Software Defined Everything - Architekturprinzipien und Automatisierung

Software Defined Everything

Software Defined Everything (SDx) beschreibt einen Architekturansatz, bei dem technische Ressourcen und Funktionen weitgehend von ihrer konkreten Hardwareimplementierung abstrahiert und softwarebasiert gesteuert, konfiguriert und verwaltet werden. Dabei werden die Funktionalitäten nicht mehr ausschließlich durch die Eigenschaften und Konfiguration der zugrunde liegenden Hardware bestimmt, sondern können über Software definiert, gesteuert und weiterentwickelt werden. SDx schafft damit programmatisch nutzbare Bausteine, die über definierte, möglichst standardisierte Schnittstellen bereitgestellt und verwaltet werden können. Diese Abstraktion reduziert die Abhängigkeit von spezifischen Hardwareimplementierungen und ermöglicht eine flexible, softwaregesteuerte Nutzung und Weiterentwicklung der zugrunde liegenden technischen Ressourcen und Funktionen. SDx ist dabei nicht als einheitlich normiertes Architekturmodell zu verstehen, sondern als übergreifendes Prinzip, das je nach technologischer Domäne unterschiedlich umgesetzt wird.

Eine konkrete Ausprägung dieses Ansatzes ist das Software Defined Data Center (SDDC). Hierbei werden die klassischen Infrastrukturdomänen Compute, Storage und Networking durch softwaredefinierte Technologien abstrahiert und als flexibel nutzbare Ressourcen bereitgestellt. Die Begriffe Software Defined Compute (SDC), Software Defined Storage (SDS) und Software Defined Networking (SDN) beschreiben dabei die jeweiligen technologischen Ausprägungen innerhalb eines SDDC.

VMware Cloud Foundation (VCF) veranschaulicht diesen Ansatz durch die Integration seiner softwaredefinierten Infrastrukturkomponenten. Dabei übernimmt VMware vSphere die Virtualisierung und Bereitstellung der Compute-Ressourcen (SDC). VMware vSAN realisiert die softwarebasierte Bereitstellung und Verwaltung von Storage-Ressourcen (SDS), während VMware NSX die Virtualisierung und softwarebasierte Steuerung der Netzwerkinfrastruktur (SDN) ermöglicht.

IT-Automatisierung

Die IT-Automatisierung beschreibt die Fähigkeit, IT-Ressourcen - einschließlich softwaredefinierter Ressourcen - und Funktionen durch definierte Regeln, Workflows, Schnittstellen und Ereignisse zu kombinieren, zu konfigurieren und zu betreiben. Technologien und Konzepte wie Infrastructure as Code (IaC), automatisierte Bereitstellung (Provisionierung), Konfigurationsmanagement und Orchestrierung ermöglichen es, Infrastrukturkomponenten und Applikationen weitgehend ohne wiederkehrende manuelle Eingriffe bereitzustellen und über ihren gesamten Lebenszyklus zu verwalten.

Eine wesentliche technische Grundlage hierfür bilden Application Programming Interfaces (APIs) und deklarative Konfigurationsmodelle. Sie ermöglichen es, die Funktionen und Ressourcen softwaredefinierter Infrastrukturen über definierte Schnittstellen anzusprechen, gewünschte Zustände zu beschreiben und Änderungen programmatisch anzustoßen. Automatisierungsplattformen nutzen diese Möglichkeiten, um Ressourcen bereitzustellen, Konfigurationen anzupassen und übergreifende Prozesse zu steuern.

Ein Beispiel hierfür ist VCF Automation. Die softwaredefinierten Infrastrukturkomponenten bilden die technologische Grundlage, auf der die Ressourcen über definierte Services und Schnittstellen, Self-Service-Funktionen und Automatisierungsworkflows bereitgestellt und konsumiert werden können. Mit Konzepten und Funktionen wie IaC, Blueprints und Policy-basierter Governance wird die softwaredefinierte Infrastruktur als bedarfsgerecht nutzbarer IT-Service verfügbar gemacht.

Eine ergänzende Perspektive bietet Red Hat Ansible. Mit Ansible lassen sich Automatisierungsaufgaben in Playbooks beschreiben und wiederverwendbar umsetzen. Module ermöglichen die Ausführung konkreter Automatisierungsaufgaben. Collections bündeln wiederverwendbare Automatisierungsinhalte, während Roles deren Strukturierung und Wiederverwendung unterstützen. Das Einsatzspektrum reicht von der Konfiguration von Betriebssystemen und Netzwerkkomponenten über Softwareinstallationen und Applikationsdeployments bis hin zur Bereitstellung und Verwaltung von Infrastrukturressourcen. Mit der Ansible Automation Platform (AAP) werden diese Automatisierungsinhalte zentral ausführbar und durch übergreifende Workflows orchestrierbar. Event-Driven Ansible ermöglicht es, auf definierte Ereignisse aus der IT-Infrastruktur zu reagieren und darauf basierend automatisierte Aktionen auszulösen.

Während VCF Automation insbesondere die Bereitstellung und konsumierbare Nutzung softwaredefinierter Infrastrukturressourcen adressiert, erweitert Ansible die Automatisierung um Konfigurations-, Integrations- und Betriebsprozesse über unterschiedliche IT-Domänen hinweg. Beide Ansätze können sich ergänzen und gemeinsam dazu beitragen, nicht nur die initiale Bereitstellung, sondern auch die Konfiguration, Integration und kontinuierliche Anpassung von IT-Services über deren gesamten Lebenszyklus zu automatisieren.

Zusammenspiel von SDx und IT-Automatisierung

SDx und IT-Automatisierung bilden damit zwei eng miteinander verbundene Aspekte einer modernen IT-Architektur: Während SDx die programmatisch nutzbaren Infrastrukturbausteine und deren softwarebasierte Steuerbarkeit bereitstellt, übernimmt die IT-Automatisierung deren standardisierte, wiederholbare und domänenübergreifende Nutzung. APIs, deklarative Modelle und Automatisierungsworkflows bilden dabei die technische Verbindung zwischen beiden Ansätzen. So lassen sich softwaredefinierte Infrastrukturen effizient in konsumierbare IT-Services überführen. Je stärker Systeme durch Software definiert werden, desto wichtiger wird die Automatisierung.

SDx – ein branchenübergreifendes Architekturprinzip

Über die klassische IT-Infrastruktur hinaus lässt sich das Software-Defined-Prinzip auch in anderen technologischen Domänen wiederfinden. Beispiele hierfür sind die Mobilität mit dem Software Defined Vehicle (SDV) sowie der Verteidigungsbereich mit Software Defined Defence (SDD).

Die technischen Ausprägungen unterscheiden sich teilweise erheblich, die zugrunde liegenden Architekturprinzipien weisen jedoch Gemeinsamkeiten auf. Dazu gehören insbesondere die Abstraktion und Modularisierung technischer Funktionen, die Entkopplung von Hardware und Software, die Bereitstellung definierter Schnittstellen sowie die Möglichkeit einer softwarebasierten Steuerung und Weiterentwicklung über den gesamten Lebenszyklus.

Dabei handelt es sich um architektonische Analogien und funktionale Gemeinsamkeiten, nicht um direkte Gleichsetzungen einzelner Technologien oder Komponenten. So lässt sich beispielsweise SDC nicht unmittelbar mit einem bestimmten Fahrzeugrechner gleichsetzen, auch wenn beide vergleichbare Aufgaben hinsichtlich der Bereitstellung und Verarbeitung von Rechenleistung erfüllen.

Eine Sonderstellung nimmt Software Defined Defence (SDD) ein. Hierbei handelt es sich weniger um eine einzelne technische Plattform oder Infrastrukturdomäne als vielmehr um ein übergeordnetes Paradigma für die softwarebasierte Digitalisierung, Vernetzung und kontinuierliche Weiterentwicklung militärischer Systeme und Fähigkeiten.

Branchenspezifische Ausprägungen der SDx-Architekturprinzipien

SDDC, SDV und SDD folgen ähnlichen Architekturprinzipien, unterscheiden sich aber erheblich hinsichtlich des betrachteten Systems, ihrer technischen Umsetzung und der Anforderungen an dessen Betrieb. Sie repräsentieren drei unterschiedliche Anwendungsbereiche desselben übergeordneten Architekturgedankens.

SDDC-Prinzipien

Das Software Defined Data Center beschreibt die **softwaredefinierte Architektur einer IT-Infrastruktur** und basiert auf der Abstraktion und softwarebasierten Bereitstellung von Compute-, Storage- und Netzwerkressourcen. Ziel ist eine flexibel nutzbare, skalierbare und automatisierbare IT-Infrastruktur. Es bildet damit eine integrierte Infrastrukturplattform, die über APIs und Automatisierungsmechanismen programmatisch verwaltet und konsumiert werden kann. Charakteristisch ist die Entkopplung der Infrastrukturressourcen von ihrer physischen Implementierung sowie deren Bereitstellung als flexibel nutzbare Services.

SDV-Prinzipien

Das Software Defined Vehicle beschreibt die **softwaredefinierte Architektur des Fahrzeugs** und verfolgt den Ansatz, Fahrzeugfunktionen zunehmend von ihrer spezifischen Hardwareimplementierung zu entkoppeln. Ziel ist ein Fahrzeug, dessen Funktionen über Software weiterentwickelt, aktualisiert und ergänzt werden können. Hierbei verschiebt sich der Schwerpunkt von einer Vielzahl funktional abgegrenzter elektronischer Steuergeräte hin zu einer stärker integrierten Fahrzeugarchitektur mit zentralisierten, domänenorientierten Rechenplattformen. Charakteristisch ist, dass Fahrzeugfunktionen zunehmend durch modulare Software definiert und über den gesamten Lebenszyklus des Fahrzeugs weiterentwickelt werden können.

SDD-Prinzipien

Software Defined Defence (SDD) bezeichnet einen übergeordneten Architektur- und Transformationsansatz, der die Modularisierung, Vernetzung und **softwarebasierte Weiterentwicklung militärischer** Systeme und deren **Fähigkeiten** verfolgt. SDD umfasst dabei nicht nur technologische und architektonische Aspekte, sondern auch Anforderungen an Governance, Standardisierung und Beschaffungsprozesse. Ziel ist eine höhere Anpassungsfähigkeit und Interoperabilität über den gesamten Systemlebenszyklus. Charakteristisch ist somit nicht eine einzelne technische Architektur, Infrastruktur oder Plattform, sondern die softwarebasierte Weiterentwicklung und Integration militärischer Fähigkeiten über unterschiedliche Systeme und Domänen hinweg. SDD verfolgt insgesamt einen umfassenderen Transformationsansatz. Die Integration und Aktualisierung solcher Systeme erfolgt unter hohen Anforderungen an Interoperabilität, Informationssicherheit, Validierung und kontrollierte Freigabe.

Gemeinsame Architekturprinzipien

Diese Gegenüberstellung verdeutlicht, dass die gemeinsame Grundlage dieser Architekturansätze in der Abstraktion, Modularisierung und softwarebasierten Steuerbarkeit technischer Funktionen liegt. Ihre konkrete Ausgestaltung wird jedoch maßgeblich durch die jeweiligen branchenspezifischen Anforderungen bestimmt.

Automatisierung als operativer Enabler softwaredefinierter Produkte und Fähigkeiten

Mit zunehmender Softwaredefinition technischer Systeme erweitert sich auch die Bedeutung der Automatisierung. Sie beschränkt sich nicht mehr ausschließlich auf die Bereitstellung und Konfiguration technischer Ressourcen, sondern umfasst zunehmend die Entwicklung, Integration, Validierung und Bereitstellung softwaredefinierter Systeme sowie deren kontinuierliche Weiterentwicklung über den gesamten Lebenszyklus.

Beim SDV steht eine durchgängige Automatisierungskette im Mittelpunkt, die von der Softwareentwicklung über die Integration und Validierung bis hin zur Bereitstellung und Aktualisierung der Fahrzeugflotte reicht. Automatisierung wird damit zu einem wesentlichen Bestandteil der Produktentwicklung und des Produktbetriebs. Fahrzeugfunktionen sind nicht mehr ausschließlich durch den Auslieferungszustand festgelegt, sondern können innerhalb der technischen und sicherheitsbezogenen Rahmenbedingungen über den gesamten Fahrzeuglebenszyklus weiterentwickelt und ergänzt werden.

Bei SDD geht diese Perspektive über die Weiterentwicklung einzelner Softwarekomponenten hinaus. Hier steht insbesondere die Integration neuer softwarebasierter Fähigkeiten in bestehende militärische Systeme und Plattformen im Vordergrund. Automatisierte Entwicklungs-, Integrations-, Validierungs- und Bereitstellungsprozesse unterstützen dabei die kontrollierte Weiterentwicklung dieser Fähigkeiten.

Eine Verbindung zu den zuvor betrachteten Automatisierungstechnologien VCF Automation und Ansible entsteht auf der Ebene der unterstützenden IT- und Entwicklungsinfrastruktur. Beide können beispielsweise die Cloud-Infrastruktur einer SDV-Software-Factory oder die IT-Plattform einer SDD-Entwicklungsumgebung automatisiert bereitstellen, konfigurieren und betreiben. Sie unterstützen damit die zugrunde liegenden Entwicklungs-, Integrations- und Testumgebungen, übernehmen jedoch nicht automatisch die Steuerung sicherheitskritischer Fahrzeugfunktionen oder das missionsbezogene Laufzeitmanagement militärischer Systeme.

Somit zeigt sich, dass Infrastrukturautomatisierung im SDV- und SDD-Kontext weiterhin eine wichtige Grundlage darstellt, jedoch nur einen Teil einer umfassenderen Automatisierungsarchitektur bildet. Erst im Zusammenspiel mit domänenspezifischen Entwicklungs-, Integrations-, Validierungs- und Deployment-Prozessen entsteht eine durchgängige Automatisierung des Produkt- und Systemlebenszyklus.

Fazit

Software Defined Everything beschreibt ein branchenübergreifendes Architekturprinzip, das die Abstraktion, Modularisierung und softwarebasierte Steuerung technischer Systeme in den Mittelpunkt stellt. Die Betrachtung von SDDC, SDV und SDD verdeutlicht, dass diese Prinzipien in unterschiedlichen technologischen Domänen Anwendung finden, deren konkrete Umsetzung jedoch durch die jeweiligen branchenspezifischen Anforderungen bestimmt wird.

Die zunehmende Softwaredefinition technischer Systeme verändert zugleich die Anforderungen an deren Automatisierung. Während im klassischen IT-Umfeld die Bereitstellung und Verwaltung von Infrastrukturressourcen im Vordergrund stehen, gewinnen bei softwaredefinierten Produkten und Fähigkeiten die automatisierte Entwicklung, Integration, Validierung und kontinuierliche Weiterentwicklung an Bedeutung.

Dabei lassen sich zwei eng miteinander verbundene Automatisierungsebenen unterscheiden, die aber unterschiedliche Aufgaben erfüllen:

- **Infrastruktur-** und **IT-Prozessautomation** umfassen die programmatische Steuerung der technischen Plattform, auf der Entwicklung, Integration und Betrieb stattfinden.

- **Software Lifecycle Automation** hingegen fokussiert die programmatische Steuerung der Produkt- und Systementwicklung sowie deren Integration, Validierung und kontinuierliche Weiterentwicklung.

Beide Ebenen ergänzen sich und bilden gemeinsam die Grundlage für eine durchgängige Automatisierungsarchitektur. Damit wird Automatisierung zu einem wesentlichen Bestandteil moderner softwaredefinierter Architekturen über ihren gesamten Produkt- und Systemlebenszyklus.

SDx schafft die programmierbaren Bausteine und IT-Automatisierung übernimmt die Aufgabe, diese intelligent zu kombinieren, zu orchestrieren und in IT-Services zu überführen. Erst das Zusammenspiel dieser Ansätze ermöglicht eine konsequent softwaregesteuerte und skalierbare IT-Servicebereitstellung.