



(12) **Offenlegungsschrift**

(21) Aktenzeichen: **10 2011 053 851.8**

(22) Anmeldetag: **22.09.2011**

(43) Offenlegungstag: **29.03.2012**

(51) Int Cl.: **H04L 12/24 (2012.01)**

(30) Unionspriorität:
12/889,064 **23.09.2010** **US**

(74) Vertreter:
Meissner, Bolte & Partner GbR, 80538, München, DE

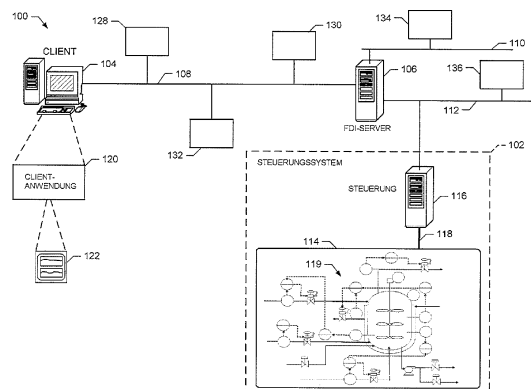
(71) Anmelder:
Fisher-Rosemount Systems, Inc., Austin, Tex., US

(72) Erfinder:
Nixon, Mark, 78681, Round Rock, Texas, US

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Serviceorientiertes Framework zur Kommunikation mit Geräten in einem Prozesssteuerungssystem**

(57) Zusammenfassung: Es wird ein serviceorientiertes Framework zum Kommunizieren mit Geräten in einem Prozesssteuerungssystem offenbart. Ein hier offenbartes beispielhaftes Verfahren zum Kommunizieren mit einem Gerät in einem Prozesssteuerungssystem umfasst das Aufrufen eines Service zum Kommunizieren mit dem Gerät in dem Prozesssteuerungssystem, wobei der Service eine generische Schnittstelle aufweist, die unabhängig von einem zur Implementierung des Prozesssteuerungssystems verwendeten Prozesssteuerungsnetzwerkprotokoll ist, Übersetzen des Service in eine oder mehrere Netzwerkoperationen zur Implementierung des Service, wobei die Netzwerkoperationen für das zum Implementieren des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch sind, und Verwenden einer Netzwerkschnittstelle, die für das zum Implementieren des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch ist, um mit dem Gerät gemäß den einen oder mehreren Netzwerkoperationen zu kommunizieren.



Beschreibung

GEBIET DER ERFINDUNG

[0001] Diese Erfindung betrifft Prozesssteuerungssysteme im Allgemeinen und spezieller ein serviceorientiertes Framework zur Kommunikation mit Geräten in einem Prozesssteuerungssystem.

ALLGEMEINER STAND DER TECHNIK

[0002] Prozesssteuerungssysteme wie jene, die in chemischen, Erdöl und anderen Prozessen verwendet werden, schließen in der Regel ein oder mehrere Prozesssteuerungen ein, die kommunikativ an mindestens einen Client- oder Bediener-Arbeitsplatzrechner und über analoge, digitale oder kombiniert analog/digitale Busse an ein oder mehrere Feldgeräte gekoppelt sind. Die Feldgeräte können beispielsweise Ventile, Ventilstellungsregler, Schalter und Sender (z. B. Temperatur-, Druck- und Durchflussratensensoren) sein, und sie können innerhalb des Prozesses Prozesssteuerungsfunktionen durchführen, wie Öffnen oder Schließen von Ventilen und Messen von Prozesssteuerungsparametern. Die Steuerungen erhalten Signale, die Prozessmessungen anzeigen, welche von den Feldgeräten vorgenommen wurden, verarbeiten diese Informationen, um eine Steerroutine zu implementieren, und generieren Steuersignale, die über die Busse oder anderen Kommunikationsleitungen an die Feldgeräte gesendet werden, um den Betrieb des Prozesses zu steuern. Auf diese Weise führen die Steuerungen Steuerungsstrategien oder -routinen unter Verwendung der Feldgeräte über die Busse und/oder andere Kommunikationsverbindungen aus und koordinieren diese.

[0003] Informationen von den Feldgeräten und den Steuerungen können einer oder mehreren Anwendungen (z. B. Routinen, Programmen usw.) als Laufzeitdaten zur Verfügung gestellt werden, die von dem Client-/Bediener-Arbeitsplatzrechner (z. B. einem System auf Prozessorbasis) ausgeführt werden, um es einem Bediener zu ermöglichen, gewünschte Funktionen in Bezug auf den Prozess durchzuführen. Bei einigen dieser Funktionen wird der aktuelle Status des Prozesses (z. B. über eine grafische Benutzeroberfläche) angezeigt, der den Prozess auswertet, den Betrieb des Prozesses (z. B. über ein visuelles Objektdiagramm) modifiziert, usw. Viele Prozesssteuerungssysteme schließen auch eine oder mehrere andere Clientstationen ein, die auch als Anwendungsstationen bezeichnet werden. Eine Anwendungsstation wird typischerweise mit einem PC, einem Arbeitsplatzrechner oder dergleichen implementiert, die über ein lokales Netzwerk (LAN) kommunikativ an die Steuerungen, Arbeitsplatzrechner der Bedienungspersonen und anderen Systeme innerhalb des Prozesssteuerungssystems gekoppelt sind.

Jede Anwendungsstation kann eine oder mehrere Strategien, Routinen oder Anwendungen ausführen, die Einsatzmanagementfunktionen, Wartungsmanagementfunktionen, virtuelle Steuerungsfunktionen, Diagnosefunktionen, Echtzeit-Überwachungsfunktionen, sicherheitsrelevante Funktionen, Konfigurationsfunktionen usw. innerhalb des Prozesssteuerungssystems durchführen.

KURZDARSTELLUNG

[0004] Die hier beschriebenen beispielhaften Verfahren, Vorrichtungen und Fertigungsartikel betreffen Prozesssteuerungssysteme im Allgemeinen und spezieller ein serviceorientiertes Framework zur Kommunikation mit Geräten in einem Prozesssteuerungssystem. In einem hier offenbarten Beispiel umfasst ein Verfahren zum Kommunizieren mit einem Gerät in einem Prozesssteuerungssystem das Aufrufen eines Services, um mit dem Gerät in dem Prozesssteuerungssystem zu kommunizieren, wobei der Service eine generische Schnittstelle bereitstellt, die unabhängig von einem Prozesssteuerungsnetzwerkprotokoll ist, das zur Implementierung des Prozesssteuerungssystems verwendet wird. In dem beispielhaften Verfahren wird der Service auch in eine oder mehrere Netzwerkoperationen übersetzt, um den Service zu implementieren, wobei die Netzwerkoperationen für das zur Implementierung des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch sind. Das beispielhafte Verfahren beinhaltet ferner die Verwendung einer Netzwerkschnittstelle, die für das Prozesssteuerungsnetzwerkprotokoll spezifisch ist, welches zur Implementierung des Prozesssteuerungssystems zur Kommunikation mit dem Gerät gemäß der einen oder den mehreren Netzwerkoperationen verwendet wird.

[0005] In einem anderen hier offenbarten Beispiel speichert ein physischer Fertigungsartikel maschinenlesbare Anweisungen, die bei Ausführung eine Maschine dazu bringen, mindestens einen Service zum Kommunizieren mit einem Gerät in einem Prozesssteuerungssystem aufzurufen, wobei der Service eine generische Schnittstelle bereitstellt, die unabhängig von einem zur Implementierung des Prozesssteuerungssystems verwendeten Prozesssteuerungsnetzwerkprotokoll ist. Die maschinenlesbaren Anweisungen führen bei Ausführung auch dazu, dass die Maschine den Service mindestens in eine oder mehrere Netzwerkoperationen übersetzt, um den Service zu implementieren, wobei die Netzwerkoperationen für das Prozesssteuerungsnetzwerkprotokoll spezifisch sind, die zur Implementierung des Prozesssteuerungssystems verwendet werden. Die maschinenlesbaren Anweisungen führen bei Ausführung ferner dazu, dass die Maschine mindestens eine Netzwerkschnittstelle verwendet, die für das Prozesssteuerungsnetzwerkprotokoll spezifisch ist, welches zur Implementierung des Prozesssteuerungssystems verwendet wird.

systems zur Kommunikation mit dem Gerät gemäß der einen oder den mehreren Netzwerkoperationen verwendet wird.

[0006] Der physische Fertigungsartikel ist vorzugsweise derart ausgebildet, dass er maschinenlesbare Anweisungen speichert, die bei ihrer Ausführung eine Maschine dazu veranlassen, wenigstens Folgendes auszuführen: Aufrufen eines Service, um mit einem Gerät in dem Prozesssteuerungssystem zu kommunizieren, wobei der Service eine generische Schnittstelle aufweist, die unabhängig von einem zur Implementierung des Prozesssteuerungssystems verwendeten Prozesssteuerungsnetzwerkprotokoll ist; Übersetzen des Service in eine oder mehrere Netzwerkoperationen, um den Service zu implementieren, wobei die Netzwerkoperationen für das zur Implementierung des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch sind; und Verwenden einer Netzwerkschnittstelle, die für das zur Implementierung des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch ist, zur Kommunikation mit dem Gerät gemäß der einen oder den mehreren Netzwerkoperationen.

[0007] Bevorzugt wird ein physischer Fertigungsartikel, wobei der Service als Servicevertrag durch eine Serviceschicht offengelegt wird, die eine Vielzahl von Services bereitstellt, jeder der Vielzahl der Services von dem zur Implementierung des Prozesssteuerungssystems verwendeten Prozesssteuerungsnetzwerkprotokoll unabhängig ist, jeder der Vielzahl von Services die Kommunikation mit Geräten in einer Vielzahl verschiedener Prozesssteuerungssysteme unterstützt, die unter Verwendung einer Vielzahl verschiedener Prozesssteuerungsnetzwerkprotokolle implementiert sind, und wobei die maschinenlesbaren Anweisungen bei Ausführung ferner herbeiführen, dass die Maschine den Service unter Verwendung von einer oder mehreren von einer Vielzahl von Nachrichten aufruft, die eine Serviceschnittstelle auf Nachrichtenbasis definieren, um den Service aufzurufen, wobei die Vielzahl von Nachrichten eine Vielzahl von Service-Nachrichtentypen und eine Vielzahl von Service-Datentypen umfasst, wobei die Vielzahl von Service-Nachrichtentypen und die Vielzahl von Service-Datentypen unabhängig von jeglichem Prozesssteuerungsnetzwerkprotokoll sind.

[0008] Bevorzugt wird weiterhin ein physischer Fertigungsartikel, wobei die maschinenlesbaren Anweisungen bei Ausführung ferner dazu führen, dass die Maschine den Service in eine oder mehrere Netzwerkoperationen zur Implementierung des Services übersetzt durch: Übersetzen des Service in eine Sequenz von Netzwerkoperationen, die für das Prozesssteuerungsnetzwerkprotokoll spezifisch sind; Lesen einer Gerätebeschreibung des Geräts, um die Sequenz von Netzwerkoperationen vorzubereiten;

und Pflegen eines Status der Sequenz von Netzwerkoperationen, um zu verfolgen, welche aus der Sequenz von Netzwerkoperationen durchgeführt worden sind, um den Service zu implementieren.

[0009] Bevorzugt wird außerdem ein physischer Fertigungsartikel, wobei die netzwerkspezifische Schnittstelle eine Netzwerk-Anwendungsprogrammchnittstelle umfasst, die mindestens eines aus einer Vielzahl von Netzwerk-Nachrichtentypen, einer Vielzahl von Netzwerk-Datentypen oder einem Objektwörterbuch umfasst und für das zur Implementierung des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch ist.

[0010] In einem anderen hier offenbarten Verfahren umfasst eine Vorrichtung zur Kommunikation mit Feldgeräten in Prozesssteuerungssystemen einen Prozessor, um ein serviceorientiertes Framework zur Kommunikation mit einer Vielzahl von Feldgeräten in einer Vielzahl von Prozesssteuerungssystemen zu implementieren, die eine Vielzahl unterschiedlicher Prozesssteuerungsnetzwerkprotokolle verwenden. Das serviceorientierte Framework umfasst in einigen Beispielen eine Serviceschicht, die eine Vielzahl von Services zur Kommunikation mit der Vielzahl von Feldgeräten implementiert, wobei jeder Service eine jeweilige generische Schnittstelle bereitstellt, die von jeglichen der Vielzahl von Prozesssteuerungsnetzwerkprotokollen unabhängig ist, die zur Implementierung der Vielzahl von Prozesssteuerungssystemen verwendet werden. In einigen Beispielen beinhaltet das serviceorientierte Framework auch eine Vielzahl von Übersetzungsschichten, wobei jede jeweilige Übersetzungsschicht jeden Service in der Vielzahl von Services in eine jeweilige Sequenz von Netzwerkoperationen übersetzt, um den jeweiligen Service zu implementieren, wobei die jeweilige Sequenz von Netzwerkoperationen für ein spezielles Prozesssteuerungsnetzwerkprotokoll spezifisch ist, das der jeweiligen Übersetzungsschicht zugeordnet ist. In einigen Beispielen umfasst das serviceorientierte Framework ferner eine Vielzahl von Netzwerkschichten, die jeweils der Vielzahl von Übersetzungsschichten zugeordnet sind, um eine Vielzahl von Netzwerkschnittstellen bereitzustellen, die für jedes der Vielzahl von Prozesssteuerungsnetzwerkprotokollen spezifisch sind, die zur Implementierung der Prozesssteuerungssysteme verwendet werden. Die beispielhafte Vorrichtung umfasst auch eine Schnittstelle zur kommunikativen Kopplung an eine auf einem Client-Gerät implementierte Anwendung, wobei die Serviceschicht durch den Prozessor implementiert wird.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

[0011] [Fig. 1](#) ist ein Blockdiagramm, welches eine beispielhafte Prozesssteuerungsumgebung einschließlich eines Feldgerät-Integrationservers zur

Implementierung eines serviceorientierten Frameworks zur Kommunikation mit Geräten in einem Prozesssteuerungssystem abbildet.

[0012] [Fig. 2](#) ist ein Blockdiagramm eines beispielhaften serviceorientierten Frameworks, das durch den Feldgerät-Integrationsserver von [Fig. 1](#) implementiert werden kann.

[0013] [Fig. 3](#) bildet beispielhafte Services ab, die von dem serviceorientierten Framework von [Fig. 2](#) zur Verfügung gestellt werden können.

[0014] [Fig. 4](#) ist ein Blockdiagramm eines beispielhaften Feldgerät-Integrationsservers, der zur Implementierung der Prozesssteuerungsumgebung von [Fig. 1](#) verwendet werden kann.

[0015] [Fig. 5](#) ist ein Blockdiagramm eines beispielhaften Client-Geräts, das zur Implementierung der Prozesssteuerungsumgebung von [Fig. 1](#) verwendet werden kann.

[0016] [Fig. 6](#) ist ein Flussdiagramm, das für ein beispielhaftes Verfahren zur Implementierung eines beispielhaften serviceorientierten Frameworks in der Prozesssteuerungsumgebung von [Fig. 1](#) repräsentativ ist.

[0017] [Fig. 7](#) ist ein Flussdiagramm, welches für ein beispielhaftes Verfahren zur Implementierung von Serviceschichtverarbeitung in dem serviceorientierten Frameworkverfahren von [Fig. 6](#) repräsentativ ist.

[0018] [Fig. 8](#) ist ein Flussdiagramm, welches für ein beispielhaftes Verfahren zur Implementierung von Übersetzungsschichtverarbeitung in dem serviceorientierten Frameworkverfahren von [Fig. 6](#) repräsentativ ist.

[0019] [Fig. 9](#) ist ein Flussdiagramm, welches für ein beispielhaftes Verfahren zur Implementierung von Netzwerkschichtverarbeitung in dem serviceorientierten Frameworkverfahren von [Fig. 6](#) repräsentativ ist.

[0020] [Fig. 10](#) ist ein Flussdiagramm, welches für einen beispielhaften Betrieb des serviceorientierten Frameworks von [Fig. 2](#) repräsentativ ist.

[0021] [Fig. 11](#) ist ein Blockdiagramm eines beispielhaften Verarbeitungssystems, das beispielhafte maschinenlesbare Anweisungen ausführen kann, die zur Implementierung von einigen oder allen der Prozesse der [Fig. 6–Fig. 10](#) verwendet werden, um das serviceorientierte Framework von [Fig. 2](#) in der Prozesssteuerungsumgebung von [Fig. 1](#) zu implementieren.

AUSFÜHRLICHE BESCHREIBUNG

[0022] Obwohl anschließend beispielhafte Verfahren, Vorrichtungen und Fertigungsartikel beschrieben werden, die unter anderen Komponenten Software und/oder Firmware einschließen, die auf Hardware ausgeführt werden, sei darauf hingewiesen, dass diese Beispiele lediglich illustrierend sind und nicht als einschränkend angesehen werden sollen. Es ist beispielsweise vorgesehen, dass beliebige oder alle der Hardware-, Software- und Firmware-Komponenten ausschließlich in Hardware, ausschließlich in Software oder in jeglicher Kombination aus Hardware und Software verkörpert sein können. Obwohl anschließend beispielhafte Verfahren, Vorrichtungen und Fertigungsartikel beschrieben werden, werden Fachleute mit durchschnittlichem Wissen daher leicht erkennen, dass die gegebenen Beispiele nicht der einzige Weg zur Implementierung derartiger Verfahren, Vorrichtungen und Fertigungsartikel sind. Obwohl die beispielhaften Verfahren, Vorrichtungen und Fertigungsartikel beispielsweise im Zusammenhang mit der Implementierung eines serviceorientierten Frameworks zur Kommunikation mit Geräten in einem Prozesssteuerungssystem beschrieben sind, sind die beispielhaften Verfahren, Vorrichtungen und Fertigungsartikel allgemeiner anwendbar und können zur Verwendung in jeglichem Automatisierungssystem, Chargenverarbeitungssystem, Fertigungssystem, industriellen Steuerungssystem, sicherheitsinstrumentierten System usw. implementiert werden.

[0023] Es können viele unterschiedliche Arten von Prozesssteuerungsnetzwerkprotokollen verwendet werden, um Prozesssteuerungssysteme zu implementieren. Beispiele solcher Prozesssteuerungsnetzwerkprotokolle beinhalten, ohne darauf beschränkt zu sein, das Foundation Fieldbus-Protokoll, das Profibus-Protokoll, das HART-Protokoll usw. Ein Prozesssteuerungssystem kann, wie bereits gesagt, eine oder mehrere Client-Anwendungen verwenden, die auf einer oder mehrere Client-Arbeitsstationen ausgeführt werden, um Informationen zu verarbeiten und mit Feldgeräten in dem Prozesssteuerungssystem zu interagieren. In Prozesssteuerungssystemen nach dem Stand der Technik zielen derartige Client-Anwendungen in der Regel jedoch auf das spezielle Prozesssteuerungsnetzwerkprotokoll, das zur Implementierung eines speziellen Prozesssteuerungssystems verwendet wird. In Prozesssteuerungssystemen nach dem Stand der Technik verwendete Client-Anwendungen müssen daher möglicherweise überarbeitet oder sogar neu geschrieben werden, wenn ein anderes (z. B. neues) Prozesssteuerungsnetzwerkprotokoll in dem Prozesssteuerungssystem verwendet wird, und diese Client-Anwendungen nach dem Stand der Technik sind über unterschiedliche Prozesssteuerungssysteme, die un-

terschiedliche Prozesssteuerungsnetzwerkprotokolle verwenden, möglicherweise nicht portierbar.

[0024] Die hier beschriebenen beispielhaften Verfahren, Vorrichtungen und Fertigungsartikel implementieren im Unterschied dazu ein serviceorientiertes Framework zur Kommunikation mit Geräten in einem Prozesssteuerungssystem, das einer Client-Anwendung (z. B. einer Prozesssteuerungsanwendung) Unabhängigkeit von jeglichem (jeglichen) speziellen Prozesssteuerungsnetzwerkprotokoll(en) ermöglicht, das bzw. die zur Implementierung des Prozesssteuerungssystems/der Prozesssteuerungssysteme verwendet wird, worin die Client-Anwendung eingesetzt wird. Das serviceorientierte Framework wird in einigen Beispielen im Kontext des Feldgeräte-Integrations-(FDI)-Standards implementiert. Ein hier beschriebenes beispielhaftes serviceorientiertes Framework umfasst eine Serviceschicht, eine oder mehrere Übersetzungsschichten und eine oder mehrere Netzwerkschichten, die jeweils der einen oder den mehreren Übersetzungsschichten zugeordnet sind. In einem derartigen Beispiel implementiert die Serviceschicht einen oder mehrere Prozesssteuerungsservices, um mit einem oder mehreren Feldgeräten in einem oder mehreren Prozesssteuerungssystemen zu kommunizieren. Jeder Service stellt eine generische Schnittstelle bereit, die unabhängig von jeglichem Prozesssteuerungsnetzwerkprotokoll ist, das zur Implementierung jedweden Prozesssteuerungssystems verwendet wird, in dem ein oder mehrere Feldgeräte laufen können. Die von der Serviceschicht implementierten Services sind in anderen Worten netzwerkprotokollneutral. Beispiele für Prozesssteuerungsservices, die durch das serviceorientierte Framework implementiert werden, umfassen, ohne darauf beschränkt zu sein: (a) einen Service zum Senden einer Nachricht an ein Feldgerät und zum Empfangen einer entsprechenden Antwort von dem Gerät; (2) einen Service zum Abonnieren veröffentlichter Daten, die von einem Feldgerät zurückgegeben werden; (3) einen Service zum Empfangen von Ereignissen, die von einem Feldgerät zurückgegeben werden; (4) einen Service zum Erhalten von Informationen, die das Prozesssteuerungssystem und eine Gruppe von Feldgeräten, die das Prozesssteuerungssystem implementieren, beschreiben; (5) einen Service zum Schreiben von Steuerungsparameterwerten in ein Feldgerät, usw.

[0025] Die eine oder mehreren Übersetzungsschichten, die in das beispielhafte serviceorientierte Framework eingeschlossen sind, zielen jeweils auf ein entsprechendes Prozesssteuerungsnetzwerkprotokoll. Eine spezielle Übersetzungsschicht übersetzt als solche jeden Service, der von der Serviceschicht bereitgestellt wird, in eine jeweilige Sequenz von Netzwerkoperationen, um den jeweiligen Service zu implementieren, wobei die jeweilige Sequenz von Netzwerkoperationen für ein spezielles Prozesssteuerungs-

netzwerkprotokoll spezifisch ist, das der speziellen Übersetzungsschicht zugeordnet ist. In einigen Beispielen liest eine Übersetzungsschicht eine Gerätebeschreibung für ein Feldgerät, mit dem ein Service kommunizieren soll, um eine spezielle Sequenz von Netzwerkoperationen vorzubereiten, die zur Implementierung des Services verwendet werden. In einigen Beispielen pflegt die Übersetzungsschicht zusätzlich oder alternativ einen Status einer speziellen Sequenz von Netzwerkoperationen, die zur Implementierung eines speziellen Services verwendet werden, um die Sequenz der Netzwerkoperationen zu verfolgen, wenn sie zur Implementierung des Services durchgeführt werden.

[0026] Die eine oder mehreren Netzwerkschichten, die in das beispielhafte serviceorientierte Framework eingeschlossen sind, zielen jeweils auf ein entsprechendes Prozesssteuerungsnetzwerkprotokoll. Eine spezielle Netzwerkschicht stellt beispielsweise eine Netzwerkschnittstelle für eine zugeordnete Übersetzungsschicht bereit, die für das spezielle Prozesssteuerungsnetzwerkprotokoll spezifisch ist, welches die Übersetzungsschicht unterstützt. Die Netzwerkschnittstelle ermöglicht den Austausch von Kommunikationstransaktionen (z. B. Befehlen, Antworten usw.) mit einem oder mehreren Feldgeräten, die das spezielle Prozesssteuerungsnetzwerkprotokoll unterstützen, gemäß der speziellen Sequenz von Netzwerkoperationen (z. B. Durchführen, Implementieren usw.), die durch die Übersetzungsschicht zur Implementierung des speziellen Services festgelegt wurden.

[0027] [Fig. 1](#) ist ein Blockdiagramm, das eine beispielhafte Prozesssteuerumgebung **100** darstellt, die einen Feldgeräte-Integrations-(FDI, field device integration)-Server **106** aufweist, um ein serviceorientiertes Framework zur Kommunikation mit Feldgeräten in einem beispielhaften Prozesssteuerungssystem **102** zu implementieren. Die beispielhafte Steuerungsumgebung **100** kann weitere Clients (nicht dargestellt) aufweisen, die kommunizierend an das Prozesssteuerungssystem **102** und/oder andere Steuerungssysteme (nicht dargestellt) gekoppelt sind.

[0028] Ein Client **104** (z. B. ein Endgerät, ein Arbeitsplatzrechner, ein PC usw.), das beispielhafte Steuerungssystem **102** und/oder der Feldgerätintegrations-(FDI)-Server **106** kommunizieren mithilfe eines FDI-Standards. Der FDI-Server **106** bildet eine Schnittstelle für Geräte an unterschiedlichen Netzen, die mit unterschiedlichen Standards arbeiten, darunter HART, Foundation Fieldbus und/oder Profibus. Im Allgemeinen stellt der FDI-Standard Spezifikationen bereit, die es Geräteherstellern und/oder Lieferanten ermöglichen, Werkzeugsätze zu erstellen, die von Kunden dazu verwendet werden können, Geräte einheitlich zu verwalten, und zwar unabhängig von dem Standard, dem die Geräte ursprünglich entspre-

chen und/oder nach dem sie gebaut wurden. FDI beinhaltet ein Gerätebeschreibungsverfahren in Textform, mit dem eine textbasierte Datei (z. B. eine XML-Datei) in einer standardmäßigen elektronischen Gerätebeschreibungssprache (EDDL, Electronic Device Description Language) ein Gerät, Verfahren, die das Gerät bereitstellt, Messungen und Vorrichtungparameter, die von dem Gerät unterstützt werden, Konfigurationsinformationen für das Gerät und/oder Interaktionen, die Benutzer mit dem Gerät haben können, beschreiben kann.

[0029] Wie anschließend detaillierter beschrieben wird, implementiert der FDI-Server **106** ein serviceorientiertes Framework, welches auf einer Nachrichtenbusarchitektur und einer serviceorientierten Architektur basiert. Die Nachrichtenbusarchitektur stellt ein Standardverfahren für Anwendungen (wie die Anwendung **120**, die auf dem Client **104** ausgeführt wird, wie anschließend detaillierter beschrieben wird) zur Verfügung, um mit dem serviceorientierten Framework zu kommunizieren. Die Nachrichtenbusarchitektur ähnelt einem Nachrichtenbus, der Nachrichten über Kommunikationskanäle sendet und empfängt. Die Nachrichtenbusarchitektur ermöglicht Anwendungen das Kommunizieren mit Prozesssteuerungs-Steuerungen, -Server, -Geräten und/oder anderen Anwendungen, ohne notwendigerweise alle spezifischen Details über die internen Operationen der Steuerungen, Server, Geräte und/oder anderen Anwendungen zu kennen.

[0030] Zusätzlich zu der Nachrichtenbusarchitektur bietet die serviceorientierte Architektur Prozesssteuerungsfunktionalität als Gruppe von Services. Unter einer derartigen serviceorientierten Architektur sind Netzwerkservices, die ein spezielles Gerät unterstützt, in einer Gerätebeschreibung zusammengefasst, die dem Gerät zugeordnet ist. Services werden über Schnittstellen aufgerufen, die durch nachrichtenbasierte Interaktionen implementiert werden, die als Teil des FDI-Standards implementiert sind. Weil Services durch Interaktionen auf Nachrichtenbasis aufgerufen werden, können die Orte des Serviceanforderers und des Service-Providers getrennt und damit innerhalb der Prozesssteuerungsumgebung **100** verteilt vorliegen.

[0031] Das Kombinieren der Nachrichtenbusarchitektur und der serviceorientierten Architektur zu dem serviceorientierten Framework kann mindestens unter einigen Bedingungen mehrere Vorteile bieten. Anwendungen können beispielsweise in unterschiedlichen Umgebungen laufen (z. B. kann eine Anwendung, die auf einem Microsoft Windows™ Host ausgeführt wird, unter Verwendung eines von mehreren Prozesssteuerungsnetzwerkprotokollen eine Schnittstelle zu einem Embedded-Gerät der Prozesssteuerung aufweisen). Ein weiterer möglicher Vorteil liegt darin, dass nicht alle Geräte alle Services unterstüt-

zen müssen. Ein weiterer möglicher Vorteil ist, dass Services die von dem tatsächlichen Prozesssteuerungsnetzwerk-(z. B. Bus)-Protokoll unabhängige Implementierung von Anwendungen auf höherer Ebene ermöglichen.

[0032] Der beispielhafte Client **104** und der beispielhafte FDI-Server **106** stehen über einen ersten Kommunikationsbus **108** in Verbindung. Der FDI-Server **106** verbindet den Kommunikationsbus **108** mit anderen Kommunikationsbussen **110**, **112**, die vom gleichen Typ wie der Kommunikationsbus **108** oder von anderen Typen sein können. Der FDI-Server **106** steht über den Kommunikationsbus **112** auch in Verbindung mit dem Steuerungssystem **102**. Der Client **104** kann somit über den FDI-Server **106** und die geeigneten Kommunikationsbusse **108** und **112** mit jeglichen der Geräte in dem Steuerungssystem **102** kommunizieren.

[0033] In einigen Beispielen können die Kommunikationsbusse **108–110** unter Verwendung eines Local Area Network-(LAN)-Protokolls (wie Ethernet), eines Wireless Fidelity-(Wi-Fi)-Protokolls (z. B. basierend auf der Standardfamilie 802.11 des Institute of Electrical and Electronic Engineers (IEEE), eines Mobilfunknetzes, des Internets usw. implementiert werden, und der Kommunikationsbus **112** kann konform mit dem Foundation Fieldbus-Protokoll, dem Profibus-Protokoll und/oder dem HART-Protokoll implementiert werden. Zusätzlich oder alternativ können ein oder mehrere der Kommunikationsbusse **108–110** konform mit dem Foundation Fieldbus-Protokoll, dem Profibus-Protokoll und/oder dem HART-Protokoll implementiert werden. Der Kommunikationsbus **112** kann zusätzlich oder alternativ konform mit einem LAN-Protokoll (wie Ethernet), einem Wi-Fi-Protokoll (z. B. basierend auf der IEEE 802.11 Standardfamilie), einem Mobilfunknetz, dem Internet usw. implementiert werden.

[0034] Das beispielhafte Steuerungssystem **102** kann jede Art von Fertigungsanlage, Prozessanlage, Automationsanlage und/oder jeden anderen Typ von Prozesssteuerungsstruktur oder -system einschließen. In einigen Beispielen kann das Steuerungssystem **102** mehrere Anlagen einschließen, die sich an unterschiedlichen Standorten befinden. Außerdem kann das Steuerungssystem **102** weitere Prozesssteuerungssysteme aufweisen, obwohl das beispielhafte Steuerungssystem **102** einem Prozesssteuerungssystem **114** zugeordnet ist.

[0035] Das beispielhafte Prozesssteuerungssystem **114** ist über einen Datenbus **118** kommunizierend an eine Steuerung **116** gekoppelt. Das Prozesssteuerungssystem **114** kann eine beliebige Anzahl von Feldgeräten (z. B. Eingabe- und/oder Ausgabegeräten) enthalten. Die Feldgeräte **119** können jeden Typ von Prozesssteuerungskomponenten ein-

schließen, der Eingaben empfangen, Ausgaben generieren und/oder einen Prozess steuern kann. Die Feldgeräte **119** können beispielsweise Eingabegeräte einschließen, beispielsweise Ventile, Pumpen, Ventilatoren, Heizer, Kühler und/oder Mischer, um einen Prozess zu steuern. Die Feldvorrichtungen **119** können zudem Ausgabevorrichtungen einschließen, wie beispielsweise Thermometer, Druckmessgeräte, Konzentrationsmessgeräte, Flüssigkeitspegelmessgeräte, Durchflussmessgeräte und/oder Dampfsensoren, um Teile eines Prozesses zu messen. Die Eingabevorrichtungen können Anweisungen von der Steuerung **116** empfangen, um einen speziellen Befehl auszuführen und eine Änderung des Prozesses zu bewirken. Die Ausgabevorrichtungen können ferner Prozessdaten, Umgebungsdaten und/oder Eingabevorrichtungsdaten messen und die gemessenen Daten als Prozesssteuerungsinformationen (z. B. Prozessdaten) an die Steuerung **116** übertragen. Diese Prozessdaten können die Werte von Variablen (z. B. gemessene Prozessvariablen und/oder gemessene Qualitätsvariablen) enthalten, die einer gemessenen Ausgabe von jedem Feldgerät **119** entsprechen.

[0036] Im dargestellten Beispiel aus [Fig. 1](#) kann die beispielhafte Steuerung **116** über den Datenbus **118** mit den Feldgeräten **119** im Prozesssteuerungssystem **114** kommunizieren. Der Datenbus **118** kann an Kommunikationskomponenten innerhalb des Prozesssteuerungssystems **114** gekoppelt sein. Die Kommunikationskomponenten können ferner E/A-Karten anschließen, um Daten von den Feldgeräten **119** zu empfangen und die Daten in ein Kommunikationsmedium umzuwandeln, das von der beispielhaften Steuerung **116** empfangen werden kann. Diese E/A-Karten können Daten von der Steuerung **108** in ein Datenformat umwandeln, das von den entsprechenden Feldgeräten **119** verarbeitet werden kann. In einem Beispiel kann der Datenbus **118** unter Verwendung des Fieldbus-Protokolls oder anderer Arten von verdrahteten und/oder drahtlosen Kommunikationsprotokollen (z. B. Profibus-Protokoll, HART-Protokoll, usw.) implementiert werden.

[0037] Die Steuerung **116** ist über eine beliebige drahtgebundene und/oder drahtlose Verbindung kommunizierend an den FDI-Server **106** gekoppelt. Die Verbindung kann in einigen Beispielen eine Firewall und/oder einen oder mehrere andere Sicherheitsmechanismen einschließen, um den Zugriff auf die Steuerung **116** zu beschränken. Die Steuerung **116** kann Prozessdaten an den FDI-Server **106** übertragen, wenn die Steuerung **116** die Prozessdaten vom Prozesssteuerungssystem **114** empfängt. In anderen Beispielen kann die Steuerung **116** in regelmäßigen Zeitintervallen (z. B. minütlich, stündlich, täglich usw.) Prozessdaten an den FDI-Server **106** übertragen. Alternativ kann der FDI-Server **106** Prozessdaten von der Steuerung **116** anfordern.

[0038] Nach dem Empfang der Prozessdaten speichert der beispielhafte FDI-Server **106** aus [Fig. 1](#) die Prozessdaten in einem Dateisystem (nicht dargestellt). Das Dateisystem kann in einer hierarchischen Weise angeordnet sein, wobei Verzeichnisse und/oder Unterverzeichnisse auf den Geräten innerhalb des Prozesssteuerungssystems **114** und/oder auf einer Routine (z. B. einer Anwendung und/oder einem Algorithmus) beruhen, die innerhalb der Steuerung **116** ausgeführt werden, um das Prozesssteuerungssystem **114** zu verwalten. In anderen Beispielen kann das Dateisystem von einem Bediener des Steuerungssystems **102** angeordnet werden. Die Prozessdaten können in einem Parameter innerhalb des zugeordneten Verzeichnisses und/oder Unterverzeichnisses gespeichert werden. In einigen Beispielen kann es sich bei dem Parameter um eine Variable handeln, die einer Routine zugeordnet ist, die auf der Steuerung **116** ausgeführt wird, oder einem Feldgerätausgang innerhalb der Prozesssteuerumgebung **100** zugeordnet ist. Der Parameter kann Metadaten einschließen, die den Typ der dem Parameter zugeordneten Prozessdaten beschreiben.

[0039] Der beispielhafte Client **104** kann einer Person zugeordnet sein, die autorisiert werden kann, um die der Prozesssteuerumgebung **100** zugeordneten Prozessdaten zu lesen, zu schreiben und/oder zu abonnieren. Der Client **106** kann auch Personal zugeordnet sein, das dem Steuerungssystem **102** zugeordnet ist und von einem entfernten Ort aus auf den externen Server **100** zugreifen kann. Der Client **104** kann unter Verwendung eines beliebigen drahtgebundenen und/oder drahtlosen Kommunikationsmediums (z. B. des Internets) über den FDI-Server **106** auf die Prozesssteuerumgebung **100** zugreifen.

[0040] Der beispielhafte FDI-Server **106** ist in der Lage, die Prozessdaten derart zu formatieren, dass die Prozessdaten von einem Benutzer einer Client-Anwendung **120**, die auf dem Client **104** ausgeführt wird, aufgerufen werden können. Das Beispiel aus [Fig. 1](#) zeigt die Client-Anwendung **120**, die Prozessdaten auf einer Schnittstelle **122** anzeigt. Die Client-Anwendung **120** kann beispielsweise eine Web-Client-Anzeigeanwendung umfassen. Der FDI-Server **106** kann Prozessdaten für eine Webserver-Anwendung formatieren, indem er eine Webseite erzeugt und/oder auf eine Vorlagenwebseite zugreift und die Datenfelder in der Webseite anordnet oder einbettet. Die Schnittstelle **122** kann über einen Webbrowser dann die Prozessdaten anzeigen, indem sie unter Verwendung von Hypertext Markup Language (HTML)-Anforderungen und Antworten auf die Webseite auf dem FDI-Server **106** zugreift. Alternativ kann der FDI-Server **106** die Prozessdaten für eine Client-Anzeigeanwendung formatieren, indem er an der Client-Anwendung **120** eine Webanwendung (z. B. ActiveX, Adobe Flash™ und/oder Silverlight™) initialisiert, die innerhalb eines Webbrowsers (z. B. der

Schnittstelle **122**) ausgeführt werden kann oder die für den Client **104** systemeigen ist (z. B. ein Windows®-Betriebssystemanwendung oder ein Anwendungs-Plug-in). In einigen Beispielen ordnet der FDI-Server **106** die Prozessdaten den entsprechenden Datenfeldern zu, bevor die Prozessdaten an die Client-Anwendung **120** übertragen werden. Nach Empfang der Prozessdaten erstellt (z. B. rendert) die Client-Anzeige-Anwendung **120** eine Anzeige innerhalb des Webbrowsers (z. B. in der Schnittstelle **122**), um die Prozessdaten in den entsprechenden Datenfeldern zu betrachten.

[0041] [Fig. 2](#) ist ein Blockdiagramm eines beispielhaften serviceorientierten Frameworks **200**, das mindestens teilweise durch den Feldgerätintegrations-Server **106** von [Fig. 1](#) implementiert werden kann. Das serviceorientierte Framework **200** befindet sich, wie in [Fig. 2](#) zu sehen ist, in Verbindung mit einer Client-Anwendung **202**, Gerätebeschreibungen **208** und einem oder mehreren Gerätenetzwerken **210**. Das serviceorientierte Framework **200** umfasst eine Serviceschicht **212**, eine Übersetzungsschicht **214**, eine Netzwerkschicht **216** und eine Sicherheitsschicht **218**.

[0042] Die beispielhafte Client-Anwendung **202** von [Fig. 2](#) ist eine Anwendung zum Konfigurieren von Geräten, eine Anwendung zur Durchführung von Gerätekalibrierung und -diagnostik und/oder eine Anwendung zum Lesen von Messwerten und Ereignissen von Geräten, die sich in einem oder mehreren Gerätenetzwerken befinden. Die Client-Anwendung **202** kann eine Benutzerschnittstelle (UI) umfassen, die gemäß einem Rich-Client-Modell oder einem Rich-Internet-Modell implementiert ist. In einem Rich-Client-Modell verwendet die UI beispielsweise einen Microsoft Windows™ oder ähnlichen Anwendungsstil. In einem Rich-Internet-Modell verwendet die UI einen Web-Client-Stil, wie eine Web-Browser-Schnittstelle oder eine Web-Browser-Plug-in-Schnittstelle. Die Client-Anwendung **202** umfasst, wie in [Fig. 2](#) dargestellt ist, Service-Verbraucher **206**, die Services von der Serviceschicht **212** anfordern und verbrauchen können. Die Client-Anwendung **202**, die der Anwendung **120** aus [Fig. 1](#) entsprechen kann, kann unter Verwendung von computerlesbaren Anweisungen implementiert werden, die beispielsweise auf dem beispielhaften Client **104** von [Fig. 1](#) und/oder dem beispielhaften Verarbeitungssystem **1100** aus [Fig. 11](#) ausgeführt werden. Die Client-Anwendung **202** kommuniziert mit dem Gerätenetzwerk/den Gerätenetzwerken **210**, das bzw. die dem Prozesssteuerungssystem **114** entsprechen kann bzw. können, über das serviceorientierte Framework **200**. Obwohl das serviceorientierte Framework **200** als durch den FDI-Server **106** implementiert beschrieben worden ist, können das beispielhafte serviceorientierte Framework **200** und/oder ausgewählte Teile des Frameworks **200** alternativ auf einem oder mehreren oder jeglicher Kom-

bination von Client(s) **104**, Server(n) **106**, Steuerung(en) **116** und/oder Gerät(en) **128–136** auf den Netzwerken **108**, **110** und **112** implementiert werden.

[0043] Die Serviceschicht **212** umfasst Service-schnittstellen **220**, Service-Nachrichtentypen **222**, Service-Datentypen **224**, Adapter **226** und Services **228**. Die durch die Serviceschicht **212** bereitgestellten beispielhaften Services **228** unterstützen den Zugriff auf die Gerätebeschreibungen **208** und das/die Gerätenetzwerk(e) **210**. Die Services **228** werden als Serviceverträge offengelegt, die es Anwendungen und/oder Geräten ermöglichen, die speziellen Services **228** anzufordern, die zur Ausführung erwünschter Fähigkeiten oder Funktionen benötigt werden. In einigen Beispielen sind Änderungen an Serviceverträgen, um neue Funktionen aufzunehmen, die durch einen Service implementiert werden, und/oder um neue Nachrichten und Datentypen aufzunehmen, möglicherweise nicht abwärtskompatibel mit den vorhandenen Serviceverträgen.

[0044] In dem dargestellten Beispiel erfolgt der Zugriff auf die Serviceschicht **212** über die Service-schnittstellen **220**, die den Services **228** Schnittstellen auf Nachrichtenbasis zur Verfügung stellen. Die Serviceschicht **212** verwaltet Serviceanfragen von Anwendungen und/oder Geräten und übersetzt Serviceverträge zur Verwendung durch die Client-Anwendungen **202**. Wenn Nachrichten zwischen einem Service und einem Serviceverbraucher hin und her gehen, können ein oder mehrere Adapter **226** verwendet werden, um die Nachrichten in Formate umzuwandeln, die der jeweilige Serviceverbraucher verstehen kann.

[0045] Der Zugriff auf die Serviceschicht **212** wird durch Richtlinien definiert. Richtlinien bieten Serviceverbrauchern einen Weg, um jegliche Verbindungs- und/oder Sicherheitsanforderungen zum Zugreifen auf Services und/oder jegliche anderen Details zu bestimmen, die mit Anfordern von Services verknüpft sind.

[0046] Die beispielhafte Übersetzungsschicht **214** übersetzt die netzwerkunabhängigen Services, die die Serviceschicht bereitstellt, um spezifische Prozesssteuerungsnetzwerkprotokolle (wie HART, Fieldbus, Profibus, die hier als Gerätenetzwerkprotokolle bezeichnet werden) zu unterstützen, die verwendet werden, um das/die Gerätenetzwerk(e) **210** (oder allgemeiner das/die Prozesssteuerungssystem(e)) zu implementieren, das das Gerät enthält bzw. die die Geräte enthalten, womit die Kommunikation gewünscht wird. Die Übersetzungsschicht **214** beinhaltet eine Anwendungsfassade **229**, um jeden netzwerkunabhängigen Service **228**, den die Serviceschicht **212** zur Verfügung stellt, in eine jeweilige Sequenz von Netzwerkoperationen **230** zu übersetzen, um den jeweiligen Service zu implementie-

ren. Die Sequenz der Netzwerkoperationen **230**, die durch die Anwendungsfassade **229** für einen speziellen Service **228** bestimmt wird, ist spezifisch für ein spezielles Prozesssteuerungsnetzwerkprotokoll **236**, das der Übersetzungsschicht **214** zugeordnet ist, und auch für die spezielle(n) Netzwerkkomponente(n) **231** (z. B. Feldgerät(e)), womit der Service **228** kommunizieren soll. Die Übersetzungsschicht **214** pflegt auch einen Status (z. B. Reihenfolge, Zeit usw.) der Ausführung der Sequenz von Netzwerkoperationen **230**, die zur Implementierung eines speziellen Service verwendet werden, um die Sequenz der Netzwerkoperationen **230** zu verfolgen, wenn sie zur Implementierung von Service **228** ausgeführt werden.

[0047] Die Übersetzungsschicht **214** empfängt Informationen über Geräte (z. B. die Netzwerkkomponenten **231**) aus den Gerätebeschreibungsdateien **208**, die beispielsweise beim Hochfahren des Systems gelesen werden. Die Gerätebeschreibungsdateien **208** enthalten die Informationen über Geräte in einer Sprache zur Beschreibung elektronischer Geräte (electronic device description language; EDDL) und/oder in einem üblichen Dateiformat **234**. EDDL-Dateien **232** repräsentieren beispielsweise Textbeschreibungen und Logik, die das Verhalten des Geräts definiert, welche von den jeweiligen Herstellern der Feldgeräte hergestellt werden, die in dem Prozesssteuerungssystem **102** aus [Fig. 1](#) enthalten sind. Die EDDL-Dateien **232** können an digitale Datenblätter der Geräte erinnern, die in dem Prozesssteuerungssystem **102** von [Fig. 1](#) enthalten sind. Die Flexibilität der EDDL-Elemente ist im Vergleich zu konventionellen Programmiersprachen begrenzt und für die Gerätebeschreibung spezifisch. Die Einfachheit der EDD-Sprache ermöglicht jedoch die einfache und effiziente Entwicklung von EDD-Gerätebeschreibungen, ist unabhängig von Hardware- und Betriebssystemplattformen, führt zu einer einheitlichen Philosophie zum Betrieb des Geräts und ist durch Interpretation sehr robust. Die EDDL-Technologie wird allgemein für Geräte mit niedriger bis durchschnittlicher Komplexität verwendet.

[0048] Das übliche Dateiformat **234** speichert Informationen zu Geräten und/oder Anwendungen, und die Dateien können zwischen Systemen, Werkzeugen, Anwendungen und/oder anderen Geräten ausgetauscht werden. In einigen Beispielen verwendet das übliche Dateiformat **234** ein Schema, das Extensible Markup Language (XML) einschließt, die flexibel ist und praktisch unbegrenzte Beschreibung mithilfe von Tags ermöglicht.

[0049] Die Netzwerkschicht **216** umfasst Datenzugriffsfunktionalität, um mit einem oder mehreren Gerätenetzwerken zu interagieren, die zur Implementierung des Prozesssteuerungssystems **102** von [Fig. 1](#) verwendet werden. In dem dargestellten Beispiel

umfasst die Netzwerkschicht **216** eine Netzwerk-Anwendungsprogrammierschnittstelle (API) **240**, die verschiedene Netzwerknachrichtentypen **242**, Netzwerkdatentypen **244** und möglicherweise ein oder mehrere Objektwörterbücher **246** unterstützt, die für ein spezielles Prozesssteuerungsnetzwerkprotokoll **236** spezifisch sind. Das beispielhafte Framework **200** kann mehrere Netzwerkschichten **216** und zugeordnete Übersetzungsschichten **214** einschließen, wobei jede Übersetzungsschicht **214** und zugeordnete Netzwerkschicht **216** für ein spezielles Gerätenetzwerkprotokoll **236** (auch als Prozesssteuerungssystemprotokoll bezeichnet, wie HART, Fieldbus und Profibus) spezifisch sind, das ein spezielles Gerätenetzwerk **210** (oder allgemeiner ein spezielles Prozesssteuerungssystem) implementiert. Das beispielhafte Framework **200** und die Netzwerkanwendungsschicht **216** erfordern daher keine Änderungen an diesen Protokollen.

[0050] Die Gerätenetze **210** verfügen über eine Schnittstelle mit dem/den Feldgeräte implementierenden Prozesssteuerungssystem(en) (z. B. dem Prozesssteuerungssystem **102** von [Fig. 2](#)) gemäß den jeweiligen Netzwerkprotokollen **236** und Netzwerkservices **238**, um dem beispielhaften Framework **200** die Fähigkeit zu verleihen, Prozesssteuerungsfeldgeräte zu konfigurieren, auf Gerätediagnostik zuzugreifen, Messwerte zu übertragen, Alarme und Ereignisse zu übertragen, und/oder andere Kommunikations- und Steuerungsfähigkeiten zu verleihen. Einige beispielhafte Fähigkeiten, die von dem serviceorientierten Framework **200** unterstützt werden, schließen Anforderungen und/oder Antworten, Veröffentlichungen und/oder Abonnieren, Übertragen von Ereignissen, Pflegen eines Verzeichnisses von Anwendungen und/oder Geräten und/oder Schreiben von Befehlen in Geräte ein. Das serviceorientierte Framework **200** verleiht der Client-Anwendung **202** über die Serviceschicht **212** Zugriff auf diese Fähigkeiten. Das serviceorientierte Framework **200** kann beispielsweise Services aufweisen, die für Anforderung/Antwort, Veröffentlichung/Abonnement, Ereignisse, Verzeichnis und Schreibfähigkeiten definiert sind.

[0051] In einigen Beispielen verwendet das serviceorientierte Framework **200** ein asynchrones Aufrufmodell zum Aufrufen von Services. Durch die Verwendung asynchroner Aufrufe wird das Blockieren von Client-Prozessen vermieden. In einigen Beispielen werden auch ein oder mehrere separate Verarbeitungs-Threads zur Durchführung von Hintergrundoperationen verwendet, um Daten von den Feldgeräten zu pollen (abzufragen). Länger laufende Operationen, wie das Lesen von Daten aus Feldgeräten, können somit mit einem Hintergrund-Thread und asynchroner Ausführung behandelt werden, um zu verhindern, dass die Implementierung eines Verarbeitungs-Threads die UI-Anwendung blockiert (wie es bei Anwendungen nach dem Stand der Tech-

nik beispielsweise passieren kann). In einigen Beispielen verwenden Serviceaufrufe Bindungen, die Sicherheitsinformationen einschließen, um die Authentifizierung der Client-Server-Verbindung zu ermöglichen. Internet Protocol Security (IPSec) und Secure Sockets Layer (SSL) können zum Sichern von Kommunikationskanälen verwendet werden, um sensible Informationen und Kommunikationskanäle zu schützen, wie den Kanal zwischen dem Client-Gerät **104** und dem FDI-Server **106**. Es kann auch mit Verschlüsselung zum Schutz von Daten sowie mit digitalen Signaturen gearbeitet werden, um Datenverfälschung zu erkennen, wobei die Möglichkeit besteht, diese Fähigkeiten ein- oder auszuschalten.

[0052] In einer beispielhaften Operation verwendet die Client-Anwendung **202** den FDI-Standard und ruft einen Service **228** mit einer generischen (z. B. netzwerkunabhängigen) Schnittstelle **220** auf, um das Ablesen des Drucks von einem Gerät anzufordern, wie einem Drucksender. Nach Erhalt der Anforderung überträgt die Serviceschicht **212** die Anforderung an die Übersetzungsschicht **214**. Die Übersetzungsschicht **214** generiert einen Befehl, der für das Gerätenetzwerk spezifisch ist, mit dem der Drucksender verbunden ist, und überträgt den Befehl über die Netzwerkanwendungsschicht **216**. Wenn das Gerätenetzwerk beispielsweise ein HART-Netzwerk ist, sendet die Client-Anwendung **202** einen Befehl an den Drucksender. Wenn das Gerätenetzwerk im Unterschied dazu ein Fieldbus-Netzwerk ist, liest die Client-Anwendung **202** einen Objekt-Wörterbucheintrag oder fordert dies an. Das Gerätenetzwerk transportiert die Anforderung zu dem Drucksender und gibt die Antwort von dem Gerät über die Netzwerkanwendungsschicht **216** an die Übersetzungsschicht **214** zurück. Die Übersetzungsschicht **214** wandelt die Antwort dann in ein generisches Antwortformat um, das für die Client-Anwendung **202** verständlich ist. In einigen anderen Beispielen erfordert die Client-Anwendung **202** ein spezifisches Format für die Antwort. In diesen Fällen kann die Client-Anwendung **202** anfordern, dass ein spezifischer Adapter **226** aufgerufen wird, wenn die Antwort an die Client-Anwendung **202** zurückgegeben wird.

[0053] Es ist in einigen Fällen vielleicht erforderlich, den Zugriff auf die Gerätenetze einzuschränken. Um diese Anforderung zu unterstützen, beinhaltet das serviceorientierte Framework **200** die Sicherheitsschicht **218**, um ein oder mehrere Authentifizierungsmodule **248** und/oder Autorisierungsmodule **250** zur Verfügung zu stellen. Die Implementierung eines Authentifizierungsmoduls **248**, um Anwendungen **202** mit dem Framework **200** zu authentifizieren, kann von der Art des verwendeten Service-Hosts abhängen. Das serviceorientierte Framework **200** ermöglicht somit die Integration von einer oder mehreren Sicherheitsschichten **218** in das Framework **200**. Wenn das serviceorientierte Framework **200**

beispielsweise in Internet Information Services (IIS) gehostet wird, kann die von IIS bereitgestellte Authentifizierungsunterstützung genutzt werden. Wenn der Service von einem Windows™ Service gehostet wird, kann eine Authentifizierung auf Nachrichtenbasis oder Transportbasis verwendet werden.

[0054] Das serviceorientierte Framework **200** bietet in einigen Beispielen Autorisierung des Benutzerzugriffs. In diesen Fällen kann ein Autorisierungsmodul **250** verwendet werden, um Zugriffsberechtigungen an Ressourcen für Benutzer, Gruppen und Rollen bereitzustellen.

[0055] In einigen Beispielen wird die Implementierung der Übersetzungsschicht **214** zwischen dem FDI-Server **106** und dem Client-Gerät **104** aufgeteilt, um beispielsweise die Leistung der Prozesssteuerungsanwendung **202** zu verbessern. In einem solchen Beispiel kann das Client-Gerät **104** die Bereiche der Transaktionsschicht **214** implementieren, die keine sensiblen Informationen enthalten, während der FDI-Server **106** die Bereiche der Transaktionsschicht **214** implementieren kann, die sensible Informationen enthalten.

[0056] In einigen Beispielen arbeiten der FDI-Server **106** und/oder das Client-Gerät **104** mit Caching, um beispielsweise die Leistung der Prozesssteuerungsanwendung **202** zu verbessern. Das Client-Gerät **104** kann beispielsweise Komponenten des serviceorientierten Frameworks **200** (wie z. B. den Service/die Services **228**) cachen, die von dem FDI-Server **106** erhalten werden. Der FDI-Server **106** kann zusätzlich oder alternativ die Gerätebeschreibung(en) **208** cachen, nachdem sie von beispielsweise einem externen Speicher, den Feldgeräten selbst usw. gelesen wurde(n). Alternativ oder zusätzlich kann der FDI-Server **106** Daten cachen, die von Feldgeräten zurückgegeben wurden, und/oder Daten, die je nach Bedarf an Feldgerät(e) geschrieben werden sollen, bis diese Daten an die Anwendung **202** zurückgegeben sind, oder in das Feldgerät/die Feldgeräte geschrieben werden.

[0057] In einigen Beispielen wird eine Programmzusammenstellung in dem serviceorientierten Framework **200** verwendet, damit Anwendungen einfacher ohne erneute Implementierung oder erneute Bereitstellung der gesamten Anwendung erweitert werden können. Dies kann durch Implementieren von Anwendungen aus zahlreichen Modulen und Design der Komponenten in loser Kopplung erreicht werden.

[0058] In einigen Beispielen unterstützt das serviceorientierte Framework **200** die Verwaltung von Ausnahmen und Protokollierung. Fehler wie Validierungsnachrichten können beispielsweise auf dem FDI-Server **106** protokolliert werden, um von Betriebspersonal und Überwachungssystemen genutzt

zu werden. Die Ausnahmenverwaltung kann sich sowohl auf asynchrone Ausnahmen als auch auf Ausnahmekoordination zwischen Client- und Server-Code beziehen. In Silverlight-Anwendungen steht in Bezug auf Protokollierung kein Zugriff auf das Dateisystem des Clients zur Verfügung, und die Ausführung auf dem Client und dem Server läuft oft asynchron. Daher werden einige oder alle der folgenden Informationen in Protokolldateien aufgenommen, die von der Client-Seite auf die Server-Seite transferiert werden: (a) der jedem protokollierten Eintrag zugeordnete Benutzer; (2) der mit dem Protokolleintrag verbundene Rechner; (3) ein Mechanismus zur Verknüpfung von verwandten Client- und Server-Protokolldateien; (4) kritische Fehler und Ausnahmen usw.

[0059] In einigen Beispielen wird/werden die Prozesssteuerungsanwendung(en) **202** implementiert, um die eingebauten Medienfähigkeiten der Plattform zu nutzen, wie das Client-Gerät **104**. Beispielsweise können folgende Richtlinien befolgt werden: (1) Das Design der Prozesssteuerungsanwendung **202** nutzt Grafikfähigkeiten, die das Client-Gerät **104** bereits zur Verfügung stellt, weil die Schnittstelle zu dem serviceorientierten Framework **200** unabhängig von jeglichem speziellen Prozesssteuerungsnetzwerkprotokoll ist, und (2) es werden eine systemeigene Grafik und/oder Vektorgrafik verwendet, um beispielsweise die Leistung von Anwendungen zu verbessern.

[0060] In einigen Beispielen unterstützt/unterstützen die Serviceschnittstelle(n) **220**, die die Schnittstelle zwischen der auf dem Client-Gerät **104** ausgeführten Anwendung und dem auf dem FDI-Server **106** ausgeführten serviceorientierten Framework **200** bereitstellt/bereitstellen, Client-Geräte, die stationär oder mobil sind, wie Rich-Desktops, intelligente Mobilgeräte, einfachere Mobilgeräte usw. Die Serviceschnittstelle(n) **220** kann bzw. können beispielsweise implementiert werden, um UI-Beschreibungen für sowohl Rich- als auch Mobilgerät-Schnittstellen zu unterstützen, die ermöglichen, dass die als Teil der Mobilgerät-Schnittstelle verwendete Navigation einer Rich-Benutzeroberfläche ähnlich ist.

[0061] In einigen Beispielen unterstützt das serviceorientierte Framework **200** Anwendung(en) **202**, die Benutzerschnittstellen im Web-Stil verwendet/verwenden. In derartigen Beispielen wird Datenanbindung verwendet, um Daten beispielsweise in tabellenförmigen oder mehrzeiligen Datenpräsentationen anzuzeigen. Dies kann den zur Implementierung der Präsentation von Gerätedaten erforderlichen Code reduzieren, wodurch die Entwicklung vereinfacht wird und Codierfehler reduziert werden. Datenanbindung ermöglicht auch automatische Synchronisation von Daten in unterschiedlichen Ansichten oder Formularen. Die Verwendung einer Zwei-Wege-Anbindung ermöglicht einem Benutzer ferner die Aktualisierung von Präsentationsdaten. Ereignisse an Navigations-

schaltflächen können zudem abgefangen werden, um unbeabsichtigte Navigation zu vermeiden, die von der Anwendung oder Webseite wegführt.

[0062] Die Übersetzungsschicht **214** speichert, wie bereits gesagt, den Status der Sequenz der Netzwerkoperationen **230**, die zur Implementierung eines speziellen Service **228** bestimmt wurden, den die Client-Anwendung **202** aufgerufen hat. Weil eine oder mehrere Anwendungen **202** mehrere Services **228** zur gleichen Zeit aufrufen kann bzw. können, ist die Übersetzungsschicht **214** so konfiguriert, dass die Statusinformationen für unterschiedliche Sequenzen von Netzwerkoperationen **230** für unterschiedliche Services **228**, die gleichzeitig ausgeführt werden, in isolierten Speicherbereichen (nicht gezeigt) gespeichert werden, damit der Status einer Sequenz durch eine andere Sequenz nicht beschädigt werden kann.

[0063] Das serviceorientierte Framework **200** schließt, wie bereits gesagt, die Sicherheitsschicht **218** ein, um ein oder mehrere Authentifizierungsmodule **248** und/oder Autorisierungsmodule **250** bereitzustellen. In einigen Beispielen unterstützt ein Authentifizierungsmodul **248** die Validierung der Anwendung **202**, des serviceorientierten Frameworks **200**, der Gerätebeschreibungsdateien **208** usw., um zu gewährleisten, dass diese Komponenten korrekt zusammenarbeiten. Die Client-seitige Validierung der Anwendung **202** kann beispielsweise verwendet werden, um zu gewährleisten, dass einem Benutzer die korrekte UI gezeigt wird. Serverseitige Validierung kann zur Validierung von Eingaben aus Datenquellen verwendet werden, wie von Daten, die von der Anwendung **202** und/oder den Feldgeräten bereitgestellt werden. Serverseitige Validierungsmechanismen können beispielsweise verwendet werden, um Daten zu beschränken, abzulehnen und/oder zu reinigen, wobei Eingabedaten auf Länge, Bereich, Format, Typ usw. geprüft werden. In einigen Beispielen (nicht gezeigt) wird isolierte Speicherung zur Speicherung von Validierungsregeln verwendet, die für eine spezielle Anwendung **202** und/oder ein spezielles Client-Gerät **104** spezifisch sind.

[0064] [Fig. 3](#) ist ein Blockdiagramm, das beispielhafte Services **228** abbildet, die von dem beispielhaften serviceorientierten Framework **200** aus [Fig. 2](#) bereitgestellt werden können. Die beispielhaften Services von [Fig. 3](#) schließen einen beispielhaften Anforderungs-Antwort-Service **304** für eine Anwendung (z. B. die Client-Anwendung **202**) ein, um eine Nachricht an ein Feldgerät zu senden und eine entsprechende Antwort von dem Gerät zu empfangen. Die beispielhaften Services **228** aus [Fig. 3](#) umfassen auch einen beispielhaften Veröffentlichungs-Abonnement-Service **308** für eine Anwendung (z. B. die Client-Anwendung **202**), um die von einem Feldgerät zurückgegebenen veröffentlichten Daten zu abonnieren. Die beispielhaften Services **228** aus [Fig. 3](#) umfassen

ferner einen beispielhaften Ereignisservice **312** für eine Anwendung (z. B. die Client-Anwendung **202**) ein, um die von einem Feldgerät zurückgegebenen Ereignisse zu erhalten. Die beispielhaften Services **228** aus [Fig. 3](#) schließen zusätzlich einen beispielhaften Verzeichnisservice **316** für eine Anwendung (z. B. die Client-Anwendung **202**) ein, um Informationen zu erhalten, die ein oder mehrere Gerätenetze **210** (oder allgemeiner z. B. ein oder mehrere zugeordnete Prozesssteuerungssysteme **102**) und eine Gruppe von Feldgeräten beschreiben, die durch das eine oder die mehreren Gerätenetze **210** miteinander verbunden sind (und so z. B. das eine oder die mehreren zugeordneten Prozesssteuerungssysteme **102** implementieren). Die beispielhaften Services **228** aus [Fig. 3](#) schließen ferner einen beispielhaften Schreibservice **320** für eine Anwendung (z. B. die Client-Anwendung **202**) ein, um Steuerungsparameterwerte in ein Feldgerät zu schreiben. Der Schreibservice **320** kann in einigen Beispielen verschiedene Zugriffbeschränkungen unterstützen (wie z. B. um Schreibvorgänge nur auf autorisierte Benutzer zu beschränken).

[0065] [Fig. 4](#) ist ein Blockdiagramm einer beispielhaften Implementierung des FDI-Servers **106** aus [Fig. 1](#). Der FDI-Server **106** des abgebildeten Beispiels umfasst eine beispielhafte Anwendungsschnittstelle **404**, mit der Schnittstelle zu einer Anwendung (z. B. der Client-Anwendung **202** aus [Fig. 2](#) und/oder der Client-Anwendung **120** aus [Fig. 1](#)). Die Anwendungsschnittstelle **404** kann beispielsweise die auf Nachrichten basierenden Serviceschnittstellen **220** implementieren, die durch die Service-Nachrichtentypen **222** und die Service-Datentypen **224** definiert werden, die unabhängig von jeglichem Prozesssteuerungsnetzwerkprotokoll sind. Die Anwendungsschnittstelle **404** kann darüber hinaus so implementiert werden, dass sie kommunikativ mit Anwendungen gekoppelt ist, die auf dem FDI-Server **106** selbst und/oder auf anderen Client-Geräten (wie z. B. dem Client **104** aus [Fig. 1](#)) ausgeführt werden.

[0066] Der FDI-Server **106** aus [Fig. 4](#) umfasst auch einen beispielhaften Prozessor **408** zur Ausführung maschinenlesbarer Anweisungen, um einen Teil oder alles des serviceorientierten Frameworks **200** aus [Fig. 2](#) zu implementieren. Der Prozessor **408** kann mit jedem Typ von Prozessor oder Prozessoren implementiert werden, wie einem oder mehreren der Prozessoren **1102**, die in das Verarbeitungssystem **1100** von [Fig. 11](#) eingeschlossen ist, die hier nachfolgend detaillierter beschrieben werden.

[0067] Der FDI-Server **106** von [Fig. 4](#) schließt ferner eine beispielhafte Netzwerkschnittstelle **412** als Schnittstelle zu einem oder mehreren Gerätenetzen ein, die ein oder mehrere Prozesssteuerungssysteme implementieren. Die Netzwerkschnittstelle **412** kann als solche ein oder mehrere der oben

beschriebenen Prozesssteuerungsnetzwerkprotokolle **236** unterstützen.

[0068] [Fig. 5](#) ist ein Blockdiagramm einer beispielhaften Implementierung des Client-Geräts **104** von [Fig. 1](#). Das Client-Gerät **104** des abgebildeten Beispiels umfasst eine beispielhafte Client-Schnittstelle **504**, die als Schnittstelle zu anderen Clients dient, die andere Prozesssteuerungsanwendungen (wie z. B. andere Client-Anwendungen **202** aus [Fig. 2](#) und/oder andere Client-Anwendung **120** aus [Fig. 1](#)) ausführen. Die Client-Schnittstelle **404** kann beispielsweise jede Art von Kommunikationsprotokoll und/oder Interprozess-Nachrichtenservice zur Kommunikation zwischen Anwendungen verwenden, die auf dem gleichen oder unterschiedlichen Geräten laufen.

[0069] Das Client-Gerät **104** aus [Fig. 5](#) umfasst auch einen beispielhaften Prozessor **508** zur Ausführung maschinenlesbarer Anweisungen, um einige oder mehrere Anwendungen **202** aus [Fig. 2](#) zu implementieren, die eine Schnittstelle zu dem serviceorientierten Framework **200** aus [Fig. 2](#) haben. Der Prozessor **508** kann mit jedem Typ von Prozessor oder Prozessoren implementiert werden, wie einem oder mehreren der Prozessoren **1102**, die in das Verarbeitungssystem **1100** von [Fig. 11](#) eingeschlossen ist, die hier nachfolgend detaillierter beschrieben werden.

[0070] Das Client-Gerät **104** aus [Fig. 5](#) umfasst ferner eine beispielhafte Serviceschnittstelle **512**, die als Schnittstelle zu dem FDI-Server **106** aus [Fig. 1](#) und/oder [Fig. 4](#) dient, der das serviceorientierte Framework **200** implementiert. Die Serviceschnittstelle **512** ist als solche das Gegenstück zu der Anwendungsschnittstelle **404** von [Fig. 4](#) und kann die Serviceschnittstellen **220** auf Nachrichtenbasis implementieren, die von den Service-Nachrichtentypen **222** und den Service-Datentypen **224** des serviceorientierten Frameworks **200** definiert sind und von jeglichem Prozesssteuerungsnetzwerkprotokoll unabhängig sind.

[0071] Obwohl in den [Fig. 4](#) beziehungsweise [Fig. 5](#) beispielhafte Arten der Implementierung von FDI-Server **106** und dem Client-Gerät **104** aus [Fig. 1](#) abbildet worden sind, können ein oder mehrere der in den [Fig. 4](#)–[Fig. 5](#) abgebildeten Elemente, Prozesse und/oder Geräte kombiniert, unterteilt, umgeordnet, weggelassen, eliminiert und/oder anderweitig implementiert werden. Die beispielhafte Anwendungsschnittstelle **404**, der beispielhafte Prozessor **408**, die beispielhafte Netzwerkschnittstelle **412**, die beispielhafte Client-Schnittstelle **504**, der beispielhafte Prozessor **508**, die beispielhafte Serviceschnittstelle **512** und/oder allgemeiner gesagt der beispielhafte FDI-Server **106** aus [Fig. 4](#) und/oder das beispielhafte Client-Gerät aus [Fig. 5](#) können ferner mittels Hardware, Software, Firmware und/oder jegliche Kombination aus Hardware, Software und/oder Firm-

ware implementiert werden. Jegliche der beispielhaften Anwendungsschnittstelle **404**, des beispielhaften Prozessors **408**, der beispielhaften Netzwerkschnittstelle **412**, der beispielhaften Client-Schnittstelle **504**, des beispielhaften Prozessors **508**, der beispielhaften Serviceschnittstelle **512** und/oder allgemeiner gesagt des beispielhaften FDI-Servers **106** und/oder des beispielhaften Client-Geräts **104** können somit durch einen oder mehrere Schaltkreise, programmierbare Prozessor(en), anwendungsspezifische integrierte Schaltkreise (ASIC(s)), programmierbare Logikvorrichtungen (PLD(s)) und/oder feldprogrammierbare Logikvorrichtungen (FPLD(S)) usw. implementiert werden. Wenn jegliche der angefügten Vorrichtungsansprüche so gelesen werden, dass eine reine Software- und/oder Firmwareimplementierung abgedeckt wird, ist mindestens eine(r/s) des beispielhaften FDI-Servers **106**, des beispielhaften Client-Geräts **104**, der beispielhaften Anwendungsschnittstelle **404**, des beispielhaften Prozessors **408**, der beispielhaften Netzwerkschnittstelle **412**, der beispielhaften Client-Schnittstelle **504**, des beispielhaften Prozessors **508** und/oder der beispielhaften Serviceschnittstelle **512** hier ausdrücklich so definiert, dass ein physisches computerlesbares Medium eingeschlossen ist, wie ein Speicher, eine DVD, CD usw., worauf diese Software und/oder Firmware gespeichert ist. Der beispielhafte FDI-Server **106** aus [Fig. 4](#) und/oder das beispielhafte Client-Gerät **104** aus [Fig. 5](#) kann ferner ein oder mehrere Elemente, Prozesse und/oder Geräte zusätzlich zu oder anstelle der in den [Fig. 4–Fig. 5](#) dargestellten umfassen und/oder kann mehr als eines von beliebigen oder allen der dargestellten Elemente, Prozesse und Geräte umfassen.

[0072] In [Fig. 6–Fig. 10](#) sind Flussdiagramme gezeigt, die für beispielhafte Prozesse repräsentativ sind, die zur Implementierung der beispielhaften Prozesssteuerungsumgebung **100**, des beispielhaften FDI-Servers **106**, des beispielhaften Client-Geräts **104**, der beispielhaften Anwendungsschnittstelle **404**, des beispielhaften Prozessors **408**, der beispielhaften Netzwerkschnittstelle **412**, der beispielhaften Client-Schnittstelle **504**, des beispielhaften Prozessors **508**, der beispielhaften Serviceschnittstelle **512** und/oder allgemeiner gesagt des serviceorientierten Frameworks **200** ausgeführt werden können. In diesen Beispielen kann der durch jedes Flussdiagramm repräsentierte Prozess durch ein oder mehrere Programme implementiert werden, die maschinenlesbare Anweisungen zur Ausführung durch einen Prozessor umfassen, wie den Prozessor **1102**, der in dem beispielhaften Verarbeitungssystem **1100** gezeigt ist, das anschließend im Zusammenhang mit [Fig. 11](#) erörtert wird. Alternativ kann das gesamte Programm oder die gesamten Programme und/oder Teile davon, die ein oder mehrere der durch die Flussdiagramme der [Fig. 6–Fig. 10](#) wiedergegebenen Prozesse implementieren, durch ein anderes Gerät als

den Prozessor **1102** implementiert werden (wie z. B. eine Steuerung und/oder jedes andere geeignete Gerät) und/oder in Firmware oder dedizierter Hardware ausgeführt werden (durch ASIC, PLS, FPLD, diskrete Logik usw. implementiert werden). Ein oder mehrere der durch die Flussdiagramme der [Fig. 6–Fig. 10](#) wiedergegebenen Prozesse oder ein oder mehrere Teile davon können auch manuell implementiert werden. Obwohl die beispielhaften Prozesse mit Bezugnahme auf die in den [Fig. 6–Fig. 10](#) beschriebenen Flussdiagramme beschrieben worden sind, können viele andere Techniken zur Implementierung der hier beschriebenen beispielhaften Verfahren und Vorrichtung(en) alternativ verwendet werden. Beispielsweise kann in Bezug auf die in den [Fig. 6–Fig. 10](#) wiedergegebenen Flussdiagramme die Reihenfolge der Ausführung der Blöcke geändert werden, und/oder einige der beschriebenen Blöcke können geändert, eliminiert, kombiniert und/oder in mehrere Blöcke unterteilt werden.

[0073] Wie bereits erwähnt können die beispielhaften Prozesse der [Fig. 6–Fig. 10](#) unter Verwendung kodierter Anweisungen (z. B. computerlesbarer Anweisungen) implementiert werden, die auf einem physischen computerlesbaren Medium gespeichert sind, wie auf einer Festplatte, einem Flash-Speicher, einem Festwertspeicher (ROM), einer CD, DVD, einem Cache, einem Direktzugriffsspeicher (RAM) und/oder jeglichem anderen Speichermedium, in dem Informationen für eine beliebige Dauer (z. B. längere Zeiträume, permanent, kurze Zeiträume, zum vorübergehenden Puffer und/oder Cachen der Informationen) gespeichert werden. Der Begriff physisches computerlesbares Medium ist hier ausdrücklich so definiert, dass jede Art von computerlesbarer Speicherung eingeschlossen ist und weitergeleitete Signale ausgeschlossen sind. Zusätzlich oder alternativ können die beispielhaften Prozesse der [Fig. 6–Fig. 10](#) unter Verwendung codierter Anweisungen (z. B. computerlesbarer Anweisungen) implementiert werden, die auf einem nicht-flüchtigen computerlesbaren Medium gespeichert sind, wie auf einem Flash-Speicher, einem ROM, einer CD, DVD, einem Cache, einem Direktzugriffsspeicher (RAM) und/oder jeglichem anderen Speichermedium, in dem Informationen für eine beliebige Dauer (z. B. längere Zeiträume, permanent, kurze Zeiträume, zum vorübergehenden Puffer und/oder Cachen der Informationen) gespeichert werden. Der Begriff nicht-flüchtiges computerlesbares Medium ist hier ausdrücklich so definiert, dass jede Art von computerlesbarem Medium eingeschlossen ist und weitergeleitete Signale ausgeschlossen sind. Die Begriffe "computerlesbar" und "maschinenlesbar" werden hier als äquivalent angesehen, sofern nichts anderes angegeben ist.

[0074] In [Fig. 6](#) ist ein beispielhafter Prozess **600** abgebildet, der ausgeführt werden kann, um das beispielhafte serviceorientierte Framework **200** aus [Fig. 2](#) zu

implementieren. Der beispielhafte Prozess **600** kann in festgelegten Intervallen, bezogen auf das Auftreten eines festgelegten Ereignisses usw. oder jeglicher Kombination davon ausgeführt werden. Der Prozess **600** aus [Fig. 6](#) wird aus der Perspektive beschrieben, dass er von dem FDI-Server **106** der [Fig. 1](#) und/oder [Fig. 4](#) ausgeführt wird, um die Prozesssteuerungsanwendung **202** zu unterstützen, die von dem Client-Gerät **104** der [Fig. 1](#) und/oder [Fig. 5](#) ausgeführt wird. In Bezug auf die vorhergehenden Figuren beginnt der Prozess **600** aus [Fig. 6](#) somit mit der Ausführung in Block **605**, wobei das Client-Gerät **104** eine Prozesssteuerungsanwendung **202** ausführt, die mit einem oder mehreren Feldgeräten in einem oder mehreren Prozesssteuerungssystemen unter Verwendung von einem oder mehreren netzwerkunabhängigen Services **228** kommunizieren soll (auch als netzwerkneutrale Services, gerätunabhängige Services, gerätneutrale Services usw. bezeichnet), die durch das serviceorientierte Framework **200** bereitgestellt werden. In Block **605** kann die Prozesssteuerungsanwendung beispielsweise einen oder mehrere von dem Anforderung-Antwort-Service **304**, dem Veröffentlichungs-Abonnement-Service **308**, dem Ereignisservice **312**, dem Verzeichnisservice **316** und/oder dem Schreibservice **320**, alle aus [Fig. 3](#), verwenden, um mit dem einen oder den mehreren Feldgeräten zu kommunizieren.

[0075] In Block **610** ermittelt das serviceorientierte Framework **200**, das auf dem FDI-Server **106** ausgeführt wird, ob eine Zugriffsrichtlinie verwendet wird. Falls eine Zugriffsrichtlinie verwendet wird (Block **610**), authentifiziert in Block **615** das Authentifizierungsmodul **248** des serviceorientierten Frameworks **200** die Prozesssteuerungsanwendung **202** zur Verwendung mit dem serviceorientierten Framework **200**. In Block **620** autorisiert zudem das Autorisierungsmodul **250** des serviceorientierten Frameworks **200** einen Benutzer der Prozesssteuerungsanwendung **202** und stellt den von dem serviceorientierten Framework **200** bereitgestellten Services **228** benutzerspezifische Zugriffsebenen (z. B. nur Lesen, Lesen und Schreiben, kein Zugriff, Administratorzugriff usw.) zur Verfügung.

[0076] Die Prozesssteuerungsanwendung **202** hat in Block **625** eine Schnittstelle zu der netzwerkunabhängigen Serviceschicht **212** des serviceorientierten Frameworks **200**. Die Serviceschicht **212** wird als netzwerkunabhängig bezeichnet, weil sie Service(s) **228** zur Kommunikation mit Feldgeräten in Prozesssteuerungssystemen zur Verfügung stellt, wobei die Services nicht von irgendeinem speziellen Prozesssteuerungsnetzwerkprotokoll abhängen. Ein beispielhafter Prozess, der zur Implementierung der Verarbeitung in Block **625** verwendet werden kann, ist in [Fig. 7](#) abgebildet und wird anschließend detaillierter beschrieben.

[0077] Die netzwerkunabhängige Serviceschicht **212** des serviceorientierten Frameworks **200** hat in Block **630** eine Schnittstelle zu einer netzwerkabhängigen Übersetzungsschicht **214** des serviceorientierten Frameworks **200**. Die Übersetzungsschicht **214** wird als netzwerkabhängig bezeichnet, weil jede in das serviceorientierte Framework **200** eingeschlossene Übersetzungsschicht **214** die von der Serviceschicht **212** bereitgestellten netzwerkunabhängigen Services **228** in jeweilige Sequenzen von Netzwerkkoperationen übersetzt, die für ein spezielles Prozesssteuerungsnetzwerkprotokoll spezifisch sind (z. B. von diesem abhängen). Ein beispielhafter Prozess, der zur Implementierung der Verarbeitung in Block **630** verwendet werden kann, ist in [Fig. 8](#) abgebildet und wird anschließend detaillierter beschrieben.

[0078] Die netzwerkunabhängige Übersetzungsschicht **214** des serviceorientierten Frameworks **200** hat in Block **635** eine Schnittstelle zu einer jeweiligen netzwerkabhängigen Netzwerkschicht **216** des serviceorientierten Frameworks **200**. Die Netzwerkschicht **216** wird als netzwerkabhängig bezeichnet, weil jede in das serviceorientierte Framework **200** eingeschlossene Netzwerkschicht **216** die Sequenzen der Netzwerkkoperationen bereitstellt, die durch seine jeweilige Übersetzungsschicht **214** mit einer Netzwerkschnittstelle ermittelt wurden, die auf das spezielle Prozesssteuerungsnetzwerkprotokoll zielt, welches die Übersetzungsschicht **214** unterstützt. Ein beispielhafter Prozess, der zur Implementierung der Verarbeitung in Block **635** verwendet werden kann, ist in [Fig. 9](#) abgebildet und wird anschließend detaillierter beschrieben.

[0079] In Block **640** werden eine oder mehrere Netzwerkschichten **216** des serviceorientierten Frameworks **200** verwendet, um eine Schnittstelle zu einem oder mehreren speziellen Gerätenetzwerken **210** zu bilden, die ein oder mehrere Prozesssteuerungssysteme implementieren, die das eine oder die mehreren Feldgeräte enthalten, mit denen die Prozesssteuerungsanwendung **202** kommunizieren soll. Die Ausführung des beispielhaften Prozesses **600** endet dann.

[0080] In [Fig. 7](#) ist ein beispielhafter Prozess **625** abgebildet, der zur Implementierung der beispielhaften Serviceschichtverarbeitung in Block **625** aus [Fig. 6](#) ausgeführt werden kann. In Bezug auf die vorhergehenden Figuren beginnt der Prozess **625** aus [Fig. 7](#) mit der Ausführung in Block **705**, in dem die Serviceschicht **212** des serviceorientierten Frameworks **200** den/die verfügbaren netzwerkunabhängigen Service(s) **228** in Form von Serviceverträgen offenlegt, worauf Prozesssteuerungsanwendung(en) **202** zugreifen kann bzw. können. In Block **710** ruft eine Prozesssteuerungsanwendung **202** einen oder mehrere der Services **228** unter Verwendung von einer oder mehreren der Serviceschnittstellen **220** auf, wo-

bei jede Serviceschnittstelle **220** einem jeweiligen Service **228** zugeordnet ist. Jede Serviceschnittstelle **220** ist beispielsweise eine Serviceschnittstelle auf Nachrichtenbasis, die netzwerkunabhängige Nachrichten verwendet, die unter Verwendung der Service-Nachrichtentypen **222** und der Service-Datentypen **224** definiert sind, die durch das serviceorientierte Framework **220** spezifiziert werden.

[0081] In Block **715** bestimmt die Serviceschicht **212**, ob eine durch ein Feldgerät in Block **710** in Reaktion auf die Prozesssteuerungsanwendung **202**, die einen speziellen Service **228** aufruft, an die Prozesssteuerungsanwendung **202** zurückgegebene Antwort (z. B. Gerätedaten, Ereignisse usw.) auf ein anwendungsspezifisches Format angepasst werden muss. Wenn eine anwendungsspezifische Anpassung der Antwort erforderlich ist (Block **715**), ruft die Serviceschicht **212** einen ausgewählten Adapter **226** auf, um die Antwortdaten auf ein Format anzupassen, das für die Prozesssteuerungsanwendung **202** spezifisch ist. Die Ausführung des beispielhaften Prozesses **625** endet dann.

[0082] In **Fig. 8** ist ein beispielhafter Prozess **630** abgebildet, der zur Implementierung der beispielhaften Übersetzungsschichtverarbeitung in Block **630** aus **Fig. 6** ausgeführt werden kann. In Bezug auf die vorhergehenden Figuren beginnt der Prozess **630** aus **Fig. 8** mit der Ausführung in Block **805**, worin eine spezielle Übersetzungsschicht **214** des serviceorientierten Frameworks **200**, die ein Prozesssteuerungsnetzwerkprotokoll unterstützt, welches zur Kommunikation mit einem speziellen Feldgerät verwendet wird, durch die Serviceschicht **212** des serviceorientierten Frameworks **200** aufgerufen wird. In Block **805** wird zudem die in die spezielle Übersetzungsschicht **214** eingeschlossene Anwendungsfassade **229** verwendet, um die Serviceoperationen, die einen speziellen von der Serviceschicht **212** bereitgestellten Service **228** implementieren, in eine jeweilige Sequenz von einer oder mehreren Netzwerkoperationen **230** zu übersetzen, um den Service **228** zu implementieren. Die einen speziellen Service **228** implementierenden Serviceoperationen sind, wie bereits beschrieben, unabhängig von jeglichem Prozesssteuerungsnetzwerkprotokoll, wohingegen die Sequenz der Netzwerkoperationen **230** spezifisch für ein spezielles Prozesssteuerungsnetzwerkprotokoll ist, das zur Implementierung eines Gerätenetzwerks **210** (und z. B. allgemeiner eines Prozesssteuerungssystems) verwendet wird, das ein spezielles Feldgerät enthält, mit dem der Service **228** kommunizieren soll.

[0083] Die Übersetzungsschicht **214** verwendet in Block **810** die Gerätebeschreibungsdateien **208** (z. B. im EDDL-Format **232** und/oder im üblichen Dateiformat **234**), um die Sequenzen der Netzwerkoperationen **230** zur Interaktion mit einem oder mehreren spe-

zifischen Feldgeräten vorzubereiten, mit denen der Service **228** kommunizieren soll.

[0084] In Block **815** führt die Übersetzungsschicht **214** die Sequenz der Netzwerkoperationen **230** (z. B. durch Aufrufen einer entsprechenden Netzwerkschicht **216**) durch und pflegt einen Status der Sequenz der Netzwerkoperationen **230** während der Durchführung. Der von der Übersetzungsschicht **214** in Block **810** gepflegte Status kann verwendet werden, um zu verfolgen, welche der Sequenz der Netzwerkoperationen **230** durchgeführt wurde und welche zu einer gegebenen Zeit noch durchgeführt werden muss, um den Service **228** zu implementieren. Nachdem die Sequenz der Netzwerkoperationen **230** durchgeführt worden ist, endet die Ausführung des beispielhaften Prozesses **630**.

[0085] In **Fig. 9** ist ein beispielhafter Prozess **635** abgebildet, der zur Implementierung der beispielhaften Netzwerkschichtverarbeitung in Block **635** aus **Fig. 6** ausgeführt werden kann. In Bezug auf die vorhergehenden Figuren beginnt der Prozess **635** aus **Fig. 9** mit der Ausführung in Block **905**, worin eine spezielle Netzwerkschicht **216** aufgerufen wird, um die Kommunikation einer Sequenz von Netzwerkoperationen, die durch eine entsprechende Übersetzungsschicht **214** bestimmt wird, über eine Schnittstelle zu der entsprechenden Netzwerk-API **240** zu ermöglichen, die für das Gerätenetzwerk **210** definiert ist, welche das/die Feldgerät(e) enthält, womit die Sequenz der Netzwerkoperationen **230** durchgeführt werden soll. Die in Block **905** verwendete Netzwerk-API unterstützt beispielsweise verschiedene Netzwerk-Nachrichtentypen **242**, Netzwerk-Datentypen **244** und möglicherweise ein oder mehrere Objektwörterbücher **246**, die für ein spezielles Prozesssteuerungsnetzwerkprotokoll **236** spezifisch sind, das das Gerätenetzwerk **210** verwendet, welches das/die Feldgerät(e) enthält, womit die Sequenz der Netzwerkoperationen **230** durchgeführt werden soll. Nachdem der Prozess den Block **905** abgeschlossen hat, endet die Ausführung des beispielhaften Prozesses **635**.

[0086] In **Fig. 10** ist ein beispielhafter Prozess **1000** dargestellt, der einen beispielhaften Betrieb des Service-Frameworks **200** aus **Fig. 2** abbildet, um einer Client-Anwendung **202** die Kommunikation mit einem Feldgerät in einem Prozesssteuerungssystem **102** zu ermöglichen. In Bezug auf die vorhergehenden Figuren ruft die Client-Anwendung in Block **1005** einen netzwerkunabhängigen Service **228** auf, der durch die Serviceschicht **212** offengelegt wird und eine generische (z. B. netzwerkunabhängige) Schnittstelle **220** aufweist, um das Lesen von Daten (z. B. überwachtem Druck) aus einem Feldgerät (z. B. einem Drucksender) anzufordern. Die Übersetzungsschicht **214** übersetzt in Block **1010** den netzwerkunabhängigen Service **228** in eine netzwerkabhängige Sequenz von Netzwerkoperationen **230**, um den Ser-

vice **228** in einem speziellen Gerätenetzwerk **210** zu implementieren, welches das Feldgerät enthält. In Block **1010** verwendet die Anwendungsfassade **229** der Übersetzungsschicht **214** beispielsweise Netzwerkkomponenteninformationen, die aus einer Gerätebeschreibungsdatei **208** erhalten wurden, um die Sequenz der Netzwerkoperationen **230** vorzubereiten. (Die Gerätebeschreibungsdatei **208** könnte beispielsweise in einer EDDL-Beschreibung **232** oder einer Beschreibung in üblichem Dateiformat **234** des Drucksenders sein).

[0087] In Block **1015** ruft die Übersetzungsschicht **214** eine von einer der Übersetzungsschicht **214** zugeordneten Netzwerkschicht **216** bereitgestellte Netzwerk-API **240** auf, um gemäß einer in Block **1010** bestimmten Sequenz von Netzwerkoperationen **230** netzwerkabhängige Transaktionen (z. B. Befehle, die gemäß den Netzwerk-Nachrichtentypen **242**, Netzwerk-Datentypen **244** und/oder Objektwörterbüchern **246** formatiert sind) an das Feldgerät (z. B. den Drucksender) zu senden. Später erhält in Block **1020** die Netzwerk-API **240** eine Antwort (die z. B. gemessene Druckdaten enthält) von dem Feldgerät (z. B. dem Drucksender). Die Übersetzungsschicht **214** übersetzt in Block **1025** das netzwerkabhängige Format der in Block **1015** erhaltenen Antwort in ein generisches netzwerkunabhängiges Antwortformat, das der Serviceschicht **212** zur Verfügung gestellt wird. In Block **1030** ruft die Serviceschicht **212** nach Bedarf einen Adapter **226** auf, um das generische netzwerkunabhängige Format der in Block **1025** vorbereiteten Antwort in ein Format umzuwandeln, das noch netzwerkunabhängig, jedoch für die Anwendung **202** spezifisch ist. In Block **1035** gibt die Serviceschicht die Antwort (z. B. unter Verwendung der Serviceschnittstelle **220** des in Block **1005** aufgerufenen Services **220**) an die Anwendung **202** zurück. Die Ausführung des beispielhaften Prozesses **1000** endet dann.

[0088] [Fig. 11](#) ist ein Blockdiagramm eines beispielhaften Verarbeitungssystems **1100**, das den FDI-Server **106** und das serviceorientierte Framework **200** aus [Fig. 2](#) implementieren kann. Das beispielhafte Verarbeitungssystem **110** kann zusätzlich oder alternativ zur Implementierung des Clients **104** verwendet werden, um die Client-Anwendungen **120**, **202** usw. auszuführen. Der Computer **1100** kann beispielsweise ein Server, ein PC, ein Mobiltelefon (z. B. ein Funktelefon), ein Personal Digital Assistant (PDA), ein Internet-Anwendungsprodukt oder jegliche andere Art von Rechengert sein.

[0089] Das beispielhafte Prozessorsystem **1100** umfasst einen Prozessor **1102** mit zugeordneten Speichern, wie einem Direktzugriffsspeicher (RAM) **1104**, einem Festwertspeicher (ROM) **1106** und einem Flash-Speicher **1108**. Der Prozessor **1102** kann unter anderem maschinenlesbare Anweisungen aus-

führen, um die in [Fig. 6–Fig. 10](#) dargestellten Prozesse zu implementieren. Der Prozessor **1102** kann jede Art von Verarbeitungseinheit sein, wie ein oder mehrere Intel®-Mikroprozessoren der Pentium®-Familie, der Itanium®-Familie und/oder der XScale®-Familie, ein oder mehrere Mikrocontroller der ARM®- und/oder PIC®-Familien von Mikrocontrollern usw. Es sind natürlich auch andere Prozessoren aus anderen Familien geeignet.

[0090] Der RAM **1104**, der ROM **1106** und/oder der Flash-Speicher **1108** können maschinenlesbare Anweisungen speichern, die die Prozesse **900** aus [Fig. 9](#) implementieren. Das RAM **1104** kann als ein Synchronous Dynamic Random Access Memory (SDRAM), Dynamic Random Access Memory (DRAM), RAMBUS Dynamic Random Access Memory (RDRAM) und/oder jede beliebige andere Art von Schreib-/Lesespeichervorrichtung implementiert sein. Der Flash-Speicher **1108** des dargestellten Beispiels weist einen Startblock **1110** auf. Der Zugriff auf das RAM **1104**, das ROM **1106** und den Flash-Speicher **1108** wird typischerweise durch einen Speichercontroller (nicht dargestellt) gesteuert.

[0091] Der Prozessor **1102** ist an eine Schnittstelle gekoppelt, wie einen Bus **1112**, zu dem andere Komponenten eine Schnittstelle haben. Im dargestellten Beispiel sind die Komponenten, die eine Schnittstelle zum Bus **1112** haben, eine Eingabevorrichtung **1114**, eine Anzeigevorrichtung **1116**, eine Massenspeichervorrichtung **1118** und ein Laufwerk für Wechselspeichervorrichtungen **1120**. Das Laufwerk für Wechselspeichervorrichtungen **1120** kann zugeordnete Wechselspeichermedien **1122** wie z. B. magnetische oder optische Medien aufweisen.

[0092] Die Eingabevorrichtung(en) **1114** erlaubt/erlauben es einem Benutzer, Daten und Befehle in den Prozessor **1102** einzugeben. Die Eingabevorrichtung **1114** kann unter Verwendung von einem oder mehreren von einer Tastatur, einer Maus, einem Touchscreen, einem Track-Pad, einem Strichcodescanner oder einer beliebigen anderen Vorrichtung implementiert sein, die es dem Benutzer ermöglicht, Informationen an den Prozessor **1102** bereitzustellen.

[0093] Bei der Anzeigevorrichtung **1116** kann es sich beispielsweise um einen Flüssigkristallanzeige-(LCD)-Monitor, einen Kathodenstrahlröhren-(CRT)-Monitor oder jede beliebige andere geeignete Vorrichtung handeln, die als eine Schnittstelle zwischen der Verarbeitungseinheit **1102** und einem Benutzer dient. Die Anzeigevorrichtung **1116**, wie in [Fig. 11](#) dargestellt, weist beliebige zusätzliche Hardware auf, die notwendig ist, um für einen Anzeigebildschirm eine Schnittstelle zum Prozessor **1102** zu bilden.

[0094] Bei der Massenspeichervorrichtung **1118** kann es sich beispielsweise um eine konventionelle

Festplatte oder ein beliebiges anderes magnetisches oder optisches Medium handeln, das von dem Prozessor **1102** abgelesen werden kann.

[0095] Bei dem Laufwerk für Wechselspeichervorrichtungen **1120** kann es sich beispielsweise um ein optisches Laufwerk wie z. B. ein CD-R-(Compact Disk-Recordable)-Laufwerk, ein CD-RW-(Compact Disk-Rewritable)-Laufwerk, ein DVD-(Digital Versatile Disk)-Laufwerk oder ein beliebiges anderes optisches Laufwerk handeln. Es kann sich alternativ beispielsweise auch um ein Laufwerk für magnetische Medien handeln. Das Wechselspeichermedium **1122** ergänzt das Laufwerk für Wechselspeichervorrichtungen **1120** insofern, als das Medium dazu ausgewählt wird, mit dem Laufwerk **1122** zu arbeiten. Wenn es sich bei dem Laufwerk für Wechselspeichervorrichtungen **1120** beispielsweise um ein optisches Laufwerk handelt, kann das Wechselspeichermedium **1122** eine CD-R-Disk, eine CD-RW-Disk, eine DVD-Disk oder jede beliebige andere geeignete optische Disk sein. Wenn es sich dagegen bei dem Laufwerk für Wechselspeichervorrichtungen **1120** um eine Vorrichtung für magnetische Medien handelt, kann das Wechselspeichermedium **1122** eine Diskette oder jedes beliebige andere geeignete magnetische Speichermedium sein.

[0096] Codierte Anweisungen zum Implementieren von einem oder mehreren der Prozesse aus [Fig. 6–Fig. 10](#) können im RAM **1104**, im ROM **1106**, im Flash-Speicher **1108**, in der Massenspeichervorrichtung **1118** und/oder auf dem Wechselspeichermedium **1122** wie z. B. einer CD oder DVD gespeichert sein.

[0097] Alternativ zur Implementierung der hier in einem System wie dem Verarbeitungssystem von [Fig. 11](#) beschriebenen Verfahren und/oder Vorrichtung(en) können die hier beschriebenen Verfahren und/oder Vorrichtung(en) in eine Struktur eingebettet werden, wie einen Prozessor oder einen ASIC (anwendungsspezifischen integrierten Schaltkreis).

[0098] Es ist aus der vorhergehenden Offenbarung zu erkennen, dass die hier beschriebenen beispielhaften Verfahren, Vorrichtung(en) und Fertigungsartikel zum Implementieren eines serviceorientierten Frameworks zur Kommunikation mit Geräten in einem Prozesssteuerungsnetzwerk verwendet werden können. Das hier offenbarte beispielhafte serviceorientierte Framework kann mindestens unter einigen Umständen gegenüber Prozesssteuerungslösungen nach dem Stand der Technik Vorteile aufweisen. In dem hier offenbarten serviceorientierten Framework können dem System beispielsweise Funktionen, Eigenschaften und Flexibilität von Anbietern hinzugefügt werden, ohne dass dies einen Einfluss auf bestehende Anwendungen hat. Als anderes Beispiel kann das hier offenbar-

te serviceorientierte Framework die Gesamtkomplexität des Systems reduzieren, weil jede Prozesssteuerungsanwendung nur zu wissen braucht, wie über Services kommuniziert wird, während spezifische Prozesssteuerungsnetzwerk-/gerätestandards, die zur Kommunikation mit Geräten verwendet werden, in einer unteren Ebene in dem Framework bereitgestellt werden. Prozesssteuerungsanwendungen können als weiteres Beispiel unabhängig von den zu Grunde liegenden Gerätenetzwerken geändert werden, weil das hier offenbarte serviceorientierte Framework eine generische netzwerkunabhängige Schnittstelle zur Kommunikation mit den Gerätenetzwerken zur Verfügung steht, wodurch Anwendungsänderungen, Updates und Ersetzungen nur auf dieser gleichen netzwerkunabhängigen Schnittstelle basieren können. Als weiteres Beispiel ermöglicht das hier offenbarte serviceorientierte Framework, dass jede Prozesssteuerungsanwendung lediglich eine einzelnen Verbindung zu dem Framework unterstützt, anstelle von mehreren Verbindungen, um jeden Netzwerktyp zu unterstützen. Das hier offenbarte serviceorientierte Framework ermöglicht als weiteres Beispiel, Services hinzuzufügen, um die von den Prozesssteuerungsgerätenetzwerken bereitgestellten Fähigkeiten zu unterstützen, und EDDL-Beschreibungen können definieren, welche Services ein spezielles Gerät zur Verfügung stellt. Das hier offenbarte serviceorientierte Framework umfasst als weiteres Beispiel gut definierte Schnittstellen zwischen Schichten, die als Konformitätsprüfungspunkte zur Validierung von Prozesssteuerungssystemimplementierungen verwendet werden können. Die netzwerkunabhängigen Serviceschnittstellen des hier offenbarten serviceorientierten Frameworks können als weiteres Beispiel Kunden ermöglichen, Geräte von beliebigen Anbietern zu erwerben und mit der Erwartung in die Architektur einzubauen, dass Prozesssteuerungsanwendungen weiterhin mit den verschiedenen Geräten kommunizieren.

[0099] Obgleich bestimmte beispielhafte Verfahren, Vorrichtungen und Fertigungsartikel beschrieben wurden, ist der Geltungsbereich des vorliegenden Patents nicht auf diese beschränkt. Dieses Patent deckt im Gegenteil alle Verfahren, Vorrichtungen und Fertigungsartikel ab, die entweder wörtlich oder gemäß der Äquivalenzlehre zur Gänze in den Rahmen der angehängten Ansprüche fallen.

ZITATE ENTHALTEN IN DER BESCHREIBUNG

Diese Liste der vom Anmelder aufgeführten Dokumente wurde automatisiert erzeugt und ist ausschließlich zur besseren Information des Lesers aufgenommen. Die Liste ist nicht Bestandteil der deutschen Patent- bzw. Gebrauchsmusteranmeldung. Das DPMA übernimmt keinerlei Haftung für etwaige Fehler oder Auslassungen.

Zitierte Nicht-Patentliteratur

- Standardfamilie 802.11 des Institute of Electrical and Electronic Engineers (IEEE)
[\[0033\]](#)

Patentansprüche

1. Verfahren zum Kommunizieren mit einem Gerät in einem Prozesssteuerungssystem, umfassend:

Aufrufen eines Service, um mit dem Gerät in dem Prozesssteuerungssystem zu kommunizieren, wobei der Service eine generische Schnittstelle aufweist, die unabhängig von einem zur Implementierung des Prozesssteuerungssystems verwendeten Prozesssteuerungsnetzwerkprotokoll ist;

Übersetzen des Service in eine oder mehrere Netzwerkoperationen, um den Service zu implementieren, wobei die Netzwerkoperationen für das zur Implementierung des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch sind; und

Verwenden einer Netzwerkschnittstelle, die für das zur Implementierung des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch ist, zur Kommunikation mit dem Gerät gemäß der einen oder den mehreren Netzwerkoperationen.

2. Verfahren nach Anspruch 1, wobei der Service als Servicevertrag durch eine Serviceschicht offengelegt wird, die eine Vielzahl von Services bereitstellt, jede der Vielzahl der Services von dem zur Implementierung des Prozesssteuerungssystems verwendeten Prozesssteuerungsnetzwerkprotokoll unabhängig ist und jeder der Vielzahl von Services die Kommunikation mit Geräten in einer Vielzahl verschiedener Prozesssteuerungssysteme unterstützt, die unter Verwendung einer Vielzahl verschiedener Prozesssteuerungsnetzwerkprotokolle implementiert sind.

3. Verfahren nach Anspruch 1 oder 2, wobei das Aufrufen des Service die Verwendung von einem oder mehreren von einer Vielzahl von Nachrichten umfasst, die eine Serviceschnittstelle auf Nachrichtenbasis definieren, um den Service aufzurufen, wobei die Vielzahl von Nachrichten eine Vielzahl von Service-Nachrichtentypen und eine Vielzahl von Service-Datentypen umfasst, wobei die Vielzahl von Service-Nachrichtentypen und die Vielzahl von Service-Datentypen unabhängig von jeglichem Prozesssteuerungsnetzwerkprotokoll sind.

4. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 1, wobei das Übersetzen des Services in eine oder mehrere Netzwerkoperationen zur Implementierung des Services Folgendes umfasst:

Übersetzen des Service in eine Sequenz von Netzwerkoperationen, die für das Prozesssteuerungsnetzwerkprotokoll spezifisch sind, und

Pflegen eines Status der Sequenz von Netzwerkoperationen, um zu verfolgen, welche aus der Sequenz von Netzwerkoperationen durchgeführt worden sind, um den Service zu implementieren.

5. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 1, wobei das Übersetzen des Service in eine oder mehrere Netzwerkoperationen zum Implementieren des Services das Lesen einer Gerätebeschreibung für das Gerät umfasst, um die eine oder mehreren Netzwerkoperationen vorzubereiten.

6. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 5, wobei die Gerätebeschreibung in mindestens einer aus einer elektronischen Gerätebeschreibungssprache oder einem üblichen Dateiformat gespeichert wird.

7. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 1, wobei die netzwerkspezifische Schnittstelle eine Netzwerk-Anwendungsprogrammchnittstelle umfasst, die mindestens eines aus einer Vielzahl von Netzwerk-Nachrichtentypen, einer Vielzahl von Netzwerk-Datentypen oder einem Objektwörterbuch umfasst und für das zur Implementierung des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch ist.

8. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 1, ferner umfassend:

Empfangen einer Antwort von dem Gerät, wobei die Antwort in einem ersten Format vorliegt, das für das zur Implementierung des Prozesssteuerungssystems verwendete Prozesssteuerungsnetzwerkprotokoll spezifisch ist;

Übersetzen der Antwort aus dem ersten Format in ein zweites Format, das von dem zur Implementierung des Prozesssteuerungssystems verwendeten Prozesssteuerungsnetzwerkprotokoll unabhängig ist, und

Aufrufen eines Adapters, um die Antwort von dem zweiten Format in ein drittes Format umzuwandeln, welches für eine Anwendung spezifisch ist, die die Antwort empfangen soll.

9. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 1, das ferner Authentifizieren einer Anwendung vor Aufrufen des Service umfasst.

10. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 1, das ferner das Einschränken der Kommunikation mit dem Gerät auf Grundlage einer Benutzerautorisierung umfasst, die beim Aufrufen des Services durchgeführt wird.

11. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 1, wobei der Service mindestens einem der Folgenden entspricht:

einem ersten Service zum Senden einer Nachricht an ein Gerät und zum Empfangen einer entsprechenden Antwort von dem Gerät;

einem zweiten Service zum Abonnieren der veröffentlichten Daten, die von dem Gerät zurückgegeben werden;

einem dritten Service zum Empfangen von Ereignissen, die von dem Gerät zurückgegeben werden;

einem vierten Service zum Erhalten von Informationen, die das Prozesssteuerungssystem und eine Vielzahl von Geräten einschließlich des Geräts, das das Prozesssteuerungssystem implementiert, beschreiben; und

einem fünften Service zum Schreiben von Steuerungsparameterwerten in das Gerät.

12. Verfahren nach einem der vorhergehenden Ansprüche, insbesondere nach Anspruch 1, wobei der Service unter Verwendung von einer oder mehreren asynchron gesendeten Nachrichten aufgerufen wird und die Netzwerkschnittstelle synchrones Polling implementiert, um Daten von dem Gerät zu erhalten.

13. Vorrichtung zum Kommunizieren mit Feldgeräten in Prozesssteuerungssystemen, wobei die Vorrichtung umfasst:

einen Prozessor zur Implementierung eines serviceorientierten Frameworks zur Kommunikation mit einer Vielzahl von Feldgeräten in einer Vielzahl von Prozesssteuerungssystemen, die unter Verwendung einer Vielzahl verschiedener Prozesssteuerungsnetzwerkprotokolle implementiert sind, wobei das serviceorientierte Framework umfasst:

eine Serviceschicht, die eine Vielzahl von Services zur Kommunikation mit der Vielzahl von Feldgeräten implementiert, wobei jeder Service eine jeweilige generische Schnittstelle hat, die von jeglichen der Vielzahl von Prozesssteuerungsnetzwerkprotokollen unabhängig ist, die zur Implementierung der Vielzahl von Prozesssteuerungssystemen verwendet werden; eine Vielzahl von Übersetzungsschichten, wobei jede jeweilige Übersetzungsschicht jeden Service in der Vielzahl von Services in eine jeweilige Sequenz von Netzwerkoperationen übersetzt, um den jeweiligen Service zu implementieren, wobei die jeweilige Sequenz von Netzwerkoperationen für ein spezielles Prozesssteuerungsnetzwerkprotokoll spezifisch ist, das der jeweiligen Übersetzungsschicht zugeordnet ist; und

eine Vielzahl von Netzwerkschichten, die jeweils der Vielzahl von Übersetzungsschichten zugeordnet sind, um eine Vielzahl von Netzwerkschnittstellen bereitzustellen, die für jedes der Vielzahl von Prozesssteuerungsnetzwerkprotokollen spezifisch sind, die zur Implementierung der Prozesssteuerungssysteme verwendet werden; und

eine Schnittstelle zur kommunikativen Kopplung einer auf einem Client-Gerät implementierten Anwendung an die durch den Prozessor implementierte Serviceschicht.

14. Vorrichtung nach Anspruch 13, wobei jeder Service als jeweiliger Servicevertrag durch die Serviceschicht offengelegt wird, jeder Service die Kommunikation mit der Vielzahl von Feldgeräten in der Vielzahl verschiedener Prozesssteuerungssysteme unterstützt, die unter Verwendung der Vielzahl verschiedener Prozesssteuerungsnetzwerkprotokolle implementiert sind, und jeder Service unter Verwendung von einer oder mehreren von einer Vielzahl von Nachrichten aufgerufen wird, die eine Service-schnittstelle auf Nachrichtenbasis definieren, um jeden von der Vielzahl von Services aufzurufen, wobei die Vielzahl der Nachrichten eine Vielzahl von Service-Nachrichtentypen und eine Vielzahl von Service-Datentypen umfasst, wobei die Vielzahl der Service-Nachrichtentypen und der Vielzahl von Service-Datentypen unabhängig von jeglichem Prozesssteuerungsnetzwerkprotokoll sind.

15. Vorrichtung nach Anspruch 13 oder 14, insbesondere nach Anspruch 13, wobei eine erste Übersetzungsschicht einen ersten Service in eine erste Sequenz von Netzwerkoperationen übersetzt durch: Lesen einer Gerätebeschreibung für ein erstes Feldgerät, mit dem der erste Service kommunizieren soll, um die erste Sequenz von Netzwerkoperationen vorzubereiten; und

Pflegen eines Status der ersten Sequenz von Netzwerkoperationen, um zu verfolgen, welche aus der ersten Sequenz von Netzwerkoperationen durchgeführt worden ist, um den ersten Service zu implementieren.

16. Vorrichtung nach einem der Ansprüche 13–15, insbesondere nach Anspruch 13, wobei eine erste netzwerkspezifische Schnittstelle durch eine erste Netzwerkschicht bereitgestellt wird, die eine Netzwerk-Anwendungsprogramm-schnittstelle umfasst, die mindestens eins aus einer Vielzahl von Netzwerk-Nachrichtentypen, einer Vielzahl von Netzwerk-Datentypen oder einem Objektwörterbuch umfasst und für das zur Implementierung eines ersten Prozesssteuerungssystems verwendete erste Prozesssteuerungsnetzwerkprotokoll spezifisch ist.

Es folgen 8 Blatt Zeichnungen

Anhängende Zeichnungen

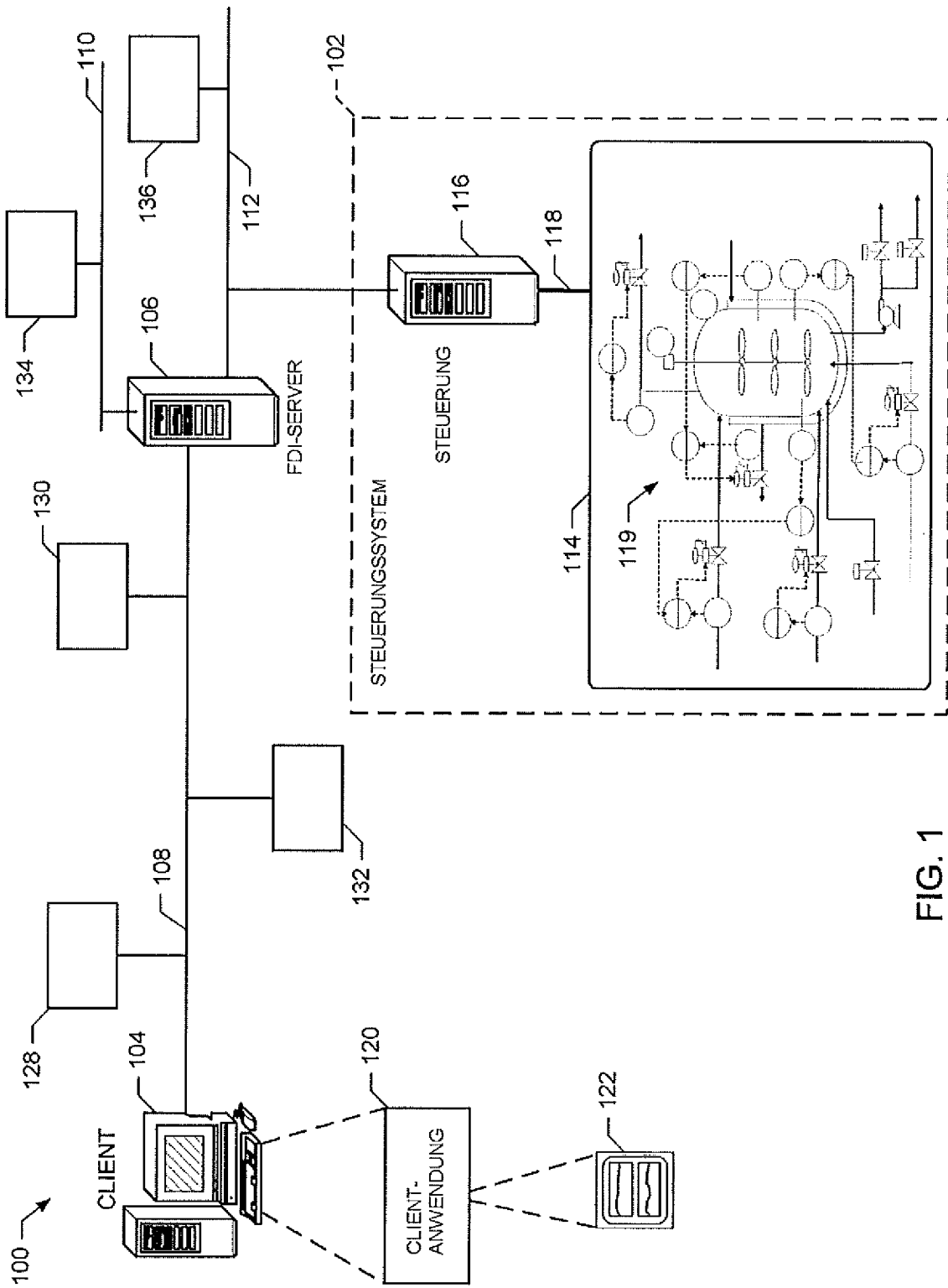


FIG. 1

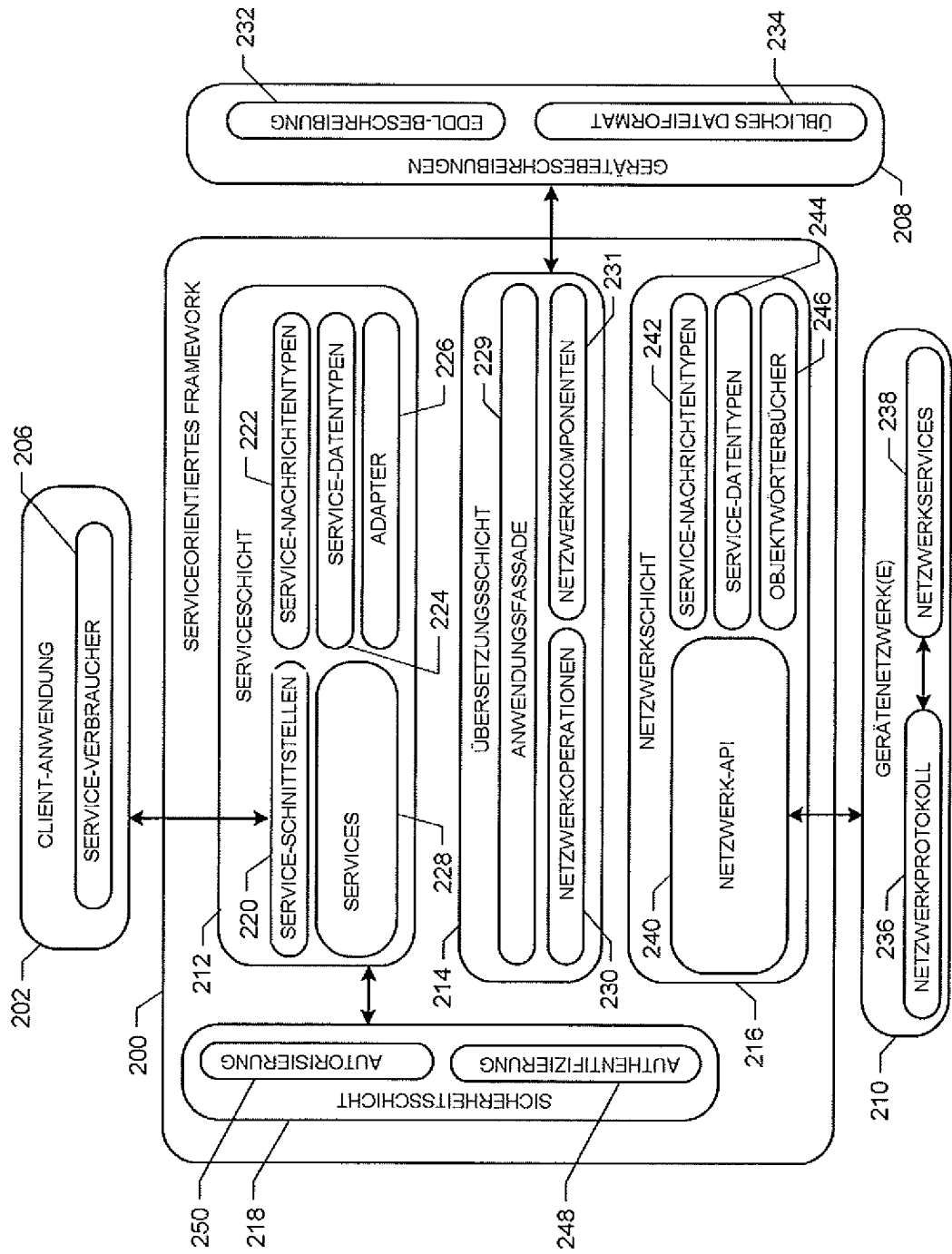


FIG. 2

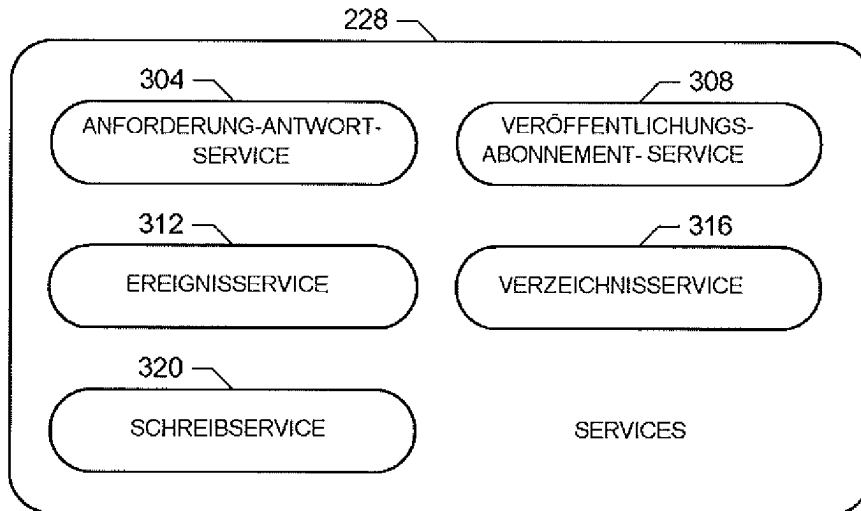


FIG. 3

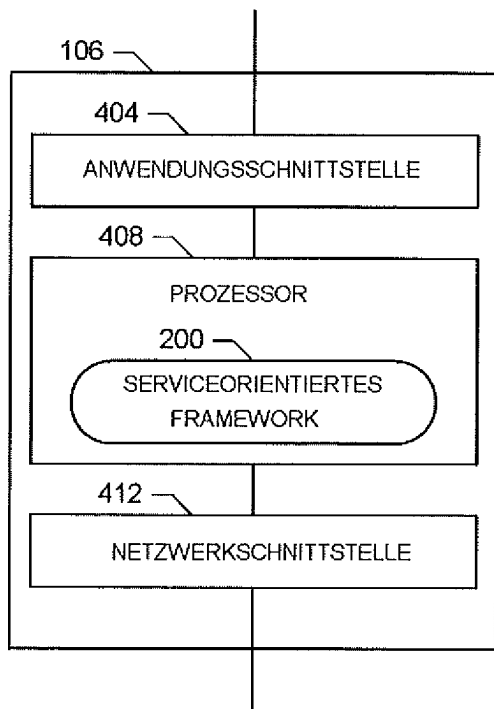


FIG. 4

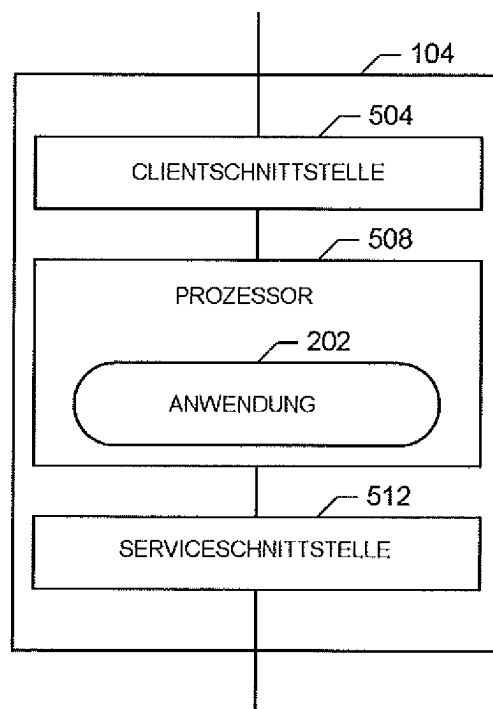


FIG. 5

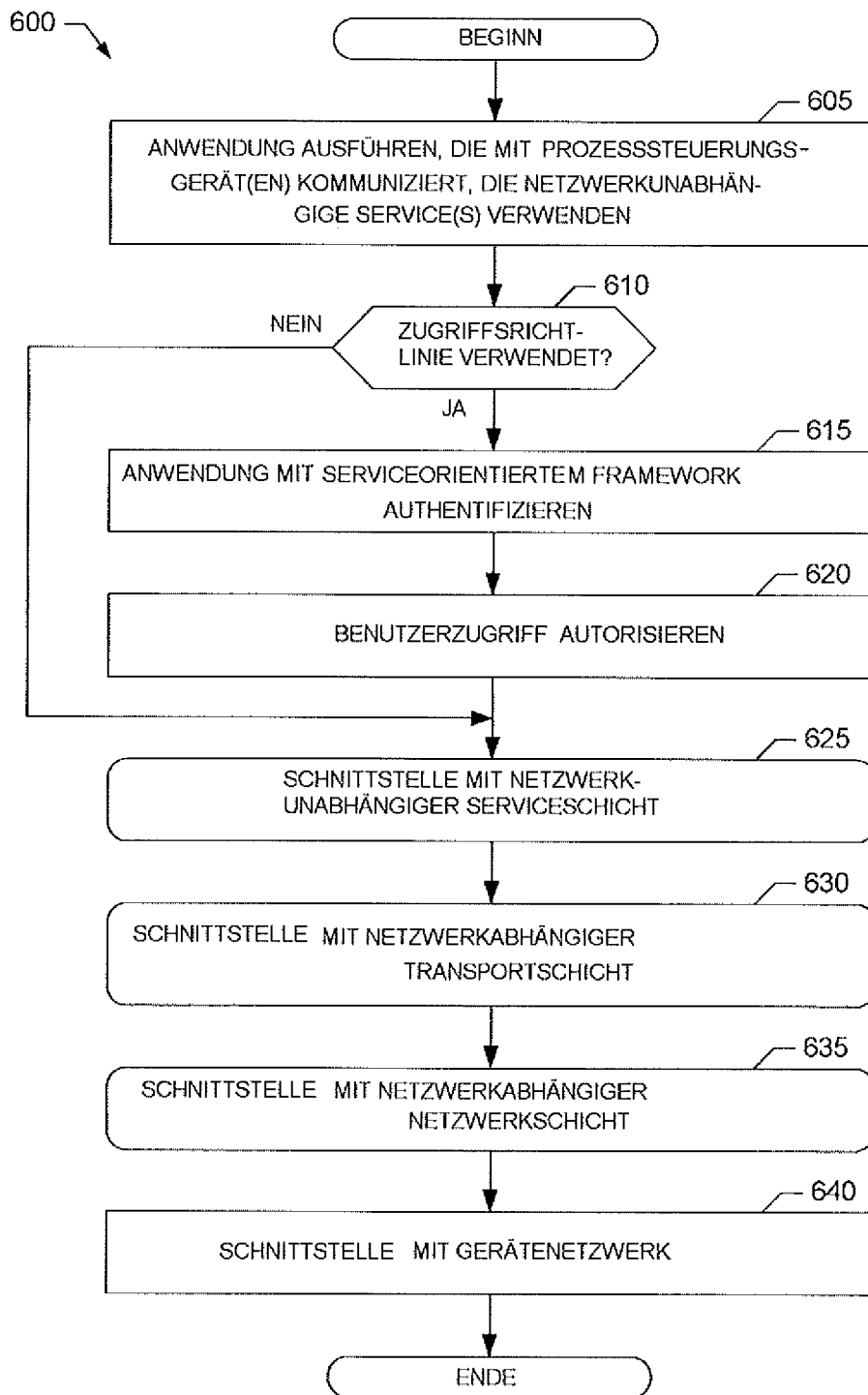


FIG. 6

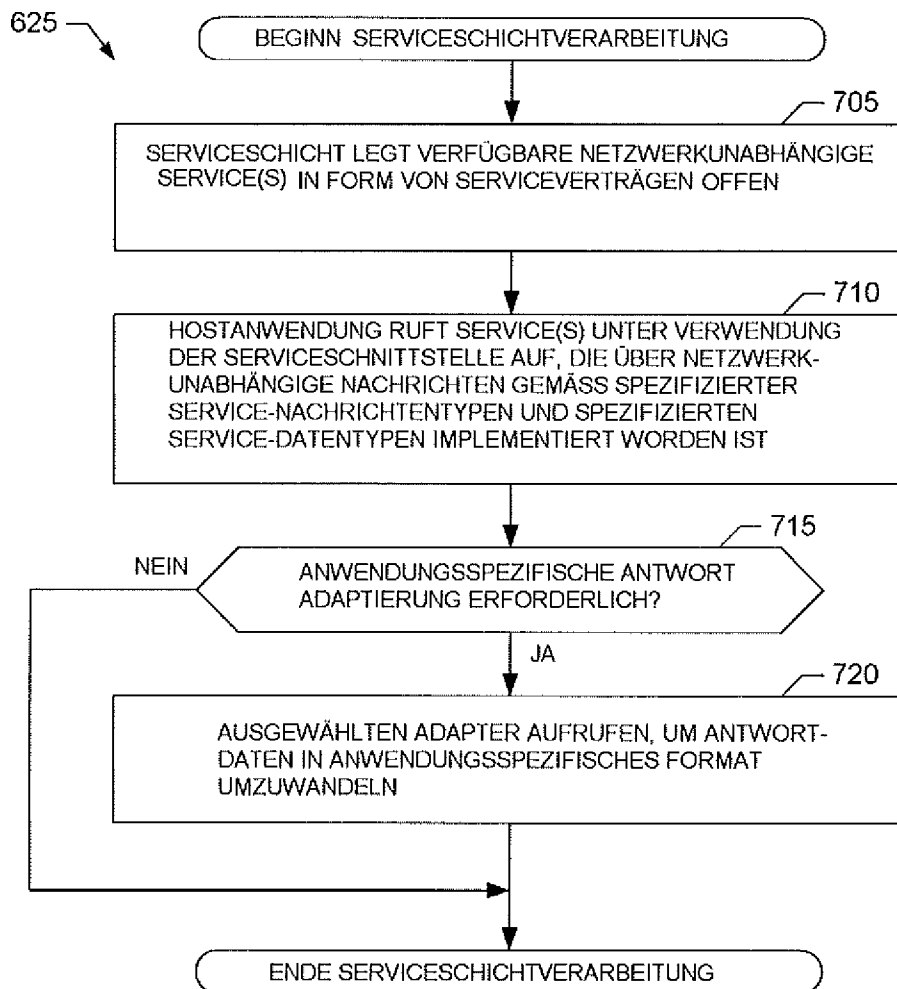


FIG. 7

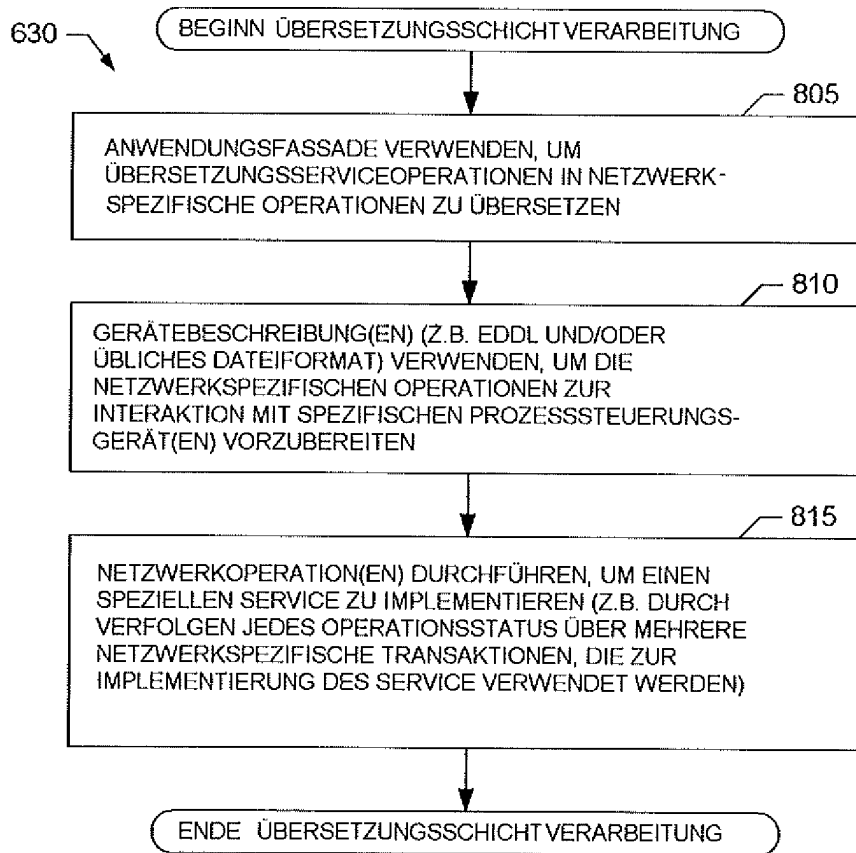


FIG. 8

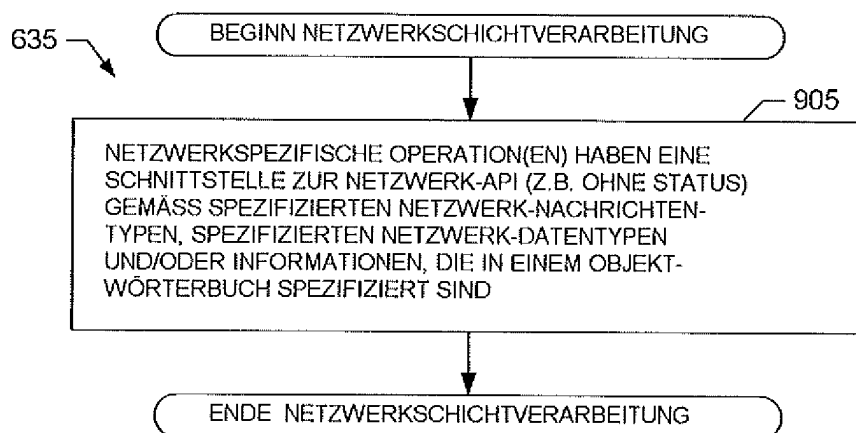


FIG. 9

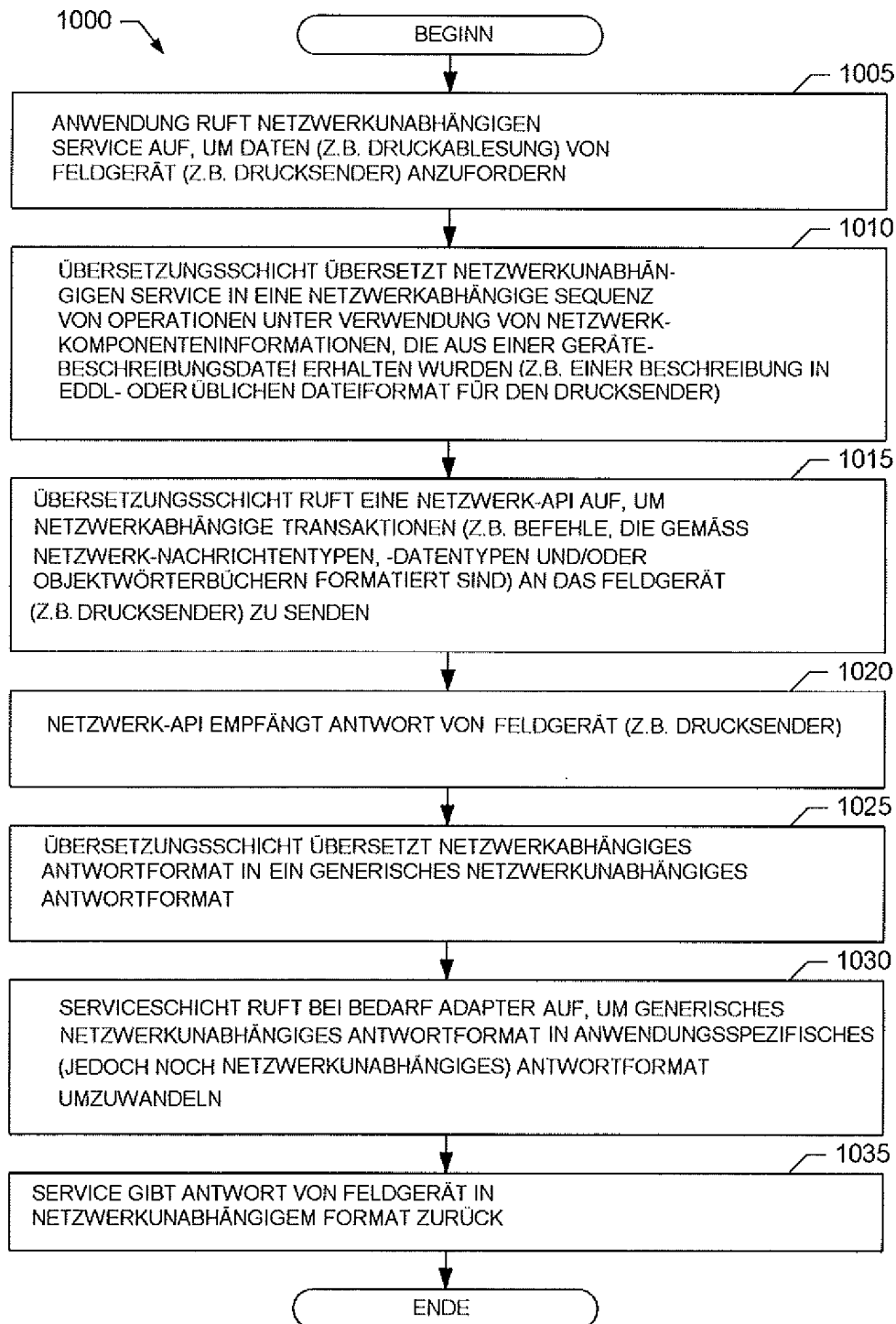


FIG. 10

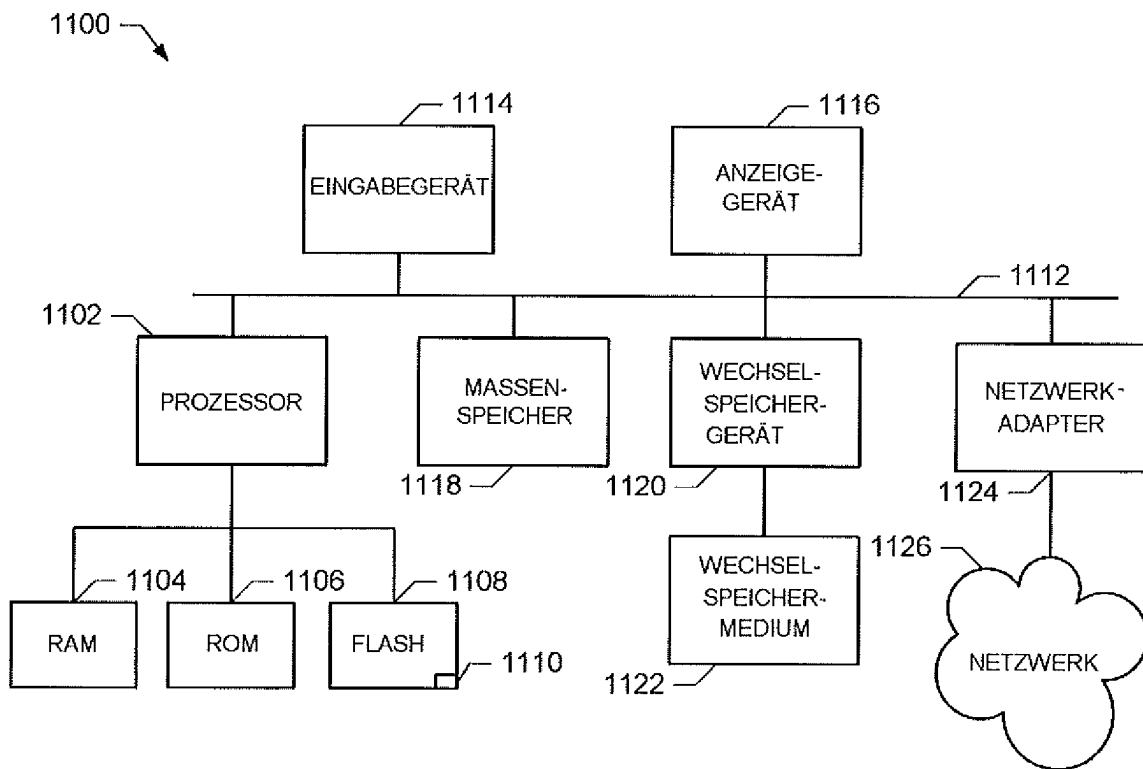


FIG. 11