmente desarrollado por IBM por Andy Stanford-Clark y Arlen Nipper para el monitoreo de los oleoductos dentro del

sistema de control industrial SCADA y está diseñado para conexiones remotas de dispositivos ligeros en una misma red en donde los dispositivos tienen recursos de red limitados. Por la importancia que tiene para este protocolo una comunicación estable y tolerante a errores de red, está diseñado para funcionar sobre el protocolo TCP/IP, ya que este garantiza la entrega de mensajes completos incluso con la pérdida de paquetes de datos intermedios.

Por estas características, ha sido adoptado para diversos usos en la industria IoT. Es un protocolo sencillo de implementar, y admite el envío de mensajes tanto desde los dispositivos a la nube como al revés. Su uso está muy extendido actualmente, con varios lenguajes de programación populares que disponen de librerías para utilizarlo. Además, el alcance de las comunicaciones es muy fácilmente escalable, sin que eso implique una inversión mucho mayor en recursos para los dispositivos remotos (AWS, s.f-a).

Elementos del protocolo MQTT

A continuación, analizaremos qué elementos y actores se definen en el protocolo MQTT. Si bien algunos detalles de la implementación pueden variar de un protocolo de cola de mensajes a otro, en líneas generales los distintos elementos del protocolo MQTT pueden trasladarse a otros protocolos similares, como AMQP.

Agente MQTT

También conocido como broker MQTT, es un servidor central en la arquitectura que coordina el tráfico de mensajes de los clientes, recibiéndolos, identificando los clientes suscritos a esos mensajes y reenviándoselos. Además, se encarga de la autenticación y autorización de cada cliente que desea enviar o recibir mensajes, así como controlar los mensajes perdidos y las sesiones de clientes (AWS, s.f-a). Otros sistemas también pueden solicitar información al broker para realizar tareas de análisis posterior y monitoreo en tiempo real de la información intercambiada a través de ese servidor.

Tema MQTT

Se refiere a un agrupamiento de mensajes con significado similar, en el sentido de que los clientes que se suscriban a ese tema querrán recibir todos los mensajes que sean publicados allí. Normalmente, los temas se organizan con nombres que sean entendibles semánticamente para favorecer el desacoplamiento entre clientes, ya que suele ser más importante el contenido del mensaje que su remitente.

Cliente MQTT

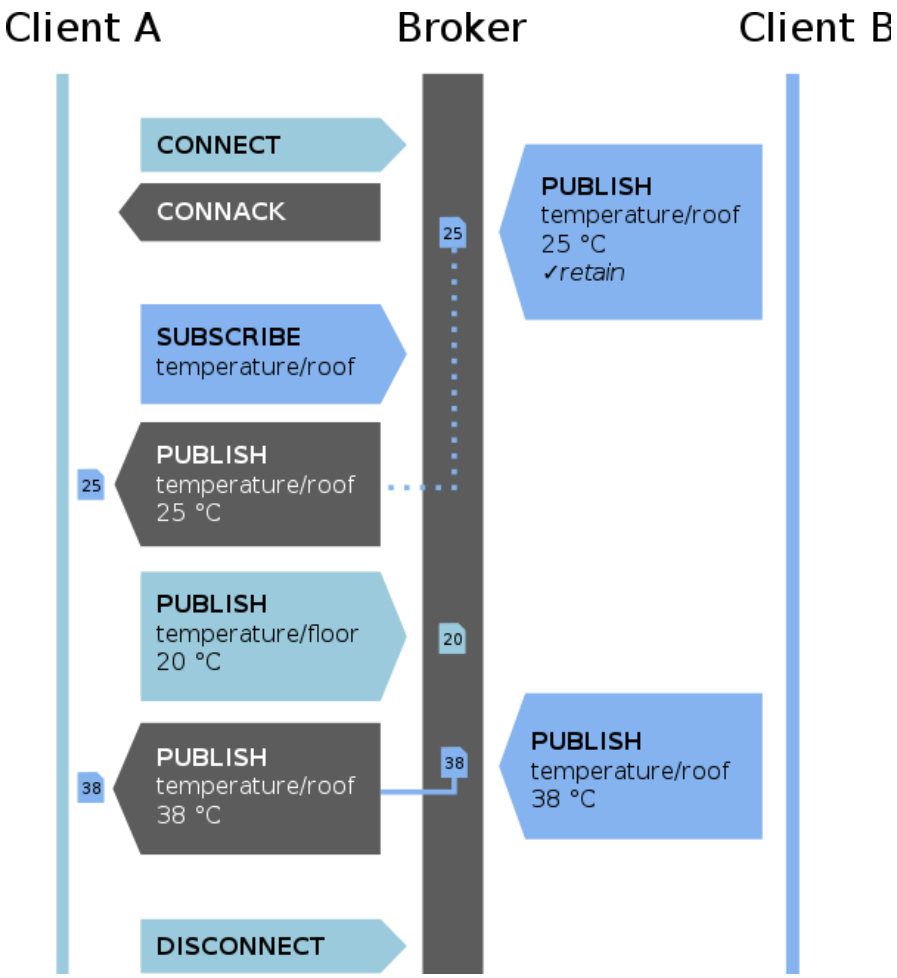
Es cualquier dispositivo que pueda comunicarse con el broker a través del protocolo MQTT. Cuando envía mensajes, se dice que actúa como editor, mientras que cuando los recibe actúa como receptor (AWS, s.f-a). Los clientes receptores deben indicar al broker el tópico al cual desean suscribirse, y esta información es lo que el broker utiliza para identificar a qué clientes debe ser reenviado un mensaje de ese tópico.

Conexión MQTT

Es el enlace entre un cliente y un agente MQTT. La conexión siempre es iniciada desde un cliente a un agente a través de un mensaje CONNECT. Si el agente acepta la conexión, responde con un mensaje CONNACK, tras el cual el cliente ya es capaz de interactuar con la red MQTT.

Una vez que un cliente establece correctamente una conexión con el agente, puede interactuar con él a partir de los siguientes mensajes:

- DISCONNECT: finaliza la conexión MQTT entre el cliente y el agente.
- SUBSCRIBE: establece un identificador para el dispositivo que lo identifica en la red, e indica la cola de mensajes de la cual solicita recibir mensajes. A partir de ese momento, cada vez que el broker reciba un mensaje en esa cola, lo reenviará al cliente que realizó la suscripción.
- PUBLISH: en el caso de un cliente, envía un mensaje a una cola para que el broker lo redistribuya a todos los clientes suscriptos a esa cola. En el caso de un broker, consiste en el reenvío de un mensaje de una cola a un cliente específico que esté suscripto.



QoS

En la comunicación entre sistemas distribuidos, siempre es importante definir claramente cómo debe proceder la comunicación en caso de fallos en el envío de mensajes o en el caso de errores durante su procesamiento por parte de los clientes. De esta forma, es posible lograr que sistemas que reaccionen de manera predecible a errores relacionados con los mensajes que envían o reciben.

Para ello, normalmente se establece para la conexión MQTT un nivel de calidad de servicio (abreviado QoS, por *Quality of* Service), que define qué medidas debe tomar cada extremo de la conexión para garantizar que el envío y procesamiento de los mensajes es exitoso. Siempre es aconsejable evaluar en cada sistema el grado de tolerancia a tener para mensajes no entregados correctamente, así como las posibles consecuencias de procesar dos veces un mismo mensaje. MQTT considera tres niveles de calidad de servicio (*Quality of Service*) (IBM, 2024), que se detallan a continuación:

QoS 0

El mensaje se entrega como máximo una vez, o no se entrega. A veces se denomina fire-and-forget. El receptor no responde al mensaje con un acuse de recibo, y dicho mensaje no es almacenado por el servidor. Si el cliente está desconectado o no disponible cuando se intenta retransmitir el mensaje, este es descartado. Se trata de la comunicación menos confiable para la recepción de mensajes, pero también es la más sencilla y rápida.

QoS 1

El mensaje se entrega al menos una vez al receptor, por lo que su entrega está eventualmente garantizada. El receptor debe responder al mensaje con un acuse de recibo para permitir al emisor borrarlo de su almacenamiento temporal. Si no recibe ningún acuse de recibo tras cierto tiempo, el emisor envía de nuevo el mensaje con un identificador de mensaje distinto, repitiendo esto hasta recibir algún acuse de recibo. Este nivel tiene la desventaja de que en teoría permite que un receptor procese varias veces el mismo mensaje, algo que debe ser evaluarse qué tan viable es de acuerdo con el significado del mensaje.

QoS 2

El mensaje PUBLISH siempre se entrega exactamente una vez. El receptor responde al mensaje con un acuse de recibo PUBREC, tras el cual puede descartar el mensaje ya enviado. En caso de no recibir el PUBREC, envía nuevamente el mismo paquete con el mismo identificador al receptor, repitiendo esto hasta recibir el PUBREC. Una vez que el emisor descarta el mensaje original para evitar su reenvío, se lo indica al receptor con un mensaje PUBREL. Al recibirlo y luego de terminar de procesar el mensaje PUBLISH original, el receptor responde al emisor nuevamente con un mensaje PUBCOMP, tras lo cual ambos extremos pueden descartar el mensaje PUBLISH con la certeza de que ya fue procesado exactamente una vez.

Seguridad

La seguridad del protocolo es una de las desventajas que presenta ya que no cuenta con mecanismos de seguridad muy avanzados, se puede mejorar esto incluyendo TLS para asegurar.

Brokers más usados

Mosquitto: Es el broker más conocido y utilizado en el ámbito doméstico es un broker Open Source desarrollado por la fundación Eclipse y distribuido bajo licencia

EPL/EDL. Está programado en C, y es multiplataforma. Es un broker liviano y adecuado para uso en servidores de baja potencia.

RabbitMQ. Es un popular broker de mensajería AMQP Open Source, que también permite emplear el protocolo MQTT a través de un Adaptador.

HiveMQ CE. La versión Community del popular HiveMQ es un broker Open Source basado en Java.

ActiveMQ.

MQTT vs AMQP

MQTT y AMQP son protocolos para el intercambio de mensajes entre servicios

AMQP. Son las siglas de Advanced Message Queuing Protocol y representan un protocolo de propósito múltiple para el intercambio de mensajes, el cual dispone de muchas opciones de configuración para que se pueda adaptar a los requerimientos de los distintos proyectos donde se desea aplicar.

Dentro del protocolo AMQP, el mensaje se divide en cabeceras y cuerpo del mensaje, lo que permite mayor versatilidad. Otra de las ventajas que tiene AMQP es que permite enrutar los mensajes de formas diferentes, según se necesite, hacia una cola o varias o haciendo el modelo publisher/subscriber de forma que un mensaje llegue a muchos suscriptores al mismo tiempo. Asimismo, dispone de colas persistentes, de manera que, si se reinicia el sistema, los mensajes no se borran.

Estas opciones hacen que AMQP sea más pesado y complejo que otras alternativas tales como MQTT.

MQTT es la abreviatura de Message Queuing Telemetry Transport, destaca por su ligereza, lo que lo convierte en la opción ideal para el Internet of Things o entornos en los que el acceso a Internet sea malo o complicado.

Cada paquete es pequeño y no dispone de todas las opciones que sí tiene AMQP, pero esto puede ser ideal para aprovechar los recursos al máximo o para conexiones efímeras o infrecuentes. La ventaja de esto es que un broker de MQTT puede conectar simultáneamente miles de clientes y es idóneo para sistemas con recursos muy limitados, como un Arduino.

En definitiva, MQTT es ideal para sistemas sencillos de mensajería, donde no se requieren escenarios complejos. Por ejemplo, MQTT se usa para enviar notificaciones a dispositivos móviles o en aplicaciones como WhatsApp.

Mientras que AMQP proporciona fiabilidad absoluta en la entrega de mensajes, en el sentido de que el broker siempre confirma que ha recibido el mensaje, MQTT tiene un funcionamiento diferente

Ejemplos de Tecnologías

Algunos ejemplos de tecnologías y aplicaciones que utilizan MQTT

- Plataformas IoT

AWS IoT Core: Proporciona una plataforma en la nube para conectar dispositivos IoT, donde MQTT facilita la comunicación entre dispositivos y la nube.

Microsoft Azure IoT Hub: Ofrece servicios de gestión y comunicación para dispositivos IoT, utilizando MQTT para la transmisión eficiente de datos.

IBM Watson IoT Platform: Utiliza MQTT para conectar dispositivos IoT y analizar datos en tiempo real.

- Hogar Inteligente

Home Assistant: Plataforma de automatización del hogar que soporta MQTT para la integración de dispositivos de diferentes fabricantes.

openHAB: Sistema de automatización del hogar que utiliza MQTT para intercomunicación entre dispositivos IoT y otros protocolos de automatización.

- Telemetría y Sensores

Node-RED: Herramienta de programación para conectar hardware y servicios online que permite la integración de datos usando MQTT.

Telegraf: Plugin para la recolección de métricas y eventos que utiliza MQTT para recibir datos de sensores y otros dispositivos IoT.

- Vehículos Conectados

Bosch IoT Suite: Plataforma para la conectividad de vehículos que utiliza MQTT para la comunicación entre vehículos y la infraestructura en la nube.

Tesla: Utiliza MQTT en ciertos sistemas para la telemetría y el monitoreo de vehículos.

- Sistemas de Gestión de Energía

OpenEMS: Sistema de gestión de energía que utiliza MQTT para la comunicación entre dispositivos de energía renovable y almacenamiento de energía.

Schneider Electric EcoStruxure: Plataforma para la gestión de energía y automatización de edificios que emplea MQTT para la comunicación entre dispositivos y sistemas.

- Agricultura Inteligente

Libelium: Utiliza MQTT en sus soluciones de agricultura para la transmisión de datos de sensores de suelo, clima, y otros parámetros.

FarmBot: Robot de agricultura de precisión que puede utilizar MQTT para la comunicación de datos y comandos.

- Automatización Industrial

Siemens MindSphere: Plataforma de IoT industrial que utiliza MQTT para la comunicación y análisis de datos en entornos de fabricación.

PTC ThingWorx: Plataforma para el desarrollo de aplicaciones industriales IoT que soporta MQTT para la interconexión de dispositivos industriales.

- Comunicaciones en Red

Mosquitto: Servidor MQTT (broker) de código abierto que facilita la comunicación en redes IoT.

HiveMQ: Plataforma MQTT para la mensajería IoT en tiempo real a gran escala.

Ejemplos de Caso de Uso

Algunos ejemplos de casos de uso más frecuentes de la tecnología MQTT

- Automatización del Hogar

Control de Iluminación Inteligente, en una casa inteligente, los dispositivos de iluminación pueden ser controlados mediante un sistema central usando MQTT.

Los interruptores envían mensajes MQTT al servidor central (broker), que luego comunica estos mensajes a las bombillas para encender, apagar o ajustar la iluminación.

Monitoreo de Sensores de Temperatura y Humedad, sensores distribuidos en diferentes habitaciones envían lecturas de temperatura y humedad a un servidor central.

Los sensores publican los datos a un tema MQTT, y la estación central se suscribe para recibir actualizaciones y ajustar sistemas de climatización.

- Agricultura Inteligente

Riego Automatizado, un sistema de riego que se activa automáticamente basado en la humedad del suelo.

Sensores de humedad en el suelo publican datos a un tema MQTT, y un sistema de riego se suscribe para activar o desactivar el riego basado en los niveles de humedad.

Monitoreo de Clima y Condiciones del Suelo, sensores que monitorean la temperatura, humedad, y otros parámetros ambientales en campos de cultivo.

Estos sensores envían datos a través de MQTT a una estación base que procesa y visualiza las condiciones en tiempo real, permitiendo ajustes en la gestión agrícola.

- Vehículos Conectados

Seguimiento de Flotas, gestión de una flota de vehículos para monitorear ubicaciones, velocidad y estado.

Cada vehículo publica datos sobre su ubicación y estado a un servidor MQTT, permitiendo a los gestores de flota seguir la posición de los vehículos y realizar ajustes logísticos.

Mantenimiento Predictivo, recolección de datos de sensores en vehículos para anticipar necesidades de mantenimiento.

Sensores en el vehículo envían datos sobre el estado del motor y otros componentes a través de MQTT, permitiendo análisis predictivos para el mantenimiento preventivo.

- Salud y Bienestar

Monitoreo de Pacientes en Casa, dispositivos médicos para monitorear signos vitales de pacientes en sus hogares.

Dispositivos como monitores de presión arterial o glucómetros envían datos a través de MQTT a servidores de salud para el monitoreo remoto por parte de profesionales médicos.

Rastreo de Actividad Física, dispositivos portátiles para el monitoreo de actividad física.

Los datos de actividad se envían a través de MQTT a una aplicación de salud que realiza seguimiento y análisis de la actividad diaria del usuario.

- Automatización Industrial

Control de Procesos Industriales, comunicación entre diferentes máquinas en una línea de producción.

Sensores y controladores en la línea de producción publican datos de rendimiento a un broker MQTT, permitiendo ajustes en tiempo real y monitoreo centralizado.

Monitoreo de Energía, seguimiento del consumo de energía en una planta industrial.

Medidores de energía envían datos a través de MQTT a un sistema central para el análisis del consumo y la optimización del uso de energía.

- Logística y Cadena de Suministro

Seguimiento de Inventario, monitoreo en tiempo real del inventario en almacenes.

Sensores de posición y estado en estanterías de almacén envían datos a través de MQTT a un sistema de gestión de inventario para la actualización continua del estado de productos.

Gestión de Cadenas de Frío, monitoreo de las condiciones de temperatura de productos sensibles en la cadena de suministro.

Sensores de temperatura en camiones y almacenes envían datos a través de MQTT, permitiendo el monitoreo continuo y la intervención inmediata si las condiciones no se mantienen dentro de los rangos requeridos.

- Gestión de Energía

Monitoreo de Paneles Solares, gestión y optimización del rendimiento de paneles solares.

Inversores y paneles solares envían datos de rendimiento a través de MQTT a una plataforma central para el monitoreo y ajuste del sistema.

Optimización del Consumo Eléctrico, control y ajuste del consumo eléctrico en edificios inteligentes.

Dispositivos de consumo eléctrico publican datos de uso a través de MQTT, y el sistema central ajusta el consumo basándose en la demanda y la disponibilidad de recursos.

Referencias

AWS. (s.f-a). *¿Qué es MQTT?*. Recuperado el 27 de junio de 2024 de https://aws.amazon.com/es/what-is/mqtt/

AWS. (s.f-b). *¿Qué es una cola de mensajes?* Recuperado el 27 de junio de 2024 de https://aws.amazon.com/es/message-queue/

Cleo. (s.f). *Securely connect to your trading partners using top-rated integration software*. Recuperado el 27 de junio de 2024 de https://www.cleo.com/solutions/integration-connectors/message-queue-protocols

IBM. (31 de enero de 2024). *Calidades de servicio proporcionadas por un cliente de MQTT*. Recuperado el 27 de junio de 2024 de https://www.ibm.com/docs/es/ibm-mq/9.1?topic=concepts-qualities-service-provided-by-mqtt-client