(19)日本国特許庁(JP)

(12)特許公報(B2)

(11)特許番号
特許第7030709号
(P7030709)

(45)発行日　令和4年3月7日(2022.3.7)

(24)登録日　令和4年2月25日(2022.2.25)

(51)国際特許分類
Ｇ０６Ｆ　21/31　(2013.01)

ＦＩ
Ｇ０６Ｆ　21/31

請求項の数　7　（全50頁）

(21)出願番号　特願2018-545465(P2018-545465)
(86)(22)出願日　平成29年8月21日(2017.8.21)
(65)公表番号　特表2019-532368(P2019-532368A)
(43)公表日　令和1年11月7日(2019.11.7)
(86)国際出願番号　PCT/US2017/047726
(87)国際公開番号　WO2018/044604
(87)国際公開日　平成30年3月8日(2018.3.8)
審査請求日　令和2年4月17日(2020.4.17)
(31)優先権主張番号　62/381,866
(32)優先日　平成28年8月31日(2016.8.31)
(33)優先権主張国・地域又は機関
米国(US)
(31)優先権主張番号　15/680,362
(32)優先日　平成29年8月18日(2017.8.18)
最終頁に続く

(73)特許権者　502303739
オラクル・インターナショナル・コーポレイション
アメリカ合衆国カリフォルニア州９４０６５レッドウッド・シティー，オラクル・パークウェイ５００
(74)代理人　110001195
特許業務法人深見特許事務所
(72)発明者　ウィルソン，グレッグ
アメリカ合衆国、７８７３１　テキサス州、オースティン、ウォルナット・クレイ・ドライブ、３９１７
(72)発明者　メダム，ベンカテスワラ・レディ
アメリカ合衆国、９５３５６　カリフォルニア州、モデスト、ノア・コート、４
最終頁に続く

(54)【発明の名称】　マルチテナントアイデンティティクラウドサービスのためのデータ管理

(57)【特許請求の範囲】
【請求項１】
クラウドベースのアイデンティティ管理を提供する方法であって、前記方法は、
１つ以上のプロセッサが、リソースを求める要求をウェブゲートがアプリケーションから受けるステップを含み、前記要求は、複数のリソースタイプのうちのあるリソースタイプに対するオペレーションを含み、前記要求は、複数のテナントのうちのあるテナントを指定し、前記リソースタイプは、スキーマ定義を含み、前記スキーマ定義は、複数の属性と、前記属性各々についてのメタデータとを含み、
前記１つ以上のプロセッサが、前記要求に基づいてマイクロサービスにアクセスするステップと、
前記１つ以上のプロセッサが、前記リソースタイプを解明するステップとを含み、前記解明するステップは、前記リソースタイプを判断し、対応するスキーマおよびスキーマ定義を取出し、前記スキーマに基づいて前記リソースタイプに必要なのはどの属性かを判断することを含み、
前記１つ以上のプロセッサが、前記判断された前記リソースタイプに必要な属性に基づいて前記オペレーションが前記リソースタイプによってサポートされていることを確認するステップと、
前記１つ以上のプロセッサが、前記テナントに関連するデータプロバイダを取得するステップと、
前記１つ以上のプロセッサが、前記データプロバイダに対して前記オペレーションの実行

を命じるステップと、
前記１つ以上のプロセッサが、前記リソースを返すステップとを含む、方法。
【請求項２】
前記リソースタイプの複数のバージョンのうちの、前記リソースタイプの少なくとも１つのバージョンは、前のバージョンに対し、非推奨の属性を示すタグを含み、前記リソースタイプの少なくとも１つのバージョンは、前のリソースタイプに対し、追加された属性を示すタグを含み、さらに、
前記１つ以上のプロセッサが、前記リソースタイプの対応するタグに基づき前記リソースタイプのバージョンを用いて前記オペレーションを実行するステップを含む、請求項１に記載の方法。
【請求項３】
前記オペレーションは、作成、更新、削除、取得、またはサーチのうちの１つを含む、請求項１または２に記載の方法。
【請求項４】
前記リソースタイプはユーザであり、前記対応するスキーマはパスワード状態を含む、請求項１に記載の方法。
【請求項５】
前記データプロバイダは、データベースプロバイダまたは軽量ディレクトリアクセスプロトコル（ＬＤＡＰ）プロバイダのうちの１つを含む、請求項１～４のいずれか１項に記載の方法。
【請求項６】
クラウドベースのアイデンティティおよびアクセス管理を提供するためのシステムであって、
複数のテナントと、
複数のマイクロサービスと、
１つ以上のプロセッサとを備え、前記１つ以上のプロセッサは、
リソースを求める要求をウェブゲートによってアプリケーションから受け、前記要求は、複数のリソースタイプのうちのあるリソースタイプに対するオペレーションを含み、前記要求は、複数のテナントのうちのあるテナントを指定し、前記リソースタイプは、スキーマ定義を含み、前記スキーマ定義は、複数の属性と、前記属性各々についてのメタデータとを含み、
前記１つ以上のプロセッサは、さらに、
前記要求に基づいて前記複数のマイクロサービスのうちのあるマイクロサービスにアクセスし、
前記リソースタイプを解明し、前記解明することは、前記リソースタイプを判断し、対応するスキーマおよびスキーマ定義を取出し、前記スキーマに基づいて前記リソースタイプに必要なのはどの属性かを判断することを含み、
前記１つ以上のプロセッサは、さらに、
前記判断された前記リソースタイプに必要な属性に基づいて前記オペレーションが前記リソースタイプによってサポートされていることを確認し、
前記テナントに関連するデータプロバイダを取得し、
前記データプロバイダに対して前記オペレーションの実行を命じ、
前記リソースを返す、システム。
【請求項７】
請求項１～５のいずれか１項に記載の方法をプロセッサに実行させるためのコンピュータ読取可能プログラム。
【発明の詳細な説明】
【技術分野】
【０００１】
関連出願の相互参照

本願は、２０１６年８月３１日出願の米国仮特許出願第６２／３８１，８６６号に基づく優先権を主張し、その開示を本明細書に引用により援用する。

【０００２】

分野

一実施形態は、概してアイデンティティ管理に関し、特にクラウドシステムにおけるアイデンティティ管理に関する。

【背景技術】

【０００３】

背景情報

一般的に、多様なデバイス（たとえばデスクトップおよびモバイルデバイス）および多様なユーザ（たとえば被雇用者、パートナー、顧客など）からアクセスされる、クラウドベースのアプリケーション（たとえば企業パブリッククラウドアプリケーション、第三者クラウドアプリケーションなど）の使用が、急激に増加している。クラウドベースのアプリケーションは、その多様性およびアクセシビリティが高いので、アイデンティティの管理およびアクセスのセキュリティが中心的な関心事になっている。クラウド環境における典型的なセキュリティの問題は、不正アクセス、アカウントのハイジャック、悪意のあるインサイダーなどである。したがって、クラウドベースのアプリケーションであっても、どこに存在するアプリケーションであっても、アプリケーションにアクセスするデバイスの種類またはユーザの種類にかかわらず、安全なアクセスが必要とされている。

【発明の概要】

【課題を解決するための手段】

【０００４】

実施形態は、クラウドベースのアイデンティティ管理を、リソースを求める要求をウェブゲートがアプリケーションから受けることにより、提供する。この要求は、複数のリソースタイプのうちのあるリソースタイプに対するオペレーションを含み、複数のテナントのうちのあるテナントを指定する。実施形態は、この要求に基づいてマイクロサービスにアクセスし、上記リソースタイプを解明し、メタデータに基づいて、上記オペレーションが上記リソースタイプによってサポートされていることを確認する。実施形態は、上記テナントに関連するデータプロバイダを取得し、このデータプロバイダに対して上記オペレーションの実行を命じ、その後リソースを返す。

【図面の簡単な説明】

【０００５】

【図１】クラウドベースのアイデンティティ管理を提供する実施形態の一例のブロック図である。

【図２】クラウドベースのアイデンティティ管理を提供する実施形態の一例のブロック図である。

【図３】クラウドベースのアイデンティティ管理を提供する実施形態の一例のブロック図である。

【図４】クラウドベースのアイデンティティ管理を提供する実施形態の一例のブロック図である。

【図５】クラウドベースのアイデンティティ管理を提供する実施形態の一例のブロック図である。

【図６】ある実施形態のシステムビューを提供するブロック図である。

【図６Ａ】ある実施形態の機能ビューを提供するブロック図である。

【図７】クラウドゲートを実現する実施形態のブロック図である。

【図８】一実施形態における複数のテナンシーを実現するシステムの一例を示す。

【図９】ある実施形態のネットワークビューのブロック図である。

【図１０】一実施形態におけるシングル・サイン・オン（single sign on：「ＳＳＯ」）機能のシステムアーキテクチャビューのブロック図である。

【図１１】一実施形態におけるＳＳＯ機能のメッセージシーケンスフローを示す図である。

【図１２】一実施形態における分散型データグリッドの一例を示す。
【図１３】一実施形態に係るアイデンティティクラウドサービス（「ＩＤＣＳ」）またはサービスとしてのアイデンティティ（「ＩＤａａＳ」）のためのデータマネージャアーキテクチャを示す。
【図１４】図１３のリソースデータマネージャによって実現される本発明の実施形態の機能フローを示す。
【図１５】一実施形態に係る自動スキーマバージョニングを示す。
【発明を実施するための形態】
【０００６】
詳細な説明
実施形態は、リソースタイプとそれに関連するスキーマとを定義するメタデータを実現する。マルチテナントシステムにおいてリソースに対するオペレーションを実行することを求める要求をメタデータを用いて解決して当該オペレーションを実行するテナントに関連するデータプロバイダを決定する。
【０００７】
実施形態が提供するアイデンティティクラウドサービスは、マイクロサービスベースのアーキテクチャを実現するとともに、マルチテナントアイデンティティおよびデータセキュリティの管理ならびにクラウドベースのアプリケーションへの安全なアクセスを提供する。実施形態は、ハイブリッドクラウドのデプロイメント（すなわちパブリッククラウドとプライベートクラウドとを組合わせたものを含むクラウドのデプロイメント）について安全なアクセスをサポートする。実施形態は、クラウド内およびオンプレミス双方におけるアプリケーションおよびデータを保護する。実施形態は、ウェブ、モバイル機器、およびアプリケーションプログラミングインターフェイス（application programming interface：「ＡＰＩ」）を介したマルチチャネルアクセスをサポートする。実施形態は、顧客、パートナー、および被雇用者など、さまざまなユーザのアクセスを管理する。実施形態は、クラウドを通じたアクセスおよびオンプレミスのアクセス双方を管理、制御、および監査する。実施形態は、新たなおよび既存のアプリケーションおよびアイデンティと統合される。実施形態は横方向にスケーラブルである。
【０００８】
一実施形態は、ステートレスな中間層環境において多数のマイクロサービスを実現することによりクラウドベースのマルチテナントアイデンティティおよびアクセス管理サービスを提供するシステムである。一実施形態において、要求された各アイデンティティ管理サービスは、リアルタイムタスクとニア・リアルタイムタスクとに分割される。リアルタイムタスクは中間層のマイクロサービスによって処理されるのに対し、ニア・リアルタイムタスクはメッセージキューにオフロードされる。実施形態は、ルーティング層および中間層によって消費されるアクセストークンを実現することにより、マイクロサービスにアクセスするためのセキュリティモデルを強化する。したがって、実施形態は、マルチテナントのマイクロサービスアーキテクチャに基づいてクラウドスケールのアイデンティティおよびアクセス管理（Identity and Access Management（「ＩＡＭ」）プラットフォームを提供する。
【０００９】
一実施形態は、組織が、その新たなビジネス構想のために高速で信頼性が高くかつ安全なサービスを迅速に開発できるようにするアイデンティティクラウドサービスを提供する。一実施形態において、アイデンティティクラウドサービスは多数のコアサービスを提供する。各コアサービスは、多くの企業が直面する固有の課題を解決する。一実施形態において、アイデンティティクラウドサービスは、たとえば、最初にユーザのオンボード／インポートを行なうとき、ユーザメンバとともにグループをインポートするとき、ユーザを作成／更新／ディスエーブル／イネーブル／削除するとき、ユーザをグループに割当てる／グループへのユーザ割当てを解除するとき、グループを作成／更新／削除するとき、パスワードをリセットするとき、ポリシーを管理するとき、アクティベーションを送信すると

きなどの、アドミニストレータをサポートする。
【００１０】
統一されたアクセスセキュリティ
一実施形態は、クラウド環境およびオンプレミス環境双方におけるアプリケーションおよびデータを保護する。本実施形態は、どのデバイスからの誰によるどのアプリケーションへのアクセスも安全にする。本実施形態は、これらの環境双方にわたる保護を提供する。なぜなら、これら２つの環境の間でセキュリティに矛盾があればリスクが高くなる可能性があるからである。たとえば、このような矛盾があった場合、販売員は、離反して競合他社に移った後であっても、その顧客関係管理（Customer Relationship Management：「ＣＲＭ」）アカウントへのアクセス権を有し続ける場合がある。したがって、実施形態は、オンプレミス環境においてプロビジョニングされたセキュリティ制御をクラウド環境に拡張する。たとえば、ある人物が会社を辞めた場合、実施形態は、そのアカウントがオンプレミスおよびクラウド双方においてディスエーブルされることを保証する。
【００１１】
一般的に、ユーザは、ウェブブラウザ、デスクトップ、携帯電話、タブレット、スマートウォッチ、その他のウェアラブル機器などの多種多様なチャネルを通してアプリケーションおよび／またはデータにアクセスし得る。したがって、一実施形態は、これらすべてのチャネルについて、これらを通るアクセスを安全なものにする。たとえば、ユーザは、その携帯電話を用いて、自身のデスクトップ上で開始したトランザクションを完了させることができる。
【００１２】
一実施形態はさらに、顧客、パートナー、被雇用者など、さまざまなユーザのアクセスを管理する。一般的に、アプリケーションおよび／またはデータは、被雇用者だけでなく、顧客または第三者によってもアクセスされる場合がある。既知の多くのシステムは、被雇用者のオンボード時に安全対策を講じるが、この安全対策は通常、顧客、第三者、パートナーなどにアクセス権を付与するときの安全対策と同じレベルではないので、結果として、適切に管理されていない者によってセキュリティが破られる可能性がある。しかしながら、実施形態は、被雇用者だけでなく各タイプのユーザのアクセスについて十分な安全対策が提供されることを保証する。
【００１３】
アイデンティティクラウドサービス
実施形態は、マルチテナントでクラウドスケールのＩＡＭプラットフォームであるアイデンティティクラウドサービス(Identity Cloud Service：「ＩＤＣＳ」）を提供する。ＩＤＣＳは、認証、認可、監査、および連携（federation）を提供する。ＩＤＣＳは、パブリッククラウドおよびオンプレミスシステム上で実行されているカスタムアプリケーションおよびサービスへのアクセスを管理する。これに代わるまたはこれに加えられる実施形態において、ＩＤＣＳは、パブリッククラウドサービスへのアクセスも管理し得る。たとえば、ＩＤＣＳを用いて、このような多様なサービス／アプリケーション／システムにわたってシングル・サイン・オン（「ＳＳＯ」）機能を提供することができる。
【００１４】
実施形態は、クラウドスケールのソフトウェアサービスを設計、構築、および配信するためのマルチテナントマイクロサービスアーキテクチャに基づく。マルチテナンシーとは、あるサービスを物理的に実現したものがありこのサービスが当該サービスを購入した複数の顧客を安全にサポートするサービスであることを言う。サービスは、異なるクライアントが異なる目的のために再使用できるソフトウェア機能またはソフトウェア機能のセット（指定された情報を取出すことまたは一組の動作を実行することなど）に、（たとえばサービスを要求しているクライアントのアイデンティティに基づく）その使用を管理するポリシーを合わせたものである。一実施形態において、サービスは、１つ以上の機能へのアクセスを可能にするメカニズムであり、このアクセスは、所定のインターフェイスを用いて提供され、サービスの記述によって明記された制約およびポリシーに従って実行される。

【００１５】
一実施形態において、マイクロサービスは独立してデプロイ可能なサービスである。一実施形態において、マイクロサービスという用語は、言語に依存しないＡＰＩを用いて相互に通信する小さな独立したプロセスから複雑なアプリケーションが構成されている、ソフトウェアアーキテクチャ設計パターンを意図している。一実施形態において、マイクロサービスは、細かく分離された小さなサービスであり、各サービスは、小さなタスクの実行に集中し得る。一実施形態において、マイクロサービスアーキテクチャスタイルは、単一のアプリケーションを小さなサービス一式として開発する手法であり、各サービスは、自身のプロセスにおいて実行され、軽量のメカニズム（たとえばＨＴＴＰリソースＡＰＩ）と通信する。一実施形態において、マイクロサービスは、同一機能すべてをまたは同一機能のうちの多くを実行するモノリシックサービスと比較すると、交換がより簡単である。加えて、マイクロサービスは各々、その他のマイクロサービスに悪影響を与えることなく更新し得る。これに対し、モノリシックサービスの一部を更新すると、当該モノリシックサービスの他の部分に望ましくないまたは意図せぬ悪影響が及ぶ可能性がある。一実施形態において、マイクロサービスはその機能を中心として有益に編成し得る。一実施形態において、マイクロサービスのコレクションのうち各マイクロサービスのスタートアップ時間は、これらのマイクロサービスのうちのすべてのサービスをまとめて実行する単一のアプリケーションのスタートアップ時間よりも遥かに短い。いくつかの実施形態において、このようなマイクロサービス各々のスタートアップ時間は約１秒以下であるのに対し、このような単一のアプリケーションのスタートアップ時間は約１分、数分、またはそれよりも長い場合がある。

【００１６】
一実施形態において、マイクロサービスアーキテクチャとは、フレキシブルで、独立してデプロイ可能なソフトウェアシステムを構築するための、サービス指向アーキテクチャ（service oriented architecture（「ＳＯＡ」））の専門化（すなわちシステム内におけるタスクの分離）および実現の手法のことである。マイクロサービスアーキテクチャにおけるサービスは、目的を達成するためにネットワークを通して相互に通信するプロセスである。一実施形態において、これらのサービスは、技術に依存しないプロトコルを使用する。一実施形態において、サービスは、細分性が小さく軽量であるプロトコルを使用する。一実施形態において、サービスは独立してデプロイ可能である。システムの機能を異なる小さなサービスに分散させることにより、システムの結束性は向上し、システムのカップリングは減少する。それにより、システム変更が容易になり、任意の時点でシステムに機能および品質を追加することが容易になる。また、それによって、個々のサービスのアーキテクチャが、絶え間ないリファクタリングを通して出現することが可能になり、したがって、大規模な事前の設計の必要性は低下しソフトウェアを早期に連続してリリースすることが可能になる。

【００１７】
一実施形態において、マイクロサービスアーキテクチャでは、アプリケーションがサービスのコレクションとして開発され、各サービスはそれぞれのプロセスを実行し軽量のプロトコルを用いて通信する（たとえばマイクロサービスごとの固有ＡＰＩ）。マイクロサービスアーキテクチャにおいて、１つのソフトウェアを個々のサービス／機能に分解することは、提供するサービスに応じて異なるレベルの粒度で行なうことができる。サービスはランタイムコンポーネント／プロセスである。各マイクロサービスは、他のモジュール／マイクロサービスに対してトークすることができる内蔵モジュールである。各マイクロサービスは、他からコンタクトできる無名ユニバーサルポートを有する。一実施形態において、マイクロサービスの無名ユニバーサルポートは、従来マイクロサービスがエクスポーズする標準通信チャネルであり（たとえば従来のハイパーテキスト転送プロトコル（「ＨＴＴＰ」）ポートのような）、同一サービス内の他のモジュール／マイクロサービスがそれに対してトークできるようにする標準通信チャネルである。マイクロサービスまたはその他の内蔵機能モジュールを包括的に「サービス」と呼ぶことができる。

【００１８】
実施形態は、マルチテナントアイデンティティ管理サービスを提供する。実施形態は、さまざまなアプリケーションとの容易な統合を保証するオープン標準に基づいており、標準ベースのサービスを通してＩＡＭ機能を提供する。
【００１９】
実施形態は、アイデンティティがアクセスできる対象、このようなアクセスを付与できる者、このようなアクセスを管理できる者などを判断し施行することを伴うユーザアイデンティティのライフサイクルを管理する。実施形態は、クラウド内でアイデンティティ管理ワークロードを実行し、このクラウド内に存在するとは限らないアプリケーションのセキュリティ機能をサポートする。これらの実施形態が提供するアイデンティティ管理サービスはクラウドから購入されてもよい。たとえば、企業は、このようなサービスをクラウドから購入してその被雇用者の当該企業のアプリケーションに対するアクセスを管理してもよい。
【００２０】
実施形態は、システムセキュリティ、大規模なスケーラビリティ、エンドユーザのユーザビリティ、およびアプリケーションのインターオペラビリティを提供する。実施形態は、クラウドの成長と、顧客によるアイデンティティサービスの使用とを扱っている。マイクロサービスに基づく基礎は、横方向のスケーラビリティ条件を扱うのに対し、サービスの綿密な調整は機能条件を扱う。これらの目標双方を達成するには、ビジネスロジックを（可能な限り）分解することにより、最終的には一貫性のあるステートレスを達成する一方で、リアルタイム処理を受けない動作論理のほとんどが、配信と処理が保証されたスケーラビリティが高い非同期イベント管理システムに、オフロードされることにより、ニア・リアルタイムにシフトする。実施形態は、コスト効率を実現しシステム管理を容易にするために、ウェブ層からデータまで完全にマルチテナントである。
【００２１】
実施形態は、さまざまなアプリケーションと統合し易くするために、業界の標準（たとえば、ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ、ＯＡｕｔｈ２、セキュリティ・アサーション・マークアップ言語（Security Assertion Markup Language）２（「ＳＡＭＬ２」)、クロスドメインアイデンティティ管理用システム（System for Cross-domain Identity Management：「ＳＣＩＭ」）、レプレゼンテーショナル・ステート・トランスファー(Representational State Transfer：「ＲＥＳＴ」)など）に従う。一実施形態は、クラウドスケールＡＰＩプラットフォームを提供し、エラスティックスケーラビリティのために横方向にスケーラブルなマイクロサービスを実現する。本実施形態は、クラウド原理を強化し、テナントごとにデータを分離したマルチテナントアーキテクチャを提供する。本実施形態はさらに、テナントセルフサービスを介してテナントごとのカスタマイズを提供する。本実施形態は、他のアイデンティティサービスとのオンデマンドの統合の際にはＡＰＩを介して利用することができ、連続したフィーチャーリリースを提供する。
【００２２】
一実施形態は、インターオペラビリティを提供し、クラウドおよびオンプレミスにおけるアイデンティティ管理（identity management：「ＩＤＭ」）機能への投資を強化する。本実施形態は、オンプレミスの軽量ディレクトリアクセスプロトコル（Lightweight Directory Access Protocol：「ＬＤＡＰ」）データからクラウドデータへの、およびその逆の、自動化されたアイデンティティ同期化を提供する。本実施形態は、クラウドと企業との間にＳＣＩＭアイデンティティバスを提供し、ハイブリッドクラウドのデプロイの各種オプションを可能にする（たとえば、アイデンティティ連携および／または同期化、ＳＳＯエージェント、ユーザプロビジョニングコネクタなど）。
【００２３】
したがって、一実施形態は、ステートレスな中間層において多数のマイクロサービスを実現することによりクラウドベースのマルチテナントアイデンティティおよびアクセス管理サービスを提供するシステムである。一実施形態において、要求された各アイデンティテ

ィ管理サービスは、リアルタイムタスクとニア・リアルタイムタスクとに分割される。リアルタイムタスクは中間層のマイクロサービスによって処理されるのに対し、ニア・リアルタイムタスクはメッセージキューにオフロードされる。実施形態は、ルーティング層によって消費されて、マイクロサービスにアクセスするためのセキュリティモデルを実施するトークンを実現する。したがって、実施形態は、マルチテナントのマイクロサービスアーキテクチャに基づくクラウドスケールのＩＡＭプラットフォームを提供する。

【００２４】

一般的に、周知のシステムは、たとえば、企業クラウドアプリケーション、パートナークラウドアプリケーション、第三者クラウドアプリケーション、および顧客アプリケーションなど、各種環境によって提供されるアプリケーションに対するサイロ化されたアクセスを提供する。このようなサイロ化されたアクセスは、複数のパスワード、異なるパスワードポリシー、異なるアカウントプロビジョニングおよびデプロビジョニング手法、異種の監査などを必要とする場合がある。しかしながら、一実施形態は、ＩＤＣＳを実現することにより、このようなアプリケーションに対し統一されたＩＡＭ機能を提供する。図１は、ユーザおよびアプリケーションをオンボードするための統一されたアイデンティティプラットフォーム１２６を提供する、ＩＤＣＳ１１８を用いる実施形態の一例のブロック図１００である。本実施形態は、企業クラウドアプリケーション１０２、パートナークラウドアプリケーション１０４、第三者クラウドアプリケーション１１０、および顧客アプリケーション１１２などのさまざまなアプリケーションにまたがるシームレスなユーザ体験を提供する。アプリケーション１０２、１０４、１１０、１１２は、異なるチャネルを通してアクセスされてもよく、たとえば、携帯電話ユーザ１０８が携帯電話１０６を介して、デスクトップコンピュータのユーザ１１６がブラウザ１１４を介して、アクセスしてもよい。ウェブブラウザ（一般的にブラウザと呼ばれる）は、ワールドワイドウェブ上で情報リソースを取得、提示、およびトラバースするためのソフトウェアアプリケーションである。ウェブブラウザの例としては、Mozilla（登録商標） Firefox（登録商標）、Google Chrome（登録商標）、Microsoft（登録商標） Internet Explorer（登録商標）、およびApple（登録商標） Safari（登録商標）が挙げられる。

【００２５】

ＩＤＣＳ１１８は、ユーザのアプリケーションの統一されたビュー１２４、（アイデンティティプラットフォーム１２６を介する）デバイスおよびアプリケーションにまたがる統一された安全なクレデンシャル、および（管理コンソール１２２を介する）統一された管理方法を、提供する。ＩＤＣＳサービスは、ＩＤＣＳ　ＡＰＩ１４２にコールすることによって取得されてもよい。このようなサービスは、たとえば、ログイン／ＳＳＯサービス１２８（たとえばＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ）、連携サービス１３０（たとえばＳＡＭＬ）、トークンサービス１３２（たとえばＯＡｕｔｈ）、ディレクトリサービス１３４（たとえばＳＣＩＭ）、プロビジョニングサービス１３６（たとえばＳＣＩＭまたはAny Transport over Multiprotocol（「ＡＴｏＭ」））、イベントサービス１３８（たとえばＲＥＳＴ）、およびロールベースアクセス制御（role-based access control：「ＲＢＡＣ」）サービス１４０（たとえばＳＣＩＭ）を含み得る。ＩＤＣＳ１１８はさらに、提供されるサービスに関するレポートおよびダッシュボード１２０を提供し得る。

【００２６】

<u>統合ツール</u>

通常、大企業では、そのオンプレミスのアプリケーションへの安全なアクセスのために、ＩＡＭシステムを適所に設けるのが一般的である。ビジネス手法は通常オラクル社の「Oracle IAM Suite」などのインハウスＩＡＭシステムを中心として成熟し標準化される。小～中規模組織でも、通常は、そのビジネスプロセスを、Microsoft Active Directory（「ＡＤ」）などの単純なディレクトリソリューションを通してユーザアクセスを管理することを中心として設計されている。オンプレミス統合を可能にするために、実施形態は、顧客がそのアプリケーションをＩＤＣＳと統合できるようにするツールを提供する。

【００２７】

図２は、オンプレミス２０６のＡＤ２０４との統合を提供する、クラウド環境２０８内のＩＤＣＳ２０２を用いる実施形態の一例のブロック図２００である。本実施形態は、たとえば、クラウドサービス２１０、クラウドアプリケーション２１２、パートナーアプリケーション２１４、および顧客アプリケーション２１６などのクラウド２０８内のさまざまなアプリケーション／サービスならびにオンプレミスアプリケーション２１８などのオンプレミスアプリケーションおよび第三者アプリケーションを含むすべてのアプリケーションにまたがる、シームレスなユーザ体験を提供する。クラウドアプリケーション２１２は、たとえば、ヒューマン・キャピタル・マネージメント（Human Capital Management：「ＨＣＭ」）、ＣＲＭ、タレント取得（たとえばオラクル社のOracle Taleoクラウドサービス）、構成、価格設定、および見積もり（Configure Price and Quote「ＣＰＱ」）などを含み得る。クラウドサービス２１０は、たとえば、サービスとしてのプラットフォーム（Platform as a Service：「ＰａａＳ」）、Ｊａｖａ（登録商標）、データベース、ビジネスインテリジェンス（business intelligence：「ＢＩ」）、文書などを含み得る。

【００２８】

アプリケーション２１０、２１２、２１４、２１６、２１８は、異なるチャネルを通してアクセスされてもよく、たとえば、携帯電話ユーザ２２０が携帯電話２２２を介して、デスクトップコンピュータのユーザ２２４がブラウザ２２６を介して、アクセスしてもよい。本実施形態は、クラウド２０８と企業２０６との間のＳＣＩＭアイデンティティバス２３４を介して、オンプレミスのＡＤデータからクラウドデータに、アイデンティティの同期化を自動的に行なう。本実施形態はさらに、クラウド２０８からオンプレミスＡＤ２０４への、（たとえばパスワード２３２を用いて）認証を連携させるためのＳＡＭＬバス２２８を提供する。

【００２９】

一般的に、アイデンティティバスは、アイデンティティ関連サービスのためのサービスバスである。サービスバスは、メッセージをあるシステムから別のシステムに伝えるためのプラットフォームを提供する。これは、たとえばサービス指向アーキテクチャ（service oriented architecture：「ＳＯＡ」）において、信頼されているシステム間で情報を交換するための制御されたメカニズムである。アイデンティティバスは、ウェブサービス、ウェブサーバプロキシなどの標準的なＨＴＴＰベースのメカニズムに従って構築された論理バスである。アイデンティティバスにおける通信は、各プロトコル（たとえばＳＣＩＭ、ＳＡＭＬ、ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔなど）に従って実行されてもよい。たとえば、ＳＡＭＬバスは、ＳＡＭＬサービスに関するメッセージを伝えるための、２つのシステム間のＨＴＴＰベースの接続である。同様に、ＳＣＩＭバスを用い、ＳＣＩＭプロトコルに従って、ＳＣＩＭメッセージを伝える。

【００３０】

図２の実施形態は、顧客のＡＤ２０４とともにオンプレミス２０６でダウンロードおよびインストールすることができる小バイナリ（たとえば大きさが１ＭＢ）のアイデンティティ（「ＩＤ」）ブリッジ２３０を実現する。ＩＤブリッジ２３０は、顧客によって選択された組織ユニット（organizational unit：「ＯＵ」）のユーザおよびグループ（たとえばユーザのグループ）をリッスンし、これらのユーザをクラウド２０８に対して同期させる。一実施形態において、ユーザのパスワード２３２はクラウド２０８に対して同期されていない。顧客は、ＩＤＣＳユーザのグループを、ＩＤＣＳ２０８において管理されているクラウドアプリケーションにマッピングすることにより、ユーザのアプリケーションアクセスを管理することができる。ユーザのグループメンバーシップがオンプレミス２０６で変更されるたびに、対応するクラウドアプリケーションアクセスは自動的に変更される。

【００３１】

たとえば、技術部門から販売部門に異動した被雇用者は、販売クラウドへのアクセスをほぼ瞬間的に取得することができ、開発者クラウドへのアクセスは失う。この変化がオンプレミスＡＤ２０４に反映されると、クラウドアプリケーションのアクセスの変更がニア・

リアルタイムで実現される。同様に、ＩＤＣＳ２０８で管理されているクラウドアプリケーションへの、この企業から去るユーザのアクセスは、取消される。完全自動化のために、顧客は、たとえばＡＤ連携サービス（「ＡＤ／ＦＳ」またはＳＡＭＬ連携を実現するその他の何らかのメカニズム）を通して、オンプレミスＡＤ２０４とＩＤＣＳ２０８との間のＳＳＯをセットアップして、エンドユーザが、単一の企業パスワード３３２を用いて、クラウドアプリケーション２１０、２１２、２１４、２１６およびオンプレミスアプリケーション２１８にアクセスできるようにしてもよい。

【００３２】

図３は、図２と同一のコンポーネント２０２、２０６、２０８、２１０、２１２、２１４、２１６、２１８、２２０、２２２、２２４、２２６、２２８、２３４を含む実施形態の一例のブロック図３００である。しかしながら、図３の実施形態において、ＩＤＣＳ２０２は、オラクルＩＤＭのようなオンプレミスＩＤＭ３０４との統合を提供する。オラクルＩＤＭ３０４は、ＩＡＭ機能を提供するための、オラクル社のソフトウェアスイートである。本実施形態は、オンプレミスアプリケーションおよび第三者アプリケーションを含むすべてのアプリケーションにまたがるシームレスなユーザ体験を提供する。本実施形態は、クラウド２０２と企業２０６との間のＳＣＩＭアイデンティティバス２３４を介したオンプレミスＩＤＭ３０４からＩＤＣＳ２０８へのユーザアイデンティティをプロビジョニングする。本実施形態はさらに、クラウド２０８からオンプレミス２０６への認証の連携のためのＳＡＭＬバス２２８（またはＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔバス）を提供する。

【００３３】

図３の実施形態において、オラクル社のオラクルアイデンティティマネージャ（Oracle Identity Manager：「ＯＩＭ」）コネクタ３０２およびオラクル社のオラクルアクセスマネージャ（Oracle Access Manager：「ＯＡＭ」）連携モジュール３０６は、オラクルＩＤＭ３０４の拡張モジュールとして実現される。コネクタは、システムに話しかける方法について物理的な認識があるモジュールである。ＯＩＭは、ユーザアイデンティティを管理するように構成されたアプリケーションである（たとえば、ユーザがアクセス権を持つべき対象とアクセス権を持つべきでない対象に基づいて異なるシステムのユーザアカウントを管理する）。ＯＡＭは、ウェブＳＳＯ、アイデンティコンテキスト、認証および認可、ポリシー管理、テスト、ロギング、監査などのアクセス管理機能を提供するセキュリティアプリケーションである。ＯＡＭはＳＡＭＬに対するビルトイン（built-in）サポートを有する。ユーザがＩＤＣＳ２０２のアカウントを有する場合、ＯＩＭコネクタ３０２およびＯＡＭ連携３０６をオラクルＩＤＭ３０４とともに使用することにより、このアカウントを作成／削除し、このアカントからのアクセスを管理することができる。

【００３４】

図４は、図２および図３と同一のコンポーネント２０２、２０６、２０８、２１０、２１２、２１４、２１６、２１８、２２０、２２２、２２４、２２６、２３４を含む実施形態の一例のブロック図４００である。しかしながら、図４の実施形態において、ＩＤＣＳ２０２は、クラウドアイデンティをオンプレミスアプリケーション２１８に拡張するための機能を提供する。本実施形態は、オンプレミスアプリケーションおよび第三者アプリケーションを含むすべてのアプリケーションにまたがるアイデンティティのシームレスなビューを提供する。図４の実施形態において、ＳＣＩＭアイデンティティバス２３４を用いることにより、ＩＤＣＳ２０２のデータを「クラウドキャッシュ」４０２と呼ばれるオンプレミスＬＤＡＰデータと同期させる。クラウドキャッシュ４０２は以下でより詳細に開示される。

【００３５】

一般的に、ＬＤＡＰに基づいて通信するように構成されたアプリケーションは、ＬＤＡＰ接続を必要とする。このようなアプリケーションはＬＤＡＰ接続をＵＲＬを用いて構築しないかもしれない（たとえばGoogle（登録商標）に接続する「www.google.com」とは違って）。なぜなら、ＬＤＡＰはローカルネットワーク上になければならないからである。図４の実施形態において、ＬＤＡＰベースのアプリケーション２１８は、クラウドキャ

ッシュ４０２に接続し、クラウドキャッシュ４０２は、ＩＤＣＳ２０２に接続してから、要求されているデータをＩＤＣＳ２０２から引出す。ＩＤＣＳ２０２とクラウドキャッシュ４０２との間の通信は、ＳＣＩＭプロトコルに従って実現されてもよい。たとえば、クラウドキャッシュ４０２はＳＣＩＭバス２３４を用いてＳＣＩＭ要求をＩＤＣＳ２０２に送信し、それに対応するデータを受信してもよい。

【００３６】

一般的に、あるアプリケーションの完全な実現は、コンシューマポータルを構築することと、外部ユーザ集団に対してマーケティングキャンペーンを実行することと、ウェブおよびモバイルチャネルをサポートすることと、ユーザ認証、セッション、ユーザプロファイル、ユーザグループ、アプリケーションロール、パスワードポリシー、セルフサービス／登録、社会的統合、アイデンティ連携などを処理することとを含む。一般的に、アプリケーションの開発者はアイデンティティ／セキュリティの専門家ではない。このため、オンデマンドのアイデンティティ管理サービスが望ましいのである。

【００３７】

図５は、図２～図４と同一のコンポーネント２０２、２２０、２２２、２２４、２２６、２３４、４０２を含む実施形態の一例のブロック図５００である。しかしながら、図５の実施形態において、ＩＤＣＳ２０２は、オンデマンドで安全なアイデンティティ管理を提供する。本実施形態は、オンデマンドの、ＩＤＣＳ２０２のアイデンティサービスとの統合を提供する（たとえばＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ、ＯＡｕｔｈ２、ＳＡＭＬ２、またはＳＣＩＭなどの標準に基づいて）。（オンプレミスであってもパブリッククラウド内またはプライベートクラウド内にあってもよい）アプリケーション５０５は、ＩＤＣＳ２０２のアイデンティティサービスＡＰＩ５０４をコールしてもよい。ＩＤＣＳ２０２が提供するサービスは、たとえば、セルフサービス登録５０６、パスワード管理５０８、ユーザプロファイル管理５１０、ユーザ認証５１２、トークン管理５１４、社会的統合５１６などを含み得る。

【００３８】

本実施形態において、ＳＣＩＭアイデンティティバス２３４を用いることにより、ＩＤＣＳ２０２内のデータを、オンプレミスのＬＤＡＰクラウドキャッシュ４０２内のデータと同期させる。さらに、ウェブサーバ／プロキシ（たとえばＮＧＩＮＸ、Apache等）上で実行している「クラウドゲート」５０２を、アプリケーション５０５が用いて、ＩＤＣＳ２０２からユーザウェブＳＳＯおよびＲＥＳＴ　ＡＰＩセキュリティを取得してもよい。クラウドゲート５０２は、クライアントアプリケーションが有効なアクセストークンを提供すること、および／またはユーザがＳＳＯセッション構築のために正常に認証することを保証することによって、マルチテナントＩＤＣＳマイクロサービスへのアクセスを安全なものするコンポーネントである。クラウドゲート５０２は以下でさらに開示される。クラウドゲート５０２（webgate／webagentと同様の実施ポイント）は、サポートされているウェブサーバの背後で実行されているアプリケーションがＳＳＯに参加することを可能にする。

【００３９】

一実施形態は、ＳＳＯおよびクラウドＳＳＯ機能を提供する。多くの組織において、オンプレミスＩＡＭおよびＩＤＣＳいずれにおいても一般的なエントリポイントはＳＳＯである。クラウドＳＳＯは、ユーザが、一回のユーザサイン・インで複数のクラウドリソースにアクセスできるようにする。組織はそのオンプレミスアイデンティティの連携を希望することが多い。したがって、実施形態は、オープン標準を利用することで、既存のＳＳＯとの統合を実現することにより、投資の節約と拡大を可能にする（たとえば、アイデンティティクラウドサービス手法への最終的な完全移行まで）。

【００４０】

一実施形態は以下の機能を提供し得る。

・アイデンティティストアを維持することにより、既に認可されているユーザアカウント、所有権、アクセス、および許可を追跡する。

・ワークフローとの統合により、アプリケーションのアクセスに必要なさまざまな承認（たとえば管理、ＩＴ、人的資源、法律、およびコンプライアンス）を簡単にする。
・選択的装置（たとえばモバイルおよびパーソナルコンピュータ（「ＰＣ」））に対するＳａａＳユーザアカウントをプロビジョニングする。ユーザポータルへのアクセスは、多数のプライベートおよびパブリッククラウドリソースを含む。
・規則および現在の職責へのコンプライアンスのための定期的な管理立証を容易にする。
【００４１】
これらの機能に加えて、実施形態はさらに、
・クラウドアプリケーションにおけるアカウントライフサイクルの管理のためのクラウドアカウントのプロビジョニング、
・よりロバストなマルチファクタ認証（multifactor authentication：「ＭＦＡ」）の統合、
・拡張モバイルセキュリティ機能、および
・動的認証オプション
を提供し得る。
【００４２】
一実施形態は、適応認証およびＭＦＡを提供する。一般的に、パスワードおよび確認のための質問は、不十分でありフィッシングなどのよくある攻撃に晒され易いとみなされてきた。現代の大半の企業体は、リスクを下げるために何らかの形態のＭＦＡに注目している。しかしながら、ソリューションが首尾よくデプロイされるためには、ソリューションをエンドユーザが簡単にプロビジョニング、維持、および理解する必要がある。なぜなら、エンドユーザは通常、そのデジタル体験を妨害するものに対し、それが何であろうと抵抗するからである。企業は、ＭＦＡを、シームレスなユーザアクセス体験のほぼトランスペアレントなコンポーネントにしつつ、私物の業務利用（bring your own device：「ＢＹＯＤ」）、社会的アイデンティティ、遠隔ユーザ、顧客、および契約者を安全に組込む方法を探している。ＭＦＡのデプロイにおいて、ＯＡｕｔｈおよびＯｐｅｎＩＤ Ｃｏｎｎｅｃｔなどの産業標準は、既存のマルチファクタソリューションの統合と、より新しい適応認証技術の導入とを保証するのに不可欠である。したがって、実施形態は、動的（または適応）認証を、利用できる情報（すなわちＩＰアドレス、場所、時刻、およびバイオメトリクス）の評価として定義することにより、ユーザセッション開始後のアイデンティを証明する。適切な標準（たとえばオープン認証（open authentication：「ＯＡＴＨ」）および高速オンライン認証（fast identity online：「ＦＩＤＯ」）の統合と、拡張可能なアイデンティティ管理フレームワークとを用いて、実施形態は、エンド・ツー・エンドの安全なＩＡＭデプロイの一部としてＩＴ組織内で簡単に採用、アップグレード、および統合できるＭＦＡソリューションを提供する。ＭＦＡおよび適応ポリシーを検討する場合、組織は、ハイブリッドのＩＤＣＳおよびオンプレミスＩＡＭ環境においてシステム間の統合を必要とするオンプレミスリソースおよびクラウドリソースにわたって一貫したポリシーを実現しなければならない。
【００４３】
一実施形態は、ユーザプロビジョニングおよび証明を提供する。一般的に、ＩＡＭソリューションの基本機能は、ユーザプロビジョニングライフサイクル全体を可能にしかつサポートすることである。これは、ユーザに対し、組織内におけるそのアイデンティティおよびロール（role）に適したアプリケーションアクセスを与えること（たとえば、ユーザのロールまたはそのロールの中で使用されるタスクもしくはアプリケーションは時間の経過に伴って変化するので）と、ユーザが組織から脱退するときに必要な、素早いユーザデプロビジョニングとを含む。これは、さまざまなコンプライアンス条件を満たすために重要であるだけでなく、不適切なインサイダーアクセスがセキュリティ侵害および攻撃の主要な原因であるので、重要である。アイデンティティクラウドソリューションにおける、自動化されたユーザプロビジョニング機能は、それ自身の権利において重要になり得るだけでなく、ハイブリッドＩＡＭソリューションの一部としても重要であり、したがって、Ｉ

ＤＣＳプロビジョニングは、企業が縮小、拡大、合併する、または既存のシステムをＩａａＳ／ＰａａＳ／ＳａａＳ環境と統合しようとする場合、移行時において、オンプレミスソリューションよりも高い柔軟性を提供し得る。ＩＤＣＳ手法は、一度限りのアップグレードにおいて時間と労力を節約することができ、必要な部門、事業部、およびシステムの適切な統合を保証する。企業ではこの技術をスケーリングする必要性が密かに発生することが多く、企業体系全体にスケーラブルなＩＤＣＳ機能を迅速に提供することは、柔軟性、コスト、および制御の点で利益をもたらし得る。

【００４４】

一般的に、被雇用者は、長年にわたり、職種の変化に応じて追加の権限が付与される（すなわち「権限のクリープ」）。規制が緩やかな企業は一般的に「立証」プロセスが欠落している。このプロセスは、企業の被雇用者の権限（たとえばネットワーク、サーバ、アプリケーション、およびデータへのアクセス権）を定期的に監査して、過剰な権限が付与されたアカウントの原因となる権限のクリープを止めるまたは減速させる管理者を必要とする。したがって、一実施形態は、定期的に実施される（少なくとも１年に一度）立証プロセスを提供し得る。さらに、合併および買収に伴い、これらのツールおよびサービスの必要性は急激に増す。ユーザが、ＳａａＳシステムに存在する、オンプレミス上に存在する、異なる部門にまたがっている、および／またはデプロビジョニングされているもしくは再度割当てられているからである。クラウドへの移動はこの状況をさらに複雑にする可能性があり、プロセスは、既存の手動管理されることが多い証明方法を超えて急速にエスカレートする可能性がある。したがって、一実施形態は、これらの機能を自動化し、高度な分析を、ユーザプロファイル、アクセス履歴、プロビジョニング／デプロビジョニング、および細分化された権利に適用する。

【００４５】

一実施形態はアイデンティティ分析を提供する。一般的に、アイデンティティ分析を、包括的な証明および立証のためにＩＡＭエンジンと統合する機能は、組織のリスクプロファイルを安全にするためには不可欠となる可能性がある。適切にデプロイされたアイデンティティ分析は、内部ポリシー全体の施行を要求する可能性がある。クラウドおよびオンプレミス全体で統一された単一管理ビューを提供するアイデンティティ分析は、予防的ガバナンス、リスク、およびコンプライアンス（governance, risk, and compliance：「ＧＲＣ」）企業環境における必要性が高く、リスクを低減しコンプライアンス規則を満たすための閉ループプロセスを提供するのに役立ち得る。したがって、一実施形態はアイデンティティ分析を提供する。アイデンティティ分析は、管理者、幹部職員、および監査役が必要とするレポートおよび分析のために、クライアントが簡単にカスタマイズすることで特定の産業条件および政府規則に適合する。

【００４６】

一実施形態は、セルフサービスおよびアクセス要求機能を提供することにより、エンドユーザの体験および効率を改善するとともに、ヘルプデスクコールに要するコストを低減する。一般的に、多数の企業はその従業員のためにオンプレミスのセルフサービスアクセス要求をデプロイするが、多くは、これらのシステムを正式な企業の壁の外側まで適切に拡張していない。従業員の用途の範囲外の、ポジティブなデジタル顧客体験が、ビジネスの信頼性を高め最終的には収入の増加に貢献し、企業は、顧客ヘルプデスクコールを減じるだけでなく顧客の満足度を高める。したがって、一実施形態は、オープン標準に基づいておりかつ必要に応じて既存のアクセス制御ソフトウェアおよびＭＦＡメカニズムとシームレスに統合される、アイデンティティクラウドサービス環境を提供する。ＳａａＳ配信モデルは、以前はシステムのアップグレードおよびメンテナンスに費やされていた時間と労力を省き、ＩＴ専門スタッフを解放してより中心的なビジネスアプリケーションに集中できるようにする。

【００４７】

一実施形態は、特権アカウント管理（privileged account management：「ＰＡＭ」）を提供する。一般的に、すべての組織は、ＳａａＳ、ＰａａＳ、ＩａａＳまたはオンプレ

ミスアプリケーションいずれを使用しても、システムアドミニストレータ、幹部職員、人事担当役員、契約者、システムインテグレータなどのスーパーユーザのアクセスクレデンシャルを用いたインサイダーによる特権アカウントの不正使用に弱い。加えて、外部の脅威は一般的に、先ず低レベルユーザアカウントを侵害し、最終的には企業システム内の特権ユーザアクセス制御に到達してこれを利用する。したがって、一実施形態は、ＰＡＭを提供することにより、このような不正なインサイダーによるアカウントの使用を防止する。ＰＡＭソリューションの主要コンポーネントはパスワードボールト（password vault）であり、これはさまざまなやり方で供給し得る。たとえば、企業サーバ上にインストールされるソフトウェアとして、これも企業サーバ上の仮想アプライアンスとして、パッケージングされたハードウェア／ソフトウェアアプライアンスとして、または、クラウドサービスの一部として、さまざまなやり方で供給し得る。ＰＡＭ機能は、エンベロープ内で保持されサイン・インおよびサイン・アウトのためのマニフェストで定期的に変更されるパスワードを格納するために使用される物理的な安全場所と同様である。一実施形態は、パスワードのチェックアウトだけでなく、タイムリミットの設定、強制的な期間変更、自動的なチェックアウトの追跡、およびすべてのアクティビティに関する報告を、可能にする。一実施形態は、要求されたリソースに、ユーザがパスワードを知らない状態で、直接接続する方法を提供する。この機能はまた、セッション管理およびその他の機能の方法に道を開く。

【００４８】

一般的に、ほとんどのクラウドサービスは、ＡＰＩおよび管理インターフェイスを利用している。これらは、侵入者がセキュリティを迂回する機会を与える。したがって、一実施形態は、ＰＡＭの実施におけるこれらの欠陥を埋める。クラウドへの移行によってＰＡＭに新たな課題が発生するからである。小規模から中規模の多くのビジネスは現在自身のＳａａＳシステム（たとえばOffice 365）を管理しているが、大企業は自身のＳａａＳおよびＩａａＳサービスの回転数を上げる個々のビジネス単位を持つことが増えている。これらの顧客は、ＰＡＭ機能がアイデンティティクラウドサービスソリューションに含まれるかまたはそのＩａａＳ／ＰａａＳプロバイダから得られるが、この責務を扱った経験がほとんどない。加えて、場合によっては、多くの異なる地理的に分散したビジネス単位が、同じＳａａＳアプリケーションの管理責任を分離しようとする。したがって、一実施形態は、こういった状況にある顧客が、既存のＰＡＭをアイデンティティクラウドサービスの全体的なアイデンティティフレームワークの中にリンクさせ、より高い安全性とコンプライアンスに向けて、ビジネスニーズが要求するクラウドロード条件に合わせて確実に調整することを、可能にする。

【００４９】

ＡＰＩプラットフォーム

実施形態が提供するＡＰＩプラットフォームは、機能のコレクションをサービスとしてエクスポーズする。ＡＰＩはマイクロサービスに集約され、各マイクロサービスは、１つ以上のＡＰＩをエクスポーズする。すなわち、各マイクロサービスは異なる種類のＡＰＩをエクスポーズし得る。一実施形態において、各マイクロサービスはそのＡＰＩを通してしか通信しない。一実施形態において、各ＡＰＩはマイクロサービスであってもよい。一実施形態において、複数のＡＰＩが１つのサービスに、このサービスが提供するターゲット機能に基づいて集約される（たとえばＯＡｕｔｈ、ＳＡＭＬ、Ａｄｍｉｎなど）。結果として、同様のＡＰＩは別々のランタイムプロセスとしてエクスポーズされない。ＡＰＩは、ＩＤＣＳが提供するサービスを使用するためにサービス顧客が利用できるようにされるものである。

【００５０】

一般的に、ＩＤＣＳのウェブ環境において、ＵＲＬは、３つの部分として、ホストと、マイクロサービスと、リソースとを含む（たとえばホスト／マイクロサービス／リソース）。一実施形態において、マイクロサービスは、特定のＵＲＬプレフィックスを有することを特徴とし（たとえば「host/oauth/v1」）、実際のマイクロサービスは「oauth/v1」

である。「oauth/v1」の下で複数のＡＰＩが存在し、たとえば、トークン（token）を要求するためのＡＰＩ：「host/oauth/v1/token」、ユーザを認証する（authorize）ためのＡＰＩ：「host/oauth/v1/authorize」などである。すなわち、ＵＲＬはマイクロサービスを実現し、ＵＲＬのリソース部分はＡＰＩを実現する。したがって、同じマイクロサービスの下で複数のＡＰＩが集約される。一実施形態において、ＵＲＬのホスト部分はテナントを特定する（たとえばhttps://tenant3.identity.oraclecloud.com:/oauth/v1/token）。

【００５１】

必要なエンドポイントを有する外部サービスと統合するアプリケーションを構成し当該構成を最新状態に保つことは、一般的に難題である。この難題を克服するために、実施形態は、パブリックディスカバリＡＰＩを周知の場所にエクスポーズし、そこから、アプリケーションは、ＡＤＣＳ　ＡＰＩを消費するために必要なＩＤＣＳに関する情報を発見する（discover）ことができる。一実施形態において、２つのディスカバリ文献がサポートされ、それらは、ＩＤＣＳ構成（たとえば、〈IDCS-URL〉/.well-known/idcs-configurationのＩＤＣＳ、ＳＡＭＬ、ＳＣＩＭ、ＯＡｕｔｈ、およびＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ構成を含む）と、（たとえば〈IDCS-URL〉/.well-known/openid-configurationの）産業標準ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ構成とである。アプリケーションは、単一のＩＤＣＳ　ＵＲＬで構成されることにより、ディスカバリ文献を取出すことができる。

【００５２】

図６は、一実施形態におけるＩＤＣＳのシステムビュー６００を提供するブロック部である。図６において、さまざまなアプリケーション／サービス６０２のうちのいずれも、ＩＤＣＳ　ＡＰＩに対してＨＴＴＰコールを行なうことにより、ＩＤＣＳサービスを使用することができる。このようなアプリケーション／サービス６０２の例は、ウェブアプリケーション、ネイティブアプリケーション（たとえばWindows（登録商標）アプリケーション、ｉＯＳ（登録商標）アプリケーション、アンドロイド（登録商標）アプリケーションなど、特定のオペレーティングシステム上で走るように構築されたアプリケーション）、ウェブサービス、顧客アプリケーション、パートナーアプリケーション、または、サービスとしてのソフトウェア（Software as a Service：「ＳａａＳ」）、ＰａａＳ、およびサービスとしてのインフラストラクチャ（Infrastructure as a Service：「ＩａａＳ」）など、パブリッククラウドによって提供されるサービスである。

【００５３】

一実施形態において、ＩＤＣＳサービスを要求するアプリケーション／サービス６０２のＨＴＴＰ要求は、オラクルパブリッククラウドＢＩＧ－ＩＰアプライアンス６０４およびＩＤＣＳ　ＢＩＧ－ＩＰアプライアンス６０６（またはロードバランサなどの同様の技術、または、適切なセキュリティルールを実現してトラフィックを保護するサービスとしてのクラウドロードバランサ（Cloud Load Balancer as a Service：「ＬＢａａＳ」）と呼ばれているコンポーネント）を通る。しかしながら、この要求はどのようなやり方で受信されてもよい。ＩＤＣＳ　ＢＩＧ－ＩＰアプライアンス６０６（または、適用できる場合は、ロードバランサまたはクラウドＬＢａａＳなどの同様の技術）において、クラウドプロビジョニングエンジン６０８は、テナントおよびサービスの調整を実行する。一実施形態において、クラウドプロビジョニングエンジン６０８は、クラウドにオンボードされている新たなテナントに対応付けられた内部セキュリティアーティファクト、または、顧客が購入した新たなサービスインスタンスを管理する。

【００５４】

このＨＴＴＰ要求は次にＩＤＣＳウェブルーティング層６１０によって受信される。このルーティング層は、セキュリティゲート（すなわちクラウドゲート）を実現し、サービスルーティングならびにマイクロサービス登録および発見６１２を提供する。要求されるサービスに応じて、ＨＴＴＰ要求は、ＩＤＣＳ中間層６１４のＩＤＣＳマイクロサービスに転送される。ＩＤＣＳマイクロサービスは、外部および内部ＨＴＴＰ要求を処理する。ＩＤＣＳマイクロサービスは、プラットフォームサービスおよびインフラストラクチャサー

ビスを実現する。ＩＤＣＳプラットフォームサービスは、ＩＤＣＳのビジネスを実現する、別々にデプロイされたＪａｖａベースのランタイムサービスである。ＩＤＣＳインフラストラクチャサービスは、ＩＤＣＳに対してインフラストラクチャサポートを提供する、別々にデプロイされたランタイムサービスである。ＩＤＣＳはさらに、ＩＤＣＳサービスによって使用される共有ライブラリとしてパッケージングされた共通コードであるインフラストラクチャライブラリと、共有ライブラリとを含む。インフラストラクチャサービスおよびライブラリは、プラットフォームサービスがその機能を実現するために要求するサポート機能を提供する。

【００５５】

プラットフォームサービス

一実施形態において、ＩＤＣＳは標準認証プロトコルをサポートし、したがって、ＩＤＣＳマイクロサービスは、ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ、ＯＡｕｔｈ、ＳＡＭＬ２、クロスドメインアイデンティティ管理のためのシステム（System for Cross-domain Identity Management++：「ＳＣＩＭ＋＋」）などのプラットフォームサービスを含む。

【００５６】

ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔプラットフォームサービスは、標準ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔログイン／ログアウトフローを実現する。対話型のウェブベースおよびネイティブアプリケーションは、標準のブラウザベースのＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔフローを推進することによりユーザ認証を要求し、ユーザの認証されたアイデンティティを伝達するＪａｖａＳｃｒｉｐｔ（登録商標）オブジェクト表記（JavaScript Object Notation（「ＪＳＯＮ」）ウェブトークン（Web Token「ＪＷＴ」）である標準アイデンティティトークンを受信する。内部において、ランタイム認証モデルはステートレスであり、ユーザの認証／セッション状態をホストＨＴＴＰクッキー（ＪＷＴアイデンティティトークンを含む）の形態で維持する。ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔプロトコルを介して開始された認証対話は、ローカルおよび連携ログインのためにユーザのログイン／ログアウトセレモニーを実現する信頼できるＳＳＯサービスに委任される。この機能のさらなる詳細は以下において図１０および図１１を参照しながら開示される。一実施形態において、ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ機能は、たとえばＯｐｅｎＩＤ　Ｆｏｕｎｄａｔｉｏｎ標準に従って実現される。

【００５７】

ＯＡｕｔｈ２プラットフォームサービスは、トークン認可サービスを提供する。これは、ユーザの権利を伝達するアクセストークンを作成し検証してＡＰＩコールを行なうためのリッチなＡＰＩインフラストラクチャを提供する。これは、ある範囲の有用なトークン付与タイプをサポートし、顧客がクライアントをそのサービスに安全に接続することを可能にする。これは、標準の２者間および３者間ＯＡｕｔｈ２トークン付与タイプを実現する。ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ（「ＯＩＤＣ」）をサポートすることにより、コンプライアントなアプリケーション（ＯＩＤＣリレーパーティ（「ＲＰ」））が、アイデンティティプロバイダとしてのＩＤＣＳと統合されることを可能にする（ＯＩＤＣ　ＯｐｅｎＩＤプロバイダ（「ＯＰ」）。同様に、ＯＩＤＣ　ＲＰとしてのＩＤＣＳをソーシャルＯＩＤＣ　ＯＰ（たとえばFacebook（登録商標）、Google（登録商標）など）と統合することにより、顧客は、アプリケーションに対する社会的アイデンティのポリシーベースアクセスを可能にする。一実施形態において、ＯＡｕｔｈ機能は、たとえば、インターネットエンジニアリングタスクフォース（Internet Engineering Task Force：「ＩＥＴＦ」）、コメント要求（Request for Comments：「ＲＦＣ」）６７４９に従って実現される。

【００５８】

ＳＡＭＬ２プラットフォームサービスは、アイデンティティ連携サービスを提供する。これは、顧客が、ＳＡＭＬアイデンティティプロバイダ（identity provider：「ＩＤＰ」）およびＳＡＭＬサービスプロバイダ（service provider：「ＳＰ」）関係モデルに基づいて、そのパートナーとの連携合意を設定することを可能にする。一実施形態において、

ＳＡＭＬ２プラットフォームサービスは、標準ＳＡＭＬ２ブラウザポストログインおよびログアウトプロファイルを実現する。一実施形態において、ＳＡＭＬ機能は、たとえばＩＥＴＦ、ＲＦＣ７５２２に従って実現される。

【００５９】

ＳＣＩＭは、ユーザアイデンティ情報を、たとえばＩＥＴＦ、ＲＦＣ　７６４２、７６４３、７６４４によって提供される、アイデンティティドメインまたは情報技術（「ＩＴ」）システム間でのユーザアイデンティティ情報の交換を自動化するためのオープン標準である。ＳＣＩＭ＋＋プラットフォームサービスは、アイデンティティ管理サービスを提供し、顧客がＩＤＣＳのＩＤＰフィーチャー（feature）にアクセスすることを可能にする。管理サービスは、アイデンティティライフサイクル、パスワード管理、グループ管理などをカバーするステートレスなＲＥＳＴインターフェイス（すなわちＡＰＩ）のセットをエクスポーズし、ウェブアクセス可能なリソースのようなアーティファクトをエクスポーズする。

【００６０】

すべてのＩＤＣＳ構成アーティファクトはリソースであり、管理サービスのＡＰＩは、ＩＤＣＳリソース（たとえばユーザ、ロール、パスワードポリシー、アプリケーション、ＳＡＭＬ／ＯＩＤＣアイデンティティプロバイダ、ＳＡＭＬサービスプロバイダ、キー、証明、通知テンプレートなど）の管理を可能にする。管理サービスは、ＳＣＩＭ標準を強化および拡張することにより、すべてのＩＤＣＳリソースに対する作成（Create）、読取り（Read）、更新（Update）、削除（Delete）、および問合せ（Query）（「ＣＲＵＤＱ」）動作のためにスキーマベースのＲＥＳＴ　ＡＰＩを実現する。加えて、ＩＤＣＳ自体の管理および構成に使用されるＩＤＣＳのすべての内部リソースは、ＳＣＩＭベースのＲＥＳＴ　ＡＰＩとしてエクスポーズされる。アイデンティティストア６１８へのアクセスはＳＣＩＭ＋＋ＡＰＩに分離される。

【００６１】

一実施形態において、たとえば、ＳＣＩＭ標準は、ＳＣＩＭ規格によって規定されるユーザおよびグループリソースを管理するように実現されるのに対し、ＳＣＩＭ＋＋は、ＳＣＩＭ規格によって規定される言語を用いてさらに他のＩＤＣＳ内部リソース（たとえばパスワードポリシー、ロール、設定など）をサポートするように構成される。

【００６２】

管理サービスは、ＳＣＩＭ２．０標準エンドポイントを、標準ＳＣＩＭ２．０コアスキーマと、必要に応じてスキーマ拡張とを用いてサポートする。加えて、管理サービスは、いくつかのＳＣＩＭ２．０準拠エンドポイント拡張をサポートすることにより、その他のＩＤＣリソースを、たとえばユーザ、グループ、アプリケーション、設定などを、管理する。管理サービスはまた、ＣＲＵＤＱ動作は実行しないがその代わりに機能サービスを、たとえば「UserPasswordGenerator」、「UserPasswordValidator」などを提供する、リモートプロシージャコールスタイル（remote procedure call-style：「ＲＰＣスタイル」）ＲＥＳＴインターフェイスのセットをサポートする。

【００６３】

ＩＤＣＳ管理ＡＰＩは、ＯＡｕｔｈ２プロトコルを認証および認可に使用する。ＩＤＣＳは、ウェブサーバ、モバイル、およびJavaScriptアプリケーションのためのシナリオといった共通のＯＡｕｔｈ２シナリオをサポートする。ＩＤＣＳ　ＡＰＩへのアクセスはアクセストークンによって保護される。ＩＤＣＳ管理ＡＰＩにアクセスするために、アプリケーションは、ＩＤＣＳ管理コンソールを通してＯＡｕｔｈ２クライアントとしてまたはＩＤＣＳアプリケーションとして（この場合ＯＡｕｔｈ２クライアントは自動的に作成される）登録される必要があり、また、所望のＩＤＣＳ管理ロールを与えられる必要がある。ＩＤＣＳ管理ＡＰＩコールを行なうとき、アプリケーションは先ず、ＩＤＣＳ　ＯＡｕｔｈ２サービスにアクセストークンを要求する。このトークンを取得した後に、このアプリケーションはアクセストークンを、そこにＨＴＴＰ認可ヘッダを含めて送信する。アプリケーションは、ＩＤＣＳ管理ＲＥＳＴ　ＡＰＩを直接使用することができる、または、Ｉ

ＤＣＳＪａｖａクライアントＡＰＩライブラリを使用することができる。
【００６４】
インフラストラクチャサービス
ＩＤＣＳインフラストラクチャサービスは、ＩＤＣＳプラットフォームサービスの機能をサポートする。これらのランタイムサービスは、（ユーザ通知、アプリケーション申込、およびデータベースに対する監査を非同期的に処理するための）イベント処理サービスと、（ジョブをスケジューリングして実行するため、たとえば、ユーザの介入が不要な長時間実行タスクを直ちに実行するまたは設定時間に実行するための）ジョブスケジューラサービスと、キャッシュ管理サービスと、（パブリッククラウドストレージサービスと統合するための）ストレージ管理サービスと、（レポートおよびダッシュボードを生成するための）レポートサービスと、（内部ユーザ認証およびＳＳＯを管理するための）ＳＳＯサービスと、（異なる種類のユーザインターフェイス（user interface：「ＵＩ」）クライアントをホストするための）ユーザインターフェイス（「ＵＩ」）サービスと、サービスマネージャサービスとを含む。サービスマネージャは、オラクルパブリッククラウドとＩＤＣＳとの間の内部インターフェイスである。サービスマネージャは、オラクルパブリッククラウドによって発行されたコマンドを管理し、このコマンドはＩＤＣＳによって実現される必要がある。たとえば、顧客が、何かを購入できる状態になる前にクラウドストア内のアカウントに対してサインアップした場合、クラウドは、テナントを作成することを依頼するための要求をＩＤＣＳに送信する。この場合、サービスマネージャは、ＩＤＣＳがサポートするとクラウドが予測するクラウド固有の動作を実現する。
【００６５】
ＩＤＣＳマイクロサービスは、ネットワークインターフェイスを通して別のＩＤＣＳマイクロサービスをコールしてもよい（すなわちＨＴＴＰ要求）。
【００６６】
一実施形態において、ＩＤＣＳはまた、データベーススキーマを使用できるようにするスキーマサービス（またはパーシステンス（persistence）サービス）を提供し得る。スキーマサービスは、データベーススキーマを管理する責任をＩＤＣＳに委任することを可能にする。したがって、ＩＤＣＳのユーザはデータベースを管理する必要がない。なぜなら、この機能を提供するＩＤＣＳサービスが存在するからである。たとえば、ユーザは、データベースを用いてテナントごとにスキーマをパーシストしてもよく、データベース内にスペースがなくなったときにはスキーマサービスが、ユーザがデータベースを自身で管理しなくてもよいように、別のデータベースを取得し上記空間を拡大するという機能を管理する。
【００６７】
ＩＤＣＳはさらに、ＩＤＣＳが必要とする／生成するデータリポジトリであるデータストアを含む。これは、（ユーザ、グループなどを格納する）アイデンティティストア６１８、（ＩＤＣＳが自身を構成するために使用する構成データを格納する）グローバルデータベース６２０、（テナントごとにスキーマを分離し顧客ごとに顧客データを格納する）オペレーショナルスキーマ６２２、（監査データを格納する）監査スキーマ６２４、（キャッシュされたオブジェクトを格納することにより実施速度を高める）キャッシングクラスタ６２６などを含む。内部および外部のすべてのＩＤＣＳコンシューマは、標準ベースのプロトコルに従ってアイデンティティサービスと統合される。これにより、ドメインネームシステム（domain name system：「ＤＮＳ」）を用いて、どこに要求をルーティングすべきかを決定することができ、アプリケーションを消費することをアイデンティティサービスの内部実現を理解することから切離す。
【００６８】
リアルタイムおよびニア・リアルタイムタスク
ＩＤＣＳは、要求されたサービスのタスクを、同期リアルタイムタスクと非同期ニア・リアルタイムタスクとに分離する。リアルタイムタスクは、ユーザが進むのに必要なオペレーションのみを含む。一実施形態において、リアルタイムタスクは、最少の遅延で実行さ

れるタスクであり、ニア・リアルタイムタスクは、バックグラウンドにおいて、ユーザが待つことなく実行されるタスクである。一実施形態において、リアルタイムタスクは、実質的に遅延なしでまたはごくわずかな遅延で実行されるタスクであり、ユーザには、ほぼ瞬時に実行されているように見えるタスクである。

【００６９】

リアルタイムタスクは、特定のアイデンティティサービスの主要なビジネス機能を実行する。たとえば、ログインサービスを要求するとき、アプリケーションは、メッセージを送信してユーザのクレデンシャルを認証しそれに対するセッションクッキーを取得する。ユーザが体験するのは、システムへのログインである。しかしながら、ユーザのログインに関しては、ユーザが誰であるかの検証、監査、通知の送信など、その他いくつかのタスクが実行されるであろう。したがって、クレデンシャルの検証は、ユーザがＨＴＴＰクッキーを与えられてセッションを開始するように、リアルタイムで実行されるタスクであるが、通知（たとえば電子メールを送信してアカウント作成を通知すること）、監査（たとえば追跡／記録）などに関連するタスクは、ユーザが最少の遅延で進むことができるよう非同期的で実行することができるニア・リアルタイムタスクである。

【００７０】

マイクロサービスを求めるＨＴＴＰ要求が受信されると、対応するリアルタイムタスクが中間層のマイクロサービスによって実行され、必ずしもリアルタイム処理を受けない演算ロジック／イベントなどの残りのニア・リアルタイムタスクは、メッセージキュー６２８にオフロードされる。メッセージキュー６２８は、配信および処理が保証された状態でスケーラビリティが高い非同期イベント管理システム６３０をサポートする。したがって、特定の挙動は、フロントエンドからバックエンドにプッシュされることにより、ＩＤＣＳが、応答時間のレイテンシを少なくすることにより、ハイレベルサービスを顧客に提供することを、可能にする。たとえば、ログインプロセスは、クレデンシャルの検証、ログレポートの提出、最後のログイン時間の更新などを含み得るが、これらのタスクは、メッセージキューにオフロードして、リアルタイムではなくニア・リアルタイムで実行することができる。

【００７１】

一例において、システムが新たなユーザを登録または作成する必要がある場合がある。システムは、ＩＤＣＳ　ＳＣＩＭ　ＡＰＩをコールしてユーザを作成する。最終結果として、ユーザがアイデンティティストア６１８において作成されたときにこのユーザがそのパスワードをリセットするためのリンクを含む通知電子メールを得る。ＩＤＣＳが、新たなユーザを登録または作成することを求める要求を受けると、対応するマイクロサービスは、オペレーショナルデータベース（図６のグローバルデータベース６２０内に位置する）にある構成データに注目し、「ユーザ作成」という動作が「ユーザ作成」イベントでマーキングされていると判断する。この動作は、構成データにおいて非同期動作であることが識別される。マイクロサービスは、クライアントに戻り、ユーザの作成が正常に行なわれたことを示すが、通知電子メールの実際の送信は延期されバックエンドにプッシュされる。そうするために、マイクロサービスは、メッセージングＡＰＩ６１６を用いてこのメッセージを、ストアであるキュー６２８に入れる。

【００７２】

キュー６２８から出すために、インフラストラクチャマイクロサービスであるメッセージングマイクロサービスは、バックグラウンドにおいて継続的に実行され、キュー６２８の中にあるイベントを探してキュー６２８をスキャンする。キュー６２８の中にあるイベントは、監査、ユーザ通知、アプリケーション申込、データ解析などのイベントサブスクライバ６３０によって処理される。イベントによって示されるタスクに応じて、イベントサブスクライバ６３０は、たとえば、監査スキーマ６２４、ユーザ通知サービス６３４、アイデンティティイベントサブスクライバ６３２などと通信し得る。たとえば、メッセージングマイクロサービスは、キュー６２８の中に「ユーザ作成」イベントを発見した場合、対応する通知ロジックを実行し対応する電子メールをユーザに送信する。

【００７３】
一実施形態において、キュー６２８は、マイクロサービス６１４によってパブリッシュされたオペレーショナルイベントと、ＩＤＣＳリソースを管理するＡＰＩ６１６によってパブリッシュされたリソースイベントとをキューの中に入れる。
【００７４】
ＩＤＣＳは、リアルタイムキャッシング構造を用いてシステムパフォーマンスおよびユーザ体験を向上させる。キャッシュそのものは、マイクロサービスとしても提供される。ＩＤＣＳは、ＩＤＣＳによってサポートされている顧客の数の増加に伴って増大するエラスティック・キャッシュクラスタ６２６を実現する。キャッシュクラスタ６２６は、以下でより詳細に開示される分散型データグリッドで実現されてもよい。一実施形態において、書込専用リソースがキャッシュをバイパスする。
【００７５】
一実施形態において、ＩＤＣＳランタイムコンポーネントは、ヘルスおよびオペレーショナルメトリクスを、オラクル社のオラクルパブリッククラウドなどのパブリッククラウドのこのようなメトリクスを収集するパブリッククラウドモニタリングモジュール６３６に対してパブリッシュする。
【００７６】
一実施形態において、ＩＤＣＳを用いてユーザを作成してもよい。たとえば、クライアントアプリケーション６０２は、ＲＥＳＴ　ＡＰＩコールを発行してユーザを作成してもよい。管理サービス（６１４のプラットフォームサービス）は、このコールをユーザマネージャ（６１４のインフラストラクチャライブラリ／サービス）に委任する。そうすると、ユーザマネージャは、このユーザを、ＩＤストア６１８内の特定テナント用ＩＤストアストライプにおいて作成する。「ユーザ作成成功（User Create Success）」の場合、ユーザマネージャは、オペレーションを検査することにより検査スキーマ６２４内のテーブルを検査し、メッセージキュー６２８に対して「identity.user.create.success」をパブリッシュする。アイデンティティサブスクライバ６３２は、このイベントをピックアップし、新たに作成されたログイン詳細を含む「ウェルカム」電子メールを、新たに作成されたユーザに送信する。
【００７７】
一実施形態において、ＩＤＣＳを用いてロールをユーザに与えて、その結果ユーザがアクションをプロビジョニングしてもよい。たとえば、クライアントアプリケーション６０２は、ＲＥＳＴ　ＡＰＩコールを発行してユーザにロールを付与してもよい。管理サービス（６１４のプラットフォームサービス）は、このコールをロールマネージャ（６１４のインフラストラクチャライブラリ／サービス）に委任してもよい。このロールマネージャは、ＩＤストア６１８内の特定テナント用ＩＤストアストライプにおけるロールを付与する。「ロール付与成功（Role Grant Success）」の場合、ロールマネージャは、監査スキーマ６２４における監査テーブルに対するオペレーションを監査し、メッセージキュー６２８に対して「identity.user.role.grant.success」をパブリッシュする。アイデンティティサブスクライバ６３２は、このイベントをピックアップしプロビジョニング付与ポリシーを評価する。付与されているロールに対するアクティブなアプリケーション付与があった場合、プロビジョニングサブスクライバは、何らかの検証を実行し、アカウント作成を開始し、ターゲットシステムをコールアウトし、ターゲットシステムにアカウントを作成し、アカウント作成が成功したとマーキングする。これらの機能各々の結果として、「prov.account.create.initiate」、「prov.target.create.initiate」、「prov.target.create.success」または「prov.account.create.success」などの対応するイベントがパブリッシュされることになり得る。これらのイベントは、直近Ｎ日間でターゲットシステムにおいて作成されたアカウントの数を合計する自身のビジネスメトリクスを有し得る。
【００７８】
一実施形態において、ＩＤＣＳはユーザのログインのために使用することができる。たとえば、クライアントアプリケーション６０２は、サポートされている認証フローのうちの

１つを用いてユーザのログインを要求してもよい。ＩＤＣＳは、ユーザを認証し、成功すると、監査スキーマ６２４における監査テーブルに対するオペレーションを監査する。失敗すると、ＩＤＣＳは、監査スキーマ６２４における失敗を監査し、メッセージキュー６２８の「login.user.login.failure」イベントをパブリッシュする。ログインサブスクライバは、このイベントをピックアップし、ユーザに対するそのメトリクスを更新し、ユーザのアクセス履歴についての追加分析を実行する必要があるか否かを判断する。

【００７９】

したがって、「制御の反転」機能を実現する（たとえば実行の流れを変更することにより、後の時点におけるオペレーションの実行を、当該オペレーションが別のシステムの支配下になるように、スケジュールする）ことにより、実施形態は、その他のイベントキューおよびサブスクライバを動的に追加して、小さなユーザサンプルに対する新たな特徴を、より広いユーザベースにデプロイする前にテストする、または、特定の内部または外部の顧客のための特定のイベントを処理することができる。

【００８０】

ステートレス機能

ＩＤＣＳマイクロサービスはステートレスである。これは、マイクロサービスそのものはステートを保持しないことを意味する。「ステート」とは、アプリケーションがその機能を果たすために使用するデータのことを言う。ＩＤＣＳは、マルチテナント機能を、すべてのステートを、ＩＤＣＳデータ層内の特定テナント向けリポジトリにパーシストすることによって提供する。中間層（すなわち要求を処理するコード）は、アプリケーションコードと同じ場所に格納されているデータを有しない。したがって、ＩＤＣＳは横方向および縦方向双方においてスケーラビリティが高い。

【００８１】

縦方向のスケーリング（またはスケールアップ／ダウン）は、システム内の１つのノードにリソースを追加する（またはこのノードからリソースを削除する）ことを意味し、１つのコンピュータにＣＰＵまたはメモリを追加することを伴うのが一般的である。縦方向のスケーラビリティによって、アプリケーションはそのハードウェアの限界までスケールアップすることができる。横方向のスケーリング（またはスケールアウト／イン）は、新たなコンピュータを分散型ソフトウェアアプリケーションに追加するといったように、より多くのノードをシステムに追加する（またはシステムからノードを削除する）ことを意味する。横方向のスケーラビリティにより、アプリケーションはほぼ無限にスケーリング可能であり、ネットワークによって提供される帯域幅の量のみの制約を受ける。

【００８２】

ＩＤＣＳの中間層がステートレスであることにより、ＣＰＵをさらに追加するだけで横方向にスケーラブルになり、アプリケーションの仕事を実行するＩＤＣＳコンポーネントは、特定なアプリケーションが走っている指定された物理的インフラストラクチャを持つ必要がない。ＩＤＣＳの中間層がステートレスであることにより、非常に多くの顧客／テナントにアイデンティティサービスを提供しているときであっても、ＩＤＣＳの可用性が高くなる。ＩＤＣＳアプリケーション／サービスを通る各パスは、専らアプリケーショントランザクションを実行するためにＣＰＵ用途に集中するが、データの格納にハードウェアを使用しない。スケーリングは、必要に応じてより多くのコピーを追加できるパーシステンス層にトランザクション用のデータが格納される一方で、アプリケーションが走っているときにより多くのスライスを追加することによって実現される。

【００８３】

ＩＤＣＳウェブ層、中間層、およびデータ層は各々独立してかつ別々にスケーリング可能である。ウェブ層をスケーリングすることにより、より多くのＨＴＴＰ要求を扱うことができる。中間層をスケーリングすることにより、より多くのサービス機能をサポートすることができる。データ層をスケーリングすることにより、より多くのテナントをサポートすることができる。

【００８４】

ＩＤＣＳ機能ビュー

図６Ａは、一実施形態におけるＩＤＣＳの機能ビューのブロック図の一例６００ｂである。ブロック図６００ｂにおいて、ＩＤＣＳ機能スタックは、サービスと、共有ライブラリと、データストアとを含む。サービスは、ＩＤＣＳプラットフォームサービス６４０ｂと、ＩＤＣＳプレミアムサービス６５０ｂと、ＩＤＣＳインフラストラクチャサービス６６２ｂとを含む。一実施形態において、ＩＤＣＳプラットフォームサービス６４０ｂおよびＩＤＣＳプレミアムサービス６５０ｂは、別々にデプロイされたＪａｖａベースのランタイムサービスであり、ＩＤＣＳのビジネスを実現する。ＩＤＣＳインフラストラクチャサービス６６２ｂは、別々にデプロイされたランタイムサービスであり、ＩＤＣＳに対するインフラストラクチャサポートを提供する。共有ライブラリは、ＩＤＣＳサービスによって使用される共有ライブラリとしてパッケージングされた共通コードであるＩＤＣＳインフラストラクチャライブラリ６８０ｂと、共有ライブラリとを含む。データストアは、ＩＤＣＳが必要とする／生成するデータリポジトリであり、アイデンティティストア６９８ｂ、グローバル構成７００ｂ、メッセージストア７０２ｂ、グローバルテナント７０４ｂ、パーソナライゼーション設定７０６ｂ、リソース７０８ｂ、ユーザ一時データ７１０ｂ、システム一時データ７１２ｂ、テナントごとのスキーマ（管理されたＥｘａＤａｔａ）７１４ｂ、オペレーショナルストア（図示せず）、キャッシングストア（図示せず）などを含む。

【００８５】

一実施形態において、ＩＤＣＳプラットフォームサービス６４０ｂは、たとえばＯｐｅｎＩＤ Ｃｏｎｎｅｃｔサービス６４２ｂ、ＯＡｕｔｈ２サービス６４４ｂ、ＳＡＭＬ２サービス６４６ｂ、およびＳＣＩＭ＋＋サービス６４８ｂを含む。一実施形態において、ＩＤＣＳプレミアムサービスは、たとえば、クラウドＳＳＯおよびガバナンス６５２ｂ、企業ガバナンス６５４ｂ、ＡｕｔｈＮブローカー６５６ｂ、連携ブローカー６５８ｂ、およびプライベートアカウント管理６６０ｂを含む。

【００８６】

ＩＤＣＳインフラストラクチャサービス６６２ｂおよびＩＤＣＳインフラストラクチャライブラリ６８０ｂは、ＩＤＣＳプラットフォームサービス６４０ｂがその仕事を実行するのに必要とする機能のサポートを提供する。一実施形態において、ＩＤＣＳインフラストラクチャサービス６６２ｂは、ジョブスケジューラ６６４ｂ、ＵＩ６６６ｂ、ＳＳＯ６６８ｂ、レポート６７０ｂ、キャッシュ６７２ｂ、ストレージ６７４ｂ、サービスマネージャ６７６ｂ（パブリッククラウド制御）、およびイベントプロセッサ６７８ｂ（ユーザ通知、アプリケーション申込、監査、データ解析）を含む。一実施形態において、ＩＤＣＳインフラストラクチャライブラリ６８０ｂは、データマネージャＡＰＩ６８２ｂ、イベントＡＰＩ６８４ｂ、ストレージＡＰＩ６８６ｂ、認証ＡＰＩ６８８ｂ、認可ＡＰＩ６９０ｂ、クッキーＡＰＩ６９２ｂ、キーＡＰＩ６９４ｂ、およびクレデンシャルＡＰＩ６９６ｂを含む。一実施形態において、クラウド計算サービス６０２ｂ（内部Nimbula）は、ＩＤＣＳインフラストラクチャサービス６６２ｂおよびＩＤＣＳインフラストラクチャライブラリ６８０ｂの機能をサポートする。

【００８７】

一実施形態において、ＩＤＣＳは、顧客エンドユーザＵＩ６０４ｂ、顧客管理ＵＩ６０６ｂ、ＤｅｖＯｐｓ管理ＵＩ６０８ｂ、およびログインＵＩ６１０ｂなど、ＩＤＣＳサービスのコンシューマのためのさまざまなＵＩ６０２ｂを提供する。一実施形態において、ＩＤＣＳは、アプリケーション（たとえば顧客アプリケーション６１４ｂ、パートナーアプリケーション６１６ｂ、およびクラウドアプリケーション６１８ｂ）の統合６１２ｂならびにファームウェア統合６２０ｂを可能にする。一実施形態において、さまざまな環境がＩＤＣＳと統合されてそのアクセス制御のニーズをサポートしてもよい。このような統合は、たとえば、アイデンティティブリッジ６２２ｂ（ＡＤ統合、ＷＮＡ、およびＳＣＩＭコネクタを提供）、アパッチエージェント６２４ｂ、またはＭＳＦＴエージェント６２６ｂによって提供される。

【００８８】
一実施形態において、内部および外部のＩＤＣＳコンシューマは、ＯｐｅｎＩＤ Ｃｏｎｎｅｃｔ６３０ｂ、ＯＡｕｔｈ２ ６３２ｂ、ＳＡＭＬ２ ６３４ｂ、ＳＣＩＭ６３６ｂ、およびＲＥＳＴ／ＨＴＴＰ６３８ｂなどの標準ベースのプロトコル６２８ｂに対するＩＤＣＳのアイデンティティサービスと統合される。これにより、ドメインネームシステム（domain name system：「ＤＮＳ」）を用いて、要求をどこにルーティングするかを判断することができ、アプリケーションの消費を、アイデンティティサービスの内部実現を理解することから切離す。
【００８９】
図６ＡのＩＤＣＳ機能ビューはさらに、ＩＤＣＳが、ユーザ通知（クラウド通知サービス７１８ｂ）、ファイルストレージ（クラウドストレージサービス７１６ｂ）、およびＤｅｖＯＰｓのためのメトリクス／警告（クラウドモニタサービス（ＥＭ）７２２ｂおよびクラウドメトリクスサービス（グラファイト）７２０ｂ）のために依存する共通機能を提供する、パブリッククラウドインフラストラクチャサービスを含む。
【００９０】
クラウドゲート
一実施形態において、ＩＤＣＳはウェブ層において「クラウドゲート」を実現する。クラウドゲートは、ウェブアプリケーションがユーザＳＳＯをアイデンティティ管理システム（たとえばＩＤＣＳ）に外部化することを可能にするウェブサーバプラグインであり、これは、企業ＩＤＭスタックと協力するＷｅｂＧａｔｅまたは Ｗｅｂ Ａｇｅｎｔ技術と同様である。クラウドゲートは、ＩＤＣＳ ＡＰＩに対するアクセスを安全にするセキュリティゲートキーパの役割を果たす。一実施形態において、クラウドゲートは、ＯＡｕｔｈに基づいてＨＴＴＰリソースを保護するためにウェブポリシー施行点（Policy Enforcement Point：「ＰＥＰ」）を提供するウェブ／プロキシサーバプラグインによって実現される。
【００９１】
図７は、クラウドゲート７０２を実現する実施形態のブロック図７００である。クラウドゲート７０２は、ウェブサーバ７１２内で実行され、ポリシー施行点（「ＰＥＰ」）の役割を果たす。ポリシー施行点は、オープン標準（たとえばＯＡｔｈ２、ＯｐｅｎＩＤ Ｃｏｎｎｅｃｔなど）を用いるＩＤＣＳポリシー決定点（Policy Decision Point：「ＰＤＰ」）と統合され、一方でウェブブラウザおよびアプリケーションのＲＥＳＴ ＡＰＩリソース７１４へのアクセスを安全にするように構成されている。いくつかの実施形態において、ＰＤＰは、ＯＡｕｔｈおよび／またはＯｐｅｎＩＤ Ｃｏｎｎｅｃｔマイクロサービス７０４で実現される。たとえば、ユーザブラウザ７０６がユーザ７１０のログインを求める要求をＩＤＣＳに送信すると、対応するＩＤＣＳ ＰＤＰは、クレデンシャルを検証した後に、このクレデンシャルが十分であるか否か（たとえば第２のパスワードなどのその他のクレデンシャルを要求するか否か）を判断する。図７の実施形態において、クラウドゲート７０２は、ローカルポリシーを有するので、ＰＥＰとしてもＰＤＰとしてもその役割を果たし得る。
【００９２】
ワンタイム・デプロイメントの一部として、クラウドゲート７０２には、ＯＡｕｔｈ２クライアントとしてのＩＤＣＳが登録され、これが、ＩＤＣＳに対してＯＩＤＣおよびＯＡｕｔｈ２オペレーションを要求することを可能にする。その後、これは、要求マッチングルール（ＵＲＬをたとえばワイルドカード、通常表現などに対して如何にしてマッチングするか）の適用を受ける、アプリケーションの保護されたリソースおよび保護されていないリソースに関する構成情報を保持する。クラウドゲート７０２をデプロイすることにより、異なるセキュリティポリシーを有する異なるアプリケーションを保護することができ、保護されるアプリケーションはマルチテナントであってもよい。
【００９３】
ウェブブラウザベースのユーザアクセス中、クラウドゲート７０２は、ユーザ認証フロー

を開始するＯＩＤＣ　ＲＰ７１８として機能する。ユーザ７１０が有効なローカルユーザセッションを有していない場合、クラウドゲート７０２は、ユーザをＳＳＯマイクロサービスにリダイレクトし、ＳＳＯマイクロサービスとともにＯＩＤＣ「認証コード」フローに参加する。このフローは、アイデンティティトークンとしてのＪＷＴの配信で終了する。クラウドゲート７０８は、ＪＷＴを検証し（たとえば署名、満了、宛先／オーディエンスなどに注目し）、ユーザ７１０に関するローカルセッションクッキーを発行する。これは、保護されているリソースへのウェブブラウザのアクセスを安全にしかつローカルセッションクッキーを発行、更新、および検証するセッションマネージャ７１６として機能する。これはまた、そのローカルセッションクッキーの削除のためのログアウトＵＲＬを提供する。

【００９４】

クラウドゲート７０２はまた、ＨＴＴＰベーシックＡｕｔｈ認証者の役割を果たし、ＩＤＣＳに対するＨＴＴＰベーシックＡｕｔｈクレデンシャルを検証する。この行動は、セッションレスおよびセッションベースの（ローカルセッションクッキー）モードでサポートされる。この場合、サーバ側ＩＤＣＳセッションは生成されない。

【００９５】

ＲＥＳＴ　ＡＰＩクライアント７０８によるプログラムアクセス中、クラウドゲート７０２は、アプリケーションの保護されているＲＥＳＴ　ＡＰＩ７１４のためのＯＡｕｔｈ２リソースサーバ／フィラー７２０の役割を果たし得る。これは、認証ヘッダおよびアクセストークンに対して要求が存在するか否かを検査する。クライアント７０８（たとえばモバイル、ウェブアプリケーション、JavaScriptなど）が（ＩＤＣＳによって発行された）アクセストークンを、保護されているＲＥＳＴ　ＡＰＩ７１４とともに使用するために示すと、クラウドゲート７０２は、ＡＰＩへのアクセスを許可する前にアクセストークンを検証する（たとえば署名、満了、オーディエンスなど）。元のアクセストークンは修正無しで送られる。

【００９６】

一般的に、ＯＡｕｔｈを用いてクライアントアイデンティティ伝播トークン（たとえばクライアントが誰であるかを示す）またはユーザアイデンティティ伝播トークン（たとえばユーザが誰であるかを示す）を生成する。本実施形態において、クラウドゲートにおけるＯＡｕｔｈの実現は、たとえばＩＥＴＦ、ＲＦＣ７５１９によって提供されるようなウェブトークンのフォーマットを定めるＪＷＴに基づく。

【００９７】

ユーザがログインすると、ＪＷＴが発行される。ＪＷＴは、ＩＤＣＳによって署名され、ＩＤＣＳにおけるマルチテナント機能をサポートする。クラウドゲートは、ＩＤＣＳが発行したＪＷＴを検証することにより、ＩＤＣＳにおけるマルチテナント機能を可能にする。したがって、ＩＤＣＳは、物理構造においても、セキュリティモデルを支持する論理ビジネスプロセスにおいてもマルチテナンシーを提供する。

【００９８】

<u>テナンシーの種類</u>

ＩＤＣＳは３種類のテナンシーとして、顧客テナンシー、クライアントテナンシー、およびユーザテナンシーを特定する。顧客またはリソーステナンシーは、ＩＤＣＳの顧客が誰であるか（すなわち作業が誰に対して実行されているか）を特定する。クライアントテナンシーは、どのクライアントアプリケーションがデータにアクセスしようとしているか（すなわちどのアプリケーションが作業を実行しているか）を特定する。ユーザテナンシーは、どのユーザがアプリケーションを用いてデータにアクセスしているか（すなわち誰によって作業が実行されているか）を特定する。たとえば、専門サービス企業が大型ディスカウントショップを対象とするシステム統合機能を提供しこの大型ディスカウントショップのシステムのアイデンティティ管理を提供するためにＩＤＣＳを使用するとき、ユーザテナンシーは、この専門サービス企業に相当し、クライアントテナンシーはシステム統合機能を提供するために使用されるアプリケーションに相当し、顧客テナンシーは大型ディ

スカウントショップである。
【００９９】
これら３つのテナンシーを分離および統合することによってクラウドベースのサービスにおけるマルチテナント機能が可能になる。一般的に、オンプレミスの物理的なマシンにインストールされているオンプレミスソフトウェアの場合、これら３つのテナンシーを特定する必要はない。なぜなら、ユーザはログインするのに物理的にマシン上にいなければならないからである。しかしながら、クラウドベースのサービス構造の場合、実施形態は、トークンを持いて、誰がどのアプリケーションを使用してどのリソースにアクセスするかを判断する。３つのテナンシーは、トークンによってコーディファイ（codify）され、クラウドゲートによって施行され、中間層のビジネスサービスによって使用される。一実施形態において、ＯＡｕｔｈサーバがトークンを生成する。さまざまな実施形態において、このトークンは、ＯＡｕｔｈ以外のセキュリティプロトコルとともに使用されてもよい。
【０１００】
ユーザ、クライアント、およびリソーステナンシーを分離することにより、ＩＤＣＳが提供するサービスのユーザには実質的なビジネス上の利点が与えられる。たとえば、そうすることにより、ビジネス（たとえば健康ビジネス）のニーズおよびそのアイデンティティ管理の問題を理解するサービスプロバイダは、ＩＤＣＳが提供するサービスを購入し、ＩＤＣＳのサービスを消費する自身のバックエンドアプリケーションを開発し、このバックエンドアプリケーションをターゲットビジネスに提供することができる。したがって、サービスプロバイダは、ＩＤＣＳのサービスを拡張してその所望の機能を提供するとともにそれらを特定のターゲットビジネスに対して差出すことができる。サービスプロバイダは、ソフトウェアを構築し実行してアイデンティサービスを提供する必要はないが、その代わりに、ＩＤＣＳのサービスを拡張しカスタマイズしてターゲットビジネスのニーズに合うようにすることができる。
【０１０１】
周知のシステムの中には、顧客テナンシーである単一のテナンシーしか説明しないものがある。しかしながら、そのようなシステムは、顧客ユーザ、顧客のパートナー、顧客のクライアント、クライアント自身、または、アクセスが顧客から委任されたクライアントなどのユーザの組合わせによるアクセスを処理するときには不十分である。本実施形態において複数のテナンシーを規定し施行することにより、これらの多様なユーザに対して管理機能を特定することが容易になる。
【０１０２】
一実施形態において、ＩＤＣＳの１エンティティは、複数のテナントに同時に属しているのではなく、１つのテナントのみに属し、「テナンシー」はアーティファクトが存在する場所である。一般的に、特定の機能を実現するコンポーネントは複数存在し、これらのコンポーネントは複数のテナントに属することが可能であるまたはインフラストラクチャに属することが可能である。インフラストラクチャは、テナントの代わりに機能する必要があるとき、テナントの代わりにエンティティサービスと対話する。この場合、インフラストラクチャそのものは自身のテナンシーを有し、顧客は自身のテナンシーを有する。要求が出されたとき、この要求に関わる複数のテナンシーが存在する。
【０１０３】
たとえば、「テナント１」に属するクライアントが、「テナント３」におけるユーザを指定する「テナント２」のためのトークンを取得することを求める要求を実行する場合がある。別の例として、「テナント１」に存在するユーザが、「テナント２」が所有するアプリケーションにおけるアクションを実行する必要がある場合がある。よって、ユーザは、「テナント２」のリソースネームスペースに行きそのためのトークンを要求する必要がある。したがって、権限の委任は、「誰が」「何を」「誰」に対して行なうことができるかを特定することによって実現される。もう１つの例として、第１の組織（「テナント１」）のために働く第１のユーザが、第２の組織（「テナント２」）のために働く第２のユーザが第３の組織（「テナント３」）がホストする文書にアクセスすることを、許可しても

よい。
【０１０４】
一例において、「テナント１」のクライアントは、「テナント３」のアプリケーションにアクセスするために「テナント２」のユーザのためのアクセストークンを要求してもよい。クライアントは、「http://tenant3/oauth/token」に行きこのトークンを求めるＯＡｕｔｈ要求を呼出すことによって当該トークンを要求してもよい。クライアントは、「クライアントアサーション」を要求に含めることにより、自身が「テナント１」に存在するクライアントであることを明らかにする。このクライアントアサーションは、クライアントＩＤ（たとえば「クライアント１」）とクライアントテナンシー（「テナント１」）とを含む。「テナント１」の「クライアント１」として、クライアントは、「テナント３」に対するトークンを求める要求を呼出す権利を有し、「テナント２」のユーザのためのトークンを所望する。したがって、「ユーザアサーション」も同じＨＴＴＰ要求の一部として送られる。生成されるアクセストークンは、アプリケーションテナンシー（「テナント３」）であるターゲットテナンシーのコンテキストにおいて発行され、ユーザテナンシー（「テナント２」）を含む。
【０１０５】
一実施形態において、データ層における各テナントは、独立したストライプとして実現される。データ管理の観点からすると、アーティファクトはテナントに存在する。サービスの観点からすると、サービスは、異種のテナントとどのようにして協力するかを知っており、複数のテナンシーは、サービスのビジネス機能における異なるディメンションである。図８は、ある実施形態において複数のテナンシーを実現するシステムの一例８００を示す。システム８００はクライアント８０２を含み、クライアント８０２は、如何にしてデータベース８０６のデータ用いて作業するかを理解しているマイクロサービス８０４が提供するサービスを要求する。このデータベースは複数のテナント８０８を含み、各テナントは対応するテナンシーのアーティファクトを含む。一実施形態において、マイクロサービス８０４は、トークンを得ようとしてhttps://tenant3/oauth/tokenを通して要求されるＯＡｕｔｈマイクロサービスである。ＯＡｕｔｈマイクロサービスの機能が、マイクロサービス８０４において、データベース８０６からのデータを用いて実行されることにより、クライアント８０２の要求が正当であるか否かが検証され、正当である場合は、異なるテナンシー８０８からのデータが使用されてトークンが構成される。したがって、システム８００は、各テナンシーに与えられるサービスをサポートするだけでなく各種テナントに代わって機能し得るサービスをサポートすることによりクロステナント環境において作業できるという点において、マルチテナントである。
【０１０６】
システム８００は好都合である。理由は次の通りである。マイクロサービス８０４はデータベース８０６のデータから物理的に切離されており、クライアントにより近い場所を通ってデータを複製することにより、マイクロサービス８０４をクライアントに対するローカルサービスとして提供することができ、システム８００はサービスのアベイラビリティを管理しそれをグローバルに提供することができる。
【０１０７】
一実施形態において、マイクロサービス８０４はステートレスである。これは、マイクロサービス８０４を走らせるマシンが、特定のテナントに対するサービスを示すマーカを保持していないことを意味する。その代わりに、テナンシーは、たとえば、入ってくる要求のＵＲＬのホスト部分にマーキングされてもよい。このテナンシーはデータベース８０６のテナント８０８のうちの１つを示す。多数のテナント（たとえば何百万ものテナント）をサポートする場合、マイクロサービス８０４は、データベース８０６への同数の接続を有することはできない。マイクロサービス８０４はその代わりに、データベースユーザというコンテキストにおいてデータベース８０６への実際の物理接続を提供する接続プール８１０を使用する。
【０１０８】

一般的に、接続は、基礎をなすドライバまたはプロバイダに接続ストリングを提供することによって構築される。接続ストリングは、特定のデータベースまたはサーバをアドレス指定するために、かつ、インスタンスおよびユーザ認証クレデンシャルを与えるために使用される（たとえば「Server=sql_box;Database=Common;User ID=uid;Pwd=password;」）。接続は、一旦構築されると、開閉が可能であり、プロパティ（たとえばコマンドタイムアウト長さ、または存在するのであればトランザクション）を設定することができる。接続ストリングは、データプロバイダのデータアクセスインターフェイスによって指示されるキーと値とのペアのセットを含む。接続プールは、データベースに対する未来の要求が必要なときに接続を再使用できるように保持されるデータベース接続のキャッシュである。接続プーリングにおいて、接続は、作成後にプールに置かれ、新たな接続を確立しなくてもよいように、再使用される。たとえば、マイクロサービス８０４とデータベース８０８との間に１０の接続が必要な場合、接続プール８１０には、すべてデータベースユーザというコンテキストにおいて（たとえば特定のデータベースユーザに関連して、たとえば、誰がこの接続の所有者か、誰のクレデンシャルが検証中なのか、それはデータベースユーザか、それはシステムクレデンシャルかなどに関連して）開いている１０の接続があるであろう。

【０１０９】

接続プール８１０内の接続は、何にでもアクセスできるシステムユーザのために作成される。したがって、テナントに代わって要求を処理するマイクロサービス８０４による監査および特権を正しく扱うために、データベース動作は、特定のテナントに割当てられたスキーマ所有者に関連する「プロキシユーザ」８１２というコンテキストで実行される。このスキーマ所有者は、このスキーマ作成の目的であったテナンシーにのみアクセスでき、このテナンシーの値はこのスキーマ所有者の値である。データベース８０６内のデータを求める要求がなされると、マイクロサービス８０４は、接続プール８１０内の接続を用いてこのデータを提供する。したがって、マルチテナンシーは、リソーステナンシーに対応付けられたデータストアプロキシユーザというコンテキストにおいて（たとえばそれに関連して）作成されたデータ接続のトップにある要求ごとに構築された特定テナント向けデータストアバインディングというコンテキストにおいて（たとえばそれに関連して）入ってくる要求を処理するステートレスでエラスティックな中間層サービスを持つことによって得られ、データベースは、サービスとは無関係にスケーリングできる。

【０１１０】

以下は、プロキシユーザ８１２を実現するための機能の例を提供する。

【０１１１】

【数１】

```
dbOperation = <prepare DB command to execute>
dbConnection = getDBConnectionFromPool()
dbConnection.setProxyUser (resourceTenant)
result = dbConnection.executeOperation (dbOperation)
```

【０１１２】

この機能において、マイクロサービス８０４は、接続プール８１０内のデータベース接続を使用する一方で、接続プール８１０から引出された接続に対する「プロキシユーザ（Proxy User）」設定を、「テナント（Tenant）」にセットし、テナントというコンテキストにおいてデータオペレーションを実行する。

【０１１３】

すべてのテーブルをストライピングすることにより同じデータベースにおいて異なるテナント用に異なるコラムを構成するとき、１つのテーブルは、混合されたすべてのテナントのデータを含み得る。これに対し、一実施形態は、テナント駆動のデータ層を提供する。

本実施形態は、異なるテナント用に同一データベースをストライピングするのではなく、テナントごとに異なる物理データベースを提供する。たとえば、マルチテナンシーは、プラガブルデータベース（たとえばオラクル社のOracle Database１２ｃ）を用いて実現されてもよく、この場合、各テナントには別々のパーティションが割当てられる。データ層では、リソースマネージャが要求を処理し、その後、その要求のデータソースを求める（メタデータとは別）。本実施形態は、要求ごとに各データソース／ストアへのランタイムスイッチを実行する。各テナントのデータをその他のテナントから分離することにより、本実施形態は改善されたデータセキュリティを提供する。

【０１１４】

一実施形態において、互いに異なるトークンは、異なるテナンシーをコーディファイする。ＵＲＬトークンは、サービスを要求するアプリケーションのテナンシーを特定し得る。アイデンティティトークンは、認証すべきユーザのアイデンティティをコーディファイし得る。アクセストークンは複数のテナンシーを特定し得る。たとえば、アクセストークンは、このようなアクセスのターゲットであるテナンシー（たとえばアプリケーションテナンシー）と、アクセス権が付与されたユーザのユーザテナンシーとをコーディファイし得る。クライアントアサーショントークンは、クライアントＩＤおよびクライアントテナンシーを特定し得る。ユーザアサーショントークンは、ユーザおよびユーザテナンシーを特定し得る。

【０１１５】

一実施形態において、アイデンティティトークンは、ユーザテナント名（すなわちユーザがどこに存在しているか）を示す少なくとも「クレーム（claim）」（セキュリティ分野の当業者が使用する）を含む。認可トークンに関連する「クレーム」は、ある主体が自身についてまたは別の主体について述べるステートメントである。ステートメントは、たとえば、名称、アイデンティ、キー、グループ、特権、または機能に関するものであってもよい。クレームは、プロバイダによって発行され、１つ以上の値が与えられた後に、セキュリティトークンサービス（security token service：「ＳＴＳ」）として一般に知られている発行者によって発行されたセキュリティトークンにパッケージングされる。

【０１１６】

一実施形態において、アクセストークンは、少なくとも、アクセストークンを求める要求がなされた時点のリソーステナント名（たとえば顧客）を示すクレーム／ステートメントと、ユーザテナント名を示すクレームと、要求しているＯＡｕｔｈクライアントの名を示すクレームと、クライアントテナント名を示すクレームとを含む。一実施形態において、アクセストークンは、以下のＪＳＯＮ機能に従って実現されてもよい。

【０１１７】

【数２】

```
{
   …
   " tok_type " : "AT",
   "user_id" : "testuser",
   "user_tenantname" : "<value-of-identity-tenant>"
   “tenant” : “<value-of-resource-tenant>”
   “client_id” : “testclient”,
   “client_tenantname”: “<value-of-client-tenant>”
  …
}
```

【０１１８】

一実施形態において、クライアントアサーショントークンは、少なくとも、クライアントテナント名を示すクレームと、要求を出しているＯＡｕｔｈクライアントの名前を示すクレームとを含む。

【０１１９】

本明細書に記載のトークンおよび／または複数のテナンシーは、ＩＤＣＳ以外のマルチテナントのクラウドベースのサービスによって実現されてもよい。たとえば、本明細書に記載のトークンおよび／または複数のテナンシーは、ＳａａＳまたは企業リソースプランニング（Enterprise Resource Planning：「ＥＲＰ」）サービスにおいて実現されてもよい。

【０１２０】

図９は、一実施形態におけるＩＤＣＳのネットワークビュー９００のブロック図である。図９は、一実施形態においてアプリケーション「ゾーン」９０４間で行なわれるネットワーク対話を示す。アプリケーションは、要求される保護レベルと、その他さまざまなシステムへの接続の実現に基づいてゾーンに分割される（たとえばＳＳＬゾーン、ｎｏ　ＳＳＬゾーンなど）。アプリケーションゾーンのうち、いくつかはＩＤＣＳ内部からのアクセスを要するサービスを提供するアプリケーションゾーンであり、いくつかはＩＤＣＳ外部からのアクセスを要するサービスを提供するアプリケーションゾーンであり、いくつかはオープンアクセスである。したがって、各保護レベルは各ゾーンに対して強化される。

【０１２１】

図９の実施形態において、サービス間の通信は、ＨＴＴＰ要求を用いて行なわれる。一実施形態において、ＩＤＣＳは、本明細書に記載のアクセストークンを用いて、サービスを提供するだけでなく、ＩＤＣＳへのアクセスおよびＩＤＣＳ自身の内部におけるアクセスを安全なものにする。一実施形態において、ＩＤＣＳマイクロサービスは、ＲＥＳＴｆｕｌインターフェイスを通してエクスポーズされ、本明細書に記載のトークンによって安全なものにされる。

【０１２２】

図９の実施形態において、さまざまなアプリケーション／サービス９０２のうちのいずれか１つが、ＩＤＣＳ　ＡＰＩに対してＨＴＴＰコールすることにより、ＩＤＣＳサービスを使用してもよい。一実施形態において、アプリケーション／サービス９０２のＨＴＴＰ要求は、オラクルパブリッククラウドロードバランシング外部仮想ＩＰアドレス（「ＶＩＰ」）９０６（またはその他同様の技術）、パブリッククラウドウェブルーティング層９０８、およびＩＤＣＳロードバランシング内部ＶＩＰアプライアンス９１０（またはその他同様の技術）を通って、ＩＤＣＳウェブルーティング層９１２により受信されてもよい。ＩＤＣＳウェブルーティング層９１２は、ＩＤＣＳの外部または内部からの要求を受信し、ＩＤＣＳプラットフォームサービス層９１４またはＩＤＣＳインフラストラクチャサービス層９１６を通してルーティングする。ＩＤＣＳプラットフォームサービス層９１４は、ＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ、ＯＡｕｔｈ、ＳＡＭＬ，ＳＣＩＭなどのＩＤＣＳの外部から呼出されたＩＤＣＳマイクロサービスを含む。ＩＤＣＳインフラストラクチャサービス層９１６は、その他のＩＤＣＳマイクロサービスの機能をサポートするためにＩＤＣＳの内部から呼出されたサポートマイクロサービスを含む。ＩＤＣＳインフラストラクチャマイクロサービスの例は、ＵＩ、ＳＳＯ、レポート、キャッシュ、ジョブスケジューラ、サービスマネージャ、キーを作るための機能などである。ＩＤＣＳキャッシュ層９２６は、ＩＤＣＳプラットフォームサービス層９１４およびＩＤＣＳインフラストラクチャサービス層９１６のためのキャッシング機能をサポートする。

【０１２３】

ＩＤＣＳへの外部アクセスおよびＩＤＣＳ内部アクセス双方のセキュリティを強化することにより、ＩＤＣＳの顧客に、それが実行するアプリケーションのための傑出したセキュリティコンプライアンスを与えることができる。

【０１２４】

図９の実施形態において、構造化照会言語（Structured Query Language：「ＳＱＬ」

）に基づいて通信するデータ層９１８およびＬＤＡＰに基づいて通信するＩＤストア層９２０以外については、ＯＡｕｔｈプロトコルを使用することにより、ＩＤＣＳ内のＩＤＣＳコンポーネント（たとえばマイクロサービス）間の通信を保護し、ＩＤＣＳ外部からのアクセスを安全なものにするために使用される同じトークンをＩＤＣＳ内のセキュリティのためにも使用する。すなわち、ウェブルーティング層９１２は、要求がＩＤＣＳの外部から受けたものであろうとＩＤＣＳの内部から受けたものであろうと、受信した要求を処理するための同じトークンおよびプロトコルを使用する。したがって、ＩＤＣＳは、システム全体を保護するために１つの一貫したセキュリティモデルを提供することにより、傑出したセキュリティコンプライアンスを可能にする。なぜなら、システム内に実現されるセキュリティモデルが少ないほど、システムの安全性は高くなるからである。

【０１２５】

ＩＤＣＳクラウド環境において、アプリケーションは、ネットワークコールを行なうことによって通信する。ネットワークコールは、ＨＴＴＰ、伝送制御プロトコル（Transmission Control Protocol：「ＴＣＰ」）、ユーザデータグラムプロトコル（User Datagram Protocol：「ＵＤＰ」）などの適用可能なネットワークプロトコルに基づいていればよい。たとえば、アプリケーション「Ｘ」は、アプリケーション「Ｙ」と、ＨＴＴＰに基づいて、アプリケーション「Ｙ」をＨＴＴＰユニフォーム・リソース・ロケータ（Uniform Resource Locator：「ＵＲＬ」）としてエクスポーズすることにより、通信し得る。一実施形態において、「Ｙ」は、各々がある機能に対応する多数のリソースをエクスポーズするＩＤＣＳマイクロサービスである。「Ｘ」（たとえば別のＩＤＣＳマイクロサービス）は、「Ｙ」をコールする必要があるとき、「Ｙ」と、呼出す必要があるリソース／機能とを含むＵＲＬを構成し（たとえばhttps:/host/Y/resource）、ウェブルーティング層９１２を通って「Ｙ」に導かれる対応するＲＥＳＴコールを行なう。

【０１２６】

一実施形態において、ＩＤＣＳ外部の呼出元は、「Ｙ」がどこにあるかを知る必要がない場合があるが、ウェブルーティング層９１２はアプリケーション「Ｙ」がどこで走っているかを知る必要がある。一実施形態において、ＩＤＣＳは、発見機能を実現する（ＯＡｕｔｈサービスによって実現される）ことにより、各アプリケーションがどこで走っているかを判断し、スタティックなルーティング情報の可用性が必要ではなくなるようにする。

【０１２７】

一実施形態において、企業マネージャ（enterprise manager：「ＥＭ」）９２２は、オンプレミスおよびクラウドベース管理をＩＤＣＳに拡張する「一枚のガラス」を提供する。一実施形態において、Chef Software社の構成管理ツールである「シェフ（Chef）」サーバ９２４は、さまざまなＩＤＣＳ層のための構成管理機能を提供する。一実施形態において、サービスデプロイメントインフラストラクチャおよび／または永続格納モジュール９２８は、テナントライフサイクル管理動作、パブリッククラウドライフサイクル管理動作、またはその他の動作のために、ＯＡｕｔｈ２ ＨＴＴＰメッセージをＩＤＣＳウェブルーティング層９１２に送信してもよい。一実施形態において、ＩＤＣＳインフラストラクチャサービス層９１６は、ＩＤ／パスワードＨＴＴＰメッセージを、パブリッククラウド通知サービス９３０またはパブリッククラウドストレージサービス９３２に送信してもよい。

【０１２８】

<u>クラウドアクセス制御 - ＳＳＯ</u>

一実施形態は、クラウドスケールＳＳＯサービスを実現するために軽量クラウド標準をサポートする。軽量クラウド標準の例としては、ＨＴＴＰ、ＲＥＳＴ、および、ブラウザを通してアクセスを提供する標準（ウェブブラウザは軽量であるため）が挙げられる。逆に、ＳＯＡＰは、クライアントを構築するためにより多くの管理、構成、およびツールを必要とする重いクラウド標準の一例である。本実施形態は、アプリケーションのためにＯｐｅｎＩＤ Ｃｏｎｎｅｃｔセマンティクスを使用することにより、ＩＤＣＳに対してユーザ認証を要求する。本実施形態は、軽量ＨＴＴＰクッキーベースのユーザセッション追跡

を用いて、ステートフルなサーバ側セッションサポートなしで、ＩＤＣＳにおけるユーザのアクティブなセッションを追跡する。本実施形態は、使用するアプリケーションに対して、認証されたアイデンティティを自身のローカルセッションに戻すマッピングを行なうときに、ＪＷＴベースのアイデンティティトークンを使用する。本実施形態は、連携されているアイデンティティ管理システムとの統合をサポートし、ＩＤＣＳに対してユーザ認証を要求するために企業デプロイメントのＳＡＭＬ　ＩＤＰサポートをエクスポーズする。

【０１２９】

図１０は、一実施形態におけるＩＤＣＳ内のＳＳＯ機能のシステムアーキテクチャビューのブロック図１０００である。本実施形態は、クライアントアプリケーションが標準ベースのウェブプロトコルを推進してユーザ認証フローを開始することを可能にする。クラウドシステムとＳＳＯの統合を要求するアプリケーションは、企業データセンターにあってもよく、遠隔パートナーデータセンターにあってもよく、またはオンプレミスの顧客によって操作されてもよい。一実施形態において、異なるＩＤＣＳプラットフォームサービスが、接続されているネイティブなアプリケーション（すなわちＩＤＣＳと統合するためにＯｐｅｎＩＤ接続を利用するアプリケーション）からのログイン／ログアウト要求を処理するためのＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ、接続されているアプリケーションからのブラウザベースのログイン／ログアウト要求を処理するためのＳＡＭＬ　ＩＤＰサービス、外部ＳＡＭＬ　ＩＤＰに対してユーザ認証を調整するためのＳＡＭＬ　ＳＰサービス、および、ローカルなまたは連携されたログインフローを含みＩＤＣＳホストセッションクッキーを管理するためのエンドユーザログインセレモニーを調整するための内部ＩＤＣＳ　ＳＳＯサービスなどの、ＳＳＯのビジネスを実現する。一般的に、ＨＴＴＰは、フォームありでまたはフォームなしで機能する。フォームありで機能するとき、このフォームはブラウザ内で見えるフォームである。フォームなしで機能するとき、これはクライアントからサーバへの通信として機能する。ＯｐｅｎＩＤ　ＣｏｎｎｅｃｔもＳＡＭＬも、フォームをレンダリングする能力を必要とするが、これは、ブラウザの存在によって実現される、または、ブラウザが存在しているかのように機能するアプリケーションによって仮想的に実行される。一実施形態において、ユーザ認証／ＳＳＯをＩＤＣＳを通して実現するアプリケーションクライアントは、ＩＤＣＳにおいて、ＯＡｕｔｈ２クライアントとして登録される必要があり、クライアント識別子およびクレデンシャル（たとえばＩＤ／パスワード、ＩＤ／証明書など）を取得する必要がある。

【０１３０】

図１０の実施形態の例は、２つのプラットフォームマイクロサービスとしてのＯＡｕｔｈ２　１００４およびＳＡＭＬ２　１００６と、１つのインフラストラクチャマイクロサービスとしてのＳＳＯ１００８とを含む、ログイン機能をまとめて提供する３つのコンポーネント／マイクロサービスを含む。図１０の実施形態において、ＩＤＣＳは「アイデンティティメタシステム」を提供する。このメタシステムにおいて、ＳＳＯサービス１００８は、異なる種類のアプリケーションに対して提供される。これらのアプリケーションは、３者間ＯＡｕｔｈフローを必要としＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔリレーパーティ（relaying party：「ＲＰ」、そのユーザ認証機能をＩＤＰにアウトソーシングするアプリケーション）として機能するブラウザベースのウェブまたはネイティブアプリケーション１０１０、２者間ＯＡｕｔｈフローを必要としＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔ　ＲＰとして機能するネイティブアプリケーション１０１１、およびＳＡＭＬ　ＳＰとして機能するウェブアプリケーション１０１２などである。

【０１３１】

一般的に、アイデンティティメタシステムは、デジタルアイデンティティのための相互運用可能なアーキテクチャであり、複数の基礎となる技術、実装、およびプロバイダの集合体を用いることを可能にする。ＬＤＡＰ、ＳＡＭＬ、およびＯＡｕｔｈは、アイデンティティ機能を提供する異なるセキュリティ標準の例であり、アプリケーションを構築するための基礎となることが可能であり、アイデンティティメタシステムは、このようなアプリケーションに対して統一されたセキュリティシステムを提供するように構成されてもよい

。ＬＤＡＰセキュリティモデルは、アイデンティティを扱うための特定のメカニズムを指定し、システムを通るすべてのパスは厳密に保護されねばならない。ＳＡＭＬは、一組のアプリケーションが、異なるセキュリティドメインの異なる組織に属する別の一組のアプリケーションとの間で安全に情報を交換できるようにするために開発されたものである。これら２つのアプリケーションの間に信頼はないので、ＳＡＭＬは、一方のアプリケーションが、同じ組織に属していない別のアプリケーションを認証できるように開発された。ＯＡｕｔｈは、ウェブベースの認証を実行するための軽量プロトコルであるＯｐｅｎＩＤＣｏｎｎｅｃｔを提供する。

【０１３２】

図１０の実施形態において、ＯｐｅｎＩＤアプリケーション１０１０がＩＤＣＳ内のＯｐｅｎＩＤサーバに接続すると、その「チャネル」はＳＳＯサービスを要求する。同様に、ＳＡＭＬアプリケーション１０１２がＩＤＣＳ内のＳＡＭＬサーバに接続すると、その「チャネル」もＳＳＯサービスを要求する。ＩＤＣＳにおいて、各マイクロサービス（たとえばＯｐｅｎＩＤマイクロサービス１００４およびＳＡＭＬマイクロサービス１００６）はアプリケーション各々を処理し、これらのマイクロサービスはＳＳＯマイクロサービス１００８からのＳＳＯ機能を要求する。プロトコルごとにマイクロサービスを追加してからＳＳＯ機能のためにＳＳＯマイクロサービス１００８を用いることにより、このアーキテクチャを拡張して任意の数のその他のセキュリティプロトコルをサポートすることができる。ＳＳＯマイクロサービス１００８は、セッションを発行し（すなわちＳＳＯクッキー１０１４が提供される）、このアーキテクチャにおいてセッションを発行する権限を有する唯一のシステムである。ＩＤＣＳセッションは、ブラウザ１００２がＳＳＯクッキー１０１４を使用することによって実現される。ブラウザ１００２はまた、ローカルセッションクッキー１０１６を用いてそのローカルセッションを管理する。

【０１３３】

一実施形態において、たとえば、ブラウザ内で、ユーザは、ＳＡＭＬに基づいて第１のアプリケーションを使用してログインし、その後、ＯＡｕｔｈなどの異なるプロトコルを用いて構築された第２のアプリケーションを使用してもよい。ユーザには、同じブラウザ内の第２のアプリケーション上のＳＳＯが与えられる。したがって、ブラウザは、ステートまたはユーザエージェントであり、クッキーを管理する。

【０１３４】

一実施形態において、ＳＳＯマイクロサービス１００８は、ログインセレモニー１０１８、ＩＤ／パスワードリカバリ１０２０、第１回ログインフロー１０２２、認証マネージャ１０２４、ＨＴＴＰクッキーマネージャ１０２６、およびイベントマネージャ１０２８を提供する。ログインセレモニー１０１８は、顧客設定および／またはアプリケーションコンテキストに基づいてＳＳＯ機能を実現し、ローカルフォーム（たとえばベーシックＡｕｔｈ）、外部ＳＡＭＬ　ＩＤＰ、外部ＯＩＤＣ　ＩＤＰなどに従って構成されてもよい。ＩＤ／パスワードリカバリ１０２０は、ユーザのＩＤおよび／またはパスワードの回復のために使用される。第１回ログインフロー１０２２は、ユーザが１回目にログインしたときに実現される（すなわちＳＳＯセッションはまだ存在しない）。認証マネージャ１０２４は、認証に成功すると認証トークンを発行する。ＨＴＴＰクッキーマネージャ１０２６は認証トークンをＳＳＯクッキーに保存する。イベントマネージャ１０２８はＳＳＯ機能に関連するイベントをパブリッシュする。

【０１３５】

一実施形態において、ＯＡｕｔｈマイクロサービス１００４とＳＳＯマイクロサービス１００８との間の対話は、ブラウザリダイレクトに基づいており、ＳＳＯマイクロサービス１００８は、ＨＴＭＬフォームを用いてユーザにチャレンジし、クレデンシャルを検証し、セッションクッキーを発行する。

【０１３６】

一実施形態において、たとえば、ＯＡｕｔｈマイクロサービス１００４は、ブラウザ１００２から認証要求を受け、３者間ＯＡｕｔｈフローに従ってアプリケーションのユーザを

認証する。よって、ＯＡｕｔｈマイクロサービス１００４は、ＯＩＤＣプロバイダ１０３０として機能し、ブラウザ１００２をＳＳＯマイクロサービス１００８にリダイレクトし、アプリケーションコンテキストに沿って進む。ユーザが有効なＳＳＯセッションを有するか否かに応じて、ＳＳＯマイクロサービス１００８は、既存のセッションを検証するかまたはログインセレモニーを実行する。認証または検証に成功すると、ＳＳＯマイクロサービス１００８は、認証コンテキストをＯＡｕｔｈマイクロサービス１００４に返す。そうすると、ＯＡｕｔｈマイクロサービス１００４はブラウザ１００２を認証（「ＡＺ」コードを有するコールバックＵＲＬにリダイレクトする。ブラウザ１００２は、ＡＺコードをＯＡｕｔｈマイクロサービス１００４に送信し、必要なトークン１０３２を要求する。また、ブラウザ１００２は、ＨＴＴＰ認証ヘッダにおいてそのクライアントクレデンシャル（ＩＤＣＳをＯＡｕｔｈ２クライアントとして登録したときに取得）を含む。これに対し、ＯＡｕｔｈマイクロサービス１００４は、要求されたトークン１０３２をブラウザ１００２に与える。一実施形態において、ブラウザ１００２に与えられるトークン１０３２は、ＪＷアイデンティティと、ＩＤＣＳ　ＯＡｕｔｈ２サーバによって署名されたアクセストークンとを含む。この機能のさらなる詳細は、以下で図１１を参照しながら開示される。

【０１３７】

一実施形態において、たとえば、ＯＡｕｔｈマイクロサービス１００４は、ネイティブアプリケーション１０１１から認可要求を受け、２者間ＯＡｕｔｈフローに従ってユーザを認証する。この場合、ＯＡｕｔｈマイクロサービス１００４の認証マネージャ１０３４は対応する認証を（たとえばクライアント１０１１から受けたＩＤ／パスワードに基づいて）実行し、トークンマネージャ１０３６は、認証に成功すると、対応するアクセストークンを発行する。

【０１３８】

一実施形態において、たとえば、ＳＡＭＬマイクロサービス１００６は、ブラウザからＳＳＯ　ＰＯＳＴ要求を受け、ＳＡＭＬ　ＳＰとして機能するウェブアプリケーション１０１２のユーザを認証する。ＳＡＭＬマイクロサービス１００６は次に、ＳＡＭＬ　ＩＤＰ１０３８として機能し、ブラウザ１００２をＳＳＯマイクロサービス１００８にリダイレクトし、アプリケーションコンテキストに沿って進む。ユーザが有効なＳＳＯセッションを有しているか否かに応じて、ＳＳＯマイクロサービス１００８は、既存のセッションを検証するか、またはログインセレモニーを実行する。認証または検証に成功すると、ＳＳＯマイクロサービス１００８は、認証コンテキストをＳＡＭＬマイクロサービス１００６に返す。そうすると、ＳＡＭＬマイクロサービスは、必要なトークンでＳＰにリダイレクトする。

【０１３９】

一実施形態において、たとえば、ＳＡＭＬマイクロサービス１００６は、ＳＡＭＬ　ＳＰ１０４０として機能してもよく、遠隔ＳＡＭＬ　ＩＤＰ１０４２（たとえばアクティブディレクトリ連携サービス（active directory federation service：「ＡＤＦＳ」）に進んでもよい。一実施形態は、標準ＳＡＭＬ／ＡＤフローを実現する。一実施形態において、ＳＡＭＬマイクロサービス１００６とＳＳＯマイクロサービス１００８との間の対話は、ブラウザのリダイレクトに基づいており、ＳＳＯマイクロサービス１００８は、ＨＴＭＬフォームを用いてユーザにチャレンジし、クレデンシャルを検証し、セッションクッキーを発行する。

【０１４０】

一実施形態において、ＩＤＣＳ内部のコンポーネント（たとえば１００４、１００６、１００８）と、ＩＤＣＳ外部のコンポーネント（たとえば１００２、１０１１、１０４２）との間の対話は、ファイアウォール１０４４を通して行なわれる。

【０１４１】

ログイン／ログアウトフロー

図１１は、一実施形態における、ＩＤＣＳによって提供されるＳＳＯ機能のメッセージシ

ーケンスフロー１１００である。ユーザがブラウザ１１０２を用いてクライアント１１０６（たとえばブラウザベースのアプリケーションまたはモバイル／ネイティブアプリケーション）にアクセスするとき、クラウドゲート１１０４は、アプリケーション施行点として機能し、ローカルポリシーテキストファイルに規定されているポリシーを施行する。クラウドゲート１１０４は、ユーザがローカルアプリケーションセッションを有していないことを検出した場合、ユーザの認証を要求する。そうするために、クラウドゲート１１０４は、ブラウザ１１０２をＯＡｕｔｈ２マイクロサービス１１１０にリダイレクトすることにより、ＯＡｕｔｈ２マイクロサービス１１１０に対するＯｐｅｎＩＤ Ｃｏｎｎｅｃｔログインフローを開始する（３者間ＡＺ Ｇｒａｎｔフローであり、範囲＝「openid profile」）。

【０１４２】

ブラウザ１１０２の要求は、ＩＤＣＳルーティング層ウェブサービス１１０８およびクラウドゲート１１０４を横断してＯＡｕｔｈ２マイクロサービス１１１０に到達する。ＯＡｕｔｈ２マイクロサービス１１１０は、アプリケーションコンテキスト（すなわちアプリケーションを記述するメタデータ、たとえば接続するアプリケーションのアイデンティティ、クライアントＩＤ、構成、アプリケーションは何ができるかなど）を構成し、ブラウザ１１０２をログインのためにＳＳＯマイクロサービス１１１２にリダイレクトする。

【０１４３】

ユーザが有効なＳＳＯセッションを有する場合、ＳＳＯマイクロサービス１１１２は、ログインセレモニーを開始することなく既存のセッションを検証する。ユーザが有効なＳＳＯセッションを有していない場合（すなわちセッションクッキーが存在しない）、ＳＳＯマイクロサービス１１１２は、顧客のログインプリファレンスに従ってユーザログインセレモニーを開始する（たとえば商標付ログインページを表示する）。そうするために、ＳＳＯマイクロサービス１１１２は、ブラウザ１１０２を、JavaScriptで実現されるログインアプリケーションサービス１１１４にリダイレクトする。ログインアプリケーションサービス１１１４はブラウザ１１０２にログインページを提供する。ブラウザ１１０２はログインクレデンシャルを含むＲＥＳＴ ＰＯＳＴをＳＳＯマイクロサービス１１１２に送信する。ＳＳＯマイクロサービス１１１２は、アクセストークンを生成し、ＲＥＳＴ ＰＯＳＴのクラウドゲート１１０４に送信する。クラウドゲート１１０４は、認証情報を管理ＳＣＩＭマイクロサービス１１１６に送信することによりユーザのパスワードを検証する。管理ＳＣＩＭマイクロサービス１１１６は、認証が成功したと判断し、対応するメッセージをＳＳＯマイクロサービス１１１２に送信する。

【０１４４】

一実施形態において、ログインセレモニー中、ログインページは同意ページを表示しない。「ログイン」オペレーションはさらなる同意を要しないからである。代わりに、アプリケーションに対してエクスポーズされている特定のプロファイル属性についてユーザに知らせるプライバシーポリシーが、ログインページ上に記載される。ログインセレモニー中、ＳＳＯマイクロサービス１１１２は顧客のＩＤＰプリファレンスを尊重し、構成され次第、構成されたＩＤＰに対する認証のためにＩＤＰにリダイレクトする。

【０１４５】

認証または検証が成功すると、ＳＳＯマイクロサービス１１１２は、ブラウザ１１０２を、ユーザの認証トークンを含む、新たに作成／更新されたＳＳＯホストＨＴＴＰクッキー（たとえば「ＨＯＳＴＵＲＬ」が示すホストのコンテキストで作成されたクッキー）を用いて、ＯＡｕｔｈ２マイクロサービス１１１０に戻るようにブラウザ１１０２をリダイレクトする。ＯＡｕｔｈ２マイクロサービス１１１０は、ＡＺコード（たとえばＯＡｕｔｈコンセプト）をブラウザ１１０２に戻しクラウドゲート１１０４にリダイレクトする。ブラウザ１１０２はＡＺコードをクラウドゲート１１０４に送信し、クラウドゲート１１０４はＲＥＳＴ ＰＯＳＴをＯＡｕｔｈ２マイクロサービス１１１０に送信してアクセストークンおよびアイデンティティトークンを要求する。これらのトークンはどちらも、ＯＡｕｔｈマイクロサービス１１１０にスコーピングされる（オーディエンストークンクレー

ムによって示される）。クラウドゲート１１０４はこれらのトークンをＯＡｕｔｈ２マイクロサービス１１１０から受ける。

【０１４６】

クラウドゲート１１０４は、アイデンティティトークンを用いて、認証されたユーザのアイデンティティをその内部アカウント表現にマッピングし、これは、このマッピングを自身のＨＴＴＰクッキーに保存してもよい。クラウドゲート１１０４は次に、ブラウザ１１０２をクライアント１１０６にリダイレクトする。すると、ブラウザ１１０２は、クライアント１１０６に到達し、対応するレスポンスをクライアント１１０６から受ける。この時点以降、ブラウザ１１０２は、アプリケーションのローカルクッキーが有効である限り、アプリケーション（すなわちクライアント１１０６）にシームレスにアクセスすることができる。ローカルクッキーが無効になると、認証プロセスは繰返される。

【０１４７】

クラウドゲート１１０４はさらに、要求に含められたアクセストークンを用いて、「userinfo」をＯＡｕｔｈ２マイクロサービス１１１０からまたはＳＣＩＭマイクロサービスから取得する。このアクセストークンは、「プロファイル」スコープによって与えられる属性の「userinfo」リソースにアクセスするには十分である。これは、ＳＣＩＭマイクロサービスを介して「／ｍｅ」リソースにアクセスするのにも十分である。一実施形態において、デフォルトで、含まれているアクセストークンは、「プロファイル」スコープの下で与えられるユーザプロファイル属性に対してのみ十分である。他のプロファイル属性へのアクセスは、クラウドゲート１１０４によって発行されたＡＺグラントログイン要求において提示された追加の（任意の）スコープに基づいて認可される。

【０１４８】

ユーザがＯＡｕｔｈ２が統合された別のアプリケーションにアクセスする場合、同じプロセスが繰返される。

【０１４９】

一実施形態において、ＳＳＯ統合アーキテクチャは、ブラウザベースのユーザログアウトに対し、同様のＯｐｅｎＩＤ　Ｃｏｎｎｅｃｔユーザ認証フローを使用する。一実施形態において、既存のアプリケーションセッションを有するユーザは、クラウドゲート１１０４にアクセスしてログアウトを開始する。その代わりに、ユーザは、ＩＤＣＳ側でログアウトを開始している場合がある。クラウドゲート１１０４は、特定用途向けのユーザセッションを終了し、ＯＡｕｔｈ２マイクロサービス１１１０に対しＯＡｕｔｈ２　ＯｐｅｎＩＤ　プロバイダ（「ＯＰ」）ログアウト要求を開始する。ＯＡｕｔｈ２マイクロサービス１１１０は、ユーザのホストＳＳＯクッキーを削除するＳＳＯマイクロサービス１１１２にリダイレクトする。ＳＳＯマイクロサービス１１１２は、ユーザのＳＳＯクッキーにおいて追跡された既知のログアウトエンドポイントに対し一組のリダイレクト（ＯＡｕｔｈ２　ＯＰおよびＳＡＭＬ　ＩＤＰ）を開始する。

【０１５０】

一実施形態において、クラウドゲート１１０４がＳＡＭＬプロトコルを用いてユーザ認証（たとえばログイン）を要求する場合、同様のプロセスが、ＳＡＭＬマイクロサービスとＳＳＯマイクロサービス１１１２との間で開始される。

【０１５１】

<u>クラウドキャッシュ</u>

一実施形態は、クラウドキャッシュと呼ばれるサービス／機能を提供する。クラウドキャッシュは、ＩＤＣＳに与えられて、ＬＤＡＰベースのアプリケーション（たとえば電子メールサーバ、カレンダーサーバー、何らかのビジネスアプリケーションなど）との通信をサポートする。なぜなら、ＩＤＣＳはＬＤＡＰに従って通信するのではないが、このようなアプリケーションはＬＤＡＰに基づいてのみ通信するように構成されているからである。典型的には、クラウドディレクトリは、ＲＥＳＴ　ＡＰＩを介してエクスポーズされ、ＬＤＡＰプロトコルに従って通信するのではない。一般的に、企業ファイアウォオールを通してＬＤＡＰ接続を管理するには、セットアップおよび管理が難しい特殊な構成が必要

である。
【０１５２】
ＬＤＡＰベースのアプリケーションをサポートするために、クラウドキャッシュは、ＬＤＡＰ通信を、クラウドシステムとの通信に適したプロトコルに変換する。一般的に、ＬＤＡＰベースのアプリケーションは、ＬＤＡＰを介してデータベースを使用する。代わりに、アプリケーションは、ＳＱＬのような異なるプロトコルを介してデータベースを使用するように構成されてもよい。しかしながら、ＬＤＡＰはツリー構造のリソースの階層表現を提供するのに対し、ＳＱＬはデータをテーブルとフィールドとして表現する。したがって、ＬＤＡＰは検索機能用であることがより望ましいであろう。一方、ＳＱＬはトランザクション機能用であることがより望ましいであろう。
【０１５３】
一実施形態において、ＩＤＣＳが提供するサービスを、ＬＤＡＰベースのアプリケーションで使用して、たとえば、アプリケーションのユーザを認証する（すなわちアイデンティティサービス）、またはアプリケーションのセキュリティポリシーを施行する（すなわちセキュリティサービス）ことができる。一実施形態において、ＩＤＣＳとのインターフェイスは、ファイアウォールを通り、ＨＴＴＰ（たとえばＲＥＳＴ）に基づく。典型的に、企業ファイアウォールは、内部ＬＤＡＰ通信へのアクセスを、当該通信がセキュア・ソケット・レイヤ（Secure Sockets Layer：「ＳＳＬ」）を実現する場合であっても許可しない。また、企業ファイアウォールは、ＴＣＰポートがファイアウォールを通してエクスポーズされることを許可しない。しかしながら、クラウドキャッシュは、ＬＤＡＰとＨＴＴＰとの間の変換を行なって、ＬＤＡＰベースのアプリケーションが、ＩＤＣＳが提供するサービスに到達できるようにし、ファイアウォールはＨＴＴＰに対してオープンである。
【０１５４】
一般的に、ＬＤＡＰディレクトリは、マーケティングおよび開発などのビジネスライン（line of business）で使用されてもよく、ユーザ、グループ、業務などを規定する。一例において、マーケティングおよび開発ビジネスは、多様な顧客を対象としている場合があり、顧客ごとに、独自のアプリケーション、ユーザ、グループ、業務などを有し得る。ＬＤＡＰキャッシュディレクトリを実行し得るビジネスラインの別の例は、無線サービスプロバイダである。この場合、無線サービスプロバイダのユーザが行なう各コールは、ＬＤＡＰディレクトリに対してユーザのデバイスを認証し、ＬＤＡＰディレクトリ内の対応する情報の一部は課金システムと同期させてもよい。これらの例において、ＬＤＡＰは、実行時に探索されるコンテンツを物理的に分離するための機能を提供する。
【０１５５】
一例において、無線サービスプロバイダは、短期マーケティングキャンペーンを支援するＩＤＣＳが提供するサービスを使用する一方で、自身のアイデンティティ管理サービスをそのコアビジネス（たとえば通常のコール）のために扱ってもよい。この場合、クラウドキャッシュは、ＬＤＡＰを、クラウドに対して実行する一組のユーザおよび一組のグループを有する場合は「平坦にする」。一実施形態において、ＩＤＣＳにおいて実現されるクラウドキャッシュの数はいくつであってもよい。
【０１５６】
<u>分散型データグリッド</u>
一実施形態において、ＩＤＣＳにおけるキャッシュクラスタは、たとえばその開示を本明細書に引用により援用する米国特許公開第２０１６／００９２５４０号に開示されている分散型データグリッドに基づいて実現される。分散型データグリッドは、分散環境またはクラスタ環境内で１つ以上のクラスタにおいてコンピュータサーバの集合体が、一緒に作業することにより情報を管理し計算などの関連動作を管理するシステムである。分散型データグリッドを用いることで、サーバ間で共有されるアプリケーションオブジェクトおよびデータを管理することができる。分散型データグリッドは、短いレスポンスタイム、高いスループット、予測可能なスケーラビリティ、継続的なアベイラビリティ、および情報の信頼性を提供する。具体的な例として、たとえばオラクル社のOracle Coherenceのデ

ータグリッドのような分散型データグリッドは、情報をインメモリに格納することによりさらに高いパフォーマンスを達成し、複数のサーバにわたって同期が取られた情報のコピーを保持するにあたって冗長性を用いることにより、サーバ故障イベント時におけるシステムの回復力とデータの継続的なアベイラビリティとを保証する。

【０１５７】

一実施形態において、ＩＤＣＳは、Coherenceなどの分散型データグリッドを実現して、すべてのマイクロサービスがブロックされることなく共有キャッシュオブジェクトへのアクセスを要求できるようにする。Coherenceは、従来のリレーショナルデータベース管理システムと比較して、より高い信頼性、スケーラビリティ、およびパフォーマンスが得られるように設計された、所有権を主張できるＪａｖａベースのインメモリデータグリッドである。Coherenceは、ピアトゥピア（すなわち中央マネージャがない）インメモリ分散型キャッシュを提供する。

【０１５８】

図１２は、データを格納しデータアクセス権をクライアント１２５０に与え本発明の実施形態を実現する分散型データグリッド１２００の一例を示す。「データグリッドクラスタ」または「分散型データグリッド」は、分散環境またはクラスタ環境内で１つ以上のクラスタ（たとえば１２００ａ、１２００ｂ、１２００ｃ）において一緒に作業することにより情報を格納し関連する計算などの動作を管理する複数のコンピュータサーバ（たとえば１２２０ａ、１２２０ｂ、１２２０ｃ、および１２２０ｄ）を含むシステムである。分散型データグリッド１２００は、クラスタ１２００ａにおいて５つのデータノード１２３０ａ、１２３０ｂ、１２３０ｃ、１２３０ｄ、および１２３０ｅとともに４つのサーバ１２２０ａ、１２２０ｂ、１２２０ｃ、１２２０ｄを含むものとして示されているが、分散型データグリッド１２００は、任意の数のクラスタおよび各クラスタにおける任意の数のサーバおよび／またはノードを含み得る。ある実施形態において、分散型データグリッド１２００は本発明を実現する。

【０１５９】

図１２に示されるように、分散型データグリッドは、一緒に作業する多数のサーバ（たとえば１２２０ａ、１２２０ｂ、１２２０ｃ、および１２２０ｄ）にデータを分散させることによってデータ格納および管理機能を提供する。データグリッドクラスタの各サーバは、たとえば、１つから２つのプロセッサソケットと１プロセッサソケット当たり２つから４つのＣＰＵコアとを有する「コモディティ（commodity）ｘ８６」サーバハードウェアプラットフォームのような、従来のコンピュータシステムであってもよい。各サーバ（たとえば１２２０ａ、１２２０ｂ、１２２０ｃ、および１２２０ｄ）は、１つ以上のＣＰＵと、ネットワークインターフェイスカード（Network Interface Card：「ＮＩＣ」）と、たとえば最小で４ＧＢのＲＡＭ最大で６４ＧＢ以上のＲＡＭを含むメモリとで構成されている。サーバ１２２０ａは、ＣＰＵ１２２２ａと、メモリ１２２４ａと、ＮＩＣ１２２６ａとを有するものとして示されている（これらの要素は他のサーバ１２２０ｂ、１２２０ｃ、１２２０ｄ上にもあるが図示されていない）。任意で、各サーバにフラッシュメモリ（たとえばＳＳＤ　１２２８ａ）を設けることで過剰な記憶容量を提供してもよい。提供時、ＳＳＤ容量は、好ましくはＲＡＭのサイズの１０倍である。データグリッドクラスタ１２００ａのサーバ（たとえば１２２０ａ、１２２０ｂ、１２２０ｃ、１２２０ｄ）は、高帯域幅のＮＩＣ（たとえばＰＣＩ－ＸまたはＰＣＩｅ）を用いて高性能ネットワークスイッチ１２２０（たとえばギガビット以上のイーサネット（登録商標））に接続されている。

【０１６０】

クラスタ１２００ａは、故障中にデータが失われる可能性を避けるために最小で４つの物理サーバを含むことが好ましいが、典型的な設備はより多くのサーバを有する。各クラスタに存在するサーバが多いほど、フェイルオーバーおよびフェイルバックの効率は高く、サーバの故障がクラスタに与える影響は小さくなる。サーバ間の通信時間を最短にするために、各データグリッドクラスタは、サーバ間の単一ホップ通信を提供する単一のスイッ

チ１２０２に限定されることが理想的である。このように、クラスタは、スイッチ１２０２上のポートの数によって制限される。したがって、典型的なクラスタは４～９６の物理サーバを含む。

【０１６１】

分散型データグリッド１２００のほとんどの広域ネットワーク（Wide Area Network：「ＷＡＮ」）構成において、ＷＡＮ内の各データセンターは、独立しているが相互に接続されているデータグリッドクラスタ（たとえば１２００ａ、１２００ｂ、および１２００ｃ）を有する。ＷＡＮは、たとえば図１２に示されるクラスタよりも多くのクラスタを含み得る。加えて、相互接続されているが独立しているクラスタ（たとえば１２００ａ、１２００ｂ、１２００ｃ）を用いることにより、および／または相互接続されているが独立しているクラスタを、互いに離れているデータセンター内に配置することにより、分散型データグリッドは、自然災害、火災、洪水、長期停電などによって生じる、１つのクラスタのすべてのサーバの同時損失を防止すべく、クライアント１２５０に対するデータおよびサービスを保証することができる。

【０１６２】

１つ以上のノード（たとえば１２３０ａ、１２３０ｂ、１２３０ｃ、１２３０ｄおよび１２３０ｅ）は、クラスタ１２００ａの各サーバ（たとえば１２２０ａ、１２２０ｂ、１２２０ｃ、１２２０ｄ）上で動作する。分散型データグリッドにおいて、ノードは、たとえばソフトウェアアプリケーション、仮想マシンなどであってもよく、サーバは、ノードがその上で動作するオペレーティングシステム、ハイパーバイザなど（図示せず）を含み得る。Oracle Coherenceのデータグリッドでは、各ノードはＪａｖａ仮想マシン（Java virtual machine：「ＪＶＭ」）である。ＣＰＵの処理能力およびサーバ上で利用できるメモリに応じて、各サーバ上に多数のＪＶＭ／ノードを設けてもよい。ＪＶＭ／ノードは、分散型データグリッドの要求に応じて、追加、起動、停止、および削除されてもよい。Oracle Coherenceを実行するＪＶＭは、起動時に自動的に参加しクラスタ化する。クラスタに加わるＪＶＭ／ノードは、クラスタメンバまたはクラスタノードと呼ばれる。

【０１６３】

各クライアントまたはサーバは、情報伝達のためにバスまたはその他の通信機構を含み、情報処理のためにバスに結合されたプロセッサを含む。プロセッサは、どのタイプの汎用または専用プロセッサであってもよい。各クライアントまたはサーバはさらに、プロセッサによって実行される命令および情報を格納するためのメモリを含み得る。メモリは、ランダムアクセスメモリ（「ＲＡＭ」）、読出専用メモリ（「ＲＯＭ」）、磁気もしくは光ディスクなどのスタティックストレージ、またはその他任意の種類のコンピュータ読取可能媒体を組合わせたもので構成することができる。各クライアントまたはサーバはさらに、ネットワークへのアクセス提供のためにネットワークインターフェイスカードなどの通信デバイスを含み得る。したがって、ユーザは、各クライアントまたはサーバに対して、直接、またはネットワークを通して遠隔から、またはその他任意の手段で、インターフェイスすることができる。

【０１６４】

コンピュータ読取可能な媒体は、プロセッサからアクセスすることが可能な利用可能な媒体であればどのようなものでもよく、揮発性媒体および不揮発性媒体、リムーバブルおよび非リムーバブル媒体、ならびに通信媒体を含む。通信媒体は、コンピュータ読取可能な命令、データ構造、プログラムモジュール、または、たとえば搬送波もしくはその他の搬送機構などの変調されたデータ信号内のその他のデータを含んでいてもよく、任意の情報伝達媒体を含む。

【０１６５】

プロセッサはさらに、液晶ディスプレイ（「ＬＣＤ」）などのディスプレイにバスを介して結合されてもよい。キーボード、およびコンピュータマウスなどのカーソル制御デバイスが、さらにバスに結合されることにより、ユーザが各クライアントまたはサーバに対してインターフェイスできるようにしてもよい。

【０１６６】
一実施形態において、メモリは、プロセッサが実行すると機能を提供するソフトウェアモジュールを格納する。モジュールは、各クライアントまたはサーバにオペレーティングシステム機能を提供するオペレーティングシステムを含む。モジュールはさらに、クラウドアイデンティティ管理機能を提供するためのクラウドアイデンティティ管理モジュールと、本明細書に開示されているその他すべての機能とを含み得る。
【０１６７】
クライアントは、クラウドサービスなどのウェブサービスにアクセスし得る。一実施形態において、ウェブサービスは、オラクル社のWebLogicサーバ上で実現されてもよい。他の実施形態ではウェブサービスの他の実装形態を使用してもよい。ウェブサービスは、クラウドデータを格納しているデータベースにアクセスする。
【０１６８】
データ管理－メタデータ駆動型フレームワーク
実施形態において、「リソースタイプ」と呼ばれる大量の各種データ／リソースを管理する必要がある。リソースタイプの例は、ユーザ、グループ、アプリケーション（「app」）、コンフィギュレーション、設定、パスワードポリシー等を含む。リソースタイプは、本発明の実施形態を使用してクラウド内で管理される管理エンティティである。
【０１６９】
各リソースタイプは、そのビヘイビア（behavior）を定義するさまざまなコンフィギュレーションを有する。あるコンフィギュレーションはスキーマ（すなわちそのタイプのリソースに関連付けることができるデータ）であり、プライマリ／コアスキーマと拡張スキーマとの両方を含む。たとえば、リソースタイプが「ユーザ」である場合、各ユーザは、このユーザに関連する４～５のスキーマを有し得る。コアスキーマは、ユーザ名（たとえば姓、名）、電話番号、住所、およびそのユーザの何らかの関連プロフィール情報を含む。拡張スキーマは、パスワード状態（たとえば、ユーザのパスワードが失効している、パスワード入力回数の上限を超過している等）またはユーザ状態（たとえばアクティブまたは非アクティブ）を含み得る。これらは「スキーマ属性」とみなされる。同様に、アプリケーションといった別のリソースタイプは、リソースが如何にして編成されるかおよびそのデータが如何に関連するかを定義する自身のスキーマを有する。
【０１７０】
一実施形態において、すべてのスキーマは属性のリストを有する。すべての属性は、その属性のビヘイビア（たとえば、それは読出か書込か、その可変性（すなわち変化するまたは変わる能力）は何か、それは必要か否か、それはサーチ可能か否か）を定義する一組のメタデータを有する。リソースタイプごとにリソースタイプ定義とスキーマ定義とがあってもよい。リソースタイプ定義は、（１）スキーマのリスト－コアスキーマおよび任意の拡張スキーマと、（２）サポートされるオペレーション－作成、交換、更新、削除、取得、サーチ、ポストサーチと、（３）データプロバイダタイプ：ＬＤＡＰプロバイダ、データベース（「ＤＢ」）プロバイダ、通知プロバイダ等、とを含み得る。スキーマ定義は、属性のリストと、属性ごとのメタデータとを含み得る。これは、その属性のプロパティおよびビヘイビアとして、データタイプ、可変性（読出し専用、読み書き、不変、書込み専用）、返却（常時、要求、なし）、ターゲット属性名、トリムスペース、最大長等、を記述している。したがって、メタデータは、リソースタイプの定義、どのスキーマがそれに関連しているか、および、各属性のメタデータは何かを、構成している。
【０１７１】
一例としての実施形態において、１７５種類のリソースタイプがある。これらのリソースタイプ各々および固有の各スキーマを管理するためにソフトウェアコードが必要である場合、各リソースタイプおよびスキーマ固有の大量のコードが必要であろう。たとえば、特定の属性に対するゲッター（getter）およびセッター（setter）（たとえば、ユーザ名をゲット（get）、ユーザ名をセット（set）、名をゲット、姓をゲット）を提供する「ＰＯＪＯ」（plain old Java object）リソースタイプの場合、すべてのリソースタイプに対

して大量のコードが必要であろう。
【０１７２】
これに対し、本発明の実施形態では、いずれのリソースタイプに対しても１つの統一されたコード／モジュールがすべての機能（たとえば、作成、更新、修正、削除、サーチ等）を果たす。先ず、実施形態は、リソースタイプが何であるかを判断し、次に、スキーマおよびスキーマ定義を調べて属性と属性定義とを判断する。その後、実施形態は、検証を行なうために、このリソースタイプのそのスキーマにはどの属性が必要かといった実行時判断を行なうことができる。ＬＤＡＰにおけるデータベース内のデータのパーシステンスを実行するとき、実施形態は、そのリソースタイプの属性が何であるかを認識しているので、ＳＱＬ更新／挿入のためにペイロードを正確に構築することができる。リソースごとに、リソースタイプおよびスキーマ定義は、構成されているプロバイダタイプおよびターゲット属性マッピングを含む。プロバイダタイプがＬＤＡＰであれば、ＬＤＡＰプロバイダが呼出されて、リソースタイプおよびスキーマ定義において構成されているディレクトリツリーの下で正しいオブジェクトクラスでＬＤＡＰにおいてデータをパーシストする。プロバイダタイプがＤＢプロバイダであれば、このプロバイダは、ＳＱＬを生成し、リソースタイプおよびスキーマ定義において構成されているようにテーブルの正しい列においてＤＢ内のデータをパーシスト／フェッチする。
【０１７３】
したがって、一実施形態は完全にデータ駆動型であるので、開発者が新たなリソースタイプの追加を希望する場合、開発者は、新たなコードを書込む代わりに、新たなリソースタイプおよびスキーマのＪＳＯＮ定義を追加するだけでよい。実施形態は完全にメタデータ駆動型であるので、データ駆動型モデルとみなされる。実施形態は、データをランタイムコードから完全に分離する（abstraction）。結果として、実施形態は、リソースタイプごとに別々のコードを書込む、維持する、またはテストする必要がない。
【０１７４】
一実施形態において、パフォーマンス上の理由により、すべてのメタデータはサービス起動時にキャッシュされる。さらに、実施形態は、メタデータを使用して文書化を駆動することができ、リソースタイプおよびスキーマ定義に基づいてＲＥＳＴ　ＡＰＩを含む外部文書を自動的に生成することができる。
【０１７５】
図１３は、一実施形態に係るＩＤＣＳまたはサービスとしてのアイデンティティ（Identity as a Service：「ＩＤａａＳ」）のデータマネージャアーキテクチャを示す。このアーキテクチャは、リソースマネージャ１３０１（または「ResourceManager」）と、ＳＣＩＭ／ＲＥＳＴ層１３１０と、ＡＰＩ層１３１１と、データプロバイダ（data provider：「ＤＰ」）層１３１２と、データストア層１３１３とを含む。
【０１７６】
リソースマネージャ１３０１は、ＩＤＣＳ／ＩＤａａＳのための共通データアクセス層である。これは、メタデータ駆動型であるので、ＳＣＩＭ準拠リソースタイプおよびスキーマ定義において定義されているどのリソースタイプも管理できる。一実施形態において、リソースマネージャ１３０１は、属性、データタイプ、必要性、カノニカル（canonical）値等のスキーマベースの検証を含む、すべてのリソースタイプのためのすべての共通ロジックを処理する。これはまた、機密属性の保護、ターゲット属性へのマッピング、認可検査およびイベント公開により、デフォルト値設定、作成／更新の日付、作成／更新を処理する。ＡＰＩ層１３１１内のリソースタイプ固有マネージャは、リソースマネージャ１３０１を拡張してどのリソースタイプ固有のオペレーションも処理する、またはＣＲＵＤＱオペレーションのための共通ロジックを拡張する。リソースマネージャ１３０１は、データストア固有インターフェイスと直接統合するデータプロバイダ層１３１３内のデータプロバイダを介してデータストア層１３１３内のデータストアと統合する。リソースマネージャ１３０１は、メタデータ駆動型であるので、リソースマネージャ１３０１またはリソースタイプマネージャ１３１１に影響を与えることなく、ランタイムスキーマカスタマ

イズおよびデータプロバイダ構成変更をサポートする。

【０１７７】

リソースタイプ（または「ResourceType」）は、ＩＤＣＳが管理するリソースのタイプである。その例は、ユーザ（User）、グループ（Group）、アプリケーション（Application）、トークン（Token）、キー（Key）等を含む。ＳＣＩＭにおいて、各リソースタイプはトップレベルのエンドポイントである（たとえば、/Users、/Groups）。各リソースタイプはリソースタイプ定義（または「ResourceTypeDef」）を有する。リソースタイプ定義は、所与のリソースタイプを記述するメタデータである。リソースタイプ定義は、もしあれば、リソースタイプ名、エンドポイント、プライマリスキーマＵＲＩおよび拡張スキーマＵＲＩを定義する。加えて、いくつかのＩＤＣＳ固有のメタデータを定義する。各リソースタイプ定義は、ＪＳＯＮのブロブ（blob）である。リソースタイプは、プレシード（pre-seed）することができる、または、実行時に作成することができる。どのリソースタイプ定義も、内部のものになるように構成することができ、これはＳＣＩＭ ＲＥＳＴ１３１０を介してアクセスできないことを意味し、または、外部のものになるように構成することができ、これはSCIM REST GET/ResourceTypesおよびGET/Schemas/〈ResourceType〉を通して発見できることを意味する。

【０１７８】

スキーマ定義（または「SchemaDef」）は、リソースタイプ（たとえば「デバイス（Device）」）全体またはその一部の内容を記述する属性定義（または「AttributeDefs」）の集合である。ＳＣＩＭは、実施形態に従い、コアメタデータおよびＩＤＣＳを定義し、メタデータを拡張する。具体的には、スキーマ定義は、そのリソースタイプのすべての属性およびサブ属性を記述する。各スキーマ定義はＪＳＯＮのブロブである。スキーマ定義はプレシードするまたは実行時に作成することができる。

【０１７９】

属性定義は、タイプ（たとえば、列、バイナリ）、基数（単数、複数、複素数）、可変性（読出し専用、読み書き等）、返却性、サーチ性等のメタデータおよび名称を定義する。属性名は、キャメルケース（たとえばcamelCased）でなければならず、それらが定義されているスキーマ内において固有のものでなければならない。一実施形態において、属性データタイプは、列（string）、ブール（Boolean）、１０進（Decimal）、整数（Integer）、日付（DateTime）、バイナリ（Binary）、参照（Reference）、または複素数（Complex）のうちの１つであってもよい。

【０１８０】

単数属性は、０．．１値を含むリソース属性である（たとえば「displayName」）。複数値属性は、０．．ｎ値を含むリソース属性である（たとえば「emails」）。単純属性は、その値がプリミティブである単数または複数値属性である（たとえば「String」）。複素数属性は、その値が１つ以上の単純属性の合成である単数または複数値属性であり（たとえば「addresses」)、サブ属性（たとえば「streetAddress」、「locality」、「postalCode」および「country」）を有する。サブ属性は、複素数属性に含まれる単純属性である。

【０１８１】

リソースは、１つ以上の属性を含むＩＤＣＳ管理アーティファクトのインスタンスである。ＳＣＩＭにおいて、リソースは、読み、操作することができるオブジェクトであり、たとえば、特定のユーザ、グループまたはトークン等である。各リソースは、グローバル一意識別子を有し、対応するリソースタイプ定義のスキーマに従う属性値を含む。

【０１８２】

一実施形態では、ＡＰＩ層１３１１において、Ｊａｖａクラスはリソースタイプごとにリソースタイプマネージャとして実現される。たとえばユーザマネージャはユーザを管理する。グループマネージャはグループを管理する。各マネージャは、そのリソースタイプのオブジェクトの管理に適したインターフェイスをエクスポーズする。すべてのリソースタイプマネージャは、リソースに関する作成、交換、更新、削除、取得およびサーチ方法を

実現する共通のアブストラクトリソースマネージャを拡張する。各リソースタイプマネージャは、必要であれば、アブストラクトリソースマネージャが実現する各方法のカスタム検証を実現できる。加えて、各リソースタイプマネージャは、必要に応じてこれらの方法を拡張できる。たとえば、ユーザマネージャは、イネーブル、ディスエーブル、ロック、アンロック、パスワード変更（changePassword）を含むユーザリソースに固有の方法をエクスポーズする。グループマネージャは、ユーザマネージャがエクスポーズしない、ユーザメンバシップ付与または取消等の方法をエクスポーズする。

【０１８３】

一実施形態において、ＡＰＩ層１３１１に対し、ＪＳＲ－３３０標準の注釈に基づくＨＫ２が使用されるであろう。各リソースタイプマネージャのカスタムＪａｖａインターフェイスには@Contractという注釈が付けられその実現には@Serviceという注釈が付けられるであろう。これにより、確実に、リソースタイプマネージャクラスは、リソースタイプマネージャの下でも〈ResourceType〉Managerimpl下でも宣伝されるサービスレジストリ内に置かれることになり、サービスロケータを介して要求可能であろう。

【０１８４】

一実施形態において、リソースマネージャ１３０１は、いずれかのリソースタイプのリソースをクエリし管理するための一組のＡＰＩを定義するステートレスな共通Ｊａｖａクラスである。リソースマネージャのインターフェイスには@Contractという注釈が付けられ、〈ResourceType〉Managerの実現には@Serviceという注釈が付けられ、これらは確実に、サービスレジストリ内に置かれることになり、ServiceLocatorを介して要求可能であろう。

【０１８５】

アブストラクトリソースマネージャ（または「AbstractResourceManager」）は、各〈ResourceType〉Managerが受け継ぐ共通のビヘイビアを提供するリソースマネージャインターフェイスを実現する。たとえば、AbstractResourceManagerは、認可を検査し、ResourceTypeDefに基づいて検証を実行し、作成、交換、更新および削除オペレーションのためにイベントを発行する。加えて、AbstractResourceManager方法は、〈ResouceType〉Managerにコールバックしてカスタム検証を可能にする。

【０１８６】

データプロバイダ層１３１２は、リソースマネージャ１３０１の下にあるプラガブル層である。これは、下にあるデータストアに対し各オペレーションを実現する。たとえば、ＪＤＢＣデータプロバイダは、ＪＤＢＣを用いてデータベースに対してトークする。ＪＮＤＩデータプロバイダは、ＪＮＤＩを用いてディレクトリサービスに対してトークする。データプロバイダは、各要求のテナントＩＤに基づいてデータストア間でスイッチするであろう。

【０１８７】

一実施形態は、先ず２種類のデータストアとしてＪＤＢＣとＪＮＤＩとをサポートする。他の実施形態は、ＮｏＳＱＬ等のその他のデータストアをサポートする。データストアは、テナント固有であり、リソースタイプ固有であってもよい。たとえば、イベントは、アプリケーションを格納するために使用されるＪＤＢＣデータベースとは別のＪＤＢＣデータベースに格納されてもよい。

【０１８８】

上述のように、実施形態は、共通リソースマネージャ１３０１が、リソースタイプとは関係なく検証、作成、およびエラー処理および例外を処理するだけでなく、メタデータに基づいてイベントを生成するので、データ駆動型である。イベントは、メッセージングサービスにおいて待ち行列に入れられ、バックエンドハンドラによって処理され、たとえば検査され、通知を生成する。

【０１８９】

図１４は、図１３のリソースデータマネージャによって実現される本発明の実施形態の機能フローを示す。機能は以下を含む。

【０１９０】
（１）１４０１で、リソースタイプは何かを解明する。
（２）１４０２で、オペレーションがリソースタイプによってサポートされていることを確認する（すなわちユーザが何をしようとしているかを確認する）。たとえば、いくつかのリソースタイプは、作成、更新、および削除をサポートし、いくつかのリソースタイプは取得およびサーチをサポートするだけである。このサポートはメタデータによって決められる。
【０１９１】
（３）１４０３で、認可検査を実施することにより、オペレーションの実行が認可されているか否かを判断する。
【０１９２】
（４）１４０４、１４０５で、たとえばカスタム検証、カスタム前処理、カスタム後処理、カスタムイベント生成のために、必要であれば（すなわち特定のリソースタイプがカスタマイズを要する）、リソースマネージャへのコールバックが実現される。このコールバックは、共通フローから呼出され、各リソースタイプがリソースタイプ固有のビヘイビアを注入できるので、新規である。
【０１９３】
（５）１４０６で、要求に基づいてデータプロバイダを取得する。所与のリソースタイプに対し、テナントに関係なくデータプロバイダは１つだけである。テナントに基づいて、データプロバイダは、（ＤＢに格納されているリソースを求めて）正しいデータベーススキーマへの接続を確立する、または、（ＬＤＡＰに格納されているリソースを求めて）正しいディレクトリツリーを指す。一実施形態では「Getdataprovider」Ｊａｖａ方法を使用する。
【０１９４】
（６）１４０６で、データプロバイダに対しオペレーションの実行を命ずる（たとえば、図１４の例においてオペレーションは「作成」）。
【０１９５】
（７）１４０７で、作成オペレーションの後に、リソースマネージャへのコールバックを実行して、何らかの後処理が必要か否かを判断する（たとえば、追加属性を結果に注入、データを修正）。プラグインコールバックをリソースマネージャに与える。
【０１９６】
（８）１４０８で、イベントを公開する。
（９）１４０９で、ＰＯＳＴレスポンスで戻ってくるリソースを返す。
【０１９７】
一実施形態において、図１４のフローにおけるメタデータの主な用途は、そのリソースタイプに対して定義されたスキーマに対するＰｏｓｔ要求１４０９と共に入ってくるペイロードの検証である。これは、キャッシュされたリソースデータに基づいて動的に行なわれる。しかしながら、リソースタイプのメタデータは検証のためだけのものではない。なぜなら、要求の最初にロードされたメタデータは、検証１４０２（有効な属性名、データタイプ、欠けている必要な属性等の検査）、認可１４０７、データプロバイダオペレーション１４３０（ターゲット属性マッピング、テーブルまたはオブジェクトクラスマッピング等）、後処理１４０９（メタデータにおけるリターン属性プロパティに基づくデータのフィルタリング）、イベント公開１４０８（メタデータにおけるどのイベントを公開するか）を含む、リソースマネージャ処理におけるどの段階でも使用できるからである。
【０１９８】
実施形態は、要求がどこから来たかを判断するとともに、リソースマネージャのユーザが誰であるかを判断する。ユーザは、エンドユーザであってもよく、別のアプリケーションであってもよく、内部ＩＤＣＳコンポーネント等であってもよい。クライアントが要求を行なうのが一般的であり、この要求は、ＨＴＴＰ要求の形態のＡＰＩ要求であり、次にリソースマネージャにハンドオーバされてリソースメタデータに基づいて処理する。

【０１９９】
一実施形態において、リソースマネージャ１３０１は、「管理サービス」と呼ばれるマイクロサービスであり、すべてのリソースタイプの管理を処理する。イベントを生成するときはメッセージングサービス（すなわちマイクロサービス）に対してトークする。再び図１３を参照して、リソースマネージャ１３０１は、管理サービスを表わし、図１３全体（下部のデータベースを除く）は、管理サービスマイクロサービスである。
【０２００】
一実施形態は、ＩＤＣＳのキャッシュコヒーレンスをサポートする。たとえば、要求取得、Get request：get/user/IDに対し、実施形態は先ずキャッシュデータプロバイダにクエリしてユーザＩＤがキャッシュされているか否かを判断する。キャッシュされていれば、データはキャッシュから返される。キャッシュされていなければ、データはデータストアから取出され、このデータは戻る途中でコヒーレンスキャッシュに追加される。
【０２０１】
テナントに基づき、実施形態は、メタデータを用いてどのＤＢ（たとえば図１３のＬＤＡＰデータパーティション「ＤＰ」１３０５）を読取るかを判断する。実施形態は、テナントに基づいてデータ層でＤＢスイッチを実行する。
【０２０２】
実施形態は、データを管理する必要があるどのサービスでも使用できる。実施形態を使用するクライアントは、ＵＩサービスコンソール、インポートジョブ、またはＩＤＣＳ内のデータを更新するものであればいずれをも含み得る。
【０２０３】
<u>データ管理－マルチテナント</u>
一実施形態は、データ層におけるマルチテナントサポートを実現する。リソースマネージャ１３０１は、要求を処理し、その後、この要求に適したデータソースを求める。この処理は、テナント駆動型であり、メタデータ機能とは別である。この機能は、実施形態に、テナントに基づくデータストア間でのランタイムスイッチを行なう能力を提供し、これはセキュリティに役立つ（すなわちテナントデータの分離）。
【０２０４】
その他周知のアイデンティティマネージャは複数のテナントを持っていない場合がある。その代わりに、周知のアイデンティティマネージャシステムの中には、テナントごとにすべてのテーブルに対しデータベース内の異なる列を用いてストライピングを実行するものがある。これらのソリューションにおいては、１つのテーブルが、混合された複数のテナントデータすべてを有することになるであろう。これは安全ではない場合がある。
【０２０５】
これに対し、一実施形態では、ストライピングの代わりに、テナントごとに異なるデータベースが使用される。実施形態は、要求ごとに、適切なデータソースへのランタイムスイッチを実行する。
【０２０６】
<u>データ管理－自動スキーマバージョニング</u>
一実施形態において、リソースタイプ（たとえば「ユーザ」リソースタイプ）の寿命にわたり、バージョン１はユーザに対するスキーマを有し得る。リソースタイプの次のバージョン２において、属性の追加または削除が必要な場合があり、そうすると、バージョン１のすべての属性を有するスキーマとバージョン２のすべての属性を有するバージョン２のスキーマとを表わすスキーマを複製する必要が生じ得る。新たなバージョン各々についてスキーマを複製し続ける必要が生じ得る。
【０２０７】
しかしながら、これに対し、一実施形態では、スキーマを複製するのではなく、スキーマ属性そのものに、バージョン属性以降追加（added since version attribute）またはバージョン属性以降非推奨（deprecated since version attribute）をタグ付けすることができる。

【０２０８】
たとえば、一実施形態は、「レッド（red）」と呼ばれるリソースタイプのバージョン１属性を有し得る。バージョン２において、レッドはもはや必要ではない。単一ユーザスキーマにおいて、実施形態は、レッド属性を、バージョン２に対して非推奨（deprecated）としてタグ付けするが、バージョン２に３つの新たな属性として、Ａ、ＢおよびＣを追加できる。これら３つの新たな属性に対し、バージョン２のためのタグが追加される。
【０２０９】
実行時において、ユーザが要求を行なったとき、この要求は、作業に使用したいスキーマのバージョンを含み得る（たとえば、バージョン１ユーザ、バージョン２ユーザ等）。実行時に、実施形態は、（メタデータ駆動型）スキーマを評価することにより、スキーマバージョン１が何を含むか、スキーマバージョン２が何を含むか、等を判断できる。
【０２１０】
ユースケースの一例：スキーマのバージョン２についてユーザの取得を要求する。この要求は、追加された非推奨の属性に基づいてユーザを取得する。このため、非推奨の属性を返すのではなく、この要求は、追加された属性を返す。これに対し、バージョン１は、追加されたまたは非推奨のタグを持たないであろう。
【０２１１】
実施形態は完全にメタデータ駆動型である。これにより、すべてのユーザの複数のバージョンを、同一のリソースマネージャサービスを通して同時にサポートできる。実施形態は、ダウンタイムゼロでスキーマ変更をサポートすることができる。
【０２１２】
図１５は、一実施形態に係る自動スキーマバージョニングを示す。図１５に示されるように、ユーザスキーマのバージョン１は、属性"name"および"type"を含む（１５０１）。バージョン２は"name"および"costcenter"を含む（１５０２）。この例において、属性「Type」は、バージョン２に含まれていないので、非推奨とされた。
【０２１３】
ユーザを取得することを求める要求がなされると、一実施形態において、デフォルトでスキーマの最新バージョン（すなわち図１５のバージョン２）が常に取出される。しかしながら、要求ペイロードはその代わりにバージョン１を要求することができる。この場合、実施形態は、このスキーマに固有のデータのサブセットを取得するであろう。実施形態はキャッシュでも機能する。
【０２１４】
大抵の周知ソリューションは、バージョンごとに別々のスキーマ定義を有する。これに対し、実施形態は、単一のスキーマ定義、および、バージョン間の変更を規定するメタデータのみを有する。１つの利点はゼロダウンタイムのサポートである。
【０２１５】
開示されている実施形態は、リソースタイプおよび関連するスキーマを定義するメタデータを実現する。マルチテナントシステムにおいてリソースに対するオペレーションの実行を求める要求は、メタデータの使用により解決されて、このオペレーションを実行するテナントに関連するデータプロバイダが決定される。
【０２１６】
本明細書ではいくつかの実施形態が具体的に例示および／または記載されている。しかしながら、開示されている実施形態の修正および変形は、本発明の精神および意図する範囲から逸脱することなく、上記教示によってカバーされ以下の請求項の範囲に含まれることが、理解されるであろう。

【図面】

【図１】

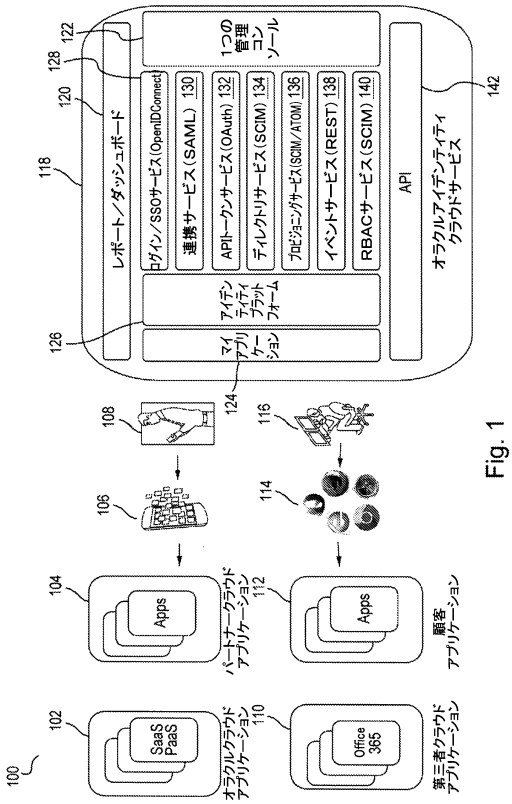

Fig. 1

【図２】

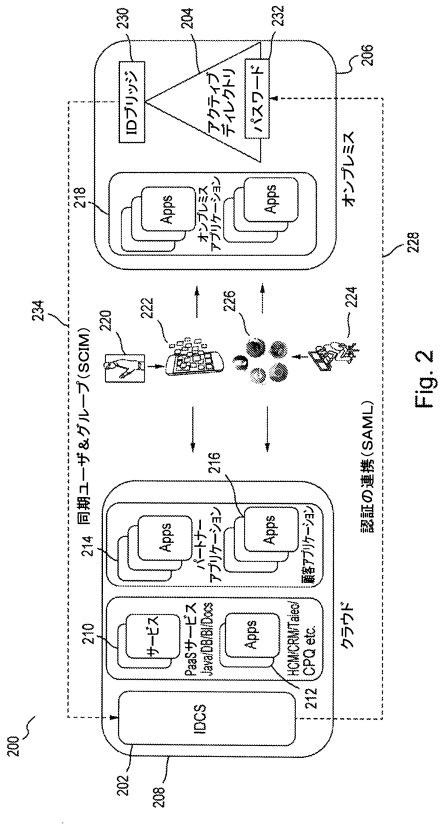

Fig. 2

【図３】

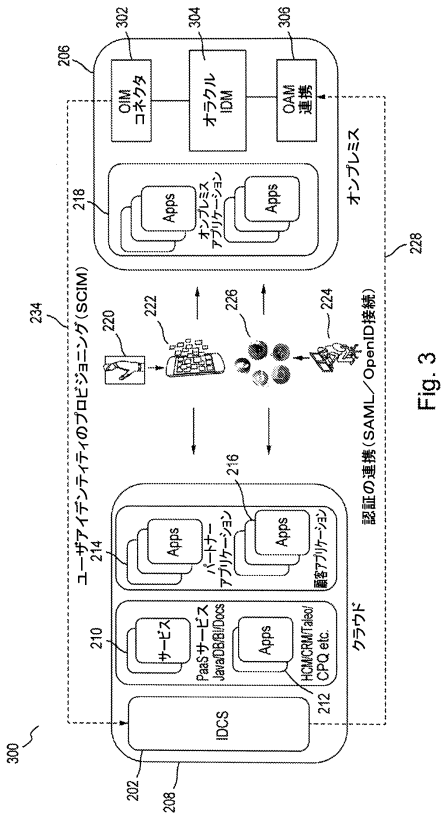

Fig. 3

【図４】

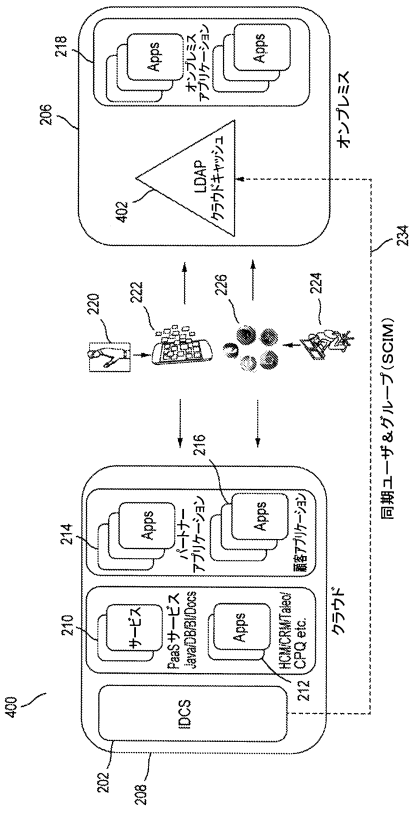

Fig. 4

【図５】

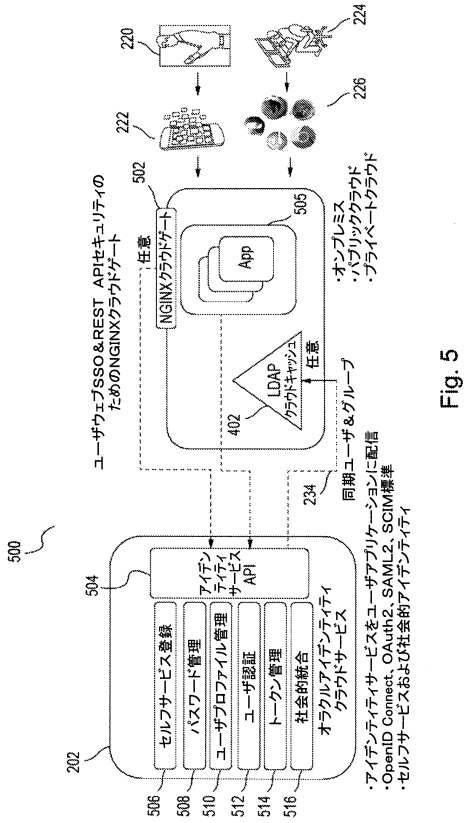

Fig. 5

【図６】

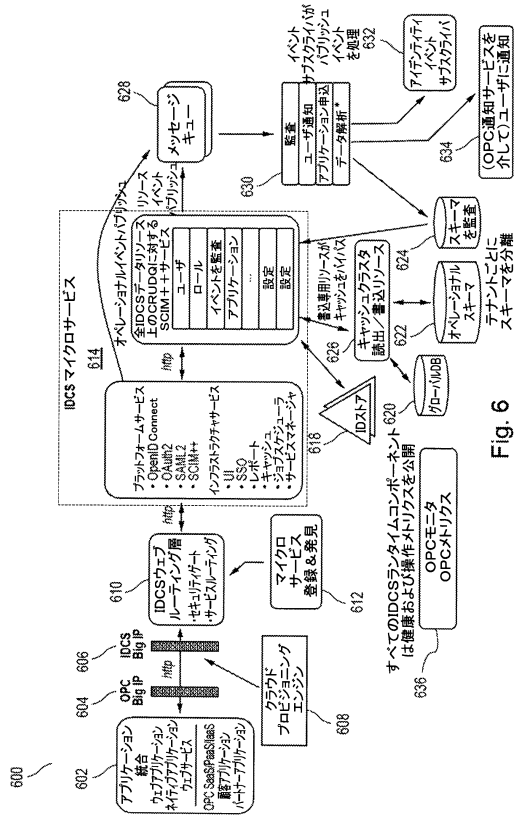

Fig. 6

【図６Ａ】

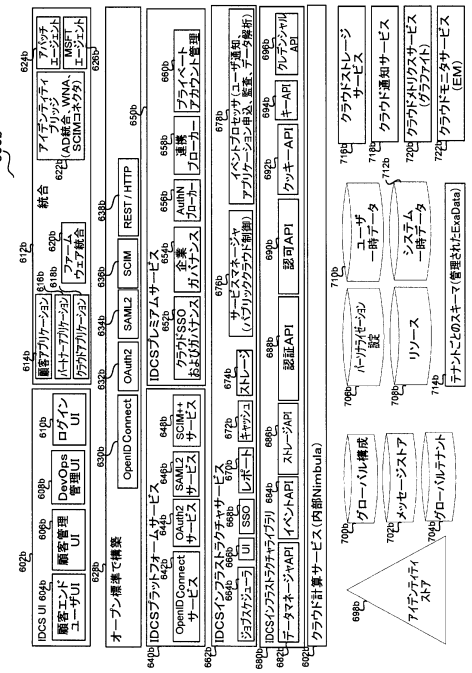

Fig. 6A

【図７】

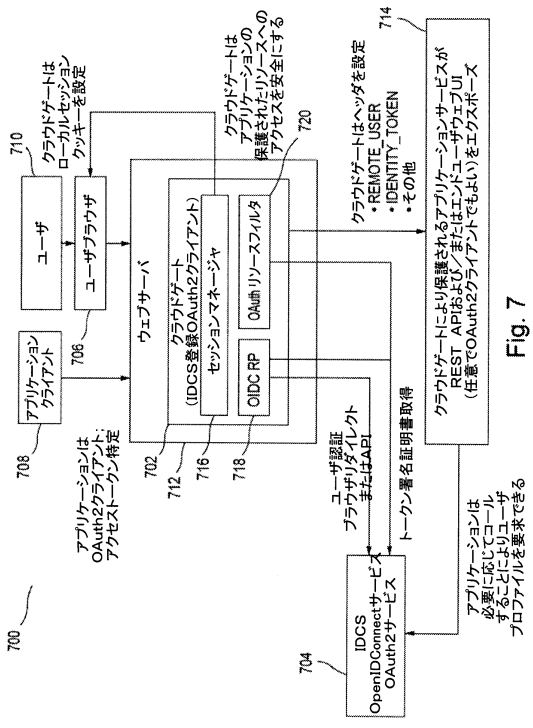

Fig. 7

【図８】

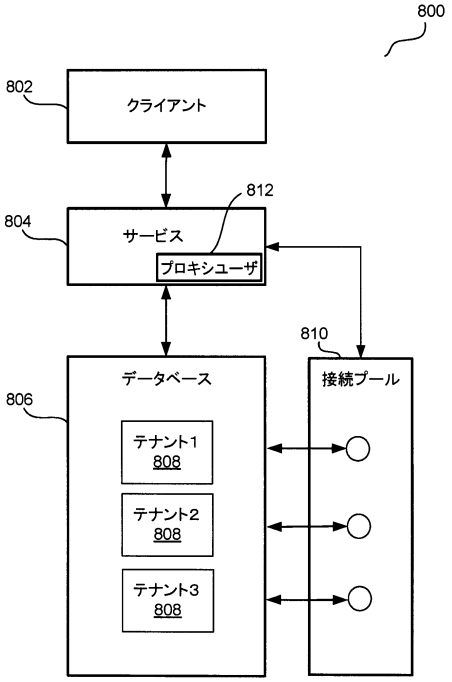

Fig. 8

【図９】

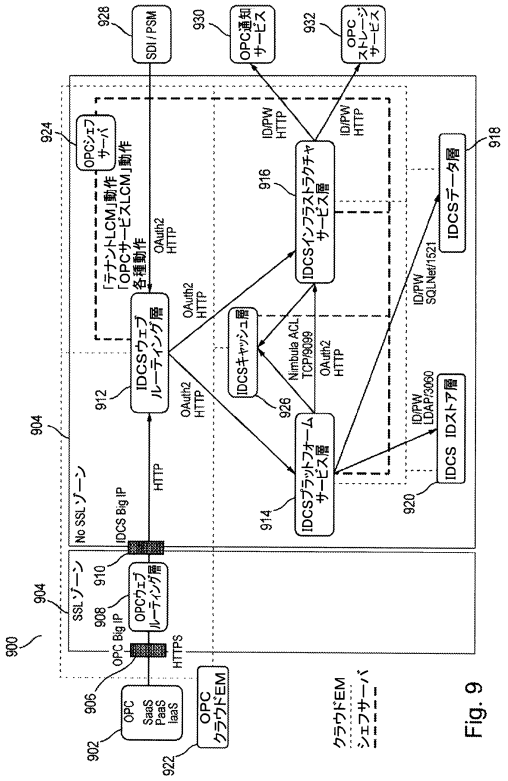

Fig. 9

【図１０】

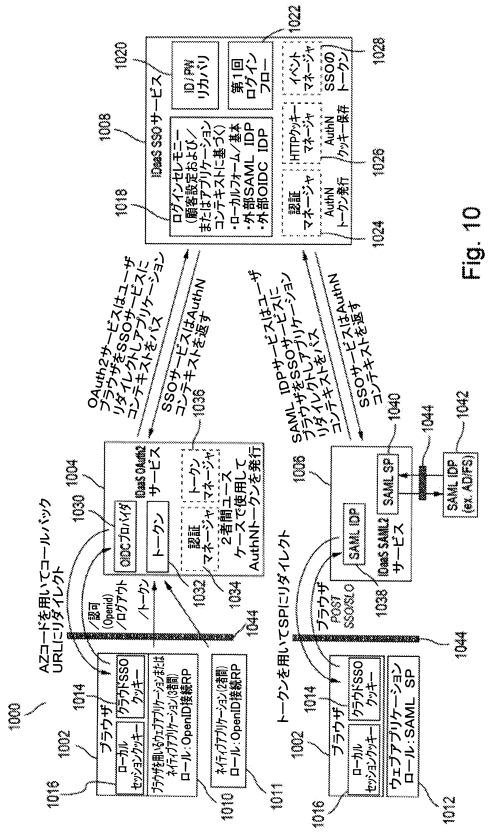

Fig. 10

【図１１】

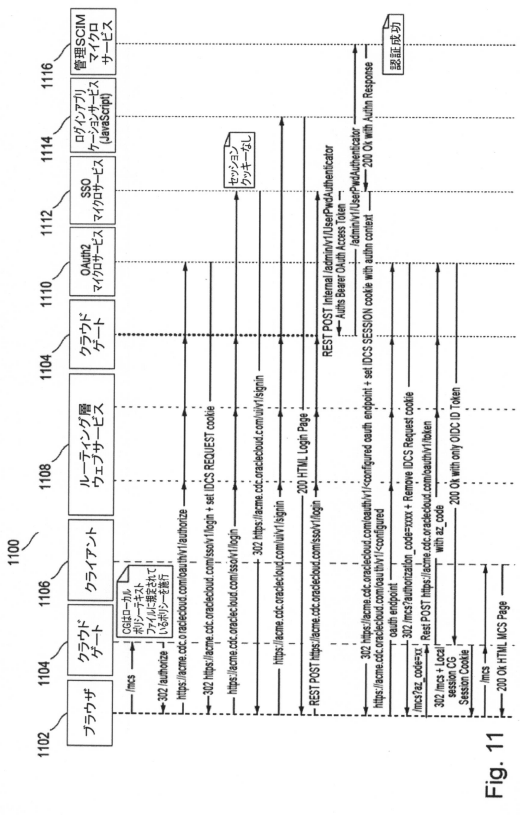

Fig. 11

【図１２】

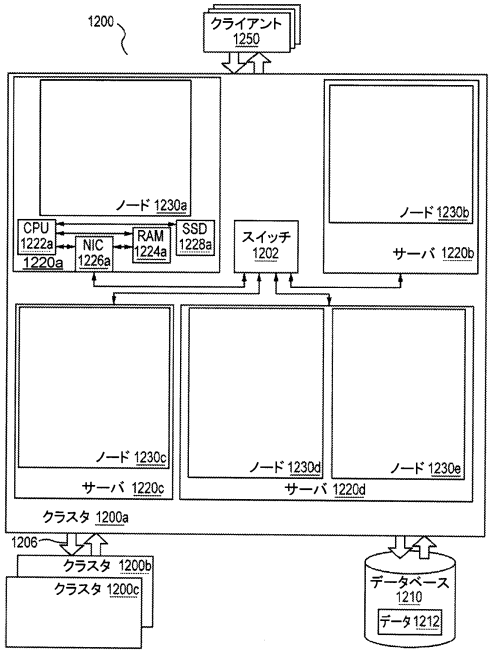

Fig. 12

【図１３】

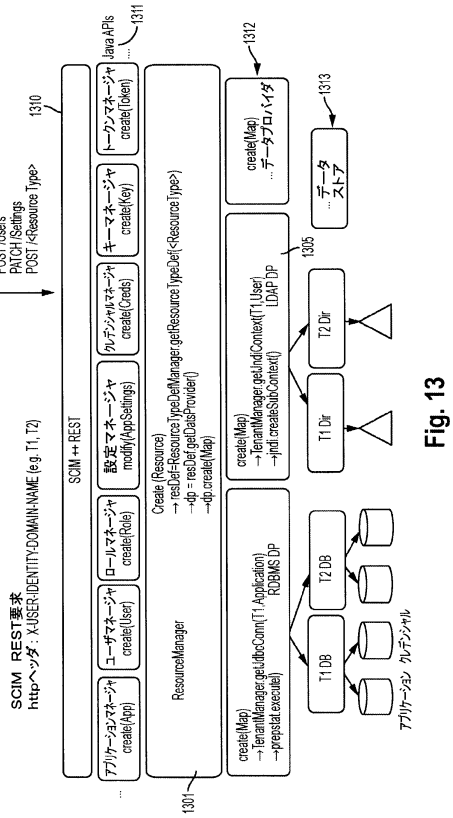

Fig. 13

【図１４】

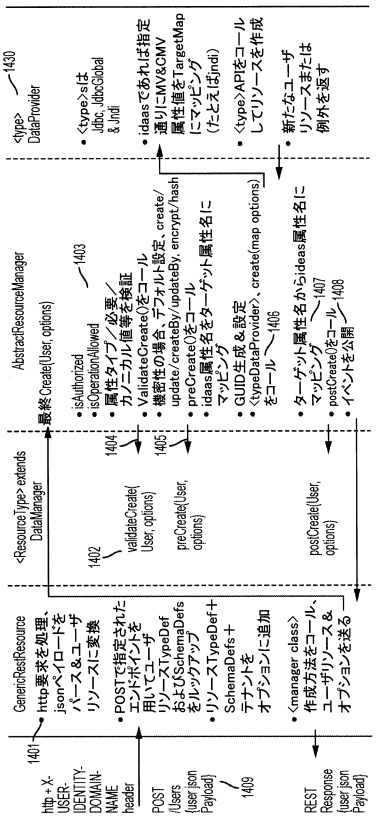

Fig. 14

【図１５】

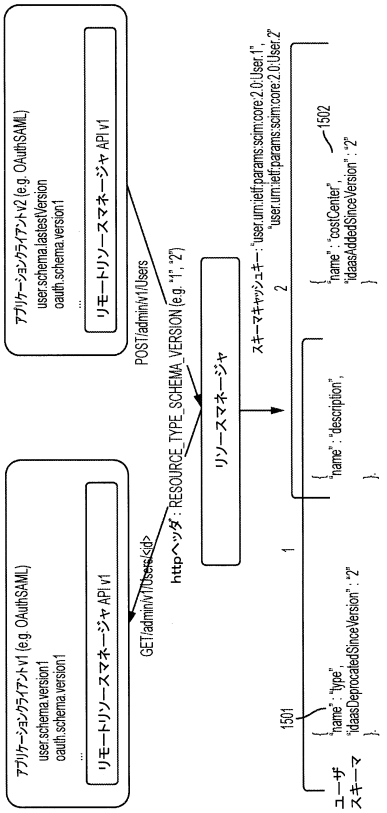

Fig. 15

フロントページの続き

(33)優先権主張国・地域又は機関
米国(US)
２０５
審査官　打出 義尚
(56)参考文献　特開２０１２－１０３８４６（ＪＰ，Ａ）
特開２０１１－１９８１０９（ＪＰ，Ａ）
(58)調査した分野　(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ　２１／３１