(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6857065号
(P6857065)
(45)発行日　令和3年4月14日(2021.4.14)　(24)登録日　令和3年3月23日(2021.3.23)

(51)Int.Cl.		F I		
G06F 21/33	(2013.01)	G06F	21/33	
H04L 9/32	(2006.01)	H04L	9/00	675B
G09C 1/00	(2006.01)	G09C	1/00	640E
G06F 21/62	(2013.01)	G06F	21/62	
G06F 21/64	(2013.01)	G06F	21/64	

請求項の数 12　(全 29 頁)

(21)出願番号　特願2017-61886(P2017-61886)
(22)出願日　平成29年3月27日(2017.3.27)
(65)公開番号　特開2018-163616(P2018-163616A)
(43)公開日　平成30年10月18日(2018.10.18)
審査請求日　令和2年3月24日(2020.3.24)

(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　110003281
特許業務法人大塚国際特許事務所
(72)発明者　矢部　健太
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　岸野　徹

最終頁に続く

(54)【発明の名称】認証認可サーバー、リソースサーバー、認証認可システム、認証方法及びプログラム

(57)【特許請求の範囲】
【請求項１】
リソースサーバーにより提供されるリソースにアクセスするためのアクセストークンを発行する認証認可サーバーであって、
クライアントからのアクセストークン発行要求に応じて、該アクセストークン発行要求の所定のパラメータに基づいて、前記リソースサーバーにより検証できる第一のアクセストークンか、または前記認証認可サーバーにより検証される第二のアクセストークンか、いずれかを発行する手段と、
発行した前記第一のアクセストークン、または前記第二のアクセストークンを要求元のクライアントに送信する手段と、
前記第二のアクセストークンとともに受信した検証要求に応じて、前記第二のアクセストークンを検証する手段と
を有することを特徴とする認証認可サーバー。
【請求項２】
請求項１に記載の認証認可サーバーであって、
前記第一のアクセストークンは署名付きアクセストークンであり、前記第二のアクセストークンは署名なしアクセストークンであることを特徴とする認証認可サーバー。
【請求項３】
リソースを提供するリソースサーバーであって、
リソースの要求と共に受信したアクセストークンが、前記リソースサーバーにより検証

できる第一のアクセストークンか、または認証認可サーバーにより検証される第二のアクセストークンかを判定する第一の判定手段と、
受信したアクセストークンが前記第一のアクセストークンであると判定された場合、要求された前記リソースについて、前記第一のアクセストークンの検証を許可するか判定する第二の判定手段と、
前記第一のアクセストークンの検証を許可するとの判定結果に応じて、前記第一のアクセストークンを検証する検証手段と
を有することを特徴とするリソースサーバー。
【請求項４】
請求項３に記載のリソースサーバーであって、
リソースに関連づけて、当該リソースの要求にともなって受信した前記第一のアクセストークンの検証を前記検証手段により行ってよいか否かを示す第一のテーブルをさらに有し、
前記第二の判定手段は、前記第一のテーブルを参照して前記第一のアクセストークンの検証を許可するか判定することを特徴とするリソースサーバー。
【請求項５】
請求項３又は４に記載のリソースサーバーであって、
発行済みの前記第一のアクセストークンは前記認証認可サーバーに登録されており、
リソースに関連づけて、当該リソースの要求にともなって受信した前記第一のアクセストークンが前記認証認可サーバーから削除されている場合にも、前記リソースの提供を許容するか否かを示す第二のテーブルをさらに有し、
前記第一のアクセストークンが前記認証認可サーバーから削除されていても、前記第二のテーブルに許容することが示されている場合には、当該第一のアクセストークンを前記検証手段により検証することを特徴とするリソースサーバー。
【請求項６】
請求項５に記載のリソースサーバーであって、
前記第一のアクセストークンを受信した場合には、前記認証認可サーバーから削除された前記第一のアクセストークンを管理する削除トークン管理サーバーに問い合わせ、前記第一のアクセストークンが前記認証認可サーバーから削除されているとの応答を受けた場合に、前記第二のテーブルを参照することを特徴とするリソースサーバー。
【請求項７】
請求項３又は４に記載のリソースサーバーであって、
前記リソースサーバーに対して前記リソースを要求するクライアントは前記認証認可サーバーに登録されており、
前記クライアントが前記認証認可サーバーから削除されている場合には、当該クライアントによる前記リソースの要求を拒絶することを特徴とするリソースサーバー。
【請求項８】
請求項７に記載のリソースサーバーであって、
前記クライアントからリソースの要求を受信した場合には、前記認証認可サーバーから削除された前記クライアントを管理する削除クライアント管理サーバーに問い合わせ、前記クライアントが前記認証認可サーバーから削除されているとの応答を受けた場合に、当該クライアントによるリソースの要求を拒絶することを特徴とするリソースサーバー。
【請求項９】
リソースサーバーと、該リソースサーバーにより提供されるリソースにアクセスするためのアクセストークンを発行する認証認可サーバーとを含む認証認可システムであって、
前記認証認可サーバーは、
クライアントからのアクセストークン発行要求に応じて、該アクセストークン発行要求の所定のパラメータに基づいて、前記リソースサーバーにより検証できる第一のアクセストークンか、または前記認証認可サーバーにより検証される第二のアクセストークンか、いずれかを発行する手段と、

発行した前記第一のアクセストークン、または前記第二のアクセストークンを要求元のクライアントに送信する手段と、
前記第二のアクセストークンとともに受信した検証要求に応じて、前記第二のアクセストークンを検証する手段とを有し、
前記リソースサーバーは、
リソースの要求と共に受信したアクセストークンが、前記リソースサーバーにより検証できる第一のアクセストークンか、または認証認可サーバーにより検証される第二のアクセストークンかを判定する第一の判定手段と、
受信したアクセストークンが前記第一のアクセストークンであると判定された場合、要求された前記リソースについて、前記第一のアクセストークンの検証を許可するか判定する第二の判定手段と、
前記第一のアクセストークンの検証を許可するとの判定結果に応じて、前記第一のアクセストークンを検証する検証手段とを有する
ことを特徴とする認証認可システム。
【請求項１０】
リソースサーバーにより提供されるリソースにアクセスするためのアクセストークンを発行する認証認可サーバーとしてコンピュータを機能させるためのプログラムであって、
クライアントからのアクセストークン発行要求に応じて、該アクセストークン発行要求の所定のパラメータに基づいて、前記リソースサーバーにより検証できる第一のアクセストークンか、または前記認証認可サーバーにより検証される第二のアクセストークンか、いずれかを発行する手段と、
発行した前記第一のアクセストークン、または前記第二のアクセストークンを要求元のクライアントに送信する手段と、
前記第二のアクセストークンとともに受信した検証要求に応じて、前記第二のアクセストークンを検証する手段と
してコンピュータを機能させるためのプログラム。
【請求項１１】
リソースを提供するリソースサーバーとしてコンピュータを機能させるためのプログラムであって、
リソースの要求と共に受信したアクセストークンが、前記リソースサーバーにより検証できる第一のアクセストークンか、または認証認可サーバーにより検証される第二のアクセストークンかを判定する第一の判定手段と、
受信したアクセストークンが前記第一のアクセストークンであると判定された場合、要求された前記リソースについて、前記第一のアクセストークンの検証を許可するか判定する第二の判定手段と、
前記第一のアクセストークンの検証を許可するとの判定結果に応じて、前記第一のアクセストークンを検証する検証手段と
してコンピュータを機能させるためのプログラム。
【請求項１２】
リソースサーバーと、該リソースサーバーにより提供されるリソースにアクセスするためのアクセストークンを発行する認証認可サーバーとを含む認証認可システムにおける認証方法であって、
前記認証認可サーバーが、
クライアントからのアクセストークン発行要求に応じて、該アクセストークン発行要求の所定のパラメータに基づいて、前記リソースサーバーにより検証できる第一のアクセストークンか、または前記認証認可サーバーにより検証される第二のアクセストークンか、いずれかを発行し、
発行した前記第一のアクセストークン、または前記第二のアクセストークンを要求元のクライアントに送信し、
前記第二のアクセストークンとともに受信した検証要求に応じて、前記第二のアクセ

ストークンを検証し、
前記リソースサーバーが、
リソースの要求と共に受信したアクセストークンが、前記リソースサーバーにより検証できる第一のアクセストークンか、または認証認可サーバーにより検証される第二のアクセストークンかを判定し、
受信したアクセストークンが前記第一のアクセストークンであると判定された場合、要求された前記リソースについて、前記第一のアクセストークンの検証を許可するか判定し、
前記第一のアクセストークンの検証を許可するとの判定結果に応じて、前記第一のアクセストークンを検証する
ことを特徴とする認証方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、署名付きアクセストークンを用いた認証認可サーバー、リソースサーバー、認証認可システム、認証方法及びプログラムに関する。
【背景技術】
【０００２】
近年、インターネット上に展開された所謂クラウドサービスの利用が拡大している。また、これらクラウドサービスは個々にＷｅｂサービスのＡＰＩ（Ａｐｐｌｉｃａｔｉｏｎ Ｐｒｏｇｒａｍｍｉｎｇ Ｉｎｔｅｒｆａｃｅ）を公開しており、他のアプリケーションやクラウドサービスからＡＰＩを介してサービスが提供する機能を利用する事が可能となっている。これらＷｅｂサービスＡＰＩではＯＡｕｔｈ ２．０と呼ばれる、認可の連携を実現させるための標準プロトコルの採用が進んでいる。
【０００３】
ＯＡｕｔｈ ２．０によれば、例えばあるサービスＡが管理するユーザーのデータを取得するＡＰＩに対して、そのユーザーから認められた範囲にてサービスＢがアクセスすることができる。このときサービスＡは、サービスＢからアクセスされる範囲を明らかにした上で、サービスＢによるＡＰＩへのアクセスに対してユーザーの明示的な認可を得ることになっている。ユーザーが明示的に認可を行うことを認可操作と称する。また、このアクセスされる範囲をＯＡｕｔｈ ２．０ではスコープと呼称し、スコープによりデータへのアクセス範囲が決定される。
【０００４】
ユーザーが認可操作を行うと、サービスＢは、サービスＡにおけるユーザーが許可した範囲のデータへのアクセスが認められたことを証明するトークン（以下、アクセストークンと称する）を受け取り、以降のサービスＡのＡＰＩへのアクセスはそのアクセストークン（認可トークンとも呼ぶ）を用いて実現できる。このユーザーの認可操作によりサービスＢがユーザーのデータに対しアクセスすることを認可したことを権限委譲と称する。なお、ＯＡｕｔｈ ２．０では、ユーザーの認可操作を元にアクセストークンを発行するサーバーを認可サーバーと呼称する。また認証機能を併せ持つ認可サーバーを認証認可サーバーと呼ぶ。また、ＡＰＩを公開するサーバーをリソースサーバー、ＡＰＩをコールする主体をクライアントと呼称する。
【０００５】
ＯＡｕｔｈ ２．０では、認可サーバーとリソースサーバーは同一サーバーで構成する事も可能としているが、リソースサーバーが複数種存在する様な大規模なシステムでは、通常認可サーバーを独立したサーバーとして構成する。その場合、上記フローにおいて、サービスＡはサービスＢからのＡＰＩ利用のたびに、取得したアクセストークンの検証を認可サーバーに依頼する。そして、その検証結果を元にＡＰＩの利用の可否を決定する。その場合、大規模システムでは、認可サーバーに負荷が集中してしまうという課題が発生する。

【０００６】
そのような課題に対して、例えば特許文献１のように、認可サーバーが発行するアクセストークンに予め認可情報（トークンID，スコープ，有効期限など）およびトークンに紐付く情報（ユーザーID、クライアントID、ユーザー名、メールアドレスなど）を付与することでリソースサーバー自身がアクセストークンを検証することを可能とし、認可サーバーにかかる負荷を低減する方法がある。また、サービスＡが認可サーバーに検証依頼することなく、自身でアクセストークンを検証する手段として、署名付きアクセストークンがある。署名付きアクセストークンとしては、ＪＷＴ（ＪＳＯＮ　Ｗｅｂ　Ｔｏｋｅｎ）、ＪＷＳ（ＪＳＯＮ　Ｗｅｂ　Ｓｉｇｎａｔｕｒｅ）が知られている。この種のアクセストークンを用いれば、サービスＡは受信した署名付きアクセストークンの署名を検証する事で、認可サーバーに確認することなく、アクセストークンが有効であるかを判断する事ができる。
【先行技術文献】
【特許文献】
【０００７】
【特許文献１】特開２００７－１４９０１０号公報
【発明の概要】
【発明が解決しようとする課題】
【０００８】
前記認可サーバーで発行する署名付きアクセストークンの認可情報、該トークンに紐付く情報、例えばユーザー情報やクライアント情報などについては、アクセストークンの有効期限内に認可サーバーによって変更になる可能性がある。そのような場合、署名付きアクセストークンを検証するリソースサーバーでは認可サーバーによって行われた変更を確認することができない。認可サーバーによって変更された今現在の署名付きアクセストークンの認可情報、該トークンに紐付く情報をリソースサーバーが確認するためには、トークン検証の際に認可サーバーに問い合わせなければならない。しかし、これでは従来のＯＡｕｔｈ　２．０仕様、実装などの認可サーバーのトークン検証、情報取得と同様にクライアント、リソースサーバーの台数が増えるに従い、認可サーバーのトークン検証、情報取得の負荷が高くなる。また、リソースサーバーが毎回認可サーバーにトークン検証、情報取得を行うことより、リソースサーバーのパフォーマンスが低下することになる。このように、認可サーバーの負荷の軽減を実現しながら、併せて発行済みのアクセストークンに係る権限や属性の変更を許容することは困難であった。
【０００９】
本発明は上記従来例に鑑みて成されたもので、アクセストークンを検証するサーバーを選択できる認証認可サーバー、リソースサーバー、認証認可システム、認証方法及びプログラムを提供することを目的とする。
【課題を解決するための手段】
【００１０】
上記目的を達成するために本発明は以下のような構成を有する。
【００１１】
本発明の一側面によれば、本発明は、リソースサーバーにより提供されるリソースにアクセスするためのアクセストークンを発行する認証認可サーバーであって、
クライアントからのアクセストークン発行要求に応じて、該アクセストークン発行要求の所定のパラメータに基づいて、前記リソースサーバーにより検証できる第一のアクセストークンか、または前記認証認可サーバーにより検証される第二のアクセストークンか、いずれかを発行する手段と、
発行した前記第一のアクセストークン、または前記第二のアクセストークンを要求元のクライアントに送信する手段と、
前記第二のアクセストークンとともに受信した検証要求に応じて、前記第二のアクセストークンを検証する手段と

を有することを特徴とする認証認可サーバー。
【００１２】
本発明の他の一側面によれば、本発明は、リソースを提供するリソースサーバーであって、
リソースの要求と共に受信したアクセストークンが、前記リソースサーバーにより検証できる第一のアクセストークンか、または認証認可サーバーにより検証される第二のアクセストークンかを判定する第一の判定手段と、
受信したアクセストークンが前記第一のアクセストークンであると判定された場合、要求された前記リソースについて、前記第一のアクセストークンの検証を許可するか判定する第二の判定手段と、
前記第一のアクセストークンの検証を許可するとの判定結果に応じて、前記第一のアクセストークンを検証する検証手段と
を有することを特徴とするリソースサーバー。
【００１３】
本発明の他の一側面によれば、本発明は、リソースサーバーと、該リソースサーバーにより提供されるリソースにアクセスするためのアクセストークンを発行する認証認可サーバーとを含む認証認可システムであって、
前記認証認可サーバーは、
クライアントからのアクセストークン発行要求に応じて、該発行要求の所定のパラメータに基づいて、前記リソースサーバーにより検証できる第一のアクセストークンか、または前記認証認可サーバーにより検証される第二のアクセストークンか、いずれかを発行する手段と、
発行した前記第一のアクセストークン、または前記第二のアクセストークンを要求元のクライアントに送信する手段と、
前記第二のアクセストークンとともに受信した検証要求に応じて、前記第二のアクセストークンを検証する手段とを有し、
前記リソースサーバーは、
リソースの要求と共に受信したアクセストークンが、前記リソースサーバーにより検証できる第一のアクセストークンか、または認証認可サーバーにより検証される第二のアクセストークンかを判定する第一の判定手段と、
受信したアクセストークンが前記第一のアクセストークンであると判定された場合、要求された前記リソースについて、前記第一のアクセストークンの検証を許可するか判定する第二の判定手段と、
前記第一のアクセストークンの検証を許可するとの判定結果に応じて、前記第一のアクセストークンを検証する検証手段とを有する
ことを特徴とする認証認可システム。
【発明の効果】
【００１４】
本発明によれば、アクセストークンを検証するサーバーを選択することが可能となる。これにより認可サーバーの負荷の軽減と、発行済みのアクセストークンに係る権限や属性の変更の許容とを両立させることができ、リソースサーバーのパフォーマンスを上げることができる。
【図面の簡単な説明】
【００１５】
【図１】実施形態１におけるネットワーク構成図
【図２】ハードウェア構成図
【図３】ソフトウェア構成図
【図４】ＯＡｕｔｈ２．０におけるアクセストークン発行シーケンスを示す図
【図５】署名付きアクセストークン発行シーケンスを示す図
【図６】グラント種別に応じて発行するアクセストークンを決定する処理を示したフロー

チャート
【図７】署名付きアクセストークンの一例を示す図
【図８】実施形態１においてリソースサーバー１０３がステップＳ５０４で実施する認可確認処理を示したフローチャート
【図９】実施形態２におけるネットワーク構成図
【図１０】実施形態２におけるソフトウェア構成図
【図１１】アクセストークン及びクライアントを削除する処理を示したフローチャート
【図１２】実施形態２における処理要求シーケンスを示す図
【図１３】実施形態２においてリソースサーバー１０３がステップＳ１２０１で実施するトークンおよびクライアント確認処理を示したフローチャート
【図１４】アクセストークン発行要求のＨＴＴＰリクエストを示す図
【発明を実施するための形態】
【００１６】
以下、本発明を実施するための形態について図面を用いて説明する。
［実施形態１］
本実施形態においては、インターネット上の各サーバーにアプリケーションが設置されていることとする。サーバーにインストールされたアプリケーションはクライアント端末と連携し、様々な機能を提供することとする。このような機能を提供する実体をサービスと称し、機能をクライアント端末に提供することをサービスの提供と称する。本実施形態の形態に係る情報処理システムである認証認可システムは、図１に示すような構成のネットワーク上に実現され、そのうえで認証方法が実行される。
【００１７】
＜認証認可システムおよびデバイスの構成＞
ＷＡＮ１００は、Ｗｉｄｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ（広域ネットワーク、以下、ＷＡＮと略す）であり、本発明ではＷｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ（以下、WWWあるいはＷｅｂと略す）システムが構築されている。ＬＡＮ１０１は、各構成要素を接続するＬｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ（ローカルエリアネットワーク、以下、ＬＡＮと略す）である。
【００１８】
認証認可サーバー１０２は、ユーザー、クライアント端末の認証及びＯＡｕｔｈを実現するための認証認可サーバーである。すなわち認証認可サーバー１０２が要求に応じてアクセストークンを発行し、また検証要求に応じてアクセストークンを検証することができる。リソースサーバー１０３は、様々なサービスを提供するリソースサーバーである。また、実施例において各サーバーは１台ずつ設置されているが、それぞれが複数台で構成されていても良い。また、実施形態において認証認可サーバー１０２とリソースサーバー１０３はＬＡＮ１０１を介して接続されているがＷＡＮ１００を介して接続することも可能である。また、認証認可サーバー１０２は不図示のデータベースサーバーとＬＡＮ１０１を介して接続し、後述する認証認可モジュールが利用するデータを格納するように構成しても良い。さらには、認証認可サーバー１０２、リソースサーバー１０３を同一のハードウェアに構成しても良い。
【００１９】
クライアント端末１０４は、パソコン、モバイル端末、画像形成装置など、認証認可サーバー１０２やリソースサーバーとデータ送受信を行うクライアント端末である。クライアント端末１０４には後述する１つまたは複数のリソースサービス連携アプリケーションがインストールされており、リソースサーバー１０３が提供するサービスを利用することで、自身が提供する機能と合わせたサービスをユーザーに提供する。
【００２０】
図２は、本実施例に係る認証認可サーバー１０２、リソースサーバー１０３、クライアント端末１０４を構成する情報処理装置の一般的なハードウェア構成を示した図である。ＣＰＵ２０１は、ＲＯＭ２０３のプログラム用ＲＯＭに記憶された、あるいはハードディ

スク等の外部メモリ２１１からＲＡＭ２０２にロードされたオペレーティングシステム（以下、ＯＳと略す）やアプリケーション等のプログラムを実行する。また、ＣＰＵ２０１は、システムバス２０４に接続される各ブロックを制御する。後述する各シーケンスの処理は、ＣＰＵ２０１によるこのプログラムの実行により実現できる。

【００２１】

ＲＡＭ２０２は、ＣＰＵ２０１の主メモリ、ワークエリア等として機能する。操作部Ｉ／Ｆ２０５は、操作部２０９からの入力を制御する。ＣＲＴＣ２０６は、ＣＲＴディスプレイ２１０の表示を制御するＣＲＴコントローラーである。ＤＫＣ２０７は、各種データを記憶するハードディスク等の外部メモリ２１１におけるデータアクセスを制御するディスクコントローラーである。ＮＣ２０８は、ＷＡＮ１００もしくはＬＡＮ１０１を介して接続されたサーバーコンピューターや他の機器との通信制御処理を実行するネットワークコントローラーである。尚、後述の全ての説明においては、特に断りのない限り実行のハード上の主体はＣＰＵ２０１であり、ソフトウェア上の主体は外部メモリ２１１にインストールされたアプリケーションプログラムである。

【００２２】

図３は、本実施形態に係る認証認可サーバー１０２、リソースサーバー１０３、クライアント端末１０４それぞれのソフトウェアモジュール構成を示した図である。各々のモジュールが各構成要素で実行されることで、各々の機能を実現する。

【００２３】

認証認可サーバー１０２は、認証認可モジュール３１０を持つ。認証認可モジュール３１０では、ユーザーやクライアント端末１０４の認証処理、認証されたユーザーやクライアント端末１０４での認可処理、アクセストークンの発行処理、署名無しアクセストークンの検証処理を行う。

【００２４】

なお、実施形態において認証認可サーバー１０２が発行するアクセストークンのうち、図４に示されるように、認可情報の取得に際して認証認可サーバー１０２への問い合わせを必要とする本実施形態のアクセストークンにはデジタル署名がなく、これを署名無しアクセストークンと呼称することとする。また、実施形態において認証認可サーバー１０２が発行するアクセストークンのうち、図５に示す、認可情報の取得に際してリソースサーバーで１０３でのアクセストークンの検証が許可されるような本実施形態のアクセストークンにはデジタル署名がついており、これを署名付きアクセストークンと呼称して区別することとする。また、実施形態において署名無しアクセストークン、あるいは署名付きアクセストークンと区別する必要がなく認可サーバー１０２が発行するアクセストークンとして総称する場合にはアクセストークンと呼称することとする。

【００２５】

リソースサーバー１０３は、リソースサーバーモジュール３２０を持つ。リソースサーバーモジュール３２０は、ＷｅｂサービスとしてＡＰＩを公開し、クライアント端末１０４からのサービス提供の要求を受け付け、サービスの提供を行う。また、クライアント端末１０４からのサービス提供の要求に対して、署名付きアクセストークンの検証処理を行ったうえでサービス提供の可否を決定する。

【００２６】

クライアント端末１０４は、認証認可サーバー連携クライアント３３０、リソースサーバー連携クライアント３３１を持つ。また、クライアント端末１０４は、ＷＷＷを利用するためのユーザーエージェントであるＷｅｂブラウザ３３２を持つ場合がある。ただし、Ｗｅｂブラウザ３３２は必須のモジュールではない。

【００２７】

認証認可サーバークライアント３３０は、認証認可サーバー１０２に対して、ユーザーやクライアントの認証要求、アクセストークンの発行要求や取得を行う、

リソースサーバー連携アプリケーション３３１は、リソースサーバー１０３よりサービス提供を受けるアプリケーションである。リソースサーバー連携アプリケーション３３１

は、以下の手順でリソースサーバー１０３よりサービス提供を受ける。まず、リソースサーバー連携アプリケーション３３１は認証認可サーバー連携クライアント３３０に対してアクセストークンの発行を依頼する。認証認可サーバー連携クライアント３３０は、リソースサーバー連携アプリケーション３３１が求めるサービスに対応したアクセストークンを認証認可サーバー１０２から取得する。認証認可サーバー連携クライアント３３０は、取得したアクセストークンを要求元のリソースサービス連携アプリケーション３３１に返却する。リソースサービス連携アプリケーション３３１は取得したアクセストークンを利用して、リソースサーバー１０２へリソース要求を行うことでサービスの提供を受けることができる。

【００２８】

＜署名なしアクセストークン＞

図４は、署名無しアクセストークンの発行と検証の流れを示したシーケンス図である。アクセストークンの発行と検証は、認証認可サーバー１０２、リソースサーバー１０３、クライアント端末１０４が連携することで実現される。

【００２９】

ステップＳ４０１において、クライアント端末１０４は認証認可サーバー１０２に対して、認証情報あるいは認可コードを送信して、アクセストークンの発行要求を行う。クライアント端末が認証認可サーバー１０２へのアクセストークン発行要求を行う際に送信する情報と処理の流れは、要求されたリソースのオーナーであるか否かによって異なる。オーナーとは、リソースへのアクセスを許可するエンティティである。

【００３０】

クライアント端末１０４が、発行要求するアクセストークンの対象となるリソースのオーナーである場合には、クライアント端末１０４自体がオーナーとして認証処理を行うため、認証認可サーバー１０２に認証情報を送信する。クライアント端末１０４が、他のオーナーからの権限委譲に基づいてアクセストークンを要求する場合には、認証認可サーバー１０３が発行した認可コードをオーナーから受信し、その認可コードをトークン発行要求と共に認証認可サーバー１０２に送信する。

【００３１】

ステップＳ４０２において、認証認可サーバー１０２はクライアント端末１０４からトークン発行要求と共に受信した情報を確認し、署名なしアクセストークンをクライアント端末１０４に発行する。

【００３２】

ステップＳ４０３において、クライアント端末１０４はステップＳ４０１で取得したアクセストークンを使用して、リソースサーバー１０３にサービスの提供を要求する。

【００３３】

ステップＳ４０４において、リソースサーバー１０３はクライアント端末１０４から要求されたサービスの処理を実行する前に、認証認可サーバー１０２にクライアント端末１０４から受信したアクセストークンを送信して、受信したアクセストークンがサービスの処理を実行することに対して適正なトークンであるかどうかを確認する。

【００３４】

ステップＳ４０５において、認証認可サーバー１０２はリソースサーバー１０３から受信したアクセストークンの検証を行い、クライアント端末１０４が要求したサービスを提供するのに適正か否かを判断した結果をリソースサーバー１０３に返す。

【００３５】

ステップＳ４０６において、リソースサーバー１０３は認証認可サーバー１０２から受信したアクセストークンの検証結果に従い、サービスの処理を実行し、その結果をクライアント端末１０４に返す。

【００３６】

＜署名付きアクセストークン＞

引き続き、本実施形態における署名付きアクセストークンの説明を行う。本実施形態で

は、通常のアクセストークンの代わりに、アクセストークン情報及びリソースオーナー情報であるアクセストークンに紐付くユーザー情報を含んだ署名付きアクセストークンを実現するためにＪＷＳ、ＪＷＴの手法を利用する。以下、本実施形態で用いるＪＷＳ（ＪＳＯＮ Ｗｅｂ Ｓｉｇｎａｔｕｒｅ）はＪＷＴ（ＪＳＯＮ Ｔｏｋｅｎ）で表現されたコンテンツをデジタル署名やＭＡＣｓ（Ｍｅｓｓａｇｅ Ａｕｔｈｅｎｔｉｃａｔｉｏｎ Ｃｏｄｅｓ）により保護して表現する手段である。またＪＷＴは、ＪＳＯＮ（ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ）をベースとしたデータ構造を用いたＵＲＬセーフなクレームの表現方法である。ＪＷＳ、ＪＷＴについては、各々ＲＦＣ（ＲＦＣ７５１５（ＪＷＳ）、ＲＦＣ７５１９（ＪＷＴ））として仕様化、公開されている。本実施例で使用するＪＷＳに含まれるクレームは、以下である。クレームとはトークンの本体となる部分で、その内容の例を表１に示す。

【００３７】

【表１】

表１　ＪＷＳに含まれるクレーム

	クレーム名 クラス	クレーム名	クレーム内容
1	Registered Claim	“iss” (Issuer)	ＪＷＴ　の発行者の識別子
2		“sub” (Subject)	ＪＷＴ　の主語となる主体の識別子
3		“aud” (Audience)	ＪＷＴ　を利用することが想定された主体の識別子一覧
4		“exp” (Expiration Time)	ＪＷＴ　の有効期限
5		“nbf” (Not Before)	ＪＷＴ　が有効になる日時
6		“iat” (Issued At)	ＪＷＴ　を発行した時刻
7		“jti” (JWT aidhi)	ＪＷＴ　のための一意な識別子（アクセストークンＩＤ）
8	Private Claim	“authz:scopes”	アクセストークンスコープリスト
9		“authz:client_id”	アクセストークンクライアントＩＤ
10		“ext:fname”	ファーストネーム
11		“ext:lname”	ラストネーム
12		“ext:locale”	ロケール名
13		“ext:tenantid@recm”	テナントＩＤ
14		“ext:email@req”	電子メールアドレス
15		“ext:appid”	アプリケーションＩＤ

【００３８】

表１のクレーム名クラス"Ｒｅｇｉｓｔｅｒｅｄ　Ｃｌａｉｍ"の各クレームは、ＪＷＴのＲＦＣ７５１９で予め定義されたクレームで以下である。すなわち、ＪＷＴの発行者の識別子"ｉｓｓ"（Ｉｓｓｕｅｒ）、ＪＷＴの主語となる主体の識別子"ｓｕｂ"（Ｓｕｂｊｅｃｔ）、ＪＷＴを利用することが想定された主体の識別子一覧"ａｕｄ"（Ａｕｄｉｅｎｃｅ）、ＪＷＴの有効期限"ｅｘｐ"（Ｅｘｐｉｒａｔｉｏｎ　Ｔｉｍｅ）、ＪＷＴが有効になる日時"ｎｂｆ"（Ｎｏｔ　Ｂｅｆｏｒｅ）、ＪＷＴを発行した時刻"ｉａｔ"（Ｉｓｓｕｅｄ　Ａｔ）、ＪＷＴのための一意な識別子"ｊｔｉ"（ＪＷＴ　ａｉｄｈｉ）を表す。上記"ｅｘｐ"、"ｎｂｔ"、"ｉａｔ"に指定する日時はＩｎｔＤａｔｅで、１９７０－０１－０１Ｔ０：０：０Ｚ　ＵＴＣから指定されたＵＴＣの日付/時刻まで秒の数を表わすＪＳＯＮ数値である。またこれらの"Ｒｅｇｉｓｔｅｒｅｄ　Ｃｌａｉｍ"の使用は任意であ

る。
【００３９】
本実施形態においては、署名付きアクセストークンを発行する認証認可サーバー１０２が以下のように値を設定する。"ｉｓｓ"として認証認可サーバー１０２のＵＲＩ、"ｓｕｂ"としてユーザーのＵＵＩＤ（Ｕｎｉｖｅｒｓａｌｌｙ Ｕｎｉｑｕｅ Ｉｄｅｎｔｉｆｉｅｒ）、"ａｕｄ"としてリソースサーバー１０３のＵＲＩを設定する。また、"ｅｘｐ"として本ＪＷＴ発行時から３６００秒、すなわち"ｉａｔ"の値＋３６００、"ｎｂｆ"として本ＪＷＴ発行時、"すなわち"ｉａｔ"と同じ値を設定する。また、"ｊｔｉ"としてアクセストークン情報のアクセストークンＩＤを設定する。
【００４０】
さらには、表１のクレーム名クラス"Ｐｒｉｖａｔｅ Ｃｌａｉｍ"の各クレームは、ＪＷＴのＲＦＣ７５１９によると、ＪＷＴ発行者と利用者の合意のものとで使用するプライベートクレーム名クラスである。他に定義されたクレーム名と衝突のないことを前提とする。本実施形態では、"Ｐｒｉｖａｔｅ Ｃｌａｉｍ"クラスのクレーム名にアクセストークン情報（アクセストークンスコープリスト "ａｕｔｈｚ：ｓｃｏｐｅｓ"、アクセストークンクライアントＩＤ "ａｕｔｈｚ：ｃｌｉｅｎｔ＿ｉｄ"）と、アクセストークンに紐付く属性情報（ファーストネーム"ｅｘｔ：ｆｎａｍｅ"、ラストネーム"ｅｘｔ：ｌｎａｍｅ"、ロケール名"ｅｘｔ：ｌｏｃａｌｅ"、テナントＩＤ"ｅｘｔ：ｔｅｎａｎｔｉｄ", 電子メールアドレス"ｅｘｔ：ｅｍａｉｌ"、アプリケーションＩＤ"ｅｘｔ：ａｐｐｉｄ"）を持つことを特徴とする。
【００４１】
具体的に本実施形態においては、署名付きアクセストークンを発行する認証認可サーバー１０２が、認可情報として"ａｕｔｈｚ：ｓｃｏｐｅｓ"、"ａｕｔｈｚ：ｃｌｉｅｎｔ＿ｉｄ"のクレームを設定する。すなわち、"ａｕｔｈｚ：ｓｃｏｐｅｓ"としてリソースサーバー１０３が取得を許可するリソースを表すスコープリストを設定する。さらに、"ａｕｔｈｚ：ｃｌｉｅｎｔ＿ｉｄ"としてリソースサーバー１０３にアクセスするクライアントのＩＤを表す"ａｕｔｈｚ：ｃｌｉｅｎｔ＿ｉｄ"を設定する。また、署名付きアクセストークンを発行する認証認可サーバー１０２が、"ａｕｔｈｚ：ｔｏｋｅｎｅｄ"のトークンに紐付く属性情報として、"ｓｕｂ"に設定したＵＵＩＤのユーザーの情報を表すクレームを以下のように設定する。すなわち、"ｅｘｔ：ｆｎａｍｅ"としてのファーストネーム、"ｅｘｔ：ｌｎａｍｅ"としてラストネーム、"ｅｘｔ：ｌｏｃａｌｅ"としてユーザーが属するロケール情報、"ｅｘｔ：ｔｅｎａｎｔｉｄ"として所属するテナントＩＤ、"ｅｘｔ：ｅｍａｉｌ"として電子メールアドレスを設定する。"ｅｘｔ：ａｐｐｉｄ"としてリソースサーバー連携アプリケーション３３１を識別するためのアプリケーションＩＤを設定する。詳細は後述する。
【００４２】
本実施形態の上記表１で示されたような内容の署名付きアクセストークンを発行する認証認可サーバー１０２は、上記表１のクレームをＪＷＴの仕様であるＲＦＣ７５１９によりＪＳＯＮとしてエンコードする。また、ＪＷＳの仕様であるＲＦＣ７５１５のコンパクトシリアライゼーション仕様に従ってデジタル署名されたコンテンツ（表１のクレームのＪＳＯＮ表現、すなわちＪＷＳペイロード）をコンパクトなＵＲＬ－ｓａｆｅ文字列として表現符号化する。本実施例の署名付きアクセストークンは、ＪＷＳコンパクトシリアライゼーション仕様に従い、エンコード済ＪＷＳヘッダー、エンコード済ＪＷＳペイロード、およびエンコード済ＪＷＳ署名を、この順序でピリオド（'. '）文字を区切り文字として連結した文字列である。
【００４３】
本実施形態においては、ＪＷＳヘッダーとしてＪＷＳの署名に使われる暗号アルゴリズムを識別する"ａｌｇ"（アルゴリズム）を用いる。具体的に本実施形態においては、"ａｌｇ"として"ＲＳ２５６"（ＲＳＡＳＳＡ－ＰＫＣＳ１＿ｖ１＿５ ｕｓｉｎｇ ＳＨＡ－２５６）を用いる。"ＲＳ２５６"文字列は、ａｌｇの値としてＩＡＮＡ ＪＳＯＮ Ｗ

ｅｂ Ｓｉｇｎａｔｕｒｅ ａｎｄ Ｅｎｃｒｙｐｔｉｏｎ Ａｌｇｏｒｉｔｈｍｓレジストリに登録されており、ＪＳＯＮ Ｗｅｂ Ａｌｇｏｒｉｔｈｍｓ（ＪＷＡ）仕様（ＲＦＣ７５１８）のＳｅｃｔｉｏｎ ３．１で定義されている。

【００４４】

なお本実施形態においてＪＷＳの署名に使われる暗号アルゴリズム"ＲＳ２５６"で使用する秘密鍵、公開鍵の鍵ペアは、認証認可サーバー１０２が予め生成しておいたものを使用する。またＪＷＳの署名を検証する公開鍵は署名付きアクセストークンを使用するリソースサーバー１０３に予め配備しておく。

【００４５】

＜認証認可サーバーにより管理されるテーブル＞

図５、図６、図７及び表２から表８にて、本実施形態に係る署名なしアクセストークンおよび署名付きアクセストークンの発行と検証の流れについて説明する。表２から表６は、本実施形態において認証認可サーバー１０２の認証認可モジュール３１０が管理するテーブルである。

【００４６】

【表２】

表２　ユーザー管理テーブル

ユーザーID (クライアント ID)	UUID	Password	User Type
john.doe@170BA	1ce42f74	********	User
523ee20c9f39@2449baxt	241332ca	**********	Client

【００４７】

表２のユーザー管理テーブルは、ユーザーとクライアントを一意に管理するユーザーＩＤ（クライアントＩＤ）項目、ユーザーＩＤ（クライアントＩＤ）の内部表現であるＵＵＩＤ（Ｕｎｉｖｅｒｓａｌｌｙ Ｕｎｉｑｕｅ Ｉｄｅｎｔｉｆｉｅｒ）項目、前記ユーザーＩＤに対応するパスワードを示すＰａｓｓｗｏｒｄ項目、ユーザー種別を示すＵｓｅｒ Ｔｙｐｅ項目から成る。認証認可モジュール３１０は、ユーザーＩＤ（クライアントＩＤ）、Ｐａｓｓｗｏｒｄの情報の組を検証し、正しければ認証情報を生成することで、各ユーザーもしくはクライアントを認証する機能（不図示）を備える。

【００４８】

【表３】

表３　ユーザー属性管理テーブル

UUID	First Name	Last Name	Tenant ID	Email	Service ID
1ce42f74	John	Doe	170BA	john.doe@example.com	PrintService
241332ca	client	device	170BA	client.device@example.com	PrintService

【００４９】

表３のユーザー属性管理テーブルは、ＵＵＩＤ項目、前記ＵＵＩＤのユーザーの姓を示すＦｉｒｓｔ Ｎａｍｅ項目、名を示すＬａｓｔ Ｎａｍｅ項目、所属するテナントを示すＴｅｎａｎｔ ＩＤ項目、電子メールアドレスを示すＥｍａｉｌ項目、使用できるサービスを示すＳｅｒｖｉｃｅ ＩＤ項目から成る。

【００５０】

【表４】
表４　クライアント属性管理テーブル

UUID	デバイス シリアル番号	Redirect URL	Service ID
241332ca	12345678	http://client.example.com/redirect	PrintService

【００５１】
表４のクライアント属性管理テーブルは、ＵＵＩＤ項目、クライアントがどのデバイスに発行されたかを示すデバイスシリアル番号項目、ＯＡｕｔｈ２．０（ＲＦＣ６７４９）プロトコル等で使用する、同クライアントのリダイレクトＵＲＬを示すＲｅｄｉｒｅｃｔ　ＵＲＬ項目、同クライアントの使用できるサービスを示すＳｅｒｖｉｃｅ　ａｉｄｈｉ項目から成る。
【００５２】
【表５】
表５　サービス管理テーブル

Service ID	Scope	URL
PrintService	owner.PrintService	https://print.srv.example.com

【００５３】
表５のサービス管理テーブルは、リソースサーバー１０３で提供するリソースに関するサービスを示すＳｅｒｖｉｃｅ　ＩＤ項目、前記Ｓｅｒｖｉｃｅ　ＩＤに相当するサービスをリソースとして表し、ＯＡｕｔｈ２．０プロトコルなどの認可要求のスコープに指定される内容であるＳｃｏｐｅ項目、前記Ｓｅｒｖｉｃｅ　ＩＤに相当するサービス（リソース）を提供するリソースサーバーのＵＲＬを示すＵＲＬ項目から成る。
【００５４】
【表６】
表６　トークン管理テーブル

Token ID	Token Type	Expiration Time	Scopes	Grant Type
Aipzi	code	600		authorization code
b2652	access token	1472710413	owner.PrintService	Client Credentials

Refresh Token ID	Refresh Token Expiration Time	ClientUUID	OwnerUUID	ApplicationID
a2849	8640000	241332ca	1ce42f74	print

【００５５】
表６のトークン管理テーブルは、トークンＩＤを示すＴｏｋｅｎ　ＩＤ項目、アクセストークン、認可コードなど前記トークンＩＤのトークン種別を示すＴｏｋｅｎ　Ｔｙｐｅ項目、前記トークンＩＤの有効期限を秒で示すＥｘｐｉｒａｔｉｏｎ　Ｔｉｍｅ項目、ＯＡｕｔｈプロトコルなどの認可要求のスコープに指定される内容であるＳｃｏｐｅ項目、ＯＡｕｔｈプロトコルなどで使用する前記トークンＩＤのＧｒａｎｔ　Ｔｙｐｅを示すＧ

ｒａｎｔ　Ｔｙｐｅ項目、前記トークンＩＤのリフレッシュトークンのＩＤを示すＲｅｆｒｅｓｈ　Ｔｏｋｅｎ　ＩＤ項目、前記リフレッシュトークンＩＤの有効期限を秒で示すＲｅｆｒｅｓｈ　Ｔｏｋｅｎ　Ｅｘｐｉｒａｔｉｏｎ　Ｔｉｍｅ項目、前記トークンＩＤの発行要求元のクライアントを示すＣｌｉｅｎｔ　ＵＵＩＤ項目、前記トークンＩＤに紐付くオーナーを示すＯｗｎｅｒ　ＵＵＩＤ項目、前記トークンＩＤを用いるリソースサーバー連携アプリケーション３３１を示すＡｐｐｌｉｃａｔｉｏｎ　ＩＤ項目から成る。Ａｐｐｌｉｃａｔｉｏｎ　ＩＤ項目の値はリソースサーバー連携アプリケーション３３１毎に決定される。認証認可サーバー連携クライアント３３０がリソースサーバー連携アプリケーション３３１からのトークンの発行要求を行う際に自動的に取得され、認証認可サーバー１０２の認証認可モジュール３１０に通知される。

【００５６】

＜アクセストークンの発行と検証＞

図５は本実施例における、署名なしアクセストークンおよび署名付きアクセストークンの発行と検証の流れを示したシーケンス図である。署名なしアクセストークンおよび署名付きアクセストークンの発行と検証は、認証認可サーバー１０２、リソースサーバー１０３、クライアント端末１０４が連携することで実現される。なお、図中"Ｒｅｆ"は参照を示しており、詳細は別図で説明する。また"Ａｌｔ"は条件分岐処理を示し、いずれかの処理のみ実行される。

【００５７】

ステップＳ５０１において、クライアント端末１０４の認証認可サーバー連携クライアント３３０は、認証認可サーバー１０２にアクセストークンの発行を要求する。認証認可サーバー連携クライアント３３０からのアクセストークン発行要求には、認証認可サーバー１０２の認証認可モジュール３１０に対してＧｒａｎｔ　Ｔｙｐｅの指定とそれに応じた認証情報あるいは認可コードが含まれる。

【００５８】

ステップＳ５０２において、認証認可サーバー１０２の認証認可モジュール３１０は、署名付きアクセストークンの発行処理を実行する。ステップＳ５０２においては、Ｇｒａｎｔ　Ｔｙｐｅとして指定された値に応じて、署名付きアクセストークンまたは署名なしアクセストークンを発行する。詳細は図６で説明する。

【００５９】

ステップＳ５０３において、クライアント端末１０４のリソースサーバー連携アプリケーション３３１は、リソースサーバー１０３にサービスの提供を要求する。このとき、リソースサーバー連携アプリケーション３３１はステップＳ５０２で認証認可サーバー１０２から取得したアクセストークンをＨＴＴＰのＡｕｔｈｏｒｉｚａｔｉｏｎヘッダーに設定して、リソースサーバー１０３にサービスの提供を要求する。

【００６０】

ステップＳ５０４において、リソースサーバー１０３のリソースサーバーモジュール３２０は、サービス提供の要求とともに受信したアクセストークンが、署名付きアクセストークンか、それとも署名なしアクセストークンかを判定する。さらに署名付きアクセストークンであれば、リソースサーバー１０３における検証処理が許可されているか否かを判定する。この詳細は図８で説明する。

【００６１】

アクセストークンの検証処理をリソースサーバー１０３で実行すると判定された場合には、ステップＳ８０３において、リソースサーバーモジュール３２０はリソース要求に付与されている署名付きアクセストークンの検証を行う。

その場合、ステップＳ５０５において、リソースサーバーモジュール３２０はあらかじめ認証認可サーバー１０２から取得した公開鍵を使用して署名付きアクセストークンの署名を検証する。署名が適切なものであった場合にはステップＳ５０６に進む。

【００６２】

ステップＳ５０６において、リソースサーバーモジュール３２０は署名付きアクセスト

ークンを復号して、表７に示されるアクセストークンの属性情報を取得する。そして、リソースサーバーモジュール３２０は、属性情報を参照してステップＳ５０３でリソースを要求したクライアント端末１０４が、十分な権限を有しているかどうかを判定する。クライアント端末１０４が権限を有していると判断された場合には、ステップＳ５０９に進む。ステップＳ５０６の結果、クライアント端末１０４が要求するリソースに対して権限が不足していると判断された場合にはエラーを応答し、一連の処理を終了する。

【００６３】

ステップＳ５０９において、リソースサーバーモジュール３２０はステップＳ５０６の結果に基づきクライアント端末１０４が要求するリソースを提供するための処理を実行し、処理結果をクライアント端末１０４に応答する。

【００６４】

一方、アクセストークンの検証処理を認証認可サーバー１０２で実行すると判定された場合には、ステップＳ８０４において、リソースサーバーモジュール３２０はリソース要求に付与されているアクセストークンの種類にかかわらず、認証認可モジュール３１０にアクセストークンの検証を要求する。その場合、ステップＳ５０７において、認証認可モジュール３１０にアクセストークンの検証を要求する。ステップＳ５０８において認証認可サーバー５０２がアクセストークンの検証処理を行い、検証結果をリソースサーバー５０３に返す。なお、ステップＳ５０７およびステップＳ５０８における処理は、ステップＳ４０４およびステップＳ４０５に示した従来のアクセストークンの検証処理と同様のため、説明を省略する。

【００６５】

＜アクセストークン発行処理＞

図６は、認証認可モジュール３１０がステップＳ５０２において行うアクセストークン発行処理を示したフローチャートである。ステップＳ６０１において、認証認可モジュール３１０はアクセストークン発行要求を送信したアクセス元のクライアントやオーナーの認証、認可を行う。このとき、認証認可モジュール３１０はアクセストークン発行要求に設定された所定のパラメータ、たとえばＧｒａｎｔ　Ｔｙｐｅを確認し、それがＣｌｉｅｎｔ　Ｃｒｅｄｅｎｔｉａｌｓ　Ｇｒａｎｔタイプリクエストであるか否かを判定する。Ｃｌｉｅｎｔ　Ｃｒｅｄｅｎｔｉａｌｓ　Ｇｒａｎｔはたとえば、クライアントのクレデンシャルをサーバー（本例ではリソースサーバー）に渡して検証されるアクセストークンの要求時に設定されるパラメータである。たとえば処理を実施するオーナーがクライアント端末であるような場合に利用される。この要求に応じて発行されるアクセストークンには、クライアント認証のための情報であるクレデンシャルが含まれ、これが署名付きアクセストークンに相当する。

【００６６】

図１４は、実施形態において認証認可サーバー連携クライアント３３０が認証認可モジュール３１０に送信するアクセストークン発行要求のＨＴＴＰリクエストの一例を示した図である。認証認可サーバー連携クライアント３３０が認証認可モジュール３１０に送信するＨＴＴＰリクエストとして、Ｃｌｉｅｎｔ　Ｃｒｅｄｅｎｔｉａｌｓ　Ｇｒａｎｔ　Ｔｙｐｅリクエスト１４１０とＡｕｔｈｏｒｉｚａｔｉｏｎ　Ｃｏｄｅ　Ｇｒａｎｔ　Ｔｙｐｅリクエスト１４２０がある。Ｃｒｅｄｅｎｔｉａｌｓ　Ｇｒａｎｔ　Ｔｙｐｅリクエスト１４１０は、ＨＴＴＰリクエストヘッダーのＧｒａｎｔ　Ｔｙｐｅ項目１４１１の値に「ｃｌｉｅｎｔ＿ｃｒｅｄｅｎｔｉａｌｓ」を設定したリクエストである。

【００６７】

Ａｕｔｈｏｒｉｚａｔｉｏｎ　Ｃｏｄｅ　Ｇｒａｎｔ　Ｔｙｐｅリクエスト１４２０は、ＨＴＴＰリクエストヘッダーのＧｒａｎｔ　Ｔｙｐｅ項目１４２１の値に「ａｕｔｈｏｒｉｚａｔｉｏｎ＿ｃｏｄｅ」を設定したリクエストである。

【００６８】

なお、Ｇｒａｎｔ　Ｔｙｐｅは図１４に示す通り定義されており、認証認可サーバー連携クライアント３３０はそれらのＧｒａｎｔ　Ｔｙｐｅを送信してもよい。

【００６９】
認証認可モジュール３１０は、アクセストークン発行要求のＨＴＴＰリクエストヘッダーのＧｒａｎｔ　Ｔｙｐｅ項目の値を確認し、「ｃｌｉｅｎｔ＿ｃｒｅｄｅｎｔｉａｌｓ」だった場合には、ステップＳ６０２に進み、それ以外だった場合にはステップＳ６０３に進む。
【００７０】
ステップＳ６０２においては、認証認可モジュール３１０は、認証、認可に成功したクライアント情報に従って署名付きアクセストークンを発行し、認証認可サーバー連携クライアント３３０に送信する。認証認可モジュール３１０は、本実施形態の署名付きアクセストークンとして表１に示したクレームの値に関して、表２から表６のデータに基づき、具体的に以下の値を設定して発行処理を行う。
【００７１】
ＪＷＳヘッダーとして、"ａｌｇ":"ＲＳ２５６"、"ｔｙｐ":"ＪＷＴ"を設定する。さらにＪＷＳペイロードとして以下をそれぞれ設定する。ＪＷＴの発行者の識別子"ｉｓｓ"（Ｉｓｓｕｅｒ）として認証認可サーバー１０２のＵＲＩ "ｈｔｔｐｓ：／／ａｕｔｈ．ｅｘａｍｐｌｅ．ｃｏｍ"を設定する。ＪＷＴの主語となる主体の識別子"ｓｕｂ"（Ｓｕｂｊｅｃｔ）として前記アクセストークン発行要求に含まれる認可コードに対応するリソースオーナーのＩＤである表２のＵＵＩＤ"２４１３３２ｃａ"を設定する。
【００７２】
ＪＷＴを利用することが想定された主体の識別子一覧"ａｕｄ"（Ａｕｄｉｅｎｃｅ）としてＳｃｏｐｅに相当するリソースサーバーのＵＲＬとして表５のＵＲＬ項目の"ｈｔｔｐｓ：／／ｐｒｉｎｔ．ｓｒｖ．ｅｘａｍｐｌｅ．ｃｏｍ"を設定する。ＪＷＴの有効期限"ｅｘｐ"（Ｅｘｐｉｒａｔｉｏｎ　Ｔｉｍｅ）として署名付きアクセストークンの発行時から３６００秒、すなわち本実施例の場合"ｉａｔ"の値＋３６００である"１４７２７１３４１３"を設定する。"ｎｂｆ"として署名付きアクセストークン発行時、すなわち"ｉａｔ"と同じ値"１４７２７０９８１３"を設定する。また本実施例においては、署名付きアクセストークンの発行時に表６のトークン管理テーブルに示すように、"ｊｔｉ"として該署名付きアクセストークンのトークンＩＤ"ｂ２６５２"を設定する。
【００７３】
アクセストークンスコープリスト"ａｕｔｈｚ：ｓｃｏｐｅｓ"には、"ｏｗｎｅｒ．ＰｒｉｎｔＳｅｒｖｉｃｅ"を設定する。アクセストークンクライアントＩＤ"ａｕｔｈｚ：ｃｌｉｅｎｔ＿ｉｄ"には、"２４１３３２ｃａ"を設定する。
【００７４】
さらには、アクセストークンに紐付く属性情報をとして、表２から表５までのそれぞれの値を格納する。"ｅｘｔ：ｆｎａｍｅ"、"ｅｘｔ：ｌｎａｍｅ"、"ｅｘｔ：ｅｍａｉｌ"属性には表３のユーザー属性管理テーブルよりＦｉｒｓｔ　Ｎａｍｅ項目、Ｌａｓｔ　Ｎａｍｅ項目、Ｅｍａｉｌ項目の値を格納する。それぞれ"Ｃｌｉｅｎｔ"、"Ｄｅｖｉｃｅ","ｃｌｉｅｎｔ．ｄｅｖｉｃｅ@ｅｘａｍｐｌｅ．ｃｏｍ"の値を格納する。
【００７５】
同様に"ｅｘｔ：ｔｅｎａｎｔｉｄ"属性には表３のユーザー属性管理テーブルよりテナントＩＤ項目の"１７０ＢＡ"の値を格納する。"ｅｘｔ：ｄｅｖ－ｓｅｒｉａｌ"属性には、表４のクライアント属性管理テーブルよりデバイスシリアル番号項目の"１２３４５６７８"を格納する。"ｅｘｔ：ａｐｐｉｄ"属性には、トークン発行要求元である、リソースサーバー連携アプリケーション３３１の識別子である"ｐｒｉｎｔ"を格納する。
【００７６】
まとめると、ステップＳ６０２によって署名付きアクセストークンのＪＷＳペイロードは以下のように指定される。
【００７７】

【表７】
表７　本実施例のＪＷＳペイロード指定値

	クレーム名クラス	クレーム名	クレーム指定値
1	Registered Claim	“iss” (Issuer)	https://auth.example.com
2		“sub” (Subject)	241332ca
3		“aud” (Audience)	https://print.srv.example.com
4		“exp” (Expiration Time)	1472713413
5		“nbf” (Not Before)	1472709813
6		“iat” (Issued At)	1472709813
7		“jti” (JWT ID)	b2652
8	Private Claim	“authz:scopes”	owner.PrintService
9		“authz:client_id”	241332ca
10		“ext:fname”	Client
11		“ext:lname”	Device
12		“ext:locale”	en_US.utf-8
13		“ext:tenantid@recm”	170BA
14		“ext:email@req”	client.device@example.com
15		“ext:appid”	print

【００７８】
さらにＪＷＳの仕様であるＲＦＣ７５１５のコンパクトシリアライゼーション仕様に従い、デジタル署名された前記ＪＷＳペイロードをコンパクトなＵＲＬ－ｓａｆｅ文字列として表現符号化（ＢＡＳＥ６４　ＵＲＬエンコード）する。そして、エンコード済前記ＪＷＳヘッダー、エンコード済ＪＷＳペイロード、およびエンコード済ＪＷＳ署名を、この順序でピリオド（'．'）文字を区切り文字として連結した文字列にして、認証認可モジュール３１０は図７に示すような署名付きアクセストークンを生成する。
【００７９】
図７は本実施形態における認証認可モジュール３１０が生成する署名付きアクセストークンの一例を示した図である。クライアント端末１０４は、リソースサーバー１０３に対しサービスの提供を要求する際にはＨＴＴＰのＡｕｔｈｏｒｉｚａｔｉｏｎヘッダーに図７に示す署名付きアクセストークンを設定する。
【００８０】
一方ステップＳ６０１でＣｌｉｅｎｔ　Ｃｒｅｄｅｎｔｉａｌｓ　Ｇｒａｎｔタイプではないと判定されると、ステップＳ６０３において、認証認可モジュール３１０はアクセストークン発行要求のＧｒａｎｔ　ＴｙｐｅがＡｕｔｈｏｒｉｚａｔｉｏｎ　Ｃｏｄｅ　Ｇｒａｎｔか否かを判定する。Ｇｒａｎｔ　ＴｙｐｅがＡｕｔｈｏｒｉｚａｔｉｏｎ　Ｃｏｄｅ　Ｇｒａｎｔだった場合にはステップＳ６０４に進み、署名無しアクセストークンを認証認可サーバー連携クライアント３３０に発行する。本ステップで発行されるアクセストークンはステップＳ４０２で発行される形式と同様のものである。
【００８１】
認証認可モジュール３１０は、トークン発行要求のＧｒａｎｔ　ＴｙｐｅがＣｌｉｅｎｔ　Ｃｒｅｄｅｎｔｉａｌｓ　Ｇｒａｎｔ、Ａｕｔｈｏｒｉｚａｔｉｏｎ　Ｃｏｄｅ　Ｇｒａｎｔのどちらでもない場合には、ステップＳ６０５に進みエラーレスポンスを認証認可サーバー連携クライアント３３０に返す。具体的には、認証認可モジュール３１０が独自に定義するＧｒａｎｔ　ＴｙｐｅであるＥｘｔｅｎｓｉｏｎ　Ｇｒａｎｔが該当する。
【００８２】
＜認可確認処理＞

図８は、リソースサーバーモジュール３２０がステップＳ５０４において行う認可確認処理を示したフローチャートである。なお図示の都合上、ステップＳ８０３とＳ８０４は、図６と重複して示した。

【００８３】

ステップＳ８０１において、リソースサーバーモジュール３２０はリソースサーバー連携アプリケーション３３１から受信したアクセストークンが署名付きアクセストークンか確認する。アクセストークンはＨＴＴＰリクエストヘッダーに設定され、リソースサーバーモジュール３２０はその内容を確認する。図７に示すような署名付きアクセストークンの場合はステップＳ８０２に進む。リソースサーバー連携アプリケーション３３１から受信したアクセストークンが署名無しアクセストークンだった場合にはステップＳ８０４に進み、リソースサーバーモジュール３２０は認証認可モジュール３１０にトークン検証要求を行う。

【００８４】

ステップＳ８０２において、リソースサーバーモジュール３２０は、要求を受けたリソースがリソースサーバー１０３自身でアクセストークンの検証をして良いリソースか、認証認可サーバーモジュール３１０でアクセストークンの検証をすべきリソースかを、認可確認テーブル（表８に示す）に基づいて判定する。その判定結果に応じて、要求を受けたリソースがリソースサーバー自身で検証して良いリソースであった場合には、ステップＳ８０３に進み、事前に認証認可サーバーから取得した公開鍵を使用して署名付きアクセストークンの署名を検証する。要求を受けたリソースが認証認可モジュール３１０でトークンの検証をすべきリソースであった場合にはステップＳ８０４に進み、リソースサーバーモジュール３２０は認証認可モジュール３１０にトークン検証要求を行う。

【００８５】

表８は、本実施形態においてリソースサーバー１０３のリソースサーバーモジュール３２０が管理し、ステップＳ８０２で参照する認証認可テーブルである。

【００８６】

【表８】

表８　認可確認テーブル

Service ID	Resource Name	署名付きアクセストークン許可情報
PrintService	Resource1	許可
PrintService	Resource2	禁止

【００８７】

表８の認可確認テーブルはリソースサーバー１０３が提供するサービスを示すＳｅｒｖｉｃｅ　ＩＤ項目、リソースサーバー１０３が提供するリソース名を示すＲｅｓｏｕｒｃｅ　Ｎａｍｅ項目、リソースに関連づけて、そのリソース要求時に署名付きアクセストークンを許可するかを示す署名付きアクセストークン許可情報項目から成る。この認可確認テーブルは、たとえばサービス及びリソースごとに予め設定される。リソースサーバーモジュール３２０が要求を受け付けたリソースの署名付きアクセストークン許可情報項目の値が「許可」の場合はステップＳ８０３に進み、「禁止」の場合はステップＳ８０４に進むことになる。ステップＳ８０３、Ｓ８０４の内容は図５で説明したとおりである。

【００８８】

例えばリソースサーバー１０３が提供するリソースの中で管理者権限が必要なリソースの場合、要求を受け付けた今まさに権限を有しているか、を確認すべきリソースである可能性がある。署名付きアクセストークンは発行するタイミングで権限を有していれば、リソース要求時点で権限を失っていても、有効期限の範囲内であればリソースを提供してしまうリスクがある仕組みである。このような判断は署名付きアクセストークンに設定する有効期限と、リソース提供に必要な権限をどこまで厳密に管理すべきか、によって決定される。例えば、権限そのものが有償で、権限を付与できる数に制限があるようなケースにおいて、署名付きアクセストークンの有効期限が十分長いと、署名付きアクセストークン

発行後に権限を外し、別のユーザーに権限を付与してから署名付きトークンを発行し、という処理を繰り返す事で、権限を付与できる数の制限が形骸化する事も考えうる。このような場合にはアクセストークンの検証は、発行時ではなく、検証時の権限の設定に基づいて検証を行う署名なしアクセストークンとするのが望ましい。したがってこのような場合には署名付きアクセストークン許可情報を「禁止」に設定しておく。表８の認可確認テーブルは、このような条件を考慮して、リソースごとに設定される。

【００８９】

以上の一連の処理によってリソースサーバー１０３は、クライアント端末１０４からのすべてのリソース要求に対して認証認可サーバー１０２にトークンの検証を依頼する必要をなくすことができる。その結果、認証認可サーバー１０２の負荷が下がることになる。さらにステップＳ５０４における認可確認処理によって、署名付きアクセストークンによるリソースサーバー１０３自身でのトークン検証を許容するリソースを特定することで、リソースサーバー１０３が提供するリソースのキャパシティを柔軟に維持することができる。

【００９０】

以上より、認証認可サーバー１０２およびリソースサーバー１０３のそれぞれにアクセストークンの検証処理を振り分けることが可能となる。それにより、検証処理要求の認証認可サーバーへの集中を防止することができる。さらに、リソースサーバーで検証処理を行うことで検証処理のレイテンシを小さくすることができる。さらに、署名付きアクセストークンの発行に適していない権限やリソースについては、署名なしアクセストークンを発行させる。それによって、認証認可サーバーにより、アクセストークンの発行時ではなく、検証時の権限等に応じてアクセストークンを検証させることができる。さらにリソースを要求する主体に応じて柔軟なリソースの提供が可能となる。

【００９１】

さらに本実施形態では、アクセストークンの検証をリソースサーバーで行うか、それとも認証認可サーバーで行うかを、クライアントがアクセストークン発行要求にセットするパラメータによりコントロールできる。このため、たとえば、アクセストークン発行後に、登録済みのクライアントやユーザーが認証認可サーバーから削除されたり、権限や属性が変更されたりする可能性が高い場合には、クライアントは署名なしアクセストークンの発行を要求すればよい。反対にたとえば、アクセストークン発行後のクライアントやユーザーの削除や、権限や属性の変更の可能性が低い場合には、クライアントは署名付きアクセストークンの発行を要求すればよい。このように、クライアントに、要求するアクセストークンを署名付きとするか、それとも署名なしとするかを決定する基準を用意しておくことで、アクセストークンの検証を行うサーバーを選択することができる。さらに、クライアントの選択を、認証認可サーバー１０２の認可確認テーブルによってオーバーライドすることもできるので、クライアントによる恣意的な選択のみに依存することを防止できる。

［実施形態２］

実施形態１において、認証認可サーバー１０２における署名付きアクセストークンの発行条件およびリソースサーバー１０３における署名付きアクセストークンの検証実行条件を定めることで、リソースを適切に管理する方法を説明した。前述したように署名付きアクセストークンの制約として、トークンに紐付く今現在の情報をトークン検証に反映することができない、というものがある。リソースサーバー１０３が提供するリソースによっては、やはり署名付きアクセストークンによって権限が制御されるリソースであっても、トークン検証時に今現在の情報を必要とするようなリソースが求められる場合がある。

【００９２】

例えば、認証認可サーバー１０２で発行済みの署名付きアクセストークンの削除処理が実行された場合に、リソースによっては署名付きアクセストークンの有効期間内であれば参照だけは許可するが、更新は許可しないといった制御を行うことが想定される。また、

情報漏えいなどにより特定のクライアント端末からのリソース要求を遮断するために、認証認可サーバー１０２に登録されているクライアント端末情報を削除することも想定される。

【００９３】

実施形態２は、上記のように署名付きアクセストークンによって権限が制御されているリソースにおいて、アクセストークンの検証時にアクセストークンの削除とクライアントの削除とを確認して、適切な制御を実現する。本実施形態に係る情報処理システムである認証認可システムは、図９に示すような構成のネットワーク上に実現される。本実施形態の構成は、実施形態１で示した図１の構成に削除トークン管理サーバー９０１と、削除クライアント管理サーバー９０２とを加えた構成となる。

【００９４】

削除トークン管理サーバー９０１は、認証認可サーバー１０２で管理するアクセストークンの中で、削除されたアクセストークンの情報を管理および提供するサーバーである。

【００９５】

削除クライアント管理サーバー９０２は、認証認可サーバー１０３で管理するクライアント情報の中で、削除されたクライアント情報を管理および提供するサーバーである。また、各サーバーは１台ずつ設置されているが、それぞれが複数台で構成されていても良い。また、各サーバーはＬＡＮ１０１を介して接続されているがＷＡＮ１００を介して接続することも可能である。また、削除トークン管理サーバー９０１と削除クライアント管理サーバー９０２は不図示のデータベースサーバーとＬＡＮを介して接続し、後述する各サーバーモジュールが利用するデータを格納するように構成しても良い。さらには削除トークン管理サーバー９０１と削除クライアント管理サーバー９０２を同一のサーバー上に構成しても良い。

【００９６】

図１０は、本実施形態に係る削除トークン管理サーバー９０１、削除クライアント管理サーバー９０２それぞれのソフトウェアモジュール構成を示した図である。各々のモジュールが各構成要素で実行されることで、各々の機能を実現する。削除トークン管理サーバー９０１は、削除トークン管理モジュール１０１０を持つ。削除トークン管理モジュール１０１０では、認証認可サーバー１０２からの削除済みのアクセストークンのトークンＩＤの受信処理、リソースサーバー１０４に対して削除済みアクセストークンのトークンＩＤの送信処理を行う。

【００９７】

削除クライアント管理サーバー９０２は、削除クライアント管理モジュール１０２０を持つ。削除クライアント管理モジュール１０２０では、認証認可サーバー１０２からの削除済みのクライアントのクライアントＩＤの受信処理、リソースサーバー１０４に対して削除済みクライアントのクライアントＩＤの送信処理を行う。

【００９８】

＜発行済みアクセストークンの削除と登録済みクライアント情報の削除＞

図１１は、認証認可サーバー１０３での発行済みアクセストークンの削除と登録済みクライアント情報の削除の流れを示したシーケンス図である。発行済みのアクセストークンの削除および登録済みクライアントの削除は、認証認可サーバー１０２、削除トークン管理サーバー９０１、削除クライアント管理サーバー９０２が連携することで実現される。なお、発行済みアクセストークンの削除処理及びクライアント情報の削除処理は認証認可サーバー１０２において管理者権限を所有する限られたユーザーのみ実行できる処理とする。

【００９９】

ステップＳ１１０１において、認証認可サーバー１０２の管理者ユーザーがクライアント端末１０４を介して認証認可サーバー１０２に、削除対象のトークンＩＤを指定してアクセストークン削除要求を送信する。

【０１００】

ステップＳ１１０２において、認証認可サーバー１０２の認証認可モジュール３１０はステップＳ１１０１においてクライアント端末１０４から受信したトークンＩＤに該当するアクセストークン情報を表６のトークン管理テーブルから削除する。

【０１０１】

ステップＳ１１０３において、認証認可モジュール３１０は、ステップＳ１１０２において削除に成功したアクセストークン情報のトークンＩＤを削除トークン管理サーバー９０１に送信する。

【０１０２】

ステップＳ１１０４において、削除トークン管理サーバー９０１の削除トークン管理モジュール１０１０は、認証認可モジュール３１０から受信したトークンＩＤを削除済みアクセストークンＩＤ管理テーブルに登録する。

【０１０３】

表９は、本実施例において削除トークン管理サーバー９０１の削除トークン管理モジュール１０１０が管理する削除済みアクセストークンＩＤ管理テーブルである。

【０１０４】

【表９】

表９　削除済みアクセストークンＩＤ管理テーブル

Ｔｏｋｅｎ　ＩＤ	Ｅｘｐｉｒａｔｉｏｎ　Ｄａｔｅ
ｂ２６５２	２０１７－０１－０１　Ｔ１２：００Ｚ

【０１０５】

表９の削除済みアクセストークンＩＤ管理テーブルは認証認可サーバー１０２で削除が完了したアクセストークンのトークンＩＤを示すＴｏｋｅｎ　ＩＤ項目、認証認可サーバー１０２でアクセストークンの削除が完了した時刻を示すＥｘｐｉｒａｔｉｏｎ　Ｄａｔｅ項目からなる。ここまででアクセストークン削除処理が完了する。ステップＳ１１０５以降はクライアントの削除処理であり、アクセストークンの削除処理とは本来別の独立した処理である。

【０１０６】

ステップＳ１１０５において、認証認可サーバー１０２の管理者ユーザーがクライアント端末１０４を介して認証認可サーバー１０２に、削除対象のクライアントＩＤを指定してクライアント削除要求を送信する。

【０１０７】

ステップＳ１１０６において、認証認可サーバー１０２の認証モジュール３１０はステップＳ１１０５においてクライアント端末１０４から受信したクライアントＩＤに該当するクライアント情報を表２のユーザー管理テーブルから削除する。また、削除したクライアント情報のＵＵＩＤ項目の値に該当するクライアント属性情報を表４のクライアント属性管理テーブルから削除する。

【０１０８】

ステップＳ１１０７において、認証認可モジュール３１０は、ステップＳ１１０６において削除に成功したクライアント情報のクライアントＩＤを削除クライアント管理サーバー９０２に送信する。

【０１０９】

ステップＳ１１０８において、削除クライアント管理サーバー９０２の削除クライアント管理モジュール１０２０は、認証認可モジュール３１０から受信したクライアントＩＤを削除済みクライアントＩＤ管理テーブルに登録する。

【０１１０】

表１０は、本実施例において削除クライアント管理サーバー９０２の削除クライアント管理モジュール１０２０が管理する　削除済みクライアントＩＤ管理テーブルである。

【０１１１】

【表１０】
表１０　削除済みクライアントＩＤ管理テーブル

Client ID	Expiration Date
523ee20c9f39@2449baxt	2017-01-01 T12:00Z

【０１１２】
表１０の削除済みクライアントＩＤ管理テーブルは認証認可サーバー１０２で削除が完了したクライアントのクライアントＩＤを示すＣｌｉｅｎｔ　ＩＤ項目、認証認可サーバー１０２でクライアントの削除が完了した時刻を示すＥｘｐｉｒａｔｉｏｎ　Ｄａｔｅ項目からなる。
【０１１３】
一度発行したアクセストークンの有効期限が切れる前にアクセストークンを削除する状況とは、例えばテナントに属するユーザーが削除された場合などが考えられる。一方で、クライアントを削除する状況は、例えばクライアントの認証認可サーバー連携クライアント３３０が不正に流出するなどセキュリティ問題が発生した場合に、以降そのクライアントからの通信を遮断する場合などが挙げられる。
【０１１４】
＜署名付きアクセストークンの発行と検証＞
図１２は、本実施例における署名付きアクセストークンの発行と検証の流れを示したシーケンス図である。この中で、リソースサーバー１０３においてトークンの検証時にクライアント端末１０４から要求されたリソースに紐付くトークン及びクライアントが削除されているかどうかを判断したうえでトークンの検証を行う一連の流れを示す。本実施形態におけるトークン及びクライアントの削除確認処理は、実施形態１の図５に示したシーケンスにおいてステップＳ５０６の後に実行される。また、トークンの発行及び検証処理は実施形態１に示した図５の内容と同様である。
【０１１５】
ステップＳ１２０１において、リソースサーバー１０３のリソースサーバーモジュール３２０は、アクセストークン及びクライアントの削除確認処理を実行する。詳細は図１３で説明する。
【０１１６】
＜削除確認処理＞
図１３は、リソースサーバーモジュール３２０がステップＳ１２０１において行う削除確認処理を示したフローチャートである。
【０１１７】
ステップＳ１３０１において、リソースサーバーモジュール３２０は、受信した処理要求の要求元であるクライアント端末１０４の認証認可サーバー連携クライアント３３０が認証認可サーバー１０２で削除されているかどうかを判定する。具体的には、リソースサーバーモジュール３３０は、削除クライアント管理サーバー９０２の表１０に示した削除済みクライアントＩＤ管理テーブルに、処理要求の元である認証認可サーバー連携クライアント３３０のクライアントＩＤが登録されているかどうかを判定する。該当するクライアントＩＤが登録されていた場合には、リソースサーバー１０３は、処理要求を送信したクライアント端末１０４が認証認可サーバー１０２から削除されたものと判断し、ステップＳ１３０４に進みエラーレスポンスを生成して、クライアント端末１０４に送信する。すなわち、この場合にはリソースの要求を拒絶する。該当するクライアントＩＤが登録されていなかった場合には、リソースサーバー１０３は、処理要求を送信したクライアント端末１０４が有効であると判断し、ステップＳ１３０２に進む。
【０１１８】
ステップＳ１３０２において、リソースサーバーモジュール３２０は、受信した署名付

きアクセストークンが認証認可サーバー１０２で削除されているかどうかを判定する。具体的には、リソースサーバーモジュール３３０は、削除トークン管理サーバー９０１の表９に示した削除済みトークンＩＤ管理テーブルに、受信した署名付きアクセストークンのトークンＩＤが登録されているかどうかを判定する。トークンＩＤが登録されていた場合には、リソースサーバーモジュール３３０は、受信した署名付きアクセストークンが認証認可サーバー１０２から削除されたものと判断し、ステップＳ１３０３に進む。

【０１１９】

クライアントＩＤが登録されていなかった場合には、リソースサーバーモジュール３３０は、受信した署名付きアクセストークンが有効であると判断し、ステップＳ５０９に進み処理を実行する。

【０１２０】

ステップＳ１３０３において、リソースサーバーモジュール３３０は、処理要求対象のリソースが、署名付きアクセストークンが認証認可サーバー１０２で削除されていても実行可能なリソースか否かを判定する。

【０１２１】

表１１は、本実施形態においてリソースサーバーモジュール３３０が管理するテーブルである。

【０１２２】

【表１１】

表１１　実行許容リソース管理テーブル

Service ID	Resource Name	実行許容情報	許容処理
PrintService	Resource1	禁止	-
PrintService	Resource2	禁止	-
PrintService	Resource3	許可	Read

【０１２３】

表１１の実行許容リソース管理テーブルはリソースサーバー１０３が提供するサービスを示すＳｅｒｖｉｃｅ　ＩＤ項目、リソースサーバー１０３が提供するリソース名を示すＲｅｓｏｕｒｃｅ　Ｎａｍｅ項目、リソースに関連づけて、そのリソース要求時に署名付きアクセストークンが削除されていても処理の実行を許可するかを示す実行許容情報項目、処理の実行が許可されている場合の具体的な処理種別を示す許容処理項目から成る。

【０１２４】

表１１を例にするとＰｒｉｎｔＳｅｒｖｉｃｅサービスのＲｅｓｏｕｒｃ３リソースに対するＲｅａｄ処理に関しては、仮に署名付きアクセストークンが処理要求時に削除されていても有効期限内であれば実行を許可することを意味する。

【０１２５】

リソースサーバーモジュール３３０は、処理要求対象が実行許容リソース管理テーブル上で許可されているとステップＳ１３０３で判定した場合には、署名付きアクセストークンが認証認可サーバー１０２で削除されていても、図５の処理をそのまま終了する。したがってステップＳ５０９に進み要求された処理を実行する。一方、処理要求対象が実行許容リソース管理テーブル上で禁止されているとステップＳ１３０３で判定した場合にはステップＳ１３０５に進みエラーレスポンスを生成して、クライアント端末１０４に送信する。

【０１２６】

ステップＳ１３０４において、リソースサーバーモジュール３３０はクライアント端末１０４の認証認可サーバー連携クライアント３３０に該当するクライアント情報が認証認可サーバー１０２から削除されていることを示すエラーメッセージを含んだレスポンスを作成して、クライアント端末１０４に送信する。

【０１２７】

ステップＳ１３０５において、リソースサーバーモジュール３３０はリソースサーバー連携アプリケーション３３１から受信した署名付きアクセストークンが認証認可サーバー１０２から削除されていることを示すエラーメッセージを含んだレスポンスを作成してクライアント端末１０４に送信する。

【０１２８】

以上より、リソースサーバー１０３がトークン及びクライアントの削除確認のために外部サーバーである削除トークン管理サーバー９０１及び削除クライアント管理サーバー９０２に問い合わせを行うことにはなるが、認証認可サーバー１０２に対する負荷を低減させることができる。そして、署名付きアクセストークンの制約であったトークン検証時のアクセストークン及びクライアントの状態を確認することができることで、リソースサーバーは柔軟にリソースをクライアント端末１０４に対して提供することができる。

【０１２９】

このため、認証認可サーバー１０２が管理する発行済みのアクセストークンあるいはクライアントが削除された場合にも、すでに発行済みのアクセストークンの使用を制限したり、あるいは権限を制限した上で、あるいは制限なしに使用を許可したりといった制御ができる。これにより、署名付きアクセストークンの発行後にも、その発行時に登録されていたアクセストークンやクライアントの削除が生じる可能性があっても、署名付きアクセストークンを発行させてリソースサーバーによる検証を行わせることができる。このため、認証認可サーバー１０２の負荷をより軽減することができる。

【０１３０】

［その他の実施例］

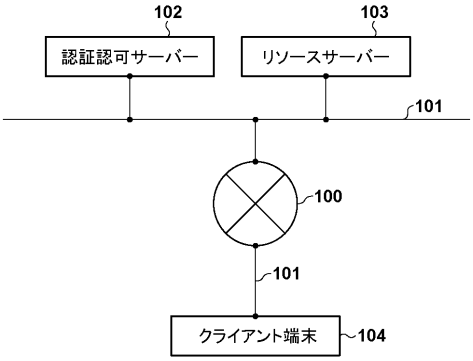
本発明は、上述の実施形態の１以上の機能を実現するプログラムを、ネットワーク又は記憶媒体を介してシステム又は装置に供給し、そのシステム又は装置のコンピュータにおける１つ以上のプロセッサーがプログラムを読出し実行する処理でも実現可能である。また、１以上の機能を実現する回路（例えば、ＡＳＩＣ）によっても実現可能である。

【符号の説明】

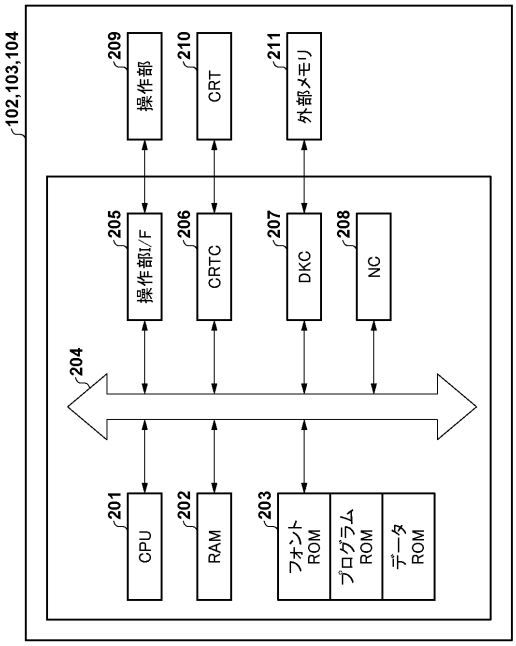
【０１３１】

１０２　認証認可サーバー、１０３　リソースサーバー、１０４　クライアント端末

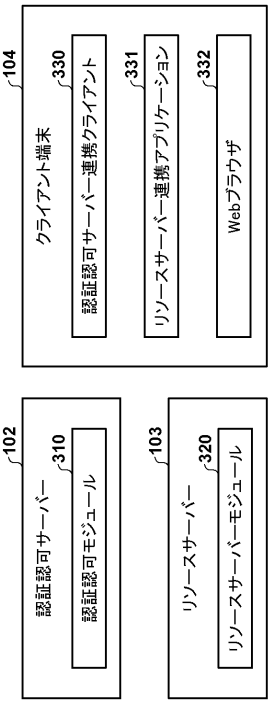
【図１】


102
103
認証認可サーバー
リソースサーバー
101
100
101
クライアント端末
104

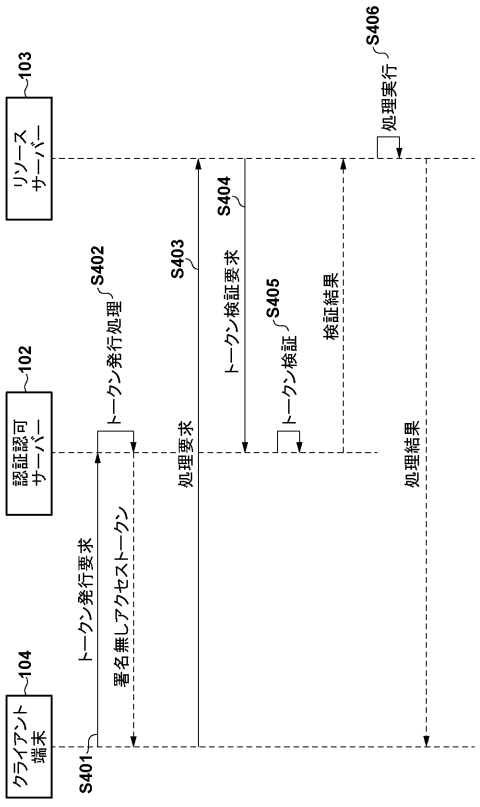
【図２】


102,103,104
209
操作部
210
CRT
211
外部メモリ
205
操作部I/F
206
CRTC
207
DKC
208
NC
204
201
CPU
202
RAM
203
フォント ROM
プログラム ROM
データ ROM

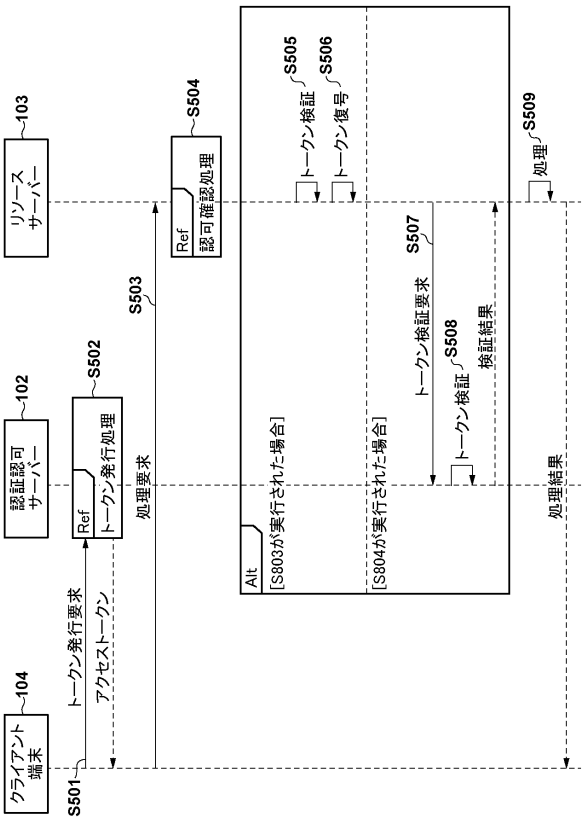
【図３】


104
クライアント端末
330
認証認可サーバー連携クライアント
331
リソースサーバー連携アプリケーション
332
Webブラウザ
102
認証認可サーバー
310
認証認可モジュール
103
リソースサーバー
320
リソースサーバーモジュール

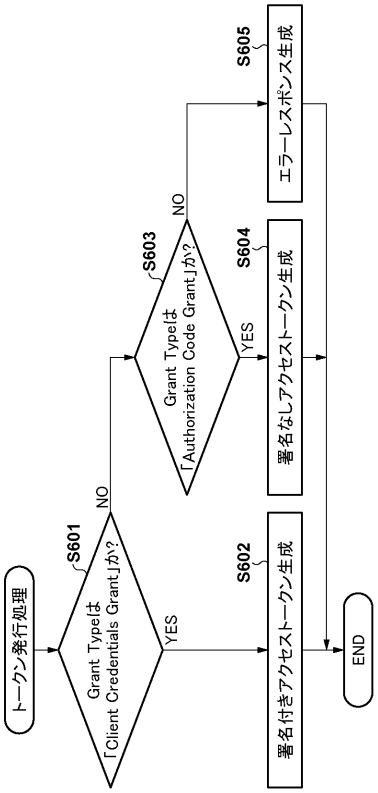
【図４】


103
リソースサーバー
S406
処理実行
S404
トークン検証要求
S403
S402
トークン発行処理
102
認証認可サーバー
S405
トークン検証
検証結果
処理要求
処理結果
トークン発行要求
署名無しアクセストークン
104
クライアント端末
S401

【図５】



【図６】



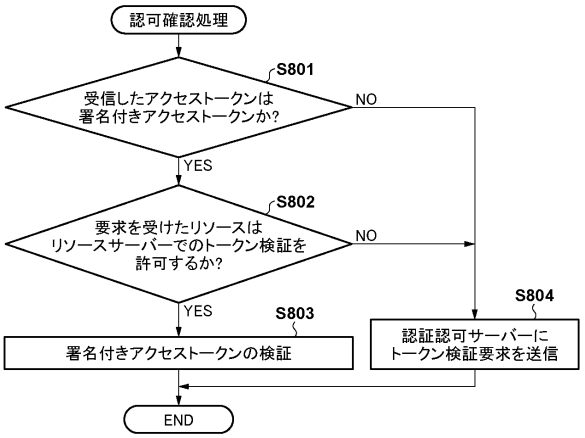
【図７】

700

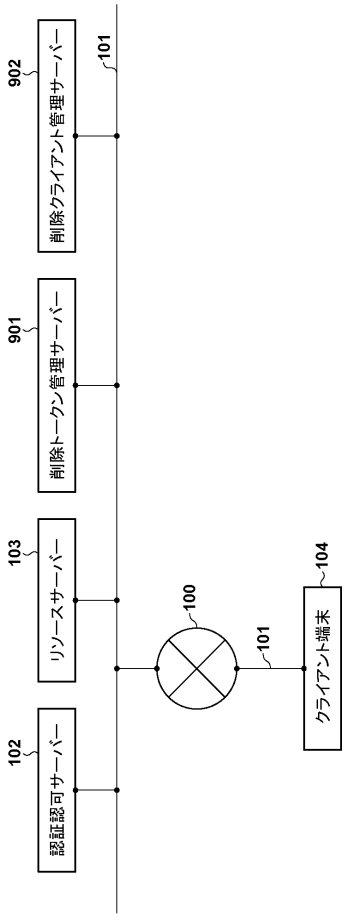
署名付きアクセストークン 例

eyJraWQiOiJrMSIsImFsZyI6IlJTMjU2In0.eyJpc3MiOiJodHRwczovL2F1dGgtc2VydmljZS5leGFtcGxlLmNvbSIsImF1ZCI6Imh0dHBzOi8vcHJpbnQtc2VydmljZS5leGFtcGxlLmNvbSIsImV4cCI6MTQ3Mzc0NzIxNSwianRpIjoiYjI2NTJmNmQ3NWIxNGFhNDIhZGZlODliODgwOGIzMzdlZjI3NmU4ZDEzNjM0MDBkOThmYWIxMDBjNzQ2M2RmMCIsImlhdCI6MTQ3Mzc0NjYxNSwibmJmIjoxNDczNzQ2NDk1LCJzdWIiOiIxY2U0MmY3NC0xMjI1LTRjYmItYjIzZi1mNTI1ZGFiYzNjZmQiLCJhdXRoejpzY29wZXMiOlsiY2xpZW50LkZvcm1DbGllbnQiLCJjbGllbnQuUHJpbnRDbGllbnQiXSwiZXh0OmZuYW1lIjoiSm9obiIsImV4dDpsbmFtZSI6IkRvZSIsImV4dDpsb2NhbGUiOiJlbl9VUy51dGYtOCIsImV4dDp0ZW5hbnRpZEByZWNtjoiMTcwQkEiLCJleHQ6ZW1haWxAcmVxIjoiam9obi5kb2VAZXhhbXBsZS5jb20iLCJleHQ6YXBwaWQiOiI0MmM1YTk2MjYwOTg0NTllOTZkNGJhY2Q3M2MyOTgyMzM5MmE3NmEwMmNjOTRkMzY5ODViN2RjMGQ3OTBiOWE0In0.O4TpYbDBFq0g20TGZwzZH97Cn80NtdcvutHDBlwUsOX1T_7nOCJNJq9F3KdFgCYApPFtzpQqhUfdZGgrKmJC8iBYAqWj7CExS9Qw4WITeZoawBXuGq8KC80PGES7uFKRPDPM1sDggjblt_nGWh4QgrwrUxLkH8SFOgShaYtWMpXxccPR2GavIOLXqb4ZgXbhmxItrcOzX0n81qc24EJeHe-8wGAICMk4YRdjfBnwUWA9be1oOUYI_NGXM2G7XdEPzA5ydtlZ69kWHZzCsPpmaIHl83EmxMHlu7Hde7DREHehRiltm4A_rb1uN_wMjUB4887qt95cVIOeJbPyYk9YhA

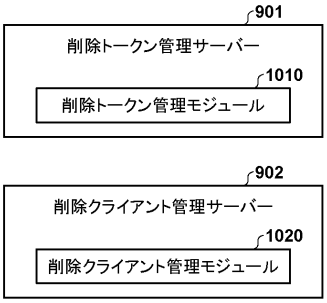
【図８】



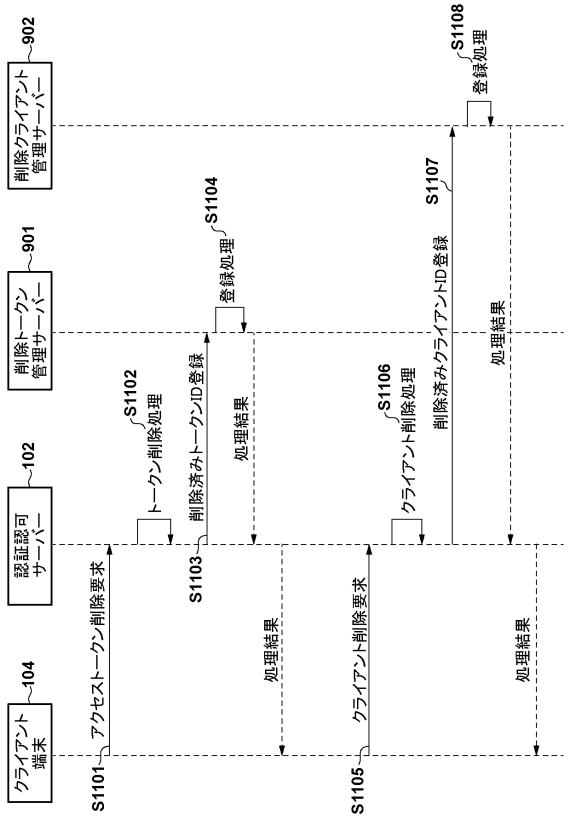
【図９】



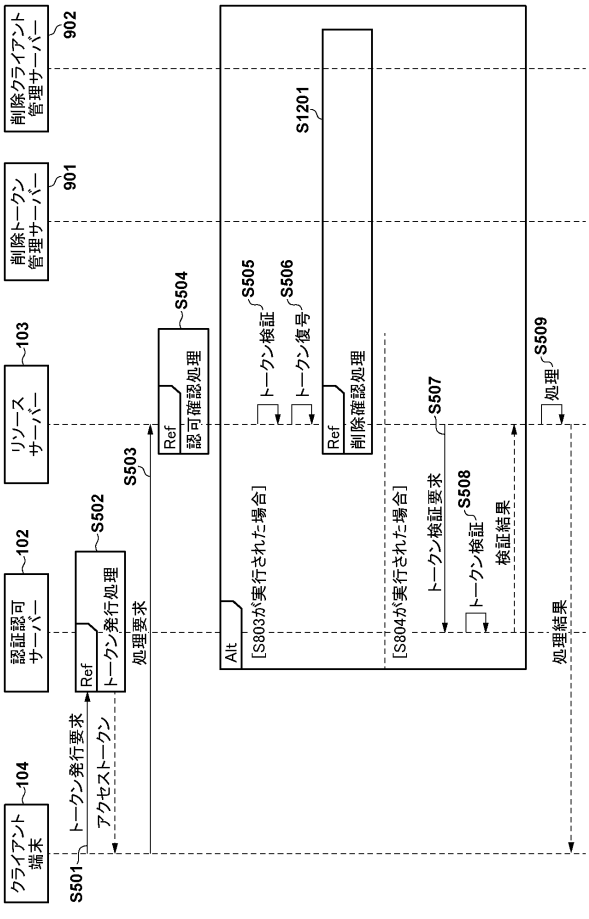
【図１０】



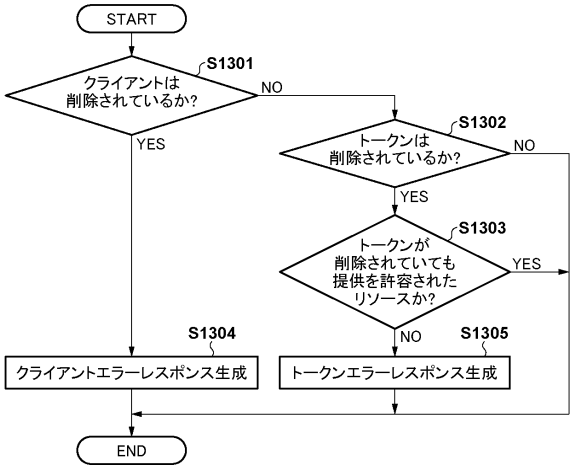
【図１１】



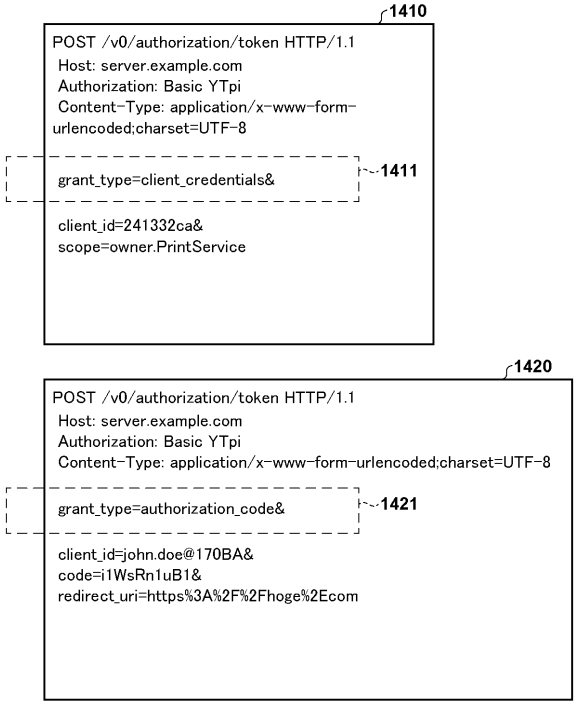
【図１２】



【図１３】


START
S1301
クライアントは
削除されているか?
NO
YES
S1302
トークンは
削除されているか?
NO
YES
S1303
トークンが
削除されていても
提供を許容された
リソースか?
YES
NO
S1304
S1305
クライアントエラーレスポンス生成
トークンエラーレスポンス生成
END

【図１４】


1410
POST /v0/authorization/token HTTP/1.1
Host: server.example.com
Authorization: Basic YTpi
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
grant_type=client_credentials&
1411
client_id=241332ca&
scope=owner.PrintService
1420
POST /v0/authorization/token HTTP/1.1
Host: server.example.com
Authorization: Basic YTpi
Content-Type: application/x-www-form-urlencoded;charset=UTF-8
grant_type=authorization_code&
1421
client_id=john.doe@170BA&
code=i1WsRn1uB1&
redirect_uri=https%3A%2F%2Fhoge%2Ecom

フロントページの続き

(56)参考文献 特開２０１３－１４０４８０（ＪＰ，Ａ）
特開２００７－１４９０１０（ＪＰ，Ａ）
特開２００４－１７１５２４（ＪＰ，Ａ）
特開２０１４－１３７６４８（ＪＰ，Ａ）
特開２０１４－１５３８０６（ＪＰ，Ａ）
米国特許出願公開第２０１５／０１５０１０９（ＵＳ，Ａ１）
特開２０１７－４３０１（ＪＰ，Ａ）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／３３
Ｇ０６Ｆ　２１／６２
Ｇ０６Ｆ　２１／６４
Ｇ０９Ｃ　１／００
Ｈ０４Ｌ　９／３２