(19)日本国特許庁(JP)　(12)公開特許公報(A)　(11)特許出願公開番号
特開2015-191508
(P2015-191508A)
(43)公開日　平成27年11月2日(2015.11.2)

| (51)Int.Cl. | | F I | | テーマコード（参考） |
|---|---|---|---|---|
| G06F 21/41 | (2013.01) | G06F | 21/20 | 141 |

審査請求 未請求　請求項の数 7　ＯＬ　（全 11 頁）

(21)出願番号　特願2014-69184(P2014-69184)
(22)出願日　平成26年3月28日(2014.3.28)

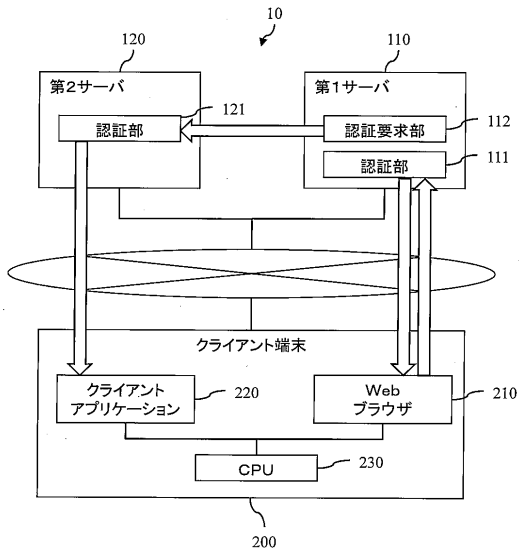
（特許庁注：以下のものは登録商標）
１．ＪＡＶＡＳＣＲＩＰＴ

(71)出願人　000233055
株式会社日立ソリューションズ
東京都品川区東品川四丁目１２番７号
(74)代理人　100091096
弁理士　平木　祐輔
(74)代理人　100102576
弁理士　渡辺　敏章
(74)代理人　100153903
弁理士　吉川　明
(72)発明者　藤本　稔
東京都品川区東品川四丁目１２番７号　株式会社日立ソリューションズ内

(54)【発明の名称】シングルサインオンシステム、シングルサインオン方法

(57)【要約】
【課題】Ｗｅｂブラウザをクライアントアプリケーションとして利用するサービスにおいて、クライアントアプリケーションがユーザ認証を受ける形態を変えることなく、シングルサインオンを実現する技術を提供する。
【解決手段】本発明に係るシングルサインオンシステムにおいて、クライアント端末はＷｅｂブラウザとＷｅｂブラウザ以外のクライアントアプリケーションを実行し、第１サーバはこれらアプリケーションのいずれか一方を認証してその結果を返信し、認証要求部は第２サーバに対してユーザ認証を要求し、第２サーバは上記アプリケーションの他方に対して認証結果を送信する。
【選択図】図１

図１



【特許請求の範囲】
【請求項１】
ユーザ認証リクエストに応じてユーザ認証を実施する第１および第２サーバ、
前記第１サーバに対してユーザ認証リクエストを送信するクライアント端末、
前記第２サーバに対してユーザ認証リクエストを送信する認証要求部、
を備え、
前記クライアント端末は、ＷｅｂブラウザおよびＷｅｂブラウザ以外のクライアントアプリケーションを実行するプロセッサを備え、
前記Ｗｅｂブラウザまたは前記クライアントアプリケーションのいずれか一方は、前記第１サーバに対して第１ユーザ認証リクエストを送信するように構成されており、
前記第１サーバは、前記Ｗｅｂブラウザまたは前記クライアントアプリケーションのいずれか一方から前記第１ユーザ認証リクエストを受け取ると、ユーザ認証を実施してその結果を送信元に対して返信し、
前記認証要求部は、前記プロセッサが前記Ｗｅｂブラウザおよび前記クライアントアプリケーションを同時に実行している間に、前記第１サーバが認証するユーザに対応するユーザを認証するよう要求する第２ユーザ認証リクエストを前記第２サーバに対して送信し、
前記第２サーバは、前記認証要求部から前記第２ユーザ認証リクエストを受け取ると、ユーザ認証を実施してその結果を前記Ｗｅｂブラウザまたは前記クライアントアプリケーションの他方に対して送信する
ことを特徴とするシングルサインオンシステム。
【請求項２】
前記第１サーバは、前記認証要求部およびユーザ認証を実施する認証部を備えており、
前記認証要求部は、前記認証部が前記クライアント端末から前記第１ユーザ認証リクエストを受け取ると、前記第２サーバに対して前記第２ユーザ認証リクエストを送信する
ことを特徴とする請求項１記載のシングルサインオンシステム。
【請求項３】
前記第１サーバは、前記クライアント端末が送信する前記第１ユーザ認証リクエストを前記第２サーバに対する前記第２ユーザ認証リクエストに変換するための対応関係を記述した対応テーブルを備え、
前記認証要求部は、前記対応テーブルを参照することにより、前記認証部が前記クライアント端末から受け取った前記第１ユーザ認証リクエストを前記２サーバに対する前記第２ユーザ認証リクエストに変換する
ことを特徴とする請求項２記載のシングルサインオンシステム。
【請求項４】
前記Ｗｅｂブラウザまたは前記クライアントアプリケーションは、前記認証要求部を実装するように構成されており、
前記認証要求部を実装した前記Ｗｅｂブラウザまたは前記クライアントアプリケーションは、前記第１サーバに対して前記第１ユーザ認証リクエストを送信するのと並行して、前記第２サーバに対して前記第２ユーザ認証リクエストを送信するように構成されている
ことを特徴とする請求項１記載のシングルサインオンシステム。
【請求項５】
前記認証要求部を実装した前記Ｗｅｂブラウザまたは前記クライアントアプリケーションは、前記第１サーバと前記第２サーバに対して、同一のユーザ認証情報を並行して送信するように構成されている
ことを特徴とする請求項４記載のシングルサインオンシステム。
【請求項６】
前記第１および第２サーバは、ユーザに関する属性情報を記述したユーザ属性データをそれぞれ保持するように構成されており、
前記Ｗｅｂブラウザまたは前記クライアントアプリケーションのいずれか一方は、前記

ユーザ属性データを更新するよう要求する第１更新リクエストを前記第１サーバに対して送信するように構成されており、
前記第１サーバは、前記Ｗｅｂブラウザまたは前記クライアントアプリケーションのいずれか一方から受け取った前記第１更新リクエストを前記ユーザ属性データに反映した上でその結果を送信元に対して返信し、
前記認証要求部は、前記クライアント端末が前記第１サーバに対して送信する前記第１更新リクエストと同一の更新内容を要求する第２更新リクエストを前記第２サーバに対して送信し、
前記第２サーバは、前記認証要求部から受け取った前記第２更新リクエストが要求する更新内容を前記ユーザ属性データに反映した上でその結果を前記Ｗｅｂブラウザまたは前記クライアントアプリケーションの他方に対して送信する
ことを特徴とする請求項１記載のシングルサインオンシステム。
【請求項７】
ユーザ認証リクエストに応じてユーザ認証を実施する第１および第２サーバ、
前記第１サーバに対してユーザ認証リクエストを送信するクライアント端末、
前記第２サーバに対してユーザ認証リクエストを送信する認証要求部、
を備え、
前記クライアント端末は、Ｗｅｂブラウザおよびウェブブラウザ以外のクライアントアプリケーションを実行するプロセッサを備え、
前記Ｗｅｂブラウザまたは前記クライアントアプリケーションのいずれか一方は、前記第１サーバに対してユーザ認証リクエストを送信するように構成されている、
情報システムにおいて、シングルサインオンを実行する方法であって、
前記プロセッサが前記Ｗｅｂブラウザまたは前記クライアントアプリケーションのいずれか一方を実施することにより、前記第１サーバに対して第１ユーザ認証リクエストを送信するステップ、
前記第１サーバが、前記Ｗｅｂブラウザまたは前記クライアントアプリケーションのいずれか一方から前記第１ユーザ認証リクエストを受け取り、ユーザ認証を実施してその結果を送信元に対して返信するステップ、
前記認証要求部が、前記プロセッサが前記Ｗｅｂブラウザおよび前記クライアントアプリケーションを同時に実行している間に、前記第１サーバが認証するユーザに対応するユーザを認証するよう要求する第２ユーザ認証リクエストを前記第２サーバに対して送信するステップ、
前記第２サーバが、前記認証要求部から前記第２ユーザ認証リクエストを受け取り、ユーザ認証を実施してその結果を前記Ｗｅｂブラウザまたは前記クライアントアプリケーションの他方に対して送信するステップ、
を有することを特徴とするシングルサインオン方法。
【発明の詳細な説明】
【技術分野】
【０００１】
本発明は、シングルサインオン技術に関するものである。
【背景技術】
【０００２】
ユーザ端末上で、複数の関連するサービスを使用する場合、ユーザの操作負担を軽減するため、いずれかのサービス上でユーザ認証を受ける（サインオンまたはログイン）と他のサービス上においてもその結果が反映されることが望ましい。このような機能はシングルサインオンと呼ばれる。シングルサインオンは例えば、各サービスが提供するクライアントアプリケーション同士が連携し、ユーザがいずれかのクライアントアプリケーションに対して入力した認証情報を他のアプリケーションに受け渡し、１回のログインで各サービスに対して同時にログインすることにより、実現できる。この手法は、クライアントアプリケーションが端末上で通信して互いに認証情報を受け渡すことを前提としている。

【０００３】
一方、近年サービスを利用するソフトウェアはＷｅｂブラウザ上で動作することが多くなっている。Ｗｅｂブラウザに入力した認証情報を、別のサービスのクライアントに渡すことは、通常セキュリティ上の理由で禁止されている。そのため上述の手法は、クライアントアプリケーションとしてＷｅｂブラウザを用いる場合においては、シングルサインオンを実現することが困難である。
【０００４】
下記特許文献１に開示されているシステムは、ユーザ端末上で常に動作するプロセスを生成し、このプロセスを経由して認証を実施することにより、シングルサインオンを実現している。
【先行技術文献】
【特許文献】
【０００５】
【特許文献１】特開２０１３－２２２４４０号公報
【発明の概要】
【発明が解決しようとする課題】
【０００６】
上記特許文献１記載の技術によってシングルサインオンを実現するためには、常駐プロセスがＷｅｂブラウザからの通信を受け付ける権限を有している必要がある。そのため、携帯端末などプロセスの権限が制限されている場合においては、同文献記載の技術を用いることは困難であると考えられる。また同文献記載の技術においては、常駐プロセスがユーザ端末上で動作するため、常駐プロセスはサービスを提供するサーバと同等のセキュリティを備えていることが必要となる。例えばＷｅｂブラウザと常駐アプリケーションとの間で独自に認証を実施することが必要となる。
【０００７】
本発明は、上記のような課題に鑑みてなされたものであり、Ｗｅｂブラウザをクライアントアプリケーションとして利用するサービスにおいて、クライアントアプリケーションがユーザ認証を受ける形態を変えることなく、シングルサインオンを実現する技術を提供することを目的とする。
【課題を解決するための手段】
【０００８】
本発明に係るシングルサインオンシステムにおいて、クライアント端末はＷｅｂブラウザとＷｅｂブラウザ以外のクライアントアプリケーションを実行し、第１サーバはこれらアプリケーションのいずれか一方を認証してその結果を返信し、認証要求部は第２サーバに対してユーザ認証を要求し、第２サーバは上記アプリケーションの他方に対して認証結果を送信する。
【発明の効果】
【０００９】
本発明に係るシングルサインオンシステムによれば、Ｗｅｂブラウザまたは他のクライアントアプリケーションが一方のサーバに対してログインすると、他方のアプリケーションも他方のサーバに対してログインした状態になる。これにより、クライアント端末上で認証処理のための新たなプログラムを動作させることなく、ＷｅｂブラウザとＷｅｂブラウザ以外のクライアントアプリケーションを用いる場合においても、シングルサインオンを実現することができる。
【図面の簡単な説明】
【００１０】
【図１】実施形態１に係るシングルサインオンシステム１０の構成図である。
【図２】対応テーブル１１３の構成とデータ例を示す図である。
【図３】第１サーバ１１０の動作を説明するフローチャートである。
【図４】第２サーバ１２０の動作を説明するフローチャートである。

【図５】実施形態２に係るシングルサインオンシステム１０の構成図である。
【図６】実施形態３に係るシングルサインオンシステム１０の構成図である。
【発明を実施するための形態】
【００１１】
＜実施の形態１＞
図１は、本発明の実施形態１に係るシングルサインオンシステム１０の構成図である。シングルサインオンシステム１０は、クライアント端末２００を使用するユーザに対してシングルサインオン機能を提供するシステムであり、第１サーバ１１０、第２サーバ１２０、クライアント端末２００を有する。これら装置はネットワークを介して互いに接続されている。
【００１２】
クライアント端末２００は、Ｗｅｂブラウザ２１０とクライアントアプリケーション２２０を実行するＣＰＵ（Ｃｅｎｔｒａｌ　Ｐｒｏｃｅｓｓｉｎｇ　Ｕｎｉｔ）２３０を備えるコンピュータである。クライアント端末２００のユーザは、Ｗｅｂブラウザ２１０を用いて第１サーバ１１０が提供するサービスを利用し、クライアントアプリケーション２２０を用いて第２サーバ１２０が提供するサービスを利用する。クライアントアプリケーション２２０はＷｅｂブラウザ以外のアプリケーションであり、セキュリティの都合上、Ｗｅｂブラウザ２１０との間でユーザ認証情報を直接的に授受することはできないように構成されている。
【００１３】
以下では記載の便宜上、Ｗｅｂブラウザ２１０またはクライアントアプリケーション２２０を動作主体として説明する場合があるが、実際にこれらアプリケーションを実行するのはＣＰＵ２３０であることを付言しておく。
【００１４】
第１サーバ１１０および第２サーバ１２０がそれぞれ提供するサービスを利用するためには、あらかじめこれらサーバにログインする必要がある。シングルサインオンシステム１０は、ユーザの利便性を高めるため、ユーザがＷｅｂブラウザ２１０を用いて第１サーバ１１０に対してログインすると、クライアントアプリケーション２２０も第２サーバ１２０に対して自動的にログインした状態となる機能を提供する。
【００１５】
第１サーバ１１０は、Ｗｅｂブラウザ２１０が利用するサービスを提供するサーバコンピュータであり、認証部１１１、認証要求部１１２、および後述する対応テーブル１１３を備える。認証部１１１は、Ｗｅｂブラウザ２１０からユーザ認証情報とともにユーザ認証リクエストを受信し、ユーザ認証を実施してその結果をＷｅｂブラウザ２１０へ返信する。認証要求部１１２は、第２サーバ１２０に対してユーザ認証情報とともにユーザ認証リクエストを送信する。
【００１６】
第２サーバ１２０は、クライアントアプリケーション２２０が利用するサービスを提供するサーバコンピュータであり、認証部１２１を備える。認証部１２１は、認証要求部１１２からユーザ認証情報とともにユーザ認証リクエストを受信し、ユーザ認証を実施してその結果をクライアントアプリケーション２２０へ送信する。
【００１７】
クライアントアプリケーション２２０は、定期的に第２サーバ１２０に対してポーリングするか、またはあらかじめクライアントアプリケーション２２０と第２サーバ１２０との間でコネクションを確立しておき第２サーバ１２０からクライアントアプリケーション２２０に対してＰＵＳＨ通信する。これにより、認証部１２１からクライアントアプリケーション２２０に対してユーザ認証結果を通知することができる。
【００１８】
図２は、対応テーブル１１３の構成とデータ例を示す図である。対応テーブル１１３は第１サーバ１１０上のユーザ認証情報と第２サーバ１２０上のユーザ認証情報との間の対

応関係を記述したデータテーブルであり、第１サーバ１１０が備えるハードディスク装置などの記憶装置に格納されている。

【００１９】

認証部１１１は、Ｗｅｂブラウザ２１０からユーザ認証リクエストを受け取ると、対応テーブル１１３を参照して、そのリクエストのユーザ認証情報が第１サーバ１１０上におけるユーザ認証情報と合致するか否かを判定することにより、ユーザ認証を実施する。認証要求部１１２は、認証部１１１がＷｅｂブラウザ２１０からのユーザ認証リクエストを許可した場合は、対応テーブル１１３を参照することによりそのユーザ認証情報を第２サーバ１２０上におけるユーザ認証情報に変換し、これを用いて第２サーバ１２０に対してユーザ認証リクエストを発行する。

【００２０】

図２においては、ユーザ認証情報の例として、第１サーバ１１０上におけるユーザＩＤとパスワードの組（第１サーバユーザＩＤ１１３１と第１サーバパスワード１１３２）を第２サーバ１２０上におけるユーザＩＤとパスワードの組（第２サーバユーザＩＤ１１３３と第２サーバパスワード１１３４）に変換する例を示したが、ユーザ認証情報の対応関係を記述することができれば、必ずしもＩＤとパスワードの組を用いる必要はない。

【００２１】

図３は、第１サーバ１１０の動作を説明するフローチャートである。以下図３の各ステップについて説明する。

【００２２】

（図３：ステップＳ３００）

クライアント端末２００のユーザは、Ｗｅｂブラウザ２１０を用いて第１サーバ１１０が提供するＷｅｂサイトにアクセスする。ユーザは、Ｗｅｂブラウザ２１０上でユーザ認証情報（例えばユーザＩＤとパスワード）を入力する。Ｗｅｂブラウザ２１０は、第１サーバ１１０に対して、そのユーザ認証情報とともにユーザ認証リクエストを送信する。認証部１１１がそのリクエストを受信すると本フローチャートが開始する。

【００２３】

（図３：ステップＳ３０１～Ｓ３０２）

認証部１１１は、Ｗｅｂブラウザ２１０が送信したユーザ認証情報を取得し、これを対応テーブル１１３が格納しているユーザ認証情報と照合することにより、ユーザ認証を実施する（Ｓ３０１）。ユーザ認証に成功した場合はステップＳ３０４へ進み、成功しなかった場合はステップＳ３０３へ進む（Ｓ３０２）。

【００２４】

（図３：ステップＳ３０３）

認証部１１１は、ユーザ認証に成功しなかった旨の応答を、Ｗｅｂブラウザ２１０に対して返信し、本フローチャートは終了する。Ｗｅｂブラウザ２１０はその応答を受け取ると、ユーザ認証できなかった旨を通知する画面を表示する。

【００２５】

（図３：ステップＳ３０４）

認証要求部１１２は、対応テーブル１１３を参照して、Ｗｅｂブラウザ２１０が送信した第１サーバ１１０上のユーザ認証情報に対応する第２サーバ１２０上のユーザ認証情報を取得する。認証要求部１１２は、取得した第２サーバ１２０上のユーザ認証情報を用いて、第２サーバ１２０に対してユーザ認証リクエストを送信する。

【００２６】

（図３：ステップＳ３０５）

認証部１２１は、認証要求部１１２が送信したユーザ認証情報を用いてユーザ認証を実施し、その結果を認証要求部１１２に対して返信する。ユーザ認証の手法としては、対応テーブル１１３と同様のＩＤとパスワードの組を記述したデータを用いてもよいし、その他の手法を用いてもよい。ユーザ認証に成功した場合はステップＳ３０６へ進み、成功しなかった場合はステップＳ３０７へ進む。

【００２７】
（図３：ステップＳ３０６）
認証要求部１１２は、認証部１２１からユーザ認証に成功した旨の通知を受信する。認証部１１１は、Ｗｅｂブラウザ２１０に対して、ユーザ認証に成功した旨の通知を送信する。例えば、ログイン完了後にＷｅｂブラウザ２１０が表示すべきＷｅｂコンテンツとともに、ランダムに生成したセッションＩＤを送信する。以後認証部１１１とＷｅｂブラウザ２１０はそのセッションＩＤを共有し、これによりＷｅｂブラウザ２１０は第１サーバ１１０上にログインした状態となる。その他適当な公知手法を用いてもよい。
【００２８】
（図３：ステップＳ３０７）
認証要求部１１２は、認証部１２１からユーザ認証に失敗した旨の通知を受信する。認証部１１１は、ユーザ認証に成功しなかった旨の応答を、Ｗｅｂブラウザ２１０に対して返信し、本フローチャートは終了する。Ｗｅｂブラウザ２１０はその応答を受け取ると、ユーザ認証できなかった旨を通知する画面を表示する。
【００２９】
図４は、第２サーバ１２０の動作を説明するフローチャートである。以下図４の各ステップについて説明する。
【００３０】
（図４：ステップＳ４００～Ｓ４０２）
認証部１２１が認証要求部１１２からユーザ認証リクエストを受け取ると、本フローチャートが開始する（Ｓ４００）。認証部１２１は、認証要求部１１２が送信したユーザ認証情報を用いてユーザ認証を実施し、その結果を認証要求部１１２に対して返信する（Ｓ４０１）。ユーザ認証に成功した場合はステップＳ４０３へ進み、成功しなかった場合はステップＳ４０４へ進む（Ｓ４０２）。
【００３１】
（図４：ステップＳ４０３）
認証部１２１は、ユーザ認証に成功した旨を認証要求部１１２に対して返信する。またこれと並行して、クライアントアプリケーション２２０に対してユーザ認証に成功した旨の通知を送信する。例えば認証部１１１からＷｅｂブラウザ２１０に対する返信と同様に、ランダムに生成したセッションＩＤをクライアントアプリケーション２２０に対して送信することができる。以後認証部１２１とクライアントアプリケーション２２０はそのセッションＩＤを共有し、これによりクライアントアプリケーション２２０は第２サーバ１２０上にログインした状態となる。その他適当な公知手法を用いてもよい。
【００３２】
（図４：ステップＳ４０３：補足）
本ステップの前提として、クライアント端末２００上でクライアントアプリケーション２２０が起動している必要がある。すなわちクライアント端末２００のユーザは、Ｗｅｂブラウザ２１０を用いてユーザ認証リクエストを送信する前に、あらかじめクライアントアプリケーション２２０を起動して待機させておくことが望ましい。
【００３３】
（図４：ステップＳ４０４）
認証部１２１は、ユーザ認証に失敗した旨の通知を認証要求部１１２に対して送信する。
【００３４】
＜実施の形態１：まとめ＞
以上のように、本実施形態１に係るシングルサインオンシステム１０において、Ｗｅｂブラウザ２１０が第１サーバ１１０に対してユーザ認証リクエストを送信すると、第１サーバ１１０はこれに対して返信するとともに、第２サーバ１２０に対してユーザ認証リクエストを送信する。第２サーバ１２０は、ユーザ認証の結果をクライアントアプリケーション２２０に対して送信する。これにより、クライアント端末２００上でＷｅｂブラウザ

２１０とクライアントアプリケーション２２０を連携させることなく、シングルサインオンを実現することができる。

【００３５】

＜実施の形態２＞

図５は、本発明の実施形態２に係るシングルサインオンシステム１０の構成図である。本実施形態２において、認証要求部１１２は、第１サーバ１１０に代えてＷｅｂブラウザ２１０上に実装されている。その他の構成は概ね実施形態１と同様であるため、以下では差異点を中心に説明する。

【００３６】

ユーザがＷｅｂブラウザ２１０を用いて第１サーバ１１０に対してユーザ認証リクエストを送信すると、認証要求部１１２はその後またはこれと並行して第２サーバ１２０に対してユーザ認証リクエストを送信する。実施形態１と同様に対応テーブル１１３をあらかじめクライアント端末２００上に格納しておき、これを用いて第１サーバ１１０上のユーザ認証情報を第２サーバ１２０上のユーザ認証情報に変換してもよいし、第１サーバ１１０と第２サーバ１２０が同一のユーザ認証情報を保持しておき、これらサーバに対して同一のユーザ認証情報を送信するようにしてもよい。

【００３７】

認証要求部１１２は、例えばＷｅｂアプリケーションとして構成することができる。具体的には、ユーザがＷｅｂブラウザ２１０上に入力した第１サーバ１１０上のユーザ認証情報を取得し、これを用いて第２サーバ１２０に対してユーザ認証リクエストを送信するスクリプトを用いて、認証要求部１１２を実装することができる。例えばログインページ内に記述されたＪａｖａＳｃｒｉｐｔなどを用いて認証要求部１１２を実装することができる。その他任意の公知技術を用いてもよい。

【００３８】

＜実施の形態３＞

実施形態１～２では、クライアント端末２００のユーザはＷｅｂブラウザ２１０を用いて第１サーバ１１０に対してログインすることを説明したが、クライアントアプリケーション２２０を用いて第１サーバ１１０に対してログインするようにシングルサインオンシステム１０を構成することもできる。本発明の実施形態３ではその構成例について説明する。

【００３９】

図６は、本実施形態３に係るシングルサインオンシステム１０の構成図である。本実施形態３において、クライアント端末２００のユーザはクライアントアプリケーション２２０を用いて第１サーバ１１０上にログインする。認証部１２１は、認証要求部１１２からユーザ認証リクエストを受け取ると、その結果を認証要求部１１２に対して返信するとともに、Ｗｅｂブラウザ２１０に対して送信する。このように、Ｗｅｂブラウザ２１０とクライアントアプリケーション２２０の役割を実施形態１～２におけるこれらの役割と入れ替えても、同様のシングルサインオン機能を実現することができる。

【００４０】

クライアントアプリケーション２２０が認証要求部１１２を実装する場合、クライアントアプリケーション２２０の一部として認証要求部１１２を組み込むことができる。対応テーブル１１３についても同様である。

【００４１】

Ｗｅｂブラウザ２１０は、実施形態１～２におけるクライアントアプリケーション２２０と同様に、定期的に第２サーバ１２０に対してポーリングするか、またはあらかじめ第２サーバ１２０との間でコネクションを確立しておき第２サーバ１２０から認証結果をＰＵＳＨ通信により受信する。第２サーバ１２０からＷｅｂブラウザ２１０またはクライアントアプリケーション２２０に対するＰＵＳＨ通信は、例えばＷｅｂＳｏｃｋｅｔなどの技術を用いて実装することができる。

【００４２】

＜実施の形態４＞

実施形態１～３では、クライアント端末２００が送信するユーザ認証情報を各サーバ間で共有することにより、シングルサインオンを実現した。ユーザ認証情報が各サーバ間で共有されていることを利用すると、例えば第１サーバ１１０上のあるユーザに関する属性情報を更新し、その更新内容を第２サーバ１２０上の同一ユーザに関する属性情報として自動的に反映することが考えられる。

【００４３】

この更新に係る処理シーケンスは、実施形態１～３においてシングルサインオンを実現する処理シーケンスと同様に実装することができる。例えば実施形態１で説明した構成を前提とする場合、（ａ）Ｗｅｂブラウザ２１０は第１サーバ１１０に対して、あるユーザに関する暗号鍵を更新するリクエストを送信し、認証部１１１はそのリクエストにしたがって第１サーバ１１０が格納しているユーザ属性データを更新し、（ｂ）認証要求部１１２は、同様のリクエストを第２サーバ１２０上の対応するユーザに対して実施するよう第２サーバ１２０に対して要求し、（ｃ）認証部１２１は、そのリクエストにしたがって第２サーバ１２０が格納しているユーザ属性データを更新するとともに、その結果をクライアントアプリケーション２２０に対して通知する、という処理シーケンスが考えられる。その他の属性情報についても同様のシーケンスにより処理することができる。

【００４４】

本発明は上記した実施形態に限定されるものではなく、様々な変形例が含まれる。上記実施形態は本発明を分かりやすく説明するために詳細に説明したものであり、必ずしも説明した全ての構成を備えるものに限定されるものではない。また、ある実施形態の構成の一部を他の実施形態の構成に置き換えることもできる。また、ある実施形態の構成に他の実施形態の構成を加えることもできる。また、各実施形態の構成の一部について、他の構成を追加・削除・置換することもできる。

【００４５】

上記各構成、機能、処理部、処理手段等は、それらの一部や全部を、例えば集積回路で設計する等によりハードウェアで実現してもよい。また、上記の各構成、機能等は、プロセッサがそれぞれの機能を実現するプログラムを解釈し、実行することによりソフトウェアで実現してもよい。各機能を実現するプログラム、テーブル、ファイル等の情報は、メモリ、ハードディスク、ＳＳＤ（Ｓｏｌｉｄ　Ｓｔａｔｅ　Ｄｒｉｖｅ）等の記録装置、ＩＣカード、ＳＤカード、ＤＶＤ等の記録媒体に格納することができる。

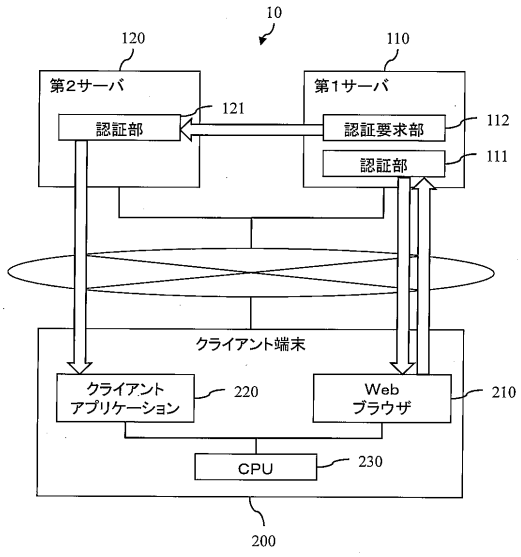
【符号の説明】

【００４６】

１０：シングルサインオンシステム、１１０：第１サーバ、１１１：認証部、１１２：認証要求部、１２０：第２サーバ、１２１：認証部、２００：クライアント端末、２１０：Ｗｅｂブラウザ、２２０：クライアントアプリケーション、２３０：ＣＰＵ。

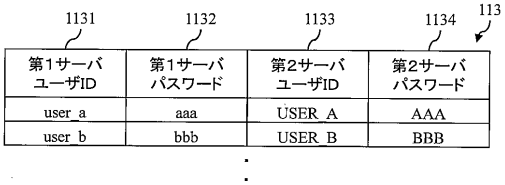
【図１】

図１



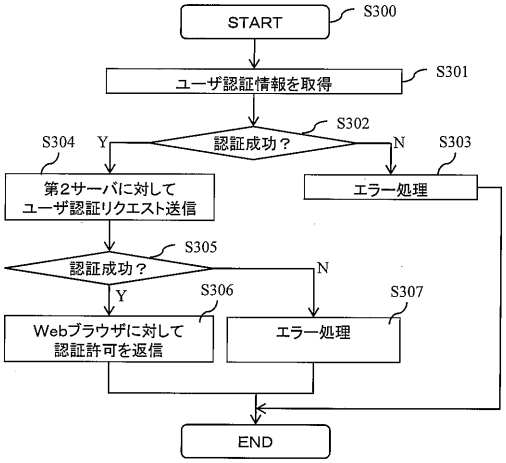

【図２】

図２



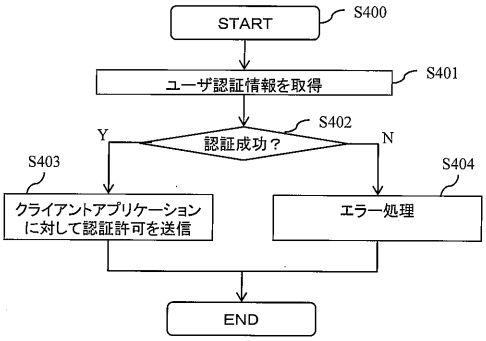

【図３】

図３



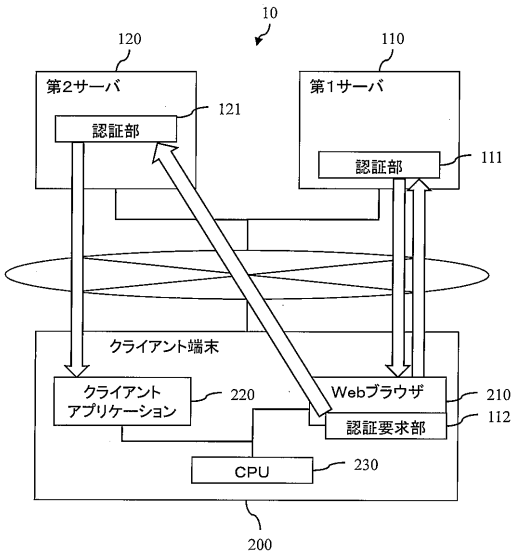

【図４】

図４



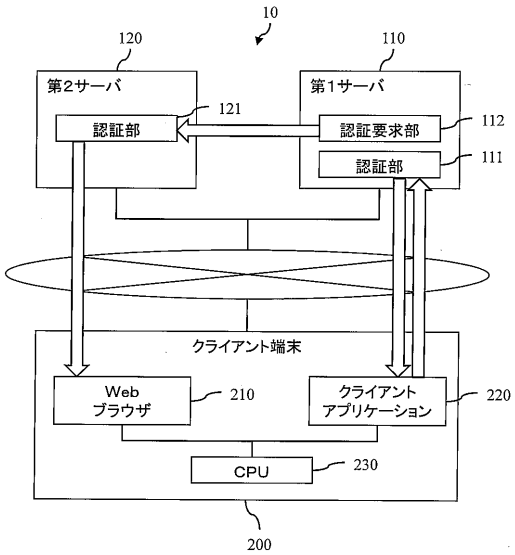

【図５】

図５



【図６】

図６


10
120
110
第2サーバ
第1サーバ
121
認証部
認証要求部
112
認証部
111
クライアント端末
Web
ブラウザ
210
クライアント
アプリケーション
220
CPU
230
200