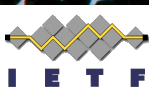


IETF 113  
March 2022  
Vienna, Austria



(somewhere on  
the other side of  
Austria) →

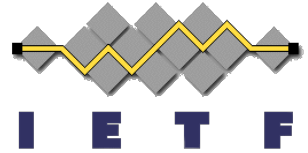


# OAuth 2.0 Demonstrating Proof-of-Possession at the Application Layer

Brian Campbell, Dr. Daniel Fett, Michael B. Jones, Torsten Lodderstedt, David Waite, John Bradley



# Expectation Setting



- 45 minutes (hopefully less)
- Jetlag...
- DPoP protocol overview
  - noting recent changes
- Wrap up

## Monday's Agenda

- **Chairs update** - Rifaat/Hannes (15 min)
- **DPoP** - Mike/Brian (45 min)
  - <https://datatracker.ietf.org/doc/draft-ietf-oauth-dpop/>
- **Redirection Attacks** - Rifaat (30 min)
  - <https://mailarchive.ietf.org/arch/msg/oauth/4-YCJzeDH4NH-ge9OF8bAbqWqIE/>
- **OAuth 2.1** - Aaron (30 min)
  - <https://datatracker.ietf.org/doc/draft-ietf-oauth-v2-1/>

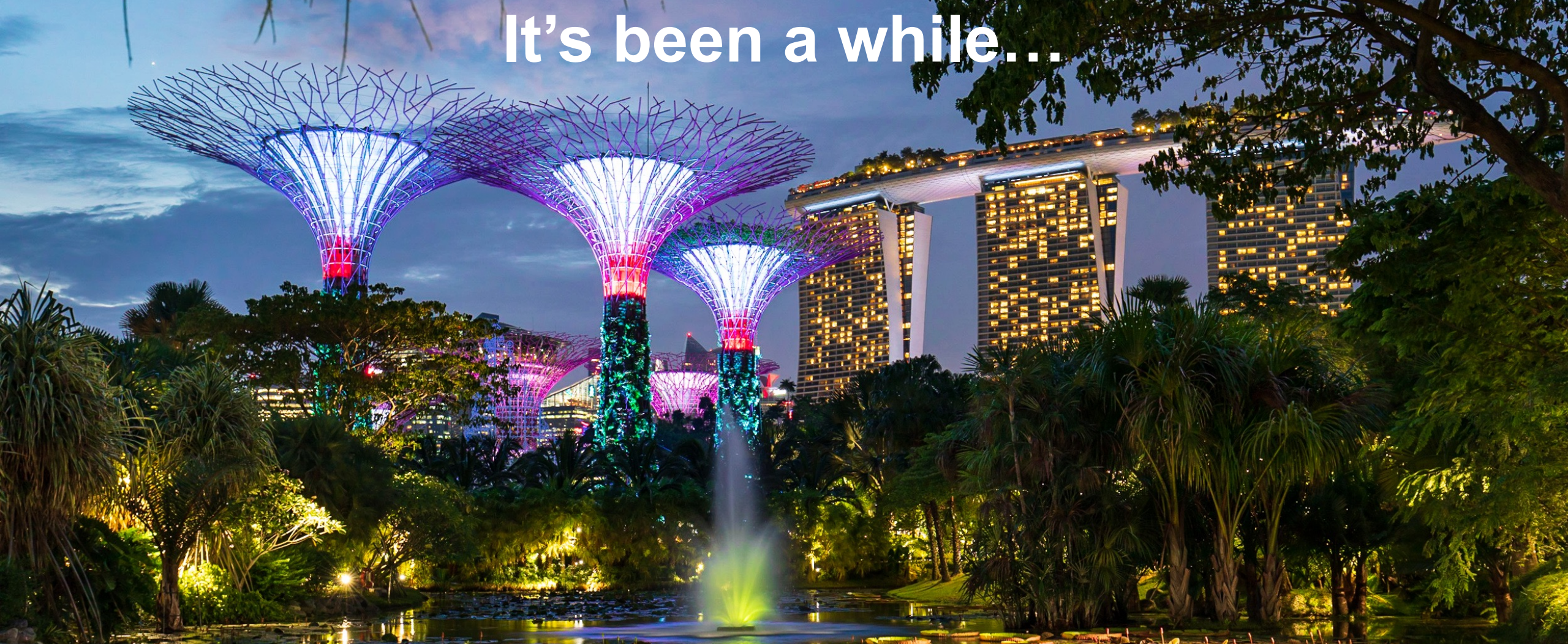
# DPoP Objectives

Pragmatic\* application-level proof-of-possession mechanism for OAuth 2.0 access tokens and refresh tokens issued to public clients

\* if not as simple as it once was or terribly efficient

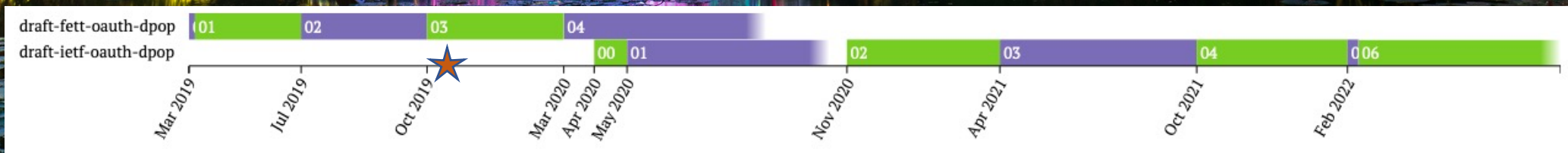


It's been a while...



## Happenings since the last in-person meeting

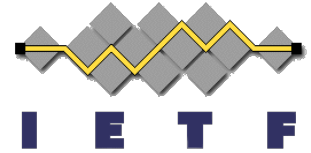
IETF 106 Singapore, Nov 2019







# Anatomy of a DPoP Proof JWT



Explicitly typed

```
{  
  "typ": "dpop+jwt",  
  "alg": "ES256",
```

Asymmetric signature algorithms only

The public key to verify the signature and proof-of-possession is being demonstrated

```
  "jwk":
```

```
{  
  "kty": "EC", "crv": "P-256"  
  "x": "18tFrhx-34tV3hRICRDY9zCkD1pBhF42UQUfWVAWBFs",  
  "y": "9VE4jf_0k_o64zbTT1cuNJajHmt6v9TDVrU0CdvGRDA"  
}
```

Only valid for a limited time window relative to creation time

```
}.
```

Unique identifier for replay checking

Minimal info about the HTTP request

-03 Hash of the access token (only for protected resource access)

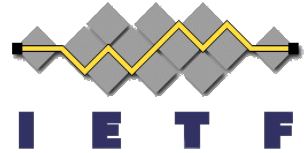
```
{  
  "jti": "-BwC3ESc6acc21Tc",  
  "htm": "POST",  
  "htu": "https://rs.example.com/important/stuff",  
  "iat": 1637259115
```

-04 server-provided nonce value (only when previously provided by the server)

```
  "ath": "fUHy02r2Z3DZ53EsNrWBb0xWXoaNy59IiKCAqksmQEO"  
  "nonce": "aiJ6sfh_zG.e.yffs0Z-Hv4w-7v"  
}
```

“... MAY contain other headers or claims as defined by <sup>6</sup> extension, profile, or deployment specific requirements”

# (code) Access Token Request



POST /token HTTP/1.1

Host: server.example.com

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

DPoP: eyJ0eXAiOiJkcG9wK2p3dCI6ImFsZyI6IktVMjU2IiwiaWdrIjp7Imt0eSI6IkV  
VDIiwieCI6Imw4dEZyaHgtMzR0VjNoUk1DUkRZOXpDa0RscEJoRjQyVVFVZl  
dWQVdCRnMiLCJ5IjoioVZFNzGpmX09rX282NHpiVFRsY3V0SmFqSG10NnY5  
VERWclUwQ2R2R1JEQSIsImNydiI6IlAtMjU2In19.eyJqdGkiOiItQndDM0V  
TYzZHY2MybFRjIiwiaHRtIjoioiUE9TVCI6Imh0dSI6Imh0dHBz0i8vc2V  
ydmVyLmV4YW1wbGUuY29tL3Rva2VuIiwiaWF0IjoxNTYyMjYyNjE2fQ.2-  
GxA6T8lP4vfrg8v-FdWP0A0zdrj8igiMLvqRMUvwnQg4PtFLbdLXiOSsX0x7  
NVY-FNyJK70nfbV37xRZT3Lg

grant\_type=authorization\_code

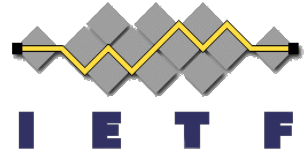
&code=Sp1xl0BeZQQYbYS6WxSbIA

&redirect\_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb

&code\_verifier=bEaL42izcC-o-xBk0K2vuJ6U-y1p9r\_wW2dFWIWgjz-

DPoP proof JWT  
in HTTP header

# Access Token Response



```
HTTP/1.1 200 OK
Content-Type: application/json
Cache-Control: no-cache, no-store
```

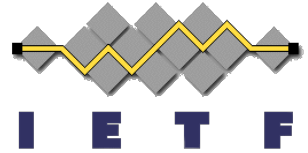
```
{
  "access_token": "Kz~8mXK1Ea1YznwH-LC-1fBAo.4Ljp~zsPE_Ne0.gxU",
  "token_type": "DPoP",
  "expires_in": 3626,
  "refresh_token": "Q..Zkm29lexi8VnWg2zPW1x-tgGad0Ibc3s3EwM_Ni4-g"
}
```

Token type indicates that the **access token** is bound to the DPoP public key

**refresh token** is bound to the DPoP public key for a public client



# (refresh) Access Token Request



POST /token HTTP/1.1

Host: server.example.com

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

DPoP: eyJ0eXAiOiJkcG9wK2p3dCI6ImFsZyI6IktVMTU2IiwiaWandrIjp7Imt0eSI6IkVDIiwieCI6Imw4dEZyaHgtMzR0VjNoUk1DUkRZOXpDa0RscEJoRjQyVVFVZlIdWQVdCRnMiLCJ5IjoioVZFNGpmX09rX282NHpiVFRsY3V0SmFqSG10NnY5VERWclUwQ2R2R1JEQSIsImNydiI6IlAtMjU2In19.eyJqdGkiOiItQndDM0VTYzZHY2MybFRjIiwiaHRtIjoiUE9TVCI6Imh0dSI6Imh0dHBz0i8vc2VydmVyLmV4YW1wbGUuY29tL3Rva2VuIiwiaWF0IjoxNTYyMjY1Mjk2fQ.pAqut2IRDm\_De6PR93SYmGBPxpwrAk90e8cP2hjiaG5QsGSuKDYW7\_X620BxqhvYC8ynrrvZLTk41mSRroapUA

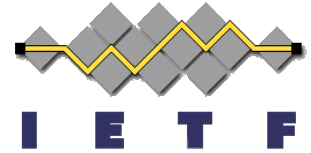
grant\_type=refresh\_token

&refresh\_token=Q..Zkm29lexi8VnWg2zPW1x-tgGad0Ibc3s3EwM\_Ni4-g

DPoP proof JWT  
in HTTP header

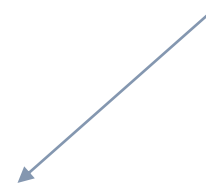
# DPoP Bound Access Token

## JWT & Introspection Response



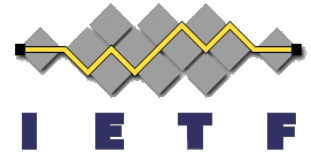
```
{
  ... other claims / members ...
  "cnf":
  {
    "jkt": "0ZcOCORZNYy-DWpqq30jZyJGHTN0d2Hg1BV3uiguA4I"
  }
}
```

Confirmation claim carries  
the SHA-256 JWK  
Thumbprint of the DPoP  
public key to which the  
access token is bound





# Protected Resource Request



GET /protectedresource HTTP/1.1

Host: resource.example.org

Authorization: DPoP Kz~8mXK1Ea1YznwH-LC-1fBAo.4Ljp~zsPE\_Ne0.gxU

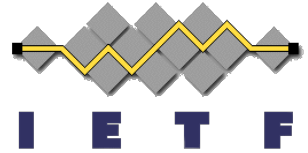
DPoP: eyJ0eXAiOiJkcG9wK2p3dCI6ImFsZyI6IkVMTmJlU2IiwiaWdrIjpw7Imt0eSI6IkVDIiwieCI6Imw4dEZyaHgtMzR0VjNoUk1DUkRZOXpDa0RscEJoRjQyVVFVZlVWQVdCRnMiLCJ5IjojOjVZFNGpmX09rX282NHpiVFRsY3VOSmFqSG10NnY5VERWclUwQ2R2R1JEQSIsImNydiI6IlAtMjU2In19.eyJqdGkiOiJlMwozVl9iS2ljOC1MQUVCIiwiaHRtIjoiaHR0cHM6Ly9yZXNvdXJjZS5leGFtcGxlLm9yZy9wcm90ZWN0ZWRyZXNvdXJjZSI6Im1hdCI6MTU2MjU2MjYxOH0.InhmpAX1WwmpBvwhok4E74kWCiGBNdavjLAeevGy32H3dbF0Jbri69Nm2ukkwb-uyUI4AUg1JSskfWIyo4UCbQ

DPoP-bound  
(reference style)  
access token

Token is  
bound to  
the key in  
the proof

DPoP  
proof

# 401 with WWW-Authenticate Challenge



## Example Response to a Protected Resource Request Without a Token

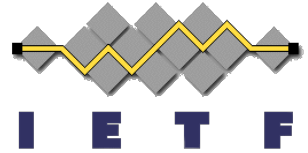
```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: DPoP algs="ES256 PS256"
```

## Example Response to a Protected Resource Request With an Invalid Token

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: error="invalid_token",
    error_description="Invalid DPoP key binding", algs="ES256"
```



# So you think you can Nonce?



- **-04** added the option for a server-provided nonce in the DPoP proof while **-05/-06** refined and clarified
  - "... DPoP-Nonce HTTP header in the response supplying a nonce value to be used when sending the subsequent request."

## Protected Resource nonce challenge

### Authorization Server nonce challenge

```
HTTP/1.1 400 Bad Request
DPoP-Nonce: 7c1P2Sczb-3yBH0-Z.Gv4w4xY
```

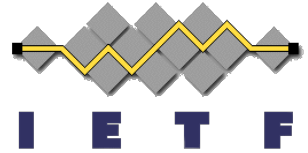
```
{
  "error": "use_dpop_nonce"
  "error_description": "AS needs nonce in DPoP proof"
}
```

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: DPoP error="use_dpop_nonce",
  error_description="plz use nonce in next DPoP proof"
DPoP-Nonce: ab3d6eqG96ye0i0999Z_HX42t77x
```

## Next nonce provided with successful response (no challenge)

```
HTTP/1.1 200 OK
DPoP-Nonce: l1m3n0peJ7S45zGPeYJbYuciQmJxj26
```

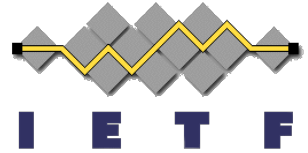
# Metadata



- Authorization Server Metadata
  - `dpop_signing_alg_values_supported`: signal support for DPoP with the JWS alg values the AS supports for DPoP proofs
- -05/-06 added Client Registration Metadata
  - `dpop_bound_access_tokens`: indicate that the client always uses DPoP when requesting tokens from the authorization server



# Binding an Authorization Code to a DPoP Key



- -05 added optional `dpop_jkt` authorization request parameter
  - SHA-256 JWK Thumbprint of the proof-of-possession public key
  - AS binds the issued authorization code to the thumbprint
  - AS checks the binding against the DPoP proof on code redemption
  - Enables end-to-end binding of the whole authorization flow
  - Can be used in conjunction with PKCE as-is
  - PAR can bind from the DPoP proof or `dpop_jkt` (need to be the same)

```
GET /authorize?response_type=code&client_id=s6BhdRkqt3&state=xyz
&redirect_uri=https%3A%2F%2Fclient%2Eexample%2Ecom%2Fcb
&code_challenge=E9MeIhoa20wvFrEMTJguCHaoeK1t8URWbuGJSstw-cM
&code_challenge_method=S256
&dpop_jkt=NzbLsXh8uDCcd-6MNwXF4W_7nowXFZAfHkxZsRGC9Xs HTTP/1.1
Host: server.example.com
```





# Next Steps (ahead of) IETF 114 Philadelphia



- Is “application/dpop+jwt” media type registration really necessary?
- Six authors is considered excessive per the RFC Style Guide
- Closing in on WGLC...?

## Internet-Draft submission

[Upload](#) [Status](#) [Instructions](#) [Approvals](#) [Manual Post Requests](#)

### Submission checks

Your draft has been verified to pass the submission checks.

**This document has more than five authors listed, which is considered excessive under normal circumstances. If you plan to request publication as an RFC, this will require additional consideration by the stream manager (for example, the IESG), and publication may be declined unless sufficient justification is provided. See [RFC 7322, section 4.1.1](#) for details.**

