(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6386074号
(P6386074)
(45)発行日　平成30年9月5日(2018.9.5)　(24)登録日　平成30年8月17日(2018.8.17)

(51)Int.Cl.　　F I
G06F 13/00 (2006.01)　　G06F 13/00 351A
H04M 11/00 (2006.01)　　G06F 13/00 ZIT
　　H04M 11/00 302

請求項の数 19　(全 62 頁)

(21)出願番号　特願2016-558396 (P2016-558396)
(86)(22)出願日　平成27年3月20日(2015.3.20)
(65)公表番号　特表2017-516193 (P2017-516193A)
(43)公表日　平成29年6月15日(2017.6.15)
(86)国際出願番号　PCT/US2015/021867
(87)国際公開番号　WO2015/143396
(87)国際公開日　平成27年9月24日(2015.9.24)
審査請求日　平成30年1月25日(2018.1.25)
(31)優先権主張番号　14/222,100
(32)優先日　平成26年3月21日(2014.3.21)
(33)優先権主張国　米国(US)
(31)優先権主張番号　14/222,083
(32)優先日　平成26年3月21日(2014.3.21)
(33)優先権主張国　米国(US)

(73)特許権者　500171497
ピーティーシー　インコーポレイテッド
アメリカ合衆国マサチューセッツ州０２４９４・ニーダム・ケンドリックストリート１４０
(74)代理人　100086771
弁理士　西島　孝喜
(74)代理人　100088694
弁理士　弟子丸　健
(74)代理人　100094569
弁理士　田中　伸一郎
(74)代理人　100067013
弁理士　大塚　文昭
(74)代理人　100109070
弁理士　須田　洋之

最終頁に続く

(54)【発明の名称】バイナリ動的ＲＥＳＴメッセージを使用したシステム及び方法

(57)【特許請求の範囲】
【請求項１】
複数のデータストリームを受信するように構成された通信ポートであって、それぞれのデータストリームは、前記データストリームが、単一のシリアル化された動的ＲＥＳＴ（Representative State Transfer）メッセージを集合的に形成する複数のデータストリームの内の１つであることを指示するフィールドを含むヘッダを有し、前記単一のシリアル化された動的ＲＥＳＴメッセージは、複数の定義部分及び複数のデータ部分を含む自己記述型コンポーネントを含み、前記複数のデータ部分のそれぞれは前記複数の定義部分のそれぞれに対応するデータ値を含み、前記複数の定義部分のそれぞれは、各データ部分の、名前、属性、及びプロパティからなるグループから選択されるデータ要素を記述するデータ構造を含むものである通信ポートと、
前記通信ポートに結合したプロセッサと、
命令を記憶したメモリと、を含む装置であって、前記命令は前記プロセッサによって実行されたときに、前記プロセッサに、
前記メモリに第１のバッファ及び第２のバッファを確立すること、
前記通信ポートを介して第１のデータストリームを受信すると、前記第１のデータストリームの所定のヘッダを調べること、
マルチパートフィールドにおいて、前記所定のヘッダが、前記データストリームがマルチパートシリアル化動的ＲＥＳＴメッセージであることを指示する値を有していることに応答して、第１の復号化された部分を生成するために、前記第１バッファにバッファリン

グされた前記第１データストリームのペイロードを復号化し、前記第１の復号化された部分を前記第２のバッファに記憶し、ここで前記第１データストリームの前記ペイロードの復号化は第２のデータストリームの完全な受信の前に開始され、前記第１データストリーム及び前記第２データストリームは、全体的に、又は部分的に、前記単一のシリアル化された動的ＲＥＳＴメッセージを集合的に含むものであること、
を実行させる、装置。
【請求項２】
前記命令は、前記プロセッサによって実行されたときに、前記プロセッサに、
現在のマルチパート識別子フィールド、マルチパート数フィールド、及びサイズフィールドにおいて、前記第１のデータストリームを調べ、ここで、前記マルチパート数フィールドで調べられた第１のパラメータ及び前記サイズフィールドで調べられた第２のパラメータは、前記第２のバッファのためのバッファサイズを決定するために使用されるものであること、
を実行させる、請求項１に記載の装置。
【請求項３】
前記命令は、前記プロセッサによって実行されたときに、前記プロセッサに、
現在のマルチパート識別子フィールド、マルチパート数フィールド、及びサイズフィールドにおいて、前記第１のデータストリームを調べ、ここで、前記現在のマルチパート識別子フィールドで調べられた第２のパラメータは、前記第２のバッファに前記第１の復号化された部分を記憶させるための位置を決定するために使用されるものであること、
を実行させる、請求項１に記載の装置。
【請求項４】
前記第１のデータストリームの前記ペイロードを復号化し、前記第１の復号化された部分を生成する復号化エンジンを含む、請求項１に記載の装置。
【請求項５】
前記データ構造は、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含む、請求項１に記載の装置。
【請求項６】
前記マルチパート数フィールドに関連する前記第１のパラメータは、前記単一のシリアル化された動的ＲＥＳＴメッセージを形成する前記複数のデータストリームの数を定め、前記単一のシリアル化された動的ＲＥＳＴメッセージは完全なメッセージを含む、請求項２に記載の装置。
【請求項７】
前記現在のマルチパート識別子フィールドに関連する前記第２のパラメータは、前記単一のシリアル化された動的ＲＥＳＴメッセージ内の所定の復号化されたデータストリームの位置を指示するメッセージ数識別子を定め、ここで、単一のシリアル化された動的ＲＥＳＴメッセージは完全なメッセージを含むものである、請求項３に記載の装置。
【請求項８】
前記通信ポートは、ウェブソケット接続を通じて１つ以上のデータストリームを受信する、請求項１に記載の装置。
【請求項９】
前記現在のマルチパート識別子フィールド、前記マルチパート数フィールド、及び前記サイズフィールドは、前記第１のデータストリームの前記ヘッダ内に位置している、請求項２に記載の装置。
【請求項１０】
前記現在のマルチパート識別子フィールド、前記マルチパート数フィールド、及び前記サイズフィールドは、前記第１のデータストリームの前記ペイロード内に位置している、請求項２に記載の装置。
【請求項１１】
コンピューティングデバイスのポートにおいて、複数のデータストリームを受信するス

テップであって、それぞれのデータストリームは、前記データストリームが、単一のシリアル化された動的ＲＥＳＴ（Representative State Transfer）メッセージを集合的に形成する複数のデータストリームの内の１つであることを指示するフィールドを含むヘッダを有し、前記単一のシリアル化された動的ＲＥＳＴメッセージは、複数の定義部分及び複数のデータ部分を含む自己記述型コンポーネントを含み、前記複数のデータ部分のそれぞれは前記複数の定義部分のそれぞれに対応するデータ値を含み、前記複数の定義部分のそれぞれは、各データ部分の、名前、属性、及びプロパティからなるグループから選択されるデータ要素を記述するデータ構造を含むものであるステップと、
前記コンピューティングデバイスのメモリに第１のバッファ及び第２のバッファを確立するステップと、
前記ポートを介して第１のデータストリームを受信すると、前記第１のデータストリームの所定のヘッダを調べるステップと、
マルチパートフィールドにおいて、前記所定のヘッダが、前記データストリームがマルチパートシリアル化動的ＲＥＳＴメッセージであることを指示する値を有していることに応答して、第１の復号化された部分を生成するために、前記第１バッファにバッファリングされた前記第１データストリームのペイロードをプロセッサを使用して復号化し、前記第１の復号化された部分を前記第２のバッファに記憶するステップであって、ここで前記第１データストリームの復号化は第２のデータストリームの完全な受信の前に開始され、前記第１データストリーム及び前記第２データストリームは、全体的に、又は部分的に、前記単一のシリアル化された動的ＲＥＳＴメッセージを集合的に含むものであるステップと、
を含むコンピュータで実施される方法。
【請求項１２】
現在のマルチパート識別子フィールド、マルチパート数フィールド、及びサイズフィールドにおいて、前記第１のデータストリームを調べるステップであって、ここで、前記マルチパート数フィールドで調べられた第１のパラメータ及び前記サイズフィールドで調べられた第２のパラメータは、前記第２のバッファのためのバッファサイズを決定するために使用されるものであるステップ、
を含む、請求項１１に記載のコンピュータで実施される方法。
【請求項１３】
現在のマルチパート識別子フィールド、マルチパート数フィールド、及びサイズフィールドにおいて、前記第１のデータストリームを調べるステップであって、ここで、前記現在のマルチパート識別子フィールドで調べられた第２のパラメータは、前記第２のバッファに前記第１の復号化された部分を記憶させるための位置を決定するために使用されるものであるステップ、
を含む、請求項１１に記載のコンピュータで実施される方法。
【請求項１４】
前記ポートは、ウェブソケット接続を通じて前記複数のデータストリームを受信する、請求項１１に記載のコンピュータで実施される方法。
【請求項１５】
前記ペイロードの復号化は、復号化エンジンによって実行される、請求項１１に記載のコンピュータで実施される方法。
【請求項１６】
前記データ構造は、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含む、請求項１１に記載のコンピュータで実施される方法。
【請求項１７】
前記マルチパート数フィールドに関連する前記第１のパラメータは、所定の完全な実行可能メッセージを形成する１つ以上のデータストリームの数を定め、前記単一のシリアル化された動的ＲＥＳＴメッセージは完全なメッセージを含む、請求項１２に記載のコンピ

ュータで実施される方法。
【請求項１８】
前記現在のマルチパート識別子フィールドに関連する前記第２のパラメータは、前記単一のシリアル化された動的ＲＥＳＴメッセージ内の所定の復号化されたデータストリームの位置を指示するメッセージ数識別子を定め、ここで、単一のシリアル化された動的ＲＥＳＴメッセージは完全なメッセージを含むものである、請求項１７に記載のコンピュータで実施される方法。
【請求項１９】
命令を記憶した非一時的コンピュータ読取可能媒体であって、前記命令は、コンピューティングデバイスのプロセッサによって実行されたときに、前記プロセッサに、
前記コンピューティングデバイスのポートにおいて、複数のデータストリームを受信するステップであって、それぞれのデータストリームは、前記データストリームが、単一のシリアル化された動的ＲＥＳＴ（Representative State Transfer）メッセージを集合的に形成する複数のデータストリームの内の１つであることを指示するフィールドを含むヘッダを有し、前記単一のシリアル化された動的ＲＥＳＴメッセージは、複数の定義部分及び複数のデータ部分を含む自己記述型コンポーネントを含み、前記複数のデータ部分のそれぞれは前記複数の定義部分のそれぞれに対応するデータ値を含み、前記複数の定義部分のそれぞれは、各データ部分の、名前、属性、及びプロパティからなるグループから選択されるデータ要素を記述するデータ構造を含むものであるステップと、
前記コンピューティングデバイスの第１のバッファ及び第２のバッファを確立するステップと、
前記ポートを介して第１のデータストリームを受信すると、前記第１のデータストリームの所定のヘッダを調べるステップと、
マルチパートフィールドにおいて、前記所定のヘッダが、前記データストリームがマルチパートシリアル化動的ＲＥＳＴメッセージであることを指示する値を有していることに応答して、第１の復号化された部分を生成するために、前記第１バッファにバッファリングされた前記第１データストリームのペイロードを使用して復号化し、前記第１の復号化された部分を前記第２のバッファに記憶するステップであって、ここで前記第１データストリームの復号化は第２のデータストリームの完全な受信の前に開始され、前記第１データストリーム及び前記第２データストリームは、全体的に、又は部分的に、前記単一のシリアル化された動的ＲＥＳＴメッセージを集合的に含むものであるステップと、
を実行させるものである、非一時的コンピュータ読取可能媒体。
【発明の詳細な説明】
【技術分野】
【０００１】
（関連出願の相互参照）
本出願は、２０１４年3月２１日に出願された米国出願番号第１４／２２２，１００号、題名「Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｏｆ Ｕｓｉｎｇ Ｂｉｎａｒｙ Ｄｙｎａｍｉｃ Ｒｅｓｔ Ｍｅｓｓａｇｅｓ」と、２０１４年3月２１日に出願された同第１４／２２２，０８３号、題名「Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｏｆ Ｕｓｉｎｇ Ｄｙｎａｍｉｃ Ｒｅｓｔ Ｍｅｓｓａｇｅｓ ｗｉｔｈ Ｗｅｂ ｓｏｃｋｅｔｓ」と、２０１４年3月２１日に出願された同第１４／２２２，０６７号、題名「Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｏｆ Ａｂｓｔａｃｔｉｎｇ Ｃｏｍｍｕｎｉｃａｔｉｏｎｇ Ｐｒｏｔｏｃｏｌ Ｕｓｉｎｇ Ｓｅｌｆ－Ｄｅｓｃｒｉｂｉｎｇ Ｍｅｓｓａｇｅｓ」と、２０１４年3月２１日に出願された同第１４／２２２，０２７号、題名「Ｃｈｕｎｋ－ｂａｓｅｄ Ｃｏｍｍｕｎｉｃａｔｉｏｎ ｏｆ Ｂｉｎａｒｙ Ｄｙｎａｍｉｃ Ｒｅｓｔ Ｍｅｓｓａｇｅｓ」と、の優先権を主張する。これらの出願のそれぞれの内容は、その全体が本明細書に参考として組み込まれる。
【０００２】
（発明の分野）

本発明は概ね、マシンツーマシン通信に関する。より具体的には、特定の実施形態において、本発明は、デバイス間で通信するためのバイナリ動的Ｒｅｐｒｅｓｅｎｔａｔｉｏｎａｌ　Ｓｔａｔｅ　Ｔｒａｎｓｆｅｒ（ＲＥＳＴ）メッセージの使用に関する。

【背景技術】

【０００３】

モノのインターネット（「ＩＯＴ」）とは、実世界に存在するシステム、デバイス、及び／又は物体のネットワーク（集合的に「システムと呼ぶ」）を指し、自動車から心臓モニタまでの範囲に及ぶ。これらの物理的物体及び／又はデバイスは、とりわけデータ通信及びデータ交換を可能にする、ソフトウェア、センサ、電子機器、通信手段などを備えている、及び／又はこれらに埋め込まれている。モノのインターネットを構成するシステムの数は、急増している。ある業界アナリストは、（工業、消費者、政府、医療、及びビジネスの各環境において）接続されているシステムの数は、今後１０年間で５０億から１兆へと増加すると予測している。このような成長の重要な部分は、セルラーネットワーク上など、無線通信を使用したデバイス及びシステムになるであろう。

【０００４】

中でも、新規に接続されたデバイスの採用に共通する問題には、動作性（例えば、ベンダー独自のプラットフォーム及び他のベンダーが提供する各種プラットフォームのデバイスを含む、レガシーデバイスとのインターフェース及び動作の容易性）と、効率（例えば、インフラストラクチャコストなど、接続デバイスの運用コスト）と、有用性（例えば、意図された目的のために実行するデバイスの有効性）と、コスト（例えば、接続デバイス及び支持するインフラストラクチャの取り付け、委託、更新、及び更新（update，and update）と、拡張性と、が挙げられる。これらの問題のある特定の共通点は、このようなデバイスの制御及び管理に使用される通信プロトコルである。例えば、プロトコルは、デバイスが相互間で通信する方法、通信する内容、通信するとき、及び通信相手の人又は物を定義する。

【０００５】

一般に普及している通信プロトコルは、ハイパーテキスト転送プロトコルベース（ＨＴＴＰ）のＲＥｐｒｅｓｅｎｔａｔｉｏｎａｌ　Ｓｔａｔｅ　Ｔｒａｎｓｆｅｒ（ＲＥＳＴ）アプリケーションプログラミングインターフェース（ＡＰＩ）であり、「ＨＴＴＰ　ＲＥＳＴｆｕｌ　ＡＰＩ」とも呼ばれる。ＨＴＴＰ　ＲＥＳＴｆｕｌ　ＡＰＩの２つの一般に普及しているオープンスタンダードフォーマットとして、ＪａｖａＳｃｒｉｐｔ　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎ（「ＪＳＯＮ」）及びＥｘｔｅｎｓｉｖｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ（「ＸＭＬ」）データフォーマットが挙げられる。ＲＥＳＴは、ＨＴＴＰなどの標準化インターフェースを採用し得る分散システムに対応した、ソフトウェアアーキテクチャの形式である。ＲＥＳＴの制約に準ずるアプリケーションは、「ＲＥＳＴｆｕｌ」と記載される場合がある。

【０００６】

ＨＴＴＰ　ＲＥＳＴｆｕｌ　ＡＰＩ及びＪＳＯＮ／ＸＭＬを採用するプロトコルは、柔軟性のあるオープンな通信用スキームを提供するが、そのトレードオフも有する。例えば、人間に読めるテキストの使用による柔軟性はあるが、プロトコルは冗長であり、そのため、非効率的である。

【０００７】

通信プロトコルの動作は、通信チャネルの影響を受ける可能性がある。例えば、ＨＴＴＰは、通信プロトコルとしての制限ではないが、コンピューティング及びネットワークインフラストラクチャ内に一般に採用される、ファイアウォール及びネットワークアドレス変換などのトラフィックルーティングデバイス及びセキュリティデバイスのため、ＨＴＴＰの使用は、場合によっては一方向通信に制限される。ＨＴＴＰポーリングは、このようなインターネットインフラストラクチャ間のＨＴＴＰの一方向の性質、及びそれに類するものを克服するために使用される、典型的なテクニックである。しかし、ＨＴＴＰポーリングの使用は、高い応答メッセージの遅延応答性、高い帯域幅消費、及び高いサーバ使用

率というトレードオフを有する。
【発明の概要】
【発明が解決しようとする課題】
【０００８】
したがって、接続されたデバイスのマシンツーマシン通信用に、ＨＴＴＰ　ＲＥＳＴｆｕｌ　ＡＰＩなどの、又はこれに基づいた、柔軟性のある拡張可能で効率的なプロトコルが必要である。また、このようなプロトコルを低遅延かつ両方向性のものとする必要がある。
【課題を解決するための手段】
【０００９】
総括すると、本明細書に記載の実施形態は、効率的、柔軟、拡張可能、かつ、コスト効果的な方法で、膨大な数のコンピューティングデバイスがコンピューティングプラットフォーム（例えば、マシンツーマシン通信及びＩｏＴ接続デバイス用）に接続することを可能にする、通信プロトコルを提供する。本通信プロトコルは、持続的接続（例えば、ＷｅｂＳｏｃｋｅｔ接続経由）を使用して、インターネットベースのコンピューティングプラットフォーム（例えば、マシンツーマシン接続性及びモノのインターネットアプリケーション用）内のデバイス間でＨＴＴＰ　ＲＥＳＴｆｕｌメッセージを送信するため、低遅延の双方向接続を更に提供する。
【００１０】
動作性及び効率を高めるため、本プロトコルでは、ＪａｖａＳｃｒｉｐｔ　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）及びＥｘｔｅｎｓｉｖｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ（ＸＭＬ）などの標準フォーマットに従った、圧縮及びバイナリ化された動的ＲＥＳＴ　ＡＰＩメッセージを採用する。本動的ＲＥＳＴ　ＡＰＩプロトコルは、マシンが簡単に使用及び生成できる、自己記述型及びモデルベースの方法で、メッセージをフォーマット化する。送信時に、本プロトコルは、動的ＲＥＳＴ　ＡＰＩメッセージを圧縮して、プロトコルに採用されているモデルベーススキーマ及び自己記述型スキーマに特有の表記法及び既知の構造要素を除外及び／又は削除することにより、メッセージの冗長性を軽減する。加えて、圧縮は、メッセージ内の特定の共通要素をバイナリシンボルに置き換える。圧縮された動的ＲＥＳＴ　ＡＰＩプロトコルは、動的ＲＥＳＴ　ＡＰＩメッセージ（例えば、ＪＳＯＮ又はＸＭＬで表現される）のサイズを、５０パーセントを超えて縮小することができ、元のメッセージが送信後に再構築されることを可能にすることが観察される。また、メッセージのバイナリ化には、ＷｅｂＳｏｃｋｅｔ接続との相乗作用があり、低遅延の双方向接続を可能にすることが観察される。
【００１１】
実際に、メッセージの冗長部分（例えば、モデルベーススキーマ及び自己記述型スキーマ）は、プロトコルが制御用の複雑なメッセージを送信及び受信することを可能にし、かつ、アップグレード及びレガシーシステムとの統合を可能にする、メッセージ構成要素の下層組織及びプロパティに部分的に関連する。接続されたデバイスはそれぞれ、典型的にこの下層情報を含む。このように、メッセージの冗長部分が送信されない場合でも、例えば、コンピューティングデバイス及び／又はサーバによって直接使用及び／又は解析されるように、メッセージ全体を送信後に再構築することが可能である。ＪＳＯＮ及びＸＭＬなどの人間に読めるテキストを使用する特定のオープンスタンダードについては、メッセージの冗長部分はまた、それぞれのデータオブジェクトの属性及び／又は名前部分（例えば、ＪＳＯＮにおける属性ペアの属性及びＸＭＬにおける名前／値ペアの名前など）を含む。
【００１２】
動作性及び効率を更に促進するため、プロトコルは、ハイパーテキスト転送プロトコル（ＨＴＴＰ）の一組の共通方法に基づいた、かつ、ＨＴＴＰの一組の共通メッセージ（例えば、ステータスコード、エラーコードなど）に対応する、要求／応答構成体をバイナリシンボルとして表現することによって圧縮する。この圧縮の効果により、メッセージサイ

ズが縮小され得ると同時に、標準ＲＥＳＴ　ＡＰＩ（例えば、ＧＥＴ、ＰＯＳＴ、ＤＥＬＥＴＥ、ＰＵＴなどのＨＴＴＰ動詞）を介した、接続されたデバイス及びレガシーシステム間の通信が可能になる。

【００１３】

本プロトコルを使用した場合、メッセージは、典型的にヘッダ及びペイロード部分を含む。特定の要求／応答構成体（例えば、ステータスコード、エラーコードなど）を採用するメッセージの場合、プロトコルは、その中にこのような要求／応答構成体を有するヘッダのみを送信し得る。これにより、送信効率が更に高まる。動的ＲＥＳＴメッセージの場合、プロトコルは、ヘッダ及びペイロード（動的ＲＥＳＴメッセージを含む）の両方を送信する。特定のＲＥＳＴｆｕｌ形式サービス（例えば、データの読み取り／書き込み操作、リモート手順の実行、イベントトンラスポート、並びにファイル転送及びアプリケーショントンネリングなどの拡張機能など）を有する他の種類のメッセージの場合、プロトコルは、ヘッダ及びペイロードの両方を送信し、ペイロードは、これらの外部ＲＥＳＴｆｕｌ形式サービスを呼び出すためのバイナリメッセージを含む。いくつかの実施例では、ヘッダは、ｉ）すべてのメッセージで採用される共通部分と、ｉｉ）接続されたデバイスとサーバとの間の特定の送信方向に対して採用される補助部分と、を含み得る。この目的のために、本プロトコルは、メッセージの送信方向又は意図された受信者に基づいて、特定のメッセージにのみ補助ヘッダを含むことにより、メッセージ送信の効率を更に高める。

【００１４】

更に拡張性を促進するため、本プロトコルは、チャンクで（すなわち、メッセージの分割された部分で）メッセージの送信を促進する。このように、それぞれのメッセージは、複数のより小さいメッセージにパーティション化され、それによって、接続されたデバイスでデータを受信及び／又は送信するときのサーバのメモリ要件を削減する。この目的のため、この機能は、モノのインターネットアプリケーションなどの、多数の接続デバイスでデータを受信／送信するときに、特に有益である。例えば、１００，０００台の接続デバイスとの１ＭＢのデータの受信又は送信を意図された、プラットフォームサーバがあるとする。このようなサーバは、接続デバイスで同時に受信又は送信される場合、すべてのデータをバッファリングするため、約１００ギガバイト（ＧＢ）（すなわち、１ＭＢのデータ×１００，０００インスタンスのデータ）のメモリ割り当てを必要とする。チャンク化は、このプラットフォームサーバにおけるメモリ制約を有益に軽減する。それぞれのメッセージを、例えば、８キロバイト（ＫＢ）のチャンクからなる複数のメッセージにパーティション化することによって、同一のプラットフォームサーバは、約８００ＭＢのバッファを有する同一の１００，０００台のデバイスから、チャンクデータをバッファリングすることができる。

【００１５】

本プロトコルは、好ましくはアプリケーションレベル（トランスポートレベルではなく）でメッセージをチャンク化及び再アセンブルして、例えば、メッセージ構造の変更を可能にする操作（例えば、所定のアプリケーションの接続性のカスタマイズを可能にすること）の制御を提供する。いくつかの実施形態において、このようなカスタマイズは、プロトコルのトランスポート層で実装される。

【００１６】

一態様では、本開示は、通信デバイスを説明する（例えば、プラットフォームサーバ、エッジサーバ、エンドポイントデバイスなど）。通信デバイスは、ポート、メモリ、及びプロセッサを含む。ポートは、バイナリフォーマットで表現されるメッセージに対応する、１つ又は２つ以上のデータストリーム（例えば、１つ又は２つ以上のパケット）を受信するように構成される。それぞれのデータストリームは、バイナリヘッダ及びバイナリ本体を含む。ポートは、ＷｅｂＳｏｃｋｅｔ接続を介して、１つ又は２つ以上のデータストリームを受信し得る。バイナリ本体は、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み、メタデータ構成体がメッセージ構成体のプロパティを記述する、メッセージの自己記述型スキーマを集合的に

形成する。

【００１７】

メモリは、（ｉ）バイナリヘッダ及びバイナリ本体に関連付けられた第１のメッセージ構造記述と、（ｉｉ）メタデータ構成体に関連付けられた第２のメッセージ構造記述と、を記憶する。

【００１８】

プロセッサは、受信されたメッセージ（例えば、バイナリ及び短縮フォーマットで表現される圧縮されたメッセージ）を作製するために、受信されたデータストリームを解析するように構成される。受信されたメッセージを作製するため、プロセッサは、受信されたメッセージのバイナリヘッダ及びバイナリ本体を決定するために、第１のメッセージ構造記述を使用して受信されたデータストリームを解析し、メタデータ構成体を形成する記述値の１つ又は２つの群を決定するために、第２のメッセージ構造記述を使用してバイナリ本体を解析し、かつ、メッセージ構成体のデータ値の１つ又は２つ以上の群を決定するために、メタデータ構成体の決定された記述値の一部を使用する。いくつかの実施例において、記述値からなる１つ又は２つ以上の群のそれぞれの群は、メッセージ構成体のデータ値からなる１つ又は２つ以上の群のそれぞれの群の所定のデータ値に関連付けられる。メタデータ構成体は、バイナリ本体内でメッセージ構成体に先行し得る。メタデータ構成体の部分、及び一組の文字列に対応するメッセージ構成体は、受信されたメッセージ内で、通信デバイスにネイティブである世界基準コードに基づいたバイナリシンボルとして表現され得る。

【００１９】

いくつかの実施例において、メッセージ構成体のそれぞれのデータ値と、メタデータ構成体のそれぞれの記述値と、に関連付けられた表記オブジェクト（例えば、ＪＳＯＮ属性－値ペアにおける属性、及び／又はＸＭＬ名前－値ペアにおける名前の文字列）が、受信されたメッセージに表示されないように、受信されたメッセージのメッセージ構成体及びメタデータ構成体の両方が、短縮フォーマットで表現される。表記オブジェクトは、受信されたメッセージの非圧縮バージョンで表示され得る。受信されたメッセージの非圧縮バージョンは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）及びＥｘｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（ＸＭＬ）からなる群から選択される、オープンスタンダートフォーマットに準拠し得る。いくつかの実施例において、バイナリヘッダは、ＨＴＴＰ ＲＥＳＴメッセージに関連付けられたフィールドを含む。ＨＴＴＰ ＲＥＳＴメッセージは、ＨＴＴＰ要求メソッド（例えば、ＧＥＴ、ＰＵＴ、ＤＥＬＥＴＥ、ＰＯＳＴ）及びＨＴＴＰステータスコード定義の少なくとも１つを表現するバイナリシンボルの少なくとも１つを含み得る。いくつかの実施例において、データストリームは、補助バイナリヘッダを含む。補助バイナリヘッダは、受信されたデータストリームのターゲット受信者のアイデンティティに関連付けられた、１つ又は２つ以上のフィールドを含み得る。

【００２０】

いくつかの実施例において、メタデータ構成体を形成する記述値からなる１つ又は２つ以上の群のそれぞれの群は、メッセージ構成体のデータ値のそれぞれの型に対する、型定義のバイナリ表現を含む。型定義は、文字列、数値、ブール値、日時オブジェクト、タイムスパンオブジェクト、インフォテーブルオブジェクト、場所オブジェクト、ＸＭＬオブジェクト、ＪＳＯＮオブジェクト、照会オブジェクト、イメージ、ハイパーリンク、イメージリンクオブジェクト、パスワードオブジェクト、ｈｔｍｌオブジェクト、テキスト、タグオブジェクト、スケジュールオブジェクト、バリアントオブジェクト、グローバル一意識別子（ＧＵＩＤ）オブジェクト、バイナリラージオブジェクト（ＢＬＯＢ）、及び整数からなる群から選択されるメンバーを含む。いくつかの実施例において、メタデータ構成体を形成する記述値からなる１つ又は２つ以上の群のそれぞれは、データ値名記述子、データ値記述記述子、及びデータ値型記述子を含み、これらの記述子は、モデルベーススキーマによって定義される。

【００２１】
いくつかの実施例において、メッセージ構成体のデータ値からなる１つ又は２つ以上の群のそれぞれの群及びメタデータ構成体の記述値からなる１つ又は２つ以上の群のそれぞれの群は、１つ又は２つ以上のバイナリマーカーを含む。１つ又は２つ以上のバイナリマーカーは、対応の１つ又は２つ以上の群のそれぞれの始まり及び終わりの少なくとも１つを示す。
【００２２】
別の態様では、本開示は、モデルベーススキーマ及び自己記述型スキーマに従ってフォーマット化された、圧縮及びバイナリ化された動的ＲＥＳＴ　ＡＰＩメッセージを伝達するための、コンピュータ実装型の方法を説明する。方法は、データストリームの第１のメッセージ構造記述及び第２のメッセージ構造記述を、コンピューティングデバイスのメモリで記憶することを含む。第１のメッセージ構造記述は、バイナリフォーマットで表現されるメッセージのバイナリヘッダ及びバイナリ本体の形状を定義する。バイナリ本体は、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み、メタデータ構成体がメッセージ構成体のプロパティを記述する、メッセージの自己記述型スキーマを集合的に形成する。第２のメッセージ構造記述は、メタデータ構成体の構造を定義する。
【００２３】
方法は、モデルベーススキーマ及び自己記述型スキーマに従ってフォーマット化された、圧縮及びバイナリ化された動的ＲＥＳＴ　ＡＰＩメッセージに対応する、１つ又は２つ以上のデータストリームを、コンピューティングデバイスのポートで受信することを更に含む。１つ又は２つ以上のデータストリームは、ＷｅｂＳｏｃｋｅｔ接続を介して受信され得る。
【００２４】
方法は、受信されたメッセージを作製するため、受信されたデータストリームを、コンピューティングデバイスのプロセッサによって解析することを更に含む。受信されたデータストリームを解析することは、受信されたメッセージのバイナリヘッダ及びバイナリ本体を決定するために、第１のメッセージ構造記述を使用して受信されたデータストリームを解析することと、受信されたメッセージのメタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定するために、第２のメッセージ構造記述を使用してバイナリ本体を解析することと、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値からなる１つ又は２つ以上の群を決定することと、を含む。メタデータ構成体は、バイナリ本体内でメッセージ構成体に先行し得る。メタデータ構成体の部分、及び一組の文字列に対応するメッセージ構成体は、受信されたメッセージ内で、コンピューティングデバイスにネイティブである世界基準コード（例えば、Ｕｎｉｃｏｄｅ－８）に基づいたバイナリシンボルとして表現され得る。
【００２５】
いくつかの実施例において、記述値からなる１つ又は２つ以上の群のそれぞれの群は、メッセージ構成体のデータ値からなる１つ又は２つ以上の群のそれぞれの群の所定のデータ値に関連付けられる。
【００２６】
いくつかの実施例において、メッセージ構成体のそれぞれのデータ値と、メタデータ構成体のそれぞれの記述値と、に関連付けられた表記オブジェクト（例えば、ＪＳＯＮ属性－値ペアにおける属性、及び／又はＸＭＬ名前－値ペアにおける名前の文字列）が、受信されたメッセージに表示されず、表記オブジェクトが受信されたメッセージの非圧縮バージョンに表示されるように、受信されたメッセージのメッセージ構成体及びメタデータ構成体の両方が、短縮フォーマットで表現される。受信されたメッセージの非圧縮バージョンは、ＪａｖａＳｃｒｉｐｔ　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）及びＥｘｔｅｎｓｉｂｌｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ（ＸＭＬ）からなる群から選択される、オープンスタンダートフォーマットに準拠し得る。

【００２７】
いくつかの実施例において、バイナリヘッダは、ＨＴＴＰ　ＲＥＳＴメッセージに関連付けられたフィールドを含む。ＨＴＴＰ　ＲＥＳＴメッセージは、ＨＴＴＰ要求メソッド（例えば、ＧＥＴ、ＰＵＴ、ＤＥＬＥＴＥ、ＰＯＳＴ）及びＨＴＴＰステータスコード定義の少なくとも１つを表現するバイナリシンボルを含み得る。いくつかの実施例において、データストリームは、補助バイナリヘッダを含む。補助バイナリヘッダは、受信されたデータストリームのターゲット受信者のアイデンティティに関連付けられた、１つ又は２つ以上のフィールドを含み得る。
【００２８】
いくつかの実施例において、メッセージ構成体のデータ値からなる１つ又は２つ以上の群のそれぞれの群及びメタデータ構成体の記述値からなる１つ又は２つ以上の群のそれぞれの群は、１つ又は２つ以上のバイナリマーカーを含む。１つ又は２つ以上のバイナリマーカーは、対応の１つ又は２つ以上の群のそれぞれの始まり及び終わりの少なくとも１つを示すことができる。
【００２９】
いくつかの実施例において、メタデータ構成体を形成する記述値からなる１つ又は２つ以上の群のそれぞれは、データ値名記述子、データ値記述記述子、及びデータ値型記述子を含む。記述子は、好ましくはモデルベーススキーマによって定義される。
【００３０】
別の態様では、本開示は、記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明する。命令は、コンピューティングデバイスのプロセッサの実行時に、データストリームの第１のメッセージ構造記述及び第２のメッセージ構造記述を、コンピューティングデバイスのメモリで記憶することを、プロセッサに実行させる。第１のメッセージ構造記述は、バイナリフォーマットで表現されるメッセージのバイナリヘッダ及びバイナリ本体の形状を定義する。バイナリ本体は、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み、メタデータ構成体がメッセージ構成体のプロパティを記述する、メッセージの自己記述型スキーマを集合的に形成する。第２のメッセージ構造記述は、メタデータ構成体の構造を定義する。
【００３１】
命令は、プロセッサによる実行時に、モデルベーススキーマ及び自己記述型スキーマに従ってフォーマット化された、圧縮及びバイナリ化された動的ＲＥＳＴ　ＡＰＩメッセージに対応する、１つ又は２つ以上のデータストリームを、コンピューティングデバイスのポートで受信することを、プロセッサに実行させる。ポートは、ＷｅｂＳｏｃｋｅｔ接続を介して、１つ又は２つ以上のデータストリームを受信し得る。
【００３２】
命令は、プロセッサによる実行時に、受信されたデータストリームを解析して受信されたメッセージを作製することを、プロセッサに更に実行させる。受信されたデータストリームを解析することは、受信されたメッセージのバイナリヘッダ及びバイナリ本体を決定するために、第１のメッセージ構造記述を使用して受信されたデータストリームを解析することと、受信されたメッセージのメタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定するために、第２のメッセージ構造記述を使用してバイナリ本体を解析することと、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値からなる１つ又は２つ以上の群を決定することと、を含む。メタデータ構成体は、バイナリ本体内でメッセージ構成体に先行し得る。メタデータ構成体の部分、及び一組の文字列に対応するメッセージ構成体は、受信されたメッセージ内で、通信デバイスにネイティブである世界基準コードに基づいたバイナリシンボルとして表現され得る。
【００３３】
いくつかの実施例において、メッセージ構成体のそれぞれのデータ値と、メタデータ構成体のそれぞれの記述値と、に関連付けられた表記オブジェクト（例えば、ＪＳＯＮ属性－値ペアにおける属性、及び／又はＸＭＬ名前－値ペアにおける名前の文字列）が、受信

されたメッセージに表示されないように、受信されたメッセージのメッセージ構成体及びメタデータ構成体の両方が、短縮フォーマットで表現される。表記オブジェクトは、受信されたメッセージの非圧縮バージョンで表示され得る。受信されたメッセージの非圧縮バージョンは、ＪａｖａＳｃｒｉｐｔ　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）及びＥｘｔｅｎｓｉｂｌｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ（ＸＭＬ）からなる群から選択される、オープンスタンダートフォーマットに準拠し得る。いくつかの実施例において、バイナリヘッダは、ＨＴＴＰ　ＲＥＳＴメッセージに関連付けられたフィールドを含む。ＨＴＴＰ　ＲＥＳＴメッセージは、ＨＴＴＰ要求メソッド（例えば、ＧＥＴ、ＰＵＴ、ＤＥＬＥＴＥ、ＰＯＳＴ）及びＨＴＴＰステータスコード定義の少なくとも１つを表現するバイナリシンボルの少なくとも１つを含み得る。いくつかの実施例において、データストリームは、補助バイナリヘッダを含む。補助バイナリヘッダは、受信されたデータストリームのターゲット受信者のアイデンティティに関連付けられた、１つ又は２つ以上のフィールドを含み得る。

【００３４】

いくつかの実施例において、メタデータ構成体を形成する記述値からなる１つ又は２つ以上の群のそれぞれの群は、メッセージ構成体のデータ値のそれぞれの型に対する、型定義のバイナリ表現を含む。型定義は、文字列、数値、ブール値、日時オブジェクト、タイムスパンオブジェクト、インフォテーブルオブジェクト、場所オブジェクト、ＸＭＬオブジェクト、ＪＳＯＮオブジェクト、照会オブジェクト、イメージ、ハイパーリンク、イメージリンクオブジェクト、パスワードオブジェクト、ｈｔｍｌオブジェクト、テキスト、タグオブジェクト、スケジュールオブジェクト、バリアントオブジェクト、グローバル一意識別子（ＧＵＩＤ）オブジェクト、バイナリラージオブジェクト（ＢＬＯＢ）、及び整数からなる群から選択されるメンバーを含む。いくつかの実施例において、メタデータ構成体を形成する記述値からなる１つ又は２つ以上の群のそれぞれは、データ値名記述子、データ値記述記述子、及びデータ値型記述子を含み、これらの記述子は、モデルベーススキーマによって定義される。

【００３５】

いくつかの実施例において、メッセージ構成体のデータ値からなる１つ又は２つ以上の群のそれぞれの群及びメタデータ構成体の記述値からなる１つ又は２つ以上の群のそれぞれの群は、１つ又は２つ以上のバイナリマーカーを含む。１つ又は２つ以上のバイナリマーカーは、対応の１つ又は２つ以上の群のそれぞれの始まり及び終わりの少なくとも１つを示す。

【００３６】

別の態様では、本開示は、プロセッサと、持続的ステートレス接続（例えば、ＷｅｂＳｏｃｋｅｔ接続）を介してデータストリームをコンピューティングデバイスに送信するように構成されたポートと、そこに記憶された命令及び動的ＲＥＳＴ　ＡＰＩモデルを有するメモリと、を含む、通信デバイスを説明する。メモリの命令は、プロセッサによる実行時に、ポートを介して、第２の通信デバイスとの１つ又は２つ以上の持続的ステートレス接続を確立することを、プロセッサに実行させる。メモリの命令は、プロセッサによる実行時に、記憶された動的ＲＥＳＴ　ＡＰＩモデルを使用して、自己記述型要求メッセージに対応する１つ又は２つ以上のデータストリームを作製することを、プロセッサに更に実行させる。メモリの命令は、プロセッサによる実行時に、データストリームがポートによって第２の通信デバイスに送信されるようにすることを、プロセッサに更に実行させる。

【００３７】

いくつかの実施例において、持続的ステートレス接続は、ＷｅｂＳｏｃｋｅｔ接続である。ポートは、ハイパーテキスト転送プロトコル（ＨＴＴＰ）ポート、セキュアハイパーテキスト転送プロトコル（ＨＴＴＰＳ）ポート、ＷｅｂＳｏｃｋｅｔ（ＷＳ）ポート、及びＷｅｂＳｏｃｋｅｔセキュア（ＷＳＳ）ポートからなる群から選択されるメンバーであり得る。ポートは、いくつかの実施例において、ポート値８０又は４４３を有する。

【００３８】

いくつかの実施例において、動的ＲＥＳＴ　ＡＰＩモデルは、モデルベーススキーマ及び自己記述型スキーマに従ってフォーマット化された、要求メッセージの記述を含む。要求メッセージの記述は、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み得、メタデータ構成体がメッセージ構成体のプロパティを記述する、メッセージの自己記述型スキーマを集合的に形成する。メッセージ構成体は、メッセージのデータオブジェクトに対応するデータ値（例えば、行データ）の１つ又は２つ以上の群を含み得、メタデータ構成体は、メッセージのプロパティオブジェクトに対応する記述値（例えば、フィールド定義）の１つ又は２つ以上の群を含む。記述値からなる１つ又は２つ以上の群のそれぞれの群は、メッセージ構成体のデータ値からなる１つ又は２つ以上の群のそれぞれの群の所定のデータ値に関連付けられ得る。いくつかの実施例において、メッセージ構成体のそれぞれのデータ値と、メタデータ構成体のそれぞれの記述値と、に関連付けられた表記オブジェクト（例えば、ＪＳＯＮ属性－値ペアにおける属性、及び／又はＸＭＬ名前－値ペアにおける名前）が、メッセージに表示されず、表記オブジェクトがメッセージの非圧縮バージョンに表示されるように、メッセージのメッセージ構成体及びメタデータ構成体の両方は、短縮フォーマットで表現される。いくつかの実施例において、自己記述型要求メッセージは、ＪａｖａＳｃｒｉｐｔ　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトからなる群から選択されるフォーマットで表現される。いくつかの実施例において、自己記述型要求メッセージは、バイナリフォーマットで表現される。

【００３９】

別の態様では、本開示は、コンピュータ実装型の方法を説明する。方法は、動的ＲＥＳＴ　ＡＰＩモデルを、コンピューティングデバイスのメモリで記憶することを含む。方法は、コンピューティングデバイスのプロセッサによって、記憶された動的ＲＥＳＴ　ＡＰＩモデルを使用して、自己記述型要求メッセージに対応する１つ又は２つ以上のデータストリームを作製することを更に含む。方法は、コンピューティングデバイスのポートを介して、第２のコンピューティングデバイスとの持続的ステートレス接続（例えば、ＷｅｂＳｏｃｋｅｔ接続）を確立することを更に含む。方法は、ポートを介して、１つ又は２つ以上のデータストリームを第２のコンピューティングデバイスに送信することを更に含む。

【００４０】

いくつかの実施例において、持続的ステートレス接続は、ＷｅｂＳｏｃｋｅｔ接続である。ポートは、ハイパーテキスト転送プロトコル（ＨＴＴＰ）ポート、セキュアハイパーテキスト転送プロトコル（ＨＴＴＰＳ）ポート、ＷｅｂＳｏｃｋｅｔ（ＷＳ）ポート、及びＷｅｂＳｏｃｋｅｔセキュア（ＷＳＳ）ポートからなる群から選択されるメンバーであり得る。ポートは、いくつかの実施例において、ポート値８０又は４４３を有する。

【００４１】

いくつかの実施例において、動的ＲＥＳＴ　ＡＰＩモデルは、モデルベーススキーマ及び自己記述型スキーマに従ってフォーマット化された、要求メッセージの記述を含む。要求メッセージの記述は、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み得、メタデータ構成体がメッセージ構成体のプロパティを記述する、メッセージの自己記述型スキーマを集合的に形成する。メッセージ構成体は、メッセージのデータオブジェクトに対応するデータ値（例えば、行データ）の１つ又は２つ以上の群を含み得、メタデータ構成体は、メッセージのプロパティオブジェクトに対応する記述値（例えば、フィールド定義）の１つ又は２つ以上の群を含む。記述値からなる１つ又は２つ以上の群のそれぞれの群は、メッセージ構成体のデータ値からなる１つ又は２つ以上の群のそれぞれの群の所定のデータ値に関連付けられ得る。いくつかの実施例において、メッセージ構成体のそれぞれのデータ値と、メタデータ構成体のそれぞれの記述値と、に関連付けられた表記オブジェクト（例えば、ＪＳＯＮ属性－値ペアにおける属性、及び／又はＸＭＬ名前－値ペアにおける名前）が、メッセージに表

示されず、表記オブジェクトがメッセージの非圧縮バージョンに表示されるように、メッセージのメッセージ構成体及びメタデータ構成体の両方は、短縮フォーマットで表現される。いくつかの実施例において、自己記述型要求メッセージは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトからなる群から選択されるフォーマットで表現される。いくつかの実施例において、自己記述型要求メッセージは、バイナリフォーマットで表現される。

【００４２】

別の態様では、本開示は、記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明する。命令は、コンピューティングデバイスのプロセッサによる実行時に、動的ＲＥＳＴ ＡＰＩモデルをコンピューティングデバイスのメモリで記憶することを、プロセッサに実行させる。命令は、プロセッサによる実行時に、動的ＲＥＳＴ ＡＰＩモデルを使用して、自己記述型要求メッセージに対応する、１つ又は２つ以上のデータストリームを作製することを、プロセッサに更に実行させる。命令は、プロセッサによる実行時に、コンピューティングデバイスのポートを介して、第２のコンピューティングデバイスとの持続的ステートレス接続（例えば、ＷｅｂＳｏｃｋｅｔ接続）を確立することを、プロセッサに更に実行させる。命令は、プロセッサによる実行時に、ポートを介して、１つ又は２つ以上のデータストリームを第２のコンピューティングデバイスに送信することを、プロセッサに更に実行させる。

【００４３】

いくつかの実施例において、動的ＲＥＳＴ ＡＰＩモデルは、モデルベーススキーマ及び自己記述型スキーマに従ってフォーマット化された、要求メッセージの記述を含む。要求メッセージの記述は、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み得、メタデータ構成体がメッセージ構成体のプロパティを記述する、メッセージの自己記述型スキーマを集合的に形成する。メッセージ構成体は、メッセージのデータオブジェクトに対応するデータ値（例えば、行データ）の１つ又は２つ以上の群を含み得、メタデータ構成体は、メッセージのプロパティオブジェクトに対応する記述値（例えば、フィールド定義）の１つ又は２つ以上の群を含む。記述値からなる１つ又は２つ以上の群のそれぞれの群は、メッセージ構成体のデータ値からなる１つ又は２つ以上の群のそれぞれの群の所定のデータ値に関連付けられ得る。いくつかの実施例において、メッセージ構成体のそれぞれのデータ値と、メタデータ構成体のそれぞれの記述値と、に関連付けられた表記オブジェクト（例えば、ＪＳＯＮ属性－値ペアにおける属性、及び／又はＸＭＬ名前－値ペアにおける名前）が、メッセージに表示されず、表記オブジェクトがメッセージの非圧縮バージョンに表示されるように、メッセージのメッセージ構成体及びメタデータ構成体の両方は、短縮フォーマットで表現される。いくつかの実施例において、自己記述型要求メッセージは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトからなる群から選択されるフォーマットで表現される。いくつかの実施例において、自己記述型要求メッセージは、バイナリフォーマットで表現される。

【００４４】

別の態様では、本開示は、コンピュータ実装型の方法を説明する。方法は、第１のコンピューティングデバイス（例えば、中間サーバ）のメモリ内に、第１の通信プロトコル及び第２の通信プロトコルを提供することを含む。第１の通信プロトコルは、（ｉ）複数の制御コード（例えば、１つ又は２つ以上のＨＴＴＰ要求メソッド及び１つ又は２つ以上のＨＴＴＰステータスコード定義に対応）と、（ｉｉ）自己記述型スキーマ及びモデルベーススキーマによって記述される複数のメッセージ構造（例えば、動的ＲＥＳＴ ＡＰＩモデルに対応）と、を含む。第２の通信プロトコルのインスタンスは、デバイスのメモリ内の第１の通信プロトコルのインスタンスに関連付けてマップされる。第１の通信プロトコルと第２の通信プロトコルとの間の関連マッピングは、持続的メモリに記憶され得る。

【００４５】
方法は、第２のコンピューティングデバイス（例えば、レガシーコンピューティングデバイス）から、第２の通信プロトコルに従ってフォーマット化されたメッセージに対応する１つ又は２つ以上の第１のデータストリームを、第１のコンピューティングデバイスのポートで受信することを更に含む。１つ又は２つ以上の第１のデータストリームは、無線接続を介して受信され得る。いくつかの実施例において、無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－５０、ＯＭＡ ＬＷ Ｍ２Ｍ、並びにＥＴＳＩ Ｍ２Ｍを含む。
【００４６】
方法は、第１のコンピューティングデバイスのプロセッサによって、受信されたメッセージが、制御メッセージ又はデータメッセージのいずれかに対応するかどうかを決定することを更に含む。
【００４７】
方法は、プロセッサによって、第２の通信プロトコルに従ってフォーマット化された、受信されたメッセージを使用して、第１の通信プロトコルに従ってフォーマット化されたメッセージに対応する、１つ又は２つ以上の第２のデータストリームを生成することを更に含む。受信されたメッセージの応答が、制御メッセージに対応すると決定された場合、方法は、プロセッサによって、第１の通信プロトコルの複数の制御コードの１つ又は２つ以上の制御コードに、受信されたメッセージをマップすることを含む。複数の制御コードの一部は、１つ又は２つ以上のＨＴＴＰ要求メソッド及び１つ又は２つ以上のＨＴＴＰステータスコード定義に対応し得る。１つ又は２つ以上のＨＴＴＰ要求メソッドは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択されるメンバーを含み得る。いくつかの実施例において、１つ又は２つ以上のＨＴＴＰ要求メソッドのそれぞれは、バイナリシンボルとして表現される。受信されたメッセージの応答が、データメッセージに対応すると判断された場合、方法は、プロセッサによって、第１の通信プロトコルに従ってフォーマット化された、対応する動的ＲＥＳＴ要求メッセージを生成することを含む。生成されたメッセージは、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み、メタデータ構成体がメッセージ構成体のプロパティを記述する、生成されたメッセージの自己記述型スキーマを集合的に形成する。
【００４８】
方法は、ポートを介して、１つ又は２つ以上の第２のデータストリームを第３のコンピューティングデバイス（例えば、プラットフォームサーバ）に送信することを更に含む。第１のコンピューティングデバイスは、１つ又は２つ以上の第２のデータストリームを、ＷｅｂＳｏｃｋｅｔ接続を介して第３のコンピューティングデバイスに送信し得る。
【００４９】
いくつかの実施例において、方法は、プロセッサによって、１つ又は２つ以上の第２のデータストリームのそれぞれにバイナリヘッダを追加することを更に含み、バイナリヘッダは、受信されたデータストリームのターゲット受信者を含む。
【００５０】
別の態様では、本開示は、記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明する。命令は、第１の通信プロトコルの定義、第２の通信プロトコルの定義、及びその間の関連マッピングを含む。第１の通信プロトコルと第２の通信プロトコルとの間の関連マッピングは、持続的メモリに記憶され得る。第１の通信プロトコルは、（ｉ）複数の制御コード（例えば、１つ又は２つ以上のＨＴＴＰ要求メソッド及び１つ又は２つ以上のＨＴＴＰステータスコード定義に対応）と、（ｉｉ）自己記述型スキーマ及びモ

デルベーススキーマによって記述される複数のメッセージ構造（例えば、動的ＲＥＳＴＡＰＩモデルに対応）と、を含む。

【００５１】

命令は、コンピューティングデバイス（例えば、中間サーバ）のプロセッサによる実行時に、第２のコンピューティングデバイス（例えば、レガシーコンピューティングデバイス）から、第２の通信プロトコルに従ってフォーマット化されたメッセージに対応する１つ又は２つ以上の第１のデータストリームを、コンピューティングデバイスのポートで受信することを、プロセッサに実行させる。１つ又は２つ以上の第１のデータストリームは、無線接続を介して受信され得る。いくつかの実施例において、無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－５０、ＯＭＡ ＬＷ Ｍ２Ｍ、並びにＥＴＳＩ Ｍ２Ｍを含む。

【００５２】

命令は、プロセッサによる実行時に、受信されたメッセージが制御メッセージ又はデータメッセージのいずれかに対応するかどうかを決定することを、プロセッサに更に実行させる。いくつかの実施例において、複数の制御コードの一部は、１つ又は２つ以上のＨＴＴＰ要求メソッド及び１つ又は２つ以上のＨＴＴＰステータスコード定義に対応する。１つ又は２つ以上のＨＴＴＰ要求メソッドは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択されるメンバーを含み得る。１つ又は２つ以上のＨＴＴＰ要求メソッドのそれぞれは、バイナリシンボルとして表示され得る。

【００５３】

命令は、プロセッサによる実行時に、第１の通信プロトコルに従ってフォーマット化されたメッセージに対応する、１つ又は２つ以上の第２のデータストリームを生成することを、プロセッサに更に実行させる。受信されたメッセージの応答が、制御メッセージに対応すると決定された場合、プロセッサは、第１の通信プロトコルの複数の制御コードの１つ又は２つ以上の制御コードに、受信されたメッセージをマップする。受信されたメッセージの応答が、データメッセージに対応すると決定された場合、プロセッサは、第１の通信プロトコルに従ってフォーマット化された、対応する動的ＲＥＳＴ要求メッセージを生成する。生成されたメッセージは、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み、メタデータ構成体がメッセージ構成体のプロパティを記述する、生成されたメッセージの自己記述型スキーマを集合的に形成する。

【００５４】

命令は、プロセッサによる実行時に、ポートを介して、１つ又は２つ以上の第２のデータストリームを第３のコンピューティングデバイス（例えば、プラットフォームサーバ）に送信することを、プロセッサに更に実行させる。いくつかの実施例において、コンピューティングデバイスは、１つ又は２つ以上の第２のデータストリームを、ＷｅｂＳｏｃｋｅｔ接続を介して第３のコンピューティングデバイスに送信する。

【００５５】

いくつかの実施例において、１つ又は２つ以上のＨＴＴＰ要求メソッドのそれぞれは、バイナリシンボルとして表現される。

【００５６】

いくつかの実施例において、命令は、プロセッサによる実行時に、１つ又は２つ以上の第２のデータストリームのそれぞれにバイナリヘッダを追加することをプロセッサに更に実行させ、バイナリヘッダは、受信されたデータストリームのターゲット受信者を含む。

【００５７】

別の態様では、本開示は、プロセッサと、ポートと、そこに記憶された命令を有するメ

モリと、を含む、システムを説明する。命令は、第１の通信プロトコルの定義、第２の通信プロトコルの定義、及びその間の関連マッピングを含む。第１の通信プロトコルは、（ｉ）複数の制御コード（例えば、１つ又は２つ以上のＨＴＴＰ要求メソッド及び１つ又は２つ以上のＨＴＴＰステータスコード定義に対応）と、（ｉｉ）自己記述型スキーマによって記述される複数のメッセージ構造と、を含む。

【００５８】

命令は、コンピューティングデバイス（例えば、中間サーバ）のプロセッサによる実行時に、第２のコンピューティングデバイス（例えば、レガシーコンピューティングデバイス）から、第２の通信プロトコルに従ってフォーマット化されたメッセージに対応する１つ又は２つ以上の第１のデータストリームを、コンピューティングデバイスのポートで受信することを、プロセッサに実行させる。１つ又は２つ以上の第１のデータストリームは、無線接続を介して受信され得る。いくつかの実施例において、無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－５０、ＯＭＡ ＬＷ Ｍ２Ｍ、並びにＥＴＳＩ Ｍ２Ｍを含む。

【００５９】

命令は、プロセッサによる実行時に、受信されたメッセージが制御メッセージ又はデータメッセージのいずれかに対応するかどうかを決定することを、プロセッサに更に実行させる。いくつかの実施例において、複数の制御コードの一部は、１つ又は２つ以上のＨＴＴＰ要求メソッド及び１つ又は２つ以上のＨＴＴＰステータスコード定義に対応する。１つ又は２つ以上のＨＴＴＰ要求メソッドは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択されるメンバーを含み得る。１つ又は２つ以上のＨＴＴＰ要求メソッドのそれぞれは、バイナリシンボルとして表示され得る。

【００６０】

命令は、プロセッサによる実行時に、第１の通信プロトコルに従ってフォーマット化されたメッセージに対応する、１つ又は２つ以上の第２のデータストリームを生成することを、プロセッサに更に実行させる。受信されたメッセージの応答が、制御メッセージに対応すると決定された場合、プロセッサは、第１の通信プロトコルの複数の制御コードの１つ又は２つ以上の制御コードに、受信されたメッセージをマップする。受信されたメッセージの応答が、データメッセージに対応すると決定された場合、プロセッサは、第１の通信プロトコルに従ってフォーマット化された、対応する動的ＲＥＳＴ要求メッセージを生成する。生成されたメッセージは、メタデータ構成体（例えば、フィールド定義のリスト）及びメッセージ構成体（例えば、行データのリスト）を含み、メタデータ構成体がメッセージ構成体のプロパティを記述する、生成されたメッセージの自己記述型スキーマを集合的に形成する。

【００６１】

命令は、プロセッサによる実行時に、ポートを介して、１つ又は２つ以上の第２のデータストリームを第３のコンピューティングデバイス（例えば、プラットフォームサーバ）に送信することを、プロセッサに更に実行させる。いくつかの実施例において、コンピューティングデバイスは、１つ又は２つ以上の第２のデータストリームを、ＷｅｂＳｏｃｋｅｔ接続を介して第３のコンピューティングデバイスに送信する。

【００６２】

いくつかの実施例において、１つ又は２つ以上のＨＴＴＰ要求メソッドのそれぞれは、バイナリシンボルとして表現される。

【００６３】

いくつかの実施例において、命令は、プロセッサによる実行時に、１つ又は２つ以上の第２のデータストリームのそれぞれにバイナリヘッダを追加することをプロセッサに更に

実行させ、バイナリヘッダは、受信されたデータストリームのターゲット受信者を含む。
【００６４】
別の態様では、本開示は、プロセッサと、複数のコンピューティングデバイスから複数のデータストリームを受信するように構成された通信ポートと、そこに記憶された命令を有するメモリと、を有する通信エンジンを説明する。複数のデータストリームのそれぞれのデータストリームは、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子（例えば、チャンクの合計数）を含む。通信ポートは、複数の持続的ステートレス接続（例えば、複数のＷｅｂＳｏｃｋｅｔ接続）を介して、複数のデータストリームを受信し得る。命令は、プロセッサによる実行時に、複数のデータストリームの第１のデータストリームを解析して、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを、プロセッサに実行させる。命令は、プロセッサによる実行時に、メモリ内の第１のバッファを割り当てることをプロセッサに更にさせ、第１のバッファは、第１の監視対象デバイスの第１のメッセージに関連付けられ、かつ、部分的に、第１のデータストリームから解析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。命令は、プロセッサによる実行時に、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、受信されたデータストリームのメッセージ部分を第１のバッファに記憶することを、プロセッサに更に実行させる。命令は、プロセッサによる実行時に、第１のバッファのメッセージ部分をデコードして、第１の監視対象デバイスの完全な第１のメッセージを生成することを、プロセッサに更に実行させる。
【００６５】
いくつかの実施例において、命令は、プロセッサによる実行時に、複数のデータストリームの第２のデータストリームを解析して、第１の監視対象デバイスに関連付けられた第２のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを、プロセッサに更に実行させる。このような実施例において、命令は、プロセッサによる実行時に、メモリ内の第２のバッファを割り当てることをプロセッサに更にさせ、第２のバッファは、第１の監視対象デバイスの第２のメッセージに関連付けられ、かつ、部分的に、第２のデータストリームから解析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。このような実施例において、命令は、プロセッサによる実行時に、第１の監視対象デバイスに関連付けられた第２のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、受信されたデータストリームのメッセージ部分を第２のバッファに記憶することを、プロセッサに更に実行させる。このような実施例において、命令は、プロセッサによる実行時に、第２のバッファのメッセージ部分をデコードして、第１の監視対象デバイスの完全な第２のメッセージを生成することを、プロセッサに更に実行させる。
【００６６】
いくつかの実施例において、命令は、プロセッサによる実行時に、複数のデータストリームの第３のデータストリームを解析して、第２の監視対象デバイスに関連付けられた第３のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを、プロセッサに更に実行させる。このような実施例において、命令は、プロセッサによる実行時に、メモリ内の第３のバッファを割り当てることをプロセッサに更に実行させ、第３のバッファは、第２の監視対象デバイスの第３のメッセージに関連付けられ、かつ、部分的に、第３のデータストリームから解析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。このような実施例において、命令は、プロセッサによる実行時に、第２の監視対象デバイスに関連付けられた第３のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、受信されたデータストリームのメッセージ部分を第３のバッファに記憶することを、プロセッサに更に実行させる。このような実施例において、命令は、プロセッサによる実行時に、第３のバッファのメッセージ部分をデコードして、第２の監視対象デバイス

の完全な第３のメッセージを生成することを、プロセッサに更に実行させる。
【００６７】
いくつかの実施例において、受信されたデータストリームのそれぞれのメッセージ部分は、受信されたデータストリームから解析されたチャンク識別番号に対応する場所で、対応のバッファ内に記憶される。他の実施例において、受信されたデータストリームのそれぞれのメッセージ部分は、連続的な方法で対応のバッファ内に記憶される。
【００６８】
いくつかの実施例において、チャンクサイズ識別子は、受信されたデータストリームのメッセージ部分のサイズに対応する。
【００６９】
いくつかの実施例において、命令は、プロセッサによる実行時に、それぞれの新規メッセージについて、新規メッセージに対応するデータストリームから解析された、チャンクサイズ識別子及び合計チャンク識別子に基づいて、メモリ内の対応のバッファを割り当てることを、プロセッサに更に実行させる。
【００７０】
いくつかの実施例において、それぞれの対応のバッファのメッセージ部分は、バッファが満杯になるとデコードされる。
【００７１】
いくつかの実施例において、命令は、プロセッサによる実行時に、対応のバッファの割り当てに使用された合計チャンク識別子に基づき、それぞれの対応バッファについて、記憶されたメッセージ部分（例えば、バイナリ文字列又はアレイ）のリストを作製することと、記憶されたメッセージ部分に関連付けられたチャンク識別番号に基づき、対応のバッファに記憶されたそれぞれのメッセージ部分について、記憶されたメッセージ部分のリストを更新することと、リストが完了したときに、対応のバッファの記憶されたメッセージ部分をデコードすることと、をプロセッサに実行させる。
【００７２】
いくつかの実施例において、チャンクサイズ識別子は、少なくとも１６ビットである。いくつかの実施例において、チャンクサイズ識別子は、少なくとも３２ビットである。いくつかの実施例において、合計チャンク識別子は、少なくとも１６ビットである。いくつかの実施例において、合計チャンク識別子は、少なくとも３２ビットである。
【００７３】
いくつかの実施例において、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子は、所定のデータストリームのヘッダ部分を集合的に形成する。いくつかの実施例において、それぞれのデータストリームは、ヘッダメッセージを含む。ヘッダメッセージは、マルチパートヘッダ部分の存在を示す、マルチパートヘッダフィールドを含み得る。このマルチパートヘッダ部分は、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を含み得る。いくつかの実施例において、マルチパートヘッダ部分は、受信されたデータストリームのメッセージ部分に対応する、発信元のデータを有するコンピューティングデバイスに関連付けられた名前識別子を含む。
【００７４】
いくつかの実施例において、命令は、プロセッサによる実行時に、第１のバッファでの第１のメッセージのデコード時に、成功メッセージに対応するデータストリームが、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、データストリームを送信したコンピューティングデバイスに送信されるようにすることを、プロセッサに更に実行させる。
【００７５】
別の態様では、本開示は、記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明する。命令は、プロセッサによる実行時に、複数のコンピューティングデバイスから複数のデータストリームを、コンピューティングデバイスのポートで受信することをプロセッサにさせ、この複数のデータストリームのそれぞれのデータストリームは、

チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子（例えば、チャンクの合計数）を含む。ポートは、複数の持続的ステートレス接続（例えば、複数のＷｅｂＳｏｃｋｅｔ接続）を介して、複数のデータストリームを受信し得る。命令は、プロセッサによる実行時に、複数のデータストリームの第１のデータストリームを解析して、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを、プロセッサに更に実行させる。命令は、プロセッサによる実行時に、メモリ内の第１のバッファを割り当てることをプロセッサに更に実行させ、第１のバッファは、第１の監視対象デバイスの第１のメッセージに関連付けられ、かつ、部分的に、第１のデータストリームから解析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。命令は、プロセッサによる実行時に、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、受信されたデータストリームのメッセージ部分を第１のバッファに記憶することを、プロセッサに更に実行させる。命令は、プロセッサによる実行時に、第１のバッファのメッセージ部分をデコードして、第１の監視対象デバイスの完全な第１のメッセージを生成することを、プロセッサに更に実行させる。

【００７６】

いくつかの実施例において、命令は、プロセッサによる実行時に、複数のデータストリームの第２のデータストリームを解析して、第１の監視対象デバイスに関連付けられた第２のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを、プロセッサに更に実行させる。このような実施例において、命令は、プロセッサによる実行時に、メモリ内の第２のバッファを割り当てることをプロセッサに更にさせ、第２のバッファは、第１の監視対象デバイスの第２のメッセージに関連付けられ、かつ、部分的に、第２のデータストリームから解析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。このような実施例において、命令は、プロセッサによる実行時に、第１の監視対象デバイスに関連付けられた第２のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、受信されたデータストリームのメッセージ部分を第２のバッファに記憶することを、プロセッサに更に実行させる。このような実施例において、命令は、プロセッサによる実行時に、第２のバッファのメッセージ部分をデコードして、第１の監視対象デバイスの完全な第２のメッセージを生成することを、プロセッサに更に実行させる。

【００７７】

いくつかの実施例において、命令は、プロセッサによる実行時に、複数のデータストリームの第３のデータストリームを解析して、第２の監視対象デバイスに関連付けられた第３のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを、プロセッサに更に実行させる。このような実施例において、命令は、プロセッサによる実行時に、メモリ内の第３のバッファを割り当てることをプロセッサに更に実行させ、第３のバッファは、第２の監視対象デバイスの第３のメッセージに関連付けられ、かつ、部分的に、第３のデータストリームから解析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。このような実施例において、命令は、プロセッサによる実行時に、第２の監視対象デバイスに関連付けられた第３のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、受信されたデータストリームのメッセージ部分を第３のバッファに記憶することを、プロセッサに更に実行させる。このような実施例において、命令は、プロセッサによる実行時に、第３のバッファのメッセージ部分をデコードして、第２の監視対象デバイスの完全な第３のメッセージを生成することを、プロセッサに更に実行させる。

【００７８】

いくつかの実施例において、受信されたデータストリームのそれぞれのメッセージ部分は、受信されたデータストリームから解析されたチャンク識別番号に対応する場所で、対応のバッファ内に記憶される。他の実施例において、受信されたデータストリームのそれ

ぞれのメッセージ部分は、連続的な方法で対応のバッファ内に記憶される。

【００７９】

いくつかの実施例において、チャンクサイズ識別子は、受信されたデータストリームのメッセージ部分のサイズに対応する。いくつかの実施例において、チャンクサイズ識別子は、少なくとも１６ビットである。いくつかの実施例において、チャンクサイズ識別子は、少なくとも３２ビットである。いくつかの実施例において、合計チャンク識別子は、少なくとも１６ビットである。いくつかの実施例において、合計チャンク識別子は、少なくとも３２ビットである。

【００８０】

いくつかの実施例において、命令は、プロセッサによる実行時に、それぞれの新規メッセージについて、新規メッセージに対応するデータストリームから解析された、チャンクサイズ識別子及び合計チャンク識別子に基づいて、メモリ内の対応のバッファを割り当てることを、プロセッサに更に実行させる。

【００８１】

いくつかの実施例において、それぞれの対応のバッファのメッセージ部分は、バッファが満杯になるとデコードされる。他の実施例において、命令は、プロセッサによる実行時に、対応のバッファの割り当てに使用された合計チャンク識別子に基づき、それぞれの対応バッファについて、記憶されたメッセージ部分（例えば、バイナリ文字列又はアレイ）のリストを作製することと、記憶されたメッセージ部分に関連付けられたチャンク識別番号に基づき、対応のバッファに記憶されたそれぞれのメッセージ部分について、記憶されたメッセージ部分のリストを更新することと、リストが完了したときに、対応のバッファの記憶されたメッセージ部分をデコードすることと、をプロセッサに実行させる。

【００８２】

いくつかの実施例において、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子は、所定のデータストリームのヘッダ部分を集合的に形成する。いくつかの実施例において、それぞれのデータストリームは、ヘッダメッセージを含む。ヘッダメッセージは、マルチパートヘッダ部分の存在を示す、マルチパートヘッダフィールドを含み得る。このマルチパートヘッダ部分は、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を含み得る。いくつかの実施例において、マルチパートヘッダ部分は、受信されたデータストリームのメッセージ部分に対応する、発信元のデータを有するコンピューティングデバイスに関連付けられた名前識別子を含む。

【００８３】

いくつかの実施例において、命令は、プロセッサによる実行時に、第１のバッファでの第１のメッセージのデコード時に、成功メッセージに対応するデータストリームが、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、データストリームを送信したコンピューティングデバイスに送信されるようにすることを、プロセッサに更に実行させる。

【００８４】

別の態様では、本開示は、コンピュータ実装型の方法を説明する。方法は、複数のコンピューティングデバイスから複数のデータストリームを、コンピューティングデバイスのポートで受信することを含み、複数のデータストリームのそれぞれのデータストリームは、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子（例えば、チャンクの合計数）を含む。複数のデータストリームは、複数の持続的ステートレス接続（例えば、複数のＷｅｂＳｏｃｋｅｔ接続）を介して、受信され得る。方法は、コンピューティングデバイスのプロセッサによって、複数のデータストリームの第１のデータストリームを解析して、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを更に含む。方法は、プロセッサによって、コンピューティングデバイスのメモリ内で、第１のバッファを割り当てることを更に含み、第１のバッファは、第１の監視対象デバイスの第１のメッセージに関連付けられ、かつ、部分的に、第１のデータストリームから解

析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。方法は、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、プロセッサによって、第１のバッファに受信されたデータストリームのメッセージ部分を記憶することを更に含む。方法は、第１のバッファのメッセージ部分をデコードして、第１の監視対象デバイスの完全な第１のメッセージを生成することを、更に含む。

【００８５】

いくつかの実施例において、方法は、プロセッサよって、複数のデータストリームの第２のデータストリームを解析して、第１の監視対象デバイスに関連付けられた第２のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを更に含む。このような実施例において、方法は、プロセッサによって、メモリ内の第２のバッファを割り当てることを更に含み、第２のバッファは、第１の監視対象デバイスの第２のメッセージに関連付けられ、かつ、部分的に、第２のデータストリームから解析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。このような実施例において、方法は、第１の監視対象デバイスに関連付けられた第２のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、プロセッサによって、第２のバッファ内に、受信されたデータストリームのメッセージ部分を記憶することを更に含む。このような実施例において、方法は、プロセッサによって、第２のバッファのメッセージ部分をデコードして、第１の監視対象デバイスの完全な第２のメッセージを生成することを更に含む。

【００８６】

いくつかの実施例において、方法は、プロセッサによって、複数のデータストリームの第３のデータストリームを解析して、第２の監視対象デバイスに関連付けられた第３のメッセージに関連付けられた、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を決定することを更に含む。このような実施例において、方法は、プロセッサによって、メモリ内の第３のバッファを割り当てることを更に含み、この第３のバッファは、第２の監視対象デバイスの第３のメッセージに関連付けられ、かつ、部分的に、第３のデータストリームから解析されたチャンクサイズ識別子及び合計チャンク識別子によって定義されたサイズを有する。このような実施例において、方法は、第２の監視対象デバイスに関連付けられた第３のメッセージに関連付けられた、それぞれの受信されたデータストリームについて、プロセッサによって、第３のバッファ内に、受信されたデータストリームのメッセージ部分を記憶することを更に含む。このような実施例において、方法は、プロセッサによって、第３のバッファのメッセージ部分をデコードして、第２の監視対象デバイスの完全な第３のメッセージを生成することを更に含む。

【００８７】

いくつかの実施例において、受信されたデータストリームのそれぞれのメッセージ部分は、受信されたデータストリームから解析されたチャンク識別番号に対応する場所で、対応のバッファ内に記憶される。他の実施例において、受信されたデータストリームのそれぞれのメッセージ部分は、連続的な方法で対応のバッファ内に記憶される。

【００８８】

いくつかの実施例において、チャンクサイズ識別子は、受信されたデータストリームのメッセージ部分のサイズに対応する。いくつかの実施例において、チャンクサイズ識別子は、少なくとも１６ビットである。いくつかの実施例において、チャンクサイズ識別子は、少なくとも３２ビットである。

【００８９】

いくつかの実施例において、方法は、プロセッサによって、それぞれの新規メッセージについて、新規メッセージに対応するデータストリームから解析された、チャンクサイズ識別子及び合計チャンク識別子に基づいてメモリ内の対応のバッファを割り当てることを更に含む。いくつかの実施例において、合計チャンク識別子は、少なくとも１６ビットである。いくつかの実施例において、合計チャンク識別子は、少なくとも３２ビットである

。
【００９０】
いくつかの実施例において、それぞれの対応のバッファのメッセージ部分は、バッファが満杯になるとデコードされる。他の実施例において、方法は、プロセッサによって、対応のバッファの割り当てに使用された合計チャンク識別子に基づき、それぞれの対応のバッファについて、記憶されたメッセージ部分（例えば、バイナリストリング又はアレイ）のリストを作製することと、プロセッサによって、記憶されたメッセージ部分に関連付けられたチャンク識別番号に基づき、対応のバッファに記憶されたそれぞれのメッセージ部分について、記憶されたメッセージ部分のリストを更新することと、プロセッサによって、リストが完了したときに、対応のバッファの記憶されたメッセージ部分をデコードすることと、を更に含む。
【００９１】
いくつかの実施例において、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子は、所定のデータストリームのヘッダ部分を集合的に形成する。それぞれのデータストリームは、ヘッダメッセージを含み得、ヘッダメッセージは、マルチパートヘッダ部分の存在を示すマルチパートヘッダフィールドを含み、マルチパートヘッダ部分は、チャンク識別番号、チャンクサイズ識別子、及び合計チャンク識別子を含む。マルチパートヘッダ部分は、受信されたデータストリームのメッセージ部分に対応する、発信元のデータを有するコンピューティングデバイスに関連付けられた名前識別子を含み得る。
【００９２】
いくつかの実施例において、方法は、第１のバッファでの第１のメッセージのデコード時に、プロセッサによって、成功メッセージに対応するデータストリームが、第１の監視対象デバイスに関連付けられた第１のメッセージに関連付けられた、データストリームを送信したコンピューティングデバイスに送信されるようにすることを更に含む。
【００９３】
いくつかの実施例において、本開示は、メッセージを表現するバイナリデータストリームを受信するように構成されたポートを含む、通信デバイスを説明する。データストリームのそれぞれは、バイナリヘッダ及びバイナリ本体を含む。バイナリ本体は、メタデータ構成体及びメッセージ構成体を含む。メッセージ構成体は、データ値の１つ又は２つ以上の群を含み、メタデータ構成体は、メッセージ構成体を形成する１つ又は２つ以上のデータ値の所定のデータ値に対応する、記述値の１つ又は２つ以上の群を含む。通信デバイスは、バイナリヘッダ（例えば、バイナリＨＴＴＰ　ＡＰＩ要求／応答）及びバイナリ本体（例えば、動的ＲＥＳＴ　ＡＰＩメッセージ）の第１の構造記述と、メタデータ構成体（例えば、メタデータ構成体の下層組織及びプロパティの記述を有するモデル／情報モデル）の第２の構造記述と、を記憶するメモリを含む。通信デバイスは、受信されたバイナリデータストリームを解析して、受信されたメッセージを作製するためのプロセッサを含む。プロセッサは、第１の構造記述を使用して受信されたバイナリデータストリームを解析して、バイナリヘッダ及びバイナリ本体を決定する。プロセッサは、第２の構造記述を使用してバイナリ本体を解析して、メタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定する。プロセッサは、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値の１つ又は２つ以上の群を決定する。
【００９４】
いくつかの実施例において、メッセージ構成体の１つ又は２つ以上のデータ値のそれぞれと、メタデータ構成体の１つ又は２つ以上の記述値のそれぞれは、バイナリマーカーによって詳述される。バイナリマーカーは、１バイト長であってよい。バイナリマーカーは、（ｉ）データ値セット又は記述値セットの始まりを定義する第１の値と、（ｉｉ）メタデータ構成体及びメッセージ構成体のそれぞれの終わりを定義する第２の値と、を含み得る。メッセージ構成体全体は、バイナリマーカーでのみ区切られ得る。いくつかの実施例において、記述値からなる１つ又は２つ以上の群のそれぞれは、データ値名記述子、データ値記述記述子、及びデータ値型記述子を有する、メタデータ構成体を形成し得る。記述

値の群は、メッセージ構成体内のそれぞれのデータ値に対する定義を、集合的に形成し得る。メタデータ構成体は、バイナリ本体の形成において、メッセージ構成体に先行し得る。いくつかの実施例において、メタデータ構成体の一部及びメッセージ構成体の一部は、世界基準コードによって定義される一組の文字に対応する。いくつかの実施例において、バイナリヘッダは暗号化されず、バイナリ本体は暗号化される。

【００９５】

いくつかの実施例において、本開示は、バイナリ動的ＲＥＳＴメッセージを使用する方法を説明する。方法は、（ｉ）バイナリヘッダ及びバイナリ本体の第１の構造記述と、（ｉｉ）メタデータ構成体の第２の構造記述と、をメモリで記憶することを含む。メタデータ構成体は、メッセージ構成体を形成する１つ又は２つ以上のデータ値の所定のデータ値に対応する、記述値の１つ又は２つ以上の群を含む。方法は、複数のメッセージを表現する複数のバイナリデータストリームを、ポートで受信することを含む。方法はまた、プロセッサを使用し、第１の構造記述を使用して受信されたバイナリデータストリームのそれぞれを解析して、バイナリヘッダ及びバイナリ本体を決定することを含む。方法はまた、プロセッサを使用し、第２の構造記述を使用して解析されたバイナリ本体を解析して、メタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定することを含む。プロセッサは、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値の１つ又は２つ以上の群を決定する。

【００９６】

バイナリデータストリームは、ＷｅｂＳｏｃｋｅｔを介して受信され得る。メッセージ構成体の１つ又は２つ以上のデータ値のそれぞれと、メタデータ構成体の１つ又は２つ以上の記述値のそれぞれと、はバイナリマーカーによって詳述され得る。いくつかの実施形態において、バイナリヘッダは暗号化されず、バイナリ本体は暗号化される。メタデータ構成体は、データ値名前記述子、データ値記述記述子、及びデータ値型記述子を含み得る。メタデータ構成体は、バイナリ本体の形成において、メッセージ構成体に先行し得る。メタデータ構成体の一部及びメッセージ構成体の一部は、世界基準コードによって定義される一組の文字に対応し得る。

【００９７】

いくつかの実施例において、本開示は、そこに記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明するものであり、命令は、プロセッサによる実行時に、バイナリヘッダ及びバイナリ本体の第１の構造記述と、メタデータ構成体の第２の構造記述と、をメモリで記憶することを、プロセッサに実行させる。メタデータ構成体は、メッセージ構成体を形成する１つ又は２つ以上のデータ値の所定のデータ値に対応する、記述値の１つ又は２つ以上の群を含む。命令は、実行されると、複数のメッセージを表現する複数のバイナリデータストリームを、ポートで受信することを、プロセッサに更に実行させる。命令は、実行されると、第１の構造記述を使用して受信されたバイナリデータストリームのそれぞれを解析して、バイナリヘッダ及びバイナリ本体を決定することを、プロセッサに更に実行させる。命令は、実行されると、第２の構造記述を使用して解析されたバイナリ本体を解析して、メタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定することを、プロセッサに更に実行させる。プロセッサは、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値の１つ又は２つ以上の群を決定する。

【００９８】

いくつかの実施例において、メッセージ構成体の１つ又は２つ以上のデータ値のそれぞれと、メタデータ構成体の１つ又は２つ以上の記述値のそれぞれは、バイナリマーカーによって詳述される。いくつかの実施例において、バイナリデータストリームは、ＷｅｂＳｏｃｋｅｔを介して受信される。いくつかの実施例において、バイナリヘッダは暗号化されず、バイナリ本体は暗号化される。メタデータ構成体は、データ値名前記述子、データ値記述記述子、及びデータ値型記述子を含み得る。メタデータ構成体は、バイナリ本体の形成において、メッセージ構成体に先行し得る。メタデータ構成体の一部及びメッセージ

構成体の一部は、世界基準コードによって定義される一組の文字に対応し得る。

【００９９】

いくつかの実施例において、本開示は、自己記述型スキーマによって定義されるメッセージの構造記述を記憶するためのメモリを有する、通信デバイスを説明する。構造記述は、所定のメッセージを記述するためのデータオブジェクト間の、１つ又は２つ以上のラベル要素及び１つ又は２つ以上の句読点要素の両方によって点在する構造を記述する。メモリは、バイナリ表現に対応する、一組のバイナリシンボルを有するコードブックを更に記憶する。通信デバイスは、通信ポートを介して、別のコンピュータデバイスに送信するバイナリデータストリームを生成するように構成された、プロセッサを含む。

【０１００】

プロセッサは、自己記述型スキーマによって定義された受信されたメッセージから、バイナリデータストリームを生成することができ、受信されたメッセージは、１つ又は２つ以上のラベル要素及び１つ又は２つ以上の句読点要素の両方によって区切られた、１つ又は２つ以上のオブジェクトを含み得る。プロセッサは、記憶された構造記述に従い、１つ又は２つ以上のデータオブジェクトに関して受信されたメッセージを解析し得る。プロセッサは、コードブックに定義されたバイナリシンボルに、解析されたデータオブジェクトの要素のそれぞれを逐次的にマップして、バイナリデータストリームを作製し得る。解析したデータオブジェクトの要素のそれぞれを連続的にマップするとき、プロセッサは、データオブジェクト間に点在する１つ又は２つ以上のラベル要素及び１つ又は２つ以上の句読点要素を、バイナリデータストリームにマップしない。

【０１０１】

別の態様では、本開示は、持続的ステートレス接続を介してデータストリームを送信及び受信するように構成されたポートを含む、通信デバイスを説明する。持続的ステートレス接続は、ＷｅｂＳｏｃｋｅｔ接続を経由し得る。通信デバイスはまた、動的ＲＥＳＴ　ＡＰＩモデルを記憶するメモリを含む。通信デバイスはまた、持続的ステートレス接続を介して第２の通信デバイスに接続するように構成された、プロセッサを含む。プロセッサは、記憶された動的ＲＥＳＴ　ＡＰＩモデルで要求メッセージをフォーマット化して、自己記述型要求メッセージを作製する。プロセッサは、自己記述型要求メッセージが、ポートを介して送信されるようにする。

【０１０２】

動的ＲＥＳＴ　ＡＰＩモデルは、データオブジェクト、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含むデータモデルを含み得る。ポートは、値８０又は４４３を有する、ＨＴＴＰ又はＨＴＴＰＳポートであり得る。自己記述型要求メッセージは、ＪａｖａＳｃｒｉｐｔ（登録商標）　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトの少なくとも１つで表示され得る。自己記述型要求メッセージは、バイナリフォーマットでフォーマット化され得る。自己記述型要求メッセージは、暗号化され得る。

【０１０３】

いくつかの実施例において、本開示は、ＷｅｂＳｏｃｋｅｔで動的ＲＥＳＴメッセージを使用する方法を説明する。方法は、バイナリ動的ＲＥＳＴ　ＡＰＩモデルを、デバイスのメモリで記憶することを含む。方法はまた、デバイスのプロセッサを介して、バイナリ動的ＲＥＳＴ　ＡＰＩモデルを使用して要求メッセージをフォーマット化して、自己記述型要求メッセージを作製することを含む。方法はまた、デバイスのポートを介して、持続的ステート接続（persistent state connection）（例えば、ＷｅｂＳｏｃｋｅｔ接続）で要求メッセージを送信することを含む。

【０１０４】

動的ＲＥＳＴ　ＡＰＩモデルは、データオブジェクト、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含むデータモデルを含み得る。ポートは、値８０又は４４３を有する、ＨＴＴＰ又はＨＴＴＰＳポートであり

得る。自己記述型要求メッセージは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトの少なくとも１つで表示され得る。自己記述型要求メッセージは、バイナリフォーマットでフォーマット化され得る。自己記述型要求メッセージは、暗号化され得る。

【０１０５】

いくつかの実施例において、本開示は、そこに記憶された命令及びバイナリ動的ＲＥＳＴ ＡＰＩモデルを有する、非一過性コンピュータ読み取り可能媒体を説明する。命令は、実行されると、自己記述型要求メッセージを生成するため、バイナリ動的ＲＥＳＴ ＡＰＩモデルを使用して要求メッセージをフォーマット化することを、プロセッサに更に実行させる。命令は、実行されると、デバイスのポートを介して、持続的ステート接続で要求メッセージを送信することを、プロセッサに更に実行させる。動的ＲＥＳＴ ＡＰＩモデルは、データオブジェクト、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含むデータモデルを含み得る。プロセッサは、自己記述型要求メッセージが、ポートを介して送信されるようにする。

【０１０６】

ポートは、値８０又は４４３を有する、ＨＴＴＰ又はＨＴＴＰＳポートであり得る。自己記述型要求メッセージは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトの少なくとも１つで表示され得る。自己記述型要求メッセージは、バイナリフォーマットでフォーマット化され得る。自己記述型要求メッセージは、暗号化され得る。

【０１０７】

別の態様では、本開示は、自己記述型メッセージを使用して、通信プロトコルを抽出する方法を説明する。方法は、デバイスのメモリ内に第１の通信プロトコル及び第２の通信プロトコルを提供することを含み、第１の通信プロトコルは、自己記述型スキーマを有する多数の制御コード及びバイナリメッセージを含む。自己記述型スキーマは、第２の通信プロトコルのインスタンスが、第１の通信プロトコルの（例えば、デバイスの持続的メモリ内の）インスタンスに関連付けてマップされる、データオブジェクトモデルに関連付けられる。方法はまた、第２の通信プロトコルにおけるメッセージの第１のインスタンス化を（例えば、無線接続を介して）、コンピューティングデバイスのポートで受信することを含む。方法はまた、コンピューティングデバイスのプロセッサによって、メッセージの第１のインスタンス化を、制御メッセージ又はデータメッセージのいずれかとして分類することを含む。方法はまた、コンピューティングデバイスのプロセッサによって、メッセージの第１のインスタンス化を変換して、メッセージの第２のインスタンス化を作製することを含み、そのメッセージが制御メッセージと分類されたとき、プロセッサは、制御メッセージを複数の制御コードからなる１つ又は２つ以上の制御コードにマップし、また、そのメッセージがデータメッセージと分類されたとき、プロセッサは、自己記述型スキーマを有する対応するバイナリメッセージにデータメッセージをマップする。方法はまた、メッセージの第２のインスタンス化を（例えば、ワイヤレス接続を介して）、コンピューティングデバイスのポートで送信することを含む。

【０１０８】

いくつかの実施例において、制御コードの関連マッピングは、ＨＴＭＬフレームワークに基づく。いくつかの実施例において、データメッセージの関連マッピングは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択される、サービス要求に基づく。無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－

５０、ＯＭＡ　ＬＷ　Ｍ２Ｍ、並びにＥＴＳＩ　Ｍ２Ｍを含み得る。データオブジェクトモデルは、動的ＲＥＳＴ　ＡＰＩオブジェクトモデルで構造化され得る。

【０１０９】

別の態様では、本開示は、プロセッサ及びメモリを含むシステムを説明するものであり、メモリは、第１の通信プロトコル及び第２の通信プロトコルを記憶し、第１の通信プロトコルは、自己記述型スキーマを有する多数の制御コード及びバイナリメッセージを含む。自己記述型スキーマは、第２の通信プロトコルのインスタンスが、第１の通信プロトコルのインスタンスに関連付けてマップされる、データオブジェクトモデルに関連付けられる。第１の通信プロトコルと第２の通信プロトコルとの間の関連マッピングは、持続的メモリに記憶され得る。メモリは命令を含み、命令は、プロセッサによる実行時に、第２の通信プロトコルにおけるメッセージの第１のインスタンス化を（例えば、無線接続を介して）、コンピューティングデバイスのポートで受信することを、プロセッサに更に実行させる。命令は、プロセッサによる実行時に、メッセージの第１のインスタンスを制御メッセージ又はデータメッセージのいずれかとして分類することを、プロセッサに更に実行させる。命令は、プロセッサによる実行時に、メッセージの第１のインスタンス化を変換して、メッセージの第２のインスタンス化を作製することを、プロセッサに更に実行させる。メッセージが制御メッセージと分類されたとき、プロセッサは、制御メッセージを複数の制御コードからなる１つ又は２つ以上の制御コードにマップし、また、メッセージがデータメッセージと分類されたとき、プロセッサは、自己記述型スキーマを有する対応するバイナリメッセージにデータメッセージをマップする。方法は、メッセージの第２のインスタンス化を、コンピューティングデバイスのポートで送信することを含む。

【０１１０】

第１の通信デバイスは、ＷｅｂＳｏｃｋｅｔ接続を介して、メッセージの第２のインスタンス化をプラットフォームサーバに送信し得る。いくつかの実施例において、無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ　ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－５０、ＯＭＡ　ＬＷ　Ｍ２Ｍ、並びにＥＴＳＩ　Ｍ２Ｍを含む。データオブジェクトモデルは、動的ＲＥＳＴ　ＡＰＩオブジェクトモデルで構造化され得る。いくつかの実施例において、制御コードの関連マッピングは、ＨＴＭＬフレームワークに基づく。いくつかの実施例において、データメッセージの関連マッピングは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択される、サービス要求に基づく。

【０１１１】

別の態様では、本開示は、第１の通信プロトコル及び第２の通信プロトコルに関する命令を有する、非一過性コンピュータ読み取り可能媒体を説明する。第１の通信プロトコルは、自己記述型スキーマを有する多数の制御コード及びバイナリメッセージを含む。自己記述型スキーマは、第２の通信プロトコルのインスタンスが、第１の通信プロトコルのインスタンスに関連付けてマップされる、データオブジェクトモデルに関連付けられる。メモリは命令を更に含み、命令は、プロセッサによる実行時に、第２の通信プロトコルにおけるメッセージの第１のインスタンス化を（例えば、ワイヤレス接続を介して）、コンピューティングデバイスのポートで受信することを、プロセッサに実行させる。命令は、プロセッサによる実行時に、メッセージの第１のインスタンスを制御メッセージ又はデータメッセージのいずれかとして分類することを、プロセッサに更に実行させる。命令は、プロセッサによる実行時に、メッセージの第１のインスタンス化をメッセージの第２のインスタンス化にマップすることを、プロセッサに更に実行させる。メッセージが制御メッセージと分類されたとき、プロセッサは、制御メッセージを複数の制御コードからなる１つ又は２つ以上の制御コードにマップし、また、ｉｉ）メッセージがデータメッセージと分類されたとき、プロセッサは、自己記述型スキーマを有する対応するバイナリメッセージ

にデータメッセージをマップする。命令は、プロセッサによる実行時に、メッセージの第２のインスタンス化を、コンピューティングデバイスのポートで送信することを、プロセッサに更に実行させる。

【０１１２】

第１の通信デバイスは、ＷｅｂＳｏｃｋｅｔ接続を介して、メッセージの第２のインスタンス化をプラットフォームサーバに送信し得る。第１の通信プロトコルと第２の通信プロトコルとの間の関連マッピングは、持続的メモリに逐次的に記憶され得る。いくつかの実施例において、無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ　ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－５０、ＯＭＡ　ＬＷ　Ｍ２Ｍ、並びにＥＴＳＩ　Ｍ２Ｍを含む。データオブジェクトモデルは、動的ＲＥＳＴ　ＡＰＩオブジェクトモデルで構造化され得る。いくつかの実施例において、制御コードの関連マッピングは、ＨＴＭＬフレームワークに基づいてもよい。いくつかの実施例において、データメッセージの関連マッピングは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択される、サービス要求に基づく。

【０１１３】

いくつかの実施例において、本開示は、バイナリ動的ＲＥＳＴメッセージのチャンクベースの通信を実行するための、通信エンジンを説明する。通信エンジンは、第１のデータストリーム及び第２のデータストリームを含む、１つ又は２つ以上のデータストリームを受信するように構成された通信プロトコルを含み、データストリームは、バイナリ動的ＲＥＳＴメッセージを含む。データストリームは、１つ又は２つ以上のデータフィールドを含み、データフィールドのそれぞれは、フィールド値、及びフィールド値の長さ識別子を含み、長さ識別子はデータフィールドのそれぞれにおいて対応するフィールド値に先行する。通信エンジンはまた、受信された第１のデータストリーム及び受信された第２のデータストリームを記憶するための、第１のバッファを含む。通信エンジンはまた、デコードされたメッセージの部分を記憶するための、第２のバッファを含む。通信エンジンはまた、第１のバッファ内にバッファリングされた、受信されたデータストリームをデコードして、所定のデコードされた部分を作製するための、プロセッサを含む。プロセッサは、第２のバッファに所定のデコードされた部分を記憶する。プロセッサは、受信された第２のデータストリームを完全に受信する前に、第１のバッファ内にバッファリングされた、受信された第１のデータストリームのデコードを開始する。

【０１１４】

通信ポートは、ＷｅｂＳｏｃｋｅｔ接続を介して、１つ又は２つ以上の入力データストリームを受信し得る。メッセージは、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを有する、データオブジェクトを含み得る。１つ又は２つ以上の入力データストリームのそれぞれは、所定の完全なメッセージ（例えば、実行可能メッセージ）を形成する、多数の１つ又は２つ以上のデータストリームの識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全なメッセージ内における、所定のデコードされたデータストリームの場所を示す、メッセージ番号識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全なメッセージの一部を形成し得る。いくつかの実施例において、通信エンジンは、入力データストリームの暗号化された部分をデコードするための、復号エンジンを含む。

【０１１５】

いくつかの実施例において、本開示は、バイナリ動的ＲＥＳＴメッセージのチャンクベースの通信を使用する方法を説明する。方法は、第１のデータストリーム及び第２のデータストリームを含む１つ又は２つ以上のデータストリームを、コンピューティングデバイスのポートで受信することを含む。それぞれのデータストリームは、１つ又は２つ以上の

データフィールドを含み得、データフィールドのそれぞれは、フィールド値、及びフィールド値の長さ識別子を含む。長さ識別子は、データフィールドのそれぞれにおいて対応するフィールド値に先行する。方法はまた、受信された第１のデータストリーム及び受信された第２のデータストリームを、第１のバッファで記憶することを含む。方法はまた、デコードされたメッセージの部分を、第２のバッファで記憶することを含む。方法はまた、プロセッサを使用し、第１のバッファ内にバッファリングされた、受信されたデータストリームをデコードして、所定のデコードされた部分を作製することを含み、その場合、プロセッサは、受信された第２のデータストリームを完全に受信する前に、第１のバッファ内にバッファリングされた受信された第１のデータストリームのデコードを開始する。

【０１１６】

通信ポートは、ＷｅｂＳｏｃｋｅｔ接続を介して、１つ又は２つ以上の入力データストリームを受信し得る。メッセージは、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを有する、データオブジェクトを含み得る。１つ又は２つ以上の入力データストリームのそれぞれは、所定の完全なメッセージ（例えば、実行可能メッセージ）を形成する、多数の１つ又は２つ以上のデータストリームの識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全なメッセージ内の他のデコードされたメッセージ部分に関して、所定のデコードされたメッセージ部分の場所を示す、メッセージ番号識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全なメッセージの一部を形成し得る。

【０１１７】

いくつかの実施例において、本開示は、そこに記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明するものであり、命令は、プロセッサによる実行時に、１つ又は２つ以上のバイナリ動的ＲＥＳＴメッセージの部分を有する、第１のデータストリーム及び第２のデータストリームを含む、１つ又は２つ以上のデータストリームを、コンピューティングデバイスのポートで受信することを、プロセッサに実行させる。命令は、プロセッサによる実行時に、受信された第１のデータストリーム及び受信された第２のデータストリームを、コンピューティングデバイスの第１のバッファで記憶することを、プロセッサに更に実行させる。命令は、実行されると、デコードされたメッセージの部分を記憶することを、コンピューティングデバイスの第２のバッファでプロセッサに更に実行させる。プロセッサは、第１のバッファ内にバッファリングされた、受信されたデータストリームをデコードして、所定のデコードされた部分を作製し、その場合、プロセッサは、受信された第２のデータストリームを完全に受信する前に、第１のバッファ内にバッファリングされた、受信された第１のデータストリームのデコードを開始する。

【０１１８】

データストリームは、１つ又は２つ以上のデータフィールドを含み得、データフィールドのそれぞれは、フィールド値、及びフィールド値の長さ識別子を含む。長さ識別子は、データフィールドのそれぞれにおいて対応するフィールド値に先行する。通信ポートは、ＷｅｂＳｏｃｋｅｔ接続を介して、１つ又は２つ以上の入力データストリームを受信し得る。メッセージは、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを有する、データオブジェクトを含み得る。１つ又は２つ以上の入力データストリームのそれぞれは、所定の完全なメッセージ（例えば、実行可能メッセージ）を形成する、多数の１つ又は２つ以上のデータストリームの識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全なメッセージ内の他のデコードされたメッセージ部分に関して、所定のデコードされたメッセージ部分の場所を示す、メッセージ番号識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全なメッセージの一部を形成し得る。

【０１１９】

いくつかの実施例において、本開示は、メッセージを表現するバイナリデータストリームを受信するように構成されたポートを含む、通信デバイスを説明する。データストリームのそれぞれは、バイナリヘッダ及びバイナリ本体を含み得る。バイナリ本体は、メタデ

ータ構成体及びメッセージ構成体を有し得、メッセージ構成体は、データ値の１つ又は２つ以上の群を含む。メタデータ構成体は、メッセージ構成体を形成する１つ又は２つ以上のデータ値の所定のデータ値に対応する、記述値の１つ又は２つ以上の群を含み得る。いくつかの実施例において、通信デバイスは、バイナリヘッダ及びバイナリ本体の第１の構造記述を記憶する、メモリを含む。メモリはまた、メタデータ構成体の第２の構造記述を記憶し得る。バイナリヘッダは暗号化されない場合があるが、バイナリ本体は暗号化される。いくつかの実施例において、通信デバイスはプロセッサを含み得る。プロセッサは、受信されたバイナリデータストリームを解析して、受信されたメッセージを複製し得る。プロセッサは、第１の構造記述を使用して受信バイナリデータストリームを解析して、バイナリヘッダ及びバイナリ本体を決定し得る。プロセッサは、第２の構造記述を使用してバイナリ本体を解析して、メタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定し得る。プロセッサは、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値の１つ又は２つ以上の群を決定し得る。いくつかの実施例において、メッセージ構成体の１つ又は２つ以上のデータ値のそれぞれと、メタデータ構成体の１つ又は２つ以上の記述値のそれぞれと、はバイナリマーカーによって詳述され得る。バイナリマーカーは、１バイト長であってよい。バイナリマーカーは、データ値セット又は記述値セットのそれぞれの始まりを定義する第１の値と、メタデータ構成体及びメッセージ構成体のそれぞれの終わりを定義する第２の値と、を含み得る。メッセージ構成体全体は、バイナリマーカーでのみ区切られ得る。いくつかの実施例において、記述値からなる１つ又は２つ以上の群のそれぞれは、データ値名記述子、データ値記述記述子、及びデータ値型記述子を有する、メタデータ構成体を形成し得る。記述値の群は、メッセージ構成体内のデータ値のそれぞれに対する定義を、集合的に形成し得る。メタデータ構成体は、バイナリ本体の形成において、メッセージ構成体に先行し得る。いくつかの実施例において、メタデータ構成体の一部及びメッセージ構成体の一部は、世界基準コードによって定義される一組の文字に対応する。いくつかの実施例において、バイナリヘッダは暗号化されず、バイナリ本体は暗号化される。

【０１２０】

いくつかの実施例において、本開示は、バイナリ動的ＲＥＳＴメッセージを使用する方法を説明する。方法は、バイナリヘッダ及びバイナリ本体の第１の構造記述と、メタデータ構成体の第２の構造記述と、をメモリで記憶することを含み得、メタデータ構成体は、メッセージ構成体を形成する１つ又は２つ以上のデータ値の所定のデータ値に対応する、記述値の１つ又は２つ以上の群を含む。メッセージ構成体の１つ又は２つ以上のデータ値のそれぞれと、メタデータ構成体の１つ又は２つ以上の記述値のそれぞれと、はバイナリマーカーによって詳述され得る。いくつかの実施例において、方法は、複数のメッセージを表現する複数のバイナリデータストリームを、ポートで受信することを含み得る。バイナリデータストリームは、ＷｅｂＳｏｃｋｅｔを介して受信され得る。いくつかの実施例において、方法は、プロセッサを使用し、第１の構造記述を使用して受信されたバイナリデータストリームを解析して、バイナリヘッダ及びバイナリ本体を決定することを含み得る。バイナリヘッダは暗号化されず、バイナリ本体は暗号化される。いくつかの実施例において、方法は、プロセッサを使用し、第２の構造記述を使用して解析されたバイナリ本体を解析して、メタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定することを含み得、プロセッサは、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値の１つ又は２つ以上の群を決定する。メタデータ構成体は、データ値名前記述子、データ値記述記述子、及びデータ値型記述子を含み得る。メタデータ構成体は、バイナリ本体の形成において、メッセージ構成体に先行し得る。メタデータ構成体の一部及びメッセージ構成体の一部は、世界基準コードによって定義される一組の文字に対応し得る。

【０１２１】

いくつかの実施例において、本開示は、そこに記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明するものであり、命令は、プロセッサによる実行時に、

バイナリヘッダ及びバイナリ本体の第１の構造記述と、メタデータ構成体の第２の構造記述と、をメモリで記憶することをプロセッサにさせ、メタデータ構成体は、メッセージ構成体を形成する１つ又は２つ以上のデータ値の所定のデータ値に対応する、記述値の１つ又は２つ以上の群を含む。メッセージ構成体の１つ又は２つ以上のデータ値のそれぞれと、メタデータ構成体の１つ又は２つ以上の記述値のそれぞれと、はバイナリマーカーによって詳述され得る。いくつかの実施例において、命令は、実行されると、複数のメッセージを表現する複数のバイナリデータストリームを、ポートで受信することを、プロセッサに更に実行させる。バイナリデータストリームは、ＷｅｂＳｏｃｋｅｔを介して受信され得る。いくつかの実施例において、命令は、実行されると、プロセッサを使用し、第１の構造記述を使用して受信されたバイナリデータストリームのそれぞれを解析して、バイナリヘッダ及びバイナリ本体を決定することを、プロセッサに更に実行させる。バイナリヘッダは暗号化されない場合があり、バイナリ本体は暗号化される。いくつかの実施例において、命令は、実行されると、プロセッサを使用し、第２の構造記述を使用して解析されたバイナリ本体を解析して、メタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定することをプロセッサに更にさせ、プロセッサは、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値の１つ又は２つ以上の群を決定する。メタデータ構成体は、データ値名前記述子、データ値記述記述子、及びデータ値型記述子を含み得る。メタデータ構成体は、バイナリ本体の形成において、メッセージ構成体に先行し得る。メタデータ構成体の一部及びメッセージ構成体の一部は、世界基準コードによって定義される一組の文字に対応し得る。

【０１２２】

いくつかの実施例において、本開示は、自己記述型スキーマによって定義される、メッセージの構造記述を記憶するためのメモリを有する、通信デバイスを説明するものであり、構造記述は、データオブジェクト間に１つ又は２つ以上のラベル要素及び１つ又は２つ以上の句読点要素によって点在する構造を記述して、所定のメッセージを記述する。メモリは、バイナリ表現に対応する、一組のバイナリシンボルを有するコードブックを更に記憶する。通信デバイスは、通信ポートを介して、別のコンピュータデバイスに送信するバイナリデータストリームを生成するように構成された、プロセッサを含み得る。プロセッサは、自己記述型スキーマによって定義された受信されたメッセージから、バイナリデータストリームを生成することができ、受信されたメッセージは、１つ又は２つ以上のラベル要素及び１つ又は２つ以上の句読点要素の両方によって区切られた、１つ又は２つ以上のオブジェクトを含み得る。プロセッサは、記憶された構造記述に従い、１つ又は２つ以上のデータオブジェクトに関して受信されたメッセージを解析し得る。プロセッサは、解析されたデータオブジェクトの要素のそれぞれを、コードブックに定義されたバイナリシンボルに逐次的にマップして、バイナリデータストリームを作製することができ、解析されたデータオブジェクトの要素のそれぞれを連続的にマップするとき、プロセッサは、データオブジェクト間に点在する１つ又は２つ以上のラベル要素及び１つ又は２つ以上の句読点要素をバイナリデータストリームにマップしない。

【０１２３】

いくつかの実施例において、本開示は、持続的ステートレス接続を介してデータストリームを送信及び受信するように構成されたポートを含む、通信デバイスを説明する。持続的ステートレス接続は、ＷｅｂＳｏｃｋｅｔ接続を経由し得る。通信デバイスは、動的ＲＥＳＴ　ＡＰＩモデルを記憶するメモリを含み得る。通信デバイスは、持続的ステートレス接続を介して第２の通信デバイスに接続するように構成された、プロセッサを含み得る。プロセッサは、記憶された動的ＲＥＳＴ　ＡＰＩモデルを使用して要求メッセージをフォーマット化して、自己記述型要求メッセージを作製し得る。動的ＲＥＳＴ　ＡＰＩモデルは、データオブジェクト、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含むデータモデルを含み得る。プロセッサは、自己記述型要求メッセージが、ポートを介して送信されるようにし得る。ポートは、値８０又は４４３を有する、ＨＴＴＰ又はＨＴＴＰＳポートであり得る。自己記述型要求メッセ

ージは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトの少なくとも１つで表示され得る。自己記述型要求メッセージは、バイナリフォーマットでフォーマット化され得る。自己記述型要求メッセージは、暗号化され得る。

【０１２４】

いくつかの実施例において、本開示は、ＷｅｂＳｏｃｋｅｔで動的ＲＥＳＴメッセージを使用する方法を説明する。方法は、バイナリ動的ＲＥＳＴ ＡＰＩモデルを、デバイスのメモリで記憶することを含み得る。方法はまた、デバイスのプロセッサによって、バイナリ動的ＲＥＳＴ ＡＰＩモデルを使用して要求メッセージをフォーマット化して、自己記述型要求メッセージを作製することを含み得る。方法は、デバイスのポートを介して、持続的ステート接続（persistent state connection）で要求メッセージを送信することを含み得る。動的ＲＥＳＴ ＡＰＩモデルは、データオブジェクト、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含むデータモデルを含み得る。プロセッサは、自己記述型要求メッセージが、ポートを介して送信されるようにする。ポートは、値８０又は４４３を有する、ＨＴＴＰ又はＨＴＴＰＳポートであり得る。自己記述型要求メッセージは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトの少なくとも１つで表示され得る。自己記述型要求メッセージは、バイナリフォーマットでフォーマット化され得る。自己記述型要求メッセージは、暗号化され得る。

【０１２５】

いくつかの実施例において、本開示は、そこに記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明するものであり、命令は、プロセッサによる実行時に、バイナリ動的ＲＥＳＴ ＡＰＩモデルを、デバイスのメモリで記憶することを、プロセッサに実行させる。命令は、実行されると、デバイスのプロセッサによって、バイナリ動的ＲＥＳＴ ＡＰＩモデルを使用して要求メッセージをフォーマット化して、自己記述型要求メッセージを作製することを、プロセッサに更に実行させる。命令は、実行されると、デバイスのポートを介して、持続的ステート接続で要求メッセージを送信することを、プロセッサに更に実行させる。動的ＲＥＳＴ ＡＰＩモデルは、データオブジェクト、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含むデータモデルを含み得る。プロセッサは、自己記述型要求メッセージが、ポートを介して送信されるようにする。ポートは、値８０又は４４３を有する、ＨＴＴＰ又はＨＴＴＰＳポートであり得る。自己記述型要求メッセージは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）オブジェクト及びＥｘｔｅｎｓｉｂｌｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（ＸＭＬ）オブジェクトの少なくとも１つで表示され得る。自己記述型要求メッセージは、バイナリフォーマットでフォーマット化され得る。自己記述型要求メッセージは、暗号化され得る。

【０１２６】

いくつかの実施例において、本開示は、自己記述型メッセージを使用して、通信プロトコルを抽出する方法を説明する。方法は、デバイスのメモリ内に第１の通信プロトコル及び第２の通信プロトコルを提供することを含み得、第１の通信プロトコルは、自己記述型スキーマを有する多数の制御コード及びバイナリメッセージを含む。自己記述型スキーマは、第２の通信プロトコルのインスタンスが、第１の通信プロトコルのインスタンスに関連付けてマップされる、データオブジェクトモデルであり得る。第１の通信プロトコルと第２の通信プロトコルとの間の関連マッピングは、持続的メモリに記憶され得る。いくつかの実施例において、制御コードの関連マッピングは、ＨＴＭＬフレームワークに基づいてもよい。いくつかの実施例において、データメッセージの関連マッピングは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択される、サービス要求に基づいてもよい。無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる

群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－５０、ＯＭＡ ＬＷ Ｍ２Ｍ、ＥＴＳＩ Ｍ２Ｍを含み得る。データオブジェクトモデルは、動的ＲＥＳＴ ＡＰＩオブジェクトモデルで構造化され得る。いくつかの実施例において、方法は、第２の通信プロトコルにおけるメッセージの第１のインスタンス化を、コンピューティングデバイスのポートで受信することを含み得る。メッセージは、無線接続を介して受信され得る。いくつかの実施例において、方法は、コンピューティングデバイスのプロセッサによって、メッセージの第１のインスタンス化を制御メッセージ又はデータメッセージのいずれかとして分類することを含み得る。いくつかの実施例において、方法は、コンピューティングデバイスのプロセッサによって、メッセージの第１のインスタンス化を変換して、メッセージの第２のインスタンス化を作製することを含み得、そのメッセージが制御メッセージと分類されたとき、プロセッサは、制御メッセージを複数の制御コードからなる１つ又は２つ以上の制御コードにマップし、また、そのメッセージがデータメッセージと分類されたとき、プロセッサは、自己記述型スキーマを有する、対応するバイナリメッセージにデータメッセージをマップする。いくつかの実施例において、方法は、メッセージの第２のインスタンス化をコンピューティングデバイスのポートで送信することを含み得る。第１の通信デバイスは、ＷｅｂＳｏｃｋｅｔ接続を介して、メッセージの第２のインスタンス化をプラットフォームサーバに送信し得る。

【０１２７】

いくつかの実施例において、本開示は、プロセッサ及びメモリを含むシステムを説明するものであり、メモリは命令を記憶し、命令は、実行されると、デバイスのメモリ内の第１の通信プロトコル及び第２の通信プロトコルを提供することを、プロセッサに実行させるものであり、第１の通信プロトコルは、自己記述型スキーマを有する多数の制御コード及びバイナリメッセージを含む。自己記述型スキーマは、第２の通信プロトコルのインスタンスが、第１の通信プロトコルのインスタンスに関連付けてマップされる、データオブジェクトモデルであり得る。第１の通信プロトコルと第２の通信プロトコルとの間の関連マッピングは、持続的メモリに記憶され得る。いくつかの実施例において、無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－５０、ＯＭＡ ＬＷ Ｍ２Ｍ、並びにＥＴＳＩ Ｍ２Ｍを含み得る。データオブジェクトモデルは、動的ＲＥＳＴ ＡＰＩオブジェクトモデルで構造化され得る。いくつかの実施例において、制御コードの関連マッピングは、ＨＴＭＬフレームワークに基づいてもよい。いくつかの実施例において、データメッセージの関連マッピングは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択される、サービス要求に基づいてもよい。いくつかの実施例において、命令は、実行されると、第２の通信プロトコルにおけるメッセージの第１のインスタンス化を、コンピューティングデバイスのポートで受信することを、プロセッサに更に実行させる。メッセージは、無線接続を介して受信され得る。いくつかの実施例において、命令は、実行されると、メッセージの第１のインスタンス化を制御メッセージ又はデータメッセージのいずれかとして分類することを、プロセッサに更に実行させる。いくつかの実施例において、命令は、実行されると、メッセージの第１のインスタンス化を変換して、メッセージの第２のインスタンス化を作製することをプロセッサに更に実行させ、そのメッセージが制御メッセージと分類されたとき、プロセッサは、制御メッセージを複数の制御コードからなる１つ又は２つ以上の制御コードにマップし、また、そのメッセージがデータメッセージと分類されたとき、プロセッサは、自己記述型スキーマを有する、対応するバイナリメッセージにデータメッセージをマップする。いくつかの実施例において、方法は、メッセージの第２のインスタンス化をコンピューティングデバイスの

ポートで送信することを含み得る。第１の通信デバイスは、ＷｅｂＳｏｃｋｅｔ接続を介して、メッセージの第２のインスタンス化をプラットフォームサーバに送信し得る。

【０１２８】

いくつかの実施例において、本開示は、そこに記憶された命令を有する非一過性コンピュータ読み取り可能媒体を説明するものであり、命令は、プロセッサによる実行時に、デバイスのメモリ内の第１の通信プロトコル及び第２の通信プロトコルを提供することをプロセッサにさせ、第１の通信プロトコルは、自己記述型スキーマを有する多数の制御コード及びバイナリメッセージを含み得る。自己記述型スキーマは、第２の通信プロトコルのインスタンスが、第１の通信プロトコルのインスタンスに関連付けてマップされる、データオブジェクトモデルであり得る。第１の通信プロトコルと第２の通信プロトコルとの間の関連マッピングは、持続的メモリに記憶され得る。いくつかの実施例において、無線接続は、Ｚｉｇｂｅｅ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、ＷｉＭａｘ、Ｗｉ－Ｆｉ、ＧＳＭ（登録商標）、ＰＣＳ、及びＤ－ＡＭＰＳからなる群から選択されるネットワーク、ＩＥＣ、６ＬｏＷＰＡＮ、Ａｎｔ、ＤＡＳＨ７、ＥｎＯｃｅａｎ、ＩＮＳＴＥＯＮ、ＮｅｕＲＦ ＯＮ、Ｓｅｎｃｅｉｖｅ、ＷｉｒｅｌｅｓｓＨＡＲＴ、Ｃｏｎｔｉｋｉ、ＴｉｎｙＯＳ、ＧＰＲＳ、ＴＣＰ／ＩＰ、ＣｏＡＰ、ＭＱＴＴ、ＴＲ－５０、ＯＭＡ ＬＷ Ｍ２Ｍ、並びにＥＴＳＩ Ｍ２Ｍを含み得る。データオブジェクトモデルは、動的ＲＥＳＴ ＡＰＩオブジェクトモデルで構造化され得る。いくつかの実施例において、制御コードの関連マッピングは、ＨＴＭＬフレームワークに基づいてもよい。いくつかの実施例において、データメッセージの関連マッピングは、ＧＥＴ要求、ＰＵＴ要求、ＰＯＳＴ要求、及びＤＥＬＥＴＥ要求からなる群から選択される、サービス要求に基づいてもよい。いくつかの実施例において、命令は、実行されると、第２の通信プロトコルにおけるメッセージの第１のインスタンス化を、コンピューティングデバイスのポートで受信することを、プロセッサに更に実行させる。メッセージは、無線接続を介して受信され得る。いくつかの実施例において、命令は、実行されると、メッセージの第１のインスタンス化を制御メッセージ又はデータメッセージのいずれかとして分類することを、プロセッサに更に実行させる。いくつかの実施例において、命令は、実行されると、メッセージの第１のインスタンス化を変換して、メッセージの第２のインスタンス化を作製することをプロセッサに更に実行させ、そのメッセージが制御メッセージと分類されたとき、プロセッサは、制御メッセージを複数の制御コードからなる１つ又は２つ以上の制御コードにマップし、また、そのメッセージがデータメッセージと分類されたとき、プロセッサは、自己記述型スキーマを有する、対応するバイナリメッセージにデータメッセージをマップする。いくつかの実施例において、方法は、メッセージの第２のインスタンス化をコンピューティングデバイスのポートで送信することを含み得る。第１の通信デバイスは、ＷｅｂＳｏｃｋｅｔ接続を介して、メッセージの第２のインスタンス化をプラットフォームサーバに送信し得る。

【０１２９】

いくつかの実施例において、本開示は、第１のデータストリーム及び第２のデータストリームを含む、１つ又は２つ以上のデータストリームを受信するように構成された通信プロトコルを含む、通信エンジンを説明するものであり、データストリームは、バイナリ動的ＲＥＳＴメッセージを含む。データストリームは、１つ又は２つ以上のデータフィールドを含み得、データフィールドのそれぞれは、フィールド値、及びフィールド値の長さ識別子を含み、長さ識別子はデータフィールドのそれぞれにおいて対応するフィールド値に先行する。通信ポートは、ＷｅｂＳｏｃｋｅｔ接続を介して、１つ又は２つ以上の入力データストリームを受信し得る。メッセージは、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを有する、データオブジェクトを含み得る。１つ又は２つ以上の入力データストリームのそれぞれは、所定の完全な実行可能メッセージを形成する、多数の１つ又は２つ以上のデータストリームの識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全な実行可能メッセージ内における、所定のデコードされたデータストリームの場所を示す、メッセージ番号識別子を

含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全な実行可能メッセージの一部を形成し得る。いくつかの実施例において、通信エンジンは、受信された第１のデータストリーム及び受信された第２のデータストリームを記憶するための、第１のバッファを含み得る。いくつかの実施例において、通信エンジンは、デコードされたメッセージの部分を記憶するための、第２のバッファを含み得る。いくつかの実施例において、通信エンジンは、第１のバッファ内にバッファリングされた、受信されたデータストリームをデコードして、所定のデコードされた部分を作製するように構成された、プロセッサを含み得る。プロセッサは、所定のデコードされた部分を第２のバッファに記憶し得、その場合、プロセッサは、受信された第２のデータストリームを完全に受信する前に、第１のバッファ内にバッファリングされた、受信された第１のデータストリームのデコードを開始する。いくつかの実施例において、通信エンジンは、第１のバッファにバッファリングされた１つ又は２つ以上のデータストリームのデコードされた部分を記憶するための、第２のバッファを含み得、第２のバッファが所定の完全な実行可能メッセージを形成する１つ又は２つ以上のデータストリームのそれぞれを含むとき、プロセッサは信号を送信する。いくつかの実施例において、通信エンジンは、暗号化される、入力データストリームの暗号化された部分をデコードするための復号エンジンを含み得る。

【０１３０】

いくつかの実施例において、本開示は、バイナリ動的ＲＥＳＴメッセージのチャンクベースの通信を使用する方法を説明する。方法は、ポートで、第１のデータストリーム及び第２のデータストリームを含む、１つ又は２つ以上のデータストリームを受信することを含み得る。データストリームは、１つ又は２つ以上のデータフィールドを含み得、データフィールドのそれぞれは、フィールド値、及びフィールド値の長さ識別子を含み、長さ識別子はデータフィールドのそれぞれにおいて対応するフィールド値に先行する。通信ポートは、ＷｅｂＳｏｃｋｅｔ接続を介して、１つ又は２つ以上の入力データストリームを受信し得る。メッセージは、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを有する、データオブジェクトを含み得る。１つ又は２つ以上の入力データストリームのそれぞれは、所定の完全な実行可能メッセージを形成する、多数の１つ又は２つ以上のデータストリームの識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全な実行可能メッセージ内における、所定のデコードされたデータストリームの場所を示す、メッセージ番号識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全な実行可能メッセージの一部を形成し得る。いくつかの実施例において、方法は、第１のバッファで、受信された第１のデータストリーム及び受信された第２のデータストリームを記憶することを含み得る。いくつかの実施例において、方法は、第２のバッファで、デコードされたメッセージの部分を記憶することを含み得る。いくつかの実施例において、方法は、プロセッサを使用し、第１のバッファ内にバッファリングされた受信されたデータストリームをデコードして、所定のデコードされた部分を作製することを含み得る。プロセッサは、受信された第２のデータストリームを完全に受信する前に、第１のバッファ内にバッファリングされた、受信された第１のデータストリームのデコードを開始する。

【０１３１】

いくつかの実施例において、本開示は、そこに記憶された命令を有する、非一過性コンピュータ読み取り可能媒体を説明するものであり、命令は、プロセッサによる実行時に、ポートで、第１のデータストリーム及び第２のデータストリームを含む、１つ又は２つ以上のデータストリームを受信することを、プロセッサに実行させる。データストリームは、１つ又は２つ以上のデータフィールドを含み得、データフィールドのそれぞれは、フィールド値、及びフィールド値の長さ識別子を含み、長さ識別子はデータフィールドのそれぞれにおいて対応するフィールド値に先行する。通信ポートは、ＷｅｂＳｏｃｋｅｔ接続を介して、１つ又は２つ以上の入力データストリームを受信し得る。メッセージは、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを有する、データオブジェクトを含み得る。１つ又は２つ以上の入力データストリー

ムのそれぞれは、所定の完全な実行可能メッセージを形成する、多数の１つ又は２つ以上のデータストリームの識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全な実行可能メッセージ内における、所定のデコードされたデータストリームの場所を示す、メッセージ番号識別子を含み得る。１つ又は２つ以上のデータストリームのそれぞれは、完全な実行可能メッセージの一部を形成し得る。いくつかの実施例において、命令は、実行されると、第１のバッファで、受信された第１のデータストリーム及び受信された第２のデータストリームを記憶することを、プロセッサに更に実行させる。いくつかの実施例において、命令は、実行されると、デコードされたメッセージの部分を、第２のバッファで記憶することを、プロセッサに更に実行させる。いくつかの実施例において、命令は、実行されると、プロセッサを使用し、第１のバッファ内にバッファリングされた、受信されたデータストリームをデコードすることを、プロセッサに更に実行させる。プロセッサは、受信された第２のデータストリームを完全に受信する前に、第１のバッファ内にバッファリングされた、受信された第１のデータストリームのデコードを開始する。

【０１３２】

本発明の一態様の実施形態からの要素は、本発明の他の態様で使用され得る（例えば、１つの独立した請求項に従属する請求項の要素は、他の独立した請求項の実施形態を更に指定するために使用され得る）。本発明の他の特徴及び利点は、次の図、発明を実施するための形態、及び請求項から明らかになるであろう。

【０１３３】

本開示の前記及び他の目的、態様、特徴、及び利点は、添付図面と併せて以下の説明を参照することによってより明白となり、より理解されるであろう。

【図面の簡単な説明】

【０１３４】

【図１】本発明の一実施形態による、プラットフォームサーバと複数のコンピューティングデバイスとの間で通信を有効にするための、例示的システムのブロック図である。

【図２】本発明の一実施形態による、ＡＰＩプロトコルに従ってフォーマット化されたメッセージを送信及び受信するための、例示的通信チャネルのブロック図である。

【図３Ａ】本発明の一実施形態による、ＡＰＩプロトコルライブラリのブロック図である。

【図３Ｂ】本発明の一実施形態による、バイナリ動的ＲＥＳＴ　ＡＰＩコンポーネントのメッセージ出力の例示的構造を示す。

【図３Ｃ】本発明の一実施形態による、ＡＰＩプロトコルライブラリで使用される、例示的ＨＴＴＰ　ＲＥＳＴ　ＡＰＩコードを示す。

【図３Ｄ】本発明の一実施形態による、ＡＰＩプロトコルライブラリで使用される、メッセージヘッダの例示的構造を示す。

【図３Ｅ】本発明の一実施形態による、ＡＰＩプロトコルライブラリの例示的コードブックである。

【図３Ｆ】本発明の一実施形態による、ＡＰＩプロトコルライブラリで使用される、マルチパートヘッダの例示的構造を示す。

【図３Ｇ】本発明の一実施形態による、ＡＰＩプロトコルライブラリで使用される、マルチパートヘッダの例示的構造を示す。

【図３Ｈ】本発明の一実施形態による、ＡＰＩプロトコルライブラリで使用される、認証本体の例示的構造を示す。

【図３Ｉ】本発明の一実施形態による、ＡＰＩプロトコルライブラリで使用される、バインド及びアンバインドメッセージ本体の例示的構造を示す。

【図４】本発明の一実施形態による、ＡＰＩプロトコルライブラリを使用した例示的メッセージ送信及び受信のブロック図である。

【図５】本発明の他の実施形態による、ＡＰＩプロトコルライブラリを使用したメッセージ送信及び受信の例のブロック図である。

【図６Ａ】本発明の一実施形態による、バイナリ動的ＲＥＳＴ　ＡＰＩによって生成された、例示的エンコードされたメッセージを示す。
【図６Ｂ】ＪａｖａＳｃｒｉｐｔ　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎフォーマットの例示的ＲＥＳＴメッセージを示す。
【図７Ａ】本発明の一実施形態による、シリアル化されたバイナリ動的ＲＥＳＴ　ＡＰＩメッセージを生成する例示的方法を示す。
【図７Ｂ】本発明の一実施形態による、シリアル化されたバイナリ動的ＲＥＳＴ　ＡＰＩメッセージを生成する例示的方法を示す。
【図７Ｃ】本発明の一実施形態による、シリアル化されたバイナリ動的ＲＥＳＴ　ＡＰＩメッセージを生成する例示的方法を示す。
【図８】本発明の一実施形態による、エンコードされたメッセージをパーティション化する例示的方法を示す。
【図９】本発明の一実施形態による、メッセージを再アセンブリする例示的方法のフローチャートである。
【図１０】本発明の一実施形態による、ＡＰＩプロトコルライブラリを使用する、ソフトウェア開発キット（ＳＤＫ）を実行するクライアントアプリケーション搭載の、例示的コンピューティングデバイスのブロック図である。
【図１１】本発明の一実施形態による、ＡＰＩプロトコルライブラリを使用してサードパーティのプロトコルをラッピングする例示的方法のブロック図である。
【図１２】本発明の一実施形態による、自己記述型メッセージを解析するための例示的方法のフローチャートである。
【図１３】本発明の一実施形態による、ＷｅｂＳｏｃｋｅｔ接続を使用して動的ＲＥＳＴメッセージを送信する例示的方法のフローチャートである。
【図１４】本発明の一実施形態による、サードパーティのプロトコルを使用して通信する例示的方法のフローチャートである。
【図１５】本発明の一実施形態による、チャンク化されたメッセージを使用して通信する例示的方法のフローチャートである。
【図１６】コンピューティングデバイス及びモバイルコンピューティングデバイスのブロック図である。
【０１３５】
本開示の特徴及び利点は、添付図面と併せて以下の詳細な説明を参照することによってより明らかとなるであろう。これらを通じて、同様の符号は対応する要素を示す。図面では、同様の参照番号は、同一の、機能的に類似する、及び／又は構造的に類似する要素を概ね示す。
【発明を実施するための形態】
【０１３６】
特許請求された発明のシステム、デバイス、方法、及び工程が、本明細書に記載される実施形態の情報を使用して開発された変形形態及び適応形態を包含することを理解されたい。本明細書に記載のシステム、デバイス、方法、及びプロセスの適応及び／又は変更は、当業者によって実行され得る。
【０１３７】
物品、デバイス、及びシステムが特定のコンポーネントを有する、含む、若しくは備えるものとして説明されている、又はプロセス及び方法が、特定の工程を有する、含む、若しくは備えるものとして説明されている説明の全体を通して、付加的に、列挙されたコンポーネントから本質的になる、又はそれらからなる、本発明の物品、デバイス、及びシステムが存在すること、また、列挙された処理工程から本質的になる、又はそれらからなる、本発明に従ったプロセス及び方法が存在することを理解されたい。
【０１３８】
本発明が動作可能のままである限り、工程の順序又は処置の実行順序は、重要ではないことを理解されたい。また、２つ以上の工程又は処置は、同時に実施し得る。

【０１３９】
任意の公開又は特許出願に関する本明細書の、例えば「背景技術」の項にある記述は、かかる公開又は特許出願が、本明細書に示される主張又は内容のいずれかに関する、従来技術を構成することを認めるものではない。「背景技術」の項は、明確にする目的で提供されるものであり、何らかの主張に関する従来技術の説明であることを意図するものではない。
【０１４０】
本明細書には、効率的、柔軟、かつコスト効果的な方法で、膨大な数のコンピューティングデバイスがコンピューティングプラットフォーム（例えば、マシンツーマシン通信及びＩｏＴ接続デバイスの通信用）に接続することを可能にする、通信プロトコルを採用した方法及びシステムが記載されている。通信プロトコルは、ネットワーク及び／又はインターネットベースのコンピューティングプラットフォーム内にあるデバイス間でメッセージ（例えば、ＨＴＴＰ　ＲＥＳＴｆｕｌメッセージ）を送信するため、持続的接続を使用して（例えば、ＷｅｂＳｏｃｋｅｔ接続経由で）低遅延の双方向接続を更に可能にする。
【０１４１】
通信プロトコルは、いくつかの例示的実施例において、一組の共通メッセージ及び要求に対し、ハイパーテキスト転送プロトコル（ＨＴＴＰ）に基づいたバイナリ要求／応答構成体を更に採用する。このようなバイナリ要求／応答構成体は、接続デバイスが、バイナリ動的ＲＥＳＴ　ＡＰＩに加え、標準ＲＥＳＴ　ＡＰＩを介してレガシーシステムとインターフェースを取ることを可能にする。
【０１４２】
実際に、通信プロトコルは、いくつかの例示的実施例において、その柔軟性を保持しながら、ＨＴＴＰ　ＲＥＳＴｆｕｌ　ＡＰＩの冗長性を有益に軽減するため、シリアル化及びバイナリ化された（例えば、バイナリシンボルで表現される）動的ＲＥＳＴ　ＡＰＩ（例えば、データ及び構成メッセージ用）を更に採用する。また、動的ＲＥＳＴ　ＡＰＩプロトコルは、自己記述型フォーマットで送信されたメッセージを記述するため、モデルベーススキーマを採用し、それによって、メッセージ構造は、メタ構成体におけるメッセージ記述に基づいて変更する（例えば、順序、長さ、型など）ことが可能になる。また、動的ＲＥＳＴ　ＡＰＩプロトコルは、メタ構成体及びメッセージ構成体の下層組織及びプロパティを記述する、データ及び情報モデルを採用する。情報モデルは、通常の動作中にそのビット数を消費することなく、メッセージの下層構造のアップグレードを有益に可能にする。
【０１４３】
通信プロトコルは、それぞれの接続に対して、小さい固定メモリ使用量を使用した大きい動的ＲＥＳＴメッセージの送信を更に促進する。
【０１４４】
図１は、本発明の一実施形態による、プラットフォームサーバ１０２（「プラットフォームサーバ１０２ａ」及び「プラットフォームサーバ１０２ｂ」のいずれかとして表示）と複数のコンピューティングデバイス１０４との間で通信を有効にするための、例示的システム１００のブロック図である。コンピューティングデバイス１０４のそれぞれは、コンピューティングデバイス１０８の群（１０８ａ、１０８ｂ、１０８ｃ、及び１０８ｄとして表示）との通信を提供及び維持する、エッジサーバ１０６に接続する。
【０１４５】
コンピューティングデバイス１０４は、いくつかの例では、物理的資産／デバイス、コンピュータアプリケーション、システム、人、データオブジェクト、及びプラットフォームサービスに関する、プロパティデータ及び情報、サービスデータ及び情報、イベントデータ及び情報などを伝達できる電子デバイスである。いくつかの実施例において、コンピューティングデバイス１０４は、コンビナートのセンサ若しくは機械装置、事務所若しくは役所のコンピュータ若しくはオフィス装置、市場の店頭機械若しくは自動販売機、建設機械若しくは自動車、発電若しくは配電装置、変電若しくは送電装置、建物のメーター、

サーバ、ネットワーキング若しくはルーティング装置、スマート家電、エクササイズマシン、医療用装置若しくはプロテーゼ装置、医療用診断装置若しくは病院用器具、商用車若しくは輸送コンテナ、自動車若しくは電動自転車、携帯電話、ラップトップ、タブレット、電子リーダー、又は衣類の電子タグである。

【０１４６】

複数の組のコンピューティングデバイス１０４にデータ及び情報を提供する一方、１つ又は２つ以上のエッジサーバ１０６は、ＡＰＩサーバ１１０（又は「接続サーバ１１０」、１１０ａ、１１０ｂ、１１０ｃ、１１０ｄ、及び１１０ｅとして表示）とも呼ばれる中間サーバと通信する。いくつかの実施例において、通信交換は、インターネット１１２ａ、ワイドエリアネットワーク１１２ｂ、又はサードパーティネットワーク１１２ｃなどのネットワークインフラストラクチャ１１２間で発生する。次いで、１つ又は２つ以上のＡＰＩサーバ１１０が、プラットフォームサーバ１０２と通信する。プラットフォームサーバ１０２、ＡＰＩサーバ１１０、及びエッジサーバ１０６は、コンピューティングデバイス１０４のデータ及び情報を提供するための分散コンピューティング環境を集合的に形成する。いくつかの実施例において、所定のＡＰＩサーバ１１０は、一組のネットワークセキュリティ機器１１４（例えば、１１４ａ～ｈ）を介して、一組のエッジサーバ１０６と通信する。ネットワークセキュリティ機器１１４は、ネットワークインフラストラクチャ１１２からＡＰＩサーバ１１０及びプラットフォームサーバ１０２を保護し、また、エッジサーバ１０６及びコンピューティングデバイス１０４の群を保護する。ネットワークセキュリティ機器１１４は、例えば、ファイアウォール又はネットワークアドレス変換（ＮＡＴ）プロトコルを実行し得る。

【０１４７】

図２は、本発明の一実施形態による、ＡＰＩプロトコルに従ってフォーマット化されたメッセージを送信及び受信するための、例示的通信チャネル２００のブロック図である。図１に関して説明されているとおり、持続的通信チャネル２００は、第１の持続的接続１０３及び第２の持続的接続１０５などの持続的接続を含む。

【０１４８】

図２に示されているとおり、プラットフォームサーバ１０２は、ＡＰＩプロトコルを定義するＡＰＩプロトコルライブラリ２０４（２０４ａとして表示）を有するサーバークライアントアプリケーションを実行する。エッジサーバ１０６は、同一のＡＰＩプロトコルライブラリ２０４（２０４ｃとして表示）を有するサーバークライアントアプリケーションを実行する。この目的のため、プラットフォームサーバ１０２とエッジサーバ１０６との間で伝達されるメッセージは、大部分において同一の構造を共有する。いくつかの実施例において、同一のＡＰＩプロトコルが、分散コンピューティング環境内のすべてのコンポーネント間で採用される。

【０１４９】

いくつかの実施例において、接続サーバ１１０は、ソース又はターゲットにほとんど関係なく、同一の方法で、伝達されたメッセージのそれぞれを提供（例えば、処理）する可能性があるため、メッセージのこの対称性は、接続サーバ１１０の操作の複雑さを軽減することを意図している。

【０１５０】

ＡＰＩプロトコルライブラリ２０４は、データ及び情報を提供するためのアプリケーションインターフェースを動的に生成する、多数のサーバークライアントアプリケーション機能を提供する。いくつかの実施例において、アプリケーションインターフェースは、バイナリメッセージを生成する、動的ＲＥＳＴＡＰＩに基づく。他の実施例において、インターフェースは、ＪａｖａＳｃｒｉｐｔ Ｏｂｊｅｃｔ Ｎｏｔａｔｉｏｎ（「ＪＳＯＮ」）又はＥｘｔｅｎｓｉｖｅ Ｍａｒｋｕｐ Ｌａｎｇｕａｇｅ（「ＸＭＬ」）データフォーマットに基づく。モデルベースの開発アプリケーションでの動的ＲＥＳＴ ＡＰＩの使用例は、２０１２年１１月１６日に出願された米国特許出願番号第１３／６７８，８８５号、題名「Ｍｅｔｈｏｄｓ ｆｏｒ Ｄｙｎａｍｉｃａｌｌｙ Ｇｅｎｅｒａｔｉｎｇ

Ａｐｐｌｉｃａｔｉｏｎ Ｉｎｔｅｒｆａｃｅｓ ｆｏｒ Ｍｏｄｅｌｅｄ Ｅｎｔｉｔｉｅｓ ａｎｄ Ｄｅｖｉｃｅｓ Ｔｈｅｒｅｏｆ」に記載されている。この出願の内容は、その全体が本明細書に参考として組み込まれる。

【０１５１】

図３Ａは、本発明の一実施形態による、ＡＰＩプロトコルライブラリ２０４のブロック図である。ＡＰＩプロトコルライブラリ２０４は、バイナリ動的ＲＥＳＴ ＡＰＩコンポーネント３０２及び固定ＨＴＴＰ ＲＥＳＴ ＡＰＩコンポーネント３０４を含む。

【０１５２】

動的ＲＥＳＴ ＡＰＩ ３０２は、データ及び情報を記述するため、モデルベーススキーマを採用しており、スキーマの入力及び出力は、ＪＳＯＮ又はＸＭＬなどの自己記述型テキストデータフォーマットで表現される。いくつかの例示的実施形態において、自己記述型は、データ片（例えば、入力又は出力、ファイル、及びそれに類するもの）内に記述的情報（例えば、自己記述型データ（ＳＤＤ））を含むという概念を指す。アプリケーションインターフェースは、データ及び情報が、進化する及び変更可能なフレームワークで表現される点において「動的」である。特に、いくつかの例示的実施例において、アプリケーションインターフェースは、物理的資産／デバイス、コンピュータアプリケーション／システム、人、データオブジェクト、又はプラットフォームサービスなどの、所定のエンティティ（「モノ」とも呼ばれる）に関する情報の種類を記述するための一連の特性定義を、自己記述型データとして採用する。

【０１５３】

動的ＲＥＳＴ ＡＰＩライブラリの情報モデルフレームワーク

いくつかの実施例において、バイナリ動的ＲＥＳＴ ＡＰＩコンポーネント３０２は、データモデル及び情報モデルを表現する、シリアル化されたバイナリデータを出力する。

【０１５４】

いくつかの実施例において、特性定義は、「プロパティ」定義３０３ａ、「サービス」定義３０３ｂ、「イベント」定義３０３ｃ、及び「属性」定義３０３ｄを含む。これらの特性定義のそれぞれは、エンティティのモデルの記述における基礎的要素の役割を果たし、１つ又は２つ以上のクラス、抽象クラス、及び／又はデータ構造を継承できる抽象クラスとして表現される。特性定義は、他の特性定義と共に、テンプレートクラスで集合的に編成される。この目的のため、コンピューティングデバイスのデータ及び情報を提供するアプリケーションの開発者は、組織的なランドスケープにおけるモノとその場所を定義する、データ、サービス、イベント、履歴活動、コラボレーション、関係、及びその類のものの完全なセットを含む、物質界の等価物を表現するエンティティ（「モノ」とも呼ばれる）をモデル化し得る。

【０１５５】

いくつかの実施例において、「モノ」は、「モノテンプレート（Thing Template）」のインスタンスとして定義される。「モノテンプレート」は、１つ又は２つ以上の「モノシェイプ（Thing Shapes）」から継承可能な、抽象クラスであり得る。プロパティ、サービス、及びイベントは、「モノシェイプ」、「モノテンプレート」、又は「モノインスタンス」によって定義され得る。「モノテンプレート」が、１つ又は２つ以上の「モノシェイプ」から継承する場合、「モノシェイプ」のプロパティ、イベント、及びサービスのすべては、「モノテンプレート」の一部になる。「モノインスタンス」が「モノテンプレート」から作製されるとき、「モノテンプレート」のプロパティ、イベント、及びサービスのすべては、「モノインスタンス」内で現実化（例えば、継承）される。

【０１５６】

いくつかの実施例において、データオブジェクトは、「インフォテーブル」及び「ストリーム」として表現される。「インフォテーブル」及び「ストリーム」は、再利用可能な、抽象データオブジェクト定義である、「データシェイプ３０５」によって記述及び定義され得る。「インフォテーブル」は、リレーショナルデータベーステーブル又はその類のものと似ており、データオブジェクトを二次元的に（例えば、列及び行で）表現する。「

ストリーム」は、連続的な時系列データ又はその類のものなどの、データポイントをキャプチャするように設計される。「ストリーム」及び「インフォテーブル」はまた、サービス及びイベントを有し得る。

【０１５７】

エンティティは、認可される場合、他の（例えば、他のエンティティに対応する）エンティティイベントにサブスクライブでき、他のエンティティサービスを使用できる。エンティティは、定義されると、ＨＴＴＰ、ＨＴＴＰＳ、又はその類のものを経由し、標準のＲｅｐｒｅｓｅｎｔａｔｉｏｎａｌ　Ｓｔａｔｅ　Ｔｒａｎｓｆｅｒ（ＲＥＳＴ）インターフェースなどのインターフェースを介して検出可能になる。したがって、モデルの名前空間は、動的ＲＥＳＴインターフェースを介して使用可能である。ユーザーがモデルをどのように定義しても、モデルは、ＲＥＳＴインターフェースとして出現し得る。モデルに対するＲＥＳＴインターフェースは、それぞれのエンティティのプロパティ、サービス、及びイベントの記述を含み得る。サービスは、ＲＥＳＴインターフェースにアクセス可能な、サーバに提供される単純又は複雑な機能である。サービスは、入力、プロセス、及び出力を有する。イベントは、データの変更及び／又はエンティティのステータスであり得る。イベントは、イベントが検出されたとき、それぞれのイベントのサブスクライバーに送信されるデータ出力を有する。

【０１５８】

動的ＲＥＳＴインターフェースは、継承又はオブジェクト指向モデルに基づく。新規のサービス、プロパティ、又は機能が「モノシェイプ」又は「モノテンプレート」レベルで定義される場合、それらの「モノシェイプ」又は「モノテンプレート」のエンティティに由来するそれぞれの「モノ」インスタンスは、そのサービス、プロパティ、又は機能を直接に継承する。動的ＲＥＳＴ　ＡＰＩ　３０２は、この動的ＲＥＳＴインターフェースを採用する。

【０１５９】

名前空間に対するＲＥＳＴインターフェースは、モデル内のそれぞれのエンティティに対するサービスを処理する方法を記述する。いくつかの例示的実施例において、新規の「モノ」がモデルで定義されるとき、その「モノ」のサービス及びデータは、一組のＲＥＳＴインターフェースとして使用可能になる。

【０１６０】

通信プロトコルの概要

図３Ｂは、本発明の一実施形態による、バイナリ動的ＲＥＳＴ　ＡＰＩコンポーネント３０２のメッセージ出力の例示的構造を示す。

【０１６１】

バイナリ動的ＲＥＳＴ　ＡＰＩから生成されるメッセージは、メタデータ構成体３０６及びメッセージ構成体３０８を含み得る。メタデータ構成体３０６は、メッセージ構成体３０８の所定の構造を記述する。集合的に、いくつかの実施例において、メタデータ構成体３０６及びメッセージ構成３０８は、メッセージ本体３０５を形成する。

【０１６２】

メッセージ構成体３０８は、サーバークライアントアプリケーションが送信することを目的とする、実際のデータセットを含み得る。いくつかの実施例において、これらのデータセットは、「行データ３０８」と呼ばれる。データセット（例えば、行データ３０８）は、１つ又は２つ以上のデータ値３１０を含み得る。メタデータ構成体３０６は、データセットを形成する所定のデータ値に対応する及び／又は含む、１つ又は２つ以上の記述データセット３１２を含み得る。記述セット３１２は、１つ又は２つ以上の記述値３１４を含み得る。いくつかの実施例において、メタデータ構成体３０６は、「データシェイプ３０６」と呼ばれ、記述セット３１２のそれぞれは、データシェイプ３０６の所定のフィールド定義を形成する。フィールド定義の例としては、名前、説明、型、及びアスペクトが挙げられ得る。これらの定義は、行データ３１０における所定のデータ３１６を記述する。

【０１６３】
いくつかの実施例において、バイナリ動的ＲＥＳＴ　ＡＰＩから生成されるメッセージは、データセット３１０のそれぞれ、及び記述セット３１２のそれぞれを詳述するための、バイナリマーカー３１８の第１の型を含む。これらのバイナリマーカー３１８は、データセット及び記述セットの始まりにマークを付け得る。加えて、メタデータ構成体３０６及びメッセージ構成体３０８のそれぞれは、メタデータ構成体３０６及びメッセージ構成体３０８の終わりをマーク付けするための、第２のバイナリマーカー３２０を含み得る。
【０１６４】
いくつかの実施例において、データセット３１０及び記述セット３１２のそれぞれは、文字列、数値、及び列挙値（例えば、３２２、３２４、３２６）として表現されるデータを含む。文字列は、テキストデータであり得る。数値は、浮動小数点数及び固定小数点数を含み得る。列挙値は概ね、データ型を含む。それぞれの文字列３２２は、コンテンツの前方記述子を提供するため、実際の文字列値に先行する長さ記述フィールドを含み得る。この目的のため、エンコードされた形状の所定の文字列３２２は、完全な文字列３２２が使用可能ではない状態で解析され得る。不明データの群は、後に続くこのような文字列データの数を識別するための、カウントフィールド３２８を含み得る。
【０１６５】
プロトコルのメモリ要件を軽減するため、いくつかの実施例において、プロトコルは、コードブックライブラリに記憶されているシンボルとして定義されたバイナリ表現を使用して、メッセージの第１の部分セットをエンコードする。いくつかの実施例において、バイナリ表現はバイト列挙値を含む。
【０１６６】
プロトコルは、所定のコンピューティングデバイスにネイティブである標準のデータフォーマットを使用して、メッセージの第２の部分セットをエンコードし得る。このような標準データフォーマットの例としては、ＵＴＦ－８などのユニコード文字が挙げられる。
【０１６７】
メッセージは、ＷｅｂＳｏｃｋｅｔプロトコルを使用した送信用に最適化された、バイナリ文字列としてフォーマット化され得る。
【０１６８】
通信プロトコル－固定ＨＴＴＰ　ＲＥＳＴ　ＡＰＩコンポーネント
図３Ａに戻ると、ＨＴＴＰ　ＲＥＳＴ　ＡＰＩコンポーネント３０４は、ハイパーテキスト転送プロトコルアプリケーションプロトコルバージョン１．１（一般に、「ＨＴＴＰ／１．１」又は「ＨＴＴＰ」と呼ばれる）内に定義され、ＩＥＴＥ／Ｗ３Ｃ　ＲＦＣ　２６１６に公開されているとおり、ＨＴＴＰフレームワークに由来するメッセージング型定義コード３３０及びステータスコード３３２を含み得る。ＨＴＴＰ　ＲＥＳＴ　ＡＰＩコードは、図３Ｃに提供されている。ステータスコード３３２の例としては、成功コード３３２ａ、クライアントエラーコード３３２ｂ、及びサーバエラーコード３３２ｃが挙げられる。メッセージング型定義コード３３０の例としては、デバイスバインディング、認証、及び要求メッセージングに関するものが挙げられる。
【０１６９】
いくつかの実施例において、メッセージング型定義コード３３０及びステータスコード３３２は、メッセージに関するルーティング情報を有する、メッセージヘッダ３３８の一部である。メッセージヘッダの構造の例は、図３Ｄに示されている。いくつかの実施例において、メッセージヘッダ３３８は、ヘッダ識別番号３４０、ルーティング情報３４２（例えば、要求識別番号３４２ａ、セッション番号３４２ｂ、及びエンドポイント識別番号３４２ｃ）、及びメソッドコード３４４を含む。メソッドコード３４４は、メッセージコンテンツ内に位置するメッセージの型を指定するために使用され得る。メッセージヘッダ３３８は、メッセージが、結合されたときに単一メッセージを形成する、メッセージの集合の一部であることを示すための、マルチパートマーカー３４６を含み得る。
【０１７０】

図３Ａを更に参照すると、いくつかの実施例において、ＡＰＩプロトコルライブラリ２０４は、所定のメッセージを送信用の固定サイズブロックにセグメント化するための、メッセージパーティショニングモジュール３３４を含む。セグメント化は、接続のそれぞれに対して小さい固定メモリ使用量を使用した大きいメッセージの送信を促進し得る。メッセージパーティショニングモジュール３３４は、所定のメッセージをチャンク化又はパーティション化して、トランスポートレベルではなくアプリケーションレベルで、そのチャンク又はパーティションを再アセンブルし得る。

【０１７１】

ＡＰＩプロトコルライブラリ２０４は、図７Ａ及び７Ｂに関して説明されているとおり、文字列３２２の長さ記述フィールド又はカウントフィールド３２８などのメタデータを採用して、完全なメッセージを有することなく、所定のメッセージコンテンツの解析を可能にし得る。つまり、メタデータは、所定のメッセージ内のメタデータに続くコンテンツを記述する。メッセージパーティショニングモジュール３３４を使用して、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２によって生成される所定のメッセージは、チャンクにパーティション化され得、チャンクのそれぞれは、完全なメッセージが受信される前にデコードされ得る。

【０１７２】

この目的のため、任意の長さのメッセージが、デコード用の２つのメッセージ長のみからなるバッファを使用して、デコードされ得る。いくつかの実施例において、サーバークライアント側アプリケーションは、好ましくは１Ｋバイトバッファと１Ｍバイトバッファとの間、更により好ましくは４Ｋバイトと６４Ｋバイトとの間、更により好ましくは４Ｋバイトと８Ｋバイトとの間の固定バッファを採用する。

【０１７３】

いくつかの実施例において、マルチパートヘッダ３４８は、完全なメッセージが複数のチャンク又はパーティション間で分散されることを示すため、メッセージ内に採用される。メッセージマルチパートヘッダ３４８の構造の例は、図３Ｆ及び３Ｇに示されている。図３Ｆ及び３Ｇに示されているとおり、メッセージマルチパートヘッダ３４８は、現在のチャンク識別番号３５０ａ、合計チャンク番号３５０ｂ、及びチャンクサイズ識別子３５０ｃを含む。メッセージマルチパートヘッダ３４８は、ヘッダ３３８（「マルチパート３４６」としても表示）とメッセージ本体３３９（「本体３３９」として表示）との間に挿入され得る。

【０１７４】

いくつかの実施例において、ＡＰＩプロトコルライブラリ２０４は、メッセージフローの方向によって、異なるメッセージ構造を採用し得る。例えば、クライアント側アプリケーションから、サーバ側アプリケーションへのインバウンドメッセージについて、宛先情報を提供するために、ルーティング情報３４２ｄがマルチパートメッセージヘッダ３４８に組み込まれ得る（図３Ｇを参照のこと）。

【０１７５】

図３Ａに戻ると、いくつかの実施例において、ＡＰＩプロトコルライブラリ２０４は、ＷｅｂＳｏｃｋｅｔを介してＡＰＩプロトコルライブラリ２０４の出力メッセージを伝達するため、ＷｅｂＳｏｃｋｅｔ　ＡＰＩ　３３６と共に使用される。ＷｅｂＳｏｃｋｅｔ　ＡＰＩ　３３６は、メッセージをストリーミングするために、伝送制御プロトコル（ＴＣＰ）の最上位に採用され得る。いくつかの実施例において、ＷｅｂＳｏｃｋｅｔ　ＡＰＩ　３３６は、ＨＴＭＬ５　ＷｅｂＳｏｃｋｅｔプロトコルの一部である。ＡＰＩプロトコルライブラリ２０４のメッセージ出力は、ＷｅｂＳｏｃｋｅｔ　ＡＰＩ　３３６を介した通信を更に向上させるため、全体的にバイナリになるように構成され得る。

【０１７６】

動的ＲＥＳＴ　ＡＰＩプロトコルライブラリを使用したバイナリメッセージのエンコードの例

図４は、本発明の一実施形態による、ＡＰＩプロトコルライブラリ２０４を使用した、

例示的メッセージ送信及び受信のブロック図である。クライアント－サーバアプリケーション４００ａは、別のクライアント－サーバアプリケーション４００ｂにメッセージを送信するための関数を呼び出すことができる。関数呼び出しは、ＡＰＩプロトコルライブラリ２０４の一部であってもよい。

【０１７７】

いくつかの実施例において、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２は、自己記述型メッセージに対応するシリアル化バイナリストリームを出力する。図６Ａは、本発明の一実施形態による、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２によって生成された、例示的エンコードされたメッセージ６０２を示す。図６Ｂは、ＪａｖａＳｃｒｉｐｔ　Ｏｂｊｅｃｔ　Ｎｏｔａｔｉｏｎ（ＪＳＯＮ）フォーマットで表現された、図６Ａの例示的エンコードされたメッセージ６０２を示す。

【０１７８】

この目的のため、本明細書に記載の実施形態は、同一のメッセージコンテンツを有するＪＳＯＮフォーマットメッセージに比べて、５０％を超えるメッセージ長の削減を示す。コロン「：」によって区切られる属性－値ペアを有する、典型的なＪＳＯＮメッセージが編成され、ペア値は、開き大括弧「｛」及び閉じ大括弧「｝」によって区切られる、別の属性－値ペアであってもよい。属性－値ペアフォーマットは、メッセージを記述するための柔軟なスキーマを提供する。

【０１７９】

所定のメッセージの情報効率を向上させるため、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２は、データオブジェクトのみでメッセージ６０２を生成し得る。バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２は、メタデータ構成体３０６の一部、すなわちデータシェイプ３１２として（構造記述として）、記述値３１４の構造記述を記憶し得る。記述値３１４は、次いで、データ値３１６の定義、すなわち、メッセージ構成体３１０内の行データ３１６を提供し得る。いくつかの実施例において、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２は、（ｉ）データセット３１０及び記述セット３１２のそれぞれの始まりをマーク付けするための、第１のバイナリマーカー３１８と、（ｉｉ）データセット３１０及び記述セット３１２のそれぞれの終わりをマーク付けするための、第２のバイナリマーカー３２０と、を採用し得る。いくつかの実施例において、ＪＳＯＮベースメッセージに関連付けられたものなど、各種の句読点及びラベル要素は、組み込む必要がない。

【０１８０】

図７Ａ～７Ｃは、本発明の一実施形態による、シリアル化されたバイナリ動的ＲＥＳＴ　ＡＰＩメッセージを生成する例示的方法を示す。

【０１８１】

特に、図７Ａ及び７Ｂは、本発明の一実施形態による、メッセージのメタデータ部分を生成する例を示す。図７Ｃは、本発明の一実施形態による、メッセージのデータ部分を生成する例を示す。

【０１８２】

ここで図７Ａに戻ると、メタデータ部分７０２のメッセージ構造は、図６Ｂに示されたＪＳＯＮメッセージのマークアップテキスト表現７０４と一緒に表示されている。ＪＳＯＮメッセージは、１つ又は２つ以上のメタデータ構成体定義セット３１２と、１つ又は２つ以上のメッセージ構成体データ値３１０と、を有する、自己記述型メッセージを示す。

【０１８３】

メタデータ部分７０２は、図３Ｂに関して説明されているとおり、「データシェイプ」３１２に対応する１つ又は２つ以上のフィールド定義を含み得る。図７Ａに示されているとおり、フィールド定義は、マーカーフィールド７１２、名前フィールド７１４、説明フィールド７１６、型フィールド７１８、アスペクトカウントフィールド７２０、及びアスペクトフィールド７２２を含む。

【０１８４】

所定のメッセージ構成体３０８は、対応するメタデータ構成体３０６の記述値３１２に

よって記述される、データ値３１０の群を含み得る。図７Ａ～７Ｃに関して説明されているとおり、メッセージ構成体３０８は、「行データ」（例では「ｒｏｗｓ」７２４として表示）と呼ばれ、メタデータ構成体３１２は、「データシェイプ」（例では「ｄａｔａｓｈａｐｅ」７２６として表示）と呼ばれる。

【０１８５】

図７Ａ～７Ｃに示されているとおり「ｄａｔａｓｈａｐｅ」７２６は、３つのフィールド定義（７２８、７３０、及び７３２）を含む。３つのフィールド定義のうちの２つの暗号化（「ｐｕｓｈＴｈｒｅｓｈｏｌｄ」７２８及び「ｐｕｓｈ　Ｔｙｐｅ」７３２）は、図７Ａに関して論じられており、３つのフィールド定義の３つ目（「ｅｄｇｅ　Ｎａｍｅ」７３０）は、図７Ｂに関して論じられている。

【０１８６】

図７Ａ～７Ｃに示されているとおり、フィールド定義７２８、７３０、及び７３２のそれぞれは、互いに共通する記述値の群、すなわち、「ｎａｍｅ」、「ｄｅｓｃｒｉｐｔｉｏｎ」、「ｂａｓｅＴｙｐｅ」、及び「ａｓｐｅｃｔｓ」を含む。これらの記述値７３４は、特性定義の種類の例示的実施例であり、図３Ａに関して説明されているとおり、動的ＲＥＳＴ　ＡＰＩ　３０２によって使用されるモデルベーススキーマのテンプレートクラスを集合的に形成し得る。

【０１８７】

図７Ａ～７Ｃに示されているとおり、定義７３４の群のそれぞれは、ＪＳＯＮフォーマットに従ってフォーマット化される、「属性」－「値」ペアを有する。メッセージのエンコードにおいては、ペアの「値」部分のみが使用される（メッセージ内）。ペアの属性部分は、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２に組み込まれる、モデルベーススキーマの構造定義の一部として記憶される。

【０１８８】

例示的ＪＳＯＮメッセージ６０２のバイナリメッセージを生成するには、いくつかの実施例において、マーカー値が最初にバイナリメッセージに追加される。図３Ｂに関して説明されているとおり、マーカー３１８（図７Ａに７１２として表示）は、メッセージ構成体３０８のデータセット３１０（「行データ」７２４など）のそれぞれと、メタデータ構成体３０６の記述セット３１２（「データシェイプ」３１２として表示される、「フィールド定義」など）のそれぞれと、を詳述する。マーカー７１２は、データセット７２４及び記述セット３１２の始まりにマークを付ける。加えて、マーカー３２０の第２の型（図３Ｂに表示）は、メタデータ構成体３０６及びメッセージ構成体３０８の終わりにマークを付ける。いくつかの実施例において、マーカー値は、１桁のバイナリ値（例えば、「１」又は「０」）であってもよい。

【０１８９】

図７Ａに示されているとおり、いくつかの実施例において、「ｎａｍｅ」値７３６、「ｄｅｓｃｒｉｐｔｉｏｎ」値７３８、「ｂａｓｅＴｙｐｅ」値７４０、及び「ａｓｐｅｃｔ」値７４２が、マーカー７１２に続く。

【０１９０】

「ｎａｍｅ」値７３６は、文字列として表現されてもよい。バイナリメッセージでは、所定の文字列が、文字列の値７４６に先行する長さ識別子７４４によって定義され得、ＵＴＦ－８フォーマット（ユニコード標準）で表現される。図７Ａに示されているとおり、文字列「ｐｕｓｈＴｈｒｅｓｈｏｌｄ」７３６は、長さ「１３」文字（１６進コードで「０ｘ０ｄ」として表示可能）を有し、かつ、ＵＴＦ－８で表現されるとき、「０ｘ７０　０ｘ７５　０ｘ７３　０ｘ６８　０ｘ５４　０ｘ６８　０ｘ７２　０ｘ６５　０ｘ７３　０ｘ６８　０ｘ６ｆ　０ｘ６ｃ　０ｘ６４」のデータ値を（１６進コードで）有する。この変換は、ボックス７４８に注釈されている。この目的のため、１６進値「０ｘ０ｄ　０ｘ７０　０ｘ７５　０ｘ７３　０ｘ６８　０ｘ５４　０ｘ６８　０ｘ７２　０ｘ６５　０ｘ７３　０ｘ６８　０ｘ６ｆ　０ｘ６ｃ　０ｘ６４」がバイナリメッセージに付加される。

【０１９１】
示されているとおり、「ｄｅｓｃｒｉｐｔｉｏｎ」値７３８は、「３９」文字（１６進コード「０ｘ５７」）を有する文字列である。ＵＴＦ－８では、文字列「Ｃｈａｎｇｅ ｔｈｒｅｓｈｏｌｄ ｔｏ ｇｅｎｅｒａｔｅ ｅｖｅｎｔ ｆｏｒ ｎｕｍｅｒｉｃ ｐｒｏｐｅｒｔｉｅｓ」は、１６進値で次のように表現される。
０ｘ４３ ０ｘ６８ ０ｘ６１ ０ｘ６ｅ ０ｘ６７ ０ｘ６５ ０ｘ２０ ０ｘ７４
０ｘ６８ ０ｘ７２ ０ｘ６５ ０ｘ７３ ０ｘ６８ ０ｘ６ｆ ０ｘ６ｃ ０ｘ６４
０ｘ２０ ０ｘ７４ ０ｘ６ｆ ０ｘ２０ ０ｘ６７ ０ｘ６５ ０ｘ６ｅ ０ｘ６５
０ｘ７２ ０ｘ６１ ０ｘ７４ ０ｘ６５ ０ｘ２０ ０ｘ６５ ０ｘ７６ ０ｘ６５
０ｘ６ｅ ０ｘ７４ ０ｘ２０ ０ｘ６６ ０ｘ６ｆ ０ｘ７２ ０ｘ２０ ０ｘ６ｅ
０ｘ７５ ０ｘ６ｄ ０ｘ６５ ０ｘ７２ ０ｘ６９ ０ｘ６３ ０ｘ２０ ０ｘ７０
０ｘ７２ ０ｘ６ｆ ０ｘ７０ ０ｘ６５ ０ｘ７２ ０ｘ７４ ０ｘ６９ ０ｘ６５
０ｘ７３。
【０１９２】
示されているとおり、「ｂａｓｅＴｙｐｅ」値７４０は、データフォーマットの型として表現される。いくつかの実施例において、データフォーマット情報は、ＡＰＩプロトコルライブラリ２０４の一部を構成するコードブックに定義されたバイナリシンボルとして、好ましくは表現される。図３Ｅは、本発明の一実施形態による、ＡＰＩプロトコルライブラリ２０４の例示のコードブックである。いくつかの実施例において、「ｎｕｍｂｅｒ」３４３のバイナリシンボルは、好ましくは８ビット長である。この変換は、ボックス７５２に注釈されている。
【０１９３】
示されているとおり、「ａｓｐｅｃｔｓ」値７４２は数値である。いくつかの実施例において、「ａｓｐｅｃｔｓ」値は2バイトのＳＨＯＲＴ（例えば、それぞれ長さ４ビットの２つのバイトを有する）である。ここで、値はＮＵＬＬ又はＥＭＰＴＹであり、１６進値で「０ｘ００ ０ｘ００」として表現される。この変換は、ボックス７５４（図７Ａ）に注釈されている。
【０１９４】
いくつかの実施例において、メタデータ構成体３０６を形成するための、定義３１２の群のバイナリ表現を付加するプロセスは、メッセージ６０２の全体に対して繰り返される。この目的のため、「ｅｄｇｅ Ｎａｍｅ」及び「ｐｕｓｈＴｙｐｅ」フィールド定義７３０、７３２の各種のデータ値が、バイナリメッセージに付加される。これらのデータ値の値は、注釈ボックス（７５６、７５８、７６０、７６２、７６４、７６６、７６８、及び７７０）に提供される。いくつかの実施例において、エンドマーカー３２０（例えば、コード「０ｘ００」）がデータ値に続き、メッセージ構成体３０６の終わりを詳述する。いくつかの実施例において、フィールド定義３１２のそれぞれは、名前識別子を使用して記述されるため、バイナリメッセージ内のそれぞれの対応のフィールド定義３１２の順序は、メッセージのエンコードにもデコードにも使用されない。
【０１９５】
図７Ｂを参照すると、いくつかの実施例において、アスペクト値の群は、データ値の群内にネストされる。アスペクトカウントフィールド７２０は、多数のこのようなネストされた群を定義し得る（ボックス７７０を参照のこと）。いくつかの実施例において、アスペクト群のそれぞれは、「アスペクト名」値７７２、「アスペクト型」値７７４、及び「アスペクトデータ」値７７６を含む。所定のアスペクト群のエンコードは、ＪＳＯＮメッセージ内の他の群のエンコードと類似であり得、例えば、文字列（例えば、７４６）は、長さ識別子（例えば、７４４）が先行し、データ型（例えば、７１８）はバイナリシンボルとして表現される。いくつかの実施例において、「アスペクト名」値７７２は文字列であり、「アスペクト型」７７４はデータ型であり、「アスペクトデータ」値７７６は、「アスペクト型」値によって記述されるフォーマットを有する値である。
【０１９６】

ここで図７Ｃを参照すると、例示的メッセージ６０２のメッセージ構成体３０８のエンコードが記述されている。図３Ｂに関して説明されているとおり、メタデータ構成体３０６は、メッセージ構成体３０８の所定の構造を記述する。この目的のため、いくつかの実施例において、メッセージ構成体３０８の行データ３１０のそれぞれは、メタデータ構成体３０６に含まれる記述値３１２によって記述される。

【０１９７】

図７Ｃに示されているとおり、メタデータ構成体３０６は、３つのフィールド定義、すなわち、「ｐｕｓｈＴｈｒｅｓｈｏｌｄ」フィールド定義７２８、「ｅｄｇｅ　Ｎａｍｅ」フィールド定義７３０、及び「ｐｕｓｈＴｙｐｅ」フィールド定義７３２を含む。この目的のため、「行データ」（７７８、７８０、７８２、７８４、及び７８６）は、これらのフィールド定義７２８、７３０、７３２に従って構造化される。

【０１９８】

いくつかの実施例において、「行データ」３１０（図７Ｃの７２４として表示）は、マーカー７１２、行カウントフィールド７８８、及び行カウントフィールド７８８の数値に対応する多数のデータ群７９０を含む。データ群７９０のそれぞれは、データ型フィールド７９２及び値フィールド７９４を含み得る。

【０１９９】

図７Ｃに示されているとおり、「行データ」７７８は、「ｐｕｓｈＴｈｒｅｓｈｏｌｄ」値「１０００００」、「ｅｄｇｅ　Ｎａｍｅ」値「ＩｎＭｅｍｏｒｙ＿Ｓｔｒｉｎｇ」、及び「ｐｕｓｈＴｙｐｅ」値「ＮＥＶＥＲ」を含む。

【０２００】

「ｐｕｓｈＴｈｒｅｓｈｏｌｄ」フィールド定義７２８によって定義されているとおり、値「１０００００」は「ＮＵＭＢＥＲ」である。いくつかの実施例において、数値は、８バイトのＤＯＵＢＬＥとして表現される。この目的のため、値「１０００００」は、ボックス７９６に示されているとおり、１６進値「０ｘ４０　０ｆ８　０ｘ６ａ　０ｘａ０　０ｘ００　０ｘ００　０ｘ００　０ｘ００」と等価である。

【０２０１】

「ｅｄｇｅ　Ｎａｍｅ」及び「ｐｕｓｈＴｙｐｅ」データは、それぞれ対応のフィールド定義７３０及び７３２によって定義されるとおり、文字列であり、前述の方法でエンコードされ得る。

【０２０２】

いくつかの実施例において、行データ７２４のバイナリ表現は、メッセージ６０２の全体に対するメッセージ構成体３０８を形成するために付加される。

【０２０３】

当然、バイナリ動的ＲＥＳＴメッセージを生成する他の方法も採用され得る。

【０２０４】

メッセージパーティショニング及びマルチパートメッセージ操作

ここで図４に戻ると、図３Ａに関して説明されているとおり、シリアル化バイナリストリームを生成していたシリアル化されたバイナリ動的ＲＥＳＴ　ＡＰＩ　３０２の後続である、ＡＰＩプロトコルライブラリ２０４のメッセージパーティショニングモジュール３３４は、エンコードされたメッセージ４０２を多数のチャンク化されたメッセージ４０６にパーティション化することができる。

【０２０５】

パーティション操作は、それぞれの接続に少量の固定メモリを使用する、大きいメッセージの送信を容易にし得る。いくつかの実施例において、メッセージパーティショニングモジュール３３４は、所定のメッセージをチャンク化又はパーティション化して、トランスポートレベルではなくアプリケーションレベルで、そのチャンク又はパーティションを再組み立てする。

【０２０６】

また、ＡＰＩプロトコルライブラリ２０４は、文字列３２２の長さ記述フィールド又は

カウントフィールド３２８などのメタデータを、所定のメッセージに挿入（例えば、追加、付加）して、完全なメッセージが使用可能ではない状態で部分的なメッセージ及びそのコンテンツの解析を可能にし得る。この目的のため、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２によって生成される所定のメッセージは、チャンクにパーティション化され得、チャンクのそれぞれは、完全なメッセージが受信される前にデコードされる。いくつかの実施例において、所定のメッセージに関連付けられたチャンクは、そのメッセージのすべてのチャンクが受信されると解析される。

【０２０７】

図８は、本発明の一実施形態による、エンコードされたメッセージをパーティション化する例示的方法を示す。メッセージパーティショニングモジュール３３４（図３Ａ）は、エンコードされたメッセージ４０２を多数のチャンク化されたメッセージ４０６にパーティション化し得る。いくつかの実施例において、メッセージの長さは、固定であっても可変であってもよい。

【０２０８】

いくつかの実施例において、メッセージパーティショニングモジュール３３４は、パーティショニングに関するメタデータ情報をヘッダ（例えば、マルチパートヘッダ３４８）に挿入し得る。図３Ｆ～３Ｇは、本発明の一実施形態による、ＡＰＩプロトコルライブラリ２０４で使用される、マルチパートヘッダ３４８の例示的構造を示す。いくつかの実施例において、マルチパートヘッダ３４８は、チャンクメッセージ識別番号３５０ａ、メッセージを形成するチャンク３５０ｂの合計数、及び所定のチャンク化されたメッセージのチャンクサイズ３５０ｃを含み得る。この目的のため、識別番号は、任意の順序でのチャンク部分のアセンブリに対応することから、チャンクメッセージは、任意の受信順序でデコードされ得る。また、チャンクメッセージの合計数は、アセンブラが、送信を完了するために必要とされる及び／又は予期されるチャンクメッセージの数を追跡することを可能にする。また、チャンクサイズは、チャンク化されたメッセージが、サイズ識別子によって定義された可変長であることを可能にする。

【０２０９】

特定の実施例において、ルーティング情報３４２ｄ（図３Ｇ、及び３Ｄも参照のこと）は、プラットフォームサーバ１０２からアウトバウンドメッセージを送信するなど、メッセージのルーティングを支援するために、マルチパートメッセージ３４８に組み込まれる。他の実施例において、ルーティング情報はまた、インバウンドメッセージに組み込まれる。

【０２１０】

いくつかの実施例において、メッセージパーティショニングモジュール３３４は、好ましくは４Ｋバイトと１０２４Ｋバイトとの間、なお更に好ましくは８Ｋバイトと１６Ｋバイトとの間のサイズを有するバッファウィンドウを含む。大きいファイル８０４（図８を参照のこと）に対して、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２は、データ交換を処理するために、バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２とは無関係に動作することができる外部ファイル転送プロトコルを採用し得る。

【０２１１】

バイナリメッセージ内のメタデータ構成体

図４に戻ると、エンコードされたメッセージ４０２を１つ又は２つ以上のチャンク化されたメッセージ４０６にパーティション化するメッセージパーティショニングモジュール３３４の後続である、ＡＰＩプロトコルライブラリ２０４は、チャンク化されたメッセージ４０６をメッセージに関するメタデータ情報と一緒にパッケージ化する。チャンク化されたメッセージ４０６は、メッセージヘッダ３３８及びメッセージ本体３３９にパッケージ化される。ＡＰＩプロトコルライブラリ２０４は、例えば、メッセージヘッダ３３８及びメッセージ本体３３９にアイデンティティ及びターゲットメタデータ情報を追加し得る。

【０２１２】

ＡＰＩプロトコルライブラリ２０４は、いくつかの実施例において、ＡＰＩプロトコルライブラリ２０４によって定義された、暗号化モジュール４１０ａを使用してメッセージを暗号化し得る。暗号化された部分は、メッセージ全体又はメッセージの一部分を含み得る。

【０２１３】

メッセージヘッダ及び本体内のメタデータ情報を採用する、他の例示的実地態様は、２０１４年3月２１日に出願された、米国特許出願番号第１４／２２２，１２３号、題名「Ｓｙｓｔｅｍ ａｎｄ Ｍｅｔｈｏｄ ｏｆ Ｍｅｓｓａｇｅ Ｒｏｕｔｉｎｇ Ｕｓｉｎｇ Ｎａｍｅ－Ｂａｓｅｄ Ｉｄｅｎｔｉｆｉｅｒ ｉｎ ａ Ｄｉｓｔｒｉｂｕｔｅｄ Ｃｏｍｐｕｔｉｎｇ Ｅｎｖｉｒｏｎｍｅｎｔ」に記載されている。この出願の内容は、その全体が本明細書に参考として組み込まれる。

【０２１４】

図３Ｄは、本発明の一実施形態による、ＡＰＩプロトコルライブラリ２０４で使用される、メッセージヘッダ３３８の例示的構造を示す。いくつかの実施例において、ヘッダ３３８は、所定のメッセージに関連付けられた要求識別番号３４２ａを含み得る。いくつかの実施例において、要求識別番号３４２ａは、様々なデータ長及びエンディアンであり得るが、最初に最大有効数字（ＭＳＢ）が付く２４桁のバイナリ数字であってもよい。

【０２１５】

いくつかの実施例において、ヘッダ３３８は、特定のコンピューティングデバイス１０４に属する所定の名前識別子に関連付けられた、セッション識別番号３４２ｂを含む。いくつかの実施例において、セッション識別番号３４２ｂは、所定のコンピューティングデバイス１０４のバインディングパスを決定するため、所定の接続サーバ１１０によって使用される。いくつかの実施例において、セッション識別番号３４２ｂは、様々なデータ長及びエンディアンであり得るが、最初に最大有効数字（ＭＳＢ）が付く３２桁のバイナリ数字であってもよい。

【０２１６】

いくつかの実施例において、ヘッダ３３８は、特定の持続的接続２００のルーティング情報に関連付けられた、エンドポイント識別番号３４２ｃを含む。エンドポイント識別番号３４２ｃは、最初に最大有効数字（ＭＳＢ）が付く３２桁のバイナリ数字であってもよい。

【０２１７】

いくつかの実施例において、ヘッダ３３８は、「メソッドコード３４４」と呼ばれる、メッセージ型フィールド３４４を含む。要求型メッセージの場合、メッセージ型フィールド３４４は、要求型メッセージに対応するコードを含み得る。いくつかの実施例において、要求型メッセージは、ＨＴＴＰフレームワークに基づき、「ＧＥＴ」要求、「ＰＯＳＴ」要求、「ＰＵＴ」要求、又は「ＤＥＬＥＴＥ」要求を含む。

【０２１８】

いくつかの実施例において、メッセージヘッダ３３８及び本体３３９内のメタデータ情報は、「エンティティ名」３４２ｄ（データ又は要求のソースに対応）、「特性」フィールド３４２ｅ、「ターゲット名」３４２ｆ（データ又は要求の意図された受信者に対応）、及び多数のメッセージカウント３４２ｇを含む。

【０２１９】

いくつかの実施例において、ＡＰＩプロトコルライブラリ２０４は、複数のバイナリ動的ＲＥＳＴ　ＡＰＩ　３０２を使用して作製される、メッセージを生成する。

【０２２０】

ＡＰＩプロトコルライブラリによってエンコードされたメッセージのデコードの例

ここで図４を参照すると、いくつかの実施例において、クライアント側又はサーバ側アプリケーションは、ＷｅｂＳｏｃｋｅｔプロトコル、又はその類のものを採用して、対応するクライアント側又はサーバ側アプリケーションにＡＰＩプロトコルライブラリ２０４の出力メッセージ２０２を送信する。対応するクライアント側又はサーバ側アプリケーシ

ョンは、メッセージをデコードするために、ミラーリングＡＰＩプロトコルライブラリ２０４を実行し得る。

【０２２１】

受信されたメッセージ２０２をデコードするため、受信されたメッセージ２０２の暗号化された部分は、復号モジュール４１０ｂを介して復号され得る。示されているとおり、暗号化された部分は、メッセージ全体又はメッセージの一部分を含み得る。復号モジュール４１０ｂは、ヘッダ３３８及びチャンク化されたメッセージ４０６を有する、復号されたメッセージを戻し得る。

【０２２２】

ＡＰＩプロトコルライブラリ２０４は、ヘッダ３３８を受信し、ヘッダ３３８のメソッドコードフィールド３４４を解析して、メッセージ型を決定し得る。メソッドコードフィールド３４４がＨＴＴＰ要求コード３３０を含む特定のインスタンスでは、ＡＰＩプロトコルライブラリ２０４は、更なる処理のためにメッセージ本体３３９を送信し得る。

【０２２３】

要求メッセージをデコードするため、いくつかの実施例において、ＡＰＩプロトコルライブラリ２０４プロセッサは、マーカー識別子、長さ識別子、フィールド型識別子、及びフィールドカウント識別子を使用して、チャンク化されたメッセージ４０６を解析し、フィールド定義のグループを取得する。メッセージ構成体は、データ値の群を含み得る。メッセージの所定のエンコードされた部分を解析するために必要なすべての情報が、エンコードされたデータに直接先行するいくつかの実施例において、ＡＰＩプロトコルライブラリ２０４は、メッセージの全体を待機する必要なく、メッセージを小分け（又はチャンク）で解析する。メッセージがチャンク化されたインスタンスでは、部分的にデコードされたメッセージが、バッファ４１８に記憶される。メッセージが全体としてデコードされるにあたり、ＡＰＩプロトコルライブラリ２０４は、クライアント－サーバアプリケーション４０１ｂによって提供される、完全なメッセージを取得するように、サービスバッファ４１８又はクライアント－サーバアプリケーションに信号を送信し得る。

【０２２４】

全体としてのデコードを実行するために、第１のエンコードされたデータストリームは、既知のデータシェイプで編成され得、また、データストリーム内の非制限要素を記述するためのメタデータを含み得る。いくつかの実施例において、メタデータは、非制限要素の前方記述子を含み得る。非制限要素は、可変数の要素を有するオブジェクトを含み得る。これには、テキスト情報（例えば、テキストの文字列）、並びに可変数のオブジェクトが含まれる場合がある。非制限要素はまた、メッセージ全体を構成する、可変数のチャンク化されたメッセージ部分を含み得る。メタデータは、様々なデータシェイプ間の区切りを識別するためのマーカーを含み得る。

【０２２５】

互いに独立した部分的メッセージのデコードが可能である状態において、クライアント側アプリケーションは、軽量アプリケーションとして実装され得る。小型固定デコードバッファは、典型的に、所定のデバイスに対する実装コストを削減する。

【０２２６】

図９は、本発明の一実施形態による、メッセージ４０４を再アセンブルする例示的方法９００のフローチャートである。エッジサーバ１０６は、第１のエンコードされたデータストリーム９０２（メッセージ４０４の一部である）を受信し得、ストリーム９０２を全体としてデコードする。エッジサーバ１０６は、チャンク化されたメッセージの数に対応する識別子９０４で、メッセージのデコードされた部分を索引付けし得る。識別子９０４は、マルチパートヘッダ４０５に採用されている現在のチャンク識別番号４０５ａであってもよい（図３Ｆ及び３Ｇを参照のこと）。

【０２２７】

エッジサーバ１０６は、後続のエンコードされたデータストリーム９０６、９０８、及び９１０を続けて受信し得る。エッジサーバ１０６は、メモリ内のメッセージ４０４のデ

コードされた部分９０６、９０８、及び９１０を続けて索引付けする。
【０２２８】
メッセージチャンク９１２が破壊されている、又は受信されていない場合、エッジサーバ１０６は、欠落しているチャンクメッセージ１１１２の再送信を待機し得る。チャンク識別子１１０４を使用することで、エッジサーバ１０６は、メッセージ４０４内のその順序から離れて、所定の受信されたエンコードされたデータストリームをデコードし得る。
【０２２９】
あるいは、ＡＰＩプロトコルライブラリ２０４が、エンコーダ又はデコーダとして採用され得る。図５は、本発明の他の実施形態による、ＡＰＩプロトコルライブラリ２０４を使用した例示的メッセージ送信及び受信のブロック図である。この実施形態では、クライアント側又はサーバ側アプリケーションは、メッセージ４０２をＡＰＩプロトコルライブラリ２０４に送信する。メッセージ４０２は、人間が読むことのできるテキスト言語フォーマットでフォーマット化され得る。例として、ＪＳＯＮ及びＸＭＬが挙げられる。メッセージ４０２は、１つ又は２つ以上のデータオブジェクトによって区切られたデータオブジェクトを含み得る。
【０２３０】
バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２は、図７Ａ～７Ｃに表示及び説明されているとおり、
記憶された構造記述に従って、メッセージ４０２を通して解析して、１つ又は２つ以上のデータオブジェクトを取得し得る。バイナリ動的ＲＥＳＴ　ＡＰＩ　３０２は、バイナリデータストリームを作製するために、コードブックに定義されたバイナリシンボルに、解析されたデータオブジェクトの要素のそれぞれを逐次的にマップし得る。
【０２３１】
ＡＰＩプロトコルライブラリのＳＤＫの例
図１０は、ソフトウェア開発キット（ＳＤＫ）を実行し、ＡＰＩプロトコルライブラリ２０４を使用する、クライアントアプリケーション搭載の例示的コンピューティングデバイス１０４のブロック図である。ＳＤＫは、７５Ｋバイト未満のフラッシュメモリ１００２上に存在し得る。いくつかの実施例において、ＳＤＫは、暗号化機構を含む、アプリケーション及びプロトコルの実装を含む。
【０２３２】
通信操作
図１１は、複数の通信プロトコルにわたる通信の例示的方法のフローチャートである。方法は、例えば、ＡＰＩサーバ１１０で実行されるＡＰＩプロトコルライブラリ２０４と連携する、ソフトウェアアプリケーションによって実行され得る。
【０２３３】
ＡＰＩプロトコルライブラリ２０４は、他の通信プロトコルと柔軟に相互作用する、動的ＲＥＳＴ　ＡＰＩインターフェースを提供する。
【０２３４】
いくつかの実施例において、方法は、ＡＰＩサーバ１１０で実行され得る。この目標を達成するため、レガシーセンサー及びコンピューティングデバイスは、通信及び制御という面においてアップグレードし得る。
【０２３５】
図１２は、本発明の一実施形態による、自己記述型メッセージを解析するための例示的方法１２００のフローチャートである。方法は、メモリに、バイナリヘッダ及びバイナリ本体の第１の構造記述（工程１２０２）と、メタデータ構成体の第２の構造記述と、を記憶することを含み得る。メタデータ構成体は、メッセージ構成体を形成する１つ又は２つ以上のデータ値の所定のデータ値に対応する、記述値の１つ又は２つ以上の群を含む（工程１２０４）。記述値は、データ値名記述子、データ値記述記述子、及びデータ値型記述を含み得る。
【０２３６】

いくつかの実施例において、方法１２００は、複数のメッセージを表現する複数のバイナリデータストリームを、ポートで受信することを含む（工程１２０６）。

【０２３７】

いくつかの実施例において、方法１２００は、バイナリヘッダ及びバイナリ本体を決定するため、第１の構造記述を使用して複数の受信されたバイナリデータストリームのそれぞれを、プロセッサを使用して解析することを含み得る（工程１２０８）。

【０２３８】

いくつかの実施例において、方法は、メタデータ構成体を形成する記述値の１つ又は２つ以上の群を決定するため、第２の構造記述を使用して解析されたバイナリ本体を、プロセッサを使用して解析することを含み得る。プロセッサは、メタデータ構成体の決定された記述値の一部を使用して、メッセージ構成体のデータ値の１つ又は２つ以上の群を決定する（工程１２１０）。メッセージ構成体の１つ又は２つ以上のデータ値のそれぞれと、メタデータ構成体の１つ又は２つ以上の記述値のそれぞれと、はバイナリマーカーによって詳述され得る。

【０２３９】

図１３は、本発明の一実施形態による、ＷｅｂＳｏｃｋｅｔを使用して動的ＲＥＳＴメッセージを送信する例示的方法１３００のフローチャートである。

【０２４０】

いくつかの実施例において、方法１３００は、バイナリ動的ＲＥＳＴ　ＡＰＩモデルを、デバイスのメモリで記憶することを含む（工程１３０２）。動的ＲＥＳＴ　ＡＰＩモデルは、データオブジェクト、イベントオブジェクト、サービスオブジェクト、プロパティオブジェクト、及び属性オブジェクトを含むデータモデルを含み得る。

【０２４１】

いくつかの実施例において、方法１３００は、デバイスのプロセッサを介し、バイナリ動的ＲＥＳＴ　ＡＰＩモデルを使用して要求メッセージをフォーマット化して、自己記述型要求メッセージを作製することを含む（工程１３０４）。

【０２４２】

いくつかの実施例において、方法１３００は、デバイスのポートを介して、持続的ステートレス接続で要求メッセージを送信することを含む（工程１３０６）。持続的ステートレス接続は、ＨＴＴＰ又はＨＴＴＰＳポートを使用して、ＷｅｂＳｏｃｋｅｔ接続を経由し得る。

【０２４３】

図１４は、本発明の一実施形態による、サードパーティプロトコルを使用した通信の例示的方法１４００のフローチャートである。方法１４００は、デバイスのメモリ内の第１の通信プロトコル及び第２の通信プロトコルを提供することを含む。第１の通信プロトコルは、自己記述型スキーマを有する多数の制御コード及びバイナリメッセージを含み得る。自己記述型スキーマは、データオブジェクトモデルであり得る。いくつかの実施例において、第２の通信プロトコルのインスタンスは、第１の通信プロトコルのインスタンスに関連付けてマップされ得る（工程１４０２）。データモデルは、動的ＲＥＳＴ　ＡＰＩオブジェクトモデルで構造化され得る。制御コードの関連マッピングは、ＨＴＭＬフレームワークに基づき得る。

【０２４４】

いくつかの実施例において、方法１４００は、第２の通信プロトコルにおいて、メッセージの第１のインスタンス化を、コンピューティングデバイスのポートで受信することを含む（１４０４）。

【０２４５】

いくつかの実施例において、方法１４００は、コンピューティングデバイスのプロセッサによって、メッセージの第１のインスタンス化を制御メッセージ又はデータメッセージのいずれかとして分類することを含む。方法１４００は、次いで、コンピューティングデバイスのプロセッサによって、メッセージの第１のインスタンス化を変換して、メッセー

ジの第２のインスタンス化を作製することを含む（工程１４０６）。メッセージが制御メッセージとして分類されると、プロセッサは、制御メッセージを１つ又は２つ以上の制御コードにマップし得る。図７に関して説明されているとおり、メッセージがデータメッセージとして分類されると、プロセッサは、自己記述型スキーマを有するバイナリメッセージにデータメッセージをマップし得る。

【０２４６】

いくつかの実施例において、方法１４００は、メッセージの第２のインスタンス化を、コンピューティングデバイスのポートで送信することを含む（工程１４０６）。

【０２４７】

図１５は、本発明の一実施形態による、チャンク化されたメッセージを使用した通信の例示的方法１５００のフローチャートである。

【０２４８】

いくつかの実施例において、方法は、第１のデータストリーム及び第２のデータストリームを含む、複数のデータストリーム間の１つ又は２つ以上のデータストリームを、ポートで受信することを含む（１５０２）。

【０２４９】

いくつかの実施例において、方法１５００は、受信された第１のデータストリーム及び受信された第２のデータストリームを、第１のバッファで記憶することを含む（工程１５０４）。

【０２５０】

いくつかの実施例において、方法１５００は、デコードされたメッセージの部分を、第２のバッファで記憶することを含む（工程１５０６）。

【０２５１】

いくつかの実施例において、方法１５００は、プロセッサを使用して、第１のバッファ内にバッファリングされた受信されたデータストリームをデコードして、所定のデコードされた部分を作製することを含む。プロセッサは、第２のバッファに所定のデコードされた部分を記憶し得る。プロセッサは、完全な第２のデータストリームを受信する前に、第１のバッファ内にバッファリングされた、受信された第１のデータストリームのデコードを開始し得る。

【０２５２】

コンピューティングデバイスの例

図１６は、本開示に記載された手法を実装するために使用することができる、コンピューティングデバイス１６００及びモバイルコンピューティングデバイス１６５０の例を示す。コンピューティングデバイス１６００は、ラップトップ、デスクトップ、ワークステーション、パーソナルデジタルアシスタント、サーバ、ブレードサーバ、メインフレーム、及びその他の適切なコンピュータなど、様々な形態のデジタルコンピュータを表すことが意図される。モバイルコンピューティングデバイス１６５０は、パーソナルデジタルアシスタント、携帯電話、スマートフォン、及びその他の類似のコンピューティングデバイスなど、様々な形態のモバイルデバイスを表すことが意図される。本明細書に示すコンポーネント、その接続及び関係、並びにその機能は、あくまで例であることを意味するものであり、制限されることを意味するものではない。

【０２５３】

コンピューティングデバイス１６００は、プロセッサ１６０２、メモリ１６０４、記憶デバイス１６０６、メモリ１６０４及び複数の高速拡張ポート１６１０に接続する高速インターフェース１６０８、並びに低速拡張ポート１６１４及び記憶デバイス１６０６に接続する低速インターフェース１６１２を含み得る。プロセッサ１６０２、メモリ１６０４、記憶デバイス１６０４、高速インターフェース１６０８、高速拡張ポート１６１０、及び低速インターフェース１６１２のそれぞれは、様々なバスを使用して相互接続され、また、共通マザーボードに、又は必要に応じてその他の方法でマウントされ得る。プロセッサ１６０２は、高速インターフェース１６０８に連結されたディスプレイ８１６などの外

部入力／出力デバイス上で、ＧＵＩに対応するグラフィック情報を表示するために、メモリ１６０４内に又は記憶デバイス１６０６上に記憶された命令を含む、コンピューティングデバイス１６００内の実行の命令を処理できる。他の実施例において、複数のプロセッサ及び／又は複数のバスは、必要に応じて複数のメモリ及びメモリの種類と共に使用され得る。また、複数のコンピューティングデバイスは、必要な操作の部分を提供するデバイスのそれぞれと（例えば、サーババンク、ブレードサーバの群、又はマルチプロセッサシステムとして）、接続され得る。

【０２５４】

メモリ１６０４は、コンピューティングデバイス１６００内の情報を記憶する。いくつかの実施例において、メモリ１６０４は、１つ又は２つ以上の揮発性メモリユニットである。いくつかの実施例において、メモリ１６０４は、１つ又は２つ以上の非揮発性メモリユニットである。メモリ１６０４は、また、磁気又は光ディスクなど、別の形態のコンピュータ読み取り可能媒体であってもよい。

【０２５５】

記憶デバイス１６０６は、コンピューティングデバイス１６００用の大容量記憶デバイスを提供することができる。いくつかの実施例において、記憶デバイス１６０６は、ストレージエリアネットワーク又はその他の構成内のデバイスを含む、フロッピーディスクデバイス、ハードディスクデバイス、光ディスクデバイス、若しくはテープデバイス、フラッシュメモリ若しくはその他の類似のソリッドステートメモリデバイス、又はデバイスのアレイなどの、コンピュータ読み取り可能媒体であり得る又は含み得る。命令は、情報担体に記憶され得る。命令は、１つ又は２つ以上の処理デバイス（例えば、プロセッサ１６０２）によって実行されると、前述のように、１つ又は２つ以上の方法を実行する。命令はまた、コンピュータ又はマシン読み取り可能媒体（例えば、メモリ１６０４、記憶デバイス１６０６、又はプロセッサ１６０２上のメモリ）などの１つ又は２つ以上の記憶デバイスによって記憶されてもよい。

【０２５６】

高速インターフェース１６０８は、コンピューティングデバイス１６００のために帯域幅を集中的に使用する動作を管理し、一方、低速インターフェース１６１２は、帯域幅をあまり集中的に使用しない動作を管理する。機能のこのような割り当ては、あくまでも例である。いくつかの実施例において、高速インターフェース１６０８は、メモリ１６０４と、ディスプレイ１６１６と（例えば、グラフィックプロセッサ又はアクセラレータを通じて）、各種の拡張カード（図示なし）を受容し得る高速拡張ポート１６１０と、に連結される。実施例において、低速インターフェース１６１２は、記憶デバイス８０６及び低速拡張ポート１６１４に連結される。各種の通信ポート（例えば、ＵＳＢ、Ｂｌｕｅｔｏｏｔｈ（登録商標）、イーサネット（登録商標）、ワイヤレスイーサネット（登録商標））を含み得る、低速拡張ポート１６１４は、キーボード、ポインティングデバイス、スキャナー、又はスイッチ若しくはルーターといったネットワーキングデバイスなどの、１つ又は２つ以上の入力／出力デバイスに、例えばネットワークアダプタを介して連結され得る。

【０２５７】

コンピューティングデバイス１６００は、図に示されているとおり、多数の異なる形態で実装され得る。例えば、標準サーバ１６２０として、又はそのようなサーバの群内で複数回実装され得る。加えて、ラップトップコンピュータ１６２２などのパーソナルコンピュータに実装され得る。また、ラックサーバシステム１６２４の一部として実装され得る。あるいは、コンピューティングデバイス８００からのコンポーネントは、モバイルコンピューティングデバイス１６５０など、モバイルデバイス（図示なし）の他のコンポーネントと結合され得る。このようなデバイスのそれぞれは、１つ又は２つ以上のコンピューティングデバイス１６００及びモバイルコンピューティングデバイス１６５０を含み得、システム全体は、互いに通信する複数のコンピューティングデバイスで構成され得る。

【０２５８】

モバイルコンピューティングデバイス１６５０は、他のコンポーネント間のプロセッサ１６５２、メモリ１６６４、ディスプレイ１６５４などの入力／出力デバイス、通信インターフェース１６６６、及びトランシーバ８６８を含み得る。モバイルコンピューティングデバイス１６５０はまた、追加の記憶域を提供するため、マイクロドライブ又は他のデバイスなどの、記憶デバイスと共に提供され得る。プロセッサ１６５２、メモリ１６６４、ディスプレイ１６５４、通信インターフェース１６６６、及びトランシーバ１６６８のそれぞれは、様々なバスを使用して相互接続され、コンポーネントのいくつかは、共通のマザーボードに、又は必要に応じて他の方法でマウントされ得る。

【０２５９】

プロセッサ１６５２は、メモリ１６６４に記憶された命令を含む、モバイルコンピューティングデバイス１６５０内の命令を実行することができる。プロセッサ１６５２は、別個の複数のアナログ及びデジタルプロセッサを含むチップからなるチップセットとして、実行され得る。プロセッサ１６５２は、例えば、ユーザーインターフェースの制御、モバイルコンピューティングデバイス１６５０によって実行されるアプリケーション、及びモバイルコンピューティングデバイス１６５０による無線通信など、モバイルコンピューティングデバイス１６５０の他のコンポーネントの調製用に提供され得る。

【０２６０】

プロセッサ１６５２は、ディスプレイ１６５４に連結された制御インターフェース１６５８及びディスプレイインターフェース１６５６を通じて、ユーザーと通信し得る。ディスプレイ１６５４は、例えば、ＴＦＴ（薄膜トランジスター液晶ディスプレイ）ディスプレイ若しくはＯＬＥＤ（有機発光ダイオード）ディスプレイ、又はその他の適切なディスプレイテクノロジーであり得る。ディスプレイインターフェース１６５６は、グラフィック及びその他の情報をユーザーに提示するため、ディスプレイ１６５４を駆動するための適切な回路機構を備え得る。制御インターフェース１６５８は、ユーザーからのコマンドを受信し、それらをプロセッサ１６５２への送信用に変換し得る。加えて、外部インターフェース１６６２は、モバイルコンピューティングデバイス１６５０の他のデバイスとの近距離通信を有効にするため、プロセッサ１６５２との通信を提供し得る。外部インターフェース１６６２は、例えば、いくつかの実施例において有線通信用に、又は、他の実施例において無線通信用に提供され得、複数のインターフェースも使用され得る。

【０２６１】

メモリ１６６４は、モバイルコンピューティングデバイス１６５０内の情報を記憶する。メモリ１６６４は、１つ又は２つ以上のコンピュータ読み取り可能媒体、１つ又は２つ以上の揮発性メモリユニット、１つ又は２つ以上の不揮発性メモリユニットのうちの１つ又は２つ以上としてとして実装され得る。増設メモリ１６７４はまた、例えば、ＳＩＭＭ（シングルインラインメモリモジュール）カードインターフェースを含み得る、拡張インターフェース１６７２を介して、モバイルコンピューティングデバイス１６５０に提供及び接続され得る。増設メモリ１６７４は、モバイルコンピューティングデバイス１６５０用の特別な記憶容量を提供することも、又は更にモバイルコンピューティングデバイス１６５０用のアプリケーション若しくはその他の情報を記憶することもできる。特に、増設メモリ１６７４は、前述のプロセスを実施又は補足するための命令を含み得、また、セキュアな情報も含み得る。このように、例えば、増設メモリ１６７４は、モバイルコンピューティングデバイス１６５０用のセキュリティモジュールとして提供され得、また、モバイルコンピューティングデバイス１６５０のセキュアな使用を許可する命令を使用してプログラミングされ得る。加えて、セキュアアプリケーションは、ハッキング不可能な方法でＳＩＭＭカードに識別情報を入れるなど、付加的な情報と共にＳＩＭＭカードを介して提供され得る。

【０２６２】

メモリは、以下に説明するとおり、例えば、フラッシュメモリ及び／又はＮＶＲＡＭメモリ（不揮発性ランダムアクセスメモリ）を含み得る。いくつかの実施例において、命令は情報キャリアに記憶される。命令は、１つ又は２つ以上の処理デバイス（例えば、プロ

セッサ１６５２）によって実行されると、前述の方法など、１つ又は２つ以上の方法を実行する。命令はまた、１つ又は２つ以上のコンピュータ又はマシン読み取り可能媒体（例えば、メモリ１６６４、増設メモリ１６７４、又はプロセッサ１６５２上のメモリ）など、１つ又は２つ以上の記憶デバイスによって記憶され得る。いくつかの実施例において、命令は、例えば、トランシーバ１６６８又は外部インターフェース１６６２を介して伝搬信号で受信され得る。

【０２６３】

モバイルコンピューティングデバイス１６５０は、必要な場所にデジタル信号処理回路機構を含み得る、通信インターフェース１６６６を介して無線で通信し得る。通信インターフェース１６６６は、数ある中でも、ＧＳＭ（登録商標）音声コール（グローバルシステムフォーモバイル通信）、ＳＭＳ（ショートメッセージサービス）、ＥＭＳ（拡張メッセージジングサービス）、若しくはＭＭＳメッセージ（マルチメディアメッセージングサービス）、ＣＤＭＡ（符号分割多元接続）、ＴＤＭＡ（時間分割多元接続）、ＰＤＣ（パーソナルデジタルセルラー）、ＷＣＤＭＡ（登録商標）（広帯域符号分割多元接続）、ＣＤＭＡ２０００、又はＧＰＲＳ（汎用パケット無線サービス）など、様々なモード又はプロトコル下で通信を提供し得る。このような通信は、例えば、無線周波数を使用するトランシーバ１６６８を介して発生し得る。加えて、短距離通信は、Ｂｌｕｅｔｏｏｔｈ（登録商標）、Ｗｉ－Ｆｉ（商標）、又は他のそのようなトランシーバ（図示なし）を使用するなどして発生し得る。加えて、ＧＰＳ（グローバルポジショニングシステム）レシーバーモジュール１６７０は、モバイルコンピューティングデバイス１６５０に付加的なナビゲーション及び場所関連の無線データを提供することができ、そのデータが、必要に応じて、モバイルコンピューティングデバイス１６５０で実行するアプリケーションによって使用され得る。

【０２６４】

モバイルコンピューティングデバイス１６５０はまた、オーディオコーデック１６６０を使用して可聴に通信することができ、ユーザーが話す情報を受信して、それを使用可能なデジタル情報に変換することができる。オーディオコーデック１６６０は、同様に、例えば、モバイルコンピューティングデバイス１６５０のハンドセットにあるスピーカーを介してなど、ユーザー用に可聴音を生成することができる。このような音は、音声電話コールからの音を含み得、録音された音（例えば、音声メッセージ、音楽ファイルなど）を含み得、また、モバイルコンピューティングデバイス１６５０で動作するアプリケーションによって生成される音も含み得る。

【０２６５】

モバイルコンピューティングデバイス１６５０は、図に示されているとおり、多数の異なる形態で実装され得る。例えば、携帯電話１６８０として実装され得る。また、スマートフォン１６８２、パーソナルデジタルアシスタント、又はその他の類似のモバイルデバイスの一部として実装され得る。

【０２６６】

本明細書に記載のシステム及び手法の様々な実施例は、デジタル電子回路機構、集積回路機構、特に設計されたＡＳＩＣ（特定用途向け集積回路）、コンピュータハードウェア、ファームウェア、ソフトウェア、及び／又はそれらの組み合わせで現実化することができる。これらの様々な実施例は、少なくとも１つのプログラマブルプロセッサを含む、プログラマブルシステムで実行可能及び／又は解釈可能である、１つ又は２つ以上のコンピュータプログラムでの実施例を含む場合があり、特別な目的又は汎用であってよく、記憶システム、少なくとも１つの入力デバイス、及び少なくとも１つの出力デバイスからデータ及び命令を受信するため、並びにそれらにデータ及び命令を送信するために連結される。

【０２６７】

これらのコンピュータプログラム（プログラム、ソフトウェア、ソフトウェアアプリケーション、又はコードとしても知られる）は、プログラマブルプロセッサに対するマシン

命令を含み、高レベルの手続き型及び／若しくはオブジェクト指向プログラミング言語で、並びに／又はアセンブリ／マシン言語で実装されてもよい。本明細書で使用されているとおり、マシン読み取り可能媒体及びコンピュータ読み取り可能媒体という用語は、プログラマブルプロセッサにマシン命令及び／又はデータを提供するために使用される、任意のコンピュータプログラム製品、装置、及び／又はデバイス（例えば、磁気ディスク、光ディスク、メモリ、プログラマブルロジックデバイス（ＰＬＤ））を指し、これは、マシン読み取り可能信号としてマシン命令を受信する、マシン読み取り可能媒体を含む。マシン読み取り可能信号という用語は、マシン命令及び／又はデータをプログラマブルプロセッサに提供するために使用される任意の信号を指す。

【０２６８】

ユーザーとの相互作用の目的で提供するため、本明細書に記載されたシステム及び手法は、ユーザーに情報を表示するためのディスプレイ装置（例えば、ＣＲＴ（陰極線管）又はＬＣＤ（液晶ディスプレイ）モニタ）と、ユーザーがコンピュータに入力を提供するのに使用するキーボード及びポインティングデバイス（例えば、マウス又はトラックボール）と、を有するコンピュータに実装され得る。他の種類のデバイスも同様に、ユーザーとの相互作用の目的で提供するために使用され得、例えば、ユーザーに提供されるフィードバックは任意のセンサリーフィードバックの形式（例えば、視覚フィードバック、聴覚フィードバック、又は触覚フィードバック）であってよく、ユーザーからの入力は、音響、音声、又は触覚入力を含む、任意の形式で受信され得る。

【０２６９】

本明細書に記載のシステム及び手法は、バックエンドコンポーネント（例えば、データサーバとして）を含む、若しくはミドルウェアコンポーネント（例えば、アプリケーションサーバ）を含む、若しくはフロントエンドコンポーネント（例えば、ユーザーが本明細書に記載のシステム及び手法の実施例と相互作用するのに使用するグラフィカルユーザーインターフェース又はＷｅｂブラウザ）を含むコンピューティングシステム、又はそのようなバックエンド、ミドルウェア、若しくはフロントエンドコンポーネントの任意の組み合わせに実装され得る。システムのコンポーネントは、デジタルデータ通信の任意の形式又は媒体（例えば、通信ネットワーク）によって相互接続され得る。通信ネットワークの例としては、ローカルエリアネットワーク（ＬＡＮ）、ワイドエリアネットワーク（ＷＡＮ）、及びインターネットが挙げられる。

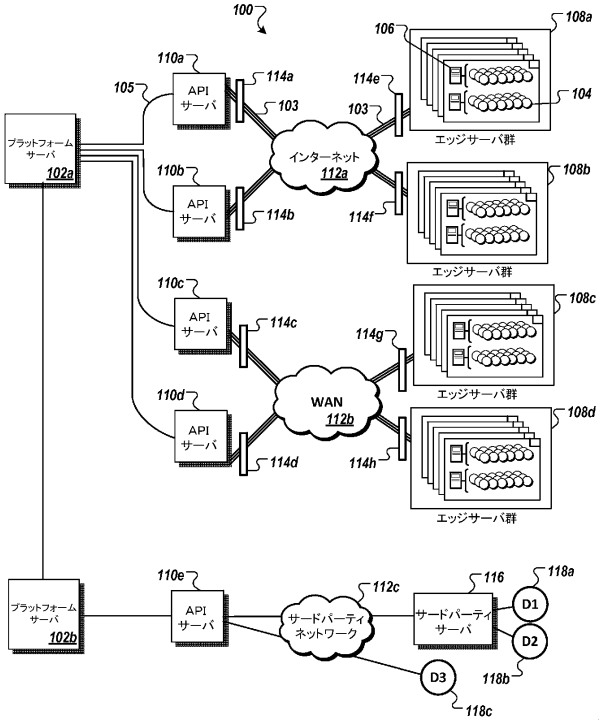
【０２７０】

コンピューティングシステムは、クライアント及びサーバを含み得る。クライアント及びサーバは概ね、互いに遠隔にあり、典型的に通信ネットワークを通じて相互作用する。クライアント及びサーバの関係は、対応のコンピュータで動作し、互いにクライアント－サーバ関係を有する、コンピュータプログラムによって生じる。

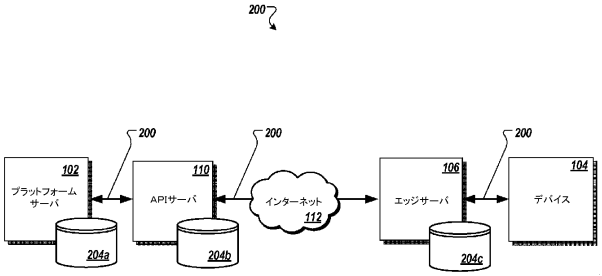
【０２７１】

本明細書に記載のシステム及び方法からなる構造、機能、及び装置の点から見て、いくつかの実施例において、動的ＲＥＳＴメッセージを使用した通信用のシステム及び方法が提供される。動的ＲＥＳＴメッセージを使用した通信を支持するための方法及び装置の特定の実施例を記述したことから、本開示の概念を組み込む他の実施形態を使用できることが、当業者には明らかになるであろう。したがって、本開示は、特定の実施例に制限されるべきではないが、むしろ、以下の請求項の趣旨及び範囲によってのみ制限されるべきである。

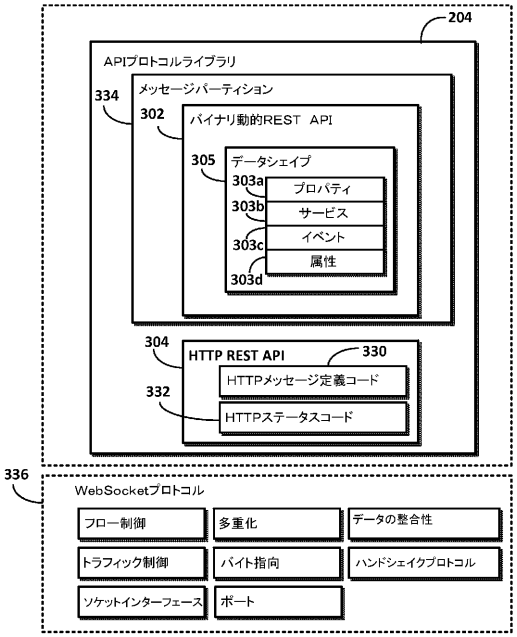
【図１】


100
プラットフォームサーバ
102a
API サーバ
110a
105
114a
103
インターネット
112a
106
108a
エッジサーバ群
104
110b
114b
114e
114f
108b
110c
114c
WAN
112b
114g
108c
110d
114d
114h
108d
プラットフォームサーバ
102b
110e
API サーバ
112c
サードパーティネットワーク
116
サードパーティサーバ
118a
D1
D2
D3
118b
118c

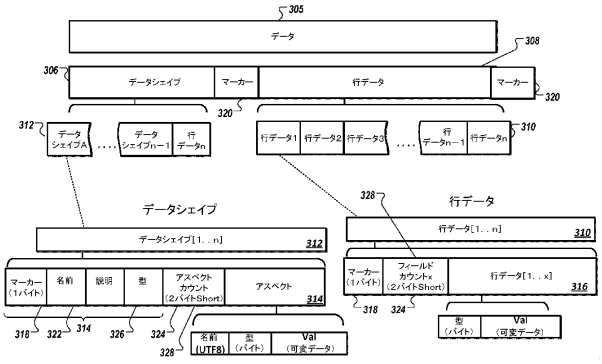
【図２】


200
102
プラットフォームサーバ
204a
110
APIサーバ
204b
インターネット
112
106
エッジサーバ
204c
104
デバイス

【図３Ａ】


204
APIプロトコルライブラリ
334
メッセージパーティション
302
バイナリ動的REST API
305
データシェイプ
303a
プロパティ
303b
サービス
303c
イベント
属性
303d
304
HTTP REST API
330
HTTPメッセージ定義コード
332
HTTPステータスコード
336
WebSocketプロトコル
フロー制御
多重化
データの整合性
トラフィック制御
バイト指向
ハンドシェイクプロトコル
ソケットインターフェース
ポート

【図３Ｂ】

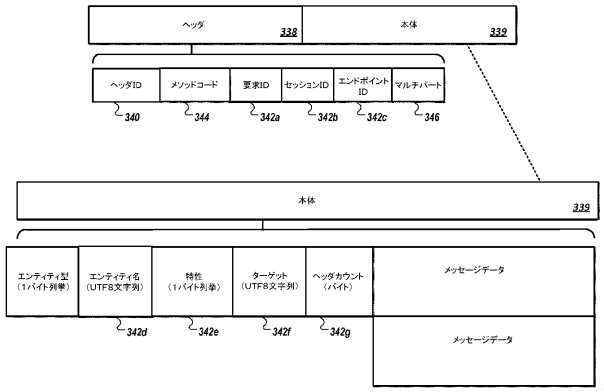

305
データ
308
306
データシェイプ
マーカー
行データ
マーカー
320
312
データシェイプA
データシェイプn−1
行データn
行データ1
行データ2
行データ3
行データn−1
行データn
310
データシェイプ
328
行データ
データシェイプ[1..n]
312
行データ[1..n]
310
マーカー（1バイト）
名前
説明
型
アスペクトカウント（2バイトShort）
アスペクト
314
フィールドカウントx（2バイトShort）
行データ[1..x]
316
318
322
326
324
名前（UTF8）
型（バイト）
Val（可変データ）
型（バイト）
Val（可変データ）

【図３Ｃ】

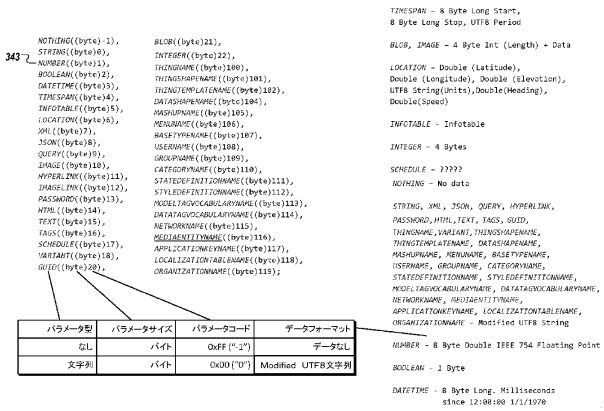
000 00001 (0x01)	GET
000 00010 (0x02)	*PUT*
000 00011 (0x03)	*POST*
000 00100 (0x04)	*DELETE*
000 01010 (0x0A)	*BIND*
000 01011 (0x0B)	*UNBIND*
000 11000 (0x14)	*AUTH*
000 11111 (0x1F)	*KEEP_ALIVE*
01000000 (0x40)	*STATUS_SUCCESS((short)200,(byte)0x40),*
01000001 (0x41)	*STATUS_CREATED((short)201,(byte)0x41),*
01000010 (0x42)	*STATUS_ACCEPTED((short)202,(byte)0x42),*
01000100 (0x44)	*STATUS_NO_CONTENT((short)204,(byte)0x44),*
01000110 (0x46)	*STATUS_PARTIAL_CONTENT((short)206,(byte)0x46),*
01100000 (0x60)	*STATUS_MULTIPLE_CHOICES((short)300,(byte)0x60),*
01100001 (0x61)	*STATUS_MOVED_PERMANENTLY((short)301,(byte)0x61),*
01100010 (0x62)	*STATUS_FOUND((short)302,(byte)0x62),*
01100011 (0x63)	*STATUS_SEE_OTHER((short)303,(byte)0x63),*
01100100 (0x64)	*STATUS_NOT_MODIFIED((short)304,(byte)0x64),*
01100101 (0x65)	*STATUS_USE_PROXY((short)305,(byte)0x65),*
01100111 (0x67)	*STATUS_TEMPORARY_REDIRECT((short)307,(byte)0x67),*
10000000 (0x80)	*STATUS_BAD_REQUEST((short)400,(byte)0x80),*
10000001 (0x81)	*STATUS_UNAUTHORIZED((short)401,(byte)0x81),*
10000010 (0x82)	*STATUS_PAYMENT_REQUIRED((short)402,(byte)0x82),*
10000011 (0x83)	*STATUS_FORBIDDEN((short)403,(byte)0x83),*
10000100 (0x84)	*STATUS_NOT_FOUND((short)404,(byte)0x84),*
10000101 (0x85)	*STATUS_METHOD_NOT_ALLOWED((short)405,(byte)0x85),*
10000110 (0x86)	*STATUS_NOT_ACCEPTABLE((short)406,(byte)0x86),*
10001000 (0x88)	*STATUS_REQUEST_TIMEOUT((short)408,(byte)0x88),*
10001001 (0x89)	*STATUS_CONFLICT((short)409,(byte)0x89),*
10010010 (0x92)	*STATUS_I_AM_A_TEAPOT((short)418,(byte)0x92),*
10010100 (0x94)	*STATUS_I_AM_BUZZED((short)420,(byte)0x94),*
10100000 (0xA0)	*STATUS_INTERNAL_ERROR((short)500,(byte)0xA0),*
10100001 (0xA1)	*STATUS_NOT_IMPLEMENTED((short)501,(byte)0xA1),*
10100010 (0xA2)	*STATUS_BAD_GATEWAY((short)502,(byte)0xA2),*
10100011 (0xA3)	*STATUS_SERVICE_UNAVAILABLE((short)503,(byte)0xA3),*
10100100 (0xA4)	*STATUS_GATEWAY_TIMEOUT((short)504,(byte)0xA4),*
11100000 (0xE0)	*STATUS_COMM_ERROR ((short)700,(byte)0xE0),*
11100001 (0xE1)	*STATUS_SERVER_REFUSED((short)701,(byte)0xE1),*
11100010 (0xE2)	*STATUS_SERVER_UNAVAILABLE((short)702,(byte)0xE2),*
11100011 (0xE3)	*STATUS_COMM_TIMEOUT((short)703,(byte)0xE3)*

330
332a
322b
322c

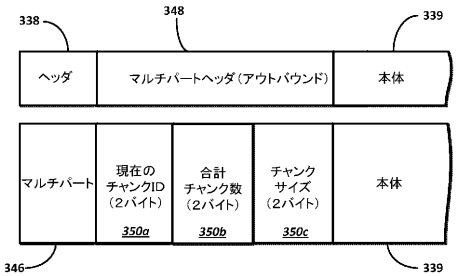
【図３Ｄ】



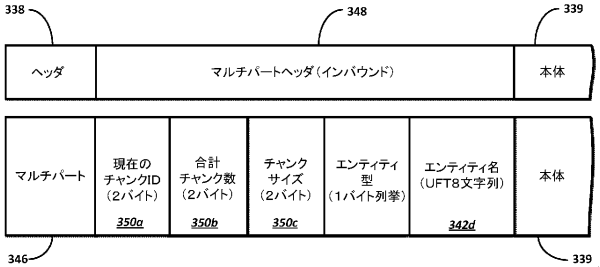
【図３Ｅ】



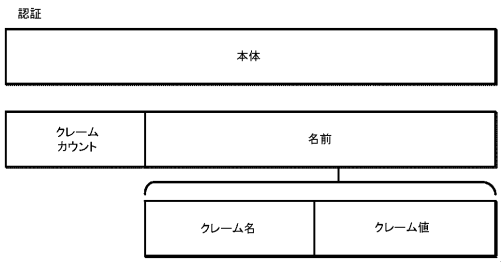
【図３Ｆ】



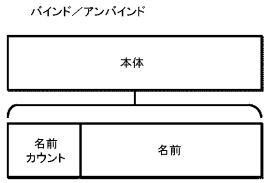
【図３Ｇ】



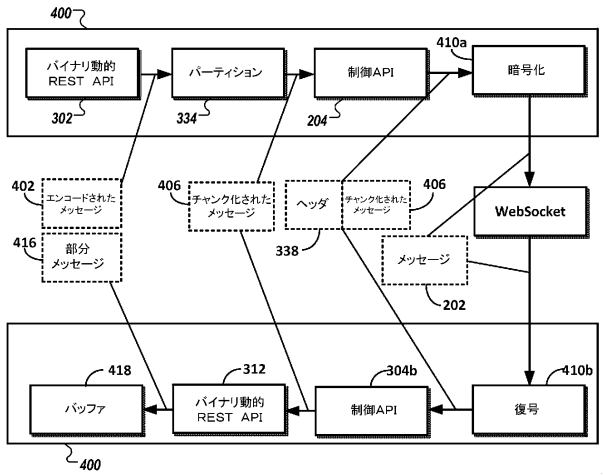
【図３Ｈ】



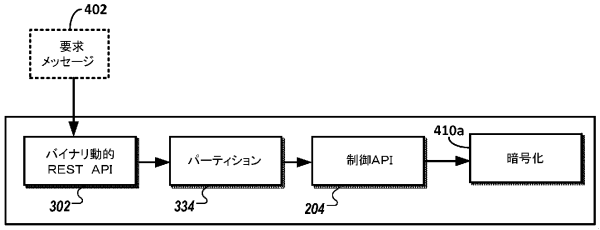
【図３Ｉ】



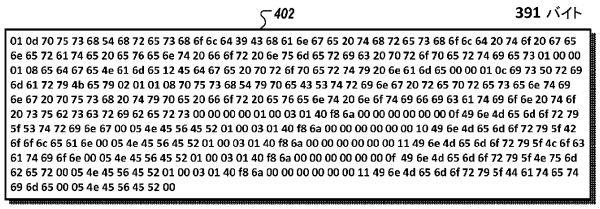
【図４】



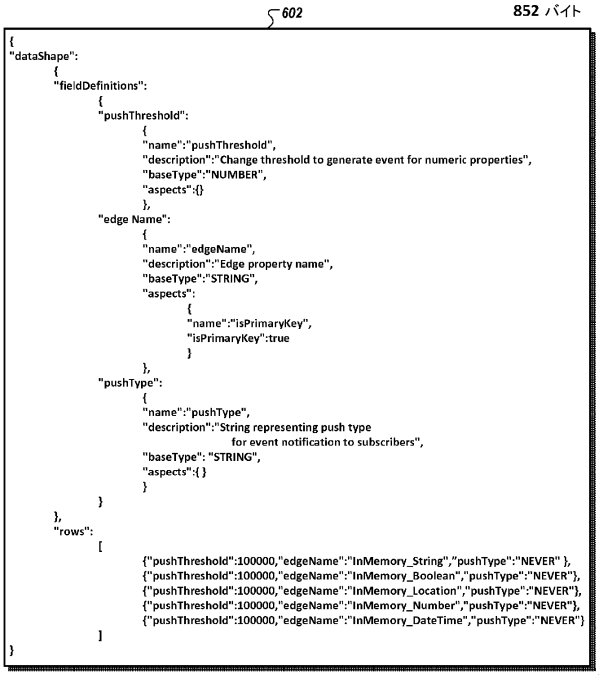
【図５】



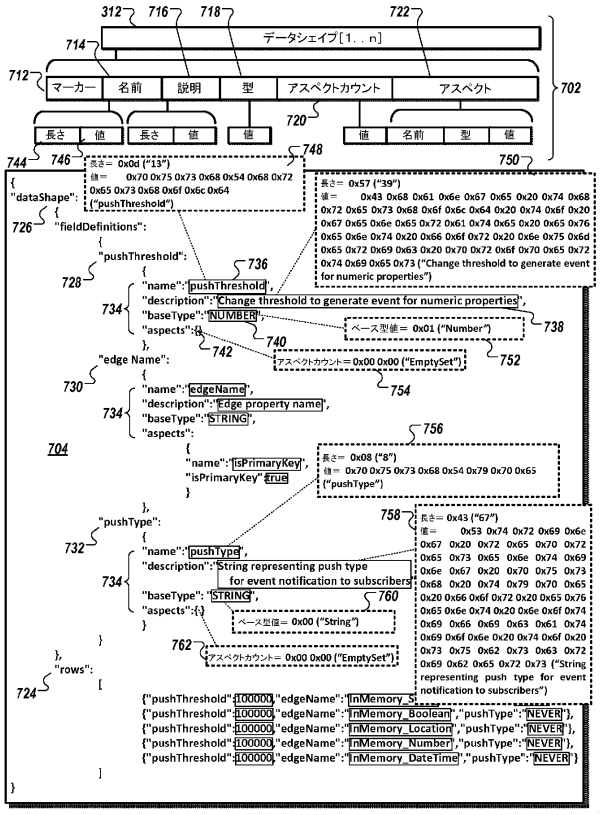
【図６Ａ】



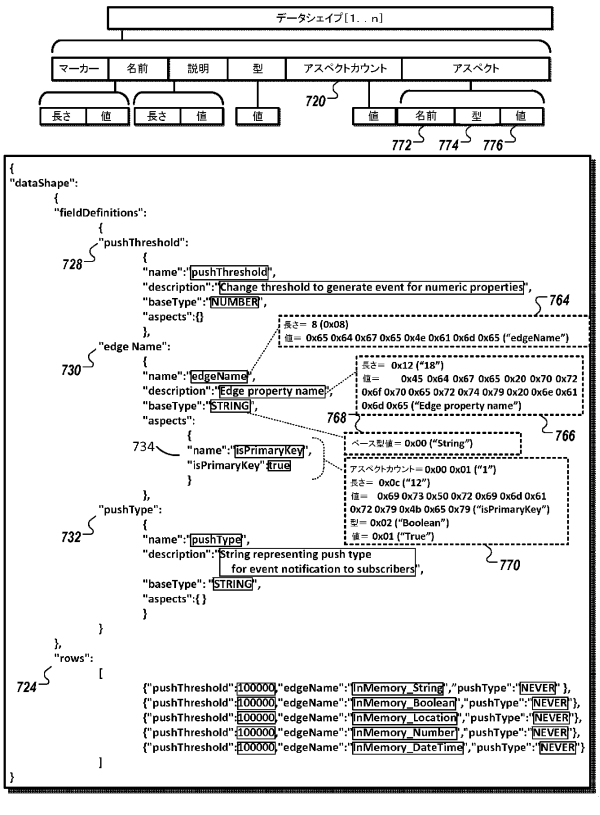
【図６Ｂ】



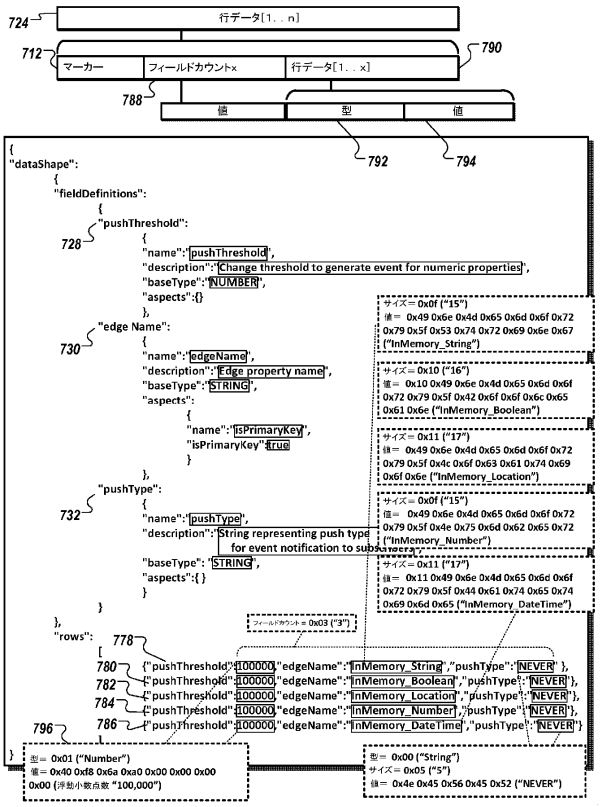
【図７Ａ】



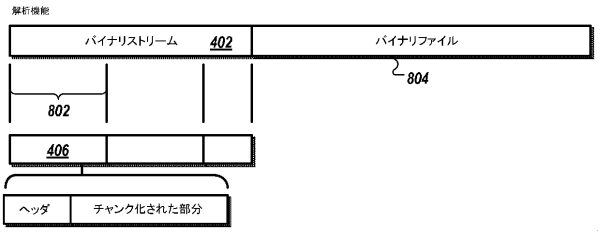
【図７Ｂ】



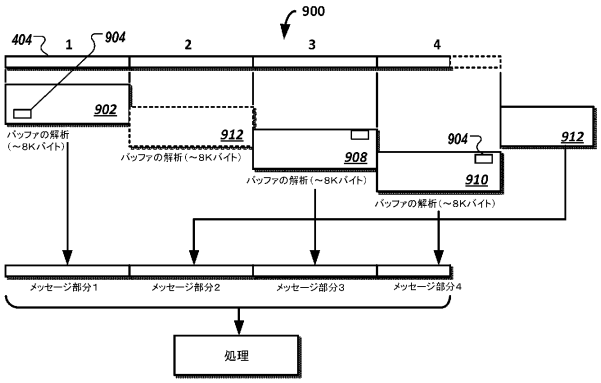
【図７Ｃ】



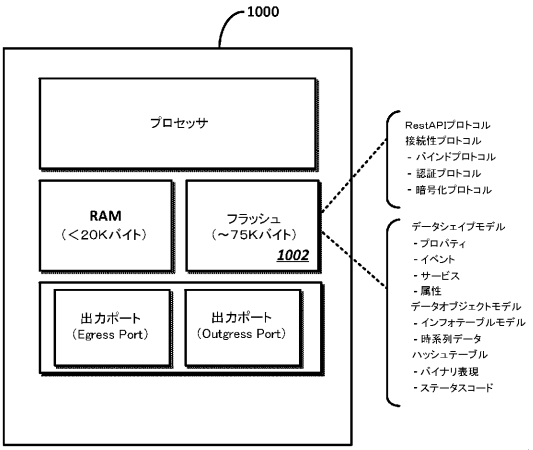
【図８】



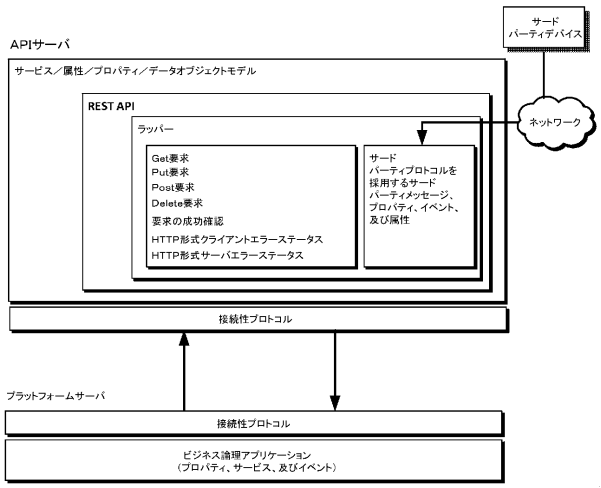
【図９】



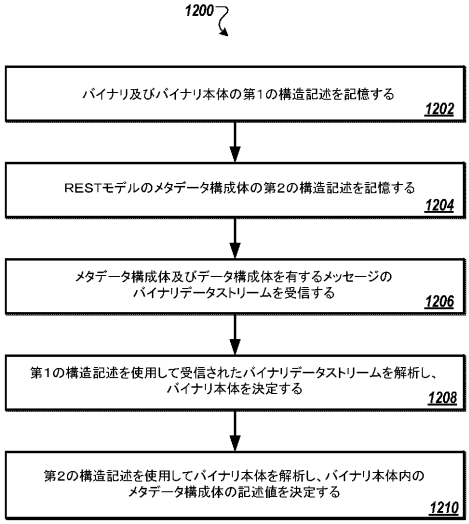
【図１０】



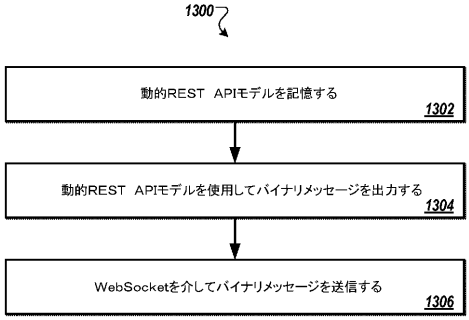
【図１１】



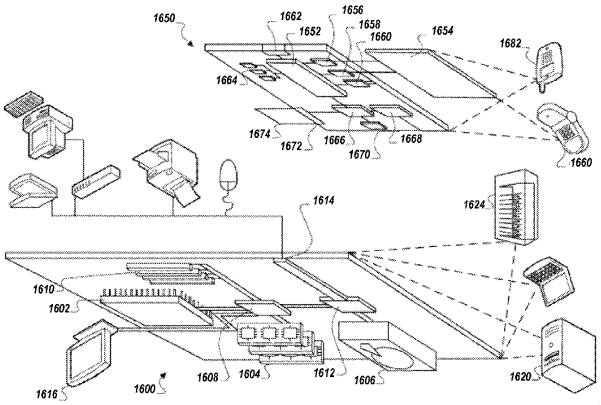
【図１２】



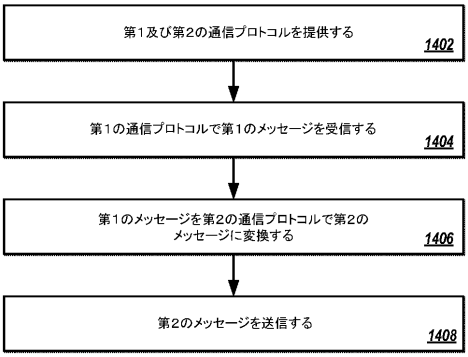
【図１３】



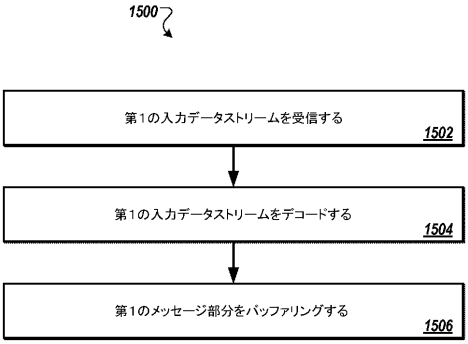
【図１６】



【図１４】



【図１５】



フロントページの続き

(31)優先権主張番号 14/222,067
(32)優先日 平成26年3月21日(2014.3.21)
(33)優先権主張国 米国(US)
(31)優先権主張番号 14/222,027
(32)優先日 平成26年3月21日(2014.3.21)
(33)優先権主張国 米国(US)

早期審査対象出願

(74)代理人 100109335
弁理士 上杉 浩
(74)代理人 100120525
弁理士 近藤 直樹
(72)発明者 バロッタ リック
アメリカ合衆国 ペンシルベニア州 １９４６０ フェニックスビル ウォーターフォール ウェイ ６１０
(72)発明者 カノーサ ジョン
アメリカ合衆国 マサチューセッツ州 ０２４９４ ニーダム ケンドリック ストリート １４０ ピーティーシー インコーポレイテッド内
(72)発明者 デレマー ボブ
アメリカ合衆国 マサチューセッツ州 ０２４９４ ニーダム ケンドリック ストリート １４０ ピーティーシー インコーポレイテッド内
(72)発明者 マホーニー マイク
アメリカ合衆国 マサチューセッツ州 ０２４９４ ニーダム ケンドリック ストリート １４０ ピーティーシー インコーポレイテッド内

審査官 北川 純次

(56)参考文献 特開２０１０－１６５２５０（ＪＰ，Ａ）
米国特許出願公開第２０１３／０２９０４４１（ＵＳ，Ａ１）

(58)調査した分野(Int.Cl.，ＤＢ名)
Ｇ０６Ｆ １３／００
Ｈ０４Ｍ １１／００