



SAP-LEITFADEN

SAP DM Shopfloor Integration

April 2026

Autoren	4
Abbildungsverzeichnis	4
Abkürzungsverzeichnis	5
Einleitung	8
Abgrenzung und Geltungsbereich	8
1 Theoretische Grundlagen	9
1.1 Protokollbeschreibungen	9
1.2 Datenstrukturen, Datenmodell, Datenprotokolle	10
2 Anwendungsfälle	15
2.1 Ergebnis-Reporting	16
2.2 Prozess-Reporting	17
2.3 Prozessverriegelung	19
3 Architekturübersicht	21
3.1 Referenzarchitektur	22
3.1.1 Integrationsschicht (Middle Layer)	23
3.1.2 Edge-Connectivity-Software	25
3.1.3 Erfahrungen und Herausforderungen bei der Einführung von Edge- Lösungen	26
3.1.4 Beispiele für Edge-Connectivity-Software und Anbieter	27
3.1.5 Handlungsempfehlung: Edge-Connectivity-Software	28
3.2 Hardware für Maschinenanbindung	29
4 Skalierung	30
4.1 Template-Ansatz	30
4.1.1 Datenstrukturierung im Unified Namespace	31
4.1.2 Governance und organisatorische Einbindung	32
4.1.3 Infrastruktur – Business-Continuity und Risikobewertung	33
4.2 Anlagenbeschaffung	34
4.2.1 Beschaffungsrichtlinie und Integration in Betriebsprozesse	34
4.2.2 Standardisierung von Kommunikationsprotokollen	35
4.2.3 Standardisierung der auszutauschenden Datensätze	35
4.3 Redundanzen	38
4.3.1 Redundanz und Hochverfügbarkeit	38
4.3.2 Logging und Monitoring	38
4.3.3 Recovery-Time-Objective (RTO) und Recovery-Point-Objective (RPO).....	39
4.3.4 Service-Levels und Betriebsorganisation	39

4.3.5	Rollen im Service-Level-Modell	39
5	Betrieb	40
5.1	Backup und Restore	40
5.2	Betriebskonzept.....	41
5.2.1	Vorfall tritt auf (Level 0).....	42
5.2.2	Lokale Analyse und Bearbeitung durch Key-User und Power-User (Level 1).....	42
5.2.3	Eskalation an zentrale IT / SAP Support (Level 2).....	42
5.2.4	Behebung und Rückmeldung der Störung	42
5.2.5	Sicherstellung des Systembetriebs durch Bereitschaftskonzept... ..	42
5.3	Rollenkonzept.....	43
5.3.1	Fachliche Definition der Anwenderrollen.....	43
5.3.2	Abgleich mit SAP-DM-Standardrollen auf der SAP BTP	43
5.3.3	Technische Ausgestaltung der Rollen und Benutzerzuordnung	44
5.3.4	Dokumentation und Governance	44
6	Fazit und nächste Schritte	44
6.1	Zusammenfassung der zentralen Themen	45
6.2	Kernaussagen	45
6.3	Empfehlungen für die nächsten Schritte.....	46
6.4	Ausblick	47

Autoren

Kürzel	Name	E-Mail	Unternehmen
ABu	Andreas Busse	andreas1.busse@zeiss.com	ZEISS AG
DNo	Dorothea Nochelski	dnochelski@deloitte.de	Deloitte Consulting GmbH
PRe	Peter Reu	peter.reu@zeiss.com	ZEISS AG
MSI	Marc Slonek	marc.slonek@zeiss.com	ZEISS AG
SPi	Sascha Pirke	sascha.pirke@mtu.de	MTU Aero Engines AG
AKr	Andreas Kronier	akronier@schueco.com	Schüco International KG
MWa	Martin Walther	mwaltherr@schueco.com	Schüco International KG

Abbildungsverzeichnis

Abbildung 1 Datentyp-Tabelle nach VDMA 40001-1:2025-07	11
Abbildung 2 Beispiel Euromap 84 V2.0	15
Abbildung 3 Ableitung der ISA-95-Pyramide zum Unified Namespace	21
Abbildung 4 Referenzarchitektur	23
Abbildung 5 Beispiel mehrstufiger Support-Prozess	42

Abkürzungsverzeichnis

Abkürzung	Bedeutung
AKS	Azure Kubernetes Service (Kubernetes-Angebot von Microsoft)
BCM	Business Continuity Management
BTP	SAP Business Technology Platform
CE	Conformité Européenne (CE-Kennzeichnung, EU-Konformität)
DM / SAP DM	SAP Digital Manufacturing
Docker Swarm	Container-Orchestrierungs-Tool von Docker
ERP	Enterprise Resource Planning
EWM	Extended Warehouse Management (SAP-Modul für Lagerverwaltung)
IoT, IIoT	Internet of Things, Industrial Internet of Things
IT	Information Technology (Informationstechnik)
JSON	JavaScript Object Notation (Datenformat)
KPI	Key Performance Indicator (Leistungskennzahl)
Kubernetes	Open-Source-Plattform zur Container-Orchestrierung
MQTT	Message Queuing Telemetry Transport

Verzeichnisse

OData	Open Data Protocol (REST-basiertes Protokoll)
OEE	Overall Equipment Effectiveness (Gesamtanlageneffektivität)
OPC UA	OPC Unified Architecture
OT	Operational Technology (Betriebstechnologie)
PoC	Proof of Concept (Machbarkeitsnachweis)
ProdCon	SAP Production Connector
REST	Representational State Transfer (Architekturstil für Web-APIs)
RPO	Recovery-Point-Objective (maximal tolerierter Datenverlustzeitraum)
RTO	Recovery-Time-Objective (maximal tolerierte Wiederanlaufzeit)
SCADA	Supervisory Control and Data Acquisition
SLA	Service-Level-Agreement (Dienstgütevereinbarung)
SPS	Speicherprogrammierbare Steuerung (PLC)
UMATI	Universal Machine Technology Interface (Standard für Maschinenkommunikation)
UNS	Unified Namespace (zentraler, einheitlicher Datenraum auf Basis eines Brokers)

Verzeichnisse

XML	Extensible Markup Language (Datenformat)
-----	--

Einleitung

Die Integration von SAP Digital Manufacturing in die bestehende Produktionslandschaft ist ein Schlüsselfaktor für eine erfolgreiche Digitalisierung der Fertigung. Nur wenn SAP DM sauber in Shopfloor-Systeme und -Prozesse eingebettet wird, lassen sich Transparenz, Steuerbarkeit und Effizienzgewinne nachhaltig realisieren.

Dieser Best-Practice-Guide wurde in Zusammenarbeit mit den DSAG-Mitgliedsunternehmen entwickelt und dient als praxisnahe Orientierungshilfe für die erfolgreiche Einführung und Verwendung von SAP DM auf dem Shopfloor. Er basiert auf den Erfahrungen und Erkenntnissen aus realen Projekten, die Unternehmen bei der Integration in ihre Produktionsumgebungen gesammelt haben.

Unser Ziel ist es, Ihnen anhand konkreter Beispiele und bewährter Methoden wertvolle Einblicke zu vermitteln, die Sie bei der Planung, Implementierung und Nutzung von SAP DM unterstützen. Dabei beleuchten wir nicht nur technische Aspekte, sondern auch organisatorische und prozessuale Herausforderungen, um eine ganzheitliche Perspektive zu bieten.

Abgrenzung und Geltungsbereich

Der Fokus dieses Leitfadens liegt auf der fachlichen und technischen Integration von Produktionsanlagen mit SAP Digital Manufacturing – insbesondere auf Architektur, Datenstrukturen und -modellen, Skalierung, Templates, Betriebsorganisation und Beschaffungsaspekten.

Sicherheitsaspekte (z. B. Netzwerksegmentierung, Zugriffsschutz, Verschlüsselung, Berechtigungskonzepte) werden bewusst nicht im Detail behandelt. Für die konkrete Ausgestaltung von Security und Compliance sind die jeweils gültigen Unternehmensrichtlinien sowie die herstellereigenen Sicherheitsempfehlungen und Normen heranzuziehen.

1 Theoretische Grundlagen

1.1 Protokollbeschreibungen

Die Integration von SAP Digital Manufacturing (SAP DM) in die Produktionsumgebung erfordert eine effiziente und stabile Kommunikation zwischen den IT-Systemen und den Shopfloor-Geräten, Maschinen und Steuerungen.

Moderne Fabriken sind oft durch Heterogenität gekennzeichnet: Unterschiedliche Maschinenhersteller, Steuerungen (z. B. SPS-Systeme) und Datenformate erfordern flexible und interoperable Schnittstellen. Hier kommen standardisierte Protokolle ins Spiel, die eine übergreifende Kommunikation ermöglichen. Ein passendes Protokoll unterstützt sowohl die technischen Anforderungen der Maschinen als auch die Geschäftsprozesse in der IT-Landschaft.

Dieser Abschnitt des Best-Practice-Guides bietet eine Übersicht über die wichtigsten Protokolle und Datenkommunikationsstandards, die bei der Integration von SAP DM in den Shopfloor von Bedeutung sind. Im Fokus stehen dabei sowohl klassische Fertigungsprotokolle als auch moderne IIoT-Ansätze, die zunehmend an Bedeutung gewinnen.

Zu den relevanten Protokollen gehören unter anderem:

- **OPC UA (Open Platform Communications Unified Architecture):** Ein moderner, herstellerunabhängiger Standard für die Kommunikation auf dem Shopfloor und die sichere Übermittlung von Daten an SAP DM.
- **MQTT (Message Queuing Telemetry Transport):** Ein leichtgewichtiges Protokoll, das speziell für IIoT-Anwendungen und die Übertragung von Sensordaten entwickelt wurde.
- **REST- und OData-basierte Services:** Wichtige Standards für die Verbindung zwischen SAP DM und Fertigungsgeräten, insbesondere zur API-basierten Integration.
- **Proprietäre Steuerungsschnittstellen** (z. B. Siemens S7, Mitsubishi MC Protocol): Häufig eingesetzte Protokolle, die spezifische Maschinensteuerungen ansprechen können.

Neben diesen branchenübergreifend nutzbaren Protokollen gibt es auch fertigungsdomänenspezifische Standards (Semantik), die auf die besonderen Anforderungen einzelner Industrien zugeschnitten sind. Beispiele hierfür sind EUROMAP (Kunststoffindustrie), Weihenstephan (Food and Beverage) oder SEMI (Halbleiterindustrie).

In diesem Leitfaden konzentrieren wir uns auf übergreifende Protokolle und verzichten bewusst auf eine detaillierte Betrachtung dieser branchenspezifischen Lösungen, da

deren Einsatz in der Regel stark von den besonderen Anforderungen der jeweiligen Industrie abhängt.

Die Auswahl des passenden Protokolls hängt von verschiedenen Faktoren ab, wie Kompatibilität zu bestehenden Shopfloor-Systemen, Echtzeitanforderungen, Datenvolumen und Sicherheitsaspekten. Zudem sollte die langfristige Skalierbarkeit und Zukunftssicherheit des gewählten Standards sorgfältig berücksichtigt werden.

1.2 Datenstrukturen, Datenmodell, Datenprotokolle

Datenmodell:

Zu Beginn eines jeden Integrationsprojekts auf dem Shopfloor steht die Erarbeitung eines Datenmodells und einer Datenstruktur. Dieser Schritt beginnt mit der abstrakten Analyse: Es wird festgelegt, welche Informationen (beispielsweise zu Maschinen, Sensoren oder Messwerten) erfasst werden sollen und wie diese zueinander in Beziehung stehen. Die spätere Speicherung und technische Umsetzung sind in diesem Stadium noch zweitrangig. Ziel ist es, die fachlichen Anforderungen und Zusammenhänge unabhängig von technischen Details logisch abzubilden.

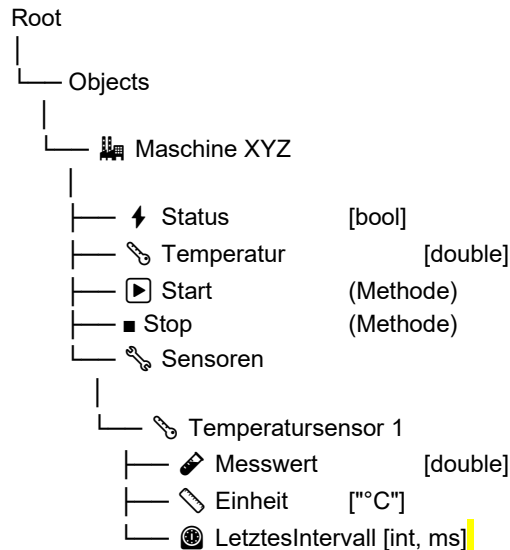
Datenstruktur:

Die Datenstruktur leitet sich i. d. R. aus dem Datenmodell ab und beschreibt, wie diese Informationen technisch organisiert und gespeichert sind.

Eine leistungsfähige Datenstruktur zeichnet sich durch folgende Merkmale aus, die Flexibilität, Verständlichkeit und Skalierbarkeit sicherstellen:

- **Objektorientierte Struktur:**
Im Zentrum steht ein hierarchisches, objektorientiertes Datenmodell. Daten werden in sogenannten Nodes (Knoten) abgebildet, die verschiedene Typen haben können, z. B. Objekt, Variable, Methode oder Datentyp. Über „Referenzen“ sind diese Nodes miteinander verknüpft, sodass komplexe Zusammenhänge und Strukturen abgebildet werden können.
- **Typisierung und Wiederverwendung:**
Durch Typdefinitionen (ObjectTypes, VariableTypes) können wiederverwendbare Modelle für Geräte, Maschinen oder Prozesse geschaffen werden. So ist zum Beispiel ein „Temperatursensor“ als Typ definiert und jede Instanz eines Sensors kann auf diesen Typ referenzieren (siehe Abbildung 1: Type Definition).
- **Klar definierte Metadaten:**
Jede Variable oder jedes Objekt kann mit Attributen wie Name, Beschreibung, Datentyp, Einheit, Engineering-Units und Zugangsbeschränkungen versehen werden, was die Interoperabilität und Lesbarkeit fördert.

- **Namespaces:**
OPC UA verwendet Namespaces zur Verwaltung von Erweiterungen und zur Vermeidung von Namenskonflikten. Dadurch lassen sich individuelle und branchenspezifische Erweiterungen strukturiert integrieren.
- **Beispielhafte Datenmodelle**
- **OPC UA** bietet eine objektorientierte Struktur, zum Beispiel:



Legende der Darstellung:

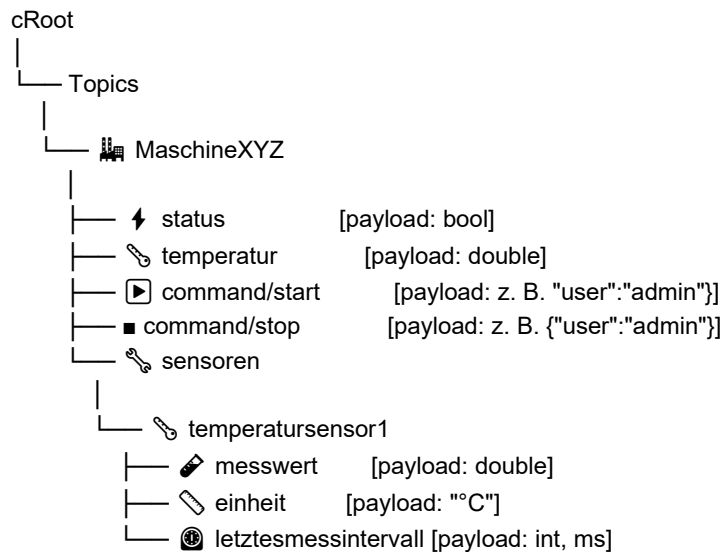
- Objekte sind als Verschachtelungen/Baumknoten abgebildet.
- Variablen sind als "Variable: Datentyp" markiert.
- Methoden sind ausdrücklich als solche bezeichnet.
- Unterobjekte (z. B. Sensoren) befinden sich eine Ebene tiefer.

Table 2 – Type Definition Table

Attribute	Value				
Attribute name	Attribute value. If it is an optional Attribute that is not set "--" will be used.				
References	NodeClass	BrowseName	Data Type	TypeDefinition	Other
ReferenceType name	NodeClass of the TargetNode.	BrowseName of the target Node.	Data Type of the referenced Node, only applicable for Variables.	TypeDefinition of the referenced Node, only applicable for Variables and Objects.	Additional characteristics of the TargetNode such as the ModellingRule or AccessLevel.
NOTE Notes referencing footnotes of the table content.					
Conformance Units					
Name of ConformanceUnit, one row per ConformanceUnit					

Abbildung 1 Datentyp-Tabelle nach VDMA 40001-1:2025-07

- **MQTT** arbeitet Topic-basiert. Die Struktur für Maschine XY könnte so aussehen:



Status- und Temperaturwerte werden als Nutzdaten (Payload) über ihre jeweiligen Topics übertragen. Befehle (wie Start/Stop) erfolgen über spezielle Command-Topics, und Sensorwerte werden separat bereitgestellt.

Fazit:

Eine gute Datenstruktur orientiert sich an klaren, wiederverwendbaren Objekt- und Variablentypen, nutzt Hierarchien, liefert aussagekräftige Metadaten und setzt auf den Namensraummechanismus. Dadurch bleibt das System transparent, erweiterbar und standardisiert – sowohl für einfache als auch für komplexe Anlagen.

1.3 Interaktionsmuster

Diese Interaktion zwischen Maschine und SAP DM ist abhängig von der Art der Kommunikation und der Architektur der Kommunikationsteilnehmer. Das Interaktionsmuster beschreibt meist einen zyklischen, durch Events oder Befehle ausgelösten Datenaustausch zwischen Maschine und der übergeordneten SAP DM, der so gestaltet ist, dass reibungslose Produktion, Transparenz und Rückverfolgbarkeit über alle Fertigungsstufen hinweg möglich sind. Grundsätzlich empfiehlt es sich, möglichst eine einheitliche Struktur im Produktionsumfeld umzusetzen. Dies verringert die Komplexität und trägt zu einer einfachen Verwaltung und Sicherstellung des Betriebs bei.

Typische Interaktionsmuster sind:

1. **Request-Response, anlagengetrieben:** Einzelne Komponenten oder Maschinen stellen Anfragen (z. B. Materialbedarf, Statusbericht) an die SAP DM. Die SAP DM antwortet mit Freigaben, Anweisungen oder Rezepten.
2. **Request-Response, MES-getrieben:** Die SAP DM sendet konkrete Befehle an die Maschinen (z. B. Produktionsstart, Rezeptwechsel). Die Maschine bestätigt die Durchführung oder meldet Fehler zurück.
3. **Event-driven (ereignisgesteuert):** Maschinen oder Komponenten melden Statusänderungen, Produktionsfortschritte oder Störungen automatisch und in Echtzeit an die SAP DM. Dort werden die Daten weiterverarbeitet und Produktionsdaten aktualisiert.
4. **Periodische Datensynchronisation (Datenbereitstellung):** Prozessdaten werden periodisch gesammelt von den Maschinen bzw. Komponenten an die SAP DM übertragen und dort zur Qualitätssicherung, Analyse oder Visualisierung genutzt.

Zur Umsetzung dieser Interaktionsmuster werden oft standardisierte Schnittstellen und Protokolle wie OPC UA, MQTT oder REST-APIs genutzt. Sie sorgen für einen zuverlässigen, sicheren und flexiblen Datenaustausch zwischen Maschinen (oft mit Steuerungen wie SPS) und SAP DM.

Architekturvarianten

Je nach Aufbau der Fertigungslinie und der eingesetzten Maschinen gibt es unterschiedliche Kommunikationsarchitekturen, wie sie hier z. B. mit der EUROMAP 84 Companion Specification beschrieben wird:

- **Variante A:** Jede Maschine bzw. Komponente verfügt über einen eigenen OPC-Server oder MQTT-Broker und kommuniziert direkt mit der SAP DM. Dies ermöglicht eine dezentrale Datenerfassung, erhöht aber die Anzahl der Schnittstellen.
- **Variante B:** Eine zentrale Linie-Steuerung („Line control“) sammelt die Daten aller Komponenten und leitet diese gesammelt an die SAP DM weiter. Die Kommunikation mit den Komponenten kann über OPC UA, MQTT, Feldbus oder EUROMAP 27 erfolgen.
- **Variante C:** Die Linie-Steuerung ist in eine der Hauptkomponenten integriert (z. B. Hauptextruder) und übernimmt von dort die Datensammlung und -weitergabe an die SAP DM.

Diese Strukturen bieten verschiedene Vorteile bezüglich Datenintegrität, Ausfallsicherheit und Administrationsaufwand. Die Wahl der passenden Architektur hängt oft von Größe und Komplexität der Linie sowie der gewünschten Durchgängigkeit der Daten ab.

Rolle von Standards

Die Verwendung von OPC UA Companion Specifications (wie EUROMAP 84) oder MQTT Topic-Konventionen erleichtert die Interoperabilität, da sie standardisierte Datenmodelle und Kommunikationsmethoden bereitstellen. So kann z. B. die SAP DM effizient und flexibel mit unterschiedlichsten Maschinen und Komponenten kommunizieren. Die OPC UA Companion Specification gibt eine praxisnahe Übersicht der üblichen Strukturen. Die folgende Darstellung zeigt dies am Beispiel der Euromap 84 V2.0.

- a) Each component has an own OPC server and is connected directly to an MES
- b) A line control collects all data from the components and forward these to the MES. The exchange between the line control and the components can be realised by OPC UA, but also by other technologies (e.g. field bus, EUROMAP 27)
- c) The line control is included in an extruder

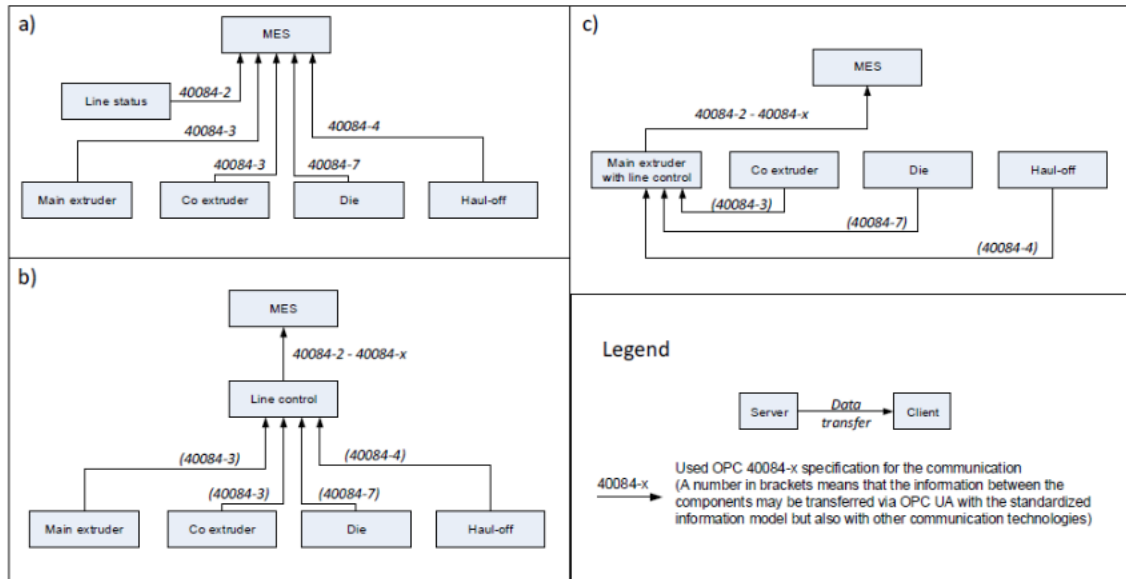


Abbildung 2 Beispiel Euromap 84 V2.0

Fazit

Die Klarheit des Interaktionsmusters und die Wahl einer passenden Architektur sind entscheidend für einen stabilen und transparenten Produktionsprozess. Die dargestellten Varianten zeigen bewährte Möglichkeiten, wie Maschinen, Linie-Steuerung und SAP DM optimal vernetzt werden können.

Zusammenfassend lässt sich festhalten, dass die etablierten, hierarchischen Architekturen weiterhin ihre Berechtigung haben, jedoch den Anforderungen moderner, hochvernetzter Produktionsumgebungen nur begrenzt gerecht werden. Eine zukunftsorientierte Antwort darauf bietet der Unified Namespace, der als moderne Architekturvariante in Kapitel 3 eingehend beschrieben wird.

2 Anwendungsfälle

Dieses Kapitel beschäftigt sich mit den drei wesentlichen Anwendungsfällen im Kontext der Shopfloor-Integration mit SAP DM: Ergebnis-Reporting, Prozess-Reporting und Prozessverriegelung. Jeder dieser Anwendungsfälle bringt spezifische Anforderungen, Interaktionsmuster und Herausforderungen mit sich – von der

Erfassung aggregierter Produktionskennzahlen bis hin zur Kombination von Echtzeitdaten und automatisierten Steuerungsvorgängen.

Für jede der drei Kategorien wird im Folgenden beschrieben:

- welche Arten von Daten erforderlich sind, um den Anwendungsfall zu realisieren,
- wie diese Daten modelliert, verarbeitet und in die Systemarchitektur eingebunden werden
- und welchen konkreten Nutzen die gewonnenen Einsichten und Steuerungsmöglichkeiten für die Produktion haben.

Darüber hinaus werden Herausforderungen wie Latenzzeiten, Datenqualität und Interoperabilität thematisiert, die bei der Umsetzung bewältigt werden müssen.

2.1 Ergebnis-Reporting

Beschreibung der Type:

Beim Ergebnis-Reporting handelt es sich um die Erfassung und Aggregation von Daten zu Produktionsprozessen. Im Mittelpunkt stehen Kennzahlen wie produzierte Menge, Gut-/Schlechteile (Fehlerquote), Auslastung der Maschinen, Produktionsfortschritt und Stillstandszeiten. Ziel ist es, eine Übersicht über die Performance und Effizienz der Anlagen zu geben – oft ein zentraler Teil der Berechnung der OEE (Overall Equipment Effectiveness).

Beispielhafte Use-Cases:

Erfassen der Gesamtanzahl produzierter Teile in einem bestimmten Zeitraum (z. B. pro Schicht, Tag, Stückzahl pro Auftrag).

- Aufzeichnung von Gut- und Schlechteilen zur Bewertung der Produktionsqualität
- Verfolgen von Maschinenlaufzeiten und Stillständen zur Ermittlung der Betriebszeit

Ziele/Business-Nutzen:

- Messung von Produktionsleistung und Qualität (KPI-Tracking, insbesondere OEE)
- Grundlage für Berichte und Dashboards (z. B. in SAP DM)
- Identifikation von Ineffizienzen und Optimierungspotenzialen in der Produktion
- Transparenz über den Produktionsstatus für das Management

Benötigte Maschinendaten und Datenmodell

- Anzahl der produzierten Teile (Zählerstände, Teilesensoren)
- Klassifikation von Gut-/Schlechteilen (z. B. basierend auf Inspektionssystemen)
- Maschinenstatus (produktionsbereit, aktiv, Instandhaltung, Stillstand)
- Zeitstempel

- Serial-/Chargennummer und Auftrag

Datenmenge und -relevanz

- Häufigkeit: Niedrig bis mittel, abhängig von Taktung/Stückzahl und Anlagen
- Volumen je Nachricht: Niedrig bis mittel, abhängig von zu erfassenden Qualitätsparametern
- Echtzeitanforderung: Kurze Latenzen hilfreich, z. B. für Andon-Board, aber eine verzögerte Verarbeitung hat keinen sofortigen Einfluss auf das Produktionsergebnis

Interaktionsmuster: Event-getriebenes Reporting

Maschinen senden regelmäßig aggregierte Statusdaten an die Edge- oder Middle-Layer-Systeme, die dann hoch an SAP DM übertragen werden.

Typische Stolpersteine:

- Inhomogene Maschinendatenerhebung erschwert einheitliche Aggregation.
- Zeitstempel-Synchronisation kann komplex sein (z. B. Abweichungen bei Maschinen-Timern).
- Datenqualität (falsche Sensorwerte oder unzureichende Detailtiefe)
- Latenzen oder Unterbrechungen bei der Datenübertragung; neben Zeitverzug könnte dies auch im Wiederanlauf zu Inkonsistenzen führen. Für Datenzwischenspeicherung s. Folgekapitel.

2.2 Prozess-Reporting

Beschreibung der Type

Im Prozess-Reporting geht es um die Erfassung und Analyse von detaillierten Produktionsdaten (z. B. von Sensoren und Steuerungen auf dem Shopfloor), um Einsichten in die tatsächlichen Abläufe der Produktion zu bekommen. Beispielsweise können Zeitreihen- und Sensordaten zur Verbesserung der Prozesse oder zur Problemanalyse genutzt werden.

Beispielhafte Use-Cases

- Aufzeichnung von Temperatur-, Druck-, Vibrations- oder Drehmomentdaten an Maschinen
- Analyse von Zeitreihen (z. B. Verarbeitungsgeschwindigkeit eines Fördersystems)
- Identifikation von Prozessvariationen (abweichende Zykluszeiten)
- Training von Machine-Learning-Modellen für prädiktive Wartung oder Prozessoptimierung

Ziele/Business-Nutzen

- Verbesserung der Prozessstabilität und -qualität durch Identifizierung von Problembereichen
- Früherkennung und Vermeidung von Qualitätsproblemen
- Tief gehende Datenanalysen zur Prozessverbesserung
- Ermöglichung von Predictive Maintenance

Benötigte Maschinendaten und Datenmodell

- Rohdaten von Sensoren (z. B. Temperatur, Druck, Geschwindigkeit)
- Prozessparameter (z. B. Durchlaufzeit, Energieverbrauch)
- Zeitstempel und enge Verlinkung zu Serial-/Chargennummern, Aufträgen und Maschinen

Datenmenge und -relevanz

- Häufigkeit: Mittel bis sehr hoch, abhängig von Anzahl und Erfassungshäufigkeit der Sensoren
- Volumen je Nachricht: Mittel bis sehr hoch, abhängig von Anzahl der Sensoren
- Echtzeitanforderung: Kurze Latenzen hilfreich für sofortigen Eingriff bei (erwarteten) Prozessabweichungen; in den meisten Anwendungsfällen spielt Latenz nur eine geringe Rolle

Interaktionsmuster

- Streaming-Daten: Kontinuierliches Übertragen der Sensordaten via IIoT-Systeme oder Middleware (z. B. MQTT oder OPC UA)
- Analyse-Pipeline: Direkte Weitergabe der Daten aus dem Edge-System an SAP DM oder eine Datenbank / ein Drittanbietersystem zur Speicherung/Analyse

Einbindung in die Architektur

- Mit Blick auf Performance-Gesichtspunkte ist bei hohen Datenpunkten oft die Verwendung einer separaten Daten-Pipeline und ggf. eines Drittsystems für die Analyse empfehlenswert.

Typische Stolpersteine

- Hohe Volumina an Sensordaten können die Infrastruktur überlasten.
- Verständliche Darstellung großer Datenmengen (z. B. für Endanwender) ist oft komplex.
- Herausforderungen bei der Datenvorbereitung und -bereinigung

- Genauigkeit und Kalibrierung der Sensoren

2.3 Prozessverriegelung

Beschreibung der Type

Prozessverriegelung beschreibt die direkte Steuerung von Produktionsabläufen aufgrund bestimmter Bedingungen. Ziel ist es, Qualität und Sicherheit durch automatisches Eingreifen sicherzustellen, z. B. beim Auftreten eines Fehlers. Solche Verriegelungsmechanismen sind in der Regel zeitkritisch.

Beispielhafte Use-Cases

- Automatische Abschaltung einer Produktionslinie bei Detektion eines Fehlers (z. B. Temperaturüberschreitungen, Vibrationen)
- Closed Loop: Automatische Anpassung von Produktionsprozessparametern während des laufenden Prozesses (z. B. Sollwertvorgaben der Anlage). So kann drohender Ausschuss im Prozess verhindert bzw. die Qualität erhöht werden.
- Sicherheitsabschaltungen bei kritischen Zuständen (z. B. beschädigte Anlagenkomponenten)
- Blockieren der Weiterverarbeitung von fehlerhaften Bauteilen

Ziele/Business-Nutzen

- Erhöhung der Sicherheit und Reduktion von Unfällen
- Vermeidung von Ausschuss oder Beschädigungen an Maschinen
- Automatisiertes und bedarfsorientiertes Reagieren auf Störungen

Benötigte Maschinendaten und Datenmodell

- Zustandsdaten der Maschinen (z. B. Fehler-Codes, Alarmer).
- Kontextinformationen, z. B. zu gesperrten Komponenten
- Ggf. Prozessparameter in Echtzeit (z. B. Temperatur, Druck, Stromstärke)
- Ggf. Entscheidungen basierend auf festgelegten Grenzwerten oder Algorithmen

Datenmenge und -relevanz

- Häufigkeit: Mittel bis sehr hoch, abhängig von Anzahl und Erfassungshäufigkeit der Sensoren
- Volumen je Nachricht: Mittel bis sehr hoch, abhängig von Anzahl der Sensoren
- Echtzeitanforderung: Mittel bis sehr hoch. Insbesondere bei harter Prozessverriegelung hat jede Latenz direkten Einfluss auf den Produktionsablauf. Abhängig von der Durchlaufzeit kann schon eine Latenz von

50 ms eine signifikante Verzögerung bedeuten. In anderen Fällen (z. B. Durchlaufzeiten von mehreren Tagen/Wochen) hat eine übliche Latenz keinen signifikanten Einfluss.

Interaktionsmuster

- Befehlsgesteuerte Kommunikation: SAP DM sendet aktiv Steuerbefehle an das Edge-System oder an Maschinensteuerungen.
- Ereignisbasierte Aktionen: Auslösung einer Verriegelung, z. B. über OPC UA an die Maschinensteuerung.

Einbindung in die Architektur

- Edge-Layer, ggf. SCADA: Führt Verriegelungen in Echtzeit durch, da Entscheidungen im Steuerungssystem getroffen werden müssen (Minimierung der Latenz ist hier entscheidend)
- SAP DM: Protokollierung und Einbindung in übergeordnete Reporting- und Analyseprozesse

Typische Stolpersteine

- Latenzzeiten bei der Übertragung von Signalen (Echtzeitanforderungen!)
- Falsch positive Signale können den Prozess unnötig stoppen.
- Fehleranfälligkeit durch komplexe Logik (z. B. Kombination vieler Event-Bedingungen)
- Hoher Implementierungsaufwand für die Integration in verschiedene Maschinen
- Fehlendes Backup/Redundanz-Konzept mit hohem Risiko für Fertigungsstillstand

3 Architekturübersicht

Lange Zeit wurde die industrielle IT/OT-Landschaft entlang der ISA-95-Automatisierungspyramide strukturiert. Daten flossen dabei stufenweise von unten nach oben: Sensoren und Steuerungen gaben Informationen an SCADA weiter, diese wiederum an SAP DM, bis schließlich ERP-Systeme auf Unternehmensebene erreicht wurden. Dieses Schichtenmodell half, Komplexität zu ordnen, führte aber auch zu Nachteilen: Datensilos, lange Durchlaufzeiten, aufwendige Integrationen auf jeder Ebene und ein hoher Wartungsaufwand.

Mit der zunehmenden Digitalisierung in der Produktion ist dieses Modell an seine Grenzen gestoßen. Unternehmen benötigen heute Daten über alle Ebenen hinweg – in Echtzeit, kontextbezogen und für viele unterschiedliche Systeme parallel nutzbar. Genau hier setzt die Unified-NameSpace-Architektur (UNS) an. An die Stelle starrer Punkt-zu-Punkt-Verbindungen tritt ein durchgängiges Informations-Backbone, das alle relevanten Datenströme in einem einzigen Namespace verfügbar macht.

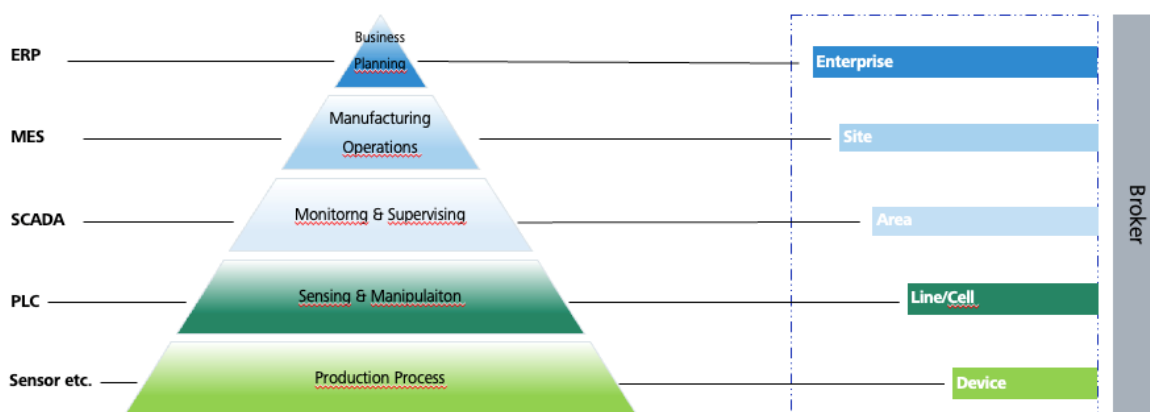


Abbildung 3 Ableitung der ISA-95-Pyramide zum Unified Namespace

Aufbauend auf den in Kapitel 2 vorgestellten klaren Integrationsmustern und Architekturvarianten zeigt dieses Kapitel mit dem Unified Namespace einen modernen Ansatz, der die Vernetzung von Maschinen, Liniensteuerung und SAP DM in einer durchgängigen Datenarchitektur neu denkt.

Das Herzstück dieser Architektur bildet eine Broker-Technologie, meist ein MQTT-Broker, der als Verteiler für sämtliche Nachrichten dient. Systeme werden nicht mehr direkt miteinander verknüpft, sondern agieren als Publisher oder Subscriber im Namespace. Auf diese Weise entsteht ein flexibles, zukunftssicheres Modell, das nahtlos skalierbar ist und die Pyramide „auflöst“.

3.1 Referenzarchitektur

Auch wenn die traditionelle, an ISA-95 und Punkt-zu-Punkt-Verbindungen orientierte Integration nach wie vor möglich ist und in vielen Umgebungen zuverlässig funktioniert, wird in diesem Kapitel bewusst ein moderner, auf einem Unified Namespace basierender Ansatz als zukunftsfähige Weiterentwicklung beschrieben.

Die Architektur gliedert sich nach den Best Practices in drei Ebenen, die miteinander über Broker verknüpft sind:

- Enterprise (zentrale Aggregation und globale IT-System-Anbindung)
- Site/Plant (lokale Kontextualisierung und Bereitstellung produktionsnaher Informationen)
- Edge (Maschinen- und Sensordaten in Echtzeit).

Zusammen bilden diese Ebenen ein durchgehendes Netz, das klassische Integrationsbarrieren der ISA-95-Pyramide eliminiert und eine flexible, standardisierte Datenarchitektur ermöglicht.

Deswegen lässt sich der Unified Namespace am besten als Weiterentwicklung der klassischen ISA-95-Pyramide verstehen. Während das traditionelle Modell Daten hierarchisch aufschichtet und schrittweise von den Maschinen bis zu den Enterprise-Systemen weitergibt, bricht die UNS-Architektur diese strenge Logik auf. Die Ebenen Enterprise, Site/Plant und Edge bleiben zwar als Orientierung bestehen, doch an die Stelle von starren Silos tritt ein durchgängiges Kommunikationsmodell.

Das verbindende Element ist dabei der Broker, der zugleich die Integrationsschicht (Middle Layer) darstellt. Ohne den Broker existiert kein UNS, denn er ist die zentrale Komponente, über die alle Daten im Unternehmen publiziert, abonniert, transportiert und kontextualisiert werden. Auf jeder Ebene – Enterprise, Site/Plant oder Edge – bilden Broker-Instanzen die Basis. Sie übernehmen die Rolle des Middle Layers, vermitteln zwischen den betrieblichen Systemen, reichern – entweder selbst oder mithilfe anderer angeschlossener Applikationen – Daten mit Kontext an und stellen sie anderen Anwendungen über den Namespace bereit.

So fungiert der zentrale Enterprise-Broker als globaler Aggregationspunkt, an dem alle Standortinformationen zusammenfließen und an ERP-, CRM- oder Analytics-Plattformen weitergeleitet werden. Die lokalen Broker auf Site- oder Plant-Ebene sind selbst Middle Layer, die Rohdaten aus der Fertigung aufnehmen, in eine semantisch nutzbare Form bringen und gezielt an Enterprise-Systeme oder lokale Anwendungen ausgeben. An der Edge-Ebene schließlich werden Broker-Instanzen direkt an Maschinen, Steuerungen und Sensoren betrieben, um Daten unmittelbar dort zu erfassen, wo sie entstehen.

Durch diese Broker-basierte Architektur wird die starre Pyramide in ein flexibles, vernetztes System transformiert. Jede Ebene kommuniziert nicht mehr ausschließlich „nach oben“ oder „nach unten“, sondern ist über den Broker mit dem gesamten Unternehmensnetzwerk verbunden. Der Middle Layer ist dadurch nicht irgendeine zusätzliche Instanz, sondern exakt die Broker-Ebene, die den Unified Namespace zum Leben erweckt und dafür sorgt, dass Daten nicht in Silos verharren, sondern dort zur Verfügung stehen, wo sie benötigt werden.

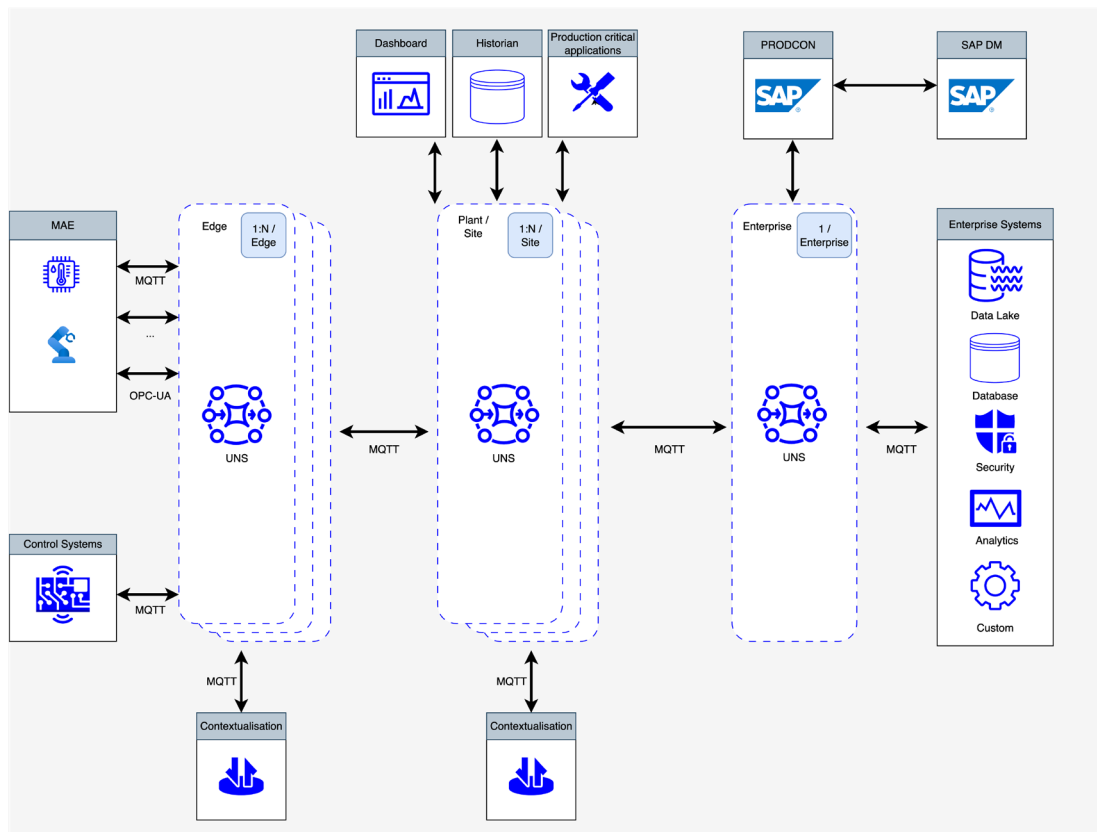


Abbildung 4 Referenzarchitektur

3.1.1 Integrationsschicht (Middle Layer)

Der Middle Layer bildet das Herzstück der Unified-Namespaces-Architektur – und er ist im Wesentlichen nichts anderes als die Broker-Plattform. Der Broker ist hier nicht nur ein technisches Hilfsmittel, sondern das zentrale Element, das alle Daten- und Informationsflüsse koordiniert.

Als Broker stellt der Middle Layer die Kommunikation zwischen IT- und OT-Welt sicher. Auf der einen Seite bindet er Shopfloor-Systeme wie SCADA, die SAP DM oder (lokale) Analyse-Tools ein, auf der anderen Seite verbindet er diese mit IT-Systemen wie ERP, EWM oder Qualitätsmanagement. Seine Rolle besteht darin, die oft unterschiedlichen Protokolle, Datenformate und Anforderungen beider Bereiche zu

abstrahieren und ihnen einen einheitlichen Zugriff über Topics im Namespace zu ermöglichen.

Darüber hinaus übernimmt der Middle Layer die Kontextualisierung eingehender Daten. Rohdaten aus der Produktion, die oft nur einfache Messwerte oder Statusmeldungen enthalten, werden hier mit den notwendigen Metadaten angereichert – etwa Version, Zeitstempel oder Einheit. Erst diese Aufbereitung macht die Daten für nachgelagerte Systeme sinnvoll nutzbar. Damit fungiert der Middle Layer nicht nur als Transportweg, sondern auch als intelligente Übersetzungsschicht, die Rohsignale in verwertbare Informationen verwandelt.

Der Middle Layer erfüllt außerdem eine zentrale Filter- und Entlastungsfunktion. Er sorgt dafür, dass nur die benötigten und bereits angereicherten Daten an den Enterprise-Broker weitergegeben werden. So werden Bandbreiten geschont und die Verarbeitungslogik in den zentralen IT-Systemen wird vereinfacht. Gleichzeitig ermöglicht die lokale Broker-Architektur, dass Analysen, Monitoring und kurzfristige Eingriffe bereits auf Site- oder Plant-Ebene erfolgen können – auch dann, wenn die Verbindung zum Enterprise-Broker unterbrochen ist.

Seine enge Verbindung zur Broker-Technologie macht den Middle Layer zudem hochgradig flexibel und skalierbar. Neue Systeme oder Produktionslinien lassen sich einfach integrieren, indem Sie Ihre Daten im Broker publizieren oder die benötigten Topics abonnieren. Es sind keine komplexen Punkt-zu-Punkt-Schnittstellen mehr erforderlich. Ergänzt wird dies durch feingranulare Sicherheits- und Berechtigungskonzepte direkt auf Ebene des Brokers, sodass sichergestellt ist, dass nur autorisierte Nutzer oder Systeme Zugriff auf bestimmte Daten erhalten.

Damit ist klar: Der Middle Layer ist identisch mit der Broker-Ebene. Er bildet die operative Drehscheibe, über die IT und OT verbunden werden, Rohdaten veredelt und nutzbar gemacht werden und das Unternehmen die Grenzen der klassischen ISA-95-Pyramide auflöst.

Beispiele für MQTT-Broker und Anbieter

In der Unified-Namespace-Architektur spielen MQTT-Broker eine zentrale Rolle als Kommunikationsplattform zwischen Maschinen, Systemen und Anwendungen. Die folgenden Anbieter und Lösungen haben sich in der Praxis bewährt:

- **HiveMQ:** Ein leistungsstarker und skalierbarer MQTT-Broker, der speziell für industrielle Anwendungen entwickelt wurde.
- **EMQX:** Eine Open-Source-Lösung, die für hohe Skalierbarkeit und Performance bekannt ist.
- **Mosquitto:** Ein leichtgewichtiger MQTT-Broker, der sich ideal für kleinere Anwendungen eignet.

- **AWS IoT Core MQTT Broker:** Eine Cloud-basierte Lösung von Amazon, die eine nahtlose Integration mit anderen AWS-Diensten ermöglicht.

3.1.2 Edge-Connectivity-Software

Die Auswahl der richtigen Edge-Connectivity-Software ist ein zentraler Erfolgsfaktor für die Integration von SAP Digital Manufacturing in den Shopfloor. Diese Software-Lösungen ermöglichen die Anbindung von Maschinen, Steuerungen und Sensoren an IT-Systeme und unterstützen dabei eine Vielzahl von Kommunikationsprotokollen. Sie sind essenziell für die Datenvorverarbeitung, -aggregation und -weiterleitung. Lösungen wie der SAP ProdCon für SAP DM und ähnliche Produkte bieten unterschiedliche Ansätze und Funktionen.

Diese Lösungen unterscheiden sich insbesondere in den folgenden Bereichen:

Unterstützte Protokolle

Die verschiedenen Software-Lösungen unterstützen eine Vielzahl von Kommunikationsprotokollen. Zu den gängigen Protokollen gehören OPC UA, MQTT, RESTful APIs und andere. Die Fähigkeit, mehrere Protokolle zu unterstützen, ist ein entscheidendes Argument für die Flexibilität und Anpassungsfähigkeit der Software. Dies ermöglicht eine nahtlose Integration in unterschiedliche Systemlandschaften und erleichtert die Kommunikation zwischen Maschinen und Anwendungen.

Installationsarten

Die Installationsarten reichen von On-Premise-Lösungen bis hin zu Cloud-basierten und hybriden Ansätzen. Zudem gibt es klassische Client-Installationen sowie moderne, containerfähige Anwendungen. Unternehmen sollten die passende Lösung basierend auf ihren Sicherheitsanforderungen, ihrer Infrastruktur und ihrem Budget wählen. Oft sind diese Edge-Connectoren auch Teil umfassender IIoT-Plattformen, Middlelayer-Software oder von MQTT-Brokern.

Skalierbarkeit und Anpassbarkeit

Die Skalierbarkeit der Software ist ein wesentlicher Faktor, insbesondere bei der Einführung, Wartung und Aktualisierung. Unternehmen, die ihre Produktion erweitern möchten, profitieren von anpassbaren Lösungen, die spezifische Anforderungen erfüllen und sich an veränderte Bedingungen anpassen können.

Benutzerfreundlichkeit

Die Software sollte intuitiv bedienbar sein, um eine einfache Implementierung und Nutzung zu gewährleisten. Eine benutzerfreundliche Oberfläche erleichtert die Schulung und den täglichen Betrieb, was zu einer höheren Akzeptanz und Effizienz führt.

3.1.3 Erfahrungen und Herausforderungen bei der Einführung von Edge-Lösungen

Unterstützte Protokolle

Die Praxis hat gezeigt, dass die Anbindung jeder exotischen Schnittstelle nicht immer rentabel ist. Es ist ratsam, Shopfloor-Equipment in Bezug auf Schnittstellen zu modernisieren und für Neubeschaffungen klare Standards vorzugeben.

Integration in bestehende Systeme

Die Integration von Edge-Connectivity-Software in bestehende Green- und Brownfield-Infrastrukturen kann komplex sein und erfordert eine sorgfältige Planung und Implementierung. Shopfloor-nahe Installationen, beispielsweise auf IIoT-PCs in oder an Maschinen, bringen neue Herausforderungen mit sich wie etwa Garantieverlust oder Eingriffe in die Maschinen (CE), und für IT-Mitarbeiter ist der physikalische Zugriff erschwert.

Freeware vs. kostenpflichtige Lösungen

Bei der Wahl einer Connect-Software sollten Unternehmen abwägen, ob sie Freeware oder etablierte Anbieter bevorzugen, die Support, Weiterentwicklungen und Skalierbarkeit bieten.

Datenvolumen

Ein erhöhtes Datenaufkommen, Taktzeiten und Verarbeitungsgeschwindigkeiten in nachgelagerten Systemen stellen Herausforderungen dar, insbesondere wenn langsamere Protokolle (wie REST) im Drittanbieterumfeld eingesetzt werden.

Shopfloor-Daten

Im Shopfloor müssen alle relevanten Daten sorgfältig erfasst werden. Connect-Software unterstützt diesen Prozess, indem sie direkt Daten von Maschinen abrufen, berechnet, anreichert, normiert und um Einheiten ergänzt.

Rollenverständnis und Zusammenarbeit

Ein klares Rollenverständnis und die Zusammenarbeit zwischen Instandhaltung, Produktionsmitarbeitern, IT und Datenkonsumenten sind entscheidend für den Erfolg der Implementierung und den Betrieb von Edge-Lösungen. Team-Mitglieder sollten über das notwendige Know-how und Verständnis verfügen, um effektiv zusammenzuarbeiten. So hat nun eine Maschinen-Downtime für Wartungsmaßnahmen andere Folgen – hier ist ein Umdenken im Shopfloor nötig.

Use-Cases

Zu Beginn wurde ein spezifischer Anwendungsfall identifiziert und umgesetzt, der schnell das Interesse in der Produktion für weitere Use-Cases wie Energiemanagements, die Gesamtanlageneffektivität (Overall Equipment Effectiveness, OEE), die Rückverfolgbarkeit (Traceability), die vorausschauende Wartung (Predictive Maintenance), die Integration von Manufacturing Execution Systems (MES), das Zustandsmonitoring (Condition-Monitoring), die Produktionsplanung sowie die Prozessautomatisierung weckten. Viele dieser Anwendungsfälle zielen auf unterschiedliche Unternehmensanwendungen ab oder erfordern lediglich eine Teilmenge der gesammelten Daten.

3.1.4 Beispiele für Edge-Connectivity-Software und Anbieter

Die folgenden Anbieter und Lösungen haben sich in der Praxis bewährt und bieten unterschiedliche Funktionalitäten für die Edge-Konnektivität.

Edge Connectivity Layer: Typische Komponenten

Der Edge Connectivity Layer bildet die Grundlage für die Anbindung von Maschinen und Geräten an die IT-Systeme. Sie stellt sicher, dass Daten von Maschinensteuerungen, Sensoren und anderen Shopfloor-Komponenten gesammelt, verarbeitet und an die Middleware weitergeleitet werden können. Typische Lösungen und Anbieter in diesem Bereich sind:

- **OPC UA Server (Maschine oder Gateway):** OPC UA Server sind essenziell für die standardisierte Kommunikation zwischen Maschinen und IT-Systemen. Sie ermöglichen die Bereitstellung strukturierter Datenmodelle und die Interoperabilität zwischen verschiedenen Geräten.
- **Kepware:** Eine führende Lösung für industrielle Konnektivität, die eine Vielzahl an Protokollen unterstützt und eine einfache Integration in bestehende Systeme ermöglicht.
- **Siemens Industrial Edge:** Eine Plattform, die speziell für die Integration von Siemens-Maschinen und -Steuerungen entwickelt wurde und eine nahtlose Datenverarbeitung am Edge erlaubt.
- **Unified Automation:** Bietet flexible OPC-UA-basierte Lösungen für die Maschinenanbindung und Datenintegration.
- **Codesys OPC UA:** Eine Software-Plattform, die eine einfache und standardisierte Kommunikation für speicherprogrammierbare Steuerungen (SPS) ermöglicht.

Protokoll-Gateways

Protokoll-Gateways spielen eine wichtige Rolle bei der Übersetzung und Vermittlung zwischen verschiedenen Kommunikationsprotokollen. Sie sind nützlich in

heterogenen Produktionsumgebungen, in denen Maschinen und Systeme mit unterschiedlichen Protokollen arbeiten. Zu den gängigen Anbietern gehören:

- **Anybus Gateways:** Unterstützen eine Vielzahl von Protokollen und ermöglichen die Integration von Geräten mit unterschiedlichen Kommunikationsstandards.
- **HMS Networks:** Bietet eine breite Palette an Lösungen für die industrielle Kommunikation, einschließlich Gateways für die Protokollübersetzung.
- **Softing EdgeConnector:** Eine leistungsstarke Lösung für die Anbindung von OPC UA und anderen Protokollen an IT-Systeme.

Edge Data Collector

Edge-Data-Collector-Lösungen sind darauf ausgelegt, Daten direkt an der Maschine zu sammeln, zu verarbeiten und an übergeordnete Systeme weiterzuleiten. Sie bieten oft zusätzliche Funktionen wie Datenvorverarbeitung, Filterung und Aggregation. Beispiele für solche Lösungen sind:

- **Ignition Edge:** Eine modulare Plattform, die Daten von Maschinen sammelt und verarbeitet, um sie für Analysen und Steuerungen bereitzustellen.
- **Azure IoT Edge:** Eine Cloud-basierte Lösung von Microsoft, die Datenverarbeitung und -analyse direkt am Edge ermöglicht.
- **AWS IoT Greengrass:** Eine Lösung von Amazon Web Services, die es ermöglicht, IIoT-Geräte mit der Cloud zu verbinden und gleichzeitig lokale Datenverarbeitung zu unterstützen.

3.1.5 Handlungsempfehlung: Edge-Connectivity-Software

1. Schnittstellenstandardisierung

- Einheitliche Standards (z. B. OPC UA, MQTT) festlegen
- Exotische/proprietäre Schnittstellen vermeiden
- Retrofit vs. Modernisierung wirtschaftlich prüfen

2. Installationsart wählen

- On-Premise, Cloud oder Hybrid nach Sicherheit, Infrastruktur, Budget auswählen
- Containerfähige, modulare Lösungen bevorzugen
- Prüfen, ob Stand-alone oder Teil einer IIoT-Plattform

3. Skalierbarkeit sicherstellen

- Lösung muss mit Datenvolumen und Use-Cases wachsen
- Anpassung von Protokollen und Datenmodellen möglich halten

4. Benutzerfreundlichkeit beachten

- Intuitive Bedienoberfläche auswählen
- Schulungen und Verantwortlichkeiten definieren
- Updates und Wartung ohne Produktionsstopp ermöglichen

5. Integration planen

- Bestandsaufnahme von Green- und Brownfield-Systemen durchführen
- Risiken (Garantieverlust, CE) beachten
- Enge Zusammenarbeit zwischen Produktion, Instandhaltung, IT

6. Kosten/Nutzen prüfen

- Freeware nur für Tests/Piloten nutzen
- Für Produktivbetrieb etablierte Anbieter mit Support wählen

7. Datenvolumen beherrschen

- Netzwerk- und Backend-Leistung dimensionieren
- Effiziente Protokolle (MQTT > REST) einsetzen
- Daten vorverarbeiten, normieren und aggregieren am Edge

8. Rollen und Zusammenarbeit klären

- Rollenmodell zwischen IT, Produktion, Instandhaltung definieren
- Auswirkungen von Wartungs-Downtimes transparent machen
- Interdisziplinäre Zusammenarbeit fördern

Die detaillierte Definition der Anwenderrollen sowie deren Überführung von fachlichen Anforderungen in technische Rollen und Berechtigungen im SAP-System wird in Kapitel 5 – insbesondere Abschnitt 5.3.1 – ausführlich beschrieben.

9. Schrittweise Use-Cases umsetzen

- Mit einem klaren Start-Use-Case beginnen (z. B. OEE, Energiemanagement)
- Schrittweise Erweiterung auf weitere Anwendungen (Traceability, Predictive Maintenance, MES, Condition-Monitoring)
- Modularität der Software nutzen, um Systembrüche zu vermeiden
- Fachbereich/Fachabteilung von Beginn an stark mit einbinden (Akzeptanz)
- Use-Case mit dem meisten Benefit und Verbreitungsgrad auswählen

3.2 Hardware für Maschinenanbindung

Die Integration der SAP DM ist bezüglich Schnittstellen und Kommunikationsprotokollen begrenzt. Die Praxis zeigt, dass die Anbindung verschiedenster Geräte eine Middleware erforderlich macht. Diese Praxiserfahrung

führt zur folgenden Hardware-Empfehlung:

- 1) OT-Router als zentraler Data-Hub an der Maschine
 - Netzwerksegmentierung zur Trennung vom Business-Netz
 - Fernwartung über Rendezvous-Server
- 2) Enlyze Spark
 - a. Prozessdaten sammeln und normieren
 - b. Edge-Device für Prozessdaten in Cloud bzw. für SAP DM
- 3) Microsoft AKS / Suse-Edge-Stack (Kubernetes-Cluster)
 - a. SAP DM Edge
 - b. Middleware zur Virtualisierung der Maschinenanbindungen

Um größtmögliche Autarkie sicherzustellen, sind vorgenannte Komponenten an jeder Maschine installiert.

4 Skalierung

4.1 **Template-Ansatz**

Der Template-Ansatz in Bezug auf die Shopfloor Integration verfolgt das Ziel, Maschinenanbindungen über Werke und Linien hinweg konsistent und effizient umzusetzen. Durch die Definition wiederverwendbarer Konfigurationsbausteine lassen sich Integrationsprojekte standardisieren und beschleunigen. Grundlage bilden interoperable Kommunikationsstrukturen und einheitliche Datenmodelle, die eine flexible Anbindung unterschiedlichster Maschinen und Steuerungen ermöglichen. Templates dienen dabei als technische und organisatorische Referenz für die Umsetzung in verschiedenen Produktionsbereichen. Die strukturierte Ablage und Nutzung dieser Daten im Unified Namespace unterstützt die Wiederverwendbarkeit und Skalierbarkeit.

Komplex wird es beim Mapping der Variablen, also der Zuordnung von Maschinenwerten zu logischen Datenpunkten im Connector. Dieser Aufwand skaliert direkt mit der Anzahl der Maschinen und kann bei großen Rollouts zu hohen Ressourcenbedarfen führen. Ein durchdachter Template-Ansatz kann hier signifikant entlasten. Wird ein standardisiertes Variablenmodell einmalig erstellt, lässt sich die Konfiguration für weitere Maschinen deutlich schneller übernehmen. Voraussetzung

dafür ist eine frühzeitige Standardisierung der Variablenbezeichnungen und Datenstrukturen, abgestimmt auf die wichtigsten Use-Cases und die eingesetzten Connector-Technologien. Templates sollten sowohl die technische Konfiguration als auch die semantische Struktur der Datenpunkte enthalten.

4.1.1 Datenstrukturierung im Unified Namespace

Die Konfiguration von Shopfloor-Assets kann effektiv über einen Unified Namespace organisiert werden. Dabei werden alle relevanten Datenpunkte – wie Status, Messwerte, Zustände oder KPIs – in einem konsistenten, hierarchisch strukturierten Informationsraum abgelegt. Dies ermöglicht eine standardisierte und kontextbezogene Abbildung der Maschinenvariablen, unabhängig vom Standort oder von der konkreten Maschineninstanz.

Durch die Verwendung von Namenskonventionen und semantisch klar definierten Topics lassen sich Templates für Variablenstrukturen erstellen, die über Werke hinweg wiederverwendbar sind. So kann z. B. ein OEE-Template für eine Maschinenklasse zentral gepflegt und lokal ausgerollt werden, ohne dass jedes Werk individuelle Mapping-Logiken entwickeln muss. Der UNS dient dabei als technisches Rückgrat für die Datenbereitstellung und -verarbeitung im Connector.

Ein typisches Beispiel ist die Integration eines OPC UA Servers nach UMATI-Standard. UMATI definiert standardisierte Informationsmodelle für Maschinen wie Werkzeugmaschinen. Der OPC UA Server stellt strukturierte Datenpunkte bereit, etwa zu Betriebszustand, Stückzahl oder Energieverbrauch. Diese werden über den Production Connector abgerufen und in SAP Digital Manufacturing weiterverarbeitet. Das Mapping erfolgt über BrowsePaths wie */MachineTool/Production/PartCount*, die im Connector auf Variablen wie *machine.production.count* abgebildet werden. Die Konfiguration kann manuell oder über Mapping-Dateien erfolgen und nutzt den Unified Namespace zur semantischen Einordnung.

Ein weiteres Beispiel für die Nutzung eines Unified Namespace ergibt sich bei MQTT-basierten Architekturen. Hier erfolgt die Datenbereitstellung über hierarchisch strukturierte Topics, z. B. */acme/munich/assembly/line1/machineA/temperature*. Ein Sensor publiziert regelmäßig unter diesem Topic, und ein Edge-Service oder SAP DM kann darauf abonnieren. Wird ein definierter Schwellenwert überschritten, z. B. Temperatur > 80 °C, löst der Subscriber einen ereignisgesteuerten Trigger aus – etwa durch Publikation eines Alerts unter */acme/munich/assembly/line1/machineA/alert*. Dieses Muster ist asynchron, entkoppelt und hoch skalierbar und eignet sich ideal für moderne IIoT-Umgebungen.

Auch REST-basierte Integrationen lassen sich in den Unified Namespace einbetten. Ein Beispiel ist die Übermittlung von Produktionsaufträgen aus einem MES oder ERP-System an ein SCADA-System. Dabei wird ein HTTP-POST-Request an die SCADA-API gesendet. Das SCADA-System verarbeitet die Daten und gibt eine Bestätigung zurück.

```
1 {
2   "orderId": "PO-2025-00123",
3   "productCode": "XG-450",
4   "quantity": 100,
5   "startTime": "2025-11-08T08:00:00Z",
6   "endTime": "2025-11-08T16:00:00Z",
7   "machineId": "LINE1-MA01",
8   "parameters": {
9     "temperature": 180,
10    "pressure": 5.2,
11    "speed": 1200
12  },
13  "priority": "high"
14 }
15
```

Die Kombination aus Template und Namespace reduziert den Aufwand für Variablen-Mapping und schafft die Grundlage für eine skalierbare, wartbare und transparente Datenarchitektur im Shopfloor.

4.1.2 Governance und organisatorische Einbindung

Ein nachhaltiger Template-Ansatz erfordert klare Governance-Strukturen und eine enge organisatorische Einbindung. Templates sollten zentral entwickelt und versioniert, aber dezentral ausgerollt und betrieben werden. Die Freigabe erfolgt über definierte Prozesse, die auch die Einbindung relevanter Stakeholder aus IT, Produktion und Instandhaltung sicherstellen. Änderungen am Template unterliegen einem kontrollierten Lifecycle-Management, das Transparenz und Nachvollziehbarkeit gewährleistet.

Ein bewährtes Modell besteht aus einem zentralen, globalen Owner, der die Templates entwickelt, pflegt und freigibt, sowie lokalen Ownern in den jeweiligen Werken, die für die Implementierung und den Betrieb verantwortlich sind. Regelmäßige Abstimmungen – etwa in einem Shopfloor Integration Change Board – stellen sicher, dass Änderungen koordiniert erfolgen, lokale Anforderungen berücksichtigt werden und der Rollout über Standorte hinweg konsistent bleibt.

Unified Namespace und Change-Management

Der Unified Namespace liegt typischerweise in der OT und damit im Fachbereich. Änderungen an Infrastruktur-Templates – etwa neue Indikatoren, Kommunikationsschnittstellen oder Mapping-Logiken – wirken sich direkt auf die IT-Integration aus. Daher müssen solche Änderungen über das Change Board als offizieller Change Request bei der IT eingereicht werden. Dies stellt sicher, dass alle

abhängigen Systeme und Schnittstellen konsistent bleiben und keine Integrationsprobleme entstehen.

Jede Änderung am Template erfordert eine Versionierung und den Rollout auf alle Bestandsverwendungen, um Schiefstände zu vermeiden. Dies betrifft sowohl die Konfiguration auf Edge-Geräten als auch die Cloud-seitige Modellierung. Mit jeder neuen Version sind Regressionstests durchzuführen, um sicherzustellen, dass bestehende Funktionalitäten nicht beeinträchtigt werden. Der Rollout erfolgt idealerweise entlang eines klaren Release-Plans, der auch Rückfallstrategien und Eskalationspfade beinhaltet.

4.1.3 Infrastruktur – Business-Continuity und Risikobewertung

Die Installation des SAP Production Connectors kann auf unterschiedliche Weise erfolgen, wobei jede Variante spezifische Vor- und Nachteile im Hinblick auf Business Continuity Management (BCM), Wartbarkeit, Skalierbarkeit und Risiko-Management mit sich bringt.

Die klassische Variante ist die lokale Installation auf einem Windows-Server je Werk. Sie bietet eine hohe Nähe zur Maschine und damit geringe Latenzzeiten. Besonders in Werken mit begrenzter IT-Infrastruktur oder bei Retrofit-Projekten ist diese Lösung attraktiv, da sie einfach zu betreiben und unabhängig von zentralen Systemen ist. Aus BCM-Sicht ermöglicht sie eine schnelle Wiederanlaufstrategie im Fehlerfall, da die Systeme lokal isoliert sind. Gleichzeitig entstehen jedoch Risiken durch inkonsistente Software-Stände, erhöhten Wartungsaufwand und eingeschränkte Skalierbarkeit.

Eine Alternative stellt die Installation auf Linux-basierten Systemen dar. Diese Variante eignet sich für produktionsnahe Edge-Geräte mit höherer IT-Reife. Sie bietet Vorteile hinsichtlich Stabilität, Sicherheit und Automatisierbarkeit, insbesondere durch die Möglichkeit, Deployments skriptgesteuert auszurollen. Die Risiken liegen hier vor allem im höheren Know-how-Bedarf und in möglichen Kompatibilitätsproblemen mit proprietären Agenten oder Maschinenprotokollen.

Für kleinere Cluster oder Pilotprojekte kann die Containerisierung über Docker Swarm eine praktikable Lösung sein. Sie erlaubt ein schnelles Setup, gute Isolation einzelner Instanzen und einfache Wiederanlaufmechanismen. Allerdings ist Docker Swarm in seiner Orchestrierungsfähigkeit begrenzt und weniger geeignet für unternehmenskritische, skalierbare Produktionsumgebungen.

Die Installation auf einer zentralen Kubernetes-Plattform ist die bevorzugte Lösung für Unternehmen mit mehreren Werken und dem Ziel einer einheitlichen Governance. Kubernetes bietet hochverfügbare Cluster, Self-Healing-Funktionen und automatisierte Rollouts. Aus BCM-Sicht ermöglicht dies eine zentrale Steuerung von Sicherheitsrichtlinien, Monitoring und Wiederanlaufstrategien. Gleichzeitig entstehen Risiken durch die Komplexität im Betrieb, potenzielle Latenzprobleme bei entfernten

Werken und die Notwendigkeit, zentrale Komponenten wie Datenbanken oder Message-Broker eigens abzusichern.

Eine hybride Struktur aus mehr als einer der zuvor genannten Möglichkeiten kann sinnvoll sein, wenn sie klar abgegrenzt und organisatorisch getragen wird – etwa durch dedizierte Verantwortlichkeiten, automatisierte Test-Pipelines und ein zentrales Change-Management. Für Unternehmen mit hoher Standardisierungsabsicht und begrenzten Ressourcen empfiehlt sich jedoch eine konsolidierte Architektur, um Wartbarkeit, Sicherheit und Skalierbarkeit langfristig sicherzustellen.

4.2 Anlagenbeschaffung

4.2.1 Beschaffungsrichtlinie und Integration in Betriebsprozesse

Im Rahmen des Beschaffungsprozesses ist eine verbindliche Beschaffungsrichtlinie zu definieren, die sicherstellt, dass neue Anlagen konform zu bestehenden Templates angefragt und geliefert werden. Dies betrifft insbesondere die Kommunikationsfähigkeit (z. B. OPC UA mit Umati-Profilen), die Datenstruktur sowie die Offenheit der Schnittstellen.

Dabei ist strategisch und finanziell zu bewerten, ob Maschinen direkt mit einem standardisierten OPC-UA-Umati-Profil bestellt werden sollen – was eine hohe Interoperabilität und geringeren Integrationsaufwand verspricht, jedoch bei späteren Änderungen kostenpflichtige Anpassungen durch den Hersteller erforderlich macht. Alternativ kann ein maximal offener Variablenkatalog gefordert werden, bei dem die Maschine möglichst viele Rohdatenpunkte bereitstellt. Die semantische Normierung erfolgt dann unternehmensintern in der Middleware – beispielsweise über ein Mapping-Script, das die Variablen auf interne Templates und Indikatoren abbildet. Diese Variante bietet mehr Flexibilität und Kontrolle, erfordert jedoch eine stärkere interne Governance und technische Expertise.

Die Templates sollten eng mit den Betriebs- und Beschaffungsprozessen verknüpft sein, insbesondere bei Retrofit-Projekten oder der Integration neuer Maschinen. Erweiterungen erfolgen idealerweise entlang konkreter Use-Cases. Verantwortlichkeiten und Eskalationspfade sollten im Template dokumentiert sein, um eine schnelle Reaktion im Fehlerfall zu ermöglichen.

Neben der technischen Konfiguration sollten Templates auch organisatorische Rahmenbedingungen wie Rollenmodelle, Freigabeprozesse und Betriebsrichtlinien enthalten. So wird sichergestellt, dass die Lösung nicht nur technisch funktioniert, sondern auch operativ getragen und kontinuierlich weiterentwickelt werden kann.

4.2.2 Standardisierung von Kommunikationsprotokollen

Ein wesentlicher Hebel für die effiziente Integration von SAP Digital Manufacturing (SAP DM) mit dem Shopfloor liegt in der Standardisierung der Hardware-Connectivity. Dies ist zwar keine zwingende Voraussetzung für die Umsetzung, bietet jedoch einen Effizienzgewinn bei der Datenintegration und -verarbeitung. Um dies zu erreichen, sollten Unternehmen bereits während des Einkaufsprozesses klare Anforderungen an die Kommunikationsfähigkeit der Hardware definieren und diese konsequent umsetzen.

Das Minimalziel einer solchen Anforderungsdefinition ist, dass alle neuen Maschinen und Geräte standardisierte Kommunikationsprotokolle (z. B. OPC UA, MQTT oder REST-APIs) unterstützen, um die Integration in die IT-Landschaft zu vereinfachen.

4.2.3 Standardisierung der auszutauschenden Datensätze

Ergänzend zu der Sicherstellung der Unterstützung der Kommunikationsprotokolle ist es ratsam, Vorgaben für einheitliche Datenmodelle zu formulieren, um sicherzustellen, dass die erzeugten Datenstrukturen und -semantiken nahtlos in die bestehende Datenarchitektur und in SAP DM integriert werden können, ohne dass aufwendige Mappings erforderlich sind.

Die Entwicklung eines „Leitfadens für Connectivity- und Datenmodellierungsstandards“ für die Beschaffung, in welchem die Erwartungen an Datenschnittstellen, empfohlene Protokolle und verfügbare Datenformate klar definiert werden, kann als unterstützende Maßnahme dienen. Dies erleichtert nicht nur die Kommunikation mit Lieferanten, sondern steigert auch die langfristige Interoperabilität im Maschinenpark. Hersteller können so von Beginn an sicherstellen, dass ihre Lieferungen den Connectivity-Standards des Unternehmens entsprechen.

Ein konsequent implementiertes, standardisiertes Datenmodell stellt sicher, dass die Hardware-Connectivity effizient genutzt werden kann und die Integration mit SAP DM sowie anderen digitalen Fertigungslösungen eine solide Grundlage erhält.

1. Grundprinzipien des Datenmodells

- **Interoperabilität:** Das Datenmodell sollte mit gängigen Standards wie OPC UA, ISA-95 oder MQTT kompatibel sein, um eine einfache Integration in bestehende Systeme zu ermöglichen.
- **Modularität:** Die Struktur des Modells sollte modular aufgebaut sein, damit zusätzliche Daten oder neue Geräte flexibel integriert werden können.
- **Standardisierte Semantik:** Einheitliche Benennungen und Definitionen für Attribute und Parameter erleichtern die Weiterverarbeitung der Daten. Z. B. sollte „Temperatur“ als Attribut in jeder Maschine gleich definiert sein.

- **Skalierbarkeit:** Das Modell muss in kleinen Umgebungen genauso gut funktionieren wie in komplexen, verteilten Produktionssystemen.
- **Transparenz:** Alle Daten, die erfasst werden, sollten nachvollziehbar sein, mit einer klaren Dokumentation zu Quelle, Datentyp, Einheit und Zweck.

Beispielhafte Struktur des Datenmodells

Die Struktur eines solchen Datenmodells sollte mehrere Ebenen abdecken:

Geräte-/Sensor-Ebene: Physische Datenquellen

- Identifikationsdaten: Seriennummer, Hersteller, Modell, Standort.
- Prozessparameter: Betriebszustand (Run/Idle/Fehler), Temperatur, Druck, Drehmoment, Materialverbrauch.
- Zustandsdaten: Betriebszeit, Alarm-/Fehler-Codes, Wartungsstatus, Energiemessung.

Shopfloor-Ebene: Aggregationsebene

- Maschinenkennzahlen: Erzeugung von Key Performance Indicators (KPIs) wie OEE (Overall Equipment Effectiveness), Verfügbarkeiten, Auslastung, Ausschussrate.
- Prozesseinheiten: Gruppierung von Geräten/Maschinen in logische Einheiten (z. B. Produktionslinie, Fertigungszelle).
- Ereignisprotokolle: Ereignisse wie Stillstände, Wartungen oder Alarmer in einer standardisierten Struktur erfassen.

Unternehmens-IT-Ebene: Geschäftsprozesse und Entscheidungen

- Produktionsauftragsdaten: Verknüpfung mit SAP-Systemen (z. B. SAP ERP oder SAP S/4HANA) für Produktionspläne, Materialbedarfe und Rückmeldungen.
- Traceability-Daten: Rückverfolgung einzelner Chargen oder Seriennummern durch die gesamte Prozesskette.
- Qualitätsdaten: Erfassung von Prüfmerkmalen und Abweichungen nach Fertigungsprozess.

Technische Umsetzung

- Datenprotokolle: OPC UA als Basisprotokoll, um Metadaten und Inhalte hierarchisch und standardisiert abzubilden. Für IIoT-Anwendungen könnte MQTT für Echtzeit-Datenübertragungen verwendet werden.
- Datenformate: Verwendung universeller Formate wie JSON oder XML zur strukturierten Abbildung und zum Transfer der Daten.

- Einheits- und Metadatenstandards: Verbindliche Einheiten (z. B. ISO-Maßeinheiten wie °C, bar) und Metainformationen zu jedem Datenpunkt (z. B. Zeitpunkt, Quelle, Einheit).

Beispielhafte Datenstruktur für eine Maschine (JSON-Format)

```
{
  "machine_id": "M12345",
  "manufacturer": "XYZ Manufacturing",
  "model": "ABC1000",
  "location": "Shopfloor 1",
  "status": {
    "operational": true,
    "current_state": "RUN",
    "temperature": 85.5,
    "pressure": 3.4,
    "last_maintenance": "2023-10-01"
  },
  "kpis": {
    "oee": 87.2,
    "availability": 92.3,
    "performance": 89.4,
    "quality": 97.6
  },
  "events": [
    {
      "event_id": "E56789",
      "type": "Alarm",
      "timestamp": "2023-10-03T14:22:11Z",
      "description": "Overheating detected",
      "severity": "High"
    }
  ]
}
```

4.3 Redundanzen

Der SAP Production Connector ist ein produktionskritischer Integrationsbaustein zwischen Shopfloor und SAP Digital Manufacturing und muss entsprechend ausfallsicher betrieben werden. Er wird als Windows-basierte Anwendung betrieben. Dieses Kapitel beschreibt die konzeptionellen Bausteine zur Ausfallsicherheit (Redundanz/Verfügbarkeit, RTO/RPO, Logging/Monitoring sowie Service-Levels und Rollenmodelle).

4.3.1 Redundanz und Hochverfügbarkeit

Der SAP Production Connector besteht aus einem Main Service (Windows Service) und Agent-Instanzen, die als eigenständige Windows Services im Hintergrund laufen und die Datenanbindung an Shopfloor-Quellen realisieren. Mehrere Agent-Instanzen können parallel und unabhängig voneinander betrieben werden; die Anzahl ist im Wesentlichen durch die Kapazität des Hosts begrenzt. Bei fatalen Fehlern kann eine Agent-Instanz stoppen und wird dann als „interrupted“ angezeigt; der Status der Agent-Instanzen kann zur Laufzeit überwacht werden.

Für die Verfügbarkeitsbetrachtung sind die Kommunikationspfade relevant: Aufrufe aus der Cloud zum Production Connector erfolgen über den Cloud Connector, während Verbindungen vom Production Connector zur Cloud direkt aufgebaut werden. Die Kommunikation zwischen Cloud Connector, Production Connector und SAP Digital Manufacturing erfolgt über RESTful Web Services; intern erfolgt die Kommunikation zwischen Host (Cloud-Services) und Agent-Instanzen über WebSocket.

Für Redundanz- und Failover-Szenarien sind konsistente Endpunkt- und Zertifikatsparameter sicherzustellen: Der interne Host wird als HTTPS-URL im Format „https://<Fully Qualified Domain Name>:<Port>“ geführt, und der FQDN soll dem Common Name (CN) des verwendeten Server-Zertifikats entsprechen. Zudem können nicht mehrere Production-Connector-Systeme mit derselben internen Host-URL angelegt werden.

Für Windows-basierte Workloads kann Hochverfügbarkeit über Cluster-/Failover-Mechanismen auf Plattform- bzw. Infrastruktur-Ebene umgesetzt werden, bei denen mehrere Knoten gemeinsam betrieben werden und bei Ausfall eines Knotens andere Knoten die Workload übernehmen (Failover), während Rollen kontinuierlich überwacht und bei Bedarf neu gestartet oder migriert werden. Ergänzend können für Windows Services Wiederherstellungsaktionen (Recovery-Settings) definiert werden, um Dienste nach Fehlern kontrolliert neu zu starten bzw. definierte Recovery-Aktionen auszulösen.

4.3.2 Logging und Monitoring

Ein zentraler Bestandteil der Ausfallsicherheit ist ein zuverlässiges Logging- und Monitoring-Konzept. Der SAP Production Connector kann in Logging-Infrastrukturen

wie Elasticsearch eingebunden werden, um Betriebsdaten, Fehlerzustände und Kommunikationsereignisse nachvollziehbar zu dokumentieren. In Kombination mit Tools wie Kibana oder Grafana lassen sich Dashboards zur Echtzeitüberwachung erstellen, die sowohl IT- als auch OT-relevante Metriken abbilden.

Monitoring und Logging sind essenziell für:

- Früherkennung von Störungen und Performance-Problemen
- Nachvollziehbarkeit von Kommunikationsfehlern
- Unterstützung bei Restore- und Recovery-Prozessen
- Einhaltung von SLA-Vorgaben durch transparente Betriebsdaten

4.3.3 Recovery-Time-Objective (RTO) und Recovery-Point-Objective (RPO)

Das RTO beschreibt die maximale Zeitspanne, die ein System nach einem Ausfall benötigen darf, um wieder betriebsbereit zu sein. In produktionsnahen Umgebungen mit hoher Taktung ist ein RTO von unter 2 Stunden empfehlenswert, insbesondere bei zentralen Installationen oder Cloud-basierten Szenarien.

Das RPO definiert den maximal tolerierbaren Zeitraum, in dem Daten verloren gehen dürfen. Bei synchroner Datenreplikation kann ein RPO von unter 5 Minuten erreicht werden. In dezentralen Setups mit manuellen Backups ist ein höherer RPO realistisch, sollte aber klar dokumentiert und kommuniziert sein.

Diese Werte können als Zielgrößen für lokale oder hybride Production-Connector-Installationen übernommen werden, müssen jedoch an die jeweilige Infrastruktur und Betriebsstrategie angepasst werden.

4.3.4 Service-Levels und Betriebsorganisation

Ein Service-Level-Agreement (SLA) ist ein zentraler Bestandteil der Betriebsstrategie für produktionskritische Systeme. Es legt verbindlich fest, welche Leistungen, Verfügbarkeiten und Reaktionszeiten im Betrieb erwartet werden und wie im Fehlerfall zu handeln ist. SLAs schaffen Transparenz und Verlässlichkeit zwischen internen Bereichen wie IT und OT sowie gegenüber externen Dienstleistern.

Ein gut definierter SLA ermöglicht:

- Verbindlichkeit: Klare Zuständigkeiten und messbare Leistungskennzahlen.
- Transparenz: Alle Beteiligten kennen die Erwartungen und Abläufe im Fehlerfall.
- Planbarkeit: Ressourcen und Prozesse können auf definierte Wiederanlaufzeiten abgestimmt werden.
- Eskalation: Bei Nichteinhaltung greifen definierte Eskalationspfade.

4.3.5 Rollen im Service-Level-Modell

In einer verteilten Werkstruktur ist eine abgestufte Service-Organisation erforderlich, um SLA-Vorgaben effizient umzusetzen:

- Key-User (Fachbereich/Produktion): Erste Anlaufstelle bei Störungen im operativen Betrieb. Meldet Probleme, dokumentiert Vorfälle und hilft bei der Fehlerbeschreibung.
- Lokale IT/OT: Verantwortlich für den First-Level-Support. Führt erste Analysen durch, prüft die lokale Infrastruktur und leitet bei Bedarf an zentrale Stellen weiter.
- Zentrale IT / Product-Owner: Zuständig für Second-Level-Support, Template-Management, Systemkonfiguration und Koordination mit SAP oder externen Dienstleistern. Trägt die Verantwortung für die Einhaltung der SLA-Vorgaben und die Weiterentwicklung der Betriebsstrategie.

5 Betrieb

5.1 Backup und Restore

Die Backup-und-Restore-Funktion dient der Sicherung und Wiederherstellung der Production-Connector-Konfiguration in Notfallszenarien (z. B. Ausfall der Installation oder Downtime). Gesichert werden Konfigurationselemente, die in SAP Digital Manufacturing definiert und in den Production Connector deployt werden. Die Funktion ist nicht als generischer Mechanismus zur Rückkehr auf einen beliebigen historischen Systemzustand („Rollback“) gedacht, sondern zur Wiederherstellung einer konsistenten Konfiguration im Production Connector.

Die Sicherungen enthalten die vollständige Konfiguration des Production Connectors, insbesondere die relevanten Konnektivitätsobjekte und Zuordnungen (z. B. Systeme, Agent-Instanzen, Shopfloor-Systeme, Benutzer sowie Benutzergruppen inkl. Rollenzuordnungen) im Rahmen der Konnektivitätskonfiguration.

Für die Backup-Erstellung in SAP Digital Manufacturing muss das erforderliche Benutzerrecht zum Erstellen von Backups vorhanden sein. Die Wiederherstellung (Restore) darf nur durch einen als Administrator berechtigten Benutzer erfolgen. Für eine erfolgreiche Wiederherstellung muss außerdem das für die Cloud-zu-Production-Connector-Kommunikation erforderliche Zertifikat verfügbar sein.

SAP Digital Manufacturing erstellt automatische Konfigurations-Backups. Diese automatische Sicherung wird mit jeder neuen Deployment-Ausführung aktualisiert und als .pbf-Datei in SAP Digital Manufacturing abgelegt. Das automatische Backup dient als primäre Wiederherstellungsbasis, um die zuletzt konsistent deployte Konfiguration wiederherzustellen.

Zusätzlich zum automatischen Backup kann ein manuelles Backup über die App „Configure Production Connectivity“ erstellt werden. Dazu wird in der App die betreffende Production-Connector-Instanz ausgewählt und im Bereich „Backup“ die Funktion „Create Backup“ ausgeführt. Ein manuell erzeugtes und heruntergeladenes

Backup dient vorrangig der Unterstützung der Fehleranalyse (Issue-/Support-Fall). Es kann auch genutzt werden, wenn im Ausnahmefall eine lokale Sicherungsdatei benötigt wird (z. B. zur Übergabe an den Support im Rahmen der Diagnose).

Der Download des aktuellen Backups erfolgt über „Download Backup“. Beim Download wird ein Passwort vergeben, da die Datei verschlüsselte Inhalte enthält. Dieses Passwort wird benötigt, wenn die Wiederherstellung aus der heruntergeladenen Datei durchgeführt wird. Die zuverlässige Aufbewahrung des Passworts ist sicherzustellen, da ohne das Passwort eine Wiederherstellung aus der heruntergeladenen Datei nicht möglich ist.

Die Wiederherstellung wird eingesetzt, wenn eine Production-Connector-Installation ausgefallen ist oder die Konfiguration nach einem Störereignis wiederhergestellt werden muss. Ein Restore ersetzt die bestehende Konfiguration vollständig durch den wiederhergestellten Stand. Während des Recovery-Prozesses können keine neuen Backups erstellt und keine Konfigurationsänderungen vorgenommen werden.

Die Wiederherstellung wird über die App „Configure Production Connectivity“ angestoßen: Im Tab „Production Connector“ wird die betroffene Instanz ausgewählt, im Bereich „Backup“ wird „Restore Backup“ ausgeführt und der Wiederherstellungsprozess gestartet. Wenn eine heruntergeladene bzw. lokal erstellte Backup-Datei verwendet wird, ist das beim Download vergebene Passwort einzugeben. Als Vorgehensprinzip gilt, bevorzugt die in der Cloud vorliegende Sicherung als Wiederherstellungsbasis zu verwenden; lokale oder hostfremde Backups sind nur in begründeten Ausnahmefällen einzusetzen.

Nach Abschluss des Restore ist sicherzustellen, dass die Production-Connector-Instanz in SAP Digital Manufacturing wieder korrekt verbunden/zugeordnet ist und dass weitere Deployments erst nach erfolgreicher Synchronisierung bzw. stabiler Verbindung durchgeführt werden. Anschließend ist die Betriebsfähigkeit durch geeignete Funktionsprüfungen zu verifizieren (z. B. Status der Agent-Instanzen, Verbindungen zu relevanten Shopfloor-Systemen sowie Kommunikationsfähigkeit zur Cloud).

Die operative Abwicklung im Störfall (Ticket-Anlage, Kommunikation, Eskalation, Bereitschaft und Rückspielung von Lösungen) erfolgt gemäß Kapitel 5.2. Die Vergabe und Governance der erforderlichen Rollen und Berechtigungen (insbesondere Administratorrechte für Restore) sind im Rollenkonzept gemäß Kapitel 5.3 geregelt.

5.2 Betriebskonzept

Der nachfolgende Prozess regelt, dass Störungen zunächst immer lokal durch Facharbeiter, Key-User und Power-User aufgenommen und bearbeitet werden (Level 0/1) und nur bei Bedarf an zentrale IT- und SAP-Support-Ebenen sowie externes Consulting (Level 2/3) eskaliert werden, die den Vorfall beheben und die Lösung wieder an den Standort zurückspielen.

Abgrenzung und Geltungsbereich

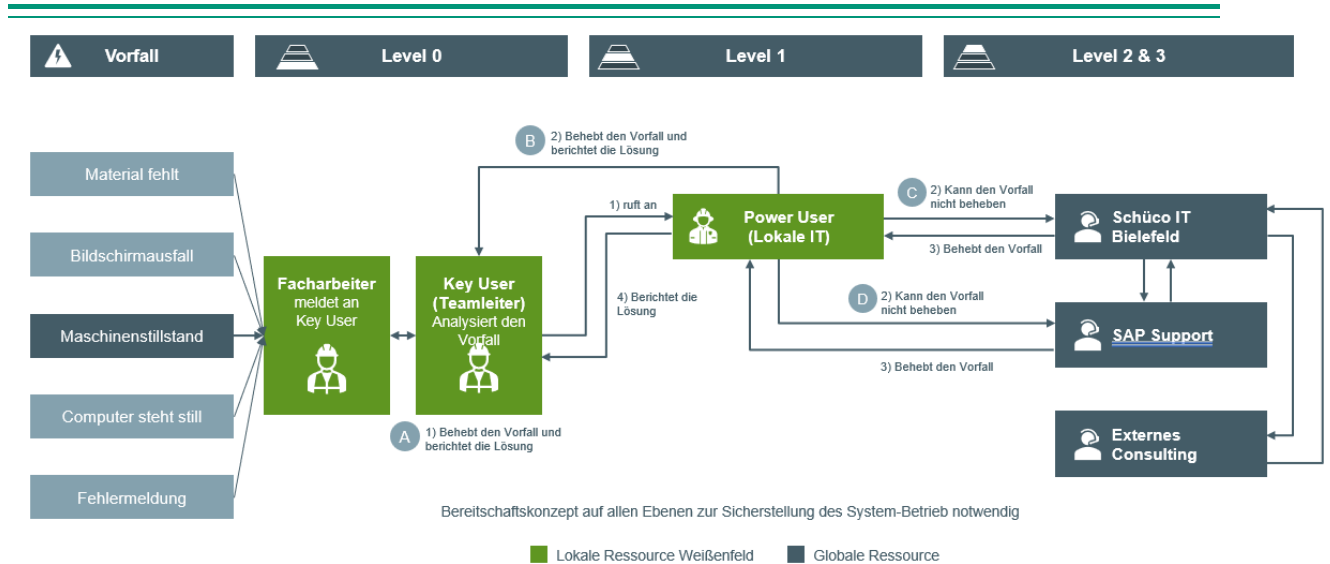


Abbildung 5 Beispiel mehrstufiger Support-Prozess

Die Grafik zeigt einen mehrstufigen Support-Prozess mit folgenden Elementen:

5.2.1 Vorfal tritt auf (Level 0)

Auf dem Shopfloor bemerkt der Facharbeiter eine Störung (z. B. Maschinenstillstand, fehlendes Material, Systemfehler). Dieser Zustand wird in der Grafik als „Vorfal“ dargestellt. Der Facharbeiter meldet den Vorfal an den Key-User bzw. Team-Leiter.

5.2.2 Lokale Analyse und Bearbeitung durch Key-User und Power-User (Level 1)

- Der Key-User (Team-Leiter) prüft die Störung.
- Falls erforderlich, wird der Power-User / die lokale IT eingebunden.
- Diese Ebene fungiert als erste Anlaufstelle für Betreuung und Fehlerbehebung.
- Ziel ist es, den Vorfal möglichst direkt vor Ort zu lösen.

5.2.3 Eskalation an zentrale IT / SAP Support (Level 2)

Kann der Vorfal auf Level 1 nicht gelöst werden, erfolgt die Eskalation an die zentrale IT / SAP Support. Reicht die zentrale IT-/SAP-Kompetenz nicht aus, wird externes Consulting hinzugezogen (z. B. spezialisierte SAP- oder Systemberater).

5.2.4 Behebung und Rückmeldung der Störung

Auf Level 2 bzw. 3 wird der Vorfal technisch und fachlich behoben. Die erarbeitete Lösung wird vom höheren Level zurück in die Organisation gespielt

5.2.5 Sicherstellung des Systembetriebs durch Bereitschaftskonzept

Die Grafik macht deutlich, dass:

- lokale Ressourcen (Facharbeiter, Key-User, Power-User) und
- globale Ressourcen (zentrale IT, SAP Support, externes Consulting)

durch ein abgestimmtes Bereitschaftskonzept auf allen Ebenen (Level 0–3) zusammenwirken müssen, um den Systembetrieb dauerhaft sicherzustellen.

5.3 Rollenkonzept

Die Definition und Umsetzung des Rollenkonzeptes im SAP-Projekt erfolgt in einem strukturierten, mehrstufigen Vorgehen. Ziel ist es, fachliche Anforderungen aus den Geschäftsprozessen konsistent in technische Rollen und Berechtigungen im SAP-System zu überführen und so eine sichere, revisionskonforme und praxisnahe Nutzung der Lösung zu gewährleisten.

5.3.1 Fachliche Definition der Anwenderrollen

Ausgangspunkt ist die fachliche Sicht auf die Geschäftsprozesse in Produktion, Logistik, Qualitätssicherung und IT-Unterstützung. Auf dieser Basis werden zunächst die benötigten Anwenderrollen definiert, zum Beispiel:

- Mitarbeiter Produktion
- Mitarbeiter Kommissionierung
- Mitarbeiter Logistik
- Team-Leiter Produktion (Key-User)
- Mitarbeiter Arbeitsvorbereitung
- Mitarbeiter QS
- SAP-DM-/System-Administratoren (dezentrale IT)

Für jede Anwenderrolle werden die wesentlichen Aufgaben und Tätigkeiten beschrieben, ohne zunächst technische Details zu betrachten. Dazu gehören unter anderem:

- Welche Prozessschritte führt die Rolle aus?
- Welche Buchungen, Rückmeldungen oder Freigaben werden durchgeführt?
- Welche Auswertungen oder Monitoring-Funktionen werden benötigt?

Das Ergebnis dieses Schrittes ist eine fachliche Rollenübersicht mit klar beschriebenen Verantwortlichkeiten und Tätigkeiten.

5.3.2 Abgleich mit SAP-DM-Standardrollen auf der SAP BTP

Im nächsten Schritt werden die fachlich definierten Rollen mit dem vorhandenen SAP-Standard-Rollenkatalog auf der SAP BTP abgeglichen. Die benötigten Einzelrollen werden den jeweiligen Anwenderrollen zugeordnet

Das Ergebnis ist eine fachlich geprüfte Liste der benötigten Systemfunktionen und Berechtigungsobjekte je Anwenderrolle.

5.3.3 Technische Ausgestaltung der Rollen und Benutzerzuordnung

Auf Basis der fachlichen Rollenbeschreibung und der dokumentierten Systemobjekte erfolgt die technische Ausgestaltung der Rollen in der SAP BTP. Hierzu werden die jeweiligen Rollen den neu angelegten Rollenkollektionen zugeordnet und im System verbucht. Im Anschluss daran können die Rollenkollektionen den jeweiligen Benutzern zugeordnet werden.

Nach der technischen Umsetzung erfolgen Integrationstests und ggf. produktionsnahe Tests:

- Kann der Benutzer alle für seine Aufgaben notwendigen Schritte ausführen?
- Sind unzulässige Berechtigungen (z. B. administrative Funktionen für operative Nutzer) ausgeschlossen?
- Auffälligkeiten werden genutzt, um Rollen iterativ nachzuschärfen.

5.3.4 Dokumentation und Governance

Die Ergebnisse des Rollen-Designs werden in geeigneter Form dokumentiert. Dazu gehören insbesondere:

- Beschreibung der fachlichen Anwenderrollen
- Zuordnung zu Rollenkollektionen
- Übersicht der Rollen je Organisationseinheit/Standort
- Regelungen zur Beantragung, Vergabe, Änderung und Entziehung von Rollen (z. B. im Rahmen des Benutzer- und Berechtigungs-Managements)

Diese Dokumentation bildet die Grundlage für:

- zukünftige Rollenerweiterungen oder -anpassungen,
- Audits und Revisionen,
- Schulungen und Support-Prozesse.

6 Fazit und nächste Schritte

Die erfolgreiche Integration von SAP Digital Manufacturing in der Produktion ist ein entscheidender Schritt zur Digitalisierung und Optimierung der Produktionslandschaft. Dieser Leitfaden hat die wesentlichen Aspekte beleuchtet, um Unternehmen bei der Planung, Implementierung und Skalierung zu unterstützen. Die zentralen Erkenntnisse dieses Leitfadens werden im Folgenden zusammengefasst.

6.1 Zusammenfassung der zentralen Themen

Architektur und Datenintegration

- Der Unified Namespace (UNS) bietet eine moderne, flexible Alternative zur traditionellen ISA-95-Pyramide. Er ermöglicht eine durchgängige, standardisierte Datenarchitektur, die Silos auflöst und Echtzeitkommunikation sicherstellt.
- Die Integrationsschicht (Middle Layer), insbesondere die Broker-Technologie, ist das Herzstück der Integration und sorgt für eine nahtlose Verbindung zwischen IT- und OT-Systemen.

Anwendungsfälle und Nutzen

- Die drei zentralen Anwendungsfälle – Ergebnis-Reporting, Prozess-Reporting und Prozessverriegelung – decken die wichtigsten Anforderungen an Transparenz, Effizienz und Sicherheit ab.
- Durch die Nutzung standardisierter Datenmodelle und Protokolle wie OPC UA und MQTT können Daten effizient erfasst, verarbeitet und analysiert werden.

Governance und Skalierung

- Ein klar definierter Template-Ansatz erleichtert die Wiederverwendbarkeit und Skalierung von Integrationslösungen über mehrere Standorte hinweg.
- Governance-Strukturen und ein abgestimmtes Change-Management sind essenziell, um Konsistenz und Transparenz sicherzustellen.

Betrieb und Ausfallsicherheit

- Ein robustes Betriebs- und Support-Konzept mit klar definierten Rollen und Eskalationswegen ist entscheidend, um den kontinuierlichen Betrieb sicherzustellen.
- Backup-und-Restore-Strategien sowie Redundanzkonzepte minimieren Risiken und gewährleisten Business-Continuity.

6.2 Kernaussagen

- **Standardisierung ist der Schlüssel:** Einheitliche Protokolle, Datenmodelle und Templates reduzieren Komplexität und beschleunigen die Integration.
- **Unified Namespace als Zukunftsmodell:** Der UNS ermöglicht eine flexible, skalierbare und zukunftssichere Datenarchitektur.

- **Governance und Zusammenarbeit:** Klare Rollen und Verantwortlichkeiten sowie ein effektives Change-Management sind unerlässlich.
- **Pilotprojekte priorisieren:** Starten Sie mit einem klar definierten Use-Case, um erste Erfolge zu erzielen und Akzeptanz zu schaffen.
- **Betrieb und Sicherheit:** Investieren Sie in ein durchdachtes Betriebs- und Backup-Konzept, um Ausfallzeiten zu minimieren und Datenintegrität zu gewährleisten.

6.3 Empfehlungen für die nächsten Schritte

Pilotprojekt starten

- Wählen Sie einen spezifischen Anwendungsfall mit hohem Nutzenpotenzial (z. B. OEE-Reporting oder Prozessverriegelung).
- Nutzen Sie den Leitfaden, um die Integration zu planen und umzusetzen.

Governance-Strukturen etablieren

- Definieren Sie ein zentrales und ein dezentrales Rollenmodell für die Betriebsorganisation.
- Richten Sie ein Change-Management-Board ein, um Änderungen an Templates und Systemen zu koordinieren.

Technologie und Infrastruktur bewerten

- Prüfen Sie die Eignung Ihrer bestehenden Shopfloor-Systeme für die Integration mit SAP DM.
- Investieren Sie in Edge-Connectivity-Software und Middleware, die Ihre Anforderungen an Protokollunterstützung und Skalierbarkeit erfüllen.

Langfristige Skalierung planen

- Entwickeln Sie Templates für wiederkehrende Anwendungsfälle und standardisieren Sie die Datenstruktur im Unified Namespace.
- Berücksichtigen Sie zukünftige Anforderungen wie Predictive Maintenance und KI-gestützte Analysen.

Schulungen und Akzeptanz fördern

- Schulen Sie Ihre Mitarbeitenden in der Nutzung von SAP DM und den zugehörigen Systemen.
- Fördern Sie die Akzeptanz durch die Einbindung der Fachbereiche in die Planung und Umsetzung.

6.4 Ausblick

Die Digitalisierung der Produktion ist ein kontinuierlicher Prozess. Mit der Einführung von SAP DM und der Nutzung moderner Architekturen wie des Unified Namespace schaffen Unternehmen die Grundlage für zukünftige Innovationen. Themen wie künstliche Intelligenz, maschinelles Lernen und prädiktive Analysen werden in den kommenden Jahren eine immer größere Rolle spielen.

Die in diesem Leitfaden beschriebenen Best Practices und Empfehlungen bieten eine solide Basis, um diese Entwicklungen zu nutzen und die Wettbewerbsfähigkeit langfristig zu sichern.

Über die DSAG

Die Deutschsprachige SAP-Anwendergruppe e. V. (DSAG) ist einer der einflussreichsten Anwenderverbände der Welt. Über 4.000 Mitgliedsunternehmen und mehr als 70.000 Mitgliedspersonen bilden ein starkes Netzwerk, das sich vom Mittelstand bis zum DAX-Konzern und über alle wirtschaftlichen Branchen in Deutschland, Österreich und der Schweiz (DACH) erstreckt. Auf Basis dieser Reichweite gewinnt der Industrieverband fundierte Einblicke in die digitalen Herausforderungen im DACH-Markt. Die DSAG nutzt diesen Wissensvorsprung, um die Interessen der SAP-Anwender zu vertreten und ihren Mitgliedern den Weg in die Digitalisierung zu ebnen.

Weitere Informationen finden Sie unter:

www.dsag.de

Impressum

Wir weisen ausdrücklich darauf hin, dass das vorliegende Dokument nicht jeglichen Regelungsbedarf sämtlicher DSAG-Mitglieder in allen Geschäftsszenarien antizipieren und abdecken kann. Insofern müssen die angesprochenen Themen und Anregungen naturgemäß unvollständig bleiben. Die DSAG und die beteiligten Autor:innen können bezüglich der Vollständigkeit und Erfolgsgemeinheit der Anregungen keine Verantwortung übernehmen.

Alle Rechte liegen, soweit nicht ausdrücklich anders gekennzeichnet, bei:

Deutschsprachige SAP Anwendergruppe e.V.

Altrottstraße 34 a

69190 Walldorf | Deutschland

Telefon +49 6227 35809-58

E-Mail info@dsag.de

dsag.de

Jedwede unerlaubte Verwendung ist nicht gestattet. Dies gilt insbesondere für die Vervielfältigung, Bearbeitung, Verbreitung, Übersetzung oder die Verwendung in elektronischen Systemen/digitalen Medien.