BIAŁA KSIĘGA

Integracja urządzeń przy użyciu protokołu MQTT

Marzec 2022

AXIS COMMUNICATIONS

Streszczenie

MQTT to standardowy protokół przesyłania wiadomości, który ułatwia sprawną i niezawodną wymianę danych między urządzeniami IoT oraz aplikacjami w chmurze. Dzięki niemu urządzenia (za pośrednictwem swoich klientów MQTT) mogą publikować wiadomości na wspólnym brokerze (serwerze) MQTT, który pośredniczy w komunikacji z innymi urządzeniami. Broker na bieżąco śledzi, kto co publikuje i kto chce zobaczyć opublikowane dane, przekazując wiadomości wyłącznie do klientów subskrybujących odpowiedni temat.

W typowym ekosystemie VMS powiadomienia o zdarzeniach Axis pochodzące z urządzeń są zazwyczaj przesyłane strumieniowo do jednej lokalizacji docelowej przy użyciu protokołu RTSP za pośrednictwem interfejsu API VAPIX/ONVIF. Jednak te same powiadomienia mogą być dystrybuowane przy użyciu protokołu MQTT za pośrednictwem wbudowanego w urządzenie klienta MQTT (dotyczy urządzeń z systemem AXIS OS 9.80 lub nowszym). Jest to możliwe zarówno w ramach ekosystemu VMS, jak i poza nim, a szczególnie przydaje się w Internecie. Następnie wielu klientów MQTT obecnych w sieci może wykorzystywać i przetwarzać powiadomienia o zdarzeniach publikowane przez urządzenie Axis. Istnieją też aplikacje analityczne ACAP opracowane przez Axis i inne firmy, które posiadają własnego klienta MQTT przeznaczonego do określonych systemów lub zastosowań dla określonych subskrybentów.

Przykładowym zastosowaniem związanym z produktami Axis są urządzenia do zliczania osób, które za pośrednictwem protokołu MQTT mogą wysyłać dane statystyczne do oprogramowania wizualizacyjnego w chmurze. Kolejnym przykładem jest czujnik drzwiowy innego producenta, który komunikuje się w protokole MQTT z urządzeniem sygnalizacyjnym i kamerą, powodując odtworzenie alarmu dźwiękowego oraz rozpoczęcie rejestrowania obrazu po każdym otwarciu drzwi.

Spis treści

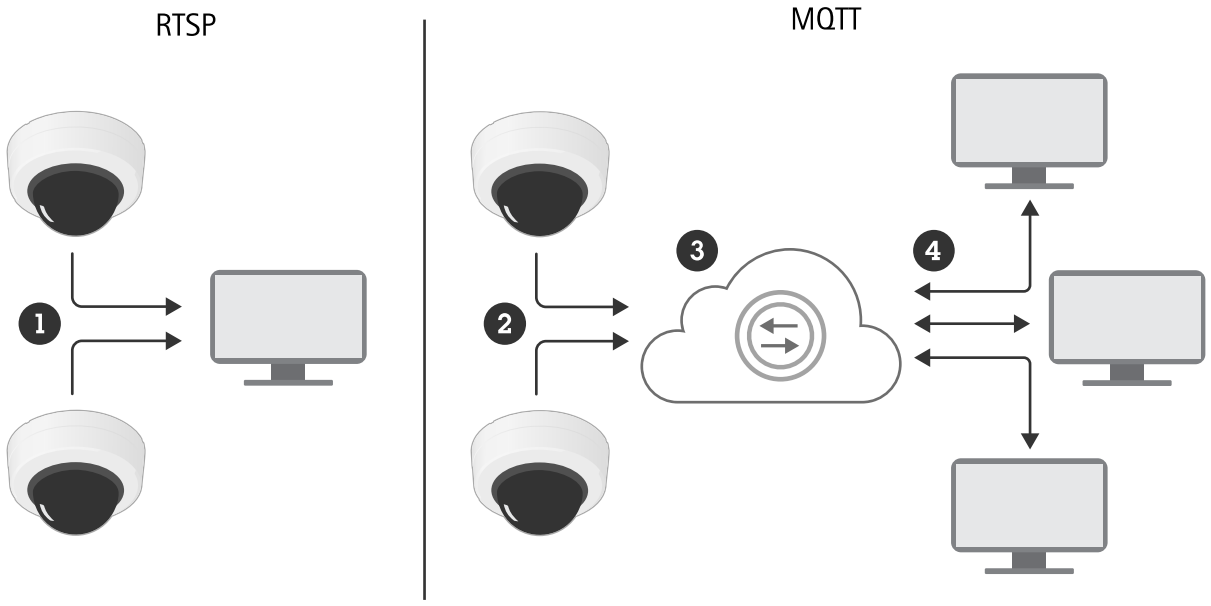
1 Wprowadzenie

MQTT (Message Queuing Telemetry Transport — przesyłanie telemetrii metodą kolejkowania wiadomości) to standardowy protokół przesyłania wiadomości przeznaczony do komunikacji w Internecie rzeczy (IoT). Został zaprojektowany z myślą o uproszczeniu integracji IoT i jest wykorzystywany w wielu branżach do podłączania urządzeń zdalnych przy jednoczesnej minimalizacji objętości kodu i obciążenia sieci. Klient MQTT zawarty w systemie operacyjnym AXIS OS może ułatwić integrację danych i zdarzeń generowanych w urządzeniu z systemami innymi niż systemy zarządzania materiałem wizyjnym (VMS).

W tym dokumencie przedstawiono techniczne omówienie protokołu MQTT, obejmujące jego typowe zastosowania, zalety i ograniczenia. Podano także informacje na temat klientów MQTT stosowanych w urządzeniach Axis i aplikacjach analitycznych ACAP.

2 Protokół MQTT

MQTT to protokół działający w modelu „publikacja/subskrypcja". Oznacza to, że pod względem wzorca przesyłania wiadomości różni się on od protokołu RTSP czy HTTP, które wykorzystują model „zapytanie/odpowiedź". W protokole RTSP jedna strona zgłasza zapytanie, a druga strona na nie odpowiada. Natomiast wiele aplikacji mobilnych do przesyłania wiadomości jest opartych na protokole MQTT lub podobnej koncepcji publikowania i subskrybowania. Istnieją także protokoły typu publikacja/subskrypcja zoptymalizowane pod kątem systemów zamkniętych lub specjalistycznych.



Podczas gdy protokół RTSP pozwala wyłącznie na komunikację typu jeden-do-jednego, MQTT umożliwia także komunikację w modelu jeden-do-wielu i wiele-do-wielu za pośrednictwem brokera.

1 *Strumień zdarzeń*
2 *Publikowanie*
3 *Broker MQTT*
4 *Subskrybowanie*

Zgodnie z koncepcją MQTT wszyscy klienci łączą się ze wspólnym brokerem (serwerem) MQTT, który na bieżąco śledzi, kto co publikuje i kto chce zobaczyć te dane. Połączenie jest najczęściej sesją TCP

obsługiwaną na porcie 1883. Klient może także nawiązać połączenie za pośrednictwem protokołu TLS (zazwyczaj port 8883) lub WebSocket (zazwyczaj port 1884/8884).

Klienci publikują wiadomości z określonym tematem. Inni klienci mogą zasubskrybować ten temat lub użyć symboli wieloznacznych, aby uzyskać dostęp do wszystkich tematów podrzędnych. Wiadomość zawiera także treść (tzw. ładunek), którą najczęściej stanowi struktura danych JSON, ciąg znaków, a nawet krótkie dane binarne. Podmiot publikujący nie wie, czy inni klienci subskrybują jego wiadomości. Broker przekazuje wiadomości tylko do klientów, którzy subskrybują określone tematy.

Sposób działania protokołu MQTT przypomina wysłanie artykułu do czasopisma. Artykuł mogą przeczytać wszystkie osoby prenumerujące czasopismo, a komunikacja odbywa się w modelu jeden-do-jednego lub jeden-do-wielu (w protokole MQTT dostępna jest także komunikacja wiele-do-wielu). Artykuł można również przeczytać długo po dacie jego pierwotnej publikacji.

Natomiast protokół RTSP pod względem sposobu działania bardziej przypomina połączenie telefoniczne. Jest jeden nadawca i jeden odbiorca poleceń, a komunikacja zawsze odbywa się w modelu jeden-do-jednego. Jeśli zamierzony rozmówca nie odbierze połączenia, wiadomość do niego nie dotrze.

Gdy do dystrybuowania powiadomień o zdarzeniach Axis pochodzących z urządzeń używany jest protokół MQTT, powiadomienia te może wykorzystywać i przetwarzać wielu klientów MQTT obecnych w sieci. Jest to rozwiązanie znacznie korzystniejsze od tradycyjnej metody (polegającej na używaniu interfejsu programowania aplikacji (API) VAPIX®/ONVIF® i protokołu RTSP), zgodnie z którą powiadomienia o zdarzeniach byłyby przesyłane strumieniowo do jednej lokalizacji docelowej.

3 Korzyści

Korzystanie z protokołu MQTT wiąże się z szeregiem korzyści. Poniżej wymieniono główne zalety protokołu MQTT w porównaniu z protokołem typu zapytanie/odpowiedź, jak na przykład RTSP:

- **Mniejsze ryzyko ujawnienia haseł urządzeń.** Klient nie potrzebuje dostępu do urządzenia ani serwera, aby uzyskać dane. Oznacza to, że klient nie musi znać hasła ani zasad działania interfejsu API. Ogranicza to ryzyko ujawnienia haseł urządzeń klientom i użytkownikom, a tym samym ryzyko wystąpienia nadużyć.
- **Pojedynczy punkt integracji.** Każdy autoryzowany klient może uzyskać dostęp do wiadomości opublikowanych przez wszystkich innych klientów za pośrednictwem jednego połączenia z brokerem. W przypadku protokołu RTSP klient musi indywidualnie połączyć się z każdym klientem, od którego chce uzyskać dane. Oznacza to, że protokół MQTT umożliwia przepływ wiadomości w modelach jeden-do-jednego, jeden-do-wielu lub wiele-do-jednego bez dodatkowego obciążania poszczególnych klientów.
- **Publikowanie i odbiór wiadomości bez naruszania zapory sieciowej.** W protokole RTSP interfejs urządzenia/serwera musi być dostępny dla klienta. Jeśli urządzenie znajduje się za zaporą, a klient jest zdalny, zaporę należy skonfigurować tak, aby zezwalała na żądania przychodzące, co ujawnia interfejs urządzenia komunikującego się przez API. Gdy pośrednikiem jest publiczny broker MQTT, klienci znajdujący się za zaporą mogą publikować/subskrybować szczegółowe dane bez potrzeby naruszania zapory sieciowej (jeśli zezwala ona na połączenia wychodzące).
- **Jakość serwisu (QoS).** W przypadku wiadomości o znaczeniu krytycznym podmiot publikujący może sprawdzić, czy została ona odebrana przez innego klienta, a jeśli nie — podjąć alternatywne kroki.
- **Zachowywanie wiadomości.** Podmiot publikujący może oznaczyć wiadomość jako do zachowania, dzięki czemu broker zachowa kopię tej wiadomości i będzie ją wysyłał nowo łączącym się klientom subskrybującym dany temat.

- **Dostępność klientów IoT.** Dostępne są pakiety klientów MQTT do wszystkich popularnych środowisk tworzenia oprogramowania, w tym Windows®, Linux®, Android™, iOS, Node.js®, PHP i Python®. W porównaniu z komunikacją w oparciu o strumień RTSP do brokera MQTT może podłączyć się znacznie więcej klientów.
- **Uproszczone monitorowanie wiadomości i debugowanie.** Istnieje kilka narzędzi MQTT, za pomocą których można monitorować wszystkie opublikowane wiadomości, a także publikować wiadomości wymagające rozwiązania problemu zależnie od reakcji subskrybentów.
- **Uproszczona struktura danych.** Ponieważ w protokole MQTT często obsługiwani są nieznani klienci, zazwyczaj treść wiadomości jest upraszczana na potrzeby subskrybenta.

4 Ograniczenia

W porównaniu z innymi protokołami MQTT ma pewne minusy:

- **Pojedynczy punkt awarii.** Niedostępność brokera sprawia, że nie można przesyłać żadnych wiadomości. Jednak można zaprojektować system w sposób taki, by obejmował brokery nadmiarowe.
- **Kto opublikował wiadomość?** Protokół MQTT z zasady koncentruje się na temacie wiadomości, a nie na tym, kto ją opublikował. Jeśli podmiot publikujący nie umieści w temacie lub treści jakiegoś identyfikatora, poznanie nadawcy wiadomości wymaga uzyskania dostępu do dziennika brokera. Dlatego przyjęła się praktyka, zgodnie z którą w zależności od zastosowania podmiot publikujący dołącza identyfikator klienta do tematu lub treści wiadomości.
- **Złośliwy klient** połączony z brokerem może publikować/odbierać wiadomości na dowolny temat, do którego ma uprawnienia. Dlatego należy zapewnić ochronę brokera (zobacz sekcję poświęconą bezpieczeństwu protokołu MQTT).
- Protokół ten **nie jest przeznaczony do ciągłego strumieniowego przesyłania materiału wizyjnego/dźwiękowego**.

Tak jak w przypadku każdego rozwiązania opartego na serwerze należy wziąć pod uwagę całkowitą przepustowość. W przypadku bardzo dużych systemów z wieloma klientami może być wymagane dynamiczne skalowanie.

5 Infrastruktura

Dość łatwo można skonfigurować lokalnego brokera Eclipse Mosquitto™ lub wykorzystać narzędzie Node-RED® jako brokera lokalnego, takiego jak Aedes. Ponadto szereg dostawców usług internetowych i innych podmiotów udostępnia zarządzane brokery MQTT, takie jak Microsoft® Azure® IoT, HiveMQ™, CloudMQTT oraz IBM® Cloud®.

Jeśli system nie posiada klientów zdalnych, zaleca się korzystanie z brokera lokalnego. Ponadto broker lokalny może działać jako proxy do brokera publicznego albo zostać skonfigurowany tak, aby działał jako proxy dla wybranych wiadomości brokera lokalnego i wybranych wiadomości brokera publicznego.

6 Bezpieczeństwo

Broker wymaga odpowiedniej ochrony w zależności od tego, jak krytyczne wiadomości są przesyłane oraz jakie zagrożenia mogą wystąpić w konkretnym systemie. Protokół MQTT udostępnia kilka schematów uwierzytelniania, na przykład brak uwierzytelniania, logowanie na podstawie nazwy użytkownika i hasła lub

certyfikat TLS klienta. Różni użytkownicy mogą mieć różne uprawnienia do publikowania i otrzymywania wiadomości w zależności od tematu. Broker może zezwalać klientom na korzystanie z niezaszyfrowanego połączenia TCP lub zaszyfrowanego połączenia TLS (na przykład HTTPS).

- **Brak uwierzytelniania.** Broker lokalny może wyłączyć uwierzytelnianie, jeśli wiadomości nie są krytyczne, a broker nie ma styczności z klientami łączącymi się przez internet. To rozwiązanie jest zalecane wyłącznie do testów, prac programistycznych w odizolowanym środowisku i prezentacji.
- **Logowanie na podstawie nazwy użytkownika i hasła.** Jest to najczęstsza konfiguracja. W zależności od poziomu zagrożeń systemu jego administrator może zezwolić wszystkim klientom MQTT na korzystanie z tej samej nazwy użytkownika i hasła lub utworzyć użytkowników z ograniczonym dostępem do tematów.
- **Certyfikat TLS klienta.** Jeśli broker jest połączony z Internetem, a wiadomości zakwalifikowano jako poufne, konfiguracja brokera powinna zezwalać wyłącznie na połączenia z klientami mającymi ważny certyfikat TLS. Ten sposób działania wymaga infrastruktury klucza publicznego i ośrodka certyfikacji, który może wystawiać wiarygodne dla brokera certyfikaty klientów. Takie rozwiązanie zazwyczaj oferują publiczni dostawcy usług internetowych korzystający z protokołu MQTT.

W pewnych sytuacjach dobrym rozwiązaniem może być podzielenie na grupy i użycie wielu brokerów — lokalnych i/lub publicznych. Sprawdzonym mechanizmem kontroli bezpieczeństwa jest oddzielenie wiadomości krytycznych od pozostałych. Stosowanie wielu brokerów zmniejsza też ryzyko związane z pojedynczym punktem awarii oraz usprawnia monitorowanie i rozwiązywanie problemów. Dodatkowym kosztem takiego rozwiązania jest potrzeba wdrożenia i utrzymywania dodatkowych brokerów.

7 Klient MQTT w urządzeniach Axis

W standardowym systemie VMS powiadomienia o zdarzeniach pochodzące z urządzeń Axis są zazwyczaj przesyłane do jednej lokalizacji docelowej przy użyciu protokołu RTSP za pośrednictwem interfejsu API VAPIX/ONVIF.

Takie same powiadomienia o zdarzeniach mogą być dystrybuowane przy użyciu protokołu MQTT za pomocą klienta MQTT wbudowanego w urządzenie Axis (wyposażone w system AXIS OS 9.80 lub nowszy). Jest to możliwe zarówno w ramach systemu VMS, jak i poza nim, a szczególnie przydaje się w Internecie. Protokół MQTT sprawia, że wielu klientów MQTT w sieci może wykorzystywać i przetwarzać powiadomienia o zdarzeniach publikowane przez urządzenie Axis.

8 Klienci MQTT w aplikacjach analitycznych ACAP

Istnieją aplikacje ACAP opracowane przez Axis oraz inne firmy, które posiadają własnego klienta MQTT przeznaczonego do określonych systemów lub zastosowań dla określonych subskrybentów. Jako przykład można wymienić Axis Publisher, który zawiera dodatkowe funkcje, struktury i zachowania potrzebne w niektórych systemach.

9 Inni klienci MQTT

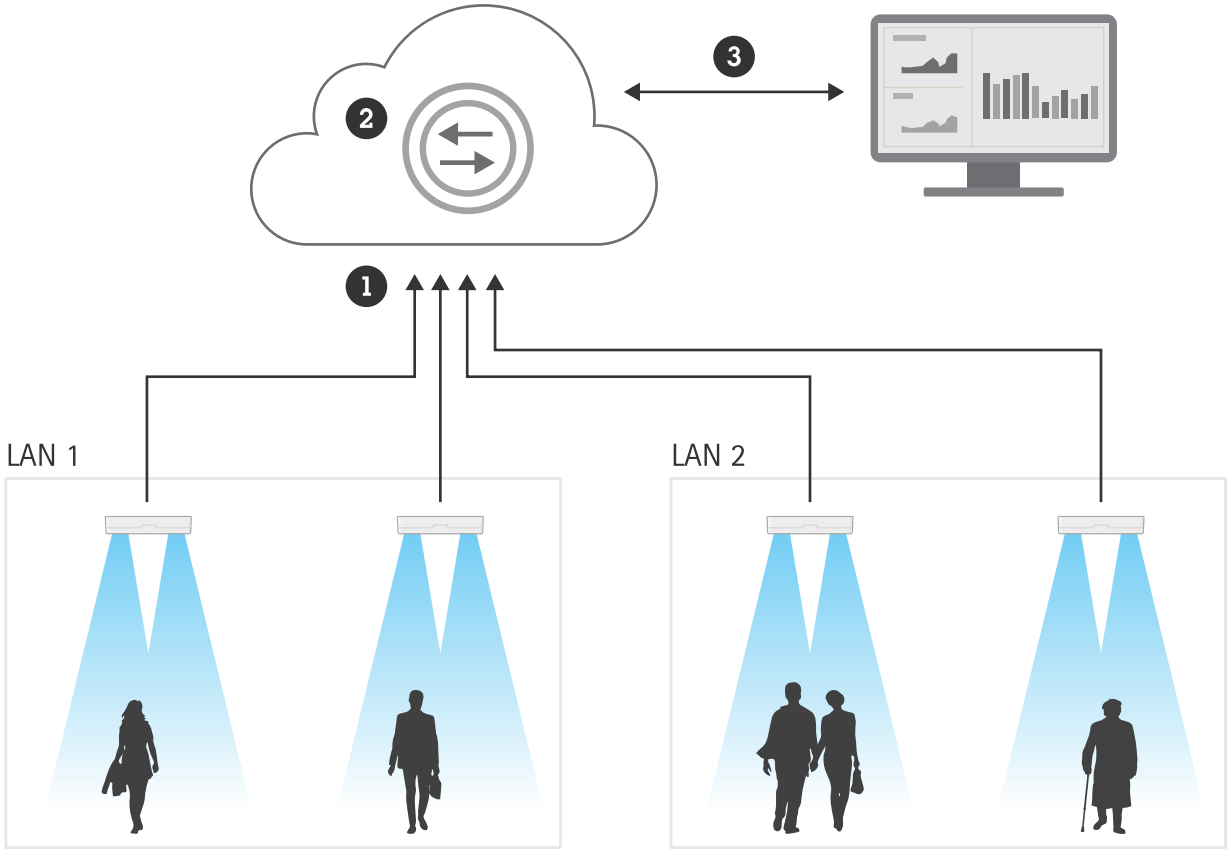
Istnieje szeroki wachlarz klientów MQTT możliwych do zainstalowania w systemach Linux, Windows, Android oraz iOS. Rozwiązania te zostały stworzone jako narzędzia lub usługi do określonych zastosowań. Protokół MQTT bardzo dobrze nadaje się do tworzenia skryptów i oprogramowania middleware, na przykład w oparciu o Node-RED/Node.js, Python czy PHP. Większość platform usług internetowych, takich jak

Microsoft Azure IoT, AWS™ i Google Cloud Platform™, oferuje brokery MQTT przeznaczone do integracji z usługami działającymi na danej platformie. Wiele czujników, aplikacji mobilnych i systemów automatyki posiada wbudowany moduł klienta MQTT.

10 Przykłady integracji urządzeń przy użyciu protokołu MQTT

10.1 Przesyłanie danych analitycznych dotyczących zliczania osób na pulpit platformy działającej w chmurze

Urządzenie z aplikacją analityczną do zliczania osób generuje powiadomienie o zdarzeniu po każdym wykryciu czyjego wejścia na zdefiniowany obszar lub wyjścia z tego obszaru. Powiadomienie jest przekazywane do klienta MQTT, który publikuje je w czasie rzeczywistym na platformie w chmurze. Na platformie w chmurze tworzone jest połączenie z oprogramowaniem do wizualizacji danych (na przykład pulpitem Microsoft® Power BI®) na potrzeby bieżącego wyświetlania danych statystycznych pochodzących z liczników osób.

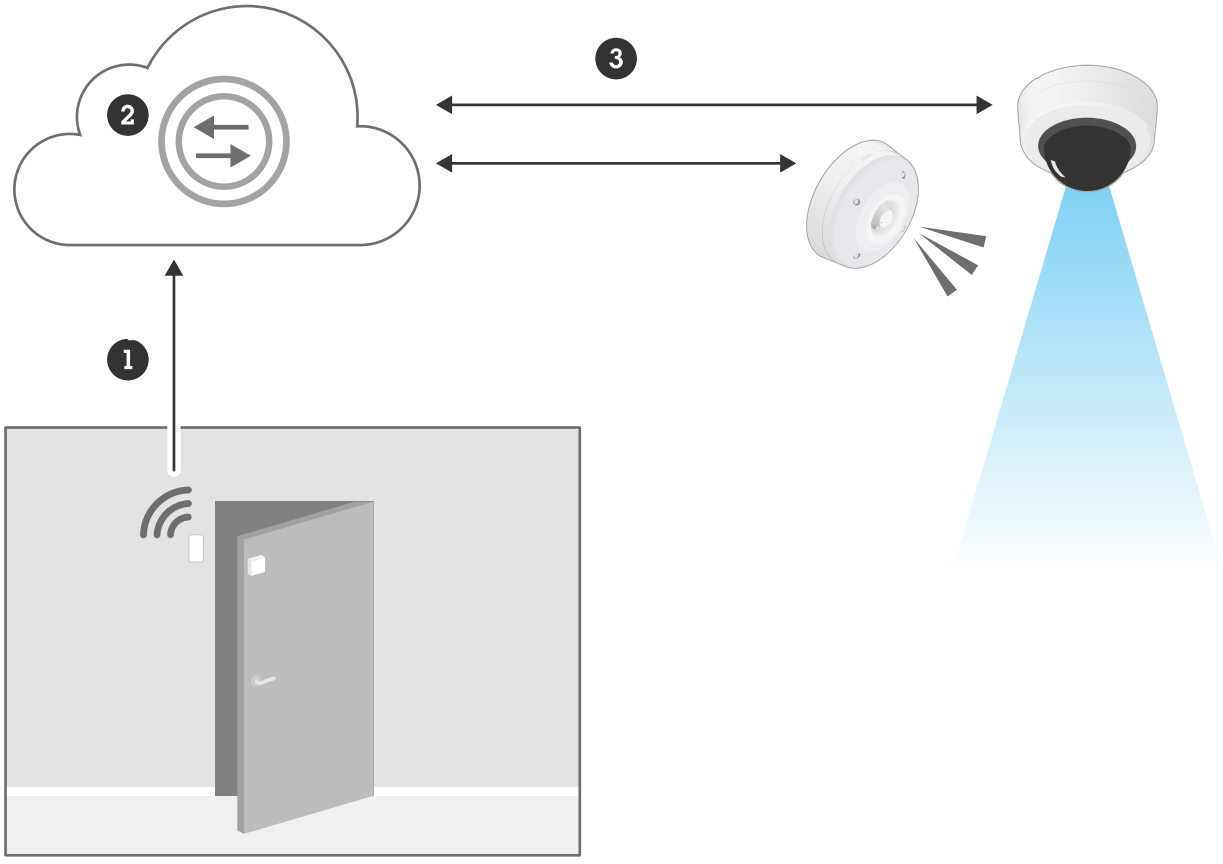


1 *Publikowanie*
2 *Broker MQTT*
3 *Subskrybowanie*

10.2 Dane z czujnika drzwiowego przesyłane przy użyciu protokołu MQTT wyzwalają alarm urządzenia sygnalizacyjnego i rejestrowanie w kamerze

Czujnik drzwiowy MQTT innej firmy wyzwala powiadomienie o zdarzeniu w przypadku otwarcia drzwi. Czujnik publikuje wiadomość MQTT w brokerze MQTT w chmurze. Urządzenie sygnalizacyjne i kamera

subskrybują temat związany z czujnikiem drzwiowym, dzięki czemu w chwili otwarcia drzwi odtwarzają alarm dźwiękowy i rozpoczynają rejestrowanie.



1 *Publikowanie*
2 *Broker MQTT*
3 *Subskrybowanie*

11 Glosariusz

ACAP	*AXIS Camera Application Platform*, platforma aplikacji zwiększających funkcjonalność i inteligencję urządzeń brzegowych
Aedes	Broker MQTT
API	*Application programming interface* (interfejs programowania aplikacji), kod umożliwiający dwóm programom wzajemną komunikację
AWS™	Platforma usług w chmurze
AXIS OS	System operacyjny urządzeń brzegowych Axis
CloudMQTT	Broker MQTT
Eclipse Mosquitto™	Otwartoźródłowy broker wiadomości, w którym zaimplementowano protokół MQTT
Google Cloud Platform™	Platforma usług w chmurze
HiveMQ™	Broker MQTT
HTTP	*Hypertext Transfer Protocol*, protokół przesyłu danych używany w sieci WWW
IBM Cloud®	Platforma usług w chmurze
IoT	*Internet of things* (Internet rzeczy), ogólnoświatowa sieć wzajemnych połączeń między urządzeniami komputerowymi zawartymi w urządzeniach codziennego użytku
JSON	*JavaScript Object Notation*, kompaktowy format plików i format wymiany danych
Microsoft® Azure® IoT	Platforma usług w chmurze
Microsoft® Power BI®	Interaktywne oprogramowanie do wizualizacji danych ze szczególnym naciskiem na analizy biznesowe
MQTT	*Message Queuing Telemetry Transport*, protokół przesyłania wiadomości przeznaczony do komunikacji w Internecie rzeczy
Node.js®	Otwartoźródłowa platforma programistyczna służąca do wykonywania kodu JavaScript po stronie serwera
Node-RED®	Narzędzie programistyczne służące do tworzenie połączeń w Internecie rzeczy
ONVIF®	Otwarte forum branżowe tworzące i promujące standardowe interfejsy, które zapewniają skuteczne współdziałanie produktów z zakresu bezpieczeństwa fizycznego opartych na protokole IP
PHP	Język skryptowy ogólnego przeznaczenia służący do tworzenia stron internetowych
Python®	Język programowania ogólnego przeznaczenia
RTSP	*Real-Time Streaming Protocol*, protokół sieciowy służący do nawiązywania i kontrolowania sesji przesyłania multimediów między punktami końcowymi
TCP	*Transmission Control Protocol*, protokół przesyłu danych należący do głównych protokołów internetowych
TLS	*Transport Layer Security*, protokół zapewniający poufność i integralność komunikacji w sieciach komputerowych
VAPIX®	Otwarty interfejs programowania aplikacji (API) do produktów Axis

WebSocket	Protokół komunikacji zapewniający dwukierunkowe kanały komunikacyjne w ramach jednego połączenia TCP
VMS	*Video management software/system* (oprogramowanie/system zarządzania materiałem wizyjnym)

12 Znaki towarowe

Android i Google Cloud Platform są znakami towarowymi firmy Google LLC.

AWS jest znakiem towarowym firmy Amazon.com, Inc. lub jej podmiotów zależnych w Stanach Zjednoczonych i/lub innych krajach.

Eclipse Mosquitto jest znakiem towarowym organizacji Eclipse Foundation, Inc.

HiveMQ jest znakiem towarowym firmy HiveMQ GmbH.

IBM i IBM Cloud są znakami towarowymi firmy International Business Machines Corp., zarejestrowanymi w wielu jurysdykcjach na świecie.

IOS jest znakiem towarowym lub zastrzeżonym znakiem towarowym firmy Cisco Systems, Inc i/lub jej podmiotów zależnych w Stanach Zjednoczonych i niektórych innych krajach oraz jest używany na podstawie licencji przez firmę Apple, Inc.

JavaScript jest zastrzeżonym znakiem towarowym firmy Oracle Corporation w Stanach Zjednoczonych.

Linux jest zastrzeżonym znakiem towarowym Linusa Torvaldsa w Stanach Zjednoczonych i innych krajach.

Microsoft, Windows, Microsoft Azure IoT i Microsoft Power BI są zastrzeżonymi znakami towarowymi firmy Microsoft Corporation.

Node.js i Node-RED są zastrzeżonymi znakami towarowymi organizacji OpenJS Foundation w Stanach Zjednoczonych i/lub innych krajach.

ONVIF jest znakiem towarowym organizacji Onvif, Inc.

Python jest zastrzeżonym znakiem towarowym organizacji Python Software Foundation.

T10162730/PL/M1.17/2203

O firmie Axis Communications

Axis umożliwia tworzenie mądrzejszego i bezpieczniejszego świata, tworząc rozwiązania zwiększające bezpieczeństwo i wydajność biznesową. Jako firma z branży technologicznej będąca liderem na rynku, Axis oferuje systemy dozoru wizyjnego, kontroli dostępu, domofonowe i rozwiązania audio. Rozwiązania te są wzbogacone o inteligentne aplikacje analityczne i wysokiej jakości szkolenia

Firma Axis zatrudnia około 4000 zaangażowanych pracowników w ponad 50 krajach i współpracuje z partnerami z sektora technologii oraz integracji systemów na całym świecie, aby dostarczać rozwiązania dla klientów. Firma Axis powstała w 1984 roku, a jej siedziba znajduje się w Lund w Szwecji