

(19)日本国特許庁(JP)

## (12)特許公報(B2)

(11)特許番号  
特許第7018437号  
(P7018437)

(45)発行日 令和4年2月10日(2022.2.10)

(24)登録日 令和4年2月2日(2022.2.2)

(51)国際特許分類  
G 0 6 F 16/13 (2019.01)F I  
G 0 6 F 16/13

請求項の数 9 (全51頁)

(21)出願番号	特願2019-514749(P2019-514749)	(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940 65レッドウッド・シティー, オラクル ・パークウェイ500
(86)(22)出願日	平成29年9月15日(2017.9.15)	(74)代理人	110001195 特許業務法人深見特許事務所
(65)公表番号	特表2019-532418(P2019-532418 A)	(72)発明者	スプラマニア, エランゴバン アメリカ合衆国、94065 カリフォ ルニア州、レッドウッド・ショアーズ、 オラクル・パークウェイ、200、2・ オウ・ピイ・662
(43)公表日	令和1年11月7日(2019.11.7)	(72)発明者	ニーラカンテシュワール, マノハリ インド、560018 バンガロール、 最終頁に続く
(86)国際出願番号	PCT/US2017/051769		
(87)国際公開番号	WO2018/053258		
(87)国際公開日	平成30年3月22日(2018.3.22)		
審査請求日	令和2年6月23日(2020.6.23)		
(31)優先権主張番号	62/395,426		
(32)優先日	平成28年9月16日(2016.9.16)		
(33)優先権主張国・地域又は機関	米国(US)		

(54)【発明の名称】 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管理クラウドサービスのため  
のテナントおよびサービス管理

## (57)【特許請求の範囲】

## 【請求項1】

アイデンティティクラウドサービス(I D C S)のためのテナントおよびサービス管理を  
提供するコンピュータが実行する方法であって、前記方法は、

ユーザのI D C Sのサブスクリプションに応答して、I D C Sクラウドインフラテナンシーを作成するステップを含み、前記I D C Sクラウドインフラテナンシーはウォレットに入れるアクセストークンを提供し、前記方法はさらに、

テナントのためのI D C Sサービスインスタンスを作成することを求める要求を、テナント自動化システム(T A S)から、ネットワークを介して受けるステップを含み、前記T A Sは、アクセストークンの権利を用いて、1つ以上の安全なアプリケーションプログラミングインターフェイス(A P I)を呼び出し、前記方法はさらに、

前記I D C Sサービスインスタンスを、タスクオーケストレータと複数の同期タスクとを含むタスク実行フレームワークを用いて作成するステップとを含み、前記タスクオーケストレータは、前記タスクをインスタンス化し、前記タスクを制御し、前記要求に対する応答を生成し、前記方法はさらに、

前記テナントのためのI D C Sサービスインスタンスが作成されたことを示す前記応答を、前記ネットワークを介して前記T A Sに送るステップを含む、方法。

## 【請求項2】

前記I D C Sサービスインスタンスは、アドミニストレータによるマニュアル介入を受けることなく作成される、請求項1に記載の方法。

**【請求項 3】**

前記応答はテナント固有のデータを含む、請求項 1 または 2 に記載の方法。

**【請求項 4】**

前記 I D C S サービスインスタンスを作成することは、ロードバランシングポリシーを構成することと、ルーティング統一資源位置指定子 ( U R L ) を作成することと、前記テナントのためのネームスペースをプロビジョニングすることとを含む、請求項 1 ~ 3 のいずれか 1 項に記載の方法。

**【請求項 5】**

前記ルーティング U R L は、パーマネント U R L 、フレンドリー U R L 、およびバニティ U R L を含む、請求項 4 に記載の方法。

10

**【請求項 6】**

前記 I D C S サービスインスタンスは、予め作成された複数のスキーマのうちの 1 つのスキーマを用いて作成される、請求項 2 に記載の方法。

**【請求項 7】**

前記タスク実行フレームワークを用いて前記 I D C S サービスインスタンスをアップグレードするステップをさらに含み、前記アップグレードは、前記テナントに関連付けられたアーティファクトのアップグレードを含む、請求項 1 ~ 6 のいずれか 1 項に記載の方法。

**【請求項 8】**

アイデンティティクラウドサービス ( I D C S ) のためのテナントおよびサービス管理を提供するためのシステムであって、前記システムは、  
メモリと、

20

前記メモリとネットワークとに結合されたプロセッサとを備え、前記プロセッサは、  
ユーザの I D C S のサブスクリプションに応答して、I D C S クラウドインフラテナンシーを作成するように構成され、前記 I D C S クラウドインフラテナンシーはウォレットに入れるアクセストークンを提供し、

テナントのための I D C S サービスインスタンスを作成することを求める要求を、テナント自動化システム ( T A S ) から、ネットワークを介して受けるように構成され、前記 T A S は、アクセストークンの権利を用いて、1 つ以上の安全なアプリケーションプログラミングインターフェイス ( A P I ) を呼び出し、

30

前記 I D C S サービスインスタンスを、タスクオーケストレータと複数の同期タスクとを含むタスク実行フレームワークを用いて作成するように構成され、前記タスクオーケストレータは、前記タスクをインスタンス化し、前記タスクを制御し、前記要求に対する応答を生成し、前記プロセッサはさらに、

前記テナントのための I D C S サービスインスタンスが作成されたことを示す前記応答を、前記ネットワークを介して前記 T A S に送るように構成される、システム。

**【請求項 9】**

請求項 1 ~ 7 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み取り可能プログラム。

40

**【発明の詳細な説明】****【技術分野】****【0 0 0 1】**

分野

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

**【背景技術】****【0 0 0 2】**

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

50

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】

【0003】

概要

実施形態は、ステートレスな中間層において複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および方法を提供する。

【0004】

特定の実施形態において、アイデンティティクラウドサービス（Identity Cloud Service）（IDCS）のためのテナントおよびサービス管理が提供される。テナントのためのIDCSサービスインスタンスを作成することを求める要求を、テナント自動化システム（Tenant Automation System）（TAS）から、ネットワークを介して受ける。IDCSサービスインスタンスを、タスクオーケストレータと複数の同期タスクとを含むタスク実行フレームワークを用いて作成する。タスクオーケストレータは、タスクをインスタンス化し、タスクを制御し、上記要求に対する応答を生成する。上記テナントのためのIDCSサービスインスタンスが作成されたことを示す応答を、ネットワークを介してTASに送る。

【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ト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（「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】本発明の実施形態に従う、IDCSサービスインスタンスライフサイクル管理のシーケンスフローを示す。

【図14】本発明の実施形態に従う、IDCSのためのテナントおよびサービス管理を提供するためのプロセスを示す。

10

20

30

40

50

【図15】本発明の実施形態に従う、ＩＤＣＳサービスインスタンスの作成を求める要求を受けたときに実行されるプロセスを示す。

【図16】本発明のある実施形態に従う、タスク実行フレームワークのブロック図を示す。

【図17】本発明のある実施形態に従う、テナントアップグレード機能のブロック図を示す。

【図18】本発明のある実施形態に従う、スキーマ作成機能のブロック図を示す。

【発明を実施するための形態】

【0006】

詳細な説明

本発明の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み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。

10

【0007】

本発明の実施形態は、ステートレスな中間層環境において複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および方法を提供する。特定の実施形態において、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。本発明の実施形態は、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。

20

【0008】

特定の実施形態において、アイデンティティクラウドサービス（IDCS）のためのテナントおよびサービス管理が提供される。テナントのためのIDCSサービスインスタンスを作成することを求める要求を、テナント自動化システム（TAS）から、ネットワークを介して受ける。IDCSサービスインスタンスを、タスクオーケストレータと複数の同期タスクとを含むタスク実行フレームワークを用いて作成する。タスクオーケストレータは、タスクをインスタンス化し、タスクを制御し、上記要求に対する応答を生成する。上記テナントのためのIDCSサービスインスタンスが作成されたことを示す応答を、ネットワークを介してTASに送る。

30

【0009】

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

40

50

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。

#### 【0010】

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。

10

#### 【0011】

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。

20

#### 【0012】

##### 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 (「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 (Single Sign On) (「SSO」) 機能を提供することができる。

30

#### 【0013】

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。マルチテナントとは、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り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。

40

#### 【0014】

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、單一

50

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H T T PリソースA P I）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遙かに短い。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。

#### 【0015】

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)（「S O A」）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。

#### 【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A P I）。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カー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ト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（「H T T P」）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カー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。

#### 【0017】

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I A M機能を提供する。

#### 【0018】

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

10

20

30

40

50

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フロードを実行し、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。

#### 【 0 0 1 9 】

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操作性を提供する。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。

10

#### 【 0 0 2 0 】

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、OAuth 2、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（「SAML 2」）、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ン・ステート・トランスマネージメント（Representation 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。

20

#### 【 0 0 2 1 】

一実施形態は、インター操作性を提供し、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）。

30

#### 【 0 0 2 2 】

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。

40

#### 【 0 0 2 3 】

50

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。

10

20

#### 【0024】

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）、トーカンサービス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。

30

#### 【0025】

##### 統合ツール

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。

40

#### 【0026】

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

50

、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。

#### 【0027】

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。

10

#### 【0028】

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、メッセージの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。

20

#### 【0029】

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。

30

#### 【0030】

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

40

50

ーション 218 にアクセスできるようにしてもよい。

#### 【0031】

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。

10

#### 【0032】

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、このアカウントからのアクセスを管理することができる。

20

#### 【0033】

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。

30

#### 【0034】

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から引き出す。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。

40

#### 【0035】

50

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。

#### 【0036】

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。

10

#### 【0037】

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など）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と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。

20

#### 【0038】

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ーポイントはSSOである。クラウドSSOは、ユーザが、1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）。

30

#### 【0039】

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。

40

50

**【 0 0 4 0 】**

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。

**【 0 0 4 1 】**

10

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。

20

**【 0 0 4 2 】**

30

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つ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なだけでなく、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、IDCS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

40

50

、コスト、および制御の点で利益をもたらし得る。

【0043】

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り当てられているからである。クラウドへの動きはこの状況をさらに混乱させる可能性があり、事態は、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。

【0044】

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。

【0045】

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、エンドユーザー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とする。

【0046】

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

10

20

30

40

50

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。

#### 【0047】

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。

#### 【0048】

##### 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トする。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をエクスポートする。すなわち、各マイクロサービスは異なる種類のAPIをエクスポートし得る。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トされない。APIは、IDCSが提供するサービスを使用するためにサービス消費者が利用できるようにされるものである。

#### 【0049】

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/）。

10

20

30

40

50

v1/token)。

#### 【0050】

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トし、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り出すことができる。

10

#### 【0051】

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。

20

#### 【0052】

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ンングエンジン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ンングエンジン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。

30

#### 【0053】

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

40

50

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。

#### 【0054】

##### 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。

#### 【0055】

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。

10

20

#### 【0056】

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。

30

#### 【0057】

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。

40

#### 【0058】

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

50

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トし、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トする。

#### 【 0059 】

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)、読み取り (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 としてエクスポートされる。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。

#### 【 0060 】

一実施形態において、たとえば、SCIM 標準は、SCIM 標準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 標準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。

#### 【 0061 】

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。

#### 【 0062 】

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、IDCS Java クライアント API ライブラリを使用することができる。

#### 【 0063 】

##### 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

10

20

30

40

50

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) ( 「 U I 」 ) 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 ( 「 U I 」 ) サービスと、サービススマネージャサービスとを含む。サービススマネージャは、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。

#### 【 0 0 6 4 】

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 ) 。

#### 【 0 0 6 5 】

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。

#### 【 0 0 6 6 】

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 ) オペレーションナルスキーマ 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。

#### 【 0 0 6 7 】

##### 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。

#### 【 0 0 6 8 】

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H T T P 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。

#### 【 0 0 6 9 】

マイクロサービスを求めるH T T P 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 6 2 8 にオフロードされる。メッセージキュー 6 2 8 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 6 3 0 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、I D C S 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。

10

#### 【 0 0 7 0 】

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、I D C S S C I M A P I 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 6 1 8 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。I D C S 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、オペレーションナルデータベース（図 6 のグローバルデータベース 6 2 0 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 A P I 6 1 6 を用いてこのメッセージを、ストアであるキュー 6 2 8 に入れる。

20

#### 【 0 0 7 1 】

キュー 6 2 8 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 6 2 8 の中にあるイベントを探してキュー 6 2 8 をスキャンする。キュー 6 2 8 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リーバ 6 3 0 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、イベントサブスクリーバ 6 3 0 は、たとえば、監査スキーマ 6 2 4 、ユーザ通知サービス 6 3 4 、アイデンティティイベントサブスクリーバ 6 3 2 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 6 2 8 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。

30

#### 【 0 0 7 2 】

一実施形態において、キュー 6 2 8 は、マイクロサービス 6 1 4 によってパブリッシュされたオペレーションナルイベントと、I D C S リソースを管理する A P I 6 1 6 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。

40

#### 【 0 0 7 3 】

I D C S 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。キャッシングそのものは、マイクロサービスとしても提供される。I

50

DCS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・キャッシングクラスタ626を実現する。キャッシングクラスタ626は、以下に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ングをバイパスする。

#### 【0074】

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ンメトリクスを、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。

#### 【0075】

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、オペレーションを監査することにより監査スキーマ624内のテーブルを監査し、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。

10

#### 【0076】

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。

20

#### 【0077】

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。

30

#### 【0078】

40

50

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。

#### 【 0 0 7 9 】

##### ステートレス機能

I D C S 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。I D C S は、マルチテナント機能を、すべてのステートを、I D C S データ層内の特定テナント向けリポジトリにパース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、I D C S 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。

10

#### 【 0 0 8 0 】

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 C P U 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。

20

#### 【 0 0 8 1 】

I D C S の中間層がステートレスであることにより、C P U 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 I D C S 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。I D C S 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、I D C S の可用性が高くなる。I D C S 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 C P U 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ライスを追加することによって実現される。

30

#### 【 0 0 8 2 】

I D C S 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 H T T P 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。

40

#### 【 0 0 8 3 】

##### I D C S 機能ビュー

図 6 A は、一実施形態における I D C S の機能ビューのブロック図の一例 6 0 0 b である。ブロック図 6 0 0 b において、I D C S 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、I D C S プラットフォームサービス 6 4 0 b と、I D C S プレミアムサービス 6 5 0 b と、I D C S インフラストラクチャサービス 6 6 2 b とを含む。一実施形態において、I D C S プラットフォームサービス 6 4 0 b および I D C S プレミアムサービス 6 5 0 b は、別々にデプロイされた J a v a ベースのランタ

50

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、ユーザー時データ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、オペレーションナルストア（図示せず）、キャッシングストア（図示せず）などを含む。

#### 【0084】

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プローカー656b、連携プローカー658b、およびプライベートアカウント管理660bを含む。

#### 【0085】

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、キャッシング672b、ストレージ674b、サービススマネージャ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。

#### 【0086】

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）、アパッチャエージェント624b、またはMSFTエージェント626bによって提供される。

#### 【0087】

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

10

20

30

40

50

を理解することから切離す。

【0088】

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（グラフィット）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。

【0089】

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WebAgent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。

10

【0090】

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OAuth2、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。

20

【0091】

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。

30

【0092】

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つつローカルセッ

40

50

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。

#### 【 0093 】

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。

#### 【 0094 】

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 API 714 のための OAuth 2 リソースサーバ / フィルタ 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 API 714 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。

#### 【 0095 】

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、RFC 7519 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。

#### 【 0096 】

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トを提供する。

#### 【 0097 】

##### テナントの種類

IDCS は 3 種類のテナントとして、顧客テナント、クライアントテナント、およびユーザテナントを特定する。顧客またはリソーステナントは、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トは、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。ユーザテナントは、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、ユーザテナントは、この専門サービス企業に相当し、クライアントテナント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、顧客テナントは大型ディスカウントショップである。

#### 【 0098 】

これら 3 つのテナント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 つのテナント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、

10

20

30

40

50

トークンをもつて、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つのテナントは、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。

#### 【0099】

ユーザ、クライアント、およびリソーステナント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。

#### 【0100】

周知のシステムの中には、顧客テナントである単一のテナント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み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ト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。

#### 【0101】

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、「テナント」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トを有し、顧客は自身のテナントを有する。要求がサブミットされたとき、この要求に関わる複数のテナントが存在する。

#### 【0102】

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。

#### 【0103】

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要求を呼び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

10

20

30

40

50

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び出す権利を有し、「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。

#### 【0104】

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。

10

#### 【0105】

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じ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。

30

#### 【0106】

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ー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。

40

#### 【0107】

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;UserID=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

50

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などに関連して）開いている 10 の接続があるであろう。

#### 【0108】

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り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トにのみアクセスでき、このテナント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、マルチテナントは、リソーステナント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。

#### 【0109】

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。

#### 【0110】

#### 【数1】

```
dbOperation = <prepare DB command to execute>
```

```
dbConnection = getDBConnectionFromPool() dbConnection.setProxyUser (resourceTenant)
```

```
result = dbConnection.executeOperation (dbOperation)
```

10

20

30

#### 【0111】

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 から引き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。

#### 【0112】

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、マルチテナントは、プラガブルデータベース（たとえばオラクル社のOracle Database 12c）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り当てられる。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。

40

50

## 【0113】

一実施形態において、互いに異なるトークンは、異なるテナント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トを特定し得る。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トを特定し得る。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ト（たとえばアプリケーションテナント）と、アクセス権が付与されたユーザのユーザテナント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トを特定し得る。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トを特定し得る。

10

## 【0114】

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）」を含む。

## 【0115】

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。

## 【0116】

20

## 【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>"

```

30

...

}

## 【0117】

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。

40

## 【0118】

本明細書に記載のトークンおよび／または複数のテナントは、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トは、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。

## 【0119】

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。

50

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。

#### 【0120】

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トされ、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。

10

#### 【0121】

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の外部から呼び出された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、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の内部から呼び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。

20

30

#### 【0122】

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。

#### 【0123】

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。

40

#### 【0124】

50

I D C S 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うことによって通信する。ネットワークコールは、H T T P、伝送制御プロトコル（Transmission Control Protocol）（「T C P」）、ユーザデータグラムプロトコル（User Datagram Protocol）（「U D P」）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、H T T Pに基づいて、アプリケーション「Y」をH T T P ユニフォーム・リソース・ロケータ（Uniform Resource Locator）（「U R L」）としてエクスポートすることにより、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トするI D C Sマイクロサービスである。「X」（たとえば別のI D C S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、呼び出す必要があるリソース/機能とを含むU R Lを構成し（たとえばhttps://host/Y/resource）、ウェブルーティング層9 1 2を通って「Y」に導かれる対応するR E S Tコールを行う。

#### 【0 1 2 5】

一実施形態において、I D C S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9 1 2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、I D C Sは、発見機能を実現する（O A u t h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。

#### 【0 1 2 6】

一実施形態において、企業マネージャ（enterprise manager）（「E M」）9 2 2は、オンプレミスおよびクラウドベース管理をI D C S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9 2 4は、さまざまなI D C S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9 2 8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、O A u t h 2 H T T PメッセージをI D C Sウェブルーティング層9 1 2に送信してもよい。一実施形態において、I D C Sインフラストラクチャサービス層9 1 6は、I D / パスワードH T T Pメッセージを、パブリッククラウド通知サービス9 3 0またはパブリッククラウドストレージサービス9 3 2に送信してもよい。

#### 【0 1 2 7】

##### クラウドアクセス制御 S S O

一実施形態は、クラウドスケールS S O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、H T T P、R E S T、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、S O A P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O p e n I D C o n n e c tセマンティックスを使用することにより、I D C Sに対してユーザ認証を要求する。本実施形態は、軽量H T T P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、I D C S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うときに、J W T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、I D C S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S A M L I D Pサポートをエクスポートする。

#### 【0 1 2 8】

図1 0は、一実施形態におけるI D C S内のS S O機能のシステムアーキテクチャビューのブロック図1 0 0 0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S S O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

10

20

30

40

50

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 I D C S 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 I D C S と統合するために Open I D Connect 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 Open I D 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 S A M L I D P サービス、外部 S A M L I D P に対してユーザ認証を調整するための S A M L S P 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 I D C S 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 I D C S S S O サービスなどの、 S S O のビジネスを実現する。一般的に、 H T T P は、 10 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。Open I D Connect も S A M L 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 / S S O を I D C S を通して実現するアプリケーションクライアントは、 I D C S において、 O A u t h 2 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 I D / パスワード、 I D / 証明書など）を取得する必要がある。

#### 【 0 1 2 9 】

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 O A u t h 2 1 0 0 4 および S A M L 2 1 0 0 6 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 S S O 1 0 0 8 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、 I D C S 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、 S S O サービス 1 0 0 8 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 O A u t h フローを必要とし Open I D Connect リレーパーティ（ relayin g party ）（「 R P 」、そのユーザ認証機能を I D P 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 1 0 1 0 、 2 者間 O A u t h フローを必要とし Open I D Connect R P として機能するネイティブアプリケーション 1 0 1 1 、および S A M L S P として機能するウェブアプリケーション 1 0 1 2 などである。 20

#### 【 0 1 3 0 】

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。 L D A P 、 S A M L 、および O A u t h 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。 L D A P 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。 S A M L 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、 S A M L 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。 O A u t h 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 O p e n I D Connect を提供する。 40

#### 【 0 1 3 1 】

図 10 の実施形態において、 O p e n I D アプリケーション 1 0 1 0 が I D C S 内の O p e n I D サーバに接続すると、その「チャネル」は S S O サービスを要求する。同様に、 50

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。

#### 【0132】

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。

#### 【0133】

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。

#### 【0134】

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問い合わせ、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。

#### 【0135】

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 OAuth 2 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

10

20

30

40

50

し、O A u t hマイクロサービス1 0 0 4は、要求されたトークン1 0 3 2をブラウザ1 0 0 2に与える。一実施形態において、ブラウザ1 0 0 2に与えられるトークン1 0 3 2は、J Wアイデンティティと、I D C S O A u t h 2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1 1を参照しながら開示される。

#### 【0 1 3 6】

一実施形態において、たとえば、O A u t hマイクロサービス1 0 0 4は、ネイティブアプリケーション1 0 1 1から認可要求を受け、2者間O A u t hフローに従ってユーザを認証する。この場合、O A u t hマイクロサービス1 0 0 4の認証マネージャ1 0 3 4は対応する認証を(たとえばクライアント1 0 1 1から受けたI D / パスワードに基づいて)実行し、トークンマネージャ1 0 3 6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。

10

#### 【0 1 3 7】

一実施形態において、たとえば、S A M Lマイクロサービス1 0 0 6は、ブラウザからS S O P O S T要求を受け、S A M L S Pとして機能するウェブアプリケーション1 0 1 2のユーザを認証する。S A M Lマイクロサービス1 0 0 6は次に、S A M L I D P 1 0 3 8として機能し、ブラウザ1 0 0 2をS S Oマイクロサービス1 0 0 8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S S Oセッションを有しているか否かに応じて、S S Oマイクロサービス1 0 0 8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、S S Oマイクロサービス1 0 0 8は、認証コンテキストをS A M Lマイクロサービス1 0 0 6に返す。そうすると、S A M Lマイクロサービスは、必要なトークンでS Pにリダイレクトする。

20

#### 【0 1 3 8】

一実施形態において、たとえば、S A M Lマイクロサービス1 0 0 6は、S A M L S P 1 0 4 0として機能してもよく、遠隔S A M L I D P 1 0 4 2(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)(「A D F S」)に進んでもよい。一実施形態は、標準S A M L / A Dフローを実現する。一実施形態において、S A M Lマイクロサービス1 0 0 6とS S Oマイクロサービス1 0 0 8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、S S Oマイクロサービス1 0 0 8は、H T M Lフォームを用いてユーザに問い合わせ、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。

30

#### 【0 1 3 9】

一実施形態において、I D C S内部のコンポーネント(たとえば1 0 0 4、1 0 0 6、1 0 0 8)と、I D C S外部のコンポーネント(たとえば1 0 0 2、1 0 1 1、1 0 4 2)との間の対話は、ファイアウォール1 0 4 4を通して行なわれる。

#### 【0 1 4 0】

##### ログイン / ログアウトフロー

図1 1は、一実施形態における、I D C Sによって提供されるS S O機能のメッセージシーケンスフロー1 1 0 0である。ユーザがブラウザ1 1 0 2を用いてクライアント1 1 0 6(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1 1 0 4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1 1 0 4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1 1 0 4は、ブラウザ1 1 0 2をO A u t h 2マイクロサービス1 1 1 0にリダイレクトすることにより、O A u t h 2マイクロサービス1 1 1 0に対するO p e n I D C o n n e c tログインフローを開始する(3者間A Z G r a n tフローであり、範囲=「openid profile」)。

40

#### 【0 1 4 1】

50

プラウザ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、プラウザ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。

#### 【0142】

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、プラウザ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はプラウザ1102にログインページを提供する。プラウザ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。

10

20

30

40

#### 【0143】

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ト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。

#### 【0144】

認証または検証が成功すると、SSOマイクロサービス1112は、プラウザ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に戻るようにプラウザ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）をプラウザ1102に戻しクラウドゲート1104にリダイレクトする。プラウザ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にスコーリングされる（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。

30

40

#### 【0145】

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、プラウザ1102をクライアント1106にリダイレクトする。すると、プラウザ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、プラウザ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、認証プロセスは繰り返される。

50

## 【0146】

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。

10

## 【0147】

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、同じプロセスが繰り返される。

## 【0148】

一実施形態において、SSO統合アーキテクチャは、プ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。

20

## 【0149】

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。

30

## 【0150】

クラウドキャッシュ

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トされ、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。

40

## 【0151】

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。

## 【0152】

50

一実施形態において、I D C S が提供するサービスを、L D A P 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、I D C S とのインターフェイスは、ファイアウォールを通り、H T T P（たとえばR E S T ）に基づく。典型的に、企業ファイアウォールは、内部L D A P 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）（「S S L」）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、T C P ポートがファイアウォールを通してエクスポート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、L D A P とH T T Pとの間の変換を行って、L D A P ベースのアプリケーションが、I D C S 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H T T P に対してオープンである。

#### 【 0 1 5 3 】

一般的に、L D A P 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。L D A P 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う各コールは、L D A P ディレクトリに対してユーザのデバイスを認証し、L D A P 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、L D A P 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。

#### 【 0 1 5 4 】

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I D C S 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、L D A P 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、I D C S 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。

#### 【 0 1 5 5 】

##### 分散型データグリッド

一実施形態において、I D C S 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2 0 1 6 / 0 0 9 2 5 4 0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。

#### 【 0 1 5 6 】

一実施形態において、I D C S 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J a v a ベースのインメモリデータグリッドである。Coherenceは、ピアトウピア（すなわち中央マネージャがない）インメモリ分散

10

20

30

40

50

型キャッシュを提供する。

【0157】

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。

【0158】

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）×86」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SSD1228a）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。

【0159】

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、サーバ間の单一ホップ通信を提供する单一のスイッチ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。

【0160】

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

10

20

30

40

50

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。

【0161】

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 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。

10

【0162】

アーキテクチャ

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」）、読み専用メモリ（「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な媒体を組み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。

20

【0163】

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。

30

【0164】

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。

40

【0165】

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。

【0166】

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

50

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。

【0167】

開示されている通り、実施形態は、マイクロサービスベースのアーキテクチャを実現することにより、クラウドベースのマルチテナントIAMサービスを提供する。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、中間層におけるマイクロサービスによって処理されるリアルタイムタスクと、メッセージキューにオフロードされるニア・リアルタイムタスクとに分割される。したがって、実施形態は、クラウドスケールのIAMプラットフォームを提供する。

【0168】

IAM機能の例

10

一実施形態において、IAM機能は、メモリに、または、その他のコンピュータ読取可能もしくは有形媒体に格納されたソフトウェアによって実現され、プロセッサによって実行される。

【0169】

アイデンティティ管理サービスの実行を求める要求を受ける。一実施形態において、この要求は、アイデンティティ管理サービスを特定するAPIに対するコールと、アイデンティティ管理サービスを実行するように構成されたマイクロサービスとを含む。一実施形態において、マイクロサービスは、その他のモジュール/マイクロサービスと通信することができる内蔵モジュールであり、各マイクロサービスは、他からコンタクト可能な無名ユニバーサルポートを有する。たとえば、一実施形態において、さまざまなアプリケーション/サービス602がIDCS APIに対してHTTPコールを行うことにより、図6に示されるIDCSマイクロサービス614を使用することができる。一実施形態において、マイクロサービスはランタイムコンポーネント/プロセスである。

20

【0170】

一実施形態において、上記要求はURLを含む。一実施形態において、マイクロサービスはURLのプレフィックスにおいて特定される。一実施形態において、URLのリソース(resource)部分がAPIを特定する。一実施形態において、URLのホスト(host)部分が、この要求に関連するリソースのテナンシーを特定する。たとえば、IDCSのウェブ環境における「host/microservice/resource」のようなURLにおいて、マイクロサービス(microservice)は、特定のURLプレフィックスを有することによって特徴付けられる。これはたとえば「host/oauth/v1」であり、実際のマイクロサービスは「oauth/v1」である。「oauth/v1」のもとで、複数のAPIが存在する。たとえば、トークンを要求するためのAPI「host/oauth/v1/token」、ユーザを認証するための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、同一のマイクロサービスのもとで複数の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>)。

30

【0171】

次に要求が認証される。一実施形態において、要求は、本明細書においてたとえば図6のウェブルーティング層610および/または図7のクラウドゲート702を参照しながら説明したように、クラウドゲートのようなセキュリティゲートによって認証される。

40

【0172】

次に、たとえば本明細書において図6のIDCS中間層614におけるマイクロサービスへのアクセスおよびIDCS「APIプラットフォーム」を参照しながら説明したように、マイクロサービスがアクセスされる。一実施形態において、マイクロサービスとの通信は、マイクロサービスの無名ユニバーサルポートを介して構成される。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トする標準通信チャネルであり(たとえば従来のHTTPポートのような)、同一サービス内の他の任意のモジュール/マイクロサービスがそれに対してトーカできるようにする

50

標準通信チャネルである。一実施形態において、マイクロサービスは、1つ以上のAPIをエクスポートすることによって1つ以上の機能を提供する。一実施形態において、マイクロサービスとの通信は、上記1つ以上のAPIを介してのみ実現される。すなわち、マイクロサービスには、このようなAPIにコールすることによってのみ、到達/コンタクトすることができる。一実施形態において、マイクロサービスとの通信は、軽量のプロトコルに従って構成される。一実施形態において、軽量のプロトコルはHTTPとRESTとを含む。一実施形態において、要求は、RESTful HTTP APIに対するコールを含む。したがって、一実施形態はディスパッチ機能を提供する。各HTTP要求はURIと動詞とを含む。本実施形態はURIからのエンドポイント(host/service/resource)をパースし、これをHTTP動詞(たとえば、POST、PUT、PATCH、またはDelete)と組み合わせることによって、適切なモジュールの適切な方法をディスパッチする(または呼び出す)。このパターンはRESTに共通であり各種パッケージ(たとえばJersey)によってサポートされる。

#### 【0173】

次に、たとえば本明細書において図6のIDCS中間層614におけるマイクロサービスへのアクセスおよびIDCS「APIプラットフォーム」を参照しながら説明したように、アイデンティティ管理サービスがマイクロサービスによって実行される。一実施形態において、マイクロサービスは、ステートレスであり、横方向にスケーラブルで、独立してデプロイ可能である。一実施形態において、マイクロサービスの各物理的実現は、複数のテナントを安全にサポートするように構成される。一実施形態において、アイデンティティ管理サービスは、ログインサービス、SSOサービス、連携サービス、トーケンサービス、ディレクトリサービス、プロビジョニングサービス、またはRbacサービスを含む。

#### 【0174】

##### IDCSのためのテナントおよびサービス管理

本発明の実施形態は、IDCSのためのテナントおよびサービス管理を有利に提供する。多くの実施形態において、タスク実行フレームワークが使用され、これは、タスクオーケストレータといくつかのタスクとを含む。タスクオーケストレータは、要求を検証し、同期的にインスタンス化し、タスクを制御し、タスクが完了すると応答を生成する。各タスクは、ロギング、イベントリング、およびメトリクス生成とともに、明確に定められた動作を実行する。状態データは、タスクコンテキストを介したこれらのタスクにわたる要求スコープ(request scoped)である。いずれかのタスクが失敗すると、先に実行されたタスクをロールバックし例外を伝達することができる。タスク実行フレームワークは、完全にスケーラブルで、アトミックで、マルチスレッドである。テナントおよびサービスは、このタスク実行フレームワークを用いてIDCSにおいて作成し管理することができる。

#### 【0175】

図13は、本発明の実施形態に従う、IDCSサービスインスタンスライフサイクル管理のシーケンスフロー1300を示す。このシーケンス図は、ユーザがIDCSをオーダーしたときに発生するプロセスフローを含む。要求は先ずグローバルシングルインスタンス(Global Single Instance)(「GSI」)システム1310にルーティングされる。これは、このサービスの費用をユーザに請求するグローバルeビジネス企業リソース計画(Enterprise Resource Planning)(「ERP」)システムであってもよい。GSIシステム1310内のモジュールは、IDCSの資格をユーザに提供するテナント自動化システム(「TAS」)1320に要求を送ることによってアカウントの作成を開始する。TAS1320内のモジュールは、この要求をIDCS1330に転送する。より具体的には、IDCS1330内のIDCSサービスマネージャモジュールが要求を受け入れ、DB、LDAPなどの観点から、IDCSの資格をユーザに提供する。IDCSサービスマネージャモジュールは、RESTベースのAPIのセットを有する。アカウントごとに仮想ロードバランサを作成するために、IDCSサービスマネージャモジュールは、Lbab1340と通信する。IDCSサービスインスタンスごとに3種類のURLを作成することができ、これらは、たとえば、パーマネントURL、フレンドリーURL、および

10

20

30

40

50

バニティ (vanity) URL を含む。これらの URL は、テナントの作成が完了すると、IDCS サービスマネージャモジュールから TAS1320 に伝えられる。各種実施形態は、既存の IDCS サービスインスタンスについて、GET、PATCH および DELETE という動作を動的に支援する。

#### 【 0176 】

一般的に、周知のシステムは、新たなユーザーアカウントを作成するためにはアドミニストレータによるマニュアル介入を必要とする。しかしながら、本発明の実施形態は、マニュアル介入無しでユーザのアイデンティティシステムを動的かつ自動的に作成することを可能にするアーティファクトを有利に作成する。IDCS1330においてユーザーアカウントが作成されると、ユーザは、他のサービスを、たとえばオラクルクラウドのようなクラウドからも得ることができる。IDCS1330は、作成されたすべてのアカウントについて新たなアーティファクトを作成するマルチテナントアプリケーションであり、この新たなアーティファクトは、たとえば、ロードバランサ内のルーティング URL、キー、証明書などのようなセキュリティアーティファクト、および、たとえばデータベース、LDAP などのようなデータ層内のアーティファクトである。すべてのユーザーアカウントについて、ユーザ、グループ、アプリケーション、アプリロール、アプリロールメンバーシップおよびグループを作成することもできる。

10

#### 【 0177 】

特定の実施形態において、ネームスペースとしてのテナントは、マルチテナントを提供する主要なメタデータである。アカウントの作成中に、IDCS サービスマネージャは、異なるシステムにおいてユーザのネームスペースを作成する。たとえば、IDCS サービスマネージャモジュールは、LBaaS1340においてルーティング URL をプロビジョニングし、データベースにおいてスキーマを作成し、LDAP において対応するコンテナを作成し、ストレージにおいて対応するコンテナを作成し、通知用の通知コンテナを作成するといったことである。したがって、IDCS サービスマネージャモジュールは、ユーザのためのいかなるアイデンティティおよびアクセス管理動作についても基礎を提供する。たとえば、ユーザは、IDCS サービスマネージャモジュールが作成したコンテナに行くことができる。アクセスのために、IDCS サービスマネージャモジュールがプロビジョニングし作成した暗号化キーを有するクッキーが作成される。

20

#### 【 0178 】

一実施形態において、IDCS1330 がアイデンティティおよびアクセス管理サービスを提供しテナントメタデータおよびアーティファクトを提供すると、その他のクラウドサービスは、この IDCS アイデンティティおよびアクセスサービスを消費することができるが、IDCS サービスの内部アーティファクトおよびメタデータを消費することはできない。

30

#### 【 0179 】

一実施形態において、TAS1320 は、IDCS サービスマネージャモジュールと通信するために REST API を呼び指す。これらの API を呼び出すために、TAS1320 は、ウォレットを有し、このウォレットからアクセストークンを引き出す。ユーザが IDCS1330 をサブスクライブすると、IDCS クラウドインフラテナント (infrastructure) と呼ばれるテナントが作成され、ウォレットに入れるアクセストークンを提供する。このウォレットはまた、アクセストークンにアクセスするアサーショントークンを有する。TAS1320 はこのアサーショントークンに、したがってアクセストークンにアクセスすることにより、IDCS API にコールする権利を取得する。実施形態は、(外部ではなく) 内部の顧客であり安全な API ベースの統合を有する TAS1320 と IDCS1330 との間の新規のセキュリティメカニズムを提供する。TAS1320 は、アサーショントークンに、したがってウォレット内のアクセストークンにアクセスできるので、IDCS API にコールするとき、メッセージヘッダ内のアクセストークンを送ることができる。IDCS API にコールするとき、TAS1320 は、キーストア (keystore) と呼ばれるパラメータを含むペイロードを生成する。次に、このキースト

40

50

アは、データベース内のテナントスキーマに格納される。よって、キーストアを安全に伝えるための新規のメカニズムが提供される。

#### 【0180】

IDSサービスマネージャモジュールは、中間層におけるIDS1330のマイクロサービスとして、テナントおよびサービスプロビジョニング調整を実行する。IDS1330は、与えられたサービスがまだ提供されていないときに、ブートストラップテナントを、データベースのみを用いてプロビジョニングする。ディレクトリサーバは任意である。このインフラテナントが作成されると、中間層におけるサービス、クラウドゲート、およびその他のIDSコンポーネントは、たとえばキーなどのインフラテナントアーティファクトの使用を開始する。サービスが準備されて実行されると、TAS1320は、RESTベースの呼び出しを実行することにより、正規のテナントを作成することができる。

10

#### 【0181】

図14は、本発明の実施形態に従う、IDSのためのテナントおよびサービス管理を提供するプロセス1400を示す。テナントに関するIDSサービスインスタンスの作成の要求を、TAS1320からネットワークを介して受ける(1410)。タスクオーケストレータといくつかの同期タスクとを含むタスク実行フレームワークを用いてIDSサービスインスタンスを作成する(1420)。タスクオーケストレータは、タスクをインスタンス化し、タスクを制御し、要求に対する応答を生成する。この応答は、テナントのためのIDSサービスインスタンスが作成されたことを示し、テナント固有のデータを含み得る。次に、応答をネットワークを介してTAS1320に送る(1430)。

20

#### 【0182】

特定の実施形態において、IDSサービスインスタンスを作成することは、ロードバランシングポリシーを構成することと、ルーティングURLを作成することと、テナントのためのネームスペースをプロビジョニングすることとを含み得る。ルーティングURLは、パーマネントURL、フレンドリーURL、およびバニティURLを含み得る。有利に、IDSサービスインスタンスを、予め作成された複数のスキーマから選択された予め作成されたスキーマを用いて作成することができる。

#### 【0183】

図15は、本発明の実施形態に従う、IDSサービスインスタンス作成要求を受けたときに実行されるプロセス1500を示す。先ず、ペイロードを受け、要求の検証を、たとえば、必須のパラメータ、任意のパラメータ、フォーマットなどに対して実施する。次に、IDSサービスインスタンスのためのLBAaSポリシーを作成することにより、ルーティング層において所与のアカウントをセットアップする。このプロセスは、アドミニストレータによるマニュアル介入を受けることなく動的に実行することができる。たとえば、ユーザは、アドミニストレータがIDSにおいてユーザを構成するのを待つ必要なく、作成された新たなアカウントを有することができる。

30

#### 【0184】

次に、IDSサービスインスタンスのためにLDAPが構成され、IDSサービスインスタンスのためにアーティファクトがデータベース内に作成される(すなわち、テナントスキーマブートストラッププロセス)。次に、リソースアーティファクトがスキーマにロードされる(スタティックデータおよびダイナミックデータ)。次に、IDSサービスインスタンスのためにキーストアが作成され、後にOAuthトークンを発見するためには使用される、特定のテナント用の証明書などのアーティファクトがインポートされる。次に、テナントクレジットストア(たとえばクレジットまたはパスワードを格納)およびSAMLプロバイダIDが作成される。次に、一組の特権(たとえば管理)を特定しアプリロールのメンバーを構成してユーザがこれらの特権を受けられるようにするアプリロールが作成される。次に、たとえば、IDS管理コンソールUIに表示することができる、オラクルヘッダーロゴ、オラクルフッターロゴなどのようなIDSサービスインスタンスのためのローディングGUIロゴまたはローディングPDFテンプレートを含む、ス

40

50

トレージアーティファクトが与えられる。

【0185】

次に、言語などのようなデフォルト設定が与えられる。次に、アプリクライアントである OAuthクライアントが作成される。ユーザがIDCSサービスインスタンスにアクセスすることを所望する場合、ユーザは、OAuthクライアントにおいて利用できるOAuthトークンを用いることによってそうすることができる。次に、管理ユーザが作成されアドミニストレータ特権が割り当てられる。ユーザおよびグループ（たとえば、システムIDグループ、読み／書きユーザグループなど）も与えられ、よって、IDCSサービスインスタンスが利用できるようになると直ちにこれらのユーザは動作を実行することができる。テナントPaaSおよびSaasアドミニストレータが次に与えられ、スタティックおよびダイナミックアーティファクトがロードされる。

10

【0186】

次に、テナントのためのIDCSサービスインスタンスが作成されたことを示す応答が生成される。この応答は、テナント固有のデータも含み得る。たとえば、プロビジョニングされているIDCSにおける各サービスインスタンスは、グローバル意識別子（Global Unique Identifier）（「GUID」）によって表され、この情報は、TAS1320に返すように伝達することができる。テナント固有のデータは、ユーザとの通信に使用されるルーティングURLも含み得る。特定の実施形態において、IDCSサービスインスタンスが作成されると、ユーザは管理コンソールにログインしそのパスワードをリセットしなければならない。管理コンソールのURLも応答の詳細に含まれ得る。提供され得る別のURLは、TAS1320に伝えられるサービスインスタンスURLである。TAS1320は、このURLを用いて、たとえば、GET、PATCH、DELETEといったさまざまな動作をIDCSサービスインスタンスに対して実行することができる。

20

【0187】

これらのタスクのうちのいずれかが失敗した場合、このプロセスは、ロールバックし、フットプリントが残らずユーザがリトライできるように、（たとえばLDAPにおいて、スキーマにおいて、など）作成されたものをすべて削除する。特定の実施形態において、このフローは、テナントアドミニストレータの作成に使用されてもよい。このフローの最後に、テナントアドミニストレータは、IDCSサービスマネージャから提供された情報を含むウェルカム電子メールを受ける。

30

【0188】

特定の実施形態において、図15の各ブロックは、独立したタスクを示す。多数のテナントを同時に作成することができる。

【0189】

より具体的には、本発明の実施形態に従い、タスク1510によって動作「JSON要求検証（JSON Request Validation）」が実行される。タスク1520によって動作「TenantLBAASBootstrap」が実行される。タスク1521によって動作「TenantIdentityBaseCreator」が実行される。タスク1522によって動作「TenantSchemaBootstrap」が実行される。タスク1523によって動作「TenantResourceArtifactsGenerator」が実行される。タスク1524によって動作「TenantKeyStoreGenerator」が実行される。タスク1525によって動作「RegularTenantKeyStoreImporter」が実行される。タスク1526によって動作「TenantCred[ential]StoreCreator」が実行される。タスク1527によって動作「SAMLProviderIDCreator」が実行される。タスク1528によって動作「BootstrapIDCSApplication」が実行される。タスク1529によって動作「SeedStorageArtifacts」が実行される。タスク1530によって動作「SeedDefaultSetting」が実行される。タスク1531によって動作「TenantAppClientCreator」が実行される。タスク1532によって動作「TenantAdminUserGroupCreatorForRegularTenant」が実行される。タスク1533によって動作「TenantAdminUserGroupAssignerForRegularTenant」が実行される。タスク1534によって動作「SeedTenantContainersUsersGroups」が実行される。タスク1535によって動作「TenantService

40

50

`eContextCreator` が実行される。タスク 1536 によって動作「`TenantAppIdContainerCreator`」が実行される。タスク 1537 によって動作「`SeedTenantPaaSaaSAdm inistrators`」が実行される。タスク 1538 によって動作「`SeedSIMArtifacts`」が実行される。タスク 1540 によって動作「応答生成 (Response Generation)」が実行される。

#### 【0190】

失敗の結果として、いくつかのタスクによって実行される動作を含む I D C S サービスインスタンスの削除 (deletion) が生じる可能性がある。より具体的には、本発明の実施形態に従い、タスク 1550 によって動作「`DeleteTenantContainer`」が実行される。タスク 1551 によって動作「`DeleteTenantSchema`」が実行される。タスク 1552 によって動作「`DeleteLBaaSVLBRs`」が実行される。

10

#### 【0191】

図 16 は、本発明のある実施形態に従う、タスク実行フレームワークのブロック図 1600 を示す。タスク実行フレームワークはタスクオーケストレータを有し、タスクオーケストレータは、要求を検証し、いくつかのタスクをインスタンス化および制御し、タスクが完了すると応答を生成する。各タスクは、ロギング、イベントリング、およびメトリクス生成とともに、明確に定められた動作を実行する。状態データは、タスクコンテキストを介したこれらのタスク全体にわたる要求スコープ (request scoped) である。いずれかのタスクが失敗すると、先に実行されたタスクをロールバックし例外を伝達することができる。タスクパフォーマ (task performer) フレームワークは、完全にスケーラブルで、アトミックで、マルチスレッドである。このフレームワークを用いてテナントおよびサービスを I D C S において作成することができる。

20

#### 【0192】

特定の実施形態において、テナントが作成されると、スキーマおよびそのアーティファクトが作成される。多くの実施形態は、予め作成された複数のスキーマを提供する。

#### 【0193】

特定の実施形態において、I D C S サービスインスタンスを作成するタスク実行フレームワークと同一のタスク実行フレームワークを用いて既存のテナントをアップグレードすることもできる。たとえば、新たなバージョンの I D C S 1330 に新たな特徴が導入されると、テナントアーティファクトを、タスクリストにおいてともにリンクされたタスクによってアップグレードすることができる。テナントアーティファクトのアップグレードは同期動作である。先ずアップグレードを必要とするテナントのリストがデータベースから探し出され、その後アップグレードされる。テナントごとに、新たなバージョンの I D C S 1330 に固有のタスクのリストが同一のタスク実行フレームワークを用いて実行される。各タスクは完全にリエンタント (re-entrant) であり、したがって、テナントのアップグレードは、テナントごとに 2 回以上試行される場合がある。アップグレード中、I D C S 1330 のサービスのいずれについても必要なダントンタイムはなく、テナントは、アドミニストレータによるマニュアル介入を受けることなくシームレスにアップグレードされる。

30

#### 【0194】

図 17 は、本発明のある実施形態に従う、テナントアップグレード機能のブロック図を示す。

40

#### 【0195】

図 18 は、本発明のある実施形態に従う、スキーマ作成についてのブロック図 1800 を示す。I D C S サービスインスタンスの作成を求める要求を受けると、テナントが、予め作成されている特定のテナントスキーマにマッピングされる。特定の実施形態において、予め作成されているスキーマは、柔軟性を与えるために、3 つのサイズ (低 (たとえば 9 G B )、中 (たとえば 39 G B ) および高 (たとえば 119 G B )) で提供される。たとえば基本企業、標準企業、および標準非企業といった、I D C S サービスインスタンスの、対応する 3 種類の資格を提供することができる。各資格は、データベースサイズのうち

50

の1つにマッピングされ、スキーマは、受けた資格要求に基づいて選択することができる。このプロセスは、I D C S サービスマネージャによりバックエンドにおいてオンザフライでシームレスに実行され、ユーザにとってはトランスペアレントである。

【0196】

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。

10

20

30

40

50

## 【図面】

### 【図 1】

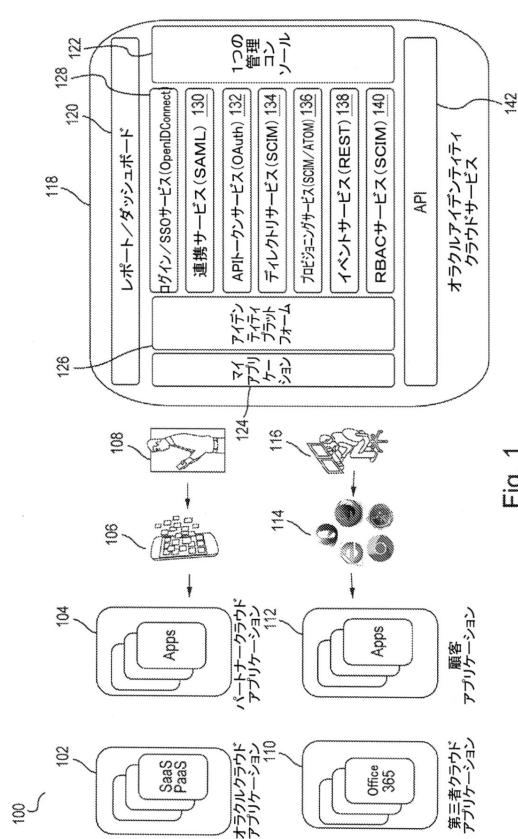


Fig. 1

【 図 2 】

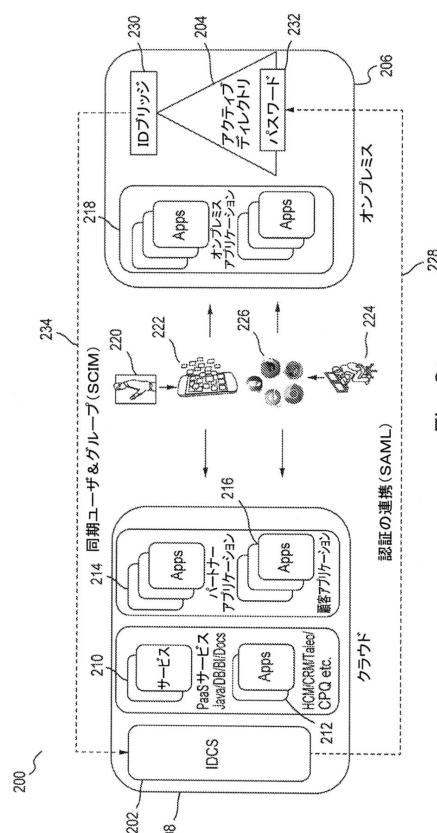
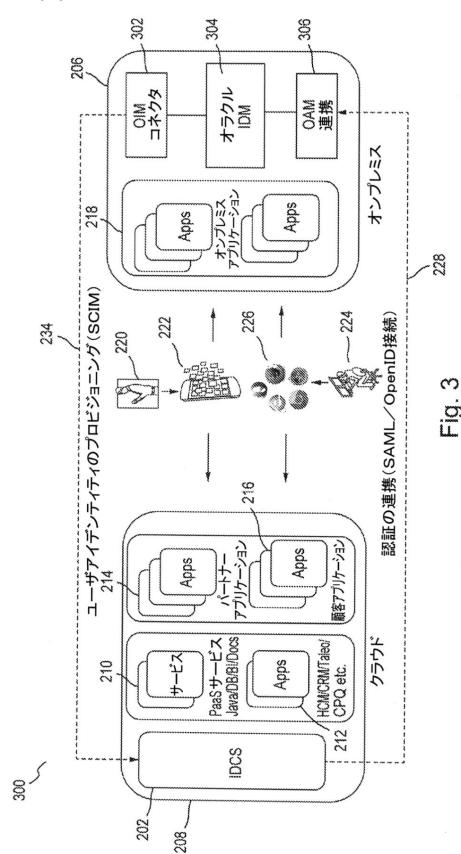


Fig. 2

【 义 3 】



3

【 4 】

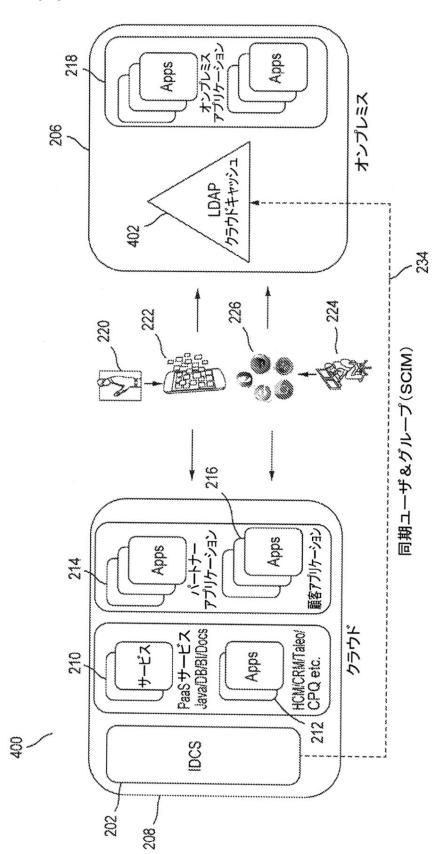
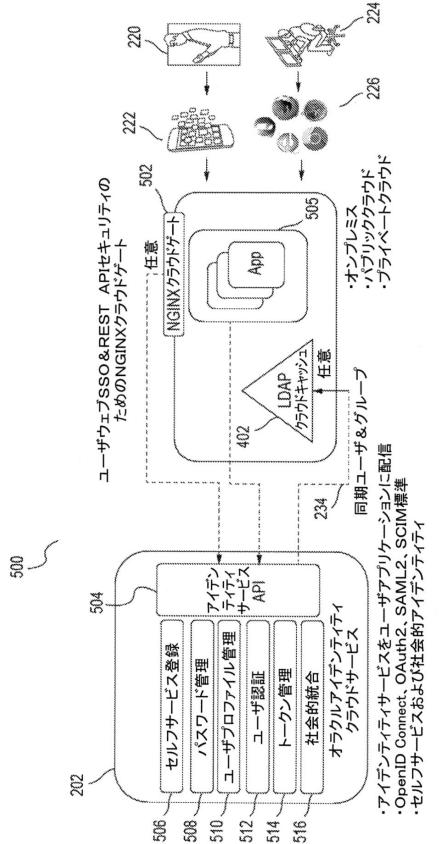


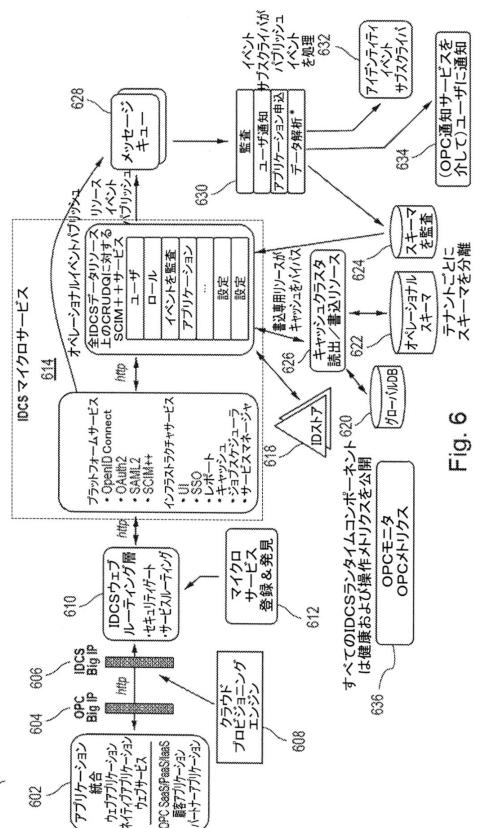
Fig. 4

【図5】



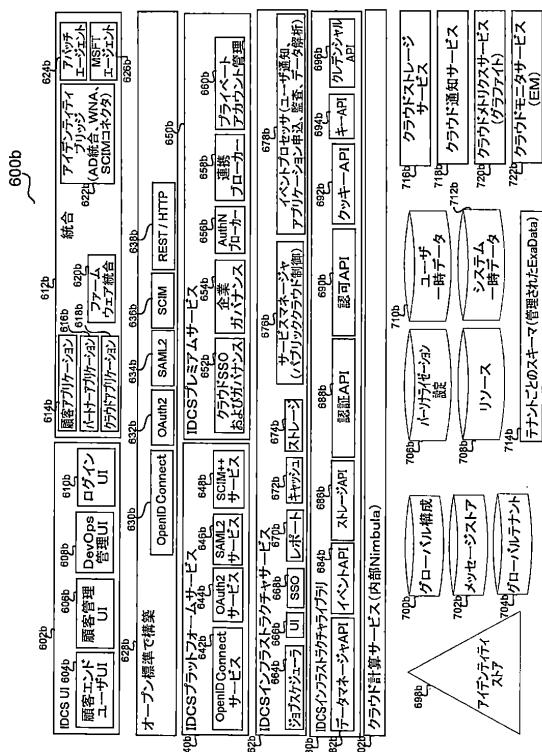
५०

【 四 6 】



6

【図 6 A】



【図7】

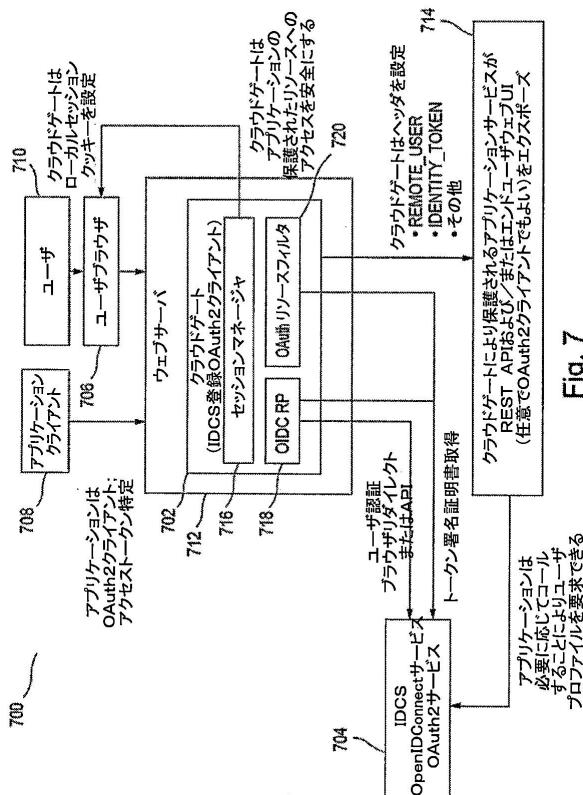


Fig. 7

【図8】

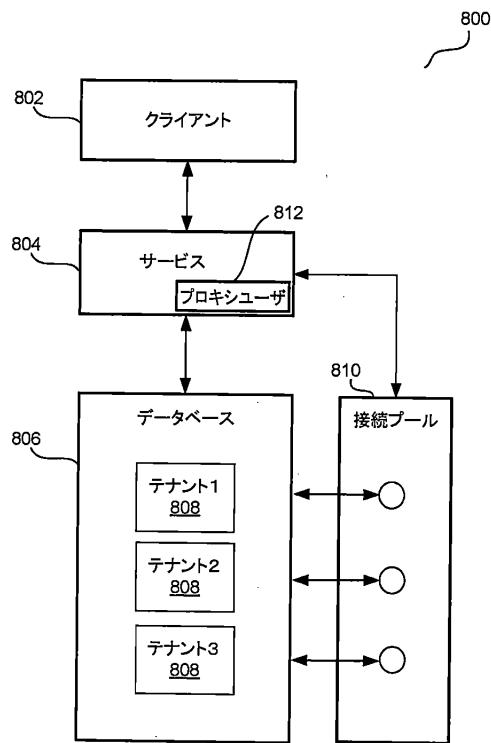


Fig. 8

【図9】

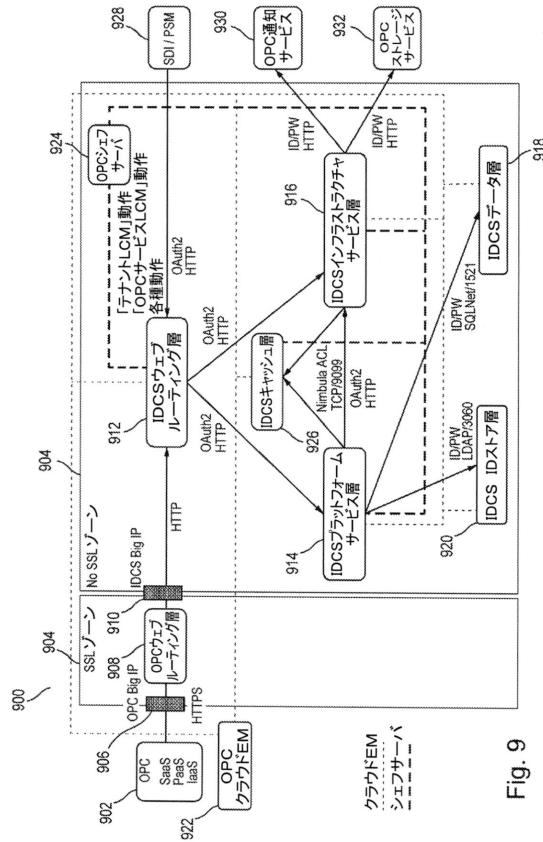


Fig. 9

【図10】

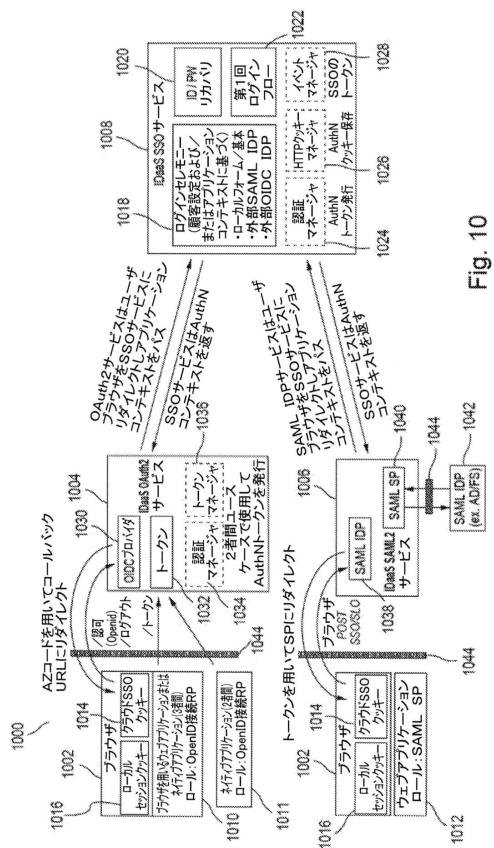


Fig. 10

【図11】

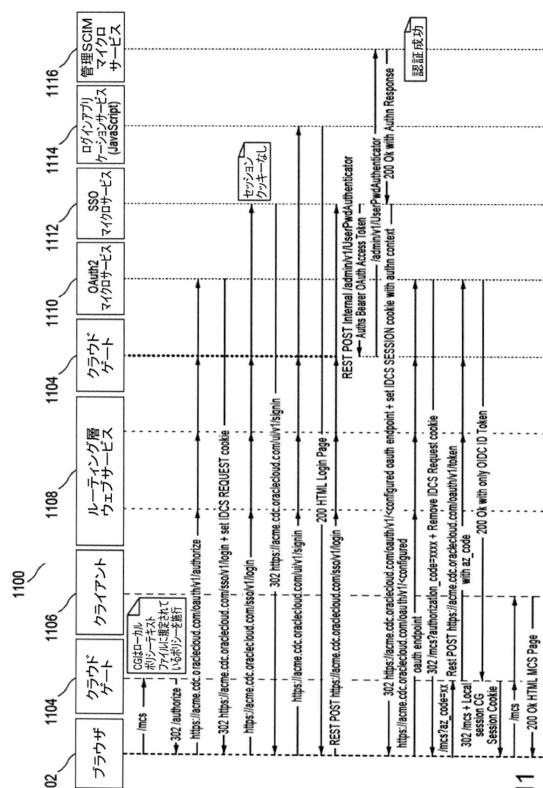


Fig. 11

【図12】

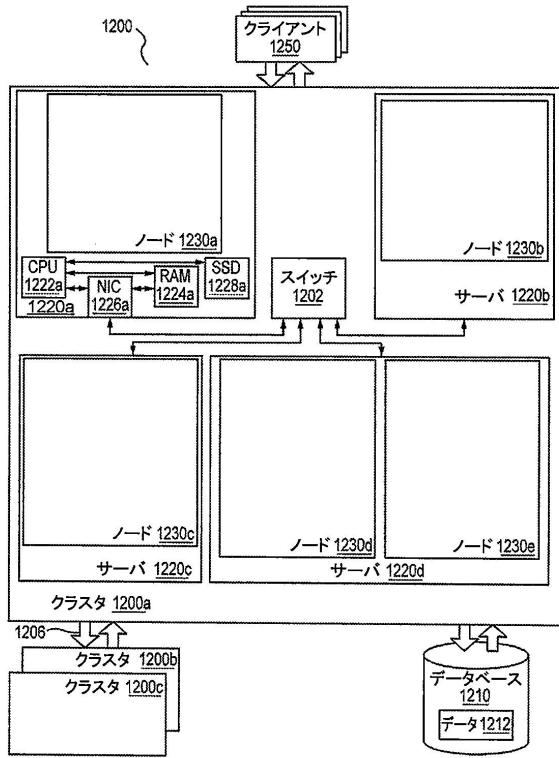


Fig. 12

【 図 1 3 】

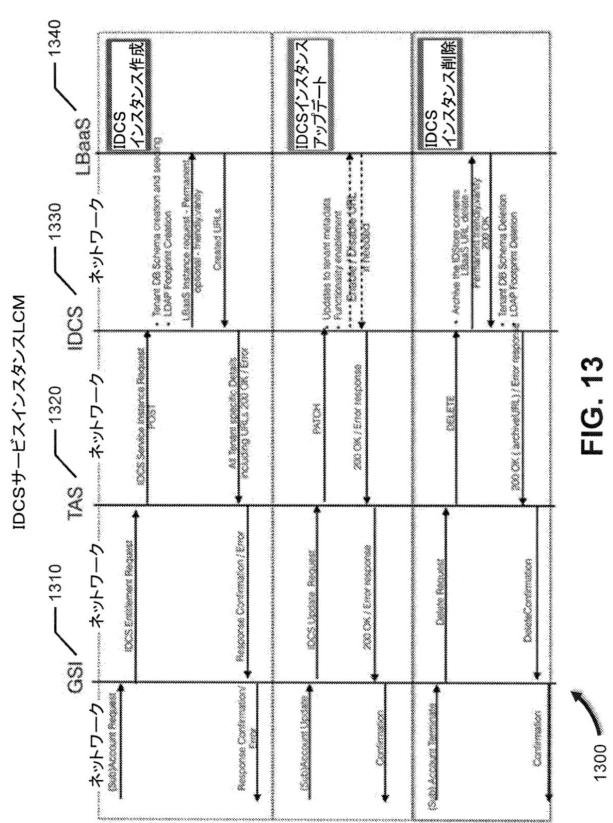
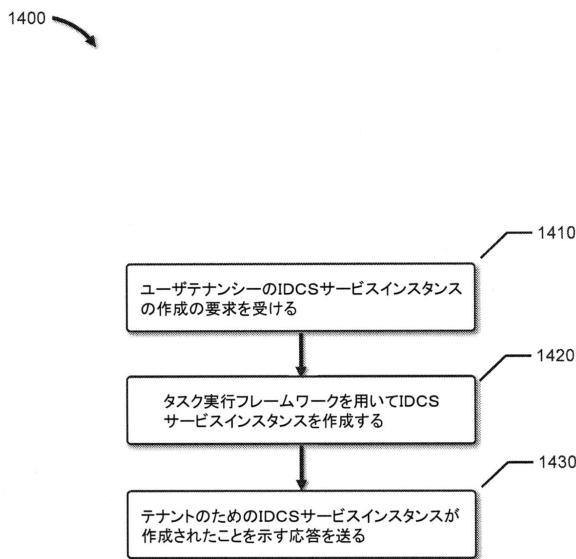


FIG. 13

### 【図14】



**FIG. 14**

【図15】

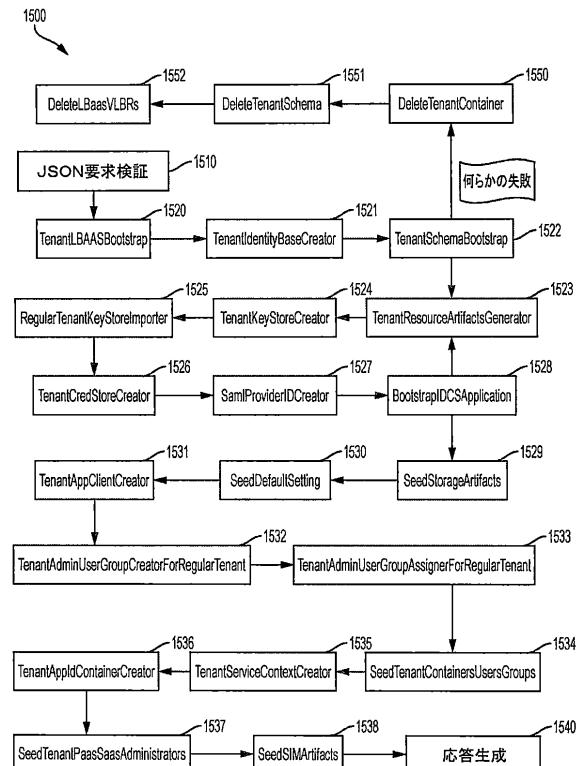
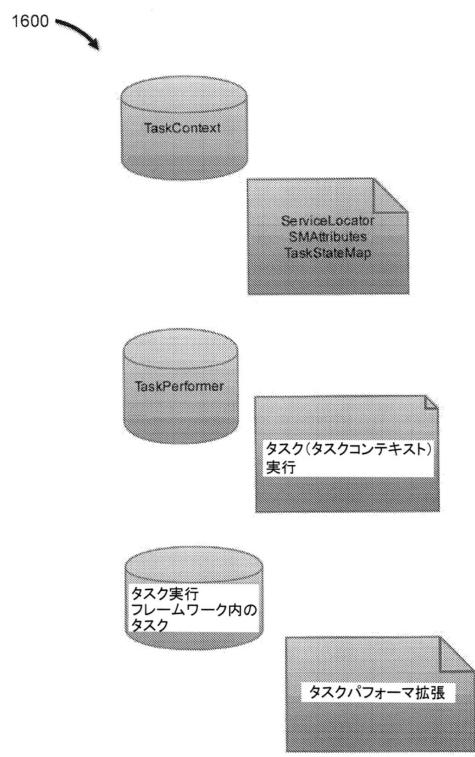


FIG. 15

### 【図16】



【図17】

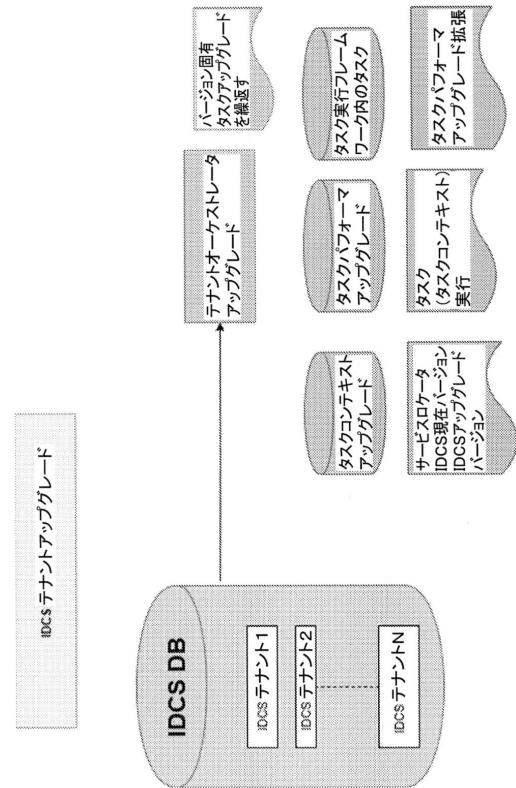
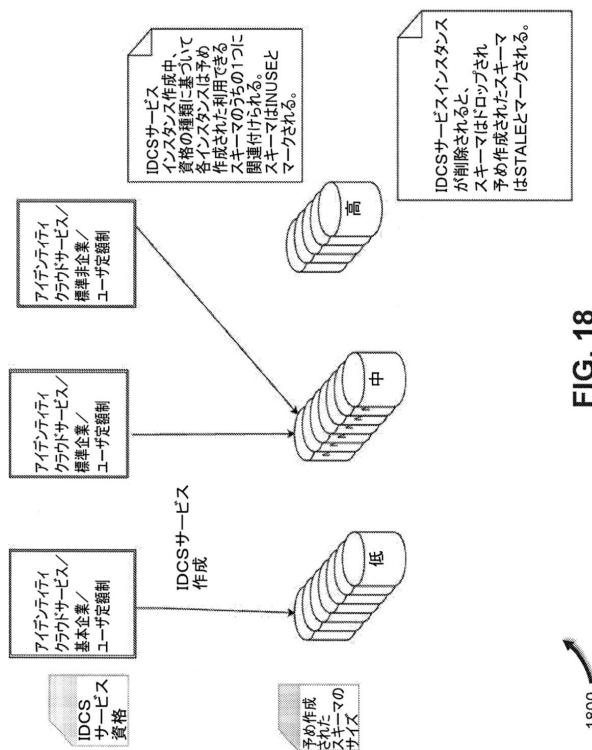


FIG. 16

【図18】



18

10

20

30

40

50

FIG. 17

---

フロントページの続き

チャムラージペット、ファースト・メイン・ロード、84/1

(72)発明者 プルショサマン, ラジエシュ

アメリカ合衆国、94555 カリフォルニア州、フリーモント、ウェルマン・テラス、3438  
6、アーデンウッド・フォレスト・コンドミニアムズ

審査官 甲斐 哲雄

(56)参考文献 特表2015-534674 (JP, A)

米国特許出願公開第2016/0142399 (US, A1)

特開2012-103846 (JP, A)

米国特許出願公開第2015/0172139 (US, A1)

特表2017-524202 (JP, A)

(58)調査した分野 (Int.Cl., DB名)

G06F 16/10 - 16/188

G06F 9/445