茨城大学
Ibaraki University

Shibboleth 環境における認証強度を考慮した認証機能の実装

岡田泰伸・清水さや子・野口宏

茨城大学
Ibaraki University

背景

- これまでの認証方式
  - ID+パスワードや多要素認証
    - ユーザへの負担
    - 有効的な攻撃手段の存在

- Passkeysが注目されている
  - パスワードを使用しない認証方式
  - 様々なサービスで導入
  - 種別によって認証レベルに差異

背景

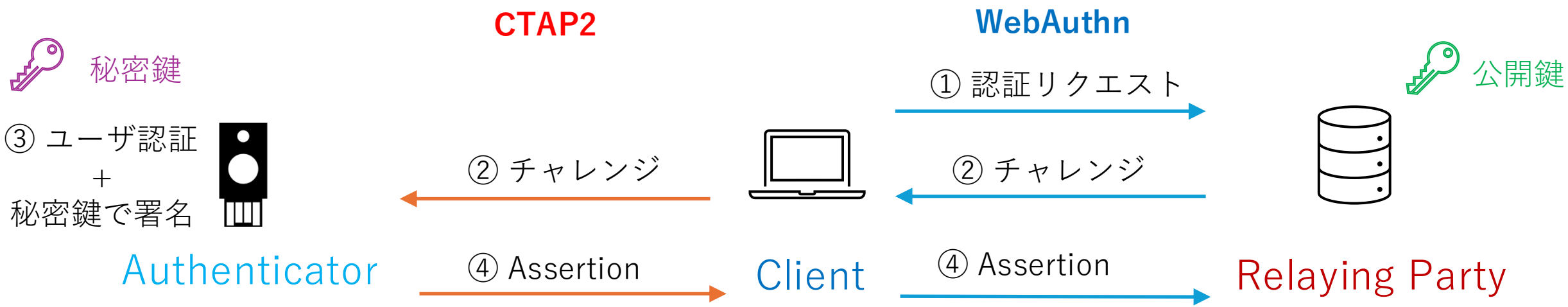
- 学術認証フェデレーション（学認）では対応が進められている

- **Shibboleth**
  2024年12月にIdP V5のプラグインとして
  WebAuthn Version 1.0.0がリリースされた
  Passkeysの利用はできるものの種別の判定はできない

FIDO2

- FIDO Alliance策定の公開鍵暗号を使用したパスワードレス認証の規格

- **WebAuthn**
  Webブラウザで利用可能なユーザ認証のためのAPI
- **CTAP2**(Client-to-Authenticator Protocol 2)
  Client Deviceと外部のAuthenticatorの通信のためのプロトコル



Passkeys

Passkeys：FIDO認証資格情報

- Passkeysの種別

Synced Passkeys

デバイス間で同期可能なPasskeys

クラウドで同期

利便性が高い

例) Google Password Manager,
iCloud Keychain,
1Password, Proton Pass

Device-bound Passkeys

デバイスに紐づくPasskeys

専用デバイスで管理

セキュリティ強度が高い

例) YubiKey, Titan Security Key,
Windows Hello

本人確認の強度

本人確認は
- 利用者の実在性を確認する**身元確認**
- 利用者の当人性を確認する**当人認証**

に分けられる

身元確認の保証レベル
IAL(Identity Assurance Level)

IAL3：	対面での身元確認
IAL2：	遠隔または対面での身元確認
IAL1：	身元確認のない自己表明

当人認証の保証レベル
AAL(Authentication Assurance Level)

AAL3：	複数の認証要素に加え 耐タンパ性が確保された ハードウェアトークン
AAL2：	複数の認証要素
AAL1：	単一または複数の認証要素

関連技術 - 本人確認の強度

本人確認は
- 利用者の実在性を確認する**身元確認**
- 利用者の当人性を確認する**当人認証**

に分けられる

身元確認の保証レベル
IAL(Identity Assurance Level)

IAL3：	対面での身元確認
IAL2：	遠隔または対面での身元確認
IAL1：	身元確認のない自己表明

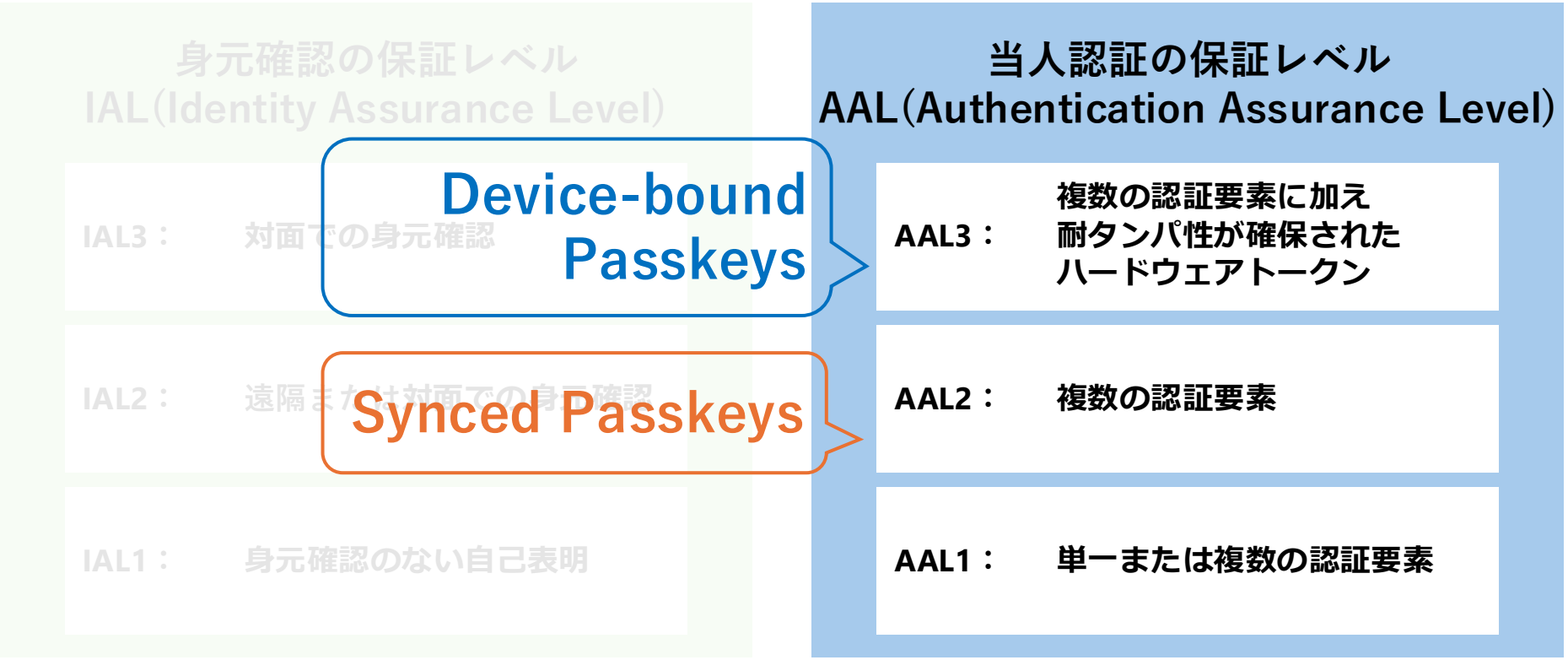
当人認証の保証レベル
AAL(Authentication Assurance Level)

AAL3：	複数の認証要素に加え 耐タンパ性が確保された ハードウェアトークン
AAL2：	複数の認証要素
AAL1：	単一または複数の認証要素

関連技術 - 本人確認の強度

本人確認は
- 利用者の実在性を確認する**身元確認**
- 利用者の当人性を確認する**当人認証**

に分けられる



現状

榊原らの研究（2025年）によってShibboleth IdP V5 で
Passkeys の種別による認証可否の判定が可能
AAL3の要件にも対応可

しかし
　　種別による認証可否の判定はIdPの設定に依存

SP主導で認証レベルを変化できないか

具体例：SP主導での認証レベルの変化

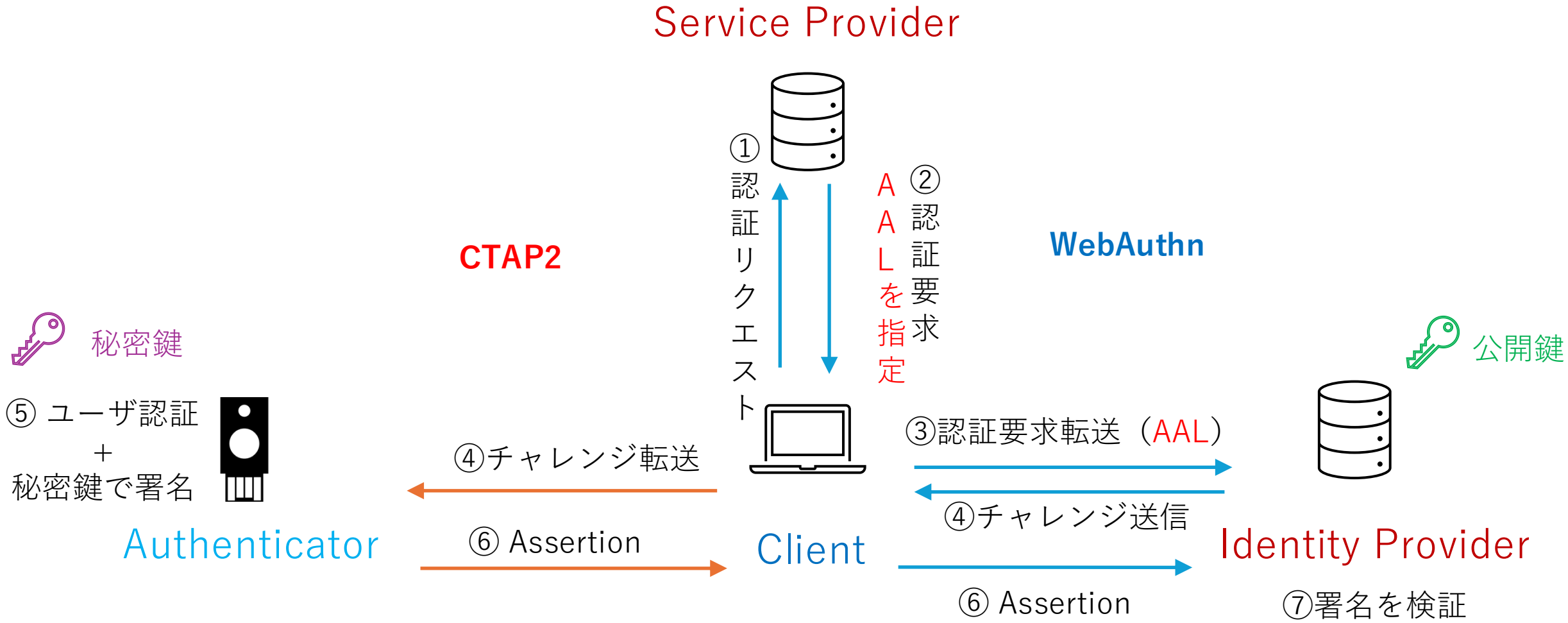
茨城大学 Ibaraki University

- 例：同一SPでも資源によって認証レベルが異なる
  - 一般ページ：AAL1で十分
  - 重要ページ：AAL2必須
  - 最重要：AAL3相当を要求したい
- 現状：IdP側の固定設定だと、SP主導で柔軟に切替しづらい

  → SPがAALを要求しIdPが解釈・実行する仕組みが必要

SP認証ポリシ設定方法

②認証要求作成の際にSPがAALを指定。
④指定されたAALに対応する認証方式を選択。
⑦送られてきたAssertionに含まれる認証機の種別が指定されたAALに対応する種別かを判断する



SP認証レベル設定方法

茨城大学 Ibaraki University

②認証要求作成の際にSPがAALを指定。

/etc/httpd/conf.d/shib.conf

```
<Location /AAL3>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  ShibRequestSetting authnContextClassRef https://www.gakunin.jp/profile/AAL3
  <RequireAll>
    Require shib-session
    Require authnContextClassRef https://www.gakunin.jp/profile/AAL3
  </RequireAll>
  ErrorDocument 401 /setup/stepup-aal3.php
</Location>
```

- Location単位でAALを指定（AAL2はhttps://www.gakunin.jp/profile/AAL2,AAL1は指定なし）
- 認証後は返却されたAuthnContextClassRefで要件充足を検証
- 不足時（401）に誘導ページへ遷移し再認証

IdPのフロー分岐と判定条件

④指定されたAALに対応する認証方式を選択。

フロー選択

- authnContextClassRefにAAL3が含まれる → authn/WebAuthn3
- AAL2が含まれる → authn/WebAuthn
- いずれも無い → 既定（パスワード認証）

認証可否

- WebAuthn3：PasskeysをDevice-boundに限定
- WebAuthn：許可範囲をany（より広く）

AAL → IdP動作

要求	フロー	許可条件
AAL3	authn/WebAuthn3	Device-boundのみ
AAL2	authn/WebAuthn	Synced/Device-bound（any）
指定なし	authn/Password	パスワード認証

認証器の情報の拡張

茨城大学 Ibaraki University

⑦送られてきたAssertionに含まれる認証機の種別が指定されたAALに対応する種別かを判断する

WebAuthnの仕様にはPasskeysの種別を直接返すパラメータがない

WebAuthnプラグインでは……

aaguid.jsonから認証器のメタデータを取得

aaguid.jsonに含まれる情報　name：認証器の名前

icon　：アイコン

aaguid.jsonを拡張

type：Passkeysの種別 を追加

それぞれの認証器の仕様に基づいてsynced / device-boundを設定

実装 - 使用した認証器

茨城大学 Ibaraki University

Name	Provider	Passkeys Type
YubiKey 5C	Yubico	**Device-bound**
Titan Security Key	Google	**Device-bound**
Windows Hello	Microsoft	**Device-bound**
Google Password Manager	Google	**Synced**
Proton Pass	Proton	**Synced**

認証器のメタデータ例 (aaguid.json)

```
"ea9b8d66-4d01-1d21-3ce4-b6b48cb575d4": {
        "name": "Google Password Manager",
        "icon_dark": "...",
        "icon_light": "...",
        "type": "synced"
}
```

実装 - 認証可否の判定

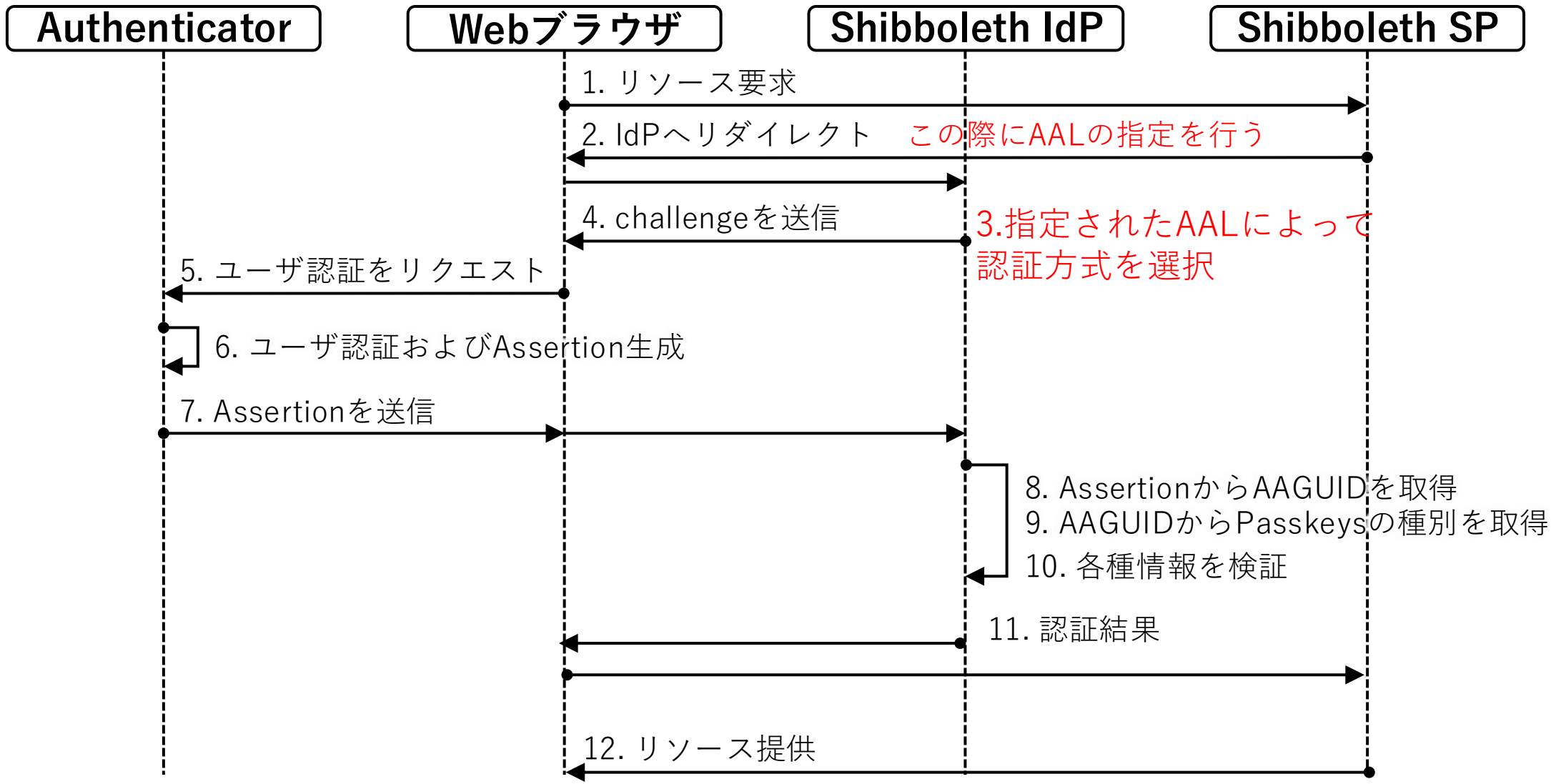
WebAuthnプラグインの機能を拡張
プロパティにidp.authn.webauthn.allowedKeyTypesを追加

idp.authn.webauthn.allowedKeyTypes

認証を許可するPasskeysの種別
synced / device-bound / any を設定可能

認証に使用された認証器の”type”と
”idp.authn.webauthn.allowedKeyTypes”の値が一致するとき認証を許可

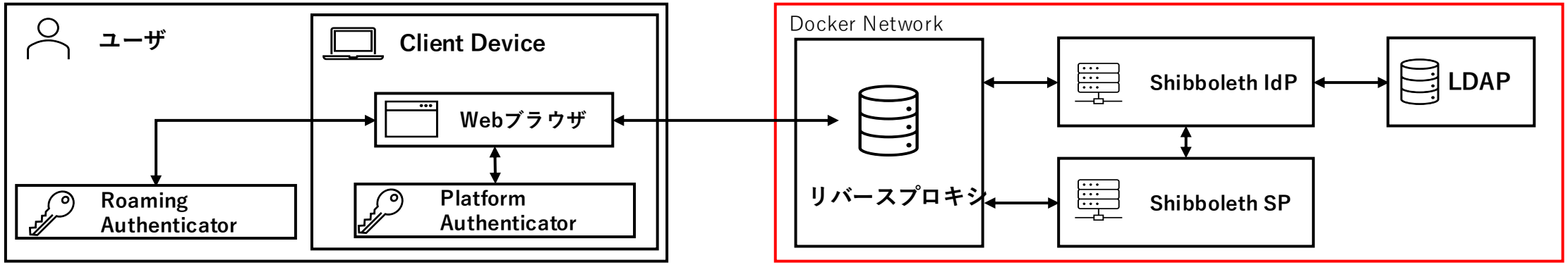
認証フロー


Authenticator
Webブラウザ
Shibboleth IdP
Shibboleth SP
1. リソース要求
2. IdPへリダイレクト　この際にAALの指定を行う
4. challengeを送信
3.指定されたAALによって
認証方式を選択
5. ユーザ認証をリクエスト
6. ユーザ認証およびAssertion生成
7. Assertionを送信
8. AssertionからAAGUIDを取得
9. AAGUIDからPasskeysの種別を取得
10. 各種情報を検証
11. 認証結果
12. リソース提供

システム構成

- Shibboleth IdP
  - IdP Version : Shibboleth IdP Version 5.1.4
  - Plugin : WebAuthn Version 1.0.0
- Shibboleth SP : Shibboleth SP Version 3.4.1
- LDAP : OpenLDAP 2.4.57

- Webブラウザ : Google Chrome Version 144.0.7559.110



確認できた挙動

①指定されたAALを満たさないPasskeysでは認証が拒絶

②AAL指定に応じてログイン画面／認証フロー切替

③ステップアップ認証
a.AAL2相当→AAL3相当へ遷移する際は再認証（ステップアップ認証）が発生
b.逆方向（AAL3→AAL2、AAL2→AAL1）では追加認証なし

① AAL3でSyncd passkeys利用による拒絶（AAL3に満たない）

opensaml::FatalProfileException

The system encountered an error at Tue Jan 13 07:39:11 2026

To report this problem, please contact the site administrator at root@localhost.

Please include the following message in any email:

opensaml::FatalProfileException at (https://nogshib-sp.cis.ibaraki.ac.jp/Shibboleth.sso/SAML2/POST)

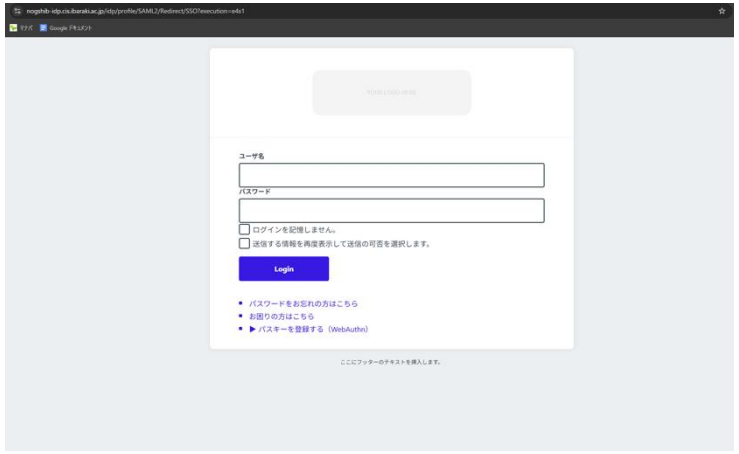
SAML response reported an IdP error.

Error from identity provider:

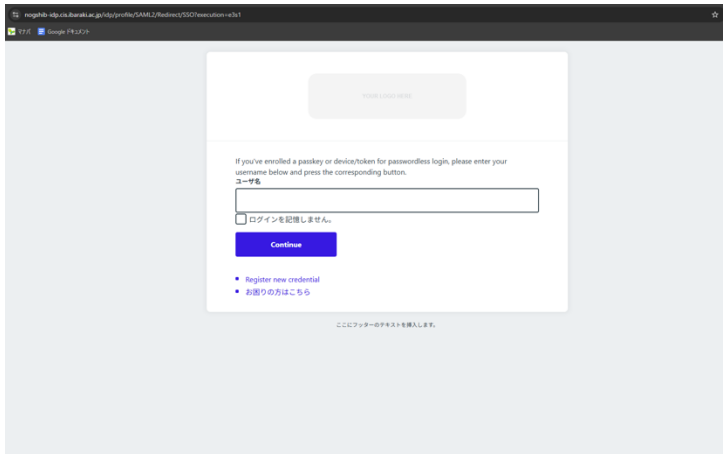
Status: urn:oasis:names:tc:SAML:2.0:status:Requester
Sub-Status: urn:oasis:names:tc:SAML:2.0:status:AuthnFailed
Message: An error occurred.

②認証フローの変化

AAL1

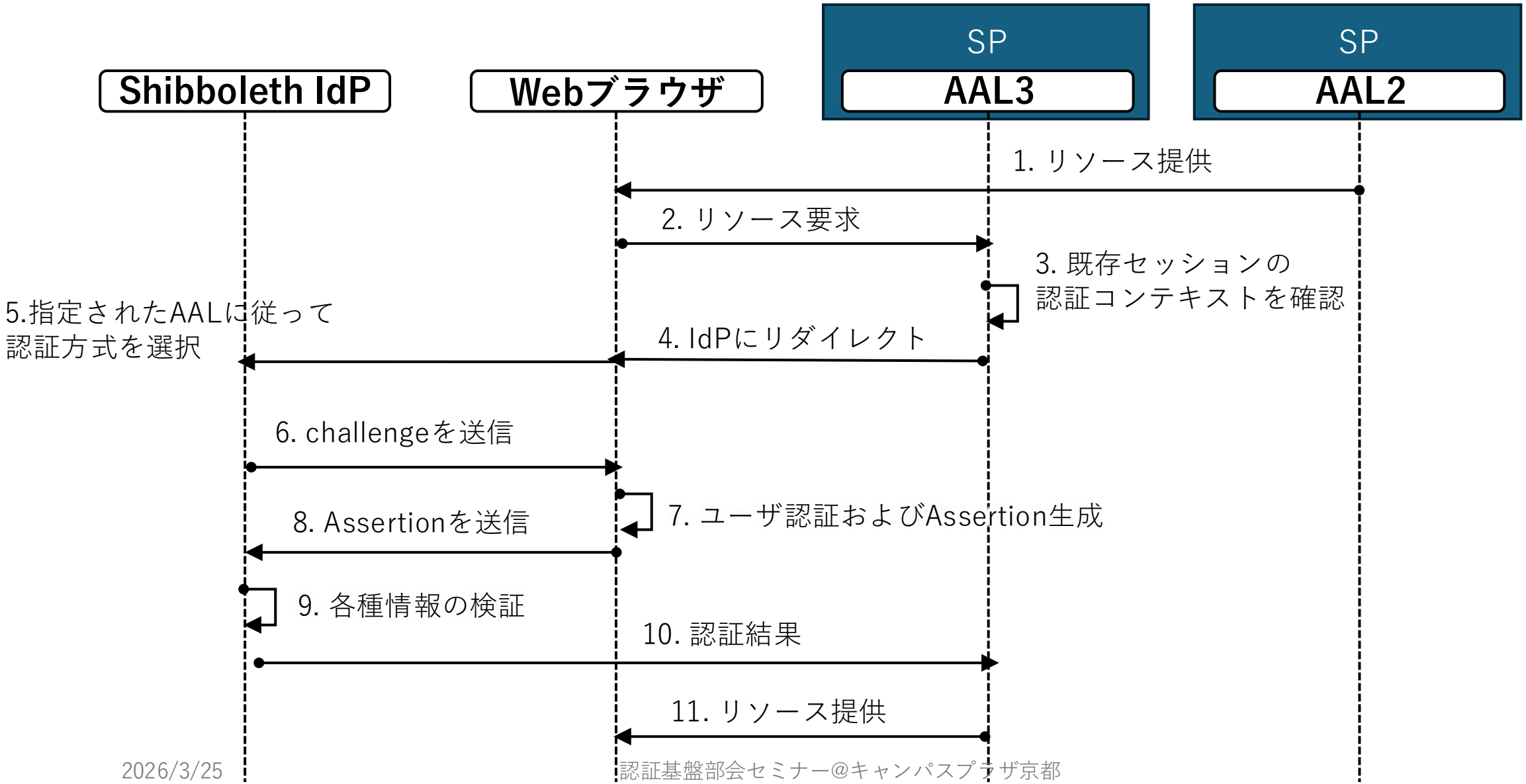


AAL 2



③ ステップアップ認証

a.AAL2相当→AAL3相当へ遷移する際は再認証（ステップアップ認証）が発生
b.逆方向（AAL3→AAL2、AAL2→AAL1）では追加認証なし



まとめ

茨城大学
Ibaraki University

- SPがAuthnContextClassRefでAALを指定し、IdPがそれに応じて認証フロー選択を実施
- AAL3相当ではdevice-bound Passkeysのみ許可し、厳格な資源アクセス制御を実現
- 不足時は再認証を誘導し、段階的認証を実現
- 既存の運用を前提に、サービス要件に応じた認証強度制御を導入可能

今後の課題

- Webauthn v1.4.0への更新と適用
- より多様な運用条件で、Passkeys種別と認証レベルの対応が妥当か継続検証
- 提案構成を基盤に、より実運用に近い環境へ適用

NIIとの共同研究[252S1-23665]
この研究は2026年度国立情報学研究所公募型共同研究(252S1-23665)の助成を受けています。