

Offene Steuerungsarchitekturen als Enabler für IT/OT- Konvergenz in der Vision von Industrie 4.0

Kerntechnologien flexibler Steuerungsarchitekturen



In der modernen Automatisierungstechnik müssen Produktionssysteme zunehmend schnell an wechselnde Anforderungen angepasst werden. Klassische, starre Steuerungsarchitekturen stoßen dabei an ihre Grenzen, weshalb flexible, modulare und erweiterbare Systeme erforderlich sind. Diese ermöglichen nicht nur kürzere Reaktionszeiten, sondern bilden auch die Grundlage für zukunftsfähige und effizient betreibbare Anlagen.

Im Zentrum steht dabei die Vision von Industrie 4.0 als durchgängige Digitalisierung und Vernetzung industrieller Prozesse. Maschinen, Steuerungen und IT-Systeme kommunizieren hierbei über offene Schnittstellen, wodurch datengetriebene und zunehmend selbstoptimierende Produktionssysteme entstehen. Industrie 4.0 bildet damit die technologische Grundlage für skalierbare und flexible Automatisierungssysteme.

Konsistenter Informationsfluss

Voraussetzung für dieses Zielbild ist ein konsistenter Informationsfluss zwischen OT- und IT-Ebene, sodass Produktionsdaten systemübergreifend ausgewertet und zur Optimierung genutzt werden können. In der Praxis wird dieser Datenaustausch jedoch durch heterogene Protokolle, unterschiedliche Datenmodelle, organisatorische Trennung, Netzwerkgrenzen und Legacy-Systeme erheblich erschwert.

Daraus ergibt sich die Notwendigkeit eines Paradigmenwechsels von proprietären, isolierten OT-Insellösungen hin zu offenen, interoperablen Systemarchitekturen. Die IT/OT-Konvergenz beschreibt dabei das Zusammenwachsen beider Domänen zu einer durchgängigen Systemlandschaft mit nahtlosem Datenaustausch.

Offene Steuerungsarchitekturen stellen hierfür eine zentrale technische Grundlage auf der OT-Ebene dar (Bild 1). Durch Standardisierung, Modularisierung und Interoperabilität ermöglichen sie die IT/OT-Konvergenz und bilden damit einen wesentlichen Enabler für Industrie 4.0.

Geschlossene und offene Steuerungsarchitekturen

Klassische Steuerungssysteme sind in der Regel proprietär und weisen nur eine eingeschränkte Interoperabilität zwischen Komponenten unterschiedlicher Hersteller auf. Sie basieren auf dedizierter Steuerungshardware und geschlossenen Systemlandschaften aus SPSen, Kommunikationskarten, Busklemmen und herstellerspezifischer Engineering-Software. Neue Funktionen wie etwa Kameraerkennung oder KI-Module müssen häufig als proprietäre Erweiterungen integriert werden, während die Anbindung externer Systeme meist nur über Gateways, spezifische Treiber oder eingeschränkte Schnittstellen möglich ist.

Offene Steuerungsarchitekturen

Demgegenüber verfolgen offene Steuerungsarchitekturen das Ziel einer herstellerunabhängigen und modular erweiterbaren Systemlandschaft. Durch standardisierte Schnittstellen wird sowohl die Integration neuer Funktionen und Module als auch die nahtlose Anbindung an IT- und OT-Systeme ermöglicht. Eine serviceorientierte Architektur sowie konsequente Modularisierung erlauben dabei flexible Funktionsanpassungen und vereinfachte Software-Updates.

Zentrale Merkmale moderner Steuerungsarchitekturen sind die Entkopplung von Hardware und Funktionalität sowie die Modularisierung von Software, sodass Funktionen bedarfsgerecht hinzugefügt oder entfernt werden können. Voraussetzung hierfür ist ein flexibles

Deployment-Konzept, das durch Virtualisierung die Hardwareunabhängigkeit von Steuerungsmodulen ermöglicht und deren einfache Bereitstellung unterstützt.

Ergänzend bildet die Standardisierung von Schnittstellen und Kommunikationsmechanismen die Grundlage für eine interoperable Systemarchitektur. Einheitliche Steuerungsschnittstellen sowie echtzeitfähige, standardisierte Kommunikationsverfahren ermöglichen konvergente Netze, in denen IT- und OT-Systeme gemeinsam operieren und Daten herstellerübergreifend austauschen.

Container-basierte Virtualisierung - vPLCs

Heute bestehen Produktionsumgebungen typischerweise aus heterogenen Steuerungssystemen und herstellerspezifischen Komponenten. Bereits der parallele Betrieb von Speicherprogrammierbaren Steuerungen (SPS), Robotersteuerungen (RC) und numerischen Steuerungen (NC) erfordert unterschiedliche proprietäre Steuerungsrechner sowie spezialisierte Kommunikationsinfrastrukturen. Ziel ist daher die Konsolidierung dieser heterogenen Steuerungsmodule auf standardisierter Hardware (Bild 2).

Containerbasierte Virtualisierung ermöglicht dabei die Kapselung von Steuerungsfunktionen als portable Deploymentseinheiten. Ein Container umfasst die Anwendung selbst sowie alle zugehörigen Bibliotheken, Binaries, Laufzeitumgebungen, Abhängigkeiten und Konfigurationsparameter wie Umgebungsvariablen. Technisch erfolgt die Virtualisierung auf Betriebssystemebene, wobei Container den Kernel des Hostsystems gemeinsam nutzen, anstatt ein eigenes Betriebssystem zu betreiben. Dadurch sind sie deutlich ressourceneffizienter und schneller als klassische virtuelle Maschinen. Isolation und Ressourcensteuerung werden durch Linux-Kernel-Mechanismen wie Namespaces und cgroups realisiert.



Autorin:
Rebekka Neumann, M. Sc.
Gruppenleiterin
„Echtzeitkommunikation und
Steuerungshardware“
Institut für Steuerungstechnik
der Werkzeugmaschinen und
Fertigungseinrichtungen (ISW)
Universität Stuttgart
<http://www.isw.uni-stuttgart.de>

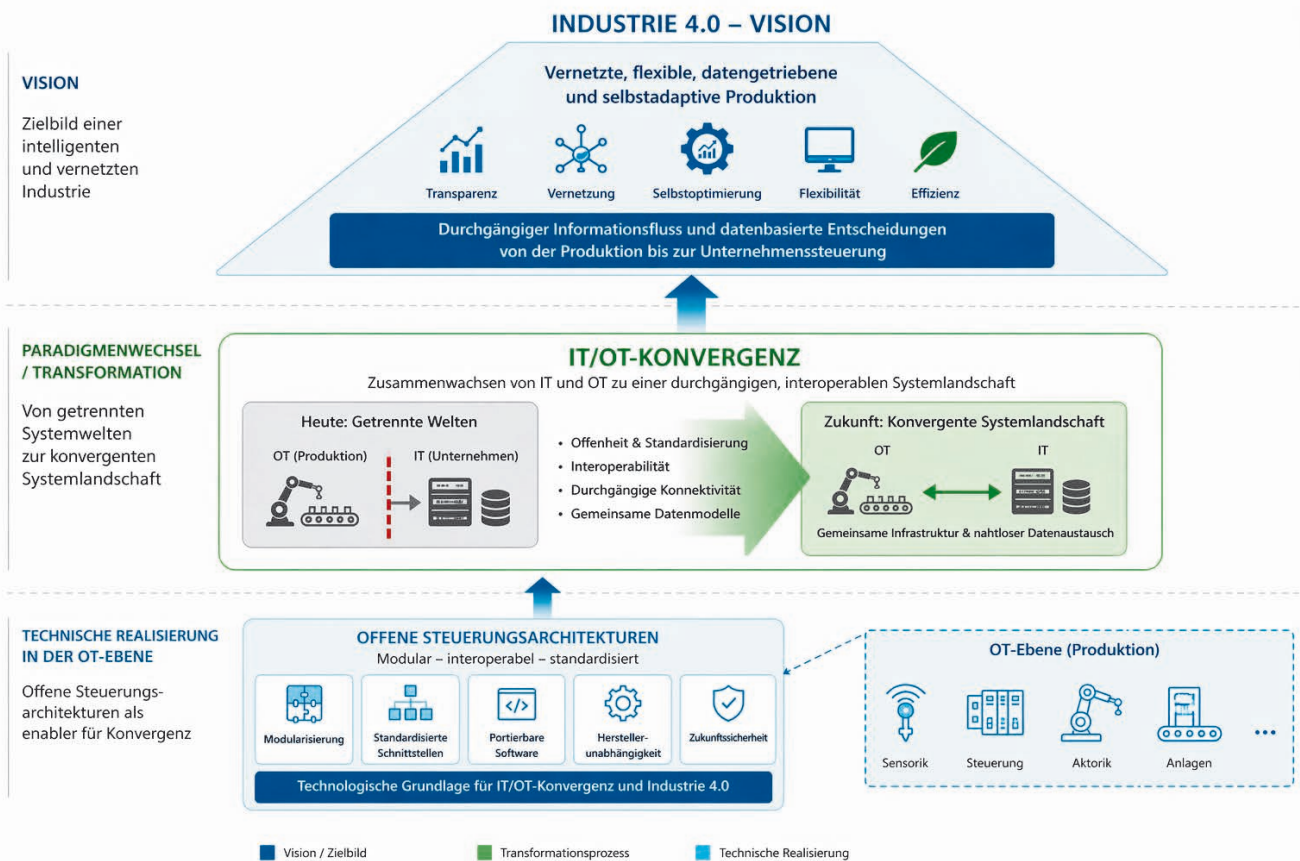


Bild 1: Zusammenhang zwischen Industrie 4.0, IT/OT-Konvergenz und offenen Steuerungsarchitekturen

Echtzeit-Container

Im industriellen Umfeld unterscheiden sich solche Container jedoch wesentlich von IT-Anwendungen, da Steuerungsfunktionen harte Echtzeitanforderungen erfüllen müssen. Containerisierte Steuerungsmodulare sind daher als Echtzeit-Container auszuführen, bei denen deterministische Latenzen und geringe Jitter-Werte gewährleistet sein müssen. Dies erfordert den Betrieb des Hostsystems als Echtzeitsystem, inklusive Echtzeitbetriebssystem, deterministischem Scheduling sowie CPU-Isolation. Virtualisierte SPSen (vPLCs) stellen solche Echtzeit-Container dar, die IEC 61131-3-basierte Steuerungsprogramme zusammen mit der jeweiligen Runtime ausführen.

Kommerzielle Lösungen wie Phoenix Contact (PLCnext), CODESYS und ctrlX AUTOMATION zeigen bereits erste Ansätze dieser Architektur, wobei letztere auf Snap-basierte Paketierung statt klassischer Container setzt. Diese Systeme stellen einen wichtigen Schritt in Richtung

hardwareunabhängiger und konsolidierter Steuerungsarchitekturen dar. Dennoch bleiben Einschränkungen bestehen, insbesondere durch proprietäre Engineering-Umgebungen und die nur eingeschränkt flexible Integration von Drittanbietersoftware oder Eigenentwicklungen, was der vollständigen Offenheit der Architektur entgegensteht.

Konvergente Kommunikation

In klassischen Steuerungsarchitekturen wird strikt zwischen deterministischen Feldbussystemen in der OT und Ethernet- bzw. TCP/IP-basierten IT-Netzwerken unterschieden. Beide Kommunikationswelten folgen unterschiedlichen Paradigmen: Während IT-Kommunikation auf Best-Effort-Übertragung mit variabler Latenz basiert, sind Feldbussysteme auf deterministische Kommunikation mit festen Zykluszeiten und harten Echtzeitanforderungen ausgelegt. Aufgrund dieser Auslegung sind Feldbusse zur Laufzeit nur eingeschränkt flexibel und schwer modifizierbar.

Konvergente Netze

Für eine IT/OT-Konvergenz müssen jedoch konvergente Netze geschaffen werden, in denen IT- und OT-Traffic gemeinsam koexistieren können. Voraussetzung hierfür ist der Einsatz offener und interoperabler Kommunikationsstandards anstelle proprietärer Protokolle. Offene Kommunikationsstandards sind beispielsweise EtherCAT und Ethernet, wobei EtherCAT ein Ethernet-basierter zyklischer Feldbus ist.

In konvergenten Netzwerken müssen die Echtzeitanforderungen der OT weiterhin vollständig eingehalten werden. Dies erfordert garantierte Zeitslots für zeitkritischen OT-Traffic innerhalb des Kommunikationszyklus, während IT-Traffic die verbleibende Bandbreite nutzen kann.

Time-Sensitive Networking

Ein geeigneter Ansatz zur Umsetzung solcher konvergenter Kommunikationsarchitekturen ist Time-Sensitive Networking (TSN). TSN stellt keine eigenständige Netzwerktechnologie dar, sondern eine Erweiterung von Ethernet zur Sicherstellung

deterministischer Kommunikation. Wesentliche Mechanismen umfassen präzise Zeitsynchronisation (IEEE 1588 Precision Time Protocol), zeitscheibenbasierte Übertragungsplanung, prioritätsbasierte Frame-Verarbeitung mit präemptivem Verhalten sowie Verfahren zur Frame-Replikation und Duplikat-Eliminierung. Dadurch wird die Grundlage geschaffen, Echtzeit- und IT-Kommunikation in einem gemeinsamen Netzwerk deterministisch und interoperabel zu betreiben.

Feldbusse und TSN

Für eine IT/OT-Konvergenz muss die Übertragung von Echtzeitdaten in konvergenten Netzwerken ermöglicht werden. Die Kommunikation zwischen Sensorik, Aktorik und Steuerung erfolgt in klassischen Architekturen über Feldbussysteme. Wie zuvor beschrieben, sind diese jedoch aufgrund ihrer proprietären Struktur und fehlenden IP-Integration nur eingeschränkt für die Einbindung in IT-Netzwerke und damit für konvergente Architekturen geeignet.

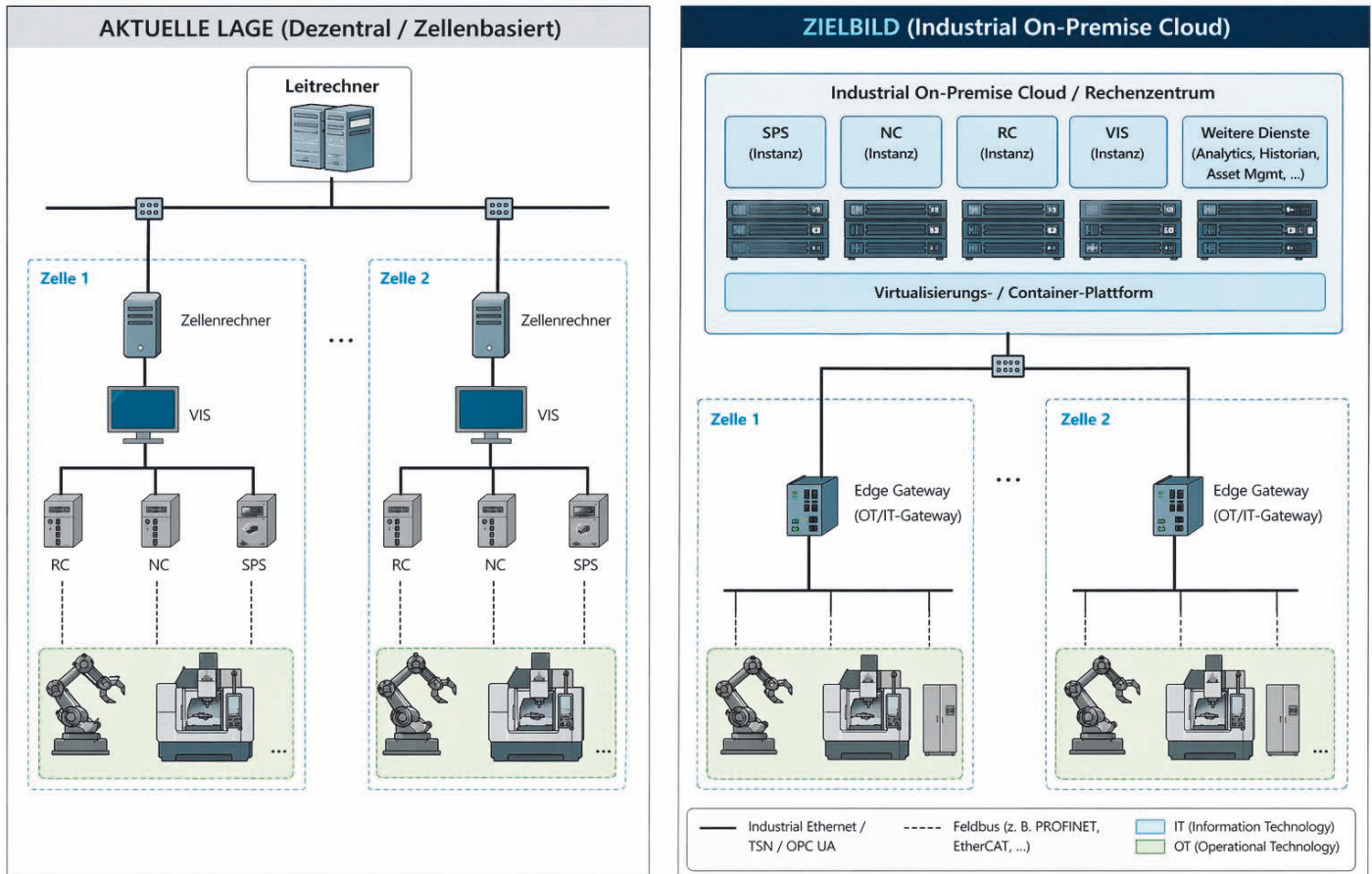


Bild 2: Aktuelle Lage und Zielbild

TSN ermöglicht grundsätzlich den Betrieb von Echtzeit- und IT-Traffic innerhalb eines gemeinsamen Ethernet-basierten Netzwerks. TSN ist jedoch kein Feldbus und kann diesen nicht direkt ersetzen, wie in Bild 3 dargestellt.

Stattdessen stellt TSN Mechanismen zur Zeitsynchronisation sowie zur deterministischen Übertragung durch Traffic-Klassen, Scheduling und Priorisierung bereit, wodurch das zeitliche Übertragungsverhalten von Feldbussen im Ethernet-Umfeld abgebildet werden kann.

OPC UA

Um die Funktionalität klassischer Feldbusse in konvergenten, TSN-basierten Netzwerken abzubilden, fehlen insbesondere drei wesentliche Aspekte: ein strukturiertes Datenmodell, semantische Beschreibung der Daten sowie die standardisierte Definition von Geräteschnittstellen. Diese Funktionen werden durch OPC UA bereitgestellt.

OPC UA ist ein herstellerunabhängiges Kommunikations- und

Informationsframework für industrielle Systeme, das sowohl Datenmodelle als auch Semantik und Schnittstellenbeschreibungen standardisiert. Branchenspezifische Informationsmodelle werden dabei über sogenannte Companion Specifications definiert.

Für die Kommunikation zur IT-Ebene wird klassisch das Client-Server-Modell verwendet. Ergänzend entwickelt die Field Level Communication (FLC)-Arbeitsgruppe ein Modell für die Echtzeitkommunikation im Feld, bei dem Publish-Subscribe-Mechanismen in Kombination mit TSN für deterministische, netzwerkbasierete Kommunikation eingesetzt werden.

Zusammenfassung

Industrie 4.0 erfordert eine durchgängige IT/OT-Konvergenz, um Produktionssysteme flexibel, datengetrieben und skalierbar zu gestalten. Klassische, proprietäre Steuerungsarchitekturen verhindern dies aufgrund fehlender Interoperabilität, heterogener Standards und starrer Hardwarebindung.

Offene Steuerungsarchitekturen adressieren diese Limitationen durch Modularisierung, Standardisierung und die Entkopplung von Hardware und Software, wodurch eine konsolidierte, erweiterbare Systemlandschaft entsteht. Zentrale technologische Enabler sind containerbasierte Virtualisierung (z. B. vPLCs) sowie konvergente Kommunikationsansätze wie TSN in Kombination mit OPC UA zur Verbindung von Echtzeitfähigkeit und semantischer Interoperabilität.

Wer schreibt:

Rebekka Neumann leitet die Forschungsgruppe „Echtzeitkommunikation und Steuerungshardware“ am Institut für Steuerungstechnik der Werkzeugmaschinen (ISW) der Universität Stuttgart. Ihre Forschung konzentriert sich auf konvergente Kommunikationsarchitekturen und Virtualisierung in der industriellen Automatisierung, mit einem besonderen Schwerpunkt auf dynamischem Echtzeit-Scheduling. ◀

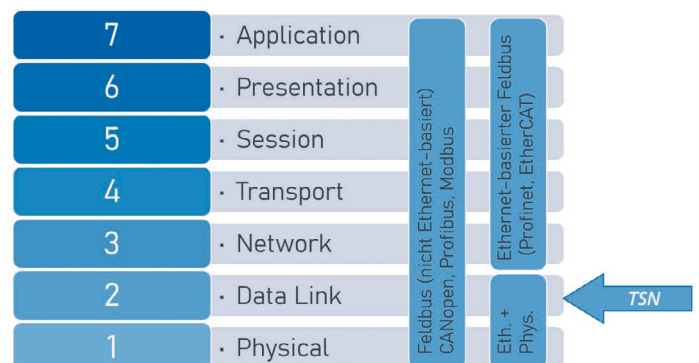


Bild 3: Feldbusse und TSN im OSI-Schichtenmodell