NORDIC INTEGRATION SUMMIT

STOCKHOLM

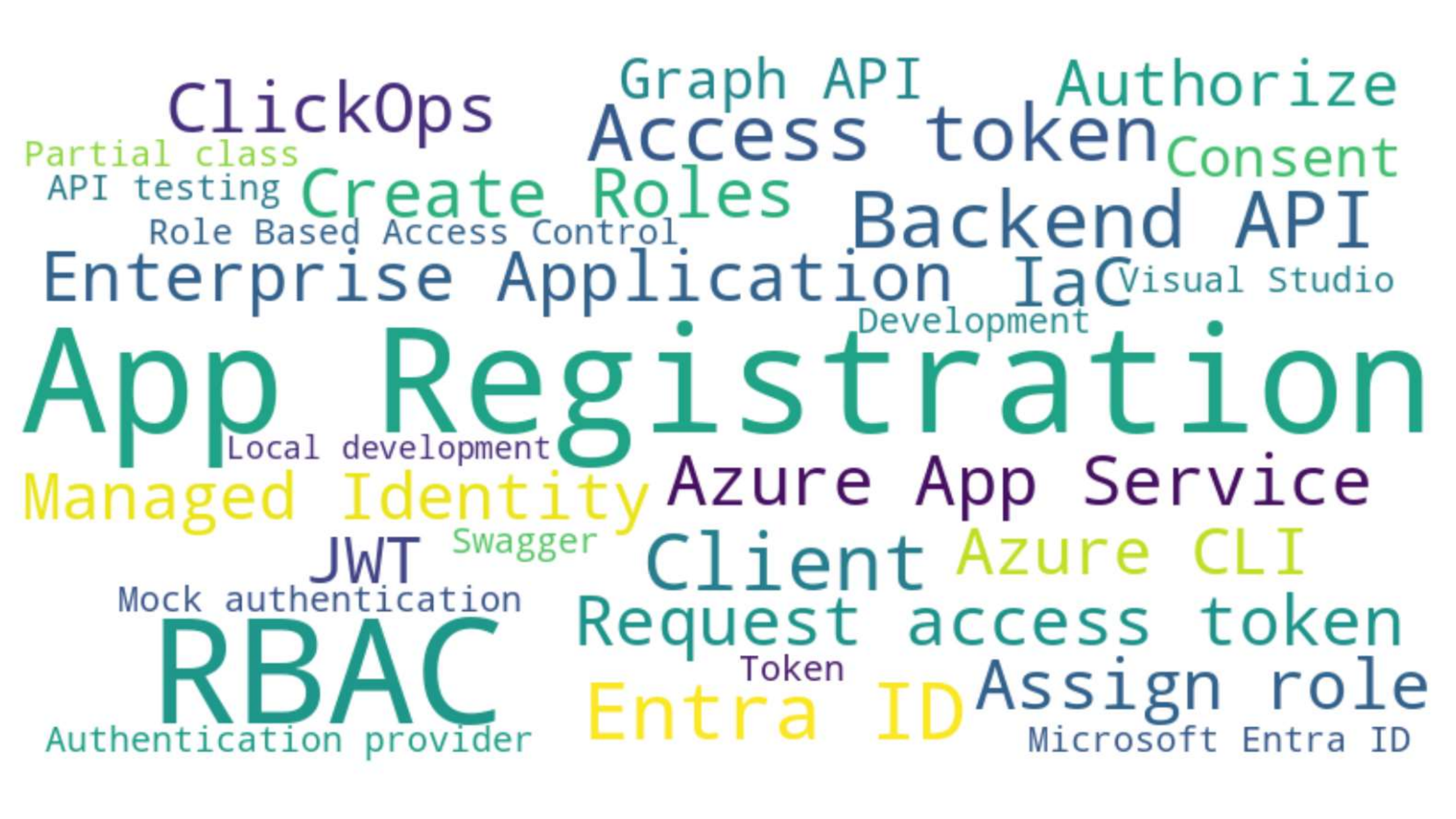
OCTOBER 15-16

WELCOME

Key-less authorization between Azure App Services

Mattias Lindberg

Lepton IT


ClickOps
Graph API
Authorize
Access token
Partial class
Consent
API testing
Create Roles
Backend API
Role Based Access Control
Enterprise Application
IaC
Visual Studio
Development
App Registration
Local development
Azure App Service
Managed Identity
Swagger
JWT
Client
Azure CLI
Mock authentication
Request access token
RBAC
Token
Assign role
Entra ID
Authentication provider
Microsoft Entra ID

What?
Why?
How?

What?

Enable RBAC (Role Based Access Control) when interacting with an API running in an Azure App Service.

Why?

We want to avoid managing access keys in our applications, it is more secure and less work to use a key-less communication pattern.

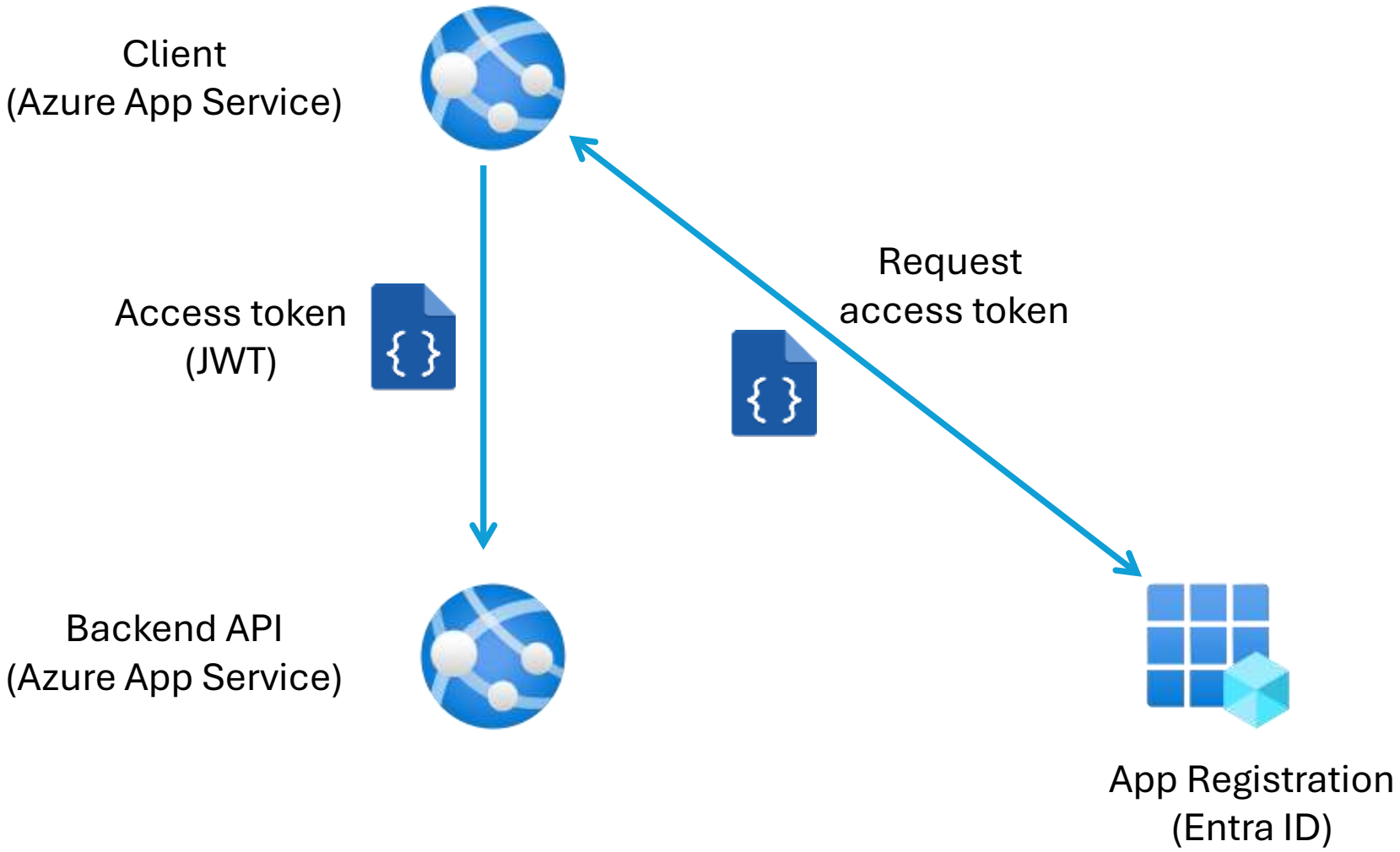
How?

Client

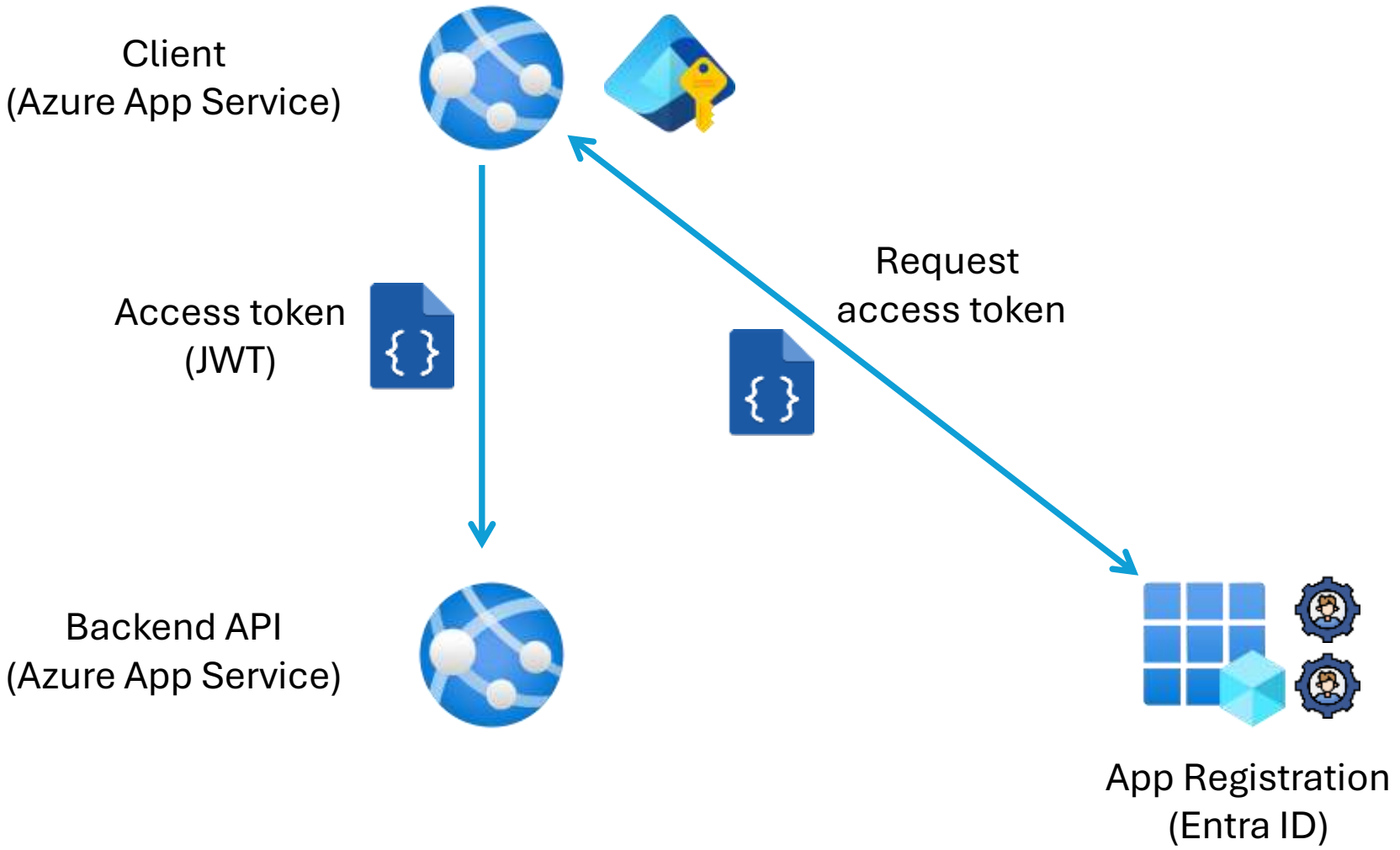
Access token
(JWT)

Backend API
(Azure App Service)

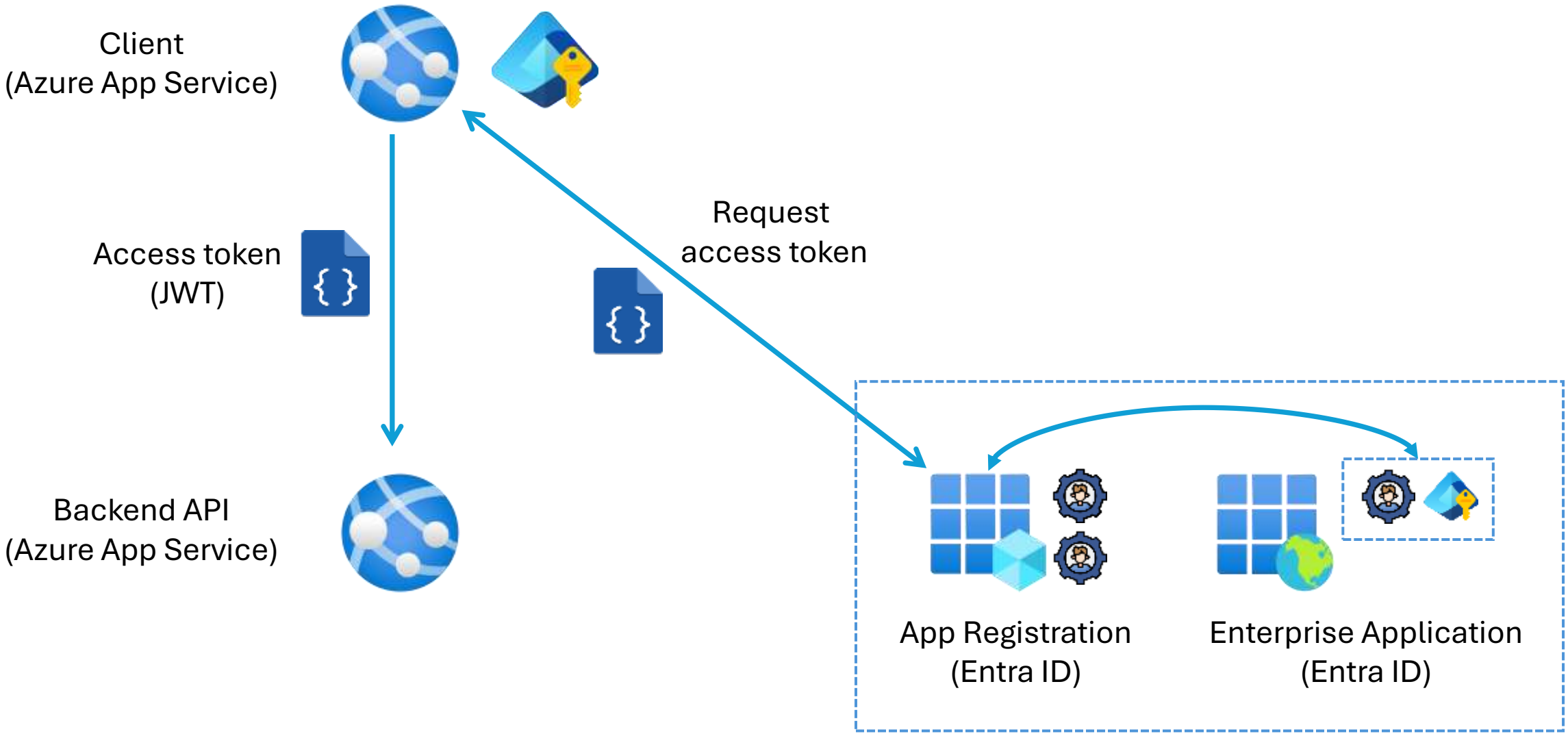
How?


Client
(Azure App Service)
Access token
(JWT)
Request
access token
Backend API
(Azure App Service)
App Registration
(Entra ID)

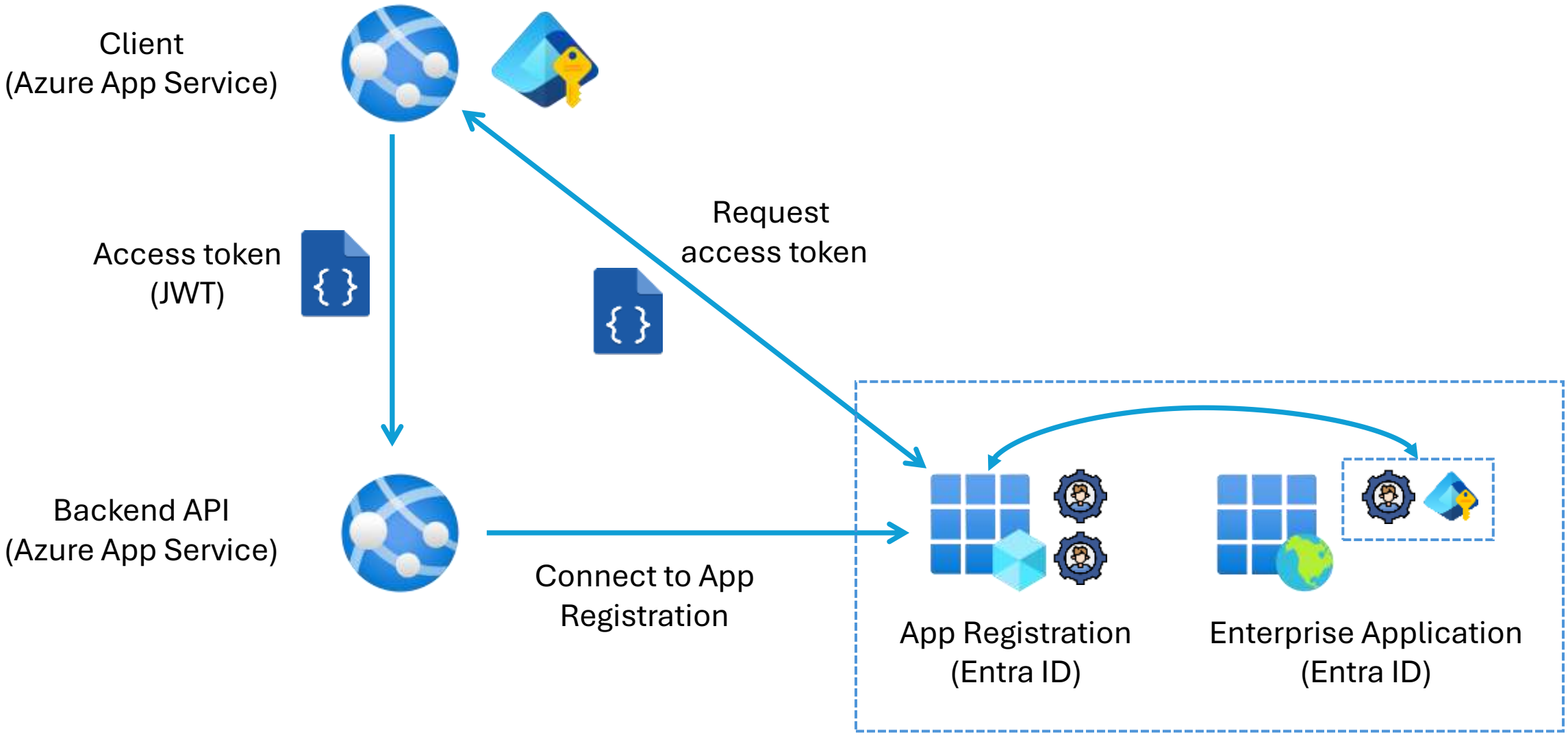
How?


Client
(Azure App Service)
Access token
(JWT)
Request
access token
Backend API
(Azure App Service)
App Registration
(Entra ID)

How?


Client
(Azure App Service)
Access token
(JWT)
Request
access token
Backend API
(Azure App Service)
App Registration
(Entra ID)
Enterprise Application
(Entra ID)

How?


Client
(Azure App Service)
Access token
(JWT)
Request
access token
Backend API
(Azure App Service)
Connect to App
Registration
App Registration
(Entra ID)
Enterprise Application
(Entra ID)

Configuration and Code

Steps in Microsoft Entra ID

ClickOps vs IaC

1. Create the App Registration in Entra ID
   - Manual work by someone with priviledged permissions in Entra ID
   - Not IaC due to limited permissions for build agents/service connections
   - Ask creator to make you Owner of the App Registration and Enterprise Application

2. Create *Application Id URI* for the App Registration
   - ”Secure” GUID vs human readable so it can be verified

3. Create Roles in the App Registration

4. Assign the role in the App Reg to the Managed Identity of the client
   - Graph API or Powershell

Steps in Microsoft Entra ID

1. Create the App Registration in Entra ID
   - Manual work by someone with priviledged permissions in Entra ID
   - Not IaC due to limited permissions for build agents/service connections
   - Ask creator to make you Owner of the App Registration and Enterprise Application

2. Create *Application Id URI* for the App Registration
   - ”Secure” GUID vs human readable so it can be verified

3. Create Roles in the App Registration

4. Assign the role in the App Reg to the Managed Identity of the client
   - Graph API or Powershell

Assign a role in an App Registration to the Managed Identity of the client

```
# ObjectId of the Managed Identity for the Client App Service, the client that should be assigned the role.
$WebAppManagedIdentity = $principalId

# ObjectId of the Enterprise Application (NOT the App Registration).
$EntAppObjectId = $servicePrincipal.Id

# Id property of the Reader role
# Taken from the Manifest JSON available in Azure Portal.
$RoleIdInAppReg = $contributorRoleId

# Execute Graph API call using Azure CLI
az rest -m POST -u https://graph.microsoft.com/v1.0/servicePrincipals/$WebAppManagedIdentity/appRoleAssignments
-b "{'principalId': '$WebAppManagedIdentity', 'resourceId': '$EntAppObjectId', 'appRoleId': '$RoleIdInAppReg'}"
```

Documentation: Assign a managed identity access to an application role (PS1 and CLI)

https://learn.microsoft.com/en-us/entra/identity/managed-identities-azure-resources/how-to-assign-app-role-managed-identity?pivots=identity-mi-app-role-cli

Steps in Backend API

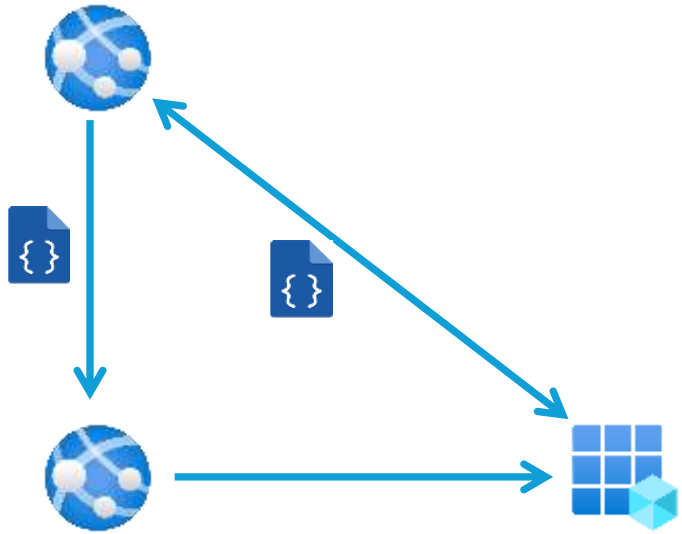
1. Request access token from App Registration

```
builder.Services.AddAuthentication( ... ).AddJwtBearer( ... )
```

2. Assert role when API endpoint is called

```
[Authorize]
[Authorize(Roles = ”Contributor”)]
[Authorize(Roles = ”Reader,Contributor”)]
```

Steps in Client



1. Request access token from App Registration

   ```
   new DefaultAzureCredentials().GetToken( ... )
   ```

2. Include access token in call to backend API

   Set the Authorization HTTP Header in call to backend API

   How do you set it when using an API imported from Swagger?

What does a JWT look like?

```
{
  "aud": "api://nis2024-demo",
  "iss": "https://sts.windows.net/f6de2f9d-0df6-4d10-b7df-ac7351225429/",
  "iat": 1729010455,
  "nbf": 1729010455,
  "exp": 1729016078,
  "acr": "1",
  "aio": "AVQAq/8YAAAACbaDZSa0Tz9rk5YJJezIZbPagZfp8jjjlfnICnmSCSnIJiFMeoCS0nWp8eT0Og9woaY",
  "amr": [
    "pwd",
    "rsa",
    "mfa"
  ],
  "appid": "04b07795-8ddb-461a-bbee-02f9e1bf7b46",
  "appidacr": "0",
  "deviceid": "31a1bcda-7581-433f-b127-d02f2dd91ce3",
  "family_name": "Lindberg",
  "given_name": "Mattias",
  "ipaddr": "90.225.164.34",
  "name": "Mattias Lindberg",
  "oid": "ffa9a7f9-5dbc-430a-a871-ee0e628ae9a3",
  "rh": "",
  "roles": [
    "Contributor"
  ],
  "scp": "Developer.Debug",
  "sub": "",
  "tid": "
  "unique_name": "mattias@lepton.se",
  "upn": "mattias@lepton.se",
  "uti": "dvuVa2uX0kaBqN4SqT5YAA",
  "ver": "1.0"
}
```

audience == scope ➔ Application ID URI

Roles assigned

Summary of configuration & code

Microsoft Entra ID

- Create and configure App Registration and Enterprise Application

Backend API

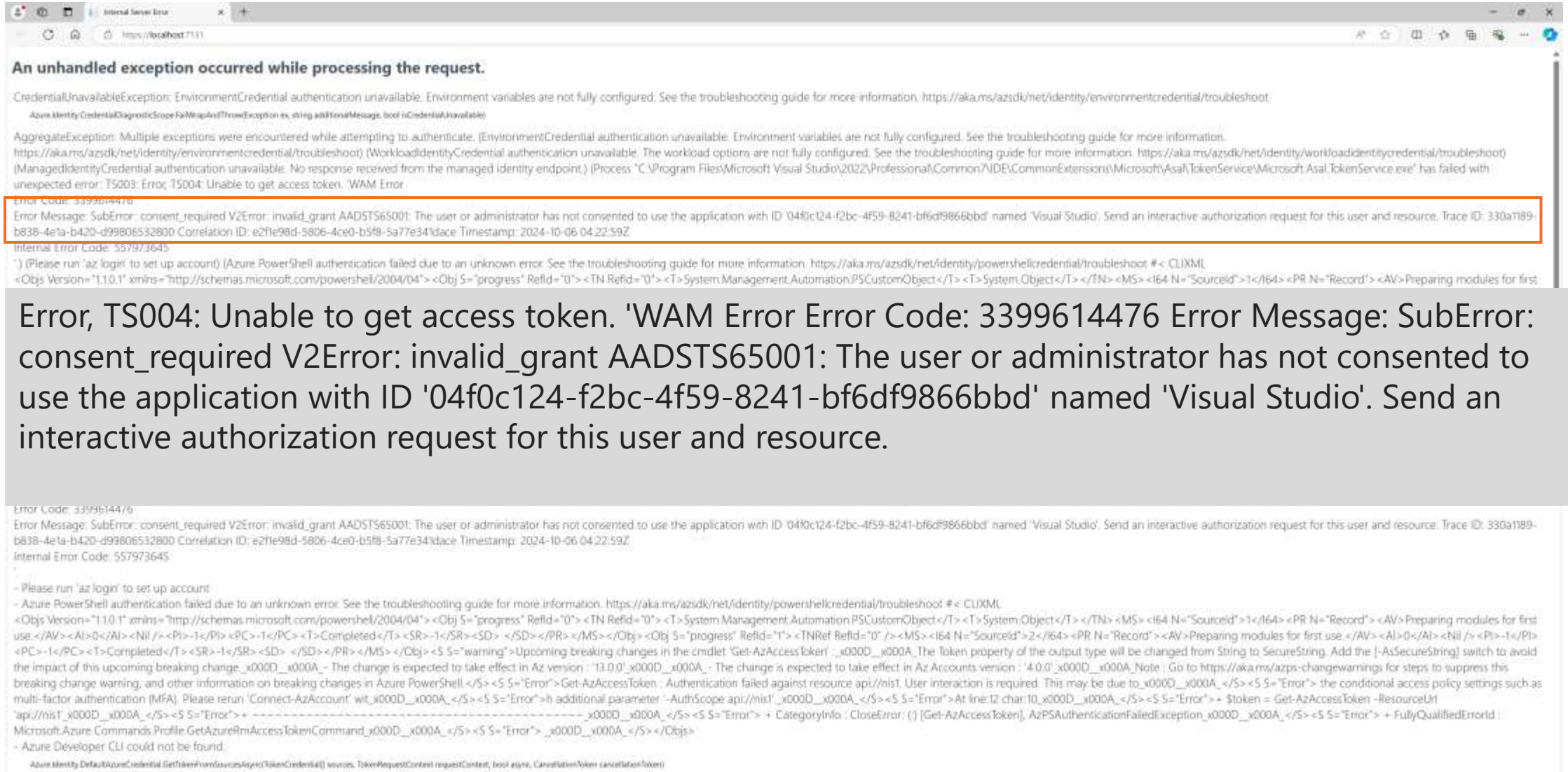
- Register the App Registration as authentication provider
  `builder.Services.AddAuthentication( ... ).AddJwtBearer( ... )`
- Use Authorize attribute to control access

Client application

- Get token from App Registration
  `new DefaultAzureCredentials().GetToken( ... )`
- Use partial class/method to inject code in generated code

Additional steps for developers running this locally

Missing consent to authenticate



Error, TS004: Unable to get access token. 'WAM Error Error Code: 3399614476 Error Message: SubError: consent_required V2Error: invalid_grant AADSTS65001: The user or administrator has not consented to use the application with ID '04f0c124-f2bc-4f59-8241-bf6df9866bbd' named 'Visual Studio'. Send an interactive authorization request for this user and resource.

Missing consent to authenticate

- To run applications in Azure the steps above are enough
- When running applications locally, e.g. in Visual Studio, you will have an issue

- This only needs to be resolve ONCE per App Registration
  - Does not need to be repeated per developer
- Again, this is a step that may require assistance from your IT Department

Missing consent to authenticate

- Configure consent URI
  - App Registration > Authentication > Add a platform > Web
    - Redirect URI: https://global.consent.azure-apim.net/redirect
- Define a scope to be used for Consent
  - App Registration > Expose an API > Add a scope
    - Define a scope to be used for consent
    - I have used ”Developer.Debug” which is reflected in the URI below
- Issue a consent request using Azure CLI

```
az login --scope api://nis2024-demo-05/Developer.Debug
```

Traces of the consent process in the JWT


```
{
  "aud": "api://nis2024-demo",
  "iss": "https://sts.windows.net/f6de2f9d-0df6-4d10-b7df-ac7351225429/",
  "iat": 1729010455,
  "nbf": 1729010455,
  "exp": 1729016078,
  "acr": "1",
  "aio": "AVQAq/8YAAAACbaDZSa0Tz9rk5YJJezIZbPagZfp8jjjlfnICnmSCSnIJiFMeoCS0nWp8eT0Og9woaYYxmi9FWUGURi7Srg7EWsTWrh/F0uFrlAX0QeogQA=",
  "amr": [
    "pwd",
    "rsa",
    "mfa"
  ],
  "appid": "04b07795-8ddb-461a-bbee-02f9e1bf7b46",
  "appidacr": "0",
  "deviceid": "31a1bcda-7581-433f-b127-d02f2dd91ce3",
  "family_name": "Lindberg",
  "given_name": "Mattias",
  "ipaddr": "90.225.164.34",
  "name": "Mattias Lindberg",
  "oid": "ffa9a7f9-5dbc-430a-a871-ee0e628ae9a3",
  "rh": "                                                  ",
  "roles": [
    "Contributor"
  ],
  "scp": "Developer.Debug",
  "sub": "                                                  ",
  "tid": "                                    ",
  "unique_name": "mattias@lepton.se",
  "upn": "mattias@lepton.se",
  "uti": "dvuVa2uX0kaBqN4SqT5YAA",
  "ver": "1.0"
}
```


Application ID for Azure CLI

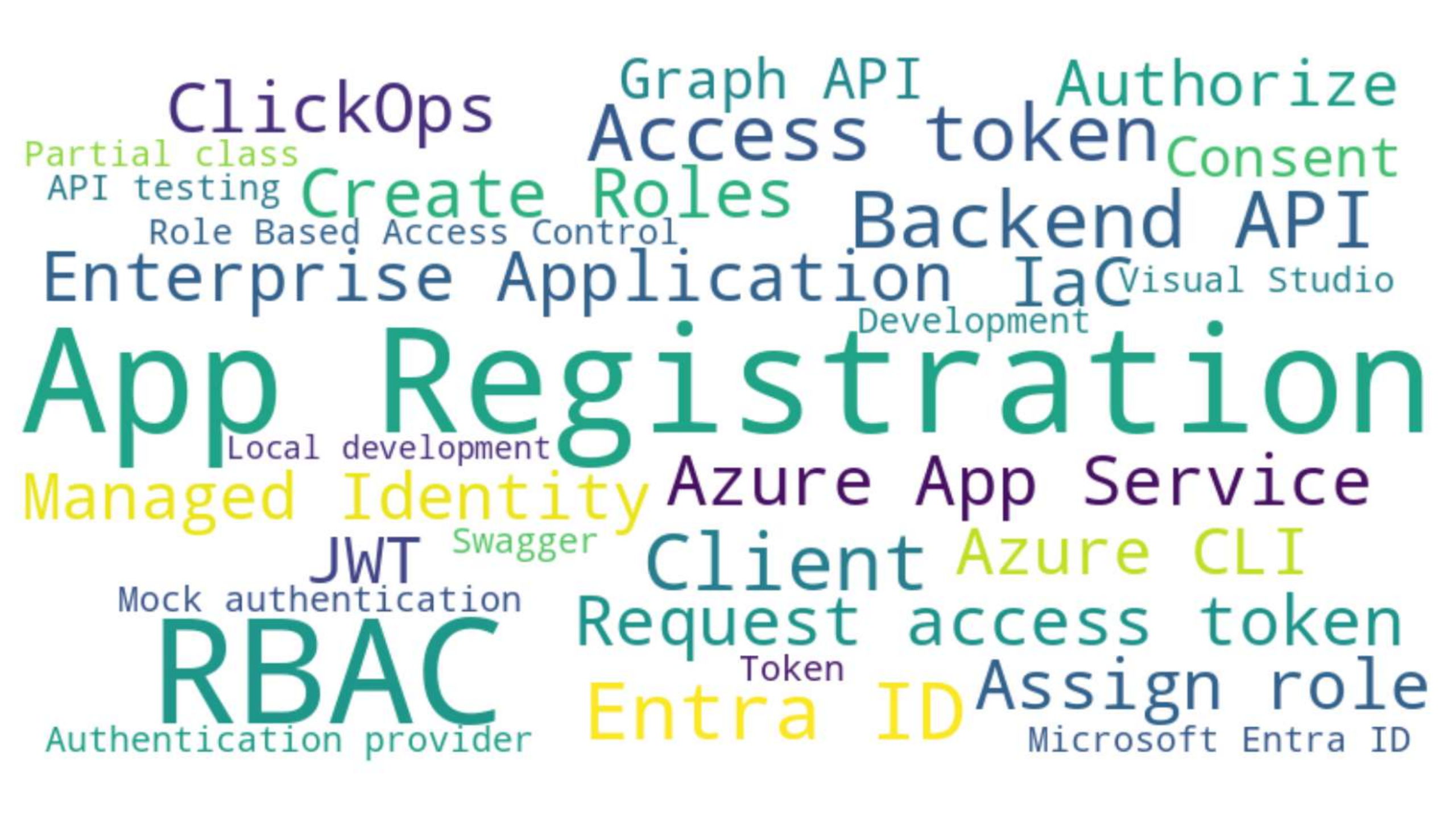
Scope that we consented

Mock authentication in Backend API

- During development you may want to run your application independent of and Azure resources, including the App Registration
  - When waiting for the IT Department
  - API testing using Postman

- Solution: Create your own authentication handler
  - Return any claims you want to use
  - https://github.com/MattiasLindberg/NIS2024/blob/main/BackendAPI/MockAuthenticationHandler.cs

Wrapping up


ClickOps
Graph API
Authorize
Access token
Partial class
Consent
API testing
Create Roles
Backend API
Role Based Access Control
Enterprise Application
IaC
Visual Studio
Development
App Registration
Local development
Azure App Service
Managed Identity
Swagger
JWT
Client
Azure CLI
Mock authentication
Request access token
RBAC
Token
Assign role
Entra ID
Authentication provider
Microsoft Entra ID

Try this out at home

- https://github.com/MattiasLindberg/NIS2024
  - Source code for the demo projects
  - PDF version of the presentation

NORDIC INTEGRATION SUMMIT

24

Session Feedback