E-Book

Kontron AIS GmbH

Sicheres und flexibles IoT-Device-Management:

Build or Buy?

Vanessa Kluge, Produktmanagerin bei Kontron AIS GmbH

Stehen Sie vor der Entscheidung: Build or Buy?

Dieses E-Book ist Ihr umfassender Leitfaden, um fundierte Entscheidungen zwischen Eigenentwicklung und dem Kauf einer fertigen IoT-Device-Management-Lösung zu treffen. Beleuchtet werden alle relevanten Aspekte: Wie sichern Sie maximale Sicherheit und Compliance für kritische Infrastrukturen? Welche Lösung passt optimal zu Ihren Skalierungsplänen und technischen Anforderungen? Mit praxisnahen Checklisten und realen Beispielen hilft Ihnen dieses E-Book, eine informierte und strategisch kluge Entscheidung zu treffen.

Januar 2025

E-Book

Inhaltsverzeichnis

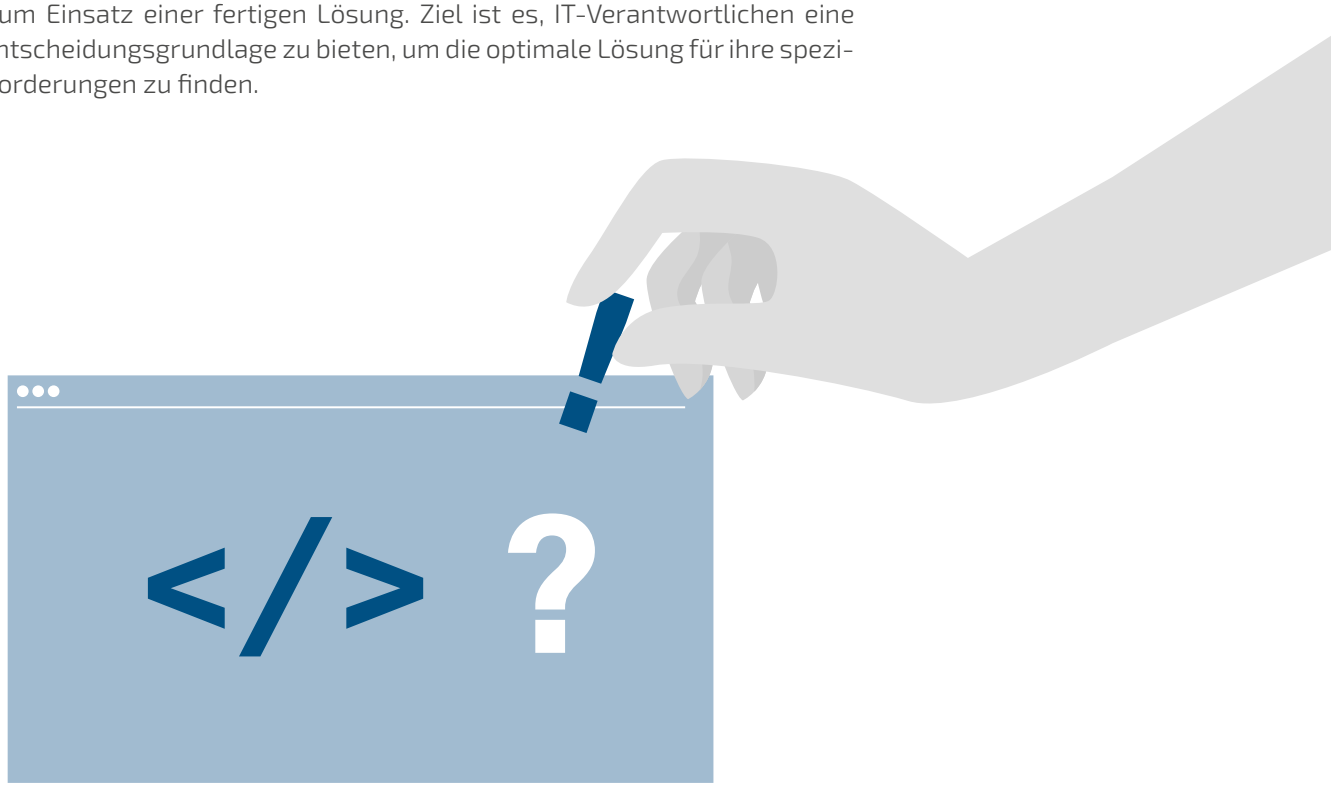
Sicheres und flexibles IoT-Device-Management: Build or Buy?

Das Internet of Things (IoT) entwickelt sich in rasantem Tempo – und mit ihm die Herausforderungen, vor denen IT-Teams und Softwareentwickler stehen. Eine der größten Aufgaben? Das effiziente Management von Edge Devices, also Geräten, die oft unter schwierigen Bedingungen, in großer Anzahl oder an entlegenen Orten eingesetzt werden. Diese Geräte sind das Rückgrat moderner IoT-Architekturen, doch ihre Verwaltung ist eine echte Gratwanderung zwischen Sicherheit, Skalierbarkeit und Effizienz.

Ein zentrales Problem ist dabei der sichere und zuverlässige Fernzugriff auf diese Geräte. Da viele IoT-Geräte der Zugang zu Maschinen- und Kundennetzwerken sind, müssen die Lösungen zu deren Verwaltung strenge Compliance-Anforderungen erfüllen. Die Sicherheit der Daten und die Integrität der Systeme haben dabei oberste Priorität, was die Entwicklung und Umsetzung entsprechender Maßnahmen zu einer anspruchsvollen Aufgabe macht.

Die Frage ist also: Entwickeln Sie eine eigene Lösung, die genau auf Ihre Bedürfnisse zugeschnitten ist? Oder setzen Sie auf eine bewährte Standardlösung vom Markt? Beide Ansätze haben Vor- und Nachteile, die sorgfältig abgewogen werden müssen. Faktoren wie Kosten, Zeitaufwand, Ressourcenverfügbarkeit und Skalierbarkeit spielen hier eine entscheidende Rolle.

Dieses E-Book beleuchtet die Herausforderungen beim Management von Edge Devices und untersucht detailliert die Vor- und Nachteile einer Eigenentwicklung im Vergleich zum Einsatz einer fertigen Lösung. Ziel ist es, IT-Verantwortlichen eine fundierte Entscheidungsgrundlage zu bieten, um die optimale Lösung für ihre spezifischen Anforderungen zu finden.



1 Welche Lösung passt zu Ihrem Unternehmen?

Selbst entwickeln oder fertig kaufen? Eine pauschale Antwort gibt es darauf nicht. Jede Option bringt ihre eigenen Vor- und Nachteile mit sich – und welche die richtige ist, hängt ganz von Ihren individuellen Anforderungen, Ressourcen und Prioritäten ab. Um Ihnen zunächst einen klaren Überblick zu verschaffen, folgen hier die wichtigsten Argumente für beide Ansätze.

1.1 Build (In-House-Entwicklung)

Vorteile

Anpassung
Die Lösung kann genau auf die speziellen Bedürfnisse des Unternehmens zugeschnitten werden, was mehr Flexibilität bei Funktionen und Funktionalität ermöglicht. Die Integration in bestehende Systeme klappt reibungslos, sodass die Software perfekt an die IT-Umgebung des Unternehmens angepasst werden kann.

Kontrolle
Unternehmen behalten die Kontrolle über den gesamten Entwicklungsprozess, Sicherheitsprotokolle und Updates. Auch das geistige Eigentum (IP) bleibt im Haus – ein potenzieller Wettbewerbsvorteil.

Skalierbarkeit
Die Lösung kann flexibel mit den Wachstumsplänen des Unternehmens mitwachsen, ohne durch Einschränkungen eines Drittanbieters limitiert zu sein.

Sicherheit
Das Unternehmen kann eigene Sicherheitsstandards und -protokolle einbauen, damit sensible Daten sicher und gemäß den internen Richtlinien verwaltet werden.

Langfristige Kosteneinsparungen möglich
Eine eigene Lösung kann sich bei langfristigem Einsatz über mehrere Projekte hinweg als kostengünstiger erweisen.

Nachteile

Hohe Anfangskosten
Die Entwicklungskosten können erheblich sein, einschließlich der Einstellung spezialisierter Entwickler*innen, des Kaufs von Entwicklungstools und der Zuweisung von Ressourcen.

Hohe Anforderungen an Fachkenntnisse
Die Entwicklung einer komplexen IoT-Flottenmanagementlösung erfordert ein Team mit spezifischen technischen Fähigkeiten, das unter Umständen schwer zu rekrutieren oder teuer zu halten sind. Die Sicherheitsanforderungen entwickeln sich ständig weiter und müssen bereits in der Entwicklungsphase berücksichtigt werden.

Zeitaufwand
Eine komplett neue Lösung von Grund auf zu entwickeln, kann ziemlich lange dauern. Dadurch verzögert sich die Implementierung und Sie verlieren möglicherweise Ihren Wettbewerbsvorteil.

Wartungsaufwand
Die laufende Wartung, Updates und Problemlösungen müssen intern bewältigt werden, was dedizierte Teams und Ressourcen erfordert.

Ausfallrisiko
Jede Entwicklung birgt das Risiko, dass sie nicht funktioniert. Das kann zu verschwendeten Ressourcen und Kosten führen. Gerade bei Neuentwicklungen kann es besonders problematisch sein, wenn Unternehmen nicht mit agiler Softwareentwicklung vertraut sind.

1.2 Buy (Drittanbieter-Lösung)

Vorteile

Schnelle Implementierung
Drittanbieter-Lösungen sind meist sofort einsatzbereit, was eine schnelle Implementierung ermöglicht.

Niedrigere Anfangskosten
Die Kosten sind initial oft geringer, da Entwicklungs- und Testphasen entfallen.

Bewährte Technologie
Marktlösungen sind getestet, etabliert und reduzieren das Risiko technischer Probleme.

Regelmäßige Updates
Anbieter stellen regelmäßig Updates, Sicherheitspatches und neue Funktionen zur Verfügung, so dass die Lösung immer auf dem neuesten Stand ist.

Support und Service
Umfangreiche Support-Dienste stellen sicher, dass Probleme schnell von erfahrenen Fachleuten gelöst werden und ein geführtes Onboarding stattfindet.

Nachteile

Keine 100 % ige Anpassung
Standardlösungen erfüllen möglicherweise nicht alle unternehmensspezifischen Anforderungen, was zu Kompromissen bei Funktionalität oder Leistung führen kann.

Abhängigkeit vom Anbieter
Das Unternehmen wird vom Anbieter für Updates, Support und Weiterentwicklungen abhängig. Das kann riskant sein, falls der Anbieter die Richtung ändert oder das Produkt einstellt.

Potenzial für höhere langfristige Kosten
Abonnementgebühren, Lizenzkosten und Kosten für zusätzliche Benutzer*innen oder Funktionen können sich im Laufe der Zeit summieren und die Lösung potenziell teurer machen.

Herausforderungen bei der Integration
Die Integration einer Lösung eines Drittanbieters in bestehende Systeme und Arbeitsabläufe kann schwierig sein und zusätzliche Ressourcen oder Middleware erfordern.

Unternehmen mit spezifischen Anforderungen, umfassender finanzieller Mittel sowie hoher technischer Kompetenz tendieren häufig zur Eigenentwicklung. Wenn jedoch ein schneller Start und grundlegende Funktionalität im Fokus stehen, ist eine bewährte Standardlösung meist die bessere Option.

Auf einen Blick zusammengefasst:

Build (In-House-Entwicklung)		Buy (Drittanbieter-Lösung)
Hoch	Anpassungsfähigkeit	Gering bis mittel
Vollständig	Kontrolle	Eingeschränkt
Maßgeschneidert	Skalierbarkeit	Standardisiert
Interne Standards	Sicherheit	Abhängig vom Anbieter
Hoch	Anfangskosten	Niedrig
Lang	Implementierungszeit	Kurz
Intern zu bewältigen	Wartungsaufwand	Extern unterstützt
Entwicklungsausfall	Risiko	Abhängig vom Anbieter

2 Aufwandsschätzung für eine eigene Lösung

Die Entwicklung einer eigenen IoT-Device-Management-Lösung für Edge Devices ist eine strategische Entscheidung, die mit signifikanten Investitionen in Zeit, Geld und Personal verbunden ist. Um fundiert entscheiden zu können, ist es wichtig, die potenziellen Kosten und Ressourcenaufwände realistisch einzuschätzen. Deshalb wird im Folgenden eine Übersicht der wichtigsten Kostenfaktoren, basierend auf typischen Lohn- und Kostenstrukturen in Europa für eine Flottengröße von 100 Geräten (erstes Jahr) und einem Anstieg um jeweils 100 weitere Geräte pro Jahr. gegeben.

2.1 Entwicklungskosten

Die Entwicklungskosten umfassen die Ausgaben für die Softwareentwicklung, einschließlich der Einstellung von Entwickler*innen, Projektmanagement und damit verbundenen Ausgaben.

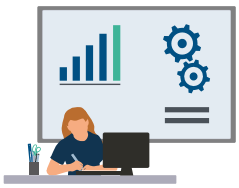
Teamzusammensetzung:

Projektmanager*in:

- 1 FTE (Vollzeitäquivalent)
- 80.000 - 100.000 € geschätztes Jahresgehalt

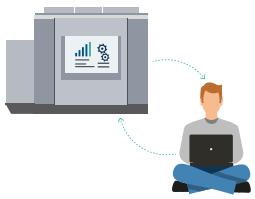
Softwareentwickler*innen:

- 3 - 5 FTEs (Front-End-, Back-End- und Edge-Computing-Experten)
- 60.000 - 90.000 € geschätztes Jahresgehalt pro Entwickler*in



UI-/UX-Designer*in:

- 1 FTE
- 50.000 - 70.000 € geschätztes Jahresgehalt



QA-/Testingenieur*innen:

- 1 - 2 FTEs
- 50.000 - 70.000 € geschätztes Jahresgehalt pro QA-Ingenieur*in

Sicherheitsspezialist*in:

- 1 FTE
- 70.000 - 100.000 € geschätztes Jahresgehalt

Zeitabschätzung:

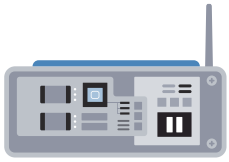
- 9 - 12 Monate (anfängliche Entwicklungszeit)

Gesamte Entwicklungskosten:

- 580.000 - 860.000 € (erstes Jahr)
- 120.000 - 180.000 € / Jahr (laufende Entwicklungskosten)

2.2 Infrastrukturkosten

Infrastrukturkosten umfassen die Kosten für Hardware, Cloud-Dienste und andere notwendige Infrastrukturen.



Hardware:

- 70.000 - 120.000 € für Edge Devices, für Server, für Netzwerkausrüstung*

  * inkl. 100 Edge Devices à 500 €, mind. 25.000 € Serverkosten (Erstanschaffung)

Cloud-Dienste:

- 78.600 - 165.000 €* Rechenleistung, Speicher, Netzwerke (z. B. AWS, Azure, Google Clou) und Analytics Monitoring)

  * je nach Komplexität der Datenverarbeitung

Entwicklungstools und Lizenzen:

- ca. 15.000 € Softwarelizenzen, Entwicklungsumgebungen

Gesamte Infrastrukturkosten:

- ca. 164.000 - 300.000 € (erstes Jahr)
- 50.000 - 110.000 € (laufende Infrastrukturkosten)

2.3 Wartungs- und Supportkosten

Laufende Wartung, Fehlerbehebung, Updates und Benutzersupport müssen ebenfalls berücksichtigt werden.

Laufendes Wartungsteam:

- 2 Entwickler*innen: 60.000 - 90.000 € / Jahr pro Entwickler*in (zur Reduzierung von Abhängigkeiten wäre ein Team aus drei Entwickler*innen empfehlenswert)
- 1 - 2 Support-Mitarbeiter*innen: 40.000 - 60.000 € / Jahr pro Person

Jährliche Wartungskosten:

- 160.000 - 300.000 €

2.4 Sicherheitsimplementierung

Die Gewährleistung der Sicherheit der IoT-Device-Management-Lösung erfordert sowohl anfängliche als auch laufende Investitionen auf mehreren Ebenen:

- **Software:** Sicherheitsaudits, Verschlüsselung, sichere Protokolle, regelmäßige Updates, Zugangskontrollen
- **Betriebssystem (inklusiv):** Gehärtete OS-Versionen, Secure Boot
- **Hardware (teilweise inkludiert):** TPM/HSM für Root-of-Trust, physischer Manipulationsschutz, Schlüsselmanagement

Anfängliche Sicherheitsimplementierung:

- Sicherheitsaudits: 40.000 - 80.000 €
- Verschlüsselungs- und sichere Kommunikationsprotokolle: 25.000 - 60.000 €
- Sicherheitstools/Softwarelizenzen: 15.000 - 40.000 €

Laufende Sicherheitsüberwachung:

- Sicherheitsspezialist*in: 40.000 - 60.000 € / Jahr (anteilig für 100 Geräte)
- Sicherheitstools: 15.000 - 40.000 € / Jahr
- Sicherheitsaudits: 20.000 - 40.000 € / Jahr

Gesamte Sicherheitskosten (erstes Jahr):

- Anfängliche Sicherheitsimplementierung: 80.000 - 180.000 €
- Laufende Sicherheitskosten: 75.000 - 140.000 € / Jahr

2.5 Skalierbarkeitsplanung

Die Skalierbarkeitsplanung umfasst alle Planungen für das zukünftige Wachstum, einschließlich der Fähigkeit, die Lösung zu skalieren, um eine größere Anzahl von Edge-Geräten verwalten zu können.

Skalierbarkeitsdesign und Implementierung:

- Architektonische Planung: 40.000 - 80.000 €
- Skalierungskosten für Cloud-Infrastruktur: 40.000 - 120.000 € / Jahr

Gesamte Skalierungskosten:

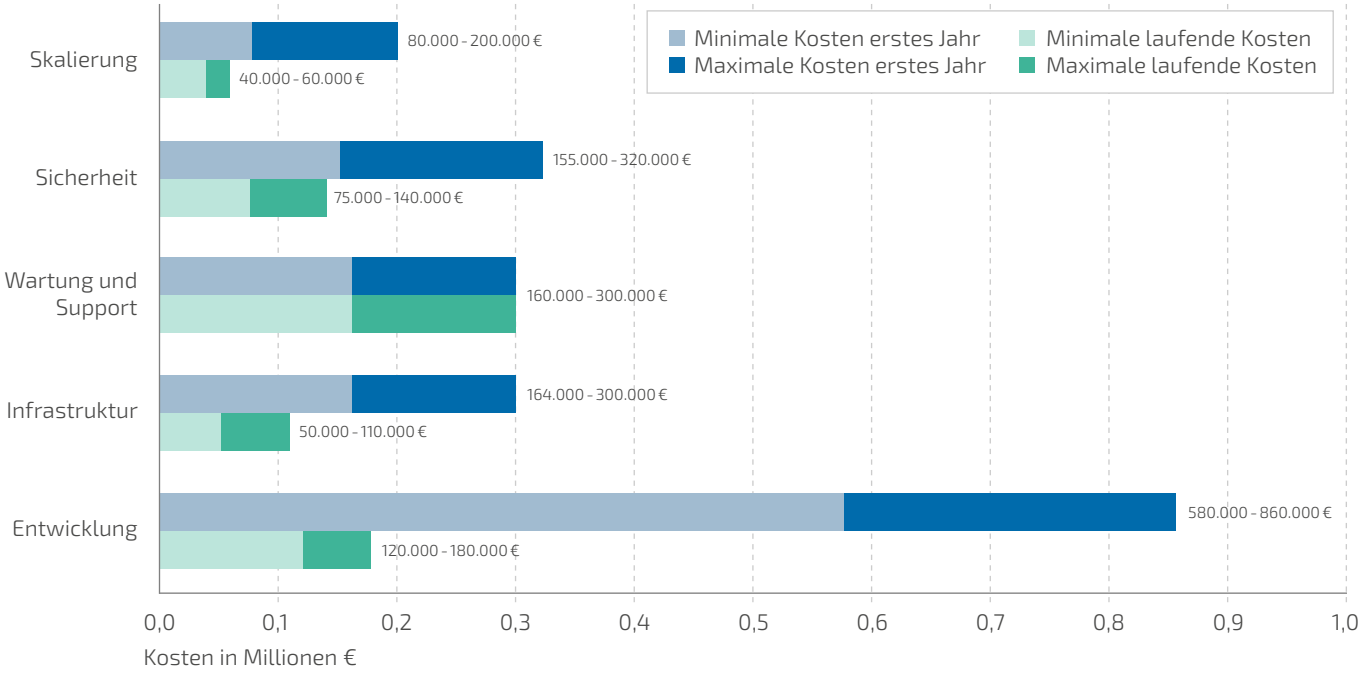
- 80.000 - 200.000 € (erstes Jahr)
- 40.000 - 60.000 € (laufende Skalierungskosten)

2.6 Gesamtkostenschätzung der Kosten (erstes Jahr) und Zeitabschätzung

Gesamtkostenschätzung

Die folgenden Übersichten fassen die zuvor geschätzten Kosten für das erste Jahr der Entwicklung einer eigenen IoT-Device-Management-Lösung zusammen:

Kostenkomponente	Kosten im ersten Jahr	laufende Kosten
Entwicklungskosten	580.000 - 860.000 €	120.000 - 180.000 €
Infrastrukturkosten	164.000 - 300.000 €	50.000 - 110.000 €
Wartung und Support	160.000 - 300.000 €	160.000 - 300.000 €
Sicherheit	155.000 - 320.000 €	75.000 - 140.000 €
Skalierung	80.000 - 200.000 €	40.000 - 60.000 €
Gesamtschätzung	**1,15 Mio - 1,98 Mio €**	**0,45 Mio - 0,79 Mio €**



Zeitabschätzung

Die Gesamtzeit für die vollständige Implementierung beträgt etwa 12 bis 18 Monate. Für diese Schätzung wurden bereits die Zeiten für die Entwicklung, Sicherheitsimplementierung und Skalierbarkeitsplanung berücksichtigt.

2.7 Schlussfolgerung

Die Entwicklung einer internen IoT-Device-Management-Lösung für Edge-Geräte erfordert erhebliche Investitionen. Die Kosten im ersten Jahr liegen voraussichtlich zwischen 1,15 und 1,98 Millionen Euro. Dieser Aufwand ermöglicht jedoch eine maßgeschneiderte Lösung, die präzise auf die Anforderungen Ihres Unternehmens abgestimmt ist. Gleichzeitig behalten Sie die volle Kontrolle über zukünftige Entwicklungen, Sicherheitsstandards und die Skalierbarkeit der Lösung.

Mit einer geschätzten Markteinführungszeit von 12 bis 18 Monaten kann die Eigenentwicklung langfristig signifikante betriebliche Vorteile bieten. Dazu gehören Effizienzsteigerungen durch optimiertes Flottenmanagement, Einsparpotenziale durch vorausschauende Wartungsfunktionen und verbesserte Datensicherheit. Die Entscheidung für eine Inhouse-Lösung sollte daher vor allem von den strategischen Zielen und den verfügbaren Ressourcen des Unternehmens abhängen.

3 Checkliste der notwendigen Schritte und Ressourcen für ein sicheres und skalierbares Device-Management

Diese Checkliste bietet Ihnen einen Überblick über die zentralen Aufgaben, die bei der Entwicklung einer eigenen IoT-Device-Management-Lösung zu beachten sind. Sie zeigt, welche Schritte notwendig sind, um eine zukunftssichere und zuverlässige Verwaltung Ihrer Geräteflotte sicherzustellen. Gleichzeitig unterstützt sie Sie dabei, zu prüfen, ob eine Eigenentwicklung für Ihre Anforderungen geeignet ist – oder ob eine fertige Lösung die effizientere Wahl darstellt.

Sicheres und redundantes Betriebssystem

Ziele:
Sicherheit und Widerstandsfähigkeit der Geräteoperationen

- Härtung des Betriebssystems: Konfiguration des Betriebssystems und seiner Komponenten, um potenzielle Angriffsflächen zu minimieren, etwa durch das Deaktivieren unnötiger Dienste und Funktionen
- Regelmäßige Updates und Patches, um bekannte Sicherheitslücken zu schließen und das Betriebssystem auf dem neuesten Stand zu halten
- Implementierung von Rollback-Funktionen
- Einrichtung einer Multi-Faktor-Authentifizierung (MFA) für den Zugang auf Betriebssystemebene

Datenschutz und Sicherheit

Ziele:
Schutz sensibler Daten und Einhaltung von Vorschriften

- Verschlüsselung der Daten im Ruhezustand und bei der Übertragung
- Implementierung anonymisierter Daten, wo möglich
- Nutzung von Protokollen und Techniken zur sicheren Datenübertragung, wie SSL / TLS für verschlüsselte Verbindungen
- Mechanismen zur Sicherstellung, dass Daten und Systeme nicht unbefugt verändert werden können, beispielsweise durch Prüfsummen und digitale Signaturen

Skalierbare und leistungsstarke Infrastruktur

Ziele:
Bewältigung von Wachstum und steigenden Lasten ohne Leistungseinbußen

- Implementierung von Lastausgleichs- und automatischen Skalierungsfunktionen
- Verwendung verteilter Systeme zur Vermeidung von Single Points of Failure
- Strategien zur Kostenoptimierung der Infrastrukturausgaben

Schwachstellenabgleich mit CVE- (Common Vulnerability Enumeration) / CWE- (Common Weakness Enumeration) Datenbanken

Ziele:
Identifizierung und Minimierung von Sicherheitsrisiken

- Regelmäßige Schwachstellenscans
- Einführung einer kontinuierlichen Überwachung auf neue Schwachstellen
- Warnmechanismus
- Automatischer Patching-Prozess für festgestellte Sicherheitslücken
- Implementierung von Tools, die Geräte automatisch mit aktuellen CVE- / CWE-Datenbanken vergleichen, um neueste Bedrohungen zu berücksichtigen

Notfallwiederherstellung und Geschäftskontinuität

Ziele:
Vorbereitung auf potenzielle Katastrophen und Sicherstellung des Betriebsfortgangs

- Regelmäßiges Backup kritischer Daten
- Regelmäßiges Testen von Wiederherstellungsverfahren

Versionsverwaltung

Ziele:
Kontrolle über Softwareversionen und Behebung von Schwachstellen

- Führen eines übersichtlichen Protokolls aller Versionen
- Sicherstellung der Abwärtskompatibilität

Einhaltung von Vorschriften und regulatorischen Anforderungen

Ziele:
Minimierung rechtlicher Risiken durch Einhaltung von Branchenstandards

- Einhaltung von Vorschriften wie DS-GVO und ISO / IEC 27001
- Durchführung regelmäßiger Audits
- Sicherstellung der Dokumentation und Berichtsmechanismen

Onboarding-Prozess

Ziele:
Sicheres und effizientes Hinzufügen neuer Geräte

- Verwendung sicherer Anmeldeinformationen und Authentifizierungen
- Automatisierung der Geräteregistrierung und -konfiguration
- Zertifikatsverwaltung

Gerätemanagement

Ziele:
Umfassende Verwaltung, Konfiguration und Wartung der Geräteflotte

- Automatisierung von Firmware-Updates und Gerätebereitstellungen
- Skalierbare Verwaltung einer großen Anzahl von Geräten
- Nutzung einer integrierten Analyseplattform für tiefere Einblicke in die Gerätegesundheit

Überwachung des Gerätezustands / Health Monitoring

Ziele:
Früherkennung und Behebung von Problemen

- Einrichtung von Warnmeldungen für ungewöhnliche Aktivitäten
- Nutzung prädiktiver Analysen zur Problemvorhersage
- Integration von Selbstreparaturmechanismen

Reibungslose Konnektivität

Ziele:
Konsistente und sichere Kommunikation zwischen den Geräten und Management-systemen

- Implementierung eines Fallback-Kommunikationskanals
- Optimierung der Konnektivität bei geringer Netzwerknutzung
- End-to-End-Verschlüsselung für Daten während der Übertragung

Remote-Support

Ziele:
Effiziente Fernwartung und -unterstützung der Geräte

- Sicherer Fernzugriff mit Prüfprotokollen
- Ermöglichung von Ferndiagnosen und -reparaturen
- Sichere Fernlöschung bei Diebstahl oder Kompromittierung

Intuitive Benutzeroberfläche

Ziele:
Benutzerfreundliche Verwaltung der Geräte

- Bereitstellung anpassbarer Dashboards und Berichte
- Verwaltung und Einschränkung von Benutzerrechten und Berechtigungen, um sicherzustellen, dass nur autorisierte Benutzer und Anwendungen auf sensible Daten und Funktionen zugreifen können

Edge-Computing-Fähigkeiten

Ziele:
Verbesserung der Leistung durch lokale Verarbeitung

- Implementierung von Sicherheit an der Edge zur Datensicherung

SBOM (Software-Stückliste)

Ziele:
Verwaltung von Schwach-stellen und Gewährleistung der Transparenz

- Regelmäßige Überprüfung der SBOM auf veraltete oder anfällige Komponenten
- Weitergabe der SBOM an Beteiligte zur Gewährleistung der Transparenz
- Nutzung automatisierter Werkzeuge für die Erstellung und Überprüfung der SBOM

Docker-Registry

Ziele:
Sicherstellung der sicheren Verwaltung von Docker-Images

- Regelmäßiges Scannen von Images auf Schwachstellen
- Implementierung von Zugriffskontrollen
- Rollenbasierter Zugriff auf die Registry
- Image-Signierung zur Überprüfung der Integrität

4 Technische Anforderungen und Lösungen

4.1 Wie IoT-Geräte angebunden werden

Viele Cloud-Anwendungen sind auf die Verarbeitung von Daten aus der physischen Welt angewiesen, z. B. von Sensoren, Maschinen oder anderen Geräten. Häufig können diese Daten jedoch nicht direkt in der Cloud verarbeitet werden, da die erforderlichen Protokolle oder eine sichere Verbindung fehlen. Hier kommen IoT-Geräte ins Spiel, die oft auch als Edge-Geräte bezeichnet werden. Ihr Name verdeutlicht ihre Funktion: Sie befinden sich am Rand des zentralen Cloud-Systems und stellen die Verbindung zur realen Welt her.

Die Hauptaufgabe solcher Geräte besteht in der Umwandlung von Daten aus einem bestehenden Format in ein für die Cloud verwertbares Format. Für diese Aufgabe reicht meist eine geringe Rechenleistung, weshalb IoT-Geräte oft kleinere, kostengünstige Computer sind. Diese Geräte werden häufig als technische Notwendigkeit angesehen, ohne dass sie für Endanwender*innen direkt sichtbaren Nutzen bieten. Unternehmen legen daher großen Wert auf die Minimierung der Hardware- und Softwarekosten – von Betriebssystemen bis zu den Anwendungen.

Trotz ihrer kompakten Bauweise und begrenzten Funktionalität sind IoT-Geräte besonders anfällig für Sicherheitsrisiken. Ihr Einsatz am Rand des Cloud-Systems macht sie zu einem potenziellen Schwachpunkt, der die gesamte Dateninfrastruktur gefährden kann. Viele dieser Geräte sind an schwer zugänglichen oder unbeaufsichtigten Orten montiert, wie direkt an Maschinen oder in abgelegenen Anlagen. Dadurch sind sie sowohl physischen als auch digitalen Angriffen ausgesetzt.

Die Sicherheitsanforderungen für IoT-Geräte werden häufig unterschätzt, was zu schwerwiegenden Folgen führen kann. Ein kompromittiertes Gerät kann nicht nur sensible Daten gefährden, sondern auch die Integrität des gesamten Systems beeinträchtigen. Daher ist es entscheidend, die Sicherheitsmaßnahmen für diese Geräteklasse gezielt zu planen und umzusetzen.

Exkurs: Pflege des Systems

Die Auslieferung eines Betriebssystems setzt einen kontinuierlichen Prozess in Gang. Ständig werden neue Sicherheitslücken entdeckt, die die Geräte, auf denen diese Software installiert ist, gefährden. Durch die stetig steigende Komplexität der Software nimmt auch die Anzahl der Sicherheitslücken stetig zu. So wurden im Jahr 2023 über 26.000 Schwachstellen[1] registriert und damit die Vorjahreszahl um ca. 1.500 übertroffen. Es ist daher äußerst wichtig, die erkannten Sicherheitslücken laufend zu analysieren und dafür zu sorgen, dass diese schnellstmöglich geschlossen werden. Berücksichtigt man zusätzlich, dass für ca. 80 % aller Schwachstellen bereits Exploits[2] (Code zum Ausnutzen der Schwachstelle) verfügbar sind, wird die Wichtigkeit einer schnellen Reaktion deutlich. Diese Schwachstellen sind auch am häufigsten Ziel von Angriffen, wobei etwa 3 von 4 Angriffen auf Schwachstellen[3] abzielen, die bereits 2 Jahre oder älter sind. Auch wenn das Risiko nicht vollständig eliminiert werden kann, wird durch die Bereitstellung eines Patches oder Updates kurz nach der Veröffentlichung der Schwachstelle das Angriffsrisiko erheblich reduziert.

Um die Sicherheit langfristig zu gewährleisten, sollte eine kontinuierliche Überwachung von Schwachstellen über geeignete Quellen wie CVE (Common Vulnerabilities and Exposures) oder CWE (Common Weakness Enumeration) erfolgen. Sobald eine kritische Schwachstelle identifiziert wird, ist es entscheidend, zügig Patches bereitzustellen und diese effizient auf die gesamte Flotte auszurollen. Dies erfordert einen klar definierten und etablierten Prozess.

Software Bill Of Materials (SBOM)

Bei der Erstellung eines Betriebssystems werden alle Komponenten wie externe Bibliotheken, Treiber oder andere Softwarebausteine erfasst und in einer Liste zusammengeführt. Dieser Vorgang kann z. B. einem etablierten Standard folgen und als CyclonDX-Datei zur Verfügung gestellt werden. Diese Datei sollte mit der Software ausgeliefert werden, so dass Anwender*innen auch eigene Überwachungen durchführen kann. Darüber hinaus ist der Import in datenbankbasierte Systeme sinnvoll, um die Überwachung zu automatisieren.

CVE- und CWE-Scans

Mit der Verfügbarkeit der Softwarekomponenten in einem Datenbanksystem kann eine automatisierte Überwachung mit jeweils aktuellen Daten der CVE- / CWE-Datenquellen (z. B. NVD = National Vulnerability Database) erfolgen. Dabei werden für jede Softwarekomponente relevante Informationen über potenzielle Sicherheitslücken in den verfügbaren Quellen bereitgestellt. So können geeignete Maßnahmen wie Patches oder Updates schnell identifiziert und umgesetzt werden, um das Risiko gezielt zu minimieren.

[1] https://www.comparitech.com/blog/information-security/cybersecurity-vulnerability-statistics und https://blog.qualys.com/vulnerabilities-threat-research/2023/12/19/2023-threat-landscape-year-in-review-part-one, 10.12.2024
[2] https://unit42.paloaltonetworks.com/state-of-exploit-development, 10.12.2024
[3] https://www.checkpoint.com/downloads/resources/cyber-security-report-2021.pdf, 10.12.2024

4.2 Was getan werden muss, um die Anwendung auf das Gerät zu bringen?

Installationsprozess:
Die Installation einer Anwendung auf einem IoT-Gerät erfordert mehrere Schritte, die sorgfältig geplant und effizient umgesetzt werden müssen. Zunächst wird die Laufzeitumgebung vorbereitet, etwa durch die Verwendung von Docker. Anschließend erfolgt die Bereitstellung der Anwendung. Automatisierte Tools und Skripte spielen hierbei eine zentrale Rolle, um den Prozess nicht nur reibungslos, sondern auch fehlerfrei zu gestalten.

1. **Vorbereitung der Laufzeitumgebung:**
   Eine Docker Engine (Runtime) muss auf dem IoT-Gerät installiert sein. Diese Laufzeitumgebung stellt sicher, dass die Anwendung in einem isolierten und konsistenten Umfeld läuft, unabhängig von der zugrunde liegenden Hardware.

2. **Bereitstellung der Anwendung:**
   Die Anwendung wird in Form eines Docker Images in einer Docker Registry abgelegt. Hier dient die Registry als zentraler Speicherort, von dem die Geräte die Images abrufen können.

3. **Automatisierung:**
   Mithilfe des Docker Command Line Interface (CLI) können Build- und Release-Prozesse automatisiert werden, um Effizienz und Genauigkeit zu maximieren. Automatisierte Prozesse minimieren menschliche Fehler und ermöglichen eine skalierbare Bereitstellung über eine große Anzahl von Geräten hinweg.

4.3 Wie kann Kontron Hardware und Software helfen?

Die auf Yocto-Linux® basierenden IoT-Devices von Kontron bringen die notwendige Docker Runtime bereits mit. Damit sind sie die optimale Basis für den Einsatz von containerisierten Anwendungen.

Update-Mechanismen:
Regelmäßige Updates sind essenziell, um Anwendungen auf IoT-Geräten sicher und funktional zu halten. Over-the-Air-Updates (OTA) ermöglichen dabei eine flexible Aktualisierung ohne physischen Zugriff. Besonders bei großen Geräteflotten ist ein schrittweiser Rollout sinnvoll: Updates werden zunächst in kleinen Gruppen getestet, bevor sie sukzessive auf alle Geräte ausgerollt werden. So lassen sich Fehler minimieren und Ausfallrisiken reduzieren.

Die IoT-Device-Management-Lösung KontronGrid erleichtert diesen Prozess erheblich und sorgt für eine zuverlässige Umsetzung. Mithilfe der Vorlagenfunktion können Updates effizient vorbereitet werden, indem Docker Container, Compose-Spezifikationen und das passende Betriebssystem in einer zentralen Vorlage gebündelt werden. Diese Templates lassen sich mit wenigen Klicks auf einzelne Geräte oder komplette Gerätegruppen anwenden, wodurch der Update-Prozess automatisiert und beschleunigt wird.

Dies ermöglicht auch sukszessive Updatemechanismen. So können gezielt Untergruppen von Geräten, etwa basierend auf Region, Kunde oder Gerätetyp, aktualisiert werden, bevor die gesamte Flotte angepasst wird. Dadurch wird sichergestellt, dass die Updates fehlerfrei funktionieren und die Geräteflotte stets mit den neuesten technologischen Entwicklungen Schritt hält.

Flexibilität und Erweiterung

Die Systemarchitektur sollte so ausgelegt sein, dass Anwendungen leicht erweitert oder modifiziert werden können, um auf neue Anforderungen zu reagieren. Dies erfordert eine flexible Systemarchitektur und geeignete Tools für die Verwaltung der Anwendungen. Der beschriebene Update-Mechanismus ist ein Beispiel für die Flexibilität, die sowohl das Betriebssystem als auch die Docker Container bzw. Docker Compose-Konfigurationen umfasst. KontronGrid unterstützt diese Flexibilität durch eine Architektur, die sowohl Updates von Betriebssystemen und Anwendungen als auch die Verwaltung von Docker Containern und Compose-Konfigurationen ermöglicht. Damit können selbst komplexe Anwendungen auf einem Gerät orchestriert und die Interaktion mehrerer Docker Container nahtlos koordiniert werden.

Automatisierung mit Docker und CLI

Automatisierung ist ein Schlüsselfaktor in professionellen Entwicklungsprozessen. Der Prozess vom Einchecken einer Quellcode-Änderung über die Erstellung der Anwendung bis hin zur Bereitstellung der erstellten Dateien muss automatisiert und sicher sein. Dazu gehört auch das Erstellen und Hochladen von Docker Images. Die native Funktionalität der Docker Kommandozeilenschnittstelle (CLI) spielt dabei eine zentrale Rolle. Sie ermöglicht, manuelle Prozesse wie Berechtigungen und das Hochladen neuer Images effizient zu automatisieren. Entwickler*innen und IT-Teams können so Zeit sparen und Fehler minimieren, indem sie sich wiederholende Aufgaben bei der Bereitstellung automatisieren.

Komplexität reduzieren mit Docker Compose

In der Realität bestehen IoT-Anwendungen häufig nicht aus einzelnen Containern, sondern aus einer Reihe von Containern, die nahtlos zusammenarbeiten müssen. Diese Interaktionen zwischen den Containern erhöhen die Komplexität der Anwendung und erfordern ein intelligentes Managementsystem. Docker Compose bietet hierfür eine clevere Lösung. Es wurde für das Management von Multi-Container-Anwendungen entwickelt und ist nahtlos in die KontronGrid-Plattform integriert.

Für Softwareentwickler*innen und Architekt*innen ist Docker Compose das Werkzeug, um die Komplexität bei der Verwaltung von Containern auf ein Minimum zu reduzieren.

Docker Compose basiert auf einer serviceorientierten Architektur (SOA), die es ermöglicht, Anwendungen in kleinere, überschaubare Services zu unterteilen, die miteinander interagieren. Dies verbessert nicht nur die Wartbarkeit und Erweiterbarkeit der Anwendungen, sondern bietet auch eine klar strukturierte Organisation der Anwendungsarchitektur. Der größte Vorteil von Docker Compose ist, dass es den gesamten Prozess der Container-Orchestrierung vereinfacht.

Anstatt einzelne Container zu verwalten, behandelt Docker Compose Anwendungen als ein zusammenhängendes System. Die gesamte Konfiguration wird in einer einzigen Datei zentralisiert, was eine übersichtliche und effiziente Verwaltung ermöglicht. Dies reduziert den Aufwand beim Deployment erheblich und vermeidet komplizierte Rollout-Strategien.

Vereinfachte Orchestrierung und Konfiguration

Dank Docker Compose wird die Zusammenarbeit der Services optimiert und die Verwaltung der Container so gestaltet, dass ein reibungsloses Zusammenspiel innerhalb von KontronGrid gewährleistet ist. Effizientes Management und klare Strukturen stehen im Vordergrund, um den Herausforderungen moderner IoT-Anwendungen gerecht zu werden.

Folgende Funktionen können realisiert werden:

1. **Erstellung von Docker Compose-Projekten** inklusive Konfiguration im YAML-Format. Dies ermöglicht eine schnelle und standardisierte Bereitstellung von Anwendungen, die aus mehreren Containern bestehen.
2. **Überwachung von Compose-Projekten** mithilfe der Compose-Console, die durch Log-Einträge detaillierte Debugging- und Diagnoseinformationen bereitstellt. Dadurch wird eine transparente Kontrolle über den Zustand der Container ermöglicht.
3. **Bearbeitung bestehender Compose-Projekte**, einschließlich der Optimierung der Anwendungskonfiguration, um eine maximale Effizienz und Anpassungsfähigkeit zu gewährleisten.
4. **Einzelnes oder gemeinsames Steuern von Compose-Projekten**, z. B. das Neustarten oder Stoppen einzelner oder aller Container. Dies erleichtert das Management, insbesondere bei global verteilten IoT-Geräten.
5. **Automatisiertes Neustarten von Containern** bei unerwartetem Verhalten oder Problemen. Diese Funktion stellt sicher, dass Fehler im Betrieb sofort behoben werden und die Anwendungen weiterhin reibungslos laufen.
6. **Löschen von Compose-Containern ohne Verlust der Konfigurationsdaten.** Die Projekte können bei Bedarf wiederhergestellt werden, was eine flexible und skalierbare Lösung für wechselnde Anforderungen bietet.
7. **Endgültiges Löschen von Compose-Projekten von einem Edge-Gerät**, um Speicherplatz freizugeben oder veraltete Anwendungen zu entfernen.
8. **Nutzung von Docker Compose als Vorlage für das Flottenmanagement.** Dies vereinfacht den Prozess der Verwaltung mehrerer Geräte und Anwendungen, da die zentralisierte Konfigurationsdatei jederzeit dupliziert und für andere Geräte oder Projekte angepasst werden kann.

Exkurs: Docker Container Anwendungen

Docker auf Linux®

Docker ist eng in die Linux®-Umgebung integriert und nutzt zahlreiche Linux®-Kernel-Features, wie cgroups und Namespaces, um isolierte und effiziente Container bereitzustellen. Dies macht Docker zu einer idealen Plattform für modulare und skalierbare Anwendungen.

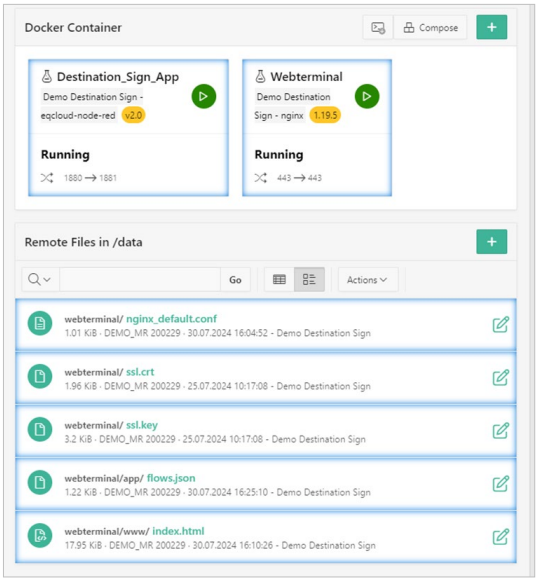
Beispiel: Zentrales Leitsystem für die Fahrzielanzeige

Ein konkretes Beispiel für den Einsatz von Docker ist die Simulation eines zentralen Leitsystems, das die Fahrzielanzeige eines Busses steuert. Diese Lösung umfasst mehrere Docker-Container:

1. NodeRed-Container (Destination_Sign_App):
   - Dieser Container simuliert den Busbetrieb, indem er über einen Mini-Webservice jede Minute die Liniennummer und das Fahrziel des virtuellen Busses ändert.
   - Das Projekt wird über Remote Files in den Container geladen und demonstriert, wie ein zentraler Dienst die Zielinformationen dynamisch steuert.
   - Obwohl die Simulation keinen realen Bus oder physische Anzeige verwendet, veranschaulicht sie das Potenzial einer Anbindung an ein zentrales Leitsystem, das Fahrziele für Busse steuert.

2. Webterminal-Container:
   - Dieser Container dient als symbolische Darstellung der Anzeigetafel, da keine physische Anzeige verfügbar ist. Die Ziel- und Liniennummer werden in einer webbasierten Schnittstelle visualisiert.
   - Zugriff über URL: https://172.16.102.191 und https://172.16.102.141
   - Das Webterminal holt die aktuellen Daten aus dem NodeRed-Container und zeigt, wie Container miteinander kommunizieren können.

3. Webserver-Container (Nginx):
   - Ein Docker-Container, der einen Webserver mit Nginx und SSL-Konfiguration bereitstellt.
   - Dies demonstriert, wie ein sicherer Webserver innerhalb einer Docker-Umgebung erstellt und betrieben werden kann.



Diese Anwendung wurde als Template auf die Geräte DEVICE001 und DEVICE002 ausgerollt, was zeigt, wie leicht Docker-Umgebungen standardisiert und skaliert werden können.

Weitere Docker-Anwendungsbeispiele

Docker ermöglicht die Erstellung vielseitiger Anwendungen, die flexibel in Containern betrieben werden können. Beispiele hierfür sind:

NodeRed-Container:

- Verwaltet die Verbindung zu Maschinen und speichert Daten in einer Datenbank

Database-Container:

- Speichert Maschinendaten und ermöglicht deren schnelle Abfrage und Analyse

UserUI-Container:

- Stellt eine Benutzeroberfläche für die Analyse von Maschinendaten bereit

Webserver oder Dashboards:

- Erstellen von Schnittstellen und Management-Plattformen für verschiedene Anwendungen

FabEagle®Connect-Container:

- Low-Code-Schnittstellen-Framework für die Realisierung komplexer Schnittstellen im Fabrikumfeld

4.4 Wie können IoT-Geräte sicher aus der Ferne gewartet werden?

Sind die IoT-Geräte in Betrieb, ist es essentiell, dass sie unterbrechungsfrei funktionieren. Das integrierte Health Monitoring kann Aufschluss über wichtige Parameter (Speicher- und CPU-Auslastung, Verbindungsstatus, Betriebszeittemperatur, etc.) der IoT-Geräte geben. Treten im Zweifel Störungen auf, muss der Service und Support reaktionsschnell agieren können, um Verbindungsunterbrechungen oder Datenverlust vorzubeugen. Über das KontronGrid kann die Fernwartung der IoT-Geräte browserbasiert entweder über Remote-Shell-Desktop (RDP) oder Secure-Shell-Dienste (SSH) erfolgen. Ersterer erlaubt einen Zugriff auf die Konsole des Geräts, letzterer den Zugriff auf die grafische Verbindung zum Desktop aus der Ferne. Liegt die Ursache der Störung auf der Maschinenebene, kann über den temporären Verbindungsaufbau eines virtuellen Netzwerks (VPN-Tunnel) verschlüsselt auf die mit dem IoT-Gateway, Server oder PC verbundenen Maschinen und Systeme im Kundennetzwerk zugegriffen werden. Das VPN-Tunneling ermöglicht nicht nur die Bedienung der Programmiersoftware oder die Parametrierung an der jeweiligen Steuerung (SPS) vorzunehmen, sondern auch die ortsunabhängige Zusammenarbeit mehrerer Service-Techniker*innen per Fernwartung an einer Maschine oder Gerät. Die Verbindungen werden dabei automatisch geloggt.

5 Cybersecurity-Anforderungen an digitale Produkte und Lösungen

Die Softwareentwicklung für IoT-Geräte muss nicht nur technisch einwandfrei sein, sondern auch eine Vielzahl gesetzlicher Anforderungen erfüllen. Dazu gehören unter anderem die Datenschutz-Grundverordnung (DS-GVO), NIS-2 und die Normen der ISO 27001. Diese Vorschriften legen klare Richtlinien für den Datenschutz, die Datensicherheit und die Verarbeitung personenbezogener Daten fest. Sie zielen darauf ab, die Integrität der Systeme und die Vertraulichkeit von Daten über den gesamten Lebenszyklus der Geräte sicherzustellen.

Programmsicherheit: Umsetzung von Security by Design und Security by Default, um Schwachstellen bereits in der Entwicklungsphase zu minimieren.

Netzwerksicherheit: Maßnahmen zum Schutz der IT-Infrastruktur vor unerlaubtem Zugriff.

Betriebssicherheit: Förderung sicherer Verhaltensweisen im Umgang mit IT und Software im Unternehmensalltag.

Endnutzer-Aufklärung: Schulungen und Maßnahmen zur proaktiven Minimierung von Risiken durch menschliche Fehler.

Disaster Recovery und Business Continuity: Strukturen für schnelle Reaktionen und Wiederherstellung nach Sicherheitsvorfällen.

Kontron spielt als Hard- und Softwarelieferant eine zentrale Rolle in der Lieferkette und unterstützt Unternehmen in den betroffenen Branchen bei der Erfüllung der Anforderungen des Cyber Resilience Act (CRA). Wir stellen sicher, dass die in den Gesamtlösungen integrierten Kontron-Komponenten, sowohl Hardware als auch Software, über den gesamten Lebenszyklus abgesichert sind und erleichtern so die Einhaltung der Cybersicherheitsanforderungen. Dabei setzen wir auf zwei maßgebliche Sicherheitsprinzipien: Security by Design und Security by Default. Diese beiden Prinzipien werden in den folgenden Kapiteln genauer vorgestellt.

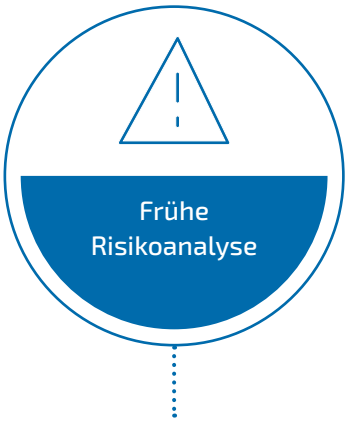
5.1 Security by Design

Sicherheit sollte nicht erst im Nachhinein betrachtet werden – sie gehört in den Mittelpunkt jeder Produktentwicklung, von der ersten Idee bis zur Umsetzung. Der Ansatz Security by Design stellt sicher, dass Sicherheitsaspekte von Anfang an systematisch in den Entwicklungsprozess integriert werden. Das ist gerade im Bereich der IoT-Lösungen essenziell, da sowohl Hardware als auch Software besonders anfällig für Cyberangriffe sind.

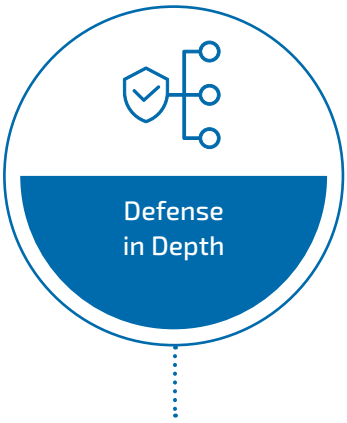
Security by Design sollte in den Mittelpunkt jeder Produktentwicklung gestellt, von der ersten Idee bis zur Umsetzung.

Mit der Einführung der **Network and Information Systems Directive (NIS-2)** und dem bevorstehenden **Cyber Resilience Act (CRA)** gewinnt dieser Ansatz weiter an Bedeutung. Beide Regelwerke fordern explizit, Produkte und Dienstleistungen so zu gestalten, dass Sicherheitsrisiken bereits in der Planungsphase minimiert werden und eine hohe Resilienz gegenüber Cyber-Bedrohungen erreicht wird.

Im Kontext von NIS-2 und CRA umfasst Security by Design insbesondere die folgenden Maßnahmen:



Bereits in der Planungsphase werden potenzielle Sicherheitsrisiken identifiziert und durch gezielte Designentscheidungen minimiert.



Mehrstufige Sicherheitsmaßnahmen sorgen dafür, dass selbst bei einem erfolgreichen Angriff nicht das gesamte System gefährdet ist. Hierbei wird eine mehrschichtige Verteidigungsstrategie angesetzt, die sicherstellt, dass das Gesamtsystem weiterhin geschützt bleibt. Diese Maßnahmen umfassen:

- Segmentierung des Netzwerks durch sichere Zonen
- Mehrfache Authentifizierung und Rechtevergabe
- Regelmäßige Patches und Updates, um neue Sicherheitslücken zu schließen

Durch regelmäßige Audits und Penetrationstests wird die Sicherheit des Systems laufend überprüft und verbessert.

5.2 Security by Default

Eine zentrale Anforderung der NIS-2-Richtlinie und des CRA ist die sogenannte Security by Default. Dieses Prinzip verlangt, dass Sicherheitsvorkehrungen standardmäßig in Systeme, Produkte und Prozesse integriert und automatisch aktiviert werden. Dadurch wird sichergestellt, dass die höchste Sicherheit bereits ab Werk gewährleistet ist, ohne dass zusätzliche Konfigurationen erforderlich sind. Um diese Vorgaben zu erfüllen, sind umfassende technische, organisatorische und regulatorische Maßnahmen notwendig. Die wichtigsten Schritte werden nachfolgend zusammengefasst.

1. Technische Maßnahmen

1.1 Sichere Standardkonfiguration

- Systeme und Produkte müssen mit sicheren Standardeinstellungen ausgeliefert werden und standardmäßig über aktivierte Sicherheitsfunktionen verfügen:
  - Deaktivierung nicht benötigter Dienste
  - Restriktive Zugriffsbeschränkungen
  - Minimierung der Angriffsfläche
  - Firewalls
  - Verschlüsselung
  - Zugriffsbeschränkungen

1.2 Automatische Sicherheitsupdates

- Automatisierte Mechanismen für Sicherheitsupdates, um Systeme stets auf dem neuesten Stand zu halten
- Regelmäßige Schwachstellenüberprüfung und zeitnahe Implementierung von Updates

1.3 Schutz vor Schwachstellen

- Proaktive Erkennung von Sicherheitslücken durch Tools für Schwachstellenmanagement
- Regelmäßige Sicherheitsüberprüfungen, inklusive Penetrationstests

1.4 Verschlüsselung und Authentifizierung

- Moderne Verschlüsselungsstandards für gespeicherte und übertragene Daten
- Multi-Faktor-Authentifizierung (MFA) als Standard
- Prinzip der geringsten Privilegien für Benutzerkonten

1.5 Netzwerksicherheit

- Netzwerksegmentierung zur Eindämmung potenzieller Sicherheitsvorfälle
- Einsatz von Intrusion Detection- und Prevention-Systemen (IDS/IPS)
- Verschlüsselung der Datenübertragung mit sicheren Protokollen wie TLS

2. Prozessuale Maßnahmen

2.1 Risikoanalyse und Sicherheitsbewertungen

- Kontinuierliche Risikoanalysen und Sicherheitsbewertungen bilden die Grundlage für gezielte Sicherheitsstrategien
- Dokumentation der Risiken und entsprechenden Gegenmaßnahmen

2.2 Incident Response und Monitoring

- Etablierung eines Incident Response Plans für Cybervorfälle
- Echtzeit-Monitoring kritischer Systeme und Netzwerke zur schnellen Identifikation von Sicherheitsvorfällen

2.3 Schulungen und Sensibilisierung

- Regelmäßige Schulungen für Mitarbeiter*innen zu aktuellen Bedrohungen und Sicherheitspraktiken
- Aufbau einer Sicherheitskultur im Unternehmen durch Awareness-Programme

2.4 Lieferketten-Sicherheit (Supply Chain Security)

- Integrierung von Sicherheitsanforderungen in Vertragsbedingungen
- Kenntnis über die Lieferkette: Vollständige Transparenz über alle Lieferanten und deren Sicherheitsstatus
- Sicherheitsüberprüfung: Regelmäßige Audits und Evaluierungen der Sicherheitsmaßnahmen von Lieferanten und Drittanbietern
- Handlungsempfehlungen: Bei identifizierten Schwachstellen geben wir klare Handlungsempfehlungen zur Verbesserung der Sicherheitslage

3. Regulatorische Maßnahmen

3.1 Nachweis der Compliance

- Dokumentation der Einhaltung von Security by Default durch:
  - Regelmäßige Audit
  - Erstellung von Berichten zur Cybersicherheit

3.2 Zusammenarbeit mit Behörden

- Meldepflichten bei Sicherheitsvorfällen einhalten
- Enge Zusammenarbeit mit Behörden zur Erfüllung regulatorischer Anforderungen

5.3 Unterschiede, Rollenverteilung und Abgrenzung

Unterschiede auf einen Blick:

Security by Design		Security by Default
Während der Entwicklungs- und Designphase	Zeitpunkt der Umsetzung	Während der Bereitstellung (Out-of-the-box)
Proaktive Integration von Sicherheit ins Design	Fokus	Voreinstellungen für maximale Sicherheit
Entwickler*innen und Designer*innen	Zielgruppe	Endnutzer*innen
Sicherheitsprobleme vermeiden	Ansatz	Sichere Nutzung ohne zusätzliche Konfiguration
Threat Modeling, sichere Architektur	Beispiele	Deaktivierung ungenutzter Dienste, sichere Defaults

Die Sicherheit digitaler Produkte und Lösungen ist eine Gemeinschaftsaufgabe. Die Normenreihe IEC 62443 der International Electrotechnical Commission (IEC) ist ein international anerkannter Standard für die Sicherheit von industriellen Automatisierungs- und Steuerungssystemen. Sie bietet einen ganzheitlichen Ansatz zur Cybersicherheit und definiert Anforderungen und Prozesse für alle Beteiligten,

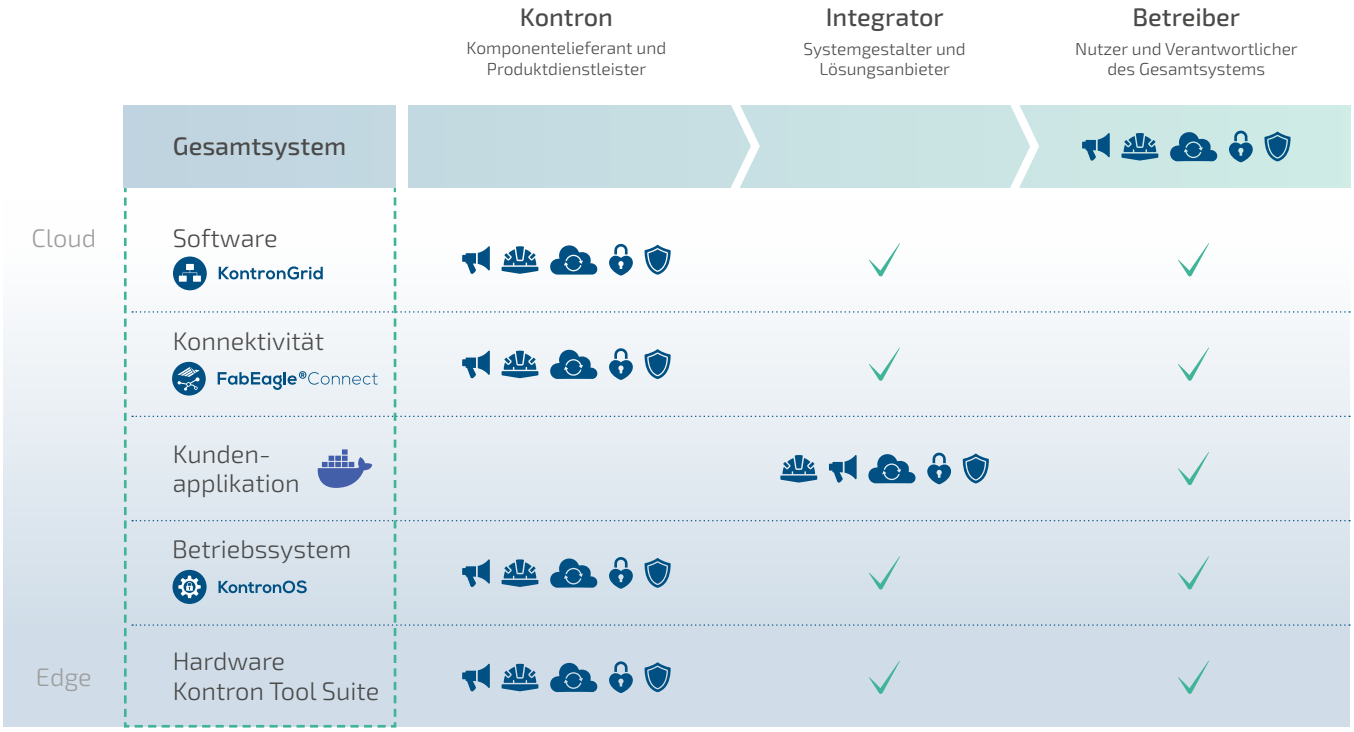
einschließlich Betreiber, Produktlieferanten und Dienstleister. Die IEC 62443 unterscheidet zwischen den einzelnen Akteuren und fordert von jedem, spezifische Teile der Norm zu erfüllen. Als führender Hersteller von Hard- und Softwarelösungen übernimmt Kontron in der Rolle des Product Supplier Verantwortung für die Sicherheitsanforderungen über den gesamten Produktlebenszyklus hinweg.

Darüber hinaus unterstützen unsere Produkte Hersteller von IoT-Lösungen, die ihre Lieferkette schützen und sichere Komponenten integrieren müssen. Kontrons Lösungen helfen Integratoren, Herstellern und Betreibern gleichermaßen, effektiv auf Sicherheitsvorfälle zu reagieren und ihre Systeme widerstandsfähiger zu machen. Ein zentraler Akteur in diesem Ökosystem ist der Integrator, der eine Vielzahl an Aufgaben übernimmt, um eine sichere und effiziente IoT-Lösung zu ermöglichen:

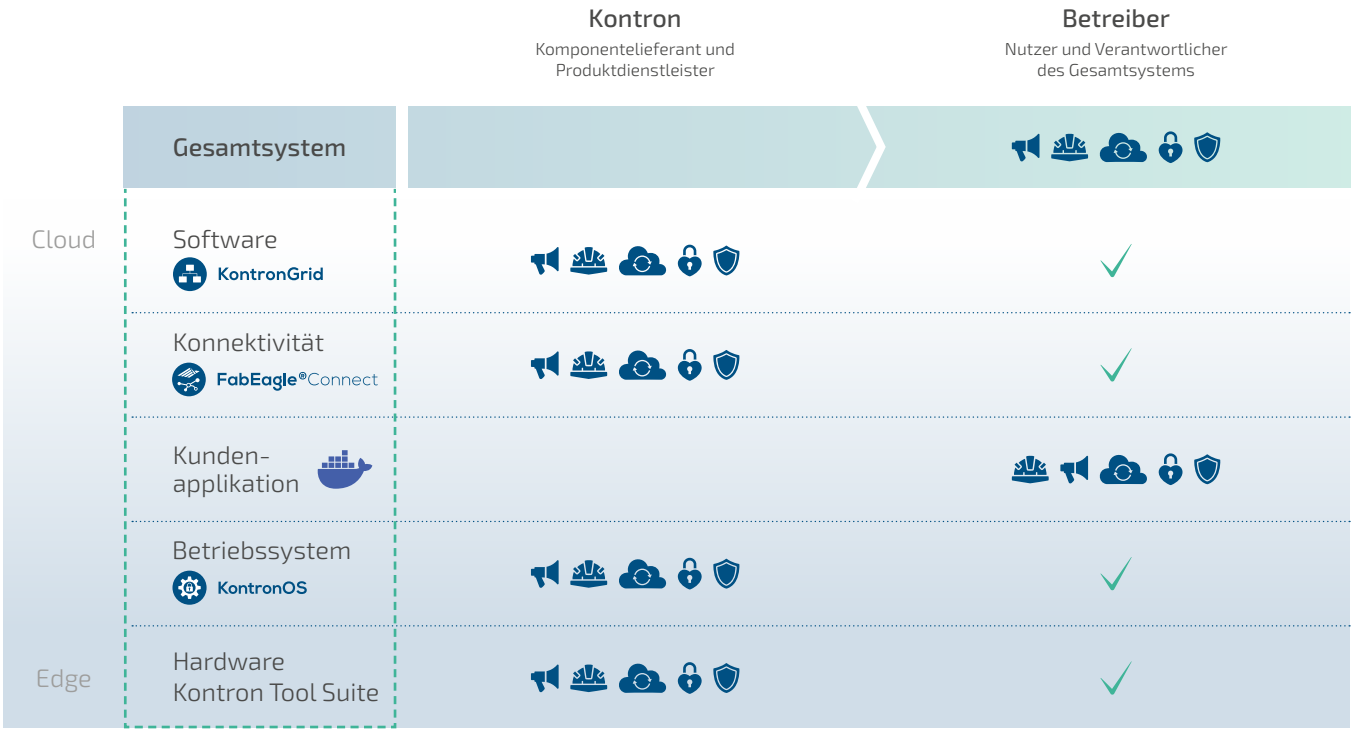
- **Systemintegration:**
  Die Kombination verschiedener Hard- und Softwarekomponenten zu einer funktionierenden IoT-Lösung. Der Integrator stellt die nahtlose Kommunikation zwischen Geräten, Plattformen und Netzwerken sicher, indem er Protokolle und APIs einbindet.
- **Customizing und Anpassung:**
  IoT-Lösungen werden an die spezifischen Anforderungen des Betreibers oder Endkunden angepasst. Dazu gehört die Entwicklung individueller Applikationen, Schnittstellen, Dashboards und Analysetools.
- **Projektmanagement:**
  Der Integrator übernimmt die Koordination von Entwicklungsprojekten, die Budgetierung und die Abstimmung zwischen allen beteiligten Parteien.
- **Technische Expertise und Beratung:**
  Er berät Betreiber und Endkunden zu Architektur, Sicherheit und Skalierbarkeit der IoT-Lösung und stellt sicher, dass alle Komponenten den relevanten Sicherheitsanforderungen entsprechen.

Im Zusammenspiel zwischen den beteiligten Akteuren – Kontron als Anbieter von Hard- und Software, Integratoren, die kundenspezifische Anwendungen entwickeln, und Betreibern, die das Gesamtsystem einsetzen und absichern – trägt jeder eine entscheidende Rolle.

Dieses Zusammenspiel, entweder mit zwei oder drei Akteuren, wird in den beiden nachfolgenden Grafiken beispielhaft skizziert.



Cyber Security Aufgaben und Verantwortlichkeiten am Beispiel eines Gesamtsystems mit **drei Akteuren**



Cyber Security Aufgaben und Verantwortlichkeiten am Beispiel eines Gesamtsystems mit **zwei Akteuren**

Endkundenaufklärung · Disaster & Business Continuity · Betriebssicherheit · Programmsicherheit · Netzwerksicherheit

5.4 Die Sicherheitsaspekte von KontronOS und KontronGrid

Als Hardware- und Softwarehersteller tragen wir die Verantwortung, den gesamten Produktlebenszyklus unserer Lösungen vollständig abzusichern. Das bedeutet, dass wir in jeder Phase – von der Entwicklung über die Implementierung bis hin zur Wartung und dem Betrieb – höchste Sicherheitsstandards gewährleisten. Unsere Lösungen, wie KontronGrid und KontronOS, bieten umfassende sicherheitsrelevante Funktionen, die den Anforderungen internationaler Standards wie ISO 27001 und IEC 2443 entsprechen. Dadurch stellen wir sicher, dass unsere Produkte nicht nur leistungsfähig, sondern auch robust gegen moderne Cyberbedrohungen sind.

ISO 27001

Die ISO 27001-Zertifizierung stellt sicher, dass ein umfassendes und systematisches Sicherheitsmanagement implementiert ist, das alle Aspekte der Informationssicherheit abdeckt. Wesentliche Bestandteile umfassen:

- **Informationssicherheitspolitiken (A.5):** Entwicklung und Durchsetzung klarer Richtlinien
- **Organisation der Informationssicherheit (A.6):** Etablierung klarer Verantwortlichkeiten und Strukturen
- **Sicherheitsmanagement bei der Personaleinstellung (A.7):** Prüfung und Schulung von Mitarbeitenden im Sicherheitskontext
- **Asset-Management (A.8):** Schutz und Verwaltung aller physischen und digitalen Vermögenswerte
- **Zugangskontrolle (A.9):** Differenzierte Zugriffsberechtigungen und Schutz vor unbefugtem Zugriff
- **Kryptografie (A.10):** Einsatz moderner Verschlüsselungsstandards
- **Physische Sicherheit (A.11):** Schutz von Einrichtungen und Geräten vor physischen Gefahren
- **Betriebssicherheit (A.12):** Sicherstellung stabiler und sicherer Betriebsprozesse
- **Kommunikationssicherheit (A.13):** Schutz von Datenübertragungen
- **Lieferantenbeziehungen (A.15):** Integration von Sicherheitsanforderungen in die Lieferkette
- **Management von Sicherheitsvorfällen (A.16):** Proaktive Identifikation und Lösung von Sicherheitsvorfällen
- **Notfallmanagement (A.17):** Planung und Reaktion auf unerwartete Ereignisse
- **Compliance (A.18):** Einhaltung gesetzlicher und regulatorischer Anforderungen

Neben den technischen Sicherheitsstandards berücksichtigen wir umfassend die Anforderungen der **Datenschutz-Grundverordnung (DS-GVO)**, um Vertraulichkeit, Integrität und Verfügbarkeit personenbezogener Daten zu gewährleisten:

Vertraulichkeit:

- **Zutrittskontrolle:**
  Zugriff auf sensible Anlagen durch abgestufte Berechtigungen, Alarmanlagen und Überwachung
- **Zugangskontrolle:**
  Schutz vor unbefugter Systembenutzung durch Passwortrichtlinien und Firewalls
- **Zugriffskontrolle:**
  Differenzierte Zugriffsrechte, Audit Trails und geschützte Deployments
- **Trennungskontrolle:**
  Strikte Trennung von Daten nach Verarbeitungszwecken und isolierte Datenbanken
- **Pseudonymisierung:**
  Schutz personenbezogener Daten durch Anonymisierungstechniken

Integrität:

- **Weitergabekontrolle:**
  Nachvollziehbarkeit der Datenweitergabe
- **Eingabekontrolle:**
  Logging von Änderungen und Zugriffen auf personenbezogene Daten

Verfügbarkeit und Belastbarkeit:

- **Verfügbarkeitskontrolle:**
  Schutz vor Datenverlust durch Backups und Redundanz
- **Wiederherstellbarkeit:**
  Regelmäßige Tests der Datenwiederherstellung

Regelmäßige Überprüfung und Evaluierung:

- Datenschutz-Management
- Incident-Response-Management
- Datenschutzfreundliche Voreinstellungen
- Auftragskontrolle bei Dritten

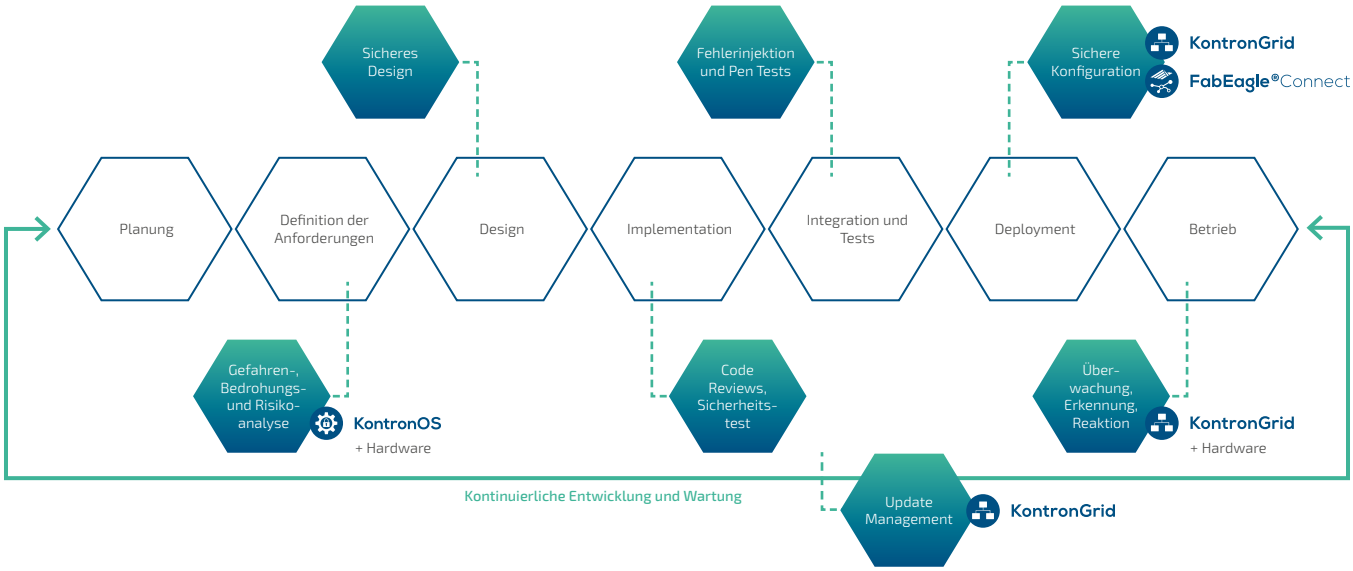
IEC 62443

Sicherheit ist ein zentrales Thema für industrielle Automatisierungssysteme. Die Normenreihe IEC 62443 bietet einen klaren Rahmen, um Cybersecurity-Risiken gezielt zu minimieren. Kontron geht hier einen wichtigen Schritt: Aktuell wird unser Entwicklungsprozess nach IEC 62443-4-1 zertifiziert. Die Zertifizierung soll im zweiten Quartal von 2025 abgeschlossen sein.

Warum ist die IEC 62443 so relevant? Die Norm ist speziell auf die Anforderungen industrieller Umgebungen zugeschnitten und ergänzt die bekannten Vorgaben der ISO 27001. Sie deckt spezifische Sicherheitsaspekte ab, die in der Operational Technology (OT) besonders wichtig sind. Mit der Einhaltung dieser Standards stellen wir sicher, dass unsere Produkte nicht nur die regulatorischen Anforderungen erfüllen, sondern auch die Herausforderungen im industriellen Betrieb sicher und zuverlässig meistern.

Mit unseren Softwarelösungen wie dem IoT-Device-Management KontronGrid und dem sicheren, linuxbasierten Betriebssystem KontronOS unterstützen wir Ihre IoT-Infrastruktur an zwei entscheidenden Punkten:

- Schwachstellen werden frühzeitig erkannt und adressiert
- Potenzielle Bedrohungen werden effektiv und sicher abgewehrt



Sicherheitsfunktionen unserer Lösungen im Detail

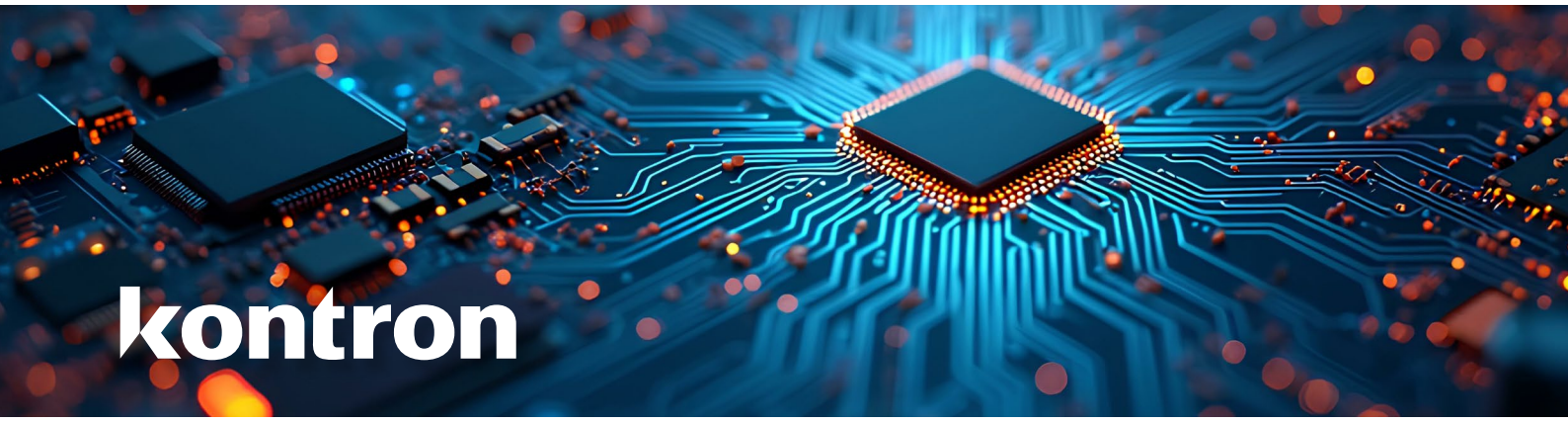
Analyse von Gefahren, Bedrohungen und Risiken

Sicherheit fängt bei der Analyse an. Deshalb setzen wir auf regelmäßige Schwachstellenanalysen, um potenzielle Risiken frühzeitig zu erkennen. Werden kritische Schwachstellen entdeckt, stellen wir schnellstmöglich Patches bereit. Gleichzeitig sorgen Fallback-Mechanismen dafür, dass die Datenintegrität und -verfügbarkeit auch in Krisensituationen erhalten bleiben.

Sichere Konfiguration
Unsere Produkte werden standardmäßig mit sicheren Voreinstellungen ausgeliefert. Damit sind sie optimal gegen gängige Angriffsmuster geschützt – ohne zusätzlichen Konfigurationsaufwand.

Überwachen, Erkennen und Reagieren
Durch kontinuierliches Monitoring und gezielte Sicherheitsüberwachung können wir Bedrohungen in Echtzeit erkennen und sofort Gegenmaßnahmen einleiten.

Update-Management
Sicherheitsupdates werden bei uns automatisiert durchgeführt. So stellen wir sicher, dass Ihre Systeme immer auf dem neuesten Stand sind und bekannte Sicherheitslücken schnell geschlossen werden.



Sicherheit auf Hardware-Ebene

Sicherheitsmaßnahmen auf Hardwareebene sind genauso entscheidend wie auf Softwareebene. Hier sind einige Beispiele aus unserem Portfolio:

- **Zertifikats- und Verschlüsselungskonfiguration:**
  Sichere BIOS-Lösungen, Secure Boot und die Generierung sicherer Schlüssel bieten Schutz direkt auf der Hardwareebene.
- **Incident Handling:**
  Unser Product Security Incident Response Team (PSIRT) überwacht Sicherheitsvorfälle, führt regelmäßige Vulnerability Checks durch und gleicht diese mit relevanten Datenbanken ab.
- **Systematische Sicherheitsupdates:**
  Zertifizierte Produkte erhalten zyklische BIOS-Updates und Patches, um Sicherheitslücken langfristig zu schließen.
- **Sicherer Entwicklungsprozess:**
  Der gesamte Fertigungsprozess wird kontrolliert, von der Produktkonfiguration bis zur Zugriffskontrolle. Unsere Prozesse sind vollständig transparent und entsprechen den Anforderungen der IEC 62443-4-1.

Betriebssicherheit mit KontronOS

Das gehärtete Betriebssystem KontronOS, basierend auf Yocto-Linux®, ist speziell darauf ausgelegt, Kundenapplikationen auf Intel® x86- oder Arm®-basierten Edge Devices sicher und zuverlässig zu betreiben. Vorinstalliert und optimiert für vernetzte Systeme, bietet KontronOS höchste Sicherheit und Aktualität – selbst in kritischen Umgebungen. Zusätzlich überzeugt es mit einer Vielzahl an Sicherheitsfeatures:

SBOM (Software Bill of Materials):

Eine vollständige Auflistung der genutzten Bibliotheken und Module sorgt für Transparenz und ermöglicht das Nachverfolgen von Sicherheitslücken, bspw. über:

- Eine Stückliste der verwendeten Bibliotheken und Module sowie deren Versionen.
- Eine systematische Dokumentation und Nachverfolgung von Sicherheitslücken, die in diesen Drittanbieter-Komponenten identifiziert wurden.

Die SBOM ermöglicht es uns, bekannte Sicherheitsprobleme proaktiv zu identifizieren, Patches bereitzustellen oder alternative Lösungen zu ermitteln.

Auto-Detection und Schwachstellenscans:

Mit diesem Feature werden Sicherheitslücken und Patching in Echtzeit automatisch erkannt. Unsere Produkte werden kontinuierlich gegen bekannte Sicherheitslücken geprüft, indem wir sie mit Datenbanken wie CVE und CWE abgleichen. Dieser Prozess umfasst:

- **Beschreibung der Sicherheitsprobleme:**
  Zuordnung der identifizierten Sicherheitslücken zu den betroffenen Komponenten.
- **Behebungshinweise:**
  Bereitstellung von Lösungen wie Patches oder Konfigurationsänderungen zur Behebung der Schwachstellen.

Durch das automatisierte Tracking von Drittanbieter-Software stellen wir sicher, dass wir immer über die aktuellen Sicherheitsprobleme und Patches informiert sind.

Netzwerksicherheit:
Die integrierte Firewall und Netzwerkzonen schützen vor unautorisierten Zugriffen und segmentieren das Netzwerk für zusätzliche Sicherheit.

Regelmäßige System-Updates und Patching:
Sicherheitspatches werden kontinuierlich eingespielt, um bekannte Schwachstellen zu schließen.

Redundanz und Fallback-Mechanismus:
Dieses Feature stellt die Verfügbarkeit bei Systemausfällen oder Angriffen sicher.

Minimalistische Betriebssystemkonfiguration:
Das Betriebssystems wird auf essenzielle Kernel-Konfigurationen reduziert und automatische Updates sind deaktiviert, was zur Minimierung potentielle Sicherheitsrisiken beiträgt. Um die Sicherheit zusätzlich zu erhöhen werden explizite Dateiberechtigungen und gerätespezifsche Logins vergeben.

Deaktivierung unnötiger Ports:
Durch das Abschalten von nicht benötigten Diensten und Ports sowie die Unterbindung von Rootzugriffen können Angriffsflächen auf die Sicherheitssysteme verringert werden.

Prinzip der minimalen Rechtevergabe:
Nur die notwendigsten Rechte werden vergeben, was das Risiko von Insider-Bedrohungen oder Fehlkonfigurationen minimiert.

Zentralisierte Protokollierung:
Logging von Benutzeraktivitäten und Dokumentation der eingesetzten Open Source Lizenzen.

- **Verschlüsselung:**
  Sämtliche Verbindungen (zur Cloud, Weboberfläche, etc.) sind mit State-of-the-Art Algorithmen verschlüsselt.
- **Integrität:**
  Durch Implementation von UEFI Secure Boot bzw. HAB Secure Boot ist die Integrität der Firmware bzw. OS-Software gewährleistet.
- **Applikationsisolation:**
  Strikte Trennung von Betriebssystem und Anwendung durch Verwendung von Containervirtualisierung (Docker) und /oder Mandatory Access Control System (AppArmor) erhöht die Betriebssicherheit.

Sicheres IoT-Device-Management mit KontronGrid

KontronGrid ist unsere IoT-Device-Management-Lösung für Edge Devices. Sie vereinfacht den Einsatz von Container-Applikationen, automatisiert Updates, überwacht Geräte und bietet schnellen Remote-Support – ideal für die effiziente Verwaltung global verteilter Geräteflotten.

Auch auf der Ebene des IoT-Device-Managements setzen wir auf Sicherheit. Mit KontronGrid bieten wir:

Update Management
Unterstützt den kontinuierlichen Entwicklungs- und Wartungszyklus durch automatisierte Updates.

Benutzer- und Zugriffsverwaltung
Multi-Faktor-Authentifizierung (MFA), separate Rechtemanagement für jeden Dienst und granulare Benutzergruppen sorgen für sichere Zugriffssteuerung.

End-to-End Verschlüsselung
Alle Daten werden durch SSL/TLS 1.2 verschlüsselt übertragen, um eine sichere Kommunikation zwischen den Diensten zu gewährleisten. Dabei sind die Kundendaten nach AES128 verschlüsselt.

Geräte- und Serviceauthentifizierung
Jedes einzelne Gerät ist strikt getrennt und jeder Dienst wird authentifiziert, was unerlaubte Zugriffe verhindert. Der Aufbau einer Remoting Verbindungen erfordert eine zweistufige Authentifizierung. Der SSH-Tunnel ist RSA 2048bit verschlüsselt. Ungenutzte VPN-Verbindungen werden automatisch beendet und Tunneling ins Kundennetzwerk ist optional.

Logging
Alle sicherheitsrelevanten Ereignisse werden lückenlos protokolliert, um eine schnelle Reaktion bei Vorfällen zu ermöglichen.

Test und Validierung

Sicherheitsrelevante Funktionen unserer Produkte werden systematisch und bevorzugt automatisiert getestet. Hier einige Beispiele:

- **HTTPS-Sicherheit von KontronGrid:**
  Vor jedem Rollout wird ein umfassender Test durchgeführt, mit der Zielvorgabe eines A+ Ratings.
- **REST-API Tests:**
  Mittels Tools wie Apache JMeter und Azure DevOps testen wir Szenarien für verschiedene Benutzertypen (Admin, ungültiger Benutzer, regulärer Benutzer), um sicherzustellen, dass alle REST-Calls sicher und korrekt funktionieren.
- **Risikobasierte Tests:**
  Zusätzlich führen wir risikobasierte Tests durch, bei denen Risikogruppen und -levels anhand der Parameter Signifikanz, Auftretenshäufigkeit, Reproduzierbarkeit und Akzeptanz definiert werden.

Um potenzielle Sicherheitsbedrohungen frühzeitig zu erkennen und gezielt entgegenzuwirken, setzen wir auf die strukturierte Modellierung von Gefahren nach dem S.T.R.I.D.E Framework. Dieses Framework hilft dabei, Sicherheitsrisiken systematisch zu analysieren und passende Gegenmaßnahmen zu entwickeln. Die sechs zentralen Bedrohungskategorien sind:

- **Spoofing:**
  Fälschung der Identität von Benutzern oder Diensten
- **Tampering:**
  Manipulation von Daten oder Systemen
- **Repudiation:**
  Abstreiten von durchgeführten Aktionen, ohne dass dies nachgewiesen werden kann
- **Information Disclosure:**
  Unbefugte Offenlegung von sensiblen Informationen
- **Denial of Service:**
  Verhinderung des Zugriffs auf Dienste durch Überlastung
- **Escalation of Privilege:**
  Erhöhung der Zugriffsrechte durch Ausnutzung von Sicherheitslücken

Diese systematische Gefahrenmodellierung bildet die Basis für unsere Sicherheitsstrategien und fließt in die Entwicklung sowie den Betrieb unserer Produkte ein.

6 Warum kompliziert, wenn es auch einfach geht?

Entwicklungsteams von IoT-Lösungen möchten sich vollständig auf die Anwendungsentwicklung konzentrieren, ohne durch manuelle oder komplexe Prozesse beim Deployment aufgehalten zu werden. Sie erwarten einen sicheren und automatisierten Deployment-Prozess, der auch für Testzwecke reibungslos funktioniert. IT-Verantwortliche legen dabei besonderen Wert auf manipulationssichere IoT-Geräte, die zuverlässig vor externen Bedrohungen geschützt sind. Selbst im Problemfall sollte es Angreifer*innen nur mit erheblichem Aufwand gelingen, auf ein Gerät zuzugreifen

Produktmanager*innen von IoT-Lösungen streben danach, den Anwendern ein erstklassiges Produkterlebnis zu bieten. Dies bedeutet, dass hohe Zuverlässigkeit, maximale Verfügbarkeit und stets aktuelle Softwareanwendungen gewährleistet sein müssen – von der Benutzeroberfläche bis zur Echtzeitdatenanalyse direkt am Edge. Chief Digital Officers erkennen darüber hinaus ein erhebliches Monetarisierungspotenzial, das über die reine physische Lösung hinausgeht. Kundenspezifische Anwendungen und die steigende Nachfrage nach Individualisierung erfordern eine flexible Skalierbarkeit und Modularität, die den gesamten Produktlebenszyklus umfassen.

Mit KontronGrid, unserer IoT-Device-Management-Lösung für Edge Devices, bieten wir eine effiziente, sichere und skalierbare Lösung für das IoT-Device-Management und die Fernüberwachung von IoT-Lösungen. Mit Funktionen wie sicherem Datenaustausch, kontinuierlichem Deployment und End-to-End-Verschlüsselung reduziert KontronGrid die IT-Komplexität, steigert die Kosteneffizienz und bietet höchste Sicherheitsstandards. Es unterstützt die Monetarisierung durch digitale Services, beschleunigt Produktentwicklungszyklen und optimiert die Kontrolle über global verteilte Geräte, indem es die Bereitstellung von Container-Applikationen vereinfacht, Flottenupdates automatisiert, Geräte überwacht und schnellen Remote-Support für die effiziente Verwaltung weltweiter Geräteverbindungen ermöglicht.

Komplettangebot aus einer Hand: Integrierte Hardware und Software – Die Antwort lautet ManagedEdge IoT-Bundle

Die Anwender profitieren von einem Komplettangebot aus einer Hand, das eine nahtlose Integration mit perfekt aufeinander abgestimmter Hard- und Software gewährleistet. Neben der Flottenmanagementlösung wird das Gesamtpaket durch anwendungsspezifische Hardware und ein darauf abgestimmtes, sicheres Betriebssystem auf Linux®-Basis abgerundet.

Alles ist in einem IoT-Bundle zusammengefasst, was zahlreiche Vorteile bietet: Die Geräte sind dank vorinstallierter und vorkonfigurierter Software sofort einsatzbereit, was einen schnellen Einstieg in das globale Gerätemanagement ermöglicht.

Gleichzeitig ermöglicht das IoT-Bundle ein kosteneffizientes Prototyping und erleichtert die Entwicklung und den Rollout neuer Anwendungen durch optimierte Prozesse. Nutzer*innen profitieren zudem von einer beschleunigten Implementierung und der Möglichkeit, ihre Lösungen flexibel und effizient zu skalieren. Damit bietet das IoT-Bundle eine zukunftssichere Basis für industrielle Anwendungen.

Das Portfolio an kompatibler Kontron Hardware mit darauf abgestimmtem sicheren Betriebssystem wächst kontinuierlich und umfasst aktuell folgende Standardserien:

- KBox A-Serie
- SOM AL-i-Serie

Kundenspezifische Hardware gehört zu den Kernkompetenzen von Kontron und ermöglicht Komponenten- und Geräteherstellern eine besonders hohe Flexibilität und Anpassungsfähigkeit an ihre Zielapplikation. Individuelle Anforderungen an Hard- und Software können so problemlos umgesetzt werden.

7 Checkliste und Entscheidungshilfe – Build or Buy?

Die folgende Checkliste hilft Ihnen schnell und einfach, die wichtigsten Kriterien für Ihre Entscheidung zu beurteilen und auf einen Blick zu sehen, welcher Ansatz für Ihr Unternehmen am besten geeignet ist: Eigenentwicklung oder fertige Lösung. Gehen Sie Punkt für Punkt durch und bewerten Sie, was für Ihre Situation entscheidend ist.

	Build (In-House-Entwicklung)	Buy (Drittanbieter-Lösung)
Zeit- und Markt-einführungsdrang **Brauchen Sie schnell eine Lösung auf dem Markt?**	☐ **Nein:** Wenn Sie Zeit haben, können Sie eine eigene Lösung entwickeln, die langfristig speziell auf Ihre Bedürfnisse zugeschnitten ist.	☐ **Ja:** Eine fertige Lösung bietet schnelle Integration und sofortige Betriebsbereitschaft.
Kostenplanung und Budget **Können Sie hohe Anfangs-investitionen tätigen oder möchten Sie kalkulierbare, laufende Kosten?**	☐ **Investitionsbereitschaft:** Eigenentwicklung erfordert anfangs höhere Ausgaben für Entwicklung, Infrastruktur und Personal, könnte aber langfristig wirtschaftlicher sein.	☐ **Kalkulierbare, laufende Kosten bevorzugt:** Die fertige Lösung bietet planbare Lizenz- oder Abonnementgebühren und reduziert hohe Anfangskosten.
Interne Ressourcen und Expertise **Haben Sie ein internes Team mit der notwendigen Expertise für die Entwicklung und Wartung einer IoT-Plattform?**	☐ **Ja, wir haben die nötigen Fähigkeiten:** Eigenentwicklung bietet Flexibilität, aber auch eine größere Verantwortung für die Wartung und Updates.	☐ **Nein, unsere Ressourcen sind begrenzt:** Die fertige Lösung reduziert die Belastung interner Teams und übernimmt Wartung, Support und Sicherheit.
Flexibilität und Anpassung **Benötigen Sie eine sehr spezifische Anpassung und tiefgreifende Integration in bestehende Systeme?**	☐ **Ja, eine maßgeschneiderte Lösung ist entscheidend:** Eine Eigenentwicklung erlaubt vollständige Anpassung an Ihre Anforderungen, jedoch auf Kosten von Zeit und Ressourcen.	☐ **Nein, eine flexible Standard-lösung reicht aus:** Eine fertige Lösung bietet oft ausreichende Anpassungsmöglichkeiten über APIs und Konfiguration, ohne den Entwicklungsaufwand.

	Build (In-House-Entwicklung)	Buy (Drittanbieter-Lösung)
Sicherheit und Compliance **Ist Sicherheit für Ihre IoT-Lösung ein kritischer Punkt?**	☐ **Ja, und wir können dies intern sicherstellen:** Wenn Sie ein starkes internes Sicherheitsteam haben, könnte die Eigenentwicklung mit maßgeschneiderten Sicherheitsprotokollen von Vorteil sein.	☐ **Ja, aber wir bevorzugen bewährte Sicherheitssysteme:** Fertige Lösungen kommen mit integrierten Sicherheitsprotokollen, kontinuierlichen Updates und CVE-Überwachung.
Skalierbarkeit und Wachstum **Planen Sie in naher Zukunft stark zu wachsen und Ihre IoT-Flotte global zu skalieren?**	☐ **Nein, unser Wachstum wird schrittweise erfolgen:** In diesem Fall könnte die Eigenentwicklung mit angepasster Skalierbarkeit langfristig vorteilhaft sein.	☐ **Ja, wir erwarten schnelles Wachstum:** Eine fertige Lösung ist von Anfang an auf Skalierbarkeit ausgelegt und unterstützt globale Expansion ohne großen Zusatzaufwand.
Wartung und langfristiger Support **Sind Sie bereit, sich langfristig um die Wartung und das Patchen der Lösung zu kümmern?**	☐ **Ja, wir können Wartung intern abdecken:** Die Eigenentwicklung erfordert ein engagiertes Team für kontinuierliche Wartung, Bugfixes und Updates.	☐ **Nein, wir wollen uns nicht um Wartung kümmern:** Die fertige Lösung bietet integrierten Support und regelmäßige Updates, was den Aufwand minimiert.
Technologische Abhängigkeit und Unabhängigkeit **Möchten Sie volle Kontrolle über die Lösung oder sind Sie bereit, teilweise abhängig von einem Drittanbieter zu sein?**	☐ **Vollständige Kontrolle ist entscheidend:** Die Eigenentwicklung bietet Ihnen vollständige Unabhängigkeit, bringt aber die Verantwortung für alle Updates und Weiterentwicklungen mit sich.	☐ **Wir bevorzugen eine bewährte Drittanbieterlösung:** Mit einer fertigen Lösung können Sie sich auf die Expertise des Anbieters verlassen, müssen aber möglicherweise Kompromisse bei der Anpassung machen.
Risiko- und Innovationsmanagement **Möchten Sie das Risiko einer Neuentwicklung auf sich nehmen, um potenziell langfristig Wettbewerbsvorteile zu erzielen?**	☐ **Ja, wir sind bereit, Risiken einzugehen:** Die Eigenentwicklung kann innovativer sein, birgt jedoch das Risiko von Verzögerungen oder Kostenüberschreitungen.	☐ **Nein, wir möchten ein bewährtes, sicheres System:** Eine fertige Lösung bietet geringeres Risiko, da sie bereits getestet und optimiert ist.

8 Fazit: „Build or Buy“ als zentrale Frage

Wenn Sie Schnelligkeit, geringe Anfangsinvestitionen, externe Expertise und Sicherheit aus einer Hand bevorzugen, empfehlen wir das **ManagedEdge IoT-Bundle**. Es bietet eine sofort verfügbare, skalierbare und sichere IoT-Management-Plattform, die es Ihnen ermöglicht, sich auf Ihre Kernanwendungen zu konzentrieren. Entscheiden Sie sich hingegen für vollständige Anpassung, langfristige Kontrolle und den Einsatz interner Ressourcen, könnte die Eigenentwicklung vorteilhaft sein – bedenken Sie dabei jedoch die damit verbundenen Zeit-, Budget- und Risikofaktoren.

Eine fertige Lösung aus Hardware und Software wie KontronGrid und KontronOS erlaubt es Unternehmen, sich auf Kernanwendungen und Geschäftserfolge zu konzentrieren, während die Verwaltung, Sicherheit und Skalierbarkeit der IoT-Flotte übernommen werden. Sie sparen sich die Komplexität des Edge-Device-Managements und profitieren von einer sofort einsatzbereiten, sicheren und skalierbaren Lösung. Gleichzeitig entlastet ein solches Komplettangebot Unternehmen von technischen und operativen Aufgaben, indem es reibungslose Integration, konstante Updates und optimierte Verwaltung bietet – ohne die Risiken und hohen Kosten einer Eigenentwicklung.

Mit einer bewährten IoT-Device-Management-Lösung lassen sich steigende Anforderungen an Innovation und Sicherheit im IoT-Umfeld pragmatisch bewältigen. Unternehmen können nicht nur die Zuverlässigkeit ihrer IoT-Geräte sicherstellen, sondern auch flexibel auf zukünftige Anforderungen reagieren.

Build or Buy? – Lassen Sie uns gemeinsam herausfinden, welcher Weg für Ihr Unternehmen die optimale Lösung ist. Klicken Sie hier für eine individuelle Beratung!

Über Kontron AIS GmbH

Die Kontron AIS GmbH setzt seit über 30 Jahren den Benchmark in industrieller Softwareentwicklung. Ein erfahrenes Team von mehr als 250 Mitarbeiter*innen steht für bewährte Softwareprodukte und maßgeschneiderte Digitalisierungslösungen. Damit gehen Maschinen- und Anlagenbauer sowie Fabrikbetreiber neue Wege in der Automatisierung und sichern sich langfristig Wettbewerbsvorteile. Gemeinsam mit den Kunden implementiert Kontron AIS weltweit intelligente Digitalisierungsstrategien für die smarte Fertigung von morgen.

Als Tochterunternehmen der Kontron AG bietet Kontron AIS ganzheitliche IoT-Lösungen aus einer Hand – von Hardware bis Software. Durch unser globales Netzwerk stellen wir lückenlose Projektbetreuung, Service und weltweiten Support sicher.

Weitere Informationen: **www.kontron-ais.com**

Firmenkontakt
Kontron AIS GmbH | Otto-Mohr-Str. 6 | 01237 Dresden | +49 (0) 351 2166 0 | sales@kontron-ais.com