ホワイトペーパー

MQTTによるデバイス統合

3月 2022

概要

MQTTは、標準のメッセージングプロトコルです。これを活用することで、IoTデバイスとクラウドアプリケーション間の効率的かつ高信頼性のデータ交換が容易に実現します。これにより、デバイスから（MQTTクライアント経由で）、他のデバイスとの通信の仲介機能を果たす共通のMQTTブローカー（サーバー）にメッセージを配信することが可能となります。ブローカーにより発信者、発信内容、受信者が追跡され、ブローカーからサブスクライブクライアントに適切なトピックのメッセージが転送される仕組みとなっています。

典型的なVMSエコシステムでは、デバイスからのAxisイベント通知は、従来、RTSPストリーミングプロトコルを使用して、VAPIX/ONVIF APIインターフェース経由で単一の宛先にストリーミングされます。しかし、MQTTプロトコルを使用すれば、同じ通知をデバイスの組み込みMQTTクライアント経由で配信することができます（AXIS OS 9.80以降のバージョンを実行しているデバイスに適用可能）。これは、VMSエコシステムの内外部で実行することが可能で、インターネット上では特に有用です。ネットワークに存在する複数のMQTTサブスクライブクライアントが、Axisデバイスから配信されたイベント通知を使用および処理できるようになります。特定のシステム、ユースケース、サブスクライバー（受信者）向けに設計された独自のMQTTクライアントが備わっているAxisおよびサードパーティ製のACAP分析アプリケーションも存在します。

Axis製品に関連するユースケースの例を挙げるならば、人数計測デバイスからMQTT経由でクラウド内のデータ視覚化ソフトウェアに統計を送信することが可能です。別の例として、MQTT経由で信号装置（シグナリング・デバイス）/カメラと通信し、ドアの開閉に応じてアラームの発信と録画を実行するサードパーティ製ドアセンサーが挙げられます。

目次

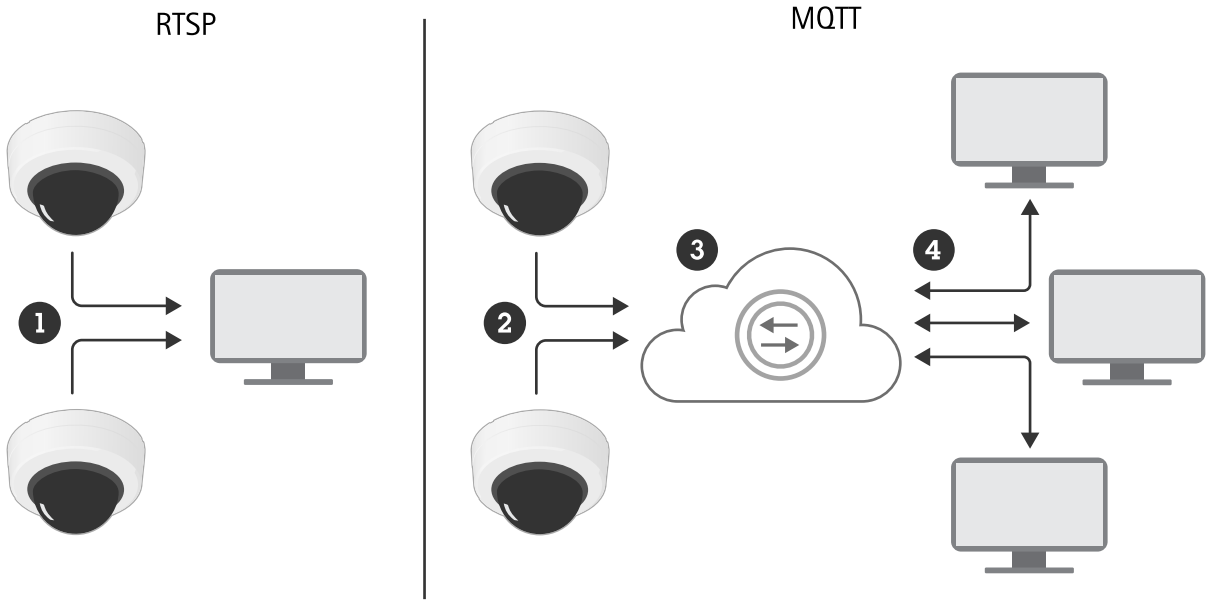
1 はじめに

MQTT（Message Queuing Telemetry Transport）は、モノのインターネット（IoT）で使われる標準の通信プロトコルです。IoTの統合を簡素化するために設計されており、小さなコードフットプリントと最小限のネットワーク帯域幅でリモートデバイスを接続するために、さまざまな業界で使用されています。AXIS OSのMQTTクライアントを利用することで、デバイスで生成されたデータとイベントをビデオ管理システム（VMS）以外のシステムに簡単に統合することができます。

本ホワイトペーパーでは、一般的なユースケース、メリット、制限といったMQTTの技術的概要についてご説明します。これには、AxisデバイスとACAP分析アプリのMQTTクライアントに関する詳細も含まれています。

2 MQTTプロトコル

MQTTはパブリッシュ/サブスクライブ（出版/購読）プロトコルです。そのため、このメッセージングパターンは、要求/応答プロトコルであるRTSPやHTTPとは異なります。RTSPの場合は、一方から要求が発行され、他方が応答する仕組みとなります。しかし、多くのモバイルメッセージングアプリは、MQTTまたは同様のパブリッシュ/サブスクライブ型モデルに基づいています。クローズドシステムや特定システムに最適化されたパブリッシュ/サブスクライブ型のプロトコルも存在します。



RTSPでは一対一の通信しか許可されませんが、MQTTの場合は、ブローカー経由で一対多または多対多の通信を実行することが可能となります。

1 イベントストリーム
2 パブリッシュ
3 MQTTブローカー
4 サブスクライブ

すべてのクライアントが共通のMQTTブローカー（サーバー）に接続するというのが、MQTTのコンセプトです。このブローカーにより、発信者、発信内容、受信者が追跡されます。通

常、ポート1883のTCPセッションで接続します。TLS（通常はポート8883）またはWebSocket（通常はポート1884/8884）を使用して、クライアントから接続することもできます。

一方のクライアントからトピックの入ったメッセージが配信されます。他方のクライアントでその特定のトピックがサブスクライブされます。または、ワイルドカードを使用して、すべてのサブトピックを取得することも可能です。また、メッセージには、通常JSONデータの構造、文字列、または短いバイナリデータのペイロードが含まれます。パブリッシャー（発信者）側には、他のクライアントがサブスクライブしているかどうかは未知の情報となります。ブローカーからは、トピックをサブスクライブしているクライアントにのみメッセージが転送されます。

MQTTの仕組みは、雑誌への記事の投稿に多少似ていると言えます。雑誌の購読者はその記事を読むことができ、これは一対一または一対多のいずれかの通信となります（MQTTの場合は、多対多の通信も可能です）。最初に記事が公開された時点からかなり時間を経ても、その記事を読むことが可能です。

対称的に、RTSPの仕組みは、電話での通話に例えることができます。コマンドには1つのソースと1つのターゲットが存在し、通信は常に一対一となります。ターゲットが電話に出なければ、ターゲットはそのメッセージを見逃します。

MQTTを使用してデバイスから配信されるAxisイベント通知は、ネットワーク内の複数のMQTTサブスクライブクライアントにより使用および処理されます。イベント通知が1つの宛先にのみ配信される従来型の方法（VAPIX®/ONVIF®アプリケーションプログラミングインターフェース [API] とRTSPを使用）に比べて、MQTTには大きなメリットがあります。

3 メリット

MQTTを使用することで、いくつかのメリットがもたらされます。RTSPなどの要求/応答プロトコルを使用する場合と比較して、MQTTには以下のような主要メリットがあります。

- **デバイスのパスワードが公開されるリスクの削減：** データを取得する上で、クライアントからデバイスやサーバーにアクセスする必要がありません。つまり、クライアント側にパスワードやAPIの仕組みに関する情報や知識が必要ないということです。これにより、デバイスのパスワードがクライアントやユーザーに公開されるリスクが軽減されるため、故意的または偶発的な誤用のリスクも削減されます。
- **単一の統合ポイント：** これが許可されている場合は、すべてのクライアントはブローカーへの単一の接続で、発信者側のクライアントすべての配信メッセージを取得することができます。RTSPの場合は、クライアントから、必要なデータを送信しているそれぞれのクライアントに接続する必要があります。つまり、MQTTでは、各クライアントに余計な負担をかけずに、一対一、一対多、多対一のメッセージフローが実現するということです。
- **完全なファイアウォールを使用したパブリッシュ/サブスクライブ：**RTSPの場合は、クライアントからデバイス/サーバーAPIにアクセスできるようになっている必要があります。デバイスがファイアウォールの背後に配置されており、クライアントがリモートの場合は、着信要求が許可されるようにファイアウォールを構成して、デバイスAPIを公開する必要があります。パブリックMQTTブローカーを仲介機能として間に挟むと、ファイアウォールの背後にあるクライアントの場合も、ファイアウォールに穴を開けることなく、特定のデータをパブリッシュ/サブスクライブすることが可能となります（ファイアウォールがアウトバウンド接続を許可している場合）。

- **QoS（Quaity of Service）：** 重要なメッセージを配信する場合、パブリッシャーはそのメッセージが受信側のクライアントにより受信されたかどうかを監視し、受信されていない場合は別の措置を実行することができます。
- **メッセージの保持：** パブリッシャーはメッセージを「保持」として指定することができます。そうすると、ブローカーでメッセージのコピーが保持され、当該トピックをサブスクライブしているクライアントから新たな接続があった場合にそのメッセージがそのクライアントに送信されます。
- **IoTクライアントの可用性：**MQTTクライアントパッケージは、Windows®、Linux®、Android™、iOS、Node.js®、PHP、Python®など、あらゆる一般的なソフトウェア開発環境で利用することができます。デバイスにRTSPデータストリームを設定する場合よりも、より多くのクライアントがブローカーに接続することができます。
- **メッセージの監視とデバッグの簡素化：** 配信済みメッセージに対するサブスクライバーの反応の有無や反応の状況を監視するため、およびメッセージをトラブルシューティングするために使用できるMQTTツールがいくつか存在します。
- **データ構造の簡略化：** 多くの場合、MQTTは未知のクライアントをターゲットにするため、通常、これを考慮に入れて、メッセージペイロードがサブスクライバーのために簡素化されます。

4 制限

他の同様のプロトコルと比較すると、MQTTにはいくつかの欠点があります。

- **単一障害点：** ブローカーが利用不可の状態に陥ると、すべてのメッセージングが機能しなくなります。しかし、インフラストラクチャーが冗長ブローカーで設計されている可能性があります。
- **メッセージ発信者の不明瞭性：** 設計上、MQTTではトピックに焦点が当てられ、メッセージの発信者はそれほど重視されません。パブリッシャーがトピックやペイロードの一部としてIDを含めない限り、メッセージの発信者を確認するには、ブローカーのログにアクセスしなければなりません。一般的に、ユースケースに基づいて、パブリッシャーはトピックやペイロードにクライアントIDを含めます。
- **悪意のあるクライアント：**ブローカーに接続して認定済みのトピックをパブリッシュ/サブスクライブしているのが悪質なクライアントである可能性があります。そのため、ブローカーを保護する必要があります（MQTTセキュリティのセクション参照）。
- **継続的なビデオ/オーディオストリーミング向けの設計ではありません。**

他のサーバーと同様に、総帯域幅スループットを考慮する必要があります。多くのクライアントを持つ非常に大規模なシステムの場合は、動的なスケーリングが必要になる可能性があります。

5 インフラストラクチャー

ローカルのEclipse Mosquitto™ブローカーを設定すること、またAedesなど、ローカルブローカーとして機能するようにNode-RED®を有効化するのは非常に簡単です。Microsoft® Azure® IoT、HiveMQ™、CloudMQTT、IBM® Cloud®など、マネージドMQTTブローカーを提供しているインターネットサービスプロバイダーや他のプロバイダーも多数存在します。

システムにリモートクライアントがない場合は、ローカルブローカーを使用することが勧められます。パブリックブローカーのプロキシとして機能するように、または特定のローカルブローカーメッセージやパブリックブローカーメッセージのプロキシとして機能するようにローカルブローカーを構成することもできます。

6 セキュリティ

メッセージの重要度および特定のシステムで発生し得る脅威に基づいて、ブローカーに適切な保護対策を講じる必要があります。MQTTでは、認証なし、ユーザー/パスワード、TLSクライアント証明書認証など、いくつかの認証スキームが提供されています。ユーザーによって、パブリッシュ/サブスクライブできるトピックに関する承認が異なる可能性があります。暗号化されていないTCPまたは暗号化されたTLS（HTTPSなど）経由でクライアントから接続できるようにブローカーを設定することができます。

- **認証なし：** メッセージの重要度が低く、ブローカーがインターネットクライアントに公開されていない場合は、ローカルブローカーで認証を無効にすることができます。この設定は、テスト、サンドボックス開発、デモにのみ使用することが勧められます。
- **ユーザー/パスワード：**これが最も一般的な設定です。システムのリスクの程度に応じて、システム管理者はすべてのMQTTクライアントに同じユーザー/パスワードを設定すること、またはトピックへのアクセスを制限するユーザーを設定することができます。
- **TLSクライアント証明書：** メッセージが機密として分類される場合で、ブローカーがインターネットに公開される場合は、有効なTLS証明書を備えたクライアントのみが許可されるようにブローカーを構成する必要があります。このスキームには、PKI（公開鍵インフラストラクチャー）と認証局（CA）が必要となります。これにより、ブローカーが信頼できるクライアント証明書が発行されます。通常、パブリックMQTTインターネットサービスプロバイダーがこのサービスを提供しています。

状況によっては、ローカルブローカーとパブリックブローカーのいずれかまたは両方の複数のブローカーを用いて、異なるユースケースをセグメント化することが適切な場合もあります。重要なメッセージと重要でないメッセージをセグメント化するのは、セキュリティ制御の一環です。複数のブローカーを配置することで、単一障害点のリスクが削減され、監視とトラブルシューティングが強化されます。追加ブローカーの展開や保守に追加コストがかかります。

7 AxisデバイスのMQTTクライアント

標準のVMSエコシステムでは、デバイスからのAxisイベント通知は、従来、RTSPストリーミングプロトコルによりVAPIX /ONVIFAPIインターフェース経由で単一の宛先に配信されていました。

MQTTプロトコルを使用して、（AXIS OS 9.80以降のバージョンを実行している）Axisデバイスの組み込みMQTTクライアント経由で同じイベント通知を配信することができます。これは、VMSエコシステムの内外部の両方に適用可能で、インターネット上で特に有用です。MQTTを使用することで、Axisデバイスから配信されたイベント通知は、ネットワーク内の複数のMQTTサブスクライブクライアントにより使用および処理されるようになります。

8 ACAP分析アプリケーションのMQTTクライアント

特定のシステム、ユースケース、サブスクライバー向けに設計された独自のMQTTクライアントが備わっているAxisおよびサードパーティ製のACAP分析アプリケーションも存在します。一例として、Axis Publisherが挙げられます。これにより、一部のシステムに必要な機能、構造、動作を追加することができます。

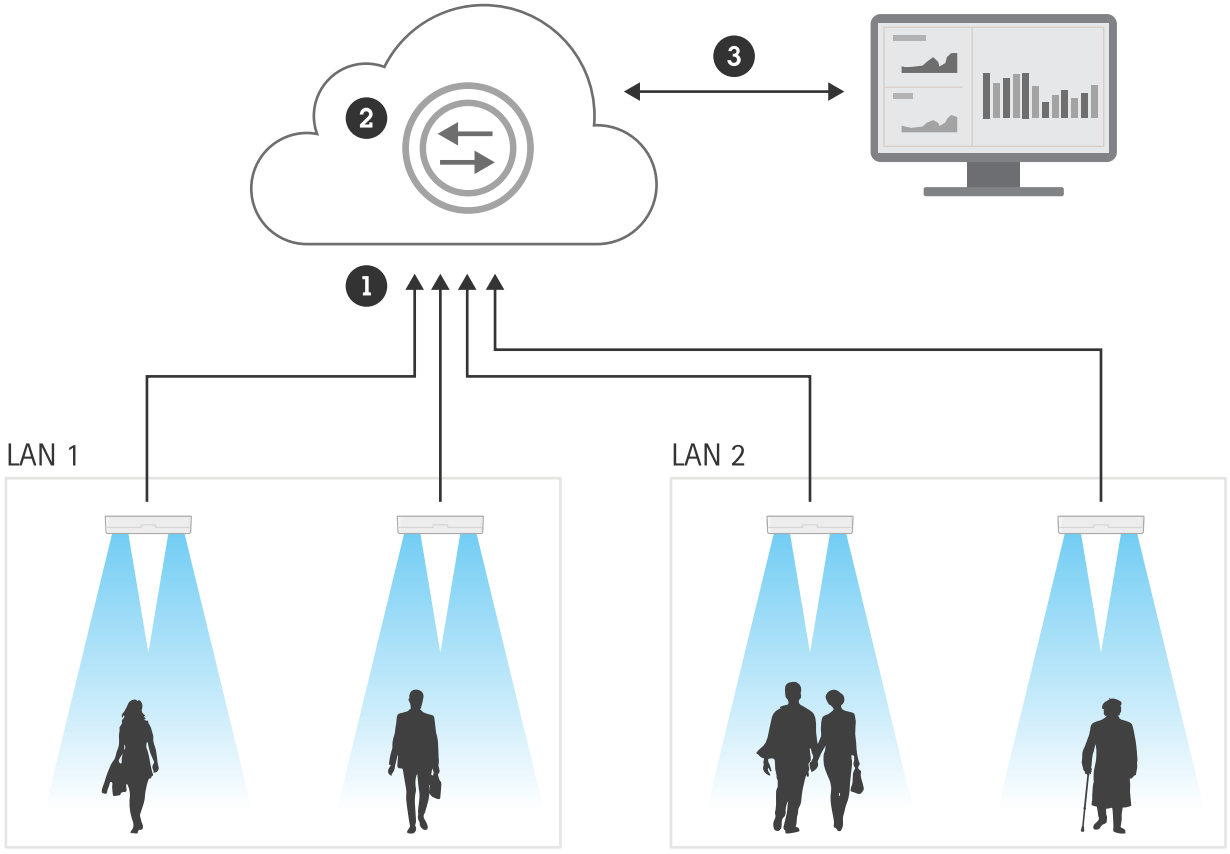
9 その他のMQTTクライアント

特定のユースケース向けのツールやサービスとして設計されているMQTTクライアントにはさまざまな種類があり、Linux、Windows、Android、iOSにインストールできるものも多くあります。MQTTは、Node-RED/Node.js、Python、PHPといったスクリプトやミドルウェアに非常に適しています。Microsoft Azure IoT、AWS™、Google CloudPlatform™など、大半のインターネットサービスプラットフォームでは、プラットフォームで実行されるサービスに統合できるMQTTブローカーが提供されています。センサーやモバイルアプリ、および（ホーム）自動化システムにはすべてMQTTクライアントがあります。

10 MQTTによるデバイス統合のユースケース例

10.1人数計測分析データをクラウドプラットフォームダッシュボードに送信

人数計測分析機能が搭載されているデバイスを使用すると、定義された領域を「出入り」する人物が検出されるたびにイベント通知が生成されます。通知はMQTTクライアントに送信され、MQTTクライアントから通知がクラウドプラットフォームにリアルタイムで配信されます。そして、クラウドプラットフォームでデータ視覚化ソフトウェア（Microsoft® Power BI®ダッシュボードなど）への接続が構築されて、人数計測分析からのリアルタイムの統計が表示されます。

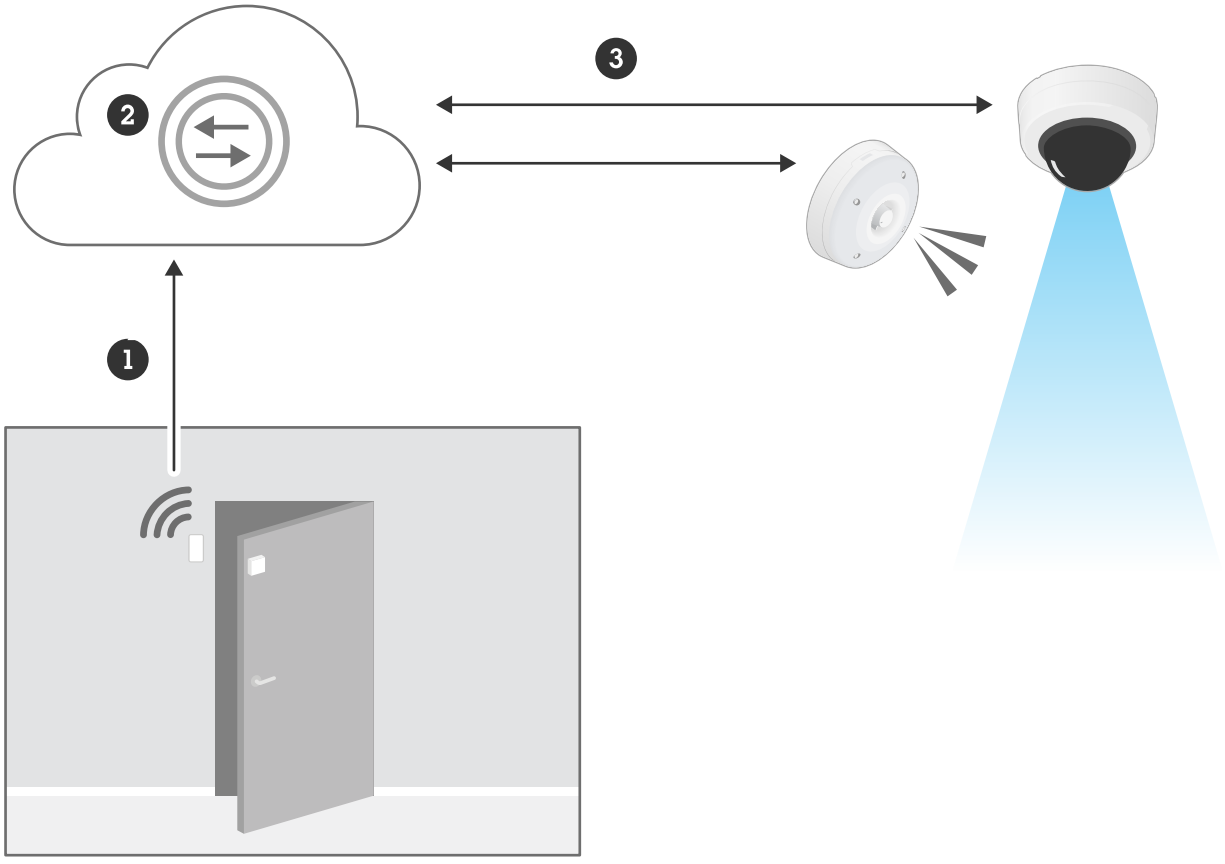


1 パブリッシュ
2 MQTTブローカー
3 サブスクライブ

10.2MQTT経由のドアセンサーデータにより、信号装置のアラーム発信とカメラ録画をトリガー

ドアが開いた際にイベント通知がトリガーされるように、サードパーティ製のMQTTドアセンサーを使用するとします。この場合、ドアセンサーからクラウド内のMQTTブローカーに

MQTTメッセージが配信されます。信号装置とカメラをドアセンサーのトピックにサブスクライブすると、ドアが開いた際にアラームが再生され、録画が開始されます。



1 パブリッシュ
2 MQTTブローカー
3 サブスクライブ

11 用語集

ACAP	*AXIS Camera Application Platform*。エッジでの機能とインテリジェンスを追加するアプリケーションのフレームワーク
Aedes	MQTTブローカー
API	*アプリケーションプログラミングインターフェイス*。2つのソフトウェアプログラムの相互通信を実現するコード
AWS™	クラウドサービスプラットフォーム
AXIS OS	Axisのエッジデバイスのオペレーティングシステム
CloudMQTT	MQTTブローカー
Eclipse Mosquitto™	MQTTプロトコルを実装するオープンソースのメッセージブローカー
Google Cloud Platform™	クラウドサービスプラットフォーム
HiveMQ™	MQTTブローカー
HTTP	*ハイパーテキスト転送プロトコル*。ワールドワイドウェブで使用されるデータ転送プロトコル
IBM Cloud®	クラウドサービスプラットフォーム
IoT	*モノのインターネット*。日常的に使用されるデバイスやアプライアンスに組み込まれたコンピューティングデバイスのインターネット経由の相互接続
JSON	*JavaScript Object Notation*。コンパクトなファイル形式＆データ交換形式
Microsoft® Azure® IoT	クラウドサービスプラットフォーム
Microsoft® Power BI®	ビジネスインテリジェンスに焦点を当てたインタラクティブなデータ視覚化ソフトウェアプログラム
MQTT	*Message Queuing Telemetry Transport*。モノのインターネットのメッセージングプロトコル
Node.js®	JavaScriptコードのサーバーサイドを実行するためのオープンソース開発プラットフォーム
Node-RED®	モノのインターネットを相互接続するためのプログラミングツール
ONVIF®	IPベースの物理的セキュリティ製品の効果的な相互運用性を実現することを目的として、標準化されたインターフェースの提供と促進を行うオープンな業界フォーラム
PHP	ウェブ開発に使用する汎用スクリプト言語
Python®	汎用プログラミング言語
RTSP	*リアルタイム・ストリーミング・プロトコル*。エンドポイント間のメディアセッションを確立および制御するためのネットワークプロトコル
TCP	*トランスミッション・コントロール・プロトコル*。主要なインターネットプロトコルの1つであるデータトランスポートプロトコル

TLS	*トランスポート・レイヤー・セキュリティ*。コンピューターネットワーク経由の通信の機密性と整合性を提供するプロトコル
VAPIX®	Axis製品のオープンアプリケーションプログラミングインターフェイス（API）
WebSocket	単一のTCP接続経由で双方向コミュニケーションチャンネルを提供する通信プロトコル
VMS	*ビデオ管理ソフトウェア*または*ビデオ管理システム*

12 商標の帰属

AndroidおよびGoogle Cloud Platformは、Google LLCの商標です。

AWSは、米国およびその他の国におけるAmazon.com, Inc.またはその関連会社の商標です。

Eclipse Mosquittoは、Eclipse Foundation, Inc.の商標です。

HiveMQは、HiveMQ GmbHの商標です。

IBMおよびIBM Cloudは、世界各地の多くの法域で登録されているInternational Business Machines Corpの商標です。

IOSは、米国およびその他の国におけるCisco Systems, Inc.および/またはその関連会社の商標または登録商標であり、Apple, Inc.のライセンスに基づき使用されています。

JavaScriptは、米国におけるOracle Corporationの登録商標です。

Linuxは、米国およびその他の国におけるLinus Torvaldsの登録商標です。

Microsoft、Windows、Microsoft Azure IoT、およびMicrosoft Power BIは、Microsoft Corporationの登録商標です。

Node.jsおよびNode-REDは、米国および/またはその他の国におけるOpenJS Foundationの登録商標です。

ONVIFは、Onvif, Inc.の商標です。

Pythonは、Python Software Foundationの登録商標です。

T10162730/ja/M1.17/2203

Axis Communicationsについて

Axisはセキュリティとビジネスパフォーマンスを向上させるソリューションを生み出すことで、 よりスマートで安全な世界の実現を目指しています。 ネットワークテクノロジー企業として、 また業界のリーダーとして、Axisはビデオ監視、 アクセスコントロール、インターコム、音声システムなどのソリューションを提供しています。 これらのソリューションは インテリジェントな分析アプリケーションによって強化され、高品質のトレーニングに支えられています。

Axisは50ヶ国以上に約4,000人の熱意にあふれた従業員を擁し、 世界中のテクノロジーおよびシステムインテグレーションパートナーと連携することで、 カスタマーソリューションをお届けしています。 Axisは1984年に設立され、本社は スウェーデンのルンドにあります。