

## Embedded System als integrierte (I)IoT-Schnittstelle



Embedded Systeme und Sensoren sind die grundlegenden Funktionsbausteine sowohl für das IoT als auch für Industrie 4.0. Zu jedem IoT-Device- und Industrie-4.0-Verwaltungsschale gehören ein eingebettetes System und mindestens ein Sensor bzw. Aktor. Hinzu kommt eine direkte oder indirekte Verbindung ins Internet, um einen Cloud-Service oder eine IoT-Serviceplattform mit Daten zu versorgen. Das eigentlich wertschöpfende Element ist allerdings die Cloud bzw. der in der Cloud genutzte Service.

Anwendungsarchitekturen wandeln sich von Zeit zu Zeit grundlegend. In der Vergangenheit wurden für zahlreiche Applikationen die Embedded Systems-Baugruppen hinsichtlich Prozessorleistung und Speichermöglichkeiten immer leistungsfähiger. Die jeweiligen Baugruppen sind als zentrale Instanzen für anspruchsvollere Datenverarbeitungsaufgaben und Bedienkonzepte im Einsatz. Bedingt durch das Internet of Things (IoT) und Industrie 4.0 und die damit einhergehende vertikale und horizontale Vernetzung verlagern sich nun ressourcenintensive Datenspeicher- und Analyseaufgaben vom Embedded System in die Cloud. Des Weiteren ersetzen Smartphones als externe Bedien-

einheiten hochauflösende LCDs und Touchscreens am Embedded System. Dadurch sinkt der Ressourcenbedarf vor Ort sehr deutlich. Gleichzeitig werden aber zusätzliche Wireless-Schnittstellen und Kommunikationssoftwarekomponenten benötigt.

Die meisten der bis 2020 im IoT vernetzten „Connected Things“ werden für Datensammel- und Monitoringaufgaben zum Einsatz kommen. Jedes „Thing“ enthält ein Embedded System mit speziellen Kommunikationsschnittstellen und mindestens einen Sensor, der den Things-Anbieter über eine Cloud mit Daten versorgt. Über diese Daten werden dem Nutzer dann spezielle Zusatzfunktionen angeboten. Das Embedded System benötigt eine direkte oder indirekte Cloud-Verbindung – z.B. über eine geeignete Middleware – und in der Wolke eine virtuelle Repräsentanz als Datenabbild. Über die Cloud existieren in einer solchen dezentralen Architektur geräteunabhängige Serviceschnittstellen, um unterschiedlichen (Smart Service-) Anwendungen den Zugriff auf die Daten einzelner Objekte zu ermöglichen. Um die vielfältigen Möglichkeiten und Kommunikationsanforderungen zukünftiger Embedded-Systems-Einsatzszenarien aufzuzeigen, folgen einige Beispiele aus der Automatisierungswelt.

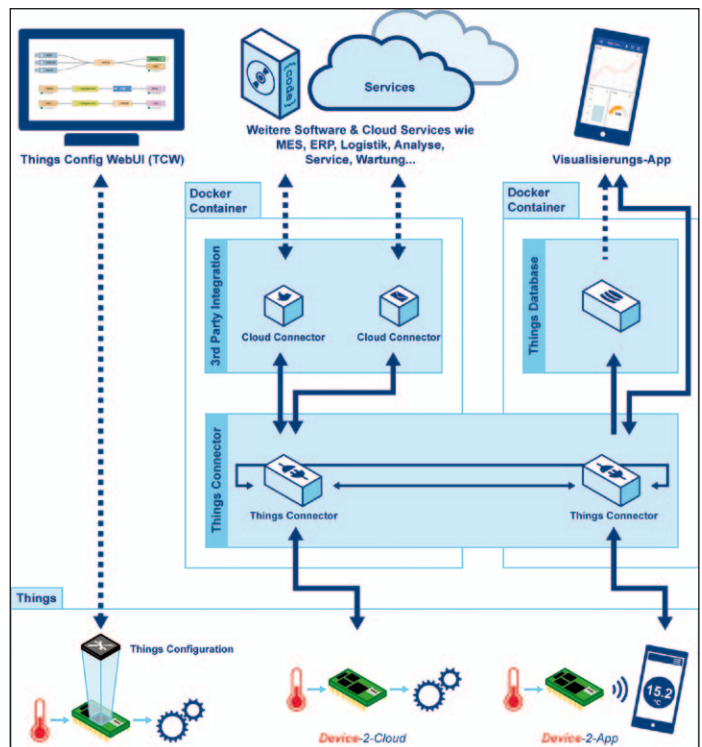
### Beispiel: Device-2-Cloud

Eine vernetzungsfähige Automatisierungsbaugruppe, zum Beispiel eine Steuerung, soll mit einer Cloud verbunden werden, um eine

vorausschauende Wartung (Predictive Maintenance) für eine komplexe Antriebsbaugruppe oder eine komplette Maschine zu ermöglichen. Trendvorhersagen durch Auswertungen größerer Datenmengen werden im IT-Umfeld schon seit Jahren unter dem Sammelbegriff „Predictive Analytics“ praktiziert. Aus diesem Grund gibt es in verschiedenen Cloud-Serviceplattformen hochentwickelte und praxiserprobte Dienste, die sich auch zur Vorhersage der Ausfallwahrscheinlichkeit einzelner Maschinenkomponenten und somit zum Festlegen geeigneter Wartungstermine eignen. Um solche Services zu nutzen, müssen vor Ort geeignete Daten erfasst und in die Cloud transportiert werden. Dabei sind nicht nur funktionale Aspekte, sondern auch der Datenschutz und die IT-Security zu beachten. Da die Steuerung in diesem Szenario lediglich eine einfache Industrial-Ethernet-Schnitt-

stelle hat, ist beispielsweise eine Middleware-Softwarekomponente (virtuelles Gateway) als Bindeglied zur Cloud erforderlich. Des Weiteren reichen die in der Steuerung vorhandenen Daten häufig für ein effektives Predictive Maintenance nicht aus. Es sind daher zusätzliche Sensoren (Wireless-Mikrocontroller plus Sensor als sogenannter „Smart Connected Sensor“) im Umfeld der Steuerung erforderlich, die ausschließlich als Datenquelle für die IT-gestützte vorausschauende Wartung dienen.

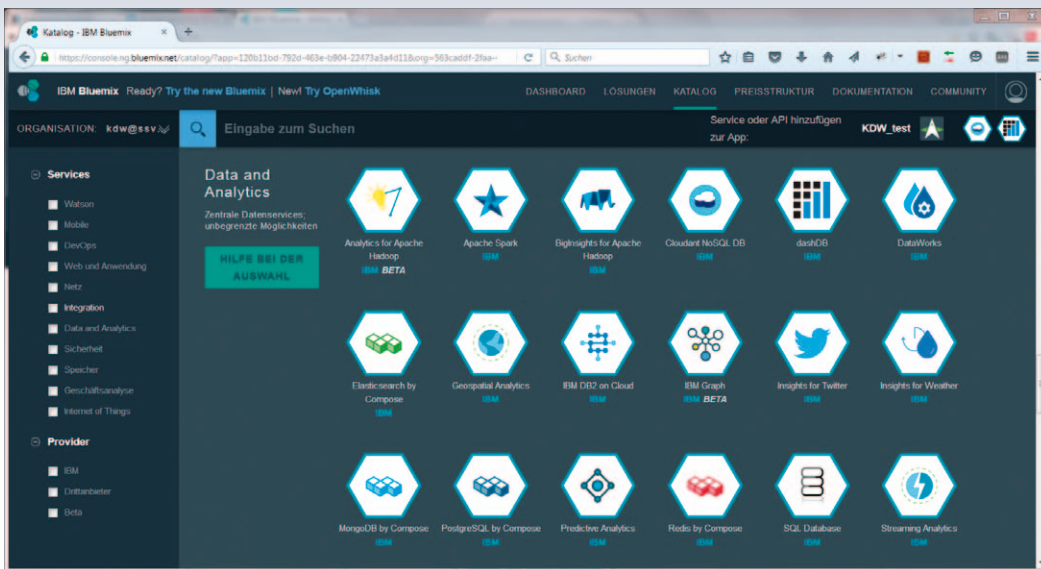
In der Praxis könnten die Steuerungsdaten zusammen mit weiteren Messdaten aus externen Smart Connected Sensors zunächst in einem virtuellen Behälter (z.B. Docker-Container als Middleware oder Private Cloud) zusammengefasst und in einer virtuellen Repräsentanz als Objekt- und Datenschnittstelle gespeichert (Bild 1) werden. Von dort aus werden selektive Datenein-



**Bild 1: Vielfach ist ein Embedded System nicht direkt, sondern über dazwischengeschaltete Softwarekomponenten mit einer Public-Cloud-Serviceplattform im Internet verbunden. Die Komponenten können z.B. aus einem IoT-Technologie-Stack wie „Thinglyfied“ stammen. In der Automatisierung lassen sich damit eine vorausschauende Wartung (Predictive Maintenance), Vor-Ort-App-Zugriffe auf Zustandsdaten oder eine Cloud-basierte Anlagenüberwachung mittels vorhandener IT-Monitoringwerkzeuge realisieren**

### Autor:

Klaus-Dieter Walter, CEO der SSV Software Systems GmbH



**Bild 2: Technologisch führende Cloud-Serviceplattformen für IoT- und Industrie 4.0-Anwendungen unterscheiden sich durch unterschiedliche Services. In der IBM Bluemix Cloud findet man beispielsweise ein sehr umfangreiches Angebot hinsichtlich Datenspeicherung und Analytics. Die Cloudant NoSQL-Datenbank eignet sich als Baugruppen- und Maschinendatenspeicher; Thinglyfied bietet hierfür einen entsprechenden Cloud Connector. Der Predictive Analytics-Service ist für Predictive Maintenance. Zusätzlich wird mit Hilfe eines speziellen Werkzeugs (SPSS Modeller) ein Vorhersagemodell erstellt**

heiten zur Auswertung an den Predictive-Service einer Public Cloud weitergegeben. Das Vorhersageergebnis stellt ein solcher Cloud-Service in der Regel per Dashboard oder entsprechender Webserviceschnittstellen für ERP-/MES-Anwendungen zur Verfügung. Bei Bedarf kann aus dem Dashboard heraus auch eine Alarmierung der Serviceverantwortlichen erfolgen. Zusätzlich kann eine beliebige Monitoringanwendung direkt auf die virtuelle Repräsentanz zugreifen, um einen Echtzeit-basierten Alarmierungs- und Benachrichtigungsservice zu schaffen.

### Beispiel: Device-2-App

In unzähligen Embedded-System-basierten Automatisierungsbaugruppen stecken inzwischen sehr wertvolle Zustandsdaten, auf die innerhalb der normalen Betriebsumgebung bisher nicht zugegriffen werden kann. So haben zum Beispiel pneumatische Subsysteme vielfach eine interne Intelligenz, die Condition Monitoring-Daten erzeugt und speichert. Aus diesen Daten lässt sich beispielsweise bestimmen, wie oft ein Ventil oder eine Hydraulikeinheit benutzt wurde und wann sie gewartet bzw. ausgetauscht werden sollte. Da sehr häufig als einzige Schnittstelle eine Verbindung zur übergeordneten Steuerung existiert, von der lediglich Aktorbefehle empfangen werden, bleiben solche wichtigen Daten ungenutzt, weil im normalen Betrieb keine Zugriffsmöglichkeit besteht.

In umgekehrter Richtung sieht es ähnlich aus. Intelligente Sensor- und Aktorbaugruppen bieten vielfältige Konfigurationseinstellungen und Diagnosemöglichkeiten. Für die Steuerung stehen in der Regel eine Programmierschnittstelle und die entsprechenden Engineering-Softwarewerkzeuge zur Verfügung - für periphere Funktionseinheiten hingegen nicht. Der Zugriff auf die Daten ist nur einem Servicetechniker mit entsprechender Spezialausrüstung möglich.

Der Zugriff auf die aktuellen Zustands- und Diagnosedaten wäre relativ einfach möglich, wenn die zum Einsatz kommenden Embedded Systeme zusätzlich eine Nahbereichsfunkschnittstelle (BLE = Bluetooth Low Energy bzw. Bluetooth Smart und NFC = Near Field Communication) besäßen. Da die meisten der im vergangenen Jahr weltweit verkauften 1,4 Milliarden Smartphones von Haus aus mit solchen Schnittstellen ausgestattet sind, sollten Embedded Systeme in Automatisierungsbaugruppen ebenfalls mindestens eine der beiden Funkschnittstellen haben. Mittels einer speziellen Smartphone-App wäre dann das direkte Auslesen selektiver Zustandsdaten und - falls erforderlich - auch ein schreibender Zugriff auf bestimmte Konfigurationsdaten möglich. Darüber hinaus ist die zum Einsatz kommende App in der Lage, die ausgelesenen Daten oder die neuen Konfigurationseinstellungen an eine Private oder Public Cloud weiterzulei-

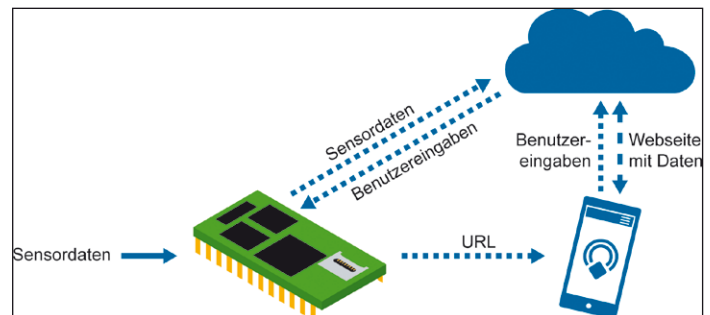
ten bzw. Historien- und andere Konfigurationsdaten aus der Cloud hinzuzuziehen und dem App-Benutzer darüber wertvolle Zusatzinformationen anzubieten.

### Embedded Physical Web

Eine Kombination aus Device-2-Cloud und Device-2-App bildet das „Physical Web“-Konzept aus dem Hause Google. Dieses recht neue Open-Source-Projekt wird dort von dem bekannten UX-Designer Scott Jenson geleitet. Die Google-Entwickler verfolgen das Ziel, Benutzerinteraktionen mit beliebigen Objekten im Internet of Things zu ermöglichen, ohne zuvor für jedes Objekt eine spezielle App installieren zu müssen. Die Grundidee dabei ist, jedem IoT-Objekt eine einfache aber

eindeutige URL zuzuweisen, über die eine dem Objekt zugeordnete Webseite aufgerufen werden kann. Anders als z.B. bei einem QR-Code wird dieser Weblink allerdings nicht ausgedruckt, auf das Objekt geklebt und mit einem Smartphone eingelezen. Im Physical Web versendet ein Objekt die URL als Broadcast-Beacon periodisch per Bluetooth an die Umgebung. Ein Smartphone kann alle Beacons im Umfeld des jeweiligen Standorts empfangen und die dazugehörigen Weblinks anzeigen. Durch einen Klick des Smartphone-Benutzers auf einen Link wird dann die betreffende Objekt-Webseite aus dem Internet geladen und mit einem Webbrowser geöffnet. Parallel zum URL-Versand kann das Embedded-Systems-basierte Objekt über eine weitere Schnittstelle mit einer IoT-Plattform in der Cloud beliebige Daten austauschen, die in die jeweilige Webseite eingebunden sind. Dadurch besteht für den Smartphone-Nutzer die Möglichkeit der direkten Interaktion mit dem in das Objekt eingebetteten System. Dabei kommen neben Bluetooth nur offene Internet-Standard-Bausteine (Webtechnologien, wie z.B. URL, HTTP, HTML, JavaScript, WebSockets, CSS) zum Einsatz. Durch die Interaktionsmöglichkeit kann ein Smartphone-Nutzer nicht nur statische Webdaten über das Objekt abrufen, sondern auch eine interaktive Transaktion ausführen, ohne zuvor eine spezielle Objekt-spezifische App installieren zu müssen.

■ SSV Software Systems GmbH  
www.ssv-embedded.de



**Bild 3: In einer Physical Web-Anwendung sendet ein Embedded System einmal je Sekunde einen Bluetooth-Beacon mit einer Webadresse (URL) aus. Ein solcher Beacon kann mit jedem Smartphone empfangen werden. Durch den Aufruf der jeweiligen Webseite im Internet wird die Möglichkeit geschaffen, aktuelle Daten eines IoT-Objekts zu betrachten und über Benutzereingaben auf das Objektverhalten einzuwirken. Da ausschließlich Standard-Webtechnologien zum Einsatz kommen, muss zuvor keine Objekt-spezifische App installiert werden**