

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第5579727号

(P5579727)

(45) 発行日 平成26年8月27日(2014. 8. 27)

(24) 登録日 平成26年7月18日(2014. 7. 18)

(51) Int. Cl.

F I

G O 6 F 15/00 (2006. 01)

G O 6 F 15/00 4 2 0 A

G O 6 F 13/00 (2006. 01)

G O 6 F 13/00 5 0 0 A

請求項の数 18 (全 24 頁)

(21) 出願番号 特願2011-533396 (P2011-533396)  
 (86) (22) 出願日 平成21年10月24日(2009. 10. 24)  
 (65) 公表番号 特表2012-507087 (P2012-507087A)  
 (43) 公表日 平成24年3月22日(2012. 3. 22)  
 (86) 国際出願番号 PCT/US2009/061974  
 (87) 国際公開番号 W02010/048597  
 (87) 国際公開日 平成22年4月29日(2010. 4. 29)  
 審査請求日 平成24年10月15日(2012. 10. 15)  
 (31) 優先権主張番号 12/258, 149  
 (32) 優先日 平成20年10月24日(2008. 10. 24)  
 (33) 優先権主張国 米国 (US)

(73) 特許権者 500046438  
 マイクロソフト コーポレーション  
 アメリカ合衆国 ワシントン州 9805  
 2-6399 レッドモンド ワン マイ  
 クロソフト ウェイ  
 (74) 代理人 100107766  
 弁理士 伊東 忠重  
 (74) 代理人 100070150  
 弁理士 伊東 忠彦  
 (74) 代理人 100091214  
 弁理士 大貫 進介

最終頁に続く

(54) 【発明の名称】 契約の組み込みとバインドによるクライアントサービスへの機能の提供

(57) 【特許請求の範囲】

【請求項 1】

分散コンピューティング環境において目的のサービスに組み込まれた契約をバインドする方法を、コンピュータに実行させるプログラムであって、前記方法は、

クライアントサービスが依存する機能面を満たす指示を受信するステップと、

前記クライアントサービスの機能面の依存性を満たす契約を識別するステップであって、前記識別される契約は、1つまたは複数のインタフェースを定義し、前記1つまたは複数のインタフェースの動作を部分的に統治するための制約を特定するプロパティのセットを保持する、ステップと、

前記1つまたは複数のインタフェースを確立して、調整するために、前記識別された契約を組み込むステップであって、

(a) 前記分散コンピューティング環境において前記1つまたは複数のインタフェースを確立するサブステップと、

(b) 前記プロパティのセットから得た値を、前記1つまたは複数の確立したインタフェースに与えることによって、前記1つまたは複数の確立したインタフェースを構成するサブステップであって、該1つまたは複数の確立したインタフェースは、構成されると、前記クライアントサービスが前記目的のサービスの一部に到達することを可能にし、前記クライアントサービスから前記目的のサービスへの通信を管理する時に前記契約によって特定される前記制約を実施するサブステップと、

を含む、前記識別された契約を組み込むステップと、

10

20

前記組み込まれた契約を前記目的のサービスにバインドするステップであって、

(a) 前記 1 つまたは複数の確立したインタフェースを、1 つまたは複数のロールインスタンスにリンクするサブステップであって、前記 1 つまたは複数のロールインスタンスは、実行されると、前記目的のサービスに機能を与える一種のコンポーネントプログラムを表す少なくとも 1 つのロールの複製を生じさせる、サブステップと、

(b) 前記目的のサービスの実行の管理を担当するファブリックコントローラを介して、前記リンクをマッピングするサブステップと、

を含む、バインドするステップと

を含む、プログラム。

#### 【請求項 2】

前記制約は、前記目的のサービスへのアクセスを制限することと、前記クライアントサービスを認証することと、前記目的のサービスへの接続を開始または終了することと、前記目的のサービスへのトラフィックの流れをフィルタリングすることと、いずれのポートがコールを受信するために使用されるかを定義することと、のうちの少なくとも 1 つに関連し、

前記方法は、前記分散コンピューティング環境内のノードに前記目的のサービスの 1 つまたは複数のロールインスタンスをデプロイするための詳細を保持するサービスモデルを提供するステップをさらに含み、該デプロイするための詳細は、前記 1 つまたは複数のロールインスタンス上に配置される入力エンドポイントと出力エンドポイントとの間の相互接続を構築する、請求項 1 に記載のプログラム。

#### 【請求項 3】

前記 1 つまたは複数の確立したインタフェースは、前記組み込まれた契約をバインドすると、前記 1 つまたは複数のロールインスタンスの前記入力エンドポイントにリンクされる、請求項 2 に記載のプログラム。

#### 【請求項 4】

前記 1 つまたは複数の確立したインタフェースと 1 つまたは複数のロールインスタンスをリンクするステップは、前記分散コンピューティング環境を横断する通信チャネルを割り当てて、前記 1 つまたは複数の確立したインタフェースを前記リンクされた入力エンドポイントに動作可能に結合するサブステップ

を含む、請求項 3 に記載のプログラム。

#### 【請求項 5】

前記割り当てられる通信チャネルは、前記 1 つまたは複数の確立したインタフェースを前記 1 つまたは複数のロールインスタンスの前記入力エンドポイントにリンクするロードベアリング (LB) チャネルを含み、

前記方法は、

前記 1 つまたは複数の確立したインタフェースで、前記クライアントサービスからのコールを受信するステップと、

LB 機構を呼び出し、前記 LB チャネルの利用可能な通信チャネルに前記コールを分配するステップと

をさらに含む、請求項 4 に記載のプログラム。

#### 【請求項 6】

前記方法は、前記 1 つまたは複数の確立したインタフェースにリンクされる前記 1 つまたは複数のロールインスタンスの前記入力エンドポイントのロケーションを探すために、ネットワークアドレスを識別するステップをさらに含む、請求項 4 に記載のプログラム。

#### 【請求項 7】

前記方法は、前記ネットワークアドレスを、前記クライアントサービスに見えるようにするステップをさらに含む、

前記クライアントサービスのコンポーネントプログラムは、前記ネットワークアドレスを利用して、前記 1 つまたは複数のロールインスタンスにコールをルーティングするように構成される、請求項 6 に記載のプログラム。

## 【請求項 8】

前記割り当てられた通信チャンネルは、前記 1 つまたは複数の確立したインタフェースを、前記 1 つまたは複数のロールインスタンスの前記入力エンドポイントにリンクする S L S チャンネルを含み、

前記方法は、

前記 1 つまたは複数の確立したインタフェースで、前記コールと前記ネットワークアドレスとを前記クライアントサービスから受信するステップと、

前記 S L S チャンネルのうち、前記 1 つまたは複数の確立したインタフェースを前記入力エンドポイントにリンクするように指定された通信チャンネルを介して、前記コールをルーティングするステップと

10

をさらに含む、請求項 7 に記載のプログラム。

## 【請求項 9】

前記バインドするステップは、前記 1 つまたは複数の確立したインタフェースによって実施される前記特定された制約に基づいて、前記 1 つまたは複数のロールインスタンスを再構成するステップ

をさらに含む、請求項 1 に記載のプログラム。

## 【請求項 10】

バインドされた契約を分散コンピューティング環境内のクライアントサービスにバインドするコンピュータ化された方法であって、

前記クライアントサービスから、該クライアントサービスの 1 つまたは複数のコンポーネントプログラムの依存性を満たす指示を受信するステップと、

20

前記クライアントサービスの前記 1 つまたは複数のコンポーネントプログラムの前記依存性を満たす機能を抽象化したものを公開する契約であって、分散コンピューティング環境内で組み込まれ、前記機能を実行する目的のサービスにバインドされる契約を割り当てるステップと、

前記クライアントサービスを、その動作を開始するようにデプロイするステップであって、

(a) 前記 1 つまたは複数のコンポーネントプログラムを前記割り当てられたバインドされる契約によって定義されたインタフェースにリンクするサブステップであって、該インタフェースは、前記割り当てられたバインドされる契約を組み込むと前記分散コンピューティング環境内で確立され、前記バインドされる契約は、前記インタフェースに関連付けられたプロパティのセットを保持する、サブステップと、

30

(b) 前記目的のサービスの実行の管理を担当するファブリックコントローラに、前記リンクの詳細を書き込むサブステップと、

(c) 前記プロパティのセットから得た値を前記確立したインタフェースに与えることによって、前記確立したインタフェースの動作を部分的に統治するための制約を、前記確立したインタフェースに適用するサブステップと、

(d) 前記 1 つまたは複数のコンポーネントプログラムを構成するために、前記適用された制約を前記クライアントサービスに公開するサブステップと、

を含む、前記クライアントサービスをデプロイするステップと

40

を含む、コンピュータ化された方法。

## 【請求項 11】

前記 1 つまたは複数のコンポーネントプログラムをインタフェースにリンクするステップは、

前記 1 つまたは複数のコンポーネントプログラムに配置された出力エンドポイントを、それぞれ識別するサブステップと、

前記分散コンピューティング環境内の通信チャンネルを割り当てて、前記確立したインタフェースを前記出力エンドポイントに動作可能に結合する、サブステップと

をさらに含む、請求項 10 に記載のコンピュータ化された方法。

## 【請求項 12】

50

前記目的のサービスは、少なくとも1つのロールの1つまたは複数のロールインスタンスを含み、

前記少なくとも1つのロールは、前記クライアントサービスの依存性を満たす前記機能を実現する、前記目的のサービスの他のロールと共に動作するコンポーネントの個々のクラスを表す、請求項11に記載のコンピュータ化された方法。

【請求項13】

前記1つまたは複数のコンポーネントプログラムを構成することは、前記1つまたは複数のコンポーネントプログラムの出力エンドポイントから発せられたコールが前記1つまたは複数のロールインスタンスの基本となるプロトコルと互換性を有するように、前記コールをフォーマットするために、前記適用された制約から指示を引き出すこと

10

を含む、請求項12に記載のコンピュータ化された方法。

【請求項14】

前記1つまたは複数のコンポーネントプログラムの前記出力エンドポイントから発せられた前記コールは、前記分散コンピューティング環境の前記割り当てられた通信チャネルを介して、前記確立したインタフェースにルーティングされる、請求項13に記載のコンピュータ化された方法。

【請求項15】

前記確立したインタフェースは、前記制約が適用されると、前記クライアントサービスのIDと共に前記適用された制約に基づいて、前記発せられたコールを前記1つまたは複数のロールインスタンスにリレーするか、前記発せられたコールをフィルタリングするように構成される、請求項13に記載のコンピュータ化された方法。

20

【請求項16】

前記クライアントサービスのIDは、前記ファブリックコントローラによって記録され、前記1つまたは複数のロールインスタンスに前記発せられたコールをリレーすると、前記発せられたコールに添付された要求を検査して、前記コールのリクエストを引き受けるか否かが確定される、請求項15に記載のコンピュータ化された方法。

【請求項17】

前記要求を前記発せられたコールに添付することは、

前記1つまたは複数のコンポーネントプログラムが前記コールを発したことを検証するために、前記クライアントサービスのIDにアクセスすることと、

30

前記クライアントサービスのIDと前記クライアントサービスの他の特性を、前記要求に組み込むことによって、前記目的のサービスにおける個々のセキュリティレベルを保証することと、

を含む、請求項16に記載のコンピュータ化された方法。

【請求項18】

分散コンピューティング環境で利用可能な契約を組み込んでバインドすることによって、クライアントサービスを目的のサービスにリンクする方法を行うコンピュータシステムであって、

プロセッサ及びメモリを備え、

当該コンピュータシステムは、クライアントサービスから、該クライアントサービスの1つまたは複数のコンポーネントプログラムの依存性を満たす指示を受け取るように構成され、

40

目的のサービスは、1つまたは複数のロールインスタンスを含み、前記1つまたは複数のロールインスタンスは、実行されると前記目的のサービスに機能を与える一種のコンポーネントプログラムを表す少なくとも1つのロールの複製を生じさせ、

当該コンピュータシステムは、前記目的のサービスの前記機能を抽象化したものを公開する契約であって、前記クライアントサービスの前記1つまたは複数のコンポーネントプログラムの前記依存性を満たし、少なくとも1つのインタフェースを定義して、前記少なくとも1つのインタフェースによって実施される制約を特定する契約を識別するように構成され、

50

当該コンピュータシステムは、前記目的のサービスと前記クライアントサービスに前記契約をバインドするファブリックコントローラをさらに備え、前記バインドするプロセスは、

(a) 前記契約を組み込むことで、前記分散コンピューティング環境内に少なくとも1つのインタフェースを確立するステップと、

(b) 前記プロパティのセットから得た値を、前記少なくとも1つの確立したインタフェースに与えることによって、前記少なくとも1つの確立したインタフェースを構成するステップであって、前記少なくとも1つの確立したインタフェースは、構成されると、前記クライアントサービスが前記少なくとも1つの確立したインタフェースを介して前記目的のサービスの一部に到達することを可能にし、前記クライアントサービスと前記目的のサービスとの間の通信を管理する時に前記契約によって特定される前記制約を実施する、ステップと、

を含み、

前記ファブリックコントローラは、前記少なくとも1つの確立したインタフェースを介して、前記クライアントサービスの前記1つまたは複数のプログラムコンポーネントを、前記目的のサービスの前記1つまたは複数のロールインスタンスにリンクするように構成されるコンピュータシステム。

【発明の詳細な説明】

【技術分野】

【0001】

本発明は、契約(contract)と呼ぶ媒介を介して、サービスアプリケーションの機能面を自動的に明示することに関する。

【背景技術】

【0002】

一般的に、開発者は、ソフトウェアアプリケーションを、その設定にかなりの自由度を持たせて書く。一例として、開発者は、そのソフトウェアアプリケーションをサポートするために提供される個々のプラットフォームの特定の制約内で動作するソフトウェアアプリケーションを確立することで、この自由度を活用することができる。従って、ソフトウェアアプリケーションに関連付けられたこの自由度によって、ソフトウェアアプリケーションは、プラットフォームと協働して動作することができる。

【0003】

一例では、ソフトウェアアプリケーションを開発するアプリケーションサービス・プロバイダは、ソフトウェアアプリケーションのこの設定を採用して、インターネットを介して遠隔でアクセス可能なプラットフォーム上で動作させることができる。この例では、ユーザがソフトウェアアプリケーションを用いて遠隔でファイルを操作できるように、プラットフォームはソフトウェアプログラムを実行する。従って、プラットフォームは、プラットフォーム上で実行するソフトウェアアプリケーションの基本的要素を確立するようになされて、遠隔で使用する現在の負荷を引き受ける。ソフトウェアアプリケーションの自由度によって、これら基本的要素を増減でき、それら要素間の協調を管理することができる。しかしながら、これら基本的要素の機能を公開する方法がないので、対象のソフトウェアアプリケーション以外のソフトウェアプログラムに、この機能を利用する能力を提供することは、不可能である。さらに、仮に他のソフトウェアプログラムが、現在実行中の基本的要素の機能を認識したとしても、ソフトウェアアプリケーションを自動的にリンクしたり、自動的に基本的要素を構成したりして、その機能を遠隔で活用するのを可能にする手段が存在しない。

【0004】

ソフトウェアアプリケーションの基本的要素を構成するための現在の解決法は、プラットフォームの管理者が基本的要素を手動で設定することによっている。このその場限りの解決法は、労力を要し、エラーが起りやすく、基本的要素を別のソフトウェアプログラムにリンクさせることは含まない。さらに、多数のソフトウェアアプリケーションの動作

10

20

30

40

50

をサポートするプラットフォームのサイズが、多数の相互接続したハードウェアコンポーネントを含んで拡張されると、この手動設定の欠点は増大される。

【発明の概要】

【0005】

この概要は、詳細な説明で下記にさらに記載する概念を簡単な形で紹介するものである。この概要は、特許を請求する内容の主要な特徴または不可欠な特徴を特定することを意図するものではなく、特許を請求する内容の範囲の画定に用いられることを意図してもない。

【0006】

本発明の実施形態は、方法、システム、および、コンピュータ実行可能命令を有するコンピュータ記憶媒体に関する。コンピュータ記憶媒体は、コンピュータ実行可能命令が実行されると、本明細書の実施形態による、本明細書で契約（contract）と呼ぶ媒介を介して、目的のサービス（例えば、分散コンピューティング環境で実行するサービスアプリケーション）の機能面をクライアントサービスに自動的に明示する方法を行う。一般的に、その方法は、サービスアプリケーション（単数または複数）の動作の基礎となるように構成された分散コンピューティング環境という状況で行われる。実施形態においては、明示された機能面がクライアントサービスの依存性を満足させることが確かめられると、契約が割り当てられる。割り当てられると、本発明の方法は、分散コンピューティング環境内の契約を組み込み、組み込まれた契約をサービスアプリケーションのコンポーネントプログラムにバインド（bind）することを含むうる。

【0007】

一般的に、契約は、インタフェースを定義し、インストール時に、インタフェースを構成するプロパティを保持する。契約の組み込み時に、そのインタフェースの中の1つが、確立され、それに関連付けられたプロパティに従って、パラメータを決定される。組み込まれた契約のバインド時に、目的のサービスを構成するコンポーネントプログラムの入力エンドポイントは、通信チャネルを介して、確立したインタフェースにリンクされる。従って、確立したインタフェースに到達できる他のサービスアプリケーションは、目的のサービスの機能面へアクセスできる。

【0008】

バインドのプロセスは、クライアントサービスを含むコンポーネントプログラムの出力エンドポイントを、確立したインタフェースにリンクするプロシージャ、および、確立したインタフェースのパラメータを決定することに基づいて目的のサービスを構成するプロシージャを含むうる。さらに、クライアントサービスは、出力エンドポイントから発せられるコールを、目的のサービスの一定の特性に形式を合わせるように構成されうる。従って、クライアントサービスからのコールは、通信チャネルを介してルーティングされて目的のプログラムに到達でき、目的のサービスの構成と互換性をもたせて、目的のサービスの機能面を適切に利用できるようになりうる。

【0009】

実施形態において、目的のサービスは、コール、および、そのコールに付随する要求を受信すると、クライアントサービスのIDを確かめうる。目的のサービスは、一般的に、クライアントサービスのIDに動的に反応することができる。従って、クライアントサービスのIDを理解すると、目的のサービスは、コールに答えるとき、結果として、サービスのレベルを調整し（例えば、サービスの機能面を操作する）、個々のクライアントサービスを提供することができる。

【0010】

この概要は、詳細な説明で下記にさらに記載する概念を簡単な形で紹介するものである。この概要は、特許を請求する内容の主要な特徴または不可欠な特徴を特定することを意図するものではなく、特許を請求する内容の範囲の画定に用いられることを意図してもない。

【0011】

本発明の実施形態を、添付図面を参照して、下記に詳細に述べる。

【図面の簡単な説明】

【0012】

【図1】本発明の実施形態の実装に使用するのに適した例示的コンピュータ環境のブロック図である。

【図2】組み込まれた契約を目的のサービスにバインドするように構成された、本発明の実施形態の実装に使用するのに適した例示的分散コンピューティング環境を示すブロック図である。

【図3】目的のサービスにバインドされた契約を、クライアントサービスにバインドするように構成された、本発明の実施形態の実装に使用するのに適した例示的分散コンピューティング環境を示すブロック図である。

10

【図4】本発明の実施形態による、サービスアプリケーション間の通信をルーティングするロードバランシング(LB)チャネルを利用する例示的ファブリックコントローラの図式的な表示である。

【図5】本発明の実施形態による、サービスアプリケーション間の通信をルーティングするSLS(stateless-switch)チャネルを利用する例示的ファブリックコントローラの図式的な表示である。

【図6】本発明の実施形態による、契約を自動的に組み込み、その組み込まれた契約を目的のサービスにバインドする方法全体を示すフロー図である。

【図7】本発明の実施形態による、クライアントサービスの依存性に基づいて、組み込まれた契約を自動的に割り当て、割り当てられた契約をクライアントサービスにバインドする方法全体を示すフロー図である。

20

【発明を実施するための形態】

【0013】

本発明の実施形態の内容を、法的要件を満たすように、特定性をもって本明細書に記載する。しかしながら、記載そのものは、特許の範囲を限定することを意図するものではない。むしろ、特許を請求する内容は、現行または今後の他の技術と組み合わせて、本明細書に記載されたステップと異なったステップや、または類似のステップを組み合わせたものなど、他の方法で実施してもよいと、発明者は考えている。

【0014】

30

本発明の実施形態は、方法、システム、および、コンピュータ実行可能命令を有するコンピュータ記憶媒体に関し、コンピュータ記憶媒体は、実行されると、本明細書の実施形態による、クライアントサービスの動作をサポートするのに利用可能な一定の機能を期待して書き込まれたクライアントサービスを自動的に調整する方法のためのコンピュータ実行可能命令を有する。クライアントサービスが依存するこれらの機能は、契約によって公開され、クライアントサービスが、分散コンピューティング環境内で実行する目的のサービスの機能に到達し、それらを採用することを可能にするための媒介として働く。適切な契約は、クライアントサービスが満たすことを求める依存性を満足させる機能を、その契約が公開するか否かに基づいて、割り当てられうる。そうすると、割り当てられた契約を組み込み(例えば、分散コンピューティング環境内にインタフェースを確立)、目的のサービスとクライアントサービスにバインドされる(例えば、目的のサービスとクライアントサービスのコンポーネントプログラムを、確立したインタフェースを介してリンクする)。従って、クライアントサービスの実行を可能にするのに要求される機能が、自動的に発見され、クライアントサービスにリンクされる。

40

【0015】

従って、一態様においては、本発明の実施形態は、コンピュータ実行可能命令を有する1つまたは複数のコンピュータ可読媒体に関する。このコンピュータ実行可能命令を実行すると、組み込まれた契約を分散コンピューティング環境内の目的のサービスにバインドする方法が提供される。最初に、この方法は、インタフェースの集合を定義する契約を識別し、識別された契約を組み込み、分散コンピューティング環境内のインタフェースの集

50

合のうちから1つのインタフェースを確立する。典型的には、契約は、各インタフェースをインストールするためのプロパティのセットを保持しており、組み込まれた契約は、クライアントサービスが目的のサービスの一部に到達するための媒介として働く。さらに、方法は、確立したインタフェースに関連付けられたプロパティのセットから得た値で、確立したインタフェースのパラメータを決定することによって、組み込まれた契約を目的のサービスにバインドすることを含んでもよい。実施形態において、バインドのプロセスは、確立したインタフェースと1つまたは複数のロールインスタンスを自動的にリンクすること、および、目的のサービスの実行の管理を担当するファブリックコントローラを介して、そのリンクをマッピングすることを含む。一般的に、ロールインスタンス (role instance) は、一種のコンポーネントプログラムを表す少なくとも1つのロール (role) の複製を体現し、そのコンポーネントプログラムは、実行されると、目的のサービスに機能を与えるものである。

10

#### 【0016】

別の態様においては、本発明の実施形態は、目的のサービスに既にバインドされた組み込まれた契約を、分散コンピューティング環境内のクライアントサービスにバインドするためのコンピュータ化された方法に関する。実施形態においては、この方法は、クライアントサービスの依存性を満たす指示をクライアントサービスから受信すること、および、クライアントサービスのコンポーネントプログラムの依存性を満たす機能を抽象化したものを公開する契約を割り当てることを含む。典型的には、契約は、前もって、分散コンピューティング環境内で組み込まれ、機能を実行する目的のサービスにバインドされる。この方法は、クライアントサービスをデプロイして、その動作を開始することをさらに含んでもよい。例示の実施形態においては、デプロイメントは、割り当てられたバインドされた契約が定義するインタフェースに、1つまたは複数のコンポーネントプログラムを自動的にリンクすることを含み、割り当てられたバインドされた契約を組み込むと、そのインタフェースが分散コンピューティング環境内に確立されて、リンクの記述を目的のサービスの実行の管理を担当するファブリックコントローラに書き込む。実施形態において、この方法は、最初に、サービスアプリケーションのロールのインスタンスの数を増やす指示を受信することを含むが、これに限定されるものではない。上述のように、ロールは、サービスアプリケーションの他のロールと共に動作して分散された機能を実現する、個々のクラスのコンポーネントを表す。一例としては、サービスアプリケーションの遠隔使用のワークロードにおける少なくとも1つの変化、または、データセンタの1つもしくは複数のノードがオフラインになることを含むイベントが起こると、指示が生じる。従って、本発明が企図するこれらのイベントおよび他のイベントは、分散したデータセンタ内で、サービスアプリケーションの追加のロールをインストールするのが望ましい状況を促進してもよい。

20

30

#### 【0017】

さらに別の態様において、本発明の実施形態は、契約を組み込み、バインドして、分散コンピューティング環境で利用できるようにすることによって、クライアントサービスを目的のサービスに自動的にリンクすることができるコンピュータシステムに関する。一般的に、データセンタは、分散コンピューティング装置を含む。コンピュータシステムは、複数のコンピュータソフトウェアコンポーネントを体現するコンピュータ記憶媒体を含んでもよい。最初に、コンピュータソフトウェアコンポーネントは、サービスアプリケーション (例えば、クライアントサービスおよび目的のサービス)、契約、および、分散コンピューティング環境を管理するように構成されるファブリックコントローラを含む。一般的に、クライアントサービスは、1つまたは複数のコンポーネントプログラムを含み、一方、目的のサービスは、1つまたは複数のロールインスタンスを含み、このロールインスタンスは、一種のコンポーネントプログラムを表す少なくとも1つのロールの複製を実現し、このコンポーネントプログラムは、実行されると、目的のサービスに機能を与えるものである。動作時、クライアントサービスは、クライアントサービスの依存性を満たす指示を出すように構成される。契約は、目的のサービスの機能を抽象化したものを公開して

40

50

もよく、それによって、クライアントサービスのコンポーネントプログラムの依存性が満たされる。契約は、さらに、少なくとも1つのインタフェースを定義するように構成される。ファブリックコントローラは、次のプロセスの1つまたは複数を順不同で行うよう構成される。すなわち、契約を組み込むことによって、分散コンピューティングプラットフォームにインタフェースを確立するプロセス、契約を目的のサービスとクライアントサービスにバインドするプロセス、および、確立したインタフェースを介して、クライアントサービスのプログラムコンポーネントを目的のサービスのロールインスタンスに自動的にリンクするプロセスである。

#### 【0018】

一般的に、目的のサービスのロールインスタンスのインスタンス化と協調的管理は、サービスモデルによって容易になる（図2の参照番号250を参照）。本明細書においては、「サービスモデル」という語は、限定的な意味ではなく、一般的に、分散コンピューティング環境内で目的のサービスのインスタンスを確立および管理することに関する情報を含む任意の通信に関する。一例においては、サービスモデルは、目的のサービスのどのロールを確立すべきか、または、データセンタ内に各ロールのインスタンスをどのようにインストールして起動すべきかの記述を含む。すなわち、サービスモデルは目的のサービスのためにはどのロールを実行すべきか、および、そのロールのインスタンスをどこにインストールすべきかに関する条件を明確に述べる役割をする。

#### 【0019】

さらに、サービスモデルは、分散コンピューティングセンター（図2および図3の参照番号200を参照）内に、ロールのインスタンスをサポートするための1つまたは複数のノード（例えば、図2、図3のノードI221, II222, III223, IV224, V225）を割り当ててもよい。これは、ファブリックコントローラが実行してもよい。従って、サービスモデルは、個々の実施形態において、目的のサービスやクライアントサービスのロールインスタンス等のコンポーネントプログラムを管理する指示を与えるインタフェース設計図として働く。すなわち、サービスモデルは、分散コンピューティング環境にわたる分散されたロケーションにデプロイされると、ファブリックコントローラが協調してコンポーネントプログラム間の活動を導くのを助ける。これらのロケーションは、典型的には、サービスモデル内のデプロイメントの詳細に記載される。一般的に、「デプロイメントの詳細」という語は、限定的な意味ではなく、ノード上のロールインスタンスのインスタンス化を管理する機構、ロールインスタンス間の通信路としてどの通信チャネルを利用するか識別する機構、および／または、目的のサービスを実行する個々の方法を記述する情報を提供する機構、を指して使われる。

#### 【0020】

目的のサービスのロールインスタンス（例えば、図2の目的のサービス205のロールA261とロールB262）は、一般的に、少なくとも1つのロールの複製をさす。一般的に、本明細書において使われるように、「ロール」という語は、広く、目的のサービスの他のロールと共に動作して、クライアントサービスの期待される依存性を満たす機能を実現する任意のクラスのコンポーネントを指す。

#### 【0021】

目的のサービスの動作、および、そのサービスの機能面を開始するために、サービスモデルは、デプロイメントの詳細と協働して、分散コンピューティング環境のノード上に、ロールインスタンスをインスタンス化する。インスタンス化は、ロールインスタンスをホストするために利用可能であることが確認されたノードを割り当てること、割り当てられたノードにロールインスタンスを配置すること、配置されたロールインスタンスを構成すること、および、ロールインスタンスに配置された入力エンドポイントと出力エンドポイント間を相互に接続すること、を最初を含む。より詳細に後述するように、契約を組み込むと、インタフェースは、ロールインスタンスの入力エンドポイントにリンクされ、目的のサービスの機能の一部へのアクセスを促進しうる。

#### 【0022】

一般的には、分散コンピューティング環境内のノードを利用して、ロールインスタンスの動作を引き受ける。本明細書において使用するように、「ノード」という語は、限定的な意味ではなく、例えば、パーソナルコンピュータ、デスクトップコンピュータ、ラップトップコンピュータ、携帯用装置、携帯電話機、家庭用電子機器など、全ての形態のコンピュータ装置を包含する。一態様においては、ノードは、ネットワーククラウドを介して相互に接続された分散された複数のコンピュータ装置のうちのコンピュータ装置を表す。一般的に、これらの分散されたコンピュータ装置は、サービスアプリケーションの様々なロールの複数のインスタンスをホストすることができる。一例として、個々のノードは、それぞれ、ロールインスタンス（単数または複数）をサポートする2つ以上のホスト環境を収容可能であってもよい。これらのロールインスタンスは、ノード上で完全に独立して実行してもよく（すなわち、サービスアプリケーションに高水準のセキュリティを適応する）、他のロールと部分的に通信して実行してもよく、または、サービスアプリケーションの1つまたは複数の他のロールとインタラクティブな状態で実行してもよい。

10

#### 【0023】

動作すると、実行中の目的のサービスをクライアントサービスにバインドして、クライアントサービスに書き込まれた期待される依存性を満たしうる。契約は、典型的には、バインドのプロセスを進めるために本発明が採用した媒介である。一実施形態においては、契約は、実行中の目的のサービスに期待するもの（すなわち、目的のサービスの機能）の抽象的な定義を公開する。別の実施形態においては、契約は、インタフェースの集合を定義し、そのインタフェースの各々に関連付けられたプロパティのセットを保持する。一般的に、インタフェースは、1つまたは複数の点で、関連付けられている。実施形態においては、プロパティを利用して、分散コンピューティング環境へのインストール時に、インタフェースを調整またはインタフェースのパラメータを決定する。一例として、プロパティは、ノードのプロトコルに部分的に依存してもよい。ファブリックコントローラが目的のサービスを見つけ、クライアントサービスのコンポーネントプログラムが目的のサービスに成功裡に到達できるように構成できるように、これらプロパティは、目的のサービスを作成するとき、適切な情報で満たされる。

20

#### 【0024】

下記のように、契約を組み込み（例えば、定義されたインタフェースの集合のうちの1つを確立）、目的のサービスにバインドしうる。ファブリックコントローラは、目的のサービスを、1つまたは複数の目的のサービスに、目的のサービス（単数または複数）の機能に部分的に基づいて、バインドすることを選択してもよい。従って、契約は、複数のサービスアプリケーションにバインドされうる。しかしながら、契約の組み込み中に確立されるインタフェースは、選択された目的のサービスのノード、ロールインスタンス等の特性に従って、異なった構成であってもよい。

30

#### 【0025】

ここまで、本発明の実施形態全体の概観を簡潔に述べたが、本発明の実施形態を実装するのに適した例示的な動作環境について、下記に記載する。

#### 【0026】

図面を参照する。特に、まず、図1を参照すると、本発明の実施形態を実装する例示的な動作環境が示されており、それをコンピュータ装置100と総称する。コンピュータ装置100は、適切なコンピュータ環境の一例に過ぎず、本発明の実施形態の使用または機能の範囲に関していかなる限定を示唆するものでもない。また、コンピュータ環境100が、図示のコンポーネントの任意の1つまたは任意の組み合わせに関して依存性または要件を有すると解釈すべきでもない。

40

#### 【0027】

本発明の実施形態は、パーソナルデータアシスタントまたは他の携帯用デバイス等のコンピュータまたは他の機械で実行される、コンポーネントプログラム等のコンピュータ実行可能命令などのコンピュータコードまたは機械で使用可能な指示という一般的な状況で、記載してよい。一般的に、ルーチン、プログラム、オブジェクト、コンポーネント、デ

50

ータ構造、などを含むコンポーネントプログラムは、個々のタスクを行うコード、または、個々の抽象データ型を実行するコードを指す。本発明の実施形態は、携帯装置、家庭用電化製品、汎用コンピュータ、専用コンピュータ装置などの様々なシステム構成で実践してもよい。本発明の実施形態は、通信ネットワークを通してリンクされる遠隔処理装置によってタスクを行う分散コンピューティング環境で実践されてもよい。

#### 【0028】

引き続き図1を参照すると、コンピュータ装置100は、バス110を含み、バス110は、メモリ112、1つまたは複数のプロセッサ114、1つまたは複数のプレゼンテーションコンポーネント116、入力／出力（I／O）ポート118、I／Oコンポーネント120、および、例示の電源122を直接または間接的に結合する。バス110は、1つまたは複数のバス（アドレスバス、データバス、またはそれらの組み合わせなど）を表す。図1の様々なブロックは、分かりやすいように線で示されているが、実際には、様々なコンポーネントの線引きは、それほど明確ではなく、比喩的であり、その線は、より正確には中間的であいまいである。例えば、ディスプレイ装置などのプレゼンテーションコンポーネントをI／Oコンポーネントとみなしてもよい。また、プロセッサはメモリを有する。本発明の発明者は、技術の性質はそのようなものであり、図1の図は本発明の1つまたは複数の実施形態に関連して用いることができる例示的なコンピュータ装置を示したものに過ぎないことを繰り返して述べる。全て図1の範囲内で考えられている「ワークステーション」、「サーバ」、「ラップトップ」、「携帯装置」などのカテゴリと、「コンピュータ」または「コンピュータ装置」との間に違いを設けていない。

#### 【0029】

コンピュータ装置100は、典型的には、様々なコンピュータ可読媒体を含む。一例として、コンピュータ可読媒体は、RAM（Random Access Memory）、ROM（Read Only Memory）、EEPROM（Electrically Erasable Programmable Read Only Memory）、フラッシュメモリなどのメモリ技術、CDROM、DVD（Digital Versatile Disk）などの光学的またはホログラフィック媒体、磁気カセット、磁気テープ、磁気ディスク記憶装置などの磁気記憶装置、または、所望の情報のコード化に使用でき、コンピュータ装置100によってアクセス可能な他の任意の媒体を含んでもよいが、これに限定されるものではない。

#### 【0030】

メモリ112は、揮発性および／または不揮発性メモリ形式のコンピュータ記憶媒体を含む。メモリ112は、取り外し可能であってもよく、取り外し不能であってもよく、それらの組み合わせであってもよい。例示的なハードウェア装置としては、固体メモリ、ハードドライブ、光ディスクドライブなどがある。コンピュータ装置100は、メモリ112またはI／Oコンポーネント120等の様々なエンティティからデータを読み取る1つまたは複数のプロセッサを含む。プレゼンテーションコンポーネント（単数または複数）116は、ユーザや他の装置にデータ表示を提供する。例示的なプレゼンテーションコンポーネントとしては、ディスプレイ装置、スピーカ、印刷コンポーネント、振動コンポーネントなどがある。I／Oポート118によって、コンピュータ装置100は、論理的にI／Oコンポーネント120などの他の装置への結合が可能になる。その一部は、内蔵式でもよい。例示的なコンポーネントとしては、マイクロフォン、ジョイスティック、ゲームパッド、衛星放送受信アンテナ、スキャナ、プリンタ、ワイヤレス装置などがある。

#### 【0031】

図2を参照すると、本発明の実施形態の実装に使用するのに適した、分散コンピューティング環境200を示すブロック図が示されている。一般的に、分散コンピューティング環境200は、図3に示すように、組み込まれた契約235を目的のサービス205にバインドし、目的のサービス205にバインドされた契約をクライアントサービスにバインドするように構成されている。分散コンピューティング環境200は、データセンタ210を含み、そのデータセンタ210は、サービスモデル250に従って、コンポーネントプログラムの動作、または目的のサービス205のロールA261およびB262のイン

スタンスを有し、サポートするように構成されている。当業者には当然のことながら、図2に示すデータセンタ210は、1つまたは複数のサービスアプリケーション（例えば、目的のサービス205）を収容するのに適する一例に過ぎず、本発明の実施形態の使用または機能の範囲に関していかなる制限を示唆するものでもない。データセンタ210を、任意の単一のノード、ノードの組み合わせ（例えば、ノードI221, II222, III223）、リソース（図示せず）、またはリソース（図示せず）にアクセスするAPIのセットに関連付けられた任意の依存性または要件を有するものとして解釈すべきでもない。さらに、図2の様々なブロックは、分かりやすくするために線で示されているが、実際には、様々なコンポーネントの線引きは、それほど明確でなく、比喩的であり、線は、より正確には、中間的であいまいである。

10

#### 【0032】

データセンタ210は、様々なノード（例えば、ノードI221, II222, III223）、各ノード上で実行するオペレーティングシステム、ロールインスタンスA261およびB262、インタフェース（例えば、インタフェース220）、および、多くは、ファブリックコントローラ215を含み、このファブリックコントローラ215は、ノードI221, II222、およびIII223にローカルにインストールされるファブリックエージェント（図示せず）を含んでもよい。ファブリックエージェントは、ファブリックコントローラ215の拡張機能として働き、特に、目的のサービス205のインストールおよび管理のために協働して機能する。さらに、ロールインスタンスA261およびB262は、入力エンドポイント（例えば、入力エンドポイント255）を介して互いに接続してもよく、入力エンドポイントからコールを発し、出力エンドポイントで、コールを受信する。一例においては、これらの接続のうちの1つまたは複数は、ネットワーククラウド（図示せず）を介して確立してもよい。様々な物理的リソースにわたって分散配置されているロールインスタンスA261、B262、および、インタフェース220が、それらの間の通信を確立するために互いのロケーションを認識するように、ネットワーククラウドは、上記に挙げたエンティティを相互に接続する。さらに、ネットワーククラウドは、インタフェース220をロールAのインスタンス261の入力エンドポイント255に動作可能に結合する通信チャネル290を介して、この通信を容易にする。一例として、ネットワーククラウドは、1つまたは複数のローカルリアネットワーク（LAN）および／または広域ネットワーク（WAN）を含むが、これに限定されるものではない。このようなネットワーク環境は、オフィス、企業規模のコンピュータネットワーク、イントラネット、およびインターネットでは一般的である。従って、本明細書ではこれ以上の記載を省略する。

20

30

#### 【0033】

さらに、本発明の実施形態は、図2に示す物理的リソースへの実装に限定されず、本明細書の実施形態の範囲内の様々な異なった種類のコンピュータ装置、機器、および、コンポーネントプログラムの任意のもので実装してもよいことに留意されたい。言い換えれば、データセンタ210の図示のノードI221, II222, III223は、説明のために、例示的な構成を示したに過ぎず、従って、コンピュータ業界では既知の、ノード、および、ノード上に常駐するロールインスタンスの任意の適切な配置を、本発明は用いてもよく、意図してもよい。

40

#### 【0034】

これらの例示的なノード、I221, II222, III223、およびデータセンタ210のロールインスタンスA261、B262は、サービスの契約を組み込み、組み込まれた契約235を目的のサービス205にバインドするという概念を導入する働きをする。それについて、今から説明する。最初に、サービスの契約は、識別を行う。一例においては、クライアントサービス（図3の参照番号305参照）に書き込まれた期待される依存性を満たすロールAのインスタンス261の機能260を抽象化したものを公開する契約が、識別される。識別された契約は、典型的には、インタフェースの集合を定義し、それぞれ、1つまたは複数のインタフェースに関連付けられたプロパティ240のセット

50

を保持する。動作時、プロパティ 240 のセットは、各インタフェースの構成をインストールし、調整する助けになる。

#### 【0035】

識別されたサービスの契約を組み込み、分散コンピューティング環境 200 のコンピュータ装置（例えば、ノード 1221）内に、インタフェースの集合のうちのインタフェース 220 を確立してもよい。詳しく前述したように、組み込まれた契約は、クライアントサービスが、目的のサービス 205 の機能 260 に到達するための媒介として働く。組み込みのプロセスは、確立したインタフェース 220 に関連付けられたプロパティ 240 のセットから得た値 230 で、確立したインタフェース 220 のパラメータを決定することを含んでもよい。一例においては、パラメータの決定は、インタフェース 220 内の暗黙のパラメータ 270 に、値 230 を与えることを含んでもよい。

10

#### 【0036】

組み込みのプロセスは、インタフェース 220 に制約 295 をインストールすることも含んでもよい。最初に、インタフェース 220 に関連付けられたプロパティ 240 のセットは、確立したインタフェース 220 の動作を部分的に統治する制約 295 を特定してもよい。さらに、確立したインタフェース 220 を、プロパティ 240 のセットから得た値 230 でパラメータを決定すると、分散コンピューティング環境 200 内で制約 295 を実施する。従って、制約 295 は、インタフェース 220 をどのように接続するかを決定する（例えば、サーバのどの外部ポートが、遠隔ウェブサーバアプリケーションからのコール 225 を受信するかを決定する）ガイドラインとして働き、一部では、インタフェース 220 をどのように構成するかを決定するガイドラインとして働く。一例として、プロパティ 240 が、特定のポート番号などの個々の制約 295 を特定すると、ファブリックコントローラ 215 は、データセンタ 210 内に個々の制約を割り当て、コール 225 が発せられると、個々の制約が目的とされるように設定する。従って、インタフェース 220 は、目的のサービス 210 の機能 260 への到達を試みると、これらの割り当てられたポート番号を使うように制限を受ける。

20

#### 【0037】

制約 295 は、インタフェース 220 の構成を支援してもよい。一例においては、制約 295 は、インタフェース 220 に、機能 260 へのアクセスを試みるものをフィルタリングするように促し、それによって目的のサービス 205 へのトラフィックの流れを制限してもよい。別の例においては、制約 295 によって、インタフェース 220 が、個々の ID を証明する機関が認証したクライアントサービスが機能 260 に到達できるようにしてもよい。さらに別の例においては、制約 295 によって、インタフェース 220、または、インタフェース 220 を介した目的のサービス 205 は、所定の時間枠が過ぎると、機能 260 への接続を終わらせるようにし、そうすることによって、プロセスが盗まれるのを防ぐようにしてもよい。

30

#### 【0038】

組み込むとすぐに、組み込まれた契約 235 は、ファブリックコントローラ 215 を介して、目的のサービス 205 にバインドしてもよい。組み込まれた契約 235 を目的のサービス 205 にバインドするプロセスは、確立したインタフェース 220 とロール A のインスタンス 261 を、通信チャネル 290 を介して自動的にリンクすることを含んでもよい。図 4、図 5 を参照して、下記により詳しく述べるように、通信チャネル 290 は、様々な形態のうちの任意のものを採用してよい。典型的には、通信チャネル 290 は、入力エンドポイント 255 を介して、目的のサービス 205 の機能 260 にインタフェース 220 を動作可能に結合する。入力エンドポイント 255、および／または、通信チャネル 290 は、将来参照するためにマッピングしてもよい。一例としては、ファブリックコントローラ 215 は、インタフェース 220 が使用するために、データストア 210 への適切な通信チャネル 290 の割当を担当してもよい。

40

#### 【0039】

実施形態においては、組み込まれた契約 235 を目的のサービス 205 にバインドする

50

と、インタフェース 220 は、ロール A のインスタンス 261 の入力エンドポイント 255 にリンクされる。リンクは、機能 260 を提供する目的のサービス 205 の複数のロケーションへのアクセスを促進する。言い換えれば、インタフェース 220 は、所望の機能 260 を提供するロール A の全ての関連するインスタンス 261 を認識する。

#### 【0040】

バインドのプロセスは、確立したインタフェース 220 によって実施される特定の制約 295 に基づいて、ロール A のインスタンス 261 を自動的に構成することをさらに含む。構成のプロセスは、参照番号 275 によって示す。実施形態においては、インタフェース 220 で実現された制約 295 は、目的のサービス 205 内での制約条件の設定の仕方に関して、ファブリックコントローラ 215 に指示する。一例においては、制約 295 は、ロール A のインスタンス 261 に誰がアクセスできるかに関する制約条件、例えば、北アメリカのクライアントサービスのみがアクセスできるなど、があることを指示してもよい。別の例においては、インタフェース 220 を、セキュリティを有効にしたインタフェースとして構成する制約 295 は、次に、着信コール 225 に対して、信頼性の証明書をレビューするように目的のサービス 205 を構成してよい。典型的には、データセンタ 210 内の目的のサービス 205 のインスタンスの数を増やすとき、新しいロールインスタンスに関する入力エンドポイント 255 を適切に構成するために、サービスモデル 250 は、制約 295 を備え、または制約 295 を参照してもよい。

#### 【0041】

バインドのプロセスは、機能 260 を有するロール A のインスタンス 261 の適切な入力エンドポイント 255 を識別して、それにリンクすることをさらに含む。一般的に、「入力エンドポイント」とは、広義に、ロール A がコール 225 を入力させて、他のエンティティがロール A に接触することを可能にするポートを指す。さらに、ポートを用いて、コール 225 内に組み込まれたリクエストにレスポンスしてもよい。このレスポンス、すなわち、「応答」は、同じ通信チャネル 295 を介して、機能 260 をリクエストするクライアントサービスに送り返してもよい。目的のサービス 205 およびクライアントサービスは、ネゴシエーション中に、互換性があるように構成されるので（例えば、組み込まれた契約 235 からの制約 295 を適用すること）、コール 225 と応答は、目的のサービス 205 およびクライアントサービスの両方が理解可能である（例えば、類似のプロトコルまたは言語）。

#### 【0042】

さらに、入力エンドポイント 255 にリンクすると、データセンタ 210 内の入力エンドポイント 255 のネットワークアドレス 265（例えば、IP アドレス）を、ファブリックコントローラ 215 に伝えて、インタフェース 220 にリンクされるロール A のインスタンス 261 のロケーションを探してもよい。このネットワークアドレス 265 は、組み込まれた契約 235 によって公開された機能 260 のロケーションを表しており、このネットワークアドレス 265 によって、機能 260 に依存するクライアントサービスは、適切なロケーション、またはロールインスタンス 261 にアクセスすることができる。さらに、ネットワークアドレス 265 は、データセンタ 210 外のエンティティがインタフェース 220 に接触するのを助ける。一般的に、バインド中に、インタフェース 220 をリンクさせると、ファブリックコントローラ 215 は、入力エンドポイント 255 のネットワークアドレス 265 を記載したものを、獲得し、保持する。

#### 【0043】

一例においては、このネットワークアドレス 265 は、クライアントサービスから見なくてもよい。従って、ファブリックコントローラ 215 は、コール 225 をサービスアプリケーションから適切な入力エンドポイント 255 にルーティングする静的な経路を自動的に確立する。別の例においては、このネットワークアドレス 265 は、クライアントサービスに見えてもよい。この場合には、クライアントサービスは、コール 225 を送信するために連絡先のアドレスを知ることが必要なレガシーアプリケーションでありうる。従って、ファブリックコントローラ 215 は、ネットワークアドレス 265 をクライアン

トサービスに公開してもよい。さらに別の例においては、このネットワークアドレス 265 は、クライアントサービスがアクセスできてよい。従って、通信チャネル 290 が動的に更新されると、クライアントサービスは、ネットワークアドレス 265 を検索して、入力エンドポイントにアクセスしてもよい。

#### 【0044】

目的のサービス 205 でコール 225 を受信すると、コール 225 を行ったクライアントサービスの ID を要求して、コール 225 の信頼性を検証してもよい。一例においては、クライアントサービスの ID が、ファブリックコントローラ 215 によって記録される。記録は、クライアントサービスをデプロイすると行われてもよいし、クライアントサービスを、組み込まれた契約 235 にバインドすると行われてもよいし、あるいは、その後の任意の時に行われてもよい。発せられコール 225 をロール A のインスタンス 261 にリレーすると、要求 281 を、発せられたコール 225 に追加してもよい。この要求 281 は、クライアントサービスの ID にアクセスして、クライアントサービスのコンポーネントプログラムがコール 225 を発したことを検証し、判断した ID とクライアントサービスの他の特性を要求 281 に組み込むことによって、生成してもよい。

10

#### 【0045】

このようにして、ファブリックコントローラ 215 は、コール 225 の発信源を基本的に保証し、クライアントサービスのために認証を与える。従って、要求 281 によって、目的のサービス 205 は、コールを発した側を検証することができ、そうすることによって、目的のサービス 205 で、個々のセキュリティレベルを保証することができる。実施形態においては、検証は、要求 281 を検査して、コール 225 のリクエストを引き受けるか否かを確定することを含んでもよい。検査は、要求 281 の内容（例えば、クライアントサービスのプロパティおよび／または能力）のチェックを含んでもよい。この内容の詳細のレベルは、一般的に、要求 281 の粒度、コール 225 を送信するクライアントサービスの種類、および／または、クライアントサービスの出力エンドポイントがサポートするプロトコルに依存する。

20

#### 【0046】

一実施形態においては、要求 281 内のクライアントサービスのプロパティは、ファブリックコントローラ 215 がクライアント装置に関して推定することができる任意の情報を含んでもよい。一例においては、クライアントサービスのジオロケーションを要求 281 の内容に設けてもよい。それに応じて、目的のサービス 205 は、より近くのノードでコール 225 を引き受ける、または、より近くのノードにコール 225 をリダイレクトしてもよい。または、目的のサービス 205 は、ジオロケーションに基づいて、コール 225 への応答を調整してもよい。例えば、ジオロケーションが、コール 225 はフランスからであることを示す場合、目的のサービス 205 は、フランス語で応答を準備してもよい。別の例においては、機能 260 への権利のリストを、要求 281 の内容に組み込んでもよい。それに応じて、目的のサービス 205 は、クライアントサービスが所有する権利に従って、目的のサービス 205 が制御するリソースにクライアントサービスがアクセスするのを制限してもよい。

30

#### 【0047】

別の実施形態においては、目的のサービス 205 は、検証 API (Application Programming Interface) 201 に問い合わせることによって、クライアントサービスの ID と権利を検証してもよい。検証 API 201 は、ファブリックコントローラ 215 がコール 225 のソースを知っているので、受信したコール 225 に関するデータを提供してもよい。従って、要求 281 が不完全または利用できない場合、目的のサービス 205 は、コール 225 を満たすか否か（例えば、機能 260 を提供）の判断を積極的に行ってもよい。

40

#### 【0048】

次に図 3 を見てみると、本発明の実施形態の実装に使用するのに適した、目的のサービスにバインドされた契約を、クライアントサービスにバインドするように構成された例示

50

的な分散コンピューティング環境 200 を示すブロック図が示されている。最初に、分散コンピューティング環境 200 は、上述のように、目的のサービスにアクセスするクライアントサービス 305 を含む。クライアントサービス 305 は、データセンタ 210 内で実行する、またはデータセンタ 210 外で遠隔接続によって実行する、あるいはデータセンタ 210 上に部分的に常駐するように構成した任意のサービスアプリケーションを表してもよい。クライアントサービス 305 は、データセンタ 210 の別々のノード（例えば、ノード IV 224 および V 225）に分散され得るコンポーネントプログラム（例えば、コンポーネントプログラム A 361 および B 362）を含んでもよい。クライアントサービス 305 がデータセンタ 210 内にある実施形態においては、ファブリックコントローラ 215 は、サービスモデル 350 に保持されたデプロイメントの詳細を踏まえて、コンポーネントプログラム A 361 および B 362 をデプロイしてもよく、クライアントサービス 305 の実行を管理してもよい。

10

#### 【0049】

例示の実施形態においては、1つまたは複数のコンポーネントプログラム A 361 および B 362 は、依存性 360 と共に、開発者によって書かれる。一般的に、クライアントサービス 305 の適切な実行は、適切な機能（例えば、図 2 の目的のサービス 205 の機能 260）を用いて、依存性 360 を満たすことに依存する。動作時、クライアントサービス 305 は、指示を伝えて、クライアントサービス 305 の依存性 360 を満たしてもよい。それに応じて、ファブリックコントローラ 215 は、依存性 360 を検査して、依存性 360 を満たす機能を抽象化したものを公開する契約を割り当ててもよい。上述のように、依存性 360 を満たす契約は、分散コンピューティング環境 200 内で前もって組み込まれてもよい。さらに、組み込まれた契約は、契約が公開する機能を実行する目的のサービスに、前もってバインドしてもよい。

20

#### 【0050】

依存性を満たす契約を割り当てると、ファブリックコントローラ 215 は、この割り当てられ、前もってバインドされた契約 335 をクライアントサービス 305 にバインドしてもよい。実施形態においては、クライアントサービス 305 をバインドするプロセスは、動作を開始するためのクライアントサービス 305 のコンポーネントプログラム A 361 および B 362 の最初のデプロイメント中であってもよい。一般的に、デプロイメントのプロセスは、コンポーネントプログラム A 361 および B 362 を、割り当てられバインドされた契約 335 によって定義されたインタフェース 220 に自動的にリンクさせることを含む。そこで、バインドされた契約 335 は、プロパティ 240 のセットから得た制約 295 で、インタフェース 220 を構成する。一実施形態においては、割り当てられ、バインドされた契約 335 を組み込むと、インタフェース 220 が、分散コンピューティング環境 200 内に確立される。さらに、リンクの詳細が、ファブリックコントローラ 215 に書き込まれてもよい。あるいは、リンクは、将来参照するために、ファブリックコントローラ 215 にアクセス可能な任意のデータストア（単数または複数）に、少なくとも一時的に記憶されてもよい。

30

#### 【0051】

例示的な実施形態において、コンポーネントプログラム A 361 および B 362 をインタフェース 220 に自動的にリンクするプロセスは、コンポーネントプログラム B 362 に配置された出力エンドポイント 375 を識別することを含んでもよく、ここではコンポーネントプログラム B 362 は、依存性 360 を呈する。一般的には、出力エンドポイント 375 は、コンポーネントプログラム B 362 が、他方に何かをリクエストするのに使うポートを表してよい。自動的にリンクを行うプロセスは、分散コンピューティング環境 200 内の通信チャネル 390 の割当を継続して、確立したインタフェース 220 を出力エンドポイント 375 に動作可能に結合してもよい。通信チャネル 390 は、典型的には、クライアントサービス 305 の出力エンドポイント 375 から発せられたコール 225 を伝達する役割をする。典型的には、コール 225 は、コンポーネントプログラム B 362 に書かれた依存性 360 を満たすようにというコンポーネントプログラム B 362 によ

40

50

るリクエストを含む。実施形態においては、依存性 360 は、クライアントサービス 305 では実行されないが、インタフェース 220 を通してリンクされた機能によって行われる外部処理またはデータ検索を含んでもよい。

#### 【0052】

バインドのプロセスを完了すると、バインドされた契約 335 は、クライアントサービス 305 と、付随する目的のサービスの両方にバインドされる。実施形態においては、クライアントサービス 305 は、インタフェース 220 の制約 295 に問い合わせ、インタフェース 220 がクライアントサービス 305 のサービスモデル 350 が特定する機能面を有することができるか否かを判定することができる。もし有していなければ、クライアントサービス 305 は、ファブリックコントローラ 215 によって、バインドされたインタフェースと目的のサービスに代わる、依存性 360 を満たす機能を有する別の契約に、再バインドされてよい。バインドされた目的のサービスがオフラインとなったときも、再バインドを行ってもよい。

10

#### 【0053】

クライアントサービス 305 の依存性 360 に目的のサービスを関連付けるために、様々な種類の契約を採用してもよい。一実施形態によると、セルフバインドされた契約を利用する。一般的には、セルフバインドされた契約は、ファブリックコントローラ 215 によって動作可能なプラグイン機構によって自動的にバインドされる。従って、ファブリックコントローラ 215 が目的のサービスまたは疑似サービスを選択すると、そのサービスが、インタフェース 220 を介して行われたコール 225 に対応する。

20

#### 【0054】

別の実施形態においては、標準的な契約が用いられる。一般的に、標準的な契約は、2つの異なった方法でバインドしてもよい。1つの例示的方法においては、目的のサービスは、それぞれ固有の名前を与えられる。そして、ファブリックコントローラ 215 は、バインドされた目的のサービスが実際にバインドされた契約 335 を組み込むことを検証することによって、固有の名前を用いて、クライアントサービス 305 と目的のサービスとの関連付けの有効性をチェックしてもよい。そして、ネットワークアドレス（例えば、ネットワークアドレス 265）を、目的のサービスの入力エンドポイントから得る。別の方法においては、データセンタ 210 によってホストされていない外部クライアントサービス 305 の出力エンドポイント 375 に関する情報、および／または、目的のサービスの入力エンドポイント（例えば、IP アドレス／DNS 名：ポート）に関する情報が、ファブリックコントローラ 215 に送られる。IP：ポートの詳細が、インタフェース 220 のために検知される。従って、ファブリックコントローラ 215 は、インタフェース 220 にリンクされたコンポーネントプログラム B 362 の出力エンドポイント 375 を構成する。しかしながら、名前を付けられた目的のサービスがバインドされた契約 335 を満たすことの検証は行われない。

30

#### 【0055】

さらに別の例示的实施形態においては、典型的には、クライアントサービス 305 が、データセンタ 210 の外部に常駐するとき、外部の契約を利用する。一般的に、外部の契約は、低レベルの抽象化したものを含み、その抽象化したものによって、クライアントサービス 305 は、クライアントサービス 305 のデプロイメントによって設定された境界内の任意の公的な IP アドレスに接触することができる。実際に、バインドは行われず、クライアントサービス 305 は、目的のサービスのネットワークアドレス 265 を提供して、その機能にアクセスするものとされる。従って、リンクされた入力エンドポイントのネットワークアドレス 265 を用いて、通信チャネル 390 を構成し、ルーティングする。

40

#### 【0056】

上述のように、バインドされた契約 335 は、確立したインタフェース 220 に関連付けられたプロパティ 240 のセットを保持してもよい。動作時、関連付けられたプロパティ 240 のセットから得た値で、確立したインタフェース 220 のパラメータを決定する

50

ことによって、制約 295 を確立したインタフェース 220 に適用してもよい。この適用された制約 295 は、コンポーネントプログラム A361 および B362 を構成するために、クライアントサービス 305 に公開してもよい。クライアントサービス 305 を構成するプロセスは、参照番号 388 によって示される。一般的には、出力エンドポイント 375 と共に、コンポーネントプログラム A361 および B362 を構成するプロセス 388 は、インタフェース 220 に適用された制約 295 からの指示を引き出すことを含む。これらの指示は、クライアントサービス 305 の態様への任意の数の構成、および、それによって提供される通信のために用いてもよい。例えば、指示は、出力エンドポイント 375 が発したコール 225 をフォーマットするために利用してもよい。特に、指示を利用してコール 225 のフォーマットを構成することによって、コール 225 は、所望の機能を実施する目的のプログラムのロールインスタンスの基礎となるプロトコルと互換性があるとしてもよい。

10

**【0057】**

一旦構成されると、クライアントサービス 305 は、依存性 360 が満たされれば、コール 225 を発してもよい。実施形態においては、コール 225 は、依存性 360 を実現するコンポーネントプログラム B362 の出力エンドポイント 375 から発してもよい。その後、コール 225 は、分散コンピューティング環境 200 の割り当てられた通信チャネル 390 を介して、確立したインタフェース 220 にルーティングされる。詳細に前述したように、確立したインタフェース 220 は、パラメータを決定されて、発せられたコール 225 を目的のサービスにリレーし、あるいは発せられたコール 225 をフィルタリングしてもよい。インタフェース 220 のこの決定は、クライアントサービス 305 の ID と共に、適用されている制約 295 に基づいてもよい。

20

**【0058】**

この例示的分散コンピューティング環境 200 は、本発明の態様を実行するために実装してもよい適切な環境の一例に過ぎず、本発明の使用または機能の範囲に関して、いかなる制限も示唆するものではない。分散コンピューティング環境 200 の図示された例示的システムアーキテクチャが、図示のコンポーネント 215, 220, 221, 225, 305, 335, 350, 360, 361 および 362 の任意の 1 つまたは任意の組み合わせに関する依存性または要件を有すると解釈するべきでもない。一部の実施形態においては、コンポーネント 215, 220, 221, 224, 225, 305, 335, 350, 360, 361 および 362 のうちの 1 つまたは複数は、スタンドアロン装置として実装してもよい。別の実施形態においては、コンポーネント 215, 220, 221, 225, 305, 335, 350, 360, 361 および 362 の 1 つまたは複数は、データセンタ 210 またはファブリックコントローラ 215 に、直接、組み込まれてもよい。当業者には当然のことながら、図 3 に示されたコンポーネント 215, 220, 221, 225, 305, 335, 350, 360, 361 および 362 は、性質および数の点で例示的なものであり、制限的なものと解釈してはならない。

30

**【0059】**

従って、任意の数のコンポーネントを採用して、本発明の実施形態の範囲内で所望の機能を達成してもよい。図 3 の様々なコンポーネントは、分かりやすいように線で示されているが、実際は、様々なコンポーネントの線引きは明確なものではなく、比喩的であり、線はより正確には、中間的な、あいまいなものである。さらに、図 3 の一部のコンポーネントは、単一のブロックで表されているが、表示は、性質および数の点で例示的なものであり、制限的なものとみなしてはならない（例えば、クライアントサービス 305 のみが図示されているが、より多くのものがインタフェース 220 に通信可能に結合されてもよい）。

40

**【0060】**

次に図 4 を見てみると、本発明の実施形態による、ロードバランシング (LB) チャネル 410 を利用して、サービスアプリケーション（例えば、クライアントサービス 305 および目的のサービス 205）間の通信（例えば、コール 225 と応答）をルーティング

50

する例示的なファブリックコントローラ 215 の図式的な表示が示されている。最初に、ファブリックコントローラ 215 がアクセス可能な接続定義 450 を与える。これらの接続定義 450 は、LB 機構 420 に指示して、目的のサービス 205 の複数のロールインスタンス 411, 412 および 413 のうちの選択された 1 つに通信をルーティングするのを助ける。コールを受信するために選択されるロールインスタンスは、例えば、コール 225 を発したクライアントサービス 305 のロール（例えば、ロール 421, 422 および 423）との互換性、そのロール（例えば、ロール 421, 422 および 423）への近接性、利用可能性など、任意の数の要素に基づいて選択してもよい。

#### 【0061】

選択されると、確立したインタフェース 220 を目的のサービス 205 のロール（例えば、ロール 411, 412 および 413）の入力エンドポイントにリンクするロードベアリング（LB）チャンネル 410 を介して、コールは、目的のサービス 205 の選択されたロールに送信される。一例においては、送信は、確立したインタフェース 220 で、クライアントサービス 305 からのコール 225 を受信すること、および、LB 機構 420 を呼び出して、LB チャンネル 410 の利用可能な通信チャンネルにコール 225 を配信することを含んでもよい。従って、ただ 1 つのネットワークアドレスが、コール 225 を送信するクライアントサービス 305 に提供される。ファブリックコントローラ 215 は、接続定義 450 を踏まえて、ロードバランシングスキームの実行を担当し、それによって、インタフェース 220 へのコール 225 の配信が、目的のサービス 205 のロール（例えば、ロール 411, 412 および 413）間に間違いなく配信される。例示的な実施形態においては、ネットワークアドレスは、インタフェース 220 および／または LB 機構 420 への仮想 IP である。LB 機構 420 は、次に、仮想 IP を、それぞれが異なるロールに関連付けられる具体的な IP に変えてもよい。

#### 【0062】

図 5 を参照すると、本発明の実施形態による、SLS チャンネル（例えば、チャンネル 510, 511 および 512）を利用して、サービスアプリケーション（例えば、クライアントサービス 305 および目的のサービス 205）間の通信（例えば、コール 521, 522 および 523 と、そのコールに応じる応答）をルーティングする例示的なファブリックコントローラ の図式的な表示が示されている。一般的に、割り当てられた通信チャンネル（図 2 の参照番号 290 を参照）は、目的のサービス 205 のロール（例えば、ロール 411, 412 および 413）の入力エンドポイントに、確立したインタフェース 220 をリンクする、SLS チャンネル 510, 511 および 512 を含んでもよい。これらの SLS チャンネル 510, 511 および 512 は、接続定義 450 によって記憶され、またファブリックコントローラ 215 によって保持されてもよい。

#### 【0063】

動作時、コール（例えば、コール 521, 522 および 523）を受信すると、ファブリックコントローラ 215 は、コールに関連付けられたネットワークアドレスを識別する。そのネットワークアドレスは、コール（例えば、コール 521）を提供するクライアントサービス 305 のロール（例えば、ロール 421）が提供してもよく、あるいは、コールの発信源に基づいて接続定義 450 が供給してもよい。このネットワークアドレスに基づいて、コール 225 は、SLS チャンネルのうち、目的のサービス 205 の適切なロールインスタンス（例えば、ロール 411）の入力エンドポイントに、確立したインタフェース 220 をリンクするように指定された通信チャンネル（例えば、510）を介して、ルーティングされる。従って、ファブリックコントローラ 215 は、インタフェース 220 にリンクされたクライアントサービスのコンポーネントプログラムまたはロールと同じ数の、到達可能であり外部からアドレス可能な入力エンドポイントがあることを保証する。このように、各出力エンドポイントが単一の入力エンドポイントに対応することによって、コールをルーティングするための単一の SLS チャンネルおよび単一のネットワークアドレスを指定する。

#### 【0064】

図6を参照すると、本発明の実施形態による、契約を自動的に組み込み、その組み込まれた契約を目的のサービスにバインドする方法600全体のフロー図が示されている。さらに、「ステップ」および／または「ブロック」という語は、本明細書においては、採用される方法の異なる要素を示すのに用いられるが、個々のステップの順序が明確に述べられている場合を除いては、これらの語は、本明細書で開示されている様々なステップ間の個々の順序を示唆していると解釈してはならない。最初に、ブロック605に示されているように、インタフェースの集合を定義する契約を識別する。さらに、この契約は、各インタフェースをインストールするためのプロパティのセットを保持する。ブロック610に示すように、識別された契約を組み込み、分散コンピューティング環境内のインタフェースの集合から、インタフェースを確立する。典型的には、組み込まれた契約は、クライアントサービスが目的のサービスの一部に到達するための媒介を導入する。ブロック615に示すように、組み込まれた契約は、確立したインタフェースに関連付けられたプロパティのセットから得た値で、確立したインタフェースのパラメータを決定することによって、目的のサービスにバインドされる。実施形態において、パラメータを決定するのプロセスは、確立したインタフェースと、その1つまたは複数のロールインスタンスとを自動的にリンクすること（ブロック620参照）と、目的のサービス（ブロック625参照）の実行の管理を担当するファブリックコントローラを介してリンクをマッピングすることを含む。明確化のために、ロールインスタンスは、一種のコンポーネントプログラムを表す少なくとも1つのロールを複製し、このコンポーネントプログラムは、実行されると、機能を目的のサービスに与える。

#### 【0065】

次に図7を見てみると、本発明の実施形態による、組み込まれた契約を、クライアントサービスの依存性に基づいて自動的に割り当て、割り当てられた契約をバインドする方法700全体を示すフロー図が示されている。最初に、ブロック705に示されるように、依存性を満たす指示を、クライアントサービスから受信する。上述のように、クライアントサービスは、1つまたは複数のコンポーネントプログラムを含む。ブロック710に示すように、クライアントサービスのコンポーネントプログラムの依存性を満たす機能を抽象化したものを公開する契約が割り当てられる。典型的には、契約は、分散コンピューティング環境内で組み込まれ、機能を実行する目的のサービスにバインドされる。ブロック715に示すように、クライアントサービスがデプロイされて、その動作を開始する。実施形態においては、デプロイメントは、割り当てられ、バインドされた契約（ブロック720参照）が定義するインタフェースに、コンポーネントプログラムを自動的にリンクすることと、目的のサービスの実行の管理を担当するファブリックコントローラにリンクの詳細を書くこと（ブロック725参照）とを含む。一般的に、割り当てられ、バインドされた契約を組み込むと、インタフェースが、分散コンピューティング環境内に確立される。

#### 【0066】

当業者には当然のことながら、図6および図7に示される実施形態の範囲内で、所望の機能を達成するために、任意の数のステップを採用してもよい。さらに、図6および図7の様々なステップは、分かりやすいように線で示しているが、実際は、様々なコンポーネントの線引きは、それほど明確でなく、比喩的であり、線は、より正確には、中間的なあいまいなものである。さらに、図6および図7の一部のステップは、単一のプロセスとして表しているが、表示は、性質および数の点で例示的なものであり、制限的であると解釈してはならない。

#### 【0067】

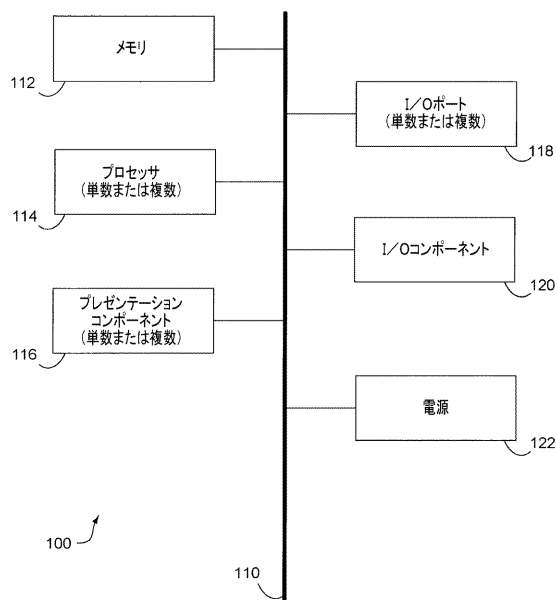
本発明の実施形態を、個々の実施形態に関連して記載したが、それらは、あらゆる点において、説明するためであり、限定するためではない。当業者には、本発明の範囲を逸脱することなく本発明の実施形態に関連する代替的な実施形態は明らかであろう。

#### 【0068】

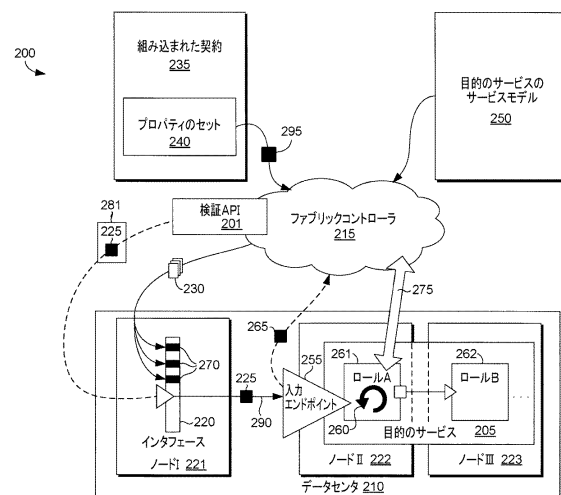
本発明は、そのシステムおよび方法に本来備わる、明らかな他の利点と共に、上述のあ

らゆる目的を達成するようになされたものであることは、上記より明らかである。当然のことながら、一部の特徴および下位の組み合わせは、有用であり、それらに言及することなく採用してもよい。これは、請求項の企図するところであり、また請求項の範囲内である。

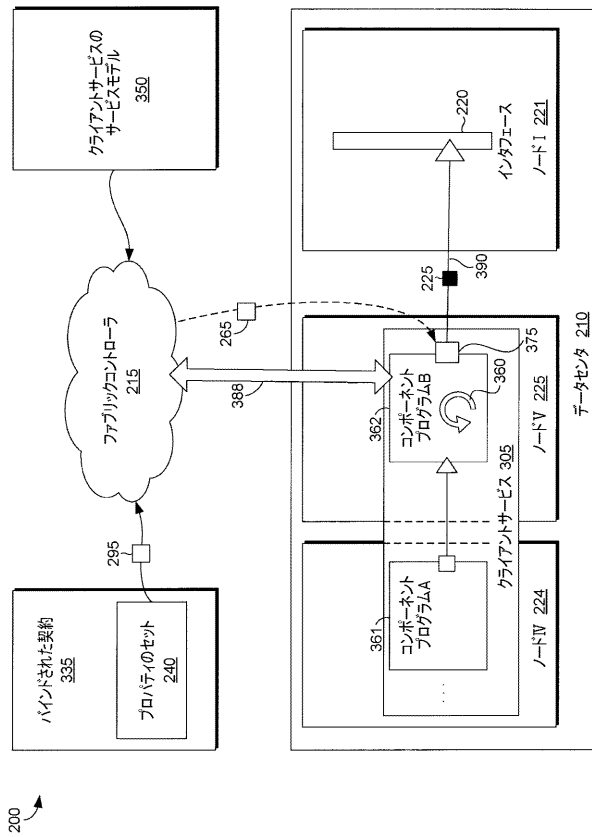
【図 1】



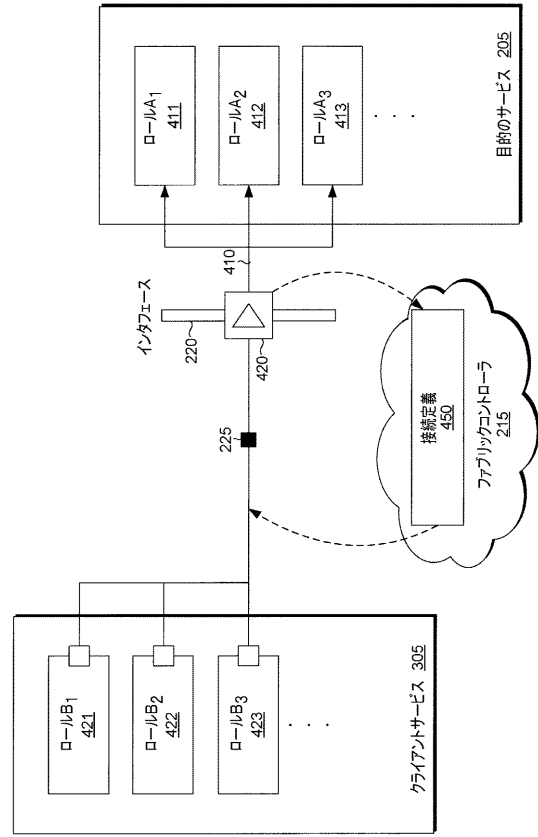
【図 2】



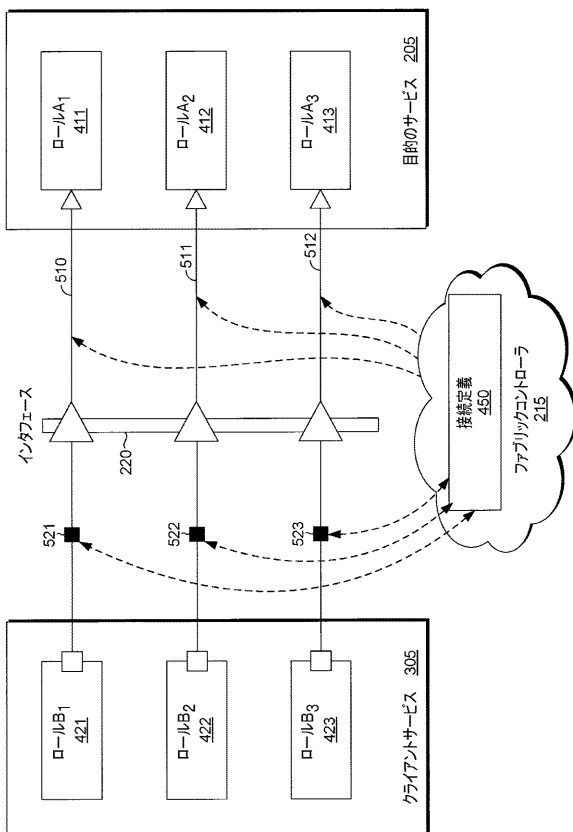
【図 3】



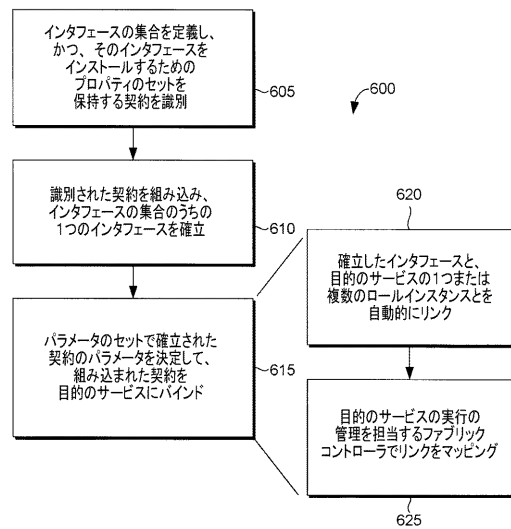
【図 4】



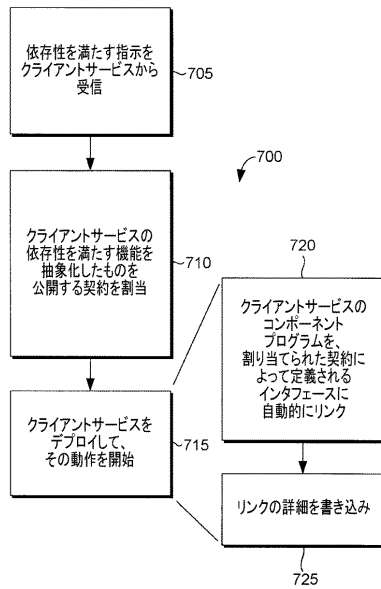
【図 5】



【図 6】



【図 7】



---

フロントページの続き

- (72)発明者 ホセ ベルナベウーオーバン  
アメリカ合衆国 98052 ワシントン州 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 コーポレーション エルシーエーインターナショナル パテント内
- (72)発明者 ユーセフ エー. カリディ  
アメリカ合衆国 98052 ワシントン州 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 コーポレーション エルシーエーインターナショナル パテント内

審査官 漆原 孝治

- (56)参考文献 特表2006-526189 (JP, A)  
特開2006-252537 (JP, A)

- (58)調査した分野(Int.Cl., DB名)  
G06F 15/00  
G06F 13/00