

(12) NACH DEM VERTRAG ÜBER DIE INTERNATIONALE ZUSAMMENARBEIT AUF DEM GEBIET DES PATENTWESENS (PCT) VERÖFFENTLICHTE INTERNATIONALE ANMELDUNG

(19) Weltorganisation für geistiges Eigentum
Internationales Büro

(43) Internationales Veröffentlichungsdatum
16. April 2020 (16.04.2020)



(10) Internationale Veröffentlichungsnummer
WO 2020/074225 A1

(51) Internationale Patentklassifikation:

H04L 12/24 (2006.01) H04L 29/08 (2006.01)
G06F 9/455 (2018.01) H04L 29/12 (2006.01)

(21) Internationales Aktenzeichen: PCT/EP2019/074941

(22) Internationales Anmeldedatum:
18. September 2019 (18.09.2019)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
18200134.7 12. Oktober 2018 (12.10.2018) EP

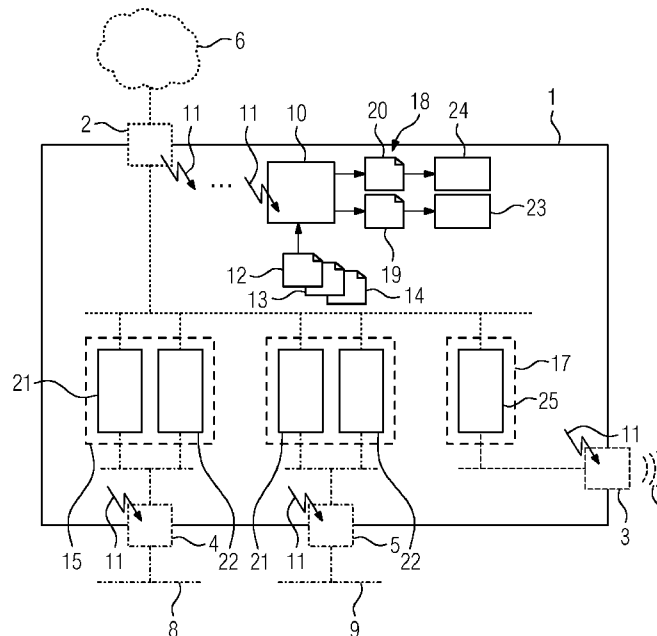
(71) Anmelder: **SIEMENS AKTIENGESELLSCHAFT**
[DE/DE]; Werner-von-Siemens-Straße 1, 80333 München (DE).

(72) Erfinder: **ALBRECHT, Harald**; Gleiwitzer Str. 555, 90475 Nürnberg (DE). **HÖME, Stephan**; Dr.-Gustav-Heinemann-Str. 26, 91126 Schwabach (DE). **TALANIS, Thomas**; Adenauerstr. 22, 91336 Heroldsbach (DE).

(81) Bestimmungsstaaten (soweit nicht anders angegeben, für jede verfügbare nationale Schutzrechtsart): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(54) Title: METHOD FOR AUTOMATICALLY CONFIGURING A SYSTEM, SYSTEM, COMPUTER PROGRAMME, AND COMPUTER-READABLE MEDIUM

(54) Bezeichnung: VERFAHREN ZUM AUTOMATISCHEN KONFIGURIEREN EINES SYSTEMS, SYSTEM, COMPUTERPROGRAMM UND COMPUTERLESBARES MEDIUM



(57) Abstract: The invention relates to a method for automatically configuring a system (1), in which method - when the system (1) is activated and/or during operation of the system (1), monitoring which physical and/or virtual hardware network interfaces (2, 3, 4, 5) the system (1) comprises, - if a hardware network interface (2, 3, 4, 5) is detected for the first time when the system (1) is activated, or a hardware network interface (2, 3, 4, 5) is newly added during operation of the system (1), this is communicated to an auto-configuration module (10) of the system (1), - the auto-configuration module (10) creates a configuration description (18) using a template file (12, 13, 14), and - the configuration description is implemented in order to configure the system (1) in accordance with said description. The invention also relates to a system (1), in particular an automation system, for carrying out such a method, to a computer programme,



WO 2020/074225 A1

(84) Bestimmungsstaaten (soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, RU, TJ, TM), europäisches (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Veröffentlicht:

— mit internationalem Recherchenbericht (Artikel 21 Absatz 3)

and to a computer-readable medium.

(57) Zusammenfassung: Verfahren zum automatischen Konfigurieren eines Systems, System, Computerprogramm und computerlesbares Medium Die Erfindung betrifft ein Verfahren zum automatischen Konfigurieren eines Systems (1), bei dem - bei Inbetriebnahme des Systems (1) und/oder im laufenden Betrieb des Systems (1) überwacht wird, welche physikalischen und/oder virtuellen Hardware-Netzwerkschnittstellen (2, 3, 4, 5) das System (1) aufweist, - für den Fall, dass eine Hardware-Netzwerkschnittstelle (2, 3, 4, 5) bei Inbetriebnahme des Systems (1) erstmals erfasst wird, oder eine Hardware-Netzwerkschnittstelle (2, 3, 4, 5) im Betrieb des Systems (1) neu hinzukommt, dies einem Autokonfigurationsmodul (10) des Systems (1) kommuniziert wird, - das Autokonfigurationsmodul (10) unter Verwendung einer Schablonen-Datei (12, 13, 14) eine Konfigurationsbeschreibung (18) erstellt, und - die Konfigurationsbeschreibung umgesetzt wird, um das System (1) gemäß dieser zu konfigurieren. Darüber hinaus betrifft die Erfindung ein System (1), insbesondere Automatisierungssystem, zur Durchführung eines solchen Verfahrens, ein Computerprogramm und ein computerlesbares Medium.

Beschreibung

Verfahren zum automatischen Konfigurieren eines Systems, System, Computerprogramm und computerlesbares Medium

5

Die Erfindung betrifft ein Verfahren zum automatischen Konfigurieren eines Systems, insbesondere eines Automatisierungssystems. Darüber hinaus betrifft die Erfindung ein System, insbesondere Automatisierungssystem, zur Durchführung eines solchen Verfahrens, ein Computerprogramm und ein computerlesbares Medium.

Automatisierungssysteme insbesondere für industrielle Automatisierungsanlagen, zum Beispiel speicherprogrammierbare Steuerungen, Industrie-PCs oder dergleichen, besitzen in der Regel eine oder auch mehrere Netzwerk-Schnittstellen, bei denen es sich sowohl um physikalische Hardware-Netzwerk-Schnittstellen als auch um virtuelle Hardware-Netzwerk-Schnittstellen, zum Beispiel einer virtuellen Maschine, handeln kann. Verschiedene Systeme aus dem Bereich der Automatisierungstechnik können bereits ab Werk mit unterschiedlichen solcher Netzwerkschnittstellen ausgerüstet sein und es ist auch möglich, dass Anwender den Ausbau an physikalischen und/oder virtuellen Hardware-Netzwerk-Schnittstellen später ändern, etwa eine oder mehrere solcher Schnittstellen hinzufügen oder entfernen.

Im Betrieb sind die Schnittstellen an verschiedene Netzwerke angeschlossen. Beispielsweise kann eine solche Schnittstelle an einem Fabrik- oder Firmennetzwerk eines Betreibers angeschlossen sein, eine oder mehrere Schnittstellen an untergeordneten Maschinen- oder Zellennetzwerken, sowie eine oder mehrere Schnittstellen an (lokalen) Drahtlosnetzwerken usw.

Anwender möchten sowohl bei einer Erstinbetriebnahme als auch im laufenden Betrieb, wenn eine Änderung des Hardwareausbaus an Schnittstellen erfolgt, möglichst rasch und einfach eine durchgängige Konnektivität zwischen den verschiedenen Netzen,

beispielsweise einem Firmen- bzw. Produktionsnetz, unterlagerten Zellen- oder Maschinennetzen und einem Drahtlosnetzwerk, herstellen bzw. erhalten - und zwar entsprechend des von ihnen gewählten bzw. vorgenommenen bzw. geänderten Hardwareausbaus an Schnittstellen.

Die Netzwerkkonnektivität soll sich dabei möglichst nahtlos einerseits in die Belange industrieller Produktionsstätten einfügen, zugleich aber auch mit Firmen-IT harmonisieren, beispielsweise mit IPv6-Routern, die auch autokonfigurierend ausgestaltet sein können, wie beispielsweise in der EP 2 940 972 A1 und der EP 3 091 714 A1 vorgesehen, und/oder mit NAT64-Routern, die adaptiv ausgestaltet sein können, wie beispielsweise in der EP 3 062 490 A1 vorgesehen.

Im Bereich der allgemeinen IT hat sich die so genannte Container-Technologie, wie beispielsweise „Docker“ oder "rkt" (gesprochen "rocket") oder "Kubernetes" bewährt (siehe <https://www.docker.com> bzw. EP 3 267 351 A1, <https://coreos.com/rkt/>, <https://kubernetes.io/>).

Anwendungs-Container stellen in der Regel "abgekapselte" Einheiten dar, die unabhängig voneinander ausgeführt werden können, egal wo sie sich befinden. Ähnlich wie bei virtuellen Maschinen sind Container eine Art Behälter für Anwendungen, in denen diese laufen können. Während virtuelle Maschinen jedoch eine ganze Computerumgebung abbilden, enthalten die Container üblicherweise lediglich die wichtigen Daten, die für die Ausführung der jeweiligen Anwendung bzw. Anwendungen benötigt werden.

Die Container-Technologien bzw. Containerisierung ermöglichen es insbesondere, eine Software bzw. Anwendung in einem Container zu verpacken, der alles umfasst, was sie zum Laufen braucht, beispielsweise Programmcode, Laufzeit, System-Tools und/oder Systembibliotheken.

Die Container-Technologien bzw. Containerisierung erlauben es, Funktionen einfach und bequem modular zu verpacken, zu transportieren und schließlich in den operativen Bereich auszurollen. Klassische Einsatzgebiete von Containern liegen
5 beispielsweise darin Applikationen, etwa eine Datenbank oder einen Web-Front-End, bequem zu verpacken.

Ein Container wird in der Regel von einem Container Image, im Falle von Docker beispielsweise einem Docker Image, erzeugt
10 (instanziiert), welches als Daten-Template dient.

Nach Auffassung der Anmelderin hat die Container-Virtualisierung auch im Bereich der Automatisierungstechnik ein großes Potential. Denkbar ist beispielsweise, im Bereich der Automatisierungstechnik (virtuelle) Netzwerkfunktionen insbesondere
15 zum Erhalt von Konnektivität, etwa bei Änderungen des Hardwareausbaus, und/oder weitere Netzwerkfunktionen, containerisiert bereitzustellen bzw. zu nutzen.

20 Damit solche Container mit (virtuellen) Netzwerkfunktionen überhaupt einsatzbereit sind bzw. werden, müssen sie jedoch mit angeschlossenen Netzwerken bzw. Schnittstellen, die mit den Netzwerken verbunden sind, "verschaltet" bzw. "verdrahtet" werden.

25

Der Anmelderin ist bekannt, dass zum Verschalten von Virtuellen Maschinen (VMs) und Containern Software-Switches zum Einsatz kommen können, die auch "virtuelle" Switches genannt werden (siehe beispielsweise
30 <https://wiki.linuxfoundation.org/networking/bridge> sowie <https://wiki.debian.org/BridgeNetworkConnections>). Solche Software-Switches sind beispielsweise Teil des Linux-Kernels und können jederzeit im laufenden Betrieb hinzugefügt, verändert und auch wieder gelöscht werden, wofür jedoch nicht unerhebliches IT- bzw. Netzwerk-Know-How erforderlich ist. Solche
35 virtuellen Switches können beispielsweise bei Linux explizit in Form sogenannter Kernel-Bridges auftreten (siehe beispielsweise die vorangegangenen Verweise zum Stand der Tech-

nik). Sie können aber beispielsweise auch implizit Teil eines virtuellen Netzwerkes zwischen einer Hardware-Netzwerk-schnittstelle (den sog. Master-Schnittstellen) und weiteren virtuellen Macvlan-Netzwerk-Schnittstellen sein (siehe bei-

5 spielsweise <http://hucu.be/bridge-vs-macvlan> sowie zu Macvlans im Allgemeinen:

<https://docs.docker.com/network/macvlan/> und

<https://sreeninet.wordpress.com/2016/05/29/macvlan-and-ipvlan/> und

10 <https://github.com/docker/libnetwork/commit/dc12a0f09b20015fba55ac20cfc892b5ab91259#diff-e3e055a269aa170439b408dbcd4ea1b7>).

Die Docker-Technologie kennt weiterhin sogenannte "Docker
15 Network Drivers" mit dem "Container Network Model", siehe dazu beispielsweise <https://docs.docker.com/network/#network-drivers> und <https://blog.docker.com/2016/12/understanding-docker-networking-drivers-use-cases/>. Diese Docker Network Drivers erfordern jedoch von einem Anwender, dass er eine to-

20 pologische Beschreibung der benötigten Netzwerke und deren Verschaltung bereitstellt, was ebenfalls nicht unbeachtliches IT- bzw. Netzwerk-Know-How erfordert und mit nicht unerheblichem Aufwand verbunden ist.

25 Auf den Docker Network Drivers aufbauend existiert weiterhin die Technologie "Docker Compose" (siehe <https://docs.docker.com/compose/>). Hierbei handelt es sich um ein Werkzeug, um Applikationen zu beschreiben, die insbesondere aus mehreren Docker-Containern bestehen, inklusive deren

30 "Netzwerk-Verdrahtung". Allerdings kann Docker Compose lediglich eines oder mehrere Netzwerke referenzieren, die zuvor mittels Docker Driver definiert bzw. angelegt und konfiguriert wurden. Im Falle einer Bridge beispielsweise wird die Bridge angelegt, bei einem Macvlan wird vermerkt, dass dieses

35 Netzwerk definiert wurde, es werden aber noch keine konkreten Netzwerkelemente erzeugt. Das bedeutet, dass bei Macvlan das Verdrahten erst später erfolgt, während bei einer Bridge

schon die Teile angelegt werden, die unabhängig vom einzelnen Container benötigt werden.

Nach Auffassung der Anmelderin spielt bei der Containerisierung von (virtuellen) Netzwerkfunktionen nicht nur deren Einbindung in das externe, physikalische Netzwerk eine wichtige Rolle, sondern auch das korrekte Verdrahten der Container untereinander und mit der Netzwerk-Außenwelt.

Der Anmelderin ist bekannt, dass ein solches Verdrahten bisher in dem durch modulare Hardware geprägten Automatisierungsbereich durch das Stecken geeigneter Funktionsmodule in einer physischen Hardware-Steuerung, wie beispielsweise einer S7-1500 der Siemens AG, realisierbar wäre. Auch hier muss ein Anwender eingreifen, über entsprechendes Know-How verfügen und es ist zusätzliche Hardware erforderlich.

Es ist daher eine Aufgabe der vorliegenden Erfindung, eine Möglichkeit dafür zu schaffen, dass eine Verdrahtung von Containern mit (virtuellen) Netzwerkfunktionen untereinander und mit externen, physikalischen Netzwerken für den Anwender automatisiert und ohne Aufwand erfolgt.

Diese Aufgabe wird gelöst durch ein Verfahren zum automatischen Konfigurieren eines Systems, insbesondere eines Automatisierungssystems, bei dem

- bei Inbetriebnahme des Systems und/oder im laufenden Betrieb des Systems überwacht wird, welche physikalischen und/oder virtuellen Hardware-Netzwerkschnittstellen das System aufweist,
- für den Fall, dass eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle bei Inbetriebnahme des Systems erstmals erfasst wird, und/oder eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle im Betrieb des Systems neu hinzukommt oder wegfällt, dies einem Autokonfigurationsmodul des Systems zusammen mit einer Information über die Art der betroffenen Hardware-Netzwerkschnitt-

stelle und/oder die Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes kommuniziert wird,

- das Autokonfigurationsmodul für wenigstens eine der betroffenen Hardware-Netzwerkschnittstellen unter Verwendung einer bevorzugt auf dem System gespeicherten, der Art der wenigstens einen betroffenen Hardware-Netzwerkschnittstelle und/oder der Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes zugeordneten Schablonen-Datei eine Container-Konfigurationsbeschreibung erstellt, die eine erste Teilkonfigurationsbeschreibung für wenigstens einen Kommunikations-Container, der wenigstens eine Anwendung umfasst, mittels derer Netzwerk-Funktionen bereitgestellt werden können, und eine zweite Teilkonfigurationsbeschreibung für Netzwerk-Funktionen zur Verbindung des wenigstens einen Kommunikations-Containers mit der wenigstens einen betroffenen Hardware-Netzwerkschnittstelle, insbesondere in Form einer Konfigurationsbeschreibung für wenigstens ein virtuelles Netzwerk, beispielsweise Macvlan, und/oder wenigstens eine virtuelle Bridge und/oder wenigstens einen virtuellen Switch, umfasst, und
- die Container-Konfigurationsbeschreibung ausgeführt wird, um das System gemäß dieser zu konfigurieren, wobei der wenigstens eine Kommunikations-Container erzeugt und mit der wenigstens einen betroffenen Hardware-Schnittstelle verbunden wird.

Darüber hinaus wird die Aufgabe durch ein System zur Durchführung des erfindungsgemäßen Verfahrens gelöst. Ein solches System ist bevorzugt derart ausgebildet und/oder eingerichtet,

- dass einem Autokonfigurationsmodul des Systems gemeldet wird, wenn eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle bei Inbetriebnahme des Systems erstmals erfasst wird, und/oder eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle im Betrieb des Systems neu hinzukommt oder wegfällt, wobei zusammen mit einer

solchen Meldung eine Information über die Art der betroffenen Hardware-Netzwerkschnittstelle und/oder die Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes kommuniziert wird,

- 5 - dass das Autokonfigurationsmodul für wenigstens eine der betroffenen Hardware-Netzwerkschnittstellen (2, 3, 4, 5) unter Verwendung einer auf dem System gespeicherten, der Art der oder der jeweils betroffenen Hardware-Netzwerkschnittstelle und/oder der Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes zugeordneten Schablonen-Datei eine Container-Konfigurationsbeschreibung erstellt, die
- 10 eine erste Teilkonfigurationsbeschreibung für wenigstens einen Kommunikations-Container, der wenigstens eine Anwendung umfasst, mittels derer Netzwerk-Funktionen für die oder die jeweils betroffene Schnittstelle bereitgestellt werden können, und eine zweite Teilkonfigurationsbeschreibung für Netzwerk-Funktionen zur Verbindung des wenigstens einen Kommunikations-Containers mit der oder der jeweils
- 15 betroffenen Schnittstelle, insbesondere in Form einer Konfigurationsbeschreibung für wenigstens ein virtuelles Netzwerk, beispielsweise Macvlan, und/oder wenigstens eine virtuelle Bridge und/oder wenigstens einen virtuellen Switch, umfasst, und
- 20 - dass die Container-Konfigurationsbeschreibung ausgeführt wird, um das System gemäß dieser zu konfigurieren, wobei der wenigstens eine Kommunikations-Container erzeugt und mit der wenigstens einen betroffenen Hardware-Schnittstelle verbunden wird.

30

Die vorliegende Erfindung schließt mit anderen Worten die „Lücke“ zwischen industriellen Anwendern und der Container-Technologie, wie beispielsweise Docker. So wird es möglich, die großen Vorteile dieser Technologie für Netzwerkfunktionen nutzbar zu machen, ohne dass es seitens der Anwender eines

35 speziellen IT- bzw. Netzwerks- bzw. Container-Know-Hows bedarf. Dabei besteht der Kerngedanke darin, unter Verwendung eines Autokonfigurationsmoduls Container mit Netzwerkfunktio-

nen, die vorliegend auch als Netzwerkfunktions-Container bzw. Kommunikations-Container bezeichnet werden, dynamisch und automatisch mit ihrer physikalischen Netz-Außenwelt zu verbinden, wofür eine oder auch mehrere Schablonen-Dateien genutzt werden.
5

Das erfindungsgemäß zum Einsatz kommende Autokonfigurationsmodul ermittelt und überwacht den Hardware-Systemausbau um damit die Systemkonfiguration in Bezug auf die verfügbaren
10 Hardware-Netzwerkschnittstellen. Hardware-Netzwerkschnittstellen können dabei sowohl in Form von physikalischen Hardware-Netzwerkschnittstellen als auch in Form von virtuellen Hardware-Netzwerkschnittstellen vorliegen. Bei einer Hardware-Netzwerkschnittstelle im Sinne der vorliegenden Anwendung handelt es sich insbesondere um eine solche, für die wenigstens ein Treiber, insbesondere wenigstens ein Betriebssystem-Treiber vorgesehen ist.
15

Selbstverständlich kann es sein, dass über die Betriebsdauer eines Systems mehrmals eine Hardware-Netzwerkschnittstelle ergänzt und/oder entfernt und/oder bei Inbetriebnahme eines Systems mehrere Hardware-Netzwerkschnittstellen erstmalig erfasst werden. Dann wird bevorzugt mehrmals, gegebenenfalls auch jedes Mal, wenn eine Netzwerkschnittstelle bei Inbetriebnahme des Systems erstmals erfasst wird, und/oder jedes
25 Mal, wenn eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle im Betrieb des Systems neu hinzukommt oder wegfällt, dies dem Autokonfigurationsmodul des Systems zusammen mit einer Information über die Art der jeweils betroffenen Hardware-Netzwerkschnittstelle und/oder die Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes kommuniziert, und mehrmals, gegebenenfalls auch jedes Mal bei einer solchen Meldung unter Verwendung einer bevorzugt auf dem System gespeicherten, der Art der jeweils betroffenen Hardware-
30 Netzwerkschnittstelle und/oder der Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes zugeordneten Schablonen-Datei eine Container-Konfigurationsbeschreibung erstellt und umgesetzt bzw. ausgeführt. Das erfindungsgemäße

System ist hierzu bevorzugt ausgebildet und/oder eingerichtet.

5 Neben der Information darüber, dass eine Änderung hinsichtlich der Schnittstellen stattgefunden hat bzw. eine Schnittstelle bei Inbetriebnahme erstmalig erfasst wird, erhält das Autokonfigurationsmodul auch Informationen über die Art der Schnittstelle. Insbesondere wird kommuniziert, welche Rolle der Schnittstelle zukommt bzw. welchen Typ diese hat. Das er-
10 möglicht beispielsweise die Rolle „upstream“ oder „enterprise-network“ anstatt der u.U. variierenden Netzwerk-Schnittstellen-Namen wie beispielsweise „ens42p66“. Diese Rolle bzw. dieser Typ kann beispielsweise explizit anstatt eines Schnittstellen-Namens in einer Schablone gefordert sein (in
15 einer solchen kann es beispielsweise „role: ...“ anstatt „device: ...“ heißen).

Das Autokonfigurationsmodul zieht dann bevorzugt eine zu der Art der Schnittstelle gehörige Schablonen-Datei heran und er-
20 stellt unter Verwendung dieser eine zu der Art der betroffenen Schnittstelle gehörige Container-Konfigurationsbeschreibung. So kann eine dynamische, Kontext-abhängige Konfiguration des Systems erfolgen.

25 Die Container-Konfigurationsbeschreibung mit deren Teilkonfigurationsbeschreibungen wird im Anschluss an deren Erstellung ausgeführt bzw. umgesetzt und der wenigstens eine Kommunikations-Container wird erzeugt und mit der wenigstens einen betroffenen Hardware-Schnittstelle verbunden. Alternativ zu dem
30 Ausdruck Konfigurationsbeschreibung könnte auch der Ausdruck Konfigurationsanweisung verwendet werden.

Die Ausführung bzw. Umsetzung bedeutet insbesondere bzw. schließt insbesondere mit ein, dass die Container-Konfigurationsbeschreibung bzw. deren Teile von wenigstens einer geeigneten Instanz, die auf dem System implementiert sein kann,
35 verarbeitet wird bzw. werden. Es kann sein, dass der erste Teil von einer Instanz und der zweite Teil von einer anderen

Instanz ausgeführt, insbesondere verarbeitet wird. Eine Container-Konfigurationsbeschreibung kann beispielsweise in Form einer Textdatei bzw. eines Textdokumentes gegeben sein, die insbesondere Anweisungen bzw. Kommandos umfasst, die von einer oder auch mehreren geeigneten Instanzen ausgeführt, insbesondere
5
5 insbesondere verarbeitet werden können.

Dabei umfasst die Container-Konfigurationsbeschreibung wenigstens zwei Teilkonfigurationsbeschreibungen.

10

Konkret eine erste, welche den oder die gewünschten bzw. erforderlichen Kommunikations-Container betrifft. Diese erste Teilkonfigurationsbeschreibung wird zwecks weiterer Verarbeitung in einem Container-System, wie beispielsweise Docker
15 Compose und/oder Kubernetes, erstellt. Mit anderen Worten wird bevorzugt eine erste Teilkonfigurationsbeschreibung erstellt, die ausgestaltet bzw. geeignet ist, um (anschließend) von einem Container-System, wie beispielsweise Docker Compose und/oder Kubernetes, verarbeitet zu werden. Die erste Teil-
20 konfigurationsbeschreibung ist insbesondere ausgestaltet bzw. geeignet, um mittels eines solchen Werkzeuges aus dieser bzw. auf Basis dieser wenigstens einen Kommunikations-Container zu erzeugen. Die erste Teilkonfiguration kann insbesondere wenigstens eine Referenz bzw. wenigstens einen Verweis auf wenigstens einen Container bzw. ein Container-Image umfassen.
25 Es kann auch nur ein Image-Name wenigstens eines Kommunikations-Containers in der ersten Teilkonfigurationsbeschreibung enthalten sein. Die jeweilige Betriebssystemkonfiguration kann dann den oder die Image-Namen noch mit dem Ort (etwa der
30 URL) einer oder mehrerer sogenannter Image-Registries vervollständigen, sobald das Container-Image auf das System geladen werden muss, damit einer oder mehrere Container aus dem Image erzeugt werden bzw. werden können. Eine Referenz kann zum Beispiel in Form einer Pfad- bzw. URL-Angabe vorliegen.

35

Die erste Teilkonfigurationsbeschreibung kann für den Fall, dass Docker Compose als Instanz zu deren Ausführung zum Ein-

satz kommt, insbesondere durch das Kommando "docker-compose <erste Teilkonfigurationsbeschreibung>" ausgeführt werden.

Für den Fall, dass Kubernetes als Instanz zur Ausführung der
5 ersten Teilkonfigurationsbeschreibung zum Einsatz kommt, kann
beispielsweise vorgesehen sein, dass ein "kubelet" auf dem
System, auf dem der Container gestartet werden soll, seiner-
seits ein (Haupt-)CNI-Plugin startet und diesem die erste
10 Teilkonfigurationsbeschreibung zur Ausführung übergibt. Die
erste Teilkonfigurationsbeschreibung liegt insbesondere in
diesem Falle bevorzugt in Form einer oder mehrerer sogenann-
ter Kubernetes-Ressourcen vor bzw. umfasst solche. Bei diesen
handelt es sich um (kleine) Textdokumente, die jeweils für
15 sich eine oder mehrere Teile der Container- und Netzwerkkon-
figuration beschreiben und durch die Kubernetes-Mechanismen
ausgeführt werden. Kubernetes-Ressourcen können von entspre-
chenden Controllern bzw. Plugins, wie den genannten CNI-
Plugins (beispielsweise Multus CNI-Plugins) ausgeführt bzw.
20 verarbeitet werden (in diesem Zusammenhang siehe z.B.
[https://kubernetes.io/docs/concepts/cluster-administra-
tion/manage-deployment/](https://kubernetes.io/docs/concepts/cluster-administration/manage-deployment/) und
[https://engineering.bitnami.com/articles/a-deep-dive-into-
kubernetes-controllers.html](https://engineering.bitnami.com/articles/a-deep-dive-into-kubernetes-controllers.html) und
<https://github.com/intel/multus-cni>). Das Haupt-CNI-Plugin
25 kann beispielsweise durch Multus gegeben sein. Dieses kann
zum Beispiel anhand seiner eigenen Konfiguration sowie der
Teilkonfigurationsbeschreibung entscheiden, welche weiteren
CNI-Plugins ggf. der Reihe nach aufzurufen sind, wie bei-
spielsweise das „bridge“ CNI-Plugin oder das „macvlan“ CNI-
30 Plugin. Das Haupt-CNI-Plugin kontrolliert hierbei die Weiter-
gabe der erforderlichen Teile aus der Teilkonfigurationsbe-
schreibung zu den aufzurufenden CNI-Plugins.

Die zweite Teilkonfigurationsbeschreibung beschreibt die er-
35forderlichen virtuellen Netzwerkfunktionen, wie beispielswei-
se virtuelle, insbesondere Software-(Ethernet-)Bridges, die
für das Verbinden des oder der Kommunikations-Container mit
der bzw. der jeweils betroffenen Schnittstelle erforderlich

sind. Die zweite Teilkonfigurationsbeschreibung betrifft insbesondere virtuelle Netzwerkinfrastruktur eines Betriebssystems des Systems bzw. dient der Einrichtung solcher. Alternativ oder zusätzlich zu virtuellen, insbesondere Software-
5 (Ethernet-)Bridges kann es sich auch um virtuelle „Kabel“ bzw. virtuelle Ethernet-Systeme handeln, insbesondere sogenannte VETHs bzw. VETH devices, und/oder um Macvlans (auch MACVLAN oder MAC VLAN geschrieben), die bevorzugt virtuelle
10 Kabel bis an eine physikalische Hardware-Netzwerkschnittstelle darstellen können.

VETHs steht dabei für Virtual Ethernet Devices, wie sie beispielsweise von Linux hinlänglich vorbekannt sind. VETHs treten in der Regel paarweise auf und fungieren sozuagen als
15 virtuelles Stück Kabel zwischen genau zwei VETH-Netzwerkschnittstellen. Sie verbinden insbesondere eine virtuelle Bridge mit einem Container, können genauso gut aber auch zwei Container direkt miteinander verbinden. Zu VETHs siehe beispielsweise das Linux-Manual ([http://man7.org/linux/man-
20 pages/man4/veth.4.html](http://man7.org/linux/man-pages/man4/veth.4.html)).

Die zweite Teilkonfigurationsbeschreibung betrifft zweckmäßiger Weise Netzwerkfunktionen auf Ebene des Data-Link-Layers (Layer 2), insbesondere wenigstens ein virtuelles Netzwerk
25 auf Layer 2, also wenigstens eine virtuelles Layer-2-Netzwerk.

Es kann auch eine Verdrahtung von Kommunikations-Containern untereinander erforderlich bzw. vorgesehen und entsprechend
30 in der Teilkonfigurationsbeschreibung berücksichtigt sein. Eine solche kann ebenfalls über die vorgenannten Mittel, beispielsweise Soft-Bridges und/oder virtuelle Kabel oder dergleichen erhalten werden.

Anhand der zweiten Teilkonfigurationsbeschreibung sorgt das Autokonfigurationsmodul insbesondere dafür, dass wenigstens
35 ein (virtuelles) internes Netzwerk, beispielsweise wenigstens ein ein internes Netzwerk bildender Software-Switch bzw. we-

nigstens eine Software-Bridge, anstatt mit dem Host-IP-Stack, wie es insbesondere bei Docker üblich ist, allein mit der betroffenen Schnittstelle verknüpft wird. Auch ist es möglich, anstatt einer (expliziten) Software-Bridge bzw. einem (expli-

5 ziten) Software-Switch den oder die Kommunikations-Container direkt mit einer ("Master" bzw. externen) IP-Schnittstelle zu verbinden. Dann übernimmt bzw. "spielt" die IP-Schnittstelle versteckt die Bridge-Funktion.

10 Über die Konfigurationsbeschreibung kann bzw. soll insbesondere IP-Konnektivität für die betroffene Schnittstelle hergestellt werden.

Das Ausführen bzw. Umsetzen der Container-Konfigurationsbe-

15 schreibung schließt zweckmäßiger Weise mit ein, dass der wenigstens eine Kommunikations-Container erzeugt und insbesondere mit der betroffenen Netzwerkschnittstelle, für welche die Container-Konfigurationsbeschreibung erstellt wurde, verbunden wird. Das Ausführen bzw. Umsetzen der Container-Konfi-

20 gurationsbeschreibung kann auch mit einschließen, dass der wenigstens eine Kommunikations-Container gestartet wird.

In besonders bevorzugter Ausgestaltung wird die Container-Konfigurationsbeschreibung erhalten bzw. erstellt, indem ei-

25 ner oder mehrere in der Schablonen-Datei vorgesehene Platzhalter durch wenigstens einen zu der (jeweils) betroffenen Hardware-Netzwerkschnittstelle gehörigen Parameter, insbesondere einen Namen dieser, ersetzt werden. Bevorzugt ist das Autokonfigurationsmodul entsprechend ausgebildet und/oder

30 eingerichtet. Das Ersetzen von einem oder mehreren Platzhaltern in der Schablonen-Datei durch einen Schnittstellen-Parameter, insbesondere Schnittstellen-Namen stellt bevorzugt die einzige Handlung dar, die durchgeführt wird, um aus der zu der betroffenen Schnittstelle gehörigen Schablonen-Datei eine

35 Container-Konfigurationsbeschreibung zu erhalten. Das heißt, die Schablonen-Datei stellt bevorzugt bereits die erforderliche Container-Konfigurationsbeschreibung dar, die jedoch noch um den Kontext, insbesondere die Information der konkret be-

troffenen Schnittstelle zu ergänzen ist, also um die Information, welche physische oder virtuelle Schnittstelle erstmals erfasst bzw. ergänzt wurde.

5 Das Ersetzen von Platzhaltern lässt sich sehr einfach durchführen und ermöglicht es, eine (gemeinsame) Schablone für beliebig viele Netzwerkschnittstellen zu verwenden, so lange sie sich durch die gleiche Art auszeichnen bzw. solange diesen die gleiche Rolle zukommt.

10

Die Information darüber, um welche Art von Schnittstelle es sich bei der (jeweils) betroffenen Schnittstelle handelt, bzw. welche Rolle dieser zukommt, kann dem Autokonfigurationsmodul bevorzugt kommuniziert werden, indem diesem wenigstens ein zu der (jeweils) betroffenen Schnittstelle gehöriger Parameter übermittelt wird, besonders bevorzugt ein zw. der zu der (jeweils) betroffenen Schnittstelle gehöriger Name. Dann werden bevorzugt bestimmte Namensregeln für die Bezeichnung der Schnittstellen verwendet. Zum Beispiel werden etwa
15 allen Schnittstellen der gleichen Art bzw. mit der gleichen Rolle, beispielsweise allen Upstream-Schnittstellen, also Schnittstellen, die mit einem Upstream-Netzwerk verbunden oder zu verbinden sind, Namen gegeben, die mit den gleichen Anfangszeichen, etwa Anfangsbuchstaben beginnen, etwa alle
20 mit „ens*“, wobei „*“ bedeutet, dass weitere Zeichen folgen, über welche die Schnittstelle von anderen Schnittstellen unterscheidbar ist.

Aus der Schablonen-Datei werden von dem Autokonfigurationsmodul insbesondere für die jeweils betroffene Schnittstelle,
30 also kontextabhängig, die Netzwerke zwischen der jeweils betroffenen Schnittstelle und dem oder den Kommunikations-Containern in einem (einzelnen) Kontext erzeugt bzw., für den Fall, dass eine Schnittstelle wegfällt, wieder entfernt.

35

Es sei angemerkt, dass es erforderlich sein kann, jedoch nicht erforderlich sein muss, dass für jede virtuelle und/oder physikalische Hardware-Netzwerkschnittstelle eines

Systems, die bei Inbetriebnahme des Systems erstmals erfasst wird bzw. im Betrieb des Systems neu hinzukommt, wenigstens ein Kommunikations-Container eingerichtet bzw. erhalten wird.

5 Sofern wenigstens eine virtuelle und/oder physikalische Hardware-Netzwerkschnittstelle bei Inbetriebnahme des Systems
erstmals erfasst wird bzw. im Betrieb des Systems neu hinzu-
kommt, für welche dies nicht erforderlich bzw. gewünscht ist,
kann vorgesehen sein, dass für diese von dem Autokonfigurati-
10 onsmodul eine Konfigurationsbeschreibung erstellt wird, wel-
che keinen bzw. keine Kommunikations-Container betrifft. Die-
se Art der Konfigurationsbeschreibung, die zur Abgrenzung ge-
genüber Container-Konfigurationsbeschreibungen, die wenig-
stens einen Container betreffen, vorliegend als Netzwerk-Kon-
15 figurationsbeschreibung bezeichnet wird, betrifft insbesonde-
re nur Netzwerkfunktionen für die bzw. die jeweilige be-
troffene Schnittstelle.

Eine weitere Ausführungsform zeichnet sich entsprechend da-
20 durch aus, dass das Autokonfigurationsmodul für wenigstens
eine weitere der betroffenen Hardware-Netzwerkschnittstellen
eine Netzwerk-Konfigurationsbeschreibung erstellt, die we-
nigstens ein virtuelles Netzwerk und/oder wenigstens eine
virtuelle Bridge und/oder wenigstens einen virtuellen Switch
25 für die wenigstens eine weitere Hardware-Netzwerkschnittstel-
le und keinen Kommunikations-Container für diese betrifft.
Das erfindungsgemäße System, insbesondere dessen Autokonfigu-
rationsmodul ist vorteilhafter Weiterbildung entsprechend
ausgestaltet und/oder eingerichtet.

30

Die bzw. die jeweilige Netzwerk-Konfigurationsbeschreibung
betrifft insbesondere wenigstens ein virtuelles Netzwerk,
welches an die (jeweils) betroffene Netzwerkschnittstelle an-
zuschließen ist bzw. angeschlossen wird. Die bzw. die jwei-
35 lige Netzwerk-Konfigurationsbeschreibung betrifft mit anderen
Worten insbesondere ein "internes Netzwerk" bzw. eine "inter-
ne Verdrahtung" für die jeweilige Schnittstelle, über welche

diese beispielsweise mit internen Netzwerken bzw. Containern, anderer Schnittstellen verbunden werden kann bzw. wird.

Die bzw. die jeweilige Netzwerk-Konfigurationsbeschreibung
5 wird - völlig analog zu der bzw. den Container-Konfigurationsbeschreibungen - zweckmäßiger Weise anschließend ausgeführt, um das das System gemäß dieser zu konfigurieren.

Die Netzwerk-Konfigurationsbeschreibung kann - ebenfalls in
10 Analogie zu der Container-Konfigurationsbeschreibung - beispielsweise in Form einer Textdatei bzw. eines Textdokumentes gegeben sein, die insbesondere Anweisungen bzw. Kommandos umfasst, die von einer oder auch mehreren geeigneten Instanzen ausgeführt, insbesondere verarbeitet werden können. Sie kann
15 in ihrer Struktur der Container-Konfigurationsbeschreibung, insbesondere deren zweiter Teilkonfigurationsbeschreibung - ähneln oder gleichen. Die vorstehend im Zusammenhang mit der Container-Konfigurationsbeschreibung erläuterten bevorzugten Merkmale können einzeln oder in Kombination auch bei der
20 Netzwerk-Konfigurationsbeschreibung gegeben sein, mit dem insbesondere einzigen Unterschied, dass diese nicht der Erzeugung von Containern dient.

Hinsichtlich der bzw. der jeweiligen Netzwerk-Konfigurations-
25 beschreibung gilt u.a. bevorzugt, dass sie - ebenfalls völlig analog zu der bzw. den Container-Konfigurationsbeschreibungen - unter Verwendung einer Schablonen-Datei von dem Autokonfigurationsmodul erstellt wird. Alternativ oder zusätzlich gilt hinsichtlich der bzw. der jeweiligen Netzwerk-Konfigurations-
30 beschreibung, dass sie bevorzugt der Verarbeitung in einem Container-System bzw. Container-Werkzeug, wie etwa Docker Compose, dient. Sie ist insbesondere ausgestaltet, um von einem solchen ausgeführt zu werden.

35 Es sei angemerkt, dass es möglich ist, dass das Autokonfigurationsmodul für mehrere ggf. auch alle Netzwerkschnittstellen, für die keine Container-Konfigurationsbeschreibungen erstellt werden, Netzwerk-Konfigurationsbeschreibungen er-

stellt. Erforderlich ist dies jedoch nicht. Konfigurationsbeschreibungen für Schnittstellen können prinzipiell auch auf andere Weise erhalten werden.

5 Das Autokonfigurationsmodul kann zwar, muss jedoch nicht für alle Netzwerkschnittstellen eines Systems, die bei Inbetriebnahme erstmalig erfasst werden bzw. im Betrieb des Systems neu hinzukommen oder wegfallen, zum Einsatz kommen. Die erfindungsgemäße Vorgehensweise kann vielmehr auch in Kombination mit der konventionellen, vorbekannten Vorgehensweise, insbesondere in Kombination mit einer (teilweise) manuellen Konfiguration- etwa durch einen Anwender -, zum Einsatz kommen. So kann vorgesehen sein, dass für wenigstens eine Schnittstelle alternativ oder zusätzlich dazu, dass das Autokonfigurationsmodul für diese zum Einsatz kommt, wenigstens eine Konfigurationsbeschreibung auf anderem Wege, etwa manuell erstellt wird bzw. worden ist.

20 Eine konventionelle, insbesondere (teilweise) manuelle Konfiguration kann natürlich auch für etwaig erforderliche andere, nicht durch das Autokonfigurationsmodul abgedeckte Aspekte vorgesehen sein, sowohl ergänzend für Schnittstellen, für welche das Autokonfigurationsmodul tätig wird bzw. bereits wurde, als auch für Schnittstellen, für welche dies nicht der Fall ist.

30 Insbesondere für den Fall, dass das Autokonfigurationsmodul nur für eine oder nur für einige Hardware-Netzwerkschnittstellen, die bei Inbetriebnahme erfasst werden bzw. später hinzukommen, zum Einsatz kommt, kann für wenigstens eine, bevorzugt sämtliche der verbleibenden Netzwerkschnittstellen jeweils eine (Netzwerk-)Konfigurationsbeschreibung, die wenigstens ein virtuelles Netzwerk, beispielsweise Macvlan, und/oder wenigstens eine virtuelle Bridge und/oder wenigstens einen virtuellen Switch für diese betrifft, erstellt werden bzw. worden sein.

In bevorzugter Ausgestaltung gilt, dass sämtliche erforderliche "interne Verdrahtung", die nicht durch das Autokonfigurationsmodul abgedeckt ist bzw. wird, auf konventionelle, vorbekannte Weise, insbesondere durch manuelle Konfiguration, erhalten wird.

Im Zusammenhang mit Macvlans sei beispielhaft auf <https://docs.docker.com/network/macvlan/> verwiesen.

Das Autokonfigurationsmodul kann als Software implementiert sein. Die Software kann dann z.B. auf allgemeiner, ohnehin für andere Zwecke vorhandener Hardware des Systems laufen, so dass für die Umsetzung der vorliegenden Erfindung kein zusätzlicher beziehungsweise spezieller Hardware-Ausbau erforderlich ist. Selbstverständlich ist es aber auch möglich, dass für das Autokonfigurationsmodul - ggf. zusätzlich zu Software dieses - spezielle, zusätzliche Hardware vorgesehen ist. Das Autokonfigurationsmodul kann auch rein Hardwareimplementiert sein.

Bei dem System kann es sich beispielsweise um einen Industrie-PC oder eine insbesondere speicherprogrammierbare Steuerung handeln. Das System kann Teil einer industriellen Automatisierungsanlage sein. Prinzipiell kann die erfindungsgemäße Autokonfiguration für jede Art von System aus dem industriellen Bereich, insbesondere der industriellen Automatisierung vorgesehen sein, welches in seiner Grundausstattung, bei Inbetriebnahme, eine oder mehrere (physikalische und/oder virtuelle) Hardware-Netzwerkschnittstellen aufweist und/oder dem im Betrieb eine oder mehrere Hardware-Netzwerkschnittstellen hinzugefügt und/oder von diesem entfernt werden.

Die erfindungsgemäße Vorgehensweise bietet mehrere große Vorteile. Einerseits müssen industrielle Anwender kein IT-Fachwissen für das Anwenden von containerisierten Netzwerkfunktionen, beispielsweise weder über Container, etwa Dock-Container, noch über zugehörige Werkzeuge, etwa Docker Network Driver, noch über Software-Switches haben, und können dennoch

die Vorteile dieser Technologie für eine automatische Konfiguration im Falle einer Änderung bzw. Ergänzung des Hardware-Schnittstellenausbaus ihrer Systeme bzw. für deren Inbetriebnahme nutzen. Es wird daher besonders einfach für Anwender,
5 Systeme aus dem Bereich der Automatisierungstechnik um Netzwerkzu- oder Abgänge zu erweitern bzw. solche zu entfernen. Eine manuelle Konfiguration mittels zusätzlichem Engineering- bzw. Konfigurations-Werkzeug(en) ist nicht mehr nötig. Es genügt das Anstecken der entsprechenden Netzwerk-Schnittstellen-
10 Hardware, an welches sich dann eine vollständig automatische Konfiguration anschließt.

In besonders bevorzugter Ausgestaltung ist vorgesehen, dass die Schablonen-Datei aus einer Mehrzahl von insbesondere auf
15 dem System gespeicherten unterschiedlichen Schablonen-Dateien ausgewählt wird. Unter einer Mehrzahl sind dabei zwei, drei, vier oder auch mehr Schablonen-Dateien zu verstehen. Die Auswahl aus einer Mehrzahl von Schablonen-Dateien erfolgt bevorzugt automatisiert anhand von in den Schablonen-Dateien ent-
20 haltenen Filterregeln. Das erfindungsgemäße System, insbesondere das Autokonfigurationsmodul dieses ist bevorzugt entsprechend ausgebildet und/eingerichtet.

Sind mehrere Schablonen-Dateien verfügbar, aus denen ausge-
25 wählt werden kann, kann besonders einfach auf eine erstmalige Erfassung und/oder das neue Hinzufügen von Schnittstellen unterschiedlicher Art reagiert werden.

Wird auf Filterregeln zurückgegriffen, können diese bei-
30 spielsweise einen der oder der der jeweils betroffenen Schnittstelle zugeordneten Namen und/oder die Art der oder der jeweils betroffenen Schnittstelle betreffen. Schnittstellen ist bzw. wird in der Regel ein eindeutiger Name zugeordnet, insbesondere von dem bzw. über das Betriebssystem, und
35 über den Namen ist es auf einfache Weise möglich, zwischen verschiedenen Arten von Schnittstellen zu unterscheiden. Beispielsweise kann vorgesehen sein, dass jeweils die ersten Zeichen, zum Beispiel Buchstaben, von Netzwerkschnittstellen

der gleichen Art bzw. mit gleicher Rolle übereinstimmen und die konkrete Individualisierung über nachfolgende Zeichen, etwa Buchstaben, erfolgt.

5 Weiterhin bevorzugt ist vorgesehen, dass für mehrere, bevorzugt für jede Art von physikalischer und/oder virtueller Hardware-Netzwerkschnittstelle, welche das System aufweisen kann, jeweils wenigstens eine, bevorzugt jeweils genau eine Schablonen-Datei insbesondere auf dem System gespeichert ist, und/oder dass für zumindest drei verschiedene Arten von physikalischen und/oder virtuellen Hardware-Netzwerkschnittstellen jeweils wenigstens eine, bevorzugt jeweils genau eine Schablonen-Datei insbesondere auf dem System gespeichert ist, wobei es sich bei den drei verschiedenen Arten von physikalischen und/oder virtuellen Hardware-Netzwerkschnittstellen besonders bevorzugt um Upstream-Schnittstellen, die insbesondere mit einem Up-stream-, etwa Firmen-Netzwerk eines Anwenders zu verbinden oder verbunden sind, und Downstream-Schnittstellen, die mit einem unterlagerten Netzwerk, insbesondere Zellen- oder Produktionsnetz, zu verbinden oder verbunden sind, und WLAN-Schnittstellen, die mit einem Drahtlos-Netzwerk zu verbinden oder verbunden sind, handelt.

Im industriellen Umfeld sind oftmals folgende drei Arten von Schnittstellen möglich:

- Upstream/Infrastruktur Netzwerkschnittstellen
Bei diesen handelt es sich insbesondere um kabelgebundene Schnittstellen (sogenannte "wired-interfaces"), die in der Regel unterlagerte Netzsegmente, etwa Feldbus-Netzsegmente, beispielsweise an eine IT-Infrastruktur, etwa ein Firmennetzwerk, anbinden.
- Wireless Netzwerkschnittstellen:
Diese Schnittstellen stellen in der Regel automatisiert lokale WLAN-Zugänge insbesondere ausschließlich zu den unterlagerten Feldbus-Netzsegmenten bereit. Damit wird es beispielsweise möglich, mobile Android- und/oder iOS-Gerä-

te auch in drahtgebundenen Automatisierungsnetzen verwenden zu können.

- Downstream Netzwerkschnittstellen:

5 Diese Schnittstellen führen in der Regel in unterlagerte Feldbus-Netzsegmente, die beispielsweise der Kommunikation zwischen Feldgeräten und einer Steuerung bzw. einem Industrie-PC einer Automatisierungsanlage dienen, an denen
10 entsprechend einzelne Maschinen, Aggregate und Automatisierungsgeräte, insbesondere einer industriellen Automatisierungsanlage, angeschlossen sind. In diese „downstream“ Netzsegmente werden bevorzugter Weise Netzwerkfunktionen von Nat64-Routern und/oder IPV6-Routern und insbesondere
15 Inbetriebsetzungs-Werkzeuge wie das sogenannte Primary Setup Tool (PST), etwa für IP-basierte Feldgeräte, automatisch bereitgestellt, sobald eine Netzwerkschnittstelle dieser Art erkannt wird und in Betrieb geht.

Sind für diese drei Arten von Schnittstellen bzw. diese drei
20 Rollen von Schnittstellen Schablonen-Dateien zum Erhalt einer zugehörigen Container-Konfigurationsbeschreibung verfügbar, können die meisten Fälle von Schnittstellen-Änderungen bzw. -Erfassungen im industriellen Bereich abgedeckt werden.

25 Eine weitere bevorzugte Ausführungsform zeichnet sich dadurch aus, dass dem Autokonfigurationsmodul das erstmalige Erfassen einer physikalischen oder virtuellen Hardware-Netzwerkschnittstelle bei Inbetriebnahme des Systems und/oder das
30 neue Hinzukommen und/oder Wegfallen einer physikalischen oder virtuellen Hardware-Netzwerkschnittstelle im Betrieb des Systems kommuniziert wird, indem ein auf dem System laufendes Betriebssystem dem Autokonfigurationsmodul ein entsprechendes Ereignis meldet, insbesondere als Plug-and-Play-Ereignis. Auf
35 diese Weise kann das Autokonfigurationsmodul auf einfachem Wege über den Hardware-Schnittstellen-Ausbau bzw. Änderungen dieses informiert werden. Das erfindungsgemäße System ist in Weiterbildung entsprechend ausgebildet und/oder eingerichtet.

Was die Ausführung bzw. Umsetzung der Konfigurationsbeschreibung und den Erhalt einer entsprechenden Konfiguration des Systems angeht, kann vorgesehen sein, dass im Rahmen der Ausführung bzw. Umsetzung der ersten Teilkonfigurationsbeschreibung der wenigstens eine Kommunikations-Container erzeugt wird. Das erfindungsgemäße System ist bevorzugt entsprechend ausgebildet und/oder eingerichtet. Die Ausführung bzw. Umsetzung der ersten Teilkonfigurationsbeschreibung kann mittels eines Container-Controller-Moduls des Systems erfolgen.

10

Das Container-Controller-Modul ist besonders bevorzugt als Software implementiert, kann aber, genau wie das Autokonfigurationsmodul prinzipiell auch spezielle, zusätzliche Hardware umfassen bzw. auch rein hardware-implementiert sein.

15

Die Ausgestaltung eines Container-Controller-Moduls wird in der Regel davon abhängen, welche Container-Technologie bzw. welche Container-Technologien zum Einsatz kommen. Das Container-Controller-Modul kann beispielsweise ein Docker Composer Werkzeug und/oder einen Kubernetes Pod Scheduler Controller umfassen oder dadurch gegeben sein. Zu Kubernetes siehe beispielsweise <https://kubernetes.io/docs/reference/command-line-tools-reference/kube-scheduler/>. Wird alternativ oder zusätzlich zu Docker und/oder Kubernetes auf andere bzw. weitere Container-Technologie(n) zurückgegriffen, kann das Container-Controller-Modul zweckmäßigerweise zu dieser bzw. diesen Technologien gehörige Werkzeuge, mittels derer Container aus einer zugehörigen Container-Konfigurationsbeschreibung erzeugt werden können, umfassen bzw. dadurch gegeben sein.

30

Eine weitere vorteilhafte Ausführungsform des erfindungsgemäßen Verfahrens zeichnet sich dadurch aus, dass die zweite Teilkonfigurationsbeschreibung von einem Netzwerk-Controller-Modul des Systems umgesetzt wird. Das erfindungsgemäße System ist bevorzugt entsprechend ausgebildet und/oder eingerichtet. Die zweite Teilkonfigurationsbeschreibung kann beispielsweise in Form eines Shell-Skriptes vorliegen, und/oder als sogenannte „systemd units“, und/oder als Clustersystem-Objekte

35

- (insbesondere Kubernetes API-objects) und/oder dergleichen mehr ausgeführt sein. Zu "systemd units" siehe beispielsweise <https://en.wikipedia.org/wiki/Systemd> und zu Kubernetes <https://en.wikipedia.org/wiki/Kubernetes>. Das Netzwerk-Controller-Modul, welches bevorzugt als Software implementiert ist, kann dann entsprechend einen Shell-Skript-Interpreter und/oder systemd und/oder einen Kubernetes-Controller umfassen oder dadurch gebildet werden.
- 10 Ein Kommunikations-Container ist bevorzugt ein Container, dessen Software-Inhalt Netzwerk-, insbesondere Netzwerk-Kommunikationsfunktionen bereitstellt, besonders bevorzugt auf den ISO/OSI-Schichten 2 und/oder 3 und/oder 4 und/oder 7. Rein Beispielhaft sei für einen Kommunikations-Container das
- 15 Umsetzen von IP-Verkehr zwischen den Protokollversionen 4 und 6 genannt. Die im Container enthaltene Software realisiert bevorzugt eine eigene Control Plane (in diesem Zusammenhang sei rein beispielhaft auf die EP 3 062 490 A1 der Siemens AG verwiesen) und eine Data Plane (in diesem Zusammenhang sei
- 20 rein beispielhaft auf WrapSix NAT64 data plane in user space, <http://wrapsix.org>; Tayga NAT64 in user space, <http://www.litech.org/tayga/> verwiesen). Alternativ dazu ist es auch möglich, dass eine Kommunikations-Container nur eine eigene Control Plane implementiert, während als Data Plane
- 25 die Data Plane des Betriebssystem-Kerns benutzt wird, wie zum Beispiel im Fall eines autokonfigurierenden IPv6-Routers, wie in der früheren europäischen Patentanmeldung 18192748.4 der Siemens AG offenbart.
- 30 Besonders bevorzugt ist weiterhin vorgesehen, dass wenigstens eine Anwendung des wenigstens einen Kommunikations-Containers gemäß der ersten Teilkonfigurationsbeschreibung Funktionen eines insbesondere automatisch konfigurierenden IPv6-
- 35 und/oder NAT64-Routers und/oder eines Namensdienst-, insbesondere DNS-Servers und/oder eines Brouters bereitstellt und/oder durch eine Software für einen WLAN-Access-Point gegeben ist oder eine solche umfasst.

Bei Netzwerkfunktionen des wenigstens einen Kommunikations-Containers, also Netzwerkfunktionen gemäß der ersten Teilkonfigurationsbeschreibung, handelt es sich insbesondere um solche, die Konnektivität der betroffenen Hardware-Netzwerk-
5 schnittstelle mit wenigstens einer weiteren Hardware-Netzwerkschnittstelle bzw. einem externen Netzwerk gewährleisten bzw. herstellen, ggf. mit Netztrennung.

Bei Netzwerkfunktionen des wenigstens einen Kommunikations-Containers kann es sich beispielsweise um folgende handeln:
10

- autokonfigurierende IPv6-Router bevorzugt samt autokonfigurierendem und/oder „autoiterativem“ DNS-Server, wie in den ebenfalls auf die Anmelderin zurückgehenden europäischen Patenanmeldungen des Aktenzeichens Nr. 18187459.5
15 und 18187456.1 sowie 17194062.0 beschrieben.
- autokonfigurierende NAT64-Router insbesondere inklusive Profinet-Systeme-Scanner wie in den ebenfalls auf die Anmelderin zurückgehenden EP 3 062 490 B1 beschrieben.
20

Welche Netzwerkfunktionen konkret über den einen oder die mehreren Kommunikations-Container bereitgestellt werden bzw. bereitzustellen sind, wird dabei von der Art der betroffenen
25 Schnittstelle abhängen. Wie vorstehende bereits angemerkt, sind für Downstream-Netzwerkschnittstellen in der Regel insbesondere IPv6- und/oder NAT64-Funktionen gewünscht bzw. erforderlich. Wird eine Wireless-Network-Schnittstelle hinzugefügt, kann erforderliche bzw. zugehörige Software containerisiert bereitgestellt werden.
30

Zusätzlich zu insbesondere IP-Konnektivität bereitstellenden bzw. gewährleistenden Netzwerkfunktionen können weitere Funktionen bzw. Dienste über einen oder mehrere Container bereit-
35 gestellt werden, etwa Inbetriebsetzungs-Werkzeuge, wie das sogenannte Primary Setup Tool (PST). Über solche können z.B. (IP-basierte) Teilnehmer eines an die betroffene Netzwerkschnittstelle angeschlossenen (Teil-)Netzwerkes, beispiels-

weise an ein Feldbussegment angeschlossene (IP-basierte) Feldgeräte, automatisch In Betrieb gesetzt werden. Die erste Teilkonfigurationsbeschreibung kann entsprechend auch Angaben zu wenigstens einem Container enthalten, der wenigstens eine Anwendung umfasst, mittels derer neben Netzwerkfunktionen andere bzw. weitere Funktionen bzw. Dienste bereitgestellt werden können, insbesondere Inbetriebsetzungsfunktionen bzw. -dienste.

10 Besonders bevorzugt kann weiterhin vorgesehen sein, dass für den Fall, dass eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle im Betrieb des Systems wegfällt, weil sie beispielsweise von einem Anwender entfernt und/oder deaktiviert wird, zugehörige Netzwerk-Funktionen deaktiviert
15 und/oder entfernt und/oder Kommunikations-Container gestoppt werden. Dies wird bevorzugt von dem Autokonfigurationsmodul veranlasst. Zugehörige Netzwerkfunktionen sind insbesondere solche, die für die betroffene Hardware-Netzwerkschnittstelle eingerichtet waren und über die besonders bevorzugt Konnektivität der betroffenen Hardware-Netzwerkschnittstelle mit wenigstens einer weiteren Hardware-Netzwerkschnittstelle bzw. einem externen Netzwerk gewährleistet wurde, ggf. mit Netz-
20 trennung. Das erfindungsgemäße System ist bevorzugt entsprechend ausgebildet und/oder eingerichtet.

25

Bei den Netzwerkfunktionen und/oder Kommunikations-Containern, die deaktiviert und/oder entfernt bzw. gestoppt werden, handelt es sich besonders bevorzugt um die diejenigen Netzwerkfunktionen bzw. Kommunikations-Container, die im Falle der erstmaligen Erfassung einer Hardware-Netzwerkschnittstelle und/oder der Ergänzung einer solchen erfindungsgemäß unter Nutzung einer Schablonen-Datei hinzukommen.

30

Hierfür kann die zu verwendende Schablonen-Datei, die von der Art der weggefallenen Netzwerk-Schnittstelle abhängt, beispielsweise Lösch- und/oder Stoppanweisungen enthalten, insbesondere Löschanweisungen bzgl. Netzwerkfunktionen zur Verbindung wenigstens eines Kommunikations-Containers mit der
35

betroffenen, weggefallenen Hardware-Netzwerkschnittstelle, beispielsweise in Form wenigstens einer virtuellen Bridge, und/oder wenigstens eines virtuellen Switches für diese Verbindung. Das erfindungsgemäße System ist bevorzugt entsprechend ausgebildet und/oder eingerichtet.

Bei der Schablonen-Datei, die im Falle des Wegfallens einer Hardware-Netzwerkschnittstelle verwendet wird, handelt es sich besonders bevorzugt um die gleiche Schablonen-Datei, die im Falle einer erstmaligen Erfassung und/oder der Ergänzung einer Hardware-Schnittstelle gleicher Art bzw. mit gleicher Rolle zur Verwendung kommt. Dann sind bevorzugt zusätzlich Lösch- und/oder Stoppanweisungen in dieser enthalten. Es sei angemerkt, dass insbesondere Kommunikations-Container betreffende Stoppanweisungen, die etwa von einem Container-Werkzeug zu interpretieren sind, auch separat von der Schablonen-Datei übermittelt werden können, etwa auf direktem oder indirektem Wege von dem Autokonfigurationsmodul an ein Container-Control-Modul.

Wird für die erstmalige Erfassung bzw. für das Hinzukommen von Schnittstellen die gleiche Schablonen-Datei verwendet, wie im Falle des Wegfallens von Schnittstellen der gleichen Art, ist bevorzugt für jede Schnittstellen-Art bzw. Schnittstellen-Rolle genau eine Schablonen-Datei vorgesehen, insbesondere auf dem System gespeichert, was eine besonders „schlanke“ Lösung für eine Autokonfiguration und ein automatisches Löschen darstellt.

Ein weiterer Gegenstand der vorliegenden Erfindung ist ein Computerprogramm umfassend Programmcode-Mittel zur Durchführung des Verfahrens.

Darüber hinaus betrifft die Erfindung ein computerlesbares Medium, das Instruktionen umfasst, die, wenn sie auf wenigstens einem Computer ausgeführt werden, den wenigstens einen Computer veranlassen, die Schritte des erfindungsgemäßen Ver-

fahrens durchzuführen. Ein Rechner kann Teil eines erfindungsgemäßen Systems sein.

Bei dem computerlesbaren Medium kann es sich beispielsweise
5 um eine CD-ROM oder DVD oder einen USB oder Flash Speicher handeln. Es sei angemerkt, dass unter einem computerlesbaren Medium nicht ausschließlich ein körperliches Medium zu verstehen sein soll, sondern ein solches beispielsweise auch in Form eines Datenstromes und/oder eines Signals, welches einen
10 Datenstrom repräsentiert, vorliegen kann.

Weitere Merkmale und Vorteile der vorliegenden Erfindung werden anhand der nachfolgenden Beschreibung einer Ausführungsform des Verfahrens sowie des Systems der vorliegenden Erfindung unter Bezugnahme auf die beiliegende Zeichnung deutlich.
15

Die einzige Figur zeigt eine rein schematische Darstellung eines erfindungsgemäßen Systems.

20 Dargestellt ist ein Ausführungsbeispiel eines erfindungsgemäßen Automatisierungssystems 1 einer in den Figuren nicht weiter dargestellten industriellen Automatisierungsanlage in rein schematischer Darstellung.

25 Bei dem Automatisierungssystem 1 handelt es sich vorliegend um eine speicherprogrammierbare Steuerung (SPS). Es sei betont, dass es sich alternativ dazu auch um eine andere Art von System handeln kann. Die erfindungsgemäße Autokonfiguration kann prinzipiell für jede Art von System aus dem industriellen Bereich, insbesondere der industriellen Automatisierung vorgesehen sein, welches in seiner Grundausstattung, bei
30 Inbetriebnahme, eine oder mehrere Hardware-Netzwerkschnittstellen aufweist und/oder dem im Betrieb eine oder mehrere Hardware-Netzwerkschnittstellen hinzugefügt und/oder von diesem entfernt werden (können).
35

Das System 1 weist vorliegend vier Hardware-Netzwerkschnittstellen 2, 3, 4, 5 auf. Alle vier Schnittstellen sind bei dem

hier dargestellten Ausführungsbeispiel physikalische Hardware-Netzwerkschnittstellen 2, 3, 4, 5. Alternativ dazu kann es sich bei einer, bei mehreren oder auch allen diesen Schnittstellen um virtuelle Hardware-Netzwerkschnittstellen, 5 beispielsweise einer oder mehrerer virtuellen Maschinen (VMs) handeln.

Für jede der vier Schnittstellen 2, 3, 4, 5 umfasst ein Betriebssystem des Automatisierungssystems 1, welches in der 10 Figur nicht dargestellt ist, jeweils wenigstens einen Treiber.

Die Schnittstellen 2, 3, 4, 5 sind an verschiedene Netzwerke 6, 7, 8, 9 angeschlossen.

15

Konkret ist die Schnittstelle 2, bei welcher es sich um eine Upstream/Infrastruktur Netzwerkschnittstelle handelt, an ein Firmen- oder Fabriknetzwerk 6 angeschlossen, welches in der Figur rein schematisch durch eine Wolke angedeutet ist. Die 20 Schnittstelle 3 ist an ein (lokales) Drahtlos-Netzwerk 7 angeschlossen und stellt somit eine Drahtlos-Netzwerk-Schnittstelle dar, und die beiden Schnittstellen 4, 5 sind jeweils an ein unterlagertes Maschinen- oder Zellennetzwerk 8, 9 angeschlossen. Die Schnittstellen 4, 5 binden vorliegend unterlagerte Feldbus-Netzsegmente an das Firmennetzwerk 6 an. Sie 25 stellen Downstream-Netzwerk-Schnittstellen dar.

Zur besseren Unterscheidung sind in der Figur die drei verschiedenen Netzwerk-Arten und zugehörigen Elemente mit verschiedenen Linien-Arten dargestellt, konkret das Firmennetzwerk 6 und zugehörige Elemente mit punktierten Linien, das 30 Drahtlos-Netzwerk 7 und zugehörige Elemente mit gestrichelten Linien und die unterlagerten Maschinen- oder Zellennetzwerke 8 und zugehörigen Elemente mit strichpunktierten Linien.

35

Von Seiten der Anwender ist es erstrebenswert bzw. gewünscht, dass möglichst rasch eine durchgehende Konnektivität zwischen dem Firmen- und Produktionsnetz 6, den unterlagerten Zellen-

bzw. Maschinennetzen 8, 9 und dem Drahtlos-Netzwerk 7 hergestellt wird, und zwar entsprechend des anwenderseitig vorgenommenen und/oder geänderten Hardware-Ausbaus an Schnittstellen 2, 3, 4, 5.

5

Um dies zu gewährleisten, ist das System 1 ausgebildet und eingerichtet, um sowohl bei Inbetriebnahme als auch im laufenden Betrieb zu überwachen, welche physikalischen und/oder virtuellen Hardware-Netzwerkschnittstellen 2, 3, 4, 5 vorhanden sind. Die Überwachung erfolgt vorliegend über ein auf dem System 1 installiertes Betriebssystem.

10

Das System 1 umfasst weiterhin ein Autokonfigurationsmodul 10.

15

Für den Fall, dass eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle 2, 3, 4, 5 bei Inbetriebnahme des Systems 1 erstmals erfasst wird, oder eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle 2, 3, 4, 5 im laufenden Betrieb des Systems 1 neu hinzukommt oder wegfällt, wird dies dem Autokonfigurationsmodul 10 kommuniziert und zwar zusammen mit einer Information über die Art der betroffenen Hardware-Netzwerkschnittstelle 2, 3, 4, 5 und/oder die Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes 6, 7, 8, 9. Die Kommunikation erfolgt vorliegend, indem das Betriebssystem dem Autokonfigurationsmodul 10 ein entsprechendes Ereignis als Plug-and-Play-Ereignis meldet. Die Meldung umfasst die Information, um welche Art von Ereignis es sich handelt, also ob eine Schnittstelle erstmals erfasst wurde oder neu hinzugekommen oder weggefallen ist. Weiterhin wird dem Autokonfigurationsmodul 10 ein Name der jeweils betroffenen Schnittstelle 2, 3, 4, 5 mitgeteilt. Die Schnittstellennamen sind derart gewählt, dass von diesen ableitbar ist, um welche Art von Schnittstelle 2, 3, 4, 5 es sich handelt, bzw. welche Rolle dieser zukommt. Bei Bedarf können neben dem Schnittstellennamen auch weitere Informationen hinzugefügt werden, beispielsweise der Busanschluss (ins-

20

25

30

35

besondere Hardware-Anschluss am PCI-Bus, USB-Bus und dergleichen), MAC-Adressen und dergleichen mehr.

In der Figur ist ein solches Hardware- bzw. Schnittstellen-
5 Ereignis, also das erstmalige Erfassen oder neue Hinzukommen
oder Wegfallen einer Schnittstelle 2, 3, 4, 5, jeweils rein
schematisch mit einem Blitz angedeutet, welcher mit der Be-
zugsziffer 11 versehen ist.

10 Bevorzugt jedes Mal, wenn eine Meldung über ein solches Hard-
ware-Ereignis 11 bei dem Autokonfigurationsmodul 10 eingeht,
wählt dieses in Reaktion darauf aus einer Mehrzahl von auf
dem System gespeicherten Schablonen-Dateien 12, 13, 14 eine
Schablonen-Datei 12, 13, 14 für den jeweiligen Kontext 15,
15 16, 17 aus.

Vorliegend sind drei Schablonen-Dateien 12, 13, 14 im System
gespeichert. Eine für jede Schnittstellen-Art bzw. -Rolle,
also eine Schablonen-Datei 12 für ein Hardware-Ereignis 11,
20 welches eine Downstream-Schnittstelle 4, 5, betrifft, eine
Schablonen-Datei 13 für ein Hardware-Ereignis 11, welches ei-
ne Drahtlos-Netzwerk-Schnittstelle 3 betrifft, und eine
Schablonen-Datei 14 für ein Hardware-Ereignis 11, welches ei-
ne Upstream-Schnittstelle 2, betrifft. Es versteht sich, dass
25 weitere Schablonen-Dateien für weitere Schnittstellen-Arten
bzw. Schnittstellen-Rollen hinterlegt sein können. Bevorzugt
ist für jede Art bzw. jede Rolle von Schnittstelle 2, 3, 4,
5, die an einem gegebenen System 1 auftreten kann, wenigstens
eine, insbesondere genau eine Schablonen-Datei 12, 13, 14 auf
30 dem System 1 hinterlegt, so dass jeder mögliche Hardware-
Ausbau bzw. jede mögliche Änderung dieses abgedeckt ist.

Die Auswahl einer Schablonen-Datei 12 aus der Mehrzahl von
Schablonen-Dateien 12, 13, 14 für den jeweiligen Kontext 15,
35 16, 17 durch das Autokonfigurationsmodul 10 erfolgt anhand
von in den Schablonen-Dateien 12, 13, 14 hinterlegten Filter-
regeln. Vorliegend betreffen die Filterregeln Namen, die den
Schnittstellen 2, 3, 4, 5 zugeordnet sind, und die dem Auto-

konfigurationsmodul 10 bei einem Hardware-Ereignis 11 kommuniziert werden bzw. wurden.

Im Folgenden wird der Ablauf beispielhaft für die Upstream-Schnittstelle 2 und die Downstream-Schnittstelle 4, die von einem Anwender neu hinzugefügt wird, beschrieben. Sind noch andere Schnittstellen 3, 5 betroffen oder fallen diese weg oder werden bei Inbetriebnahme erstmalig erfasst, kann der Ablauf analog sein.

10

In Reaktion auf das Erfassen der Schnittstelle 2 wählt das Autokonfigurationsmodul 10 die zugehörige Schablonen-Datei 14 aus, also diejenige Schablonen-Datei 14, die für ein solches Hardware-Ereignis 11, konkret für Upstream-Schnittstellen vorgesehen ist.

15

Die Auswahl der Schablonen-Datei 14 für die Schnittstelle 2 erfolgt dabei anhand des Namens der Schnittstelle 2, der vorliegend "eth0" lautet, wobei betont sei, dass dies rein beispielhaft zu verstehen ist. Im vorliegenden Falle wird bzw. wurde der Upstream-Schnittstelle 2 gezielt dieser Name gegeben.

20

Die ausgewählte Schablonendatei 14 für die Upstream-Schnittstelle 2 lautet:

25

```
match:
```

```
  device: "eth0"
```

```
networks:
```

30

```
  - {{device.name}}:
```

```
    script:
```

```
      add:
```

```
        - docker network create -d bridge  
          1uplink
```

35

```
        - DSBR=`docker network inspect -format  
          '{{.Id|printf "br-%.12s"}}'  
          1uplink`
```

```
        - ip link set {{device.name}} master $DSBR
```

```
remove:  
- Docker network rm luplink
```

wobei dies rein beispielhaft zu verstehen ist und eine Schablonendatei 14 für eine Upstream-Schnittstelle 2 auch anders
5 lauten kann.

Hinsichtlich der drei Absätze unter „add:“ in der beispielhaft genannten Schablonendatei 14, denen jeweils ein Spiegelstrich voransteht, gilt folgendes: Über den ersten Absatz
10 wird ein internes Netzwerk für die Schnittstelle 2 angelegt, über den zweiten Absatz der tatsächliche Gerätenamen der virtuellen Bridge ermittelt, die zuvor als Teil des Netzwerkes angelegt wurde (bei dem genannten Beispiel ein Linux-Gerätenamen,
15 was nicht einschränkend zu verstehen ist), und über den dritten Absatz wird die Netzwerkschnittstelle 2 an die virtuelle Bridge angeschlossen.

Über den Absatz unter „remove:“ kann das interne Netzwerk zur
20 Schnittstelle 2 wieder entfernt werden. Dieser Befehl würde greifen bzw. greift, wenn die Schnittstelle 2 wieder wegfallen sollte bzw. wegfällt.

Wie man erkennt, werden der Upstream-Schnittstelle 2 im Rahmen des hier beschriebenen Ausführungsbeispiels keine Kommunikations-Container zugeordnet. Für diese Rolle („Upstream“) sind vorliegend keine Container erforderlich, wobei dies rein
25 beispielhaft zu verstehen ist. Selbstverständlich ist es prinzipiell möglich, dass alternativ dazu auch wenigstens einer Upstream-Schnittstelle 2 wenigstens ein Container zugeordnet wird.
30

Als weiteres Beispiel für eine Schablonen-Datei 14 sei genannt:

35

```
match:  
  device: "eth0"  
networks:
```

```
- {{device.name}}:
  script:
    add:
      - docker network create -d macvlan
5      -o parent={{device.name}}
      luplink
    remove:
      - Docker network rm luplink
```

10 Wie man erkennt, ist das zweite Beispiel eines für Macvlan.
Über die Zeile „-o parent={{device.name}}“ kann dabei der für
die Ausführung bzw. Verarbeitung zum Einsatz kommenden In-
stanz, insbesondere Docker, mitgeteilt werden, dass später
Container in diesem Netzwerk mit der Netzwerkschnittstelle
15 eth0 mittels Macvlan zu koppeln sind.

Aus der Schablonen-Datei 14 (gemäß dem ersten oder zweiten
Beispiel) wird von dem Autokonfigurationsmodul 10 eine Netz-
werk-Konfigurationsbeschreibung 18 für die Schnittstelle 2
20 erstellt, nämlich:

```
#!/bin/bash
docker network create -d bridge \
luplink
25 DSBR=`docker network inspect -format \
`{{.Id|printf "br-%.12s"}}` \
luplink`
ip link set eth0 master $DSBR
```

30 Unter Verwendung des zweitgenannten Beispiels einer Schablon-
nen-Datei 14 folgende Netzwerk-Konfigurationsbeschreibung 18
erstellt:

```
#!/bin/bash
35 docker network create -d macvlan -o parent=eth0 \
luplink
```

Damit kann der für die Ausführung bzw. Verarbeitung zum Einsatz kommenden Instanz, insbesondere Docker, das Netzwerk vom Typ Macvlan angekündigt werden. Es werden insbesondere noch keine virtuellen Netzwerkkomponenten, etwa im Linux-Kernel, angelegt. Es wird lediglich vermerkt, dass mit eth0 per Macvlan zu verbinden sein wird.

Die Erstellung der Netzwerk-Konfigurationsbeschreibung 18 aus der Schablonen-Datei 14 erfolgt, indem das Autokonfigurationsmodul 10 den mehrfach in der Schablonen-Datei 14 auftauchenden Platzhalter `{{device.name}}` durch konkrete Werte, vorliegend den konkreten Schnittstellen-Namen `ens33p0` ersetzt.

Es sei angemerkt, dass in der Figur neben der Netzwerk-Konfigurationsbeschreibung 18 zwei Teilkonfigurationsbeschreibungen 19, 20 gezeigt sind. Diese beiden sind für den Fall vorhanden, dass die erstellte Konfigurationsbeschreibung einen oder mehrere Kommunikations-Container 21, 22 betrifft (Container-Konfigurationsbeschreibung 18), wie es für die Schnittstelle 4 der Fall ist und weiter unten noch im Einzelnen beschrieben wird. Was die Schnittstelle 2 angeht, kann man vereinfacht davon ausgehen, dass die Netzwerk-Konfigurationsbeschreibung 18 nur durch den oberen Teil 20 gebildet wird.

Die Netzwerk-Konfigurationsbeschreibung 18 (bzw. 20) wird umgesetzt bzw. ausgeführt, um das System 1 gemäß dieser zu konfigurieren.

Das System 1 weist ein Container-Controller-Modul 23 und ein Netzwerk-Controller-Modul 24 auf, die vorliegend beide Software-implementiert sind.

Das Netzwerk-Controller-Modul 24 dient bei dem hier beschriebenen Beispiel der Umsetzung bzw. Ausführung der Netzwerk-Konfigurationsbeschreibung 18 (bzw. 20) für die Schnittstelle 2. Es ist vorliegend, da die Netzwerk-Konfigurationsbeschrei-

5 bung 18 für die Schnittstelle 2 in Form eines Shell-Scripts bereitgestellt wird, konkret als Shell-Script-Interpreter ausgeführt bzw. umfasst einen solchen. Es versteht sich, dass das Netzwerk-Controller-Modul 24, wenn die Netzwerk-Konfigurationsbeschreibung 18 in anderer Form erzeugt wird, auch anders ausgeführt sein kann, beispielsweise als systemd oder Kubernetes Controller bzw. solche umfassen kann.

10 Durch die Ausführung der Netzwerk-Konfigurationsbeschreibung 18 durch das Netzwerk-Controller-Modul 24 wird eine virtuelle Bridge angelegt und die Schnittstelle 2 mit dieser virtuellen Bridge verbunden. Mit dieser virtuellen Bridge können später beispielsweise Kommunikations-Container 21, 22, die anderen Schnittstellen 4 zugeordnet sind bzw. werden, verbunden werden.
15 den.

20 Hinsichtlich der im laufenden Betrieb hinzukommenden Downstream-Schnittstelle 4 gilt im Rahmen des hier beschriebenen Ausführungsbeispiels das Folgende:

25 In Reaktion auf das neue Hinzufügen der Schnittstelle 4, insbesondere durch einen Anwender, wählt das Autokonfigurationsmodul die Schablonen-Datei 12 aus, also diejenige Schablonen-Datei 12, die für ein solches Hardware-Ereignis 11, konkret für Downstream-Schnittstellen vorgesehen ist.

30 Die Auswahl erfolgt dabei anhand des Namens der hinzugefügten Schnittstelle, der vorliegend "ens33p2" lautet, bzw. anhand eines insbesondere ersten Teils des Namens "ens*", wobei betont sei, dass dies rein beispielhaft zu verstehen ist. Im vorliegenden Falle wird Downstream-Schnittstellen, welche der Anbindung von Feldbus-Netzwerksegmenten dienen, gezielt ein Name gegeben, der mit den drei Buchstaben "ens" beginnt.

35 Die ausgewählte Schablonen-Datei 12 für die Schnittstelle 4 sieht vorliegend wie folgt aus, wobei betont sei, dass dies eine rein beispielhafte Fassung ist, und Schablonen-Dateien

für Downstream-Schnittstellen 4, 5 auch eine andere Form aufweisen können:

```
match:
5   device: "ens*"
networks:
  - {{device.name}}:
    script:
      add:
10    - ip link add br{{device.name}} type bridge
      - ip link set br{{device.name}} up
      - ip link set {{device.name}} master br{{device.name}}
      - ip link {{device.name}} up
      remove:
15    - ip link del br{{device.name}}
containers:
  v6router:
    image: ...
    networks:
20    - uplink
      - {{device.name}}
  anat64:
    image: ...
    networks:
25    - {{device.name}}
```

"match:" bzw. die ersten beiden Zeilen dieser Datei stellen dabei die den Schnittstellen-Namen betreffende Filterregel dar. Beginnt ein Name einer erstmals erfassten, neu hinzugefügten oder wegfallenden Schnittstelle 2, 3, 4, 5 mit "ens*", wird diese Schablonen-Datei 12 ausgewählt.

Als zweites Beispiel für eine Schablonen-Datei 12 für eine Downstream-Schnittstelle 4 sei genannt:

```
35
match:
  device: "ens*"
networks:
```

```

- {{device.name}}:
  script:
    add:
      - docker network create -d bridge
5      2{{device.name}}_downstream
      - DSBR=`docker network inspect -format
        `{{.Id|printf "br-%.12s"}}`
        2{{device.name}}_downstream`
      - ip link set {{device.name}} master $DSBR
10    remove:
      - Docker network rm 2{{device.name}}_downstream

containers:
  v6router:
    image: ...
15  networks:
      - 1uplink
      - 2{{device.name}}_downstream
  anat64:
    image: ...
20  networks:
      - 2{{device.name}}_downstream

```

Dieses Beispiel ist insbesondere vorteilhaft, wenn ein der Netzwerk-Schnittstelle 2 zugeordnetes internes Netzwerk bereits auf anderem Wege angelegt wurde. Die Schablone 12 beschreibt nun, wie das noch fehlende Bridge-Network für die neue Schnittstelle 4 anzulegen ist: diese ist ein Shell-Skript, welches das Netzwerk per Docker-Kommando anlegt und die neue Schnittstelle 4 unter die Kontrolle der Bridge stellt. Mit dem ersten Beispiel übereinstimmend handelt es sich um ein Skript und es wird ein Bridge-Netzwerk angelegt.

Hinsichtlich der in dem Abschnitt „add“ aufgeführten drei Absätze bzw. Kommandos (jeweils hinter einem Spiegelstrich stehend) gilt, dass der erste ein internes Netzwerk mit einer virtuellen Bridge unter dem Namen „2ens33p2_downstream“ anlegt, wobei der Anteil „ens33p2“ der Name der Netzwerkschnittstelle 4 ist, der zweite den tatsächlichen Gerätenamen

(in dem genannten Beispiel Linux-Gerätenamen) der virtuellen Bridge ermittelt, die zuvor als Teil des Netzwerks angelegt wurde, und der dritte die Netzwerkschnittstelle 4 an die virtuelle Bridge anschließt, die zuvor als Teil des Netzwerkes angelegt wurde.

Der darauf folgend, konkret nach „remove:“ folgende Absatz entfernt das Netzwerk samt virtueller Bridge zwischen dem Container 21 und der Schnittstelle 4. Diese Absatz greift für den Fall, dass die Schnittstelle 4 wieder wegfällt.

Hinsichtlich der beiden Absätze nach „networks:“ gilt, dass „luplink“ den Bezug zum bereits früher angelegten internen Netzwerk an der Schnittstelle 2 herstellt, und „2{{device.name}}_downstream“ den Bezug zu dem vorangegangen für die Schnittstelle 4 angelegten internen Netzwerk.

Über “- 2{{device.name}}_downstream“ in der letzten Zeile der Schablone 12 wird der Container an das interne Netzwerk zur Schnittstelle 4 angeschlossen. Eine direkte Verbindung mit dem internen Netzwerk der Schnittstelle 2 ist beispielhaft nicht erforderlich, stattdessen wird diese Verbindung indirekt über den v6router-Container hergestellt.

Als drittes Beispiel für eine Schablonen-Datei 12 für eine Downstream-Schnittstelle 4 sei ferner genannt:

```
match:
  device: "ens*"
networks:
  - {{device.name}}:
    script:
      add:
        - docker network create -d macvlan
          -o parent={{device.name}}
            2{{device.name}}_downstream
      remove:
        - Docker network rm 2{{device.name}}_downstream
```

```
containers:
  v6router:
    image: ...
    networks:
5     - luplink
      - 2{{device.name}}_downstream
  anat64:
    image: ...
    networks:
10    - 2{{device.name}}_downstream
```

Wie man erkennt, ist dieses Beispiel wiederum eines für Macvlan.

15 Das Autokonfigurationsmodul 10 erstellt unter Verwendung der Schablonen-Datei 12 (gemäß dem erstgenannten bzw. dem zweitgenannten bzw. dem drittgenannten Beispiel) eine Container-Konfigurationsbeschreibung 18, die eine erste Teilkonfigurationsbeschreibung 19 und eine zweite Teilkonfigurationsbe-

20 schreibung 20 umfasst.

Es sei angemerkt, dass in der vereinfachten Figur aus Gründen der besseren Übersichtlichkeit der Ablauf der Erstellung einer Konfigurationsbeschreibung 18 nur ein Mal für beide

25 Schnittstellen 2 und 4 schematisch dargestellt ist. Das Element mit dem Bezugszeichen 18 repräsentiert daher sowohl die für die Schnittstelle 2 erstellte Netzwerk-Konfigurationsbeschreibung als auch die für die Schnittstelle 4 erstellte Container-Konfigurationsbeschreibung mit den zwei Teilkonfigurationsbeschreibungen 19 und 20.

30

Die Erstellung der ersten und der zweiten Teilkonfigurationsbeschreibung 19, 20 der Container-Konfigurationsbeschreibung 18 aus der Schablonen-Datei 12 erfolgt dabei, indem das Autokonfigurationsmodul 10 den mehrfach in der Schablonen-Datei

35 12 auftauchenden Platzhalter {{device.name}} durch konkrete Werte, vorliegend den konkreten Schnittstellen-Namen, also ens33p2 ersetzt.

Die erste Teilkonfigurationsbeschreibung 19 (und die zugehörigen Zeilen 14 bis 23 der Schablonen-Datei 12) betrifft vorliegend zwei Kommunikations-Container 21, 22, die jeweils wenigstens eine Anwendung umfassen, mittels derer Netzwerk-

5 Funktionen bereitgestellt werden können.

Die erste Teilkonfigurationsbeschreibung 19 der Container-Konfigurationsbeschreibung 18 sieht - ausgehend von dem erstgenannten Beispiel einer Schablone 12 -konkret wie folgt aus,

10 wobei diese wieder rein beispielhaft zu verstehen ist:

```
version: "3"
services:
  v6router:
15   image: ...
      networks:
        - lupstream
        - 2downstream
  anat64:
20   image: ...
      networks:
        - 2downstream
networks:
  lupstream:
25   external: upstream
  2downstream:
      external: ens33p2
```

Ausgehend von dem vorstehend zweitgenannten Beispiel einer Schablonen-Datei 12 für die Downstream-Schnittstelle 4 ergibt sich für die erste Teilkonfigurationsbeschreibung 19:

30

```
version: "3"
services:
35  v6router:
      image: ...
      networks:
        - lupstream
```

```
    - 2ens33p2_downstream
  anat64:
    image: ...
    networks:
5    - 2ens33p2_downstream
networks:
  lupstream:
    external: true
  2ens33p2_downstream:
10   external: true
```

Ausgehend von dem dritten Beispiel einer Schablonen-Datei 12 für die Downstream-Schnittstelle 4 ergibt sich für die erste Teilkonfigurationsbeschreibung 19:

```
15 version: "3"
services:
  v6router:
    image: ...
20   networks:
    - lupstream
    - 2ens33p2_downstream
  anat64:
    image: ...
25   networks:
    - 2ens33p2_downstream
networks:
  lupstream:
    external: true
30  2ens33p2_downstream:
    external: true
```

Wie man erkennt, werden vorliegend - für alle drei Beispiele - Netzwerkfunktionen eines IPv6-Routers und eines NAT64-Routers in containerisierter Form bereitgestellt.

Die jeweilige erste Teilkonfigurationsbeschreibung 19 stellt die Konfigurationsbeschreibung für die in dem jeweiligen Kon-

text 15, 16, 17 gewünschten bzw. benötigten Kommunikations-
Container 21, 22 dar. Sie betrifft also den vom Kontext 15,
16, 17 abhängigen, von der Art bzw. Rolle der betroffenen
Schnittstelle 4 abhängigen Container-Ausbau. Sie dient insbe-
5 sondere der weiteren Verarbeitung in einem Container-System
bzw. Container-Werkzeug, wie etwa Docker Compose.

Die jeweilige zweite Teilkonfigurationsbeschreibung 20 (und
die zugehörigen Zeilen 3 bis 12 bzw. 13 der jeweils zugehöri-
10 gen Schablonen-Datei 12) betrifft Netzwerk-Funktionen zur
Verbindung der Kommunikations-Container 20, 21 mit der be-
troffenen Hardware-Netzwerkschnittstelle 4 in Form einer Kon-
figurationsbeschreibung für vorliegend eine virtuelle Bridge
auf Betriebssystemebene bzw. - für den Fall, dass eine
15 Schnittstelle 4 nicht wie vorliegen hinzugefügt sondern ent-
fernt wird, Löschanweisungen ("remove: ...") für eine sol-
che. Diese greifen für den Fall, dass dem Autokonfigurations-
modul 10 kommuniziert wurde, dass eine Downstream-Schnitt-
stelle 4 zur Anbindung von Feldbus-Netzwerksegmenten wegge-
20 fallen ist.

Die zweite Teilkonfigurationsbeschreibung 20 sieht - ausge-
hend von dem erstgenannte Beispiel der Schablonen-Datei 12 -
konkret wie folgt aus, was wiederum rein beispielhaft zu ver-
25 stehen ist:

```
#!/bin/bash  
ip link add brens33p2 type bridge  
ip link set brens33p2 up  
30 ip link set ens33p2 master brens33p2  
ip link ens33p2 up
```

Aus dem vorstehend zweitgenannten Beispiel der Schablonen-
Datei 12 erhält man für die zweite Teilkonfigurationsbe-
35 schreibung 20:

```
#!/bin/bash
```

```
docker network create -d bridge \  
2ens33p2 downstream  
DSBR=`docker network inspect -format  
'{{.Id|printf "br-%.12s"}}'  
5 2ens33p2 downstream`  
ip link set ens33p2 master $DSBR
```

Wie man erkennt, gleicht dieses Beispiel einer zweiten Teil-
konfigurationsbeschreibung 20 in seiner Struktur dem Beispiel
10 einer Netzwerk-Konfigurationsbeschreibung 18, welches vorstehend für die Schnittstelle 2 beschrieben wurde.

Aus dem dritten Beispiel der Schablonen-Datei 12 erhält man für die zweite Teilkonfigurationsbeschreibung 20:

```
15 #!/bin/bash  
docker network create -d macvlan \  
-o parent=2ens33p2 \  
2ens33p2 downstream
```

20 Die jeweilige zweite Teilkonfigurationsbeschreibung 20 beschreibt - von der Konfigurationsbeschreibung 19 für den Container-Ausbau getrennt - erforderliche virtuellen Netzwerkfunktionen, die in dem jeweiligen Kontext 15, 16, 17 für das
25 Verbinden der jeweiligen Kommunikations-Container 21, 22 mit der jeweiligen Schnittstelle 4 erforderlich sind. Vorliegend liegt die generierte Beschreibung als Shell-Script vor. Alternativ kann sie auch als "systemd units", als Clustersystem-Objekte (Kubernetes) oder dergleichen mehr ausgeführt
30 sein.

Anhand dieser Beschreibung 20 sorgt das Autokonfigurationsmodul 10 insbesondere dafür, dass wenigstens ein das interne Netzwerk bildender Software-Switch anstatt mit dem Host-IP-
35 Stack, wie es bei Docker üblich ist, allein mit der neu hinzugekommenen Schnittstelle 4 verknüpft wird.

Die erste Teilkonfigurationsbeschreibung 19 und die zweite Teilkonfigurationsbeschreibung 20 (gemäß dem ersten, zweiten oder dritten Beispiel) werden umgesetzt bzw. ausgeführt, um das System 1 gemäß dieser zu konfigurieren.

5

Die Umsetzung bzw. Ausführung der ersten, die Container 21, 22 betreffende Teilkonfigurationsbeschreibung 19 erfolgt dabei mittels des Container-Controller-Moduls 23 und die Umsetzung der zweiten, die Betriebssystemebene betreffenden Teilkonfigurationsbeschreibung 20 durch das Netzwerk-Controller-Modul 24.

10

Das Container-Controller-Modul 23 ist vorliegend Softwareimplementiert, konkret als Docker Composer Werkzeug ausgeführt bzw. umfasst ein solches, wobei dies beispielhaft zu verstehen ist. Wird alternativ oder zusätzlich zu Docker beispielsweise auf Kubernetes als Container-Technologie zurückgegriffen, kann das Container-Controller-Modul 23 auch als Kubernetes Pod Scheduler Controller ausgeführt sein bzw. einen solche umfassen. Andere oder weitere Container-Technologien sind natürlich ebenfalls möglich.

15

20

Es sei angemerkt, dass im Rahmen der hier beschriebenen Beispiele die zweite Teilkonfigurationsbeschreibung 20 vor der ersten Teilkonfigurationsbeschreibung 19 ausgeführt wird, wobei dies nicht einschränkend zu verstehen ist. Im Fall von Kubernetes beispielsweise werden die beiden Teilkonfigurationsbeschreibungen 19, 20 in den Cluster „eingespielt“ und dann automatisch in der richtigen Reihenfolge durch die Cluster-Mechanismen ausgeführt (wie beispielsweise kubelet und CNI plugins). Für den Fall, dass Kubernetes als Instanz zur Ausführung der ersten Teilkonfigurationsbeschreibung zum Einsatz kommt, kann beispielsweise vorgesehen sein, dass ein "kubelet" auf dem System, auf dem die Container 21, 22 gestartet werden sollen, seinerseits ein (Haupt-)CNI-Plugin startet und diesem die erste Teilkonfigurationsbeschreibung 19 zur Ausführung übergibt. Das Haupt-CNI-Plugin kann beispielsweise durch Multus gegeben sein. Dieses kann

30

35

dann zum Beispiel anhand seiner eigenen Konfiguration sowie der Teilkonfigurationsbeschreibung 19 entscheiden, welche weiteren CNI-Plugins ggf. der Reihe nach aufzurufen sind, wie beispielsweise das „bridge“ CNI-Plugin oder das „macvlan“ CNI-Plugin. Das Haupt-CNI-Plugin kontrolliert hierbei die Weitergabe der erforderlichen Teile aus der Teilkonfigurationsbeschreibung 19 zu den aufzurufenden CNI-Plugins.

10 Infolge der Ausführung der ersten Teilkonfigurationsbeschreibung 19 durch das Container-Controller-Modul 23 werden die Kommunikations-Container 21, 22 mit den IPv6- und NAT64-Router-Funktionen erzeugt.

15 Das Autokonfigurationsmodul 10 teilt dem Container-Controller-Modul 23 vorliegend weiterhin direkt oder indirekt mit, dass die Kommunikations-Container 21, 22 im vorliegenden Falle zu starten sind, da die Schnittstelle 4 hinzugefügt und nicht entfernt wurde. Wäre letzteres alternativ der Fall, würde das Autokonfigurationsmodul 12 dem Container-Controller-Modul 23 mitteilen, dass die Kommunikations-Container 21, 20 22 nicht zu starten sondern zu stoppen sind.

Das Netzwerk-Controller-Modul 24 für die Ausführung bzw. Umsetzung der zweiten Teilkonfigurationsbeschreibung 20 ist, wie oben bereits angemerkt, ebenfalls Software-implementiert und als Shell-Script-Interpreter ausgeführt bzw. umfasst einen solchen.

Es sei angemerkt, dass für den Fall, dass anstelle einer 30 Downstream-Schnittstelle 4 eine andere Art von Schnittstelle 3 bzw. eine Schnittstelle 3, mit einer anderen Rolle hinzugefügt wird, der Ablauf völlig analog sein kann. Lediglich der konkrete Kommunikations-Container-Ausbau gemäß der ersten Teilkonfigurationsbeschreibung 19 und die virtuellen Netz- 35 funktionen gemäß der zweiten Teilkonfigurationsbeschreibung 20 können bzw. werden dann abweichen. Für den Fall, dass beispielsweise eine Drahtlos-Netzwerkschnittstelle 3 hinzugefügt wird, würden nicht die Funktionen eines IPv6- und NAT64-Rou-

ters containersiert bereitgestellt, sondern zum Beispiel eine geeignete Software für einen WLAN Access-Point. Ein entsprechender Kommunikations-Container 25 für diesen Kontext 17 ist in der Figur ebenfalls rein schematisch dargestellt.

5

Selbstverständlich ist es auch möglich, dass alternativ zu den vorstehenden Beispielen für die Schnittstelle 2 von dem Autokonfigurationsmodul 10 anstelle einer Netzwerk-Konfigurationsbeschreibung 18 eine Container-Konfigurationsbeschreibung 18 erstellt wird, wenn für die Schnittstelle 2 einer oder mehrere Kommunikations-Container 21, 22, 25 vorgesehen werden sollen. Der Ablauf für die Schnittstelle 2 könnte dann analog zu demjenigen sein, der vorstehend für die Schnittstelle 4 beschrieben wurde.

15

Sofern einer Schnittstelle aufgrund ihrer Rolle wenigstens ein Kommunikations-Container 21, 22, 25 zuzuordnen ist, kann eine Container-Konfigurationsbeschreibung 18 erstellt werden, und für den Fall, dass kein Kommunikations-Container 21, 22, 25 erforderlich bzw. gewünscht ist, eine Netzwerk-Konfigurationsbeschreibung 18.

20

Weiterhin sei betont, dass das Autokonfigurationsmodul 10 für sämtliche Netzwerkschnittstellen 2, 3, 4, 5 eines Systems 1 zum Einsatz kommen kann, jedoch nicht muss. Die erfindungsgemäße Vorgehensweise kann selbstverständlich auch in Kombination mit der aus dem Stand der Technik vorbekannten Vorgehensweise zum Einsatz kommen. Beispielsweise kann alternativ dazu, dass, wie vorstehend beschrieben, das Autokonfigurationsmodul 10 für die Schnittstelle 2 eine Netzwerk-Konfigurationsbeschreibung 18 erstellt, eine solche auf konventionelle Weise insbesondere im Rahmen einer manuellen Konfiguration - etwa durch einen Nutzer - erstellt werden bzw. worden sein. Die manuell erhaltene Netzwerk-Kommunikationsbeschreibung 18 kann dabei zum Beispiel genauso aussehen, wie die vorstehend genannte, vom Autokonfigurationsmodul 18 unter Verwendung der Schablonen-Datei 14 erhaltene.

30

35

In bevorzugter Ausgestaltung gilt zumindest für diejenigen Schnittstellen 3, 4, 5 eines Systems 1, für die jeweils einer oder mehrere Kommunikations-Container 21, 22, 25 vorzusehen sind, dass für diese von dem Autokonfigurationsmodul 10 eine Container-Konfigurationsbeschreibung 18 automatisch erstellt wird. Wenn ein möglichst umfassender automatischer Ablauf gewünscht ist, was in der Regel besonders vorteilhaft ist, können zusätzlich auch für sämtliche Schnittstellen 2 eines Systems, für die keine Kommunikations-Container 21, 22, 25 erforderlich sind, Netzwerk-Konfigurationsbeschreibungen 18 von dem Autokonfigurationsmodul 10 erstellt werden, wie vorstehend für die Schnittstelle 2 beispielhaft beschrieben.

Durch die erfindungsgemäße Vorgehensweise kann die „Lücke“ zwischen industriellen Anwendern und der Container-Technologie, wie beispielsweise Docker, geschlossen werden. So wird es möglich, die großen Vorteile dieser Technologie für Netzwerkfunktionen nutzbar zu machen, ohne dass es seitens der Anwender eines speziellen IT- bzw. Netzwerks- bzw. Container-Know-Hows bedarf. Unter Verwendung des Autokonfigurationsmoduls 10 und der Schablonen-Dateien 12, 13, 14 können Container mit Netzwerkfunktionen dynamisch und automatisch mit ihrer physikalischen Netz-Außenwelt verbunden werden, dies passend für den jeweiligen Kontext 15, 16, 17. Industrielle Anwender müssen kein IT-Fachwissen über Container, etwa Docker Container, oder über zugehörige Werkzeuge, etwa Docker Network Driver, oder auch über Software-Switches haben, und können dennoch die Vorteile dieser Technologie für eine automatische Konfiguration im Falle einer Änderung bzw. Ergänzung des Hardware-Schnittstellenausbaus ihrer Systeme bzw. für deren Inbetriebnahme nutzen. Es wird besonders einfach für Anwender, Systeme 1 aus dem Bereich der Automatisierungstechnik um Netzwerkzu- oder Abgänge zu erweitern. Eine manuelle Konfiguration mittels zusätzlichem Engineering- bzw. Konfigurations-Werkzeug(en) ist nicht mehr nötig. Es genügt das Anstecken der entsprechenden Netzwerk-Schnittstellen-Hardware, an welches sich dann eine vollständig automatische Konfiguration anschließt.

Obwohl die Erfindung im Detail durch das bevorzugte Ausführungsbeispiel näher illustriert und beschrieben wurde, so ist die Erfindung nicht durch die offenbarten Beispiele eingeschränkt und andere Variationen können vom Fachmann hieraus
5 abgeleitet werden, ohne den Schutzzumfang der Erfindung zu verlassen.

Patentansprüche

1. Verfahren zum automatischen Konfigurieren eines Systems (1), bei dem

- 5 - bei Inbetriebnahme des Systems (1) und/oder im laufenden Betrieb des Systems (1) überwacht wird, welche physikalischen und/oder virtuellen Hardware-Netzwerkschnittstellen (2, 3, 4, 5) das System (1) aufweist,
- 10 - für den Fall, dass eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle (2, 3, 4, 5) bei Inbetriebnahme des Systems (1) erstmals erfasst wird, und/oder eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle (2, 3, 4, 5) im Betrieb des Systems (1) neu hinzukommt oder wegfällt, dies einem Autokonfigurationsmodul (10) des
- 15 Systems (1) zusammen mit einer Information über die Art der betroffenen Hardware-Netzwerkschnittstelle (2, 3, 4, 5) und/oder die Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes (6, 7, 8, 9) kommuniziert wird,
- 20 - das Autokonfigurationsmodul (10) für wenigstens eine der betroffenen Hardware-Netzwerkschnittstellen (2, 3, 4, 5) unter Verwendung einer Schablonen-Datei (12, 13, 14) eine Konfigurationsbeschreibung (18) erstellt, die eine erste Teilkonfigurationsbeschreibung (19) für wenigstens einen Kommunikations-Container (21, 22, 25), der wenigstens eine
- 25 Anwendung umfasst, mittels derer Netzwerk-Funktionen bereitgestellt werden können, und eine zweite Teilkonfigurationsbeschreibung (20) für Netzwerk-Funktionen zur Verbindung des wenigstens einen Kommunikations-Containers (21, 22, 25) mit der wenigstens einen betroffenen Hardware-
- 30 Netzwerkschnittstelle (2, 3, 4, 5) umfasst, und
- die Konfigurationsbeschreibung ausgeführt wird, um das System (1) gemäß dieser zu konfigurieren, wobei der wenigstens eine Kommunikations-Container (21, 22, 25) erzeugt und mit der wenigstens einen betroffenen Hardware-
- 35 Schnittstelle (2, 3, 4, 5) verbunden wird.

2. Verfahren nach Anspruch 1,
dadurch gekennzeichnet, dass die Konfigurationsbeschreibung
(18) erhalten wird, indem einer oder mehrere in der Schablonen-Datei (12) vorgesehene Platzhalter durch wenigstens einen
5 zu der betroffenen Hardware-Netzwerkschnittstelle (2, 3, 4,
5) gehörigen Parameter ersetzt werden.
3. Verfahren nach Anspruch 1 oder 2,
dadurch gekennzeichnet, dass die Schablonen-Datei (12) aus
10 einer Mehrzahl von unterschiedlichen Schablonen-Dateien (12,
13, 14) ausgewählt wird.
4. Verfahren nach Anspruch 3,
dadurch gekennzeichnet, dass die Auswahl anhand von in den
15 Schablonen-Dateien (12, 13, 14) enthaltenen Filterregeln erfolgt.
5. Verfahren nach Anspruch 3 oder 4,
dadurch gekennzeichnet, dass für mehrere Arten von physikalischer und/oder virtueller Hardware-Netzwerkschnittstelle (2,
20 3, 4, 5), welche das System (1) aufweisen kann, jeweils wenigstens eine Schablonen-Datei (12, 13, 14) gespeichert ist,
und/oder dass für zumindest drei verschiedene Arten von physikalischen und/oder virtuellen Hardware-Netzwerkschnittstellen
25 (2, 3, 4, 5) jeweils wenigstens eine Schablonen-Datei
(12, 13, 14) gespeichert ist.
6. Verfahren nach einem der vorhergehenden Ansprüche,
dadurch gekennzeichnet, dass dem Autokonfigurationsmodul (10)
30 das erstmalige Erfassen einer physikalischen oder virtuellen Hardware-Netzwerkschnittstelle (2, 3, 4, 5) bei Inbetriebnahme des Systems (1) und/oder das neue Hinzukommen und/oder Wegfallen einer physikalischen oder virtuellen Hardware-Netzwerkschnittstelle (2, 3, 4, 5) im Betrieb des Systems (1)
35 kommuniziert wird, indem ein auf dem System (1) laufendes Betriebssystem dem Autokonfigurationsmodul (10) ein entsprechendes Ereignis meldet.

7. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass der wenigstens eine Kommunikations-Container (21, 22, 25) von einem Container-Controller-Modul (23) des Systems (1) erzeugt wird.

5

8. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die zweite Teilkonfigurationsbeschreibung (20) von einem Netzwerk-Controller-Modul (24) des Systems (1) ausgeführt wird und/oder dass die erste Teilkonfigurationsbeschreibung (19) von einem Container-Controller-Modul (23) des Systems (1) ausgeführt wird.

10

9. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass wenigstens eine Anwendung des wenigstens einen Kommunikations-Containers (21, 22, 25) gemäß der ersten Teilkonfigurationsbeschreibung (19) Funktionen eines IPv6- und/oder NAT64-Routers und/oder eines Namensdienst-Clients oder Servers und/oder eines Brouters bereitstellt und/oder durch eine Software für einen WLAN-Access-Point gegeben ist oder eine solche umfasst.

15

20

10. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass für den Fall, dass eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle (2, 3, 4, 5) im Betrieb des Systems wegfällt, das Autokonfigurationsmodul (10) das Entfernen und/oder Deaktivieren von Netzwerk-Funktionen für die betroffene Schnittstelle (2, 3, 4, 5) und/oder das Stoppen von Kommunikations-Containern für die betroffene Schnittstelle (2, 3, 4, 5) veranlasst.

25

30

11. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass das Autokonfigurationsmodul (10) für wenigstens eine weitere der betroffenen Hardware-Netzwerkschnittstellen (2, 3, 4, 5) eine Netzwerk-Konfigurationsbeschreibung (18) erstellt, die wenigstens ein virtuelles Netzwerk und/oder wenigstens eine virtuelle Bridge und/oder wenigstens einen virtuellen Switch für die wenigstens eine

35

weitere Hardware-Netzwerkschnittstelle (2, 3, 4, 5) und keinen Kommunikations-Container für diese betrifft.

12. System (1) zur Durchführung des Verfahrens nach einem der
5 vorhergehenden Ansprüche, wobei das System (1) derart ausgebildet und/oder eingerichtet ist,

- dass einem Autokonfigurationsmodul (10) des Systems (1) gemeldet wird, wenn eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle (2, 3, 4, 5) bei Inbetriebnahme des Systems erstmals erfasst wird, und/oder eine
10 physikalische oder virtuelle Hardware-Netzwerkschnittstelle (2, 3, 4, 5) im Betrieb des Systems (1) neu hinzukommt oder wegfällt, wobei zusammen mit einer solchen Meldung eine Information über die Art der betroffenen Hardware-Netzwerkschnittstelle (2, 3, 4, 5) und/oder die Art eines mit dieser verbundenen oder zu verbindenden Netzwerkes (6, 7, 8, 9) kommuniziert wird,

- dass das Autokonfigurationsmodul (10) für wenigstens eine der betroffenen Hardware-Netzwerkschnittstellen (2, 3, 4, 5) unter Verwendung einer Schablonen-Datei (12, 13, 14) eine Konfigurationsbeschreibung (18) erstellt, die eine erste Teilkonfigurationsbeschreibung (10) für wenigstens einen Kommunikations-Container (21, 22, 25), der wenigstens eine Anwendung umfasst, mittels derer Netzwerk-
20 Funktionen für die oder die jeweils betroffene Schnittstelle (2, 3, 4, 5) bereitgestellt werden können, und eine zweite Teilkonfigurationsbeschreibung (20) für Netzwerk-Funktionen zur Verbindung des wenigstens einen Kommunikations-Containers (21, 22, 25) mit der oder der jeweils betroffenen Schnittstelle (2, 3, 4, 5) umfasst, und
25

- dass die Konfigurationsbeschreibung (18) ausgeführt wird, um das System (1) gemäß dieser zu konfigurieren, wobei der wenigstens eine Kommunikations-Container (21, 22, 25) erzeugt und mit der wenigstens einen betroffenen Hardware-Schnittstelle (2, 3, 4, 5) verbunden wird.
35

13. System nach Anspruch 12,
dadurch gekennzeichnet, dass das Autokonfigurationsmodul (10)
ausgebildet und/oder eingerichtet ist, um einen oder mehrere
in der Schablonen-Datei (12, 13, 14) vorgesehene Platzhalter
5 durch wenigstens einen zu der oder der jeweils betroffenen
Hardware-Netzwerkschnittstelle (2, 3, 4, 5) gehörigen Parame-
ter zu ersetzen und so die Konfigurationsbeschreibung (18) zu
erhalten.
- 10 14. System nach Anspruch 12 oder 13,
dadurch gekennzeichnet, dass das Autokonfigurationsmodul (10)
ausgebildet und/oder eingerichtet ist, um die Schablonen-
Datei (12, 13, 14) aus einer Mehrzahl unterschiedlichen
Schablonen-Dateien (12, 13, 14) auszuwählen.
- 15 15. System nach Anspruch 14,
dadurch gekennzeichnet, dass das Autokonfigurationsmodul (10)
ausgebildet und/oder eingerichtet ist, um die Auswahl anhand
von in den Schablonen-Dateien (12, 13, 14) enthaltenen Fil-
20 terregeln zu treffen.
16. System nach Anspruch 14 oder 15,
dadurch gekennzeichnet, dass für mehrere Arten von physikali-
schen und/oder virtuellen Hardware-Netzwerkschnittstellen (2,
25 3, 4, 5), welche das System (1) aufweisen kann, jeweils we-
nigstens eine Schablonen-Datei (12, 13, 14) auf dem System
(1) gespeichert ist, und/oder dass für zumindest drei ver-
schiedene Arten von physikalischen und/oder virtuellen Hard-
ware-Netzwerkschnittstellen (2, 3, 4, 5) jeweils wenigstens
30 eine Schablonen-Datei (12, 13, 14) auf dem System (1) gespei-
chert ist.
17. System nach einem der Ansprüche 12 bis 16,
dadurch gekennzeichnet, dass das System (1) ausgebildet
35 und/oder eingerichtet ist, um dem Autokonfigurationsmodul
(10) das erstmalige Erfassen einer physikalischen oder virtu-
ellen Hardware-Netzwerkschnittstelle (2, 3, 4, 5) bei Inbe-
triebnahme des Systems (1) und/oder das neue Hinzukommen

und/oder Wegfallen einer physikalischen oder virtuellen Hardware-Netzwerkschnittstelle (2, 3, 4, 5) im Betrieb des Systems (1) zu kommunizieren, indem ein auf dem System laufendes Betriebssystem dem Autokonfigurationsmodul (10) ein entsprechendes Ereignis meldet.

18. System nach einem der Ansprüche 12 bis 17, dadurch gekennzeichnet, dass ein Container-Controller-Modul (23) des Systems (1) ausgebildet und/oder eingerichtet ist, um den wenigstens einen Kommunikations-Container (21, 22, 25) zu erzeugen.

19. System (1) nach einem der Ansprüche 12 bis 18, dadurch gekennzeichnet, dass das System (1) ein Netzwerk-Controller-Modul (24) umfasst, welches ausgebildet und/oder eingerichtet ist, um die zweite Teilkonfigurationsbeschreibung (20) auszuführen, und/oder das System (1) ein einem Container-Controller-Modul (23) umfasst, welches ausgebildet und/oder eingerichtet ist, um die erste Teilkonfigurationsbeschreibung auszuführen.

20. System (1) nach einem der Ansprüche 12 bis 19, dadurch gekennzeichnet, dass das System (1) derart ausgebildet und/oder eingerichtet ist, dass wenigstens eine Anwendung des wenigstens einen Kommunikations-Containers (21, 22, 25) gemäß der ersten Teilkonfigurationsbeschreibung (19) Funktionen eines IPv6- und/oder NAT64-Routers und/oder eines Namensdienst-Clients oder Servers und/oder eines Brouters bereitstellt und/oder durch eine Software für einen WLAN-Access-Point gegeben ist oder eine solche umfasst.

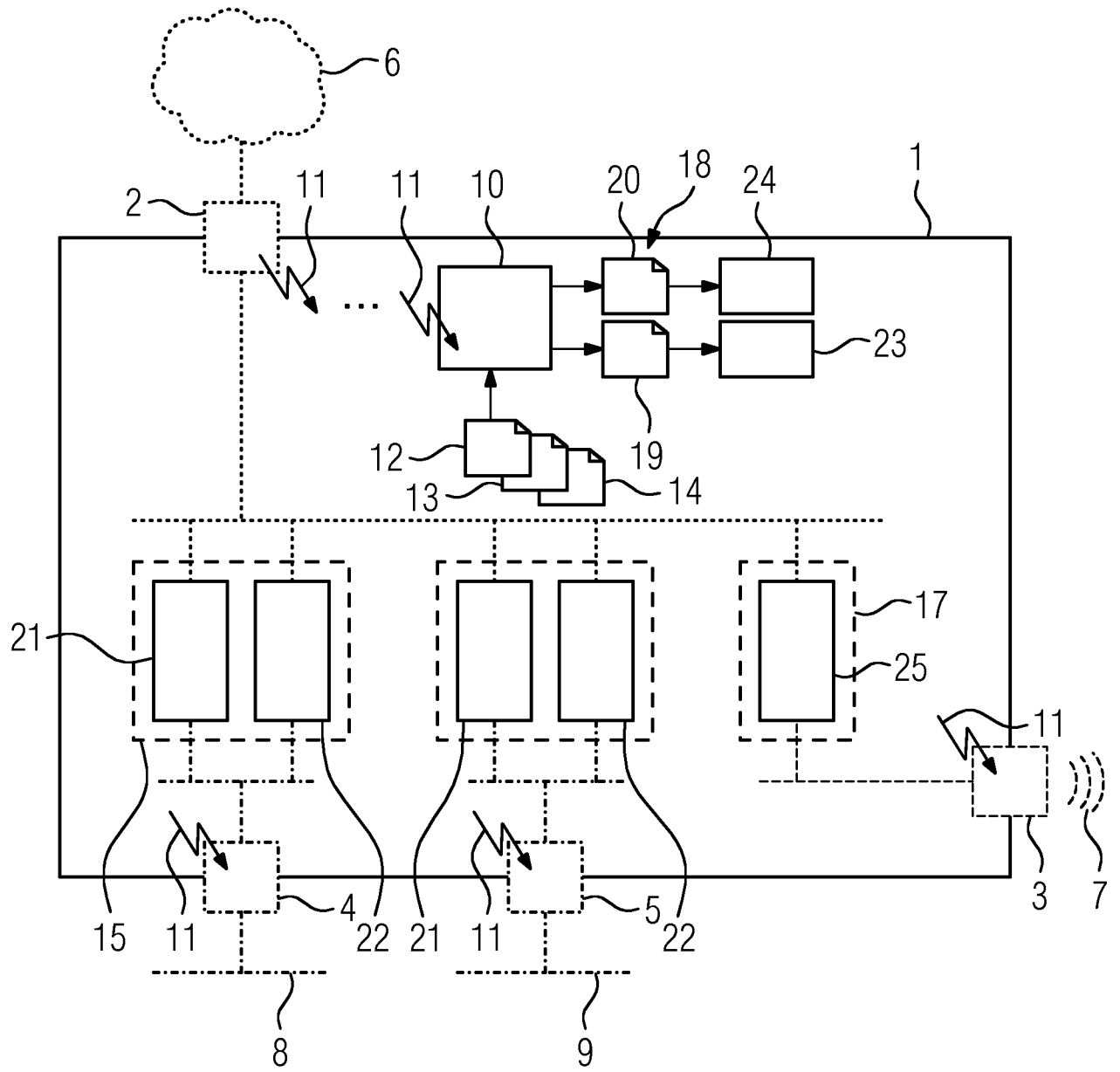
21. System (1) nach einem der Ansprüche 12 bis 20, dadurch gekennzeichnet, dass das System (1) derart ausgebildet und/oder eingerichtet ist, dass für den Fall, dass dem Autokonfigurationsmodul (10) gemeldet wird, dass eine physikalische oder virtuelle Hardware-Netzwerkschnittstelle (2, 3, 4, 5) im Betrieb des Systems (1) wegfällt, das Autokonfigurationsmodul (10) das Entfernen und/oder Deaktivieren von Netz-

werk-Funktionen und/oder virtuellen Netzwerken für die betroffene Schnittstelle (2, 3, 4, 5) veranlasst.

22. System (1) nach einem der Ansprüche 12 bis 21,
5 dadurch gekennzeichnet, dass das Autokonfigurationsmodul (10) ausgebildet und/oder eingerichtet ist, für wenigstens eine weitere der betroffenen Hardware-Netzwerkschnittstellen (2,
3, 4, 5) eine Netzwerk-Konfigurationsbeschreibung (18) zu erstellen, die wenigstens ein virtuelles Netzwerk und/oder we-
10 nigstens eine virtuelle Bridge und/oder wenigstens einen virtuellen Switch für die wenigstens eine weitere Hardware-Netzwerkschnittstelle (2, 3, 4, 5) und keinen Kommunikations-Container für diese betrifft.

15 23. Computerprogramm umfassend Programmcode-Mittel zur Durchführung des Verfahrens nach einem der Ansprüche 1 bis 11.

24. Computerlesbares Medium, das Instruktionen umfasst, die, wenn sie auf wenigstens einem Computer ausgeführt werden, den
20 wenigstens einen Computer veranlassen, die Schritte des Verfahrens nach einem der Ansprüche 1 bis 11 durchzuführen.



INTERNATIONAL SEARCH REPORT

International application No.

PCT/EP2019/074941

A. CLASSIFICATION OF SUBJECT MATTER		
H04L 12/24 (2006.01)i; G06F 9/455 (2018.01)i; H04L 29/08 (2006.01)i; H04L 29/12 (2006.01)n		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols) H04L; G06F		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) EPO-Internal, COMPENDEX, INSPEC, WPI Data		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	VAN ROSSEM STEVEN ET AL. "NFV service dynamicity with a DevOps approach: Insights from a use-case realization" <i>2017 IFIP/IEEE SYMPOSIUM ON INTEGRATED NETWORK AND SERVICE MANAGEMENT (IM), IFIP</i> , 08 May 2017 (2017-05-08), pages 674-679 DOI: 10.23919/INM.2017.7987357 XP033127640	1-5,7-16,18-24
A	chapters I, II and IV	6,17
X	CN 108282376 A (UNIV JIANGNAN) 13 July 2018 (2018-07-13)	1-5,7-16,18-24
A	paragraph [0004] - paragraph [0011]	6,17
A	US 2018287883 A1 (JOSHI PRABODH [US] ET AL) 04 October 2018 (2018-10-04) paragraphs [0035] - [0040], [0048], [0056]	1-24
<input checked="" type="checkbox"/> Further documents are listed in the continuation of Box C. <input checked="" type="checkbox"/> See patent family annex.		
<p>* Special categories of cited documents:</p> <p>“A” document defining the general state of the art which is not considered to be of particular relevance</p> <p>“E” earlier application or patent but published on or after the international filing date</p> <p>“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)</p> <p>“O” document referring to an oral disclosure, use, exhibition or other means</p> <p>“P” document published prior to the international filing date but later than the priority date claimed</p> <p>“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention</p> <p>“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone</p> <p>“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art</p> <p>“&” document member of the same patent family</p>		
Date of the actual completion of the international search 09 December 2019		Date of mailing of the international search report 19 December 2019
Name and mailing address of the ISA/EP European Patent Office p.b. 5818, Patentlaan 2, 2280 HV Rijswijk Netherlands Telephone No. (+31-70)340-2040 Facsimile No. (+31-70)340-3016		Authorized officer Böhmert, Jörg Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/EP2019/074941

C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	Ahmed Elgamil. "Templating your Dockerfile like a boss!" 29 December 2016 (2016-12-29), Retrieved from the Internet: https://blog.dockbit.com/templating-your-dockerfile-like-a-boss-2a84a67d28e9 [retrieved on 2019-04-09] XP055579167 page 3 - page 6	1-24
A	Mark Church. "Docker Reference Architecture: Designing Scalable, Portable Docker Container Networks" 20 June 2017 (2017-06-20), pages 1-37, Retrieved from the Internet: https://success.docker.com/api/articles/networking/pdf [retrieved on 2018-09-13] XP055507040 the whole document	1-24

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/EP2019/074941

Patent document cited in search report	Publication date (day/month/year)	Patent family member(s)	Publication date (day/month/year)
CN 108282376 A	13 July 2018	NONE	
US 2018287883 A1	04 October 2018	NONE	

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES INV. H04L12/24 G06F9/455 H04L29/08 ADD. H04L29/12		
Nach der Internationalen Patentklassifikation (IPC) oder nach der nationalen Klassifikation und der IPC		
B. RECHERCHIERTE GEBIETE		
Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole) H04L G06F		
Recherchierte, aber nicht zum Mindestprüfstoff gehörende Veröffentlichungen, soweit diese unter die recherchierten Gebiete fallen		
Während der internationalen Recherche konsultierte elektronische Datenbank (Name der Datenbank und evtl. verwendete Suchbegriffe) EPO-Internal, COMPENDEX, INSPEC, WPI Data		
C. ALS WESENTLICH ANGESEHENE UNTERLAGEN		
Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
X	VAN ROSSEM STEVEN ET AL: "NFV service dynamicity with a DevOps approach: Insights from a use-case realization", 2017 IFIP/IEEE SYMPOSIUM ON INTEGRATED NETWORK AND SERVICE MANAGEMENT (IM), IFIP, 8. Mai 2017 (2017-05-08), Seiten 674-679, XP033127640, DOI: 10.23919/INM.2017.7987357 [gefunden am 2017-07-20]	1-5, 7-16, 18-24
A	Kapitel I, II und IV	6,17
X	CN 108 282 376 A (UNIV JIANGNAN) 13. Juli 2018 (2018-07-13)	1-5, 7-16, 18-24
A	Absatz [0004] - Absatz [0011]	6,17
	----- -/-	
<input checked="" type="checkbox"/> Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen <input checked="" type="checkbox"/> Siehe Anhang Patentfamilie		
* Besondere Kategorien von angegebenen Veröffentlichungen : "A" Veröffentlichung, die den allgemeinen Stand der Technik definiert, aber nicht als besonders bedeutsam anzusehen ist "E" frühere Anmeldung oder Patent, die bzw. das jedoch erst am oder nach dem internationalen Anmeldedatum veröffentlicht worden ist "L" Veröffentlichung, die geeignet ist, einen Prioritätsanspruch zweifelhaft erscheinen zu lassen, oder durch die das Veröffentlichungsdatum einer anderen im Recherchenbericht genannten Veröffentlichung belegt werden soll oder die aus einem anderen besonderen Grund angegeben ist (wie ausgeführt) "O" Veröffentlichung, die sich auf eine mündliche Offenbarung, eine Benutzung, eine Ausstellung oder andere Maßnahmen bezieht "P" Veröffentlichung, die vor dem internationalen Anmeldedatum, aber nach dem beanspruchten Prioritätsdatum veröffentlicht worden ist "T" Spätere Veröffentlichung, die nach dem internationalen Anmeldedatum oder dem Prioritätsdatum veröffentlicht worden ist und mit der Anmeldung nicht kollidiert, sondern nur zum Verständnis des der Erfindung zugrundeliegenden Prinzips oder der ihr zugrundeliegenden Theorie angegeben ist "X" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann allein aufgrund dieser Veröffentlichung nicht als neu oder auf erfinderischer Tätigkeit beruhend betrachtet werden "Y" Veröffentlichung von besonderer Bedeutung; die beanspruchte Erfindung kann nicht als auf erfinderischer Tätigkeit beruhend betrachtet werden, wenn die Veröffentlichung mit einer oder mehreren Veröffentlichungen dieser Kategorie in Verbindung gebracht wird und diese Verbindung für einen Fachmann naheliegend ist "&" Veröffentlichung, die Mitglied derselben Patentfamilie ist		
Datum des Abschlusses der internationalen Recherche		Absenddatum des internationalen Recherchenberichts
9. Dezember 2019		19/12/2019
Name und Postanschrift der Internationalen Recherchenbehörde Europäisches Patentamt, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016		Bevollmächtigter Bediensteter Böhmert, Jörg

C. (Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN		
Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	US 2018/287883 A1 (JOSHI PRABODH [US] ET AL) 4. Oktober 2018 (2018-10-04) Absätze [0035] - [0040], [0048], [0056] -----	1-24
A	Ahmed Elgamil: "Templating your Dockerfile like a boss!", 29. Dezember 2016 (2016-12-29), XP055579167, Gefunden im Internet: URL: https://blog.dockbit.com/templating-your-dockerfile-like-a-boss-2a84a67d28e9 [gefunden am 2019-04-09] Seite 3 - Seite 6 -----	1-24
A	Mark Church: "Docker Reference Architecture: Designing Scalable, Portable Docker Container Networks", 20. Juni 2017 (2017-06-20), Seiten 1-37, XP055507040, Gefunden im Internet: URL: https://success.docker.com/api/articles/networking/pdf [gefunden am 2018-09-13] das ganze Dokument -----	1-24

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/EP2019/074941

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
CN 108282376	A	13-07-2018	KEINE

US 2018287883	A1	04-10-2018	KEINE
