(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第4818542号
(P4818542)
(45)発行日　平成23年11月16日(2011.11.16)　(24)登録日　平成23年9月9日(2011.9.9)

(51)Int.Cl.　　F I
G06Q 50/00 (2006.01)　　G06F 17/60　140
G06F 11/34 (2006.01)　　G06F 11/34　C

請求項の数 18　(全 28 頁)

(21)出願番号　特願2001-241933(P2001-241933)
(22)出願日　平成13年8月9日(2001.8.9)
(65)公開番号　特開2002-92221(P2002-92221A)
(43)公開日　平成14年3月29日(2002.3.29)
審査請求日　平成20年8月4日(2008.8.4)
(31)優先権主張番号　0020441.2
(32)優先日　平成12年8月18日(2000.8.18)
(33)優先権主張国　英国(GB)

(73)特許権者　398038580
ヒューレット・パッカード・カンパニー
HEWLETT-PACKARD COMPANY
アメリカ合衆国カリフォルニア州パロアルト　ハノーバー・ストリート　3000
(74)代理人　110000039
特許業務法人アイ・ピー・エス
(72)発明者　グレーメ・ジョン・プラウドラー
イギリス　ブリストル　ストーク・ギフォード　タッチストーン・アヴェニュー　5

審査官　塩田　徳彦

最終頁に続く

(54)【発明の名称】コンピューティングプラットフォームにおけるサービスの実行

(57)【特許請求の範囲】
【請求項１】
それぞれプロセスを実行する複数の第1のコンピューティング環境を提供するコンピューティングプラットフォームにおいてリクエスタに対するサービスを実行する方法であって、
前記リクエスタが、実行される前記サービスに含まれるプロセスの少なくとも一部に対して指定された信頼のレベルを規定する仕様を、前記コンピューティングプラットフォームに対して供給することと、
前記複数の第１のコンピューティング環境それぞれが、前記信頼のレベルが指定されたプロセスを、前記供給された仕様において前記プロセスに対して指定された信頼のレベルに従って実行することと、
前記コンピューティングプラットフォームが、前記信頼のレベルが指定されたプロセスの少なくとも一部の実行のログを取得することと、
前記コンピューティングプラットフォームが、前記信頼のレベルに従って実行されたプロセスの実行のログを、前記リクエスタに提供することと、
を含む方法。
【請求項２】
前記コンピューティングプラットフォームは、前記信頼のレベルが指定された前記プロセスの実行の一部のみのログを取得する
請求項１に記載の方法。

【請求項３】
前記複数の第１のコンピューティング環境は、
その外部から保護された第２のコンピューティング環境
を含む
請求項１または２に記載の方法。
【請求項４】
前記第２のコンピューティング環境は、前記コンピューティングプラットフォームの完全性を測定する監視プロセスを実行する
請求項３に記載の方法。
【請求項５】
前記コンピューティングプラットフォームは、前記プロセスの実行および前記プロセスの実行のログの取得を、前記第１のコンピューティング環境に割付けるサービス管理プロセスを実行する
請求項１～４のいずれかに記載の方法。
【請求項６】
前記サービス管理プロセスは、前記第２のコンピューティング環境に割り付けられて実行される
請求項５に記載の方法。
【請求項７】
前記複数の第１のコンピューティング環境の１つ以上は、コンパートメントであり、
前記コンパートメントは、
動作または環境の制約によって、その外部から受けうる影響に対し保護されたコンピューティングエンジン
を含む
請求項５記載の方法。
【請求項８】
前記コンピューティングエンジンは、Ｊａｖａ仮想マシンである
請求項７記載の方法。
【請求項９】
前記第１のコンピューティング環境内に、１つ以上のコンパートメントが配置される
請求項５に記載の方法。
【請求項１０】
前記コンピューティングエンジンを、許可されない場合は入力データに対して動作しないように制約する
請求項７～９のいずれかに記載の方法。
【請求項１１】
前記入力データにはデータタイプが付され、前記プロセスには動作タイプが付され、前記プロセスは、前記入力データに付されたデータタイプと前記プロセスに付された動作タイプとが適合するときにのみ実行される
請求項１０記載の方法。
【請求項１２】
前記入力データには所有者があり、前記プロセスは、前記所有者に前記入力データの使用を通知するように要求され得る
請求項１０または１１に記載の方法。
【請求項１３】
前記入力データに所有者があるときには、前記プロセスは、前記所有者から前記入力データの使用に対する同意を取得するように要求され得る
請求項１０または１１記載の方法。
【請求項１４】
前記監視プロセスは、前記サービスが実行されたときに、前記コンピューティングプラ

ットフォームの完全性基準を前記リクエスタに提供する
請求項４に記載の方法。
【請求項１５】
保護されて、ユーザに対して信頼できるデータを提供するコンピューティング環境と、
プロセスを実行する１つ以上のコンパートメントと
を備え、
前記保護されたコンピューティング環境は、
リクエスタから、実行されるサービスに含まれるプロセスの少なくとも一部に対して指定される信頼のレベルを規定する仕様の供給を受ける手段と、
前記信頼のレベルが指定されたプロセスを、前記コンパートメントに割り付け、前記コンパートメントに、前記割り付けたプロセスを、前記プロセスに対して指定された信頼のレベルに従って実行させる手段と、
前記プロセスの実行のログの少なくとも一部を取得する手段と、
前記信頼のレベルに従って実行されたプロセスの実行のログを、前記リクエスタに提供する手段と
を含む
コンピューティングプラットフォーム。
【請求項１６】
前記コンパートメントの１つ以上は、前記保護されたコンピューティング環境の内部に配置される
請求項１５に記載のコンピューティングプラットフォーム。
【請求項１７】
前記コンパートメントは、
仮想コンピューティングエンジン
を含む
請求項１５または１６に記載のコンピューティングプラットフォーム。
【請求項１８】
前記仮想コンピューティングエンジンは、Ｊａｖａ仮想マシンである
請求項１７記載のコンピューティングプラットフォーム。
【発明の詳細な説明】
【０００１】
【発明の属する技術分野】
本発明は、コンピューティングプラットフォームにおけるサービスの実行に関し、特にサービスの一部またはすべての信頼性のあるまたはトラステッドな（trusted）実行が要求される場合の、サービスの実行に適したコンピューティングプラットフォームに関する。
【０００２】
【発明が解決しようとする課題】
通常の商業生活では、二者が、一方が他方のためにサービスを実行することに同意する場合、両者の権利および義務を指定するだけでなくサービスがどのように実行されるかという重要な面もまた指定する契約書が、しばしば作成される。また、サービスが満足に実行されたかを決定することができるように、証拠が記録されることも望ましい。ますます、サービスはコンピューティングプラットフォームによって電子的に提供される。契約の使用の適用と契約が満足に実行された証拠を提供するメカニズムとが、現在このコンテキストにおいて欠けている。
【０００３】
【課題を解決するための手段】
通常の商業生活における場合と同じ理由で（両者に対する指導、論議の解決）、契約書と類似するものが存在すること、また契約の実行の証拠が存在することも望ましい。従って、本発明は、コンピューティングプラットフォームにおいてリクエスタに対するサービスを実行する方法であって、リクエスタが、実行されるサービスの仕様をコンピューティン

グプラットフォームに対して供給し、サービスの仕様が、サービスのプロセスの少なくとも一部に対して信頼の指定されたレベルを規定すること、コンピューティングプラットフォームが、仕様に従ってサービスを実行し、信頼のレベルが指定されたプロセスの少なくとも一部の実行をロギングすることと、コンピューティングプラットフォームが、信頼の指定されたレベルに従って実行されるプロセスの実行のログをリクエスタに提供することと、を含む方法を提供する。

【０００４】

このように、本発明により、電子的に受取られた要求に応じてコンピューティングプラットフォーム上での満足のいくサービスの実行の証拠を提供することが可能になる。本サービスは、コンピューティングプラットフォームに対して指定することができ、サービスの結果（これらがリクエスタによって要求される場合であるが、これらは他の場所で要求される場合もある）に加えて、リクエスタには、コンピューティングプラットフォームによってサービスが満足に実行された証拠が提供される。

【０００５】

証拠の提供は、それ自体で重要である。コンピューティングプラットフォームの標準のコンピューティング環境において取得可能であるより一般に大きい、信頼のレベルを要求するものとして、サービスにおける特定のプロセスを指定する能力により、リクエスタには一層大きい値が提供される。これらプロセスが信頼の所望のレベルに従って実行されたという証拠は、サービスの実行が成功したことの貴重な証拠を提供する。

【０００６】

有利には、コンピューティングプラットフォームは、物理的かつ論理的に保護されたコンピューティング環境を含む。これは、コンピューティングプラットフォームの完全性を測定する監視プロセスを行うように適応されてよい。この保護されたコンピューティング環境は、「Trusted Computing Platform」と題され２０００年２月１５日に出願された、本出願人による同時係属の国際特許出願ＰＣＴ／ＧＢ００／００５２８号において述べられるような「トラステッド（信頼される（trusted））コンポーネント」によって提供されるものであってよい。サービスが実行された時のコンピューティングプラットフォームの完全性基準は、所望の信頼のレベルに従ってサービスが実行された証拠と共に、リクエスタに返されてよい。

【０００７】

好ましくは、サービス管理プロセスは、プロセスの実行および実行のロギングを、コンピューティングプラットフォームにおけるまたはそれに関連する別個のコンピューティング環境に割付ける。このプロセスは、保護されたコンピューティング環境内に配置されてよい。

【０００８】

これら別個のコンピューティング環境は、コンパートメントであってよく、その各々が、動作または環境の制約によってコンパートメントの外部からの影響に対して保護されるコンピューティングエンジンを含む。コンパートメントの一例は、Ｊａｖａ仮想マシン（Virtual Machine）を含むＪａｖａサンドボックス（sandbox）である。かかるコンパートメントは、保護されたコンピューティング環境の内側または外側（あるいは両方）に配置することができる。

【０００９】

コンピューティングエンジンは、それらに割付けられるプロセスの実行中に特定の入力データに対し動作を実行しないようにされてもよい。これは、タイピングによって達成されてよい。すなわち、入力データにはデータタイプが提供され、プロセスには動作タイプが提供され、動作タイプおよびデータタイプが一致しない場合はプロセスの動作が行われない。

【００１０】

実行ロギングは、プロセスに対する入力データ、プロセスからの出力データおよびプロセスにおいて実行されるプログラム命令の一部またはすべてのロギングを含んでよい。この

データのすべてをロギングすることができ、あるいはデータの一部またはすべてをサンプルすることができる（特に、プログラム命令データの場合に有用である）。サンプリングは、例えば、その時限りの（nonce）またはランダムな値（それ自体、実行ロギングデータの一部を形成してよい）によって初期化される機能に従って、不規則であることが望ましい。実行ロギングデータは、リクエスタに返される前に暗号化されてよい。実行ロギングパラメータは、オリジナルのサービス仕様において決定されてよく、あるいはサービス管理プロセスによって決定されてよい。

【００１１】

更なる態様において、本発明は、コンピューティングプラットフォームの適当なユーザに対し信頼できるデータを提供するよう適応された、物理的かつ論理的に保護されたコンピューティング環境と、そこで実行されるプロセスが確実に実行されるよう十分に制約された方法で動作するように構成された、１つまたは複数のコンパートメントと、を備え、指定されたプロセスが、１つまたは複数のコンパートメントにおいてユーザに対して実行され、指定されたプロセスの結果が、保護されたコンピューティング環境から信頼できるデータでユーザに返されることが可能なコンピューティングプラットフォームを提供する。

【００１２】

かかるコンピューティングプラットフォームは、特に、ユーザに対してサービスを実行し、サービスのプロセスが信頼の所望のレベルに従って実行された証拠を提供するように、適切に適応される。

【００１３】

ここで、本発明の好ましい実施の形態を、単に例として、添付図面を参照して説明する。

【００１４】

【発明の実施の形態】

本発明の実施の形態を説明する前に、概して本発明の実施の形態を実行するために適したタイプのトラステッド（信頼される（trusted））コンピューティングプラットフォームを、図１ないし図５を参照して説明する。トラステッドコンピューティングプラットフォームのこの説明は、その構成の主要な要素と、そのプラットフォームのユーザに対し、コンピューティングプラットフォームの状態を示す完全性基準を提供する際のその役割と、ユーザに対するかかる基準の通信と、を述べる。このコンテキストにおいて、「ユーザ」は、リモートコンピューティングエンティティ（本発明の実施の形態において、リクエスタはこのカテゴリに入ってよい）等のリモートユーザであってよい。トラステッドコンピューティングプラットフォームは、更に、その内容が引用をもって本明細書内に包含されたものとする、「Trusted Computing Platform」と題され２０００年２月１５日に出願された本出願人による国際特許出願ＰＣＴ／ＧＢ００／００５２８号において述べられている。当業者は、本発明が、その動作に関し、正確に後述するようなトラステッドコンピューティングプラットフォームの使用に頼るものではない、ということを認めるであろう。すなわち、かかるトラステッドコンピューティングプラットフォームを使用することは、本発明において要求される機能を成し遂げる、可能な唯一の方法ではなく１つの方法である。

【００１５】

本明細書で述べる種類のトラステッドコンピューティングプラットフォームは、物理的なトラステッド装置が組込まれたコンピューティングプラットフォームである。このトラステッド装置の機能は、プラットフォームの識別を、そのプラットフォームの完全性基準を提供する確実に測定されたデータと結合するということである。識別および完全性基準は、プラットフォームの信頼性を保証するために用意されているトラステッドパーティ（trusted party（ＴＰ））によって提供される予測された値と比較される。一致がある場合、プラットフォームの少なくとも一部が、完全性基準の範囲によって正確に動作していることを意味する。

【００１６】

ユーザは、プラットフォームと他のデータを交換する前に、プラットフォームの正確な動

作を検査する。ユーザは、これを、トラステッド装置に対しその識別および完全性基準を提供するよう要求することによって行う。（任意に、トラステッド装置は、それ自体がプラットフォームの正確な動作を検査することができなかった場合、識別の証拠を提供するのを拒否する。）ユーザは、識別の証明と識別基準とを受取り、それらを、真であると信じる値と比較する。それら適当な値は、ＴＰかまたはユーザによって信頼される他のエンティティによって提供される。トラステッド装置によって報告されるデータがＴＰによって提供されるデータと同じである場合、ユーザはプラットフォームを信頼する。これは、ユーザがエンティティを信頼するためである。エンティティは、プラットフォームを、それが予め識別を検査しプラットフォームの適当な完全性基準を決定しているために信頼する。

【００１７】

ユーザは、プラットフォームのトラステッド動作を確立すると、プラットフォームと他のデータを交換する。ローカルユーザの場合、その交換は、プラットフォーム上で実行中のあるソフトウェアアプリケーションとの対話によって行われてよい。リモートユーザの場合、概して本発明の実施の形態の場合のように、交換は、セキュア（secure）トランザクションを含んでよい。いずれの場合も、交換されるデータは、トラステッド装置によって「署名（sign）」される。そして、ユーザは、データが、その振る舞いを信用することができるプラットフォームと交換されているという、より高い信頼性を有することができる。

【００１８】

トラステッド装置は、暗号化プロセスを使用するが、必ずしもそれら暗号化プロセスに対して外部インタフェースを提供しない。また、最も望ましい実現は、トラステッド装置を不正操作できないように（tamperproof）して、それらを他のプラットフォーム機能に対してアクセス不可能にすることにより秘密を保護し、許可されていない変更から実質的に影響を受けない環境を提供する。タンパプルーフ化は不可能であるため、最も近いものは、耐タンパ性を有する（tamper-resistant）かまたはタンパ検出する（tanper-detecting）トラステッド装置である。従って、トラステッド装置は、好ましくは耐タンパ性を有する１つの物理的コンポーネントから構成される。

【００１９】

耐タンパ性に関連する技術は、セキュリティの分野における当業者には周知である。これら技術には、タンパリングに抵抗する方法（トラステッド装置の適当なカプセル化等）、タンパリングを検出する方法（トラステッド装置のケーシングにおける、仕様外電圧、Ｘ線または物理的完全性の損失の検出等）、およびタンパリングが検出される時のデータを除去する方法が含まれる。適当な技術の更なる説明は、http://www.cl.cam.ac.uk/～mgk25/tamper.htmlにおいて見ることができる。なお、タンパプルーフ化は、本発明の最も望ましい特徴であるが、本発明の通常の動作には入らず、従って本発明の範囲を逸脱するため、本明細書ではこれ以上詳細には説明しない。

【００２０】

トラステッド装置は、好ましくは、捏造する（forge）ことが困難でなければならないため物理的に１つである。最も好ましくは、偽造する（counterfeit）ことが困難でなければならないため、耐タンパ性（tamper-resistant）を有する。それは、一般に、ローカルにもまたある距離をおいても、識別を証明することが要求されるため、暗号化プロセスを使用することができるエンジンを有し、それが関連するプラットフォームのある完全性基準を測定する少なくとも１つの方法を含む。

【００２１】

図１において、トラステッドプラットフォーム１０が図に示されている。プラットフォーム１０は、プラットフォームの物理的「ユーザインタフェース」を提供する、キーボード１４、マウス１６および視覚表示装置（ＶＤＵ）１８の標準的な特徴を含む。また、トラステッドプラットフォームのこの実施の形態は、スマートカードリーダ１２を含む。スマートカードリーダは、すべてのトラステッドプラットフォームの本質的な要素ではないが

、後述する種々の好ましい実施の形態において採用される。スマートカードリーダ１２のそばに、トラステッドプラットフォームとのトラステッドなユーザ対話を可能にするスマートカード１９が示されている（トラステッドプラットフォームとのローカルなトラステッドなユーザ対話のためにスマートカードを使用することは、本発明の実施の形態には関連しているが本発明の機能には概して関連はしておらず、このコンテキストにおいて使用されてよいが、本明細書では詳細には説明しない。この態様は、その出願の内容が引用をもって本明細書内に包含されたものとする、「Smartcard User Interface for Trusted Computing Platform」と題され２０００年3月3日に出願された、本出願人による国際特許出願ＰＣＴ／ＧＢ００／００７５１号において更に述べられている）。プラットフォーム１０には、複数のモジュール１５があり、そのプラットフォームに対し適当な本質的にあらゆる種類のトラステッドプラットフォームの他の機能的要素である（かかる要素の機能的重要性は、本発明には関連しておらず、本明細書ではこれ以上説明しない）。

【００２２】

図２に示すように、トラステッドコンピューティングプラットフォーム１０のマザーボード２０は、（標準コンポーネントの中で）メインプロセッサ２１と、メインメモリ２２と、トラステッド装置２４と、データバス２６およびそれぞれ制御ライン２７およびライン２８と、プラットフォーム１０に対するＢＩＯＳプログラムを含むＢＩＯＳメモリ２９と、マザーボードのコンポーネントとスマートカードリーダ１２、キーボード１４、マウス１６およびＶＤＵ１８との間の対話を制御する入力／出力（ＩＯ）装置２３と、を含む。メインメモリ２２は、一般に、ランダムアクセスメモリ（ＲＡＭ）である。動作時、プラットフォーム１０は、オペレーティングシステム、例えばＷｉｎｄｏｗｓ　ＮＴ（商標）を、ハードディスク（図示せず）からＲＡＭにロードする。更に、動作時、プラットフォーム１０は、プラットフォーム１０によって実行されるであろうプロセスまたはアプリケーションをハードディスク（図示せず）からＲＡＭにロードする。

【００２３】

一般に、ＢＩＯＳプログラムが特別に予約されたメモリ領域に配置されるパーソナルコンピュータでは、最初のメガバイトの上位６４Ｋはシステムメモリとなり（Ｆ０００ｈ～ＦＦＦｈアドレス）、メインプロセッサは、業界標準に従ってこのメモリ位置を最初に見るように構成される。

【００２４】

本プラットフォームと従来からのプラットフォームとの重要な差異は、リセット後、メインプロセッサがトラステッド装置によって最初に制御され、その後トラステッド装置が、制御を特定プラットフォーム向けＢＩＯＳプログラムに渡し、それによってその特定プラットフォーム向けＢＩＯＳプログラムが、標準通りにすべての入力／出力装置を初期化する、ということである。ＢＩＯＳプログラムが実行された後、制御は標準通りにＢＩＯＳプログラムによって、一般にハードディスクドライブ（図示せず）からメインメモリ２２にロードされるＷｉｎｄｏｗｓ　ＮＴ（商標）等のオペレーティングシステムプログラムに渡される。

【００２５】

明らかに、通常の手続きからのこの変更には、業界標準の実現に対する変更が必要であり、それによりメインプロセッサ２１は、その最初の命令を受取るためにトラステッド装置２４にアドレスするよう命令される。この変更は、メインプロセッサ２１に異なるアドレスをハードコードすることによって簡単に行われ得る。代替的に、トラステッド装置２４に標準ＢＩＯＳプログラムアドレスが割当てられてよく、その場合、メインプロセッサ構成を変更する必要はない。

【００２６】

トラステッド装置２４内に、ＢＩＯＳブートブロックが含まれることが非常に望ましい。これにより、完全性基準の取得が破壊されることが防止され（欠陥のあるソフトウェアプロセスが存在する場合に発生する可能性がある）、ＢＩＯＳ（正確な場合であっても）がオペレーティングシステムに対し適当な環境を構築し損なう状況をもたらす、欠陥のある

ソフトウェアプロセスが防止される。

【００２７】

述べられるトラステッドコンピューティングプラットフォーム環境において、トラステッド装置２４は単一の別個のコンポーネントであるが、代替的に、トラステッド装置２４の機能がマザーボード上で複数の装置に分割されてよく、あるいはプラットフォームの現存する標準装置の１つまたは複数に統合されてもよい、ということが考えられる。例えば、機能およびそれらの通信が破壊される可能性が無い場合、トラステッド装置の機能のうちの１つまたは複数をメインプロセッサ自体に統合することが可能である。しかしながら、これには恐らく、トラステッドな機能によって単独に使用するために、プロセッサ上に別個のリードが必要となる。更にまたは代替的に、本実施の形態ではトラステッド装置は、マザーボード２０への統合に適応されるハードウェア装置であるが、ドングル（ｄｏｎｇｌｅ）等、要求される時にプラットフォームに取付けることができる「取外し可能（removable）」装置として実現されてもよい。トラステッド装置が統合されるか取外し可能であるかは、設計の選択の問題である。しかしながら、トラステッド装置が分離可能である場合、トラステッド装置とプラットフォームとの間に論理的結合を提供するメカニズムが存在しなければならない。

【００２８】

トラステッド装置２４は、図３に示すように、多数のブロックを備える。システムリセット後、トラステッド装置２４は、セキュアブートプロセスを実行することにより、プラットフォーム１０のオペレーティングシステム（システムクロックおよびモニタのディスプレイを含む）が適切にかつセキュアな方法で実行していることを確実にする。セキュアブートプロセス中、トラステッド装置２４は、コンピューティングプラットフォーム１０の完全性基準を取得する。また、トラステッド装置２４は、セキュアデータ転送、および例えば、暗号化／復号化および署名／検査を介するそれとスマートカードとの間の認証を実行することができる。また、トラステッド装置２４は、ユーザインタフェースのロッキング等、種々のセキュリティ制御ポリシーをセキュアに実施することができる。特定の好ましい構成では、コンピューティングプラットフォームに対するディスプレイドライバは、トラステッド装置２４内に配置され、その結果、ローカルユーザは、トラステッド装置２４によってディスプレイに提供されるデータの表示を信頼することができる。これは、その内容が引用をもって本明細書内に包含されたものとする、「System for Providing a Trustworthy User Interface」と題され２０００年５月２５日に出願された、本出願人による国際特許出願ＰＣＴ／ＧＢ００／０２００５号に更に述べられている。

【００２９】

特に、トラステッド装置は、トラステッド装置２４の全動作を制御し、トラステッド装置２４における他の機能およびマザーボード２０上の他の装置と対話するようプログラムされたコントローラ３０と、プラットフォーム１０から完全性基準を取得する測定機能３１と、指定されたデータに署名し、指定されたデータを暗号化し、復号化する暗号化機能３２と、スマートカードを認証する認証機能３３と、マザーボード２０のデータバス２６、制御ライン２７およびアドレスライン２８それぞれにトラステッド装置２４を接続する適当なポート（３６、３７および３８）を有するインタフェース回路３４と、を備える。トラステッド装置２４におけるブロックの各々は、トラステッド装置２４の適当な揮発性メモリ領域４および／または不揮発性メモリ領域３に（一般にコントローラ３０を介して）アクセスすることができる。更に、トラステッド装置２４は、周知の方法で、耐タンパ性であるように設計される。

【００３０】

パフォーマンス上の理由で、トラステッド装置２４は、特定用途向け集積回路（ＡＳＩＣ）として実現されてよい。しかしながら、柔軟性のために、トラステッド装置２４は、好ましくは、適当にプログラムされたマイクロコントローラである。ＡＳＩＣおよびマイクロコントローラは共に、マイクロエレクトロニクスの分野において周知であるため、本明細書ではこれ以上詳細には考慮しない。

【００３１】
トラステッド装置２４の不揮発性メモリ３に格納されるデータの１つの項目は、証明書（certificate）３５０である。証明書３５０は、トラステッド装置２４の少なくとも公開鍵３５１とトラステッドパーティ（ＴＰ）によって測定されるプラットフォーム完全性基準の認証された値３５２とを含む。証明書３５０は、トラステッド装置２４に格納される前に、ＴＰによってＴＰの秘密鍵を使用して署名される。後の通信セッションにおいて、プラットフォーム１０のユーザは、取得される完全性基準を認証された完全性基準３５２と比較することにより、プラットフォーム１０の完全性を検査することができる。一致する場合、ユーザは、プラットフォーム１０が破壊されていないことを確信することができる。ＴＰの一般に入手可能な公開鍵を知っていることにより、証明書３５０の簡単な検査が可能になる。また、不揮発性メモリ３５は、識別（ＩＤ）ラベル３５３を含む。ＩＤラベル３５３は、例えば通し番号等、いずれかのコンテキスト内で一意の従来からのＩＤラベルである。ＩＤラベル３５３は、概して、トラステッド装置２４に関連するデータのインデクス付けおよびラベル付けに対して使用されるが、トラステッドな状況下ではプラットフォーム１０の識別を証明するためにはそれだけでは不十分である。
【００３２】
トラステッド装置２４には、それが関連するコンピューティングプラットフォーム１０の完全性基準を確実に測定しまたは取得する少なくとも１つの方法が備えられている。本実施の形態では、完全性基準は、測定機能３１により、ＢＩＯＳメモリのＢＩＯＳ命令のダイジェストを生成することによって取得される。かかる取得される完全性基準は、上述したように検査されると、プラットフォーム１０がハードウェアまたはＢＩＯＳプログラムレベルで破壊されていないという高レベルの確信を、プラットフォーム１０の潜在的なユーザに与える。他の周知のプロセス、例えばウイルスチェッカは、一般に、オペレーティングシステムおよびアプリケーションプログラムコードが破壊されていないことをチェックするために適所にある。
【００３３】
測定機能３１は、ハッシュプログラム３５４とトラステッド装置２４の秘密鍵３５５とを格納する不揮発性メモリ３と、取得された完全性基準をダイジェスト３６１の形態で格納する揮発性メモリ４と、にアクセスすることができる。適当な実施の形態では、揮発性メモリ４は、プラットフォーム１０へのアクセスを可能にするために使用することができる１つまたは複数の認証されたスマートカード１９の公開鍵および関連するＩＤラベル３６０ａ～３６０ｎを格納するためにも使用されてよい。
【００３４】
１つの好ましい実施の形態では、完全性基準は、ダイジェストと同様にブール値を含み、それは、後に明らかとなる理由により、測定機能３１によって揮発性メモリ４に格納される。
【００３５】
ここで、完全性基準を取得する好ましいプロセスを、図４を参照して説明する。
【００３６】
ステップ５００において、スイッチオン時、測定機能３１は、データ、制御およびアドレスライン（２６、２７および２８）に対するメインプロセッサ２１のアクティビティを監視することにより、トラステッド装置２４がアクセスされる最初のメモリであるか判断する。従来からの動作では、メインプロセッサは最初に、ＢＩＯＳプログラムを実行するためにＢＩＯＳメモリに向けられていた。しかしながら、本発明によれば、メインプロセッサ２１は、メモリとして作用するトラステッド装置２４に向けられる。ステップ５０５において、トラステッド装置２４がアクセスされる最初のメモリである場合、ステップ５１０において、測定機能３１は、揮発性メモリ３に、トラステッド装置２４がアクセスされる最初のメモリだったことを示すブール値を書込む。そうでない場合、ステップ５１５において、測定機能は、トラステッド装置２４がアクセスされる最初のメモリでなかったことを示すブール値を書込む。

【００３７】
トラステッド装置２４が最初にアクセスされた装置でない場合、当然ながら、トラステッド装置２４がまったくアクセスされない可能性がある。これは、例えば、メインプロセッサ２１がＢＩＯＳプログラムを最初に実行するよう操作された場合である。これら環境において、プラットフォームは動作するが、完全性基準が入手可能でないため、その完全性を要求時に検査することができなくなる。更に、トラステッド装置２４が、ＢＩＯＳプログラムがアクセスされた後にアクセスされた場合、ブール値は、プラットフォームの完全性の欠如を明確に示すことになる。
【００３８】
ステップ５２０において、メインプロセッサ２１によってメモリとしてアクセスされる場合、メインプロセッサ２１は、ステップ５２５において、測定機能３１から格納されたネイティブのハッシュ命令３５４を読出す。ハッシュ命令３５４は、メインプロセッサ２１によって処理されるためにデータバス２６に渡される。ステップ５３０において、メインプロセッサ２１は、ハッシュ命令３５４を実行し、ステップ５３５において、それらを用いて、ＢＩＯＳメモリ２９の内容を読出しハッシュプログラムに従ってそれらの内容を処理することにより、ＢＩＯＳメモリ２９のダイジェストを計算する。ステップ５４０において、メインプロセッサ２１は、計算されたダイジェスト３６１を、トラステッド装置２４の適当な不揮発性メモリ位置４に書込む。そして、ステップ５４５において、測定機能３１は、ＢＩＯＳメモリ２９のＢＩＯＳプログラムを呼出し、従来からの方法で実行が継続する。
【００３９】
明らかに、要求される信頼の範囲によって、完全性基準が計算されてよい多数の異なる方法がある。ＢＩＯＳプログラムの完全性の測定は、プラットフォームの基礎にある処理環境の完全性に対し基本的なチェックを提供する。完全性基準は、ブートプロセスの妥当性に関する推論を可能にする形態でなければならず、完全性基準の値は、プラットフォームが正確なＢＩＯＳを使用してブートしたか検査するために使用することができる。任意に、ＢＩＯＳ内の個々の機能ブロックは、それら自体のダイジェスト値を有することができ、全体的なＢＩＯＳダイジェストはこれら個々のダイジェストのダイジェストである。これにより、ポリシーが、ＢＩＯＳ動作のいずれの部分が意図された目的に対して不可欠であるか、およびいずれが無関係であるかを述べることができる（この場合、個々のダイジェストは、ポリシー下の動作の妥当性を確立することができるような方法で格納されなければならない）。
【００４０】
他の完全性チェックは、プラットフォームに取付けられる、種々の他の装置、コンポーネントまたは装置が存在し、正しい動作順序にあることを確立することを含む。１つの例では、ＳＣＳＩコントローラに関連するＢＩＯＳプログラムは、周辺装置との通信が信頼できるものであることを確実にするように検査することができる。他の例では、プラットフォーム上の他の装置、例えばメモリ装置またはコプロセッサの完全性を、一貫した結果を確実にするようチャレンジ／応答（challenge/response）型の対話を行うことによって検査することができる。トラステッド装置２４が分離可能なコンポーネントである場合、対話のかかる形態の中には、トラステッド装置１４とプラットフォームとの間の適当な論理的結合を提供するために望ましいものがある。また、本実施の形態において、トラステッド装置２４は、プラットフォームの他の部分とのその通信の主な手段としてデータバスを利用するが、それほど都合よくはないが、ハードワイヤード経路または光路等の代替的な通信経路を提供することも可能である。更に、本実施の形態において、トラステッド装置２４は、メインプロセッサ２１に対し完全性基準を計算するよう命令するが、他の実施の形態では、トラステッド装置２４自体が１つまたは複数の完全性基準を測定するよう構成される。
【００４１】
好ましくは、ＢＩＯＳブートプロセスは、ブートプロセス自体の完全性を検査するメカニ

ズムを含む。かかるメカニズムは、例えば、Ｉｎｔｅｌのドラフト「Wired for Management baseline specification v2.0-BOOT Integrity Service」から既知であり、ロードする前のソフトウェアまたはファームウェアのダイジェストを計算することを含む。かかる計算されたダイジェストは、公開鍵がＢＩＯＳに知られているトラステッドエンティティによって提供される証明書に格納された値と比較される。そして、ソフトウェア／ファームウェアは、計算された値が証明書からの予測された値と一致し、証明書が、トラステッドエンティティの公開鍵を使用することによって妥当であると証明された場合にのみ、ロードされる。そうでない場合、適当な例外処理ルーチンが呼出される。

【００４２】

任意に、トラステッド装置２４は、計算されたＢＩＯＳダイジェストを受取った後、その計算されたダイジェストが適当な値に一致しない場合、証明書のＢＩＯＳダイジェストの適当な値を調べてＢＩＯＳに制御を渡さなくてもよい。更に、または代替的に、トラステッド装置２４は、自身がアクセスされた最初のメモリでなかった場合、ブール値を調べてＢＩＯＳに制御を返さなくてもよい。これらの場合のいずれかおいても、適当な例外処理ルーチンが呼びだされる。

【００４３】

図５は、ＴＰと、プラットフォームに組込まれるトラステッド装置２４と、トラステッドプラットフォームの完全性を検査したい（リモートプラットフォームの）ユーザと、による動作の流れを示す。ユーザがローカルユーザである場合、図５に示すものと実質的に同じステップが含まれる、ということは認められよう。いずれの場合も、ユーザは、一般に、検査を行うためにある形態のソフトウェアアプリケーションを基にする。リモートプラットフォームまたはトラステッドプラットフォームにおいてソフトウェアアプリケーションを実行することが可能である。しかしながら、リモートプラットフォームにおいてさえ、ソフトウェアアプリケーションが何らかの方法で破壊される可能性がある。従って、高レベルな完全性のために、ソフトウェアアプリケーションは、ユーザのスマートカード上に常駐し、ユーザは、検査の目的でスマートカードを適当なリーダに挿入することになることが予測される。

【００４４】

最初の例では、トラステッドプラットフォームを保証するＴＰは、プラットフォームのタイプを調べることによりそれを保証するか否かを判断する。これは、ポリシーの問題となる。すべてが適当である場合、ステップ６００において、ＴＰは、プラットフォームの完全性基準の値を測定する。そして、ＴＰは、ステップ６０５において、プラットフォームに対する証明書を生成する。証明書は、ＴＰにより、トラステッド装置の公開鍵および任意にそのＩＤラベルを、測定される完全性基準に添付し、ストリングにＴＰの秘密鍵を署名することによって生成される。

【００４５】

トラステッド装置２４は、後に、その秘密鍵を使用してユーザから受取る入力データを処理し、秘密鍵を知らずに入力／出力ペアを生成することが統計的に不可能であるように、出力データを作成することにより、その識別を証明することができる。これにより、秘密鍵を知っていることが、この場合の識別の基礎を形成する。明らかに、識別の基礎を形成するために対称暗号化を使用することは可能である。しかしながら、対称暗号化を使用する欠点は、ユーザが自身の秘密をトラステッド装置と共有する必要がある、ということである。更に、ユーザと秘密を共有する必要がある結果、対称暗号化は、原則としてユーザに対し識別を証明するためには十分である一方で、トラステッド装置またはユーザからもたらされる検査を完全に確信することができない第三者に対し、識別を証明するには不十分である。

【００４６】

ステップ６１０において、トラステッド装置２４は、証明書３５０をトラステッド装置２４の適当な不揮発性メモリ位置３に書込むことによって初期化される。これは、好ましくは、マザーボード２０に取付けられる前にトラステッド装置２４とのセキュア通信によっ

て行われる。証明書をトラステッド装置２４に書込む方法は、秘密鍵を書込むことによってスマートカードを初期化するために使用される方法に類似する。セキュア通信は、ＴＰにのみ知られた、製造中にトラステッド装置（またはスマートカード）に書込まれ、トラステッド装置２４にデータを書込むことを可能にするために使用される、「マスタ鍵」によってサポートされる。マスタ鍵を知らずにトラステッド装置２４にデータを書込むことは不可能である。

【００４７】

プラットフォームの動作中の後の時点で、例えばそれがスイッチオンされるかまたはリセットされる時、ステップ６１５において、トラステッド装置２４は、プラットフォームの完全性基準３６１を取得し格納する。

【００４８】

ユーザは、プラットフォームとの通信を希望する時、ステップ６２０において、乱数等のナンス（nonce）を生成し、ステップ６２５において、トラステッド装置２４にチャレンジする（challenge）（プラットフォームのオペレーティングシステム、または適当なソフトウェアアプリケーションは、チャレンジを認識し、それを、一般にＢＩＯＳタイプのコールを介して、適当な方法でトラステッド装置２４に渡すように構成される）。ナンスは、ユーザを、信頼できないプラットフォームによる古いが本物の署名の再現によってもたらされる欺き（deception）（「再現攻撃（replay attack)」と呼ばれる）から保護するために使用される。ナンスを提供し応答を検査するプロセスは、周知の「チャレンジ／応答(challenge/response)」プロセスの例である。

【００４９】

ステップ６３０において、トラステッド装置２４は、チャレンジを受取り適当な応答を生成する。これは、測定される完全性基準のダイジェストおよびナンス、任意にそのＩＤラベルであってよい。そして、ステップ６３５において、トラステッド装置２４は、その秘密鍵を使用してダイジェストに署名し、署名されたダイジェストに証明書３５０を添付してユーザに返す。

【００５０】

ステップ６４０において、ユーザは、チャレンジ応答を受取り、ＴＰの既知の秘密鍵を使用して証明書を検査する。そして、ユーザは、ステップ６５０において、証明書からトラステッド装置２４の公開鍵を抽出し、それを使用してチャレンジ応答から署名されたダイジェストを復号化する。そして、ステップ６６０にいて、ユーザは、チャレンジ応答内部のナンスを検査する。次に、ステップ６７０において、ユーザは、チャレンジ応答から抽出する計算された完全性基準を、証明書から抽出する適当なプラットフォーム完全性基準と比較する。ステップ６４５、６５５、６６５または６７５において、上述した検査ステップのいずれかが失敗すると、プロセス全体はステップ６８０で終了し、更なる通信は発生しない。

【００５１】

すべてが適当であると仮定すると、ステップ６８５および６９０において、ユーザおよびトラステッドプラットフォームは、他のプロトコルを使用して他のデータに対するセキュア通信をセットアップする。この場合、プラットフォームからのデータは、好ましくはトラステッド装置２４によって署名される。

【００５２】

この検査プロセスの更なる精密化が可能である。チャレンジャがチャレンジを通して、プラットフォーム完全性基準の値とそれが取得された方法との両方に気付くことが望ましい。これら両方の情報により、チャレンジャがプラットフォームの完全性に関する適当な判断を行うことができることが望ましい。また、チャレンジャは、異なる多くのオプションを利用可能である。すなわち、それは、完全性基準がトラステッド装置２４において妥当であると認識されることを認めてよく、あるいは、代替的に、完全性基準の値がチャレンジャが保持する値と等しい場合にのみ、プラットフォームが適切なレベルの完全性を有することを認めてもよい。（またはそこに保持してこれら２つの場合において異なるレベル

の信頼を有するようにしてもよい。）
【００５３】
上記説明は、トラステッドコンピューティングプラットフォームの汎用構造、目的および対話の振舞いを示す。ここで、図６を参照して、本発明の実施の形態で採用されるトラステッドコンピューティングプラットフォームの論理アーキテクチャを説明する。
【００５４】
図６に示す論理アーキテクチャは、トラステッドコンポーネント２４とコンピュータプラットフォームの残りの部分との間の物理的相違を一致させることにより、標準コンピュータプラットフォーム空間４００とトラステッドコンポーネント空間４０１との間の論理分割を示す。論理空間（ユーザ空間）４００は、コンピュータプラットフォーム１０のマザーボード２０上に物理的に存在するトラステッドコンポーネント２４以外のすべてを含み、論理空間（トラステッド空間）４０１は、トラステッドコンポーネント２４内に存在するすべてを含む。
【００５５】
ユーザ空間４００は、ユーザプラットフォームの標準の論理素子すべてを含むが、その多くはここには示されていない（本発明の動作には特に重要でないため）か、またはトラステッドコンピューティングプラットフォームのメインオペレーティングシステムの制御下にある、標準コンピューティング環境４２０に包含される。標準コンピューティング環境４２０を表す論理空間は、インターネット等の外部ネットワーク４０２（示されている例では、これは、トラステッドプラットフォームからのサービスのリクエスタと通信するためにとられる経路である）との通信を提供するために必要なものを含む、標準ドライバを含むようにとられる。また、ここでは、標準コンピューティング環境４２０の論理空間内において、コンピューティングプラットフォームの標準計算機能も包含されている。ユーザ空間４００内に示す他のコンポーネントは、コンパートメント４１０である。これらコンパートメントについては後述する。
【００５６】
トラステッド空間４０１は、トラステッドコンポーネント２４内のプロセッサおよびメモリによってサポートされる。トラステッド空間４１０は、トラステッド空間４０１の内部のコンポーネントと共に、コンパートメント４１０および標準コンピューティング環境４２０と対話するための通信コンポーネントを含む。標準コンピューティング環境４２０とトラステッド空間４０１との間にセキュア通信経路があることが望ましい（その内容が引用をもって本明細書内に包含されたものとする、２０００年２月１５日に出願された、本出願人による同時係属国際特許出願ＰＣＴ／ＧＢ００／００５０４号）。代替的な実施の形態は、ユーザ空間４００を含まないトラステッド空間４０１と外部ネットワーク４０２との間の直接接続を含んでよい。本構成では、トラステッド空間４０１とリモートユーザとの間でのみ交換される情報が、ユーザ空間４００を暗号化されて通過する。また、トラステッド空間４０１は、種々の動作から取得されるデータを収集し、このデータを、これら動作の完全性を検査したい者によって望まれる形態で提供するイベントロガー４７２と、トラステッド空間４０１外の通信とトラステッド空間４０１内の記録の提供（例えば、イベントロガー４７２による）とに要求される（後述するように）暗号化機能４７４と、ロギングされたイベントが期待されたものと同じであるか判断するために使用される予測アルゴリズム４７６と、トラステッドな方法で実行されるサービスの実行を調整するサービス管理機能４７８と、を含む（代替的な実施の形態では、サービス管理機能がユーザ空間４００に常駐することが可能であるが、これには、サービス管理機能４７８自体のより大量の暗号化通信および監視が必要となる。トラステッド空間４０１内にサービス管理機能４７８が常駐することにより、より単純な解決法が提供される）。また、トラステッド空間４０１内には、トラステッドコンパートメント４６０も常駐する。
【００５７】
ここで、コンパートメント４１０、４６０について更に説明する。コンパートメント４１０、４６０は、仮想コンピューティングエンジン４１１、４６１を含む環境であり、そこ

では、これら仮想コンピューティングエンジン上で実行するプロセスの動作または特権が制限される。コンパートメント内のコンピューティングエンジン上で実行されるプロセスは、外部の影響による干渉およびのぞき（prying）から高レベルに分離されて実行される。また、かかるプロセスは、不適当なデータに対するプロセスによる干渉またはのぞきを高レベルに制限して実行される。トラステッド空間４０１で作用することにより与えられる、外部からの影響からの同程度の保護が無い場合であっても、コンパートメントに対して制約が与えられるため、これら特性は信頼性の程度の結果である。コンパートメントの周知の形態は、Ｊａｖａサンドボックス（sandbox）であり、その場合、仮想コンピューティングエンジン４１１、４６１はＪａｖａ仮想マシン（Java Virtual Machine（ＪＶＭ））である。Ｊａｖａ仮想マシンおよびＪａｖａ内のセキュリティの処理は、Ｓｕｎ Ｍｉｃｒｏｓｙｓｔｅｍｓ Ｊａｖａウェブサイト（http://java.sun.com、特にhttp://java.sun.com/security）に述べられている。サンドボックスを実現するために、Ｊａｖａプラットフォームは、３つの主なコンポーネントに基づく。すなわち、クラスローダ、バイトコードベリファイヤおよびセキュリティマネージャである。各コンポーネントは、システムの完全性を維持するために重要な役割を果たす。同時に、これらコンポーネントは、正しいクラスのみがロードされることと、それらクラスが正しいフォーマットであることと、信頼されないクラスが危険な命令を実行しないことと、信頼されないクラスが保護されたシステムリソースにアクセスすることが許可されないことと、を保証する。各コンポーネントについては、例えば「Secure Computing with Java™:Now and the Future」と題された白書か、またはＪａｖａ Ｄｅｖｅｌｏｐｍｅｎｔ Ｋｉｔ １．１．Ｘにおいて（共に、例えばhttp://java.sun.comにおいて、Ｓｕｎ Ｍｉｃｒｏｓｙｓｔｅｍｓから取得可能である）更に述べられている。コンパートメント化環境におけるＪａｖａ仮想マシンの使用の例は、ＨＰ Ｐｒｅｓｉｄｉｕｍ ＶｉｒｔｕａｌＶａｕｌｔ（ＨＰ Ｐｒａｅｓｉｄｉｕｍ ＶｉｒｔｕａｌＶａｕｌｔの基本的な詳細は、http://www.hp.com/security/products/virtualvault/papers/brief_4.0/において述べられている）によって提供される。

【００５８】

このように、各コンパートメントは、プロセス要素を実行するための計算エンジンとしてＪａｖａ仮想マシン４１１、４６１を含む（後述するように、サービス管理プロセス４７８によってコンパートメントに割当てられる）。また、各コンパートメント４１１、４６１内には、コンパートメントが他のシステム要素と（特に、通信ツール４７０によりトラステッド空間４０１と）有効に通信するのを可能にする通信ツール４１２、４６２と、ＪＶＭ４１１、４６１上で実行されるプロセスの詳細をロギングし詳細をトラステッド空間４０１のイベントロガー４７２に返す監視プロセス４１３、４６３と、ＪＶＭ４１１、４６１によりコンパートメントとしての動作に必要なデータを保持し、コンパートメントに割付けられるプロセス要素によって使用するためのメモリ４１４、４６４と、が含まれる。

【００５９】

図６に示すコンパートメントには２つのタイプがある。コンパートメント４１０は、ユーザ空間４００に与えられており、コンパートメントの固有のセキュリティによってのみ保護されている。このため、コンパートメント４１０は、攻撃または破損に対して比較的セキュアである。しかしながら、特にクリティカルであるかまたは特に秘密であるプロセス要素については、ユーザ空間４００から完全に隔離されることが望ましい場合がある。これは、トラステッド空間４０１内に「トラステッド」コンパートメント４６０を配置することによって達成することができる。他の点では、コンパートメント４６０の機能は、コンパートメント４１０の機能とまったく同じである。トラステッド装置２４自体に物理的にトラステッドコンパートメント４６０を配置することに対する代替例は、トラステッドコンパートメント４６０を、トラステッド装置２４が保護されるのと同じ方法でタンパリングから物理的に保護される別個の物理要素内に配置することである。この場合、トラステッド装置２４とセキュアコンパートメント４６０を含む耐タンパエンティティとの間に

もセキュア通信経路が提供されることが有利である。

【００６０】

トラステッドモジュール２４内の「オペレーティングシステム」と「コンパートメント保護」とがハードウェアまたはファームウェアにハードコードされてよく、トラステッド空間４０１外部のデータまたはプロセスへのアクセスがハードウェアまたはファームウェアゲートキーパによって管理されてよいため、トラステッドコンパートメント４６０は、コンポーネント４１０より高レベルの信頼を提供する。これにより、トラステッドコンパートメント内のプロセスがその制御を破壊するかまたは望ましくないプロセスによって影響されることが非常に困難になる。

【００６１】

提供される保護されたコンパートメント４６０の数は、一方では利用可能な非常にトラステッドな処理の容量の大きさと、他方ではプラットフォームコストおよびプラットフォームパフォーマンスと、の間のバランスである。利用可能なコンパートメント４１０の数は、コストに大きく影響を与える可能性が低いが、プラットフォームパフォーマンスとプロセスの実行に関する証拠を収集する能力とのバランスである。

【００６２】

トラステッドコンピューティングプラットフォームによって実行されるプロセスの複雑性により、システムにおいていかなる数のコンパートメント４１０およびトラステッドコンパートメント４６０があってもよい。

【００６３】

本発明の態様は、リクエスタに対するサービスを実行するサービスプラットフォーム（好ましくは、トラステッドコンピューティングプラットフォーム）の使用に関する。この場合、リクエスタは、サービスが特定の方法で実行されるように要求する。特に、リクエスタは、全サービス内で所定のサービス要素が所定の程度の信頼性または安全性で実行されるよう要求する。概して、サービス要素は、３つのカテゴリに分割することができる。すなわち、絶対に信頼されるべきサービス要素と、信頼できることが要求されるサービス要素と、サービスが機能するために適当に動作する必要があるが、特別な程度の信頼は要求しないサービス要素と、である。本質的に信頼されなければならないサービスには、一般に、トラステッドコンピューティングプラットフォームにおけるソフトウェア環境の状態について報告するために使用されるもの（これは、原則的に、図１ないし図５に示すようなトラステッドコンピューティングプラットフォームの適当な使用により満足される）と、サービスの実行の証拠を提供するものと、である。信頼できることが要求され得るサービス要素は、関連する者の１つ（最もあり得るのはリクエスタ）が、その完全性のある程度の保証が要求されるほど十分に重要である、関連するサービス要素を処理したい場合である。例は、特定のコミュニティに対して重要であることが知られている機能であってよい（例えば、土木工学では、例えば耐力コンポーネントの設計において、あるタイプの計算が正確かつ信頼できる方法で実行されることを保証することが特に重要である場合がある）。他のサービス要素は、特定レベルの信頼は関係しない（しかしながら、トラステッドコンピューティングプラットフォームを使用してサービスを実行することにより、信頼性の幾分かの保証を簡単に提供することは可能である）。

【００６４】

図７は、リクエスタが、図６に示すようなサービスプラットフォーム上でサービスが実行されるよう手配する際に使用する、本発明の実施の形態によるプロセスの主な要素を示す。最初のステップは、リクエスタが、ステップ７０１においてトラステッドコンポーネント２４を認証することにより、コンピューティングプラットフォームがトラステッドコンピューティングプラットフォームであることを検査することである。このプロセスは、本質的に、図５に示すものである。このプロセスは、相互認証の１つであってよく、トラステッドコンピューティングプラットフォームは、ある種のサービスを実行する前に、リクエスタもまたトラステッドコンピューティングプラットフォームであることを要求してよい。

【００６５】
ステップ７０２において、リクエスタは、サービスを定義するために必要な材料を、トラステッドコンピューティングプラットフォームに送信する。この材料には２つの要素がある。すなわち、一方は、実行されるべきサービスとサービスの実行時のあらゆる状況とを定義する「契約」であり、他方は、サービスの実行に必要なカスタムデータである（リクエスタがまだ知らないデータが、一般に、トラステッドコンピューティングプラットフォームには入手可能である）。この材料は、トラステッドコンポーネント２４の暗号化機能を使用するか、またはコンピューティングプラットフォームのいずれかで利用可能な暗号化機能を使用して、好ましくはトラステッドコンポーネント２４によって提供される機能（鍵格納またはセッション鍵通信等）と協働して、復号化されるために暗号化形式で送信される。一般に、リクエスタは、セッション鍵によって材料をセキュアにし、その後トラステッドコンポーネント２４のみが知る鍵によってセッション鍵をセキュアにする。リクエスタは、保護されたセッションキーと保護された材料とを、トラステッドコンピューティングプラットフォームに送信する。トラステッドコンポーネント２４（特に、その暗号化機能４７４）は、セッション鍵を回復し、従来からのセキュリティ方法によって材料のソースおよび完全性を検査する。

【００６６】
契約は、サービスに対し、トラステッドコンピューティングプラットフォームが、いかにサービスを実行するか、すなわち何のサービス要素が実行されなければならないかと、それらが、サービスの結果をリクエスタに返す（あるいは、恐らく第三者に転送する）ことができるようにいかに結合されるべきであるかと、を解釈することができるように十分に指定しなければならない。更に、契約は、全体としてのサービス、または個々のサービス要素がいかに実行されるべきかに関し、リクエスタによって認められるあらゆる状況を指定する。本発明の実施の形態に特に関係することは、特定のサービス要素（プロセス）がトラステッドなまたはセキュアな方法で実行されなければならない、という要求である。例えば、リクエスタは、プロセスの第１のセットに対して絶対的な信頼が不可欠であることを要求し、プロセスの第２のセットに対して高い信頼を要求し、プロセスの第３のセットに対しては特定の要求をしなくてよい。これら要求は、プロセスの第１のセットをトラステッドコンパートメント４６０で実行し、プロセスの第２のセットをコンパートメント４１０で実行することによって満たされ、プロセスの第３のセットは、コンピューティングプラットフォームのオペレーティングシステムの制御の下で標準コンピューティング環境４２０において実行することができる。また、リクエスタは、契約が確実に実行され、同時に、要求される信頼の状態の下で実行された、という契約の実行の証拠も要求してよい。一般に、この要求は、絶対的な信頼を必要とし、トラステッドコンパートメントを提供する際か、または標準コンパートメントが安全な環境で実行されているという証拠を提供する際に、トラステッドコンポーネント２４の使用を要求する。後述するように、証拠は、プロセスの入力、実行および出力、特にコンパートメントまたはトラステッドコンパートメントで実行されるそれらをロギングすることによって、提供することができる。

【００６７】
より詳細には、契約は通常、以下を含む。すなわち、サービスの少なくとも一部を構成するプロセス（それは、提案しているサービス全体の他の部分は他の場所で実行されるということか、またはサービス全体が、本契約が補助的である「マスタ契約」によってカバーされる、ということであってよい）と、トラステッドコンピューティングプラットフォームにおいて実行されなければならないプロセス（一般に、要求は、すべてのプロセスがトラステッドコンピューティングプラットフォームで実行されるが、すべてがコンパートメントまたはトラステッドコンパートメント内で実行されるとは限らない、ということであってよい）と、サービスにおけるプロセスの順序と、プロセスの明白な記述（名前またはソース）および恐らくかかる記述の完全性を検査するデータ（プロセスを実行するプログラムの署名されたダイジェスト等）と、である。また、契約は、信頼レベルを指定してよく、あるいは、特に、特定のプロセスがコンパートメントまたはトラステッドコンパート

メント内で実行されるべきである（あるいは、部分的にコンパートメントまたはトラステッドコンパートメント内で実行されるべきである、この可能性については後述する）、ということを指定してよい。また、契約は、プロセスの完全性がいかにして決定されるか、すなわち入力および出力が記録されるかまたは恐らくサンプルされるか（そうである場合、いかにしてサンプルされるか）を判断してよく、プロセスの態様のスケジューリング、すなわち、プロセスの実行がいかにサンプルされるか、コンパートメント（またはトラステッドコンパートメント）自体内での実行がいかにして発生するか、およびプロセスがコンパートメント（またはトラステッドコンパートメント）の内外でいかにしてスワップされるかを判断してもよい。この情報はすべて、トラステッドコンピューティングプラットフォーム（より詳細には、サービス管理プロセス４７８）により解釈されるために機械読取り可能フォーマットで提供されなければならない。ＡＳＮ．１等のプログラミング方法が、１つの可能性である。

【００６８】

ステップ７０３において、トラステッドコンピューティングプラットフォームは、契約提案を受入れるかまたは拒絶する。提案は、トラステッドコンピューティングプラットフォームが契約要求に従ってサービスを実行するために必要な構成を有していない場合（例えば、利用可能なトラステッドコンパートメントを有しておらず、これらが契約提案によって明示的かまたは暗示的に要求される場合）か、トラステッドコンピューティングプラットフォームがリクエスタを信頼しない場合か、またはサービスがトラステッドコンピューティングプラットフォームにプログラムされる受入れの他のパラメータ外にある場合に、拒絶される可能性がある。適当なプロトコルが存在する場合、この段階で契約提案を交渉することも可能である（トラステッドコンピューティングプラットフォームが、いずれの契約条件を満たすことができないかを示し代替案を提案し、リクエスタが、代替案が受入れ可能か判断するか、またはそれ自身の新たな代替案を提案することによる。そして、プロセスは、契約が合意されるかまたは行詰まりに達するまで繰返される）。

【００６９】

契約提案が受入れられると、トラステッドコンピューティングプラットフォームが契約提案と共に必要な情報をすべて受取るとサービスを実行することが可能になる（代替的に、カスタムデータが、トラステッドコンピューティングプラットフォームが契約提案を受取るまで保持されてよく、受入れ後に提供される）。ステップ７０４において、トラステッド空間４０１のサービス管理プロセス４７８は、サービスをプロセスに分割し、それらプロセスを、適当にトラステッドコンパートメント４６０、コンパートメント４１０およびオペレーティングシステム４２０に割付ける。更に、サービス管理プロセス４７８は、割付けられたプロセスと共に、実行の証拠を提供するために必要なあらゆる監視プロセスを割当てる。これらは、契約自体において指定されてよく、あるいは、サービス管理プロセス４７８が、契約において要求されるあるレベルの証拠を提供するために必要な監視プロセスを決定してもよい。

【００７０】

ステップ７０５において、サービス（またはサービスを構成するプロセス）が実行され、必要な証拠がロギングされる。証拠のロギングについては、後に詳述する。ステップ７０６において、サービスの結果が、要求される場合に提供され（これは、リクエスタに対してであっても、または第三者に対してであってもよく、あるいは特定の場所に配置されてもよく、これは、サービスの特徴およびその目的によって決まる）、証拠ロギングプロセスの結果は、サービス管理プロセス４７８によって提供されるサービス全体のモデルに従ってイベントロガー４７２によってアセンブルされる場合に、トラステッドコンポーネント２４に返される（これについても、後に詳述する）。そして、ステップ７０７において、サービス証拠（および、サービスの結果がリクエスタにとって秘密である場合は、サービスの結果）が、暗号化されリクエスタに返される。

【００７１】

サービスの実行の主な要素を、図８に示す。リクエスタ８０１は、サービス管理プロセス

８０３によって受取られる場合に、契約８０２をトラステッドコンピューティングプラットフォームに提供する。契約は、結果がどこに提供されるべきか（この場合、リクエスタに返される）を指定するが、また要求される証拠のレベルも指定する。この場合、その証拠は、少なくともあるレベルの信頼が指定されるサービスにおけるプロセスに対して提供される。そして、契約は、トラステッドコンピューティングプラットフォームが実行することができるほど十分詳細にサービスを指定する。

【００７２】

契約が受信され受入れられると、サービス管理プロセス８０３は、動作を開始する。最初のジョブは、サービス管理プロセス８０３が、契約におけるプロセスをトラステッドコンピューティングプラットフォーム内の異なる計算エンジンに割付けるということである（考えられるところでは、プロセスは、トラステッドコンピューティングプラットフォーム外部に下請けに出すことも可能であるが、概して実行の証拠が要求される場合ではない）。この場合、プロセス１は、いかなる信頼要求も有しておらず、トラステッドコンピューティングプラットフォーム（ここでは、オペレーティングシステム８０６で示す）の標準コンピューティング環境に割付けることができ、プロセス２は、あり得る最高レベルの信頼を要求し、そのためトラステッドコンパートメント８０５に割付けられる（代替例では、プロセス２は、特にトラステッドコンパートメントの使用を要求する）。プロセス３は、ここではトラステッド実行が要求されることを示すが、「最高」よりは下である「要求される」レベルの信頼を有する。サービス管理プロセス８０３は、この信頼のレベルが、その特定のコンピューティングプラットフォームにおいていかにして満足されるべきであるか判断することができる。この例では、プロセス３は、コンパートメント（ここでは、コンパートメントＡ）８０７において実行されることが適当であると決定されるとする。プロセス４は、ここではコンパートメントで実行される「標準」プロセス（トラステッドコンピューティングプラットフォームに対し既知であるか、またはトラステッドコンピューティングプラットフォームに既知のソースから判断可能なタイプの周知のプロセス）であり、それはコンパートメントＢ（８０８）に割付けられる。

【００７３】

プロセスが異なるコンピューティングエンジンに割付けられる時、各プロセスに割付けられる証拠のロギングを決定することもまた必要である。これは、契約において、更なる解釈が必要でないほど十分適切に指定されてよい。例えば、プロセス１は、まったくロギングを要求せず（但し、それは、後述するように、ロギングされるトラステッドコンピューティングプラットフォームの通常のコンピューティング環境において実行されるプロセスに対して完全に可能である）、プロセス３は、プロセスに対するすべての入力および出力がロギングされる（但し、動作自体の詳細ではない）ことを要求する。プロセス２に対しては、契約は、ロギングが１００％であると規定する。これは、プロセスに対するすべての入力およびそれら入力がプロセスに提供されるシーケンスと、プロセスからのすべての出力および出力がプロセスによって提供されるシーケンスと、プロセスによって実行されるすべてのプログラム命令およびそれらが実行されるシーケンスと、をロギングすることを含む。プロセス４に対しては、信頼のレベルが「標準プロファイル」でセットされ、かかるプロファイルは、関連コミュニティを通してプロセス４等の標準プロセスに対し合意されることが可能であるか、または、リクエスタ８０１とトラステッドコンピューティングプラットフォームとの両方に対し他の方法で周知であることも可能である。

【００７４】

入力、出力および実行データ（またはこのデータのサブセット）の簡単なロギングに対し、代替方法が可能である。入力、出力および実行データはすべて、ロギングプロセスによってサンプルすることができる。サンプリング間隔は、契約において明示的に述べることができ、あるいは、契約において提供されるかまたは言及される機能によって、好ましくは秘密の初期値（恐らく契約の下で提供される）と共に決定されてよい。かかる機能が使用される場合、暗号化ハッシュまたは暗号（最も有利には、最高速度で出力データを生成するため、ストリーム暗号）等、エントロピが大きい機能を使用することが望ましい。初

期化後、機能は、データストリームを生成する。その後、そのデータストリームを検査することができ、その出力に特定のパターンまたが現れる場合、データはサンプルされダイジェストが計算される（値のシーケンスを有効にロギングする例示的なプロセスは、（１）シーケンス（機能）の現存する値にシーケンスに添付される新たな値を連結し、（２）データに対し、ＳＨＡ－１（国立標準技術研究所（National Institute of Standards and Technology（ＮＩＳＴ））によるAnnouncement of Weakness in the Secure Hash Standard、１９９４において述べられている）等のハッシュアルゴリズムを施し、（３）ハッシュアルゴリズムによって生成されるダイジェストを、シーケンスの新たな値としてセットする、というものである）。パターンの長さが短いほど、平均サンプルレートがより速くなる。このように、プロセスの証拠を、データのボーレートより低い平均レートで収集することができる。その出力に対するパターンマッチングと共に高出力エントロピの機能（ハッシュまたは暗号等）を使用することにより、データがサンプルされない時を予測することを困難にする不規則なサンプリングレートが提供される。秘密の初期値を使用することにより、不規則なサンプリングレートを１つの値として表す都合のよい方法が提供される。ロギングプログラム命令（プロセスの速度がサンプリングおよびロギングプロセスによって厳密に制限される可能性がある場合）に対し、プログラム命令のより高いボーレートを処理するために、予め決められた方法またはシーケンスで動作するハッシュエンジンの組合せを使用することが望ましい場合がある。

【００７５】

個々のプロセスに対するロギングの決定に加えて、サービス管理プロセス４７８は、別個の計算エンジンに対しカスタムデータを渡す必要のある場合もある。このデータの一部は、契約が提供される暗号化に加えて、特に暗号化されてよく、あるいは契約が、かかるデータがトラステッド空間４０１外に渡す前に暗号化されるように要求してもよい（例えば、それは、コンパートメントで実行するプロセスによって使用されることが意図されてよいが、コンパートメント自体は、サービス管理プロセス４７８によってコンパートメントに渡される時にデータを復号化するよう適応される）。プロセスには、消費する入力データがすべて渡される必要はなく、一部のデータは、標準データか、トラステッドコンピューティングプラットフォームにおいて利用可能であるかまたはそれがアクセス可能であることが既知であるデータとなる。プロセスによるデータの使用については後述する。

【００７６】

収集される証拠の特徴が確立されると、そのデータが収集された時にいかにしてアセンブルされるかを確立することが望ましい。これは、例えば、イベントロギングプロセス４７２に、証拠が収集されるすべてのプロセスの識別を、各かかるプロセスに対して収集されるべき証拠の特徴の指示と共に提供することによって、達成されてよい。

【００７７】

そして、プロセスおよび各プロセスに関連する証拠プロセスは、関連する計算エンジンに送られる（そのため、プロセス１は、実行されるために一般的な計算環境に送られ、プロセス２は、トラステッドコンパートメント８０５によって実行されるために送られ、プロセス３は、１つのコンパートメント８０７によって実行されるために送られ、プロセス４は、他のコンパートメント８０８によって実行されるために送られる）。サービスプロセスおよび関連する証拠プロセスの実行の方法は、プロセスが標準コンピューティング環境８０６で実行されるか、またはコンパートメント８０７、８０８すなわちトラステッドコンパートメントにおいて実行されるかによって異なる。それらは、サービスプロセス（そのため証拠プロセス）が、部分的に１つの環境で行われ部分的に他の環境で行われるようにスワップされる場合、更に変化する。

【００７８】

標準コンピューティング環境で実行されるプロセスについては、証拠の要求が無くてよい（図８におけるプロセス１の場合のように）。必ずしもそのようにはならないが、ロギングは、例えば標準コンピューティング環境で実行中でありトラステッド空間４０１におけるイベントロガー８０４と対話しているエージェントによって実行されてよい。かかる構

成は、「Data Event Logging in Computer Platform」と題され２０００年５月２５日に出願された、本出願人による同時係属国際特許出願ＰＣＴ／ＧＢ００／０２００４号において述べられている。概して、標準コンピューティング環境で実行されるプロセスについて、入力データの使用に対し特定の要求はなされないが、適当な場合は、入力データの使用に対する制約（コンパートメントの場合で後述する）が使用されてよい。

【００７９】

ここで、コンパートメントで実行中のプロセスに対し（ユーザ空間かトラステッド空間かに関らず）、プロセスの実行の段階（本明細書では、明確にするためにサービスプロセスとして説明する）について説明する。この例では、図６のコンパートメント４１０におけるサービスプロセスの実行について考える。コンパートメントは、仮想コンピューティングエンジン４１１としてのＪＶＭと、（特に）サービス管理プロセス４７８およびイベントロギングプロセス４７２とのデータの交換を可能にする通信プロセス４１２と、ＪＶＭ４１１において実行されるプロセスの詳細をロギングし詳細をイベントロガー４７２に返す監視プロセス４１３と、ＪＶＭ４１１が必要とするデータを保持するメモリ４１４と、を含む。サービスプロセスは、通信プロセス４１２とメモリ４１４に配置される必要なデータとを通してコンパートメントに提供される。そして、サービスプロセスはＪＶＭ４１１にロードされ、監視プロセス４１３は、サービスプロセスと共に提供される監視要求に従って起動されることが可能になる。

【００８０】

任意に、セキュア通信のために、コンパートメントにトラステッドコンポーネント２４が設けられてよい（それには、例えば、暗号識別および適当な暗号機能が提供されてよい）。これが行われる場合、プロセスとサービス管理プロセスによってそれと共に送られるデータとは、暗号化され、コンパートメントによって復号化されてよい。

【００８１】

コンパートメント環境の制約された特徴（特に、クラスローダ等の特徴）により、意図されたもの以外のコードのローディングが防止される。これにより、ＪＶＭ４１１に対するサービスプロセスのセキュアローディングが可能になる。好ましい変更では、この方法を、サービスプロセスによって使用される入力データにまで拡張することができる。好ましくは、サービスプロセスによって使用されるデータは、そのデータに対する使用許可情報を含む（使用許可が付加されていない場合、デフォルトが提供されることが可能であり、論理的に、これは無制限の使用を可能にする）。そして、サービスプロセス、特にコンパートメント等の制約された環境で実行されるサービスプロセスは、そのサービスプロセスが関連する許可の下で権限を与える場合にのみ入力データが使用されるように、適応されることが可能である。これは、サービスプロセスに対する入力データ（および論理的にサービスからの出力データも）にラベルでタイピングし、同様にサービスプロセスによってデータに対して実行される動作をタイピングすることにより、達成されてよい。タイプラベルは、機能（「保険証書を探索することに関連するプロセスにおいて、およびクレジットカードの適用に対する承認を得ることに関連するプロセスにおいてのみの使用」）か、または動作状態（「所定時刻の前に実行するプロセスにおいてのみの使用」）か、または両方を示してよい。プロセスによるデータの意図された使用がそのデータのタイプラベルと矛盾する場合、そのデータはプロセスによって使用されず、例外が発生してよい。このように、入力および出力データとサービスプロセス動作とのタイピングによってデータの誤用が防止される。

【００８２】

コンパートメント４１０がプロセスに入力データを返す時（これは、関連データがサービス管理プロセスによって提供された場合はメモリ４１４から、それが周知のデータである場合かまたはデータ位置への参照がサービスプロセスに渡された場合はコンピューティングプラットフォーム（または他の場所）のメインメモリからであってよく、あるいは特にトラステッドコンポーネント２４から要求されてもよい）、監視プロセス４１３は、要求される場合、入力データのセキュアログ（および、有利には、あらゆる関連するタイプタ

グ）を生成する。上述したように、これを行う有効な方法は、好ましくはコンパートメント内部の一時的なシーケンスにデータを添付することにより、このロギングされたデータをすべてリンクリスト（最終的にトラステッドコンポーネント２４によって署名される）に格納する。このデータは、プロセスを記述するログに付加することができる。採用される監視プロセスは、トラステッドコンポーネント２４によってトラステッドコンピューティングプラットフォームの完全性を監視するために使用されるプロセスと本質的に同様であってよい。

【００８３】

入力データは、第三者から取得されてもローカルに存在してもよいが、識別された「所有者」を有している。これらの場合、所有者には、所有者のデータがサービスプロセスの入力データとしていつ使用されるかが通知されることが望ましい場合がある。また、プロセスが、その使用に対するアクティブな同意を取得するために所有者と連絡をとることも望ましい場合がある。入力データに加えて、サービスプロセスは、それら自体が所有者を有することができる他のプロセスまたはルーチンを呼出すことも可能である。ここでまた、本発明の実施の形態では、コンパートメントは、これら所有者に対しそれらのプロセスまたはルーチンの使用を通知することができる。

【００８４】

同様に、監視プロセス４１３は、最も論理的には命令およびそれらが実行される順序の値を示すことにより、サービスプロセスの実行を監視することができる。上述したように、サンプリングの使用は、生成される証拠の量かまたはそれを取得する際の計算負荷を低減するために、監視プロセスのこの態様において特に有用であり得る。好ましい解決法は、不規則なサンプルレートを生成するために、ナンスで初期化されるハッシュアルゴリズムまたはストリーム暗号を使用することである。入力データに関し、結果としてのデータ（不規則なサンプリングの場合のナンス、ログおよびシーケンス値を含む）は、最終的にトラステッドコンポーネント２４によって署名されるために提供される。

【００８５】

サービスプロセスは、出力データを生成すると、（概して）メモリ４１４にそれを書込み、好ましくは出力データと共に、そのデータに許可される使用のタイプを含める。出力データのセキュアログは、本質的に入力データに対する場合と同じ方法で生成されてよい。

【００８６】

コンパートメントに対するサービスプロセスの実行の終了時、２つのタイプの出力がある。１つは、標準として、サービスプロセスの出力結果である。更に、異なるタイプの情報に対してしばしば使用される異なる監視プロセスと共に、データおよび命令の完全な記録からサンプリングプロセスの計算された値までのあらゆる形態の、１つまたは複数の入力データ、出力データおよびプロセス実行データを含む監視プロセス４１３によって生成されるログがある（サービス管理プロセスによって要求される場合）。このログは、サービス管理プロセスによりコンパートメントに渡される証拠要求に従ったものとなる。

【００８７】

トラステッドコンピューティングプラットフォームの標準コンピューティング環境におけるサービスプロセスの実行、またはコンパートメントまたはトラステッドコンパートメントにおける実行以外の、他の可能性もあり得る。これは、サービスプロセスが一方の実行環境と他方の実行環境との間でスワップされる、というものであり、例えば、標準コンピューティング環境からコンパートメントまたは保護されたコンパートメントへのプロセスのスワッピングか、またはコンパートメントと保護されたコンパートメントとの間のプロセスのスワッピングである。このスワッピングは、契約において指定することができ、あるいは、システムリソースを最適化する一方で契約の要求を満たすために、サービス管理プロセスによって決定することができる。他の可能性は、プロセススワップがプロセス自体によって起動されるというものであり、例えば、そのタイプにより、標準コンピューティング環境では実行することができず、コンパートメント等のより保護された環境でのみ実行することができるデータが受取られた場合である。このようなプロセスのスワッピン

グは、本質的に、リモートプロシージャコール等のよく理解された技術を使用する、物理的に別個のプロセッサ間のプロセスのスワッピングと、マルチタスクコンピュータ内のプロセスのスワッピングと、同様である。この場合、「第1のコンピュータ」は、物理的に「第２のコンピュータ」内にあってよい（またはその逆）が、これは、適用される必要のある技術にいかなる基本的な差異ももたらさない。このため、当業者は、プロセスのスワッピングをいかにして達成することができるかをよく理解するであろうから、これについてはこれ以上説明しない。

【００８８】

入力データと無関係なサービスプロセスに対し、サービスプロセスの実行は予測可能であってよい。プロセスがよく理解されている場合（本実施例のプロセス４）、サービスプロセス証拠の予測された値がプロセスの実行と無関係に生成されることが可能である（周知のプロセスに対し、予測値は、例えばトラステッドコンポーネント２４に格納されてよい）。コンパートメントがかかるサービスプロセスを実行し実行の証拠を収集している場合、この証拠を予測された証拠と比較することができる。満足のいく一致が無い場合、サービスプロセスの実行を終了し例外を発生させることができる。

【００８９】

別々のサービスプロセス９００の各々が実行された後、サービスプロセスの結果と証拠とは、それぞれサービス管理プロセス８０３とイベントロガー８０４とに返される（図９に最も明確に示すように）。例外があった場合も、サービス管理プロセス８０３に通知される。１つのサービスプロセスが他のサービスプロセスの結果に依存する場合、サービス管理プロセスかまたはスケジュールされるか割付けられる新たなサービスプロセスによりデータが提供されてよい。サービス管理プロセス８０３は、要求されるに従って別々のプロセスの結果をアセンブルし、すべてのサービスプロセスが実行された時サービス結果を含むことになる（例外がまったくなかったと仮定）。

【００９０】

イベントロギングプロセス８０４は、例えばユーザ空間のエージェントを使用することにより（上述したように）、コンパートメント外部で実行されるサービスプロセスに対しそれ自体で証拠を収集してもよい。更に、それは、それらサービスプロセスに対して与えられる要求に従って、コンパートメント内で実行するサービスプロセスからプロセス証拠を受取ることになる。このプロセス証拠は、期待されるものに対してチェックされてよい。イベントロギングプロセス８０４は、コンピューティングエンジンに対するサービスプロセスの割付けを企てるサービス管理プロセス８０３から受取る情報と、各プロセス（サービスプラン９０１における）に対する証拠要求と、を含む。イベントロギングプロセスは、サービスプロセスによって生成される証拠をチェックすることにより、それがサービスプラン９０１において示される証拠要求から期待されるものと一致するか（例えば、フォーマット）を判断することができる。入力データと無関係な方法で実行するプロセスに対し、更なるチェックが可能である。すなわち、かかるプロセスに対する予測される標準の証拠結果を、トラステッド空間４０１のメモリ９０２内に格納することができる。いずれの場合も、不一致が見付かった場合は適当な例外を生成することができる。

【００９１】

イベントロギングプロセス８０４は、まったく例外を見出さない場合、サービスプラン９０１に従ってプロセス証拠を結合することができる（証拠の結合の方法は、契約によって指定されてよい）。そして、そのサービス証拠を、サービス管理プロセス８０３に返すことができる。

【００９２】

この時、サービス管理プロセス８０３は、サービス結果とサービス証拠とを有する。また、それは、トラステッドコンポーネント２４の通常の動作に従って、サービス実行時にトラステッドコンピューティングプラットフォームに対する完全性データを取得することも可能である。この時、サービス管理プロセス８０３は、要求される時はいつでも、最も一般的にはリクエスタ８０１に返すようにサービス結果を送信することができる。望ましい

場合、これらには、トラステッドコンポーネント２４の暗号化プロセス９０３を使用してトラステッドコンポーネント２４の秘密鍵によって署名することができる。サービス証拠は、概して、トラステッドコンポーネント２４による署名後に、しばしばサービスの実行時にトラステッドコンピューティングプラットフォームに対する（署名された）完全性基準と共に、リクエスタ８０１に返される。

【００９３】

当業者は、本明細書で述べられたアーキテクチャおよび使用されるプロセスに対し、本発明の範囲を逸脱することなく多くの変更が行われ得る、ということを認めるであろう。図６ないし図９に示す実施の形態は、サービス管理プロセスおよびイベントロガー等の重要なコンポーネントがトラステッド空間４０１内に配置される場合、特にセキュアであるように適応される。これらプロセスの一部またはすべてがユーザ空間４００に移動される本発明の実施の形態を提供することができる。ユーザ空間４００においてかかるプロセスを実行する利点は、より単純なトラステッドコンポーネント２４と、そのためにより安価なアーキテクチャを使用することができる、ということである。トラステッドコンピューティングプラットフォームはトラステッドコンポーネント２４によって監視されるため、ユーザ空間４００は、非常に信頼できる。特に、イベントロギングプロセスは、サービスの実行のトラステッドな特徴に大きく影響を与えることなく、ユーザ空間４００に置換えることができる。サービス管理プロセスは、ユーザ空間への置換えがより困難である可能性があるが、トラステッド空間４０１とのセキュア通信に適応されたコンパートメント内でそれ自体を実行することはを恐らく可能である。

【００９４】

この手法の使用は、一者の、他者によるサービス実行に対する要求に制限されない。トラステッドコンピューティングプラットフォームは、相互にトラステッドな者とすることができ、それは、クライアントがそのデータをサービスに解放したくなかった場合、サービスプロバイダがそのプログラムをクライアントに解放したくなかった場合、またはサービスが異なるサプライヤからのクライアントによって要求される場合に使用することができる。相互にトラステッドな者は、必要な場合（または他の方法で指定される）を除いてデータおよびサービスの分離を保証するため、および契約が遂行された時にデータおよびサービスの完全な抹消を保証するために、関連する他のすべての者によって信頼される。

【００９５】

トラステッドコンピューティングプラットフォームは、データおよびサービスの抹消を保証した場合、サービスをサービスが満足に実行された証拠と共に識別するために十分な情報をそれ自体で（恐らくセキュアな記憶域に）保持することが望ましい場合がある。そうでない場合、トラステッドコンピューティングプラットフォームまたはその所有者は、あらゆる後続する問題の発生時に不利となる。

【００９６】

以下に本発明の実施態様の例を列挙する。

【００９７】

〔実施態様１〕　コンピューティングプラットフォームにおいてリクエスタに対するサービスを実行する方法であって、

該リクエスタが、実行される該サービスの仕様を該コンピューティングプラットフォームに対して供給し、該サービスの該仕様が、該サービスのプロセスの少なくとも一部に対して信頼の指定されたレベルを規定すること、

前記コンピューティングプラットフォームが、前記仕様に従って前記サービスを実行し、信頼のレベルが指定された前記プロセスの少なくとも一部の実行をロギングすることと、

前記コンピューティングプラットフォームが、信頼の前記指定されたレベルに従って実行される前記プロセスの前記実行のログを前記リクエスタに提供することと、

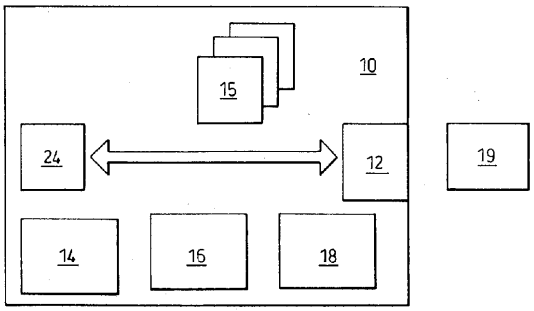
を含む方法。

〔実施態様２〕　前記仕様において信頼のレベルが指定されている前記プロセスの少なくとも一部に対し、実行のロギングを行わない実施態様１記載の方法。

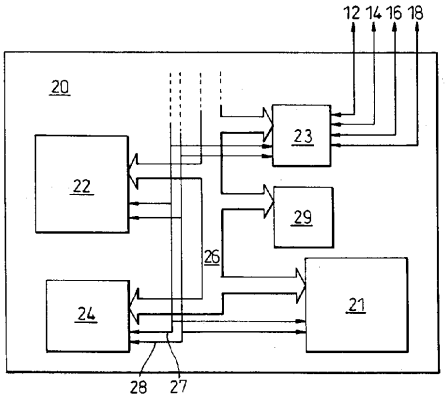
〔実施態様３〕　前記コンピューティングプラットフォームは、物理的かつ論理的に保護されたコンピューティング環境を含む実施態様１または２記載の方法。
〔実施態様４〕　前記物理的かつ論理的に保護されたコンピューティング環境は、前記コンピューティングプラットフォームの完全性を測定する監視プロセスを含む実施態様３記載の方法。
〔実施態様５〕　サービス管理プロセスは、プロセスの実行および実行のロギングを、前記コンピューティングプラットフォーム内かまたはそれに関連する別個のコンピューティング環境に割付ける実施態様１ないし４のいずれか1項記載の方法。
〔実施態様６〕　前記サービス管理プロセスは、前記保護されたコンピューティング環境内に配置される実施態様３に従属する実施態様５記載の方法。
〔実施態様７〕　前記別個のコンピューティング環境の１つまたは複数は、コンパートメントであり、動作または環境の制約により該コンパートメントの外部からの影響に対し保護されたコンピューティングエンジンを含むものである実施態様５記載の方法。
〔実施態様８〕　前記コンピューティングエンジンは、Ｊａｖａ仮想マシンである実施態様７記載の方法。
〔実施態様９〕　１つまたは複数のコンパートメントは、前記保護されたコンピューティング環境内に配置される実施態様３に従属する実施態様７または８記載の方法。
〔実施態様１０〕　前記コンピューティングエンジンを、許可されない場合は入力データに対して動作しないように制約する実施態様７ないし９のいずれか１項記載の方法。
〔実施態様１１〕　入力データにデータタイプを提供し、プロセスに動作タイプを提供し、動作タイプおよびデータタイプが一致しない場合は動作を行わない実施態様１０記載の方法。
〔実施態様１２〕　入力データは所有者を有してよく、前記プロセスは、該所有者に該入力データの使用を通知するように要求されてよい請求項１０または１１記載の方法。
〔請求項１３〕　入力データは所有者を有してよく、その場合、前記プロセスは、該所有者から該入力データの使用に対する同意を取得するように要求されてよい実施態様１０または１１記載の方法。
〔実施態様１４〕　プロセスを、１つの別個の環境と他の別個の環境との間でスワップしてよい実施態様５記載の方法。
〔実施態様１５〕　実行ロギングは、プロセスに対する入力データのロギングを含む実施態様１ないし１４のいずれか1項記載の方法。
〔実施態様１６〕　実行ロギングは、プロセスからの出力データのロギングを含む実施態様１ないし１５のいずれか1項記載の方法。
〔実施態様１７〕　実行ロギングは、プロセスの実行において実行されるプログラム命令のロギングを含む実施態様１ないし１６のいずれか1項記載の方法。
〔実施態様１８〕　ロギングされたデータを、サンプリングプロセスに従ってサンプルする実施態様１５ないし１７のいずれか1項記載の方法。
〔実施態様１９〕　前記サンプリングプロセスを、不規則なサンプリングを提供する機能に従って実行する実施態様１８記載の方法。
〔実施態様２０〕　ロギングされたデータのダイジェストを、前記実行ロギングデータの一部として取得する実施態様１５ないし１９のいずれか1項記載の方法。
〔実施態様２１〕　前記実行ロギングデータを、前記リクエスタに送信する前に暗号化する実施態様１ないし２０のいずれか1項記載の方法。
〔実施態様２２〕　前記仕様は、前記サービスの前記プロセスの少なくとも一部に対する実行ロギングパラメータを規定する実施態様１ないし２１のいずれか1項記載の方法。
〔実施態様２３〕　前記監視プロセスは、前記サービスが実行された時に前記コンピューティングプラットフォームの完全性基準を前記リクエスタに提供する実施態様４に従属する実施態様１記載の方法。
〔実施態様２４〕　コンピューティングプラットフォームであって、
該コンピューティングプラットフォームの適当なユーザに対し信頼できるデータを提供す

るよう適応された、物理的かつ論理的に保護されたコンピューティング環境と、
そこで実行されるプロセスが確実に実行されるように十分に制約された方法で動作するように構成された、１つまたは複数のコンパートメントと、
を具備し、
指定されたプロセスが、前記１つまたは複数のコンパートメントにおいてユーザに対して実行されてよく、該指定されたプロセスの結果が、前記保護されたコンピューティング環境から信頼できるデータで前記ユーザに返されてよいコンピューティングプラットフォーム。
〔実施態様２５〕　前記コンパートメントの１つまたは複数は、前記保護されたコンピューティング環境の外部に配置される実施態様２４記載のコンピューティングプラットフォーム。
〔実施態様２６〕　前記コンパートメントの１つまたは複数は、前記保護されたコンピューティング環境の内部に配置される実施態様２４または２５記載のコンピューティングプラットフォーム。
〔実施態様２７〕　各コンパートメントは、仮想コンピューティングエンジンを含む実施態様２４ないし２６のいずれか1項記載のコンピューティングプラットフォーム。
〔実施態様２８〕　前記仮想コンピューティングエンジンは、Ｊａｖａ仮想マシンである実施態様２７記載のコンピューティングプラットフォーム。
〔実施態様２９〕　前記保護されたコンピューティング環境は、前記コンピューティングプラットフォームの完全性を測定するように適応された監視プロセスを含む実施態様２４ないし２８のいずれか1項記載のコンピューティングプラットフォーム。
〔実施態様３０〕　前記コンピューティングプラットフォームは、前記サービス内のプロセスに割当てられる信頼のレベルを含むサービス記述を受取り、前記コンパートメントに対し前記プロセスの少なくとも一部を割付けるように適応されたサービス管理プロセスを含む実施態様２４ないし２９のいずれか1項記載のコンピューティングプラットフォーム。
〔実施態様３１〕　サービス管理プロセスは、前記保護されたコンピューティング環境内に配置される実施態様３０記載のコンピューティングプラットフォーム。
【図面の簡単な説明】
【図１】トラステッド装置を含み、本発明の実施の形態における使用に適したコンピューティングプラットフォームを示す図である。
【図２】スマートカードリーダを介してスマートカードと、およびモジュールのグループと通信するように配置されたトラステッド装置を含むマザーボードを示す図である。
【図３】トラステッド装置をより詳細に示す図である。
【図４】コンピューティング装置の完全性基準を取得することに関連するステップを示すフローチャートである。
【図５】トラステッドコンピューティングプラットフォームと、その完全性を検査するトラステッドプラットフォームを含むリモートプラットフォームと、の間の通信を確立することに関連するステップを示すフローチャートである。
【図６】図１に示すような、本発明の実施の形態における使用に適応したコンピューティングプラットフォームの論理アーキテクチャを概略的に示す図である。
【図７】リクエスタと本発明の実施の形態におけるコンピューティングプラットフォームとの間の相互作用を示すフローチャートである。
【図８】本発明の実施の形態によるコンピューティングプラットフォームによるサービスの実行を示す略図である。
【図９】サービス要素の実行における証拠の収集とリクエスタに対する証拠の提供とを示す図である。

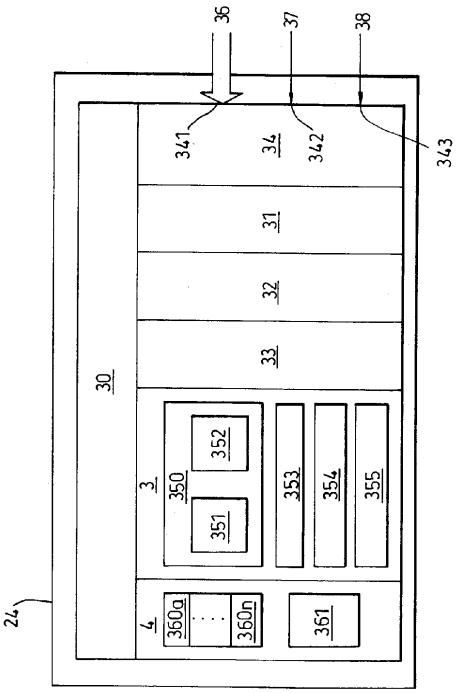
【図１】


10
15
24
12
19
14
16
18

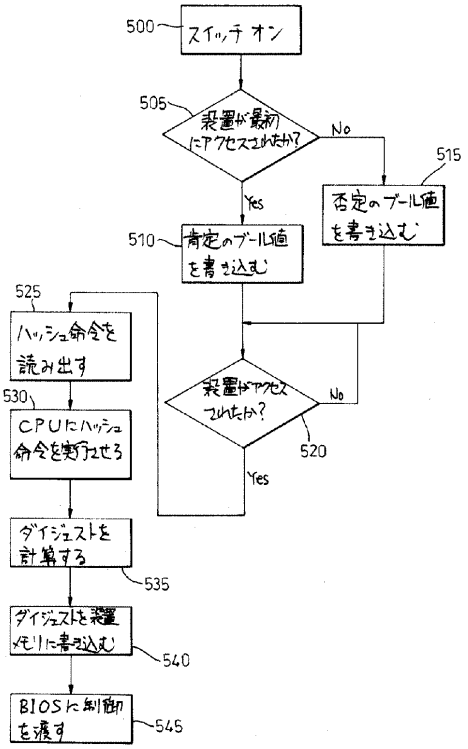
【図２】


12 14 16 18
20
23
22
29
26
24
21
28 27

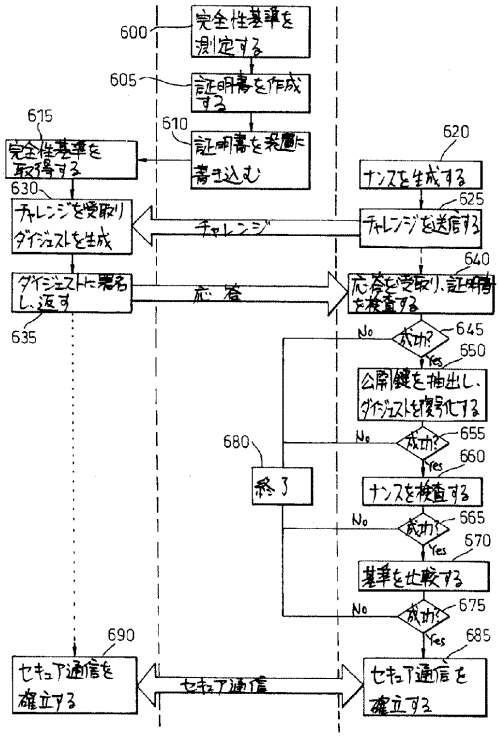
【図３】


36
37
38
341
342
343
34
31
32
33
30
3
350
352
351
353
354
355
4
360a
360n
361
24

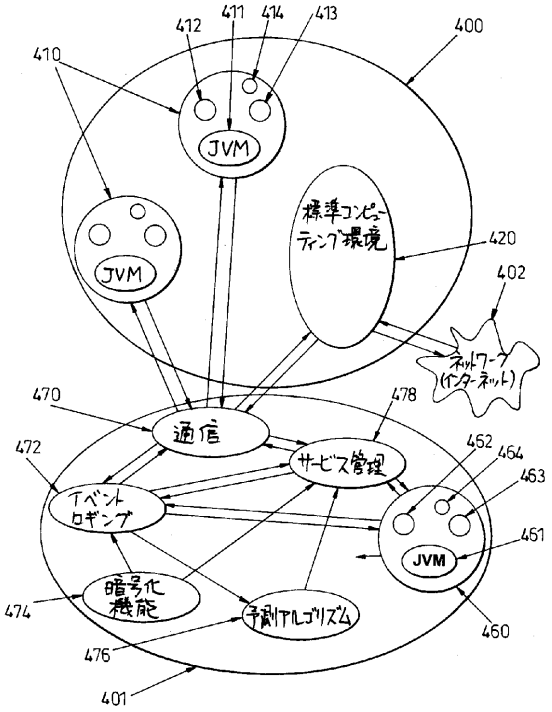
【図４】


500
スイッチ オン
505
装置が最初にアクセスされたか?
No
Yes
515
否定のブール値を書き込む
510
肯定のブール値を書き込む
525
ハッシュ命令を読み出す
530
CPUにハッシュ命令を実行させる
ダイジェストを計算する
535
ダイジェストを装置メモリに書き込む
540
BIOSに制御を渡す
545
装置がアクセスされたか?
No
520
Yes

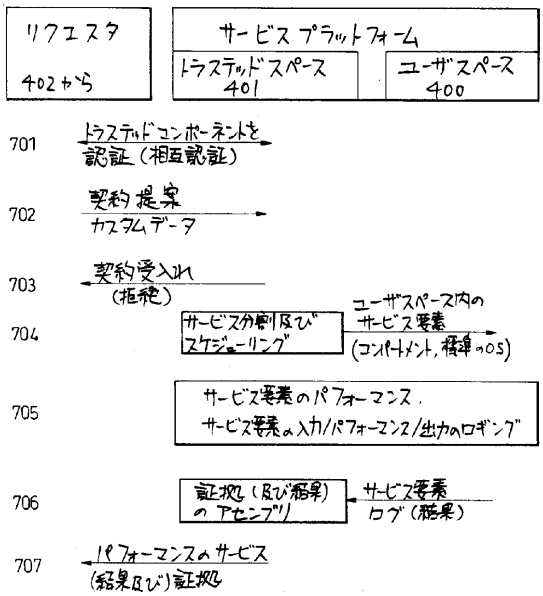
【図５】


600
完全性基準を測定する
605
証明書を作成する
610
証明書を装置に書き込む
615
完全性基準を取得する
620
ナンスを生成する
625
チャレンジを送信する
630
チャレンジを受取りダイジェストを生成
チャレンジ
640
応答を受取り、証明書を検査する
ダイジェストに署名し、返す
応答
635
No
成功?
645
Yes
550
公開鍵を抽出し、ダイジェストを復元する
No
成功?
655
Yes
660
ナンスを検査する
680
終了
No
成功?
565
Yes
570
基準を比較する
No
成功?
675
Yes
685
690
セキュア通信を確立する
セキュア通信
セキュア通信を確立する

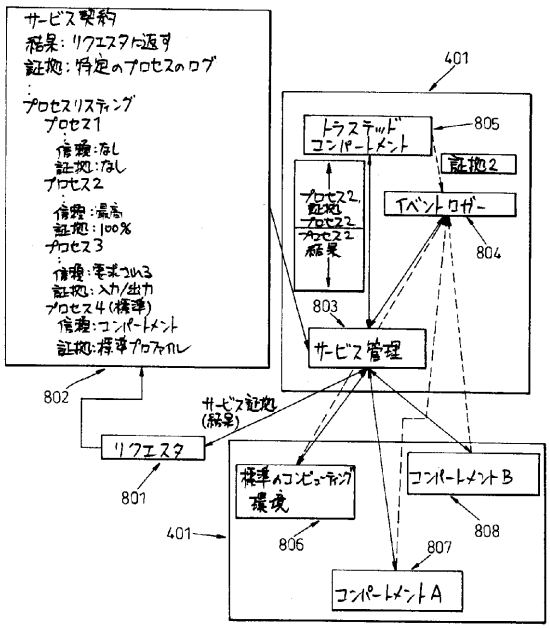
【図６】


412
411
414
413
400
410
JVM
標準コンピューティング環境
420
402
JVM
ネットワーク（インターネット）
470
通信
478
472
サービス管理
462
464
463
イベントロギング
JVM
451
暗号化機能
予測アルゴリズム
474
460
476
401

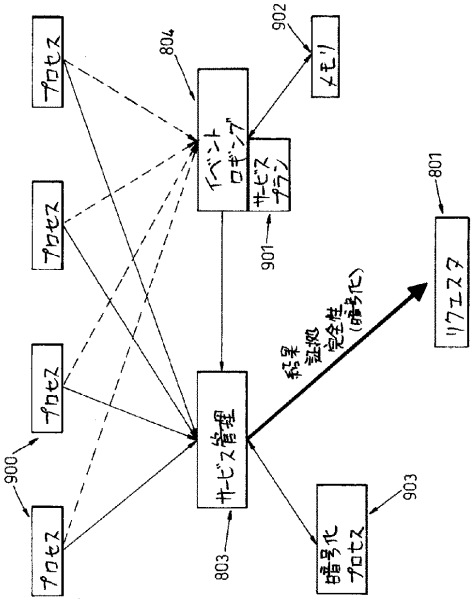
【図７】


リクエスタ
402から
サービスプラットフォーム
トラステッドスペース
401
ユーザスペース
400
701
トラステッドコンポーネントを認証（相互認証）
702
契約提案
カスタムデータ
703
契約受入れ
（拒絶）
704
サービス分割及びスケジューリング
ユーザスペース内のサービス要素
（コンパートメント，標準のOS）
705
サービス要素のパフォーマンス，
サービス要素の入力/パフォーマンス/出力のロギング
706
証拠（及び結果）のアセンブリ
サービス要素ログ（結果）
707
パフォーマンスのサービス
（結果及び）証拠

【図８】


サービス契約
結果：リクエスタに返す
証拠：特定のプロセスのログ
プロセスリスティング
プロセス1
信頼：なし
証拠：なし
プロセス2
信頼：最高
証拠：100%
プロセス3
信頼：要求される
証拠：入力/出力
プロセス4（標準）
信頼：コンパートメント
証拠：標準プロファイル
802
401
トラステッドコンパートメント
805
証拠2
プロセス2，証拠
プロセス2，プロセス2，結果
イベントロガー
804
803
サービス管理
サービス証拠（結果）
リクエスタ
801
標準のコンピューティング環境
コンパートメントB
808
401
806
807
コンパートメントA

【図９】


プロセス
804
902
メモリ
イベントロギング
サービスプラン
プロセス
801
901
プロセス
結果，証拠，完全性（暗号化）
リクエスタ
900
プロセス
サービス管理
プロセス
803
暗号化プロセス
903

フロントページの続き

(56)参考文献 米国特許第０６０５８４２６（ＵＳ，Ａ）
特開平０８－２４１２２７（ＪＰ，Ａ）
国際公開第００／０１９３２４（WO，Ａ１）
国際公開第９８／０３２０７３（WO，Ａ１）
国際公開第００／０４８０６３（WO，Ａ１）
特開平０５－３４２０６５（ＪＰ，Ａ）
特表２００２－５２６８３０（ＪＰ，Ａ）
特表２０００－５０８１０４（ＪＰ，Ａ）
特表２００２－５３６７５７（ＪＰ，Ａ）
国際公開第９９／６４９４６（WO，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
G06Q 10/00 - 50/00
G06F 11/34