



IIP-Ecosphere Whitepaper

IIP-Ecosphere Plattform - Anforderungen (Funktionale und Qualitäts-Sicht)

Version 1.0

Holger Eichelberger, Christian Sauer,
Amir Shayan Ahmadian, Michael Schicktanz,
Andreas Dewes, Gregory Palmer, Claudia Niederée

White Paper IIP-2021/002



IIP-Ecosphere
Next Level Ecosphere for
Intelligent Industrial Production

Disclaimer

Die Inhalte des Dokuments wurden mit großer Sorgfalt erstellt. Obwohl die Informationen mit größtmöglicher Sorgfalt erstellt wurden, besteht kein Anspruch auf sachliche Richtigkeit, Vollständigkeit und/oder Aktualität, insbesondere kann diese Publikation nicht den besonderen Umständen des Einzelfalles Rechnung tragen.

Eine Verwendung liegt daher in der eigenen Verantwortung des Lesers. Jegliche Haftung wird ausgeschlossen. Dieses Dokument enthält Material, das dem Urheberrecht einzelner oder mehrerer IIP-Ecosphere-Konsortialparteien unterliegt. Alle Rechte, auch der auszugsweisen Vervielfältigung, liegen bei den Urhebern.

Dieses Dokument spiegelt nur die Ansicht der Autoren zum Zeitpunkt der Veröffentlichung wider. Das Bundesministerium für Wirtschaft und Energie bzw. der zuständige Projektträger haften nicht für die Verwendung der hierin enthaltenen Informationen.

Veröffentlichung: März, 2021 auf <https://www.iip-ecosphere.eu/>

DOI: 10.5281/zenodo.4485774

Gefördert durch:



Bundesministerium
für Wirtschaft
und Energie

Executive Summary

Dieses Dokument beschreibt die Anforderungen an die IIP-Ecosphere Plattform. Die Anforderungen basieren auf Diskussionen mit den Partnern und den Arbeitspaketen bzw. Teilprojekten (bzw. deren Repräsentanten) von IIP-Ecosphere, wie etwa den Demonstratoren. Als Grundlagen wurden eine im Projekt erstellte Übersicht aktueller Industrie 4.0 Plattformen als Grundlage sowie eine Anforderungserhebung auf Benutzersicht einbezogen.

Das Dokument umfasst 141 Anforderungen mit insgesamt 181 Unter-Anforderungen in 14 Themengebieten und dient nun der Ableitung einer gemeinsamen Architektur sowie der Realisierung der IIP-Ecosphere Plattform. Es ist geplant, dieses Dokument bei Aufscheinen neuer Anforderungen fortzuschreiben um damit dem IIP-Ecosphere Ökosystem aber auch weiteren Plattformentwicklungen als Grundlage zu dienen.

Inhaltsverzeichnis

1	Einleitung.....	5
1.1	Motivation und Ziele	5
1.2	Interaktionen mit anderen Initiativen	5
1.3	Struktur des Dokuments	6
2	Übersicht IIP-Ecosphere Plattform.....	7
3	Anforderungen.....	14
3.1	Terminologie	14
3.2	Quellen für Anforderungen.....	15
3.3	Generelle Anforderungen	16
3.4	Konnektoren und Verbindungen/Interkonnektivität	18
3.5	Heterogenes, dynamisches Deployment	20
3.6	Sicherheit	24
3.7	Datenschutz	24
3.8	Zentrale Speicherdienste	27
3.9	Data Sharing	28
3.10	Datenintegration	29
3.11	Konfigurierbarkeit	30
3.12	Optimiertes bzw. adaptives Deployment	32
3.13	KI-(Service-)Baukasten.....	33
3.14	Adaptive Dienst/Komponenten-Auswahl	36
3.15	Virtualisierung	37
3.16	Anwendungsunterstützung.....	37
4	Abgleich mit der Benutzungssicht (Usage View)	41
5	Priorisierung und Anforderungskonflikte.....	44
6	Zusammenfassung	46
7	Referenzen.....	47
1	Anhang: Anforderungs-Fragebogen.....	51

1 Einleitung

1.1 Motivation und Ziele

Die Digitalisierung der Industrie sowie die zunehmende Vernetzung erhöhen die Leistungsfähigkeit und Komplexität technischer Systeme und der zugehörigen Prozesse. Im industriellen Umfeld wird der intelligenten Produktion (Industrie 4.0) hohes Innovationspotenzial eingeräumt, wobei künstliche Intelligenz (KI) als Schlüsseltechnologie gilt. Die Vision des im KI-Innovationswettbewerb des Bundesministeriums für Wirtschaft und Energie (BMWi) geförderten Projekts IIP-Ecosphere ist es, einen Innovationssprung im Bereich der Selbstoptimierung der Produktion auf Basis vernetzter, intelligenter, autonomer Systeme zur Steigerung der Produktivität, Flexibilität, Robustheit und Effizienz zu erreichen. Ziel ist es, ein neuartiges Ökosystem – die Next Level Ecosphere for Intelligent Industrial Production (IIP-Ecosphere) – aufzubauen, die eine „nächste Ebene“ in der intelligenten Produktion ermöglicht.

Eine der Aktivitäten zur Umsetzung dieser Vision ist die Realisierung einer unternehmensübergreifenden virtuellen Plattform, die sowohl existierende Geräte und Fabrikinstallationen in herstellerunabhängiger Weise miteinander verbindet, als auch auf diesen Installationen KI in flexibler Weise bereitstellt. Neben der niedrigschwelligen Anwendung von KI-Verfahren sollen dabei KI-Verfahren möglichst optimal auf Fertigungsgeräten bereitgestellt werden und mit Plattformdiensten verknüpfbar sein, wie beispielsweise mit Speicherdiensten oder einer fabrikweiten Datenintegration. Die Plattform soll zudem eine gemeinsame Nutzung von Daten (Data Sharing) ermöglichen als auch dafür erforderliche Sicherheitsmechanismen bereitstellen.

Dieses Dokument ist ein Schritt zur Realisierung der IIP-Ecosphere-Plattform. Das vorliegende Dokument beschreibt die Vision der zu realisierenden Plattform-Dienste, deren Abgrenzung zu existierenden Plattformen und, insbesondere, die Anforderungen an die zu realisierende Plattform. Vision und Anforderungen wurden in intensiver Diskussion mit den Partnern des Projekts ermittelt. Die Konzepte und Anforderungen werden im Rahmen des Community-Dialogs des IIP-Ecosphere-Projekts mit weiteren Stakeholdern diskutiert und evaluiert. Das kann und soll zu einer überarbeiteten Version dieses Dokuments führen.

1.2 Interaktionen mit anderen Initiativen

Existierende Plattformen für die industrielle Produktion sowie individuelle (innovative) technische Aktivitäten können sich mit unserer Planung überlappen. Dies bietet aber Raum für Interaktion, Synergien und Abstimmungen mit anderen Initiativen und Aktivitäten. In diesem Dokument ist dies wie folgt berücksichtigt:

- **Industrie 4.0 Standards und Spezifikationen** ermöglichen die Interaktion von Industrie 4.0 Plattformen. Die IIP-Ecosphere Plattform wird Standards in diesem Bereich wie Verwaltungsschalen [26, 40], RAMI 4.0 [31], Industrie 4.0 Sprache [27, 38] OPC UA [22] oder OneM2M [21] berücksichtigen, umsetzen und unterstützen.
- **Andere Industrie 4.0 Plattformen** können ähnliche oder komplementäre Funktionalitäten bereitstellen. Die IIP-Ecosphere Plattform ist daher insbesondere als offene und virtuelle Plattform konzipiert, die sich mit anderen Plattformen integrieren und wechselseitig Fähigkeiten (soweit verfügbar) nutzen kann. Unsere Planung berücksichtigt hierzu die Übersicht industrieller Plattformen [33], die in IIP-Ecosphere erarbeitet wurde.
- Wichtige **Teilfunktionalitäten** existieren bereits, insbesondere als **Open Source**, z.B., die Realisierung von Verwaltungsschalen im BaSyS/BaSyX-Projekt [1, 6] sowie die Umsetzung von Protokollen oder Diensten im Eclipse IoT Ökosystem [8]. Diese Funktionalitäten sollen nicht neu entwickelt, sondern geeignet integriert werden.

- **Konzeptionelle Zusammenarbeit** mit dem Ziel der Standardisierung, z.B. mit dem Verein Deutscher Werkzeugmaschinenfabriken (VDW) als Partner in IIP-Ecosphere oder anderen Initiativen bzw. Förderprojekten im Bereich Industrie 4.0, z.B., Plattform Industrie 4.0 [25], BaSyS [1], OneM2M [21] oder “VWS vernetzt” [41].

1.3 Struktur des Dokuments

Dieses Dokument ist wie folgt strukturiert: Im nächsten Kapitel geben wir eine Übersicht über die (Planungen zur) IIP-Ecosphere Plattform. In Kapitel 3 beschreiben wir die Anforderungen an die Plattform entlang der Konzepte, die in der Übersicht in Kapitel 2 eingeführt wurden. In Kapitel 4 diskutieren wir den Abgleich der Anforderungen mit [36] und in 5 beschreiben wir eine initiale Priorisierung der Anforderungen sowie Anforderungskonflikte. Kapitel 6 fasst dieses Dokument zusammen und in Kapitel 7 verwendete Arbeiten aufgezählt werden. Anhang 1 enthält die (sprachlich und bzw. in Formatierung leicht angepasste) Anforderungsanfrage an die IIP-Ecosphere Projektteile, insbesondere die Demonstratoren.

2 Übersicht IIP-Ecosphere Plattform

In diesem Kapitel geben wir eine kurze Übersicht über die IIP-Ecosphere Plattform, sowohl in Form der angedachten Plattformdienste als auch durch eine initiale Übersichtsarchitektur. Dabei soll diese Übersicht die Strukturierung der folgenden Anforderungen motivieren und erläutern.

Abbildung 1 visualisiert (als eine Art Mind-Map) die von der IIP-Ecosphere Plattform bereitzustellenden Dienste. Diese Vision wurde von der Querschnittsgruppe „Architektur“ erarbeitet. Die Querschnittsgruppe setzt sich im Wesentlichen aus den Partnern des Think Tank „Plattformen“¹ (Universität Hildesheim/Software Systems Engineering, Universität Koblenz-Landau) und des KI-Accelerator (Siemens, KI-Protect, Lenze, Bitmotec) zusammen, umfasst aber auch interessierte bzw. assoziierte Partner wie Phoenix Contact.

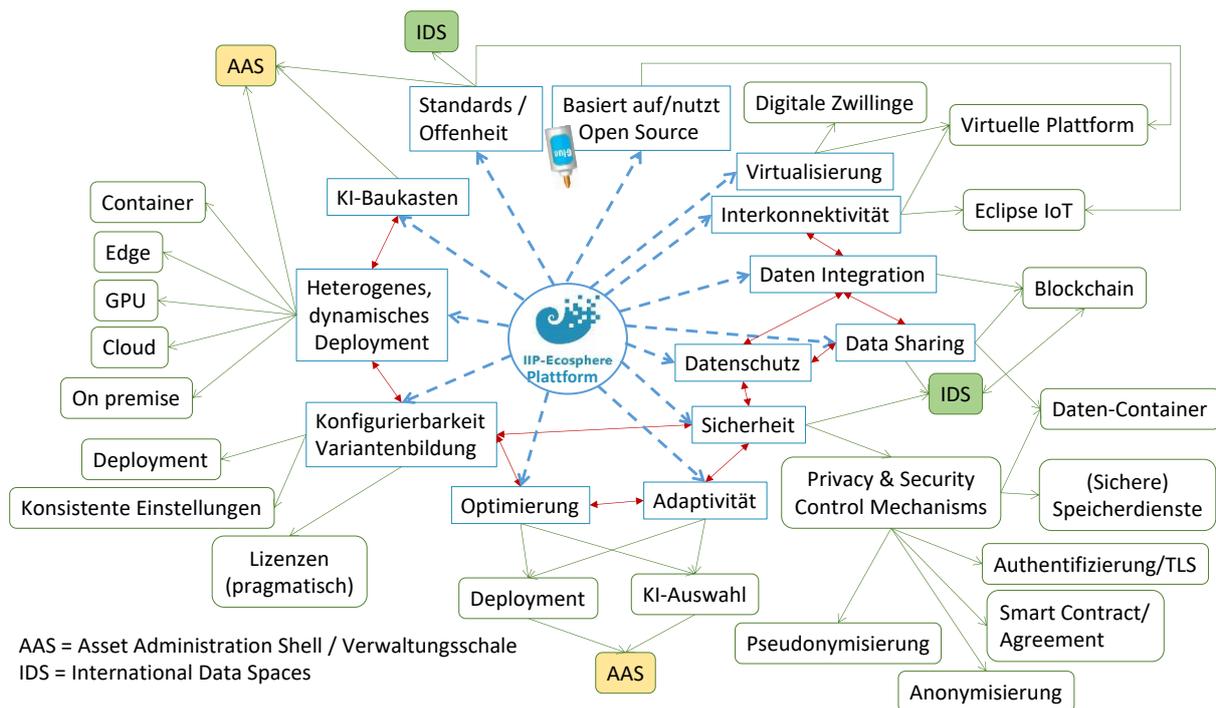


Abbildung 1: Mind-Map-Übersicht der angestrebten Plattform-dienste und -funktionalitäten

Die IIP-Ecosphere Plattform besteht aus (entgegen dem Uhrzeigersinn in Abbildung 1)

- Einem offenen und erweiterbaren **KI-Baukasten**, der wiederverwendbare KI-Bausteine in Form von Diensten bzw. Komponenten enthält. Die Plattform wird hier Schnittstellen und Mechanismen zur Verfügung stellen, um die KI-Dienste auf Ressourcen zu verteilen und mit Daten zu versorgen bzw. diese Daten nachfolgenden Komponenten zur Verfügung zu stellen – sowohl für das Training als auch für die Laufzeit/Prädiktion. Die konkreten Bausteine und deren spezielle Schnittstellen (wünschenswerterweise strukturiert nach KI-Verfahren), die auf die Plattform-Schnittstellen für Dienste und Daten aufsetzen, sind insbesondere mit den Demonstratoren und dem KI-Accelerator abzustimmen. Die Prinzipien der Schnittstellen, die durch individuelle Verwaltungsschalen (Asset Administration Shell, AAS) [26] repräsentiert werden, werden im Zuge der Architekturdefinition der Plattform festgelegt.
- Einem Mechanismus zum **heterogenen, dynamischen Deployment** ausgewählter Plattform-Funktionalitäten wie KI-Bausteinen auf verschiedene **Ressourcen**. Dabei entscheidet die Plattform aufgrund von Informationen wie Zusicherungen und Ressourcenbedarfe über die zu verteilenden Dienste bzw. Komponenten, die sich in Form von Verwaltungsschalen selbst

¹ Die Projektstrukturen werden auf <http://iip-eosphere.eu/> beschrieben.

beschreiben. Analog werden die möglichen Ziel-Ressourcen, wie beispielsweise Edge-Devices oder GPU-Server in Form von Verwaltungsschalen beschrieben. Als Deployment-Einheiten sollen Container (z.B. Docker-Container [5]) verwendet werden. Primär zielt die IIP-Ecosphere Plattform auf **vor-Ort/on-premise** Installationen. Eine **Cloud-basierte** Ausführung der Container ist optional und sollte in den Planungen berücksichtigt werden. Auch wenn wir prinzipiell den LNI-Empfehlungen zum Edge Deployment [20] folgen, geht IIP-Ecosphere mit diesen drei Deployment-Ebenen (Edge, on-premise Server und Cloud) wie in [36] detailliert über aktuelle Vorschläge hinaus.

- Zur systematischen und konsistenten **Konfigurierbarkeit** der Plattform wird die IIP-Ecosphere Plattform Techniken zur **Variantenbildung** einsetzen. Dies ermöglicht es, bereits vor der Ausführung der Plattform zu bestimmen, ob die jeweilige Plattformkonfiguration konsistent und lauffähig ist und vereinfacht damit Installations- und Wartungsarbeiten. Da in der Konfiguration ebenfalls die auf der Plattform laufenden Lösungen/Applikationen und deren benötigte Dienste/Komponenten beschrieben werden, ermöglicht die Konfiguration auch den applikationsgetriebenen Bau der Deployment-Einheiten. Zur Realisierung ist EASy-Producer [9] vorgesehen, ein flexibles Software Produktlinien-Werkzeug, das sowohl komplexe Konfigurationen modellieren, deren Konsistenz und Validität prüfen und Software-Anpassungen durch generative Verfahren realisieren kann als auch Mechanismen zur Steuerung von Laufzeit-Anpassung bzw. Adaptivität bereitstellt.
- Die Konfigurationsmodellierung/Konfigurierbarkeit wird durch (dynamische) Informationen aus den Verwaltungsschalen der eingesetzten Komponenten ergänzt. Dadurch wird eine **Optimierung** des heterogenen Deployments auf die verfügbaren Ressourcen aber auch des konkreten Einsatzes von KI-Komponenten in den Containern ermöglicht. Die Optimierung bezieht sich zunächst auf eine Zuordnung zum Plattform- bzw. Applikationsstart basierend auf (dynamische) Informationen aus den Verwaltungsschalen der Ressourcen.
- Die Optimierung des Deployments und der verwendeten KI-Dienste/Komponenten kann aber ebenso zur Laufzeit erfolgen, d.h., zur **Adaptivität** der Plattform führen. Hierfür sind geeignete Mechanismen zur Ableitung von Adaptionentscheidungen, zur Laufzeitanpassung von Containern, Komponenten und (KI-)Modellen zu untersuchen und zu realisieren.
- The Plattform wird diverse **Sicherheitsmechanismen** wie beispielsweise Speicherdienste mit verschiedenen Qualitätscharakteristiken (inklusive sichere bzw. Verschlüsselte Speicherdienste) zur bereitstellen und diese allen Diensten/Komponenten, insbesondere den Edge Geräten zur Verfügung stellen. (Externe) Datenverbindungen sollten optional durch Mechanismen der IDS² Referenzarchitektur [14] gesichert werden.
- Mechanismen zum **Datenschutz**. Artikel 25 DSGVO (Datenschutz durch Technikgestaltung und durch datenschutzfreundliche Voreinstellungen, PbD – Privacy by Design) schreibt vor, dass sowohl zum Zeitpunkt der Festlegung der Mittel für die Verarbeitung als auch zum Zeitpunkt der eigentlichen Verarbeitung geeignete technische und organisatorische Maßnahmen (z.B. Security & Privacy Controls [17], die dafür ausgelegt sind, die Datenschutzgrundsätze wie etwa Datenminimierung wirksam umzusetzen und die notwendigen Garantien in die Verarbeitung aufzunehmen) in einem System integriert werden müssen.

² International Data Spaces [13]

Die technische Maßnahmen (z.B. NIST Controls, siehe Abbildung 2) sind zu generisch, um direkt in ein Systemmodell integriert zu werden.

Privacy Control (NIST)	Design Strategy [52, 108]
AP - Authority and Purpose	
(AP-1) Authority to Collect	Restrict
(AP-2) Purpose Specification	Consent
AR - Accountability, Audit, and Risk Management	
(AR-1) Governance and Privacy Program	Audit, Log, Report, Uphold
(AR-2) Privacy Impact and Risk Assessment	Report, Supply, Create
(AR-3) Privacy Requirements for Contractors and Service Providers	Demonstrate
(AR-4) Privacy Monitoring and Auditing	Demonstrate
(AR-5) Privacy Awareness and Training	Report, Supply, Explain
(AR-6) Privacy Reporting	Report
(AR-7) Privacy-Enhanced System Design and Development	All strategies
(AR-8) Accounting of Disclosures	Notify, Log, Report
DI - Data Quality and Integrity	
(DI-1) Data Quality	Update, Retract
(DI-2) Data Integrity and Data Integrity Board	Demonstrate
DM - Data Minimization and Retention	
(DM-1) Minimization of Personally Identifiable Information	Minimize, Hide
(DM-2) Data Retention and Disposal	Minimize, Hide
(DM-3) Minimization of PII Used in Testing, Training, and Research	Minimize
IP - Individual Participation and Redress	
(IP-1) Consent	Consent
(IP-2) Individual Access	Choose, Update, Retract
(IP-3) Redress	Update, Retract
(IP-4) Complaint Management	Demonstrate
SE - Security	
(SE-1) Inventory of Personally Identifiable Information	Supply, Update, Maintain, Uphold
(SE-2) Privacy Incident Response	Control, Enforce
TR - Transparency	
(TR-1) Privacy Notice	Notify
(TR-2) System of Records Notices and Privacy Act Statements	Demonstrate
(TR-3) Dissemination of Privacy Program Information	Demonstrate, Inform
UL - Use Limitation	
(UI-1) Internal Use	Minimize, Hide, Uphold
(UI-2) Information Sharing with Third Parties	Minimize, Demonstrate

Abbildung 2: Die Zuordnung zwischen Datenschutz-Design-Strategien und NIST Privacy Controls.

Wir verwenden eine Auswahl von acht vorhandenen Datenschutz-Designstrategien (Minimize, Hide, Separate, Aggregate, Inform, Control, Enforce, Demonstrate), einschließlich ihrer konkreten Spezifikationen unter Verwendung von Teilstrategien, Datenschutz-Design Muster und datenschutzfreundliche Technologien (Privacy Enhancing Technologies, PET), wodurch die Realisierung der Maßnahmen vereinfacht wird. Wir verwenden die Feature-Modellierung um verschiedene Zuordnungen zwischen Strategien, Muster und PETs zu zeigen. Die Feature-Modellierung ermöglicht die Erfassung von Variabilität in einem System in Bezug auf Merkmale und Beziehungen zwischen ihnen. Ein Auszug aus dem resultierenden Feature Model ist in Abbildung 3 vorgestellt.

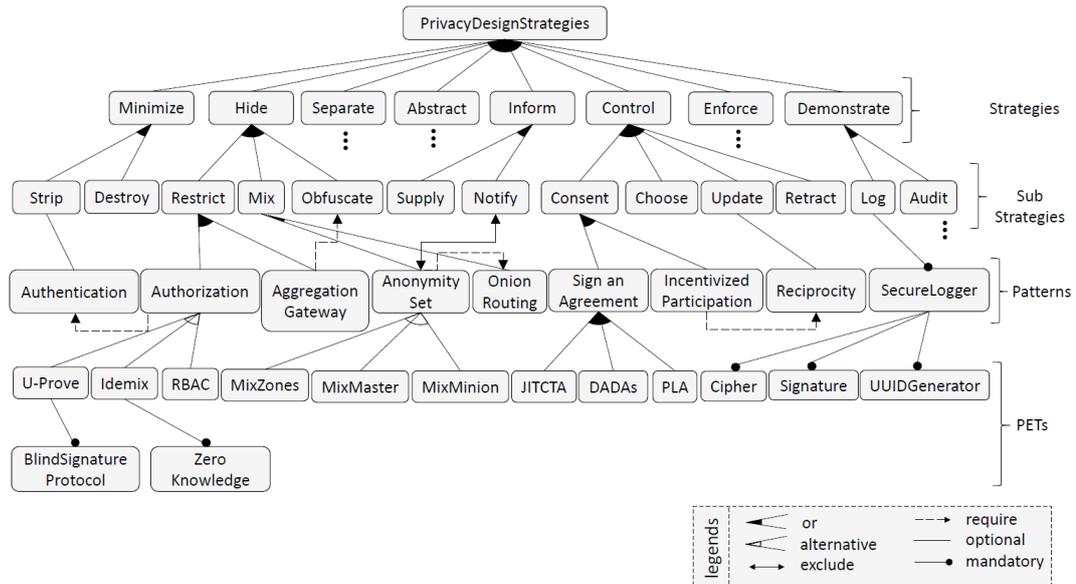


Abbildung 3: Ein Auszug aus dem Feature Model für Privacy Design Strategien

- Dienste und Komponenten für die **Datenintegration** aus heterogenen Quellen. Die Entwicklung ganzheitlicher Modelle zur Steigerung der Effektivität und Effizienz der Produktion erfordert eine systematische Integration der Daten aus verschiedenen Disziplinen wie z.B. Engineering, Produktionsplanung oder Manufacturing. Es gilt zu beachten, dass die Beschaffenheit der Daten stark anhand der zu integrierenden Quellen variieren. Beispielsweise müssen streaming-lastige Daten aus der OT-Domäne mit semantischen Daten aus der Fertigungsplanung kombiniert werden. Nach der Datenintegration müssen die Daten für Lösungen/Applikationen und KI-Algorithmen in passender Struktur, Granularität und Geschwindigkeit zur Verfügung gestellt werden.
- IIP-Ecosphere möchte den Austausch von Daten (**Data Sharing**) zwischen Partnern im Ökosystem fördern und unterstützen. Dafür soll die Plattform (in Absprache mit dem Think Tank "Data") Mechanismen zur Verfügung stellen, die den Datenaustausch unter Berücksichtigung von Datenschutz und Datensicherheit ermöglichen, z.B., durch Smart Contracts oder Datennutzungsvereinbarungen.
- Dienste für die **Integration von Daten** zur Laufzeit, d.h., die Kombination und Integration von Maschinen- und Produktionsdaten mit weiteren produktions-relevanten Informationen wie Auftragsdaten oder Konstruktionspläne. Diese Daten können wiederum der Künstlichen Intelligenz zur Verfügung gestellt werden um die dort vorgeschlagenen Entscheidungen zu verbessern.
- Protokolle zur **Interkonnektivität** auf verschiedenen Ebenen, deren Daten prinzipiell allen Komponenten der Plattform zur Verfügung gestellt werden sollen. Dazu gehören Protokolle bzw. Kommunikationsframeworks aus dem Industrie 4.0 Bereich, die die Verbindung zu Maschinen und bereits installierten Plattformen („südliche“ Interkonnektivität³) ermöglichen, wie z.B. OPC UA [22], Protokolle zur Verbindung von IIP-Ecosphere Plattformen (horizontale Interkonnektivität) wie auch die Anbindung von Steuersystemen (SCADA, MES) bzw. Visualisierungen, die durch die Funktionalität der IIP-Ecosphere-Plattform von KI-Verfahren

³ Die "geografische" Richtungsbezeichnung der „nördlichen“ bzw. oberen oder „südlichen“ unteren Schnittstelle ist aus dem Sprachgebrauch rund um IT/OT und Verwaltungsschalen entlehnt.

profitieren können („nördliche“ Interkonnektivität). Hierfür sind insbesondere Komponenten aus Eclipse IoT [8] vorgesehen.

- **Virtualisierung** der Plattform und ihrer Komponenten. Zunächst ist die IIP-Ecosphere Plattform als virtuelle Plattform ausgelegt, d.h., sie setzt auf vorhandene Dienste, Protokolle und Plattform-Installationen auf und ergänzt und erweitert diese. Es ist nicht das Ziel, existierende Plattformen wie in [33] untersucht zu ersetzen, sondern diese geeignet zu ergänzen. Zudem ist die Plattform selbst virtualisiert, von den Containern des dynamischen und heterogenen Deployments bis zu den zentralen Bausteinen der Plattform. Weiterhin soll die IIP-Ecosphere Plattform **digitale Zwillinge** [37] integrieren, sei es durch Bereitstellung geeigneter Frameworks, durch die Verwaltung bereitgestellter Container oder durch Verbindung zu installierten (virtuellen) Maschinen. Dabei müssen die zu verwendenden digitalen Zwillinge als Verwaltungsschalen der Plattform bekannt gemacht werden.

Die soeben beschriebene Kernfunktionalität der IIP-Ecosphere Plattform setzt sich in vielen Aspekten von existierenden Plattformen [33] ab, insbesondere durch die Kombination von on-premise Installation, heterogenem Deployment zu verschiedensten Ressourcen inklusive Edge-Devices, einem erweiterbaren KI-Baukasten, Optimierung und Adaption von Diensten, strombasierter Datenintegration, sowie innovativer Sicherheits- und Datenschutzmechanismen. Allerdings ist die skizzierte Grundfunktionalität auch sehr umfangreich. Daher werden die Partner auf vorhandene **Open Source** Komponenten zurückgreifen, so dass die Plattform an vielen Stellen insbesondere die Verbindungen zwischen diesen Komponenten (Glue-Code) und weitere Funktionalität beisteuert. Wir erwarten, dass Integrationsschwierigkeiten zwischen verschiedenen Komponenten durch geschickte Verwendung von Container-Technologie vermieden und darüber hinaus sogar zusätzliche Flexibilität erreicht werden kann.

Zur Strukturierung der folgenden Anforderungen in thematische Bereiche stellt Abbildung 4 eine initiale Übersichtsarchitektur dar, die die oben beschriebenen Dienste und Fähigkeiten abdeckt. Diese Übersichtsarchitektur wird in den folgenden Arbeitsschritten und Diskussionen mit den Partnern in eine umsetzbare Detailarchitektur verfeinert.

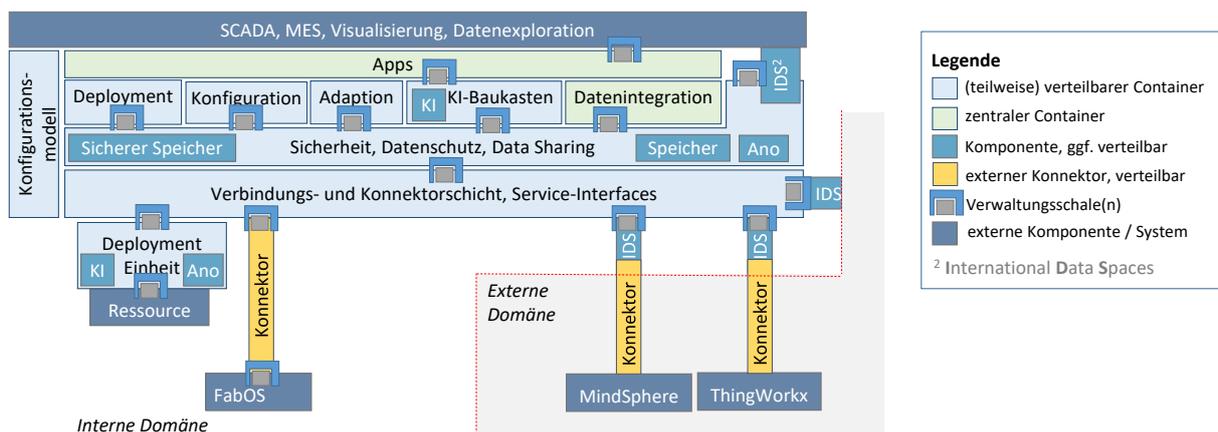


Abbildung 4: Initiale Übersichtsarchitektur der IIP-Ecosphere Plattform

Die Einbindung von aktiven Ressourcen in die IIP-Ecosphere Plattform erfolgt durch Schnittstellen und **Deployment-Einheiten** (Container). Die Schnittstellen deklarieren sowohl Ressourcen-Eigenschaften als auch Funktionalität in Form von Verwaltungsschalen. Die Container beinhalten die auf eine Ressource zugeschnittene Funktionalität. Maschinen bzw. Fabriken können direkt mittels (wiederverwendbarer) **Konnektoren** an die Plattform angebunden werden – je nach Sicherheitsdomäne ggf. unter Verwendung von IDS-Funktionalität. Die **Verbindungs- und Konnektorschicht** verwaltet die Konnektoren und kann auf dieser Ebene auch Verbindungen zwischen

mehreren installierten Instanzen der Plattform ermöglichen. Falls erforderlich können Teile der Verbindungs- und Konnektorschicht sowie Konnektoren selbst Teil einer Deployment-Einheit werden, z.B., um direkt mit der Ressource über Protokolle wie MQTT oder Informationsmodelle/Kommunikationsframeworks wie OPC UA zu kommunizieren. Zudem definiert diese Schicht die grundlegenden Service-Schnittstellen.

Die **Sicherheits-, Datenschutz und Data Sharing** Schicht⁴ verwaltet Berechtigungen und stellt zusätzliche Komponenten wie (sichere) Datenspeicher oder Anonymisierungs- bzw. Pseudonymisierungs-Komponenten (**Ano** in Abbildung 4) zur Verfügung. Während Datenspeicher eher als zentrale Dienste zu verstehen sind, sind Anonymisierungs- und Pseudonymisierungskomponenten verteilbar und können selbst in den Deployment-Einheiten verwendet werden. Ein Beispiel aus den IIP-Ecosphere Demonstratoren für die Anonymisierung ist die Erstellung von Werker-Profilen im Fertigungsprozess um damit die Maschinen-Parametrierung zu unterstützen. Hier ist die Frage, wie man diese persönlichen Daten einfach anonymisieren kann. Diese Schicht wird auch durch geeignete Mechanismen die Datenzugriffe auf Verwaltungsschalen und auch auf Verbindungs- und Konnektorebene steuern können. Weiterhin stellt diese Schicht auch abgesicherte externe Protokolle bereit, z.B. für die „nördliche“, plattformseitige, vertikale Integration.

Das **Deployment** ist für die Erzeugung und Bereitstellung angepasster Deployment-Einheiten zuständig. Dies erfolgt unter Verwendung des **Konfigurationsmodells**⁵ bzw. der **Konfigurationskomponente**. Die **Adaption** (implizit auch die Optimierung) verwendet insbesondere die Laufzeitinformationen, die aus Verwaltungsschalen in das Konfigurationsmodell übernommen werden und passt die Laufzeitkonfiguration so an, dass das Deployment bzw. die verwendeten KI-Komponenten optimal genutzt werden können. Um dies durchzusetzen, wird auf geeignete Verwaltungsschalen-Schnittstellen der Deployment-Einheiten bzw. der Ressourcen zurückgegriffen.

Die **Datenintegration** verknüpft zusätzliche Information wie Baupläne mit den Produktionsdaten. Die erweiterten Datenmodelle stehen der Plattform, insbesondere den KI-Diensten (und durch das Deployment auch den zugehörigen Ressourcen unter Berücksichtigung der Sicherheitsvorgaben) zur Verfügung. Die dabei erstellten Modelle können in geeigneten Datenspeichern abgelegt werden. Teile der Datenintegration könnten während der Architekturdefinition verteilbar werden, d.h., auch z.B. auf Edge-Geräten verfügbar sein.

Der **KI-Baukasten** stellt Schnittstellen für (verteilmare) KI-Dienste zur Verfügung und implementiert bzw. integriert Beispiel-KI-Dienste (insbesondere über das Arbeitspaket KI-Accelerator). KI-Dienste mit erhöhtem Ressourcenbedarf, z.B. zum Training, können beispielsweise auf GPU-Server verteilt und ausgeführt werden, während KI-Komponenten mit geringem Ressourcenbedarf oder hoher Ausführungsgeschwindigkeit auf Edge-Ressourcen ausgeführt werden können. Ein genanntes Beispiel aus den IIP-Ecosphere Demonstratoren ist hier, dass die Forward-Propagation eines Neuronalen Netzwerks auf einem Edge-Gerät und die Backward-Propagation zum Training auf einer geeigneten zentralen Ressource erfolgt. KI-Dienste können hinzugefügt werden und sind dann analog zu den bereits mitgelieferten Diensten/Komponenten verteilbar.

Unter Verwendung des Konfigurationsmodells können die Dienste und Komponenten der Plattform zu Lösungen bzw. Applikationen (**Apps**) zusammengestellt werden und ggf. mit zusätzlichem Programm-Code ergänzt werden. Eine Applikation kann eigene applikations-spezifische Dienste definieren. Die

⁴ Sicherheit und Datenschutz reichen typischerweise über mehrere Schichten. Wir stellen hier nur die besonderen Funktionen für IIP-Ecosphere und nicht etwa den Einfluss auf die Konnektoren dar.

⁵ Analog zur Sicherheit reicht eine konsistente Konfiguration und das zugehörige Modell üblicherweise über alle Schichten einer Architektur. Wir stellen hier nur die Komponente dar, mit der der Benutzer die Plattform konfigurieren kann.

auszuführenden Applikationen bestimmen die benötigten KI- und Plattform-Komponenten (wie z.B. den Anonymisierungsbedarf nahe an Produktionsmaschinen) und definieren damit die Grundlage für das Deployment, die Optimierung und die Adaption.

Externe Schnittstellen können zur Kommunikation von IIP-Ecosphere Plattform-Instanzen, zum Einspeisen weiterer Daten oder zur Steuerung der Produktion verwendet werden. Die externen Komponenten/Systeme können dabei (unter Berücksichtigung der Sicherheitsvorgaben) auf KI-Dienste, Plattformdienste und Daten einer IIP-Ecosphere-Plattform zugreifen, diese in Grenzen verändern (um z.B. einen Produktionsauftrag zu starten) bzw. diese visualisieren.

3 Anforderungen

Wir formulieren die Anforderungen an die IIP-Ecosphere Plattform nach der Idee der **kontrollierten textuellen Anforderungen** bzw. der **Anforderungsschablonen** [32]. Eine Anforderung wird dabei mit einem (möglichst) einfachen Satz beschrieben. Wir verwenden (sinngemäß) das folgende Muster:

<Bedingung>?
 <Aktor/Komponente>
 <Klassifikation>
 <wem>?
 <Funktionalität oder Eigenschaft>
 (zur Verfügung stellen | beinhalten | sein ...). <Quelle>?

Dabei sind Elemente in spitzen Klammern wie <Aktor/Komponente> geeignet zu ersetzen. Elemente mit einem „?“ am Ende sind optional, während die letzte Zeile alternative Satzenden andeutet. Nach dem Satzende soll zur Nachvollziehbarkeit die Quelle der Anforderung als Kürzel angegeben werden.

Als Klassifikation verwenden wir „muss“ (deutet auf höchste Priorität hin), „soll“ (für mittlere Priorität, wünschenswert) und „kann“ (für niedrige Priorität, nice-to-have). Plural-Formen können natürlich sinngemäß verwendet werden. Neben dieser Klassifikation ist eine detailliertere Priorisierung erforderlich. Eine initiale Priorisierung der Anforderungen geben wir in Kapitel 4.

Anforderungen, die logisch aus mehreren Teilen bestehen, werden durch eine Hauptanforderung sowie ein oder mehrere Unterpunkte (jeweils wieder als kontrollierte textuelle Anforderungen) beschrieben. Falls erforderlich, kann eine Anforderung durch weitere Sätze (gekennzeichnet durch „Erläuterung:“) näher beschrieben werden.

Wir definieren zunächst notwendige Begriffe in Abschnitt 3.1, definieren in Abschnitt 3.2 die Quellen für die Anforderungen sowie deren Kürzel auf, besprechen in Abschnitt 3.3 generelle Anforderungen, und besprechen dann in Abschnitt 3.4 in jeweiligen Unterabschnitten die Anforderungen für die Konzepte aus Abbildung 4 (der Beschreibungsreihenfolge dort folgend). Leider kann es sein, dass aufgrund von unklarem Formatierungsverhalten in Microsoft Word Referenzen zu Unteranforderungen nicht korrekt dargestellt werden, insbesondere bei Anforderungen mit höherer Identifikationsnummer.

3.1 Terminologie

Wir fassen nun kurz die im Folgenden verwendeten Begriffe und deren Bedeutung zusammen.

- In der industriellen Produktion werden die verwendeten Geräte und Netzwerke oft in zwei Bereiche unterschieden [15], die
 - Operationale Technologie (Operational Technology, **OT**) für die Produktionssteuerung und -überwachung, insbesondere bestehend aus Fertigungsmaschinen, Steuergeräten, Edge-Geräten, etc.
 - Informationstechnologie (Information Technology, **IT**) für die „höhere“ Steuerung der Produktion unter Verwendung üblicher Geräte, wie Server oder Cloud-Technologie.
- Jedes Asset im Industrie 4.0 Kontext soll mit einer sogenannten **Verwaltungsschale** umgeben werden, die jeweils geeignet ist, das Asset bezüglich der Anwendungsfälle der Industrie 4.0 minimal, aber hinreichend zu beschreiben [40]. Verwaltungsschalen können sowohl

funktionale Eigenschaften, Schnittstellen als auch Qualitätseigenschaften der beschriebenen Assets repräsentieren.

- Ein **Konfigurationsmodell** dient dazu Konfigurationsmöglichkeiten und –abhängigkeiten zu beschreiben und kann auf Software oder auch auf Hardware angewendet werden. Ein Konfigurationsmodell definiert den Raum aller Konfigurationsmöglichkeiten und beschreibt damit die möglichen ausprägbaren Instanzen einer Software bzw. einer Hardware. Eine **Konfiguration** beschreibt eine einzelne Instanz. Eine Konfiguration ist gültig, wenn alle Abhängigkeiten erfüllt sind. Derartige Prüfungen können durch Modell-Analysen wie z.B. Constraint-Reasoning realisiert werden. Eine Konfiguration kann (oft automatisch) **instantiiert** werden, d.h., die zugrundeliegende Software/Hardware wird so angepasst, z.B., durch Auslassen von Alternativen, dass die konfigurierte Instanz realisiert wird. Anpassungen können sowohl **vor der Laufzeit** (diese sind dann teilweise irreversibel) als **auch zur Laufzeit** möglich sein und sich sogar gegenseitig beeinflussen. Beispiele für Konfigurationsmodelle aus dem Bereich der **Software-Produktlinien (SPL)** sind Feature-Modelle [3, 18], siehe Abbildung 3 für ein Beispiel, Entscheidungsmodelle [34] oder auch topologische Modelle [9].
- Eine **virtuelle Plattform** setzt auf vorhandene Dienste und Plattformen auf und integriert diese, um weitere, höherwertige Dienste anzubieten. Die IIP-Ecosphere Plattform wird als virtuelle Plattform realisiert, um die bereits existierenden Dienste vorhandener Industrie 4.0-Plattformen zu verknüpfen ohne diese erneut entwickeln zu müssen.
- Wir verwenden im Folgenden den Begriff „**Plattform**“ als Kurzform für die virtuelle IIP-Ecosphere Plattform.
- Wir verwenden den Begriff „**Dienst**“ bzw. „**Service**“ synonym für eine „**Komponente**“ mit expliziter, definierter Schnittstelle.
- Eine **Plattform-Service/Komponente** bezeichnet generell alle erwähnten Komponenten bzw. Komponenten-Arten der IIP-Ecosphere Plattform, insbesondere (verteilbare) KI-Komponenten, Sicherheitskomponenten, Konnektoren, etc.
- Bei einer **Pseudonymisierung** wird ein Name oder ein anderes Identifizierungsmerkmal durch ein **Pseudonym** (zumeist ein Code, bestehend aus mehrstelligen Buchstaben- oder Zahlenkombinationen) ersetzt, um die Feststellung der Identität des Betroffenen auszuschließen oder wesentlich zu erschweren [39].

3.2 Quellen für Anforderungen

Als Quellen für Anforderungen wurden die folgenden Stakeholder und Dokumente mit einbezogen:

- Der **Förderantrag** zu IIP-Ecosphere, insbesondere das Projekt-Konzept mit der initialen Referenz-Architektur der Plattform, sowie die einzelnen Teilvorhabensbeschreibungen.
- **Diskussionen** mit verschiedenen **Teilprojekt von IIP-Ecosphere**, die als Nutzer der Plattform verstanden werden können. Dazu gehören insbesondere die vier IIP-Ecosphere Demonstratoren und das Experimentierfeld, aber auch die Think Tanks „Daten“ bzw. „KI & Produktion“. Die Querschnittsgruppe hat einen Fragebogen zur Abfrage der Bedarfe (insbesondere der Demonstratoren) entwickelt und diese verteilt. Die Ergebnisse wurden in die Anforderungen integriert und werden entsprechend gekennzeichnet. Der Fragebogen ist in Abschnitt 1 dargestellt.
- **Diskussionen** der **Querschnittsgruppe „Architektur“** in IIP-Ecosphere. Die Querschnittsgruppe führt Partner mit besonderem Interesse an Plattformenthemen zusammen, insbesondere die Partner des Think Tank „Plattformen“ und Partner aus dem KI-Accelerator. Die Querschnittsgruppe hat sowohl die gemeinsame Vision der Plattformdienste (siehe Abschnitt 2) entwickelt, die Übersicht über aktuelle Industrie 4.0 Plattformen [33] als auch die zugrundeliegenden Anforderungen aus Benutzungssicht (usage view) [36] erstellt. Weiterhin

wurden die grundlegenden Anforderungen in diesem Dokument in der Querschnittsgruppe gesammelt und diskutiert.

- **Standards und gesetzliche Vorgaben** können an einigen Stellen Anforderungen festlegen bzw. diese beeinflussen.
- Die bereits in Abschnitt 1.2 erwähnte **Übersicht aktueller Industrie 4.0 Plattformen** [33] dient insbesondere zur Untermauerung, zur Validierung und zur Abgrenzung der Anforderungen.
- Ebenso dient die Sammlung von Anforderungen aus Benutzungssicht [36] als Grundlage für die Anforderungen in diesem Dokument.
- Diskussionen der Anforderungen mit **weiteren Stakeholdern**, z.B. dem Industrial Advisory Board von IIP-Ecosphere oder externen Partnern im Rahmen des Arbeitskreises „Plattformen“.

Im Folgenden kennzeichnen wir die Herkunft einzelner Anforderungen mit:

Quelle	Kennzeichen
Antrag	[A]
Demonstratoren	[D]
Think Tanks	[T]
Experimentierfeld	[E]
Querschnittsgruppe „Architektur“	[Q]
Standards, gesetzliche Vorgaben	[S]
Plattformübersicht [33]	[P]
IIP-Ecosphere Industrial Advisory Board (IAB)	[I]
Benutzungssicht / Usage View [36], siehe Abschnitt 4	[U]
IIP-Ecosphere Partner (initiativ, z.B., nach Klausurtagung)	[V]
IIP-Ecosphere Arbeitsgruppen, Wettbewerbsphase, externe Diskussionen	[X]

Eine Mehrfachkennzeichnung ist möglich wenn eine Anforderung von verschiedenen Quellen genannt wurde.

3.3 Generelle Anforderungen

In diesem Abschnitt werden generelle und übergreifende Anforderungen an die Plattform festgehalten.

- R1. Die Plattform **muss** herstellerunabhängig bzw. technologieneutral sein. [A]
- R2. Die Plattform **muss** standardorientiert sein. [A]
- R3. Die IIP-Ecosphere-Plattform **muss** als virtuelle Plattform realisiert werden. [A]
Erläuterung: Eine virtuelle Plattform integriert bestehende Dienste/Plattformen, um höherwertige Dienste, insbesondere KI-basierte Dienste, anzubieten.
- R4. Die Plattform **soll** komponenten-/service-orientiert entworfen und realisiert werden. [Q, D]
- Komponenten/Dienste **müssen** mit ihrer Schnittstelle (Eingabe, Ausgabe) beschreiben sein. [Q, U]
 - Komponenten/Dienste **müssen** mit Meta-Informationen (Version, Kategorisierung) ausgestattet sein. [T, U]
 - Komponenten/Dienste **müssen** einen abfragbaren Zustand haben. [U]
 - Die Ausführung von Diensten **muss** überwacht werden (Monitoring). [U]
 - Die Dienst-Überwachung **soll** parametrisierbar sein. [U]
 - Die Dienst-Überwachung **soll** von anwendungsspezifischen Diensten realisiert werden. [U]
 - Überwachte Werte **sollen** durch Trigger geprüft und weitergemeldet werden. [U]
- Erläuterung:* Ein möglichst modularer Aufbau zum einfachen Einbinden in existierende Systeme ist erwünscht. Der Zustand einer Komponente/eines Dienstes signalisiert beispielsweise „in Betrieb“, „passiv“, „Fehler“ oder „in Adaptierung“. Monitoring kann durch

verschiedene Realisierungstechniken wie Code-Injektion, Proben oder Dienste (R4.e) umgesetzt werden. Werte sollen mit standardisierten (R4.e) bzw. applikationsspezifischen (R4.f) Mechanismen geprüft und ggf. an geeignete Stellen weitergeleitet werden, so dass Maßnahmen ergriffen werden können (siehe auch R118.a sowie Abschnitte 3.12 und 3.14)

R5. Die Plattform **soll** aus Open Source Komponenten aufgebaut werden. [A]

Erläuterung: Dabei sind insbesondere die Lizenz-Vorgaben von IIP-Ecosphere zu berücksichtigen, insbesondere, dass copy-left OpenSource Lizenzen im industriellen Umfeld als problematisch angesehen werden und daher weitgehend vermieden werden sollen.

R6. Die Plattform **soll** weitere optionale Komponenten einbinden können, die nicht R1 oder R4 entsprechen. [Q]

Erläuterung: Beispiele sind kommerzielle Komponenten der Partner.

R7. Komponenten/Container-Schnittstellen sowie deren Fähigkeiten **müssen** durch Verwaltungsschalen beschrieben werden. [Q, U]

a. Die Beschreibungskonzepte für Verwaltungsschalen **sollen** einheitlich sein. [D]

Erläuterung: Die Beschreibungskonzepte sollen mit den Stakeholdern, insbesondere den Demonstratoren abgestimmt werden. Verwendung von Standards bzw. Standardisierung der Beschreibungskonzepte ist wünschenswert.

Erläuterung: Soweit möglich sollen Verwaltungsschalen auch für die Komponenten/Service-Kommunikation verwendet werden. Allerdings kann es aus Geschwindigkeits-/Volumengründen (R70) erforderlich sein, einheitlich auf eine alternative Kommunikationsform auszuweichen, die dann Industrie 4.0 Protokolle verwenden soll (R14).

R8. In der Plattform **sollen** SPL-Ansätze zur Variantenbildung verwendet werden. [A]

a. Die Plattform **muss** ein integriertes Konfigurationsmodell für Applikationen, Dienste und Plattformeigenschaften enthalten. [A, U]

b. Die SPL-Ansätze **müssen** die automatisierte Validierung des Konfigurationsmodells ermöglichen. [A, U]

c. Die SPL-Ansätze **müssen** automatisierte Ableitung von Plattform-Instanzen unterstützen. [A]

Erläuterung: SPL-Ansätze sollen verwendet werden, wo dies sinnvoll bzw. wissenschaftlich herausfordernd ist, z.B., um die Integration optionaler Komponenten systematisch zu realisieren oder diese aus einer Plattforminstanz völlig entfernen zu können. Das unterliegende Konfigurationsmodell (R8.a) muss automatisch validierbar sein (R8.b), üblicherweise unter Berücksichtigung applikationsspezifischer Bedingungen/Constraints und automatisiert instantiierbar sein (R8.c).

R9. Die Plattform **soll** die Verfügbarkeit ihrer Dienste sicherstellen. [Q]

Erläuterung: Dazu gehört insbesondere die produktions-nahe (ggf. autonome) Ausführung komplexer Programme (Industrie-PCs, Edge-Geräte).

R10. Die produktionskritischen Plattformoperationen **sollen** die Anforderung der weichen Echtzeit (Antwortzeit < 100ms) erfüllen. [T, Q]

Erläuterung: Sicherheitskritische harte Echtzeitanforderungen sind insbesondere in der IT-Domäne out of scope. Die Plattform kann generell nur weiche Echtzeitanforderungen für die von ihr bereitgestellten Operationen einhalten, d.h., für die Operationen für die die Plattform verantwortlich ist. Für externe Erweiterungsfunktionen, Komponenten bzw. Dienste, die auf Schnittstellen der Plattform aufbauen, ist das entweder nicht möglich und die Verantwortung dafür ist extern oder eine Einhaltung ist nur durch Adaptation bzw. im Extremfall Terminierung der jeweiligen Komponente / des jeweiligen Dienstes möglich.

R11. Die Plattform **muss** eine Dokumentation ihrer bereitgestellten Funktionen bereitstellen. [Q]

a. Die Plattform **muss** eine Entwicklerdokumentation bereitstellen.

b. Die Plattform **muss** eine Anwenderdokumentation bereitstellen.

c. Die Plattform **muss** eine Dokumentation der Ausführung bereitstellen.

Erläuterung: Dabei handelt es sich sowohl um Dokumentation für Administratoren und Bediener (R11.b) wie z.B. ein Benutzerhandbuch, als auch um Dokumentation für Programmierer (z.B. Code-Dokumentation, R11.a). Verwaltungsschalen sollen geeignet dokumentiert sein und stellen damit (teilweise) eine maschinenlesbare Dokumentation dar. Weiterhin sollen Verwaltungsschalen geeignete Dokumentationen referenzieren. Verschiedene Operationen, insbesondere Änderungen von Ressourceneigenschaften, Diensten oder Applikationen müssen zur Laufzeit der Plattform dokumentiert werden (Logging, R11.c).

R12. Datenverarbeitungsschritte **sollen** einheitlich dokumentiert werden. [Q]

a. Falls ausreichend Informationen über die verwendeten Komponenten verfügbar ist, **kann** die Plattform die Dokumentation automatisch aus dem Konfigurationsmodell ableiten. [Q]

Erläuterung: Diese Dokumentation ist aus Datenschutzgründen erforderlich. Die Information könnte z.B. aus den Verwaltungsschalen der aktiven Komponenten/Dienste automatisiert abgeleitet werden.

3.4 Konnektoren und Verbindungen/Interkonnektivität

Konnektoren stellen sowohl Außen- als auch Innenverbindungen dar. Konnektoren können Zugriff auf Daten bieten aber auch Steuerinformationen transportieren.

R13. Die Plattform **muss** sich mit anderen Akteuren verbinden können. [A]

a. Die Plattform **muss** sich direkt mit Industrie 4.0 Geräten verbinden können. [A]

b. Die Plattform **muss** sich mit anderen Industrie 4.0 Plattformen verbinden können. [A]

c. Die Plattform **muss** sich mit anderen Instanzen der IIP-Ecosphere Plattform verbinden können [A]

d. Die Einbindung der genannten Akteure **muss** dynamisch zur Laufzeit möglich sein. [D]

Erläuterung: R13.b und R13.c sind primär relevant für eine virtuelle Plattform (R3). Dies impliziert, dass Geräteinformationen (R25.b), Ressourcenallokationen (Abschnitt 3.12) und Adaptionen (Abschnitt 3.14) über Plattformen hinweg je nach Freigaben möglich sein können. R13.a spiegelt den speziellen Fokus auf Edge-Computing wieder, den die Projektpartner mit der Plattform-Vision vereinbart haben.

R14. Die Plattform **muss** Konnektoren anbieten, um die offene und flexible Einbindung von Industrie 4.0/IoT-Standards und (internen) Datenquellen zu ermöglichen. [A, Q]

a. Die Plattform **muss** OPC UA (Informationsmodell) und MQTT (Protokoll) unterstützen. [D]

b. Die Plattform **kann** TCP/IP unterstützen. [T]

c. Die Plattform **kann** Bluetooth LE unterstützen. [D]

Erläuterung: Die Plattform soll möglichst keine Protokolle fix vorgeben, auch nicht für den internen Transport. Hier sollen (aus Normierungs- und Lizenzgründen) Protokolle eingesetzt werden können, die im jeweiligen Anwendungsumfeld als relevant und geeignet bewertet werden, auch externe Protokolle bzw. Protokolle von dritten. Dies betrifft ebenfalls die Transportformate sofern diese nicht durch das Protokoll festgelegt sind, z.B. bei MQTT ist der Payload nicht definiert. Weiterhin betrifft dies ebenso die (ggf. strombasierte) zeitkritische Kommunikation nach R10. R14.b und R14.c sind hier als Beispiele für optionale/zusätzliche Protokolle genannt.

R15. Konnektoren für eine Art von Daten, z.B., Geräteverbindungen **sollen** einheitlich gestaltet sein. [Q]

R16. Die Einbindung von Konnektoren in die Plattform **muss** offen und erweiterbar sein. [A]

R17. Konnektoren **sollen** nicht an andere Plattform-Funktionalitäten gebunden sein, so dass Konnektoren bei Bedarf verteilbar sind. [A]

- a. Konnektoren **sollen** im Konfigurationsmodell beschrieben werden. [U]
- b. Konnektoren **können** durch die Plattform verwaltet werden. [U]
- c. Konnektoren **können** parametrisierbar sein. [U]

Erläuterung: Durch die Kombination von Konfigurationsmodell und Verwaltung durch die Plattform ergibt sich wie bei R20 die Möglichkeit zur aktiven Wiederverwendung von Konnektor-Informationen.

R18. Konnektoren **können** durch geeignete Verfahren abgesichert sein (optional) und dabei Sicherheits- bzw. Herkunftsinformationen in die Daten einfügen. [A, Q]

R19. Die Plattform **soll** intern nur wenige unterschiedliche Datenformate für Produktionsdaten verwenden. [Q]

- a. Die Plattform **soll** (strukturierte) Eingabedaten von der „südlichen“ Schnittstelle wie Messergebnisse (Float32, Int16/32, Bool32, String, Arrays davon), Zeit (Datum, Dauer), Prüfergebnis, Prüfplatz, Seriennummer, Codes/Kategorien, Ereignisse, Alarme oder Fertigungszeiten unterstützen. [D]

Erläuterung: Ein Datensatz kann beispielsweise 50 Werte verschiedener Messergebnis-Typen in einem Prozessschritt von 20-30 s beinhalten.

- b. Die Plattform **soll** (strukturierte) Eingabedaten von der „nördlichen“ Schnittstelle wie Produkt, Produktnummer, Produktkategorie, Sicherheitsbestand, durchschnittlicher Abgang, Losgröße, Bearbeitungszeit, Rüstzeit, Liefertermin (Plan vs. Ist), Durchlaufzeit (Plan vs. Ist), Störungen und Störungshäufigkeit mit Kategorisierung (aber ohne Personenbezug), Maschinenverfügbarkeiten (MTTR, prozentual) unterstützen. [D]
- c. Die Plattform **soll** weitere Datenformate wie JSON/XML über Restful API bieten. [D]
- d. Die Plattform **soll** (strukturierte) Ausgabedaten auf der „nördlichen“ Schnittstelle wie optimale Reihenfolge, optimale Kapazitätsauslastung, optimale kurzfristige Personalplanung, alternative Bearbeitungsstationen bei Störung, Fehlerklassifikation oder Handlungsempfehlung anbieten. [D]
- e. Die Plattform **soll** Ausgabedaten pro Verarbeitungstakt von 5 s ermöglichen. [D]
- f. Die Plattform **soll** Mechanismen zur Formatanpassung bzw. Formatkonversion bereitstellen, die im Konfigurationsmodell beschrieben werden. [Q, D]

Erläuterung: Andere Ansätze wie [21] sprechen hier von einem „Semantic Model“ oder einem „Semantic Mapping“, oft unter Rückgriff auf Ontologien aus dem Bereich Semantic Web. Da für eine Passung der Komponenten, Dienste oder Deployment-Einheiten (siehe auch Abschnitt 3.5) die Formate im Konfigurationsmodell beschrieben sein müssen und EASy-Producer [9] sehr reiche Constraint- bzw. Ausdruckssprachen besitzt, sind wir davon überzeugt, das sich mit den vorhandenen Mitteln und dem Konfigurationsmodell effiziente Übersetzungs- und Anpassungsmechanismen bzw. –konnektoren automatisiert ableiten lassen.

- g. Die Plattform **soll** Mechanismen zur Anpassung bzw. Manipulation von Daten-Meta-Informationen bereitstellen, die im Konfigurationsmodell beschrieben werden. [T]

Erläuterung: Bei der Anpassung (R19.f) oder Integration (Abschnitt 3.10) von Daten ist es erforderlich, die Meta-Informationen (R79) auf dem aktuellen Stand zu halten.

Erläuterung: Wünschenswert wäre ein generisches Datenformat, das durch das Konfigurationsmodell (z.B. für individuelle Applikationen) bzw. abgeleitete Metainformationen beschrieben ist, so dass beispielsweise Sicherheitsmechanismen auf den Daten operieren können. Hierbei sind übliche Modellierungen für Verwaltungsschalen sowie vergleichbare Mechanismen von OPC UA [22] bzw. oneM2M [21] zu berücksichtigen. Dies

gilt ebenso für die Mechanismen zur Formatanpassung in R19.f. Anforderungen für personenbezogene Daten wie R48 sind ebenso zu berücksichtigen.

- R20. Die Plattform **muss** anwendungs-spezifische Datenpfade ermöglichen. [Q, D]
- Die Datenpfade **müssen** im Konfigurationsmodell definiert werden können. [Q]
 - Die Datenpfade **können** Eigenschaften/Parameter haben. [U]
 - Die Datenpfade **sollen** durch die Plattform verwaltet werden. [U]

Erläuterung: Datenpfade (in [36]) „Relationen“ genannt) verbinden Konnektoren, Komponenten und Services. Datenpfade legen fest, welche Format-Anpassungs-, KI-, Verarbeitungs- oder Speicherdienste bzw. -komponenten Daten auf dem Weg von Quellen, z.B. Sensoren bis hin zu Aktoren oder zur Anwendung bzw. Visualisierung durchlaufen. Dabei definieren Datenpfade, welche Eingaben Voraussetzung für die Verarbeitung sind und welche Ausgaben durch die Verarbeitung geliefert werden. Da die Anwendungen, Datenformate und Komponenten/Dienste im Konfigurationsmodell beschrieben werden, ist es für eine konsistente Funktionsweise der Plattform wichtig, auch die Datenpfade zu beschreiben. Inspiriert von anderen Ansätzen wie [42] macht die aktive Verwaltung von Datenpfaden und damit deren Wiederverwendung durch die Plattform Sinn, auch wenn die Informationen aus Konsistenzgründen im Wesentlichen im Konfigurationsmodell enthalten sein sollten (R20.a).

- R21. Die Konnektoren bzw. Verbindungen **sollen** wenig Einfluss auf den Datendurchsatz haben. [Q, D]
- R22. Die Plattform **soll** einen Datendurchsatz von 500 GByte pro Jahr ermöglichen. [D]

Erläuterung: Diese Anforderung wurde ohne weitere Details von einem Demonstrator im Rahmen der Konnektoren genannt. Wir interpretieren diese als den Durchsatz am Dateneingang ohne Aggregation oder Filterung. Es ist wichtig darauf hinzuweisen, dass der Datendurchfluss bei Demonstratoren um Größenordnungen höher liegen kann, z.B., vor und nach der Datenintegration (R91) oder, wie in einer Präsentation angemerkt, bei 50 ms Takt je nach Produktionsschritt zwischen 5 GByte und 600 GByte pro Tag.

3.5 Heterogenes, dynamisches Deployment

Das Deployment von Plattform-Komponenten ermöglicht der IIP-Ecosphere Plattform, vorhandene und zugreifbare Ressourcen (z.B., Edge, Server, GPU, Cloud) flexibel zu nutzen. Da üblicherweise verschiedene Ressourcen mit unterschiedlichen technischen Eigenschaften eingesetzt werden, ist ein heterogenes Deployment mit (möglichst wenig technischer) Abstimmung auf herstellerspezifische Eigenschaften erforderlich. Dies führt zu einer Abstraktion des Betriebssystems und der relevanten Funktionen und vereinheitlicht so die Ausführungsumgebung für die Deployment-Einheiten. Aus technischer Sicht ist eine Deployment-Einheit ein Container („ECS runtime“ in [36], für Edge-Geräte auch „edge runtime“ in [20]). Prinzipiell können mehrere Container auf einer Ressource ausgeführt werden (die Menge aller Container bzw. „edge runtimes“ wird in [20] als „edge infrastructure“ bezeichnet). Das Deployment ist insofern dynamisch, da die Container möglichst automatisch von der Plattform basierend auf dem Konfigurationsmodell bzw. den Applikations-Konfigurationen erstellt und bereitgestellt werden sollen. Durch die automatisierte Bereitstellung kann die Anzahl der (IIP-Ecosphere) Container pro Ressource minimiert werden, wobei nicht ausgeschlossen ist, dass die Abstraktion der Ausführungsumgebung durch weitere, herstellerspezifische Container realisiert wird, die mit den IIP-Ecosphere Containern kommunizieren. Die Funktionen zur Erstellung, die Bereitstellung und die Zuteilung der Container (siehe auch Abschnitt 3.12) bildet das Management des Deployments (in Abbildung 4 kurz „Deployment“ genannt, in [20] als „edge management system“ bezeichnet).

R23. Die Plattform **muss** dynamische Deployment Einheiten unterstützen. [Q, A]

Erläuterung: Ein möglichst modularer Aufbau zum einfachen Einbinden in existierende Systeme ist erwünscht. Lokal sollen möglichst schlanke Module ohne großen Plattform-Overhead ermöglicht werden. [D]

R24. Die Plattform **muss** ein Deployment von Plattform-Komponenten auf unterschiedliche Arten von Ressourcen bzw. Hardware unterstützen. [Q, A, U]

Erläuterung: Lokale Verfahren sind für die Prozesssteuerung nötig (OT), da die Latenz z.B. zu Cloud-Systemen sonst zu hoch ist. Produktionssteuerung dahingegen kann zentralisiert (IT) erfolgen. [D]

R25. Die Eigenschaften bzw. Funktionalitäten einer Ressource **müssen** als Verwaltungsschale beschrieben werden. [Q, U]

- a. Die Verwaltungsschale einer Ressource **soll** durch eine Ressourcen-Abstraktionsschicht („ECS runtime“ in [36]) realisiert werden. [Q, U]
- b. Die Verwaltungsschale verfügbarer Ressourcen **muss** der Plattform bekannt gemacht werden. [Q, U]
- c. Die Plattform **muss** die verfügbaren Ressourcen verwalten. [P, U]
- d. Die Plattform **kann** Verfahren anbieten um die Verwaltung der verfügbaren Ressourcen zu vereinfachen. [P, U]
- e. Die Verwaltungsschale einer Ressource **muss** statische Eigenschaften der Ressource beschreiben. [Q]
- f. Die Verwaltungsschale einer Ressource **muss** dynamische Eigenschaften der Ressource beschreiben. [Q]
- g. Die Verwaltungsschale einer Ressource **muss** Funktionen zum Deployment von Deployment Einheiten enthalten. [Q]
- h. Die Verwaltungsschale einer Ressource **soll** Funktionen zum Austausch von Deployment Einheiten zur Laufzeit enthalten. [Q]

Erläuterung: Dies ermöglicht eine Adaption des Verhaltens. Die Funktionalität könnte bereits implizit in R25.a enthalten sein oder sogar aufgrund des Overheads im Produktionsumfeld nicht anwendbar sein.

Erläuterung: Diese Anforderung bezieht sich insbesondere auf den „Device Description Store“, das „Device Configuration Tool“ und die „ECS runtime“ in [36], d.h., auch auf die Abstraktion der Herstellerabhängigkeit (R25.a) sowie das on/offboarding (R25.a) oder die Geräteverwaltung (R25.b). Übliche Verwaltungsfunktionen werden in diesem Dokument nicht benannt, d.h., menschliche Interaktionen (Freigaben), Verwaltungsvereinfachungen wie Gerätetemplates oder Importfunktionen für „Asset Data provider“ [36] sind wünschenswert aber auch durch existierende Plattformen gut abgedeckt [33]. Zugriff auf unterliegende Plattformen über Konnektoren wird durch R13 impliziert. Statische Eigenschaften in R25.e sind z.B. das Betriebssystem und die Prozessorarchitektur (relevant für die Auswahl von Containern) oder die Anzahl von CPU/GPU/TPU-Kernen. Dynamische Eigenschaften in R25.f sind beispielsweise die Auslastung der Ressource, die Speicherbelegung oder die Auslastung von CPU/GPU/TPU-Kernen. Das Update der Plattform/Abstraktionsschicht wird in R136 beschrieben.

R26. Die Plattform **muss** ein Deployment (R24) auf on-premise-Ressourcen unterstützen. [Q, A]

R27. Die Plattform **soll** ein Deployment (R24) auf verbundene IIP-Ecosphere Plattformen unterstützen. [A]

Erläuterung: Dies erlaubt es Unternehmen, die sich gegenseitig vertrauen, teure Ressourcen wie GPU-Cluster gemeinsam zu verwenden. Entsprechende Sicherheitsmechanismen müssen vorhanden und konfiguriert sein. Ein Geschäftsmodell kann die Abgeltung regeln.

Dementsprechend muss die Geräteverwaltung (R25.c, R25.d) transparent gestaltet und mit entsprechenden Zugriffsmechanismen ausgestattet sein.

R28. Die Plattform **kann** ein (optionales) Deployment (R24) auf Cloud-basierte Ressourcen unterstützen. [E]

- a. Cloud-basierte Ressourcen **müssen** im Konfigurationsmodell (als Optionen) beschrieben werden. [Q, E, A]
- b. Die Plattform **kann** eine Cloud-Integration mit der Google Cloud ermöglichen. [D]
- c. Die Plattform **kann** eine Cloud-Integration mit Gaia-X ermöglichen. [Q]

Erläuterung: Bei einer Plattforminstantiierung mit abgeschalteten Cloud-basierten Ressourcen müssen sämtliche zugehörigen Abhängigkeiten bzw. Programmcode aus dem Plattformcode entfernt werden, um das Vertrauen der Anwender in die Plattform zu erhöhen. Je nach Cloud-Anbindung (privat, hybrid, öffentlich) kann hier unterschiedliche Performanz möglich sein, z.B., könnte bei einer privaten Cloud ein Datentransfer nahe an weicher Echtzeit möglich sein, während dies bei einer öffentlichen Cloud eher unwahrscheinlich ist. Prinzipiell gelten die in diesem Dokument genannten Performanz-Anforderungen im Millisekunden-Bereich (z.B., Abtastrate 0.2 ms, 8 ms Maschinentakt, 5 s Schritttakt, 25 s Prozesstakt), allerdings werden Abwägungen bei der Integration externer Dienste zu treffen sein. Ziel ist es, dass die Plattform mit der Verfügbarkeit von Laufzeiteigenschaften von Ressourcen und Komponenten (R25.h) alternative Ausführungs-Ressourcen abwägen und die geeignetste für die jeweilige Anwendung aussuchen und durchsetzen kann. Auch für Cloud-Integrationen soll die Ressourcen-Abstraktion einheitlich sein („ECS runtime“ in [36], R25.a).

R29. Eine Deployment Einheit **muss** eine explizite Schnittstelle in Form einer Verwaltungsschale anbieten.

- a. In der Verwaltungsschale **müssen** die (Qualitäts-)Eigenschaften der Deployment Einheit und ihre funktionalen Schnittstellen definiert sein. [Q]
- b. Die Verwaltungsschale der Ziel-Ressource (R25) **soll** in der Verwaltungsschale der Deployment Einheit verfügbar gemacht werden. [Q]
- c. Die Verwaltungsschale enthaltener Komponenten/Services **können** in der Verwaltungsschale der Deployment Einheit verfügbar gemacht werden. [Q]

R30. Eine Deployment Einheit **muss** als Container gekapselt werden. [Q]

- a. Die Container auf IT-Ebene **müssen** technologisch einheitlich sein. [Q]
- b. Die Container auf OT-Ebene **können** technologisch unterschiedlich sein. [Q]
- c. Die Plattform **kann** die Integration externer Container Registries ermöglichen. [U]

Erläuterung: Es kann erforderlich sein, dass je nach Ressource, z.B. für spezielle Edge-Geräte, Container für eine spezifische Container-Technologie verwendet werden müssen. Diese Information soll im Konfigurationsmodell hinterlegt sein, so dass eine Generierung der Container ermöglicht wird (siehe R32). Container werden üblicherweise aufgrund von Basis-Containern bzw. Container-Templates erstellt. Diese können intern vorliegen, aber auch von externen Quellen bezogen bzw. dorthin geladen werden (R30.c).

R31. Ein Container **soll** nur die erforderlichen Komponenten/Services enthalten. [A]

- a. Die benötigten Komponenten **müssen** im Konfigurationsmodell festgelegt werden. [Q]
Erläuterung: Dies können beispielsweise die Komponenten/Dienste für eine Applikation sein.
- b. Container **können** optionale Komponenten/Services enthalten. [A]

Erläuterung: Dies ermöglicht eine Adaption des Verhaltens der laufenden Applikation bzw. gesamten Plattform und erlaubt damit mehr als die benötigten Komponenten in R31.

- c. Container **können** Komponenten/Services dynamisch austauschen. [A]
Erläuterung: Dies ermöglicht eine Adaption des Verhaltens der laufenden Applikation bzw. der Plattform und umgeht R31.b.
- R32. Ein Container **kann** die zur Ausführung benötigten Daten oder Modelle enthalten. [T, D]
Erläuterung: KI-Modelle können ggf. nur auf entsprechenden IT-Ressourcen gelernt und vorbereitet werden, z.B., auf Ressourcen mit hinreichend vielen GPUs. In dem Fall soll der ausführende Container die Modelle enthalten oder über die Plattform Zugriff auf die Modelle erhalten und diese herunterladen und ggf. updaten können (siehe für Freigabe auch R118). In [36] beschreiben wir derartige Modelle als Parameter eines Dienstes, wobei entsprechende Update-Funktionalität für die Modelle angeboten werden kann. Zu diesen Modellen gehören ggf. auch PLC-Code bzw. Matlab/Simulink-Modelle aber insbesondere auch standardisierte KI-Modelle wie ONNX [23] (siehe R115). Formate wie ONNX können wiederum von einigen Geräten direkt ausgeführt werden, was entsprechende Berücksichtigung bei der Geräteabstraktion (R25) erfordert. Da möglicherweise nicht alle Verfahren Daten oder Modelle benötigen, ist diese Anforderung als kann-Anforderung beschrieben, für die jeweiligen Komponenten aber als muss-Anforderung zu betrachten.
- R33. Ein Container **kann** einen lokalen Datenspeicher enthalten. [Q]
Erläuterung: Es kann erforderlich sein, dass einzelne Komponenten eine lokale Datenbank benötigen, z.B., aus Performanzgründen. Daten können dann zusätzlich in zentrale Datenspeicher geschrieben werden. Anforderungen in Abschnitt 3.8 mit Ausnahme von R71 (Zugriff von allen Ressourcen) und R75 (Cloud-basierte Speicherdienste) gelten sinngemäß.
- R34. Die Erstellung von Containern durch die Plattform **soll** automatisiert, basierend auf den Einstellungen im Konfigurationsmodell, erfolgen. [A]
a. Dabei **kann** eine Modell-Validierung der Ausführbarkeit vor Erstellung bzw. vor Ausführung durchgeführt werden. [Q]
b. Die Plattform **kann** extern bereitgestellte Container unterstützen (z.B. für digitale Zwillinge). [E]
Erläuterung: Diese Anforderung soll R31 konsistent ermöglichen. R34.a beruht auf R8.b.
- R35. Die Edge-Geräte und die darauf installierten Container **müssen** das Auslesen von Produktionsdaten mit einer Abtastrate von ungefähr 2 ms ermöglichen. [T]
Erläuterung: Mindestens die Echtzeit-Anteile der OT-Edge-Geräte müssen diese Anforderung erfüllen und die Nicht-Echtzeit/IT-Software auf den Edge-Geräten darf diese Anforderung nicht stören. Sonst gilt für Plattformgeräte sinngemäß R10.
- R36. Die Plattform **soll** das Konfigurieren von Ressourcen ermöglichen. [U]
a. Die Plattform **soll** das Schreiben von Konfigurationseinstellungen aus der Ressourcenverwaltung auf Ressourcen ermöglichen. [U]
b. Die Plattform **soll** das Lesen von Konfigurationseinstellungen von Ressourcen in die Ressourcenverwaltung ermöglichen. [U]
Erläuterung: Lesen und Schreiben (spezifischer) Konfigurationseinstellungen erleichtert die Verwaltung. Die Kombination beider Operationen erlaubt ein Backup der Einstellungen. Beispiele für Einstellungen sind Netzwerkeinstellungen, Sicherheitseinstellungen, (technische) Einstellungen für Datenverbindungen, etc. Je nach Realisierung der Plattform, kann die Funktionalität auch durch in Code/Diensten/Applikationen eingebettete Konfigurationen realisiert werden, die durch modell-basierte Instantiierung (R8.c) effizient und konsistent in Code überführt werden.
- R37. Die Plattform **soll** die Fernwartung von Ressourcen ermöglichen. [U]
Erläuterung: Fernwartung ist eine Standard-Funktionalität im Gerätemanagement von IIoT-Plattformen [33].

3.6 Sicherheit

Sicherheitsmechanismen sowie gesicherte Verfahren zum Data Sharing sind Kernfunktionalitäten der IIP-Ecosphere Plattform.

- R38. Die Plattform **muss** die Nutzung ihrer Dienste nur für autorisierte Personen sicherstellen. [Q, P]
Erläuterung: Autorisierung kann über die in R40 genannten Mechanismen realisiert werden. Hierbei soll tiefer liegende Funktionalität wie z.B. die Sicherheitsmechanismen von Verwaltungsschalen integriert und verwendet werden.
- R39. Die Plattform **muss** sicherstellen, dass (personenbezogene) Daten nur von autorisierten Personen geändert werden. [Q, P]
- R40. Die Plattform **muss** die üblichen Sicherheitsmechanismen wie Role-Based Access Control (RBAC) und Transport Level Security (TLS) bereitstellen. [P]
- Rollen **können** im Konfigurationsmodell vorgegeben werden. [Q]
 - Zertifikate **können** im Konfigurationsmodell vorgegeben werden. [Q]
- R41. Die Sicherheitsmechanismen **sollen** mit üblichen Verzeichnisdiensten integrierbar sein. [P]
Erläuterung: Ein Beispiel ist das Lightweight-Directory-Access-Protocol (LDAP).
- Verzeichnisdienste **müssen** im Konfigurationsmodell konfiguriert werden. [Q]
 - Falls kein Verzeichnisdienst vorhanden ist, **muss** ein Mechanismus zur Verwaltung von Benutzeraccounts zur Verfügung gestellt werden. [P]
- R42. Weitere Sicherheitsmechanismen **müssen** einheitlich über das Konfigurationsmodell konfiguriert werden. [P]
- R43. Sicherheitsmechanismen **müssen** ihre (Qualitäts-)Eigenschaften und ihre aufrufbaren Funktionen als Verwaltungsschalen beschreiben. [Q]
- Die Verwaltungsschale eines Sicherheitsmechanismus **soll** den jeweiligen Einfluss auf die Performanz beschreiben. [A, Q]
Erläuterung: Damit kann die Plattform aufgrund der Konfiguration auf die (grobe) Einhaltung konfigurierter Performanzziele prüfen.
 - Ausgewählte Sicherheitsmechanismen **sollen** als Deployment Einheiten verteilbar sein. [D]
- R44. Das Konfigurationsmodell **soll** IDS-basierte Konnektoren als optional konfigurierbar anbieten. [E]
Erläuterung: Bei einer Plattforminstantiierung ohne IDS-Konnektoren müssen sämtliche IDS-Abhängigkeiten aus dem Plattformcode entfernt werden, um das Vertrauen in die Plattform zu erhöhen.

3.7 Datenschutz

Die Plattform soll aktuelle Vorgaben zum Datenschutz einhalten bzw. die Einhaltung unterstützen.

- R45. Die Plattform **muss** die faire und rechtmäßige Verarbeitung personenbezogener Daten bereitstellen. [Q, S]
Erläuterung: Diese Anforderung beruht auf dem Grundsatz der Transparenz.
- R46. In der Plattform **muss** die Erhebung von personenbezogenen für festgelegte, eindeutige und legitime Zwecke erfolgen. [Q, S]
Erläuterung: Dies kann durch die Dokumentation der Datenverarbeitungsschritte (R12) unterstützt werden.
- R47. Die Plattform **soll** die Verarbeitung von personenbezogenen Daten vermeiden. [Q, S]
- Auf der Plattform laufende Applikationen **sollen** die Verarbeitung von personenbezogenen Daten vermeiden. [Q]

Erläuterung: Diese Anforderung weist auf den Grundsatz der Datensparsamkeit hin. Während die Mechanismen der Plattform möglichst mit generischen Daten in einem einheitlichen Format arbeiten sollen (R19), unterliegen die Daten von KI-Applikationen die auf der Plattform laufen der Kontrolle und dem Bedürfnis des Benutzers. Daher ist diese Anforderung als Hinweis gedacht und ggf. durch eine Betriebsvereinbarung regelbar (siehe auch R66), falls diese Anforderung nicht durch Plattform-Mechanismen durchsetzbar ist.

- b. Auf der Plattform eingesetzte Modelle **sollen** gegen Angriffe auf die Privatsphäre geschützt werden. [T]

Erläuterung: Forschungsergebnisse [35] haben gezeigt, dass derartige Angriffe durchgeführt werden können und Anwendungen dagegen geschützt werden müssen.

- R48. Die Plattform **muss** personenbezogene Daten in einer Form speichern, die die Identifizierung der betroffenen Personen nur so lange ermöglicht, wie es für die Zwecke, für die sie verarbeitet werden, erforderlich ist. [Q, S, D]

Erläuterung: Diese Anforderung beruht auf dem Grundsatz der Speicherbegrenzung und korreliert mit R66.

- R49. Die Plattform **muss** personenbezogene Daten dem Zweck angemessen und erheblich sowie auf das für die Zwecke der Verarbeitung notwendige Maß beschränkt verarbeiten. [Q, S]

- a. Auf der Plattform laufende Applikationen **sollen** personenbezogene Daten dem Zweck angemessen und erheblich sowie auf das für die Zwecke der Verarbeitung notwendige Maß beschränkt verarbeiten. [Q, S]

Erläuterung: Diese Anforderung weist auf den Grundsatz der Datenminimierung hin. Während die Mechanismen der Plattform möglichst mit generischen Daten in einem einheitlichen Format arbeiten sollen (R19), unterliegen die Daten von KI-Applikationen die auf der Plattform laufen der Kontrolle und dem Bedürfnis des Benutzers. Daher ist diese Anforderung als Hinweis gedacht und ggf. durch eine Betriebsvereinbarung regelbar (siehe auch R66), falls diese Anforderung nicht durch Plattform-Mechanismen durchsetzbar ist.

- R50. Die Plattform **muss** verschiedene Kategorien der Verarbeitung personenbezogener Daten ausweisen. [Q, S]

Erläuterung: Dies kann im Zusammenhang mit der Dokumentation der Datenverarbeitungsschritte (R9) erfolgen.

- R51. Die Plattform **muss** die Berechtigung, Löschung oder Sperrung personenbezogener Daten gewährleisten. [Q, S]

- R52. Die Plattform **muss** personenbezogene Daten in einem strukturierten, gängigen und maschinenlesbaren Format speichern. [Q, S]

Erläuterung: Diese Anforderung beruht auf dem Recht auf Datenübertragbarkeit. Die Realisierung muss mit R19 abgestimmt sein.

- R53. Die Plattform **muss** einen Mechanismus für Benachrichtigungen bezüglich Berichtigung, Löschung, Blockierung, Leckage bereitstellen. [Q, S]

Erläuterung: Der ggf. austauschbare oder erweiterbare Mechanismus soll mit den Governance-Regeln des Betreibers der Plattform-Instanz in Einklang stehen. Falls die Plattform nur innerhalb eines Unternehmens verwendet wird, kann dies ggf. durch eine Betriebsvereinbarung geregelt werden (siehe auch R66). Es ist insbesondere sicherzustellen, dass Außenschnittstellen der Plattform z.B. beim Übergang von Daten oder beim Data Sharing zwischen Unternehmen diese Anforderungen berücksichtigen.

- R54. Die Plattform **muss** Möglichkeiten zum Widerspruch gegen die Verarbeitung personenbezogener Daten sicherstellen. [Q, S]

Erläuterung: Falls die Plattform nur innerhalb eines Betriebs verwendet wird, kann dies ggf. durch eine Betriebsvereinbarung geregelt werden (siehe auch R66). Es ist insbesondere sicherzustellen, dass Außenschnittstellen der Plattform z.B. beim Übergang von Daten oder beim Data Sharing zwischen Unternehmen diese Anforderungen berücksichtigen.

R55. Die Plattform **muss** Möglichkeiten zum Widerspruch gegen die Direktvermarktung personenbezogener Daten bereitstellen. [Q, S]

Erläuterung: Falls die Plattform nur innerhalb eines Unternehmens verwendet wird, kann dies ggf. durch eine Betriebsvereinbarung geregelt werden (siehe auch R66). Es ist insbesondere sicherzustellen, dass Außenschnittstellen der Plattform z.B. beim Übergang von Daten oder beim Data Sharing zwischen Unternehmen diese Anforderungen berücksichtigen.

R56. Die Plattform **muss** Möglichkeiten zum Widerspruch gegen die Weitergabe personenbezogener Daten an Dritte bereitstellen. [Q, S]

Erläuterung: Falls die Plattform nur innerhalb eines Unternehmens verwendet wird, kann dies ggf. durch eine Betriebsvereinbarung geregelt werden (siehe auch R66). Es ist insbesondere sicherzustellen, dass Außenschnittstellen der Plattform z.B. beim Übergang von Daten oder beim Data Sharing zwischen Unternehmen diese Anforderungen berücksichtigen.

R57. Die Plattform **muss** Möglichkeiten zum Einspruch der Entscheidungsunterstützung auf der Grundlage der automatisierten Verarbeitung personenbezogener Daten anbieten. [Q, S]

R58. Die Plattform **muss** die Aufdeckung von Verstößen gegen personenbezogene Daten und ihrer Kommunikation mit betroffenen Personen unterstützen. [Q, S]

R59. Die Plattform **muss** die Autorisierung hinsichtlich des Zugriffs auf personenbezogene Daten sicherstellen. [Q, P]

Erläuterung: Autorisierung kann über die in R40 genannten Mechanismen realisiert werden.

R60. Die Plattform **muss** Datenschutz Grundsätze gewährleisten, wenn für die Verarbeitung personenbezogener Daten eine Zustimmung erforderlich ist. [Q]

R61. Die Plattform **soll** den Nutzern die Möglichkeit geben, ihre Anforderungen an personenbezogene Daten zu kontrollieren. [Q]

R62. Die Plattform **soll** die Bewertung der Datenschutzfolgeabschätzung erleichtern, um Bedrohungen und Risiken bei der Verarbeitung personenbezogener Daten zu identifizieren. [Q]

R63. Die Plattform **soll** einen Mechanismus zur Erfassung der Datenschutz- und Sicherheitsanforderungen der Benutzer bereitstellen. [Q]

R64. Die Datenschutzmechanismen **sollen** die Informationen in vorgegebenen Datenfeldern anonymisieren. [D]

a. Die Vorgabe der Datenfelder **soll** über das Konfigurationsmodell erfolgen. [Q]

Erläuterung: In einer Applikation könnten beispielsweise für Werker anonyme IDs verwendet werden. Weiterhin könnten Maschinendaten mit dem Zeitstempel und der anonymen Werker-ID anonymisiert werden.

R65. Die Datenschutzmechanismen **sollen** in Freitext enthaltene personenbezogene Daten erkennen und anonymisieren. [D]

a. Die Vorgabe der Datenfelder **kann** über das Konfigurationsmodell erfolgen. [Q]

Erläuterung: Alle Freitextfelder zu anonymisieren kann zeitkritisch werden. Viele Freitextfelder zur Anonymisierung freizugeben kann das Konfigurationsmodell überladen. Ein entsprechender Kompromiss ist zu finden und umzusetzen.

R66. Die Pseudonymisierung **soll** Pseudonyme nur für die tatsächlich benötigte Zeit im System halten. [D]

- a. Nach Ablauf der Zeit **soll** der gleichen Person ein neues Pseudonym zugeordnet werden. [D]
 - b. Zweck und Umfang **sollen** bei interner Verwendung mit der Betriebsvereinbarung abgeglichen werden. [D]
 - c. Pseudonymisierung **soll** bei externer Verwendung erst nach Zustimmung der betreffenden Person angewendet werden. [D]
- R67. Die Plattform **soll** generierten Cookies oder ähnlichen Identifikationsmerkmalen, die auf Endgeräten gespeichert werden, erfassen und klassifizieren. [S]
- R68. Die Plattform **soll** die Möglichkeit zur automatischen Löschung solcher (R61) Identifikationsmerkmalen, sowie zur Löschung auf Wunsch des Nutzers geben. [S]

3.8 Zentrale Speicherdienste

Eine für Applikationen und Demonstratoren wichtige Plattformfunktionalität ist die Möglichkeit, beliebige Daten abzulegen. Aktuelle Industrie 4.0 Plattformen [33] bieten verschiedene Formen von „Data Lakes“ an, die teilweise auf Cloud-Technologie basieren. Einige Edge-Device-Hersteller bieten ebenfalls ähnliche Funktionalität an. Allerdings ist die Verwendbarkeit, Flexibilität und Datensicherheit über Herstellergrenzen bislang relativ gering. Das gilt auch für die deklarative und flexible Auswahl von Speicherdiensten abhängig von Qualitätskriterien wie Datenvolumen oder Datenfrequenz. Neben den zentralen Datenspeicherdiensten können Deployment-Einheiten lokale Datenspeicherdienste, z.B. als Puffer, enthalten (R33).

- R69. Die Plattform **muss** die Einbindung alternativer Speicherdienste mit unterschiedlichen Qualitätseigenschaften anbieten. [D]
Erläuterung: Denkbare Qualitätseigenschaften sind z.B., Geschwindigkeit, Kapazität, Sicherheit.
- R70. Ein Speicherdienst **muss** seine (Qualitäts-)Eigenschaften und Funktionen als Verwaltungsschale beschreiben. [Q]
 a. Die Verwaltungsschale **kann** Informationen für direkten Zugriff auf den Speichermechanismus enthalten. [X]
Erläuterung: Direkte Kommunikations-Endpunkte können erforderlich sein, um bestmögliche Performanz zu erreichen. Wünschenswert wäre hierbei nur wenige Protokolle für den Zugriff zu verwenden. Aktuell finden Standardisierungsarbeiten in diese Richtung in der Plattform Industrie 4.0 [25] statt.
- R71. Die Speicherdienste **sollen** allen Ressourcen zur Verfügung stehen. [D]
 a. Der konkrete Speicherdienst **soll** aufgrund vorgegebener Eigenschaften transparent von der Plattform ausgewählt werden. [D]
- R72. Die Plattform **muss** mindestens einen konkreten on-premise Speicherdienst realisieren. [D]
 a. Die Plattform **kann** Anbindungen an Datenbanken unterstützen. [D]
Erläuterung: Hier ist mindestens eine Anbindung an Oracle DB wünschenswert.
- R73. Die Speicherdienste **sollen** die folgenden Arten von Daten unterstützen [T].
 a. Die Speicherdienste **sollen** strukturierte Daten speichern können. [Q]
 b. Die Speicherdienste **sollen** heterogene Zeitreihen speichern und mit potenten Komprimierungsfunktionen bearbeiten können. [T]
 c. Die Speicherdienste **sollen** unstrukturierte Daten wie Logs speichern können. [T, D]
 d. Die Speicherdienste **sollen** gelabelte Daten speichern können. [U]
 e. Das Datenschema für Speicherdienste strukturierter Daten **soll** im Konfigurationsmodell definiert werden. [Q, D]
 f. Das Datenschema **muss** mit den Daten gespeichert werden. [T]
Erläuterung: Änderungen im Konfigurationsmodell führen zu Änderungen des Datenschemas der Speicherdienste. Hinzufügen neuer Informationen sollte möglich sein.

Nicht-kompatible Änderungen können abgelehnt werden. Das Datenschema ist für die Meta-Informationen im Data Sharing relevant (R79) und muss daher zugreifbar sein.

- R74. Speicherdienste **sollen** Schreib-Zugriff in weicher Echtzeit ermöglichen. [D]
- Echtzeit-Speicherdienste **sollen** drei Fließkommawerte bei 0.2ms Abtastrate abspeichern können. [D]
 - Echtzeit-Speicherdienste **sollen** 10 Fließkommawerte bei 8ms Abtastrate abspeichern können. [D]
 - Echtzeit-Speicherdienste **soll** die Integration der Daten aus Teilanforderungen a und b gemeinsam speichern können. [D]
 - Echtzeit-Speicherdienste **sollen** mit 2GByte Daten pro Stunde abspeichern können. [D]
- Erläuterung:* R74.c wurde aufgenommen, damit (lokale) Speicherdienste (R33) diese Anforderung ggf. auch ohne vorhandene Datenintegration berücksichtigen müssen. R74.a-R74.c können ggf. durch lokale Speicherdienste erfüllt werden. R74.d bedingt noch höhere Durchsatzraten, die ggf. verteilte Speicherung erfordern und nicht mehr durch lokale Speicherdienste realisiert werden können.
- R75. Speicherdienste **sollen** Lese-Zugriff in weicher Echtzeit ermöglichen. [T]
- R76. Die Plattform **muss** Funktionalität zur Abfrage verfügbarer Daten bereitstellen. [T]
- Die Plattform **kann** automatisierte Mechanismen zum Löschen von Daten bereitstellen. [T]
 - Die Plattform **soll** den Zugriff auf Teilmengen zugreifbarer Daten ermöglichen. [T]
 - Die Plattform **soll** die Weiterverarbeitung zugreifbarer Daten ermöglichen. [T]
- Erläuterung:* Zur Weiterverarbeitung können ganze Datensätze aber auch Teile bereitgestellt werden, wobei hier nicht der Mechanismus/die Spezifikation zur Auswahl der Datensätze festgelegt werden soll. Bereitgestellte Datensätze können beispielsweise als Stream oder Container verfügbar gemacht werden. Die Speicherdienste können für Datenexploration [36] unter entsprechenden Zugriffsrechten verfügbar gemacht werden, ggf. nach vorheriger Kopie von Teildatensätzen.
- R77. Die Plattform **kann** optionale Cloud-basierte Speicherdienste anbieten. [E, D]
- Das Konfigurationsmodell **muss** die Verwendung von Cloud-basierte Speicherdiensten (R72) als Option anbieten. [E, A]
- Erläuterung:* Bei einer Plattforminstantiierung ohne Cloud-basierte Speicherdienste müssen sämtliche Cloud-Abhängigkeiten und -Funktionalitäten aus dem Plattformcode entfernt werden, um das Vertrauen in die Plattform zu stärken.
- Die Plattform **kann** als Cloud-Speicherdienst AWS anbieten. [D]
- Explanation:* AWS Frankfurt wurde explizit von einem Demonstrator genannt.
- Die Plattform **kann** als Cloud-Speicherdienst Google anbieten. [D]
- R78. Die Plattform **muss** einen Speicher für Dienste anbieten. [U]
- Die Plattform **muss** einen Speicher für einzelne Dienste nach Version anbieten. [U]
 - Die Plattform **muss** einen Speicher für Deployment-Einheiten anbieten. [U]
- Erläuterung:* Diese Anforderung bezieht sich auf den „service store“ in [36]. Der Binärcode der Dienste muss geeignet und versioniert verfügbar gemacht werden (R78.a). Zu Deployment-Einheiten zusammengefügte Dienste und Applikationsmechanismen müssen den Ressourcen zum Download/zur Verwendung bereitgestellt werden (R78.b).

3.9 Data Sharing

Flexibles und konfigurierbares Data Sharing zwischen verbundenen Unternehmen ist eine innovative und neuartige Funktionalität der IIP-Ecosphere Plattform. Data Sharing erlaubt die Freigabe eigener Daten und die Nutzung fremder Daten über Außen-Schnittstellen der Plattform.

- R79. Daten **müssen** Meta-Informationen haben. [T]
- Die Meta-Informationen **müssen** eine eindeutige Identifikation beinhalten. [T]
 - Die Meta-Informationen **müssen** das Datenformat beschreiben. [T]
 - Die Meta-Informationen **sollen** die Größe der Daten beschreiben. [T]
 - Die Meta-Informationen **müssen** einen Zeitstempel der Entstehung/Veränderung enthalten. [T]
 - Meta-Informationen **müssen** die Zugriffs-/Nutzungsrechte beinhalten. [T]
 - Meta-Informationen **sollen** die Herkunft/den Datenpfad beschreiben. [T]
- Erläuterung:* Diese Anforderung verfeinert R19.g. Die Meta-Informationen sollen sowohl die genaue Herkunft, Vorverarbeitung durch Komponenten/Services, die Struktur (Namen, Typen, „Semantik“ der Felder) als auch die Zugriffs- und Nutzungsrechte (z.B. durch Smart Contracts) beschreiben. Die Meta-Informationen können auch weitere Sicherheitsinformationen wie Hashcodes enthalten.
- R80. Daten **sollen** im Konfigurationsmodell beschrieben werden. [Q]
- Erläuterung:* Da alle für Anwendungen benötigten Datenformate im Konfigurationsmodell beschrieben sind, lassen sich für diese „Schutzklassen“ festlegen. Die Schutzklassen bestimmen die bereitzustellenden (IDS-)Konnektoren, die Anpassung der Datenformate bzw. die notwendige Anonymisierung oder Pseudonymisierung (siehe auch Abschnitte 3.4, 3.6 und 3.7). Weitere Mechanismen sind denkbar. Aus der Konfigurationsmodell-Beschreibung der Daten und deren Verarbeitung lassen sich direkt zentrale Meta-Informationen ableiten (R79).
- R81. Für den Datenaustausch über die Grenzen der Plattform-Instanz hinaus **sollen** sichere Container verwendet werden. [A, T]
- Erläuterung:* Beim Verarbeiten größerer Datenmengen z.B. eines KI-Modells oder zu lernender Daten auf fremden Ressourcen (siehe R27, R28 und R71) kann es erforderlich sein, sowohl die Daten möglichst weitreichend vor dem Zugriff des Ressourcenbesitzers zu schützen. Dennoch sollen die Daten auf den fremden Ressourcen geeignet verarbeitet werden können.
- R82. Die Plattform **muss** Funktionalität zur Abfrage verfügbarer Daten bereitstellen. [T]
- Die Plattform **muss** die für einen authentifizierten Benutzer zugreifbaren Daten anhand der Zugriffs-/Nutzungsrechte bereitstellen. [T]
 - Die Plattform **muss** das Löschen zugreifbarer Daten anhand der Zugriffs-/Nutzungsrechte ermöglichen. [T]
- Erläuterung:* Diese Anforderung erweitert R76 im Sinne des manuellen Zugriffs/Data Sharing. Ein Benutzer soll die Möglichkeit haben, die verfügbaren bzw. zum Data Sharing bereitstehenden Datensätze zu identifizieren und (falls erforderlich) zu löschen.

3.10 Datenintegration

Die Datenintegration in der IIP-Ecosphere Plattform soll KI-Algorithmen und Applikationen zusätzliche Perspektiven auf die Produktion jenseits reiner Produktionsdaten bieten. Dafür sollen in der (strombasierten) Datenintegration Modelle wie Konstruktionsdaten mit Produktionsdaten angereichert werden.

- R83. Die Datenintegration **soll** als Plattform-Container bereitgestellt werden. [Q]
- Erläuterung:* Auch wenn die Datenintegration intern verteilt ist und wahrscheinlich fest installiert wird, soll dies in Container-basierter Weise erfolgen. Eine Installation kann durchaus aus mehreren Containern bestehen, solange ein Container die Verwaltungsschale bereitstellt.
- R84. Die Datenintegration **soll** auf weitere Datenquellen wie internal data stores. [Q, A]

Erläuterung: Hierfür sind ggf. erforderliche Sicherheitsmechanismen (siehe Abschnitt 3.6) zu definieren, bereitzustellen und zu konfigurieren.

- R85. Die Datenintegration **muss** ihre (Qualitäts-)Eigenschaften und Funktionen als Verwaltungsschale beschreiben. [Q, D]

Erläuterung: Dies gilt sowohl für den Zugriff „von oben“ als auch „von“ den KI-Algorithmen aus und ebenso für den Zugriff von Edge-Devices aus.

- R86. Die Funktionalität der Datenintegration **soll** durch das Konfigurationsmodell festgelegt werden. [Q]

Erläuterung: Damit kann geprüft werden ob Dienste bzw. Applikationen die notwendigen Daten erhalten können.

- R87. Die Datenintegration **muss** sowohl streaming-basierte OT- als auch semantische IT-Daten integrieren können. [Q]

- R88. Die Datenintegration **muss** programmiersprachen- und deployment-unabhängig Applikationen und KI-Modellen Zugriff auf integrierte Daten ermöglichen. [Q]

- R89. Die Plattform **muss** der Datenintegration den schreibenden Zugriff auf Daten ermöglichen. [Q]

Erläuterung: Durch die Datenintegration geänderte und erzeugte Daten sollen in geeigneten Datenspeichern (siehe R33, Abschnitt 3.8) abgelegt werden können. Die Datenspeicher sind im Konfigurationsmodell festzulegen (R86). Die Datenintegration muss ggf. Daten-Metainformationen ergänzen oder anpassen (R19.g).

- R90. Die Datenintegration **muss** ermöglichen, den Zugriff auf Daten zu schützen. [Q]

Erläuterung: Diese Anforderung soll nochmals auf die Notwendigkeit des Datenschutzes (Abschnitt 3.7) in diesem Bereich hinweisen.

- R91. Die Datenintegration **muss** die Anforderung der weichen Echtzeit erfüllen. [Q]

Erläuterung: Diese Anforderung überträgt die in R10 genannte weiche Echtzeitanforderung für produktionsnahe Plattformoperationen auf die Datenintegration. Demonstrator-Beispiel für Integration unterschiedlicher Daten in eine Datentabelle, z.B., Werte im Maschinentakt alle 8 ms und externe Sensoren alle 0.2 ms bzw. OT Abtastrate von 2 ms (R35). Eine Anwendung der Datenintegration kann die Aggregation gemessener Maschinen-Daten sein, um beispielsweise Eingangsdaten von 7 GByte pro Stunde auf relevante, speicherbare Daten von 2 GByte pro Stunde zu reduzieren.

- R92. Die Datenintegration **muss** ermöglichen, dass historische Daten abrufbar sind. [Q]

3.11 Konfigurierbarkeit

Eine systematische Konfigurierbarkeit der Plattform erleichtert die Installation, die Wartung und die Definition von Applikationen. Optionale Komponenten wie beispielsweise Cloud-Integrationen oder IDS-Funktionalität können durch geeignete Instantiierungsmaßnahmen komplett aus einer Plattform-Instanz entfernt werden, was das Vertrauen in die Plattform erhöhen soll. Dadurch könnten Plattform-Instanzen entstehen, denen erforderliche Funktionalität für die Applikationen fehlt. Daher ist es wichtig, Konfigurationen auf deren Validität zu prüfen. Konfigurationsinformationen können weiterhin zu verschiedenen (späteren) Zeitpunkten verwendet werden, beispielsweise können Festlegungen, die vor der Laufzeit getroffen werden, den Entscheidungsspielraum für die Adaption sinnvoll einschränken.

Konfigurationseigenschaften betreffen üblicherweise mehrere oder sogar alle Funktionsbereiche. Daher sind Anforderungen an das Konfigurationsmodell teilweise in anderen Abschnitt genannt, ohne dass die Anforderungen hier wiederholt werden.

Die folgenden Anforderungen erzwingen keine vollständige Konfigurierbarkeit, da sich der Think Tank Plattformen auf die Erforschung von Plattformkonzepten für Industrie 4.0 konzentriert und nicht auf die lückenlose und vollständige Umsetzung im Sinne eines Produkts.

- R93. Die Plattform **muss** systematisch in Form eines Konfigurationsmodells konfigurierbar sein. [A]
Erläuterung: Diese Anforderung nimmt R8.a auf.
- R94. Die Plattform **muss** das Konfigurationsmodell automatisch auf Inkonsistenzen und Fehler prüfen können. [A, Q]
a. Die Prüfung eines Konfigurationsmodells mit 50 Ressourcen und 5 Applikationen **soll** in weniger als 1 Sekunde durchführbar sein. [Q]
Erläuterung: Diese Anforderung konkretisiert R8.b.
- R95. Die Plattform **muss** aus einer Konfiguration automatisch instantiierbar sein. [A]
a. Die Instantiierung einer Plattform-Instanz mit 50 Ressourcen und 5 Applikationen **soll** innerhalb von 15 Minuten erfolgen. [Q]
- R96. Das Konfigurationsmodell **muss** optionale und alternative Plattformkomponenten/Dienste bzw. abbilden. [A]
a. Das Konfigurationsmodell **muss** Eigenschaften der Plattformkomponenten/Dienste beschreiben.
Erläuterung: Beispiele sind Lizenz einschränkungen (R6) basierend auf einem stark vereinfachten Lizenzmodell, Sicherheitskomponenten (R40, R41, R44), Cloud-Einbindung (R74), Performanzaspekte etc. Einige dieser Eigenschaften sollen geeignet geschützt werden (z.B. read-only nach erster Festlegung).
- R97. Das Konfigurationsmodell **muss** die Applikationen, die auf der Plattform laufen, umfassen. [Q]
a. Eine Applikationskonfiguration **muss** die konfigurierten Dienste der Applikation umfassen. [Q, U]
b. Eine Applikationskonfiguration **muss** die konfigurierten Konnektoren der Applikation umfassen. [Q, U]
c. Eine Applikationskonfiguration **muss** die Datenpfade der Applikation enthalten. [Q, U]
d. Eine Applikationskonfiguration **soll** alternative Dienste enthalten. [Q, U]
e. Das Konfigurationsmodell **kann** Applikations-Templates unterstützen. [U]
- R98. Das Konfigurationsmodell **soll** Anpassungen zu verschiedenen Zeiten im Software-Lebenszyklus unterstützen. [A, Q, U]
Erläuterung: Das Konfigurationsmodell soll sowohl zur Anpassung der IIP-Ecosphere Plattform vor der Laufzeit, zur Steuerung von Optimierungen zur Startzeit als auch zur Steuerung von Anpassungen zur Laufzeit verwendet werden.
- R99. Informationen aus dem Konfigurationsmodell **können** durch interne Konnektoren anderen Komponenten zur Verfügung gestellt werden. [Q]
Erläuterung: Damit können Eigenschaften aus dem Konfigurationsmodell wiederum in seiner Verwaltungsschale dargestellt werden.
- R100. Das Konfigurationsmodell **kann** dezentral modelliert werden. [Q]
Erläuterung: Ein zentrales Modell kann einer lokalen Optimierung im Wege stehen. Allerdings sind hierzu Vorarbeiten aus dem BMBF-geförderten DevOpt Projekt [4] erforderlich.
- R101. Informationen, die in den Verwaltungsschalen der Komponenten/Dienste bereitgestellt werden, **sollen** automatisiert in das Konfigurationsmodell eingeblendet werden. [Q]
a. Die Übernahme der Informationen für ein Konfigurationsmodell mit 50 Ressourcen und 5 Applikationen **soll** in weniger als 1 Sekunde durchführbar sein. [Q]

Erläuterung: Damit keine Inkonsistenzen zwischen dem Konfigurationsmodell und den Komponenten bzw. deren (Laufzeit-)Eigenschaften entstehen, sollen relevante Informationen aus den Verwaltungsschalen automatisiert übernommen werden.

3.12 Optimierte bzw. adaptives Deployment

Die automatische Optimierung des Deployments durch die Plattform erleichtert Administratoren die Installation und die Bereitstellung der Plattform. Allerdings können sich relevante Eigenschaften während der Laufzeit ändern. Daher wird ein adaptives Deployment angestrebt, d.h., die erneute Optimierung und Anpassung des Deployments zur Laufzeit. Die Optimierung kann R9 (Availability) unterstützen bzw. realisieren.

R102. Die Plattform **muss** Deployment Einheiten zu vorhandenen Ressourcen zuteilen. [A, U]

- a. Die Plattform **muss** die Validität der Ressourcenzuteilung überprüfen. [U]
- b. Die Plattform **soll** Ressourcen zur Datenexploration bereitstellen. [U]

Erläuterung: Dabei handelt es sich zunächst um eine (nicht notwendigerweise optimale) Zuteilung von Containern zu Ressourcen. Die Zuteilung ist auf Validität zu prüfen und der Benutzer ist ggf. zu informieren. Eine spezielle Form der Zuteilung ist die Bereitstellung angeforderter Ressourcen zur Datenexploration (R104.b) im Rahmen einer Data Science Toolchain [36]. Da hierbei ggf. zentrale Ressourcen, z.B. GPUs benötigt werden, die einem händisch zusammengestellten Container bereitgestellt werden sollen, der in Konkurrenz zu genutzten Ressourcen der Applikationen, z.B. zum erneuten Training von KI-Modellen stehen, liegt diese Form der Zuteilung in der Verantwortung der Plattform, d.h., dem adaptiven Deployment. Die Ressourcenzuteilung kann sich über Konnektoren auf unterliegende Plattformen bzw. deren Ressourcen auswirken (R13).

R103. Die Plattform **soll** die Zuteilung von Deployment Einheiten zu Ressourcen optimieren. [A]

- a. Die Optimierung **soll** die Ressourcennutzung (CPU, Speicher, etc.) berücksichtigen. [Q]
- b. Die Optimierung **soll** Netzwerkeigenschaften berücksichtigen. [Q]
- c. Die Optimierung **soll** Komponenteneigenschaften berücksichtigen. [U]

Erläuterung: Komponenteneigenschaften (c, R96.a) können die Verteilung von Komponenten beeinflussen, ggf. Ressourcen ausschließen, z.B., Ressourcen außerhalb einer Fabrik wegen rechtlicher Gründe oder Lizenzgründe.

R104. Die Optimierung der Ressourcenzuteilung **muss** das Konfigurationsmodell als Grundlage zur Optimierung einsetzen. [Q, A, U]

- a. Die Optimierung **muss** die Ressourcenzuteilung im Konfigurationsmodell abbilden. [Q, A]
- b. Die Optimierung **muss** die Validität des Konfigurationsmodells sicherstellen. [Q, A, U]

Erläuterung: Das Konfigurationsmodell beinhaltet die Information über die Applikationen, die benötigten Komponenten, die Ressourcen sowie deren Laufzeitinformationen. R104.b beruht auf R94 und R8.b.

R105. Die Durchsetzung der Ressourcenzuteilung **muss** über Plattformfunktionen und Verwaltungsschalen erfolgen. [Q]

Erläuterung: Hierzu sind R25 und R32 zu berücksichtigen.

R106. Die Optimierung **muss** "statisch" zum Start der Plattform/Applikation erfolgen. [A]

R107. Die Optimierung **soll** dynamisch zur Laufzeit erfolgen. [Q, A]

- a. Die Optimierung **muss** die Validität des Konfigurationsmodells regelmäßig überprüfen. [Q]
- b. Die Ressourcen-Optimierung einer Plattform-Instanz mit 50 Ressourcen und 5 Applikationen **soll** innerhalb von 15 Minuten erfolgen. [Q]

Erläuterung: Damit wird Adaption der laufenden Applikationen bzw. der Plattform ermöglicht, z.B., um die Zuteilung anzupassen, wenn sich verfügbare Ressourcen ändern,

die Konfiguration geändert wird, oder über/unernutzte Ressourcen erkannt werden, etc. Die Durchsetzung erfolgt dann zur Laufzeit über R105.

- R108. Die Optimierung **muss** die Ausführbarkeit der Applikationen berücksichtigen. [Q]
Erläuterung: Die Optimierung darf die Ausführung von Applikationen mit verteilten Komponenten nicht stören, d.h., die (dynamische) Optimierung darf keine negativen Auswirkungen auf die Ausführung von Produktionsprozessen haben bzw. die erforderlichen Anpassungsmechanismen müssen entsprechend ausgelegt sein.
- R109. Die Optimierung **soll** Transparenz und Nachvollziehbarkeit über die Entscheidungsfindung bieten. [D]

3.13 KI-(Service-)Baukasten

Der offene und erweiterbare KI-Baukasten ist ein weiterer innovativer Baustein der IIP-Ecosphere Plattform, insbesondere da nur wenige Industrie 4.0 Plattformen bislang die Notwendigkeit für benutzer-spezifische KI-Komponenten erkannt bzw. umgesetzt haben [33]. Die Elemente im KI-Baukasten werden Komponenten/Services sein, die über Verwaltungsschalen ihre Schnittstellen und Eigenschaften veröffentlichen und (falls möglich und sinnvoll) im Sinne des Deployments (Abschnitt 3.5) verteilbar sind, z.B. Trainings-Dienste auf GPU-Ressourcen und zugehörige Prädiktions-Dienste auf Edge-Geräte. Die Plattform wird hierfür als Grundlagen Schnittstellen für verteilbare Dienste (Abschnitt 3.5), Datenpfade (R20), Datenspeicher (Abschnitt 3.8), Data-Sharing (Abschnitt 3.9) und Datenschutz (Abschnitt 3.7) bereitstellen. Die Verknüpfung der Dienste erfolgt über Datenpfade (R20) im Rahmen der Anwendungs-Konfiguration (Abschnitt 3.16).

Der Spezifikation der tatsächlichen Schnittstellen für KI-Dienste sowie das eigentliche Design des KI-Baukastens basierend auf aktuellem KI-Verständnis und KI-Wissen erfordert eine enge und konstruktive Zusammenarbeit zwischen verschiedenen Partnern. Daher werden im Folgenden zwar Rahmen-Anforderungen an den KI-Baukasten genannt, z.B., auch verschiedenste KI-Verfahren (R114) oder KI-Frameworks (R113), die die Stakeholder im Rahmen der Anforderungsermittlung genannt haben und in die Plattform integriert sehen möchten. Für eine Umsetzung des KI-Baukastens müssen die KI-Verfahren geeignet als Dienste (basierend auf Plattform-Schnittstellen und Diensten) abstrahiert und mit KI-spezifischen Schnittstellen ergänzt werden. Dabei müssen die zu integrierenden KI-Verfahren für die Umsetzung und Integration priorisiert werden, da es ggf. nicht möglich ist, im Rahmen der Projektzeit alle gewünschten Verfahren zu integrieren (soll-Anforderungen). Hier ist es für IIP-Ecosphere wichtiger, den Weg zu zeigen und auf Standardisierung/Normung der Schnittstellen hinzuarbeiten, als möglichst viele KI-Verfahren zu integrieren. Dies (und die zugehörigen Entscheidungen) kann nicht ausschließlich durch den Think Tank „Plattformen“ erfolgen und liegt daher im Verantwortungsbereich mehrere Projektteile, z.B., insbesondere Think Tank „KI und Produktion“, KI-Accelerator oder auch individuelle Demonstratoren. Der Think Tank „Plattformen“ wird zu diesem Entwurf (wie oben beschrieben) beitragen und die Ergebnisse geeignet in die Plattform übernehmen, so dass die KI-Schnittstellen integraler Bestandteil der Plattform werden und dass die dazu notwendigen, auf bestehenden Frameworks basierenden Implementierungen der Plattform (und damit den darauf laufenden Applikationen) zur Konfiguration, zum Deployment und zur Ausführung bereitstehen.

Hierbei ist insbesondere auf R9 hinzuweisen: KI-Dienste erweitern die Plattformfunktionalität, d.h., sofern KI-Dienste mit der Plattform ausgeliefert werden, liegt die Verantwortung für die Qualität der Dienste (insbesondere Terminierung, Speichernutzung, Datentyp-Konformität und Antwortzeiten) zwar zunächst beim Ersteller aber dann implizit bei der IIP-Ecosphere Plattform. Bei externen Diensten liegt die Einhaltung der erforderlichen Qualität nicht bei der Plattform, d.h., wie für R9 beschrieben hat die Plattform ggf. nur eingeschränkte Mittel zur Verfügung, um die Einhaltung der Qualität zu überwachen bzw. durchzusetzen. Dies gilt insbesondere für die Komposition von Diensten zu

Anwendungen (Abschnitt 3.16), bei denen die Verantwortung für die Gesamtqualität entsprechend beim Benutzer liegt.

- R110. Der KI-Baukasten **muss** Schnittstellen für relevante KI-Komponenten in der industriellen Produktion definieren. [A, Q]
- Die Schnittstellen des KI-Baukastens/der Dienste **müssen** als Verwaltungsschalen realisiert werden. [Q]
 - Die Verwaltungsschalen der KI-Komponenten **müssen** sowohl funktionale als auch Qualitäts-Eigenschaften beschreiben. [Q]
- Erläuterung:* Dabei kann es sich um einheitliche/generische bzw. heterogene Schnittstellen handeln. Wünschenswert ist, dass nur absolut notwendige Schnittstellen definiert werden (siehe auch Format-Sparsamkeit in R19).
- R111. Der KI-Baukasten **muss** erweiterbar sein, so dass Benutzer KI-Komponenten der Plattform bzw. der Plattforminstanz hinzufügen können. [A, Q]
- Erläuterung:* Hierfür müssen die KI-Komponenten R110 erfüllen.
- R112. Die Komponenten/Services im KI-Baukasten **sollen** in Deployment Einheiten auf verfügbare Ressourcen verteilbar sein. [A]
- Parameter der Komponenten **sollen** im Konfigurationsmodell beschreiben sein. [U]
 - Eigenschaften der Verteilbarkeit **sollen** im Konfigurationsmodell beschrieben sein. [A, Q]
 - Die Verteilbarkeit **soll** für individuelle Komponenten einschränkbar sein. [D]
- Erläuterung:* Die (dynamische) Verteilung der Komponenten basierend auf deren Angebot und Ressourcenverbrauch (R24) sowie deren Einsatz innerhalb von Applikationen (R131) ist eine Kernfunktionalität der Plattform. Hierfür kann es erforderlich sein, die Komponenten je nach Funktion mit verschiedenen Schnittstellen auszustatten, z.B., eine für Training und eine für Prädiktion. Allerdings kann es sein, dass Komponenten bewusst zentral ausgeführt werden sollen, was entweder durch Eigenschaften (R96.a), eine Einschränkung der Ziel-Ressourcen, einen entsprechenden Ressourcenbedarf oder – falls erforderlich – durch eine entsprechende Einstellungen im Konfigurationsmodell (c, R96.a) bzw. den unterliegenden Verwaltungsschalen (R101) repräsentiert werden kann.
- R113. Der KI-Baukasten **soll** KI-Komponenten in üblichen Programmiersprachen und Programmierumgebungen unterstützen. [D]
- Der KI-Baukasten **soll** Python, Pandas/Numpy [24], Knime [19], Scikit-learn, Tensorflow und Rapidminer [30] unterstützen. [T, D]
- Erläuterung:* Im KI-Umfeld werden diverse Programmiersprachen (z.B. Python) und Frameworks (z.B. Tensorflow) verwendet. Diese sind auf systematische Weise auch im Deployment zu unterstützen, was ggf. die Spezifikation geeigneter Abhängigkeiten (im Konfigurationsmodell) erfordert. Siehe auch R6 und R96. Da die Firma Rapidminer Projektpartner ist, soll der Integration des Rapidminer-Ansatzes im Vergleich zum Knime-Ansatz (beide wurden im gleichen Demonstrator-Kontext genannt) Priorität eingeräumt werden. R113.a benennt hier einige Beispiele, ist aber nicht vollständig zu sehen.
- Der KI-Baukasten **soll** eine Middleware zum Agenten-Management basierend auf der Industrie 4.0 Sprache [27, 38] bereitstellen. [T]
- Erläuterung:* Diese Middleware sollte ein standardisiertes Erstellen und Löschen von Softwareagenten übernehmen können, sowie eine standardisierte Kommunikation (erweitertes Contract Net Protocol [27]) nutzen (ähnlich zu JADE- Java Agent Development Framework [16]). Das könnte dann auch FIPA [10] subsumieren.
- Erläuterung:* Je nach Verfahren kann es erforderlich sein, Berechnungen auf geeigneten IT-Ressourcen wie GPUs durchzuführen, die Lösungen dann zu exportieren und mit den

- Verfahren auf die jeweiligen ausführenden Ressourcen wie Edge-Geräte zu deployen (siehe R32).
- R114. Der KI-Baukasten **soll** relevante produktions-übliche bzw. produktions-geeignete KI-Verfahren bereitstellen. [D, T]
- KI-Verfahren im KI-Baukasten **sollen** generisch, parametrisierbar und anpassbar sein. [D]
 - Der KI-Baukasten **soll** Transfer Learning unterstützen. [D]
 - Der KI-Baukasten **soll** Reinforcement Learning unterstützen [T, D].
Erläuterung: Hier soll insbesondere Deep Q Learner wie in Tensorflow unterstützt werden.
 - Der KI-Baukasten **soll** einfache statistische Verfahren zur Entscheidungsfindung unterstützen. [T]
 - Der KI-Baukasten **soll** Neuronale Netze unterstützen. [T]
 - Der KI-Baukasten **soll** Zeitreihenklassifikation unterstützen. [D]
Erläuterung: Dies ergänzt R114.e. Es sollen insbesondere Residual Neural Network und Feed Forward Neural Network wie in Tensor Flow unterstützt werden.
 - Der KI-Baukasten **soll** Anomalieerkennung unterstützen. [D]
Erläuterung: Es soll insbesondere One Class SVM und AutoEncoder wie in Scikit-learn unterstützt werden.
 - Der KI-Baukasten **soll** Verfahren zur Zustandserkennung bereitstellen. [D]
Erläuterung: Die Verfahren sind hier als Basismenge aufgezählt um den Entwurf des KI-Baukastens zu unterstützen. R114.h könnte durch R114.g realisiert werden. Die Bereitstellung und Integration fällt in die Verantwortlichkeit der nutzenden Demonstratoren, des Think Tank „KI & Produktion“ sowie, für die Nutzung in weiterem Umfeld, dem „KI-Accelerator“ (insbesondere für R114.a).
- R115. Der KI-Baukasten **soll** KI-Modelle in standardisierten Formaten unterstützen. [V, E]
Erläuterung: Modelle in standardisierten Formaten wie z.B. ONNX [23] unterstützen die Anwendung verschiedener (Implementierungen von) KI-Verfahren wie auch deren Deployment. Der KI-Baukasten soll die Verwendung derartiger Formate unterstützen und fördern.
- R116. Der KI-Baukasten **soll** Verfahren zur Vor- und Nachverarbeitung von Daten zur Verfügung stellen [T].
- Verfahren zur Vor- und Nachverarbeitung **sollen** generisch, parametrisierbar und anpassbar sein. [D]
 - Der KI-Baukasten **kann** Dienste zum Feature Design / digitalen Signalverarbeitung (Bandpassfilter, FTF-Transformationen, Running-RMS) bereitstellen. [T]
Erläuterung: Die Verfahren sind hier als Basismenge aufgezählt um den Entwurf des KI-Baukastens zu unterstützen. Die Bereitstellung und Integration fällt in die Verantwortlichkeit der nutzenden Demonstratoren, des Think Tank „KI & Produktion“ sowie, für die Nutzung in weiterem Umfeld, dem „KI-Accelerator“.
- R117. Der KI-Baukasten und die enthaltenen KI-Verfahren **sollen** Transparenz und Nachvollziehbarkeit über ihre Entscheidungsfindung bieten. [D]
- R118. Die Plattform **muss** dem KI-Baukasten zentrale Dienste bereitstellen.
- Die Plattform **soll** dem KI-Baukasten Warn- und Alarmierungsdienste bereitstellen. [D, U]
 - Die Plattform **muss** dem KI-Baukasten Speicherdienste bereitstellen. [D]
 - Die Plattform **muss** dem KI-Baukasten Sicherheitsdienste bereitstellen. [Q]
 - Die Plattform **muss** dem KI-Baukasten Datenschutz-Dienste bereitstellen. [Q]
 - Die Plattform **muss** dem KI-Baukasten Datenintegrations-Dienste bereitstellen. [Q]

Erläuterung: Einige Teilanforderung dienen nur zur Klärung, dass zentrale Plattformdienste insbesondere dem KI-Baukasten bereitgestellt werden müssen. Genannt sind hier insbesondere Speicherdienste (R118.b vs. Abschnitt 3.6), Sicherheitsdienste (R118.c vs. Abschnitt 3.7), Datenschutz-Dienste (R118.d vs. Abschnitt 3.8) und Datenintegrations-Dienste (R118.e vs. Abschnitt 3.10). Die Verknüpfung der Dienste erfolgt über Datenpfade (R20) im Rahmen der Anwendungs-Konfiguration (Abschnitt 3.16). Nicht erwähnt ist beispielsweise das dynamische bzw. adaptive Deployment oder der adaptive Austausch von KI-Verfahren, da diese Plattformdienste für den KI-Baukasten möglichst transparent zu realisieren sind.

- R119. Das Training von KI-Methoden **soll** automatisch und parallel im Hintergrund erfolgen und das trainierte Modell dann den Komponenten der Plattform angeboten werden. [D]
- Die Freigabe des trainierten Modells **soll** manuell durch einen Benutzer mit entsprechenden Rechten erfolgen. [D]
 - Die Freigabe des trainierten Modells **soll** über Einstellungen im Konfigurationsmodell erfolgen. [D]
 - Die Freigabe des trainierten Modells **kann** (wenn im Konfigurationsmodell festgelegt) automatisch erfolgen. [Q]
 - Die durch eine Freigabe eines trainierten Modells angestoßenen Änderungen **sollen** nachvollziehbar und rückverfolgbar sein. [D]
 - Die durch eine Freigabe angestoßenen Änderungen **sollen** reversibel sein. [D]

Erläuterung: a entspricht der Freigabe-Stufe 1 in [28], d.h., ereignisgetriebenes Lernen mit menschlicher Freigabe. b weicht die Freigabe-Stufe 1 in Richtung teil-automatisierter Freigaben für konfigurierte Komponenten bzw. Anwendungen auf. c entspricht Freigabe-Stufe 2, d.h., automatisches Freigeben und Lernen [28], ggf. basierend auf konfigurierbaren Freigabe- oder Gütekriterien.

3.14 Adaptive Dienst/Komponenten-Auswahl

Üblicherweise ist die Auswahl von KI-Methoden und KI-Komponenten zurzeit relativ statisch. Dennoch kann sich die Einsatzumgebung im Produktionsumfeld schnell ändern, z.B., weil neue Sensoren bzw. neue Edge-Devices in einen Produktionsablauf eingebracht werden, die die Datenmengen, die Datenarten oder auch deren Qualität ändern. In derartigen Situationen kann es einfacher sein, die verwendete KI-Methode anzupassen bzw. Parameter der KI-Methode/KI-Komponente zu ändern. Dies kann sowohl mit dem dynamischen Deployment (siehe Abschnitt 3.5) als auch ggf. effizienter mit einer Anpassung der KI-Komponente in ihrem Deployment gelöst werden. Die Adaption kann R9 (Availability) unterstützen bzw. realisieren.

- R120. Das Konfigurationsmodell **muss** für eine KI-Methode alternative KI-Komponenten beschreiben können. [Q, A]
- R121. Die Plattform **soll** die Verwendung von KI-Methoden überwachen und Situationen erkennen, die einen Austausch von KI-Methoden erforderlich machen. [Q]
- R122. Die Plattform **soll** basierend auf dem Konfigurationsmodell über den optimalen Einsatz von KI-Methoden im Rahmen von R120 entscheiden können. [Q]
- Die Adaption **muss** auf dem Konfigurationsmodell beruhen. [Q]
 - Die Adaption **muss** die Validität des Konfigurationsmodells regelmäßig überprüfen. [Q]
 - Die Adaption **muss** ihre Entscheidungen im Konfigurationsmodell hinterlegen. [Q]
 - Die Adaption **muss** die Validität des Konfigurationsmodells sicherstellen. [Q, U]
 - Die Adaptation einer Plattform-Instanz mit 50 Ressourcen und 5 Applikationen **soll** innerhalb von 1 Minute erfolgen. [Q]
 - Die Adaptation **soll** Komponenteneigenschaften berücksichtigen. [U]

Erläuterung: Die Entscheidung kann statisch erfolgen, soll aber (üblicherweise) dynamisch zur Laufzeit passieren. R122.d beruht auf R8.b. Komponenteneigenschaften (R122.f, R96.a) können die Verteilung von Komponenten beeinflussen, ggf. Ressourcen ausschließen, z.B., Ressourcen außerhalb einer Fabrik wegen rechtlicher Gründe oder Lizenzgründe.

- R123. Die Durchsetzung der Adaption **muss** über Plattformfunktionen und Verwaltungsschalen erfolgen. [Q]
 a. Die Plattform **kann** Mechanismen zum Update von Diensten anbieten. [U]
Erläuterung: Die Adaption kann sich über Konnektoren auf unterliegende Plattformen bzw. deren Dienste auswirken (R13).
- R124. Die Adaption **muss** die Ausführbarkeit der Applikationen berücksichtigen. [Q]
Erläuterung: Die Adaption darf die Ausführung von Applikationen mit verteilten Diensten/Komponenten nicht stören, d.h., die Adaption darf keine negativen Auswirkungen auf die Ausführung von Produktionsprozessen haben bzw. die erforderlichen Anpassungsmechanismen müssen entsprechend ausgelegt sein.
- R125. Die Adaption **soll** Transparenz und Nachvollziehbarkeit über die Entscheidungsfindung bieten. [D]
- R126. Die Plattform **soll** selbstständig nach Möglichkeiten der Optimierung (des KI-Einsatzes) suchen und diese zentral als Vorschlag anbieten. [D, Q]
Erläuterung: Analog zu R119 kann die Adaption autonom ablaufen oder aber, je nach Konfigurationseinstellungen, Freigaben erfragen. R119 gilt hier analog.

3.15 Virtualisierung

Digitale Zwillinge werden im Bereich der industriellen Produktion für verschiedene Zwecke eingesetzt. Leider ist für digitale Zwillinge zurzeit keine Standardisierung absehbar [37].

- R127. Die Plattform **soll** die Einbindung von Digitalen Zwillingen ermöglichen. [Q]
- R128. Digitale Zwillinge **müssen** als Verwaltungsschale (Interface) beschrieben sein. [Q]
- R129. Die Plattform **kann** bereitgestellte Container mit installiertem digitalem Zwilling unterstützen und verwalten. [Q, D]
Erläuterung: Aufgrund der fehlenden Standardisierung [37] ist von einer manuellen Installation als Container auszugehen. Falls die Container den Vorgaben der Plattform entsprechen, z.B., bezüglich Verteilbarkeit (siehe R34.b), dann kann die Plattform die Verwaltung der Container übernehmen. Weitere Installationsformen sind unter anderem virtuelle Maschinen oder direkte Installation auf einem Rechner. Diese Installationsformen können von der Plattform wahrscheinlich nicht verwaltet werden.
- R130. Die Plattform **kann** Komponenten zur Ausführung von Digitalen Zwillingen integrieren. [Q]
Erläuterung: Beispielsweise kann die Plattform entsprechende Frameworks wie Eclipse Ditto [7] bereitstellen um die Integration von Digitalen Zwillingen zu erleichtern.

3.16 Anwendungsunterstützung

Industrielle Plattformen bieten oft die Möglichkeit, vorkonfigurierte Applikationen auszuführen bzw. bereitgestellte Applikationen (als „Apps“) auf eine Plattform zu laden [33]. Um die Anwendbarkeit und Bekanntheit der IIP-Ecosphere Plattform zu steigern, benennen wir hier die Möglichkeit eines „App Stores“ im Wissen, dass Ressourcen aufgrund der bewilligten Teilprojektbeschreibungen ggf. nur nach Umwidmung (ggf. im KI-Accelerator) zur Verfügung stehen könnten. Ähnlich ist zurzeit der Einsatz von Ressourcen zur Realisierung eines Benutzerinterfaces oder von Visualisierungen zu bewerten. Da Anwendungen aus Plattform- und KI-Diensten zusammengesetzt sind, liegt die letztendliche Verantwortung für die Qualität (z. B. Antwortzeiten, Speichernutzung, Terminierung) der Anwendung beim Anwendungsdesigner oder Benutzer, wie in Abschnitt 3.13 erläutert.

- R131. Die Plattform **muss** die Erstellung und Bereitstellung von Applikationen für die intelligente Produktion unterstützen. [Q]
- Das Konfigurationsmodell **muss** Applikationen sowie deren benötigte Dienste, Konnektoren, Datenpfade bzw. die erforderlichen Ressourcen beschreiben können. [Q]
 - Das Konfigurationsmodell **muss** die Versionierung von Applikationen und Diensten/Komponenten ermöglichen. [U]
 - Das Konfigurationsmodell **kann** die Parametrisierung von Applikationen ermöglichen. [U]
 - Das Konfigurationsmodell **soll** Applikations-Templates [36] zur vereinfachten Konfiguration von Anforderungen unterstützen. [U]
 - Das Konfigurationsmodell **muss** abhängige Applikationen oder Dienste beschreiben. [U]
 - Die Konfiguration von Applikationen und Datenpfaden **kann** in grafischer Weise erfolgen. [Q]
 - Die Plattform **muss** das Deployment von Applikationen unterstützen. [U]
 - Die Plattform **soll** die Kommunikation zwischen Plattform-Instanzen unterstützen. [U]
 - Die Plattform **soll** die Kommunikation zwischen Applikationen unterstützen. [U]
- Erläuterung:* Applikationen bestehen insbesondere aus benötigten Komponenten bzw. Diensten die von der Plattform angeboten werden (z.B. Datenschutzdienste, KI-Dienste) aber auch applikationsspezifische Dienste (R132), ggf. alternativen Komponenten bzw. Diensten, sowie den Datenpfaden (R20) zwischen den Komponenten und Datenendpunkten (z.B., Sensoren, Aktoren, Speicherdienste). Applikationen können Abhängigkeiten zu Diensten bzw. weiteren Applikationen deklarieren (R131.e), die z.B. vor der abhängigen Applikation gestartet werden müssen. Das Deployment von Applikationen (R131.g) folgt aus dem Deployment von Komponenten/Diensten, muss aber gesondert benannt werden, da es auch die Konfiguration von Datenkonnektoren [36] beinhalten/erfordern kann. Verbindungen zwischen Applikationen (R131.i) müssen ggf. manuell erstellt werden und soll insbesondere zur Verbindung zwischen Plattformen dienen (R131.h).
- R132. Die Plattform **kann** applikations-spezifische Komponenten unterstützen. [Q]
- Das Konfigurationsmodell **muss** applikations-spezifische Dienste unterstützen. [Q]
- R133. Die Plattform **muss** Laufzeit-Unterstützung für Anwendungen bereitstellen. [U]
- Die Plattform **muss** den Status der Dienste kennen. [U]
 - Die Plattform **muss** den Status der laufenden Anwendungen kennen. [U]
 - Die Plattform **kann** Funktionen anbieten um den Status der laufenden Anwendungen anzupassen. [U]
 - Die Plattform **soll** Funktionen anbieten um Fehlerzustände in einer Anwendung zu beheben. [U]
- Erläuterung:* Der Status ist eine Zusammenfassung des Betriebszustands eines Dienstes/einer Applikation, also z.B., „deployed“, „starting“, „running“, „failed“, „reconfiguring“. Der Status kann aufgrund des Status der Dienste der Anwendung bestimmt werden. In Grenzen kann es erforderlich den Status explizit zu setzen, für einige Funktionen wie Deployment, Optimierung (Abschnitt 3.12) und Adaption (Abschnitt 3.14) muss die Plattform den Status geeignet anpassen. In Grenzen soll die Plattform die Ausführung auf dieser Ebene überwachen und Fehler beheben (R133.d), z.B., durch den Neustart von Diensten.

- R134. Die Plattform **muss** das Entfernen von Applikationen/Diensten unterstützen. [U]
- Die Plattform **muss** das Entfernen deployter Applikationen von den Ressourcen unterstützen. [U]
 - Die Plattform **kann** das Entfernen von Applikationen aus dem Konfigurationsmodell unterstützen. [U]
 - Die Plattform **kann** das Außerdienststellen von Diensten ermöglichen. [U]
 - Die Plattform **soll** das Lösen von Verbindungen zwischen Applikationen ermöglichen. [U]
 - Die Plattform **soll** das Lösen von Verbindungen zwischen Plattformen ermöglichen. [U]
Erläuterung: Hierbei handelt es sich um die Gegen-Funktion zum Deployment von Applikationen (R134). Je nach Implementierungen können Dienste durch neuere Versionen ersetzt, außer Dienst gestellt (aber in der Plattform behalten) oder ggf. sogar gelöscht werden (R134.c). Verbindungen zwischen Applikationen und Plattformen (R134.d, R134.e) müssen ggf. manuell beendet werden [36].
- R135. Die Plattform **soll** das Update von Applikationen unterstützen. [U]
Erläuterung: Updating an application in a production environment is a critical activity. This can be done by shutting down the application (offline) or (partially) during runtime (online). If necessary, update steps can be denied as part of the validation of the configuration model (R94), e.g. if compatibility between interfaces is violated. Through update operations, possibilities for the evolution of applications are enabled. This may also include an update of the connection between applications and platforms (R131.h, R131.i).
- R136. Die Plattform **kann** Funktionen zum Update der Plattform enthalten. [U]
- Die Plattform **soll** ein Update der Ressourcen-Abstraktionsschicht ermöglichen. [U]
Erläuterung: Die Ressourcen-Abstraktionsschicht („ECS runtime“ in [36]) wird hier als Bestandteil der Plattform und weniger der Applikationen/Dienste gesehen. Diese Abstraktionssicht muss ggf. im Laufe der Entwicklung der IIP-Ecosphere Plattform auf einen neuen Stand gebracht werden, was im Rahmen dieser Anforderung geschehen kann. Idealerweise kann die Abstraktionsschicht sich selbst updaten, ggf. ist allerdings auch eine Interaktion mit dem (Store des) Geräteherstellers erforderlich.
- R137. Die Plattform **kann** Funktionen zur Absicherung des Deployments von Applikationen unterstützen [U].
- Die Plattform **kann** die Simulation der Integration von Ressourcen unterstützen. [U]
 - Die Plattform **kann** eine Pre-Deployment-Simulation unterstützen. [U]
 - Die Plattform **kann** einen Test-Durchlauf einer Applikation unterstützen. [U]
Erläuterung: Das Deployment einer Applikation in eine Produktionsumgebung ist eine kritische Aktivität. Daher werden in [36] Mechanismen wie Simulationen (R137.a) vorgeschlagen, auch um den Ressourcenbedarf abzuschätzen, zu validieren und ggf. anzupassen.
- R138. Die Plattform **kann** einen App-Store bereitstellen. [Q]
Erläuterung: Aufgrund der gültigen Teilprojektbeschreibungen ist zurzeit kein App-Store vorgesehen, obwohl eine entsprechende Funktionalität sehr hilfreich wäre. Daher ist diese Anforderung als „kann“ priorisiert. Das Entfernen einer Applikation/eines Dienstes aus dem Konfigurationsmodell (R134.b) kann ein Entfernen aus dem Store nach sich ziehen. Weiterhin sollten Dienste/Anwendungen bei Installation in einer Plattform-Instanz in deren Konfigurationsmodell übernommen werden.
- R139. Die Plattform **kann** eine Benutzeroberfläche bereitstellen. [Q. P]
- Die Plattform **kann** eine Benutzeroberfläche für Agentenmanagement und -entwicklung bereitstellen. [T]

- b. Die Plattform **kann** eine Benutzeroberfläche für die „Tools“ aus [36] bereitstellen. [U]
Erläuterung: Industrielle Plattformen bieten üblicherweise eine Benutzeroberfläche, z.B., unter Verwendung webbasierter Technologien. Da das Ziel des Think Tanks „Plattformen“ die Erforschung und Evaluation von Plattformkonzepten ist und weniger auf die Entwicklung einer vollumfänglichen Industrie 4.0 Plattform oder auf Benutzeroberflächen liegt, sind primär keine Ressourcen für die Entwicklung einer Benutzeroberfläche vorgesehen. Dennoch wäre eine (einfache) Benutzeroberfläche für eine einfach nutzbare und demonstrierbare Plattform wünschenswert.
- R140. Die Plattform **soll** Datenanalysten bei der Erstellung von KI-Diensten unterstützen („Data Science tool chain“ in [36]). [U]
- a. Die Plattform **soll** geeignete „nördliche“ Schnittstellen zur Verfügung stellen. [U]
- b. Die Plattform **soll** geeignete Ressourcen zur Verfügung stellen. [U]
Erläuterung: R140.a erweitert R19.d und R140.b bietet Zugriff auf das Deployment (Abschnitt 3.5) und die Ressourcenzuteilung (Abschnitt 3.12).
- R141. Die Plattform **kann** Visualisierung von Applikationen oder Daten ermöglichen. [Q, D]
Erläuterung: Dies kann insbesondere durch externe „nördliche“ Konnektoren bzw. über entsprechende Verwaltungsschalen erfolgen. Generell sind Visualisierungsfähigkeiten aufgrund der gültigen Teilprojektbeschreibungen außerhalb der Plattform zu sehen, obwohl Visualisierungsfähigkeiten in verschiedenen Umgebungen wie z.B. zur Demonstration sehr nützlich sind. Denkbar ist die Anbindung externer (teilweise durch die Demonstratoren genannte) Frameworks wie Grafana [11], BaSys4Dash [2], Rapidminer [30] oder Pandas [24]. Die Auswahl zur Unterstützung soll in Absprache mit den Demonstratoren und den jeweiligen Plänen zur öffentlichen Demonstration erfolgen.

4 Abgleich mit der Benutzungssicht (Usage View)

Im Rahmen der Anforderungserhebung haben Partner des IIP-Ecosphere-Projekts neben dieser funktionalen Anforderungssammlung zur Abrundung eine Benutzungssicht (usage view, nach [12]) erhoben. Das Ergebnis wird in [36] vorgestellt und umfasst die Beschreibung des sogenannten System Under Consideration, 18 Entitäten, 19 Rollen und 67 Aktivitäten. Die Erhebung der Benutzungssicht erfolgte in einer Reihe von Workshops mit IIP-Ecosphere Partnern. Dabei wurde die Vision aus Abschnitt 2 als Rahmen (Scope) vorgestellt.

Die Erkenntnisse aus [36] wurden in die funktionalen Anforderungen in diesem Dokument eingearbeitet. Hierfür wurden die Anforderungen in dieses Dokument während der Erstellung der Benutzungssicht nicht mehr geändert und ein Abgleich zum Ende der Arbeiten an [36] durchgeführt. Falls die Benutzungssicht nun eine bereits zum Zeitpunkt der Integration bekannten Anforderung beschreibt, wird die Benutzungssicht mittels „[U]“ als Quelle in den Anforderungen eingetragen. Falls die Benutzungssicht eine neue Anforderung formuliert, wird die Menge der Anforderungen in diesem Dokument ergänzt und die Ergänzung in Tabelle 1 verzeichnet. Damit kann dieses Dokument als mehr technische Brücke zwischen Benutzungssicht und Funktionaler Sicht/Architektur nach [12] verstanden werden.

Tabelle 1 beschreibt jede, direkt durch den Abgleich mit [36] hinzugekommenen Anforderungen. Durch die dabei durchgeführten Strukturänderungen sind teilweise weitere Anforderungen hinzugekommen bzw. existierende Anforderungen wurden zu Referenzzwecken detailliert – diese Änderungen sind hier nicht beschrieben. Aufgeführt sind die betroffenen Anforderungen in diesem Dokument, die auslösenden Abschnitte in [36] (wir benennen insbesondere die ersten in Inhaltsfolge, nicht notwendigerweise alle referenzierenden Abschnitte) sowie die Art der Änderung, d.h., Erweiterung (Hinzufügen von Anforderungen oder Unteranforderungen) oder Klarstellung der Beschreibung existierender Anforderungen. Es wurden keine Anforderungen während des Abgleichs entfernt.

Der Abgleich erfolgte entlang der Reihenfolge der Kapitel in [36], d.h., Tabelle 1 war ursprünglich nach den Abschnitten in [36] sortiert. Durch finale Revisionen in [36] konnte die Sortierung nicht aufrechterhalten werden. Die Einträge „[Platform inter-connections]“ beziehen sich auf initiale Beschreibungen von Aktivitäten zur Verwaltung von Datenverbindungen zwischen Applikationen/Plattformen. Diese wurden bereits beim Abgleich berücksichtigt, letztlich aber auf eine spätere Version von [36] verschoben. Diese zusätzlichen Anforderungen wurden aus diesem Dokument nicht mehr entfernt.

Anforderung	Abschnitt/Thema in [36]	Erw.	Klarst.
R4.d, R4.e	Abschnitt 3.8 „Services“ (monitoring)	X	
R4.c	Abschnitt 3.8 „Services“ (state) Abschnitt 3.16 „Application“ (state)	X	
R17.a, R17.b	Abschnitt 3.8 „Services“ (connectors)	X	X
R20.b, R20.c	Abschnitt 3.8 „Services“ (relations)		X
R78	Abschnitt 3.9 „Service store“	X	
R131.d	Abschnitt 3.9 „Service store“ (templates)	X	
R25	Abschnitt 3.13 “Device description store”, Abschnitt 3.14 “Device configuration tool”, Abschnitt 4.13 “Asset Data provider”, Abschnitt 5.1.1.2 “Adding an ECS device“		X
R123.a	Abschnitt 3.15 “Runtime application / service distribution tool“ (updates)	X	
R102.a	Abschnitt 3.14 “Device configuration tool“	X	

Anforderung	Abschnitt/Thema in [36]	Erw.	Klarst.
R140	Abschnitt 3.18 "Data Science tool chain"	X	
R139.b	Abschnitt 3 alle "tools"	X	
R25.a, R25.c, R25.d	Abschnitt 4.2 "Edge device provider" (onboarding), Abschnitt 4.3 "Edge runtime provider"	X	
R25.e, R25.f	Abschnitt 4.2 "Edge device provider" (capabilities)		X
R27	Abschnitt 4.2 "Edge device provider"		X
R28	Abschnitt 4.4 "Cloud provider"		X
R131.b	Abschnitt 5.1.6.1 "Deploying an (distributed) application" (versioning)	X	
R104.b, R122.d, R8.b,	Abschnitt 5.1.6.1 "Deploying an (distributed) application" (validation, checking)	X	X
R131.e	Abschnitt 5.1.6.1 "Deploying an (distributed) application" (validation, checking)	X	
R136	Abschnitt 5.1.6.1 "Deploying an (distributed) application" (pre-deployment simulation), Abschnitt 5.1.6.2 "Updating an (distributed) application" (deployment testing) [Plattform inter-connections]	X	
R17.c, R20.b, R112.a, R131.c, R131.g	Abschnitt 5.1.6.1 "Deploying an (distributed) application" (parameters)	X	
R135	Abschnitt 5.1.6.2 "Updating an (distributed) application"	X	
R133	Abschnitt 5.1.6.3 "Uninstalling an (distributed) application" [Plattform inter-connections]	X	
R136	Abschnitt 5.1.6.5 "Updating ECS runtime"	X	
R36	Abschnitt 5.1.6.7 "Replace an ESC device" Abschnitt 5.1.6.8 "Restore an ECS device"	X	
R37	Abschnitt 5.1.6.9 "Remote management of ECS devices"	X	
R136	Abschnitt 5.1.6.4 „Update of the ECS management system“ Abschnitt 5.1.6.5 "Update of an ECS runtime"	X	
R96.a, R103.c, R122.f	Abschnitt 5.1.4.1 „Provision of a service“	X	
R11.c	Abschnitt 5.1.4.1 „Provision of a service“ (logging)	X	
R134.c	Abschnitt 5.1.4.3 „Discontinuation of a service“	X	
R133.h	[Plattform inter-connections]	X	
R4.e	Abschnitt 5.1.4.4 "Defining monitoring parameters for services", Abschnitt 5.1.5.2 "Defining notification triggers (conditions) for monitoring parameters"	X	
R137.a	Abschnitt 5.1.7.2 "Simulating the Integration of ECS devices for an application"	X	
R30.c	Abschnitt 5.1.7.5 "Integrating external container registries"	X	
R133	Abschnitt 5.1.7.6 "Changing the state of an application", Abschnitt 5.1.7.7 "Restoring an operational application"	X	
R102.b	Abschnitt 5.2.1.1 "Creating a data exploration process"	X	
R76	Abschnitt 5.2.1.1 "Creating a data exploration process"		X
R73.d	Abschnitt 5.2.2.10 "Backup of labeled data sets (for further learning/training)"	X	

Tabelle 1: Abgleich der funktionalen Anforderungen mit der Benutzungssicht, Erw.=Erweiterung, Klarst.=Klarstellung

Neben verpflichtenden Plattformdiensten kann das Dienst-Konzept der IIP-Ecosphere-Plattform auch zur Durchführung von Wartungsarbeiten eingesetzt werden, z.B., Dienste zum Update von Firmware auf Feld- oder Edge-Geräten (Abschnitt 5.1.3.7 „Update firmware“, [36]). Derartige optionale Dienste sind nicht in eigenständige Anforderungen umgesetzt worden.

Nicht alle Aktivitäten aus [36] schlagen sich direkt in diesem Dokument nieder. Das ist darin begründet, dass bereits in [36] diverse Aktivitäten aus dem Umfang der IIP-Ecosphere-Plattform ausgeschlossen wurden. Weiterhin gibt es diverse Aktivitäten, die auf den größeren Lebenszyklus eines Systems zielen, die wir hier nicht abbilden können. Dennoch haben wir versucht, so viele wie mögliche Aktivitäten und Informationen zu übernehmen, was sich in 10 neuen Anforderungen (mit 20 Unter-Anforderungen), insgesamt 53 neuen Unter-Anforderungen und 9 geänderten/geklärten Anforderungen darstellt. Durch die oben angesprochene Umstrukturierung bestehender Anforderungen sind weitere 10 Unter-Anforderungen hinzugekommen.

5 Priorisierung und Anforderungskonflikte

Die in Abschnitt 3 beschriebenen Anforderungen sind durch die drei Kategorien „muss“, „soll“ und „kann“ bereits grob priorisiert. Bezogen auf alle Anforderungen (inklusive Unter-Anforderungen) haben wir 41% als „muss“, 43% als „soll“ und 16% als „kann“ Anforderungen priorisiert. Die Priorisierung lehnt sich dabei an den Vorschlag von [36] an, wobei ggf. Anpassungen im Kontext der hier beschriebenen Anforderungen erforderlich waren.

Für die Umsetzung in eine Plattformarchitektur bzw. Realisierung als Quellcode empfehlen wir die folgende Themen-Reihenfolge:

- 1) Konnektoren und Verbindungen (Abschnitt 3.4)
- 2) Verteilbarkeit von Anwendungen
 - Heterogenes, dynamisches Deployment (Abschnitt 3.5)
 - Konfigurierbarkeit (Abschnitt 3.11)
 - Anwendungsunterstützung (Abschnitt 3.16)
 - Virtualisierung (Abschnitt 3.15)
- 3) Dienste
 - KI-(Service-)Baukasten (Abschnitt 3.13)
 - Zentrale Speicherdienste (Abschnitt 3.8)
 - Datenintegration (Abschnitt 3.10)
- 4) Fortgeschrittene Funktionalität
 - Data Sharing (Abschnitt 3.9)
 - Optimiertes bzw. adaptives Deployment (Abschnitt 3.12)
 - Adaptive Dienst/Komponenten-Auswahl (Abschnitt 3.14)

Dabei legen wir die wahrscheinlichen Abhängigkeiten zu Grunde und haben, von der Basis der Plattform her beginnend, möglichst schnell die Verteilung von Diensten sowie die Entwicklung von Anwendungen zur Demonstrierbarkeit bzw. für erste Experimente im Blick. Sicherheit (Abschnitt 3.6) und Datenschutz (Abschnitt 3.7) sind (wie eigentlich auch Virtualisierung und Konfigurierbarkeit), Querschnittsthemen, die an geeigneter Stelle (eigentlich von Grund auf) mitberücksichtigt werden müssen. Die „Generellen Anforderungen“ (Abschnitt 3.3) bilden den Rahmen und sind in jedem Schritt geeignet zu berücksichtigen.

Natürlich werden im Rahmen der weiteren Umsetzung Abwägungen zwischen Anforderungen getroffen werden müssen bzw. Konflikte identifiziert werden, die dann geeignet gelöst werden müssen. Wir führen hier nur einige Beispiele von Anforderungskonflikten auf.

- Zusätzliche Datenformate wie JSON/XML via REST (R19.c) könnten mit der generellen Anforderung nach einheitlichen Schnittstellen basierend auf AAS (R7) kollidieren. Der Wunsch nach REST ist durchaus verständlich, da es doch eine gängige Technologie ist, die auch als Schnittstellenansatz in verschiedenen aktuellen Plattformen eingesetzt wird [33]. Allerdings ist zu berücksichtigen, dass AAS-Umsetzungen oft auf REST aufsetzen, d.h., in dem Fall kann R19.c implizit als erfüllt angesehen werden, insbesondere wenn die entsprechenden Daten einfach als Zeichenketten in JSON/XML in AAS verwendet werden können oder in geeignete interne Datenformate umgewandelt werden können (siehe Transportformat in R14).
- Die manuelle Freigabe automatisierter Entscheidungen (R119) kann den Grundzielen dynamischer Optimierung und Adaption entgegenstehen. Aus diesem Grund haben wir die verschiedenen Freigabe-Stufen diskutiert und mit optionalen Anforderungen Möglichkeiten vorgesehen, über die Konfiguration Freigaben zu automatisieren. Damit sollte unterschiedlichen Bedürfnissen Rechnung getragen sein.

- Der KI-Service-Baukasten in Abschnitt 3.13 benennt viele Anforderungen, die zumindest technische Abhängigkeits-Konflikte auslösen könnten, z.B. die verschiedenen KI-Frameworks. Dies ist auch ein Grund, warum die meisten technischen/methodischen Anforderungen in Abschnitt 3.13 optional sind, und ggf. nicht auflösbare Konflikte zu einer zulässigen Nichterfüllung von Anforderungen führen darf.
- Die Anforderungen für Benutzerschnittstelle und Visualisierung in Abschnitt 3.16 kollidieren mit den Arbeitsplänen und den geplanten Ressourcen. Dennoch ist es wünschenswert und nicht ausgeschlossen, dass Teile davon im günstigsten Fall realisiert werden. Daher sind diese Anforderungen ebenso als optional/nice-to-have eingestuft.
- Die Möglichkeit zur Konfiguration von Datenformaten kann mit der Integrität der Daten in Konflikt stehen. Diese Konflikte können teilweise durch globale Einschränkungen (Constraints) beschrieben und bei der Konfiguration einer Plattforminstanz verhindert werden. Zudem können anwendungsspezifische Einschränkungen verwendet werden. Allerdings ist es durchaus möglich, dass nicht alle potentiellen Konflikte beschrieben werden können bzw. spezifiziert wurden. Zudem können Konflikte bestehen, die grundsätzlich nicht vermeidbar sind.
- Die Qualitäts-Anforderungen (R10, R19.a, R19.e, R22, R28, R35, R91) können weitere Konflikte auslösen, insbesondere bei der Realisierung/Evaluierung. Analog können Sicherheits- und Datenschutzerfordernungen (Abschnitte 3.6 und 3.7) andere Qualitätseigenschaften beeinflussen, z.B. Performanz. Soweit wir sehen können sind aufgrund der vorgesehenen Technologien keine Konflikte zu erwarten. Technisch können diese teilweise durch austauschbare Alternativkomponenten gelöst werden, oder aber auch durch diese erst eingeführt werden.

6 Zusammenfassung

Dieses Dokument beschreibt die Ergebnisse der Anforderungsermittlung aus funktionaler Sicht. Hierzu wurden sowohl die in [36] beschriebene Benutzungssicht als auch eine eigenständige Anforderungsermittlung durch Gespräche, Interviews und Befragung (Fragebogen) integriert. Insgesamt wurden 141 Anforderungen mit 181 Unter-Anforderungen in 14 Themenbereichen erhoben, beschrieben, kategorisiert und initial priorisiert. Abbildung 5 stellt die Verteilung alle Anforderungen (inklusive Unter-Anforderungen) auf die 14 Themenbereiche grafisch dar.

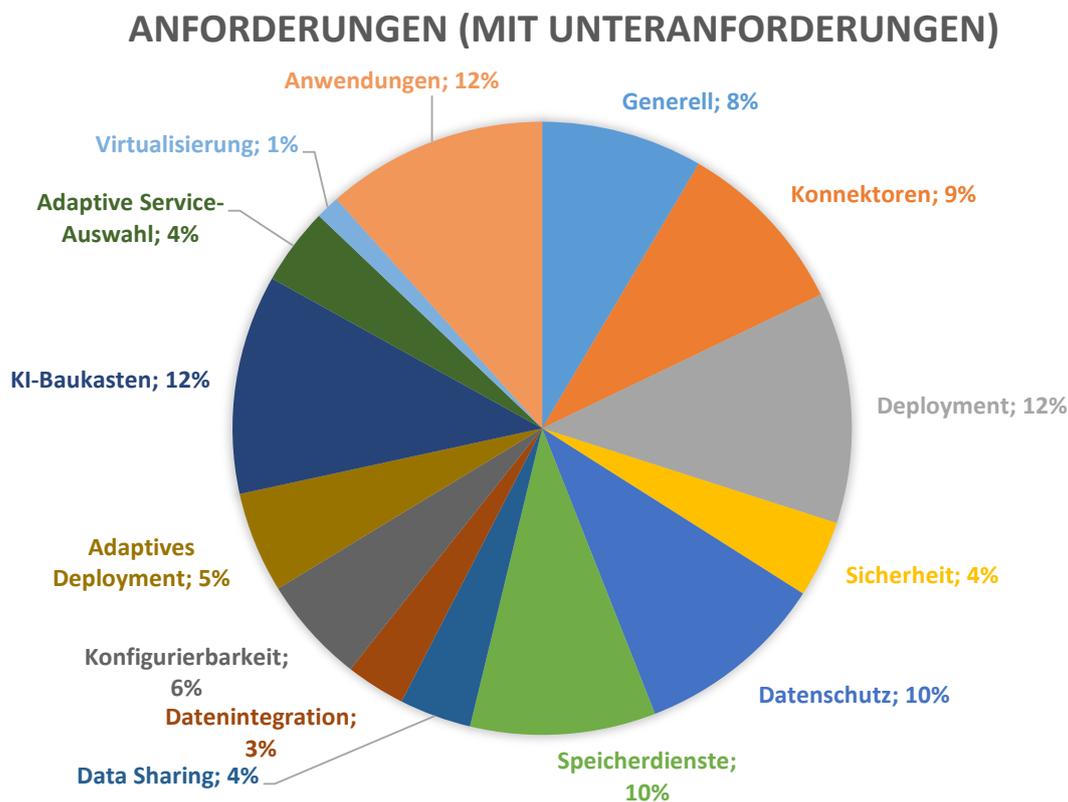


Abbildung 5: Verteilung der Anforderungen auf Themenbereichen

Basierend auf der IIP-Ecosphere Plattform-Übersicht [33], die die Grundlage für die Entwicklung einer gemeinsamen Plattform-Vision gelegt hat, und basierend auf den Anforderungen aus Benutzungssicht [36] komplettiert dieses Dokument die Anforderungsermittlung für die IIP-Ecosphere-Plattform. Wie bereits für die Benutzungssicht beschrieben, werden wir weitere Anforderungen, die ggf. während der Projektlaufzeit aufscheinen, in inkrementelle Versionen dieses Dokuments fortschreiben. Später hinzukommende Anforderungen werden wahrscheinlich nicht mehr die Entwicklung der IIP-Ecosphere-Plattform (stark) beeinflussen können, sollen aber zukünftigen Arbeiten als umso vollständigere Anforderungssammlung dienen.

In IIP-Ecosphere werden als nächstes die hier und in [36] beschriebenen Anforderungen in eine Architektur und dann in eine Realisierung der Plattform umgesetzt.

Wir bedanken uns bei allen, die mit ihren Beiträgen zu dieser Anforderungssammlung aktiv beigetragen haben.

7 Referenzen

- [1] BaSys 4.0, <https://www.basys40.de/>
- [2] BaSys Satellitenprojekte, BaSys4Dash, <https://www.basys40.de/satellitenprojekte/>
- [3] K. Czarnecki and C. H. P. Kim. Cardinality-based feature modeling and constraints: a progress report. Intl. Workshop on Software Factories at OOPSLA'05, 2005.
- [4] DevOpt – DevOps für Selbst-Optimierende Emergente Systeme, <https://sse.uni-hildesheim.de/forschung/projekte/devopt/>
- [5] Docker, <https://www.docker.com/>
- [6] Eclipse BaSyX, <https://www.eclipse.org/basyx/>
- [7] Eclipse Ditto, <https://projects.eclipse.org/projects/iot.ditto>
- [8] Eclipse IoT, <https://iot.eclipse.org/>
- [9] H. Eichelberger, C. Qin, R. Sizonenko, K. Schmid, Using IVML to model the topology of big data processing pipelines, International Systems and Software Product Line Conference, 204-208, 2016
- [10] FIPA <http://www.fipa.org/index.html>; <http://www.fipa.org/specs/fipa00061/SC00061G.pdf>
- [11] Grafana, <https://grafana.com/>
- [12] The Industrial Internet Reference Architecture Technical Report, <https://www.iiconsortium.org/pdf/IIRA-v1.9.pdf>
- [13] International Data Spaces Association, <https://www.internationaldataspaces.org/>
- [14] International Data Spaces Association, Reference Architecture Model, Version 3.0, April 2019, <https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-Reference-Architecture-Model-3.0.pdf>
- [15] ITOperations, IT/OT convergence, <https://searchitoperations.techtarget.com/definition/IT-OT-convergence>
- [16] JADE, <http://jade.tilab.com/>
- [17] Joint Task Force, Security and Privacy Controls for Information Systems and Organizations, Draft NIST Special Publication 800-53, rev. 5, <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5-draft.pdf>
- [18] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, A. S. Peterson, Feature-Oriented Domain Analysis (FODA) Feasibility Study, CMU/SEI-90-TR-21 ESD-90-TR-222, 1990
- [19] Knime, <https://www.knime.com/>
- [20] LNI 4.0 Testbed Edge Configuration – Usage View, https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/LNI4.0-Testbed-Edge-Configuration_UsageViewEN.pdf
- [21] One2M, <https://www.onem2m.org/>
- [22] OPC foundation, <https://opcfoundation.org>
- [23] Open Neural Network Exchange, <https://onnx.ai/>

- [24] Pandas, <https://pandas.pydata.org/>
- [25] Plattform Industrie 4.0, <https://www.plattform-i40.de>
- [26] Plattform Industrie 4.0, Die Verwaltungsschale im Detail, 2019, <https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/verwaltungsschale-im-detail-pr%C3%A4sentation.html>
- [27] Plattform Industrie 4.0, Industrie 4.0 Sprache, <https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/hm-2018-sprache.html>
- [28] Plattform Industrie 4.0, Wegweiser KI In der Industrie 4.0: Orientierung, Anwendungsbeispiele, Handlungsempfehlungen, 2020, <https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/ki-in-der-industrie-4-0-orientierung-anwendungsbeispiele-handlungsempfehlungen.html>
- [30] Rapidminer, <https://rapidminer.com/>
- [31] Reference Architecture Model Industrie 4.0, <https://www.plattform-i40.de/PI40/Redaktion/EN/Downloads/Publikation/rami40-an-introduction.html>
- [32] C. Rupp und die Sophisten, Requirements-Engineering und Management – Professionelle, iterative Anforderungsanalyse für die Praxis, 4. Auflage, 2006
- [33] C. Sauer, H. Eichelberger, A. Ahmadian, A. Dewes, J. Jürjens, Aktuelle Industrie 4.0 Plattformen – Eine Übersicht, IIP-Ecosphere Whitepaper IIP-2020/001, 2020
- [34] K. Schmid and I. John, A Customizable Approach To Full-Life Cycle Variability Management, Science of Computer Programming 53(3), 259-284, 2004
- [35] R. Shokri, M. Stronati, C. Song, V. Shmatikov, Membership Inference Attacks against Machine Learning Models, IEEE Symposium on Security and Privacy, 2017
- [36] H. Stichweh, C. Sauer, H. Eichelberger, IIP-Ecosphere Plattform Requirements (Usage View), Version 1.0, Januar 2021, IIP-2021/001
- [37] VDI Statusreport, Simulation und digitaler Zwilling im Anlagenlebenszyklus – Standpunkte und Thesen, Februar 2020
- [38] VDI/VDE 2193 Blatt 1/2, Sprache für I4.0-Komponenten, <https://www.vdi.de/richtlinien/details/vdivde-2193-blatt-1-sprache-fuer-i40-komponenten-struktur-von-nachrichten>, <https://www.vdi.de/richtlinien/details/vdivde-2193-blatt-2-sprache-fuer-i40-komponenten-interaktionsprotokoll-fuer-ausschreibungsverfahren>
- [39] Wikipedia, Anonymisierung und Pseudonymisierung, https://de.wikipedia.org/wiki/Anonymisierung_und_Pseudonymisierung
- [40] ZVEI, Beispiele zur Verwaltungsschale der Industrie 4.0-Komponente – Basisteil, 2016, https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2016/November/Bispiele_zur_Verwaltungsschale_der_Industrie_4.0-Komponente_-_Basisteil/Beispiele-Verwaltungsschale-Industrie-40-Komponente-White-Paper-Final.pdf
- [41] VWS vernetzt, <https://vwsvernetzt.de/>
- [42] R. Wöstmann, P. Schlunder, F. Temme, R. Klinkenberg, J. Kimberger, A. Spichtinger, M. Goldhacker, J. Deuse, Conception of a Reference Architecture for Machine Learning in the Process Industry, in Intl. Conference on Big Data, 2020.

Über die Autoren



Dr. Holger Eichelberger ist stellvertretender Leiter der Arbeitsgruppe Software Systems Engineering am Institut für Informatik der Universität Hildesheim. Er forscht auf den Gebieten Software-Produktlinien, modellbasiertes Engineering, Performanz-Monitoring und Performanz-Analysen. Insbesondere interessiert er sich für die Integrationen dieser Gebiete um dadurch adaptive Software-Systeme zu erstellen. In IIP-Ecosphere leitet er den Think Tank „Plattformen“ sowie den KI-Accelerator. Er studierte Informatik an der Universität Würzburg und promovierte dort über das automatische Layout von UML-Diagrammen.

Fotograf: Daniel Kunzfeld



Dr. Christian Severin Sauer ist Postdoktorand in der Arbeitsgruppe Software Systems Engineering am Institut für Informatik der Universität Hildesheim. Sein Forschungsinteresse konzentriert sich auf die Erhebung und Modellierung von Domänenwissen für erklärungs-fähige und kontextsensitive KI-Anwendungen. Er studierte an der Universität Hildesheim und promovierte in Computer Science an der University of West London. Während seiner Promotion untersuchte und entwickelte er Verfahren zur Wissenserhebung und Wissensmodellierung für erklärungs-fähige und kontextsensitive KI-Anwendungen.



Dr. Amir Shayan Ahmadian ist Postdoktorand an der Fakultät für Informatik der Universität Koblenz-Landau. Sein Forschungsinteresse konzentriert sich auf die Herausforderungen beim Entwurf und der Umsetzung sicherer und datenschutzfreundlicher Softwaresysteme sowie auf die aktuellen Entwicklungen in Industrie 4.0. Er studierte Informatik an der Universität Paderborn und promovierte in Informatik an der Universität Koblenz-Landau. Während seiner Promotion entwickelte er eine Methodik um das Prinzip des „Datenschutz durch Technikgestaltung“ zu operationalisieren.



Michael Schicktanz hat an der Technischen Hochschule Nürnberg: sein Bachelorstudium der Wirtschaftsinformatik und an der Otto-Friedrich-Universität Bamberg: sein Masterstudium der Wirtschaftsinformatik absolviert. Von 2015 bis 2017 war er bei Allianz SE & Allianz Life in den USA im Post-Graduierten- Programm mit Fokus auf Digitalisierung und moderne Software-Technologien. Seit 2017 arbeitet er bei der Siemens AG (Erlangen) und ist dort Projekt Portfolio Manager und Solution Architect.



Dr. Andreas Dewes hat an der Sorbonne Universität Paris und der französischen Kernenergiebehörde (CEA) im experimentellen Quantencomputing promoviert. Er hat mehrere Software-Unternehmen gegründet und ist Geschäftsführer der KIProtect GmbH, die moderne technische Software-Lösungen für Datenschutz und Datensicherheit entwickelt. Im Rahmen von IIP-Ecosphere entwickelt KIProtect GmbH gemeinsam mit den Konsortialprojektpartnern und assoziierten Unternehmen eine Lösung für die sichere und datenschutzkonforme Nutzung von Industrie- & IoT Daten.



Dr. Gregory Palmer ist ein PostDoc am L3S-Forschungszentrum. Er schloss seine Promotion 2019 an der Fakultät für Informatik der Universität Liverpool unter der Leitung von Prof. Karl Tuyls und Prof. Rahul Savani ab. Während seiner gesamten Doktorarbeit entwickelte Gregory eine Reihe von Ansätzen, die es unabhängigen Lernenden ermöglichen, Multi-Agenten-Lernpathologien innerhalb vollständig kooperativer Teamspiele zu überwinden. Darüber hinaus arbeitete er mit der HAL-Allergiegruppe an der Automatisierung der Inspektion undurchsichtiger flüssiger Impfstoffe. In von IIP-Ecosphere ist Gregory für die Leitung des Think Tank Data verantwortlich und arbeitet im Rahmen des Think Tank AI und Produktion an robustem Reinforcement Learning für Industrieumgebungen.



Dr. Claudia Niederée arbeitet als Forschungsgruppenleiterin am L3S-Forschungszentrum in Hannover. Bevor sie zum L3S kam, arbeitete sie bei Fraunhofer. Sie hat ein Diplom in Informatik und wurde an der TU Hamburg-Harburg in Informatik promoviert. Ihre Forschungsschwerpunkte sind KI, Diversität in Information, digitales Vergessen, Social-Media-Analyse und intelligente Systeme u.a. für die Produktion. Sie hat mehr als 100 wissenschaftliche Artikel veröffentlicht.

1 Anhang: Anforderungs-Fragebogen

Dieser Anhang enthält die (sprachlich und bzw. in Formatierung leicht angepasste) Anforderungsanfrage an die IIP-Ecosphere Projektteile, insbesondere die Demonstratoren.

Szenario/AP: _____

Die Think Tanks (TT) und der KI-Accelerator benötigen Ihre Hilfe für die gerade anstehenden Planungsschritte. Hierfür haben wir einige Fragen zusammengestellt, deren Beantwortung uns bei unserer Arbeit sehr helfen würden. Dabei handelt es sich um Fragen, die vielleicht bei Ihnen bereits diskutiert wurden bzw. deren Antwort sogar ihrer Arbeit helfen könnte. Uns ist durchaus bewusst, dass es sich zurzeit hierbei um vorläufige Planungen handelt und sich die Details während der Umsetzung ändern können.

Umgekehrt hoffen sowohl der TT Plattformen als auch der KI-Accelerator, dass die Ideen und später die Verfahren und Technologien, die wir entwickeln, wiederum den Szenarien/Demonstratoren helfen werden, z.B., durch Wiederverwendung von KI-Bausteinen oder Diensten, die Sie nicht entwickeln können/werden. An dieser Stelle ist vielleicht etwas Fantasie gefragt, denn einige Fragen zielen auf ihre (zukünftigen) Wünsche, die wir gerne in unsere Planung mit aufnehmen (und falls möglich auch umsetzen) werden.

Wir bedanken uns für Ihre Unterstützung. Rückfragen bzw. Antworten auf die Fragen bitte einfach an Holger Eichelberger. Natürlich stellen sich der Think Tank „Plattformen“ bzw. der KI-Accelerator auch gerne bei Ihnen vor, auf Absprache, bei einem IIP-Ecosphere Event,...

Kurzbeschreibung des Szenarios, insbesondere mit Übersichtsbild (z.B., Komponenten), ggf. schrittweisem Ablauf

Bitte geben Sie uns eine Kurzbeschreibung des Szenarios/Demonstrators basierend auf dem aktuellen Planungsstand, sodass wir das geplante Ziel und die Einsatzumgebung besser verstehen und einordnen können. Dabei wäre für uns ein Übersichtsbild oder sogar ein schrittweiser Ablauf sehr hilfreich, auch um die Interaktionen (besser) zu verstehen. Ein Übersichtsbild könnte beispielsweise eine Art Entwurfsdiagramm z.B., basierend auf Komponenten oder Services sein.

- ...

Welche Inputs werden (basierend auf Übersichtsbild, z.B., mit Daten, Volumen, Frequenz, Typ/Protokoll) auftreten?

Basierend auf der Kurzbeschreibung bzw. dem Übersichtsbild, welche Eingaben sind im Szenario/Demonstrator angedacht. Dabei wäre es für uns hilfreich zu verstehen, welche Arten von Daten in welcher Geschwindigkeit bzw. Größe auftreten könnten. Uns ist durchaus bewusst, dass es sich hierbei zurzeit um grobe Schätzungen handeln könnte...

- ...

Welche Outputs (basierend auf Übersichtsbild, z.B., mit Daten, Volumen, Frequenz, Typ/Protokoll) werden auftreten?

Basierend auf der Kurzbeschreibung bzw. dem Übersichtsbild, welche Ausgaben sind im Szenario/Demonstrator angedacht. Dabei wäre es für uns hilfreich zu verstehen, welche Arten von Daten in welcher Geschwindigkeit bzw. Größe auftreten könnten. Uns ist durchaus bewusst, dass es sich hierbei zurzeit um grobe Schätzungen handeln könnte...

- ...

Verwendete KI-Verfahren/Komponenten/Frameworks

Ein Ziel der IIP-Ecosphere Plattform und des KI-Accelerators ist es, einen KI-Baukasten (easy-to-use AI) für die intelligente Produktion zu entwickeln, der die Umsetzung zukünftiger KI-Anwendungen erleichtert. Hierzu wäre es für uns bereits jetzt hilfreich, zu verstehen, welche KI-Verfahren bei Ihrem Szenario möglicherweise zum Einsatz kommen. Gibt es für diese KI-Verfahren vielleicht bereits Ideen, ob existierende Komponenten bzw. Frameworks hierbei als Grundlage verwendet werden sollen?

-

Welche IT-Ressourcen (GPU, Cloud, Edge, PLC-Edge, ... mit grober Beschreibung) sind geplant?

Welche IT-Ressourcen (mit Rechenkapazitäten) werden in Ihrem Szenario/Demonstrator wahrscheinlich vorkommen. Die Ressourcen führen beispielsweise KI-Verfahren aus oder speichern Daten. Dabei darf es sich um lokale Ressourcen nahe an der Produktion handeln (Steuergerät mit Edge-Funktionalität, Edge, oder auch Maschine) oder auch um eher zentrale Ressourcen (GPU-Cluster, private/hybride/öffentliche Cloud) handeln. Für die geplanten Ressourcen wäre für uns die Anzahl und eine grobe Beschreibung hilfreich.

- ...

Sind lokale KI-Verfahren/Vorverarbeitung (nahe an der Produktionstechnik) hilfreich oder sind nur zentralisierte Verfahren geplant?

Ein Dienst, den die IIP-Plattform wahrscheinlich anbieten wird, ist es Komponenten auf (nahezu) beliebigen Ressourcen mit Rechenkapazitäten zu verteilen. Dabei versuchen wir sowohl verschiedene Hersteller zu unterstützen, relevante Standards einzusetzen (z.B. die Verwaltungsschale) als auch flexibel zu sein, d.h., KI-Verfahren sollen nur dort eingesetzt werden, wo sie auch Sinn machen. Angenommen, ein derartiger Dienst würde zur Verfügung stehen. Wäre dieser Dienst für Sie (Ihr Szenario/Ihren Demonstrator oder auch allgemein) hilfreich? Falls ja, was würden Sie sich von einem derartigen Dienst wünschen?

-

Welche bereitgestellten Dienste der IIP-Ecosphere Plattform würden helfen, dass KI-Anwendungen schneller, einfacher, besser umzusetzen?

Jetzt ist etwas Fantasie erforderlich... Die IIP-Ecosphere Plattform soll verschiedene hilfreiche Software-Dienste zur Verfügung stellen, die Ihre Anwendung unterstützen, verbessern, sicherer oder auch flexibler machen. Initiale Beispiele sind bereits zum Kickoff angeklungen: Datenintegration, Sicherheit oder Anonymisierung/Pseudonymisierung von Daten. Welche Dienste wären für Sie (im Szenario oder allgemein) sonst noch hilfreich? Welche Funktionalitäten bzw. Eigenschaften sollten derartige Dienste (auch die oben genannten) haben, damit diese für Sie hilfreich werden?

-

Wenn Dienste/Eigenschaften zur Laufzeit veränderbar wären, welche wären für Sie hilfreich/profitabel/wünschenswert?

Und jetzt wird es noch etwas wilder... Die Produktion ändert sich üblicherweise über die Zeit. Hierbei könnte Ihnen die Plattform helfen, wenn sie hinreichend Flexibilität erlauben würde. Ein Beispiel für den oben genannten Dienst, der KI-Komponenten auf die verfügbaren Ressourcen verteilt, könnte sein, dass

die Verteilung zur Laufzeit überprüft und optimiert wird, natürlich möglichst ohne negative Effekte zu verursachen. Generell kann diese Flexibilität durch manuellen Eingriff angefordert werden oder die Plattform könnte sie auch selbstständig auslösen. Wäre das für Sie hilfreich/wünschenswert? Auf welche Eigenschaften oder Dienste würde das aus Ihrer Sicht zutreffen (gerne mit Beschreibung)?

-

Werden personenbezogene Daten verarbeitet? Welche Arten von personenbezogene Daten und in welchem Verarbeitungsprozess werden diese verarbeitet?

Mögliche Hinweise aus der DGSVO um Datenpfade in der Plattform zu planen.

DSGVO: „personenbezogene Daten“ alle Informationen, die sich auf eine identifizierte oder identifizierbare natürliche Person (im Folgenden „betroffene Person“) beziehen...

DSGVO: „Verarbeitung“ jeden mit oder ohne Hilfe automatisierter Verfahren ausgeführten Vorgang oder jede solche Vorgangsreihe im Zusammenhang mit personenbezogenen Daten wie das Erheben, das Erfassen, die Organisation, das Ordnen, die Speicherung, die Anpassung oder Veränderung, das Auslesen, das Abfragen, die Verwendung, die Offenlegung durch Übermittlung, Verbreitung oder eine andere Form der Bereitstellung, den Abgleich oder die Verknüpfung, die Einschränkung, das Löschen oder die Vernichtung.

-

Welche technische oder organisatorische Maßnahmen (wie z. B. Pseudonymisierung) sind vorhanden bzw. denkbar, um Datenschutzgrundsätze (DSGVO Artikel 5 – wie z. B. Datenminimierung) wirksam umzusetzen?

Um Datenpfade sicherer zu gestalten, können (erweiterte, innovative) Mechanismen zum Datenschutz eingesetzt werden. Welche Bedarfe und Möglichkeiten sehen Sie im Rahmen Ihres Demonstrators/Szenarios bzw. Ihrer Arbeit.

....