draft-lodderstedt-oauth-jwt-introspection-response-00

Vladimir Dzhuvinov, Torsten Lodderstedt
Travis Spencer, Mark Dobrinic

IETF-101
March 21 2018, London

What is it?

- Proposes an additional JWT based response type for Token Introspection (RFC 7662)

```
HTTP/1.1 200 OK
Content-Type: application/json

 {
   "sub": "Z5O3upPC88QrAjx00dis",
   "aud": "https://protected.example.net/resource",
   "extension_field": "twenty-seven",
   "scope": "read write dolphin",
   "iss": "https://server.example.com/",
   "active": true,
   "exp": 1419356238,
   "iat": 1419350238,
   "client_id": "l238j323ds-23ij4",
   "username": "jdoe"
 }
```

```
HTTP/1.1 200 OK
Content-Type: application/jwt

eyJraWQiOiIxIiwiYWxnIjoiUlMyNTYifQ.eyJzdWIiOiJaNU8zdXBQQzg4UXJBa
ngwMGRpcyIsImF1ZCI6Imh0dHBzOlwvXC9wcm90ZWN0ZWQuZXhhbXBsZS5u
ZXRcL3Jlc291cmNlIiwiZXh0ZW5zaW9uX2ZpZWxkIjoidHdlbnR5LXNldmVuIiwic2
NvcGUiOiJyZWFkIHdyaXRlIGRvbHBoaW4iLCJpc3MiOiJodHRwczpcL1wvc2Vyd
mVyLmV4YW1wbGUuY29tXC8iLCJhY3RpdmUiOnRydWUsImV4cCI6MTQxOTM
1NjIzOCwiaWF0IjoxNDE5MzUwMjM4LCJjbGllbnRfaWQiOiJsMjM4ajMyM2RzLTI
zaWo0IiwidXNlcm5hbWUiOiJqZG9lIn0.HEQHf05vqVvWVnWuEjbzUnPz6JDQVR
69QkxgzBNq5kk-sK54ieg1STazXGsdFAT8nUhiiV1f_Z4HOKNnBs8TLKaFXokhA
0MqNBOYI--2unVHDqI_RPmC3p0NmP02Xmv4hzxFmTmpgjSy3vpKQDihOjhwN
Bh7G81JNaJqjJQTRv_1dHUPJotQjMK3k8_5FyiO2p64Y2VyxyQn1VWVlgOHlJw
hj6BaGHk4Qf5F8DHQZ1WCPg2p_-hwflNfXh1_buSjxyDRF4oe9pKy6ZB3ejh9qI
Mm-WrwltuU1uWMXxN6eS6tUtpKo8UCHBwLWCHmJN7KU6ZojmaISspdS23lEL
Alyw
```

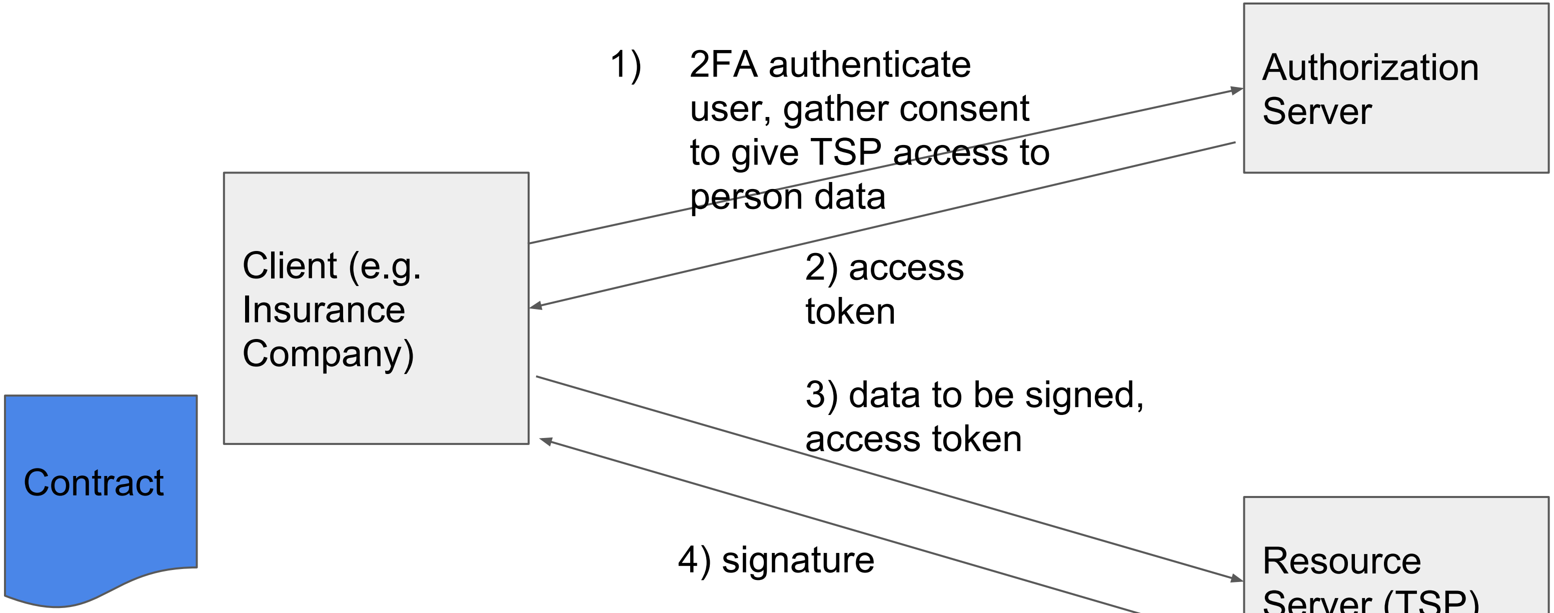
What is it good for?

- Allows the AS to sign and/or encrypt the Token Introspection response
- Signing gives RS a cryptographic proof
  - that a particular AS has issued the access token and
  - what data the AS asserted in the access token
- Encryption may allow intermediaries to fetch the access token without getting access to the access token’s payload
- Signing may be used to ensure the access token’s integrity and authenticity in such cases

Use Case 1: Qualified Electronic Signature

- RS is Trusted Service Provider according to eIDAS (EU directive on **e**lectronic **ID**entification, **A**uthentication and trust **S**ervices)
- Offers remotely activated electronic signatures
- Can be used with an access token issued by an AS complying with eIDAS level of assurance substantial (identity proofing and authentication)

Use Case 1: Qualified Electronic Signature (2)



Authorization Server

1) 2FA authenticate user, gather consent to give TSP access to person data

2) access token

Client (e.g. Insurance Company)

Contract

3) data to be signed, access token

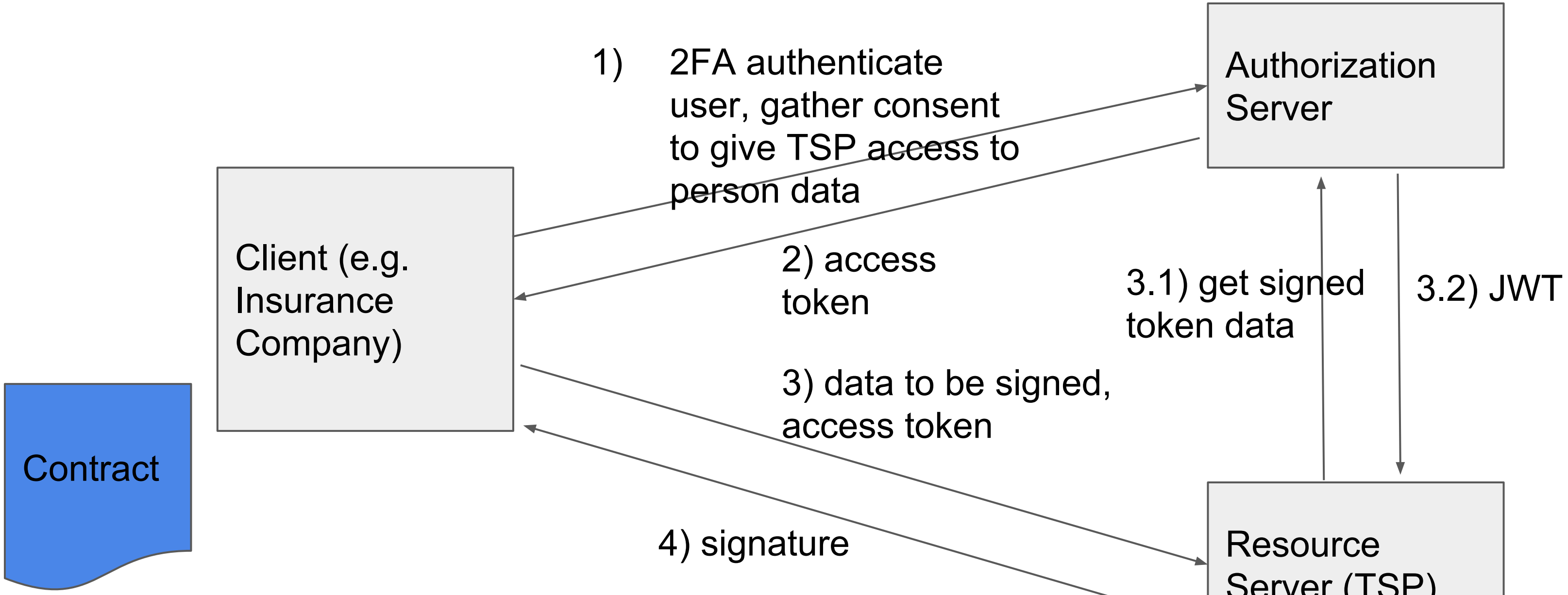
4) signature

Resource Server (TSP)

Use Case 1: Qualified Electronic Signature (3)

- The RS (TSP) is obliged to keep an audit trail of the whole process, including **what entity performed the identity verification and authentication**
- Signed Access Tokens help the RS to securely link the transaction back to the respective AS
- Structured access tokens? Not the easiest choice if the clients wants to access multiple RSs based on the same authorization grant
- **Token Introspection is easier to use from a client perspective but currently lacks digitally signed tokens.**

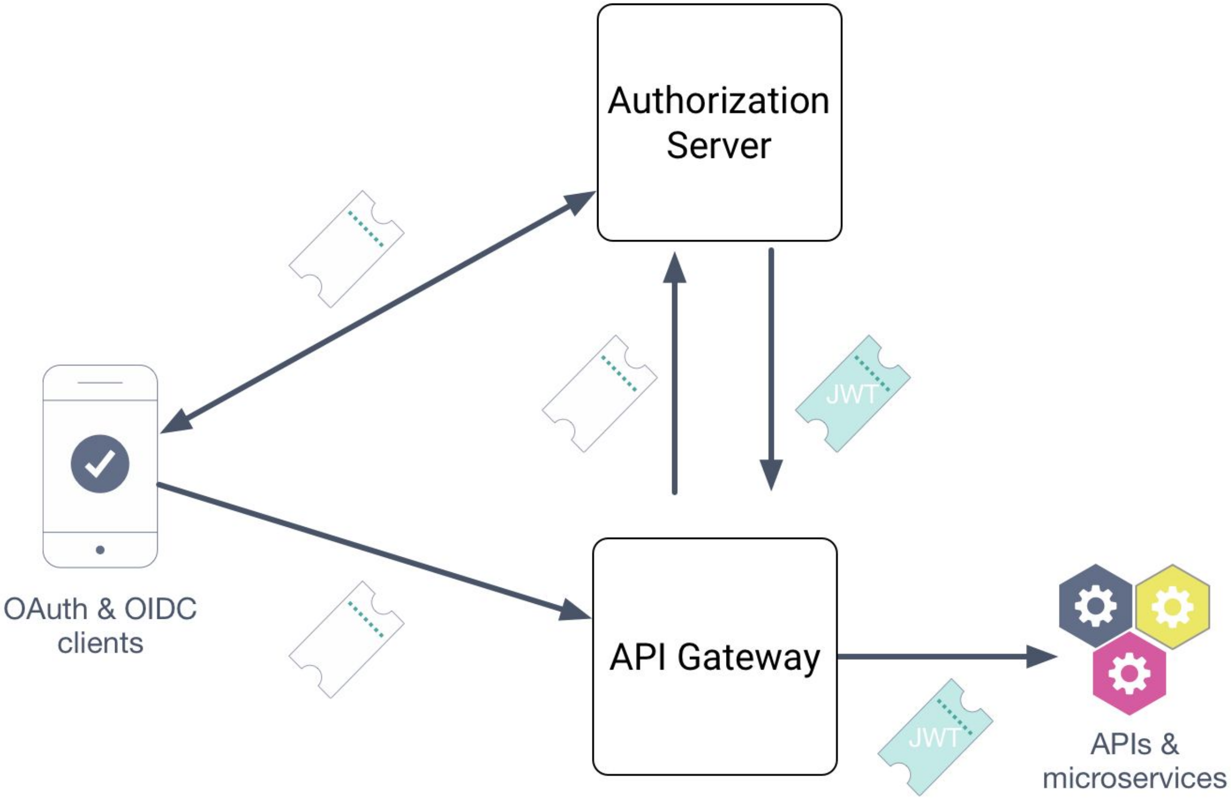
Use Case 1: Qualified Electronic Signature (4)


Authorization Server
1) 2FA authenticate user, gather consent to give TSP access to person data
Client (e.g. Insurance Company)
2) access token
3.1) get signed token data
3.2) JWT
3) data to be signed, access token
Contract
4) signature
Resource Server (TSP)

Use Case 2: Phantom Token Pattern

- Issue “handle token” or by-ref token to apps
- Send by-ref token to API which is fronted by an API gateway
- API gateway converts by-ref token to by-value JWT token using introspection
- Gateway caches using by-ref token as cache key
- Gateway forwards by-value JWT token to back-end microservices
- Back-end microservices can verify JWT off-line without communication to AS

Use Case 2: Phantom Token Pattern (2)



Status

Current

- Published revision -00
- Describes JWT response format
- Describes use of meta data to determine response type

To Be Discussed

- Add data to allow RS to determine AS
- Reusability of same JWT on multiple requests
- Other response formats

TODO

- HTTP cache-control & Expires response header with a value teamed to the expiration time of JWT
- HTTP status codes, 204 in particular for expired input tokens (active = false)
- Add prose around Accept request header
- IANA
- Security Considerations