

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第6719445号  
(P6719445)

(45) 発行日 令和2年7月8日 (2020.7.8)

(24) 登録日 令和2年6月18日 (2020.6.18)

(51) Int.Cl.	F I
GO 6 F 16/23 (2019.01)	GO 6 F 16/23
GO 6 F 16/27 (2019.01)	GO 6 F 16/27
GO 6 F 16/182 (2019.01)	GO 6 F 16/182

請求項の数 14 (全 23 頁)

(21) 出願番号	特願2017-502609 (P2017-502609)	(73) 特許権者	314015767
(86) (22) 出願日	平成27年7月13日 (2015.7.13)		マイクロソフト テクノロジー ライセンシング, エルエルシー
(65) 公表番号	特表2017-531844 (P2017-531844A)		アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ
(43) 公表日	平成29年10月26日 (2017.10.26)		
(86) 国際出願番号	PCT/US2015/040084	(74) 代理人	100118902
(87) 国際公開番号	W02016/010873		弁理士 山本 修
(87) 国際公開日	平成28年1月21日 (2016.1.21)	(74) 代理人	100106208
審査請求日	平成30年7月13日 (2018.7.13)		弁理士 宮前 徹
(31) 優先権主張番号	14/332,509	(74) 代理人	100120112
(32) 優先日	平成26年7月16日 (2014.7.16)		弁理士 中西 基晴
(33) 優先権主張国・地域又は機関	米国 (US)	(74) 代理人	100138759
前置審査			弁理士 大房 直樹

最終頁に続く

(54) 【発明の名称】 リアル・タイム・プロビジョニングのための動的更新ストリーム

(57) 【特許請求の範囲】

【請求項 1】

マルチテナント・システムにおける複数のサーバー・インスタンスであって、前記サーバー・インスタンスは、複数の異なるテナントにサービスを供与し、各テナントは、異なる組織に対応し、かつ対応するテナント・データを有し、前記マルチテナント・システムにおける前記サーバー・インスタンスは、各テナントに対応する前記テナント・データを他のテナントから分離する、複数のサーバー・インスタンスと、

複数の異なる要求元テナントの各々から変更要求を受け取るように構成された通知ストリーム・ジェネレーターであって、前記通知ストリーム・ジェネレーターは、前記受け取られた変更要求をテナント毎セグメントに細分化するように構成された変更細分化コンポーネントを備え、前記テナント毎セグメントの各々は、前記要求元テナントのうちの特定の1つに対応するとともに、前記特定の要求元テナントに対応するテナント・データへの変更を示す複数の変更要求の組を含む、通知ストリーム・ジェネレーターと、

を備えるコンピューター・システムであって、

前記通知ストリーム・ジェネレーターは、前記テナント毎セグメントの各々に対して、前記要求元テナントのうちの前記特定の1つに割り当てられた1つのサーバー・インスタンスに通知するように構成され、前記通知されたサーバー・インスタンスの各々は、当該サーバー・インスタンスが通知を受けた前記テナント毎セグメントにおける全ての変更要求を行い終えるまで、前記通知ストリーム・ジェネレーターとの接続を維持し、その後、前記接続を解放する、

10

20

コンピューター・システム。

【請求項 2】

前記マルチテナント・システムはクラウド・ベース・システムである、請求項 1 に記載のコンピューター・システム。

【請求項 3】

前記通知ストリーム・ジェネレーターは、前記テナント毎セグメントの各々に対して、対応するサーバー・インスタンスが前記テナント毎セグメントにおける変更要求の全てを行うまで、前記テナント毎セグメントにおける一組のテナント変更を前記対応するサーバー・インスタンスへ通信するための排他的同期ストリームを形成するように構成され、前記排他的同期ストリームは、前記テナント・データへのテナント変更の組の通信を排他的に行う、請求項 1 に記載のコンピューター・システム。

10

【請求項 4】

前記通知ストリーム・ジェネレーターは、保留変更ストアを備え、前記変更細分化コンポーネントは、前記受け取られた変更要求を前記テナント毎セグメントに細分化し、前記テナント毎セグメントを前記保留変更ストアに格納するように構成される、請求項 1 に記載のコンピューター・システム。

【請求項 5】

前記通知されたサーバー・インスタンスの各々は、通知を受けた前記複数の変更要求の組に対応する所与のテナント毎セグメントからの全ての変更が引き出され終わるまで、前記通知ストリーム・ジェネレーターとの接続を維持し、前記所与のテナント毎セグメントから前記変更を引き出す、請求項 4 に記載のコンピューター・システム。

20

【請求項 6】

前記通知ストリーム・ジェネレーターは、前記保留変更ストア内のテナント毎セグメントに対応するブックマークを所与のサーバー・インスタンスの各々に提供することによって、前記所与のサーバー・インスタンスに通知する、請求項 5 に記載のコンピューター・システム。

【請求項 7】

前記所与のサーバー・インスタンスは、前記変更要求を引き出して前記変更要求を行うときに、個々のサーバー・インスタンス毎に、リソース使用制約を強要する、請求項 6 に記載のコンピューター・システム。

30

【請求項 8】

追加の要求元テナントから受け取られた変更要求の追加の各組に対して、前記通知ストリーム・ジェネレーターは、前記変更要求の追加の組を処理するように追加のサーバー・インスタンスに通知する、請求項 7 に記載のコンピューター・システム。

【請求項 9】

前記サーバー・インスタンスの各々は、別個の物理サーバー上に実装されたサーバー・インスタンスを備える、請求項 8 に記載のコンピューター・システム。

【請求項 10】

複数の異なるテナントにサービスを供与し、前記異なるテナントの各々に対する別個の分離されたテナント・データを維持するマルチテナント・システムを備えるコンピューター・システムであって、

40

前記マルチテナント・システムは、

テナント・データにおける変更要求を処理する複数の異なるサーバーと、

対応する要求元テナントから前記変更要求を受け取るように構成された通知ストリーム・ジェネレーターであって、前記通知ストリーム・ジェネレーターは、前記受け取られた変更要求をテナント毎セグメントに細分化するように構成された変更細分化コンポーネントを備え、前記テナント毎セグメントの各々は、前記要求元テナントのうちの特定の 1 つに対応するとともに、前記特定の要求元テナントに対応するテナント・データへの変更を示す複数の変更要求の組を含む、通知ストリーム・ジェネレーターと、

を備え、前記通知ストリーム・ジェネレーターは、各要求元テナントに、前記複数の異

50

なるサーバーのうちの1つのサーバーであって対応するテナント毎セグメントにおける変更要求を処理するサーバーを割り当てるように構成され、前記割り当てられたサーバーは、前記通知ストリーム・ジェネレーターが他のテナントに対する変更要求を処理することを当該サーバーに割り当てる前に、当該サーバーが割り当てられた対応するテナントに対するテナント毎セグメントにおける全ての変更要求を処理する、

コンピューター・システム。

【請求項11】

前記複数の異なるサーバーは、複数の異なるサーバー・インスタンスを備える、請求項10に記載のコンピューター・システム。

【請求項12】

前記通知ストリーム・ジェネレーターは、前記テナント毎セグメントの各々に対して、対応するサーバー・インスタンスが前記テナント毎セグメントにおける変更要求の全てを行うまで、前記テナント毎セグメントにおける一組のテナント変更を前記対応するサーバー・インスタンスへ通信するための排他的同期ストリームを形成するように構成され、前記排他的同期ストリームは、前記テナント・データへのテナント変更の組の通信を排他的に行う、請求項10に記載のコンピューター・システム。

【請求項13】

前記要求元テナントに割り当てられた前記サーバーは、前記要求元テナントに対応する前記セグメントから前記テナント変更を引き出し、前記要求元テナントに対応する前記テナント・データにおいて前記テナント変更を行い、前記セグメントからの前記テナント変更の全てが前記要求元テナントに対応する前記テナント・データに対して行われたときに、前記通知ストリーム・ジェネレーターに通知する、請求項10に記載のコンピューター・システム。

【請求項14】

前記テナント変更の組を格納する保留変更ストアを更に備え、前記変更細分化コンポーネントは、前記保留変更ストアにおける前記テナント変更の組を細分化する、請求項10に記載のコンピューター・システム。

【発明の詳細な説明】

【背景技術】

【0001】

[0001] コンピューター・システムは現在広く使用されている。あるコンピューター・システムはローカル・コンピューター・システムであり、一方他のコンピューター・システムはリモート・サーバー環境において使用される。

【0002】

[0002] 会社または他の組織が、コンピューター・システムのローカル（オンプレミス）処理系（implementation）とコンピューター・システムのリモート・サーバー処理系（クラウド・ベース処理系のような）との間で切り替えるのは、珍しいことではない。一例として、会社は、それらの電子メール・システム、それらの文書共有システム、またはそれらの業務システムのローカル処理系とリモート・サーバー処理系との間でときどき切り替える。業務システムのいくつかの例には、とりわけ、企業リソース計画（ERP）システム、顧客関係管理（CRM）システム、取扱商品（LOB：line-of-business）システムが含まれる。これらは、会社がローカルのオンプレミス処理系とリモート・サーバーまたはクラウド・ベース処理系との間で切り替えるタイプのコンピューター・システムのいくつかの例に過ぎない。

【0003】

[0003] 同様に、組織の中には、混成処理系を有するものもある。このような処理系では、サービスの一部がコンピューター・システムのローカルのオンプレミス・コンポーネントによって実行され、他のサービスがリモート・サーバーまたはクラウド・ベース環境において実行される。混成システムでは、組織が周期的に一定のサービスをオンプレミス処理系からクラウド・ベース処理系に移動しつつ、他のサービスはオンプレミスに維持す

10

20

30

40

50

ることが珍しくない。

【 0 0 0 4 】

[0004] リモート・サーバーまたはクラウド・ベース処理系、あるいは混成処理系を有する会社の中には、比較的大規模なものもある。企業の組織は、たとえば、何千人もの従業員を有する場合もある。したがって、これらのコンピューター・システムのリモート・サーバーまたはクラウド・ベース処理系は、多数の個人に対応(serve)しなければならない。コンピューター・システムに対して行われるトランザクションおよび変更の多くは、多数のユーザー・アカウントまたはユーザー・データの変更を必然的に伴う。

【 0 0 0 5 】

[0005] 一例として、企業組織がある計算システム機能をオンプレミス処理系からクラウド・ベース処理系に移動することを望む場合、これは企業のコンピューター・システムに多くの異なる更新を行うことを必要とする可能性がある。具体的な例として、企業がその電子メール・システムを移動させている場合、これは、多数の従業員アカウントの作成を必然的に伴う可能性がある。アカウントの作成は、多くの場合、シリアルで行われ、大量の時間を要する可能性がある。加えて、多数の変更が行われる場合、これらの変更は、大量の処理およびメモリー・オーバーヘッド、ならびにリモート・サーバーまたはクラウド・ベース処理系の帯域幅を消費する可能性がある。

【 0 0 0 6 】

[0006] 多くのクラウド・ベースまたはリモート・サーバー処理系は、マルチテナント・システムでもある。即ち、これらは複数の異なるテナントにあるレベルのサービスを提供したが、テナントが複数の異なる組織であることが多い。1つのテナントが多数の変更を行うと、他のテナントによって感知される処理能力に悪影響を及ぼす可能性がある。

【 0 0 0 7 】

[0007] 以上の論述は、一般的な背景情報のために示されたに過ぎず、特許請求する主題の範囲を判断するときの補助として使用されることを意図するのではない。

【 発明の概要 】

【 0 0 0 8 】

[0008] テナントの変更は、マルチテナント・サービスにおいて受け入れられる。テナントの変更は、テナント毎に細分化され、個々の更新ストリームが、個々のテナント・ストリーム・プロセッサによって形成される(establish)。各テナント・ストリーム・プロセッサは、所与のテナントに対する変更を、このテナントに対する変更が完了するまで行う。

【 0 0 0 9 】

[0009] この摘要は、詳細な説明において以下で更に説明する概念から選択したものを、簡略化した形態で紹介するために設けられている。この摘要は、特許請求する主題の主要な特徴や必須の特徴を特定することを意図するのではなく、特許請求する主題の範囲を判断するときの補助として使用されることを意図するのでもない。特許請求する主題は、背景において注記した欠点の内、任意のものを解決する実施態様にも全ての欠点を解決する実施態様にも限定されるのではない。

【 図面の簡単な説明 】

【 0 0 1 0 】

【図 1】図 1 は、クラウド・ベースのマルチテナント・サービス・アーキテクチャの一例のブロック図である。

【図 2】図 2 は、通知ストリーム・ジェネレーターの一例を更に詳細に示すブロック図である。

【図 3】図 3 は、図 1 に示すアーキテクチャの動作の一例を示す流れ図である。

【図 4】図 4 は、マルチテナント・クラウド・アーキテクチャの一例におけるマルチテナント機能(multi-tenancy)の異なるレベルを表すブロック図である。

【図 5】図 5 は、移動体デバイスの例を示す。

【図 6】図 6 は、移動体デバイスの例を示す。

10

20

30

40

50

【図 7】図 7 は、移動体デバイスの例を示す。

【図 8】図 8 は、計算環境の一例のブロック図である。

【発明を実施するための形態】

【 0 0 1 1 】

[0016] 図 1 は、クラウド・ベースのマルチテナント・サービス・アーキテクチャ 1 0 0 の一例のブロック図である。アーキテクチャ 1 0 0 は、複数の異なるテナント・システム 1 0 4 ~ 1 0 6 によってアクセス可能なクラウド・ベース・マルチテナント・サービス・システム 1 0 2 を含む。図 4 に関して以下で更に詳細に説明するが、クラウド・ベース・マルチテナント・サービス・システム 1 0 2 は、テナント・システム 1 0 4 ~ 1 0 6 を使用するテナントに対して、あるレベルのマルチテナント・サービスを提供する。実例として、テナントは、システム 1 0 2 によって提供されるマルチテナント・サービスが、所望のレベルにおいて、テナント毎に分けられる(separated)または分離される(isolated)ように、組織を分ける。また、図 4 に関して以下で更に詳細に説明するが、とりわけ、インフラストラクチャのレベル、アプリケーション・プラットフォームのレベル、またはアプリケーション・ソフトウェアのレベルで、マルチテナント・サービスを提供することができる。つまり、システム 1 0 2 によって提供される特定のレベルのマルチテナント機能(multi-tenancy)に応じて、特定のクライアントに対応する情報がそのレベルにおいて分けられ、他のテナントによってアクセスできないように分離される。

10

【 0 0 1 2 】

[0017] 一例として、システム 1 0 2 がサービスとしてのインフラストラクチャ(IaaS)を提供する場合、システム 1 0 2 のインフラストラクチャ・コンポーネントは、テナント 1 0 4 ~ 1 0 6 によって共有されるが、テナント情報は他の方法で別個に維持される。システム 1 0 2 がサービスとしてのプラットフォーム(PaaS)を提供する場合、プラットフォーム・コンポーネントはテナント 1 0 4 ~ 1 0 6 によって共有される。システム 1 0 2 がサービスとしてのアプリケーション・ソフトウェア(SaaS)を提供する場合、テナント 1 0 4 ~ 1 0 6 にサービスするために、システム 1 0 2 によって共通のアプリケーションが実行される。これらの実施態様のいずれにおいても、システム 1 0 2 はあるレベルのマルチテナント機能を提供するので、異なるテナント 1 0 4 ~ 1 0 6 に対応する情報は別個に維持される。これについては図 4 に関して以下で更に詳細に説明する。

20

【 0 0 1 3 】

[0018] 図 1 に示すシステム 1 0 2 の例は、通知ストリーム・ジェネレーター 1 0 8、テナント・ストリーム・プロセッサ 1 1 2 ~ 1 1 4 をその内部に含むテナント別処理システム 1 1 0、テナント・レベル別リソース 1 1 6、および図 4 に関して以下で更に詳細に論ずる種々の他のクラウド処理系コンポーネント 1 2 4 を含む。図 1 に示す例では、テナント・レベル別リソースは、テナント・データ 1 1 8 ~ 1 2 0 を含み、テナント・データ 1 1 8 ~ 1 2 0 は、それぞれ、各テナント・システム 1 0 4 ~ 1 0 6 に対応する情報を表し、この情報は、他のテナントから別個に維持されこれらによるアクセスは不可能となっている。

30

【 0 0 1 4 】

[0019] テナント・システム 1 0 4 ~ 1 0 6 は、実例として、クラウド・ベース・マルチテナント・サービス・システム 1 0 2 に対する変更に対応する。例えば、テナント・システム 1 0 4 は、システム 1 0 2 において、電子メール・サービスをオンプレミス処理系からクラウド・ベース処理系に移動させているとしてもよい。この場合、テナント・システム 1 0 4 は、その電子メール・システム全体をオンプレミス・バージョンからシステム 1 0 2 に移動させるための変更に対応する。他の例では、テナント・システム 1 0 6 が他の組織を取得したとしてもよく、したがって、多数の電子メール・アカウントをシステム 1 0 2 におけるそのクラウド・ベース処理系に追加することが必要となる可能性がある。更に他の例では、テナント・システム 1 0 4 を使用する組織が編成し直されたとしてもよい。したがって、大多数の従業員記録を変更して、彼らがもはや人事部の一部ではなく、今では販売部の一部であることを示す必要がある可能性がある。これらの例のいずれにおいて

40

50

も、または多種多様の他の例においても、テナントに合わせてクラウド・ベース処理系に変更すると、数量が比較的大きくなる可能性がある。しかしながら、システム 102 は、他のテナントにサービスする際にシステム 102 の処理能力に影響を及ぼさないような方法で、これらの変更を処理する。

#### 【0015】

[0020] テナント更新を行うときのシステム 102 の動作についての更に詳細な説明は、以下で図 2 および図 3 に関して行う。これより、理解を深めるために、端的な全体像を示す。最初に、テナント 104 ~ 106 の内 1 つがテナント変更を求める要求をシステム 102 に対して行うと仮定する。テナント変更は、要求元テナントに対応するテナント・データー 118 ~ 120 に対して要求される変更を示す。通知ストリーム・ジェネレーター 108 は、テナント別処理システム 110 と共に、テナント毎同期ストリームを作成する。一例では、テナント 104 ~ 106 によって受け取られた変更をテナント毎セグメントに組み分ける。次いで、所与のセグメントに対応するブックマークを、テナント・ストリーム・プロセッサ 112 ~ 114 の内の 1 つに受け渡す(hand)。テナント・ストリーム・プロセッサ 112 ~ 114 は、実例として、異なるサーバーであるか、少なくとも異なるサーバー・インスタンスである。このブックマークを受け取ったテナント・ストリーム・プロセッサは、次に、通知ストリーム・ジェネレーター 108 との通信リンク(同期ストリーム)を維持し、同期ストリームを通じて(over)テナント変更を入手して、必要な場合はブックマークを使用して、対応するテナント・データーに対してテナント変更を行う。同じテナント・ストリーム・プロセッサ 112 は、これらのテナント変更が通知ストリーム・ジェネレーター 108 から出尽くす(drain)まで、同じテナントに対する変更を行いつける。次いで、変更を完了したことを通知ストリーム・ジェネレーター 108 に示す。テナント・ストリーム・プロセッサが所与のテナントに対して変更を行っている間に、第 2 テナントが変更要求を通知ストリーム・ジェネレーター 108 に出した場合、ジェネレーター 108 は、システム 110 における第 2 テナント・ストリーム・プロセッサと、別の同期ストリームを形成する(establish)。第 2 ストリーム・プロセッサ 110 は、これらの変更が出し尽くされるまで、第 2 テナントに対する変更を行いつける。これは、各テナントが変更を行うために指定されたサーバーによるそれ自体の処理ストリーム(または同期ストリーム)を有するように、変更を要求する異なるテナント毎に続けられる。サーバーは、それ自体の作業負荷制約をサーバー・レベルで強制する。このように、任意の所与のテナントによって要求される変更の数には関係なく、他のテナントによって感知されるシステム 102 の処理能力は影響を受けない。

#### 【0016】

[0021] 図 2 は、通知ストリーム・ジェネレーター 108 の一例の更に詳細なブロック図である。ジェネレーター 108 は、実例として、変更細分化コンポーネント 180、保留変更ストア(pending change store) 182 を含み、更に他の単位体 184 も含むことができる。また、図 3 は、ジェネレーター 108 が、実例として、テナント 104 ~ 106 からテナント変更 186 を受け取り、これらをセグメント 188 ~ 190 に細分化することも示す。各セグメント 188 ~ 190 は異なるテナント 104 ~ 106 に対する変更を表す。また、図 3 は、一旦テナント変更 106 を細分化したなら、変更細分化コンポーネント 180 が、異なるセグメント 188 ~ 190 に対応するブックマーク 192 ~ 194 を、異なるテナント・ストリーム・プロセッサ 112 ~ 114 に送ることも示す。テナント・ストリーム・プロセッサ 112 ~ 114 は、ストア 182 から変更を入手して、次いで対応するテナントのために、対応するテナント・データー 118 ~ 120 に対して変更を行う。テナント・ストリーム・プロセッサ 112 ~ 114 は、ブックマーク 192 ~ 194 を使用して、対応するセグメント 188 ~ 190 におけるテナント変更が全て行われることを確保する。つまり、各テナント・ストリーム・プロセッサ 112 ~ 114 は、1 つのテナントに対する 1 組の変更を排他的に入手するために同期ストリームを有し、そのテナントのためだけの変更を同期ストリームによって受け取り、そのテナントに対する全ての変更が対応するセグメントから出尽くされるまで、同じテナントに対する

全ての変更を行う。次いで、ブックマークを変更細分化コンポーネント 180 に戻し、テナント・ストリーム・プロセッサが空いており異なるテナントに対する変更を行えることを示す。

【0017】

[0022] 図3は、クラウド・ベース・システム102においてテナント・データに対するテナント変更186を行うときにおける通知ストリーム・ジェネレーター108およびクラウド・ベース・マルチテナント・サービス・システム102の動作の一例を示す流れ図である。変更細分化コンポーネント180は、最初に、テナント変更186を1つ以上のテナント・システム104~106から受け取る。これは、図4においてブロック200によって示されている。次いで、コンポーネント180は変更186を異なるセグメント188~190に細分化する。セグメント188~190は、実例として、各セグメント188~190が同じテナントによって提出されたテナント変更の全てを含み、これらの変更のみを含むように、テナント毎のセグメントとなっている。保留中の変更をテナント毎に細分化することは、図4におけるブロック202によって示されている。

10

【0018】

[0023] 変更細分化コンポーネント180は、次いで、同じプロセッサが所与のテナントに対する変更の全てを処理できるように、テナント・ストリーム・プロセッサ112~114をセグメントの各々に割り当てる。ここで示す例では、各テナント・ストリーム・プロセッサ112は別のサーバー、または少なくともサーバーの別のインスタンスである。つまり、各テナントは、その要求した変更を行う専用のサーバーまたはサーバー・インスタンスを有する。

20

【0019】

[0024] 図3に示す例では、変更細分化コンポーネント180は、セグメント188~190に対応する、対応ブックマーク192~194を送ることによって、テナント・ストリーム・プロセッサ112~114を各テナントに割り当てる。ブックマークを送ることは、図4ではブロック204によって示されている。尚、2つ以上のテナントが比較的小さいテナントである場合、これらを同じテナント・ストリーム・プロセッサ112~114に接続してもよいことを注記しておく。この場合、テナント作業負荷は種々のテナント・ストリーム・プロセッサ112~114間で割り当てられるが、テナントとサーバー（またはテナント・ストリーム・プロセッサ）との間には必ずしも1対1の対応がなくともよい。テナント・ストリーム・プロセッサ間でテナントを割り当てることは、図4ではブロック206によって示されている。

30

【0020】

[0025] しかしながら、以上で説明したように、一例では、サーバー・インスタンス毎に1つのテナント・ストリームがある（これに沿って、テナント・ストリーム・プロセッサは、1つのテナントに対するテナント変更を排他的に受け取り、テナント変更を行う）。これはブロック208によって示されている。ストリームは、セグメントにおける全ての変更が行われるまで維持される。

【0021】

[0026] 行う必要があるテナント変更をテナント・ストリーム・プロセッサに通知することは、種々の異なる方法で行うことができる。例えば、これらの変更が保留変更ストア182における新たなセグメントに対応する場合、コンポーネント180は、新たなセグメントに対応する新たなブックマークを、変更を行うのを待っているテナント・ストリーム・プロセッサ（例えば、他のテナントまたはテナント・ストリームに未だ割り当てられていないもの）に送る。これは、ブロック210によって示されている。勿論、変更細分化コンポーネント180は、他の方法でもテナント・ストリーム・プロセッサに保留中の変更を通知することができ、これはブロック212によって示されている。

40

【0022】

[0027] 一旦テナント・ストリーム・プロセッサ（例えば、プロセッサ112）に、それがセグメント188~190の内の1つ（例えば、セグメント188）からの変更

50

を行わなければならないことが通知されたなら、対応するセグメントからテナント変更を引き出す。これは、図4ではブロック214によって示されている。このようにするとき、テナント・ストリーム・プロセッサ112は、実例として、セグメント118からの全ての変更を入手して行い終わるまで、通知ストリーム・ジェネレーター108との接続を維持する。これは、ハイパーテキスト・トランスファー・プロトコル(HTTP)ロング・ポーリングを使用して、または他の常時接続メカニズムを使用することによって行うことができる。接続の維持はブロック216によって示されている。テナント・ストリーム・プロセッサ112は、変更が対応するセグメントから出尽くされ、ブックマークがテナント・ストリーム・プロセッサ112によって解放されるまで、所与のテナントに対するテナント変更の全てを処理する。これは、図4ではブロック218によって示されている。テナント・ストリーム・プロセッサは、他の方法でもこれらの変更を引き出して変更を行うことができ、これはブロック220によって示されている。

#### 【0023】

[0028] テナント・ストリーム・プロセッサは、対応するセグメントから変更を引き出しつつ、所与のテナントに対して、対応するテナント・データにおいてこれらの変更を実行する。これはブロック222によって示されている。各テナント・ストリーム・プロセッサはサーバー・インスタンスに対応するので、テナント変更に対して帯域幅制約を強要する(enforce)のはサーバー・インスタンス自体である。例えば、サーバー・インスタンス自体が、CPU使用、メモリー使用、帯域幅使用等に対して制約を強要する。各テナント104~106は変更を行うためにそれ自体のサーバー・インスタンスを有するので、テナント104~106は、その対応するサーバー・インスタンス(例えば、その関連するテナント・ストリーム・プロセッサ)によって強制される使用制約を超えることができない。これは、図4ではブロック224によって示されている。このように、特定のテナント104~106が、クラウド処理系の機能を使いすぎて(overwhelm)、他のテナントに損害を与える可能性はない。テナント・ストリーム・プロセッサ(またはサーバー・インスタンス)は同時に他のテナントからの変更を処理しようとしないので(一度に1つのテナントに対する変更を排他的に処理している)、いずれの所与のテナント104~106の作業負荷も、クラウド・ベース処理系の他のテナントに提供される処理能力(performance)に影響を及ぼすことはない。代わりに、各テナントは、処理能力については、同じ扱いを受ける。何故なら、各テナントは、その変更を行うそれ自体のサーバー・インスタンス(またはテナント・ストリーム・プロセッサ)を有するからである。尚、テナント・ストリーム・プロセッサは他の方法でもテナントに対する変更を実行することに留意されたい。これはブロック226によって示されている。

#### 【0024】

[0029] また、規模調整(scaling)も容易に行うことができる。新たな機械またはサーバー・インスタンスを追加することによって、サービス・テナントに対する容量は直接そして素早く増加する。新たな機械またはサーバー・インスタンスは、変更を処理するために使用することができる、入手可能なストリーム・プロセッサに追加される。

#### 【0025】

[0030] 各テナント・ストリーム・プロセッサ112~114は、1組の変更を行った後に、対応するセグメントにおいてこのテナントに対する変更が他にも未だあるか否かを判定する。これは、図4ではブロック228によって示されている。ある場合、処理はブロック214に戻り、テナント・ストリーム・プロセッサは、対応するセグメントからこれらの変更を引き出し、テナント・データに対してこれらをコミットする。

#### 【0026】

[0031] しかしながら、ブロック228において、対応するセグメントにはこのテナントに対してそれ以上変更がないと判定した場合、テナント・ストリーム・プロセッサは、通知ストリーム・ジェネレーター108に、そのテナント変更を完了し、同じまたは異なるテナントに対する変更をコミットするために他の通知を待つことを指示する。一実施形態では、テナント・ストリーム・プロセッサ112~114は、ブックマークを解放



するまたは通知ストリーム・ジェネレーター 108 に戻すことによって、これを行うことができる。これは、図 4 の流れ図ではブロック 230 によって示されている。

【0027】

[0032] このように、通知ストリーム・ジェネレーター 108 は、テナント毎に同期ストリーム（またはテナント・ストリーム）生成できることがわかる。これは、オンボーディング（onboarding）アクティビティ（新たなユーザーをクラウド・ベース・システムに追加する等）および他のアクティビティのために、サーバーの数を増やして対応する（scale out）方法を可能にする。このシステムは、別個のテナント・ストリームを処理することができるサーバーまたはサーバー・インスタンスの数によってのみ制限されるに過ぎないので、大量のテナントの必要性に応じるために規模を調整することができる。ストリームは、異なるサーバーまたはサーバー・インスタンスによって独立して実行されるので、追加のテナント・ストリームのために、必要に応じて追加のサーバーまたはサーバー・インスタンスを呼び出すことができる。これは、異なる機械にまたがって異なるテナントを広げるためのオンボーディング・アクティビティを可能にする。

【0028】

[0033] 通知ストリーム・ジェネレーター 108 は、いつテナントが変更したかをテナント・ストリーム・プロセッサに通知するだけで、システムが動的に規模を調整することを可能にする。通常、これは、いずれの所与の時点においても、クラウド・ベース・システムにアクセスするテナントの集合全体と比較すると、小さな 1 組のテナントである。これは全体的なレイテンシを低減する。何故なら、各テナント・ストリーム・プロセッサは、変更を受け取って入手可能になるに連れて、変更を処理するからである。また、これは自動規模調整も促進する。何故なら、いずれの新たなサーバーまたはサーバー・インスタンスであっても、これらがオンラインになると直ちに、テナント変更を行うために通知ストリーム・ジェネレーター 108 がこれらを使用することが可能になるからである。これは、多数の個々のテナント毎のデータ・パイプに相当し、1 つのテナントがシステムを使いすぎて他のテナントに処理能力不足を感知させる可能性がある 1 本の太いパイプにはならない。1 つのサーバーまたはサーバー・インスタンスが、ボトルネックまたは障害点として作用することはない。加えて、通知ストリーム・ジェネレーター 108 は、テナント・ストリーム・プロセッサ 112 ~ 114 に、1 つ 1 つのオブジェクト変更を列挙せず、変更が行われる予定であることを通知する。代わりに、テナント毎に変更をセグメントに組み分ける。これも処理能力を向上させる。

【0029】

[0034] 本論述では、プロセッサおよびサーバーについて述べた。一実施形態では、プロセッサおよびサーバーは、別々には示されていないメモリーおよびタイミング回路が付随するコンピューター・プロセッサを含む。これらは、これらが属しそして有効化されるシステムまたはデバイスの機能部分であり、これらのシステムにおける他のコンポーネントまたは単位体（item）の機能を行い易くする。

【0030】

[0035] また、多数のユーザー・インターフェース・ディスプレイについても論じた。これらは、多種多様の異なる形態を取ることができ、多種多様の異なるユーザー作動可能な入力メカニズムをそこに配置することができる。例えば、ユーザー作動可能な入力メカニズムは、テキスト・ボックス、チェック・ボックス、アイコン、リンク、ドロップダウン・メニュー、検索ボックス等とすることができる。また、これらを多種多様の異なる方法で作動させることもできる。例えば、ポイントおよびクリック・デバイス（トラック・ボールまたはマウス等）を使用してこれらを作動させることができる。ハードウェア・ボタン、スイッチ、ジョイスティックまたはキーボード、サム・スイッチまたはサム・パッド等を使用してこれらを作動させることができる。更に、仮想キーボードまたは他の仮想アクチュエータを使用してこれらを作動させることもできる。加えて、これらが表示される画面がタッチ感応画面である場合、タッチ・ジェスチャを使用してこれらを作動させることができる。また、これらを表示するデバイスが音声認識コンポーネントを有する場合

、音声コマンドを使用してこれらを作動させることができる。

【 0 0 3 1 】

[0036] また、多数のデータ・ストアについても論じた。これらは各々複数のデータ・ストアに分けられることに留意されたい。全ては、これらにアクセスするシステムに対してローカルであることができ、全てがリモートであることができ、または一部がローカルであり、他がリモートであることもできる。これらの構成の全ては、本明細書において考えられる範囲内である。

【 0 0 3 2 】

[0037] また、図は、各ブロックに帰せられる機能を有する多数のブロックを示す。尚、それよりも少ないコンポーネントによってその機能が実行されるように、それよりも少ないブロックを使用できることに留意されたい。また、もっと多くのブロックを使用して、もっと多くのコンポーネントに機能を分散することもできる。

【 0 0 3 3 】

[0038] 図 4 は、システム 1 0 2 を実現することができるマルチテナント・クラウド・アーキテクチャ 1 2 4 (クラウド 1 2 4) の更に詳細な実施形態を示すブロック図である。図 4 は、他のクラウド処理系コンポーネント 1 2 2 (図 1 に示す) の一例を更に詳細に示す。

【 0 0 3 4 】

[0039] クラウド・コンピューティング(cloud computing)は、サービスを配信するシステムの物理位置や構成をエンド・ユーザーが知る必要なく、計算、ソフトウェア、データ・アクセス、およびストレージ・サービスを提供する。種々の実施形態では、クラウド・コンピューティングは、適したプロトコルを使用して、インターネットのようなワイド・エリア・ネットワークを通じてサービスを配信する。例えば、クラウド・コンピューティング・プロバイダーは、ワイド・エリア・ネットワークを通じてアプリケーションを配信し、これらにはウェブ・ブラウザまたは任意の他の計算コンポーネントによってアクセスすることができる。アーキテクチャ 1 0 0 のソフトウェアまたはコンポーネント、および対応するデータは、遠隔地にあるサーバー上に格納することができる。クラウド・コンピューティング環境における計算リソースは、リモート・データ・センターの場所に統合することができ、またはこれらを散在させることができる。クラウド・コンピューティング・インフラストラクチャは、共有データ・センターを介してサービスを配信することができるが、これらはユーザーには 1 つのアクセス・ポイントのように見える。つまり、本明細書において説明するコンポーネントおよび機能は、クラウド・コンピューティング・アーキテクチャを使用して、遠隔地にあるサービス・プロバイダから提供することができる。代わりに、これらを従来のサーバーから提供することもでき、あるいはクライアント・デバイス上に直接インストールすることも、または他の方法でインストールすることもできる。

【 0 0 3 5 】

[0040] この説明は、パブリック・クラウド・コンピューティングおよびプライベート・クラウド・コンピューティングの双方を含むことを意図している。クラウド・コンピューティング(パブリックおよびプライベート双方)は、実質的に継ぎ目のないリソースのプーリング(pooling)を提供し、更に基礎となるハードウェア・インフラストラクチャを管理および構成する必要性の低減にも備える。

【 0 0 3 6 】

[0041] パブリック・クラウドは、ベンダーによって管理され、通例、同じインフラストラクチャを使用して複数の消費者をサポートする。また、パブリック・クラウドは、プライベート・クラウドとは逆に、ハードウェアの管理からエンド・ユーザーを解放することができる。プライベート・クラウドは、組織自体によって管理することができ、インフラストラクチャは他の組織とは共有されないのが通例である。更に、組織は、設置および修理等のように、ある程度ハードウェアの保守も行う。

【 0 0 3 7 】

10

20

30

40

50

[0042] 図4は、システム102におけるコンポーネント122が、実例として、仮想化システム126、インフラストラクチャ・コンポーネント128、アプリケーション・プラットフォーム・コンポーネント130、およびアプリケーション・ソフトウェア・コンポーネント132を含むことを示す。インフラストラクチャ・コンポーネント128は、セキュリティ・コンポーネント134、ハードウェア/ソフトウェア・インフラストラクチャ136、サーバー138、負荷分散コンポーネント140、ネットワーク・コンポーネント142を含むことができ、他のコンポーネント144も含むことができる。

【0038】

[0043] アプリケーション・プラットフォーム・コンポーネント130は、実行ランタイム・コンポーネント146、オペレーティング・システム・コンポーネント148、データベース・コンポーネント150、ウェブ・サーバー・コンポーネント152を含むことができ、他のコンポーネント154も含むことができる。アプリケーション・ソフトウェア・コンポーネント132は、実例として、ユーザー・インターフェース・コンポーネント156、アプリケーション・ワークフロー158、アプリケーション・ロジック160、データベース・システム162を含み、他の単位体164も含むことができる。

【0039】

[0044] アーキテクチャ124によって実現されるマルチテナント機能のレベルに応じて、仮想化システム126は物理計算デバイス・コンポーネント122を1つ以上の仮想デバイスに電子的に分ける。これらのデバイスの各々は、計算タスクを実行するために使用し管理することができる。

【0040】

[0045] マルチテナント・クラウド・コンピューティング・アーキテクチャ124は、多数の異なるモデルにしたがって、多数の異なるレベルでサービスを提供することができる。例えば、これらは、とりわけ、サービスとしてのインフラストラクチャ(IaaS)、サービスとしてのプラットフォーム(PaaS)、およびサービスとしてのソフトウェア(SaaS)を含むことができる。IaaSは最も基本的であり、それよりも高いレベルのモデル(それぞれPaaSおよびSaaS)の各々が、それよりも低いモデルの詳細を抜き取る(abstract from)。

【0041】

[0046] IaaSでは、物理または仮想機械が、他のリソースと共に提供される。監視コンポーネント(supervisory component)(ハイパーバイザーと呼ばれることもある)が仮想機械を実行する。比較的多数の仮想機械を実行し、種々のテナントの必要性に応じて規模を拡大および縮小するために、複数の異なるハイパーバイザーを使用することができる。また、IaaSモデルは、要求に応じて追加のリソース(他のインフラストラクチャ・コンポーネント128等)も提供することができる。テナントがそのアプリケーションを展開するために、これらはオペレーティング・システム・イメージおよびそのアプリケーション・ソフトウェアをクラウド・インフラストラクチャ・コンポーネント128上にインストールする。そして、テナントは、オペレーティング・システムおよびアプリケーション・ソフトウェアを維持することを責務とする。

【0042】

[0047] PaaSは、アプリケーション・プラットフォーム・コンポーネント130をサービスとして提供するクラウド・アーキテクチャ124を含む(involve)。アプリケーション開発者は、基礎のハードウェアおよびソフトウェア・レイヤを管理する必要なく、クラウド・プラットフォーム・コンポーネント130上で彼らのソフトウェアを開発し実行することができる。

【0043】

[0048] SaaSは、アプリケーション・コンポーネント132におけるアプリケーション・ソフトウェアおよびデータベースへのアクセスを与えるアーキテクチャ124を含む。クラウド・アーキテクチャ124は、インフラストラクチャ・コンポーネント128、およびアプリケーションを実行するプラットフォーム・コンポーネント130を管理

する。また、クラウド 1 2 4 はアプリケーション・ソフトウェアをアプリケーション・コンポーネント 1 3 2 にインストールして動作させ、テナントはこのソフトウェアにアクセスするが、クラウド・インフラストラクチャ・コンポーネント 1 2 8 も、アプリケーションが実行するプラットフォーム・コンポーネント 1 3 0 も管理しない。このような実施態様では、仮想化システム 1 2 6 が、変化する作業負荷に応ずるために、ランタイムにおいて複数の仮想機械を提供する。負荷分散装置(load balancer)はこれらの仮想機械にわたって作業を分散する。このプロセスは、多くの場合、アプリケーションへの 1 つのアクセス・ポイントだけを見るテナントに対して透過的である。

【 0 0 4 4 】

[0049] マルチテナント環境では、いずれの機械も、テナント・システム 1 0 4 ~ 1 0 6 を展開する 1 つよりも多いユーザー組織にサービス供与する(serve)ことができる。しかしながら、マルチテナント機能は、クラウド・アーキテクチャの 3 つのレイヤ ( I a a S、P a a S、および S a a S ) の全てに適用することができる。正確なマルチテナント機能の度合いは、コア・アプリケーション (またはアプリケーション・コンポーネント 1 3 2 ) の内どの位がテナント 1 0 4 ~ 1 0 6 にわたって共有されるように設計されているかに基づくことができる。マルチテナント機能の度合いが比較的高いと、データベース方式(database schema)を共有することが可能であり、業務ロジック(business logic)、ワークフロー、およびユーザー・インターフェース・レイヤーのカスタム化をサポートする。比較的低い度合いのマルチテナント機能では、I a a S および P a a S コンポーネント 1 2 8 および 1 3 0 が、それぞれ、各テナントに専用である専用仮想化コンポーネントを有するアプリケーション・コンポーネント 1 3 2 によって共有される。

【 0 0 4 5 】

[0050] 以上の論述は、サービスとしてのソフトウェアを提供するクラウド・ベース・マルチテナント・サービス・システム 1 0 2 に関して進めてきた。したがって、仮想化システム 1 2 6 は、各テナント 1 0 4 ~ 1 0 6 に、それら自体の安全で別々の仮想計算環境をアプリケーション・ソフトウェア・レベル上で提供するために、別々の仮想機械を設ける。つまり、各テナント 1 0 4 ~ 1 0 6 は、それら自体のアプリケーション (それら自体の電子メール・アプリケーション、文書管理アプリケーション、業務システム等のようなアプリケーション) に対する変更を行うことができる。図 1 ~ 図 3 に関して以上で説明したように行われるとき、これらは、他のテナントによって観察される(observe)処理能力に影響を及ぼすことなく、そのようにすることができる。

【 0 0 4 6 】

[0051] また、システム 1 0 2 のあるエレメントをクラウド内に配置することができるが、他のエレメントは配置されないことも考えられる。一例として、データベースおよびデータ・ストアをクラウド 1 2 4 の外側に配置し、クラウド 1 2 4 を介してアクセスすることができる。他の例では、他のコンポーネントがクラウド 1 2 4 の外側にあることも可能である。これらがどこに位置するかには関係なく、テナント・システム 1 0 4 ~ 1 0 6 におけるデバイスによって、ネットワーク 3 0 0 (ワイド・エリア・ネットワークまたはローカル・エリア・ネットワーク)を通じてこれらに直接アクセスすることができ、遠隔地(remote site)におけるサービスによってこれらをホストすることができ、あるいはクラウドを介してサービスとして提供し、クラウド内に存在する接続サービスによってアクセスすることができる。これらのアーキテクチャは全て、本明細書において考えられる範囲内である。

【 0 0 4 7 】

[0052] また、アーキテクチャ 1 0 0 またはその部分を多種多様の異なるデバイス上に配置できることにも留意されたい。これらのデバイスの一部には、サーバー、デスクトップ・コンピューター、ラップトップ・コンピューター、タブレット・コンピューター、またはパーム・トップ・コンピューター、セル・フォン、スマート・フォン、マルチメディア・プレーヤー、パーソナル・デジタル・アシスタント等のような他の移動体デバイスが含まれる。

## 【 0 0 4 8 】

[0053] 図 5 は、本システム（またはその部分）を展開することができるユーザーまたはクライアントのハンドヘルド・デバイス 1 6 として使用することができる、ハンドヘルドまたは移動体計算デバイスの一例の簡略ブロック図である。図 6 および図 7 はハンドヘルドまたは移動体デバイスの例である。

## 【 0 0 4 9 】

[0054] 図 5 は、システム 1 0 0 のコンポーネントまたはテナント 1 0 4 ~ 1 0 6 を実行することができる、あるいはアーキテクチャ 1 0 0 とインターフェースする、あるいは双方が可能なクライアント・デバイス 1 6 のコンポーネントの全体的なブロック図である。デバイス 1 6 において、ハンドヘルド・デバイスが他の計算デバイスと通信することを可能にし、ある実施形態ではスキミングによってというように、情報を自動的に受信するためのチャンネルを提供する通信リンク 1 3 が設けられている。通信リンク 1 3 の例には、赤外線ポート、シリアル / U S B ポート、イーサネット（登録商標）・ポートのようなケーブル・ネットワーク・ポート、および 1 つ以上の通信プロトコルによる通信を可能にするワイヤレス・ネットワーク・ポートが含まれる。通信プロトコルには、汎用パケット無線サービス（G P R S）、L T E、H S P A、H S P A + ならびに 3 G および 4 G 無線プロトコル、I X r t t、更にはショート・メッセージ・サービスが含まれる。これらは、ネットワークへのセルラ・アクセスを与えるために使用されるワイヤレス・サービス、ならびにネットワークへのローカル・ワイヤレス接続を提供する W i - F i プロトコル、および B l u e t o o t h プロトコルである。

## 【 0 0 5 0 】

[0055] 他の実施形態の下では、S D カード・インターフェース 1 5 に接続されたリムーバブル・セキュア・デジタル（S D）カードにおいてアプリケーションまたはシステムが受け取られる。S D カード・インターフェース 1 5 および通信リンク 1 3 は、バス 1 9 に沿ってプロセッサ 1 7（他の図において先に説明したプロセッサまたはサーバーも具体化することができる）と通信し、バス 1 9 は、メモリー 2 1 および入力 / 出力（I / O）コンポーネント 2 3、ならびにクロック 2 5 およびロケーション・システム 2 7 にも接続されている。

## 【 0 0 5 1 】

[0056] I / O コンポーネント 2 3 は、一実施形態では、入力および出力動作を容易にするために設けられる。デバイス 1 6 の種々の実施形態のための I / O コンポーネント 2 3 は、ボタン、タッチ・センサー、多重タッチ・センサー、光またはビデオ・センサ、音声センサ、タッチ・スクリーン、近接センサ、マイクロフォン、傾きセンサ、および重力スイッチのような入力コンポーネント、ならびにディスプレイ・デバイス、スピーカー、および / またはプリンター・ポートのような出力コンポーネントを含むことができる。他の I / O コンポーネント 2 3 も同様に使用することができる。

## 【 0 0 5 2 】

[0057] クロック 2 5 は、実例として、時刻および日にちを出力するリアル・タイム・クロック・コンポーネントを含む。また、実例として、プロセッサ 1 7 にタイミング機能を与えることができる。

## 【 0 0 5 3 】

[0058] ロケーション・システム 2 7 は、実例として、デバイス 1 6 の現在の地理的位置を出力するコンポーネントを含む。これは、例えば、全地球測位システム（G S P）受信機、L O R A N システム、推測航法システム、セルラ三角測量システム、または他の測位システムを含むことができる。また、これは、例えば、所望の地図、ナビゲーション経路、および他の地理的機能を生成する地図作成ソフトウェアまたはナビゲーション・ソフトウェアも含むことができる。

## 【 0 0 5 4 】

[0059] メモリー 2 1 は、オペレーティング・システム 2 9、ネットワーク設定 3 1、アプリケーション 3 3、アプリケーション・コンフィギュレーション設定 3 5、データー

10

20

30

40

50

・ストア 37、通信ドライバー 39、および通信コンフィギュレーション設定 41 を格納する。メモリ 21 は、全てのタイプの有形揮発性および不揮発性コンピューター読み取り可能メモリ・デバイスを含むことができる。また、これはコンピューター記憶媒体（以下で説明する）も含むことができる。メモリ 21 は、コンピューター読み取り可能命令を格納し、プロセッサ 17 によってこのコンピューター読み取り可能命令が実行されると、このプロセッサに、命令にしたがってコンピューター実装ステップまたは機能を実行させる。同様に、デバイス 16 は、クライアント業務システム 24 を有することができる。クライアント業務システム 24 は種々の業務用アプリケーションを実行すること、あるいはテナント・システム 104 ~ 106 の一部または全部を具体化することができる。プロセッサ 17 は、他のコンポーネントによって、それらの機能も同様にやり易くするために、有効化することができる。

10

#### 【0055】

[0060] ネットワーク設定 31 の例には、プロキシー情報、インターネット接続情報、およびマッピングのようなものが含まれる。アプリケーション・コンフィギュレーション設定 35 は、特定の企業またはユーザーに合わせてアプリケーションを改造する (tailor) 設定を含む。通信コンフィギュレーション設定 41 は、他のコンピューターと通信するためのパラメーターを提供し、GPRS パラメーター、SMS パラメーター、接続ユーザー名およびパスワードのような項目を含む。

#### 【0056】

[0061] アプリケーション 33 は、デバイス 16 上に既に格納されているアプリケーション、または使用中にインストールされたアプリケーションであることが可能であるが、これらはオペレーティング・システム 29 の一部であること、またはデバイス 16 の外部でホストされることも同様に可能である。

20

#### 【0057】

[0062] 図 6 は、デバイス 16 がタブレット・コンピューター 600 である一例を示す。図 6 において、コンピューター 600 は、ユーザー・インターフェース表示画面 602 と共に示されている。画面 602 は、タッチ・スクリーン（したがって、ユーザーの指からのタッチ・ジェスチャを使用してアプリケーションと対話処理することができる）、あるいはペンまたはスタイラスからの入力を受け取るペン対応インターフェースとすることができる。また、これは画面上の仮想キーボードを使用することもできる。勿論、例えば、ワイヤレス・リンクまたは USB ポートのような適した取り付けメカニズムを介して、これをキーボードまたは他のユーザー入力デバイスに取り付けてもよい。また、実例として、コンピューター 600 は音声入力を受け取ることもできる。

30

#### 【0058】

[0063] デバイス 16 の更に他の例も同様に使用することができる。これらは、フィーチャー・フォン、スマート・フォン、または移動体電話機を含むことができる。この電話機は、電話番号を入力する (dialing) ための 1 組のキーパッド、アプリケーション画像、アイコン、ウェブ・ページ、写真、およびビデオを含む画像を表示することができるディスプレイ、ならびにディスプレイ上に示される項目を選択するための制御ボタンを含むことができる。この電話機は、汎用パケット無線サービス (GPRS) および I X r t t、ならびにショート・メッセージ・サービス (SMS) 信号のような、セルラ・フォン信号を受信するためのアンテナを含むことができる。ある実施形態では、電話機は SD カードを受け入れるセキュア・デジタル (SD) カード・スロットを含むこともできる。

40

#### 【0059】

[0064] また、移動体デバイスは、パーソナル・デジタル・アシスタント (PDA)、またはマルチメディア・プレーヤー、またはタブレット計算デバイス等（以後 PDA と呼ぶ）であることも可能である。PDA は、誘導性画面を含むことができる。誘導性画面は、スタイラスがこの画面の上方に位置付けられると、スタイラス（またはユーザーの指のような他のポインター）の位置を検知する。これによって、ユーザーは画面上で項目を選択する、強調する、移動させることができ、更に描画および記入することができる。ま

50

た、PDAは多数のユーザー入力キーまたはボタンも含むことができ、これらは、ディスプレイに接触することなく、ディスプレイ上に表示されるメニュー選択肢または他の表示選択肢全域をユーザーがスクロールすること、更にユーザーがアプリケーションを変更することまたはユーザー入力機能を選択することを可能にする。PDAは、他のコンピューターとのワイヤレス通信を可能にする内部アンテナおよび赤外線送信機/受信機、ならびに他の計算デバイスへのハードウェア接続を可能にする接続ポートを含むことができる。このようなハードウェア接続は、通例、シリアルまたはUSBポートを介して他のコンピューターに接続するクレードルを介して行われる。したがって、これらの接続はネットワーク外(non-network)接続である。

【0060】

10

【0065】 図7は、スマート・フォン71を示す。スマート・フォン71は、アイコンまたはタイルあるいは他のユーザー入力メカニズム75を表示するタッチ感応ディスプレイ73を有する。メカニズム75は、アプリケーションを実行する、通話する、データ転送動作を実行する等のためにユーザーによって使用することができる。一般に、スマート・フォン71は、移動体オペレーティング・システム上に構築され、フィーチャー・フォンよりも進んだ計算能力および接続機能(connectivity)を提供する。

【0061】

【0066】 尚、デバイス16の他の形態も可能であることを注記しておく。

【0067】 図8は、アーキテクチャ100、またはその一部(例えば)を展開することができる計算環境の一例である。図8を参照すると、いくつかの実施形態を実現するためのシステム例は、コンピューター810の形態とした汎用計算デバイスを含む。コンピューター810のコンポーネントは、処理ユニット820(以上で論じたプロセッサまたはサーバーを構成することができる)、システム・メモリー830、およびシステム・バス821を含むことができるが、これらに限定されるのではない。システム・バス821は、システム・メモリーから処理ユニット820までを含む種々のシステム・コンポーネントを結合する。システム・バス821は、メモリー・バスまたはメモリー・コントローラ、周辺バス、および種々のバス・アーキテクチャの内任意のものを使用するローカル・バスを含む、様々なタイプのバス構造の内任意のものとすることができる。一例として、そして限定ではなく、このようなアーキテクチャは、業界標準アーキテクチャ(ISA)バス、マイクロ・チャンネル・アーキテクチャ(MCA)バス、拡張ISA(EISA)バス、ビデオ電子規格連合(VESA)ローカル・バス、およびMezzanineバスとしても知られる周辺コンポーネント相互接続(PCI)バスを含む。図1に関して説明したメモリーおよびプログラムは、図8の対応する部分に展開することができる。

20

30

【0062】

【0068】 コンピューター810は、通例、種々のコンピューター読み取り可能媒体を含む。コンピューター読み取り可能媒体は、コンピューター810によってアクセスすることができる任意の入手可能な媒体とすることができ、揮発性および不揮発性双方の媒体、ならびにリムーバブルおよび非リムーバブル媒体を含む。一例として、そして限定ではなく、コンピューター読み取り可能媒体は、コンピューター記憶媒体および通信媒体を含むことができる。コンピューター記憶媒体は、変調データ信号または搬送波とは異なり、これを含まない。これは、揮発性および不揮発性双方の、リムーバブルおよび非リムーバブル媒体を含むハードウェア記憶媒体を含み、コンピューター読み取り可能命令、データ構造、プログラム・モジュール、または他のデータというような情報の格納のための任意の方法または技術で実現される。コンピューター記憶媒体は、RAM、ROM、EEPROM、フラッシュ・メモリーまたは他のメモリー技術、CD-ROM、デジタル・バーサタイル・ディスク(DVD)または他の光ディスク・ストレージ、磁気カセット、磁気テープ、磁気ディスク・ストレージまたは他の磁気記憶デバイス、または所望の情報を格納するために使用することができそしてコンピューター810によってアクセスすることができる他のあらゆる媒体を含むが、これらに限定されるのではない。通信媒体は、通例、コンピューター読み取り可能命令、データ構造、プログラム・モジュール、また

40

50

はその他のデーターを移送メカニズム内に具体化し、任意の情報配信媒体を含む。「変調データー信号」という用語は、その信号内に情報をエンコードするような方法で、その特性の1つ以上が設定または変更された信号を意味する。一例として、そして限定ではなく、通信媒体は、有線ネットワークまたは直接有線接続というような有線媒体と、音響、RF、赤外線、および他のワイヤレス媒体というようなワイヤレス媒体とを含む。以上の内任意のものの組み合わせも、コンピューター読み取り可能媒体の範囲に含まれてしかるべきである。

【0063】

[0069] システム・メモリー830は、リード・オンリー・メモリー(ROM)831およびランダム・アクセス・メモリー(RAM)832のような、揮発性および/または不揮発性メモリーの形態としたコンピューター記憶媒体を含む。基本入力/出力システム833(BIOS)は、起動中におけるように、コンピューター810内部にあるエレメント間で情報を転送するのに役立つ基本的なルーチンを含み、通例ROM831に格納される。RAM832は、通例、処理ユニット820によって直ちにアクセス可能なデーターおよび/またはプログラム・モジュール、および/または現在処理ユニット820によって処理されているデーターおよび/またはプログラム・モジュールを含む。一例として、そして限定ではなく、図8は、オペレーティング・システム834、アプリケーション・プログラム835、他のプログラム・モジュール836およびプログラム・データー837を示す。

【0064】

[0070] また、コンピューター810は、他のリムーバブル/非リムーバブル、揮発性/不揮発性コンピューター記憶媒体も含むことができる。一例としてに過ぎないが、図8は、非リムーバブル、不揮発性磁気媒体に対して読み取りまたは書き込みを行うハード・ディスク・ドライブ841、およびCD-ROMまたは他の光媒体のようなリムーバブル、不揮発性光ディスク856に対して読み取りまたは書き込みを行う光ディスク・ドライブ855を示す。この動作環境例において使用することができる他のリムーバブル/非リムーバブル、揮発性/不揮発性コンピューター記憶媒体には、磁気テープ・カセット、フラッシュ・メモリー・カード、デジタル・バーサタイル・ディスク、デジタル・ビデオ・テープ、ソリッド・ステートRAM、ソリッド・ステートROM等が含まれるが、これらに限定されるのではない。ハード・ディスク・ドライブ841は、通例、インターフェース840のような非リムーバブル・メモリー・インターフェースを介してシステム・バス821に接続され、光ディスク・ドライブ855は、通例、インターフェース850のようなリムーバブル・メモリー・インターフェースによってシステム・バス821に接続される。

【0065】

[0071] 代わりに、または加えて、本明細書において説明した機能は、少なくとも部分的に、1つ以上のハードウェア・ロジック・コンポーネントによって実行することもできる。例えば、そして限定ではなく、使用することができる例示的なタイプのハードウェア・ロジック・コンポーネントには、フィールド・プログラマブル・ゲート・アレイ(FPGA)、特定プログラム集積回路(ASIC)、特定プログラム標準製品(ASSP)、システム・オン・チップ・システム(SOC)、複合プログラマブル・ロジック・デバイス(CPLD)等を含むことができる。

【0066】

[0072] 以上で論じ図8に示すこれらのドライブおよびそれに関連するコンピューター記憶媒体は、コンピューター810のためのコンピューター読み取り可能命令、データー構造、プログラム・モジュール、および他のデーターの格納を行う。図8では、例えば、ハード・ディスク・ドライブ841は、オペレーティング・システム844、アプリケーション・プログラム845、他のプログラム・モジュール846、およびプログラム・データー847を格納することが示される。尚、これらのコンポーネントは、オペレーティング・システム834、アプリケーション・プログラム835、他のプログラム・モジュ



ール 8 3 6、およびプログラム・データー 8 3 7 と同一であること、または異なることもできる。オペレーティング・システム 8 4 4、アプリケーション・プログラム 8 4 5、他のプログラム・モジュール 8 4 6、およびプログラム・データー 8 4 7 は、ここでは、少なくともこれらが異なるコピーであることを示すために、異なる番号が与えられている。  
【 0 0 6 7 】

[0073] ユーザーは、キーボード 8 6 2、マイクロフォン 8 6 3、およびマウス、トラックボール、またはタッチ・パッドのようなポインティング・デバイス 8 6 1 というような入力デバイスによって、コマンドおよび情報をコンピューター 8 1 0 に入力することができる。他の入力デバイス（図示せず）には、ジョイスティック、ゲーム・パッド、衛星ディッシュ、スキャナー等を含むことができる。これらおよび他の入力デバイスは、多くの場合、ユーザー入力インターフェース 8 6 0 を介して処理ユニット 8 2 0 に接続される。ユーザー入力インターフェース 8 6 0 は、システム・バスに結合されるが、パラレル・ポート、ゲーム・ポート、またはユニバーサル・シリアル・バス（USB）のような他のインターフェースおよびバス構造によって接続されてもよい。視覚ディスプレイ 8 9 1 または他のタイプのディスプレイ・デバイスも、ビデオ・インターフェース 8 9 0 のようなインターフェースを介して、システム・バス 8 2 1 に接続される。モニターに加えて、コンピューターは、スピーカー 8 9 7 およびプリンター 8 9 6 のような他の周辺出力デバイスも含むことができ、これらは出力周辺インターフェース 8 9 5 を介して接続することができる。

【 0 0 6 8 】

[0074] コンピューター 8 1 0 は、リモート・コンピューター 8 8 0 のような 1 つ以上のリモート・コンピューターへの論理接続を使用して、ネットワーク接続環境（networked environment）において動作することもできる。リモート・コンピューター 8 8 0 は、パーソナル・コンピューター、ハンドヘルド・デバイス、サーバー、ルーター、ネットワーク PC、ピア・デバイス、または他の一般的なネットワーク・ノードであってよく、通例、コンピューター 8 1 0 に関して先に説明したエレメントの多くまたは全部を含む。図 8 に示す論理接続は、ローカル・エリア・ネットワーク（LAN）8 7 1 およびワイド・エリア・ネットワーク（WAN）8 7 3 を含むが、他のネットワークを含むこともできる。このようなネットワーキング環境は、事務所、企業規模のコンピューター・ネットワーク、イントラネット、およびインターネットでは極普通である。

【 0 0 6 9 】

[0075] LAN ネットワーキング環境において使用される場合、コンピューター 8 1 0 は、ネットワーク・インターフェースまたはアダプター 8 7 0 を介して LAN 8 7 1 に接続することができる。WAN ネットワーキング環境において使用される場合、コンピューター 8 1 0 は、通例、インターネットのような WAN 8 7 3 を介して通信を行うモデム 8 7 2 または他の手段を含む。モデム 8 7 2 は、内蔵型でも外付けでもよく、ユーザー入力インターフェース 8 6 0 または他のしかるべきメカニズムを介してシステム・バス 8 2 1 に接続することができる。ネットワーク接続環境では、コンピューター 8 1 0 に関して図示したプログラム・モジュールまたはその一部が、リモート・メモリー記憶デバイスに格納されてもよい。一例として、そして限定ではなく、図 8 は、リモート・アプリケーション・プログラム 8 8 5 を、リモート・コンピューター 8 8 0 上に存在するものとして示す。尚、図示したネットワーク接続は一例であり、コンピューター間に通信リンクを張る他の手段を使用してもよいことは認められよう。

【 0 0 7 0 】

[0076] また、本明細書において説明した異なる実施形態を異なる方法で組み合わせることができることも注記してしかるべきである。即ち、1 つ以上の実施形態の一部を 1 つ以上の他の実施形態の一部と組み合わせることができる。このことの全て(all of this)は、本明細書において考えられる範囲内である。

【 0 0 7 1 】

[0077] 例 1 は、コンピューター・システムであって、以下を含む。

[0078] マルチテナント・システムにおける複数のサーバー・インスタンスであって、サーバー・インスタンスが、複数の異なるテナントにサービスし、各テナントが異なる組織に対応し対応するテナント・データーを有し、マルチテナント・システムにおけるサーバー・インスタンスが各テナントに対応するテナント・データーを他のテナントから分離する。

【 0 0 7 2 】

[0079] 複数の異なる要求元テナントの各々から 1 組の変更要求を受ける通知ストリーム・ジェネレーターであって、各組の変更要求が、要求元テナントに対応するテナント・データーに対する変更を示し、通知ストリーム・ジェネレーターが、要求元テナントから受けた各組の変更要求について異なるサーバー・インスタンスに通知し、通知を受けたサーバー・インスタンスが、通知を受けた 1 組の変更要求における全ての変更を行い終えるまで、各通知を受けたサーバー・インスタンスが通知ストリーム・ジェネレーターとの接続を維持し、その後この接続を解放する。

10

【 0 0 7 3 】

[0080] 例 2 は、任意のまたは全ての以前の例のコンピューター・システムであって、マルチテナント・システムがクラウド・ベース・システムである。

[0081] 例 3 は、任意のまたは全ての以前の例のコンピューター・システムであって、通知ストリーム・ジェネレーターが以下を含む。

【 0 0 7 4 】

[0082] 複数組の変更要求をテナント毎のセグメントに細分化する変更細分化コンポーネント。

20

[0083] 例 4 は、任意のまたは全ての以前の例のコンピューター・システムであって、通知ストリーム・ジェネレーターが以下を含む。

【 0 0 7 5 】

[0084] 保留変更ストアであって、変更細分化コンポーネントが、複数組の変更要求を、保留変更ストアにおけるテナント毎のセグメントに細分化する。

[0085] 例 5 は、任意のまたは全ての以前の例のコンピューター・システムであって、通知を受けたサーバー・インスタンスの各々が、通知された 1 組の変更要求に対応する所与のテナント毎セグメントから全ての変更が引き出され終えるまで、通知ストリーム・ジェネレーターとの接続を維持し、所与のテナント毎のセグメントから変更を引き出す。

30

【 0 0 7 6 】

[0086] 例 6 は、任意のまたは全ての以前の例のコンピューター・システムであって、通知ストリーム・ジェネレーターが、保留変更ストアにおけるテナント毎のセグメントに対応するブックマークを各所与のサーバー・インスタンスに与えることによって、所与のサーバー・インスタンスに通知する。

【 0 0 7 7 】

[0087] 例 7 は、任意のまたは全ての以前の例のコンピューター・システムであって、所与のサーバー・インスタンスが、変更要求を引き出してこの変更要求行うときに、個々のサーバー・インスタンス毎に、リソース使用制約を強要する。

【 0 0 7 8 】

40

[0088] 例 8 は、任意のまたは全ての以前の例のコンピューター・システムであって、追加の要求元テナントから受けた変更要求の追加の 1 組毎に、通知ストリーム・ジェネレーターが、追加の 1 組の変更要求を処理することを、追加のサーバー・インスタンスに通知する。

【 0 0 7 9 】

[0089] 例 9 は、任意のまたは全ての以前の例のコンピューター・システムであって、サーバー・インスタンスの各々が以下を含む。

[0090] 別個の物理サーバー上に実装されたサーバー・インスタンス。

【 0 0 8 0 】

[0091] 例 1 0 は、以下を含むコンピューター・システムである。

50

[0092] 複数の異なるテナントにサービス供与し、異なるテナントの各々毎に別個の分離されたテナント・データーを維持するマルチテナント・システムであって、このマルチテナント・システムが以下を含む。

【 0 0 8 1 】

[0093] テナント・データーにおいてテナント変更を処理する複数の異なるサーバー。

[0094] 対応する要求元テナントからの複数組のテナント変更を受け取り、対応する 1 組のテナント変更を処理するために、複数の異なるサーバーから 1 つのサーバーを各要求元テナントに割り当てる通知ストリーム・ジェネレーターであって、サーバーが、通知ストリーム・ジェネレーターが他のテナントに対するテナント変更を処理するためにサーバーを割り当てる前に、当該サーバーが割り当てられた対応するテナントに対する 1 組のテナント変更における全てのテナント変更を処理する。

10

【 0 0 8 2 】

[0095] 例 1 1 は、任意のまたは全ての以前の例のコンピューター・システムであって、複数の異なるサーバーが、複数の異なるサーバー・インスタンスを含む。

[0096] 例 1 2 は、任意のまたは全ての以前の例のコンピューター・システムであって、通知ストリーム・ジェネレーターが以下を含む。

【 0 0 8 3 】

[0097] 複数組の変更要求をセグメントに細分化する変更細分化コンポーネントであって、各セグメントが 1 つの要求元テナントに対応する。

[0098] 例 1 3 は、任意のまたは全ての以前の例のコンピューター・システムであって、要求元テナントに割り当てられたサーバーが、要求元テナントに対応するセグメントからテナント変更を引き出し、要求元テナントに対応するテナント・データーにおいてテナント変更を行い、セグメントからのテナント変更の全てが、要求元テナントに対応するテナント・データーに対して行われたときに、通知ストリーム・ジェネレーターに通知する。

20

【 0 0 8 4 】

[0099] 例 1 4 は、任意のまたは全ての以前の例のコンピューター・システムであって、更に以下を含む。

[00100] 複数組のテナント変更を格納する保留変更ストア。

【 0 0 8 5 】

[00101] 例 1 5 は、任意のまたは全ての以前の例のコンピューター・システムにおいて、変更細分化コンポーネントが、複数組のテナント変更を保留変更ストアにおいて細分化する。

30

【 0 0 8 6 】

[00102] 例 1 6 は、任意のまたは全ての以前の例のコンピューター・システムにおいて、マルチテナント・システムがクラウド・コンピューティング・アーキテクチャにおいて実現される。

【 0 0 8 7 】

[00103] 例 1 7 は、方法であって、以下のステップを含む。

[00104] 第 1 テナントから、マルチテナント・システムにおける第 1 テナントに対応する第 1 テナント・データーに対して要求された変更を示す第 1 組のテナント変更を受け取るステップ。

40

【 0 0 8 8 】

[00105] 第 2 テナントから、マルチテナント・システムにおける第 2 テナントに対応する第 2 テナント・データーに対して要求された変更を示す第 2 組のテナント変更を受け取るステップ。

【 0 0 8 9 】

[00106] 第 1 組のテナント変更のマルチテナント・システムにおける第 1 サーバーへの伝達のために第 1 排他的同期ストリームを形成するステップであって、第 1 サーバーが第 1 組のテナント変更における要求された変更の全てを行うまで、排他的同期ストリーム

50

が、第 1 組のテナント変更以外の他の組のテナント変更を伝達する同期ストリームを除外する。

【 0 0 9 0 】

[00107] 第 2 組のテナント変更のマルチテナント・システムにおける第 2 サーバーへの伝達のために第 2 排他的同期ストリームを形成するステップであって、第 2 サーバーが第 2 組のテナント変更における要求された変更の全てを行うまで、排他的同期ストリームが、第 2 組のテナント変更以外の、他の組のテナント変更を伝達する同期ストリームを除外する。

【 0 0 9 1 】

[00108] 例 1 8 は、任意のまたは全ての以前の例の方法であって、更に以下のステップを含む。

10

[00109] 通知ストリーム・ジェネレーターによって、第 1 セグメントにおける第 1 組のテナント変更を保留変更データ・ストアに格納するステップであって、第 1 セグメントが第 1 テナントのみに対するテナント変更を含む。

【 0 0 9 2 】

[00110] 通知ストリーム・ジェネレーターによって、第 2 セグメントにおける第 2 組のテナント変更を保留変更データ・ストアに格納するステップであって、第 2 セグメントが第 2 テナントのみに対するテナント変更を含む。

【 0 0 9 3 】

[00111] 例 1 9 は、任意のまたは全ての以前の例の方法において、第 1 排他的同期ストリームを形成するステップが以下のステップを含む。

20

[00112] 第 1 組のテナント変更によって示される、第 1 テナントから受けた変更要求(requested changes)の全てを第 1 サーバーが行うまで、第 1 サーバーと通知ストリーム・ジェネレーターとの間に第 1 接続を維持し、その後、他のテナントに対する追加組のテナント変更を第 1 サーバーに通知できるように、第 1 サーバーを第 1 接続から解放するステップ。

【 0 0 9 4 】

[00113] 例 2 0 は、任意のまたは全ての以前の例の方法において、第 2 排他的同期ストリームを形成するステップが以下のステップを含む。

[00114] 第 2 組のテナント変更によって示される、第 2 テナントから受けた変更要求(requested changes)の全てを第 2 サーバーが行うまで、第 2 サーバーと通知ストリーム・ジェネレーターとの間に第 2 接続を維持し、その後、他のテナントに対する追加組のテナント変更を第 2 サーバーに通知できるように、第 2 サーバーを第 2 接続から解放するステップ。

30

【 0 0 9 5 】

[00115] 以上、構造的特徴および/または方法論的アクトに特定の文言で本主題について説明したが、添付する特許請求の範囲において定められる主題は、必ずしも以上で説明した具体的な特徴やアクトには限定されないことは理解されよう。逆に、以上で説明した具体的な特徴およびアクトは、特許請求の範囲を実現する形態例として開示されたままである。

40

【図 1】

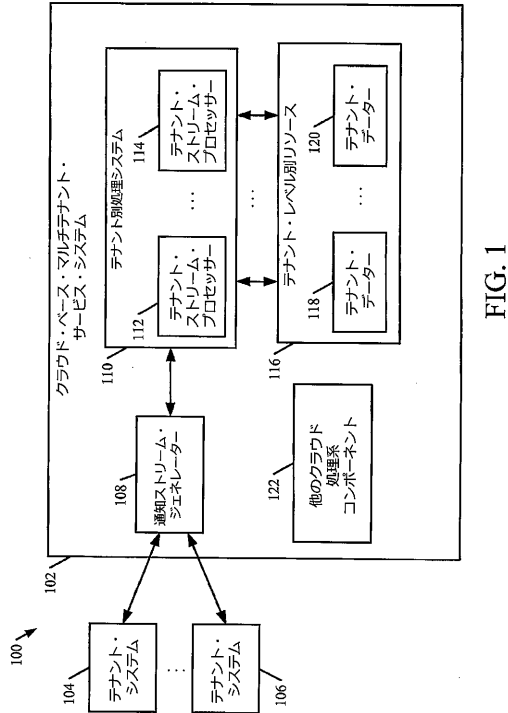


FIG. 1

【図 2】

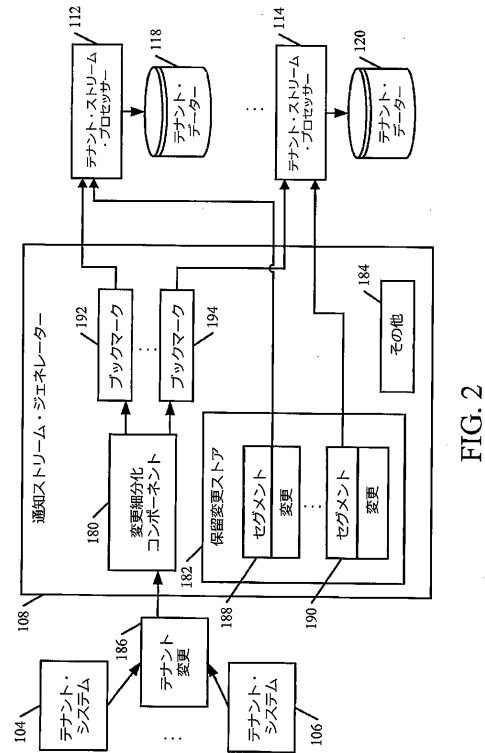


FIG. 2

【図 3】

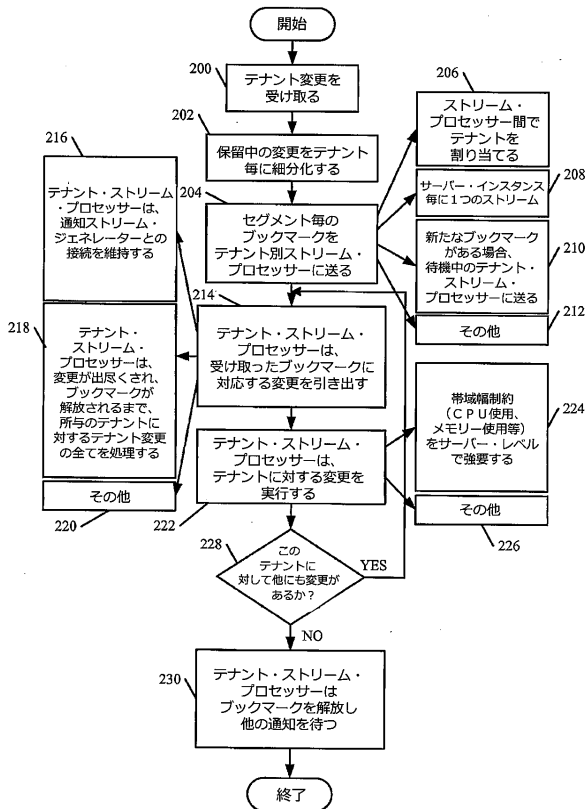


FIG. 3

【図 4】

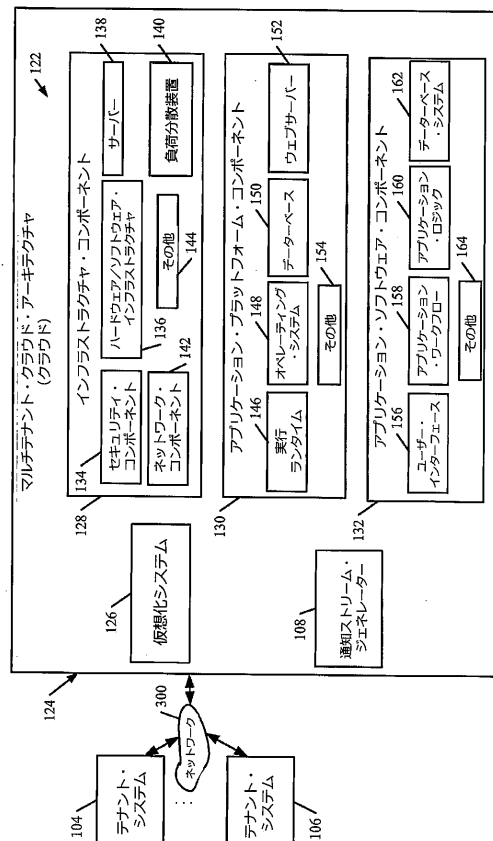


FIG. 4

【図 5】

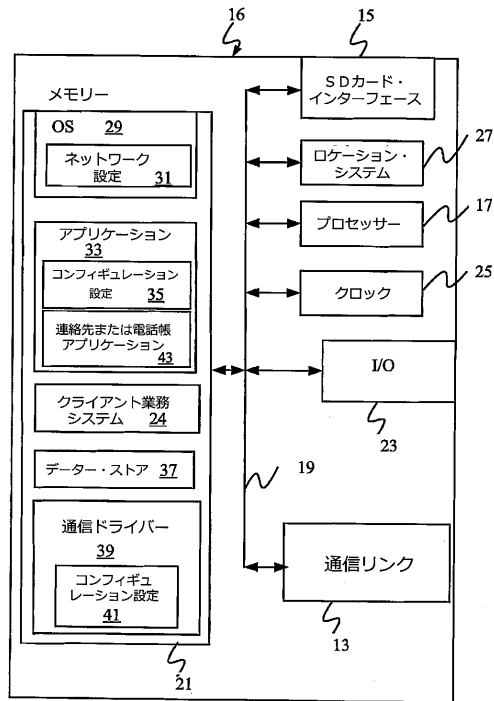


FIG. 5

【図 6】

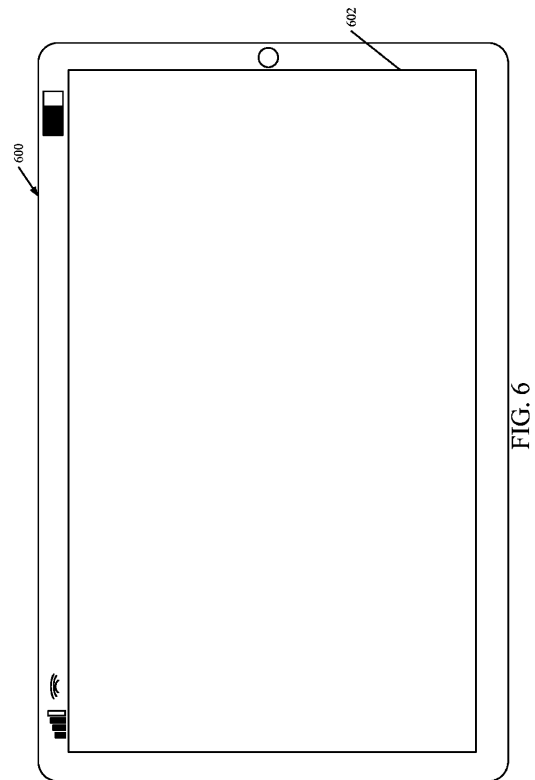


FIG. 6

【図 7】

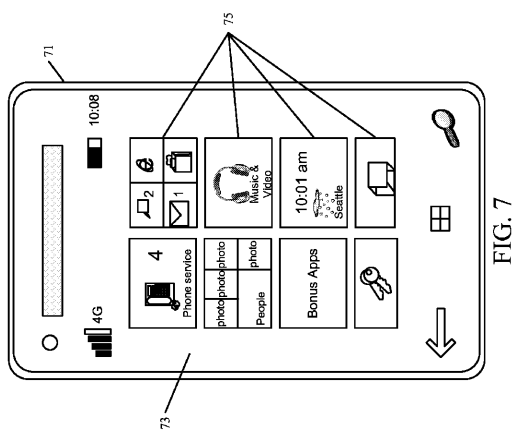


FIG. 7

【図 8】

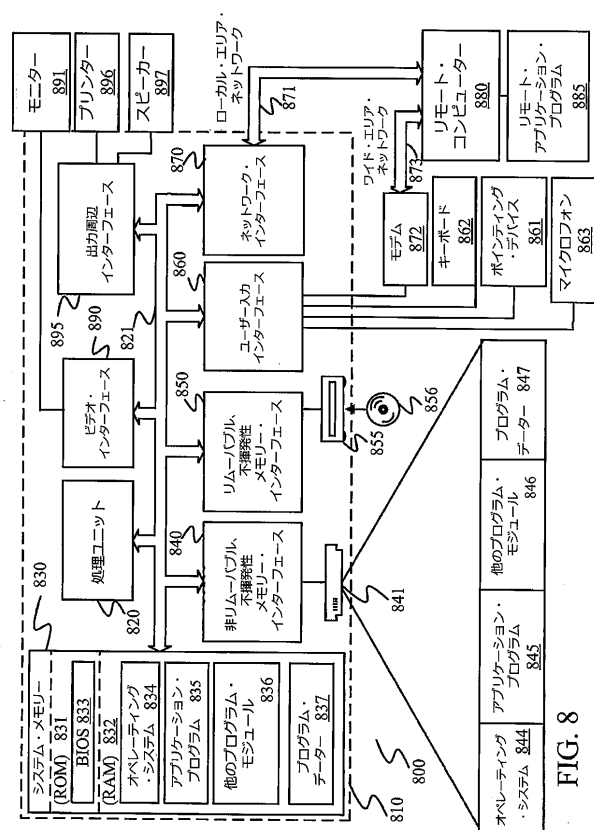


FIG. 8

## フロントページの続き

- (72)発明者 バンサル, ヴィブル  
アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ (8 / 1172)
- (72)発明者 ゴイス, マーカス・ヴィニシウス・シルヴァ  
アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ (8 / 1172)
- (72)発明者 コバーン, マーク  
アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ (8 / 1172)
- (72)発明者 ガウラヴ, スラジュ  
アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ (8 / 1172)
- (72)発明者 パッタピラマン, スワミナサン  
アメリカ合衆国 ワシントン州 98052 レッドモンド ワン マイクロソフト ウェイ, マイクロソフト テクノロジー ライセンシング, エルエルシー, エルシーエイ - インターナショナル・パテンツ (8 / 1172)

審査官 早川 学

- (56)参考文献 特表2014-504409(JP, A)  
特開2014-106728(JP, A)

- (58)調査した分野(Int.Cl., DB名)  
G06F 16/00 - 16/958