// SOFTWARE DEFINED DEFENSE

SOFTWARE DEFINED DEFENSE

Was landbasierte Verteidigungsplattformen vom automotivebasierten Software Defined Vehicle lernen können

Autoren: Stefan Meyer, Julio Hohloch, Bernd Schaefer

Inhaltsverzeichnis

Abstract

Software Defined Defense (SDD) steht für den Wandel von hardwarezentrierten Systemen hin zu flexiblen, softwaregesteuerten Verteidigungsplattformen. Ziel ist eine kontinuierlich anpassbare, interoperable und sichere Verteidigungsfähigkeit durch Updates, modulare Erweiterungen und standardisierte Schnittstellen.

Aus den Erfahrungen der Automobilindustrie mit dem Software Defined Vehicle (SDV) lassen sich wertvolle Lehren ziehen: Standardisierung, klare Softwarearchitekturen und funktionsorientierte Organisationen sind entscheidend, während ein zu starker Hardwarefokus und fehlende Integration den Fortschritt hemmen.

Das daraus abgeleitete Software Defined Defense Vehicle verbindet SDD-Prinzipien mit SDV-Methoden: zonenbasierte Hardware, zentrale Rechenplattformen, Abstraktionsebenen und einheitliche Datenmodelle schaffen Flexibilität, Cyber-Resilienz und Updatefähigkeit. Sichere Backend-Anbindungen ermöglichen datengetriebene Wartung und Einsatzsteuerung, erfordern jedoch konsequente Security-by-Design-Ansätze.

Organisatorisch wird Software zum Kernprodukt, Hersteller agieren als Systemintegratoren, und neue Vertragsmodelle fördern kontinuierliche Leistungsfähigkeit statt Einmalbeschaffung.

SDD kann damit nicht mehr als optionaler Ansatz gesehen werden, sondern vielmehr als der Schlüssel für Geschwindigkeit, Anpassbarkeit und technologische Überlegenheit in der Verteidigung.

1. Software Defined Defense ist kein „Nice to Have“

Software Defined Defense (SDD) ist aktuell einer der zentralen Trends in der militärischen Technologieentwicklung. Das Konzept beschreibt aber mehr als nur einen weiteren Modebegriff, nämlich einen grundlegenden Paradigmenwechsel. Es ist der Übergang von gewachsenen, hardwarezentrierten Systemlandschaften hin zu anpassungsfähigen Plattformen, deren Wirkung, Sicherheit und Interoperabilität primär durch Software bestimmt und über den kompletten Lebenszyklus fortlaufend weiterentwickelt wird.

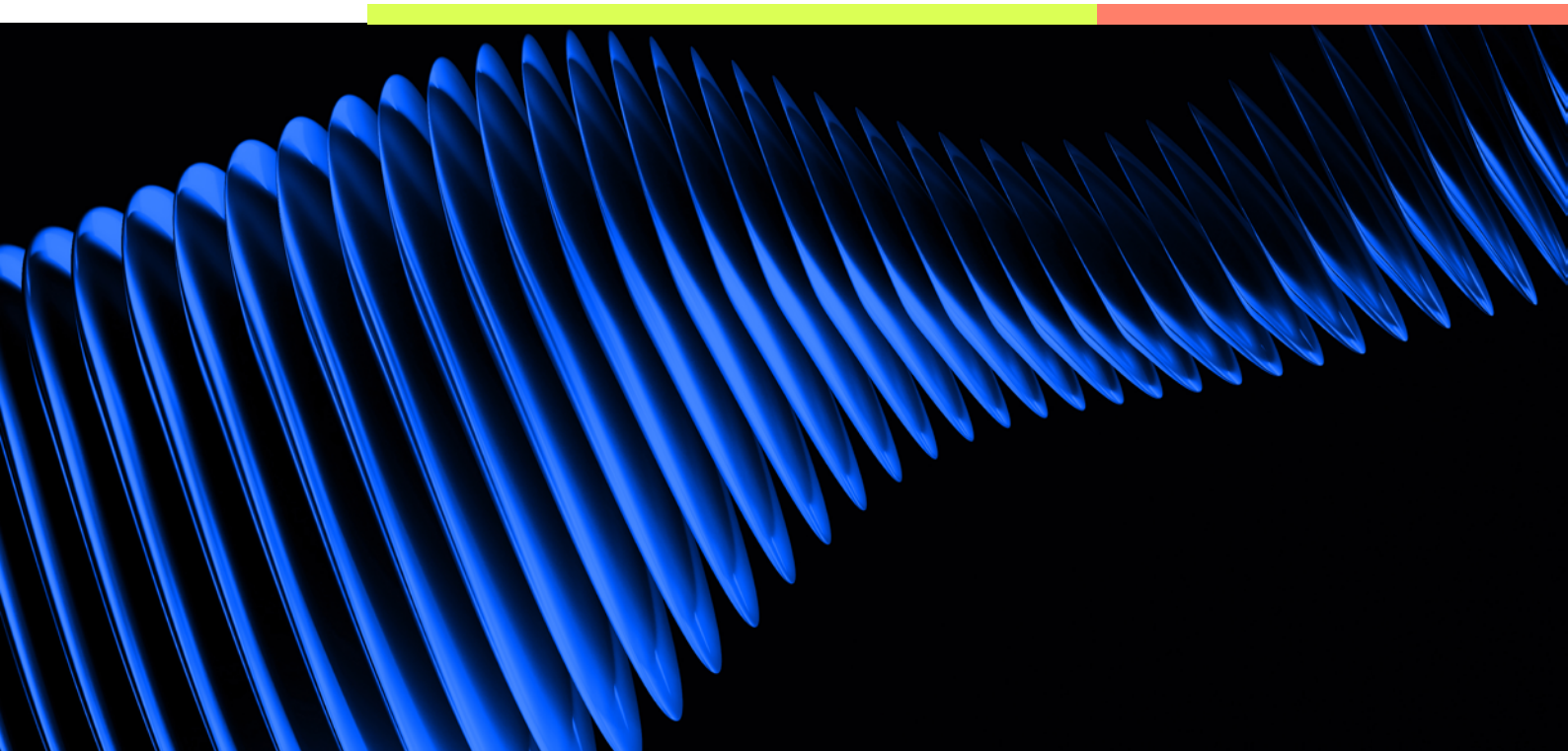
Angesichts der zunehmenden Dynamik sicherheitspolitischer Bedrohungen und der rasant voranschreitenden Digitalisierung ist die Fähigkeit, Streitkräfte schnell, flexibel und wirksam an neue Szenarien anzupassen, entscheidend für ihre Handlungsfähigkeit. Die Konflikte in der Ukraine und im Nahen Osten zeigen deutlich, in welch rasantem Tempo sich die Einsatzszenarien und -bedarfe verändern können. Ausrüstung, die gestern noch hochaktuell war, kann morgen schon durch neue Taktiken, Sensorik und Wirkmittel veraltet sein, wenn sie nicht schnell genug angepasst werden kann.

SDD verspricht genau dort Abhilfe: Eine Verteidigungsfähigkeit, die nicht durch starre Release-Zyklen begrenzt wird, sondern über kontinuierliche Software-Updates, Feature-Erweiterungen und sicherheitsrelevante Patches Schritt hält.

Die Vorteile von SDD liegen auf der Hand:

- Steigerung der Leistungsfähigkeit durch kontinuierliche Software-Updates und Funktionsverbesserungen im laufenden Betrieb
- Erhöhte Flexibilität bei der Anpassung an neue Bedrohungen, Einsatzszenarien oder Technologien
- Längere Nutzungsdauer komplexer Systeme dank HW-Modernisierung via Softwareupdates durch Entkopplung von Hardware und Software
- Bessere Interoperabilität durch standardisierte Datenmodelle, Schnittstellen und Architekturen
- Kosteneffizienz durch modulare Upgrades statt teurer Neubeschaffungen

Um diese Potentiale ausschöpfen zu können, darf SDD nicht als einzelnes Insel-IT-Projekt verstanden werden, sondern muss sich als ein Transformationsprogramm verstehen, bei dem Software- und Hardwarearchitektur neu gedacht werden. Die Softwarearchitektur steht im Zentrum und am Anfang der Entwicklung neuer, auf SDD basierender Fahrzeuge, die Hardwarearchitektur stellt die nötigen Ressourcen in einer möglichst flexiblen und zukunftsfähig gestalteten Topologie zur Verfügung.



2. Status Quo in der Automobilindustrie und Lehren für die Rüstungsindustrie

Mit diesem grundlegenden Perspektivenwechsel gehen erhebliche technologische, aber auch organisatorische Herausforderungen für die Rüstungsindustrie einher. Andere Branchen haben vergleichbare Transformationen bereits durchlaufen bzw. befinden sich mittendrin. Insbesondere die Automobilindustrie ist ein naheliegender Vergleich, weil sie unter dem Stichwort Software Defined Vehicle (SDV) ähnliche Ziele wie SDD im zivilen Bereich verfolgt: Fahrzeuge, deren Fähigkeiten durch Softwarepakete wachsen, deren Sicherheitslücken über Over-the-Air-Updates (OTA) geschlossen werden und deren Funktionsumfang in und mit einem Ökosystem von Partnern erweitert wird.

2.1 Status Quo in der Automobilindustrie

Die heutigen PKW und LKW basieren auf klassischen Fahrzeugarchitekturen, d.h. sie bestehen aus einer in der Vergangenheit ständig gewachsenen Anzahl an Steuergeräten, die über verschiedene Bussysteme wie CAN, CAN-FD, LIN, FlexRay und MOST miteinander verbunden sind und untereinander kommunizieren. Typischerweise werden Fahrzeugfunktionen in einem dedizierten Steuergerät abgebildet und Hardware und Software werden als Paket von einem Zulieferer bezogen.

Für einfache Fahrzeugfunktionen und eine begrenzte Zahl von Steuergeräten hat dieser Ansatz gut funktioniert. Bei komplexen Fahrzeugfunktionen, die auf eine Vielzahl an Sensoren und Aktuatoren zugreifen und eine hohe Rechenleistung mit Echtzeitfähigkeit erfordern, wie es z.B. für hochautomatisierte Fahrfunktionen der Fall ist, stößt dieser Ansatz aber schnell an seine Grenzen.

Die Antwort der Automobilindustrie auf diese Herausforderungen sind neue Fahrzeugarchitekturen, bei denen wenige Hochleistungsrechner (HPCs) im Zentrum stehen, die über leistungsstarke Bussysteme mit hoher Bandbreite verbunden sind. Die meisten traditionellen Hersteller verfolgen beim Übergang von der klassischen zur neuen Architektur einen evolutionären Ansatz, bei dem bestehende Steuergeräte schrittweise durch Zentralrechner ersetzt werden. Neue Marktteilnehmer wie Tesla haben auf der

„Grünen Wiese“ ohne historisch gewachsene Altlasten einen radikaleren Ansatz umgesetzt.

Bisher wurden seitens der OEM hauptsächlich zwei unterschiedliche Strategien zur Umsetzung gebracht. Eine Variante ist die Integration über Funktionsdomänen. Dabei werden einzelne HPCs jeweils einer bestimmten Funktionsgruppe wie Fahrerassistenz oder Infotainment gewidmet und die verbleibenden Steuergeräte (ECUs) entsprechend ihren Funktionen angebunden. Eine andere Strategie ist die räumliche Clusterung der Funktionen, bei der die verbleibenden ECUs auf Basis ihrer Lage im Fahrzeug mit zonalen HPCs verbunden werden. Ein gängiger, weil den typischen, domänenorientierten Organisationsstrukturen traditioneller OEMs näherer Ansatz, ist eine Kombination aus domänenorientierten Zentralrechnern und daran angebunden Zonencontrollern.

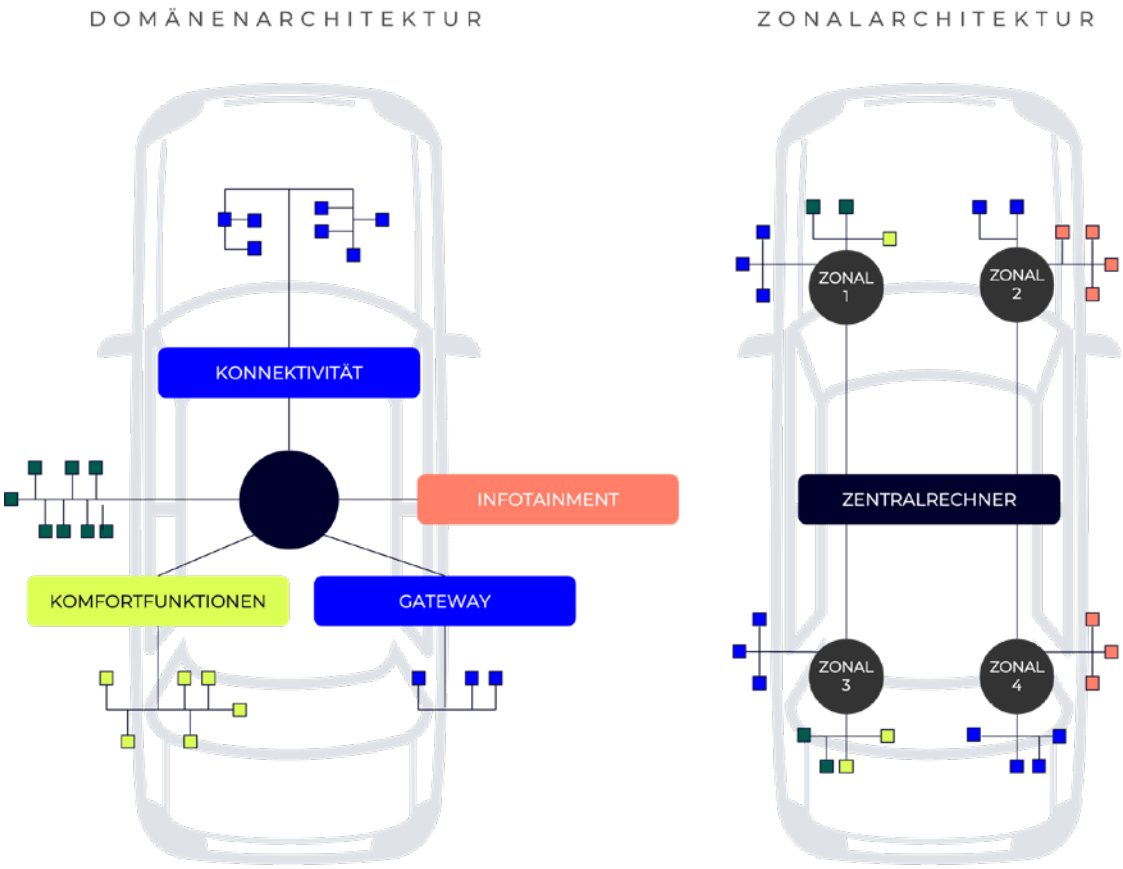


Abbildung 1: Architekturvergleich

Die externe Kommunikation der Fahrzeuge mit Backendsystemen oder z.B. Smartphones über LTE- und WLAN-Module oder USB-Schnittstellen ist in aktuellen (Premium-)Modellen bereits Standard. Über OTA-Updates werden in regelmäßigen Zyklen Fahrzeugfunktionen aktualisiert oder neue Funktionen (gegen entsprechendes Entgelt) zur Verfügung gestellt. Mit dem Konzept des Software Update Management Systems (SUMS) existieren zudem regulatorisch verankerte Rahmen in verschiedenen Märkten, z.B. UNECE R156 oder GB 44495-2024, die die kontinuierliche Aktualisierung von Fahrzeugsoftware zum festen Bestandteil des Produktlebenszyklus machen.

Gleichzeitig hat es die Automobilindustrie in der Vergangenheit vielfach versäumt, übergreifende Standards für die Software zu definieren, unternehmensintern wie auch branchenweit. Manche modernen Architekturen leiden bspw. darunter, dass zu ihrer Entstehung zwar die Hardwaretopologie State-of-the-Art war, die zugrundeliegenden Standards für Schnittstellen und Datenmodelle in der Software als Fundament aber gefehlt haben.

Branchenweit sind frühe Versuche gescheitert, gemeinsame Standards für diese Fundatemente des SDV zu definieren oder haben nur begrenzt Wirkung entfaltet. Statt ein gemeinsames, deutsches Autobetriebssystem zu entwickeln, wie es 2021 angeregt wurde, haben die Hersteller begonnen eigene, im Zweifel inkompatible, softwarezentrierte Architekturen zu entwickeln.

Dem technologischen Wandel folgt bei den etablierten Herstellern in der Automobilindustrie meist die Anpassung der Strukturen in den Unternehmen: Klassisch komponentenorientierte Organisationen, die Hardware und Software eng gekoppelt entwickeln, müssen sich zu funktions- und systemorientierten Organisationen transformieren. Der Fokus verlagert sich vom einzelnen Steuergerät hin zu Ende-zu-Ende-Funktionalitäten – eine Umstellung, die nicht ohne Brüche verläuft. Und meist wird der organisatorische und kulturelle Wandel erst nach dem technologischen eingeleitet und nicht parallel dazu oder gar vorgelagert. Dadurch behindern vielfach noch immer komponentenorientierte „Silo“-Denkweisen den erfolgreichen – auch technologischen – Übergang.

Diese Beispiele zeigen, dass die Automobilindustrie auf dem Weg hin zu softwarebasierten Fahrzeugarchitekturen bereits wertvolle Erfahrungen gesammelt hat und Technologien erfolgreich im Feld erprobt. Die Umstellung von verteilten Steuergerätelandschaften hin

zu zentralisierten HPCs ist dabei in vollem Gange und wird im iterativen Ansatz umgesetzt.

Allerdings zeigt sich auch, dass dieser Transformationsprozess nicht ohne Herausforderungen verläuft und weiterhin verlaufen wird. In der Vergangenheit lag der Fokus vieler Hersteller zu stark auf der Hardwareebene, während der Aufbau skalierbarer, wiederverwendbarer und gut wartbarer Softwarearchitekturen zu wenig berücksichtigt wurde.

Die Ausgangslagen, Herangehensweisen und Fortschritte der unterschiedlichen Hersteller sind sehr unterschiedlich. Viele etablierte OEMs haben nach wie vor große Probleme mit dem Wandel, sowohl technologisch als auch kulturell. Neuere Marktteilnehmer aus den USA oder die Wettbewerber aus China sind dank radikalerer Konzepte auf Basis von Elektrofahrzeugen dem Zielbild des Software Defined Vehicle oft wesentlich näher.

Die gewonnenen Erkenntnisse verdeutlichen, dass der Erfolg eines SDV maßgeblich von einer konsequent softwarezentrierten Denkweise, klaren Entwicklungsprozessen und einer stärkeren Trennung von Hardware- und Softwareverantwortlichkeiten abhängt. Die Branche hat also bereits wichtige erste Schritte gemacht, muss aber weiterhin lernen, Software als das Kernprodukt zu begreifen und dieses entsprechend organisatorisch wie technologisch umzusetzen.

2.2 Lessons Learned für den Verteidigungssektor

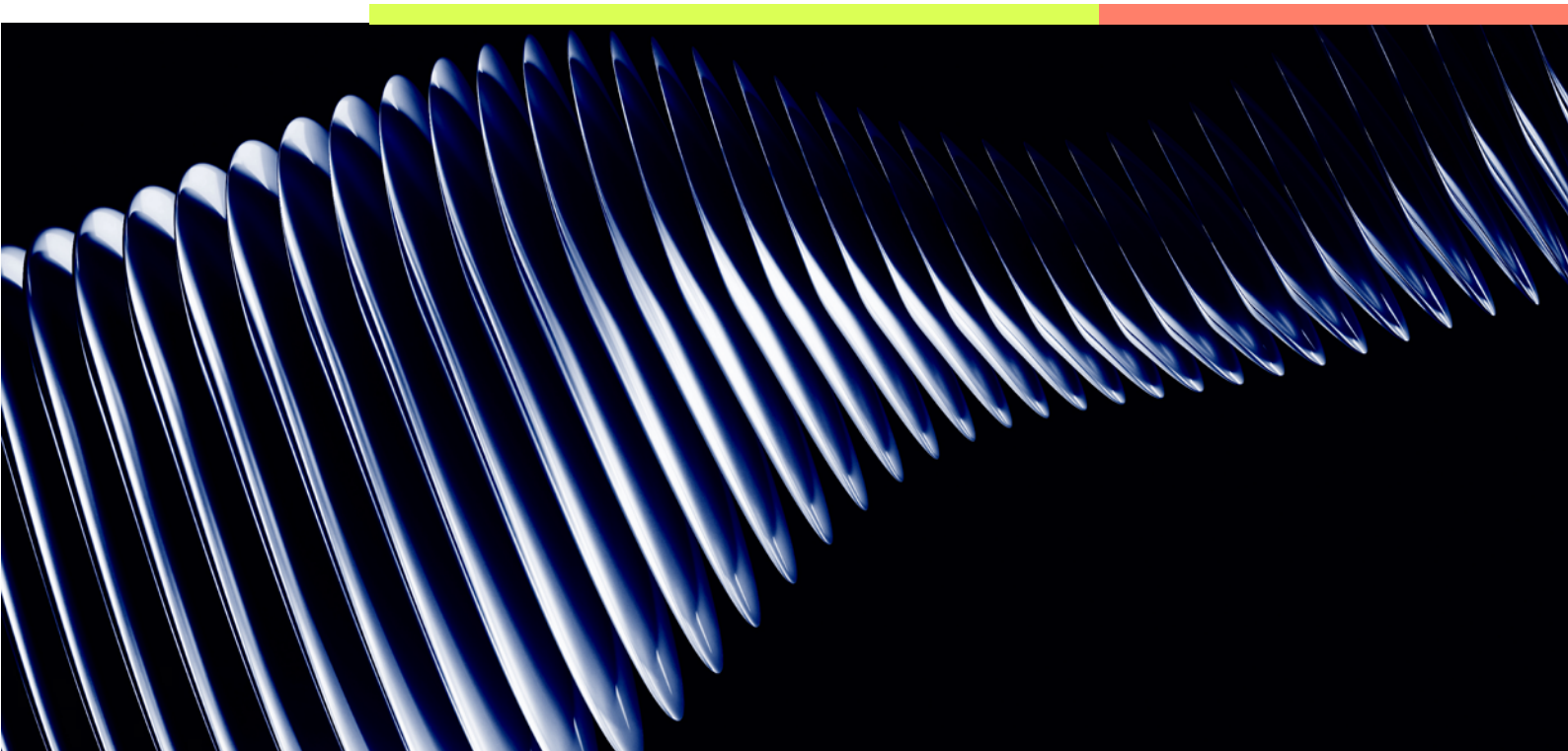
Die Transformation etablierter, komponentenorientierter Fahrzeugarchitekturen und Unternehmen zu softwarezentrierten Systemen und Organisationen ist anspruchsvoll, das zeigen die Erfahrungen aus der Automobilindustrie. Die Rüstungsindustrie kann und sollte aus diesem wertvollen Erfahrungsschatz lernen.

Die Rüstungsindustrie verfügt dank verschiedener Initiativen z.B. der DARPA, oder NATO bereits über Grundlagen für eine Standardisierung. Besonders hervorzuheben ist die NATO Generic Vehicle Architecture (NGVA), definiert in STANAG 4754. Sie bietet eine robuste, wenn auch nicht vollumfängliche, Ausgangsbasis, um das Konzept von Software Defined Defense zumindest im Bereich der Landfahrzeuge konkret umzusetzen. Dank ihrer langen Tradition im Systems Engineering ist die Rüstungsindustrie außerdem gewohnt, komplexe Systemverbünde ganzheitlich zu betrachten. Das erleichtert die Umsetzung

funktionsorientierter Organisationsmodelle.

Die Lehre aus SDV für SDD kann nicht „Copy & Paste“ sein, sondern „Prinzipien übernehmen, Kontexte respektieren, bekannte Fehler vermeiden“. Standardisierung und konsequente Modularisierung schaffen die Hebel für Geschwindigkeit und Qualität; OTA-Fähigkeit wird zum Motor kontinuierlicher Wirkung; funktionsorientierte Organisationen vermeiden Übergabereibung.

Wenn es der Rüstungsindustrie gelingt, die Erkenntnisse und Erfahrungen aus der Automobilindustrie mit dem SDV zu Nutzen und mit den eigenen Stärken zu verbinden, kann die erfolgreiche Transformation zum SDD auch in der durch die geopolitischen Umstände notwendig gewordenen Geschwindigkeit gelingen.



3. Software Defined Defense Vehicle (SDDV): Die konvergente Plattform aus SDD-Prinzipien und SDV-Methoden

Durch die enge, technologische Verwandtschaft und die durch die NGVA etablierte Referenzarchitektur, eigenen sich militärische Landfahrzeuge besonders, um die Anforderungen und Chancen von SDD und das Potential der Lessons Learned aus der Automobilindustrie und dem SDV greifbar zu machen. Die Summe dieser beiden Faktoren beschreibt im Kern ein SDDV, ein Software Defined Defense Vehicle.

Zu unterscheiden sind dabei zwei grundlegende Bereiche: Das Basisfahrzeug und die einsatzspezifischen Aufbauten, Missionscontainer und Sonderaufbauten, die sich auf standardisierte Grundfahrzeuge setzen lassen.

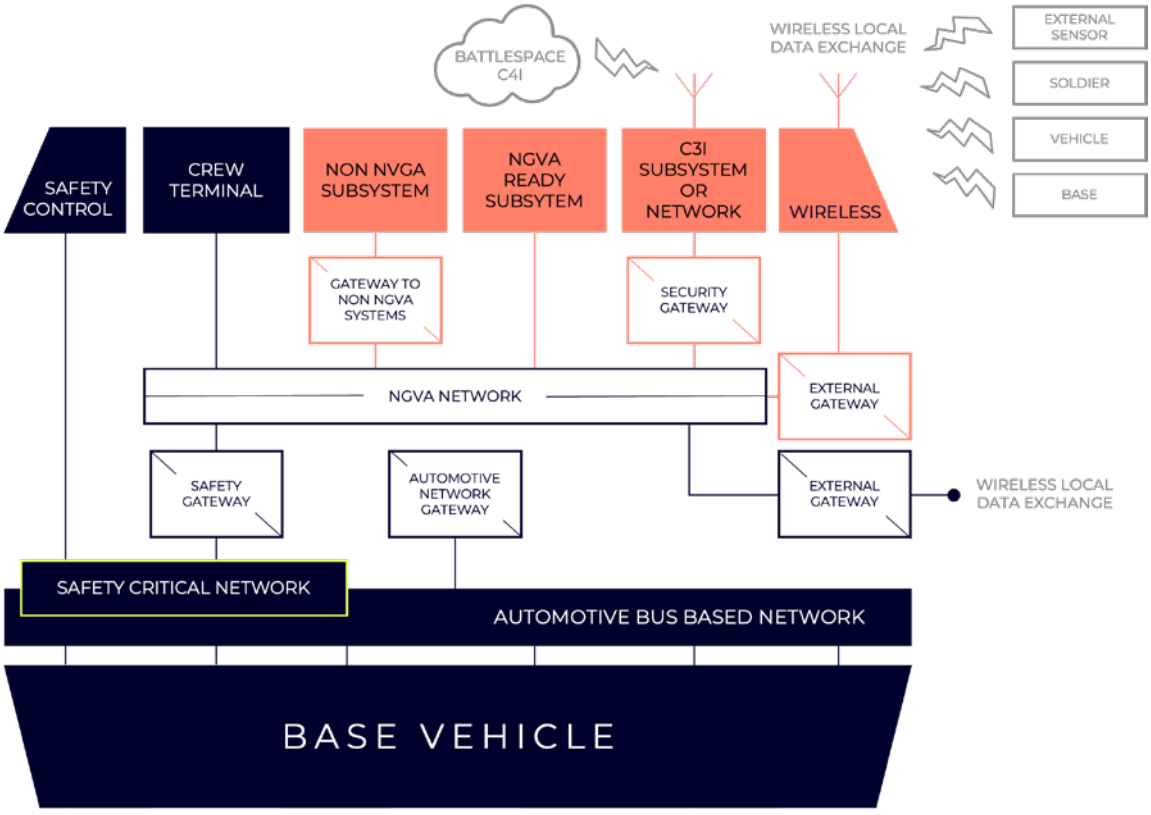


Abbildung 2: NGVA-Referenzarchitektur

Die Aufbauten sind im Grundsatz modular gestaltet und verfügen über Schnittstellen zum Basisfahrzug, die ihren Betrieb ermöglichen. Die NGVA gibt hier eine gute Grundlage, um diese modularen Aufbauten im Sinne eines SDDV zu gestalten. Das Basisfahrzeug selbst wird in der NGVA jedoch nur über Schnittstellen und ein „Automotive Bus Based Network“ abgebildet. Die nächste Entwicklungsstufe des SDDV liegt somit im Basisfahrzeug selbst, dessen Softwarearchitektur und darauf aufbauend die elektrische und elektronische Architektur so gestaltet sein müssen, dass Funktionsintegration, Updatefähigkeit und Cyber-Resilienz von Anfang an gewährleistet sind.

Die Basis dafür, und damit für ein SDDV, ist die Fähigkeit der Software- und Hardwarearchitektur, flexibel auf sich ändernde Anforderungen und damit Anpassungen der Software reagieren zu können. Das kann nur gewährleistet werden, wenn idealerweise jede Softwarekomponente flexibel auf jede Hardwarekomponente verschoben werden kann. Die Softwarearchitektur muss ins Zentrum rücken und die Hardwarearchitektur auf dieser Basis gestaltet werden.

Entscheidend für die softwarezentriere Architektur des SDDV sind zwei wesentliche Festlegungen:

1. Die Definition von Abstraktionsebenen
2. Der Aufbau eines Datenmodells

3.1 SDDV-Abstraktionsebenen

Die saubere Definition von Abstraktionsebenen legt die Fähigkeiten und Ressourcen fest, die die Architektur des SDDV der Software bereitstellt. Typische Abstraktionsebenen sind z.B. Rechenleistung sowie Sensoren und Aktuatoren, aber auch Fahrzeugsignale, Speicherschutz und weitere. Für jede dieser Ebenen werden Gruppen gebildet, die sich in ihren Eigenschaften und Fähigkeiten unterscheiden. Betrachten wir zwei Beispiele im Detail:

1. Rechenleistung
2. Sensoren und Aktuatoren

3.1.1 Abstraktionsebene Rechenleistung

Die notwendige Rechenleistung des SDDV wird über verschiedene Klassen definiert. Dabei gilt die Maßgabe, so wenige Klassen wie möglich aber so viele wie nötig zu definieren, da für jede Klasse einheitliche Entwicklungs- und Laufzeitumgebungen etabliert werden müssen, um die Plug-and-Play-Fähigkeit der Softwarekomponenten auf verschiedenen Hardwarekomponenten sicherzustellen. Für das SDDV nehmen wir beispielhaft drei Klassen an: Rechenleistung, Echtzeitfähigkeit und funktionale Sicherheit. Geeignete Entwicklungs- und Laufzeitumgebungen für diese Klassen sind beispielsweise QNX, AUTOSAR Adaptive oder Linux.

3.1.2 Abstraktionsebene Sensoren und Aktuatoren

Bei der standardisierten Schnittstelle der Software zu Sensoren und Aktuatoren sind grundsätzlich zwei Ansätze denkbar.

Beim Low-Level Ansatz bietet die SDDV-Architektur der Software z.B. die direkte Steuerung der Aktuatoren über Pulsweitenmodulation (PWM) an. Dieser Ansatz ist besonders geeignet, wenn Hardware und Software vom gleichen Lieferanten bezogen werden, da der Lieferant die Hardware im Detail kennt und über die direkte Schnittstelle optimal ansteuern kann.

Beim High-Level Ansatz stellt die Architektur der Software höher aggregierte Schnittstellen zur Verfügung wie gewünschte Motordrehzahl oder benötigtes Drehmoment. Die zugrunde liegenden Berechnungen für die tatsächliche Ansteuerung z.B. der Hardware übernimmt die Schnittstelle. Dieser Ansatz ist besonders geeignet, wenn Hardware und Software getrennt bezogen werden.

Für das SDDV nehmen wir beispielhaft aufgrund geringerer Stückzahlen und niedrigerer Preissensibilität des Rüstungssektors im Vergleich zur Automobilindustrie einen kombinierten Ansatz an. Dieser bedeutet zwar höhere Kosten durch z.B. mehr Overhead und notwendige Rechenleistung, da verschiedene Ebenen bedient werden müssen, gewährleistet aber maximale Flexibilität.

3.2 SDDV-Datenmodell

Neben den Abstraktionsebenen ist der Aufbau eines einheitlichen Datenmodells die zweite wesentliche Grundlage für die softwarezentrierte Architektur des SDDV. Entscheidend hierfür ist die Definition von technologieunabhängigen Datenpunkten (z.B. Geschwindigkeit, Drehmoment, Temperatur, Höhe) und der Aggregation dieser Datenpunkte in sinnvolle Gruppen (z.B. Fahrzeugzustand, Umgebungsbedingungen). Die SDDV-Architektur stellt den Softwarekomponenten somit keine Einzelsignale zur Verfügung, sondern die entsprechenden Gruppen. Über die Kommunikationsmatrix wird sichergestellt, dass die von einer Applikation benötigten Signale entsprechend geroutet werden.

3.3 Technologiefestlegung

Der nächste Schritt auf dem Weg zum SDDV ist die Umsetzung des technologieunabhängigen Datenmodells und der Abstraktionsebenen in reale Schnittstellen und damit die Festlegung, welche Technologien für die Codegenerierung in der Architektur genutzt werden. Entscheidend für die Auswahl der Technologien sind insbesondere ein möglichst hoher Grad der Automatisierung und die vorhandenen Kompetenzen in der Entwicklungsorganisation. Aus der Vielzahl der Technologien, die auf dem Markt verfügbar sind, wie DDS, SOME/IP oder Linux D-Bus, wird ein geeignetes Set von wenigen Technologien ausgewählt, das standardisiert über alle Entwicklungsbereiche genutzt wird.

Die Technologiefestlegung und die mit ihr verbundene geringe Diversität der genutzten Toolchains bringen viele Vorteile mit sich. Ressourcenverschwendung in der Hardware z.B. durch unnötigen Speicherbedarf konkurrierender Libraries wird vermieden, Sicherheitslücken werden reduziert und die Flexibilität der Entwicklungsteams wird erhöht, wenn nur wenige, aber dafür von allen beherrschte Technologien verwendet werden.

3.4 SDDV-Softwarearchitektur

Die definierten Abstraktionsebenen, Datenmodelle und Technologien sind die Grundlage für die Softwarearchitektur des SDDV. Auf dieser Basis werden für jede Funktion die Wirkketten z.B. im klassischen IPO-Modell beschrieben, um die Abhängigkeiten und Datenflüsse zu modellieren. Aus diesem Modell kann mit entsprechendem Tooling automatisiert die Kommunikationsmatrix erstellt werden. Wenn die Signalflüsse mit entsprechenden Annotationen für Anforderungen an z.B. Latenz oder Bandbreite versehen werden, können außerdem direkte Anforderungen an die Hardwaretopologie abgeleitet werden.

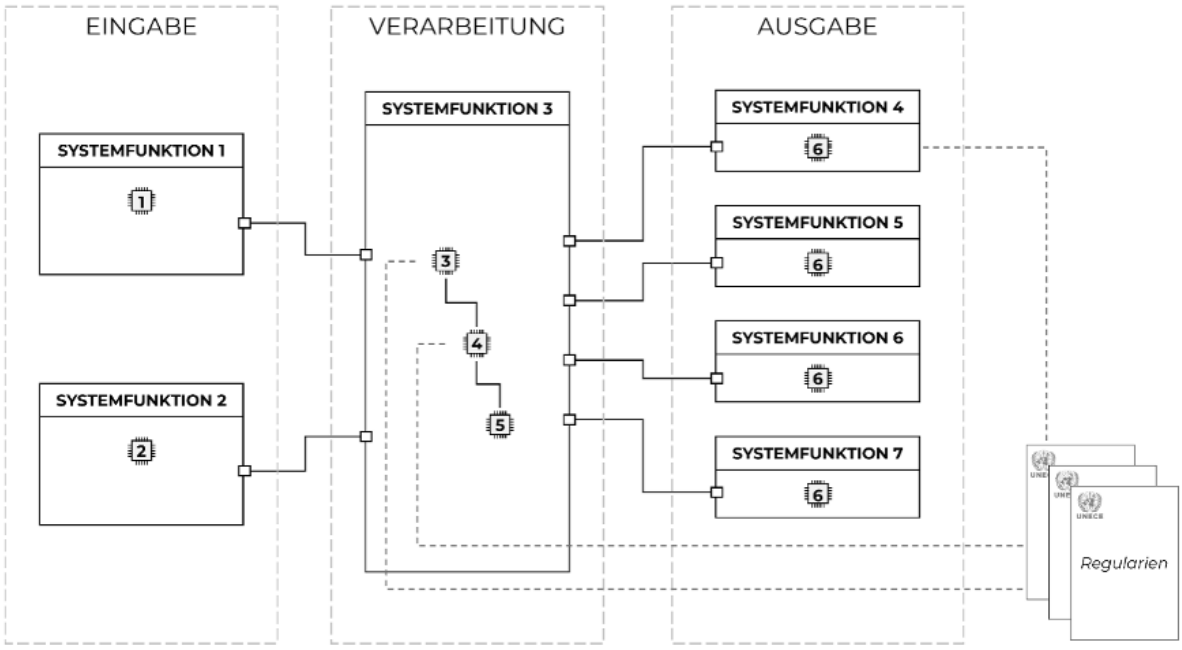


Abbildung 3: Schematische Darstellung einer Wirkkette

3.5 Cybersecurity

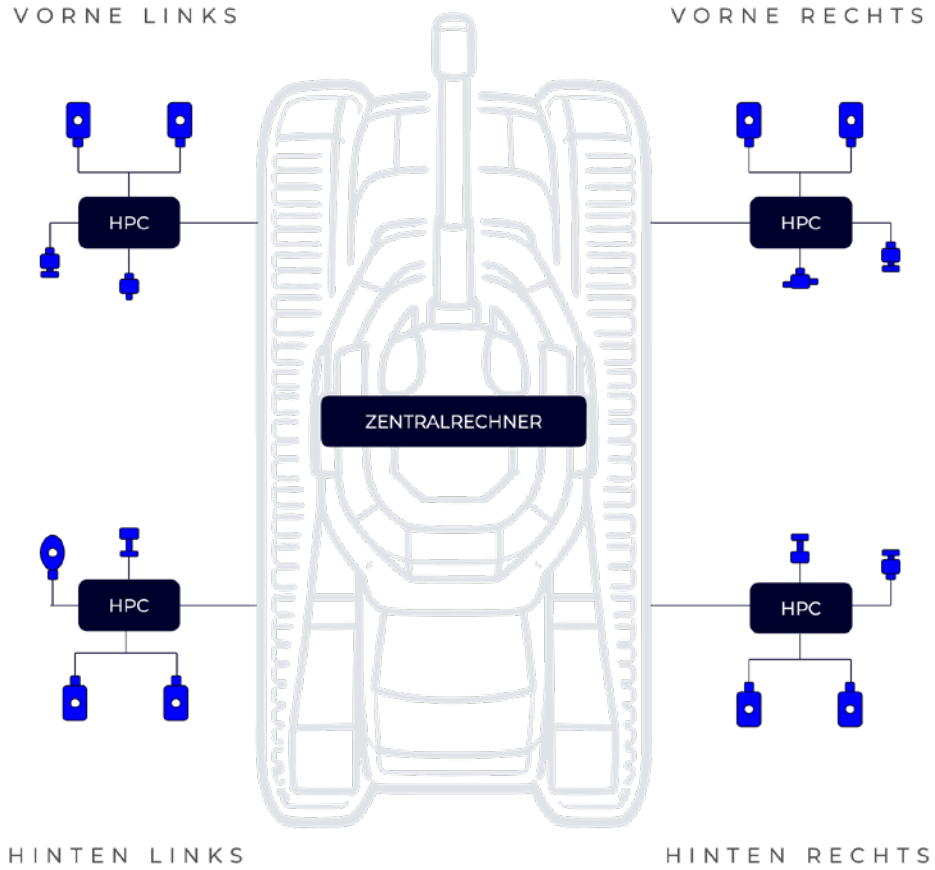
Das Thema Cybersecurity ist in der Defensebranche von größter Bedeutung und daher gesondert zu betrachten. Sicherheit wird nun nicht mehr primär physisch, sondern vielmehr durch mehrschichtige Mechanismen auf Architekturebene erzielt. Erforderlich sind u.a. Zero-Trust-Prinzipien, harte Mandantentrennung (SoC-/Hypervisor-/Container-Isolation) sowie konsequente Least-Privilege- und Policy Enforcement Mechanismen für jede Softwarekomponente. Dazu kommen kryptographisch abgesicherte Boot- und Update-Ketten (Secure Boot, Measured Boot, Remote Attestation), eine kontinuierlich gepflegte SBOM inklusive automatisiertem Vulnerability- und Patch-Management sowie ein Governance-Modell, das Update Frequenzen, Rollout-Risiken und Zulassungsabhängigkeiten beherrscht. Auf Laufzeitebene sind Intrusion/Anomaly-Detection-Systeme erforderlich, die sowohl signalbasierte als auch modellgestützte Verfahren zur Angriffserkennung nutzen und mit Reaktionslogiken für Containment, Rate Limiting oder isolierbaren Micro-Recovery-Zonen gekoppelt sind.

Darüber hinaus gewinnen deterministische Safety-/Security-Partitionen, manipulationsresistente Kommunikationspfade (MACsec, TLS mit Mutual Authentication), robuste Schlüsselverwaltung (HSM-gestützt) sowie Integritäts- und Herkunftsnachweise für alle Datenströme an Bedeutung. Für einsatzkritische Funktionen müssen abgestufte Degradations-, Redundanz- und Fallback Konzepte existieren, die auch bei teilweiser Kompromittierung oder gestörter Konnektivität, etwa über lokale Entscheidungslogik, abgesicherte Notbetriebsmodi oder fail operational-Architekturen, weiterarbeiten.

3.6. SDDV-Hardwaretopologie

Parallel zur Softwarearchitektur wird die Topologie der Hardware entwickelt. Die softwarezentrierte Herangehensweise ermöglicht grundsätzliche flexible Topologien für z.B. unterschiedliche Modellreihen innerhalb der gleichen Architektur.

Der zentrale Grundgedanke der Hardwaretopologie im SDDV ist die Zonenbildung mit HPCs (Zonencontroller), welche die Ein- und Ausgangssignale der umliegenden Sensoren und Aktoren bündeln, (z.B. vorne links, vorne rechts, hinten links, hinten rechts) und weiterverarbeiten. Damit werden Kabellängen, Gewicht und Kosten reduziert und Durchbrüche der Panzerungen minimiert. Gleichzeitig wird durch verminderte Komplexität und Störquellen die Robustheit erhöht.



AAbbildung 4: Schematische Darstellung Hardwaretopologie im SDDV

Die standardisierten Technologien ermöglichen eine flexible Verteilung der Softwarekomponenten auf den Zonencontrollern. Dadurch können einerseits Funktionen auf entsprechende, räumlich naheliegende Hardware gemappt werden. Andererseits gewährleistet diese Flexibilität eine gewisse Redundanz, da kritische Funktionen bei Ausfall eines Controllers durch andere übernommen werden können. Um die Notwendigkeit weiterer Redundanzen zu ermitteln, z.B. durch verteilte Zentralrechner im vorderen und hinteren Bereich des Fahrzeugs, bietet sich ein konsequentes Systems Engineering an, um die unvermeidlichen Zielkonflikte (z.B. räumliche Verteilung vs. minimierte Kabellängen und Durchbrüche) auf Basis der zu erwartenden Einsatzzwecke abzuwägen.

Für die Vernetzung innerhalb der Topologie empfiehlt die NGVA getrennte Bussysteme für Kommunikation mit hohen Anforderungen an Latenz und Bandbreite, da sich diese beiden Anforderungen klassischerweise gegenseitig schaden. Standard für hohe Anforderungen an die Latenz und Echtzeitkommunikation sind CAN-Busse, für hohe Bandbreite Ethernet. Allerdings ist die Leistungsfähigkeit der Bussysteme auch von den Betriebssystemen der verbundenen Controller abhängig und moderne Ethernet-Busse können vielfach auch bei der Latenz mit CAN-Bussen konkurrieren. Für das SDDV bietet sich daher ein Kompromiss an, bei dem das Backbone grundsätzlich ein Ethernet-Bus ist, der an kritischen Punkten mit höchsten Echtzeit-Anforderungen durch einen zusätzlichen CAN-Bus ergänzt wird.

3.7 Lieferketten

Das SDDV wirkt sich auch direkt auf die unterliegenden Lieferketten aus. Durch die Standardisierung auf wenige Middleware- und Kommunikationstechnologien müssen Zulieferer ihre Komponenten konsequent an ein einheitliches Architektur- und Sicherheitsmodell anpassen. Das reduziert die Variantenvielfalt und erleichtert den Austausch von Komponenten, erhöht aber zugleich den Anpassungsdruck auf Lieferanten, welche die entsprechenden Kompetenzen und Toolchains erst aufbauen müssen.

Zugleich verlagern sich die bisherigen Abhängigkeiten. Weniger proprietäre Hardware, dafür mehr software- und datengetriebene Integrationspunkte, die neue Risiken für Vendor-Lock-in oder technologische Engpässe schaffen können. OEMs müssen daher technologische Reifegrade, Security-Compliance und Updatefähigkeit stärker als Bewertungs- und Steuerungsgrößen in der Lieferantenlandschaft verankern.

4. Betrachtungen zur Nutzung von Backendfunktionen

Die Integration von Backendfunktionen in Militärfahrzeugen bietet erhebliche Vorteile in Bezug auf Konnektivität, Effizienz und operative Transparenz. Über sichere Cloud- oder lokale Backends können Fahrzeuge Echtzeitdaten zu Fahrzeugzustand, Missionsstatus, Logistikanforderungen und Umgebungsbedingungen austauschen. Dies ermöglicht Potenziale bzgl. präventiver Wartung, schnellere Softwareupdates, verbesserte Lageerkennung und koordiniertem Flottenmanagement.

Eine zentrale Verwaltung wie ein Kommandostand kann die Fahrzeugleistung überwachen, Ausfälle vorhersagen und die Missionsplanung auf der Grundlage präziser, kontinuierlich aktualisierter Daten optimieren. Solche Fähigkeiten entsprechen den modernen Verteidigungstrends hin zu softwaredefinierten und netzwerkzentrierten Operationen, bei denen die digitale Infrastruktur die Reaktionsfähigkeit verbessert und Ausfallzeiten reduziert.

Die Gewährung des Backend-Zugriffs auf Militärgerät bringt jedoch auch erhebliche Herausforderungen und Risiken mit sich. Das Hauptproblem ist die Cybersecurity: Die Backend-Konnektivität vergrößert die Angriffsfläche und setzt Fahrzeuge potenziell Datenabgriffen, Fernmanipulationen oder feindlichen Eingriffen aus.

Die Gewährleistung einer sicheren Authentifizierung, Verschlüsselung und Isolierung kritischer Systeme ist dabei unerlässlich. Die Abhängigkeit von der Netzwerkverfügbarkeit und der Backend-Infrastruktur kann auch die Widerstandsfähigkeit in Umgebungen mit eingeschränkter bzw. nicht vorhandener Konnektivität beeinträchtigen, in denen die Kommunikationsverbindungen aus verschiedensten Gründen unterbrochen sind. Darüber hinaus werfen insbesondere Datenhoheit und Zugriffskontrolle strategische und ethische Fragen auf, wenn Teile der Backend-Infrastruktur von Drittanbietern verwaltet werden.

5. Fazit und Ausblick

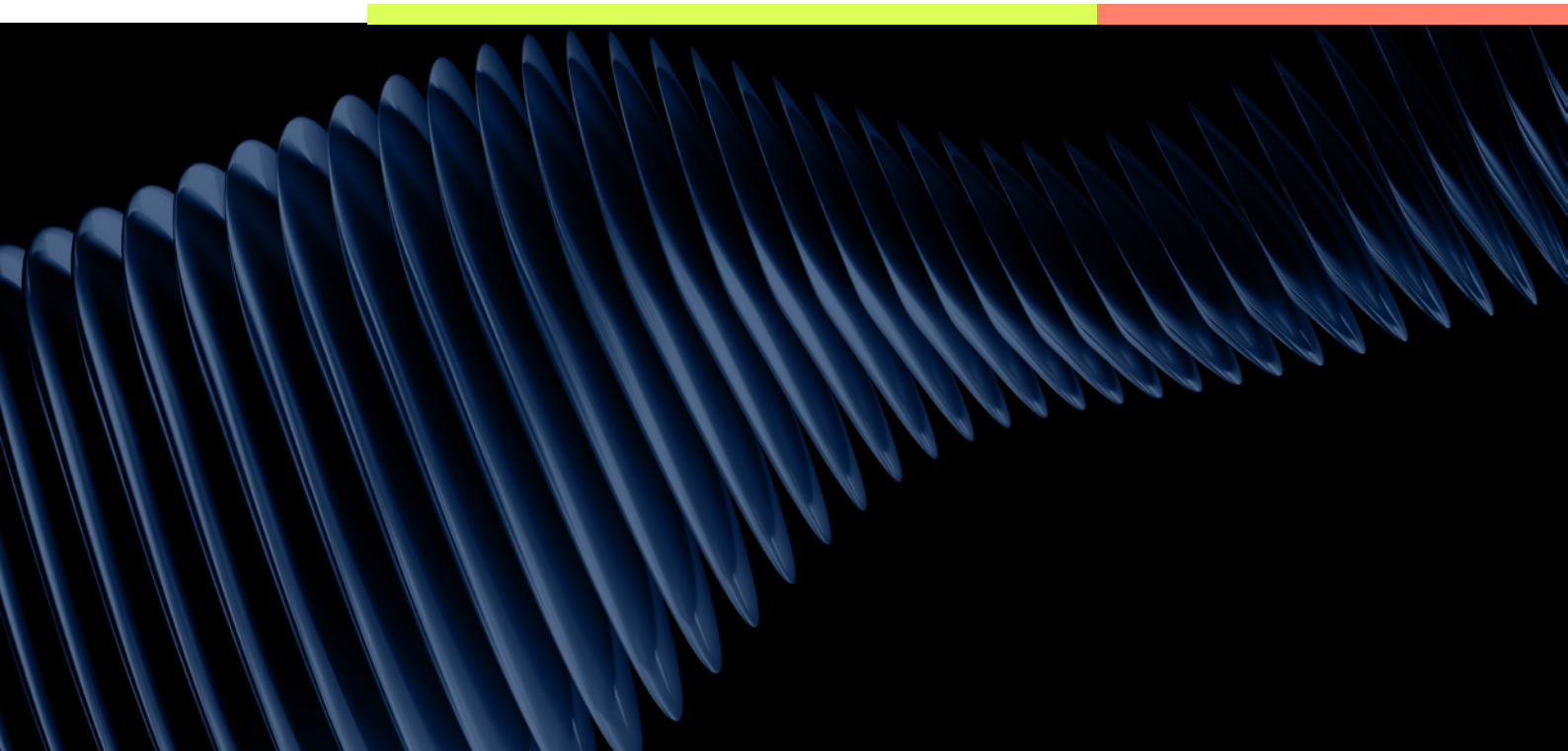
SDD steht für einen echten Paradigmenwechsel: Wirkung, Anpassbarkeit und Interoperabilität werden primär durch Software erzielt und über den gesamten Lebenszyklus weiterentwickelt. Aus den Erfahrungen der zivilen Automobilindustrie mit dem SDV lässt sich für Landplattformen schlüssig ein SDDV mit folgenden Kernpunkten ableiten: Zentrale Rechenplattformen, klare Abstraktionsebenen, einheitliche Datenmodelle und die OTA-Fähigkeit erhöhen Tempo, Leistungsfähigkeit und Nutzungsdauer. Zugleich zeigt der SDV-Weg auch mögliche Fallstricke auf, etwa ein zu starker Hardwarefokus oder mangelnde Standardisierung. Die Rüstungsindustrie kann hiervon gezielt lernen: Software zuerst und mit Priorität denken, Standards konsequent anwenden und ausbauen sowie die Modularität von Lieferartefakten erzwingen.

Mit der reduzierten physischen Trennung im Fahrzeug (zonenbasierte HPCs statt vieler isolierter ECUs) verschiebt sich das Cybersecurity-Zielbild: Sicherheit entsteht nicht mehr primär durch Verkabelungstopologien, sondern durch architekturverankerte Schutzmechanismen. Kurz: Security-by-Design und Resilienz ersetzen die allzu oft vorhandene Illusion physischer Sicherheit.

Das SDDV hat zudem tiefgreifende organisatorische und kommerzielle Implikationen: Hersteller entwickeln sich zu Systemintegratoren, Funktionen werden Ende-zu-Ende verantwortet. Governance, Zertifizierung und Abnahme müssen von einer punktuellen Freigabe auf kontinuierlich belegte Betriebssicherheit umgestellt werden. In der Lieferkette gewinnen klar vorgegebene Schnittstellen, wiederverwendbare Plattformen und IP-Regelungen an Gewicht. Geschäfts- und Vertragsmodelle verschieben sich zu leistungs- und wirkungsbasierten Verträgen, Lifecycle-Services und softwarebasierten Upgrades statt Einmalbeschaffung. Offene Standards und korrekt definierte APIs ermöglichen Wettbewerb auf Funktions-Ebene und senken den Vendor Lock in Effekt unter Bewahrung der staatlichen Souveränität.

Die Kernbotschaft lautet daher: Die Rüstungsindustrie kann die Erkenntnisse der zivilen Automobilindustrie bzgl. des SDV nutzen, bekannte Fehler vermeiden und durch konsequent softwarezentriertes Denken, harte Cyber-Resilienz und passende Organisations- und Vertragsformen den Wandel hin zu einem echten SDDV beschleunigen.

Wer Architektur, Prozesse und Geschäftsmodelle jetzt auf „Software als Kernprodukt“ ausrichtet, gewinnt Tempo, Wirkung und Überlegenheit, ohne dabei Sicherheit zu opfern.



Ihr direkter Ansprechpartner rund um Defence:

Bernd Schäfer
Managing Director | Defence

bernd.schaefer@p3-group.com

Address

P3 group GmbH
Heilbronner Straße 86
70191 Stuttgart
Germany

Contact

+49 711 252 749-0
mail@p3-group.com
www.p3-group.com