BC/NAV TECH DAYS 2022

mibuso.com

OAuth revealed

Arend-Jan Kauffmann
MVP | Kauffmann IT Services

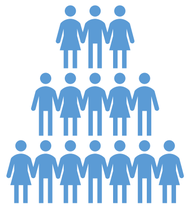
www.bctechdays.com

Introduction

Freelance Technical Consultant & Trainer

Microsoft MVP Business Applications

experience with
Dynamics NAV / Business Central
since 2002

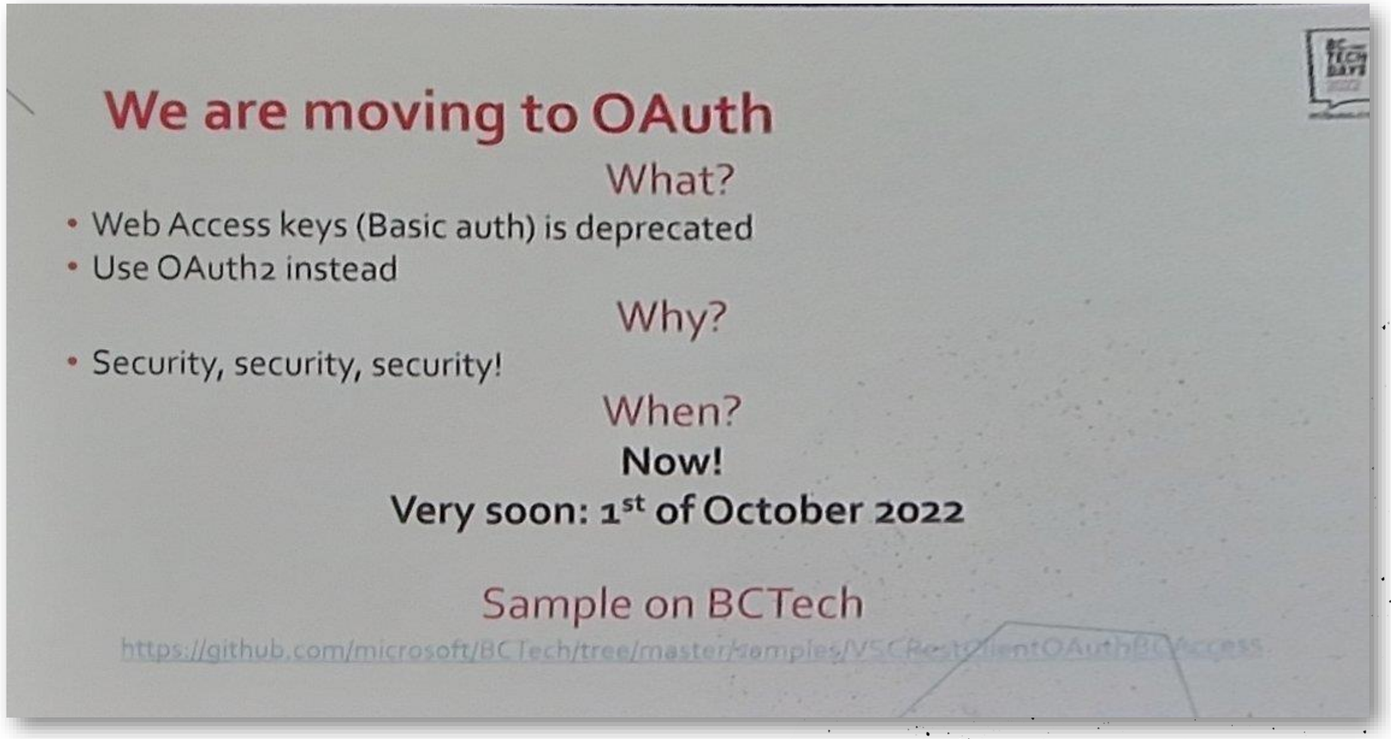


co-founder & chairman
Dutch Dynamics Community

http://kauffmann.nl

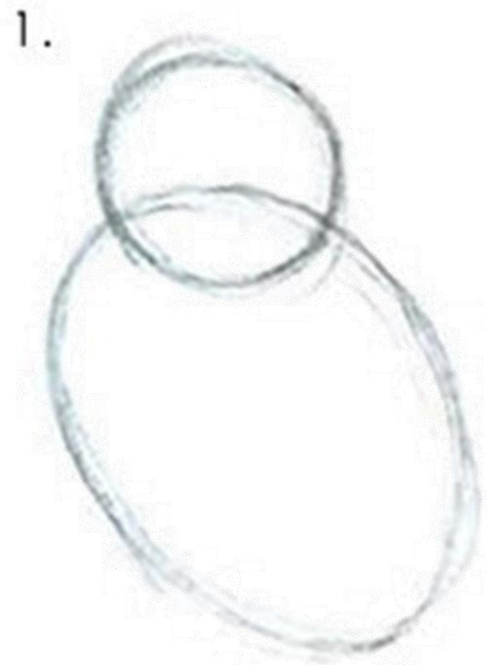
Email: aj@kauffmann.nl
Twitter: @ajkauffmann

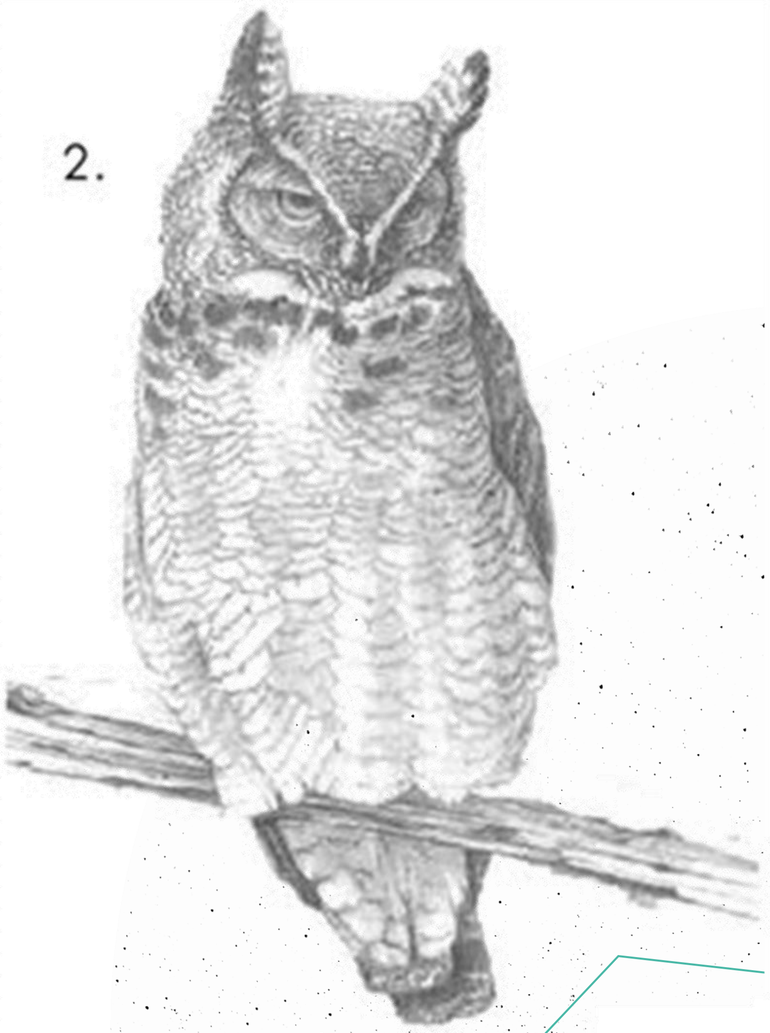
Why OAuth?




```
□□ /D□□M□.teams.microsoft.comclienttypewindows/ /G□K}h□ /E□<[□□□□A□□□□5□□□□□      □
□□ /E□□h□>.teams.microsoft.comauthtokenBearer%3DeyJ0eXAiOiJKV1QiLCJub25jZSI6IjR0d1hfcVJxRH1WNVc1bEF6SnJRUzh
```

How to draw an owl





1. Draw some circles

2. Draw the rest of the [redacted] owl

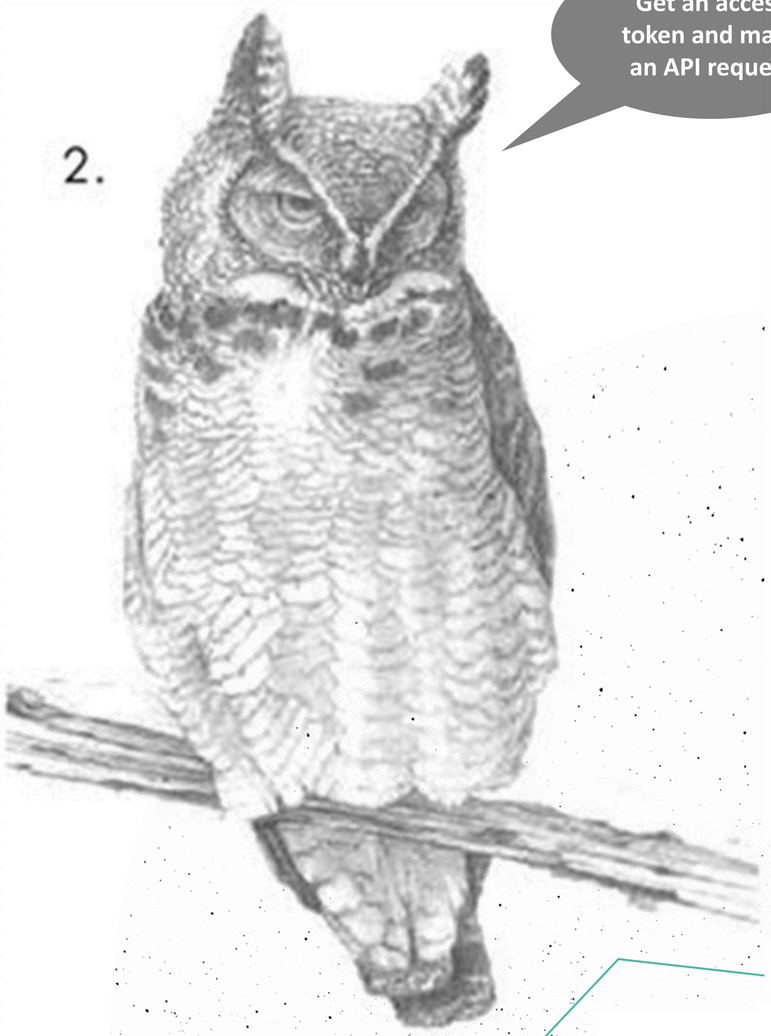
How to draw an owl

BC/NAV TECH DAYS 2022
mibuso.com

Create a secret

Create an app registration

1. Draw some circles



Get an access token and make an API request

2. Draw the rest of the [redacted] owl

How to draw an owl

App registration

Service principal

Platform

Public client

Create an app registration

1.

Redirect URI

Create a secret

Scope

Confidential client

Secret

Certificate

2.

Get an access token and make an API request

Resource

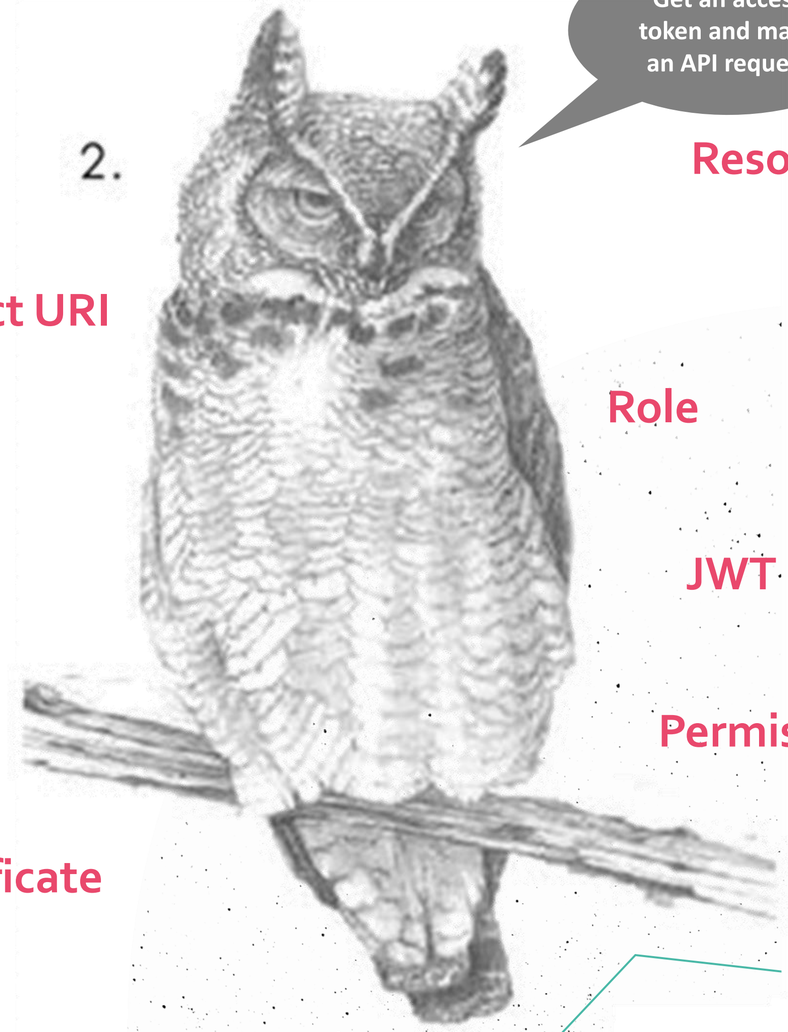
Access token

Role

JWT

Azure

Permission



1. Draw some circles

2. Draw the rest of the owl

BC/NAV TECH DAYS 2022
mibuso.com

OAuth 2.0

Industry-standard protocol for authorization

Provides specific authorization flows for

- Web applications
- Desktop applications
- Mobile phones
- Living room devices

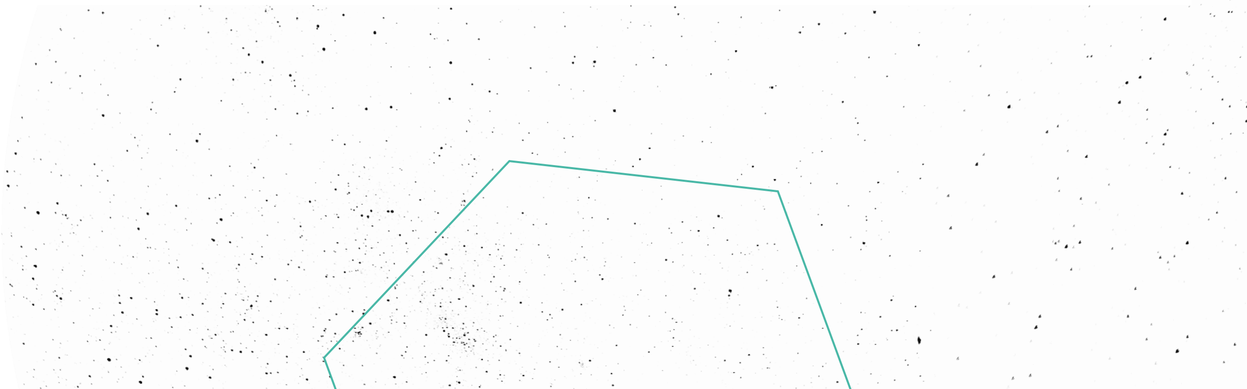
Managed by Internet Engineering Task Force (IETF)

- https://tools.ietf.org/html/rfc6749

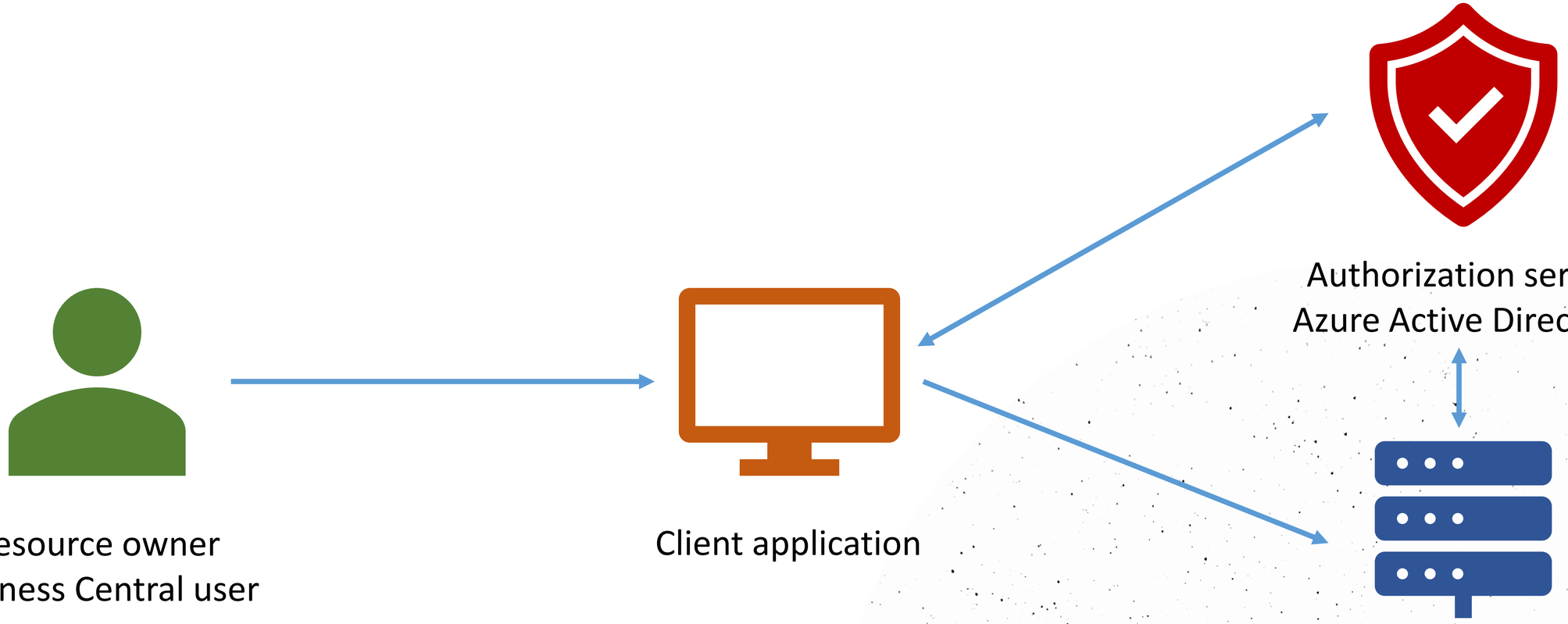
The OAuth 2.0 Authorization Framework

Abstract

The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf. This specification replaces and obsoletes the OAuth 1.0 protocol described in RFC 5849.

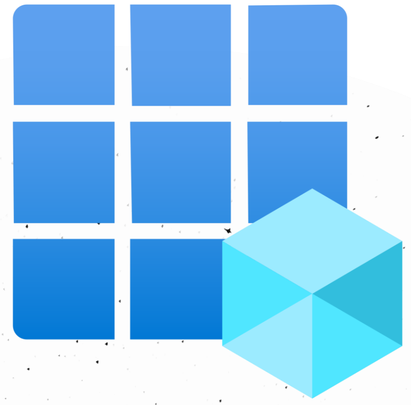


4 roles


Authorization server
Azure Active Directory
Resource owner
Business Central user
Client application
Resource server
Business Central

App Registration (Application)

- Azure Portal > Azure Active Directory > App registrations
- Definition of the application
  - Name, logo, and publisher
  - Redirect URIs
  - Secrets
  - API dependencies (required resources)
  - Published APIs / resources / scopes
  - …



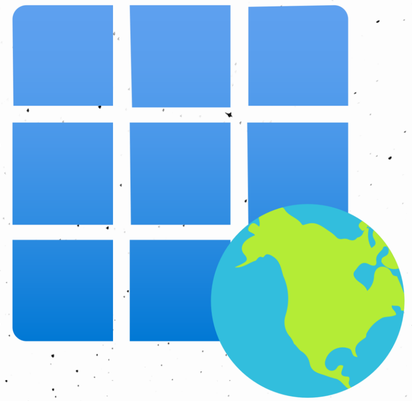
- Only exists in 1 single directory, aka its **home directory**

Enterprise Application (Service Principal)

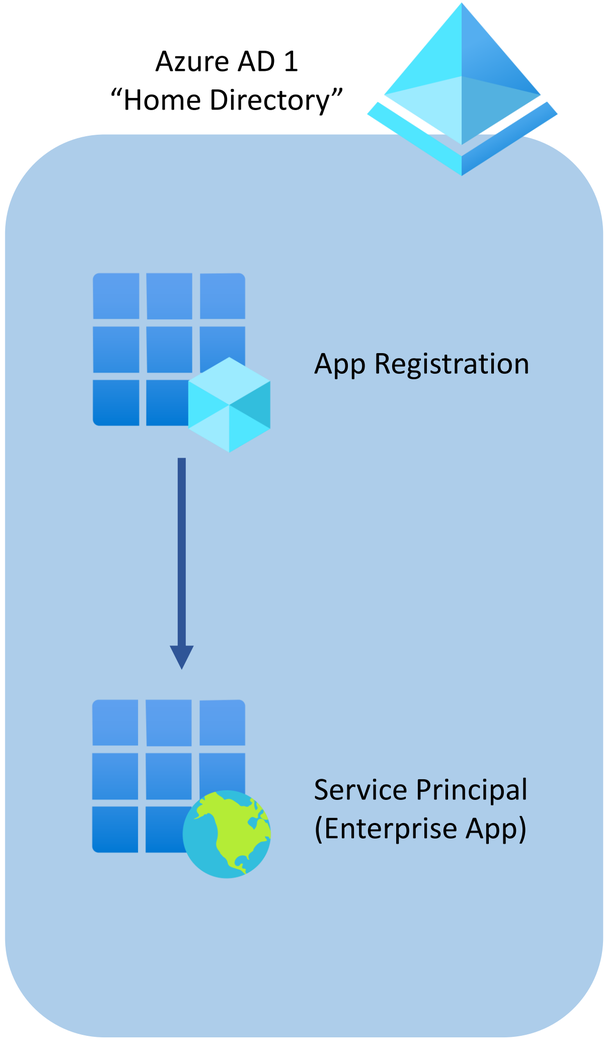
BC/NAV TECH DAYS 2022
mibuso.com

- Azure Portal > Azure Active Directory > Enterprise applications

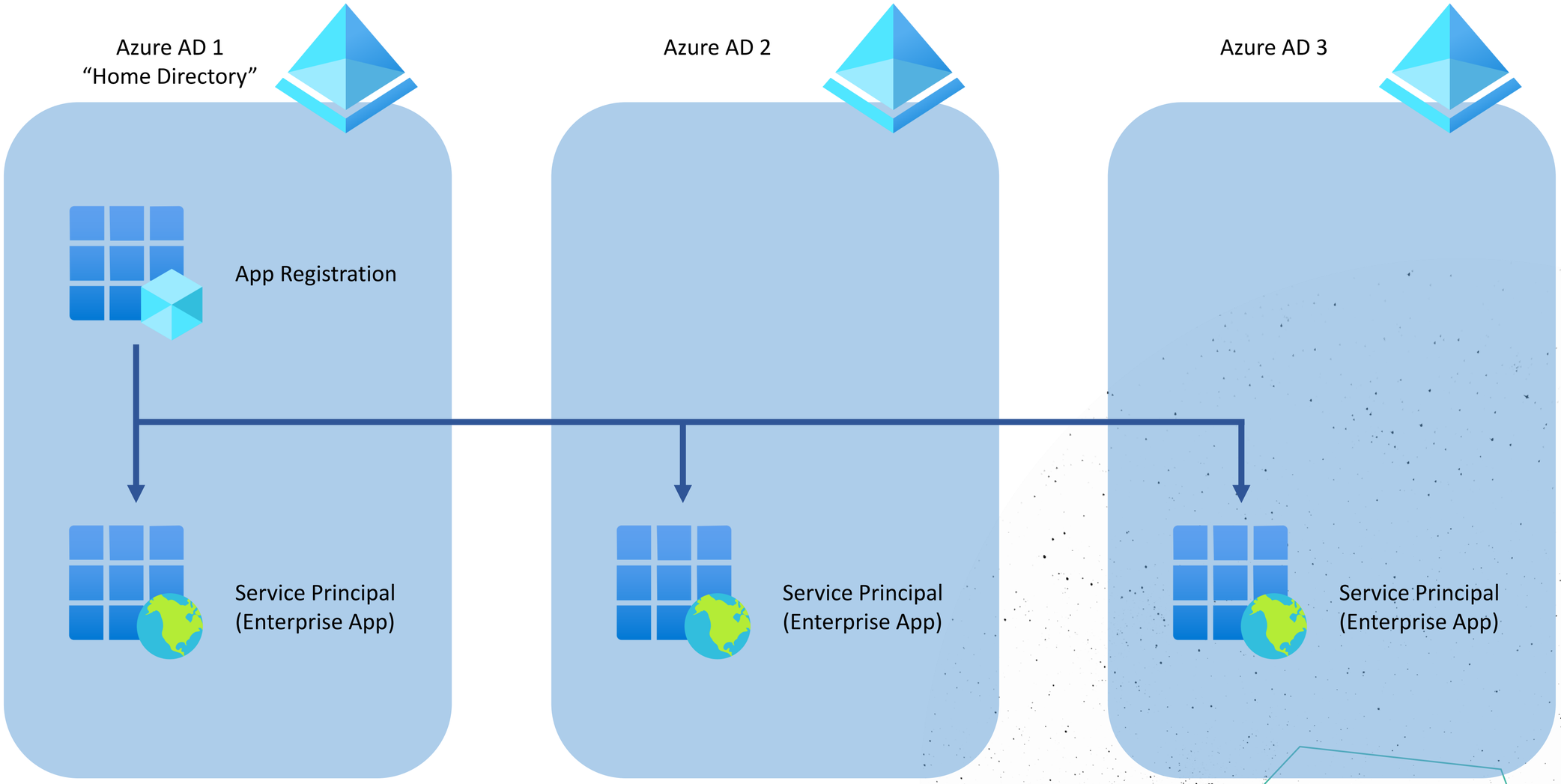
- Instance of an app registration in a directory
- Manages granted permissions
- Only permissions in the same directory



Active Directory – single tenant



Active Directory – multitenant



What is an App Registration used for?

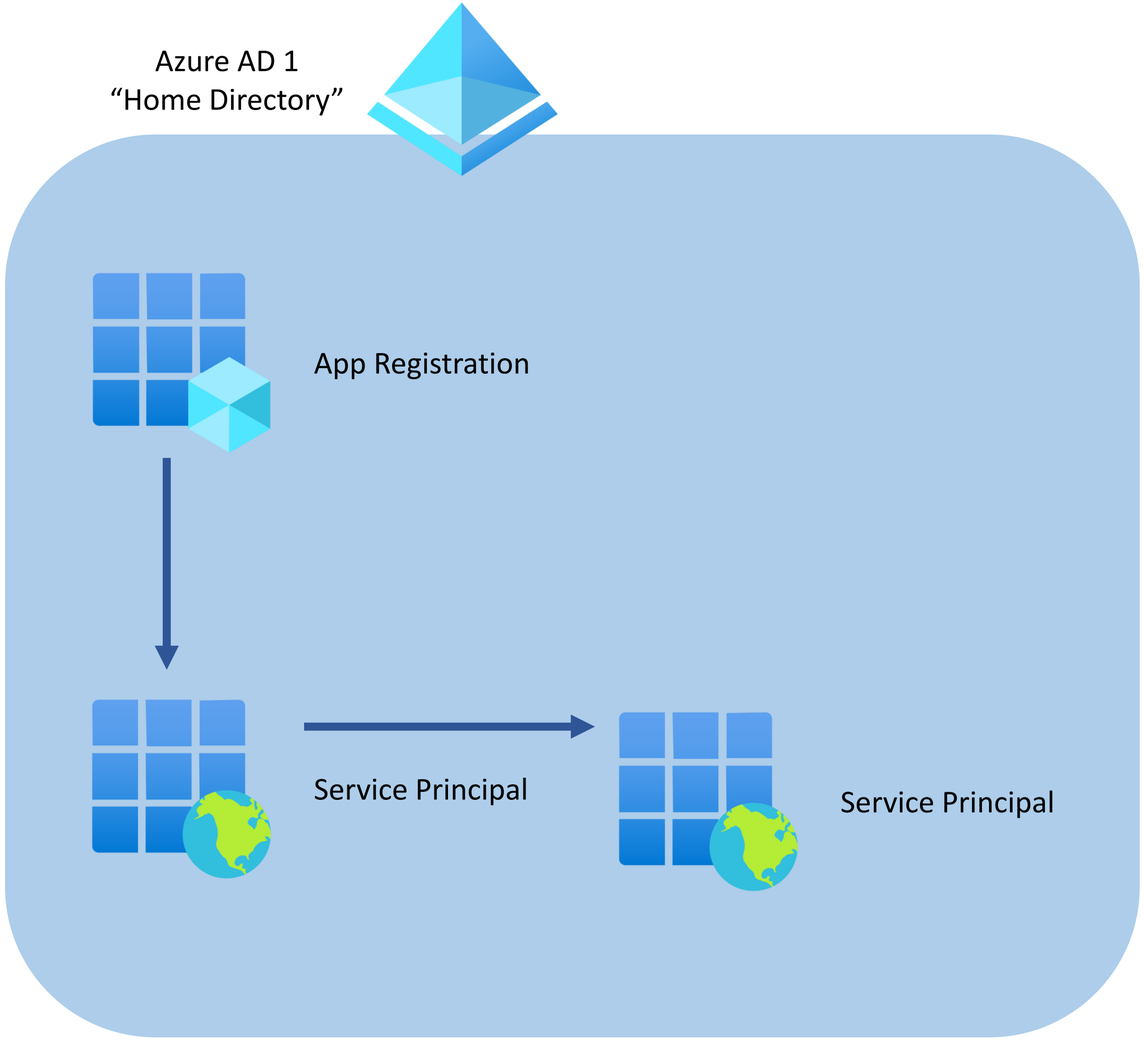
BC/NAV TECH DAYS 2022
mibuso.com

- User authentication
  - Access to account information as stored in Azure AD

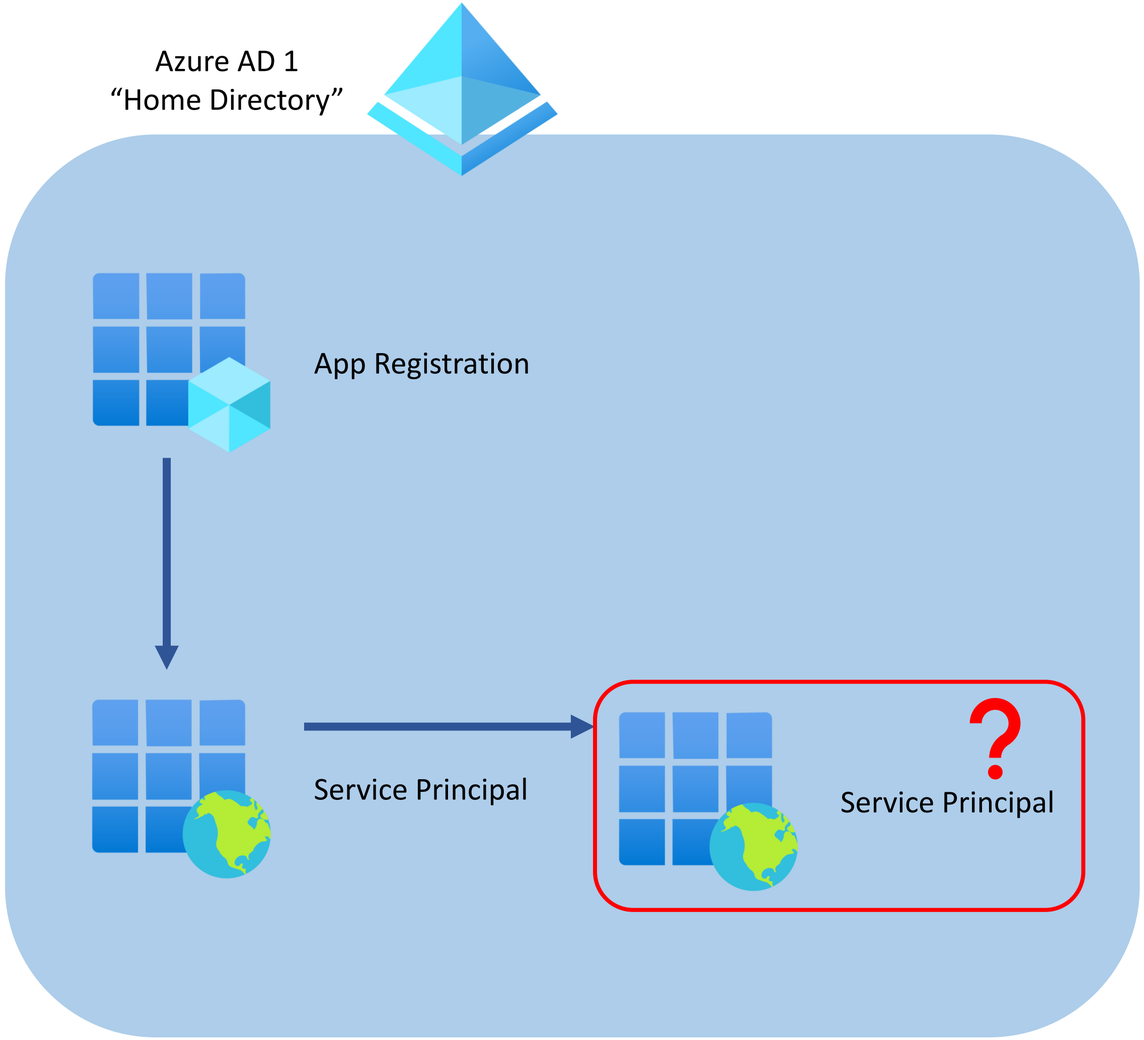
- Accessing resources
  - Data exposed by other applications
  - E.g. SharePoint, Mail, Business Central
  - Access must be granted

- Exposing resources
  - Allowing other applications to access data
  - Defined by APIs, permissions and roles

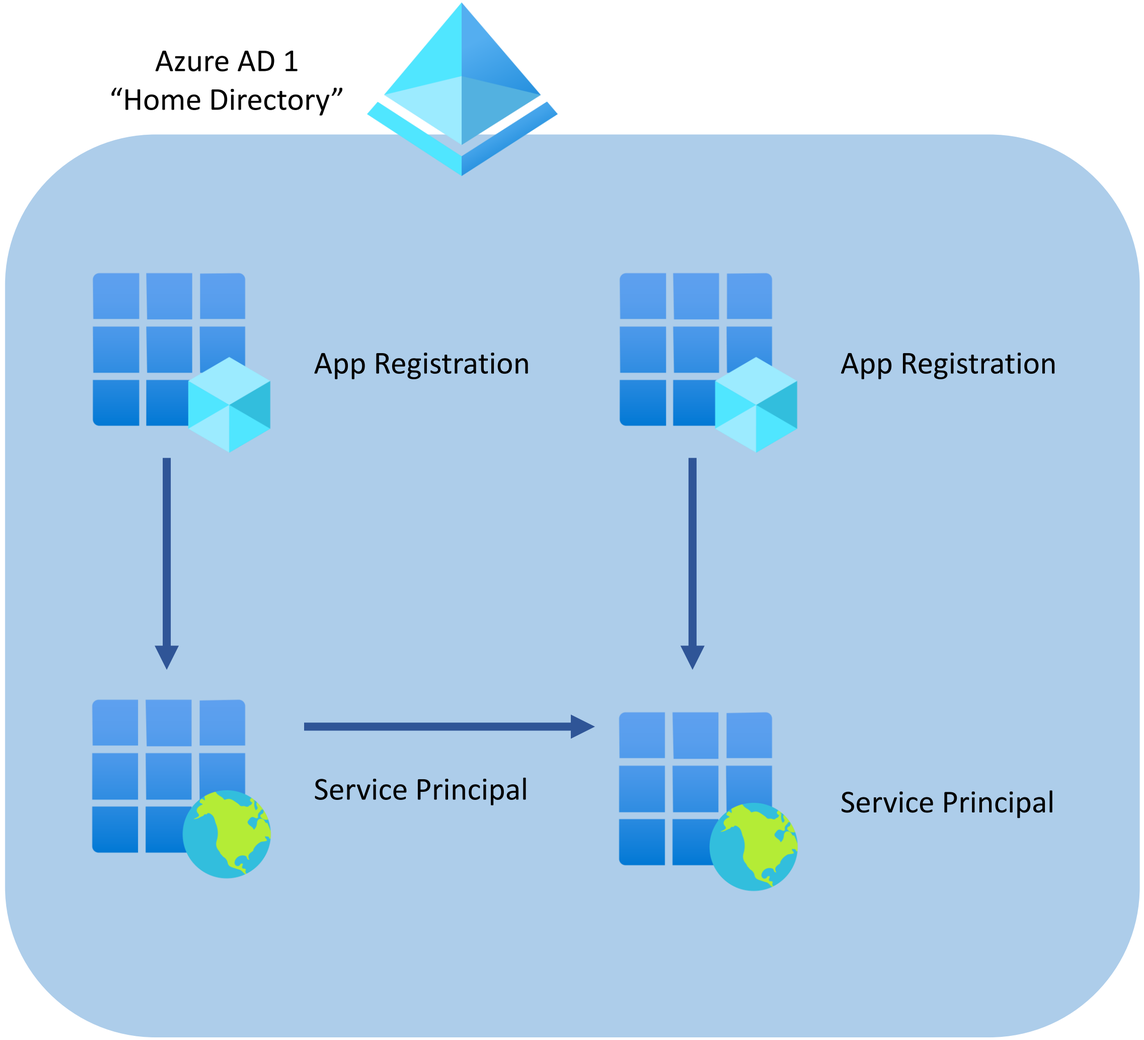
Accessing resources



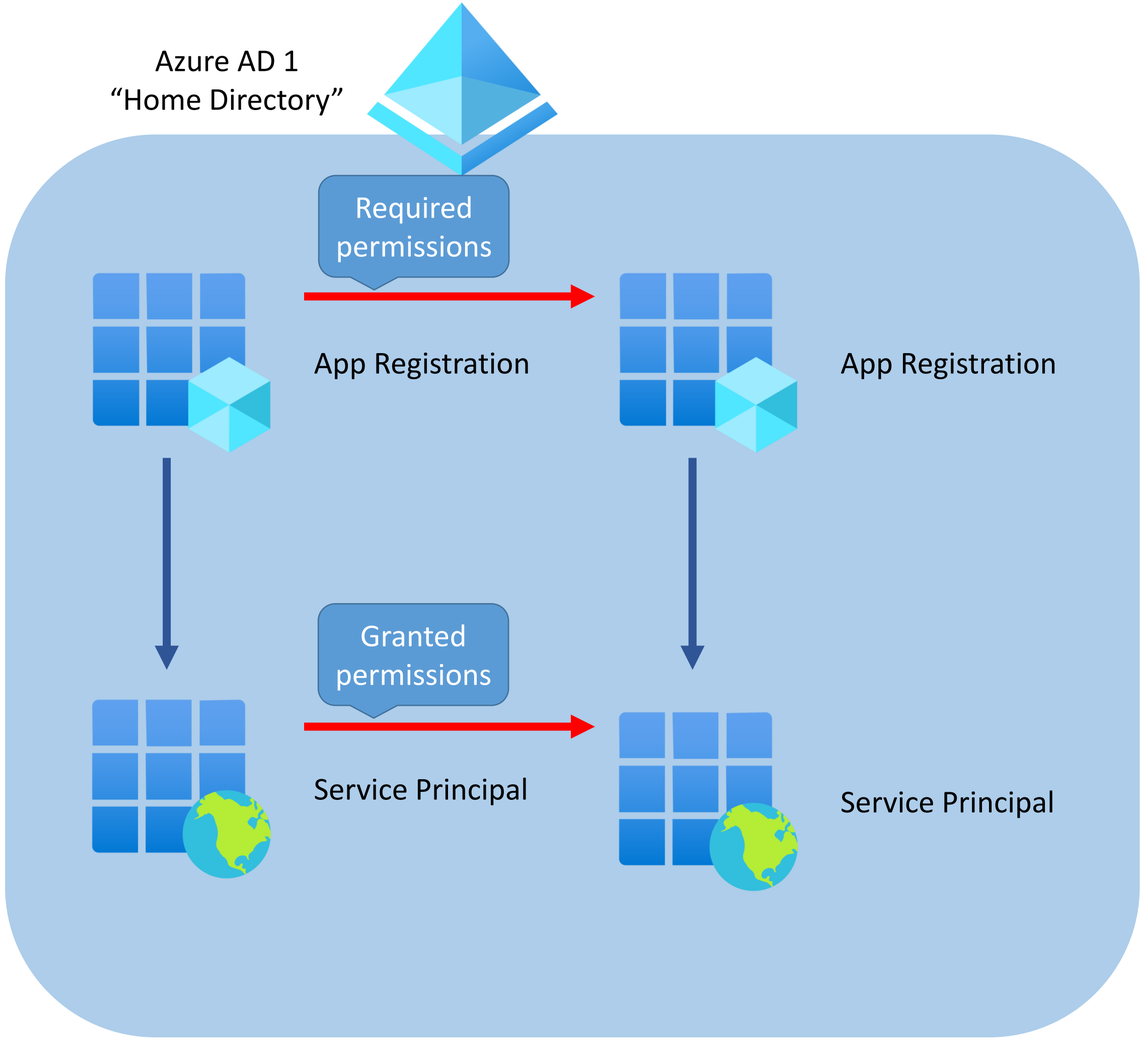
Accessing resources



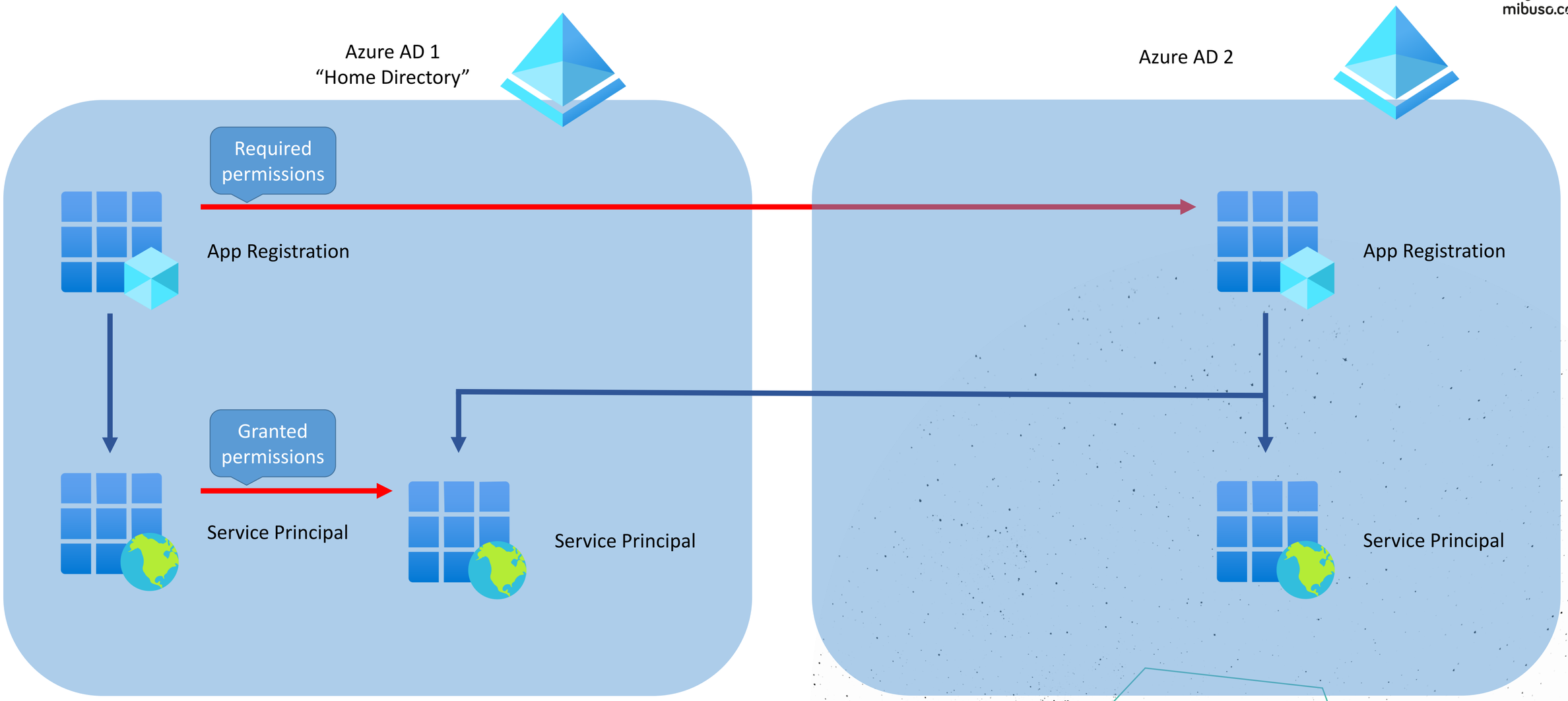
Accessing resources – single tenant



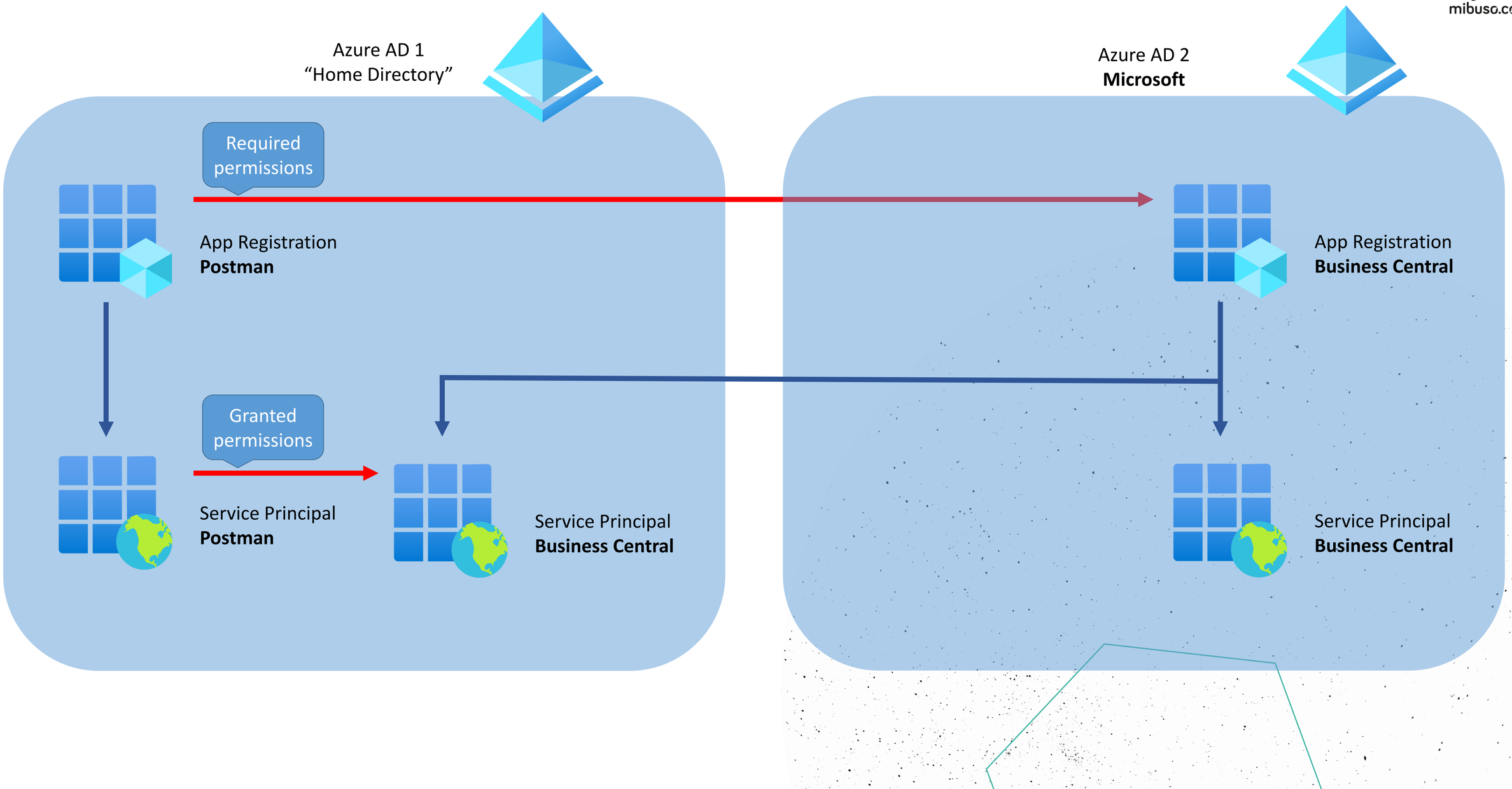
Accessing resources – single tenant



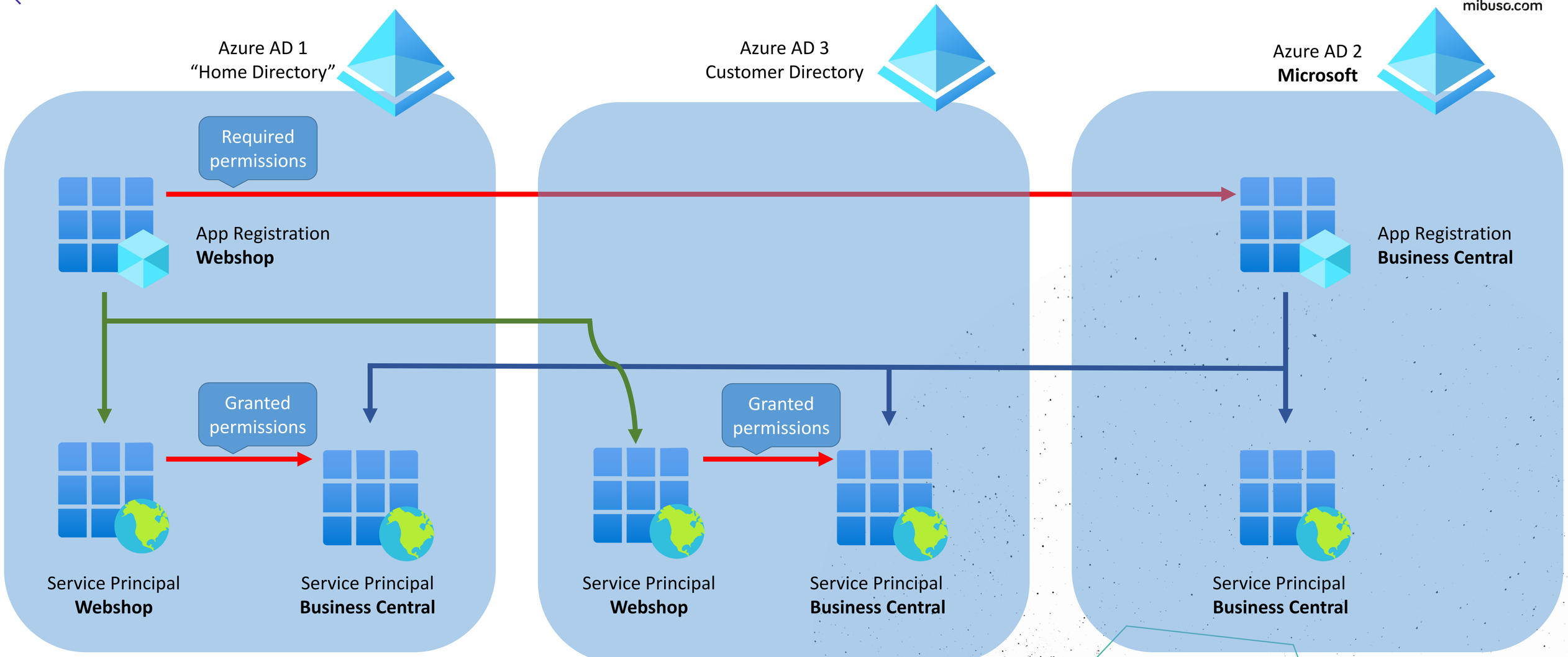
Accessing resources – multitenant



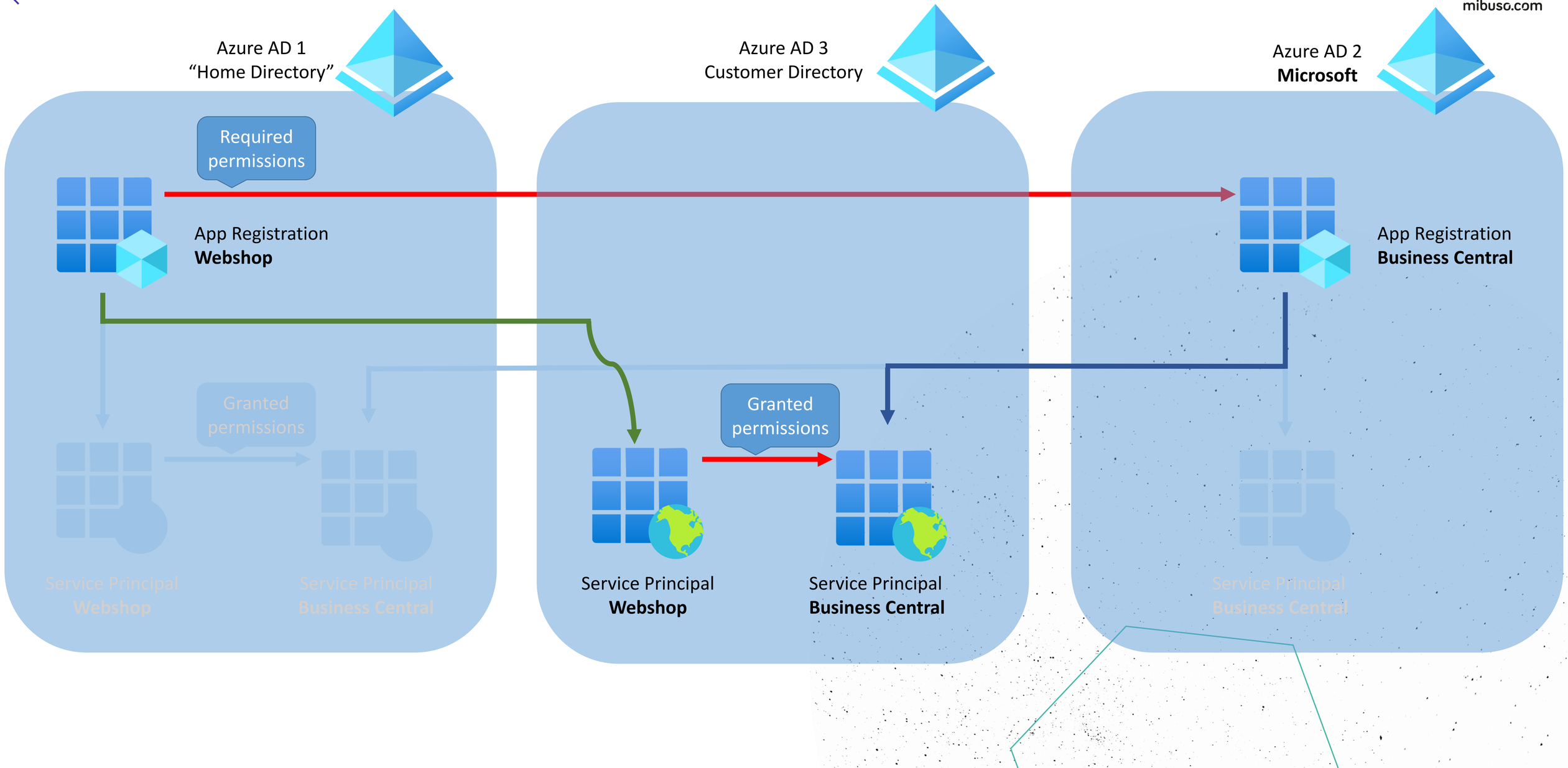
Accessing resources – multitenant



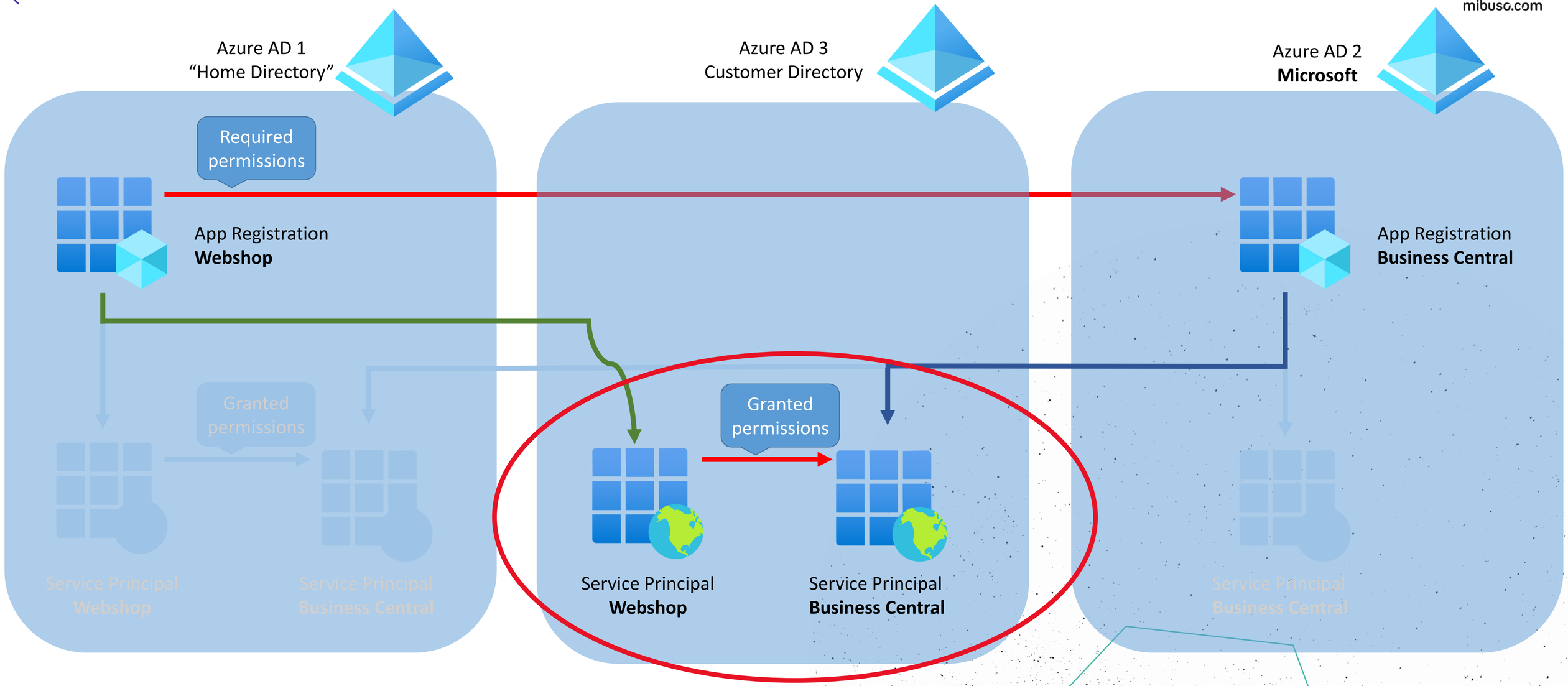
Accessing resources – multitenant



Accessing resources – multitenant



How are these Service Principals created?

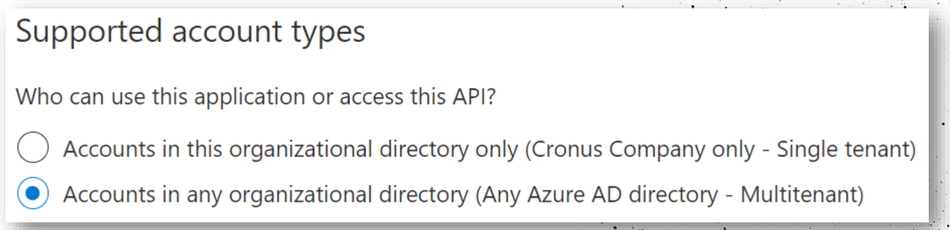


How are these Service Principals created?

BC/NAV TECH DAYS 2022
mibuso.com

- In the **Home Directory**: when the Application is created
  - Using the Azure Portal with **App registration**
  - When using the **Graph API** it requires an extra API request

- In the **foreign Directory**
  - Only when the Application is **multitenant**
  - First time when permissions are **granted**



What are permissions?

BC/NAV TECH DAYS 2022
mibuso.com

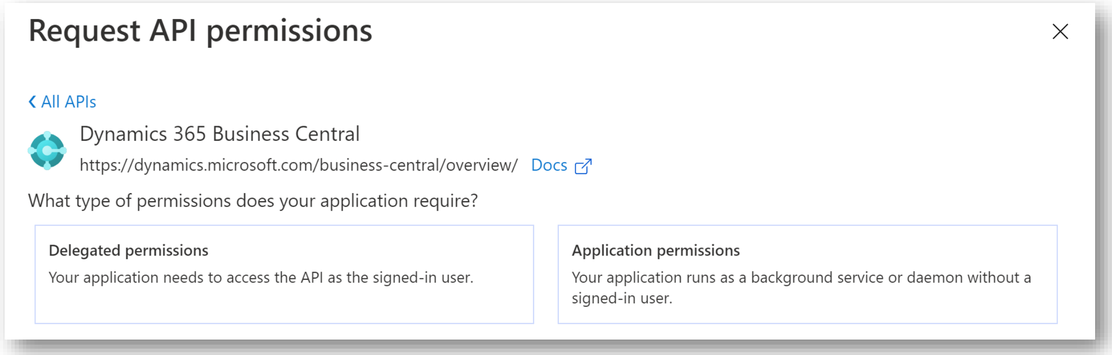
- An application requires **access** to specific **resources**
  - A resource is another app registration that exposes data
  - Exposing data is usually done with a REST endpoint

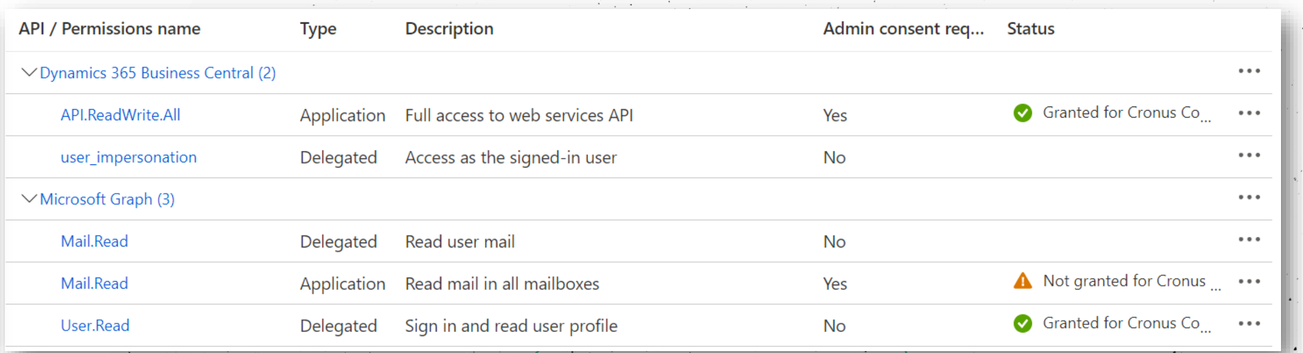
- Access to the exposed data is managed by **Scopes** and **Roles**
  - Scopes are used to access data **on behalf of a user**
  - Roles are used for **access management**

- Two types of Roles
  - User Role – assigned to a user
  - Application Role – special role type, no user assignment required

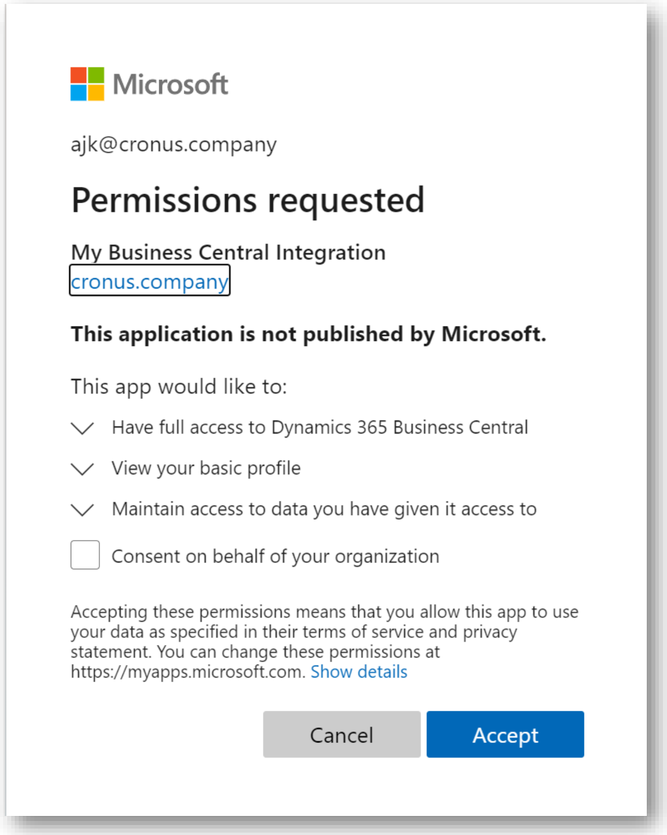
Examples of scopes and roles

- Scope
  - https://api.businesscentral.dynamics.com/user_impersonation
  - https://graph.microsoft.com/Mail.Read
- Application Role
  - https://api.businesscentral.dynamics.com/API.ReadWrite.All
  - https://graph.microsoft.com/Mail.Read





Granting permissions aka consent



User consent flow *is the process of a user* ***granting authorization*** *to an application to access protected resources* ***on their behalf***

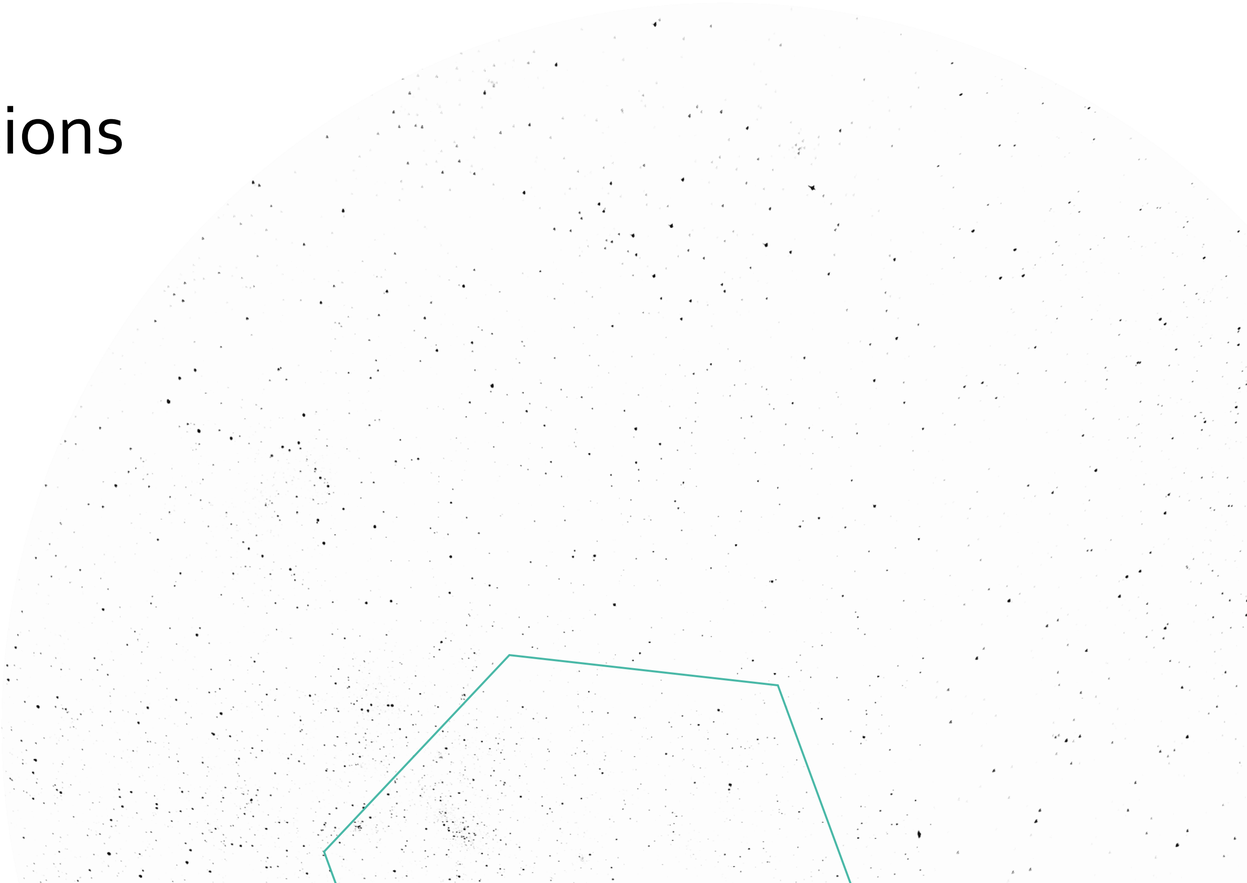
Admin consent flow *is the process of an* ***admin granting authorization*** *to an application to access protected resources* ***for all users***

For an application role type, it is required to grant ***Admin consent***

Consent experiences explained:
https://docs.microsoft.com/en-us/azure/active-directory/develop/application-consent-experience

What did we not touch yet?

- Public app vs. confidential app
  - Redirect URI
  - Secret
- Dynamic permissions vs. static permissions
- Access token
  - Authorization flows



Access token

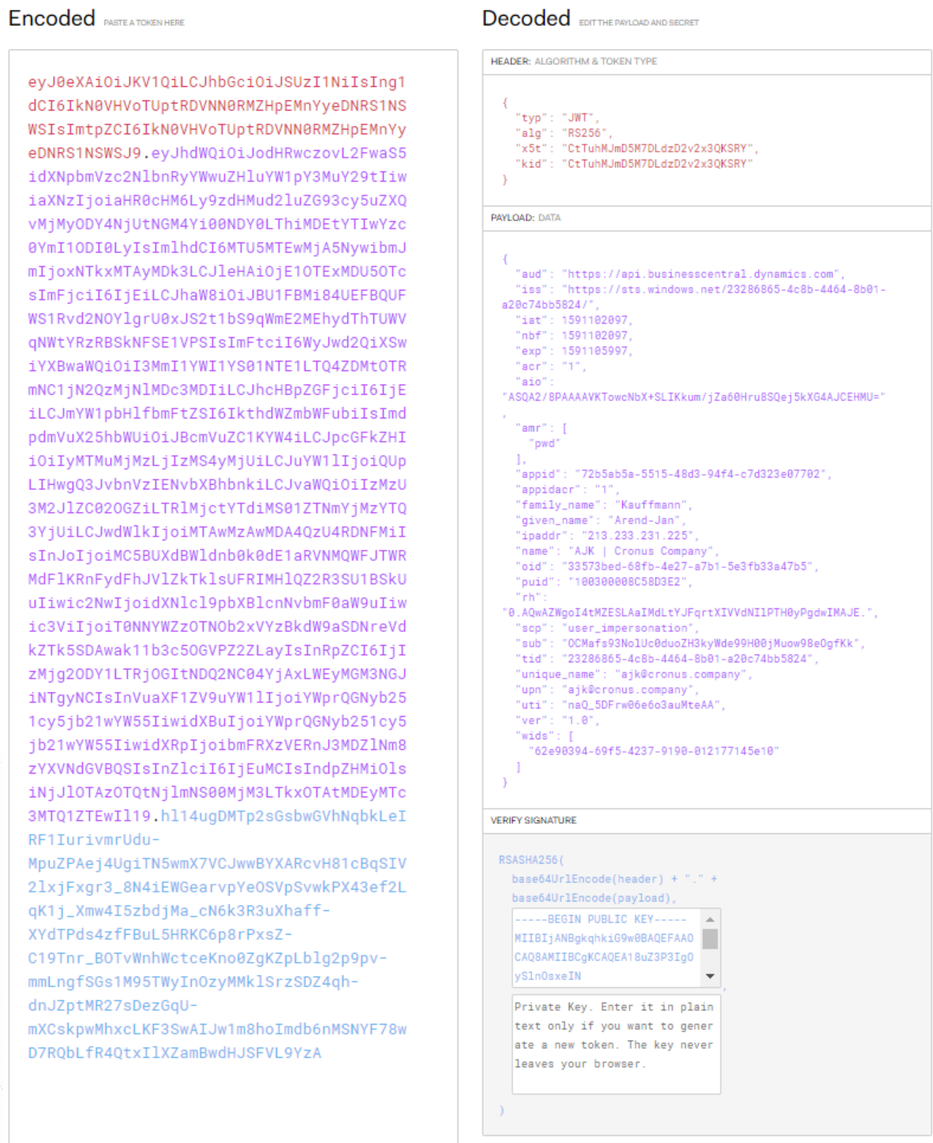
- Access token is used to prove permissions
- Contains all required information
  - User
  - Azure AD (tenant)
  - Granted permissions

```
get https://api.businesscentral.dynamics.com/v2.0/production/api/v2.0/
Authorization: Bearer {{accesstoken}}
```


```
{
  "token_type": "Bearer",
  "scope": "https://api.businesscentral.dynamics.com/user_impersonation",
  "expires_in": 3599,
  "ext_expires_in": 3599,
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiIsIng1dCI6IkN0VHVoTUptRDVNN0RMZHpEMnYyeDNRS1NSWSIsImtpZCI6IkN0VHVoTUptRDVNN0RMZHpEMnYyeDNRS1NSWSJ9.eyJhdWQiOiJodHRwczovL2FwaS5idXNpbmVzc2NlbnRyYWwuZHluYW1pY3MuY29tIiwiaXNzIjoiaHR0cHM6Ly9zdHMud2luZG93cy5uZXQvMjMyODY4NjUtNGM4Yi00NDY0LThiMDEtYTIwYzc0YmI1ODI0LyIsImlhdCI6MTU5MTEwMjA5NywibmJmIjoxNTkxMTAyMDk3LCJleHAiOjE1OTExMDU5OTcsImFjciI6IjEiLCJhaW8iOiJBU1FBMi84UEFBQUFWS1Rvd2NOYlgrU0xJS2t1bS9qWmE2MEhydThTUWVqNWtYRzRBSkNFSE1VPSIsImFtciI6WyJwd2QiXSwiYXBwaWQiOiI3MmI1YWI1YS01NTE1LTQ4ZDMtOTRmNC1jN2QzMjNlMDc3MDIiLCJhcHBpZGFjciI6IjEiLCJmYW1pbHlfbmFtZSI6IkthdWZmbWFubiIsImdpdmVuX25hbWUiOiJBcmVuZC1KYW4iLCJpcGFkZHIiOiIyMTMuMjMzLjIzMS4yMjUiLCJuYW1lIjoiQUpLIHwgQ3JvbnVzIENvbXBhbnkiLCJvaWQiOiIzMzU3M2JlZC02OGZiLTRlMjctYTdiMS01ZTNmYjMzYTQ3YjUiLCJwdWlkIjoiMTAwMzAwMDA4QzU4RDNFMiIsInJoIjoiMC5BUXdBWldnb0k0dE1aRVNMQWFJTWRMdFlKRnFydFhJVlZkTklsUFRIMHlQZ2R3SU1BSkUuIiwic2NwIjoidXNlcl9pbXBlcnNvbmF0aW9uIiwic3ViIjoiT0NNYWZzOTNOb2xVYzBkdW9aSDNreVdkZTk5SDAwak11b3c5OGVPZ2ZLayIsInRpZCI6IjIzMjg2ODY1LTRjOGItNDQ2NC04YjAxLWEyMGM3NGJiNTgyNCIsInVuaXF1ZV9uYW1lIjoiYWprQGNyb251cy5jb21wYW55IiwidXBuIjoiYWprQGNyb251cy5jb21wYW55IiwidXRpIjoibmFRXzVERnJ3MDZlNm8zYXVNdGVBQSIsInZlciI6IjEuMCIsIndpZHMiOlsiNjJlOTAzOTQtNjlmNS00MjM3LTkxOTAtMDEyMTc3MTQ1ZTEwIl19.hl14ugDMTp2sGsbwGVhNqbkLeIRF1IurivmrUdu-MpuZPAej4UgiTN5wmX7VCJwwBYXARcvH81cBqSIV2lxjFxgr3_8N4iEWGearvpYeOSVpSvwkPX43ef2LqK1j_Xmw4I5zbdjMa_cN6k3R3uXhaff-XYdTPds4zfFBuL5HRKC6p8rPxsZ-C19Tnr_BOTvWnhWctceKno0ZgKZpLblg2p9pv-mmLngfSGs1M95TWyInOzyMMklSrzSDZ4qh-dnJZptMR27sDezGqU-mXCskpwMhxcLKF3SwAIJw1m8hoImdb6nMSNYF78wD7RQbLfR4QtxIlXZamBwdHJSFVL9YzA"
}
```

Access token

- Json Web Token
- Base64 encoded value
- Inspect at:
  - https://jwt.io
  - https://jwt.ms
- Signed by a private key
- Used by Business Central to
  - Identify user and tenant
  - Determine access permissions



Access token

- Json Web Token
- Base64 encoded value
- Inspect at:
  - https://jwt.io
  - https://jwt.ms
- Signed by a private key
- Used by Business Central to
  - Identify user and tenant
  - Determine access permissions

PAYLOAD: DATA


```
{
  "aud": "https://api.businesscentral.dynamics.com",
  "iss": "https://sts.windows.net/23286865-4c8b-4464-8b01-
a20c74bb5824/",
  "iat": 1591102097,
  "nbf": 1591102097,
  "exp": 1591105997,
  "acr": "1",
  "aio":
"ASQA2/8PAAAAVKTowcNbX+SLIKkum/jZa60Hru8SQej5kXG4AJCEHMU="
,
  "amr": [
    "pwd"
  ],
  "appid": "72b5ab5a-5515-48d3-94f4-c7d323e07702",
  "appidacr": "1",
  "family_name": "Kauffmann",
  "given_name": "Arend-Jan",
  "ipaddr": "213.233.231.225",
  "name": "AJK | Cronus Company",
  "oid": "33573bed-68fb-4e27-a7b1-5e3fb33a47b5",
  "puid": "10030000BC58D3E2",
  "rh":
"0.AQwAZWgoI4tMZESLAaIMdLtYJFqrtXIVVdNI1PTH0yPgdwIMAJE.",
  "scp": "user_impersonation",
  "sub": "OCMafs93No1Uc0duoZH3kyWde99H00jMuow98eOgfKk",
  "tid": "23286865-4c8b-4464-8b01-a20c74bb5824",
  "unique_name": "ajk@cronus.company",
  "upn": "ajk@cronus.company",
  "uti": "naQ_5DFrw06e6o3auMteAA",
  "ver": "1.0",
  "wids": [
    "62e90394-69f5-4237-9190-012177145e10"
  ]
}
```

How to get the access token

BC/NAV TECH DAYS 2022
mibuso.com

- Implicit grant flow
- Authorization code grant flow
- On-behalf-of flow
- Client credentials flow
- Device code flow
- Resource owner password credential flow
- SAML bearer assertion flow

How to get the access token

BC/NAV TECH DAYS 2022
mibuso.com

- Implicit grant flow
- ✓ Authorization code grant flow
- On-behalf-of flow
- ✓ Client credentials flow
- Device code flow
- Resource owner password credential flow
- SAML bearer assertion flow

Authorization flow

BC/NAV TECH DAYS 2022
mibuso.com

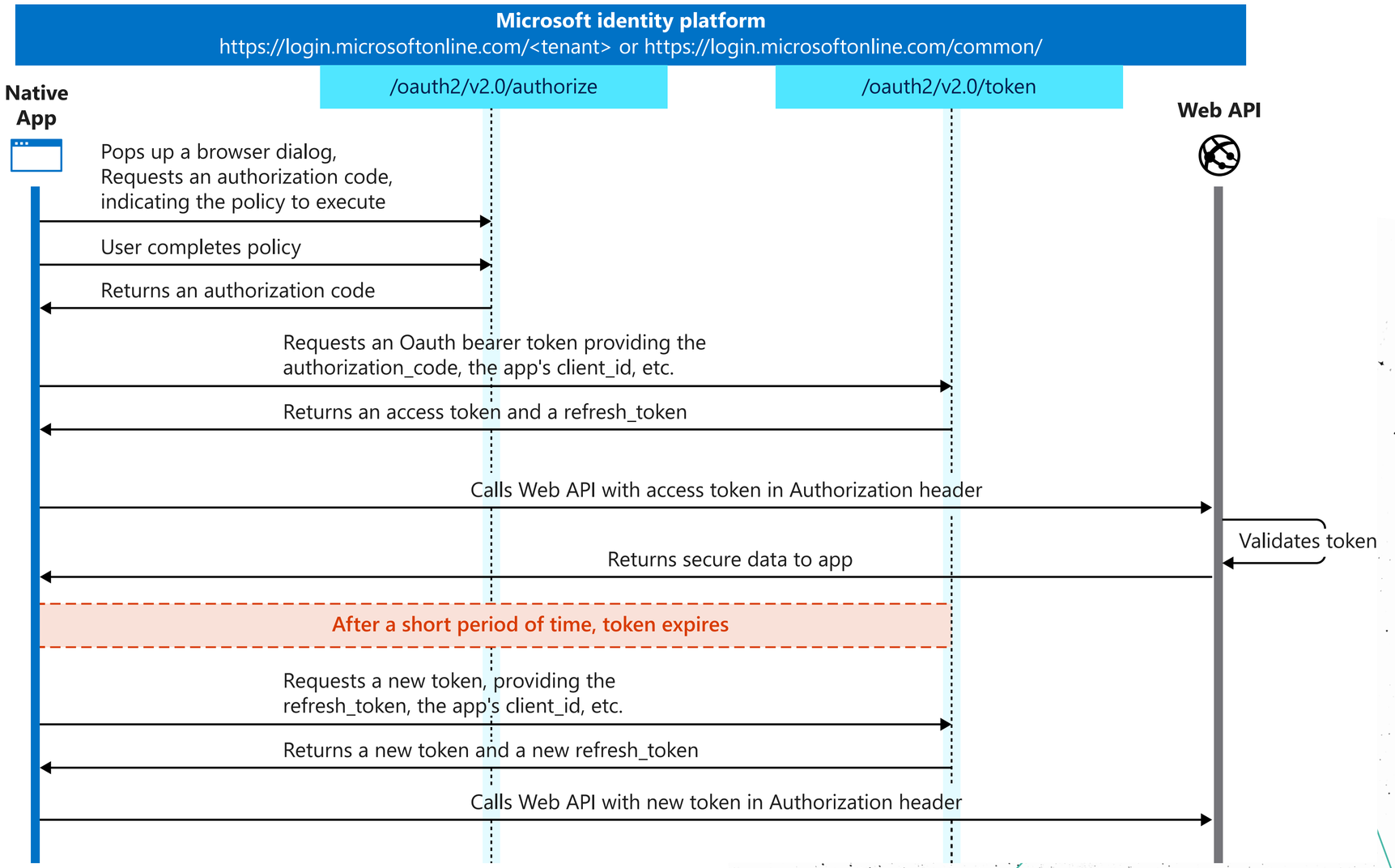
Authorization code grant flow

- Used in a user-interactive session
- Requires user to log in
- Access resource on behalf of the user
- Permission type = scope (aka delegated)
- Needs a user account in the resource application

Client credentials flow

- Aka service to service authentication
- Used in a background session with no signed-in user
- Permission type = application role
- Needs an application account in the resource application

Authorization code grant flow



Authorization code grant flow – step 1

BC/NAV TECH DAYS 2022
mibuso.com

```
GET https://login.microsoftonline.com/cronus.company/oauth2/v2.0/authorize (1)
?client_id=72b5ab5a-5515-48d3-94f4-c7d323e07702 (2)
&response_type=code (3)
&redirect_uri=https://login.microsoftonline.com/common/oauth2/nativeclient (4)
&response_mode=query (5)
&scope=https://api.businesscentral.dynamics.com/user_impersonation (6)
```

Comments

1. Domain or Azure tenant id. Use **organizations** for multitenant applications
2. The client id (aka application id) as assigned in the app registration
3. Indicates that you expect an authorization code as response
4. Where you want the authorization code to be returned
5. How the authorization code should be added to the redirect uri
6. Which permission you are requestion

Authorization code grant flow – step 2

BC/NAV TECH DAYS 2022
mibuso.com

```
GET https://login.microsoftonline.com/common/oauth2/nativeclient 1
?code=0.AYIA...  2
&session_state=7e5b63be-21b1-4821-9455-d235e2738002 3
```

Comments

1. The redirect URI as defined in the request in step 1
2. An authorization code that can be redeemed for an access token
3. Optional parameter that can be used for single sign out

Authorization code grant flow – step 3

BC/NAV TECH DAYS 2022
mibuso.com

```
POST https://login.microsoftonline.com/cronus.company/oauth2/v2.0/token 1
Content-Type: application/x-www-form-urlencoded 2

grant_type=authorization_code 3
&code=0.AYIA... 4
&client_id=72b5ab5a-5515-48d3-94f4-c7d323e07702 5
&redirect_uri=https://login.microsoftonline.com/common/oauth2/nativeclient
&scope=https://api.businesscentral.dynamics.com/user_impersonation
&client_secret=n6F8Q~CfC5fclnC84BWQOBtjxUN~OhHUep8ZobLE 6
```

Comments

1. POST request to the token endpoint
2. Content-Type of the body, always application/x-www-form-urlencoded
3. The authorization flow type that is used
4. The authorization code as retrieved in the previous step
5. The client details used in the first step to request the authorization code
6. The client secret to identify the client (= password)

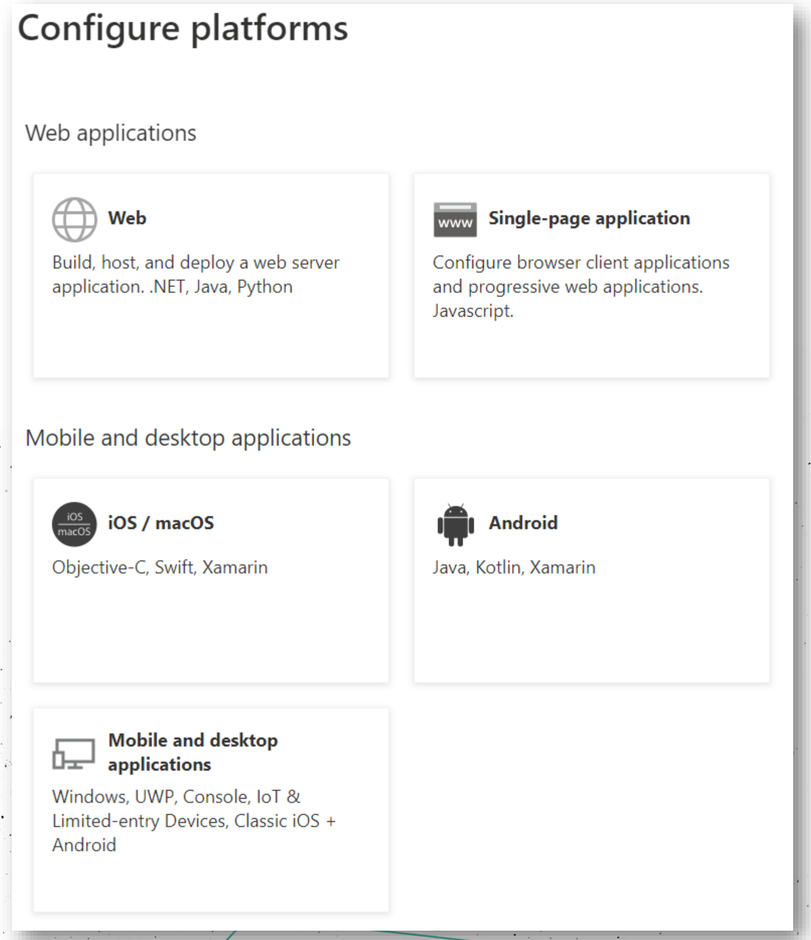
Your PC ran into a problem that it couldn't handle, and now it needs to restart We're just collecting some error info, and then we'll restart for you.

If you'd like to know more, you can search online later for this error: DPC_WATCHDOG_VIOLATION

Redirect URI

- Is required for user interactive flows
- Used by Azure to send the user back to the requesting application
- Extra parameters are appended to the URI
  - Authorization code
  - Session state

- Registered under a platform
- Platform type determines if a secret or certificate is required



Confidential or Public app?

BC/NAV TECH DAYS 2022
mibuso.com

Referred to as platform

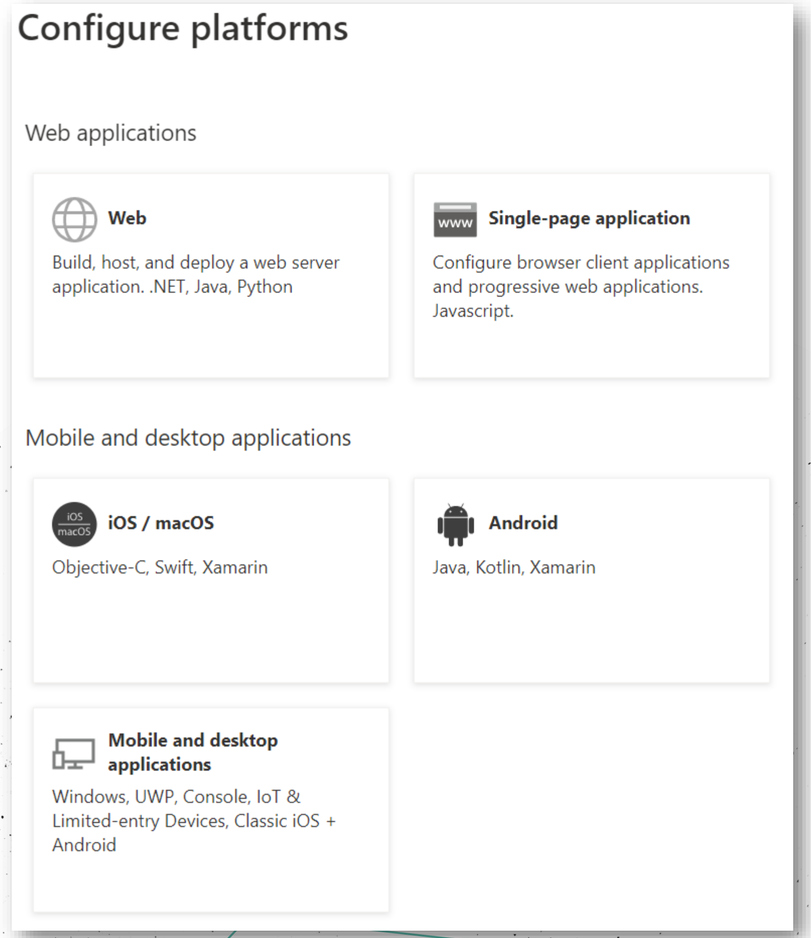
- Web application = confidential app
- Single-page application, mobile and desktop application = public app

Confidential app

- Runs on a server
- No direct access by the user
- Can keep a secret
- Authorization code travels from client to server

Public application

- Runs on a device or desktop
- Or in a web browser (single-page app)
- Not trusted to keep a secret
- Authorization code is kept inside the application



Client credentials flow

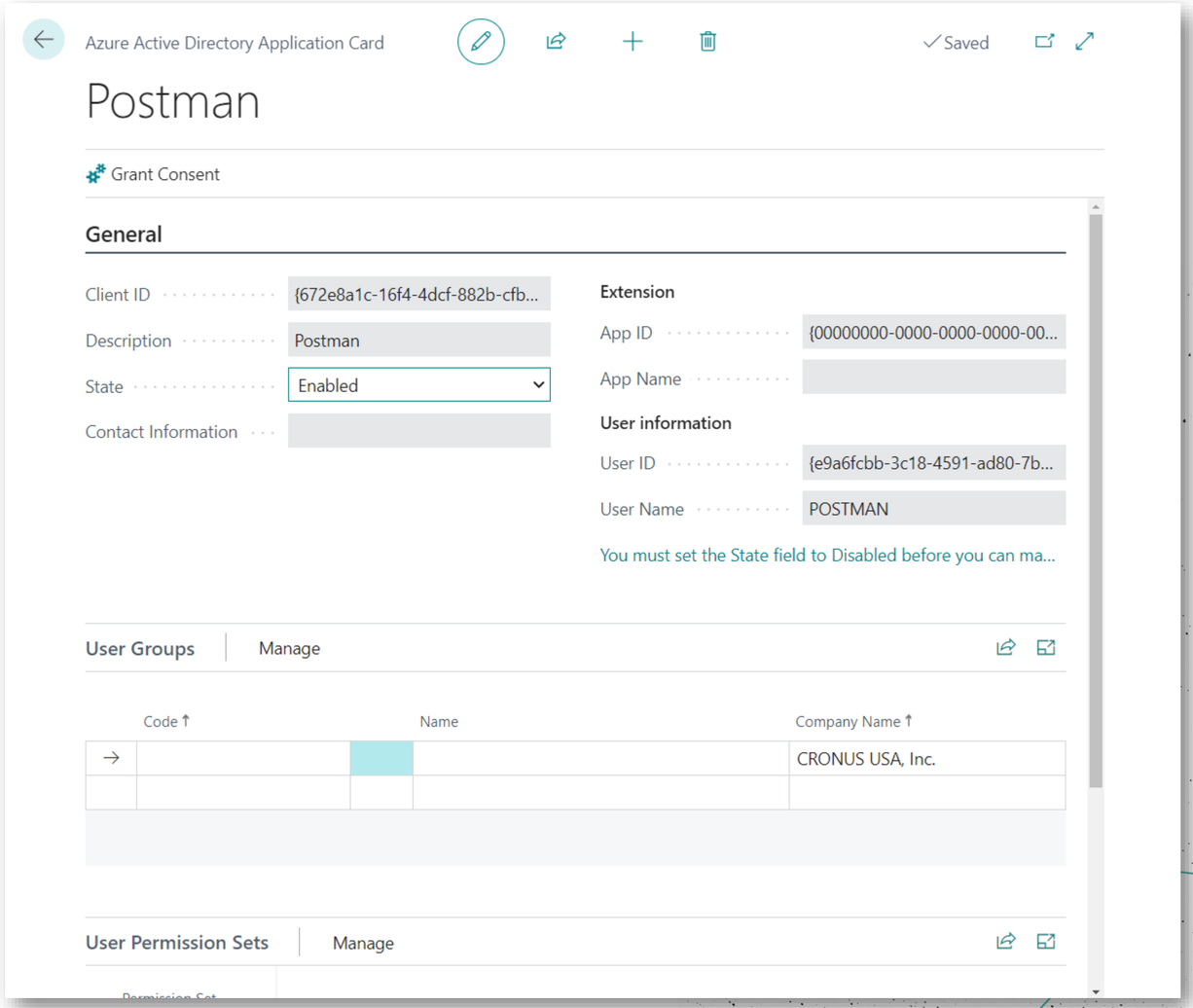
```
POST https://login.microsoftonline.com/cronus.company/oauth2/v2.0/token 1
Content-Type: application/x-www-form-urlencoded 2

grant_type=client_credentials 3
&client_id=72b5ab5a-5515-48d3-94f4-c7d323e07702 4
&client_secret=n6F8Q~CfC5fclnC84BWQOBtjxUN~OhHUep8ZobLE
&scope=https://api.businesscentral.dynamics.com/.default 5
```

Comments

1. POST request to the token endpoint. Domain or Azure AD tenant-id is required!
2. Content-Type of the body, always application/x-www-form-urlencoded
3. The authorization flow type that is used
4. The client id and secret
5. Scope must end with /.default, this refers to the required application permissions as configured in the app registration

Application account in Business Central



Secrets and certificates

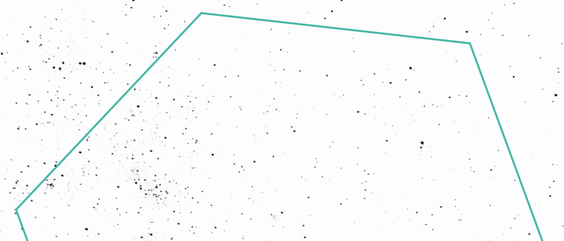
BC/NAV TECH DAYS 2022
mibuso.com

Secrets

- Is a password, used to prove the identity of a client application
- Part of the request body when requesting an access token
- Generated by Azure
- Max lifetime of 24 months
- Should NOT be stored in code
- Recommended to only use in development / test scenarios

Certificates

- More secure way to prove identity
- JWT token, signed with the private key of the certificate
- Never leaves the application
- Lifetime depends on the certificate
- Recommended for production scenarios



Static vs. dynamic permissions

BC/NAV TECH DAYS 2022
mibuso.com

Static permissions

- Specify all required permissions in the Azure portal
- Stored at app registration level and copied to all service principals
- Allows administrator to pre-consent for the organization
- Required for service-to-service authentication
- Scope: https://api.businesscentral.dynamics.com/.default

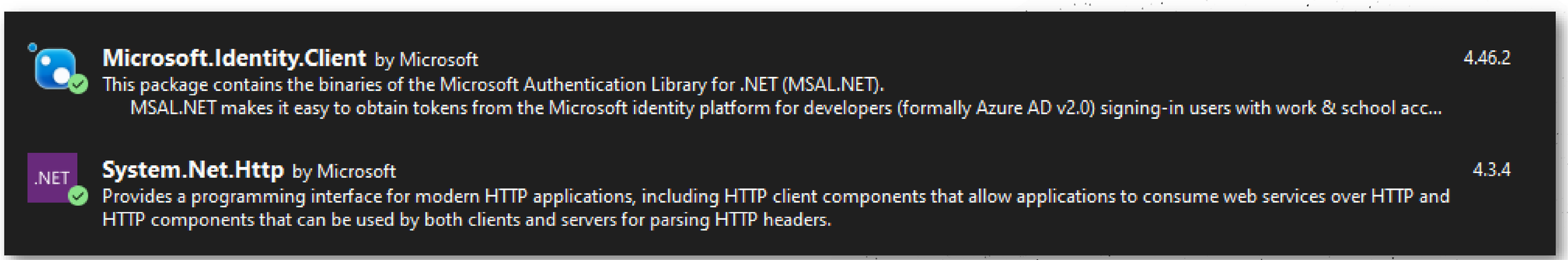
Dynamic permissions

- Specify required permissions during authorization flow
- Only stored at the service principal
- Admin user can consent for whole organization
- Allows to dynamically and gradually request permissions
- Scope: https://api.businesscentral.dynamics.com/API.ReadWrite.All

Using OAuth in C#

BC/NAV TECH DAYS 2022
mibuso.com

- Add Nuget packages
  - Microsoft.Identity.Client → MSAL.NET library
  - System.Net.Http → includes the HttpClient



Using OAuth in C#

BC/NAV TECH DAYS 2022
mibuso.com

- Create a client application object
  - Choose between public client or confidential client

```
static string clientId = "cd25daa5-8463-466c-94ce-1c541dcc1c79";
static string clientSecret = "mR28Q~EYuEjwROi2n2SOvmY43RlovQbvYXshjcn_";
static string azureTenant = "cronus.company";
static string[] scopesDynamic = new string[] { "https://api.businesscentral.dynamics.com/user_impersonation offline_access" };
static string[] scopesStatic = new string[] { "https://api.businesscentral.dynamics.com/.default" };


static IPublicClientApplication publicApp =
    PublicClientApplicationBuilder
        .Create(clientId)
        .WithTenantId(azureTenant)
        .WithDefaultRedirectUri()
        .Build();

static IConfidentialClientApplication confidentialApp =
    ConfidentialClientApplicationBuilder
        .Create(clientId)
        .WithClientSecret(clientSecret)
        .WithAuthority(AzureCloudInstance.AzurePublic, AadAuthorityAudience.AzureAdMultipleOrgs)
        .Build();
```

Using OAuth in C#

BC/NAV TECH DAYS 2022
mibuso.com

- Call the AcquireToken… function to get an access token

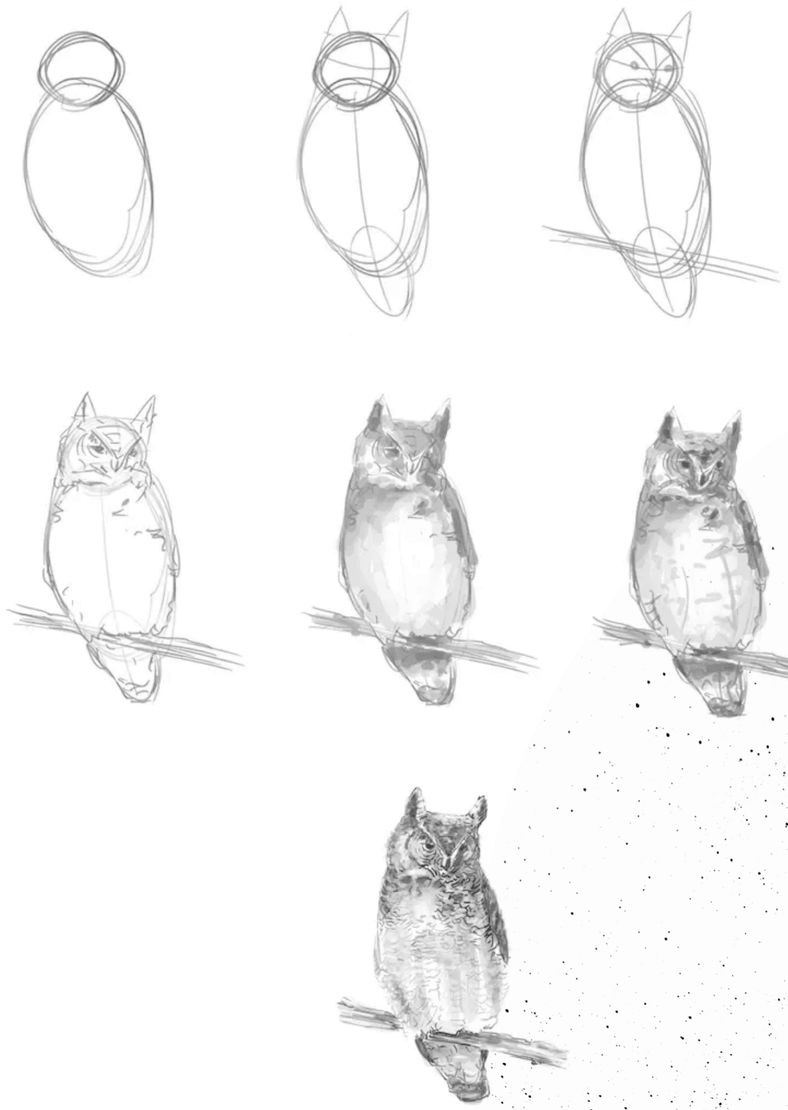
- Public client supports:
  - Interactive flow
  - Silent flow

```
result = publicApp.AcquireTokenInteractive(scopesDynamic)
                  .ExecuteAsync()
                  .Result;
```

- Confidential client supports:
  - Get by authorization code
  - With client credentials

```
result = confidentialApp.AcquireTokenForClient(scopesStatic)
                .WithAuthority(AzureCloudInstance.AzurePublic, azureTenant)
                .ExecuteAsync()
                .Result;
```

How to draw an owl



BC/NAV TECH DAYS 2022
mibuso.com

Any Questions?

BC/NAV TECH DAYS 2022
mibuso.com

Thank You!