(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6025480号
(P6025480)
(45)発行日　平成28年11月16日(2016.11.16)　(24)登録日　平成28年10月21日(2016.10.21)

(51)Int.Cl.　　F I
G06F 21/33 (2013.01)　　G06F 21/33 350
G06F 21/45 (2013.01)　　G06F 21/45

請求項の数 14　(全 24 頁)

(21)出願番号　特願2012-214267(P2012-214267)
(22)出願日　平成24年9月27日(2012.9.27)
(65)公開番号　特開2014-67378(P2014-67378A)
(43)公開日　平成26年4月17日(2014.4.17)
審査請求日　平成27年9月28日(2015.9.28)

(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　100126240
弁理士　阿部　琢磨
(74)代理人　100124442
弁理士　黒岩　創吾
(72)発明者　田村　存
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　青木　重徳

最終頁に続く

(54)【発明の名称】認可サーバーシステム、権限移譲システム、その制御方法、およびプログラム

(57)【特許請求の範囲】
【請求項１】
ネットワークを介して接続されたデバイス機器へサービスを提供するサーバーシステムと、前記サービスを利用するアプリケーションを備えた前記デバイス機器と通信することが可能な認可サーバーシステムであって、
前記アプリケーションに対して前記サービスを利用する権限を移譲するための要求を受信するとともに、前記サービスを利用する権限を特定するための第１の権限情報を受信する受信手段と、
前記要求が受信されたことに応じて、前記要求とともに前記受信手段により受信された第１の権限情報を基に権限を特定し、特定された権限内の少なくとも一部の権限を前記アプリケーションに対し与え、与えられた権限を特定するための第２の権限情報を発行する発行手段と、
前記発行手段により発行された第２の権限情報を前記要求の要求元へ送信する送信手段と、を有する認可サーバーシステム。
【請求項２】
前記発行手段は、特定された権限内の少なくとも一部の権限を前記アプリケーションに対し与える際に、要求元から要求された権限が前記特定された権限内の権限であるか否かを判断し、前記特定された権限内の権限であると判断したことに応じて、要求された権限を前記アプリケーションに対し与え、与えられた権限を特定するための第２の権限情報を発行することを特徴とする請求項１に記載の認可サーバーシステム。

【請求項３】
前記デバイス機器は、前記アプリケーションに代わり、前記アプリケーションに対して前記サービスを利用する権限を移譲するための要求を行うクライアント手段を更に有し、
前記受信手段は、前記クライアント手段に対して前記サービスを利用する権限を移譲するための要求を受信し、
前記発行手段は、前記デバイス機器を操作するユーザーの権限を前記クライアント手段に対し与え、与えられた権限を特定するための前記第１の権限情報を発行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システム。
【請求項４】
前記認可サーバーシステムは、認可操作を行うための認可画面をユーザーが操作するブラウザへ提供する提供手段を更に有し、
前記提供手段は、前記クライアントに対して前記サービスを利用する権限を移譲するための要求が受信されたことに応じて、前記認可画面をブラウザへ提供し、
前記発行手段は、ユーザーが前記認可画面を用いて認可操作を行ったことに応じて、前記ユーザーの権限を前記クライアントに対し与え、与えられた権限を特定するための前記第１の権限情報を発行することを特徴とする請求項３に記載の認可サーバーシステム。
【請求項５】
前記発行手段は、前記アプリケーションに与える権限に制限を設けるための前記第１の権限情報、および前記アプリケーションに与える権限に制限を設けないための前記第１の権限情報の両方を発行することが可能であって、
更に、前記受信手段により前記アプリケーションに与える権限に制限を設けないための前記第１の権限情報が受信された場合、前記発行手段は、要求元から要求された権限が前記特定された権限内の権限であるか否かを判断することなく、要求された権限を前記アプリケーションに対し与えることを特徴とする請求項３または４に記載の認可サーバーシステム
【請求項６】
前記受信手段により前記第１の権限情報が受信された場合、前記発行手段は、前記第１の権限情報を基にユーザーを特定し、特定されたユーザーは前記アプリケーションに対して与えられる権限を有するユーザーであるか否かを判断し、前記アプリケーションに対して与えられる権限を有するユーザーであると判断したことに応じて前記第２の権限情報を発行することを特徴とする請求項３乃至５の何れか１項に記載の認可サーバーシステム。
【請求項７】
前記発行手段は、前記クライアント手段から受信した前記クライアント手段のスコープを基に、前記アプリケーションに対し与える権限と同じ権限が前記クライアント手段に与えられているか否かを判断し、前記クライアント手段に与えられていると判断したことに応じて前記第２の権限情報を発行することを特徴とする請求項３乃至６の何れか１項に記載の認可サーバーシステム。
【請求項８】
前記受信手段は、前記アプリケーションが前記サービスを利用する際に送信した前記第２の権限情報を、前記サーバーシステムを介して受信し、
前記送信手段は、受信された前記第２の権限情報を基に特定した権限に関する情報を送信し、
前記サーバーシステムは、受信した前記権限に関する情報を基に権限を確認し、確認された権限で前記サービスを前記アプリケーションに対して利用させることを特徴とする請求項１乃至７の何れか１項に記載の認可サーバーシステム。
【請求項９】
前記デバイス機器は印刷部、およびスキャナ部の内、少なくとも１つの画像処理部を搭載した画像形成装置であり、
前記サーバーシステムは、印刷サービス、および帳票サービスの内、少なくとも１つの画像処理サービスを提供するサーバーシステムであり、

前記認可サーバーシステムは、前記画像形成装置、および前記サーバーシステムと通信することが可能であることを特徴とする請求項１乃至８の何れか１項に記載の認可サーバーシステム。

【請求項１０】

前記デバイス機器はスマートフォンであり、

前記認可サーバーシステムは、前記スマートフォンと通信することが可能であることを特徴とする請求項１乃至９の何れか１項に記載の認可サーバーシステム。

【請求項１１】

ネットワークを介して接続されたデバイス機器へサービスを提供するサーバーシステムと、前記サービスを利用するアプリケーションを備えた前記デバイス機器と通信することが可能な認可サーバーシステムを制御する制御方法であって、

受信手段は、前記アプリケーションに対して前記サービスを利用する権限を移譲するための要求を受信するとともに、前記サービスを利用する権限を特定するための第１の権限情報を受信し、

発行手段は、前記要求が受信されたことに応じて、前記要求とともに前記受信手段により受信された第１の権限情報を基に権限を特定し、特定された権限内の少なくとも一部の権限を前記アプリケーションに対し与え、与えられた権限を特定するための第２の権限情報を発行し、

送信手段は、前記発行手段により発行された第２の権限情報を前記要求の要求元へ送信することを特徴とする制御方法。

【請求項１２】

ネットワークを介して接続されたデバイス機器へサービスを提供するサーバーシステムと、前記サービスを利用するアプリケーションを備えた前記デバイス機器と通信することが可能な認可サーバーシステムで実行されるプログラムであって、

前記アプリケーションに対して前記サービスを利用する権限を移譲するための要求を受信するとともに、前記サービスを利用する権限を特定するための第１の権限情報を受信する受信ステップと、

前記要求が受信されたことに応じて、前記要求とともに前記受信ステップにおいて受信された第１の権限情報を基に権限を特定し、特定された権限内の少なくとも一部の権限を前記アプリケーションに対し与え、与えられた権限を特定するための第２の権限情報を発行する発行ステップと、

前記発行ステップにおいて発行された第２の権限情報を前記要求の要求元へ送信する送信ステップと、を有するプログラム。

【請求項１３】

ネットワークを介して接続されたデバイス機器へサービスを提供するサーバーシステムと、前記サービスを利用するアプリケーションを備えた前記デバイス機器と、認可サーバーシステムと、を含む権限移譲システムであって、

前記デバイス機器は、

前記アプリケーションに対して前記サービスを利用する権限を移譲するための要求を送信するとともに、前記サービスを利用する権限を特定するための第１の権限情報を送信する第１の送信手段を有し、

前記認可サーバーシステムは、

前記アプリケーションに対して前記サービスを利用する権限を移譲するための要求を受信するとともに、前記サービスを利用する権限を特定するための第１の権限情報を受信する受信手段と、

前記要求が受信されたことに応じて、前記要求とともに前記受信手段により受信された第１の権限情報を基に権限を特定し、特定された権限内の少なくとも一部の権限を前記アプリケーションに対し与え、与えられた権限を特定するための第２の権限情報を発行する発行手段と、

前記発行手段により発行された第２の権限情報を前記要求の要求元へ送信する第２の送

信手段と、を有し、
前記サーバーシステムは、
前記要求元から取得した前記第２の権限情報を基に権限を確認し、前記アプリケーションに対してサービスを提供する提供手段を有することを特徴とする権限移譲システム。
【請求項１４】
ネットワークを介して接続されたデバイス機器へサービスを提供するサーバーシステムと、前記サービスを利用するアプリケーションを備えた前記デバイス機器と、を含む権限移譲システムを制御する制御方法であって、
前記サービスにおけるユーザーの権限を前記デバイス機器に移譲させる第１の移譲ステップと、
前記デバイス機器に移譲された権限を更に前記アプリケーションに移譲する場合、前記第１の権限移譲ステップにおいてデバイス機器に移譲された権限の中に前記アプリケーションが必要とする権限が含まれていることに応じて、前記デバイス機器に移譲された権限の内、前記アプリケーションに必要な権限を前記アプリケーションへ移譲させる第２の移譲ステップと、を含む制御方法。
【発明の詳細な説明】
【技術分野】
【０００１】
権限移譲を行う認可サーバーシステム、権限移譲システム、その制御方法、およびプログラムに関する。
【背景技術】
【０００２】
近年、ＰＤＦ形式の電子文書を作成するサービス、およびその電子文書を保存するサービスをインターネットを介して端末に提供するサーバーが広く運営されるようになった。ユーザーは端末を介してサービスを利用することで、その端末に電子文書の作成機能がない場合でも電子文書が作成できるようになる他、端末の記憶容量以上の電子文書を保存できるようにもなる。
【０００３】
更に、クラウドが注目されるに伴い、複数のサービスを連携させて新たな付加価値を創造する機会は益々増加している。例えば、サービスを利用し作成したＰＤＦ形式の電子文書が端末を経由することなく別のサービスに直接保存される、と言ったサービスの連携が可能である。その一方で、サービスが連携することにより課題が生まれる。
【０００４】
それは、ユーザーが望んだ以上の情報がサービス間で交換された場合、ユーザーデータや個人情報の漏えいリスクが高まると言うことである。サービス連携時に、ユーザーの望む結果を提供するためのサービス以外のサービスがユーザーデータ、個人情報等を取得することは望ましくない。一方で、サービスの提供者からするとサービス連携の仕組みは容易に実装できるものが好ましい。
【０００５】
このような状況において、ＯＡｕｔｈ（特許文献１、非特許文献１、および非特許文献２を参照）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が策定されている。ＯＡｕｔｈを複数のサービスに実装することで、例えば、ユーザーから特定の権限でサービスＡへのアクセスが認められた外部サービスＢは、ユーザーの認証情報を用いることなくサービスＡへアクセスできるようになる。このとき、サービスＡは外部サービスＢからアクセスされるデータ、および利用されるサービスの範囲と言った権限の内容をユーザーに対し明らかにした上で、外部サービスＢによるアクセスにユーザーの明示的な了承、即ち認可を必要とする構成になっている。ユーザーが認可画面を介して明示的に認可を行う行為を認可操作と言う。
【０００６】
ユーザーが認可操作を行うと、サービスＡは外部サービスＢに対してユーザーによって

認可された特定の権限を与える。そして、外部サービスＢは、認可された権限でアクセスが認められることを証明するトークン、即ち認可トークンをサービスＡから直接的、または間接的に受け取る。以降、外部サービスＢはその認可トークンを用いてサービスＡにアクセスできる。ユーザーが認可操作を行った結果、サービスを利用する主体、先程の例で言えば外部サービスＢに認可トークンを保持させるための一連の流れを、ユーザーは外部サービスＢに権限を移譲する、と言う。なお、上述した通り、サービスを利用する主体に実際の権限を与えるのはサービスを利用させる主体であり、先程の例で言えば、ユーザーから認可操作が行われたことを確認したサービスＡが権限を与える。

【０００７】

なお、この技術はサービス間の連携にのみ使用されるわけではなく、ユーザーが操作する端末のアプリケーションがＯＡｕｔｈを用いてインターネット上のサービスと連携する形態でも利用されていることが知られている。例えば、アプリケーションの追加・削除が可能であるスマートフォンに複数のアプリケーションをインストールし、各アプリケーションがインターネット上のサービスと連携する形態である。その中でも、ソーシャル・ネットワーキング・サービス（以下、ＳＮＳと呼ぶ）と呼ばれるサービスと、スマートフォンのアプリケーションがＯＡｕｔｈを用いて連携する形態が代表的な例である。

【０００８】

スマートフォンにインストールされたアプリケーションは、ユーザーの代理としてＳＮＳにアクセスすることになる。ユーザーは、ＳＮＳを利用するのに必要な最小限の機能の権限、例えば記事を投稿する権限のみをアプリケーションに移譲することで、スマートフォンにＳＮＳの認証情報を保存させることなく、かつアプリケーションは適正な権限でＳＮＳと連携することができる。

【先行技術文献】

【特許文献】

【０００９】

【特許文献１】特開２０１２－００８９５８号公報

【非特許文献】

【００１０】

【非特許文献1】“Ｔｈｅ ＯＡｕｔｈ １．０ Ｐｒｏｔｏｃｏｌ”、［ｏｎｌｉｎｅ］ Ｅ．Ｈａｍｍｅｒ－Ｌａｈａｖ、２０１２年9月 <ＵＲＬ ｈｔｔｐ：／／ｔｏｏｌｓ．ｉｅｔｆ．ｏｒｇ／ｈｔｍｌ／ｒｆｃ５８４９>

【非特許文献2】“Ｔｈｅ ＯＡｕｔｈ ２．０ Ａｕｔｈｏｒｉｚａｔｉｏｎ Ｆｒａｍｅｗｏｒｋ ｄｒａｆｔ－ｉｅｔｆ－ｏａｕｔｈ－ｖ２－３１”、［ｏｎｌｉｎｅ］ Ｄ．Ｈａｒｄｔ．、２０１２年9月 <ＵＲＬ ｈｔｔｐ：／／ｔｏｏｌｓ．ｉｅｔｆ．ｏｒｇ／ｈｔｍｌ／ｄｒａｆｔ－ｉｅｔｆ－ｏａｕｔｈ－ｖ２－３１>

【発明の概要】

【発明が解決しようとする課題】

【００１１】

ＯＡｕｔｈのような権限移譲プロトコルを用いて端末上の各アプリケーションに権限を移譲する場合、従来技術では、アプリケーションが端末に追加される毎にユーザーは各アプリケーションに対する認可操作を行わないといけないので利便性に欠ける。そこで、各アプリケーションに対する認可操作をより少ない回数、例えば、１回の認可操作で各アプリケーションに権限を移譲させる仕組みを端末、およびサービスに導入することが将来的に考えられる。また、各アプリケーションは異なる目的で作成されているため利用するクラウドサービスが異なるので、各アプリケーションが必要とする権限は多様となることが予想される。

【００１２】

しかし、ユーザーの認可操作を介さずに各アプリケーションに権限を移譲する場合の仕組みが考えられていないので、アプリケーションに対して過剰な権限を移譲してしまう恐れがある。

【００１３】
仮に、過剰な権限が悪意のあるアプリケーションに移譲されてしまい、そのアプリケーションがリソースサーバーにアクセスしてきた場合、クラウドサービスに保管したデータが漏えいする、または破壊されるというリスクが発生する。また、例え、悪意はない開発者によりアプリケーションが開発された場合であっても、そのアプリケーションにバグがあり予期せぬ動作をすれば、上述したリスクと同等のリスクが発生する。
本発明では上述した課題を解決することを目的の１つとする。
【課題を解決するための手段】
【００１４】
本願発明の目的を達成する認可サーバーシステムは、ネットワークを介して接続されたデバイス機器へサービスを提供するサーバーシステムと、前記サービスを利用するアプリケーションを備えた前記デバイス機器と通信することが可能な認可サーバーシステムであって、前記アプリケーションに対して前記サービスを利用する権限を移譲するための要求を受信するとともに、前記サービスを利用する権限を特定するための第１の権限情報を受信する受信手段と、前記要求が受信されたことに応じて、前記要求とともに前記受信手段により受信された第１の権限情報を基に権限を特定し、特定された権限内の少なくとも一部の権限を前記アプリケーションに対し与え、与えられた権限を特定するための第２の権限情報を発行する発行手段と、前記発行手段により発行された第２の権限情報を前記要求の要求元へ送信することを特徴とする。
【発明の効果】
【００１５】
本願発明により、アプリケーションに対して過剰な権限を移譲することを防ぐことが可能となる。
【図面の簡単な説明】
【００１６】
【図１】認可サーバー２００、リソースサーバー２１０、画像形成装置３００から構成される権限移譲システムを示す図。
【図２】本願発明に係る認可サーバー２００、リソースサーバー２１０、画像形成装置３００のハードウェア構成図。
【図３】本願発明に係る認可サーバー２００、リソースサーバー２１０、画像形成装置３００のモジュール構成図。
【図４】本願発明に係る画像形成装置３００で実行されるトークン発行の要求フロー。
【図５】本願発明に係る認可サーバー２００で実行される親トークン発行のフロー。
【図６】本願発明に係る認可サーバー２００で実行される子トークン発行のフロー。
【図７】本願発明に係るリソースサーバー２１０で実行されるリソースアクセスを受けた際の処理フローである。
【図８】本願発明に係る画像形成装置３００が取得済みのトークンを示すトークンテーブル。
【図９】本願発明に係る認可サーバー連携クライアント４００が有する認可サーバー連携情報。
【図１０】本願発明に係る認可サーバー２００が有する権限テーブル。
【図１１】本願発明に係る認可サーバー２００が有する発行済みトークンテーブル。
【図１２】本願発明に係る認証画面、および認可画面。
【発明を実施するための形態】
【００１７】
始めに、本願発明の実施形態の概要と、用語の説明を行う。
【００１８】
本願発明の実施形態においては、インターネット上で帳票データを生成する帳票サービスと、インターネット上のデータを取得して印刷する印刷サービスとがインターネット上のサーバーに設置されていることを想定している。以降、帳票サービスや印刷サービスの

ように、ネットワークからデバイス機器へ提供されている機能をリソースサービスと呼ぶ。

【００１９】

本願発明の実施形態においては端末であるデバイス機器として画像形成装置を例に説明する。画像形成装置上にインストールされた印刷アプリケーション、および帳票アプリケーションがリソースサービスを利用することを想定している。以降、印刷アプリケーションや帳票アプリケーションのようにリソースサービスを利用するアプリケーションをリソースサービス連携アプリケーションと呼ぶ。

【００２０】

更に、本願発明の実施形態における権限の移譲の処理にはＯＡｕｔｈの仕組みを利用することとする。ＯＡｕｔｈでは、ユーザーから移譲された権限の範囲を特定するための情報をトークンと呼ばれるデータ形式で表現する。サービスを利用される主体、即ちリソースサービスを提供するサーバーは、トークンに基づいて特定される権限の中で外部からのアクセスを許す。トークンを始めとする権限の範囲を特定するために用いられる情報を本願発明では権限情報と称する。また、サービスを利用する主体、即ち画像形成装置、またはリソースサービス連携アプリケーションに移譲される権限の範囲を規定する情報をスコープと称し、ユーザーが画像形成装置に対して権限移譲した場合に発行されるトークンを親トークンと称する。本願発明の実施形態では、ユーザーの権限を画像形成装置のようなデバイス機器に移譲することもポイントの１つである。

【００２１】

例えば、画像形成装置上に印刷アプリケーション、および帳票アプリケーションが存在する場合を考える。従来技術では、ユーザーは、印刷アプリケーションからリソースサービスを利用する場合は印刷アプリケーションに、帳票アプリケーションからリソースサービスを利用する場合は帳票アプリケーションに対してそれぞれ個別に認可操作を行う必要がある。ユーザーの立場で考えると、同一の画像形成装置からリソースサービスを利用する場合には、例えば一度の認可操作でそれぞれのアプリケーションからリソースサービスを利用できるようになった方が良い。そこで、アプリケーションに権限を移譲する際、ユーザーに代わり画像形成装置がアプリケーションに権限を移譲することでユーザーの認可操作の回数を低減させる。ユーザーは画像形成装置に権限を移譲した段階で、アプリケーションにも権限を移譲することを暗に認可したこととなる。

【００２２】

なお、本願発明の実施形態では画像形成装置に権限を移譲する際にユーザーに認可操作を行わせるが、アプリケーションに権限を移譲する際にはユーザーに認可操作を行わせない。以上の様に、本願発明では、ユーザーの権限をデバイス機器に移譲することでユーザーの認可操作を従来よりも少ない回数に減らしつつ、将来的に追加される各アプリケーションへの権限移譲を可能にする。

【００２３】

ここで、アプリケーションへ権限移譲をする方法について説明する。アプリケーションへ権限移譲をする方法として始めに思いつく構成は、画像形成装置が取得した親トークンをアプリケーションで共有する形態にする形態であろう。しかし、アプリケーションが親トークンを共有する形態を取った場合、全てのアプリケーションに全く同じ権限が与えられたことになってしまいセキュリティ上好ましくない。セキュリティ上好ましくない理由は、上述の本願発明の課題の項目で述べた通りである。

【００２４】

例えば、印刷アプリケーションは印刷権限が必要だが帳票生成権限は不要である。また帳票アプリケーションは帳票生成権限が必要だが印刷権限は不要である。このように個々のアプリケーションがそれぞれ異なる目的のためのものであれば、それぞれのアプリケーションに必要な権限は異なるため、夫々のアプリケーションには出来る限り必要最低限の権限を与える形態がセキュリティ上は好ましい。

【００２５】

そこで本願発明の実施形態では、リソースサービス連携アプリケーションに親トークンを直接使わせるのではなく、画像形成装置に移譲された権限の内、リソースサービス連携アプリケーションが本来必要とする権限に限定したトークンを発行することが発明のポイントの１つである。このトークンを子トークンと呼び、画像形成装置からリソースサービス連携アプリケーションへ更に権限を移譲した場合に発行されるトークンである。

【００２６】

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。

【００２７】

＜実施例１＞

本実施例では、ユーザー、デバイス機器、デバイス機器上のアプリケーションの３者間で権限を移譲させていく際に、デバイス機器上のアプリケーションに過剰な権限が移譲されないようにするための具体的な構成を説明する。

【００２８】

本実施例に係る権限移譲システムは、図１に示すようなネットワーク上に実現される。１００は、Ｗｉｄｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ（ＷＡＮ１００）であり、本発明ではＷｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ（ＷＷＷ）システムが構築されている。１０１は各構成要素を接続するＬｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ（ＬＡＮ１０１）である。

【００２９】

２００はＯＡｕｔｈを実現するための認可サーバーであり、認可サービスモジュールを備えている。２１０はリソースサーバーであり、印刷サービスや帳票サービスといったリソースサービスを備えている。なお１台のリソースサーバーに備えられるリソースサービスは１つであっても、複数でもよい。また、認可サーバー、およびリソースサーバーは１台ではなく複数台で構成されるサーバー群であっても良い。認可サーバーシステムと称した場合、認可サービスモジュールを備えた１台のサーバー、または複数台のサーバー群を指しており、リソースサーバー２１０、および後述する画像形成装置３００は含まないサーバー群を意味する。リソースサーバーシステムについても同様であり、リソースサービスを備えた１台のサーバー、または複数台のサーバーを指している。サーバーシステムが複数台で構成される場合、サーバーシステムに備えられたモジュールは複数台のサーバーに分割して配置し、モジュールの一部を備えた複数台のサーバー同士が連携して機能を実行するようにしても良い。

【００３０】

３００は画像形成装置であり、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いる。ユーザーはそれらのリソースサービス連携アプリケーションを用いてリソースサービスを利用する。また認可サーバー２００、リソースサービス２１０、画像形成装置３００はそれぞれＷＡＮネットワーク１００、およびＬＡＮ１０１を介して接続されている。なお認可サーバー２００、リソースサービス２１０、画像形成装置３００はそれぞれ個別のＬＡＮ上に構成されていてもよいし、同一のＬＡＮ上に構成されていてもよい。また、認可サーバー２００、リソースサービス２１０は同一のサーバー上に構成されていてもよい。

【００３１】

本実施例に係る権限移譲システムは、図２に示すようなハードウェア構成のサーバー、および画像形成装置から成るシステム上に実現される。図２は、認可サーバー２００、またはリソースサーバー２１０と、画像形成装置３００とがネットワークなどを介して通信可能に接続されている様子を示す。

【００３２】

まず、認可サーバー２００の構成について説明する。なお、図２に示されるハードウェアブロック図は一般的な情報処理装置のハードウェアブロック図に相当するものとし、本実施例の認可サーバー２００には一般的な情報処理装置のハードウェア構成を適用できる。また、リソースサーバー２１０についても同様である。図２において、ＣＰＵ２０１は、ＲＯＭ２０３のプログラム用ＲＯＭに記憶された、或いはハードディスク２１１からＲ

ＡＭ２０２にロードされたＯＳ、またはアプリケーション等のプログラムを実行する。ここでＯＳとはコンピュータ上で稼動するオペレーティングシステムの略語であり、以下オペレーティングシステムのことをＯＳと呼ぶ。後述する各フローチャートの処理はこのプログラムの実行により実現できる。ＲＡＭ２０２は、ＣＰＵ２０１のワークエリアとして機能する。

【００３３】

キーボードコントローラ（ＫＢＣ）２０５は、キーボード２０９や不図示のポインティングデバイスからのキー入力を制御する。ＣＲＴコントローラ（ＣＲＴＣ）２０６は、ＣＲＴディスプレイ２１０の表示を制御する。ディスクコントローラ（ＤＫＣ）２０７は各種データを記憶するハードディスク（ＨＤ）２１１やフロッピー（登録商標）ディスク（ＦＤ）等におけるデータアクセスを制御する。ＰＲＴＣ２０８は、接続された画像形成装置３００との間の信号の交換を制御する。ネットワークカード（ＮＣ）２１２はネットワークに接続されて、ネットワークに接続された画像形成装置や他の機器との通信制御処理を実行する。なお、後述の説明においては、特に断りのない限りサーバーにおける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ハードディスク（ＨＤ）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。

【００３４】

次に、画像形成装置３００の構成について説明する。３０１は画像形成装置３００のＣＰＵであり、ＲＯＭ３０２、または外部メモリ３０３に記憶された制御プログラムに基づいてシステムバス３０４に接続される各ブロックを制御する。ＣＰＵ３０１の処理により生成された画像信号が印刷部Ｉ／Ｆ３０５を介して、印刷部（画像形成装置エンジン）３０６に出力情報として出力される。また、原稿を読み取るためのスキャナ部（不図示）を備えていても良く、画像形成装置３００は印刷部、またはスキャナ部の内、少なくとも一方の画像処理部を有しているものとする。ＣＰＵ３０１は、入力部３０７を介して認可サーバー２００との通信処理が可能となっており、画像形成装置３００内の情報等を認可サーバー２００に通知できる。

【００３５】

ＲＯＭ３０２内のプログラムＲＯＭには、ＣＰＵ３０１の制御プログラム等を記憶している。ＲＯＭ３０２内のフォント用ＲＯＭには、出力情報を生成する際に使用するフォントデータ等を記憶している。ＲＯＭ３０２内のデータ用ＲＯＭには、ハードディスク等の外部メモリ３０３がない画像形成装置の場合、認可サーバー２００上で利用される情報等を記憶している。

【００３６】

ＲＡＭ３０８は、ＣＰＵ３０１のワークエリアとして機能するＲＡＭであり、増設ポート（不図示）に接続されるオプションＲＡＭによりメモリ容量を拡張することができるように構成されている。また、ＲＡＭ３０８は、出力情報展開領域、環境データ格納領域、ＮＶＲＡＭ等に用いられる。外部メモリ３０３は、メモリコントローラ（ＭＣ）３０９によりアクセスを制御される。外部メモリ３０３はオプションとして接続され、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。また、操作部３１１はユーザーによる操作を受け付けるためのスイッチ、およびＬＥＤ表示器等で構成されている。なお、後述の説明においては、特に断りのない限り画像形成装置における実行のハード上の主体はＣＰＵ３０１であり、ソフトウェア上の主体は外部メモリ３０３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。

【００３７】

図３は本実施例に係る、認可サーバー２００、リソースサーバー２１０、および画像形成装置３００の夫々に備えられたモジュールの構成を示す図である。図３に示す何れのモジュールも、図２で示した各装置のＣＰＵが外部メモリにインストールされたアプリケーションプログラムをＲＡＭを利用して実行することで実現されるモジュールである。認可サーバー２００は認可サーバーモジュール６００を備え、認可サーバーモジュール６００はユーザー識別部６０１、クライアント検証部６０２、親トークン発行部６０３、子トー

クン発行部６１１、子トークン検証部６２０を備える。リソースサーバー２１０はリソースサーバーモジュール７００を備え、リソースサーバーモジュール７００は子トークン権限確認部７０１、リソース要求処理部７０２を備える。

【００３８】

また、画像処理装置３００は認可サーバー連携クライアント４００、リソースサーバー連携アプリケーション５００、Ｗｅｂブラウザ９００を備える。上述の通り、リソースサーバー連携アプリケーション５００は複数インストールされていても良い。認可サーバー連携クライアント４００、リソースサーバー連携アプリケーション５００、およびＷｅｂブラウザ９００は画像処理装置に標準で搭載されている、またはユーザーが後からインストールしたものである。Ｗｅｂブラウザ９００は一般的なＷｅｂブラウザでありサーバーから提供される画面を表示するものである。

【００３９】

認可サーバー連携アプリケーション４００は親トークン要求部４０１、親トークン記憶部４０２、子トークン要求転送部４０３を備える。また、リソースサーバー連携アプリケーション５００は子トークン要求部５０１、子トークン記憶部５０２、リソース要求部５０３を備える。上述の通り、リソースサーバーモジュール７００は複数インストールされていても良い。以上が、本発明を実施する上で必要となるシステム構成、ハードウェア構成、モジュール構成の説明になる。

【００４０】

次に、図３で示したモジュールがどのように連携し本実施例を実施するかの説明を行う。各モジュールの詳細な動きの説明を行う前に、本実施例のポイントである、認可サーバー２００が親トークンを基に子トークンを発行する一連の流れの概要を図３を用いて説明しておく。

【００４１】

始めに、ユーザーは、画像処理装置３００上のリソースサーバー連携アプリケーション５００に対し、リソースサーバーモジュール７００の機能を利用する必要がある指示を行う（１）。続けてリソースサーバー連携アプリケーション５００は、リソースサーバーモジュール７００の利用に必要な子トークンがあるか否かを確認し、子トークンがなければ認可サーバー連携クライアント４００に対し子トークンを要求する（２）。

【００４２】

認可サーバー連携クライアント４００は親トークンを用いて認可サーバーモジュール６００に対し子トークンを要求する（３）。認可サーバー連携クライアント４００の要求に応じて、認可サーバーモジュール６００は受信した親トークンを利用し子トークンを発行し、発行した子トークンを認可サーバー連携クライアント４００に返す（４）。（４）の認可サーバーモジュール６００が親トークンを利用して子トークンを発行する処理は本実施例のポイントの１つであり、その詳細な内容は後述する。

【００４３】

認可サーバー連携クライアント４００は子トークンをリソースサーバー連携アプリケーション５００に返す。（５）リソースサーバー連携アプリケーション５００は子トークンを用いてリソースサーバーモジュール７００に対してリソースサービスの利用要求を行う（６）。リソースサーバーモジュール７００はリソースサーバー連携アプリケーション５００から受け取った子トークンの検証を認可サーバーモジュール６００に依頼する（７）。検証の結果を受信し（８）、要求元であるリソースサーバー連携アプリケーション５００の権限で要求を処理できると判断した場合に、リソースサーバーモジュール７００はサービスの提供を行う。以上が、認可サーバー２００が親トークンを基に子トークンを発行する一連の流れの概要である。

【００４４】

では、図３に示した各モジュールの詳細な動きを説明して行く。始めに、認可サーバー連携クライアント４００が、認可サーバーモジュール６００から親トークンを取得する処理を説明する。本処理は、ユーザーがＷｅｂブラウザ９００を利用して認可サーバー連携

クライアント４００にアクセスすることで開始される。例えば、ユーザーが認可サーバー連携クライアント４００を画像形成装置３００にインストールした際に、Ｗｅｂブラウザ９００で認可サーバー連携クライアント４００へアクセスし親トークンを発行するよう、画像形成装置３００からユーザーへ通知すると良い。この構成であれば、認可サーバー連携クライアント４００が親トークンを必要とする際に、親トークンが親トークン記憶部４０２に保持されていない状態を回避できる。なお、認可サーバー連携クライアント４００にユーザーの権限を移譲することは、画像形成装置３００に対して権限を移譲したことと同義である。

【００４５】

図４ｃのステップＳ１２０１で親トークン要求部４０１は、ユーザーが操作するＷｅｂブラウザ９００によるアクセスを受け付ける。そして、親トークン要求部４０１は、Ｗｅｂブラウザを認可サーバーモジュール６００にリダイレクトさせる。リダイレクト先の認可サーバーのＵＲＬや認可サーバー連携クライアント４００の認証情報は、図９ａで示される認可サーバー連携情報４６０、あるいは図９ｂで示される認可サーバー連携情報４６１として管理されている。

【００４６】

また、図９ａ、および図９ｂは、親トークン要求部４０１が認可サーバーモジュール６００に要求する親トークンのスコープも表している。ここで図９ａは、リソースサーバー連携アプリケーションとして印刷アプリケーションと帳票アプリケーションが存在することを示しており、夫々のアプリケーションは印刷権限、および帳票生成権限を必要としている。この例の場合、親トークン要求部４０１は、印刷権限を移譲するためのスコープであるｐｒｉｎｔと、帳票生成権限を移譲するためのスコープであるｃｒｅａｔｅＦｏｒｍの両方を持つ親トークンを要求してもよい。こうすることで、認可サーバー連携クライアント４００は、１つの親トークンを用いて、ｐｒｉｎｔスコープを持つ子トークンと、ｃｒｅａｔｅＦｏｒｍスコープを持つ子トークンとを発行してらえる。しかし、ｐｒｉｎｔでもｃｒｅａｔｅＦｏｒｍでもない第三のスコープが必要なアプリケーションを利用するためには、第三のスコープを持つ親トークンをあらためて発行させる必要がある。このようなトークンを、アプリケーションに与える権限に制限を設けるためのトークンと呼ぶ。

【００４７】

一方で図９ｂは、親トークン要求部４０１が、任意のスコープの子トークンを発行できる親トークンを要求する場合の認可サーバー連携情報を示している。任意のスコープの子トークンを発行できる親トークンをワイルドカード親トークンと呼び、認可サーバー連携クライアント４００はワイルドカード親トークンのスコープとしてｗｉｌｄｃａｒｄスコープを要求することとしている。ｗｉｌｄｃａｒｄスコープを持つ親トークンを利用すると、第三のスコープが必要なアプリケーションが増えた場合でも、再度、認可サーバー２００に親トークンを発行してもらう必要がない。認可サーバー連携クライアント４００は、ワイルドカード親トークンを利用して、任意のスコープを持つ子トークンを発行させることができる。このように、ワイルドカード親トークンは、リソースサーバー連携アプリケーションが新たに追加される画像形成装置３００のようなデバイス機器に好適であり、ユーザーは親トークン再発行のための認可操作を行う必要がなくなり利便性が向上する。ワイルドカード親トークンのように任意のスコープを持つトークンを、アプリケーションに与える権限に制限を設けないためのトークンと呼ぶ。どちらのトークンが発行されるかは、認可サーバー連携クライアント４００に設定されたスコープのみならず、後述するユーザーに設定されたスコープにも依存する。詳細は後述する。

【００４８】

ステップＳ１２０２で親トークン要求部４０１は、親トークン発行のための認可コードを取得する。ここで、認可コード発行のための詳細なフローを説明しておく。認可サーバーモジュール６００は、ユーザーを認証するために図１２ａに示されるような認証画面８００をリダイレクトしてきたＷｅｂブラウザ９００に表示させ、ユーザーに認証情報を入力させ、入力された認証情報を基に認証を行う。そして、認可サーバーモジュール６００

は、認証されたユーザーに認可を求めるために、図１２ｂに示されるような認可画面８０１をＷｅｂブラウザ９００に表示させ、認可操作を行わせる。図１２ｂは一例であり、特に、表示される権限の内容はデータの内容に限られず、サービスの名称や、操作内容であっても良い。認証、および認可が行われると、Ｗｅｂブラウザ９００は、認可サーバーモジュール６００から親トークン要求部４０１へのリダイレクト要求、および認可操作が行われたことを示す認可コードを受信する。親トークン要求部４０１は、認可サーバーモジュール６００からリダイレクトされたＷｅｂブラウザ９００のアクセスを受け付ける。親トークン要求部４０１はこのリダイレクト時にＷｅｂブラウザ９００から認可コードを取得できる。

【００４９】

ステップＳ１２０３で親トークン要求部４０１は、ステップＳ１２０２で認可コードを得られたか否かを確認する。確認の結果として認可コードを得られたと判断された場合はステップＳ１２０４に遷移する。また認可コードを得られなかったと判断された場合はステップＳ１２５０に遷移する。ステップＳ１２０４で親トークン要求部４０１は、ステップＳ１２０２で取得した認可コードを用いて、認可サーバーモジュール６００に親トークンの発行を要求する。

【００５０】

ステップ１２０５親トークン要求部４０１は、ステップＳ１２０４の応答として得られた親トークンを親トークン記憶部４０２に記憶させる。そして、ステップＳ１１０３に遷移し、処理を継続する。ステップＳ１２５０で認可サーバー連携クライアント４００は、ユーザーの認可を得られず子トークンを取得できなかった旨をリソースサーバー連携アプリケーション５００の子トークン要求部５０１に通知し、フローを終了する。

【００５１】

次に、認可サーバーモジュール６００が親トークン発行用の認可コードを発行する処理を図５ａを用いて説明する。認可サーバーモジュール６００が、Ｓ１２０１でリダイレクトさせられたＷｅｂブラウザ９００からのアクセスを受け付けることで本フローが開始される。

【００５２】

ステップＳ２００１で認可サーバーモジュール６００は、認可サーバー連携クライアント４００からリダイレクトさせられたＷｅｂブラウザ９００を操作するユーザーのアクセスを受け付ける。ステップＳ２００２でユーザー識別部６０１は、ステップＳ２００１でリダイレクトさせられたＷｅｂブラウザ９００を操作するユーザーの識別を行う。これは、図１２ａで示した認証画面８００を介し入力されたユーザーの認証情報で識別される。識別したユーザーが正当なユーザーであればステップＳ２００３に遷移する。識別したユーザーが正当なユーザーでなければ終了する（不図示）。

【００５３】

ステップＳ２００３で認可サーバーモジュール６００は、親トークンを要求している認可サーバー連携クライアント４００の情報をユーザーのアクセスから取り出す。ステップＳ２００４でクライアント検証部６０２は、ステップＳ２００３で取り出した情報を用いて認可サーバー連携クライアント４００の検証を行う。本実施例では、図９ａ、または図９ｂに示したクライアントＩＤ、およびクライアントシークレットを用いて検証を行う。

【００５４】

ステップＳ２００５で認可サーバーモジュール６００は、ステップＳ２００４の検証結果を確認する。確認の結果、親トークンを要求しているのが正当な認可サーバー連携クライアントであると判断された場合はステップＳ２００６に遷移する。また正当でないと判断された場合はステップＳ２０５０に遷移する。ステップＳ２００６で親トークン発行部６０３は、ステップＳ２００２で受け付けたユーザーのアクセスから、認可サーバー連携クライアント４００が要求しているスコープを取り出す。

【００５５】

ステップＳ２００７で親トークン発行部６０３は、ステップＳ２００６で取り出したス

コープの親トークンを発行してよいか確認する。なお判断の詳細は後述する。ステップＳ２００８で親トークン発行部６０３は、ステップＳ２００７の確認の結果、取り出したスコープの親トークンを発行してよいと判断された場合はステップＳ２００９に遷移する。また発行できないと判断された場合はステップＳ２０５０に遷移する。ステップＳ２００９で親トークン発行部６０３は、認可サーバー連携クライアント４００に対する親トークンを発行してよいかユーザーに確認を行う。これは、図１２ｂに示すような認可画面８０１を表示し、ユーザーに認可操作を求めることで確認できる。なお、本実施例におけるユーザーの認可操作はこの１回のみとなり、アプリケーションに権限を移譲する際は親トークンを基に権限移譲の判断処理が行われる。以後、アプリケーションに権限を移譲するのはユーザーではなく、ユーザーから権限が委譲された画像形成装置３００に備えられた認可サーバー連携クライアント４００となる。

【００５６】

ステップＳ２０１０で親トークン発行部６０３は、ステップＳ２００９でユーザーの認可を得られたか確認する。確認の結果として認可を得られたと判断された場合はステップＳ２０１１に遷移する。また認可を得られなかったと判断された場合はステップＳ２０５０に遷移する。ステップＳ２０１１で親トークン発行部６０３は、親トークンを発行するための認可コードを生成する。ステップＳ２０１２で親トークン発行部６０３は、ステップＳ２００１で受け付けたＷｅｂブラウザ９００を介したユーザーからのアクセスを、アクセス元である認可サーバー連携クライアント４００に再度リダイレクトさせる。この再度のリダイレクトによるＷｅｂブラウザ９００を操作するユーザーのアクセスには、ステップＳ２０１２で生成した認可コードが含まれる。リダイレクトが完了するとフローを終了する。

【００５７】

ステップＳ２０５０で認可サーバーモジュール６００は、ステップＳ２００１で受け付けたＷｅｂブラウザ９００を介したユーザーからのアクセスを、アクセス元である認可サーバー連携クライアント４００に再度リダイレクトさせる。この再度のリダイレクトの要求とともに親トークンを発行できない旨の通知がなされる。リダイレクトが完了するとフローを終了する。

【００５８】

次に、認可コードを用いて行う親トークンを発行するか否かの判断処理を図５ｂを用いて説明する。本フローは図５ａにおけるステップＳ２００７を詳細化したものである。ステップＳ２１０１で親トークン発行部６０３は、ステップＳ２００４で検証された認可サーバー連携クライアント４００に対し、ステップＳ２００６で取り出したスコープの親トークンを発行するか否かを判断する。発行してよいと判断された場合はステップＳ２１０２に遷移し、発行してはいけないと判断された場合はステップＳ２１５０に遷移する。この判断時には、図１０ａに示すクライアント権限テーブル６５０を用いて判断が行われる。

【００５９】

クライアント権限テーブル６５０によるとｃｌｉｅｎｔＡＢＣは親トークンを発行する権限ｉｓｓｕｅＰａｒｅｎｔＴｏｋｅｎを備えている。そのため親トークンを要求した認可サーバー連携クライアント４００のクライアントＩＤがｃｌｉｅｎｔＡＢＣだった場合、親トークン発行部６０３は親トークンを発行してもよいと判断する。一方ｃｌｉｅｎｔＧＨＩは親トークンを発行する権限ｉｓｓｕｅＰａｒｅｎｔＴｏｋｅｎを備えていない。そのため親トークンを要求した認可サーバー連携クライアント４００のクライアントＩＤがｃｌｉｅｎｔＧＨＩだった場合、親トークン発行部６０３は親トークンを発行しないと判断する。

【００６０】

更に、ステップＳ２１０２で親トークン発行部６０３は、ステップＳ２００２で識別されたユーザーに対し、ステップＳ２００６で取り出したスコープの親トークンを発行してよいかを判断する。この判断時には、図１０ｂに示すユーザー権限テーブル６６０を用い

判断が行われる。例えばステップＳ２００２で識別されたユーザーがｃｌｏｕｄＵｓｅｒＸだった場合を考える。ユーザー権限テーブル６６０によると、ｃｌｏｕｄＵｓｅｒＸにはｐｒｉｎｔスコープ、およびｃｒｅａｔｅＦｏｒｍスコープを持つ親トークンが発行される。そのため要求された親トークンのスコープがｐｒｉｎｔとｃｒｅａｔｅＦｏｒｍのいずれか１つ、または両方だった場合、親トークン発行部６０３は親トークンを発行してもよいと判断する。また、例えばステップＳ２００２で識別されたユーザーがｃｌｏｕｄＵｓｅｒＹだった場合を考える。ユーザー権限テーブル６６０によると、ｃｌｏｕｄＵｓｅｒＹはワイルドカード親トークンを発行してもよいユーザーである。そのためワイルドカード親トークンの発行を要求された場合、親トークン発行部６０３はワイルドカード親トークンを発行してよいと判断する。

【００６１】

ステップＳ２１０３で親トークン発行部６０３は、ステップＳ２１０１およびステップＳ２１０２の判断の結果を受け、ステップＳ２００６で取り出したスコープの親トークンを発行してよいと最終的に判断し、フローを終了する。ステップＳ２１５０で親トークン発行部６０３は、ステップＳ２１０１およびステップＳ２１０２の判断の結果を受け、ステップＳ２００６で取り出したスコープの親トークンを発行してはいけないと最終的に判断し、フローを終了する。

【００６２】

次に、認可サーバーモジュール６００が親トークンを発行する処理を図５ｃを用いて説明する。本フローはＳ２０１２を経て認可コードを得た認可サーバー連携クライアント４００から、認可サーバーモジュール６００が親トークンの発行要求を受け付けることで開始される。ステップＳ２２０１で認可サーバーモジュール６００は、認可サーバー連携クライアント４００から親トークンの発行要求を受け付ける。ステップＳ２２０２で認可サーバーモジュール６００は、ステップＳ２２０１で受け付けた親トークンの発行要求から、親トークンを要求している認可サーバー連携クライアント４００の情報を取り出す。

【００６３】

ステップＳ２２０３でクライアント検証部６０２は、ステップＳ２２０２で取り出した情報を用いて認可サーバー連携クライアント４００の検証を行う。クライアント検証部６０２は図９ａにある、クライアントＩＤとクライアントシークレットを用いて検証を行う。ステップＳ２２０４で認可サーバーモジュール６００は、ステップＳ２２０３の検証結果を確認する。確認の結果、親トークンを要求しているのが正当な認可サーバー連携クライアントであると判断された場合はステップＳ２２０５に遷移する。また正当でないと判断された場合はステップＳ２２５０に遷移する。

【００６４】

ステップＳ２２０５で認可サーバーモジュール６００は、ステップＳ２２０１で受け付けた親トークンの発行要求から認可コードを取り出す。ステップＳ２２０６で親トークン発行部６０３は、ステップＳ２２０５で取り出された認可コードが正当なものか否かを確認する。確認の結果として正当な認可コードと判断された場合はステップＳ２２０７に遷移する。また正当でないと判断された場合はステップＳ２２５０に遷移する。

【００６５】

ステップＳ２２０７で親トークン発行部６０３は、ステップＳ２２０５で取り出された認可コードに対応する親トークンを発行し、親トークンの要求元である認可サーバー連携クライアント４００に発行した親トークンを返す。親トークンを返すとフローを終了する。なお、ここで発行した親トークンは、図１１ａで示すような、発行済み親トークンテーブル６７０で管理する。図１１ａでは、ｐａｒｅｎｔＴｏｋｅｎ１１３３５５７７とｐａｒｅｎｔＴｏｋｅｎＷＷＸＸＹＹＺＺの２つの親トークンが発行されたあとの状態を表している。またそれぞれの親トークンは、ｐｒｉｎｔスコープとｃｒｅａｔｅＦｏｒｍスコープを特定するための親トークンと、ワイルドカードスコープを特定するためのワイルドカード親トークンを示している。夫々の親トークンを保持した認可サーバー連携クライアント４００を備えた画像形成装置３００は、夫々のトークンから特定される権限が認可サ

ーバー２００から与えられたことになる。
【００６６】
例えば、認可サービスモジュール６００は、ｐａｒｅｎｔＴｏｋｅｎ１１３３５５７７を基にユーザーが持っている印刷サービスを利用するためのｐｒｉｎｔ権限、および帳票サービスを利用するためのｃｒｅａｔｅＦｏｒｍ権限が認可サーバー連携クライアント４００に移譲されたことを特定できるようになる。一方、認可サービスモジュール６００は、ｐａｒｅｎｔＴｏｋｅｎＷＷＸＸＹＹＺＺを基に、ユーザーが持っているリソースサーバー２１０が提供する全てのサービスを利用するためのｗｉｌｄｃａｒｄ権限が認可サーバー連携クライアント４００に移譲されたことを特定できるようになる。ステップＳ２２５０で親トークン発行部６０３は、親トークンの要求元である認可サーバー連携クライアント４００に親トークンを発行できない旨を通知しフローを終了する。以上が、ユーザーの権限を画像形成装置３００に移譲し、移譲された権限を特定するための親トークンを発行する詳細な説明となる。
【００６７】
次に、親トークンが発行されたあとの子トークンの発行について説明する。以下のフローは図３の（１）から（１０）の流れを詳細に説明したものとなる。図４ａは本実施例に係る画像処理装置３００上のリソースサーバー連携アプリケーション５００がユーザーの操作に応じて行う処理のフローである。本フローはユーザーがリソースサーバー連携アプリケーション５００を操作することで開始される。
【００６８】
ステップＳ１００１でリソースサーバー連携アプリケーション５００は、リソースサーバーモジュール７００が提供するサービスを受ける必要があるユーザーの操作を受け付ける。ステップＳ１００２でリソースサーバー連携アプリケーション５００は、操作を行っているユーザーに対応する子トークンを記憶しているかを子トークン記憶部５０２に確認する。子トークン記憶部５０２が記憶している取得済み子トークンテーブル５５０は図８ｂに示す通りである。ここでユーザーがｄｅｖｉｃｅＵｓｅｒＡ、またはｄｅｖｉｃｅＵｓｅｒＢであれば、それぞれ対応する子トークンｃｈｉｌｄＴｏｋｅｎ１２３４５６７８、またはｃｈｉｌｄＴｏｋｅｎＡＢＣＤＥＦＧＨが見つかる。確認の結果、操作を行っているユーザーに対応する子トークンが見つかればステップＳ１００３に遷移し、見つからなければステップＳ１０１１に遷移する。ステップＳ１００３でリソース要求部５０３は、ステップＳ１００２で見つかった子トークンを、子トークン記憶部５０２から取り出す。ステップＳ１００４でリソース要求部５０３は、ステップＳ１００３で取り出した子トークンを用いて、リソースサーバー２１０上のリソースサーバーモジュール７００にアクセスし、処理を依頼する。リソースサーバー連携アプリケーション５００のフローを終了する。
【００６９】
ステップＳ１０１１で子トークン要求部５０１は、認可サーバー連携クライアント４００に対し子トークンの発行を要求する。このとき子トークン要求部５０１は、リソースサーバー連携アプリケーション５００が必要とするスコープを認可サーバー連携クライアント４００に通知する。本実施例では、アプリケーションに与える権限はリソースサーバー連携アプリケーション５００からの通知に基づいて決定するものとしている。この際、認可サーバー連携クライアント４００は、リソースサーバー連携アプリケーション５００が通知したスコープを基に、アプリケーションに権限を与えても良いか否かを判断しても良い。この場合、認可サーバー連携クライアント４００はアプリケーション毎に移譲しても良い権限を管理する必要がある。
【００７０】
ステップＳ１０１２で子トークン要求部５０１は、ステップＳ１０１１の要求の応答として、認可サーバーモジュール６００から子トークンを得られたかを確認する。確認の結果として子トークンが得られたと判断された場合はステップＳ１０１３に遷移し、得られなかったと判断された場合はステップＳ１０５０に遷移する。ステップＳ１０１３で子ト

ークン要求部５０１は、ステップＳ１０１１の応答として得られた子トークンを子トークン記憶部５０２に記憶させる。ステップＳ１０１４でリソース要求部５０３は、ステップＳ１０１１の応答として得られた子トークンを用いてリソースサーバーモジュール７００にアクセスし、処理を依頼する。処理の依頼が完了するとリソースサーバー連携アプリケーション５００のフローを終了する。ステップＳ１０５０でリソースサーバー連携アプリケーション５００は、ステップＳ１０１１の応答として子トークンが得られず、リソースサーバーモジュール７００に処理を依頼できない旨をユーザーに通知しフローを終了する。

【００７１】

次に、認可サーバー連携クライアント４００がリソースサーバー連携アプリケーション５００の要求に応じて子トークンを返す処理を図４ｂを用いて説明する。本フローは、Ｓ１０１１においてリソースサーバー連携アプリケーション５００から子トークン要求を受け付けることで開始される。ステップＳ１１０１で認可サーバー連携クライアント４００は、リソースサーバー連携アプリケーション５００からの子トークン要求を受け付ける。

【００７２】

ステップＳ１１０２で認可サーバー連携クライアント４００は親トークン記憶部４０２に問い合わせ、リソースサーバー連携アプリケーション５００を操作しているユーザーに対応する親トークンを記憶しているか確認する。親トークン記憶部４０２が記憶している取得済み親トークンテーブル４５０は図８ａに示す通りである。ここでユーザーがｄｅｖｉｃｅＵｓｅｒＡ、またはｄｅｖｉｃｅＵｓｅｒＢであれば、それぞれに対応する親トークンｐａｒｅｎｔＴｏｋｅｎ１１３３５５７７、またはｐａｒｅｎｔＴｏｋｅｎＷＷＸＸＹＹＺＺが見つかる。確認の結果、ユーザーに対応する親トークンが見つかればステップＳ１１０３に遷移し、見つからなければステップＳ１１５０に遷移する。

【００７３】

ステップＳ１１５０で認可サーバー連携クライアント４００は、親トークンを用いて子トークンを取得できなかった旨をリソースサーバー連携アプリケーション５００の子トークン要求部５０１に通知し、フローを終了する。ステップＳ１１０３で認可サーバー連携クライアント４００は、ステップＳ１１０２で確認した親トークンを親トークン記憶部４０２から取り出す。ステップＳ１１０４で子トークン要求転送部４０３は、ステップＳ１１０３で取り出した親トークンを用いて、認可サーバーモジュール６００に子トークン発行の要求を送信する。またその際、リソースサーバー連携アプリケーション５００の子トークン要求部５０１から通知されたスコープも認可サーバーモジュール６００に送信する。アクセスする先の認可サーバーのＵＲＬや認可サーバー連携クライアント４００の認証情報は、図９ａで示される認可サーバー連携情報４６０、あるいは図９ｂで示される認可サーバー連携情報４６１として管理されている。

【００７４】

ステップＳ１１０５で子トークン要求転送部４０３は、ステップＳ１１０４の応答として、子トークンを得られたか否かを確認する。確認の結果として子トークンを得られたと判断されればステップＳ１１０６に遷移し、得られなかったと判断されればステップＳ１１５０に遷移する。ステップＳ１１０６で認可サーバー連携クライアント４００は、ステップＳ１１０６の応答として得られた子トークンをリソースサーバー連携アプリケーション５００の子トークン要求部５０１に送信しフローを終了する。このように、認可サーバー連携クライアント４００がリソースサーバー連携アプリケーション５００に代わり子トークンの発行要求を行うプロキシとして機能する。認可サーバー連携クライアント４００を備えた画像形成装置３００は、ユーザーの代理として権限移譲を行う。

【００７５】

認可サーバーモジュール６００が子トークンを発行する処理を図６ａを用いて説明する。本フローは認可サーバーモジュール６００が、認可サーバー連携クライアント４００から子トークンの発行要求を受け付けることで開始される。ステップＳ２３０１で子トークン発行要求受信部６１０は、認可サーバー連携クライアント４００から子トークンの発行

要求を受け付ける。ステップＳ２３０２で子トークン発行要求受信部６１０は、ステップＳ２３０１で受け付けた子トークンの発行要求とともに認可サーバー連携クライアント４００のアクセスから、子トークンを要求している認可サーバー連携クライアント４００の情報を取り出す。ステップＳ２３０３でクライアント検証部６０２は、ステップＳ２３０２で取り出した情報を用いて認可サーバー連携クライアント４００の検証を行う。例えば、図９ａにある、クライアントＩＤとクライアントシークレットを用いて検証を行う。

【００７６】

ステップＳ２３０４で子トークン発行要求受信部６１０は、ステップＳ２３０３の検証結果を確認する。確認の結果、子トークンを要求している要求元が正当な認可サーバー連携クライアントであると判断された場合はステップＳ２３０５に遷移する。また正当でないと判断された場合はステップＳ２３５０に遷移する。ステップＳ２３０５で子トークン発行部６１１は、ステップＳ２３０１で受け付けた子トークン発行要求とともに、認可サーバー連携クライアント４００のアクセスから親トークン、および認可サーバー連携クライアント４００が要求している子トークンのスコープを取り出す。ステップＳ２３０６で子トークン発行部６１１は、ステップＳ２３０５で取り出したスコープの子トークンを発行してよいか確認する。なお判断の詳細は後述する。

【００７７】

ステップＳ２３０７で子トークン発行部６１１は、ステップＳ２３０６の確認の結果、取り出したスコープの子トークンを発行してよいと判断された場合はステップＳ２３０８に遷移する。また発行してはいけないと判断された場合はステップＳ２３５０に遷移する。ステップＳ２３０８で子トークン送信部６１２は、ステップＳ２３０１で受け付けた子トークン発行要求の応答として子トークンを発行し、子トークンの要求元である認可サーバー連携クライアント４００に返す。子トークンを返すとフローを終了する。

【００７８】

発行した子トークンは図１１ｂで示すような、発行済み子トークンテーブル６８０で管理される。図１１ｂでは、ｃｈｉｌｄＴｏｋｅｎ１２３４５６７８とｃｈｉｌｄＴｏｋｅｎＡＢＣＤＥＦＧＨの２つの子トークンが発行されたあとの状態を表している。またそれぞれｐｒｉｎｔスコープが与えられた子トークンと、ｃｒｅａｔｅＦｏｒｍスコープが与えられた子トークンを示している。例えば、認可サービスモジュール６００は、ｃｈｉｌｄＴｏｋｅｎ１２３４５６７８を基に、認可サーバー連携クライアント４００が持っている印刷サービスを利用するためのｐｒｉｎｔ権限がリソースサーバー連携アプリケーション５００に移譲されたことを特定できるようになる。一方、認可サービスモジュール６００は、ｃｈｉｌｄＴｏｋｅｎＡＢＣＤＥＦＧＨを基に、別の認可サーバー連携クライアント４００が持っている帳票サービスを利用するためのｃｒｅａｔｅＦｏｒｍ権限が別のリソースサーバー連携アプリケーション５００に移譲されたことを特定できるようになる。ステップＳ２３５０で子トークン発行要求受信部６１０は、子トークンの要求元である認可サーバー連携クライアント４００に、子トークンを発行できない旨を通知し、フローを終了する。

【００７９】

図６ｂは本実施例に係る、子トークンの発行の判断の詳細なフローである。本フローは図６ａにおけるステップＳ２３０６を詳細化したものである。ステップＳ２４０１で子トークン発行部６１１は、ステップＳ２３０３で検証された認可サーバー連携クライアント４００に任意のスコープの子トークンを発行しても良い権限があるか否かを判断する。任意のスコープの子トークンを発行してよいと判断された場合はステップＳ２４０３に遷移する。それ以外の場合はステップＳ２４０２に遷移する。判断には、図１０ａに示すようなクライアント権限テーブル６５０を用いる。クライアント権限テーブル６５０によるとｃｌｉｅｎｔＤＥＦは任意のスコープの子トークンを発行する権限ａｎｙＳｃｏｐｅを備えている。そのため子トークンを要求した認可サーバー連携クライアント４００のクライアントＩＤがｃｌｉｅｎｔＤＥＦだった場合、子トークンを発行してよいと判断される。

【００８０】

ステップＳ２４０２で子トークン発行部６１１は、ステップＳ２３０３で検証された認可サーバー連携クライアント４００に、ステップＳ２３０５で取り出したスコープの子トークンを発行しても良い権限があるか否かを判断する。発行してよいと判断された場合はステップＳ２４０３に遷移し、発行してはいけないと判断された場合はステップＳ２４５０に遷移する。判断には、図１０ａに示すようなクライアント権限テーブル６５０を用いられる。クライアント権限テーブル６５０によるとｃｌｉｅｎｔＡＢＣはｐｒｉｎｔスコープとｃｒｅａｔｅＦｏｒｍスコープの一方、または両方のスコープを持つ子トークンを発行する権限を備えている。そのためクライアントＩＤがｃｌｉｅｎｔＡＢＣであり、かつ要求されたスコープがｐｒｉｎｔスコープとｃｒｅａｔｅＦｏｒｍスコープの一方、または両方であれば子トークンを発行してよいと判断される。また要求されたスコープが第三のスコープを含む場合、ｃｌｉｅｎｔＡＢＣ　が権限を持つｐｒｉｎｔスコープとｃｒｅａｔｅＦｏｒｍスコープのみを持つ子トークンの発行を許可する構成でもよいし、子トークンの発行を拒否する構成でもよい。

【００８１】

ステップＳ２４０３で子トークン発行部６１１は、ステップＳ２３０５で取り出した親トークンがワイルドカード親トークンか否かを確認する。確認の結果としてワイルドカード親トークンと判断された場合はステップＳ２４０５に遷移する。またワイルドカード親トークンでないと判断された場合はステップＳ２４０６に遷移する。判断には、発行済み親トークンテーブル６７０を用いる。例えば、ステップＳ２３０５で取り出した親トークンがｐａｒｅｎｔＴｏｋｅｎＷＷＸＸＹＹＺＺであればワイルドカード親トークンである。

【００８２】

ステップＳ２４０４で子トークン発行部６１１は、ステップＳ２３０５で取り出した親トークンを用いて、ステップＳ２３０５で取り出したスコープの子トークンを発行してよいか判断する。ここで子トークンを発行してよいと判断されればステップＳ２４０５に遷移し、発行してはいけないと判断された場合はステップＳ２４５０に遷移する。例えば、ステップＳ２３０５で取り出した親トークンがｐａｒｅｎｔＴｏｋｅｎ１１３３５５７７であれば、これはｐｒｉｎｔスコープとｃｒｅａｔｅＦｏｒｍスコープを持つ親トークンである。そしてステップＳ２３０５で取り出した、要求されたスコープがｐｒｉｎｔスコープであれば子トークンを発行してよいと判断される。ステップＳ２４０３、およびステップＳ２４０４の判断処理により、認可サーバー２００は、親トークンを基に特定された権限内の少なくとも一部の権限がリソースサーバー連携アプリケーション５００に与えることになる。そして、認可サーバー２００は与えた権限を特定するための子トークンを発行することになる。換言すれば、認可サーバー２００がリソースサーバー連携アプリケーション５００に与える最大の権限は、親トークから特定される権限内の権限であって、リソースサーバー連携アプリケーション５００に必要な権限に限られることになる。

【００８３】

ステップＳ２４０５で子トークン発行部６１１は、親トークンに紐付けられたユーザーがステップＳ２３０５で取り出したスコープに対応する権限を有しているか否かを判断する。親トークンが発行されているにも関わらず、再度、ユーザーが有する権限を確認する理由は、ワイルドカード親トークンに対応するためである。ワイルドカード親トークンは要求された任意の権限をアプリケーションに対して与えることができる、権限を与えることに制限が無いトークンである。よって、子トークンを要求された時点でユーザーに権限が無いにもかかわらず子トークンが発行される危険がある。そのため、ステップＳ２４０５でユーザーの権限の確認を行っている。

【００８４】

子トークンを発行してよいと判断された場合はステップＳ２４０６に遷移し、発行してはいけないと判断された場合はステップＳ２４５０に遷移する。判断が行われる場合、子トークン発行部６１１は、親トークンに紐付けられたユーザーを発行済み親トークンテーブル６７０を用いて確認し、さらにそのユーザーの権限をユーザー権限テーブル６６０で

判断する。例えば、ステップＳ２３０５で取り出した親トークンがｐａｒｅｎｔＴｏｋｅｎ１１３３５５７７であれば、このトークンに紐付けられたユーザーはｃｌｏｕｄＵｓｅｒＸである。ユーザー権限テーブル６６０によれば、ｃｌｏｕｄＵｓｅｒＸ はｐｒｉｎｔスコープの権限とｃｒｅａｔｅＦｏｒｍスコープの権限有していることが確認できる。そのため、ステップＳ２３０５で取り出したスコープがｐｒｉｎｔであれば、子トークンを発行してよいと判断される。

【００８５】

ステップＳ２４０６で子トークン発行部６１１は、ステップＳ２４０１からステップＳ２４０５の判断の結果をあわせ、ステップＳ２３０５で取り出したスコープの子トークンを発行してよいと判断してフローを終了する。ステップＳ２４５０で子トークン発行部６１１は、ステップＳ２３０５で取り出したスコープの子トークンを発行してはいけないと判断してフローを終了する。以上が、認可サーバー２００が、親トークンを基に子トークンを発行する処理の詳細な説明となる。上述した通り、認可サーバー２００は親トークンの権限のみならず、認可サーバー連携クライアント４００が有する権限、更に、ユーザーが有する権限の確認も行う。最後に、リソースサーバー連携アプリケーション５００が子トークンを用いてリソースサービスを利用する処理について説明する。

【００８６】

図７ａは本実施例に係る、リソースサーバーモジュール７００が、子トークンを用いたリソースアクセスを制御する際のフローである。本フローはリソースサーバーモジュール７００が、リソースサーバー連携アプリケーション５００からアクセスを受けることで開始される。ステップＳ２５０１でリソースサーバーモジュール７００は、リソースサーバー連携アプリケーション５００からのアクセスを受ける。ステップＳ２５０２でリソースサーバーモジュール７００は、リソースサーバー連携アプリケーション５００によるアクセスから子トークンを取り出す。ステップＳ２５０３で子トークン権限確認部７０１は、ステップＳ２５０２で取り出した子トークンを認可サーバーモジュール６００に渡し、子トークンから特定される権限を確認する。

【００８７】

ステップＳ２５０４で子トークン権限確認部７０１は、ステップＳ２５０３で認可サーバー２００が子トークンを基に特定した権限に関する情報を受信し、受信された情報から確認した権限で、ステップＳ２５０１でアクセスを受け付けたリソースへのアクセスを許可してよいか否かを判断する。許可してよいと判断された場合はステップＳ２５０５へ遷移し、拒否すると判断された場合はステップＳ２５５０に遷移する。例えば、ステップＳ２５０３で確認された権限がｐｒｉｎｔであり、ステップＳ２５０１でアクセスを受け付けたリソースへのアクセスに必要な権限がｐｒｉｎｔだった場合、アクセスを許可してよいと判断される。

【００８８】

ステップＳ２５０５でリソース要求処理部７０２は、ステップＳ２５０１で受け付けたリソースへのアクセスを許可し、サービスの提供を行う。ステップＳ２５５０でリソースサーバーモジュール７００は、ステップＳ２５０１で受け付けたリソースへのアクセスを拒否し、フローを終了する。

【００８９】

図７ｂは本実施例に係る、認可サーバーモジュール６００における子トークンの検証フローである。本フローは認可サーバーモジュール６００が子トークンを受け取ることで開始される。ステップＳ２６０１で子トークン検証部６２０は、リソースサーバーモジュール７００から子トークンを受け取る。

【００９０】

ステップＳ２６０２で子トークン検証部６２０は、ステップＳ２６０１で受け取った子トークンで移譲された権限を確認し、リソースサーバーモジュール７００に返し、フローを終了する。例えば、図１１ｂによれば、受け取った子トークンがｃｈｉｌｄＴｏｋｅｎ１２３４５６７８だった場合、移譲された権限としてｐｒｉｎｔが特定した権限に関する

情報として子トークン権限確認部７０１に送信される。以上が、リソースサーバー連携アプリケーション５００が子トークンを用いてリソースサービスを利用する処理の説明となる。

【００９１】

実施例１によればユーザー、デバイス機器、およびデバイス機器上のアプリケーションの３者間で権限を移譲させていく際に、ユーザーの認可操作の回数を低減させつつ、デバイス機器上のアプリケーションに過剰な権限が移譲されないようにすることが可能になる。

【００９２】

＜その他の実施例＞

本願発明の実施例１では、親トークンを発行する必要があるという前提で説明したが、予め認可サーバー連携クライアント４００がユーザー毎に親トークンを管理している状態でも良い。この場合、画像形成装置３００に認可サーバー連携クライアント４００がインストールされた段階で親トークンが使用でき、画像形成装置３００を利用するユーザーが増加することには対処し難くなるが、子トークンを発行する処理に対応は可能である。

【００９３】

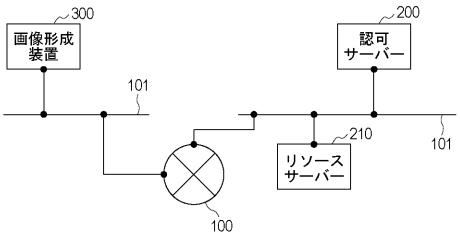
本願発明の実施例１では、リソースサービスとして帳票サービス、および印刷サービスと言った画像処理サービスを例に説明したが、これに限らずその他のサービス、例えばゲームアプリケーション、または音楽のコンテンツ配信サービスであっても良い。また、端末であるデバイス機器として画像形成装置を例に説明したが、これに限らずその他のデバイス機器、例えば、スマートフォン、または音楽機器であっても良い。また、リソースサービス連携アプリケーションとして帳票アプリケーション、印刷アプリケーションを説明したが、これに限らずその他のアプリケーション、例えば、アプリケーション管理ソフト、または音楽アプリケーションであっても良い。このように、本願発明を実施する各主体に制限はない。更に、リソースサービスは複数あることを前提に説明したが、単体であっても良い。

【符号の説明】

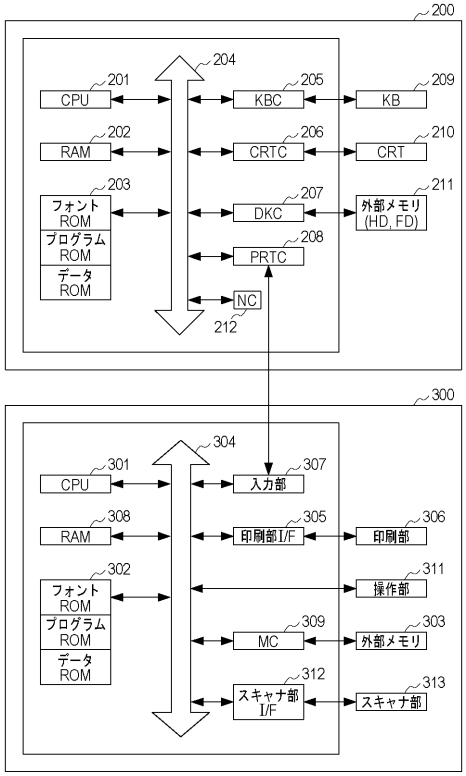
【００９４】

２００　認可サーバー
２１０　リソースサーバー
３００　画像形成装置
４００　認可サーバー連携クライアント
５００　リソースサーバー連携アプリケーション
６００　認可サーバーモジュール
６１０　子トークン発行要求受信部
６１１　子トークン発行部
６１２　子トークン送信部
７００　リソースサーバーモジュール

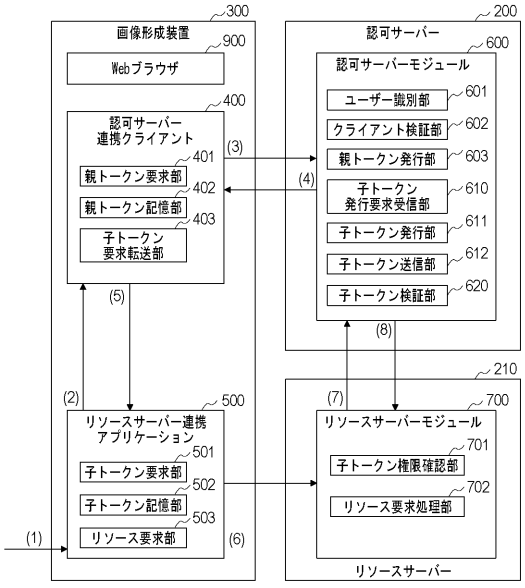
【図１】


300
画像形成装置
101
200
認可サーバー
210
リソースサーバー
101
100

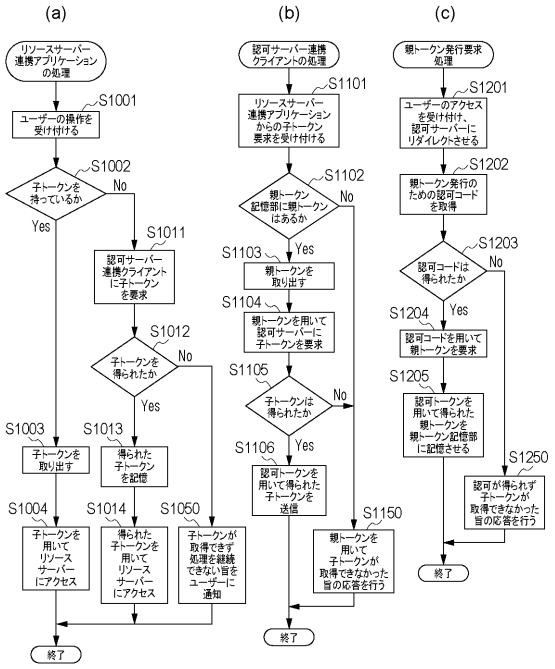
【図２】


200
204
CPU
201
KBC
205
KB
209
RAM
202
CRTC
206
CRT
210
フォントROM
203
DKC
207
外部メモリ(HD, FD)
211
プログラムROM
PRTC
208
データROM
NC
212
300
304
CPU
301
入力部
307
RAM
308
印刷部I/F
305
印刷部
306
フォントROM
302
操作部
311
プログラムROM
MC
309
外部メモリ
303
データROM
スキャナ部I/F
312
スキャナ部
313

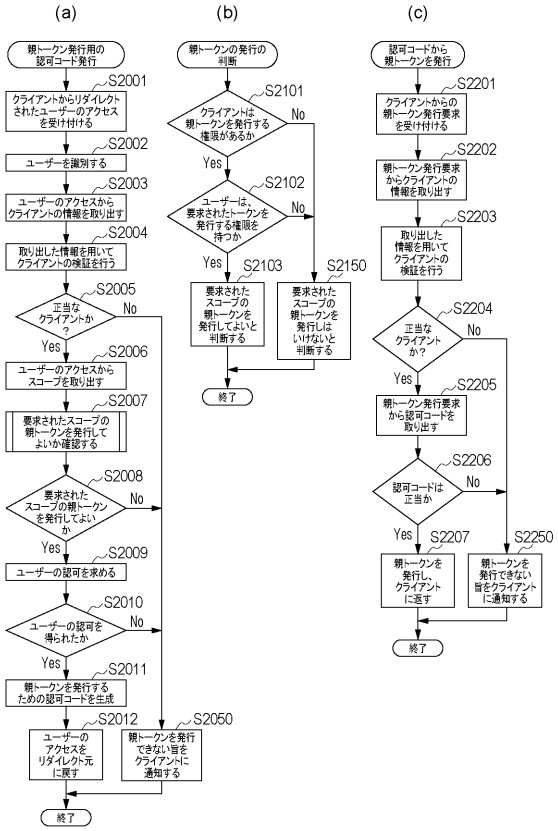
【図３】


300
画像形成装置
900
Webブラウザ
400
認可サーバー連携クライアント
401
親トークン要求部
402
親トークン記憶部
403
子トークン要求転送部
500
リソースサーバー連携アプリケーション
501
子トークン要求部
502
子トークン記憶部
503
リソース要求部
200
認可サーバー
600
認可サーバーモジュール
601
ユーザー識別部
602
クライアント検証部
603
親トークン発行部
610
子トークン発行要求受信部
611
子トークン発行部
612
子トークン送信部
620
子トークン検証部
210
700
リソースサーバーモジュール
701
子トークン権限確認部
702
リソース要求処理部
リソースサーバー
(1)
(2)
(3)
(4)
(5)
(6)
(7)
(8)

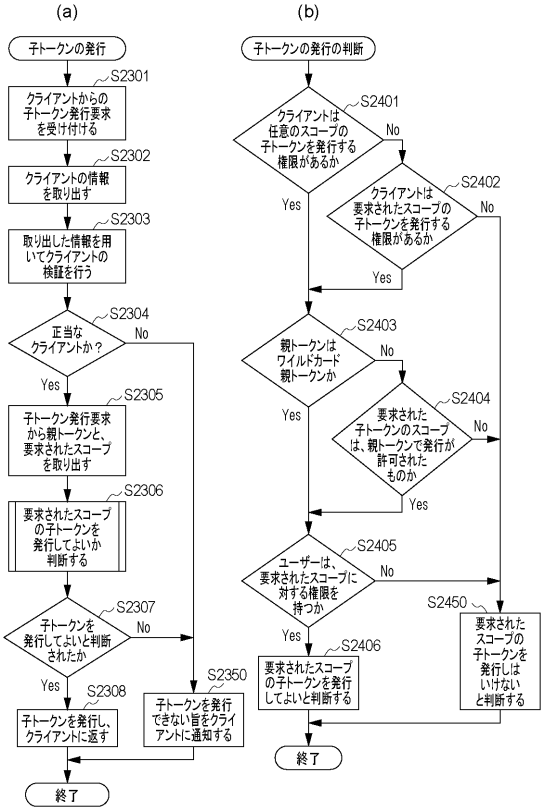
【図４】


(a)
(b)
(c)
リソースサーバー連携アプリケーションの処理
S1001
ユーザーの操作を受け付ける
S1002
子トークンを待っているか
No
Yes
S1011
認可サーバー連携クライアントに子トークンを要求
S1012
子トークンを得られたか
No
Yes
S1003
子トークンを取り出す
S1013
得られた子トークンを記憶
S1004
子トークンを用いてリソースサーバーにアクセス
S1014
得られた子トークンを用いてリソースサーバーにアクセス
S1050
子トークンが取得できず処理を継続できない旨をユーザーに通知
終了
認可サーバー連携クライアントの処理
S1101
リソースサーバー連携アプリケーションからの子トークン要求を受け付ける
S1102
親トークン記憶部に親トークンはあるか
No
Yes
S1103
親トークンを取り出す
S1104
親トークンを用いて認可サーバーに子トークンを要求
S1105
子トークンは得られたか
No
Yes
S1106
認可トークンを用いて得られた子トークンを送信
S1150
親トークンを用いて子トークンが取得できなかった旨の応答を行う
終了
親トークン発行要求処理
S1201
ユーザーのアクセスを受け付け、認可サーバーにリダイレクトさせる
S1202
親トークン発行のための認可コードを取得
S1203
認可コードは得られたか
No
Yes
S1204
認可コードを用いて親トークンを要求
S1205
認可トークンを用いて得られた親トークンを親トークン記憶部に記憶させる
S1250
認可が得られず子トークンが取得できなかった旨の応答を行う
終了

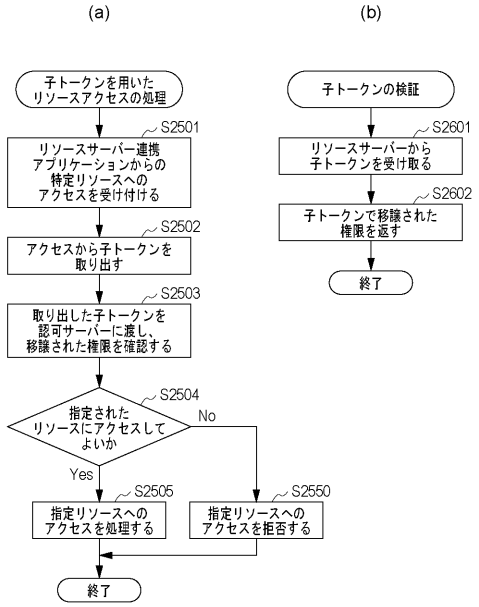
【図５】



【図６】



【図７】



【図８】

(a)

450

ユーザーID	親トークン
deviceUserA	parentToken11335577
deviceUserB	parentTokenWWXXYYZZ
:	:

(b)

550

ユーザーID	子トークン
deviceUserA	childToken12345678
deviceUserB	childTokenABCDEFGH
:	:

【図９】

(a)

460

URL	http://AuthorizationServer/endpoint/
Client ID	clientABC
Client Secret	secretXXX
Scope	print, createForm

(b)

461

URL	http://AuthorizationServer/endpoint/
Client ID	clientABC
Client Secret	secretXXX
Scope	wildcard

【図１０】

(a)

650

Client ID	権限
clientABC	issueParentToken, print, createForm
clientDEF	issueParentToken, anyScope
clientGHI	anyScope

(b)

660

User ID	権限
cloudUserX	print, createForm
cloudUserY	wildcard

【図１１】

(a)

670

親トークン	Scope	UserID
parentToken11335577	print, createForm	cloudUserX
parentTokenWWXXYYZZ	wildcard	cloudUserY
:	:	:

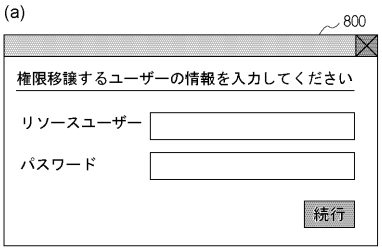
(b)

680

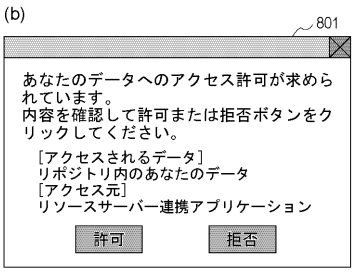
子トークン	Scope	UserID
childToken12345678	print	cloudUserX
childTokenABCDEFGH	createForm	cloudUserY
:	:	:

【図１２】

(a)



(b)



フロントページの続き

(56)参考文献 米国特許第８５３３７９６（ＵＳ，Ｂ１）
特開２０１２－０９３８０１（ＪＰ，Ａ）
特開２００５－３４１０９０（ＪＰ，Ａ）
特開２００６－２２１５０６（ＪＰ，Ａ）
後藤 浩行、他，ＯＡｕｔｈにおけるトークンを用いた権限管理方式の提案と実装，情報処理学会第７３回（平成２３年）全国大会講演論文集（３），日本，一般社団法人情報処理学会，２０１１年 3月 2日，3Ｙ－8，ｐ．3－５１７～3－５１８
岩片 靖、他，Ｌｉｎｕｘ認証のすべて 統合認証からシングルサインオンまで 第3回，日経Ｌｉｎｕｘ，日本，日経ＢＰ社，２０１１年１０月 8日，第１３巻、第１１号，ｐ．１５３－１５８

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ２１／３３
Ｇ０６Ｆ ２１／４５