(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6334920号
(P6334920)
(45)発行日　平成30年5月30日(2018.5.30)　(24)登録日　平成30年5月11日(2018.5.11)

(51)Int.Cl.　F I
G06Q 50/10 (2012.01)　G06Q 50/10
G06F 21/44 (2013.01)　G06F 21/44

請求項の数 13　(全 18 頁)

(21)出願番号　特願2014-1241(P2014-1241)
(22)出願日　平成26年1月7日(2014.1.7)
(65)公開番号　特開2015-130073(P2015-130073A)
(43)公開日　平成27年7月16日(2015.7.16)
審査請求日　平成29年1月5日(2017.1.5)

(73)特許権者　000001007
キヤノン株式会社
東京都大田区下丸子３丁目３０番２号
(74)代理人　100076428
弁理士　大塚　康徳
(74)代理人　100112508
弁理士　高柳　司郎
(74)代理人　100115071
弁理士　大塚　康弘
(74)代理人　100116894
弁理士　木村　秀二
(74)代理人　100130409
弁理士　下山　治
(74)代理人　100134175
弁理士　永川　行光

最終頁に続く

(54)【発明の名称】権限管理サーバー及び権限管理方法

(57)【特許請求の範囲】
【請求項１】
登録されたクライアントからの、リソースに対するユーザーのアクセス権限の委譲を要求する認可要求に応じて、前記認可要求を検証し、検証が成功した場合に前記クライアントに対して認可トークンを発行する発行手段と、
リソース要求が前記認可トークンとともにあった場合、前記認可トークンを検証し、検証が成功した場合に前記リソースに対するアクセスを許可する検証手段と、を有し、
前記検証手段は、前記認可トークンの正当性を検証するとともに、前記リソースに対するアクセス数が前記リソース要求を行ったクライアントに対して設定されているアクセス上限数を超えたか否かも検証し、前記認可トークンが正当であり、かつ前記リソースに対するアクセス数が前記アクセス上限数を超えない場合に、前記認可トークンの検証が成功したと判定されることを含むことを特徴とする権限管理サーバー。
【請求項２】
前記検証手段における検証の処理の他に、前記発行手段が前記認可要求を検証する際にも、前記発行手段が前記リソースに対するアクセス数が前記リソース要求を行ったクライアントに対して設定されているアクセス上限数を超えたか否かを検証し、前記発行手段は、前記認可要求が正当であり、かつ前記リソースに対するアクセス数が前記アクセス上限数を超えない場合に、前記認可トークンを発行することを特徴とする請求項１に記載の権限管理サーバー。
【請求項３】

前記アクセス上限数は、単位期間あたりのアクセス上限数であることを特徴とする請求項１又は２に記載の権限管理サーバー。
【請求項４】
クライアントあたりの前記アクセス上限数を入力するための入力画面を第１の端末に表示させて、前記入力画面に対して入力された値を前記アクセス上限数として設定する設定手段を更に有することを特徴とする請求項１乃至３のいずれか一項に記載の権限管理サーバー。
【請求項５】
前記入力画面にはさらに、前記アクセス上限数に加算できる上限加算値を入力するための入力欄が含まれ、前記設定手段は、前記入力画面において入力された値を設定し、
前記リソースに対するアクセス数が前記アクセス上限数に到達したクライアントについては、前記クライアントからの要求に応じて、前記アクセス上限数に前記上限加算値を加算することを特徴とする請求項４に記載の権限管理サーバー。
【請求項６】
前記入力画面にはさらに、前記アクセス上限数の引き上げを認めるか否かを示す加算許可情報を入力するための入力欄が含まれ、前記設定手段は、前記入力画面において入力された値を設定し、
前記リソースに対するアクセス数が前記アクセス上限数に到達したクライアントについては、前記加算許可情報により前記アクセス上限数の引き上げが認められている場合には、前記クライアントからの要求に応じて、前記アクセス上限数に前記上限加算値を加算することを特徴とする請求項５に記載の権限管理サーバー。
【請求項７】
前記発行手段は、前記登録されたクライアントからの認可要求に応じて認可トークンを発行し、
前記入力画面にはさらに、クライアント期限の入力欄が含まれ、前記設定手段は、前記入力画面において入力された値を設定し、
前記権限管理サーバーはさらに、前記登録されたクライアントのうち、リソースにアクセスしていない期間が前記クライアント期限を超えたクライアントを削除する手段を更に有することを特徴とする請求項４乃至６のいずれか一項に記載の権限管理サーバー。
【請求項８】
前記クライアントからの要求に応じて、前記登録されたクライアントを削除する削除手段をさらに有することを特徴とする請求項１乃至６のいずれか一項に記載の権限管理サーバー。
【請求項９】
前記クライアント期限は、前記第１の端末に表示された入力画面から入力され、設定されることを特徴とする請求項７に記載の権限管理サーバー。
【請求項１０】
前記認可トークンを検証した結果、当該認可トークンが正当であると検証された場合、認証情報の入力を要求することなくリソースに対するアクセスを許可することを特徴とする請求項１乃至９のいずれか一項に記載の権限管理サーバー。
【請求項１１】
請求項１乃至１０のいずれか一項に記載の権限管理サーバーと、
前記権限管理サーバーに接続された第２の端末と、
前記第２の端末からリソース要求が前記認可トークンとともにあった場合には、前記認可トークンの検証を前記権限管理サーバーに要求し、前記権限管理サーバーから、前記認可トークンを認める応答があった場合には、前記第２の端末から要求されたリソースを提供するリソースサーバーと
を有することを特徴とするリソース提供システム。
【請求項１２】
請求項１乃至１０のいずれか一項に記載の権限管理サーバーとしてコンピューターを機

能させるためのプログラム。
【請求項１３】
発行手段と検証手段とを有する権限管理サーバーにより実行される権限管理方法であって、
前記発行手段が、登録されたクライアントからの、リソースに対するユーザーのアクセス権限の委譲を要求する認可要求に応じて、前記認可要求を検証し、検証が成功した場合に前記クライアントに対して認可トークンを発行する発行工程と、
前記検証手段が、リソース要求が前記認可トークンとともにあった場合、前記認可トークンを検証し、検証が成功した場合に前記リソースに対するアクセスを許可する検証工程とを有し、
前記検証工程では、前記認可トークンの正当性を検証するとともに、前記リソースに対するアクセス数が前記リソース要求を行ったクライアントに対して設定されているアクセス上限数を超えたか否かも検証し、前記認可トークンが正当であり、かつ前記リソースに対するアクセス数が前記アクセス上限数を超えない場合に、前記認可トークンの検証が成功したと判定されることを含むことを特徴とする権限管理方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、リソースへのアクセス権限を管理する権限管理サーバー及び権限管理方法に関する。特に、クライアントIDごとにリソース利用回数を管理・制限する権限管理サーバー及び権限管理方法に関する。
【背景技術】
【０００２】
近年、スマートフォンやタブレットコンピューターといったモバイル端末が急速に普及している。このようなモバイル端末に対して、インターネット上のアプリケーションストアなどを通じて、アプリケーション開発者が開発したアプリケーションを容易に公開・販売できる仕組みが用意されている。また、モバイル端末向けのアプリケーション開発において、モバイル端末単体では実現が困難な機能を、インターネット上のWebサービスとして提供し、Webサービス利用料金を徴収するといったインターネットサービス事業も出現してきている。特に、サーバーサイドのコード開発・サーバー運用が不要で、WebサービスAPIを利用した分だけ課金するというBaaS(Backend as a Service)というWebサービス提供形態が出てきている。
【先行技術文献】
【特許文献】
【０００３】
【特許文献１】特開2004-310652号公報
【発明の概要】
【発明が解決しようとする課題】
【０００４】
前述のBaaSのようなWebサービスを利用した端末のアプリケーションを開発・運用する場合、BaaSと利用契約するのはアプリケーション開発者である。例えば、BaaSが、モバイル端末で表示できない電子文書ファイルを別の電子文書ファイルフォーマットに変換するAPIを提供していたとする。開発者は、アプリケーション内でフォーマット変換の必要なときに、BaaSのAPIを呼び出すようにアプリケーションを実装しておけばよい。一方で、エンドユーザーには、フォーマット変換はアプリケーションの一機能として見えるが、そのバックエンドで動作するBaaSについては存在を意識する必要がない。アプリケーション開発者は、アプリケーションストアなどを通じて、エンドユーザーからアプリケーション購入料金・使用料金などの収入を得ることができる。一方で、アプリケーション開発者は、配布したアプリケーションから利用された分のWebサービス利用料金をBaaSに支払う必要がある。

【０００５】
ここで、アプリケーション開発者にとって、BaaSに支払う料金をエンドユーザーから得られる収入以下にコントロールしたい、という課題がある。しかしながら、アプリケーションの配布数やアプリケーションからのWebサービス呼出数をあらかじめ予測するのは困難で、BaaSからの課金をあらかじめ正確に見積もるのは難しい。その例として、以下の３つのケースが挙げられる。ケース１：アプリケーション配布数が急増して、API呼出しが急増する。ケース２：一部のヘビーユーザーの端末から大量にAPIが呼び出されて、API呼出し回数の上限に達してしまう。これにより、他のユーザーからのAPI呼出しができなくなる、あるいは、BaaSから開発者に想定以上の料金請求が来てしまう、などのケースが有り得る。ケース３：配布したアプリケーションが使われなくなって、エンドユーザーからアプリケーション開発者へのアプリケーション課金収入が減少する。この場合は、BaaSに支払う料金も抑制する必要があるが、段階的従量課金の料金メニューのような場合、下位のメニューに切り替えるなどの判断・手間・タイムラグなどが発生する。
【０００６】
従来、BaaS事業者の利用規約などでは、契約した料金プランの利用上限を超えた場合、契約者に警告が通知されて、上位プランへの切替を要求される、といったサービス運用が一般的である。しかしながら、前述したように、配布したアプリケーションがどれだけWebサービスを利用するかは予測困難である。そのため、前述の各ケースのような事が発生すると、開発者にとってはアプリケーション販売・課金の機会損失、エンドユーザーにとってはアプリケーションの機能が使用できない等の迷惑・影響が発生する。
【０００７】
特許文献１によれば、ウェブサーバにおいて、オブジェクトごとの同時アクセス数を管理・制限する先行技術が公開されている。しかしながら、この先行技術は、配布したアプリケーションによるAPI呼出しおよびそれによる課金を制御するものではない。
【０００８】
本発明が解決する課題は、アプリケーション開発者自身が配布したアプリケーションによるWebサービスAPI呼出しおよびそれによる課金を制御可能な仕組みを提供することである。
【課題を解決するための手段】
【０００９】
上記の課題を解決するため、本発明のシステムは以下のような構成を有する。
【００１０】
登録されたクライアントからの、リソースに対するユーザーのアクセス権限の委譲を要求する認可要求に応じて、前記認可要求を検証し、検証が成功した場合に前記クライアントに対して認可トークンを発行する発行手段と、
リソース要求が前記認可トークンとともにあった場合、前記認可トークンを検証し、検証が成功した場合に前記リソースに対するアクセスを許可する検証手段と、を有し、
前記検証手段は、前記認可トークンの正当性を検証するとともに、前記リソースに対するアクセス数が前記リソース要求を行ったクライアントに対して設定されているアクセス上限数を超えたか否かも検証し、前記認可トークンが正当であり、かつ前記リソースに対するアクセス数が前記アクセス上限数を超えない場合に、前記認可トークンの検証が成功したと判定されることを含むことを特徴とする権限管理サーバー。
【発明の効果】
【００１１】
本発明によれば、アプリケーションによるWebサービスAPIの呼出しなど、リソースへのアクセスの回数を、クライアントのアクセス権限を制御することで、クライアントごとに制限・制御することが可能となる。また、リソースへのアクセス回数の上限の柔軟な設定が可能となる。
【図面の簡単な説明】
【００１２】

【図１】本発明を実施するためのシステム構成およびネットワーク構成を示す図
【図２】情報処理機能ハードウェア構成図
【図３】本システムのソフトウェア構成説明図
【図４】テナント管理テーブル、ユーザー管理テーブルを示す図
【図５】API課金メニュー管理テーブル、テナント属性情報管理テーブルを示す図
【図６】クライアント証明書管理テーブル、クライアント管理テーブル、認可トークン管理テーブル、API呼出し回数管理テーブルを示す図
【図７】Webサービス利用登録フロー説明図
【図８】利用登録画面、API利用設定画面を示す図
【図９】クライアント登録処理フロー、認可トークン発行フロー説明図
【図１０】リソース要求API処理フロー説明図
【図１１】API呼出し上限数への加算処理フロー説明図
【図１２】API呼出し上限数への加算選択UI、コスト回収選択UIを示す図
【図１３】クライアントID削除処理フロー説明図
【図１４】クライアントID自動削除処理フローチャート
【発明を実施するための形態】
【００１３】
以下、本発明を実施するための最良の形態について図面を用いて説明する。本実施の形態においては、Webサービスからアプリケーションに対してセキュアなAPI認可手段を提供するために、インターネット標準であるOAuth 2.0に準拠した構成とする。特に、モバイル端末向けアプリケーションに対して、個々の端末を識別し、端末ごとにアクセス制御を実施可能とするために、OAuth 2.0のClient Credentials GrantによるAPI認可手段を提供する。
【００１４】
＜システム構成＞
図１は、本発明を実施するためのリソース提供システムのシステム構成およびネットワーク構成の一例を示している。ネットワーク１０１は、インターネットもしくはイントラネットなどである。ネットワーク機器１０２はルータやスイッチなど、ネットワークどうしを接続する機器である。ファイアウォール１０３はネットワーク間の通信許可の制御を行う。ＬＡＮ（Local Area Network）１０５は、コンピューター等の機器を接続する末端のネットワークであるが、有線通信のネットワークに限らず、無線LANや携帯電話通信ネットワークなどの無線通信網の場合もある。認可サーバー１１１は、リソースサーバー１１２等に対するユーザーあるいはクライアント（これらについては後述する）のアクセス権限を管理する権限管理サーバー。リソースサーバー１１２は、たとえばアプリケーション処理などのサービスをリソースとして提供するサービスである。クライアントコンピューター１２１、１２２は、たとえばパーソナルコンピューター、タブレットコンピューター、スマートフォンなどであり、これによりアプリケーションプログラム等を実行するとともに、リソースサーバー１１２にアクセスする。
【００１５】
図２は、認可サーバー１１１、リソースサーバー１１２、クライアントコンピューター１２１、１２２の情報処理機能のモジュール構成図を示している。ユーザーインターフェース２０１は、ディスプレイ、キーボード、マウス、タッチパネルなどによる情報の入出力を行う。これらのハードウェアを備えないコンピューターは、リモートデスクトップやリモートシェルなどにより、他のコンピューターから接続・操作することも可能である。ネットワークインターフェース２０２は、ＬＡＮなどのネットワークに接続して、他のコンピューターやネットワーク機器との通信を行う。ＲＯＭ２０４は組込済みプログラムおよびデータが記録されているＲＯＭである。ＲＡＭ２０５はデータやプログラム等の一時メモリ領域となる。二次記憶装置２０６はＨＤＤに代表されるような記憶装置であり、プログラムファイルやデータファイルを格納する。ＣＰＵ２０３は、ＲＯＭ２０４、ＲＡＭ２０５、二次記憶装置２０６などから読み込んだプログラムを実行する。各部は入出力イ

ンターフェース２０７を介して接続されている。

【００１６】

＜ソフトウェア構成＞

図３は、本システムのソフトウェア構成を示している。認可サーバー１１１は、HTTPサーバーモジュール３０１、Webアプリケーション３０２、データベース３０５を備える。HTTPサーバーモジュール３０１は、クライアントからのWebアクセスの要求と応答の通信を管理・制御し、必要に応じて、要求をWebアプリケーション３０２に転送する。Webアプリケーション３０２は、ブラウザーにHTML等のWebドキュメントや操作画面を提供するWebUI３０３と、RESTに代表されるようなWebサービスAPIにより認可処理を受け付ける認可API３０４を備える。データベース３０５は、Webアプリケーション３０２が使用するデータを格納する。Webアプリケーション３０２からの要求に応じて、データベース３０５は各種テーブルのレコードを追加・読出・更新・削除する。

【００１７】

リソースサーバー１１２は、HTTPサーバーモジュール３１１、Webアプリケーション３１２を備える。HTTPサーバーモジュール３１１は、クライアントからのWebアクセスの要求と応答の通信を管理・制御し、必要に応じて、要求をWebアプリケーション３１２に転送する。Webアプリケーション３１２は、RESTに代表されるようなWebサービスAPIにより各種の処理を受け付けるAPI３１３を備える。API３１３は、リソースサーバーが提供する機能に必要な処理を実行して、クライアントからのAPI呼出し要求に対する応答を生成し、HTTPサーバーモジュール３１１を介してクライアントに応答を返す。なお、リソースサーバーが提供する機能としては様々なものが有り得るので、Webアプリケーション３１２のみでは実行できない場合は、不図示の他のアプリケーションや他のサーバーに機能の実行を依頼して、応答を得ることも可能である。

【００１８】

ブラウザー３２１は、クライアントコンピューター１２１にインストールされて実行可能である。ブラウザー３２１は、WebUI３０３が提供するHTML等のWebドキュメントや操作画面を受信して表示し、ユーザーによる操作結果などをWebUI３０３に送信する。

【００１９】

アプリケーション３３１は、クライアントコンピューター１２２にインストールされて実行可能である。アプリケーション３３１は、API３１３にアクセスして、リソースサーバー１１２が提供する各種機能を利用可能である。

【００２０】

ここで、図３の構成において、各モジュールがOAuth 2.0におけるどのロールに当たるかを説明する。認可サーバー１１１が、OAuth 2.0の"Authorization Server"ロールである。リソースサーバー１１２が、OAuth 2.0の"Resource Server"ロールである。アプリケーション３３１が、OAuth 2.0の"Client"ロールおよび"Resource Owner"ロールである。以降の説明では、各モジュールは前記のOAuthのロールとしてAPI認可フローを実行する。また、本文中あるいは図中における「クライアント」とは、OAuth 2.0の"Client"ロールとして、認可API３０４およびAPI３１３などのWebサービスAPIの要求元として動作する個々のアプリケーション３３１の事を指す。

【００２１】

＜認可サーバー内のテーブル＞

図４、図５、図６は、認可サーバー１１１内のデータベース３０５の各種テーブルを示している。テナント管理テーブル４００は、テナントIDを管理するためのテーブルである。テナントID４０１はテナントIDを格納するカラムである。テナントID４０１は、認可サーバーおよびリソースサーバーによって提供するWebサービスが、様々な組織・個人などに利用される場合、セキュアにリソースを分離するための単位である。このようなシステムは一般にマルチテナントシステムと呼ばれる。

【００２２】

ユーザー管理テーブル４１０は、ユーザーを管理するためのテーブルである。テナント

ID４１１はユーザーが所属するテナントIDを格納するカラムである。ユーザーID４１２は、対応するテナントIDに属するユーザーIDを格納するカラムである。メールアドレス４１３はユーザーのメールアドレスを格納するカラムである。パスワード４１４はユーザーのパスワードを格納するカラムである。権限４１５はユーザーが所属するテナントにおいて付与されている権限を格納するカラムである。ここでは、権限４１５として、テナント内のすべてのデータに対する権限を持つテナント管理者と、制限された権限のみを持つ一般というユーザー権限があるものとする。

【００２３】

図５において、API課金メニュー管理テーブル５００は、認可サーバー１１１で用意された課金メニューを管理するテーブルである。課金メニューID５０１は課金メニューIDを格納するカラムである。課金メニュー名５０２は課金メニュー名を格納するカラムである。API呼出し上限数５０３は、１クライアントIDあたりAPI呼出し上限数を格納するカラムである。なお本実施形態では、API呼出し上限数は、予め定めた単位期間内の上限数である。API呼出しはリソースサーバー１１２が提供するリソースへのアクセスであるから、アクセス数の上限あるいはアクセス上限数と言い替えることもできる。単価５０４は課金ユニットの単価を格納するカラムである。本実施形態では、API呼出し上限数５０３で定義された１クライアントIDあたりAPI呼出し上限数を１回利用する権利を１課金ユニットとして換算し、利用した課金ユニット数分を課金する例を示している。しかしながら、課金形態は多様なものが有り得るので、ここでは一例を示すに留める。

【００２４】

テナント属性管理テーブル５１０は、各テナントの属性を管理するテーブルである。テナントID５１１はテナントIDを格納するカラムである。課金メニューID５１２はそのテナントが選択している課金メニューIDを格納するカラムである。初期上限数５１３は１クライアントIDあたりAPI呼出し上限数の初期値を格納するカラムである。加算許可５１４はAPI呼出し上限数への加算を許可／不許可の設定値である加算許可情報を格納するカラムである。上限加算値５１５は１クライアントIDあたりAPI呼出し上限数への加算数を格納するカラムである。クライアント期限５１６は、一定期間を超えてリソースへのアクセスを行っていないクライアントを自動削除する機能を遂行する際に参照される期限（一定期間）を格納するカラムである。

【００２５】

図６において、クライアント証明書管理テーブル６００は、クライアント証明書を管理するためのテーブルである。クライアント証明書はユーザーによるAPI利用設定に応じて作成され、例えばユーザーからクライアントに配布されてクライアントの認証のために用いられる。シリアル番号６０１はクライアント証明書のシリアル番号を格納するカラムである。発行者６０２は証明書の発行者を格納するカラムである。主体者６０３は証明書の主体者を格納するカラムである。開始日時６０４は証明書の有効期間の開始日時を格納するカラムである。終了日時６０５は証明書の有効期間の終了日時を格納するカラムである。テナントマスターDN６０６はテナントマスターDN（Distinguished Name）を格納するカラムである。

【００２６】

クライアント管理テーブル６１０は、クライアント管理のための各種情報を収めたテーブルである。クライアントID６１１はクライアントIDを格納するカラムである。シークレット６１２はクライアントのシークレットを保存するカラムである。テナントID６１３はクライアントが所属するテナントIDを格納するカラムである。種別６１４はクライアントの種別を格納するカラムである。クライアントの種別６１４として、テナントの管理権限を持つマスターと制限された権限のみを持つ一般のクライアント権限がある。DN６１５はクライアントのテナントマスターDNを格納するカラムである。クライアント管理テーブル６１０によって、OAuth 2.0のClientを個別に識別して管理する。

【００２７】

認可トークン管理テーブル６２０は認可トークンを管理するためのテーブルである。認

可トークンID６２１は各認可トークン固有のIDを格納するカラムである。クライアントID６２２は認可トークンの発行対象のクライアントIDを格納するカラムである。有効期限６２３は認可トークンの有効期限を格納するカラムである。

【００２８】

API呼出し管理テーブル６３０はAPIが呼出された回数をクライアントごとに管理するためのテーブルである。クライアントID６３１はクライアントIDを格納するカラムである。対象月６３２はAPI呼出し回数集計の対象年月を格納するカラムである。本実施形態では前述した単位期間として暦上の１月を採用し、月毎にAPI呼出し回数を集計するが、集計の期間は年や週など別の単位や期間でもよい。上限数６３３は、API呼出し回数の上限数の設定値を格納するカラムである。呼出し回数６３４は実際にクライアントからAPIが呼び出された回数を格納するカラムである。最終アクセス日時６３５はクライアントから最後にAPIが呼び出された最終アクセス日時を格納するカラムである。

【００２９】

＜ユーザーの利用登録手順＞

図７、図８を用いて、認可サーバー１１１、リソースサーバー１１２によって提供するWebサービスに利用登録するための処理フローを説明する。利用登録するユーザーは、アプリケーション３３１の開発者を主たるユーザーとする。

【００３０】

ユーザーは、ブラウザー３２１を使用し、所定のURIを指定してHTTPサーバーモジュール301へと利用登録画面の取得要求を送信すると、それをHTTPサーバーモジュール３０１経由で受信したWebUI３０３が提供する利用登録画面８００を取得して、表示する（S701，S702，S703）。利用登録画面８００は、利用者の情報や料金メニューを入力するための入力画面である。なお、ブラウザー３２１とWebUI３０３との間の要求／応答の交換はHTTPサーバーモジュール３０１経由となるが、以下の説明ではこの点は省略する。利用者情報８０１はユーザーのメールアドレスやパスワードを入力する利用者情報入力フィールドである。料金メニュー８０２は料金メニューの選択フィールドである。メニューの各項目には課金メニューIDが関連付けられており、選択された項目に応じて課金メニューIDが特定される。WebUI３０３は、利用登録画面取得要求に応じて、API課金メニュー管理テーブル５００を読み出して、料金メニュー８０２の選択肢を提供する。登録ボタン８０３は利用登録要求を送信するボタンである。ユーザーが利用者情報８０１にユーザーを識別するための情報を入力して、料金メニューを８０２で選択して、登録ボタン８０３を押下すると、ブラウザー３２１は、利用者の例えばメールアドレスやパスワードなどの識別情報と、選択された課金メニューIDとを含む利用登録要求をWebUI３０３に送信する（S704）。WebUI３０３は、利用登録要求に応じて、まずテナント管理テーブル４００にテナントIDを新規追加する。WebUI３０３は、入力された利用者情報に従って、ユーザー管理テーブル４１０にユーザーのレコードを追加し、権限４１５にはテナント管理者の権限を付与する。これにより、このユーザーは作成されたテナントの設定値などを変更することができる。WebUI３０３は、作成されたテナントIDをテナント属性管理テーブル５１０のテナントID５１１に、料金メニュー８０２で選択された課金メニューIDを課金メニューID５１２に追加的に格納する。WebUI３０３は、クライアント管理テーブル６１０に、種別６１４が「マスター」のクライアント（マスタークライアントと呼ぶ）を１つ作成する。また、作成したマスタークライアントのDN６１５と同一のテナントマスターDN６０６を持つクライアント証明書を作成し、その他の証明書情報をクライアント証明書管理テーブル６００のフィールド６０１、６０２、６０３、６０４、６０５に格納する（S705）。これらのテナントIDをはじめとする登録処理が完了すると、ブラウザー３２１に登録完了を応答する（S706）。

【００３１】

次に、ブラウザー３２１は、WebUI３０３からAPI利用設定画面８１０を取得して表示する（S707，S708，S709）。初期上限数８１１は１クライアントIDあたりのAPI呼出し上限数（アクセス上限数）の初期値を入力するフィールド（入力欄）である。加算許可８１２

はクライアントからのAPI呼出し上限数の加算の許可／不許可を選択するチェックボックスである。上限加算数８１３は１クライアントIDあたりのAPI呼出し上限数への加算数を入力するフィールドである。クライアント期限８１４は、クライアントがAPIを一定期間利用しない場合にそのクライアントIDを削除する設定において、その一定期間を（自動削除期限）を入力するフィールドである。

【００３２】

設定ボタン８１５はAPI利用設定の要求を送信するボタンである。ユーザーがAPI利用設定画面８１０にて、各設定値を入力・選択して、設定ボタン８１５を押下すると、ブラウザー３２１は設定要求をWebUI３０３に送信する（S710）。設定要求を受信したWebUI３０３は、テナント属性管理テーブル５１０の初期上限値５１３、加算許可５１４、上限加算値５１５、クライアント期限５１６に、API利用設定画面８１０にて入力された値を格納する（S711）。WebUI３０３は、ブラウザー３２１に設定完了を応答する（S712）。次に、ブラウザー３２１は、クライアント証明書取得要求をWebUI３０３に送信する（S713）。WebUI３０３は、作成したクライアント証明書をクライアント証明書管理テーブル６００から読み出して、ブラウザー３２１に応答する（S714，715）。

【００３３】

＜クライアント登録処理＞

次に、図９を用いて、クライアントの登録処理、認可トークンの発行処理のフローを説明する。アプリケーション３３１には、API利用設定に応じてユーザーに引き渡されたクライアント証明書が、クライアントであるアプリケーション３３１の開発者によって組み込まれており、そのアプリケーション３３１がクライアントコンピューター１２２にインストールされる。

【００３４】

アプリケーション３３１はHTTPサーバーモジュール３０１にクライアント登録要求を送信する（S901）。HTTPサーバーモジュール３０１は、クライアント登録要求に対して、クライアント証明書を呼出元に要求する。アプリケーション３３１はクライアント証明書をHTTPサーバーモジュール３０１に送信し、HTTPサーバーモジュール３０１は受信したクライアント証明書が有効であれば、クライアント登録要求を認可API３０４に転送する（S902，S903）。なお、クライアント証明書の認証は、たとえば、受信したクライアント証明書をクライアント証明書管理テーブル６００と照合することで行い、登録されていれば有効すなわち認証成功と判定できる。また本実施形態では、アプリケーション３３１が認可サーバー１１１の正当なクライアントであることを認証するためにクライアント証明書を用いているが、Basic認証やDigest認証など他の認証方式を用いることも可能である。

【００３５】

認可API３０４は、受信したクライアント証明書から得られたシリアル番号６０１を用いて、クライアント証明書管理テーブル６００を検索し、テナントマスターDN６０６を特定する。さらに、認可API３０４は、クライアント管理テーブル６１０を検索し、特定したテナントマスターDN６０６と同じDN６１５を持ち、かつ、クライアント種別６１４が「マスター」であるレコード（すなわちS705で登録したマスターレコード）を取得する（S904）。認可API３０４は、取得したマスターレコードのテナントID６１３を読み出す。認可API３０４は、クライアント管理テーブル６１０にレコードを追加し、UUIDに代表されるようなユニークなIDを採番してクライアントID６１１に格納し、前記読み出したテナントIDをテナントID６１３に格納する。シークレット６１２にも自動生成した十分な文字列長のシークレットを格納し、種別６１４には「一般」を格納する。認可API３０４は、API呼出し回数管理テーブルにレコードを追加し、前記採番されたクライアントIDをクライアントID６３１に格納する。また、対象年月６３２には現在の年月を格納し、上限数６３３にはテナント属性管理テーブル５１０に設定されている該当テナントの１クライアントあたりAPI呼出し上限数の初期値５１３を格納し、呼出し回数６３４には初期値０を格納する（S905）。認可API３０４は、アプリケーション３３１にクライアント登録要求の応答として、生成したクライアントIDとシークレットを返信する（S906）。アプリケーション

３３１は、受信したクライアントIDとシークレットを、後で読み出し可能なように記憶領域に保存しておく（S907）。以上がアプリケーション３３１をクライアントとして認可サーバー１１１に登録する処理フローであり、認可サーバー１１１が発行したクライアント証明書を持つ正当なクライアントのみが認可サーバー１１１に登録できる。

【００３６】

アプリケーション３３１はリソースサーバー１１２へのアクセスに際して、認可サーバー１１１から認可トークンを取得し、それによりユーザーからのアクセス権限の委譲を受ける。そのためにアプリケーション３３１は、前述の取得したクライアントIDおよびシークレットを使用して、認可API３０４に認可トークン要求（あるいは認可要求とも呼ぶ）を送信する（S908）。認可API３０４は、受信したクライアントIDおよびシークレットと一致するクライアントIDおよびシークレットがクライアント管理テーブル６１０に存在することを検証し、存在するならば要求元のクライアントを認証する（S909）。認可API３０４は、API呼出し回数管理テーブル６３０を要求元のクライアントIDで検索し、当月のAPI呼出し回数６３４とAPI呼出し回数上限数の設定値６３３とを取得する（S910）。認可API３０４は、当月のAPI呼出し回数６３４がAPI呼出し回数上限数の設定値６３３より少ないかどうかを判定する（S911）。ステップ911の判定がYesの場合、認可トークン管理テーブル６２０にレコードを追加して、認可トークンを生成する（S912）。認可API３０４は、生成した認可トークンの認可トークンID６２１および有効期限６２３をアプリケーション３３１に応答する（S913）。ステップ911の判定がNoの場合、認可API３０４はAPI呼出し回数が上限に到達したことを通知する上限到達エラーをアプリケーション３３１に応答する（S914）。発行された認可トークンは、その認可トークンを利用するクライアントが、リソースサーバーのユーザーからリソース（本例ではAPI呼出しあるいはAPIを介したサービスの提供）へのアクセス権限の委譲を受けていることを示す。

【００３７】

クライアントID登録後、初回の認可トークン要求時は、前述のステップS910，911のAPI呼出し回数上限到達判定は実質的には必要なく、認可トークンをアプリケーション３３１に発行してよい。しかしながら、認可トークンには有効期限６２３があるため、有効期限切れ後は、アプリケーション３３１はステップS908からの処理を再度実行して、別の認可トークンを再度要求する必要がある。２回目以降の認可トークン要求時に、前述のステップS910，911のAPI呼出し回数上限到達判定をして、API呼出し回数が既に上限に到達している場合は、認可トークンは発行されない。発行された認可トークンを用いて、認可されたAPIを呼び出すのがOAuth 2.0のAPI認可フローであるので、API呼出し回数が既に上限に到達している場合は、アプリケーション３３１からのリソースサーバー１１２のAPI３１３への呼出しが抑止される。これにより、リソースサーバー１１２への通信トラフィック軽減、CPU処理の軽減などの効果を得ることができる。

【００３８】

＜リソース要求処理＞

次に図１０を用いて、取得した認可トークンを使用してリソースサーバー１１２のAPI３１３を利用する処理フローを説明する。アプリケーション３３１は、取得した認可トークンを添付してリソース要求をAPI３１３に送信する（S1001）。リソース要求は、リソースデータベース１１２がアプリケーションにリソース（あるいはサービス）を提供するためのAPIを利用するための要求である。また要求対象のAPIは図１０では、API３１３相当する。API３１３は、認可API３０４に、受信した認可トークンの検証要求を送信する（S1002）。認可API３０４は、認可トークン管理テーブル６２０から、受信した認可トークンの認可トークンIDを検索し、該当する認可トークンIDがあった場合、現在日時が有効期限６２３に以前であることを検証する。有効期限が満了していないなら、その認可トークンの発行先のクライアントID６２２が、クライアント管理テーブル６１０に存在するか判定することで該当のクライアントIDが有効であることを確認する（S1003）。認可API３０４は、ステップS1003の検証処理の結果、該当クライアントIDが有効であれば受信した認可トークンが有効であると判定する（S1004）。該当する認可トークンが認可トークン管理

テーブルに未登録であったり、あるいは有効期限が満了していたり、あるいはクライアントIDが無効な場合、S1003の検証処理の結果、認可トークンは無効（あるいは正当ではない）と判定される。その場合、すなわちステップS1004の判定がNoの場合、認可API３０４は、認可トークン検証結果としてトークン無効エラーをAPI３１３に応答する（S1005）。API３１３は、アプリケーション３３１にリソース要求APIの応答として、トークン無効エラーを返信する（S1006）。

【００３９】

一方、認可トークンを検証した結果、当該認可トークンの正当性が検証された場合、認証情報の入力を要求することなくリソースに対するアクセスを許可する。そこでステップS1004の判定がYesの場合、認可API３０４は、API呼出し回数管理テーブル６３０を認可トークンの発行先のクライアントIDで検索し、当月のAPI呼出し回数６３４とAPI呼出し回数上限数の設定値６３３を取得する（S1007）。認可API３０４は、当月のAPI呼出し回数６３４がAPI呼出し回数上限数の設定値６３３より少ないかどうかを判定する（S1008）。ステップS1008の判定がYesの場合、API呼出し回数管理テーブル６３０の当月のAPI呼出し回数６３４に１を加算する（S1009）。認可API３０４は、API３１３に認可トークン検証結果が成功（OK）であることを応答する（S1010）。API３１３は、ステップS1001で受信したリソース要求の処理を実行し、応答を生成する（S1011）。API３１３は、リソース要求に対する応答として、ステップS1011で生成したリソース応答およびAPI呼出し成功（OK）をアプリケーション３３１に返信する（S1012）。ステップS1008の判定がNoの場合、認可API３０４はAPI３１３に認可トークン検証応答として、API呼出し回数が上限を超えたことを示す上限到達エラーを返信する（S1013）。API３１３は、アプリケーション３３１に、リソース要求に対する応答として、上限到達エラーを返信する（S1014）。

【００４０】

＜上限引き上げ処理＞

次に、図１１、図１２を用いて、API呼出し回数が上限に達してしまったときに、API呼出し上限数に加算して、上限を引き上げる処理フローを説明する。アプリケーション３３１は、前述のステップS914またはステップS1014で、自身のクライアントIDからのAPI呼出し回数が上限に達してしまったことの通知を受ける（S1101）。API呼出し回数の上限到達の通知を受けた場合、API呼出し上限数加算を選択するUI１２００を表示する（S1102）。アプリケーション３３１は、UI１２００にて、ユーザーが上限の加算を選択したかどうかを判定する（S1103）。ステップS1103の判定がNoの場合は処理を終了する。ステップS1103の判定がYesの場合、アプリケーション３３１はコスト回収処理を実施するUI１２１０を表示し、上限を追加するコスト回収の同意を選択させる（S1104）。このとき、UI１２１０に表示される回数や料金は予め決めた値でもよいが、サーバーに登録された値を用いてもよい。その場合、上限値に追加できる値はテナント属性管理テーブル５１０の上限加算値５１５に、課金メニューID５１２に応じた単価５０４はAPI課金メニュー管理テーブルに登録されている。また、上限への加算を許可するか否かを示す加算許可５１４もテナント属性管理テーブル５１０に登録されている。そこで、認可API３０４から上限到達エラーを送信する際に、エラーとともに、加算許可５１４、上限加算値５１５、単価５０４をアプリケーション３３１に送信してもよい。その場合、S1101の直後に加算が許可されているか判定し、許可されていなければ図１１の手順を終了する。また許可されている場合には、上限加算値５１５と単価５０４とを参照して、追加する回数と料金のそれぞれをUI１２１０に表示する。

【００４１】

アプリケーション３３１は、ユーザーがコスト回収に同意し、アプリケーションのユーザーから、アプリケーションの開発者または提供者へのコスト回収処理に成功したかどうかを判定する（S1105）。UI１２１０でアプリケーションのユーザーが「同意」ボタンを押したなら、ステップS1105ではコスト回収処理に成功したと判定する。ステップS1105の判定がNoの場合、処理を終了する。ステップS1105の判定がYesの場合、アプリケーション３３１は、API３０４に対して、クライアントIDおよびシークレットを指定して、API呼出

し上限数に加算する設定API（不図示）を呼び出す（S1106）。認可API３０４は、前述のステップS909同様、要求元のクライアントを認証する（S1107）。認可API３０４は、クライアント管理テーブル６１０から、クライアントID６１１が要求元のクライアントIDと一致するレコードを検索し、クライアントIDが所属するテナントIDを特定する。認可API３０４は、テナント属性管理テーブル５１０の前記テナントIDのレコードを読みだして、１クライアントIDあたりAPI呼出し上限数への加算値５１５を取得する。認可API３０４は、API呼出し回数管理テーブル６３０内の６３１が要求元のクライアントIDで、かつ対象年月６３２が現在の年月のレコードに対し、API呼出し上限数の設定値６３３に、前述の取得した加算値５１５を加算する（S1108）。これにより、API呼出し上限数の設定値６３３には、新しい上限数が設定される。認可API３０４は、API呼出し上限数に加算する設定APIに対する応答として、アプリケーション３３１に成功（OK）を返す。なお、S1108の冒頭において、まず該当するクライアントの属するテナントの、テナント属性管理テーブル５１０に登録された加算許可５１４を参照し、不許可であればkさん失敗の応答をアプリケーションに応答してもよい。

【００４２】

以上の手順により、APIの使用回数、換言すればリソースへのアクセス回数が予め定めた上限値に達した際に、その上限値を引き上げることが可能となる。上記手順では加算許可情報が参照されているが、上限値の引き上げを無条件に許可するように構成してもよい。この場合には図８の加算許可情報の入力欄８１２は必要ない。

【００４３】

＜クライアントID削除処理＞

次に、図１３を用いて、不必要になったクライアントIDを削除する処理フローを説明する。この処理は、アプリケーション３３１のユーザーが、API３１３を呼び出すことによって実現しているアプリケーション３３１内の機能が不要になったときなど、その機能を無効化する場合に使用可能である。または、アプリケーション３３１のユーザーが、アプリケーションの開発者または提供者からの課金を解約した場合に、該当クライアントのWebサービスAPI利用料金を削減するために使用可能である。例えば、ユーザーがアプリケーションの開発者または提供者からの課金を解約した場合、無料アプリケーションとして一部の機能のみは使用継続可能だが、一部の有料機能は使用不可とする、といったケースで有用である。

【００４４】

アプリケーション３３１において、アプリケーションの開発者または提供者からアプリケーションのユーザーに対する課金の解約処理を実行する（S1301）。アプリケーション３３１は、前記解約処理が成功したかどうかを判定する（S1302）。ステップS1302の判定がNoの場合、処理を終了する。ステップS1302の判定がYesの場合、アプリケーション３３１は、認可API３０４に対して、クライアントID・シークレット指定で、クライアントID削除APIを呼び出す（S1303）。認可API３０４は、前述のステップS909同様、要求元のクライアントを認証する（S1304）。認可API３０４は、クライアント管理テーブル６１０から、要求元のクライアントIDがクライアントID６１１に一致するレコードを削除する（S1305）。認可API３０４は、クライアントID削除APIの応答として、成功（OK）をアプリケーション３３１に返信する（S1306）。これにより、不必要となったクライアントIDをアプリケーション３３１から削除可能で、翌月からのWebサービスAPI利用料金を適切に減らすことができる。

【００４５】

次に、図１４を用いて、一定期間呼出しがないクライアントIDを自動削除する処理フローを説明する。この処理は、図１３に記載したようなクライアントIDの削除手順を踏まずに、ユーザーがアプリケーションを使用しなくなった、あるいは、アプリケーションがアンインストールされてしまった、などのケースにおいて有効である。クライアントIDが登録されたままになっていると、WebサービスAPI呼出しが無いのにもかかわらず、アプリケーション開発者にはWebサービスAPI利用料金が継続して請求され続けることになる。Web

サービスAPI呼出しが無くなったクライアントIDを自動削除することによって、WebサービスAPI利用料金を適切に減らすことが可能となる。結果として、アプリケーション開発者は、使われなくなったアプリケーション３３１の分は、余計なコストを払う必要がなくなる。

【００４６】

データベース３０５に対して、不図示のプログラムあるいはストアドプロシージャなどを用いて、定期的にバッチ処理を実行し、次のような手順で一定期間呼出しがないクライアントIDを自動削除する。図１４の手順は周期的に（一定期間ごとに）実行される。特にクライアント期限５１６の単位を周期として実行するのが望ましい。本例ではクライアント期限５１６が日数を単位としているので１日周期で実行するのが望ましい。まず、クライアント管理テーブル６１０から、種別６１４が一般であるクライアントID６１１およびテナントID６１３を取得する（S1401）。前記取得したクライアントIDで、API呼出し回数管理テーブル６３０を検索し、該当クライアントIDの最終アクセス日時６３５のうち、最新の日時を取得する（S1402）。現在日時から前記取得した該当クライアントIDの最終アクセス日時を引いた日数が、テナント属性管理テーブル５１０の該当テナントIDに対するクライアント期限５１５より大きいかを判定する（S1403）。ステップS1403の判定がNoの場合、処理を終了する。ステップS1403の判定がYesの場合、該当クライアントIDをクライアント管理テーブル６１０から削除する（S1404）。これにより、一定期間アクセスがないクライアントIDを自動削除可能で、翌月からのWebサービスAPI利用料金を適切に減らすことができる。

【００４７】

以上説明したように、認可サーバー１１１は、リソースサーバー１１２のAPI３１３へのAPI利用要求に対して、OAuth 2.0の認可フローに準拠したAPI認可を提供する。特に、クライアントコンピューター１２２のアプリケーション３３１ごとに、クライアントIDを払い出す事が可能で、クライアントIDごとにリソースサーバー１１２のAPI３１３へのAPI呼出し回数を管理・制限することが可能である。また、OAuth 2.0の認可フローで必須の処理である認可トークン発行時、および、認可トークン検証時に、要求元クライアントIDごとにAPI呼出し回数の上限到達の検証が可能である。これにより、認可サーバー１１１に保存されたテナントごとのテナント属性管理テーブル５１０およびAPI利用画面８１０等を用いて、アプリケーション開発者自身が、配布したアプリケーション３３１からのAPI呼出し回数を制御可能となり、冒頭で述べた課題を解決する効果を得ている。

【００４８】

なお本実施形態ではAPIの呼出し回数の上限値との比較を、認可トークンの発行を要求する場合と、認可トークンを用いてその検証を受ける場合との２つの場合に行っている。これにより、使用できない認可トークンの発行を抑制できる。しかし、単に使用回数の制限を行う目的であれば、認可トークンの発行を要求する場合の比較は行わなくともよい。

【００４９】

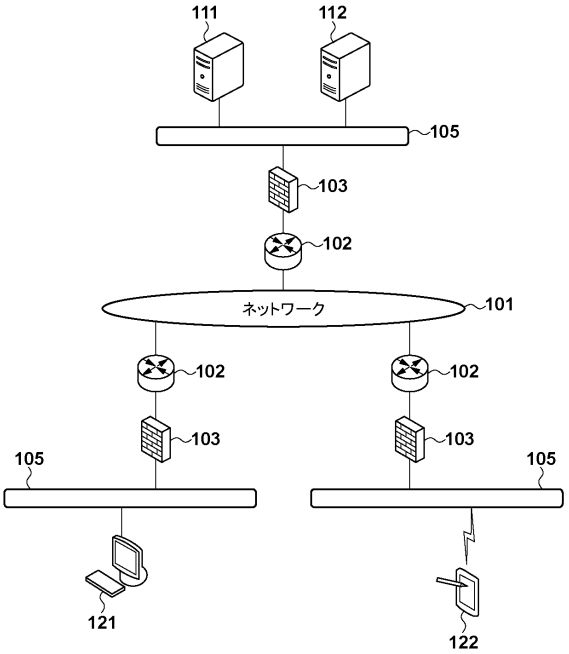
また、図１３、図１４で説明したクライアントIDの削除は、本実施形態における認可トークンの発行や検証とは独立して実施できる。

【００５０】

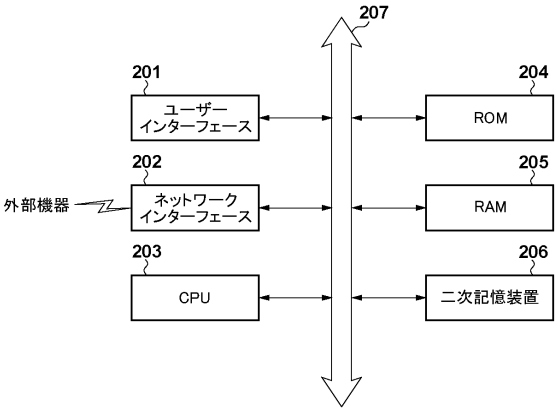
［その他の実施形態］

また、本発明は、以下の処理を実行することによっても実現される。即ち、上述した実施形態の機能を実現するソフトウェア（プログラム）を、ネットワーク又は各種記憶媒体を介してシステム或いは装置に供給し、そのシステム或いは装置のコンピュータ（またはＣＰＵやＭＰＵ等）がプログラムを読み出して実行する処理である。

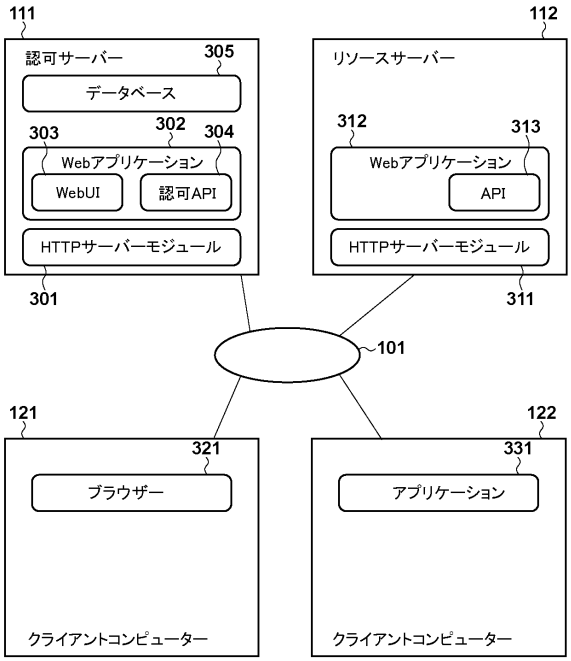
【図１】



【図２】



【図３】



【図４】

400

テーブル名	テナント管理テーブル
テナントID	
TN001	
TN002	

401

410

テーブル名	ユーザー管理テーブル			
テナントID	ユーザーID	メールアドレス	パスワード	権限
TN001	U001@TN001	aaa@bbb.com	********	テナント管理者
TN002	U002@TN002	xxx@yyy.com	********	一般

411 412 413 414 415

【図５】

500

テーブル名	API課金メニュー　管理テーブル		
課金メニューID	課金メニュー名	API呼出し上限数	課金ユニットの単価
MN001	A	100	¥100
MN002	B	1,000	¥500
MN003	C	10,000	¥3,000

501 502 503 504

510

テーブル名	テナント属性情報　管理テーブル				
テナントID	課金メニューID	初期上限数	加算許可	上限加算値	クライアント期限(日)
TN001	MN001	100	TRUE	100	90
TN002	MN003	10,000	FALSE	0	365

511 512 513 514 515 516

【図６】

600

テーブル名	クライアント証明書　管理テーブル				
シリアル番号	発行者	主体者	開始日時	終了日時	テナントマスターDN
00abcdef000000000001	Root CA 01	U001@TN001	2013/05/30	2015/05/30	CN=U001, OU=TN001
00abcdef000000000002	Root CA 01	U002@TN002	2013/09/30	2015/09/30	CN=U002, OU=TN002

601 602 603 604 605 606

610

テーブル名	クライアント管理テーブル				
クライアントID	シークレット	テナントID	種別	DN	
U001@TN001	**********	TN001	マスター	CN=U001, OU=TN001	
053753a39d3e4e648213f17eb1331a31@TN001	**********	TN001	一般		
543ae4f3998be4eb7ed92ea99e43f2aeTN001	**********	TN001	一般		

611 612 613 614 615

620

テーブル名	認可トークン　管理テーブル		
認可トークンID	クライアントID	有効期限	
AT_0000001	053753a39d3e4e648213f17eb1331a31@TN001	2013/05/30 24:00:00	
AT_0000002	543ae4f3998be4eb7ed92ea99e43f2aeTN001	2013/06/01 24:00:00	

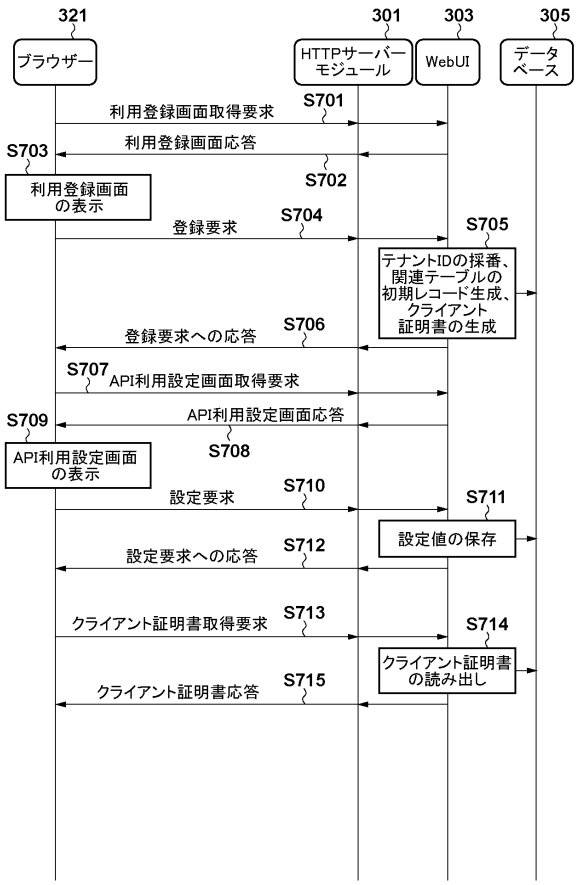
621 622 623

630

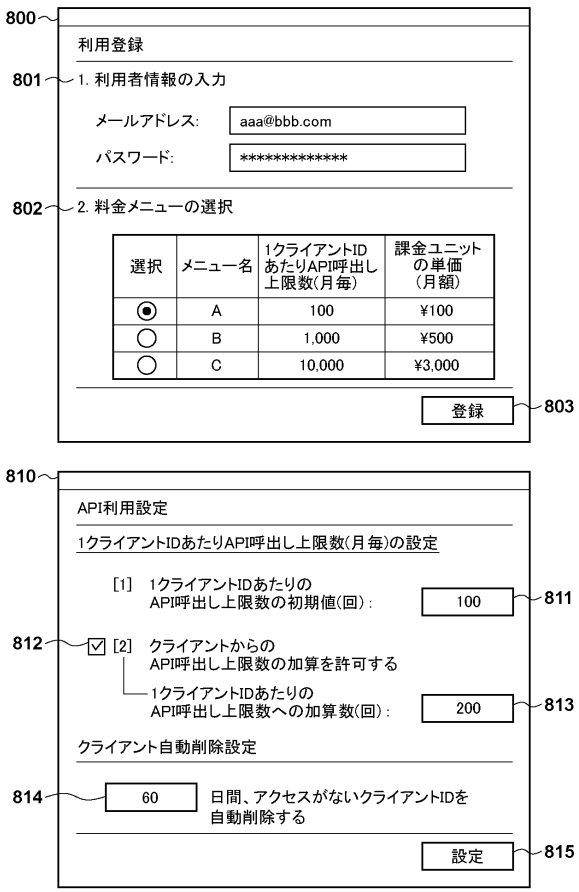
テーブル名	API呼出し　管理テーブル				
クライアントID	対象月	上限数	API呼出し回数	最終アクセス日時	
053753a39d3e4e648213f17eb1331a31@TN001	2013/07	100	92	2013/07/20 11:32:00	
543ae4f3998be4eb7ed92ea99e43f2aeTN001	2013/07	300	225	2013/07/25 20:05:32	

631 632 633 634 635

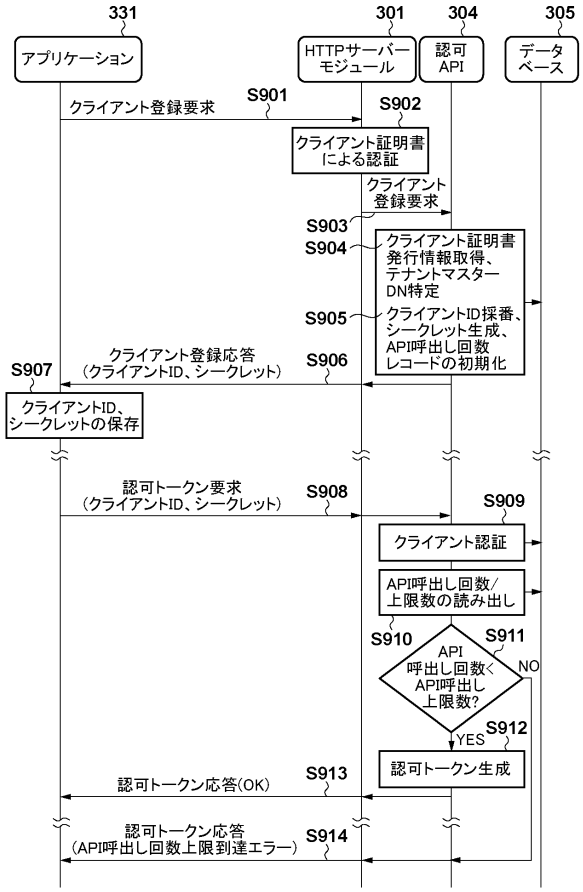
【図７】



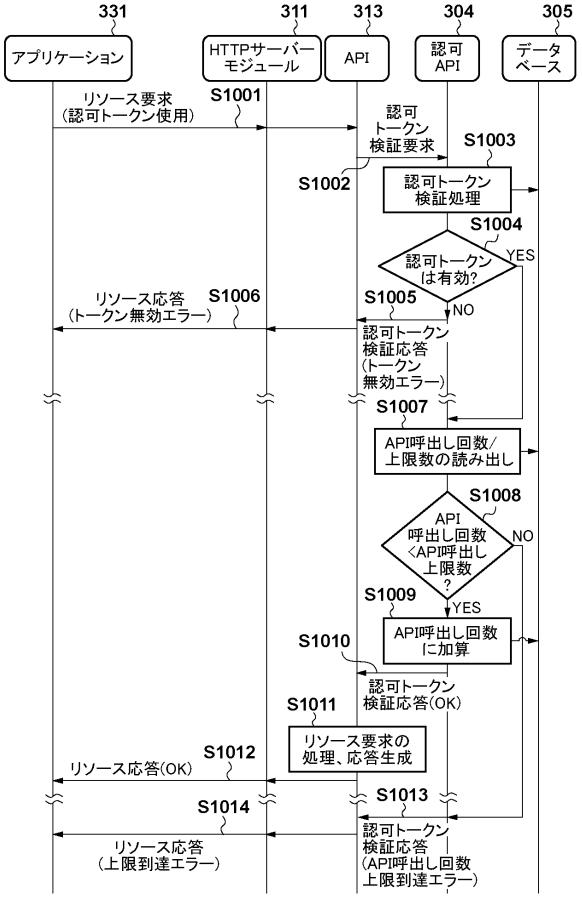
【図８】



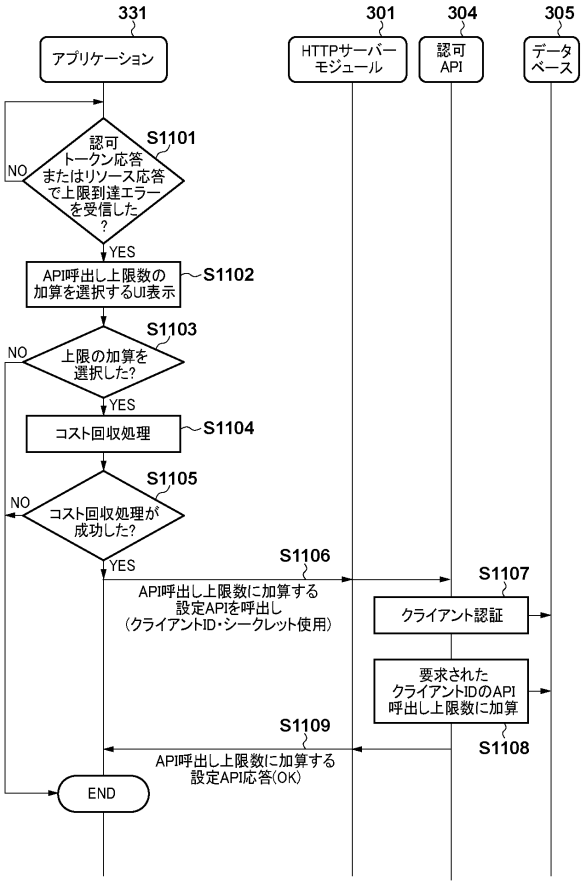
【図９】



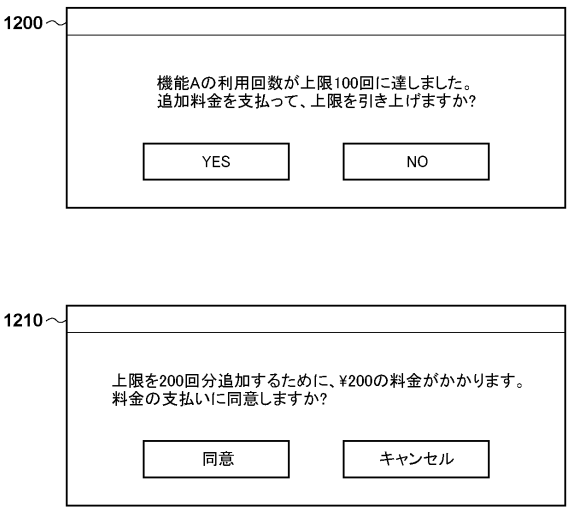
【図１０】



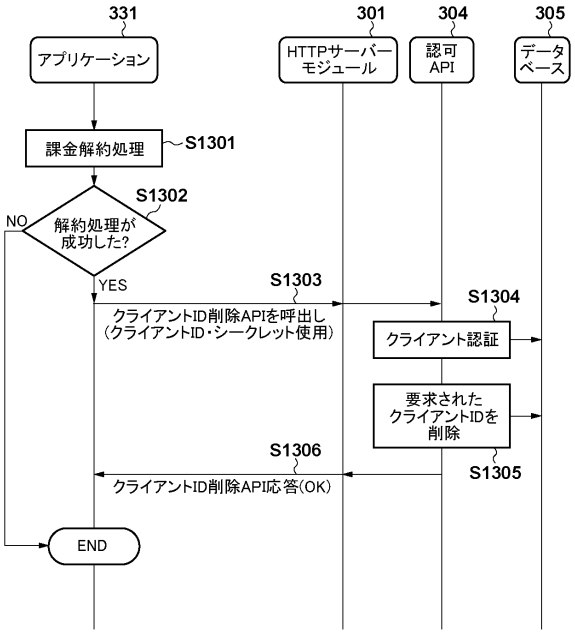
【図１１】



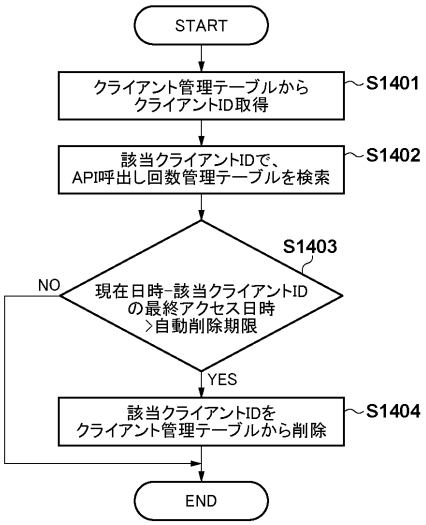
【図１２】



【図１３】


331
301
304
305
アプリケーション
HTTPサーバー
モジュール
認可
API
データ
ベース
課金解約処理
S1301
S1302
解約処理が
成功した?
NO
YES
S1303
クライアントID削除APIを呼出し
（クライアントID・シークレット使用）
S1304
クライアント認証
要求された
クライアントIDを
削除
S1305
S1306
クライアントID削除API応答(OK)
END

【図１４】


START
クライアント管理テーブルから
クライアントID取得
S1401
該当クライアントIDで、
API呼出し回数管理テーブルを検索
S1402
S1403
現在日時−該当クライアントID
の最終アクセス日時
>自動削除期限
NO
YES
該当クライアントIDを
クライアント管理テーブルから削除
S1404
END

フロントページの続き

(72)発明者　松田　浩太郎
東京都大田区下丸子３丁目３０番２号　キヤノン株式会社内

審査官　貝塚　涼

(56)参考文献　特開２００５－３３９２４７（ＪＰ，Ａ）
特開２００４－１２７１７２（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｑ　１０／００　－　９９／００
Ｇ０６Ｆ　２１／４４