

The logo for SECTOR 2021 Hybrid Event features the word "SECTOR" in large, bold, white capital letters. A stylized white outline of the CN Tower is positioned behind the letter "O". Below "SECTOR" is the text "2021 HYBRID EVENT" in a smaller, white, sans-serif font. The background is a dark blue gradient with a starry, particle-like effect.

**SECTOR**

2021 HYBRID EVENT

# Secure and Scalable Development with Microsoft 365 and Azure AD

Wed, Nov 3, 2021

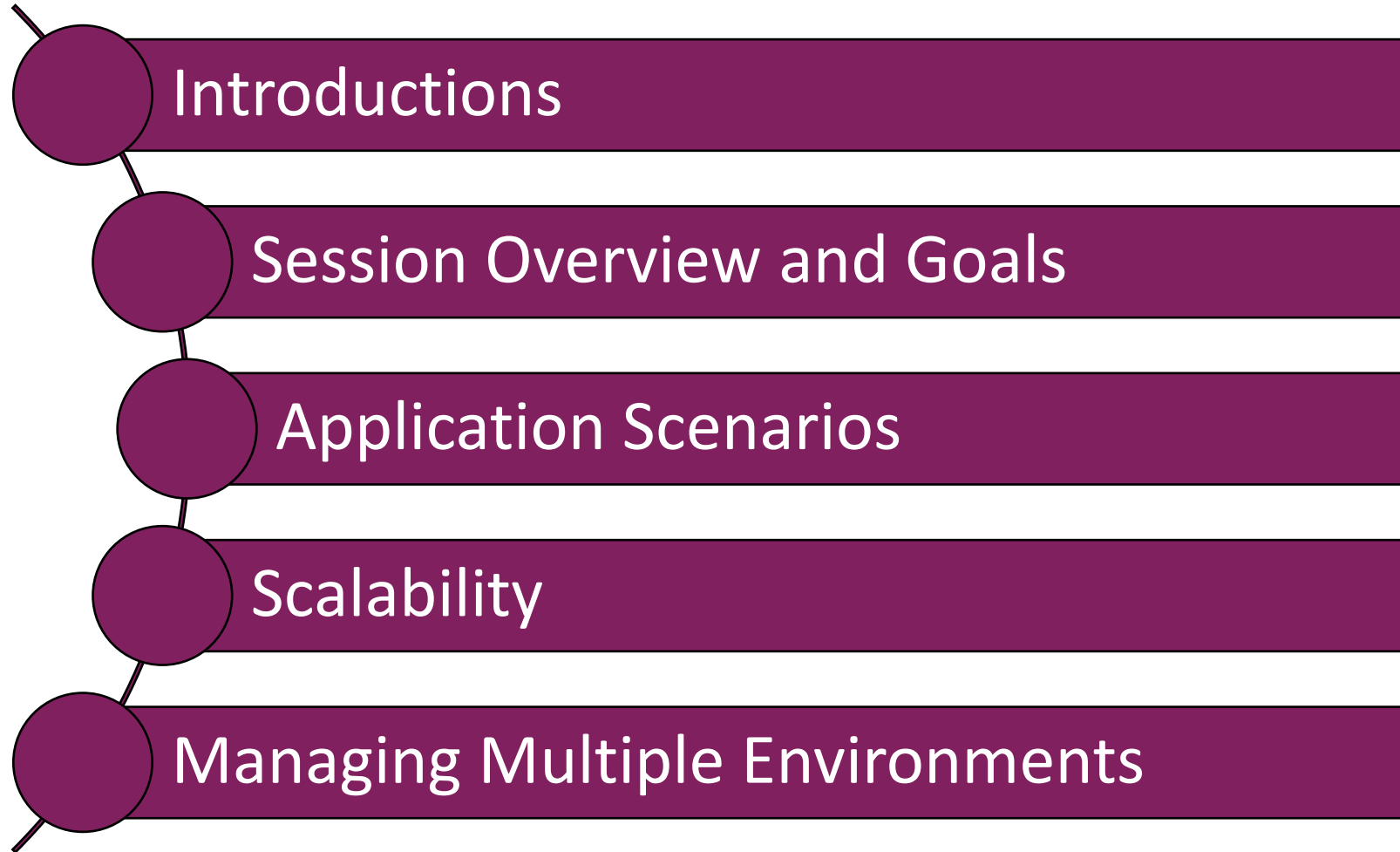
# Peter Carson



- President, Envision IT and Extranet User Manager
- 11-time Office Apps and Services Microsoft MVP
- [peter@envisionit.com](mailto:peter@envisionit.com)
- [blog.petercarson.ca](http://blog.petercarson.ca)
- [www.envisionit.com](http://www.envisionit.com)
- [www.extranetusermanager.com](http://www.extranetusermanager.com)
- Twitter @carsonpeter
- President Toronto SharePoint User Group



# Agenda



# Session Overview and Goals

Open and  
transparent

Review of how we  
build our modern,  
cloud-based apps

Not a product  
pitch

Azure Active  
Directory is the  
security  
infrastructure

Secure and  
scalable Microsoft  
365 applications is  
the goal

Managing non-  
prod  
environments

# Security Through Obscurity in SharePoint Online

- **Hiding <> Securing**
- **Hidden lists**
- **Hiding SharePoint columns**
- **Hiding SharePoint as a whole**
- **Lesser evils**
  - Workflows that move entries into more secure lists
  - Permissions only to items you have created

# Application Scenarios



Teams Provisioning  
Open Source Solution



EUM Suite

# Links

- **Teams Provisioning**

- [www.envisionit.com/products/teams-provisioning](http://www.envisionit.com/products/teams-provisioning)

- **Extranet User Manager**

- [www.extranetusermanager.com](http://www.extranetusermanager.com)

- **Event Page**

- [www.extranetusermanager.com/resources/events/sector-conference-2021](http://www.extranetusermanager.com/resources/events/sector-conference-2021)

# Teams Provisioning Open Source Solution



- **Teams governance**
- **Self-service request form for a new Team**
- **Approval workflow**
- **Workspace templates**
  - Teams tabs and channels
  - Team site structure
  - OneNote
  - Planner



# Technology Stack

Team / Site Request

Division \*  
Demo

Site Template \*  
Modern Team Site

Title \*

Purpose

Alias

Public Group  
 Yes

Create Team  
 Yes

Submit Cancel

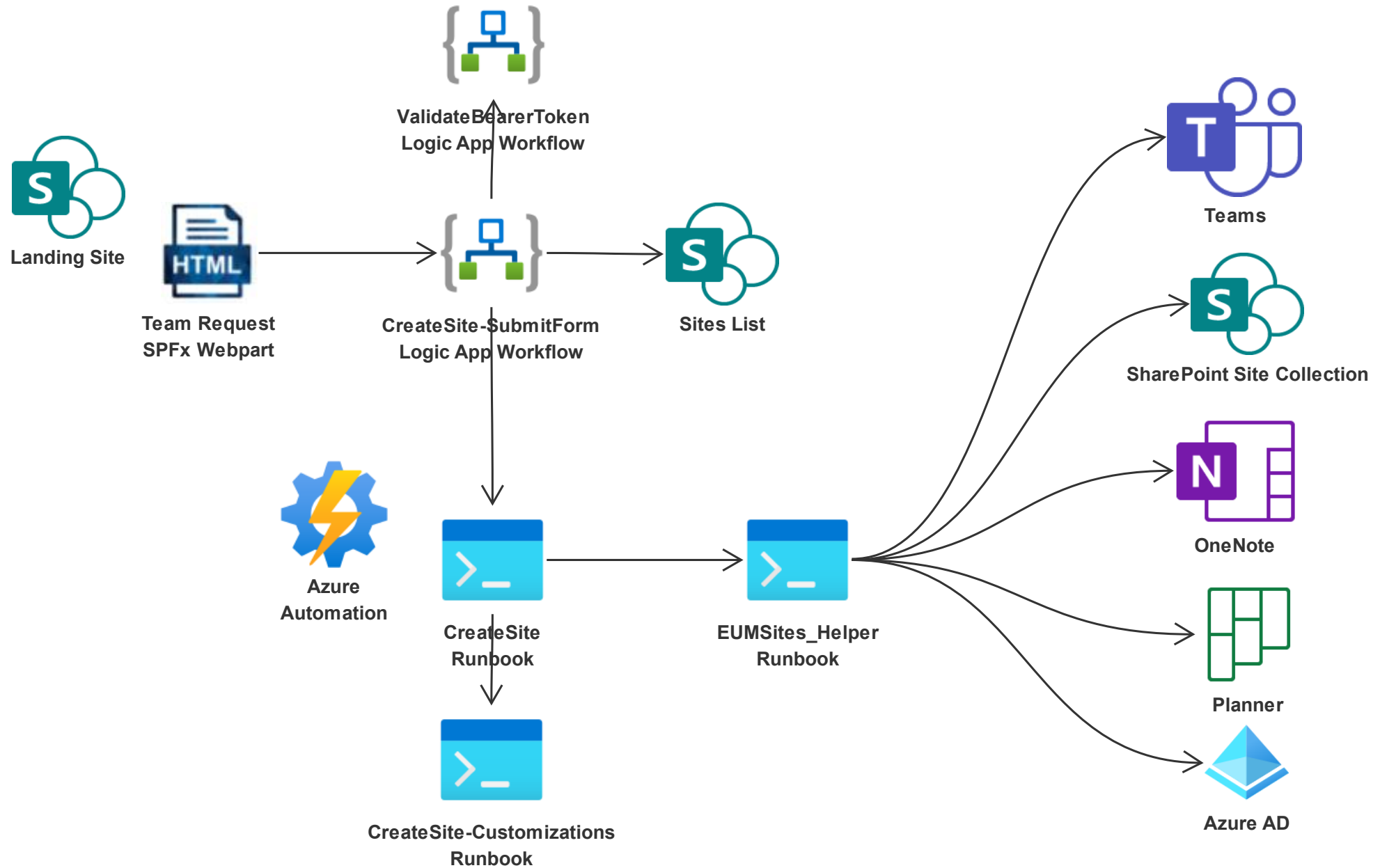


# Zero Trust Security Design

- **Never trust, always verify**
- **Client side apps and JavaScript are inherently untrusted**
  - Any user can use Developer Toolbar to manipulate variables and change code paths
  - Secrets such as shared access signatures are not secret
  - Business rules can be bypassed
- **APIs should not trust their callers**
  - Do you know who is calling you? That should never be a provided parameter
  - Access tokens are the best way to validate callers
  - Verify all of your parameters
- **Browsers are an untrusted environment**
  - Any secure code needs to run in an access controlled server environment
  - Can still be serverless like Azure Automation, Logic Apps, or App Services

# Using Logic Apps or Power Automate as a Secure API

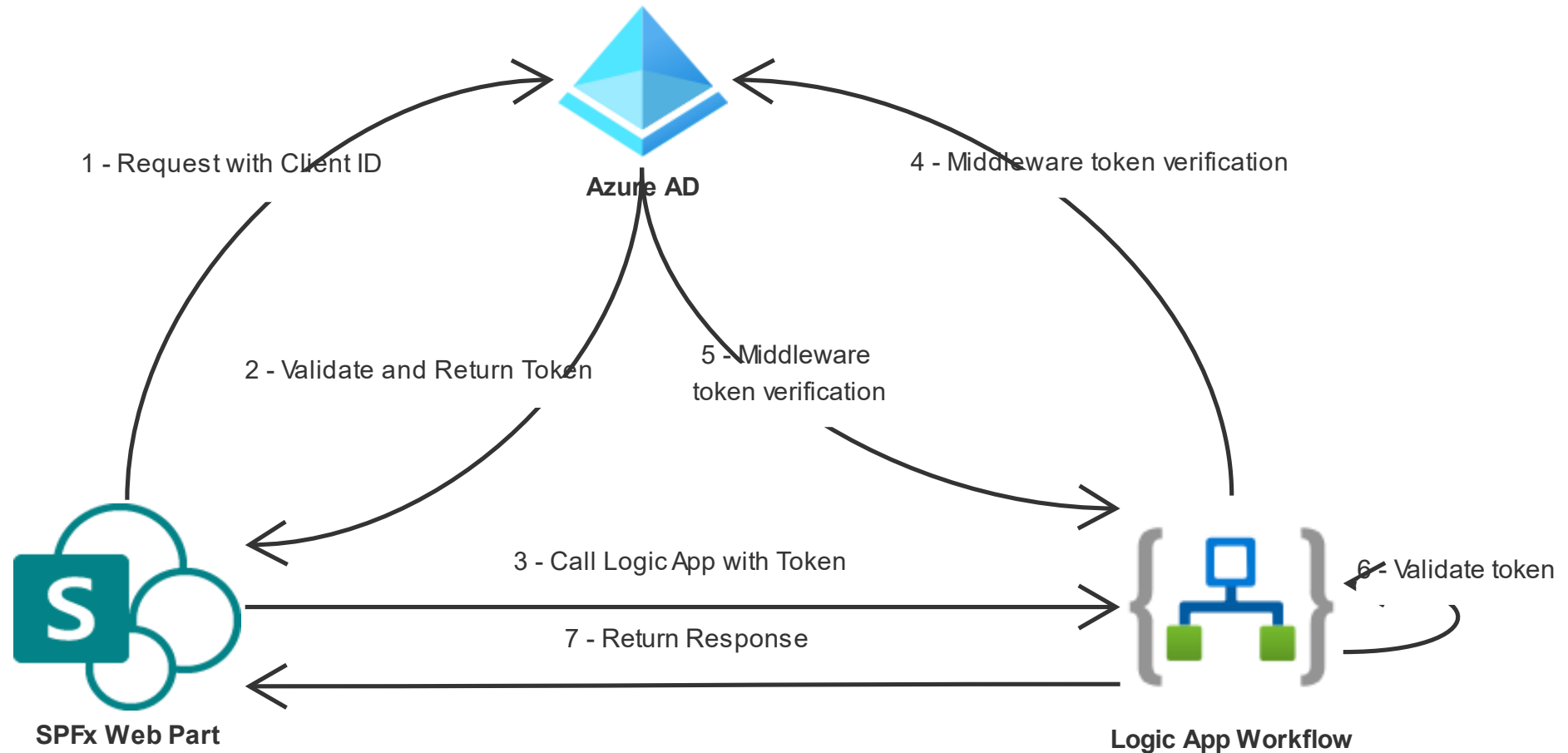
- **Trigger for the workflow is an HTTP POST REST method**
  - Could also be a GET, PUT, PATCH, or DELETE
- **Logic Apps / Flow generates a URL**
  - Includes a Shared Access Signature secret
- **Can also configure Azure AD authentication**
  - Register an app in Azure AD
  - Record the Client ID in the Logic App Authorization Policy
- **SharePoint Framework (SPFx) webparts have plumbing to support this**



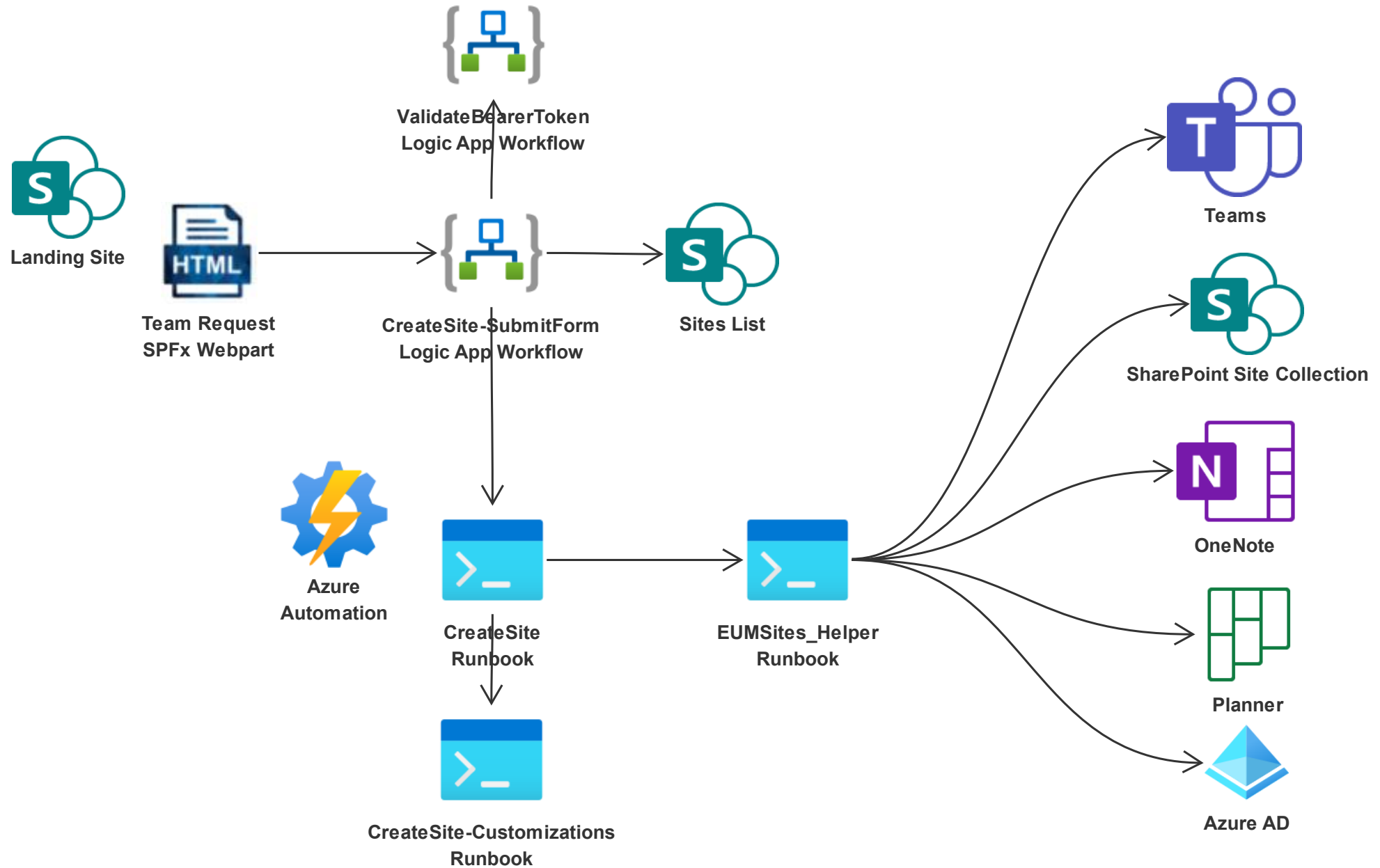
# Demo

- **Landing Site – Team Request**
- **Submit Form Logic App**
  - Signature in trigger
  - Authorization
  - Call to Validate Bearer Token
  - Return of UPN
- **Validate Bearer Token**
  - Different format of User Strings

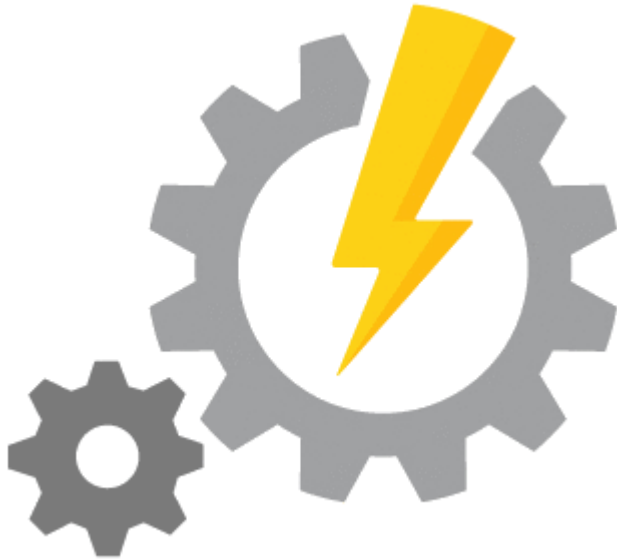
# OAuth2 SPFx to Logic App Authentication Flow



<https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow>



# Azure Automation

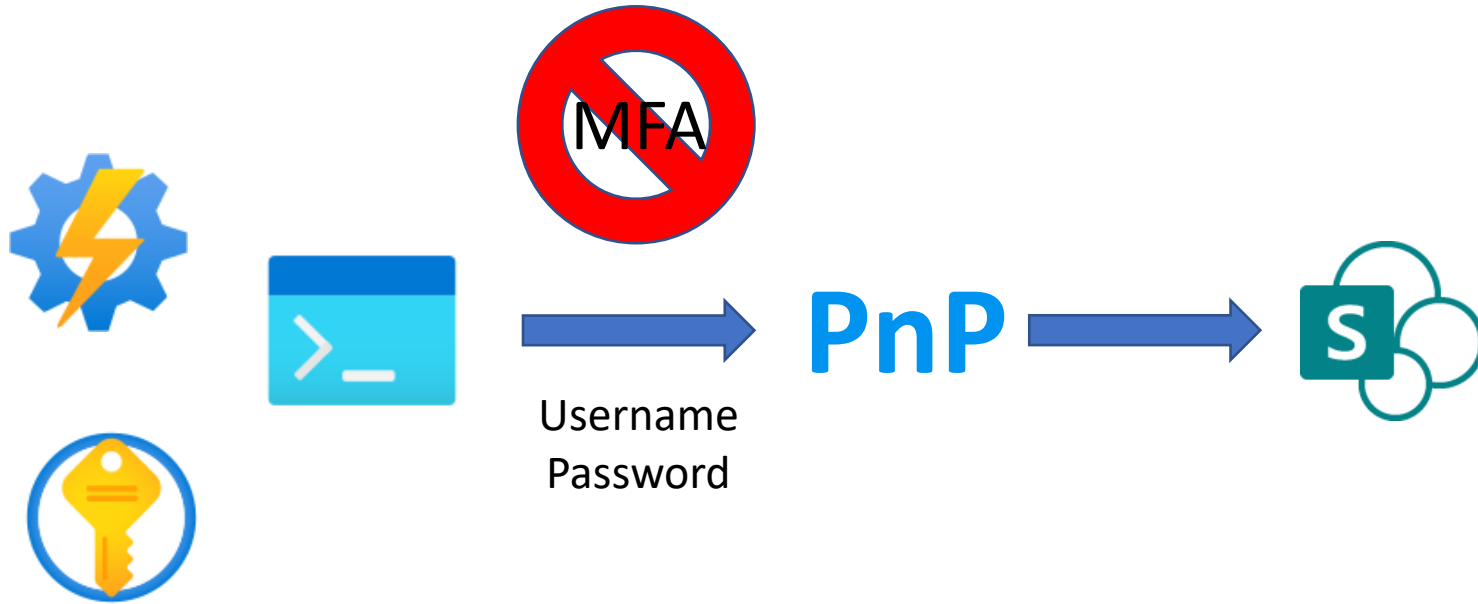


- **Run PowerShell scripts in the cloud**
- **No management of the VM needed, Azure takes care of that**
- **Very cost effective**
  - 500 minutes of runtime included free per month
  - \$.002/minute USD after that

<https://azure.microsoft.com/en-ca/services/automation>



# How Not To Do Security

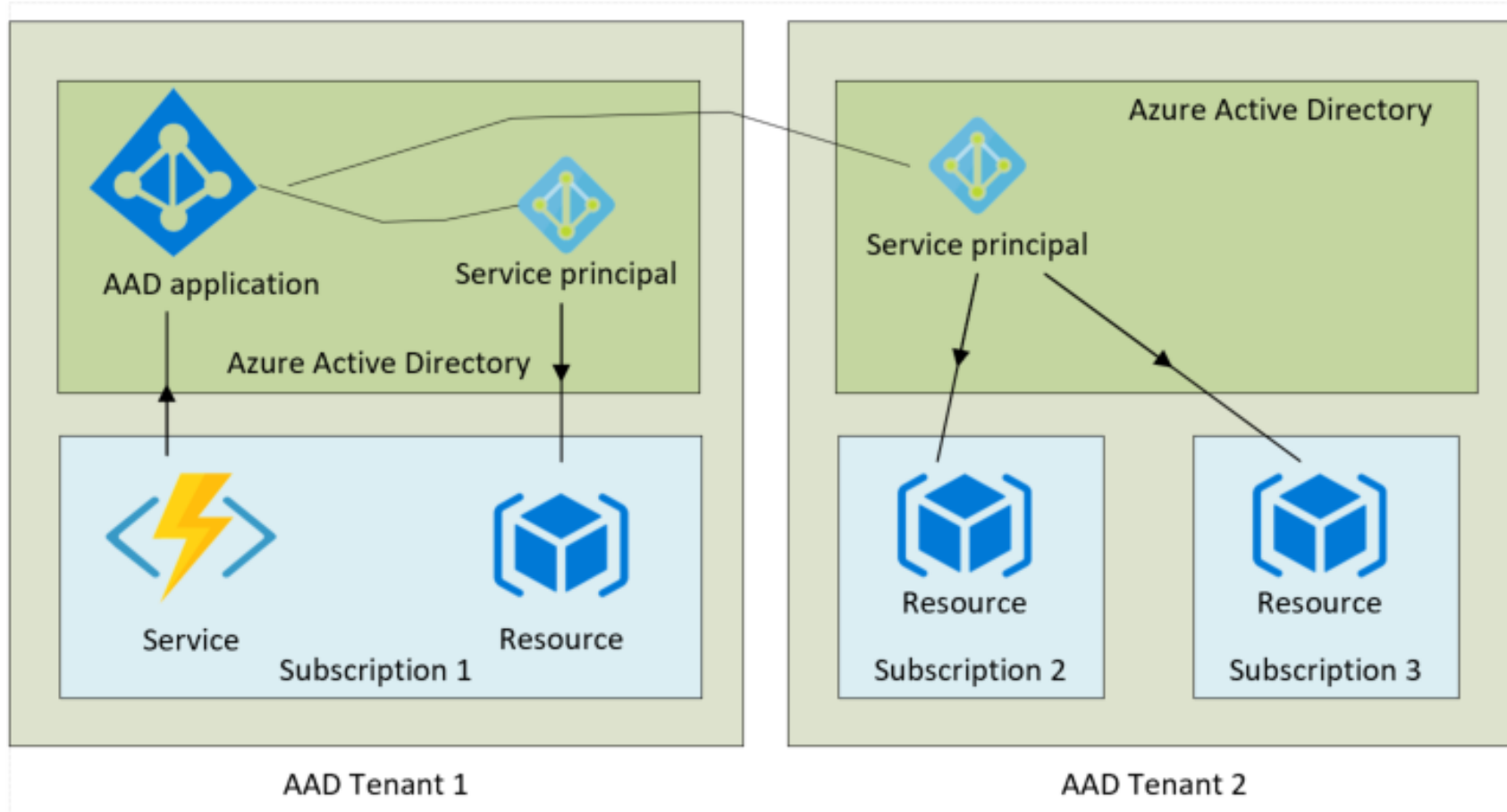


- Username and password in Credentials
- Stored in private Key Vault
- Account needs to have MFA disabled
- Big security hole
- Use a Service Principal

# User Objects, Application Objects and Service Principals

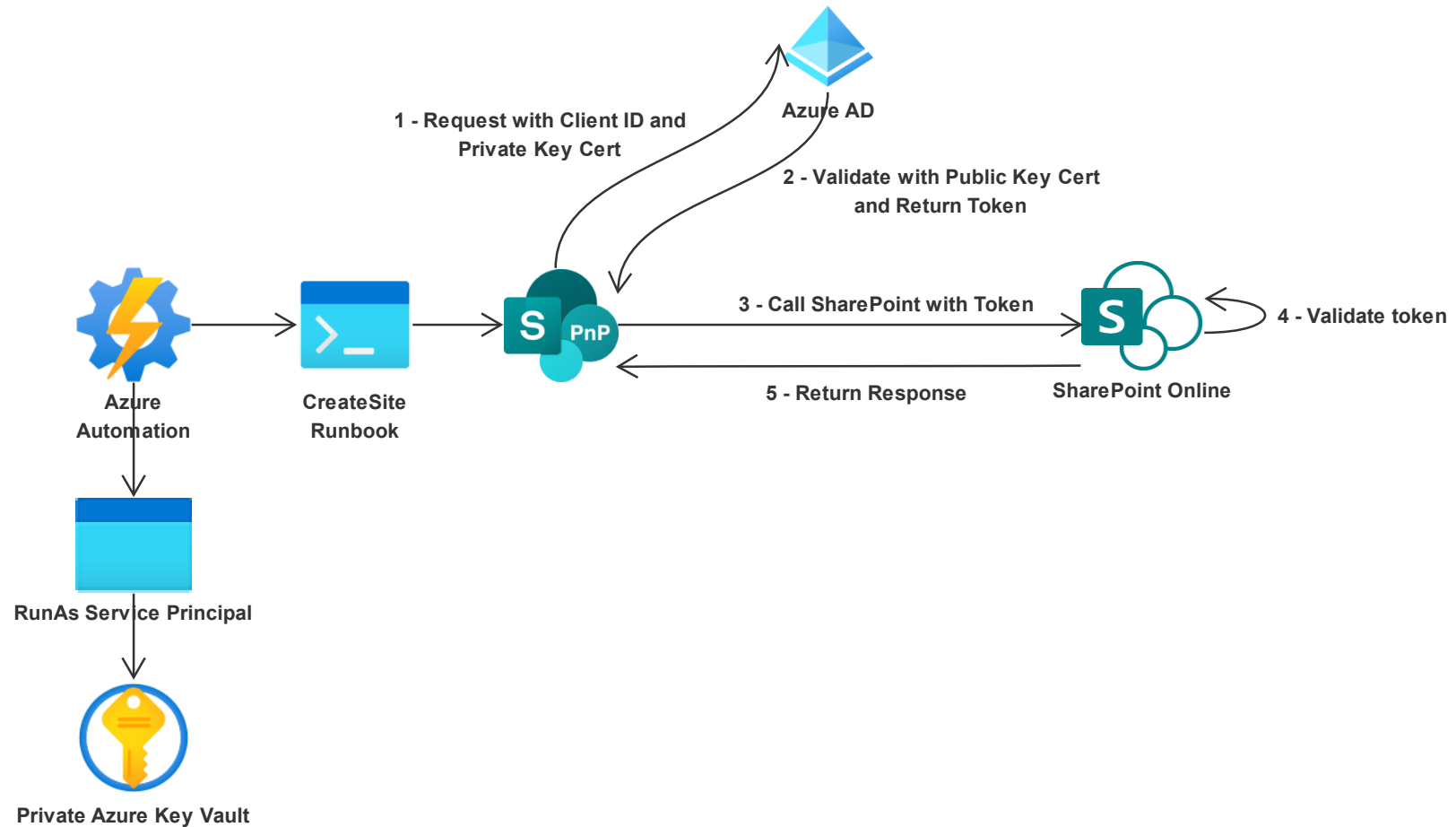
- **Common practice was often to use user credentials in integration scenarios**
  - Credentials could be stored securely in Azure Key Vault
- **MFA breaks this**
  - Daemon process or back end code can't process an MFA request
- **Disabling MFA on "service" accounts is a big security hole**
- **Service principals are the preferred approach**
- **Authentication done through a client ID and secret or certificate**
- **Certificates are the preferred approach**
  - Azure App Service will manage certificates
  - Azure Automation Run As accounts are service principles with managed certificates
  - Private Azure Key Vault under the hood of both

# Application Objects and Service Principals



<https://endjin.com/blog/2019/01/managing-applications-using-azure-ad-service-principals-and-managed-identities>

# OAuth2 Azure Automation to SharePoint Authentication Flow



# Running in Azure Automation

1. **Create an Azure Automation Account**
2. **Leave the Create Azure Run As account on as Yes**
3. **Once provisioned, open and go to Run as accounts and open the account**
4. **Copy the Display Name**
5. **Go to App Registrations in Azure AD**
6. **Filter for the Client ID**
7. **Assign the appropriate API permissions**
8. **Create your runbooks, and use the service principal to authenticate**

# Demo

- **Azure Automation Account**
  - RunAs
- **Azure AD App Registrations**
  - API Permissions

# Running in Azure Automation

```
# Get Azure Run As Connection Name
$connectionName = "AzureRunAsConnection"
# Get the Service Principal connection details for the Connection name
$servicePrincipalConnection = Get-AutomationConnection -Name $connectionName

$Conn = Connect-PnPOnline -Tenant $servicePrincipalConnection.TenantId -ClientId
$servicePrincipalConnection.ApplicationId -Thumbprint
$servicePrincipalConnection.CertificateThumbprint -Url $URL -ReturnConnection
```

# Testing Authentication Locally

1. Register your app in Azure AD
2. Add the appropriate API permissions
3. Obtain a certificate or create a self-signed certificate
4. Upload the public key certificate (.cer) to the Azure AD App Registration
5. Install private key certificate (pfx) in Personal store if running locally (MMC Certificates snap-in for My user account)
6. Use the Client ID from Azure AD and the thumbprint from the certificate to authenticate
7. Use the new PnP
  1. Remove the SharePointPnPPowerShellOnline module
  2. Import the PnP.PowerShell module

<https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/security-apponly-azuread>



# SolarWinds Hack and Golden Tickets / Golden SAML

- **Supply chain hack**
  - Sophisticated hackers injected malware into SolarWinds products
- **Affected top-level US federal agencies and nongovernment organizations**
  - 18,000 customers installed updates with vulnerabilities
- **Password spray was used extensively**
- **Once systems were initially compromised, SAML private key certificate was compromised to allow signing forged SAML tokens**
  - These can be used to validate other SSO systems
  - Can impersonate any user and roles
  - MFA and password change have no impact
- **These can be used to access any federated system**
- **Key is gaining control of the private key certificate**

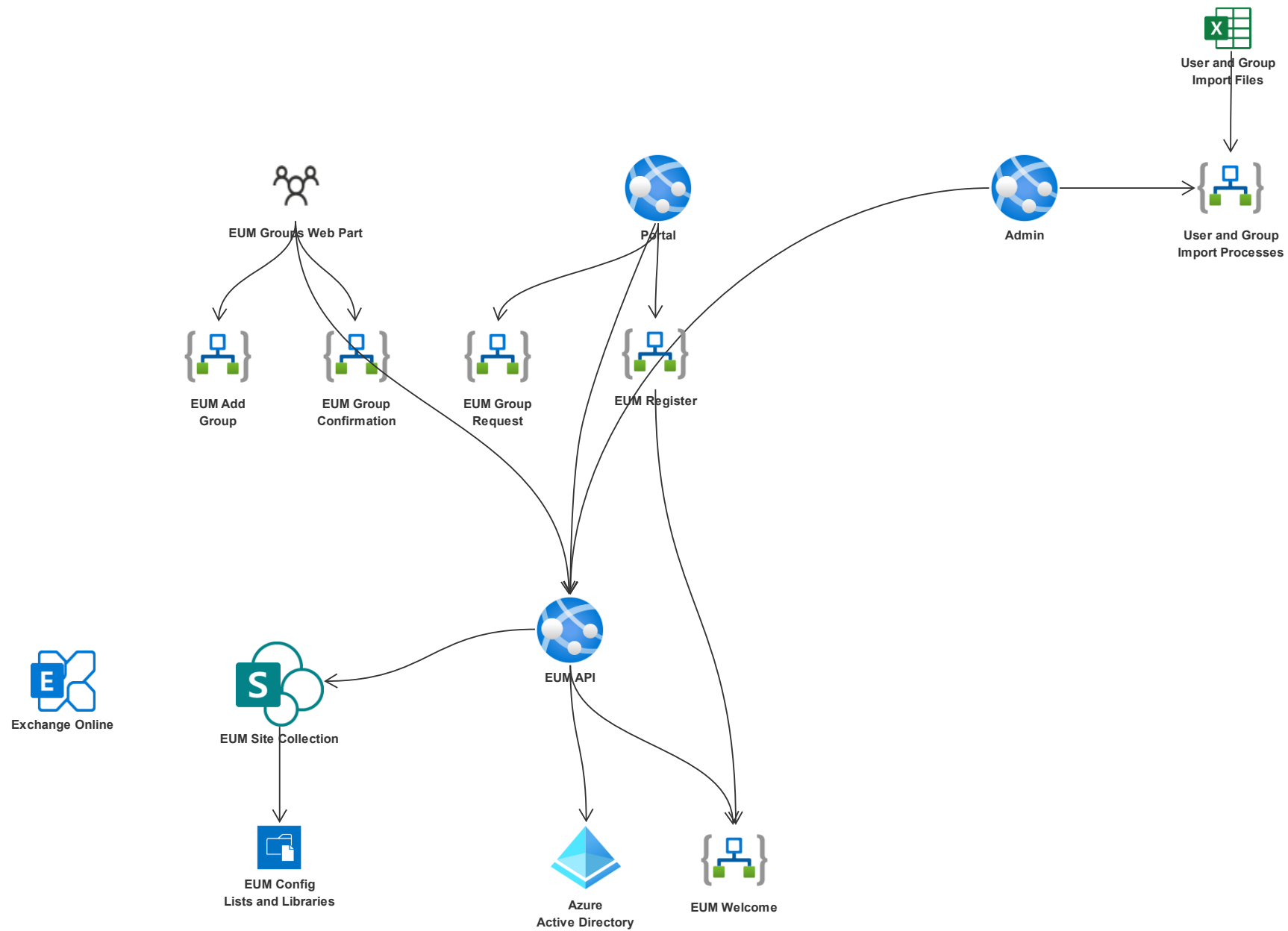
# Best Practices for Certificate Management

- **Certificates need to be stored securely in locations such as Azure Key Vault**
  - Consider Premium Hardware Security Module protected keys
- **Rotate private keys regularly**
- **Never create a production local certificate for testing purposes**
  - If one does exist remove the public certificate so it is no longer valid
- **Ideally let Azure create and manage the certificates**

# EUM Suite

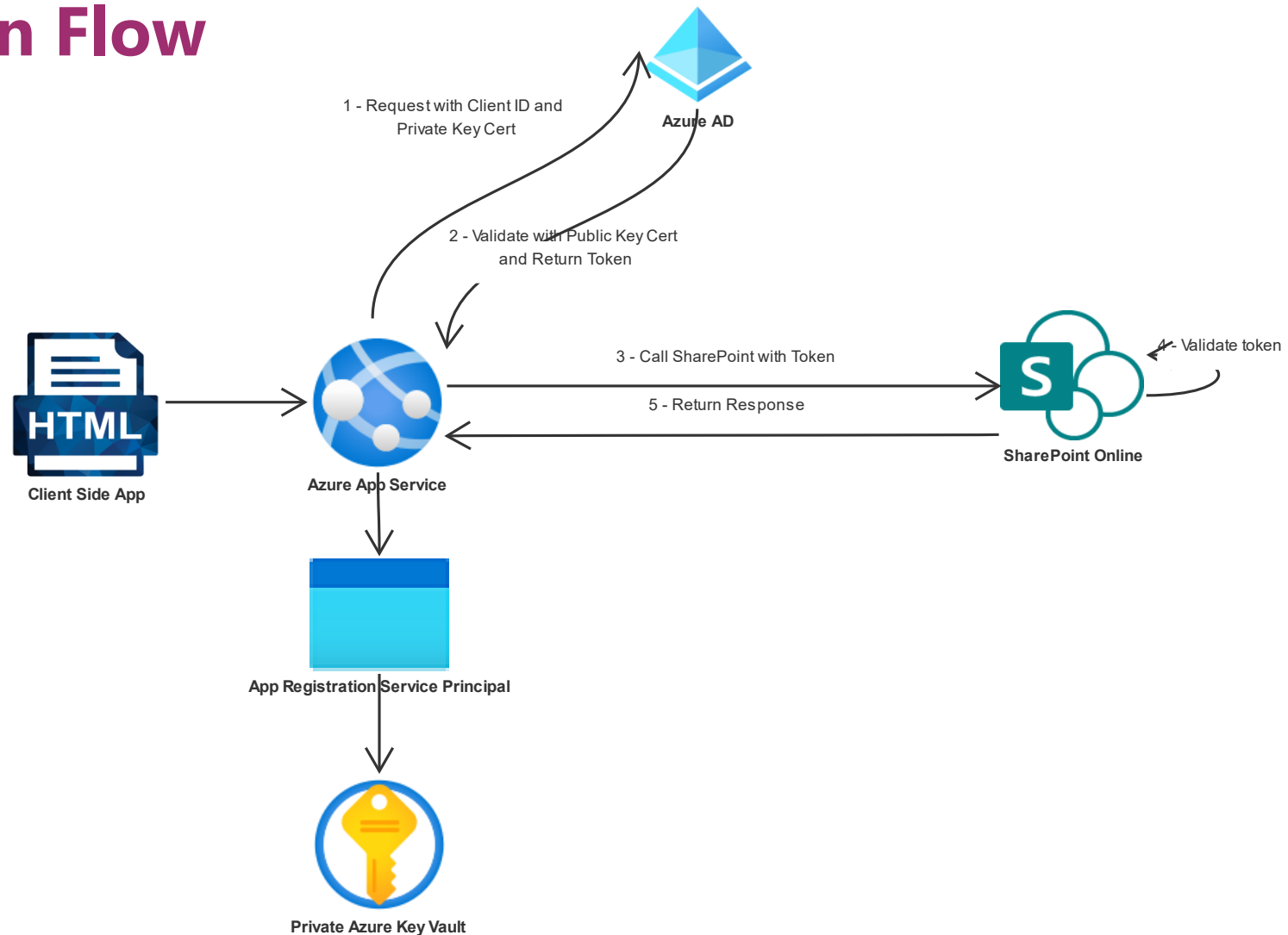


- Angular front-end UI
- .NET Core 5 middleware
- Hosted in Azure App Services
- Built on top of Azure AD B2B and SharePoint Online



# EUM Admin Demo

# OAuth2 Azure App Service to SharePoint Authentication Flow



# API Permissions

## Delegated

- **Need a user token to request a delegated token**
- **Typically only applicable for interactive applications**
- **Permissions are applied to the app registration**
- **Delegated user also needs appropriate permissions**
- **Some APIs only support delegated permissions**
  - Planner Tasks

## Application

- **Runs in the context of the application**
- **User can be set in SharePoint updates to show the correct user in version history**
- **Auditing happens in the context of the app**
- **Permissions tend to be very broad**
  - Sites.FullControl.All has access to ALL site collections
  - Graph Sites supports selected sites

# Demo

- **sector-eum-demo-a\_EUM\_API API permissions**



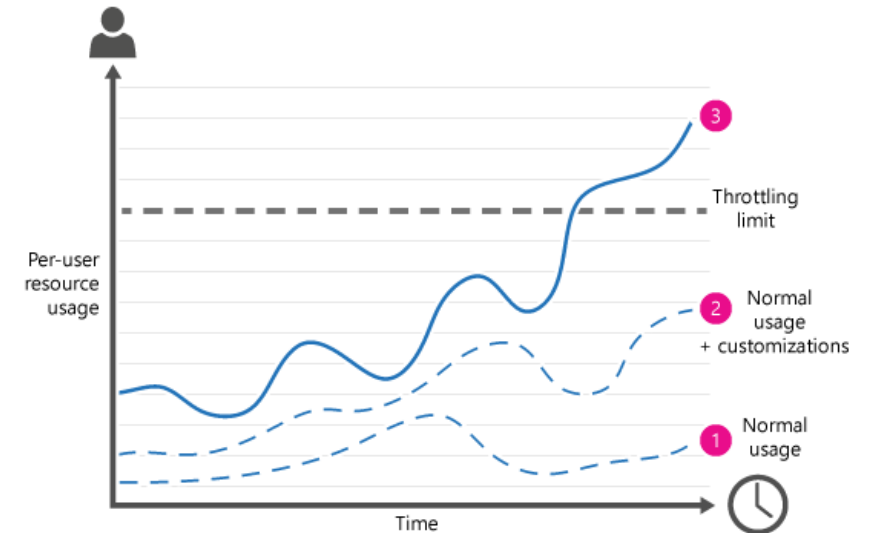
# Throttling

- Used by Microsoft to prevent overuse of resources
- REST calls fail with 429 ("Too many request") or 503 ("Server busy")
- 429 provides a recommended wait before retry
- Ignoring may block completely
- Microsoft does not provide many details or metrics

## Recommendations

- Decorate your traffic
- Respect retry recommendations

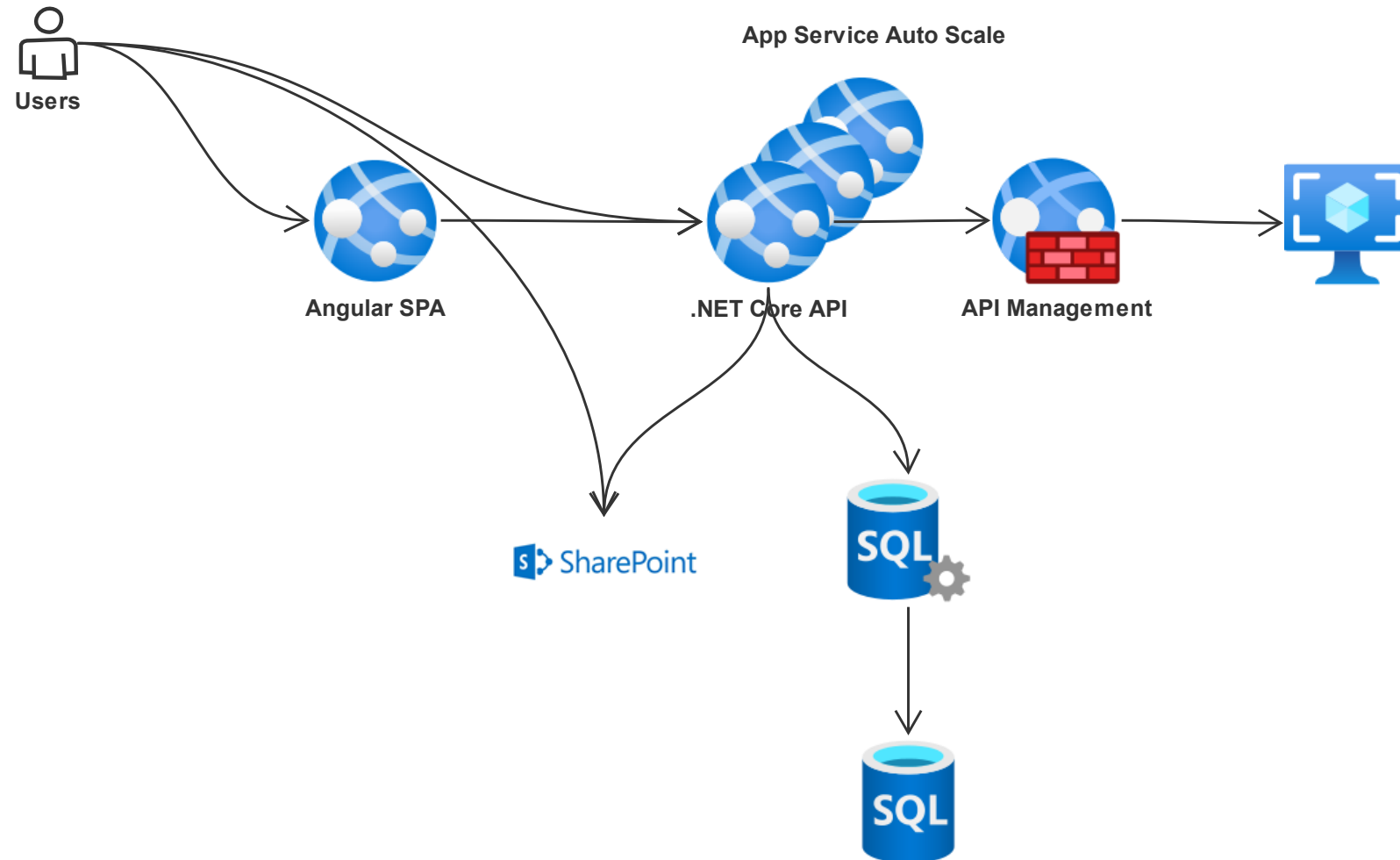
<https://docs.microsoft.com/en-us/sharepoint/dev/general-development/how-to-avoid-getting-throttled-or-blocked-in-sharepoint-online>



# User vs. App Throttling

- **User throttling is based on requests per user per second**
  - No defined SLA
  - Depends on the overall usage of SharePoint Online
- **Delegated App requests are treated like user requests**
  - 300 users accessing SharePoint through a delegate app looks like 300 users accessing SharePoint
- **Application Permission requests are treated like one app**
  - 300 users accessing SharePoint through an app-only app looks like one user accessing SharePoint
  - Higher threshold than user requests, but not defined
- **In an App Service hosted model, scale out with nodes not just for CPU and memory, but also App Registrations**
  - Pool of App registrations stored in SQL
  - As new API nodes come online, use additional App Registrations to spread the requests
  - Reduces the risk of throttling

# Scalable Architecture



# Importance of Managing Multiple Environments

Development



Azure AD

QA



Azure AD

Production

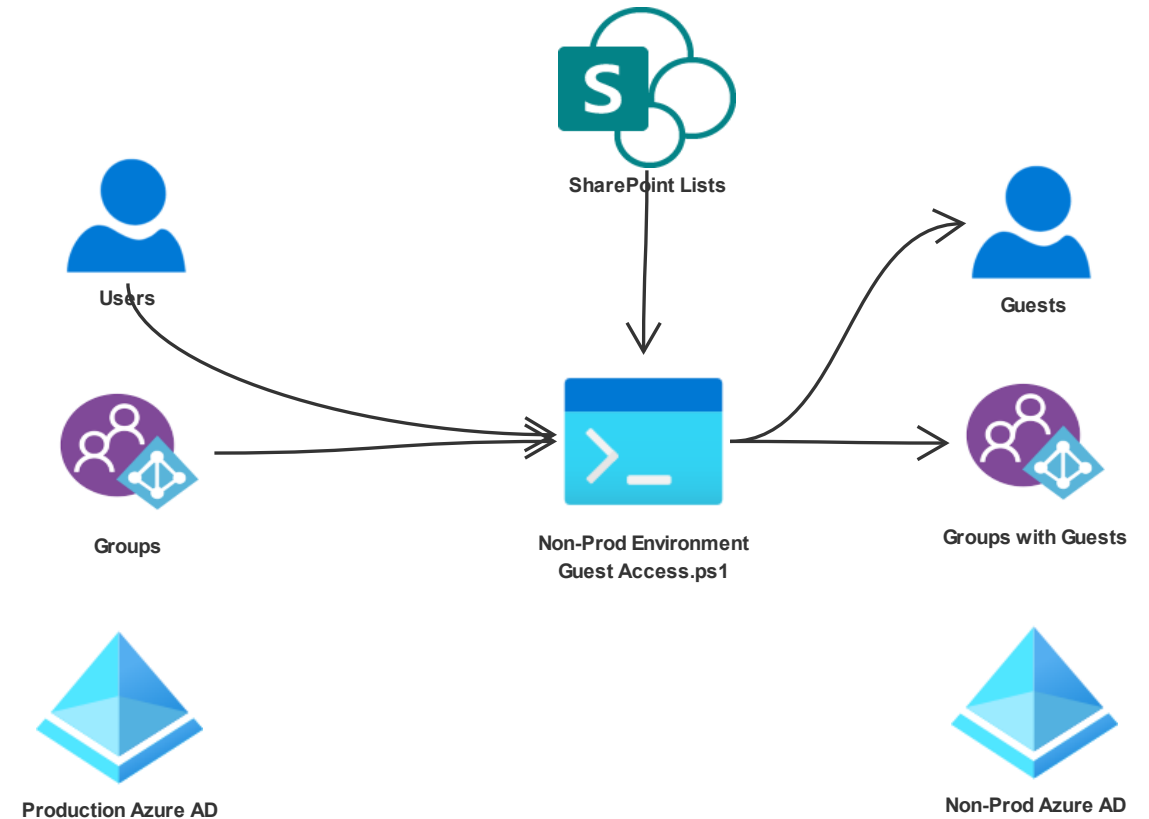


Azure AD

- **Completely separate tenants for each environment**
- **Not just another site collection**
- **Full isolation between environments**
- **Developer subscriptions are free - [Developer Program - Microsoft 365](#)**
  - These do expire if left unused
- **May want to consider paid tenants for non-prod with 1-2 users**
  - ~ \$10 / month

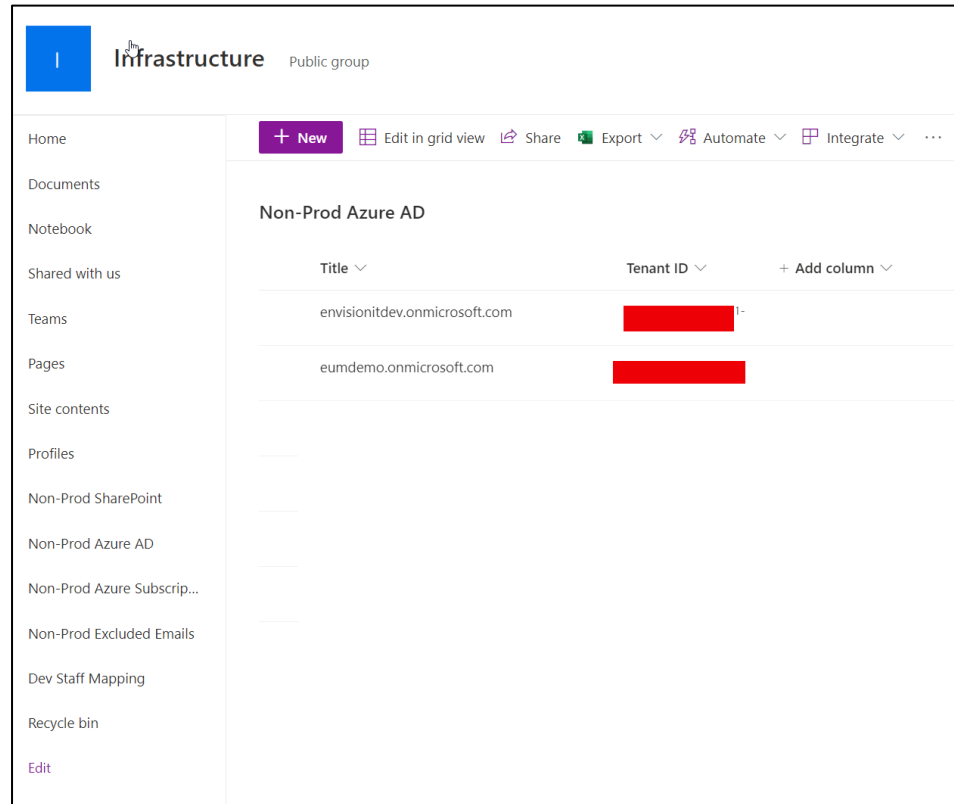
# Using Azure AD B2B to Manage Access to Environments

- PowerShell script syncs prod groups and users as Guests in non-prod
- Rights can be assigned in SharePoint and Azure to these groups
- Developers use their prod credentials
  - No browser profiles or InPrivate sessions needed



# SharePoint Lists to Manage Environment Info

- SharePoint Lists to manage all different tenants and subscriptions
- Single tenant can have multiple Azure subscriptions



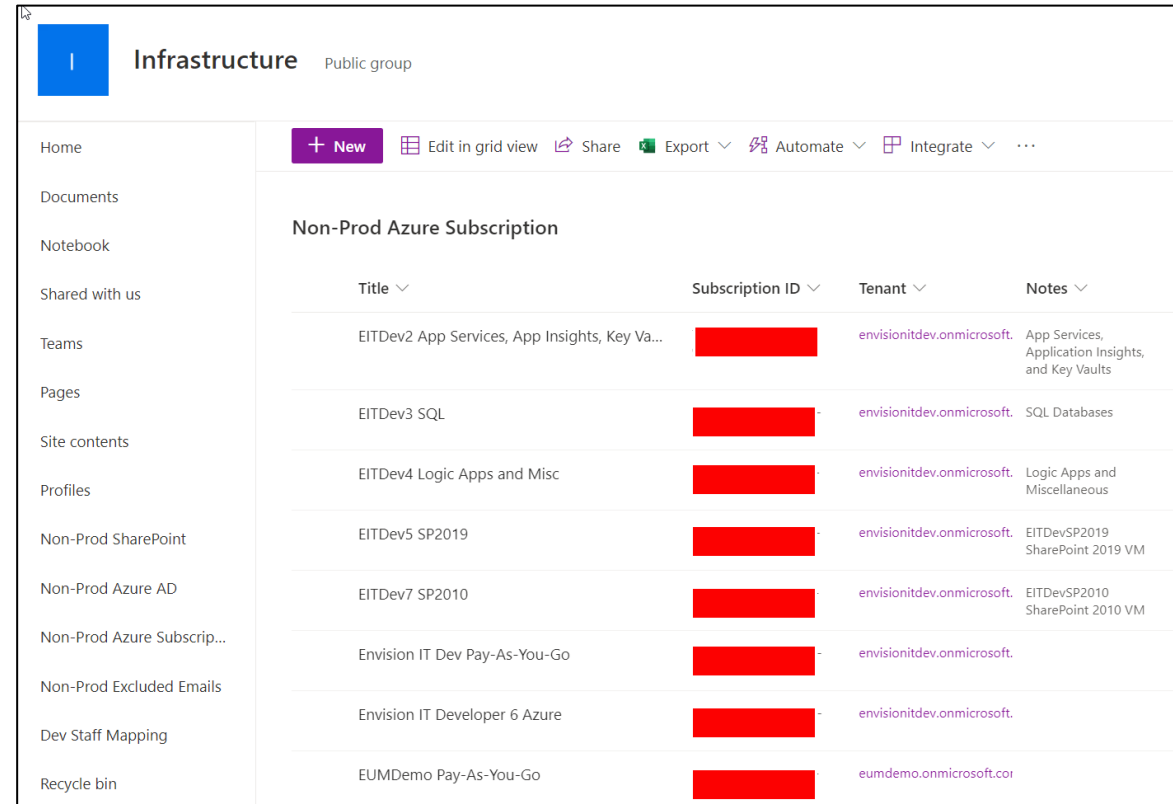
Infrastructure Public group

Home Documents Notebook Shared with us Teams Pages Site contents Profiles Non-Prod SharePoint Non-Prod Azure AD Non-Prod Azure Subscrip... Non-Prod Excluded Emails Dev Staff Mapping Recycle bin Edit

+ New Edit in grid view Share Export Automate Integrate ...

### Non-Prod Azure AD

Title	Tenant ID	+ Add column
envisionitdev.onmicrosoft.com	[REDACTED]	
eumdemo.onmicrosoft.com	[REDACTED]	



Infrastructure Public group

Home Documents Notebook Shared with us Teams Pages Site contents Profiles Non-Prod SharePoint Non-Prod Azure AD Non-Prod Azure Subscrip... Non-Prod Excluded Emails Dev Staff Mapping Recycle bin

+ New Edit in grid view Share Export Automate Integrate ...

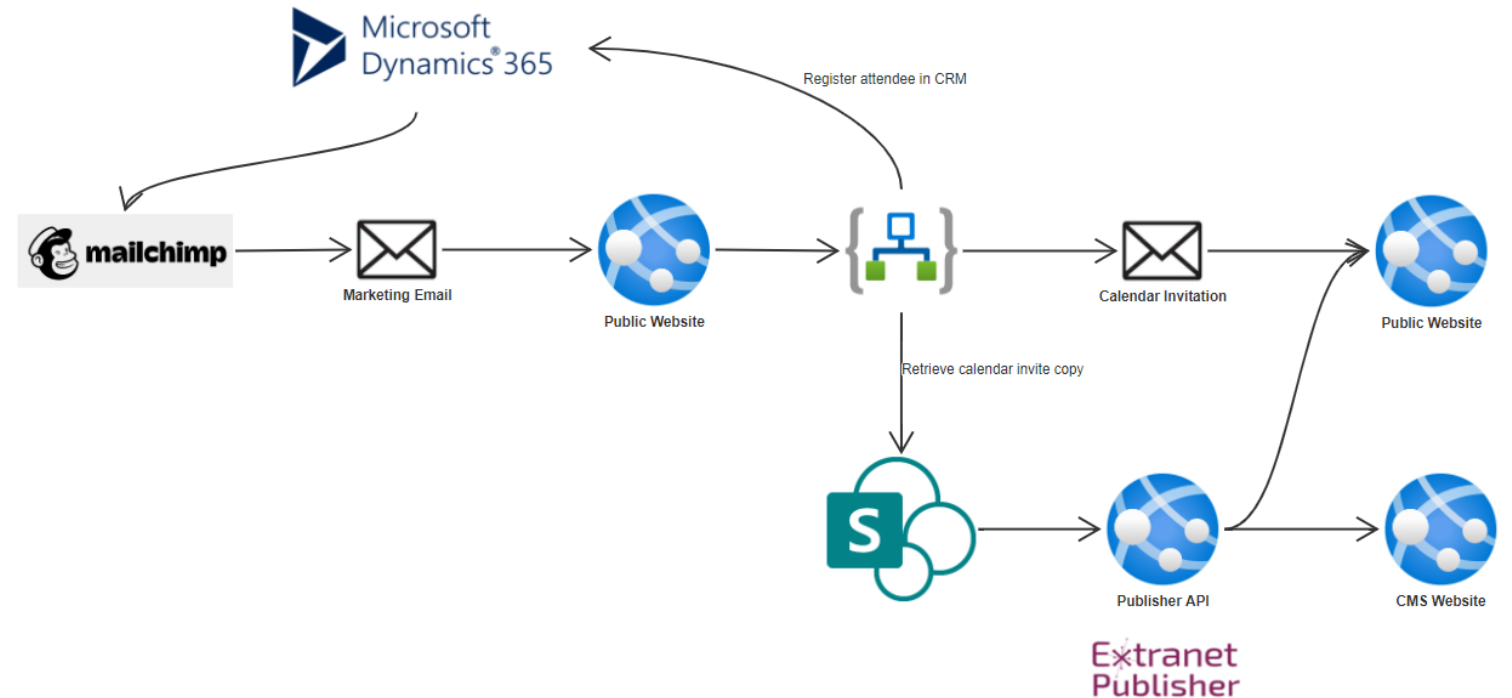
### Non-Prod Azure Subscription

Title	Subscription ID	Tenant	Notes
EITDev2 App Services, App Insights, Key Va...	[REDACTED]	envisionitdev.onmicrosoft.	App Services, Application Insights, and Key Vaults
EITDev3 SQL	[REDACTED]	envisionitdev.onmicrosoft.	SQL Databases
EITDev4 Logic Apps and Misc	[REDACTED]	envisionitdev.onmicrosoft.	Logic Apps and Miscellaneous
EITDev5 SP2019	[REDACTED]	envisionitdev.onmicrosoft.	EITDevSP2019 SharePoint 2019 VM
EITDev7 SP2010	[REDACTED]	envisionitdev.onmicrosoft.	EITDevSP2010 SharePoint 2010 VM
Envision IT Dev Pay-As-You-Go	[REDACTED]	envisionitdev.onmicrosoft.	
Envision IT Developer 6 Azure	[REDACTED]	envisionitdev.onmicrosoft.	
EUMDemo Pay-As-You-Go	[REDACTED]	eumdemo.onmicrosoft.co	

# Extranet User Manager Website Project

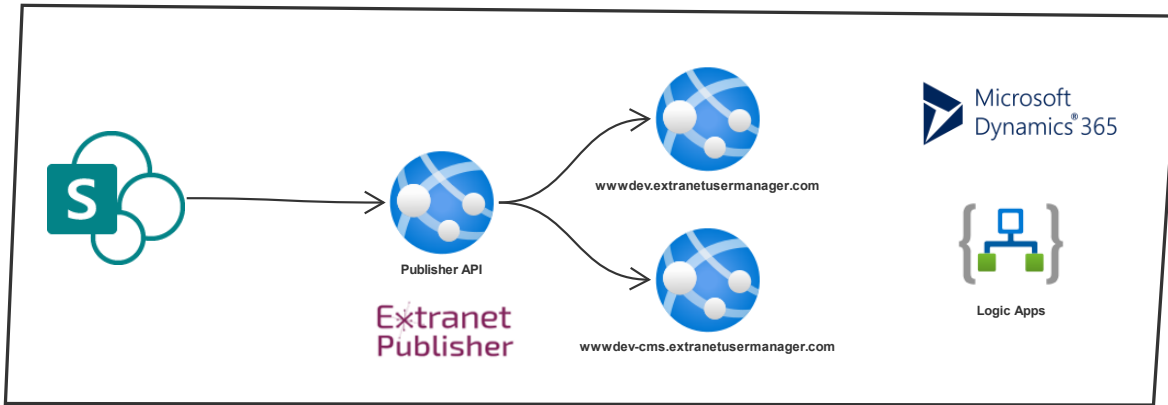
## Project Highlights:

- .NET Core 3.1 Website
- SharePoint Online Content Repository
- Extranet Publisher CMS website
- Gated Content and Webinar Custom HTML Forms and underlying Logic App workflows
- Dynamics 365 and MailChimp integrations
- DevOps Project

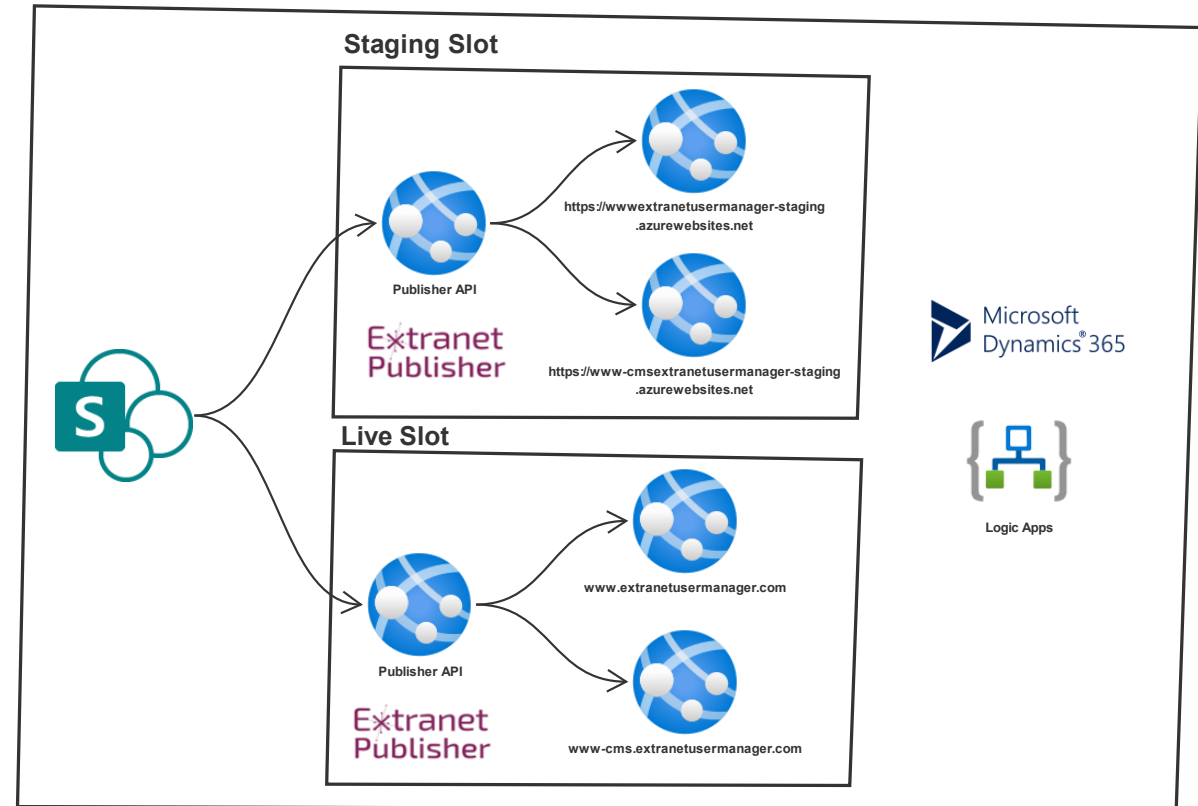


# Extranet User Manager Website Environments

## Non-Prod



## Production





# Links

- **Teams Provisioning**

- [www.envisionit.com/products/teams-provisioning](http://www.envisionit.com/products/teams-provisioning)

- **Extranet User Manager**

- [www.extranetusermanager.com](http://www.extranetusermanager.com)

- **Event Page**

- [www.extranetusermanager.com/resources/events/sector-conference-2021](http://www.extranetusermanager.com/resources/events/sector-conference-2021)

# Thank you!

## Questions?

