DOCUMENTO TÉCNICO

Integración de dispositivos con MQTT

Marzo 2022

AXIS COMMUNICATIONS

Resumen

MQTT es un protocolo de mensajería estándar que permite un intercambio de datos eficaz y seguro entre dispositivos IoT y aplicaciones en la nube. Gracias a este protocolo, los dispositivos (a través de sus clientes MQTT) pueden publicar mensajes en un broker (servidor) MQTT común, que hace de intermediario en la comunicación con otros dispositivos. El broker lleva un registro de quién publica qué y quién quiere ver los datos, y reenvía los mensajes únicamente a los clientes suscritos al tema correcto.

En un ecosistema VMS típico, las notificaciones de eventos de Axis procedentes de dispositivos se transmiten tradicionalmente a un único destino a través de una interfaz API VAPIX/ONVIF utilizando el protocolo de transmisión RTSP. Sin embargo, las mismas notificaciones pueden distribuirse usando el protocolo MQTT a través del cliente MQTT integrado del dispositivo (disponible para dispositivos con la versión 9.80 o posterior del SO AXIS). Este sistema puede utilizarse tanto dentro de ecosistemas VMS como fuera y resulta especialmente práctico en la comunicación a través de internet. Varios clientes MQTT suscritos de la red pueden utilizar y procesar las notificaciones de eventos publicadas por el dispositivo Axis. También existen aplicaciones de analítica ACAP de Axis y de terceros que tienen sus propios clientes MQTT para sistemas, casos de uso y suscriptores específicos.

Un ejemplo de uso vinculado a productos Axis lo encontramos en los dispositivos de recuento de personas, que pueden enviar a través de MQTT estadísticas a software de visualización de datos en la nube. En otro ejemplo, un sensor de puerta de un tercero se comunica mediante MQTT con un dispositivo de señalización y una cámara, que emiten una alarma e inician una grabación cada vez que se abre la puerta.

Índice

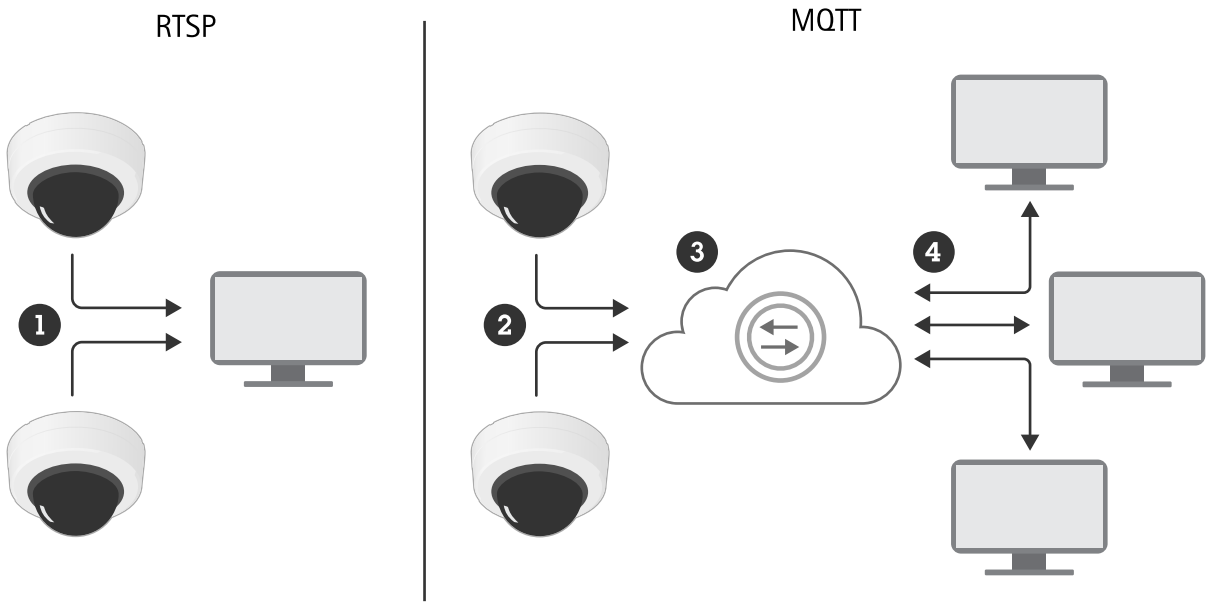
1 Introducción

MQTT (Message Queuing Telemetry Transport) es un protocolo de mensajería estándar para internet of things (IoT). Se diseñó para simplificar la integración del IoT y se utiliza en una amplia variedad de sectores para conectar dispositivos remotos con una huella de código pequeña y un ancho de banda de red mínimo. El cliente MQTT del SO AXIS puede simplificar la integración de los datos y eventos producidos en el dispositivo con sistemas que no sean sistemas de gestión de vídeo (VMS).

Este documento técnico presenta información técnica sobre MQTT, además de ejemplos de uso habituales, ventajas y limitaciones. También proporciona detalles sobre los clientes MQTT de dispositivos Axis y apps de analítica de la ACAP.

2 Protocolo MQTT

MQTT es un protocolo de publicación/suscripción. Y eso significa que usa un patrón de transmisión de mensajes diferente de RTSP o HTTP, que son protocolos de solicitud/respuesta. Con RTSP, un extremo envía una solicitud y el otro responde. En cambio, muchas apps de mensajería para dispositivos móviles utilizan MQTT o sistemas parecidos de publicación/suscripción. Existen también protocolos de publicación/suscripción optimizados para sistemas cerrados o específicos.



Y mientras que RTSP permite solo la comunicación uno a uno, MQTT abre la puerta también a la comunicación uno a muchos y muchos a muchos a través del broker.

1 *Flujo de eventos*
2 *Publicación*
3 *Broker MQTT*
4 *Suscripción*

La idea detrás de MQTT es que todos los clientes se conectan a un broker (servidor) MQTT común, que controla quién publica cada contenido y quién quiere ver los datos. La conexión suele ser una sesión TCP en el puerto 1883. Sin embargo, un cliente puede conectarse también a través de TLS (normalmente desde el puerto 8883) o usando WebSocket (generalmente el puerto 1884/8884).

Los clientes publican mensajes con un tema. Otro cliente se suscribe a ese tema concreto o usa comodines para acceder a todos los temas. Un mensaje puede incluir también una carga útil, que suele ser una estructura de datos JSON, una cadena o incluso datos binarios cortos. El publicador no sabe si otros clientes están suscritos. El broker solo enviará mensajes a los clientes con suscripciones a temas.

Con MQTT, el funcionamiento es parecido a enviar un artículo a una revista. Las personas que se suscriban a la revista podrán leer el artículo y la comunicación puede ser de tipo uno a uno o uno a muchos (y con MQTT incluso es posible la comunicación de muchos a muchos). Además, el artículo también puede leerse mucho después de su fecha de publicación inicial.

En cambio, RTSP tiene un funcionamiento más similar al de un teléfono. Los comandos tienen un origen y un destino, y la comunicación es siempre de uno a uno. Si su destino no responde al teléfono, el mensaje se pierde.

Cuando se utiliza MQTT para distribuir notificaciones de eventos Axis desde dispositivos, varios clientes MQTT suscritos de la red pueden usar y procesar las notificaciones. Y esto ofrece una gran ventaja en comparación con el procedimiento tradicional (el uso de la API o interfaz de programación de aplicaciones VAPIX®/ONVIF® y RTSP), en el que las notificaciones de eventos se transmitían únicamente a un solo destino.

3 Ventajas

Son muchas las ventajas de usar MQTT. En comparación con un protocolo de solicitud/respuesta como RTSP, las principales ventajas de MQTT:

- **Reducción del riesgo de exposición de las contraseñas de los dispositivos.** No es necesario que el cliente acceda a un dispositivo o servidor para obtener los datos. Por lo tanto, el cliente no tiene que conocer la contraseña ni saber cómo funciona la API. Esto reduce el riesgo de exposición de las contraseñas de los dispositivos a clientes y usuarios, por lo que disminuye el riesgo de uso inapropiado de forma accidental o deliberada.
- **Único punto de integración.** Si tienen la autorización pertinente, todos los clientes pueden acceder a los mensajes publicados por todos los demás clientes con una única conexión a un broker. En RTSP, un cliente tiene que conectarse a cada cliente que tiene los datos que necesita. Por lo tanto, el flujo de mensajes de MQTT puede ser de uno a uno, de uno a muchos o de muchos a uno sin ninguna carga adicional para cada cliente.
- **Publicación y suscripción con un cortafuegos intacto**. En RTSP, el cliente tiene que poder acceder a la API del dispositivo/servidor. Si el dispositivo está detrás de un cortafuegos y el cliente es remoto, es necesario configurar el cortafuegos para que permita las solicitudes entrantes, lo que implica exponer la API del dispositivo. Con un broker MQTT público de por medio, los clientes que están detrás de un cortafuegos pueden publicar/suscribirse a datos concretos sin reducir la seguridad del cortafuegos (siempre que el cortafuegos permita las conexiones salientes).
- **Calidad de servicio.** Al publicar un mensaje crítico, un publicador puede supervisar si ese mensaje ha llegado a otro cliente y tomar medidas si no lo ha hecho.
- **Mensajes retenidos.** Los publicadores pueden marcar un mensaje como retenido, lo que significa que el broker guardará una copia del mensaje y lo enviará a nuevos clientes conectados que estén suscritos a ese tema.
- **Disponibilidad para clientes IoT.** Existen paquetes para clientes MQTT para todos los principales entornos de desarrollo de software, como Windows®, Linux®, Android™, iOS, Node.js®, PHP y Python®. Hay muchos más clientes que pueden conectarse a un broker, en comparación con el procedimiento de configuración de un flujo de datos RTSP en un dispositivo.

- **Proceso simplificado de supervisión de mensajes y depuración.** Existen varias herramientas MQTT que permiten supervisar todos los mensajes publicados y también publicar mensajes para solucionar problemas a raíz de la reacción de los suscriptores.
- **Estructura simplificada de los datos.** Como MQTT a menudo se dirige a clientes desconocidos, la carga útil del mensaje lo tiene en cuenta y simplifica los datos que envía al suscriptor.

4 Limitaciones

En comparación con protocolos alternativos, MQTT presenta varios inconvenientes:

- **Punto único de fallo** Si el broker no está disponible, no se envía ningún mensaje. Sin embargo, la infraestructura puede incorporar brokers redundantes.
- **¿Quién ha publicado el mensaje?** Por diseño, MQTT se centra en el tema, no en quién ha publicado el mensaje. A menos que el publicador incluya algún tipo de identificador en el tema o la carga útil, será necesario acceder al registro del broker para saber quién ha publicado el mensaje. Es habitual que el publicador incluya algún tipo de identificación del cliente en el tema o la carga útil en función del caso de uso.
- **Un cliente malicioso** conectado al broker puede publicar/suscribirse a cualquier tema para el que tenga autorización. Es importante proteger el broker (véase el apartado sobre la seguridad de MQTT).
- **No está diseñado para la transmisión continua de vídeo/audio**.

Como ocurre con cualquier servidor, hay que valorar el uso total de ancho de banda. En sistemas muy grandes con muchos clientes, puede ser necesario un escalado dinámico.

5 Infraestructura

Es relativamente sencillo configurar un broker Mosquitto™ local o activar Node-RED® para que actúe como un broker local, como Aedes. Existen también varios proveedores de servicios de internet y otros proveedores que ofrecen MQTT gestionados, como Microsoft® Azure® IoT, HiveMQ™, CloudMQTT e IBM® Cloud®.

Si un cliente no tiene clientes remotos, se recomienda usar un broker local. Un broker local puede actuar también como proxy de un broker público o puede configurarse para que actúe como un proxy de determinados mensajes del broker local y los mensajes del broker público.

6 Seguridad

El broker necesita la protección adecuada en función del nivel de confidencialidad de los mensajes y de las amenazas a las que está expuesto un sistema concreto. MQTT ofrece diferentes opciones de autenticación, como sin autenticación, usuario/contraseña y autenticación con el certificado del cliente TLS. Diferentes usuarios pueden tener distintas autorizaciones en relación con los temas de publicación o suscripción. El broker puede permitir a los clientes conectarse a través de un TCP sin cifrar o través de un TLS cifrado (como HTTPS).

- **Sin autenticación.** Un broker local puede desactivar la autenticación si los mensajes no son críticos y el broker no está expuesto a clientes de internet. Se recomienda utilizar esta opción únicamente para pruebas, desarrollo en sandbox y demostraciones.

- **Usuario/contraseña.** Este es el ajuste más habitual. En función de los riesgos del sistema, el administrador del sistema puede usar un mismo usuario/contraseña para todos los clientes MQTT o crear usuarios con acceso restringido solo a determinados temas.
- **Certificados de clientes TLS.** En el caso de los brokers expuestos a internet que transmiten mensajes considerados delicados, el broker debe configurarse de modo que el acceso se restrinja a los clientes con un certificado TLS válido. Para usar este sistema, es necesaria una PKI (infraestructura de clave pública) y una autoridad de certificación que pueda emitir certificados para clientes aceptada por el broker. Los proveedores de servicios de internet MQTT públicos suelen ofrecer esta opción.

En algunas situaciones, es preferible segmentar diferentes casos de uso entre varios brokers, ya sean brokers locales y/o públicos. La segmentación de mensajes críticos y no críticos es un control de seguridad. Además, al disponer de varios brokers se reduce el riesgo asociado a un punto único de fallo y se optimiza la supervisión y la resolución de problemas. El coste es más elevado a causa del trabajo de implantación y mantenimiento de más brokers.

7 Cliente MQTT en dispositivos Axis

En un ecosistema VMS estándar, las notificaciones de eventos de Axis procedentes de dispositivos se transmiten tradicionalmente a un único destino a través de una interfaz API VAPIX/ONVIF utilizando un protocolo de transmisión RTSP.

Las mismas notificaciones de eventos pueden distribuirse usando el protocolo MQTT a través del cliente MQTT integrado de un dispositivo Axis (con SO AXIS 9.80 o una versión posterior). Este sistema puede utilizarse tanto dentro de ecosistemas VMS como fuera y resulta especialmente práctico en la comunicación a través de internet. Con MQTT, varios clientes MQTT suscritos de la red puede utilizar y procesar las notificaciones de eventos publicadas por el dispositivo Axis.

8 Clientes MQTT en aplicaciones de analítica ACAP

Existen aplicaciones ACAP de Axis y de terceros que tienen sus propios clientes MQTT para sistemas, casos de uso y suscriptores específicos. Axis Publisher es un buen ejemplo de cliente que añade las funciones, estructuras y comportamientos que necesitan algunos sistemas.

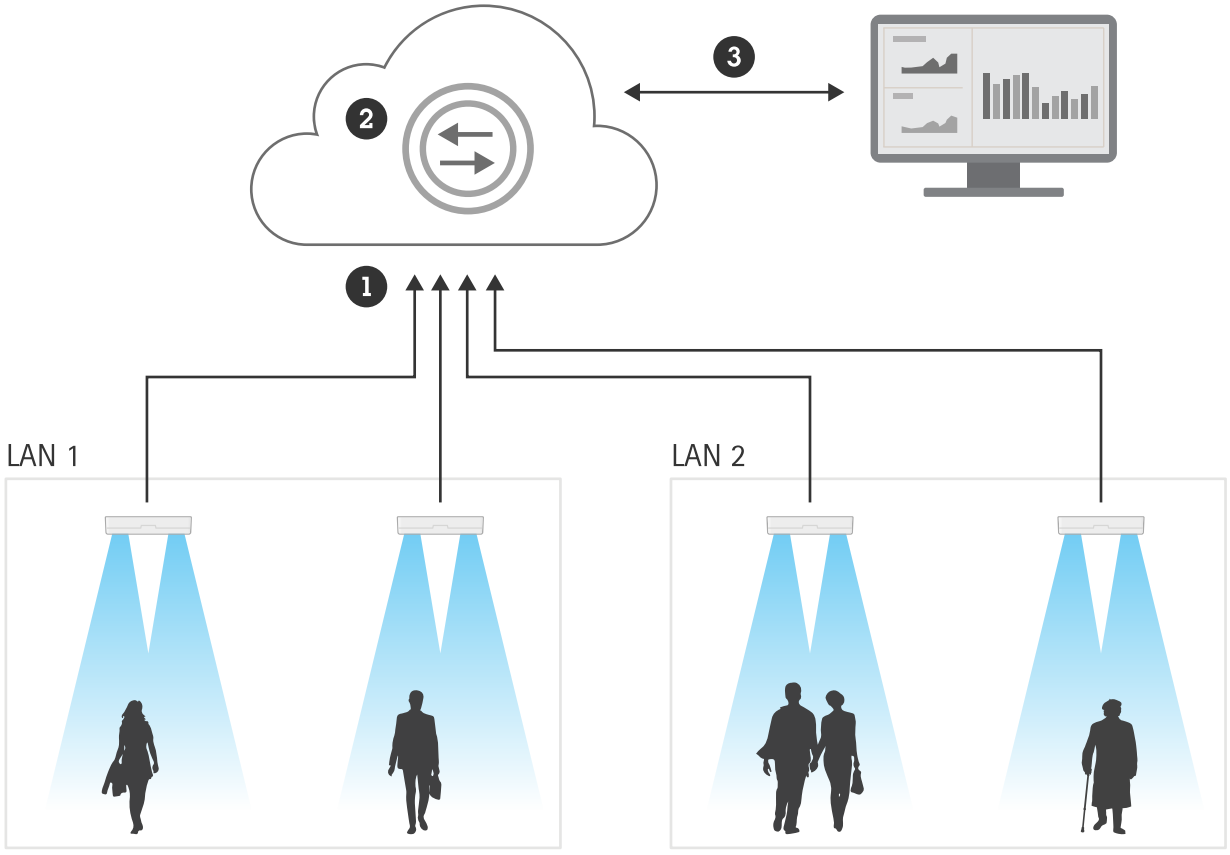
9 Otros clientes MQTT

Existe un amplio abanico de clientes MQTT que pueden instalarse en Linux, Windows, Android o iOS y que están diseñados como herramientas o servicios para casos de uso específicos. MQTT funciona especialmente bien para scripting y middleware, como Node-RED/Node.js, Python y PHP. La mayoría de plataformas de servicios de internet, como Microsoft Azure IoT, AWS™ y Google Cloud Platform™, ofrecen brokers de MQTT para su integración en los servicios disponibles a través de la plataforma. Existen sensores, aplicaciones móviles y sistemas de automatización (domótica) con un cliente MQTT.

10 Ejemplos prácticos de integración de dispositivos con MQTT

10.1 Datos de analítica de recuento de personas en un panel de una plataforma en la nube

Un dispositivo con analítica de recuento de personas genera una notificación de evento cada vez que detecta a una persona entrando o saliendo en una zona definida. La notificación se traslada al cliente MQTT, que la publica en tiempo real en la plataforma en la nube. En la plataforma en la nube, se establece una conexión con un software de visualización de datos (como un panel Microsoft® Power BI®) para mostrar las estadísticas en tiempo real de los contadores de personas.

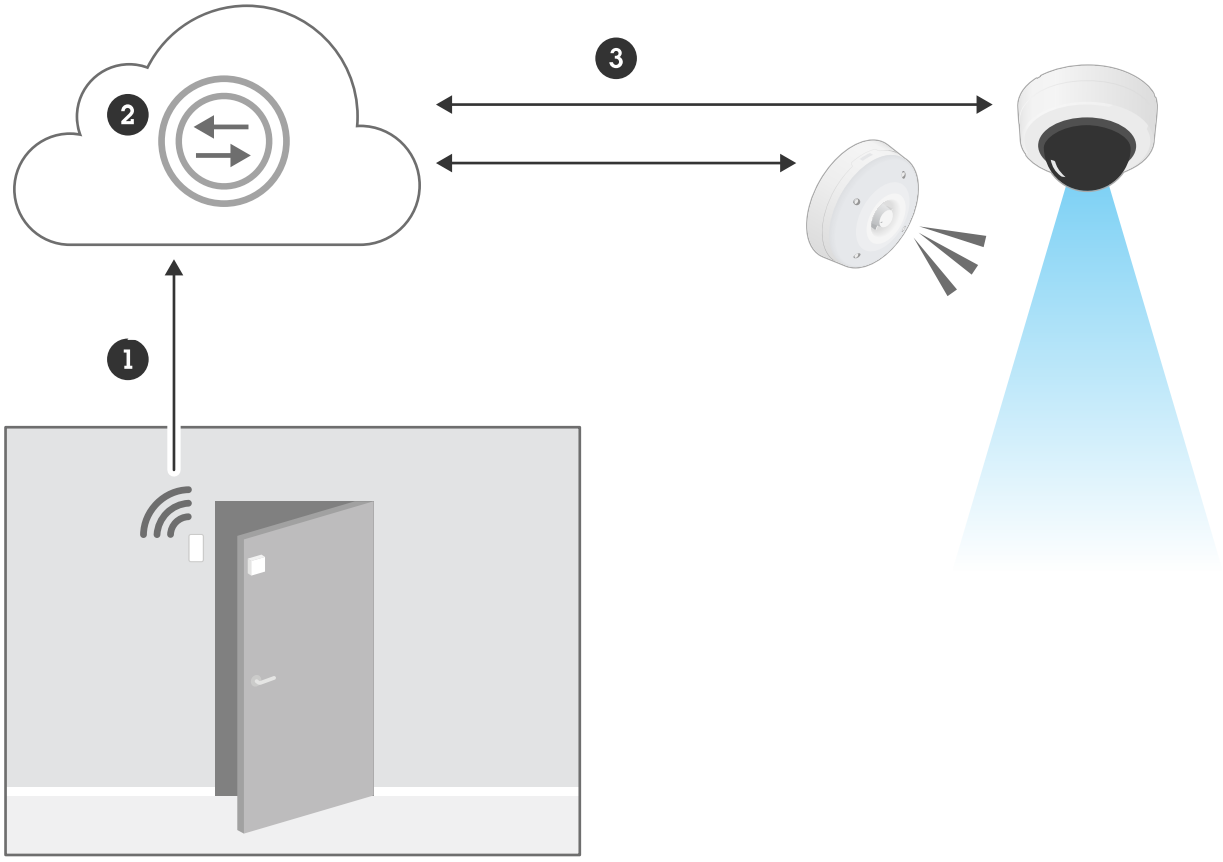


1 *Publicación*
2 *Broker MQTT*
3 *Suscripción*

10.2 Los datos de sensores de puerta por MQTT activan alarmas de dispositivos de señalización y grabaciones de cámaras

Un sensor de puerta MQTT de un tercero se utiliza para activar una notificación de evento si se abre la puerta. El sensor de la puerta publica un mensaje MQTT en el broker MQTT en la nube. El dispositivo de

señalización y la cámara se suscriben al tema del sensor de puerta y emiten una alarma e inician una grabación si se abre la puerta.



1 *Publicación*
2 *Broker MQTT*
3 *Suscripción*

11 Glosario

ACAP	*Plataforma de aplicaciones de cámaras AXIS*, una arquitectura de aplicaciones que aportan más funcionalidades e inteligencia en el extremo
Aedes	Un broker MQTT
API	*Interfaz de programación de aplicaciones*, código que permite la intercomunicación de los programas de software
AWS™	Una plataforma de servicios en la nube
SO AXIS	El sistema operativo de los dispositivos en el extremo de Axis
CloudMQTT	Un broker MQTT
Eclipse Mosquitto™	Un broker de mensajes de código abierto que aplica protocolos MQTT
Google Cloud Platform™	Una plataforma de servicios en la nube
HiveMQ™	Un broker MQTT
HTTP	*Protocolo de transferencia de hipertexto*, un protocolo de transferencia de datos utilizado en internet
IBM Cloud®	Una plataforma de servicios en la nube
IoT	*Internet of things*, la interconexión a través de internet de dispositivos informáticos integrados en electrodomésticos y equipos y dispositivos de uso cotidiano
JSON	*JavaScript Object Notation*, un formato compacto de archivos e intercambio de datos
Microsoft® Azure® IoT	Una plataforma de servicios en la nube
Microsoft® Power BI®	Un software de visualización de datos interactivo especializado en inteligencia empresarial
MQTT	*Message Queuing Telemetry Transport*, un protocolo de mensajería estándar para internet of things
Node.js®	Una plataforma de desarrollo de código abierto para ejecutar código JavaScript en servidores
Node-RED®	Una herramienta de programación para la conexión del internet of things
ONVIF®	Un foro abierto del sector que proporciona y promueve interfaces estandarizadas para la interoperabilidad efectiva de productos de seguridad física basados en IP
PHP	Un lenguaje de scripts de tipo general pensado para el desarrollo web
Python®	Un lenguaje de programación de tipo general
RTSP	*Protocolo de transmisión en tiempo real*, un protocolo de red para establecer y controlar sesiones de comunicación entre terminales
TCP	*Protocolo de control de transmisiones*, un protocolo de transporte de datos que es uno de los más utilizados en internet
TLS	*Capa de transporte seguro*, un protocolo que garantiza la confidencialidad y la integridad de las comunicaciones por redes de ordenadores
VAPIX®	La interfaz de programación de aplicaciones (API) abierta de los productos Axis

WebSocket	Un protocolo de comunicaciones que proporciona canales de comunicación bidireccionales a través de una única conexión TCP
VMS	*Software de gestión de vídeo* o *sistema de gestión de vídeo*

12 Marcas comerciales

Android y Google Cloud Platform son marcas comerciales de Google LLC.

AWS es una marca comercial de Amazon.com, Inc. o sus empresas afiliadas en Estados Unidos y/u otros países.

Eclipse Mosquitto es una marca comercial de Eclipse Foundation, Inc.

HiveMQ es una marca comercial de HiveMQ GmbH.

IBM e IBM Cloud son marcas comerciales de International Business Machines Corp, registrada en muchas jurisdicciones en todo el mundo.

IOS es una marca comercial o una marca comercial registrada de Cisco Systems, Inc y/o sus empresas afiliadas en Estados Unidos y otros países y Apple, Inc. la utiliza con licencia.

JavaScript es una marca comercial registrada de Oracle Corporation en Estados Unidos.

Linux es una marca comercial registrada de Linus Torvalds en Estados Unidos y otros países.

Microsoft, Windows, Microsoft Azure IoT y Microsoft Power BI son marcas comerciales registradas de Microsoft Corporation.

Node.js y Node-RED son marcas comerciales registradas de OpenJS Foundation en Estados Unidos y/u otros países.

ONVIF es una marca comercial de Onvif, Inc.

Python es una marca comercial registrada de Python Software Foundation.

T10162730/ES/M1.17/2203

Acerca de Axis Communications

Axis contribuye a crear un mundo más inteligente y seguro a través de soluciones para mejorar la seguridad y el rendimiento empresarial. Como empresa de tecnología de red y líder del sector, Axis ofrece soluciones de videovigilancia, control de acceso y sistemas de audio e intercomunicación. Se ven reforzadas por aplicaciones de análisis inteligentes y respaldadas por formación de alta calidad.

Axis tiene alrededor de 4000 empleados dedicados en más de 50 países y colabora con socios de integración de sistemas y tecnología en todo el mundo para ofrece soluciones personalizadas. Axis se fundó en 1984 y la sede está en Lund, Suecia