



(12)

Offenlegungsschrift

(21) Aktenzeichen: **10 2011 001 528.0**

(22) Anmelddatum: **24.03.2011**

(43) Offenlegungstag: **15.12.2011**

(30) Unionspriorität:

12/730,895

24.03.2010 US

(51) Int Cl.:

H04L 12/16 (2011.01)

H04L 29/02 (2011.01)

(72) Erfinder:

Neitzel, Lee Allen, Austin, Tex., US; Zhou, Ling, Austin, Tex., US; Nixon, Mark J., Round Rock, Tex., US

(71) Anmelder:

Fisher-Rosemount Systems, Inc., Austin, Tex., US

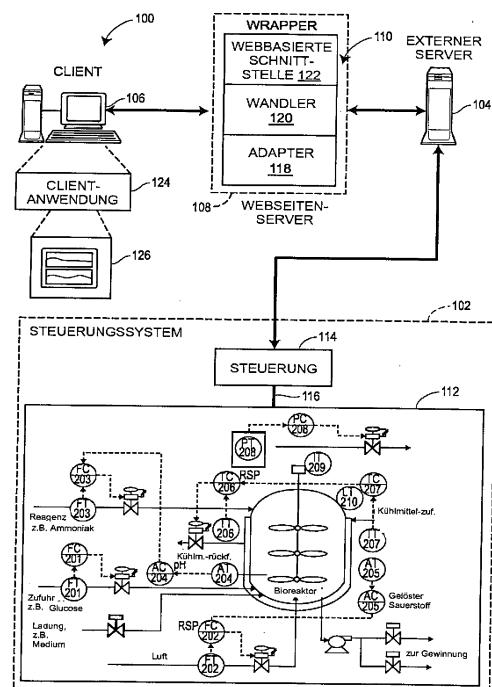
(74) Vertreter:

Meissner, Bolte & Partner GbR, 80538, München, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: **Verfahren und Vorrichtungen zum Zugreifen auf Prozessdaten, die auf einem Server gespeichert sind**

(57) Zusammenfassung: Offenbart werden beispielhafte Verfahren und Vorrichtungen zum Zugreifen auf Prozessdaten, die auf einem Server gespeichert sind. Ein offenbartes Beispielverfahren umfasst das Empfangen einer Anforderung zum Zugreifen auf Prozessdaten über einen Webbrowser, das Identifizieren eines Servers, auf dem wenigstens ein Teil der Prozessdaten gespeichert ist, wobei der Server einem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, das Zugreifen auf den Server zum Empfangen der Prozessdaten, das Umwandeln der Prozessdaten aus einem Format, das dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, in ein Webbrowserformat, und das Einbetten von wenigstens einem ersten Teil der umgewandelten Prozessdaten in wenigstens ein Datenfeld zur Anzeige über den Webbrowser.



Beschreibung**GEBIET DER OFFENBARUNG**

[0001] Die vorliegende Offenbarung betrifft allgemein Prozesssteuerungssysteme und insbesondere Verfahren und Vorrichtungen zum Zugreifen auf Prozessdaten, die auf einem Server gespeichert sind.

ALLGEMEINER STAND DER TECHNIK

[0002] Datensysteme wie z. B. Steuerungssysteme, Herstellungsautomatisierungssysteme und andere industrielle Systeme, wie sie in chemischen, Mineralöl- und sonstigen Prozessen verwendet werden, können Daten auf einem Server speichern, um es berechtigten Benutzern zu ermöglichen, von jedem beliebigen Ort aus auf die Daten zuzugreifen. Typischerweise können die Daten auf Server gespeichert sein, die OPC-konform sind. OPC ist eine Gruppe von Standards, die protokollbasierte Schnittstellen für Server definieren, die dazu verwendet werden, Zugriff auf industrielle Prozesssteuerungsdaten, Automatisierungsdaten, Herstellungsdaten und/oder Losdaten zu bieten. OPC definiert außerdem Formatierungsstandards für die Interoperabilität der Datenverpackung innerhalb der Server im Zusammenhang mit industriellen Prozessen.

[0003] Gegenwärtig können Clients sich mit einem Server verbinden, indem sie eine Anwendung erzeugen, die für einen OPC-Server oder einen Datentyp, der auf einem OPC-Server gespeichert ist, spezifisch ist. Beispielsweise kann eine Anwendung als ein ausführbares Programm in einer verfahrensorientierten Programmiersprache geschrieben sein, die mit einer spezifischen Betriebsplattform und einem Kommunikationsprotokoll eines entsprechenden OPC-Servers übereinstimmt. Die Anwendung kann von einem Server-Anbieter oder einem Client-Anbieter erzeugt werden, der die Anwendung speziell für den jeweiligen OPC-Server konfiguriert. Bei Beispielen, in denen ein Client auf Daten zugreifen muss, die auf mehreren OPC-Servern gespeichert sind, muss unter Umständen eine separate Anwendung für jeden OPC-Server erzeugt werden, um die spezifischen Schnittstellenanforderungen der OPC-Server zu erfüllen. Ferner muss jede dieser Anwendungen unter Umständen modifiziert werden, wenn die Daten, die auf dem entsprechenden OPC-Server gespeichert sind, auf einen anderen Server migriert werden, oder wenn sich die Schnittstellenanforderungen des OPC-Servers ändern.

KURZFASSUNG

[0004] Es werden nun beispielhafte Verfahren und Vorrichtungen zum Zugreifen auf Prozessdaten beschrieben, die auf einem Server gespeichert sind. In einem Beispiel beinhaltet ein Verfahren das Empfangen einer Anforderung zum Zugreifen auf Prozessdaten über einen Webbrowser, das Identifizieren eines Servers, der wenigstens einen Teil der Prozessdaten speichert, wobei der Server einem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, und das Zugreifen auf den Server, um die Prozessdaten zu empfangen. Das Beispielverfahren beinhaltet ferner das Umwandeln der Prozessdaten aus einem Format, das dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, in ein Webbrowserformat, und das Einbetten von wenigstens einem ersten Teil der umgewandelten Prozessdaten in wenigstens ein Datenfeld zur Anzeige über einen Webbrowser.

[0005] Eine Beispielvorrichtung weist einen Adapter zum Identifizieren eines Servers auf, der wenigstens einen Teil der Prozessdaten speichert, die einer Anforderung zum Zugreifen auf die Prozessdaten über einen Webbrowser zugeordnet sind, wobei der Server einem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, und zum Zugreifen auf den Server, um die Prozessdaten zu empfangen. Die Beispielvorrichtung weist ferner einen Wandler zum Umwandeln der Prozessdaten aus einem Format, das dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, in ein Webbrowser-Format, und eine webbasierte Schnittstelle zum Einbetten von wenigstens einem ersten Teil der umgewandelten Prozessdaten in wenigstens ein Datenfeld zur Anzeige über den Webbrowser auf.

[0006] Ein Aspekt der Erfindung betrifft ein maschinenzugängliches Medium, auf dem Anweisungen gespeichert sind, die bei ihrer Ausführung die Maschine dazu veranlassen eine Anforderung zum Zugreifen auf Prozessdaten über einen Webbrowser zu empfangen; einen Server zu identifizieren, auf dem wenigstens ein Teil der Prozessdaten gespeichert ist, wobei der Server einem Interoperabilitäts-Datenverpackungsformat zugeordnet ist; auf den Server zuzugreifen, um die Prozessdaten zu empfangen; die Prozessdaten aus einem Format, das dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, in ein Webbrowserformat umwandeln; und

wenigstens einen ersten Teil der umgewandelten Prozessdaten in wenigstens ein Datenfeld zur Anzeige über den Webbrowser einzubetten.

[0007] Ein weiterer Aspekt betrifft ein maschinenzugängliches Medium, bei dem die maschinenzugänglichen Anweisungen bei ihrer Ausführung die Maschine dazu veranlassen, die Anforderung zu authentifizieren, bevor sie den Server identifiziert, auf dem die Prozessdaten gespeichert sind.

[0008] Ein weiterer Aspekt betrifft ein maschinenzugängliches Medium, bei dem die maschinenzugänglichen Anweisungen bei ihrer Ausführung die Maschine dazu veranlassen, den Server durch Zugreifen auf eine Liste zu identifizieren, die auf einen Identifikator verweist, der der Anforderung zugeordnet ist.

[0009] Ein weiterer Aspekt betrifft ein maschinenzugängliches Medium, bei dem die maschinenzugänglichen Anweisungen bei ihrer Ausführung die Maschine dazu veranlassen, wenigstens den ersten Teil der umgewandelten Prozessdaten in das wenigstens eine Datenfeld einzubetten durch:

Unterteilen der umgewandelten Prozessdaten nach wenigstens einem Datentyp;

Ermitteln, welcher Datentyp den einzelnen Datenfeldern zugeordnet ist;

Erzeugen eines Datenfelds für jeden Datentyp; und

Einbetten der Prozessdaten, die den einzelnen Datentypen zugeordnet sind, in ein entsprechendes Datenfeld.

[0010] Ein weiterer Aspekt betrifft ein maschinenzugängliches Medium, bei dem die maschinenzugänglichen Anweisungen bei ihrer Ausführung die Maschine dazu veranlassen, zu ermitteln, welcher Datentyp den einzelnen Datenfeldern zugeordnet ist, durch:

Ermitteln von Metadaten, die den einzelnen Datenteilen zugeordnet sind; und

Zuweisen der Metadaten zu dem entsprechenden Datenfeld.

[0011] Ein weiterer Aspekt betrifft ein maschinenzugängliches Medium, bei dem die Anforderung von einem Client stammt, der den Webbrowser benutzt.

[0012] Ein weiterer Aspekt betrifft ein maschinenzugängliches Medium, bei dem die maschinenzugänglichen Anweisungen bei ihrer Ausführung die Maschine zu Folgendem veranlassen:

Einbetten des wenigstens einen Datenfelds einschließlich des ersten Teils der umgewandelten Prozessdaten in eine Webseite; und

Zurücksenden der Webseite einschließlich der Daten in dem wenigstens einen Datenfeld an den Client;

Verwenden des Webbrowsers, um wenigstens den ersten Teil der umgewandelten Prozessdaten auf der Webseite anzuzeigen.

[0013] Ein weiterer Aspekt betrifft ein maschinenzugängliches Medium, bei dem die Anforderung von einem Client stammt, der eine Client-Anzeigeanwendung benutzt, die in dem Webbrowser ausgeführt wird.

[0014] Ein weiterer Aspekt betrifft ein maschinenzugängliches Medium, bei dem die maschinenzugänglichen Anweisungen bei ihrer Ausführung die Maschine dazu veranlassen, eine Anweisung an den Webbrowser zu senden, die die Client-Anzeigeanwendung dazu veranlasst, auf das wenigstens eine Datenfeld mit wenigstens dem ersten Teil der umgewandelten Prozessdaten zuzugreifen und diese anzuzeigen.

KURZE BESCHREIBUNG DER FIGUREN

[0015] [Fig. 1](#) ist ein Blockdiagramm, das eine Steuerungsumgebung darstellt, wobei ein beispielhaftes Steuerungssystem kommunizierend an einen OPC-Server gekoppelt ist, auf den ein Client über einen Webseitenserver zugreifen kann, der einen Wrapper aufweist.

[0016] [Fig. 2](#) ist ein Funktionsdiagramm des beispielhaften Wrappers aus [Fig. 1](#).

[0017] [Fig. 3](#) zeigt eine beispielhafte Schnittstelle für Serveridentifikationsprozessdaten, die in einem Webbrower angezeigt werden.

[0018] [Fig. 4](#) zeigt eine beispielhafte Schnittstelle für ein Serverhierarchieverzeichnis, das in einem Webbrower angezeigt wird.

[0019] [Fig. 5](#) zeigt eine beispielhafte Schnittstelle für Prozessdaten, die in einem Webbrower angezeigt werden.

[0020] [Fig. 6](#) zeigt eine beispielhafte Schnittstelle, die mit dem beispielhaften Wrapper aus [Fig. 1](#) und/oder [Fig. 2](#) kommuniziert, um Prozessdaten in einem Webbrowser anzuzeigen.

[0021] [Fig. 7A](#) und [Fig. 7B](#) zeigen eine beispielhafte Schnittstelle, die auf einer drahtlosen Vorrichtung angezeigt wird, welche mit dem beispielhaften Wrapper aus [Fig. 1](#) und/oder [Fig. 2](#) kommuniziert, um Prozessdaten auf einer Client-Anwendung anzuzeigen.

[0022] [Fig. 8](#) zeigt eine beispielhafte Schnittstelle, die Prozessdaten in einem Webbrowser anzeigt.

[0023] [Fig. 9A–Fig. 9C](#) und [Fig. 10](#) sind Ablaufdiagramme eines beispielhaften Verfahrens, das dazu benutzt werden kann, den beispielhaften Webseitenserver, den beispielhaften OPC-Server, den beispielhaften Wrapper, eine webbasierte Schnittstelle, einen OPC-Wandler und/oder einen Adapter aus [Fig. 1](#) und/oder [Fig. 2](#) zu implementieren.

[0024] [Fig. 11](#) ist ein Blockdiagramm eines beispielhaften Prozessorsystems, das dazu benutzt werden kann, die hier beschriebenen beispielhaften Verfahren und Vorrichtungen zu implementieren.

DETAILLIERTE BESCHREIBUNG

[0025] Obwohl die nachfolgende Offenbarung beispielhafte Verfahren und Vorrichtungen beschreibt, die neben anderen Komponenten Software und/oder Firmware enthalten, die auf Hardware ausführbar ist, ist zu beachten, dass diese Beispiele nur veranschaulichend sind und nicht als einschränkend auszulegen sind. Beispielsweise ist es vorgesehen, dass ein Teil der Hardware-, Software- und Firmwarekomponenten oder alle davon ausschließlich in Hardware, ausschließlich in Software oder in jeder beliebigen Kombination aus Hardware und Software verkörpert sein können. Obwohl also im Folgenden beispielhafte Verfahren und Vorrichtungen beschrieben werden, werden Durchschnittsfachleute ohne Weiteres verstehen, dass die bereitgestellten Beispiele nicht der einzige Weg zum Implementieren dieser Verfahren und Vorrichtungen sind. Während beispielsweise die beispielhaften Verfahren und Vorrichtungen im Zusammenhang mit und/oder als übereinstimmend mit OPC beschrieben werden, können die beispielhaften Verfahren und Vorrichtungen durch jedes andere Interpretierbarkeits-Datenverpackungsformat implementiert sein, das in industriellen Anwendungen, Herstellungsanwendungen, Prozesssteuerungsanwendungen, Automatisierungsanwendungen usw. benutzt werden kann.

[0026] Gegenwärtig veröffentlicht die OPC Foundation eine Gruppe von OPC-Spezifikationen, die Programmierschnittstellen definieren, welche von Webbrowsern und/oder Client-Programmschnittstellen zum Zugreifen auf Server benutzt werden können, auf denen Prozessdaten angeordnet sind. Diese Schnittstellen sind in Bezug auf Verfahren und/oder Instanzen definiert, die in den Servern eingeleitet werden können. Ferner legen die OPC-Spezifikationen Parameter fest, die von OPC-bezogenen Servern an eine Anforderung stellende Client-Anwendungen übermittelt werden können. Die OPC-Spezifikationen definieren Schnittstellen, die unterschiedliche Protokolle benutzen können, die in verfahrensorientierten Programmiersprachen wie z. B. C++, Visual Basic und/oder C# geschrieben sind.

[0027] Um auf diese Server zuzugreifen, erzeugen Clients gegenwärtig Anwendungen, bei denen es sich um ausführbare Programme handelt, die in einer verfahrensorientierten Programmiersprache geschrieben sind, um mit den Protokollen der OPC-Schnittstellen dieser Server übereinzustimmen. Die Anwendungen werden erzeugt, um Zugriff für einen oder mehrere Benutzer bereitzustellen, und zwar unabhängig vom Aufenthaltsort des Benutzers, damit dieser Prozessdaten und/oder jeden anderen Typ von Daten, die auf einem OPC-Server gespeichert sein können, lesen, schreiben und/oder beziehen kann. Die Anwendungen können auf jeder beliebigen Rechenvorrichtung (z. B. einem Personalcomputer, einem Laptop, einem Smartphone usw.) ausgeführt werden, wodurch ein Benutzer von jedem Aufenthaltsort aus auf Prozessdaten (z. B. Objekte) zugreifen kann. In anderen Beispielen können die Anwendungen auf zugeordneten OPC-Servern als angepasste Webseite angeordnet sein, die auf Runtime-Frameworks, Datenbanken und/oder andere Datenquellen in einem OPC-Server zugreift. Ein berechtigter Benutzer kann über jede beliebige Vorrichtung, die dazu in der Lage ist, sich mit dem Internet zu verbinden, auf diese Anwendung zugreifen.

[0028] Die Anwendungen können vordefinierte Datenfelder mit Querverweis auf Verzeichnispositionen innerhalb eines OPC-Servers aufweisen. Die Anwendung kann auf Grundlage des Prozessdatentyps (z. B. eines Objekttyps) die Datenfelder für eine bestimmte Anzeige (z. B. Textfelder, Graphen, Diagramme, Tabellen usw.) vorformatieren. Ferner können die Client-Anwendungen für eine Plattform und/oder ein Kommunikationsprotokoll erzeugt werden, die bzw. das für einen OPC-Server spezifisch ist. Beispielsweise werden die Client-An-

wendungen typischerweise mit Adressen codiert, die den gewünschten OPC-Servern entsprechen. Aufgrund dieser Codierungsmerkmale und Verweise auf Verzeichnisorte in einem OPC-Server sind die Client-Anwendungen typischerweise statisch und können nur mit einem bestimmten OPC-Server kommunizieren.

[0029] Da eine spezialisierte Anwendung für jeden OPC-Server und/oder eine Anwendung für jeden Teil von Daten von einem OPC-Server vorliegt, muss ein Client (z. B. ein Verfahrensanlagenbetreiber oder ein Ingenieur) unter Umständen zahlreiche Anwendungen entwickeln, um auf Daten auf einem oder mehreren OPC-Servern zuzugreifen. Wenn zudem eine Schnittstelle zu einem OPC-Server modifiziert wird, ein Datenverzeichnis auf einem OPC-Server modifiziert wird und/oder Prozessdaten auf einen anderen OPC-Server migriert wird, muss die entsprechende Anwendung so modifiziert werden, dass sie die Änderungen reflektiert oder übernimmt. Die Notwendigkeit einer regelmäßigen Aktualisierung von Anwendungen zum Zugreifen auf Prozessdaten auf einem OPC-Server kann zu Situationen führen, in denen einige Clients unter Umständen nicht dazu in der Lage sind, auf Prozessdaten zuzugreifen. Änderungen an Prozessdatenpositionen können zudem zu defekten Links zwischen einer Anwendung und den Prozessdaten führen, die auf einem OPC-Server angeordnet sind.

[0030] Die hier beschriebenen beispielhaften Verfahren und Vorrichtungen ermöglichen es einer Client-Anwendung an einem Client-Standort, über einen Wrapper auf jeden beliebigen OPC-Server zuzugreifen. Der beispielhafte Wrapper dient als eine Schnittstelle zwischen dem Client und einem OPC-Server, indem er Datenfelder und/oder Datenanzeigen auf Grundlage der Prozessdaten erzeugt, die vom Client angefordert werden. In einigen Beispielen kann der Wrapper in einem OPC-Server enthalten sein und für einen Client über eine Webserveranwendung und/oder eine Client-Anzeigeanwendung zugänglich sein. In anderen Beispielen kann der Wrapper an einem Client-Standort (z. B. auf einer Rechenvorrichtung, die dem Client und/oder Benutzer zugeordnet ist) installiert sein. In weiteren Beispielen kann der Wrapper in einer Rechenvorrichtung (z. B. einem Server) enthalten sein, die kommunizierend zwischen einem Client und einem OPC-Server gekoppelt ist.

[0031] Der beispielhafte Wrapper kann als eine Webserveranwendung und/oder eine Client-Anzeigevorrichtung implementiert sein. Die Webserveranwendung ist ein Framework (z. B. eine ASP.NET-Framework-Anwendung), die Webseiten erzeugt, bei der Prozessdaten in Datenfelder, Graphen, Diagramme, Tabellen usw. eingebettet sind, auf die ein Client mithilfe eines Webbrowsers zugreifen kann. Im Beispiel der Webserveranwendung verwaltet der Wrapper Prozessdaten (z. B. Objekte) innerhalb einer Webseite. Im Gegensatz dazu ist die Client-Anzeigeanwendung (z. B. eine ActiveX-Steuerungs- und/oder eine Silverlight™-Anwendung) ein Framework, das als eine Runtime-Anwendung (z. B. eine Plug-in-Anwendung) in einem Webbrower angezeigt werden kann. In diesen Beispielen kann die Client-Anzeigeanwendung in einer Rechenvorrichtung am Client-Standort installiert sein. Die Client-Anzeigeanwendung kann innerhalb eines Webbrowsers gestartet werden, wenn der Webbrower auf einen Wrapper, der sich auf dem OPC-Server befindet, zugreift und/oder damit kommuniziert. Außerdem kann die Client-Anzeigeanwendung empfangene Daten und/oder Datenfeldinformationen vom Wrapper konfigurieren, derart, dass die Daten in Datenfelder, Graphen, Diagrammen, Tabellen usw. im Webbrower angezeigt werden.

[0032] Der beispielhafte Wrapper kann einen Adapter aufweisen, der mit einem oder mehreren OPC-Servern kommuniziert. Der Adapter kann Transmission Control Protocol (TCP), Hypertext Transfer Protocol (HTTP) und/oder Extensible Markup Language (XML) benutzen, um mit einem OPC-Server zu kommunizieren und auf Prozessdaten zuzugreifen, die von einem Client angefordert werden. Der beispielhafte Wrapper weist außerdem einen OPC-Wandler auf, um OPC-formatierte Prozessdaten, die vom OPC-Server empfangen werden, in ein Webseitenformat umzuwandeln. Bei einem Webseitenformat kann es sich um Hyper Text Markup Language (HTML) und/oder ein anderes Format zur Anzeige auf einer Client-Anzeigevorrichtung und/oder zur Anzeige in einer Webseite handeln. Ferner weist der Wrapper eine webbasierte Schnittstelle auf, die die Prozessdaten an einen Webbrower übermittelt, der für den Client über eine Webseite und/oder eine Client-Anzeigevorrichtung anschaubar ist.

[0033] Der beispielhafte Wrapper kann Datenfelder für angeforderte Prozessdaten (z. B. Objekte) automatisch erzeugen, konfigurieren und/oder formatieren, indem er einen oder mehrere Typen der angeforderten Prozessdaten ermittelt. Bei den Prozessdatentypen kann es sich um Serveridentifizierungsinformation, Information zur Dateiverzeichnisorganisation, numerische Daten, String-Daten, Statusdaten einer Steuerungsvorrichtung, Alarmdaten und/oder alle anderen Daten handeln, die mit einem Prozesssteuerungssystem, einem Herstellungssystem usw. in Zusammenhang stehen können. In anderen Beispielen kann der Wrapper einen Datentyp anhand der Inhalte der Prozessdaten bestimmen. Durch das Bestimmen der Datentypen kann der beispielhafte Wrapper geeignete Datenfelder mit den entsprechenden Prozessdaten erzeugen, um sie in einem Webbrower anzuzeigen. Bei den Datenfeldern zum Anzeigen der Daten kann es sich um Tabellen,

Textfelder, Zahlenfelder, Graphen, Diagramme, Animationen usw. handeln. In einem Beispiel kann der Wrapper anhand eines Prozessdatentyps und/oder eines Anzeigetyps (z. B. eines Smartphone-Webbrowsers oder eines Arbeitsplatzrechner-Webbrowsers) der Prozesssteuerungsinformationen XSLT-(Extensible Stylesheet Language Transformation)-basierte Vorlagen wählen.

[0034] Durch das automatische Erzeugen von Datenfeldern und das Belegen (z. B. Einbetten) dieser Datenfelder mit angeforderten Prozessdaten befreit der beispielhafte Wrapper den Client von der Notwendigkeit, anforderungsspezifische Anwendungen zu erzeugen, um auf Daten auf OPC-Servern zuzugreifen. Da der beispielhafte Wrapper die Datenfelder auf Grundlage des Datentyps dynamisch formatiert, beeinträchtigen Änderungen am OPC-Server und/oder an Prozessdatenspeicherpositionen innerhalb des OPC-Servers nicht die Fähigkeit eines Client, auf Prozessdaten zuzugreifen. Mit anderen Worten, ein Client kann auf Prozessdaten zugreifen, ohne wissen zu müssen, wie ein OPC-Server verwaltet wird, wo sich die Prozessdaten auf einem OPC-Server befinden, wie Prozessdaten migriert werden und/oder was der Prozessdatentyp ist. Da es ferner mittels des Wrappers möglich ist, über einen Webbrowser auf die Prozessdaten zuzugreifen, muss der Client keine Anwendung erzeugen, die explizit an ein Kommunikationsprotokoll, ein Schnittstellenprotokoll und/oder ein Betriebsprotokoll oder einen bestimmten OPC-Server gebunden ist. Außerdem kann der beispielhafte Wrapper unabhängig von den Protokollen, die einem OPC-Server zugeordnet sind, für jeden beliebigen OPC-Server implementiert sein. Da der beispielhafte Wrapper zudem nicht an ein bestimmtes OPC-Serverprotokoll gebunden ist, kann der Wrapper effizienter installiert und/oder gewartet werden als OPC-Server-spezifische Anwendungen.

[0035] [Fig. 1](#) ist ein Blockdiagramm, das eine Steuerungsumgebung **100** darstellt, wobei ein beispielhaftes Steuerungssystem **102** kommunizierend an einen OPC-(Open Packaging Convention)-Server **104** gekoppelt ist, auf den ein Client **106** mittels eines Webseitenservers **108** zugreifen kann, der einen Wrapper **110** aufweist. Obwohl der Webseitenserver **108** und/oder der Wrapper **110** als kommunizierend an den OPC-Server **104**, der Prozessdaten vom Steuerungssystem **102** empfängt, gekoppelt dargestellt sind, können der Webseitenserver **108** und/oder der Wrapper **110** auch an andere OPC-Server gekoppelt sein, die Prozessdaten von anderen Systemen, Herstellungsanlagen, Automatisierungsanlagen, Industriesystemen usw. empfangen. Obwohl zudem der beispielhafte OPC-Server **104** als kommunizierend an das Steuerungssystem **102** gekoppelt dargestellt ist, kann der OPC-Server **104** auch an andere Steuerungssysteme kommunizierend gekoppelt sein. Außerdem kann die beispielhafte Steuerungsumgebung **100** weitere Clients (nicht dargestellt) aufweisen, die über den Webseitenserver **108** und/oder andere Webseitenserver (nicht dargestellt) kommunizierend an den OPC-Server **104** gekoppelt sein können.

[0036] Das beispielhafte Steuerungssystem **102** kann jeden beliebigen Typ von Herstellungsanlage, Prozessanlage, Automatisierungsanlage und/oder jeden anderen Typ von Prozesssteuerungsstruktur oder -system umfassen. In einigen Beispielen kann das Steuerungssystem **102** mehrere Anlagen an unterschiedlichen Standorten aufweisen. Obwohl das beispielhafte Steuerungssystem **102** ein Prozesssteuerungssystem **112** zeigt, kann das Steuerungssystem **102** auch weitere Prozesssteuerungssysteme aufweisen.

[0037] Das beispielhafte Prozesssteuerungssystem **112** ist über einen Datenbus **116** kommunizierend an eine Steuerung **114** gekoppelt. Das Prozesssteuerungssystem **112** kann eine beliebige Anzahl von Feldgeräten (z. B. Eingabe- und/oder Ausgabevorrichtungen) aufweisen. Bei den Feldgeräten kann es sich um jeden beliebigen Typ von Prozesssteuerungskomponente handeln, die dazu in der Lage ist, Eingaben zu empfangen, Ausgaben zu erzeugen und/oder einen Prozess zu steuern. Bei den Feldgeräten kann es sich beispielsweise um Vorrichtungen wie Ventile, Pumpen, Gebläse, Heizungen, Kühler und/oder Mischer zum Steuern eines Prozesses handeln. Außerdem kann es sich bei den Feldgeräten um Ausgabevorrichtungen wie z. B. Thermometer, Druckmesser, Konzentrationsmesser, Fluidpegelmesser, Durchflussmesser und/oder Dampfsensoren zum Messen von Teilen eines Prozesses handeln. Die Eingabevorrichtungen können von der Steuerung **114** Anweisungen zum Ausführen eines bestimmten Befehls empfangen und eine Veränderung des Prozesses bewirken. Ferner können die Ausgabevorrichtungen Prozessdaten, Umgebungsdaten und/oder Eingabevorrichtungsdaten messen und die gemessenen Daten als Prozesssteuerungsinformation (z. B. Prozessdaten) an die Steuerung **114** übertragen. Diese Prozessdaten können die Werte von Variablen (z. B. gemessenen Prozessvariablen und/oder gemessenen Qualitätsvariablen) enthalten, die einer Messwertausgabe einem jeweiligen Feldgerät entsprechen.

[0038] In dem dargestellten Beispiel aus [Fig. 1](#) kann die beispielhafte Steuerung **114** über den Datenbus **116** mit den Feldgeräten im Prozesssteuerungssystem **106** kommunizieren. Dieser Datenbus **116** kann an Kommunikationskomponenten innerhalb des Prozesssteuerungssystems **112** gekoppelt sein. Bei den Kommunikationskomponenten kann es sich um E/A-Karten zum Empfang von Daten von den Feldgeräten und zum Um-

wandeln der Daten in ein Kommunikationsmedium handeln, das von der beispielhaften Steuerung **114** empfangen werden kann. Zusätzlich können diese E/A-Karten Daten von der Steuerung **114** in ein Datenformat umwandeln, das von den entsprechenden Feldgeräten verarbeitet werden kann. In einem Beispiel kann der Datenbus **116** unter Verwendung des Feldbusprotokolls oder anderer Typen drahtgebundener und/oder drahtloser Kommunikationsprotokolle (z. B. Profibus-Protokoll, HART-Protokoll usw.) implementiert sein.

[0039] Die Steuerung **114** ist über eine beliebige drahtgebundene und/oder drahtlose Verbindung kommunizierend an den OPC-Server **104** gekoppelt. In einigen Beispielen kann die Verbindung einen Firewall und/oder einen anderen Sicherheitsmechanismus aufweisen, um den Zugriff auf die Steuerung **114** einzuschränken. Die Steuerung **114** kann Prozessdaten an den OPC-Server **104** übertragen, sobald die Steuerung **114** die Prozessdaten vom Prozesssteuerungssystem **112** empfängt. In anderen Beispielen kann die Steuerung **114** Prozessdaten in regelmäßigen Zeitintervallen (z. B. minütlich, stündlich, täglich usw.) an den OPC-Server **104** übertragen. Alternativ kann der OPC-Server **104** Prozessdaten von der Steuerung **114** anfordern.

[0040] Werden die Prozessdaten empfangen, speichert der beispielhafte OPC-Server **104** aus [Fig. 1](#) die Prozessdaten in einem Dateisystem. Das Dateisystem kann in hierarchischer Weise geordnet sein, wobei Verzeichnisse und/oder Unterverzeichnisse auf den Vorrichtungen im Prozesssteuerungssystem **112** basieren und/oder auf einer Routine (z. B. einer Anwendung und/oder einem Algorithmus) basieren, die innerhalb der Steuerung **114** abläuft, um das Prozesssteuerungssystem **112** zu verwalten. In anderen Beispielen kann das Dateisystem von einem Betreiber des Steuerungssystems **102** geordnet werden. Die Prozessdaten können an 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 einem Dienstprogramm zugeordnet ist, das in der Steuerung **114** arbeitet, oder einer Feldgerätausgabe innerhalb des Prozesssteuerungssystems **112** zugeordnet ist. Der Parameter kann Metadaten enthalten, die den Typ der Prozessdaten beschreiben, die dem Parameter zugeordnet sind.

[0041] Den einzelnen Verzeichnissen, Unterverzeichnissen, Dateien und/oder Parameter kann ein Endpunkt zugewiesen sein. Dem OPC-Server **104** kann auch ein Endpunkt zugewiesen sein. Diese Endpunkte können nach Sicherheitszugriff, Lesezugriff, Bezugszugriff und/oder Schreibzugriff gruppiert sein. Ein Endpunkt kann eine Adresse, ein Bindungselement und/oder ein Vertragselement aufweisen, das der Wrapper **110** dazu benutzen kann, auf die Prozessdaten zuzugreifen, die auf dem OPC-Server **104** gespeichert sind. Der Wrapper **110** kann mithilfe der Endpunkte auf die Prozessdaten zugreifen, wie es in der US-Patentanmeldung Nr. 12/637,439 beschrieben ist, die hiermit in den vorliegenden Gegenstand vollständig mit einbezogen wird.

[0042] Der beispielhafte Wrapper **110** aus [Fig. 1](#) ist im Webseitenserver **108** enthalten. Der Webseitenserver **108** ist eine Vorrichtung und/oder eine Anwendung, die als eine Schnittstelle zwischen dem OPC-Server **104** und dem Client **106** dient. In einigen Beispielen kann der Webseitenserver **108** als eine Schnittstelle im OPC-Server **104** enthalten sein. In anderen Beispielen kann der Webseitenserver **108** in einer Rechenvorrichtung am Client **106** installiert sein. In weiteren Beispielen kann der Webseitenserver **108** auf einem Server oder einer anderen Rechenvorrichtung implementiert sein, die den Client **106** kommunizierend an den OPC-Server **104** koppelt. Der beispielhafte Wrapper **110** innerhalb des Webseitenservers **108** empfängt vom Client **106** Anforderungen zum Zugriff auf Prozessdaten (z. B. Objekte), fordert die Prozessdaten vom OPC-Server **104** an, wandelt die Prozessdaten in ein Format um, das für den Client anschaubar ist, und erzeugt und/oder formatiert Datenfelder, um die Prozessdaten zur Anzeige in einem Webbrowser einzubetten. Der beispielhafte Wrapper **110** kann außerdem Sicherheitsmerkmale wie z. B. eine Benutzerauthentifizierung und/oder Verschlüsselung enthalten, um sicherzustellen, dass nur berechtigte Benutzer die Prozessdaten anschauen und/oder auf sie zugreifen können.

[0043] Um auf den OPC-Server **104** zuzugreifen, weist der beispielhafte Wrapper **110** einen Adapter **118** auf. Der beispielhafte Adapter **118** kann TCP-, HTTP- und/oder XML-basierte Kommunikation verwenden, um über eine beliebige drahtgebundene und/oder drahtlose Verbindung mit dem OPC-Server **104** zu kommunizieren. Der Wrapper **110** weist ferner einen OPC-Wandler **120** auf, um die Prozessdaten, die vom Adapter **120** empfangen werden, in ein Format (z. B. XSLT-basierte Vorlagen) umzuwandeln, das über einen Webbrowser anschaubar ist. Der beispielhafte Wrapper **110** weist außerdem eine webbasierte Schnittstelle **122** auf, um eine Schnittstelle zum Client **106** bereitzustellen, um die Prozessdaten zu formatieren, auszuführen, einzubetten und/oder anzuzeigen.

[0044] Der beispielhafte Client **106** kann einer Person zugeordnet sein, die dazu berechtigt ist, Prozessdaten, die auf dem OPC-Server **104** gespeichert sind, zu lesen, zu schreiben und/oder zu beziehen. Der Client **106** kann auch Personen zugeordnet sein, die dem Steuerungssystem **102** zugeordnet sind und die von einem

entfernten Standort auf den OPC-Server **104** zugreifen können. Der Client **106** kann unter Verwendung eines drahtgebundenen und/oder drahtlosen Kommunikationsmediums (z. B. Internet) über den Webseitenserver **108** auf den OPC-Server **104** zugreifen.

[0045] In einem Beispiel, in dem der Client **106** eine Anforderung zum Zugreifen auf Prozessdaten (z. B. Objekte) erzeugt, empfängt der Wrapper **110** eine Anforderungsnachricht vom Client **106**. Insbesondere kann die webbasierte Schnittstelle **122** die Anforderung empfangen. Nach Empfang der Anforderung leitet die webbasierte Schnittstelle **122** die Anforderung weiter an den Adapter **118**. Der Adapter **118** benutzt die Informationen in der Anforderung (z. B. eine URL-(Uniform Resource Locator)-Zieladresse), um einen OPC-Server (z. B. den OPC-Server **104**) zu identifizieren, auf dem die angeforderten Prozessdaten gespeichert sind. Der Adapter **118** greift dann auf den OPC-Server **104** zu, um die Prozessdaten abzurufen. Die Prozessdaten können gemäß einem Interoperabilitäts-Datenverpackungsformat, das dem OPC entspricht, auf dem OPC-Server **104** gespeichert sein. Der Adapter **118** kann Endpunkte, die den Prozessdaten zugeordnet sind, dazu benutzen, auf die Prozessdaten zuzugreifen und/oder diese zu lesen. Der Adapter **118** leitet dann die Prozessdaten, die er vom OPC-Server **104** empfangen hat, an den OPC-Wandler **120** weiter, der die Prozessdaten aus einem Format, das dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, in ein Webbrowserformat umwandelt. Der OPC-Wandler **120** leitet die umgewandelten Prozessdaten dann an die webbasierte Schnittstelle **122** weiter. Die webbasierte Schnittstelle **122**bettet sodann wenigstens einen Teil der umgewandelten Prozessdaten in ein oder mehrere entsprechende Datenfelder (oder Vorlagen) ein und/oder setzt sie darin ein, um sie über einen Webbrowser auf einer Webseite anzuzeigen, die für den Client **106** anschaubar ist.

[0046] Die webbasierte Schnittstelle **122** kann die Prozessdaten in ein oder mehrere Datenfelder einbetten, indem sie die umgewandelten Prozessdaten nach einem Datentyp unterteilt, der den Prozessdaten zugeordnet ist, und ermittelt, welcher Datentyp einem Datenfeld zugeordnet ist, und ein Datenfeld für jeden Datentyp erzeugt, und die Daten, die den einzelnen Datentypen zugeordnet sind, in das zugeordnete Datenfeld setzt. Die webbasierte Schnittstelle **122** kann einen Datentyp aus den Prozessdaten ermitteln, indem sie Metadaten ermittelt, die den einzelnen Teilen der Prozessdaten zugeordnet sind, und die Metadaten dem zugeordneten Datenfeld zuweist.

[0047] Die beispielhafte webbasierte Schnittstelle **122** ist dazu in der Lage, die Prozessdaten derart zu formatieren, dass die Prozessdaten mithilfe einer Client-Anwendung **124** für den Client **106** anschaubar sind. Das Beispiel aus [Fig. 1](#) zeigt, dass die Client-Anwendung **124** Prozessdaten an einer Schnittstelle **126** anzeigt, die einen Webbrowser beinhaltet kann. Die Client-Anwendung **124** kann eine Webserver-Anwendung und/oder eine Client-Anzeigevorrichtung aufweisen. Der Wrapper **110** kann Prozessdaten für eine Webserver-Anwendung formatieren, indem er eine Webseite erzeugt und/oder auf eine Vorlagenwebseite zugreift und die Datenfelder in die Webseite einsetzt oder einbettet. Die Schnittstelle **126** kann dann über einen Webbrowser die Prozessdaten anzeigen, indem sie mithilfe von HTML-Anforderungen und -Antworten auf die Webseite, die im Wrapper **110** angeordnet ist, und/oder den Webseitenserver zugreift. Alternativ kann der Wrapper **110** die Prozessdaten für eine Client-Anzeigeanwendung formatieren, indem er eine Webanwendung (z. B. ActiveX, Adobe Flash™ und/oder Silverlight™) an der Client-Anwendung **124** startet, die innerhalb eines Webbrowsers (z. B. der Schnittstelle **126**) ausführbar ist. In einigen Beispielen kann der Client **106** die Client-Anzeigeanwendung vor dem Anschauen der Prozessdaten herunterladen und/oder installieren. Der Wrapper **110** überträgt die Prozessdaten und/oder die Datenfelder an die Client-Anzeigeanwendung. In einigen Beispielen ordnet der Wrapper **110** die Prozessdaten den entsprechenden Datenfeldern zu, bevor er die Prozessdaten an die Client-Anzeigeanwendung überträgt. Werden die Prozessdaten empfangen, erzeugt die Client-Anzeigeanwendung eine Anzeige im Webbrowser (z. B. der Schnittstelle **126**) (führt sie also z. B. aus), damit die Prozessdaten in den entsprechenden Datenfeldern anschaubar werden.

[0048] Außerdem kann der Client **106** die Client-Anwendung **124** durch das Modifizieren von Datenfeldern und/oder das Anzeigen der Prozessdaten im Webbrowser anpassen. Beispielsweise kann der Client **106** eine Position innerhalb des Webbrowsers festlegen, an der ein Datenfeld angezeigt wird. Ferner kann der Client **106** die Farbe, die Schriftgröße, die Zahlerschreibweise und/oder sämtliche anderen grafischen Darstellungen der Prozessdaten in den Datenfeldern modifizieren.

[0049] Der Client **106** kann auf den OPC-Server **104** zugreifen, um Prozessdaten zu lesen, zu schreiben und/oder zu beziehen. Das Beziehen der Prozessdaten kann das Erhalten einer Berechtigung von dem beispielhaften Wrapper **110** und/oder dem OPC-Server **104** dazu beinhalten, regelmäßige und/oder kontinuierliche Aktualisierungen angeforderter Prozessdaten zu empfangen, wenn die Prozessdaten von der Steuerung **114** übertragen werden. Das Lesen der Prozessdaten kann das Lesen des aktuellen Werts der Prozessdaten beinhalten, die auf dem OPC-Server **104** gespeichert sind. Das Schreiben der Prozessdaten kann das Empfan-

gen von Werten vom Client **106** beinhalten, um einen Parameter, der als Prozessdaten auf dem OPC-Server **104** gespeichert ist, zu modifizieren oder zu ändern. Das Schreiben der Daten kann auch das Modifizieren eines Status, eines Alarms und/oder eines Flags beinhalten, der bzw. das den Prozessdaten zugeordnet ist. Wird ein geschriebener Wert empfangen, kann der beispielhafte OPC-Server **104** den geschriebenen Wert an die Steuerung **114** übertragen, um einen Betriebsvorgang des Prozesssteuerungssystems **112** zu ändern und/oder zu modifizieren. Um dem Client **106** den Zugriff auf die Prozessdaten zu ermöglichen, kann der beispielhafte Wrapper **110** Sicherheitsfunktionen implementieren, darunter Verschlüsselung, Authentifizierung, Integritätscodes und/oder benutzerspezifische Zugriffskontrolllisten. In Beispielen, in denen ein Benutzer und/oder der Client **106** nicht für den Zugriff auf Prozessdaten berechtigt ist, kann der beispielhafte Wrapper **110** ausschließlich Lesezugriff auf die Prozessdaten bereitstellen oder alternativ keinerlei Zugriff auf die Prozessdaten zulassen.

[0050] [Fig. 2](#) ist ein Funktionsblockdiagramm des beispielhaften Wrappers aus [Fig. 1](#). Der Wrapper **110** weist den Adapter **118**, den OPC-Wandler **120** und die webbasierte Schnittstelle **122** aus [Fig. 1](#) auf. Die einzelnen Funktionsblöcke im Wrapper **110** aus [Fig. 2](#) können eine Vielzahl von Clients und/oder OPC-Server unterstützen, oder der Wrapper **110** kann alternativ auch Funktionsblöcke für jeden kommunizierend angekoppelten OPC-Server und/oder Funktionsblöcke für jeden Client (z. B. den Client **106**) aufweisen.

[0051] Um Sicherheitsmaßnahmen wie Verschlüsselung und/oder Endpunktzugriffskontrolle für die Kommunikation mit dem Client **106** und/oder der Client-Anwendung **124** bereitzustellen, weist der beispielhafte Wrapper **110** aus [Fig. 2](#) eine Sicherheitsverarbeitungseinheit **202** auf. Die beispielhafte Sicherheitsverarbeitungseinheit **202** kann beispielsweise Verschlüsselungsverarbeitungseinheiten und/oder Digitalsignaturgeneratoren aufweisen, um die ausgehende Kommunikation vor nicht berechtigten Dritten zu schützen. Die Verschlüsselungsverarbeitungseinheiten können einen beliebigen Typ von Verschlüsselungscodierer benutzen, um Kommunikation, die für die Client-Anwendung **124** bestimmt ist, in einem Format zu formatieren, das für nicht berechtigte Benutzer unlesbar ist. Die Digitalsignaturgeneratoren schützen die Kommunikation für die Client-Anwendung **124** vor Manipulation durch nicht berechtigte Dritte. Die Digitalsignaturgeneratoren können jeden beliebigen Typ eines kryptographisch sicheren Signaturgenerators (z. B. Hash-Codes) benutzen, der das Erkennen von Werten ermöglicht, die von nicht berechtigten Dritten zwischen der Client-Anwendung **124** und dem Wrapper **110** modifiziert wurden. Außerdem kann die Sicherheitsverarbeitungseinheit **202** andere Formen der Kommunikationssicherheit aufweisen, darunter Authentifizierungsmechanismen und/oder Zugriffskontrollen. Die beispielhafte Sicherheitsverarbeitungseinheit **202** kann verschlüsselte und/oder signierte Kommunikation decodieren, die von der Client-Anwendung **124** und/oder dem Client **106** stammt. Nach dem Decodieren der Kommunikation überträgt die Sicherheitsverarbeitungseinheit **202** die Kommunikation an den vorgesehenen Zielort im Wrapper **110**.

[0052] In dem Beispiel aus [Fig. 2](#) ist die Sicherheitsverarbeitungseinheit **202** kommunizierend an einen oder mehrere Clients einschließlich des Clients **106** gekoppelt. Die Sicherheitsverarbeitungseinheit **202** kann Anforderungsnachrichten von den Clients und/oder anderen Individuen anhand von Identifizierungsinformation herausfiltern, derart, dass nur berechtigte Clients auf die Prozessdaten auf einem gewünschten OPC-Server zugreifen können. Außerdem kann die Sicherheitsverarbeitungseinheit **202** Prozessdaten und/oder Datenfelder an eine Client-Anzeigeanwendung weiterleiten, die in der Client-Anwendung **124** implementiert ist. In anderen Beispielen ermöglicht die Sicherheitsverarbeitungseinheit **202** eine sichere Kommunikation zwischen dem Client **106** und der webbasierten Schnittstelle **122**, die eine Webseite aufweisen kann, welche Prozessdaten anzeigt.

[0053] Um die Datenkommunikation zwischen einem oder mehreren Client(s) und der beispielhaften webbasierten Schnittstelle **122** zu verwalten, weist der beispielhafte Wrapper **110** aus [Fig. 2](#) eine Sitzungssteuereinrichtung **204** auf. Die beispielhafte Sitzungssteuereinrichtung **204** verwaltet eine Zugriffssitzung eines Clients (z. B. des Clients **106** aus [Fig. 1](#)), der mit dem Wrapper **110** kommuniziert. Die Zugriffssitzung stellt einen offenen Kommunikationsweg zwischen der webbasierten Schnittstelle **122** und einem Client dar. Eine Zugriffssitzung kann für jeden einzelnen Client erzeugt werden, der auf die webbasierte Schnittstelle **122** zugreift, da jeder Client Zugriff auf unterschiedliche Datenquellen und/oder Datentypen von unterschiedlichen OPC-Servern anfordern kann. Auf diese Weise sorgt die beispielhafte Sitzungssteuereinrichtung **204** dafür, dass die webbasierte Schnittstelle **122** nur die von einem Client angeforderten Prozessdaten an den Client bereitstellt.

[0054] Die Sitzungssteuereinrichtung **204** leitet eine Sitzung ein, wenn sie eine Anforderungsnachricht zum Zugreifen auf Prozessdaten von einem Client empfängt. Die Anforderung kann in Form einer Anforderung über einen Webbrowser und/oder über die Client-Anwendung **124** erfolgen. Bis eine Sitzung von der Sitzungssteuereinrichtung **204** eröffnet wird, kann die Sitzungssteuereinrichtung **204** andere Anforderungen eines Clients

zurückweisen. Während die Sitzung offen ist, leitet die Sitzungssteuereinrichtung **204** jede Anforderungsnachricht von einem Client an die webbasierte Schnittstelle **122** weiter. Zusätzlich kann die Sitzungssteuereinrichtung **204** Verweise auf ausgewählte Prozessdaten und ihre zugeordneten Lese- oder Schreibendpunkte speichern, die den Datenfeldern und/oder Prozessdaten zugeordnet sind, die auf einer Webseite anschaubar sind.

[0055] Die beispielhafte webbasierte Schnittstelle **122** im Wrapper **110** aus [Fig. 2](#) stellt eine Schnittstelle zu den Clients bereit, indem sie Prozessdaten verwaltet, formatiert und/oder konfiguriert. Die webbasierte Schnittstelle **122** empfängt vom OPC-Wandler **120** Prozessdaten (z. B. Objekte) in einem Format, das auf einem Webbrower anschaubar ist. Die beispielhafte webbasierte Schnittstelle **122** verarbeitet außerdem Anforderungen von Clients für den Zugriff auf Prozessdaten. Wird eine Anforderung von einem Client empfangen, leitet die webbasierte Schnittstelle **122** eine Zugriffssitzung ein, indem sie eine Anweisung an die Sitzungssteuereinrichtung **204** sendet und die Anforderung an den Adapter **118** weiterleitet. In einigen Beispielen empfängt die webbasierte Schnittstelle **122** die Anforderung, nachdem die Sicherheitsverarbeitungseinrichtung **202** ermittelt hat, dass die Anforderung einem authentifizierten Client zugeordnet ist. Außerdem kann die webbasierte Schnittstelle **122** eine Zugriffssitzung schließen, wenn ein Client einen Webbrower schließt und/oder beendet und/oder wenn eine Client-Anwendung eine Anweisung zum Beenden der Sitzung sendet.

[0056] Wenn die webbasierte Schnittstelle **122** Prozessdaten empfängt, die einer Anforderung zugeordnet sind, ermittelt die webbasierte Schnittstelle **122** einen oder mehrere Datentyp(en), die den Prozessdaten zugeordnet sind, indem sie die Prozessdaten an eine Datenverarbeitungseinheit **206** weiterleitet. Die beispielhafte Datenverarbeitungseinheit **206** identifiziert Metadaten, die in Teilen der Prozessdaten enthalten sein können. Alternativ greift die Datenverarbeitungseinheit **206** auf eine Datentypdatenbank **208** auf, um anhand eines Wertetyps, einer Eigenschaft, eines VariablenTyps und/oder eines anderen Identifikators, der den Prozessdaten zugeordnet ist, die Prozessdaten einem entsprechenden Datenfeld zuzuweisen. Die beispielhafte Datentypdatenbank **208** kann als EEPROM (Electronically Erasable Programmable Read-Only Memory, elektronisch löscherbarer programmierbarer Lesespeicher), RAM (Random Access Memory, Lese-/Schreibspeicher), ROM (Read-Only Memory, Lesespeicher) und/oder jeder andere Speichertyp implementiert sein.

[0057] Die beispielhafte Datenverarbeitungseinrichtung **206** aus [Fig. 2](#) ordnet Prozessdaten einem Datenfeld zu und/oder konfiguriert sie, indem sie, wenn ein oder mehrere Datentypen ermittelt wurden, die den Prozessdaten zugeordnet sind, die Prozessdaten einem oder mehreren Datenanzeigeveriablen eines Datenfelds zuweist, das sich auf einen Datentyp bezieht. Die Datenverarbeitungseinrichtung **206** und/oder die webbasierte Schnittstelle können auch die numerischen, alphanumerischen und/oder Flag-Werte der Prozessdaten im passenden Datenfeld speichern.

[0058] Außerdem kann die Datenverarbeitungseinrichtung **206** auf die Datentypdatenbank **206** zugreifen, um eine Vorlage zum Anzeigen, der Prozessdaten zu erhalten. Bei der Vorlage kann es sich um eine XSLT-Vorlage handeln, die XML-Information (z. B. Prozessdaten) für die Anzeige auf einem Webbrower in HTML umwandelt. Die Datenverarbeitungseinrichtung **206** kann eine Vorlage aufgrund eines Prozessdatentyps, von Werten der Prozessdaten, Eigenschaften der Prozessdaten und/oder Zielanzeigeverinformation auswählen. Die Zielanzeigeverinformation kann einen Typ von Webbrower und/oder eine Vorrichtung identifizieren, die die Prozessdaten anzeigen soll. Beispielsweise kann die Zielanzeigeverinformation angeben, ob die Prozessdaten auf einem Smartphone-Webbrower oder einem Webbrower an einem Arbeitsplatzrechner angezeigt werden sollen. Eine Vorlage für ein Smartphone kann dazu konfiguriert sein, Prozessdaten auf einem relativ kleinen Bildschirm anzuzeigen, im Gegensatz zu einer Vorlage für einen Arbeitsplatzrechner-Webbrower.

[0059] Die webbasierte Schnittstelle **122** empfängt den oder die Datentypen der Prozessdaten von der Datenverarbeitungseinrichtung **206** und stellt die Datenfelder mit den Prozessdaten zusammen. Die webbasierte Schnittstelle **122** kann dann die Prozessdaten (z. B. die zusammengestellten Teile der Prozessdaten, die den jeweiligen Datenfeldern zugeordnet sind) in die passenden Datenfelder einer Vorlage einbetten und/oder einsetzen. Die webbasierte Schnittstelle **122**bettet die Prozessdaten in das oder die Datenfeld(er) ein, indem sie mithilfe der Information und/oder der Vorlage(n) von der Datenverarbeitungseinrichtung **206** die Prozessdaten nach Datentyp(en) unterteilt, der oder die den Prozessdaten zugeordnet sind, um zu ermitteln, welcher Datentyp einem Datenfeld zugeordnet ist, und ein Datenfeld für jeden Datentyp erzeugt, und die Daten, die den einzelnen Datentypen zugeordnet sind, in das zugeordnete Datenfeld setzt. In einigen Beispielen kann ein Datenfeld mehr als einem Datentyp zugeordnet sein. Die webbasierte Schnittstelle **122** kann dann die Prozessdaten in eine Webseite einbetten, die das oder die Datenfelder enthält.

[0060] In Beispielen, in denen ein Client Prozessdaten mithilfe einer Client-Anwendung (z. B. der Client-Anwendung **124**) anfordert, ermittelt die beispielhafte webbasierte Schnittstelle **122** den Anwendungstyp. Die

webbasierte Schnittstelle **122** kann den Typ der Client-Anwendung ermitteln, indem sie ein Protokoll und/oder eine Anwendungssprache identifiziert, das oder die der Anforderungsnachricht zugeordnet ist, und/oder Metadaten in der Anforderungsnachricht, die der Client-Anwendung zugeordnet sind. Beispielsweise benutzt ein Client eine Webserveranwendung, die Protokolle aufweist, die der Anwendung ASP.NET zugeordnet sind, während ein Client, der eine Client-Anzeigeanwendung benutzt, Protokolle und/oder Nachrichten aufweisen kann, die einer Silverlight™ oder einer ActiveX-Anwendung zugeordnet sind. Je nach Typ der Client-Anwendung erzeugt und/oder konfiguriert die beispielhafte webbasierte Schnittstelle **122** den passenden Webbrowsers und/oder eine Programmschnittstelle, um die Prozessdaten in den Datenfeldern anzuzeigen.

[0061] Nach dem Ermitteln eines Client-Anwendungstyps formatiert die webbasierte Schnittstelle **122** und/oder die Datenverarbeitungseinrichtung **206** die Prozessdaten und/oder die Datenfelder zur Anzeige. In Beispielen, in denen die Client-Anwendung **124** einer Webserveranwendung zugeordnet ist, empfängt die webbasierte Schnittstelle **122** über einen Webbrowsers eine Anforderung vom Client **106**, leitet die Anforderung an den Adapter **118** weiter, um auf die Prozessdaten zuzugreifen, und empfängt die Prozessdaten vom OPC-Wandler **120**. Die Anforderung vom Client **106** über den Webbrowsers kann in der Form eines HTML-Dokuments erfolgen. Ferner kann die webbasierte Schnittstelle **122** eine Anweisung an die Sitzungssteuereinrichtung **204** senden, um den Webbrowsers des Clients **106** der neu erzeugten Zugriffssitzung zuzuordnen. Die webbasierte Schnittstelle **122** erzeugt dann eine Webseite mit den Datenfeldern, die die Prozessdaten enthalten. Die Datenfelder können beispielsweise Listen, Tabellen, Graphen, Diagramme, grafische Anzeigen, Animationen usw. enthalten. Ferner können die Positionen der Datenfelder von einer Vorlage festgelegt sein. Die webbasierte Schnittstelle **122** überträgt dann die Inhalte der Webseite an den Webbrowsers des Clients **106**, um die Webseite am Standort des Clients anzuzeigen. Die Prozessdaten auf der Webseite werden in einem formatierten Kontext über ein HTTP-, XML-, XSLT- und/oder beliebiges anderes Internet-Websiten-Übertragungsformat vom Webbrowsers an den Client **106** übertragen.

[0062] In Beispielen, in denen der Webbrowsers eine Client-Anzeigeanwendung aufweist, empfängt die beispielhafte webbasierte Schnittstelle **122** über einen Webbrowsers eine Anforderung vom Client **106**, leitet die Anforderung an den Adapter **118** weiter, um auf die Prozessdaten zuzugreifen, empfängt die Prozessdaten vom OPC-Wandler **120** und startet die Client-Anzeigeanwendung im Webbrowsers. Die Anforderung vom Client **106** über den Webbrowsers kann in der Form eines Methodenaufrufs erfolgen. Nach dem Formatieren und/oder Konfigurieren der Prozessdaten überträgt die webbasierte Schnittstelle **122** die Prozessdaten in einem Format, das Protokollen der Client-Anzeigeanwendung zugeordnet ist, an einen Client. Die Client-Anzeigeanwendung erzeugt dann eine Anzeige im Webbrowsers am Client-Standort, die die Prozessdaten in den zugeordneten Datenfeldern zeigt.

[0063] Die Client-Anzeigeanwendung kann einen beliebigen Programm-Client aufweisen, der in einigen Beispielen ohne einen Webbrowsers auf die webbasierte Schnittstelle **122** zugreifen kann. In diesen Beispielen ordnet die webbasierte Schnittstelle **122** den Programm-Client der neu erzeugten Zugriffssitzung zu und leitet Anforderungsprozessdaten und/oder Datenfelder über Protokolle, die dem Programm-Client zugeordnet sind, an den Client weiter. Der Programm-Client erzeugt dann eine Anzeige, die die Prozessdaten mit den zugeordneten Datenfeldern zeigt.

[0064] Die beispielhafte webbasierte Schnittstelle **122** bietet den Clients Lesezugriff, Schreibzugriff und/oder Bezugszugriff. Für Clients, die Lesezugriff anfordern, leitet die webbasierte Schnittstelle **122** eine einzelne Anforderung aktueller Prozessdaten an den Adapter **118** weiter. Nach dem Empfangen und Formatieren der Prozessdaten stellt die webbasierte Schnittstelle **122** die angeforderten Prozessdaten an die Client-Anwendung (en) bereit.

[0065] Wenn der Client Bezugszugriff anfordert, kann die webbasierte Schnittstelle **122** alternativ regelmäßige Nachrichten an den Adapter **118** senden, um die Prozessdaten in Zeitintervallen zu empfangen. In einigen Beispielen kann der Client die Zeitintervalle zum Empfangen der Prozessdaten festlegen. Nachdem die webbasierte Schnittstelle **122** die Webseite und/oder die Anzeigekonfiguration für eine Programmanwendung erzeugt hat, stellt die webbasierte Schnittstelle **122** regelmäßige und/oder kontinuierliche Aktualisierungen der Datenfelder mit neueren Prozessdaten bereit. Die webbasierte Schnittstelle **122** stellt die Aktualisierungen über eine bereits aktive Zugriffssitzung bereit, die die Kommunikation mit einem anfordernden Client aufrechterhält. Die Aktualisierungen der Datenfelder können Trendkurven, Prozessstatusalarme und/oder Flags beinhalten, wobei die neuesten Prozessdaten auf dem entsprechenden OPC-Server gespeichert sind. Auf diese Weise ermöglicht die webbasierte Schnittstelle **122** es einem Client, auf die neuesten Prozessdaten zuzugreifen, ohne einen Webbrowsers aktualisieren zu müssen und/oder ohne die Prozessdaten regelmäßig anfordern zu müssen.

[0066] In weiteren Beispielen, in denen Clients Schreibzugriff anfordern können, empfängt die beispielhafte webbasierte Schnittstelle **122** über ein HTML-Dokument oder einen Methodenaufruf den geschriebenen Prozessdatenwert von der Webseite und/oder Programmanwendung. Die webbasierte Schnittstelle **122** identifiziert sodann eine Variable, die dem geschriebenen Datenwert zugeordnet ist. In anderen Beispielen kann die webbasierte Schnittstelle **122** auf die Datenverarbeitungseinrichtung **206** zugreifen, um den Datenwert einem Datentyp zuzuweisen, der in der Datentypdatenbank **208** festgelegt ist. Die webbasierte Schnittstelle **122** sendet dann eine Anweisung an den OPC-Wandler **120** und/oder den Adapter **118**, um den Datenwert an die passende Variable und/oder den passenden Teil des OPC-Servers weiterzuleiten. Der OPC-Server kann dann den geschriebenen Datenwert speichern und/oder den geschriebenen Datenwert am passenden Ort in einer Steuerung speichern.

[0067] Die beispielhafte webbasierte Schnittstelle **122** kann Client-Anpassungsinformation in einer Datenbank (nicht dargestellt) speichern, damit die webbasierte Schnittstelle **122** beim nächsten Mal, wenn derselbe Client denselben Typ von Prozessdaten anfordert, die Prozessdaten in dem oder den Datenfeldern teilweise auf Grundlage der vorherigen Anpassung des Clients formatieren kann. Die webbasierte Schnittstelle **122** kann die Client-Anpassung durch beliebige Modifikationen ermitteln, die von einem Client über einen Webbrowser und/oder eine Programmanwendung ausgeführt werden, um das Erscheinungsbild und/oder die Datenanzeige der Prozessdaten in dem oder den Datenfeldern zu verändern. Ein Client kann eine Farbe, eine Schriftgröße, eine Zahlenschreibweise und/oder sämtliche anderen grafischen Darstellungen der Prozessdaten modifizieren.

[0068] Um auf OPC-Server (z. B. den OPC-Server **104**) zuzugreifen, weist der beispielhafte Wrapper **110** aus [Fig. 2](#) den Adapter **118** auf. Der beispielhafte Adapter **118** kann TCP, HTTP, XML und/oder ein beliebiges anderes Kommunikationsprotokoll verwenden, um über eine beliebige drahtgebundene und/oder drahtlose Verbindung mit dem oder den OPC-Servern zu kommunizieren. Der Adapter **118** empfängt von der webbasierten Schnittstelle **122** Anforderungen zum Zugreifen auf einen OPC-Server. Wird eine Anforderung empfangen, identifiziert der Adapter **118** einen OPC-Server, auf dem Prozessdaten gespeichert sind, indem er auf eine OPC-Server-Verweisdatenbank **210** zugreift, um eine Client-Anforderung einem bestimmten OPC-Server zuzuweisen. Die beispielhafte OPC-Server-Verweisdatenbank **210** kann durch EEPROM, RAM, ROM und/oder jeden beliebigen anderen Speichertyp implementiert sein.

[0069] In einigen Beispielen kann die Client-Anforderung eine Webadresse und/oder einen Identifikator des OPC-Servers enthalten. Der Adapter **118** kann auf die OPC-Server-Verweisdatenbank **210** zurückgreifen, um eine Position des angeforderten OPC-Servers zu ermitteln. In einigen Beispielen kann der Adapter **118** auf zwei oder mehr OPC-Server zugreifen, um auf die angeforderten Prozessdaten zuzugreifen. Nach dem Ermitteln der Position des OPC-Servers sendet der Adapter **118** eine Anforderung der Prozessdaten an den OPC-Server. Der Adapter **118** kann auf die Prozessdaten auf dem OPC-Server zugreifen, indem er eine Verzeichnis- und/oder eine Dateistruktur der OPC-Daten ermittelt, die auf dem Server gespeichert sind. Der Adapter **118** kann diese Verzeichnis- und/oder Dateistruktur an die webbasierte Schnittstelle **122** übermitteln, um es einem Client zu ermöglichen, ein oder mehrere Verzeichnisse und/oder Unterverzeichnisse über einen Webbrowser und/oder eine Programmanwendung auszuwählen. Durch Auswählen der Verzeichnisse und/oder Unterverzeichnisse legt der Client die Prozessdaten fest, die angeschaut werden sollen und/oder auf die zugegriffen werden soll. In anderen Beispielen kann der Client die Verzeichnis- und/oder Unterverzeichnisposition der gewünschten Prozessdaten enthalten. In einem weiteren Beispiel kann der Client Variablennamen, Prozessdatenidentifikatoren und/oder jede beliebige andere Datenidentifikationsinformation angeben, die der Adapter **118** dazu benutzen kann, durch einen OPC-Server zu navigieren und auf die angeforderten Prozessdaten zuzugreifen.

[0070] Der beispielhafte Adapter **118** kann mithilfe zugewiesener Endpunkte durch die Verzeichnisse, Unterverzeichnisse und/oder Dateien auf einem OPC-Server navigieren. Der Adapter **118** kann beispielsweise auf die OPC-Server-Verweisdatenbank **210** zurückgreifen, um einen Endpunkt zu identifizieren, der einem OPC-Server entspricht. Der OPC-Server kann dann auf Grundlage einer Hierarchie, Lesezugriff, Schreibzugriff und/oder Bezugszugriff Endpunkte an den Adapter **118** zurücksenden, die den Prozessdaten zugeordnet sind. Der Adapter **118** kann dann ermitteln, welcher Endpunkt oder welche Endpunkte den angeforderten Prozessdaten zugeordnet ist bzw. sind, und benutzt den oder die Endpunkte, um die Positionen auf dem OPC-Server zu durchzusuchen und/oder zu finden, an denen die Prozessdaten gespeichert sind.

[0071] In Beispielen, in denen der Adapter **118** eine Lese- und/oder Bezugsanweisung von der webbasierten Schnittstelle **122** empfängt, kann der Adapter **118** auf entsprechende Lese- und/oder Bezugsendpunkte zugreifen, die den angeforderten Prozessdaten zugeordnet sind. Ferner, in Beispielen, in denen ein Client ein regelmäßiges Intervall zum Empfangen von Prozessdaten angeben kann (z. B. zum Beziehen von Prozessda-

ten), kann der Adapter **118** den OPC-Server in dem regelmäßigen Intervall nach den gewünschten Prozessdaten abfragen.

[0072] In weiteren Beispielen, in denen ein Client einen Prozessdatenwert in ein Datenfeld schreiben kann, empfängt der Adapter **118** den Wert und/oder den zugeordneten Wertidentifikator von der webbasierten Schnittstelle **122**. Der Adapter **118** navigiert dann durch den OPC-Server (z. B. mithilfe von Schreibendpunkten), um die Datei- und/oder Verzeichnisposition zu finden, die dem geschriebenen Datenwert zugeordnet ist. Der Adapter **118** speichert sodann den geschriebenen Wert an der passenden Position auf dem OPC-Server. In einigen Beispielen, in denen ein Client einen Wert schreiben kann, kann der Adapter **118** einen Wert empfangen, nachdem der OPC-Wandler **120** den Wert aus einem Webseiten- und/oder einem Programmanwendungsformat in ein OPC-Format umgewandelt hat.

[0073] Der beispielhafte Adapter **118** aus [Fig. 2](#) weist Funktionen auf, die es dem Adapter **118** ermöglichen, zu unterschiedlichen OPC-Server eine Schnittstelle herzustellen und/oder mit ihnen zu kommunizieren, wobei diese OPC-Server mit unterschiedlichen Protokollen, Schnittstellen, Betriebssystemen und/oder Dateisystemen arbeiten können. Die OPC-Server-Verweisdatenbank **210** kann Verweise auf die Protokolle, Schnittstellen, Betriebssysteme und/oder Dateisysteme enthalten, die den einzelnen OPC-Servern zugeordnet sind. Wenn dann der Adapter **118** einen OPC-Server für den Zugriff identifiziert, kann der Adapter **118** die Protokoll-, Schnittstellen-, Betriebssystem- und/oder Dateisysteminformation, die dem OPC-Server zugeordnet ist, benutzen, um auf geeignete Weise mit dem OPC-Server zu kommunizieren und/oder eine Schnittstelle zu ihm herzustellen.

[0074] Nach dem Empfangen der und/oder Zugreifen auf die angeforderten Prozessdaten von einem OPC-Server leitet der beispielhafte Adapter **118** die Prozessdaten an den OPC-Wandler **120** weiter. Der beispielhafte OPC-Wandler **120** aus [Fig. 2](#) wandelt Prozessdaten aus einem beliebigen OPC- und/oder verpackungsbezogenen Format in ein Format um, das auf einem Webbrowser und/oder einer anderen Programmanwendung anschaubar ist. Nach dem Umwandeln der Prozessdaten in ein Webbrowserformat und/oder ein anderes Programmanwendungsformat leitet der OPC-Wandler **120** die umgewandelten Prozessdaten an die webbasierte Schnittstelle **122** weiter. Ferner kann der OPC-Wandler **120** geschriebene Werte und/oder Prozessdaten von der webbasierten Schnittstelle **122** empfangen. In diesen Beispielen wandelt der OPC-Wandler das Webbrowser- und/oder Programmanwendungsformat der geschriebenen Daten in ein OPC-Format um, das der Adapter **118** benutzen kann, um die geschriebenen Daten auf einem OPC-Server zu speichern. Der OPC-Wandler **120** kann beliebige Anwendungen, Frameworks, Datenumwandlungsalgorithmen usw. benutzen, die vom OPC und/oder einer beliebigen anderen Datenverpackungskonvention vorgegeben werden können.

[0075] Obwohl eine beispielhafte Art und Weise der Implementierung des Wrappers **110** in [Fig. 2](#) dargestellt ist, können auch eine oder mehrere der Schnittstellen, Datenstrukturen, Elemente, Prozesse und/oder Vorrichtungen aus [Fig. 2](#) kombiniert, geteilt, umgeordnet, fortgelassen, eliminiert und/oder in anderer Weise implementiert werden. Beispielsweise können die beispielhafte Sicherheitsverarbeitungseinrichtung **202**, die beispielhafte Sitzungssteuerung **204**, die beispielhafte webbasierte Schnittstelle **122**, die beispielhafte Datenverarbeitungseinrichtung **206**, die beispielhafte Datentypdatenbank **208**, der beispielhafte OPC-Wandler **120**, der beispielhafte Adapter **118** und/oder die beispielhafte OPC-Server-Verweisdatenbank **210** aus [Fig. 2](#) separat und/oder in jeder beliebigen Kombination implementiert sein, wobei beispielsweise maschinenzugängliche oder maschinenlesbare Anweisungen benutzt werden, die von einer oder mehreren Rechenvorrichtungen (z. B. der beispielhaften Prozessorplattform P10 aus [Fig. 11](#)) ausgeführt werden.

[0076] Außerdem können die beispielhafte Sicherheitsverarbeitungseinrichtung **202**, die beispielhafte Sitzungssteuerung **204**, die beispielhafte webbasierte Schnittstelle **122**, die beispielhafte Datenverarbeitungseinrichtung **206**, die beispielhafte Datentypdatenbank **208**, der beispielhafte OPC-Wandler **120**, der beispielhafte Adapter **118**, die beispielhafte OPC-Server-Verweisdatenbank **210** und/oder allgemeiner der Wrapper **110** durch Hardware, Software, Firmware und/oder jede beliebige Kombination aus Hardware, Software und/oder Firmware implementiert sein. So können beispielsweise die beispielhafte Sicherheitsverarbeitungseinrichtung **202**, die beispielhafte Sitzungssteuerung **204**, die beispielhafte webbasierte Schnittstelle **122**, die beispielhafte Datenverarbeitungseinrichtung **206**, die beispielhafte Datentypdatenbank **208**, der beispielhafte OPC-Wandler **120**, der beispielhafte Adapter **118**, die beispielhafte OPC-Server-Verweisdatenbank **210** und/oder allgemeiner der Wrapper **110** jeweils durch einen oder mehrere Schaltkreise, eine oder mehrere programmierbare Verarbeitungseinrichtungen, einen oder mehrere anwendungsspezifische integrierte Schaltkreise (ASICs, application specific integrated circuits), eine oder mehrere programmierbare logische Vorrichtungen (PLDs, programmable logic devices) und/oder feldprogrammierbare logische Vorrichtungen (FPLDs, field programmable logic devices) usw. implementiert sein. Ferner kann der beispielhafte Wrapper **110** aus [Fig. 2](#) ein oder mehrere

Elemente, Prozesse und/oder Vorrichtungen zusätzlich zu oder anstelle der in [Fig. 2](#) gezeigten aufweisen und/oder kann mehr als ein Element, einen Prozess bzw. eine Vorrichtung der dargestellten Elemente, Prozesse und Vorrichtungen aufweisen.

[0077] [Fig. 3](#) zeigt eine beispielhafte Schnittstelle **300** für Serveridentifikationsprozessdaten, die auf einem Webbrowser angezeigt werden. Die Schnittstelle **300** kann als Muster und/oder Vorlage bezeichnet werden. Die Schnittstelle **300** ist als ein Anwendungsfenster dargestellt. In anderen Beispielen kann die Schnittstelle **300** jedoch Navigationsfunktionen enthalten, die einem Webbrowser und/oder einem anderen Programm-Client zugeordnet sind. Während die beispielhafte Schnittstelle **300** ferner als ein Weg zum Anzeigen von Prozessdaten in Datenfeldern gezeigt ist, können auch andere beispielhafte Schnittstellen zum Anzeigen von Prozessdaten erzeugt werden.

[0078] Das Beispiel aus [Fig. 3](#) zeigt, dass die Schnittstelle **300** Serveridentifikationsinformationen anzeigt, die einer Servereigenschaften-Registerkarte **302** zugeordnet sind. Die Schnittstelle **300** weist einen Navigationsbereich **304** und einen Datenbereich **306** auf. Die Bereiche **304** und **306** können als allgemeine Vorlagen zum Anzeigen von Prozessdaten vorkonfiguriert sein. Der beispielhafte Wrapper **110** aus [Fig. 1](#) und/oder [Fig. 2](#) kann dann die Vorlage oder das Muster dazu benutzen, ein oder mehrere Datenfelder und die zugeordneten Prozessdaten anzugeben und/oder einzubetten. Beispielsweise kann der Datenbereich **306** dazu vorkonfiguriert sein, einen Variablen- und/oder einen Parameternamen (z. B. Attribut) und einen Datenwert (z. B. Wert) anzuzeigen, die den Prozessdaten zugeordnet sind. Dann kann der Wrapper **110** die Datenfelder, die der Serviceidentifikation zugeordnet sind, in die Attributspalte und/oder die Wertspalte einbetten und/oder einsetzen. Beispielsweise können Prozessdaten, die den OPC-Servernamen beschreiben, als ein Attribut „Servername“ mit einem Wert „OPC DA Server“ identifiziert sein.

[0079] Ein Client kann auf die Prozessdaten, die auf der Schnittstelle **300** angezeigt sind, zugreifen, indem er eine Webadresse und/oder eine Internetprotokoll-(IP)-Adresse in einen Webbrowser eingibt. Der Webbrowser kann dann zum Wrapper **110** und/oder zum Webseitenserver **108** aus [Fig. 1](#) navigieren. Der Wrapper **110** löst die Webadresse und/oder die IP-Adresse zum OPC-Server auf, der in der Schnittstelle **300** angezeigt ist. Der Wrapper **110** kann dann die Prozessdaten abrufen, die dem OPC-Server zugeordnet sind, die Prozessdaten in ein Webbrowserformat umwandeln, den oder die Datentypen der Prozessdaten ermitteln, eine Vorlage auswählen, die der Schnittstelle **300** zugeordnet ist, die Prozessdaten in die Schnittstelle **300** einbetten und die Schnittstelle **300** über den Webbrowser für den Client anzeigen. In diesem Beispiel ermittelt der Wrapper **110** den oder die Prozessdatentypen durch Identifizieren der Attributmetadaten, die den Datenwerten zugeordnet sind. Der Wrapper **110** kann dann eine Liste der Datenwerte erzeugen, die dem Attributnamen entsprechen.

[0080] Der beispielhafte Navigationsbereich **304** zeigt eine Verzeichnis- und/oder eine Dateistruktur der OPC-Server-Identifikationsinformation, die dem OPC-Server zugeordnet ist.

[0081] Ein Client kann durch die Verzeichnisstruktur im Navigationsbereich **304** navigieren, um Prozessdaten auszuwählen, die der OPC-Serverinformation zugeordnet sind, die im Datenbereich **306** angezeigt wird. Der Client kann selektiv andere OPC-Serverdaten anschauen, indem er andere Verzeichnisse im Navigationsbereich **304** auswählt. Der beispielhafte Wrapper **110** kann diese Verzeichnisstruktur (z. B. Prozessdaten) vom OPC-Server abrufen und die Verzeichnisstruktur mithilfe der Vorlage des Navigationsbereichs **304** mit Datenfeldern konfigurieren.

[0082] [Fig. 4](#) zeigt eine beispielhafte Schnittstelle **400** (z. B. ein Muster) für ein Serverhierarchieverzeichnis, das in einem Webbrowser angezeigt wird. Das Hierarchieverzeichnis wird angezeigt, indem die Objektbaum-Registerkarte **402** ausgewählt wird, und kann eine Datenstruktur von Prozessdaten darstellen, die auf einem OPC-Server gespeichert ist. Der beispielhafte Wrapper **110** aus [Fig. 1](#) und/oder [Fig. 2](#) ruft die Hierarchieverzeichnisstruktur ab, die auf dem OPC-Server gespeichert ist, wandelt die Daten in ein Webbrowserformat um und ermittelt den Datentyp oder die Datentypen, der oder die einer Datenstruktur zugeordnet ist bzw. sind. Der Wrapper **110** konfiguriert und organisiert dann das Hierarchieverzeichnis und setzt es in Datenfelder in einem Navigationsbereich **404** ein. Ein Client kann durch die Verzeichnisse im Navigationsbereich **404** navigieren, um ein Verzeichnis und/oder ein Unterverzeichnis auszuwählen, das die gewünschten Prozessdaten enthält. Das Hierarchieverzeichnis kann derart organisiert sein, dass es eine Struktur von Vorrichtungen in einem Steuerungssystem und/oder eine Struktur eines Steuerungsdienstprogramms reflektiert, das von einer Steuerung ausgeführt wird. Der beispielhafte Wrapper **110** kann Endpunktadressen benutzen, um durch die Verzeichnisstruktur zu navigieren und das Hierarchieverzeichnis in einem Webbrowser an den Client bereitzustellen.

[0083] Die beispielhafte Schnittstelle **400** weist ferner einen Filterbereich **406** auf, um Verzeichnisse anhand von Eingaben zu filtern, die von einem Client bereitgestellt werden. Die Schnittstelle **400** weist außerdem einen Datenbereich **408** auf, der Datenfelder mit Variabel- und/oder Parameternamen (z. B. Attribut) und einen entsprechenden Prozessdatenwert (z. B. Wert) für ein ausgewähltes Verzeichnis enthält. In dem Beispiel aus [Fig. 4](#) wählt der Client das Verzeichnis PID1. Entsprechend ruft der Wrapper **110** die Prozessdaten ab, die dem Verzeichnis PID1 zugeordnet sind, undbettet die Prozessdaten ein, damit sie durch den Webbrower im Datenbereich **408** angezeigt werden.

[0084] [Fig. 5](#) zeigt eine beispielhafte Schnittstelle **500** für Prozessdaten, die in einem Webbrower angezeigt werden. Die Prozessdaten werden durch Auswählen einer Datenlisten-Registerkarte **502** im Webbrower angezeigt. Wenn ein Client die Datenlisten-Registerkarte **502** auswählt, ruft der beispielhafte Wrapper **110** aus [Fig. 1](#) und/oder [Fig. 2](#) Prozessdaten ab. Die Schnittstelle **500** weist einen Bezugsbereich **504** und einen Datenbereich **506** auf. Die Gestaltung der Bereiche **504** und/oder **506** kann anhand des Prozessdatentyps vorkonfiguriert sein, der unter der Datenlisten-Registerkarte **502** abgerufen werden kann.

[0085] Der Bezugsbereich **504** erlaubt es einem Client, eine Aktualisierungsrate der Prozessdaten festzulegen, die im Datenbereich **506** angezeigt werden. In diesem Beispiel legt der Client eine Aktualisierungsrate von 5000 Millisekunden (msec) fest. Der Bezugsbereich **504** enthält außerdem einen Client-Identifikator (z. B. Client-ID), einen OPC-Serveridentifikator (Server-ID) und die ausgewählte Aktualisierungsrate (z. B. Aktualisierungsrate).

[0086] Der beispielhafte Datenbereich **506** zeigt ausgewählte Prozessdaten einschließlich einer Verzeichnisposition der Prozessdaten auf dem OPC-Server (z. B. Instanz-ID), einen Datenwert (z. B. Datenwert), einen Status der Prozessdaten (z. B. Statuscode) und einen Zeitpunkt, an dem die Prozessdaten erzeugt wurden (z. B. Zeitstempel). Der Datenbereich **506** enthält Schaltflächen, mit denen Prozessdaten zur Anzeige hinzugefügt oder daraus entfernt werden können. Der Datenbereich **506** enthält außerdem Schaltflächen zum Lesen oder Schreiben der Prozessdaten.

[0087] In dem Beispiel aus [Fig. 5](#) ermittelt der Wrapper **110**, dass die anzuzeigenden Prozessdaten einem Datentyp zugeordnet sind, der in der Form von Werten in einer Tabelle angezeigt werden kann. In anderen Beispielen können die Prozessdaten in einem Graphen, einem Diagramm, einer Animation, einem Grafikindikator usw. angezeigt werden. Beispielsweise kann der Datenbereich **506** in der Schnittstelle **500** derart vorkonfiguriert sein, dass für den Fall, dass einige Prozessdaten einer Grafikanzeige zugeordnet sind, der Wrapper **110** eine vorkonfigurierte Grafikanzeige aufrufen kann, um die Prozessdaten in den passenden Datenfeldern anzuzeigen.

[0088] [Fig. 6](#) zeigt eine beispielhafte Schnittstelle **600**, die mit dem beispielhaften Wrapper **110** kommuniziert, um Prozessdaten in einem Webbrower anzuzeigen. Die beispielhafte Schnittstelle **600** enthält eine Navigationsleiste **602**, die ein Betreiber dazu benutzen kann, einen Feldgerätnamen einzugeben, um Prozessdaten anzuschauen, die dem Feldgerät zugeordnet sind. In einem anderen Beispiel kann ein Benutzer ein Hierarchieverzeichnis ähnlich dem Verzeichnis benutzen, das in Zusammenhang mit [Fig. 4](#) beschrieben wurde, um ein Feldgerät zu suchen. Die Schnittstelle **600** enthält auch eine Menüleiste **604**, die Optionen zur Schnittstellenherstellung mit dem Feldgerät TT101 enthält. In diesem Beispiel wird eine Option „Manuelle Einrichtung“ ausgewählt.

[0089] Die beispielhafte Schnittstelle **600** aus [Fig. 6](#) kann Zielanzeigeeinformationen an den Wrapper **110** bereitstellen, die angeben, dass Prozessdaten, die dem Feldgerät TT101 zugeordnet sind, in einem Webbrower oder an einem Arbeitsplatzrechner oder einem Laptop angezeigt werden sollen. Wenn ein Feldgerät (z. B. ein Objekt) ausgewählt oder eingegeben wird, empfängt der beispielhafte Wrapper **110** außerdem den Feldgerätnamen (z. B. TT101). Der Wrapper **110** benutzt den Feldgerätidentifikator, um einen Typ von Feldgerät (z. B. einen Objekttyp), dem Feldgerät zugeordnete Eigenschaften und/oder dem Feldgerät zugeordnete Prozessdaten (z. B. Werte) zu identifizieren. Der Wrapper **110** benutzt diese Information zusammen mit der Zielanzeigeeinformation, um eine Vorlage zum Anzeigen der Prozessdaten in Datenfeldern auszuwählen. Der Wrapper **110** benutzt diese Information außerdem, um auf die Prozessdaten von einem OPC-Server zuzugreifen. Der Wrapper **110** kann eine Vorlage auswählen, indem er auf die Datentypdatenbank **208** zugreift und Vorlagen für Webbrower sucht, die auf einem Arbeitsplatzrechner ausgeführt werden. Der Wrapper **110** kann dann die Vorlagen anhand eines Vorrichtungstyps dem Feldgerät TT101 eingrenzen. Der Wrapper **110** kann anschließend das oder die Datenfelder in der Vorlage auswählen, das oder die den Prozessdaten entspricht bzw. entsprechen.

[0090] In dem Beispiel aus [Fig. 6](#) kann der Wrapper **110** eine Vorlage **606** und eine Vorlage **608** an die Schnittstelle **600** bereitstellen. Die Vorlagen **606** und **608** enthalten Datenfelder mit Prozessdaten (z. B. Analogeingang (AE), %-Bereich, Einheiten, Unterer Bereich, Oberer Bereich, Registerkarte, Datum und Deskriptor), die dem Feldgerät TT101 zugeordnet sind. Der beispielhafte Wrapper **110** kann die Prozessdaten, die in den Vorlagen **606** und **608** angezeigt sind, ausführen, indem er die Prozessdaten anhand von Metadaten in den Prozessdaten dem oder den passenden Datenfeldern zuordnet und darin einfügt. In anderen Beispielen können Vorlagen Graphen, Diagramme, Grafiken und/oder eine beliebige andere Datendarstellung enthalten.

[0091] [Fig. 7A](#) und [Fig. 7B](#) zeigen eine beispielhafte Schnittstelle **700**, die auf einer drahtlosen Vorrichtung **702** angezeigt wird, die mit den beispielhaften Wrappern **110** aus [Fig. 1](#) und/oder [Fig. 2](#) kommuniziert, um Prozessdaten auf einer Client-Anwendung anzuzeigen. Bei der drahtlosen Vorrichtung **702** kann es sich um eine beliebige tragbare Rechenvorrichtung handeln, darunter beispielsweise ein Smartphone, einen persönlichen digitalen Assistenten (PDA), ein Webtelefon usw. Die Beispiele aus [Fig. 7A](#) und [Fig. 7B](#) zeigen, dass der beispielhafte Wrapper **110** Prozessdaten in einer Vorlage ausführt, die für die drahtlose Vorrichtung **702** konfiguriert wurde.

[0092] In [Fig. 7A](#) enthält die Benutzerschnittstelle **700** ein Objektsuchfeld **704** und eine Tastatur **706**. Die Objektsuche **704** erlaubt es einem Benutzer, über die Tastatur **706** einen Feldgerätidentifikator (z. B. OPC DA/DEVICE/TT101) einzugeben. Das Objektsuchfeld **704** stellt das eingegebene Feldgerät TT101 an den Wrapper **110** bereit. Außerdem kann die drahtlose Vorrichtung **702** die Zielanzeigeeinformation an den Wrapper **100** senden. Der Wrapper **110** benutzt den TT101-Feldgerätidentifikator, um Prozessdaten und/oder Eigenschaften zu suchen, die dem Feldgerät TT101 zugeordnet sind. Außerdem benutzt der beispielhafte Wrapper **110** die Zielanzeigeeinformation, um eine Vorlage **708** auszuwählen, die zur Anzeige auf dem relativ kleinen Bildschirm der drahtlosen Vorrichtung **702** formatiert ist.

[0093] Der Wrapper kann die Prozessdaten den entsprechenden Datenfeldern der Vorlage **708** zuordnen, sie darin einfügen und ausführen, wie in [Fig. 7B](#) gezeigt. Außerdem kann der Wrapper **110** eine Menüleiste **710** zum Navigieren zu unterschiedlichen Vorlagen anzeigen. Die Menüleiste **710** kann in der Vorlage **708** enthalten sein und/oder kann durch Navigationsprozessdaten festgelegt werden, die der Verzeichnisposition OPC DA/DEVICE/TT101 zugeordnet sind. Alternativ kann die Anzeige der Menüleiste **710** von der Client-Anwendung verwaltet werden. Das Beispiel aus [Fig. 7B](#) zeigt, dass der Wrapper **110** die Vorlage **708** auswählt, die für die drahtlose Vorrichtung **702** formatiert ist, während der Wrapper **110** in [Fig. 6](#) die Vorlagen **606** und **608** auswählt, die für den Webbrowser an einem Arbeitsplatzrechner formatiert sind.

[0094] [Fig. 8](#) zeigt eine beispielhafte Schnittstelle **800**, die Prozessdaten in einem Webbrowser anzeigt. Die Schnittstelle **802** enthält eine Navigationsleiste **802**, die ein Benutzer dazu benutzen kann, eine Dateiposition oder einen Pfad der Prozessdaten einzugeben. Eine Vorlage **804** zeigt eine Tabelle von Prozessdaten, die in Datenfeldern angezeigt werden. Die Vorlage **804** enthält Datenfelder für Hersteller, Vorrichtungstyp und Version. Ein Benutzer kann den Pfad <HTTP://OPCDASERVER/DEVICE/DEVICECOLLECTION> angeben, um auf Prozessdaten zuzugreifen, die Vorrichtungen zugeordnet sind, die in einer Gruppe geordnet sind, oder um diese zu verfolgen. Auf die Prozessdaten in der Vorlage **804** kann als eine XML-Datei von einem OPC-Server zugegriffen werden:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="file:///Ccd.xsl"?>
<catalog>
  <device>
```

```

<manufacturer>Rosemount</manufacturer>
<type>3051MV</type>
<version>5</version>
</device>
<device>
    <manufacturer>Micromotion</manufacturer>
    <type>MM MV</type>
    <version>2</version>
</device>
<device>
    <manufacturer>Fisher</manufacturer>
    <type>DVC6010P</type>
    <version>3</version>
</device>
<catalog>

```

[0095] Die oben gezeigte beispielhafte XML-Datei führt den Hersteller, den Vorrichtungstyp und die Version der drei Vorrichtungen unter der Zeile `<catalog>` auf. Um diese Information in der Schnittstelle **800** anzuzeigen, kann der beispielhafte Wrapper **110** die XML-Datei in eine XSLT-Datei umwandeln, die die Vorlage **804** zum Anzeigen der Prozessdaten festlegt. Die umgewandelte XSLT-Datei kann Folgendes enthalten:

```

<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- Edited by XMLSpy® -->
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:template match="/">
        <html>
            <body>
                <h2>My Device Collection</h2>
                <table border="1">
                    <tr bgcolor="#9acd32">
                        <th>Manufacturer</th>
                        <th>Device Type</th>
                        <th>Version</th>
                    </tr>
                    <xsl:for-each select="catalog/device">
                        <tr>
                            <td>
                                <xsl:value-of select="manufacturer"/>
                            </td>

```

```

        <td>
        <xsl:value-of select="type"/>
        </td>
        <td>
        <xsl:value-of select="version"/>
        </td>
    </tr>
</xsl:for-each>
</table>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

[0096] Die Zeile `<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">` legt fest, dass die Vorlage **804** in der Schnittstelle **800** angezeigt werden soll. Der beispielhafte Wrapper **110** kann diese Vorlage anhand von Information (z. B. Zielanzeigeinformation, Objekttyp usw.) ermitteln, die er von der Schnittstelle **800** empfängt. Die XSLT-Datei legt auch die Spaltentitel (z. B. Hersteller, Vorrichtungstyp und Version) für die Vorlage **804** und Anweisungen (z. B. `<xsl:value-of select="manufacturer"/>`, `<xsl:value-of select="type"/>` und `<xsl:value-of select="version"/>`) zum Formatieren der Prozessdaten in die Datenfelder fest.

[0097] [Fig. 9A](#), [Fig. 9B](#), [Fig. 9C](#) und [Fig. 10](#) sind Ablaufdiagramme eines Beispielverfahrens, das ausgeführt werden kann, um die beispielhafte Sicherheitsverarbeitungseinrichtung **202**, die beispielhafte Sitzungssteuerung **204**, die beispielhafte webbasierte Schnittstelle **122**, die beispielhafte Datenverarbeitungseinrichtung **206**, die beispielhafte Datentypdatenbank **208**, den beispielhaften OPC-Wandler **120**, den beispielhaften Adapter **118**, die beispielhafte OPC-Server-Verweisdatenbank **210** und/oder allgemeiner den Wrapper **110** aus [Fig. 1](#) und/oder [Fig. 2](#) zu implementieren. Das Beispielverfahren aus [Fig. 9A](#), [Fig. 9B](#), [Fig. 9C](#) und/oder [Fig. 10](#) kann von einem Prozessor, einer Steuerung und/oder jeder anderen geeigneten Verarbeitungsvorrichtung ausgeführt werden. Beispielsweise kann das Beispielverfahren aus [Fig. 9A](#), [Fig. 9B](#), [Fig. 9C](#) und/oder [Fig. 10](#) in codierten Anweisungen verkörpert sein, die auf einem beliebigen fassbaren computerlesbaren Medium wie z. B. einem Flash-Speicher, einer CD, einer DVD, einer Diskette, einem ROM, einem RAM, einem programmierbaren ROM (PROM), einem elektronisch programmierbaren ROM (EPROM), einem elektronisch löschenbaren PROM (EEPROM), einer optischen Speicherdisc, einer optischen Speichervorrichtung, einer Magnetspeicherdisc, einer Magnetspeichervorrichtung und/oder jedem beliebigen anderen Medium gespeichert sein, das dazu benutzt werden kann, Programmcode und/oder Anweisungen in Form von Verfahren und Datenstrukturen zu tragen oder zu speichern, und auf das ein Prozessor, ein universeller oder spezieller Computer oder eine andere Maschine mit einem Prozessor (z. B. die beispielhaften Verarbeitungsplattform P10, die unten im Zusammenhang mit [Fig. 11](#) erörtert wird), zugreifen kann. Kombinationen der Vorstehenden fallen ebenfalls in den Umfang computerlesbarer Medien.

[0098] Die Verfahren umfassen beispielsweise Anweisungen und/oder Daten, die einen Prozessor, einen universellen oder speziellen Computer oder eine spezielle Verarbeitungsmaschine dazu veranlassen, ein oder mehrere bestimmte Verfahren zu implementieren. Alternativ können einige oder alle Beispielverfahren aus [Fig. 9A](#), [Fig. 9B](#), [Fig. 9C](#) und/oder [Fig. 10](#) unter Verwendung einer oder mehrerer beliebiger Kombinationen aus ASIC(s), PLD(s), PFLD(s), diskreter Logik, Hardware, Firmware usw. implementiert werden.

[0099] Auch können einige oder alle Beispielverfahren aus [Fig. 9A](#), [Fig. 9B](#), [Fig. 9C](#) und/oder [Fig. 10](#) statt dessen mithilfe manueller Vorgänge oder als eine Kombination beliebiger der vorstehenden Verfahren implementiert werden, beispielsweise einer beliebigen Kombination von Firmware, Software, diskreter Logik und/oder Hardware. Ferner können zahlreiche andere Verfahren zum Implementieren der Beispielvorgänge aus [Fig. 9A](#), [Fig. 9B](#), [Fig. 9C](#) und/oder [Fig. 10](#) verwendet werden. Beispielsweise kann die Reihenfolge der Ausführung der Blöcke geändert werden, und/oder einer oder mehrere der beschriebenen Blöcke können geändert, eliminiert, unterteilt oder kombiniert werden. Zusätzlich können einige oder alle Beispielverfahren aus [Fig. 9A](#), [Fig. 9B](#), [Fig. 9C](#) und/oder [Fig. 10](#) sequenziell und/oder parallel z. B. durch separate Verarbeitungsstränge, Prozessoren, Vorrichtungen, diskrete Logik, Schaltkreise usw. ausgeführt werden.

[0100] Die Beispielverfahren **900** aus [Fig. 9A](#), [Fig. 9B](#) und [Fig. 9C](#) erlauben es einem Client, auf Prozessdaten zuzugreifen, die auf einem OPC-Server gespeichert sind. Eine Vielzahl von Beispielverfahren **900** kann parallel oder in Serie ausgeführt werden, um für eine Vielzahl von Clients auf Prozessdaten zuzugreifen. Außerdem kann in Beispielen, in denen die angeforderten Prozessdaten auf zwei oder mehr OPC-Servern gespeichert sein können, ein Beispielverfahren **900** für jeden OPC-Server implementiert werden, oder es kann alternativ ein einzelnes Beispielverfahren **900** für die OPC-Server implementiert werden.

[0101] Das Beispielverfahren **900** aus [Fig. 9A](#) beginnt, indem eine Anforderung zum Zugreifen auf Prozessdaten empfangen wird (Block **902**). Die Anforderung kann von der beispielhaften Sicherheitsverarbeitungseinrichtung **202** und/oder der webbasierten Schnittstelle **122** aus [Fig. 2](#) stammen. Das Beispielverfahren **900** fordert dann Sicherheitszugangsdaten an (z. B. die Sicherheitsverarbeitungseinrichtung), um auf die Prozessdaten zuzugreifen (Block **904**). Zu Sicherheitszugangsdaten können ein Benutzername und ein Passwort, ein Zugriffscode, ein eindeutiger Identifikator usw. gehören. In einigen Beispielen können die Sicherheitszugangsdaten in der Anforderung enthalten sein. Als nächstes bestimmt das Beispielverfahren **900**, ob die empfangenen Sicherheitszugangsdaten die Authentifizierung bestehen (Block **906**). Wenn die Sicherheitszugangsdaten in Ordnung sind und die Authentifizierung bestehen, erzeugt das Beispielverfahren **900** eine Zugriffssitzung mit Lese-, Schreib- und/oder Bezugszugriff, abhängig von der Anforderung des Clients (Block **908**). Das Beispielverfahren **900** identifiziert dann einen OPC-Server, der der Anforderung zugeordnet ist (z. B. greift der Adapter **118** auf die OPC-Server-Verweisdatenbank **210** aus [Fig. 2](#) zu) (Block **910**).

[0102] Wenn das Beispielverfahren **900** allerdings nicht dazu in der Lage ist, die Sicherheitszugangsdaten zu bestätigen (Block **906**), kann das Beispielverfahren **900** eine Zugriffssitzung mit ausschließlichem Lesezugriff erzeugen (Block **912**). Das Beispielverfahren **900** identifiziert dann einen OPC-Server, der der Anforderung zugeordnet ist (Block **910**). Wenn das Beispielverfahren **900** allerdings nicht dazu in der Lage ist, die Sicherheitszugangsdaten zu bestätigen (Block **906**), kann das Beispielverfahren **900** alternativ den Zugriffssitzung auf die angeforderten Prozessdaten verweigern.

[0103] Das Beispielverfahren **900** aus [Fig. 9A](#) fährt fort, indem es (z. B. über den Adapter **118**) auf den identifizierten OPC-Server (Block **914**) zugreift und auf ein oder mehrere Verzeichnisse zugreift, das oder die der Anforderung zugeordnet sind (Block **916**). In Beispielen, in denen die angeforderten OPC-Daten sich auf mehr als einem OPC-Server befinden, kann das Beispielverfahren **900** außerdem die Verzeichnisse und/oder Dateien suchen, die sich auf dem oder den OPC-Servern befinden (Block **916**). Das Beispielverfahren **900** kann die angeforderten Prozessdaten mithilfe von Endpunkten, über manuelle Client-Navigation und/oder anhand von Information suchen, die in der Anforderung vom Client bereitgestellt wird. Nach dem Zugreifen auf den Speicherort der angeforderten Prozessdaten empfängt das Beispielverfahren **900** eine Anforderung zum Lesen, Schreiben und/oder Beziehen der angeforderten Prozessdaten. In einigen Beispielen kann die Anforderung zum Zugriff auf Prozessdaten eine Anforderung zum Lesen, Schreiben und/oder Beziehen der Prozessdaten beinhalten. In anderen Beispielen kann der Client eine weitere Anforderung senden, um über eine Lese-, Schreib- und/oder Bezugsfunktion auf die Prozessdaten zuzugreifen.

[0104] Das Beispielverfahren **900** aus [Fig. 9B](#) bestimmt, ob die Anforderung einer Lese-, Schreib- und/oder Bezugsfunktion zugeordnet ist (Block **918**). Besteht die Anforderung darin, Prozessdaten zu lesen (Block **918**), fährt das Beispielverfahren **900** fort, indem es die angeforderten Prozessdaten (z. B. über den Adapter **118**) von dem oder den zugeordneten OPC-Servern abruft (Block **920**). Das Beispielverfahren **900** wandelt die Prozessdaten dann aus einem OPC-Format in ein Webbrowser- und/oder Programmanwendungsformat um (z. B. über den OPC-Wandler **120**) (Block **922**).

[0105] Besteht die Anforderung jedoch darin, Prozessdaten zu beziehen (Block **918**), fährt das Beispielverfahren **900** fort, indem es die angeforderten Prozessdaten auf dem OPC-Server bezieht (Block **924**). Das Beispielverfahren **900** kann die Prozessdaten beziehen, indem es ein Zeitintervall festlegt, nach dem eine Anforderungsnachricht an einen OPC-Server gesendet wird, um die neuesten Prozessdaten anzufordern. Nach dem Beziehen der Prozessdaten ruft das Beispielverfahren **900** die angeforderten Prozessdaten vom OPC-Server ab (Block **920**). Ferner kann das Beispielverfahren **900** fortfahren, die angeforderten Prozessdaten in den festgelegten Zeitintervallen vom OPC-Server abzurufen. Bei jedem Abrufen der Prozessdaten wandelt das Beispielverfahren **900** die Prozessdaten dann aus einem OPC-Format in ein Webbrowser- und/oder Programmanwendungsformat um (z. B. über den OPC-Wandler **120**) (Block **922**).

[0106] Wenn die Anforderung jedoch darin besteht, Prozessdaten zu schreiben (Block **918**), fährt das Beispielverfahren **900** fort, indem es den geschriebenen Datenwert empfängt, der vom Client bereitgestellt wird, und den Datenwert in einen OPC-Formatwert umwandelt (z. B. über den OPC-Wandler **120**) (Block **926**). Das

Beispielverfahren **900** schreibt dann den Prozessdatenwert an die passende Datenposition im OPC-Server (z. B. über den Adapter **118**) (Block **928**). Das Beispielverfahren **900** kann dann die geschriebenen Prozessdaten auf dem OPC-Server lesen und die Prozessdaten in ein Webbrowserformat umwandeln (Block **922**). Das Beispielverfahren **900** kann einen geschriebenen Wert zurücklesen, um dem Client zu demonstrieren, dass ein geschriebener Wert korrekt in den OPC-Server geschrieben wurde.

[0107] Nach dem Umwandeln der Prozessdaten in ein OPC-Format fährt das Beispielverfahren **900** fort, indem es den oder die Datentypen, die den Prozessdaten zugeordnet sind, ermittelt (z. B. über die Datenverarbeitungseinrichtung **206**) (Block **930**). Das Beispielverfahren **900** aus [Fig. 9C](#) ermittelt dann, ob die Anforderung vom Client einer Webserveranwendung oder einer Client-Anzeigeanwendung (z. B. einer Programm-anwendung) zugeordnet ist (Block **932**). Wenn die Anforderung vom Client einer Webserveranwendung zugeordnet ist, erzeugt und/oder konfiguriert das Beispielverfahren **900** Datenfelder für jeden Datentyp (z. B. über die webbasierte Schnittstelle **122**) (Block **934**). Das Beispielverfahren **900**bettet dann die Prozessdaten in das oder die entsprechenden Datenfelder ein oder setzt sie darin ein und stellt die Prozessdaten in dem oder den Datenfeldern zusammen (Block **936**). Als nächstes ermöglicht das Beispielverfahren **900** die Anzeige der Datenfelder einschließlich der Prozessdaten auf einer Webseite, auf die aber einen Webbrowser zugegriffen wird, der vom Client betrieben wird (Block **938**).

[0108] Falls allerdings die Anforderung vom Client einer Client-Anzeigeanwendung zugeordnet ist (Block **932**), ruft das Beispielverfahren **900** aus [Fig. 9C](#) die Client-Anzeigeanwendung am Client-Standort auf (Block **944**). Das Aufrufen der Client-Anzeigeanwendung kann das Öffnen der Anwendung in einem Webbrowser beinhalten, der vom Client benutzt wird, um auf die Prozessdaten zuzugreifen (z. B. eine ActiveX- oder Silverlight™-Anwendung). Das Aufrufen der Client-Anzeigeanwendung kann auch das Starten einer Programmanwendung beinhalten, um formatierte Prozessdaten und/oder Datenfelder zu empfangen. Das Beispielverfahren **900** kann dann eine Anforderung von der Client-Anzeigeanwendung empfangen, die angibt, dass die Anwendung aufgerufen wurde und zum Empfang von Prozessdaten bereit ist (Block **946**). Das Beispielverfahren **900**bettet dann die Prozessdaten in das oder die entsprechenden Datenfelder ein und/oder setzt sie darin ein und/oder ordnet sie ihnen zu (Block **948**) und überträgt die Prozessdaten und/oder das oder die Datenfelder an die Client-Anzeigeanwendung (Block **950**). Das Beispielverfahren **900** kann auch anhand von Typen der Prozessdaten und/oder eines Anzeigetyps der Prozessdaten eine Vorlage für das oder die Datenfelder auswählen. Nach dem Empfang der Prozessdaten sorgt das Beispielverfahren **900** dafür, dass die Client-Anzeigeanwendung die Datenfelder so konfiguriert, dass die entsprechenden Prozessdaten in einer Schnittstelle und/oder Anzeige der Client-Anzeigeanwendung gezeigt werden (Block **952**).

[0109] Das Beispielverfahren **900** aus [Fig. 9C](#) fährt fort, indem es ermittelt, ob der Client einen Teil der Prozessdaten und/oder des oder der Datenfelder angepasst hat (Block **940**). Wenn das Beispielverfahren **900** ermittelt, dass der Client die Prozessdaten und/oder das oder die Datenfelder nicht angepasst hat, kehrt das Beispielverfahren **900** zum Empfang einer Anforderung zum Zugreifen auf Prozessdaten von demselben und/oder einem anderen Client zurück (Block **902**). Wenn das Beispielverfahren **900** dagegen ermittelt, dass der Client die Prozessdaten und/oder das oder die Datenfelder angepasst hat, speichert das Beispielverfahren **900** die angepassten Daten (Block **942**). Das Beispielverfahren **900** speichert die Anpassungsinformation, derart, dass die Prozessdaten im angepassten Format angezeigt werden können, wenn derselbe Client zu einem anderen Zeitpunkt auf dieselben Prozessdaten zugreift. Das Beispielverfahren **900** kehrt dann zum Empfang einer Anforderung zum Zugreifen auf Prozessdaten von demselben und/oder einem anderen Client zurück (Block **902**).

[0110] Das Beispielverfahren **1000** aus [Fig. 10](#) ermittelt eine Vorlage und/oder Datenfelder zum Anzeigen von Prozessdaten, die von einem OPC-Server empfangen wurden. Eine Vielzahl von Beispielverfahren **1000** kann parallel oder in Serie ausgeführt werden, um eine Vielzahl von Vorlagen auszuführen. Außerdem kann in Beispielen, in denen die angeforderten Prozessdaten auf zwei oder mehr OPC-Servern gespeichert sein können, ein Beispielverfahren **1000** für jeden OPC-Server implementiert werden, oder es kann alternativ ein einzelnes Beispielverfahren **1000** für die OPC-Server implementiert werden.

[0111] Das Beispielverfahren **1000** aus [Fig. 10](#) beginnt, indem es im Wrapper **110** eine Anforderung zum Anschauen eines Objekts (z. B. Prozessdaten) empfängt (Block **1002**). Bei dem Objekt kann es sich um ein Feldgerät und/oder um Prozessdaten handeln, die einem Feldgerät zugeordnet sind. Das Beispielverfahren **1000** kann (z. B. über die webbasierte Schnittstelle **122**) Zielanzeigearinformation identifizieren, die der Anforderung zugeordnet ist (Block **1004**). Die Zielanzeigearinformation kann in der Anforderung enthalten sein und/oder kann separat von einem Webbrowser oder einer Client-Anwendung übertragen werden. Als nächstes ermittelt das Beispielverfahren **1000** (z. B. über die webbasierte Schnittstelle **122**, den Adapter **118** und/oder

die Datenverarbeitungseinrichtung **206**) einen Typ des angeforderten Objekts (Block **1006**), Eigenschaften, die dem Objekt zugeordnet sind (Block **1008**) und/oder greift auf Prozessdaten (z. B. dem Objekt zugeordnete Werte) zu (Block **1010**). Das Zugreifen auf die Prozessdaten beinhaltet das Zugreifen auf einen OPC-Server, wie im Zusammenhang mit [Fig. 9A–Fig. 9C](#) beschrieben.

[0112] Das Beispielverfahren **1000** fährt fort, indem es (z. B. über die Datenverarbeitungseinrichtung **206**) eine Vorlage zum Anzeigen des Objekts aufgrund des Objekttyps, der Eigenschaften, der Prozessdaten und/oder der Zielanzeigeinformation ermittelt (Block **1012**). Das Beispielverfahren **1000** kann (z. B. über die webbasierte Schnittstelle **122**) das Objekt (z. B. die Prozessdaten) mit entsprechenden Datenfeldern in der Vorlage verknüpfen und/oder sie darin einfügen. Als nächstes führt das Beispielverfahren **1000** das Objekt und/oder Prozessdaten, die dem Objekt zugeordnet sind, in der ausgewählten Vorlage aus (z. B. über die webbasierte Schnittstelle **122**) (Block **1014**). Das Beispielverfahren **1000** zeigt das Objekt und/oder Prozessdaten, die dem Objekt zugeordnet sind, in einer Vorlage an, die in einer Schnittstelle angezeigt wird (z. B. über die webbasierte Schnittstelle **122**) (Block **1016**). Das Beispielverfahren **1000** kann (z. B. über die webbasierte Schnittstelle **122**) ermitteln, ob weitere Objekte vorliegen, die angezeigt werden sollen (Block **1018**). Wenn das Beispielverfahren **1000** ermittelt, dass weitere Objekte vorliegen, empfängt das Beispielverfahren **1000** die weiteren Objekte (Block **1002**). Wenn das Beispielverfahren **1000** ermittelt, dass keine weiteren Objekte vorliegen, endet das Beispielverfahren **1000**.

[0113] [Fig. 11](#) ist ein Blockdiagramm eines beispielhaften Prozessorsystems P10, das dazu benutzt werden kann, die hier beschriebenen beispielhaften Verfahren und Vorrichtungen zu implementieren. Beispielsweise können Prozessorsysteme, die dem beispielhaften Prozessorsystem P10 ähnlich sind oder mit ihm identisch sind, benutzt werden, um die beispielhafte Sicherheitsverarbeitungseinrichtung **202**, die beispielhafte Sitzungssteuerung **204**, die beispielhafte webbasierte Schnittstelle **122**, die beispielhafte Datenverarbeitungseinrichtung **206**, die beispielhafte Datentypdatenbank **208**, den beispielhaften OPC-Wandler **120**, den beispielhaften Adapter **118**, die beispielhafte OPC-Server-Verweisdatenbank **210** und/oder allgemeiner den Wrapper **110** aus [Fig. 1](#) und/oder [Fig. 2](#) zu implementieren. Obwohl das beispielhafte Prozessorsystem P10 als eine Mehrzahl von Peripherievorrichtungen, Schnittstellen, Chips, Speicher usw. enthaltend beschrieben ist, können ein oder mehrere dieser Elemente aus anderen beispielhaften Prozessorsystemen fortgelassen werden, die zum Implementieren von einem oder mehreren der beispielhaften Sicherheitsverarbeitungseinrichtung **202**, der beispielhaften Sitzungssteuerung **204**, der beispielhaften webbasierten Schnittstelle **122**, der beispielhaften Datenverarbeitungseinrichtung **206**, der beispielhaften Datentypdatenbank **208**, des beispielhaften OPC-Wandlers **120**, des beispielhaften Adapters **118**, der beispielhaften OPC-Server-Verweisdatenbank **210** und/oder allgemeiner des Wrappers **110** benutzt werden.

[0114] Wie in [Fig. 12](#) gezeigt, weist das Prozessorsystem P10 einen Prozessor P12 auf, der an einen Verbindungsbus P14 gekoppelt ist. Der Prozessor P12 weist einen Registersatz oder einen Registerraum P16 auf, der in [Fig. 12](#) als vollständig auf dem Chip angeordnet dargestellt ist, der aber alternativ auch vollständig oder teilweise außerhalb des Chips angeordnet sein kann und direkt über dedizierte elektrische Verbindungen und/oder über den Verbindungsbus P14 an den Prozessor P12 gekoppelt sein kann. Bei dem Prozessor P12 kann es sich um jeden beliebigen geeigneten Prozessor, eine Verarbeitungseinheit oder einen Mikroprozessor handeln. Obwohl in [Fig. 12](#) nicht dargestellt, kann das System P10 ein Multiprozessorsystem sein und kann daher einen oder mehrere weitere Prozessoren enthalten, die identisch mit dem Prozessor P12 oder diesem ähnlich sind und die kommunizierend an den Verbindungsbus P14 gekoppelt sind.

[0115] Der Prozessor aus [Fig. 12](#) ist an einen Chipsatz P18 gekoppelt, der eine Speichersteuerung P20 und eine Peripherie-Eingangs-/Ausgangs-(E/A)-Steuerung P22 aufweist. Wie allgemein bekannt, stellt ein Chipsatz typischerweise E/A- und Speicherverwaltungsfunktionen sowie eine Mehrzahl allgemeiner und/oder spezieller Register, Timer usw. bereit, die für einen oder mehrere Prozessoren, die an den Chipsatz P18 gekoppelt sind, zugänglich oder nutzbar sind. Die Speichersteuerung P20 führt Funktionen aus, die es dem Prozessor P12 (oder den Prozessoren, wenn mehrere Prozessoren vorliegen) ermöglicht, auf einen Systemspeicher P24 und einen Massenspeicher P25 zuzugreifen.

[0116] Bei dem Systemspeicher P24 kann es sich um jeden gewünschten Typ von flüchtigem und/oder nicht-flüchtigem Speicher handeln, wie z. B. statischen Schreib-/Lesespeicher (SRAM, static random access memory), dynamischen Schreib-/Lesespeicher (DRAM, dynamic random access memory), Flash-Speicher, Lesespeicher (ROM, read-only memory) usw. Bei dem Massenspeicher P25 kann es sich um jeden gewünschten Typ von Massenspeichervorrichtung handeln. Wenn beispielsweise das beispielhafte Prozessorsystem P10 dazu benutzt wird, den Wrapper **110** ([Fig. 2](#)) zu implementieren, kann der Massenspeicher P25 eine Festplatte, ein optisches Laufwerk, eine Bandspeichervorrichtung usw. aufweisen. Wenn das Prozessorsystem P10

dazu benutzt wird, die beispielhafte Datentypdatenbank **210** und/oder die beispielhafte OPC-Server-Verweisdatenbank **210** zu implementieren, kann es sich bei dem Massenspeicher P25 alternativ um Festspeicher (z. B. einen Flash-Speicher, einen RAM-Speicher usw.), einen Magnetspeicher (z. B. eine Festplatte) oder jeden beliebigen anderen Speicher handeln, der zur Massenspeicherung in der beispielhaften Datentypdatenbank **210** und/oder der beispielhaften OPC-Server-Verweisdatenbank **210** geeignet ist.

[0117] Die Peripherie-E/A-Steuerung P22 führt Funktionen aus, die es dem Prozessor P12 ermöglichen, über einen Peripherie-E/A-Bus P32 mit Peripherie-Eingabe/Ausgabe-(E/A)-Vorrichtungen P26 und P28 und einer Netzwerkschnittstelle P30 zu kommunizieren. Bei den Peripherie-E/A-Vorrichtungen P26 und P28 kann es sich um jeden beliebigen Typ von E/A-Vorrichtung handeln, beispielsweise um eine Tastatur, eine Anzeige (z. B. eine Flüssigkristallanzeige (LCD), eine Kathodenstrahlröhre (CRT usw.), eine Navigationsvorrichtung (z. B. eine Maus, einen Trackball, ein kapazitives Berührungsfeld, einen Joystick usw.) usw. Die Netzwerkschnittstelle P30 kann beispielsweise eine Ethernet-Vorrichtung, eine ATM-(asynchronous transfer mode)-Vorrichtung, eine 802.11-Vorrichtung, ein DSL-Modem, ein Kabelmodem, ein zelluläres Modem usw. sein, die es dem Prozessorsystem P10 ermöglicht, mit einem anderen Prozessorsystem zu kommunizieren.

[0118] Obwohl die Speichersteuerung P20 und die E/A-Steuerung P22 in **Fig. 12** als separate Funktionsblöcke im Chipsatz P18 dargestellt sind, können die Funktionen, die von diesen Blöcken ausgeübt werden, in einem einzelnen Halbleiterschaltkreis integriert sein oder unter Verwendung von zwei oder mehr separaten integrierten Schaltkreisen implementiert sein.

[0119] Wenigstens einige der oben beschriebenen beispielhaften Verfahren und/oder Vorrichtungen werden durch ein oder mehrere Software- und/oder Firmware-Programme implementiert, die auf einem Computerprozessor ausgeführt werden. Allerdings können auch dedizierte Hardware-Implementierungen, darunter, ohne darauf eingeschränkt zu sein, anwendungsspezifische integrierte Schaltkreise, programmierbare Logik-Arrays und sonstige Hardware-Vorrichtungen dazu konstruiert werden, einen entweder ganz oder teilweise Teil oder die Gesamtheit der hier beschriebenen beispielhaften Verfahren und/oder Vorrichtungen zu implementieren. Ferner können auch alternative Software-Implementierungen, darunter, ohne darauf eingeschränkt zu sein, verteilte Verarbeitung und/oder komponenten/objektverteilte Verarbeitung, parallele Verarbeitung oder virtuelle Maschinenverarbeitung dazu konstruiert werden, die hier beschriebenen beispielhaften Verfahren und/oder Vorrichtungen zu implementieren.

[0120] Es ist außerdem zu beachten, dass die hier beschriebenen beispielhaften Verfahren und/oder Vorrichtungen auf einem konkreten Speichermedium gespeichert sind, beispielsweise: einem Magnetmedium (z. B. einer Magnetdisc oder einem Magnetband); einem magneto-optischen oder optischen Medium wie z. B. einer optischen Disc; oder einem Festmedium wie z. B. einer Speicherkarte oder anderen Packung, die einen oder mehrere Lesespeicher (nicht-flüchtige Speicher), Lese-/Schreibspeicher oder andere wieder beschreibbare (flüchtige) Speicher enthält. Entsprechend kann die hier beschriebene beispielhafte Software und/oder Firmware auf einem konkreten Speichermedium wie den oben beschriebenen oder auf Nachfolger-Speichermedien gespeichert sein. Soweit die vorstehende Beschreibung beispielhafte Komponenten und Funktionen unter Bezugnahme auf bestimmte Standards und Protokolle beschreibt, versteht es sich, dass der Umfang dieses Patents nicht auf diese Standards und Protokolle beschränkt ist. Beispielsweise stellen die einzelnen Standards für die Internet- und andere paketvermittelte Netzwerkübertragung (z. B. Transmission Control Protocol (TCP)/Internet Protocol (IP), User Datagram Protocol (UDP)/IP, HyperText Markup Language (HTML), HyperText Transfer Protocol (HTTP)) Beispiele des aktuellen Stands der Technik dar. Diese Standards werden regelmäßig durch schnellere oder effizientere Äquivalente mit derselben allgemeinen Funktion ersetzt. Daher sind ersetzte Standards und Protokolle mit den gleichen Funktionen Äquivalente, die von diesem Patent vorgesehen wurden und damit in den Umfang der begleitenden Ansprüche fallen.

[0121] Obwohl das vorliegende Patent beispielhafte Verfahren und Vorrichtungen beschreibt, die Software oder Firmware enthalten, die auf Hardware ausführbar ist, ist außerdem zu beachten, dass diese Beispiele nur veranschaulichend sind und nicht als einschränkend auszulegen sind. Beispielsweise ist es vorgesehen, dass ein Teil dieser Hardware- und Softwarekomponenten oder alle davon ausschließlich in Hardware, ausschließlich in Software oder in jeder beliebigen Kombination aus Hardware und/oder Software verkörpert sein können. Obwohl also die vorstehende Beschreibung beispielhafte Verfahren, Systeme und maschinenzugängliche Medien beschrieben hat, sind die Beispiele nicht der einzige Weg zur Implementierung dieser Verfahren, Systeme und maschinenzugänglichen Medien. Obwohl also bestimmte beispielhafte Verfahren, Systeme und maschinenzugängliche Medien beschrieben wurden, ist der Abdeckungsumfang dieses Patents nicht auf diese beschränkt. Im Gegenteil deckt dieses Patent alle Verfahren, Systeme und maschinenzugängliche Medi-

en ab, die entweder wörtlich oder unter der Äquivalenzdoktrin billigermaßen in den Umfang der beiliegenden Ansprüche fallen.

Patentansprüche

1. Verfahren zum Zugreifen auf Prozessdaten, die auf einem Server gespeichert sind, wobei das Verfahren Folgendes umfasst:

Empfangen einer Anforderung zum Zugreifen auf Prozessdaten über einen Webbrower; Identifizieren eines Servers, auf dem wenigstens ein Teil der Prozessdaten gespeichert ist, wobei der Server einem Interoperabilitäts-Datenverpackungsformat zugeordnet ist; Zugreifen auf den Server, um die Prozessdaten zu empfangen; Umwandeln der Prozessdaten aus einem Format, das dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, in ein Webbrowerformat; und Einbetten von wenigstens einem ersten Teil der umgewandelten Prozessdaten in wenigstens ein Datenfeld zur Anzeige über einen Webbrower.

2. Verfahren nach Anspruch 1, ferner umfassend: Authentifizieren der Anforderung vor dem Identifizieren des Servers, auf dem die Prozessdaten gespeichert sind.

3. Verfahren nach Anspruch 1, wobei das Identifizieren des Servers umfasst:

Zugreifen auf eine Liste, die auf einen Identifikator verweist, der der Anforderung zugeordnet ist.

4. Verfahren nach Anspruch 1, wobei das Interoperabilitäts-Datenverpackungsformat einer Open Packaging Convention entspricht.

5. Verfahren nach Anspruch 1, wobei das Einbetten von wenigstens einem ersten Teil der umgewandelten Prozessdaten in das wenigstens eine Datenfeld Folgendes umfasst:

Unterteilen der umgewandelten Prozessdaten nach wenigstens einem Datentyp;

Ermitteln, welcher Datentyp den einzelnen Datenfeldern zugeordnet ist;

Erzeugen eines Datenfelds für jeden Datentyp; und

Einbetten der Prozessdaten, die den einzelnen Datentypen zugeordnet sind, in ein entsprechendes Datenfeld.

6. Verfahren nach Anspruch 5, wobei das Ermitteln, welcher Datentyp welchem der einzelnen Datenfelder entspricht, Folgendes umfasst:

Ermitteln von Metadaten, die den einzelnen Datenteilen zugeordnet sind; und

Zuweisen der Metadaten zu dem entsprechenden Datenfeld.

7. Verfahren nach Anspruch 1, wobei die Anforderung von einem Client stammt, der den Webbrower benutzt.

8. Verfahren nach Anspruch 7, ferner umfassend:

Einbetten des wenigstens einen Datenfelds einschließlich des wenigstens ersten Teils der umgewandelten Prozessdaten in eine Webseite; und

Zurücksenden der Webseite einschließlich der Daten in dem wenigstens einen Datenfeld an den Client mithilfe des Webbrowsers, um wenigstens den ersten Teil der umgewandelten Prozessdaten auf der Webseite anzuzeigen.

9. Verfahren nach Anspruch 1, wobei die Anforderung von einem Client stammt, der eine Client-Anzeigeanwendung benutzt, die in dem Webbrower ausgeführt wird.

10. Verfahren nach Anspruch 9, ferner Folgendes umfassend: Senden einer Anweisung an den Webbrower, die die Client-Anzeigeanwendung dazu veranlasst, auf das wenigstens eine Datenfeld mit dem wenigstens ersten Teil der umgewandelten Prozessdaten zuzugreifen und diese anzuzeigen.

11. Verfahren nach Anspruch 1, wobei auf dem Server ein erster Teil der Prozessdaten gespeichert ist.

12. Verfahren nach Anspruch 11, ferner umfassend:

Identifizieren eines zweiten Servers, auf dem ein zweiter Teil der Prozessdaten gespeichert ist, wobei der zweite Server dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist;

Zugreifen auf den zweiten Server, um den zweiten Teil der Prozessdaten zu empfangen;

Umwandeln des zweiten Teils der Prozessdaten aus dem Format, das dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, in ein Webbrowserformat; und
Einbetten des zweiten Teils der umgewandelten Prozessdaten in das wenigstens eine Datenfeld zur Anzeige zusammen mit dem ersten Teil der umgewandelten Prozessdaten über den Webbrowser.

13. Verfahren nach Anspruch 1, ferner umfassend:
Empfangen von Zielanzeigeinformation; und
Ermitteln einer Vorlage, die das wenigstens eine Datenfeld enthält, das der Zielanzeigeinformation und den Prozessdaten entspricht.

14. Vorrichtung zum Zugreifen auf Prozessdaten, die auf einem Server gespeichert sind, wobei die Vorrichtung Folgendes umfasst:
einen Adapter zum:
Identifizieren eines Servers, auf dem wenigstens ein Teil von Prozessdaten gespeichert ist, die einer Anforderung zugeordnet sind, über einen Webbrowser auf die Prozessdaten zuzugreifen, wobei der Server einem Interoperabilitäts-Datenverpackungsformat zugeordnet ist; und
Zugreifen auf den Server, um die Prozessdaten zu empfangen; einen Wandler zum Umwandeln der Prozessdaten aus einem Format, das dem Interoperabilitäts-Datenverpackungsformat zugeordnet ist, in ein Webbrowserformat; und
eine webbasierte Schnittstelle, um wenigstens einen Teil der umgewandelten Prozessdaten zur Anzeige über den Webbrowser in wenigstens ein Datenfeld einzubetten.

15. Vorrichtung nach Anspruch 14, wobei die webbasierte Schnittstelle dazu dient, die Anforderung zum Zugreifen auf Prozessdaten über den Webbrowser zu empfangen.

16. Vorrichtung nach Anspruch 15, wobei die Anforderung von einem Client stammt, der den Webbrowser benutzt.

17. Vorrichtung nach Anspruch 16, wobei die webbasierte Schnittstelle zu Folgendem dient:
Einbetten des wenigstens einen Datenfelds mit wenigstens dem Teil der umgewandelten Prozessdaten in eine Webseite; und
Zurücksenden der Webseite einschließlich der Daten in dem wenigstens einen Datenfeld an den Client;
Verwenden des Webrowsers, um wenigstens den Teil der umgewandelten Prozessdaten auf der Webseite anzuzeigen.

18. Vorrichtung nach Anspruch 15, wobei die Anforderung von einem Client stammt, der eine Client-Anzeigeanwendung benutzt, die in dem Webbrowser ausgeführt wird.

19. Vorrichtung nach Anspruch 18, wobei die webbasierte Schnittstelle dazu dient, eine Anweisung an den Webbrowser zu senden, die die Client-Anzeigeanwendung dazu veranlasst, auf das wenigstens eine Datenfeld mit wenigstens dem Teil der umgewandelten Prozessdaten zuzugreifen und diese anzuzeigen.

20. Vorrichtung nach Anspruch 18, wobei die Client-Anzeigeanwendung wenigstens eine von einer ActiveX™-Anwendung, einer Silverlight™-Anwendung, einer Adobe Flash™-Anwendung, einer Hypertext Markup Language 5-Anwendung oder einer Sun JavaFX™-Anwendung enthält.

21. Vorrichtung nach Anspruch 14, ferner aufweisend eine Sicherheitsverarbeitungseinrichtung zum Authentifizieren der Anforderung.

22. Vorrichtung nach Anspruch 14, wobei der Adapter den Server identifiziert, indem er auf eine Liste zugreift, die auf einen Identifikator verweist, der der Anforderung an den Server zugeordnet ist.

23. Vorrichtung nach Anspruch 14, wobei das Interoperabilitäts-Datenverpackungsformat einer Open Packaging Convention entspricht.

24. Vorrichtung nach Anspruch 14, wobei die webbasierte Schnittstelle dazu dient, wenigstens den Teil der umgewandelten Prozessdaten in das wenigstens eine Datenfeld einzubetten durch:
Unterteilen der umgewandelten Prozessdaten nach wenigstens einem Datentyp;
Ermitteln, welcher Datentyp den einzelnen Datenfeldern zugeordnet ist;
Erzeugen eines Datenfelds für jeden Datentyp; und

Einbetten der Prozessdaten, die den einzelnen Datentypen zugeordnet sind, in ein entsprechendes Datenfeld.

25. Vorrichtung nach Anspruch 24, wobei das Ermitteln, welcher Datentyp welchem der einzelnen Datenfelder entspricht, eine Datenverarbeitungseinrichtung beinhaltet zum:
Ermitteln von Metadaten, die den einzelnen Datenteilen zugeordnet sind; und
Zuweisen der Metadaten zu dem entsprechenden Datenfeld.

26. Vorrichtung nach Anspruch 14, wobei auf dem Server ein erster Teil der Prozessdaten gespeichert ist.

Es folgen 13 Blatt Zeichnungen

Anhängende Zeichnungen

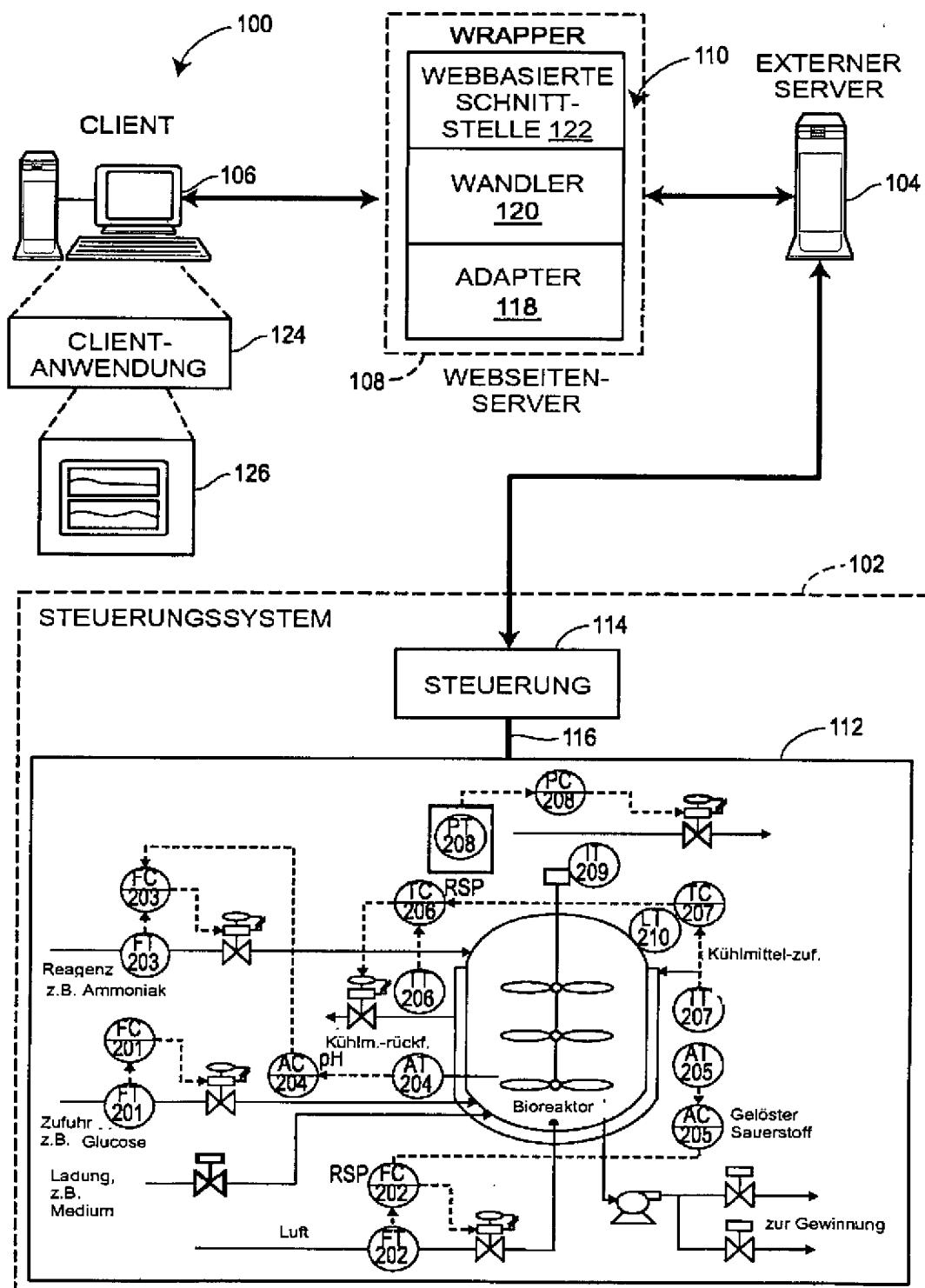


FIG. 1

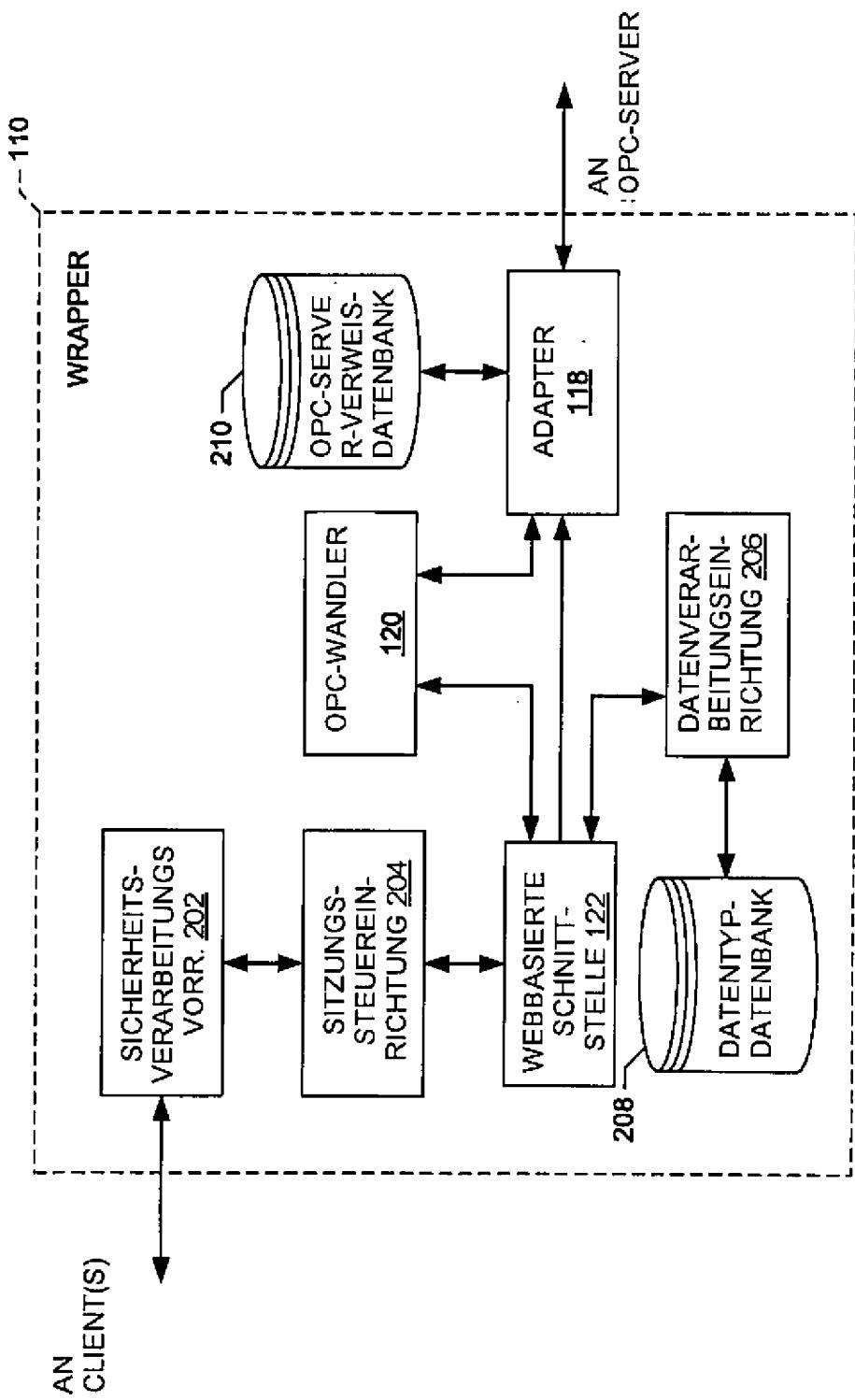
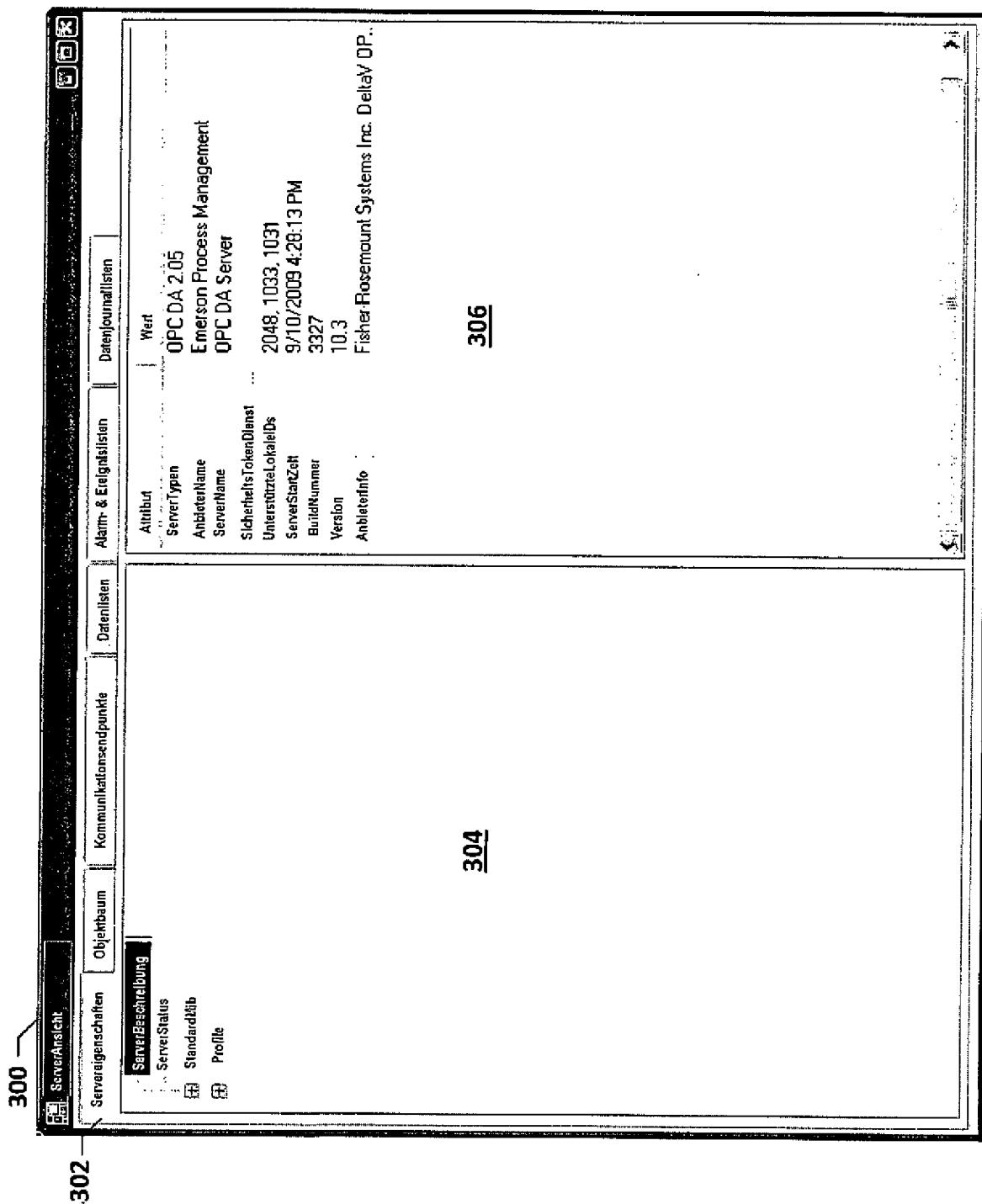
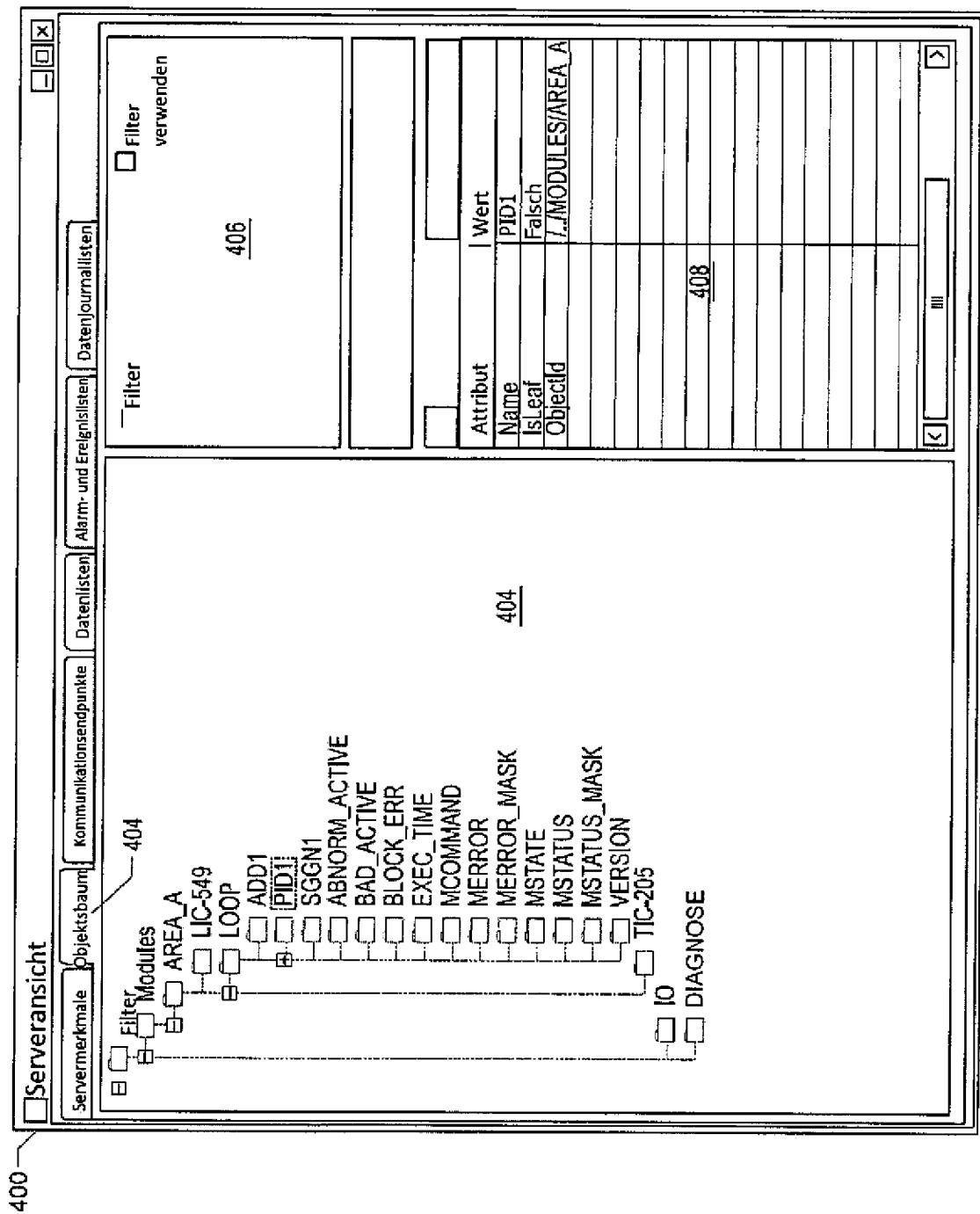


FIG. 2

3
FIG.





500

Server-Ansicht

Modifizieren Objektbaum Kommunikationsendpunkte Datenlisten Alarm- & Ereignislisten Datenjournalisten

502

Zugeordnete Endpunkte für ausgewählte Liste

Aktiviert	Client-ID	Server-ID	Update-Rate	Stück
Wahr	232152295	738923058	5000	0

504

Aktivieren Deaktivieren Entfernen Berichten Lesen Schreiben

Geltend für

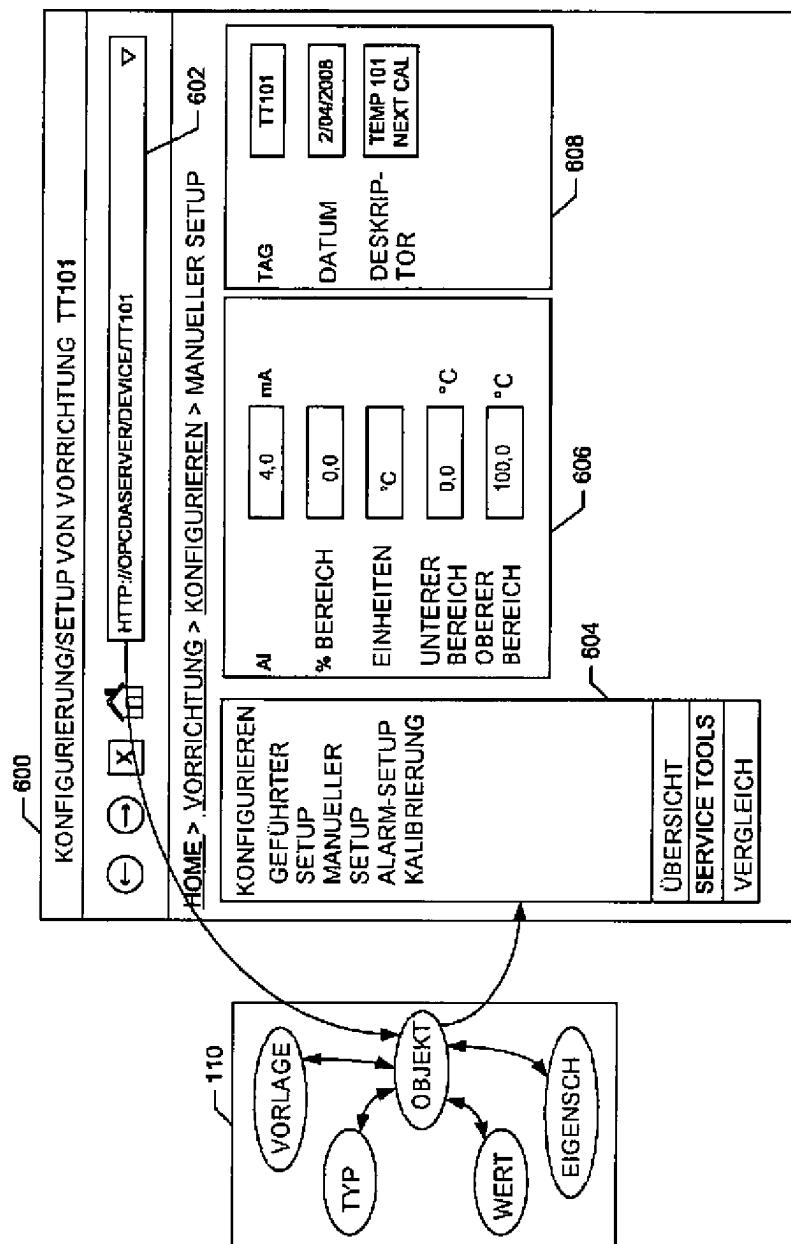
Auswählen Alle

Hinzufügen

Aktiviert	Instant-ID	Datenwert	Statuscode	Zeitstempel	Datentyp	Aktual
Wahr	1L00P/SGGN1/0U	-9.170589	0x0000	2009-09-11	System.Sinh...	
Wahr	1L00P/PID1/IN.CV	43.86825	0x0000	2009-09-11	System.Sinh...	
Wahr	1L00P/PID1/SP.CV	50	0x0000	2009-09-11	System.Sinh...	
Wahr	1L00P/RID1/RV.CV	43.86825	0x0000	2009-09-11	System.Sinh...	
Wahr	1L00P/RID1/OUT.CV	53.22187	0x0000	2009-09-11	System.Sinh...	

506

FIG. 5



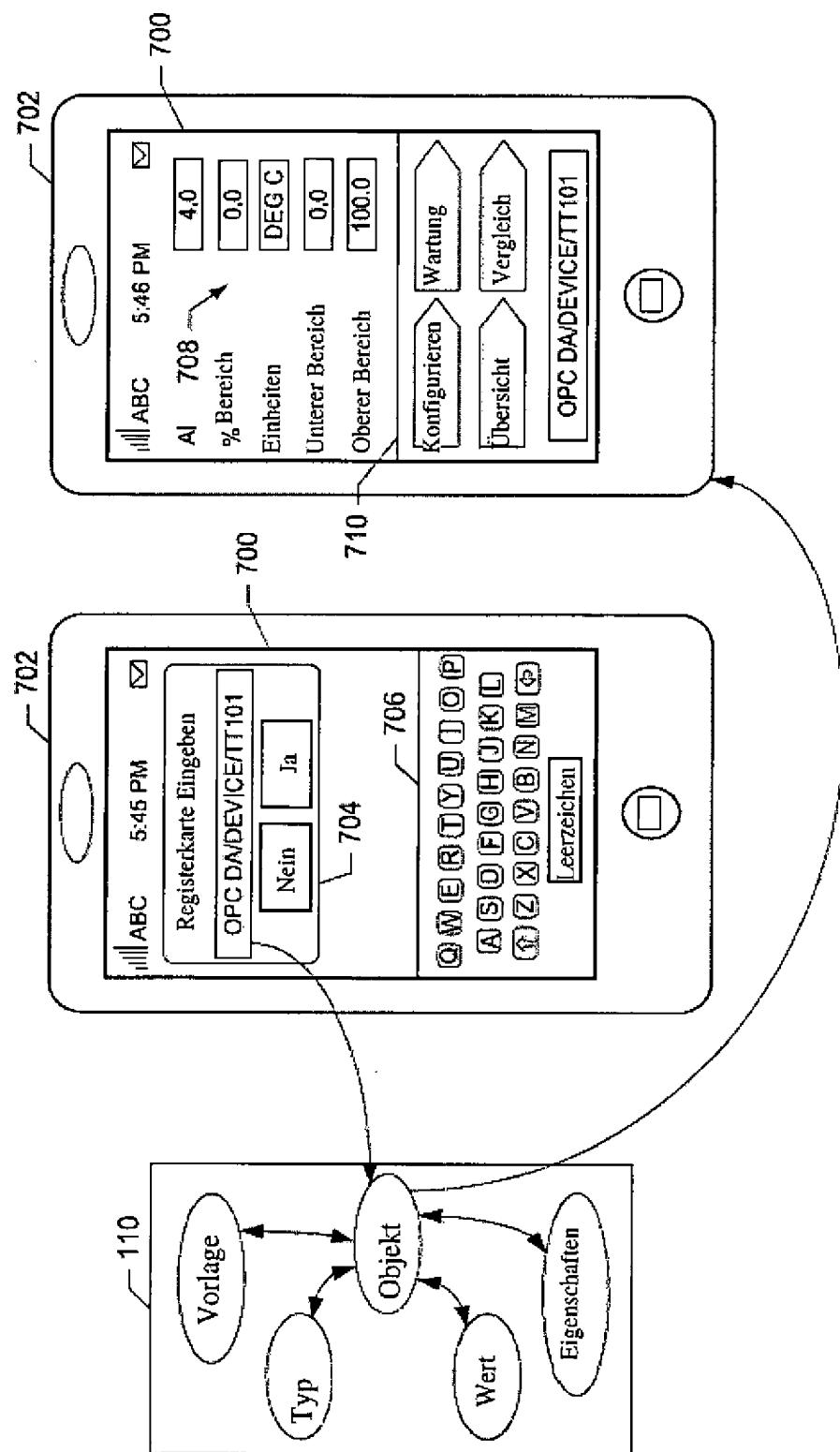


FIG. 7A

FIG. 7B

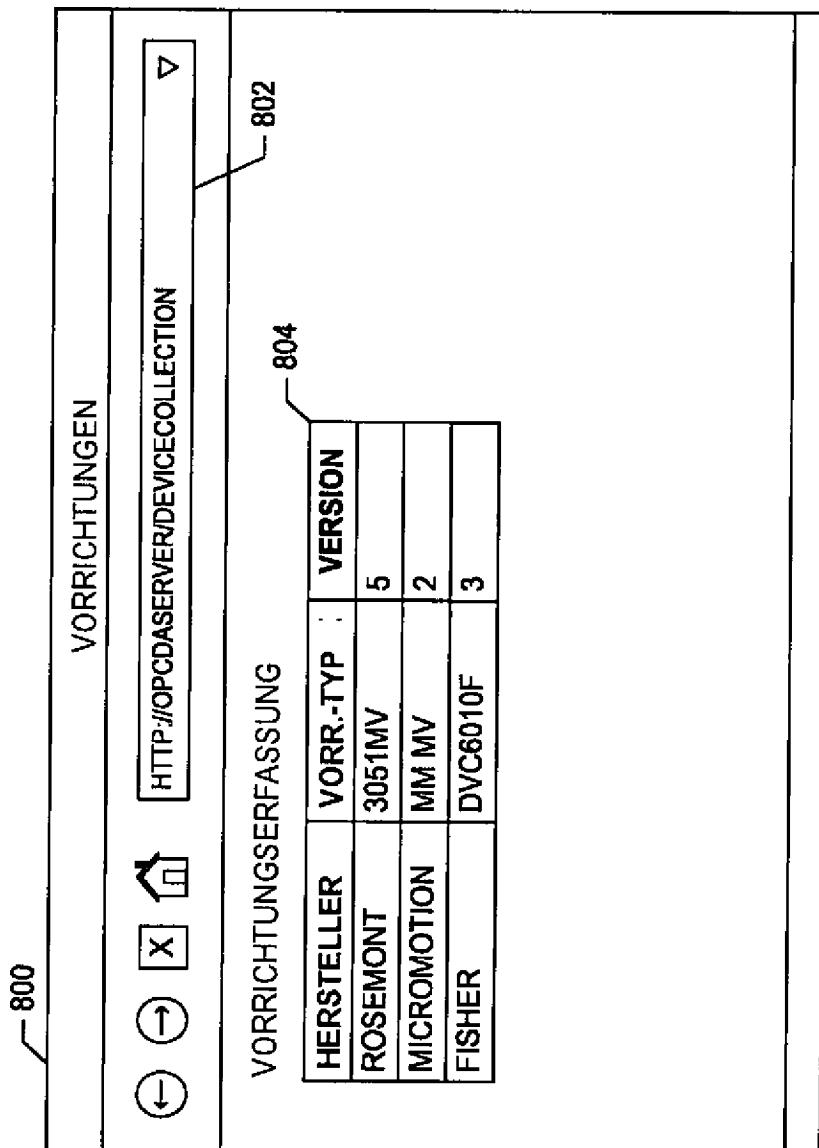


FIG. 8

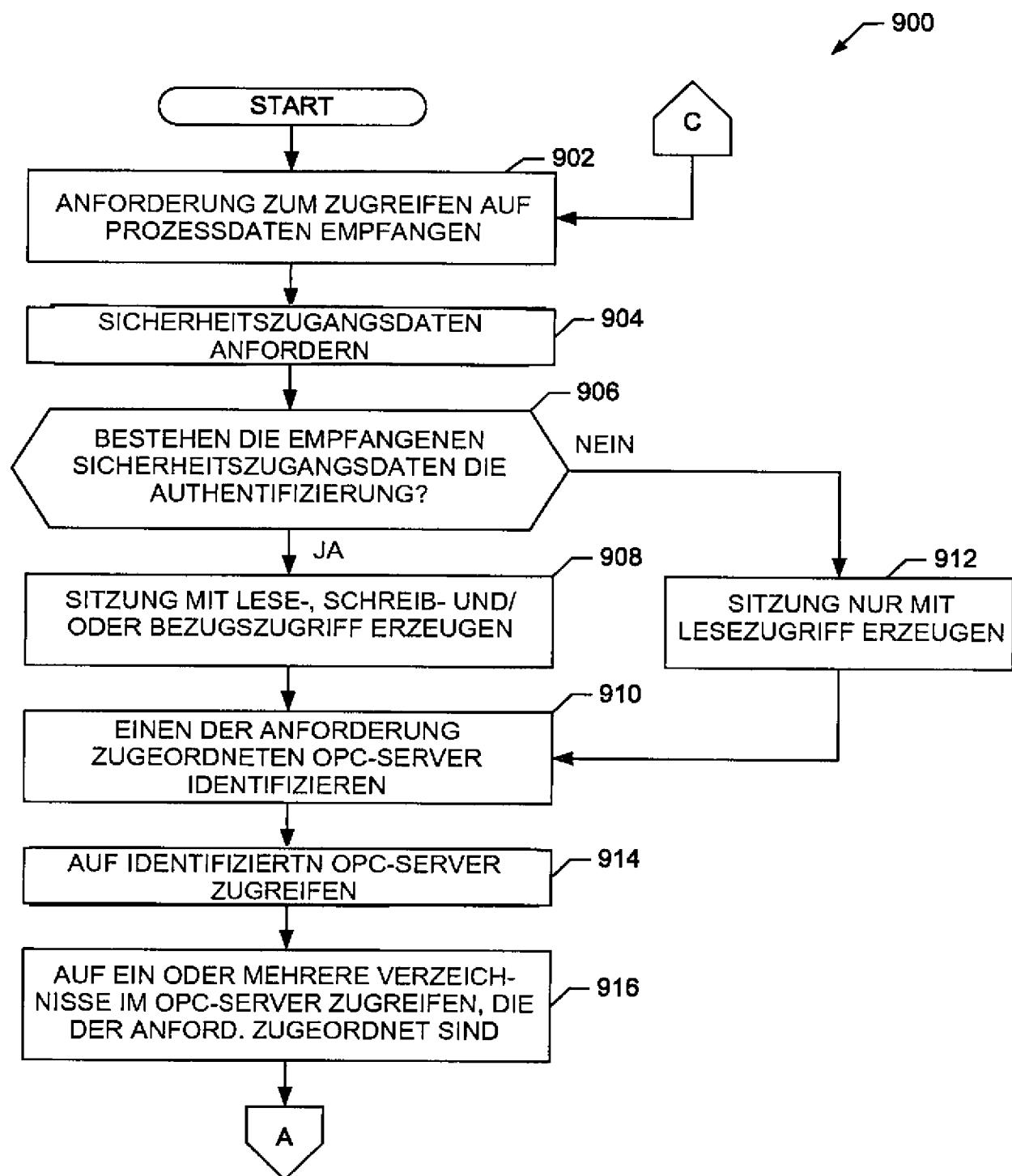


FIG. 9A

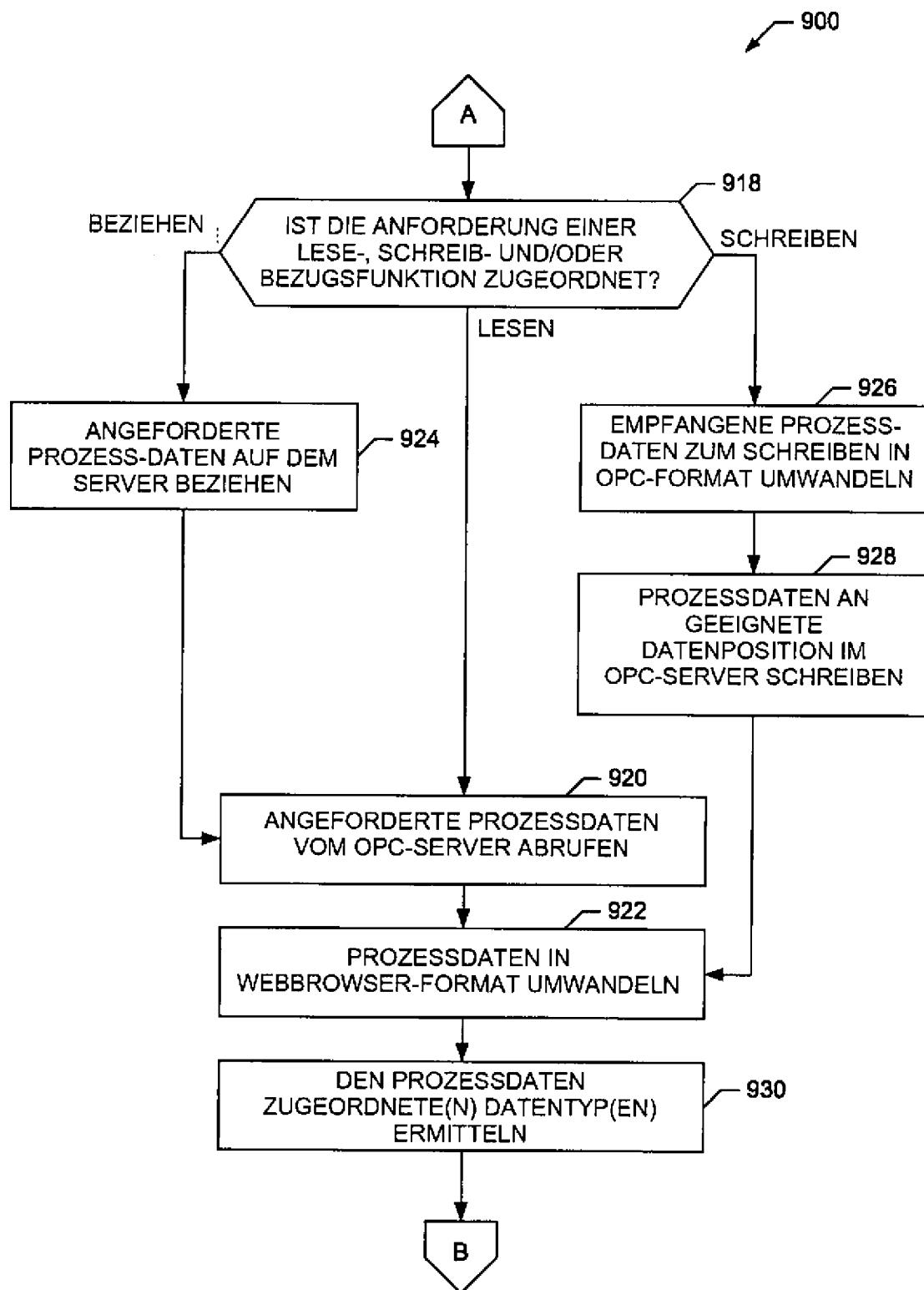


FIG. 9B

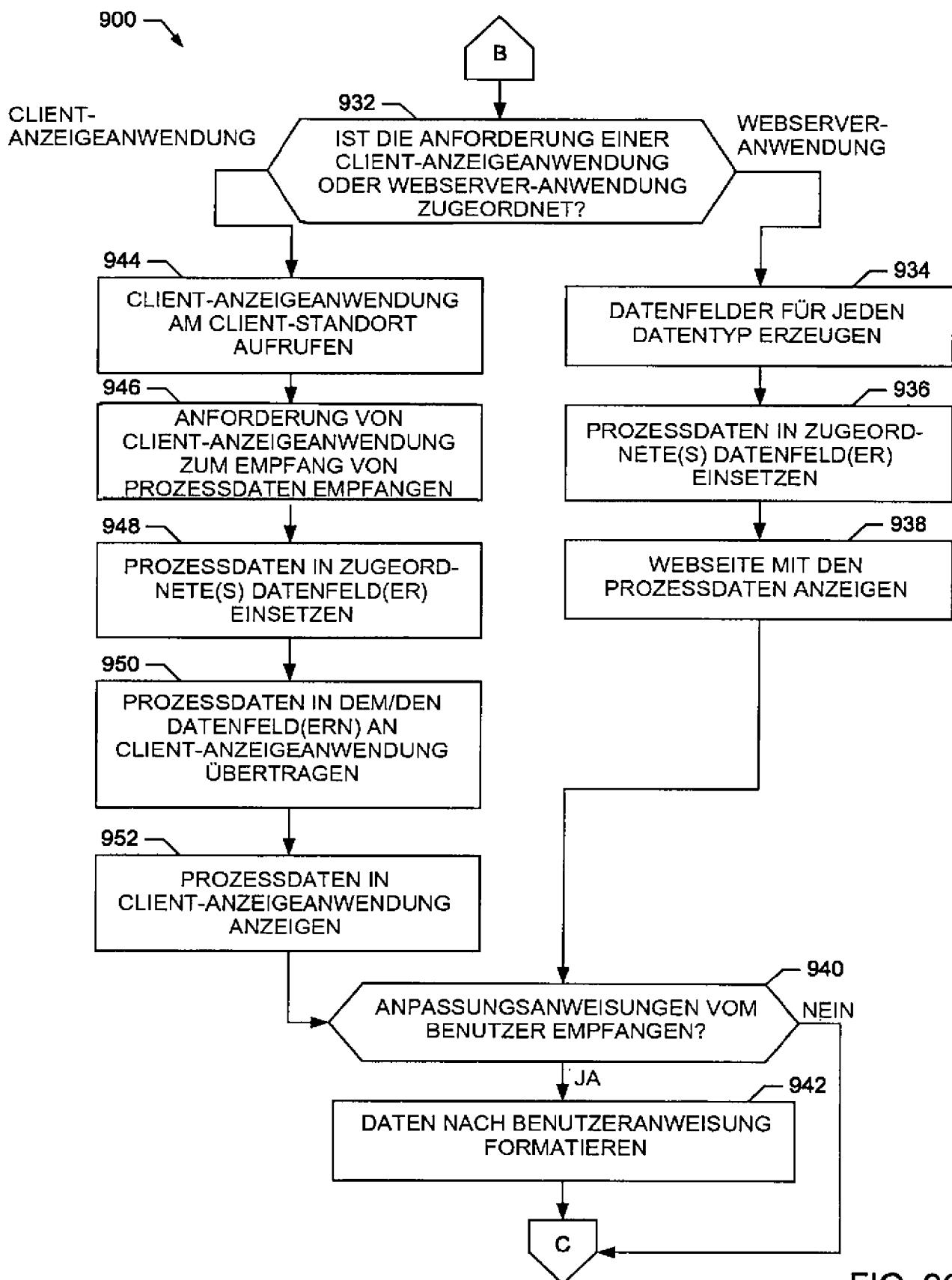


FIG. 9C

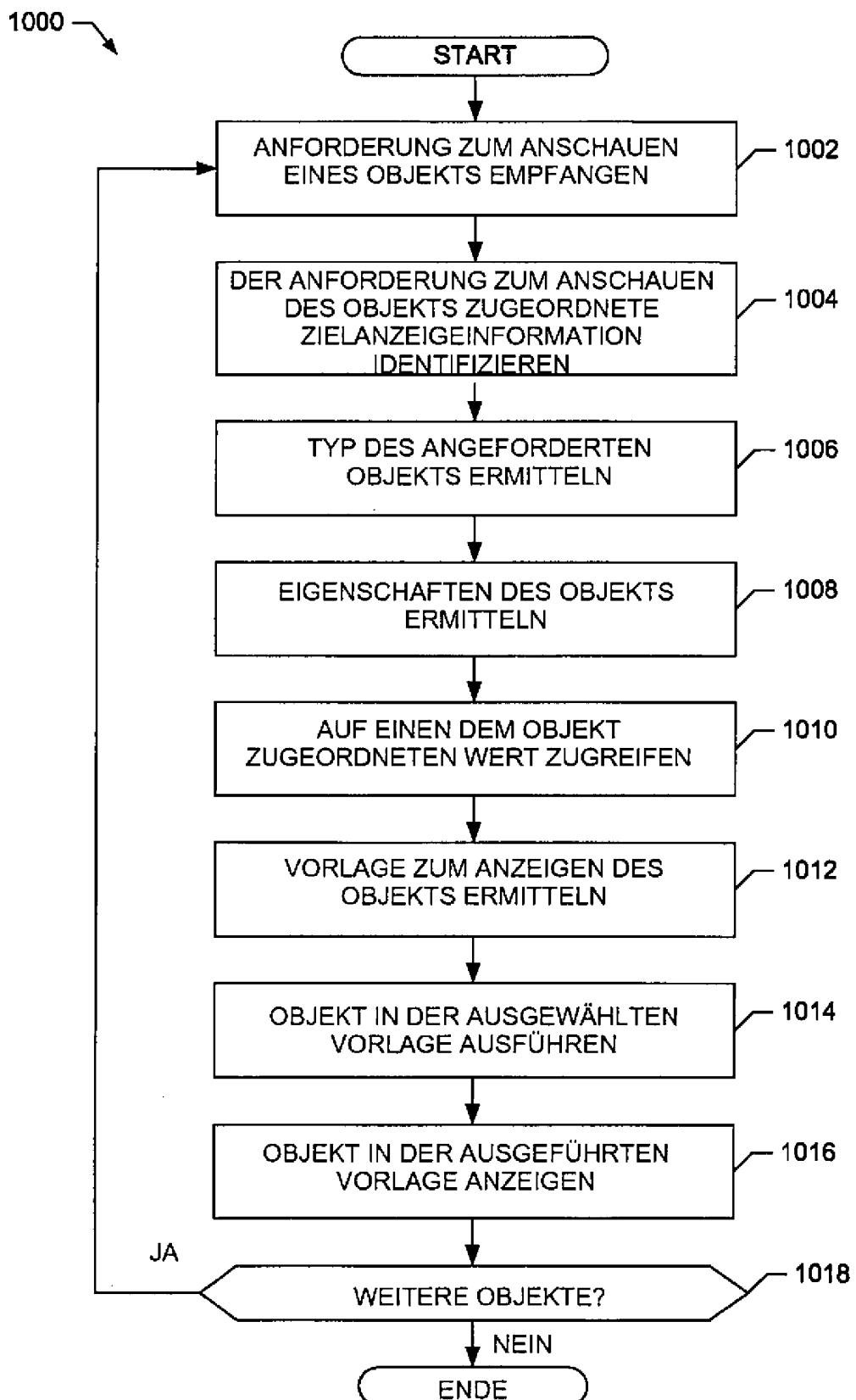


FIG. 10

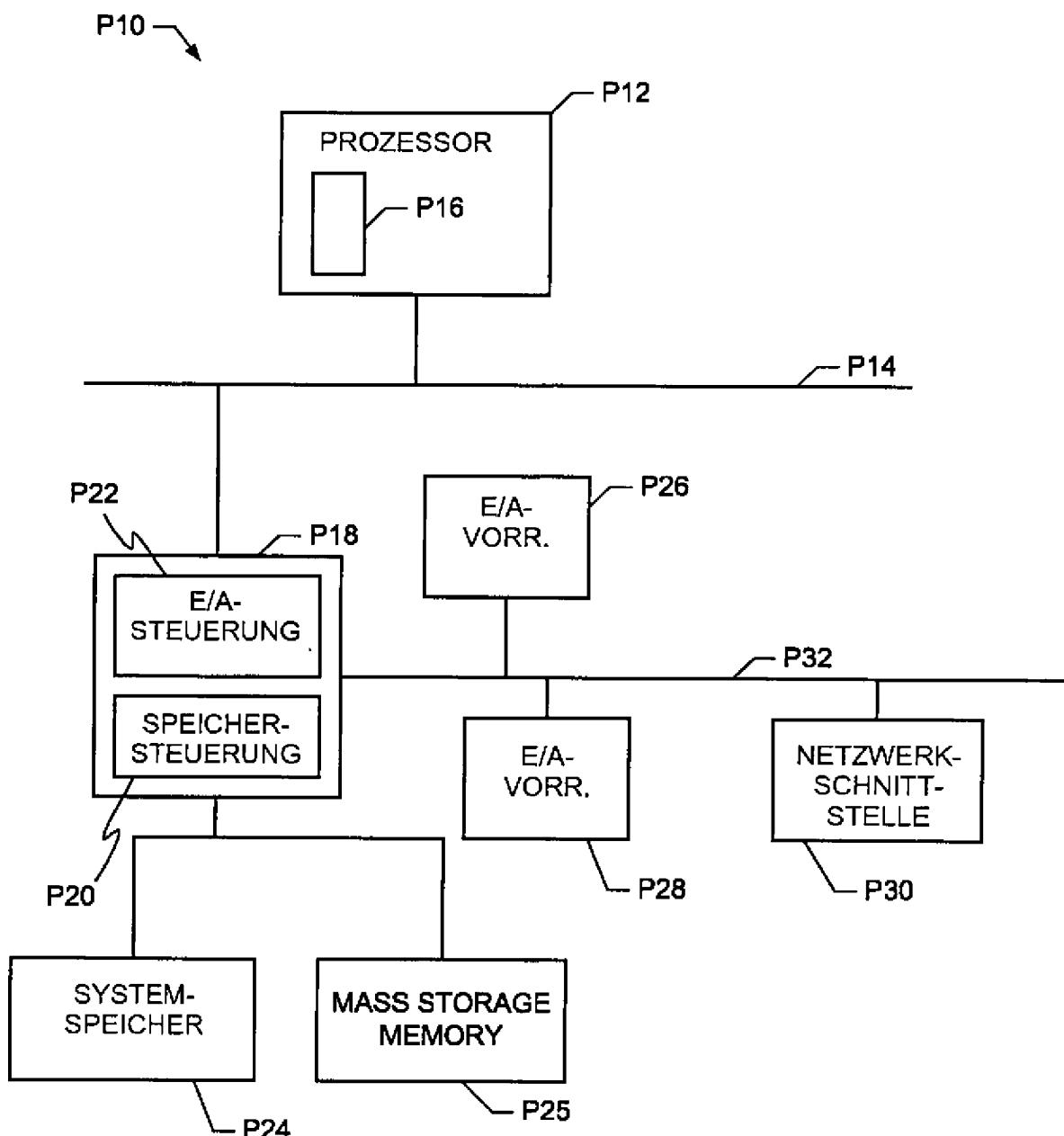


FIG. 11