How Do You Get an Access Token in Azure?

The .NET Developer's Guide to Token Credentials

 nestenius.se

Why Did I Specialize In This Topic?

AZ-204 Azure
Developer Associate

How Did I Approach This?

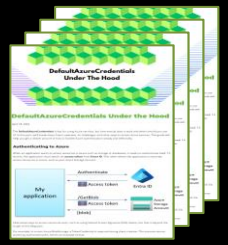
My Philosophy for Learning

You Don’t Really Understand It Until You Can Explain It!

My Study Process

Cloud Debugger

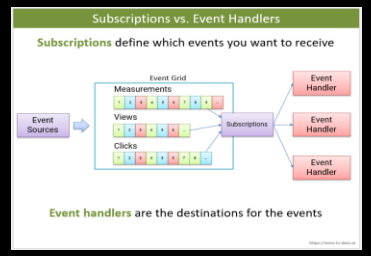
- Explore and learn Azure
- Open-source



Blog Posts

CONTENT

Presentations



Azure for Developers Training Workshop

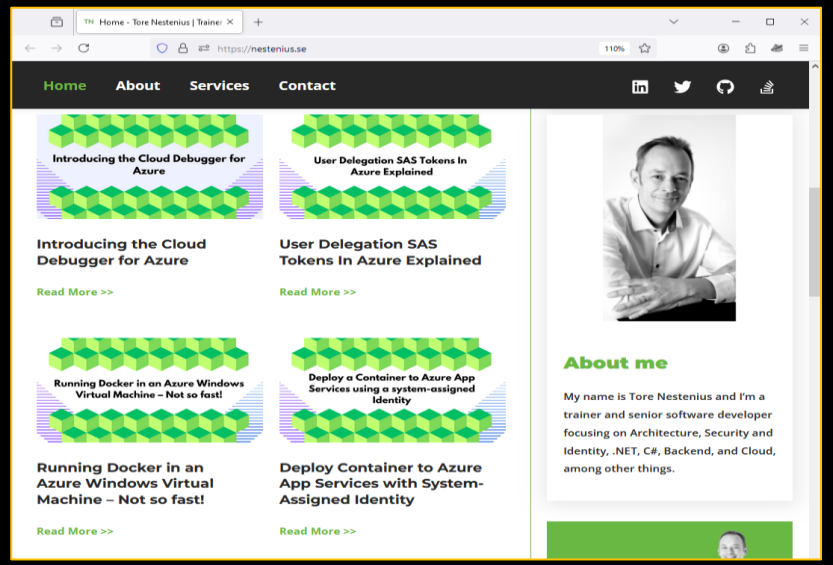
Did I Make it?

Passing the Certificate

Who Am I?

About Tore Nestenius

Work: tn-data.se

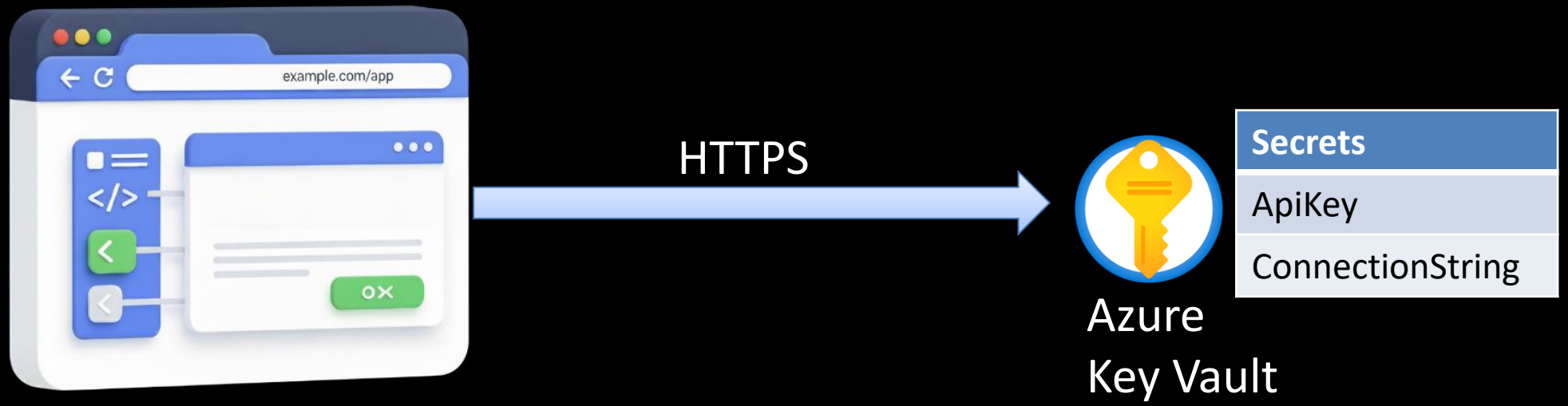


Blog: nestenius.se

meetup.com/net-skane

Azure Authentication Fundamentals

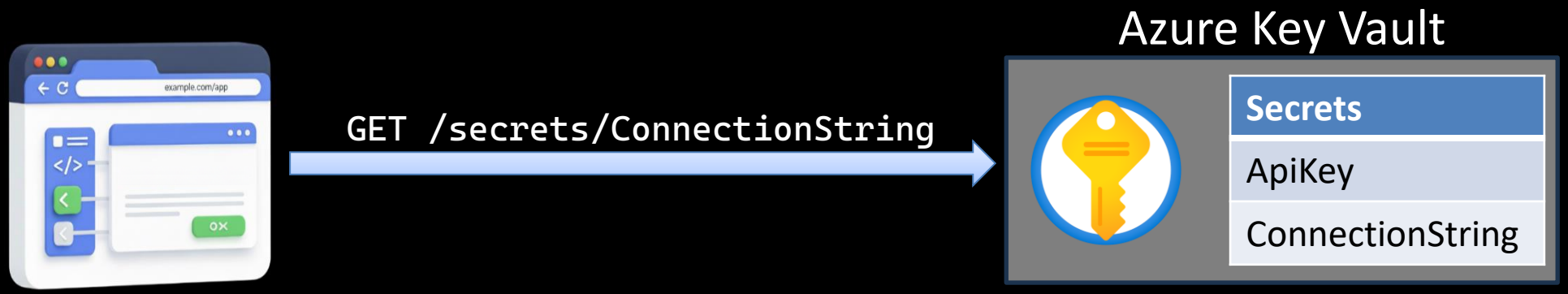
Our App Needs Secrets From Azure Key Vault



Let’s Try The Most Obvious Approach First

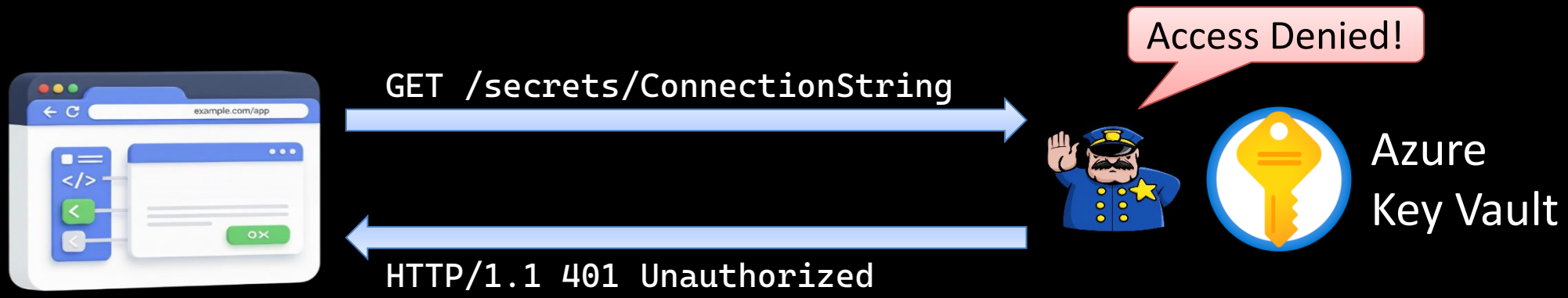
Live Coding #1

```
var url = "https://my-demo-keyvault-633.vault.azure.net";

var secret = await $"{url}/secrets/ApiKey"
                    .SetQueryParam("api-version", "7.4")
                    .GetJsonAsync<Secret>();

Console.WriteLine($"Secret value: {secret.value}");

public record Secret(string value, string id, object attributes);
```



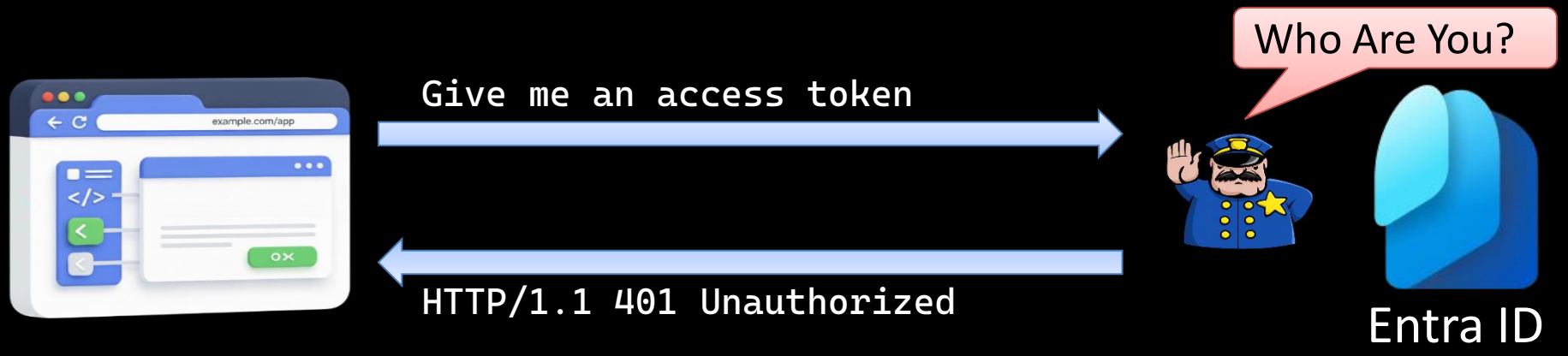
Authentication In Azure

We Need An Access Token To Access Key Vault



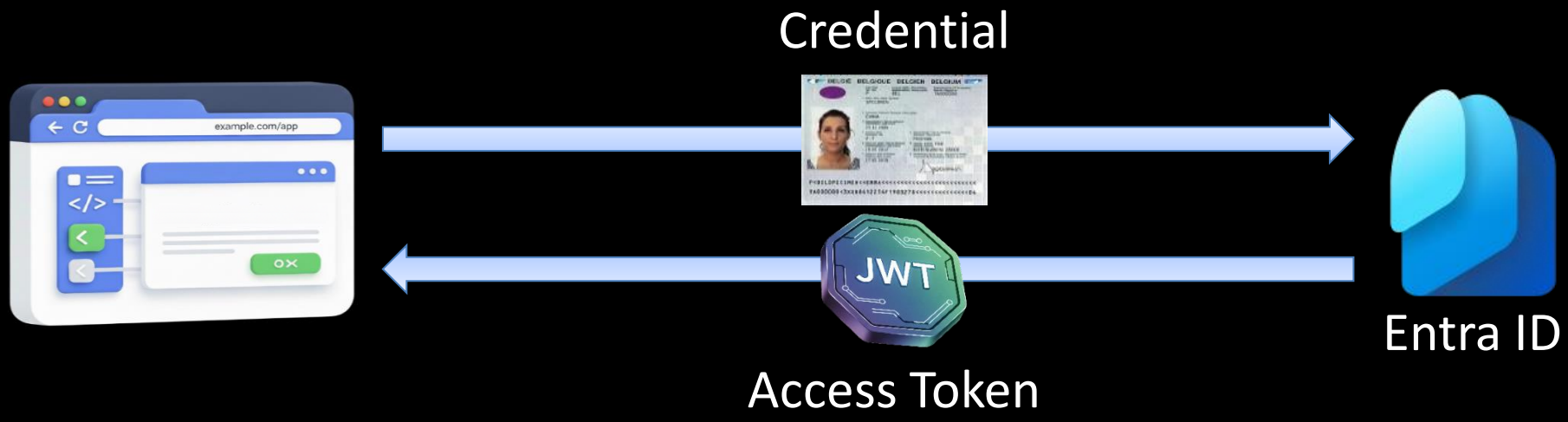
How Do I Get An Access Token?

The App Must Authenticate With Entra ID First



How Does My App Authenticate?

The App Provides A Valid Credential

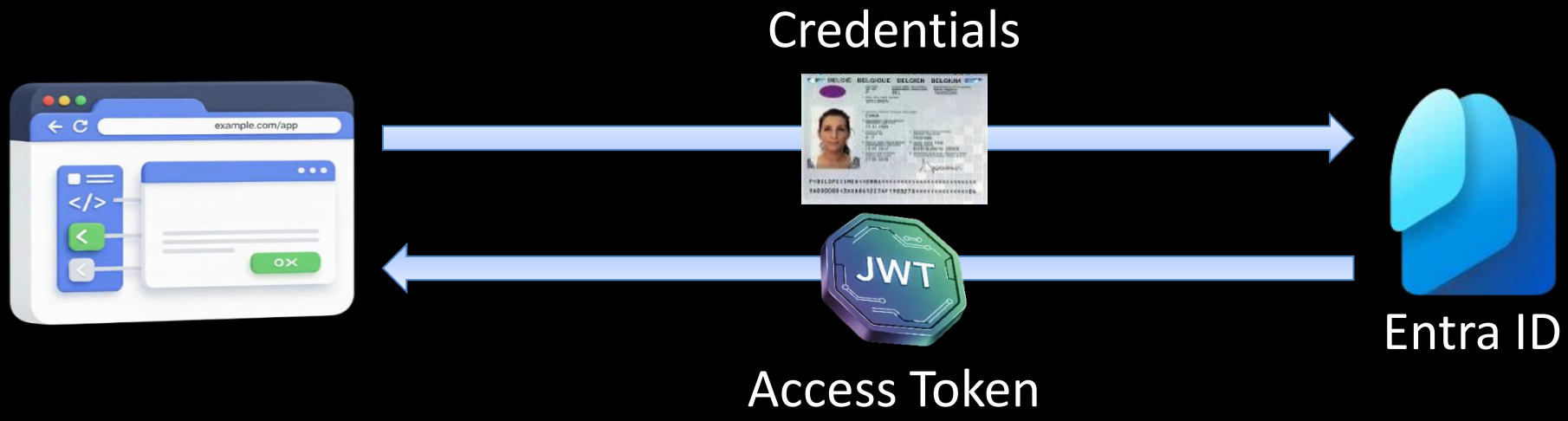


Let's See This In Practice

Live Coding #2

```
// Get Access Token
var options = new AzureCliCredentialOptions
{ TenantId = "567d82a1-7f61-4da2-b955-d3244ea6e976" };

var credential = new AzureCliCredential(options);
var context = new TokenRequestContext(["https://vault.azure.net/.default"]);

var accessToken = await credential.GetTokenAsync(context, default);

Console.WriteLine($"Token acquired:\r\n{accessToken.Token}\r\n\r\n");
```

What Is Inside The Access Token?



Exploring the Access Token

```
var accessToken = await credential.GetTokenAsync(tokenContext);
```

Text Visualizer

Expression: token.Token

String manipulation: JWT Decode

Header:


```
{
  "typ": "JWT",
  "alg": "RS256",
  "x5t": "yEUwmXWL107Cc-7QZ2WSbeOb3sQ",
```


Payload:


```
{
  "aud": "https://management.azure.com",
  "iss": "https://sts.windows.net/567d82a1-7f61-4da2-b955-d3244ea6e976/",
  "iat": 1761760935,
  "nbf": 1761760935,
  "exp": 1761766174,
  "acr": "1",
  "acrs": [
    "p1"
  ],
  "aio":
"AZQAa/8aAAAAHLbEmTOwM9BbHPyTTwjJszfuHYnjRJ/63cvXE8S72SSeM0cdS2lOg4b8HuXsgD
```


Signature:

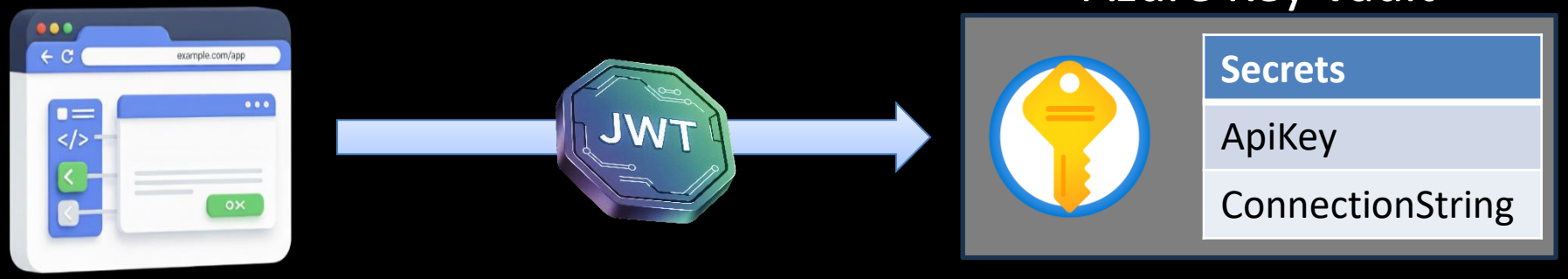

```
ShMkq5fwaIIwUlxbvMEp-7VcO2_9qe0RNS5qRIY0QawHz9RRuYA5AXRr3NESQfPG0YhL8KN8F4b
SEb83Qm31fRbwHD3vmfToDsO0QmuksYr8MJbq2_UHjOdsNEyTstXiLSkgtsOQ4kZxesnXqTZkfT
GKzBLe7A36t2XnTu8wAiAXXLf1yTy0AMQgGMfwYHrVRF9JxCh2H3BtK1wjlEPtRhtpUYS2Xo1sr
xW-BEcpWhsdyqkVuMDppj0kNJQsClY2IgKuWrmWFo_HSdQP2y3b12YoHFOoUsU0xwoTp1-
```


☑ Word Wrap

Using The Access Token

Live Coding #4

```
var secret1 = await $"{url}/secrets/ApiKey"
                    .SetQueryParam("api-version", "7.4")
                    .WithOAuthBearerToken(accessToken.Token)
                    .GetJsonAsync<Secret>();

var secret2 = await $"{url}/secrets/ConnectionString"
                    .SetQueryParam("api-version", "7.4")
                    .WithOAuthBearerToken(accessToken.Token)
                    .GetJsonAsync<Secret>();

Console.WriteLine($"ApiKey:           {secret1.value}");
Console.WriteLine($"ConnectionString: {secret2.value}");
```



What Is A Token Credential?

It Knows How To Get An Access Token

```
var credential = new AzureCliCredential();
```

What Credentials Exist?

The Many Types Of Credentials

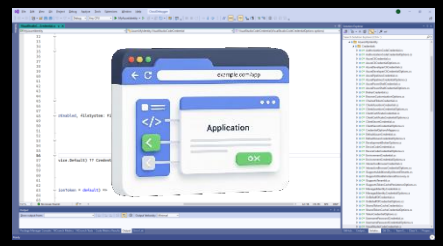
AuthorizationCodeCredential
AzureCliCredential
AzureDeveloperCliCredential
AzurePipelinesCredential
AzurePowerShellCredential
BrokerCredential
ChainedTokenCredential
ClientAssertionCredential
ClientCertificateCredential
ClientSecretCredential
DefaultAzureCredential

DeviceCodeCredential
EnvironmentCredential
InteractiveBrowserCredential
ManagedIdentityCredential
OnBehalfOfCredential
SharedTokenCacheCredential
UsernamePasswordCredential
VisualStudioCodeCredential
VisualStudioCredential
WorkloadIdentityCredential
...

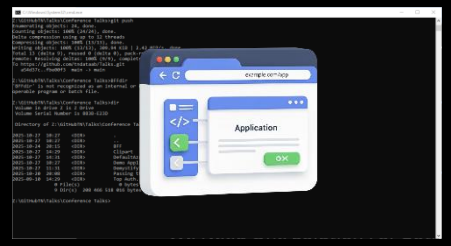
Why Do We Need So Many?

We Need To Support Many Environments

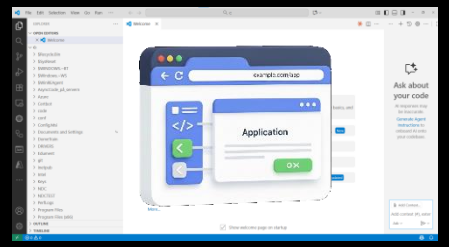
Outside Azure



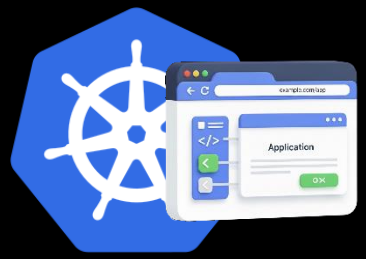
Visual Studio



Command line

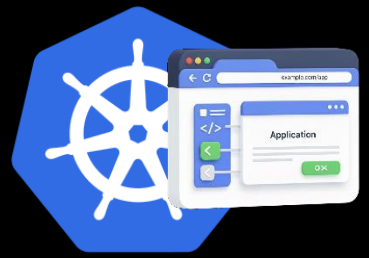


Visual Studio Code



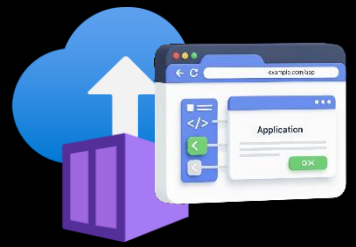
Kubernetes

Inside Azure

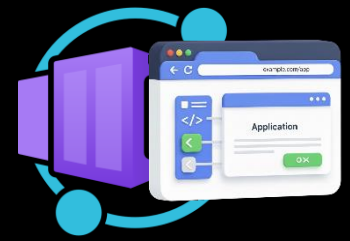


Kubernetes

App Services



Container Instances

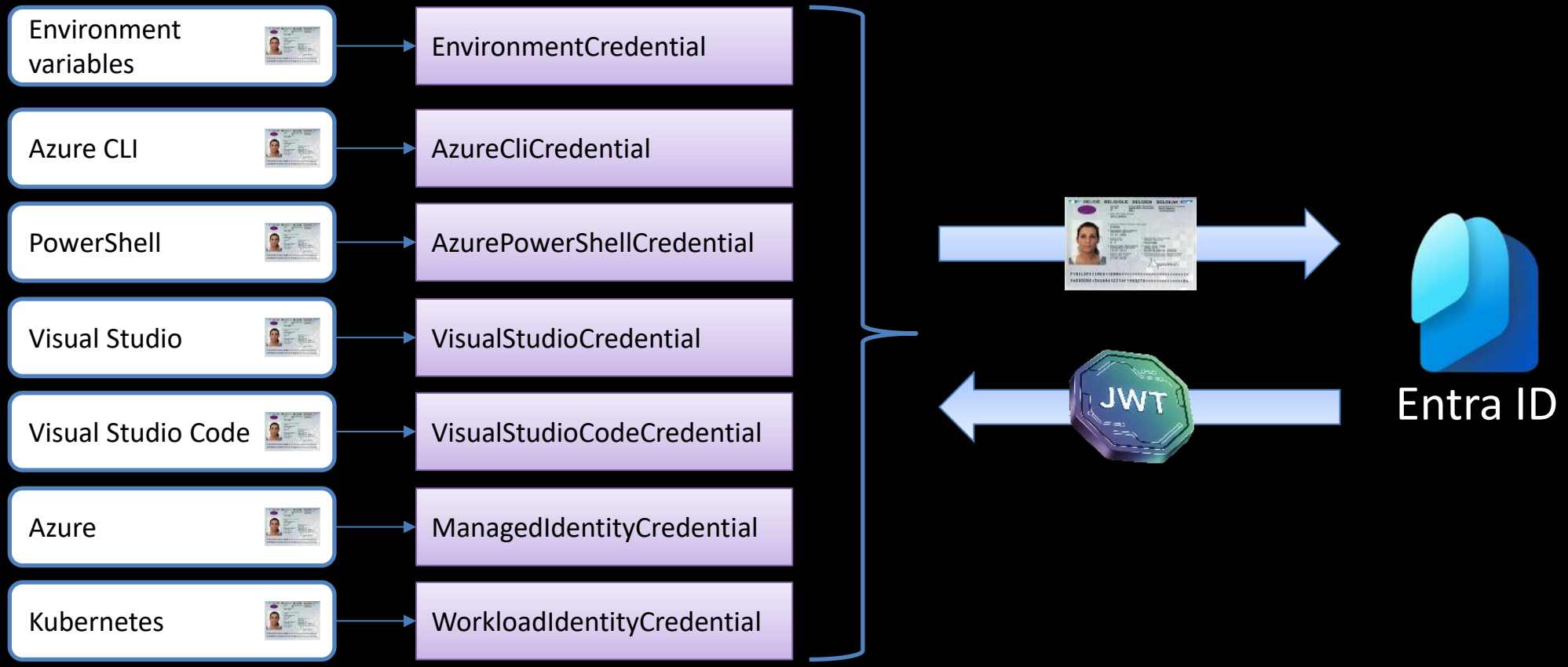


Container Apps

We Need One Solution For All!

Credential Sources

Where Do The Credentials Come From?



What Does This Mean For Our Apps?

Credential Sources

Different Environments Need Different Credentials

Joe's Machine

VisualStudioCredential

Build Server

EnvironmentCredential

Anna's Machine

AzureCliCredential

Azure

ManagedIdentityCredential

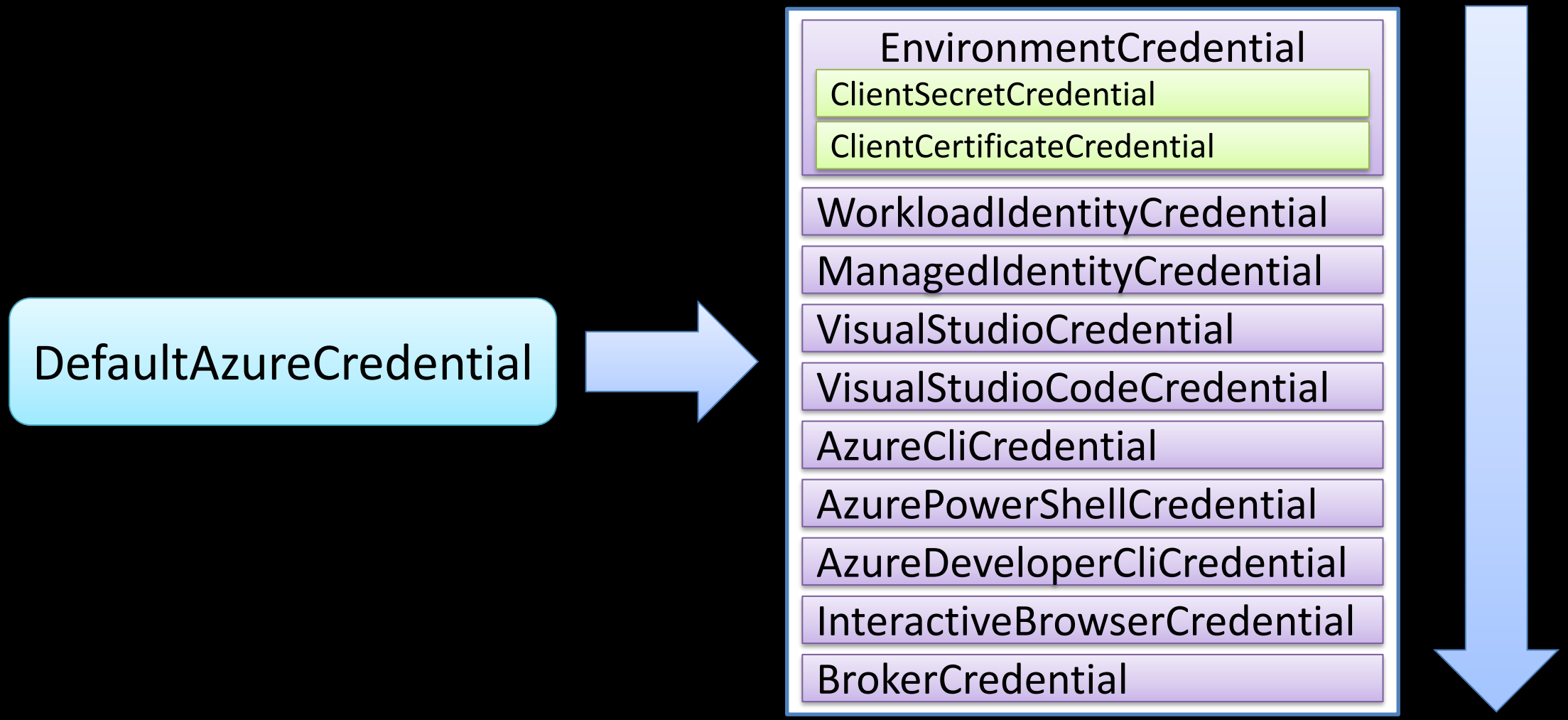
What Can This Look Like In Code?

Live Coding #5

```
var environment = "Anna";

TokenCredential? credential = null;

switch (environment)
{
    case "Joe":
        credential = new VisualStudioCredential();
        break;
    case "Anna":
        credential = new AzureCliCredential();
        break;
    case "BuildServer":
        credential = new EnvironmentCredential();
        break;
    case "Azure":
        credential = new ManagedIdentityCredential();
        break;
}
//Use credential to get token
```

DefaultAzureCredential

DefaultAzureCredential To The Rescue



Live Coding #6

Joe's Machine

VisualStudioCredential

Anna's Machine

AzureCliCredential

Build Server

EnvironmentCredential

Azure

ManagedIdentity Credential

```
var options = new DefaultAzureCredentialOptions
{
    TenantId = "567d82a1-7f61-4da2-b955-d3244ea6e976"
};

var credential = new DefaultAzureCredential(options);

//Use credential to get token
```

Specifying TenantId is a best practice!

DefaultAzureCredential Trouble!

You Might Get The Wrong Credential

DefaultAzureCredential

- EnvironmentCredential
- WorkloadIdentityCredential
- ManagedIdentityCredential
- VisualStudioCredential
- VisualStudioCodeCredential
- AzureCliCredential
- AzurePowerShellCredential
- AzureDeveloperCliCredential
- InteractiveBrowserCredential
- BrokerCredential

You expected credentials here

Read/Write

But you received this one!

Read Only

Can be hard to troubleshoot

What Else Is A Problem?

How Can We Find Out Which Credential Was Chosen?

DefaultAzureCredential

- EnvironmentCredential
- WorkloadIdentityCredential
- ManagedIdentityCredential
- VisualStudioCredential
- VisualStudioCodeCredential
- AzureCliCredential
- **AzurePowerShellCredential**
- AzureDeveloperCliCredential
- InteractiveBrowserCredential
- BrokerCredential

What Else Is A Problem?

Exploring Timeouts When No Credentials Are Found

```
// Benchmark DefaultAzureCredentials

var credential = new DefaultAzureCredential();

var context = new TokenRequestContext(["https://vault.azure.net/.default"]);

var accessToken = await credential.GetTokenAsync(context);

Console.WriteLine($"Token acquired:\r\n {accessToken.Token}\r\n\r\n");
```

One Solution: Disabling Unwanted Credentials

```
// Exclude certain credentials
var options = new DefaultAzureCredentialOptions
{
    ExcludeEnvironmentCredential = false,
    ExcludeManagedIdentityCredential = false,
    ExcludeVisualStudioCredential = true,
    ExcludeVisualStudioCodeCredential = true,
    ExcludeAzureCliCredential = false,
    ExcludeAzurePowerShellCredential = true,
    ExcludeAzureDeveloperCliCredential = false,
    ExcludeWorkloadIdentityCredential = true,
    ExcludeBrokerCredential = true,
    ExcludeInteractiveBrowserCredential = true
};

var credential = new DefaultAzureCredential(options);
```

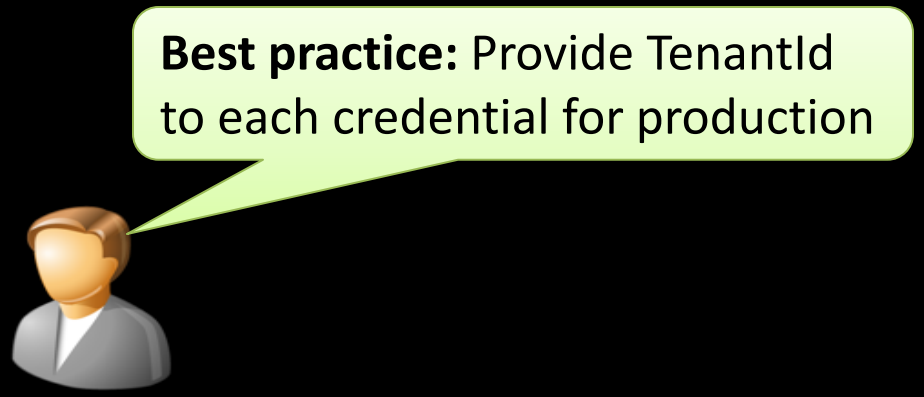
ChainedTokenCredential

Custom Control When You Need It

```
var credential = new ChainedTokenCredential(
                     new EnvironmentCredential(),
                     new VisualStudioCredential(),
                     new AzureCliCredential(),
                     new ManagedIdentityCredential());
```

ChainedTokenCredential

- EnvironmentCredential
- VisualStudioCredential
- AzureCliCredential
- ManagedIdentityCredential



Who Uses These Token Credentials?

The Main Consumer Is The Azure SDK For .NET

Azure SDK Libraries

Azure.Security.KeyVault.Secrets by azure-sdk,
This is the Microsoft Azure Key Vault Secrets client library

Token Credential

Default Azure Credentials

Direct HTTP Requests to Azure APIs

```
// Using the Access Token and some refactoring
var vaultUrl = "https://my-demo-keyvault-633.vault.azure.net";

var secret1 = await $"{vaultUrl}/secrets/ApiKey"
                        .SetQueryParam("api-version", "7.4")
                       .WithOAuthBearerToken(accessToken.Token)
                       .GetJsonAsync<KeyVaultSecret>();
```

Used By Over 100+ Azure SDK Client Libraries!

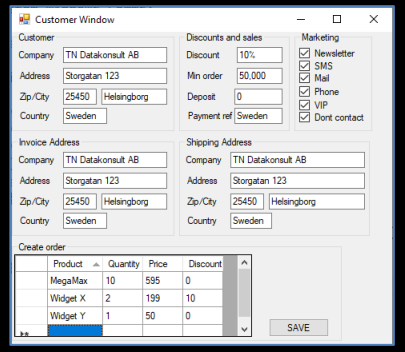
Using The Azure SDK for .NET

```
// Step 1: Create Key Vault client
var client = new SecretClient(vaultUri: new Uri(url),
                              credential: credential);

// Step 2: Get secrets from Key Vault
var response1 = await client.GetSecretAsync("ApiKey");
var response2 = await client.GetSecretAsync("ConnectionString");

// Step 3: Extract secrets from the response
KeyVaultSecret secret1 = response1.Value;
KeyVaultSecret secret2 = response2.Value;

Console.WriteLine($"ApiKey:           {secret1.Value}");
Console.WriteLine($"ConnectionString: {secret2.Value}");
```

Interactive Browser Credential

How Can Joe Access His Secrets?

Joe



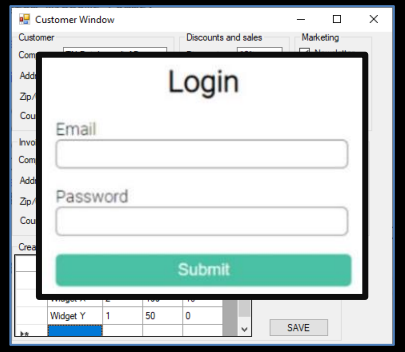
Desktop / Mobile

Get Joe's Secrets

How Should We Not Do It?

Don't Build Login Forms Into Your App

Joe



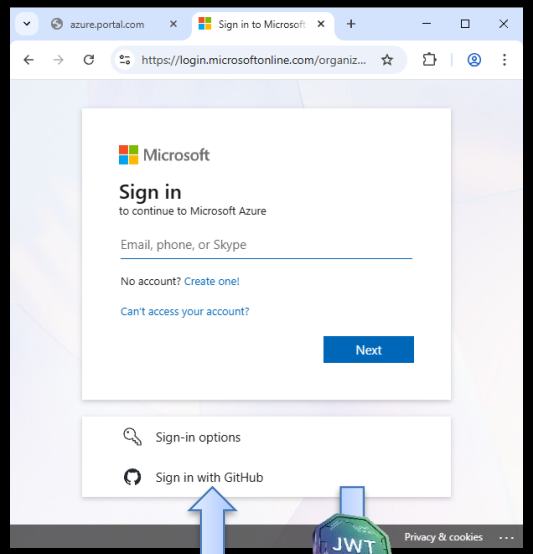
Desktop / Mobile

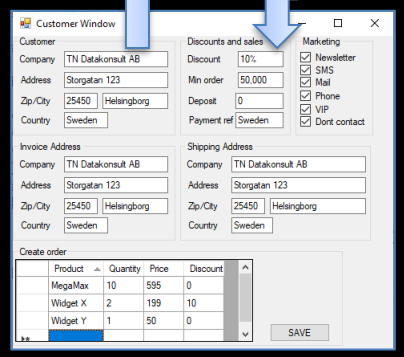
Get Joe's Secrets

What Is The Correct Way?

Interactive Browser Credential

Delegate Authentication To The Browser

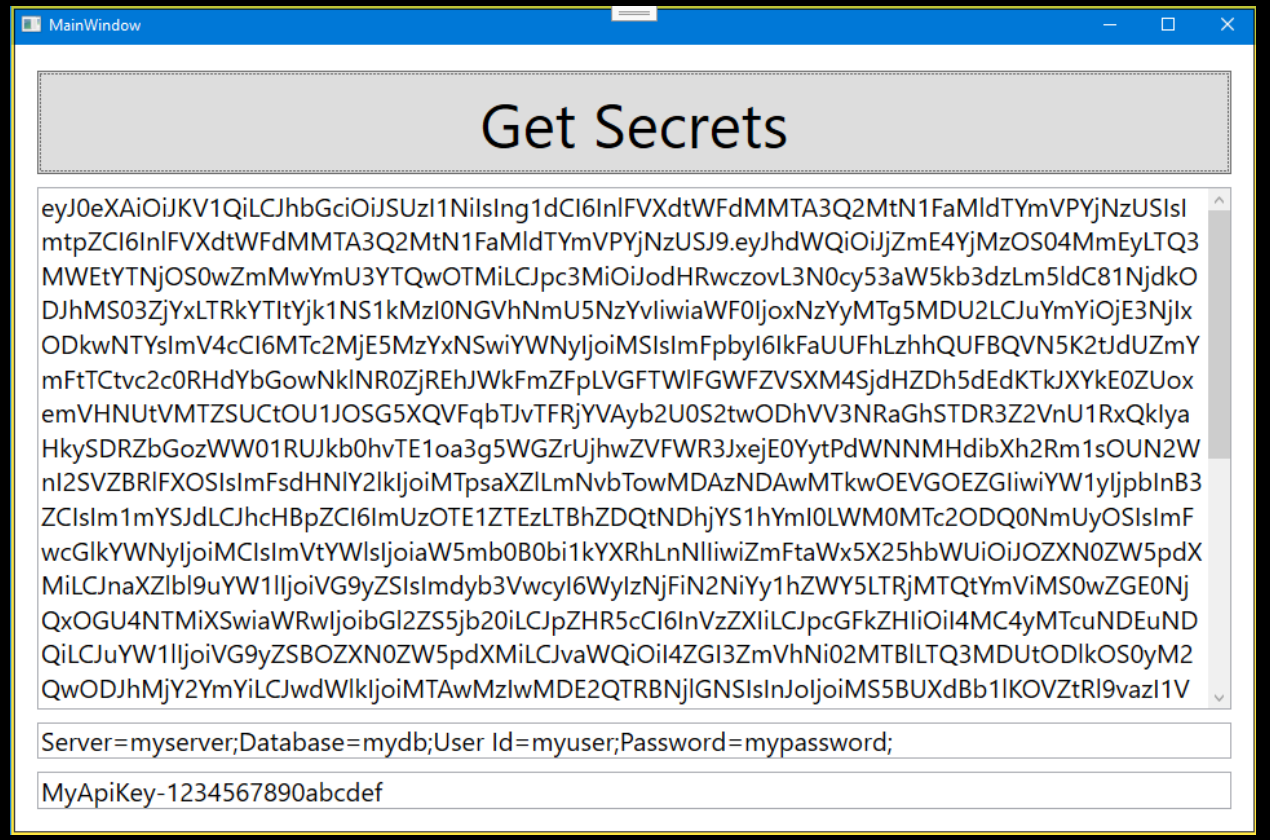




Desktop / Mobile

Entra ID

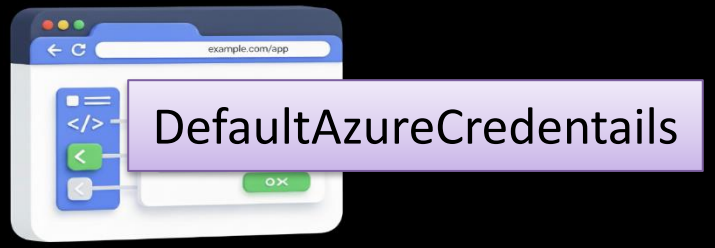
Exploring Interactive Browser Credential



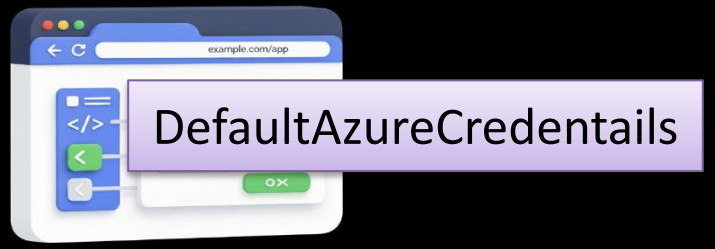
Local Development

Should We Use DefaultAzureCredentials For Local Dev?

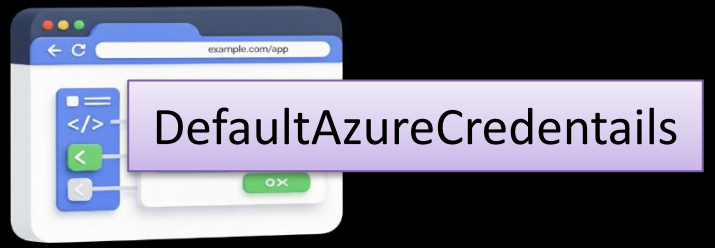
Developer
(Admin)



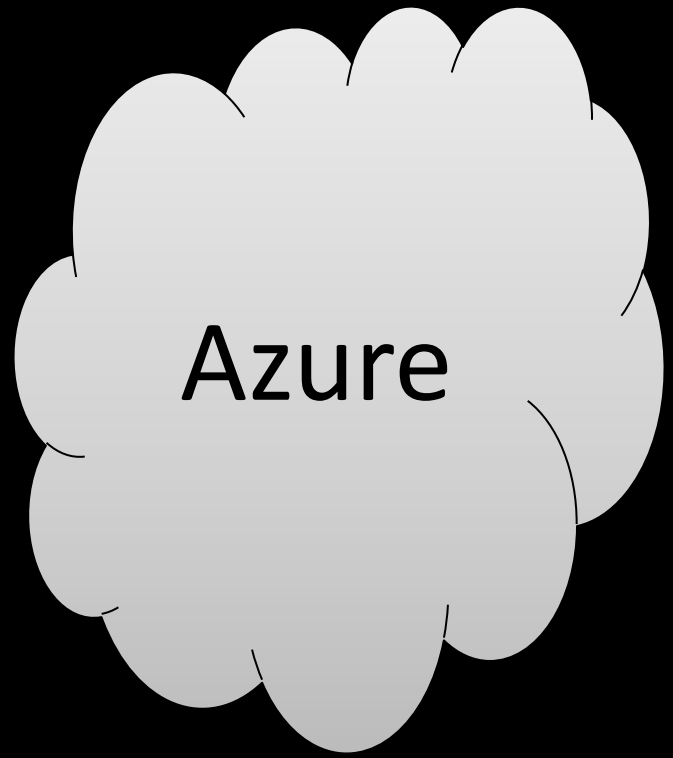
App #1



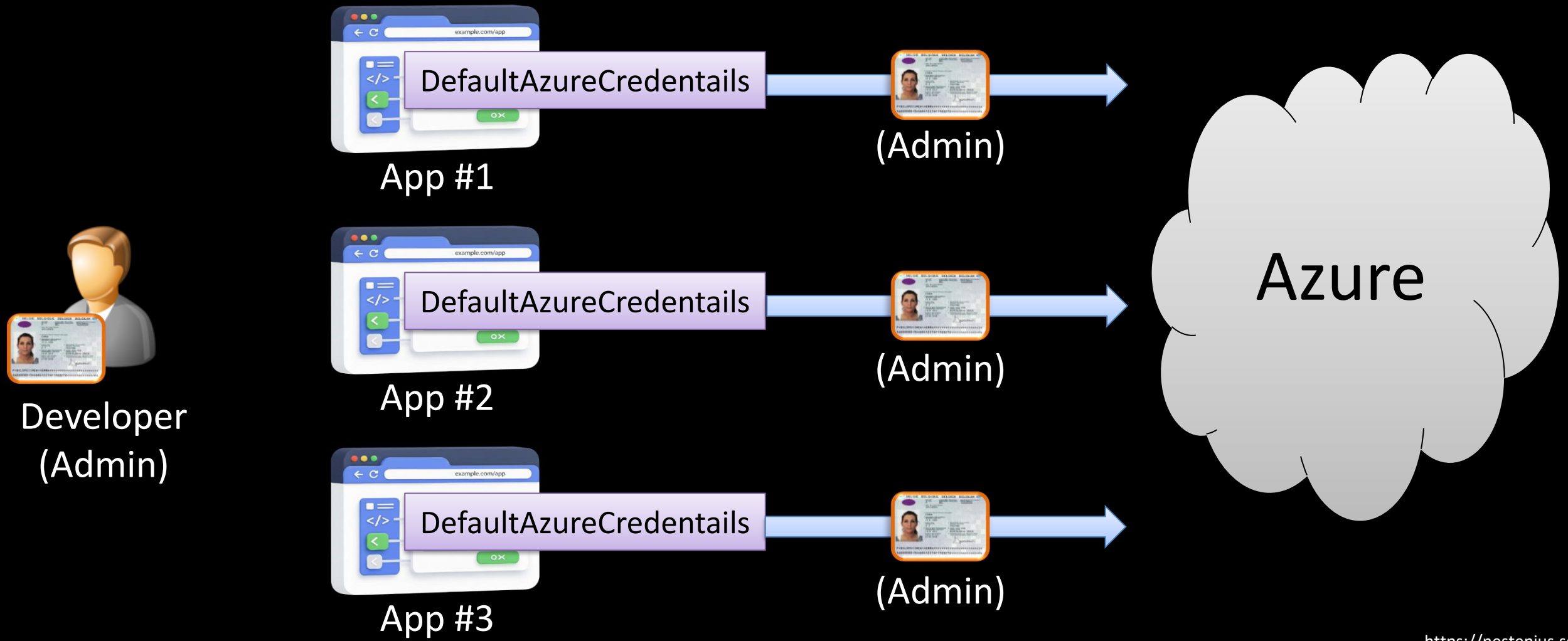
App #2



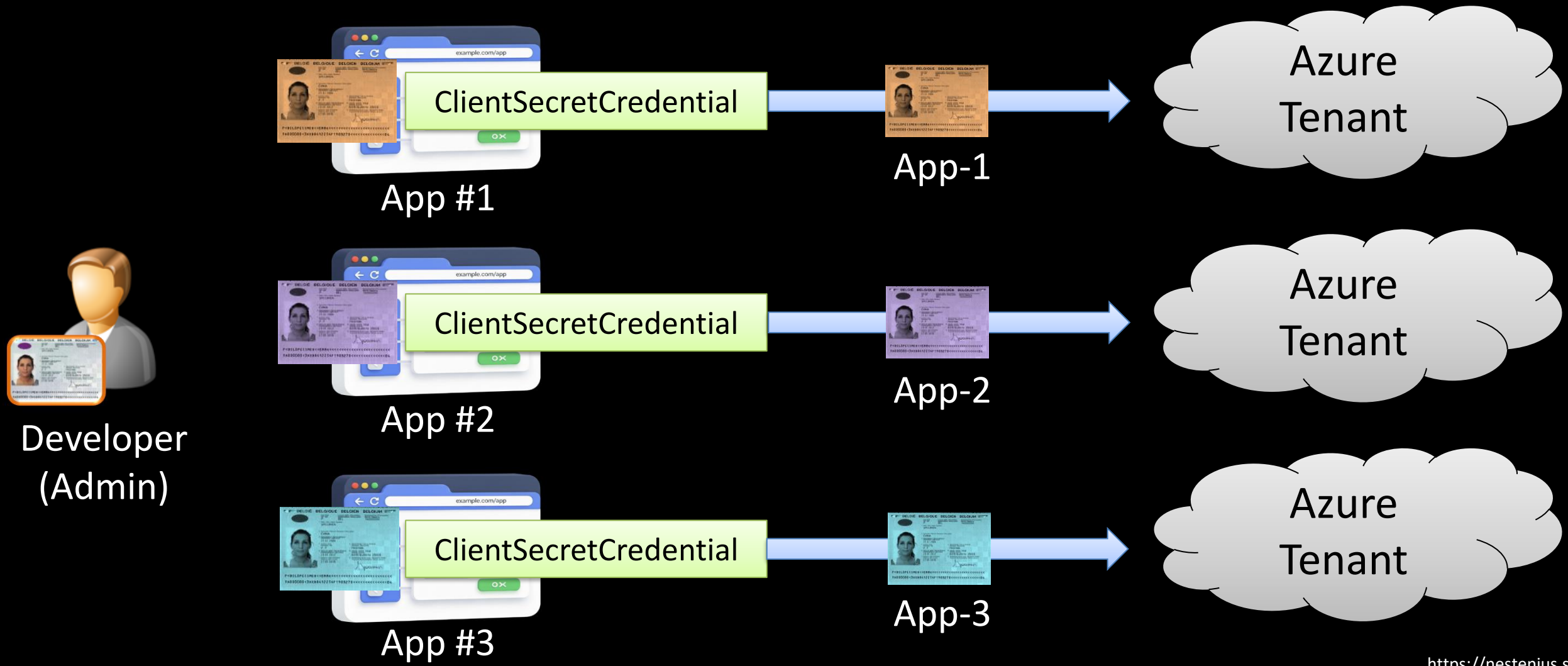
App #3



No, It Hides Authentication Issues in Production

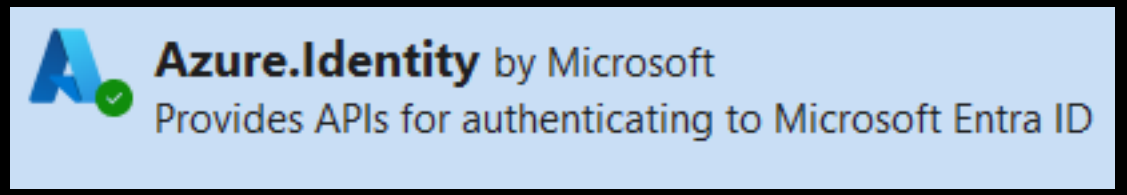


Better: Give Each App Its Own Identity



Frequently Asked Questions

How Can We Inspect The Internals In Even More Detail?



Custom Instrumentation and Logging

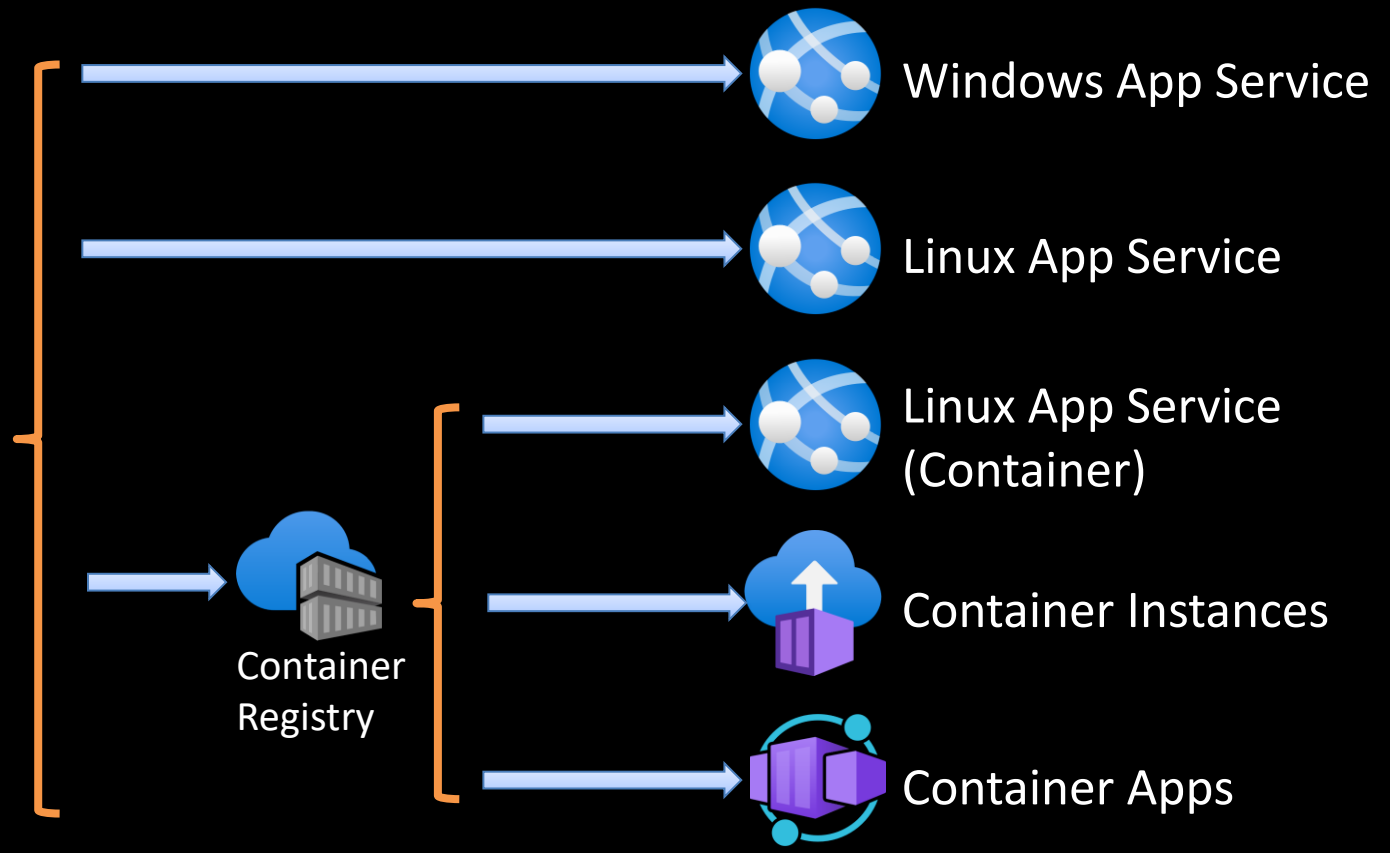
MyAzureIdentity

Warning: Only for learning purposes!

Exploring The Internals Of Token Credentials

Cloud Debugger

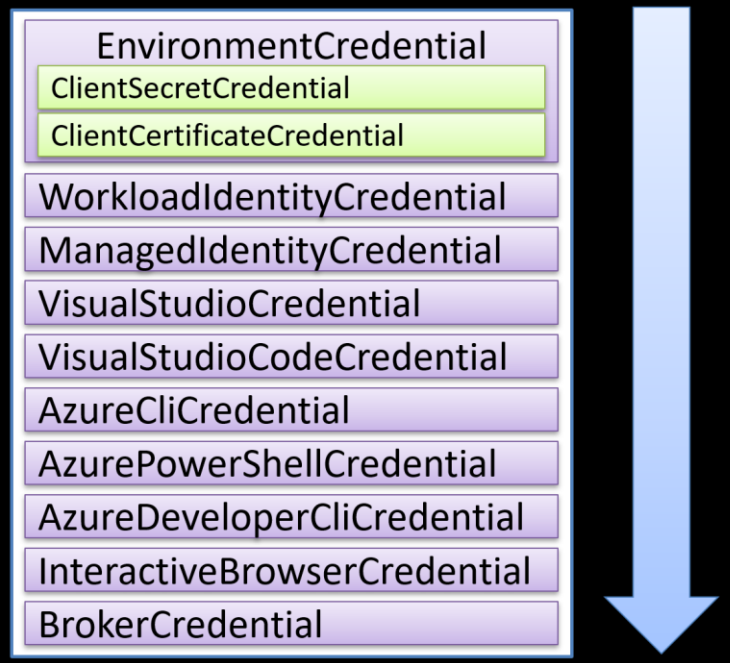
MyAzureIdentity



Advanced DefaultAzureCredential

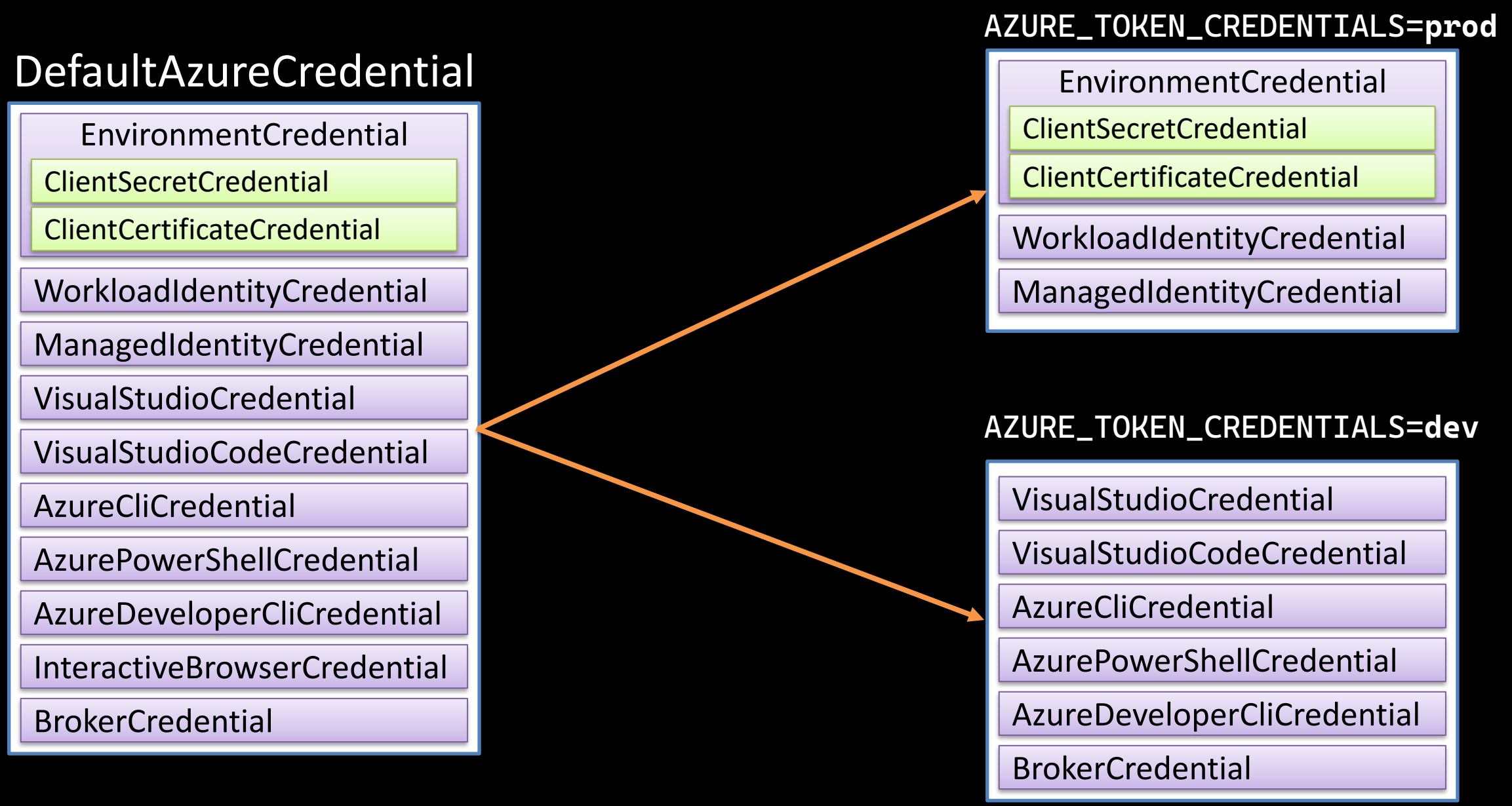
Advanced DefaultAzureCredential

By Default, It Will Try These Credentials



However, This Can Be Customized!

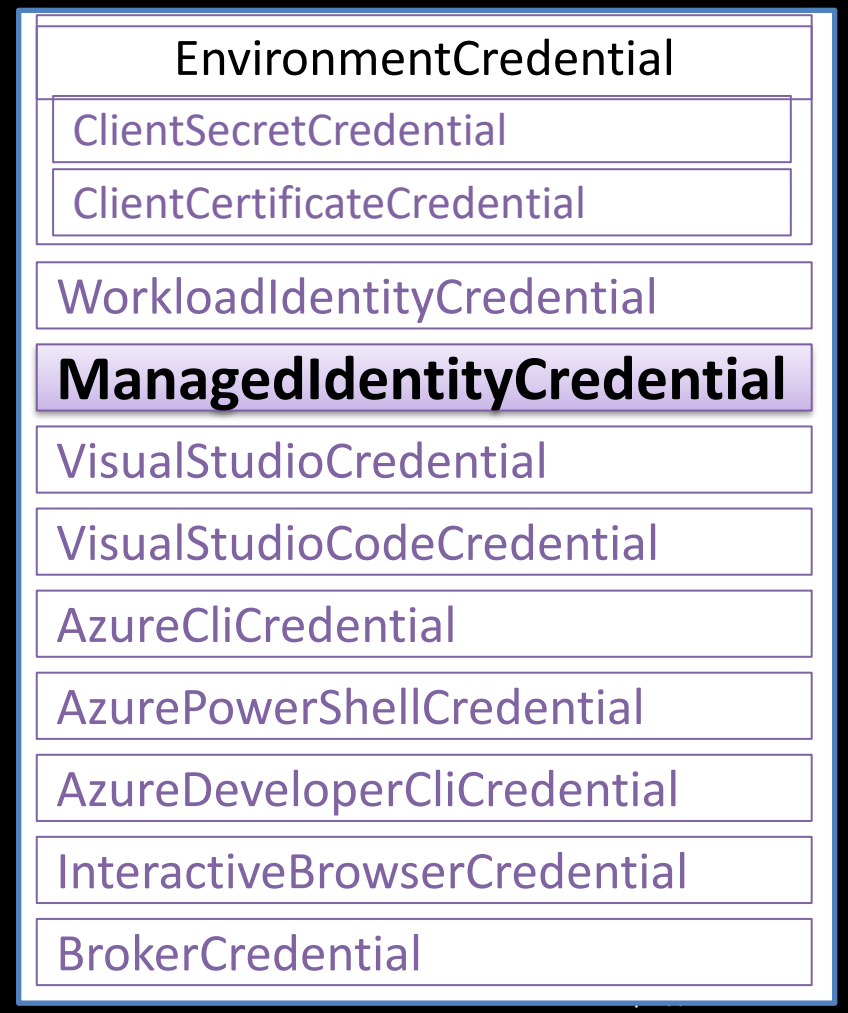
Advanced DefaultAzureCredential



A Specific Credential Can Also Be Selected

Environment Variable

```
AZURE_TOKEN_CREDENTIALS=ManagedIdentityCredential
```



QUESTIONS?

Cloud Debugger
https://github.com/tndata/CloudDebugger

Presentation
https://github.com/tndataab/PublicBlogContent

Blog: nestenius.se

Work: tn-data.se