OpenID Connect Overview

May 5, 2014

Michael B. Jones

Identity Standards Architect – Microsoft
OpenID  Working Group Members

• Key working group participants:
  – Nat Sakimura – Nomura Research Institute – Japan
  – John Bradley – Ping Identity – Chile
  – Breno de Medeiros – Google – US
  – Axel Nennker – Deutsche Telekom – Germany
  – Torsten Lodderstedt – Deutsche Telekom – Germany
  – Roland Hedberg – Umeå University – Sweden
  – Andreas Åkre Solberg – UNINETT – Norway
  – Chuck Mortimore – Salesforce – US
  – Brian Campbell – Ping Identity – US
  – George Fletcher – AOL – US
  – Justin Richer – Mitre – US
  – Nov Matake – Independent – Japan
  – Mike Jones – Microsoft – US

• By no means an exhaustive list!
OpenID Connect Intro

- Simple identity layer on top of OAuth 2.0
- Enables clients to verify identity of end-user
- Enables clients to obtain basic profile info
- REST/JSON interfaces → low barrier to entry
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 aggregated and distributed claims
• Maximizes simplicity of implementations
  – Uses existing IETF specs: OAuth 2.0, JWT, etc.
  – Lets you build only the pieces you need
Presentation Overview

- Introduction
- Design Philosophy
- A Look Under the Covers
- Overview of Connect Specs
- Timeline
- Next Steps
- Resources
UserInfo endpoint for simple claims about user

Designed to work well on mobile phones
How We Make It Simple

• Build on OAuth 2.0
• Use JavaScript Object Notation (JSON)
• Build only the pieces that you need

• Goal: Easy implementation on all modern development platforms
Complex Things Possible

- Encrypted Claims
- Aggregated Claims
- Distributed Claims
Key Diffs from OpenID 2.0

- 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
Connect Interop Status

- Fifth round of interop testing in progress
- Interop data at [http://osis.idcommons.net/](http://osis.idcommons.net/)
- By the numbers:
  - 20 implementations participating
  - 110 feature tests defined
  - 147 members of interop mailing list
Deployment Status

• Production deployments by:
  – Google
  – Microsoft
  – Deutsche Telekom
  – Ping Identity
  – AOL
  – Salesforce
  – Yahoo! Japan
  – Softbank
  – mixi

• Many more under way
A Look Under the Covers

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

```json
{
    "iss": "https://server.example.com",
    "sub": "248289761001",
    "aud": "0acf77d4-b486-4c99-bd76-074ed6a64ddf",
    "iat": 1311280970,
    "exp": 1311281970,
    "nonce": "n-0S6_WzA2Mj"
}
```
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
userInfoClaimsExample =
{
    "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=token%20id_token
&client_id=0acf77d4-b486-4c99-bd76-074ed6a64ddf
&redirect_uri=https%3A%2F%2Fclient.example.com%2Fcb
&scope=openid%20profile
&state=af0ifjs1dkj
&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
GET /userinfo?schema=openid HTTP/1.1
Host: server.example.com
Authorization: Bearer mF_9.B5f-4.1JqM
OpenID Connect Specs Overview

OpenID Connect Protocol Suite

4 Feb 2014
http://openid.net/connect

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
Timeline

- Artifact Binding working group formed, Mar 2010
- Major design issues closed at IIW, May 2011
  - Result branded “OpenID Connect”, May 2011
- Functionally complete specs, Jul 2011
- 2nd interop testing round, Sep-Nov 2011
- Simpler specs incorporating dev feedback, Oct 2011
- Published First Implementer’s Drafts, Dec 2011
- 3rd interop testing round, Feb 2012 to May 2012
- OpenID Connect won Best Innovation/New Standard award at EIC, April 2012
- Revised specs incorporating more feedback, June 2012
- 4th interop testing round, June 2012 to June 2013
- Second Implementers Drafts published, June 2013
- 5th interop testing round began, June 2013
- Final specifications approved, February 2014
Next Steps

• Continued deployment
• Continued interop testing
• Transition from OpenID 2.0 to OpenID Connect
  – Google, Yahoo!, etc. ending OpenID 2.0 support
• Finish session management, form post specs
• Self-certification program being developed
Self-Certification Program

- Goal to enable implementations to certify that they meet criteria defined by WG, OIDF
- Certification work will be done by party seeking certification – not 3rd party
- Plan is to develop certification test suite based on Roland Hedberg’s interop testing software
- Price TBD but will be low – intended to cover administrative costs – not be a profit center
Resources

• OpenID Connect
  – http://openid.net/connect/
• Frequently Asked Questions
  – http://openid.net/connect/faq/
• Working Group Mailing List
  – http://lists.openid.net/mailman/listinfo/openid-specs-ab
• Interop Wiki
  – http://osis.idcommons.net/
• Interop Mailing List
  – http://groups.google.com/group/openid-connect-interop
• Mike Jones’ Blog
  – http://self-issued.info/
• Nat Sakimura’s Blog
  – http://nat.sakimura.org/
• John Bradley’s Blog
  – http://www.thread-safe.com/
Aggregated Claims

Identity Provider

Signed Claims

Data Source

Claim Values

Relying Party
Distributed Claims

Identity Provider

Data Source

Signed Claims

Data Source

Claim Refs

Relying Party
Basic Client Profile

• 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](http://openid.net/specs/openid-connect-basic-1_0.html)
Implicit Client Profile

• 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](http://openid.net/specs/openid-connect-implicit-1_0.html)
• 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
Messages & Standard

- Messages spec defines data formats exchanged in OpenID Connect messages
- Standard spec is HTTP binding for Messages
  - (Basic and Implicit are profiles of Messages and Standard)
- Needed for OPs, native client apps, and RPs needing functionality not in Basic
  - E.g., requesting claims not in default UserInfo set

- http://openid.net/specs/openid-connect-messages-1_0.html
- http://openid.net/specs/openid-connect-standard-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](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](http://openid.net/specs/oauth-v2-multiple-response-types-1_0.html)
OpenID Form Post Response Mode

- 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](http://openid.net/specs/oauth-v2-form-post-response-mode-1_0.html)