HITACHI
Inspire the Next

Keycloak DevDay 2025
@greet Hotel Darmstadt, Darmstadt, Germany

Keycloak’s Compliance with Security Specifications: from OAuth 2.0, OIDC to OID4VCI

March 6, 2025

Takashi Norimatsu

Open Source Program Office / OSS Solution Center
Hitachi, Ltd.

Self Introduction

HITACHI
Inspire the Next

Takashi Norimatsu (**tnorimat** in GitHub) :

Ph.D. Student, Senior OSS Specialist, Hitachi, Ltd., Japan

Certified Information Systems Security Professional (CISSP),

Keycloak maintainer (since Oct 2021),

Technical lead of Keycloak community “OAuth SIG”

- Contributing security features to Keycloak since 2017.
  - W3C Web Authentication API support (Passkeys)
  - Security features support (e.g., RFC 7636 PKCE, RFC 8705 OAuth MTLS, OIDC CIBA, RFC 9126 PAR, RFC 8032/8037 EdDSA, RFC 9449 DPoP, RFC 9207 OAuth2 Authz Server Issuer Identification)
  - API security profiles support (e.g., FAPI 1.0 Baseline, FAPI 1.0 Advanced, FAPI-CIBA, FAPI 2.0 Baseline, FAPI 2.0 Message Signing, OAuth 2.1)

HITACHI
Inspire the Next

Contents

1. Keycloak-supported specifications
2. Pick-up some specifications
   - FAPI
   - OID4VCI

HITACHI
Inspire the Next

1. Keycloak-supported specifications
2. Pick-up some specifications
   - FAPI
   - OID4VCI

Keycloak-supported specifications

HITACHI
Inspire the Next

Keycloak supported specifications in the following categories:

- OAuth 2.0
  by Internet Engineering Task Force (IETF)
- OpenID Connect (OIDC)
  by OpenID Foundation
- Financial-grade API Security Profiles (FAPI)
  by OpenID Foundation
- Open Banking Security Profiles
  by some country's regulatory body
- Security Assertion Markup Language 2.0 (SAMLv2)
  by Organization for the Advancement of Structured Information Standards (OASIS)
- User Managed Access (UMA)
  by Kantara Initiative
- Web Authentication API (WebAuthn)
  by World Wide Web Consortium (W3C)
- The Federal Information Processing Standard Publication 140-2 (FIPS 140-2)
  by U.S. National Institute of Standards and Technology (NIST)

Definition

HITACHI
Inspire the Next

What does “**supporting a specification**” mean?

Implementing features by following the specification.

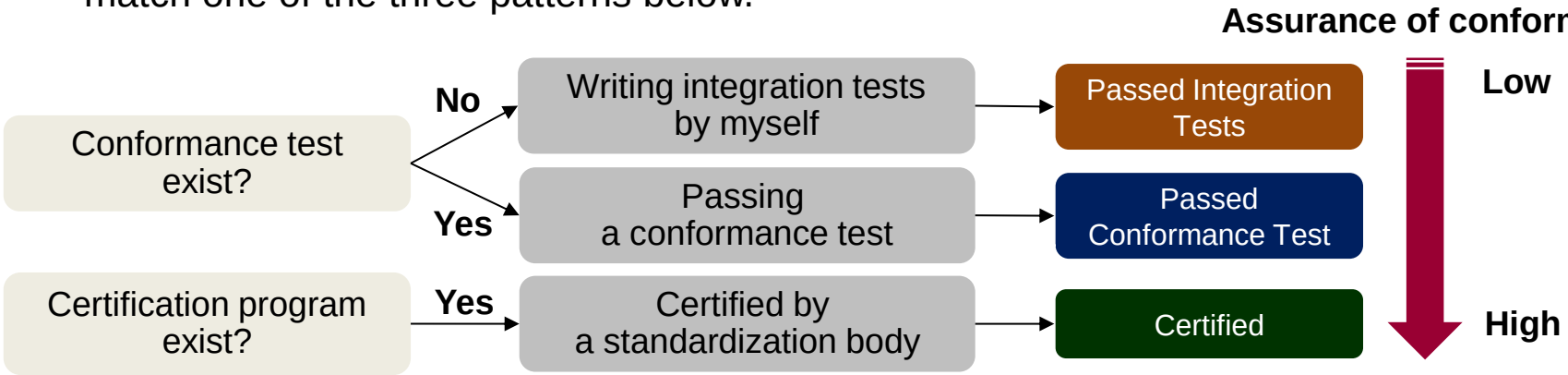
How can we check whether Keycloak supports a specification?

Confirming Keycloak can pass a **conformance test** of the specification.
Some standardization bodies provide conformance tests of their specifications.

Making Keycloak got certified by “**Certificate Program**”.
Some standardization bodies certifies a product with their specifications.
Ex. NIST : FIPS 140-2,3
FIDO Alliance : FIDO2 (WebAuthn, CTAP2)
OIDF : OIDC and its related specifications, FAPI,
Open Banking security profiles

Definition

This slide assumes that Keycloak supports a specification if Keycloak and the specification match one of the three patterns below.



What this slide describes are **not** Keycloak project's official opinion.
These were derived by just my investigation against Keycloak 26.1.

Practical Problems

HITACHI
Inspire the Next

- I heard that Keycloak can pass some conformance tests*, but this Keycloak is different from my Keycloak with my configuration in my deployment.
  It is OK to say that my Keycloak can also pass the conformance test?
- It is better to run the conformance test on your Keycloak and confirm the Keycloak can pass the test.

*: https://github.com/keycloak/kc-sig-fapi?tab=readme-ov-file#passed-conformance-tests-per-keycloak-version

Keycloak-supported specifications : OIDC, FAPI

HITACHI
Inspire the Next

Standardization Body: OpenID Foundation (OIDF)

Certified

#	Specification	WG	Status
1	OpenID Connect Core 1.0	OIDC	Final
2	OpenID Connect Discovery 1.0	OIDC	Final
3	OpenID Connect Dynamic Client Registration 1.0	OIDC	Final
4	OAuth 2.0 Multiple Response Type Encoding Practices	OIDC	Final
5	OAuth 2.0 Form Post Response Mode	OIDC	Final
6	OpenID Connect RP-Initiated Logout 1.0	OIDC	Final
7	OpenID Connect Session Management 1.0	OIDC	Final
8	OpenID Connect Front-Channel Logout 1.0	OIDC	Final
9	OpenID Connect Back-Channel Logout 1.0	OIDC	Final
10	OpenID Connect Client Initiated Backchannel Authentication Flow - Core 1.0	MODRNA	Final
11	Financial-grade API Security Profile 1.0 - Part 1: Baseline	FAPI	Final
12	Financial-grade API Security Profile 1.0 - Part 2: Advanced	FAPI	Final
13	JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)	FAPI	Final
14	Financial-grade API: Client Initiated Backchannel Authentication Profile	FAPI	Implementer’s Draft

Keycloak-supported specifications : OIDC, FAPI

HITACHI
Inspire the Next

Standardization Body: OpenID Foundation (OIDF)

Passed Conformance Test

#	Specification	WG	Status
15	FAPI 2.0 Security Profile	FAPI	Implementer's Draft
16	FAPI 2.0 Message Signing	FAPI	Draft

Standardization Body: OpenID Foundation (OIDF)

Passed Self-Integration Tests

#	Specification	WG	Status
17	OpenID for Verifiable Credential Issuance	OIDC	Implementer's Draft

⚠ "Supporting a specification" does **not** mean that Keycloak officially supports features of the specification. Some supports of specifications are treated as "**Preview**" or "**Experimental**" features.
Ex. support for #17 "OpenID for Verifiable Credential Issuance" is treated as "**Experimental**" feature by Keycloak 25 or later.

Keycloak-supported specifications : OIDC (details)

HITACHI
Inspire the Next

Standardization Body: OpenID Foundation (OIDF)

#	Specification	Conformance Profile	Status	Version
1	OpenID Connect Core 1.0	Basic OP	Certified	2.3.0, 18.0.0
2	OpenID Connect Core 1.0	Implicit OP	Certified	2.3.0, 18.0.0
3	OpenID Connect Core 1.0	Hybrid OP	Certified	2.3.0, 18.0.0
4	OpenID Connect Core 1.0	Config OP	Certified	2.3.0, 18.0.0
5	OpenID Connect Core 1.0	Dynamic OP	Certified	2.3.0, 18.0.0
6	OpenID Connect Core 1.0	Form Post OP	Certified	18.0.0
7	OpenID Connect Core 1.0	3rd Party-Init OP	Not Yet	-
8	OpenID Connect RP-Initiated Logout 1.0	RP-Initiated OP	Certified	18.0.0
9	OpenID Connect Session Management 1.0	Session OP	Certified	18.0.0
10	OpenID Connect Front-Channel Logout 1.0	Front-Channel OP	Certified	18.0.0
11	OpenID Connect Back-Channel Logout 1.0	Back-Channel OP	Certified	18.0.0

Keycloak-supported specifications : FAPI (details)

HITACHI
Inspire the Next

Standardization Body: OpenID Foundation (OIDF)

#	Specification	Conformance Profile	Status	Version
1	FAPI 1.0 - Advanced	FAPI Adv. OP w/ MTLS	Certified	15.0.2
2	FAPI 1.0 - Advanced	FAPI Adv. OP w/ MTLS, PAR	Certified	15.0.2
3	FAPI 1.0 - Advanced	FAPI Adv. OP w/ Private Key	Certified	15.0.2
4	FAPI 1.0 - Advanced	FAPI Adv. OP w/ Private Key, PAR	Certified	15.0.2
5	FAPI 1.0 - Advanced	FAPI Adv. OP w/ MTLS, JARM	Certified	15.0.2
6	FAPI 1.0 - Advanced	FAPI Adv. OP w/ Private Key, JARM	Certified	15.0.2
7	FAPI 1.0 - Advanced	FAPI Adv. OP w/ MTLS, PAR, JARM	Certified	15.0.2
8	FAPI 1.0 - Advanced	FAPI Adv. OP w/ Private Key, PAR, JARM	Certified	15.0.2
9	FAPI-CIBA Profile	FAPI-CIBA OP poll w/ MTLS	Certified	15.0.2
10	FAPI-CIBA Profile	FAPI-CIBA OP poll w/ Private Key	Certified	15.0.2
11	FAPI-CIBA Profile	FAPI-CIBA OP Ping w/ MTLS	Certified	15.0.2
12	FAPI-CIBA Profile	FAPI-CIBA OP Ping w/ Private Key	Certified	15.0.2

Keycloak-supported specifications : FAPI (details)

HITACHI
Inspire the Next

Standardization Body: OpenID Foundation (OIDF)

#	Specification	Conformance Profile	Status	Version
13	FAPI 2.0 Security Profile & Message Signing	FAPI2SP MTLS + MTLS	Passed Conformance Test	23.0.0
14	FAPI 2.0 Security Profile & Message Signing	FAPI2SP private key + MTLS	Passed Conformance Test	23.0.0
15	FAPI 2.0 Security Profile & Message Signing	FAPI2SP OpenID Connect	Passed Conformance Test	23.0.0
16	FAPI 2.0 Security Profile & Message Signing	FAPI2MS JAR	Passed Conformance Test	23.0.0
17	FAPI 2.0 Security Profile & Message Signing	FAPI2MS JARM	Passed Conformance Test	23.0.0
18	FAPI 2.0 Security Profile & Message Signing	FAPI2SP MTLS + DPoP	Passed Conformance Test	26.1.0
19	FAPI 2.0 Security Profile & Message Signing	FAPI2SP private key + DPoP	Passed Conformance Test	26.1.0

Keycloak-supported specifications : Open Banking

HITACHI
Inspire the Next

Standardization Body: OpenID Foundation (OIDF)

#	Specification	Conformance Profile	Status	Version
1	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP w/ MTLS	Certified	15.0.2
2	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP w/ Private Key	Certified	15.0.2
3	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP w/ MTLS, PAR	Certified	15.0.2
4	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP w/ Private Key, PAR	Certified	15.0.2
5	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP w/ MTLS, JARM	Certified	15.0.2
6	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP w/ Private Key, JARM	Certified	15.0.2
7	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP w/ MTLS, PAR, JARM	Certified	15.0.2
8	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP w/ Private Key, PAR, JARM	Certified	15.0.2
9	Brazil Open Banking (FAPI 1.0 - Advanced)	BR-OB Adv. OP DCR	Not Yet	-
10	Brazil Open Finance (FAPI-BR v2)	BR-OF Adv. OP w/ Private Key, PAR	Passed Conformance Test	23.0.1
11	Brazil Open Finance (FAPI-BR v2)	BR-OF Adv. OP DCR	Not Yet	-

Keycloak-supported specifications : FAPI, OpenBanking

HITACHI
Inspire the Next

Standardization Body: OpenID Foundation (OIDF)

#	Specification	Conformance Profile	Status	Version
12	UK Open Banking (FAPI 1.0 - Advanced)	UK-OB Adv. OP w/ MTLS	Passed Conformance Test	20.0.0
13	UK Open Banking (FAPI 1.0 - Advanced)	UK-OB Adv. OP w/ Private Key	Passed Conformance Test	20.0.0
14	Australia CDR (FAPI 1.0 - Advanced)	AU-CDR Adv. OP w/ Private Key	Certified	15.0.2
15	Australia CDR (FAPI 1.0 - Advanced)	AU-CDR Adv. OP w/ Private Key, PAR	Certified	15.0.2

Keycloak-supported specifications : OAuth2

HITACHI
Inspire the Next

Standardization Body: Internet Engineering Task Force (IETF)

Passed Self-Integration Tests

#	Specification	Status
1	RFC 6749: The OAuth 2.0 Authorization Framework	RFC
2	RFC 6750: The OAuth 2.0 Authorization Framework: Bearer Token Usage	RFC
3	RFC 7009: OAuth 2.0 Token Revocation	RFC
4	RFC 7521: Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants	RFC
5	RFC 7523: JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants	RFC
6	RFC 7591: OAuth 2.0 Dynamic Client Registration Protocol	RFC
7	RFC 7592: OAuth 2.0 Dynamic Client Registration Management Protocol	RFC
8	RFC 7636: Proof Key for Code Exchange by OAuth Public Clients	RFC
9	RFC 7662: OAuth 2.0 Token Introspection	RFC
10	RFC 8414: OAuth 2.0 Authorization Server Metadata	RFC
11	RFC 8628: OAuth 2.0 Device Authorization Grant	RFC
12	RFC 8693: OAuth 2.0 Token Exchange	RFC
13	RFC 8705: OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens	RFC

Keycloak-supported specifications : OAuth2

HITACHI
Inspire the Next

Standardization Body: Internet Engineering Task Force (IETF)

Passed Self-Integration Tests

#	Specification	Status
14	RFC 9101: The OAuth 2.0 Authorization Framework: JWT-Secured Authorization Request (JAR)	RFC
15	RFC 9126: OAuth 2.0 Pushed Authorization Requests	RFC
16	RFC 9207: OAuth 2.0 Authorization Server Issuer Identification	RFC
17	RFC 9449: Demonstration of Proof-of-Possession at the Application Layer (DPoP)	RFC
18	The OAuth 2.1 Authorization Framework	Internet Draft

“Supporting a specification” does **not** mean that Keycloak officially supports features of the specification. Some supports of specifications are treated as “**Preview**” or “**Experimental**” features.

Ex. support for #12 “RFC 8693: OAuth 2.0 Token Exchange” and #17 “RFC 9449: Demonstration of Proof-of-Possession at the Application Layer (DPoP)” are treated as “**Preview**” feature by Keycloak 23 or later.

Keycloak-supported specifications : UMA, SAML, WebAuthn

HITACHI
Inspire the Next

Standardization Body: Kantara Initiative

Passed Self-Integration Tests

#	Specification	Status
1	User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization (version 2.0)	Recommendation
2	Federated Authorization for User-Managed Access (UMA) 2.0	Recommendation

Standardization Body: Organization for the Advancement of Structured Information Standards (OASIS)

Passed Self-Integration Tests

#	Specification	Status
1	Security Assertion Markup Language 2.0 (SAML 2.0)	Published

Standardization Body: World Wide Web Consortium (W3C)

Passed Self-Integration Tests

#	Specification	Status
1	Web Authentication: An API for accessing Public Key Credentials Level 2	Recommendation

Standardization Body: U.S. National Institute of Standards and Technology (NIST)

Passed Self-Integration Tests

#	Specification	Status
1	The Federal Information Processing Standard Publication 140-2 (FIPS 140-2)	Published

Regression Tests for a new version of Keycloak

HITACHI
Inspire the Next

- Keycloak community activity : OAuth SIG (Special Interest Group) mainly supports OAuth and its related specifications to Keycloak.
  GitHub repository : https://github.com/keycloak/kc-sig-fapi (*1)
  CNCF slack channel : **#keycloak-oauth-sig**
  OAuth SIG works on security standards in this talk
  (Passkeys, OAuth 2.1, DPoP, OID4VCI, etc.)

- Whenever a new version of Keycloak is released, OAuth SIG runs conformance tests for all OIDF's specifications that Keycloak has already supported against it.

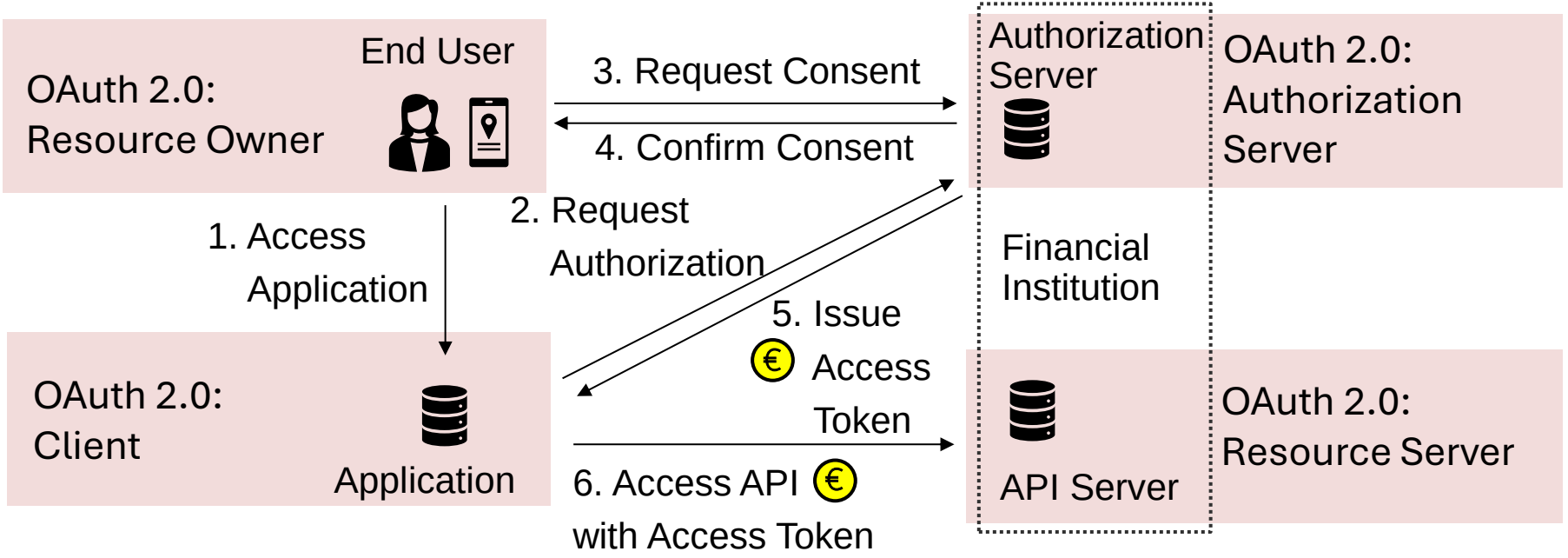
- OAuth SIG publishes the conformance test run results:
  https://github.com/keycloak/kc-sig-fapi?tab=readme-ov-file#passed-conformance-tests-per-keycloak-version

HITACHI
Inspire the Next

1. Keycloak-supported specifications

2. Pick-up some specifications
- FAPI
- OID4VCI

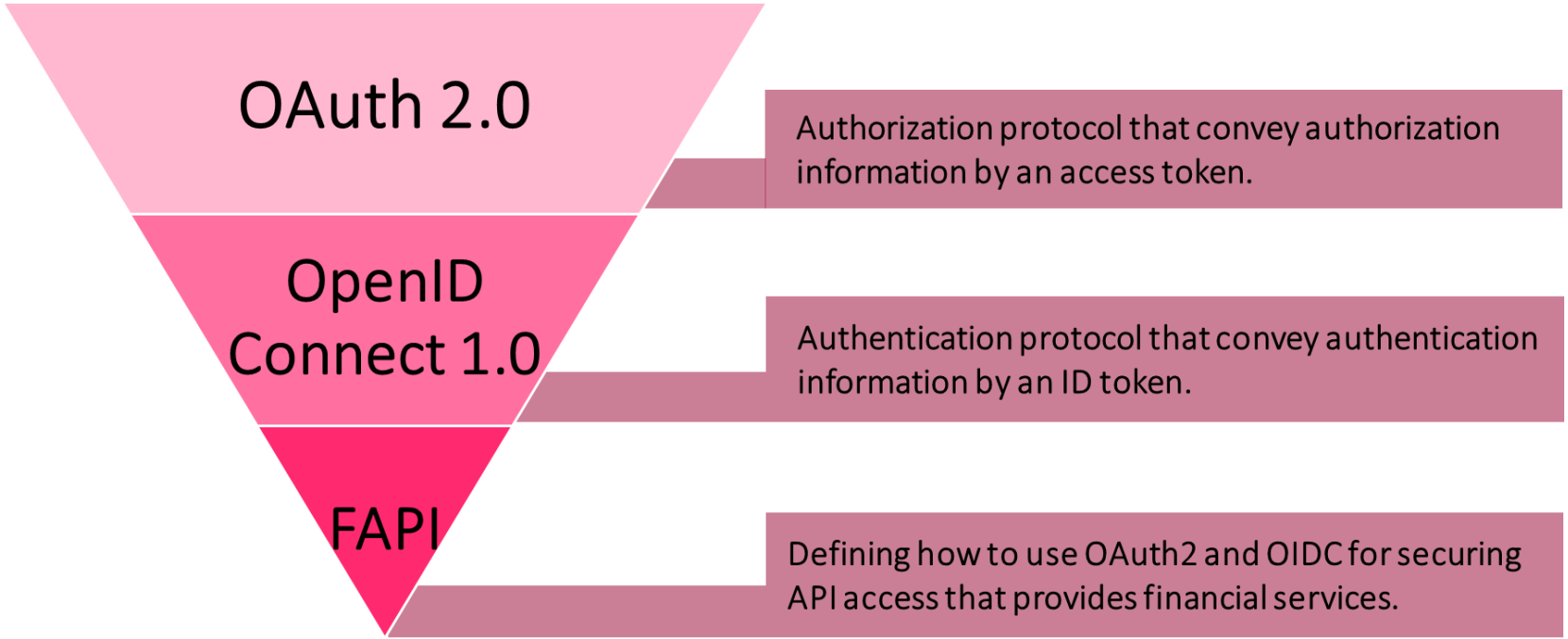
Background: Accessing APIs providing financial services



Financial-grade API (FAPI) security profiles

HITACHI
Inspire the Next

- Hardening OAuth 2.0 authorization and OpenID Connect 1.0 authentication protocol. It is standardized by OpenID Foundation.



Open Banking security profiles based on FAPI

HITACHI
Inspire the Next


[UK : OpenBanking]

OpenBanking Financial Grade API (FAPI) Profile

OpenBanking CIBA Profile


[Australia : Consumer Data Right (CDR)]

Consumer Data Right Security Profile


[Brazil : Open Banking Brazil]

Open Banking/Finance Brazil Financial-grade API Security Profile

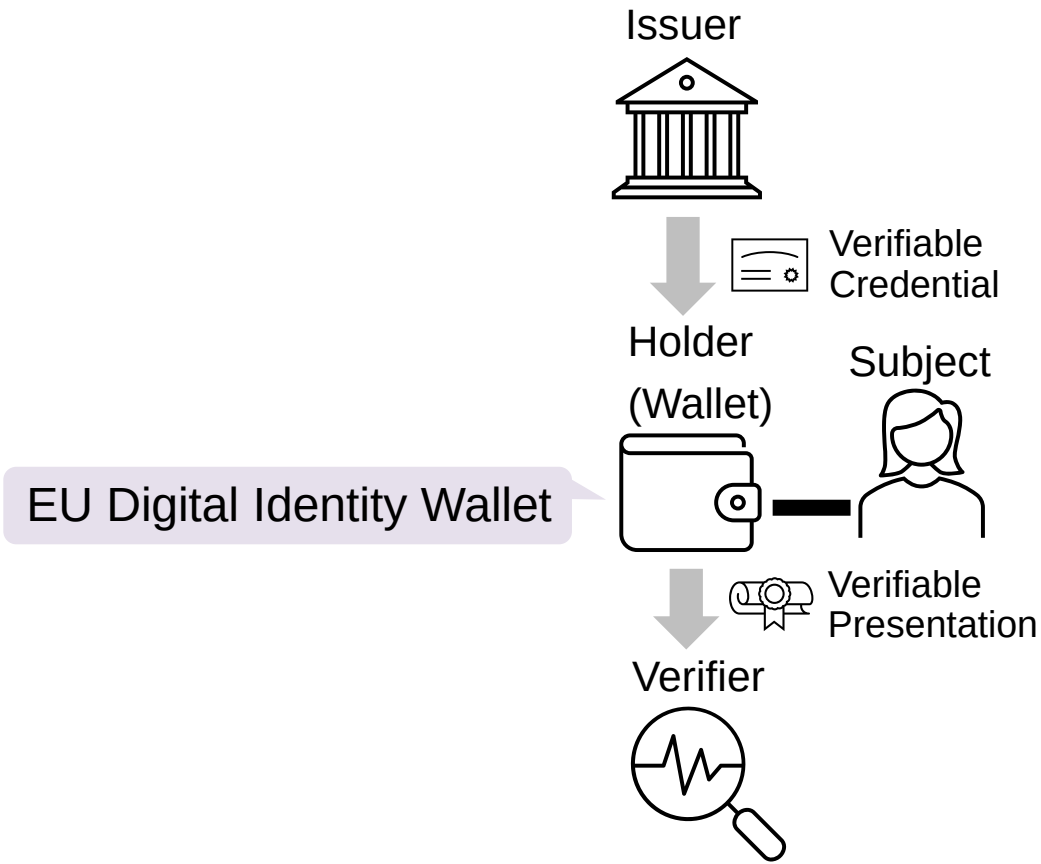
HITACHI
Inspire the Next

1. Keycloak-supported specifications

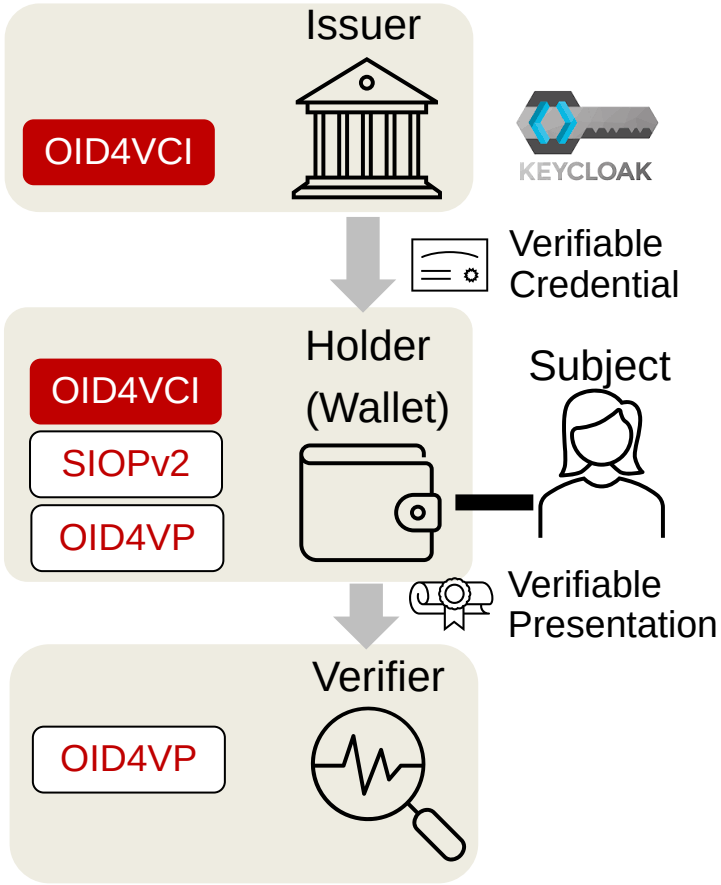
2. Pick-up some specifications

- FAPI
- **OID4VCI**

Background: Managing personal digital data



OpenID for Verifiable Credentials Issuance (OID4VCI)



EU Digital Identity Wallet

OpenID for Verifiable Credential Issuance (OID4VCI):= Defining an API that is used to issue Verifiable Credentials. (quoted from *2)

Self-Issued OpenID Provider v2 (SIOPv2) := An OpenID Provider controlled by the End-User.
(quoted from *3)

OpenID for Verifiable Presentations (OID4VP) := Defining a mechanism on top of OAuth 2.0 that enables presentation of Verifiable Credentials as Verifiable Presentations. (quoted from *4)

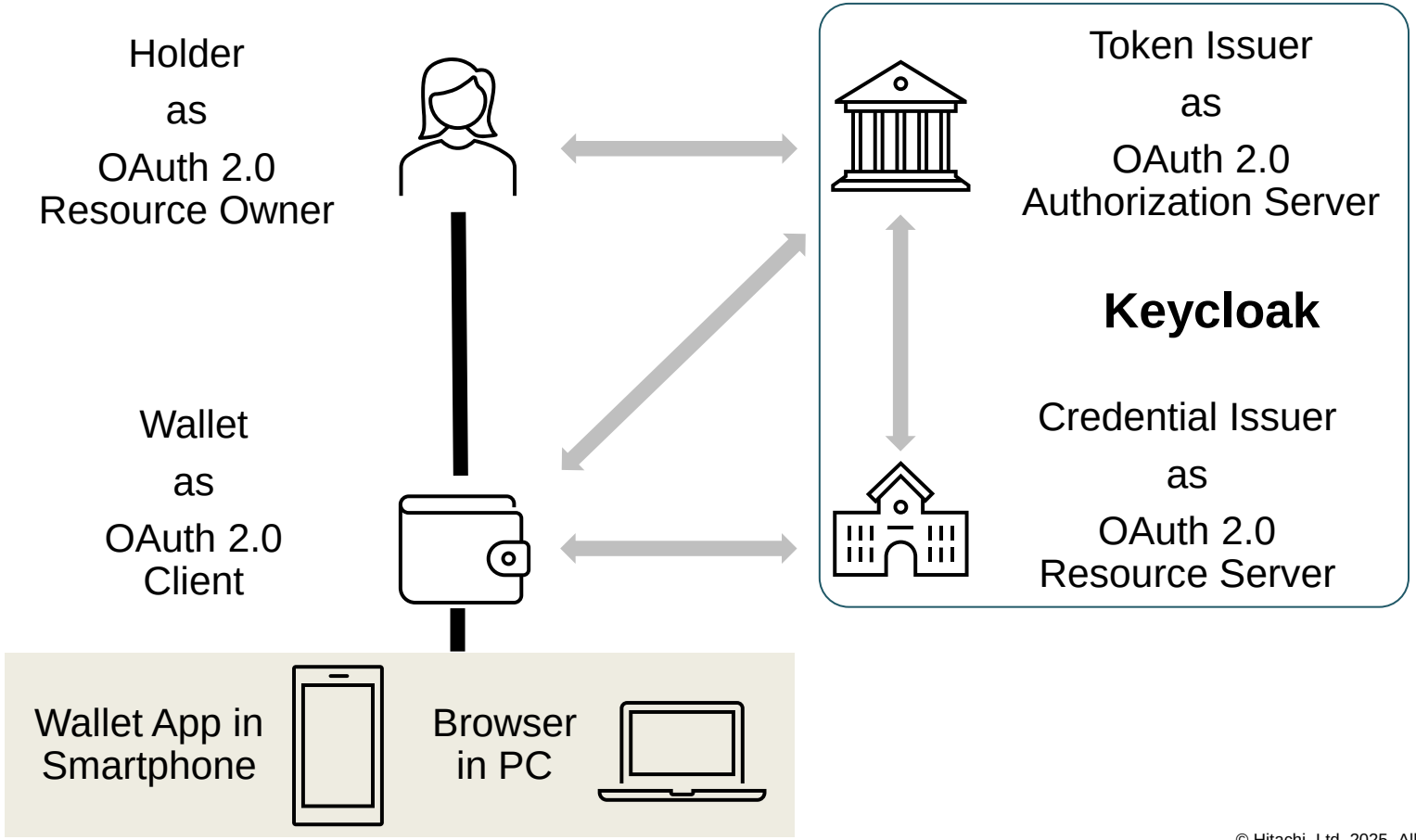
*2 : https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html

*3 : https://openid.net/specs/openid-connect-self-issued-v2-1_0.html

*4 : https://openid.net/specs/openid-4-verifiable-presentations-1_0.html

OpenID for Verifiable Credentials Issuance (OID4VCI)

HITACHI
Inspire the Next



OpenID for Verifiable Credentials Issuance (OID4VCI)

HITACHI
Inspire the Next

Keycloak's community is working on supporting OID4VCI.

Motivation :

The European Commission released "The European Digital Identity Wallet Architecture and Reference Framework (*1)" which describes that OID4VCI MUST be implemented as an Issuance Protocol.

➡ Keycloak can be used as an Issuer in this framework if Keycloak supports OID4VCI.

Credential Formats :

- JWT VC (*2)
  Standardized by Decentralized Identity Foundation (DIF)
- Selective Disclosure JWT (SD-JWT) (*3)
  Standardized by Internet Engineering Task Force (IETF)
- Verifiable Credentials Data Model (VCDM) (*4)
  Standardized by World Wide Web Consortium (W3C)
- ISO.18013-5 Mobile driving license (mDL) (*5)
  Standardized by International Organization for Standardization (ISO)

Credential Issuance Protocol

- OpenID for Verifiable Credential Issuance (OID4VCI) (*6)
  Standardized by OpenID Foundation (OIDF)

  Issuance Flows :
  - Pre-authorization code flow
  - Authorization code flow

*1 : https://digital-strategy.ec.europa.eu/en/library/european-digital-identity-wallet-architecture-and-reference-framework

*2 : https://identity.foundation/jwt-vc-presentation-profile/
*3 : https://www.ietf.org/archive/id/draft-ietf-oauth-selective-disclosure-jwt-07.html
*4 : https://www.w3.org/TR/vc-data-model/
*5: https://www.iso.org/standard/69084.html
*6 : https://openid.net/specs/openid-4-verifiable-credential-issuance-1_0.html

OpenID for Verifiable Credentials Issuance (OID4VCI)

HITACHI
Inspire the Next

OID4VC Support - General	
OID4VCI-supported Keycloak's version	26.0.0 or later
OID4VCI support feature level	Experimental
Referred Version of OID4VCI specification	Implementer's Draft (draft 13)

OID4VC Management/Administration	
Admin REST API direct access	✅
Admin CLI (kcadm)	✅
Admin Console	❌

VC Issuance Flow	
Pre-Authorized Code Flow	✅
Authorized Code Flow	✅
Authorization Request Parameter	
scope	✅
authorization_details (RFC 9396 RAR)	❌
issuer_state	❌

VC Issuance Variation	
Immediate Credential Issuance	✅
Deferred Credential Issuance	❌
Batch Credential Issuance	❌

VC Credential Offer	
Same-Device	✅
Cross-Device	❓

VC Format	
SD-JWT VC (IETF)	✅
JWT VC (DIF)	❓
LDP VC (W3C)	❓
mDL (ISO.18013-5)	❌

VC Issuance Proof (Key Binding)	
jwt (SD-JWT)	❌
cwt	❌
ldp_vp (VCDM)	❌

OpenID for Verifiable Credentials Issuance (OID4VCI)

HITACHI
Inspire the Next

- Current focus points
  - How to determine which VC is issued
    - Client-based to Scope-based
  - Where we define the credentials
    - Per Client (client attributes) to Per Realm (protocol mapper's configuration or realm attributes)
  - "Protocol" attribute of a client for OID4VCI
    - Different Protocol (oid4vc) to Same Protocol (oidc)

      Different Protocol : oidc client for OAuth2/OIDC while oid4vc client for OID4VCI

      Same Protocol : oidc client for both OAuth2/OIDC and OID4VCI

      KC25 implementation : Pre-authorization code flow : protocol = oid4vc

      Authorization code flow : protocol = oidc
  - VC Format
    - mDL (ISO.18013-5)
  - VC Issuance Proof (Key Binding)
    - jwt (SD-JWT)
  - Follow the latest version of OID4VCI specification (Implementer's Draft)
    - Draft version 13 (KC 25 followed) to 14 (current ver), …

Summary

HITACHI
Inspire the Next

- Keycloak supported a lot of security specifications, but we need to take care that what “supporting a specification” means.
- Among them, Keycloak supported specifications for securing API access and managing personal digital data (e.g., EUDI wallet).
- Keycloak community OAuth SIG runs regression tests of some specifications on new version of Keycloak (e.g., OIDC, FAPI, OpenBanking).

Appendix

HITACHI
Inspire the Next

#	Specification	Standardization Body	Status	Support Lv. by KC	Conformance Test exist?	Certificate Program exist?	Self-Integration Test Passed?	Conformance Test Passed?	Certi-fied?
1	RFC 6749: The OAuth 2.0 Authorization Framework	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
2	RFC 6750: The OAuth 2.0 Authorization Framework: Bearer Token Usage	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
3	RFC 7009: OAuth 2.0 Token Revocation	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
4	RFC 7521: Assertion Framework for OAuth 2.0 Client Authentication and Authorization Grants	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
5	RFC 7523: JSON Web Token (JWT) Profile for OAuth 2.0 Client Authentication and Authorization Grants	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
6	RFC 7591: OAuth 2.0 Dynamic Client Registration Protocol	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
7	RFC 7592: OAuth 2.0 Dynamic Client Registration Management Protocol	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
8	RFC 7636: Proof Key for Code Exchange by OAuth Public Clients	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
9	RFC 7662: OAuth 2.0 Token Introspection	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
10	RFC 8414: OAuth 2.0 Authorization Server Metadata	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
11	RFC 8628: OAuth 2.0 Device Authorization Grant	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
12	RFC 8693: OAuth 2.0 Token Exchange	IETF (OAuth WG)	RFC	Preview	-	-	✅	-	-
13	RFC 8705: OAuth 2.0 Mutual TLS Client Authentication and Certificate Bound Access Tokens	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
14	RFC 9101: The OAuth 2.0 Authorization Framework: JWT-Secured Authorization Request (JAR)	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
15	RFC 9126: OAuth 2.0 Pushed Authorization Requests	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
16	RFC 9207: OAuth 2.0 Authorization Server Issuer Identification	IETF (OAuth WG)	RFC	Supported	-	-	✅	-	-
17	RFC 9449: Demonstration of Proof-of-Possession at the Application Layer (DPoP)	IETF (OAuth WG)	RFC	Preview	-	-	✅	-	-
18	The OAuth 2.1 Authorization Framework	IETF (OAuth WG)	Internet Draft	Supported	-	-	✅	-	-

Appendix

HITACHI
Inspire the Next

#	Specification	Standardization Body	Status	Support Lv. by KC	Conformance Test exist?	Certificate Program exist?	Self-Integration Test Passed?	Conformance Test Passed?	Certi-fied?
19	OpenID Connect Core 1.0	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
20	OpenID Connect Discovery 1.0	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
21	OpenID Connect Dynamic Client Registration 1.0	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
22	OAuth 2.0 Multiple Response Type Encoding Practices	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
23	OAuth 2.0 Form Post Response Mode	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
24	OpenID Connect RP-Initiated Logout 1.0	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
25	OpenID Connect Session Management 1.0	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
26	OpenID Connect Front-Channel Logout 1.0	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
27	OpenID Connect Back-Channel Logout 1.0	OIDF (OIDC WG)	Final	Supported	✅	✅	✅	✅	✅
28	OpenID for Verifiable Credential Issuance	OIDF (OIDC WG)	Implementer's Draft	Experimental	ー(coming soon?)	- (coming soon?)	✅	-	-
29	OpenID Connect Client Initiated Backchannel Authentication Flow - Core 1.0	OIDF (MODRNA WG)	Final	Supported	✅	✅	✅	✅	✅
30	Financial-grade API Security Profile 1.0 - Part 1: Baseline	OIDF (OIDC FAPI)	Final	Supported	✅	✅	✅	✅	✅
31	Financial-grade API Security Profile 1.0 - Part 2: Advanced	OIDF (OIDC FAPI)	Final	Supported	✅	✅	✅	✅	✅
32	JWT Secured Authorization Response Mode for OAuth 2.0 (JARM)	OIDF (OIDC FAPI)	Final	Supported	✅	✅	✅	✅	✅
33	Financial-grade API: Client Initiated Backchannel Authentication Profile	OIDF (OIDC FAPI)	Implementer's Draft	Supported	✅	✅	✅	✅	✅
34	FAPI 2.0 Security Profile	OIDF (OIDC FAPI)	Implementer's Draft	Supported	✅	✅	✅	✅	❌
35	FAPI 2.0 Message Signing	OIDF (OIDC FAPI)	Draft	Supported	✅	✅	✅	✅	❌
36	The Federal Information Processing Standard Publication 140-2 (FIPS 140-2)	NIST	Published	Supported	✅	✅	✅	❓	❓
37	Web Authentication: An API for accessing Public Key Credentials Level 2	W3C	Recommendation	Supported	✅	✅	✅	❌	❌
38	User-Managed Access (UMA) 2.0 Grant for OAuth 2.0 Authorization (version 2.0)	Kantara Initiative	Recommendation	Supported	❓	❓	✅	❓	❓
39	Federated Authorization for User-Managed Access (UMA) 2.0	Kantara Initiative	Recommendation	Supported	❓	❓	✅	❓	❓
40	Security Assertion Markup Language 2.0 (SAML 2.0)	OASIS	Published	Supported	❓	❓	✅	❓	❓

Trademarks

HITACHI
Inspire the Next

- OpenID is a trademark or registered trademark of OpenID Foundation in the United States and other countries.
- GitHub is a trademark or registered trademark of GitHub, Inc. in the United States and other countries.
- Red Hat is a trademark or registered trademark of Red Hat, Inc. in the United States and other countries.
- X is a trademark or registered trademark of X CORP. in the United States and other countries.
- Facebook is a trademark or registered trademark of Meta Platforms, Inc. in the United States and other countries.
- Other brand names and product names used in this material are trademarks, registered trademarks, or trade names of their respective holders.

HITACHI
Inspire the Next

END

Keycloak’s Compliance with Security Specifications: from OAuth 2.0, OIDC to OID4VCI

March 6, 2025

Takashi Norimatsu

Open Source Program Office / OSS Solution Center
Hitachi, Ltd.


Hitachi Social Innovation is
POWERING GOOD