백서

MQTT와의 장치 통합

3월 2022

요약

MQTT는 IoT 장치와 클라우드 애플리케이션 간의 효율적이고 신뢰할 수 있는 데이터 교환을 용이하게 하는 표준 메시징 프로토콜입니다. 이를 통해 장치(MQTT 클라이언트를 통해)가 다른 장치와의 통신을 중재하는 공통 MQTT 브로커(서버)에 메시지를 게시할 수 있습니다. 브로커는 누가 무엇을 게시하고 누가 데이터를 보고 싶어하는지 추적하여 해당 주제를 구독하는 클라이언트에게만 메시지를 전달합니다.

일반적 VMS 생태계에서 장치의 Axis 이벤트 알림은 전통적으로 RTSP 스트리밍 프로토콜을 사용하는 VAPIX/ONVIF API 인터페이스를 통해 단일 대상으로 스트리밍됩니다. 그러나 동일한 알림을 장치의 내장 MQTT 클라이언트를 통해 MQTT 프로토콜을 사용하여 배포할 수 있습니다(AXIS OS 9.80 이상 버전을 실행하는 장치에 적용 가능). 이는 VMS 생태계 내부와 외부 모두에서 가능하며 특히 인터넷을 통해 배포하는 데 유용합니다. 따라서 네트워크의 다중 구독 MQTT 클라이언트는 Axis 장치에서 게시한 이벤트 알림을 사용하고 처리할 수 있습니다. 특정 시스템, 사용 사례 및 구독자를 위해 고안된 자체 MQTT 클라이언트가 있는 Axis 및 타사 ACAP 분석 애플리케이션도 있습니다.

Axis 제품과 관련된 사용 사례의 예로 인원 계수 장치는 MQTT를 통해 클라우드의 데이터 시각화 소프트웨어에 통계를 보낼 수 있습니다. 다른 예에서는, 타사 도어 센서가 MQTT를 통해 도어가 열릴 때마다 경보를 울리고 녹화를 시작하는 신호 장치 및 카메라와 통신합니다.

목차

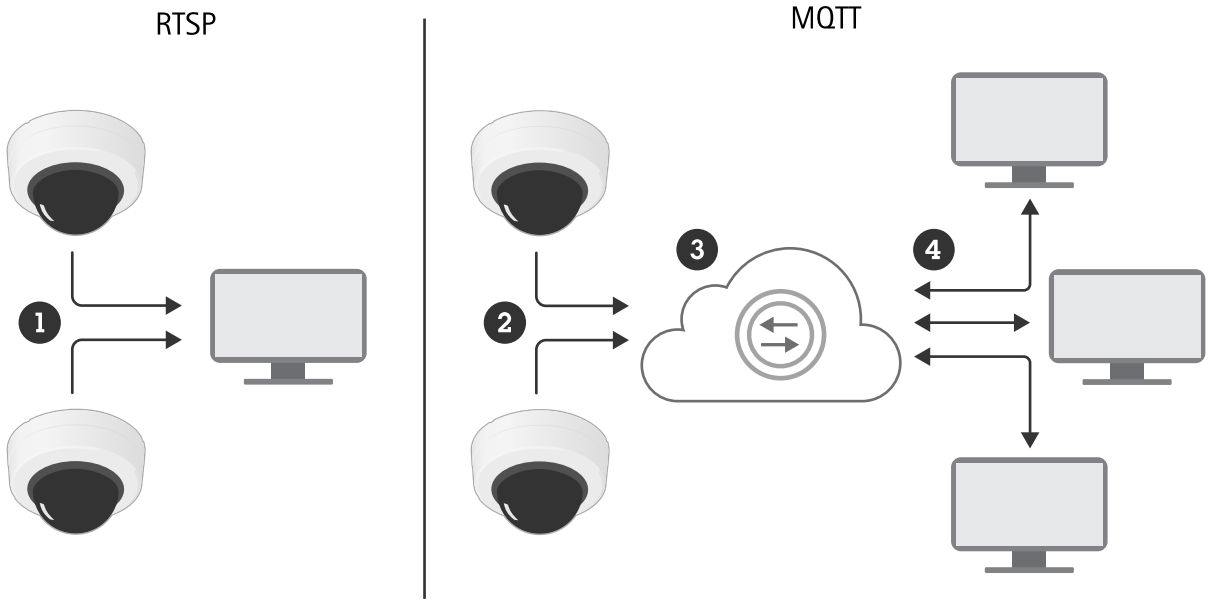
1 서론

MQTT(Message Queuing Telemetry Transport)는 사물 인터넷(IoT)을 위한 표준 메시징 프로토콜입니다. 단순화된 IoT 통합을 위해 설계되었으며 작은 코드 공간(small code footprint)과 최소 네트워크 대역폭으로 원격 장치를 연결하기 위해 다양한 산업에서 사용됩니다. AXIS OS에 내장된 MQTT 클라이언트를 통해 장치에서 생성된 데이터 및 이벤트를 영상 관리 시스템(VMS)이 아닌 시스템에 간편하게 통합할 수 있습니다.

이 백서에서는 일반적인 사용 사례, 이점 및 제한 사항을 포함하여 MQTT에 대한 기술적 배경을 제공합니다. Axis 장치에 내장된 MQTT 클라이언트 및 ACAP 분석 앱에 대한 세부 정보도 제공합니다.

2 MQTT 프로토콜

MQTT는 게시/구독 프로토콜입니다. 이는 요청/응답 프로토콜인 RTSP 또는 HTTP와 다른 메시징 패턴을 갖고 있음을 의미합니다. RTSP를 사용하면, 한쪽에서 요청을 보내고 다른 쪽에서 응답합니다. 대신에, 많은 모바일 메시징 앱은 MQTT 또는 유사한 게시/구독 개념을 기반으로 합니다. 폐쇄형 또는 특정 시스템에 최적화된 게시/구독 프로토콜도 있습니다.



RTSP는 일대일 통신만 허용하지만, MQTT는 브로커를 통한 일대다 통신 또는 다대다 통신도 가능하게 합니다.

1 *이벤트 스트림*
2 *게시*
3 *MQTT 브로커*
4 *구독*

MQTT 개념은 모든 클라이언트가 공통 MQTT 브로커(서버)에 연결하는 것이며, 브로커는 누가 무엇을 게시하고 누가 데이터를 보고 싶어하는지 추적합니다. 연결은 일반적으로 포트 1883의 TCP 세션

입니다. 클라이언트는 TLS(일반적으로 포트 8883) 또는 WebSocket(일반적으로 포트 1884/8884)을 통해 연결할 수도 있습니다.

클라이언트는 주제가 있는 메시지를 게시합니다. 다른 클라이언트는 특정 주제를 구독하거나 와일드카드를 사용하여 모든 하위 주제에 접근할 수 있습니다. 메시지에는 일반적으로 JSON 데이터 구조, 문자열 또는 심지어 짧은 이진 데이터인 페이로드도 포함됩니다. 게시자는 다른 클라이언트가 구독 중인지 알 수 없습니다. 브로커는 주제를 구독하는 클라이언트에만 메시지를 전달합니다.

MQTT의 경우 잡지에 기사를 보내는 것과 같은 역할을 합니다. 잡지 구독자는 기사를 읽을 수 있으며 이는 일대일 또는 일대다 통신이 될 수 있습니다(MQTT는 다대다 통신도 가능하게 합니다). 기사는 게재된 후에도 오랫동안 읽을 수 있습니다.

이에 비해 RTSP를 사용하는 것은 전화를 거는 것과 더 비슷할 것입니다. 명령을 위한 하나의 소스와 하나의 대상이 있으며, 이는 항상 일대일 통신입니다. 대상이 전화를 받지 않았다면, 대상은 메시지를 놓친 것입니다.

MQTT가 장치의 Axis 이벤트 알림을 배포하는 데 사용되는 경우, 네트워크의 다중 구독 MQTT 클라이언트가 알림을 사용하고 처리할 수 있습니다. 이는 이벤트 알림이 하나의 단일 대상으로 스트리밍되는 기존 방식(VAPIX®/ONVIF® 애플리케이션 프로그래밍 인터페이스(API) 및 RTSP를 사용하는)과 비교하여 커다란 이점입니다.

3 이점

MQTT를 사용하면 몇 가지 이점이 있습니다. RTSP와 같은 요청/응답 프로토콜을 사용하는 것과 비교할 때 MQTT의 주요 이점은 다음과 같습니다.

- **장치 암호가 노출될 위험이 줄어듭니다.** 클라이언트가 데이터를 가져오기 위해 장치나 서버에 액세스할 필요가 없습니다. 즉, 클라이언트는 암호를 알 필요도 없고 API 작동 방식도 알 필요가 없습니다. 이에 따라 장치 암호가 클라이언트와 사용자에게 노출될 위험이 줄어들어 고의적이거나 우발적인 오용 위험이 줄어듭니다.
- **단일 통합 지점.** 승인된 경우, 모든 클라이언트는 브로커에 대한 단일 연결을 통해 다른 모든 클라이언트의 게시된 메시지를 가져올 수 있습니다. RTSP에서 클라이언트는 데이터를 가져올 각 클라이언트에 연결해야 합니다. 이는 MQTT 메시지 흐름이 각 클라이언트에 대한 추가 부담 없이 일대일, 일대다 또는 다대일일 수 있음을 의미합니다.
- **손상되지 않은 방화벽을 통해 게시 및 구독** RTSP에서는 클라이언트가 장치/서버 API에 액세스할 수 있어야 합니다. 장치가 방화벽 뒤에 있고 클라이언트가 원격인 경우, 들어오는 요청을 허용하여 장치 API를 노출하도록 방화벽을 구성해야 합니다. 중간에 공용 MQTT 브로커가 있으면 방화벽 뒤에 있는 클라이언트가 방화벽을 손상시키지 않고 특정 데이터를 게시/구독할 수 있습니다(방화벽이 아웃바운드 연결을 허용하는 경우).
- **서비스 품질(QoS).** 중요한 메시지를 게시할 때, 게시자는 해당 메시지가 다른 클라이언트에 의해 수신되었는지 모니터링하고, 수신되지 않은 경우 대체 조치를 취할 수 있습니다.
- **보관된 메시지.** 게시자는 메시지를 보관 상태로 표시할 수 있습니다. 즉, 브로커가 메시지 사본을 보관하고 해당 주제를 구독하는 새로 연결된 클라이언트에 해당 메시지를 보냅니다.

- **IoT 클라이언트 가용성.** MQTT 클라이언트 패키지는 Windows®, Linux®, Android™, iOS, Node.js®, PHP 및 Python®을 포함한 모든 일반 소프트웨어 개발 환경에 사용할 수 있습니다. 장치에 RTSP 데이터 스트림을 설정하는 것과 비교하여 브로커에 연결할 수 있는 클라이언트가 더 많습니다.
- **단순화된 메시지 모니터링 및 디버깅.** 게시된 모든 메시지를 모니터링하고, 구독자가 반응할 경우 문제 해결을 위해 메시지를 게시하고 구독자가 반응하는 방식을 모니터링하는 데 사용할 수 있는 여러 MQTT 도구가 있습니다.
- **단순화된 데이터 구조.** MQTT는 종종 알 수 없는 클라이언트를 대상으로 하기 때문에 메시지 페이로드는 일반적으로 구독자를 위해 단순화하기 위해 이를 고려합니다.

4 제한

대체 프로토콜과 비교할 때 MQTT에는 몇 가지 단점이 있습니다.

- **단일 실패 지점.** 브로커를 사용할 수 없으면 모든 메시징이 작동을 멈춥니다. 그러나 인프라는 이중화 브로커를 이용해 설계할 수 있습니다.
- **누가 메시지를 게시했습니까?** 기본적으로 MQTT는 메시지를 게시한 사람이 아니라 주제에 중점을 둡니다. 게시자가 주제 또는 페이로드의 일부로 일부 ID를 포함하지 않을 경우, 메시지를 게시한 사람을 알려면 브로커의 로그에 액세스해야 합니다. 게시자는 사용 사례에 따라 주제 또는 페이로드에 일부 클라이언트 ID를 포함하는 것이 일반적입니다.
- 브로커에 연결된 **악성 클라이언트**는 접근 권한이 부여된 모든 주제를 게시/구독할 수 있습니다. 브로커를 보호해야 합니다(MQTT 보안 섹션 참조).
- 브로커는 **연속적인 비디오/오디오 스트리밍**용으로 고안되지 않았습니다.

어느 서버의 경우와도 마찬가지로, 총 대역폭 처리량을 고려해야 합니다. 클라이언트가 많은 초대형 시스템에는 동적 확장이 필요할 수 있습니다.

5 인프라

로컬 Eclipse Mosquitto™ 브로커를 설정하거나 Node-RED®가 Aedes와 같은 로컬 브로커로 작동하도록 하는 것은 상당히 쉽습니다. Microsoft® Azure® IoT, HiveMQ™, CloudMQTT, IBM® Cloud®와 같은 관리형 MQTT 브로커를 제공하는 인터넷 서비스 공급자 및 기타 업체도 많이 있습니다.

시스템에 원격 클라이언트가 없는 경우 로컬 브로커를 사용하는 것이 좋습니다. 로컬 브로커는 공용 브로커에 대한 프록시 역할을 할 수 있거나, 선택한 로컬 브로커 메시지 및 공용 브로커 메시지에 대한 프록시 역할을 하도록 구성할 수도 있습니다.

6 보안

브로커는 메시지의 중요도와 특정 시스템이 경험할 수 있는 위협에 따라 적절한 보호가 필요합니다. MQTT는 무인증, 사용자/암호 및 TLS 클라이언트 인증서 인증을 비롯한 여러 가지 인증 방식을 제공합니다. 사용자마다 게시하거나 구독할 수 있는 주제에 대해 각각 다른 권한을 가질 수 있습

니다. 브로커는 클라이언트가 암호화되지 않은 TCP 또는 암호화된 TLS(예: HTTPS)를 통해 연결하도록 허용할 수 있습니다.

- **무인증.** 메시지가 중요하지 않고 브로커가 인터넷 클라이언트에 대해 노출되지 않는 경우 로컬 브로커는 인증을 비활성화할 수 있습니다. 이것을 테스트, 샌드박스 개발 및 데모용으로만 사용하는 것이 좋습니다.
- **사용자/암호.** 이것은 가장 일반적인 설정입니다. 시스템이 갖고 있는 위험에 따라, 시스템 관리자는 모든 MQTT 클라이언트가 동일한 사용자/암호를 공유하도록 하거나 제한적 주제 접근 권한을 갖는 사용자를 생성할 수 있습니다.
- **TLS 클라이언트 인증서.** 메시지가 민감한 것으로 분류되는 인터넷 노출 브로커의 경우, 브로커를 유효한 TLS 인증서가 있는 클라이언트만 허용하도록 구성해야 합니다. 이 방식에는 PKI(공개 키 인프라) 및 브로커가 신뢰하는 클라이언트 인증서를 발급할 수 있는 인증 기관이 필요합니다. 공용 MQTT 인터넷 서비스 공급자가 일반적으로 이를 제공합니다.

일부 상황에서는 다중 브로커(로컬 및/또는 공개 브로커)를 통해 서로 다른 사용 사례를 분할하는 것이 적합할 수 있습니다. 중요한 메시지와 중요하지 않은 메시지를 구분하는 것은 보안 제어입니다. 또한 다중 브로커는 단일 실패 지점의 위험을 줄이고 모니터링 및 문제 해결을 향상시킵니다. 비용은 추가 브로커의 추가 배포 및 유지관리입니다.

7 Axis 장치에 내장된 MQTT 클라이언트

표준 VMS 생태계에서 장치의 Axis 이벤트 알림은 전통적으로 RTSP 스트리밍 프로토콜을 사용하는 VAPIX/ONVIF API 인터페이스를 통해 단일 대상으로 스트리밍됩니다.

Axis 장치(AXIS OS 9.80 이상 버전을 실행하는)의 내장 MQTT 클라이언트를 통해 MQTT 프로토콜을 사용하여 동일한 이벤트 알림을 배포할 수 있습니다. 이는 VMS 생태계 내부와 외부 모두에서 적용 가능하며 특히 인터넷을 통해 배포하는 데 유용합니다. MQTT를 통해 네트워크의 다중 구독 MQTT 클라이언트는 Axis 장치에서 게시한 이벤트 알림을 사용하고 처리할 수 있습니다.

8 ACAP 분석 애플리케이션에 내장된 MQTT 클라이언트

특정 시스템, 사용 사례 및 구독자를 위해 고안된 자체 MQTT 클라이언트가 있는 Axis 및 타사 ACAP 앱이 있습니다. Axis Publisher는 일부 시스템에서 요구하는 추가 기능, 구조 및 동작을 추가하는 한 가지 예입니다.

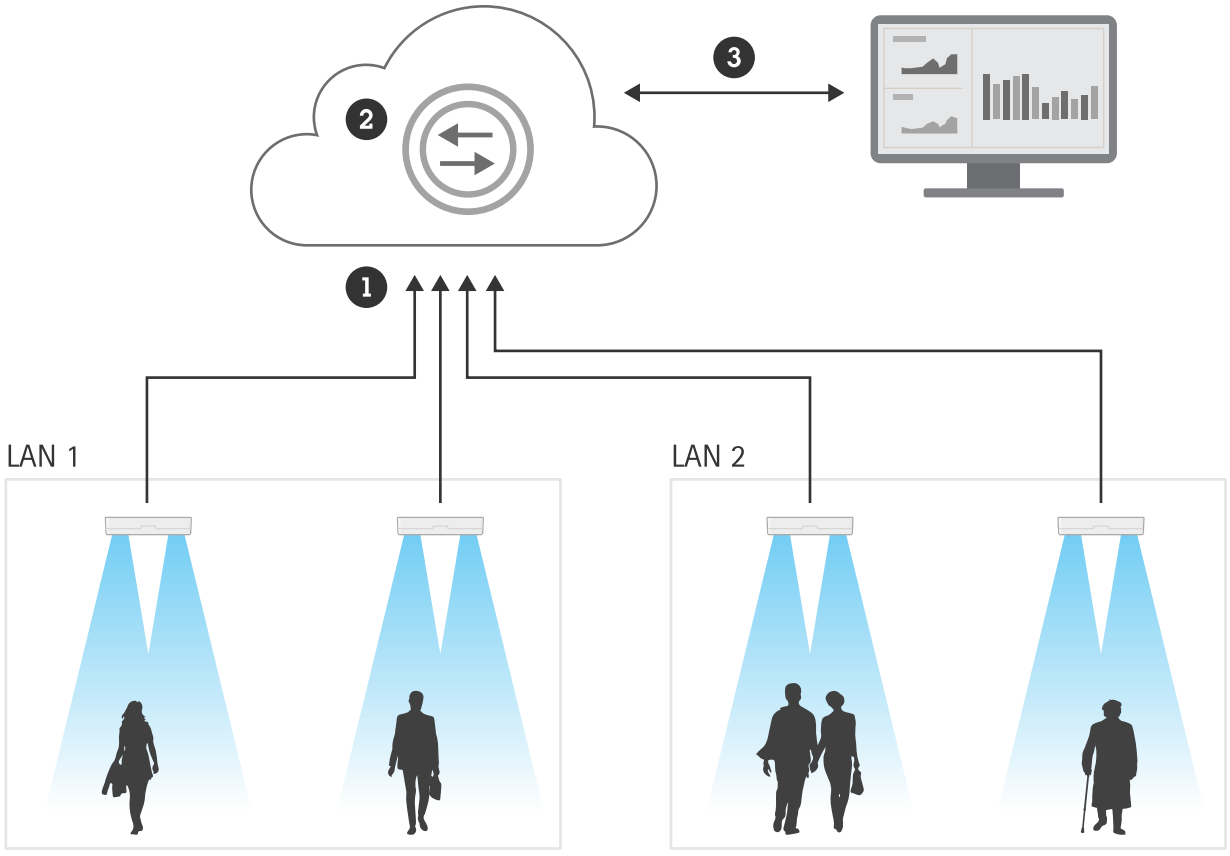
9 기타 MQTT 클라이언트

특정 사용 사례를 위한 도구 또는 서비스로 고안되고 Linux, Windows, Android 및 iOS에 설치할 수 있는 광범위한 종류의 MQTT 클라이언트가 있습니다. MQTT는 Node-RED/Node.js, Python 및 PHP와 같은 스크립팅 및 미들웨어에 매우 적합합니다. Microsoft Azure IoT, AWS™, Google Cloud Platform™과 같은 대부분의 인터넷 서비스 플랫폼은 플랫폼에서 실행되는 서비스에 통합할 MQTT 브로커를 제공합니다 . MQTT 클라이언트가 내장된 센서, 모바일 앱 및 (홈) 오토메이션 시스템이 있습니다.

10 장치를 MQTT와 통합하는 사용 사례의 예

10.1클라우드 플랫폼 대시보드에 인원 계수 분석 데이터 전송

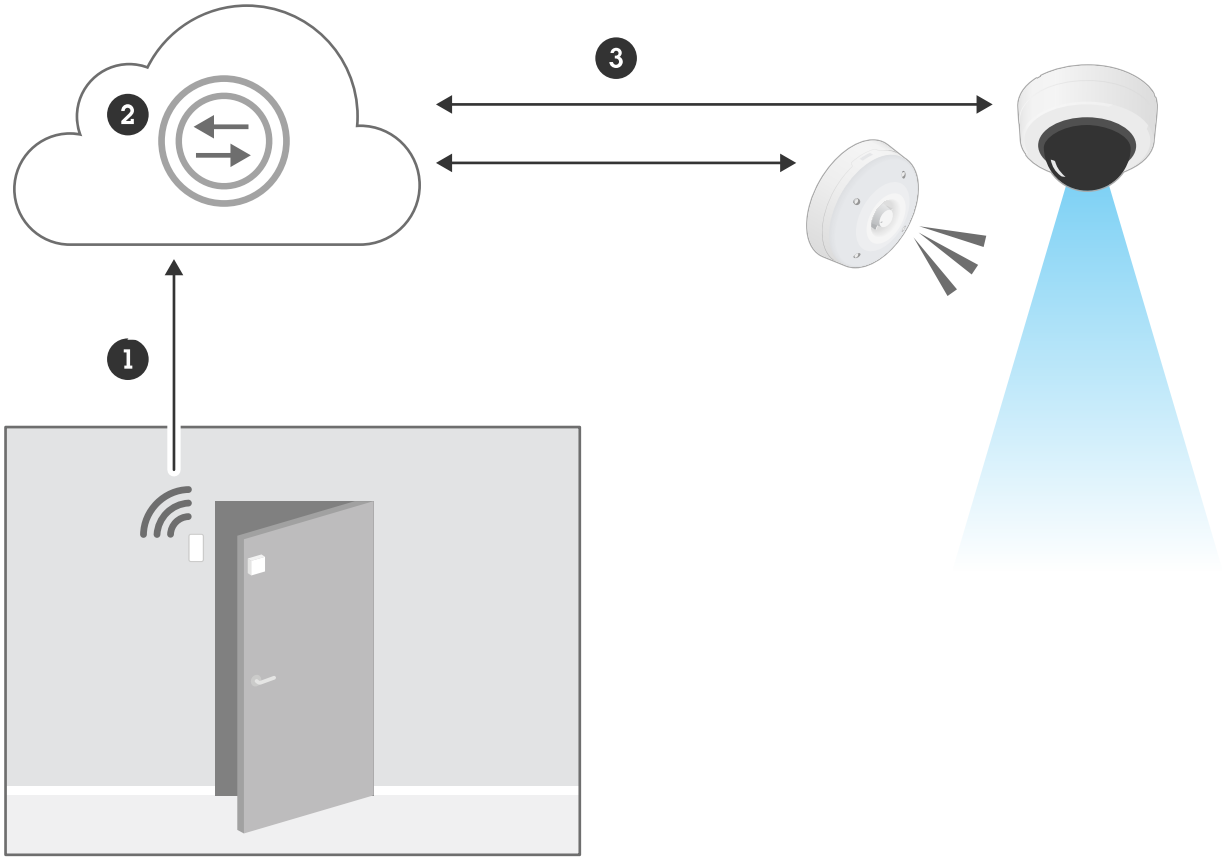
인원 계수 분석 기능이 있는 장치는 사람이 정의된 영역을 "출입"하는 것이 감지될 때마다 이벤트 알림을 생성합니다. 알림은 MQTT 클라이언트로 전달되고, MQTT 클라이언트는 알림을 클라우드 플랫폼에 실시간으로 게시합니다. 클라우드 플랫폼에서는 데이터 시각화 소프트웨어(예: Microsoft® Power BI® 대시보드)에 대한 연결이 생성되어 인원 계수기의 실시간 통계를 표시합니다.



1 게시
2 MQTT 브로커
3 구독

10.2MQTT를 통해 전달되는 도어 센서 데이터가 신호 장치 알람 및 카메라 녹화를 트리거

타사 MQTT 도어 센서는 도어가 열릴 경우 이벤트 알림을 트리거하는 데 사용됩니다. 도어 센서는 클라우드의 MQTT 브로커에 MQTT 메시지를 게시합니다. 신호 장치와 카메라는 도어 센서의 주제를 구독하고 도어가 열리면 알람을 울리고 녹화를 시작합니다.



1 *게시*
2 *MQTT 브로커*
3 *구독*

11 용어집

ACAP	*AXIS Camera Application Platform* - 에지에서 기능과 인텔리전스를 추가하는 애플리케이션용 프레임워크
Aedes	MQTT 브로커
API	*애플리케이션 프로그래밍 인터페이스* - 두 소프트웨어 프로그램이 서로 통신할 수 있도록 하는 코드
AWS™	클라우드 서비스 플랫폼
AXIS OS	Axis의 에지 장치용 운영 체제
CloudMQTT	MQTT 브로커
Eclipse Mosquitto™	MQTT 프로토콜을 구현하는 오픈 소스 메시지 브로커
Google Cloud Platform™	클라우드 서비스 플랫폼
HiveMQ™	MQTT 브로커
HTTP	*하이퍼텍스트 전송 프로토콜(Hypertext Transfer Protocol)* - 월드 와이드 웹에서 사용되는 데이터 전송 프로토콜
IBM Cloud®	클라우드 서비스 플랫폼
IoT	*사물 인터넷(Internet of things)* - 일상적인 장치 및 가전 제품에 내장된 컴퓨팅 장치의 인터넷을 통한 상호 연결
JSON	*자바스크립트 객체 주석(JavaScript Object Notation)* - 압축 파일 형식 및 데이터 교환 형식
Microsoft® Azure® IoT	클라우드 서비스 플랫폼
Microsoft® Power BI®	비즈니스 인텔리전스에 중점을 두는 대화형 데이터 시각화 소프트웨어 프로그램
MQTT	*메시지 대기 텔레메트리 전송(Message Queuing Telemetry Transport)* - 사물 인터넷(IoT)을 위한 메시징 프로토콜
Node.js®	JavaScript 코드 서버측 실행을 위한 오픈 소스 개발 플랫폼
Node-RED®	사물 인터넷 연결을 위한 프로그래밍 도구
ONVIF®	IP 기반 물리적 보안 제품의 효과적인 상호 운용성을 위해 표준화된 인터페이스를 제공하고 홍보하는 개방형 산업 포럼
PHP	웹 개발을 위한 범용 스크립팅 언어
Python®	범용 프로그래밍 언어
RTSP	*실시간 스트리밍 프로토콜(Real-Time Streaming Protocol)* - 엔드포인트 간의 미디어 세션을 설정하고 제어하기 위한 네트워크 프로토콜

TCP	*전송 제어 프로토콜(Transmission Control Protocol)* - 주요 인터넷 프로토콜 중 하나인 데이터 전송 프로토콜
TLS	*전송 레이어 보안(Transport Layer Security)* - 컴퓨터 네트워크를 통한 통신의 기밀성과 무결성을 제공하는 프로토콜
VAPIX®	Axis 제품을 위한 개방형 애플리케이션 프로그래밍 인터페이스(API)
WebSocket	단일 TCP 연결을 통해 양방향 통신 채널을 제공하는 통신 프로토콜
VMS	*영상 관리 소프트웨어* 또는 *영상 관리 시스템*

12 상표 속성

Android 및 Google Cloud Platform은 Google LLC의 상표입니다.

AWS는 미국 및/또는 기타 국가에서 Amazon.com, Inc. 또는 그 계열사의 상표입니다.

Eclipse Mosquitto는 Eclipse Foundation, Inc.의 상표입니다.

HiveMQ는 HiveMQ GmbH의 상표입니다.

IBM 및 IBM Cloud는 전 세계 여러 관할지에 등록된 International Business Machines Corp의 상표입니다.

IOS는 미국 및 기타 특정 국가에서 Cisco Systems, Inc 및/또는 그 계열사의 상표 또는 등록 상표이며 Apple, Inc.의 라이선스 하에 사용됩니다.

JavaScript는 미국 Oracle Corporation의 등록 상표입니다.

Linux는 미국 및 기타 국가에서 Linus Torvalds의 등록 상표입니다.

Microsoft, Windows, Microsoft Azure IoT 및 Microsoft Power BI는 Microsoft Corporation의 등록 상표입니다.

Node.js 및 Node-RED는 미국 및/또는 기타 국가에서 OpenJS Foundation의 등록 상표입니다.

ONVIF는 Onvif, Inc.의 상표입니다.

Python은 Python Software Foundation의 등록 상표입니다.

T10162730/KO/M1.17/2203

Axis Communications 정보

Axis는 보안 및 새로운 비즈니스 성과를 개선하기 위한 솔루션을 창조하여 더 스마트하고 안전한 세상을 가능하게 합니다. 네트워크 기술 회사이자 업계 리더인 Axis는 비디오 감시, 접근 제어, 인터콤, 오디오 시스템 솔루션을 제공합니다. 이러한 솔루션은 지능형 분석 애플리케이션으로 향상되고, 고품질 교육의 지원을 받습니다.

Axis에서는 50개 이상의 나라에 약 4,000명의 전담 직원이 있으며 전 세계 기술 및 시스템 통합 파트너와 협력하여 고객 솔루션을 제공합니다. Axis는 1984년에 설립되었으며 본사는 스웨덴 룬드에 있습니다