OpenID

Introduction to OpenID Connect

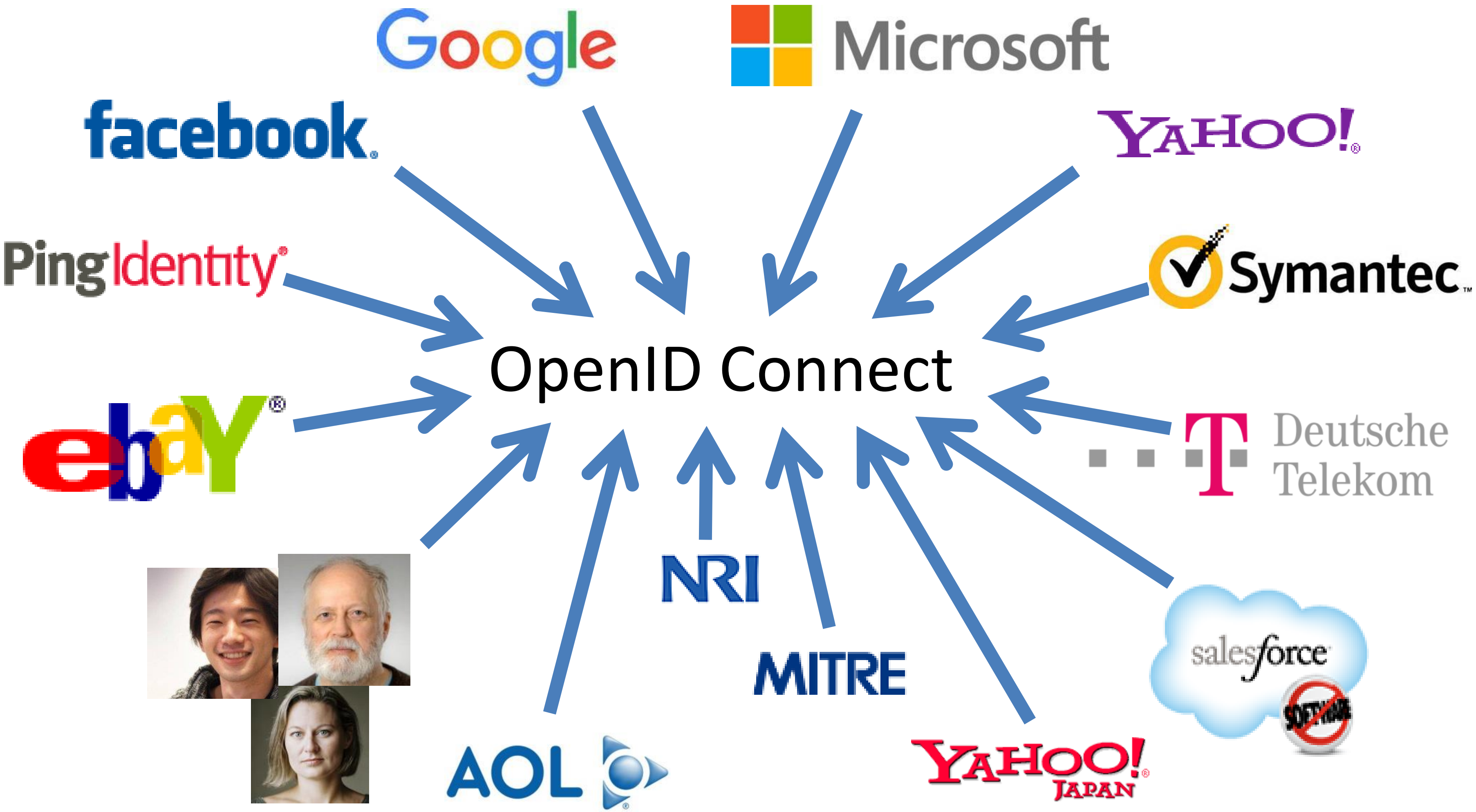
April 30, 2019

Michael B. Jones

Identity Standards Architect – Microsoft

Working Together

OpenID



OpenID Connect

What is OpenID Connect?

- Simple identity layer on top of OAuth 2.0
- Enables Relying Parties (RPs) to verify identity of end-user
- Enables RPs to obtain basic profile info
- REST/JSON interfaces → low barrier to entry
- Described at https://openid.net/connect/

You’re Probably Already Using OpenID Connect!

- If you have an Android phone or log in at AOL, Deutsche Telekom, Google, Microsoft, NEC, NTT, Salesforce, Softbank, Symantec, Verizon, or Yahoo! Japan, you’re already using OpenID Connect
  - Many other sites and apps large and small also use OpenID Connect

OpenID Connect Range

- Spans use cases, scenarios
  - Internet, Enterprise, Mobile, Cloud
- Spans security & privacy requirements
  - From non-sensitive information to highly secure
- Spans sophistication of claims usage
  - From basic default claims to specific requested claims to collecting claims from multiple sources
- Maximizes simplicity of implementations
  - Uses existing IETF specs: OAuth 2.0, JWT, etc.
  - Lets you build only the pieces you need

Numerous Awards

- OpenID Connect won 2012 European Identity Award for Best Innovation/New Standard
  - http://openid.net/2012/04/18/openid-connect-wins-2012-european-identity-and-cloud-award/
- OAuth 2.0 won in 2013
- JSON Web Token (JWT) & JOSE won in 2014
- OpenID Certification program won 2018 Identity Innovation Award
- OpenID Certification program won 2018 European Identity Award

Presentation Overview

- Introduction
- Design Philosophy
- Timeline
- A Look Under the Covers
- Overview of OpenID Connect Specs
- More OpenID Connect Specs
- OpenID Certification
- Resources

Design Philosophy

OpenID

Keep Simple Things Simple

Make Complex Things Possible

Keep Simple Things Simple

UserInfo endpoint for simple claims about user

Designed to work well on mobile phones

How We Made It Simple

- Built on OAuth 2.0
- Uses JavaScript Object Notation (JSON)
- You can build only the pieces that you need

- *Goal: Easy implementation on all modern development platforms*

Make Complex Things Possible

- Encrypted Claims
- Aggregated Claims
- Distributed Claims

Key Differences from OpenID 2.0

OpenID

- Support for native client applications
- Identifiers using e-mail address format
- UserInfo endpoint for simple claims about user
- Designed to work well on mobile phones
- Uses JSON/REST, rather than XML
- Support for encryption and higher LOAs
- Support for distributed and aggregated claims
- Support for session management, including logout
- Support for self-issued identity providers

OpenID Connect Timeline

- Artifact Binding working group formed, Mar 2010
- Major design issues closed at IIW, May 2011
  - Result branded “OpenID Connect”
- 5 rounds of interop testing between 2011 and 2013
  - Specifications refined after each round of interop testing
- Won Best New Standard award at EIC, April 2012
- Final specifications approved, February 2014
- Errata set 1 approved November 2014
- Form Post Response Mode spec approved, April 2015
- OpenID 2.0 to Connect Migration spec approved, April 2015
- OpenID Provider Certification launched, April 2015
- OpenID Federation spec work begun, July 2016
- Relying Party Certification launched, December 2016
- Logout Implementer’s Drafts approved, March 2017
- OpenID Certification program won awards in March 2018 and April 2018
- OpenID Connect for Identity Assurance spec work begun, March 2019

A Look Under the Covers

OpenID

- ID Token
- Claims Requests
- UserInfo Claims
- Example Protocol Messages

ID Token

- JWT representing logged-in session
- Claims:
  - `iss` – Issuer
  - `sub` – Identifier for subject (user)
  - `aud` – Audience for ID Token
  - `iat` – Time token was issued
  - `exp` – Expiration time
  - `nonce` – Mitigates replay attacks

ID Token Claims Example

OpenID

```
{
 "iss": "https://server.example.com",
 "sub": "248289761001",
 "aud": "0acf77d4-b486-4c99-bd76-074ed6a64ddf",
 "iat": 1311280970,
 "exp": 1311281970,
 "nonce": "n-0S6_WzA2Mj"
}
```

Claims Requests

- Basic requests made using OAuth scopes:
  - `openid` – Declares request is for OpenID Connect
  - `profile` – Requests default profile info
  - `email` – Requests email address & verification status
  - `address` – Requests postal address
  - `phone` – Requests phone number & verification status
  - `offline_access` – Requests Refresh Token issuance
- Requests for individual claims can be made using JSON "`claims`" request parameter

UserInfo Claims

- `sub`
- `name`
- `given_name`
- `family_name`
- `middle_name`
- `nickname`
- `preferred_username`
- `profile`
- `picture`
- `website`
- `gender`
- `birthdate`
- `locale`
- `zoneinfo`
- `updated_at`
- `email`
- `email_verified`
- `phone_number`
- `phone_number_verified`
- `address`

UserInfo Claims Example

```
{
 "sub": "248289761001",
 "name": "Jane Doe",
 "given_name": "Jane",
 "family_name": "Doe",
 "email": "janedoe@example.com",
 "email_verified": true,
 "picture": "http://example.com/janedoe/me.jpg"
}
```

Authorization Request Example

```
https://server.example.com/authorize
 ?response_type=id_token%20token
 &client_id=0acf77d4-b486-4c99-bd76-074ed6a64ddf
 &redirect_uri=https%3A%2F%2Fclient.example.com%2Fcb
 &scope=openid%20profile
 &state=af0ifjsldkj
 &nonce=n-0S6_WzA2Mj
```

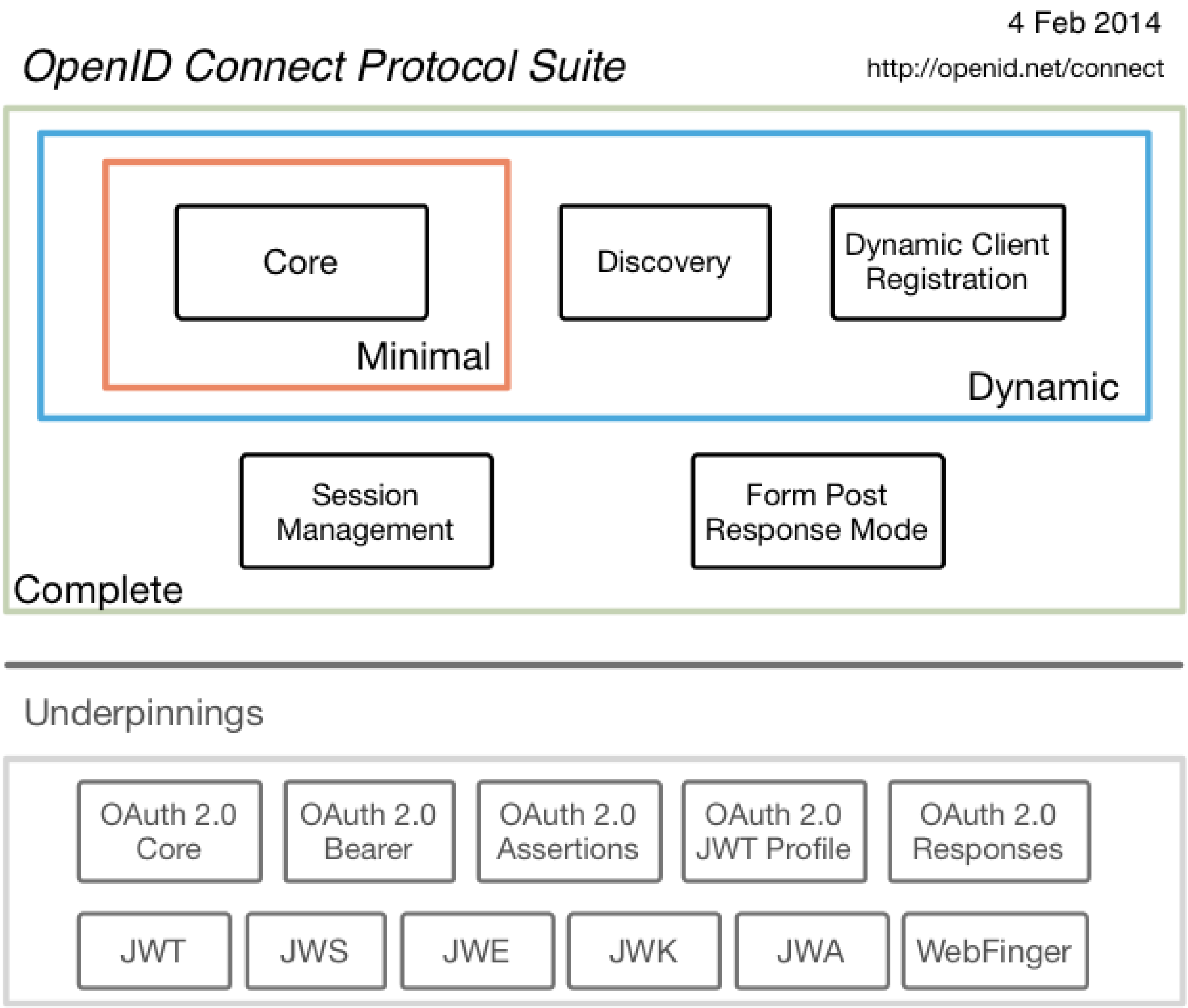
Authorization Response Example

```
HTTP/1.1 302 Found
Location: https://client.example.com/cb
 #access_token=mF_9.B5f-4.1JqM
 &token_type=bearer
 &id_token=eyJhbGzI1NiJ9.eyJz9Glnw9J.F9-V4IvQ0Z
 &expires_in=3600
 &state=af0ifjsldkj
```

UserInfo Request Example

```
GET /userinfo HTTP/1.1
Host: server.example.com
Authorization: Bearer mF_9.B5f-4.1JqM
```

OpenID Connect Specs Overview



4 Feb 2014

OpenID Connect Protocol Suite

http://openid.net/connect

Core

Minimal

Discovery

Dynamic Client Registration

Dynamic

Session Management

Form Post Response Mode

Complete

Underpinnings

OAuth 2.0 Core

OAuth 2.0 Bearer

OAuth 2.0 Assertions

OAuth 2.0 JWT Profile

OAuth 2.0 Responses

JWT

JWS

JWE

JWK

JWA

WebFinger

Additional Final Specifications (1 of 2)

- OpenID 2.0 to OpenID Connect Migration
  - Defines how to migrate from OpenID 2.0 to OpenID Connect
    - Has OpenID Connect identity provider also return OpenID 2.0 identifier, enabling account migration
  - http://openid.net/specs/openid-connect-migration-1_0.html
  - Completed April 2015
  - Google shut down OpenID 2.0 support in April 2015
  - Yahoo, others also plan to replace OpenID 2.0 with OpenID Connect

Additional Final Specifications (2 of 2)

- OAuth 2.0 Form Post Response Mode
  - Defines how to return OAuth 2.0 Authorization Response parameters (including OpenID Connect Authentication Response parameters) using HTML form values auto-submitted by the User Agent using HTTP POST
  - A “form post” binding, like SAML and WS-Federation
    - An alternative to fragment encoding
  - http://openid.net/specs/oauth-v2-form-post-response-mode-1_0.html
  - Completed April 2015
  - In production use by Microsoft, Ping Identity

Session Management / Logout (works in progress)

OpenID

- Three approaches specified by the working group:
  - Session Management
    - https://openid.net/specs/openid-connect-session-1_0.html
    - Uses HTML5 postMessage to communicate state change messages between OP and RP iframes
  - Front-Channel Logout
    - https://openid.net/specs/openid-connect-frontchannel-1_0.html
    - Uses HTTP GET to load image or iframe, triggering logout (similar to SAML, WS-Federation)
  - Back-Channel Logout
    - https://openid.net/specs/openid-connect-backchannel-1_0.html
    - Server-to-communication not using the browser
    - Can be used by native applications, which have no active browser
- All support multiple logged in sessions from OP at RP
- Unfortunately, no one approach best for all use cases
- Certification tests being developed
  - WG plans to test multiple implementations before making specs Final

Federation Specification (work in progress)

OpenID

- OpenID Connect Federation specification
  - – https://openid.net/specs/openid-connect-federation-1_0.html
- Enables establishment and maintenance of multi-party federations using OpenID Connect
- Defines hierarchical JSON-based metadata structures for federation participants
- Implementer's Draft status reached
- Substantial changes since then
  - – ***Please review!***

Identity Assurance Specification (work in progress)

OpenID

- OpenID Connect for Identity Assurance
  - https://openid.net/specs/openid-connect-4-identity-assurance.html
- Representation for verified person data
  - Enables legal compliance for some use cases
- New specification by Torsten Lodderstedt
  - ***Please review!***

What is OpenID Certification?

- Enables OpenID Connect and FAPI implementations to be certified as meeting the requirements of defined conformance profiles
  - Goal is to make high-quality, secure, interoperable OpenID Connect implementations the norm
- An OpenID Certification has two components:
  - Technical evidence of conformance resulting from testing
  - Legal statement of conformance
- Certified implementations can use the “OpenID Certified” logo

What value does certification provide?

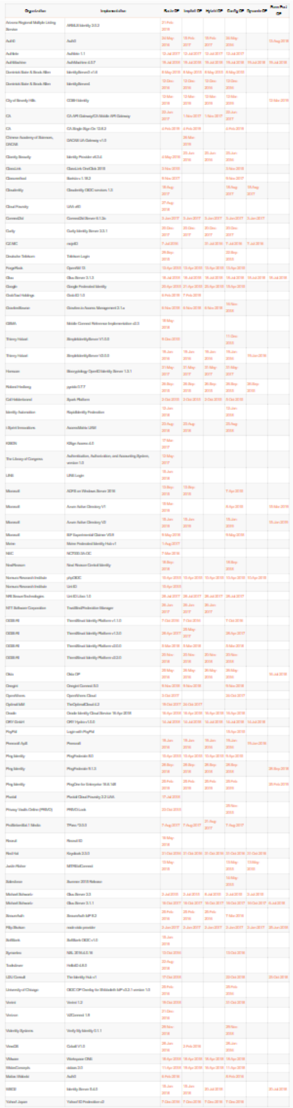
- Technical:
  - Certification testing gives confidence that things will “just work”
  - No custom code required to integrate with implementation
  - Better for all parties
  - Relying parties explicitly asking identity providers to get certified
- Business:
  - Enhances reputation of organization and implementation
  - Shows that organization is taking interop seriously
  - Customers may choose certified implementations over others

OpenID Connect Certification Profiles

- Six conformance profiles of OpenID Providers:
  - Basic OpenID Provider
  - Implicit OpenID Provider
  - Hybrid OpenID Provider
  - OpenID Provider Publishing Configuration Information
  - Dynamic OpenID Provider
  - Form Post OpenID Provider
- Six corresponding conformance profiles of OpenID Relying Parties:
  - Basic Relying Party
  - Implicit Relying Party
  - Hybrid Relying Party
  - Relying Party Publishing Configuration Information
  - Dynamic Relying Party
  - Form Post Relying Party

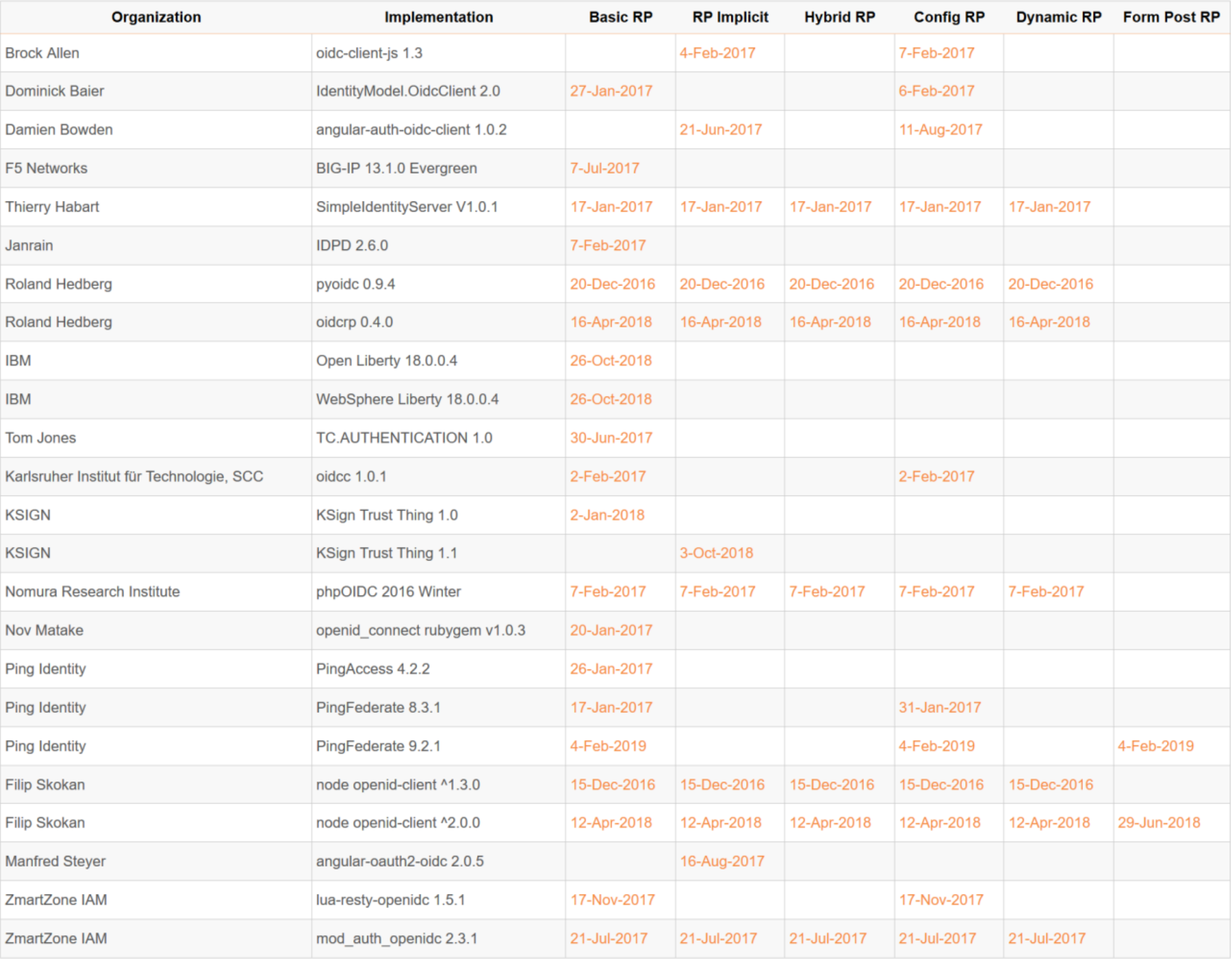
OpenID Connect OP Certifications

- OpenID Provider certifications at https://openid.net/certification/#OPs
  - 281 profiles certified for 91 implementations by 74 organizations
- Recent additions:
  - Arizona Regional Multiple Listing Service, City of Beverly Hills, CA, Chinese Academy of Sciences, GrabTaxi Holdings, Microsoft, Ping Identity, SoftBank
- Each entry link to zip file with test logs and signed legal statement
  - ***Test results available for public inspection***



OpenID Connect RP Certifications

- Relying Party certifications at https://openid.net/certification/#RPs
  - 65 profiles certified for 26 implementations by 18 organizations
- Recent additions:
  - IBM, Ping Identity



Organization	Implementation	Basic RP	RP Implicit	Hybrid RP	Config RP	Dynamic RP	Form Post RP
Brock Allen	oidc-client-js 1.3		4-Feb-2017		7-Feb-2017		
Dominick Baier	IdentityModel.OidcClient 2.0	27-Jan-2017			6-Feb-2017		
Damien Bowden	angular-auth-oidc-client 1.0.2		21-Jun-2017		11-Aug-2017		
F5 Networks	BIG-IP 13.1.0 Evergreen	7-Jul-2017					
Thierry Habart	SimpleIdentityServer V1.0.1	17-Jan-2017	17-Jan-2017	17-Jan-2017	17-Jan-2017	17-Jan-2017	
Janrain	IDPD 2.6.0	7-Feb-2017					
Roland Hedberg	pyoidc 0.9.4	20-Dec-2016	20-Dec-2016	20-Dec-2016	20-Dec-2016	20-Dec-2016	
Roland Hedberg	oidcrp 0.4.0	16-Apr-2018	16-Apr-2018	16-Apr-2018	16-Apr-2018	16-Apr-2018	
IBM	Open Liberty 18.0.0.4	26-Oct-2018					
IBM	WebSphere Liberty 18.0.0.4	26-Oct-2018					
Tom Jones	TC.AUTHENTICATION 1.0	30-Jun-2017					
Karlsruher Institut für Technologie, SCC	oidcc 1.0.1	2-Feb-2017			2-Feb-2017		
KSIGN	KSign Trust Thing 1.0	2-Jan-2018					
KSIGN	KSign Trust Thing 1.1		3-Oct-2018				
Nomura Research Institute	phpOIDC 2016 Winter	7-Feb-2017	7-Feb-2017	7-Feb-2017	7-Feb-2017	7-Feb-2017	
Nov Matake	openid_connect rubygem v1.0.3	20-Jan-2017					
Ping Identity	PingAccess 4.2.2	26-Jan-2017					
Ping Identity	PingFederate 8.3.1	17-Jan-2017			31-Jan-2017		
Ping Identity	PingFederate 9.2.1	4-Feb-2019			4-Feb-2019		4-Feb-2019
Filip Skokan	node openid-client ^1.3.0	15-Dec-2016	15-Dec-2016	15-Dec-2016	15-Dec-2016	15-Dec-2016	
Filip Skokan	node openid-client ^2.0.0	12-Apr-2018	12-Apr-2018	12-Apr-2018	12-Apr-2018	12-Apr-2018	29-Jun-2018
Manfred Steyer	angular-oauth2-oidc 2.0.5		16-Aug-2017				
ZmartZone IAM	lua-resty-openidc 1.5.1	17-Nov-2017			17-Nov-2017		
ZmartZone IAM	mod_auth_openidc 2.3.1	21-Jul-2017	21-Jul-2017	21-Jul-2017	21-Jul-2017	21-Jul-2017	

A Very International Effort

- European programmers developed and operate the certification test suites:
  - Roland Hedberg, Sweden
  - Joseph Heenan, UK
  - Serkan Özkan, Turkey
  - Tomas Pazderka, Czech Republic
  - Filip Skokan, Czech Republic
  - Hans Zandbelt, Netherlands
- OpenID Connect leadership also very international:
  - Nat Sakimura, Japan
  - John Bradley, Chile
  - Michael Jones, United States

Use of Self-Certification

OpenID

- OpenID Certification uses self-certification
  - Party seeking certification does the testing
  - (rather than paying a 3rd party to do the testing)
- Simpler, quicker, less expensive, more scalable than 3rd party certification
- Results are nonetheless trustworthy because
  - Testing logs are made available for public scrutiny
  - Organization puts its reputation on the line by making a public declaration that its implementation conforms to the profile being certified to

How does OpenID Certification work?

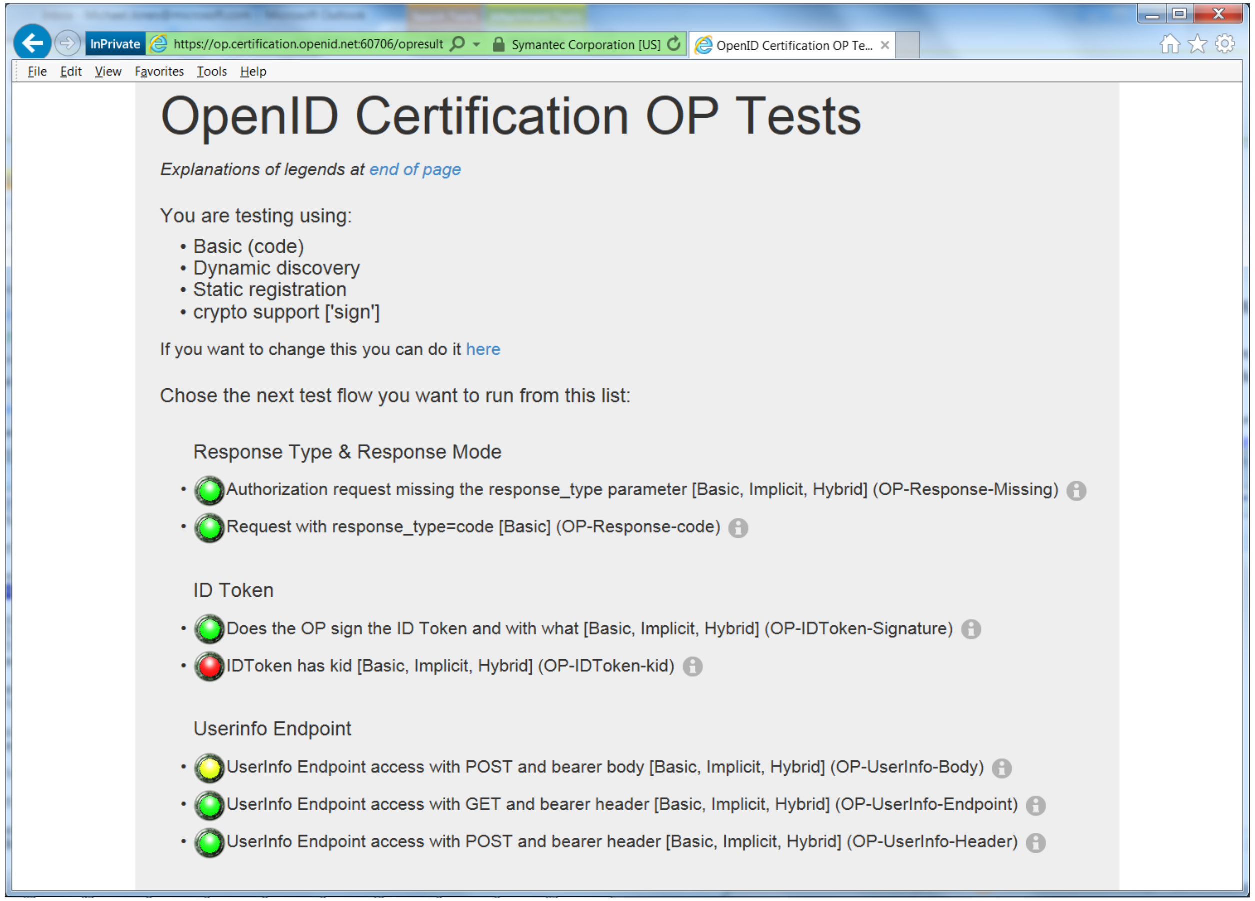
- Organization decides what profiles it wants to certify to
  - For instance, “Basic OP”, “Config OP”, and “Dynamic OP”
- Runs conformance tests publicly available at https://op.certification.openid.net/ or https://rp.certification.openid.net/
- Once all tests for a profile pass, organization submits certification request to OpenID Foundation containing:
  - Logs from all tests for the profile
  - Signed legal declaration that implementation conforms to the profile
- Organization pays certification fee (for profiles not in pilot mode)
- OpenID Foundation verifies application is complete and grants certification
- OIDF lists certification at https://openid.net/certification/ and registers it in OIXnet at http://oixnet.org/openid-certifications/

What does certification cost?

- Not a profit center for the OpenID Foundation
  - Fees there to help cover costs of operating certification program
- Member price
  - $200 for Connect, $500 for FAPI
    - Connect price will change to $500 in June 2019
- Non-member price
  - $999 for Connect, $2,500 for FAPI
    - Connect price will change to $2,500 in June 2019
- New profiles in pilot mode are available to members for free
- Costs described at https://openid.net/certification/fees/

Example Testing Screen

OpenID


InPrivate
https://op.certification.openid.net:60706/opresult
Symantec Corporation [US]
OpenID Certification OP Te...
File Edit View Favorites Tools Help

OpenID Certification OP Tests

Explanations of legends at end of page

You are testing using:

- Basic (code)
- Dynamic discovery
- Static registration
- crypto support ['sign']

If you want to change this you can do it here

Chose the next test flow you want to run from this list:

Response Type & Response Mode

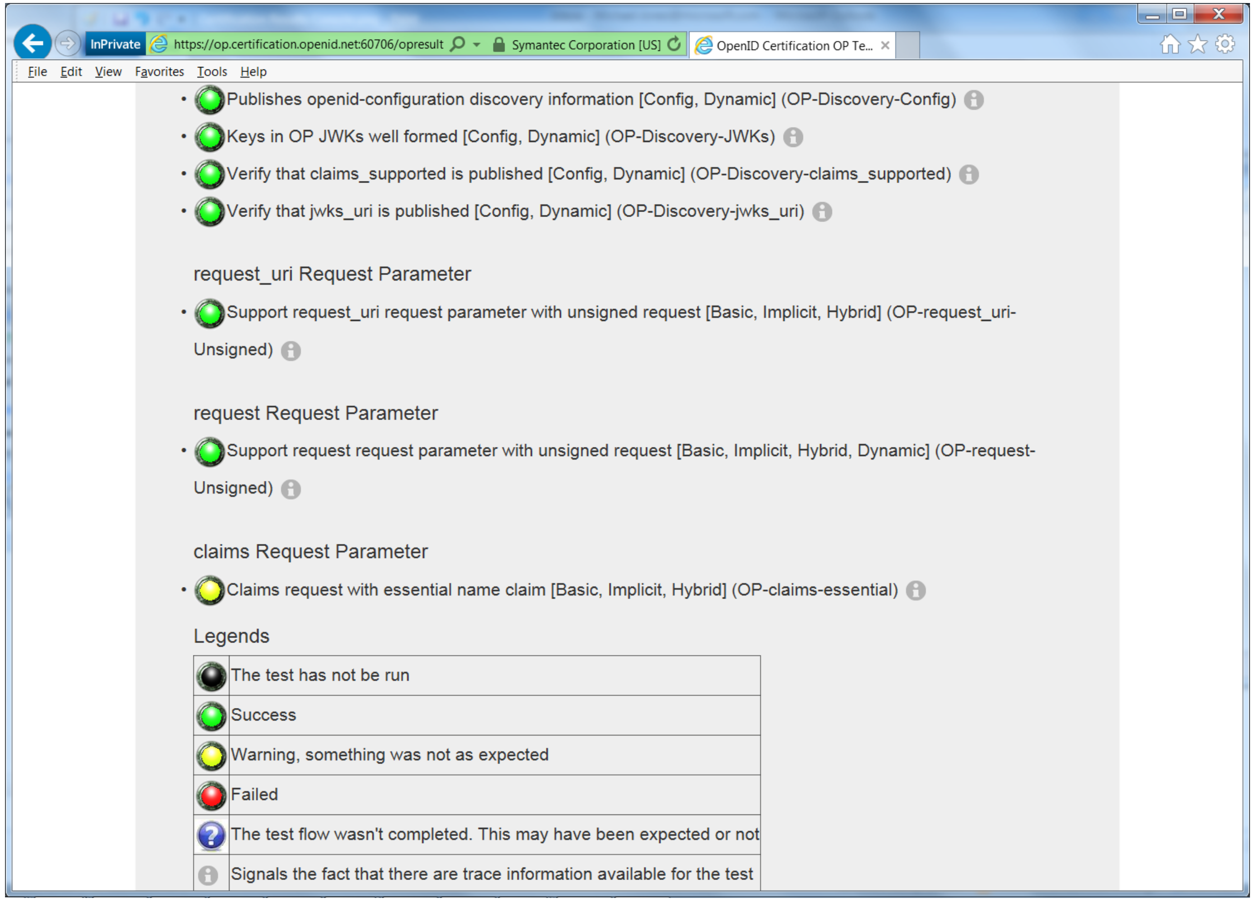
- 🟢 Authorization request missing the response_type parameter [Basic, Implicit, Hybrid] (OP-Response-Missing) ⓘ
- 🟢 Request with response_type=code [Basic] (OP-Response-code) ⓘ

ID Token

- 🟢 Does the OP sign the ID Token and with what [Basic, Implicit, Hybrid] (OP-IDToken-Signature) ⓘ
- 🔴 IDToken has kid [Basic, Implicit, Hybrid] (OP-IDToken-kid) ⓘ

Userinfo Endpoint

- 🟡 UserInfo Endpoint access with POST and bearer body [Basic, Implicit, Hybrid] (OP-UserInfo-Body) ⓘ
- 🟢 UserInfo Endpoint access with GET and bearer header [Basic, Implicit, Hybrid] (OP-UserInfo-Endpoint) ⓘ
- 🟢 UserInfo Endpoint access with POST and bearer header [Basic, Implicit, Hybrid] (OP-UserInfo-Header) ⓘ


InPrivate
https://op.certification.openid.net:60706/opresult
Symantec Corporation [US]
OpenID Certification OP Te...
File Edit View Favorites Tools Help

- 🟢 Publishes openid-configuration discovery information [Config, Dynamic] (OP-Discovery-Config) ⓘ
- 🟢 Keys in OP JWKs well formed [Config, Dynamic] (OP-Discovery-JWKs) ⓘ
- 🟢 Verify that claims_supported is published [Config, Dynamic] (OP-Discovery-claims_supported) ⓘ
- 🟢 Verify that jwks_uri is published [Config, Dynamic] (OP-Discovery-jwks_uri) ⓘ

request_uri Request Parameter

- 🟢 Support request_uri request parameter with unsigned request [Basic, Implicit, Hybrid] (OP-request_uri-Unsigned) ⓘ

request Request Parameter

- 🟢 Support request request parameter with unsigned request [Basic, Implicit, Hybrid, Dynamic] (OP-request-Unsigned) ⓘ

claims Request Parameter

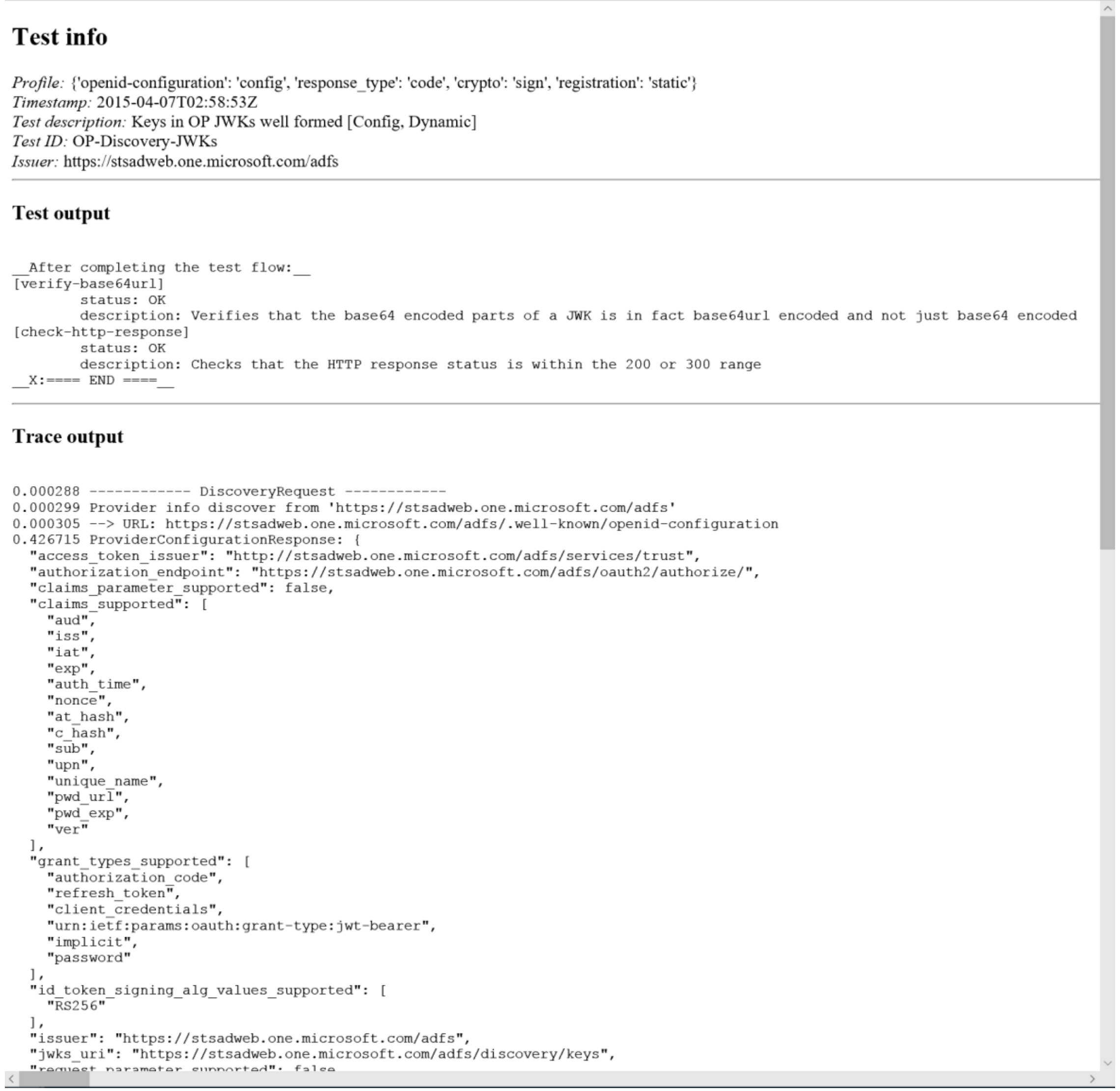
- 🟡 Claims request with essential name claim [Basic, Implicit, Hybrid] (OP-claims-essential) ⓘ

Legends

⚫	The test has not be run
🟢	Success
🟡	Warning, something was not as expected
🔴	Failed
?	The test flow wasn't completed. This may have been expected or not
ⓘ	Signals the fact that there are trace information available for the test

Log from a Conformance Test

OpenID



Test info

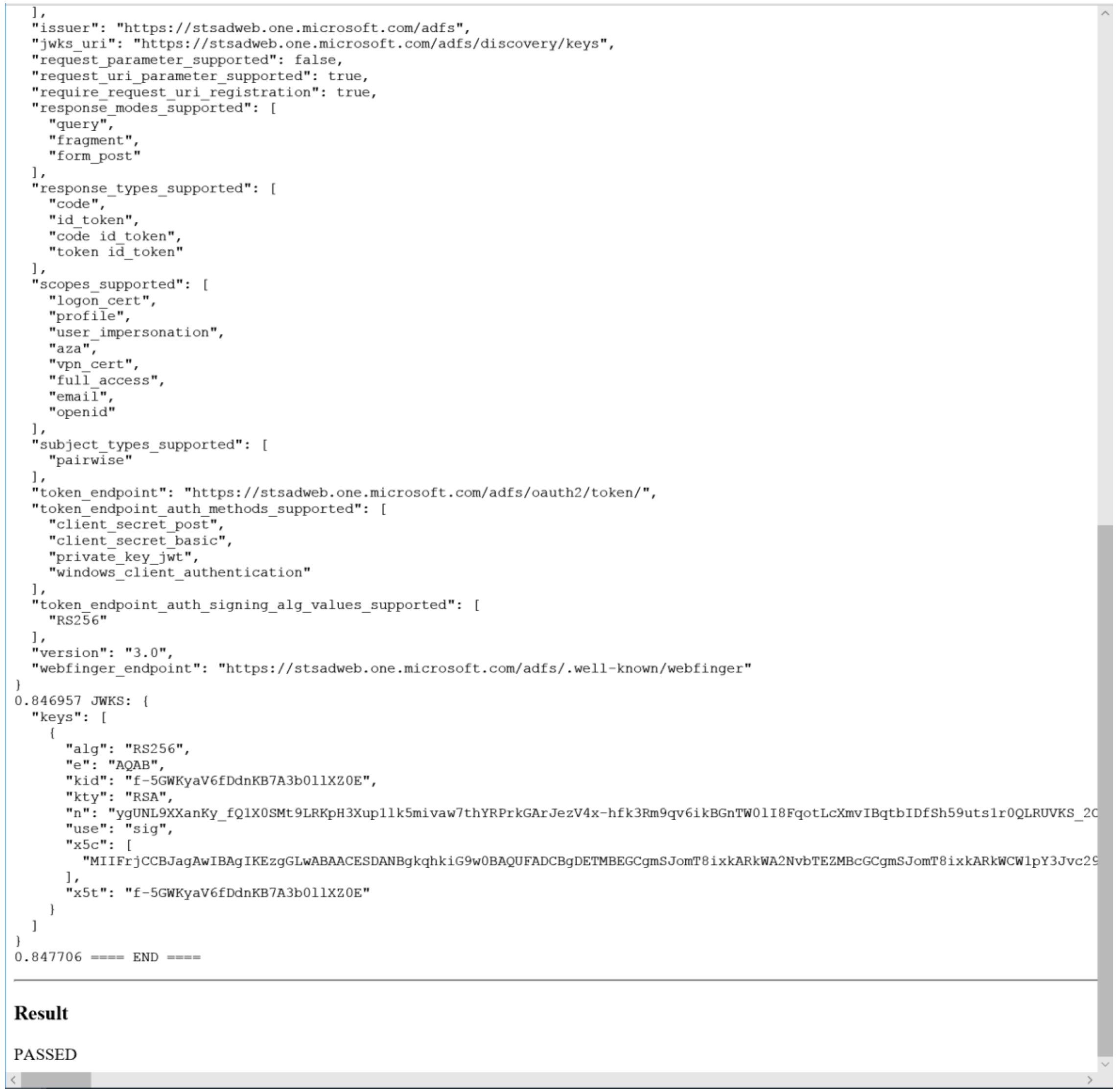
Profile: {'openid-configuration': 'config', 'response_type': 'code', 'crypto': 'sign', 'registration': 'static'}
Timestamp: 2015-04-07T02:58:53Z
Test description: Keys in OP JWKs well formed [Config, Dynamic]
Test ID: OP-Discovery-JWKs
Issuer: https://stsadweb.one.microsoft.com/adfs

Test output

```
__After completing the test flow:__
[verify-base64url]
        status: OK
        description: Verifies that the base64 encoded parts of a JWK is in fact base64url encoded and not just base64 encoded
[check-http-response]
        status: OK
        description: Checks that the HTTP response status is within the 200 or 300 range
__X:==== END ====__
```

Trace output

```
0.000288 ------------ DiscoveryRequest ------------
0.000299 Provider info discover from 'https://stsadweb.one.microsoft.com/adfs'
0.000305 --> URL: https://stsadweb.one.microsoft.com/adfs/.well-known/openid-configuration
0.426715 ProviderConfigurationResponse: {
  "access_token_issuer": "http://stsadweb.one.microsoft.com/adfs/services/trust",
  "authorization_endpoint": "https://stsadweb.one.microsoft.com/adfs/oauth2/authorize/",
  "claims_parameter_supported": false,
  "claims_supported": [
    "aud",
    "iss",
    "iat",
    "exp",
    "auth_time",
    "nonce",
    "at_hash",
    "c_hash",
    "sub",
    "upn",
    "unique_name",
    "pwd_url",
    "pwd_exp",
    "ver"
  ],
  "grant_types_supported": [
    "authorization_code",
    "refresh_token",
    "client_credentials",
    "urn:ietf:params:oauth:grant-type:jwt-bearer",
    "implicit",
    "password"
  ],
  "id_token_signing_alg_values_supported": [
    "RS256"
  ],
  "issuer": "https://stsadweb.one.microsoft.com/adfs",
  "jwks_uri": "https://stsadweb.one.microsoft.com/adfs/discovery/keys",
  "request_parameter_supported": false
```



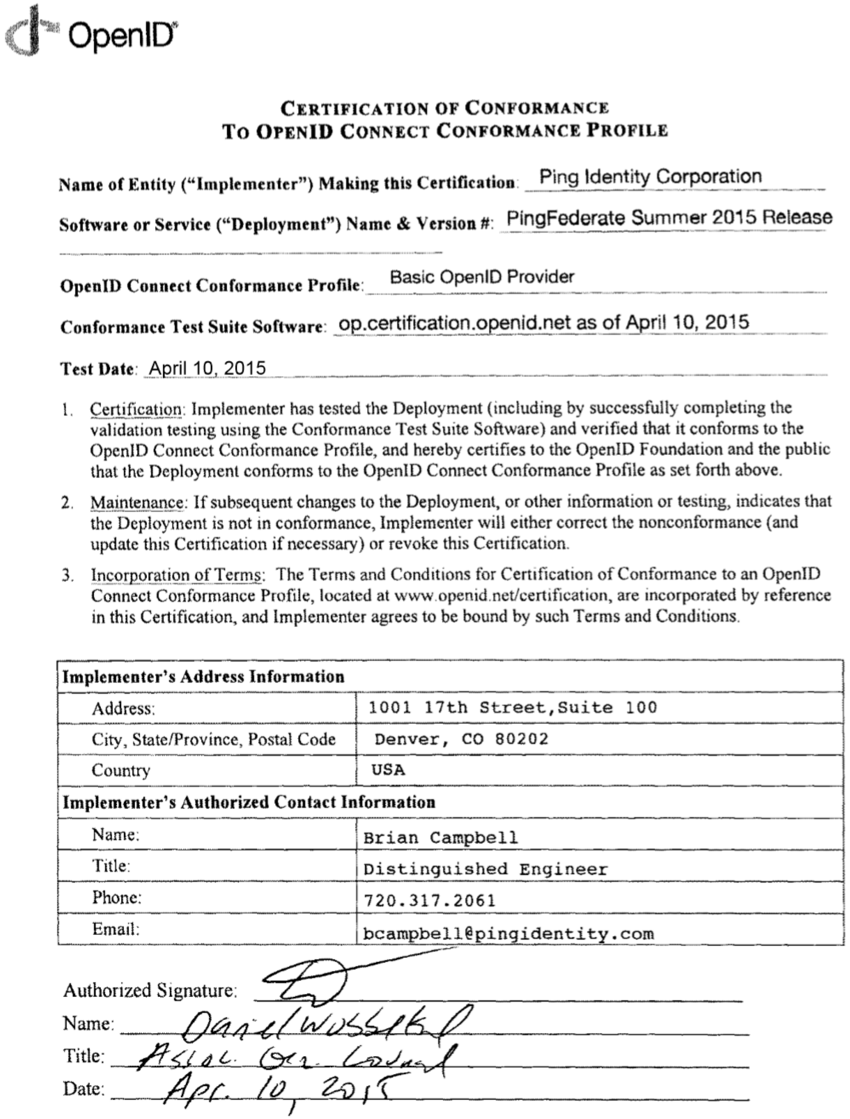
```
  ],
  "issuer": "https://stsadweb.one.microsoft.com/adfs",
  "jwks_uri": "https://stsadweb.one.microsoft.com/adfs/discovery/keys",
  "request_parameter_supported": false,
  "request_uri_parameter_supported": true,
  "require_request_uri_registration": true,
  "response_modes_supported": [
    "query",
    "fragment",
    "form_post"
  ],
  "response_types_supported": [
    "code",
    "id_token",
    "code id_token",
    "token id_token"
  ],
  "scopes_supported": [
    "logon_cert",
    "profile",
    "user_impersonation",
    "aza",
    "vpn_cert",
    "full_access",
    "email",
    "openid"
  ],
  "subject_types_supported": [
    "pairwise"
  ],
  "token_endpoint": "https://stsadweb.one.microsoft.com/adfs/oauth2/token/",
  "token_endpoint_auth_methods_supported": [
    "client_secret_post",
    "client_secret_basic",
    "private_key_jwt",
    "windows_client_authentication"
  ],
  "token_endpoint_auth_signing_alg_values_supported": [
    "RS256"
  ],
  "version": "3.0",
  "webfinger_endpoint": "https://stsadweb.one.microsoft.com/adfs/.well-known/webfinger"
}
0.846957 JWKS: {
  "keys": [
    {
      "alg": "RS256",
      "e": "AQAB",
      "kid": "f-5GWKyaV6fDdnKB7A3b0llXZ0E",
      "kty": "RSA",
      "n": "ygUNL9XXanKy_fQ1X0SMt9LRKpH3Xup1lk5mivaw7thYRPrkGArJezV4x-hfk3Rm9qv6ikBGnTW0lI8FqotLcXmvIBqtbIDfSh59uts1r0QLRUVKS_2C
      "use": "sig",
      "x5c": [
        "MIIFrjCCBJagAwIBAgIKEzgGLwABAACESDANBgkqhkiG9w0BAQUFADCBgDETMBEGCgmSJomT8ixkARkWA2NvbTEZMBcGCgmSJomT8ixkARkWCW1pY3Jvc29
      ],
      "x5t": "f-5GWKyaV6fDdnKB7A3b0llXZ0E"
    }
  ]
}
0.847706 ==== END ====
```

Result

PASSED

Certification of Conformance

- Legal statement by certifier stating:
  - Who is certifying
  - What software
  - When tested
  - Profile tested
- Commits reputation of certifying organization to validity of results



OpenID

Certification of Conformance
To OpenID Connect Conformance Profile

Name of Entity ("Implementer") Making this Certification: Ping Identity Corporation

Software or Service ("Deployment") Name & Version #: PingFederate Summer 2015 Release

OpenID Connect Conformance Profile: Basic OpenID Provider

Conformance Test Suite Software: op.certification.openid.net as of April 10, 2015

Test Date: April 10, 2015

1. Certification: Implementer has tested the Deployment (including by successfully completing the validation testing using the Conformance Test Suite Software) and verified that it conforms to the OpenID Connect Conformance Profile, and hereby certifies to the OpenID Foundation and the public that the Deployment conforms to the OpenID Connect Conformance Profile as set forth above.
2. Maintenance: If subsequent changes to the Deployment, or other information or testing, indicates that the Deployment is not in conformance, Implementer will either correct the nonconformance (and update this Certification if necessary) or revoke this Certification.
3. Incorporation of Terms: The Terms and Conditions for Certification of Conformance to an OpenID Connect Conformance Profile, located at www.openid.net/certification, are incorporated by reference in this Certification, and Implementer agrees to be bound by such Terms and Conditions.

Implementer's Address Information	
Address:	1001 17th Street,Suite 100
City, State/Province, Postal Code	Denver, CO 80202
Country	USA
Implementer's Authorized Contact Information	
Name:	Brian Campbell
Title:	Distinguished Engineer
Phone:	720.317.2061
Email:	bcampbell@pingidentity.com

Authorized Signature:

Name: Daniel Wossikal

Title: Assoc. Gen. Counsel

Date: Apr. 10, 2015

How does certification relate to interop testing?

- OpenID Connect held 5 rounds of interop testing – see http://osis.idcommons.net/
  - Each round improved implementations and specs
  - By the numbers: 20 implementations, 195 members of interop list, > 1000 messages exchanged
- With interop testing, by design, participants can ignore parts of the specs
- Certification raises the bar:
  - Defines set of conformance profiles that certified implementations meet
  - Assures interop across full feature sets in profiles

Can I use the certification sites for interop testing?

- Yes – please do!
- The OpenID Foundation is committed to keeping the conformance test sites up and available for free to all
- Many projects using conformance testing for regression testing
  - Once everything passes, you're ready for certification!
- Test software is open source Python using Apache 2.0 license
  - Some projects have deployed private instances for internal testing
  - Available as a Docker container

Favorite Comments on OpenID Certification

OpenID

- Eve Maler – VP of Innovation at ForgeRock
  - “You made it as simple as possible so every interaction added value.”
- Jaromír Talíř – CZ.NIC
  - “We used and still are using certification platform mainly as testing tool for our IdP. Thanks to this tool, we have fixed enormous number of bugs in our platform an even some bugs in the underlying library.”
- Brian Campbell – Distinguished Engineer at Ping Identity
  - “The process has allowed us to tighten up our implementation and improve on the already solid interoperability of our offerings in the OpenID Connect ecosystem.”
- William Denniss – Google
  - “We have built the RP tests into the continuous-integration testing pipeline for AppAuth.”

What's next for OpenID Certification?

OpenID

- Additional Connect profiles being developed:
  - Third Party Initiated Login
  - RP-Initiated Logout, Session Management, Front-Channel Logout, Back-Channel Logout
  - Refresh Token Behaviors
- Additional FAPI profiles being developed:
  - FAPI RP
  - FAPI CIBA OP
  - FAPI CIBA RP
- Certification for additional specifications is anticipated:
  - E.g., HEART, MODRNA, iGov, EAP, etc.

OpenID Certification Call to Action

- Certify your OpenID Connect and FAPI implementations now
- Help us test the new tests
- Join the OpenID Foundation and/or the OpenID Connect working group

OpenID Connect Resources

OpenID

- OpenID Connect
  - https://openid.net/connect/
- Frequently Asked Questions
  - https://openid.net/connect/faq/
- Working Group Mailing List
  - https://lists.openid.net/mailman/listinfo/openid-specs-ab
- OpenID Certification Program
  - https://openid.net/certification/
- Certified OpenID Connect Implementations Featured for Developers
  - https://openid.net/developers/certified/
- Mike Jones' Blog
  - http://self-issued.info/
- Nat Sakimura's Blog
  - http://nat.sakimura.org/
- John Bradley's Blog
  - http://www.thread-safe.com/

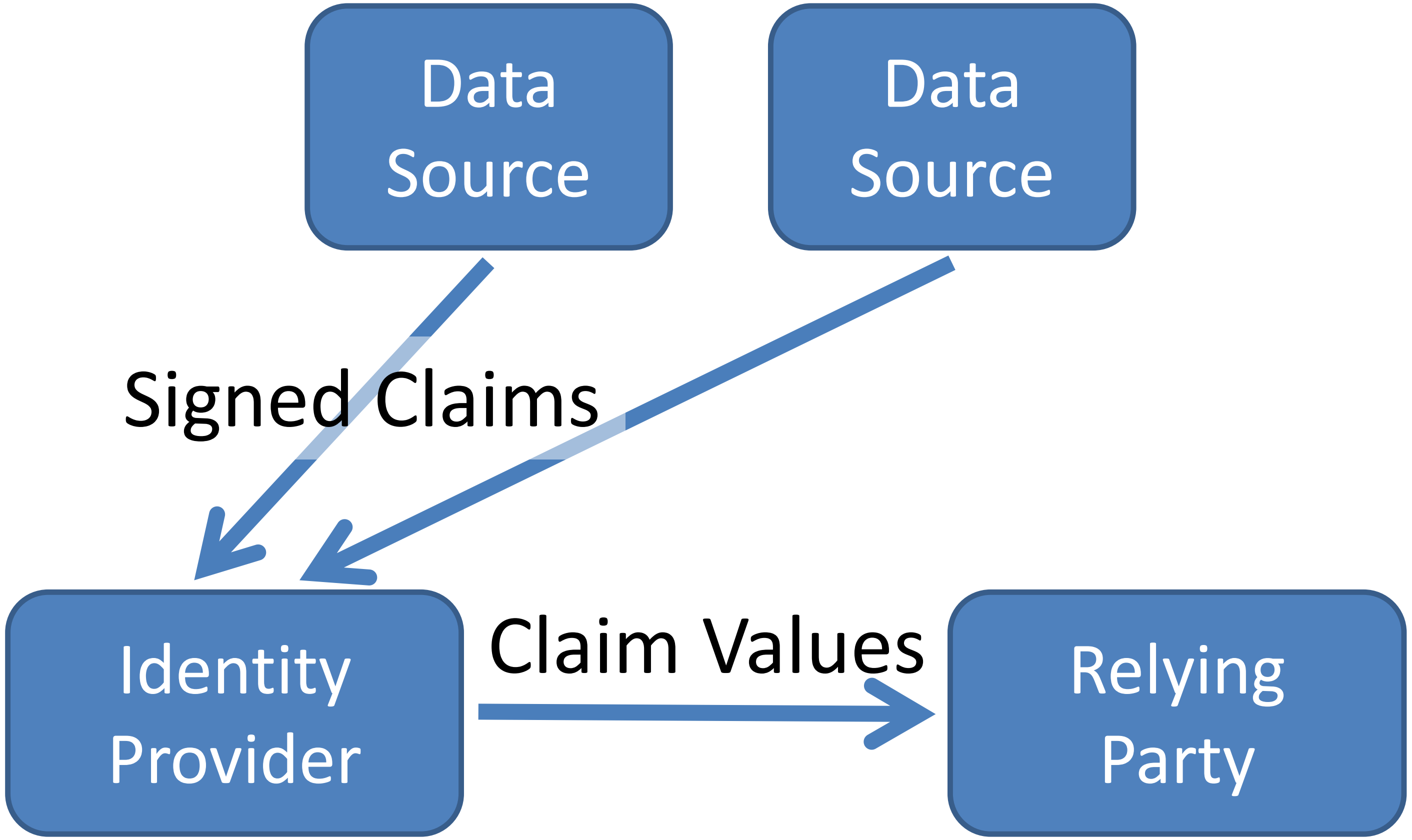
Open Conversation

OpenID

- How are you using OpenID Connect?
- What would you like the working group to know or do?

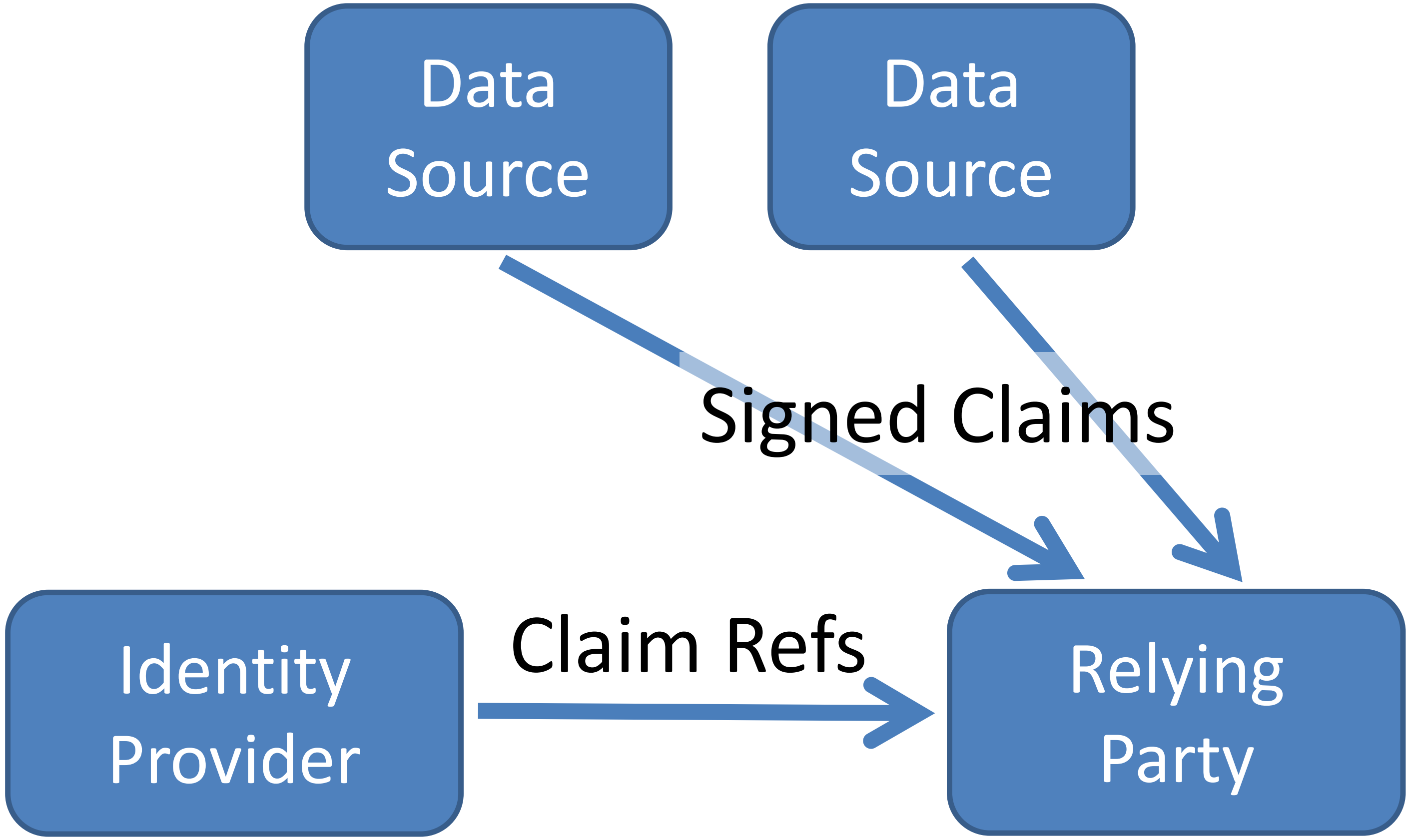
BACKUP SLIDES

Aggregated Claims



Distributed Claims

OpenID



Basic Client Implementer's Guide

- Single, simple, self-contained Web client spec
  - For clients using OAuth "code" flow
- All you need for Web server-based RP
  - Using pre-configured set of OPs

- http://openid.net/specs/openid-connect-basic-1_0.html

Implicit Client Implementer's Guide

- Single, simple, self-contained Web client spec
    - For clients using OAuth "implicit" flow
- All you need for user agent-based RPs
    - – Using pre-configured set of OPs

- http://openid.net/specs/openid-connect-implicit-1_0.html

Core Specification

OpenID

- Defines data formats and messages used for OpenID Connect authentication and claims
- http://openid.net/specs/openid-connect-core-1_0.html

Discovery & Registration

- Enables dynamic configurations in which sets of OPs and RPs are not pre-configured
  - Necessary for open deployments
- Discovery enables RPs to learn about OP endpoints
- Dynamic registration enables RPs to use OPs they don’t have pre-existing relationships with

- http://openid.net/specs/openid-connect-discovery-1_0.html
- http://openid.net/specs/openid-connect-registration-1_0.html

Session Management

- For OPs and RPs needing session management capabilities
  - Enables logout functionality
  - Enables account switching

- http://openid.net/specs/openid-connect-session-1_0.html

OAuth Response Types

- Defines and registers additional OAuth response types:
  - `id_token`
  - `none`
- And also defines and registers combinations of `code`, `token`, and `id_token` response types

- http://openid.net/specs/oauth-v2-multiple-response-types-1_0.html

Form Post Response Mode

OpenID

- Defines how to return OAuth 2.0 Authorization Response parameters using HTML form values auto-submitted by User Agent using HTTP POST
- http://openid.net/specs/oauth-v2-form-post-response-mode-1_0.html