

(19) 日本国特許庁(JP)

(12) 特 許 公 報(B2)

(11) 特許番号

特許第4332436号  
(P4332436)

(45) 発行日 平成21年9月16日(2009.9.16)

(24) 登録日 平成21年6月26日(2009.6.26)

(51) Int.Cl.

F I

G 0 6 F 17/30 (2006.01)

G 0 6 F 13/00 (2006.01)

G 0 6 F 15/00 (2006.01)

G 0 6 F 17/30 1 1 O C

G 0 6 F 17/30 1 1 O F

G 0 6 F 17/30 1 2 O B

G 0 6 F 13/00 5 2 O D

G 0 6 F 15/00 3 9 O

請求項の数 1 (全 14 頁)

(21) 出願番号 特願2003-584957 (P2003-584957)  
 (86) (22) 出願日 平成15年4月11日(2003.4.11)  
 (65) 公表番号 特表2005-522786 (P2005-522786A)  
 (43) 公表日 平成17年7月28日(2005.7.28)  
 (86) 国際出願番号 PCT/US2003/011144  
 (87) 国際公開番号 W02003/088089  
 (87) 国際公開日 平成15年10月23日(2003.10.23)  
 審査請求日 平成16年10月26日(2004.10.26)  
 (31) 優先権主張番号 10/122,082  
 (32) 優先日 平成14年4月12日(2002.4.12)  
 (33) 優先権主張国 米国(US)

(73) 特許権者 508157727  
 フォトメディア テクノロジーズ エルエルシー  
 FotoMedia Technologies, LLC  
 アメリカ合衆国 03801 ニューハン  
 プシャー州 ポーツマス フリート スト  
 リート 155  
 (74) 代理人 100068755  
 弁理士 恩田 博宣  
 (74) 代理人 100105957  
 弁理士 恩田 誠  
 (74) 代理人 100142907  
 弁理士 本田 淳

最終頁に続く

(54) 【発明の名称】 プライベート・画像とメタデータのサーチ及び送受のためのネットワーク上のフォト・シェアリング・アーキテクチャ

(57) 【特許請求の範囲】

【請求項 1】

ネットワークに接続された少なくとも1つのサーバと多数のピア・ノードとを含むアクセス制限されたリソース共有システムにおいて、2以上のプライベート・リソース・レボジトリにわたってリソースのサーチを実行する方法であって、データデジタル画像を始めとするリソースを記憶し、且つ該リソースに関連するメタデータに基づいて該リソースをサーチするためのネットワークを該リソースは共有し、

リソース及び関連メタデータのストレージを各々のピア・ノードで維持する工程と、関連メタデータは、少なくとも1つのメタデータ・ボキャブラリに基づいていることと；

第1のピア・ノードが、サーバに、第1のピア・ノードに記憶された各々のリソースに関連するメタデータ・ボキャブラリと、プライベートのものとして指定された関連するメタデータ・ボキャブラリとを示す工程と；

前記サーバでの第2のピア・ノードからのサーチ・クエリに応じて、前記第1のピア・ノードによってサーバに示されたメタデータ・ボキャブラリをサーチ・クエリが使用するか否か、および使用されるメタデータ・ボキャブラリが前記第1のピア・ノードによってプライベートのものとして指定されたメタデータ・ボキャブラリの中にあるか否かを判定する工程；

第1のピア・ノードによって示されたメタデータ・ボキャブラリをサーチ・クエリが使用するが、前記第1のピア・ノードによってプライベートのものとして指定されたメタデータ・ボキャブラリの中にはないというサーバによる判定に応じて、サーバによりサーチ

10

20

・クエリを処理する工程と；

第1のピア・ノードによって示されたメタデータ・ボキャブラリをサーチ・クエリが使用し、かつ前記第1のピア・ノードによってプライベートのものとして指定されたメタデータ・ボキャブラリの中にあるというサーバによる判定に応じて、処理のために第1のピア・ノードにサーチ・クエリを渡すか、または第2のピア・ノードにサーチ・クエリを埋め込んだ第1のピア・ノードに対するロケータを渡す工程と；

から成る方法。

【発明の詳細な説明】

【技術分野】

【0001】

10

(関連出願の相互参照)

本発明は、2001年10月1日出願された、発明の名称「ネットワーク上のフォト・シェアリング・アーキテクチャ(NETWORK-BASED PHOTOSHARING ARCHITECTURE)」(2215P/P214)である米国特許出願出願番号第09/968,393号の一部継続出願である。この出願は、本願の譲受人に譲渡されており、引用により本願に組み込まれるものとする。

(発明の属する技術分野)

本発明はデジタル画像の電子記憶と共有に関し、より詳細には、フォト・シェアリングアーキテクチャの改良に関する。

【背景技術】

20

【0002】

過去数年にわたって、フォト・シェアリング技術は、フォト愛好家によって広く受け入れられるようになってきている。デジタル画像をサーバ上に記憶したり、他人がインターネットを介してそれを閲覧したりするために、ユーザが特定のサイトにデジタル画像をアップロードすることを可能にするウェブサイトが現在数多く存在する。メタデータは、通常は1つの画像又は複数の画像のグループに関連付けられるものであり、一般に、フォト・シェアリング・サイトによってサポートされている。しかしながら、今日のウェブ上でのフォト・シェアリングを阻害している最重要項目の1つは、画像とそれに関連するメタデータに利用可能なプライバシーが欠如していることである。

【0003】

30

ウェブ上で画像を共有するために利用可能な選択肢は、現在いくつか存在する。1つの選択肢は、1人のユーザ又は複数ユーザのグループが、自身の共有のためのサイトを構築し、従来の利用可能なアクセス制御機構を通じてそのサイトへのアクセスを制限することである。しかしながら、これにはコストがかかり、技術的にほとんどの人の手に負えないものとなりかねない。さらには、ユーザが2つ以上のそのような「プライベート」(private, 非公開)サイトにわたって画像をサーチすることを可能にする有効な機構は現在存在しない。

【0004】

もう1つの選択肢は、個人及びグループが、Yaga<sup>TM</sup>を始めとする現在のピアツーピアネットワークのいくつかに、多くのコストを負担せず、セットアップの際に非常に高い技術的専門知識を必要とせず、且つウェブサイトを維持する必要なく、自身の所有する画像をホストすることである。これらのピアツーピアシステムのうちの一部は、固定メタデータフィールドの小セットを使用してサーチするための限定的なサポートを提供するが、現在のピアツーピアネットワーク上で発見できる画像は、それらのメタデータと同様にパブリック(public、公開)なものであるため、システム上のすべてのユーザにアクセスが利用可能である。

40

【0005】

さらなる選択肢は、ユーザ及び小さなユーザーグループが、従来のウェブ上のフォト・シェアリング・サービスを使用して、自身の画像を共有することである。このようなサービスは、限られた量のプライバシーしか提供しない。従来のアクセス制御機構により、ユ

50

ーザ又はグループは、誰が画像とその関連メタデータを見てよいかを指定することができ、それらのサイトの一部は、そのサイトがサポートする限定された量のメタデータでのサーチを可能にするサーチ設備を提供する。しかしながら、現在のフォト・シェアリング・サービスは、画像とメタデータ（少なくともそのコピー）の両方の所有権を有している。この意味で、画像とメタデータはプライベートなものではない。実際、それらのサイトの大半に関する利用者規定は、画像とメタデータをプライベートにしておく責任を負わず、ほとんどの場合では、一旦画像がフォト・シェアリング・サイトにアップロードされると、画像とメタデータはいずれもフォト・シェアリング・サイトの所有物となることを規定している。

【発明の開示】

【発明が解決しようとする課題】

【0006】

従って、ユーザとグループが画像を共有し、且つ画像とメタデータへのアクセスを制限することを可能にするシステムが要求されている。さらに、そのようなシステムは、画像の所有者の意思に従って画像及びデータへのアクセスを制限するような様式で、ユーザが2以上のプライベート画像保存サイトにわたってサーチを実行することを可能にするようなものであるべきである。本発明はそのような要求を取り扱っている。

【課題を解決するための手段】

【0007】

本発明は、アクセス制限されたリソース共有システムにおいて、2以上のプライベート・リソース・レポジトリにわたってリソースのサーチを実行する方法である。システムは、ネットワークに接続された、少なくとも1つのサーバノードと多数のピア・ノードとを含んでいる。データデジタル画像を始めとするリソースが、リソースに関連するメタデータとマッチするタームを含むクエリの発行に基づいて、ノードからサーチされる。方法は、リソース及び関連メタデータのストレージを、各々のピア・ノードに維持する工程を含む。関連メタデータは少なくとも1つのメタデータ・ボキャブラリに基づいている。第1のピア・ノードが、サーバに、第1のピア・ノードに記憶された各々のリソースに関連するメタデータ・ボキャブラリと、プライベートのものとして指定された関連するメタデータ・ボキャブラリとを示す。サーバでの第2のピア・ノードからのサーチ・クエリに応じて、第1のピア・ノードによってサーバに示されたメタデータ・ボキャブラリをサーチ・クエリが使用するか否か、および使用されるメタデータ・ボキャブラリが第1のピア・ノードによってプライベートのものとして指定されたメタデータ・ボキャブラリの中にあるか否かを判定する。第1のピア・ノードによって示されたメタデータ・ボキャブラリをサーチ・クエリが使用するが、第1のピア・ノードによってプライベートのものとして指定されたメタデータ・ボキャブラリの中にはないというサーバによる判定に応じて、サーバによりサーチ・クエリを処理する。第1のピア・ノードによって示されたメタデータ・ボキャブラリをサーチ・クエリが使用し、かつ第1のピア・ノードによってプライベートのものとして指定されたメタデータ・ボキャブラリの中にあるというサーバによる判定に応じて、処理のために第1のピア・ノードにサーチ・クエリを渡すか、または第2のピア・ノードにサーチ・クエリを埋め込んだ第1のピア・ノードに対するロケータを渡す。

【0008】

本明細書に開示した方法及びシステムに従って、本発明はユーザに、ユーザのメタデータのプライバシーを維持しつつ、そのメタデータに基づく画像のサーチをシステムのすべてのノードにわたって実行可能にする方法を提供する。

【発明を実施するための最良の形態】

【0009】

本発明は、ウェブ・ベースのピアツーピアフォト・シェアリング・サービスを提供する方法及びシステムに関する。以下の説明は、当業者が発明を製造及び使用することを可能にするために示すものであって、特許出願とその要件に即して提供される。好ましい実施形態に関する種々の改変や、本明細書で説明している総括的な原理及び特徴は、当業者に

10

20

30

40

50

容易に理解されるだろう。したがって、本発明は、示した実施形態に限定されることは意図しておらず、本明細書で説明している原理及び特徴に適合する最も広い範囲に一致することを意図している。

【 0 0 1 0 】

本発明を、メタデータを適用するターゲットがデジタル画像である、好ましい実施形態に関して説明するが、メタデータは任意のタイプのデジタルリソースに適用し得る。

同時係属出願出願番号第 0 9 / 9 6 8 , 3 9 3 号は、ネットワーク中のすべてのワークステーションとコンピュータが、自身の画像を記憶すると共に、ネットワーク上の他のユーザに対するサーバとして機能する、ウェブ上のピア・ツー・ピアフォト・シェアリング・サービスを提供する。このフォト・シェアリング・サービスは、ピア・サーバとして知られる少なくとも 1 つの中心サーバを含んでいる。ピア・サーバは、クライアント・コンピュータ又はピア・ノードを通じてユーザに利用可能である。フォト・シェアリング・サービスは、ユーザが、自分のコンピュータ上で画像ストレージを維持することを可能にし、ユーザとゲストが、システムにサポートされた広範なメタデータに基づいて、別のユーザのピア・ノードにわたって画像をサーチすることを可能にする。フォト・シェアリング・サービスの利点は、ユーザが自分の独立したフォト・シェアリング・サイトをセットアップする必要性をなくし、サービス・プロバイダのフォト・シェアリングにより遭遇するストレージの問題を解決し、さらにはサービスのユーザにより遭遇するフォト・シェアリングの使い勝手の問題をも解決する。

【 0 0 1 1 】

図 1 は、本発明の好ましい実施形態による、ピアツーピア ( P 2 P ) フォト・シェアリングシステムを示すブロック図である。本発明によれば、システム 1 0 は、ピア・サーバ 1 4 を含む中心フォト・シェアリングウェブサイト 1 2 と、多数のピア・ノード 1 6 とを含んでいる。ピア・サーバ 1 4 と、ピア・ノード 1 6 の各々とは、インターネットを始めとするネットワーク上で互いに通信することができる。好ましい実施形態では、ユーザ 1 8 は、標準ウェブブラウザの使用により、ピア・ノード 1 6 でない装置又はクライアント ( 図示しない ) から、中心サイト 1 2 にアクセスすることが可能である。

【 0 0 1 2 】

好ましい実施形態では、ピア・ノード 1 6 はウェブサイトかコンピュータのいずれかを表わし、通常、特定のユーザ 1 8 のデジタル画像 2 0 を記憶している。ピア・ノード 1 6 に対するユーザインタフェースは多くの異なる方法で実装することが可能ではあるが、好ましい実施形態では、ピアユーザインタフェースはウェブブラウザとして実装され、代わりに、システム 1 0 のために特に設計されたアプリケーションであってもよい。各ピア・ノード 1 6 は、2 以上のユーザの画像 2 0 を記憶し得る。例えば、家庭用 P C を共有してはいるが自身の画像を別々に取り扱っている 2 人の家族のメンバーが、その共有 P C 上でシステム 1 0 に関する別々のアカウントを保持してもよい。デジタル画像 2 0 は、画像データを含む画像ファイルとして記憶される。各画像は、画像について記述且つ分類する、該画像に関連するメタデータ 2 2 をさらに有する。メタデータ 2 2 は、以下に説明するように、ユーザ 1 8 によって、又はピア・ノード 1 6 によって自動的に、画像 2 0 に関連付けられ得る。さらに、メタデータ 2 2 の一部又は全部は、画像の捕捉時にデジタルカメラによって画像 2 0 に関連付けられ得る。各画像 2 0 はさらに、サムネイル画像 2 4 と呼ばれる画像データのより小さな表示である特定のタイプのメタデータと関連付けられ得る。

【 0 0 1 3 】

フォト・シェアリング・サービス 1 0 は、ユーザ 1 8 が静止アルバム中に記憶するために自身のコンピュータからウェブサーバまで画像をアップロードすることによりデジタル画像をポストするという従来のフォト・シェアリングモデルとは対照的である。代わりに、上述の実施形態では、フォト・シェアリング・サービス 1 0、ピア・ノード 1 6 が、実際の画像データのストレージを維持し、各画像のメタデータ 2 2 ( 及び、特にサムネイル画像 2 4 ) のみがピア・サーバ 1 4 にアップロードされる。これにより、ユーザが、問題の画像 2 0 ( 又は画像のグループ、アルバム、サウンドクリップ、映画、メタデータを有

するものなら何でも)を見つけるためにピア・サーバ14に記憶されたメタデータ22をサーチするクエリを構築することが可能となる。

【0014】

例えば、ユーザ18は、メタデータ22に基づくサーチ基準の提示(submit)によって、画像20を閲覧するための画像アルバム26を動的に作成することができる。例えば図1で、ピア・サーバ14にメタデータ22をアップロードすることにより、ユーザ18aが中心サイト12上に共有画像20を有していると仮定する。その後、サーチ基準とマッチするメタデータを有する画像20を閲覧するために、ユーザ18bはピア・サーバ14にサーチを提示し得る。これに回答して、ピア・サーバ14は、サーチ基準とマッチする画像20に対する画像ロケータ(例えばURL)のリストをピア・ノード16bに返し、ピア・ノード16bは、必要に応じて、その画像ロケータを使用して、マッチしている画像を検索する要求を送る。

10

【0015】

サーチが行なわれる場所であるサーバ14にメタデータ22をキャッシュすることの欠点の1つは、ユーザ18が自身のメタデータ22に対する制御を失うことである。本発明は、ピア・ノード16又はピア・ノード16のグループが、ピア・サーバ14でサムネイルを始めとするいかなるメタデータ22をもキャッシュすることなく、画像20及びメタデータ22を記憶することを可能にする拡張機能を、フォト・シェアリングシステム10に供給することにより、この問題点を解決する。また、この拡張機能により、ユーザは、さらに以下に説明するように、自身のプライベート画像に関連するメタデータ・ボキャブラリを使用して、自身の1又は複数のピア・ノード16上の画像20をサーチすることもできる。

20

【0016】

本発明によれば、ピア・ノードの所有者は、自身のメタデータ22と画像20のプライバシーを守るために使用できる、2つのレベルのプライバシーを有する。いずれのレベルのプライバシーでも、画像20とメタデータ22はピア・サーバ14上でキャッシュされず、ピア・ノード16は、ピア・ノード16がプライベートメタデータ22及び画像20を含むことをサーバ14に示す。第1のレベルのプライバシーでは、ピア・ノード16は、どのメタデータ22のボキャブラリを、ピア・ノード16が記憶している画像20が使用するかを、ピア・サーバ14に示す。第2のレベルのプライバシーでは、ピア・サーバ14はどのメタデータ22のボキャブラリをピア・ノード上の画像20が使用するかを知らず、ピア・ノードにより高いセキュリティレベルを提供する。

30

【0017】

図2は、中心サイトのピア・サーバ14のコンテンツを示すブロック図である。好ましい実施形態では、ピア・サーバ14はウェブサーバ・アプリケーション50、メタデータ・ボキャブラリ・ライブラリ52、ユーザ・グループ・アカウント・データベース54、及びキャッシュ56を含む。

【0018】

ウェブサーバ・アプリケーション50は、ピア・ノード16の能力に適合するようにフォーマットされたページを与える。ウェブサーバ・アプリケーション50はフォーム駆動型のユーザインタフェース66を含むが、これはユーザ18に、根底にあるスキーマ言語を知るためのシンタックスを指定せずにカスタムのメタデータ・ボキャブラリ84を定義する、容易で直観的な方法を提供する。

40

【0019】

メタデータ・ボキャブラリ・ライブラリ52は、メタデータ・ボキャブラリ84又はスキーマを記憶及び管理するためのものである。ボキャブラリ・ライブラリ52は、ユーザ18によって作成されたカスタムのメタデータ・ボキャブラリ84と、特定の画像22に関連しクライアントコンピュータ16からアップロードされた実際のメタデータ値との両方を記憶する。

【0020】

50

好ましい実施形態では、ボキャブラリ・ライブラリ 52 は、ユニバーサル・スキーマ、共有スキーマ及びプライベート・スキーマを含み、これらは好ましい実施形態では R D F と X M L を使用して定義される。システム 10 中の画像 20 はすべて、ユニバーサル・スキーマによって指定されたメタデータ 22 と関連付けられていることが要求される。各メタデータ・ボキャブラリ 84 は、そのボキャブラリ中でメタデータ・プロパティを指定し、そのボキャブラリと適合するために遵守されなければならない制約を指定する。ユーザ 18 及びグループは、自身のスキーマを定義してもよく、それはユニバーサル・スキーマを含んでもよいし、他のボキャブラリ 84 から借用してもよい。

【0021】

キャッシュ 56 は、より頻繁にアクセスされる画像 20 に関連するメタデータ 22 を記憶するために使用され、より迅速なサーチを提供する。メタデータ 22 は、キャッシュ 56 において、ピア・サーバに構成されたキャッシュポリシーに基づいて、他の画像 20 由来のメタデータ 22 と自動的に取り替えられ得る。

【0022】

ユーザ・グループ・アカウント・データベース 54 は、各登録ユーザ 18 のユーザアカウント及び対応する連絡情報並びにプリファランスを記憶する。ユーザ・グループは、共通のポリシーをさらに共有してもよく、そのようなポリシーには、許可設定、U I オプション、必要な及び任意選択のメタデータ・ボキャブラリ、予約リスト、イベント／通知ポリシー、及びキャッシュポリシーが含まれ得る。

【0023】

ユーザ・アカウント・データベース 54 により、必須の (mandatory) ボキャブラリ 84 が、特定のターゲットリソースに関連づけられる。例えば、ある特定ユーザ 18 は、特定のメタデータ 22 のセットを有する自身のプライベート・フォトの全部を常に供給されることを望むとする。その人のアカウントは、適切なメタデータ・ボキャブラリ 84 をサポートするメタデータの割り当てが、画像 20 がシステム 10 上に記憶される前に要求されることを示すように構成されるだろう。要求されるメタデータの例として、アカウントの所有者についてのデータ (例えば名前、住所など) のボキャブラリ 84 がある。所与のターゲットタイプのためには、多数のボキャブラリ 84 が必要となり得る。

【0024】

デジタル静止画像 20 は、ただ一つのターゲットリソースのタイプである必要はない。必要なボキャブラリが指定され得る他のタイプの画像ファイルの例には、低速度撮影 (time-lapse) 画像、バースト画像、パノラマ画像のような複数画像のファイルが含まれる。音声ファイル、映画及びテキストドキュメントのような非画像のターゲットリソースもサポートされ得る。本発明は、考えられる限りメタデータが関連付けられたすべてのリソースに適用される。

【0025】

留意すべきは、ボキャブラリ・ライブラリ 52 の使用は、本発明の実施に必要なわけではなく、好ましいということである。この場合、各ユーザ・アカウント・レコードは、2つのレベルのプライバシーをサポートするために必要とされる必要な情報を含んでいる。

【0026】

本発明の好ましい実施形態によれば、ピア・サーバ 14 によって管理される各ユーザ・アカウント・レコードは、プライベート・データ・ボキャブラリ・リスト 90 及びプライベート・サーチ・インジケータ 92 を含んでいる。プライベート・データ・ボキャブラリ・リスト 90 は、どのメタデータ・ボキャブラリ 84 をピア・ノード 16 が使用するかを識別する。列挙された各メタデータ・ボキャブラリ 84 について、ユーザアカウントは、対応するプロパティのリスト (図示しない) も含むだろう。

【0027】

プライベート・サーチ・インジケータ 92 は、ユーザのピア・ノード 16 上に記憶された画像によってどのメタデータ・ボキャブラリ 84 が使用されるかを、ユーザが明らかにしたいか否かを示すために使用される。好ましい実施形態では、プライベート・サーチ・

10

20

30

40

50

インジケータ 9 2 は、ピア・ノード 1 6 がプライベートのメタデータ 2 2 を維持している場合には ( T R U E ) ( 真 ) に設定され、ピア・ノードのメタデータ 2 2 がパブリックである場合には ( F A L S E ) ( 偽 ) に設定される、ブール ( Boolean ) 型論理である。プライベート・サーチ・インジケータ 9 2 が T R U E で、プライベートのメタデータ 2 2 を示している場合、ピア・ノードのユーザは、そのプライベート・メタデータ 2 2 を保護するために 2 つのレベルのプライバシーのうちの 1 つを使用するという選択肢を有する。

#### 【 0 0 2 8 】

第 1 のレベルのプライバシーでは、どのメタデータ・ボキャブラリ 8 4 をピア・ノードがサポートするか ( 即ちどのボキャブラリ 8 4 がピア・ノード 1 6 上の画像 2 0 によって使用されるか ) を、ピア・ノード 1 6 がピア・サーバ 1 4 に指定する。その後、ピア・サーバ 1 4 によって受け取られ、且つそのようなボキャブラリ 8 4 を使用するサーチ・クエリが、ピア・ノード 1 6 に送られ、ピア・ノード 1 6 はサーチを取り扱う。その一方で、ピア・ノード 1 6 によってサポートされたボキャブラリ 8 4 からのプロパティを含まないサーチは、サーバ 1 4 によって処理される。この第 1 のレベルではプライバシーが幾分失われるが、ピア・ノード 1 6 によってサポートされたボキャブラリからのプロパティを含まないサーチが、処理のためにピア・ノード 1 6 に送られることがないので、実行が改善されるという利点がある。

#### 【 0 0 2 9 】

第 2 レベル ( より高いレベル ) のプライバシーでは、どのメタデータ・ボキャブラリ 8 4 をピア・ノード 1 6 がサポートするかを、ピア・ノード 1 6 はピア・サーバ 1 4 に指定しない。この場合、ピア・ノード 1 6 のユーザのためにサーバ上に維持されたプライベート・データ・ボキャブラリリスト 9 0 は、空になり、サーバ 1 4 は、従来のアクセス制御フィルタを通すすべてのサーチを、処理のためにピア・ノード 1 6 に渡すだろう。

#### 【 0 0 3 0 】

図 3 - 5 は、メタデータを使用してシステム 1 0 中に位置するリソースをサーチしつつ、ピア・ノード 1 6 上のプライベート・メタデータのプライバシーを保証する、3 つの異なる技術を示すフローチャートである。図 3 は、要求中のピア・ノード 1 6 及びサーチされている最中のピア・ノード 1 6 が、いずれも、ファイアウォールによって保護されていてもよいし保護されていなくてもよい、一般的なプライベート・メタデータのサーチ・検索プロセスの第 1 実施形態を示す。この実施形態では、要求中のピア・ノード 1 6 は、ウェブブラウザ又はクライアントアプリケーションを有する任意の電子機器であってよい。図 4 及び 5 は、図 3 に示したのと同じ機能を提供するが、特定のファイアウォール条件に適合した時に最適化 ( optimization ) を提供する、プライベート・メタデータのサーチ・検索プロセスの代替実施形態を示す。これらのプロセスは、図 3 に示した一般的方法よりも優れたパフォーマンスを生じ得る。

#### 【 0 0 3 1 】

ここで図 3 を参照すると、プライベート・メタデータ・サーチを可能にするプロセスは、ステップ 1 0 2 で、ユーザがシステム 1 0 における所望の画像又は他のリソースの位置を決定するようサーチ・クエリを構築することを可能にするスクリーンを、ピア・サーバ 1 4 がピア・ノード 1 6 に提示することから始まる。好ましくは、ピア・サーバ 1 4 は、システム 1 0 にサポートされたユーザ選択用のメタデータ・ボキャブラリ 8 4 のリストを表示する。ステップ 1 0 4 において、ユーザは、どのメタデータ・ボキャブラリ 8 4 をサーチに使用するかを選択し、そのボキャブラリに対応する問題のプロパティを選択し、システム 1 0 がマッチを見つけようと試みる当該選択されたプロパティに対する値を供給することにより、サーチ・クエリを構築する。

#### 【 0 0 3 2 】

ユーザがクエリの構築を終了したのに応じて、ピア・ノード 1 6 は、ステップ 1 0 6 で、ピア・サーバ 1 4 にそのクエリを提示する。その後、図 3 に示されるように、ピア・サーバはステップ 1 0 8 , 1 1 4 及び 1 2 0 で 3 つの別個のアクティビティ ( 任意の順番で、又は並行して ) を実行する。それらのステップは、各々の 3 つのアクティビティの各々

10

20

30

40

50

における最初のステップである。

【0033】

第1のアクティビティはステップ108で始まる。ステップ108で、ピア・サーバ14はピア・ノード16によってピア・サーバ14に送信されたメタデータ22を含むメタデータ・キャッシュ56をサーチする。クエリ・ストリングとマッチし、且つ問い合わせ中のユーザがアクセス権限を有する各リソースについて、ステップ110で、ピア・サーバ14は、リソースにアクセスするために要求中のピア・ノード16が使用するリソース・ロケータを作成する。ステップ112で、ピア・サーバ14は、ステップ108、114及び120で始まった3つのアクティビティが完了するのを待つ。

【0034】

第2のアクティビティはステップ114で始まる。ステップ114で、ピア・サーバ14はユーザ・アカウント・レコード54をサーチし、プライベート・メタデータ22を維持し、且つどのメタデータ・ボキャブラリ84をピア・ノード16のリソース(例えば画像)が使用するかを指定する、ピア・ノード16を見つける。ステップ116で、ピア・サーバ14はリストされたボキャブラリ84とサーチ・クエリをマッチさせる。ピア・サーバ14がマッチのあるユーザ・アカウント・レコードを見つけた場合、ステップ118で、最終処理のために、ピア・サーバはクエリに対応するピア・ノード16に送る。

【0035】

第3のアクティビティはステップ120で始まる。ステップ120で、ピア・サーバ14は、プライベート・メタデータ22がサポートされているがメタデータ・ボキャブラリ84がピア・サーバ14に識別されていないことを示す、すべてのユーザ・アカウント・レコード54の位置を特定する。各マッチするユーザアカウント54について、ピア・サーバ14は、ステップ118で、処理のためにクエリに対応するピア・ノード16に送る。

【0036】

各ピア・ノード16は、サーチ・クエリを受け取り、ステップ122で、リソースをマッチさせるために、そのプライベート・メタデータ22のデータベースをサーチする。各マッチするリソースについて、ステップ124でピア・ノード16はリソース・ロケータを作成し、ステップ126でそれをピア・サーバ14に返す。ピア・サーバ14はステップ112でそれらの応答を待つ。別の実施形態では、サーチ・クエリを処理したピア・ノード16は、ピア・サーバ14がサーチ・クエリを渡す場合にピア・サーバ14が他のピア・ノード16に要求中のピア・ノード16のURLを送ると仮定して、サーチを要求したピア・ノード16にいかなるリソース・ロケータをも直接返すこともできる。

【0037】

ステップ112で、ピア・サーバ14がピア・ノード16からのクエリに対する応答(又はタイムアウト要求)をすべて受け取る場合、ピア・サーバ14は、ステップ128で、要求中のピア・ノード16に、すべてのマッチするリソースに対するリソース・ロケータを送る。その後、要求中のピア・ノード16は、受け取ったリソース・ロケータを使用して、所望のデータをサーチする。

【0038】

注意:ピア・ノード16から返された情報を完璧に隠すために、ピア・ノード16はそれらの応答を暗号化しなければならない。好ましい実施形態では、これは、要求中のピア・ノード16に関連する公開鍵を使用して行われる。そのような鍵はピア・ノード16より多くの方法で得ることができる。好ましい実施形態では、要求中のピア・ノード16は、サーチ・クエリと共にピア・サーバ14へ鍵を送る。その後、ピア・サーバ14は、クエリを送る各ピア・ノード16に鍵を送る。別の実施形態では、公開鍵を、ピア・ノード16が検索できる既知の位置に記憶しておいてもよい。そのような既知のレポジトリの例は、LDAPディレクトリ、Versionを始めとする認証機関、及びピア・サーバ14自体である。各ピア・ノード16は、応答を、クエリ要求に暗号化するだろう。そのような要求は要求中のピア・ノードの秘密鍵でのみ解読することができる。

10

20

30

40

50

## 【 0 0 3 9 】

図 4 は、ファイアウォールによって保護されていないピア・ノード 1 6 のために最適化された、プライベート・メタデータ・サーチ・検索プロセスの第 2 実施形態を示すフローチャートである。図 3 に示したプロセスと同様に、このプロセスは、ピア・ノード 1 6 を保護するファイアウォールの存在にかかわらず機能する。しかしながら、このプロセスは、ほとんどの場合、図 3 で示した方法よりも、ファイアウォールの背後にないピア・ノード 1 6 の場合に、より優れたパフォーマンスを提供する。要求中のピア・ノード 1 6 はファイアウォールの背後にあってもよいし、なくてもよい。サーチ・検索プロセスは、クエリ応答がピア・サーバ 1 4 を通るように経路指定されず、むしろ、応答が要求中のピア・ノード 1 6 に直接送られるという点で、追加のプライバシーを提供する。また、ピア・サーバ 1 4 が多くの要求及び応答を処理している場合に、図 3 のプロセスによって処理されるクエリよりも、より優れたパフォーマンスを提供し得る。この方法のデータの暗号化は、今日最も一般に使用される方法（例えば SSL 接続）により提供することができる。

10

## 【 0 0 4 0 】

図 4 のサーチ・検索プロセスは、ステップ 2 0 2 で、システム 1 0 が、ユーザ 1 8 によるクエリの構築を可能にするスクリーンをユーザ 1 8 に提示することから始まる。ステップ 2 0 4 で、ユーザ 1 8 は、使用するメタデータ・ボキャブラリ 8 4 を選択し、問題のプロパティを選択し、システム 1 0 がマッチを見つけることを試みる当該プロパティに対する値を供給するにより、サーチ・クエリを構築する。ユーザがクエリの構築を終了したのに応じて、ピア・ノード 1 6 は、ステップ 2 0 6 で、ピア・サーバ 1 4 にクエリを提示する。次に、ピア・サーバ 1 4 は、ステップ 2 0 8、2 1 4 及び 2 2 0 で 3 つの別個のアクティビティ（任意の順序で、又は並行に）を実行する。これらのステップは、各々の 3 つのアクティビティの最初のステップである。

20

## 【 0 0 4 1 】

第 1 のアクティビティは、ステップ 2 0 8 で始まる。ステップ 2 0 8 で、ピア・サーバ 1 4 はピア・ノード 1 6 によってピア・サーバ 1 4 に送信されたメタデータ 2 2 を含むメタデータキャッシュ 5 6 をサーチする。クエリ・ストリングとマッチし、且つ問い合わせ中のユーザがアクセス権限を有する各リソースについて、ステップ 2 1 0 で、ピア・サーバ 1 4 は、リソースにアクセスするために要求中のピア・ノード 1 6 が使用するリソース・ロケータを作成する。ステップ 2 1 2 で、ピア・サーバ 1 4 は、ステップ 2 0 8、2 1 4 及び 2 2 0 で始まった 3 つのアクティビティを完了するのを待つ。

30

## 【 0 0 4 2 】

第 2 のアクティビティはステップ 2 1 4 で始まる。ステップ 2 1 4 で、ピア・サーバ 1 4 がユーザ・アカウント・レコード 5 4 をサーチし、プライベート・メタデータ 2 2 を維持し、且つどのメタデータ・ボキャブラリ 8 4 をピア・ノード 1 6 のリソース（例えば画像）が使用するかを指定する、ピア・ノード 1 6 をを見つける。ステップ 2 1 6 で、ピア・サーバ 1 4 はリストされたボキャブラリとサーチ・クエリをマッチさせる。ピア・サーバ 1 4 がマッチのあるレコードを見つけた場合、ステップ 2 1 8 で、ピア・サーバ 1 4 は要求中のクライアントによって送られたクエリを含むピア・ノード・ロケータを構築する。

40

## 【 0 0 4 3 】

ステップ 2 2 0 で、ピア・サーバ 1 4 は、プライベート・メタデータ 2 2 をサポートすることを示し、且つボキャブラリ 8 4 がピア・サーバ 1 4 に識別されていない、すべてのピア・ノード・レコードの位置を特定する。ピア・サーバ 1 4 は、サーチ・クエリ中にあるプロパティとマッチするプロパティを含むボキャブラリ 8 4 を有するピア・ノード 1 6 に対するレコードを見つけた場合、ステップ 2 1 8 で、ピア・サーバ 1 4 はそれらのピア・ノード 1 6 の各々をポイントする各ピア・ノード・ロケータを作成し、クエリをそのピア・ノード・ロケータ中に埋め込む。

## 【 0 0 4 4 】

ステップ 2 1 8 で、埋込まれたクエリを有するピア・ノード・ロケータをすべて構築し終えた後、ピア・サーバ 1 4 はステップ 2 1 2 の待機プロセスに、そのピア・ノード・ロ

50

ケータを供給する。

【 0 0 4 5 】

ピア・サーバ 1 4 が、ステップ 2 1 2 で、ピア・ノード・ロケータ（又はタイムアウト要求）をすべて受け取る場合、ステップ 2 2 2 で、ピア・サーバ 1 4 は、要求中のピア・ノード 1 6 に、ピア・ノード・ロケータを送る。その後、ステップ 2 2 4 で、要求中のピア・ノード 1 6 は、返されたピア・ノード・ロケータを使用して、リソース・ロケータで識別されたピア・ノード 1 6 にクエリを送る。（注意：ファイアウォールの背後にあるピア・ノード 1 6 は、クエリがピア・プロキシを通じて経路指定されるべきであることを各ピア・ノード・ロケータに示すことにより、サポートされ得る）。

【 0 0 4 6 】

リソース・ロケータのうちの 1 つの受け取りに応じて、各ピア・ノード 1 6 は、ステップ 2 2 6 で、そのメタデータデータベースをサーチし、クエリとマッチするリソースを見つける。見つかったマッチするリソースの各々について、ステップ 2 2 8 で、ピア・ノード 1 6 はピア・ノード・ロケータを作成する。ステップ 2 3 0 で、ピア・ノードは任意の作成されたピア・ノード・ロケータを、要求中のピア・ノード 1 6 に返す。最後に、ステップ 2 3 2 で、要求中のピア・ノード 1 6 は、ピア・ノード・ロケータを使用してリソースをサーチし、クエリの結果をユーザに提示する。

【 0 0 4 7 】

図 5 は、ファイアウォールによって保護されているピア・ノードのために最適化された、プライベート・メタデータ・サーチ・検索プロセスの第 3 実施形態を示すフローチャートである。この図 5 のプロセスは、ファイアウォールによって保護されているピアノード 1 6 とファイアウォールによって保護されていないピア・ノード 1 6 の両方に対して機能するが、ファイアウォールによって保護されていないピア・ノード 1 6 に対する図 4 のプロセスに対しては実益がない。要求中のピア・ノード 1 6 はファイアウォールで保護されてもよいし、保護されなくてもよい。図 4 のプロセスと同様に、このプロセスは、クエリ応答がピア・サーバ 1 4 を通るように経路指定されないという点で、図 3 で示されるプロセスに対して、追加のプライバシーを提供する。また、ピア・サーバ 1 4 が多くの要求及び応答を処理している場合に、図 3 の方法によって処理されるクエリよりも、より優れたパフォーマンスを提供し得る。この方法のデータの暗号化は、今日最も一般に使用される方法（例えば SSL 接続）により提供することができる。

【 0 0 4 8 】

先の 2 つの実施形態と同様、サーチ・検索プロセスは、スクリーンをユーザ 1 8 に提示し（ステップ 3 0 2 ）、ユーザ 1 8 がサーチ・クエリを構築する（ステップ 3 0 4 ）ことから始める。一旦要求中のピア・ノード 1 6 がピア・サーバ 1 4 にクエリを提出すると（ステップ 3 0 6 ）、ピア・サーバ 1 4 はステップ 3 0 8 , 3 1 6 , 3 2 2 で最初に開始された 3 つのアクティビティを実行する。

【 0 0 4 9 】

第 1 のアクティビティは、ステップ 3 0 8 で始まる。ステップ 3 0 8 で、ピア・サーバ 1 4 はピア・ノード 1 6 によってピア・サーバ 1 4 に送信されたメタデータ 2 2 を含むメタデータキャッシュ 5 6 をサーチする。クエリ・ストリングとマッチし、且つ問い合わせ中のユーザがアクセス権限を有する各リソースについて、ステップ 3 1 0 で、ピア・サーバ 1 4 は、リソースにアクセスするために要求中のピア・ノード 1 6 が使用するリソース・ロケータを作成する。ステップ 3 1 2 で、ピア・サーバ 1 4 は、要求中のピア・ノード 1 6 にすべてのリソース・ロケータを送る。

【 0 0 5 0 】

第 2 のアクティビティはステップ 3 1 6 で始まる。ステップ 3 1 6 で、ピア・サーバ 1 4 がユーザ・アカウント・レコード 5 4 をサーチし、プライベート・メタデータ 2 2 を維持し、どのメタデータ・ボキャブラリ 8 4 がピア・ノード 1 6 のリソース（例えば画像）を使用するかを指定する、ピア・ノード 1 6 をを見つける。ステップ 3 1 8 では、ピア・サーバ 1 4 はリストされたボキャブラリとサーチ・クエリをマッチさせる。ピア・サーバ 1

10

20

30

40

50

4 がマッチのあるユーザ・アカウント・レコードを見つけた場合、ステップ 3 2 0 で、ピア・サーバ 1 4 は対応するピア・ノード 1 6 に、要求中のピア・ノード 1 6 に対するリソース・ロケータと共に、クエリを送る。

【 0 0 5 1 】

第 3 のアクティビティはステップ 3 2 2 で始まる。ステップ 3 2 2 で、ピア・サーバ 1 4 は、プライベート・メタデータ 2 2 がサポートされ、且つメタデータ・ポキャブラリ 8 4 がピア・サーバ 1 4 に識別されていないことを示す、すべてのユーザ・アカウント・レコード 5 4 の位置を特定する。各マッチするユーザ・アカウント 5 4 について、ステップ 3 2 0 で、ピア・サーバ 1 4 は対応するピア・ノード 1 6 に、要求中のピア・ノード 1 6 に対するリソース・ロケータと共に、クエリを送る。

10

【 0 0 5 2 】

各ピア・ノード 1 6 は、サーチ・クエリを受け取り、ステップ 3 2 4 で、リソースをマッチさせるために、そのプライベート・メタデータ 2 2 のデータベースをサーチする。各マッチするリソースについて、ステップ 3 2 6 で、ピア・ノード 1 6 はリソース・ロケータを作成する。ステップ 3 2 8 で、各ピア・ノード 1 6 は、ピア・サーバ 1 4 から受け取った要求中のピア・ノード 1 6 のリソース・ロケータを使用して、要求中のピア・ノード 1 6 とのネットワーク接続を確立する。各ピア・ノード 1 6 はこの接続を使用して、要求中のピア・ノード 1 6 に対して作成したリソース・ロケータを送る。この接続は、必要な場合に、要求中のピア・ノード 1 6 が要求することを可能にするために、オープンにされる。

20

【 0 0 5 3 】

ステップ 3 1 4 で、要求中のピア・ノード 1 6 は、ピア・サーバ 1 4 及びピア・ノード 1 6 からすべてのリソース・ロケータを集める。要求中のピア・ノード 1 6 がすべてのリソース・ロケータを受け取るか、又は、タイムアウト期間が満了した後で、要求中のピア・ノード 1 6 はリソース・ロケータを使用して、クエリの結果をユーザに提示するために必要とされるデータをサーチする。

【 0 0 5 4 】

ユーザのメタデータ及び画像のプライバシーを守るピアツーピアフォト・シェアリング・サービスについて開示した。本発明は、示した実施形態に従って説明したが、当業者には実施形態に対するバリエーションがあり、任意のバリエーションが本発明の精神及び範囲にあることが容易に認識されるだろう。従って、当業者には、特許請求の範囲の精神及び範囲から逸脱することなく、多くの改変をなし得るであろう。

30

【図面の簡単な説明】

【 0 0 5 5 】

【図 1】本発明の好ましい実施形態によるネットワーク上のフォト・シェアリングシステムを示すブロック図。

【図 2】中心サイトのピア・サーバのコンテンツを示すブロック図。

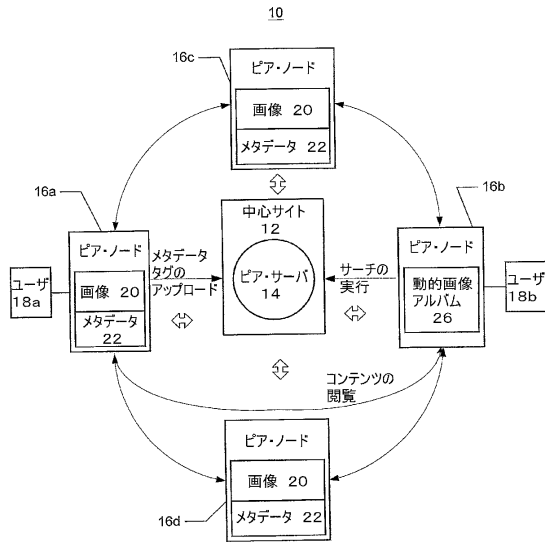
【図 3】メタデータを使用してシステム中に位置するリソースをサーチしつつ、ピア・ノード上のプライベートメタデータのプライバシーを保証する、3つのプロセスを示すフローチャート。

40

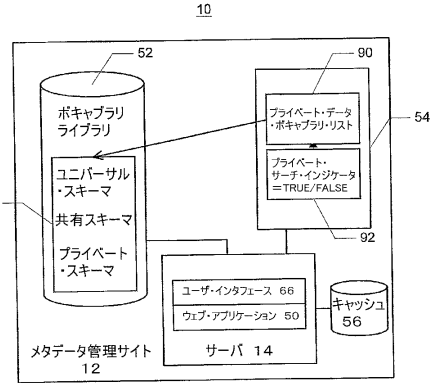
【図 4】メタデータを使用してシステム中に位置するリソースをサーチしつつ、ピア・ノード上のプライベートメタデータのプライバシーを保証する、3つのプロセスを示すフローチャート。

【図 5】メタデータを使用してシステム中に位置するリソースをサーチしつつ、ピア・ノード上のプライベートメタデータのプライバシーを保証する、3つのプロセスを示すフローチャート。

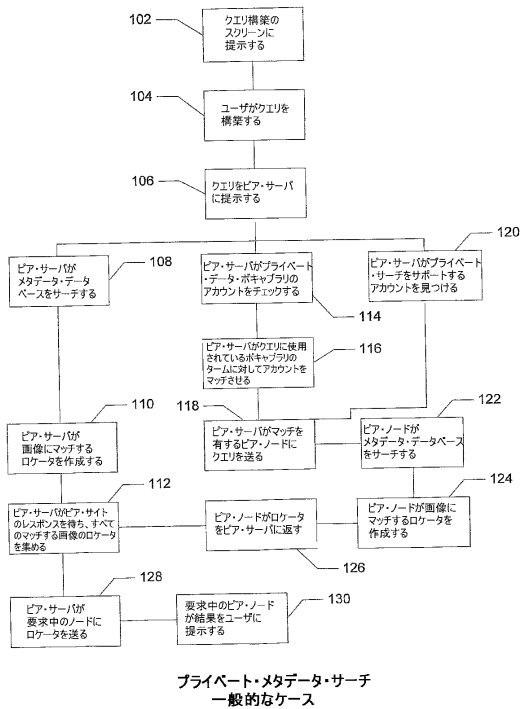
【図 1】



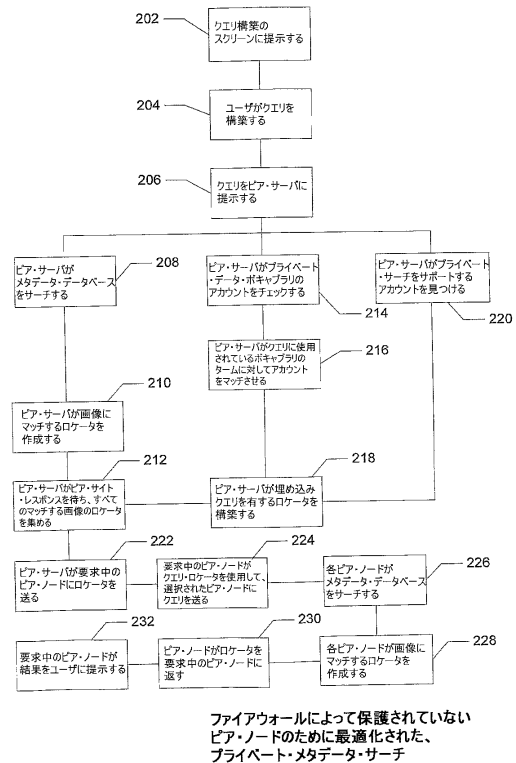
【図 2】



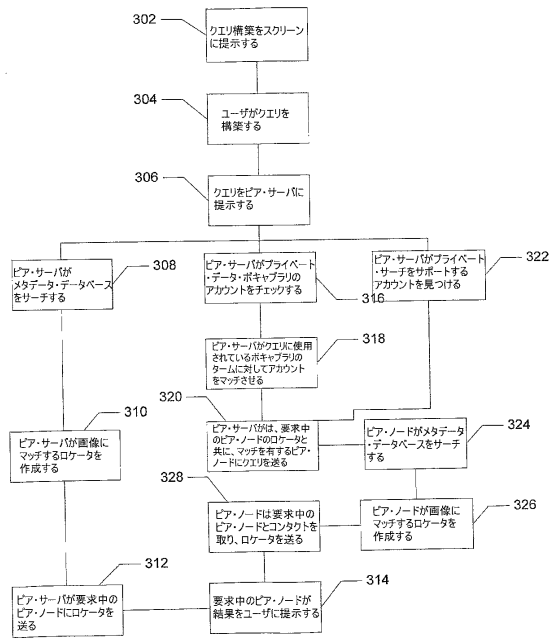
【図 3】



【図 4】



【図 5】



ファイアウォールによって保護された  
ピア・ノードのために最適化された、  
プライベート・メタデータ・サーチ

---

フロントページの続き

(74)代理人 100149641

弁理士 池上 美穂

(72)発明者 モリス、ロバート ピー .

アメリカ合衆国 2 7 6 0 6 ノースカロライナ州 ラレー フォードランド ドライブ 6 0 2  
1

(72)発明者 スペンセン、ヒュー

アメリカ合衆国 2 7 5 0 2 ノースカロライナ州 アベックス バージャロン ウェイ 1 1 0

(72)発明者 スカーディーノ、パトリシア

アメリカ合衆国 2 7 5 0 2 ノースカロライナ州 アベックス サマー オークス ドライブ  
3 2 3 3

審査官 波内 みさ

(56)参考文献 特開平 1 1 - 0 1 5 8 5 6 ( J P , A )

特開 2 0 0 1 - 0 2 8 0 0 4 ( J P , A )

特開平 1 1 - 2 1 3 0 1 4 ( J P , A )

Carl Lagoze, James R. Davis, Dienst: an architecture for distributed document libraries, Communications of the ACM , 米国, ACM , 1 9 9 5 年 4 月, Volume 38 , Issue 4 , 47

(58)調査した分野(Int.Cl. , D B 名)

G06F 17/30

G06F 13/00

G06F 15/00