



► In der industriellen Kommunikation erlauben Container die identische Ausführung von Softwaremodulen in unterschiedlichen Umgebungen.

## Container-Anwendungen erklärt

# Von den Grundlagen bis zum Einsatz

**Container-Anwendungen lassen sich unabhängig von externen Komponenten, der Umgebung und dem eingesetzten Betriebssystem ausführen. Diese Flexibilität bietet große Freiheiten und erlaubt ihren Einsatz vor Ort genauso wie in der Cloud. Nicht zuletzt deshalb erobern sie sich jetzt auch in der industriellen Fertigung ihren Platz.**

Container sind gekapselte, virtuelle Server und stellen eine Umgebung für die Ausführung (schlanker) Softwaremodule mit einem eng abgegrenzten Funktionsbereich bereit. In diesen Container-Anwendungen sind alle benötigten Komponenten als Bibliothek enthalten. So ist sichergestellt, dass die Ausführung unabhängig von externen Komponenten und der Umgebung möglich ist und sich die Container-Anwendung, unabhängig von ihrer jeweiligen Ausführungsumgebung, immer identisch verhält. Kommen mehrere Container-Anwendungen auf einer Hardwareplattform zum Einsatz, laufen diese als unabhängige Instanzen und sind voneinander und vom Gesamtsystem getrennt. Sie teilen sich aber das Betriebssystem und gegebenenfalls bestimmte Hardwareressourcen, sie können auch aufeinander aufbauen oder Daten untereinander austauschen. Die 2013 als Open-Source-Projekt gestartete Docker-Lauf-

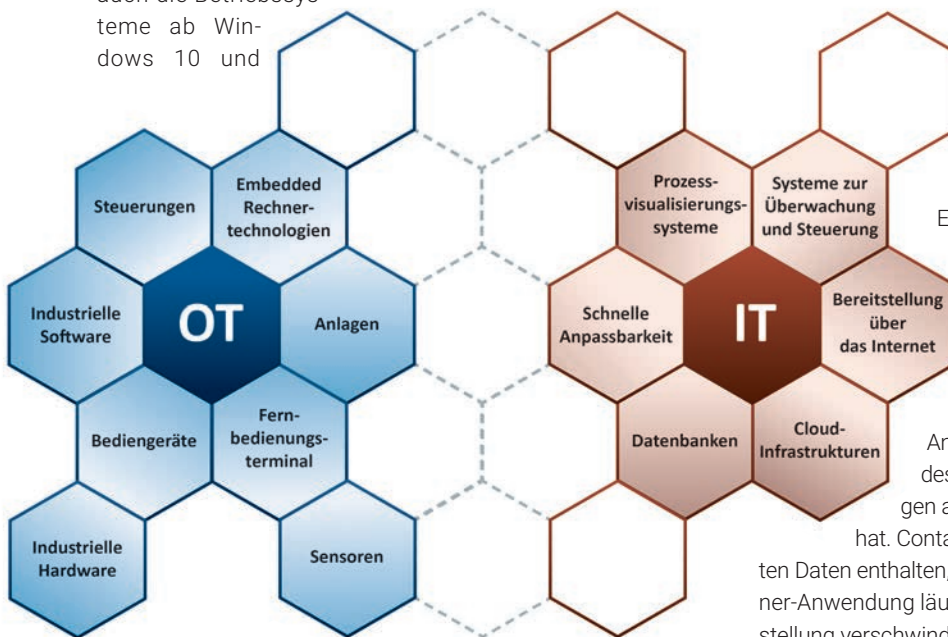
zeitumgebung (Docker Engine) hat sich in der Zwischenzeit zum De-facto-Standard für Container-Umgebungen entwickelt.

### Unterschied zu virtuellen Maschinen

Wie virtuelle Maschinen greift auch die Container-Technik über eine Abstraktionsschicht auf die Nachbildung eines Hardware- oder Softwareobjekts zu. Der Unterschied ist allerdings, dass für die Ausführung von Containern die Virtualisierung eine Ebene höher aufsetzt und statt der physikalischen Hardware das Betriebssystem emuliert wird, das dann auch die Container-Laufzeitumgebung bereitstellt. So wird es möglich, dass sich mehrere Container-Anwendungen ein Betriebssystem mit den vorhandenen Ressourcen ohne feste Reservierung teilen. Da außerdem in jeder Container-Anwendung nur die benötigten Betriebssystemkomponenten und Bibliotheken enthalten sein müssen, gilt die Container-Technik als besonders leichtgewichtig und ressourcenschonend.

### Lokal oder in der Cloud lauffähig

Ursprünglich wurde die Container-Technik für Linux-Betriebssysteme entwickelt. Seit der Einführung von Hyper-V können auch die Betriebssysteme ab Windows 10 und



► Industrie 4.0 erfordert eine umfassende Integration von Produktions- und IT-Welt.

Windows Server 2016 für Virtualisierungsaufgaben und damit für Container-Anwendungen genutzt werden. Daneben unterstützen auch Cloud-Plattformen wie z.B. Amazon Web Services (AWS), Microsoft Azure oder Google Cloud Computing den Einsatz von Containern, sodass heute eine Container-Laufzeitumgebung auf Hosts oder in Serverumgebungen, lokal oder in der Cloud, aufsetzend auf den Betriebssystemen Windows, Linux oder Mac OS genutzt werden kann. Aufgrund ihrer geringen Größe können Container-Anwendungen je nach Bedarf auch problemlos aus lokalen Architekturen in Bare-

Metal-Systeme sowie in Public-, Private-, Hybrid- und Multi-Cloud-Umgebungen verschoben werden.

### Vor- und Nachteile von Container-Anwendungen

Ein großer Vorteil von Container-Anwendungen ist die Bereitstellung innerhalb weniger Sekunden: Nach dem Pull (Herunterladen) ist nur noch der Start über die Ausführung eines einzelnen Befehls notwendig. Auf diesem Weg lassen sich auch neue Versions- und Konfigurationsstände in Sekundenschnelle übernehmen. Ändert sich die Auslastung, kann die Arbeit schnell auf mehrere Container-Anwendungen verteilt werden, sofern diese Funktion unterstützt wird. Weitere Vorteile sind die Verwendung identischer Hardwareressourcen bei einem parallelen Einsatz und die Unterstützung verschiedener Betriebssystemumgebungen sowie hybrider Architekturen, sodass die Verwendung auch in einer Cloud-Umgebung möglich ist.

Für den produktiven Einsatz bieten Container-Anwendungen darüber hinaus den Vorteil, dass sie leicht von einer zentralen Stelle aus verwaltet werden können. Das kann entweder vor Ort über ein Device Management System (DMS), über einen Container-Orchestrator oder über die Cloud (z.B. Microsoft Azure IoT Edge, AWS IoT Greengrass) erfolgen. Softing bietet hier Komplettlösungen bestehend aus DMS, IPCs und eigenen Container-Anwendungen zur Datenerfassung an.

Ein Nachteil von Container-Anwendungen ist, dass außerhalb des Containers auf die Einhaltung von Sicherheitsstandards geachtet werden muss. Bei einem parallelen Einsatz mehrerer Container-Anwendungen auf einem Rechner ist es außerdem möglich, dass ein Angriff auf eine Container-Anwendung über die gemeinsame Verwendung des Betriebssystemkerns unmittelbar Auswirkungen auch auf die anderen Container-Anwendungen hat. Container sollten nach Möglichkeit keine persistenten Daten enthalten, da sie nur so lange existieren, wie die Container-Anwendung läuft. Beim Beenden oder einer erneuten Bereitstellung verschwinden alle Daten, die erzeugt wurden. Um das zu verhindern, kann für die Speicherung ein Laufwerk außerhalb der Container-Umgebung verwendet werden. Anwender müssen darüber hinaus spezielles Knowhow über Container-Anwendungen und die dafür notwendige Infrastruktur aufbauen. Die Produktfamilie DataFeed EdgeConnector von Softing nutzt Container für eine leistungsfähige Einbindung von Steuerungen in Industrie-4.0-Anwendungen. Erfahren Sie mehr dazu im zweiten Teil dieses Artikels in der September-Ausgabe des SPS-MAGAZINs. ■

Direkt zur Übersicht auf **i-need.de**  
www.i-need.de/f/9751



Sebastian Schenk,  
Produktmanager,  
Softing Industrial Automation GmbH  
industrial.softing.com