(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第5787640号
(P5787640)
(45)発行日　平成27年9月30日(2015.9.30)　(24)登録日　平成27年8月7日(2015.8.7)

(51)Int.Cl.　F I
*G06F 21/62 (2013.01)*　G06F 21/62

請求項の数 5　(全 23 頁)

(21)出願番号　特願2011-140881(P2011-140881)
(22)出願日　平成23年6月24日(2011.6.24)
(65)公開番号　特開2013-8229(P2013-8229A)
(43)公開日　平成25年1月10日(2013.1.10)
審査請求日　平成26年6月23日(2014.6.23)

(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　松ヶ下　勇人
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】認証システムおよび認証方法およびプログラム

(57)【特許請求の範囲】
【請求項１】
ＵＲＬに対応する画面を提供するか否かをロール情報によって管理し、ＡＰＩの実行権限をロール情報によって管理し、データを配信するか否かをロール情報によって管理する管理手段と、
リソースに対するアクセス可否確認と認証トークンとを受信する受信手段と、
前記受信手段により受信された前記認証トークンに関連付けられたロール情報を決定する決定手段と、
前記受信手段により受信されたアクセス可否確認に対応するリソース種別がＵＲＬリソースの場合に、前記決定手段により決定された前記ロール情報と前記ＵＲＬリソースのロール情報とに基づいてアクセスを許可するか否か前記管理手段の管理内容に基づいて検証するＵＲＬ検証手段と、
前記ＵＲＬ検証手段によりアクセスを許可すると判定された場合、前記ＵＲＬリソースに対応する画面を提供する提供手段と、
前記受信手段により受信されたアクセス可否確認に対応するリソース種別がＡＰＩを介してのウェブサービスの実行である場合に、前記決定手段により決定された前記ロール情報と前記ＡＰＩの実行権限のロール情報とに基づいてアクセスを許可するか否かを前記管理手段の管理内容に基づいて検証するＡＰＩ検証手段と、
前記ＡＰＩ検証手段によりアクセスを許可すると判定された場合、前記ＡＰＩを実行する実行手段と、

前記受信手段により受信されたアクセス可否確認に対応するリソース種別がデータの配信である場合に、前記決定手段により決定された前記ロール情報と前記データの配信のロール情報とに基づいてアクセスを許可するか否かを前記管理手段の管理内容に基づいて検証するデータ配信検証手段と、
前記データ配信検証手段によりアクセスを許可すると判定された場合、前記データを配信する配信手段を備えることを特徴とする認証システム。
【請求項２】
ユーザーのユーザー識別情報を用いて前記ユーザーが所属するテナントと前記ユーザーのロール情報を特定できるユーザー情報を記憶する第１記憶手段と、
前記受信手段により前記ユーザーから認証要求としてユーザー識別情報が受信された場合、受信されたユーザー識別情報と前記第１記憶手段に記憶された前記ユーザー情報とに基づいて前記ユーザーが正当なユーザーであるか否かを判定する判定手段と、
前記判定手段により前記ユーザーが正当なユーザーであると判定された場合、前記ユーザーの認証トークンを生成する生成手段と、
前記生成手段により生成された前記ユーザーの認証トークンと、前記ユーザー情報に基づいて特定される前記ユーザーのロール情報とが関連づけられた認証トークン情報を記憶する第２記憶手段を更に備え、
前記決定手段は、前記認証トークン情報を用いて前記認証トークンに関連付けられたロール情報を決定することを特徴とする請求項１に記載の認証システム。
【請求項３】
データリソースからロール情報を特定できるリソース情報を記憶する前記第２記憶手段を更に有し、
前記ＵＲＬ検証手段、ＡＰＩ検証手段およびデータ配信検証手段は、前記認証トークン情報と前記リソース情報を用いて検証することを特徴とする請求項２に記載の認証システム。
【請求項４】
ＵＲＬに対応する画面を提供するか否かをロール情報によって管理し、ＡＰＩの実行権限をロール情報によって管理し、データを配信するか否かをロール情報によって管理する管理工程と、
リソースに対するアクセス可否確認と認証トークンとを受信する受信工程と、
前記受信工程により受信された前記認証トークンに関連付けられたロール情報を決定する決定工程と、
前記受信工程により受信されたアクセス可否確認に対応するリソース種別がＵＲＬリソースの場合に、前記決定工程により決定された前記ロール情報と前記ＵＲＬリソースのロール情報とに基づいてアクセスを許可するか否か前記管理工程の管理内容に基づいて検証するＵＲＬ検証工程と、
前記ＵＲＬ検証工程によりアクセスを許可すると判定された場合、前記ＵＲＬリソースに対応する画面を提供する提供工程と、
前記受信工程により受信されたアクセス可否確認に対応するリソース種別がＡＰＩを介してのウェブサービスの実行である場合に、前記決定工程により決定された前記ロール情報と前記ＡＰＩの実行権限のロール情報とに基づいてアクセスを許可するか否かを前記管理工程の管理内容に基づいて検証するＡＰＩ検証工程と、
前記ＡＰＩ検証工程によりアクセスを許可すると判定された場合、前記ＡＰＩを実行する実行工程と、
前記受信工程により受信されたアクセス可否確認に対応するリソース種別がデータの配信である場合に、前記決定工程により決定された前記ロール情報と前記データの配信のロール情報とに基づいてアクセスを許可するか否かを前記管理工程の管理内容に基づいて検証するデータ配信検証工程と、
前記データ配信検証工程によりアクセスを許可すると判定された場合、前記データを配信する配信工程を備えることを特徴とする認証方法。

【請求項５】
ＵＲＬに対応する画面を提供するか否かをロール情報によって管理し、ＡＰＩの実行権限をロール情報によって管理し、データを配信するか否かをロール情報によって管理する管理工程と、
リソースに対するアクセス可否確認と認証トークンとを受信する受信工程と、
前記受信工程により受信された前記認証トークンに関連付けられたロール情報を決定する決定工程と、
前記受信工程により受信されたアクセス可否確認に対応するリソース種別がＵＲＬリソースの場合に、前記決定工程により決定された前記ロール情報と前記ＵＲＬリソースのロール情報とに基づいてアクセスを許可するか否か前記管理工程の管理内容に基づいて検証するＵＲＬ検証工程と、
前記ＵＲＬ検証工程によりアクセスを許可すると判定された場合、前記ＵＲＬリソースに対応する画面を提供する提供工程と、
前記受信工程により受信されたアクセス可否確認に対応するリソース種別がＡＰＩを介してのウェブサービスの実行である場合に、前記決定工程により決定された前記ロール情報と前記ＡＰＩの実行権限のロール情報とに基づいてアクセスを許可するか否かを前記管理工程の管理内容に基づいて検証するＡＰＩ検証工程と、
前記ＡＰＩ検証工程によりアクセスを許可すると判定された場合、前記ＡＰＩを実行する実行工程と、
前記受信工程により受信されたアクセス可否確認に対応するリソース種別がデータの配信である場合に、前記決定工程により決定された前記ロール情報と前記データの配信のロール情報とに基づいてアクセスを許可するか否かを前記管理工程の管理内容に基づいて検証するデータ配信検証工程と、
前記データ配信検証工程によりアクセスを許可すると判定された場合、前記データを配信する配信工程をコンピュータに実行させるためのプログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、マルチテナントサービスにおける複数のリソースに対するアクセス制御方法に関する。
【背景技術】
【０００２】
従来、Ｗｅｂアプリケーションは、サービス提供先の企業や組織ごとに専用のサーバーを用意して提供する形態が主であった。しかしながら、提供先ごとに専用のサーバーを用意する形態はコスト効率が低下する。そのため、近年、共有のサーバー上に展開した同一のＷｅｂアプリケーションを複数の企業や組織に提供する“マルチテナントサービス”という形態が注目されている。ここで“テナント”とは、従来の専用サーバーを用いてサービスを提供していた企業や組織の単位を意味する。
【０００３】
テナントごとに専用のサーバーを用いる方法と比較して、マルチテナントサービスはコスト面で優れているがセキュリティ面で課題がある。従来の形態では、テナントが所有するデータがテナントごとの専用のサーバーで管理されており、物理的に分離されているためデータ漏洩のリスクは低い。しかしながら、マルチテナントサービスは複数テナントのデータを共有のサーバーで管理するため、物理的に分離されておらず、データ漏洩のリスクが高くなる。そこで、マルチテナントサービスでは、テナント間でのデータ漏洩を防ぐために、論理的にデータを分離する仕組みが必須である。
【０００４】
例えば、先行技術では、データを論理的に分離するためのキーとしてテナントＩＤを用いた方法が提案されている。このテナントＩＤを、ユーザーを識別するための属性であるユーザーＩＤと紐づけ、テナントが所有するデータにも同様にテナントＩＤを付与する方

法によってマルチテナントサービスを実現している。より詳細には、ユーザー認証によってユーザーＩＤとともにテナントＩＤを特定し、データアクセス時に、同一のテナントＩＤが付与されたデータのみアクセスを許可するアクセス制御方法である。

【先行技術文献】

【特許文献】

【０００５】

【特許文献１】特開２０１０－２６６５３号公報

【発明の概要】

【発明が解決しようとする課題】

【０００６】

マルチテナントサービスの場合、従来のテナントごとに専用のサーバーを用いる形態と比較してコスト削減を図る事が可能となる。しかしながら、例えば、先行技術で提案されているマルチテナントサービスにおけるアクセス制御方法は、その目的に特化して設計、開発されている。そのため、マルチテナントサービスの課題を解決するためのデータアクセス制御方法に対して、専用の設計、開発、および運用、保守のコストがかかってしまうという課題があった。

【０００７】

一方、従来のＷｅｂアプリケーションのアクセス制御方法としてロールを用いた方法が知られている。たとえば、WWW（Ｗｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ）に展開する有償のＷｅｂアプリケーションは、ユーザー認証の機構を備え、有償利用の契約をしたユーザーのみアクセスできるよう制御する方法を備えるのが通常である。その方法として、WWW上のＵＲＬ（Ｕｎｉｆｏｒｍ　Ｒｅｓｏｕｒｃｅ　Ｌｏｃａｔｏｒ）へのアクセス権に対応してロールを定義し、ユーザーがロールを保持しているかでアクセス制御する方法が知られている。

【０００８】

また、従来のアクセス制御方法として、ユーザーの権限によって実行可能な機能を制御する方法が知られている。たとえば、ユーザー情報の取得、作成、削除は管理者権限が必要であり、管理者である事を示すロールをユーザーが保持しているかによって、機能の実行を許可するか、拒否するかを制御する方法である。

【０００９】

本発明は、マルチテナントサービスにおいて、従来のアクセス制御方法と統一的な方法を用いて課題を解決する事により、出来る限り専用のコストをかけずに実現する事を目的としている。

【課題を解決するための手段】

【００１０】

上記課題を解決するために本願の認証システムは、ＵＲＬに対応する画面を提供するか否かをロール情報によって管理し、ＡＰＩの実行権限をロール情報によって管理し、データを配信するか否かをロール情報によって管理する管理手段と、リソースに対するアクセス可否確認と認証トークンとを受信する受信手段と、前記受信手段により受信された前記認証トークンに関連付けられたロール情報を決定する決定手段と、前記受信手段により受信されたアクセス可否確認に対応するリソース種別がＵＲＬリソースの場合に、前記決定手段により決定された前記ロール情報と前記ＵＲＬリソースのロール情報とに基づいてアクセスを許可するか否か前記管理手段の管理内容に基づいて検証するＵＲＬ検証手段と、前記ＵＲＬ検証手段によりアクセスを許可すると判定された場合、前記ＵＲＬリソースに対応する画面を提供する提供手段と、前記受信手段により受信されたアクセス可否確認に対応するリソース種別がＡＰＩを介してのウェブサービスの実行である場合に、前記決定手段により決定された前記ロール情報と前記ＡＰＩの実行権限のロール情報とに基づいてアクセスを許可するか否かを前記管理手段の管理内容に基づいて検証するＡＰＩ検証手段と、前記ＡＰＩ検証手段によりアクセスを許可すると判定された場合、前記ＡＰＩを実行する実行手段と、前記受信手段により受信されたアクセス可否確認に対応するリソース種

別がデータの配信である場合に、前記決定手段により決定された前記ロール情報と前記データの配信のロール情報とに基づいてアクセスを許可するか否かを前記管理手段の管理内容に基づいて検証するデータ配信検証手段と、前記データ配信検証手段によりアクセスを許可すると判定された場合、前記データを配信する配信手段を備えることを特徴とする。

【発明の効果】

【００１１】

本発明により、コストをかけることなくサービスを実現する事ができる。

【図面の簡単な説明】

【００１２】

【図１】システム構成図。

【図２】各装置のハードウェア構成図。

【図３】ログインサービスのソフトウェアモジュールの説明図。

【図４】アクセス制御サービスのソフトウェアモジュールの説明図。

【図５】サービスのソフトウェアモジュールの説明図。

【図６】ユーザー情報のデータ構造。

【図７】リソース情報のデータ構造。

【図８】ＡＰＩ権限情報のデータ構造。

【図９】一般的なＷｅｂアプリケーションのアクセスシーケンス。

【図１０】アクセス制御のフローチャート。

【図１１】ユーザー情報のデータ例。

【図１２】リソース情報のデータ例。

【図１３】ＡＰＩ権限情報のデータ例。

【図１４】一例としてのＷｅｂアプリケーションの画面フロー図。

【図１５】一例としてのＷｅｂアプリケーションのアクセスシーケンス。

【図１６】ロール制御情報のデータ例。

【発明を実施するための形態】

【００１３】

以下、本発明を実施するための最良の形態について図面を用いて説明する。

【００１４】

図１は、本発明の実施例のシステム構成を示すブロック図である。

【００１５】

図中１０は、Ｗｉｄｅ　Ａｒｅａ　Ｎｅｔｗｏｒｋ（ＷＡＮ１０）であり、本発明ではＷｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ（WWW）システムが構築されている。図中１１は各構成要素を接続するＬｏｃａｌ　Ａｒｅａ　Ｎｅｔｗｏｒｋ（ＬＡＮ１１）である。

【００１６】

図中１２はＷＡＮ１０を介して各サービスに対してＷｅｂリクエストを発行する複数のクライアント１２Ａ、１２Ｂであり、より具体的にはWWWシステムを利用するためのＷｅｂブラウザを備えたコンピュータである。なお、クライアント１２Ａおよび、クライアント１２Ｂは、不図示のファイアウォール装置により、ＷＡＮ１０に対するリクエスト以外の通信が遮断されている。

【００１７】

図中１３はＷＡＮ１０およびＬＡＮ１１を介して、クライアント１２からのＷｅｂリクエスト要求に応じて、ユーザーを認証するためのログイン画面を提示し、ユーザーからのログインリクエストを受け付けるログインサービス１３である。

【００１８】

図中１４はＬＡＮ１１を介して、ログインサービス１３および、一つないし複数のサービス１５からのアクセス許可リクエストを受け付けるアクセス制御サービス１４である。

【００１９】

図中１５はＷＡＮ１０およびＬＡＮ１１を介して、クライアント１２からのＷｅｂリクエスト要求に応じて、各種サービスを提供する一つないし複数のサービス１５Ａ、サービ

ス１５Ｂである。
【００２０】
図中１６はＬＡＮ１１を介してアクセス制御サービス１４からのデータアクセス要求をうけつけるデータベースサービス１６である。データベースサービス１６は、一般的なＤＢＭＳ（ＤａｔａＢａｓｅ　Ｍａｎａｇｅｍｅｎｔ　Ｓｙｓｔｅｍ）が構成されており、アクセス制御サービス１４からのデータアクセスクエリをうけつけ、該当するデータを応答する。
【００２１】
図２は、図１中のクライアント１２、ログインサービス１３、アクセス制御サービス１４、サービス１５、あるいはデータベースサービス１６のハードウェア構成を示すブロック図である。図中、２１は内部バスで接続される各デバイス（後述のＲＯＭ、ＲＡＭ他）を直接或いは間接的に制御し、本発明を実現するためのプログラムを実行するＣＰＵである。２２はＢＩＯＳが格納してあるＲＯＭである。２３はＣＰＵ２１のワーク領域として利用されたり、本発明を実現するためのソフトウェアモジュールをロードするための一時記憶として利用されたりするＲＡＭ（直接記憶装置）。２４は基本ソフトウェアであるＯＳやソフトウェアモジュールが記憶されているＨＤＤ（ハードディスクドライブ）、もしくはＳＳＤ（ソリッドステートドライブ）などの間接記憶装置。２５は入力装置であり不図示のキーボードやポインティングデバイスなどである。２６は出力装置でありディスプレイが接続される。２７はＷＡＮ１０ないしはＬＡＮ１１に接続するためのＩ／Ｆ（Ｉｎｔｅｒｆａｃｅ）であり、一つないしは複数備えている。
【００２２】
これらハードウェアでは、起動後ＣＰＵ２１によりＢＩＯＳが実行されＯＳがＨＤＤ２４からＲＡＭ２３に実行可能にロードされる。ＣＰＵ２１はＯＳの動作に従って後述する各種ソフトウェアモジュールをＨＤＤ２４からＲＡＭ２３に随時、実行可能にロードする。各種ソフトウェアモジュールは上記各デバイスの協調によりＣＰＵ２１によって実行され動作する。また、Ｉ／Ｆ２７はＬＡＮ１１に接続されており、ＯＳの動作に従ってＣＰＵ２１により制御され、各サーバーに格納されたサービス間のリクエストの送受信を実現している。また、Ｉ／Ｆ２７はＬＡＮ１１を経由してＷＡＮ１０に接続されており、ＯＳの動作に従ってＣＰＵ２１により制御され、ＷＷＷシステムにおける通信を実現している。
【００２３】
また、図中１のログインサービス１３、アクセス制御サービス１４、サービス１５、あるいはデータベースサービス１６は、一台、ないしは複数台の図２で示されたハードウェア構成のサーバーによって構成される。複数台のサーバーで構成する場合は、不図示のロードバランサー装置や、不図示のソフトウェアモジュールによって負荷分散構成、もしくは冗長化構成を採用する事ができる。
【００２４】
図３は、ログインサービス１３上で動作するソフトウェアモジュールの構成図である。なお各ソフトウェアモジュールは図２で示したＨＤＤ２４に記憶されており、前述したようにＣＰＵ２１によってＲＡＭ２３にロードされ実行される。
【００２５】
Ｗｅｂサーバー３１は、クライアント１２からのＷｅｂリクエストを受け付けるＷｅｂインタフェースを備えたＷｅｂアプリケーションサーバーである。
【００２６】
ログインアプリケーション３２は、Ｗｅｂサーバー３１上にアプリケーションとして構成され、Ｗｅｂサーバー３１が受け付けたＷｅｂリクエストに対して、ログイン画面を生成する。
【００２７】
アクセス制御エージェント３３は、Ｗｅｂサーバー３１上にフィルタリングアプリケーションとして構成され、ログインアプリケーション３２に対するＷｅｂリクエストをフィ

ルタリングする。そして、アクセス制御サービス１４に構成されるアクセス制御エージェントＩ／Ｆ４１と通信することによりユーザーの認証処理を実行する。

【００２８】

以後、上記ソフトウェアモジュールの協調により実行される一連の認証処理をログインサービス１３で実行される処理と記載する。なお、ログインサービス１３で実行されるユーザーの認証処理の詳細については後述する。

【００２９】

図４は、アクセス制御サービス１４上で動作するソフトウェアモジュールの構成図である。なお各ソフトウェアモジュールは図２で示したＨＤＤ２４に記憶されており、前述したようにＣＰＵ２１によってＲＡＭ２３にロードされ実行される。

【００３０】

図中４１はログインサービス１３、サービス１５に構成されるアクセス制御エージェント３３および５３からのリクエスト受付およびレスポンス応答を行うアクセス制御エージェントＩ／Ｆ４１である。

【００３１】

図中４２はサービス１５に構成されるアクセス制御サービスＩ／Ｆ５４からのＡＰＩ呼出の受付およびＡＰＩ実行結果の応答を行うアクセス制御サービスＡＰＩ４２である。なお、ＡＰＩとは、Ａｐｐｌｉｃａｔｉｏｎ　Ｐｒｏｇｒａｍ　Ｉｎｔｅｒｆａｃｅの略である。

【００３２】

アクセス制御部４３は、アクセス制御エージェントＩ／Ｆ４１、アクセス制御サービスＡＰＩ４２からアクセス権確認を受け付け、制御するアプリケーションモジュールである。アクセス部４３は、ＤＢドライバ部４４を介してデータベースサービス１６のデータ取得や更新を行う。

【００３３】

以後、上記ソフトウェアモジュールの協調により実行される一連のアクセス制御処理をアクセス制御サービス１４で実行される処理と記載する。なお、アクセス制御サービス１４で実行されるアクセス制御処理の詳細は後述する。

【００３４】

図５は、サービス１５Ａ、１５Ｂ上で動作するソフトウェアモジュールの構成例である。なお各ソフトウェアモジュールは図２で示したＨＤＤ２４に記憶されており、前述したようにＣＰＵ２１によってＲＡＭ２３にロードされ実行される。

【００３５】

Ｗｅｂサーバー５１は、クライアント１２からのＷｅｂリクエストを受け付けるＷｅｂインタフェースを備えたＷｅｂアプリケーションサーバーである。

【００３６】

Ｗｅｂアプリケーション５２は、Ｗｅｂサーバー５１上にアプリケーションとして構成され、Ｗｅｂサーバー５１が受け付けたＷｅｂリクエストに対して、サービスを提供する画面を生成する。

【００３７】

アクセス制御エージェント５３は、Ｗｅｂサーバー５１上にフィルタリングアプリケーションとして構成され、Ｗｅｂアプリケーション５２に対するＷｅｂリクエストをフィルタリングする。そして、アクセス制御サービス１４に構成されるアクセス制御エージェントＩ／Ｆ４１と通信することによりユーザーの認証確認処理および、アクセス制御処理を実行する。

【００３８】

図中５４は、アクセス制御サービス１４に構成されるアクセス制御サービスＡＰＩ４２を呼び出すためのアクセス制御サービスＩ／Ｆ５４である。アクセス制御サービスＩ／Ｆ５４は、Ｗｅｂアプリケーション５２から利用する事ができるよう構成される。

【００３９】

以後、上記ソフトウェアモジュールの協調により実行される一連のＷｅｂアプリケーション処理をサービス１５で実行される処理と記載する。なお、サービス１５で実行されるＷｅｂアプリケーション処理の詳細については後述する。

【００４０】

図６はユーザー情報のデータ構造および、ユーザー認証時に生成する認証トークン情報のデータ構造である。ユーザー情報は、ユーザーテーブル６０、ユーザーロールテーブル６１で構成され、データベースサービス１６にて管理されている。また、認証トークン情報は認証トークンキャッシュ６３で構成され、アクセス制御サービス１４のＲＡＭ２３に格納される。

【００４１】

ユーザーテーブル６０は、ユーザーを識別するためのユーザーＩＤ６０１、秘匿情報であるパスワード６０２、ユーザーのデータアクセス範囲を示すユーザータイプＩＤ６０３、ユーザーが所属するテナントを識別するためのテナントＩＤ６０４から成る。なお、ユーザーＩＤは、ユーザー識別情報と呼ばれることもある。

【００４２】

ユーザーロールテーブル６１は、ユーザーを識別するためのユーザーＩＤ６１１、ユーザーに設定されているロール情報を示すロールＩＤ６１２から成る。

【００４３】

認証トークンキャッシュ６３は、認証トークンを識別するための認証トークンＩＤ６３１、ユーザーを識別するためのユーザーＩＤ６３２、ユーザーに設定されている全てのロールＩＤであるロールＩＤ配列６３３から成る。これにより認証トークンとロールＩＤ配列とが関連付けられて管理される。

【００４４】

なお、認証トークンキャッシュ６３のデータはアクセス制御サービス１４においてユーザーの認証処理が実行され、認証が成功した時に生成される。

【００４５】

図７はロール情報およびリソース情報のデータ構造である。ロール情報はロールテーブル７０、リソース情報はリソーステーブル７１で構成される。また、ロールとリソースの関係はリソースロールテーブル７２で構成される。これらテーブルはデータベースサービス１６にて管理されている。

【００４６】

ロールテーブル７０は、ロールを識別するためのロールＩＤ７０１、ロールのカテゴリを識別するためのロールカテゴリＩＤ７０２から成る。

【００４７】

リソーステーブル７１は、リソースを識別するためのリソースＩＤ７１１、リソースのカテゴリを識別するためのリソースカテゴリＩＤ７１２、リソースとして管理される情報である保護アイテム７１３、リソースに対する権限情報を示す権限７１４から成る。

【００４８】

リソースロールテーブル７２は、リソースを識別するためのリソースＩＤ７２１、ロールを識別するためのロールＩＤ７２２から成る。

【００４９】

図８はテナント情報およびＡＰＩの実行権限情報のデータ構造である。テナント情報はテナントテーブル８０、ＡＰＩの実行権限情報はＡＰＩ権限テーブル８１で構成され、データベースサービス１６にて管理されている。

【００５０】

テナントテーブル８０は、テナントを識別するためのテナントＩＤ８０１、テナントに属するユーザーに設定されるユーザータイプＩＤ８０２、テナントのカテゴリを識別するためのテナントカテゴリＩＤ８０３から成る。

【００５１】

ＡＰＩ権限テーブル８１は、ＡＰＩを識別するためのファンクションＩＤ８１１、ロー

ルを識別するためのロールＩＤ８１２、および９び、操作テナントカテゴリＩＤ８１３、被操作テナントカテゴリＩＤ８１４からなる。なお、操作テナントカテゴリＩＤ８１３は、ＡＰＩ実行者が所属するテナントのカテゴリを識別するためのＩＤである。また、被操作テナントカテゴリＩＤ８１４は、ＡＰＩ実行対象のデータが所属するテナントのカテゴリを識別するためのＩＤである。

【００５２】

図６、図７、図８で説明した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。

【００５３】

以下、本発明の各サービスにおける処理フローについてシーケンスおよびフローチャートを用いて説明する。

【００５４】

図９はクライアント１２のＷｅｂブラウザからサービス１５に対してＷｅｂリクエストを行った場合の基本シーケンスである。なお、以後、クライアント１２のＷｅｂブラウザでの制御をクライアント１２の制御として説明する。

【００５５】

シーケンスＳ９．１において、クライアント１２は、サービス１５のＷｅｂサーバー５１に対してＷｅｂリクエストを実行する。Ｗｅｂサーバー５１は、フィルタリングアプリケーションであるアクセス制御エージェント５３に対して、Ｗｅｂリクエストを通知する（シーケンスＳ９．２）。

【００５６】

シーケンスＳ９．３において、アクセス制御エージェント５３は、アクセス制御サービス１４のアクセス制御エージェントＩ／Ｆ４１を介してアクセス制御部４３にて認証確認を行う。このとき、Ｗｅｂリクエストに含まれている認証トークンをアクセス制御エージェントＩ／Ｆ４１を介してアクセス制御部４３に通知する。アクセス制御部４３は、通知された認証トークンが認証トークンキャッシュ６３に格納されているかを検証する。シーケンスＳ９．３では初回アクセスであるため、通知された認証トークンは認証トークンキャッシュ６３に格納されていない。そのため、認証されていないと判断しアクセス制御エージェントＩ／Ｆ４１を介してアクセス制御エージェント５３に、ログインサービス１３に遷移するよう応答する。

【００５７】

シーケンスＳ９．４において、アクセス制御エージェント５３は、クライアント１２をログインサービス１３のログインアプリケーション３２にリダイレクトさせる。シーケンスＳ９．５において、ログインアプリケーション３２はログイン画面を生成し、クライアント１２に提示する。

【００５８】

シーケンスＳ９．６において、クライアント１２は、シーケンスＳ９．５において生成されたログイン画面を介してユーザーからのログイン指示を受けて、その際入力されたユーザー情報をログインアプリケーション３２にログイン通知する。ここで、ユーザー情報としては、ユーザーを識別するためのユーザーＩＤ、および秘匿情報であるパスワードが通知される。

【００５９】

ログイン通知を受けたログインアプリケーション３２は、シーケンスＳ９．７にて、アクセス制御エージェント３３および、アクセス制御エージェントＩ／Ｆ４１を介して、アクセス制御部４３に対して認証リクエスト（認証要求）を行う。

【００６０】

認証リクエストを受けたアクセス制御部４３は認証リクエストに含まれるユーザーＩＤ，パスワードが正当であるか検証する。その際、アクセス制御部４３はシーケンスＳ９．８にて、ＤＢドライバ部４４を介してデータベースサービス１６のユーザーテーブル６０に格納されているユーザーＩＤ６０１、パスワード６０２情報と比較検証する。なお、秘

匿情報であるパスワード６０２は、例えば、非可逆なハッシュ関数を適用し、秘匿化されて格納されていることが好ましい。その場合は、通知された認証リクエストのパスワード情報をパスワード６０２に格納する際に適用した関数で秘匿化し、比較することで検証が行われる。

【００６１】

シーケンスＳ９．８において、アクセス制御部４３は、検証の結果、ユーザー情報が正当である場合は認証トークンを生成し、認証トークンキャッシュ６３に格納する。その際、ＤＢドライバ部４４を介して、ユーザーロールテーブル６１からユーザーＩＤをキーとして、ロールＩＤ６１２を全て取得し、ユーザーＩＤと共に格納する。そして、アクセス制御部４３は、シーケンスＳ９．９にて、アクセス制御エージェントＩ／Ｆ４１を介してアクセス制御エージェント３３に生成した認証トークンを通知する。

【００６２】

シーケンスＳ９．１０において、アクセス制御エージェント３３は、受け付けた認証トークンを付与して、クライアント１２を、シーケンスＳ９．１にてリクエストされたＷｅｂサーバー５１にリダイレクトさせる。そして、シーケンスＳ９．１１において、Ｗｅｂサーバー５１はシーケンスＳ９．２と同様に、アクセス制御エージェント５３にＷｅｂリクエストを通知する。

【００６３】

シーケンスＳ９．１２において、アクセス制御エージェント５３は、アクセス制御サービス１４のアクセス制御エージェントＩ／Ｆ４１を介してアクセス制御部４３にて認証確認を行う。このとき、Ｗｅｂリクエストに含まれている認証トークンをアクセス制御エージェントＩ／Ｆ４１を介してアクセス制御部４３に通知する。アクセス制御部４３は、通知された認証トークンが認証トークンキャッシュ６３に格納されているかを検証する。通知された認証トークンは、シーケンスＳ９．８において認証トークンキャッシュに保存済みである。よって、シーケンスＳ９．１２において、アクセス制御エージェント５３により通知された認証トークンは認証トークンキャッシュ６３に格納されていると判断される。そのため、アクセス制御部４３は、認証されていると判断し、リソースアクセス可否確認を行う。リソースアクセス可否確認の処理詳細については後述する。次に、アクセス制御部４３は、リソースアクセスを許可と判断した場合は、ＤＢドライバ部４４を介してユーザーテーブル６０からユーザー情報を取得する。そして、アクセス制御部４３は、取得したユーザー情報をアクセス制御エージェントＩ／Ｆ４１を介してアクセス制御エージェント５３に通知する。

【００６４】

シーケンスＳ９．１３において、アクセス制御エージェント５３は、Ｗｅｂアプリケーション５２に対してＷｅｂリクエストおよび、ユーザー情報を通知する。ユーザー情報の通知を受けたＷｅｂアプリケーション５２は、不図示の業務用の画面を生成し、シーケンスＳ９．１４にてクライアント１２に提示する。そして、クライアント１２は、シーケンスＳ９．１５にてユーザーからの画面操作を受けて、シーケンスＳ９．１６にてＷｅｂサーバー５１に対して操作されたことを示すＷｅｂリクエストを通知する。

【００６５】

シーケンスＳ９．１７、シーケンスＳ９．１８、およびシーケンスＳ９．１９は、それぞれ、シーケンスＳ９．１１、シーケンスＳ９．１２、およびシーケンスＳ９．１３と同様の処理であるため、説明を省く。

【００６６】

次に、シーケンスＳ９．１５におけるユーザーの操作に伴って、アクセス制御サービス１４のアクセス制御サービスＡＰＩ４２のＡＰＩが実行されるケースとして説明する。

【００６７】

シーケンスＳ９．２０において、Ｗｅｂアプリケーション５２は、アクセス制御サービスＩ／Ｆ５４を介してアクセス制御サービスＡＰＩ４２のＡＰＩを呼び出す。その際、ＡＰＩの引数として認証トークンを通知する。

【００６８】
シーケンスＳ９．２１において、アクセス制御サービスＡＰＩ４２はＡＰＩ実行権限の確認を行う。ＡＰＩ実行権限の確認処理詳細については後述する。ＡＰＩ実行が許可された場合、アクセス制御サービスＡＰＩ４２はＡＰＩの処理内容に従ってアクセス制御部４３に対してデータ取得をリクエストする（シーケンスＳ９．２２）。その際、認証トークンを通知する。
【００６９】
シーケンスＳ９．２３において、アクセス制御部４３はデータアクセスの可否の確認処理を行う（シーケンスＳ９．２３）。データアクセスの可否の確認処理詳細については後述する。データアクセスが許可された場合、アクセス制御部４３はＤＢドライバ部４４を介してデータを取得し、シーケンスＳ９．２４においてアクセス制御サービスＡＰＩ４２へ通知する。
【００７０】
シーケンスＳ９．２５において、アクセス制御サービスＡＰＩ４２は、取得したデータをもとにＡＰＩ応答を生成し、アクセス制御サービスＩ／Ｆ５４を介してＷｅｂアプリケーション５２に通知する。
【００７１】
シーケンスＳ９．２６において、Ｗｅｂアプリケーション５２はＡＰＩ応答に従った画面を生成し、クライアント１２に提示する。
【００７２】
上記、図９を用いて説明した基本シーケンスによって、ユーザーの認証および、ユーザーのアクセス権制御処理を実行する。
【００７３】
図１０は、図９を用いて説明した基本シーケンスにおける、アクセス制御サービス１４におけるアクセス制御処理フローである。
【００７４】
以下、図９におけるシーケンスＳ９．２０のアクセス制御サービスＡＰＩ４２へのＡＰＩ呼び出しにおいて、ステップＳ１００１が実行されるフローを説明する。
【００７５】
アクセス制御サービスＡＰＩ４２は、ステップＳ１００１にてＡＰＩ呼び出しを受けると、ステップＳ１００２にて通知された認証トークンの有効性を確認する。より具体的には、アクセス制御サービスＡＰＩ４２は、アクセス制御部４３に対して、通知された認証トークンが認証トークンキャッシュ６３に格納されているかを確認する。そして、認証トークンが無効であると判断された場合（認証トークンが格納されていない場合）は、ステップＳ１００３にて認証トークンが無効であるため、ＡＰＩが実行できない旨を応答する。認証トークンが有効である場合、ステップＳ１００４にて、アクセス制御サービスＡＰＩ４２は、アクセス制御部４３を介して認証トークンをキーにユーザーＩＤを取得する。そして、アクセス制御サービスＡＰＩ４２は、続けてユーザーテーブル６０、テナントテーブル８０よりユーザー情報、テナント情報を取得する。
【００７６】
次に、ステップＳ１００５において、アクセス制御サービスＡＰＩ４２は、ＤＢドライバ部４４を介してＡＰＩ権限テーブルから呼び出されたＡＰＩのファンクションＩＤをキーに情報を取得する。アクセス制御サービスＡＰＩ４２はステップＳ１００６にて、取得したユーザー情報およびテナント情報とＡＰＩ権限情報を比較する。ステップＳ１００７にて、アクセス制御サービスＡＰＩ４２は、ＡＰＩが操作する対象のテナントのカテゴリＩＤと、取得したテナント情報のテナントカテゴリＩＤ８０３とを用いて、ロールＩＤ８１２を取得する。そして、ユーザー情報に取得したロールＩＤが含まれていない場合はＡＰＩ実行を拒否し、ステップＳ１００８にて認可エラーとして応答する。ユーザー情報に取得したロールＩＤが含まれている場合はＡＰＩ実行を許可し、ステップＳ１００９にてＡＰＩを実行する。この処理がＡＰＩ検証処理である。

【００７７】
ステップＳ１０１０において、アクセス制御サービスＡＰＩ４２は、ＡＰＩ実行内容にリソースアクセスを含まない場合は、ステップＳ１０１１にてＡＰＩの実行結果を生成して応答する。ＡＰＩ実行内容にリソースアクセスが含まれる場合は、アクセス制御サービスＡＰＩ４２は、ステップＳ１０１２において、アクセス制御部４３に対してリソースアクセス可否確認を行う。ここでリソースとしては、データベースサービス１６に格納されている情報であるデータリソースや、サービス１５が提供するＷｅｂアプリケーションのＵＲＬリソースである。アクセス制御部４３におけるリソースアクセス可否確認の処理は後述する。
【００７８】
ステップＳ１０１３において、アクセス制御サービスＡＰＩ４２は、アクセス制御部４３におけるリソースアクセス可否確認の結果が拒否だった場合は、ステップＳ１００８にて認可エラーとして応答する。アクセス制御サービスＡＰＩ４２は、アクセス制御部４３におけるリソースアクセス可否確認の結果が許可だった場合は、取得したリソース情報をもとにＡＰＩの実行結果を生成して応答する。
【００７９】
以下、図９におけるシーケンスＳ９．１２、シーケンスＳ９．１８、シーケンスＳ９．２２におけるアクセス制御部４３へのアクセス権限確認においてステップＳ１０２１が実行されるフローを説明する。また、ステップＳ１０２１は、図１０におけるステップＳ１０１１のリソースアクセス可否確認においても実行される。
【００８０】
シーケンスＳ９．１２、シーケンスＳ９．１８では、ＵＲＬリソースに対するリソースアクセス可否確認として、また、シーケンスＳ９．２２では、データリソースに対するリソースアクセス可否確認として、ステップＳ１０２１が実行される。
【００８１】
ステップＳ１０２１において、アクセス制御部４３は、リソースアクセス可否確認のリクエストを受け付ける。この際、リソースアクセスを実行するユーザーの認可トークン、対象のリソースカテゴリ、保護アイテム情報、および保護アイテムに対するアクションを取得する。ここで、保護アイテム情報とは、リソースカテゴリがＵＲＬリソースである場合はＵＲＬ、データリソースである場合は、ユーザータイプＩＤおよび取得条件となる。また、アクションは、保護アイテムに対するＣＲＵＤ（Ｃｒｅａｔｅ，　Ｒｅａｄ，　Ｕｐｄａｔｅ，　Ｄｅｌｅｔｅ）から選択される。
【００８２】
ステップＳ１０２２において、アクセス制御部４３は受け付けた認証トークンが認証トークンキャッシュ６３に格納されているかを確認し、有効性を検証する。検証の結果、無効であった場合はステップＳ１０２３にて認証トークンが無効である旨を通知する。検証の結果、認証トークンが有効であった場合、アクセス制御部は、受信した認証トークンに関連づけられたユーザーＩＤ６３２およびロールＩＤ配列６３３を取得する（ステップＳ１０２４）。なお、本願ではロールＩＤのことをロール情報と呼ぶ場合もある。
【００８３】
ステップＳ１０２５において、アクセス制御部４３は、リソースアクセス可否確認依頼に含まれるリソースカテゴリ（リソース種別）を特定する。ＵＲＬリソースであった場合、アクセス制御部４３の処理はステップＳ１０２６へ、データリソースだった場合、ステップＳ１０２７へ進む。
【００８４】
ステップＳ１０２６、ステップＳ１０２７ではアクセス制御部４３は、リソースカテゴリＩＤ、保護アイテム情報をキーとしてリソーステーブル７１および、リソースロールテーブル７２よりリソースに紐づいた全てのロールＩＤおよび権限を取得する。
【００８５】
ステップＳ１０２８において、アクセス制御部４３は、取得したロールＩＤおよび権限

と、リクエストで受け付けた認証トークンに紐づくロールＩＤおよび、アクションを比較する。つまり、各テーブルの管理内容に基づいてＳ１０２８の処理が実現される。そして、ステップＳ１０２９において、アクセス制御部４３は比較検証の結果、アクセス権限がない場合、つまりアクセス拒否の場合、ステップＳ１０３０にてアクセス拒否通知を行う。アクセス制御部４３は比較検証の結果、アクセス権限がある場合、つまりアクセス許可の場合、ステップＳ１０３１にて対象のリソースを取得する。たとえば、リソースカテゴリがデータリソースであった場合は、指定の取得範囲を条件として、ＤＢドライバ部４４を介してデータを取得する。この際、必ず許可されたユーザータイプＩＤの範囲で取得データ範囲を絞り込まれる。結果、権限を保持していない他のテナントのデータを取得する事を防ぐことができる。Ｓ１０２８－Ｓ１０２９の処理が、ＵＲＬ検証処理、または、データ配信検証処理に相当する。

【００８６】

ステップＳ１０３２において、アクセス制御部４３は取得したリソースおよび、アクセス許可を通知する。

【００８７】

上記、図９の基本シーケンスおよび、図１０のアクセス制御フローにより、ロール定義およびロール制御という統一的な方式によって、ＵＲＬリソース、データリソース、およびＡＰＩ実行権限確認を実現する事ができる。

【００８８】

次に、図６、図７、図８で説明した各データ構造をもつテーブルに対して、データ例として図１１、図１２、図１３を例示する。そして、サービス１５に展開されるサービスを図１４として例示し、具体的な業務フローとしてアクセス制御フローを図１５、図１６を用いて説明する。ここで説明するデータやサービスは一例であり、本発明の内容が説明した内容に制限されるものではない。

【００８９】

図１１において、１１１は、ユーザーテーブル６０のデータ例である。次に、１１２はユーザーロールテーブル６１のデータ例である。

【００９０】

図１２において、１２１はロールカテゴリの定義例である。本例では、データ管理権限を示す管理ロール、ユーザーとの利用契約を示す製品ロール、そして、データに対するアクセス範囲を示すテナントロールが定義されている。

【００９１】

図１２において、１２３はリソースカテゴリの定義例である。本例では、Ｗｅｂアクセスする対象としてのＵＲＬリソース、データベースサービス１６で管理されているデータを示すデータリソースが定義されている。

【００９２】

図１２において、１２２はロールテーブルのデータ例である。次に１２４はリソースロールテーブル（ロール管理情報ともいう）のデータ例である。そして、１２５はリソーステーブルのデータ例である。

【００９３】

図１３において、１３１はテナントカテゴリの定義例である。本例では、ユーザーとの利用契約を行う販売者が所属する販売テナント、利用者である顧客テナント、および、ユーザー本人である自身というカテゴリが定義されている。

【００９４】

図１３において、１３２はテナントテーブルのデータ例である。図１３のテナントテーブル１３２は、新規ユーザーをユーザーテーブル１１１に追加する際に利用される。次に１３３はＡＰＩ権限テーブルのデータ例である。本データ例では、以下のＡＰＩを例示している。利用契約を行ったユーザーのテナントを、販売者が作成するためのＡＰＩであるＣｒｅａｔｅＴｅｎａｎｔ。ユーザーのロール定義の設定を変更するＣｈａｎｇｅＲｏｌｅ。および、テナントに所属するユーザーを検索するためのＳｅａｒｃｈＵｓｅｒという

ＡＰＩを例示している。
【００９５】
図１４は、サービス１５が、Ｗｅｂアプリケーション例として、ユーザーやユーザーのロール設定を管理するためのＷｅｂアプリケーション（ユーザー管理サービスとする）をサービスしたときの画面フロー例である。
【００９６】
１４０１は、ログインサービス１３で生成されるログイン画面の例である。ユーザーが図中のユーザーＩＤおよびパスワードを入力のうえでログイン画面を押下し、ログイン成功およびアクセスが許可されると１４０２のメニュー画面に遷移する。本例では、ユーザーテーブルのデータ例１１１に登録されているＣｕｓｔｏｍｅｒＡｄｍｉｎ０１というユーザーでログインしたとする。
【００９７】
１４０２は、ユーザー管理サービスのメニュー画面の例である。ユーザーが図中のユーザー検索リンクを押下しアクセスが許可されると、１４０３のユーザー検索画面に遷移する。
【００９８】
１４０３は、ユーザー管理サービスのユーザー検索画面の例である。ユーザーが図中のユーザー名に検索項目を入力し検索ボタンを押下し、ＳｅａｒｃｈＵｓｅｒ　ＡＰＩの実行権限が許可されると、ユーザー検索が実行され、１４０４の検索結果画面に遷移する。ここで、検索項目としては「＊」というワイルドカード（全件検索）指定したとする。
【００９９】
１４０４は、ユーザー管理サービスのユーザー検索画面における検索結果画面の例である。ここでは、ユーザーテーブルのデータ例１１１に、ＣｕｓｔｏｍｅｒＡｄｍｉｎ０１が所属するＴＡ０００００００２テナントのユーザーが全て表示される。
【０１００】
図１５は、図１４の画面フローに従ってユーザーが操作した場合のシーケンスである。
【０１０１】
シーケンスＳ１５．１において、クライアント１２はサービス１５のメニュー画面１４０２へＷｅｂリクエストを行う。シーケンスＳ１５．２において、サービス１５はアクセス制御エージェントＩ／Ｆ４１に対して認証確認を行う。ここで認証確認フローは図９のシーケンスＳ９．１－Ｓ９．３に対応する。
【０１０２】
シーケンスＳ１５．３において、サービス１５は、クライアント１２をログインサービス１３にリダイレクトさせる。そして、シーケンスＳ１５．４において、ログインサービス１３はログイン画面１４０１を提示する。これらの処理は、図９のシーケンスＳ９．４－Ｓ９．５に対応する。
【０１０３】
シーケンスＳ１５．５において、ユーザーはユーザーＩＤ：ＣｕｓｔｏｍｅｒＡｄｍｉｎ０１としてログイン操作を行う。ログイン操作を受けたログインサービス１３はアクセス制御エージェントＩ／Ｆ４１に対して認証処理を依頼する。これらの処理は、図９のシーケンスＳ９．６に対応する。
【０１０４】
シーケンスＳ１５．６における認証処理については、図９のシーケンスＳ９．７にて説明済みであるため省略する。ここで、認証成功した場合、アクセス制御部４３では、認証トークンキャッシュ６３に、生成した認証トークンのＩＤ、ユーザーＩＤ：ＣｕｓｔｏｍｅｒＡｄｍｉｎ０１および、ロールＩＤを格納する。ここで、ロールＩＤとしてはユーザーロールテーブルのデータ例１１２に記載の「ＣｕｓｔｏｍｅｒＡｄｍｉｎ、Ｃｕｓｔｏｍｅｒ、ＴＡ０００００００２、Ｐｒｏｖｉｓｉｏｎｉｎｇ」が格納される。
【０１０５】
認証を受けたログインサービス１３は、認証トークンを付与してシーケンスＳ１５．７

にて、クライアント１２をサービス１５のメニュー画面１４０２へリダイレクトさせる。
【０１０６】
シーケンスＳ１５．８において、サービス１５はアクセス制御エージェントＩ／Ｆ４１を介してアクセス制御部４３に対して認証確認、アクセス権限確認、およびユーザー情報取得を行う。
【０１０７】
アクセス制御部４３では、認証トークンキャッシュ６３に認証トークンが格納されているかを確認し、格納されている場合は、ユーザーＩＤおよびロールＩＤ配列を取得する。今回は格納されているため、ユーザーＩＤ：ＣｕｓｔｏｍｅｒＡｄｍｉｎ０１、ロールＩＤ配列「ＣｕｓｔｏｍｅｒＡｄｍｉｎ、Ｃｕｓｔｏｍｅｒ、ＴＡ００００００００２、Ｐｒｏｖｉｓｉｏｎｉｎｇ」が取得される。
【０１０８】
次に、アクセス制御部４３では、図１０におけるステップＳ１０２１が実行される。このとき、対象のリソースとして「ｈｔｔｐ：ｘｘｘ．ｃｏｍ／ｍｅｎｕ／ｘｘｘ．ｈｔｍｌ」が渡されるとする。これは、例えば、ユーザーが上記アドレスをブラウザに入力することにより実現される。このリソースはリソーステーブルのデータ例１２５のリソースＩＤ：Ｒ００００００００１に格納されているデータと一致する。そして、リソースＩＤ：Ｒ００００００１は、リソースロールテーブルのデータ例１２４にて、ロールＩＤ：Ｃｕｓｔｏｍｅｒに割り当てられている。アクセス制御部４３は図１０におけるステップＳ１０２８にて、ロールＩＤ：Ｃｕｓｔｏｍｅｒが、取得したロールＩＤ配列に含まれるかを確認する。具体的には、認証トークンキャッシュ６３から、認証トークンに関連付けられたロール情報が決定される。その決定されたロール情報が、取得したロールＩＤ配列に含まれるかが確認される。つまり、アクセス制御部４３が、リソースカテゴリとしてＵＲＬである場合、そのＵＲＬに一致するロールＩＤを図１２のリソースロールテーブル１２４から取得する。そしてリソースロールテーブル１２４から取得されたロールＩＤ認証トークンに割り当てられたロールＩＤ配列とに基づいて、アクセスを許可するか否かを判定する。なお、この処理は、他の段階でも同様に実行される。今回のデータ例ではロールＩＤ配列にＣｕｓｔｏｍｅｒが含まれているため、アクセス制御部４３は、アクセス許可としてユーザーテーブルのデータ例１１１から情報を取得し、サービス１５に通知する。シーケンスＳ１５．９にてサービス１５はクライアント１２にメニュー画面１４０２を提示する。ここまでは、図１４におけるメニュー画面１４０２の表示、画面１４０２におけるＡＰＩの実行、リソースデータの提供という3段階（3レイヤー）の第1段階（第1レイヤー）の処理となる。　次に、メニュー画面１４０２にてユーザーがユーザー検索リンクを押下した場合のシーケンスを説明する。
【０１０９】
シーケンスＳ１５．１０において、ユーザー検索が選択されると、サービス１５はシーケンスＳ１５．１１にてアクセス制御エージェントＩ／Ｆ４１を介して、アクセス制御部４３に認証確認、アクセス権限確認、ユーザー情報取得を依頼する。
【０１１０】
アクセス制御部４３では、認証トークンキャッシュ６３に認証トークンが格納されているかを確認し、格納されている場合は、ユーザーＩＤおよびロールＩＤ配列を取得する。今回は格納されているため、ユーザーＩＤ：ＣｕｓｔｏｍｅｒＡｄｍｉｎ０１、ロールＩＤ配列「ＣｕｓｔｏｍｅｒＡｄｍｉｎ、Ｃｕｓｔｏｍｅｒ、ＴＡ００００００００２、Ｐｒｏｖｉｓｉｏｎｉｎｇ」が取得される。
【０１１１】
次に、アクセス制御部４３では、図１０におけるステップＳ１０２１が実行される。このとき、対象のリソースとして「ｈｔｔｐ：ｘｘｘ．ｃｏｍ／ｓｅａｒｃｈ／ｘｘｘ．ｈｔｍｌ」が渡されるとする。このリソースはリソーステーブルのデータ例１２５のリソースＩＤ：Ｒ００００００００２に格納されているデータと一致する。そして、リソースＩＤ：Ｒ００００００２は、リソースロールテーブルのデータ例１２４にて、ロールＩＤ：Ｐ

ｒｏｖｉｓｉｏｎｉｎｇに割り当てられている。アクセス制御部４３は図１０におけるステップＳ１０２８にて、ロールＩＤ：Ｐｒｏｖｉｓｉｏｎｉｎｇが、取得したロールＩＤ配列に含まれるかを確認する。今回のデータ例ではロールＩＤ配列にＰｒｏｖｉｓｉｏｎｉｎｇが含まれているため、アクセス制御部４３は、アクセス許可としてユーザーテーブルのデータ例１１１から情報を取得し、サービス１５に通知する。シーケンスＳ１５．１２にてサービス１５はクライアント１２にユーザー検索画面１４０３を提示する。ここまでは、図１４におけるメニュー画面１４０２の表示、画面１４０２におけるＡＰＩの実行、リソースデータの提供という３段階（３レイヤー）の第２段階（第２レイヤー）の処理となる。

【０１１２】

次に、ユーザー検索画面１４０３にてユーザーが検索項目としてワイルドカードである「＊」を入力し、ユーザー検索ボタンを押下した場合のシーケンスを説明する。

【０１１３】

シーケンスＳ１５．１３において、ユーザー検索が実行されると、サービス１５はシーケンスＳ１５．１４において、アクセス制御サービスＡＰＩ４２に対して、Ｓｅａｒｃｈ Ｕｓｅｒ　ＡＰＩを実行する。この際、認証トークンを通知する。

【０１１４】

シーケンスＳ１５．１５において、アクセス制御サービスＡＰＩ４２では、図１０におけるステップＳ１００１が実行される。このとき、ユーザー検索が実行されたため、対象のＡＰＩとして「ＳｅａｒｃｈＵｓｅｒ」が渡される。アクセス制御サービスＡＰＩ４２は、認証トークンを検証し、ユーザーＩＤおよびロールＩＤ配列を取得する。次に、図１０におけるステップＳ１００５が実行され、ＡＰＩ権限テーブルのデータ例１３３からファンクションＩＤ：ＳｅａｒｃｈＵｓｅｒが取得される。そして、ロールＩＤ配列のロールＩＤ：ＣｕｓｔｏｍｅｒＡｄｍｉｎ、Ｃｕｓｔｏｍｅｒが該当する二つのデータが取得される。

【０１１５】

アクセス制御サービスＡＰＩ４２では、ステップＳ１００６において、ＡＰＩ権限テーブルから取得したデータより、操作者のテナントカテゴリＩＤ：ＣｕｓｔｏｍｅｒＴｅｎａｎｔにおける、Ｓｅａｒｃｈ Ｕｓｅｒ　ＡＰＩの実行権限を以下と判断する。被操作テナントカテゴリ：ＣｕｓｔｏｍｅｒＴｅｎａｎｔ、Ｓｅｌｆの範囲で許可される。

【０１１６】

シーケンスＳ１５．１６において、アクセス制御サービスＡＰＩ４２は、図１０におけるステップＳ１０１０にて、データリソースへのアクセスとしてアクセス制御部４３にリソースアクセス可否確認依頼を実行する。その際、認証トークンおよび、データ取得範囲として、ユーザーデータテーブルに対する「＊」を通知する。

【０１１７】

シーケンスＳ１５．１７において、アクセス制御部４３では、認証トークンキャッシュ６３に認証トークンが格納されているかを確認し、格納されている場合は、ユーザーＩＤおよびロールＩＤ配列を取得する。今回は格納されているため、ユーザーＩＤ：ＣｕｓｔｏｍｅｒＡｄｍｉｎ０１、ロールＩＤ配列「ＣｕｓｔｏｍｅｒＡｄｍｉｎ、Ｃｕｓｔｏｍｅｒ、ＴＡ０００００００２、Ｐｒｏｖｉｓｉｏｎｉｎｇ」が取得される。

【０１１８】

次に、アクセス制御部４３では、図１０におけるステップＳ１０２１が実行される。このとき、対象のリソースとして「ＣｕｓｔｏｍｅｒＴｅｎａｎｔ、Ｓｅｌｆ」が渡されるとする。アクセス制御部４３において、データアクセス可能な範囲は、操作者が所属するテナントまででであるため、保護アイテムは、ユーザーテーブルデータ例１１１に登録されているＴＹ０００００００２となる。このリソースはリソーステーブルのデータ例１２５のリソースＩＤ：Ｒ０００００００４に格納されているデータと一致する。そして、リソースＩＤ：Ｒ００００００４は、リソースロールテーブルのデータ例１２４にて、ロールＩＤ：ＴＡ０００００００２に割り当てられている。アクセス制御部４３は図１０におけ

るステップＳ１０２８にて、ロールＩＤ：ＴＡ０００００００２が、取得したロールＩＤ配列に含まれるかを確認する。今回のデータ例ではロールＩＤ配列に含まれているため、アクセス制御部４３は、ＴＹ０００００００２の範囲でアクセス許可としてユーザーテーブルのデータ例１１１から情報を取得する。その際、データ範囲がワイルドカードであるため、ＴＹ０００００００２の範囲でユーザーデータテーブルから取得可能な全データが取得され、シーケンスＳ１５．１８にてアクセス制御サービスＡＰＩ４２に通知する。

【０１１９】

シーケンスＳ１５．１９において、アクセス制御サービスＡＰＩ４２は、Ｓｅａｒｃｈ Ｕｓｅｒ　ＡＰＩの応答として、取得したユーザー情報をサービス１５に応答する。

【０１２０】

シーケンスＳ１５．２０において、サービス１５は取得したユーザー情報から検索結果画面１４０４を生成してクライアント１２に提示（配信）する。

【０１２１】

ここまでは、図１４におけるメニュー画面１４０２の表示、画面１４０２におけるＡＰＩの実行、リソースデータの提供という３段階（３レイヤー）の第３段階（第３レイヤー）の処理となる。

【０１２２】

本願では、全ての段階（全てのレイヤー）の実行確認をロールにて行うことにより、開発、および運用、保守のコストを軽減することができる。

【０１２３】

上記、図１５のシーケンスおよび、図１０のアクセス制御フローにより、ロール定義およびロール制御という統一的な方式によって、ＵＲＬリソース、データリソース、およびＡＰＩ実行権限確認を実現する事ができる。

【０１２４】

図１６において１６１はユーザー管理サービスのメニューでロール管理を選択し、不図示のロール設定画面よりユーザーのロール設定を変更した場合に実行されるＣｈａｎｇｅ Ｒｏｌｅ　ＡＰＩの実行可否を定義するロール操作可否テーブルのデータ例である。

【０１２５】

図１５のシーケンスＳ１５．１５において、アクセス制御サービスＡＰＩ４２は、ＡＰＩ実行が許可となった場合に、以下の処理を行う。

【０１２６】

ロール操作可否テーブルから、ＡＰＩ実行者のロールＩＤで絞り込みを行う。ロール設定の変更対象であるロールＩＤから、ロールカテゴリＩＤを取得し、被操作ロールカテゴリＩＤにて絞り込みを行う。そこで、被操作ロールカテゴリが「＊」である場合は、可否を確認する。最後に、ロールＩＤで絞り込みを行い、結果「＊」である場合は、可否を確認する。ロールＩＤが存在しない場合は、拒否と判断する。そして、可否の確認結果、Ａｌｌｏｗである場合はＡＰＩを実行し、Ｄｅｎｙである場合はＡＰＩ実行を拒否する。

【０１２７】

本発明では、ロール操作可否テーブルのデータ例１６１の定義１６１１に示している通り、被操作ロールカテゴリＩＤがＭａｎａｇｅｍｅｎｔＲｏｌｅである場合、操作ロールＩＤがＡｄｍｉｎロールをもつ必要がある。

【０１２８】

本発明では、ロール操作可否テーブルのデータ例１６１の定義１６１２に示している通り、被操作ロールＩＤ：Ｃｕｓｔｏｍｅｒは誰からも操作されないよう定義されている。これにより、他のテナントカテゴリＩＤのロールＩＤを誤って設定する事を防ぐことができる。すなわち、他のテナントカテゴリＩＤのＵＲＬリソースに対するアクセス制限や、ＡＰＩの実行制限を行う事ができる。

【０１２９】

本発明では、ロール操作可否テーブルのデータ例１６１の定義１６１４に示している通り、被操作ロールカテゴリＩＤがＴｅｎａｎｔＲｏｌｅである場合、どのような操作ロー

ルＩＤ、被操作ロールＩＤであっても可否を「Ｄｅｎｙ」と設定する。これにより、ＴｅｎａｎｔＲｏｌｅカテゴリのロールを、所属外のテナントに誤って設定する事を防ぐことができる。すなわち、自身が所属するテナント外のテナントデータに対するアクセス制限をかけることが可能となる。

【０１３０】

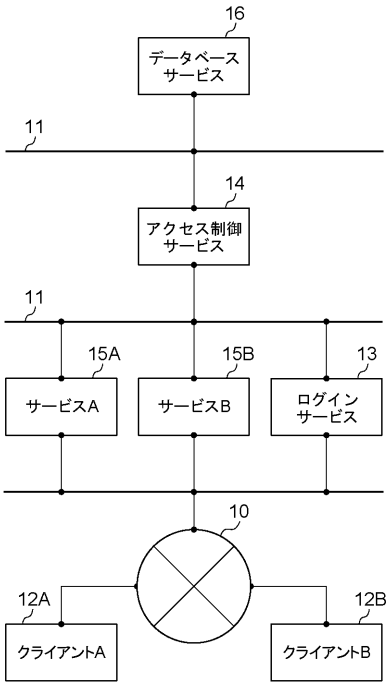
本願の処理は認証システムにより実行される。なお、認証システムは、１台の情報処理装置（例えば、サーバー）で構成されても良いし、複数台の情報処理装置（例えば、サーバー）で構成されても良い。

【符号の説明】

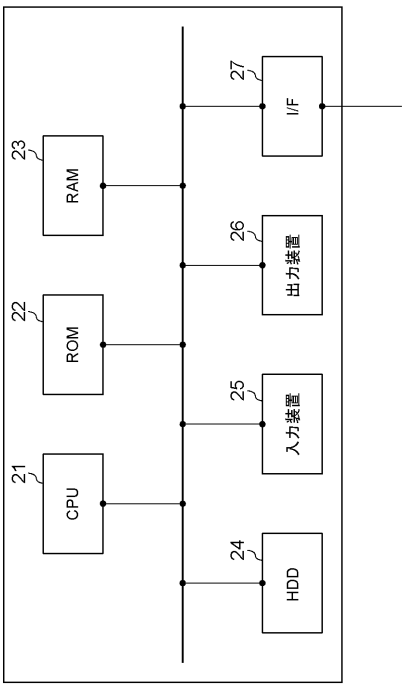
【０１３１】

１０　ＷＡＮ
１１　ＬＡＮ
１２　クライアント
１３　ログインサービス
１４　アクセス制御サービス
１５　サービス
１６　データベースサービス

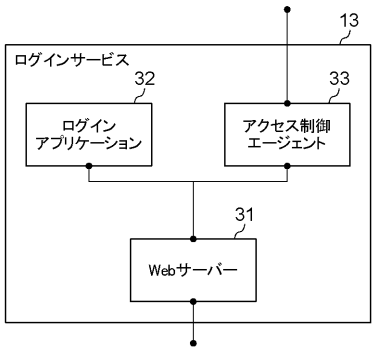
【図１】



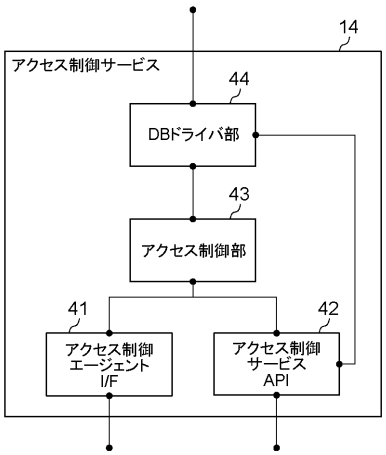

【図２】



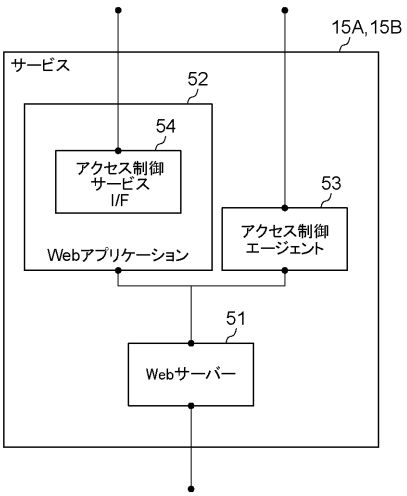
【図３】


13
ログインサービス
32
33
ログイン
アプリケーション
アクセス制御
エージェント
31
Webサーバー

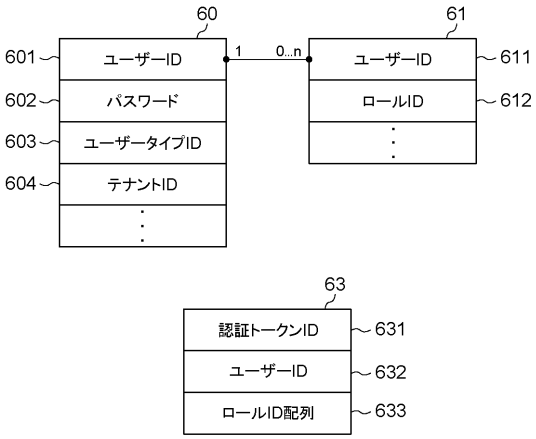
【図４】


14
アクセス制御サービス
44
DBドライバ部
43
アクセス制御部
41
42
アクセス制御
エージェント
I/F
アクセス制御
サービス
API

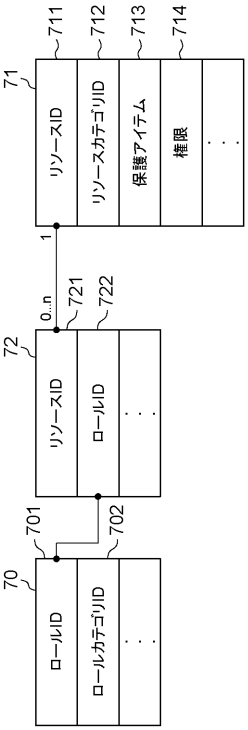
【図５】


15A,15B
サービス
52
54
アクセス制御
サービス
I/F
Webアプリケーション
53
アクセス制御
エージェント
51
Webサーバー

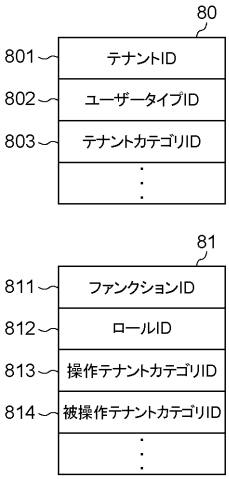
【図６】


60
61
601
ユーザーID
1
0...n
ユーザーID
611
602
パスワード
ロールID
612
603
ユーザータイプID
604
テナントID
63
認証トークンID
631
ユーザーID
632
ロールID配列
633

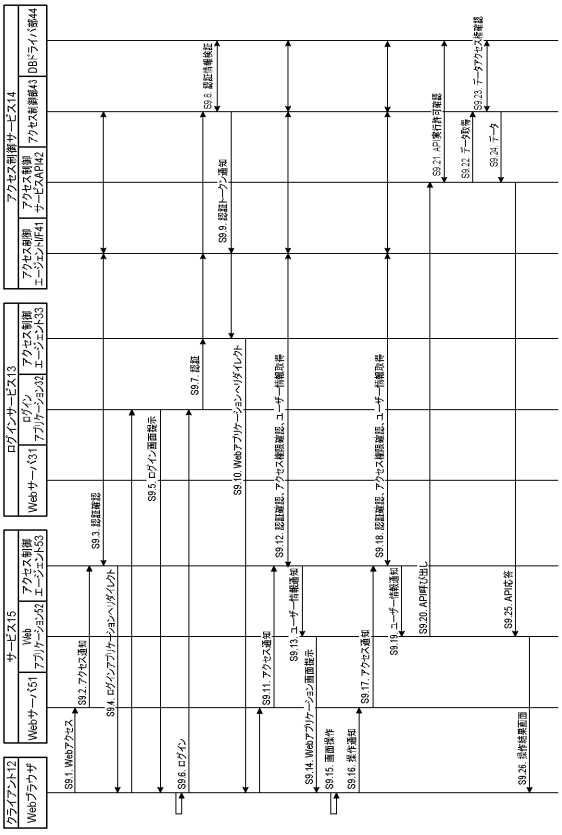
【図７】


71
711
712
713
714
リソースID
リソースカテゴリID
保護アイテム
権限
1
0..n
721
722
72
リソースID
ロールID
701
702
70
ロールID
ロールカテゴリID

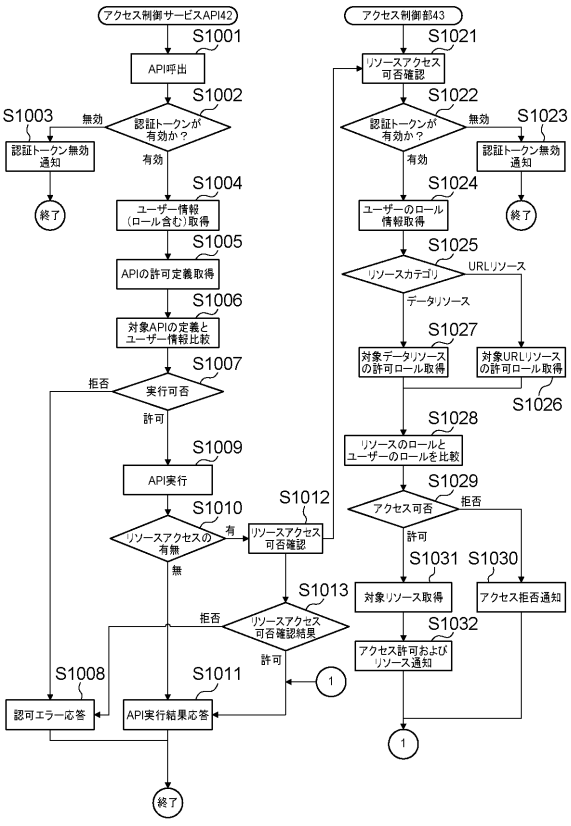
【図８】


80
801
テナントID
802
ユーザータイプID
803
テナントカテゴリID
81
811
ファンクションID
812
ロールID
813
操作テナントカテゴリID
814
被操作テナントカテゴリID

【図９】


クライアント12
Webブラウザ
サービス15
Webサーバ51
Webアプリケーション52
アクセス制御エージェント53
ログインサービス13
Webサーバ31
ログインアプリケーション32
アクセス制御エージェント33
アクセス制御エージェントIF41
アクセス制御サービスAPI42
アクセス制御部43
DBドライバ部44
アクセス制御サービス14
S9.1. Webアクセス
S9.2. アクセス通知
S9.3. 認証確認
S9.4. ログインアプリケーションへリダイレクト
S9.5. ログイン画面表示
S9.6. ログイン
S9.7. 認証
S9.8. 認証情報検証
S9.9. 認証トークン通知
S9.10. Webアプリケーションへリダイレクト
S9.11. アクセス通知
S9.12. 認証確認部、アクセス権限確認部、ユーザー情報取得
S9.13. ユーザー情報通知
S9.14. Webアプリケーション画面表示
S9.15. 画面操作
S9.16. 操作通知
S9.17. アクセス通知
S9.18. 認証確認部、アクセス権限確認部、ユーザー情報取得
S9.19. ユーザー情報通知
S9.20. API呼び出し
S9.21. API実行許可確認
S9.22. データ取得
S9.23. データアクセス権確認
S9.24. データ
S9.25. API応答
S9.26. 処理結果画面

【図１０】


アクセス制御サービスAPI42
S1001
API呼出
S1002
認証トークンが有効か？
無効
S1003
認証トークン無効通知
終了
有効
S1004
ユーザー情報（ロール含む）取得
S1005
APIの許可定義取得
S1006
対象APIの定義とユーザー情報比較
S1007
実行可否
拒否
許可
S1009
API実行
S1010
リソースアクセスの有無
有
S1012
リソースアクセス可否確認
無
S1013
リソースアクセス可否確認結果
拒否
許可
S1008
認可エラー応答
S1011
API実行結果応答
終了
1
アクセス制御部43
S1021
リソースアクセス可否確認
S1022
認証トークンが有効か？
無効
S1023
認証トークン無効通知
終了
有効
S1024
ユーザーのロール情報取得
S1025
リソースカテゴリ
URLリソース
データリソース
S1027
対象データリソースの許可ロール取得
対象URLリソースの許可ロール取得
S1026
S1028
リソースのロールとユーザーのロールを比較
S1029
アクセス可否
拒否
許可
S1031
対象リソース取得
S1030
アクセス拒否通知
S1032
アクセス許可およびリソース通知
1

【図１１】

111

| ユーザーテーブル | | | |
|---|---|---|---|
| *ユーザーID* | *パスワード* | *ユーザータイプID* | *テナントID* |
| SalesAdmin01 | ********** | TY00000001 | TA00000001 |
| SalesUser01 | ********** | TY00000001 | TA00000001 |
| CustomerAdmin01 | ******** | TY00000002 | TA00000002 |
| CustomerUser01 | ******** | TY00000002 | TA00000002 |
| … | … | … | … |

112

| ユーザーロールテーブル | |
|---|---|
| *ユーザーID* | *ロールID* |
| SalesAdmin01 | SalesAdmin |
| SalesAdmin01 | Sales |
| SalesAdmin01 | TA00000001 |
| SalesAdmin01 | Provisioning |
| SalesUser01 | Sales |
| SalesAdmin01 | TA00000001 |
| CustomerAdmin01 | CustomerAdmin |
| CustomerAdmin01 | Customer |
| CustomerAdmin01 | TA00000002 |
| CustomerAdmin01 | Provisioning |
| CustomerUser01 | Customer |
| CustomerUser01 | TA00000002 |
| … | … |

【図１２】

121

| ロールカテゴリテーブル | |
|---|---|
| *ID* | *説明* |
| ManagementRole | 管理ロール |
| ProductRole | 製品ロール |
| TenantRole | テナントロール |
| … | … |

122

| ロールテーブル | |
|---|---|
| *ロールID* | *ロールカテゴリID* |
| SalesAdmin | ManagementRole |
| Sales | ManagementRole |
| CustomerAdmin | ManagementRole |
| Customer | ManagementRole |
| Provisioning | ProductRole |
| TA00000001 | TenantRole |
| TA00000002 | TenantRole |
| … | … |

123

| リソースカテゴリテーブル | |
|---|---|
| *ID* | *説明* |
| UrlResource | URLリソース |
| DataResource | データリソース |
| … | … |

124

| リソースロールテーブル | |
|---|---|
| *リソースID* | *ロールID* |
| R00000001 | Sales |
| R00000001 | Customer |
| R00000002 | Provisioning |
| R00000003 | TA00000001 |
| R00000004 | TA00000002 |
| … | … |

125

| リソーステーブル | | | |
|---|---|---|---|
| *リソースID* | *リソースカテゴリID* | *保護アイテム* | *権限* |
| R00000001 | UrlResource | http://xxx.com/menu/* | Read |
| R00000002 | UrlResource | http://xxx.com/search/* | Read, Update |
| R00000003 | DataResource | TY00000001 | Create, Read, Update, Delete |
| R00000004 | DataResource | TY00000002 | Create, Read, Update, Delete |
| … | … | … | … |

【図１３】

132

| テナントテーブル | | |
|---|---|---|
| *テナントID* | *ユーザータイプID* | *テナントカテゴリID* |
| TA00000001 | TY00000001 | SalesTenant |
| TA00000002 | TY00000002 | CustomerTenant |
| … | … | … |

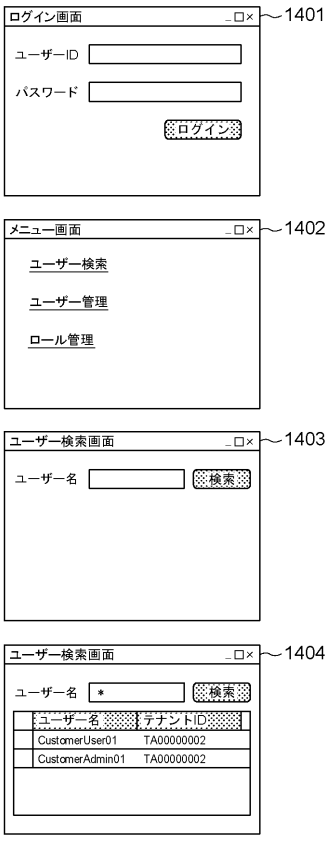
131

| テナントカテゴリテーブル | |
|---|---|
| *ID* | *説明* |
| SalesTenant | 販売テナント |
| CustomerTenant | 顧客テナント |
| Self | 自身 |
| … | … |

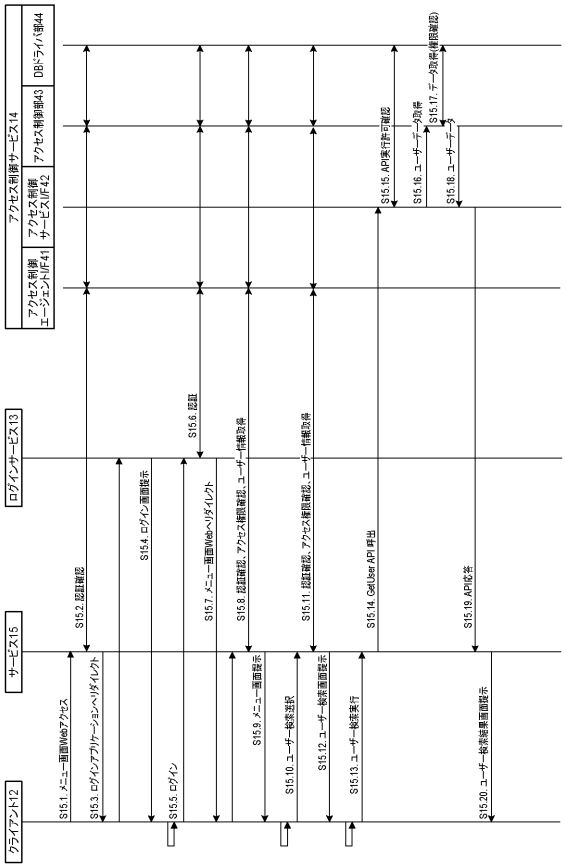
133

| API権限テーブル | | | |
|---|---|---|---|
| *ファンクションID* | *ロールID* | *属性テナントカテゴリID* | *属性テナントカテゴリID* |
| CreateTenant | Sales | SalesTenant | CustomerTenant |
| ChangeRole | SalesAdmin | SalesTenant | SalesTenant |
| ChangeRole | CustomerAdmin | CustomerTenant | CustomerTenant |
| SearchUser | SalesAdmin | SalesTenant | SalesTenant |
| SearchUser | Sales | SalesTenant | Self |
| SearchUser | CustomerAdmin | CustomerTenant | CustomerTenant |
| SearchUser | Customer | CustomerTenant | Self |
| SearchUser | … | … | … |

【図１４】



【図１５】




【図１６】

161

| ロール操作可否 | | | |
|---|---|---|---|
| 操作ロールID | 被操作ロールカテゴリID | 被操作ロールID | 可否 |
| CustomerAdmin | ManagementRole | CustomerAdmin | Allow |
| CustomerAdmin | ManagementRole | Customer | Deny |
| CustomerAdmin | ProductRole | * | Allow |
| CustomerAdmin | TenantRole | * | Deny |
| Customer | * | * | Deny |

フロントページの続き

(56)参考文献 特開２００５－３０１６０２（ＪＰ，Ａ）
特開２０１１－０８６１７２（ＪＰ，Ａ）
特開２０００－２０７３６３（ＪＰ，Ａ）
大関覚 外，ＴＷＸ－２１を基盤としたビジネスＳａａＳ，日立評論，日立評論社，２００９年
7月 1日，第91巻，第7号，第20-25頁

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ ２１／６２