

# Deutschlandweiter I4.0-Demonstrator

## Technisches Konzept und Implementierung

*Dipl.-Ing. Felix Mayer*

*Dipl.-Ing. Dorothea Pantförder*

*Prof. Dr.-Ing. Christian Diedrich*

*Prof. Dr.-Ing. Birgit Vogel-Heuser*

Um den Begriff *Industrie 4.0* (I4.0) ist aktuell ein regelrechter Hype entstanden. Viele Firmen und Institute beschäftigen sich aktuell mit dem Begriff und seiner genauen Auslegung. Daraus resultierend existieren viele verschiedene Vorstellungen darüber, was unter Industrie 4.0 genau zu verstehen ist, wobei diese Vorstellungen des Öfteren vergleichsweise inkompatibel zueinander sind.

Aus diesem Grund haben sich mehrere Institute und Lehrstühle dazu entschieden einen exemplarischen Industrie-4.0-Demonstrator zu konzipieren und anschließend umzusetzen [1]. Der Demonstrator soll aufzeigen, welche Potentiale im Bereich Industrie 4.0 für die Automatisierungstechnik vorhanden sind und wie diese ausgeschöpft werden können. Durch den Demonstrator soll ein Kernthema von Industrie 4.0, nämlich die Kopplung und Vernetzung räumlich getrennter Produktionsanlagen, exemplarisch gezeigt werden. Diese Kopplung soll weitestgehend automatisch geschehen und sowohl dynamisch, wie auch skalierbar sein. Außerdem sollen Szenarien entwickelt werden, die sich kurz, knapp und anschaulich, innerhalb kurzer Zeit präsentieren lassen. Diese Szenarien sollen überzeugen, eine Vorstellung von Industrie 4.0 vermitteln und die Machbarkeit derartiger Vorhaben beweisen.

Basierend auf den Fähigkeiten und Potentialen der am Demonstrator beteiligten Anlagen wurden zunächst fünf unterschiedliche Szenarien entwickelt: Produktion, Diagnose, Qualitätssicherung, Optimierung und Rekonfiguration. Um die Anschaulichkeit des Demonstrators sicherzustellen, wird als erstes Szenario das Produktionsszenario umgesetzt, wobei große Teile der dabei entwickelten und getesteten Bausteine auch bei allen weiteren Szenarien Verwendung finden werden.

Für die Realisierung des Demonstrators sind die Informationsflüsse und ihre Darstellung entscheidend, ein Transport realer Güter, zum Beispiel der Flaschen, zwischen den Anlagen, oder die Herstellung realer Joghurts ist dabei letztlich nicht nötig. Die Schaffung des verteilten Systems und damit die informationstechnische Kopplung der beteiligten Anlagen, ist für die erfolgreiche Realisierung des Demonstrators wichtig. Dazu ist ein einheitlicher Kommunikationsstandard zwischen den Anlagen nötig, damit Daten ausgetauscht werden können und die Funktion des Gesamtsystems ermöglicht wird. Hierfür ist wiederum ein einheitliches Datenmodell nötig. Mit diesem Datenmodell und Technologien wie Datencontainer und Agenten, können dann Daten ausgetauscht werden. Mit dieser flexiblen und einheitlichen Schnittstelle ist der Demonstrator dann einfach auf weitere Szenarien zu erweitern, so dass grundsätzlich Alles möglich ist.

## Architektur

Essentiell für die Produktion innerhalb des Produktionsszenarios ist das Vorhandensein eines Anbieters. Anbieter ist derjenige, dessen Name und Logo auf dem Endprodukt aufgedruckt wird und den der Kunde mit dem Endprodukt verbindet. Jeder Anbieter besitzt einen Produktionsverbund, um das Endprodukt fertigen zu können. An diesem Produktionsverbund sind Zulieferer mit entsprechenden Produktionsanlagen beteiligt. Für die im Produktionsszenario behandelte Joghurtproduktion sind das zum Beispiel die verschiedenen Zulieferer, mit ihren Anlagen für Joghurt, Becher, Deckel und für die Abfüllung. Ein Anbieter kann natürlich auch selbst ebenfalls Teil der Produktionskette sein, zum Beispiel am Ende der Produktionskette als Abfüller.



In dem – in Abbildung 1 ersichtlichen – ersten Beispiel existieren drei Produktionsverbünde (blaue Umrandungen) für Joghurt, Marmelade und Quark. Jeder der Verbünde wird durch einen dazugehörigen Anbieter (blaue Zylinder) dem Endkunden gegenüber vertreten, zum Beispiel durch eine entsprechende Bestellwebsite und dem Namen und Logo auf dem Endprodukt. Die orangenen Diabolos in der Abbildung stellen die verschiedenen Zulieferanlagen für ein Produkt beziehungsweise Anbieter dar. Natürlich gibt es auch weiterhin Anlagen, die nicht Teil eines Produktionsnetzwerks sind (zum Beispiel die Anlagen A00, A15 und A17).

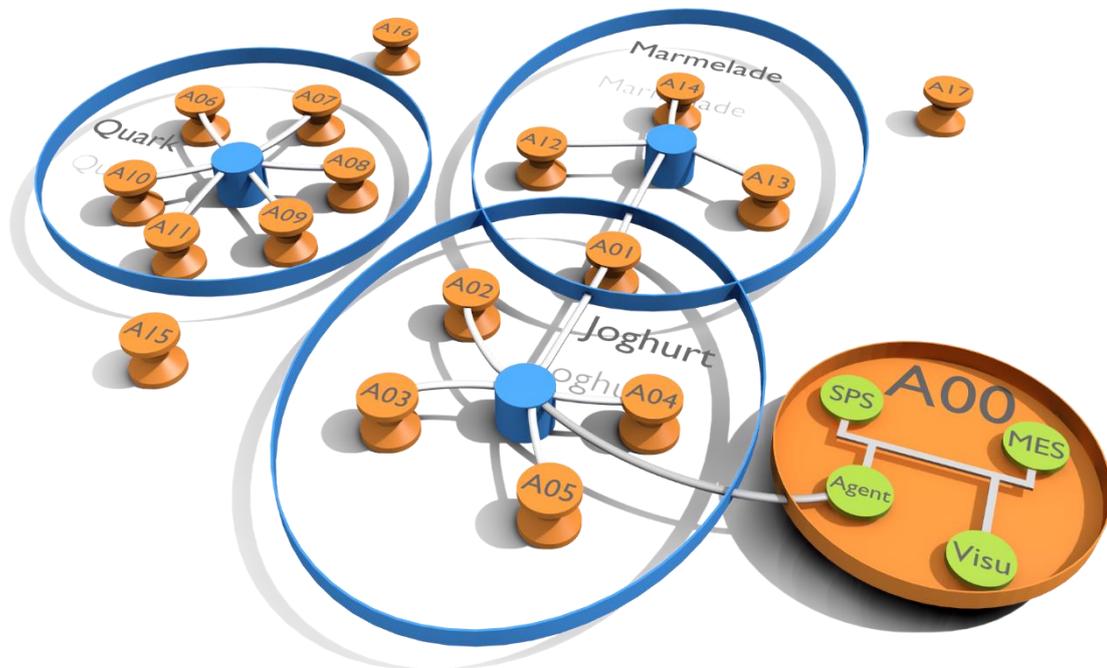


Abbildung 1: Logische Beispielarchitektur

Die in Abbildung 1 sichtbaren Verbindungen zwischen den einzelnen Anlagen und Anbietern stellen die logischen Verbindungen zwischen den Beteiligten dar. Jeder Anbieter hat eine Verbindung zu den für ihn relevanten Zulieferanlagen, welche sich zuvor am Produktionsverbund angemeldet haben. Die logischen Verbindungen sind normalerweise auch reale Netzwerk- beziehungsweise Internetverbindungen, über die zum Beispiel neue Unteraufträge zwischen den einzelnen Anlagen verteilt werden. Eine Anlage kann auch in mehr als einem Produktionsnetzwerk angemeldet sein, wie zum Beispiel die Anlage A01, die sowohl im Joghurtnetzwerk, als auch im Quarknetzwerk beteiligt ist. Die Mitgliedschaft in mehreren Produktionsverbänden ist vor allem in Hinblick auf einen höhere Anlagenauslastung sinnvoll, sofern das hergestellte Produkt auch von mehreren Anbietern angefragt wird. Da die Größe eines Produktionsverbandes nicht von vornherein festgelegt ist, können Anlagen und Unternehmen jederzeit einem Verbund beitreten, oder ihn wieder verlassen.

Für die gesamte Kommunikation zwischen dem Anbieter und einer Anlage ist ein sogenannter *Koppelagent* zuständig, der das verwendete dienstorientierte Kommunikationsprotokoll versteht und umsetzt. Der Koppelagent implementiert eine definierte Schnittstelle nach Außen, in Richtung des Anbieters hin. Nach Innen, in Richtung des Unternehmens beziehungsweise der Anlage hin, kann jeder Koppelagent eine eigene Schnittstelle implementieren, um den jeweiligen spezifischen Gegebenheiten vor Ort, wie der Netzwerkarchitektur und den Verwendeten IT-Systemen, Rechnung zu tragen. Auf Seiten des Anbieters ist ein sogenannter *Anbieteragent* für die Kommunikation mit den Koppelagenten zuständig.

In Abbildung 2 ist eine alternative Darstellung beziehungsweise Gestaltung eines einzelnen Produktionsverbandes dargestellt. Der dargestellte Produktionsverbund ist derjenige, welcher aktuell zwischen den am Demonstrator beteiligten Instituten und Lehrstühlen installiert wird. Der Aufbau entspricht dem prinzipiellen Aufbau aus Abbildung 1, wobei ein Institut mit einem Unternehmen gleichzusetzen



ist. Zusätzlich ist hier aber die Möglichkeit berücksichtigt, dass ein Unternehmen mehrere (unabhängige) Anlagen betreibt, die teilweise oder vollständig Teil des Produktionsverbundes sind. Jede dieser Anlagen stellt dabei eine unabhängige Einheit dar, mit entsprechendem Informationsmodell und IT-Systemen und wird innerhalb des Verbundes durch jeweils einen eigenen Koppelagenten repräsentiert (in der Abbildung die Gesamtheit aller Dienste), der das dienstorientierte Netzwerkprotokoll umsetzt. Je nach Standort sind unterschiedlich viele Anlagen Teil des Demonstrators, die teilweise das gleiche Produkt fertigen können, teilweise aber auch unterschiedliche Aufgaben ausführen können. So existieren zum Beispiel mehrere Anlagen zur Joghurtproduktion, aber nur eine Anlage zur Abfüllung des Joghurts in Becher.

Anlagen, welche das gleiche Produkt herstellen, konkurrieren bei der Auftragsvergabe durch den Anbieter um den Auftragszuschlag. Ein Auftrag wird nach einer Vielzahl unterschiedlicher Kriterien vergeben, darunter die Kosten und der Liefertermin. Über die Vergabe entscheidet der Anbieteragent nach Einholung aller relevanter Angebote von den am Produktionsverbund beteiligten Anlagen.

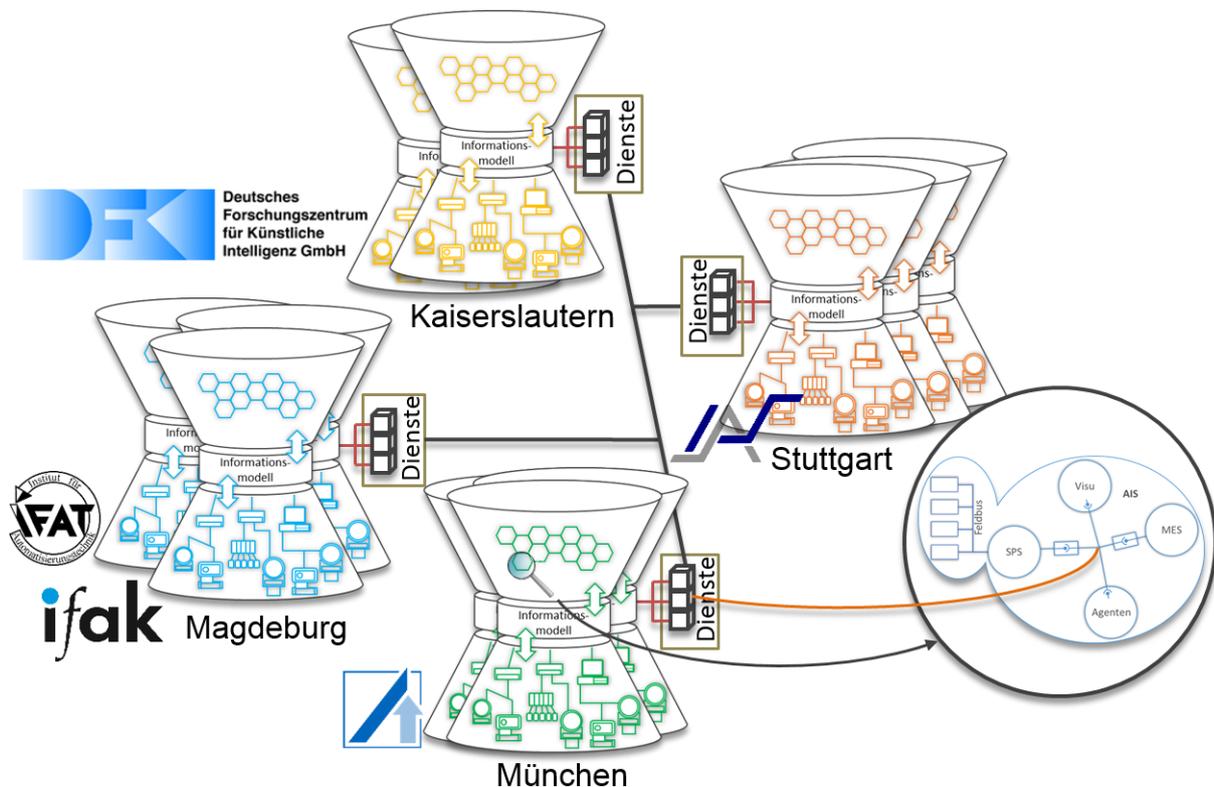


Abbildung 2: Deutschlandweiter Industrie-4.0-Demonstrator

Wie in Abbildung 2, am Beispiel München, angedeutet kann jedes Unternehmen einen eigenen, spezifischen Innenaufbau haben, ohne die Kompatibilität zum Produktionsverbund aufgeben zu müssen. Durch die Verwendung des Koppelagenten findet eine vollständige Kapselung des Innenlebens einer Anlage beziehungsweise eines Unternehmens statt.

Das Innenleben einer Zulieferanlage kann sehr komplex sein, aber auch beispielsweise so einfach wie in Abbildung 3 aussehen. Dort ist der Aufbau der Anlage A00 abgebildet, die sich dem Produktionsverbund anschließen möchte. Die Anbindung an das Produktionsnetzwerk geschieht natürlich auch hier weiterhin über einen Koppelagenten, der die Kommunikation nach Außen übernimmt. Die Anlage besteht außerdem aus nur einer SPS, einem MES und einer Visualisierung. Alle diese IT-Systeme sind über ein unternehmensinternes Netz miteinander verbunden, kommunizieren aber nicht zwingend untereinander. Die genaue interne Architektur ist dabei frei – es muss nur ein passender Koppelagent mit den entsprechenden Schnittstellen nach Innen und Außen angeboten werden. Der Koppelagent nimmt entsprechende Anfragen des Anbieters entgegen und leitet die dazugehörigen Aktionen ein, wie zum



Beispiel den Start der Produktion, indem die relevanten Daten an das MES übertragen werden. Zu diesem Zweck implementiert der Koppelagent diverse Dienste, die gemeinsam die Schnittstelle nach Außen hin darstellen. Teile der Dienste, das heißt ihr Name und ihre Funktion, sind dabei vom Netzwerkprotokoll vorgeschrieben, wie zum Beispiel ein Index-Dienst, der alle verfügbaren Dienste auflistet (*White Pages*). Ob der Koppelagent auch nach Innen dienstorientiert arbeitet ist, ebenso wie die Architektur, dem jeweiligen Anlagenbetreiber freigestellt.

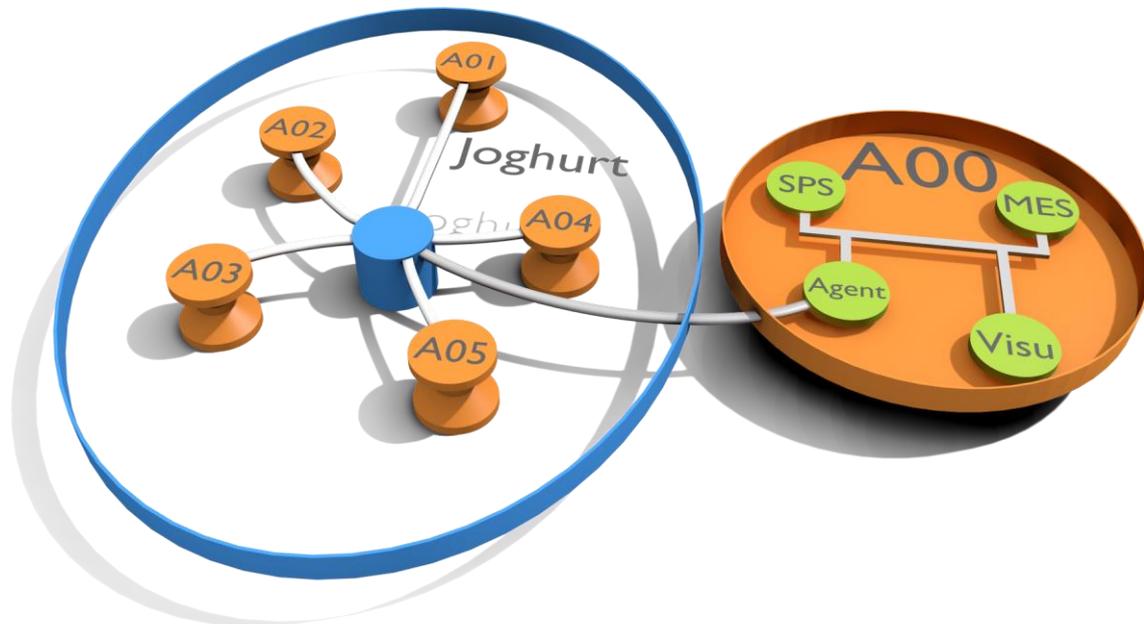


Abbildung 3: Interner Aufbau von Anlage A00 und Verbindungsaufbau zum bestehenden Netzwerk

Ein alternativer und wesentlich komplexerer Aufbau eines Unternehmens und dessen Netzwerks ist in Abbildung 4 ersichtlich. Im Gegensatz zum vorherigen Beispiel aus Abbildung 3, gibt es hier nicht nur mehrere unterschiedliche IT-Systeme, sondern diese sind hier auch wesentlich enger gekoppelt. Die IT-Systeme sind nicht nur in einem gemeinsamen Netzwerk, sondern auch datentechnisch gekoppelt, das heißt zur Kommunikation wird ein gemeinsames Datenmodell verwendet. Der eigentliche Datenaustausch erfolgt dabei über ein kanonisches Datenformat, weshalb die einzelnen IT-Systeme nicht direkt an das Netzwerk angeschlossen sind, sondern über sogenannte *Terminals*, welche dazu dienen die proprietären Datenformate der diversen IT-Systeme in das kanonische und damit allgemeinverständliche Datenformat – und zurück – zu wandeln. Die Terminals können als größtenteils unabhängiges Programm implementiert sein, das zwischen IT-System und Netzwerk sitzt (wie beispielsweise zwischen MES und Netzwerk), oder auch direkt innerhalb eines IT-Systems implementiert sein (wie beispielsweise im Diagnosesystem). Durch die Vereinheitlichung des Datenmodells und –formats sind unternehmensintern und später auch über das gesamte Produktionsnetzwerk hinweg weiterführende Szenarien denk- und umsetzbar.

Eines dieser möglichen Szenarien ist, neben der Netzarchitektur, ebenso in Abbildung 4 dargestellt. Beispielsweise kann ein Diagnosesystem dienstbasiert direkt auf freigegebene Daten des PLS und des PAM zugreifen und entsprechende Diagnosefunktionen ausführen. Im Beispiel wird eine Temperaturplausibilisierung unter Zuhilfenahme des PAM durchgeführt. Hierzu holt sich das Diagnosesystem historische Daten des PAM ((1) und (2)) und vergleicht diese mit aktuellen Temperaturen direkt aus dem Prozess (5), die über das PLS abgerufen werden ((3) und (4)).

Somit können viele bisher manuell durchgeführte Aufgaben automatisiert werden, wodurch sich nicht nur ein Geschwindigkeitsvorteil ergibt, sondern zusätzlich eine Reduktion der Fehlerhäufigkeit. Das volle Potential einer umfassenden Diagnose ergibt sich aber erst im Zusammenspiel mehrerer Anlagenbetreiber unter Einbeziehung der Komponentenhersteller. Diese Einbeziehung ist nur möglich,



wenn eine Anbindung der unternehmensinternen Daten an das Internet erfolgt. Auch dies lässt sich mit Hilfe des Koppelagenten realisieren und die bereits existente Dienstorientierung nach Außen repräsentieren. Zusätzlich lassen sich auf diese Weise auch weitere anlagen- und unternehmensübergreifende Aufgaben bearbeiten, indem die gesammelten Daten autorisierten Mitgliedern des Verbundes zur Verfügung gestellt werden.

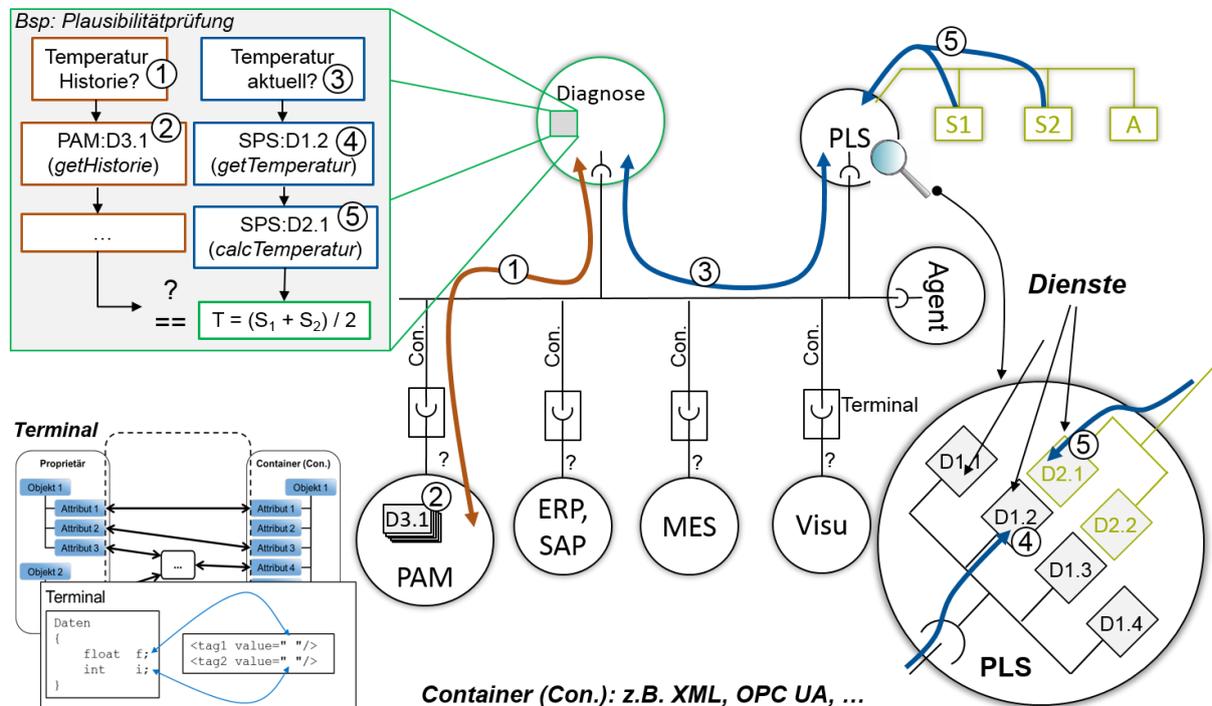


Abbildung 4: Beispielhafte IT-Systeme innerhalb eines Unternehmens

Um Teil eines Produktionsverbundes zu werden, sind nur wenige Schritte erforderlich. Ein Anlagenbetreiber, der einem Netzwerk betreten möchte, holt sich die IP-Adresse des Anbieters vom Anbieter und verbindet seinen Koppelagent mit dem Anbieteragent, der im Folgenden alle weiteren Aktionen innerhalb des Produktionsnetzwerks koordiniert, zum Beispiel die Auftragsverteilung, oder die oben beschriebene, verteilte Diagnose. Das Beitreten zu einem Netzwerk geschieht durch ein Verbinden mit dem Anbieter auf IP-Ebene und einer anschließenden Authentifizierung (Security).

Der Aufbau des Netzwerkverkehrs zwischen Anbieter und Anlagen und damit die Kommunikation zwischen den einzelnen Agenten wird durch das neukonzipierte JPP (Joghurt-Produktions-Protokoll) festgelegt.

## Protokoll (JPP)

Das JPP ist ein dienstorientiertes, textbasiertes Protokoll, mit dem sich beliebige Daten verbindungs- und dienstorientiert austauschen lassen. Es setzt auf das *Transmission Transport Protocol* (TCP) als Transportprotokoll auf und implementiert darüber liegenden Schichten, ähnlich wie zum Beispiel das *Hypertext Transfer Protocol*.

Das JPP besteht intern seinerseits wiederum aus drei vollständig unabhängigen, aufeinander aufbauenden Schichten (Kapselung):

1. Basisschicht  
Grundlegender Aufbau (Protokoll-Syntax) des Protokolls, Header-Definition, Umfang, ...
2. Dienstschicht  
Dienste (Dienst-Syntax), Datenformat, ...



3. Datenschicht  
Daten, Datenmodell (Semantik), ...

Die drei Schichten des JPP sind zur Verdeutlichung in Abbildung 5 dargestellt.



Abbildung 5: Die Protokollschichten, -namen und -aufgaben

Die drei Schichten können unabhängig voneinander definiert und implementiert werden. Aktuell ist die Basisschicht definiert und fast vollständig implementiert. Zusätzlich ist eine erste einfache Version der Dienstschicht definiert und implementiert. Die Datenschicht ist Teil weiterer Arbeitspakete. Jede Schicht ist so spezifiziert, dass sie keine Einschränkungen an die darüber liegenden Schichten stellt und jeweils auf der darunterliegenden Schicht aufbaut.

Für das im Rahmen des Demonstrators verwendete Netzwerkprotokoll (JPP) ist, wie bereits erwähnt, der Unternehmens- und Anlagenaufbau irrelevant. Der Koppelagent muss für die Teilnahme an einem Produktionsverbund das JPP implementieren und die gewünschten Dienste auf Basis der Basisschicht anbieten, das heißt die Syntax und den Protokollaufbau beachten. Wie die Daten innerhalb des Unternehmens beziehungsweise der Anlage weiter verarbeitet werden ist nicht von Belang. Der Agent kann direkt mit der SPS kommunizieren, den „Umweg“ über ein MES gehen, oder auch einen Auftragszettel ausdrucken, der im Anschluss manuell abgearbeitet wird. Ebenso ist es unerheblich, wie der Agent zu seiner Entscheidung kommt. Es gilt das Prinzip, dass die Anlage selbst „immer noch am Besten weiß“, wie man Kosten, Auslastung, ... am besten berechnet, beziehungsweise feststellt. Möglich ist hier eine Entscheidung auf Modellbasis, aber auch eine einfache Ja-Nein-Aussage. Eine Festlegung auf ein bestimmtes Prinzip schränkt lediglich die allgemeine Verwendbarkeit unverhältnismäßig ein und wurde daher nicht verfolgt, ist also nicht Teil des Protokolls, dessen Spezifikation somit nur die Kommunikation zwischen den Agenten umfasst.

Im Rahmen des beschriebenen Produktionsszenarios starten fast alle Anfragen von Seiten des Anbieteragents (Server) aus. Er stellt an bestimmte Dienste (Dienstschicht) beispielsweise Anfragen nach Kosten und Lieferzeiten für bestimmte Produkte – zum Beispiel 100 Becher Erdbeerjoghurt. Die Syntax für diese Anfrage ist innerhalb der Basisschicht definiert, das Datenformat und die Datensemantik innerhalb der Datenschicht. Daraufhin bekommt er von den einzelnen Koppelagenten (beziehungsweise den Diensten) der Anlagen (Clients) die Kosten und Lieferzeiten zurück, sofern die Anlage grundsätzlich in der Lage ist, den gewünschten Joghurt zu produzieren. Für diese Kommunikation zwischen Anbieter und Zulieferer sind lediglich die Ausgangsstoffe und das Endprodukt von Belang, nicht der Prozess, der aus den Ausgangsstoffen das Endprodukt erstellt. Ebenso ist die Anlagenstruktur seitens des Zulieferers nicht von Belang, entscheidend ist lediglich das Liefern des gewünschten Produkts unabhängig vom exakten Herstellungsprozess (1000kg Joghurt kann sowohl vollautomatisiert hergestellt



werden, als auch manuell durch eine Vielzahl von Arbeitern). Da der Anbieter nicht weiß, wie der Zulieferer sein Produkt herstellt und die technischen Vorgänge weder wissen will noch muss (Produkt und Produkteigenschaften zählen), liegt die Entscheidung für ein Vorgehen zur Entschlussfassung alleine beim Zulieferer, der diese basierend auf der Datenschicht in seinem Koppelagenten implementiert, so dass dieser wunschgemäß auf Anfragen des Anbieters reagieren kann.

## Protokollspezifikation

Im Rahmen der Entwicklung des Demonstrators und basierend auf den daraus resultierenden Anforderungen, wurde eine Spezifikation für das Netzwerkprotokoll JPP (Joghurt-Produktions-Protokoll) erarbeitet. Das JPP ist ein sehr schlankes Protokoll und basiert auf den folgenden grundlegenden Eigenschaften:

- Die einzelnen Elemente des Protokolls sind durch Zeilenumbrüche ( $\backslash n$ ) ohne Zeilenrücklauf ( $\backslash r$ ) voneinander getrennt
- Jedes Datenpakets beginnt mit einem dreizeiligen *Header*
- Die ersten drei Buchstaben der ersten Zeile des Headers sind *JPP (Identifizier)*
- Nach dem Identifizier folgt ohne Leerzeichen die Versionsnummer
- In der zweiten Zeile steht die Gesamtgröße des Datenpakets inklusive Header in Byte
- In der dritten Zeile folgt eine *Sequenznummer* um asynchrone Kommunikation zu erleichtern
- Auf den Header folgt in der vierten Zeile die *Paketaktion*, gefolgt von dazugehörigen optionalen Elementen

Der Identifizier, zusammen mit der Versionsnummer, dient dazu, ein JPP-Paket zu erkennen und die richtige Abarbeitungsroutine auszuwählen, sofern mehrere Versionen des JPP existieren. Da es sich um insgesamt vier ASCII-Zeichen (zu je einem Byte) handelt, lässt sich der Identifizier nebst Versionsnummer auch binär als vorzeichenlose 32Bit-Ganzzahl lesen. Die Angabe der Gesamtgröße des Datenpakets dient dazu feststellen zu können, wann ein Paket vollständig übertragen wurde. Für die Sicherstellung der richtigen Reihenfolge der Pakete, sowie der Fehlerfreiheit sind tieferliegende Protokollschichten und Protokolle, wie TCP, verantwortlich. Da das JPP keine Festlegung bezüglich der Abarbeitung macht (synchron oder asynchron), ist zur Vereinfachung einer asynchronen Abarbeitung eine Sequenznummer Teil des Headers. Sie dient auf Seiten des Anfragenden dazu, eine einfache Zuordnung zwischen Anfrage und Antwort zu ermöglichen. Auf den Header folgt dann direkt eine der folgenden vier Paketaktionen beziehungsweise Basisdienste der Basisschicht:

- *CALL service(param1, param2, ...)*
- *RETURN data*
- *ERROR error*
- *INDEX [service]*

Mit diesen vier Paketaktionen sind alle wichtigen Aufgaben des Protokolls umsetzbar. Die CALL-Aktion ruft die Methode beziehungsweise den höheren Dienst *service* (der Dienstschicht) mit den Parametern *param1, param2, ...* auf. Die Parameter dürfen dabei jedes beliebige unterstützte Datenformat haben, ihre Anzahl ist ebenso beliebig. Die Rückgabe eines solchen Aufrufs wird über die RETURN-Aktion übermittelt. Die Zuordnung zwischen Aufruf und Rückgabe kann im asynchronen Fall über die Sequenznummer erreicht werden. Sofern mehr als ein Rückgabewert für eine Methode beziehungsweise einen Dienst zurückgegeben werden soll, kann dies in Form eines Arrays geschehen, dessen Syntax weiter unten näher erläutert wird. Sofern während eines Aufrufs ein Fehler auftritt, wird dieser, in Form eines Fehlercodes und einer Fehlerbeschreibung, über die ERROR-Aktion zurückgegeben. Die INDEX-Aktion realisiert den Verzeichnisdienst eines Agenten (White Pages), indem sie, sofern ohne Parameter aufgerufen, eine Liste aller verfügbaren Methoden beziehungsweise Dienste zurückgibt. Wird die INDEX-Aktion mit Parameter (*service*) aufgerufen, gibt sie eine Beschreibung der Methode beziehungsweise des Dienstes zurück. Diese Beschreibung enthält eine Liste der erforderlichen Parameter, ebenso wie eine Liste der Rückgaben. Die beiden Listen werden in Form zweier Arrays zurückgegeben.



Obwohl es sich beim JPP prinzipiell um ein textbasiertes Protokoll handelt, können auch binäre Daten und Datenstrukturen übertragen werden. Alle nicht-binären Daten werden als ASCII-Strings repräsentiert. Binäre Daten werden auch binär übertragen, sie werden lediglich durch die Zeichenfolge *2#len#* eingeleitet, wobei *len* die Länge der Binärdaten in Byte angibt. Diese Einleitung dient dazu, dem Protokollparser die nun kommenden binären Daten und ihre Größe anzukündigen. Arrays werden in eckigen Klammern übertragen und können wiederum aus beliebigen Datentypen bestehen. Einzelne Elemente eines Arrays werden durch ein Komma getrennt. Im Gegensatz dazu werden Strukturen mittels geschweifeter Klammern umschlossen, wobei auch hier die einzelnen Elemente durch Komma getrennt werden.

Durch die OOP-artige Kapselung der Anlagen und Zugriff mittels klarem Interface ist außer der Definition des Interfaces keine weitere „Standardisierung“ nötig. Der Anbieter fragt lediglich Ergebnisse ab, die genaue Berechnung dieser bleibt – wie beschrieben – im Bereich der Anlagen und deren Betreiber. Des Weiteren ermöglicht die Kapselung eine klare Trennung zwischen den beiden Bereichen Organisation (Anbieteragent / Server) und Produktion (Koppelagent / Client).

Für das Produktionsszenario ergibt sich dann folgender Ablauf: Die Anlagen (Clients) verbinden sich zum Anbieter und Authentifizieren sich. Sobald der Anbieter einen neuen Auftrag erhält, werden die relevanten Daten verarbeitet und der Auftrag entsprechend der Verhandlung unter den Anlagen verteilt. Sollte es sich um einen sehr großen Auftrag handeln, der aufgeteilt werden soll, so ist es Aufgabe des Anbieters diesen Auftrag in kleinere Einheiten zu teilen.

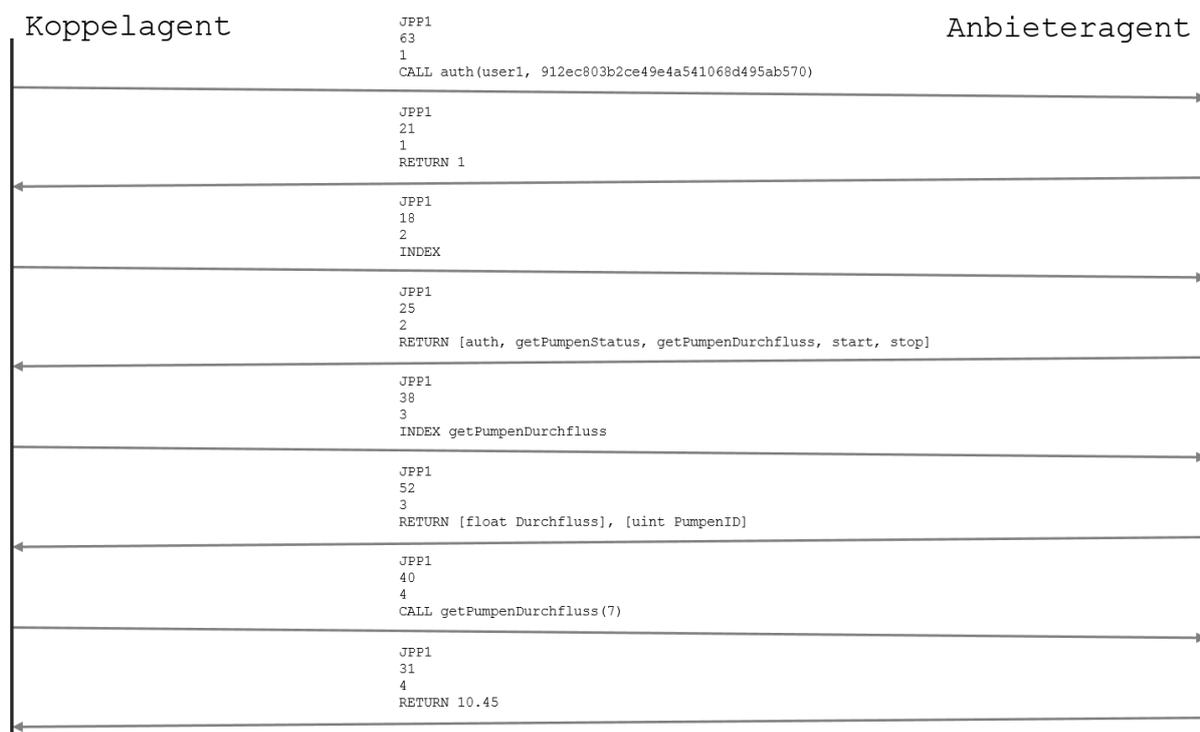


Abbildung 6: Beispielhaftes Sequenzdiagramm einer Kommunikation

Das in Abbildung 6 dargestellte Sequenzdiagramm zeigt eine beispielhafte Kommunikation zwischen einem Koppelagent (Client) und einem Anbieteragent (Server). Diese Kommunikation beginnt mit dem Verbindungsaufbau und dem Aufruf des Authentifizierungsdienstes, dem der Nutzernamen und ein md5-verschlüsseltes Passwort übergeben werden. Die Rückgabe – ebenfalls mit der Sequenznummer 1 – zeigt eine erfolgreiche Authentifizierung an. Mit dem nächsten Paket im Beispiel fordert der Koppelagent eine Liste aller verfügbaren Dienste an, die ihm im Anschluss geliefert wird. Daran anschließend wird vom Koppelagent die Definition des Dienstes *getPumpenDurchfluss* angefordert, welche ebenfalls zurückgegeben wird. Zuletzt wird im Beispiel der Dienst *getPumpenDurchfluss* mit der *PumpenID* 7 aufgerufen, welcher den Wert 10,45 zurückgibt. Dieser Ablauf könnte unter Anderem in Form



einer Zustandsmaschine implementiert werden, aber auch eine eventbasierte Implementierung auf Basis der Sequenznummern ist möglich.

## Agenten

Wie in den vorherigen Kapiteln erläutert, muss ein Anlagenagent für eine erfolgreiche Teilnahme an einem Produktionsnetzwerk das JPP implementieren und die entsprechenden Netzwerkschnittstellen nach Außen und Innen zur Verfügung stellen, sowie die nötigen Dienste anbieten.

Somit ergibt sich eine dreiteilige Struktur für einen Anlagenagenten. Nach Außen muss eine Netzwerkverbindung mittels TCP/IP zum Anbieteragenten hergestellt werden können. Die Datenübertragung folgt dann dem JPP, wobei der Ablauf der Kommunikation beispielsweise in Form einer Zustandsmaschine abgearbeitet werden kann. Nach Innen muss eine Verbindung zum unternehmenseigenen Netzwerk und den daran angeschlossenen IT-Systemen hergestellt werden können. Außerdem müssen die für die verschiedenen Aufgaben nötigen Daten ausgetauscht werden können. In einem sehr einfachen Fall könnte diese Verbindung zum Beispiel aus einer direkten OPC-Verbindung zwischen Agent und SPS bestehen, so dass der Agent direkt die Produktion anstoßen könnte. Der dritte Teil des Agenten sind die aufrufbaren Dienste. Welche Dienste angeboten werden müssen ist dabei implementierungsabhängig. Für das Produktionsszenario ist einer der Dienste zum Beispiel der Dienst zur Preisverhandlung für die Produktion einer bestimmten Menge Joghurt und der Dienst zum Starten der Produktion. Für die wunschgemäße Arbeit der Dienste ist außerdem eine Wissensbasis notwendig, auf deren Basis die nötigen Entscheidungen getroffen werden können.

Die Struktur des Anbieteragenten ist ebenfalls dreiteilig. Der erste Teil ist für die Kommunikation mit den Anlagenagenten zuständig und implementiert daher ebenfalls das JPP auf Basis von TCP/IP. Somit stellt dieser Teil die Anbindung nach Außen hin dar. Der zweite Teil ist die Anbindung an das (interne) webbasierte Kundenportal – diese Anbindung ist zum Beispiel über eine Datenbank realisierbar. Der dritte Teil des Anbieteragenten sind wieder die diversen Dienste, die für die Auftragsvergabe notwendig sind.

## Produkt-, Prozess und Anlagenmodell

Sofern beliebige Anlagen miteinander in einem Verbund arbeiten sollen, ist hierfür ein allumfassendes Modell erforderlich, in dem unter anderem die Produkteigenschaften, der Produktionsprozess, die Anlagenstruktur und die Anlagenkomponenten vollständig abgebildet werden müssen. Ein solches allumfassendes Modell ist aber nicht notwendig, da ein Anlagenbetreiber seine Anlage wahrscheinlich immer nur einem Verbund anschließen wird, der im Wesentlichen solche Produkte herstellt, die normalerweise auf seiner Anlage produziert werden. So wird ein Mineralölproduzent seine Anlage nicht an einem Milchproduktverbund anmelden. Seine Anlage kann zwar theoretisch fähig sein auch Joghurt und Quark zu produzieren, allerdings wird dieses Vorhaben an weiteren Randbedingungen scheitern. Dazu gehört die fehlende Lebensmittelechtheit der Anlage, die fehlende Dokumentation des Prozesses, die lange Umrüstzeit, die fehlende Erfahrung mit der Joghurtproduktion und vieles mehr. Neben der Anlage gibt es zusätzliche Umweltrandbedingungen, die eine völlig freie Verwendung einer Anlage für beliebige Produkte verhindert. Dazu gehörten zum Beispiel Möglichkeiten zur Anlieferung der Rohstoffe (Zufahrten, Brücken, Türöffnungen, ...), Verfügbarkeit der Rohstoffe und weitere nichtfunktionale Anforderungen an die Anlage.

Ein allumfassendes Modell ist daher sehr schwer zu entwickeln und zu erstellen, da zusammenfassend unter anderem die folgenden Bereiche abgedeckt werden müssen:

- Produkteigenschaften und -mengen
- Rohstoffeigenschaften und -mengen
- Anlagenkomponenten, -funktion, -struktur
- Komponentengrenzwerte (zum Beispiel Maximalviskosität des Produkts, maximale Festkörpergröße und -anzahl)



- Nichtfunktionale Anforderungen (Lebensmittelechtheit, Umrüstzeiten, Logistik, ...)
- ...

Fokussiert man die Betrachtung jedoch auf die Ausgangs- und Endprodukte eines Industriezweigs (zum Beispiel Milchprodukte) und betrachtet den exakten Herstellungsprozess – welcher ohnehin äußerst schwer vollständig abbildbar ist (manuelle Herstellung) – nicht, kann das Modell deutlich vereinfacht werden, da die prinzipielle Tauglichkeit der Anlage für ein bestimmtes Produkt vom Anlagenbetreiber garantiert wird. Der Anlagenbetreiber, welcher das meiste Wissen über seine Anlage hat, muss dann lediglich die Wissensbasis des Koppelagenten mit Informationen zu den verfügbaren Rohstoffen und tatsächlich möglichen Endprodukten (Joghurt zum Beispiel nur bei lebensmitteltauglichen Anlagen) sowie deren festen und veränderlichen Eigenschaften füllen. Der Koppelagent kann dann mit dem Anbieter verhandeln, ob und zu welchen Konditionen das gewünschte Produkt mit welchen Eigenschaften hergestellt werden kann. Ein Ressourcen- und Prozessmodell ist nicht nötig.



Abbildung 7: Wissensbasis des Anlagenagenten und Verhandlung

Wenn ein Betreiber zum Beispiel angibt Joghurt aus Milch und Erdbeeren produzieren zu können, ist damit impliziert, dass es sich bei der verwendeten Anlage um eine lebensmitteltaugliche Anlage handelt, die Milch und Erdbeeren in geeigneter – aber unbekannter – Weise zu Joghurt verarbeiten kann. Diese Information wird in der Wissensbasis des Anlagenagenten abgelegt, so dass dieser, wie in Abbildung 8 ersichtlich, bei einer Anfrage passend antworten kann.

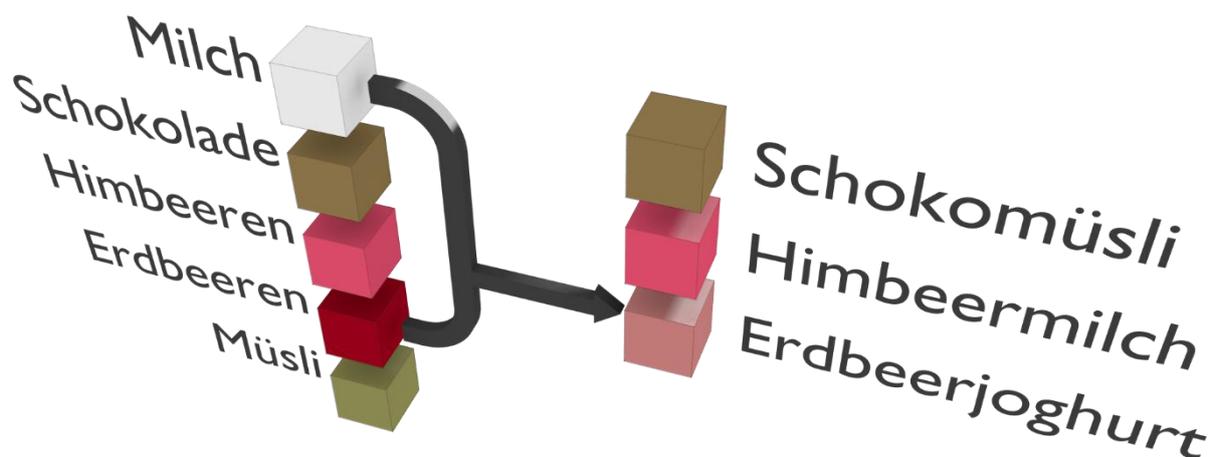


Abbildung 8: Beispiel für die Wissensbasis

Natürlich müssen sich alle Anlagenbetreiber vorher darüber einigen, was unter einem bestimmten Produkt zu verstehen ist – was also die unveränderlichen Grundeigenschaften eines Produktes sind.



Alle weiteren für den Anbieter interessanten Fakten sind von der Anlage größtenteils unabhängig, wie zum Beispiel der Preis.

Trotzdem gilt auch hier weiterhin, dass es jedem Anbieter und Anlagenbetreiber frei steht (im Sinne der (losen) Kopplung und Kapselung) einen anderen Ansatz zum Bestimmen der Anlagenfähigkeiten zu nutzen. Sollte sich ein Anbieter für ein alternatives Vorgehen entscheiden, sind bereits existierende Koppelagenten unter Umständen inkompatibel, weshalb sich eine Ausweichstrategie in Form eines optionalen Rückfalls auf die Standardlösung anbietet.

Insgesamt unterstützt der Ansatz somit eine divergente und heterogene Systemlandschaft innerhalb eines Produktionsnetzwerks, innerhalb dessen trotzdem prinzipiell alles möglich ist.

## Zusammenfassung

Im Rahmen dieses Dokuments wird die Initiative mehrerer Lehrstühle und Institute für einen gemeinsamen, deutschlandweiten Industrie-4.0-Demonstrator vorgestellt. Dieser Demonstrator befindet sich aktuell in der Entwicklung und Implementierung [1] und soll die verteilten Anlagen der Institute zu einem gemeinsamen Produktionsverbund zusammenschließen. Der gesamte Ansatz soll später auch auf die Industrie übertragbar sein. Für den Demonstrator wurden in gemeinsamer Arbeit verschiedene Szenarien entwickelt, die nun umgesetzt werden.

Auf Basis der gemeinsamen Szenarien wurde am Lehrstuhl AIS der TU München, gemeinsam mit dem ifak Magdeburg und dem Lehrstuhl IAS der Universität Stuttgart, eine Architektur, sowie darauf aufbauend ein geeignetes Netzwerkprotokoll entwickelt und implementiert. Ebenso wurde damit begonnen ein erstes Modell zum Datenaustausch zwischen den beteiligten Anlagen zu konzipieren. Der gesamte Ansatz verfolgt eine vollständige Kapselung der ohnehin unabhängigen Anlagen, um weiterhin eine hohe Selbstständigkeit dieser zu gewährleisten und um die Betreiber nicht zu einer vollständigen Vereinheitlichung aller Vorgänge und Schnittstellen zu zwingen.

## Literatur

- [1] C. Diedrich, A. Fay, J. Grützner, P. Göhner, B. Vogel-Heuser, M. Weyrich, M. Wollschlaeger: „Automatisierungstechnischer Forschungsanlagenverbund für Industrie 4.0“. In: *Markt & Technik*. WEKA FACHMEDIEN GmbH, Haar.

