Microsoft Identity platform Developer training

Sander van den Hoven

sandervd@microsoft.com

Cloud Solution Architect

Irina Kostina

Irina.Kostina@Microsoft.com

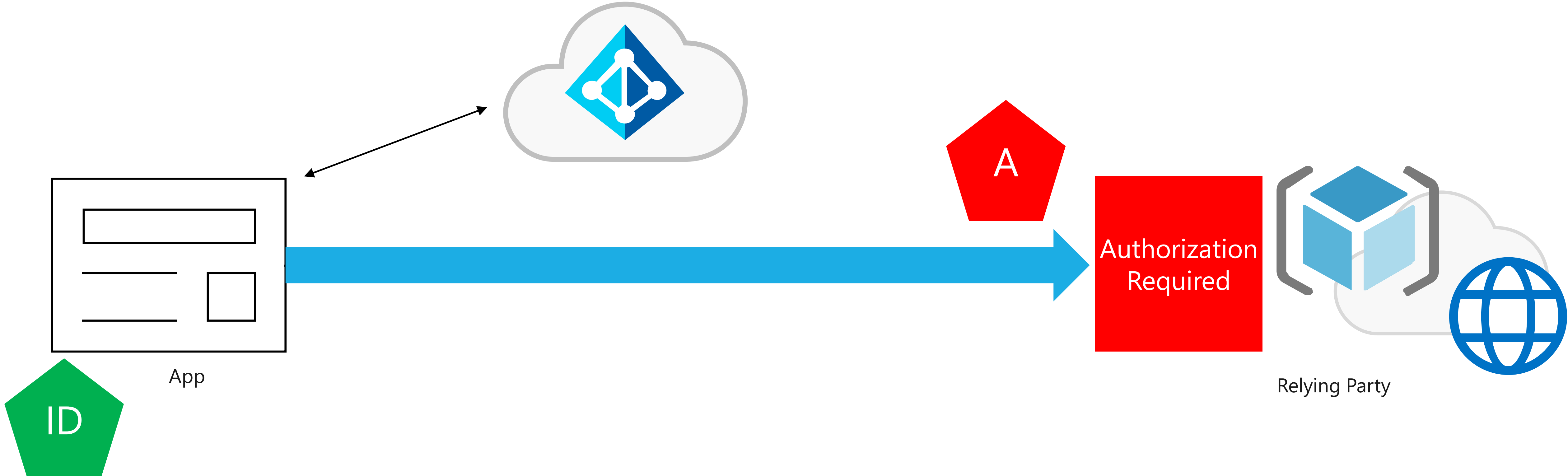
Cloud Solution Architect

Microsoft

App roles and groups

Recap

- How to add authentication – ID token vs Access Token
- Permission and consent
- Authorization and API calling another API



What are groups and roles?

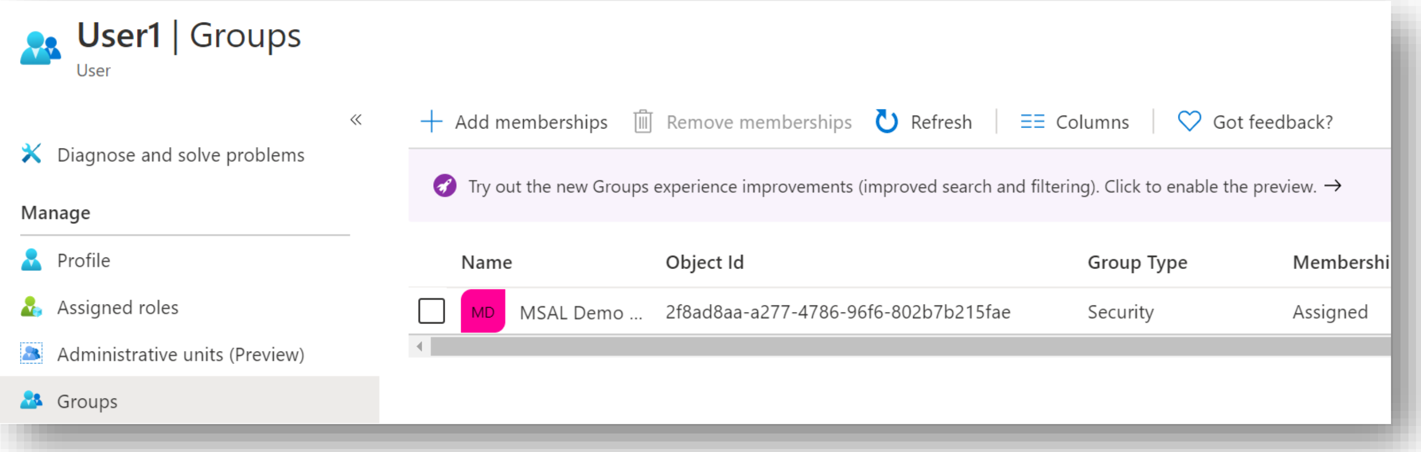
- Groups
  Groups are the Azure AD Groups where a user is member of.

- Roles
  Roles are Application specific Roles defined in the App Manifest which can be assigned to user or groups in the associated Azure AD Enterprise Application

Authorization on Groups

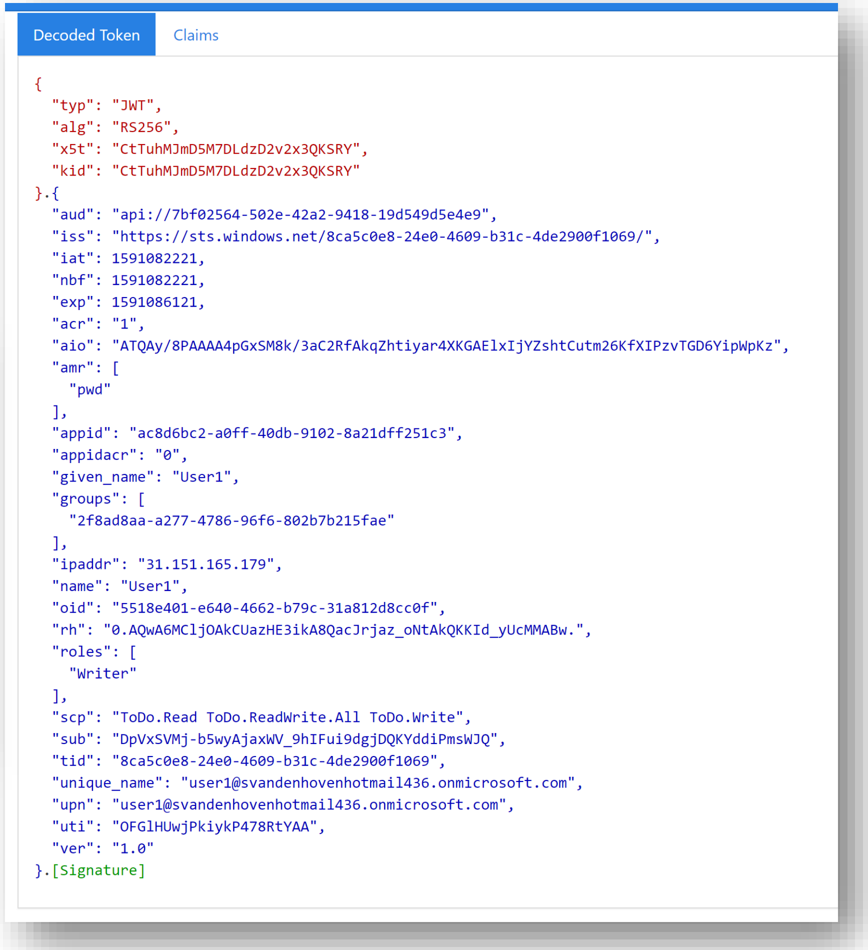
Groups are the groups a user is member of. You can define what type of groups needs to be added to the token.

Groups are tenant specific!



Try out the new Groups experience improvements (improved search and filtering). Click to enable the preview. →

	Name	Object Id	Group Type	Membershi
□	MD MSAL Demo ...	2f8ad8aa-a277-4786-96f6-802b7b215fae	Security	Assigned



```
{
  "typ": "JWT",
  "alg": "RS256",
  "x5t": "CtTuhMJmD5M7DLdzD2v2x3QKSRY",
  "kid": "CtTuhMJmD5M7DLdzD2v2x3QKSRY"
}.{
  "aud": "api://7bf02564-502e-42a2-9418-19d549d5e4e9",
  "iss": "https://sts.windows.net/8ca5c0e8-24e0-4609-b31c-4de2900f1069/",
  "iat": 1591082221,
  "nbf": 1591082221,
  "exp": 1591086121,
  "acr": "1",
  "aio": "ATQAy/8PAAAA4pGxSM8k/3aC2RfAkqZhtiyar4XKGAElxIjYZshtCutm26KfXIPzvTGD6YipWpKz",
  "amr": [
    "pwd"
  ],
  "appid": "ac8d6bc2-a0ff-40db-9102-8a21dff251c3",
  "appidacr": "0",
  "given_name": "User1",
  "groups": [
    "2f8ad8aa-a277-4786-96f6-802b7b215fae"
  ],
  "ipaddr": "31.151.165.179",
  "name": "User1",
  "oid": "5518e401-e640-4662-b79c-31a812d8cc0f",
  "rh": "0.AQwA6MCljOAkCUazHE3ikA8QacJrjaz_oNtAkQKKId_yUcMMABw.",
  "roles": [
    "Writer"
  ],
  "scp": "ToDo.Read ToDo.ReadWrite.All ToDo.Write",
  "sub": "DpVxSVMj-b5wyAjaxWV_9hIFui9dgjDQKYddiPmsWJQ",
  "tid": "8ca5c0e8-24e0-4609-b31c-4de2900f1069",
  "unique_name": "user1@svandenhovenhotmail436.onmicrosoft.com",
  "upn": "user1@svandenhovenhotmail436.onmicrosoft.com",
  "uti": "OFGlHUwjPkiykP478RtYAA",
  "ver": "1.0"
}.[Signature]
```

Using groups for authorization (Method 1)

Call Microsoft Graph to retrieve signed-in user’s group membership

```
POST https://graph.microsoft.com/v1.0/me/getMemberGroups
{ "securityEnabledOnly": true }
```

Requires *Group.Read.All* permission which requires admin consent

Request ‘groups’ claim in the token (Method 2)

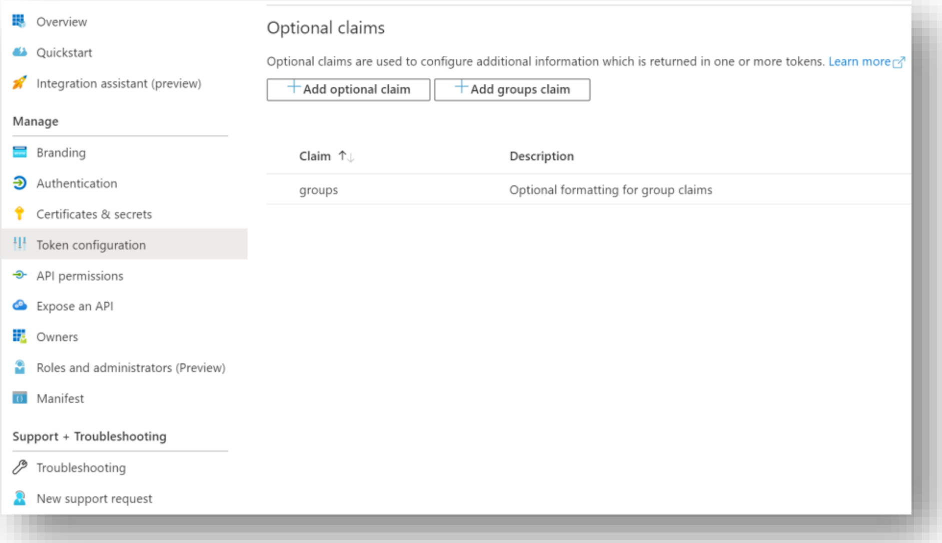
Set Application object’s "groupMembershipClaims" to “All" or “SecurityGroup” in Manifest

```
"groupMembershipClaims": "SecurityGroup",
```

Alternative it to configure this in the Azure AD Portal

The ‘groups’ claim will be an array of groups object IDs

```
"groups": [
   "dc3eb7a4-156f-4a7d-991b-72b0119aa029",
   "fb33ed9c-fb73-4d2f-8e5a-4df647ca7c5d"
 ],
```



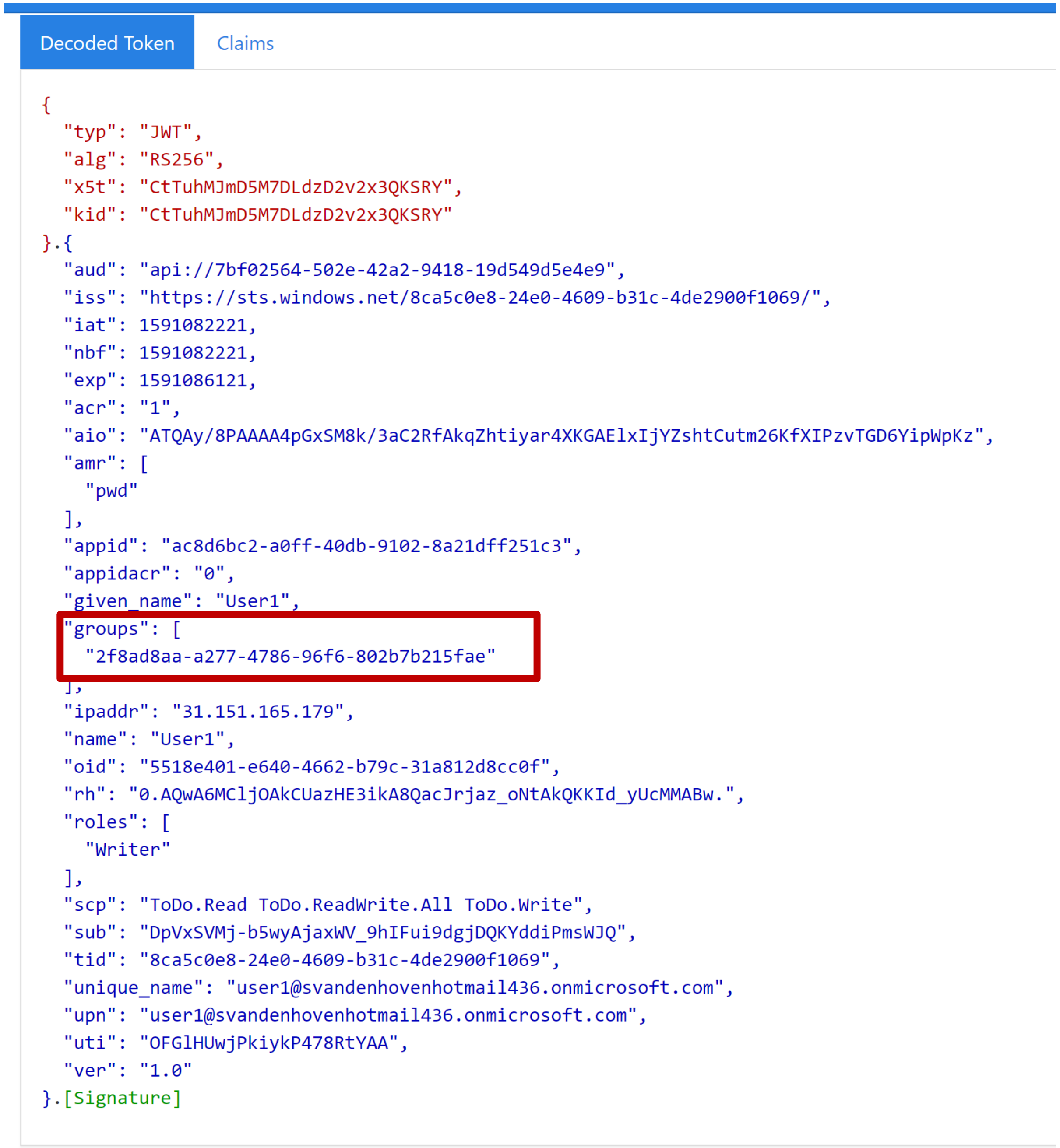
Optional claims

Optional claims are used to configure additional information which is returned in one or more tokens. Learn more

Claim ↑↓	Description
groups	Optional formatting for group claims

Nested groups are not supported by Azure AD, yet.

Demo Group Claims



```
{
  "typ": "JWT",
  "alg": "RS256",
  "x5t": "CtTuhMJmD5M7DLdzD2v2x3QKSRY",
  "kid": "CtTuhMJmD5M7DLdzD2v2x3QKSRY"
}.{
  "aud": "api://7bf02564-502e-42a2-9418-19d549d5e4e9",
  "iss": "https://sts.windows.net/8ca5c0e8-24e0-4609-b31c-4de2900f1069/",
  "iat": 1591082221,
  "nbf": 1591082221,
  "exp": 1591086121,
  "acr": "1",
  "aio": "ATQAy/8PAAAA4pGxSM8k/3aC2RfAkqZhtiyar4XKGAElxIjYZshtCutm26KfXIPzvTGD6YipWpKz",
  "amr": [
    "pwd"
  ],
  "appid": "ac8d6bc2-a0ff-40db-9102-8a21dff251c3",
  "appidacr": "0",
  "given_name": "User1",
  "groups": [
    "2f8ad8aa-a277-4786-96f6-802b7b215fae"
  ],
  "ipaddr": "31.151.165.179",
  "name": "User1",
  "oid": "5518e401-e640-4662-b79c-31a812d8cc0f",
  "rh": "0.AQwA6MCljOAkCUazHE3ikA8QacJrjaz_oNtAkQKKId_yUcMMABw.",
  "roles": [
    "Writer"
  ],
  "scp": "ToDo.Read ToDo.ReadWrite.All ToDo.Write",
  "sub": "DpVxSVMj-b5wyAjaxWV_9hIFui9dgjDQKYddiPmsWJQ",
  "tid": "8ca5c0e8-24e0-4609-b31c-4de2900f1069",
  "unique_name": "user1@svandenhovenhotmail436.onmicrosoft.com",
  "upn": "user1@svandenhovenhotmail436.onmicrosoft.com",
  "uti": "OFGlHUwjPkiykP478RtYAA",
  "ver": "1.0"
}.[Signature]
```

Groups Claim limitations

- To ensure that the token size doesn’t exceed HTTP header size limits, Azure AD limits the number of objectIds that it includes in the groups claim.
- If a user is member of more groups than the overage limit (150 for SAML tokens, 200 for JWT tokens), then Azure AD does not emit the groups claim in the token.
  - Instead, it includes an overage claim in the token that indicates to the application to query the Graph API to retrieve the user’s group membership

Consequences for groups in Multi Tenant Applications

Every Tenant has its own groups

Groups mapping becomes application responsibility

Advice if possible is not to use groups for Multi Tenant applications, but Application Roles

Application Roles

Easier to work with than groups as programming model is simpler

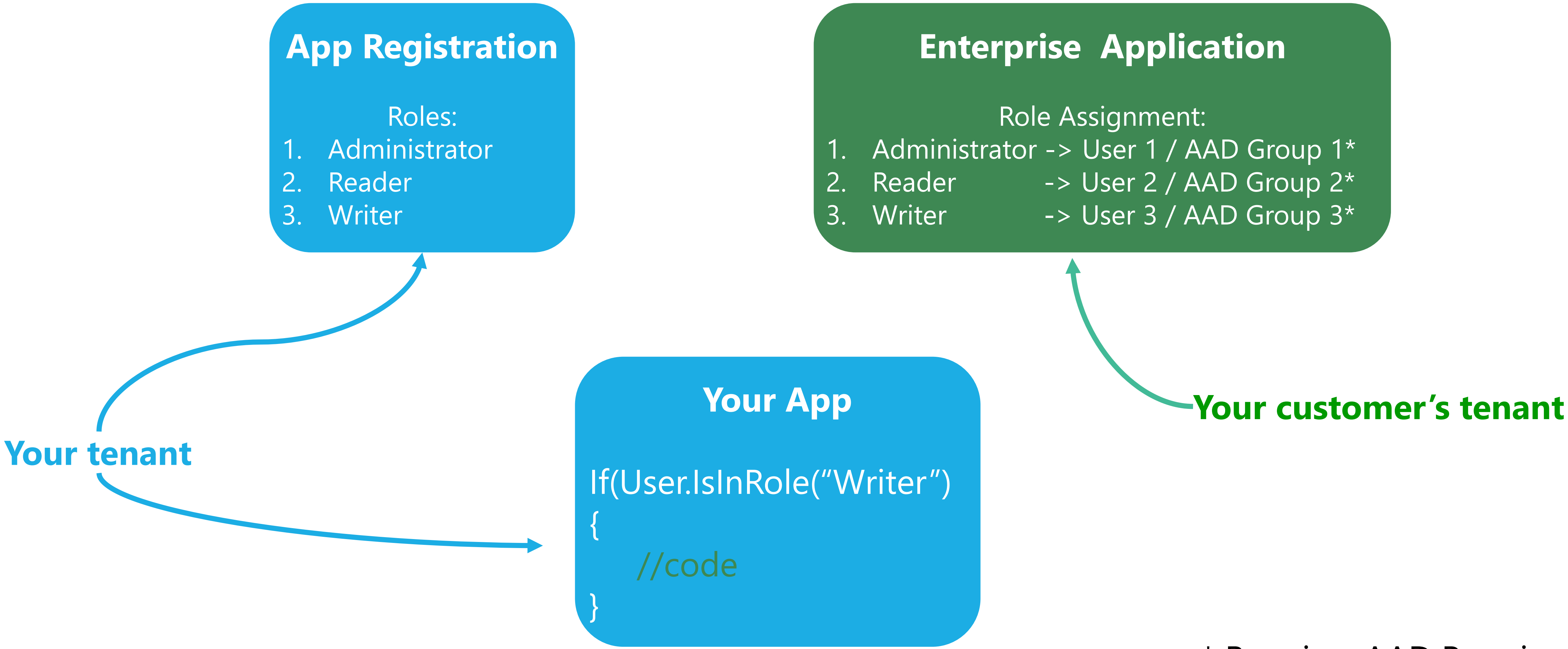
Application roles are used to assign permissions to users

Application roles are defined in an AAD app's manifest.

Thus they are specific to an application and removing an app from AAD will make these roles go away

They are provided to an app in the roles claim

Authorization on Application Roles



App Registration

Roles:

1. Administrator
2. Reader
3. Writer

Enterprise Application

Role Assignment:

1. Administrator -> User 1 / AAD Group 1*
2. Reader -> User 2 / AAD Group 2*
3. Writer -> User 3 / AAD Group 3*

Your App

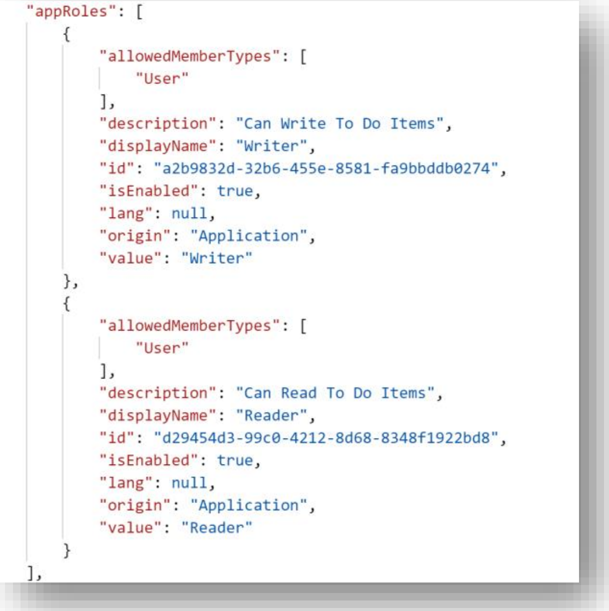
```
If(User.IsInRole("Writer")
{
    //code
}
```

Your tenant

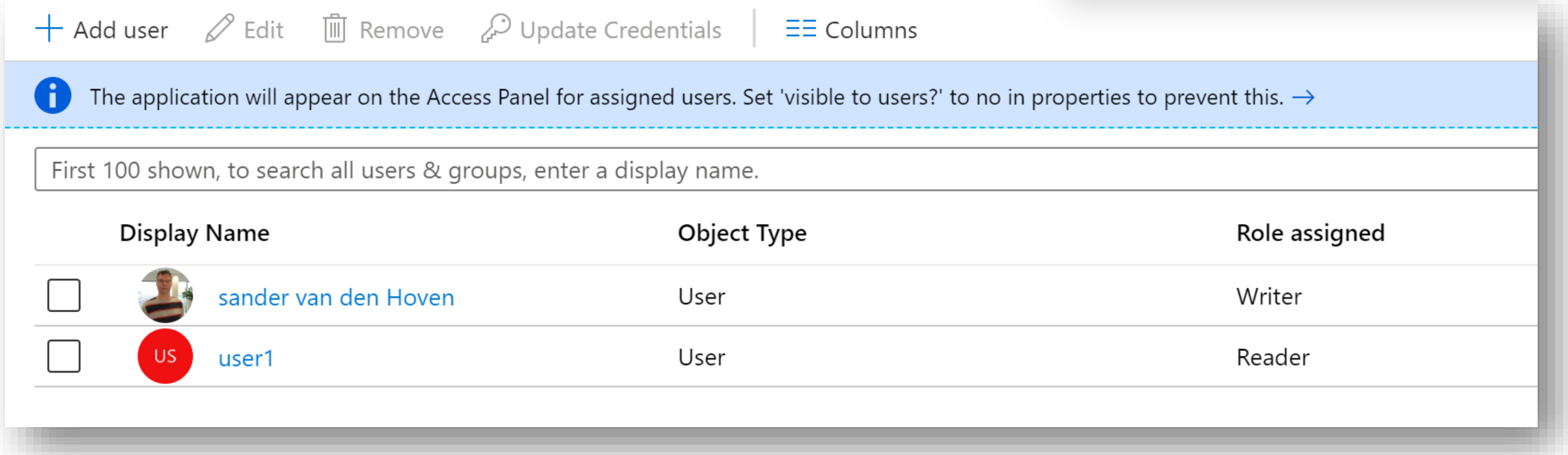
Your customer's tenant

* Requires AAD Premium

Demo
Define App Roles and Assign Users



```
"appRoles": [
    {
        "allowedMemberTypes": [
            "User"
        ],
        "description": "Can Write To Do Items",
        "displayName": "Writer",
        "id": "a2b9832d-32b6-455e-8581-fa9bbddb0274",
        "isEnabled": true,
        "lang": null,
        "origin": "Application",
        "value": "Writer"
    },
    {
        "allowedMemberTypes": [
            "User"
        ],
        "description": "Can Read To Do Items",
        "displayName": "Reader",
        "id": "d29454d3-99c0-4212-8d68-8348f1922bd8",
        "isEnabled": true,
        "lang": null,
        "origin": "Application",
        "value": "Reader"
    }
],
```



The application will appear on the Access Panel for assigned users. Set 'visible to users?' to no in properties to prevent this. →

First 100 shown, to search all users & groups, enter a display name.

	Display Name	Object Type	Role assigned
□	sander van den Hoven	User	Writer
□	user1	User	Reader

How to authorize on Application Roles

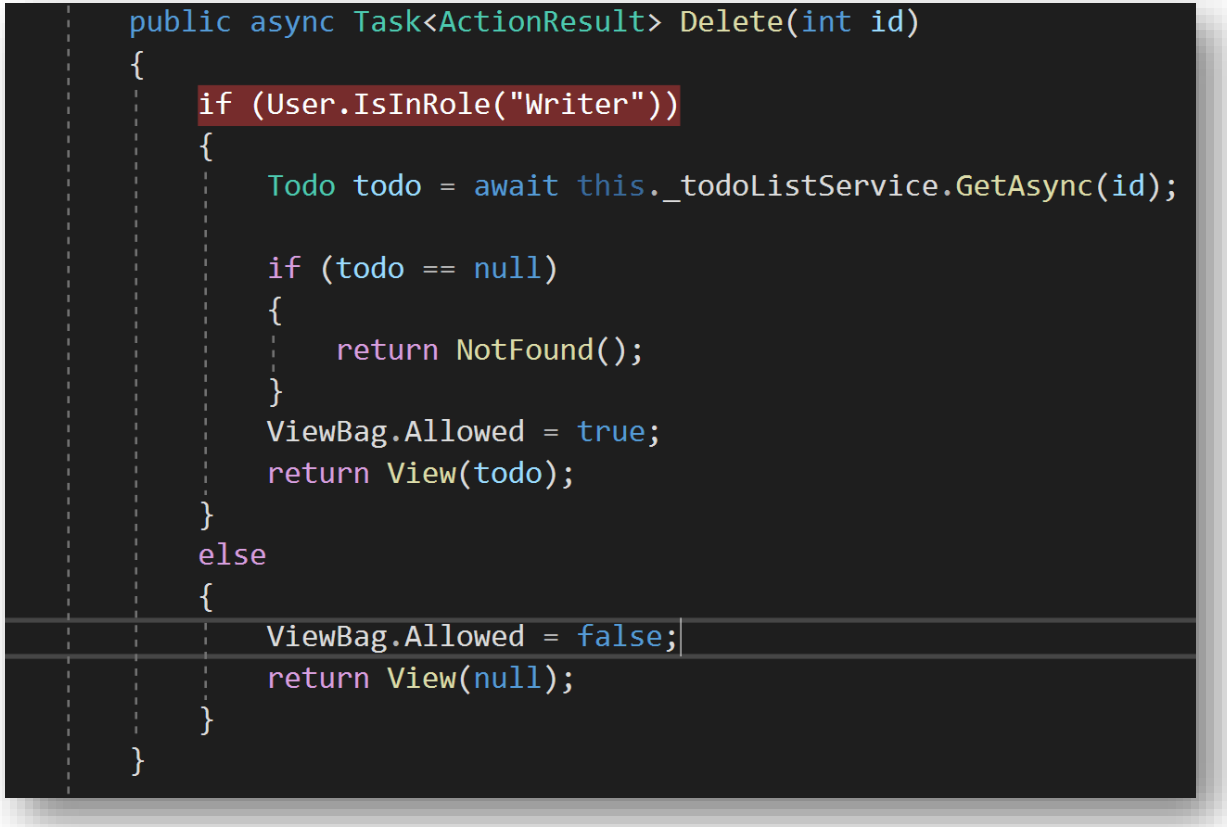
Receive assigned roles in the user’s token in roles claim

Two options:

1. Authorize in the client app (check the id_token)
2. Authorize in the API (check the access_token)

Authorization is to determine if the user has a certain value in the roles claim

Demo App Roles



```
public async Task<ActionResult> Delete(int id)
{
    if (User.IsInRole("Writer"))
    {
        Todo todo = await this._todoListService.GetAsync(id);

        if (todo == null)
        {
            return NotFound();
        }
        ViewBag.Allowed = true;
        return View(todo);
    }
    else
    {
        ViewBag.Allowed = false;
        return View(null);
    }
}
```

Homework

- Use Application roles
  https://github.com/Azure-Samples/active-directory-aspnetcore-webapp-openidconnect-v2/blob/master/5-WebApp-AuthZ/5-1-Roles

- Use groups
  https://github.com/Azure-Samples/active-directory-aspnetcore-webapp-openidconnect-v2/blob/master/5-WebApp-AuthZ/5-2-Groups

Microsoft

Next Session:

Azure B2C
WEWC593 Thursday June 18th 14:00-15:00

Sign up for the webinar at
http://aka.ms/modernIdentityForDevelopers