(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
30. August 2012 (30.08.2012)

WIPO | PCT

(10) Internationale Veröffentlichungsnummer
WO 2012/113545 A2

(51) Internationale Patentklassifikation: Nicht klassifiziert

(21) Internationales Aktenzeichen: PCT/EP2012/000763

(22) Internationales Anmeldedatum:
22. Februar 2012 (22.02.2012)

(25) Einreichungssprache: Deutsch

(26) Veröffentlichungssprache: Deutsch

(30) Angaben zur Priorität:
10 2011 012 227.3
24. Februar 2011 (24.02.2011) DE

(71) Anmelder *(für alle Bestimmungsstaaten mit Ausnahme von US)*: **GIESECKE & DEVRIENT GMBH** [DE/DE]; Prinzregentenstraße 159, 81677 München (DE).

(72) Erfinder; und
(75) Erfinder/Anmelder *(nur für US)*: **SPITZ, Stephan** [DE/DE]; Parkstraße 18, 85757 Karlsfeld (DE).

(74) Gemeinsamer Vertreter: **GIESECKE & DEVRIENT GMBH**; Patent- und Lizenzabteilung, Prinzregentenstraße 159, 81677 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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, 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.

(84) Bestimmungsstaaten *(soweit nicht anders angegeben, für jede verfügbare regionale Schutzrechtsart)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), eurasisches (AM, AZ, BY, KG, KZ, MD, 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, ML, MR, NE, SN, TD, TG).

Erklärungen gemäß Regel 4.17:

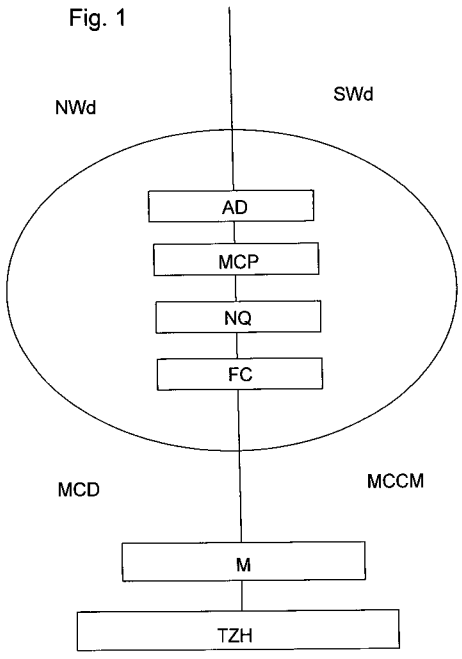
— *hinsichtlich der Berechtigung des Anmelders, ein Patent zu beantragen und zu erhalten (Regel 4.17 Ziffer ii)*

[Fortsetzung auf der nächsten Seite]

(54) Title: METHOD FOR INTERCHANGING DATA IN A SECURE RUNTIME ENVIRONMENT

(54) Bezeichnung : VERFAHREN ZUM DATENAUSTAUSCH IN EINER GESICHERTEN LAUFZEITUMGEBUNG

Fig. 1



(57) Abstract: The invention relates to a method for interchanging data between a secure runtime environment (SWd), in which a number of secure applications (TL) can be executed, and a non-secure environment (NWd) of a microprocessor unit (MP), in particular in a mobile terminal, in which application data (AD) and control data (MCP, NQ) are transmitted via different buffers.

(57) Zusammenfassung: Die Erfindung betrifft ein Verfahren zum Datenaustausch zwischen einer gesicherten Laufzeitumgebung (SWd), in welcher eine Anzahl an sicheren Applikationen (TL) ausführbar ist, und einer unsicheren Umgebung (NWd) einer Mikroprozessoreinheit (MP), insbesondere in einem mobilen Endgerät, bei dem die Übertragung von Applikationsdaten (AD) und Steuerdaten (MCP, NQ) über verschiedene Puffer erfolgt.

WO 2012/113545 A2

— *hinsichtlich der Berechtigung des Anmelders, die Priorität einer früheren Anmeldung zu beanspruchen (Regel 4.17 Ziffer iii)*

Veröffentlicht:

— *ohne internationalen Recherchenbericht und erneut zu veröffentlichen nach Erhalt des Berichts (Regel 48 Absatz 2 Buchstabe g)*

Verfahren zum Datenaustausch in einer gesicherten Laufzeitumgebung

Die Erfindung betrifft ein Verfahren zum Datenaustausch zwischen einer gesicherten Laufzeitumgebung, in welcher eine Anzahl an sicheren Applikationen ausführbar ist, und einer unsicheren Umgebung einer Mikroprozessoreinheit, insbesondere in einem mobilen Endgerät.

Gesicherte Laufzeitumgebungen sind aus dem Stand der Technik bekannt und ermöglichen eine gegen Angriffe geschützte Ausführung von Programmen mit einer Mikroprozessoreinheit. Unter Mikroprozessoreinheit ist dabei die Gesamtheit der zur Programmausführung verwendeten Hardware zu verstehen, insbesondere der eigentliche Mikroprozessor sowie entsprechende flüchtige und nicht-flüchtige Speicher, welche bei der Programmausführung zur Ablage von Daten verwendet werden.

Um den Sicherheitsanforderungen, den Speicherbeschränkungen und Kommunikationsmechanismen von Mikroprozessoreinheiten mit gesicherten Laufzeitumgebungen gerecht zu werden, ist es erforderlich, eine Multitasking-Eigenschaft zu optimieren.

Es ist Aufgabe der vorliegenden Erfindung, ein Verfahren zum Datenaustausch zwischen einer gesicherten Laufzeitumgebung und einer unsicheren Umgebung einer Mikroprozessoreinheit anzugeben, welches eine verbesserte Multitasking-Eigenschaft ermöglicht. Eine weitere Aufgabe der Erfindung besteht darin, eine Mikroprozessoreinheit anzugeben, welche verbesserte Multitasking-Eigenschaften aufweist.

Diese Aufgaben werden gelöst durch ein Verfahren gemäß den Merkmalen des Patentanspruches 1 sowie eine Mikroprozessoreinheit gemäß den Merk-

malen des Patentanspruches 13. Vorteilhafte Ausgestaltungen der Erfindung sind in den abhängigen Patentansprüchen angegeben.

Bei dem erfindungsgemäßen Verfahren von Datenaustausch zwischen einer gesicherten Laufzeitumgebung, in welcher eine Anzahl an sicheren Applikationen ausführbar ist, und einer unsicheren Umgebung einer Mikroprozessoreinheit, insbesondere in einem mobilen Endgerät, erfolgt die Übertragung von Applikationsdaten und Steuerdaten über verschiedene Puffer.

Hierdurch wird eine strikte Prozessisolierung ermöglicht, die das sichere Nachladen von Binär-Code ermöglicht. Weiterhin wird ein schnellerer Datenaustausch von Applikationen der unsicheren Umgebung mit Prozessen in der gesicherten Laufzeitumgebung ermöglicht.

Es ist bevorzugt, wenn die Steuerdaten unterschiedlicher Art über unterschiedliche Puffer übertragen werden. Ebenso ist es bevorzugt, wenn, die Umschaltung zwischen der gesicherten Laufzeitumgebung und der unsicheren Umgebung betreffende, Kontrolldaten über einen eigenen, sicheren Puffer übertragen werden. Anhand der Kontrolldaten kann ein Wechsel zwischen gesicherter Laufzeitumgebung und unsicherer Umgebung erfolgen. Hierdurch können insbesondere schnelle Kontextwechselzeiten erzielt werden, wodurch sich eine gute Performance bei einem Taskwechsel zwischen Prozessen ergibt.

In einer weiteren zweckmäßigen Ausgestaltung basiert die Übertragung der Applikationsdaten und der Steuerdaten sowie optional der Kontrolldaten auf einem ARM-Monitorcode, der in einer Monitor-Einheit implementiert ist, welche Schnittstellen zu der gesicherten Laufzeitumgebung und der unsicheren Umgebung aufweist.

Es ist weiter bevorzugt, wenn die Übertragung der Applikationsdaten und der Steuerdaten zwischen der gesicherten Laufzeitumgebung und einem Treiber der unsicheren Umgebung erfolgt. Zweckmäßigerweise legt ein in der unsicheren Umgebung, insbesondere in einer Schnittstelle des Treibers für die Steuerdaten, implementierter Scheduler fest, welche der sicheren Applikationen in der gesicherten Laufzeitumgebung zur Ausführung gebracht wird.

Es ist weiterhin zweckmäßig, wenn der Datenaustausch unter Nutzung eines, von der gesicherten Laufzeitumgebung und der unsicheren Umgebung les- und/oder beschreibbaren, Speicherbereichs eines Speichers erfolgt. Der Speicherbereich wird vorzugsweise durch eine Kontrollnachricht initialisiert. Insbesondere ist hierbei vorgesehen, dass die über Steuernachrichten gesicherte Laufzeitumgebung über für diese bestimmte Daten in dem Speicherbereich benachrichtigt wird. Dabei werden die Steuerdaten vorzugsweise mit einem eindeutigen Sitzungs-Kennzeichner (Session-ID) versehen, anhand dem die gesicherte Laufzeitumgebung die Steuernachricht einer der in der gesicherten Laufzeitumgebung ausgeführten Applikation zuordnen kann.

In einer weiteren vorteilhaften Ausgestaltung wird jedem in der gesicherten Laufzeitumgebung ablaufenden Prozess eine definierte, nicht überschreitbare Rechenzeit zugewiesen. Diese darf aus Sicherheitsgründen nicht überschritten werden. Hierdurch kann eine strikte Prozessisolierung erzielt werden.

Gemäß einer weiteren zweckmäßigen Ausgestaltung weist der in der gesicherten Laufzeitumgebung ablaufende Prozess nachfolgende Threadstruktur auf: Kennzeichner (ID) des Threads; aktueller Zustand des Threads; lokaler

Exception Handler der Thread; Priorität des Threads. Vorzugsweise weist ein jeweiliger Prozess nachfolgende Task-Struktur auf: Momentaner Zustand der Task; Task-Kennzeichner der Erzeuger-Task; externer Exception Handler der Task; Rechenzeitkontingent der Task; Anzahl der Threads, die die Task aktivieren oder bereitstellen kann; Priorität und Rechte der Task. Durch die beschriebenen Thread- und/oder Task-Strukturen müssen bei einem Kontextwechsel keine großen Datenmengen umkopiert werden. Hierdurch lassen sich schnelle Kontextwechselzeiten erzielen.

Die Erfindung schafft weiterhin eine Mikroprozessoreinheit mit einer gesicherten Laufzeitumgebung und einer unsicheren Umgebung, welche derart ausgestaltet ist, dass zum Datenaustausch zwischen der gesicherten Laufzeitumgebung und der unsicheren Umgebung die Übertragung von Applikationsdaten und Steuerdaten über verschiedene Puffer erfolgt. Der Begriff der Mikroprozessoreinheit ist dabei wiederum weit zu verstehen und umfasst alle für die Ausführung des Datenaustauschs notwendigen Hardware-Komponenten, z.B. einen tragbaren Datenträger und insbesondere eine Chip-Karte.

Die Erfindung betrifft darüber hinaus ein mobiles Endgerät, insbesondere ein Mobiltelefon, welches eine entsprechende Mikroprozessoreinheit beinhaltet.

Die Erfindung wird nachfolgend näher anhand von Ausführungsbeispielen in der Zeichnung erläutert. Es zeigen:

Fig. 1 eine schematische Darstellung des erfindungsgemäßen Verfahrens,

Fig. 2 eine schematische Darstellung der zur Verwirklichung des erfindungsgemäßen Verfahrens notwendigen Komponenten einer Mikroprozessoreinheit,

Fig. 3 eine schematische Darstellung, anhand der die Funktionsweise des erfindungsgemäßen Verfahrens erläutert wird, und

Fig. 4 eine schematische Darstellung eines Anwendungsbeispiels des erfindungsgemäßen Verfahrens.

Anhand von Fig. 1 wird der Datenaustausch zwischen einer gesicherten Laufzeitumgebung SWd und einer unsicheren Umgebung NWd einer Mikroprozessoreinheit beschrieben, welche in der Form einer sog. ARM-Trust-Zone ausgebildet ist. Die ARM-Trust-Zone stellt eine bekannte Technologie dar, mit der in einer Mikroprozessoreinheit ein geschützter Bereich generiert wird, der als gesicherte Laufzeitumgebung SWd zur Durchführung von als Trustlets bezeichneten Applikationen verwendet wird. Die gesicherte Laufzeitumgebung wird als „Secure World" bezeichnet, die unsichere Umgebung als „Normal World". In der hier beschriebenen Ausführungsform ist die ARM-Trust-Zone in einer Hardware-Plattform, der sog. Trust-Zone-Hardware, eines mobilen Endgerätes, bspw. einem Mobiltelefon, implementiert. Die Laufzeitumgebung stellt dabei eine Software-Schicht zwischen der Applikations- und der Betriebssystem-Schicht der Mikroprozessoreinheit dar.

Fig. 1 zeigt schematisch eine solche Mikroprozessoreinheit mit einer gesicherten Laufzeitumgebung SWd mit einer Kommunikationseinheit MCCM, die als sog. MobiCore-Kommunikationsmodul ausgebildet ist. Die Kommunikationseinheit MCCM nutzt dabei das Betriebssystem MC (MobiCore) der

gesicherten Laufzeitumgebung SWd. Ferner dargestellt ist die unsichere Umgebung NWd mit einem Treiber MCD, der als sog. MobiCore-Treiber ausgebildet ist. Als Betriebssystem wird Rich OS verwendet. Gesicherte Laufzeitumgebung SWd und unsichere Umgebung NWd sind in einer sog. Trust-Zone-Hardware TZH verwirklicht.

Für den Datenaustausch zwischen gesicherter Laufzeitumgebung SWd und unsicherer Umgebung NWd ist eine Monitoreinheit M vorgesehen. Die Übertragung von Applikationsdaten AD, Steuerdaten MCP (MobiCore Control Protocol Daten), Steuerdaten NQ (Notification Queue) und Kontrolldaten FC (sog. Fast Calls) erfolgt über jeweilige, verschiedene Puffer. Die Übertragung der Applikationsdaten AD, der Steuerdaten MCP und NQ sowie der Kontrolldaten FC basiert auf ARM-Monitorcode, der in der Monitoreinheit M implementiert ist, welche Schnittstellen zu der gesicherten Laufzeitumgebung SWd und der unsicheren Umgebung NWd aufweist.

Fig. 2 zeigt die zur Verwirklichung des erfindungsgemäßen Verfahrens notwendigen Komponenten einer Mikroprozessoreinheit MP. Diese weist die bereits beschriebene gesicherte Laufzeitumgebung SWd und die unsichere Umgebung NWd auf. Die gesicherte Umgebung wird auch als Trust-Zone TZ bezeichnet. In dieser ist wenigstens ein als Trustlet TL bezeichnete Applikation enthalten. Über eine applikationsspezifische Schnittstelle (MC Trustlet API) kommuniziert diese mit einem Betriebssystem der gesicherten Laufzeitumgebung MC, bspw. MobiCore, (Block B1). Darüber hinaus sind Treiber DRV in der gesicherten Laufzeitumgebung SWd enthalten (Block B2).

In der unsicheren Umgebung, welches als Betriebssystem z.B. Rich OS nutzt, ist wenigstens eine Applikation APP vorgesehen, welche über eine anwendungsspezifische Schnittstelle (Application specific API) mit einem Applika-

tionsverbinder TLC (sog. Trustlet Connector) im Block A1 Daten austauschen kann. Der Applikationsverbinder kann über eine Schnittstelle TCI mit einer Applikation TL in der gesicherten Laufzeitumgebung kommunizieren. In der unsicheren Umgebung NWd ist ferner in einem Block A2 ein Treiber MCD, bspw. ein MobiCore-Treiber, enthalten, dem eine applikationsspezifische Schnittstelle (MC-Driver API) zugeordnet ist. Darüber hinaus ist in einem Block A3 ein virtueller Treiber VDRV in der unsicheren Umgebung enthalten. Der MobiCore-Treiber MCD kann über eine Schnittstelle MCI mit dem Betriebssystem MC der gesicherten Laufzeitumgebung kommunizieren. Über eine Schnittstelle DCI ist eine Kommunikation zwischen dem virtuellen Treiber VDRV und dem Treiber DRV der gesicherten Laufzeitumgebung möglich.

Erfindungsgemäße Eigenschaften der Mikroprozessoreinheit sind ein ausgelagerter Prozess-Scheduler in der unsicheren Umgebung in dem MobiCore-Treiber MCD. In dem Betriebssystem der gesicherten Laufzeitumgebung, bspw. MobiCore, ist ein optimierter µKernel enthalten, welcher bspw. keine Inter-Prozess-Kommunikation umfasst. In MC erfolgt ein preemptives Multitasking mit Zeitkontingenten. Darüber hinaus umfasst MC einen optimierten Task-Kontext. Schließlich umfasst die Mikroprozessoreinheit ein mehrschichtiges Treiberkonzept in den Blöcken A1, A2, A3, welche optimiert auf eine asynchrone Kommunikation mit einer multitaskingfähigen Umgebung in B1 sind.

Das mehrschichtige Treiberkonzept wird nachfolgend näher anhand Fig. 3 erläutert. Der MobiCore-Treiber MCD (Block A2) ist, wie in Fig. 3 dargestellt, aufgebaut und umfasst drei Schnittstellen für die Übertragung von Steuerdaten MCP und NQ sowie Kontrolldaten FC zwischen der unsicheren Umgebung und der gesicherten Laufzeitumgebung SWd. In dem MobiCore-

Betriebssystem (Block B1) sind ferner eine Laufzeitmanagementeinheit MCRT und ein Kontrolldaten-Handler FCH vorgesehen. Darüber hinaus ist die eingangs bereits erwähnte Monitoreinheit M zur Koordination des Datenaustauschs zwischen gesicherter Laufzeitumgebung SWd und unsicherer Umgebung NWd dargestellt.

Die der Übertragung der Steuerdaten MCP zugeordnete Schnittstelle ist für die Steuerung des MobiCore-Betriebssystems MC hauptverantwortlich. Hier wird entschieden, welche Tasks des Betriebssystems gestartet und gestoppt werden. Die von dem MobiCore-Betriebssystem gelieferten Daten werden auf richtige Formatierung überprüft. Für die Kommunikation wird ein spezieller Puffer in einem Speicher reserviert, der über eine Kontrollnachricht FC initialisiert wird. Der Speicher wird als World Shared Memory bezeichnet. Auf diesen kann sowohl von der unsicheren Umgebung NWd als auch von der gesicherten Laufzeitumgebung SWd zugegriffen werden.

Die der Übertragung der Steuerdaten NQ zugeordnete Schnittstelle ist dafür verantwortlich, mittels Nachrichten die Laufzeitmanagementeinheit MCRT zu informieren, dass Daten zur Abholung in dem Speicher bereit liegen. Diese Daten können von dem MobiCore-Treiber MCD stammen, d.h. zur Datenkommunikation zwischen einer Applikation der unsicheren Umgebung NWd mit einer bestimmten Applikation (Trustlet TL) der gesicherten Laufzeitumgebung SWd gehören. Im Falle einer Kommunikation auf der Schicht des MobiCore-Treibers MCD sind die Nachrichten mit einem Kennzeichner, einer sog. Session ID, versehen, anhand der das MobiCore-Betriebssystem MC die Nachricht eindeutig einer bestimmten Applikation TL der gesicherten Laufzeitumgebung SWd zuordnen kann.

Ebenso können Steuerdaten aus der Schicht der Steuerdaten MCP im Puffer der Steuerdaten NQ sein. In beiden Fällen informiert die NQ zugeordnete Schnittstelle über das Auslösen eines speziellen Interrupts (vorzugsweise ein spezieller Trust-Zone-Interrupt SIQ) die Laufzeitmanagement-Einheit MCRP über bereitgestellte Daten.

Über die den Kontrolldaten FC zugeordnete Schnittstelle findet der eigentliche Wechsel zwischen der unsicheren Umgebung NWd und der gesicherten Laufzeitumgebung SWd statt. Hierzu gibt es drei Möglichkeiten, wie mit der Monitoreinheit M interagiert werden kann. Über sog. Fast Calls, N-SIQ-Nachrichten oder NQ-IRQ-Nachrichten. Letztere werden als Notification IRQ bezeichnet. Die beiden ersten schalten lediglich von der unsicheren Umgebung in die gesicherte Laufzeitumgebung um. Im Fall von NQ-IRQs ist auch eine Umschaltung in umgekehrter Richtung möglich.

Die Aufgabe des Schedulers übernimmt die den Steuerdaten MCP zugeordnete Schnittstelle im MobiCore-Treiber MCD der unsicheren Umgebung. Der Treiber entscheidet, welche MobiCore-Task zur Ausführung gebracht wird.

Das beschriebene Konzept ermöglicht Optimierungen im µKernel-Ansatz. Im Vergleich zu klassischen µKernel-Ansätzen ist im MobiCore-Betriebssystem MC keine Interprozesskommunikation IPC implementiert. MobiCore-Prozesse können dennoch Daten austauschen über einen gemeinsam benutzten Speicher (World Shared Memory). MobiCore-Prozessen wird darüber hinaus eine bestimmte Rechenzeit zugewiesen, die aus Sicherheitsgründen nicht überschritten werden kann.

Ein MobiCore-Prozess besitzt einfache Threads- bzw. Task-Strukturen. Hierdurch müssen bei einem Kontextwechsel keine großen Datenmengen kopiert werden. Dadurch ergeben sich schnelle Kontextwechselseiten.

Die Thread-Struktur ist wie folgt: ID des Threads, aktueller Zustand des Threads, lokaler Exception Handler der Threads, Priorität des Threads. Die Task-Struktur ist wie folgt: Momentaner Zustand der Task, Task ID der Erzeuger-Task, externer Exception Handler der Task, Rechenzeitkontingent der Task, Anzahl der Threads, die die Task aktivieren oder spenden kann, Priorität und Rechte der Task.

Die Übertragung von Applikationsdaten, Steuerdaten und Kontrolldaten über verschiedene Puffer kann für die in Fig. 4 dargestellte Anwendung genutzt werden.

Eine Applikation in einem sicheren Bereich eines mobilen Telefons H1, H2, ..., Hn kommuniziert mit einem zentralen Hintergrundsystem (Datenbank D) und bekommt von dort eine Information zur Darstellung im Display des Sicherheitsmodus übermittelt. Die darzustellende Information kann bspw. eine Zahlenkolonne, ein Bild, ein Logo usw. sein. In regelmäßigen Abständen modifiziert das Hintergrundsystem D die im Sicherheitsmodus darzustellende Information. Die darzustellende Information wird zusätzlich einem breiten Nutzerkreis zeitgleich öffentlich bekannt gegeben. Der Anwender des Endgeräts H1, H2, ..., Hn, das eine sichere Anzeigevorrichtung umfasst, kann über einen zweiten Kommunikationskanal die aktuell gültige Information überprüfen. Der zweite Kommunikationskanal kann bspw. ein Internetfähiger Rechner, der Browser des Mobiltelefons, das eine Web-Verbindung aus der sicheren Welt umfasst, eine Tageszeitung usw. sein.

Hiermit steht nicht der Schutz von sicherheitskritischen Daten, z.B. kryptografische Schlüssel, im Vordergrund. Stattdessen ist hier die Nutzerwahrnehmung eines sicheren Displays oder eines sicheren Eingabemittels von Bedeutung. Hierdurch kann das Vertrauen von Endanwendern in mobile Endgeräte, bspw. für mobile Bankanwendungen oder Zahlungsanwendungen, erhöht werden.

Hierzu sind in dem Datenbanksystem D Informationen hinterlegt, die in den Endgeräten H1, H2, ..., Hn mittels eines in den Endgeräten implementierten Aktualisierungs-Servers über einen Aktualisierungs-Client der Datenbank ausgewechselt werden. Die Information ist in Fig. 4 beispielhaft ein Weihnachtsbaum. Dieselbe Information wird auch über einen in dem Datenbanksystem integrierten Web-Server öffentlich beliebigen Systemen zur Verifikation VS über einen öffentlichen Kanal v zugänglich gemacht. Der Ablauf der Aktualisierung der Information ist wie folgt:

1. Der Update-Server kontaktiert über einen sicheren Kanal s alle in der Datenbank D gelisteten mobilen Endgeräte H1, H2, ..., Hn, um eine neue Information zu verschicken. Dies bedeutet, in regelmäßigen Abständen modifiziert das Hintergrundsystem die im Sicherheitsmodus des mobilen Endgeräts H1, H2, ..., Hn darzustellende Information.
2. Um eine erfolgreiche Aktualisierung durchzuführen, muss der Aktualisierungs-Client sich bei dem Aktualisierungs-Server in dem Endgerät authentisieren. Dies kann bspw. mit einem Client-Zertifikat erfolgen. Ebenso muss auch der Aktualisierungs-Server des Endgeräts beim Aktualisierungs-Client der Datenbank nachweisen, dass der richtige Server für die Aktualisierung kontaktiert wurde. Dies kann mittels eines Server-Zertifikats erfolgen.

3. Nach der erfolgreichen gegenseitigen Authentisierung wird über den gesicherten Kanal s zwischen dem Aktualisierungs-Server der Handgeräte und dem Aktualisierungs-Client der Datenbank die neue Information (hier: Weihnachtsbaum) in das Endgerät in einem nur dem Sicherheitsmodus zugänglichen Bereich eingespielt.
4. Optional ist die Information mit einem digitalen Wasserzeichen geschützt und für jedes Endgerät personalisiert.
5. Die Personalisierung kann bspw. in der gesicherten Laufzeitumgebung überprüft werden. Falls sie nicht für das Endgerät passend ist, werden bestimmte Funktionalitäten der gesicherten Laufzeitumgebung blockiert. Dies hat bspw. zur Folge, dass keine mobilen Zahlvorgänge möglich sind.

Der Ablauf der Verifikation durch den Endanwender ist wie folgt:

1. Der Endanwender führt eine Aktion durch, die das mobile Endgerät H1, H2, ..., Hn in einen Sicherheitsmodus schaltet.
2. Das betreffende mobile Endgerät zeigt nun an einer bestimmten Stelle im sicheren Bildschirm eine beliebige Information, hier den Weihnachtsbaum, an.
3. Der Anwender des Endgeräts kann nun über einen zweiten, parallelen Kanal überprüfen, ob die Information auf seinem mobilen Endgerät mit der anderweitig veröffentlichten Information überein stimmt.

Hierdurch wird das Vertrauen des Endanwenders in ein Mobiltelefon, speziell für Bezahl- und Bankanwendungen, erhöht. Angriffe, die einen Sicherheitszustand eines elektronischen Endgeräts vortäuschen, werden deutlich erschwert. Der Benutzer wird in die Lage versetzt, zu überprüfen, ob sich

das mobile Gerät im Sicherheitsmodus befindet. Hierdurch wird ein höheres Vertrauen des Anwenders in die oben genannten Applikationen geschaffen.

Patentansprüche

1. Verfahren zum Datenaustausch zwischen einer gesicherten Laufzeitumgebung (SWd), in welcher eine Anzahl an sicheren Applikationen (TL) ausführbar ist, und einer unsicheren Umgebung (NWd) einer Mikroprozessoreinheit (MP), insbesondere in einem mobilen Endgerät, bei dem die Übertragung von Applikationsdaten (AD) und Steuerdaten (MCP, NQ) über verschiedene Puffer erfolgt.

2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, dass die Steuerdaten (MCP, NQ) unterschiedlicher Art über unterschiedliche Puffer übertragen werden.

3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, dass, die Umschaltung zwischen der gesicherten Laufzeitumgebung (SWd) und der unsicheren Umgebung (NWd) betreffende, Kontrolldaten (FC) über einen eigenen sicheren Puffer übertragen werden.

4. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Übertragung der Applikationsdaten (AD) und der Steuerdaten (MCP, NQ) sowie optional der Kontrolldaten (FC) auf einem ARM Monitorcode basiert, der in einer Monitoreinheit (M) implementiert ist, welche Schnittstellen zur der gesicherten Laufzeitumgebung (SWd) und der unsicheren Umgebung (NWd) aufweist.

5. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass die Übertragung der Applikationsdaten (AD) und der Steuerdaten (MCP, NQ) zwischen der gesicherten Laufzeitumgebung (SWd) und einem Treiber (MCD) der unsicheren Umgebung (NWd) erfolgt.

6. Verfahren nach Anspruch 5, dadurch gekennzeichnet, dass ein in der unsicheren Umgebung (NWd), insbesondere in einer Schnittstelle des Treibers (MCD) für die Steuerdaten (MCP), implementierter Scheduler festlegt, welche der sicheren Applikationen (TL) in der gesicherten Laufzeitumgebung (SWd) zur Ausführung gebracht wird.

7. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass der Datenaustausch unter Nutzung eines, von der gesicherten Laufzeitumgebung (SWd) und der unsicheren Umgebung (NWd) les- und/oder beschreibbaren, Speicherbereichs eines Speichers (WSM) erfolgt.

8. Verfahren nach Anspruch 7, dadurch gekennzeichnet, dass über Steuernachrichten die gesicherte Laufzeitumgebung über für diese bestimmte Daten in dem Speicherbereich benachrichtigt wird.

9. Verfahren nach Anspruch 8, dadurch gekennzeichnet, dass die Steuernachrichten mit einem eindeutigen Sitzungs-Kennzeichner versehen werden, anhand dem die gesicherte Laufzeitumgebung (SWd) die Steuernachricht einer der in der gesicherten Laufzeitumgebung (SWd) ausgeführten Applikationen (TL) zuordnen kann.

10. Verfahren nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass jedem in der gesicherten Laufzeitumgebung (SWd) ablaufenden Prozess eine definierte, nicht überschreitbare Rechenzeit zugewiesen wird.

11. Verfahren nach Anspruch 10, dadurch gekennzeichnet, dass ein jeweiliger Prozess nachfolgende Threadstruktur aufweist: Kennzeichner (ID) des Threads; Aktueller Zustand des Threads; Lokaler Exception Handler der Threads; Priorität des Threads.

12. Verfahren nach Anspruch 10, dadurch gekennzeichnet, dass ein jeweiliger Prozess nachfolgende Taskstruktur aufweist: Momentaner Zustand der Task; Task-Kennzeichner der Erzeuger-Task; Externer Exception Handler der Task; Rechenzeitkontingent der Task; Anzahl der Threads, die die Task aktivieren oder bereitstellen kann; Priorität und Rechte der Task.

13. Mikroprozessoreinheit mit einer gesicherten Laufzeitumgebung (SWd) und einer unsicheren Umgebung (NWd), welche derart ausgestaltet ist, dass zum Datenaustausch zwischen der gesicherten Laufzeitumgebung (SWd) und der unsicheren Umgebung (NWd) die Übertragung von Applikationsdaten (AD) und Steuerdaten (MCP, NQ, FC) über verschiedene Puffer erfolgt.

14. Mikroprozessoreinheit nach Anspruch 13, dadurch gekennzeichnet, dass diese derart ausgestaltet ist, dass durch diese ein Verfahren nach einem der Ansprüche 2 bis 12 ausführbar ist.

15. Mobiles Endgerät, umfassend eine Mikroprozessoreinheit nach Anspruch 13 oder 14.

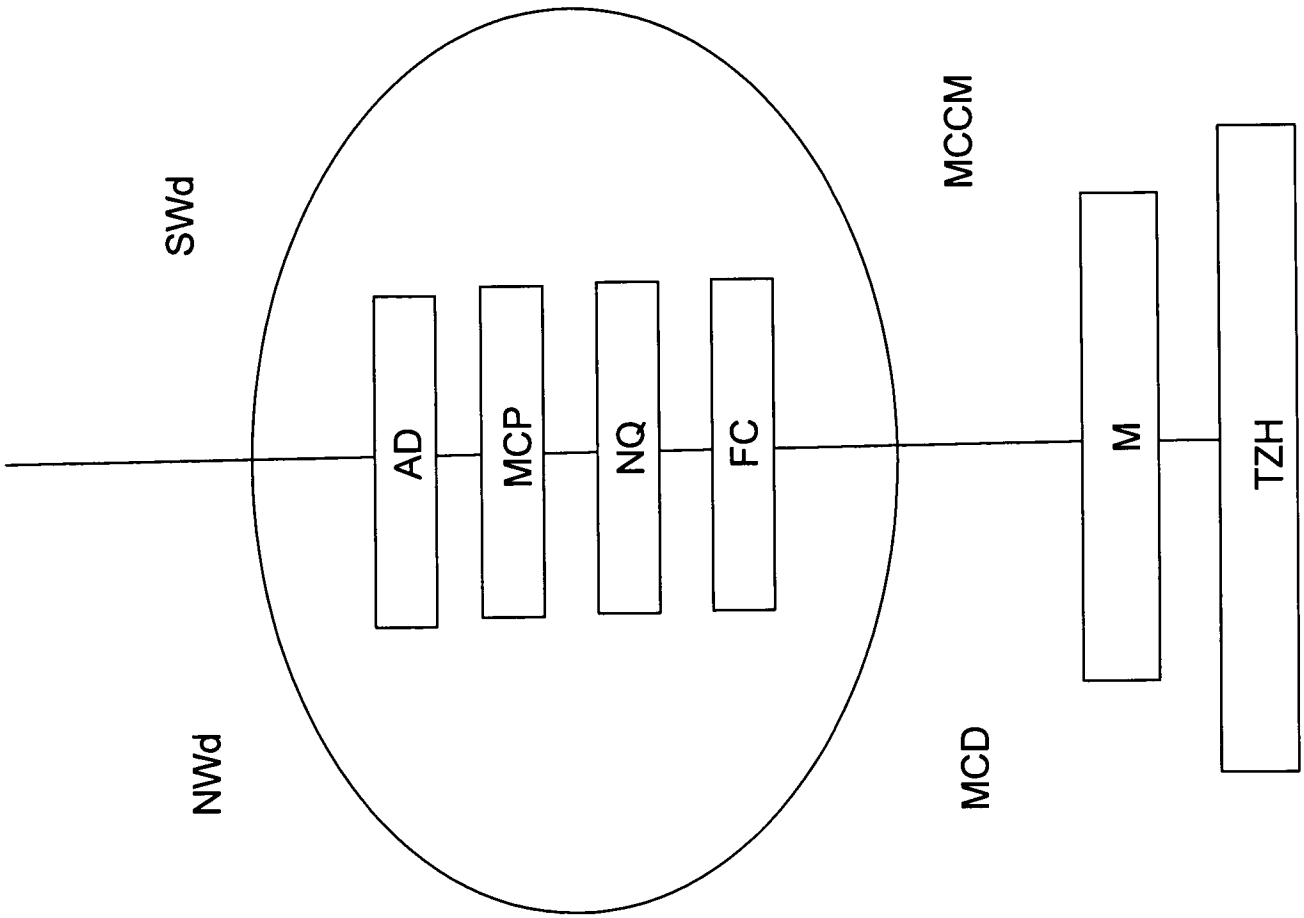


Fig. 1

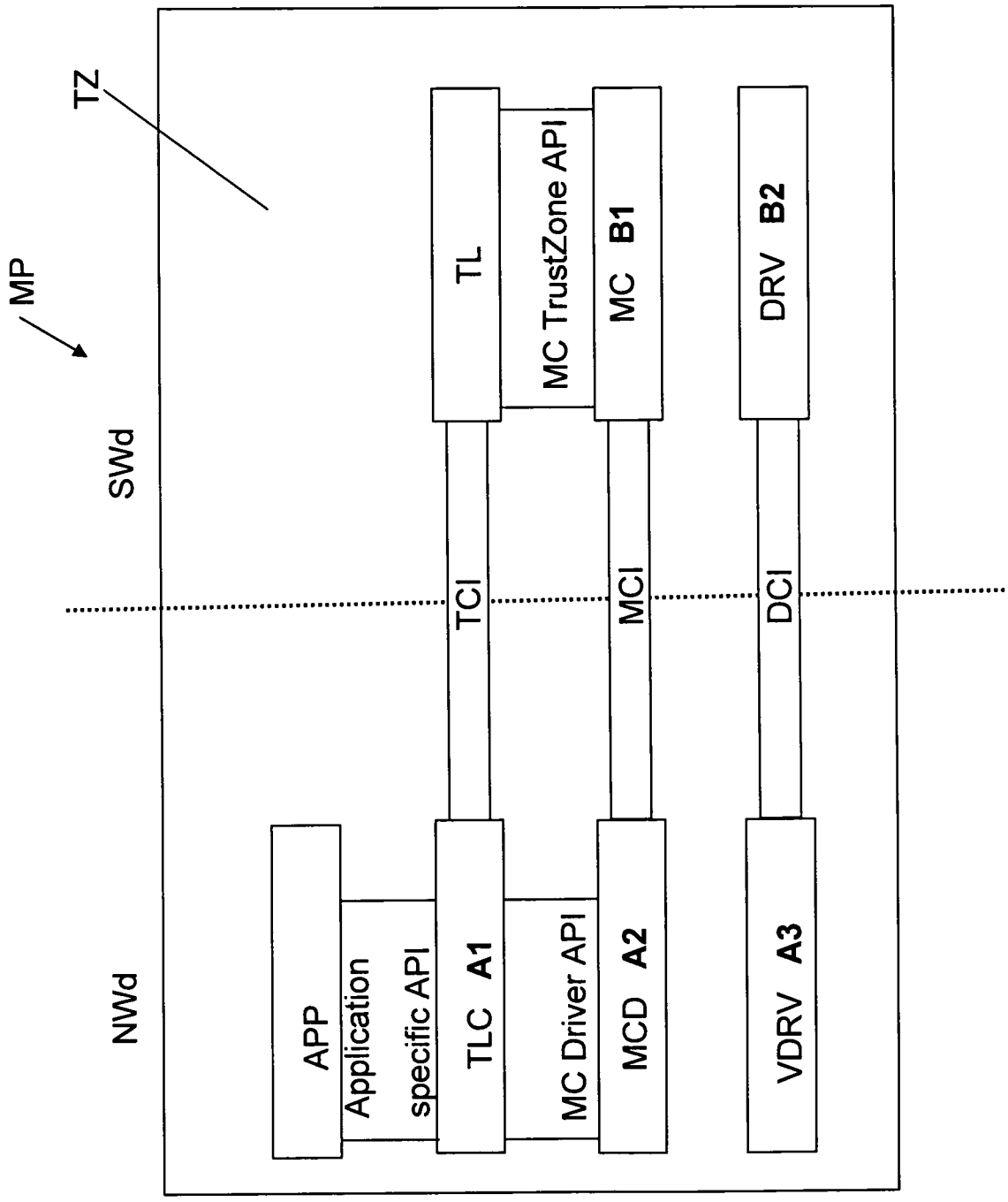

TZ
MP
SWd
NWd
TL
MC TrustZone API
MC B1
DRV B2
TCI
MCI
DCI
APP
Application specific API
TLC A1
MC Driver API
MCD A2
VDRV A3

Fig. 2

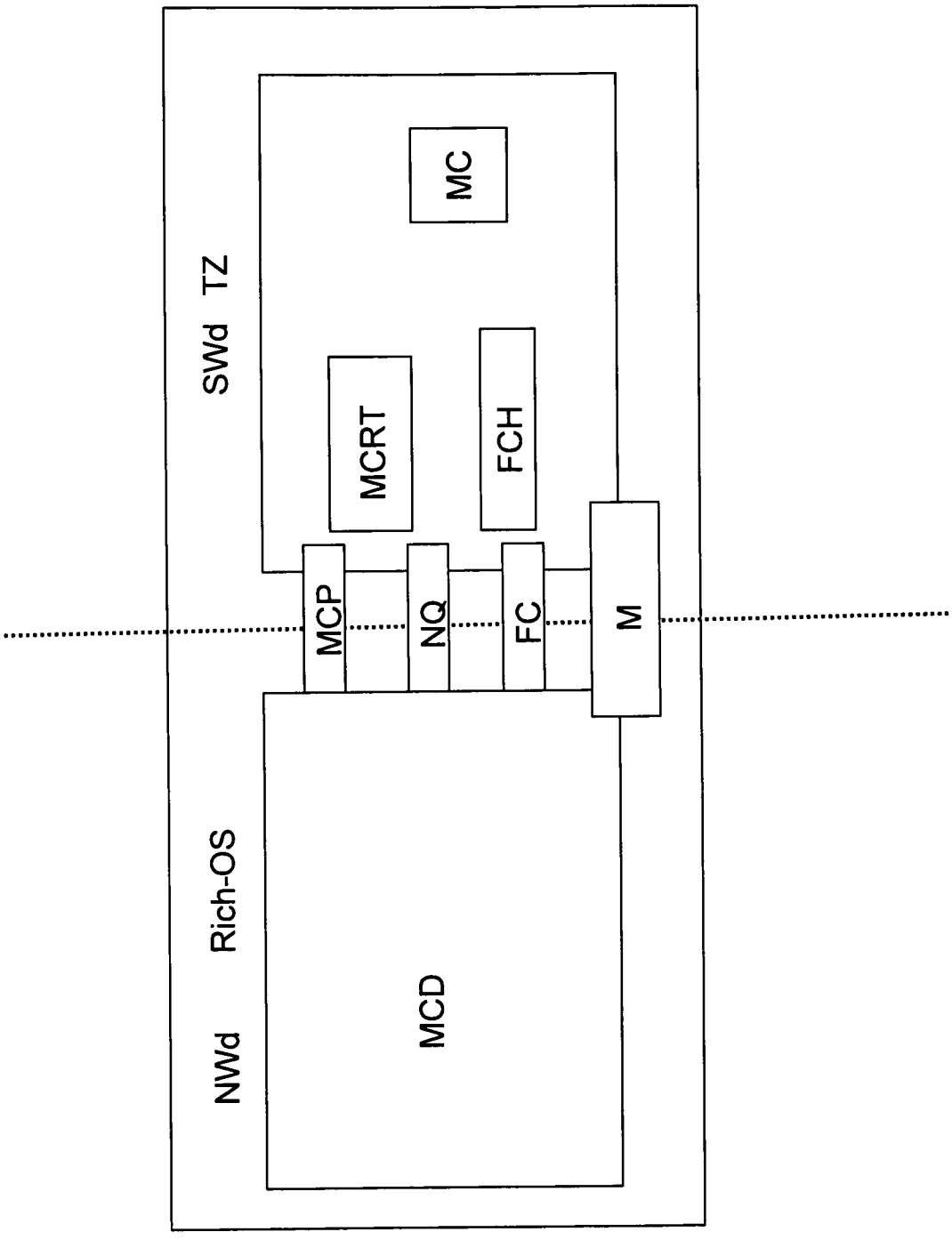


Fig. 3

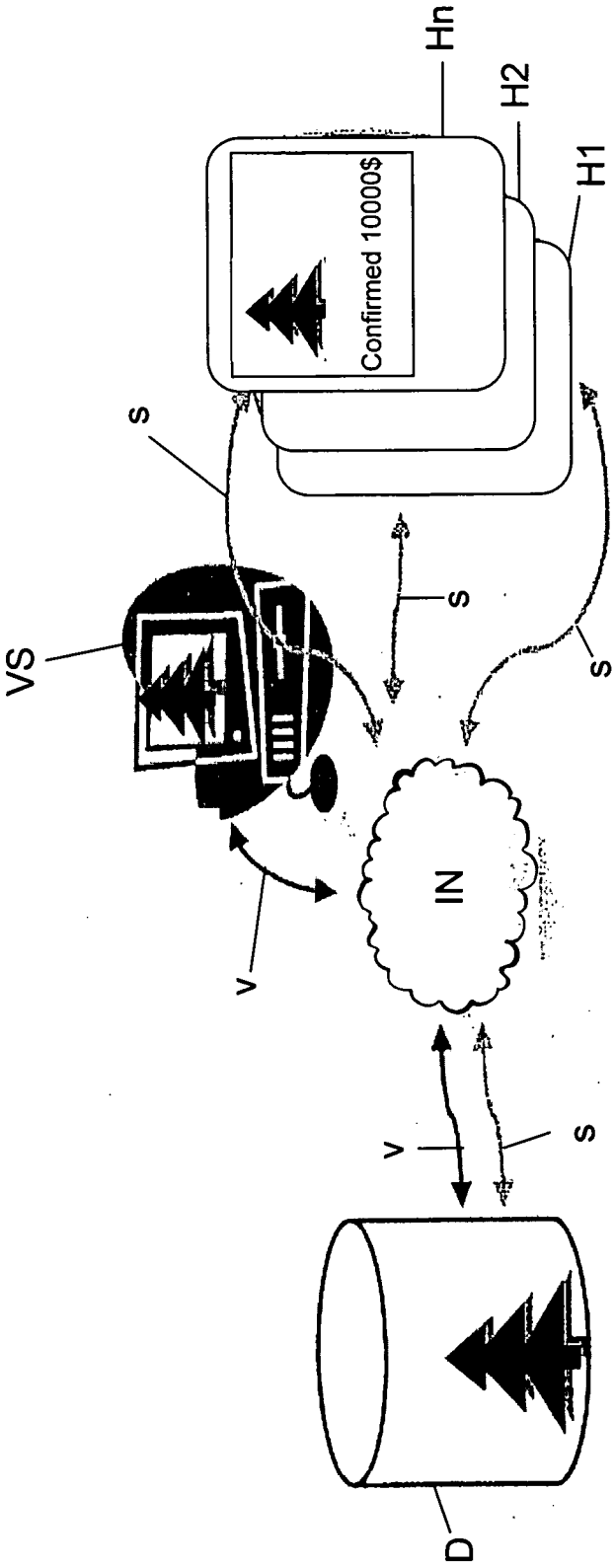


Fig. 4