(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5607170号
(P5607170)
(45)発行日　平成26年10月15日(2014.10.15)　(24)登録日　平成26年9月5日(2014.9.5)

(51)Int.Cl.　　F I
G06F 21/53 (2013.01)　　G06F 21/00 153
G06K 19/07 (2006.01)　　G06K 19/00 N
G06F 21/77 (2013.01)　　G06F 21/02 177

請求項の数 18　(全 18 頁)

(21)出願番号　特願2012-537321 (P2012-537321)
(86)(22)出願日　平成22年11月3日(2010.11.3)
(65)公表番号　特表2013-510352 (P2013-510352A)
(43)公表日　平成25年3月21日(2013.3.21)
(86)国際出願番号　PCT/EP2010/006693
(87)国際公開番号　WO2011/054498
(87)国際公開日　平成23年5月12日(2011.5.12)
審査請求日　平成24年7月2日(2012.7.2)
(31)優先権主張番号　0905312
(32)優先日　平成21年11月5日(2009.11.5)
(33)優先権主張国　フランス(FR)

(73)特許権者　512116055
トラステツド・ロジツク
フランス国、エフ－９２１９０・ムドン、リユ・ドウ・ラ・ベルリ、６
(74)代理人　110001173
特許業務法人川口國際特許事務所
(72)発明者　ルノー，ニコラ
フランス国、エフ－９２１９０・ムドン、リユ・ドウ・ラ・ベルリ、６、トラステツド・ロジツク気付
(72)発明者　ペテイラード，エリツク
フランス国、エフ－９２１９０・ムドン、リユ・ドウ・ラ・ベルリ、６、トラステツド・ロジツク気付

最終頁に続く

(54)【発明の名称】安全なポータブルオブジェクト

(57)【特許請求の範囲】
【請求項１】
（ａ）オブジェクト本体と、（ｂ）プロセッサ、ならびに同じアプリケーションフォーマットを有する第１および第２のアプリケーションが記憶される１つまたは複数のメモリを含むマイクロモジュールとを含む、スマートカードタイプの安全なポータブルオブジェクトであって、前記第１および第２のアプリケーションを実行することができる第１および第２の実行エンジンをさらに含むこと、および前記第１のアプリケーションは、第１の実行空間内で前記第１の実行エンジンにより実行され、前記第２のアプリケーションは、第１の実行空間と異なる第２の実行空間内で前記第２の実行エンジンにより実行されることを特徴とする、安全なポータブルオブジェクト。
【請求項２】
第２の実行エンジンが第１の実行エンジンと異なることを特徴とする、請求項１に記載のポータブルオブジェクト。
【請求項３】
第１の実行エンジンが、第１の仮想マシン（ＶＭ１）の実行エンジン（インタプリタ１）であること、および第２の実行エンジンが、第２の仮想マシン（ＶＭ２）の実行エンジン（インタプリタ２）であることを特徴とする、請求項１また２のいずれか一項に記載のポータブルオブジェクト。
【請求項４】
仮想マシン（ＶＭ１、ＶＭ２）が、Ｊａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンであること

、および第１および第２のアプリケーションが、Ｊａｖａ　Ｃａｒｄ（ＴＭ）アプリケーションであることを特徴とする、請求項３に記載のポータブルオブジェクト。
【請求項５】
仮想マシンの実行空間内で仮想マシンにより管理されるアプリケーションを実行するために仮想マシンにより必要とされるセキュリティ手段が、仮想マシンごとに異なること、およびアプリケーションは、アプリケーションの実行に必要とされるセキュリティレベルに基づき、前記仮想マシン内に配置されることを特徴とする、請求項３または４のいずれか一項に記載のポータブルオブジェクト。
【請求項６】
仮想マシンの実行空間内で仮想マシンにより管理されるアプリケーションを実行するために仮想マシンにより必要とされるセキュリティ手段が、仮想マシンごとに異なること、およびアプリケーションは、アプリケーションの認証、またはアプリケーションの検証結果に基づき、前記仮想マシン内に配置されることを特徴とする、請求項３または４のいずれか一項に記載のポータブルオブジェクト。
【請求項７】
実行空間が特定の資源に及ぶこと、および別の実行空間が前記特定の資源に及ばないことを特徴とする、請求項１から６のいずれか一項に記載のポータブルオブジェクト。
【請求項８】
特定の資源が、メモリゾーン内に記憶されるデータを伴う前記メモリゾーン、通信チャネル、通信またはデータの記憶装置、あるいはコードライブラリであることを特徴とする、請求項７に記載のポータブルオブジェクト。
【請求項９】
仮想マシン（ＶＭ１、ＶＭ２）の資源が共有されることを特徴とする、請求項３から８のいずれか一項に記載のポータブルオブジェクト。
【請求項１０】
共有資源が、アプリケーションに充てられるメモリ（ヒープ）、または仮想マシンに共通の一部のコードライブラリを含むことを特徴とする、請求項９に記載のポータブルオブジェクト。
【請求項１１】
前記オブジェクトが、近距離無線通信（ｎｅａｒ　ｆｉｅｌｄ　ｒａｄｉｏｆｒｅｑｕｅｎｃｙ　ｃｏｍｍｕｎｉｃａｔｉｏｎ）用モジュールを含む携帯電話に挿入されるように設計される加入者識別モジュールであること、前記加入者識別モジュールは、前記近距離無線通信用モジュールに向けてルーティングされる第１の通信インタフェース、および第２の通信インタフェースを有すること、および第１の通信インタフェースへのアクセスが、第１の実行空間内だけで許可されることを特徴とする、請求項１から１０のいずれか一項に記載のポータブルオブジェクト。
【請求項１２】
第１のアプリケーションが、第２の実行空間内での第２の実行エンジンによる任意の実行を除外して、第１の実行空間内で第１の実行エンジンにより排他的に実行され、第２のアプリケーションが、第２の実行空間内での第１の実行エンジンによる任意の実行を除外して、第２の実行空間内で第２の実行エンジンにより排他的に実行されることを特徴とする、請求項１から１１のいずれか一項に記載のポータブルオブジェクト。
【請求項１３】
（ａ）オブジェクト本体と、（ｂ）プロセッサ、ならびに同じアプリケーションフォーマットを有する第１および第２のアプリケーションが記憶される１つまたは複数のメモリを含むマイクロモジュールとを含む、スマートカードタイプの安全なポータブルオブジェクトを安全にする方法であって、
前記第１および第２のアプリケーションを実行することができる第１および第２の実行エンジンが提供される段階と、
第１の実行エンジンは、第１の実行空間内で第１のアプリケーションを実行する段階と

、
第２の実行エンジンは、第１の実行空間と異なる第２の実行空間内で第２のアプリケーションを実行する段階と
を含むことを特徴とする方法。
【請求項１４】
インストールすべき各アプリケーションに対して割り当てられる実行空間が識別される段階と、
前記アプリケーションは、前記実行空間内で実行されるようにインストールされる段階と
をさらに含むことを特徴とする、請求項１３に記載の方法。
【請求項１５】
実行空間が、少なくとも、前記アプリケーションの提供業者の識別情報により識別されることを特徴とする、請求項１４に記載の方法。
【請求項１６】
実行空間が、少なくとも、前記アプリケーションの検証または認証のレベルにより決定されることを特徴とする、請求項１４に記載の方法。
【請求項１７】
前記第１の実行空間が資源へのアクセスを許可し、前記第２の実行空間が前記資源へのアクセスを許可せず、割り当てるべき前記実行空間は、少なくとも、前記アプリケーションが前記資源にアクセスする必要性により識別されることを特徴とする、請求項１４に記載の方法。
【請求項１８】
前記資源が、前記安全なポータブルオブジェクトの通信インタフェースであることを特徴とする、請求項１７に記載の方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、一般に、標準化された安全なポータブルオブジェクトたとえばスマートカードの分野に関する。そのようなオブジェクトは（ａ）カード本体および（ｂ）マイクロモジュールを含む。マイクロモジュールは、プロセッサ、ならびに第１および第２のアプリケーションが記憶される１つまたは複数のメモリを含む。
【背景技術】
【０００２】
スマートカードは、一般に、特に制限されたハードウェアおよびソフトウェアの資源を備える標準化された安全なポータブルオブジェクトである。
【０００３】
現在利用可能なカードの一部が、特にいわゆるＪａｖａ（登録商標）　Ｃａｒｄ（ＴＭ）カードが、仮想マシンを含む。そのような仮想マシンは実行エンジン（インタプリタ）を含む。実行エンジンは、スマートカードアプリケーションからの命令を解釈し、特に、実際に前記仮想マシンにより規定される実行空間内で命令を実行することができる。
【０００４】
一部のカードが複数のＪａｖａ　Ｃａｒｄ（ＴＭ）アプリケーションを実装する。その場合、すべてのＪａｖａ　Ｃａｒｄ（ＴＭ）アプリケーションが同じ実行空間内部で実行され、しばしば同じセキュリティ規則により支配される。したがって、たとえセキュリティ規則が必要なくても、実行のためにハイレベルのセキュリティを必要とすべきではない一部のアプリケーションが、仮想マシンにより規定される実行空間内部で、アクセスすることができるべきではない機密資源にアクセスすることができる。それはセキュリティ問題につながる。さらに、たとえハイレベルのセキュリティ規則が一部のアプリケーションのためだけに必要であっても、広範な安全対策がすべてのアプリケーションのために実装され、期待される信頼性にかんがみて、そのような対策の適用が正当化されない安全対策

を含む。一実施例として、冗長な計算実行または冗長なデータ記憶が、どんなアプリケーションであるかに関係なく、すべてのアプリケーションに対して使用される。その結果、カード資源を必要としないアプリケーションの実行のためにカード資源が使用され、カード性能に影響を及ぼす。

【発明の概要】

【発明が解決しようとする課題】

【０００５】

上記を考慮して、本発明が解決しようとする１つの技術的問題が、複数のアプリケーションを実行することができるスマートカードタイプの安全なポータブルオブジェクトを作成し、従来技術の前述の欠点を埋め合わせるという技術的問題であり、前記複数のアプリケーションを実行するセキュリティは、特にこれらのアプリケーションが特定の資源へのアクセスを必要としないときに強化され、セキュリティ手段の実装がカード性能に対して与える影響は、前記手段が一部のアプリケーションの実行に対して正当化されないとき、限定される。

【課題を解決するための手段】

【０００６】

上記の技術的問題に対して本発明により提供される解決策は、第一に、（ａ）オブジェクト本体と、（ｂ）プロセッサ、ならびに第１および第２のアプリケーションが記憶される１つまたは複数のメモリを備えるマイクロモジュールとを含む、スマートカードタイプの安全なポータブルオブジェクトであって、前記第１および第２のアプリケーションを実行することができる第１および第２の実行エンジンをさらに含むこと、ならびに前記第１のアプリケーションは、第１の実行空間内で前記第１の実行エンジンにより実行され、前記第２のアプリケーションは、第１の実行空間と異なる第２の実行空間内で前記第２の実行エンジンにより実行されることを特徴とする安全なポータブルオブジェクトを含む。

【０００７】

第二に、本発明により提供される解決策は、（ａ）オブジェクト本体と、（ｂ）プロセッサ、ならびに第１および第２のアプリケーションが記憶される１つまたは複数のメモリとを備えるマイクロモジュールとを含む、スマートカードタイプの安全なポータブルオブジェクトを安全にする方法であって、

前記第１および第２のアプリケーションを実行することができる第１および第２の実行エンジンが提供される段階と、

第１の実行エンジンは、第１の実行空間内で第１のアプリケーションを実行する段階と、

第２の実行エンジンは、第１の実行空間と異なる第２の実行空間内で第２のアプリケーションを実行する段階と

を含むことを特徴とする方法を含む。

【０００８】

したがって、第１のアプリケーションが、たとえば、重要なセキュリティ資源に及ぶ実行空間で実行されなければならない場合、第２のアプリケーションは、前記資源にアクセスすることができず、オブジェクトをより安全にする。さらに、実行エンジンに応じて、アプリケーションごとに異なるセキュリティ手段を実装することが可能であり、したがって、カード性能を改善する。

【０００９】

有利なことに、（ａ）第２の実行エンジンは第１の実行エンジンと異なり、（ｂ）第１の実行エンジンは第１の仮想マシンの実行エンジンであり、第２の実行エンジンは第２の仮想マシンの実行エンジンであり、（ｃ）仮想マシンはＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンであり、第１および第２のアプリケーションはＪａｖａ　Ｃａｒｄ（ＴＭ）アプリケーションであり、（ｄ）仮想マシンが自分の実行空間内で管理するアプリケーションを実行するために仮想マシンにより必要とされるセキュリティ手段は、仮想マシンごとに異なり、アプリケーションは、アプリケーションの実行に必要とされるセキュリティレベル

に応じて、前記仮想マシン内に配置され、（ｄ）仮想マシンにより自分の実行空間内で管理されるアプリケーションを実行するために仮想マシンにより必要とされるセキュリティ手段は、仮想マシンごとに異なり、アプリケーションは、アプリケーションの認証、またはアプリケーションの検証の結果に応じて、前記仮想マシン内に配置され、（ｅ）一方の実行空間が特定の資源に及び、もう一方の実行空間が前記特定の資源に及ばず、（ｆ）特定の資源は、メモリゾーンおよび前記メモリゾーン内に記憶されるデータ、通信チャネル、通信またはデータの記憶装置、あるいはコードライブラリであり、（ｇ）仮想マシンの資源は共有され、（ｈ）共有資源は、アプリケーションに充てられたメモリ、または仮想マシンに共通の一部のコードライブラリを含み、（ｉ）前記オブジェクトは、近距離無線通信（ｎｅａｒ　ｆｉｅｌｄ　ｒａｄｉｏｆｒｅｑｕｅｎｃｙ　ｃｏｍｍｕｎｉｃａｔｉｏｎ）用モジュールを含む携帯電話に挿入されるように設計される加入者識別モジュールであり、前記加入者識別モジュールは、前記近距離無線通信用モジュールにルーティングされる第１の通信インタフェース、および第２の通信インタフェースを有し、第１の通信インタフェースへのアクセスが、第１の実行空間内だけで許可され、（ｊ）第１のアプリケーションは、第２の実行空間内での第２の実行エンジンによる任意の実行を除外して、第１の実行空間内で第１の実行エンジンにより排他的に実行され、第２のアプリケーションは、第２の実行空間内での第１の実行エンジンによる任意の実行を除外して、第２の実行空間内で第２の実行エンジンにより排他的に実行され、（ｋ）本発明による方法は、（ａ）インストールすべき各アプリケーションに割り当てる実行空間が識別され、前記アプリケーションは、前記実行空間内で実行されるようにインストールされるステップと、（ｂ）実行空間が、少なくとも、前記アプリケーションの提供業者の識別情報により識別されるステップと、（ｃ）実行空間が、少なくとも、前記アプリケーションの検証または認証のレベルにより識別されるステップと、（ｄ）前記第１の実行空間が資源へのアクセスを提供し、前記第２の実行空間が前記資源へのアクセスを許可せず、割り当てるべき前記実行空間は、少なくとも、前記アプリケーションが前記資源にアクセスする必要性により識別されるステップと、（ｅ）前記資源は、前記安全なポータブルオブジェクトの通信インタフェースであるステップとをさらに含む。

【００１０】

本発明は、添付図面を参照する、以下の限定しない説明を読んだ後、さらによく理解されよう。

【図面の簡単な説明】

【００１１】

【図１】２つの別個のＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンを含む、本発明によるオブジェクトの実施形態の一実施例の略図である。

【図２】２つの仮想マシン、アプリケーションに充てられた共通に置かれるメモリを含む、本発明によるオブジェクトの実施形態の一実施例の略図である。

【図３】第１の仮想マシンが第１の通信手段にアクセスし、第２の仮想マシンが第２の通信手段にアクセスし、前記第１および第２の通信手段は分かれている２つの仮想マシンを含む、本発明によるオブジェクトの実施形態の一実施例の略図である。

【発明を実施するための形態】

【００１２】

本発明は、スマートカードタイプの安全なポータブルオブジェクトに関する。オブジェクトは、第一に、オブジェクト本体、たとえば標準寸法のプラスチックカード本体、第二に、マイクロモジュールを含む。

【００１３】

本発明によるオブジェクトのマイクロモジュールは、少なくとも１つのプロセッサと、少なくとも２つのアプリケーション、およびアプリケーションのデータが記憶される少なくとも１つのメモリとを含む。２つのアプリケーションはプロセッサにより断続的に実行される。

【００１４】

本発明によるオブジェクトの資源、特に、ハードウェア資源、より具体的にはメモリサイズは制限されている。

【００１５】

本発明によれば、第１のアプリケーションが、第１の実行空間内で第１の実行エンジンにより実行され、第２のアプリケーションが、第２の実行空間内で第２の実行エンジンにより実行される。有利なことに、第１の実行エンジンは、第１の仮想マシンの一部であり、第２の実行エンジンは、第２の仮想マシンの一部である。したがって、本発明によるオブジェクトは、少なくとも２つの仮想マシン、たとえばＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンを含む。２つの仮想マシンは場合によっては、一部の資源を共有することができる。仮想マシンは、異なるレベルのセキュリティを提供することができる。

【００１６】

実行空間

各アプリケーションは、特定の特徴を有する実行空間内で実行されるように設計される。これらの特徴は、たとえば、

実行空間の性質、より詳細には、特定の１組の命令により規定されるプロセッサまたは仮想マシンのタイプ、

特に、データの機密度またはデータの重要性（データの完全性を保護する重要性）に従ってメモリ内の一部のデータを読み出すまたは編集する可能性などの、資源にアクセスする手段、周辺装置または別のシステムとの通信用チャネル、あるいはコードライブラリの提供、

アプリケーションのコード、およびアプリケーションにより処理されるデータを、別のアプリケーションまたは外部の当事者から保護することを意図するセキュリティ手段

を含んでもよい。

【００１７】

本発明は、共通の特徴として少なくとも実行空間の性質を有する実行空間内で実行されるように設計される少なくとも２つのアプリケーションを含むオブジェクトに関する。たとえば、２つアプリケーションは、Ｊａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンで実行されるようにいずれも設計される２つのＪａｖａ　Ｃａｒｄ（ＴＭ）アプリケーションでもよい。２つのアプリケーションの別の特徴に関しては、共通のものがあれば、異なるものもある。

【００１８】

本発明によれば、オブジェクトは、別個の実行空間を有する２つのアプリケーションを提供する。しかしながら、これらの空間は、２つのアプリケーションに共通の要件に応じて、一部の構成要素を実際に共有する。

【００１９】

より一般的には、本発明によるオブジェクトは、いくつかの実行空間を提供することができ、いくつかのアプリケーションの実行を許可することができ、各アプリケーションは、自分自身の実行空間を有する、またはアプリケーションは、基準たとえば必要とされる１組の特徴および必要とされるセキュリティのレベルに従って、より少数の実行空間内でグループ化される。

【００２０】

資源の共有

ポータブルオブジェクトは、アプリケーション間で分配される、量が制限された異なる資源を含む。前記オブジェクトの１つの資源がメモリである。オブジェクトの大部分が、ユーザとの対話、別のコンピュータシステムとのデータの交換、または外部の物理システムに対する動作またはその観測を可能にする追加の機能構成要素をさらに含む。これらの資源はすべて、所与の資源がアプリケーションのために取っておかれたため、またはいくつかのアプリケーション間で共有されるため、アプリケーション間で分配されなければならず、この場合、本発明によるオブジェクトはこの分配を管理する。

【００２１】

メモリ

本発明によるオブジェクトは、１つまたは複数のタイプであってもよい１つまたは複数のメモリを含む。メモリは、たとえばＲＡＭ、ＥＥＰＲＯＭタイプなどの恒久的に書き換え可能なメモリ、またはＲＯＭであってもよい。本発明によるオブジェクトに含まれるメモリの量は特に制限される。特に、ＲＯＭメモリのサイズは約数百Ｋｂであり、ＥＥＰＲＯＭメモリのサイズは約数十Ｋｂである。

【００２２】

各アプリケーションはコンピュータプログラムであり、そのコードは、本発明によるポータブルオブジェクトのメモリ内に記憶される。さらに、各アプリケーションは少なくとも１つのメモリ内にデータを記憶する。

【００２３】

本発明によれば、各実行空間は、各実行空間にはっきり限定して割り当てられた１つまたは複数のメモリ部分を有する。別のメモリ部分は、いくつかの実行空間の間で共有されるソフトウェア構成要素にさらに割り当てられてもよい、またはいくつかの実行空間に同時にアクセス可能な共有メモリゾーンを形成してもよい。

【００２４】

予約資源

一部の資源が、アプリケーションまたはアプリケーションのカテゴリのために予約される。このとき、資源は実行空間により直接管理され、資源を使用する必要がある任意のアプリケーションが、その実行空間により実行される。

【００２５】

一部のオブジェクトは、一部のアプリケーションだけにより正当に使用されてもよい１つまたは複数の通信またはデータの記憶装置を有する。その場合、これらの装置を実行空間内で直接管理することが有利である。装置にアクセスする必要があるアプリケーションは、前記実行空間内部で実行され、一方、その他のアプリケーションは、そのようなアクセスを提供しない実行空間内で実行される。

【００２６】

本発明によるオブジェクトが、承認された提供業者だけにより使用されてもよい資源を含んでもよい。これらの資源は周辺装置であってもよいが、同様に、たとえば使用すると料金の支払いが必要なコードライブラリであってもよい。その場合、承認されたアプリケーションを、関係がある資源を備える実行空間内で実行し、その他のアプリケーションを、前記資源へのアクセスを許可しない実行空間内で実行することが有利である。

【００２７】

共有資源

一部の資源が、いくつかのアプリケーション間で、またはすべてのアプリケーション間でさえ共有される必要がある。このとき、異なる実行空間がそのような資源を使用することを許可し、可能にすることが有利である。それを達成する１つの手段が、実行空間の外部に論理的に位置する特定のソフトウェア構成要素の助けを借りて、関係がある各資源を管理することである。実行空間は、構成要素を使用して、当該の資源にアクセスする。

【００２８】

通信チャネル

潜在的に共有される資源の重要な具体的実例が通信手段である。１つのアプリケーションだけがその通信手段を使用する場合、通信手段に、実行空間により管理されるようにさせ、アプリケーションがその実行空間により実行されることを保証させることが可能である。一方、いくつかのアプリケーションが通信手段を使用する可能性が高い場合、ソフトウェア構成要素が通信手段の共有を許可しなければならない。構成要素は、実行空間の一部であっても（その場合、前記実行空間内で実行されるアプリケーションだけが、前記通信手段にアクセスできる）、またはすべてのアプリケーションが前記通信手段にアクセスすることができるように、実行空間の外部にあり、すべての実行空間により使用可能であってもよい。

【００２９】
本発明の実施形態の一様態では、本発明によるオブジェクト内で実行されてもよいアプリケーションは、識別子、たとえば番号または文字列により示され、通信手段は、メッセージを処理すべきアプリケーションの識別子の形で受信者が指示されるメッセージを受信することができるようにする。その場合、受信されるメッセージを最前線で処理する責任を負うソフトウェア構成要素が、メッセージの受信者であるアプリケーションの識別子を識別するために、受信されるあらゆるメッセージを少なくとも部分的に復号する。前記ソフトウェア構成要素は、メッセージを受信する可能性が高い各アプリケーションに対して、前記アプリケーションの識別子を前記アプリケーションが実行される実行空間の識別子と関連付ける対応表にさらにアクセスすることができる。次に、前記ソフトウェア構成要素は、受信者であるアプリケーションが実行される実行空間にメッセージを転送する。必要であれば、前記ソフトウェア構成要素は、たとえば、前記ソフトウェア構成要素の実行空間をアクティブにし、かつ前記アプリケーションから受信されるメッセージを処理するための手順をトリガすることにより、受信者であるアプリケーションによるメッセージの処理をトリガする。
【００３０】
ソフトウェア構成要素の分配
本発明による安全なポータブルオブジェクトは、同じタイプの少なくとも２つの安全な実行空間を含む。これらの空間は仮想マシンにより規定される。これらの仮想マシンは、たとえばＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンであってもよい。実行空間が、一般にいくつかのソフトウェア構成要素により規定され、ソフトウェア構成要素の中でも、典型的には厳密な意味で仮想マシンにより形成される実行エンジンが識別されることができ、その他の構成要素が、追加要素たとえばデバイスドライバ、メモリマネージャ、セキュリティマネージャ、またはコードライブラリ（ＡＰＩ）として出現する。本発明では、これらの構成要素の一部が、実行空間の間で共有される。換言すれば、前記構成要素のコードはメモリ内に一度だけ記憶される。構成要素に関する限り、別の構成要素が、実行空間に特有である。その点、別の構成要素は、もう一方の構成要素を除外して、一方の実行空間だけにアクセス可能である、またはこれらの別の構成要素のいくつかのバージョンが存在するが、異なる特性を有する。
【００３１】
実行エンジン
本発明では、実行空間は性質が同じである。すなわち、実行空間が実行できるアプリケーションフォーマットは同じである。したがって、特に、可能なときはいつでも、異なるアプリケーションに対して同じ実行エンジンを使用し、したがって、実行エンジンを共有することは当然である。アプリケーションがネイティブフォーマットである場合、すなわちプロセッサ（物理構成要素）により直接実行されることが意図される場合、実行エンジンは、本来共有される前記プロセッサである。アプリケーションが別のフォーマットである場合、すなわち、適合した実行エンジンが仮想マシンである場合、仮想マシンを実装するコードの共有が可能であるが強いられることはない。
【００３２】
本発明の実施形態の一様態では、たとえ実行エンジンが同じアプリケーションを実行することができる同じ基本的機能特性を有しても、実行エンジンは、異なる第２の特徴を有し、したがって、異なるコードを有する。たとえば、一方の実行エンジンが有してもよいが、他方の実行エンジンが有してはいけない第２の特徴の実施例が提示された。
【００３３】
実施形態の一実施例では、実行エンジンが、アプリケーションにより処理されるデータを保護するように設計される追加のセキュリティ手段を含む。たとえば、オブジェクトに記憶されたデータ、または外部システムと通信されたデータは、暗号化されてもよい。これらの安全対策は、計算時間、および場合によってはメモリサイズに関して費用がかかるので、機密データを処理しないアプリケーションに、よりよい性能を提供するために、別

の実行エンジンはそのようなセキュリティ手段を含まない。

【００３４】

実施形態の一実施例では、実行が中断させられた場合、たとえばオブジェクトへの電源が突然遮断された場合、または前記オブジェクトが過熱した場合、またはオブジェクトの物理構成要素が損傷を受けた場合を含み、実行エンジンは、処理されるデータの完全性を保護するように設計される追加のセキュリティ手段を含む。想定される外乱はまた、第三者がアクセスできるべきではないデータを読み出そうとする、またはオブジェクトにより不法行為を達成しようとする、たとえば許可を出されるべきではない銀行取引を許可しようとする第三者からのオブジェクトへの攻撃を含むことがある。そのようなセキュリティ手段は、しばしば冗長な計算の実行または冗長なデータの記憶を含む。したがって、そのようなセキュリティ手段は、期待される信頼性にかんがみて、必ずしも正当化されるわけではない費用がかかる。別の実行エンジンは、データの完全性の損失に対してもよりよい性能を提供するそのような冗長性を提供しない。

【００３５】

実施形態の一実施例では、実行エンジンが、アプリケーションに割り当てられたメモリ部分の外側へのアクセスを防ぐために、実行エンジンにより実行されるアプリケーションによるメモリアクセスの制御を含み、一方、別の実行エンジンは、そのような制御を提供しない。したがって、提供業者が十分信頼できると考えられ、かつアプリケーションに付随する証明書により確認されることができるアプリケーションのために、またはメモリアクセスが自動的にまたは手作業で最初に検証されたアプリケーションのために、より高速に、より小さなメモリ要件でアプリケーションを実行する第２の実行エンジンが取っておかれる。

【００３６】

実施形態の一実施例では、一方の実行エンジンが、他方より高速であるが、一部の物理資源たとえば電力消費に関してコストがより高い。この場合、実行エンジンの選択は、実行性能と物理資源の消費量の間の妥協の結果である。

【００３７】

実施形態の一実施例では、一方の実行エンジンが他方の実行エンジンより効率的であるが、実行エンジンの使用は、実行エンジンにより実行されるアプリケーションの数、あるいは特徴たとえばアプリケーションのサイズまたは複雑さに依存する価格付けの影響を受ける。

【００３８】

たった今説明された、本発明の様態の一変形形態では、実行空間は、同じタイプであるが、一部の詳細が異なる。実行空間は、たとえば、同じ規格を満たすが、必ずしも互いに適合するわけではない何らかの拡張を追加して提供する仮想マシンにより規定されてもよい。このとき、それぞれの不適合の拡張ファミリーのための実行空間を実行エンジンに提供することが有利である。

【００３９】

本発明の別の変形形態では、実行エンジンは、コードの一部を共有する。それは、実行エンジンが十分似せられ、たとえばある種の特定の動作の完了だけが異なるという条件で可能であり、望ましくさえある。本変形形態の実施形態のいくつかの様態が可能である。実施形態の２つの様態では、実行エンジンは、各命令に対して実行される復号と呼ばれるコードの一部が、命令により実行される動作を復号し、各動作に特有なコードの別の一部に制御を渡すことが意図される仮想マシンである。

【００４０】

実施形態の第１の様態では、特に命令復号コードを含む実行エンジンのコアが、２つの実行空間に共通である。メモリセルが、アクティブな実行空間の識別子を含む。復号コードが、実行空間に応じて別々に実行されなければならない命令を検出したとき、アクティブな実行空間内で実行される動作に対応するコードのアドレスをテーブルから読み出し、このように突き止められたコードに制御を渡す。

【００４１】
実施形態の第２の様態では、各実行空間が、各実行空間に特有の実行エンジンを有する。復号コードが、考慮中の両方の実行空間内で同じ方法で実行されなければならない命令を検出したとき、前記復号コードは、２つの実行空間に共通のコードの一部に、実行を渡す。
【００４２】
本発明によれば、仮想マシンは、オブジェクトの同じメモリ内または異なるメモリ内に含まれやすいことに留意されたい。第１の実施例では、２つのＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンが、オブジェクトの単一のＲＯＭメモリ内に含まれる。第２の実施例では、第１の仮想マシンが、前記オブジェクトのＲＯＭメモリ内に含まれ、第２の仮想マシンが、オブジェクトの別のメモリ内に、すなわちＥＥＰＲＯＭメモリ内に含まれる。
【００４３】
追加構成要素
実行空間は実行エンジンまたはその他を共有するということとは無関係に、自分自身の別の追加構成要素を共有する、または有することができる。たとえば、典型的なＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンは、アプリケーションのコードを構成する命令を解釈する責任を負う実行エンジンに加えて、差別化されたライブラリを、たとえばＪａｖａ　Ｃａｒｄ　ＡＰＩ（アプリケーションプログラミングインタフェース）、ファイアウォール、アプリケーションレジストリ、メモリマネージャ、ディスパッチャ、およびアプリケーションの必要性に応じて追加で開発されたライブラリをいくつか含む。場合に応じて、各構成要素が、異なるアプリケーションにより共有される、または各アプリケーションが自分自身の構成要素を有することが好ましいことがある。典型的には、共有はメモリを節約することができるようにし、アプリケーション間の通信を容易にするが、一方、分離は、隔離を増大させ、当該の構成要素の別の実装形態を可能にする。いずれの場合も、共有されることができる構成要素があり、一方、分離される構成要素もあることに留意されたい。
【００４４】
実行空間の選択
本発明では、各アプリケーションが、オブジェクト内で利用可能な実行空間から、実行空間を割り当てられる。この割当ての実施形態のいくつかの様態が可能であり、特に、割当てが識別され、有効になるときが異なる。また、割当ては、いくつかの基準に基づいてもよい。割当方法および割当基準の一部の実施例がここで言及されたが、別の様態が可能であることが理解される。
【００４５】
割当方法
本発明により安全なポータブルオブジェクト上にアプリケーションをインストールするには、少なくとも、オブジェクト内にアプリケーションのコードをロードすることを伴い、その結果、前記コードはオブジェクトのメモリ内に配置される。ローディングは、前記コードを前記メモリ内に記憶するようオブジェクトを適切にプログラムした後、前記コードをオブジェクトに転送することにあってもよいが、別の方法が実行可能であり、特に、アプリケーションのコードは、オブジェクトが作成されたとき、前記オブジェクトのＲＯＭ内に配置されてもよい。本発明では、アプリケーションをインストールする方法は、アプリケーションに割り当てられた実行空間を識別することを伴う。割当空間は、ローディングの前、後、またはローディングと一緒に、１つまたは複数のステップで識別されてもよい。
【００４６】
割当ての第１の様態では、アプリケーションのコードがオブジェクトのメモリ内にロードされたとき、コードは、割当基準に従って識別される実行空間の１つに割り当てられるそのメモリのゾーン内に配置される。メモリは、たとえば、オブジェクトが製造されている間に前記コードがロードされる場合にはＲＯＭでも、オブジェクトが製造された後にコードがロードされる場合にはＥＥＰＲＯＭタイプの書き換え可能メモリでもよい。このと

き、アプリケーションは、アプリケーションをインストールし実行することを担当する実行空間に割り当てられる。

【００４７】

割当ての第２の様態では、アプリケーションのコードがオブジェクトのメモリ内にロードされたとき、アプリケーションに供給されるコード、オブジェクト内にすでに存在するコード、または２つの組合せを実行することを伴うインストール過程が行われる。この方法の間、実行空間は、割当基準に基づき識別され、識別された実行空間とアプリケーションを関連付ける情報がメモリに記憶される。任意選択で、実行空間は、アプリケーションをインストールするための追加手順を実行する。アプリケーションが実行される必要があるとき、実行に責任を負うソフトウェア構成要素が、アプリケーションを実行空間と関連付ける前記情報を読み出し、そのように識別された実行空間内でのアプリケーションの実行をトリガする。

【００４８】

割当ての第３の様態では、オブジェクトのユーザからの明示的要求で、あるいは内部または外部の刺激、たとえば時間イベント（たとえばタイマイベントまたはタイムアウトイベント）あるいは装置または通信手段に対する要求の受信に従って、アプリケーションが実行されるとき、実行に責任を負うソフトウェア構成要素は、割当基準に基づき実行空間を識別し、そのように識別された実行空間によるアプリケーションの実行をトリガする。

【００４９】

割当基準

割当ては、アプリケーションの開発者または提供業者により決定されてもよい。その場合、アプリケーションは、アプリケーションの開発者または提供業者により配布され、オブジェクト内にロードされたとき、アプリケーションの実行可能コードに加えて、使用すべき実行空間を識別する情報の少なくとも１つの断片を含む。

【００５０】

異なる実行空間を、アプリケーションの各開発者または各提供業者に、あるいは開発者または提供業者のグループに割り当てることが可能である。その場合、アプリケーションは、カードの開発者または提供業者を識別する情報の断片と共にカード内にロードされる。

【００５１】

実行空間は、アプリケーションに供給される指示に応じて、アプリケーションをオブジェクト上にインストールする人により、あるいはインストール中またはインストール後にオブジェクト上で自動的に選択されてもよい。情報は、アプリケーションの必要性、またはアプリケーションの実行に関する推奨に関係がある。情報はたとえば、

アプリケーションにより実行されるセキュリティのレベル、
典型的に処理されるデータの機密性のレベル、
アプリケーションにより期待される信頼性のレベル、
アプリケーションが使用する可能性が高い装置、
アプリケーションにより使用される通信手段、
実行エンジンによりサポートされる拡張、
アプリケーションが機能するために必要とされるコードライブラリ、
より一般的には、アプリケーションが機能するために必要とされる任意の資源
と関係があってもよい。

【００５２】

たとえば、第１の実行空間が特定の装置へのアクセスを許可するが、一方、第２の実行空間が同じ装置へのアクセスを拒否する場合、装置へのアクセスを要求するアプリケーションは、第１の実行空間に割り当てられ、一方、その他のアプリケーションは、第２の実行空間に割り当てられる。

【００５３】

実行空間は、何らかの特徴を決定するために、アプリケーションの静的解析により、す

なわち、アプリケーションのコードの、手作業での調査または自動的調査により決定されてもよい。たとえば、以下が行われてもよい：

一部のメモリアクセスが、アプリケーションに許可されるメモリゾーンの起こり得るオーバランから保護されず、静的解析が、許可されたメモリアクセスだけをアプリケーションが実行し、かつすべてのメモリアクセスがアプリケーションに対して検証された実行空間を割り当てることを示し、静的解析が、許可されるゾーンの外側へのメモリアクセスをアプリケーションが実行することができることを示す実行空間を割り当てる、および／または

アプリケーションへの一部の攻撃から保護するための手段を含み、静的解析が、アプリケーションがそのような攻撃を免れず、かつ保護のためのそのような手段を備えない実行空間をアプリケーションに割り当てることを示し、静的解析が、アプリケーションの挙動がそのような攻撃により中断させられないことを示す実行空間を割り当てる。

【００５４】

これらの異なる基準は、異なる方法で組み合わせられてもよいことに留意されたい。たとえば、実行空間が、アプリケーションの一部の提供業者に割り当てられてもよく、別の規則が、未知の提供業者のアプリケーションに割り当てられる実行空間を識別するために使用されてもよい。アプリケーションに任意選択で供給される情報が使用され、情報が見つからない場合、静的解析により実行空間が識別されてもよい。逆に、静的解析が、常に決定的であるわけではないが、あらゆる場合に実行されることができ、静的解析が決定的ではないときだけ、アプリケーションに供給される指示が使用されることができる。ここで説明される選択方法を組み合わせる、識別の別の方法が実行できる。

【００５５】

実施例１：別個の仮想マシン

図１に示される、本発明の実施形態の第１の実施例では、ポータブルオブジェクトは、少なくとも２つのＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンＶＭ１およびＶＭ２が実行されるスマートカードである。

【００５６】

各仮想マシンは、自分自身の実行エンジンであるインタプリタ１、インタプリタ２、自分自身のライブラリＪＣ　ＡＰＩ１、ＪＣ　ＡＰＩ２、および自分自身のメモリ、すなわちアプリケーションに充てられたヒープであるヒープ１、ヒープ２を有する。その構成は、仮想マシンＶＭ１およびＶＭ２が、アプリケーションが自分のヒープであるヒープ１、ヒープ２の外側のメモリにアクセスできないようにするので、第１の実行空間内と第２の実行空間内で実行されるアプリケーション間のハイレベルの隔離を提供するという有利な点を提供する。各アプリケーションは、特定の仮想マシンのヒープ内に記憶され、これらのデータを前記ヒープ内に記憶し、前記仮想マシン内部で実行される。

【００５７】

一部の構成要素が仮想マシン間で共有される。単一のＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンを備える通常のＪａｖａ　Ｃａｒｄ（ＴＭ）スマートカードでは、レジストリは、特に、インストールされたアプリケーションのリストを管理するソフトウェア構成要素であり、各アプリケーションに対する指示、たとえばアプリケーションの可用性（インストール中、実行可能、ブロックされたなど）およびアプリケーションに与えられた許可を含む。実施形態の本実施例では、レジストリは実行空間の間で共有され、通常の情報に加えて、各アプリケーションについて、アプリケーションに割り当てられた実行空間の識別子を保持する。

【００５８】

さまざまなＩＳＯ７８１６、ＳＷＰ、ＵＳＢ、ＲＦ、およびスマートカードの別のインタフェースのためのＩＯ入出力ドライバも仮想マシン間で共有され、その結果、カードは、各インタフェース用に１つの入出力ドライバだけを有する。

【００５９】

スマートカードにより受信されるメッセージは、受信者であるアプリケーションを決定

するために、受信される各メッセージを解析することにより、メッセージをディスパッチし、かつ前記メッセージに対する前記アプリケーションの実行をトリガするソフトウェア構成要素により処理される。受信されるメッセージのそのディスパッチャはまた、仮想マシン間で共有される。メッセージが受信されたとき、受信者であるアプリケーションが識別されると、受信メッセージディスパッチャは、受信者であるアプリケーションが実行される実行空間を識別するために、レジストリに問い合わせて、前記実行空間内部でのアプリケーションによるメッセージの処理をトリガする。

【００６０】

実施例２：ＳＤセキュリティドメイン

仮想マシン間にアプリケーションを分配させるためには、既存の分類インフラストラクチャを使用するのが有利であり、ここで、既存の分類インフラストラクチャの展開について説明される。

【００６１】

図１に示されるように、セキュリティドメインＳＤ１およびＳＤ２と呼ばれる特定のアプリケーションが、仮想マシンＶＭ１およびＶＭ２のそれぞれの中で実行される。セキュリティドメインは、１組のアプリケーションのセキュリティ関連データを管理する。一般に、アプリケーションの各提供業者は、供給業者のアプリケーションすべてで使用されるセキュリティドメインも提供する。セキュリティドメインは、特に提供業者に特有の秘密鍵を保持し、別のアプリケーションのために、特にアプリケーションのインストールおよびデータのローディングのために、一部の動作を実行し、当該の提供業者（または必要とされる鍵が与えられた代表者）だけが、自分自身のアプリケーションに対して作用することができるように、アプリケーションのコードおよびロードされるデータが暗号化され、認証される。

【００６２】

本実施例では、各仮想マシンが１つのセキュリティドメインだけを含むことが有利である。その方法では、各仮想マシンＶＭ１およびＶＭ２は、アプリケーションの提供業者に関連付けられる。実際には、仮想マシンの分離が、第１の仮想マシンＶＭ１上で実行されるアプリケーションと、第２の仮想マシンＶＭ２上で実行されるアプリケーションの間に、ある程度の隔離を提供する。特に、これは、提供業者１のアプリケーションが提供業者２のデータにアクセスすることができないという保証を提供し、逆の場合も同様である。

【００６３】

アプリケーションが、カード内にロードされると、セキュリティドメインがカード内にすでにロードされ、アプリケーションにより参照されると考えられるため、またはセキュリティドメインが当該のアプリケーションと同時にロードされるため、アプリケーションは、関連するセキュリティドメインを有する。カード上のアプリケーションマネージャが、仮想マシン内へのアプリケーションのインストールをトリガし、提供されたセキュリティドメインを動作させる。仮想マシンがカード内にすでに存在していない場合、アプリケーションマネージャは、メモリの一部を取っておき、メモリのその部分を使用する新しい仮想マシンを始動させる。次に、アプリケーションマネージャは、新しい仮想マシン内への新しいセキュリティドメインのインストール、および次に、新しい仮想マシン内への新しくロードされたアプリケーションのインストールをトリガする。セキュリティドメインが供給されず、すでに存在もしていない場合、またはセキュリティドメインが新規であり、供給されているが、新しい仮想マシンが始動させられることができない場合（たとえば、カードがその機能をサポートしていないため、または不十分な量のメモリが利用可能であるため）、アプリケーションのローディングは失敗する。

【００６４】

したがって、本実施例で説明される実装形態に適合するスマートカードが、高いセキュリティ要件を伴うアプリケーションのための媒体の役割を安全に果たすことができる。たとえば、カードは、キャッシュカードの役割（１つの仮想マシン（ＶＭ）が、カード所有者の銀行により提供されるアプリケーションに充てられる）、およびトランスポートカー

ドの役割（１つの仮想マシンが、交通機関の事業者により提供されるアプリケーションに充てられる）を同時に果たしてもよい。さらに、カードは、携帯電話に挿入されるＳＩＭカードとすることができ、通信アプリケーションに充てられる仮想マシンの実行を可能にすることができる。

【００６５】

実施例３：異なる仮想マシン

本実施例では、スマートカードは、各アプリケーションに対して一部の仮想マシンが別の仮想マシンより適切であるように、たとえ仮想マシンが同じアプリケーションを実行することができても、異なる特性を有するいくつかの仮想マシンの実行を可能にする。違いは、異なるアプリケーションのセキュリティ、および異なるアプリケーションにより処理されるデータに関係がある。

【００６６】

スマートカード内にアプリケーションをロードしている間、アプリケーションのコードは、ベリファイア（ｖｅｒｉｆｉｅｒ）と呼ばれるソフトウェア構成要素により解析される。この構成要素は、カード内にロードされ、カードにより実行される、または厳密な意味でロードする前にスマートカードのセキュリティの責任を負うエンティティにより実行される。ベリファイアは、セキュリティの含意を含むアプリケーションのコードの特性を決定するために、アプリケーションのコードの静的解析を実行する。アプリケーションのコードの解析は、アプリケーションのコードに供給される任意選択の注釈によりサポートされてもよい。特に、ベリファイアは、アプリケーションが、許可されないメモリまたは別の資源にアクセスすることができるというリスクがまったくないと結論を下してもよい。したがって、ベリファイアは、安全なアプリケーション（ベリファイアは、禁止されたアクセスがないことを保証する）を安全ではないアプリケーション（保証はまったく提供されない）と区別することにより、またはより正確な結果を提供することにより、アプリケーションのセキュリティのレベルを計算する。

【００６７】

アプリケーションがベリファイアにより十分安全であると識別された場合、中にロードされ、その後、メモリまたは別の資源への一部のアクセス、すなわちベリファイアが不可能であると保証するアクセスを検証しない仮想マシンにより実行される。一方、ベリファイアがそのような保証を提供しない場合、アプリケーションは、潜在的に危険な各アクセスを注意深く検証し、実行中、許可されないアクセスを拒否する別の仮想マシンにより実行される。その方法では、同じスマートカードが、どんな適切なアプリケーションも実行することができるが、より早く安全であると保証されることができるアプリケーションを実行する。

【００６８】

実施形態の一代替形態では、スマートカードは、同じアプリケーションを実行することができる２つの仮想マシンを提供するが、第１の仮想マシンは、第１の仮想マシンで実行されるアプリケーションにより操作されるデータの完全性または機密性に影響を及ぼすことを意図するソフトウェアまたはハードウェアの攻撃から保護するための手段を有し、一方、第２の仮想マシンは、そのような保護手段を有しない。取扱いに注意を要するデータを処理するアプリケーションは、第１の仮想マシンに割り当てられ、一方、その他のアプリケーションは、保護手段がないことにより、より効率的にされる第２の仮想マシンに割り当てられる。第２の仮想マシンはまた、開発中のアプリケーションの試験およびデバッグのために使用されることができる。この最後の使用事例では、両方の仮想マシンは、仮想マシン内で実行されるアプリケーションの機能的挙動の見地からすれば同一であるが、外部からの観察という見地からすれば異なることが必要であることに留意されたい。

【００６９】

実施例４：共有メモリ

図２に提示される実施形態の実施例では、スマートカードは、少なくとも２つのＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンＶＭ１、ＶＭ２を含み、各仮想マシンは、自分自身の実

行エンジンであるインタプリタ１、インタプリタ２、および自分自身のライブラリＪＣ　ＡＰＩ１、ＪＣ　ＡＰＩ２を有する。しかしながら、ヒープアプリケーションに充てられるメモリは共有される。本来、単独である構成要素、たとえばレジストリ、受信メッセージディスパッチャ、およびＩＳＯ７８１６、ＳＷＰ、ＵＳＢ、ＲＦなどのためのＩＯドライバは一般に、異なる仮想マシンにより同様に共有される。

【００７０】

アプリケーションに充てられるメモリの共有は、各仮想マシンがアプリケーションのために自分自身特有のメモリ空間を有する場合に対してメモリを節約することができるようにする。実際には、空きメモリ空間を必要としない一方の仮想マシン内に設置される空きメモリ空間の一部が、他方の仮想マシンにとって失われるので、メモリをそれぞれに分割することは、メモリの損失につながる。そのような共有はまた、メモリ管理に充てられるメモリに関して、スケールメリットを可能にすることがある。この面は、特にＲＡＭに関する資源が制限されるスマートカードにとって、特に重要である。

【００７１】

アプリケーションが記憶されるメモリが共有されるということはまた、アプリケーション間のデータの交換を容易にする。したがって、その構成は、アプリケーション間の通信の可能性が必要とされる場合、およびセキュリティがアプリケーションを互いから隔離することによる以外で提供されることができる場合に好ましい。メモリを共有することはまた、一方のアプリケーションが、もう一方の仮想マシン内で実行されるもう一方のアプリケーションにより提供されるインタフェースを呼び出すことを許可し、したがって、前記もう一方の仮想マシン内で実行されるコードを一方の仮想マシンからトリガすることを許可することができる。

【００７２】

また、たとえ仮想マシンが分かれていても、仮想マシンと独立して実装されるコードの一部のライブラリ、たとえばデータフォーマットを処理するコードのライブラリ、または数値計算ライブラリへの共有アクセスができてもよい。

【００７３】

この実施例の一部として、仮想マシンは異なる可能性を提供する。たとえば、一方の仮想マシンが、いくつかの方法により一部のセキュリティ検証の実行をバイパスすることを意図する物理的攻撃たとえばスマートカードの電源の中断から保護するための手段を有する。仮想マシンは、スマートカード所有者が不法に修正しようとすることができる取扱いに注意を要するデータを処理するアプリケーションの実行に充てられる。もう一方の仮想マシンは、そのような保護の手段を提供せず、したがって、より高速である。もう一方の仮想マシンは、ハイレベルの保護を必要としないアプリケーションのために使用される。

【００７４】

別の実施例では、Ｊａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンは、Ｊａｖａ　Ｃａｒｄ（ＴＭ）言語に不適合でもよい別の拡張を追加する。その場合、言語に特有の拡張を必要とするアプリケーションは、当該の拡張を提供する仮想マシンにより実行される。どんな特定の拡張も必要としないアプリケーションは、任意の仮想マシンにより実行されてもよく、最高の性能を提供する仮想マシンが選択されることが好ましい。

【００７５】

実施例５：別個の通信

図３に示される本実施例では、スマートカードは、携帯電話に挿入されるように設計される加入者識別モジュール（ＳＩＭカード）であり、前記携帯電話は、近距離無線通信用モジュール（ＮＦＣモジュール）を含む。

【００７６】

このとき、カードは、電話機との、より詳細には前記電話機のＮＦＣモジュールとの通信のための第１の論理インタフェースを有する。その第１の論理インタフェースは、ＳＷＰ（Ｓｉｎｇｌｅ　Ｗｉｒｅ　Ｐｒｏｔｏｃｏｌ）プロトコルを使用して携帯電話とのカード通信を実装するインタフェースである。したがって、インタフェースは、前記モジュ

ールにルーティングされ、通信は、ＳＷＰプロトコルを使用して実行される。

【００７７】

さらに、カードは、電話機との、より詳細には前記電話機のメインプロセッサとの通信のための第２の論理インタフェースを有する。その第２の論理インタフェースは、通信プロトコル、または規格ＩＳＯ７８１６で規定されるプロトコルに基づき、電話機とのカード通信を実装するインタフェースである。

【００７８】

本実施例では、スマートカードは、２つの別個のＪａｖａ　Ｃａｒｄ（ＴＭ）仮想マシンＶＭ１およびＶＭ２の実行を可能にする。２つの仮想マシンＶＭ１、ＶＭ２のそれぞれは、自分自身の実行エンジンであるインタプリタ１、インタプリタ２、および自分自身のライブラリＪＣ　ＡＰＩ１、ＪＣ　ＡＰＩ２を有する。第１の仮想マシン（ＶＭ１）は、非接触アプリケーションに、たとえばトランスポートアプリケーションに充てられ、電話機のＮＦＣモジュールとの通信用論理インタフェースに単独でアクセスできる。第２の仮想マシンＶＭ２は、電話機アプリケーションに充てられ、プロセッサの電話機能を実装するために電話機のプロセッサとの通信用インタフェースに単独でアクセスできる。

【００７９】

したがって、第１の通信インタフェースへのアクセスは、第１の実行空間内だけで許可され、一方、第２の通信インタフェースへのアクセスは、第２の実行空間内だけで許可される。

【００８０】

有利なことに、一部の構成要素が、仮想マシン間で共有されたままであることに留意されたい。したがって、カードは、すべての仮想マシン内にインストールされたアプリケーションを列挙するアプリケーションレジストリを１つだけ有する。たとえば、それは、インタフェースを使用して、別のインタフェースと通信するように設計されるアプリケーションをロードすることができるようにする。

【００８１】

ヒープ１、ヒープ２のアプリケーションに充てられるメモリは、仮想マシン間で共有されても、されなくてもよいことにさらに留意されたい。各手法の利点は、上記で概要を述べられる実装形態に関連して説明された。メモリの一部が、特に、２つの実行空間に共通のコードライブラリが常に共有されてもよいが、一方、共有しないことが好ましいデータ、特に、取扱いに注意を要するデータまたは機密データのためのメモリが、各仮想マシン内部に取っておかれる。

【図１】

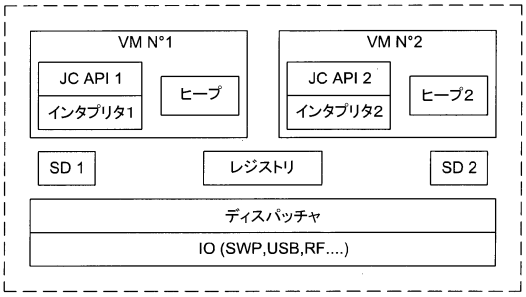

Fig. 1

【図２】

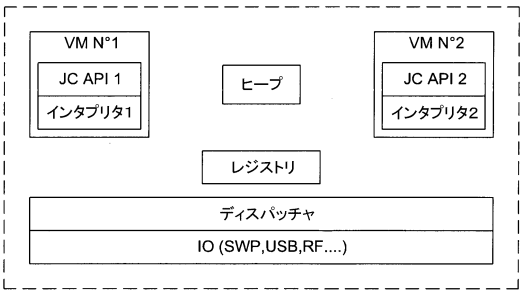

Fig. 2

【図３】

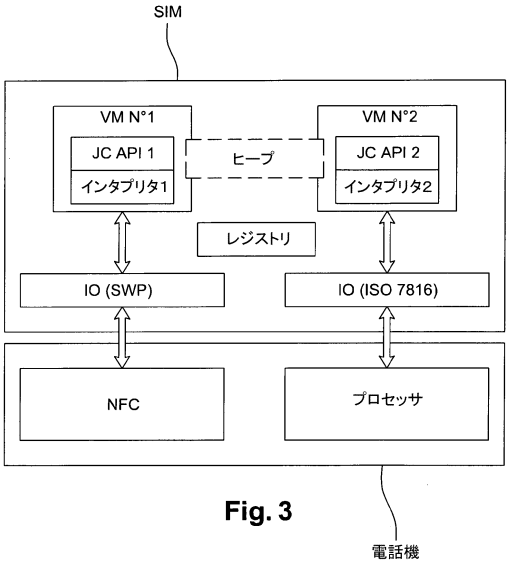

Fig. 3

フロントページの続き

審査官　木村　励

(56)参考文献　特開２００２－３６６９１４（ＪＰ，Ａ）
国際公開第２００８／１４６１２５（WO，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／５０　－　２１／５７
Ｇ０６Ｆ　２１／７０　－　２１／８５
Ｇ０６Ｋ　１９／００　－　１９／１８