(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7019697号
(P7019697)

(45)発行日　令和4年2月15日(2022.2.15)

(24)登録日　令和4年2月4日(2022.2.4)

(51)国際特許分類

Ｇ０６Ｆ　21/60　(2013.01)		Ｇ０６Ｆ	21/60	３４０
Ｇ０６Ｆ　21/62　(2013.01)		Ｇ０６Ｆ	21/60	３２０
Ｈ０４Ｌ　9/32　(2006.01)		Ｇ０６Ｆ	21/62	３０９
		Ｈ０４Ｌ	9/32	２００Ｚ

請求項の数　16　（全27頁）

(21)出願番号　特願2019-531494(P2019-531494)
(86)(22)出願日　平成29年8月30日(2017.8.30)
(65)公表番号　特表2019-530109(P2019-530109A)
(43)公表日　令和1年10月17日(2019.10.17)
(86)国際出願番号　PCT/AU2017/050928
(87)国際公開番号　WO2018/039722
(87)国際公開日　平成30年3月8日(2018.3.8)
審査請求日　令和2年7月17日(2020.7.17)
(31)優先権主張番号　2016903450
(32)優先日　平成28年8月30日(2016.8.30)
(33)優先権主張国・地域又は機関
オーストラリア(AU)

(73)特許権者　590003283
コモンウェルス サイエンティフィック アンド インダストリアル リサーチ オーガナイゼーション
オーストラリア２６０１オーストラリアン・キャピタル・テリトリー、アクトン、クルーニーズ・ロス・ストリート
(74)代理人　100108855
弁理士　蔵田 昌俊
(74)代理人　100103034
弁理士　野河 信久
(74)代理人　100153051
弁理士　河野 直樹
(74)代理人　100179062
弁理士　井上 正

最終頁に続く

(54)【発明の名称】　ブロックチェーン上の動的アクセス制御

(57)【特許請求の範囲】
【請求項１】
能力を生成することによる動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、
（ａ）前記オブジェクトに関する能力を生成するための要求を、送り側から受け取ることと、
（ｂ）前記送り側に対する前記オブジェクトに関する前記能力の存在を決定することと、
（ｃ）前記送り側に対する前記オブジェクトに関する前記能力が存在しないと決定される場合、前記送り側に対する前記オブジェクトに関する前記能力を生成することと、
（ｄ）前記送り側に対する前記オブジェクトに関する前記能力を前記ブロックチェーンシステム上に記憶させることと
を備え、前記能力は、前記オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、コンピュータ実装方法。
【請求項２】
成功メッセージを前記送り側に送ることをさらに備える、請求項１に記載のコンピュータ実装方法。
【請求項３】
オブジェクトに関する能力を許可する動的能力ベースのアクセス制御のためのコンピュータ実装方法であって、前記能力は、前記オブジェクトへの安全な参照であり、前記能力は

、ブロックチェーンシステム上に記憶され、前記コンピュータ実装方法は、
（ａ）ターゲットに対するオブジェクトに関する能力を許可するための要求を、送り側から受け取ることと、前記送り側は、能力動作を開始する第１のユーザプロセスであり、前記ターゲットは、前記能力動作にしたがって能力を取得する第２のユーザプロセスである、
（ｂ）前記ターゲットに対する前記オブジェクトのための前記送り側に対するアクセス権を決定することと、
（ｃ）前記オブジェクトに関する前記能力が前記ターゲットに対して許可されることを、前記アクセス権が可能にするか否かを決定することと、
（ｄ）前記ターゲットに対する前記オブジェクトに関する前記能力を前記ブロックチェーンシステム上に記憶させることと
を備え、前記能力は、前記オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、コンピュータ実装方法。
【請求項４】
前記能力は、請求項１の方法にしたがって生成される、請求項３の方法にしたがうブロックチェーンシステム上の能力を許可するためのコンピュータ実装方法。
【請求項５】
ステップ（ｃ）は、能力が前記オブジェクトに対して許可されているか否かを決定することを備える、請求項３に記載のコンピュータ実装方法。
【請求項６】
前記ターゲットに対する前記オブジェクトに関する前記能力は、前記送り側に対する前記オブジェクトに関する前記能力のサブセットである、請求項３、４または５に記載のコンピュータ実装方法。
【請求項７】
請求項１から６のうちのいずれか１項にしたがうブロックチェーン上に能力を生成するためのコンピュータ読取可能命令を有する非一時的コンピュータ読取可能媒体。
【請求項８】
能力を生成することによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、前記システムは、
（ａ）前記オブジェクトに関する能力を生成するための要求を、送り側から受け取ることと、
（ｂ）前記送り側に対する前記オブジェクトに関する前記能力の存在を決定することと、
（ｃ）前記送り側に対する前記オブジェクトに関する前記能力が存在しないと決定される場合、前記送り側に対する前記オブジェクトに関する前記能力を生成することと、
（ｄ）前記送り側に対する前記オブジェクトに関する前記能力を記憶させることと、ここにおいて、前記能力は、前記オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、
を行うように前記ブロックチェーンシステム上で実行する１つ以上のスマートコントラクトインスタンスを備える、システム。
【請求項９】
オブジェクトに関する能力を許可することによる動的能力ベースのアクセス制御のためのシステムであって、前記能力は、前記オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、前記システムは、
（ａ）ターゲットに対する前記オブジェクトに関する能力を許可するための要求を、送り側から受け取ることと、前記送り側は、能力動作を開始する第１のユーザプロセスであり、前記ターゲットは、前記能力動作にしたがって能力を取得する第２のユーザプロセスである、
（ｂ）前記ターゲットに対する前記オブジェクトのための前記送り側に対するアクセス権を決定することと、
（ｃ）前記オブジェクトに関する前記能力が前記ターゲットに対して許可されることを、

前記アクセス権が可能にするか否かを決定することと、
（ｄ）前記ターゲットに対する前記オブジェクトに関する前記能力を記憶させることと、
ここにおいて、前記能力は、前記オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、
を行うように前記ブロックチェーンシステム上で実行する１つ以上のスマートコントラクトインスタンスを備える、システム。
【請求項１０】
オブジェクトに関する能力を削除することによる動的能力ベースのアクセス制御のためのコンピュータ実装方法であって、前記能力は、前記オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、前記コンピュータ実装方法は、
（ａ）前記オブジェクトに関する能力を削除するための要求を、送り側から受け取ることと、前記送り側は、能力動作を開始する第１のユーザプロセスであり、ターゲットは、前記能力動作にしたがって能力を失う第２のユーザプロセスである、
（ｂ）前記送り側に対する前記オブジェクトに関する前記能力の存在を決定することと、
（ｃ）前記送り側に対する前記オブジェクトに関する前記能力が存在すると決定される場合、前記送り側に対する前記オブジェクトに関する前記能力を削除することと
を備える、コンピュータ実装方法。
【請求項１１】
能力は、請求項１の方法にしたがって生成される、請求項１０の方法にしたがう能力を削除するためのコンピュータ実装方法。
【請求項１２】
オブジェクトに関する能力を無効にすることによる動的能力ベースのアクセス制御のためのコンピュータ実装方法であって、前記能力は、前記オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、前記コンピュータ実装方法は、
（ａ）ターゲットに対するオブジェクトに関する能力を無効にするための要求を、送り側から受け取ることと、前記送り側は、能力動作を開始する第１のユーザプロセスであり、前記ターゲットは、前記能力動作にしたがって能力を失う第２のユーザプロセスである、
（ｂ）前記ターゲットに対する前記オブジェクトに関する前記送り側に対するアクセス権を決定することと、
（ｃ）前記オブジェクトに関する前記能力が前記ターゲットに対して無効にされることを、前記アクセス権が可能にするか否かを決定することと、
（ｄ）前記ターゲットに対する前記オブジェクトに関する前記能力を無効にすることと
を備える、コンピュータ実装方法。
【請求項１３】
能力は、請求項１の方法にしたがって生成される、請求項１２の方法にしたがう能力を無効にするためのコンピュータ実装方法。
【請求項１４】
前記能力が、マスタ能力であるか否かを決定することをさらに備える、請求項１２または１３に記載のコンピュータ実装方法。
【請求項１５】
オブジェクトに関する能力を削除ことによる動的能力ベースのアクセス制御のためのシステムであって、前記能力は、前記オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、前記システムは、
（ａ）オブジェクトに関する能力を削除するための要求を、送り側から受け取ることと、前記送り側は、能力動作を開始する第１のユーザプロセスであり、ターゲットは、前記能力動作にしたがって能力を失う第２のユーザプロセスである、
（ｂ）前記送り側に対する前記オブジェクトに関する前記能力の存在を決定することと、
（ｃ）前記送り側に対する前記オブジェクトに関する前記能力が存在すると決定される場合、前記送り側に対する前記オブジェクトに関する前記能力を削除することと、
を行うように前記ブロックチェーンシステム上で実行する１つ以上のスマートコントラク

トインスタンスを備える、システム。
【請求項１６】
オブジェクトに関する能力を無効にすることによる動的能力ベースのアクセス制御のためのシステムであって、前記能力は、前記オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、前記システムは、
（ａ）ターゲットに対するオブジェクトに関する能力を無効にするための要求を、送り側から受け取ることと、前記送り側は、能力動作を開始する第１のユーザプロセスであり、前記ターゲットは、前記能力動作にしたがって能力を失う第２のユーザプロセスである、
（ｂ）前記ターゲットに対する前記オブジェクトに関する前記送り側に対するアクセス権を決定することと、
（ｃ）前記オブジェクトに関する前記能力が前記ターゲットに対して無効にされることを、前記アクセス権が可能にするか否かを決定することと、
（ｄ）前記ターゲットに対する前記オブジェクトに関する前記能力を無効にすることと、
を行うように前記ブロックチェーンシステム上で実行する１つ以上のスマートコントラクトインスタンスを備える、システム。
【発明の詳細な説明】
【関連出願の相互参照】
【０００１】
［０００１］
本願は、２０１６年８月３０日に出願された豪州仮特許出願番号第２０１６９０３４５０号の優先権を主張し、その内容は、参照によってここに組み込まれている。
【技術分野】
【０００２】
［０００２］
本開示は、コンピュータセキュリティシステムにおいて使用されるブロックチェーン上の動的アクセス制御を実装するためのコンピュータ実装方法、ソフトウェア、および、システムに関する。
【背景】
【０００３】
［０００３］
アクセス制御は、企業における異なる部門および産業における異なる会社に渡る制御された情報共有のような、分散システムにおけるセキュリティの重要な特徴である。会社における異なる部門は、異なるポリシー、異なるアクセス制御の実装を有することが多く、レガシーシステムは、相互運用による問題を頻繁に生成するだろう。
【０００４】
［０００４］
多くのアクセス制御システムでは、個人へのアクセス権を追跡することは、個人の役割の変更および部門間移動により困難であることがある。さらに、アクセス権を追跡することは、アクセス制御とアクセス権が異なるシステムによって管理されている場合、より困難になる。アクセス権の監査証跡（audit trial）を使用することができるが、相関させることは困難であり、セキュリティポリシーモデルの組み合わせにより、監査証跡の複雑性が増加する。分散アクセス制御システムでは、アクセス権の制限が問題であり、アクセス権は、既知のユーザを超えて、拡散または漏れるかもしれない。
【０００５】
［０００５］
本明細書中に含まれる、文書、行為、素材、デバイス、物品またはこれらに類するものの任意の議論は、本願の各主張の優先日より前に存在したことにより、これらの事項のうちのいずれかまたはすべてが、先行技術ベースの一部を形成する、または、本開示に関連する分野で共通の一般知識であったという容認としてはとられない。
【０００６】

［０００６］
本明細書を通して、「備える（comprise）」との用語、あるいは、「備える（comprises）」または「備える（comprising）」のようなバリエーションは、述べた要素、整数またはステップ、あるいは、要素、整数またはステップのグループの包含を暗示すると理解されるが、他の何らかの要素、整数またはステップ、あるいは、要素、整数またはステップのグループの除外を暗示するものではないことも理解されるだろう。
【概要】
【０００７】
［０００７］
能力（capability）を生成することによる動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、能力は、ブロックチェーンシステム上に記憶され、
（ａ）オブジェクトに関する能力を生成するための要求を、送り側から受け取ることと、
（ｂ）送り側に対するオブジェクトに関する能力の存在を決定することと、
（ｃ）送り側に対するオブジェクトに関する能力が存在しないと決定される場合、送り側に対するオブジェクトに関する能力を生成することと、
（ｄ）送り側に対するオブジェクトに関する能力をブロックチェーンシステム上に記憶させることとを備え、能力は、オブジェクトに関するアクセス制御を動的に決定するために使用されることができる。
【０００８】
［０００８］
これは、いくつかの利点を有しており、監査証跡として使用されることができ、分散台帳（distributed ledger）と呼ばれる、計算量的に安全な不正操作ができない不変データ記憶装置であることから、ブロックチェーンシステムのその使用により、特に有利である。他の利益は、さまざまな認証委譲ポリシー（authorization-delegation policies）と動的アクセス制御を可能にすることを含んでいる。動的アクセス制御は、委譲の伝搬を選択的に可能にするまたは制限する能力と委譲を無効にする能力を含んでいる。さらに、能力は、ブロックチェーン上に記録されることができ、したがって、アクセス伝搬が知られる。これは、アクセス権の非常に細かい制御を許可し、アクセス権がどの程度細かいかに依存して、「混乱した使節の問題」（Confused Deputy problem）を解決することを手助けする。ある目的のために与えられたアクセス権を有するプログラムが、アクセス権のオリジナルの意図に反する他の何らかの目的のためにこれらのアクセス権を適用し、したがって、それが許可すべきでないことを許可するとき、混乱した使節の問題が起こる。この問題の古典的な例は、ログファイルと課金情報ファイルを含むディレクトリに書き込むことを許可されたプログラムに関係する。プログラムは、デバッグ情報を書き込むであろうファイルのパラメータを使用する。ユーザは、課金情報ファイルをプログラムに供給でき、したがって、課金情報を上書きする。これは、システム設計の間に意図されてはなかったかもしれないが、プログラムが必要なアクセス権を有する場合、おそらく、悪意あるまたは誤ったユーザ制御下で、この行為を実行するかもしれない。それが、この問題を克服することを助けることから、本発明は有利である。
【０００９】
［０００９］
方法は、成功メッセージを送り側に送ることをさらに備えていてもよい。
【００１０】
［００１０］
能力を許可することによる動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、能力は、ブロックチェーンシステム上に記憶され、
（ａ）ターゲットに対してオブジェクトに関する能力を許可するための要求を、送り側から受け取ることと、

（ｂ）ターゲットに対するオブジェクトのための送り側に対するアクセス権を決定することと、
（ｃ）オブジェクトに関する能力がターゲットに対して許可されることを、アクセス権が可能にするか否かを決定することと、
（ｄ）ターゲットに対するオブジェクトに関する能力をブロックチェーンシステム上に記憶させることとを備え、能力は、オブジェクトに関するアクセス制御を動的に決定するために使用されることができる。
【００１１】
［００１１］
ステップ（ｃ）は、能力がオブジェクトに対して許可されているか否かを決定することを備えていてもよい。
【００１２】
［００１２］
ターゲットに対するオブジェクトに関する能力は、送り側に対するオブジェクトに関する能力のサブセットであってもよい。
【００１３】
［００１３］
非一時的コンピュータ読取可能媒体は、上記の方法にしたがうブロックチェーン上に能力を生成するためのコンピュータ読取可能命令を有する。
【００１４】
［００１４］
能力を生成することによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、能力は、ブロックチェーンシステム上に記憶され、システムは、
ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスであって、
（ａ）オブジェクトに関する能力を生成するための要求を、送り側から受け取り、
（ｂ）送り側に対するオブジェクトに関する能力の存在を決定し、
（ｃ）送り側に対するオブジェクトに関する能力が存在しないと決定される場合、送り側に対するオブジェクトに関する能力を生成し、
（ｄ）送り側に対するオブジェクトに関する能力を記憶させ、ここにおいて、能力は、オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスを備える。
【００１５】
［００１５］
能力を許可することによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、能力は、ブロックチェーン上に記憶され、システムは、
ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスであって、
（ａ）ターゲットに対してオブジェクトに関する能力を許可するための要求を、送り側から受け取り、
（ｂ）ターゲットに対するオブジェクトのための送り側に対するアクセス権を決定し、
（ｃ）オブジェクトに関する能力がターゲットに対して許可されることを、アクセス権が可能にするか否かを決定し、
（ｄ）ターゲットに対するオブジェクトに関する能力を記憶させ、ここにおいて、能力は、オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスを備える。
【００１６】
［００１６］
能力を削除することによる動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、能力は、ブロックチェーンシステム上に記憶され、

（ａ）オブジェクトに関する能力を削除するための要求を、送り側から受け取ることと、
（ｂ）送り側に対するオブジェクトに関する能力の存在を決定することと、
（ｃ）送り側に対するオブジェクトに関する能力が存在すると決定される場合、送り側に対するオブジェクトに関する能力を削除することとを備える。
【００１７】
［００１７］
能力を無効にすることによる動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、能力は、ブロックチェーンシステム上に記憶され、
（ａ）ターゲットに対するオブジェクトに関する能力を無効にするための要求を、送り側から受け取ることと、
（ｂ）ターゲットに対するオブジェクトに関する送り側に対するアクセス権を決定することと、
（ｃ）オブジェクトに関する能力がターゲットに対して無効にされることを、アクセス権が可能にするか否かを決定することと、
（ｄ）ターゲットに対するオブジェクトに関する能力を無効にすることとを備える。
【００１８】
［００１８］
コンピュータ実装方法は、能力が、マスタ能力であるか否かを決定することをさらに備えていてもよい。
【００１９】
［００１９］
能力を削除ことによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、能力は、ブロックチェーンシステム上に記憶され、システムは、ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスであって、
（ａ）オブジェクトに関する能力を削除するための要求を、送り側から受け取り、
（ｂ）送り側に対するオブジェクトに関する能力の存在を決定し、
（ｃ）送り側に対するオブジェクトに関する能力が存在すると決定される場合、送り側に対するオブジェクトに関する能力を削除する、ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスを備える。
【００２０】
［００２０］
能力を無効にすることによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、能力は、ブロックチェーンシステム上に記憶され、システムは、
ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスであって、
（ａ）ターゲットに対するオブジェクトに関する能力を無効にするための要求を、送り側から受け取り、
（ｂ）ターゲットに対するオブジェクトに関する送り側に対するアクセス権を決定し、
（ｃ）オブジェクトに関する能力がターゲットに対して無効にされることを、アクセス権が可能にするか否かを決定し、
（ｄ）ターゲットに対するオブジェクトに関する能力を無効にする、ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスを備える。
【図面の簡単な説明】
【００２１】
［００２１］
以下の図面を参照して、本開示の例を説明する。
【図１】［００２２］　図１は、例示的なアプリケーションシナリオを図示している。
【図２】［００２３］　図２は、能力を生成するためのコンピュータ実装方法を図示している。

【図３】［００２４］　図３は、能力を許可するためのコンピュータ実装方法を図示している。
【図４】［００２５］　図４は、能力を削除するためのコンピュータ実装方法を図示している。
【図５】［００２６］　図５は、能力を無効にするためのコンピュータ実装方法を図示している。
【図６】［００２７］　図６は、能力を呼び出すためのコンピュータ実装方法を図示している。
【図７】［００２８］　図７は、例示的な送り側を図示している。
【実施形態の説明】
【００２２】
［００２９］
本発明は、概して、ブロックチェーンシステム上で動的アクセス制御を実装するための方法ソフトウェアおよびシステムに関する。本開示では、スマートコントラクトのインテグリティは、アクセス制御ロジックと管理の安全な処理を確実にする。スマートコントラクトは、多くのブロックチェーンシステムの計算インフラストラクチャの部分を形成し、これは、能力動作を実行するとともに能力動作の履歴を記憶するために使用されることができる。
【００２３】
［００３０］
本開示において、ブロックチェーンシステム上で能力が実装され、能力に関する動作がトランザクションとして記憶される。したがって、ブロックチェーンは、能力に関するすべての動作についての事実上不変かつ偽造不可能な履歴となる。ブロックチェーンのコピーは、能力の現在の状態を表す。１つの例では、能力の記録がスマートコントラクト記憶スペース中に記憶される。この記憶スペースは、変数を含む任意のデータ構造であってもよい。能力に関する動作は、能力記録の最新の状態を追跡するための変数を修正する。動作をトリガするためのトランザクションは、能力記録への変更と修正に対する監査証跡を提供するために、ブロックチェーン上に記録される。
【００２４】
ブロックチェーン
［００３１］
ブロックチェーンは、ピアツーピアネットワーク内のすべてのノードによって管理される公共共有台帳となるように設計された。ブロックチェーンシステムは、伝統的な銀行および支払いシステムのような、任意の中央信用機関に依存しない。その代わりに、信用は、ネットワーク内のノード間の相互作用からの創発特性（emergent property）として達成される。ブロックチェーンの完全なコピーは、ブロックチェーン内で今までに実行されたすべてのトランザクションを含み、したがって、ブロックチェーン上でトランザクションを行う当事者は、高レベルの確実性で、他の当事者が要求するように他の当事者がトランザクションを行うことができることを検証できる。さらなる重大性は、ブロックチェーンは偽造やハッキングに対して立証可能にロバストであることである。この１つの重要な重大性は、金銭の取引のための銀行あるいはアクセス権のトランザクションのための信用セキュリティプログラムのような信用できる第三者を、利用する必要がないことである。
【００２５】
［００３２］
概念としてのブロックチェーンは、最初、ビットコイン（Bitcoin）のインフラストラクチャにおいて定義された。それ以来、その概念は、暗号通貨またはトークンを必要とすることなく、任意のトランザクションを検証し、記憶するためにブロックチェーンを使用する分散型台帳に一般化されている。多くの暗号通貨は、現在、これらのプラットフォームのベースとして、ブロックチェーンを使用している。有名なブロックチェーンシステムは、ビットコインとイーサリアム（Ethereum）を含む。

【００２６】
トランザクション
［００３３］
上記のように、ブロックチェーンは、本質的に、合意したプロトコルに基づいて、システムに参加するすべてのノードによって共有されるトランザクションのリストである。各トランザクションは、ある当事者のアドレスから別の当事者のアドレスへの特定の量のデジタル資産の移動を表す。したがって、アドレス機能は銀行口座に類似している。例えば、アリスは、ブロックチェーン上でトランザクションを生成することにより、ボブに５ビットコインを支払うことができ、彼女のアドレスのうちの１つから５ビットコインを移動し、ボブのアドレスを出力として特定する。
【００２７】
［００３４］
ブロックチェーンは、生じたすべてのトランザクションを記録し、分散型台帳と呼ばれる耐タンパ性で不変のデータ記憶装置を提供する。参加するエンティティの全体ネットワークは、分散型台帳に含まれる取引について合意に達する。トランザクションは、本質的に、取引することを望むどんなユーザも表すことができる識別可能なデータパッケージである。多くのブロックチェーンシステムにおいて、データパッケージは、ビットコインまたはイーサのような金銭的価値の表示であるが、これらは、スマートコントラクトにおけるパラメータと関数呼び出しの結果も含むことができる。ブロックチェーンシステムにおけるトランザクションのインテグリティは、暗号技術によって確実にされる。通貨のブロックチェーンの完全なコピーを含むブロックチェーンシステムに接続および参加しているコンピュータは、「フルノード」として知られる。
【００２８】
ブロック
［００３５］
ブロックチェーンシステムにおいて、トランザクションはブロックに集約される。各ブロックは、ハッシュと呼ばれる前のブロックの数学的関数計算を含んでいる。この数学的関数計算は、所定の特定のトランザクションを計算することは容易であるが、所定の特定のハッシュを逆にすることは困難である。これは、トランザクション中のコンテンツが多少なりとも修正されているか否かを決定する手段を表す。トランザクション中にハッシュを含めることにより、これは、ブロックになされた何らかの変更がそのブロックのハッシュを変更するであろうチェーンを生成し、それは、次のブロック中で再計算され、記憶されなければならない。これは、次のブロックのハッシュを変更し、チェーンの終わりまで、再計算等されなければならない。
【００２９】
［００３６］
したがって、各ブロックはまた、前のブロックへの参照を含むことにより、前のブロック（「親」ブロック）にリンクされる。これは、現在のブロックからジェネシスブロックとして知られている本当に最初のブロックへのブロックのチェーンを生成する効果を有している。前のブロックのハッシュは別の方法では知られないことから、各ブロックは、時系列的に前のブロックの後に来ることが保証される。いったん、各ブロックがしばらくの間（典型的に、ビットコインにおいて、これは、約６０分または平均で６つのブロックである）チェーン中にあると、各ブロックは、修正するのに計算面で非実用的でもある。なぜなら、その後のすべてのブロックも、再発生させなければならないからである。これらの特性は、ブロックチェーン上の能力に関連するトランザクションを偽造することを困難にするものである。
【００３０】
スマートコントラクト
［００３７］
ブロックチェーンシステム中のスマートコントラクトは、契約の法的概念を複製すること

を意図している。すなわち、契約は、契約の当事者に義務を課す相互の合意であり、スマートコントラクトは、トランザクション上の義務または条件を自動的に課す方法である。
【００３１】
［００３８］
ビットコインとイーサリアム（および、たいていの他のブロックチェーンシステム）は、トランザクションを検証する目的のために、スクリプトを利用する。スマートコントラクトをスクリプトとして実装することが可能であり、通常のトランザクションが検証されるのと同じ方法で動作するだろう。これは、ビットコインが採用したやり方であるが、検証されるトランザクションについて、スクリプトが順番に実行しなければならないことから、これは非常に困難である。イーサリアムでは、スマートコントラクトは、トランザクションを検証することとは区別される。結果として、スマートコントラクトは、契約を開始したトランザクションとは独立して存在することができる。他の違いは、スマートコントラクトは、それが実行される前に展開されなければならず、いったん展開されると、明示的に終了しない限り、スマートコントラクトは実行するだろう。ビットコインとイーサリアムの両方で、スクリプトは、契約が自動的に実行されるために必要とされるすべての関数呼び出し（ＯＰコード）を含んでいる。両方のブロックチェーンシステム上のすべてのノードは、スクリプトを実行する。
【００３２】
［００３９］
ブロックチェーンシステム中のスクリプトの使用には多くの制限がある。ビットコインに関する１つのこのような制限は、スクリプト言語がスマートコントラクトを書き込むためにチューリング完全（Turing-complete）ではないとみなされることである。結論は、ビットコインスクリプトは、任意のループを生成するために使用されないということである。この制限は、攻撃者が、ビットコインシステム上で、サービス妨害（ＤＯＳ）タイプの攻撃として機能する無限にループするスクリプトを実行することを回避することができる。対照的に、イーサリアムは、スマートコントラクトを書き込むための組み込み型チューリング完全スクリプト言語を有している。イーサリアムがＤＯＳ攻撃の可能性を克服する方法は、スクリプトによって必要とされるＣＰＵサイクルに比例した料金（これはイーサリアムでガスとして知られている）を支払うことを、通貨を取引する当事者が必要とされることである。したがって、攻撃者はスクリプトを実行するために支払う膨大な通貨量を必要とすることから、スクリプトは、無限にループできない。
【００３３】
［００４０］
文献では、用語「スマートコントラクト」は、スマートコントラクトを実行するために使用されるコードと、スマートコントラクトを実際に実行することまたは実行されたスマートコントラクトとの両方を指すために交換可能に使用される。明確にするために、本開示では、用語「プロセスインスタンス」は、実行、および、スマートコントラクトによって提供されるサービスを指す。用語「スクリプト」は、プロセスインスタンスとして実行されることができるスマートコントラクトコードを指す。
【００３４】
［００４１］
本開示は、実際のブロックチェーン自体（すなわち、シーケンシャルに追加されるブロックを有する公共共有台帳）を指すために用語「ブロックチェーン」を使用する。本開示はまた、ブロックチェーンシステムとブロックチェーンネットワークに関連する用語ブロックチェーンも使用する。用語「ブロックチェーンシステム」は、ブロックチェーンを動作させるすべてのコンポーネントを指すように意図されている。これは、財布、コード、トランザクション、ブロックチェーンネットワークとともに、ブロックチェーン自体を含む。本開示で使用されるブロックチェーンシステムの例は、ビットコインとイーサリアムを含む。用語ブロックチェーンネットワーク（例えば、イーサリアムブロックチェーンネットワーク）が使用される場合、これは、インターネットのような通信ネットワークを介し

て互いに通信することができるブロックチェーンコードを実行するコンピュータを指すことが意図されている。ブロックチェーンネットワーク中のコンピュータは、ノードと呼ばれ、フルノードは、完全なブロックチェーンのコピーを含む、ブロックチェーンネットワーク中のコンピュータである。

【００３５】

［００４２］

以下は、実例となる例である。以下は、イーサリアムの使用を説明しているが、これは、ブロックチェーンシステムの実例となるように意図されているにすぎない。この主な理由は、イーサリアムがもともとスマートコンタクトをサポートしているからである。スマートコントラクトは、イーサリアムブロックチェーン上にオブジェクトを保持する能力として使用される。これらは、コードの機能を含み、他のコントラクトと相互作用し、新たなコントラクトを生成し、決定し、データを記憶し、イーサを他に送ることができる。

【００３６】

ブロックチェーンコンポーネント

［００４３］

本開示は、ブロックチェーン上の３つの別個のコンポーネントに言及する。これらのコンポーネントは、以下のとおりである。

【００３７】

動的アクセス制御インターフェース

［００４４］

このコンポーネントは、ブロックチェーンプロセス実行を外界に接続する。インターフェースは、送り側が呼び出す外部アプリケーションプログラム可能インターフェース（ＡＰＩ）関数を利用可能にする。送り側は、能力動作を開始するユーザプロセスである。

【００３８】

動作

［００４５］

インターフェースからの関数呼び出しは、次の動作「生成能力」１５２、「許可能力」１５４、「削除能力」１５６、「無効化能力」１５８、および、「呼び出し能力」１５９のうちの１つを呼び出してもよい。これらの動作は、ブロックチェーン上のプロセスインスタンスであってもよい。これらのプロセスインスタンスは、生成、許可、削除、および、無効化能力のプロセスロジックの多くを取り扱い、プロセス状態をブロックチェーン上にも記憶してもよい。

【００３９】

記憶マネージャ

［００４６］

これは、能力データベースとのすべての相互作用を取り扱うコンポーネントである。上記のすべての動作は、能力データベースへのアクセスを少なくとも必要とし、データベース中に記憶されているデータを変更することも伴ってもよい。典型的には、記憶マネージャは、ブロックチェーンに追加される任意の能力トランザクションを追跡するだろう。

【００４０】

オフチェーンコンポーネント

［００４７］

システム中に３つの主なオフチェーンコンポーネントがある。

【００４１】

送り側

［００４８］

上記のように、送り側は、能力動作を開始するユーザプロセスである。

【００４２】

オブジェクト

［００４９］

能力動作は、典型的には、オブジェクトに関連して行われる。典型的には、ユーザは、オブジェクトを生成し、その後、そのオブジェクトへの能力を他のユーザに許可する。オブジェクトは、典型的には、ユーザプロセスまたはリソースであり、任意のユーザプロセスまたはコンピューティングリソースのようなものであってもよい。オブジェクトは、文書やファイルのようなデジタルオブジェクトであってもよく、または、オブジェクトは、建物や場所のような物理的オブジェクトであってもよい。オブジェクトはまた、コンピュータシステムへの電子的アクセスであってもよい。コンピュータシステム内で、オブジェクトは、例えば、変数、データ構造、機能、または方法であることがある。しかしながら、オブジェクトはまた、ネットワークリソース、例えば、別のコンピュータ、ネットワークプリンタまたは記憶装置のように、コンピュータの外部にあることがある。オブジェクトは、そのオブジェクトに対する一意的識別子として機能する、文字列のようなコンピューティングリソースによって表される物理的オブジェクトであることもある。

【００４３】

ターゲット

［００５０］

ターゲットは、送り側の能力動作にしたがって、能力を取得してもよいまたは失ってもよいユーザプロセスである。

【００４４】

例示的なシステム

［００５１］

図１は、提案するシステムの例示的な実例である。このシステムでは、アリス１１０、ボブ１１２、および、キャロル１１４はユーザである。多数のオフチェーンプログラムとして、送り側１２０、ターゲット１３０、および、オブジェクト１４０がある。上記のように、送り側は、通常、能力要求を開始するプログラムであり、送り側が彼らのためにターゲット能力を変更することを望む場合、ターゲットは、オプション的に特定される。典型的に、オブジェクトは、送り側１２０またはターゲット１３０がアクセスしようとするものであり、あるいは、代替的に、１つ以上の能力が許可されるまたは修正されるものに関する。

【００４５】

［００５２］

ブロックチェーンネットワーク１８０上に、多数のスマートコントラクトのプロセスインスタンスがある。動的アクセス制御インターフェースは、外部ＡＰＩ呼び出しを可能にするスマートコントラクトのプロセスインスタンスである。能力として「生成能力」１５２、「許可能力」１５４、「削除能力」１５６、「無効化能力」１５８、および「呼び出し能力」１５９に関する特定の動作に関連する多数のスマートコントラクトのプロセスインスタンスもある。記憶マネージャとして動作するスマートコントラクトのプロセスインスタンスもあり、このプロセスインスタンスは、能力記憶装置１７０プロセスインスタンスと相互作用する。

【００４６】

［００５３］

これらの動作と上記の動作のバリエーションは、より精巧な動作を導出するために使用することができる。これは、サポートされることができる異なる種類の権利がある場合になされてもよい。このセクションで使用される能力は、単なる例示的な表現であり、決して唯一の権利の表現ではない。

【００４７】

［００５４］

上述したように、上記で概説したような動作「生成能力」１５２、「許可能力」１５４、「削除能力」１５６、「無効化能力」１５８、および「呼び出し能力」１５９は、スマートコントラクト記憶スペース中の変数を修正し、能力記録の最新の状態を追跡する。動作をトリガするトランザクションは、能力記録への変更および修正に対する監査証跡を提供

するためにブロックチェーン上に記録される。
【００４８】
能力
［００５５］
能力は、データへの制御されたアクセスの意味を説明するための技術として、デニスとヴァンホーンによって提案された。本開示における能力は、安全なオブジェクト参照である。能力は、システムがその能力に関係付けられているオブジェクトへの特権またはアクセス権のセットを維持するオブジェクトへの参照である。能力ベースのセキュリティモデルでは、安全なオブジェクト参照は、オブジェクトとアクセス権の両方への参照である。アクセス権は、オブジェクトへのアクセスの制限のセットであり、アクセスは、オブジェクトを入力するまたは使用することを含むことができる。能力、特に、アクセス能力は、文書およびファイルのようなデジタルオブジェクトとともに、建物または場所のような物理的オブジェクト、ならびに、コンピュータシステムへの電子的アクセスに関連してもよい。
【００４９】
［００５６］
能力は、（送り側からターゲットまたはユーザプロセスから他の何らかのユーザプロセスのような）ある当事者から別の当事者に送るまたは通信することができるという意味で、通信可能である。能力は、他の何らかの通信に酷似したデータとして通信されてもよい。これは、能力のソースが来た場所をユーザプロセスが決定することを可能にする。
【００５０】
［００５７］
さらに、オブジェクト参照がコピーすることが容易でないまたは計算上偽造することが実行不可能であるように、システムが保護を提供するという点で、能力は安全である。本開示において、ブロックチェーンシステムの使用によって、保護はイネーブルにされる。保護されたオブジェクトの参照のような能力は、１つの実施形態では、ブロックチェーン上の能力に影響を与えるためにブロックチェーン上で実行されてもよい、スマートコントラクトにおける特権的命令の使用を通してのみ生成されることができる。
【００５１】
能力ベースのアクセス制御
［００５８］
ユーザがオブジェクトをアンロックする有効な能力を持っていない限り、能力ベースのアクセス制御システムは、オブジェクトへのアクセスをブロックする。したがって、オブジェクトにアクセスするために、ユーザは、能力を使用しなければならない。これは、オブジェクトにアクセスすることを望んでいるユーザ、ドアをアンロックするために有効な鍵を最初に必要とする、ロックされたドアを有する部屋におけるオブジェクトに概念的に類似している。
【００５２】
［００５９］
能力ベースのアクセス制御は、鍵がユーザに割り振られることができ、ユーザによって再配布できることもできる鍵とロックシナリオのような、能力の所有に基づいており、能力は、ユーザ間の周りで伝搬または渡されることができる。能力の漏出があるとき、制限の問題が生じ、これは、能力伝搬を制限することができないことの結果である。
【００５３】
［００６０］
能力を有するユーザプロセスは、その能力に対するアクセス権によって決定されるような、ある限られた方法で、オブジェクトと相互作用する能力を与えられてもよい。アクセス権の例は、読み取り、書き込み、読み取り／書き込み、および、許可された記憶ロケーションまたは拒否された記憶ロケーションを含む。
【００５４】
［００６１］

これは、ユーザプロセスに与えられたアクセス権の層がある場合、一般的に、階層的保護として動作する他のセキュリティモデルタイプから能力を区別する。階層保護モデルでは、最も信用されるユーザプロセスにアクセス権の最高量が与えられ、最も低く信用されるユーザプロセスにアクセス権の最低量が与えられる。これらのタイプのシステムでは、パス名のようなオブジェクトへの偽造可能な参照は、アクセス権を決定するのに十分ではなく、アクセスは、オブジェクトへのアクセスを要求するユーザプロセスの周囲のオーソリティに依存する。

【００５５】

生成能力

［００６２］

以下は、図１中に図示したような例示的なシステムを利用するための例示的な方法を説明している。アリス１１０は、送り側プログラム１２０へのアクセスを有し、ボブは、ターゲットプログラム１３０へのアクセスを有している。

【００５６】

［００６３］

最初、アリス１１０は、オブジェクト１４０を生成する。オブジェクトの所有者としてのアリスは、オブジェクトへの完全なアクセスを有しており、したがって、オブジェクト上の任意の動作を実行する権限がある。アリスの送り側プログラム１２０は、オブジェクト１４０に対する送り側１２０に関する能力を最初に生成するために、動的アクセス制御インターフェース１５０のＡＰＩにアクセスする。

【００５７】

［００６４］

図２中に図示したように、動的アクセス制御インターフェース１５０は、オブジェクトに関する能力を生成するための要求を、送り側から受け取る２１０。動的アクセス制御インターフェース１５０は、その後、生成能力１５２を呼び出す。生成能力１５２プロセスインスタンスは、最初に、送り側１２０に対するオブジェクトに関する能力の存在を決定する２２０。すなわち、送り側に対するオブジェクトに関する1つの能力があるにすぎず、したがって、このステップは、能力がすでに生成されていないことを確実にするためのものである。送り側に対するオブジェクトに関する能力の存在は、このような能力が記憶マネージャ１６０によって記憶されているか否かを問い合わせることにより、決定されてもよい。記憶マネージャ１６０は、１つ以上の能力記憶装置１７０から問い合わせてもよい。能力記憶装置１７０が能力を含まない場合、能力は存在しないと決定される。送り側に対するオブジェクトに関する能力が存在しないと決定される場合、生成能力１５２プロセスインスタンスは、送り側に対するオブジェクトに関する能力を生成してもよい。

【００５８】

［００６５］

いったん、送り側に対するオブジェクトに関する能力が生成されると、記憶マネージャ１６０は、能力を記憶してもよい。能力のデータ構造は、オブジェクトと権利が任意の長さの文字列である（オブジェクト、権利、派生物）のタプルとして表すことができる。記憶マネージャは、能力を保存するとき、この表現を使用してもよい。ブロックチェーン上の能力を記憶するとき、記憶マネージャは、トランザクションのハッシュを取得してもよい。

【００５９】

［００６６］

以下は、生成能力に関する例示的なアルゴリズムである：

```
Input: object
1 senderCaps = capsdb.get(msg.sender)
2 for all cap in senderCaps do
3     if cap.object == object then
4         return false
5     end if
```

```
6 end for
7 newCap = {object,“master”,0x0}
8 senderCaps.push(newCap)
9 return true
```

【００６０】

［００６７］

最初、アリス１１０は、ボブ１１２を信用し、アリス１１０は、ボブ１１２が彼女のオブジェクト１４０への完全なアクセスを有することを望む。アリスは、オブジェクト１４０へのボブのアクセスを許可したい。システムは、許可能力動作でこのシナリオを取り扱う。

【００６１】

［００６８］

図３中に図示したように、動的アクセス制御インターフェース１５０は、ターゲットに対してオブジェクトに関する能力を許可するための要求を、送り側から受け取る３１０。動的アクセス制御インターフェース１５０は、その後、許可能力１５４プロセスインスタンスを呼び出す。許可能力１５４プロセスインスタンスは、最初に、ターゲット１３０に対するオブジェクト１４０のための送り側１２０に対するアクセス権を決定する３２０。すなわち、許可能力プロセスインスタンスは、オブジェクトに対して送り側が何のアクセス権を有しているかを確認する。許可能力プロセスインスタンスは、その後、オブジェクトに関する能力がターゲットに対して許可されることを、アクセス権が可能にするか否かを決定する３３０。すなわち、許可能力プロセスインスタンスは、送り側がオブジェクト１４０に関する能力をターゲット１３０に許可することを、送り側のアクセス権が可能するか否かを決定する３３０。いったん、送り側に対するオブジェクトに関する能力が許可されると、記憶マネージャは、能力をブロックチェーン上に記憶してもよい３４０。

【００６２】

［００６９］

この例では、ステップ３２０と３３０は、同じプロセスインスタンスにより実行されるが、これらは、異なるプロセスインスタンスにより実行されることが可能である。実際に、すべての例のように、ステップのそれぞれは、1つのプロセスインスタンス内でまたは複数のプロセスインスタンスに渡って実行されることができる。

【００６３】

［００７０］

以下は、許可能力に関する例示的なアルゴリズムである：

```
Input: object,target,rights
1 senderCaps = capsdb.get(msg.sender)
2 for all cap in senderCaps do
3    if cap.object == object then
4       if (“master” in cap.rights) OR (“grant” in cap.rights) then
5       targetCaps 〈- capsdb.get(target)
6       if ((“master” NOT in targetCaps) AND (rights != “master”))
7          newCap = {object,rights,msg.sender}
8          targetCaps.push(newCap)
9          end if
10           return true
11       end if
12    end if
13 end for
14 return false
```

【００６４】

削除能力

［００７１］

ある時期に、アリス１１０は、オブジェクト１４０をボブ１１２ともはや共有する必要性がないというケースがあるかもしれない。このケースでは、アリスは、単に、オブジェクト１４０に関する能力を削除することを望むかもしれない。

【００６５】

［００７２］

図４は、能力動作を削除するための例示的な方法である。この例では、動的アクセス制御インターフェース１５０は、オブジェクトに関する能力を削除するための要求を、送り側から受け取る４１０。動的アクセス制御インターフェース１５０は、その後、削除能力１５６プロセスインスタンスを呼び出す。削除能力１５６プロセスインスタンスは、最初、送り側に対するオブジェクトに関する能力の存在と所有権を決定する４２０。すなわち、削除能力プロセスインスタンスは、送り側がオブジェクトに対して何のアクセス権を有しているかを確認し、特に、送り側が所有者であるか否かを決定する。削除能力プロセスインスタンスは、その後、オブジェクトに関する能力が削除されることをアクセス権が可能にするか否かを決定する。すなわち、削除能力プロセスインスタンスは、送り側がオブジェクト１４０に関する能力を削除することを、送り側のアクセス権が可能にするか否かを決定する４３０。典型的に、送り側がオブジェクト１４０所有者であると決定される場合、送り側は、オブジェクトに関する能力を削除できるだろう。その後、記憶マネージャは、能力を削除してもよい４４０。これは、実際に能力をデータベースから取り除くステップである。

【００６６】

［００７３］

削除能力に関する例示的なアルゴリズムは：

```
Input: object
1 senderCaps = capsdb.get(msg.sender)
2 for all cap in senderCaps do
3    if cap.object == object then
4       if (“master” in cap.rights) then
5          result= revokeAllNonMasterCap(msg.sender,object)
6          if result then
7             senderCaps = senderCaps.remove(cap)
8             return true
9          end if
10       end if
11    end if
12 end for
13 return false
```

【００６７】

［００７４］

削除能力に関する第２の例示的なアルゴリズムは：

```
Input: object
1 senderCaps = capsdb.get(msg.sender)
2 for all cap in senderCaps do
3    if (cap.object == object) && (“master” in cap.rights) then
4       for all key in capsdb do
5          caps = capsdb.get(key)
6          for all c in caps do
7             if (c.object == object) && (“master” NOT in c.rights) then
8                caps = caps.remove(c)
9             end if
10         end for
```

```
11        end for
12        return true
13     end if
14 end for
15 return false
```

【００６８】

無効化能力

［００７５］

アリス１１０とボブ１１２が疎遠になった後、アリス１１０は、彼女のオブジェクト１４０にアクセスを有するボブ１１２をもはや信用しない。アリスは、オブジェクト１４０へのボブのアクセスを無効化したい。残念ながら、アリスは、ボブに、オブジェクトへの完全なアクセスを許可してしまっており、これは、ボブがそのアクセスを彼の友人であるキャロル１１４に委譲できることを意味する。このケースでは、アクセスは、アリスの認識を超えて広がる。これは、隔離問題（confinement problem）と呼ばれるものである。無効化能力１５８動作は、ボブ１１２から派生したすべての能力を無効化し、削除することにより、このシナリオを取り扱うことができる。

【００６９】

［００７６］

図５は、無効化能力動作（revoke capability operation）のための例示的な方法である。この例では、動的アクセス制御インターフェース１５０は、オブジェクトに関する能力を無効にするための要求を、送り側から受け取る５１０。動的アクセス制御インターフェース１５０は、その後、無効化能力１５８プロセスインスタンスを呼び出す。無効化能力１５８プロセスインスタンスは、最初、送り側に対するオブジェクトに関する能力の存在と所有権を決定する５２０。すなわち、無効化能力プロセスインスタンスは、送り側がオブジェクトに対して何のアクセス権を有しているかを確認し、特に、送り側が所有者であるか否かを決定する。

【００７０】

［００７７］

無効化能力プロセスインスタンスは、その後、ターゲットに対するオブジェクトに関する能力が無効にされることを、オブジェクトに関する送り側に対するアクセス権が可能にするか否かを決定する５３０。すなわち、無効化能力プロセスインスタンスは、送り側がターゲットに対するオブジェクト１４０に関する能力を無効化することを、送り側のアクセス権が可能にするか否かを決定する。典型的に、送り側がオブジェクト１４０の所有者であると決定される場合、送り側は、オブジェクトに対する能力を無効化できるだろう。その後、記憶マネージャは、能力を取り除いてもよい５４０。これは、データベースから能力を取り除くステップである。

【００７１】

［００７８］

以下は、無効化能力に関する例示的なアルゴリズムである：

```
Input: object,target,rights
1 senderCaps = capsdb.get(msg.sender)
2 for all cap in senderCaps do
3    if cap.object == object && ((“master” in cap.rights) OR (“revoke” in cap.r
ights)) then
4       targetCaps = capsdb.get(target)
5       for all c in targetCaps do
6          if c.object == object then
7             if rights in c.rights then
8                targetCaps = targetCaps.remove(c)
9             end if
```

```
10          end if
11       end for
12       return true
13    end if
14 end for
15 return false
```

【００７２】

［００７９］

ボブ１１２は、オブジェクト１４０上でアクティビティを実行することを望むかもしれない。サービスリソース所有者は、呼び出し能力動作を呼び出すことにより、ボブがアクティビティを実行することを可能にされているとチェックするだろう。

【００７３】

［００８０］

図６は、呼び出し能力動作（invoke capability operation）のための例示的な方法である。この例では、動的アクセス制御インターフェース１５０は、オブジェクトに関する能力を呼び出すための要求を、送り側から受け取る６１０。動的アクセス制御インターフェース１５０は、その後、呼び出し能力１５９プロセスインスタンスを呼び出す。呼び出し能力１５９プロセスインスタンスは、最初、送り側に対するオブジェクトに関する能力の存在と所有権を決定する６２０。すなわち、呼び出し能力プロセスインスタンスは、送り側がオブジェクトに対して何のアクセス権を有しているかを確認し、特に、送り側が所有者であるか否かを決定する。

【００７４】

［００８１］

呼び出し能力１５９プロセスインスタンスは、その後、ターゲットに対するオブジェクトに関する能力が呼び出されることを、オブジェクトに関する送り側に対するアクセス権が可能にするか否かを決定する６３０。すなわち、呼び出し能力プロセスインスタンスは、送り側がターゲットに対するオブジェクト１４０に関する能力を呼び出すことを、送り側のアクセス権が可能にするか否かを決定する。典型的に、送り側がオブジェクト１４０の所有者であると決定される場合、送り側は、オブジェクトに関する能力を呼び出すことができる。

【００７５】

［００８２］

以下は、呼び出し能力に関する例示的なアルゴリズムである：

```
Input: object,target,rights
1 targetCaps = capsdb.get(target)
2 for all cap in targetCaps do
3    if cap.object == object then
4       if rights in cap.rights then
5          return true
6       end if
7    end if
8 end for
9 return false
```

【００７６】

成功および失敗メッセージ

［００８３］

上記の動作のうちのいずれかにおいて、方法はさらに、呼び出しプロセスインスタンスまたは送り側に、成功またはエラーメッセージを戻すことを備えていてもよい。成功メッセージは、能力動作が通常に完了したことを示す一方で、エラーメッセージは、能力動作が進まなかったことを示している。エラーメッセージは、例えば、アクセス権が動作の進行

を可能にしなかったことを示してもよい。例えば、呼び出し能力のために、上記で説明した方法に続き、呼び出し能力１５９プロセスインスタンスは、送り側に対してオブジェクトの起動に対する成功またはエラーを表すメッセージを戻してもよい６４０。

【００７７】

ハッシュ値

［００８４］

上記の動作のうちのいずれかにおいて、送り側にハッシュ値を戻すことをさらに備えていてもよい。１つの例では、ハッシュ値は、ボブ１１２への能力を無効にするためのトランザクションのようなものである。ハッシュ値は、その後、ブロックチェーンにおいてトランザクションを検索するために使用することができる。トランザクションを検索することは、トランザクション時間またはコンテンツを調査すること、あるいは、能力データベースの現在の状態を決定することに有用であるかもしれない。例えば、トランザクション記録は、ｗｗｗ．ｂｌｏｃｋｃｈａｉｎ．ｉｎｆｏのような公的に利用可能なファシリティを使用して、トランザクション識別子としてトランザクションのハッシュを使用してアクセスされてもよい。

【００７８】

［００８５］

ハッシュを生成するために使用される実際のハッシュアルゴリズムは、ブロックチェーンシステムに依存する。例えば、トランザクションのハッシュは、情報の２５６ビット表現を生成するために、ＳＨＡ－２５６アルゴリズムを使用して決定されてもよい。セキュアハッシュアルゴリズム（ＳＨＡ）ファミリ中の他のアルゴリズムを含む、他のハッシュアルゴリズムが使用されてもよいことを認識すべきである。いくつかの特定の例は、ＳＨＡ３－２２４、ＳＨＡ３－２５６、ＳＨＡ３－３８４、ＳＨＡ３－５１２、ＳＨＡＫＥ１２８、ＳＨＡＫＥ２５６含む、ＳＨＡ－３サブセット中のインスタンスを含む。他のハッシュアルゴリズムは、ＲＡＣＥ Ｉｎｔｅｇｒｉｔｙ Ｐｒｉｍｉｔｉｖｅｓ Ｅｖａｌｕａｔｉｏｎ Ｍｅｓｓａｇｅ Ｄｉｇｅｓｔ（ＲＩＰＥＭＤ）ファミリ中のものを含んでいてもよい。特定の例は、ＲＩＰＥＭＤ－１６０を含んでもよい。他のハッシュ関数は、Ｚｅｍｏｒ－Ｔｉｌｌｉｃｈハッシュ関数に基づくファミリ、および、ナップザックベースのハッシュ関数を含んでいてもよい。

【００７９】

スマートコントラクトの展開

［００８６］

例示的なブロックチェーンシステムであるイーサリアムにおいて、スマートコントラクトは、それが実行される前に展開されなければならない。プロセスインスタンスとしてスマートコントラクトを展開するために、スクリプトのコンパイルされたコード、および、アプリケーションバイナリインターフェース（ＡＢＩ）が必要とされる。ＡＢＩは、動的アクセス制御インターフェース１７０とどのように相互作用するかを定義する。ＡＰＩは、利用されるソースコードに対するインターフェースを定義する一方で、ＡＢＩは、特定のアーキテクチャ上の２つ以上のソフトウェア間の低レベルバイナリインターフェースを定義する。ＡＢＩは、プロセスインスタンスがそれ自体とどのように相互作用するか、プロセスインスタンスがイーサリアムネットワーク１５０とどのように相互作用するか、および、プロセスインスタンス１７０が任意のコードライブラリとどのように相互作用するかを定義する。スマートコントラクトスクリプトをコンパイルするとき、コンパイラは、典型的に、コンパイルされたコードとＡＢＩの両方を生成するだろう。動的アクセス制御インターフェース１７０と相互作用できるように、送り側１２０とターゲット１４０の両方が、動的アクセス制御インターフェース１７０に対するＡＢＩへのアクセスを有していてもよい。

【００８０】

例示的な送り側

［００８７］

図７中に示す送り側１２０は、プロセッサ７０２、メモリ７１０、ネットワークインターフェースデバイス７０６、および、バス７０４を介して互いに通信するインターフェースデバイス７０７を含んでいる。メモリは、図１～６を参照して説明したプロセスのための命令７１２、７１４、および、７１６、ならびに、データを記憶している。プロセッサは、プロセスを実施するために、メモリからの命令を実行する。

【００８１】

［００８８］

プロセッサ７０２は、メモリ７１０上に記憶されている命令を実行する。プロセッサ７０２は、オフチェーンインターフェースデバイス７０６からの、ユーザ１１０によるプロセスへの入力を受け取る。プロセッサ７０２は、ＡＰＩモジュール７１２にしたがう命令を決定する。命令は、動的アクセス制御インターフェース１５０と通信するための関数であってもよい。プロセッサ７０２は、オフチェーンネットワーク１９０上でインターフェース１８０と通信するためのインターフェースモジュール７１６中に記憶されている命令を実行してもよい。

【００８２】

可変実装と用途

［００８９］

能力ベースのアクセス制御は、コンピュータセキュリティシステムの１つのコンポーネントであるということが意図されている。ほとんどのコンピュータセキュリティシステムは、何らかの形式の認証を必要とするだろう。認証は、本開示で与えられる例において説明したような、イーサリアムのような基礎をなすブロックチェーンシステムによって取り扱われてもよい。例えば、人のアドレスに関係付けられたオブジェクトがあってもよいが、このシステムは、人が正しく認証され、正しい人がそのアドレスに関係付けられていると仮定する。認証の多くのバリエーションが、この能力ベースのシステムとともに使用することができることに留意すべきである。

【００８３】

［００９０］

例示的な用途は、本開示に説明したような能力を利用するために、ブロックチェーンシステムと相互作用するネットワークファイルシステムである。このような方法で、ユーザプロセスは、ネットワークファイルシステム中のファイルへの読み取りアクセスの能力を許可されることができる。ユーザプロセスがファイルへの読み取りアクセスを要求し、能力をパラメータとして渡してもよい。ネットワークファイルシステムは、ユーザプロセスがファイルへの適切な読み取りアクセスを確実に有するように、呼び出し能力関数を呼び出してもよい。

【００８４】

［００９１］

さらなる例示的な用途は、リソース制限を有するかもしれないオブジェクトへのアクセスのリアルタイム監視のためのものである。すなわち、能力の活動を追跡することができることから、本開示におけるアプローチの使用は、オブジェクトのリソース使用を監視することが可能であってもよいことを意味する。例えば、１５個の読み取り動作のリソース制限を有するオブジェクトへの読み取り能力を与えられた２０個のユーザプロセスがある場合、ユーザプロセスのうちの四分の三のみが、オブジェクトにアクセスできる。代わりに、いったんユーザプロセスが適切なアクションを実行すると、オブジェクトにアクセスするための能力が無効化され、再割り振りされるか、または、代替的に、アクセスを要求する別のユーザプロセスとアクセスが通信できる場合、システムが１５個の固定された制限を課してもよい。これは、アクセスが必要とされるとき、ユーザプロセスが一般的にアクセスを要求する周囲の認証オーソリティシステムとは対照的である。システムは、ユーザプロセスがアクセスを要求するとき、アクセスを許可してもよく、または、拒否してもよい。

【００８５】

［００９２］
本開示で説明したような能力は、アクセスを委譲するために特に適用可能である。委譲された権限は、多くの状況で発生するかもしれない。現実世界の例では、政府は道路のルールを作成する権限を有しているかもしれないが、政府は、実際の道路ルールの生成を人または政府機関に委譲してもよい。委譲された者は、政府と同じアクセスまたは政府に与えられたものより少ないアクセスを有していてもよい。例えば、政府は、交通ルールを生成し、維持するためのアクセスを有していてもよいが、定期的にルールを更新するために、政府機関のような道路および海上サービスにアクセスを委譲してもよい。本開示の能力アプローチは、このような用途を実装することを単純明快にする。
【００８６】
［００９３］
ブロックチェーンシステム上の上記の能力ベースのアクセス制御は、自動化アクセス制御のために使用されてもよい。自動化アクセス制御の１つの形態は、送り側または送り側のあるカテゴリが、ある能力を委譲されている場合である。１つのこのような例は、デバイスによって実行される認証であってもよい。この例では、水の飲用品質の品質保証を行う水質測定デバイスのようなデバイスのカテゴリは、都市の水供給に渡る水質の測定を記録するための（データベースのような）オブジェクトへのアクセスを与えられてもよい。デバイスが水測定装置の特定のカテゴリであることを考えると、測定データベースへのアクセスを有するように、そのカテゴリ中のすべての水測定デバイスに委譲される能力があってもよい。オブジェクトへのアクセス制御を自動化することができるので、これは有利であり、デバイスに手動で権限を与える必要はない。
【００８７】
［００９４］
上記で開示した動作に関するバリエーションは、他の種類のポリシーをサポートしてもよい。これらのポリシーは、委譲を含み、能力が別のユーザに移動され、または、あるユーザから別のユーザに複製される。ポリシーはまた、第１のユーザが能力を有し、第２のユーザが第１のユーザの能力を無効化する能力を有するような無効化ポリシーを含んでいてもよい。ポリシーは、ユーザ毎のアクセスよりもより複雑なアクセスポリシーをさらに含んでいてもよい。例えば、いくつかのドキュメントは、役割とユーザの関係付けがシステムによって別々に管理される、特定の役割によってアクセスされてもよい。
【００８８】
［００９５］
さまざまなバリエーションおよび／または修正が、特許請求の範囲において規定した範囲から逸脱することなく、特定の実施形態になされてもよいことが当業者によって認識されるだろう。
【００８９】
［００９６］
本開示の技術は、さまざまなテクノロジーを使用して実装することができることを理解すべきである。例えば、ここで説明した方法は、適切なコンピュータ読取可能媒体上に存在する一連のコンピュータ実行可能命令によって実装されてもよい。適切なコンピュータ読取可能媒体は、揮発性（例えば、ＲＡＭ）ならびに／あるいは不揮発性（例えば、ＲＯＭ、ディスク）メモリ、搬送波、および送信媒体を含んでいてもよい。例示的な搬送波は、ローカルネットワーク、または、インターネットのような公にアクセス可能なネットワークとともに、デジタルデータストリームを伝える電子、電磁または光信号の形態をとってもよい。
【００９０】
［００９７］
以下の議論から明らかであるように、そうではないと具体的に述べられない限り、説明全体を通して、「推定すること」または「処理すること」または「演算こと」または「計算すること」、「最適化すること」または「決定すること」または「表示すること」または

「最大化すること」またはこれらに類するもののような用語を利用した議論は、コンピュータシステムのレジスタおよびメモリ内で物理（電子）量として表されるデータを処理して、コンピュータシステムメモリまたはレジスタ、あるいは、他のそのような情報記憶装置、送信または表示デバイス内で物理量として類似して表される他のデータへと変換する、コンピュータシステムまたは類似する電子コンピューティングデバイスのアクションおよびプロセスを指していることが理解される。

【００９１】

［００９８］

したがって、本実施形態は、すべての点で実例的あるとみなされ、制限的であるとはみなされない。

以下に本願の出願当初の特許請求の範囲に記載された発明を付記する。

［Ｃ１］

能力を生成することによる動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、

（ａ）前記オブジェクトに関する能力を生成するための要求を、送り側から受け取ることと、

（ｂ）前記送り側に対する前記オブジェクトに関する能力の存在を決定することと、

（ｃ）前記送り側に対する前記オブジェクトに関する能力が存在しないと決定される場合、前記送り側に対する前記オブジェクトに関する能力を生成することと、

（ｄ）前記送り側に対する前記オブジェクトに関する能力を前記ブロックチェーンシステム上に記憶させることとを備え、前記能力は、前記オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、コンピュータ実装方法。

［Ｃ２］

成功メッセージを前記送り側に送ることをさらに備える、Ｃ１に記載のコンピュータ実装方法。

［Ｃ３］

能力を許可する動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、

（ａ）ターゲットに対してオブジェクトに関する能力を許可するための要求を、送り側から受け取ることと、

（ｂ）前記ターゲットに対する前記オブジェクトのための前記送り側に対するアクセス権を決定することと、

（ｃ）前記オブジェクトに関する能力が前記ターゲットに対して許可されることを、前記アクセス権が可能にするか否かを決定することと、

（ｄ）前記ターゲットに対するオブジェクトに関する能力を前記ブロックチェーンシステム上に記憶させることとを備え、前記能力は、前記オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、

コンピュータ実装方法。

［Ｃ４］

前記能力は、Ｃ１の方法にしたがって生成される、Ｃ３の方法にしたがうブロックチェーンシステム上の能力を許可するためのコンピュータ実装方法。

［Ｃ５］

ステップ（ｃ）は、能力がオブジェクトに対して許可されているか否かを決定することを備える、Ｃ３に記載のコンピュータ実装方法。

［Ｃ６］

前記ターゲットに対する前記オブジェクトに関する能力は、前記送り側に対する前記オブジェクトに関する能力のサブセットである、Ｃ３、４または５に記載のコンピュータ実装方法。

［Ｃ７］

Ｃ１から６のうちのいずれか１項にしたがうブロックチェーン上に能力を生成するためのコンピュータ読取可能命令を有する非一時的コンピュータ読取可能媒体。
［Ｃ８］
能力を生成することによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、
前記システムは、
前記ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスであって、
（ａ）前記オブジェクトに関する能力を生成するための要求を、送り側から受け取り、
（ｂ）前記送り側に対する前記オブジェクトに関する能力の存在を決定し、
（ｃ）前記送り側に対する前記オブジェクトに関する能力が存在しないと決定される場合、前記送り側に対する前記オブジェクトに関する能力を生成し、
（ｄ）前記送り側に対する前記オブジェクトに関する能力を記憶させ、ここにおいて、前記能力は、前記オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、
前記ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスを備える、
システム。
［Ｃ９］
能力を許可することによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、
前記システムは、
前記ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスであって、
（ａ）ターゲットに対して前記オブジェクトに関する能力を許可するための要求を、送り側から受け取り、
（ｂ）前記ターゲットに対する前記オブジェクトのための前記送り側に対するアクセス権を決定し、
（ｃ）前記オブジェクトに関する能力が前記ターゲットに対して許可されることを、前記アクセス権が可能にするか否かを決定し、
（ｄ）前記ターゲットに対する前記オブジェクトに関する能力を記憶させ、ここにおいて、前記能力は、前記オブジェクトに関するアクセス制御を動的に決定するために使用されることができる、
前記ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスを備える、
システム。
［Ｃ１０］
能力を削除することによる動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、
（ａ）前記オブジェクトに関する能力を削除するための要求を、送り側から受け取ることと、
（ｂ）前記送り側に対する前記オブジェクトに関する能力の存在を決定することと、
（ｃ）前記送り側に対する前記オブジェクトに関する能力が存在すると決定される場合、前記送り側に対する前記オブジェクトに関する能力を削除することとを備える、
コンピュータ実装方法。
［Ｃ１１］
能力は、Ｃ１の方法にしたがって生成される、Ｃ１０の方法にしたがう能力を削除するためのコンピュータ実装方法。
［Ｃ１２］

能力を無効にすることによる動的アクセス制御のためのコンピュータ実装方法であって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、
（ａ）ターゲットに対するオブジェクトに関する能力を無効にするための要求を、送り側から受け取ることと、
（ｂ）前記ターゲットに対する前記オブジェクトに関する前記送り側に対するアクセス権を決定することと、
（ｃ）前記オブジェクトに関する能力が前記ターゲットに対して無効にされることを、前記アクセス権が可能にするか否かを決定することと、
（ｄ）前記ターゲットに対する前記オブジェクトに関する能力を無効にすることとを備える、
コンピュータ実装方法。
［Ｃ１３］
能力は、Ｃ１の方法にしたがって生成される、Ｃ１２の方法にしたがう能力を無効にするためのコンピュータ実装方法。
［Ｃ１４］
前記能力が、マスタ能力であるか否かを決定することをさらに備える、Ｃ１２または１３に記載のコンピュータ実装方法。
［Ｃ１５］
能力を削除ことによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、
前記システムは、
前記ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスであって、
（ａ）オブジェクトに関する能力を削除するための要求を、送り側から受け取り、
（ｂ）前記送り側に対する前記オブジェクトに関する能力の存在を決定し、
（ｃ）前記送り側に対する前記オブジェクトに関する能力が存在すると決定される場合、前記送り側に対する前記オブジェクトに関する能力を削除する、
前記ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスを備える、
システム。
［Ｃ１６］
能力を無効にすることによる動的アクセス制御のためのシステムであって、能力は、オブジェクトへの安全な参照であり、前記能力は、ブロックチェーンシステム上に記憶され、
前記システムは、
前記ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスであって、
（ａ）ターゲットに対するオブジェクトに関する能力を無効にするための要求を、送り側から受け取り、
（ｂ）前記ターゲットに対する前記オブジェクトに関する前記送り側に対するアクセス権を決定し、
（ｃ）前記オブジェクトに関する能力が前記ターゲットに対して無効にされることを、前記アクセス権が可能にするか否かを決定し、
（ｄ）前記ターゲットに対する前記オブジェクトに関する能力を無効にする、
前記ブロックチェーン上で実行する１つ以上のスマートコントラクトインスタンスを備える、
システム。

【図面】

【図１】

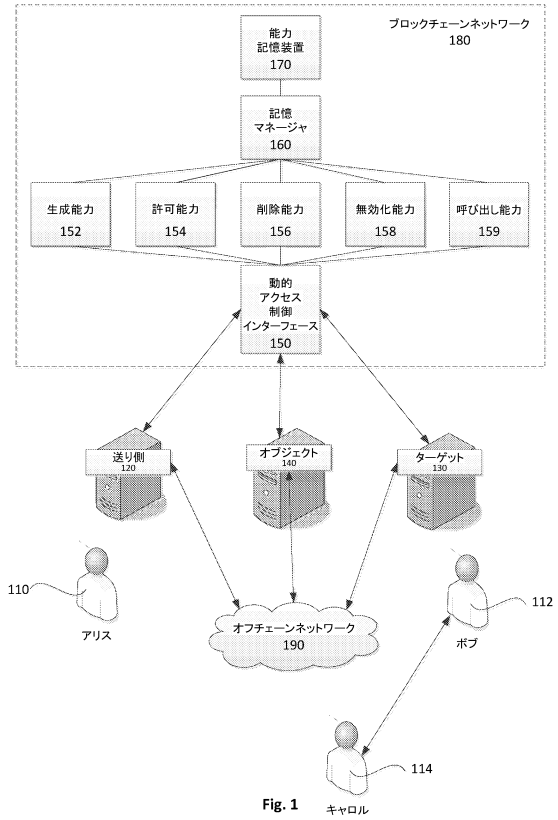

Fig. 1

【図２】

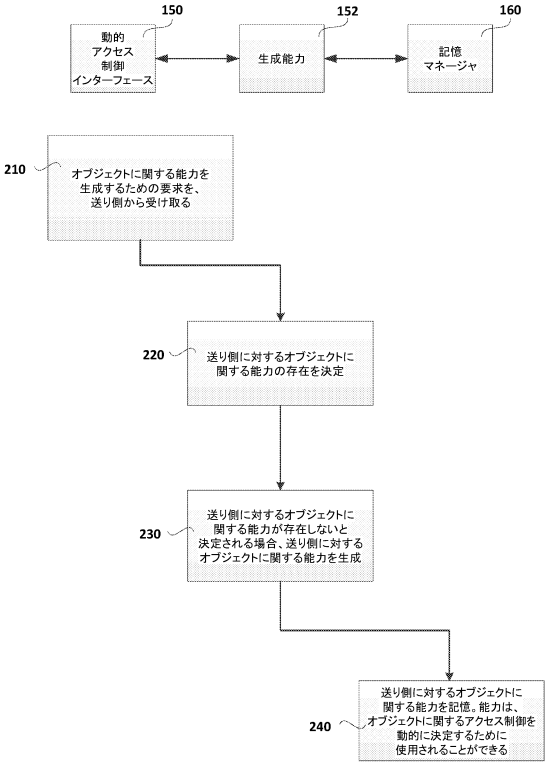

Fig. 2

【図３】

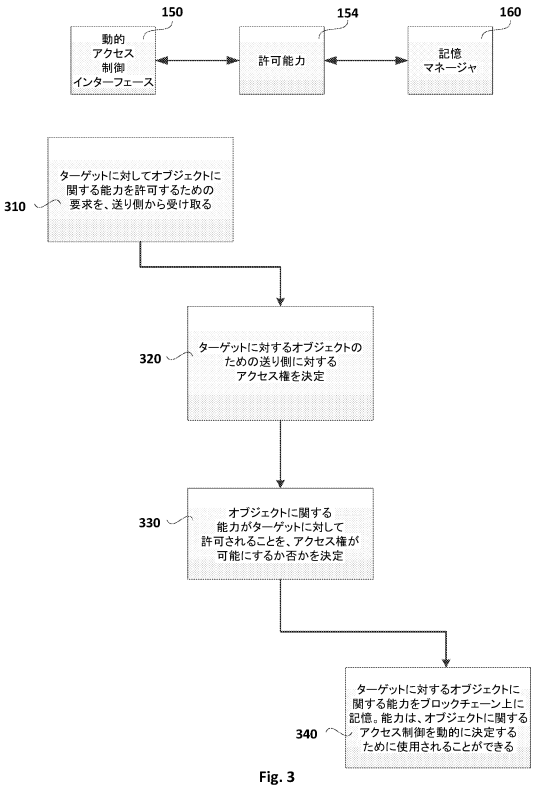

Fig. 3

【図４】

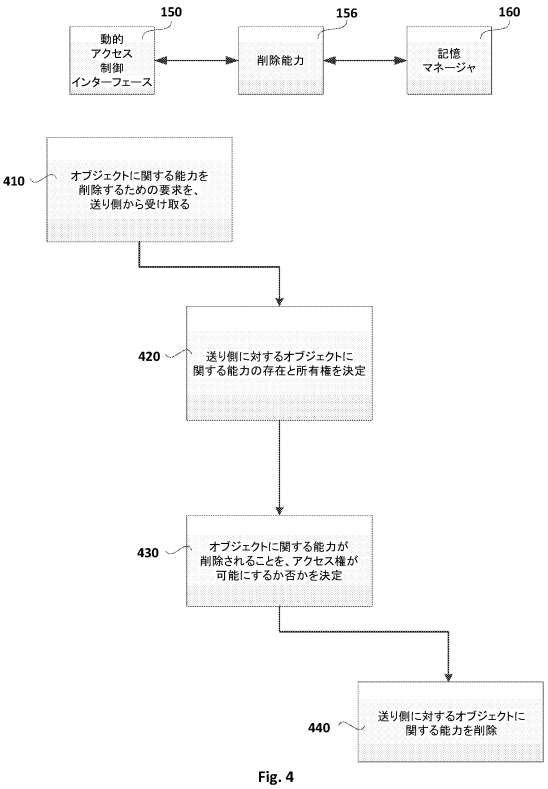

Fig. 4

【図５】

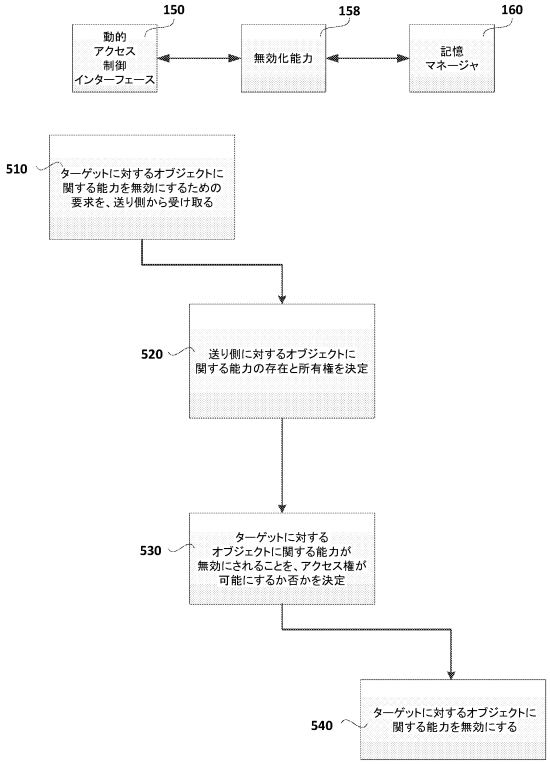

Fig. 5

【図６】

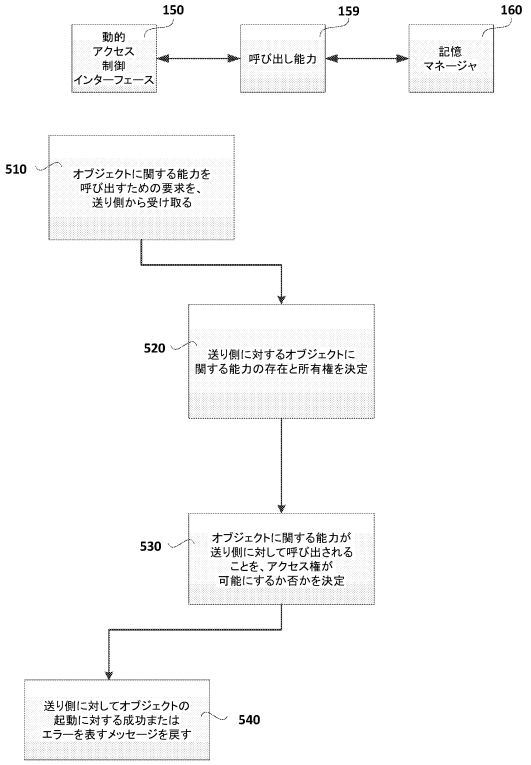

Fig. 6

【図７】

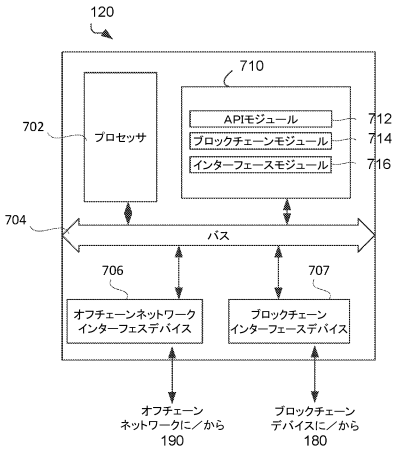

Fig. 7

フロントページの続き

(74)代理人　100199565
弁理士　飯野 茂
(74)代理人　100162570
弁理士　金子 早苗
(72)発明者　ステープルズ、マーク
オーストラリア国、２６０１ オーストラリアン・キャピタル・テリトリー、アクトン、クルーニーズ・ロス・ストリート（番地なし）、コモンウェルス・サイエンティフィック・アンド・インダストリアル・リサーチ・オーガナイゼーション気付
(72)発明者　リンバ、ポール
オーストラリア国、２６０１ オーストラリアン・キャピタル・テリトリー、アクトン、クルーニーズ・ロス・ストリート（番地なし）、コモンウェルス・サイエンティフィック・アンド・インダストリアル・リサーチ・オーガナイゼーション気付
審査官　青木 重徳

(56)参考文献　特開２０００－０９９４７７（ＪＰ，Ａ）
米国特許出願公開第２０１６／００２８５５２（ＵＳ，Ａ１）
ZYSKIND,G. et al.，Decentralizing Privacy:Using Blockchain to Protect Personal Data，2015 IEEEE CS Security and Privacy Workshops,21-22 May 2015,pages 180-184，2015年05月21日
浅川 直輝，４社が試す潜在能力，日経コンピュータ ｎｏ．９１６ NIKKEI COMPUTER，日本，日経ＢＰ社 Nikkei Business Publications,Inc.，2016年07月05日

(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／６０
Ｈ０４Ｌ　９／３２
Ｇ０６Ｆ　２１／６２