(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号

特許第6006533号
(P6006533)

(45)発行日　平成28年10月12日(2016.10.12)　(24)登録日　平成28年9月16日(2016.9.16)

(51)Int.Cl.		F I		
G06F 21/41	(2013.01)	G06F	21/41	
G06F 21/33	(2013.01)	G06F	21/33	
H04L 9/32	(2006.01)	H04L	9/00	675D

請求項の数 10　(全 29 頁)

(21)出願番号　特願2012-120140 (P2012-120140)
(22)出願日　平成24年5月25日 (2012.5.25)
(65)公開番号　特開2013-246655 (P2013-246655A)
(43)公開日　平成25年12月9日 (2013.12.9)
審査請求日　平成27年5月25日 (2015.5.25)

(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】認可サーバー及びクライアント装置、サーバー連携システム、トークン管理方法

(57)【特許請求の範囲】

【請求項１】

クライアント装置からリソースサーバーへのアクセス要求を、該要求と関連して前記クライアント装置から受信した有効な認可情報に基づいて認可する認可サーバーであって、

認証情報とともにクライアント装置から受信した発行要求に応じて、前記リソースサーバーへアクセスするために利用される認可情報と、新たな認可情報を前記認証情報なしに再発行するために利用される更新認可情報とを発行する発行手段と、

更新認可情報とともに受信したリフレッシュ処理要求に応じて、新たな更新認可情報及び新たな認可情報を再発行し、再発行された認可情報及び更新認可情報に関連付けて、新たな更新認可情報及び認可情報を再発行するために前記発行手段により発行した更新認可情報を初回更新認可情報として記憶する再発行手段と、

更新認可情報とともに受信した無効化要求に応じて、受信した前記更新認可情報が初回更新認可情報として関連付けられている更新認可情報を無効化する無効化手段と

を有することを特徴とする認可サーバー。

【請求項２】

前記無効化手段は、さらに、受信した前記更新認可情報も更に無効化することを特徴とする請求項１に記載の認可サーバー。

【請求項３】

前記無効化手段は、受信した前記更新認可情報が初回更新認可情報として関連付けられている更新認可情報に加えて、受信した前記更新認可情報が初回更新認可情報として関連

付けられた認可情報も無効化することを特徴とする請求項１または２に記載の認可サーバー。
【請求項４】
前記再発行手段は、新たな認可情報を再発行した場合、リフレッシュ処理要求とともに受信した更新認可情報を無効化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認可サーバー。
【請求項５】
リソースサーバーに対して、認可サーバーにより発行された認可情報とともにアクセス要求を送信して前記リソースサーバーによるサービスを要求するクライアント装置であって、
前記認可サーバーにより発行された認可情報と、新たな認可情報を認証情報なしに再発行するために利用される更新認可情報と、前記認可情報を再発行するために前記認可サーバーが最初に発行した初回更新認可情報とを関連付けて記憶する手段と、
記憶した前記初回更新認可情報とともに無効化要求を前記認可サーバーに送信して、前記初回更新認可情報に関連付けられた更新認可情報の無効化を要求する無効化要求手段とを有することを特徴とするクライアント装置。
【請求項６】
アクセス要求に対して、前記認可情報が無効であるとの応答を受信した場合、無効であると応答された前記認可情報に関連付けられた更新認可情報とともに、リフレッシュ処理要求を前記認可サーバーに対して送信する手段を更に有し、
前記リフレッシュ処理要求に対して、前記更新認可情報が無効であると応答された場合に、前記無効化要求手段による更新認可情報の無効化を要求することを特徴とする請求項５に記載のクライアント装置。
【請求項７】
クライアント装置からリソースサーバーへのアクセス要求を、該要求と関連して前記クライアント装置から受信した有効な認可情報に基づいて認可する認可サーバーと、前記リソースサーバーに対して、前記認可サーバーにより発行された認可情報とともにアクセス要求を送信して前記リソースサーバーによるサービスを要求するクライアント装置と、前記クライアント装置に対してサービスを提供する前記リソースサーバーとを含むサーバー連携システムであって、
前記認可サーバーは、
認証情報とともにクライアント装置から受信した発行要求に応じて、前記リソースサーバーへアクセスするために利用される認可情報と、新たな認可情報を前記認証情報なしに再発行するために利用される更新認可情報とを発行する発行手段と、
更新認可情報とともに受信したリフレッシュ処理要求に応じて、新たな更新認可情報及び新たな認可情報を再発行し、再発行された認可情報及び更新認可情報に関連付けて、新たな更新認可情報及び認可情報を再発行するために前記発行手段により発行した更新認可情報を初回更新認可情報として記憶する再発行手段と、
更新認可情報とともに受信した無効化要求に応じて、受信した前記更新認可情報が初回更新認可情報として関連付けられている更新認可情報、及び、受信した前記更新認可情報を無効化する無効化手段とを有し、
前記クライアント装置は、
前記認可サーバーにより発行された認可情報と、新たな認可情報を認証情報なしに再発行するために利用される更新認可情報と、前記認可情報を再発行するために前記認可サーバーが最初に発行した初回更新認可情報とを関連付けて記憶する手段と、
記憶した前記初回更新認可情報とともに無効化要求を前記認可サーバーに送信して、前記初回更新認可情報に関連付けられた更新認可情報の無効化を要求する無効化要求手段とを有することを特徴とするサーバー連携システム。
【請求項８】
クライアント装置からリソースサーバーへのアクセス要求を、該要求と関連して前記ク

ライアント装置から受信した有効な認可情報に基づいて認可する認可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であって、
認証情報とともにクライアント装置から受信した発行要求に応じて、前記リソースサーバーへアクセスするために利用される認可情報と、新たな認可情報を前記認証情報なしに再発行するために利用される更新認可情報とを発行する発行手段と、
更新認可情報とともに受信したリフレッシュ処理要求に応じて、新たな更新認可情報及び新たな認可情報を再発行し、再発行された認可情報及び更新認可情報に関連付けて、新たな更新認可情報及び認可情報を再発行するために前記発行手段により発行した更新認可情報を初回更新認可情報として記憶する再発行手段と、
更新認可情報とともに受信した無効化要求に応じて、受信した前記更新認可情報が初回更新認可情報として関連付けられている更新認可情報を無効化する無効化手段と
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９】
リソースサーバーに対して、認可サーバーにより発行された認可情報とともにアクセス要求を送信して前記リソースサーバーによるサービスを要求するクライアント装置としてコンピューターを機能させるためのプログラムであって、
前記認可サーバーにより発行された認可情報と、新たな認可情報を認証情報なしに再発行するために利用される更新認可情報と、前記認可情報を再発行するために前記認可サーバーが最初に発行した初回更新認可情報とを関連付けて記憶する手段と、
記憶した前記初回更新認可情報とともに無効化要求を前記認可サーバーに送信して、前記初回更新認可情報に関連付けられた更新認可情報の無効化を要求する無効化要求手段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１０】
クライアント装置からリソースサーバーへのアクセス要求を、該要求と関連して前記クライアント装置から受信した有効な認可情報に基づいて認可する認可サーバーと、前記リソースサーバーに対して、前記認可サーバーにより発行された認可情報とともにアクセス要求を送信して前記リソースサーバーによるサービスを要求するクライアント装置と、前記クライアント装置に対してサービスを提供する前記リソースサーバーとを有するサーバー連携システムにおけるトークン管理方法であって、
前記認可サーバーが、認証情報とともにクライアント装置から受信した発行要求に応じて、前記リソースサーバーへアクセスするために利用される認可情報と、新たな認可情報を前記認証情報なしに再発行するために利用される更新認可情報とを発行する発行工程と、
前記クライアント装置が、前記認可サーバーにより発行された認可情報と、新たな認可情報を認証情報なしに再発行するために利用される更新認可情報と、前記認可情報を再発行するために前記認可サーバーが最初に発行した初回更新認可情報とを関連付けて記憶する工程と、
前記クライアント装置が、アクセス要求に対して、前記認可情報が無効であるとの応答を受信した場合、無効であると応答された前記認可情報に関連付けられた更新認可情報とともにリフレッシュ処理要求を前記認可サーバーに対して送信する工程と、
前記認可サーバーが、更新認可情報とともに受信したリフレッシュ処理要求に応じて、新たな更新認可情報及び新たな認可情報を再発行し、再発行された認可情報及び更新認可情報に関連付けて、新たな更新認可情報及び認可情報を再発行するために前記発行工程により発行した更新認可情報を初回更新認可情報として記憶する再発行工程と、
前記クライアント装置が、記憶した前記初回更新認可情報とともに無効化要求を前記認可サーバーに送信して、前記初回更新認可情報に関連付けられた更新認可情報の無効化を要求する無効化要求工程と、
前記認可サーバーが、更新認可情報とともに受信した無効化要求に応じて、受信した前記更新認可情報が初回更新認可情報として関連付けられている更新認可情報を無効化する無効化工程と

を有することを特徴とするトークン管理方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、たとえば複数のオンラインサービスシステムをマッシュアップして実現されるオンラインシステムにおいて、複数のオンラインサービスシステム間のアクセスを制御するための認可サーバー及びクライアント装置、サーバー連携システム、トークン管理方法に関する。
【背景技術】
【０００２】
近年、クラウドサービスと呼ばれる、インターネットを介してソフトウェアの機能を提供するシステムが注目を集めている。最近では、複数のクラウドサービスを連携させて、新しいシステムを提供するケースが増えている。
【０００３】
クラウドサービスを連携させるに際して、サービスを提供するシステム間のアクセス制御を安全且つ容易に行う仕組みとしてＯＡｕｔｈと呼ばれる技術がある（非特許文献１）。
【０００４】
ＯＡｕｔｈは、連携先のシステムのアクセス権限に限り、連携元のシステムに対してユーザーのアクセス権限を移譲する技術である。この技術により、連携元のシステムは、連携先のシステムに対して当該ユーザーの権限でアクセスでき、連携先システムが提供するサービスを利用したサービスをユーザーに提供できる。ＯＡｕｔｈによるシステムのアクセス権限移譲の構成を、連携するシステムの夫々の認証機構が備えることで、連携先のシステムにおけるユーザーＩＤやパスワードなどの、セキュリティに関わる認証情報を連携元のシステムに記憶させることなく、安全にシステム間の連携を実現することができる。ＯＡｕｔｈでは、連携元のシステムからのアクセスを許可する認可情報に有効期間を設けており、認可情報の有効期間が経過した後に再発行する仕組みや、認可情報および認可情報の再発行を無効にする仕組みも設けている。
【０００５】
一方、ＯＡｕｔｈとは異なる別の認証技術ではあるが、利用許可に有効期間を設定する方法がある。そこにおいて、各システムで利用許可の有効期間を延長する方法や、有効期間が切れた後でも一定期間ならば各システムでの検証を省略して再度利用を許可する方法が従来から知られている（特許文献１）。
【先行技術文献】
【特許文献】
【０００６】
【特許文献１】特表２０１１－５１９０８７号公報
【非特許文献】
【０００７】
【非特許文献１】"Ｔｈｅ　ＯＡｕｔｈ　２．０　Ａｕｔｈｏｒｉｚａｔｉｏｎ　Ｐｒｏｔｏｃｏｌ　ｄｒａｆｔ－ｉｅｔｆ－ｏａｕｔｈ－ｖ２－２５"、Ｅ．Ｈａｍｍｅｒ、２０１２年１０月９日、＜ＵＲＬ　ｈｔｔｐ://ｔｏｏｌｓ.ｉｅｔｆ.ｏｒｇ／ｈｔｍｌ／ｄｒａｆｔ－ｉｅｔｆ－ｏａｕｔｈ－ｖ２－２５＞
【発明の概要】
【発明が解決しようとする課題】
【０００８】
しかしながら、現状のＯＡｕｔｈにおける、認可情報の有効期間が経過した後に再発行する構成にはセキュリティの面で課題がある。具体的には、クライアントの認証情報および更新認可情報が漏えいした場合、悪意のあるユーザーによる認可情報の不正な再発行が行われてしまう問題である。また、クライアントの認証情報および更新認可情報の漏えい

に対応するためサーバーシステム内で管理しているクライアントの認証情報を変更する場合、サーバーシステムの運用に大きな影響を与えることになる。

【０００９】

本願発明は、上述した課題を解決する認可システムを提供することを目的の１つとする。具体的には、より安全性を高めた認可サーバー及びクライアント装置、サーバー連携システム、トークン管理方法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、本発明は以下の構成を有する。

【００１１】

クライアント装置からリソースサーバーへのアクセス要求を、該要求と関連して前記クライアント装置から受信した有効な認可情報に基づいて認可する認可サーバーであって、

認証情報とともにクライアント装置から受信した発行要求に応じて、リソースサーバーへアクセスするために利用される認可情報と、新たな認可情報を前記認証情報なしに再発行するために利用される更新認可情報とを発行する発行手段と、

更新認可情報とともに受信したリフレッシュ処理要求に応じて、新たな更新認可情報及び新たな認可情報を再発行し、再発行された認可情報及び更新認可情報に関連付けて、新たな更新認可情報及び認可情報を再発行するために前記発行手段により発行した更新認可情報を初回更新認可情報として記憶する再発行手段と、

更新認可情報とともに受信した無効化要求に応じて、受信した前記更新認可情報が初回更新認可情報として関連付けられている更新認可情報を無効化する無効化手段とを有する。

【００１２】

他の観点によれば本発明は以下の構成を有する。

【００１３】

リソースサーバーに対して、認可サーバーにより発行された認可情報とともにアクセス要求を送信して前記リソースサーバーによるサービスを要求するクライアント装置であって、

前記認可サーバーにより発行された認可情報と、新たな認可情報を認証情報なしに再発行するために利用される更新認可情報と、前記認可情報を再発行するために前記認可サーバーが最初に発行した初回更新認可情報とを関連付けて記憶する手段と、

記憶した前記初回更新認可情報とともに無効化要求を前記認可サーバーに送信して、前記初回更新認可情報に関連付けられた更新認可情報の無効化を要求する無効化要求手段とを有する。

【００１４】

さらに他の観点によれば本発明は以下の構成を有する。

【００１５】

クライアント装置からリソースサーバーへのアクセス要求を、該要求と関連して前記クライアント装置から受信した有効な認可情報に基づいて認可する認可サーバーと、リソースサーバーに対して、認可サーバーにより発行された認可情報とともにアクセス要求を送信して前記リソースサーバーによるサービスを要求するクライアント装置と、前記クライアント装置に対してサービスを提供するリソースサーバーとを含むサーバー連携システムであって、

前記認可サーバーは、

認証情報とともにクライアント装置から受信した発行要求に応じて、リソースサーバーへアクセスするために利用される認可情報と、新たな認可情報を前記認証情報なしに再発行するために利用される更新認可情報とを発行する発行手段と、

更新認可情報とともに受信したリフレッシュ処理要求に応じて、新たな更新認可情報及び新たな認可情報を再発行し、再発行された認可情報及び更新認可情報に関連付けて、新たな更新認可情報及び認可情報を再発行するために前記発行手段により発行した更新認可

情報を初回更新認可情報として記憶する再発行手段と、
更新認可情報とともに受信した無効化要求に応じて、受信した前記更新認可情報が初回更新認可情報として関連付けられている更新認可情報、及び、受信した前記更新認可情報を無効化する無効化手段とを有し、
前記クライアント装置は、
前記認可サーバーにより発行された認可情報と、新たな認可情報を認証情報なしに再発行するために利用される更新認可情報と、前記認可情報を再発行するために前記認可サーバーが最初に発行した初回更新認可情報とを関連付けて記憶する手段と、
記憶した前記初回更新認可情報とともに無効化要求を前記認可サーバーに送信して、前記初回更新認可情報に関連付けられた更新認可情報の無効化を要求する無効化要求手段とを有する。

【００１６】
他の観点によれば本発明は以下の構成を有する。

【００１７】
クライアント装置からリソースサーバーへのアクセス要求を、該要求と関連して前記クライアント装置から受信した有効な認可情報に基づいて認可する認可サーバーと、前記リソースサーバーに対して、前記認可サーバーにより発行された認可情報とともにアクセス要求を送信して前記リソースサーバーによるサービスを要求するクライアント装置と、前記クライアント装置に対してサービスを提供するリソースサーバーとを有するサーバー連携システムにおけるトークン管理方法であって、
前記認可サーバーが、認証情報とともにクライアント装置から受信した発行要求に応じて、リソースサーバーへアクセスするために利用される認可情報と、新たな認可情報を前記認証情報なしに再発行するために利用される更新認可情報とを発行する発行工程と、
前記クライアント装置が、前記認可サーバーにより発行された認可情報と、新たな認可情報を認証情報なしに再発行するために利用される更新認可情報と、前記認可情報を再発行するために前記認可サーバーが最初に発行した初回更新認可情報とを関連付けて記憶する工程と、
前記クライアント装置が、アクセス要求に対して、前記認可情報が無効であるとの応答を受信した場合、無効であると応答された前記認可情報に関連付けられた更新認可情報とともにリフレッシュ処理要求を前記認可サーバーに対して送信する工程と、
前記認可サーバーが、更新認可情報とともに受信したリフレッシュ処理要求に応じて、新たな更新認可情報及び新たな認可情報を再発行し、再発行された認可情報及び更新認可情報に関連付けて、新たな更新認可情報及び認可情報を再発行するために前記発行手段により発行した更新認可情報を初回更新認可情報として記憶する再発行工程と、
前記クライアント装置が、記憶した前記初回更新認可情報とともに無効化要求を前記認可サーバーに送信して、前記初回更新認可情報に関連付けられた更新認可情報の無効化を要求する無効化要求工程と、
前記認可サーバーが、更新認可情報とともに受信した無効化要求に応じて、受信した前記更新認可情報が初回更新認可情報として関連付けられている更新認可情報を無効化する無効化工程とを有する。

【発明の効果】

【００１８】
本発明によれば、クライアントの認証情報および更新認可情報が漏えいした場合にも、認可情報の不正な再発行を防げる。

【００１９】
また、本発明によれば、クライアントの認証情報の変更をせずに済むためクライアントのシステムを停止することはなく、小さい影響範囲で更新認可情報の漏えいに対応できる。

【図面の簡単な説明】

【００２０】

【図１】システム全体図
【図２】サーバーのハードウェア構成図
【図３】外部サーバーのソフトウェア構成図
【図４】アクセス管理サーバーのソフトウェア構成図
【図５】帳票サーバーのソフトウェア構成図
【図６】外部サービスシステム１０３が保持するデータ構造を示したテーブルの図
【図７】アクセス管理サービスシステム１０４が保持するアカウントテーブルの図
【図８】アクセス管理サービスシステム１０４が保持する認可コードのデータ構造を示したテーブルの図
【図９】アクセス管理サービスシステム１０４が保持する認可情報のデータ構造を示したテーブルの図
【図１０】アクセストークン発行のフロー図
【図１１】認可画面の一例を示した図
【図１２】本発明によって制御される認可処理のフロー図
【図１３】第１の実施形態における認可情報無効化処理のフロー図
【図１４】第２の実施形態における認可情報無効化処理のフロー図
【図１５】ＯＡｕｔｈの権限移譲フロー図
【図１６】ＯＡｕｔｈの課題１を示す図
【図１７】ＯＡｕｔｈの課題２を示す図
【発明を実施するための形態】
【００２１】
まず本発明の課題の具体例を幾つか紹介する。ＯＡｕｔｈによりアクセス権限移譲を行う際には、連携先システムにおいて、連携元システムの権限確認と、連携元システムを利用しているユーザーの権限確認を行う。連携元システムとはユーザーが直にアクセスしてサービスを要求するシステムである。連携先システムとは、連携元システムと連携し、連携元システムに対してサービス等（リソースを含む）を提供するシステムである。以降、ＯＡｕｔｈの定義に従い、連携元システムをクライアント、連携先システムをリソースサーバー、リソースサーバーの認証情報および認可情報を管理するサーバーを認可サーバーと呼ぶ。ここで、認可サーバーが管理する認証情報とは、リソースサーバーを利用するユーザーやシステムを認証するためのセキュリティ情報である。例えばユーザーＩＤおよびパスワードである。認可サーバーが管理する認可情報とは、ＯＡｕｔｈの権限移譲フローによって認可サーバーが発行する、リソースサーバーへのアクセスを許可する情報である。認可情報を、ＯＡｕｔｈではアクセストークンと呼ぶ。クライアントはリソースサーバーにアクセスする際に、リソースサーバーにアクセストークンを送る。リソースサーバーは受け取ったアクセストークンを認可サーバーで確認させ、アクセス可否を判定する。これにより、クライアントは認可サーバーが管理している認証情報を知らずに、リソースサーバーが利用できる。
【００２２】
図１５を用いて、ＯＡｕｔｈの権限移譲フローについて説明する。なお、連携元システムを利用しているユーザーをリソースオーナー（単にオーナーとも呼ぶ）、ユーザーが操作している情報処理端末が備えるＷｅｂブラウザをユーザーエージェントと呼ぶ。
【００２３】
１５０１において、オーナーがユーザーエージェントを介してクライアントを操作している。この状態で、クライアントがリソースサーバーを利用するために、ＯＡｕｔｈのフローを開始する。１５０２において、クライアントがユーザーエージェントを認可サーバーにリダイレクトさせる。この際クライアントは、クライアント自身を一意に識別するクライアントＩＤとリダイレクトＵＲＬとを認可サーバーに送る。
【００２４】
１５０３において、認可サーバーは、ユーザーエージェントを介してオーナーの認証を行う。オーナーの認証は例えば、ユーザーエージェントに認証画面を表示し、オーナーに

対して認可サーバーが管理しているユーザーIDとパスワードの入力を要求する方法で行われる。オーナーの認証に成功すると、認可サーバーは認証したオーナーがリソースサーバーに対して適切なアクセス権限を持つか判定する。

【００２５】

１５０４において、認可サーバーはユーザーエージェントを介して、アクセス権限を持つと判定されたオーナーに対して、クライアントによるリソースサーバーへのアクセスの認可確認を行う。オーナーへの認可確認は例えば、認可確認画面を表示し、オーナーに認可ボタンの押下を要求する方法などである。オーナーが認可を行うと、認可サーバーは認可コードを生成する。認可コードとは、オーナーがクライアントに対して、リソースサーバーへのアクセスを許可したことを示す情報である。

【００２６】

１５０５において、認可サーバーは、１５０２でクライアントから渡されたリダイレクトＵＲＬに対して、生成した認可コードを送る。

【００２７】

１５０６において、クライアントは認可サーバーに対して、リソースサーバー利用のための認可情報を要求する。この際、クライアントは受け取った認可コードと、クライアントの認証情報を送る。クライアントの認証情報は例えば、クライアントのＩＤとパスワードである。認可サーバーは、受け取った認可コードの確認と、クライアントの認証を行う。クライアントの認証に成功すると、認可サーバーはリソースサーバーがクライアントに対して連携を許可しているかを確認する。認可コードが有効であり、かつリソースサーバーがクライアントに対して連携を許可していることが確認されると、認可サーバーはリソースサーバーに対する認可情報を生成する。

【００２８】

１５０７において、認可サーバーは生成した認可情報をクライアントに送る。

【００２９】

１５０８において、クライアントは認可情報をリソースサーバーに送り、リソースサーバーの利用要求を行う。

【００３０】

１５０９において、リソースサーバーはアクセス許可判定のために、認可サーバーに対して受け取った認可情報を送る。認可サーバーは受け取った認可情報の確認を行う。

【００３１】

１５１０において、認可サーバーはリソースサーバーに認可情報の確認結果を返す。リソースサーバーは認可情報の確認結果に従い、クライアントに対してアクセスの可否を判断する。

【００３２】

ここで、ＯＡｕｔｈでは１５０９、１５１０のように、認可サーバーでのアクセストークンの確認のみでリソースサーバーに対するアクセス可否を判断している。そのため、アクセストークンが正規のクライアント以外のシステムに流出すると、意図しないクライアントがアクセストークンを発行した正規のクライアントになりすましてリソースサーバーを利用できる。このような問題から、ＯＡｕｔｈは安全のためにアクセストークンの有効期間を短く設定する事を推奨している。一方、アクセストークンの有効期間が短いと、オーナーが一度認可したシステムでも、１５０４のオーナーに対する認可確認が毎回発生するため、利便性が低くなる。そのため、ＯＡｕｔｈは利便性を高くするために、１５０６でのアクセストークン発行時に、アクセストークンと合わせてアクセストークン更新を認可した更新認可情報を発行する方法を提供している。この更新認可情報をリフレッシュトークンと呼ぶ。リフレッシュトークンとは、オーナーに対する認可確認をせずに、アクセストークンを発行するための情報である。認可サーバーはリフレッシュトークンと共に発行されたアクセストークンと同じ権限か、権限を狭めた新たなアクセストークンを発行する。

【００３３】

リフレッシュトークンを用いて新たなアクセストークンを発行する際には、クライアントは認可サーバーに対してリフレッシュトークンと、クライアントの認証情報を送る。クライアントの認証情報は例えば、クライアントのＩＤとパスワードである。認可サーバーはリフレッシュトークンの確認とクライアントの認証を行う。クライアントの認証に成功すると、認可サーバーはリソースサーバーがクライアントに対して連携を許可しているかを確認する。リフレッシュトークンが有効であり、かつリソースサーバーがクライアントに対して連携を許可していることが確認されると、認可サーバーは新たなアクセストークンとリフレッシュトークンを発行する。この際、認可サーバーは利用したリフレッシュトークンを無効にする。以降、リフレッシュトークンを用いてアクセストークンの再発行をする処理をリフレッシュ処理と呼ぶ。なお、トークンの再発行を、トークンの更新あるいはトークンのリフレッシュと呼ぶこともあるが、これらは本実施形態では同義である。

【００３４】

ここで、リフレッシュトークンおよびクライアントの認証情報が流出した場合、不正なクライアントが正規のクライアントになりすましてリフレッシュ処理を行える。リフレッシュ処理を行った不正なクライアントは、発行されたアクセストークンを用いて正規のクライアントになりすましてリソースサーバーを利用出来る。また不正なクライアントは、リフレッシュ処理によって発行された新たなリフレッシュトークンを利用することで、正規クライアントになりすましてリソースサーバーを利用し続けられる。さらに、不正なクライアントが発行したアクセストークンを提供するだけで、様々なクライアントが容易に、リソースサーバーを不正利用できてしまう。

【００３５】

ＯＡｕｔｈが公開している範囲でリフレッシュトークンおよびクライアントの認証情報の流出に対応するには、２つの方法がある。１つ目の方法は、リフレッシュトークンを指定し、指定したリフレッシュトークンと、そのリフレッシュトークンとペアになるアクセストークンをそれぞれ無効にする方法である。この方法によりリフレッシュトークンが無効になるので、リフレッシュ処理を防げる。

【００３６】

しかし、この方法には課題がある。図１６を用いて本方法の課題を説明する。１６０１において、不正クライアントが認可サーバーに対してリフレッシュ処理を要求する。ここで不正クライアントとは、クライアントから流出した認証情報およびリフレッシュトークンを所持しているシステムとする。不正クライアントは正規クライアントの認証情報を利用しているため、認可サーバーでのクライアント認証に成功する。さらに、リフレッシュトークンは正しいものであるため、認可サーバーでのリフレッシュトークン確認も成功する。これにより、認可サーバーはリフレッシュ処理を行い、新たなアクセストークンとリフレッシュトークンを発行する。その後、認可サーバーはリフレッシュ処理に利用したリフレッシュトークンを無効にする。１６０２において、認可サーバーは不正クライアントに対し、発行したアクセストークンとリフレッシュトークンを渡す。

【００３７】

この状態で、１６０３において、クライアントがリフレッシュトークンとアクセストークンの無効化を要求する。しかし、クライアントが持つリフレッシュトークンは、１６０１でのリフレッシュ処理により既に無効となっている。またクライアントは、１６０１のリフレッシュ処理で発行されたリフレッシュトークンを知る事が出来ない。そのためクライアントは、不正クライアントが１回でもリフレッシュ処理を行うと、それ以降のリフレッシュ処理を無効化できないという課題を持つ。

【００３８】

２つ目の方法は、正規クライアントの認証情報を更新する方法である。例えば、認可サーバーでクライアントのパスワードを変更すると、変更前の認証情報しか知らない不正なクライアントによるリフレッシュ処理を防げる。この方法によれば、不正なクライアントがリフレッシュ処理を行っていても、リフレッシュ処理を防げる。

【００３９】

しかし、この方法は影響範囲が大きいという課題を持つ。図１７を用いて本方法の課題を説明する。１７０１でオーナーＡがクライアントを利用しており、１７０２でクライアントが、以前発行したアクセストークンを用いてリソースサーバーへのアクセスを要求したとする。１７０３において、リソースサーバーはアクセス許可判定のためにアクセストークンを認可サーバーに送る。ここで、アクセストークンが無効であったとする。そのため、１７０４において、認可サーバーはリソースサーバーにアクセストークン無効を返す。リソースサーバーはアクセストークンが無効であったため、アクセスを拒否する。１７０５において、リソースサーバーはクライアントにアクセストークン無効を返し、アクセスを拒否する。

【００４０】

１７０６において、クライアントは、アクセストークンが無効であったため、リフレッシュトークンを利用してアクセストークンの再発行を試みる。認可サーバーにリフレッシュトークンとクライアントの認証情報を送りリフレッシュ処理を要求する。ここで、リフレッシュトークンが無効であったとする。リフレッシュトークンが無効である場合は例えば、すでにリフレッシュ処理が行われていて無効になっていた場合や、リフレッシュトークンの有効期限が経過して無効になっている場合がある。認可サーバーは１７０７で、クライアントにリフレッシュトークン無効を返す。

【００４１】

クライアントはリフレッシュトークン無効を受け、不正にリフレッシュ処理がされた可能性を検知する。そのため、１７０８において、認可サーバーに対してクライアントの認証情報の変更を要求する。認証情報の変更は例えば、パスワード変更である。認可サーバーはクライアントの認証情報変更要求を受け、認可サーバーで管理しているクライアントの認証情報を更新する。その後、１７０９でクライアントに対して、クライアントの認証情報変更終了を通知する。

【００４２】

ここで、認可サーバーでのクライアント認証は、アクセストークン発行の際と、リフレッシュ処理の際に行われる。そのため、パスワードを変更する際には、これらの処理を止める必要がある。つまり、１７０８、１７０９の間はクライアントでのこれらの処理が停止される。ここで、ＯＡｕｔｈではひとつのクライアントを複数のオーナーが利用するモデルである。例えば、オーナーＡによる１７０１のクライアント利用と並行して、オーナーＢからのクライアント利用要求１７１０、オーナーＣからのクライアント利用要求１７１１がある。そのため、オーナーＡの認可によって発行されたリフレッシュトークンが無効であった場合にも関わらず、１７０８、１７０９の間は、オーナーＢやオーナーＣからの要求によるアクセストークン発行処理やリフレッシュ処理も停止してしまう。つまり、もしクライアントを停止すると、クライアントを利用している全てのオーナーに影響が出る。

【００４３】

本発明は、このような課題を解決するものであり、クライアントの認証情報およびリフレッシュトークンが流出した場合に、小さい影響範囲でアクセストークンの再発行を防ぎ、システムの不正利用を防ぐ仕組みを提供するものである。

【００４４】

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。

【００４５】

［実施形態１］

インターネット上で、様々なサービス提供者が様々なオンラインサービスを提供している。単一のサービス提供者が運営している単体のオンラインサービスもあれば、複数のサービス提供者が運営している複数のオンラインサービスを組み合わせて、１つのソリューションを実現するという手法も存在する。後者は、マッシュアップと呼ばれ、表向きはあたかも１つのＷｅｂサイトあるいはＷｅｂサービスとして見える。しかしながら、実際にはバックエンドでは、複数のオンラインサービスが連携・連動して、必要な機能を組み合

わせてソリューションを実現する。なお、ここで言うオンラインサービスとは、Ｗｅｂサイト、Ｗｅｂアプリケーション、Ｗｅｂサービスなどが提供する機能群のことである。Ｗｅｂサイト、Ｗｅｂアプリケーション、Ｗｅｂサービスなどは、サーバーコンピューターで実行されるソフトウェアである。マッシュアップにより構成されたシステムを本実施形態ではサーバー連携システムあるいは単に連携システムとも呼ぶ。

【００４６】

＜オンラインサービスシステムの構成例＞

図１は、各種オンラインサービスが存在するネットワーク構成を示している。インターネット１００は、インターネットなどの外部から接続可能なパブリックなネットワークである。イントラネット１０１はＬＡＮなどの外部から接続不可能なプライベートなネットワークである。情報機器端末１０２は、インターネット１００を介して、パーソナルコンピューターやモバイル端末などのオンラインサービスを利用する際に使用される情報機器端末である。本例では２台の端末１０２Ａと端末１０２Ｂとが示されているが、いずれを用いてもよいため特に区別をしない限り情報機器端末１０２と称する。ＯＡｕｔｈにおいては、情報機器端末１０２を操作するユーザーをオーナー、情報機器端末１０２が備えるＷｅｂブラウザをユーザーエージェントと呼ぶ。　外部サービスシステム１０３は、後述する帳票サービスシステム１０５をオンラインでマッシュアップしたオンラインサービスシステムである。ＯＡｕｔｈにおいてはクライアントと呼ばれる。なお本実施形態では、クライアントが装置であることを明確にするためにクライアント装置と呼ぶことがある。外部サービスシステム１０３は１台ないし複数台の外部サーバーで構成され、負荷分散装置１０８によりインターネット１００からのリクエストを分散して処理する事が出来るよう構成されている。なお、図上は外部サービスシステム１０３を構成する外部サーバーは１０３Ａと１０３Ｂの２台となっているが、実際には１台から複数台の外部サーバーによって構成される。

【００４７】

アクセス管理サービスシステム１０４は、ユーザーの認証情報及び認可情報を管理するサービスシステムである。ＯＡｕｔｈにおいては認可サーバーと呼ばれる。アクセス管理サービスシステム１０４は1台ないし複数台のアクセス管理サーバーで構成され、負荷分散装置１０８によりインターネット１００およびイントラネット１０１からのリクエストを分散して処理する事が出来るよう構成されている。なお、図上はアクセス管理サービスシステム１０４を構成するアクセス管理サーバーは１０４Ａと１０４Ｂの２台となっているが、実際には１台から複数台のアクセス管理サーバーによって構成される。

【００４８】

帳票サービスシステム１０５は、インターネット１００を介した情報機器端末１０２あるいは外部サービスシステム１０３からのリクエストに応じて帳票を生成するオンラインサービスシステムである。ＯＡｕｔｈにおいてはリソースサーバーと呼ばれる。帳票サービスシステム１０５は1台ないし複数台の帳票サーバーで構成され、負荷分散装置１０８によりインターネット１００からのリクエストを分散して処理する事が出来るよう構成されている。また、帳票サービスシステム１０５はイントラネット１０１を介してのリクエストを負荷分散装置１０８により分散して処理する事も出来る。なお、図上は帳票サービスシステム１０５を構成する帳票サーバーは１０５Ａと１０５Ｂの２台となっているが、実際には１台から複数台の帳票サーバーによって構成される。また本例ではリソースサーバーとして帳票サービスシステムを例示しているが、ウエブを介してサービスを提供するサーバーであれば適用可能である。

【００４９】

＜サーバーコンピューターのハードウェア構成＞

図２は、図１に示した各種サーバーを構成するＷｅｂサイト、Ｗｅｂアプリケーション、Ｗｅｂサービスなどのソフトウェアを実行するサーバーコンピューターの情報処理機能の論理構成を示している。

【００５０】

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、マウスなどによる、情報の入出力を行うハードウェアである。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ハードウェアである。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し、各種サービスを実現する。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記憶装置である。ＲＡＭ２０５は、一時メモリ領域である。二次記憶装置２０６は、ＨＤＤに代表されるような外部記憶装置である。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。

【００５１】

＜外部サーバーの機能構成＞

図３は、外部サーバー１０３Ａの内部構造を示したブロック図である。リクエスト処理部３０１は、外部サービスシステム１０３がインターネット１００を経由して受信した機能リクエストを処理する処理部である。機能制御部３０２はリクエスト処理部３０１からの要求を受け、必要な処理を行い、応答データを呼び出し元に返す。機能連携データ３０３は、外部サービスシステム１０３が連携しているシステムに対するリクエストを生成するためのデータを管理する、認可コード管理部３０４は、認可コードのデータを管理する。トークン管理部３０５は認可情報のデータを管理する。外部サーバー１０３Ｂは、外部サーバー１０３Ａと異なる機能を提供することもできるが、本例では外部サーバー１０３Ａと同じ構成を有し、同じ機能を提供して負荷を分散している。また、外部サーバー１０３Ａが提供する機能としては、たとえばネットワークプリントサービスがある。たとえば、外部サーバー１０３Ａはクライアントとして、ユーザーエージェントであるウエブブラウザから印刷対象の帳票データの所在や名称の指定とともにその帳票データの印刷のリクエストを受信する。それに応じて外部サーバー１０３Ａは帳票サービスシステム１０５にアクセスしてユーザーに要求された帳票データを獲得する。そして要求に応じて指定されたデータとマージしたうえで印刷データに変換し、これも指定されたネットワークにある印刷装置あるいは印刷サーバーに印刷データを送信して印刷させる。もちろんこれは一例であって、外部サービスシステム１０３が提供するサービスは印刷サービスとは限らない。

【００５２】

＜アクセス管理サーバーの機能構成＞

図４は、アクセス管理サーバー１０４Ａの内部構造を示したブロック図である。アクセス管理サーバー１０４Ｂも同様である。アクセス管理リクエスト処理部４０１は、アクセス管理サービスシステム１０４がインターネット１００及びイントラネット１０１を経由して受信した認証および認可リクエストを処理する処理部である。また、アクセス管理リクエスト処理部４０１は、アクセス制御部４０２から返される応答データを呼び出し元に返す。アクセス制御部４０２は、認証データ管理部４０３及び認可データ管理部４０４から取得するデータに基づき、認証および認可リクエストに対して応答データを生成し、アクセス管理リクエスト処理部４０１に応答データを返す。認証データ管理部４０３は、ユーザーアカウントのデータを管理する。具体的にはユーザー固有のＩＤとパスワードの組などである。なおＩＤとパスワードの組をクレデンシャルとも呼ぶ。認可データ管理部４０４は、認可情報のデータを管理する。認可データ管理部４０４は認可情報のみならず更新認可情報のデータも管理する。認可データ管理部４０４は、最新の認可情報および更新認可情報を記憶するのみならず、後述するように、あるユーザーに対して最初に発行された更新認可情報と当該ユーザーに対する最新の更新認可情報との関連付けも記憶している。

【００５３】

＜帳票サーバー１０５の機能構成＞

図５は、帳票サーバー１０５Ａの内部構造を示したブロック図である。帳票サーバー１

０５Ｂも同様である。帳票リクエスト処理部５０１は、インターネット１００を経由して帳票データ生成リクエスト及び帳票データ取得リクエストを受信する。帳票制御部５０２は、帳票リクエスト処理部５０１が受信したリクエストに応じて必要な処理を行い、応答データを呼び出し元に返す。また、帳票制御部５０２はイントラネット１０１経由でアクセス管理サービスシステム１０４に対して認証リクエストを送信し、認証結果を受信する。また、アクセス管理サービスシステム１０４に対して認可確認リクエストを送信し、認可確認結果を受信する。帳票データ処理部５０３は、帳票制御部５０２から帳票データ生成リクエストを受信して、帳票データを生成する。また、帳票データ処理部５０３は生成した帳票データを応答として帳票制御部５０２に返す。帳票データ管理部５０４は、帳票データ処理部５０３における帳票データ生成処理に使用される帳票フォームデータ及び帳票データを登録、管理する。また、帳票制御部５０２からの帳票データ取得リクエストを受信して、応答として帳票データを返す。

【００５４】

＜外部サービスシステムにより管理される情報＞

図６は、外部サービスシステム１０３が保持する認可情報、認可コードおよび外部サービスシステムの認証情報のデータ構造をテーブル形式で示した図である。認可情報および認可に関連する情報は認可情報管理テーブル６００、認可コードは認可コードテーブル６１０、外部サービスシステムの認証情報はクライアントクレデンシャルテーブル６２０で管理される。

【００５５】

認可情報管理テーブル６００は、連携対象のシステム名を示す連携先システム名６０１、認可情報を示すアクセストークンＩＤ６０２、更新認可情報を示すリフレッシュトークンＩＤ６０３、初回リフレッシュトークンＩＤ６０４を含む。アクセストークンＩＤ６０３には、アクセス管理サーバーが発行するアクセストークンを保存する。リフレッシュトークンＩＤ６０３には、アクセス管理サーバーが発行するリフレッシュトークンを保存する。初回リフレッシュトークンＩＤ６０４には、最初の認可処理時にアクセス管理サーバーが発行する最初のリフレッシュトークンを保存する。また、リフレッシュ処理を実施すると、アクセストークンＩＤ６０２およびリフレッシュトークンＩＤ６０３は、再発行された認可情報および更新認可情報によりそれぞれ更新される。しかしながら、初回リフレッシュトークンＩＤ６０４には、最初のリフレッシュ処理に利用した更新認可情報が引き継いで記憶される。なおいずれもフィールド名称がトークンＩＤとなっているが、格納しているのはトークンのＩＤではなくトークンそのものである。

【００５６】

認可コードテーブル６１０は、連携対象のシステム名を示す連携先システム名６１１と、アクセス管理サービスシステム１０４が生成する認可コードとを一意に識別する認可コードＩＤ６１２とを含む。

【００５７】

認証情報テーブル６２０は、連携対象のシステム名を示す連携先システム名６２１と、連携対象のシステムに対して外部サービスシステム１０３を認証するためのクライアントＩＤ６２２と、パスワード６２３とを含む。図６の各データ構造に格納されるデータの処理詳細については後述する。なお外部サーバーシステム１０３が管理する認可情報管理テーブル６００の１つのレコードを、本実施形態ではクライアント認可関連情報と呼ぶこともある。

【００５８】

＜アクセス管理サービスシステムにより管理される情報＞

図７、図８、図９にアクセス管理サービスシステム１０４が保持する、認可及び認証に係る各種情報を示す。図７は、アクセス管理サービスシステム１０４が保持するユーザー情報およびシステム情報のデータ構造をテーブル形式で示した図である。ユーザー情報はユーザーテーブル７００、システム情報はクライアントテーブル７１０で管理される。

【００５９】

ここで管理されているユーザー情報とは、アクセス管理サービス１０４が管理しているシステム（ＯＡｕｔｈにおけるリソースサーバー）のユーザーの情報である。本実施形態においては、管理しているシステムの例として帳票サービスシステム１０５のユーザーが登録されている。

【００６０】

ここで管理されているシステム情報とは、ＯＡｕｔｈにおいて、認可サーバー（本例のアクセス管理サービスシステムに相当）がアクセストークンの発行およびリフレッシュ処理の際に行う、クライアントの認証に利用する認証情報である。本実施形態においては、クライアントである外部サービスシステム１０３を識別するためのクライアントＩＤとパスワードとを含む。

【００６１】

ユーザーテーブル７００は、ユーザーＩＤ７０１とパスワード７０２から成るユーザー情報を含む。クライアントテーブル７１０は、クライアントのＩＤを示すクライアントＩＤ７１１と、パスワード７１２から成るシステム情報を含む。図７の各データ構造に格納されるデータの処理詳細については後述する。

【００６２】

図８は、アクセス管理サービスシステム１０４が保持する認可コードのデータ構造をテーブル形式で示した図である。認可コードは認可コードテーブル８００で管理される。

【００６３】

認可コードテーブル８００は、認可コードを一意に識別する値を示す認可コードＩＤ８０１と、認可を実施したユーザーを一意に識別するユーザーＩＤ８０２から成る。認可コードは、リソースサーバーのアクセス権限を有するオーナーによるリソースサーバーへのアクセスの許可、すなわち本例では帳票サービスシステム１０５に対するアクセス権限を有するユーザーによる帳票サービスシステム１０５へのアクセスの許可を示すコードである。すなわち、認可コードに基づいて、リソースサーバーへのアクセス権限がオーナーからクライアントへと委譲される。図８のデータ構造に格納されるデータの処理詳細については後述する。

【００６４】

（認可情報）

図９は、アクセス管理サービスシステム１０４が保持する認可情報のデータ構造をテーブル形式で示した図である。認可情報は認可情報管理テーブル９００で管理される。なお認可情報とはアクセストークンに対応し、更新認可情報とはリフレッシュトークンに対応している。また図９においてフィールド９０１～９０８の一組の情報を本実施形態では認可関連情報と呼ぶことがある。

【００６５】

認可情報管理テーブル９００は、ひとつのアクセス権限ごとに、認可情報であるアクセストークンＩＤ９０１と、アクセストークン発行日時９０２と、アクセストークン有効日時（すなわち有効期限）９０３とを持つ。また、更新認可情報であるリフレッシュトークンＩＤ９０４と、リフレッシュトークン発行日時９０５と、リフレッシュトークン有効日時９０６とを持つ。さらに認可情報管理テーブル９００は、アクセス権限ごとに、当該アクセストークンによるリソースサーバー（すなわち帳票サービスシステム）へのアクセスを許可したユーザーのユーザーＩＤ９０７と、初回リフレッシュトークンＩＤ９０８を持つ。初回リフレッシュトークンＩＤ９０８には、当該アクセス権限に関する最初のアクセストークンと合わせて発行されたリフレッシュトークンＩＤを保存する。

【００６６】

なお、アクセス管理サービスシステム１０４がアクセストークンのリフレッシュ処理を行うと、アクセストークンおよびリフレッシュトークンともに再発行される。再発行されたアクセストークンも、元となるアクセストークンと同じひとつのアクセス権限に係るものである。しかし再発行されたアクセストークン等の認可関連情報とその元となるアクセストークン等の認可関連情報と区別する場合には、それぞれに「元」や「新」を付するこ

とにする。すなわちリフレッシュトークンを用いてリフレッシュ処理が行われると、新アクセストークン及び新リフレッシュトークンが再発行される。再発行前のものはそれぞれ元アクセストークン及び元リフレッシュトークンと呼ぶ。またひとつのアクセス権限に係る最初のアクセストークンおよびリフレッシュトークンをそれぞれ初回アクセストークンおよび初回リフレッシュトークンと呼ぶ。リフレッシュ処理により再発行されたアクセストークン及びリフレッシュトークンは、それらの発行日時や有効日時とともに許可情報管理テーブル９００に保存される。有効日時は、発行日時に所定の期間を加えて得ることができる。ユーザーＩＤ９０７には、リフレッシュ処理のために利用されたリフレッシュトークンに関連付けて格納されているユーザーＩＤを格納し、初回リフレッシュトークンＩＤ９０８には、利用したリフレッシュトークンに関連付けて格納されている初回リフレッシュトークンの値を格納する。ただし、利用したリフレッシュトークンに関連付けて保存されている初回リフレッシュトークンの値が"ｎｕｌｌ"であった場合には、リフレッシュ処理に利用したリフレッシュトークンの値が初回リフレッシュトークンであるので、その値を当該アクセス権限の初回リフレッシュトークンＩＤ９０８として保存する。もちろんｎｕｌｌの代わりに初回リフレッシュトークンを保存しておいてもよい。　また、リフレッシュ前のアクセストークンの有効日時およびリフレッシュトークンの有効日時がいずれも満了している場合には、当該アクセス権限に係る一組の情報はもはや不要なので、認可情報管理テーブル９００から削除できる。リフレッシュ前の元アクセストークンの有効日時が満了していない場合には、その元アクセストークンが使用される可能性があることから、その元アクセストークンに対応した認可関連情報は削除せずにそのまま残す。また認可情報管理テーブル９００にアクセスする際に、トークンの有効日時が満了しているか判定し、満了していればその都度削除してもよい。アクセス権限の認証の際、またはトークンのリフレッシュの際には、有効日時を参照してアクセス権限またはリフレッシュの権限が判定されるので、削除は必須ではない。しかし削除により記憶領域の効率化を図ることができる。なおリフレッシュトークンは、いったん利用されてリフレッシュ処理が行われると無効化される。

【００６７】

ここで図６と図９の認可情報管理テーブルの関係について簡単に説明する。外部サービスシステム１０３すなわちクライアントが管理する認可情報管理テーブル６００には、ひとつのアクセス権限について、当該クライアントが管理する最新のアクセストークンおよびリフレッシュトークン及びその初回リフレッシュトークンがクライアント認可関連情報として格納されている。またアクセス管理サービスシステム１０４すなわち認可サーバーが管理する認可情報管理テーブル９００には、ひとつのアクセス権限について、有効なアクセストークンまたは有効なリフレッシュトークンを含む認可関連情報がすべて格納されている。このため、適正なリフレッシュ処理によりトークンが更新されている限りは、最新のアクセストークン及びリフレッシュトークンについては、後述する手順により、外部サービスシステム１０３（クライアント）とアクセス管理サービスシステム１０４（認可サーバー）との間で同期が保たれている。しかしリフレッシュトークン及びクライアントクレデンシャルの漏えいにより不正なリフレッシュ処理（再発行処理または更新処理とも呼ぶ）が行われるとその同期が失われる。

【００６８】

図６及び図９の認可情報管理テーブルはその状態を例示している。図９の認可情報管理テーブル９００には、"EFGH5678"を初回アクセストークンとするアクセス権限について３組の認可情報が登録されている。このうち最新のアクセストークンは"MNOP3456"であり、最新のリフレッシュトークンは"8901JKL"、初回リフレッシュトークンは"0123ABCD"である。これに対して図６の認可情報管理テーブル６００では、初回リフレッシュトークン"0123ABCD"に対応したアクセストークンは"IJKL9012"であり、リフレッシュトークンは"4567EFGH"である。これらのアクセストークン及びリフレッシュトークンは認可情報管理テーブル９００にも格納されているが、すでに最新のものではない。すなわち、同期は失われている。その原因はたとえば、上述したように漏えいしたリフレッシュトークン及びクラ

イアントクレデンシャルを用いて、本来なら無権限の第三者が要求した不正なリフレッシュ処理などである。
図９のデータ構造に格納されるデータの処理詳細については後述する。
【００６９】
＜アクセストークン発行処理＞
以下、本発明における処理フローについてフローチャートを用いて説明する。
【００７０】
図１０は、アクセス管理サービスシステム１０４が、外部サービスシステム１０３に対して、帳票サービスシステム１０５の利用を許可するアクセストークンを発行するアクセストークン発行フローである。図１０において外部サービスシステム１０３と帳票サービスシステム１０５はそれぞれ別のサービス提供者が運営しているオンラインサービスシステムである。また、帳票サービスシステム１０５はアクセス管理サービス１０４によって、別のサービスを含むユーザーからのアクセスが制御されている。外部サービスシステム１０３は、帳票サービスシステム１０５が提供する帳票データを利用したサービスたとえば印刷サービスをユーザーに対して提供する。
【００７１】
ここで、外部サービスシステム１０３が帳票サービスシステム１０５を利用するためには、外部サービスシステム１０３に対して帳票生成指示を行ったユーザーが、外部サービスシステム１０３および帳票サービスシステム１０５の両方のユーザーでなければならない。また、帳票生成サービスシステム１０５に対して実際に帳票生成リクエストを送信するのは外部サービスシステム１０３である。そのため、外部サービスシステム１０３も帳票サービスシステム１０５のユーザーでなければならない。そのうえで、帳票生成指示を行ったユーザーの権限の範囲内で、帳票サービスシステム１０５を外部サービスシステム１０３が利用できるようにしなくてはならない。具体的には、外部サービスシステム１０３に対してサービスを要求するユーザーは、外部サービスシステム１０３に対して帳票サービスシステム１０５の利用を許可して、外部サービスシステム１０３による帳票サービスシステム１０５の利用の認可をしなければならない。なお、以降の説明において情報機器端末１０２を操作するユーザーを「ユーザー」と呼称する。
【００７２】
ステップＳ１００１において、情報機器端末１０２Ａはウエブブラウザなどのユーザーエージェントを実行しており、ユーザーＡによる操作を受け付けている。ユーザーＡが情報機器端末１０２Ａを操作し、外部サービスシステム１０３に対して帳票生成指示を行うと、その帳票生成指示は、ネットワークを介して外部サービスシステム１０３で実行されている例えばウエブサービスに対して送信される。なお情報機器端末１０２Ａと外部サービスシステム１０３との間は本例ではＨＴＴＰであり、本手順で説明する外部サービスシステム１０３による中核的な処理はそのバックエンドとして実行されるが、これ以降ではＨＴＴＰの部分についての説明は省略する。
【００７３】
ステップＳ１００２において、外部サービスシステム１０３は情報機器端末１０２Ａから帳票生成指示を受け付ける。その後、外部サービスシステム１０３は、帳票サービスシステム１０５に対するアクセストークンを認可情報管理テーブル６００に所持しているかの確認を行う。もし帳票サービスシステム１０５に対するアクセストークンを所持している場合、アクセストークン発行フローを終了する。帳票サービス１０５に対するアクセストークンを所持していない場合、外部サービスシステム１０３はステップＳ１００３において、アクセス管理サービスシステム１０４に対して認可要求を送信する。
【００７４】
ステップＳ１００４において、アクセス管理サービスシステム１０４は外部サービス１０３からの認可要求を受けると、ユーザーＡに対して認証処理を促す認証画面（不図示）を生成し、情報機器端末１０２Ａが備えるＷｅｂブラウザ（不図示）に対して送信して表示させる。

【００７５】
ステップＳ１００５において、ユーザーＡは、情報機器端末１０２ＡのＷｅｂブラウザに表示された認証画面に、ユーザーＩＤとパスワードとを認証情報として入力する。情報機器端末１０２Ａは、アクセス管理サービスシステム１０４に対して入力された認証要求を送る。
【００７６】
ステップＳ１００６において、アクセス管理サービスシステム１０４は情報機器端末１０２から認証要求を受け、ユーザーＩＤおよびパスワードを検証する。具体的には、認証要求に含まれるユーザーＩＤとパスワードの組み合わせが認証データ管理部４０３に格納されたユーザーテーブル７００に登録されているかを判定する。
【００７７】
受信したユーザーＩＤとパスワードの組合せがユーザーテーブル７００に登録されていた場合には、情報機器端末１０２を操作するユーザーＡを帳票サービスシステム１０５のユーザーであると判断し、ステップＳ１００９に進み、処理を継続する。
【００７８】
ステップＳ１００９において、アクセス管理サービス１０４は後述する認可画面１１００を生成し、情報機器端末１０２が備えるＷｅｂブラウザ（不図示）に送信する。
【００７９】
ステップＳ１０１０において、情報機器端末１０２が備えるＷｅｂブラウザは認可画面１１００を受信して表示する。ユーザーが認可画面１１００の認可ボタン１１０２を押下すると、情報機器端末１０２Ａはアクセス管理サービスシステム１０４に対して、認可承認を送信する。
【００８０】
ステップＳ１０１１において、アクセス管理サービス１０４は、受け取った認可承認を基に認可コードを生成し、認可データ管理部４０４で管理される認可コードテーブル８００に、認可したユーザーＩＤと関連付けて格納する。さらにアクセス管理サービス１０４は、情報機器端末１０２Ａが備えるＷｅｂブラウザ（不図示）を外部サービスシステム１０３にリダイレクトさせ、生成した認可コードをステップＳ１００３における要求に対する応答として外部サービスシステム１０３に返す。
【００８１】
ステップＳ１０１２において、外部サービスシステム１０３は受け取った認可コードを認可コードテーブル６１０に格納する。その後、アクセス管理サービスシステム１０４に対して、認可コードと認証情報テーブル６２０に格納されているクライアントＩＤとパスワードとともにアクセストークンの発行要求であるアクセストークン要求を送信する。
【００８２】
ステップＳ１０１３において、アクセス管理サービス１０４はアクセストークン要求を受け、外部サービスシステム１０３の認証を行う。具体的には、アクセストークン要求に含まれるクライアントＩＤとパスワードの組合せが認証データ管理部４０３に格納されたクライアントテーブル７１０に登録されているかを判定する。
【００８３】
クライアントＩＤとパスワードの組み合わせがクライアントテーブル７１０に登録されていた場合には、アクセストークン要求を行った外部サービスシステム１０３を帳票サービスシステム１０５のユーザーであると判断する。アクセス管理サービスシステム１０４は、外部サービスシステム１０３を認証すると、ステップＳ１０１４に進み、処理を継続する。
【００８４】
ステップＳ１０１４において、アクセス管理サービス１０４はアクセストークン要求に含まれる認可コードの検証を行う。具体的には、アクセストークン要求とともに受信した認可コードが認証データ管理部４０３に格納された認可コードテーブル８００に登録されているかを判定する。

【００８５】
認可コードが認可コードテーブル８００に登録されていた場合には、ユーザーによって帳票サービスシステム１０５の利用が許可されていると判断し、ステップＳ１０１６に進み、処理を継続する。
【００８６】
ステップＳ１０１６において、アクセス管理サービスシステム１０４は、アクセストークンおよびリフレッシュトークンを生成し、認可データ管理部４０４で管理される認可情報管理テーブル９００に生成したトークンを格納する。その際、トークンを生成した時刻をアクセストークン発行日時９０２およびリフレッシュトークン発行日時９０５に設定する。また、アクセストークンの有効期間をアクセストークン有効日時９０３に、リフレッシュトークンの有効期間をリフレッシュトークン有効日時９０６に設定する。ユーザー情報として、検証に成功した認可コードを発行したユーザーＩＤをユーザーＩＤ９０７に設定する。そして、初回リフレッシュトークンＩＤ９０８に、最初に発行されたリフレッシュトークン情報を設定する。
【００８７】
本実施形態において、ユーザーが最初に連携システムを利用した際に、アクセストークンおよびリフレッシュトークンが発行された場合は、初回リフレッシュトークンＩＤ９０８には該当するリフレッシュトークンなしを意味する"ｎｕｌｌ"を設定している。ここで、初回リフレッシュトークンＩＤ９０８に、発行したリフレッシュトークン情報としてリフレッシュトークンＩＤ９０４と同じ値を設定してもよい。
【００８８】
その後、アクセス管理サービスシステム１０４は、生成したアクセストークンとリフレッシュトークンをステップＳ１０１３の応答として外部サービス１０３に返す。
【００８９】
ステップＳ１０１７において、外部サービス１０３は受け取ったアクセストークンおよびリフレッシュトークンを、トークン管理部３０５で管理される認可情報管理テーブル６００に格納する。具体的には、受け取ったアクセストークンをアクセストークンＩＤ６０２に設定し、受け取ったリフレッシュトークンをリフレッシュトークンＩＤ６０３および初回リフレッシュトークンＩＤ６０４に設定する。
【００９０】
こうして発行したアクセストークンおよびリフレッシュトークンを保存することで、アクセストークン発行フローを終了する。
【００９１】
一方ステップＳ１００６において、受信したユーザーＩＤとパスワードの組合せがユーザーテーブル７００に登録されていない場合には、情報機器端末１０２を操作するユーザーＡを帳票サービスシステム１０５のユーザーではないと判断する。その後、外部サービスシステム１０３に対し、ステップＳ１００４の応答として認証エラーを返す。ステップＳ１００７において、外部サービスシステム１０３は認証エラーを受け取ると、認証エラー画面（不図示）を生成し、情報機器端末１０２Ａに送る。その後、ステップ１００８において、情報機器端末１０２が備えるＷｅｂブラウザ（不図示）にエラー画面を送信して表示させ、処理を終了する。
【００９２】
またステップＳ１０１３において、クライアントＩＤ、パスワードの組合せがクライアントテーブル７１０に登録されていない場合には、アクセストークン要求を行った外部サービスシステム１０３を帳票サービスシステム１０５のユーザーではないと判断する。その後、認証エラーを外部サービスシステム１０３に送信して、ステップＳ１０１５に進む。
【００９３】
ステップＳ１０１４において、認可コードが認可コードテーブル８００に登録されていなかった場合には、ユーザーによって帳票サービスの利用が許可されていないと判断する

。その後、認可エラーを外部サービスシステム１０３に送信して、ステップＳ１０１５に進む。

【００９４】

ステップＳ１０１５において、外部サービスシステム１０３がアクセス管理サービスシステム１０４から認証エラー、あるいは認可エラーを受け取る。外部サービスシステム１０３は、受け取ったエラーに対応した認証エラー画面あるいは認可エラー画面（不図示）を生成し、情報機器端末１０２に送る。

【００９５】

ステップ１００８において、情報機器端末１０２Ａは、受け取ったエラー画面を表示し、処理を終了する。

【００９６】

以上の手順により発行されたアクセストークンをクライアントすなわち外部サービスシステム１０３で保存管理し、リソースサーバーすなわち帳票サービスシステム１０５へのアクセスのために用いることで、外部サービスシステム１０３は、ユーザーのクレデンシャルの開示を受けることなく、ユーザーの帳票サービスシステム１０５に対するアクセス権限を利用することができる。

【００９７】

＜画面例＞

図１１は、アクセス管理サービスシステム１０４がステップＳ１００４において生成する認可画面を示した図である。認可画面１１００は、情報表示部１１０１と認可ボタン１１０２、認可キャンセルボタン１１０３から構成される。情報表示部１１０１は、認可されるサービスと、認可されたサービスが実行するサービスの情報をユーザーに対して示す情報表示部である。本実施形態においては、認可されるサービスとは外部サービスシステム１０３であり、認可されたサービスが実行するサービスとは帳票サービスシステム１０５を指す。認可ボタン１１０２は、認可を承認する際にユーザーによって押下されるボタンである。認可キャンセルボタン１１０３は、ユーザーが認可を拒否する場合に押下されるボタンである。

【００９８】

＜帳票サービスシステムに対するアクセス手順＞

図１２は、ユーザーが外部サービスシステム１０３に対して、帳票サービスシステム１０５の利用を許可する認可処理のフローを示した図である。図１２の手順は、図１０によるトークンの発行手順を含んでおり、さらに帳票サービスシステム１０５に対して外部サービスシステム１０３がアクセスする手順まで含んでいる。

【００９９】

ステップＳ１００１とＳ１００２は図１０で説明したフローと同じである。また、ステップＳ１２０１は図９におけるステップＳ１００３からＳ１０１７までの、アクセストークンの発行フローを示す。図１２においてはこのステップは外部サービスシステム１０３による処理として示されているが、これは記載上の便宜であって、実際には図１０に記載した通りに、他のシステムと連携してトークンを発行している。ステップＳ１２０１により、要求されたサービスに対するアクセストークンがなければ新たに発行される。

【０１００】

ステップＳ１２０２において、外部サービスシステム１０３は、認可情報管理テーブル６００に格納されている、帳票サービスシステム利用のためのアクセストークンを利用し、帳票サービスシステム１０５に対して帳票生成要求を送る。すなわち、クライアントがリソースサーバーに対してサービスを要求するためのアクセス要求を送信する。ここで「利用」するとは、サービス要求のメッセージとともに、その要求について要求元が認可されていることを示すアクセストークンを要求先に送信することである。ここで、アクセストークン「ＩＪＫＬ９０１２」を渡したとする。

【０１０１】

ステップＳ１２０３において、帳票サービスシステム１０５は、アクセス管理サービス

システム１０４に対して、帳票生成要求で送られたアクセストークンの検証を要求する。
【０１０２】
ステップＳ１２０４において、アクセス管理サービスシステム１０４は、受け取ったアクセストークンの検証を行う。具体的には、受け取ったアクセストークンが認可情報管理テーブル６００に登録されているかの判定を行い、登録されていた場合は、アクセストークンが有効期間内かの判定を行う。アクセストークンが登録されており、かつ有効期間内であった場合は、アクセストークン有効を応答として返す。アクセストークンが登録されていなかった場合や、登録されていたが有効期間外であった場合は、アクセストークン無効を応答として返す。ここで、検証を行った時刻が「２０１１年４月１日１５時」であり、アクセストークン「ＩＪＫＬ９０１２」が渡されたとする。この場合、認可情報管理テーブル９００にはアクセストークン「ＩＪＫＬ９０１２」が登録されているが、アクセストークン有効日時９０３に設定されている時刻を過ぎているため、アクセストークン無効と判断される。
【０１０３】
ステップＳ１２０５において、帳票サービスシステム１０５は、アクセス管理サービス１０４から返されたアクセストークン検証結果を受け取る。アクセストークンが有効の場合、帳票作成機能へのアクセスを許可し、ステップＳ１２０６に進む。アクセストークンが無効の場合、外部サービスシステム１０３に対して、アクセストークン無効をステップＳ１２０３の応答として返し、ステップＳ１２０９に進む。
【０１０４】
ステップＳ１２０６において、帳票サービスシステム１０５は帳票を生成し、生成した帳票データをステップＳ１２０３の応答として外部サービスシステム１０３に返す。
【０１０５】
ステップＳ１２０７において、外部サービスシステム１０３は帳票サービスシステム１０５から帳票データを受け取り、情報機器端末１０２Ａに送信する。
【０１０６】
ステップＳ１２０８において、情報機器端末１０２Ａは、情報機器端末１０２Ａが備えるＷｅｂブラウザ（不図示）に受け取った帳票データを表示し、帳票生成指示を正常に終了する。
【０１０７】
（トークンの再発行処理）
ステップＳ１２０９において、外部サービスシステム１０３は、アクセストークン無効を受けてアクセス管理サービス１０４に対してリフレッシュ処理を要求する。具体的には、認可情報管理テーブル６００に保存されている、帳票サービスシステム１０５のリフレッシュトークンおよび帳票サービスシステム１０５に対するクライアントＩＤ及びパスワードとともに、リフレッシュ処理要求をアクセス管理サービスシステム１０４に送る。ここで、リフレッシュトークン「４５６７ＥＦＧＨ」を渡したとする。
【０１０８】
ステップＳ１２１０において、アクセス管理サービスシステム１０４は、外部サービスシステム１０３の認証を行う。具体的には、リフレッシュ処理要求に含まれるクライアントＩＤとパスワードの組合せが、認証データ管理部４０３に格納されたクライアントテーブル７１０に登録されているかを判定する。
【０１０９】
クライアントＩＤ、パスワードの組合せがクライアントテーブル７１０に登録されていた場合には、帳票サービスシステム１０５が外部サービスシステム１０３との連携を許可していると判断し、ステップＳ１２１３に進み、処理を継続する。
【０１１０】
クライアントＩＤ、パスワードの組合せがクライアントテーブル７１０に登録されていない場合には、外部サービスシステム１０３に対して認証エラーを返し、ステップＳ１２１１に進む。

【０１１１】
ステップＳ１２１１において、外部サービスシステム１０３は認証エラーを受け取ると、認証エラー画面（不図示）を生成し、情報機器端末１０２Ａに送る。その後、ステップ１２１２において、情報機器端末１０２が備えるＷｅｂブラウザ（不図示）にエラー画面を表示し、処理を終了する。
【０１１２】
ステップＳ１２１３において、アクセス管理サービスシステム１０４は、リフレッシュ処理要求で送られたリフレッシュトークンの検証を行う。具体的には、受け取ったリフレッシュトークンが認可情報管理テーブル６００に登録されているかの判定を行い、登録されていた場合は、さらにリフレッシュトークンが有効期間内かの判定を行う。受信したリフレッシュトークンが登録されており、かつ有効期間内であった場合は、リフレッシュトークンを有効と判断し、ステップＳ１２１４に進み、処理を継続する。
【０１１３】
受信したリフレッシュトークンが登録されていないか、あるいは登録されているが有効期間内でなかった場合は、リフレッシュトークンを無効と判断し、外部サービスシステム１０３に対してトークン無効を応答として返し、ステップＳ１２１７に進む。ここで、「２０１１年４月１日１４時３０分」にリフレッシュトークン「４５６７ＥＦＧＨ」が渡されたとする。この場合、認可情報管理テーブル９００にはリフレッシュトークン「４５６７ＥＦＧＨ」が登録されているが、リフレッシュトークン有効日時９０６に設定されている時刻を過ぎているため、リフレッシュトークン無効と判断される。
【０１１４】
ステップＳ１２１４において、アクセス管理サービスシステム１０４は、新たにアクセストークンとリフレッシュトークンを生成し、認可データ管理部４０４で管理される認可情報管理テーブル９００に格納する。その際、認可情報管理テーブル９００から、ステップＳ１２１０で検証したリフレッシュトークンに設定されている初回リフレッシュトークンＩＤ９０８の値を取得し、新たに格納するトークンの初回リフレッシュトークンとして登録する。なお、初回リフレッシュトークンＩＤ９０８の値が"ｎｕｌｌ"であった場合は、検証に利用したリフレッシュトークンを初回リフレッシュトークンとして登録する。
【０１１５】
ステップＳ１２１５において、アクセス管理サービス１０４は、検証に利用したリフレッシュトークンを無効にする。具体的には、認可情報管理テーブル９００の対応するリフレッシュトークンについて、リフレッシュトークン有効日時９０６の値をリフレッシュトークン発行日時９０５の値に更新する。なお、本実施形態ではリフレッシュトークンの有効日時を更新することで無効化を行ったが、他の方法で実施してもよい。例えば、認可情報管理テーブル９００にリフレッシュトークン有効フラグの項目を定義し、その項目の値を更新することで無効化を行ってもよい。その後、アクセス管理サービスシステム１０４は、新たに発行したアクセストークンおよびリフレッシュトークンを、ステップＳ１２１０の応答として外部サービスシステム１０３に返す。
【０１１６】
ステップＳ１２１６において、外部サービスシステム１０３は、認可情報管理テーブル６００に登録されている、帳票サービスシステム１０５のアクセストークンおよびリフレッシュトークンの値を、受け取ったアクセストークンおよびリフレッシュトークンで更新する。具体的には、認可情報管理テーブル６００に登録されている帳票サービスシステムのアクセストークンＩＤ６０２およびリフレッシュトークンＩＤ６０３に、受け取ったアクセストークンとリフレッシュトークンをそれぞれ設定する。この際初回リフレッシュトークンＩＤ６０４は更新しない。
【０１１７】
その後ステップＳ１２０２に進み、外部サービスシステム１０３は再び帳票生成要求を行う。
【０１１８】

ステップＳ１２１４において、外部サービスシステム１０３は、リフレッシュトークン無効応答を受けて、認可情報の無効化を要求する。具体的には、認可情報管理テーブル６００に登録されている帳票サービスシステムの初回リフレッシュトークンの値をアクセス管理サービス１０４に送る。ここで、初回リフレッシュトークンとして「０１２３ＡＢＣＤ」を渡したとする。

【０１１９】

ステップＳ１２１５において、アクセス管理サービス１０４は、後述のトークン無効化処理を行う。

【０１２０】

ステップＳ１２１６において、外部サービスシステム１０３は、帳票サービスシステム１０５の認可情報の削除を行う。具体的には、認可情報管理テーブル６００に登録されている帳票サービスシステム１０５のアクセストークンＩＤ６０２、リフレッシュトークンＩＤ６０３、初回リフレッシュトークンＩＤ６０４の値を削除する。

【０１２１】

その後ステップＳ１２０１に進み、外部サービスシステム１０３は再びアクセストークン発行処理を行う。

【０１２２】

（アクセストークンの無効化処理）

ステップＳ１２１７では、外部サービスシステム１０３が、本実施形態の特徴である、初回リフレッシュトークンを利用したリフレッシュトークンの無効化処理をアクセス管理サービススステム１０４に対して要求する。ここでは、無効化処理要求とともに、認可情報管理テーブル６００に登録された、帳票サービスシステム１０５に対する初回リフレッシュトークンを送信する。無効化処理要求を受信したアクセス管理サービススステム１０４は、ステップＳ１２１８において、受信したリフレッシュトークンが初回リフレッシュトークンとして登録されたリフレッシュトークンの無効化処理を実行する。この詳細は図１３を参照して後述する。リフレッシュトークンの無効化処理を行ったなら、無効化を完了した旨を外部サービスシステム１０３に応答する。応答を受信した外部サービスシステム１０３は、ステップＳ１２１９において、無効化されたリフレッシュトークンを含むクライアント認可関連情報を認可情報管理テーブル６００から削除する。なお、ステップＳ１２１９の時点では、アクセストークンはすでに無効となっている。この後、あらためて外部サービススステム１０３はステップＳ１２０１に進んでアクセストークン及びリフレッシュトークンを新たに発行する処理を行う。なお本例では不要としているが、リフレッシュ処理要求と同様に、帳票サービスシステム１０５に対するクライアントＩＤとパスワードとの組を併せて送信するように構成してもよい。その場合には、アクセス管理サービススステム１０４は、ステップＳ１２１８の直前に、ステップＳ１２１０と同様に権限の認証を行い、認証が成功したならステップＳ１２１８でリフレッシュトークンの無効化を行う。

【０１２３】

＜リフレッシュトークン無効化処理手順＞

図１３は、第１の実施形態におけるリフレッシュトークンの無効化処理、すなわち図１２のステップＳ１２１８の詳細なフローを示した図である。

【０１２４】

ステップＳ１３０１において、アクセス管理サービスシステム１０４は、リフレッシュトークンの無効化要求と共に、リフレッシュトークンを受け付ける。ここで、リフレッシュトークン「０１２３ＡＢＣＤ」が渡されたとする。

【０１２５】

ステップＳ１３０２において、アクセス管理サービス１０４は、受け取ったリフレッシュトークンが初回リフレッシュトークンである全てのリフレッシュトークンを無効とする。具体的には、受け取ったリフレッシュトークンが、認可情報管理テーブル９００の初回リフレッシュトークンＩＤ９０８に登録されているリフレッシュトークンを無効化する。たとえば無効化は、リフレッシュトークン有効日時９０６の値をリフレッシュトークン発

行日時９０５の値に更新することで行う。今回の例では初回リフレッシュトークンＩＤ９０８の値が「０１２３ＡＢＣＤ」であるリフレッシュトークン「４５６７ＥＦＧＨ」、「８９０１ＩＪＫＬ」が対象となる。ただし、リフレッシュトークン「４５６７ＥＦＧＨ」は既に無効となっているため、無効とされるリフレッシュトークンは「８９０１ＩＪＫＬ」のみとなる。

【０１２６】

ステップＳ１３０３において、アクセス管理サービスシステム１０４は、受け取ったリフレッシュトークンを無効とする。具体的には、認可情報管理テーブル９００の対応するリフレッシュトークンについて、リフレッシュトークン有効日時９０６の値をリフレッシュトークン発行日時９０５の値に更新する。なお、最初にアクセストークンとリフレッシュトークンを発行した際に、発行したリフレッシュトークンを初回リフレッシュトークンとして登録している場合には、ステップＳ１３０２において当該リフレッシュトークンも無効化されるので本ステップは省略しても良い。また、初回リフレッシュトークンが"ｎｕｌｌ"であるリフレッシュトークンを無効とするフローは、ＯＡｕｔｈで定義されている従来通りの無効化方式である。今回の例では、リフレッシュトークン「０１２３ＡＢＣＤ」が対象となる。ただし、リフレッシュトークン「０１２３ＡＢＣＤ」は既に無効となっているため、本ステップではリフレッシュトークンの無効化は行われない。

【０１２７】

なお、受信したリフレッシュトークンが、認可情報管理テーブル９００のどの初回リフレッシュトークンにも該当しない場合にはステップＳ１３０２ではなにも行わない。こうすることで、従来のＯＡｕｔｈの手順で無効化要求を発行するクライアントに対する互換性を維持できる。

【０１２８】

これらの処理により、最初に発行されたリフレッシュトークンが元となる一連のリフレッシュ処理で発行されている、全てのリフレッシュトークンが無効となる。

【０１２９】

（本実施形態の効果）

リフレッシュトークンが無効である場合は、リフレッシュ処理がされずに有効期限が経過していた場合と、リフレッシュ処理が行われて新リフレッシュトークンが発行された結果、当該リフレッシュトークンが無効化された場合の２パターンとなる。

【０１３０】

外部サービスシステム１０３には最新のリフレッシュトークンが記憶されているため、リフレッシュ処理がされずに有効期限が経過していた場合には、帳票サービスシステムの全てのリフレッシュトークンが無効となっているはずである。そのため、帳票サービスシステムの全てのリフレッシュトークンを無効にしても、システムに影響は与えない。

【０１３１】

一方、リフレッシュ処理が行われてリフレッシュトークンが無効になっていた場合には、外部サービスシステム１０３が知らないリフレッシュ処理が行われたこととなる。例えば、不正な外部サービスシステム（不図示）がリフレッシュ処理を行った場合などである。この場合、不正な外部サービスシステムが、外部サービスシステム１０３に成り代わって帳票サービスシステム１０５を利用している可能性がある。このような場合でも本実施形態によるリフレッシュトークンの無効化を行う事で、不正に発行したリフレッシュトークンも無効となる。そのため、無効化処理以降の不正な外部サービスシステムによるリフレッシュ処理を防止できる。これにより、帳票サービスが不正な外部サービスシステムによって、不正に利用し続けられてしまうことを防げる。

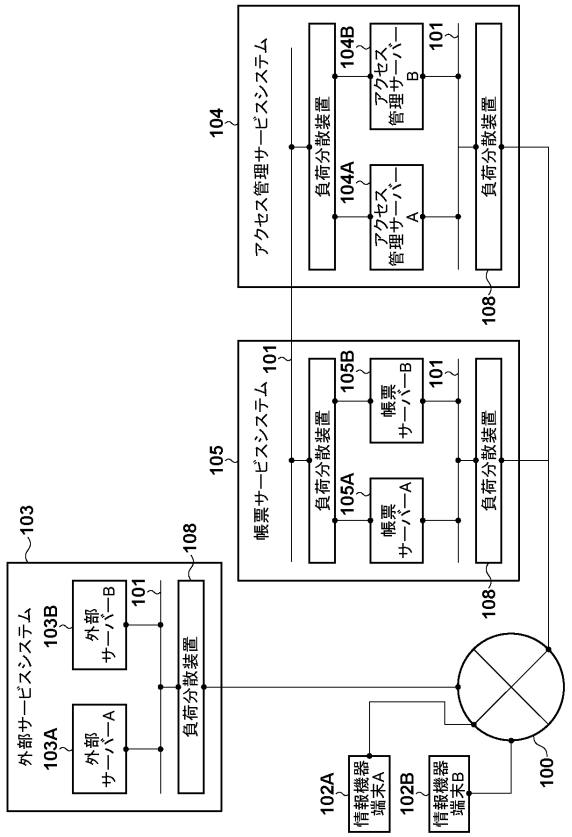
【０１３２】

［実施形態２］

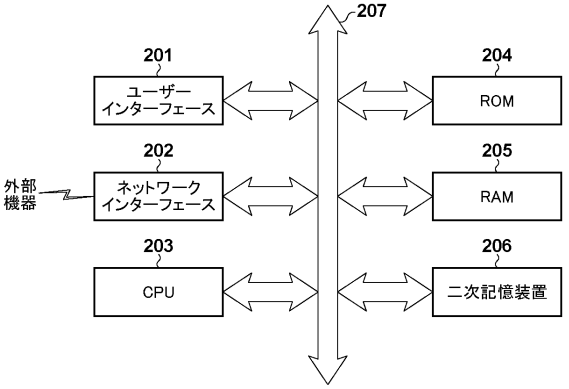
次に、本発明を実施するための第２の実施形態について図面を用いて説明する。第１の実施形態においては、最初に発行されたリフレッシュトークンが元となる一連のリフレッシュ処理で発行されている全てのリフレッシュトークンを無効にしている。

【０１３３】
本実施形態では、リフレッシュトークンを無効にする際に、合わせてアクセストークンも無効とすることで、第１の実施形態において、不正な外部サービスシステムによる成りすましをすぐに防ぐ方法を説明する。なお、本実施形態においては、第１の実施形態と同一部分に関する説明は省略し、その差異についてのみ説明する。なお差異は図１３の手順を図１４の手順に置換したことであり、それについて具体例で説明する。
【０１３４】
＜無効化処理手順＞
図１４は、第２の実施形態における認可情報無効化処理のフローを示した図である。ステップＳ１３０１からＳ１３０３は図１３と同様である。
【０１３５】
ステップＳ１４０１において、アクセス管理サービス１４は、ステップＳ１３０２およびＳ１３０３において無効としたリフレッシュトークンに対応するアクセストークンを無効にする。具体的には、認可情報管理テーブル９００において、無効化対象となったリフレッシュトークンが登録されているデータに対して、アクセストークン有効日時９０３の値をアクセストークン発行日時９０２の値に更新する。なお、本実施形態ではアクセストークンの有効日時を更新することで無効化を行ったが、他の方法で実施してもよい。例えば、アクセストークンテーブル９００にアクセストークン有効フラグの項目を定義し、その項目の値を更新することで無効化を行ってもよい。ここで、ステップＳ１３０１において、「２０１１年４月１日１４時３０分」にリフレッシュトークン「０１２３ＡＢＣＤ」が渡されたとする。この場合、リフレッシュトークン「０１２３ＡＢＣＤ」、「４５６７ＥＦＧＨ」、「８９０１ＩＪＫＬ」が無効化の対象となる。そのため、アクセストークン「ＥＦＧＨ５６７８」、「ＩＪＫＬ９０１２」、「ＭＮＯＰ３４５６」が無効となる。ただし、アクセストークン「ＥＦＧＨ５６７８」、「ＩＪＫＬ９０１２」は既に無効であるため、アクセストークン「ＭＮＯＰ３４５６」のみが無効化される。
【０１３６】
これらの処理により、最初に発行されたリフレッシュトークンを元に行われたリフレッシュ処理で発行された、全てのリフレッシュトークンおよび、全てのアクセストークンが無効になる。
【０１３７】
実施形態１に記載の方法でリフレッシュトークンを無効にすることで、不正な外部サービスシステムからのリフレッシュ処理を防げる。さらに、実施形態２に記載の方法により、アクセストークンの無効化も合わせて行える。これにより、不正な外部サービスシステムによりリフレッシュ処理が行われて、帳票サービスが不正利用されていた場合に、不正な外部サービスシステムからの帳票サービスの利用をすぐに止められる。
【０１３８】
また、本実施形態に係る発明を実装した認可サーバーは、本実施形態に係る発明を実装したクライアントに対応するのみならず、従来のＯＡｕｔｈプロトコルを実装したクライアントに対する互換性も維持することができる。
【０１３９】
［その他の実施例］
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。

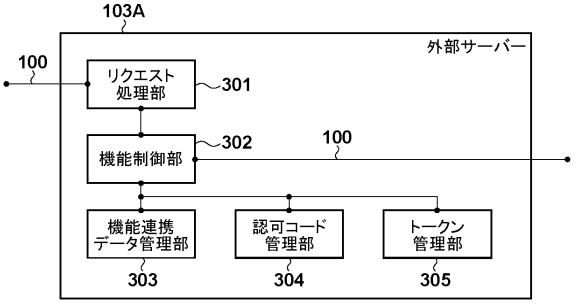
【図１】



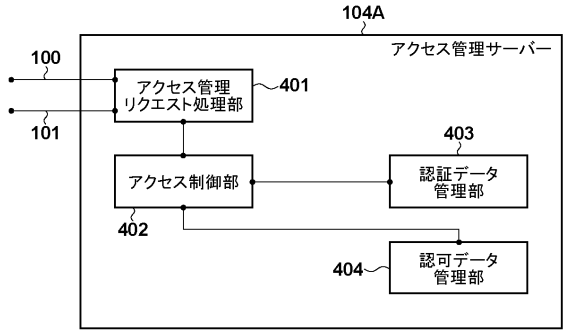
【図２】



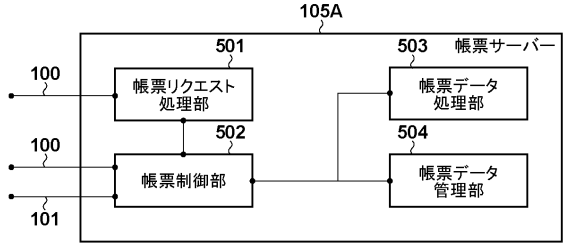
【図３】



【図４】



【図５】



【図６】

600

601 連携先システム名	602 アクセストークンID	603 リフレッシュトークンID	604 初回リフレッシュトークンID
帳票サービスシステム	IJKL9012	4567EFGH	0123ABCD
…	…	…	…

認可情報管理テーブル

610

611 連携先名	612 認可コードID
帳票サービスシステム	ABCD1234
…	…

認可コードテーブル

620

621 連携先名	622 クライアントID	623 パスワード
帳票サービスシステム	System01	**********
…	…	…

認証情報テーブル

【図７】

700

701 ユーザーテーブル 702	
ユーザーID	パスワード
User01	*******
User02	********
…	…

710

711 クライアントテーブル 712	
クライアントID	パスワード
System01	**********
System02	******
…	…

【図８】

800

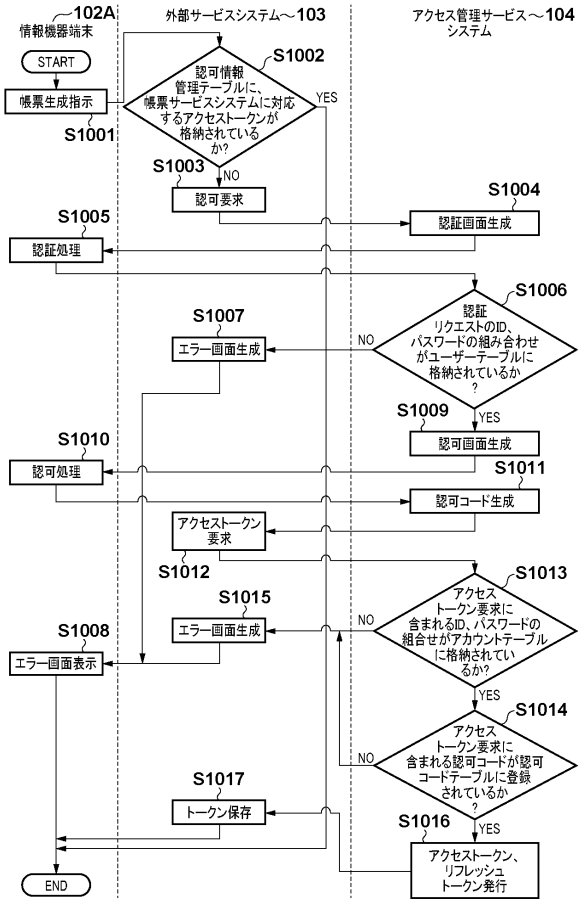
801 認可コードテーブル 802	
認可コードID	ユーザーID
ABCD1234	User01
WXYZ9876	User02
…	…

【図９】

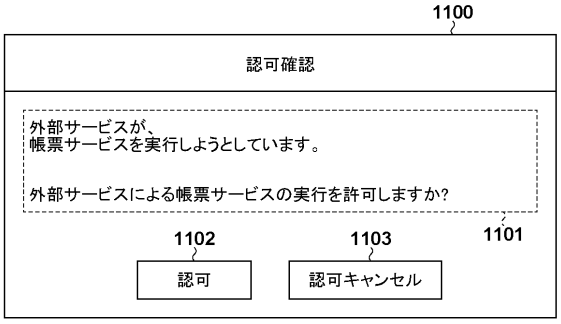
900 認可情報管理テーブル

901 アクセストークンID	902 アクセストークン発行日時	903 アクセストークン有効日時	904 リフレッシュトークンID	905 リフレッシュトークン発行日時	906 リフレッシュトークン有効日時	907 ユーザーID	908 初回リフレッシュトークンID
EFGH5678	20120401120000	20120401130000	0123ABCD	20120401120000	20120501120000	User01	null
IJKL9012	20120401130000	20120401140000	4567EFGH	20120401130000	20120501130000	User01	0123ABCD
MNOP3456	20120401140000	20120401150000	8901IJKL	20120401140000	20120501140000	User01	0123ABCD
QRST7890	20120401120000	20120401130000	9876ZYXW	20120401120000	20120501120000	User02	null
UVWX1234	20120401130000	20120401140000	5432VUTS	20120401130000	20120501130000	User02	9876ZYXW
…	…	…	…	…	…	…	…

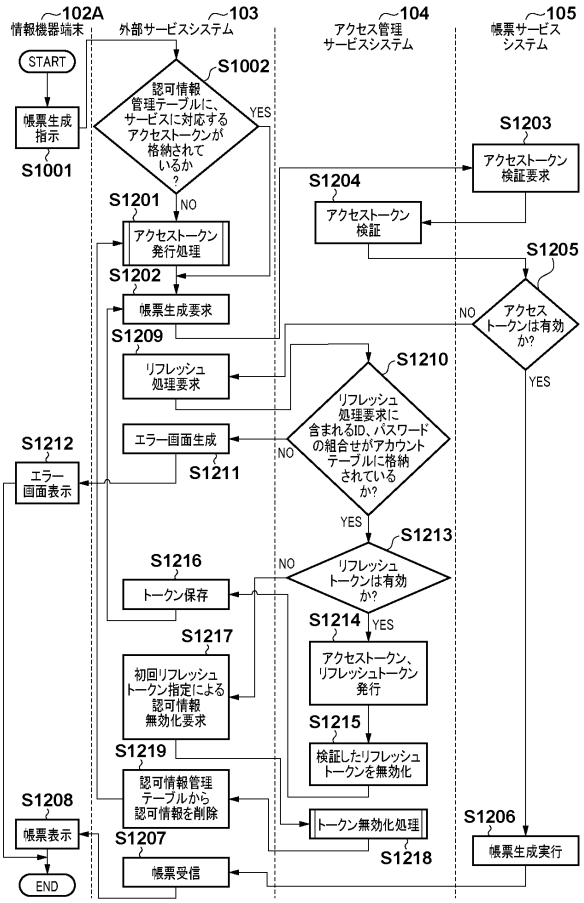
【図１０】



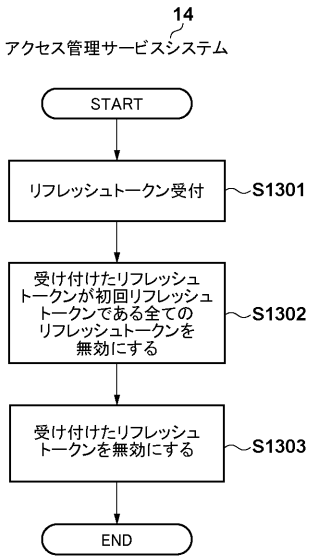
【図１１】



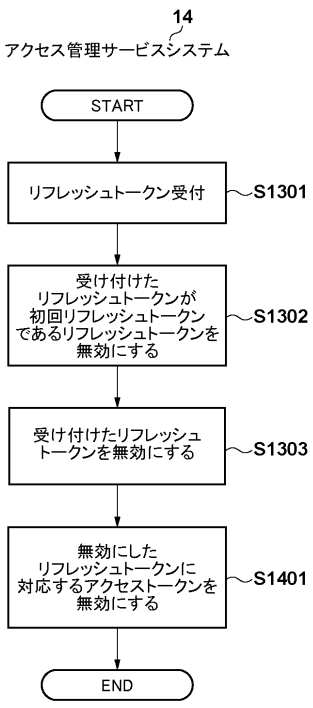
【図１２】



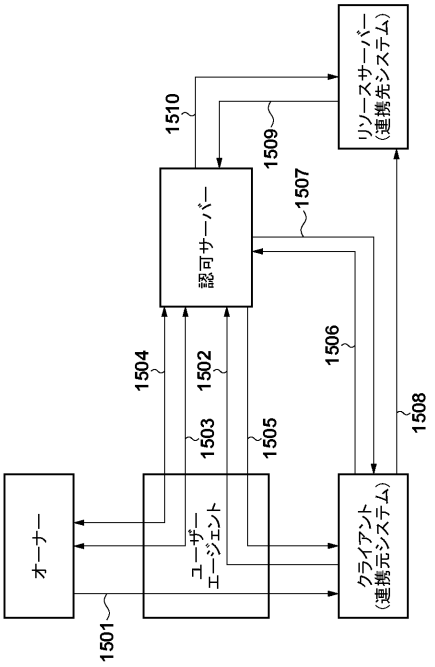
【図１３】



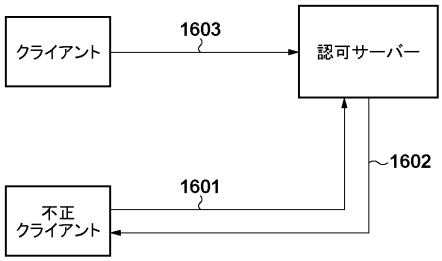
【図１４】



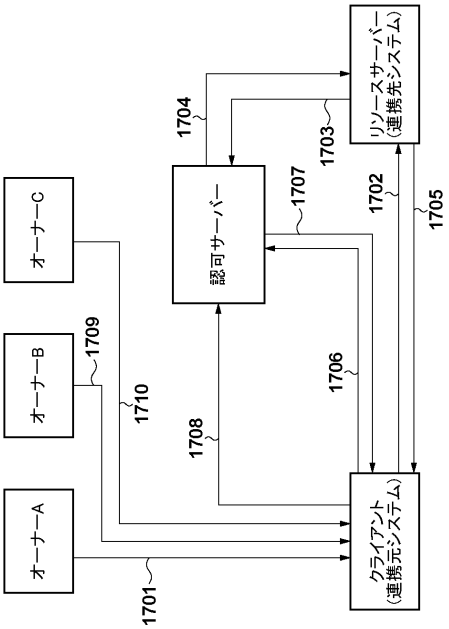
【図１５】


リソースサーバー
（連携先システム）
1510
1509
認可サーバー
1507
1504
1502
1506
1503
1505
1508
オーナー
ユーザー
エージェント
クライアント
（連携元システム）
1501

【図１６】


クライアント
1603
認可サーバー
1602
1601
不正
クライアント

【図１７】


リソースサーバー
（連携先システム）
1704
1703
認可サーバー
1707
1702
1705
オーナーC
オーナーB
1709
1706
1710
1708
オーナーA
クライアント
（連携元システム）
1701

フロントページの続き

(72)発明者　茂垣　俊介
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　宮司　卓佳

(56)参考文献　特開２００５－０９９９８０（ＪＰ，Ａ）
米国特許出願公開第２００９／０２８２２３９（ＵＳ，Ａ１）
Maciej P.Machulak他, User-Managed Access to Web Resources, Proceeding DIM'10 Proceedings of the 6th ACM workshop on Digital identity management, ２０１０年, p.35-p.44

(58)調査した分野(Int.Cl., ＤＢ名)
Ｇ０６Ｆ　２１／３０－Ｇ０６Ｆ２１／４６
Ｈ０４Ｌ　９／３２