

Neue Freiheiten durch Docker-Container

Funktional haben die Softwarekomponenten in der OT-Ebene einer Automatisierungslandschaft eine sehr viel größere Bedeutung als die Hardware. Trotz allem sind die zum Einsatz kommenden Rechnerbaugruppen nach wie vor keine offenen Plattformen, auf denen sich jede gewünschte Software einsetzen lässt. Docker-Apps sollen das jetzt ändern.

Von Zeit zu Zeit übernimmt die Automatisierungstechnik praxiserprobte Technologiebausteine aus der IT, um sie in die eigenen Produkte und Lösungen zu integrieren. Beispiele dafür gibt es viele: PC-Architekturen, COM/DCOM (OPC), Ethernet, TCP/IP, HTTP, Linux, Bluetooth, Wi-Fi, XML, HTML, SQL und IoT-Architekturen gehören auf jeden Fall dazu.

Verschiedene Bausteine haben sogar einen technischen Paradigmenwechsel in der Automatisierung verursacht: Ethernet hat z. B. die klassischen Feldbusse verdrängt und Linux eine allgemeine Open-Source-Akzeptanz geschaffen. Das Internet der Dinge mit MQTT, JSON, virtuellen (Daten-) Repräsentanzen und allem, was sonst noch dazugehört, hat zu Industrie-4.0-Anwendungen inklusive des digitalen Zwillinges und ähnlichen Dingen geführt.

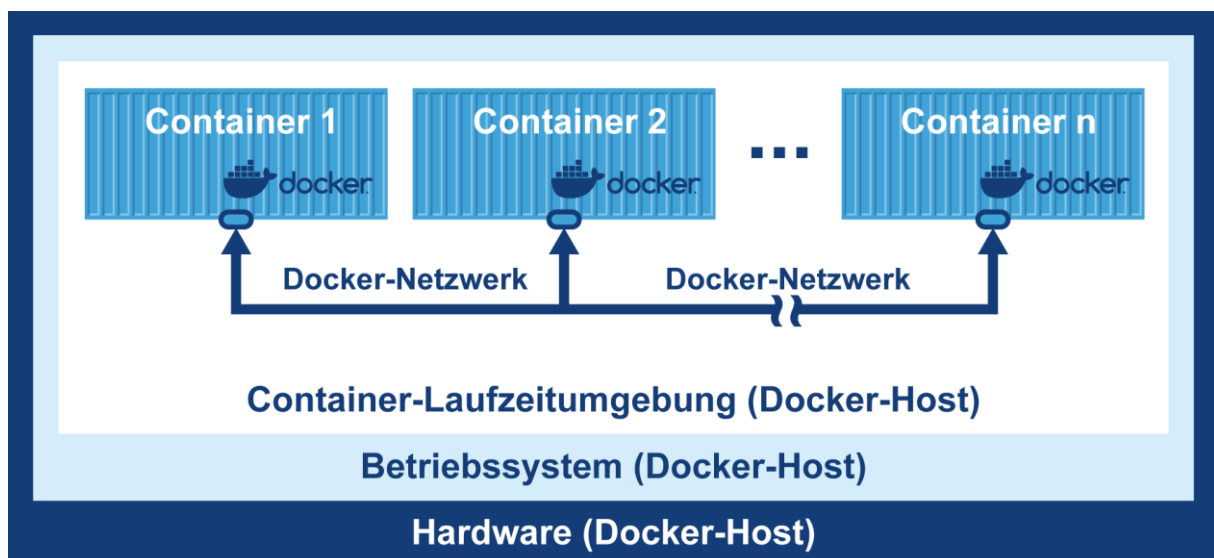


Abbildung 1: Durch eine Container-Engine mit integrierter Container-Laufzeitumgebung wird eine Baugruppe zum Docker-Host, auf dem sich Anwendungen als einzelne Docker-Container ausführen lassen. Im Gegensatz zu anderen Virtualisierungstechniken beinhaltet ein Container kein eigenes Betriebssystem. Stattdessen wird das jeweilige Host-Betriebssystem genutzt. Daher sind die einzelnen Docker-Container relativ kompakt.

Seit einiger Zeit befassen sich immer mehr Automatisierungsanbieter mit der Docker-Container-Technologie. Unter dem Sammelbegriff „Docker“ verbirgt sich ein hochentwickeltes und weit verbreitetes Virtualisierungskonzept, das in der Linux-Welt entwickelt und Anfang 2013 unter einer Apache-2.0-Open-Source-Lizenz veröffentlicht wurde.

Mit einem Docker-Container lassen sich Softwareanwendungen nahezu plattformunabhängig weitergeben, um sie auf Rechnern mit 32- bzw. 64-Bit-AMD/Intel-, Arm- oder RISC-V-Prozessoren sowie Linux-, Microsoft-Windows-, macOS- und anderen UNIX-ähnlichen Betriebssystemen zu betreiben. Die Bereitstellung und Installation einer Docker-basierten Anwendung erfolgt nahezu unabhängig vom Zielsystem über Registry-Server und Repositories. Die zu installierende Applikation wird dabei als Docker-Container-Image wahlweise auf einem Internet-Server (z. B. Docker Hub) oder einer lokalen Serveranwendung zur Verfügung gestellt.

Autark und vernetzbar

Eine Anwendung in einem Docker-Container, der auf einem Docker-Host ausgeführt wird, stellt keine besonderen Anforderungen an spezielle Betriebssystemversionen oder im Host-Dateisystem vorinstallierte Laufzeitumgebungen (z. B. eine spezielle Version einer Programmiersprache) oder andere Bibliotheken (beispielsweise ein OpenSSL mit der TLS-Version 1.3). Alles, was von der Containeranwendung benötigt wird, ist auch im Container selbst mit dem jeweils erforderlichen Versionsstand vorhanden. Da Container-Images sehr kompakt sind, sollte jede einzelne Anwendung ein separates Docker-Container-Image bilden. Die Container auf einem Host können über die Docker-Laufzeitumgebung ein Netzwerk mit IP-Adressen usw. erzeugen, um miteinander zu kommunizieren (siehe Abbildung 1). Für die IP-Kommunikation zwischen den Docker-Containern lassen sich die gleichen Funktionen und Protokolle nutzen, die auch bei der Rechnernetzung in einem LAN zum Einsatz kommen. Dadurch können auch Container auf unterschiedlichen Host-Systemen ohne weitere Änderungen miteinander kommunizieren. Darüber hinaus unterstützen Docker den Microservice-Gedanken: eine komplexe Anwendungssoftware lässt sich aus einzelnen Prozessen zusammensetzen, die in separaten Containern laufen und über Standardnetzwerkschnittstellen miteinander kommunizieren.

Docker als Funktionserweiterung

Im Moment sehen Automatisierungstechnikanbieter den Docker primär als Funktionserweiterung für ihre eigenen Produkte an. Fehlen zum Beispiel einem Edge-Gateway oder einer Steuerung bestimmte Funktionen, um eine Problemlösung vollständig zu realisieren, kann sie der Anwender als Docker-Container einfach nachinstallieren. Dafür ist kein Spezialwissen erforderlich, da in der IT-Welt hierfür bereits sehr komfortable Werkzeuge existieren, die sich per Webbrowser bedienen lassen (z. B. Portainer). Damit wird die Installation einer Docker-basierten Funktionserweiterung auf einer Automatisierungsbaugruppe genauso einfach wie die Installation einer App auf einem Smartphone.

Der Hersteller einer Automatisierungs-Hardware muss für das jeweilige Produkt allerdings die entsprechenden Voraussetzungen schaffen, indem eine zu Betriebssystem (Linux oder Windows) und Prozessorarchitektur (AMD/Intel, Arm) passende Container-Engine mit integrierter Container-Laufzeitumgebung (Container-Runtime) bereits ab Werk vorinstalliert ist oder die entsprechende Unterstützung für die nachträgliche Installation im Feld angeboten wird. Die betreffende Baugruppe wird durch diese Erweiterung zum Docker-Host. Auf einer Docker-Host-Hardware kann der Anwender die benötigten Funktionen bei Bedarf als „Automatisierungs-Apps“ selbst hinzufügen, zur Ausführung bringen und auch jederzeit wieder entfernen.

Da die gesamte Docker-Technologie aber gleichzeitig auch einen De-facto-Standard bildet, der in der IT seit Jahren millionenfach zur Auslieferung, Pflege und Weiterentwicklung von Anwendungssoftware genutzt wird, bilden Docker-fähige Automatisierungsbaugruppen auch ein interessantes neues Betätigungsfeld für Softwareentwickler. Eine Smart-Factory-Anwendungssoftware, die direkt in der OT-Ebene zum Einsatz kommen soll und per Ethernet-LAN mit der Umgebung kommuniziert, lässt sich per Docker-Container vollständig hardwareunabhängig entwickeln und auf den Geräten verschiedener Hersteller mit entsprechender Container-Engine einsetzen. Die Docker-Technologie könnte in der Automatisierungstechnik daher zu einem weiteren Paradigmenwechsel führen, da hier das Potenzial für ein neues Softwaremarktsegment beachtlicher Größe schlummert, in das weltweit tausende Anbieter einsteigen könnten.

Digitale Softwaremarktplätze

Motiviert durch den Erfolg der Smartphone-App-Stores sind inzwischen auch verschiedene Anbieter industrieller Baugruppen und Systeme dabei, ähnliche App-Store-Konzepte für ihre Produkte zu

realisieren. Die meisten bereits existierenden Plattformen sind allerdings nur zur Unterstützung der eigenen Produkte gedacht – also als Anbieter-Ökosystem. Ein großer Teil der angebotenen Apps kommt vom App-Store-Betreiber selbst. Einige wenige wurden von ausgesuchten Partnern entwickelt.

Herstellerunabhängige App-Stores für Automatisierungsanwendungen, die Docker-Technologie nutzen, befinden sich offensichtlich noch in der Entwicklung. Pünktlich zur diesjährigen Hannover Messe wurde der „Open Industry 4.0 Alliance Community App Store (OI4 Community App Store)“ von der gleichnamigen I4.0-Allianz vorgestellt. Diese Allianz will allerdings eine Open-Edge-Computing-Referenzarchitektur etablieren und ein Community-Prüfsiegel für industrielle Docker-Apps vergeben.

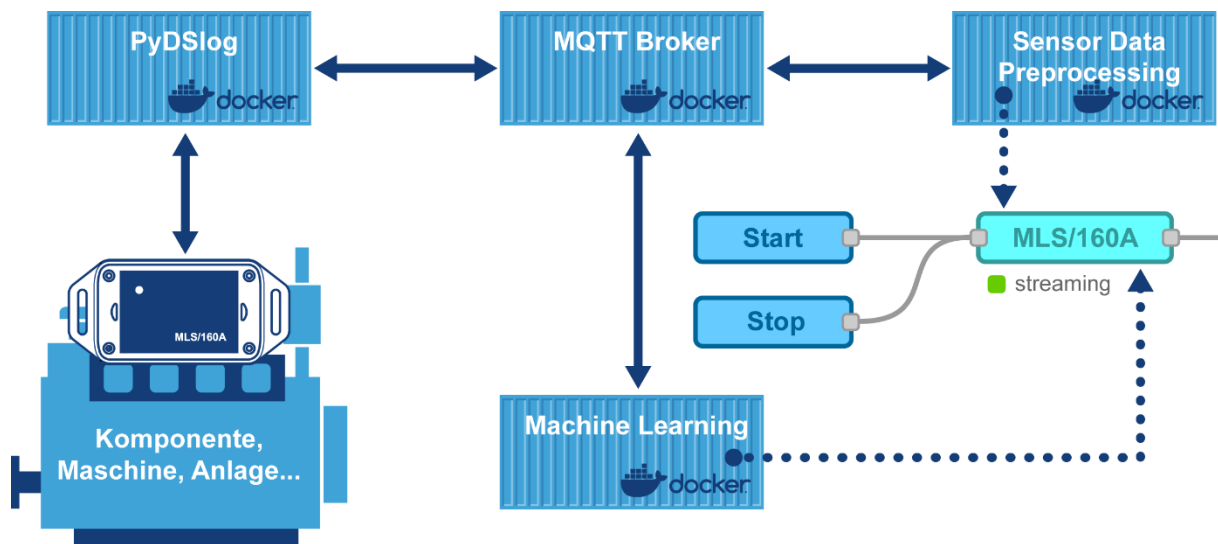


Abbildung 2: Bei der Vorstellung des „Industrial APP Marketplace“ zeigten die Unternehmen Knowtion, PerFact, Phoenix Contact und SSV Software Systems ein Docker-basiertes Condition-Monitoring-Beispiel. Die Zustandsbestimmung von Produktions- und Werkzeugmaschinen erfolgt über drei Docker-Container (PyDSlog, Sensor Data Preprocessing, Machine Learning), die über einen vierten Docker (MQTT Broker) miteinander kommunizieren. Zur Konfiguration und Fernsteuerung des Maschinensensors wurde ein Node-RED-Erweiterungsbaustein eingesetzt.

Ein weiteres Beispiel ist der „Industrial APP Marketplace“ der OWL-Maschinenbau-Initiative. Bei der Vorstellung dieses industriellen Docker-App-Store Ende April zeigten die Unternehmen Knowtion, PerFact, Phoenix Contact und SSV Software Systems bereits ein neuartiges Condition-Monitoring-Beispiel für Produktions- und Werkzeugmaschinen als Docker-Verbundsystem (siehe Abbildung 2). Die gesamte Applikation besteht aus insgesamt vier Containern, von denen jeder jeweils eine separate Docker-App bildet. Die Kommunikation der einzelnen Container untereinander erfolgt über einen MQTT-Broker, der ebenfalls als Docker-App implementiert wird.

Trotz der zahlreichen Vorteile für Anbieter und Anwender sollte man allerdings auch die Risiken solcher Docker-Apps nicht unbeachtet lassen. Durch die Verbreitung von Anwendungsprogrammen über App-Stores können Cyberangreifer auch gezielt Schadsoftware verteilen, um Edge-Gateways und Steuerungen missbräuchlich zu nutzen. Malware-Funktionen lassen sich in einem Docker relativ gut tarnen. Es gibt hier schon zahlreiche negative Beispiele aus der IT-Welt.

Innovationsgewinner

Hardware-herstellerunabhängige Docker-Apps, die sich ein Anwender auf einer Automatisierungsbaugruppe jederzeit selbst installieren und sogar nach dem Tausch einer Baugruppe weiter benutzen kann, hören sich im ersten Moment nach einer sinnvollen Innovation an. Am Anfang gewinnen dadurch

zunächst einmal alle Beteiligten: Die Hardware-Hersteller könnten ihre Kunden auf das riesige Angebot hochentwickelter Anwendungssoftware hinweisen, zu dem auch professioneller Support, Updates und Weiterentwicklungen als DevOps angeboten werden. Softwareanbieter können eine Anwendung einmal entwickeln und als Docker-Container an viele Anwender mit völlig unterschiedlicher Automatisierungs-Hardware liefern. Der Anwender hätte endlich eine freie Auswahl an hochwertiger Software und einen echten Wettbewerb, der zu besseren Produkten und Lösungen führt; siehe die Beispiele in der IT-Welt. Mittelfristig werden Automatisierungsbaugruppen durch die Docker-Technologie allerdings den Vendor-Lock-in verlieren, weil die Hardware ohne Softwareanpassungen austauschbar wird. Das wird zu Veränderungen in der Anbieterlandschaft führen. Eine solche Phase hat es in der IT vor vielen Jahren auch einmal gegeben. Nach diesem Paradigmenwechsel war der gesamte Markt sehr viel größer, stabiler und die Produkte deutlich innovativer.