Azure Authentication Fundamentals

Jan Vidar Elven | Martin Ehrnst

Evidi | Vipps

MVP Dagen

Jan Vidar Elven

Evidi AS

Tech Lead Cloud Platform & Security @ Evidi Solutions

MVP Security

@JanVidarElven

https://gotoguy.blog

Martin Ehrnst

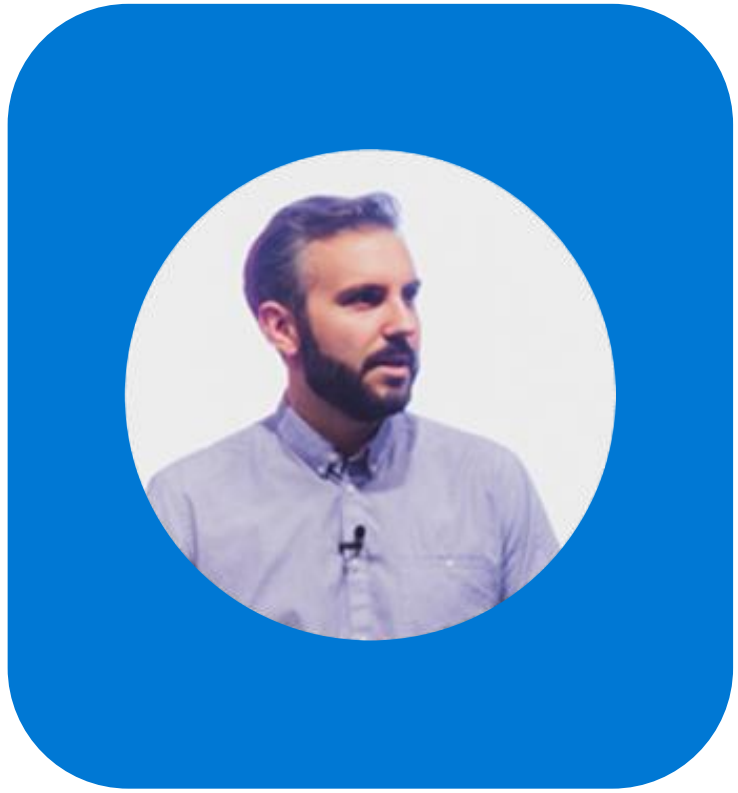
Vipps AS

Lead Platform Architect

MVP Azure

@ehrnst

https://adatum.no



Cloud Identity Platform

Authentication vs. Authorization

AuthN

- Verification of identity

AuthZ

- Granting permission to authenticated identity

Support industry standard protocols & open-source libraries

Microsoft Identity Platform
Azure AD | Microsoft Entra

Uses OIDC (Open ID Connect) for AuhtN

Uses OAuth 2.0 for AuthZ

Other main scenarios:

- Multi-Factor Authentication (MFA)
- Conditional Access
- SSO
- Governance
- Identity Protection

Zero Trust Identity

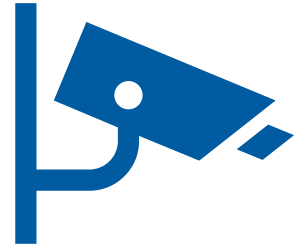
Main Principles of Zero Trust

Verify explicitly

- Azure AD Identity Protection
- Azure AD Authentication Methods
- Azure AD Conditional Access

Use least privileged access

- Azure AD PIM
- Microsoft Entra Identity Governance
- Azure AD Identity Secure Score



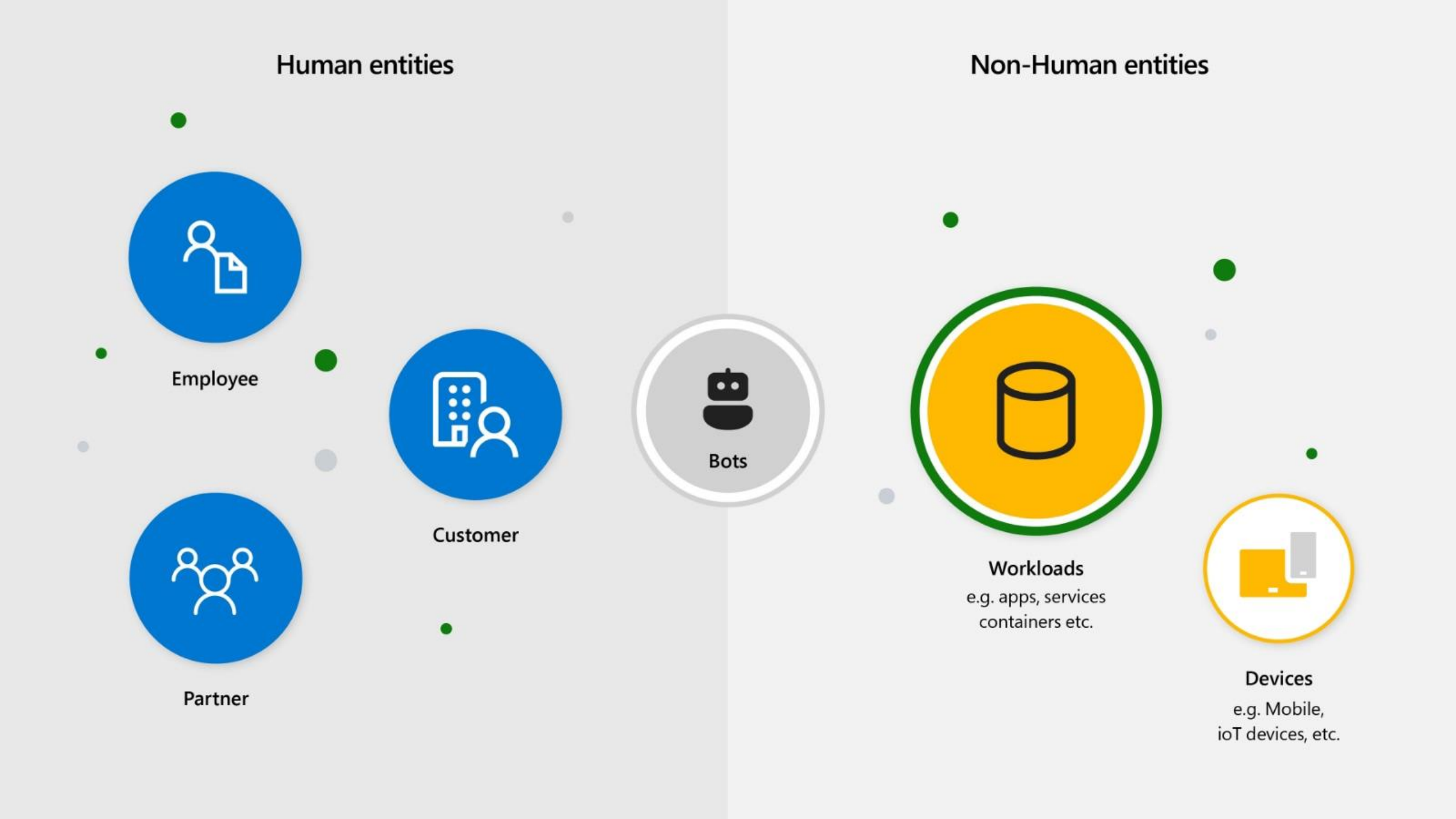
Assume breach

- Azure AD Risky Users
- Azure AD Risky Sign-ins
- Log Integration Azure Sentinel

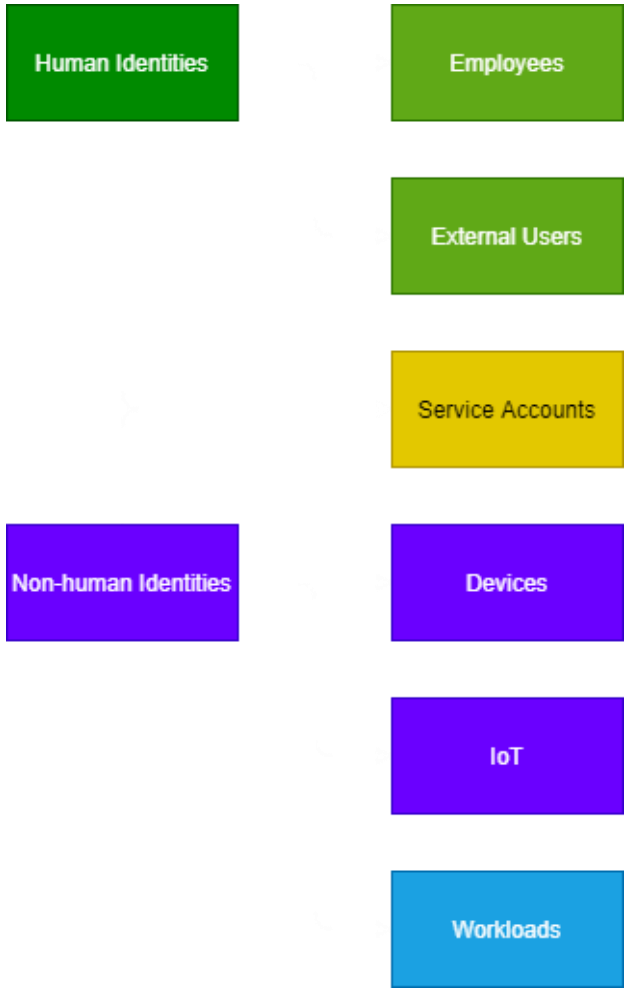
MVP-Dagen

Identity Authentication & Authorization

- Conditional Access
  - Conditions
  - Controls
- Privileged Access
  - PIM
  - PAM
- Roles
  - RBAC – Role Based Access Control
  - ABAC – Attribute Based Access Control
  - Custom Roles


Human entities
Employee
Customer
Partner
Bots
Non-Human entities
Workloads
e.g. apps, services
containers etc.
Devices
e.g. Mobile,
ioT devices, etc.

Human vs. Service / Workload Identities



Service Accounts

A User Account created with Privileges

Often Single-Factor Auth

Used for Legacy / Basic Auth

Service Principals

Application Identities and Managed Identities

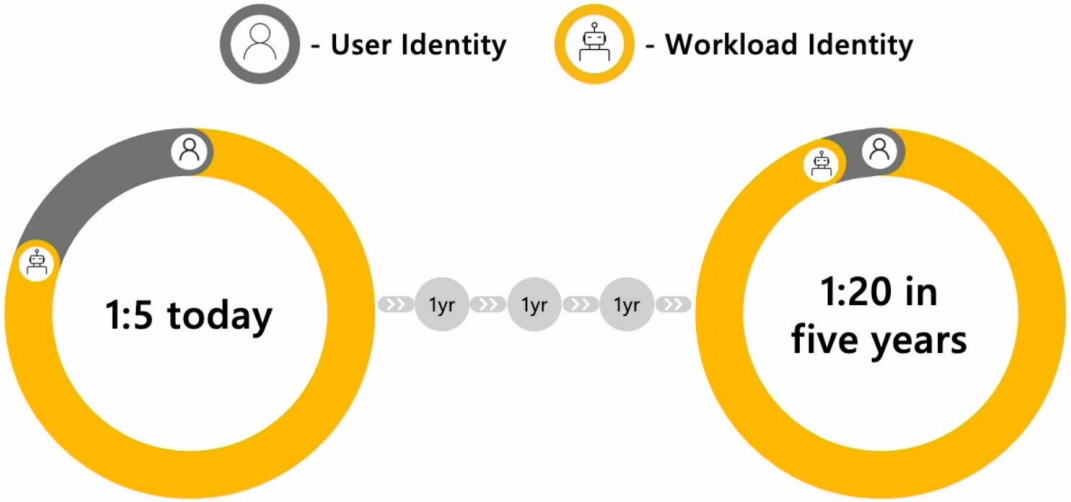
Modern Authentication for Services that support Azure AD

Represented as Enterprise Applications in Azure AD

Microsoft Entra Workload Identities (from Ignite)

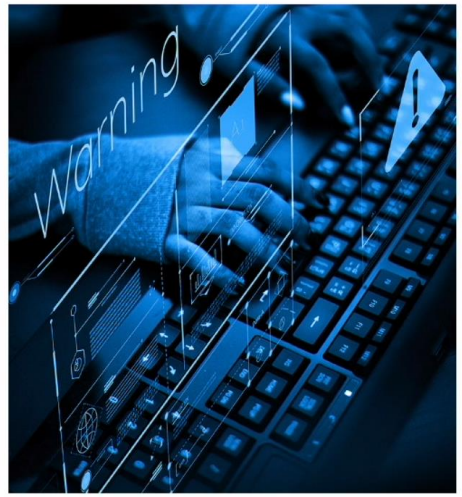
Workload identities are proliferating

Ratio of user identities vs. workload identities



Source: Microsoft Security internal research 2021

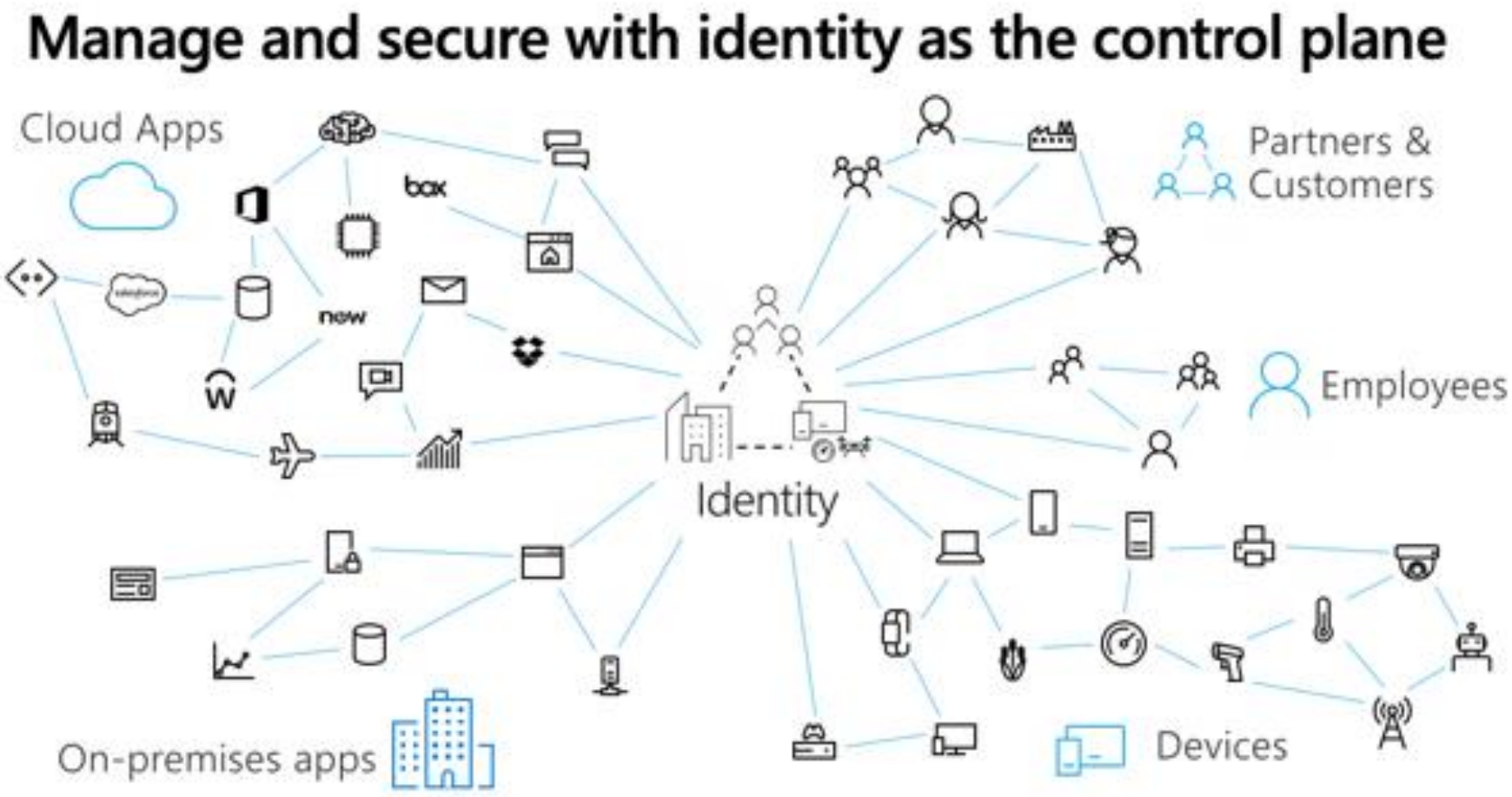
New emerging attack surface



Attackers have started targeting workload identities, mainly driven by **a lack of solutions and security capabilities** to protect workload identities.

Microsoft Entra Workload Identities (from Ignite)

Application Identities in Azure AD



OpenID Connect

OAUTH 2

SAML V2.0
Security Assertion Markup Language

WS-Fed

Scenarios for Application Identities in Azure AD

Develop own App/API

Resource Access (Service Principal)

SaaS application / Gallery

On-Premises Application / Application Proxy

Application publishing, ADC

App Registration

Establishes a trust with Microsoft as IDP

Acts as the application definition

Support multi-tenancy and Microsoft accounts

Only exist in its home tenant

Enterprise application

The application identity within a tenant (SPN)

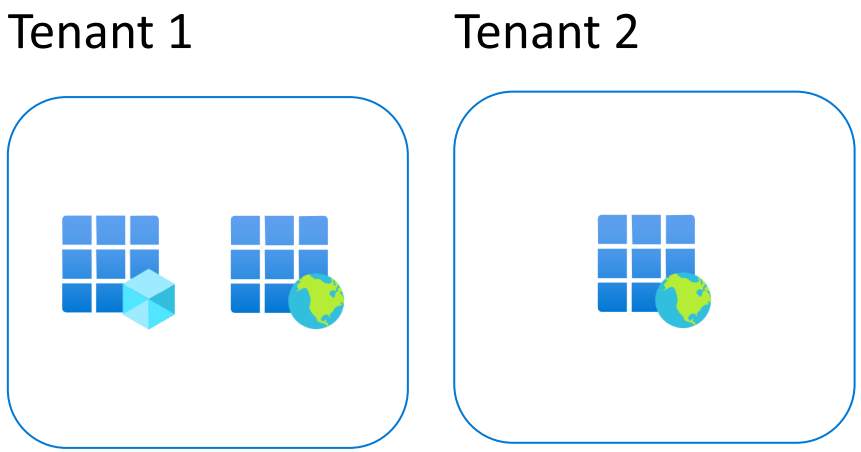
Can be assigned access to resources

Can have users assigned to it

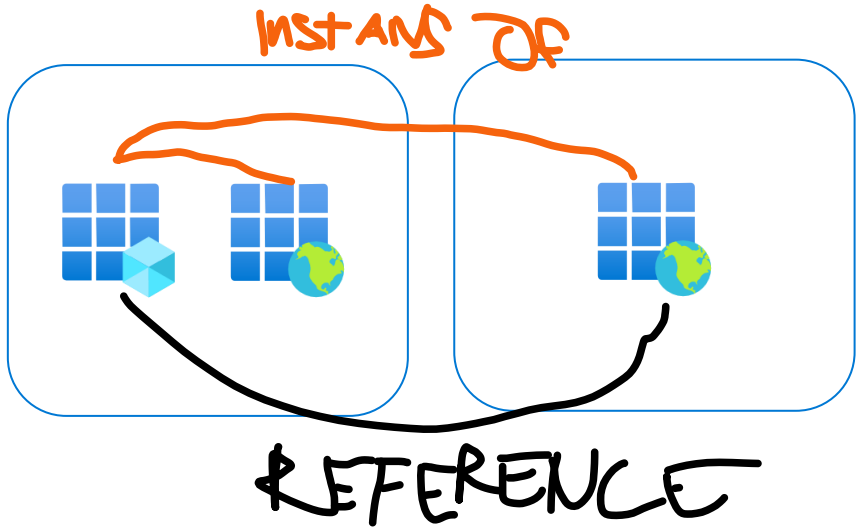
Relationship

Tenant 1

Relationship



Relationship



Application types

Single-page applications (SPA)

Web apps (server-side) and that access API's

Daemon & non-interactive apps

Native Apps/Public client

CLI

OAuth 2.0 grant flows

- Implicit flow
- Authorization Code flow
- Client credentials flow
- Resource Owner password flow
- Device Code flow

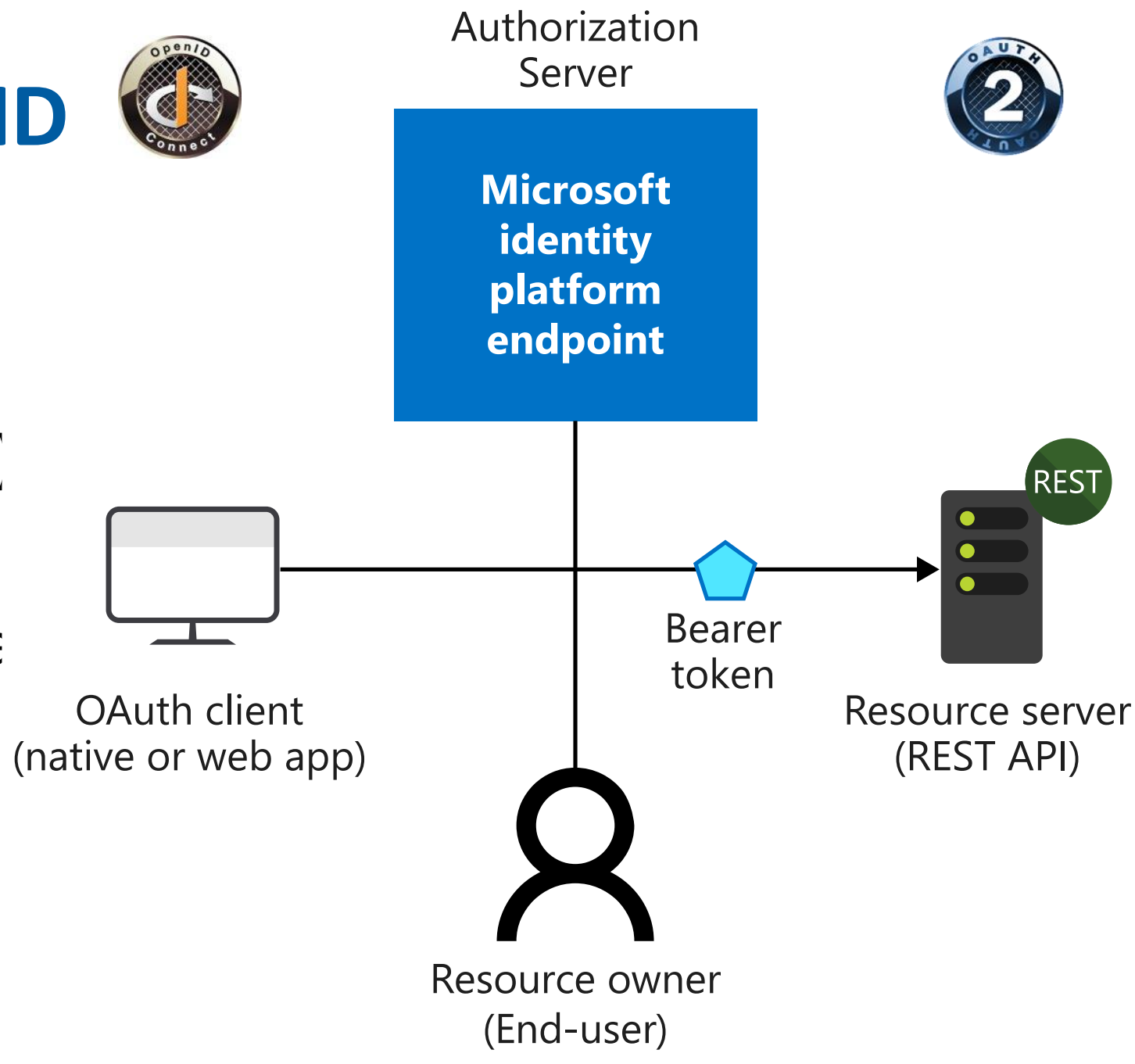
https://docs.microsoft.com/en-us/azure/active-directory/develop/msal-authentication-flows

OAuth2 and OpenID Connect..

AuthN & AuthZ

- OpenID Connect (OIDC) Authentication
- Oauth 2.0 for Authoriza

Roles in OAuth 2.0 ->



Tokens

JSON based

OpenID Connect

- OAuth 2.0
- ID Token
- Verify Identity
- JSON web tokens (JWT)
- Don’t confuse ID Token with Access Token!

Access Token

- For authorizing (AuthZ) and access to resource API
- Access Token audience via App Registration in Azure AD

Tokens how it looks


```
{
  "aud": "50e116c5-a13d-4c84-95e3-bd7d5ce5d786",
  "iss": "https://login.microsoftonline.com/a/v2.0",
  "iat": 1666003262,
  "nbf": 1666003262,
  "exp": 1666008420,
  "aio":
"AZQAa/8TAAAAe6vS6t0udp3SKfdkTxfiBOSZxvF1vgoOOwRMpdRd0M
2T/Fj6WDlPTYCjuqgTGDIwZaBJQzaDoh+eDYNmveAbgEOrfQOZhU43X
Vv3e2jOC8jfRIY9Geiq+oMuVHg4wP70NpqFLFqy/1obqXkeRL/VdKJY
yacjLvl/K+1dXbMOeyOxNbRjLfjPTVPvfC4/XST7",
  "azp": "1950a258-227b-4e31-a9cf-717495945fc2",
  "azpacr": "0",
  "name": "Martin Ehrnst",
  "oid": "28c95564-f342-4174-aae9-49934603f109",
  "preferred_username": "martin.ehrnst@vipps.no",
  "rh":
"0.ASAAXcJbgGSO1k6NJDiDyQaMWqlxHjL6d5dCl75hOfCWtSkgAH8.
",
  "roles": [
    "PlatformSchool"
  ],
  "scp": "PlatformSchoolUser",
  "sub": "KiACi2AT9qClGYHFHlUjnutfKhGn0JdRMw1QdGYmuLs",
  "tid": "f70021fb-e41e-4b11-b7d3-916059575dd6",
  "uti": "0ir3G3DJNEyvEwwtjv4MAA",
  "ver": "2.0"
}
```


MVP-Dagen

Challenges in Azure Services Authentication

Operations

Using Accounts, Credentials and Secrets for Automation or Management Operations

Policy exemptions & monitoring security breach

Overprivilege, permission gap

Example: Logic App needs to Start or Stop a VM

Development

–Managing Secrets and Credentials between Components in a Solution

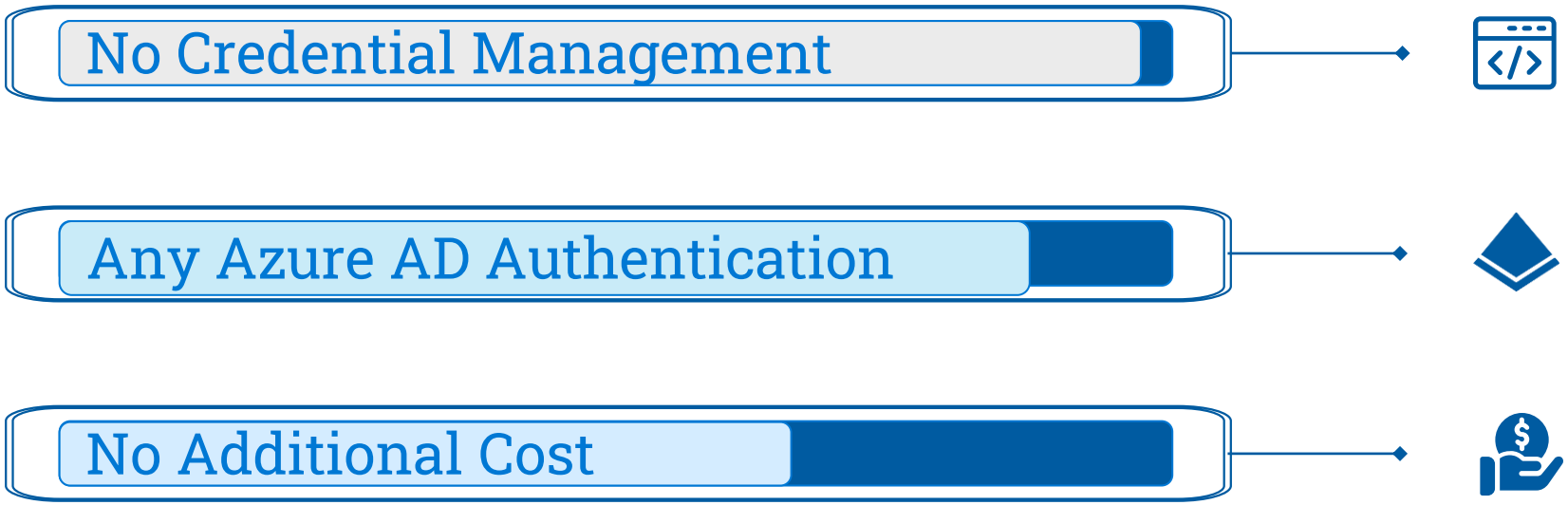
–Lifecycle challenges

–Example: App Service needs to access Azure SQL, Storage Account, Key Vault etc.

“Managed Identity is an Identity Connected to your Azure Service”

Can be Used for Connecting to any Resources that support Azure Active Directory (Azure AD) Authentication

Benefits of Managed Identities



Types of Managed Identites

System-Assigned

Part of specific resource

Lifecycle follows resource

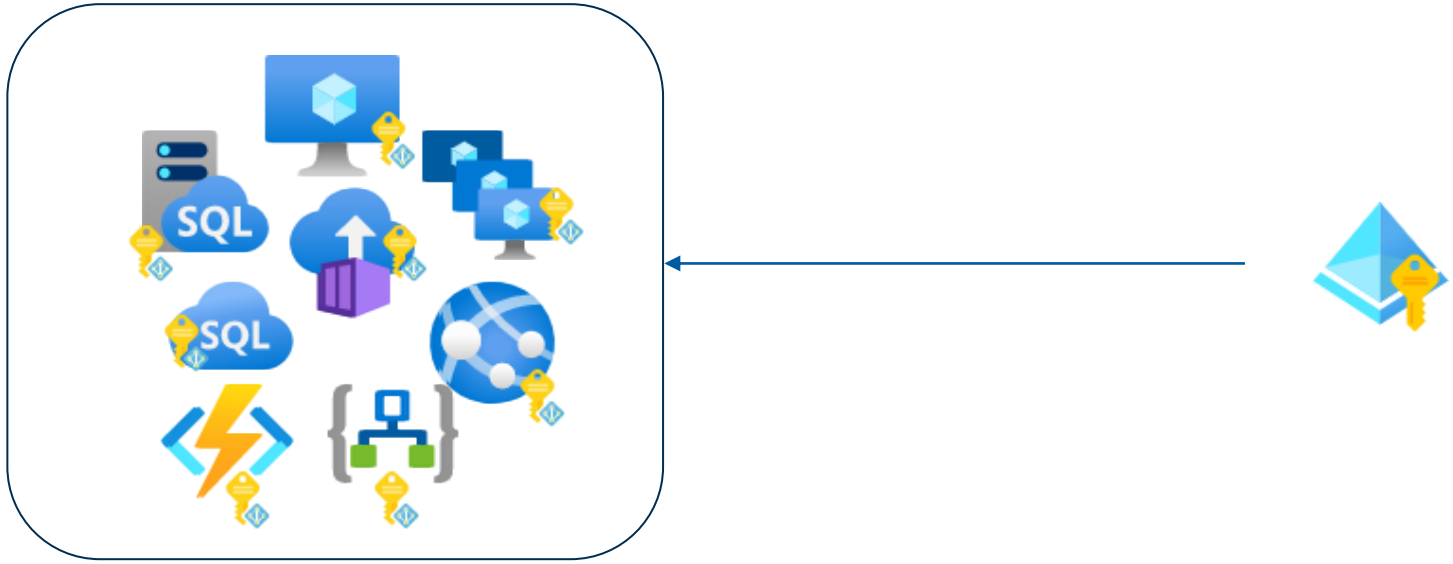
Cannot be shared

User-Assigned

Standalone Azure resource

Independent lifecycle

Shared between Azure resources



I can use Managed Identities when…

As a Developer, I want to build an application using

Source:

Azure Resources
Azure VMs
Azure App Services
Azure Functions
Azure Container instances
Azure Kubernetes Service
Azure Logic Apps
Azure Storage
….

that accesses

Target:

Any target that supports Azure Active Directory Authentication:
- **Your applications**
- **Azure Services:**
  - Azure Key Vault
  - Azure Storage
  - Azure SQL…

without having to manage any credentials!

For example, I want to build an application using ***Azure App Services*** that accesses **Azure Storage** without having to manage any credentials.

I can use Managed Identities when…

OPS… SEC…

As a ~~Developer~~, I want to build an application using

Source:

Azure Resources
Azure VMs
Azure App Services
Azure Functions
Azure Container instances
Azure Kubernetes Service
Azure Logic Apps
Azure Storage
….

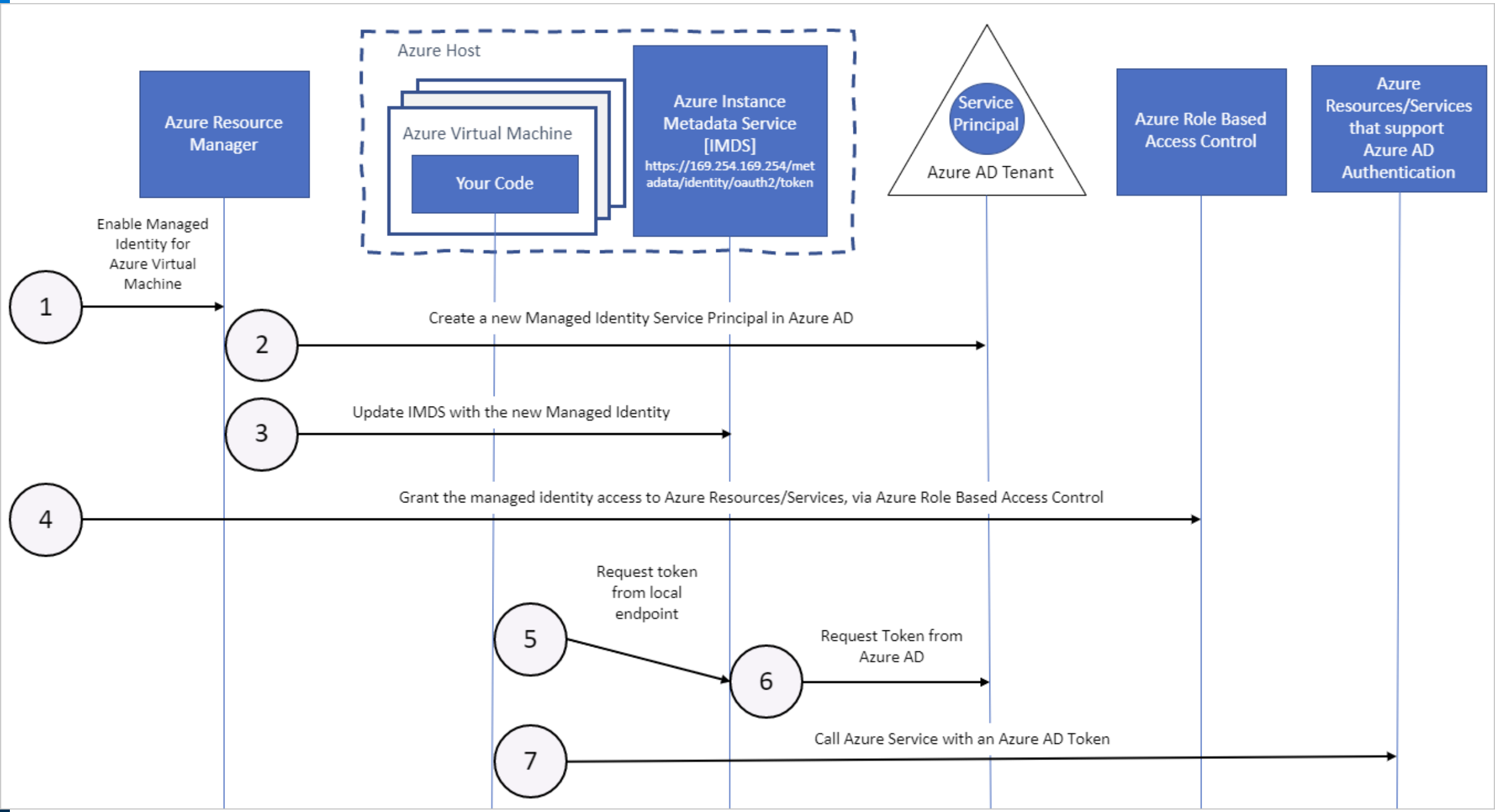
that accesses

Target:

Any target that supports Azure Active Directory Authentication:
- **Your applications**
- **Azure Services:**
  - Azure Key Vault
  - Azure Storage
  - Azure SQL…

without having to manage any credentials!

For example, I want to build an application using ***Azure App Services*** that accesses **Azure Storage** without having to manage any credentials.


Azure Host
Azure Virtual Machine
Your Code
Azure Instance Metadata Service [IMDS]
https://169.254.169.254/metadata/identity/oauth2/token
Azure Resource Manager
Service Principal
Azure AD Tenant
Azure Role Based Access Control
Azure Resources/Services that support Azure AD Authentication
Enable Managed Identity for Azure Virtual Machine
1
2
Create a new Managed Identity Service Principal in Azure AD
3
Update IMDS with the new Managed Identity
4
Grant the managed identity access to Azure Resources/Services, via Azure Role Based Access Control
Request token from local endpoint
5
6
Request Token from Azure AD
7
Call Azure Service with an Azure AD Token

Takk til våre sponsorer

audiocodes

glasspaper

POINT:TAKEN

EPOS

aztek

Evidi

KPMG

CloudWay

spirhed

Noroff
School of technology
and digital media

blinQ

amesto
Fortytwo

SparebankenVest

IT styring

INNOFACTOR

MVP-Dagen


Tusen takk!
MVP-Dagen