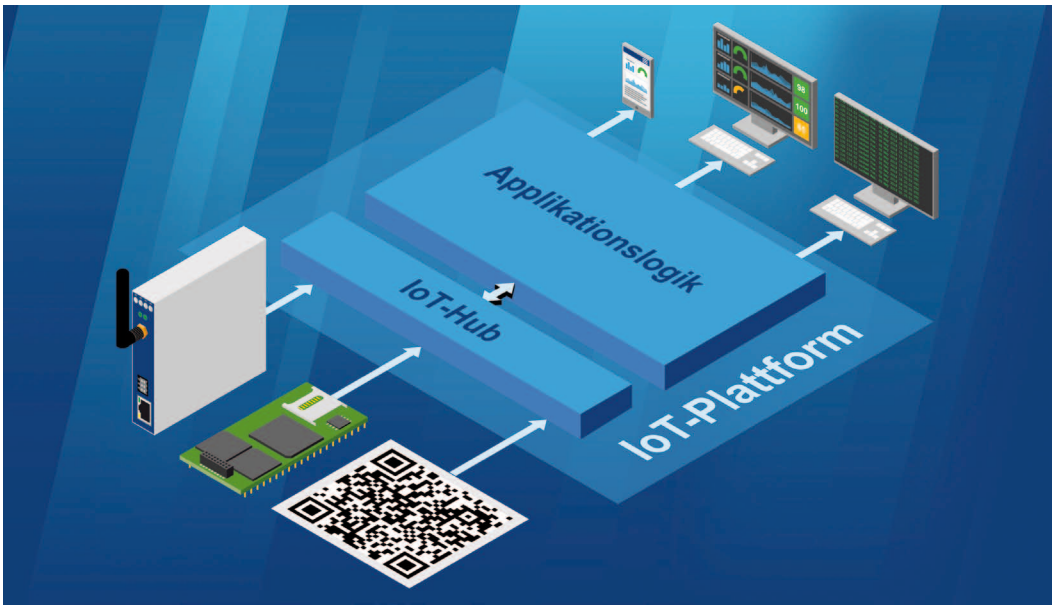
Produktinnovation durch IoT-Plattformintegration

Digitale Plattformen haben das Potenzial, sowohl ganze Branchen als auch Produkte vollständig zu verändern. Trotz allem verhält sich die Anbieterseite in der Automatisierung nach wie vor sehr zurückhaltend, besonders wenn es um IoT-Plattformintegrationen geht. Den Anwendern bleibt im Moment nur das IoT-Retrofitting.



Die Problematik

Der Funktionsumfang führender Cloud- und IoT-Plattformen entwickelt sich seit einigen Jahren in einem rasanten Tempo weiter. Geräte- und Maschinenhersteller, die weiterhin darauf verzichten, diese Funktionen für ihre Produkte zu nutzen, verlieren umgekehrt proportional zu diesem Trend an Wettbewerbsfähigkeit. Insofern ist jeder in der Automatisierung, der Baugruppen, Geräte oder Maschinen vermarktet, in dem ein Stück Software zum Einsatz kommt, gut beraten, sich intensiver mit den technischen Details und Anwendungsmöglichkeiten der IoT-Plattformen von Amazon, Microsoft, Siemens und Co. auseinanderzusetzen und seine Produkte auf den aktuellen Stand der digitalen Technik zu bringen.

Autor:
Klaus-Dieter Walter ist CEO bei der SSV Software Systems GmbH

SSV Software Systems GmbH
www.ssv-embedded.de

Bausteine einer IoT-Plattform

IoT-Plattformen bestehen aus verschiedenen Schichten, die je nach Anbieter unterschiedlich ausgeprägt sind. Trotz zum Teil völlig verschiedener Namensgebung lassen sich in den meisten Plattformen funktional bedingt zumindest zwei Ebenen identifizieren:

1. Für die Anbindung der einzelnen Geräte, Sensoren und Aktoren (IoT-Devices) existiert in der Regel ein IoT-Hub bzw. Device Gateway als zentraler Nachrichtenhub, über den die gesamte Kommunikation zwischen Plattform und IoT-Devices abgewickelt wird und in dem auch die IT-Sicherheit des Übertragungskanals realisiert ist.

2. Alle anderen Bausteine einer Plattform sind in der Applikationslogik zusammengefasst. Dazu gehören zum einen die zahlreichen Standarddienste (Services) der jeweiligen Plattform (z. B. verschiedene Datenbanken, diverse Laufzeitumgebungen für Apps, Datenanalysedienste mit KI-Algorithmen, Benachrichtigungsfunktionen usw.) sowie zum anderen eine Rules Engine (Data Flow Engine) als Datenverarbeitungslogik der ein- und ausgehenden IoT-Device-Daten. Für die Services und die Rules Engine werden darüber hinaus Softwarebausteine (Anwendungen) entwickelt, um die Einzelfunktionen der Standarddienste aufgabenbezogen miteinander zu verknüpfen.

Durch den IoT-Hub hat die gesamte Anwendung eine sternförmige Topologie mit den einzelnen Geräten als Endpunkten. Ein solcher Hub bildet die Verteilerfunktion zwischen allen Geräten einer Anwendung und den Services der IoT-Plattform. Insofern lassen sich die wichtigsten Anforderungen in drei Themen gliedern:

Sicherheit:

Als zentraler Kommunikationsendpunkt für alle IoT-Devices einer Anwendung ist im IoT-Hub auch die IT-Security zur Absicherung aller Verbindungen zu finden. Dazu gehört die Device-Authentifizierung (jedes Gerät, dass über den IoT-Hub kommunizieren darf, muss zuvor sicher identifiziert werden), das Gewährleisten einer vertraulichen (verschlüsselten) Datenübertragung und das Sicherstellen der Datenintegrität. Darüber hinaus ist eine Security-Managementfunktion erforderlich, die zum einen die Verwaltung der gerätebasierten Authentifizierung ermöglicht und zum anderen die vollständige Kontrolle für den gesamten Gerätezugriff auf die IoT-Plattform gewährleistet.

Kommunikationsvielfalt:

Ein IoT-Hub muss mehrere Kommunikationsmuster unterstützen. In der Praxis wird nicht nur der Klassiker „Gerät-an-Plattformanwendung“ als Kommunikationsrichtung benötigt. Auch „Plattformanwendung-an-Gerät“ und „Gerät-an-Gerät“ wird in vielen Anwendungen benötigt. Dabei sind selbstverständlich die unterschiedlichen Sicherheitsanforderungen zu beachten. Datenübertragungen von der Plattform zu einem Gerät (also IoT-Device-Schreibzugriffe), sind auf jeden Fall deutlich riskanter, als die Kommunikation in umgekehrter Richtung.

Skalierbarkeit:

Im Extremfall muss ein IoT-Hub auch mit Millionen gleichzeitig verbundener Geräte sicher und zuverlässig funktionieren, die Zigmillionen von Ereignissen je Sekunde erzeugen. Die Herausforderung ist dabei ein möglichst konstantes Zeitverhalten für den Datendurchsatz, unabhängig davon, wie viele aktive Geräte gerade mit einem Hub verbunden sind und welche Nachrichtenanzahl dadurch verursacht wird.

AWS IoT als praktisches Beispiel

AWS IoT (AWS IoT Core) wird als Cloud-basierte IoT-Plattform angeboten. Sie ermöglicht beliebigen Geräten über eine per TLS 1.2 gesicherte Internetverbindung die Kommunikation bzw. Interaktion mit AWS-Cloudanwendungen. Genaugenommen ist AWS IoT eine zusätzliche Funktionsebene, die den bereits seit 2006 verfügbaren Amazon Web Services nachträglich vorgeschaltet wurde, um die AWS an die besonderen Anforderungen des Internets der Dinge anzupassen.

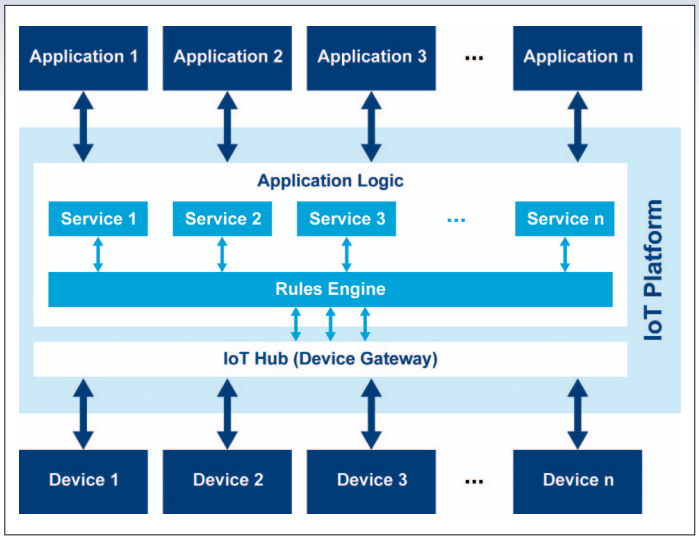


Bild 1: IoT-Plattformen sind mehrschichtig aufgebaut. Über einen Hub werden die Geräte als IoT-Devices sternförmig angebunden. Mit Hilfe der Applikationslogik, die aus Standardservices und individuell erstellten Anwendungen besteht, werden aus den Device-Daten Informationen gewonnen. Zusätzlich gibt es eine Rules Engine, die nach dem Datenflussprinzip von ein- und ausgehenden Daten durchlaufen wird und programmierbare Aktionen auslösen kann.

Aus Sicht der AWS-IoT-Architekten sammeln IoT-Anwendungen entweder Daten in der Cloud, um sie mit speziellen Anwendungen im Umfeld der AWS-Standardfunktionen zu bearbeiten. Des Weiteren sind auch Anwendungsszenarien denkbar, in denen eine AWS-Applikation direkt mit IoT-Devices kommuniziert, z. B. um sie per Alexa Voice Service (AVS) fernzusteuern. Insofern gehört ein sehr umfangreiches Sicherheitssystem mit einem vielfältigen Rechtemanagement zur Plattform.

Als IoT-Hub

kommt eine modulare Lösung zum Einsatz. Die Schnittstelle zu den Geräten bildet ein sogenanntes Device Gateway, mit dem alle IoT-Devices einer Anwendung per TLS verbunden sind. Wegen der Besonderheiten des TLS-Protokolls besteht für MQTT und WebSockets die Möglichkeit, relativ langlebige Verbindungen herzustellen, um nicht immer wieder einen erneuten (zeit- und datenintensiven) TLS-Handshake zu verursachen. Zur Entgegennahme und zum Versenden von Nachrichten per MQTT steht eine Publish/Subscribe-Brokerfunktion (Message Broker) zur Verfügung. Sie unterstützt mittels einer speziellen Serverfunktion auch die Nachrichtenübermittlung per HTTP-POST-Request (ein solcher HTTP-Request wird funktional als MQTT-Publish behandelt).

Leistungsfähigkeit

In Bezug auf die Leistungsfähigkeit von Device Gateway und Message Broker weist Amazon darauf hin, dass diese Funktionseinheiten als verwaltete Services vollständig automatisch skalieren und auch für sehr große Projekte mit mehr als einer Milliarde IoT-Devices geeignet sind. Amazon hebt in diesem Zusammenhang hervor, dass der Message Broker auch mit Millionen gleichzeitig bestehender MQTT-Verbindungen umgehen und z. B. eine eingehende Publish-Nachricht an Millionen Subscriber verteilen kann.

Device Shadow

Eine Besonderheit des AWS-IoT-Hubs ist der Device Shadow. Dieser Funktionsbaustein dient als virtuelles Datenabbild (virtuelle Repräsentanz) für jedes mit dem Device Gateway verbundene Gerät, in dem der jeweils letzte Gerätestatus bis zu einem Jahr lang gespeichert wird. Diese virtuelle Repräsentanz wird durch ein JSON-basiertes Dokument gebildet (max. 8 KBytes pro Gerät). Auf die JSON-Daten können z. B. andere AWS-Anwendungen oder externe Mobilgeräte mit einem speziellen REST-API zugreifen. Der Device Shadow wird von Zeit zu Zeit mit dem jeweiligen Gerät synchronisiert. Dabei werden neue (Zustands-) Daten vom Gerät in das JSON-Dokument geschrieben (Reported State), aber auch Variable ausgelesen und an das Gerät übertragen (Desired State), was zu einem geänderten Gerätezustand führen kann.

Die AWS-IoT-Applikationslogik besteht im Wesentlichen aus einer speziellen Rules Engine und den

AWS-Servicename	Beschreibung
DynamoDB	NoSQL-basierter Datenbankdienst, der sowohl Schlüsselwert- (Key-Value) als auch Dokumentdatenstrukturen unterstützt.
EC2	EC2 dient als Abkürzung für Elastic Compute Cloud. Dieser Name steht in der Amazon-Cloud für skalierbare Rechnerkapazität, um eigene Anwendungen auf virtuellen Computern in der Cloud auszuführen.
Kinesis	Skalierbarer Datenstreaming-Dienst für Echtzeitanalysen. Mit Hilfe solcher Datenanalysen lassen sich z. B. Anomalien in IoT-Datenströmen erkennen.
Lambda	Ereignisgesteuerter Dienst, der speziellen Code als Reaktion auf bestimmte Ereignisse ausführt. AWS Lambda zählt zur Kategorie des sogenannten Serverless Computing, da sich der Anwender nur um den Code für die Lambda-Funktion und die Ereignisverknüpfung dieser Funktion, nicht aber um die Serverressourcen kümmern muss.
S3	S3 steht für Simple Storage Service. Hinter dem Namen verbirgt sich eine sehr mächtige Datenspeicherinfrastruktur für AWS-Cloudanwendungen. S3 erlaubt die Speicherung und den Abruf von Datenobjekten nahezu beliebiger Größe. S3 lässt sich z. B. mit AWS Lambda kombinieren, um bei der Änderung eines Speicherobjekts eine Lambda-Funktion aufzurufen.
SageMaker	Dieser Dienst ermöglicht das Erstellen und Nutzen von Machine-Learning-Modellen, um z. B. von IoT-Geräten eintreffende Daten zu analysieren und mittels Regression oder Klassifizierung automatisierte Entscheidungen zu treffen (Beispiel: Automatische Gesichtserkennung per Deep Learning, um einer bestimmten Person den Zutritt zu einem Gebäudeteil zu gestatten).
SNS	Einfacher Benachrichtigungsdienst (SNS = Simple Notification Service) für AWS-Cloudanwendungen. SNS ermöglicht das Senden und Empfangen von SMS und E-Mail. Unterstützt den Nachrichtenversand an HTTP(S)-Endpunkte sowie spezielle Push-Nachrichten an iOS- und Android-Mobilgeräte. Per SNS kann z. B. auch die Ausführung einer Lambda-Funktion ausgelöst werden.
SQS	SQS ist die Abkürzung für Simple Queue Service. Dieser Dienst ermöglicht den Aufbau von Nachrichtenwarteschlangen (z. B. FIFO-Warteschlangen), um verteilte Cloudanwendungen miteinander kommunizieren zu lassen.

Tabelle 1: Kurzübersicht der wichtigsten Amazon Web Services (AWS), die von einer IoT-Anwendung genutzt werden können. Bei jedem einzelnen Dienst handelt es sich um eine relativ komplexe Anwendung, deren Praxiseinsatz allerdings zum Teil umfangreiche Spezialkenntnisse erfordert.

Stufe	Merkmal	Beschreibung
1	Einfaches technisches Produkt	Zum Beispiel elektrische Ständerbohrmaschine für Werkstätten. Die wesentlichen Bedienelemente sind der Ein- und Ausschalter für die Netzspannung des internen Elektromotors sowie ein gusseisernes Handrad, um den Bohrkopf in vertikaler Richtung zu bewegen. Elektronik und Software kamen in dieser frühen Phase noch nicht zum Einsatz.
2	Smartes Produkt	Durch den Einzug der Mikroelektronik hat die Bohrmaschine nun eine interne Drehzahlsteuerung und in der Gehäusefrontplatte ein Bedienpanel mit LCD. Der Benutzer kann zwischen verschiedenen Drehzahlen wählen und den Bohrwerkzeugen jeweils einen eigenen Betriebsstunden- und Bohrvorgangszähler zuweisen, um sie rechtzeitig auszutauschen.
3	Smartes, kommunikationsfähiges Produkt	Die Bohrmaschine bietet inzwischen eine Ethernet-LAN- oder Funkschnittstelle für die Kommunikation mit einem PC, Smartphone oder Tablet sowie einen elektronisch gesteuerten Vertikalantrieb für den Bohrkopf. Bohrprogramme mit Werkzeug- und Drehzahlauswahl sowie definierter Vertikalbewegung können erstellt, gespeichert und ausgeführt werden. Die erfassten Betriebsdaten eines Bohrvorgangs lassen sich mit Zeitstempel und weiteren Angaben in einer Datenbank erfassen.
4	Produktsystem	Die Ständerbohrmaschine ist zum systemfähigen Produkt geworden. Sie lässt sich nun in eine Fertigungslinie integrieren und von einer zentralen Anlagensteuerung hinsichtlich Werkzeugwechsel und Bohrvorgang fernsteuern. Dafür gibt es einen optionalen Feldbusadapter, der Modbus-, Profibus-, Profinet und andere industrielle Kommunikationsstandards unterstützt. Zusätzlich wird weiteres Systemzubehör, wie z. B. eine Werkstückzuführung, angeboten.
5	Systemverbund (System of Systems)	Zur Bohrmaschine gehören inzwischen verschiedene weitere programmierbare Systeme, wie eine Absaug- und Reinigungsvorrichtung, ein Schneidölzuführsystem zur Kühlung von Bohrloch und Werkzeug, ein Kamerasystem mit Werkstückerkennung per künstlicher Intelligenz sowie ein Cloud-basiertes CAD-System für das Bohrprozess-Engineering. Dazu existiert eine Materialdatenbank zur Optimierung von Bohrkopfdrehzahl, Bohrwerkzeugauswahl, Schneidölzufuhr usw. Zusätzlich gibt es ein Condition Monitoring für alle Baugruppen und Systeme einer Anlage, um eine präventive Wartung zu ermöglichen.

Tabelle 2: Übersicht zu den einzelnen Entwicklungsstufen technischer Produkte in den vergangenen Jahrzehnten am Beispiel einer elektrischen Ständerbohrmaschine. Für die Gesamtfunktion im Systemverbund ist bereits eine übergeordnete IoT-Plattform erforderlich. Obwohl die Stufe 5 den aktuellen Stand der Technik repräsentiert (den Produkt-Service-Hybrid), erfolgt die Plattformintegration in der Regel nach wie vor per IoT-Retrofitting.

Standard-Webdiensten der Amazon Cloud (siehe Tabelle 1). Mit Hilfe der Rules Engine lassen sich alle von den IoT-Geräten ankommenden Daten vorverarbeiten, speichern, analysieren und an andere Dienste weitergeben. Dafür werden Regelwerke mit einer SQL-ähnlichen Syntax aufgestellt. Trifft eine Regel zu, wird die dazu gehörende Aktion ausgeführt.

Software Development Kits

Als Entwicklerunterstützung stellt Amazon verschiedene Software Development Kits (SDKs) für unterschiedliche Programmiersprachen und Laufzeitumgebungen zur Verfügung. Das sind zum einen die sogenannten AWS IoT Device SDKs. Sie unterstützen die Anbindung unterschiedlicher Gerätearchitekturen an den IoT-Hub. Zusätzlich gibt es zum anderen eine Vielzahl von Anwendungs-SDKs, um in allen gängigen Sprachen eigene Applikationen für die AWS-Cloud zu entwickeln.

Systemverbund erfordert Plattformintegration

Das äußere Erscheinungsbild und die interne technische Realisierung nahezu aller technischer Produkte hat sich in den vergangenen Jahrzehnten deutlich verändert (siehe Tabelle 2 mit dem Beispiel einer fiktiven Ständerbohrmaschine). Dieser Trend wird sich in immer kürzeren Innovationszyklen fortsetzen und neben den technischen Eigenschaften auch die Geschäftsmodelle einbeziehen.

Für den gegenwärtigen Stand der Technik (Stufe 5 in der Tabelle 2, die Ständerbohrmaschine als Produkt-Service-Hybrid) wird auf jeden Fall eine Plattformintegration der jeweiligen Produkte benötigt, um den durch Industrie-4.0-Konzepte angestrebten Systemverbund zu realisieren. Einige IoT-Plattformfunktionen können dabei direkt vor Ort, also als Edge-Anwendung, ablaufen, andere erfordern einen externen Cloudservice.

Beispiel Condition-Monitoring

Eine Condition-Monitoring-Lösung, die beispielsweise in einem „System of Systems“ bei ungewöhnlichen Betriebsbedingungen über frühzeitige Warnungen auf den drohenden Ausfall einzelner Bauelemente hinweist, lässt sich mit Hilfe einer IoT-Plattform sowohl lokal an der Edge als auch in der Cloud realisieren. Über geeignete Sensoren (z. B. Strom, Vibration, Geräuschentwicklung, weil jede Maschine eine spezifische Strom-, Schwingungs- und Schallcharakteristik aufweist) werden fortlaufend Zustandsdaten erfasst und an die IoT-Plattform übermittelt. Dort wird per Machine-Learning-Algorithmen oder anderer statistischer Verfahren die jeweilige Bauteilbeanspruchung bestimmt und die verbleibende Restnutzungsdauer berechnet. Mit dem Ergebnis lassen sich die Instandhaltung optimieren und über Produktivitätsverbesserungen quantifizierbare Kosteneinsparungen erzielen. ◄

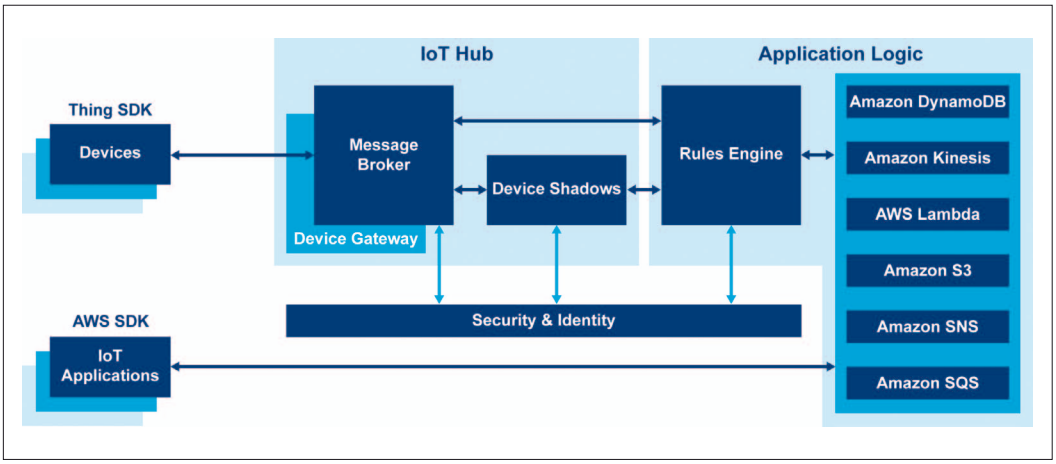


Bild 2: Neben der Applikationslogik besteht AWS IoT aus einem modularen IoT-Hub, der protokollseitig MQTT-, HTTP 1.1- und WebSockets-Verbindungen ermöglicht, wobei der Schwerpunkt eindeutig bei MQTT liegt. In Bezug auf die Sicherheit wird erwartet, dass die Geräte TLS 1.2 mit gegenseitiger Authentifizierung unterstützen und mit einem X.509-Zertifikat plus einem privaten Schlüssel ausgestattet sind.