References:
https://datatracker.ietf.org/doc/draft-ietf-oauth-rar/ - OAuth 2.0 Rich Authorization Requests
RFC 2567 - three access level: “end user” < “operator” < “administrator”
https://github.com/istopwg/ippsample/wiki/IPP-and-OAuth - wiki page

New parameter called “authorization_details”

https://datatracker.ietf.org/doc/draft-ietf-oauth-rar/ introduces a new OAuth parameter called “authorization_details”. It is used instead of the parameter “scope” (or together with it). The value of the parameter is a JSON object that contains a required field “type”. The value of “type” should be an ASCII string that uniquely identifies a schema of the JSON object and meanings of its fields. The definition of “type” should also define the way two “authorization_details” objects can be compared with each other, as well as, modifications of the content of “authorization_details” that may be introduced by Authorization Server (AS).
The parameter “authorization_details” allows for easy extension of the protocol by defining new types. We can define at the beginning a very basic type and require that all implementations must support it. An IPP client will be able to choose any other type if both printer and AS support it. In the examples below, I named the basic type as "https://pwg.org/ipp/oauth2/basic" (it does not have to be a URI).

Two examples of possible simple types of “authorization_details”:

Example 1 (access to a particular printer):

```
authorization_details = {
  "type": "https://pwg.org/ipp/oauth2/basic",
  "locations": [
          "https://my.printer.intranet/ipp/print"    // printer’s URL verified by a certificate
      ],
  “access_level”: “operator”    // one of the three access levels from RFC 2567
}
```

Example 2 (access to a group of printers):

```
authorization_details = {
  "type": "https://pwg.org/ipp/oauth2/basic",
  "groups": [“printersA”, “printersB”, “printersC”],  // asks for access to any group
  “access_level”: “operator”    // one of the three access levels from RFC 2567
}
```

Example 1 - flow

- Replace “scopes” with the “authorization_details”
- Drop Token Exchange Request (simpler implementation)
- Client always asks for a particular printer (one authorization per single printer)

1. Client sends Get-Printer-Attributes to the printer

The client queries the printer "https://my.printer.intranet/ipp/print" with Get-Printer-attributes. The printer returns the following IPP attributes:

```
oauth-authorization-server-uri: “https://auth.server.intranet”
oauth-types-supported: [ "https://pwg.org/ipp/oauth2/basic" ]
```

2. Client checks existing OAuth sessions with AS

The client chooses "https://pwg.org/ipp/oauth2/basic" as the type of authorization_details (the printer does not support anything else so there are no other choices). The client decides that it requires an “operator” access level to the printer.
First, the client checks if it already has any access token from AS that matches the following criteria:

- Issued by the AS “https://auth.server.intranet”.
- Has the location "https://my.printer.intranet/ipp/print".
- Has an access level equals “operator” or “administrator”.

If the existing OAuth session matching these criteria is found, the Client reuses the existing access token. Otherwise, the Client initiates an authorization procedure with the server to obtain a new access token.

3. Client goes through an authorization procedure and get new token

The client sends to the AS an Authorization Request with the following parameter:

```
authorization_details = {
  "type": "https://pwg.org/ipp/oauth2/basic",
  "locations": [
         "https://my.printer.intranet/ipp/print"    // printer’s URL verified by a certificate
      ],
  “access_level”: “operator”
}
```

After completing the authorization procedure the client sends Token Request and receives a response with the following parameter:

```
authorization_details = {
  "type": "https://pwg.org/ipp/oauth2/basic",
  "locations": [
         "https://my.printer.intranet/ipp/print",
         "https://my.other.printer.intranet/ipp/print",  // AS can extend the parameter
         "https://also.printer.intranet/ipp/print"
      ],
  “access_level”: “operator”
}
```

4. The printer verifies the access token

The printer sends a Token Introspection Request with the access token sent by the client. The AS's response contains the following parameter:

```
authorization_details = {
  "type": "https://pwg.org/ipp/oauth2/basic",
  "locations": [
         "https://my.printer.intranet/ipp/print",
         "https://my.other.printer.intranet/ipp/print",
         "https://also.printer.intranet/ipp/print"
      ],
  "access_level": "operator"
}
```

Example 2 - flow

- Replace "scopes" with the "authorization_details"
- Token Exchange Request is still required (more complicated implementation)
- Client always asks for access to group of printers
- Separate endpoint access token for each printer

1. Client sends Get-Printer-Attributes to the printer

The client queries the printer "https://my.printer.intranet/ipp/print" with Get-Printer-attributes. The printer returns the following IPP attributes:

```
oauth-authorization-server-uri: "https://auth.server.intranet"
oauth-types-supported: [ "https://pwg.org/ipp/oauth2/basic" ]
oauth-groups: [ "printersA", "printersB" ]   // groups of printers/users the printer belong to
```

2. Client checks existing OAuth sessions with AS

The client chooses "https://pwg.org/ipp/oauth2/basic" as the type of authorization_details (the printer does not support anything else so there are no other choices). The client decides that it requires an "operator" access level to the printer.
First, the client checks if it already has any access token from AS that matches the following criteria:

- Issued by the AS "https://auth.server.intranet".
- Has in groups any of the following: "printersA", "printersB".
- Has an access level equals "operator" or "administrator".

If the existing OAuth session matching these criteria is found, the Client reuses the existing access token to get an endpoint access token via Token Exchange Request. Otherwise, the Client initiates an authorization procedure with the server to obtain a new access token.

3. Client goes through an authorization procedure and get new token

The client sends to the AS an Authorization Request with the following parameter:

```
authorization_details = {
  "type": "https://pwg.org/ipp/oauth2/basic",
  "groups": [ "printersA", "printersB" ],
  "access_level": "operator"
}
```

After completing the authorization procedure the client sends Token Request and receives a response with the following parameter:

```
authorization_details = {
  "type": "https://pwg.org/ipp/oauth2/basic",
  "groups": [ "printersA" ],
  "access_level": "administrator"
}
```

At the end, the Client sends a Token Exchange Request with a obtained access token and a resource_id equals "https://pwg.org/ipp/oauth2/basic" (the printer's URL must be verified by its certificate). The AS returns an endpoint access token that must be used in communication with the printer.

4. The printer verifies the access token

The printer sends a Token Introspection Request with the endpoint access token sent by the client. The AS's response contains the following parameter:

```
authorization_details = {
  "type": "https://pwg.org/ipp/oauth2/basic",
  "groups": [ "printersA" ],
  "access_level": "administrator"
}
```