Understanding JWT/CWT, OpenID and Related Ecosystem

Michael Jones, John Bradley
Aaron Parecki
JWT, OpenID Connect, CWT, and Verifiable Claims

Michael B. Jones – Microsoft and John Bradley – Yubico
W3C Workshop on Strong Authentication and Identity
December 10, 2018
JSON Web Token (JWT) – RFC 7519

• Representation of claims in JSON
• Can be signed with JSON Web Signature (JWS) – RFC 7515
• Can be encrypted with JSON Web Encryption (JWE) – RFC 7516
• Algorithms used extensible using IANA JOSE Algorithms Registry
  • For instance, ed25519 added and secp256k1 being added
• By design, does not use any form of JSON canonicalization
  • Base64url encodes values to maintain content integrity instead
• JWTs used by OpenID Connect, many other applications
ID Token Claims Example

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

• Simple identity layer on top of OAuth 2.0
• Enables RPs to verify identity of end-user
• Enables RPs to obtain basic profile info
• REST/JSON interfaces → low barrier to entry
• Described at http://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 and Verifiable Claims

• Aggregated and Distributed Claims
• Self-Issued Identities
• Representation of Claim Verification Information
OpenID Connect: Aggregated and Distributed Claims

• OpenID Connect Core §5.6.2
• Defines how JWTs can contain claims signed by others
  • Issuers of aggregated and distributed claims can be different than JWT issuer
• For example, credit score signed by credit agency and payment information signed by bank
• Aggregated claims pass 3rd party claims by value
• Distributed claims pass 3rd party claims by reference
OpenID Connect: Self-Issued Identities

• OpenID Connect Core §7
• Digital identity controlled directly by you
  • Backed by public/private key pair
  • Sometimes called “user-centric identity” or “self-sovereign identity”
• Claims in self-issued identities
  • Self-issued claims signed by you
  • Aggregated and distributed claims signed by 3rd parties
• Implementations in Japan and at Microsoft
OpenID Connect: Representation of Claim Verification Information

• Syntax for providing metadata about claims along with claims
  • For instance, saying that name, address, and payment info validated by a particular bank
    • At a particular time
    • In a particular jurisdiction
    • Under a particular legal framework

• Also ways of requesting claims with particular validation information
• New work proposed by Torsten Lodderstedt at most recent IIW
  • Ideas contributed to OpenID Connect working group
CBOR Web Token (CWT) – RFC 8392

• Binary equivalent of JWT
  • Uses CBOR – RFC 7049 – instead of JSON
• Secured with CBOR Object Signing and Encryption (COSE) – RFC 8152
• Can be more compact than JWTs because no base64url encoding
• Good fit for IoT applications and bandwidth-constrained channels
IndieAuth

OAuth for the Open Web

Aaron Parecki
aaronpk.com
W3C Social Web Working Group

• Chartered to create open APIs for social networking, to enable social communication on the web
• Active from July 2014 to February 2018
• Identity and authentication was out of scope for REC-track documents

https://www.w3.org/wiki/Socialwg
W3C Social Web Working Group

W3C Recommendations Published:
• Webmention
• Linked Data Notifications
• Micropub
• Activity Streams
• WebSub
• ActivityPub

W3C Notes Published:
• Social Web Protocols
• JF2
• Post Type Discovery
• IndieAuth

https://www.w3.org/wiki/Socialwg#Specifications
@aaronpk My curiosity is piqued, where can I find more about Aperture?
March 12, 2018 7:13pm +00:00

Grant Richmond liked a post on aaronparecki.com
March 12, 2018 7:02pm +00:00

Jared Hanson liked twitter.com/aaronpk/status/973255081519808512 and aaronparecki.com/2018/03/12/10/homebrew-microblog
March 12, 2018 6:08pm +00:00

quill.p3k.io
Marty McGuire
marty.mcguire

Micropub for a static Neocities website
This post gives more technical detail for the recent talk that I gave at Bring-a-Hack NYC. In it, I describe a system that copies posts from Ghost Party’s Instagram automatically to the Ghost Party Website at ghostparty.today.
@aaronpk My curiosity is piqued, where can I find more about Aperture?

March 12, 2018 7:13pm +00:00

It's my super in-development IndieWeb reader, so I haven't done much in the way of docs or screenshots yet. I'm working on a summary post right now though!

Reply with Quill

---

Aaron Parecki

It's my super in-development IndieWeb reader, so I haven't done much in the way of docs or screenshots yet. I'm working on a summary post right now though!

Location: Portland, Oregon • Time: 64°F

Mon, Mar 12, 2018 12:32pm -07:00

Have you written a response to this? Let me know the URL:

Send Webmention

Posted in replies using monocle.p3k.io

---

Micro.blog

micro.blog
Follow me from Mastodon: aaronpk@aaronparecki.com
How can I comment on this [blog post, photo, issue, etc] without having an account there?
How can I sign in to an app that lets me post to my account?
Traditional OAuth

- Sign in with GitHub
- Sign in with Google
- Sign in with Twitter
- Sign in with Eventbrite
IndieAuth: Bring your own identity

Sign in with your domain

https://you.example.com

Login
URLs for Identity

• aaronparecki.com
• mastodon.social/@aaronpk
• gitlab.com/aaronpk
• twitter.com/aaronpk
IndieAuth Summary

• User IDs are URLs – bring your own identity
• Applications are identified by URLs – no pre-registration necessary
• Authorization server is discovered from the user’s URL
• User ID is returned at the end of the OAuth exchange
Sign in to Aperture

https://aaronparecki.com

Log In
This app is requesting the following scopes. You can edit the scopes that will be granted to this application.

**Publishing**
- [ ] create
  Allows the application to create posts and upload to the Media Endpoint
- [ ] update
  Allows the application to update posts
- [ ] delete
  Allows the application to delete posts
- [ ] media
  Allows the application to upload to the Media Endpoint

**Reading**
- [x] read
  Allows the application to read content from channels
- [ ] follow
  Allows the application to follow and unfollow feeds
- [ ] channels
  Allows the application to manage your channels

**Channels**

- [x] Approve
## Channels

<table>
<thead>
<tr>
<th>Channel</th>
<th>Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notifications</td>
<td>2</td>
</tr>
<tr>
<td>Twitter Mentions</td>
<td>2</td>
</tr>
<tr>
<td>IndieWeb Friends</td>
<td>28</td>
</tr>
<tr>
<td>micro.blog</td>
<td>1</td>
</tr>
<tr>
<td>micro.blog discover</td>
<td>1</td>
</tr>
<tr>
<td>Instagram</td>
<td>1</td>
</tr>
<tr>
<td>IndieWeb</td>
<td>2</td>
</tr>
<tr>
<td>IndieWebCat</td>
<td>2</td>
</tr>
</tbody>
</table>

[aperture.p3k.io]
IndieAuth Providers

- micro.blog
- WordPress Plugin
- Drupal Plugin
- withknown.com

and more!

- Selfauth – PHP
- Dobrado – PHP
- Acquiescence – Ruby
- Cellar Door – Node.js
- Microblog.pub – Python

indieweb.org/IndieAuth
IndieAuth Summary

An extension to the OAuth authorization code flow

- Prompt user for their identity (URL input, browser extension auto-fill, etc)
- Discover user’s authorization endpoint
- Send the user there to ask their permission
- On the redirect back, exchange the authorization code for an access token and the user’s canonical URL
Learn More

https://indieauth.net

https://aaronparecki.com/2018/07/07/7/oauth-for-the-open-web