Building a National-Scale IdP

Shortcomings of OIDC

BRENDON VICENTE R. SILVA
FREDERICO SCHARDONG, PhD
RICARDO F. CUSTÓDIO, PhD

The Brazilian Context

→ About 210 million citizens;

→ Strong regional, cultural, and economic diversity;

→ Digital inequality;

- Wealth inequality.
- Varying internet access and device availability.

→ Low digital literacy in significant portions of the population.

Identity systems *must* adapt to different realities and capabilities

The Digital Identity Landscape

→ Gov.br

- Maintained by the executive branch of government;
- Main portal for access to government services.

The Digital Identity Landscape

→ Gov.br

- Maintained by the executive branch of government;
- Main portal for access to government services.

→ ICP-Brasil

- Maintained by the executive branch of government;
- Provides a X.509 Public Key Infrastructure.

The Digital Identity Landscape

→ Gov.br
- Maintained by the executive branch of government;
- Main portal for access to government services.

→ ICP-Brasil
- Maintained by the executive branch of government;
- Provides a X.509 Public Key Infrastructure.

→ IdRC
- Maintained by the judiciary branch of government;
- Based on citizens' primary records.

Complementary — not competing — systems

Civil Registry Role

→ A Century-Old Institution;

- Established in 1874;
- Issues the *foundational identity*;

→ Over 7000 registry offices across all municipalities;

→ Federated and decentralized structure.

About IdRC

→ Source of information is decentralized (public registrars);

→ When a birth certificate is issued, an eID is issued;
When a death certificate is issued, the eID is deactivated;

→ Only supports OIDC/OAuth 2.0, will adopt VCs;

→ 210 million registered users, about 12 million monthly active users;

Where Standards Fall Short #1

→ Standards[1] define two kinds of ***Level of Assurance*** **(LoA)**:

- **Identity Assurance Level (IAL):** strength of identity proofing during registration;
- **Authenticator Assurance Level (AAL):** reflects the authentication process.

→ OIDC Core exposes only one mechanism to represent the level of assurance;

- The ***Authentication Context Reference*** **(`acr`)** claim.

Combining LoAs for Real-World Compatibility

→ IdRC evaluates both LoAs:

- From *identity proofing*;
- From *authentication event*.

→ Only the ***lowest*** between them is considered

Authentication LoA		Identity LoA: LOW	Identity LoA: SUBSTANTIAL	Identity LoA: HIGH
LOW	**1AF of any category**	LOW	LOW	LOW
SUBSTANTIAL	**MFA of any category**	LOW	SUBSTANTIAL	SUBSTANTIAL
SUBSTANTIAL	**ICP-Brasil authentication**	-	SUBSTANTIAL	SUBSTANTIAL
HIGH	**MFA with inherence factors**	-	SUBSTANTIAL	HIGH
HIGH	**2D Facial Biometrics + In-person Validation**	-	-	HIGH

Authentication Methods References

→ *Authentication Methods References* (`amr`)...

→ A claim defined in OIDC to indicate the authentication factors employed by the user.

→ Purely **enumerative,** with **no semantics**.

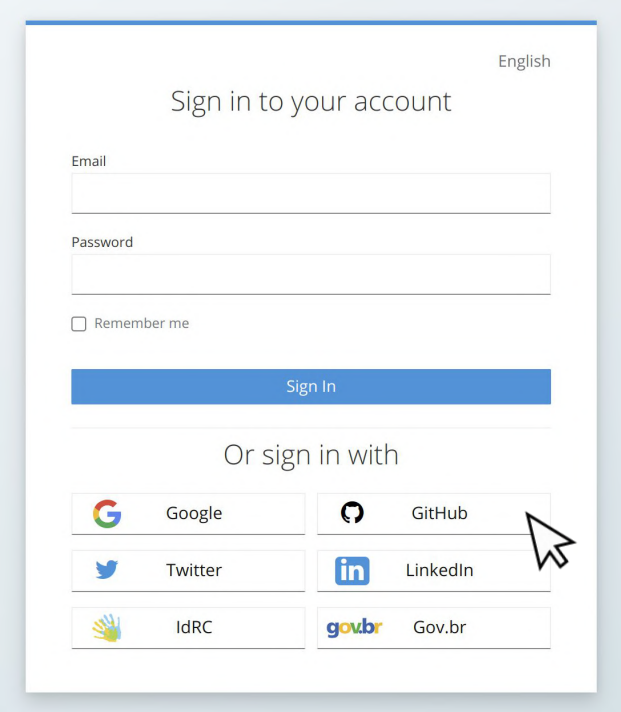

```
{
 "exp": 1756765376,
 "iat": 1756765076,
 "auth_time": 1756765076,
 "acr": "substantial",
 "amr": [ "sms", "pwd" ],
 "jti": "0c23f70e-5089-4246-98fc-288819b3dd50",
 "iss": "issuer.com",
 "aud": "account",
 "sub": "c213cced-4d5f-4542-8c54-817e4a09a353",
 "typ": "Bearer",
 "azp": "account-console",
 "nonce": "6f7dbfb3-37b9-4114-9e79-bcbcb7f5dfc4",
 "session_state":
"6752fc7e-6ec9-4c2b-a3ca-e6a8bc6bc81c",
 "scope": "openid",
 "sid": "6752fc7e-6ec9-4c2b-a3ca-e6a8bc6bc81c"
}
```

Authentication Methods References

→ Contextual information is lost:

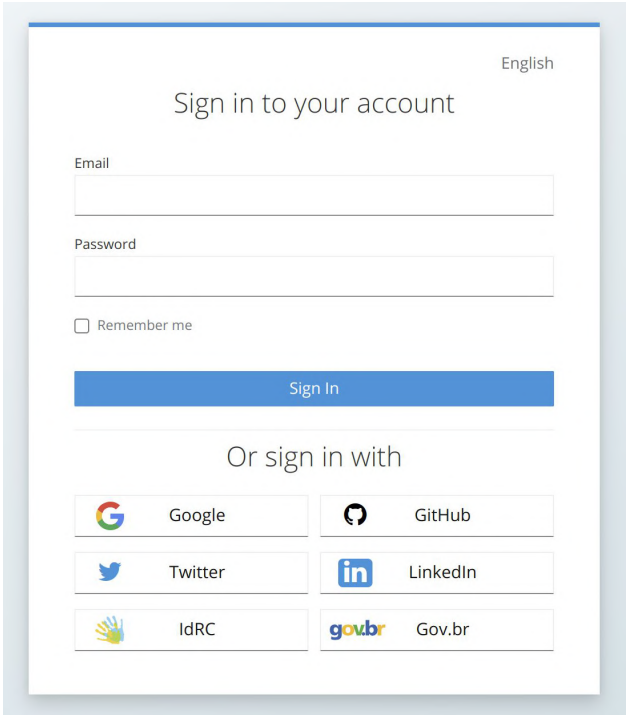
- Which Identity Provider validated each factor?
- What is the LoA associated with each method?
- When was each factor validated?
- Which password policy was used?

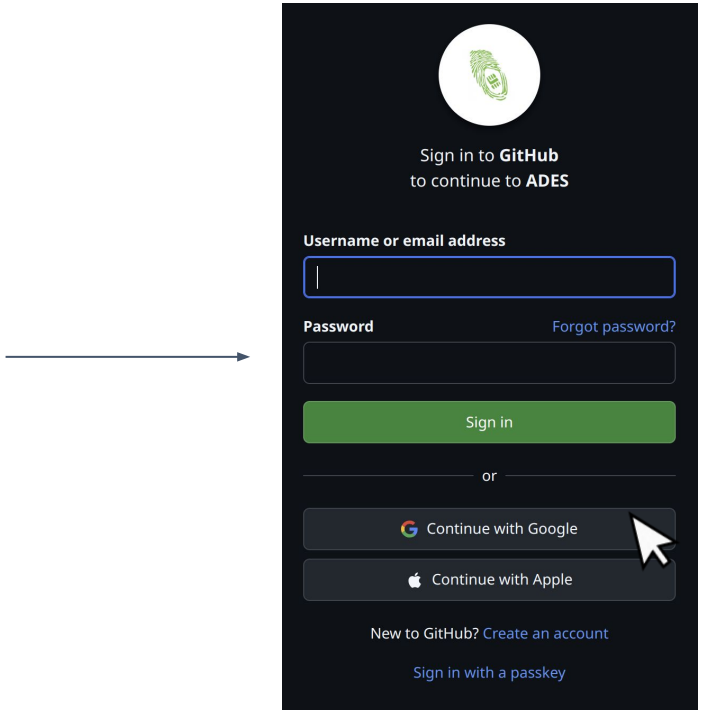

```
{
 "exp": 1756765376,
 "iat": 1756765076,
 "auth_time": 1756765076,
 "acr": "substantial",
 "amr": [ "sms", "pwd" ],
 "jti": "0c23f70e-5089-4246-98fc-288819b3dd50",
 "iss": "issuer.com",
 "aud": "account",
 "sub": "c213cced-4d5f-4542-8c54-817e4a09a353",
 "typ": "Bearer",
 "azp": "account-console",
 "nonce": "6f7dbfb3-37b9-4114-9e79-bcbcb7f5dfc4",
 "session_state":
"6752fc7e-6ec9-4c2b-a3ca-e6a8bc6bc81c",
 "scope": "openid",
 "sid": "6752fc7e-6ec9-4c2b-a3ca-e6a8bc6bc81c"
}
```



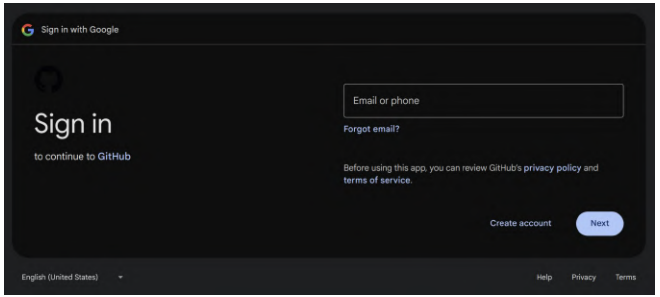
User starts authentication process and selects GitHub as the Identity Provider

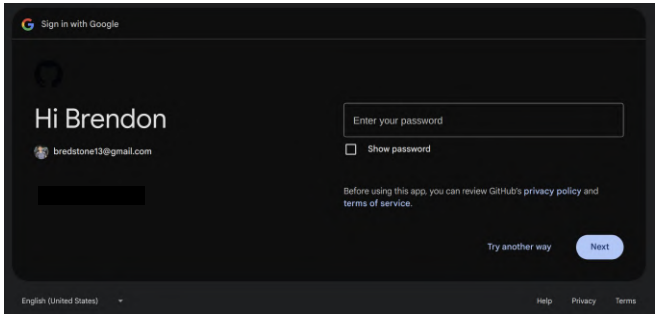
ILLUSTRATIVE EXAMPLE #1

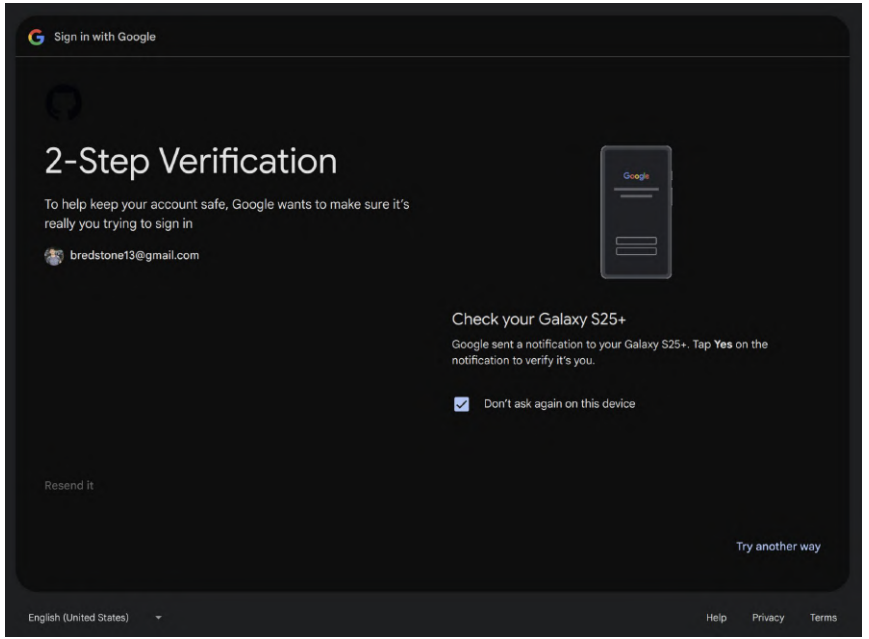




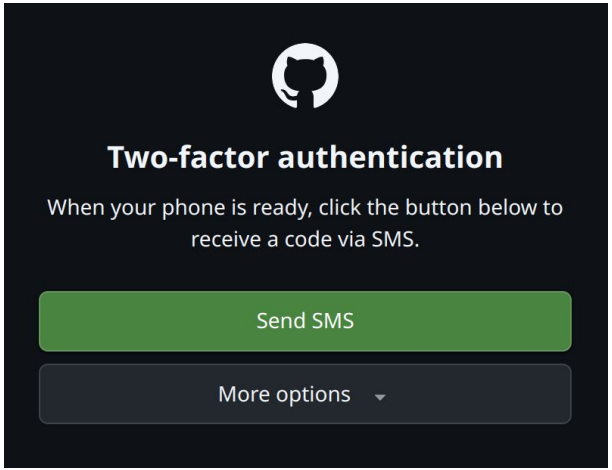
Inside GitHub, user then selects Google as their Identity Provider

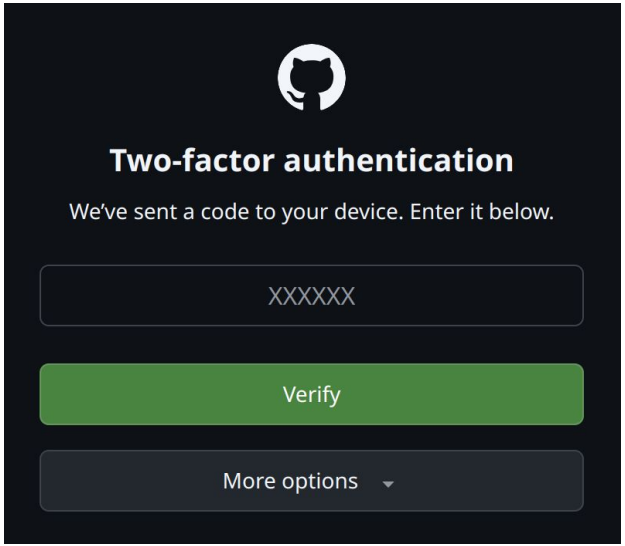






User utilizes password and WebAuthn to authenticate into Google Account





User is redirected back to GitHub, then authenticates using a TOTP delivered via SMS

Finally, the user is authenticated to the initial application.

The resulting token may contain the following `amr`:

```
["pwd", "user", "sms"]
```

→ OIDC lacks a standardized way of requesting authentication factors;

- As a result, applications can't specify **how** factors should be presented or processed.

→ Only high-level authentication requirements can be requested through the `acr_values` parameter;

- This provides no instruction for requesting which requirements must be used;
- Lack of defining relationships between the utilized factors. (*e.g.*, `pwd` AND `otp`, or `face` OR `hwk`)

Thus, the resulting `amr` fails to describe the user's authentication journey to a satisfactory level of detail.

This lack of description has tangible consequences:

→ Lack of semantic clarity

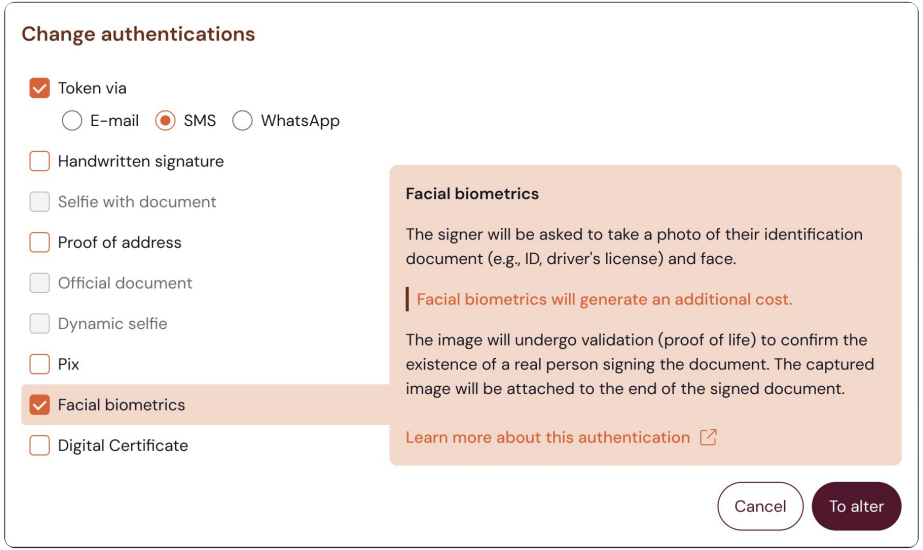
→ Regulatory non-alignment

→ Audit and accountability limitations

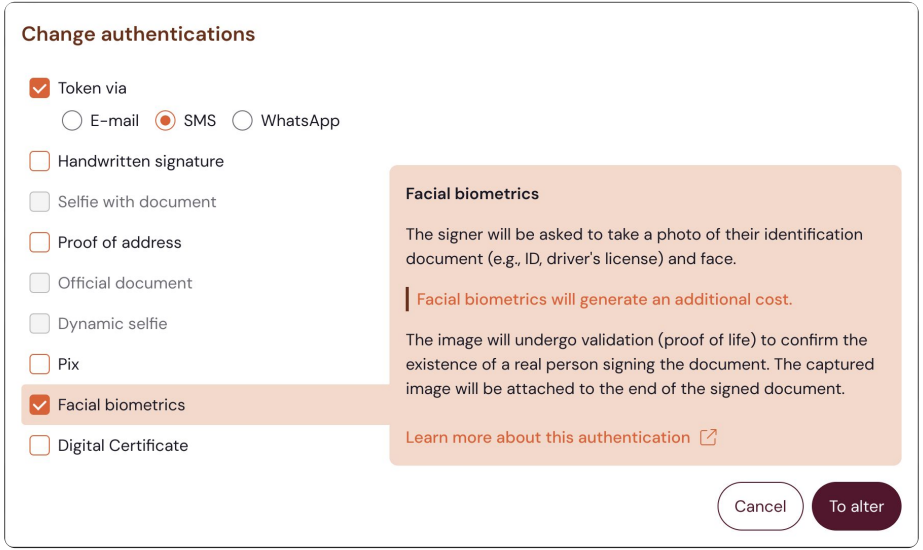
→ A digital signature platform, where you can invite signatories to sign a document.

→ The document manager can specify how other users should be authenticated. This includes:

- Specific authentication methods;
- Minimum authentication requirements.



→ This exemplifies a gap in OIDC, which often results in applications having to resort to **proprietary solutions,** which could have a number of unintended consequences due to lack of standardization.



Proposal:
OIDC for Authentication Context (OIDC4AC)

OBJECTIVES

→ Describe in details the authentication factors employed;

→ Enable applications to negotiate authentication methods alongside specific metadata.

DESIGN PRINCIPLES

→ Compatible with OIDC and existing implementations;

→ Gradual adoption;

→ Aligned, in vocabulary and structure, with the OIDF family of specifications.

Authentication Factors Representation

→ The solution revolves around the introduction of a new claim: `amr_details`

→ A JSON array where each element represents a single factor.

```json
{
  "amr": ["pwd", "face"],
  "amr_details": [ {
    "auth_method": "pwd",
    "src": {
      "iss": "https://idp.example.com",
      "trust_framework": "example-trust",
      "assurance_level": "low",
      "time": "2025-10-14T12:03:25Z"
    },
    "auth_details": {
      "derivation_algo": "argon2id",
      "iterations": 3,
      "salt_bytes": 16,
      "created_at": "2025-09-12T08:41:17Z"
    }
  }, {
    "auth_method": "face",
    "src": {
      "iss": "https://external.idp.com",
      "time": "2025-10-14T13:21:48Z",
      "location": { "ip_address": "203.0.113.42" }
    },
    "auth_details": {
      "image_format": "JPEG2000",
      "image_quality_score": 0.92,
      "liveness_result": "passed",
      "pad_method": "texture_analysis"
    }
  } ]
}
```

"amr_details"

→ Each entry in `amr_details` consists of:

`auth_method` (required):

- String identifier of the factor being detailed (*e.g.*, `pwd`, `hwk`, `sms`);
- Its value must correspond to an entry in amr claim.

```
{
  "amr": ["pwd", "face"],
  "amr_details": [ {
    "auth_method": "pwd",
    "src": {
      "iss": "https://idp.example.com",
      "trust_framework": "example-trust",
      "assurance_level": "low",
      "time": "2025-10-14T12:03:25Z"
    },
    "auth_details": {
      "derivation_algo": "argon2id",
      "iterations": 3,
      "salt_bytes": 16,
      "created_at": "2025-09-12T08:41:17Z"
    }
  }, {
    "auth_method": "face",
    "src": {
      "iss": "https://external.idp.com",
      "time": "2025-10-14T13:21:48Z",
      "location": { "ip_address": "203.0.113.42" }
    },
    "auth_details": {
      "image_format": "JPEG2000",
      "image_quality_score": 0.92,
      "liveness_result": "passed",
      "pad_method": "texture_analysis"
    }
  } ]
}
```

"amr_details"

→ Each entry in `amr_details` consists of:

`auth_method` (required):

- String identifier of the factor being detailed (*e.g.*, `pwd`, `hwk`, `sms`);
- Its value must correspond to an entry in amr claim.

`src` (required):

- JSON object summarizing the source of authentication.

```
{
  "amr": ["pwd", "face"],
  "amr_details": [ {
    "auth_method": "pwd",
    "src": {
      "iss": "https://idp.example.com",
      "trust_framework": "example-trust",
      "assurance_level": "low",
      "time": "2025-10-14T12:03:25Z"
    },
    "auth_details": {
      "derivation_algo": "argon2id",
      "iterations": 3,
      "salt_bytes": 16,
      "created_at": "2025-09-12T08:41:17Z"
    }
  }, {
    "auth_method": "face",
    "src": {
      "iss": "https://external.idp.com",
      "time": "2025-10-14T13:21:48Z",
      "location": { "ip_address": "203.0.113.42" }
    },
    "auth_details": {
      "image_format": "JPEG2000",
      "image_quality_score": 0.92,
      "liveness_result": "passed",
      "pad_method": "texture_analysis"
    }
  } ]
}
```

"amr_details"

→ Each entry in `amr_details` consists of:

`auth_method` (required):

- String identifier of the factor being detailed (*e.g.*, `pwd`, `hwk`, `sms`);
- Its value must correspond to an entry in amr claim.

`src` (required):

- JSON object summarizing the source of authentication.

`auth_details` (required):

- JSON object with specific metadata depending on the authentication method detailed.
- (e.g., hash algorithm used to store a password, facial biometrics algorithm, TOTP code size)

```
{
  "amr": ["pwd", "face"],
  "amr_details": [ {
    "auth_method": "pwd",
    "src": {
      "iss": "https://idp.example.com",
      "trust_framework": "example-trust",
      "assurance_level": "low",
      "time": "2025-10-14T12:03:25Z"
    },
    "auth_details": {
      "derivation_algo": "argon2id",
      "iterations": 3,
      "salt_bytes": 16,
      "created_at": "2025-09-12T08:41:17Z"
    }
  }, {
    "auth_method": "face",
    "src": {
      "iss": "https://external.idp.com",
      "time": "2025-10-14T13:21:48Z",
      "location": { "ip_address": "203.0.113.42" }
    },
    "auth_details": {
      "image_format": "JPEG2000",
      "image_quality_score": 0.92,
      "liveness_result": "passed",
      "pad_method": "texture_analysis"
    }
  } ]
}
```

Authentication Factors Request

→ Enables applications to express complex authentication requirements by combining **logical operators, parameter constraints, and contextual metadata**

- This is supported through OIDC's "claims" parameter

→ Introduces a set of operators for authentication factors request:

- `one_of` and `all_of`
- `min` and `max`
- `max_age`

```
{
  "id_token": {
    "amr_details": [ {
      "all_of": [ {
        "auth_method": {
          "value": "pwd"
        }
      }, {
        "one_of": [ {
          "auth_method": {
            "value": "otp"
          }
        }, {
          "auth_method": {
            "value": "face"
          }
        } ]
      } ]
    } ]
  }
}
```

Benefits & Applications

1. Enhanced auditability and traceability

Enables detailed registry of ***how***, ***where***, ***when*** and ***by whom*** each authentication factor was validated.

Benefits & Applications

2. Enhanced options for RPs

OIDC4AC allows RPs to more precisely express their authentication requirements.

Benefits & Applications

3. Granular access control and risk-based authentication

Applications can make more informed decisions.
Example: dynamically adjust permissions based on the assurance level, location metadata, or number of authentication attempts.

Benefits & Applications

4. Semantic interoperability

Due to its alignment with terminologies from other OpenID family profiles, the solution promotes integration within identity ecosystems and can be gradually adopted.

Thank you!

BRENDON VICENTE R. SILVA <BREDSTONE13@GMAIL.COM>
FREDERICO SCHARDONG <FREDERICO.SCHARDONG@ROLANTE.IFRS.EDU.BR>
RICARDO F. CUSTÓDIO <RICARDO.CUSTODIO@UFSC.BR>