

(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
11. April 2013 (11.04.2013)



(10) Internationale Veröffentlichungsnummer
WO 2013/050154 A1

- (51) Internationale Patentklassifikation:
G06F 21/53 (2013.01) *G06F 9/54* (2006.01)
- (21) Internationales Aktenzeichen: PCT/EP2012/004159
- (22) Internationales Anmeldedatum:
4. Oktober 2012 (04.10.2012)
- (25) Einreichungssprache: Deutsch
- (26) Veröffentlichungssprache: Deutsch
- (30) Angaben zur Priorität:
10 2011 115 135.8
7. Oktober 2011 (07.10.2011) DE
- (71) Anmelder: GIESECKE & DEVRIENT GMBH
[DE/DE]; Prinzregentenstrasse 159, 81677 München (DE).
- (72) Erfinder: SPITZ, Stephan; Parkstraße 18, 85757
Karlsfeld (DE). KÖHLER, Markus; Ganselieselstrasse
38b, 81739 München (DE). MARTINI, Ullrich;
Zeppelinstr. 61, 81669 München (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, BN, 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, PA, 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,
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,

[Fortsetzung auf der nächsten Seite]

(54) Title: MICROPROCESSOR SYSTEM WITH SECURED RUNTIME ENVIRONMENT

(54) Bezeichnung : MIKROPROZESSORSYSTEM MIT GESICHERTER LAUFZEITUMGEBUNG

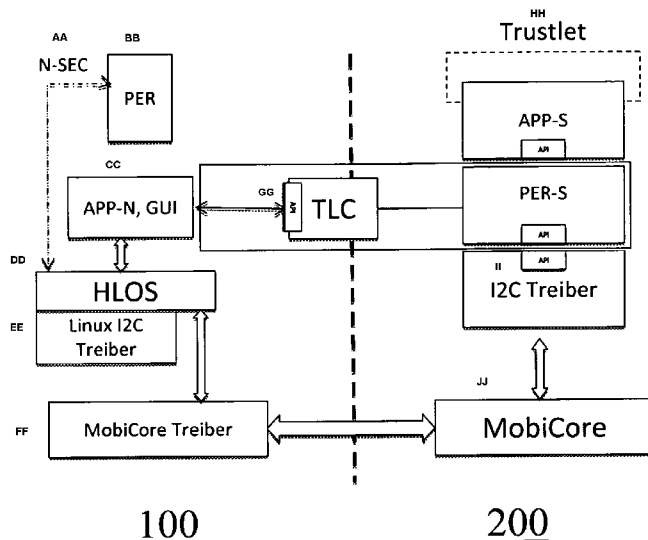


Fig. 1

FIG.1
EE Linux I2C driver
II I2C driver
FF MobiCore driver

(57) Abstract: Microprocessor system that is implemented or can be implemented in a mobile terminal and comprises: a normal operating system designed to generate and maintain a non-secure runtime environment and a security operating system designed to generate and maintain a secured runtime environment, and an operating system interface between the normal operating system and the security operating system, said operating interface being designed to control communication between the non-secure runtime environment and the secured runtime environment on the operating system level, and at least one filter interface that is designed to securely control communication between the non-secure runtime environment and a secured runtime environment on a level different from the operating system level.

(57) Zusammenfassung: Mikroprozessorsystem, in einem mobilen Endgerät implementierbar oder implementiert, und das aufweist: ein Normalbetriebssystem, das zum Erzeugen und Unterhalten einer unsicheren Laufzeitumgebung eingerichtet ist, ein Sicherheitsbetriebssystem, das zum Erzeugen und Unterhalten einer gesicherten Laufzeitumgebung eingerichtet ist, und eine Betriebssystem-Schnittstelle zwischen dem Normalbetriebssystem und dem Sicherheitsbetriebssystem, die

[Fortsetzung auf der nächsten Seite]



SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

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

Erklärungen gemäß Regel 4.17:

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

Veröffentlicht:

— *mit internationalem Recherchenbericht (Artikel 21 Absatz 3)*

dazu eingerichtet ist, Kommunikation zwischen der unsicheren Laufzeitumgebung und der gesicherten Laufzeitumgebung auf Betriebssystem-Ebene gesichert zu steuern, und mindestens eine Filter-Schnittstelle, die dazu eingerichtet ist, Kommunikation zwischen der unsicheren Laufzeitumgebung und der gesicherten Laufzeitumgebung auf einer von der Betriebssystem-Ebene unterschiedlichen Ebene gesichert zu steuern.

Die Erfindung betrifft ein in einem mobilen Endgerät implementierbares oder implementiertes Mikroprozessorsystem mit einer gesicherten Laufzeit-
5 umgebung.

Ein Mikroprozessorsystem umfasst einen Mikroprozessor und ein oder mehrere flüchtige und nichtflüchtige Speicher, in funktioneller Sichtweise insbesondere zumindest einen Betriebssystemspeicher, in dem ein Betriebssystem
10 abgespeichert ist, ein oder mehrere Applikationsspeicher, in denen durch den Mikroprozessor unter Steuerung des Betriebssystems ausführbare Applikationen abgespeichert sind, und einen Arbeitsspeicher, in dem durch den Mikroprozessor zu bearbeitende Daten ablegbar sind. Das Betriebssystem erzeugt und unterhält während seines Betriebs eine Laufzeitumgebung, in
15 der die Applikationen auf dem Mikroprozessorsystem lauffähig sind. Das Mikroprozessorsystem ist in der Regel auf einem Chip, z.B. Halbleiterchip, implementiert, eventuell auch auf mehreren verbunden Chips.

Unter einem mobilen Endgerät wird ein Mobiltelefon, Smartphone, PDA
20 (Personal Digital Assistant) oder ähnliches Endgerät verstanden.

Unter der Bezeichnung ARM Trustzone Architektur ist eine zweigeteilte Laufzeit-Architektur der Firma ARM für ein Mikroprozessorsystem bekannt, die zwei Laufzeitumgebungen umfasst. Eine erste, „Normal Zone“ oder
25 „Normal World“ genannte unsichere Laufzeitumgebung ist durch ein Normalbetriebssystem gesteuert. Eine zweite, „Trustzone“ oder „Trusted World“ oder „Secure World“ genannte gesicherte oder vertrauenswürdige Laufzeitumgebung ist durch ein Sicherheitsbetriebssystem gesteuert.

30 Das Normalbetriebssystem kann beispielsweise ein gängiges Betriebssystem wie Android, Windows Phone, Symbian oder dergleichen sein.

Durch die Anmelderin der vorliegenden Patentanmeldung wird unter der Markenbezeichnung MOBICORE ein Sicherheitsbetriebssystem für in mobile Endgeräte zu implementierende Chips hergestellt und vertrieben. Insbesondere sicherheitskritische Applikationen und manche Peripherie-Funktionen (z.B. Tastaturtreiber) werden durch das Sicherheitsbetriebssystem MOBICORE auf sichere Weise gesteuert. Applikationen unter dem Sicherheitsbetriebssystem werden auch als Trustlets bezeichnet, in Anlehnung an die Begriffe „Trust“ (Vertrauen) und „Applet“. Weitere Applikationen und andere Peripherie-Funktionen (z.B. Displaytreiber, NFC-Schnittstellen-Treiber, Radio Interface Layer RIL) werden durch das Normalbetriebssystem (auch Rich OS oder HLOS genannt) gesteuert. Das Sicherheitsbetriebssystem umfasst einen kryptographischen Schlüssel, den Authentisierungsschlüssel Kauth, mittels dessen sich das Sicherheitsbetriebssystem gegenüber einem Hintergrundsystem identifizieren kann. Wahlweise ist der Authentisierungsschlüssel Kauth zusätzlich zum Betreiben eines sicheren Kommunikationskanals zwischen dem Chip bzw. Mobilendgerät und dem Hintergrundsystem vorgesehen. Das Hintergrundsystem ist beispielsweise ein an sich bekanntes Hintergrundsystem eines Mobilfunknetzes.

20

Zugriffe des Normalbetriebssystems aus der unsicheren Laufzeitumgebung in die gesicherte Laufzeitumgebung, z.B. zum Aufruf einer Applikation in der gesicherten Laufzeitumgebung (Trustlet), oder zum Austausch von Daten, erfolgen standardmäßig ausschließlich über einen innerhalb der unsicheren Laufzeitumgebung implementierten, aber durch das Sicherheitsbetriebssystem kontrollierten und auf Betriebssystemebene funktionierenden Sicherheitstreiber, z.B. Mobicore-Treiber. Der Sicherheitstreiber kann z.B. Aufgaben des Normalbetriebssystems an das Sicherheitsbetriebssystem abtreten und durch das Sicherheitsbetriebssystem bereitgestellte Ergebnisse zu

Aufgaben an das Normalbetriebssystem weiterleiten. Der Sicherheitstreiber vermittelt einen gesicherten Kommunikationskanal zwischen dem Normalbetriebssystem und dem Sicherheitsbetriebssystem, der für den Austausch kleinerer Datenmengen völlig ausreichend ist.

5

Auf eine in der gesicherten Laufzeitumgebung implementierte Applikation (Trustlet) oder Peripherie-Funktion ist ein Zugriff nur durch das Sicherheitsbetriebssystem möglich. Ein direkter Zugriff auf das Trustlet aus dem Normalbetriebssystem heraus ist dagegen nicht möglich.

10

Unter dem Normalbetriebssystem implementiert und damit weitestgehend ungesichert sind (mit Ausnahme z.B. der Tastatur) so gut wie alle Peripherie-Strukturen des mobilen Endgeräts, also über den zentralen Mikroprozessor hinausgehende Strukturen des mobilen Endgeräts, da die jeweiligen Peripherie-Treiber in der unsicheren Laufzeitumgebung Normal Zone implementiert sind.

15

Manche Kommunikationsvorgänge der Peripherie-Strukturen mit dem Betriebssystem oder über das Betriebssystem sind sicherheitskritisch, z.B. solche, bei denen geheime Daten wie z.B. die Geheimzahl PIN verarbeitet werden, oder bei denen sicherheitsrelevante Kommandos verarbeitet werden, z.B. ein Kommando zur Bestätigung oder Veranlassung einer Zahlungsverkehrstransaktion. Für solche sicherheitskritischen Kommunikationsvorgänge wäre eine gesicherte Anbindung der Peripherie-Strukturen wünschenswert, um ein Ausspähen von Daten oder Manipulieren von Kommandos zu verhindern.

20

25

Die direkteste Möglichkeit, den Betrieb der Peripherie-Strukturen sicherer zu machen ist, die jeweiligen Peripherie-Treiber statt unter dem Normalbe-

- triebssystem unter dem Sicherheitsbetriebssystem zu implementieren. (Der relativ kleine und kompakte Tastaturtreiber in einem mobilen Endgerät mit Mobicore als Sicherheitsbetriebssystem ist beispielsweise häufig bereits standardmäßig unter dem Sicherheitsbetriebssystem implementiert.) Dies hat
- 5 allerdings zur Folge, dass sämtliche Zugriffe auf die Peripherie-Struktur zwangsweise durch das Sicherheitsbetriebssystem erfolgen müssen. Da unter dem Sicherheitsbetriebssystem für sämtliche Zugriffe aufwändige Sicherheitsmaßnahmen wie Authentisierungen und Verschlüsselungen erforderlich sind, verlangsamt die Verlagerung der Treiber vom Normalbetriebssystem
- 10 tem in das Sicherheitsbetriebssystem den Betrieb des mobilen Endgeräts ganz erheblich. Ein erheblicher Anteil von Zugriffen auf eine Peripherie-Struktur ist aber nicht sicherheitskritisch und könnte genauso gut unter dem unsicheren aber schnelleren Normalbetriebssystem laufen.
- 15 Applikationen für unterschiedlichste Anwendungszwecke, beispielsweise Zahlungsverkehrstransaktionen (Banking, Payment) oder Ticketverkäufe für Verkehrsmittel (Ticketing) oder Bonuspunkte- und Rabattsysteme (Loyalty) werden verbreitet zumindest teilweise unter Steuerung des Sicherheitsbetriebssystems, lauffähig in der gesicherten Ausführungsumgebung, implementiert.
- 20 Zumindest Teile der Applikation sind dabei zusätzlich oder nur unter Steuerung des Normalbetriebssystems, lauffähig in der unsicheren Ausführungsumgebung, implementiert, insbesondere Teile der Applikation, die auf eine Interaktion mit einem Benutzer oder einer Benutzerschnittstelle des mobilen Endgeräts gerichtet sind. Sofern Kommunikation zwischen Teilen
- 25 der Applikation in der gesicherten und der unsicheren Laufzeitumgebung erforderlich ist, erfolgt diese auf Betriebssystemebene über den Sicherheitstreiber, was die Kommunikation langsam macht.

Der Erfindung liegt die Aufgabe zu Grunde, ein in einem mobilen Endgerät implementierbares oder implementiertes Mikroprozessorsystem mit einer gesicherten Laufzeitumgebung zu schaffen, das eine sichere und gleichzeitig effiziente Kommunikation zwischen der unsicheren Laufzeitumgebung und
5 der gesicherten Laufzeitumgebung ermöglicht.

Die Aufgabe wird gelöst durch ein Mikroprozessorsystem nach Anspruch 1. Vorteilhafte Ausgestaltungen der Erfindung sind in den abhängigen Ansprüchen angegeben.

10

Das erfindungsgemäße Mikroprozessorsystem gemäß dem unabhängigen Anspruch 1 ist in einem mobilen Endgerät implementierbar oder implementiert und weist auf:

- ein Normalbetriebssystem, das zum Erzeugen und Unterhalten einer unsi-
15 cheren Laufzeitumgebung eingerichtet ist,
- ein Sicherheitsbetriebssystem, das zum Erzeugen und Unterhalten einer gesicherten Laufzeitumgebung eingerichtet ist, und
- eine Betriebssystem-Schnittstelle zwischen dem Normalbetriebssystem und dem Sicherheitsbetriebssystem, die dazu eingerichtet ist, Kommunikation
20 zwischen der unsicheren Laufzeitumgebung und der gesicherten Laufzeitumgebung auf Betriebssystem-Ebene gesichert zu steuern.

Das Mikroprozessorsystem ist gekennzeichnet durch mindestens Filter-Schnittstelle, die dazu eingerichtet ist, Kommunikation zwischen der unsi-
25 cheren Laufzeitumgebung und der gesicherten Laufzeitumgebung auf einer von der Betriebssystem-Ebene unterschiedlichen Ebene gesichert zu steuern.

Die Filter-Schnittstelle ermöglicht eine beschleunigte und dabei gesicherte Kommunikation, da durch sie neben dem Kommunikationskanal auf Be-

triebssystemebene ein weiterer sicherer Kommunikationskanal bereitgestellt wird, der für eine direktere Kommunikation genutzt werden kann als eine Kommunikation über den Umweg des Betriebssystems.

- 5 Daher ist gemäß Anspruch 1 ein Mikroprozessorsystem geschaffen, das eine sichere und gleichzeitig effiziente Kommunikation zwischen der unsicheren Laufzeitumgebung und der gesicherten Laufzeitumgebung ermöglicht.

10 Die Filter-Schnittstelle hat eine erste, zum Ablaufen in der unsicheren Laufzeitumgebung eingerichtete unsichere Teil-Filterschnittstelle und eine zweite, zum Ablaufen in der sicheren Laufzeitumgebung eingerichtete sichere Teil-Filterschnittstelle. Hierdurch kann die Filter-Schnittstelle zwischen der sicheren und der unsicheren Laufzeitumgebung vermitteln.

- 15 Gemäß einer Alternative ist als Filter-Schnittstelle eine Hardware-Filter-Schnittstelle vorgesehen, die dazu eingerichtet ist, Kommunikation zwischen der unsicheren Laufzeitumgebung und der gesicherten Laufzeitumgebung auf Hardware-Ebene gesichert zu steuern. Die sichere Kommunikation zwischen der gesicherten und der unsicheren Laufzeitumgebung kann also un-
- 20 mittelbar und direkt auf Hardware-Ebene und daher beschleunigt erfolgen.

Die Steuerung auf Hardware-Ebene erfolgt wahlweise als Steuerung durch direkte Ansteuerung von Hardware-Komponenten wie z.B. Peripherie-

25 Strukturen wie Display, SIM-Karte etc. (vgl. unten), oder wahlweise als Steuerung über ein Bussystem, beispielsweise über einen I2C Bus.

Gemäß der Alternative, wonach als Filter-Schnittstelle eine Hardware-Filter-Schnittstelle vorgesehen ist, weist das Mikroprozessorsystem wahlweise weiter auf:

zumindest einen durch das Normalbetriebssystem ansteuerbaren, in der unsicheren Laufzeitumgebung lauffähigen Treiber für eine Peripherie-Struktur des mobilen Endgeräts, und einen durch das Sicherheitsbetriebssystem ansteuerbaren, in der gesicherten Laufzeitumgebung lauffähigen – vorzugsweise reduzierten – Treiber für die Peripherie-Struktur, der vorzugsweise lediglich eine Teilmenge der Funktionalitäten hat, welche der in der unsicheren Laufzeitumgebung lauffähige Treiber hat. Die Hardware-Filter-Schnittstelle ist hierbei dazu eingerichtet, Kommunikation zwischen dem in der unsicheren Laufzeitumgebung lauffähigen Treiber für eine Peripherie-Struktur und dem in der gesicherten Laufzeitumgebung lauffähigen vorzugsweise reduzierten Treiber für die Peripherie-Struktur auf Hardware-Ebene gesichert zu steuern.

Die Hardware-Filter-Schnittstelle ermöglicht also eine direkte Kommunikation von unter dem Normalbetriebssystem implementierten Treiberteilen mit unter dem Sicherheitsbetriebssystem implementierten Treiberteilen über die Hardware-Filter-Schnittstelle.

Als Peripherie-Struktur des Endgeräts ist beispielsweise vorgesehen: eine NFC-Schnittstelle (NFC-Interface), ein am Verbindungsaufbau ins Mobilfunknetz beteiligter Radio Interface Layer RIL, ein zur Authentisierung und Verschlüsselung verwendetes Sicherheitselement SE (z.B. SIM-Karte oder USIM-Karte oder Secure MicroSD-Karte oder festimplementiertes Embedded Secure Element), ein Display, insbesondere Touch-Screen-Display, eine PIN-Prüfungs-Routine, mit der die über die Tastatur eingegebene Geheimzahl PIN (Personal Identification Number) verifizierbar ist, Einwahlroutinen zur Einwahl in vom Mobilfunknetz unterschiedliche Netze wie beispielsweise das Internet (IP) bzw. virtuelle private Netzwerke (VPN), Pufferspeicher (z.B. Frame Buffer).

Wahlweise ist, sofern Treiber für mehrere Peripherie-Strukturen in dem Mikroprozessorsystem implementiert sind, für alle Peripherie-Strukturen eine gemeinsame Hardware-Schnittstelle vorgesehen, die Zugriffe zwischen
5 der unsicheren und der gesicherten Laufzeitumgebung auf Hardware-Ebene ausschließlich zwischen den zur selben Peripherie-Struktur gehörigen Treibern ermöglicht und Zugriffe zwischen zu unterschiedlichen Peripherie-Strukturen gehörigen Treibern auf Hardware-Ebene verhindert. Dadurch, dass die direkten Hardware-Ebenen-Zugriffe nur den einander entsprechenden
10 den Treibern zur selben Peripherie-Struktur vorbehalten sind, ist ein missbräuchlicher Zugriff einer Schadsoftware, beispielsweise eines Tojaners oder dergleichen, auf den Treiber abgewehrt.

Zugriffe zwischen zu unterschiedlichen Peripherie-Strukturen gehörigen
15 Treibern könnten höchstens über den Sicherheitstreiber auf Betriebssystemebene erfolgen, sofern diese Zugriffe gemäß einer Überprüfung durch den Sicherheitstreiber zulässig sind. Der Sicherheitstreiber würde z.B. Schadsoftware erkennen und abhalten.

20 Wahlweise ist die Hardware-Filter-Schnittstelle dazu eingerichtet, (z.B. in der unsicheren Laufzeitumgebung) an den Treiber für eine Peripherie-Struktur gerichtete Zugriffe auf Sicherheitsrelevanz hin zu prüfen und im Fall, dass der Zugriff als sicherheitsrelevant erkannt wird, den Zugriff an den reduzierten Treiber in der gesicherten Laufzeitumgebung zu leiten, und
25 im Fall, dass der Zugriff als nicht sicherheitsrelevant erkannt wird, den Zugriff an den Treiber in der unsicheren Laufzeitumgebung zu leiten.

Wahlweise ist der in der gesicherten Laufzeitumgebung lauffähige Treiber für eine Peripherie-Struktur dahingehend reduziert, dass er lediglich Funkti-

onalitäten umfasst, die zur Abarbeitung sicherheitsrelevanter Zugriffe erforderlich sind. Der in der unsicheren Laufzeitumgebung lauffähige Treiber für eine Peripherie-Struktur umfasst hingegen vorzugweise sämtliche Funktionalitäten des Treibers.

5

Gemäß einer weiteren Alternative ist als Filter-Schnittstelle eine Applikations-Filter-Schnittstelle vorgesehen, die dazu eingerichtet ist, Kommunikation zwischen der unsicheren Laufzeitumgebung und der gesicherten Laufzeitumgebung auf Applikations-Ebene gesichert zu steuern. Durch die Applikations-Filter-Schnittstelle wird ein direkter sicherer Kommunikationskanal auf Applikations-Ebene bereitgestellt, der eine schnellere Kommunikation ermöglicht als über den Umweg über die Betriebssystemebene und den auf Betriebssystemebene arbeitenden Sicherheitstreiber.

10

15 Das Mikroprozessorsystem weist weiter wahlweise zumindest eine Applikation auf, die zumindest teilweise durch das Normalbetriebssystem ansteuerbar und in der unsicheren Laufzeitumgebung lauffähig ist, und zumindest teilweise durch das Sicherheitsbetriebssystem ansteuerbar und in der gesicherten Laufzeitumgebung lauffähig ist. Dabei ist die Applikations-Filter-Schnittstelle dazu eingerichtet, Kommunikation zwischen der in der unsicheren Laufzeitumgebung lauffähigen zumindest teilweisen Applikation und der in der gesicherten Laufzeitumgebung lauffähigen zumindest teilweise Applikation auf Applikations-Ebene gesichert zu steuern.

20

25 Die Applikation kann beispielsweise eine Banking- oder Payment-Applikation für Zahlungsverkehrstransaktionen sein, oder eine Ticketing-Applikation für Kauf und Verwaltung von Fahrkarten für unterschiedlichste Zwecke, z.B. Verkehrsmittel. Zumindest sicherheitsrelevante Funktionalitäten der Applikation sind wahlweise unter dem Sicherheitsbetriebssystem

implementiert. Ergänzende, nicht sicherheitsrelevante Funktionalitäten sind wahlweise zusätzlich oder ausschließlich unter dem Normalbetriebssystem implementiert.

- 5 Im Unterschied zu Treibern, wo der Treiber in den meisten Fällen überwiegend unter dem Normalbetriebssystem implementiert sein wird, wird bei Applikationen wie z.B. Banking-Applikationen meistens der überwiegende Teil unter dem Sicherheitsbetriebssystem implementiert sein.
- 10 Wahlweise ist, sofern mehrere Applikationen in dem Mikroprozessorsystem implementiert sind, für alle Applikationen eine gemeinsame Applikations-Filter-Schnittstelle vorgesehen, die Zugriffe zwischen der unsicheren und der gesicherten Laufzeitumgebung auf Applikations-Ebene ausschließlich zwischen den zur selben Applikation gehörigen Teilen in der unsicheren
15 bzw. gesicherten Laufzeitumgebung ermöglicht und Zugriffe zwischen zu unterschiedlichen Applikationen gehörigen Teilen auf Applikations-Ebene verhindert. Dadurch, dass die direkten Applikations-Ebenen-Zugriffe nur den einander entsprechenden Teilen der selben Applikation vorbehalten sind, ist ein missbräuchlicher Zugriff einer Schadsoftware, beispielsweise
20 eines Tojaners oder dergleichen, auf die Applikation abgewehrt.

Applikationszugriffe zwischen unterschiedlichen Applikationen über die Grenze zwischen der gesicherten und der unsicheren Laufzeitumgebung hinweg können nur höchstens auf Betriebssystemebene über den Sicherheits-
25 treiber erfolgen.

Wahlweise ist die Applikations-Filter-Schnittstelle dazu eingerichtet, (z.B. in der unsicheren Laufzeitumgebung) an die Applikation gerichtete Zugriffe auf Sicherheitsrelevanz hin zu prüfen und im Fall, dass der Zugriff als si-

cherheitsrelevant erkannt wird, den Zugriff an die zumindest teilweise Applikation in der gesicherten Laufzeitumgebung zu leiten, und im Fall, dass der Zugriff als nicht sicherheitsrelevant erkannt wird, den Zugriff an die zumindest teilweise Applikation in der unsicheren Laufzeitumgebung zu
5 leiten.

Die Applikations-Filter-Schnittstelle kann alternativ oder zusätzlich zur Hardware-Filter-Schnittstelle vorgesehen sein. Wahlweise sind die Applikations-Filter-Schnittstelle und die Hardware-Filter-Schnittstelle beide vorgesehen und zum Zusammenwirken eingerichtet.
10

Gemäß einer Weiterbildung der Erfindung ist die Filter-Schnittstelle mit einer Benutzer-Ausgabeeinrichtung, beispielsweise einem Lautsprecher, Vibrationsalarm oder dergleichen eines mobilen Endgeräts (z.B. Mobiltelefons),
15 derart gekoppelt, dass anlässlich jedes über die Filterschnittstelle erfolgenden Zugriffs von der unsicheren Laufzeitumgebung in die gesicherte Laufzeitumgebung oder / und von der gesicherten Laufzeitumgebung in die unsichere Laufzeitumgebung, also über die Grenze zwischen sicherer und unsicherer Laufzeitumgebung hinweg, durch die Benutzer-Ausgabeeinrichtung
20 eine für einen Benutzer des Mikroprozessors wahrnehmbare Benutzer-Meldung ausgegeben wird. Hierdurch wird der Benutzer auf jeden über die Grenze zwischen sicherer und unsicherer Laufzeitumgebung hinweg erfolgenden Zugriff aufmerksam gemacht und kann z.B. erkennen, wenn sicherheitskritische Daten eingehen und von der Filter-Schnittstelle an die gesicherte Laufzeitumgebung weitergeleitet werden oder /und wenn Daten
25 aus der gesicherten Laufzeitumgebung an die unsichere Laufzeitumgebung ausgegeben werden. Die Benutzer-Ausgabe kann beispielsweise als Melodie gestaltet sein, die durch das mobile Endgerät ausgegeben wird.

Wahlweise werden für Zugriffe über die Grenze zwischen sicherer und unsicherer Laufzeitumgebung hinweg, einerseits von der unsicheren Laufzeitumgebung in die gesicherte Laufzeitumgebung und andererseits von der gesicherten Laufzeitumgebung in die unsichere Laufzeitumgebung, unterschiedliche, für den Benutzer unterscheidbare Benutzer-Ausgaben ausgegeben. Beispielsweise werden je nach Zugriffsrichtung, von der unsicheren Laufzeitumgebung in die gesicherte Laufzeitumgebung oder von der gesicherten Laufzeitumgebung in die unsichere Laufzeitumgebung, unterschiedliche Melodien ausgegeben.

10

Gemäß einer Variante wird die Benutzer-Meldung stets während des Betriebs des Mikroprozessors innerhalb der gesicherten Laufzeitumgebung betrieben. Bei dieser Variante wird also die anlässlich eines Zugriffs von der unsicheren Laufzeitumgebung in die gesicherte Laufzeitumgebung ausgegebene Benutzer-Meldung zeitlich unmittelbar nach dem Zugriff ausgegeben, um dem Benutzer mitzuteilen, dass der Betrieb nun tatsächlich in der gesicherten Laufzeitumgebung erfolgt. Anlässlich eines Zugriffs von der gesicherten Laufzeitumgebung in die unsichere Laufzeitumgebung wird bei dieser Variante die Benutzer-Meldung zeitlich unmittelbar vor dem Zugriff ausgegeben, um den Benutzer zu warnen, dass die gesicherte Laufzeitumgebung sogleich verlassen wird. Alternativ oder zusätzlich kann eine Benutzer-Meldung ausgegeben werden, nachdem durch den Programmablauf die gesicherte Laufzeitumgebung tatsächlich verlassen worden ist. Ebenso kann alternativ oder zusätzlich eine Benutzer-Meldung ausgegeben werden, bevor durch den Programmablauf die gesicherte Laufzeitumgebung betreten wird.

Im Folgenden wird die Erfindung an Hand von Ausführungsbeispielen und unter Bezugnahme auf die Zeichnung näher erläutert, in der in schematischer Darstellung zeigen:

Fig. 1 ein Mikroprozessorsystem mit einer Applikations-Filter-Schnittstelle, gemäß einer Ausführungsform der Erfindung;

Fig. 2 ein Mikroprozessorsystem mit einer Hardware-Filter-Schnittstelle, gemäß einer Ausführungsform der Erfindung;

Fig. 3 ein Mikroprozessorsystem mit einer Applikations-Filter-Schnittstelle und einer Hardware-Filter-Schnittstelle, gemäß einer Ausführungsform der Erfindung.

- 10 Fig. 1 zeigt ein Mikroprozessorsystem mit einer ARM-Architektur, mit einer unsicheren Laufzeitumgebung 100 und einer gesicherten Laufzeitumgebung 200, und mit einer je zum Teil in der unsicheren Laufzeitumgebung 100 und der sicheren Laufzeitumgebung 200 implementierten Applikations-Filter-Schnittstelle TLC (Trustlet Connector), gemäß einer Ausführungsform der
- 15 Erfindung. Die unsichere Laufzeitumgebung 100 ist durch ein Normalbetriebssystem HLOS gesteuert. Die gesicherte Laufzeitumgebung 200 ist durch ein Sicherheitsbetriebssystem MobiCore gesteuert. Zugriffe zwischen der unsicheren Laufzeitumgebung 100 und der gesicherten Laufzeitumgebung 200 auf Betriebssystemebene erfolgen über einen Sicherheitstreiber
- 20 MobiCore-Treiber. Unter dem Sicherheitsbetriebssystem MobiCore, lauffähig in der gesicherten Laufzeitumgebung 200, ist der überwiegende Teil APP-S einer Applikation APP, beispielsweise einer Zahlungsverkehr-Applikation, implementiert. Unter dem Normalbetriebssystem HLOS, lauffähig in der unsicheren Laufzeitumgebung 100, ist der restliche Teil APP-N
- 25 der Applikation APP, beispielsweise Zahlungsverkehr-Applikation, implementiert. Die erfindungsgemäße Applikations-Filter-Schnittstelle TLC (Trustlet Connector) ermöglicht einen direkten Zugriff zwischen dem unter dem Sicherheitsbetriebssystem MobiCore, lauffähig in der gesicherten Laufzeitumgebung 200, implementierten Teil APP-S der Applikation APP und

- dem unter dem Normalbetriebssystem HLOS, lauffähig in der unsicheren Laufzeitumgebung 100, implementierten Teil APP-N der Applikation APP, unmittelbar auf Applikationsebene. Ein Treiber für eine Peripherie-Struktur PER, beispielsweise ein Touch-Screen-Display, ist gemäß Fig. 1 auf her-
- 5 kömmliche Weise unter dem Normalbetriebssystem HLOS implementiert und kann höchstens über den Sicherheitstreiber MobiCore Treiber mit der gesicherten Laufzeitumgebung 200 kommunizieren. Ein neuartiger, unter dem Sicherheitsbetriebssystem MobiCore implementierter reduzierter Treiber PER-S kann aus der unsicheren Laufzeitumgebung 100 heraus höchstens
- 10 über den Sicherheitstreiber MobiCore Treiber erreicht werden. In der unsicheren Laufzeitumgebung 100 kommuniziert der Peripherie-Treiber PER über ein I2C Bussystem, gesteuert durch den Linux I2C Treiber, mit dem Normalbetriebssystem HLOS. In der gesicherten Laufzeitumgebung 200 kommuniziert der reduzierte Peripherie-Treiber PER-S über ein in der gesi-
- 15 cherten Laufzeitumgebung 200 implementiertes I2C Bussystem, gesteuert durch den dortigen I2C Treiber, mit dem Sicherheitsbetriebssystem MobiCore. Kommunikation zwischen den Peripherie-Treibern PER und PER-S kann höchstens auf Betriebssystemebene über den MobiCore Treiber erfolgen.
- 20 Fig. 2 zeigt ein Mikroprozessorsystem mit einer Hardware-Filter-Schnittstelle, nämlich dem virtuellen I2C Treiber, gemäß einer Ausführungsform der Erfindung. Teile mit gleicher Bezeichnung wie in Fig. 1 entsprechen den Teilen aus Fig. 1, soweit nicht hier anderweitig beschrieben. Unter dem Normalbetriebssystem HLOS ist ein Treiber für eine Peripherie-Struktur
- 25 PER, beispielsweise ein Touch-Screen-Display, implementiert. Ein neuartiger reduzierter Peripherie-Treiber PER-S für die Peripherie-Struktur ist unter dem Sicherheitsbetriebssystem MobiCore implementiert. Der je zum Teil in der unsicheren Laufzeitumgebung 100 und der sicheren Laufzeitumgebung 200 implementierte virtuelle I2C Treiber (beispielhafte Hardware-Filter-

Schnittstelle) ermöglicht eine gesicherte Kommunikation zwischen dem unter dem Normalbetriebssystem HLOS implementierten Peripherie-Treiber PER und dem unter dem Sicherheitsbetriebssystem implementierten reduzierten Peripherie-Treiber PER-S. MobiCore auf Hardware-Ebene, genauer auf der Ebene des I2C Busses. In der unsicheren Laufzeitumgebung 100 kommuniziert der Peripherie-Treiber PER über den Linux I2C Treiber mit dem Normalbetriebssystem HLOS. In der gesicherten Laufzeitumgebung 200 kommuniziert der reduzierte Peripherie-Treiber PER-S über den dortigen I2C Treiber mit dem Sicherheitsbetriebssystem MobiCore. Die Hardware-Filter-Schnittstelle, nämlich der virtuelle I2C Treiber, stellt einen gesicherten Kommunikationskanal zwischen dem Linux I2C Treiber des Bussystems unter HLOS und dem I2C Treiber des Bussystems unter MobiCore zur Verfügung, auf einer Hardware-Ebene, die näher an der Ebene des Peripherie-Treibers ist als die Betriebssystem-Ebene. Im Ausführungsbeispiel aus Fig. 2 können die Teile APP-N, APP-S der Applikation APP nur höchstens auf Betriebssystemebene miteinander kommunizieren, nämlich über den MobiCore-Treiber.

Fig. 3 zeigt ein Mikroprozessorsystem mit einer Applikations-Filter-Schnittstelle TLC und einer Hardware-Filter-Schnittstelle virtueller I2C Treiber, gemäß einer Ausführungsform der Erfindung. Fig. 3 stellt somit einerseits eine Kombination der Ausführungsformen aus Fig. 1 und Fig. 2 dar. Darüber hinaus arbeiten die Applikations-Filter-Schnittstelle TLC und die Hardware-Filter-Schnittstelle virtueller I2C Treiber zusammen. Auf Seiten der unsicheren Laufzeitumgebung 100 kommuniziert die Hardware-Filter-Schnittstelle (virtueller I2C Treiber) über das Bussystem Linux I2C Treiber mit dem Peripherie-Treiber PER. Zur gesicherten Laufzeitumgebung 200 hin kommuniziert die Hardware-Filter-Schnittstelle (virtueller I2C Treiber) mit der Applikations-Filter-Schnittstelle TLC und wechselt so von der Hard-

ware-Ebene auf die Applikations-Ebene. Die Applikations-Filter-Schnittstelle TLC kommuniziert schließlich in die gesicherte Laufzeitumgebung 200 hinein auf Applikations-Ebene mit dem reduzierten Peripherie-Treiber PER-S.

Patentansprüche

1. Mikroprozessorsystem, in einem mobilen Endgerät implementierbar oder implementiert, und das aufweist:
- 5 - ein Normalbetriebssystem (HLOS), das zum Erzeugen und Unterhalten einer unsicheren Laufzeitumgebung (100) eingerichtet ist,
- ein Sicherheitsbetriebssystem (MobiCore), das zum Erzeugen und Unterhalten einer gesicherten Laufzeitumgebung (200) eingerichtet ist, und
- eine Betriebssystem-Schnittstelle (MobiCore-Treiber) zwischen dem
- 10 Normalbetriebssystem (HLOS) und dem Sicherheitsbetriebssystem (MobiCore), die dazu eingerichtet ist, Kommunikation zwischen der unsicheren Laufzeitumgebung (100) und der gesicherten Laufzeitumgebung (200) auf Betriebssystem-Ebene gesichert zu steuern,
- gekennzeichnet durch**
- 15 mindestens eine Filter-Schnittstelle (virtueller I2C Treiber; TLC), die dazu eingerichtet ist, Kommunikation zwischen der unsicheren Laufzeitumgebung (100) und der gesicherten Laufzeitumgebung (200) auf einer von der Betriebssystem-Ebene unterschiedlichen Ebene gesichert zu steuern.
- 20 2. Mikroprozessorsystem nach Anspruch 1, wobei als Filter-Schnittstelle eine Hardware-Filter-Schnittstelle (virtueller I2C Treiber) vorgesehen ist, die dazu eingerichtet ist, Kommunikation zwischen der unsicheren Laufzeitumgebung (100) und der gesicherten Laufzeitumgebung (200) auf Hardware-Ebene gesichert zu steuern.
- 25 3. Mikroprozessorsystem nach Anspruch 2, wobei die Steuerung auf Hardware-Ebene als Steuerung über ein Bussystem (I2C) erfolgt.
4. Mikroprozessorsystem nach Anspruch 2 oder 3, das weiter aufweist:
- 30 zumindest einen durch das Normalbetriebssystem (HLOS) ansteuerbaren, in der unsicheren Laufzeitumgebung (100) lauffähigen Treiber (PER) für eine

Peripherie-Struktur des mobilen Endgeräts, und

weiter gekennzeichnet durch

einen durch das Sicherheitsbetriebssystem (MobiCore) ansteuerbaren, in der gesicherten Laufzeitumgebung (200) lauffähigen, vorzugsweise reduzierten

- 5 Treiber (PER-S) für die Peripherie-Struktur, der vorzugsweise lediglich eine Teilmenge der Funktionalitäten hat, welche der in der unsicheren Laufzeitumgebung (100) lauffähige Treiber (PER) hat,

und dadurch, dass

die Hardware-Filter-Schnittstelle (virtueller I2C Treiber) dazu eingerichtet

- 10 ist, Kommunikation zwischen dem in der unsicheren Laufzeitumgebung (100) lauffähigen Treiber (PER) für eine Peripherie-Struktur und dem in der gesicherten Laufzeitumgebung (200) lauffähigen vorzugsweise reduzierten Treiber (PER-S) für die Peripherie-Struktur auf Hardware-Ebene gesichert zu steuern.

15

5. Mikroprozessorsystem nach Anspruch 4, wobei die Hardware-Filter-Schnittstelle (virtueller I2C Treiber) dazu eingerichtet ist, an den Treiber (PER, PER-S) für eine Peripherie-Struktur gerichtete Zugriffe auf Sicherheitsrelevanz hin zu prüfen und im Fall, dass der Zugriff als sicherheitsrelevant

20 erkannt wird, den Zugriff an den vorzugsweise reduzierten Treiber (PER-S) in der gesicherten Laufzeitumgebung (200) zu leiten, und im Fall, dass der Zugriff als nicht sicherheitsrelevant erkannt wird, den Zugriff an den Treiber (PER) in der unsicheren Laufzeitumgebung (100) zu leiten.

25

6. Mikroprozessorsystem nach einem der Ansprüche 1 bis 5, wobei als Filter-Schnittstelle eine Applikations-Filter-Schnittstelle (TLC) vorgesehen ist, die dazu eingerichtet ist, Kommunikation zwischen der unsicheren Laufzeitumgebung (100) und der gesicherten Laufzeitumgebung (200) auf Applikations-Ebene gesichert zu steuern.

7. Mikroprozessorsystem nach Anspruch 6, das weiter aufweist:

zumindest eine Applikation (APP),

dadurch gekennzeichnet, dass

- 5 die Applikation (APP) zumindest teilweise (APP-N) durch das Normalbetriebssystem (HLOS) ansteuerbar und in der unsicheren Laufzeitumgebung (100) lauffähig ist, und zumindest teilweise (APP-S) durch das Sicherheitsbetriebssystem (MobiCore) ansteuerbar und in der gesicherten Laufzeitumgebung (200) lauffähig ist,

- 10 **und dadurch, dass**

die Applikations-Filter-Schnittstelle (TLC) dazu eingerichtet ist, Kommunikation zwischen der in der unsicheren Laufzeitumgebung (100) lauffähigen zumindest teilweisen Applikation (APP-N) und der in der gesicherten Laufzeitumgebung (200) lauffähigen zumindest teilweisen Applikation (APP-S)

- 15 auf Applikations-Ebene gesichert zu steuern.

8. Mikroprozessorsystem nach Anspruch 7, wobei die Applikations-Filter-Schnittstelle (TLC) dazu eingerichtet ist, an die Applikation (APP) gerichtete Zugriffe auf Sicherheitsrelevanz hin zu prüfen und im Fall, dass der Zugriff
20 als sicherheitsrelevant erkannt wird, den Zugriff an die zumindest teilweise Applikation (APP-S) in der gesicherten Laufzeitumgebung (200) zu leiten, und im Fall, dass der Zugriff als nicht sicherheitsrelevant erkannt wird, den Zugriff an die zumindest teilweise Applikation (APP-N) in der unsicheren Laufzeitumgebung (100) zu leiten.

25

9. Mikroprozessorsystem nach einem der Ansprüche 1 bis 8, wobei die Filter-Schnittstelle (virtueller I2C Treiber; TLC) mit einer Benutzer-Ausgabeeinrichtung derart gekoppelt ist, dass anlässlich jedes über die Filter-Schnittstelle (virtueller I2C Treiber; TLC) erfolgenden Zugriffs von der

unsicheren Laufzeitumgebung (100) in die gesicherte Laufzeitumgebung (200) oder / und von der gesicherten Laufzeitumgebung (200) in die unsichere Laufzeitumgebung (100) durch die Benutzer-Ausgabeeinrichtung eine für einen Benutzer des Mikroprozessors wahrnehmbare Benutzer-Meldung
5 ausgegeben wird.

10. Mikroprozessorsystem nach Anspruch 9, wobei die Filter-Schnittstelle (virtueller I2C Treiber; TLC) mit der Benutzer-Ausgabeeinrichtung derart gekoppelt ist, dass für Zugriffe einerseits von der unsicheren Laufzeitumgebung (100) in die gesicherte Laufzeitumgebung (200) und andererseits von
10 der gesicherten Laufzeitumgebung (200) in die unsichere Laufzeitumgebung (100), unterschiedliche, für den Benutzer unterscheidbare Benutzer-Ausgaben ausgegeben werden.

1/3

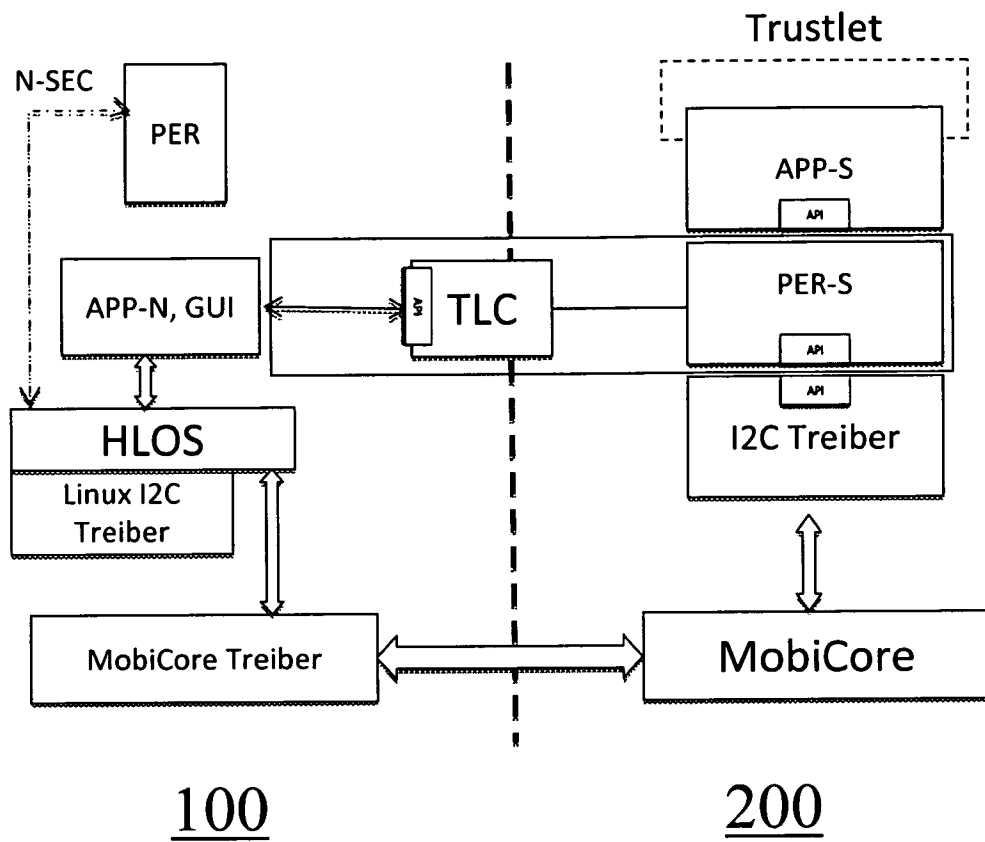


Fig. 1

2/3

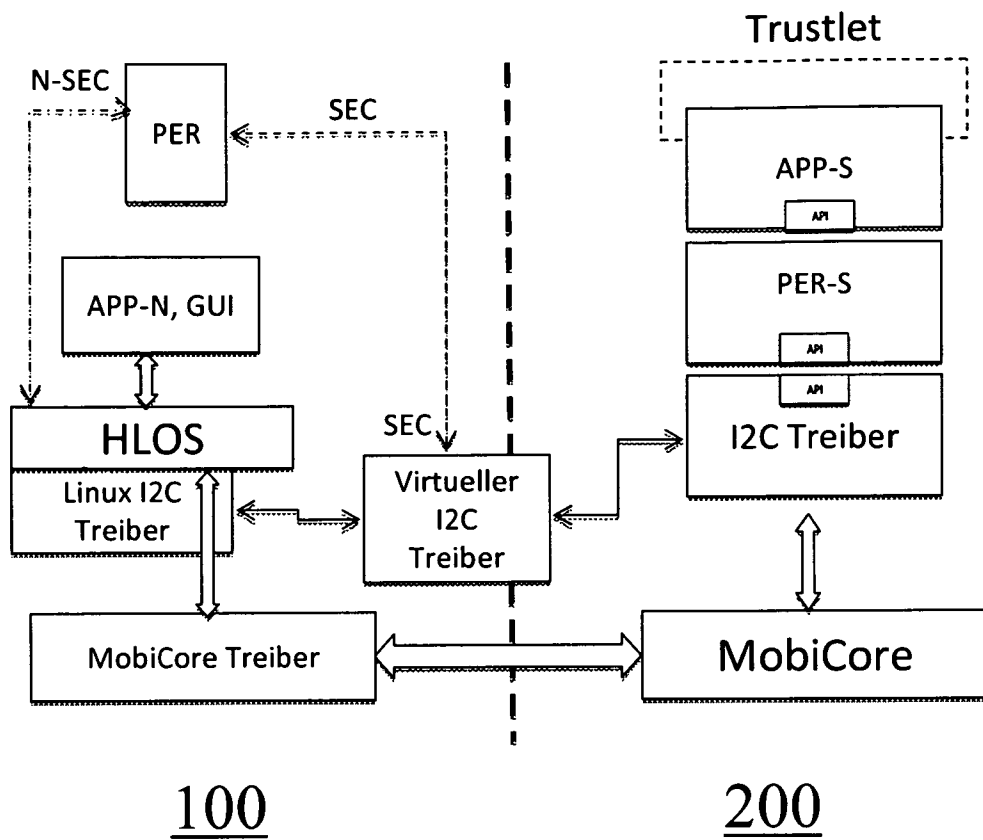


Fig. 2

3/3

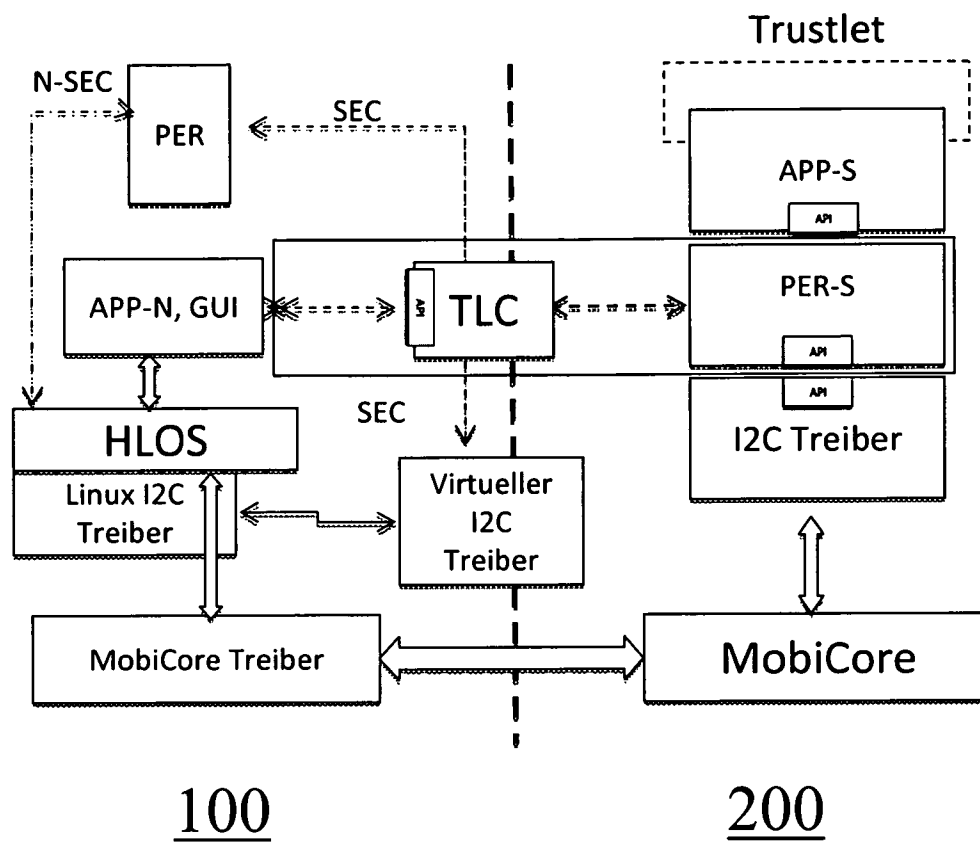


Fig. 3

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2012/004159

A. CLASSIFICATION OF SUBJECT MATTER
 INV. G06F21/53 G06F9/54
 ADD.

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, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2005/033972 A1 (WATSON SCOTT F [US]) 10 February 2005 (2005-02-10) the whole document	1-10
Y	WEI HUANG ET AL: "Virtual machine aware communication libraries for high performance computing", PROCEEDINGS OF THE 2007 ACM/IEEE CONFERENCE ON SUPERCOMPUTING, 2007, SC'07 : DATE: 10 - 16 NOV. 2007, RENO, NV, USA, IEEE, PISCATAWAY, NJ, USA, 10 November 2007 (2007-11-10), pages 1-12, XP031577205, ISBN: 978-1-59593-764-3 paragraph [0004]; figure 3 ----- -/-	1-10



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

18 December 2012

Date of mailing of the international search report

04/01/2013

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
 NL - 2280 HV Rijswijk
 Tel. (+31-70) 340-2040,
 Fax: (+31-70) 340-3016

Authorized officer

Mäenpää, Jari

INTERNATIONAL SEARCH REPORT

International application No

PCT/EP2012/004159

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	ARM LIMITED: "ARM Security Technology - Building a Secure System using TrustZone Technology", INTERNET CITATION, 30 April 2009 (2009-04-30), pages I-XII,1, XP002660015, Retrieved from the Internet: URL:http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf [retrieved on 2011-09-27] the whole document -----	1-10
A	GIESECKE & DEVRIENT: "Mobicore - Giesecke & Devrient's Secure OS for ARM TrustZone Technology (White Paper)", INTERNET CITATION, 1 January 2010 (2010-01-01), pages 1-8, XP002688064, Retrieved from the Internet: URL:http://www.gi-de.com/gd_media/media/en/documents/brochures/mobile_security_2/MobiCore_EN.pdf [retrieved on 2012-11-27] the whole document -----	1-10

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/EP2012/004159

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2005033972 A1	10-02-2005	AU 2004252921 A1	06-01-2005
		CA 2530441 A1	06-01-2005
		EP 1644802 A2	12-04-2006
		JP 2007529044 A	18-10-2007
		KR 20060039405 A	08-05-2006
		US 2005033972 A1	10-02-2005
		WO 2005001666 A2	06-01-2005

INTERNATIONALER RECHERCHENBERICHT

Internationales Aktenzeichen

PCT/EP2012/004159

A. KLASSIFIZIERUNG DES ANMELDUNGSGEGENSTANDES
INV. G06F21/53 G06F9/54
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, WPI Data

C. ALS WESENTLICH ANGESEHENE UNTERLAGEN

Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
Y	US 2005/033972 A1 (WATSON SCOTT F [US]) 10. Februar 2005 (2005-02-10) das ganze Dokument	1-10
Y	WEI HUANG ET AL: "Virtual machine aware communication libraries for high performance computing", PROCEEDINGS OF THE 2007 ACM/IEEE CONFERENCE ON SUPERCOMPUTING, 2007, SC'07 : DATE: 10 - 16 NOV. 2007, RENO, NV, USA, IEEE, PISCATAWAY, NJ, USA, 10. November 2007 (2007-11-10), Seiten 1-12, XP031577205, ISBN: 978-1-59593-764-3 Absatz [0004]; Abbildung 3 ----- -/-	1-10

☒ 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
18. Dezember 2012	04/01/2013
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 Mäenpää, Jari

C. (Fortsetzung) ALS WESENTLICH ANGESEHENE UNTERLAGEN		
Kategorie*	Bezeichnung der Veröffentlichung, soweit erforderlich unter Angabe der in Betracht kommenden Teile	Betr. Anspruch Nr.
A	ARM LIMITED: "ARM Security Technology - Building a Secure System using TrustZone Technology", INTERNET CITATION, 30. April 2009 (2009-04-30), Seiten I-XII,1, XP002660015, Gefunden im Internet: URL:http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf [gefunden am 2011-09-27] das ganze Dokument -----	1-10
A	GIESECKE & DEVRIENT: "Mobicore - Giesecke & Devrient's Secure OS for ARM TrustZone Technology (White Paper)", INTERNET CITATION, 1. Januar 2010 (2010-01-01), Seiten 1-8, XP002688064, Gefunden im Internet: URL:http://www.gi-de.com/gd_media/media/en/documents/brochures/mobile_security_2/MobiCore_EN.pdf [gefunden am 2012-11-27] das ganze Dokument -----	1-10

INTERNATIONALER RECHERCHENBERICHT

Angaben zu Veröffentlichungen, die zur selben Patentfamilie gehören

Internationales Aktenzeichen

PCT/EP2012/004159

Im Recherchenbericht angeführtes Patentdokument	Datum der Veröffentlichung	Mitglied(er) der Patentfamilie	Datum der Veröffentlichung
US 2005033972 A1	10-02-2005	AU 2004252921 A1	06-01-2005
		CA 2530441 A1	06-01-2005
		EP 1644802 A2	12-04-2006
		JP 2007529044 A	18-10-2007
		KR 20060039405 A	08-05-2006
		US 2005033972 A1	10-02-2005
		WO 2005001666 A2	06-01-2005
