

VOM GERÄT ZUM SMART DEVICE

FRANK BEHRMANN

Ohne je ein Handbuch oder Anleitungen gelesen zu haben, benutzt heutzutage jeder sein Smartphone, Tablet oder Smartwatch: einfach, intuitiv, angepasst an seine Bedürfnisse, angebunden an die Cloud, vernetzt mit anderen Geräten und Nutzern.

Warum haben wir nicht die gleichen Ansprüche an die Bedienung von Industriegeräten?

Ja, ich gebe es zu: Auch ich lese ungern Bedienungsanleitungen. Wenn doch, liegt es meist daran, dass das Gerät mit Touch-Display, Kamera, Computing-Power und jeder Menge Software zwar alle Voraussetzungen mitbringt, „smart“ zu sein, aber erst die Vernetzung im Internet es zu einem Smart Device macht.

Genauso haben kommerzielle und industrielle Geräte, gleich ob Werkzeugmaschine in der Fabrik, Pfandautomat im Supermarkt, Zugangssysteme in einem Gebäude, oder medizinisches Analysegerät in der Klinik neben ihren spezifischen Aufgaben das Potential sich zu vernetzen, also „smart“ zu werden. Was fehlt also noch?

Auch hier gilt: es ist nicht nur die Anwendung, die ein Gerät „smart“ macht, sondern die vernetzte Lösung im industriellen Internet der Dinge (IIoT) in Kombination mit einer sehr einfachen und intuitiven Bedienung.

Machen wir uns mit den verschiedenen Begrifflichkeiten, die im Umdel von IIoT kursieren vertraut: im Kontext einer vernetzten Lösung wird das dezentrale Computing-System, also zum Beispiel das Embedded System im Gerät oder eine intelligente Sensor-Box, als „Edge“ bezeichnet. Dem gegenüber wird das Computing-System virtualisiert als „Cloud“ umschrieben.

Während früher die Frage war: soll die Anwendung lokal mit begrenzter Computing-Power ausgeführt, verarbeitet und gespeichert werden, oder sende ich alle Rohdaten in das Datacenter und verarbeite sie dort mit nahezu unbegrenzter Rechenleistung, so lautet sie heute: wie realisiere ich

„Für mich stellt sich nicht mehr die Frage: Edge ODER Cloud. Meine Antwort lautet: Edge UND Cloud!“

eine Lösung durchgängig von der lokalen Anwendung bis zu den Cloud-Diensten? Welche Funktionen realisiere ich lokal, also mit Edge Computing und welche remote mit Cloud Computing, und wie spielen diese zusammen? Schon bei der Konzeption ist es essentiell, die Anwendung durchgängig von der lokalen Applikation bis zu den Cloud-Diensten zu planen und ganzheitlich mit der passenden Hard- und Software IoT-kompatibel zu entwickeln. Das ist, was ich „Edge-to-Cloud Engineering“ nenne.

Damit komme ich wieder zurück auf meine am Anfang gestellte Frage des „smar-

ten“ Geräts. Nehmen wir ein traditionelles Zugangssystem, das aus Sprechstelle, lokaler Gegenstelle und Türöffner besteht. Der Kunde möchte es nun „smartifizieren“: ein Video des Besuchers übertragen, Bewohner mit biometrischer Gesichtserkennung authentifizieren und von jedem Ort mobil bedienen. Das erfordert eine Lösung nach IoT-Paradigmen mit einem durchgängigen „Edge-to-Cloud“ Engineering.

Die Aufgabenstellung ist durchaus anspruchsvoll: welches ist die passende Computing Plattform, welche Schnittstellen, Protokolle und Sicherheitskonzepte sollen verwendet werden? Über welche IoT- und Cloud Plattform können die Daten zuverlässig und sicher ausgetauscht werden?

Im industriellen Umfeld steigt die Komplexität, weil durch die spezifischen Aufgaben die Geräte auch spezielle Anforderungen haben, seien es die Umgebungsbedingungen im Feld, die Verfügbarkeit von 24/7, oder die Sicherheit der Funktion und Daten auf allen Ebenen.

Neben der reinen Funktion gilt es, dem Anwender die smarte Nutzererfahrung zu ermöglichen, die er im privaten Umfeld gewohnt ist, und eine intuitive Bedienung mit einem modernen Design in Einklang zu bringen - damit ein Anwender genau wie ich einfach loslegen kann, ohne das Handbuch bis zur letzten Seite intensiv studieren zu müssen □

FRANK BEHRMANN, HY-LINE COMPUTER COMPONENTS, ist als Manager Software, Digitalisierung und IoT bei dem Unternehmen verantwortlich für das Lösungsangebot in diesen Innovationsbereichen. Schon bei Nokia Siemens Networks, Telefonica und Huawei beriet er Industriekunden bei der Umsetzung von Lösungen im IoT-Umfeld.