(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
12. Dezember 2024 (12.12.2024)

WIPO | PCT

(10) Internationale Veröffentlichungsnummer
WO 2024/251587 A1

(51) Internationale Patentklassifikation:
*G06F 21/55* (2013.01)

(21) Internationales Aktenzeichen: PCT/EP2024/064795

(22) Internationales Anmeldedatum:
29. Mai 2024 (29.05.2024)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
23177485.2 06. Juni 2023 (06.06.2023) EP

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

(72) Erfinder: **FALK, Rainer**; Herderstr. 9, 85435 Erding (DE). **FEIST, Christian Peter**; Willibaldstr. 136, 80689 München (DE). **HORNUNG, Peter**; Schronfeld 71, 91054 Erlangen (DE). **MANTEL, Martin**; Konrad-Adenauer-Str. 7, 91325 Adelsdorf (DE). **PYKA, Stefan**; Burgerfeld 47 d, 85570 Markt Schwaben (DE). **SPERL, Franz**; Kastlstr. 5, 92526 Oberviechtach (DE). **ZESCHG, Thomas**; Winterstraße 5, 81543 München (DE).

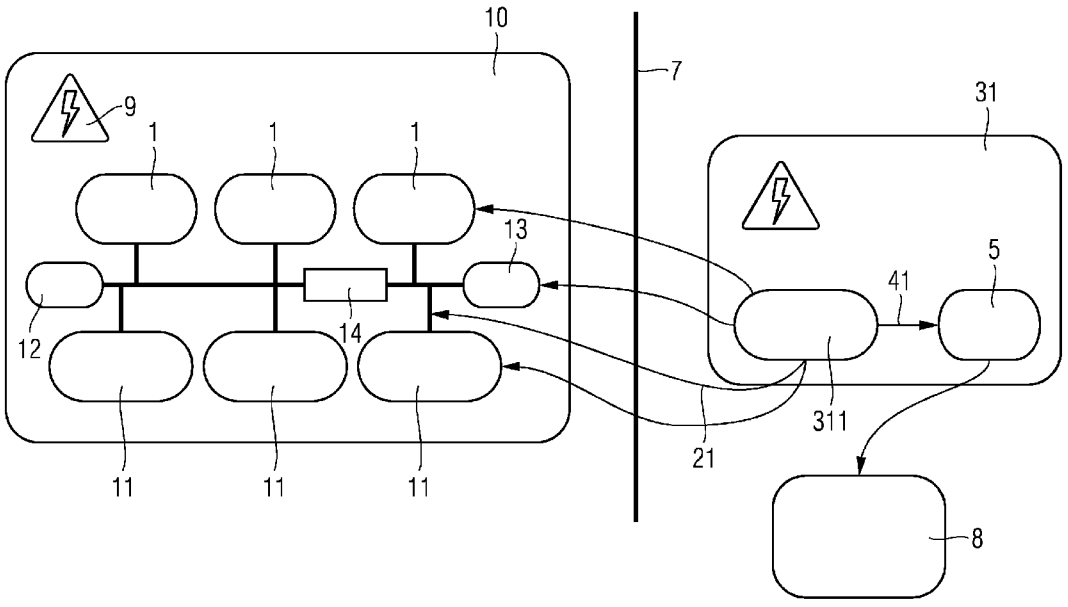
(74) Anwalt: **SIEMENS PATENT ATTORNEYS**; Postfach 22 16 34, 80506 München (DE).

(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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO,

(54) **Title:** UNIFIED MONITORING OF A PROGRAMMABLE SYSTEM

(54) **Bezeichnung:** EINHEITLICHES ÜBERWACHEN EINES PROGRAMMIERBAREN SYSTEMS

FIG 3




(57) **Abstract:** The invention relates to a monitoring unit (31) for monitoring a programmable system (10), said monitoring unit comprising: - a detection component designed to capture raw data regarding occurring events in the programmable system (10), the occurring events being generated by a plurality of software components (1) of the programmable system (10), - a derivation component designed to derive an application functionality of the plurality of software components (1) from the occurring events captured by the raw data, the application functionality being designed as a functionality for executing a main task for which the associated software component is intended, and - a generation component (311) designed to generate a log data record (21), the log data record (21) indicating the application functionality of the plurality of software components (1) of the programmable system (10). The invention also relates to a

WO 2024/251587 A1

*[Fortsetzung auf der nächsten Seite]*

JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

**(84) Bestimmungsstaaten** *(soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart)*: ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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)*

higher-level technical system and to an associated method.

**(57) Zusammenfassung:** Einheitliches Überwachen eines programmierbaren Systems Die Erfindung betrifft ein Überwachungseinheit (31) zum Überwachen eines programmierbaren Systems (10), aufweisend: - Eine Erfassungskomponente, ausgebildet Rohdaten bezüglich auftretender Ereignisse auf dem programmierbaren System (10) zu erfassen, wobei die auftretenden Ereignisse durch eine Vielzahl von Software-Komponenten (1) des programmierbaren Systems (10) erzeugt werden, eine Ableitungskomponente, ausgebildet jeweils eine Anwendungsfunktionalität der Vielzahl an Software-Komponenten (1) aus den durch die Rohdaten erfassten auftretenden Ereignisse abzuleiten, wobei die Anwendungsfunktionalität als eine Funktionalität zur Ausführung einer Hauptaufgabe, für welche die jeweilige Software-Komponente vorgesehen ist, ausgebildet ist, und - eine Generierungskomponente (311), ausgebildet einen Log-Datensatz (21) zu generieren, wobei der Log-Datensatz (21) die Anwendungsfunktionalität der Vielzahl von Software-Komponenten (1) des programmierbaren Systems (10) angibt. Außerdem betrifft die Erfindung ein übergeordnetes technisches System und ein zugehöriges Verfahren.

Beschreibung

Einheitliches Überwachen eines programmierbaren Systems

Unabhängig vom grammatikalischen Geschlecht eines bestimmten Begriffes sind Personen mit männlicher, weiblicher oder anderer Geschlechteridentität mit umfasst.

HINTERGRUND DER ERFINDUNG

Gebiet der Erfindung

Die vorliegende Erfindung betrifft eine Überwachungseinheit zum Überwachen eines programmierbaren Systems. Außerdem betrifft die Erfindung ein übergeordnetes technisches System und ein zugehöriges Verfahren.

Beschreibung des Stands der Technik

Industrielle Geräte zeichnen sich immer stärker durch eine zunehmende Flexibilisierung ihrer Funktionalität aus. Diese wird beispielsweise ermöglicht durch nachladbare Applikationen, mit denen die Gerätefunktionalität auch im Feld einfach und schnell angepasst werden kann. Zudem ist eine zunehmende Vernetzung von industriellen Geräten in Automatisierungsanlagen festzustellen. Diese genannten, aber auch weitere Aspekte erhöhen die Komplexität, die Fehleranfälligkeit und die Angreifbarkeit solcher industriellen Anlagen.

Um dieser Komplexität zu begegnen, wird oftmals ein Logging-System (= Logging-Komponente) installiert, welches Daten von möglichst vielen Komponenten aus der Anlage sammelt (beispielsweise in einer Log-Datenbank), aggregiert und den Anwendern für Auswertungen zur Verfügung stellt. Diese Auswertungen können dann dazu dienen, Fehler im Gesamtsystem zu finden, aber auch, erfolgte oder versuchte Angriffe und mögliche Schwachstellen zu entdecken und Gegenmaßnahmen einzuleiten.

Allerdings ist das Verfahren zum Sammeln der Daten nicht durchgängig auf verlässliche Art und Weise implementiert. Bisher werden Logdaten jeweils durch die laufenden Applikationen generiert und mit Hilfe eines geräte-spezifischen Logging-Mechanismus in eine Log-Datenbank geschrieben. Diese können dann durch spezielle Anwendungen wie z.B. Intrusion Detection Systeme (IDS) ausgelesen und analysiert werden. Moderne IDS arbeiten zwar oft mit statischen Analysemethoden, verwenden aber auch heuristische Methoden. Grundvoraussetzung ist aber immer, dass dem IDS die für die Analyse notwendigen Daten durch Sensorik o.Ä. in Form von Logdaten bereitgestellt werden.

Die Bereitstellung von Daten durch laufende Applikationen für das Logging ist naturgemäß nicht verlässlich, diese Daten können entweder (teilweise) unterdrückt oder entweder schon bei der Erzeugung oder auch bei der Übertragung in die Log-Datenbank manipuliert werden. Letzteres ist insbesondere dann relevant, wenn das Schreiben der Log-Datenbank durch eine zentrale Komponente erfolgt, der vor dem Schreiben die jeweiligen Logdaten übergeben werden.

Die Art und die Anzahl der Logdaten hängen in der Regel von der Applikation oder der Systemkomponente ab, welche die Daten bereitstellt. Diese können unterdrückt werden oder von einem Angreifer manipuliert werden.

Wenn Apps von Drittanbietern bereitgestellt werden, muss in der Praxis davon ausgegangen werden, dass diese Log-Daten nicht auf eine einheitliche Art bereitstellen. Auch ist es möglich, dass manche Apps keine Logdaten bereitstellen, oder dass sie nur unzureichende Logdaten bereitstellen. Auch können manche Apps, insbesondere Apps von Drittanbietern, keine oder nicht ausreichende (z.B. hinsichtlich zu erfüllender Security-Standards) Logdaten bereitstellen.

Werden die Logdaten zentral von einer Logging-Komponente verarbeitet und in eine Log-Datenbank geschrieben, hängt die Verlässlichkeit des gesamten Logging-Mechanismus davon ab, wie diese Logging-Komponente ins Gesamtsystem integriert ist. Insbesondere wenn Logdaten von einem IDS (z.B. HIDS, NIDS, ...) analysiert und weiterverarbeitet werden, ist es wichtig, dass die Logdaten sowohl integer als auch vollständig sind.

Auch ist bekannt, die durch einen Benutzer auf einem System ausgeführten Aktionen zu loggen. Dabei können insbesondere auch Kommandos geloggt werden, die mit erhöhten Privilegien ausgeführt wurden (z.B. mittels sudo bei Linux-basierten Systemen). Eine Erweiterung dieses Ansatzes umfasst das Loggen von ausgeführten Kommandos eines Benutzers (z.B. Administrator), nachdem dieser eine Session mit erhöhten Privilegien gestartet hat. Dies ermöglicht eine Zuordnung von als root (UID=0) ausgeführten Aktionen zu einem interaktiven Benutzer (z.B. bestimmtes Administratorkonto).

Von Windows und Linux sind Auditing-Subsysteme bekannt. Bei Linux kann z.B. auditd verwendet werden, um Zugriffsereignisse (z.B. Schreibzugriff auf eine Datei durch eine Applikation) zu erkennen und zu melden. Diese werden anschließend in einer Logdatei, z.B. /var/log/audit/audit.log, hinterlegt. Unter Windows kann die Audit Policy konfiguriert werden, um z.B. Änderungen an bestimmten Dateien zu protokollieren. Diese Ereignisse werden dann unter bestimmten IDs (ID: Identifier) im Windows Event Log abgespeichert, wo sie betrachtet und ausgewertet werden können.

Auch sind Security Information and Event Management Systeme (SIEM-Systeme) bekannt, um Security-Events zu loggen und auch auszuwerten. Dadurch können laufende Angriffe erkannt werden.

Auch sind Intrusion Detection Systeme (IDS) bekannt, die Angriffe erkennen. Diese können die Konfiguration auf Hosts (HIDS, host-based intrusion detection system) oder die Netzwerkkommunikation (NIDS, network-based intrusion detection

system) analysieren. Ein IDS kann allgemein eine Änderung des Verhaltens (Anomaly-based IDS) oder ein bekanntes Angriffsmuster (Signature-based IDS) als Angriff erkennen.

Auch sind Lösungen für ein File Integrity Monitoring (FIM) bekannt, die Änderungen an einem Dateisystem erkennen.

Die Aufgabe der Erfindung besteht darin, eine Lösung für eine verbesserte Überwachung eines programmierbaren Systems anzugeben.

ZUSAMMENFASSUNG DER ERFINDUNG

Die Erfindung ergibt sich aus den Merkmalen der unabhängigen Ansprüche. Vorteilhafte Weiterbildungen und Ausgestaltungen sind Gegenstand der abhängigen Ansprüche. Ausgestaltungen, Anwendungsmöglichkeiten und Vorteile der Erfindung ergeben sich aus der nachfolgenden Beschreibung und den Zeichnungen.

Die Erfindung betrifft eine Überwachungseinheit zum Überwachen eines programmierbaren Systems,
aufweisend:

- Eine Erfassungskomponente, ausgebildet Rohdaten bezüglich auftretender Ereignisse auf dem programmierbaren System zu erfassen, wobei die auftretenden Ereignisse durch eine Vielzahl von Software-Komponenten des programmierbaren Systems erzeugt werden,

- eine Ableitungskomponente, ausgebildet jeweils eine Anwendungsfunktionalität der Vielzahl an Software-Komponenten aus den durch die Rohdaten erfassten auftretenden Ereignisse abzuleiten,
wobei die Anwendungsfunktionalität als eine Funktionalität zur Ausführung einer Hauptaufgabe, für welche die jeweilige Software-Komponente vorgesehen ist, ausgebildet ist, und

- eine Generierungskomponente, ausgebildet einen Log-Datensatz, insbesondere umfassend ein Datum oder mehrere Daten, zu generieren, wobei der Log-Datensatz die Anwendungsfunktionalität der Vielzahl von Software-Komponenten des programmierbaren Systems angibt.

Das programmierbare System ist insbesondere ausgebildet als:
- ein industrielles System und/oder
- ein Automatisierungssystem und/oder
- eine programmierbare Anlage, insbesondere Produktionsanlage.

Das programmierbare System weist insbesondere auf:
- industrielle Geräte und/oder
- app-fähige Geräte und/oder
- programmierbare Geräte.

Die auftretenden Ereignisse durch die Vielzahl von Software-Komponenten sind zu verstehen als tatsächlich auftretende Ereignisse und/oder tatsächlich durchgeführte Ereignisse. Die auftretenden Ereignisse haben insbesondere Auswirkungen auf die Vielzahl von Software-Komponenten, System-Komponenten, RAM-Arbeitsspeicher, Node Version Manager (NVM) und/oder internen Bus des programmierbaren Systems.

Der generierte Log-Datensatz umfasst vorzugsweise eine Identifikationsinformation des programmierbaren Systems. Weiterhin kann der generierte Log-Datensatz die erkannte Anwendungsfunktionalität angeben. Außerdem kann bei der Ableitung der Anwendungsfunktionalität durch die Ableitungskomponente ermittelt werden, welche Software-Komponenten bei der erkannten Anwendungsfunktionalität involviert sind. Die Information zu den bei der erkannten Anwendungsfunktionalität involvierten Softwarekomponenten kann ebenfalls in dem Log-Datensatz umfasst sein. Weiterhin kann der Log-Datensatz eine Zeitstempel-Information umfassen, die abhängig vom Zeitpunkt oder Zeitraums des Auftretens der Rohdaten ermittelt wird.

Reine Anwendungsdaten, die die Applikation zur Laufzeit generiert, (d.h. innere Daten einer App, zum Beispiel, das, was Algorithmen der App machen; wie die App ihre Aufgaben umsetzt; das was innerhalb der App wie abläuft; also alles, das über den nach außen sichtbaren Input und Output der App hinausgeht), kann die Überwachungseinheit natürlich nicht erfassen oder rekonstruieren.

Die Ableitungskomponente ist ausgebildet jeweils die Anwendungsfunktionalität der Vielzahl an Software-Komponenten aus den durch die Rohdaten erfassten auftretenden Ereignisse abzuleiten. Die Anwendungsfunktionalität wird somit nicht gemessen, sondern abgeleitet. „Abgeleitet" ist auch zu verstehen als „ermittelt", „synthetisiert", „aus den Rohdaten geschlussfolgert" und/oder „aus den Rohdaten geschlossen". Die Anwendungsfunktionalität gibt somit an, was von die Ableitungskomponente erwartet wird, welche Anwendungsfunktionalität stattgefunden hat/ausgeführt wurde. Die Anwendungsfunktionalität der Vielzahl an Software-Komponenten ist somit insbesondere zu verstehen als eine Schätzung und/oder ein Erwartungswert der Anwendungsfunktionalität der Vielzahl von Software-Komponenten; u.a. unter Anwendung von Regeln und/oder eine künstliche Intelligenz (KI).

Die Anwendungsfunktionalität ist als eine Funktionalität zur Ausführung einer Hauptaufgabe, für welche die jeweilige Software-Komponente vorgesehen ist, ausgebildet. Die Hauptaufgabe ist dabei zu verstehen, als das, was die Software-Komponente in erster Linie umsetzen (auch „ausführen" oder „erfüllen") soll, d.h. für was sie ausgebildet ist zu machen. „In erster Linie" bedeutet, dabei, dass falls eine Software-Komponente mehrere Aufgaben hat, die vorrangige Aufgabe, die Hauptaufgabe ist. Durch die Umsetzung, auch Ausführung und/oder Erfüllung, ihrer Hauptaufgabe erfüllt die Software-Komponente einen Zweck. Der Zweck ist auch als ein Ziel zu verstehen, dass durch die Umsetzung der Hauptaufgabe erreicht wird. Für die Umsetzung ihrer Hauptaufgabe benötigt die Software-Komponente die genannte Funktionalität.

Die Anwendungsfunktionalität ist dabei eine übergeordnete Funktionalität, auch als Hauptaufgabe und/oder Zweck und/oder Aufgabe der jeweiligen Software-Komponenten der Vielzahl an Software-Komponenten zu verstehen. Die Anwendungsfunktionalität ist somit die Funktionalität, für die die jeweilige Software-Komponente in erster Linie vorgesehen ist. Insbesondere ist die Anwendungsfunktionalität eine Steuerfunktion bei einer Steuer-Software-Komponente, eine Produktionsfunktion bei einer Produktion-Software-Komponente und/oder eine Fräsfunktion bei einer Fräs-Software-Komponenten. Wenn z.B. eine App eine Steuerungsfunktionalität für eine Maschine realisiert (virtualisierte PLC, virtualisiertes Steuergerät), dann kann der Log-Datensatz eine Maschinensteuerfunktionalität betreffen, z.B. Maschine in Standby, Hochlauf, Betrieb, Maschine in Wartungsmodus, Fehlermodus, ein Stück produziert, ein Stück Ausschuss etc.

Die Software-Komponenten sind insbesondere ausgebildet als:
- Apps, auch als Applikationen und/oder Anwendungsprogramme bezeichenbar, und/oder
- Software-Programme,

jeweils insbesondere von Drittanbietern stammend und dadurch auf das programmierbare System nach Auslieferung durch den Hersteller nachladbar.

Ein Aspekt der Erfindung besteht darin, einen Logging-Mechanismus bereitzustellen, welcher zum einen zuverlässig und unabhängig von der Logik der einzelnen Software-Komponenten (Applikationen) den Log-Datensatz generiert, um die Abhängigkeit von dem durch eine Software-Komponente selbst bereitgestellten Log-Datensatz zu reduzieren. Des Weiteren kann dieser Logging-Mechanismus so ausgelegt sein, der generierte Log-Datensatz zuverlässig in eine Log-Datenbank für die weitere Auswertung durch einen Anwender bzw. durch ein IDS schreiben zu können. Dazu kann er den Log-Datensatz z.B. lokal bis zur erfolgten Übertragung zwischenspeichern oder durch eine kryptographische Prüfsumme schützen.

Hierfür wird eine Überwachungseinheit, insbesondere ein „Trusted Observer" (TO), vorgeschlagen, der system-weit, zentral, unabhängig und vertrauensvoll auftretende Ereignisse erfasst, indem er die laufenden Software-Komponenten, Systemkomponenten oder anderweitiges Systemverhalten beobachtet, selbständig aus seinen Beobachtungen den Log-Datensatz generiert, und diesen insbesondere in eine Log-Datenbank überträgt.

Dies bedeutet, dass nicht mehr die Software-Komponenten selbst die Logdaten erzeugen, sammeln und/oder in die Datenbank schreiben müssen – dies kann zusätzlich zu der Idee der Erfindung weiterhin geschehen, ist aber nicht obligatorisch. Durch die vorgestellte Überwachungseinheit ist es nicht mehr notwendig, dass eine Software-Komponente den auswertbaren Log-Datensatz erzeugt, sondern der Log-Datensatz wird in der Überwachungseinheit erzeugt. Dieser Log-Datensatz kann als synthetischer Log-Datensatz betrachtet werden, da er nicht die eigene Funktionalität der Überwachungseinheit angibt, sondern eine Fremdfunktionalität synthetisiert. Der Log-Datensatz wird somit nicht von der Ereignis-erzeugenden Komponente selbst erstellt.

In einer Weiterbildung der Erfindung sind die auftretenden Ereignisse ausgebildet als:

- ein Verhalten des programmierbaren Systems und/oder
- eine Reaktion des programmierbaren Systems und/oder
- eine Aktion auf dem programmierbaren System und/oder
- eine Auswirkung auf das programmierbare System.

Die genannten Optionen des/auf dem/des/auf das programmierbare System, gelten insbesondere für weitere Software-Komponenten (Apps) des programmierbaren Systems und/oder Komponenten des programmierbaren Systems und/oder Ressourcen des programmierbaren Systems.

Somit werden insbesondere das Laufzeitverhalten einer Applikation oder einer Systemkomponente, welche Auswirkungen auf Systemressourcen (CPU, Speicher, interne Bussysteme, Netzwerkschnittstellen, ...) hat, durch die Überwachungseinheit und deren Erfassungskomponente erfasst.

In einer weiteren Weiterbildung der Erfindung weisen die auftretenden Ereignisse auf:

- einen Zugriff auf einen Speicher und/oder
- eine Konfiguration eines Speichers und/oder
- eine Verwendung eines Datenbusses und/oder
- einen Aufruf eines Betriebssystems und/oder
- einen Zugriff auf eine Ein- und/oder Ausgabekomponente und/oder
- einen Zugriff auf eine Datei und/oder
- einen Zugriff auf das programmierbare System von außerhalb des programmierbaren Systems und/oder
- einen Zugriff auf das programmierbare System über einen Seitenkanal und/oder
- einen Aufbau und/oder ein Nutzen und/oder ein Beenden einer Netzwerkverbindung und/oder
- einen Aufbau und/oder ein Nutzen und/oder ein Beenden einer Kommunikationsverbindung.

Die auftretenden Ereignisse sind auch als Low-Level-Events zu verstehen.

Die auftretenden Ereignisse sind im Konkreten ausgebildet als:

- Zugriffe auf bestimmte System-Speicherbereiche ohne DMA oder mit DMA (CPU-Register, RAM, interner Flash, externer Flash, ...)
- System-interne Datenbusse
- Zugriffe auf das System über Seitenkanäle (z.B. Spannungsimpulse, Temperatur, ...)
- Scheduling des Systems, z.B. über das Laufzeitverhalten von Apps
- Konfiguration der MMU, MPU, ...

- Alle Komponenten, die auch in herkömmlichen Systemen Logdaten generieren wie z.B. laufende Applikationen
- der Aufbau, die Nutzung und/oder das Beenden einer Kommunikationsverbindung,
- die Nutzung einer Interprozess-Kommunikation,
- ein direkter Speicherzugriff (DMA),
- der Zugriff auf Ein-/Ausgabe-Komponenten (z.B. Sensoren, Aktuatoren),
- das Ändern von Dateien in Ordnern, die für Konfigurationsdateien von Applikationen vorgesehen sind (z.B. auf dem Host, oder ein dedizierter Ordner für ein proprietäres App-Format).

In einer weiteren Weiterbildung der Erfindung ist die Anwendungsfunktionalität ausgebildet als:
- eine übergeordnete Funktionalität und/oder
ein Zweck der jeweiligen Software-Komponenten der Vielzahl an Software-Komponenten.

Die Anwendungsfunktionalität ist dabei eine übergeordnete Funktionalität, auch als Hauptaufgabe und/oder Zweck und/oder Aufgabe der jeweiligen Software-Komponenten der Vielzahl an Software-Komponenten zu verstehen. Die Anwendungsfunktionalität ist somit die Funktionalität, für die die jeweilige Software-Komponente in erster Linie vorgesehen ist. Insbesondere ist die Anwendungsfunktionalität eine Steuerfunktion bei einer Steuer-Software-Komponente, eine Produktionsfunktion bei einer Produktion-Software-Komponente und/oder eine Fräsfunktion bei einer Fräs-Software-Komponenten. Wenn z.B. eine App eine Steuerungsfunktionalität für eine Maschine realisiert (virtualisierte PLC, virtualisiertes Steuergerät), dann kann der Log-Datensatz eine Maschinensteuerfunktionalität betreffen, z.B. Maschine in Standby, Hochlauf, Betrieb, Maschine in Wartungsmodus, Fehlermodus, ein Stück produziert, ein Stück Ausschuss etc.

In einer weiteren Weiterbildung der Erfindung ist die Generierungskomponente außerdem ausgebildet, den Log-Datensatz

und die im Log-Datensatz enthaltenen Daten in einem einheitlichen Format zu generieren.

Dies bedeutet, dass die Log-Daten des Log-Datensatzes ein einheitliches Format aufweisen. Das einheitliche Format ist auch als ein gleiches Format zu verstehen. Das einheitliche Format hat den Vorteil, dass die Daten untereinander vergleichbar sind und/oder auf die gleiche Art weiterverarbeitet werden können.

Im Stand der Technik ist das Verfahren zum Sammeln vom Log-Daten hingegen nicht auf verlässliche Art und Weise implementiert. Die Art und die Anzahl der Log-Daten hängen in der Regel von der Applikation oder der Systemkomponente ab, welche die Log-Daten bereitstellt. So können die Log-Daten auch unterdrückt werden oder von einem Angreifer manipuliert werden. Wenn Apps von Drittanbietern bereitgestellt werden, muss in der aktuellen Praxis davon ausgegangen werden, dass die Drittanbieter Log-Daten nicht auf eine einheitliche Art bereitstellen. Auch ist es möglich, dass manche Apps gar keine Log-Daten bereitstellen, oder dass sie nur unzureichende Log-Daten bereitstellen.

Durch diese Ausführungsform der Erfindung wird hingegen erreicht, dass auch auf offenen Plattformen, auf denen Apps mit unterschiedlichem Ursprung ausgeführt werden, verlässlich und einheitlich Log-Daten für alle ausgeführten Apps vorliegen.

In einer weiteren Weiterbildung der Erfindung ist die Erfassungskomponente außerdem ausgebildet auf das programmierbare System im Gesamten zuzugreifen, um die Rohdaten zu erfassen.

Dies bedeutet, dass die Erfassungskomponente auf das gesamte programmierbare System Zugriff hat und Rohdaten des gesamten programmierbaren Systems erfasst. Die Erfassungskomponente hat somit auf alle auf dem programmierbaren System vorliegenden Komponenten, insbesondere Datenbusse, Memory-Controller, Scheduling-Informationen, Zugriff. Dies hat den Vorteil, dass

sich keine Software-Komponente der Beobachtung / Rohdatenerfassung durch die Erfassungskomponente entziehen kann, da sie andernfalls nicht mehr ihre zugedachte Funktionalität ausführen könnte, denn ihre zugedachte Funktionalität wird immer zu auftretenden Ereignissen auf dem programmierbaren System führen. Die Überwachungseinheit würde diesen Entzug erkennen und dies insbesondere als auftretendes Ereignis in den Rohdaten erfassen und insbesondere als ein sicherheitsrelevantes Ereignis (security-relevantes Event) melden.

In einer weiteren Weiterbildung der Erfindung weist die Überwachungseinheit außerdem einen Integritätsschutz auf. Der Integritätsschutz schützt die Überwachungseinheit vor Manipulation.

Die Integration der Überwachungseinheit in das programmierbare System (Gesamtsystem) kann so gestaltet sein, dass die Überwachungseinheit nicht deaktiviert oder umgangen werden kann. Außerdem ist die Überwachungseinheit und deren Konfiguration nicht einfach manipulierbar. Der Integritätsschutz hat den Vorteil, dass dadurch umsetzbar ist, dass nur vertrauenswürdiges Personal Zugriff auf die Überwachungseinheit hat und diese konfigurieren oder deaktivieren kann. Zusätzlich ist eine physikalische Separierung der Überwachungseinheit im programmierbaren System von Vorteil. So ist eine zuverlässige Erfassung der Rohdaten gewährleistet.

In einer weiteren Weiterbildung der Erfindung ist die Generierungskomponente außerdem ausgebildet, den Log-Datensatz kryptographisch zu schützen.

Der Log-Datensatz weist somit einen kryptographischen Schutz auf. Dies hat den Vorteil, dass der Log-Datensatz nicht einfach manipulierbar ist.

In einer weiteren Weiterbildung der Erfindung besteht zwischen der Überwachungseinheit und der Vielzahl der Software-Komponenten eine logische Abhängigkeit.

Die logische Abhängigkeit besteht insbesondere zwischen der Funktion von Software-Komponenten und der Konfiguration der Überwachungseinheit. Insbesondere wird abhängig von der konkreten Konfiguration der Überwachungseinheit Schlüsselmaterial generiert, das dann von den Software-Komponenten verwendet wird. Eine Re-Konfiguration und/oder Deaktivierung der Überwachungseinheit ist dadurch nicht möglich. Die Überwachungseinheit ist insbesondere Teil einer App-Laufzeitumgebung, die API-Aufrufe der ausgeführten Apps überwacht.

In einer weiteren Weiterbildung der Erfindung weist die Überwachungseinheit außerdem auf:

- eine Bindungskomponente, ausgebildet kryptographisches Schlüsselmaterial bereitzustellen, wobei die Verwendung des kryptographisches Schlüsselmaterials für die Vielzahl der Software-Komponenten obligatorisch zur Ausführung derer zugehörigen Anwendungsfunktionalität ist.

Durch die Bindungskomponente wird eine Bindung zwischen der Vielzahl der Software-Komponenten und der Überwachungseinheit erzeugt. Die zugehörige Anwendungsfunktionalität der Vielzahl der Software-Komponenten ist auch als deren vorgesehene Anwendungsfunktionalität zu verstehen und korrespondiert insbesondere mit der Anwendungsfunktionalität, welche durch den Log-Datensatz angegeben ist und durch die Ableitungskomponente aus den auftretenden Ereignissen abgeleitet wird. Durch die Ausführung der zugehörigen Anwendungsfunktionalität werden die auftretenden Ereignisse erzeugt, welche wiederrum von der Erfassungskomponente in Form der Rohdaten erfasst werden.

In einer weiteren Weiterbildung der Erfindung weist die Überwachungseinheit außerdem auf:

- eine Datenbank, ausgebildet den Log-Datensatz zu speichern.

Bevorzugt sind die Überwachungseinheit und die Datenbank vertrauenswürdig miteinander verbunden. Manipulationen an der

Datenbank sollten nicht möglich sein. Bevorzugt ist die Log-Datenbank deshalb ein Bestandteil der Überwachungseinheit.

Die Datenbank kann so gestaltet sein, dass sie vom beobachteten programmierbaren System aus lesbar ist, es ist allerdings auch vorstellbar, dass die Datenbank nur von der Überwachungseinheit aus lesbar ist.

Bevorzugt ist die Datenbank außerdem vom beobachteten programmierbaren System getrennt, wodurch auch ein Schreiben und Speichern des Log-Datensatzes zuverlässig ist.

In einer weiteren Weiterbildung der Erfindung weist die Überwachungseinheit außerdem auf:

- eine Analysekomponente, ausgebildet den Log-Datensatz hinsichtlich sicherheitsrelevanter Ereignisse zu analysieren (d.h. auszuwerten).

Diese Analyse dient dazu, Fehler im programmierbaren System zu finden, aber auch, erfolgte Angriffe und mögliche Schwachstellen zu entdecken und Gegenmaßnahmen einzuleiten. Von der Analysekomponente werden hierfür insbesondere spezielle Anwendungen wie z.B. Intrusion Detection Systeme (IDS (z.B. HIDS, NIDS, ...)) oder SIEM (Security information and event management) verwendet. Eine Überwachungseinheit ist mit dieser Eigenschaft nicht als Alternative zu einem IDS zu sehen, sondern als Einheit, die für ein IDS in zuverlässiger Weise auswertbare Daten zur Verfügung stellt.

Die Erfindung umfasst außerdem ein technisches System, aufweisend:

- Überwachungseinheit nach einem der vorherigen Ansprüche, und
- ein programmierbares System, aufweisend:
  - eine Laufzeitumgebung für die Vielzahl von Software-Komponenten.

Die Überwachungseinheit ist hierbei bevorzugt extern des programmierbaren Systems angeordnet. Damit ist die Überwachungseinheit unabhängig von dem programmierbaren System und vor Manipulation geschützt.

Die Erfindung umfasst außerdem ein Verfahren zum Generieren eines Log-Datensates,
mit den Schritten:

- Ein Erfassen von Rohdaten bezüglich auftretender Ereignisse auf einem programmierbaren System, wobei die auftretenden Ereignisse durch eine Vielzahl von Software-Komponenten des programmierbaren Systems erzeugt werden, und wobei die auftretenden Ereignisse insbesondere Auswirkungen auf die Vielzahl von Software-Komponenten, System-Komponenten, RAM-Arbeitsspeicher, Node Version Manager (NVM) und/oder internen Bus des programmierbaren Systems haben,

- ein Ableiten jeweils einer Anwendungsfunktionalität der Vielzahl an Software-Komponenten aus den durch die Rohdaten erfassten auftretenden Ereignisse,
wobei die Anwendungsfunktionalität als eine Funktionalität zur Ausführung einer Hauptaufgabe, für welche die jeweilige Software-Komponente vorgesehen ist, ausgebildet ist, und

- das Generieren des Log-Datensatzes, wobei der Log-Datensatz die Anwendungsfunktionalität der Vielzahl von Software-Komponenten des programmierbaren Systems angibt.

KURZE BESCHREIBUNG DER ZEICHNUNGEN

Die Besonderheiten und Vorteile der Erfindung werden aus den nachfolgenden Erläuterungen mehrerer Ausführungsbeispiele anhand der schematischen Zeichnungen ersichtlich.

Es zeigen

Fig. 1 ein Ablaufdiagramm und Komponenten nach dem Stand der Technik,

Fig. 2 ein Ablaufdiagramm und Komponenten gemäß der Erfindung, und

Fig. 3 eine schematische Darstellung eines technischen Systems aufweisend erfindungsrelevante Komponenten.

DETAILLIERTE BESCHREIBUNG DER ERFINDUNG

Fig. 1 zeigt einen herkömmlichen Ansatz für das Logging: Applikationen 1 und/oder Software-Komponenten 1 selbst generieren und stellen Log-Datensätze 2 an einen Logging-Mechanismus 3 zu Verfügung. Der Logging-Mechanismus schreibt 4 die Log-Datensätze 2 in eine Log-Datenbank 5. Sowohl die Generierung 2 der Log-Datensätze als auch das Schreiben 4 der Daten kann hier angegriffen werden.

Fig. 2 zeigt einen Ansatz gemäß der Erfindung für das Logging: Applikationen 1 und/oder Software-Komponenten 1 werden von der Überwachungseinheit 31 überwacht. Die Überwachungseinheit 31 generiert die Log-Datensätze 21 und schreibt 41 die Log-Datensätze 21 in eine Log-Datenbank 5. Sowohl die Generierung der Log-Datensätze 21 als auch das Schreiben 41 der Log-Datensätze 21 unterliegt der Überwachungseinheit 31 und ist davor geschützt angegriffen zu werden.

Fig. 3 zeigt ein technisches System, aufweisend:

- ein programmierbares System 10, aufweisend:
  - eine Laufzeitumgebung für die Vielzahl von Software-Komponenten 1,
  - die Vielzahl von Software-Komponenten 1,
  - System-Komponenten 11
  - einen RAM-Arbeitsspeicher 12

- einen Node Version Manager (NVM) 13
- einen internen Bus 14
- eine eigene Stromversorgung 9.

- eine erfindungsgemäße Überwachungseinheit 31, aufweisend:
  - eine Erfassungskomponente 311, ausgebildet Rohdaten bezüglich auftretender Ereignisse auf dem programmierbaren System 10 zu erfassen, wobei die auftretenden Ereignisse durch eine Vielzahl von Software-Komponenten 1 des programmierbaren Systems 10 erzeugt werden,
    und wobei die auftretenden Ereignisse insbesondere Auswirkungen auf die Vielzahl von Software-Komponenten 1, System-Komponenten 11, RAM-Arbeitsspeicher 12, Node Version Manager (NVM) 13 und/oder internen Bus 14 des programmierbaren Systems 10 haben,
  - eine Ableitungskomponente (nicht gezeigt), ausgebildet jeweils eine Anwendungsfunktionalität der Vielzahl an Software-Komponenten 1 aus den durch die Rohdaten erfassten auftretenden Ereignisse abzuleiten, und
  - eine Generierungskomponente, ausgebildet Log-Datensätze 21 zu generieren, wobei die Log-Datensätze 21 die Anwendungsfunktionalität der Vielzahl von Software-Komponenten 2 des programmierbaren Systems 10 angeben,
  - eine Datenbank 5, ausgebildet die Log-Datensätze 41 zu speichern,
- eine Analysekomponente 8, ausgebildet die Log-Datensätze 41 hinsichtlich sicherheitsrelevanter Ereignisse zu analysieren.

Die Überwachungseinheit 31 ist durch eine physikalische Separierung 7 vom beobachteten programmierbaren System 10 getrennt. Die beobachteten Log-Datensätze schreibt 41 die Überwachungseinheit 31 in eine Log-Datenbank 5, die von einem entsprechenden Analyzer-Tool 8 ausgelesen werden können.

Obwohl die Erfindung im Detail durch die Ausführungsbeispiele näher illustriert und beschrieben wurde, ist die Erfindung durch die offenbarten Beispiele nicht eingeschränkt und andere Variationen können vom Fachmann daraus abgeleitet werden, ohne den Schutzumfang der Erfindung zu verlassen.

Patentansprüche

1. Überwachungseinheit (31) zum Überwachen eines programmierbaren Systems (10),
aufweisend:

- Eine Erfassungskomponente, ausgebildet Rohdaten bezüglich auftretender Ereignisse auf dem programmierbaren System (10) zu erfassen, wobei die auftretenden Ereignisse durch eine Vielzahl von Software-Komponenten (1) des programmierbaren Systems (10) erzeugt werden,

- eine Ableitungskomponente, ausgebildet jeweils eine Anwendungsfunktionalität der Vielzahl an Software-Komponenten (1) aus den durch die Rohdaten erfassten auftretenden Ereignisse abzuleiten,
wobei die Anwendungsfunktionalität als eine Funktionalität zur Ausführung einer Hauptaufgabe, für welche die jeweilige Software-Komponente vorgesehen ist, ausgebildet ist,
und

- eine Generierungskomponente (311), ausgebildet einen Log-Datensatz (21) zu generieren, wobei der Log-Datensatz (21) die Anwendungsfunktionalität der Vielzahl von Software-Komponenten (1) des programmierbaren Systems (10) angeben.

2. Überwachungseinheit (31) nach Anspruch 1,
wobei die auftretenden Ereignisse ausgebildet sind als:

- ein Verhalten des programmierbaren Systems (10) und/oder
- eine Reaktion des programmierbaren Systems (10) und/oder
- eine Aktion auf dem programmierbaren System (10) und/oder
- eine Auswirkung auf das programmierbare System (10).

3. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
wobei die auftretenden Ereignisse aufweisen:
- einen Zugriff auf einen Speicher (12) und/oder
- eine Konfiguration eines Speichers (12) und/oder
- eine Verwendung eines Datenbusses (14) und/oder
- einen Aufruf eines Betriebssystems und/oder
- einen Zugriff auf eine Ein- und/oder Ausgabekomponente und/oder
- einen Zugriff auf eine Datei und/oder
- einen Zugriff auf das programmierbare System (10) von außerhalb des programmierbaren Systems und/oder
- einen Zugriff auf das programmierbare System (10) über einen Seitenkanal und/oder
- einen Aufbau und/oder ein Nutzen und/oder ein Beenden einer Netzwerkverbindung und/oder
- einen Aufbau und/oder ein Nutzen und/oder ein Beenden einer Kommunikationsverbindung.

4. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
wobei die Anwendungsfunktionalität ausgebildet ist als:
- eine übergeordnete Funktionalität und/oder
- ein Zweck
der jeweiligen Software-Komponenten (1) der Vielzahl an Software-Komponenten (1).

5. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
wobei die Generierungskomponente (311) außerdem ausgebildet ist, den Logdatensatz und von ihm aufgewiesene Log-Daten (21) in einem einheitlichen Format zu generieren.

6. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
wobei die Erfassungskomponente außerdem ausgebildet ist auf das programmierbare System (10) im Gesamten zuzugreifen, um die Rohdaten zu erfassen.

7. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
außerdem einen Integritätsschutz aufweisend.

8. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
wobei die Generierungskomponente (311) außerdem ausgebildet ist, den Log-Datensatz (21) kryptographisch zu schützen.

9. Überwachungseinheit nach einem der vorhergehenden Ansprüche,
wobei zwischen der Überwachungseinheit (31) und der Vielzahl der Software-Komponenten (1) eine logische Abhängigkeit besteht.

10. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
außerdem aufweisend:
- eine Bindungskomponente, ausgebildet kryptographisches Schlüsselmaterial bereitzustellen, wobei die Verwendung des kryptographisches Schlüsselmaterials für die Vielzahl der Software-Komponenten obligatorisch zur Ausführung derer zugehörigen Anwendungsfunktionalität ist.

11. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
außerdem aufweisend:
- eine Datenbank (5), ausgebildet den Log-Datensatz (21) zu speichern.

12. Überwachungseinheit (31) nach einem der vorhergehenden Ansprüche,
außerdem aufweisend:
- eine Analysekomponente, ausgebildet den Log-Datensatz (21) hinsichtlich sicherheitsrelevanter Ereignisse zu analysieren.

13. Technisches System, aufweisend:
- Überwachungseinheit (31) nach einem der vorherigen Ansprüche, und
- ein programmierbares System (10), aufweisend:
- eine Laufzeitumgebung für die Vielzahl von Software-Komponenten (1).

14. Verfahren zum Generieren von einem Log-Datensatz (21), mit den Schritten:

- Ein Erfassen von Rohdaten bezüglich auftretender Ereignisse auf einem programmierbaren System (10), wobei die auftretenden Ereignisse durch eine Vielzahl von Software-Komponenten (1) des programmierbaren Systems (10) erzeugt werden,

- ein Ableiten jeweils einer Anwendungsfunktionalität der Vielzahl an Software-Komponenten (10) aus den durch die Rohdaten erfassten auftretenden Ereignisse,
wobei die Anwendungsfunktionalität als eine Funktionalität zur Ausführung einer Hauptaufgabe, für welche die jeweilige Software-Komponente vorgesehen ist, ausgebildet ist, und

- das Generieren des Log-Datensatzes (21), wobei der Log-Datensatz (21) die Anwendungsfunktionalität der Vielzahl von Software-Komponenten (1) des programmierbaren Systems (10) angeben.

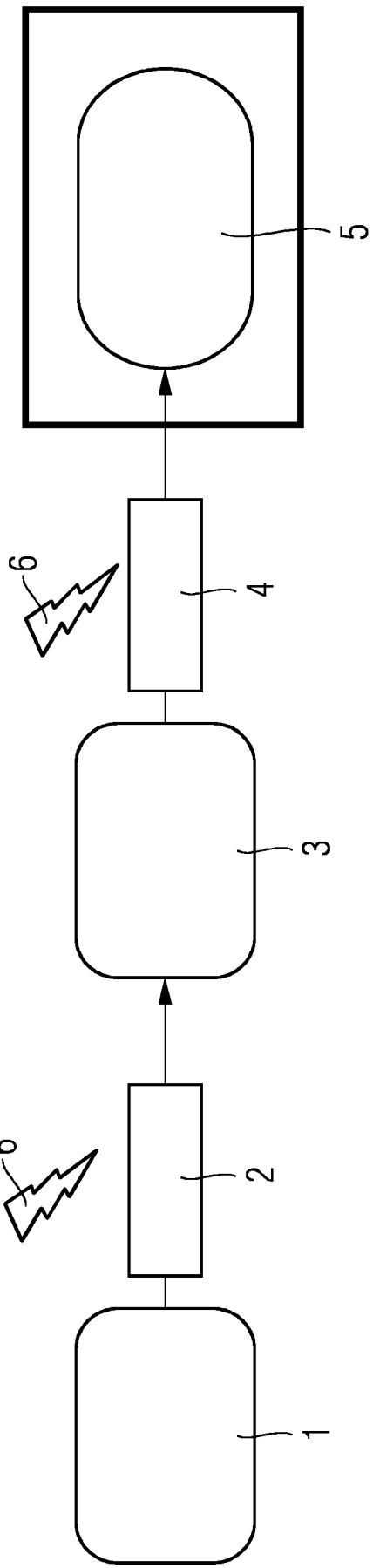



FIG 1

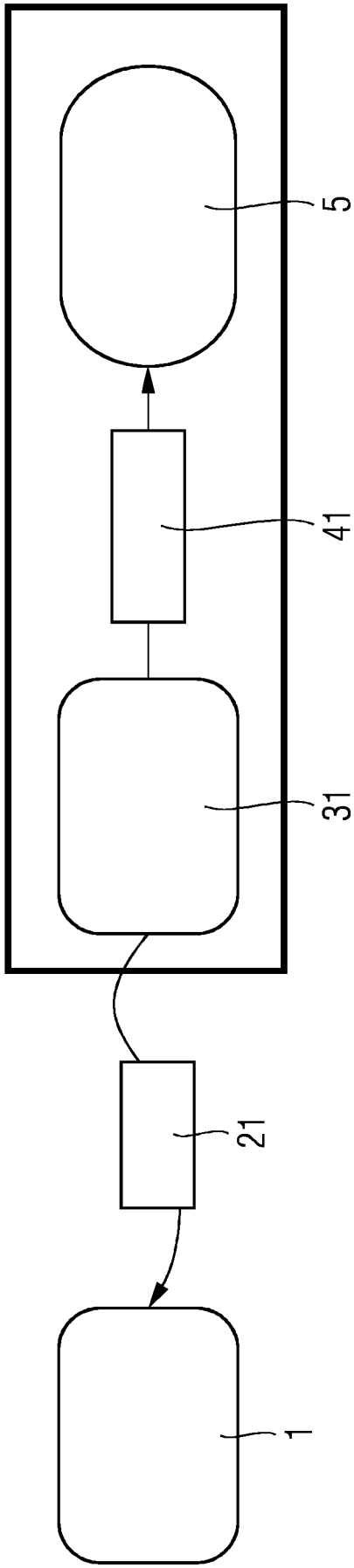



FIG 2

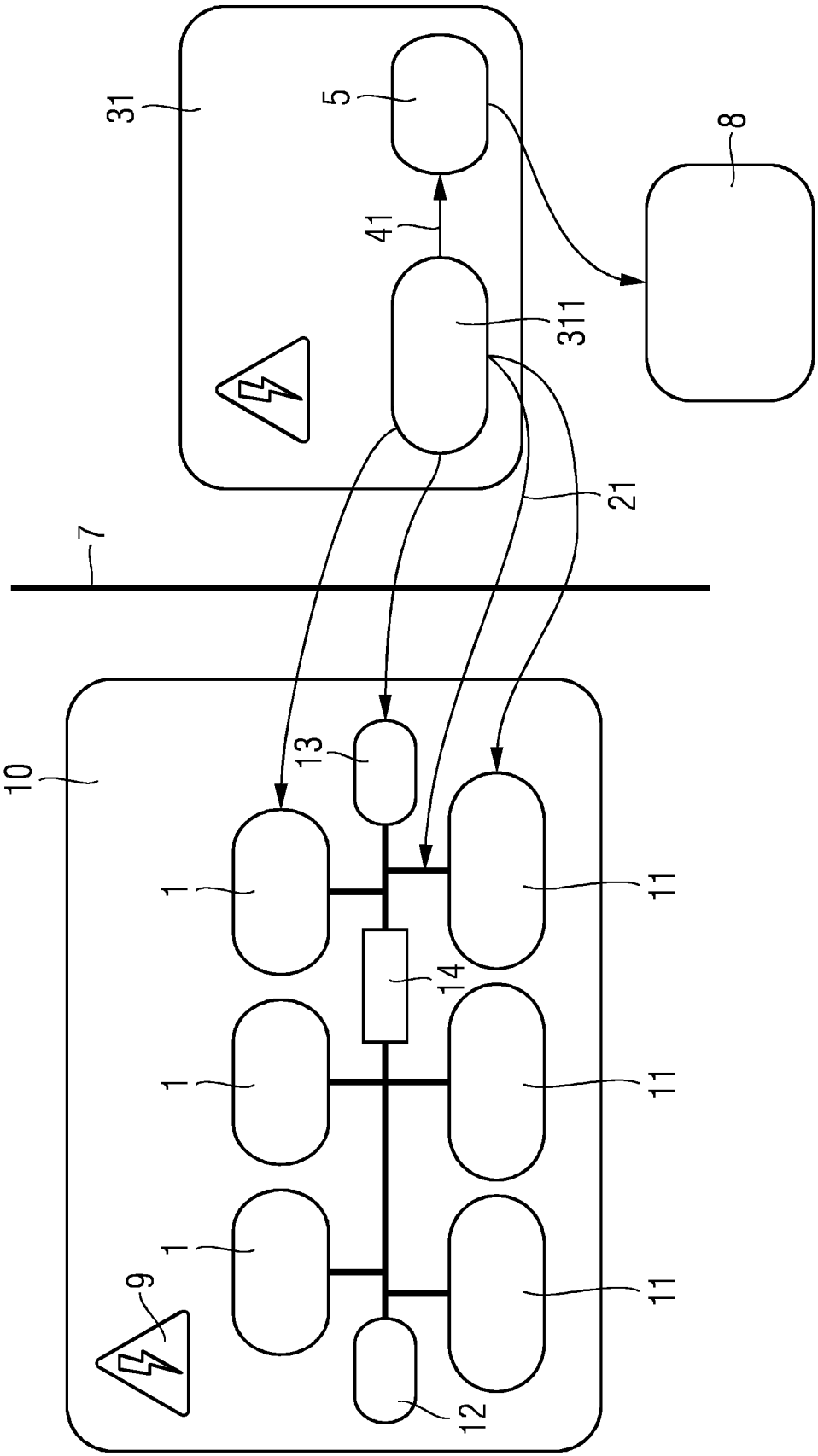



FIG 3

# INTERNATIONAL SEARCH REPORT

International application No.
**PCT/EP2024/064795**

## A. CLASSIFICATION OF SUBJECT MATTER

***G06F 21/55***(2013.01)i

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)

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

## C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | EP 2252038 A1 (NOVELL INC [US]) 17 November 2010 (2010-11-17)<br>abstract<br>paragraph [0005]<br>paragraph [0035] - paragraph [0043] | 1-14 |
| A | US 2002078381 A1 (FARLEY TIMOTHY P [US] ET AL) 20 June 2002 (2002-06-20)<br>abstract<br>paragraph [0066]<br>paragraph [0080] | 1-14 |
| A | US 2008127346 A1 (OH HYUNGGEUN [KR] ET AL) 29 May 2008 (2008-05-29)<br>abstract<br>paragraph [0013] | 1-14 |

☐ Further documents are listed in the continuation of Box C. ☑ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"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)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"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
"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
"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
"&" document member of the same patent family

| Date of the actual completion of the international search | Date of mailing of the international search report |
|---|---|
| **20 August 2024** | **30 August 2024** |

| Name and mailing address of the ISA/EP | Authorized officer |
|---|---|
| **European Patent Office**<br>**p.b. 5818, Patentlaan 2, 2280 HV Rijswijk**<br>**Netherlands (Kingdom of the)**<br>Telephone No. **(+31-70)340-2040**<br>Facsimile No. **(+31-70)340-3016** | **Mezödi, Stephan**<br>Telephone No. |

**INTERNATIONAL SEARCH REPORT**
**Information on patent family members**

International application No.
**PCT/EP2024/064795**

| Patent document cited in search report | Publication date (day/month/year) | Patent family member(s) | Publication date (day/month/year) |
|---|---|---|---|
| EP 2252038 A1 | 17 November 2010 | EP 2252038 A1 | 17 November 2010 |
| | | US 2010293128 A1 | 18 November 2010 |
| US 2002078381 A1 | 20 June 2002 | AU 6295801 A | 12 November 2001 |
| | | EP 1277326 A2 | 22 January 2003 |
| | | JP 4700884 B2 | 15 June 2011 |
| | | JP 2004537075 A | 09 December 2004 |
| | | US 2002078381 A1 | 20 June 2002 |
| | | WO 0184285 A2 | 08 November 2001 |
| US 2008127346 A1 | 29 May 2008 | NONE | |

# INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen
PCT/EP2024/064795

**A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES**
INV. G06F21/55
ADD.

Nach der Internationalen Patentklassifikation (IPC) oder nach der nationalen Klassifikation und der IPC

**B. RECHERCHIERTE GEBIETE**

Recherchierter Mindestprüfstoff (Klassifikationssystem und Klassifikationssymbole )
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

**C. ALS WESENTLICH ANGESEHENE UNTERLAGEN**

| Kategorie* | Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile | Betr. Anspruch Nr. |
|---|---|---|
| X | EP 2 252 038 A1 (NOVELL INC [US])<br>17. November 2010 (2010-11-17)<br>Zusammenfassung<br>Absatz [0005]<br>Absatz [0035] - Absatz [0043]<br>----- | 1-14 |
| A | US 2002/078381 A1 (FARLEY TIMOTHY P [US] ET AL) 20. Juni 2002 (2002-06-20)<br>Zusammenfassung<br>Absatz [0066]<br>Absatz [0080]<br>----- | 1-14 |
| A | US 2008/127346 A1 (OH HYUNGGEUN [KR] ET AL) 29. Mai 2008 (2008-05-29)<br>Zusammenfassung<br>Absatz [0013]<br>----- | 1-14 |

☐ Weitere Veröffentlichungen sind der Fortsetzung von Feld C zu entnehmen ☒ 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 | Absendedatum des internationalen Recherchenberichts |
|---|---|
| 20. August 2024 | 30/08/2024 |

| Name und Postanschrift der Internationalen Recherchenbehörde | Bevollmächtigter Bediensteter |
|---|---|
| Europäisches Patentamt, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | Mezödi, Stephan |



Formblatt PCT/ISA/210 (Blatt 2) (April 2005)

# INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen
PCT/EP2024/064795

| Im Recherchenbericht angeführtes Patentdokument | Datum der Veröffentlichung | Mitglied(er) der Patentfamilie | Datum der Veröffentlichung |
|---|---|---|---|
| EP 2252038 A1 | 17-11-2010 | EP 2252038 A1 | 17-11-2010 |
| | | US 2010293128 A1 | 18-11-2010 |
| US 2002078381 A1 | 20-06-2002 | AU 6295801 A | 12-11-2001 |
| | | EP 1277326 A2 | 22-01-2003 |
| | | JP 4700884 B2 | 15-06-2011 |
| | | JP 2004537075 A | 09-12-2004 |
| | | US 2002078381 A1 | 20-06-2002 |
| | | WO 0184285 A2 | 08-11-2001 |
| US 2008127346 A1 | 29-05-2008 | KEINE | |