(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2021-39719
(P2021-39719A)
(43)公開日　令和3年3月11日(2021.3.11)

(51)Int.Cl.		F I		テーマコード（参考）
G06F 21/60	(2013.01)	G06F	21/60	360
H04L 9/14	(2006.01)	H04L	9/00	641

審査請求 未請求　請求項の数 10　ＯＬ　（全 21 頁）

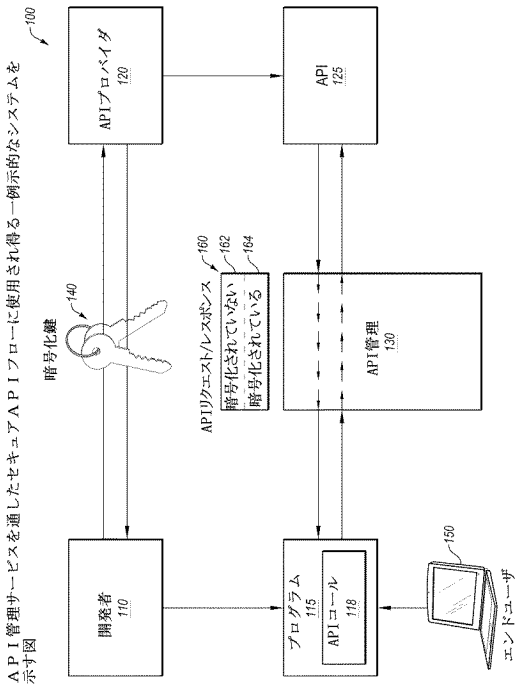
(21)出願番号　特願2020-46327 (P2020-46327)
(22)出願日　令和2年3月17日 (2020.3.17)
(31)優先権主張番号　16/559405
(32)優先日　令和1年9月3日 (2019.9.3)
(33)優先権主張国・地域又は機関
米国 (US)

(71)出願人　000005223
富士通株式会社
神奈川県川崎市中原区上小田中４丁目１番１号
(74)代理人　100107766
弁理士　伊東　忠重
(74)代理人　100070150
弁理士　伊東　忠彦
(72)発明者　バーラミ・メフディ
アメリカ合衆国，カリフォルニア州　９４０８５，サニーヴェイル，イースト　アークス　アヴェニュ　１２４０番　フジツウ　ラボラトリーズ　アメリカ内

最終頁に続く

(54)【発明の名称】セキュアＡＰＩフロー

(57)【要約】
【課題】　セキュアＡＰＩフローを提供する。
【解決手段】　一方法が、アプリケーションプログラミングインターフェース（ＡＰＩ）に対する入力データを取得するステップと、ＡＰＩに対する入力データをＡＰＩのプロバイダの公開鍵を使用して暗号化するステップとを含み得る。当該方法は、ＡＰＩを呼び出すＡＰＩリクエストをＡＰＩ管理サーバに送信するステップであり、ＡＰＩリクエストは、ＡＰＩに対するＡＰＩコールと暗号化された入力データとを含む、ステップをさらに含み得る。ＡＰＩリクエストは、ＡＰＩ管理サーバがＡＰＩコールに基づきＡＰＩ管理サービスを実行できるが暗号化された入力データを公開鍵で復号できないフォーマットであり得る。
【選択図】　図１



API管理サービスを通したセキュアＡＰＩフローに使用され得る一例示的なシステムを示す図

【特許請求の範囲】
【請求項１】
アプリケーションプログラミングインターフェース（ＡＰＩ）に対する入力データを取得するステップと、
前記ＡＰＩに対する前記入力データを前記ＡＰＩのプロバイダの公開鍵を使用して暗号化するステップと、
前記ＡＰＩを呼び出すＡＰＩリクエストをＡＰＩ管理サーバに送信するステップであり、前記ＡＰＩリクエストは、前記ＡＰＩに対するＡＰＩコールと前記暗号化された入力データとを含み、前記ＡＰＩリクエストは、前記ＡＰＩ管理サーバが前記ＡＰＩコールに基づきＡＰＩ管理サービスを実行できるが前記暗号化された入力データを前記公開鍵で復号できないフォーマットである、ステップと、
を含む方法。
【請求項２】
前記ＡＰＩ管理サーバから前記ＡＰＩリクエストに対するレスポンスを受信するステップであり、前記ＡＰＩリクエストに対する前記レスポンスは、前記入力データに基づく前記ＡＰＩの出力を含み、前記出力は、クライアントの公開鍵を使用して前記ＡＰＩプロバイダにより暗号化される、ステップをさらに含む請求項１に記載の方法。
【請求項３】
前記ＡＰＩリクエストに対する前記レスポンスを前記クライアントの秘密鍵を使用して復号するステップをさらに含む請求項２に記載の方法。
【請求項４】
前記ＡＰＩリクエストは、クライアント識別子とＡＰＩプロバイダ識別子とＡＰＩ管理承認構造とを含む暗号化されていないデータコンポーネントをさらに含む、請求項１に記載の方法。
【請求項５】
開発者データを前記ＡＰＩリクエストの一部であるように暗号化するステップであり、前記開発者データは、前記入力データを取得する前に開発者により提供される、ステップをさらに含む請求項１に記載の方法。
【請求項６】
前記プロバイダの前記公開鍵は、前記ＡＰＩリクエストを生成するコンピュータプログラム内に埋め込まれ、それにより、前記入力データの前記暗号化はユーザ入力なしに生じる、請求項１に記載の方法。
【請求項７】
前記ＡＰＩ管理サービスは、トラフィック制御、管理ポリシー実施、及びセキュリティポリシー実施のうち少なくとも１つを含む、請求項１に記載の方法。
【請求項８】
プロセッサに動作を実行させるコンピュータプログラムであって、前記動作は、
アプリケーションプログラミングインターフェース（ＡＰＩ）に対する入力データを取得することと、
前記ＡＰＩに対する前記入力データを前記ＡＰＩのプロバイダの公開鍵を使用して暗号化することと、
通信コンポーネントに、前記ＡＰＩを呼び出すＡＰＩリクエストをＡＰＩ管理サーバに送信するように指示することであり、前記ＡＰＩリクエストは、前記ＡＰＩに対するＡＰＩコールと前記暗号化された入力データとを含み、前記ＡＰＩリクエストは、前記ＡＰＩ管理サーバが前記ＡＰＩコールに基づきＡＰＩ管理サービスを実行できるが前記暗号化された入力データを前記公開鍵で復号できないフォーマットである、ことと、
を含む、コンピュータプログラム。
【請求項９】
前記動作は、前記ＡＰＩ管理サーバから前記ＡＰＩリクエストに対するレスポンスを受信することであり、前記ＡＰＩリクエストに対する前記レスポンスは、前記入力データに

基づく前記ＡＰＩの出力を含み、前記出力は、クライアントの公開鍵を使用して前記ＡＰＩプロバイダにより暗号化される、ことをさらに含む、請求項８に記載のコンピュータプログラム。

【請求項１０】

システムであって、

１つ以上のプロセッサと、

前記１つ以上のプロセッサにより実行されることに応答して当該システムに動作を実行させる命令を含む１つ以上の非一時的コンピュータ読取可能媒体と、

を含み、前記動作は、

アプリケーションプログラミングインターフェース（ＡＰＩ）に対する入力データを取得することと、

前記ＡＰＩに対する前記入力データを前記ＡＰＩのプロバイダの公開鍵を使用して暗号化することと、

通信コンポーネントに、前記ＡＰＩを呼び出すＡＰＩリクエストをＡＰＩ管理サーバに送信するように指示することであり、前記ＡＰＩリクエストは、前記ＡＰＩに対するＡＰＩコールと前記暗号化された入力データとを含み、前記ＡＰＩリクエストは、前記ＡＰＩ管理サーバが前記ＡＰＩコールに基づきＡＰＩ管理サービスを実行できるが前記暗号化された入力データを前記公開鍵で復号できないフォーマットである、ことと、

を含む、システム。

【発明の詳細な説明】

【技術分野】

【０００１】

本開示の実施形態は、セキュアなアプリケーションプログラミングインターフェース（ＡＰＩ）フローに関する。

【背景技術】

【０００２】

ＡＰＩは、二次的なシステムからの１つ以上のプログラミング機能へのアクセスを許可するために使用される。例えば、クライアントデバイス上で動作しているプログラムが、動作中にＡＰＩを呼び出すことがあり、ＡＰＩリクエストをＡＰＩプロバイダに送信することがある。ＡＰＩプロバイダはＡＰＩリクエストを処理し、レスポンスを送信することができる。頻繁に使用されるＡＰＩの一例がFACEBOOK（登録商標）ログインＡＰＩであり、第三者はこれを介し、関連づけられたＡＰＩコールと共にFACEBOOK（登録商標）ログインＡＰＩプロバイダにそのログインクレデンシャルを送信することによりユーザを確認し、該ＡＰＩプロバイダはＡＰＩコールを処理し、レスポンスを送信する。

【発明の概要】

【０００３】

本開示の１つ以上の実施形態は、アプリケーションプログラミングインターフェース（ＡＰＩ）に対する入力データを取得するステップと、ＡＰＩに対する入力データをＡＰＩのプロバイダの公開鍵を使用して暗号化するステップとを含む方法を含み得る。当該方法は、ＡＰＩを呼び出すＡＰＩリクエストをＡＰＩ管理サーバに送信するステップであり、ＡＰＩリクエストは、ＡＰＩに対するＡＰＩコールと暗号化された入力データとを含む、ステップをさらに含み得る。ＡＰＩリクエストは、ＡＰＩ管理サーバがＡＰＩコールに基づきＡＰＩ管理サービスを実行できるが暗号化された入力データを公開鍵で復号できないフォーマットであり得る。

【０００４】

実施形態の目的及び利点は、少なくとも特許請求の範囲において特に指し示された要素、特徴、及び組み合わせにより実現され、達成される。

【０００５】

前述の一般的な説明及び以下の詳細な説明の双方が単に例であり、説明的であり、限定でないことが理解されるべきである。

【図面の簡単な説明】
【０００６】
例示的な実施形態が、添付図面の使用を通してさらなる特定性及び詳細と共に記載され、説明される。
【図１】ＡＰＩ管理サービスを通したセキュアＡＰＩフローに使用され得る一例示的なシステムを示す図である。
【図２】セキュアＡＰＩフローで使用される一例示的なＡＰＩリクエストを示す。
【図３】セキュアＡＰＩフローで使用される一例示的なＡＰＩリクエストの一代替図を示す。
【図４】図４、図５Ａ、及び図５Ｂは、セキュアＡＰＩフローの一例示的な実装の一例示的なスイムレーン図及び関連フローチャートをそれぞれ示す。
【図５Ａ】図４、図５Ａ、及び図５Ｂは、セキュアＡＰＩフローの一例示的な実装の一例示的なスイムレーン図及び関連フローチャートをそれぞれ示す。
【図５Ｂ】図４、図５Ａ、及び図５Ｂは、セキュアＡＰＩフローの一例示的な実装の一例示的なスイムレーン図及び関連フローチャートをそれぞれ示す。
【図６】一例示的なコンピューティングシステムを示す。
【発明を実施するための形態】
【０００７】
本開示は、ＡＰＩ管理サービスを使用するとき秘密データを安全にするために公開及び秘密鍵ペアを使用することに関する。ＡＰＩ管理サービスには、ＡＰＩのクライアントとＡＰＩプロバイダとの間で「仲立」又は「仲介」として動作するサービスを含み得る。例えば、ＡＰＩ管理サービスは、クレデンシャルを収集し、かつ／あるいは複数の異なるＡＰＩプロバイダからの複数のＡＰＩへのアクセスを提供することができる。ＡＰＩ管理サービスへのアクセスを有することにより、ユーザは単一のシステムを通して複数のＡＰＩにアクセスすることができる。しかしながら、ＡＰＩ管理サービスを通してＡＰＩリクエストを送信するとき、ＡＰＩリクエストデータがＡＰＩ管理サービスに晒される可能性がある。ＡＰＩリクエストをサブミットするとき、リクエストデータは機密データの可能性がある。例えば、FACEBOOK（登録商標）ログインＡＰＩのリクエストをサブミットするとき、リクエストは、ユーザのFACEBOOK（登録商標）アカウントのパスワードを含む。
【０００８】
本開示は、ＡＰＩ管理サービスを通過するＡＰＩフローなどの、ＡＰＩ管理サービスと対話するためのセキュアなアプローチを提供する実施形態を提供する。開発者は、ＡＰＩを使用する許可を要求するとき、ＡＰＩプロバイダと公開鍵を交換することができる。開発者は、開発者により開発されたソフトウェアがＡＰＩコールを呼び出すときに使用される開発者秘密鍵及びＡＰＩプロバイダ公開鍵を組み込むことができる。例えば、ＡＰＩリクエストと共に送信される入力データ及び／又は開発者データ（ＡＰＩアクセス情報など）は、ＡＰＩプロバイダの公開鍵を使用して暗号化されてもよい。入力データ及び／又は開発者データを暗号化することにより、ＡＰＩ管理サービスは、入力データを観察又は復号できることなく、ＡＰＩリクエストに対して期待される管理サービスを依然として実行し得る。例えば、ＡＰＩリクエストのメタデータは暗号化されていない可能性がある。ＡＰＩ管理サービスを実行した後、ＡＰＩ管理サーバはＡＰＩリクエストをＡＰＩプロバイダに転送し得る。ＡＰＩプロバイダは、その独自の秘密鍵を使用して入力データ及び／又は開発者データを復号し、入力データを使用して入力データを使用するＡＰＩを呼び出し得る。ＡＰＩからのレスポンスは、開発者の公開鍵を使用して暗号化されてもよい。レスポンスは反対に通信されてもよく、再度、ＡＰＩ管理サーバは、レスポンスデータを観察又は復号できることなく管理サービスを実行する。ＡＰＩクライアントは、開発者の秘密鍵を使用してレスポンスを復号し得る。このようにして、ＡＰＩへの入力データ／開発者データとＡＰＩを呼び出した後のレスポンス（例えば、ＡＰＩの出力）の双方が、ＡＰＩ管理サービスを通過するときに暗号化されている。
【０００９】

本開示の特定の実施形態は、ＡＰＩ管理サービスの従前の反復を上回る向上を提供し得る。例えば、本開示の実施形態は、情報の限られた露出を可能にすることによりＡＰＩクライアントとＡＰＩプロバイダとの間のよりセキュアな対話を提供することができる。さらに、本開示は、ＡＰＩ管理サービスが実行されることを依然として可能にする方法でそうすることができる。本明細書で用いられるとき、用語「ＡＰＩクライアント」は、ＡＰＩコールを呼び出す任意のエンティティ、デバイス、プログラム等を参照し得る。

【００１０】

１つ以上の例示的な実施形態が、添付の図面を参照して説明される。

【００１１】

図１は、本開示の１つ以上の実施形態による、セキュアＡＰＩフローに使用され得る一例示的なシステム１００を示す図である。システム１００は、開発者（developer）１１０及びＡＰＩプロバイダ１２０を含み得る。開発者１１０は、ＡＰＩプロバイダ１２０により提供されるＡＰＩ１２５を呼び出すＡＰＩコール１１８を有するプログラム１１５を開発することができる。ＡＰＩ管理サービス１３０は、ＡＰＩ管理サービスを提供することができる。エンドユーザデバイス１５０は、ＡＰＩコール１１８を呼び出すプログラム１１５を利用することができる。

【００１２】

動作において、開発者１１０及びＡＰＩプロバイダ１２０は、暗号化鍵１４０を交換することができる。例えば、開発者１１０が秘密／公開鍵ペアを生成してもよく、開発者１１０により生成された公開鍵をＡＰＩプロバイダ１２０に提供してもよい。これら及び他の実施形態において、開発者１１０は、秘密／公開鍵ペアを一般的に開発者１１０のために利用してもよく、あるいはそのようなペアを特定のプログラム（プログラム１１５など）のために生成してもよく、あるいはそのようなペアをプログラム１１５の特定のエンドユーザ又はライセンスのために生成してもよい。別の例として、ＡＰＩプロバイダ１２０が秘密／公開鍵ペアを生成してもよく、ＡＰＩプロバイダ１２０により生成された公開鍵を開発者１１０に提供してもよい。これら及び他の実施形態において、ＡＰＩプロバイダ１２０は、秘密／公開鍵ペアを一般的にＡＰＩプロバイダ１２０のために利用してもよく、あるいはそのようなペアを特定のＡＰＩ（ＡＰＩ１２５など）のために生成してもよく、あるいはそのようなペアを特定の開発者１１０のために生成してもよい。

【００１３】

開発者１１０は、プログラム１１５を開発及び／又は実装するときに暗号化鍵１４０を利用することができる。いくつかの実施形態において、プログラム１１５は、ＡＰＩコール１１８を呼び出すとき、ＡＰＩプロバイダ１２０から開発者１１０により受信された公開鍵を使用してプログラム１１５がＡＰＩコール１１８への入力データを暗号化し得るように、コード化されてもよい。さらに、プログラム１１５は、開発者１１０及び／又はプログラム１１５によるＡＰＩ１２５へのアクセスを可能にするためのセキュア情報などの開発者データを暗号化してもよい。そうする際、プログラム１１５は、ＡＰＩリクエスト１６０の一部分を暗号化してもよく（例えば、暗号化された部分１６４）、ＡＰＩリクエスト１６０の一部分を暗号化されないままにしてもよい（例えば、暗号化されていない部分１６２）。このような暗号化されていない情報には、ＡＰＩリクエスト１６０に関するメタデータ、例えば、リクエストがどこにルーティングされるべきか、利用されるＨＴＴＰ動詞機能（verb function）などを含んでもよい。このようにして、ＡＰＩ管理サービス１３０は、ＡＰＩプロバイダ１２０の秘密鍵なしでは暗号化された部分を復号できないと同時に、ＡＰＩ管理サービス１３０により観察可能な暗号化されていない部分１６２に基づいてＡＰＩリクエスト１６０に対する管理タスクを実行することができる。任意の管理タスクを実行した後、ＡＰＩ管理サービス１３０はＡＰＩリクエスト１６０をＡＰＩプロバイダ１２０に転送してもよく、それにより、ＡＰＩ１２５は入力データを処理することができる。

【００１４】

いくつかの実施形態において、暗号化された部分１６４を有するＡＰＩリクエスト１６

０を受信したとき、ＡＰＩプロバイダ１２０は、ＡＰＩプロバイダ１２０の秘密鍵を使用して、暗号化された部分１６４を復号することができる。通信及びＡＰＩ管理は、入力データに対して動作するＡＰＩ１２５からのＡＰＩレスポンスに基づいて、同様の方法で反対に生じてもよい。例えば、ＡＰＩプロバイダ１２０は、開発者１１０の公開鍵を使用してレスポンスを暗号化し、レスポンスをサブミットし（submit）、レスポンスの部分を暗号化されないままにしてもよく（例えば、メタデータ）、それにより、ＡＰＩ管理サービス１３０は、レスポンスに対して管理サービスを実行することができる。任意の管理タスクを実行した後、ＡＰＩ管理サービス１３０はレスポンスをプログラム１１５へ転送してもよく、それにより、プログラム１１５はレスポンスを利用することができる。例えば、プログラム１１５は、プログラム１１５が開発者１１０の秘密鍵を利用してレスポンスを復号するように、コード化されてもよい。

【００１５】

いくつかの実施形態において、本開示はＡＰＩ認証との互換性を残す。例えば、開発者１１０がその公開鍵をＡＰＩプロバイダ１２０に提供するとき、ＡＰＩプロバイダ１２０は、ＡＰＩ１２５へのアクセスを許可するクレデンシャル又は他の認証情報を開発者１１０に提供してもよい。いくつかの実施形態において、認証情報は、暗号化された部分１６４又は暗号化されていない部分１６２の一部でもよい。例えば、ハイパーテキストトランスファープロトコル（ＨＴＴＰ）認証について、ＡＰＩリクエスト１６０の中でユーザ名及びパスワードが提供されて、ＡＰＩプロバイダ１２０に対して真正性を証明してもよい。このような例において、ユーザ名及び／又はパスワードは、暗号化された部分１６４に含まれてもよい。認証の他の例には、ＡＰＩキー、オープン認証（ＯＡｕｔｈバージョン１．０、２．０等を含むＯＡｕｔｈ）が含まれる。

【００１６】

いくつかの実施形態において、ＡＰＩ認証情報は、開発者データの一部として暗号化された部分１６４の一部である。開発者データの１つ以上の他のコンポーネントが、開発者１１０及び／又はプログラム１１５のセキュア識別情報などの、暗号化された部分１６４の一部でもよい。

【００１７】

開発者１１０は、ＡＰＩプロバイダ１２０からのＡＰＩ１２５の使用の許可を開始し得る任意のエンティティ、当事者（party）、又は組織を含むことができる。例えば、開発者１１０は、ソフトウェアプログラマ、ウェブ開発者、アプリ開発者、ハードウェア開発者等を含んでもよい。これら及び他の実施形態において、プログラム１１５は、ＡＰＩコール１１８を呼び出す任意の実行可能命令を含んでもよい。例えば、プログラム１１５は、ソフトウェアプログラム、ウェブスクリプト、ハードウェアスクリプト等を含んでもよい。

【００１８】

エンドユーザデバイス１５０は、プログラム１１５を動作させてＡＰＩコール１１８を呼び出す任意のデバイスを含むことができる。例えば、プログラム１１５がウェブスクリプトである場合、エンドユーザデバイス１５０は、スクリプトを処理するブラウザ又は他のウェブアクセスソフトウェアを有するデバイスでもよい。別の例として、プログラム１１５が（例えば、スマートウォッチ又は他のモノのインターネット（ＩＯＴ）デバイスの）ハードウェアスクリプトである場合、エンドユーザデバイスは、ハードウェアスクリプトを実行するデバイスでもよい。いくつかの実施形態において、エンドユーザデバイス１５０は、入力データのためのユーザ入力を受信することができる。例えば、エンドユーザデバイス１５０のユーザが、ＡＰＩリクエスト１６０における入力データとして使用されるテキスト文字列、画像等を入力してもよい。いくつかの実施形態において、開発者１１０は、例えばテスト、初期実装、開発等の間、エンドユーザデバイス１５０として操作してもよい。

【００１９】

ＡＰＩプロバイダ１２０は、ＡＰＩ１２５などのＡＰＩへのアクセスを開発、ホスト、

又はその他の方法で提供し得る任意のエンティティ、当事者、又は組織を含むことができる。ＡＰＩプロバイダ１２０は、任意数のＡＰＩへのアクセスを提供することができる。

【００２０】

暗号化鍵１４０は、通信を安全にするために使用される任意タイプの公開／秘密鍵ペアリング及び／又は暗号化アルゴリズムを含むことができる。例えば、暗号化鍵１４０は、セキュアシェル（secure shell、ＳＳＨ）公開鍵暗号化で動作してもよい。別の例として、暗号化鍵１４０は、データ暗号化標準（Data Encryption Standard、ＤＥＳ）、トリプルＤＥＳ、ＲＳＡ、Ｂｌｏｗｆｉｓｈ、Ｔｗｏｆｉｓｈ、高度暗号化標準（Advanced Encryption Standard、ＡＥＳ）等と関連して導出及び／又は使用されてもよい。

【００２１】

ＡＰＩ管理サービス１３０は、ＡＰＩへのアクセスを提供し、かつ／あるいはＡＰＩプロバイダ１２０及び／又はエンドユーザデバイス１５０を行き来する又はこれらの間のトラフィックに対する管理タスクを実行するように構成された、任意のシステム、デバイス、コンポーネント、サーバ等を含むことができる。例えば、管理タスクは、トラフィック制御（例えば、ＡＰＩコール数の制限、ＡＰＩコールがサブミットされるレートの制御等）、管理ポリシー（例えば、ＡＰＩリクエストの分類、ＡＰＩリクエストの優先順位付け等）、セキュリティポリシー（例えば、エンドユーザデバイス１５０のクレデンシャルの確認等）などを含んでもよい。

【００２２】

「中間者」として示されているが、いくつかの実施形態において、ＡＰＩ管理サービス１３０は、ＡＰＩクライアントの一部として動作してもよく（例えば、第三者ＡＰＩ管理サービスへのプログラム呼び出し、プログラム１１５に含まれるソフトウェアパッケージなどの、プログラム１１５の一部としてコード化されてもよい）、それにより、ＡＰＩプロバイダ１２０は、プログラム１１５がＡＰＩ管理サービス１３０を使用していることを認識しなくてもよい。さらに又は代わりに、ＡＰＩ管理サービス１３０は、ＡＰＩプロバイダ１２０の一部として動作してもよく、それにより、プログラム１１５及び／又は開発者１１０は、ＡＰＩプロバイダ１２０がＡＰＩ管理サービス１３０を使用していることを認識しなくてもよい。いくつかの実施形態において、他方に認識させない能力は、ＡＰＩリクエスト／レスポンス１６０が向けられるアドレス（例えば、直接他の当事者でなくＡＰＩ管理サービス１３０へ）に基づいてもよい。

【００２３】

本開示の範囲から逸脱することなく、システム１００に対して変更、追加、又は省略がなされてもよい。例えば、記載された方法における異なる要素の指定は、本明細書に記載される概念の説明を助けることが意図され、限定的ではない。さらに、システム１００は、任意数の他の要素を含んでもよく、あるいは記載されたものとは他のシステム又は文脈において実現されてもよい。例えば、図１のコンポーネントのうち任意のものが、さらなるコンポーネントに分割されてもよく、あるいはより少ないコンポーネントに結合されてもよい。

【００２４】

図２は、本開示の１つ以上の実施形態による、セキュアＡＰＩフローで使用される一例示的なＡＰＩリクエスト２００を示す。ＡＰＩリクエスト２００は、図１のＡＰＩリクエスト／レスポンス１６０と同様であり、あるいは相当し得る。図２に示すように、ＡＰＩリクエスト２００は、暗号化されていない部分２１０及び暗号化された部分２２０を含み得る。暗号化されていない部分２１０は、通信中にＡＰＩ管理サービスに可視の部分を含むことができ、暗号化された部分は、秘密鍵がエンティティにより保持されている場合のみ見える部分を含むことができる（例えば、ＡＰＩプロバイダは、暗号化部分を暗号化するために使用された公開鍵に関連づけられた秘密鍵を維持する）。

【００２５】

いくつかの実施形態において、暗号化されていない部分２１０は、クライアント識別子フィールド２１２、プロバイダ識別子フィールド２１４、及び／又は承認情報フィールド

２１６を含んでもよい。いくつかの実施形態において、クライアント識別子フィールド２１２は、ＡＰＩを呼び出すデバイス、コンポーネント、システム、ソフトウェア等、及び／又はユーザ（ＡＰＩクライアントと呼ばれる）のための識別子を含んでもよい。そのような識別子には、非機密の（例えば、クライアント識別子が潜在的に悪意のある当事者により観察されることがＡＰＩクライアントにとって有害でない）識別子を含んでもよい。クライアント識別子フィールド２１２内のクライアント識別子は、ＡＰＩレスポンスがどこに送信されるべきかについての識別子又は他の位置情報の役割を果たしてもよい。このような識別子の例には、インターネットプロトコル（ＩＰ）アドレス、媒体アクセス制御（ＭＡＣ）アドレス、ユーザ名等を含んでもよい。

【００２６】

いくつかの実施形態において、プロバイダ識別子フィールド２１４は、ＡＰＩをホスト又はその他の方法で提供するデバイス、コンポーネント、システム、ソフトウェア等、及び／又は組織が識別され得る識別子を含んでもよい。そのような識別子には、非機密の識別子を含んでもよい。プロバイダ識別子フィールド２１４内のＡＰＩプロバイダ識別子は、ＡＰＩリクエストがどこに送信されるべきかについての識別子又は他の位置情報の役割を果たしてもよい。このような識別子の例には、ＩＰアドレス、ＭＡＣアドレス、ＵＲＬ、組織名等を含んでもよい。

【００２７】

承認情報フィールド２１６は、ＡＰＩサービスマネージャ及び／又はＡＰＩプロバイダとの間でＡＰＩクライアントを確認するために使用される任意の情報を含んでもよい。例えば、承認情報フィールド２１６は、ＡＰＩクライアントがＡＰＩサービスマネージャにより検証され得るためのトークン又は他の認証情報を含んでもよい。別の例として、承認情報フィールド２１６は、ＡＰＩプロバイダとの間でクライアントを認証するために使用される認証情報を含んでもよい。いくつかの実施形態において、承認情報フィールド２１６は、暗号化されていない部分２１０でなく暗号化された部分２２０に含まれてもよい。

【００２８】

いくつかの実施形態において、暗号化された部分２２０は、機密認証情報２２２及び／又はデータペイロード２２４を含んでもよい。機密認証情報２２２は、ＡＰＩクライアントがＡＰＩプロバイダとの間で自身を認証し得るための情報だが、潜在的に悪意のある当事者により観察された場合に危険にさらす可能性がある情報（例えば、ＡＰＩクライアントがＡＰＩプロバイダとの間で自身を認証するために使用するパスワード）を含んでもよい。データペイロード２２４は、ＡＰＩリクエスト２００が送信されているＡＰＩへの入力として提供される任意の情報を含んでもよい。例えば、データペイロード２２４は、テキスト、画像、ファイル、又は他の任意のデータを含んでもよい。

【００２９】

本開示の範囲から逸脱することなく、ＡＰＩリクエスト２００に対して変更、追加、又は省略がなされてもよい。例えば、記載された方法における異なる要素の指定は、本明細書に記載される概念の説明を助けることが意図され、限定的ではない。さらに、ＡＰＩリクエスト２００は、任意数の他の要素を含んでもよく、あるいは記載されたものとは他のシステム又は文脈において実現されてもよい。例えば、ＡＰＩリクエスト２００は、暗号化されていない部分２１０又は暗号化された部分２２０のいずれかに任意の数又はタイプのフィールドを含んでもよい。

【００３０】

図３は、本開示の１つ以上の実施形態による、セキュアＡＰＩフローで使用される一例示的なＡＰＩリクエスト３００の一代替図を示す。ＡＰＩリクエスト３００は、ＡＰＩ管理サービスを通したＡＰＩコールルーティングの様々な部分間の論理的関係を示す。

【００３１】

ブロック３１０は、ＡＰＩ管理サービスを通してルーティングされるＡＰＩコールを示す。ブロック３１０から延びる矢印により示されるように、ＡＰＩリクエスト３００は、ＡＰＩリクエスト３００が最初にどこにルーティングされるべきかを識別するＡＰＩマネ

ージャＵＲＬ３１２を含んでもよい。さらに又は代わりに、ＡＰＩリクエスト３００は、ＡＰＩ管理サービスがＡＰＩリクエスト３００を取得するためにいずれのメソッドを利用し得るかを示すハイパーテキストトランスファープロトコル（ＨＴＴＰ）メソッド３１４を含んでもよい（例えば、リクエストは、ＰＯＳＴ、ＰＵＳＨ、ＧＥＴ等のＨＴＴＰメソッドを介して送信されてもよい）。例えば、ＰＵＳＨコマンドは、ＡＰＩ管理ＵＲＬ３１２におけるＡＰＩ管理サーバにＡＰＩリクエストを自動的にプッシュすることができる。いくつかの実施形態において、ＡＰＩリクエスト３００は、ＡＰＩ管理サービスの使用を承認する認証情報を含んでもよい。

【００３２】

破線ブロック３２０は、ネイティブＡＰＩコールの潜在的に暗号化された部分（例えば、ＡＰＩ管理システムに送信される要素のないＡＰＩコール）を表し得る。例えば、ネイティブＡＰＩコールは、ヘッダ３２１、クエリ３２２、及びボディ３２３を含んでもよい。ヘッダ３２１、クエリ３２２、及び／又はボディ３２３の何らかの組み合わせに含まれるＡＰＩリクエスト３００は、ＡＰＩ認証３２５（例えば、ＡＰＩプロバイダとの間のＡＰＩクライアントの認証中に使用される情報又はクレデンシャル）、１つ以上のリクエストパラメータ３２６（例えば、ＡＰＩに対する入力データ）、リクエストネイティブＡＰＩ　ＵＲＬ３２（例えば、呼び出されるＡＰＩにアクセスするためのＵＲＬ）、及び／又はネイティブＨＴＴＰメソッド３２８（例えば、ＡＰＩプロバイダがＡＰＩリクエスト３００を取得するためにいずれのメソッドを利用し得るか）を含んでもよい。

【００３３】

ＡＰＩプロバイダによるＡＰＩの実装及び／又はＡＰＩ管理サービスの実装に依存して、ネイティブＡＰＩリクエストの異なるデータコンポーネントが、ヘッダ３２１、クエリ３２２、及び／又はボディ３２３間の異なるコンポーネントに存在し得る。例えば、ヘッダ３２１は、リクエストネイティブＡＰＩ　ＵＲＬ３２７及びＡＰＩ認証を含んでもよく、クエリ３２２は、ネイティブＨＴＴＰメソッド３２８を含んでもよく、ボディ３２３は、リクエストパラメータ３２６を含んでもよい。これら及び他の実施形態において、ヘッダ３２１、クエリ３２２、及び／又はボディ３２３の任意の組み合わせが暗号化されてもよい。例えば、ヘッダ３２１が暗号化されなくてもよく、クエリ３２２及びボディ３２４が暗号化されてもよい。

【００３４】

本開示の範囲から逸脱することなく、ＡＰＩリクエスト３００に対して変更、追加、又は省略がなされてもよい。例えば、記載された方法における異なる要素の指定は、本明細書に記載される概念の説明を助けることが意図され、限定的ではない。さらに、ＡＰＩリクエスト３００は、任意数の他の要素を含んでもよく、あるいは記載されたものとは他のシステム又は文脈において実現されてもよい。例えば、ＡＰＩリクエスト３００は、図３に示されるものとは他のコンポーネントを含んでもよい。

【００３５】

図４、図５Ａ、及び図５Ｂは、本開示の１つ以上の実施形態による、セキュアＡＰＩフローの一例示的な実装の一例示的なスイムレーン図４００及び関連フローチャートの方法５００をそれぞれ示す。図４は、開発者４１０、ＡＰＩプロバイダ４２０、ＡＰＩ管理サーバ４３０、及びＡＰＩクライアント４１５間のいずれのエンティティが、図５Ａ及び図５Ｂに示される動作５０５～５９０に対応する（５０５～５９０とラベル付けされた）種々の動作を実行するかを示す。

【００３６】

スイムレーン図４００及び／又は方法５００の１つ以上の動作は、システム１００、開発者１１０、ＡＰＩプロバイダ１２０、ＡＰＩ管理サービス１３０、及び／又はエンドユーザ１５０などの、システム若しくはデバイス又はこれらの組み合わせにより実行され得る。別個のブロックとして示されているが、スイムレーン図４００及び／又は方法５００の種々のブロックは、所望の実装に依存してさらなるブロックに分割されてもよく、より少ないブロックに結合されてもよく、あるいは除去されてもよい。さらに、開発者４１０

、ＡＰＩプロバイダ４２０、ＡＰＩ管理サーバ４３０、及びＡＰＩクライアント４１５間の特定のエンティティが特定のタスクを実行するが、これらは単なる例であり、異なるエンティティが種々のアクション及び／又はさらなるアクションを実行してもよいことが理解される。

【００３７】

ブロック５０５において、開発者４１０が、ＡＰＩクライアント４１５により使用されるべきメッセージの暗号化のための公開及び秘密鍵を生成し得る。いくつかの場合に、秘密鍵は、ＡＰＩクライアント４１５などの特定のＡＰＩクライアントに固有でもよく、あるいは開発者により開発されたソフトウェアが使用されるすべての場合に使用される秘密鍵でもよい。これら及び他の実施形態において、秘密鍵及び公開鍵は、開発者４１０により開発されるソフトウェアプログラムにエンコードされてもよく、それにより、秘密鍵及び／又は公開鍵は、ソフトウェアの利用者にはアクセスできない。数学的に記載すると以下のとおりである。

$$KeyPair_{client}=(PK_c,\ PR_c)$$

$KeyPair_{client}$は、クライアントのために生成される鍵のペアに対応し、PK_cは、開発者４１０により生成される公開鍵であり、PR_cは、ＡＰＩクライアント４１５のために開発者４１０により生成される秘密鍵である。

【００３８】

ブロック５１０において、開発者４１０が、ＡＰＩへのアクセスを要求するリクエストをＡＰＩプロバイダ４２０にサブミットし得る。アクセスを要求する典型的なメッセージに追加で、開発者４１０は、ブロック５０５で生成された公開鍵（例えば、公開鍵PK_c）を含めてもよく、それにより、ＡＰＩプロバイダ４２０は、開発者４１０により生成されたＡＰＩクライアント４１５の公開鍵へのアクセスを有し得る。これら及び他の実施形態において、開発者４１０は、開発されたソフトウェアプログラムに含まれ得るアクセストークンを要求してもよく、それにより、ＡＰＩが呼び出されたとき、アクセストークンが利用され、かつ／あるいは、開発されたソフトウェアプログラムが実行しているときにＡＰＩプロバイダへのリクエストに含まれる。さらに又は代わりに、ユーザ名／パスワード、ＯＡｕｔｈ、ＯＡｕｔｈ２、ダイジェストなどの任意の他タイプのＡＰＩ認証アプローチが利用されてもよい。

【００３９】

ブロック５１５において、ＡＰＩプロバイダ４２０が、ＡＰＩプロバイダ４２０により使用されるべきメッセージの暗号化のための公開鍵及び秘密鍵を生成し得る。例えば、ＡＰＩプロバイダ４２０は、ＡＰＩクライアント４１５などの任意のＡＰＩクライアントとの間でメッセージを暗号化及び復号するために公開鍵及び秘密鍵のペアを利用してもよい。数学的に記載すると以下のとおりである。

$$KeyPair_{MG}=(PK_{MG},\ PR_{MG})$$

$KeyPair_{MG}$は、ＡＰＩプロバイダ４２０により生成される鍵のペアに対応し、PK_{MG}は、ＡＰＩプロバイダ４２０により生成される公開鍵であり、PR_{MG}は、ＡＰＩプロバイダ４２０により生成される秘密鍵である。

【００４０】

ブロック５２０において、ＡＰＩプロバイダ４２０が、開発者４１０にＡＰＩへのアクセスを付与し得、ブロック５１５で生成されたＡＰＩプロバイダ公開鍵を開発者４１０に提供し得る。例えば、ＡＰＩプロバイダ４２０は、ＡＰＩプロバイダ４２０の公開鍵（PK_{MG}）と、ＡＰＩにアクセスするとき開発者４１０及び／又はＡＰＩクライアント４１５により使用されるべき任意の認証情報（例えば、ＡＰＩ認証データ構造）とを含むメッセージを開発者４１０に送信してもよい。

【００４１】

ブロック５２５において、ＡＰＩ管理サーバへの登録が生じ得る。図４に示すように、動作５２５ａにおいて、開発者４１０は、ＡＰＩ管理サーバ４３０に登録することができ、ＡＰＩ管理サーバ４３０は、ＡＰＩ管理サーバ４３０への登録の認可／肯定応答（ackn

owledgment）を提供することができる。さらに又は代わりに、動作５２５ｂにおいて、ＡＰＩプロバイダ４２０は、ＡＰＩ管理サーバ４３０に登録することができ、ＡＰＩ管理サーバ４３０は、ＡＰＩ管理サーバ４３０への登録の対応する認可／肯定応答を提供することができる。これら及び他の実施形態において、登録は、登録当事者を認証する情報、例えば、サブスクリプション名、サブスクリプション識別子、プライマリキー、トークン、開発者４１０／ＡＰＩプロバイダ４２０のユーザ名（例えば、電子メールアドレス）などを含んでもよい。いくつかの実施形態において、動作５２５ａ及び５２５ｂの双方が実行されてもよく、それにより、ＡＰＩプロバイダ４２０及び開発者４１０は双方、ＡＰＩ管理サーバ４３０に登録でき、これは、同じＡＰＩ管理サービス又は区別可能なＡＰＩ管理サービスを含んでもよい。

【００４２】

ブロック５３０において、開発者４１０が、ＡＰＩアクセス情報及び暗号化情報をＡＰＩクライアントに提供し得る。例えば、開発者４１０は、ＡＰＩ認証情報及びＡＰＩプロバイダ４２０の公開鍵を組み込んだソフトウェアを配布してもよく、それにより、ソフトウェアは、ＡＰＩプロバイダ４２０へのＡＰＩリクエストの部分を暗号化することができる。別の例として、ソフトウェアは、開発者４１０により生成された秘密鍵をエンコードしてもよく、それにより、ソフトウェアは、ＡＰＩプロバイダ４２０からのメッセージを復号することができる。さらなる例として、開発者４１０は、ＡＰＩアクセス情報（例えば、ＡＰＩ認証トークンなど）及び暗号化情報（例えば、秘密鍵PR_C及び公開鍵PK_{MG}）を、開発されたソフトウェアとは別個の区別可能なコンポーネント又はクレデンシャルとして提供してもよい。そのような情報、又は開発者４１０に固有及び／又は開発者４１０により使用される他の情報は、開発者データとして特徴づけられてもよい。

【００４３】

ブロック５３５において、ＡＰＩクライアント４１５が、ＡＰＩプロバイダ４２０のＡＰＩに対する入力データを取得し得る。例えば、配布されたソフトウェアが動作しているとき、それがユーザ入力を要求してもよく、あるいはＡＰＩ管理サーバ４３０により管理されるＡＰＩプロバイダ４２０のＡＰＩへの入力としてサブミットされるべき値又は他のデータを生成又はその他の方法で取得してもよい。

【００４４】

ブロック５４０において、ブロック５３５の入力データが、ＡＰＩプロバイダ４２０の公開鍵を使用して暗号化され得る。例えば、配布されたソフトウェアは、入力データ及び／又はＡＰＩリクエストの他のコンポーネントをPK_{MG}を使用して暗号化することができる。さらに、開発者データ（例えば、ブロック５３０で配布されたソフトウェアで提供されるものなどの、ＡＰＩへのアクセスを許可するための認証情報）が暗号化され、リクエスト／リクエストの暗号化された部分の一部に含まれてもよい。数学的に記述すると、ＡＰＩクライアント４１５（開発者４１０により配布されたソフトウェアを介したＡＰＩクライアント４１５を含む）は、以下の演算を実行することができる。

$$ENC(Req, PK_{MG}), \&\ MG\ API\ Auth$$

ＥＮＣは、暗号化鍵（PK_{MG}）を使用してコンテンツ（Ｒｅｑ）を暗号化する機能を含むことができ、Ｒｅｑは、暗号化されるべきブロック５３５の入力データ及び／又はＡＰＩリクエストの他のデータを表すことができ、ＭＧ　ＡＰＩ　Ａｕｔｈは、ブロック５２５で提供される情報などの、ＡＰＩ管理サーバ４３０にアクセスするために使用される認証情報を表すことができる。これら及び他の実施形態において、ＡＰＩリクエストは、暗号化された部分及び暗号化されていない部分の双方を含んでもよい。

【００４５】

ブロック５４５において、暗号化された入力データを有するＡＰＩリクエストが、ＡＰＩ管理サーバ４３０に送信され得る。例えば、ＡＰＩクライアント４１５は、ＡＰＩリクエストをＡＰＩ管理サーバ４３０に送信してＡＰＩリクエストに対する管理サービスを実行することができる。

【００４６】

ブロック５５０において、ＡＰＩ管理サーバ４３０が、ＡＰＩリクエストの暗号化されていない部分に基づいて、ＡＰＩリクエストに対するＡＰＩ管理サービスを実行し得る。例えば、ＡＰＩ管理サーバ４３０は、トラフィック制御を実行し、管理ポリシーを実施し（enforce）、セキュリティポリシーを実施し、リクエスト数に制限を課すなどが可能である。いくつかの実施形態において、ＡＰＩ管理サーバ４３０は、ＡＰＩリクエストを破棄してもよく（例えば、割り当て量を超えた場合）、ＡＰＩリクエストを遅延又は延期させてもよく、１つのリクエストを他のものより上に優先順位付けてもよい。これら及び他の実施形態において、このようなタスクは、暗号化されていないＡＰＩリクエストのメタデータに基づいて実行されてもよい。

【００４７】

ブロック５５５において、ＡＰＩ管理サーバ４３０が、ＡＰＩリクエストをＡＰＩプロバイダ４２０に転送し得る。

【００４８】

ブロック５６０において、ＡＰＩプロバイダ４２０が、ＡＰＩプロバイダ４２０の秘密鍵（例えば、PR_{MG}）を使用してＡＰＩリクエストを復号し得る。いくつかの実施形態において、ＡＰＩリクエストが復号される前に、ＡＰＩリクエストに対して実行される予備的承認があってもよい。復号は、数学的に以下のように記述され得る。

$$DEC(Req, PR_{MG})$$

ＤＥＣは、鍵PR_{MG}を使用したコンテンツＲｅｑの復号アルゴリズムを表すことができる。

【００４９】

ブロック５６５において、ＡＰＩが、ブロック５３５で取得された入力データを使用して呼び出され、出力を生成し得る。

【００５０】

ブロック５７０において、ＡＰＩプロバイダ４２０が、ブロック５１０で開発者４１０により提供された公開鍵を使用して、ＡＰＩの呼び出しからのレスポンス（例えば、ＡＰＩの出力）を暗号化し得る。例えば、数学的に記述されると以下のとおりである。

$$ENC(Res, PK_{C})$$

【００５１】

ブロック５７５において、ブロック５７０からの暗号化されたレスポンスデータを有するＡＰＩレスポンスが、ＡＰＩプロバイダ４２０からＡＰＩ管理サーバ４３０に送信され得る。これら及び他の実施形態において、ＡＰＩレスポンスは、暗号化された部分（例えば、ＡＰＩにより生成されたレスポンス）と暗号化されていない部分（例えば、レスポンスがどこに送られるべきかの識別情報）とを含んでもよい。

【００５２】

ブロック５８０において、ＡＰＩ管理サーバ４３０が、ブロック５７５で送信されたＡＰＩレスポンスの暗号化されていない部分に基づいて管理サービスを実行し得る。

【００５３】

ブロック５８５において、ＡＰＩレスポンスが、ＡＰＩ管理サーバ４３０からＡＰＩクライアント４１５へ転送され得る。

【００５４】

ブロック５９０において、ＡＰＩレスポンスが、秘密鍵PR_{C}を使用してＡＰＩクライアント４１５により復号され得る。数学的に記述すると、ＡＰＩクライアント４１５は以下の演算を実行することができる。

$$DEC(Req, PR_{C})$$

【００５５】

このようなアプローチを使用することにより、ＡＰＩ管理サーバ４３０は、リクエスト及びレスポンスの実際の内容へのアクセスを有することなく、ＡＰＩリクエスト及びレスポンスに対する管理サービスを実行し続けることができる。

【００５６】

本開示の範囲から逸脱することなく、スイムレーン図４００及び／又は方法５００に対して変更、追加、又は省略がなされてもよい。例えば、スイムレーン図４００及び／又は方法５００の動作は異なる順序で実現されてもよい。さらに又は代わりに、２つ以上の動作が同時に実行されてもよい。さらに、概説された動作及びアクションは例として提供されているに過ぎず、動作及びアクションのいくつかは、開示される実施形態の本質を損なうことなく、任意であってもよく、より少ない動作及びアクションに結合されてもよく、あるいはさらなる動作及びアクションへ拡張されてもよい。

【００５７】

図６は、本開示に記載される少なくとも１つの実施形態による、一例示的なコンピューティングシステム６００を示す。コンピューティングシステム６００は、プロセッサ６１０、メモリ６２０、データ記憶装置６３０、及び／又は通信ユニット６４０を含んでもよく、これらはすべて通信上結合されてもよい。図１のシステム１００の一部又は全部が、開発者１１０、ＡＰＩプロバイダ１２０、ＡＰＩ管理サービス１３０、及び／又はエンドユーザデバイス１５０を含む、コンピューティングシステム６００と一貫性のあるコンピューティングシステムとして実現されてもよい。

【００５８】

一般に、プロセッサ６１０は、種々のコンピュータハードウェア又はソフトウェアモジュールを含む、任意の適切な専用若しくは汎用コンピュータ、コンピューティングエンティティ、又は処理デバイスを含んでもよく、任意の適用可能なコンピュータ読取可能記憶媒体に記憶された命令を実行するように構成されてもよい。例えば、プロセッサ６１０は、マイクロプロセッサ、マイクロコントローラ、デジタル信号プロセッサ（ＤＳＰ）、特定用途向け集積回路（ＡＳＩＣ）、フィールドプログラマブルゲートアレイ（ＦＰＧＡ）、又はプログラム命令を解釈及び／又は実行するよう及び／又はデータを処理するように構成された任意の他のデジタル若しくはアナログ回路を含んでもよい。

【００５９】

図６において単一のプロセッサとして示されているが、プロセッサ６１０は、本開示に記載の任意数の動作を個々又は集合的に実行するように構成された、任意数のネットワーク又は物理位置にわたり分散された任意数のプロセッサを含んでもよい。いくつかの実施形態において、プロセッサ６１０は、メモリ６２０、データ記憶装置６３０、又はメモリ６２０及びデータ記憶装置６３０に記憶されたプログラム命令を解釈及び／又は実行し、かつ／あるいはデータを処理してもよい。いくつかの実施形態において、プロセッサ６１０は、データ記憶装置６３０からプログラム命令を取り出し、プログラム命令をメモリ６２０にロードしてもよい。

【００６０】

プログラム命令がメモリ６２０にロードされた後、プロセッサ６１０は、プログラム命令、例えば、図４及び図５のスイムレーン図４００及び／又は方法５００にそれぞれ示される動作のうち任意のものを実行する命令などを実行することができる。例えば、プロセッサ６１０は、ＡＰＩリクエストへの入力データを取得することができ、そのデータをＡＰＩプロバイダの公開鍵を使用して暗号化することができる。

【００６１】

メモリ６２０及びデータ記憶装置６３０は、記憶されたコンピュータ実行可能命令又はデータ構造を搬送し又は有するコンピュータ読取可能記憶媒体又は１つ以上のコンピュータ読取可能記憶媒体を含むことができる。そのようなコンピュータ読取可能記憶媒体は、プロセッサ６１０などの汎用又は専用コンピュータによりアクセスされ得る任意の利用可能な媒体でもよい。例えば、メモリ６２０及び／又はデータ記憶装置６３０は、１つ以上の公開又は秘密鍵を記憶することができる。いくつかの実施形態において、コンピューティングシステム６００は、メモリ６２０及びデータ記憶装置６３０のいずれかを含んでもよく、あるいは含まなくてもよい。

【００６２】

限定でなく例として、そのようなコンピュータ読取可能記憶媒体は、ランダムアクセス

メモリ（ＲＡＭ）、読取専用メモリ（ＲＯＭ）、電気的消去可能プログラマブル読取専用メモリ（ＥＥＰＲＯＭ）、コンパクトディスク読取専用メモリ（ＣＤ‐ＲＯＭ）若しくは他の光ディスク記憶装置、磁気ディスク記憶装置若しくは他の磁気記憶デバイス、フラッシュメモリデバイス（例えば、ソリッドステートメモリデバイス）、又はコンピュータ実行可能命令又はデータ構造の形式で所望のプログラムコードを搬送又は記憶するために使用でき、かつ汎用又は専用コンピュータによりアクセスできる任意の他の記憶媒体を含む、非一時的なコンピュータ読取可能記憶媒体を含んでもよい。上記の組み合わせもまた、コンピュータ読取可能記憶媒体の範囲内に含まれてもよい。コンピュータ実行可能命令は、例えば、プロセッサ６１０に特定の動作又は動作のグループを実行させるように構成された命令及びデータを含んでもよい。

【００６３】

通信ユニット６４０は、ネットワークを介して情報を送信又は受信するように構成された任意のコンポーネント、デバイス、システム、又はこれらの組み合わせを含んでもよい。いくつかの実施形態において、通信ユニット６４０は、他の場所、同じ場所における他のデバイス、又はさらには同じシステム内の他のコンポーネントと通信してもよい。例えば、通信ユニット６４０は、モデム、ネットワークカード（無線又は有線）、光通信装置、赤外線通信装置、無線通信装置（アンテナなど）、及び／又はチップセット（Ｂｌｕｅｔｏｏｔｈ（登録商標）装置、８０２．６装置（メトロポリタンエリアネットワーク（ＭＡＮ）など）、ＷｉＦｉ装置、ＷｉＭａｘ（登録商標）装置、セルラー通信設備等）などを含んでもよい。通信ユニット６４０は、ネットワーク及び／又は本開示に記載される任意の他のデバイス又はシステムとの間でデータが交換されることを可能にし得る。例えば、通信ユニット６４０は、システム６００が、コンピューティングデバイス及び／又は他のネットワークなどの他のシステムと通信することを可能にし得る。

【００６４】

当業者は本開示を検討した後、本開示の範囲から逸脱することなくシステム６００に対して変更、追加、又は省略がなされ得ることを認識し得る。例えば、システム６００は、明示的に例示及び記載されたものより多くの又は少ないコンポーネントを含んでもよい。

【００６５】

前述の開示は、開示された正確な形式又は特定の分野に本開示を限定することは意図されない。したがって、本明細書に明示的に記載されているか又は暗示されているかにかかわらず、本開示に対する種々の代替実施形態及び／又は修正が本開示に照らして可能なことが企図される。このように本開示の実施形態を説明したが、本開示の範囲から逸脱することなく形式及び詳細において変更がなされ得ることが認識され得る。したがって、本開示は、特許請求の範囲によってのみ限定される。

【００６６】

いくつかの実施形態において、本開示に記載される異なるコンポーネント、モジュール、エンジン、及びサービスが、コンピューティングシステム上で実行するオブジェクト又はプロセスとして（例えば、別個のスレッドとして）実現されてもよい。本明細書に記載されるシステム及びプロセスのいくつかは、一般に、（汎用ハードウェアに記憶され、及び／又は汎用ハードウェアにより実行される）ソフトウェアで実現されるものとして記載されるが、特定のハードウェア実装、又はソフトウェアと特定のハードウェア実装との組み合わせもまた可能であり、企図される。

【００６７】

本明細書において、特に別記の特許請求の範囲（例えば、別記の特許請求の範囲の本文）において用いられる用語は、一般に「開放的」な用語として意図されている（例えば、用語「含んでいる」は、「含んでいるがこれに限定されない」と解釈されるべきであり、用語「有する」は、「少なくとも有する」と解釈されるべきであり、用語「含む」は、「含むがこれに限定されない」と解釈されるべきである、等）。

【００６８】

さらに、特定数の導入された請求項記載が意図されている場合、そのような意図は請求

項に明示的に記載され、そのような記載がない場合、そのような意図は存在しない。例えば、理解の助けとして、以下の別記の特許請求の範囲は、請求項記載を導入するために、導入フレーズ「少なくとも１つの」及び「１つ以上の」の使用を含むことがある。しかしながら、そのようなフレーズの使用は、不定冠詞「一の」（"a"又は"an"）による請求項記載の導入が、同じ請求項が導入フレーズ「１つ以上の」又は「少なくとも１つの」と「一の」などの不定冠詞とを含むときでも、そのような導入された請求項記載を含む任意の特定の請求項を１つのそのような記載のみ含む実施形態に限定することを暗に示すように見なされるべきではない（例えば、「一の」（"a"及び／又は"an"）は「少なくとも１つの」又は「１つ以上の」を意味するよう解釈されるべきである）。請求項記載を導入するために用いられる定冠詞の使用についても同様である。

【００６９】

さらに、特定数の導入された請求項記載が明示的に記載されている場合であっても、当業者は、そのような記載は少なくとも記載された数を意味するよう解釈されるべきであることを認識するであろう（例えば、他の修飾語を伴わない「２つの記載」というただそれだけの記載は、少なくとも２つの記載、又は２つ以上の記載を意味する）。さらに、「Ａ、Ｂ、及びＣ等のうち少なくとも１つ」又は「Ａ、Ｂ、及びＣ等のうち１つ以上」と類似の規定が用いられている例において、一般に、そのような構造は、Ａ単独、Ｂ単独、Ｃ単独、Ａ及びＢ共に、Ａ及びＣ共に、Ｂ及びＣ共に、又はＡ、Ｂ、及びＣ共に等を含むことが意図される。例えば、用語「及び／又は」の使用は、このようにみなされることが意図される。

【００７０】

さらに、明細書においてか、特許請求の範囲においてか、又は図面においてかにかかわらず、２つ以上の代替的な用語を提示するいかなる分離的なワード又はフレーズも、用語のうち１つ、用語のうちいずれか、又は双方の用語を含む可能性を考慮するよう理解されるべきである。例えば、フレーズ「Ａ又はＢ」は、「Ａ」又は「Ｂ」又は「Ａ及びＢ」の可能性を含むよう理解されるべきである。

【００７１】

しかしながら、そのようなフレーズの使用は、不定冠詞「一の」（"a"又は"an"）による請求項記載の導入が、同じ請求項が導入フレーズ「１つ以上の」又は「少なくとも１つの」と「一の」などの不定冠詞とを含むときでも、そのような導入された請求項記載を含む任意の特定の請求項を１つのそのような記載のみ含む実施形態に限定することを暗に示すように見なされるべきではない（例えば、「一の」（"a"及び／又は"an"）は「少なくとも１つの」又は「１つ以上の」を意味するよう解釈されるべきである）。請求項記載を導入するために用いられる定冠詞の使用についても同様である。

【００７２】

さらに、用語「第１」、「第２」、「第３」等の使用は、本明細書において必ずしも特定の順序を含意するために使用されるものではない。一般に、用語「第１」、「第２」、「第３」等は、異なる要素間で区別するために使用される。用語「第１」、「第２」、「第３」等が特定の順序を含意することの具体的な提示なしでは、これらの用語は特定の順序を含意するよう理解されるべきではない。

【００７３】

本明細書に記載される全ての例及び条件付き言語は、本発明及び発明者が当該技術分野を促進するために寄与した概念を理解する際に読者を助けるための教育的目的が意図され、このように具体的に記載された例及び条件に限定されないものとみなされるべきである。本開示の実施形態が詳細に説明されたが、本開示の主旨及び範囲から逸脱することなく種々の変更、置換、及び改変をこれに行えることを理解されたい。

【００７４】

開示された実施形態の前の説明は、当業者が本開示を製造又は使用することができるように提供される。これらの実施形態に対する種々の修正は、当業者には容易に明らかであり、本明細書で定義される一般的原理は、本開示の主旨又は範囲から逸脱することなく他

の実施形態に適用され得る。したがって、本開示は、本明細書に示される実施形態に限定されることは意図されず、本明細書に開示された原理及び新規の特徴と一致する最も広い範囲を与えられるべきである。

【００７５】

上記の実施形態につき以下の付記を残しておく。

（付記１）

アプリケーションプログラミングインターフェース（ＡＰＩ）に対する入力データを取得するステップと、

前記ＡＰＩに対する前記入力データを前記ＡＰＩのプロバイダの公開鍵を使用して暗号化するステップと、

前記ＡＰＩを呼び出すＡＰＩリクエストをＡＰＩ管理サーバに送信するステップであり、前記ＡＰＩリクエストは、前記ＡＰＩに対するＡＰＩコールと前記暗号化された入力データとを含み、前記ＡＰＩリクエストは、前記ＡＰＩ管理サーバが前記ＡＰＩコールに基づきＡＰＩ管理サービスを実行できるが前記暗号化された入力データを前記公開鍵で復号できないフォーマットである、ステップと、

を含む方法。

（付記２）

前記ＡＰＩ管理サーバから前記ＡＰＩリクエストに対するレスポンスを受信するステップであり、前記ＡＰＩリクエストに対する前記レスポンスは、前記入力データに基づく前記ＡＰＩの出力を含み、前記出力は、クライアントの公開鍵を使用して前記ＡＰＩプロバイダにより暗号化される、ステップをさらに含む付記１に記載の方法。

（付記３）

前記ＡＰＩリクエストに対する前記レスポンスを前記クライアントの秘密鍵を使用して復号するステップをさらに含む付記２に記載の方法。

（付記４）

前記ＡＰＩリクエストは、クライアント識別子とＡＰＩプロバイダ識別子とＡＰＩ管理承認構造とを含む暗号化されていないデータコンポーネントをさらに含む、付記１に記載の方法。

（付記５）

開発者データを前記ＡＰＩリクエストの一部であるように暗号化するステップであり、前記開発者データは、前記入力データを取得する前に開発者により提供される、ステップをさらに含む付記１に記載の方法。

（付記６）

前記プロバイダの前記公開鍵は、前記ＡＰＩリクエストを生成するコンピュータプログラム内に埋め込まれ、それにより、前記入力データの前記暗号化はユーザ入力なしに生じる、付記１に記載の方法。

（付記７）

前記ＡＰＩ管理サービスは、トラフィック制御、管理ポリシー実施、及びセキュリティポリシー実施のうち少なくとも１つを含む、付記１に記載の方法。

（付記８）

１つ以上のプロセッサにより実行されることに応答してシステムに動作を実行させる命令を含む１つ以上の非一時的コンピュータ読取可能媒体であって、前記動作は、

アプリケーションプログラミングインターフェース（ＡＰＩ）に対する入力データを取得することと、

前記ＡＰＩに対する前記入力データを前記ＡＰＩのプロバイダの公開鍵を使用して暗号化することと、

通信コンポーネントに、前記ＡＰＩを呼び出すＡＰＩリクエストをＡＰＩ管理サーバに送信するように指示することであり、前記ＡＰＩリクエストは、前記ＡＰＩに対するＡＰＩコールと前記暗号化された入力データとを含み、前記ＡＰＩリクエストは、前記ＡＰＩ管理サーバが前記ＡＰＩコールに基づきＡＰＩ管理サービスを実行できるが前記暗号化さ

れた入力データを前記公開鍵で復号できないフォーマットである、ことと、
を含む、１つ以上のコンピュータ読取可能媒体。
（付記９）
前記動作は、前記ＡＰＩ管理サーバから前記ＡＰＩリクエストに対するレスポンスを受信することであり、前記ＡＰＩリクエストに対する前記レスポンスは、前記入力データに基づく前記ＡＰＩの出力を含み、前記出力は、クライアントの公開鍵を使用して前記ＡＰＩプロバイダにより暗号化される、ことをさらに含む、付記８に記載の１つ以上のコンピュータ読取可能媒体。
（付記１０）
前記動作は、前記ＡＰＩリクエストに対する前記レスポンスを前記クライアントの秘密鍵を使用して復号することをさらに含む、付記９に記載の１つ以上のコンピュータ読取可能媒体。
（付記１１）
前記ＡＰＩリクエストは、クライアント識別子とＡＰＩプロバイダ識別子とＡＰＩ管理承認構造とを含む暗号化されていないデータコンポーネントを含む、付記８に記載の１つ以上のコンピュータ読取可能媒体。
（付記１２）
前記動作は、クライアントデータを前記ＡＰＩリクエストの一部であるように暗号化することをさらに含む、付記８に記載の１つ以上のコンピュータ読取可能媒体。
（付記１３）
前記プロバイダの前記公開鍵は、前記ＡＰＩリクエストを生成するコンピュータプログラム内に埋め込まれ、それにより、前記入力データの前記暗号化はユーザ入力なしに生じる、付記８に記載の１つ以上のコンピュータ読取可能媒体。
（付記１４）
前記ＡＰＩ管理サービスは、トラフィック制御、管理ポリシー実施、及びセキュリティポリシー実施のうち少なくとも１つを含む、付記８に記載の１つ以上のコンピュータ読取可能媒体。
（付記１５）
システムであって、
１つ以上のプロセッサと、
前記１つ以上のプロセッサにより実行されることに応答して当該システムに動作を実行させる命令を含む１つ以上の非一時的コンピュータ読取可能媒体と、
を含み、前記動作は、
アプリケーションプログラミングインターフェース（ＡＰＩ）に対する入力データを取得することと、
前記ＡＰＩに対する前記入力データを前記ＡＰＩのプロバイダの公開鍵を使用して暗号化することと、
通信コンポーネントに、前記ＡＰＩを呼び出すＡＰＩリクエストをＡＰＩ管理サーバに送信するように指示することであり、前記ＡＰＩリクエストは、前記ＡＰＩに対するＡＰＩコールと前記暗号化された入力データとを含み、前記ＡＰＩリクエストは、前記ＡＰＩ管理サーバが前記ＡＰＩコールに基づきＡＰＩ管理サービスを実行できるが前記暗号化された入力データを前記公開鍵で復号できないフォーマットである、ことと、
を含む、システム。
（付記１６）
前記動作は、前記ＡＰＩ管理サーバから前記ＡＰＩリクエストに対するレスポンスを受信することであり、前記ＡＰＩリクエストに対する前記レスポンスは、前記入力データに基づく前記ＡＰＩの出力を含み、前記出力は、クライアントの公開鍵を使用して前記ＡＰＩプロバイダにより暗号化される、ことをさらに含む、付記１５に記載のシステム。
（付記１７）
前記動作は、前記ＡＰＩリクエストに対する前記レスポンスを前記クライアントの秘密

鍵を使用して復号することをさらに含む、付記１６に記載のシステム。

（付記１８）

前記ＡＰＩリクエストは、クライアント識別子とＡＰＩプロバイダ識別子とＡＰＩ管理承認構造とを含む暗号化されていないデータコンポーネントをさらに含む、付記１５に記載のシステム。

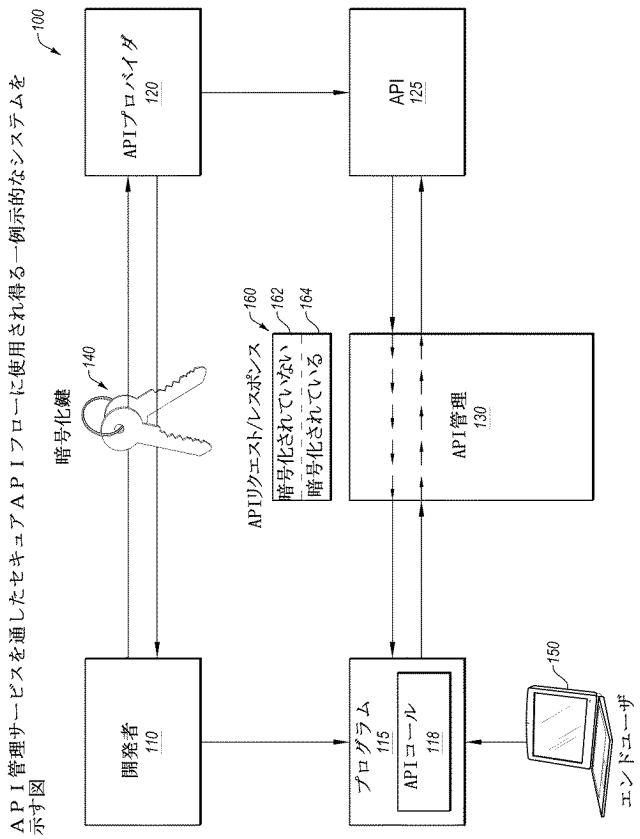
（付記１９）

前記動作は、開発者データを前記ＡＰＩリクエストの一部であるように暗号化することであり、前記開発者データは、前記入力データを取得する前に開発者により提供される、ことをさらに含む、付記１５に記載のシステム。

（付記２０）

前記プロバイダの前記公開鍵は、前記ＡＰＩリクエストを生成するコンピュータプログラム内に埋め込まれ、それにより、前記入力データの前記暗号化はユーザ入力なしに生じる、付記１５に記載のシステム。

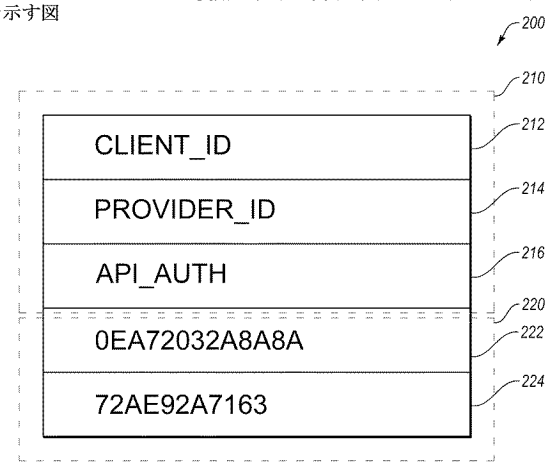
【図１】



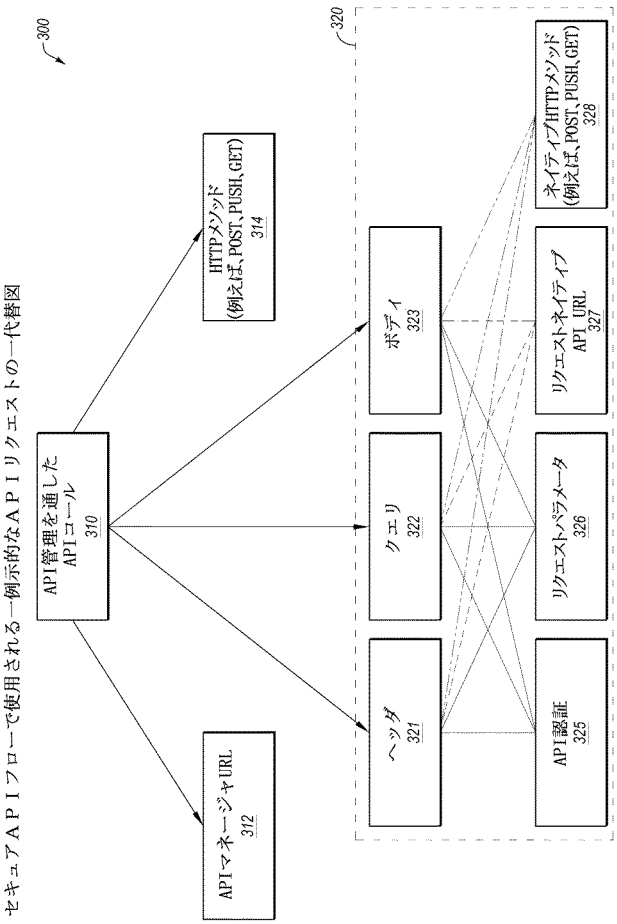
ＡＰＩ管理サービスを通したセキュアＡＰＩフローに使用され得る一例示的なシステムを示す図

【図２】

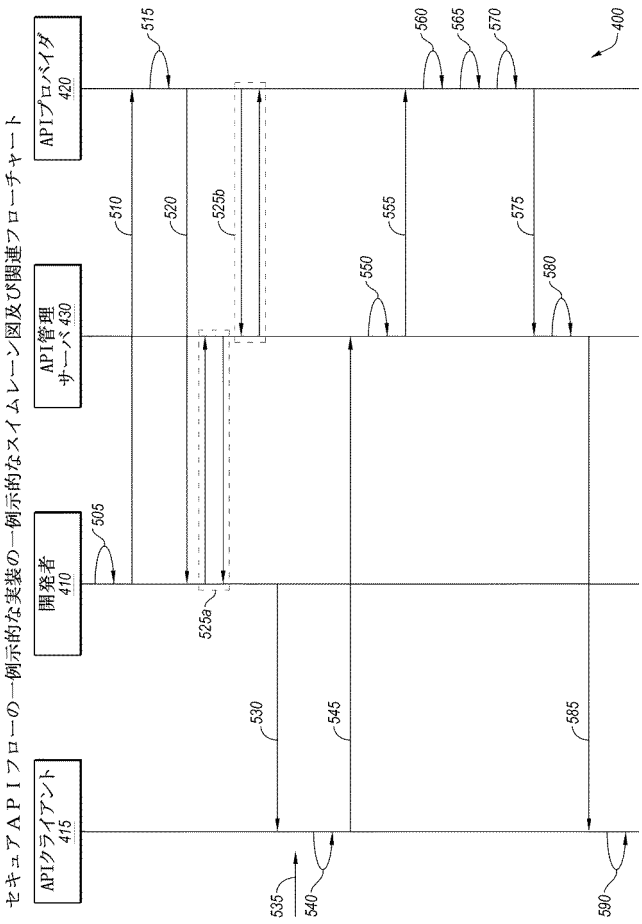
セキュアＡＰＩフローで使用される一例示的なＡＰＩリクエストを示す図



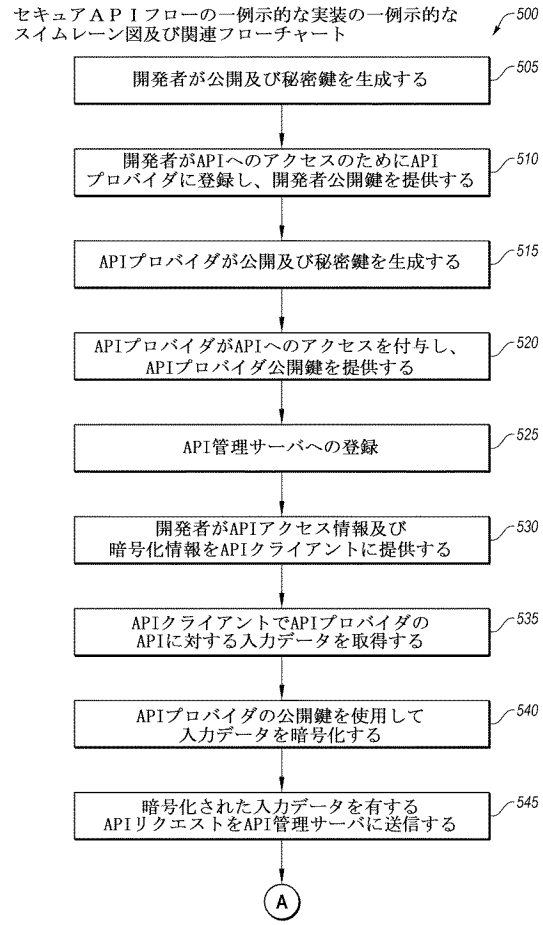
【図３】


セキュアＡＰＩフローで使用される一例示的なＡＰＩリクエストの一代替図
300
320
API管理を通したAPIコール 310
APIマネージャURL 312
HTTPメソッド (例えば、POST、PUSH、GET) 314
ボディ 323
クエリ 322
ヘッダ 321
ネイティブHTTPメソッド (例えば、POST、PUSH、GET) 328
リクエストネイティブAPI URL 327
リクエストパラメータ 326
API認証 325

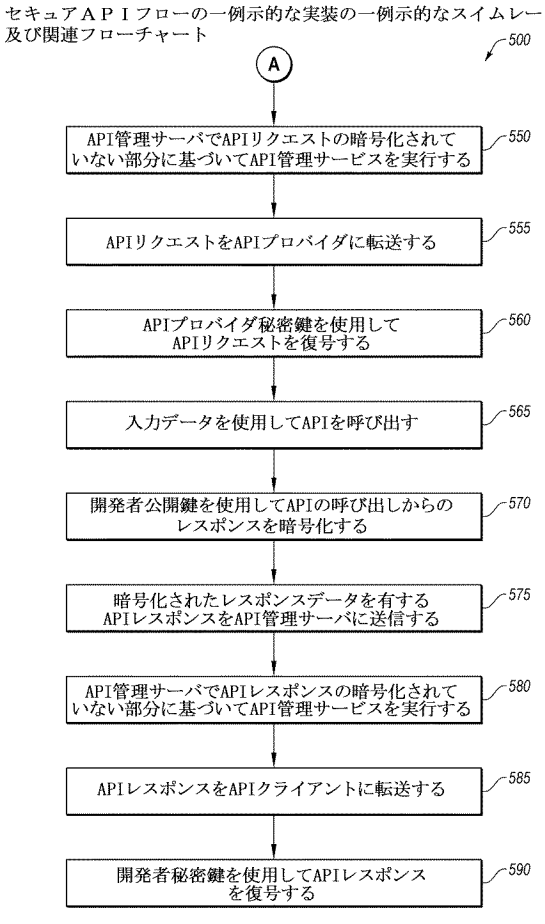
【図４】


セキュアＡＰＩフローの一例示的な実装の一例示的なスイムレーン図及び関連フローチャート
400
APIプロバイダ 420
API管理サーバ 430
開発者 410
APIクライアント 415
505
510
515
520
525a
525b
530
535
540
545
550
555
560
565
570
575
580
585
590

【図５Ａ】

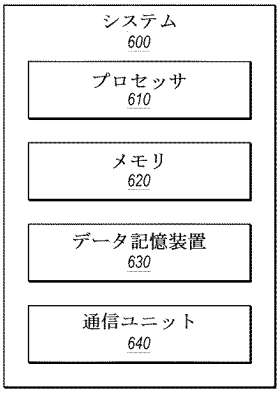

セキュアＡＰＩフローの一例示的な実装の一例示的なスイムレーン図及び関連フローチャート
500
開発者が公開及び秘密鍵を生成する 505
開発者がAPIへのアクセスのためにAPIプロバイダに登録し、開発者公開鍵を提供する 510
APIプロバイダが公開及び秘密鍵を生成する 515
APIプロバイダがAPIへのアクセスを付与し、APIプロバイダ公開鍵を提供する 520
API管理サーバへの登録 525
開発者がAPIアクセス情報及び暗号化情報をAPIクライアントに提供する 530
APIクライアントでAPIプロバイダのAPIに対する入力データを取得する 535
APIプロバイダの公開鍵を使用して入力データを暗号化する 540
暗号化された入力データを有するAPIリクエストをAPI管理サーバに送信する 545
A

【図５Ｂ】


セキュアＡＰＩフローの一例示的な実装の一例示的なスイムレーン図及び関連フローチャート
500
A
API管理サーバでAPIリクエストの暗号化されていない部分に基づいてAPI管理サービスを実行する 550
APIリクエストをAPIプロバイダに転送する 555
APIプロバイダ秘密鍵を使用してAPIリクエストを復号する 560
入力データを使用してAPIを呼び出す 565
開発者公開鍵を使用してAPIの呼び出しからのレスポンスを暗号化する 570
暗号化されたレスポンスデータを有するAPIレスポンスをAPI管理サーバに送信する 575
API管理サーバでAPIレスポンスの暗号化されていない部分に基づいてAPI管理サービスを実行する 580
APIレスポンスをAPIクライアントに転送する 585
開発者秘密鍵を使用してAPIレスポンスを復号する 590

【図６】

一例示的なコンピューティングシステムを示す図



フロントページの続き

(72)発明者　チェン・ウェイ－ペン
アメリカ合衆国，カリフォルニア州　９４０８５，サニーヴェイル，イースト　アークス　アヴェニュ　１２４０番　フジツウ　ラボラトリーズ　アメリカ内