“車が通れるほど大きな”
セキュリティーホールを抑えながら
ログインしたい

Kyoto Tech Talk #8

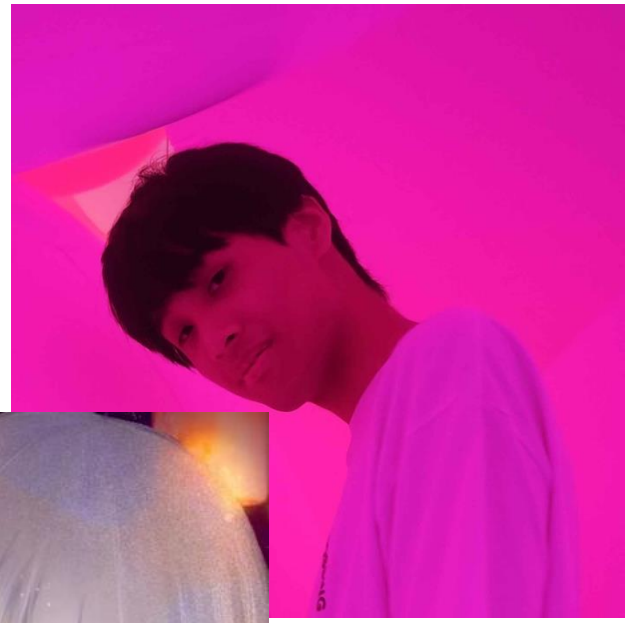
2025/5/29

京都産業大学 上村太成

WSOFT

自己紹介

- taiseiue
  - X: @taiseiue
  - taiseiue.jp
  - C#/PHP/JavaScript
  - バックエンド/プログラミング言語
  - おひとり様サークル => WSOFT
    - wsoft.ws



WSOFT

導入

- 最近スタジオを手伝ってます

スタジオふみ

導入

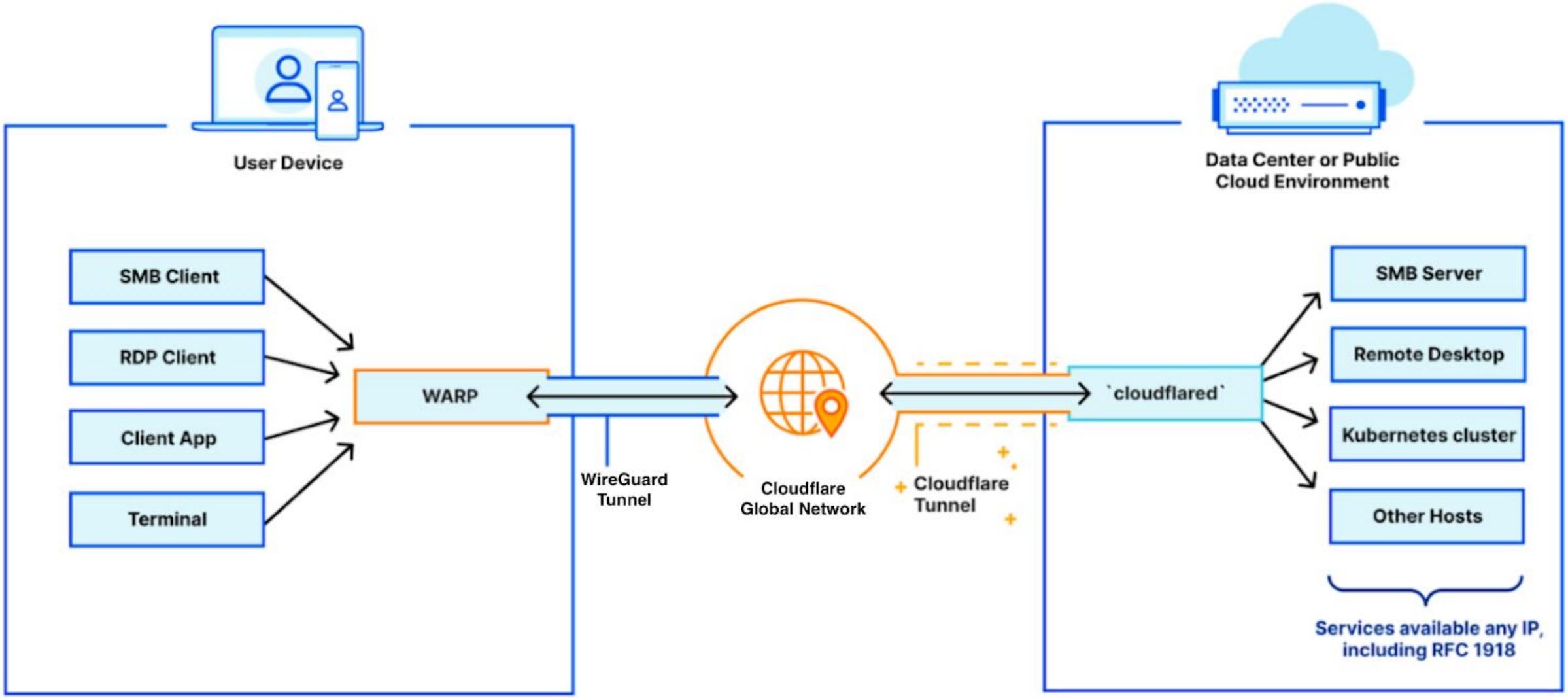
- スタジオふみで使うアプリに認証をかけたい
  - 備品管理アプリ
  - メディアサーバー
  - ファイルサーバー
  - Minecraftサーバー

導入

- アプリごとの認証を使う
  - 備品管理アプリ　→ NextAuth
  - メディアサーバー　→ サーバー付属のものを使う
  - ファイルサーバー　→ Active Directory
  - Minecraftサーバー → すべてを受け入れる

管理がめんどう

Cloudflare ZeroTrust


User Device
SMB Client
RDP Client
Client App
Terminal
WARP
WireGuard Tunnel
Cloudflare Global Network
Cloudflare Tunnel
`cloudflared`
Data Center or Public Cloud Environment
SMB Server
Remote Desktop
Kubernetes cluster
Other Hosts
Services available any IP, including RFC 1918

認証

- Cloudflare Zero Trust
  - 特定のアプリやトラフィックに認証を掛けれる
  - だけじゃなくて、HTTPならリクエストヘッダにおまけがついてくる

  - Cf-Access-Jwt-AssertionヘッダのをAPIに投げるとユーザー名やメールアドレスが返ってくる

  - SSO的なことができる

ログイン方法を考える

Wstsoft.info@gmail...

Zero Trust の概要
Analytics
Gateway
Access
ネットワーク
マイ チーム
ログ
CASB
DLP
DEX
Email Security 新規
ブラウザ分離 新規
設定

← 認証に戻る

ログイン方法を追加する

ID プロバイダーを選択

- Azure AD
- Centrify
- Facebook
- GitHub
- Google Workspace
- Google
- LinkedIn
- Okta
- OneLogin
- One-time PIN
- OpenID Connect
- PingOne
- SAML
- Yandex

Wstsoft.info@gmail...

Zero Trust の概要

Analytics

Gateway

Access

ネットワーク

マイ チーム

ログ

CASB

DLP

DEX

Email Security 新規

ブラウザ分離 新規

設定

← 認証に戻る

ログイン方法を追加する

ID プロバイダーを選択

- Azure AD
- Centrify
- GitHub
- LinkedIn
- Okta
- OneLogin
- One-time PIN
- OpenID Connect
- PingOne
- SAML
- Yandex

Wstsoft.info@gmail...

Zero Trust の概要

Analytics

Gateway

Access

ネットワーク

マイ チーム

ログ

CASB

DLP

DEX

Email Security 新規

ブラウザ分離 新規

設定

← 認証に戻る

ログイン方法を追加する

ID プロバイダーを選択

Azure AD

Centrify

Google

LinkedIn

Okta

OneLogin

One-time PIN

OpenID Connect

PingOne

SAML

Yandex

Googleでログインする

Google にログイン

アカウントの選択

「cloudflareaccess.com」に移動

上村太成

上村太成

別のアカウントを使用

このアプリを使用する前に、cloudflareaccess.com のプライバシー ポリシーと利用規約をご確認ください。

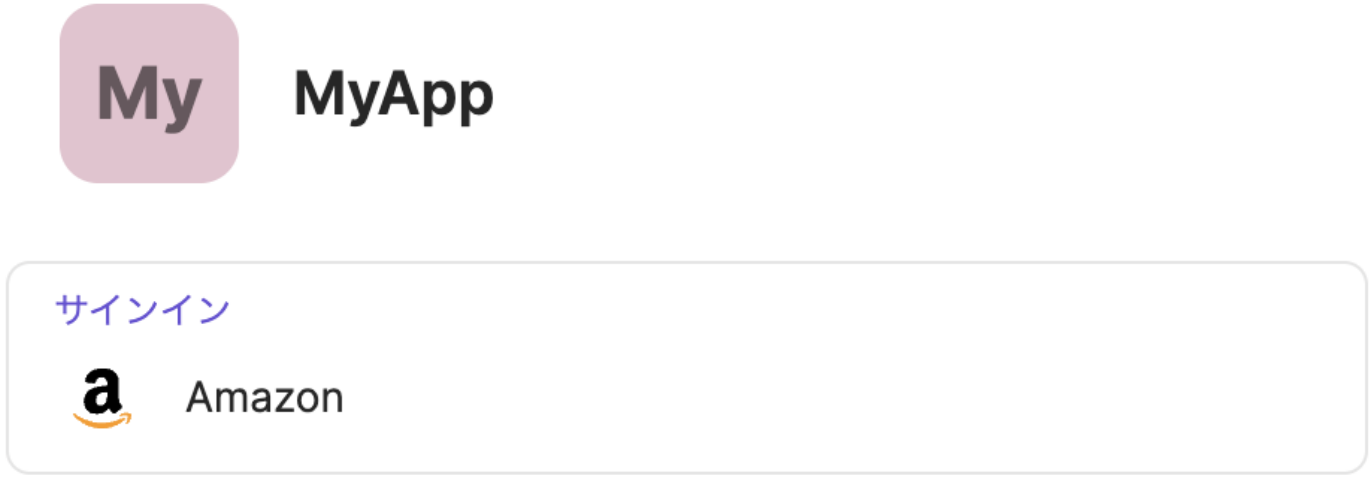
上村太成

上村太成

別のアカウントを使用

Googleでログインの問題点

- (どういうわけか)一部の人はGoogleアカウントを複数持っている
- 何にどれをつかっているか覚えていない



1Passwordにもこんな機能が

何のコミュニティ用に使ってるか明確なIdP？

Discordやん

Applications
Teams
Server Insights
Embed Debugger
Documentation

Search ⌘ K

Intro
Change Log
API Reference

QUICK START
Overview of Apps
Getting Started

INTERACTIONS
Overview
Receiving and Responding
Application Commands

COMPONENTS

OAuth2

OAuth2 enables application developers to build applications that utilize authentication and data from the Discord API. Within Discord, there are multiple types of OAuth2 authentication. We support the authorization code grant, the implicit grant, client credentials, and some modified special-for-Discord flows for Bots and Webhooks.

Shared Resources

The first step in implementing OAuth2 is registering a developer application and retrieving your client ID and client secret. Most people who will be implementing OAuth2 will want to find and utilize a library in the language of their choice. For those implementing OAuth2 from scratch, please see RFC 6749 for details. After you create your application with Discord, make sure that you have your `client_id` and `client_secret` handy. The next step is to figure out which OAuth2 flow is right for your purposes.

OAuth2 URLs

URL	DESCRIPTION
https://discord.com/oauth2/authorize	Base authorization URL
https://discord.com/api/oauth2/token	Token URL
https://discord.com/api/oauth2/token/revoke	Token Revocation URL

> ⚠ In accordance with the relevant RFCs, the token and token revocation URLs will **only** accept a content type of `application/x-www-form-urlencoded`. JSON content is not permitted and will return an error.

OAuth2.0で認証？

IDENTITY OPENID セキュリティ プライバシー

単なる OAuth 2.0 を認証に使うと、車が通れるほどのどでかいセキュリティー・ホールができる

2012/02/02 17 Comments oauth OpenID Connect security hole web セキュリティ・ホール

OAuth 2.0 の **implicit grant flow** を認証に使うと、車が通れる程どてかいセキュリティ・ホールが開くよ、と言う、ジョン・ブラッドレー氏[1]による良記事。コメントも読み応えあります。ちょっとチェックした見たところは、全滅。RP側を治さなきゃいけないから、とっとと公開アナウンスしたほうが良いのでしょうね。いちいちコンタクトしてられないし。
Facebook や、その他OAuthログインしているサイトはみんなチェク！

The problem with OAuth for Authentication. | Thread Safe

In some of the feedback I have gotten on the openID Connect spec, the statement is made that Connect is too complicated. That OAuth 2.0 is all you need to do authentication. Many point to Identity Pro…

https://www.sakimura.org/2012/02/1487/

IDENTITY OPENID セキュリティ プライバシー

単なる OAuth 2.0 を認証に使うと、車が通れるほどのどでかいセキュリティー・ホールができる

2012/02/02 17 Comments oauth OpenID Connect security hole web セキュリティ・ホール

2.2k SHARES

OAuth 2.0 の **implicit grant flow** を認証に使うと、車が通れる程どてかいセキュリティ・ホールが開くよ、と言う、ジョン・ブラッドレー氏[1]による良記事。コメントも読み応えあります。ちょっとチェックした見たところは、全滅。RP側を治さなきゃいけないから、とっとと公開アナウンスしたほうが良いのでしょうね。いちいちコンタクトしてられないし。
Facebook や、その他OAuthログインしているサイトはみんなチェク！

The problem with OAuth for Authentication. | Thread Safe

In some of the feedback I have go
Connect is too complicated. That OAuth 2.0 is all you need to do authentication. Many point to Identity Pro…

OAuth2→OIDCを使う

https://www.sakimura.org/2012/02/1487/

Wstsoft.info@gmail...
Zero Trust の概要
Analytics
Gateway
Access
ネットワーク
マイ チーム
ログ
CASB
DLP
DEX
Email Security 新規
ブラウザ分離 新規
設定

← 認証に戻る

ログイン方法を追加する

ID プロバイダーを選択

- Azure AD
- Centrify
- Facebook
- GitHub
- Google Workspace
- Google
- LinkedIn
- Okta
- OneLogin
- One-time PIN
- OpenID Connect
- PingOne
- SAML
- Yandex

OpenID Connect (OIDC)

- OAuth2の拡張
- OAuth2は「認可」に対してOIDCは「認証」が目的

- OAuth2のアクセストークンに加えてIDトークンを持つ

OpenID Connect (OIDC)

- OAuth2の拡張
- OAuth2は「認可」に対してOIDCは「認証」が目的

- OAuth2のアクセストークンに加えてIDトークンを持つ

OAuth2のプロキシ的なものを作り、
それが追加でIDトークンを返せばいい

DiscordのOAuth2.0をOIDCにプロキシする

discord-oidc-proxy

Public

Pin Watch 0 Fork 0 Star 0

main 1 Branch 0 Tags Go to file t Add file Code

taiseiue 簡単に初期設定できるようにした 3a31d01 · 1 minute ago 7 Commits

.vscode	Initial commit (by create-cloudflare CLI)	4 days ago
keys	deploy.shかいた	5 hours ago
src	package.jsonを書いた	5 hours ago
test	Initial commit (by create-cloudflare CLI)	4 days ago
.editorconfig	Initial commit (by create-cloudflare CLI)	4 days ago
.gitignore	deploy.shかいた	5 hours ago
.prettierrc	Initial commit (by create-cloudflare CLI)	4 days ago
LICENSE	Initial commit	4 days ago
README.md	簡単に初期設定できるようにした	1 minute ago
package.json	簡単に初期設定できるようにした	1 minute ago
pnpm-lock.yaml	いかにも今ドキな感じになった	4 days ago
pnpm-workspace.yaml	いかにも今ドキな感じになった	4 days ago

About

The proxy that enables applications using OpenID Connect (OIDC) authentication to authenticate users through Discord's OAuth2 system.

discord oidc-proxy

Readme
MIT license
Activity
0 stars
0 watching
0 forks

Languages

TypeScript 97.1% Shell 2.9%

Suggested workflows

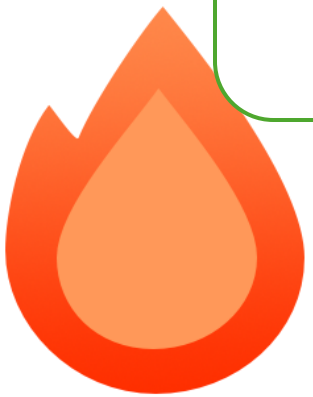
Based on your tech stack

onfigure

https://github.com/taiseiue/discord-oidc-proxy

CLOUDFLARE®
Workers

“いかにも今ドキ”を実現

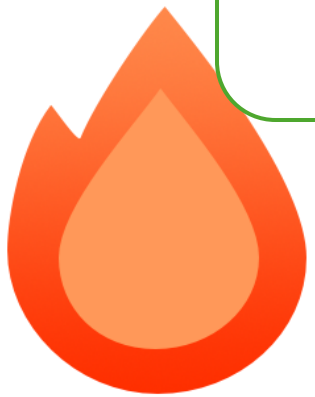


Hono

pnpm

CLOUDFLARE®
Workers

./setup.sh && pnpx publish



Hono

使い方

ためすくん

名前 (必須)
ログイン ページでこの ID プロバイダーを識別するための一意の名前。

ためすくん

アプリ ID (必須)

1377495552765198376

クライアント シークレット (必須)

認証 URL (必須)

https://tamesukun.wsoft.workers.dev/a

トークン URL (必須)

https://tamesukun.wsoft.workers.dev/tc

証明書 URL (必須)

https://tamesukun.wsoft.workers.dev/.v

コード交換のための証明キー (PKCE)

オン

PKCE は、クロスサイト リクエスト フォージェリ (CSRF) 攻撃や認証コード インジェクション攻撃を防ぐ認証コード フローの拡張機能です。ID プロバイダーが機密クライアントの PKCE を

ルールを追加する

ルール タイプ、セレクタ、セレクタ値を使用してポリシーのスコープを定義します。セレクタは、ユーザーに満たしてもらいたい基準または属性のタイプです。リスト、ログイン方法、デバイスのポスチャ チェックを追加して、追加のセレクタを構成します。 セレクタのドキュメント

包含 OR

複数の Include ルールが設定されている場合、ユーザーは条件の 1 つを満たすだけで済みます。

セレクター	クレーム名	請求額	
OIDC Claims - ためすくん	name	taiseiue	×

要求 AND

ユーザーがアクセスを許可されるには、すべての Require ルールを満たす必要があります。

セレクター	値	
Country	Japan ×	×

セレクター	値	
Login Methods	OpenID Connect・ためすくん ×	×

作り方

Draft	N. Sakimura
	NRI
	J. Bradley
	Ping Identity
	M. Jones
	Microsoft
	B. de Medeiros
	Google
	C. Mortimore
	Salesforce
	August 3, 2015

TOC

OpenID Connect Implicit Client Implementer's Guide 1.0 - draft 20

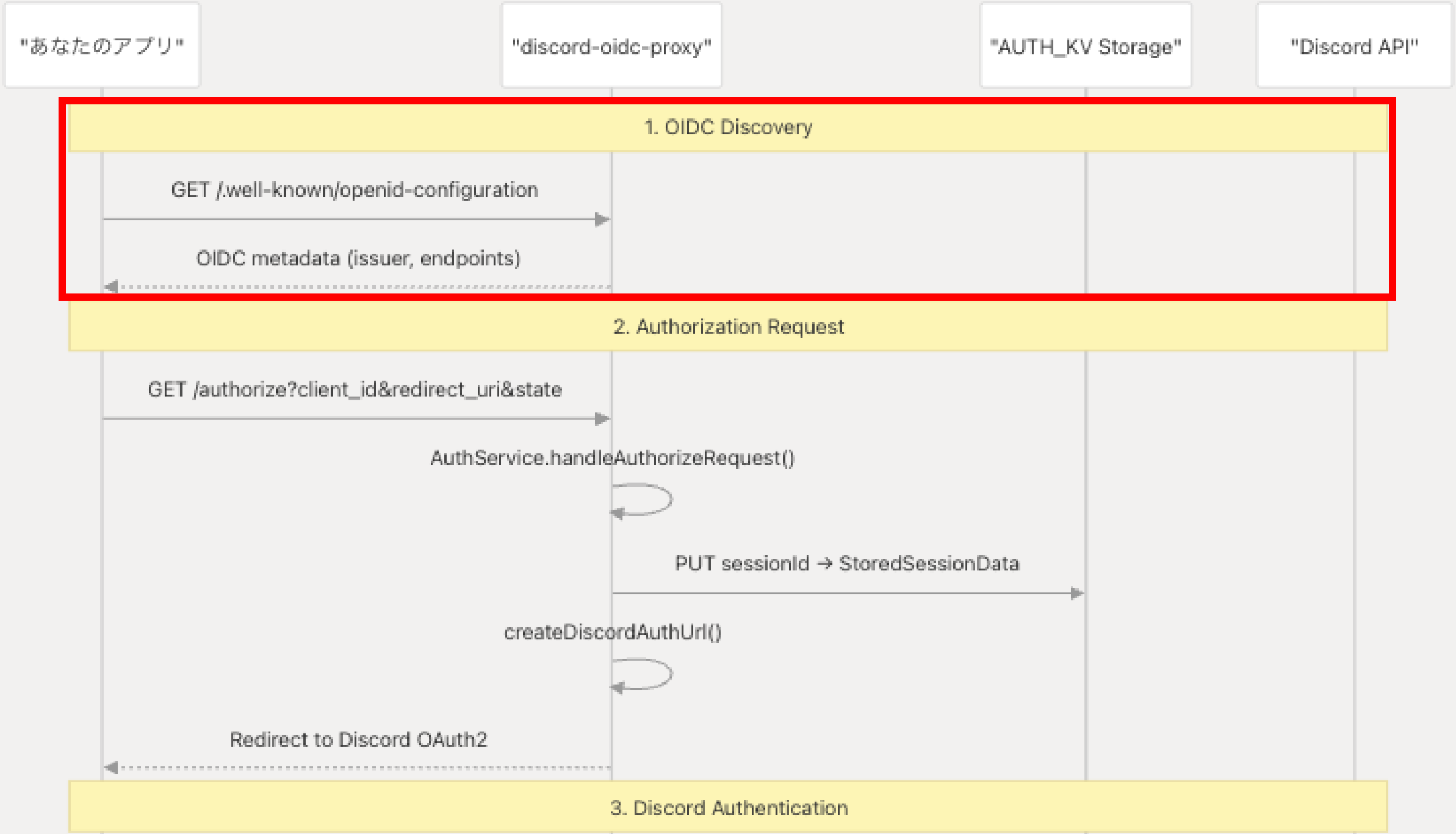
Abstract

OpenID Connect 1.0 は, OAuth 2.0 プロトコルの上にシンプルなアイデンティティレイヤーを付与したものである. このプロトコルは, Client が Authorization Server の認証結果に基づいて End-User のアイデンティティを検証することを可能にする. また同時に End-User の必要最低限のプロフィール情報を, 相互運用可能かつ RESTful な形で取得することも可能にする.

本ドキュメント OpenID Connect Implicit Client Implementer's Guide 1.0 は, OpenID Connect Core 1.0 仕様のサブセットであり, OAuth 2.0 Implicit Flowを利用して Webベースの Relying Party を実装する際に読みやすいように構成されている. 本ドキュメントは Core 仕様と重複したコンテンツを含んでいるが, それは実装者が本仕様を読むだけで OAuth Implicit Flow を利用した Relying Party を実装できるよう意図したものである.

OpenID Provider および Web ベース以外のアプリケーションの実装者は, Core 仕様を参照のこと.

Table of Contents


"あなたのアプリ"
"discord-oidc-proxy"
"AUTH_KV Storage"
"Discord API"
1. OIDC Discovery
GET /.well-known/openid-configuration
OIDC metadata (issuer, endpoints)
2. Authorization Request
GET /authorize?client_id&redirect_uri&state
AuthService.handleAuthorizeRequest()
PUT sessionId → StoredSessionData
createDiscordAuthUrl()
Redirect to Discord OAuth2
3. Discord Authentication

OIDCディスカバリ

属性のリクエストメソッド | 情報

リクエストメソッドは、プロバイダーアプリケーションで設定されます。

- ○ GET
- ◉ POST

セットアップ方法 | 情報

◉ 発行者 URL を通じた自動入力

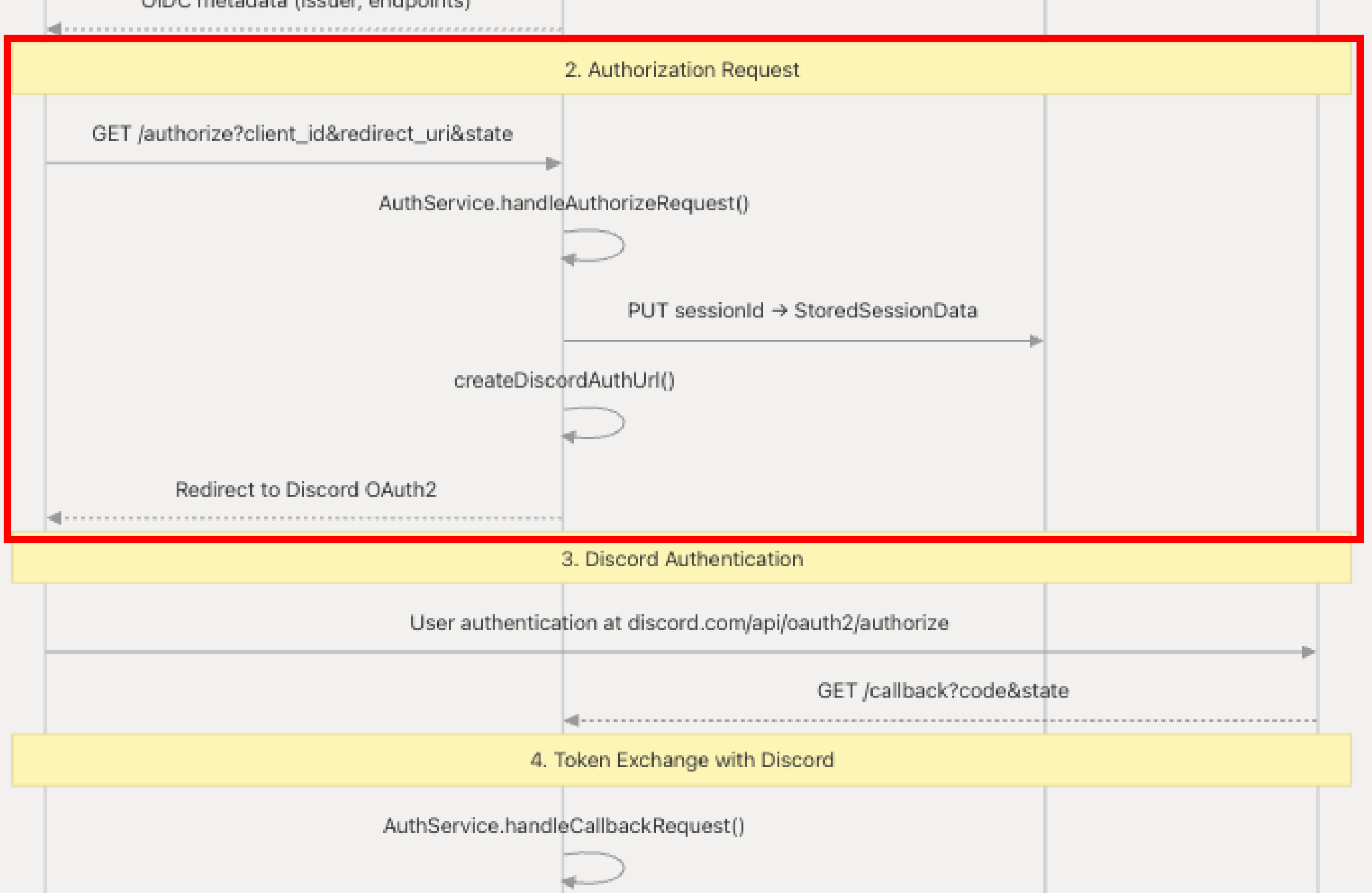
発行者 URL を指定すると、Cognito によって認証エンドポイント、トークンエンドポイント、userInfo エンドポイント、および jwks_uri の値が設定されます。

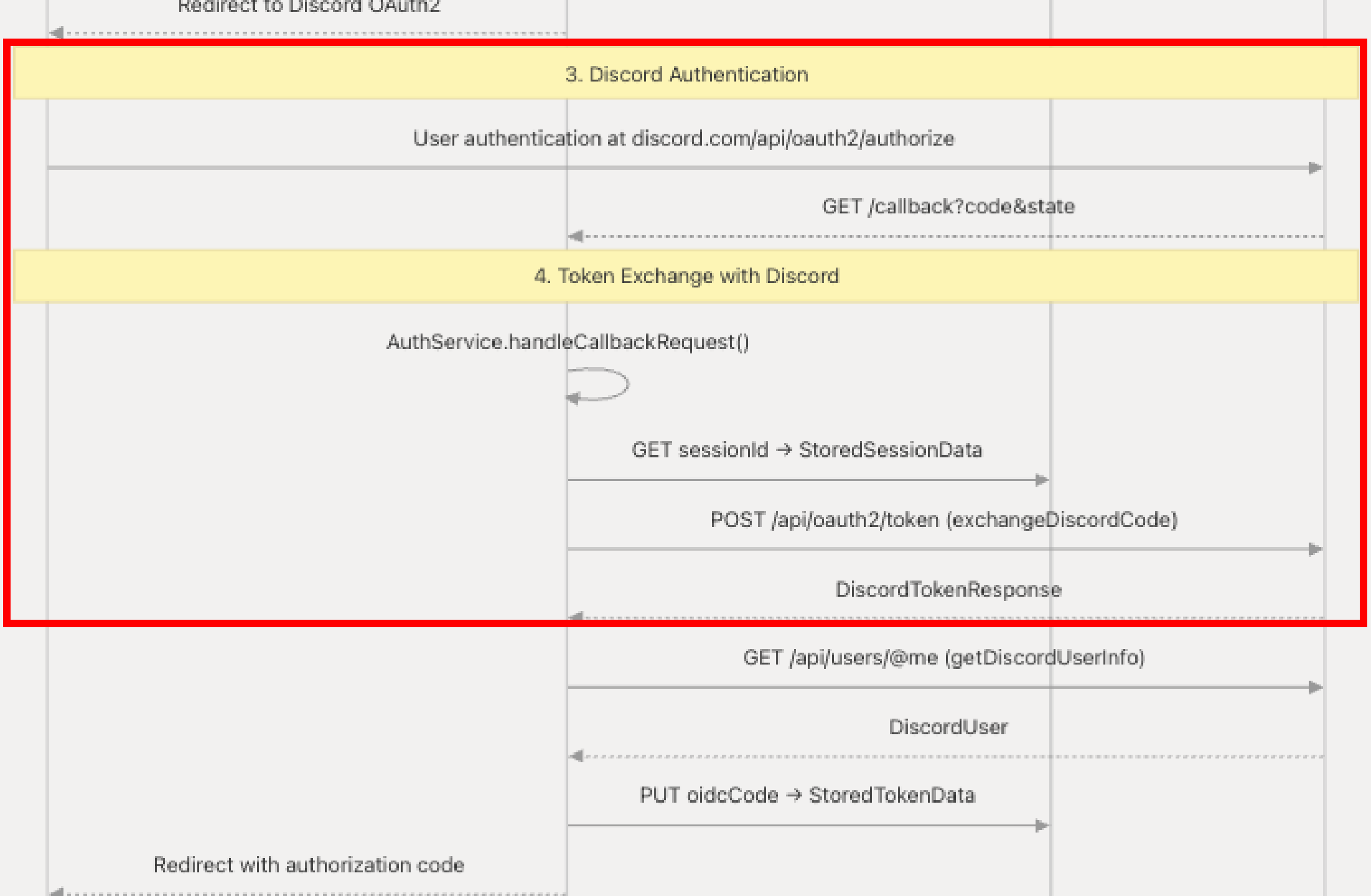
発行者 URL | 情報

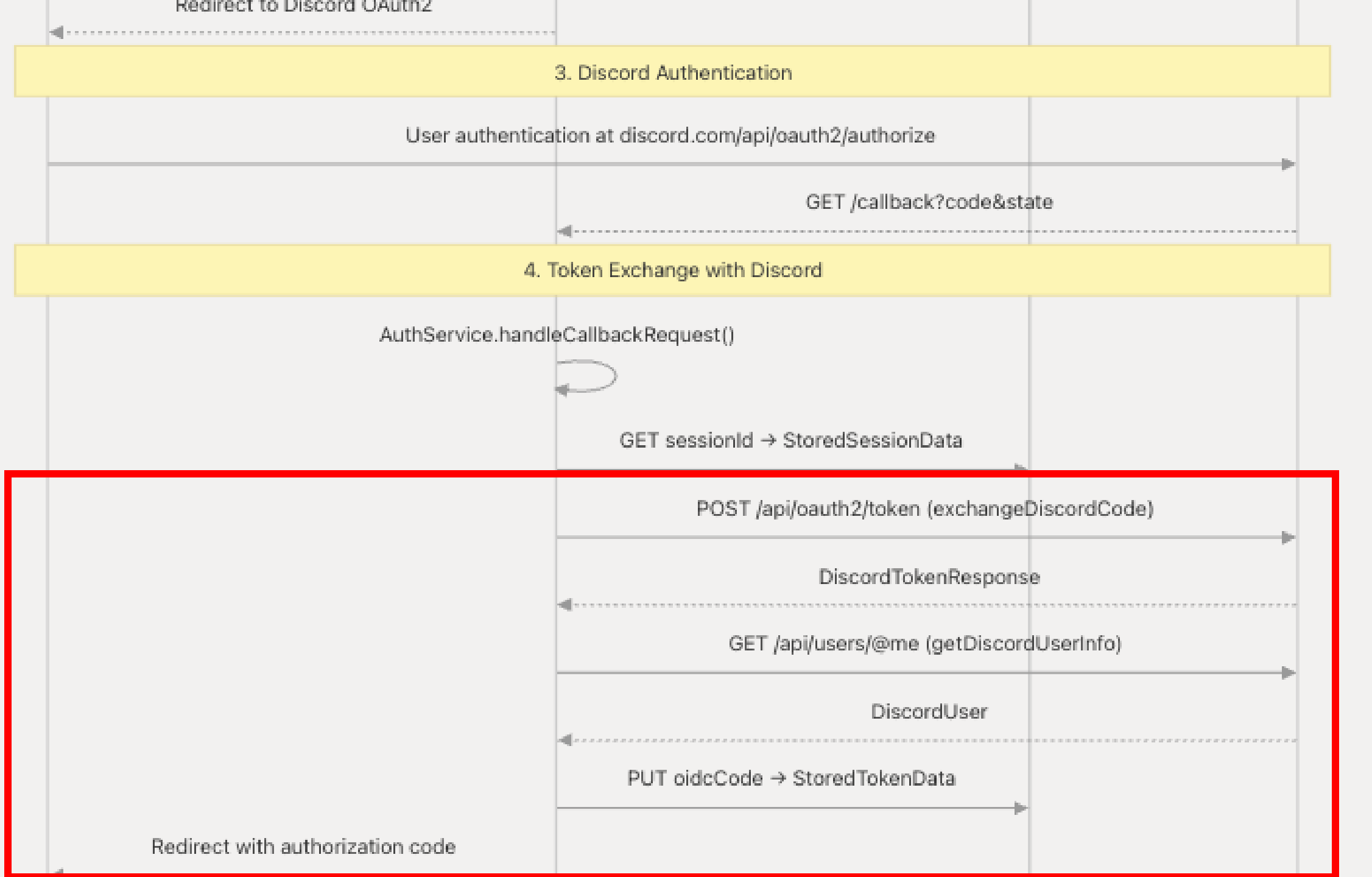
OIDC プロバイダーから受け取った発行者 URL を入力します。

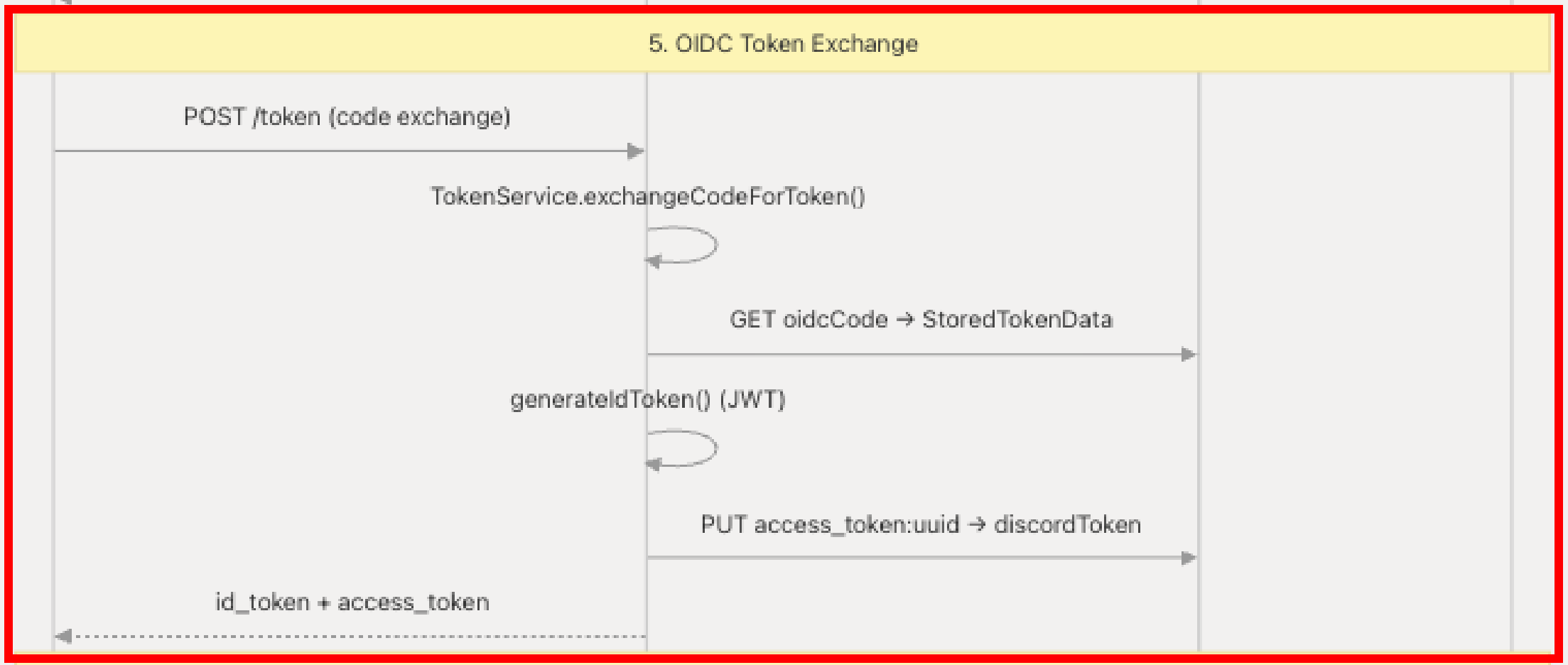
https://discord-oidc-proxy.wsoft.workers.dev

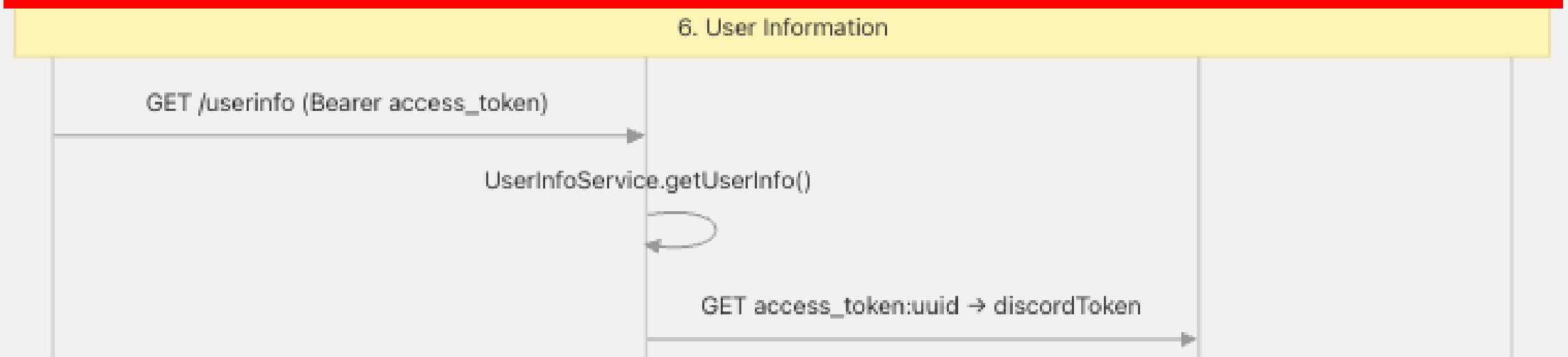
```
{
  "issuer": "https://discord-oidc-proxy.wsoft.workers.dev",
  "authorization_endpoint": "https://discord-oidc-proxy.wsoft.workers.dev/authorize",
  "token_endpoint": "https://discord-oidc-proxy.wsoft.workers.dev/token",
  "jwks_uri": "https://discord-oidc-proxy.wsoft.workers.dev/.well-known/jwks.json",
  "userinfo_endpoint": "https://discord-oidc-proxy.wsoft.workers.dev/userinfo",
  "response_types_supported": [
    "code"
  ],
  "subject_types_supported": [
    "public"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "scopes_supported": [
    "openid",
    "profile",
    "email"
  ],
  "token_endpoint_auth_methods_supported": [
    "client_secret_post"
```


OIDC metadata (issuer, endpoints)
2. Authorization Request
GET /authorize?client_id&redirect_uri&state
AuthService.handleAuthorizeRequest()
PUT sessionId → StoredSessionData
createDiscordAuthUrl()
Redirect to Discord OAuth2
3. Discord Authentication
User authentication at discord.com/api/oauth2/authorize
GET /callback?code&state
4. Token Exchange with Discord
AuthService.handleCallbackRequest()


Redirect to Discord OAuth2
3. Discord Authentication
User authentication at discord.com/api/oauth2/authorize
GET /callback?code&state
4. Token Exchange with Discord
AuthService.handleCallbackRequest()
GET sessionId → StoredSessionData
POST /api/oauth2/token (exchangeDiscordCode)
DiscordTokenResponse
GET /api/users/@me (getDiscordUserInfo)
DiscordUser
PUT oidcCode → StoredTokenData
Redirect with authorization code


Redirect to Discord OAuth2
3. Discord Authentication
User authentication at discord.com/api/oauth2/authorize
GET /callback?code&state
4. Token Exchange with Discord
AuthService.handleCallbackRequest()
GET sessionId → StoredSessionData
POST /api/oauth2/token (exchangeDiscordCode)
DiscordTokenResponse
GET /api/users/@me (getDiscordUserInfo)
DiscordUser
PUT oidcCode → StoredTokenData
Redirect with authorization code


5. OIDC Token Exchange
POST /token (code exchange)
TokenService.exchangeCodeForToken()
GET oidcCode → StoredTokenData
generateIdToken() (JWT)
PUT access_token:uuid → discordToken
id_token + access_token


6. User Information
GET /userinfo (Bearer access_token)
UserInfoService.getUserInfo()
GET access_token:uuid → discordToken


GET oidcCode → StoredTokenData
generateIdToken() (JWT)
PUT access_token:uuid → discordToken
id_token + access_token
6. User Information
GET /userinfo (Bearer access_token)
UserInfoService.getUserInfo()
GET access_token:uuid → discordToken
GET /api/users/@me (getDiscordUserInfo)
Fresh DiscordUser data
OIDC user claims


GET oidcCode → StoredTokenData
generateIdToken() (JWT)
PUT access_token:uuid → discordToken
id_token + access_token
6. User Information
GET /userinfo (Bearer access_token)
UserInfoService.getUserInfo()
GET access_token:uuid → discordToken
GET /api/users/@me (getDiscordUserInfo)
Fresh DiscordUser data
OIDC user claims

まとめ

- OP(OpenID Provider)はそこそこ簡単に作れる

- OAuth2が使えるなら(理屈上)OIDCのログインが生やせる
  - たとえばX(Twitter)とか
  - Misskeyも？

- 自分で認証関係のもの作るの怖いね

“車が通れるほど大きな”
セキュリティーホールを抑えながら
ログインしたい

Kyoto Tech Talk #8

2025/5/29

京都産業大学 上村太成

WSOFT