研究記録管理・公開・検証システム arχves の web 認証基盤への対応

小林聡研究室
S063078　藤坂達也

arχves とは

- 投稿された研究記録の機密性・非改竄性を維持し、研究記録としての価値を高めるためのシステム

arχves の特徴

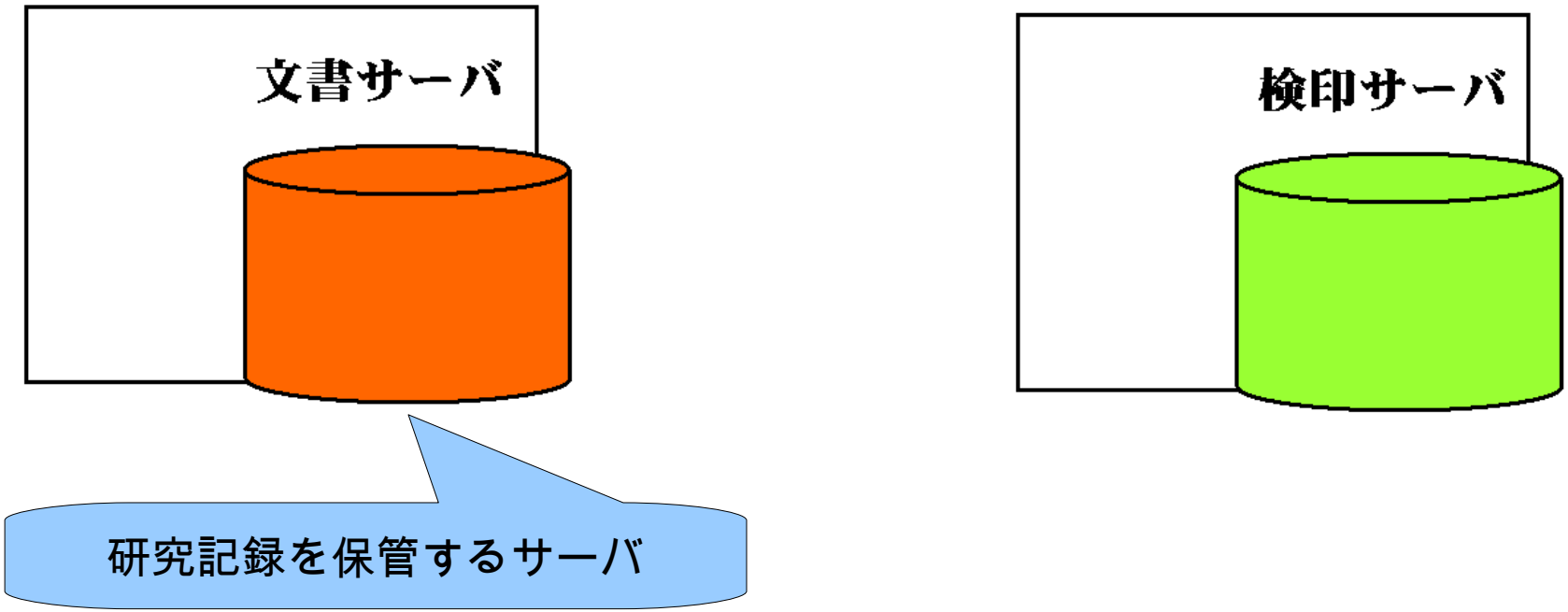
- 電子署名を用いた研究記録への四重署名
- 確認者による研究記録への確認機能
- 研究記録の公開範囲の細かな設定が可能
- 数式表現に対応
- Ruby on Rails による実装

研究記録への四重署名

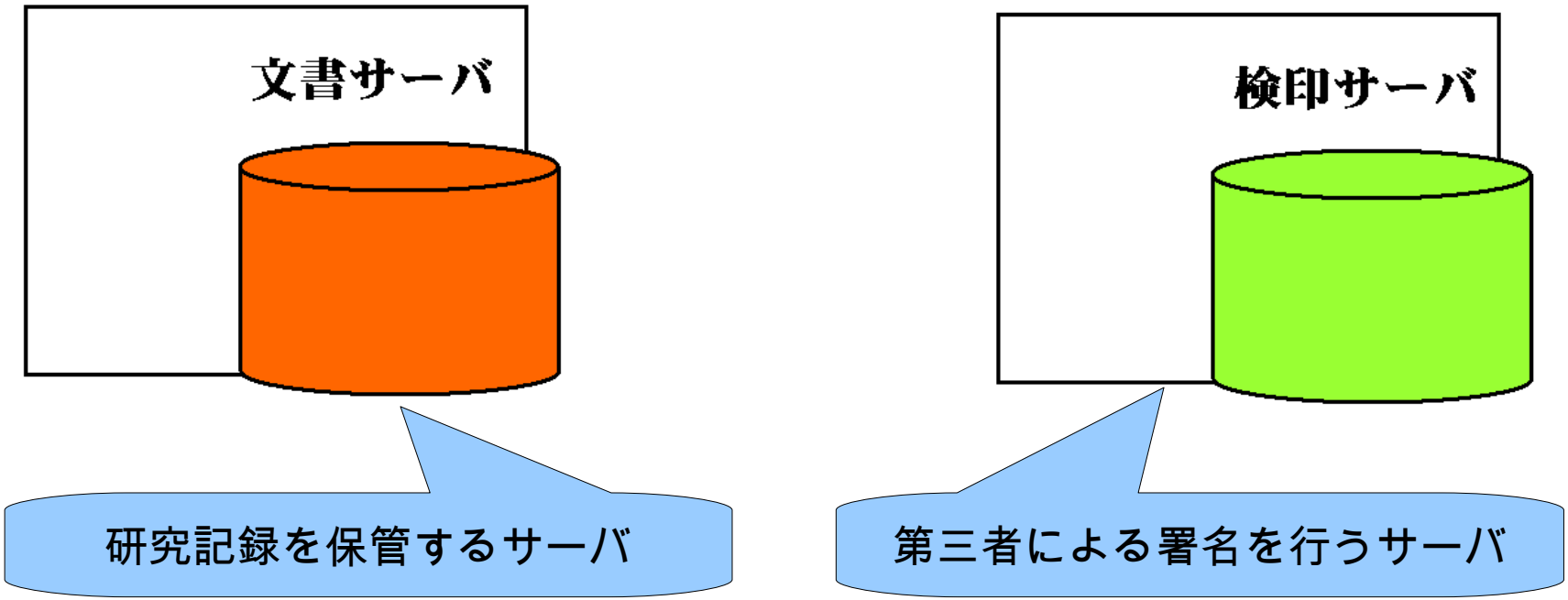
- 文書作成者による署名（ユーザ署名）
  - 作成者が本人であることを確実にする
- 第三者による署名（サーバ署名）
  - 研究記録のある時点での存在を証明
- 確認者による署名（確認者署名）
  - 確認者が確認者自身であることを確実にする
- 第三者による署名（サーバ署名）
  - 研究記録をある時点で確認者が内容確認した証明

四重署名処理の流れ

四重署名処理の流れ

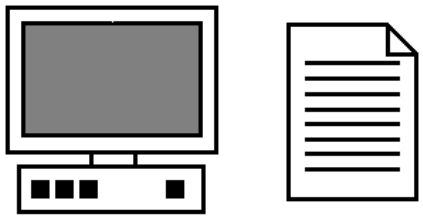


四重署名処理の流れ

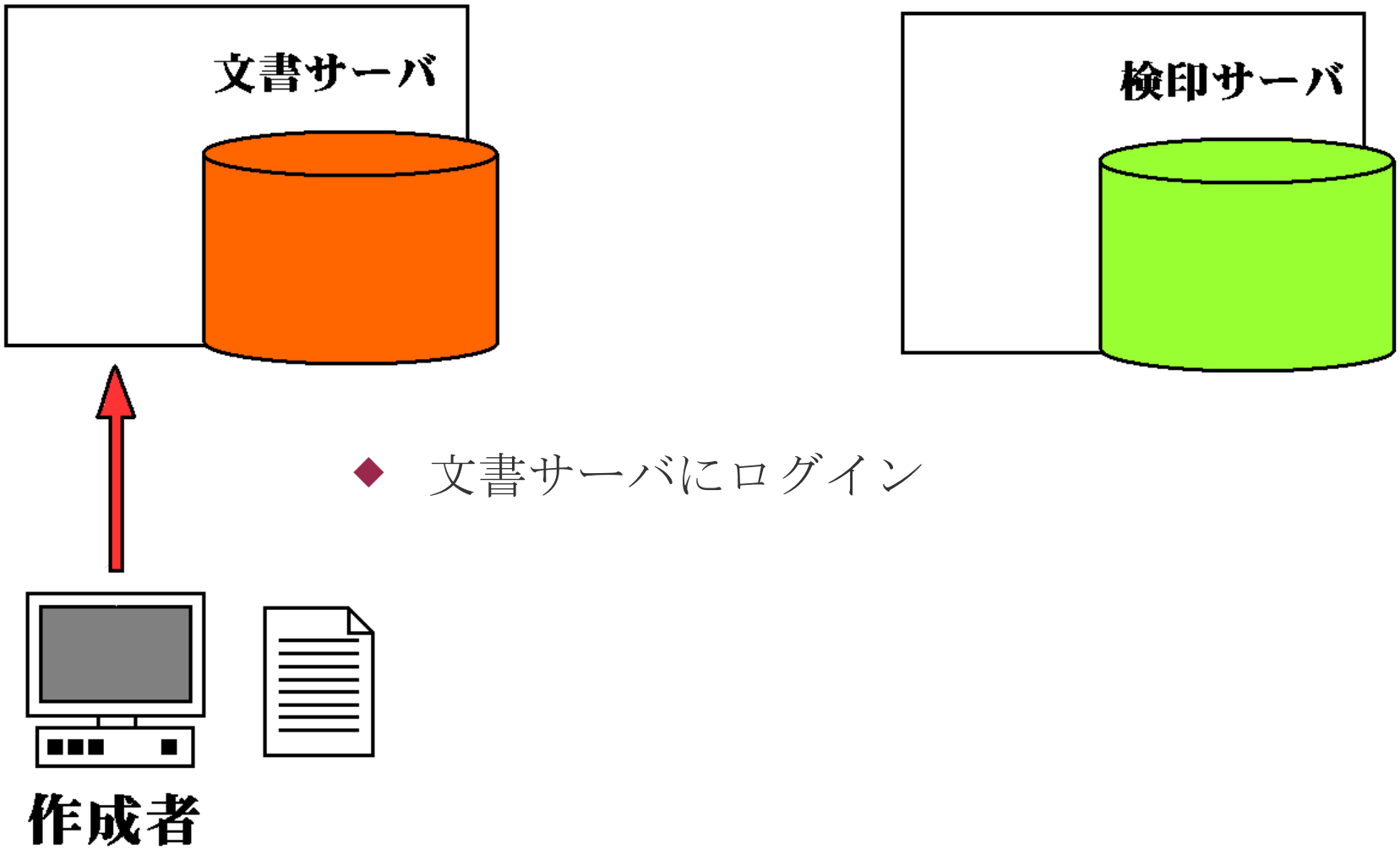


四重署名処理の流れ

作成者による二重署名

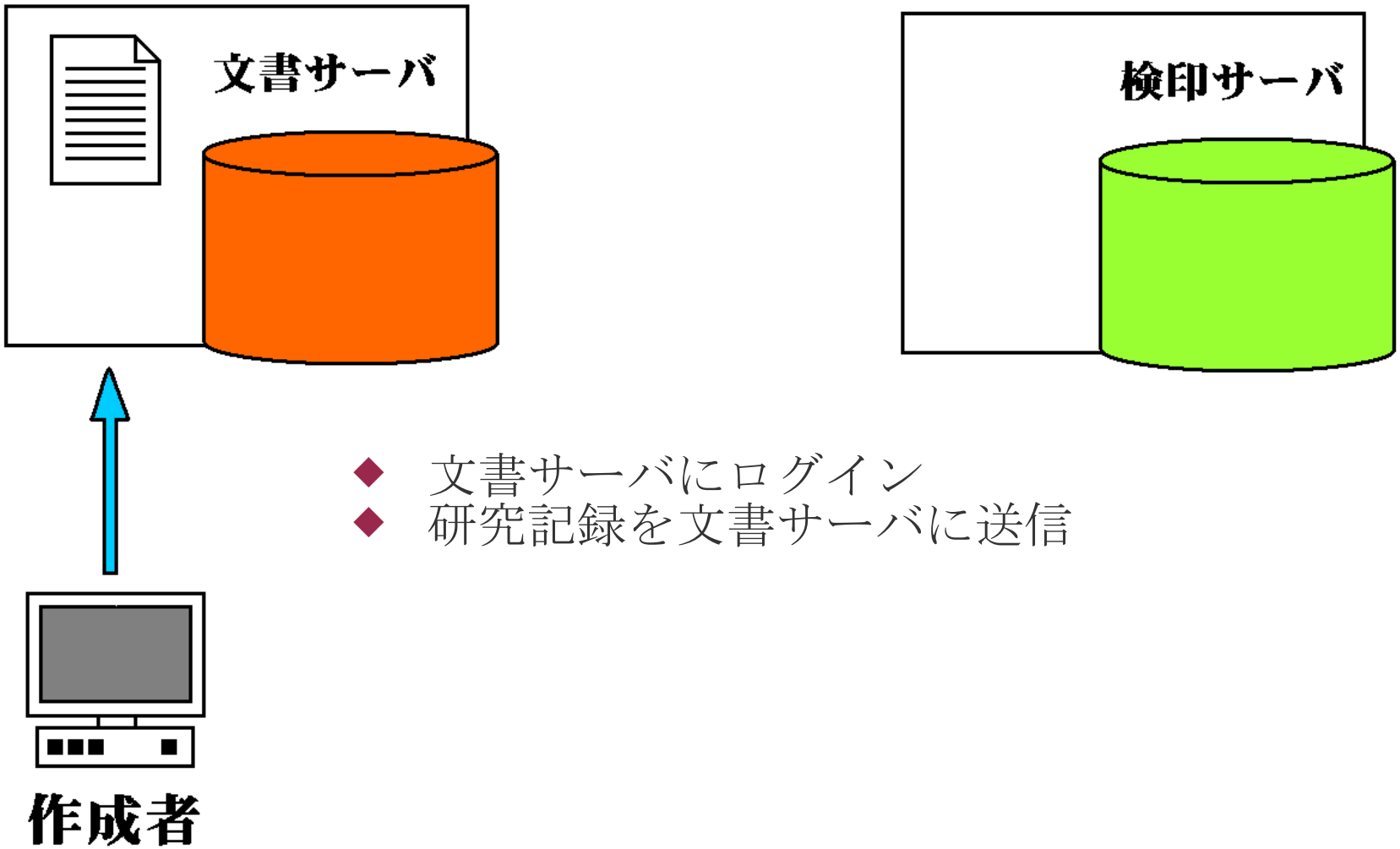


四重署名処理の流れ



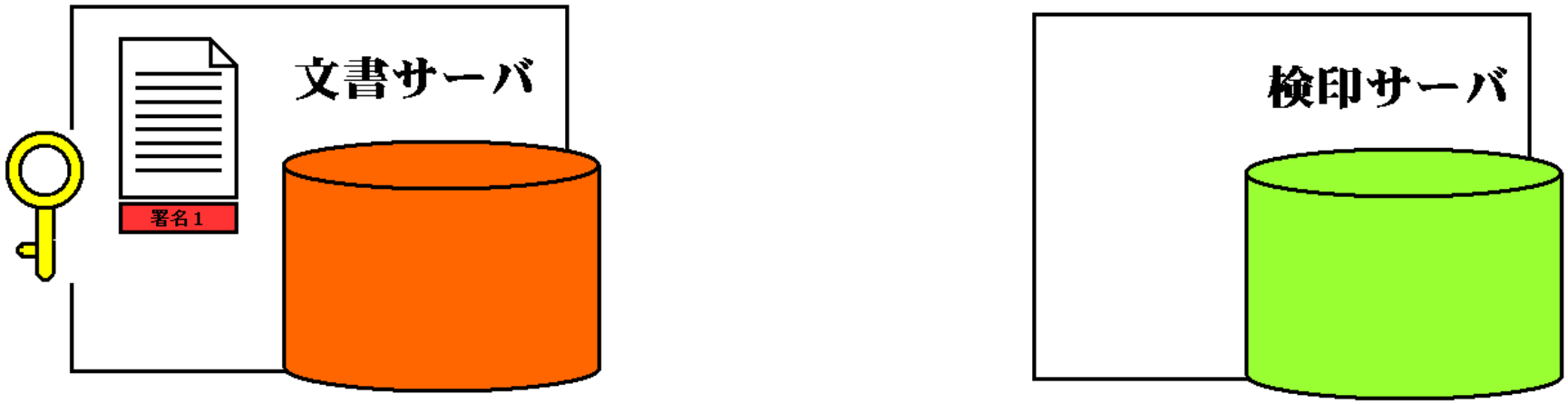
- 文書サーバにログイン

四重署名処理の流れ



- 文書サーバにログイン
- 研究記録を文書サーバに送信

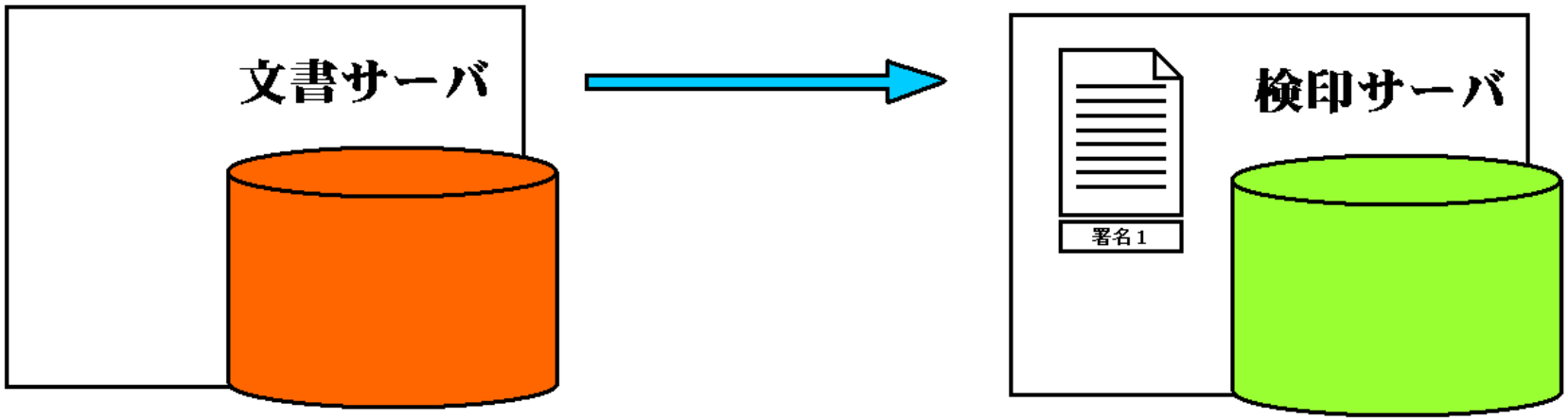
四重署名処理の流れ



- 文書サーバにログイン
- 研究記録を文書サーバに送信
- 作成者の秘密鍵で研究記録に署名

作成者

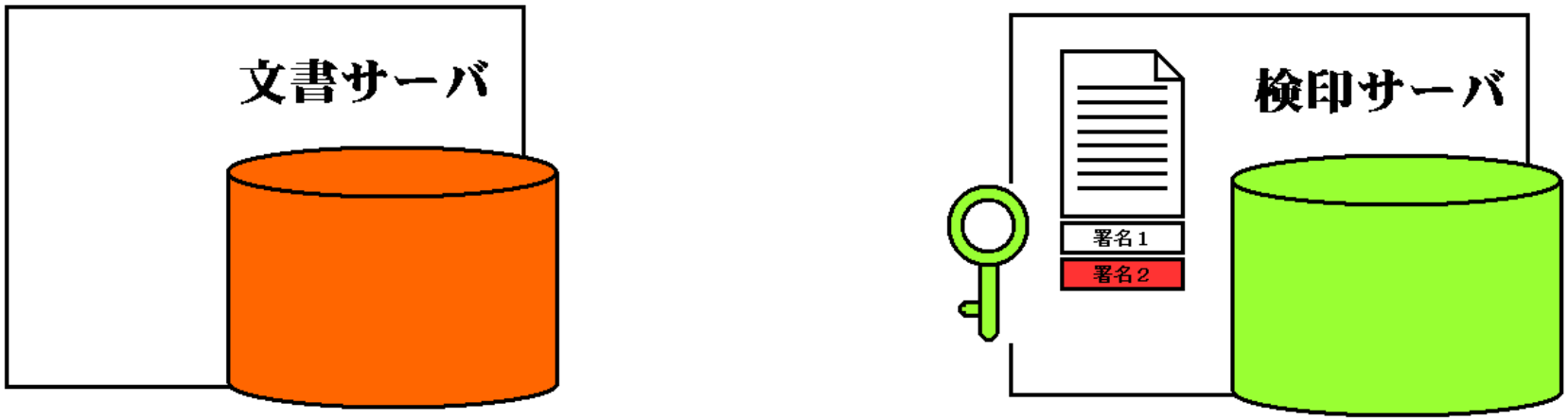
四重署名処理の流れ



- 文書サーバにログイン
- 研究記録を文書サーバに送信
- 作成者の秘密鍵で研究記録に署名
- 研究記録＋署名を検印サーバに送信

作成者

四重署名処理の流れ



- 文書サーバにログイン
- 研究記録を文書サーバに送信
- 作成者の秘密鍵で研究記録に署名
- 研究記録＋署名を検印サーバに送信
- 検印サーバの秘密鍵で研究記録＋署名１に署名

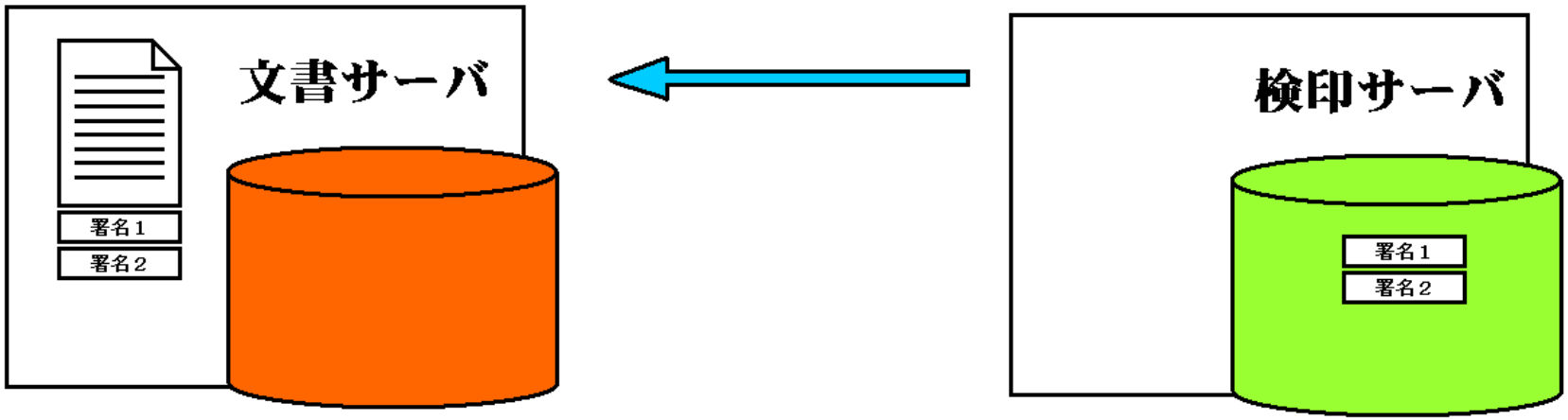
作成者

四重署名処理の流れ

- 文書サーバにログイン
- 研究記録を文書サーバに送信
- 作成者の秘密鍵で研究記録に署名
- 研究記録＋署名を検印サーバに送信
- 検印サーバの秘密鍵で研究記録＋署名１に署名
- 署名部をデータベースに保存

作成者

四重署名処理の流れ



- 文書サーバにログイン
- 研究記録を文書サーバに送信
- 作成者の秘密鍵で研究記録に署名
- 研究記録＋署名を検印サーバに送信
- 検印サーバの秘密鍵で研究記録＋署名１に署名
- 署名部をデータベースに保存
- 研究記録＋署名部を文書サーバに送信

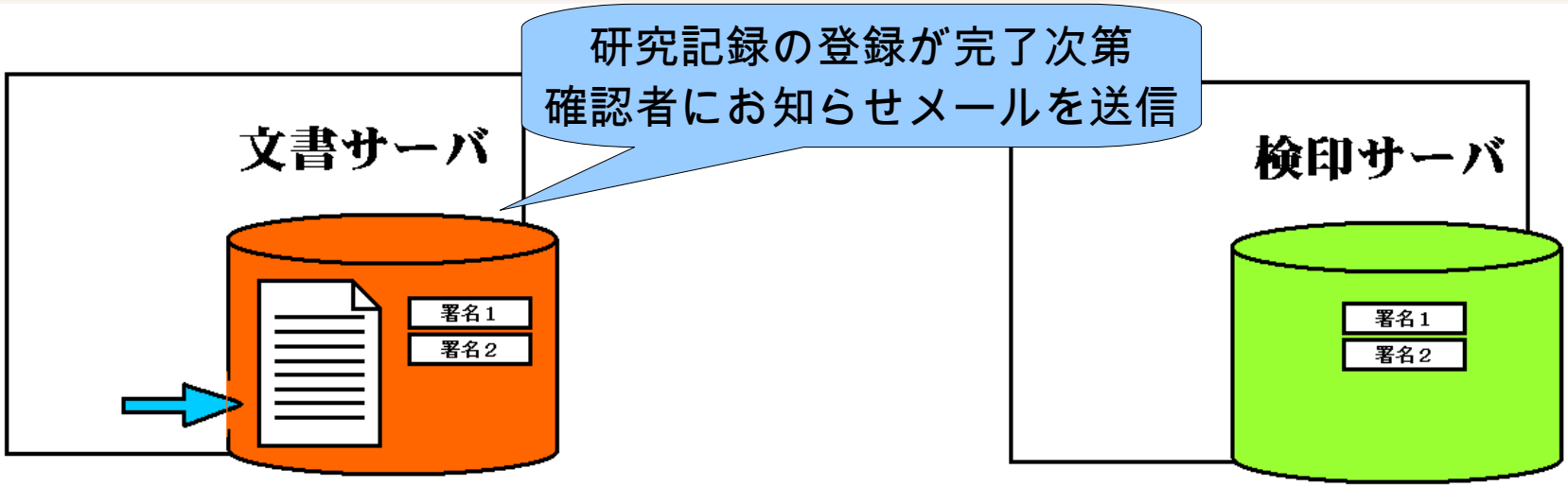
作成者

四重署名処理の流れ

作成者

- 文書サーバにログイン
- 研究記録を文書サーバに送信
- 作成者の秘密鍵で研究記録に署名
- 研究記録＋署名を検印サーバに送信
- 検印サーバの秘密鍵で研究記録＋署名１に署名
- 署名部をデータベースに保存
- 研究記録＋署名部を文書サーバに送信
- 研究記録＋署名部をデータベースに保存

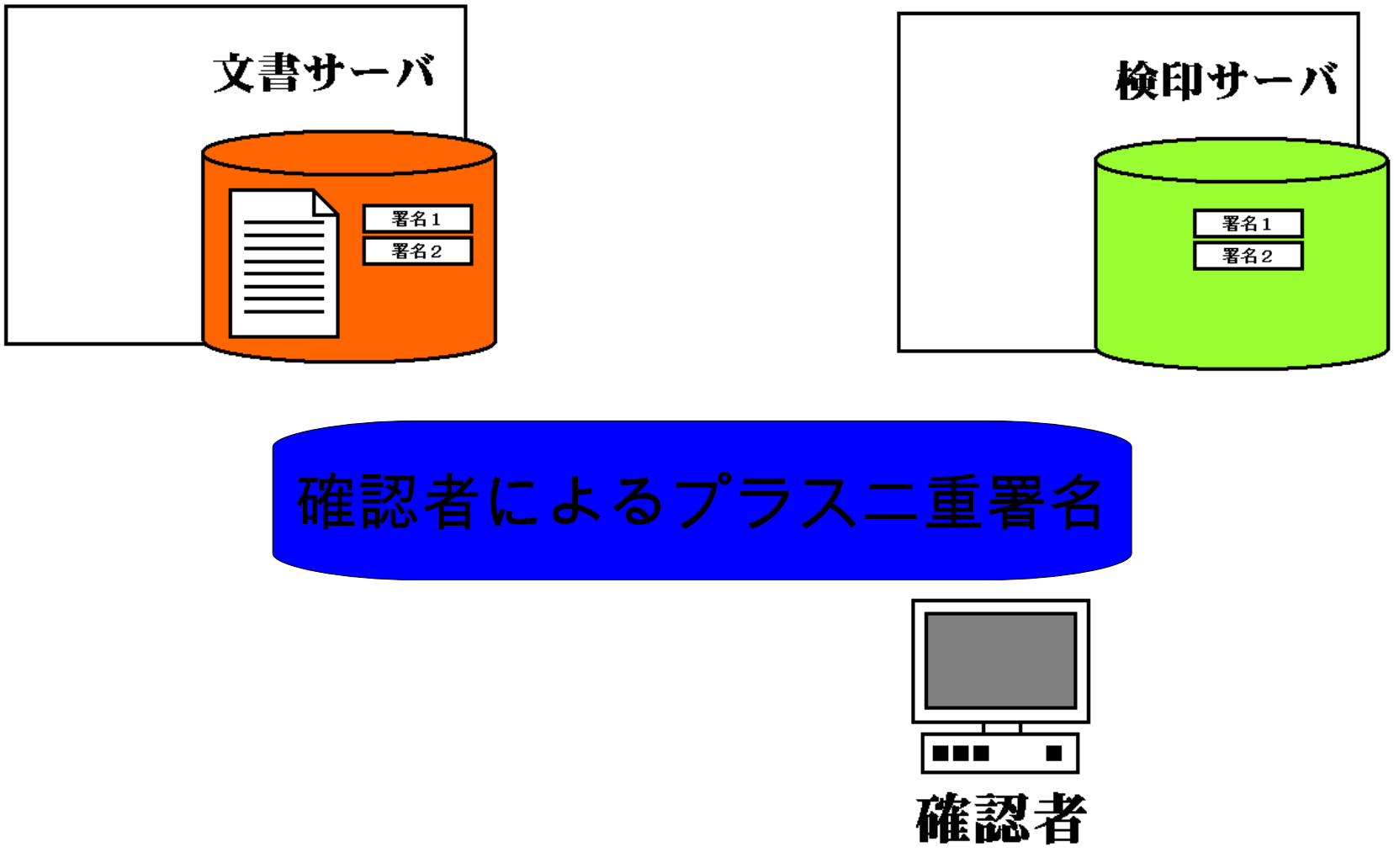
四重署名処理の流れ



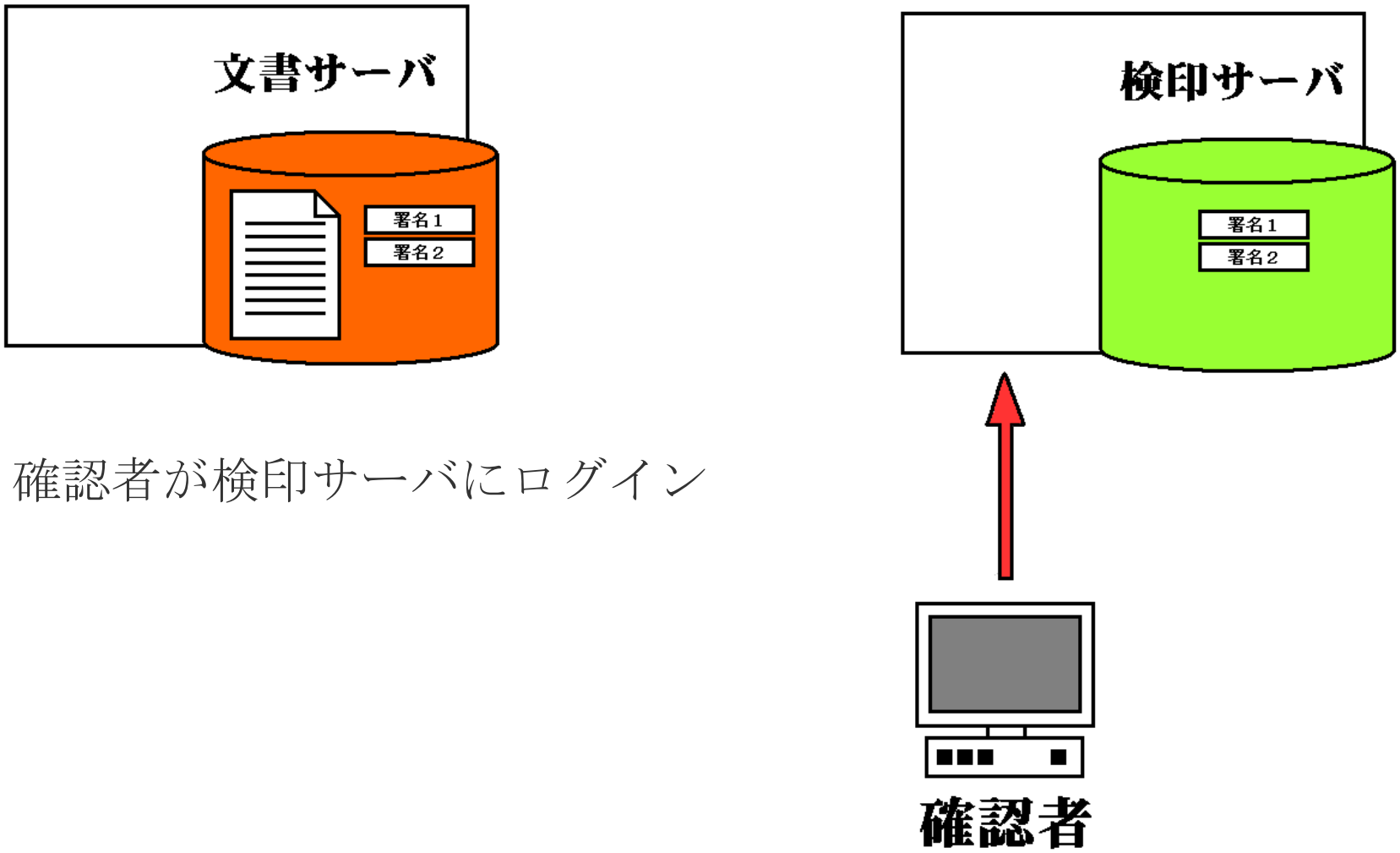
作成者

- 文書サーバにログイン
- 研究記録を文書サーバに送信
- 作成者の秘密鍵で研究記録に署名
- 研究記録＋署名を検印サーバに送信
- 検印サーバの秘密鍵で研究記録＋署名１に署名
- 署名部をデータベースに保存
- 研究記録＋署名部を文書サーバに送信
- 研究記録＋署名部をデータベースに保存

四重署名処理の流れ

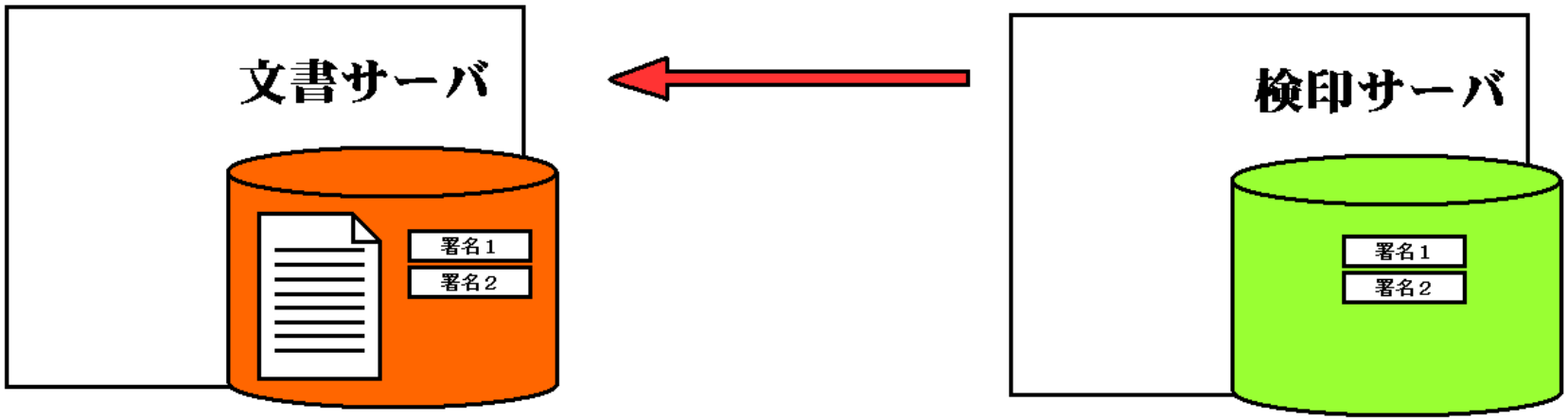


四重署名処理の流れ



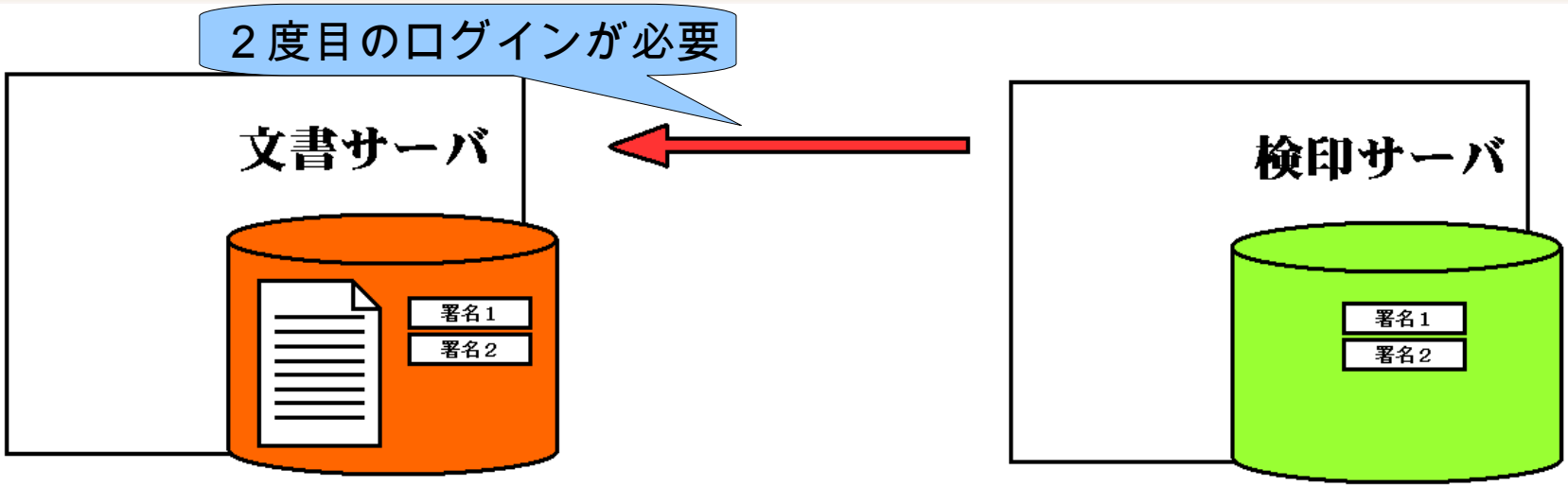
- 確認者が検印サーバにログイン

四重署名処理の流れ



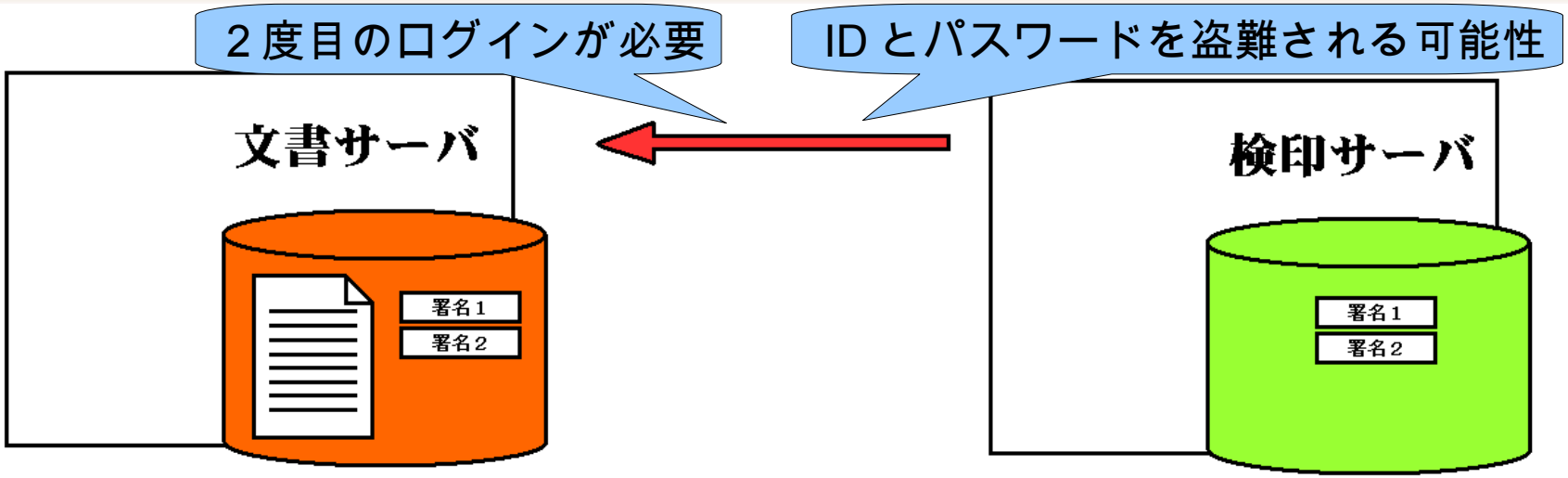
- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン

四重署名処理の流れ



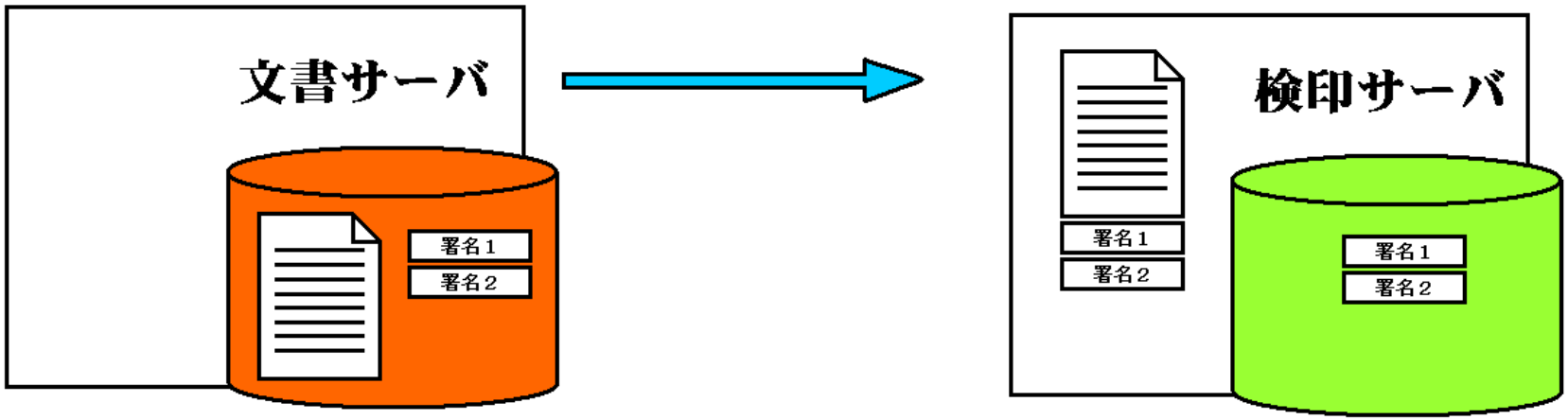
- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン

四重署名処理の流れ



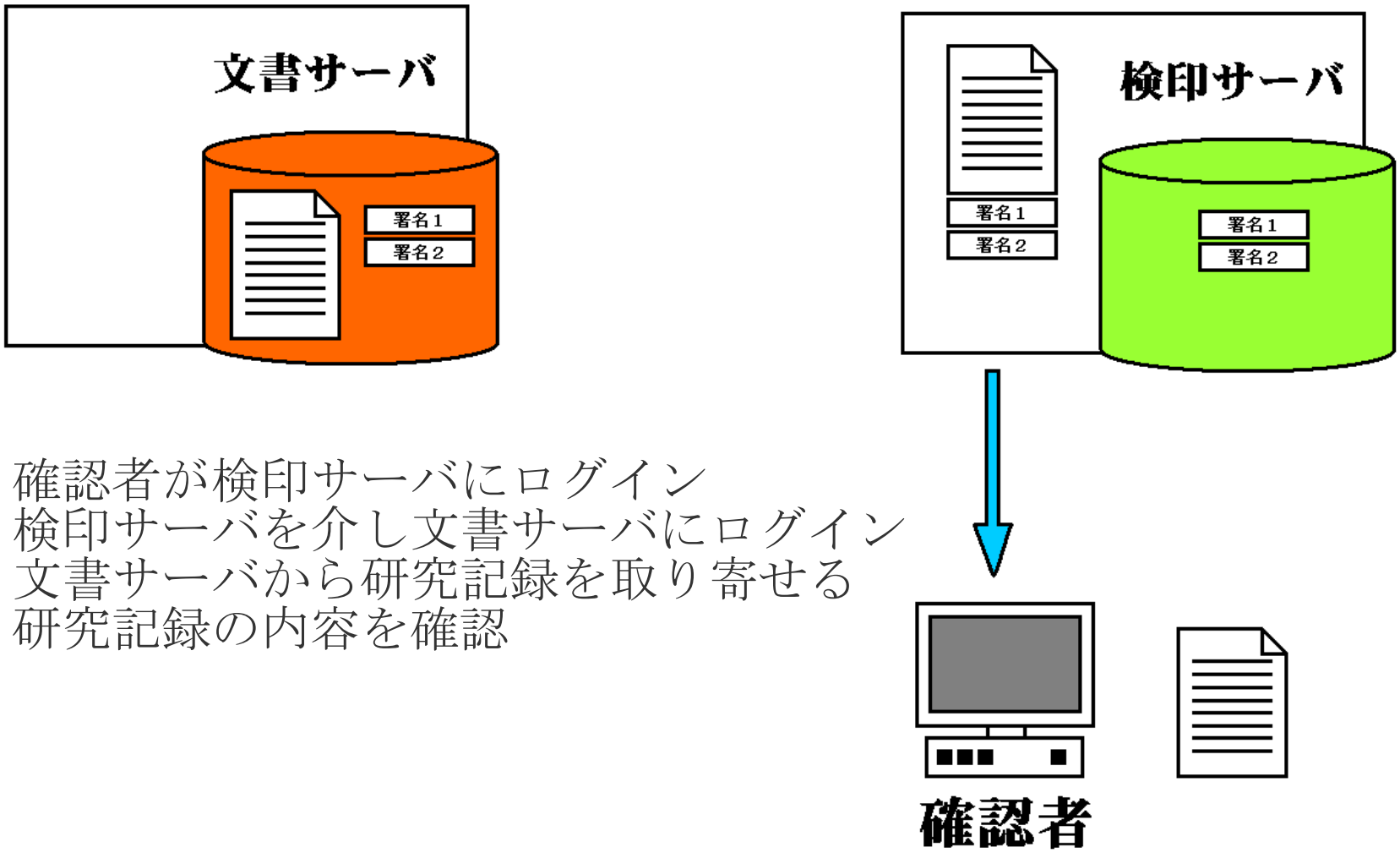
- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン

四重署名処理の流れ



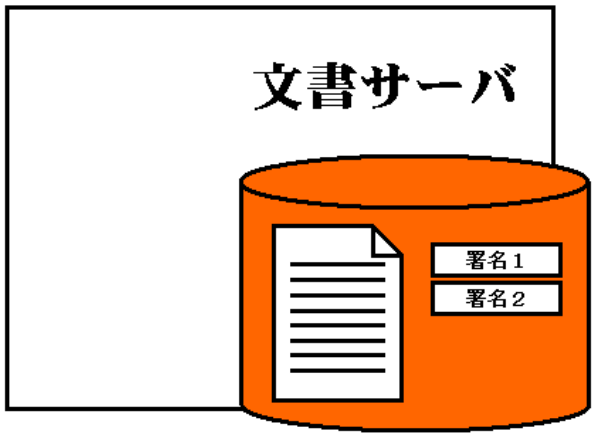
- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン
- 文書サーバから研究記録を取り寄せる

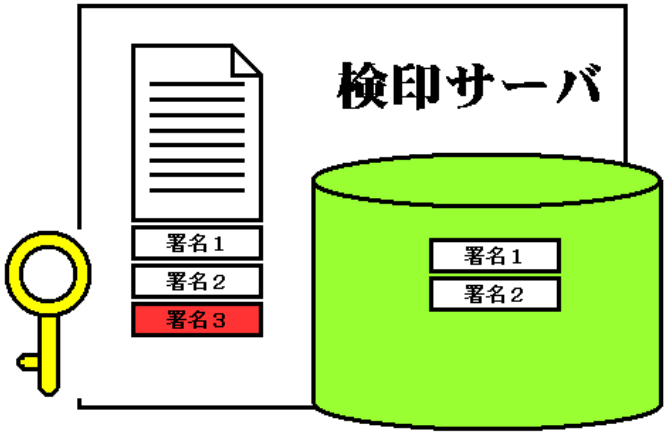
四重署名処理の流れ



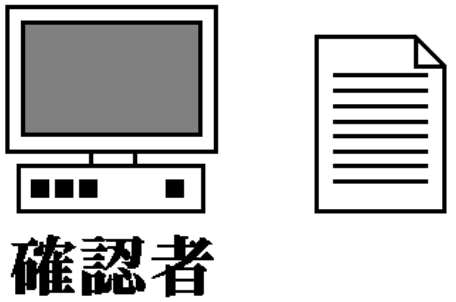
- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン
- 文書サーバから研究記録を取り寄せる
- 研究記録の内容を確認

四重署名処理の流れ

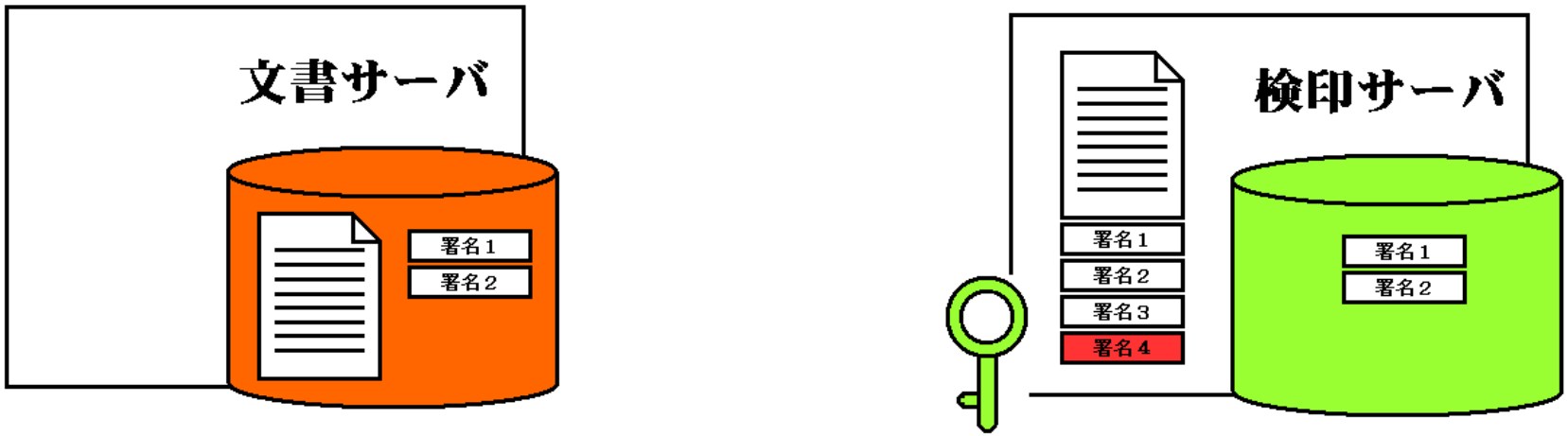




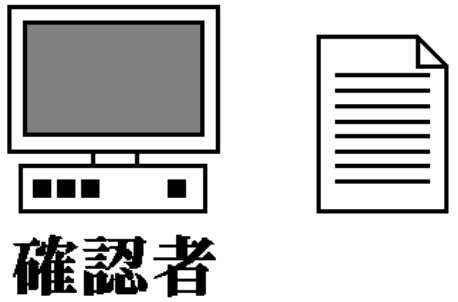
- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン
- 文書サーバから研究記録を取り寄せる
- 研究記録の内容を確認
- 確認者の秘密鍵で署名



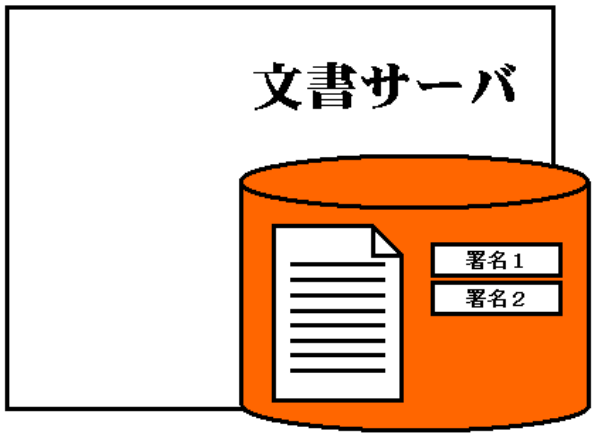
四重署名処理の流れ

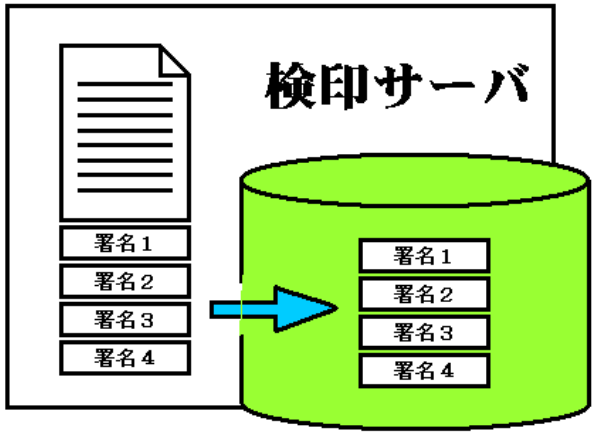


- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン
- 文書サーバから研究記録を取り寄せる
- 研究記録の内容を確認
- 確認者の秘密鍵で署名
- 検印サーバの秘密鍵で署名

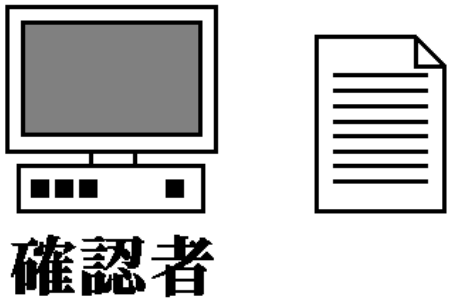


四重署名処理の流れ

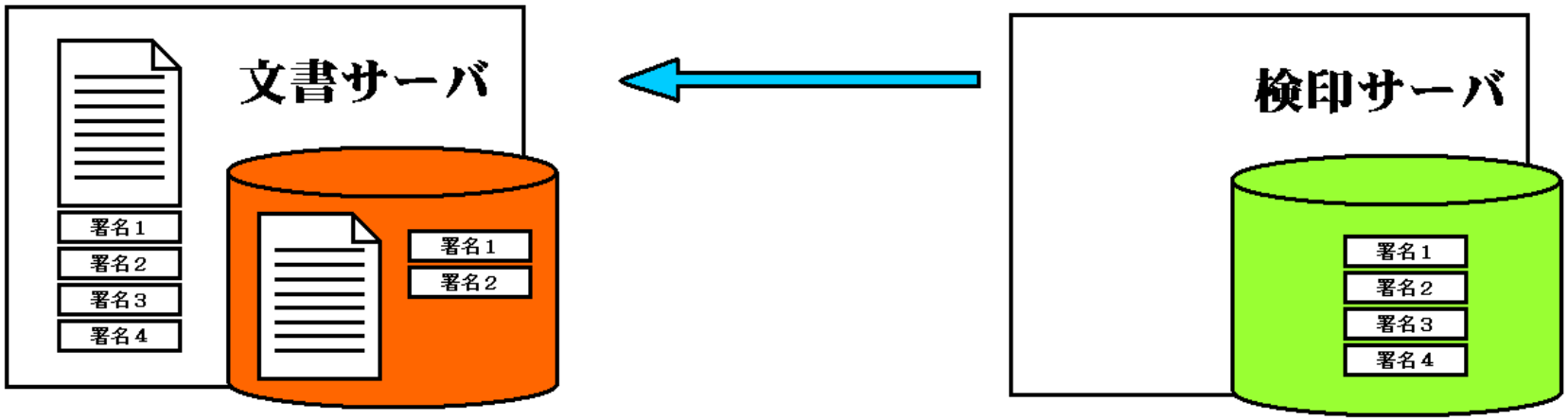




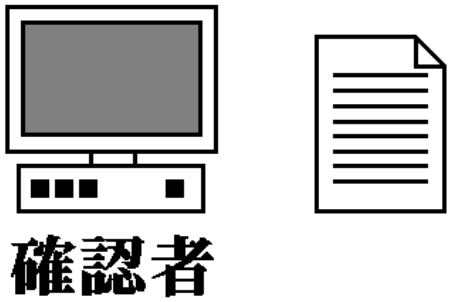
- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン
- 文書サーバから研究記録を取り寄せる
- 研究記録の内容を確認
- 確認者の秘密鍵で署名
- 検印サーバの秘密鍵で署名
- 署名部をデータベースに保存



四重署名処理の流れ

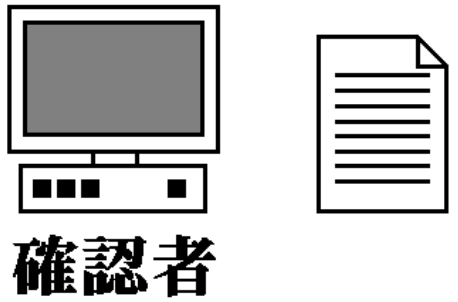


- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン
- 文書サーバから研究記録を取り寄せる
- 研究記録の内容を確認
- 確認者の秘密鍵で署名
- 検印サーバの秘密鍵で署名
- 署名部をデータベースに保存
- 研究記録＋署名部を文書サーバに送信



四重署名処理の流れ

- 確認者が検印サーバにログイン
- 検印サーバを介し文書サーバにログイン
- 文書サーバから研究記録を取り寄せる
- 研究記録の内容を確認
- 確認者の秘密鍵で署名
- 検印サーバの秘密鍵で署名
- 署名部をデータベースに保存
- 研究記録＋署名部を文書サーバに送信
- 研究記録＋署名部をデータベースに保存



確認者の署名

- 研究記録が証拠として機能するためには、本人の確認だけでは不十分
- 内容を確認できる第三者の署名により、研究記録としての証拠能力をもつ
- 研究記録の知的財産としての価値を高める

岡崎康司 , 隅蔵康一　編集 , 「理系なら知っておきたいラボノートの書き方」株式会社羊土社 ,2007 年

確認者に求められる条件

- 共同開発者でないこと
- 利害関係にないこと
- 記録の内容を理解できること

→arχves の確認者の署名機能によって、
距離的に離れた相手でも記録の確認ができる

岡崎康司 , 隅蔵康一　編集 ,「理系なら知っておきたいラボノートの書き方」
株式会社羊土社 ,2007 年

Web 認証基盤とは

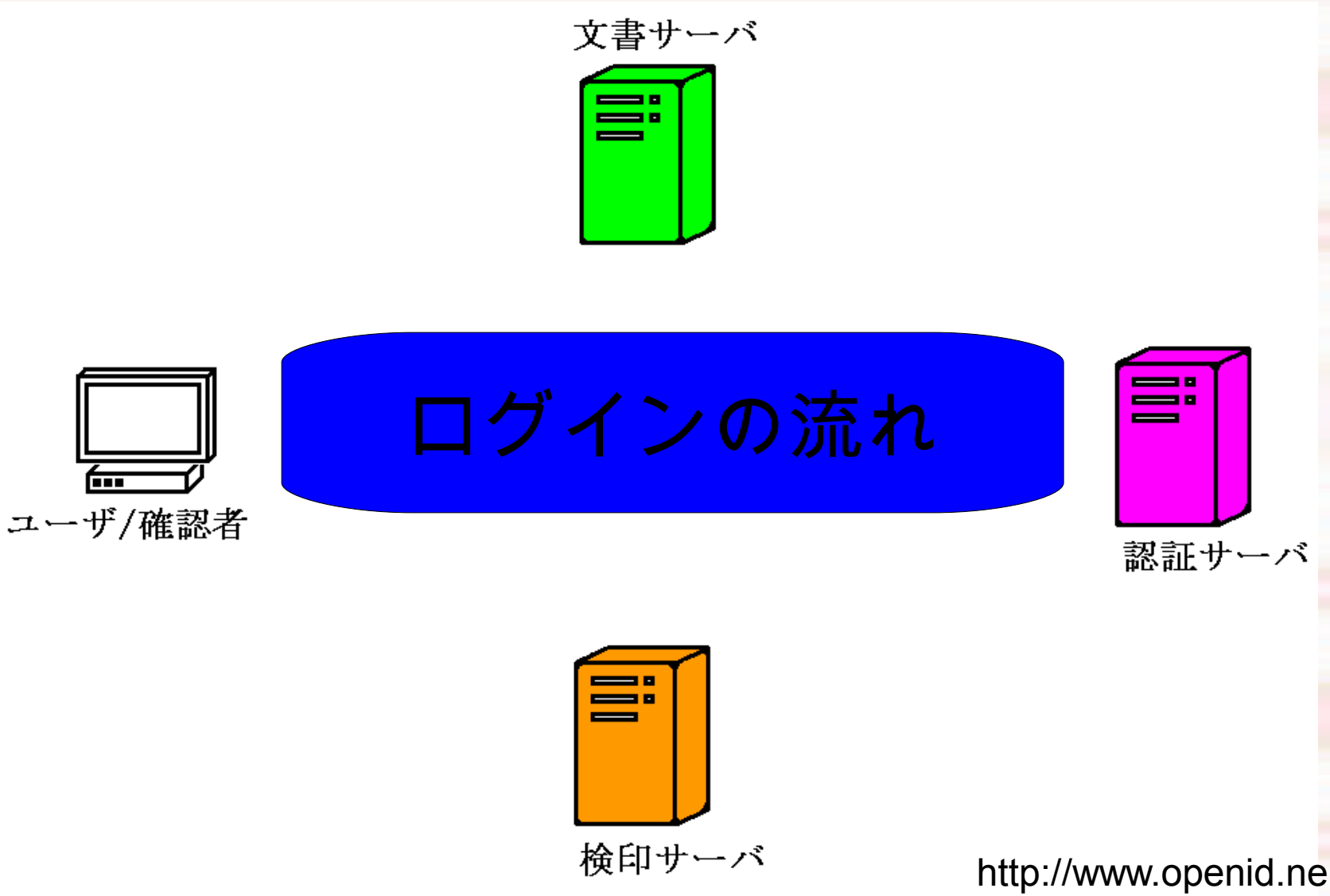
- Ｗｅｂ認証を１つのサーバで行うシステム
- 認証サーバを複数のシステムで利用するシステム
- シングルサインオンを実現

Web 認証基盤とは

- Ｗｅｂ認証を１つのサーバで行うシステム
- 認証サーバを複数のシステムで利用するシステム
- シングルサインオンを実現

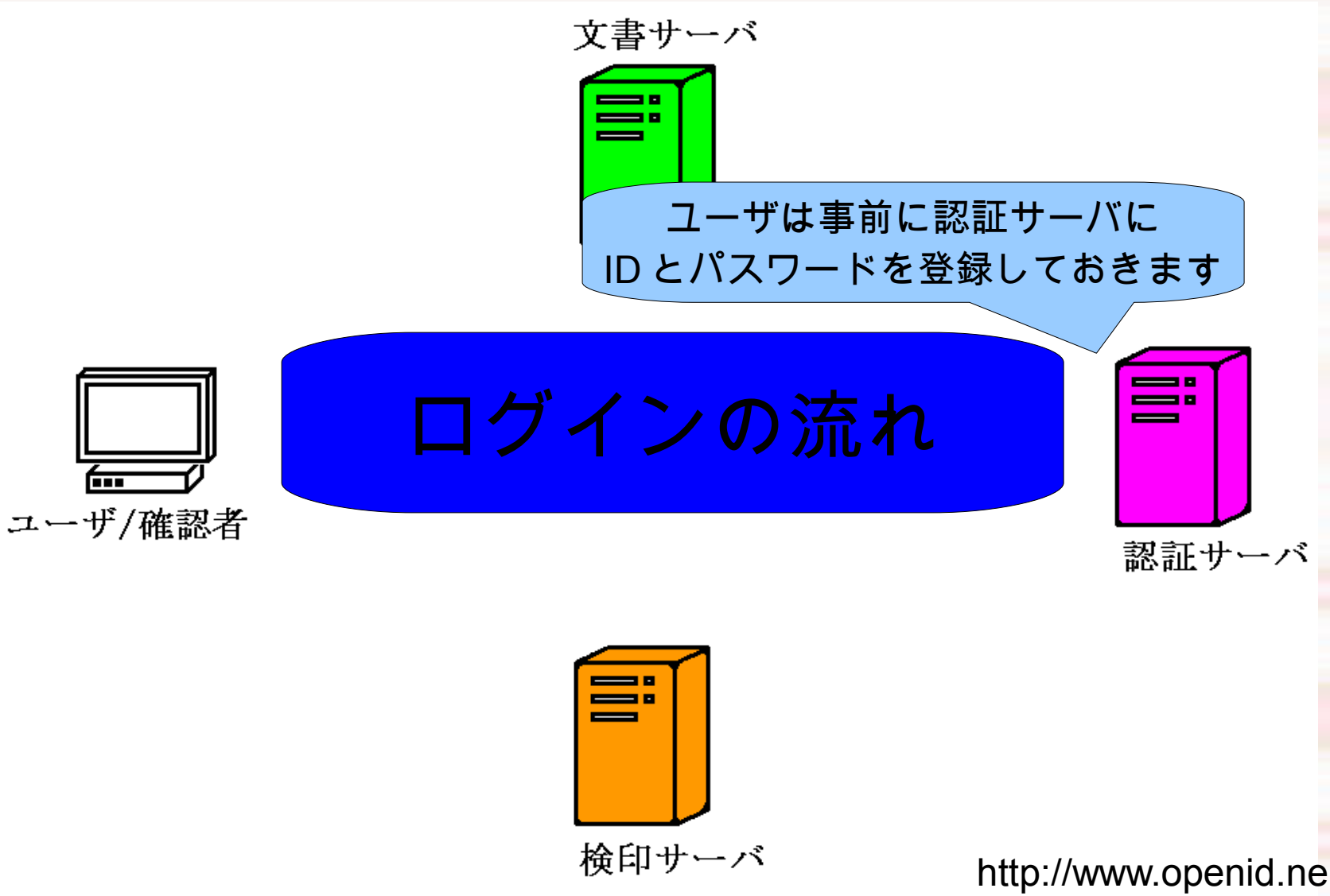
シングルサインオンとは
- 一度ユーザ認証を行えば、
  複数のサーバでのユーザ認証をパスできること
- 登録のたびにIDとパスワードを考えなくてよい
- 覚えるパスワードの数が減るので、
  複雑なパスワードを設定しやすい

ログインの流れ



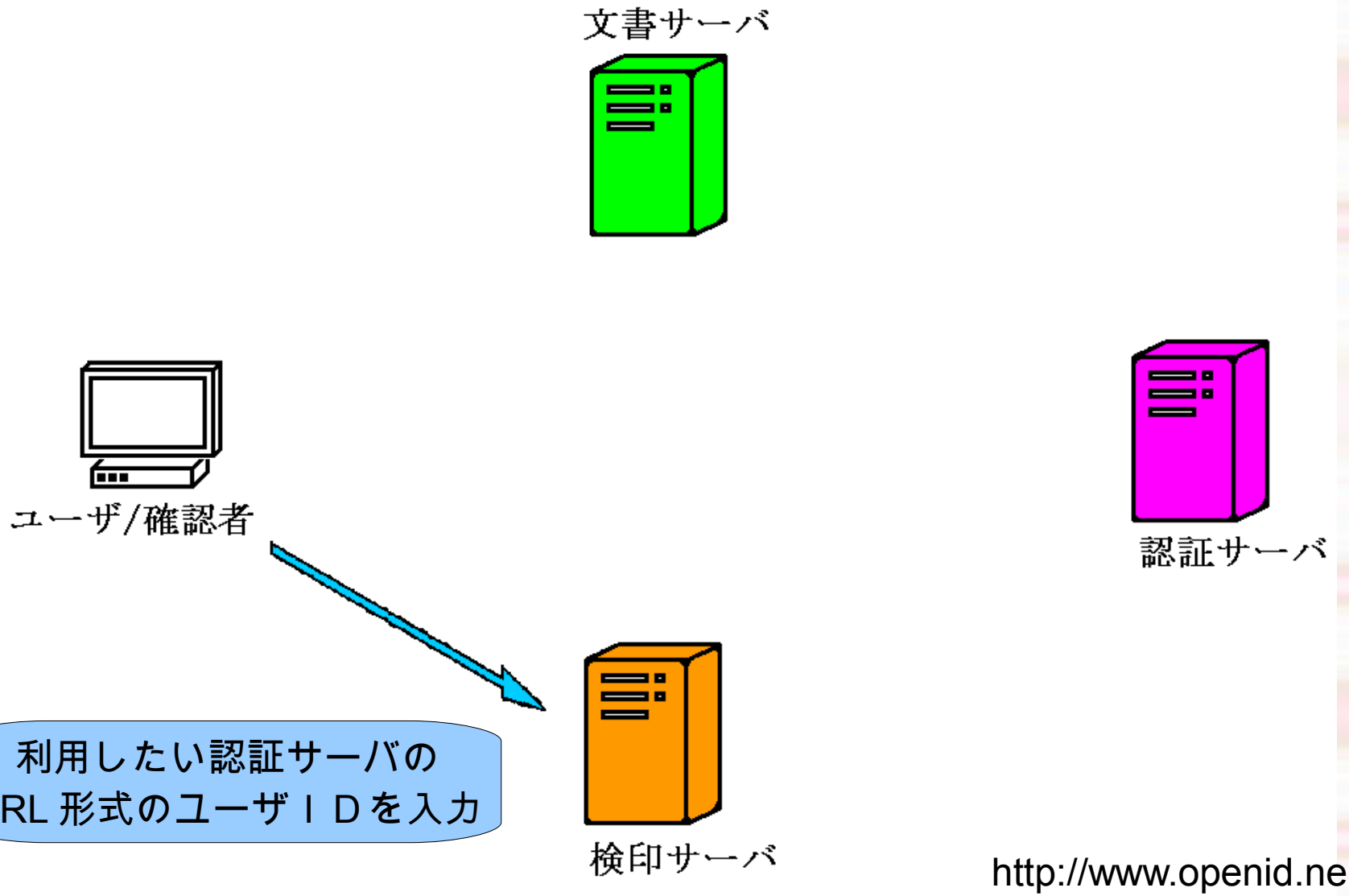
http://www.openid.ne.jp/

ログインの流れ



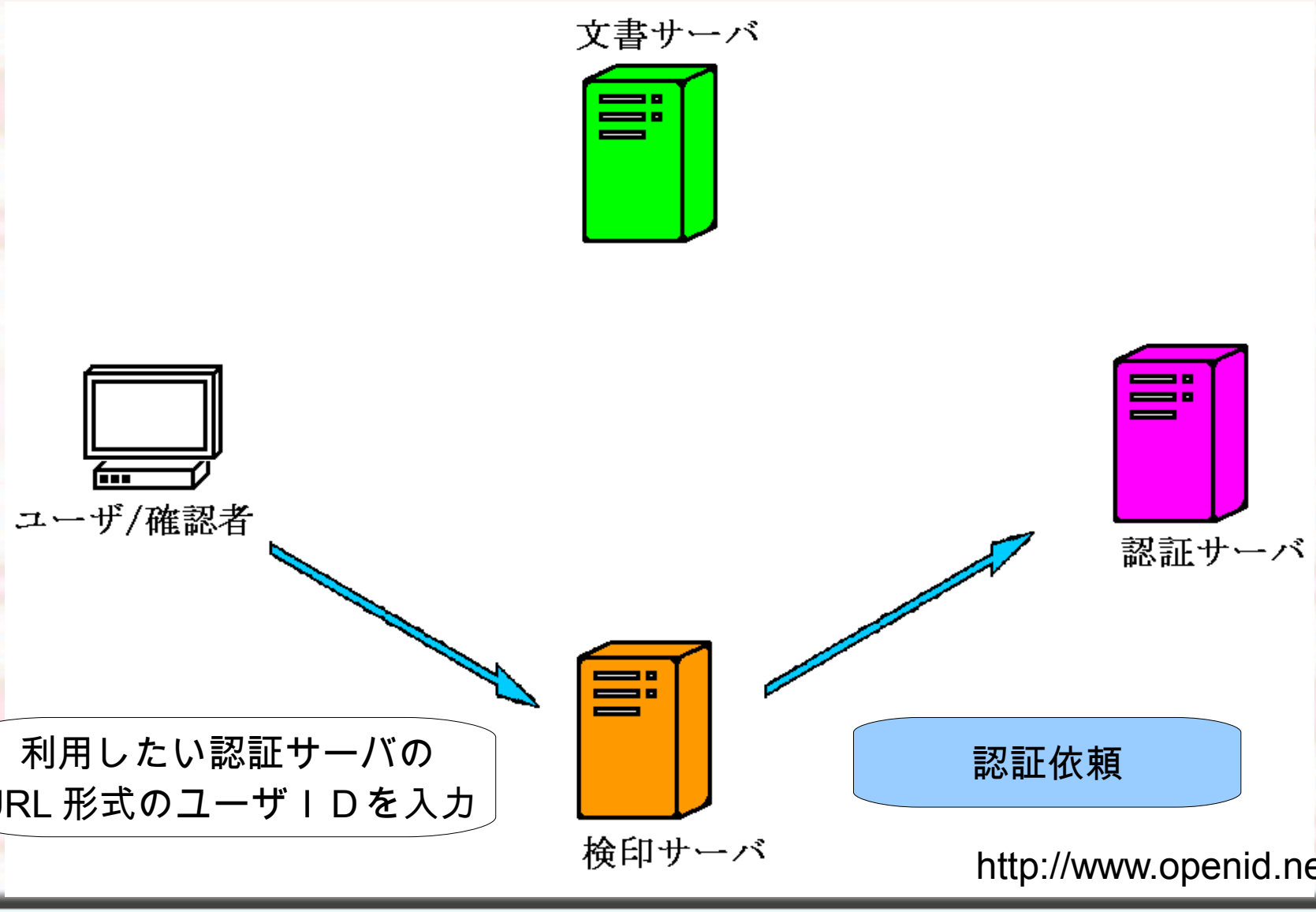
http://www.openid.ne.jp/

ログインの流れ

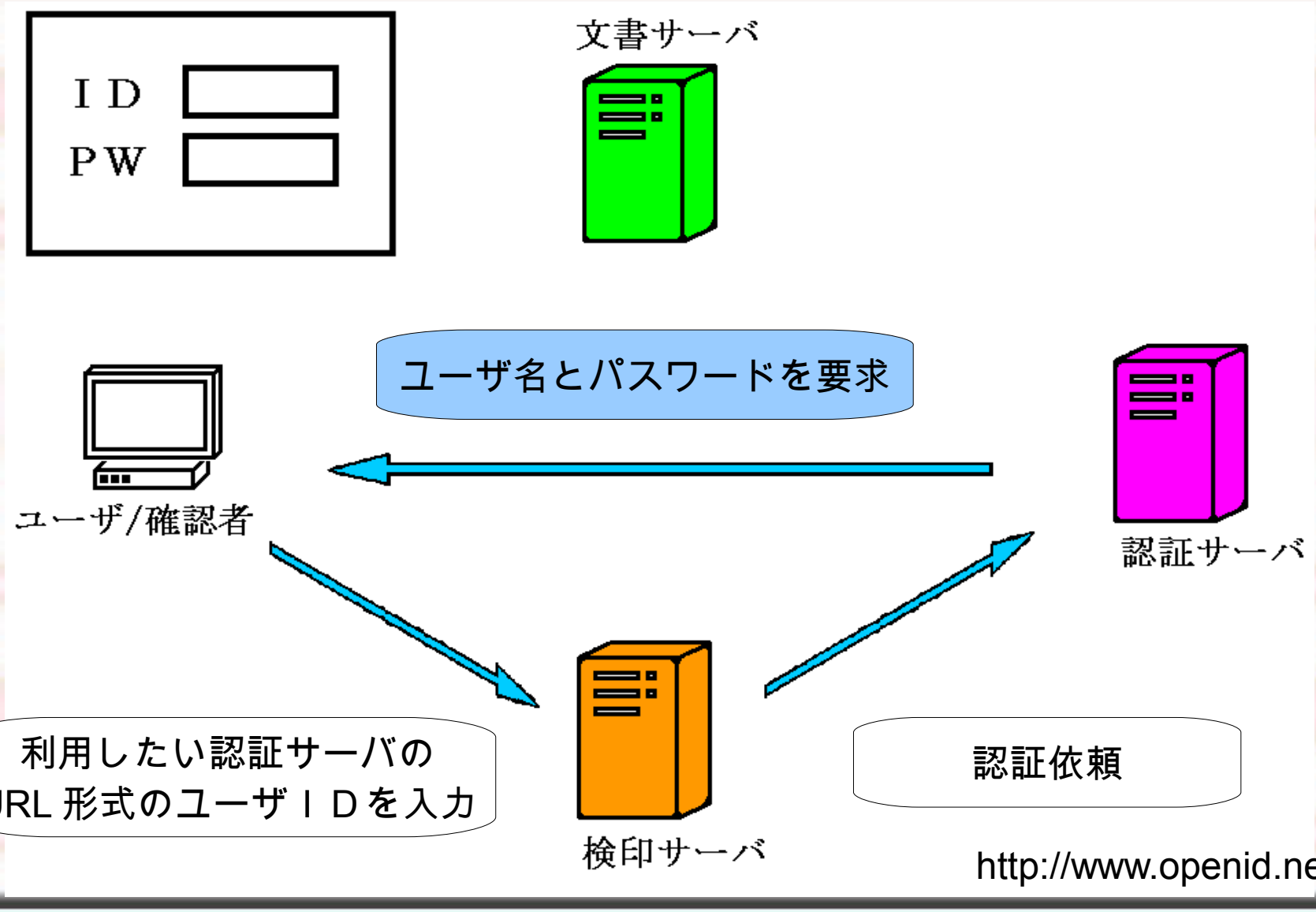


http://www.openid.ne.jp/

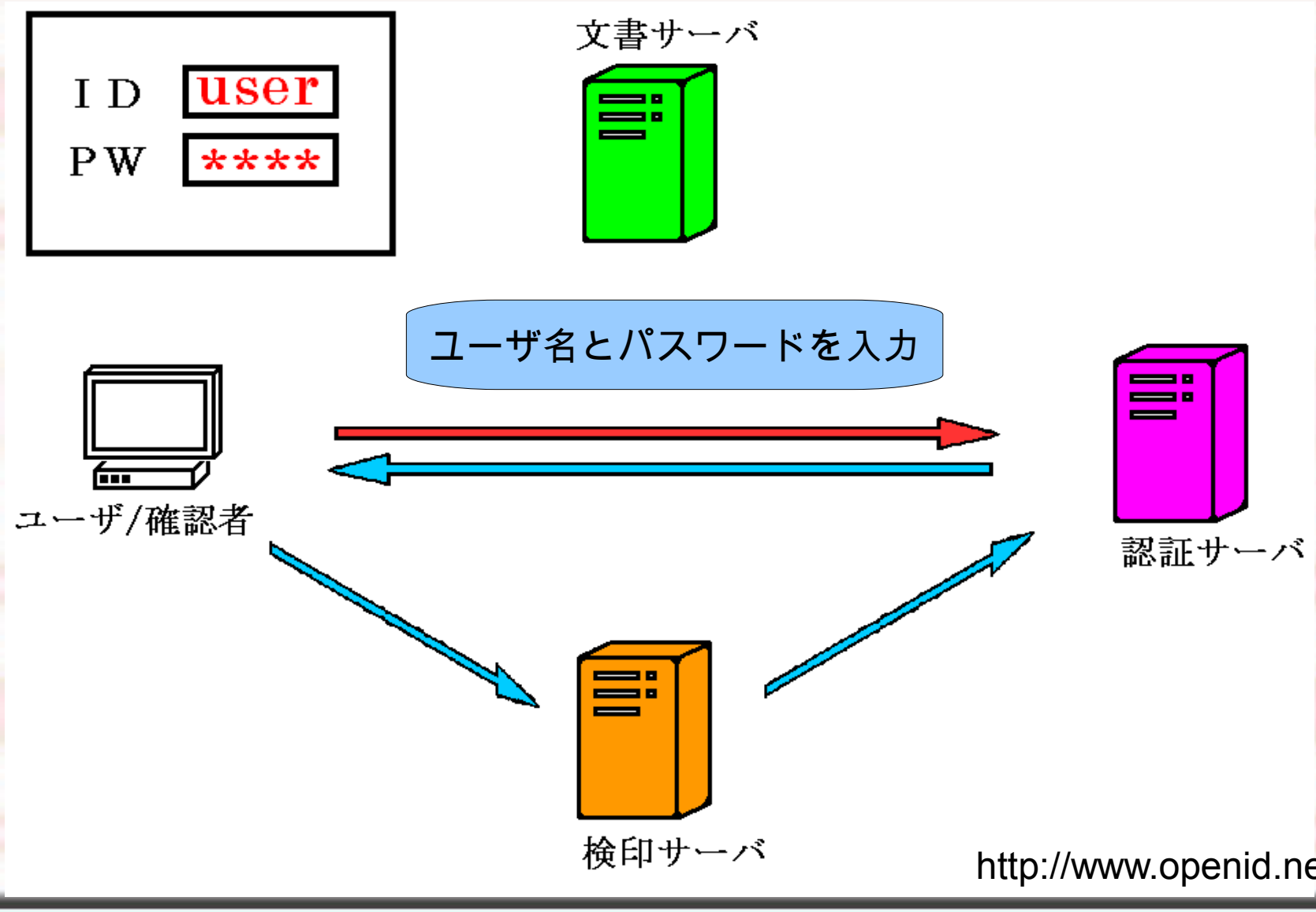
ログインの流れ


文書サーバ
ユーザ/確認者
認証サーバ
利用したい認証サーバの
URL 形式のユーザＩＤを入力
検印サーバ
認証依頼
http://www.openid.ne.jp/

ログインの流れ

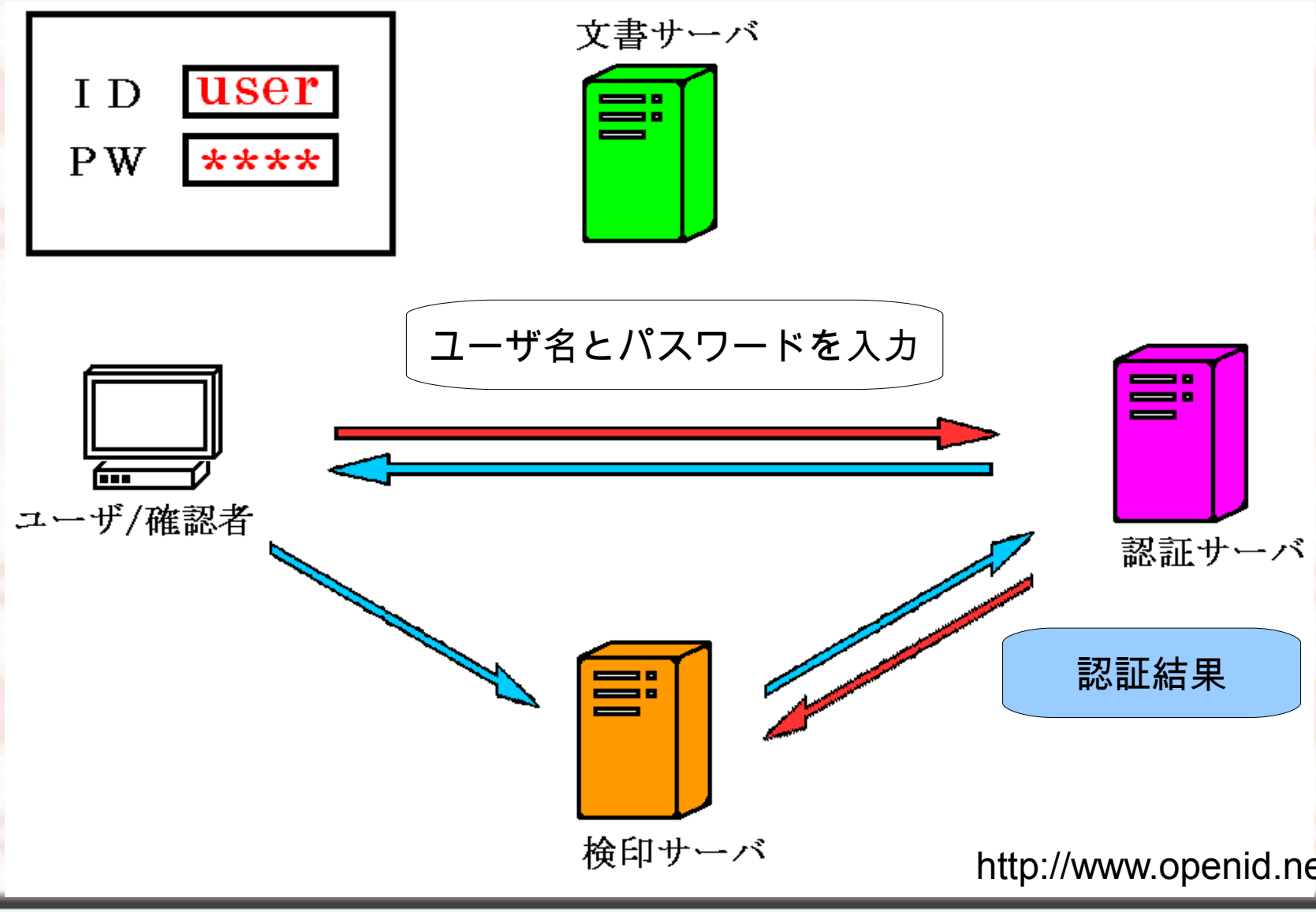

ＩＤ
ＰＷ
文書サーバ
ユーザ名とパスワードを要求
ユーザ/確認者
認証サーバ
利用したい認証サーバの
URL 形式のユーザＩＤを入力
検印サーバ
認証依頼
http://www.openid.ne.jp/

ログインの流れ

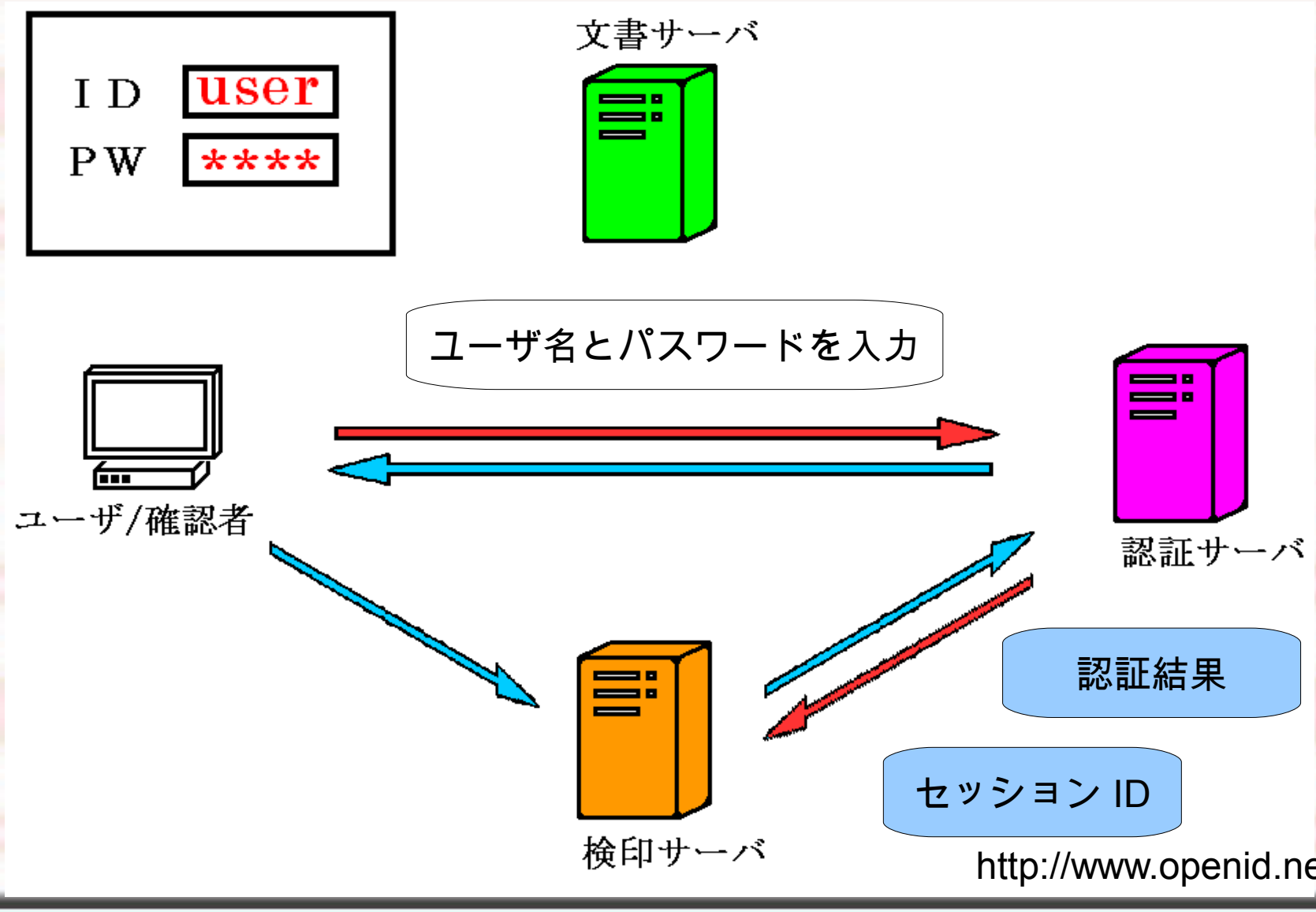


http://www.openid.ne.jp/

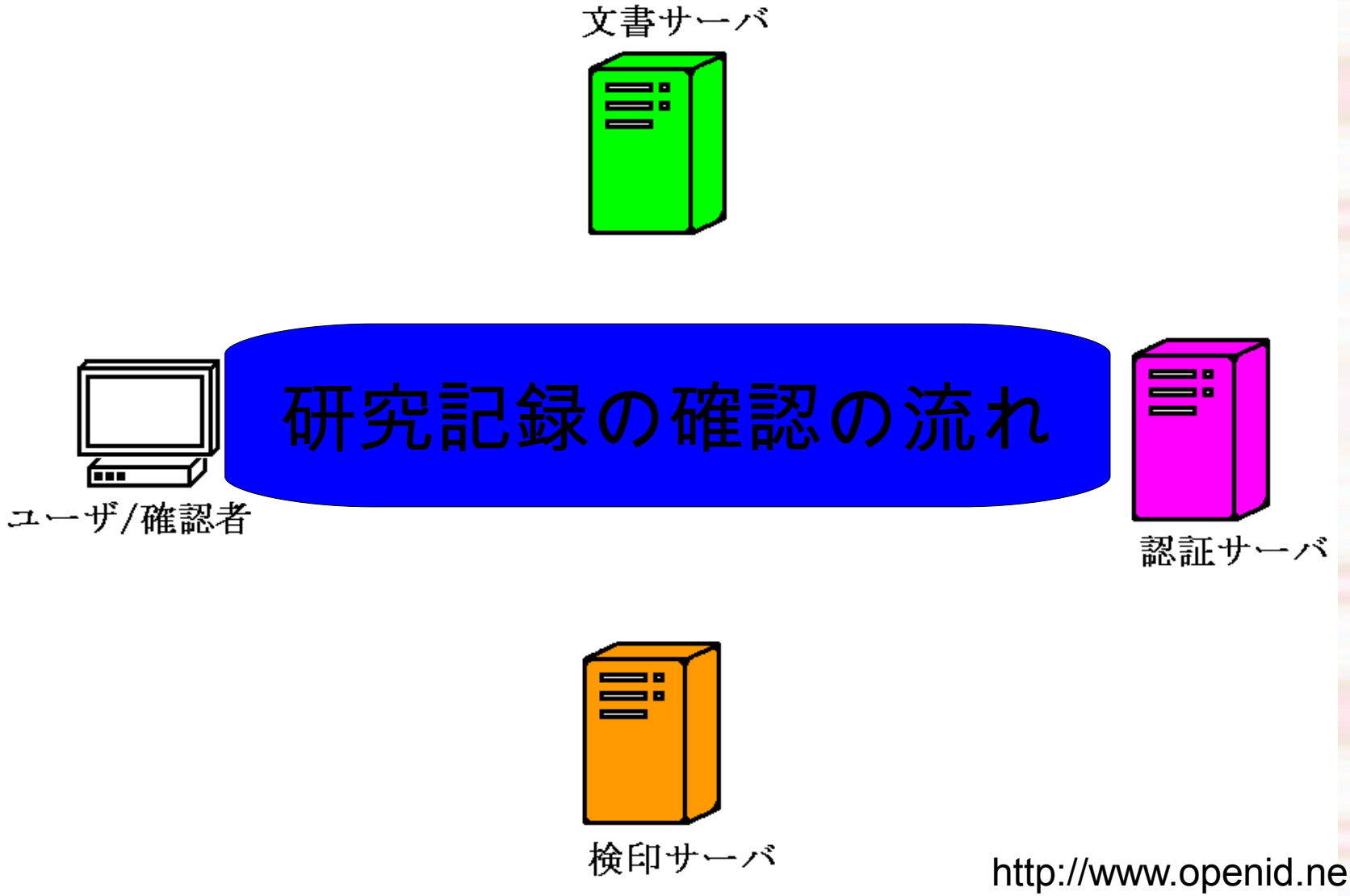
ログインの流れ


ＩＤ user
ＰＷ ****
文書サーバ
ユーザ名とパスワードを入力
ユーザ/確認者
認証サーバ
認証結果
検印サーバ
http://www.openid.ne.jp/

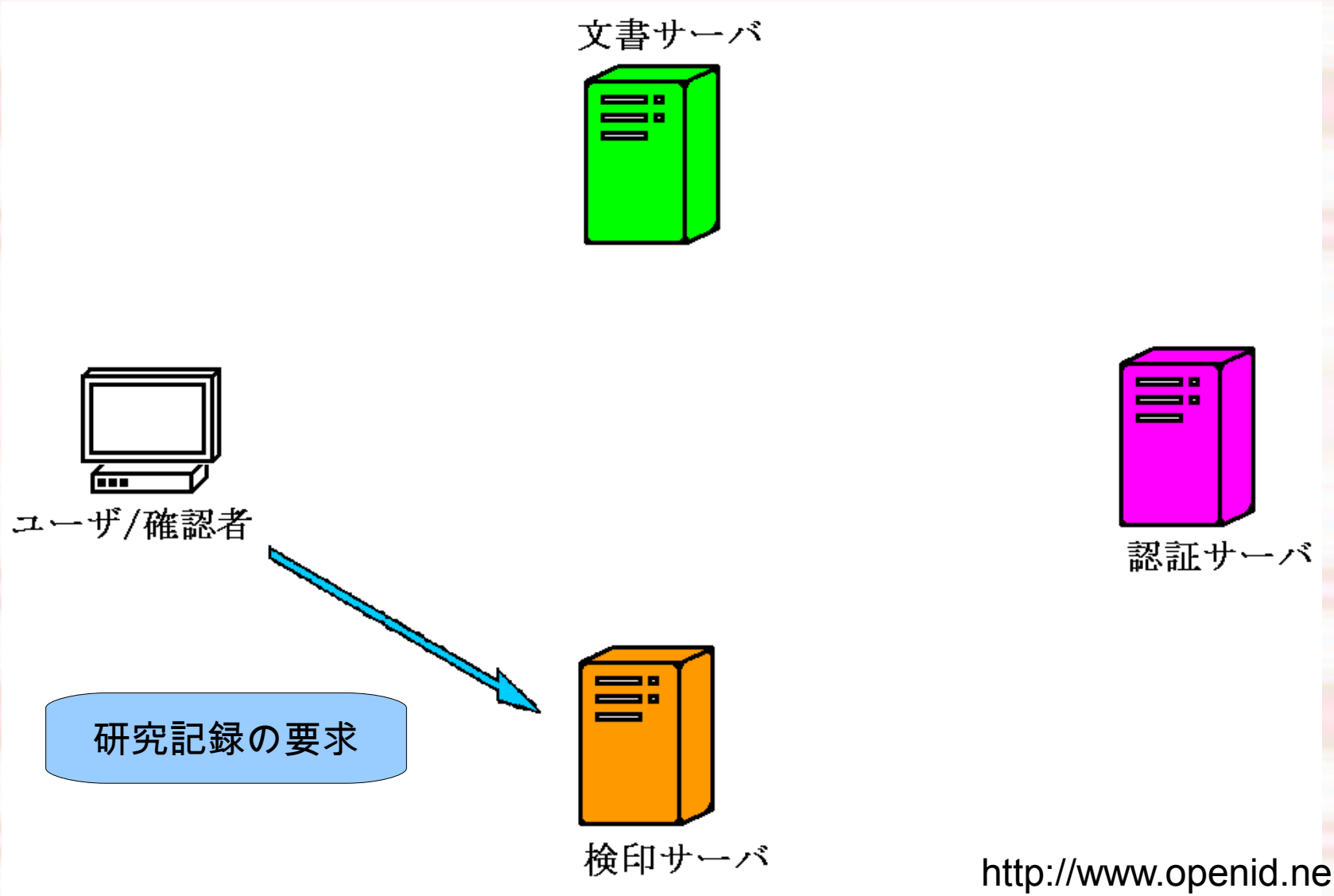
ログインの流れ


ＩＤ user
ＰＷ ****
文書サーバ
ユーザ名とパスワードを入力
ユーザ/確認者
認証サーバ
認証結果
セッション ID
検印サーバ
http://www.openid.ne.jp/

研究記録の確認の流れ

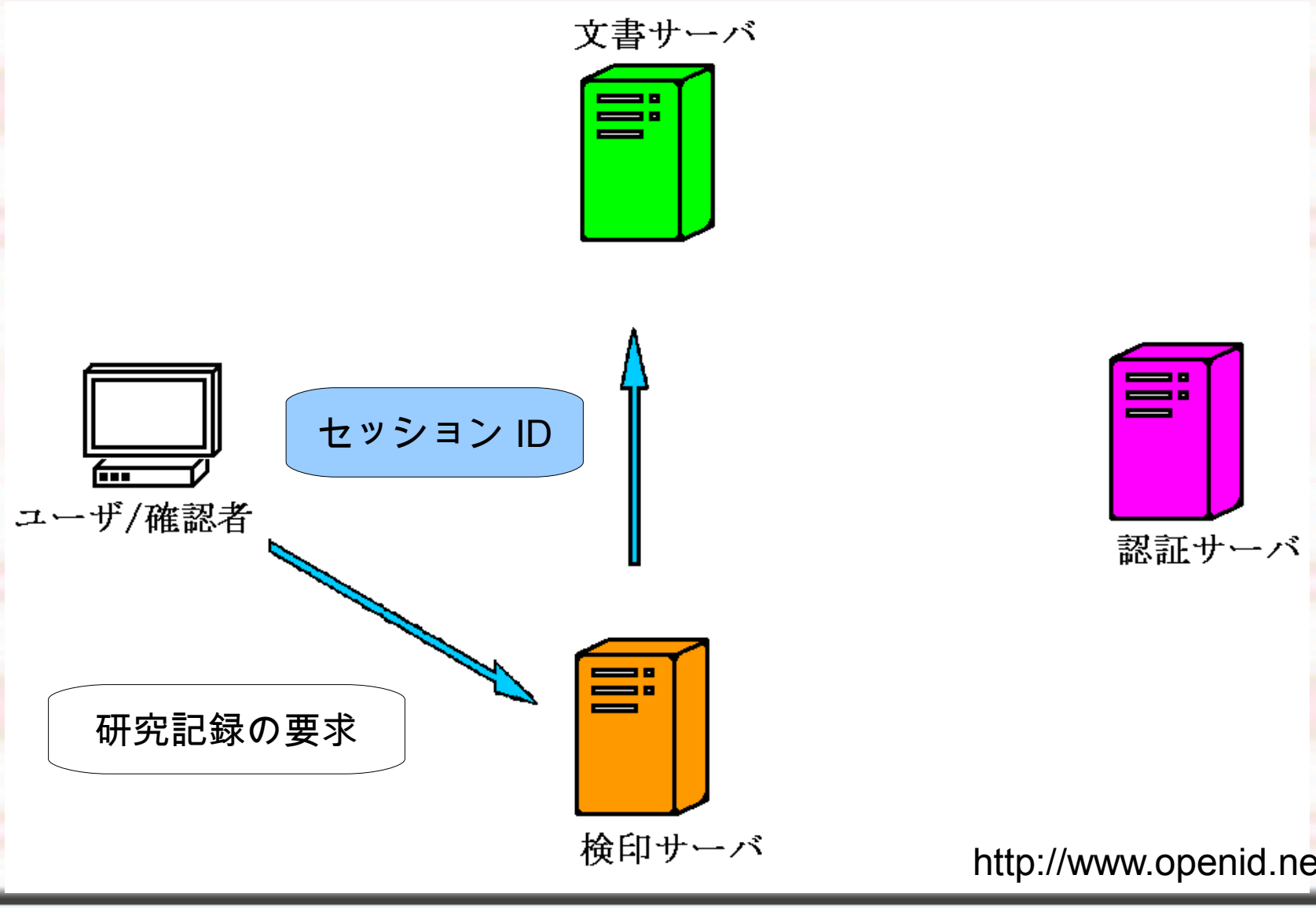


研究記録の確認の流れ



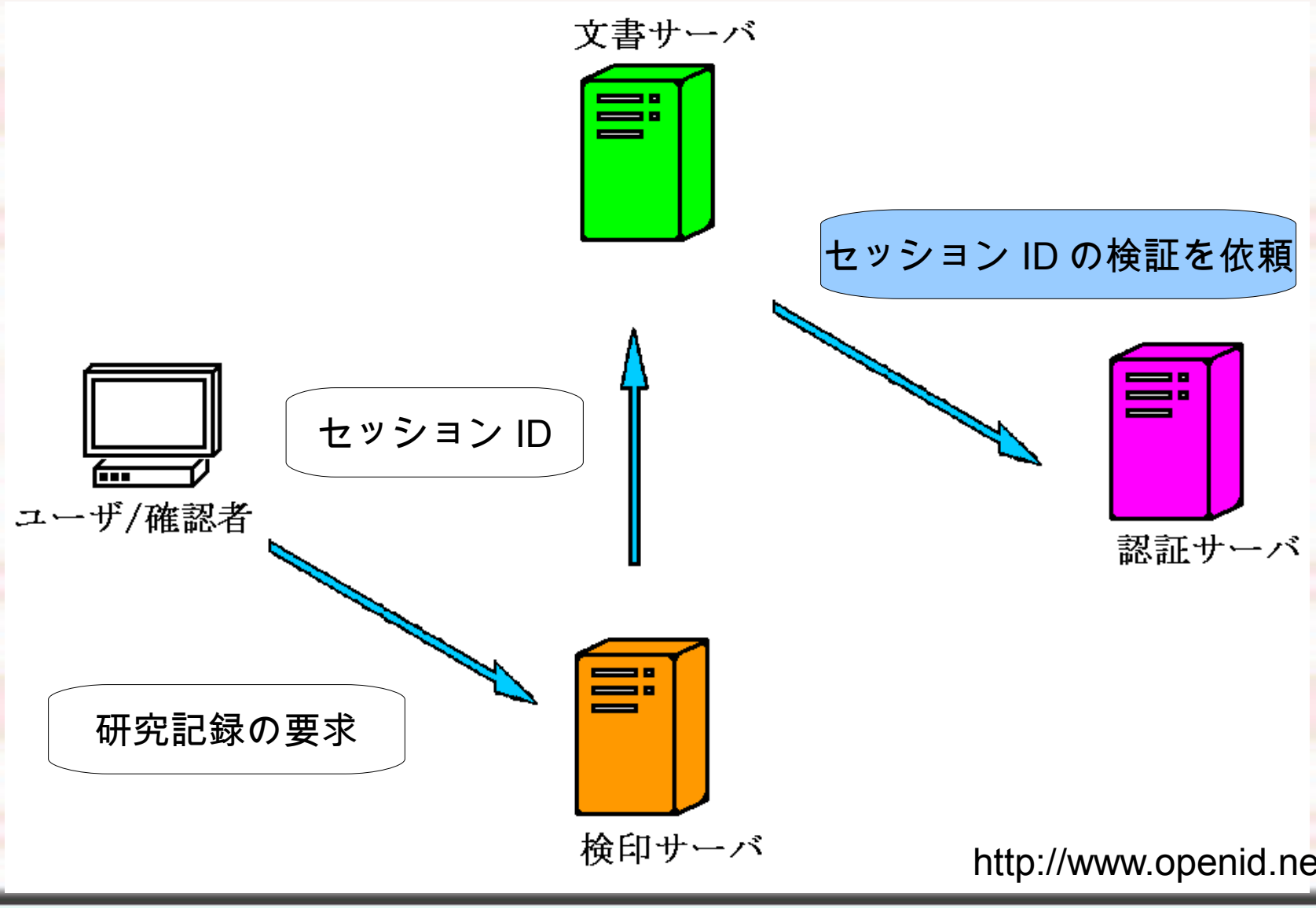
http://www.openid.ne.jp/

研究記録の確認の流れ



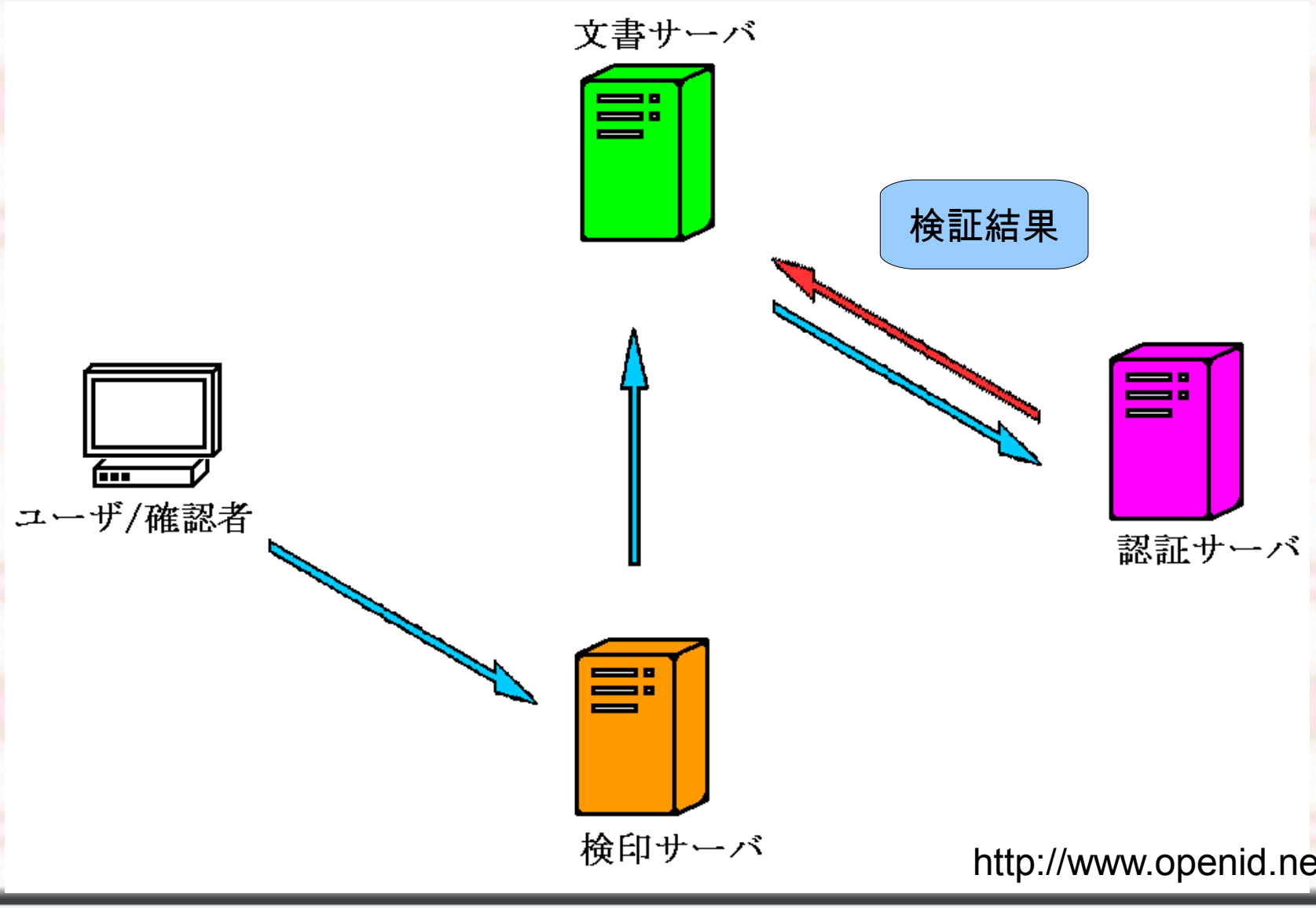
http://www.openid.ne.jp/

研究記録の確認の流れ

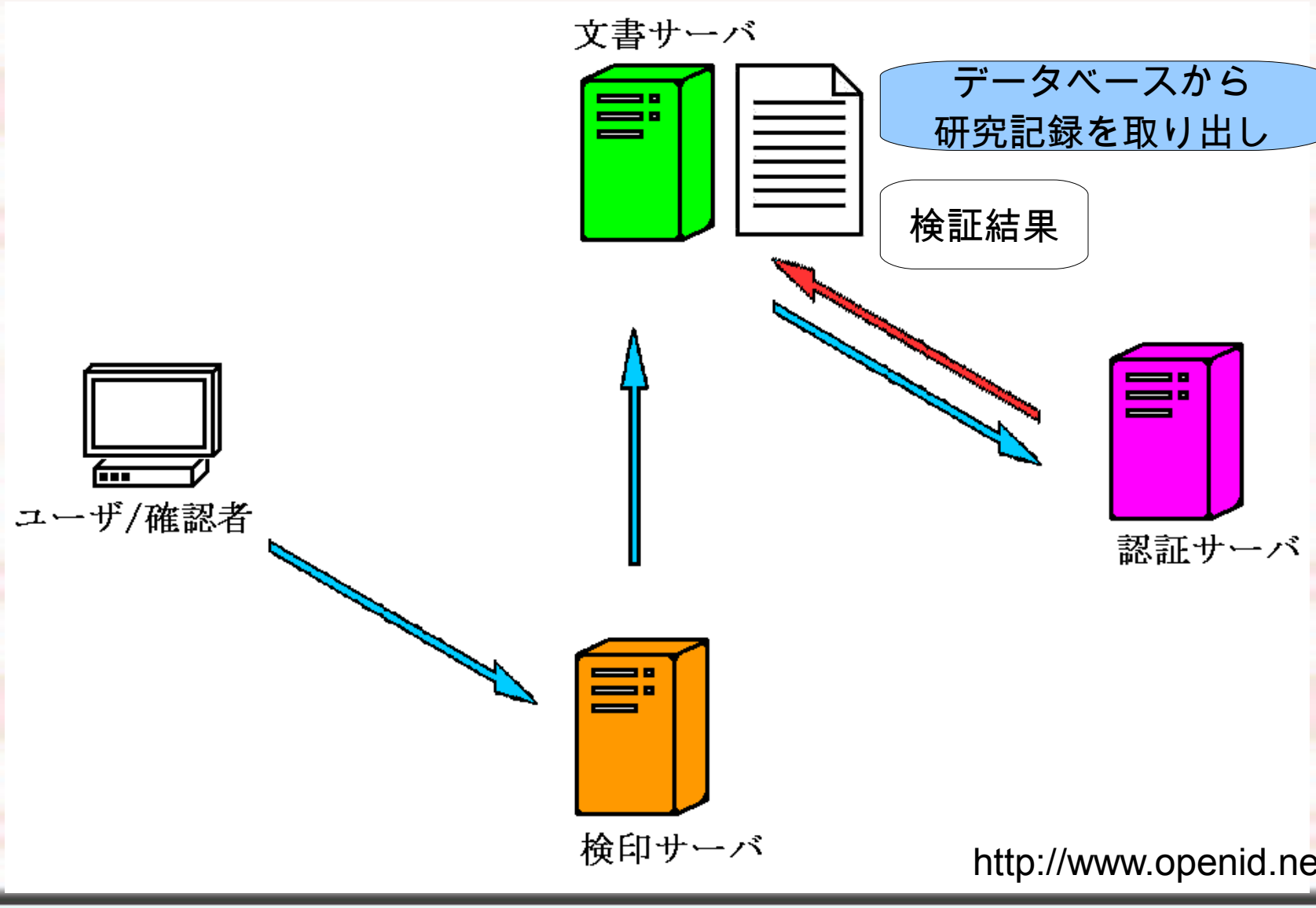


http://www.openid.ne.jp/

研究記録の確認の流れ

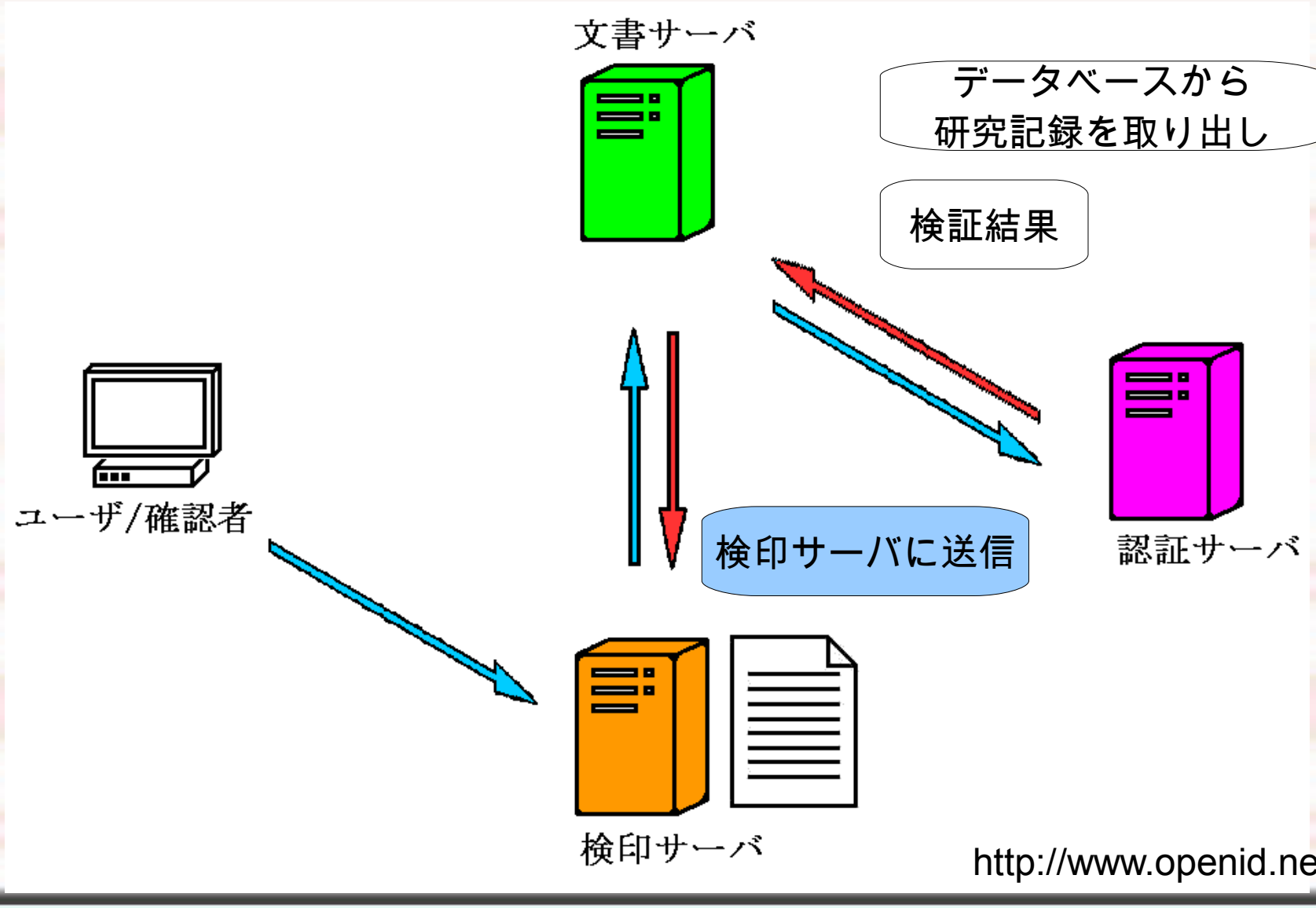


研究記録の確認の流れ



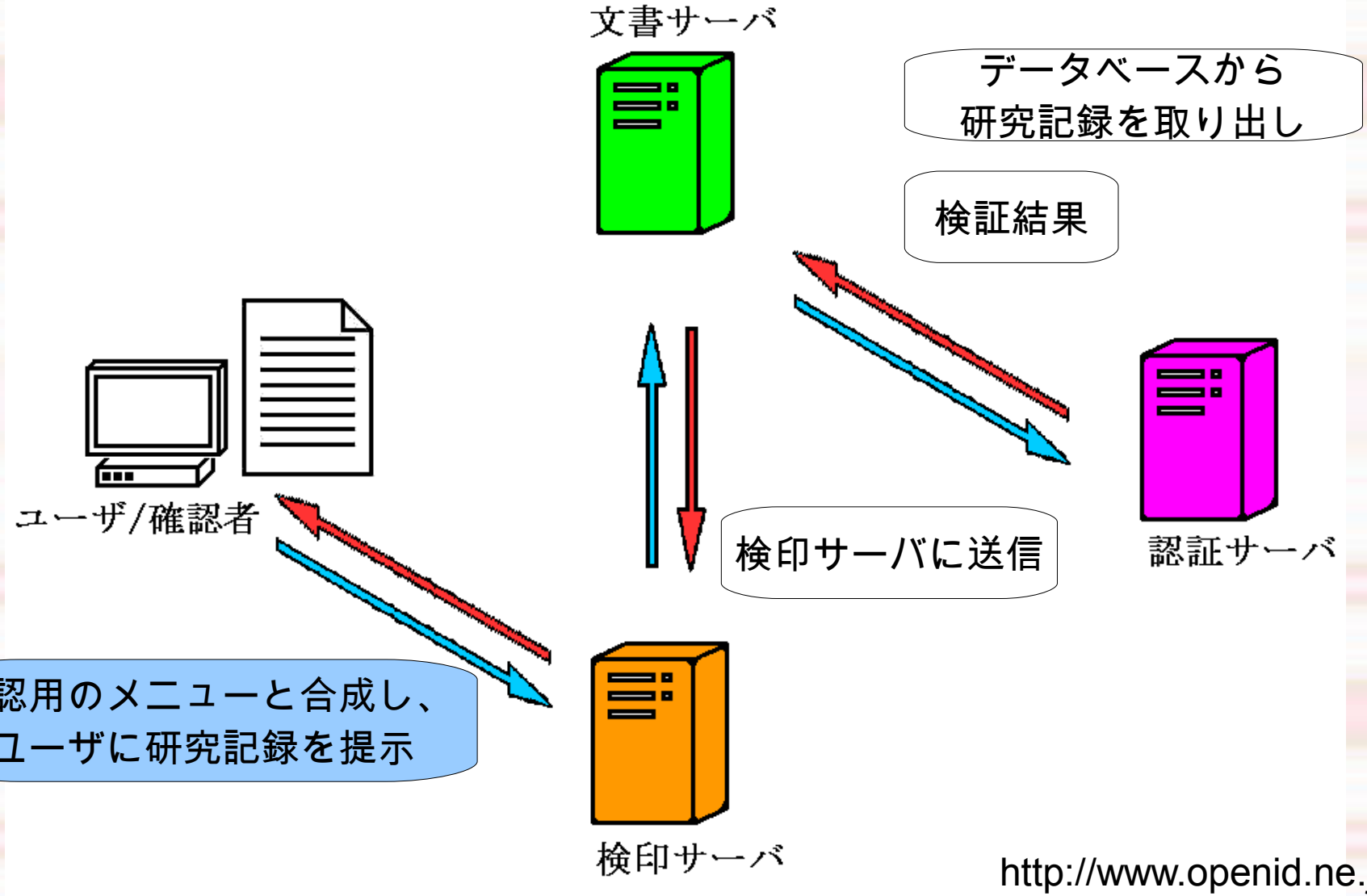
http://www.openid.ne.jp/

研究記録の確認の流れ



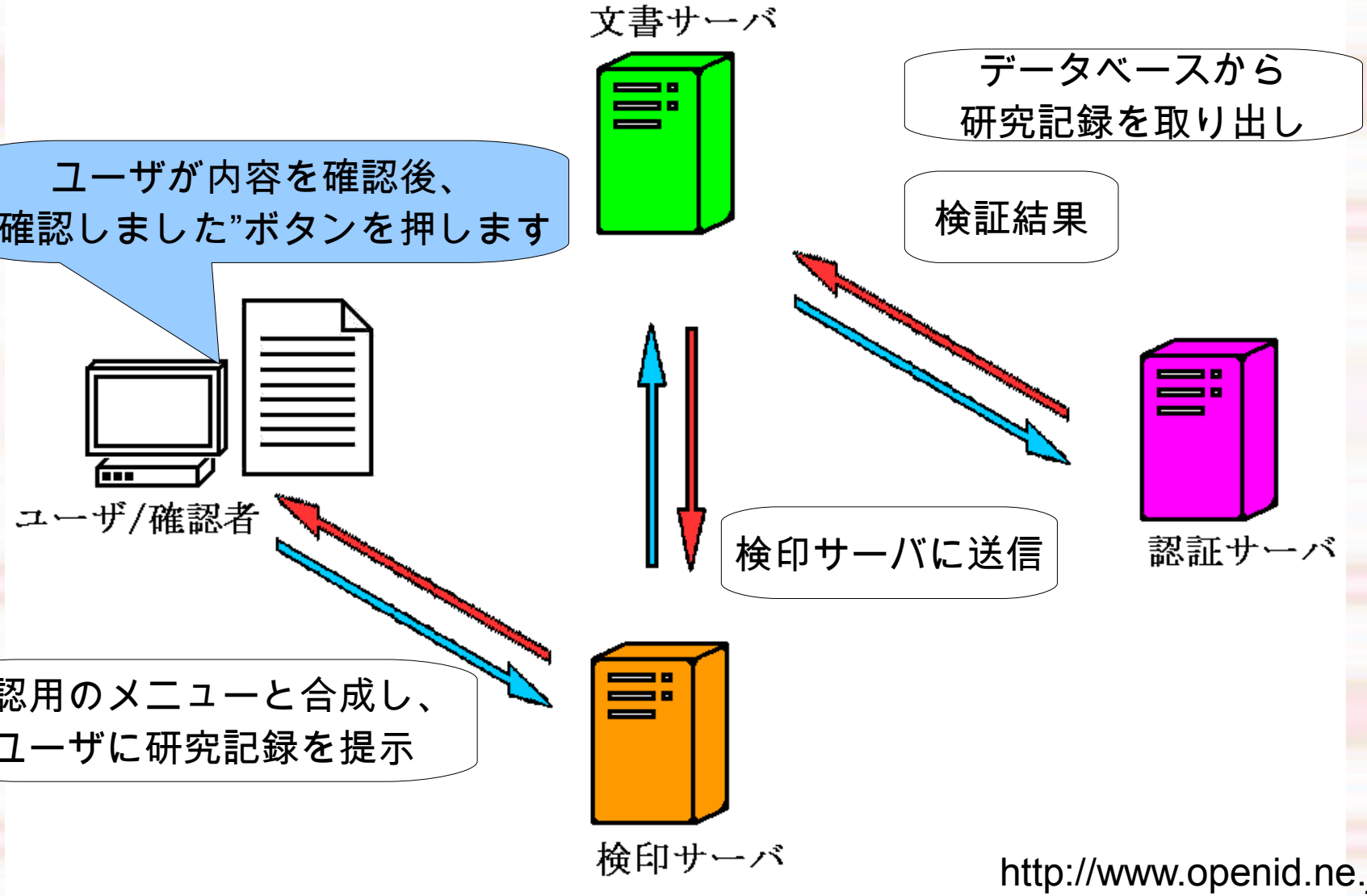
http://www.openid.ne.jp/

研究記録の確認の流れ



http://www.openid.ne.jp/

研究記録の確認の流れ



http://www.openid.ne.jp/

OpenID とは

- １つの ID を異なるウェブサイトで利用できるようにする認証システムのこと
- 対応サイトで利用される ID の名称
- ユーザ固有の URL 形式の ID を用いる
- 認証サービスを行うサイトを選べる
- 認証の結果と必要最小限の情報のみをサービス側に通知
- サービス側は ID/ パスワードを保有・管理しない

Shibboleth

- 米国の EDUCAUSE 標準の認証システム
- 国立情報学研究所が学認として支援している
- 認証は所属大学で実施
- 学内でのシングルサインオンを実現可能
- 学外の公開サービスへのシングルサインオンを実現可能

GakuNin

OpenID を採用した理由

OpenID のメリット	Shibboleth のデメリット
•資料などが多い •既存の認証サーバを利用可能	•日本語の資料が少ない •独自に認証サーバの導入が必要

Web 認証基盤対応による効用

- １回のログインで各機能を利用可能

  システム利用において、別のサーバにアクセスするたびに、ログインし直す必要を回避

- セキュリティ強化

  検印サーバを介して文書サーバにアクセスする際の、ID とパスワードを盗難される可能性を防止

今後の課題

- 研究記録を対象として、自然言語処理・知的処理を行う
- 携帯情報端末との連携を行う

以上、ご清聴ありがとうございました。

公開範囲の設定

- 研究記録ごとに公開範囲を設定
- チェックボックスによる公開範囲の選択
- 階層構造

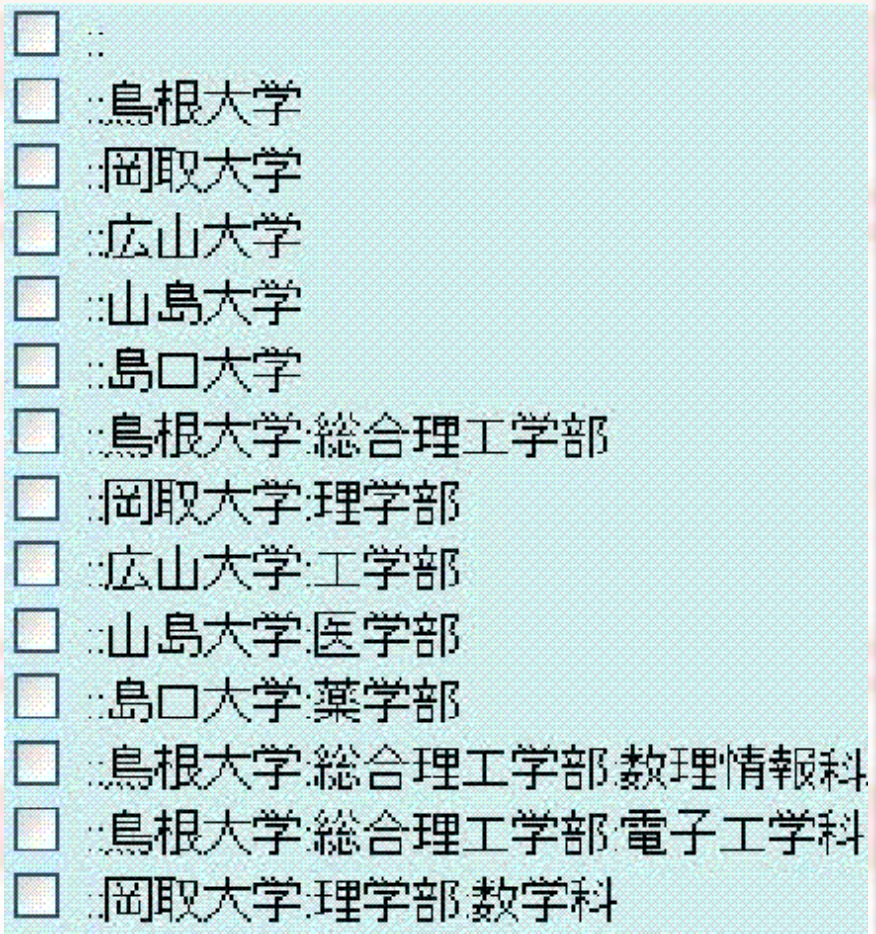
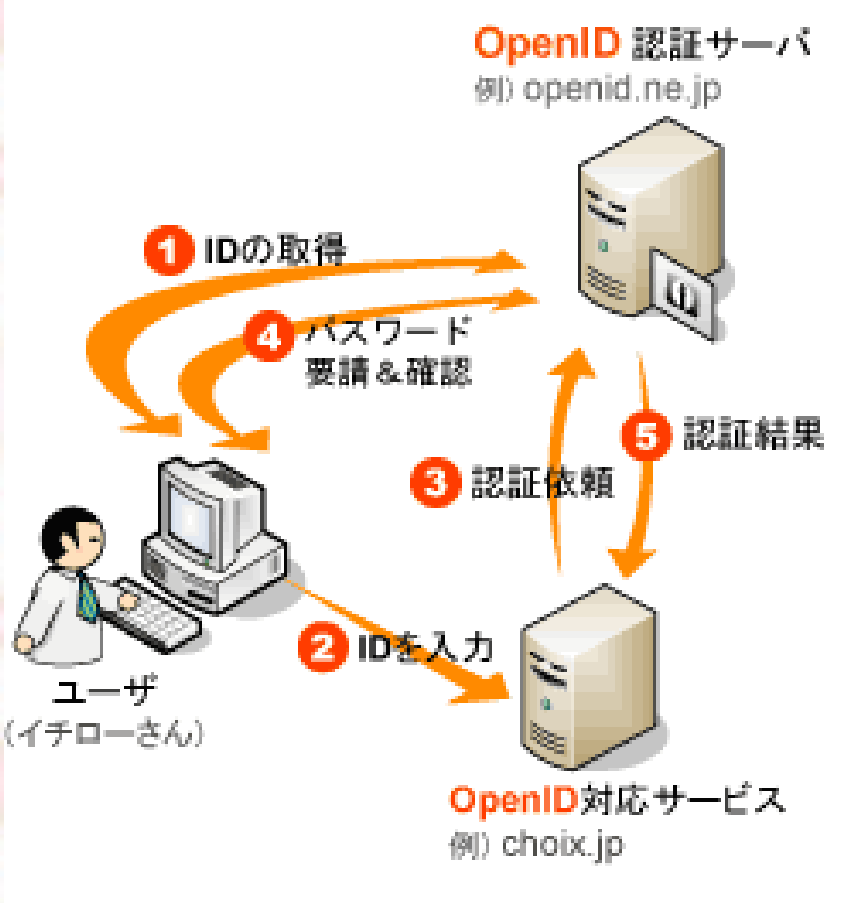


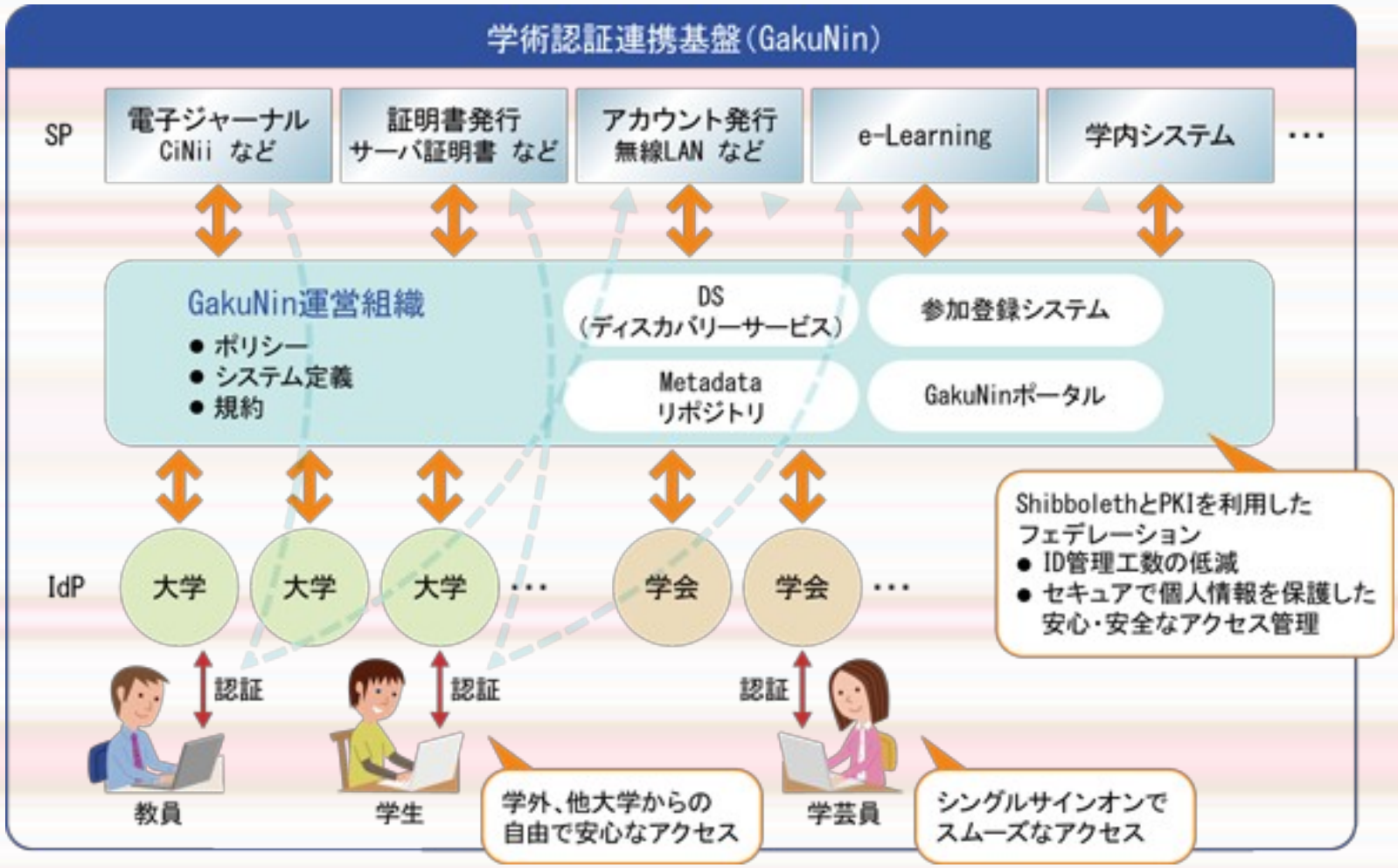
図１．公開グループ選択画面

OpenIDでの認証の流れ



http://www.openid.ne.jp/

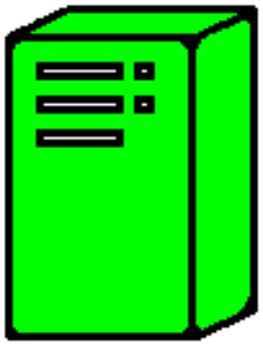
学術認証連携基盤とは



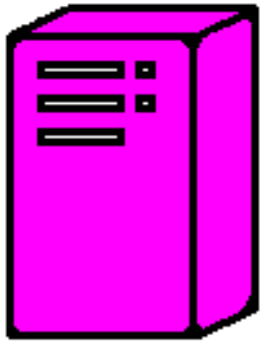
http://www.gakunin.jp/

アクセスの流れ

認証サーバ



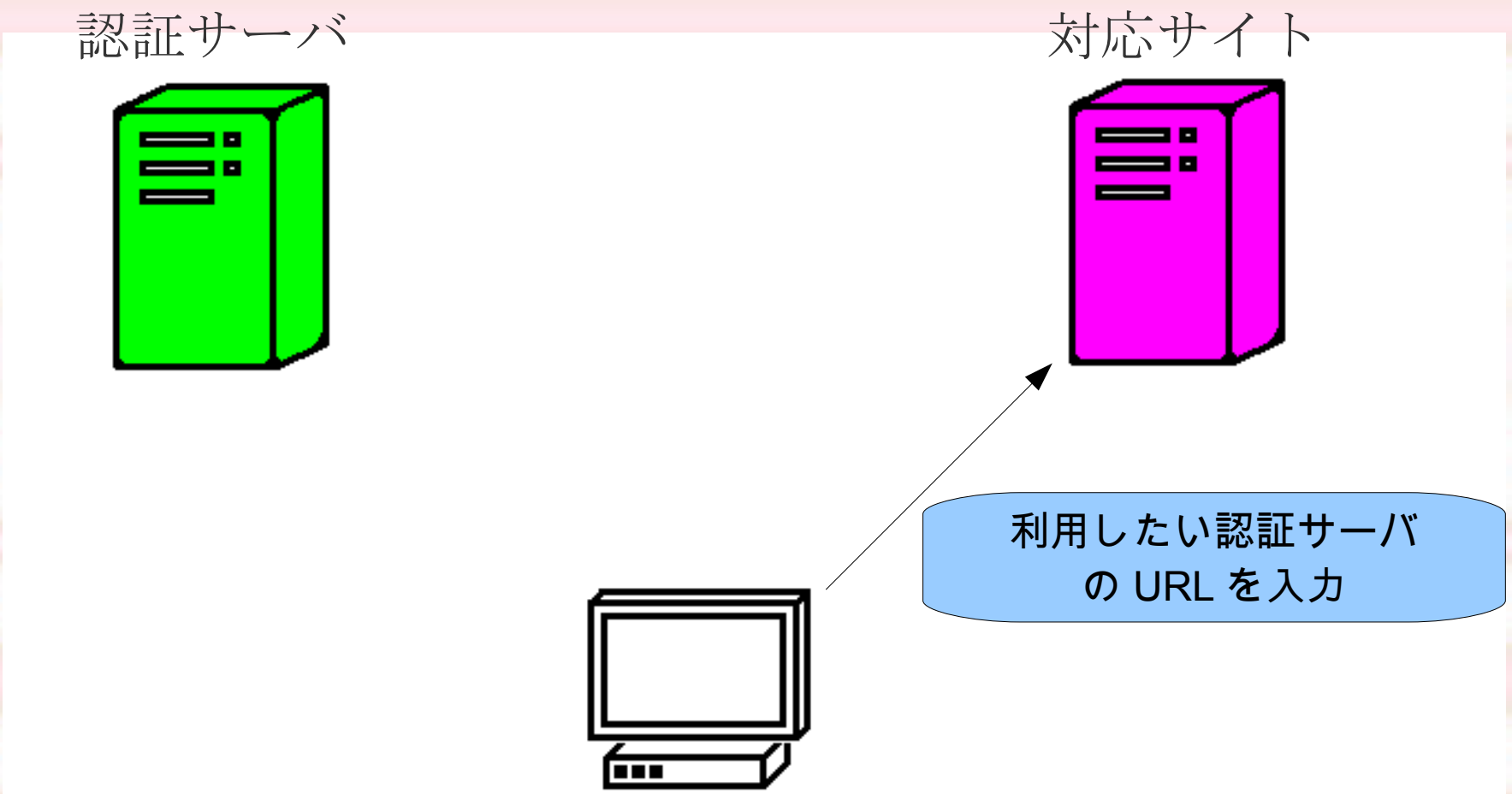
対応サイト



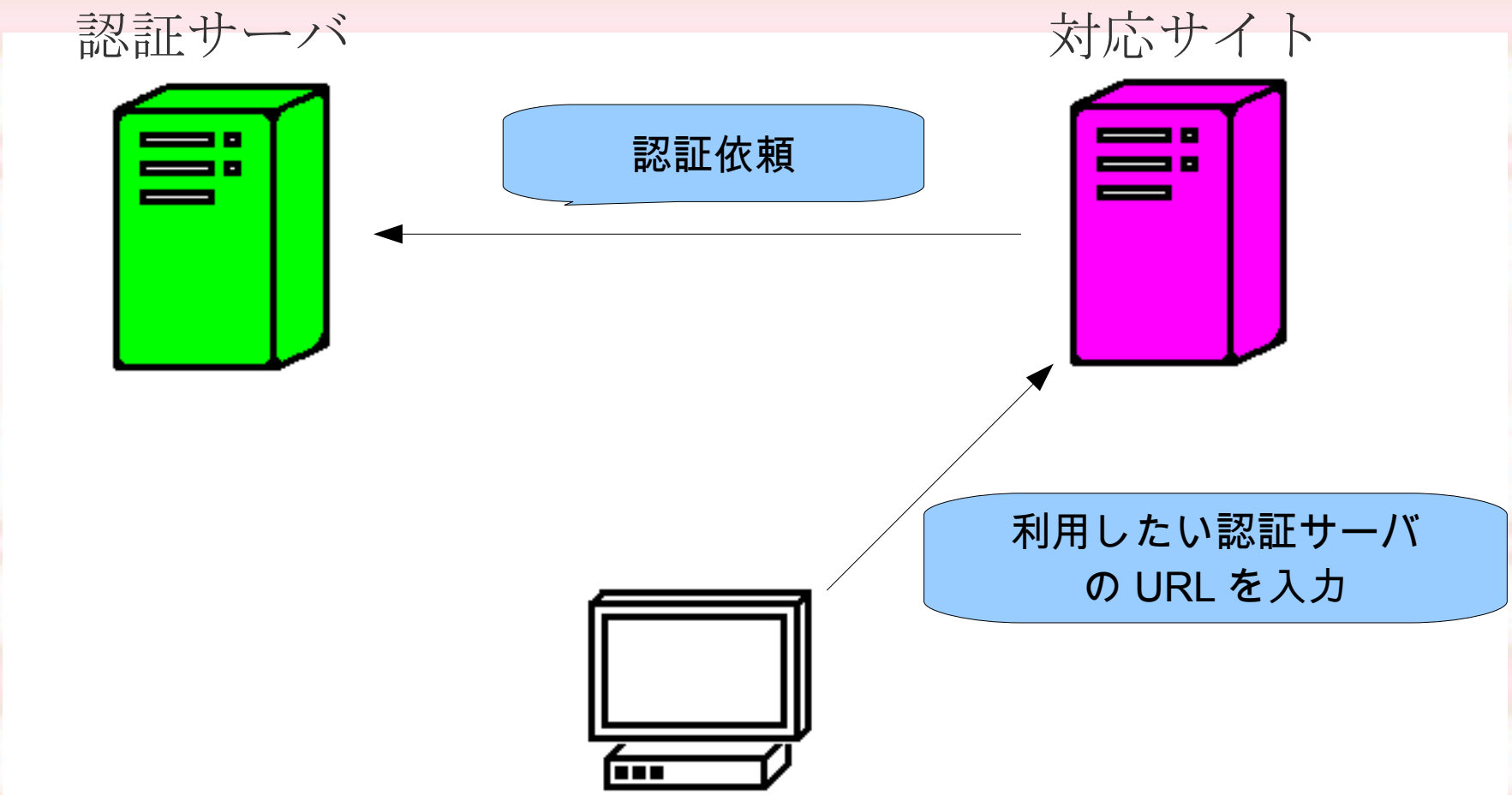
ID を取得

- ユーザ名
- パスワード

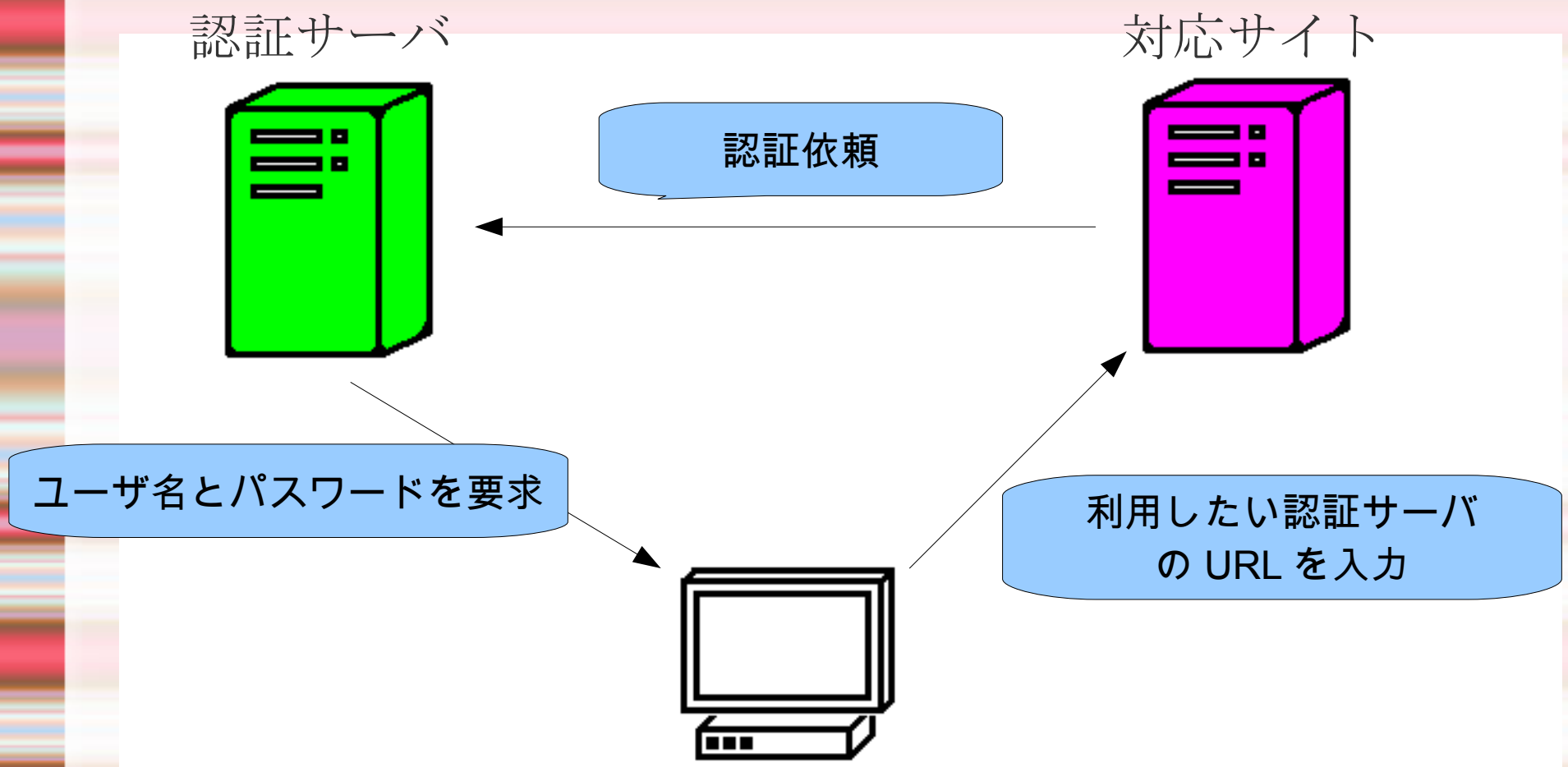
アクセスの流れ



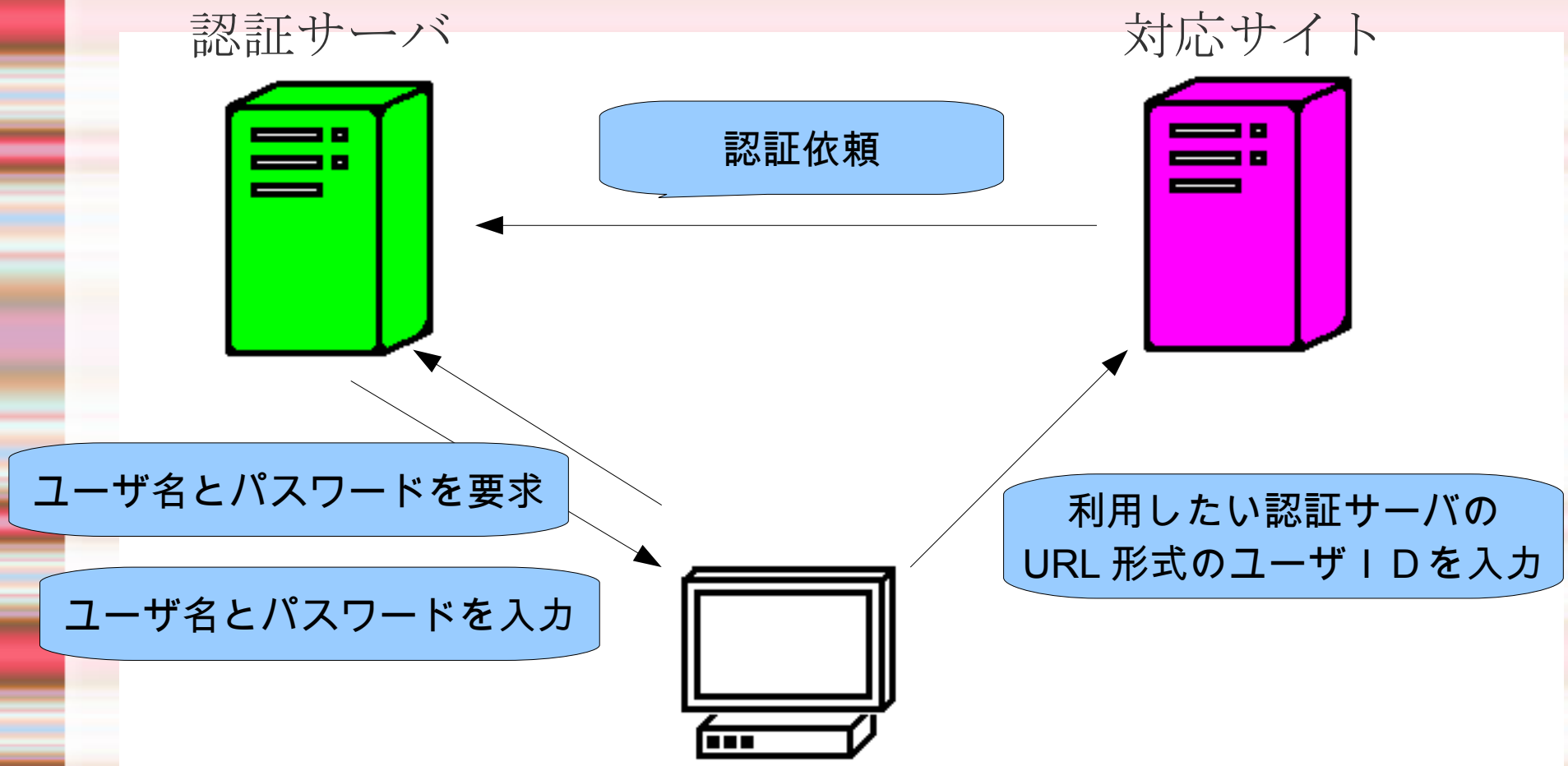
アクセスの流れ



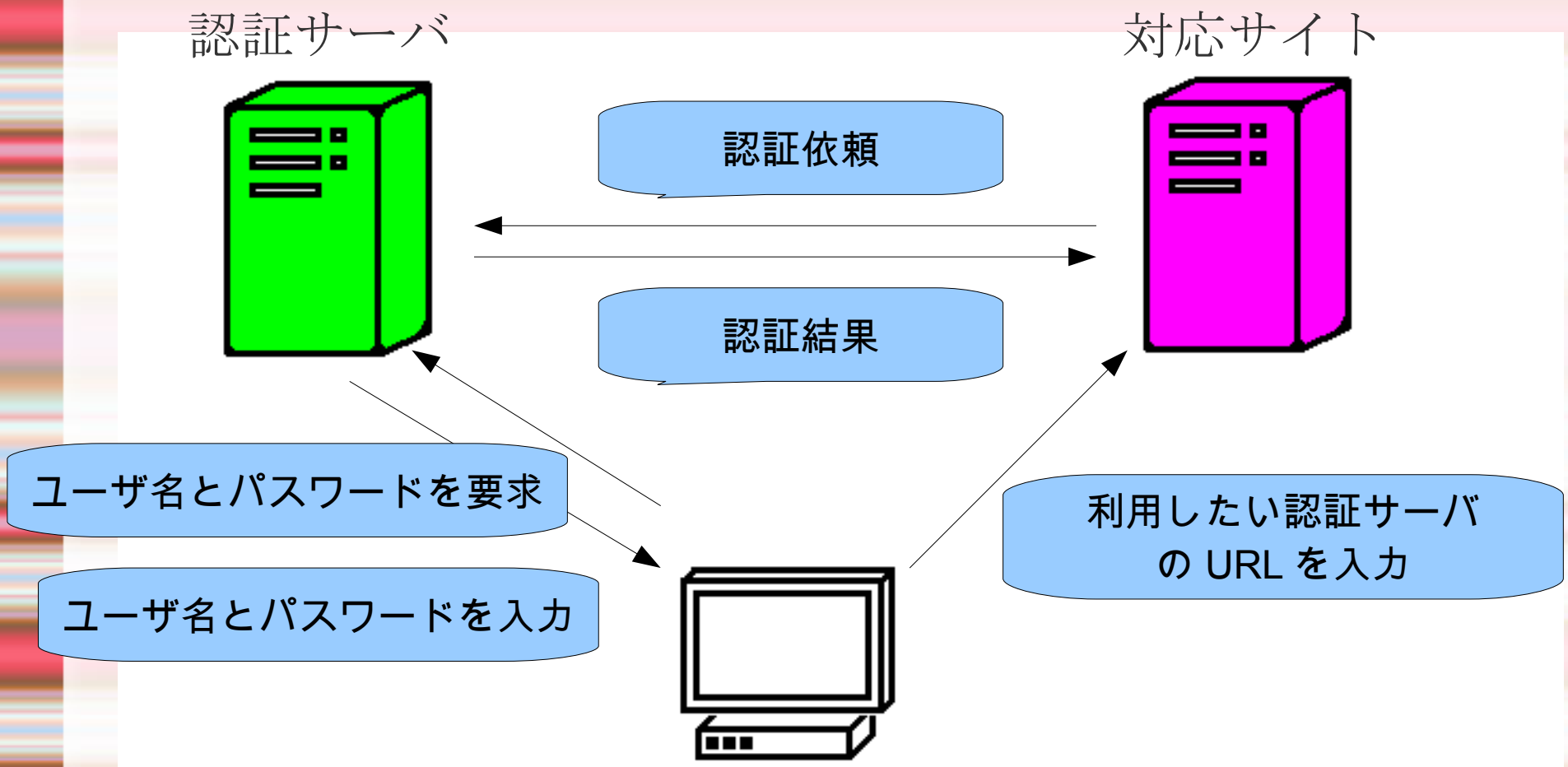
アクセスの流れ



アクセスの流れ


認証サーバ
対応サイト
認証依頼
ユーザ名とパスワードを要求
ユーザ名とパスワードを入力
利用したい認証サーバの
URL 形式のユーザＩＤを入力

アクセスの流れ


認証サーバ
対応サイト
認証依頼
認証結果
ユーザ名とパスワードを要求
ユーザ名とパスワードを入力
利用したい認証サーバ
の URL を入力

今後の予定

- ログイン機能の強化
  - OpenID または Shibboleth の導入
- rails から rails2or3 へ移行
- 公開範囲の設定方法の改良
  - グループ数増加への対応
  - Ajax の導入

現在の問題点

- 検印サーバを介して文書サーバにアクセスする際、ＩＤとパスワードを盗難される可能性がある

  →セキュリティを強化したい

- システム利用において、別のサーバにアクセスするたびに、ログインし直す必要がある

  →１回のログインで各機能を利用したい

シングルサインオン

- 一度ユーザ認証を行えば、複数のサーバでのユーザ認証をパスできる
- 複数サーバでの認証を１回の認証によって実現すること
- １回のログインで複数のシステムを利用できるようにすること

四重署名処理の流れ