

Container-Technologie für die Industrie 4.0



Überblick

Industrielle Umgebungen geraten zunehmend in das Visier von Cyberkriminellen. Durch die Verbesserung der digitalen Kommunikationsinfrastruktur lassen sich IT- und Cybersecurity-Probleme entschärfen. Dieses Dokument vermittelt ein grundlegendes Verständnis der Container-Technologie und zeigt ihre Leistungsfähigkeit im Kontext von Industrie 4.0, angewandt auf industrielle sichere Gateways.

Heterogene Softwareanwendungen in einer sicheren Umgebung betreiben

Viele Branchen stehen vor der Herausforderung, sich für die Industrie 4.0 fit machen zu müssen. Das ist meistens ein komplexes Projekt und erfordert oft einen enormen Aufwand an Kosten und Arbeitszeit. Ein zentraler Punkt, den die moderne Industrie dabei berücksichtigen muss, ist die sichere digitale Kommunikation: Sie ist die Basis für die Verwaltung von Zugriffsregelungen, die Fernüberwachung von Maschinen und viele andere Funktionen. Jede Branche hat ihre individuelle Ausgangssituation sowie ihre eigenen Anforderungen. Standard-Softwareanwendungen erfüllen diese oft nur unzureichend, und der Einsatz einer umfassenden und kundenspezifischen Lösung könnte so teuer sein, dass es letztlich nicht mehr praktikabel ist. Für diese Situation bietet die Container-Technologie eine einfache und erschwingliche Hilfe, und darüber werden wir schreiben.

Hintergrund

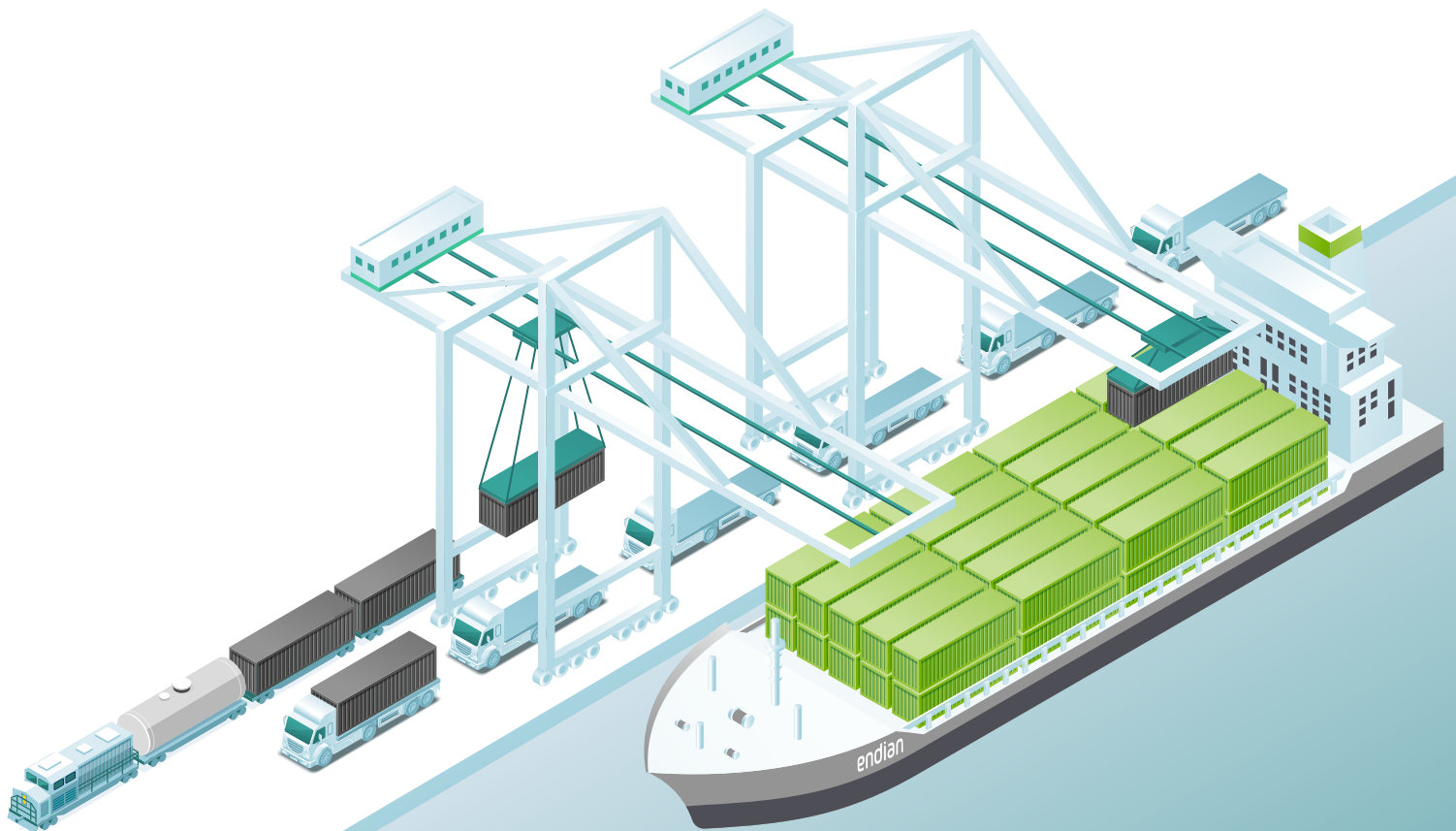
In diesem Abschnitt erklären wir das notwendige Grundwissen für das Verständnis des Themas Container.

Container-Technologie

Beginnen wir mit einem realen Beispiel aus der Logistik. Viele Unternehmen haben die gleiche Herausforderung: Wie können sie ihre eigenen Waren um die Welt transportieren und dabei die Kosten niedrig halten? Die Idee, dieses Thema zu lösen, ist die Verwendung von Versandcontainern. Aber was ist ein Versandcontainer? Es ist eine Metallbox mit fester Standardgröße, die mit jeder Art von Waren gefüllt werden kann. Das Konzept dahinter ist, dass ein Unternehmen für die Box als Einheit eine feste Gebühr bezahlt, unabhängig vom Inhalt; ein mit Gold gefüllter Versandcontainer könnte die gleiche Gebühr haben wie ein anderer, der halb mit Müll gefüllt ist. Es gibt natürlich einige Einschränkungen zu beachten, aber der wichtige Punkt, den wir zeigen wollen, ist, dass der Versandbehälter inhaltsunabhängig ist und als Einheit betrachtet wird.

Die Logistikbranche hat sich für Container entschieden, denn sie entsprechen den Anforderungen von Standard LKW-Anhängern und speziellen Frachtschiffen, die für diese Größen ausgelegt sind. Mit Hilfe von Kränen am Dock lassen sich die Container als Einheiten leicht von einem LKW-Anhänger in einen Laderaum und umgekehrt bewegen. Diese Technik senkt die Kosten und macht den Versand von Waren im Container effizienter.

Nachdem wir nun ein Verständnis für Frachtcontainer haben, wollen wir dieses Konzept auf die Informatik übertragen.



Software-Container

Eine Software-Anwendung besteht aus dem Kern der Anwendung, ihren Abhängigkeiten, Einstellungen und ihrer Konfiguration. Ein Software-Container, den wir von nun an kurz „Container“ nennen, ist eine Einheit, die nicht nur die Anwendung selbst enthält, sondern auch ihre Bibliotheken und alles, was sie zum ordnungsgemäßen Funktionieren benötigt. Auf die einzelnen Teile werden wir später noch eingehen. Einen Punkt möchten wir bereits an dieser Stelle unterstreichen: So wie ein Kran einen Frachtcontainer aufnehmen und als einzelne Einheit an einem anderen Ort platzieren kann, so lässt sich auch ein Container von einer Maschine zu einer anderen bewegen oder kopieren.

Betrachten wir folgendes Beispiel: Ziel ist, die gleiche Anwendung mit den gleichen Einstellungen auf einer Reihe von Rechnern zu installieren. Der traditionelle Weg in diesem Fall wäre, die Software auf jeder einzelnen Maschine zu installieren und zu konfigurieren. Dies ist eine sehr zeitaufwändige, sich wiederholende Arbeit, die aufgrund des hohen Fehlerrisikos zu Störungen und Ausfällen führen kann. Mit Containern lässt sich diese Aufgabe auf intelligentere Weise erledigen; dafür wird die Anwendung in einem einzigen Container installiert und konfiguriert, um sie im Anschluss ohne zusätzlichen Aufwand auf jede Maschine zu verteilen. Diese Vorgehensweise lässt sich vollständig automatisieren, was nicht zu unterschätzen ist. Das spart nicht nur Zeit, sondern verringert auch den Arbeitsaufwand und das Fehlerrisiko. In diesem Fall spielt es dank der Parallelisierung keine Rolle, ob die Anwendung auf fünf oder auf fünftausend Maschinen installiert wird.

Bevor wir auf weitere technische Aspekte eingehen, soll das folgende Beispiel als Metapher helfen, einige Konzepte zu erklären:

- Ein **Kunde (Top)** möchte sein Wissen über Wildtiere verbessern und sucht ein Buch zu diesem Thema.
- Der **Bibliothekar (Middle)** kennt den Bibliotheksprozess sehr gut. Er führt folgende Schritte durch:
 - Nimmt die Anfrage des Kunden entgegen
 - Leitet die Anfrage an die Bibliotheksinfrastruktur weiter
 - Bekommt das richtige Buch von der Bibliotheksinfrastruktur
 - Gibt das Buch an den richtigen Kunden aus

Konzentrieren wir uns auf diesen Punkt; der Bibliothekar ist die Verbindung zwischen Kunde und Bibliotheksinfrastruktur. Der Kunde darf nicht selbst auf die Bibliotheksinfrastruktur zugreifen und sich ein Buch nehmen.

- Die **Bibliotheksinfrastruktur (Base)** ist der Motor, der Speicher und die Verwaltung der Bibliothek.

In ähnlicher Weise lässt sich nun die Software betrachten, die in folgende Schichten unterteilt ist:

- Die **Software-Anwendung (Top)** muss ihre eigenen Interessen verfolgen, um die Arbeit richtig zu erledigen; sie muss auf die vom Betriebssystem (OS) bereitgestellten Ressourcen zugreifen, so wie der Kunde einer Bibliothek auf ein Buch zugreifen muss. Ein Beispiel für eine Software-Anwendung könnte der bevorzugte Internet-Browser sein.
- Die **Bibliothek (Middle)** ist eine versteckte Softwareschicht in der Mitte. Sie ist vergleichbar mit dem Bibliothekar in der Bibliothek. Sie führt die folgenden Schritte aus:
 - Nimmt die Anfrage der Anwendung entgegen und leitet sie an das Betriebssystem weiter
 - Erhält die angeforderte Ressourcenzugriffsberechtigung zurück
 - Gibt die Berechtigung an die anfordernde Anwendung aus

So wie der Kunde in der Bibliothek keinen direkten Zugriff auf die Infrastruktur der Bibliothek hat, hat auch die Anwendung keinen direkten Zugriff auf das Betriebssystem.

- Das **Betriebssystem (Base)** ist die Engine und der Ressourcenmanager des Softwaresystems. Einige Beispiele für Betriebssysteme sind Windows, Mac OS, Linux, Android, iOS.

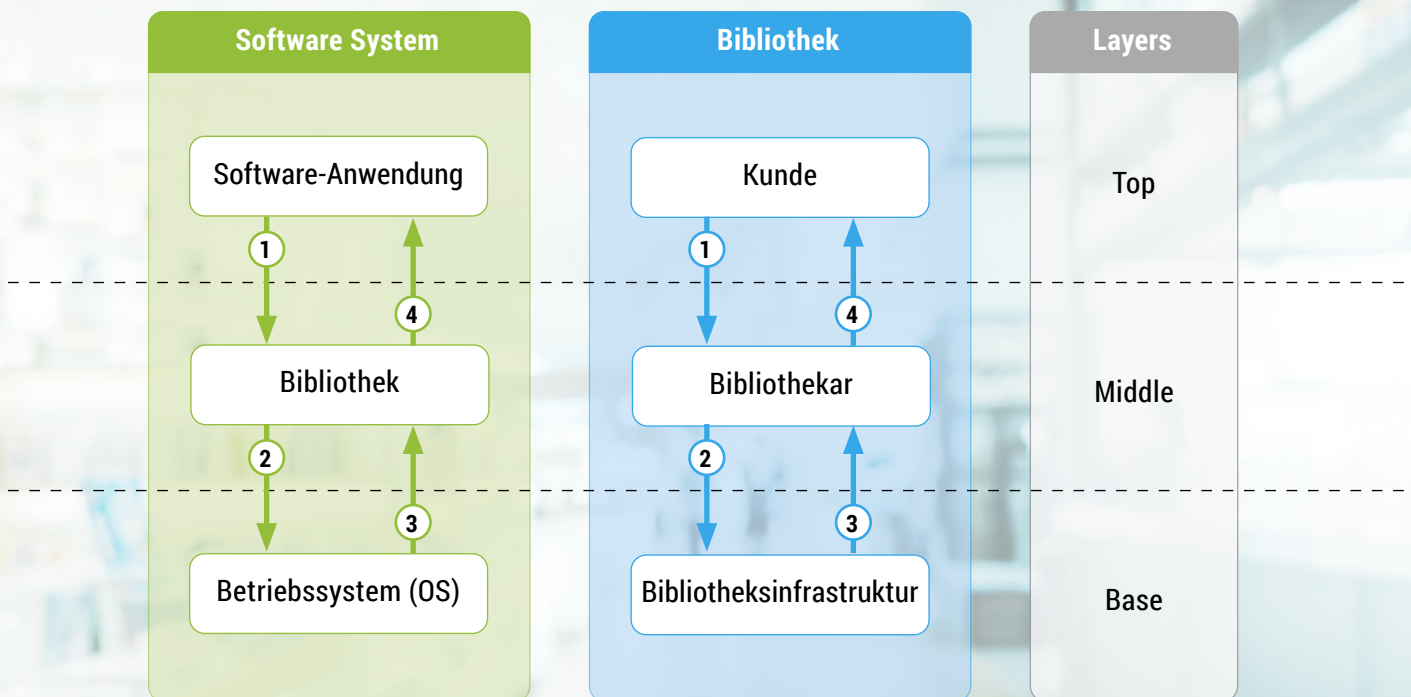


Abbildung 02

Abbildung 02 gibt ein visuelles Verständnis von dem, was oben bereits beschrieben wurde. Die Softwareschichten zeigen auf sehr vereinfachte Weise, wie traditionell eine Anwendung auf einem Standardcomputer läuft, wie beispielsweise auf einem PC.

Der nächste Schritt besteht darin, die Betriebssystemschicht für den Moment beiseite zu legen und die Anwendungs- und Bibliotheksschichten in eine einzige Einheit zu packen; dies ist unser Container.

Lassen Sie uns diesen wichtigen Punkt hervorheben; ein Container besteht aus einer Anwendung und den Bibliotheken. Da wir die OS-Schicht entfernt haben, ist der Container also OS-unabhängig.

Container Management Engine (CME)

Da wir nun wissen, wie ein Container aufgebaut ist, wollen wir eine neue Schicht zwischen dem Betriebssystem und den Containern einfügen, so wie in Abbildung 03 dargestellt. Bei dieser Schicht handelt es sich um die Container Management Engine (CME), die in der Abbildung durch Docker repräsentiert wird. Zum Zeitpunkt der Erstellung dieses Dokuments ist Docker die populärste CME. Die CME ist der wichtigste Bestandteil der Container-Technologie. Sie ist für fast jedes Betriebssystem verfügbar und ist dafür verantwortlich, die richtigen benötigten Ressourcen zur richtigen Zeit für jeden Container bereitzustellen, für den sie zuständig ist. Wir können uns das CME für Container ähnlich vorstellen wie ein Schiff oder einen LKW-Anhänger für Transportcontainer. Beide kümmern sich um die Container, für die sie in diesem Moment zuständig sind.

Wenn wir über CMEs schreiben, müssen wir ein weiteres Konzept einführen: den Unterschied zwischen Container und Instanz. Wir können einen Container als eine Offline-Einheit betrachten; um es klar zu sagen, er ist bereit, aber im Stand-by. Eine Instanz eines Containers hingegen ist ein live laufender Container.

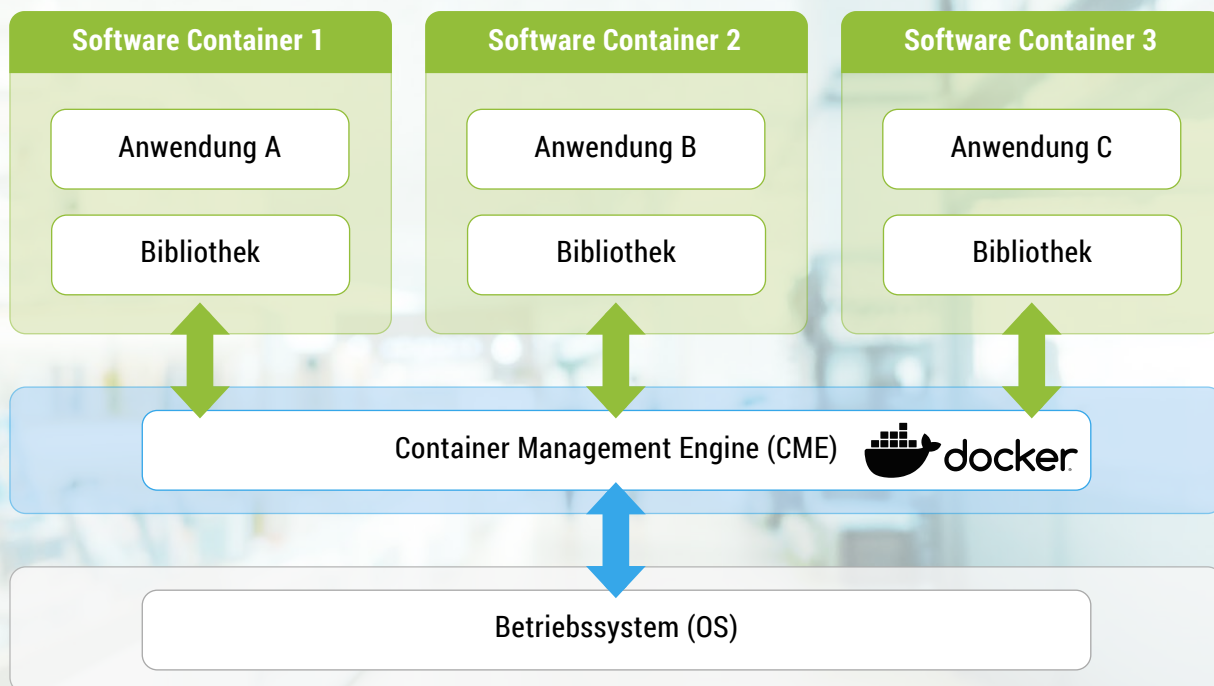


Abbildung 03

Die Stärke von Docker besteht darin, dass es die vom zugrunde liegenden Betriebssystem bereitgestellten Ressourcen für alle seine Container freigibt. Docker ist es egal, auf welchem Rechner es läuft, es muss nur wissen, dass es einige begrenzte Ressourcen zur Verfügung hat. Die Docker-Engine verwaltet diese Ressourcen, um die Bedürfnisse jedes einzelnen Containers, der ihr unterstellt ist, bestmöglich zu erfüllen.

Dadurch starten Anwendungen sehr schnell, denn sie verschwenden keine Zeit mit dem vorherigen Starten des Betriebssystems. Die CME sorgt für alles, was sie brauchen.

Eine Sache, die wir nicht unterschätzen sollten, ist die Fähigkeit von Docker, mehrere Instanzen desselben Containers auf einer einzigen Maschine zu starten. Betrachten wir das folgende Beispiel; stellen wir uns ein Webportal eines Internetshops vor. Bei normaler Aktivität nehmen wir an, dass eine Instanz des Containers der Shop-Anwendung ausreicht, um die Kundenanfragen zu bedienen. An einem Tag, an dem wir einen großen Rabatt im Shop ankündigen, würden wir eine Flut von Anfragen erwarten, die die einzelne Instanz der Anwendung möglicherweise nicht bewältigen kann. Der CME kann die Auslastung der Instanz überwachen und eine oder mehrere neue Instanzen desselben Containers starten, um den enormen Anstieg der Anfragen zu bewältigen. So kann der Shop der Flut von Anfragen standhalten und jeden einzelnen Kunden zufriedenstellen.

Im Allgemeinen könnte eine Anwendung aufgrund falscher Einstellungen oder aus anderen Gründen nicht mehr richtig funktionieren. Stellen Sie sich vor, ein Benutzer arbeitet mit einer Anwendung, stellt etwas falsch ein, und am Ende startet die Anwendung nicht mehr. Das ist die Art von Situation, die kein Benutzer erleben möchte. In einer traditionellen Umgebung könnte dieses Problem schwerwiegend sein. Da wir uns mit Containern auskennen, haben wir eine einfache Möglichkeit, diese Art von Problem zu lösen; wir stoppen einfach die Instanz der Anwendung und starten sie erneut mit sauberen Einstellungen.

Lassen Sie uns diesen Punkt betonen; wir starten die Instanz der Anwendung erneut, ohne den CME oder irgendetwas anderes zu stoppen.

Container-Technologie in Industrieumgebungen steuern und beherrschen

Die Secure Digital Platform von Endian kommt nicht nur als Motor für die digitale Transformation ins Spiel, sondern auch als Mittel zur Schaffung eines geschäftlichen Mehrwerts für Unternehmen. Sie integriert die Container-Technologie in ihre intelligenten sicheren 4i Edge X Gateways und unterstützt daher Container-basierte kundenspezifische Software-Anwendungen.

Wie der Name des Gateways „Edge“ schon andeutet, entsprechen derartige Geräte voll und ganz dem Konzept des „Edge Computing“, das dank der Container-Technologie Dienste in die Nähe der Anwender bringt und gleichzeitig den Aufwand für die Datenkommunikation im Netzwerk stark reduziert.

Industrieunternehmen können das leistungsstarke Ökosystem von Endian nutzen, um eigene Software auf ihren Produkten zu installieren. So hat ein Maschinenbauer beispielsweise die Möglichkeit, seine Produkte mit einer Software auszustatten, die zur Erfassung von Prozessdaten dient. Diese Daten kann er beispielsweise als Basis für Predictive Maintenance nutzen. Und nicht nur das: Gleichzeitig kann der Maschinenbauer seinen Kunden die Möglichkeit bieten, ihre individuelle Software zu installieren. Diese Option verschafft der Maschine einen zusätzlichen Mehrwert.

Die Kunden wiederum können ihre eigene Software auf den Maschinen aufspielen. Dabei könnte es sich beispielsweise um eine Software handeln, mit der sich Daten vorverarbeiten oder Maschinenzustände überwachen lassen.

Weder die Hersteller der Maschinen noch die Kunden müssen sich dabei um die Datensicherheit Gedanken machen, denn sie wird durch die zugrundeliegende Secure Digital Platform von Endian gewährleistet.

Generell kann die Industrie von der Leistungsfähigkeit des Ökosystems von Endian profitieren, denn es unterstützt ihre benutzerdefinierten Anwendungen vollständig, so dass sie es wie ihr eigenes System nutzen können.

Fazit

Im industriellen Umfeld ist die sichere digitale Kommunikation Voraussetzung für Industrie 4.0-Fähigkeit. Die Endian Secure Digital Platform ermöglicht eine sichere Kommunikation, und die Integration mit einem CME gibt den Anwendern so viel Flexibilität und Freiheit, dass sie die Lösung als integralen Bestandteil des internen Industriesystems betrachten können.

In diesem Dokument haben wir in einfachen Worten die Container-Technologie erklärt und die Vorteile und Anwendungsmöglichkeiten in industriellen Umgebungen aufgezeigt. Die Leistungsfähigkeit der Container-Management-Engines, die auf den 4i Edge X Secure Gateways von Endian eingesetzt werden und die zentral über das Switchboard verwaltet werden können, ist unserer Meinung nach sehr vielversprechend und offen für viele Entwicklungen.

Referenzen

- Endian „4i Edge X“: <https://www.endian.com/products/secure-digital-platform/endian-4i-edge-x/>
- Endian „Switchboard“: <https://www.endian.com/products/secure-digital-platform/switchboard/>
- Docker: <https://docker.com>
- Gartner Research - Edge Computing:
<https://www.gartner.com/en/doc/3889058-the-edge-completes-the-cloud-a-gartner-trend-insight-report>

