

Architektur einer adaptiven Plattform für unternehmensübergreifende datenbasierte Dienste mit dem International Data Spaces

Architecture of an adaptive platform for cross-enterprise data-based services with the International Data Spaces

Christian Olms^a
Christian Nissen^a
Arkadius Schier^a
Jens Leveling^a
Rebecca Rademacher^a
Michael ten Hompe^{a,b}

^aFraunhofer Institut für Materialfluss und Logistik IML, Joseph-von-Fraunhofer-Str. 2-4, 44227 Dortmund, Deutschland

^bLehrstuhl für Förder- und Lagerwesen der Technischen Universität Dortmund, Joseph-von-Fraunhofer-Str. 2-4, 44227 Dortmund, Deutschland

Die DE4L Plattform soll datenzentrierte Geschäftsmodelle für KEP-Dienstleister ermöglichen. Die Integration des International Data Spaces (IDS) in DE4L ist durch die Herausforderungen diverser Datenerzeugungsgeschwindigkeiten in Internet der Dinge Anwendungen und der Heterogenität von IDS-Konsumenten beeinflusst. Um beide Herausforderungen in einem bestehenden IDS-Konnektor für Sensordaten zu berücksichtigen, erweitert dieser Beitrag diesen IDS-Konnektor um die Verwendung von Nachrichtenverteiltern und Anwendungscontainern. Ergänzt um diese beiden Konzepte kann eine prototypische Erweiterung des IDS-Konnektors auf beide Herausforderungen adaptiv reagieren.

[Schlüsselwörter: International Data Spaces, Internet der Dinge, Softwarearchitektur]

The DE4L platform is designed to enable data-centric business models for CEP service providers. The integration of the International Data Space (IDS) in DE4L is influenced by the challenges of various data generation speeds in Internet of things applications and the heterogeneity of IDS consumers. To address both challenges in an existing IDS connector for sensor data, this article extends this IDS connector to include the use of message distributors and application containers. Complemented by these two concepts, a prototypical extension of the IDS connector can respond adaptively to both challenges. [Keywords: International Data Spaces, Internet of things, Software Architecture]

1 EINFÜHRUNG

Aufgrund der massiv steigenden Bestell-, Liefer- und Retouren aufkommen im B2C-Bereich steigt auch der Bedarf an Kurier-, Express- und Paketdienstleistungen (KEP). Zustellfahrzeuge machen bereits bis zu 30% des innerstädtischen Verkehrs aus und verursachen dabei ca. 80% der Staus. Die Digitalisierung der „letzten Meile“, die zwischen 13% und 75% der gesamten Kosten einer Lieferkette verursacht [Gev09, S.4], bietet dabei enormes Potential an Effizienzsteigerungen.

Die sogenannte „letzte Meile“ der Lieferkette umfasst die adressatengenaue Zustellung und Abholung von Paketen an der Haustür des Empfängers. Das Ziel besteht darin, das hohe Verkehrsaufkommen, Kosten durch Falschzustellungen, Einlagerungen sowie wiederholte Zustell-/Abholversuche zu vermeiden. Neue Kundenanforderungen wie z. B. die Zustellung temperatursensibler Waren oder kurzfristige Änderungen des Zustellorts erfordern eine verstärkte Zusammenarbeit der Dienstleister.

1.1 MOTIVATION

In diesem Zusammenhang ist umfangreiches Wissen über Geschäftspartner, wie korrekte Adressdaten sowie Liefer- und Abholorte, zwingend notwendig, um Güter sicher zu ihrem Ziel transportieren zu können. Ein wesentlicher Teil des Wissens liegt jedoch meist nicht digital vor, z. B. über besondere Zustellorte, Anwesenheitszeiten, Abstellgenehmigungen, hilfsbereite Nachbarn, Öffnungszeiten oder auch Ansprechpartner und Wege für große Firmengebäuden und Grundstücke. KEP-Dienstleister versuchen mit Hilfe von gepflegten Adressverzeichnissen

unvollständige Adressen aufzulösen und Namen ggf. zu korrigieren. Aufgrund des zusätzlich nötigen Erfahrungswissens ist dies häufig unzureichend und ein Wechsel der Zustellmitarbeiter (z. B. bei Krankheit und Urlaub) oder des Dienstleisters meist problematisch, da es viel Zeit braucht, das nötige Geschäftspartner- und Erfahrungswissen aufzubauen.

1.2 AUSGANGSSITUATION

Innerhalb des Vorhabens „Data Economy 4 Advanced Logistics (DE4L)“ des Förderprogramms „Smarte Datenwirtschaft“ soll eine Handelsplattform für Daten und datenbasierte Dienstleistungen mit garantierter Datensouveränität, Rechtssicherheit und Datenschutz entwickelt werden [DE4L19].

Teilnehmer des Systems sammeln durch Internet der Dinge Anwendungen während der normalen Auslieferung von Paket- und Briefsendungen über mitgeführte Sensoren an Postboten, Fahrradkurieren, Drohnen und Paketlieferautos verschiedenste Daten wie z. B. Koordinaten, Lärm-, Straßenverkehrs-, Umwelt-, Erschütterungs-, Bild- und 3D-Videodaten. Zusätzlich können Teilnehmer externe Datendienste anbieten, wie zum Beispiel die Auflösung unvollständiger Adressen.

Durch Internet der Dinge Anwendungen gesammelte Daten stellen einen elementaren Bestandteil für Datendienste innerhalb der DE4L-Plattform dar. Um diese Daten nutzbar zu machen, ist eine Überführung von technologie-spezifischen Protokollen und Formaten in offene Protokolle (z.B. MQTT) und Datenformate (z.B. JSON) und eine sichere und vertrauenswürdige Integration dieser Datenquellen in die DE4L-Plattform erforderlich [DE4L19, S.3]. Eine solche Integration von Internet der Dinge Datenquellen ist von der Herausforderung diverser Datenerzeugungsgeschwindigkeiten und Datenerzeugungsmustern geprägt [Rap19, S. 97058-97059].

1.3 NEUHEIT DES LÖSUNGSANSATZES

Der IDS ist ein Ansatz zur Übermittlung und Integration von Daten unterschiedlicher Unternehmen mit Berücksichtigung der Datensouveränität und kann somit als Grundlage für Datendienste innerhalb der DE4L-Plattform dienen. Dieser Beitrag beschreibt ein Konzept für die Verwendung des IDS in der DE4L-Plattform, welches diese beiden Herausforderungen berücksichtigt. Dazu wird eine bestehende Integrationsplattform für Internet der Dinge Daten in den IDS aufgegriffen.

1.4 GLIEDERUNG

Die weitere Struktur des Beitrags ergibt sich wie folgt: Kapitel 2 beschreibt den Stand der Technik. Kapitel 3 beschreibt die Anforderungen an einen IDS Konnektor, der beide Herausforderungen berücksichtigt. Kapitel 4 disku-

tiert Konzepte zur Lösung dieser Anforderungen und Kapitel 5 zeigt die Umsetzbarkeit dieser Konzepte in einer prototypischen Implementierung. Kapitel 6 fasst diesen Beitrag zusammen. Abschließend gibt Kapitel 7 einen Ausblick auf anschließende Arbeiten.

2 STAND DER WISSENSCHAFT UND TECHNIK

2.1 INTERNATIONAL DATA SPACES

Die Initiative International Data Spaces (IDS) adressiert den Entwurf eines Referenzarchitekturmodells und zugehörige Referenzimplementierungen für industrielle Datenräume (engl. Data Space).

Der IDS stellt eine Verbindung zwischen Kommunikations- und Basic-Data-Service-Architekturen und abstrakteren Smart-Data-Service-Architekturen dar. Diese bilden die Grundlage für die Entwicklung und Bereitstellung neuer datenbasierter Services und Produkte in unterschiedlichsten Bereichen, während die dafür nötigen Daten im Bereich der Basic-Data-Services auf Basis des Internet der Dinge und den darin eingebundenen Sensorsystemen gesammelt werden.

Ziel des IDS ist die Entwicklung eines dezentralen „Ökosystems von Daten“, welches einen kontrollierten Austausch und Nutzung von Daten jeglicher Art zwischen Organisationen ermöglicht (eng. Ecosystem of Data). Der Datenaustausch soll über Datenmärkte, welche über dieses System zugänglich sind, realisiert werden (engl. Data Markets). Außerdem werden über diese Märkte datenbasierte Services und Geschäftsmodelle angeboten.

Eines der Grundprinzipien des IDS ist der Erhalt der Souveränität über die eigenen Daten. Dieses Prinzip schließt die Übertragung von Eigentumsrechten an zentrale Instanzen oder Anbieter aus (engl. Security and Data Sovereignty). Die Sicherung der Datenhoheit wird durch vom Eigentümer festgelegte Nutzungsbeschränkungen gewährleistet, welche vor der Nutzung vollständig akzeptiert werden müssen. Um einem Missbrauch der Daten zuvorzukommen, erhalten potenzielle Teilnehmer erst im Anschluss an eine Evaluation und Zertifizierung Zugriff zum IDS (engl. Trust).

Ein standardisierter Zugang soll für neue Beteiligte durch IDS-Konnektoren erfolgen (engl. Standardized Interoperability). Über diese werden zudem neue Anwendungen eingespeist (engl. Value adding Apps). Die Funktionalität und Sicherheit dieser IDS-Konnektoren fußt auf den drei Themenfeldern Vertrauensanker, vertrauenswürdige Plattform und Datennutzungskontrolle. [IDS19, S. 9 10].

Die IDS-Referenzarchitektur umfasst fünf Ebenen und beschreibt diese aus den drei Perspektiven Sicherheit, Zer-

tifizierung und Steuerung (siehe Abbildung 1). In der Geschäfts-Ebene werden dazu die Rollen aller Beteiligten und Interaktionen zwischen diesen beschrieben. Entscheidend ist die Definition der Nutzungsvereinbarung für einen gesicherten Datenaustausch zwischen Endpunkten im IDS [IDS19, S.21]. Die festgehaltenen Interaktionen dienen unter anderem dazu, die oben genannten Anforderungen in der Funktions-Ebene zu konkretisieren [IDS19, S. 29]. Innerhalb der Prozess-Ebene werden diese Interaktionen nun in den drei Prozessen der Integration, des Datenaustauschs und der Veröffentlichung bzw. Nutzung von Daten in Bezug auf den Zugriff unterschiedlicher Systemkomponenten beschrieben [IDS19, S. 33]. Auf Informations-Ebene werden wiederum die Informationen, die während dieser Prozesse entstehen, zusammengefasst, beschrieben und Forderungen an den Umgang mit diesen Informationen in Bezug auf die in der Geschäfts-Ebene festgelegten Rollen aufzeigt [IDS19, S. 40].

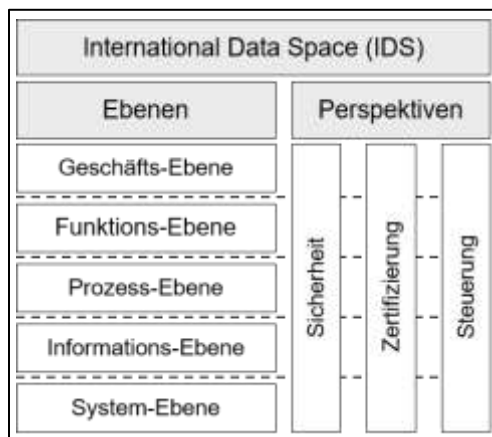


Abbildung 1. Struktur der IDS-Referenzarchitektur [IDS19, S.11]

Abschließend werden auf System-Ebene die zur Realisierung nötigen technischen Komponenten beschrieben und in Zusammenhang zu den vorher definierten Prozessen gestellt. Dabei sind der IDS-Konnektor, der Broker und der App-Store als die drei Hauptkomponenten des IDS identifiziert worden. Der für den Datenaustausch zuständige IDS-Konnektor stellt Metadaten an den Broker zur Verfügung und stellt außerdem die Verbindung zum App-Store dar, um neue über den IDS zur Verfügung gestellte Anwendungen zu beanspruchen. Die konkrete Implementierung der Systemkomponenten (insb. des IDS-Konnektors) sind abhängig von der Funktionalität, die dieser je nach Anwendungsfall erfüllen muss. [IDS19, S. 61]

Insgesamt stellt der IDS eine technische Infrastruktur für den Austausch von beliebigen Daten zur Verfügung und hat keinen direkten fachlichen Bezug zum Anwendungsfeld Logistik.

2.2 NACHRICHTENGETRIEBENE INTEGRATION VON SERVICES

Nachrichtengetriebene Integration bezeichnet die Kommunikation zwischen Services durch eine Vielzahl von kleinen, in hoher Frequenz erzeugten und zugestellten Datenpaketen, die durch ein Nachrichtensystem in zuverlässiger Weise vermittelt werden. Diese Art des Datenaustauschs ermöglicht eine asynchrone und lose gekoppelte Integration von Services. [Hoh04, S.52-56]

Nachrichtenkanäle bezeichnen Slots, in denen Services Nachrichten einstellen können (Nachrichtenproduzenten) und aus dem Services Nachrichten entnehmen können (Nachrichtenkonsumenten). Nachrichtenkanäle können unterschiedliche Semantiken haben. Bei einem Kanal mit *Point-to-Point* Kommunikation wird eine eingestellte Nachricht genau einem Nachrichtenkonsumenten des Nachrichtenkanals ausgeliefert [Hoh04, S. 103-105]. Im Gegensatz dazu wird bei der *Publish-Subscribe*-Kommunikation an ein *Topic* gesendet, welche von allen Konsumenten abonniert muss um die Nachricht zu erhalten [Hoh04, S. 106-110].

2.3 KONZEPTE ADAPTIVER SYSTEME

Mendonça et al. definieren als ein selbstadaptives System ein System, welches sein Verhalten an Umgebungsbedingungen unter unklaren Ausführungsbedingungen anpasst, um Qualitätsanforderungen zu bewahren oder zu verbessern [Men19, S. 1]. Dazu stellen Sie fest, dass Kontrollschleifen aus den Aufgaben Überwachung, Analyse, Planung und Ausführung bestehen und Kontrollschleifen essentielle Bestandteile von selbstadaptiven Systemen sind [Men19, S.4].

2.4 INTERNET DER DINGE REFERENZARCHITEKTUR

Raptis et al. beschreiben drei Charakteristiken von Internet der Dinge Anwendungen [Rap19]:

- Datenpräsenz: Daten können entweder von spezifisch definierten und lokalisierten Quellen oder von allgegenwärtigen Datengeneratoren erfasst werden.
- Datenkoordination: Die Koordination der (Industrie-)Prozesse kann entweder zentralisiert oder hierarchisch erfolgen.
- Datenverarbeitung: Die Verarbeitung von Daten kann entweder zentralisiert oder verteilt erfolgen.

Raptis et al. identifizieren außerdem für die Klasse von Anwendungen, welche durch allgegenwärtige Datenpräsenz, hierarchische Koordination sowie zentralisierte Datenverarbeitung charakterisiert werden, eine Referenzarchitektur [Rap19, IV.B, S. 97061ff].

Diese Referenzarchitektur setzt sich aus drei Schichten zusammen [Rap19, S. 97064]:

- Die Feldschicht enthält allgegenwärtige und verteilte Geräte die über unterschiedliche Technologien angesprochen werden.
- Die Zugangsschicht vermittelt den Datenaustausch zwischen der Feldschicht und der Kontrollschicht und stellt eine lokale hierarchische Koordination und Verwaltung von verschiedenen Geräten in der Feldschicht bereit.
- Die Kontrollschicht ist für die zentrale Steuerung und Verarbeitung der Daten aus verschiedenen Internet der Dinge Systemen in der Feldschicht verantwortlich.

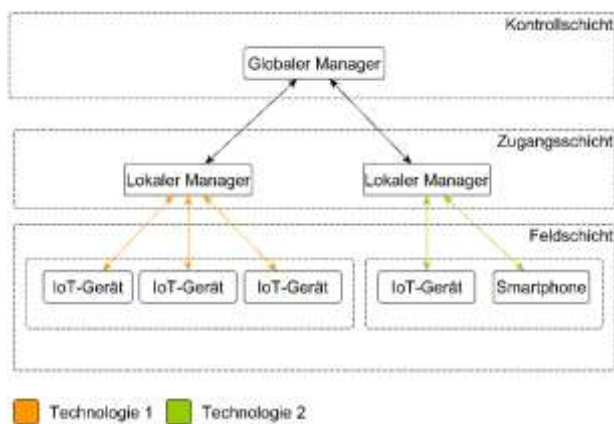


Abbildung 2. Referenzarchitektur für industrielle Internet der Dinge Anwendungen [Rap19, Abb. 6]

2.5 BESTEHENDE VORARBEITEN

Innerhalb vorangegangener Ausarbeitungen wurde festgestellt, dass insbesondere die allgegenwärtige Datenpräsenz eine häufige Eigenschaft in Industrie 4.0 Anwendungen in der Logistik darstellt und Internet der Dinge Anwendungen in der Logistik durch eine heterogene Technologievielfalt geprägt sind. Aus diesen Randbedingungen wurde eine Sub-Architektur für Systeme der Zugangsschicht mit einem Fokus auf einer ereignisgetriebenen Verarbeitung und Integration von Messwerten abgeleitet [Lev18].

Grundidee dieser Architektur ist, die Funktionalität der Zugangsschicht über lose gekoppelte Services bereit zu stellen, wobei jeder Service jeweils genau eine Funktionalität bereitstellt. Ein weiterer Aspekt ist, dass keine zentrale Koordination der Services existiert, sondern die Services über Kanäle mit *Publish-Subscribe* Kommunikation eines Nachrichtensystems miteinander kommunizieren. Ein weiteres Kernkonzept ist, dass keine zentrale Steuerung des Ablaufs vorgegeben ist. Stattdessen wird der Ablauf implizit durch die Kommunikation der Services durch die Kanäle mit *Publish-Subscribe* Kommunikation gesteuert.

Technologiespezifische Details eines Systems in der Feldschicht werden in einem Device-Adapter-Service gekapselt. Die Daten der Systeme werden durch Adapter bereitgestellt. Durch Basis-Services können zusätzliche Funktionen, wie zum Beispiel Umwandlungen von Datenformaten, realisiert werden.

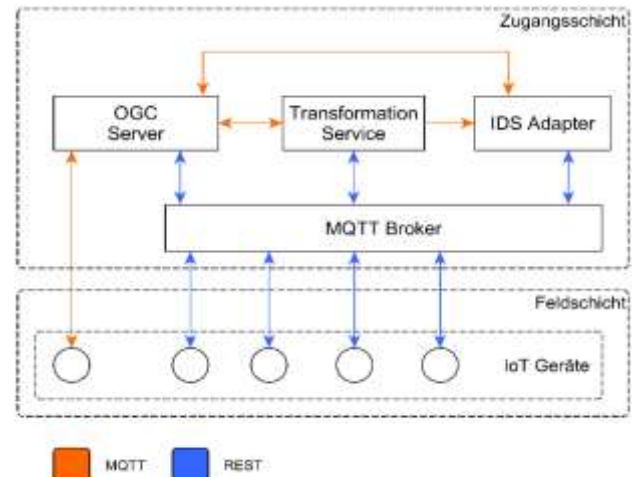


Abbildung 3. Architektur des IDS-Konnektor für Sensordaten nach [Nast20]

In einer weiteren Vorarbeit wurde bereits ein IDS-Konnektor für Sensordaten entwickelt. Dieser Ansatz umfasst vier Komponenten, einen OGC-API-Server (Device-Adapter), einen Transformation-Service (ein Basis-Service), einen IDS-Adapter (für die Kommunikation mit genau einem IDS-Konsumenten) und einen MQTT-Broker als Nachrichtensystem (siehe Abbildung 3) [Nast20]. Die implizite Koordination der Verarbeitung von neuen Messwerten erfolgt über die Registrierung der einzelnen Services an verschiedenen Kanälen des Nachrichtensystems. Der Transformation-Service ist ein optionaler Basis-Service, der die Daten aus dem OGC-Datenformat in das semantische Datenmodell für Sensordaten im IDS-Informationsmodell überführt. Alternativ können die Daten auch im OGC-Format verbleiben. Der IDS-Adapter stellt die Daten aus dem OGC-Server für IDS-Datenkonsumenten bereit. Dazu instanziiert der IDS-Konnektor für jeden IDS-Konsumenten eine separate IDS-Adapter Instanz [Nast20, S.3]. Dadurch wird die Kommunikation zwischen dem IDS-Konnektor und einem IDS-Konsumenten komplett separiert und ist für andere IDS-Konsumenten nicht sichtbar.

2.6 ANWENDUNGSCONTAINER

Weiterhin wurde in einer Vorarbeit ein Anwendungscontainer als ein Laufzeitcontainer definiert, der eine Anwendung mit allen benötigten Abhängigkeiten enthält und somit unabhängig vom ausführenden Host ist. Weiterhin sind die jeweiligen Anwendungen in Anwendungscontainern von anderen Funktionen und Daten isoliert [Lev18, S.3]. Eine bekannte Laufzeitumgebung für Anwendungscontainer ist die Docker Engine. Die Docker-Engine bietet

die Funktionalität, über eine Schnittstelle neue Container zu erzeugen und bestehende Container zu verwalten [Docker20].

3 ANFORDERUNGEN

Dieses Kapitel beschreibt die Anforderungen an den IDS-Konnektor für Sensordaten aus den zuvor beschriebenen Vorarbeiten (vergleiche [Nast20]). Für Basis-Services innerhalb eines IDS-Konnektor, die sich an die unterschiedliche Datenerzeugungsgeschwindigkeiten anpassen, ergeben sich folgende Anforderungen:

- A1: Der IDS-Konnektor muss Nachrichten entsprechend eines Kriteriums aus einem Eingangskanal auf die verfügbaren Instanzen des Basis-Services verteilen. Die Kriterien dürfen nicht eingeschränkt werden.
- A2: Der IDS-Konnektor muss dynamisch Instanzen des Basis-Services skalieren (d.h. dynamisch erzeugen und beenden können).

Damit der IDS-Konnektor adaptiv an unterschiedliche IDS-Konsumenten angepasst werden kann, muss der IDS-Konnektor zudem folgende Anforderung erfüllen:

- A3: Der IDS-Konnektor muss für jeden IDS-Konsumenten einen IDS-Adapter erstellen. Damit muss die gesamte Kommunikation mit einem Konsumenten isoliert werden.

4 KONZEPTE ZUR UMSETZUNG DER ANFORDERUNGEN

Dieses Kapitel stellt relevante Konzepte zur Realisierung der Anforderungen aus Kapitel 3 vor.

4.1 VERTEILUNG VON DATEN AUF SERVICE INSTANZEN

Laut Hohpe und Woolf [Hoh04, S. 502ff] gibt es drei grundlegende Ansätze, um Nachrichten eines Kanals in einem Nachrichtensystem auf mehrere konkurrierenden Instanzen eines Nachrichtenkonsumenten zu verteilen (im Folgenden Verarbeitungskonsument genannt):

- Der Ansatz von konkurrierenden Konsumenten (engl. „competing consumers“) basiert auf der Registrierung aller Instanzen des Verarbeitungskonsumenten für den Kanal beim Nachrichtensystem. Das Nachrichtensystem stellt eine neue Nachricht an genau eine Instanz zu. In diesem Fall hängt die Verteilung der Nachrichten von der impliziten Implementierung im Nachrichtensystem ab. Dieser Ansatz funktioniert nur für Nachrichtenkanäle mit *Point-to-Point* Kommunikation. [Hoh04, 502ff]

- Der Ansatz des Nachrichtenverteilers (engl. „message dispatcher“) beruht auf der Idee, dass ein zusätzlicher Konsument (Nachrichtenverteiler-Service) alle Nachrichten des Kanals zugestellt bekommt. Dieser Konsument verteilt neue Nachrichten anhand einer Verteilungsstrategie an Instanzen des Verarbeitungskonsumenten. Der Nachrichtenverteiler kann über ein beliebiges Protokoll Nachrichten an Instanzen des Verarbeitungskonsumenten weiterleiten. [Hoh04, S.508ff]
- Beim Ansatz des selektiven Konsumenten (engl. selective consumer) wird eine Nachricht anhand eines Selektionskriteriums an einen Konsumenten weitergeleitet. Dazu stellt der Nachrichtenproduzent Selektionskriterien zur Verfügung. Anhand dieser Selektionskriterien können entweder das Nachrichtensystem oder die registrierten Konsumenten auswählen, welcher Konsument eine Nachricht verarbeitet. Für diesen Ansatz ist entweder eine Unterstützung durch das Nachrichtensystem oder eine explizite bzw. implizite Koordination der Konsumenten nötig. [Hoh04, 515ff]

Für eine Nachrichtenverteilung an Instanzen eines Basis-Services wurde der Ansatz des Nachrichtenverteilers ausgewählt. Dieser Ansatz benötigt keine Unterstützung durch das Nachrichtensystem und die Koordination und die Verteilungsstrategie ist explizit im Nachrichtenverteiler-Service definiert. Die Auswahl des Protokolls und der Verteilungsstrategie ist vom Anwendungsfall abhängig.

Der Nachrichtenverteiler-Service bekommt neue Nachrichten vom Eingangskanal zugestellt und leitet diese entsprechend des Verteilungskriterium an genau eine Instanz des Basis-Services weiter. Alle Basis-Service Instanzen stellen ihr Ergebnis in Form einer Nachricht im Nachrichtentopic zur Verfügung. Ein Nachrichtenverteiler-Service arbeitet blockierungsfrei und realisiert nur die Verteilung der Nachrichten [Hoh04, S. 511]. Die Services des IDS-Konnektors kommunizieren mit anderen Basis-Services über Nachrichten-topics, ohne mit einer expliziten Instanz des Basis-Service zu interagieren.

4.2 DYNAMISCHE SERVICE-SKALIERUNG

Die Erstellung eines Anwendungscontainers und die Verwendung einer geeigneten Container-Laufzeitumgebung (siehe Abschnitt 2.6) ermöglicht technologisch die dynamische Skalierung von Basis-Service Instanzen. Die Aufgaben der Überwachung, Analyse und Planung [vergleiche Men19, S.4] über den Lebenszyklus von Basis-Service Instanzen wird dabei ebenfalls im Nachrichtenverteiler-Service anhand eines Skalierungskriterium getroffen [Hoh04, S. 510]. Wenn der Nachrichtenverteiler-Service die Planung der benötigten Instanzen abgeschlossen hat, beauftragt der Service die Container-Laufzeitumgebung,

die benötigten Instanzen des Basis-Services zu starten oder zu beenden.

4.3 VERWALTUNG DIVERSEER IDS-KONSUMENTEN

Unter der Annahme, dass IDS-Konnektoren über ein nachrichtenbasiertes Netzwerkprotokoll miteinander kommunizieren, ist das Nachrichtenverteiler Konzept zur Verteilung von Netzwerknachrichten an unterschiedliche IDS-Konsumenten ebenfalls nutzbar. Dabei muss für das verwendete Protokoll ein Nachrichtenverteiler-Service verfügbar sein, der eine dynamische Anpassung seiner Konfiguration unterstützt und eine Analyse von nicht zustellbaren Protokollnachrichten an einen individuell konfigurierbaren Service ermöglicht.

5 PROTOTYPISCHE ERWEITERUNG DES IDS-KONNEKTORS FÜR SENSORDATEN

Dieses Kapitel beschreibt eine prototypische Erweiterung des vorgestellten IDS-Konnektor für Sensordaten mit den Konzepten aus dem vorherigen Kapitel (vergleiche Abbildung 4). Der Prototyp verwendet dabei einen Geräte-Simulator und einen angepassten Message-Adapter anstatt des OGC-Servers.

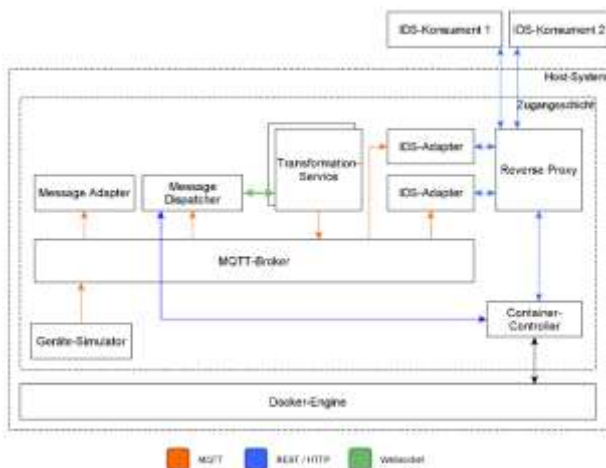


Abbildung 4. Aufbau der prototypischen Umsetzung (eigene Darstellung)

Der Message-Adapter konvertiert die (simulierten) Gerätedaten aus dem herstellereigenen Format (in diesem Fall Protocol-Buffer Daten) in eine textuelle Repräsentation und sendet diese Repräsentation an den MQTT-Broker. Der Transformation-Service konvertiert die simulierten Gerätedaten in das semantische Datenmodell für Sensordaten des IDS. Die IDS-Adapter Instanzen empfangen diese transformierten Nachrichten und stellen sie für einen Abruf durch IDS-Konsumenten bereit.

Der Message-Dispatcher-Service implementiert den im vorherigen Kapitel vorgestellten Nachrichtenverteiler-

Service und leitet über das Websocket-Protokoll die Gerätedaten an Instanzen des Transformation-Service weiter.

Der Container-Controller kapselt die Schnittstelle zur Ausführungsumgebung, in diesem Fall die Docker-Engine [Docker20], und ermöglicht das Erzeugen und Löschen weiterer Instanzen des Transformation-Service und der IDS-Adapter. Die IDS-Konnektoren kommunizieren innerhalb des Prototyps über HTTP(S). Der Reverse-Proxy ist eine HTTP(S)-spezifische Implementierung des Nachrichtenverteilers, der die Bedingungen aus Anforderung 3 erfüllt. Alle Services werden durch die Docker Engine ausgeführt und sind mit Java oder Python implementiert.

5.1 VERTEILUNG VON GERÄTEDATEN AUF TRANSFORMATION-SERVICE INSTANZEN

Alle Transformation-Service Instanzen registrieren sich nach ihrem Start beim Message-Dispatcher-Service. Der Message-Dispatcher-Service puffert neue Nachrichten und verteilt die gepufferten Nachrichten gleichmäßig unter den registrierten Transformation-Service Instanzen.

5.2 AUTOMATISCHE SKALIERUNG DER TRANSFORMATION-SERVICE INSTANZEN

In Abhängigkeit von der Anzahl der gepufferten Nachrichten entscheidet der Message-Dispatcher-Service, ob eine weitere Instanz des Transformation-Service notwendig ist, und beauftragt den Container-Controller die weiteren Instanzen zu erzeugen.

Alle 500ms überprüft der Message-Dispatcher-Service die Anzahl der gepufferten Nachrichten. Wenn mehr als $1000 \cdot 2^{N-1}$ Nachrichten gepuffert werden, erzeugt der Message-Dispatcher-Service eine zusätzliche Transformation-Service Instanz, wobei N die Anzahl der Transformation-Service Instanzen darstellt. Wenn der Message-Dispatcher-Service keine Nachrichten puffert, beendet der Message-Dispatcher-Service eine Transformation-Service Instanz, wobei immer mindestens eine Transformation-Service Instanz vorhanden ist.

5.3 VERWALTUNG DIVERSEER IDS-KONSUMENTEN

IDS-Konsumenten kommunizieren über HTTPS mit dem Prototyp. Dabei werden alle Anfragen von einem Reverse-Proxy entgegengenommen und an den zugehörigen IDS-Adapter weitergeleitet. Der Reverse-Proxy wird in dem vorliegenden Beitrag durch das Tool Traefik realisiert [Traefik20]. Traefik bekommt diese Zuordnungsinformationen über Metadaten der Docker-Engine. Weiterhin leitet Traefik alle nicht zugeordneten HTTP-Anfragen an den Container-Controller weiter. Der Container-Controller analysiert die weitergeleiteten Anfragen und erzeugt für bisher unbekannte IDS-Konsumenten die entsprechenden IDS-Adapter Instanzen mit den passenden Meta-Daten für die Traefik Konfiguration.

6 ZUSAMMENFASSUNG

Die DE4L Plattform soll datenzentrierte Geschäftsmodelle für seine Teilnehmer ermöglichen. Die Integration des International Data Spaces in DE4L ist durch die Herausforderungen der diversen Datenerzeugungsgeschwindigkeiten bei Internet der Dinge Anwendungen und der heterogenen Landschaft von IDS-Konsumenten beeinflusst.

Der vorliegende Beitrag erweitert den IDS-Konnektor durch eine Integration der Konzepte des Datenverteilers und der Anwendungscontainer. Die Umsetzbarkeit dieser Konzepte wird durch die prototypische Erweiterung des bestehenden IDS-Konnektors aufgezeigt. Der Prototyp skaliert dynamisch die Anzahl der Instanzen des Transformation-Service in Abhängigkeit von der Datenerzeugungsgeschwindigkeit. Anfragen verschiedener IDS-Konsumenten werden innerhalb des Prototyps von unterschiedlichen, zustandsbehafteten IDS-Adapter Instanzen verarbeitet.

7 AUSBLICK

Der nächste Schritt in der Entwicklung des DE4L Ökosystems ist es, einen Aushandlungsprozess für eine Bezahlung der Benutzung von Datenquellen zu ermöglichen. Als Grundlage hierzu ist eine Abbildung der verwendeten Datenquellen in eine Distributed-Ledger-Technologie angedacht. Im International Data Space ist konzeptionell eine Ausführungsplattform für Datenauswertungsanwendungen in IDS Konnektoren geplant. Der in diesem Beitrag vorgestellte IDS-Konnektor wird in Forschungs- und Industrieprojekten in eine industriereife Softwarelösung überführen.

LITERATUR

- [DE4L19] Fraunhofer Institut für Materialfluss und Logistik (IML): *Vorhabenbeschreibung – Data Economy 4 Advanced Logistics – Teilvorhaben: Unternehmensübergreifende datenbasierte Dienste*. Version 1.0, Dortmund, 13.03.2019.
- [Docker20] Docker Engine API v1.40 Reference by Docker Inc. URL: <https://docs.docker.com/engine/api/v1.40> (Stand: 30.07.2020).
- [Gevaers09] Gevaers, Roel; Van de Voorde, Eddy; Vanelander, Thierry: *Characteristics of innovations in last mile logistics - using best practices, case studies and making the link with green and sustainable logistics*, Association for European Transport and contributors, 2009.
- [HF20] Hyperledger, hosted by The Linux Foundation: Hyperledger Fabric URL: <https://www.hyperledger.org/use/fabric> (Stand: 27.07.2020).
- [Hoh04] Hohpe, Gregor; Woolf, Bobby: *Enterprise Integration Patterns – Designing, Building and deploying message solutions*. Pearson Education, USA, 2004. – ISBN 0-321-20068-3.
- [IDS19] International Data Spaces Association: *Reference Architecture Model*. Version 3.0, Dortmund, 2019, URL: <https://www.internationaldataspaces.org/wp-content/uploads/2019/03/IDS-Reference-Architecture-Model-3.0.pdf> (Stand: 22.07.2020).
- [Lev18] Leveling, Jens; Weickmann, Luise; Nissen, Christian; Kirsch, Christopher: *Event-Driven Architecture for Sensor Data Integration for Logistics Services*. IEEE International Conference on Industrial Engineering and Engineering Management (IEEM), S.536-540, Bangkok, 2018, doi: 10.1109/IEEM.2018.8607460.
- [Men19] Mendonça, C. Nabor; Jamshidi, Pooyan; Garlan, David; Pahl, Claus: *Developing Self-Adaptive Microservice Systems: Challenges and Directions*. IEEE Software, S. 99 ff., 2019 doi: 10.1109/MS.2019.2955937.
- [Nast20] Nast, Michael; Rother, Benjamin; Gólatowski, Frank; Timmermann, Dirk; Leveling, Jens; Olms, Christian; Nissen, Christian: *Work-in-Progress: Towards an International Data Spaces Connector for the Internet of Things*. 16th IEEE International Conference on Factory Communication Systems (WFCS), Porto, Portugal, 2020, S. 1-4, doi: 10.1109/WFCS47810.2020.9114503.
- [Rap19] Raptis, Theofanis P; Passarella, Andrea; Conti, Marco: *Data Management in Industry 4.0: State of the Art and Open Challenges*. IEEE Access, Bd. 7, S. 97052–97093, 2019, doi: 10.1109/ACCESS.2019.2929296.
- [Traefik20] Traefik, The Cloud Native Edge Router. URL: <https://containo.us/traefik/> (Stand: 03.08.2020).