Secure Development with Microsoft 365 and Azure AD (Part 1 of 2)

Thursday, April 20, 2021

12 - 1 PM Eastern Time

Peter Carson

- President, Extranet User Manager
- Office Apps and Services Microsoft MVP
- peter.carson@extranetusermanager.com
- blog.petercarson.ca
- www.extranetusermanager.com
- Twitter @carsonpeter
- President Toronto SharePoint User Group

2008

Envision IT built custom Extranet solution

2009

2010

2011

Productization of code base begins

2012

Extranet User Manager (EUM) Installer created

2014

Office 365 support

2015

EUM Brand and Website launched

2016

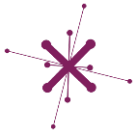
Azure B2B support

2017

Partner Program launched

2018

EUM V4 Launched



2019

User-Centric EUM Login Teams, SPFx, and Flow

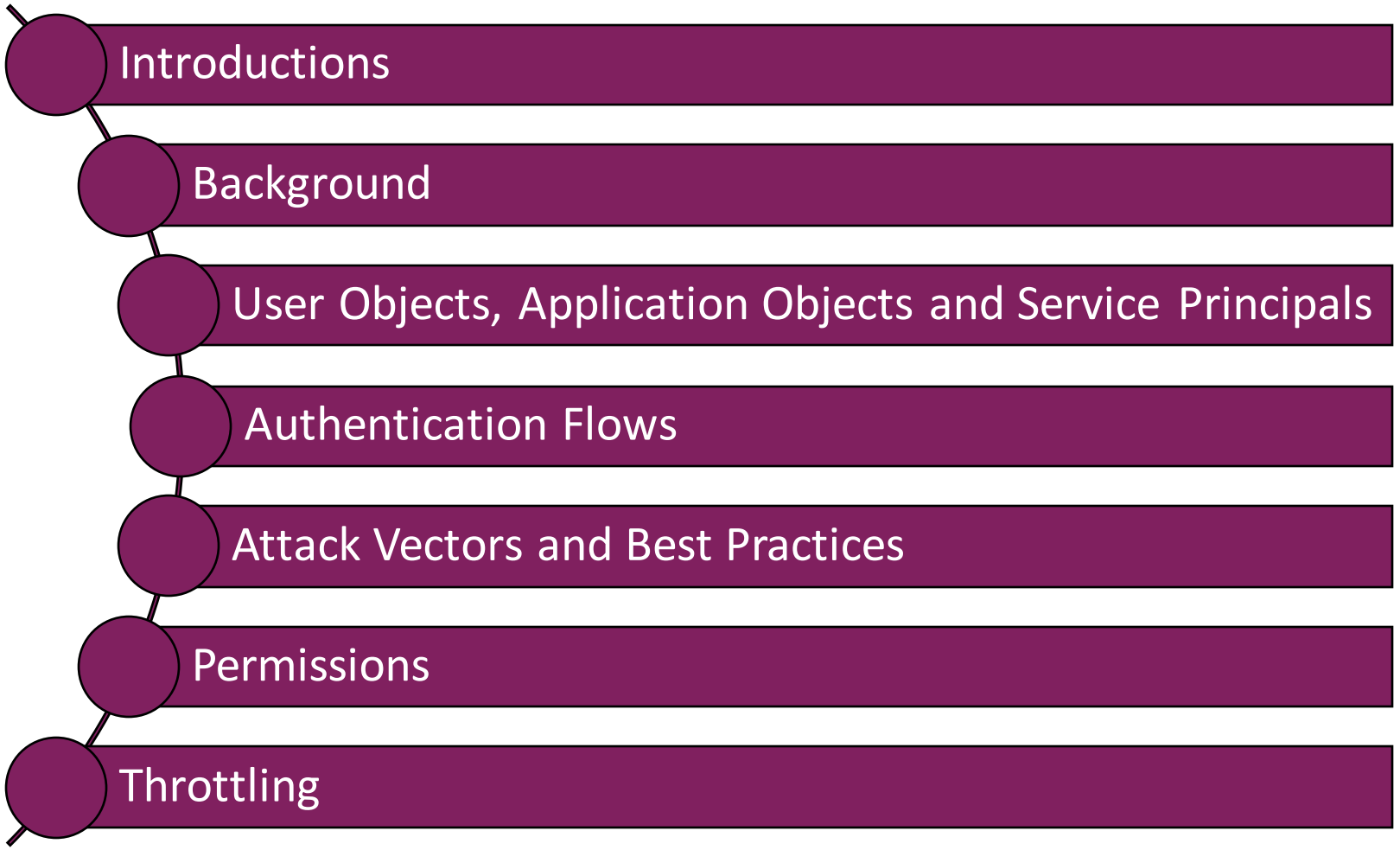
2021

EUM Products Website Relaunched

Customers around the Globe

100+ Customers Deployed Globally

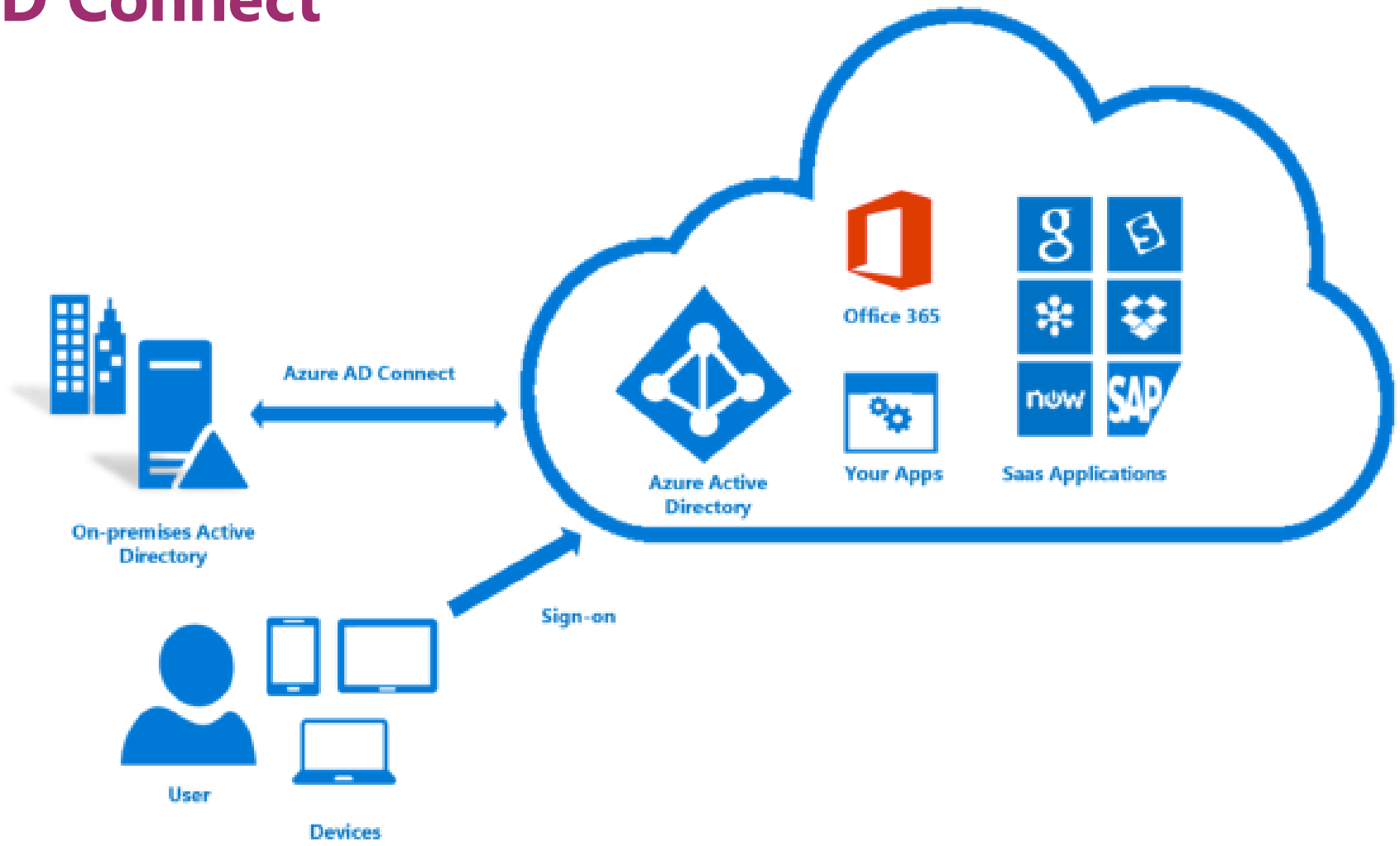
Agenda


Introductions
Background
User Objects, Application Objects and Service Principals
Authentication Flows
Attack Vectors and Best Practices
Permissions
Throttling

Microsoft Forms Poll

https://bit.ly/3x9cBpd

Azure AD Connect



https://docs.microsoft.com/en-us/azure/active-directory/hybrid/whatis-azure-ad-connect

Azure AD Connect versus Connect Cloud Sync

Connect	Connect Cloud Sync
• Runs on premise • Syncs custom AD attributes • Supports pass-through authentication • Object attribute filtering • Password, device, and group writebacks	• Runs in the cloud with a lightweight agent on premise • Can sync multiple disconnected forests

https://docs.microsoft.com/en-us/azure/active-directory/cloud-sync/what-is-cloud-sync

RSAConference2020

San Francisco | February 24 – 28 | Moscone Center

HUMAN ELEMENT

SESSION ID: IDY2-F03

Breaking Password Dependencies: Challenges in the Final Mile at Microsoft

Alex Weinert (@alex_t_weinert), Director of Identity Security, Microsoft
Lee Walker, Principal Program Manager, Microsoft IT Identity and Access

https://www.rsaconference.com/usa/agenda/breaking-password-dependencies-challenges-in-the-final-mile-at-microsoft

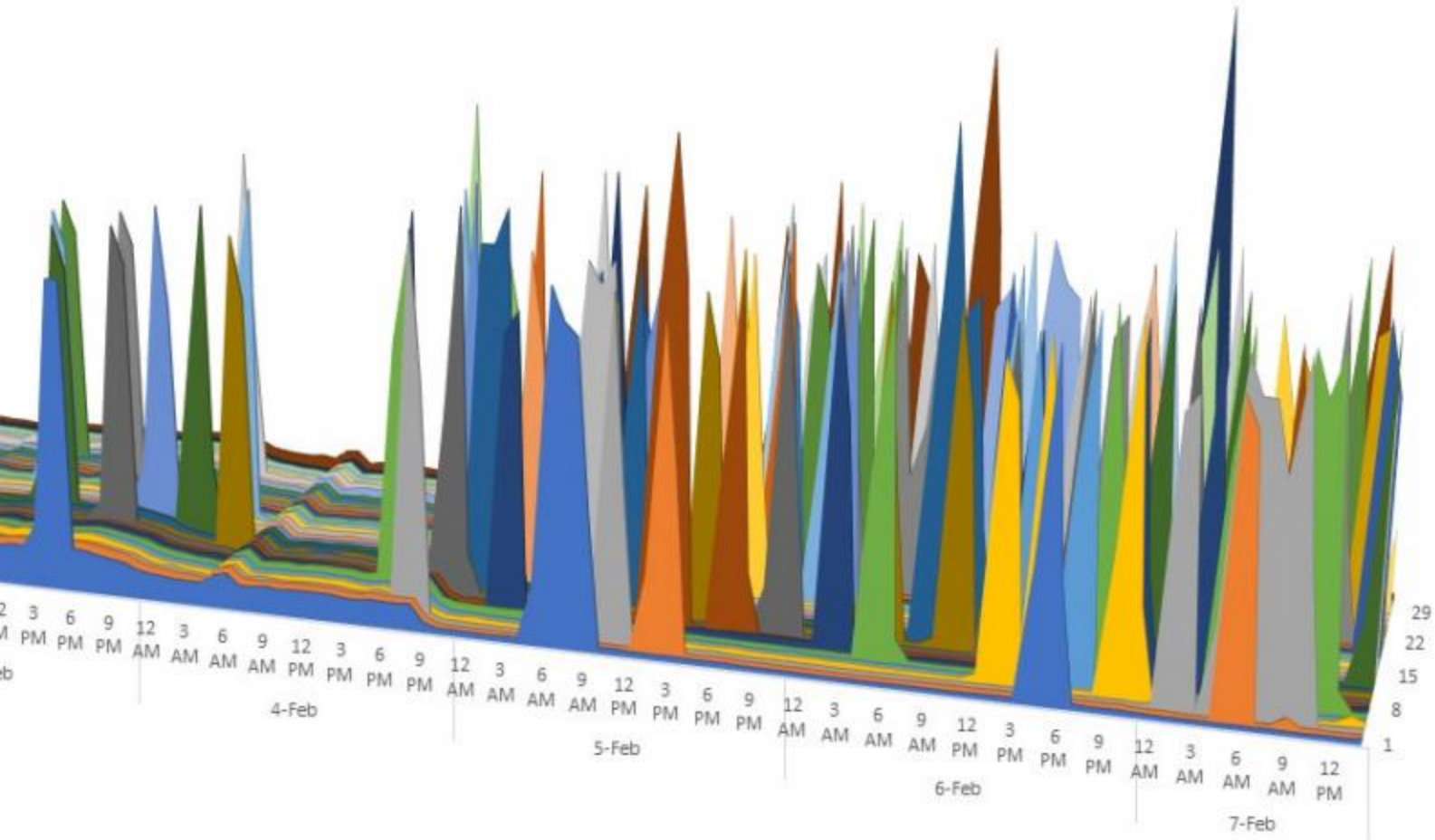
>1.2M

compromised accounts in January 2020

>99.9%

compromised accounts did not have MFA

~40% (480k accounts in January) compromised by Password Spray*



Josi@contoso.com	Spring2020!
Chance@wingtiptoys.com	Spring2020!
Rami@fabrikam.com	Spring2020!
TomH@cohowinery.com	Spring2020!
AnitaM@cohovineyard.com	Spring2020!
EitokuK@cpandl.com	Spring2020!
Ramanujan@Adatum.com	Spring2020!
Maria@Treyresearch.net	Spring2020!
LC@adverture-works.com	Spring2020!
EW@alpineskihouse.com	Spring2020!
info@blueyonderairlines.com	Spring2020!
AiliS@fourthcoffee.com	Spring2020!
M39@litwareinc.com	Spring2020!
Margie@margiestravel.com	Spring2020!
Ling-Pi997@proseware.com	Spring2020!
PabloP@fineartschool.net	Spring2020!
GiseleD@tailspintoys.com	Spring2020!
Luly@worldwideimporters.com	Spring2020!

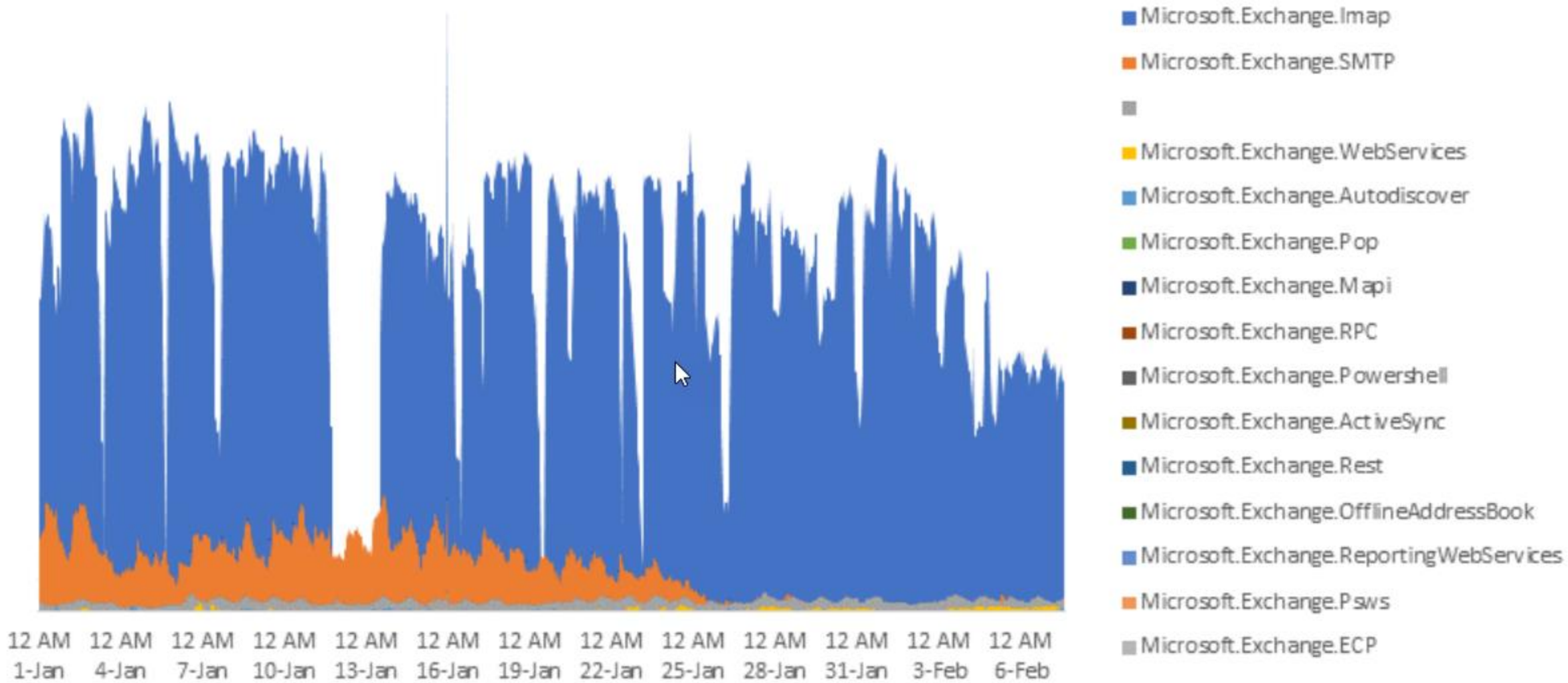
Microsoft

*Cloud detected only

>99%

of Password Spray attacks use legacy auth

~40% (480k accounts in January) compromised by replay



>97%

of Replay attacks use legacy auth

Office 365 Security Best Practices

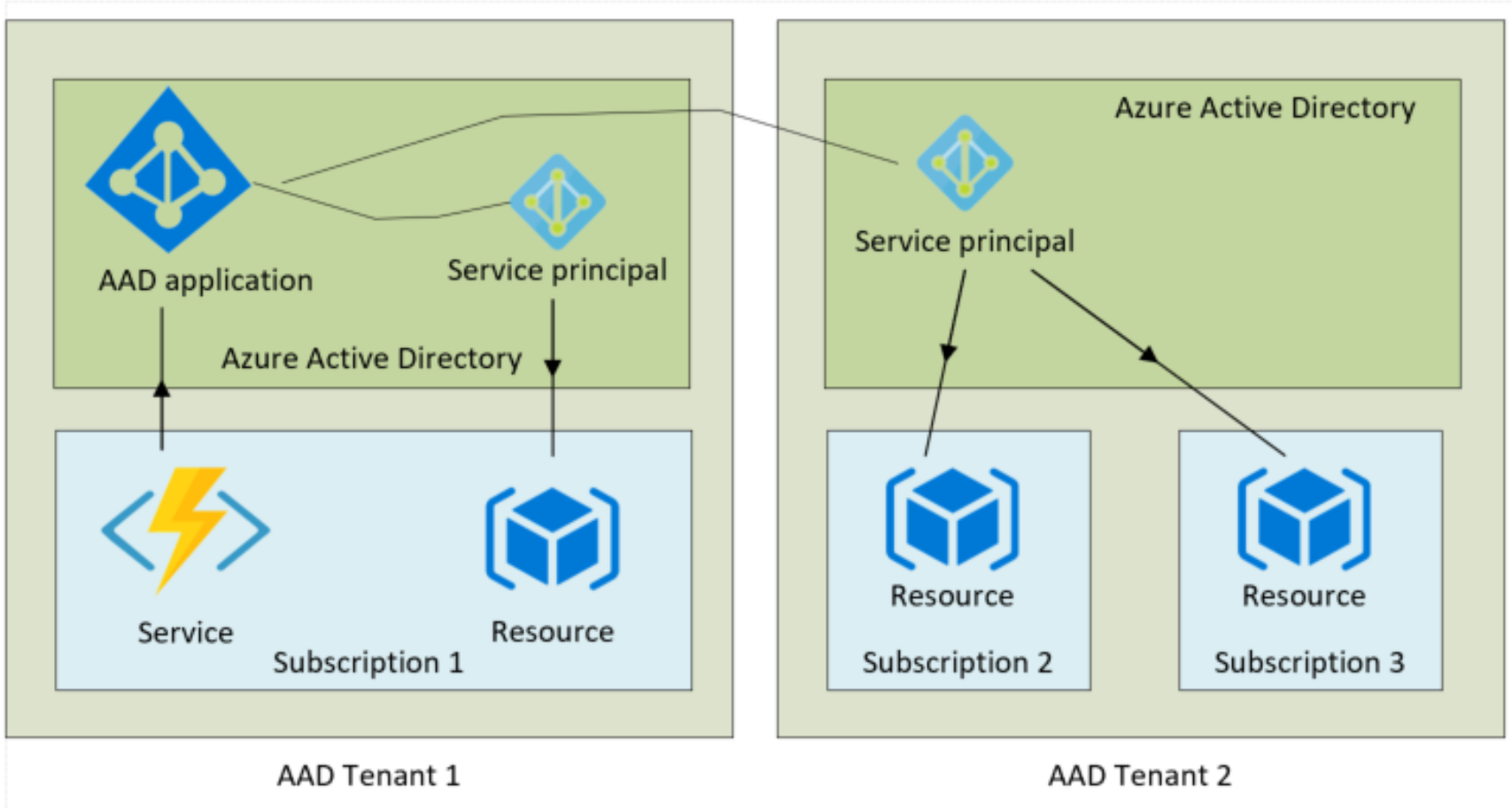
1. Set up multi-factor authentication
2. Train your users (Educate)
3. Use dedicated cloud only admin accounts
4. Raise the level of protection against malware in mail
5. Protect Against Ransomware
6. Stop auto-forwarding for email
7. Use Office Message Encryption
8. Protect your email from phishing attacks
9. Protect against malicious attachments and files with ATP Safe Attachments
10. Protect against phishing attacks with ATP Safe Links

https://docs.microsoft.com/en-us/office365/admin/security-and-compliance/secure-your-business-data?view=o365-worldwide

User Objects, Application Objects and Service Principals

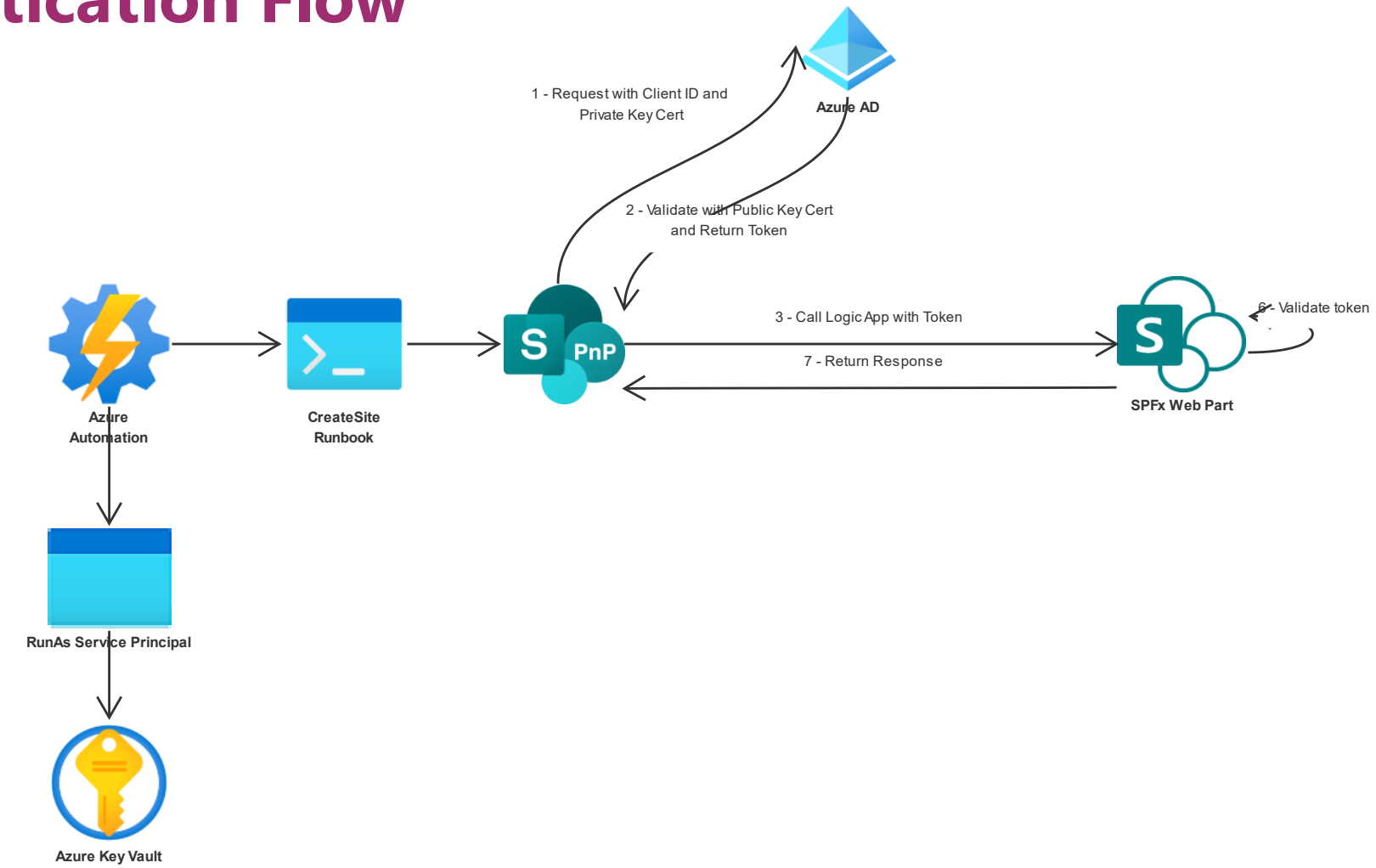
- **Common practice was often to use user credentials in integration scenarios**
  - Credentials could be stored securely in Azure Key Vault
- **MFA breaks this**
  - Daemon process or back end code can't process an MFA request
- **Disabling MFA on "service" accounts is a big security hole**
- **Service 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 Application ID – which is the Client ID
5. Go to App Registrations in Azure AD
6. Search for the client ID
7. Assign the appropriate API permissions
8. Create your runbooks, and use the service principal to authenticate

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**

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

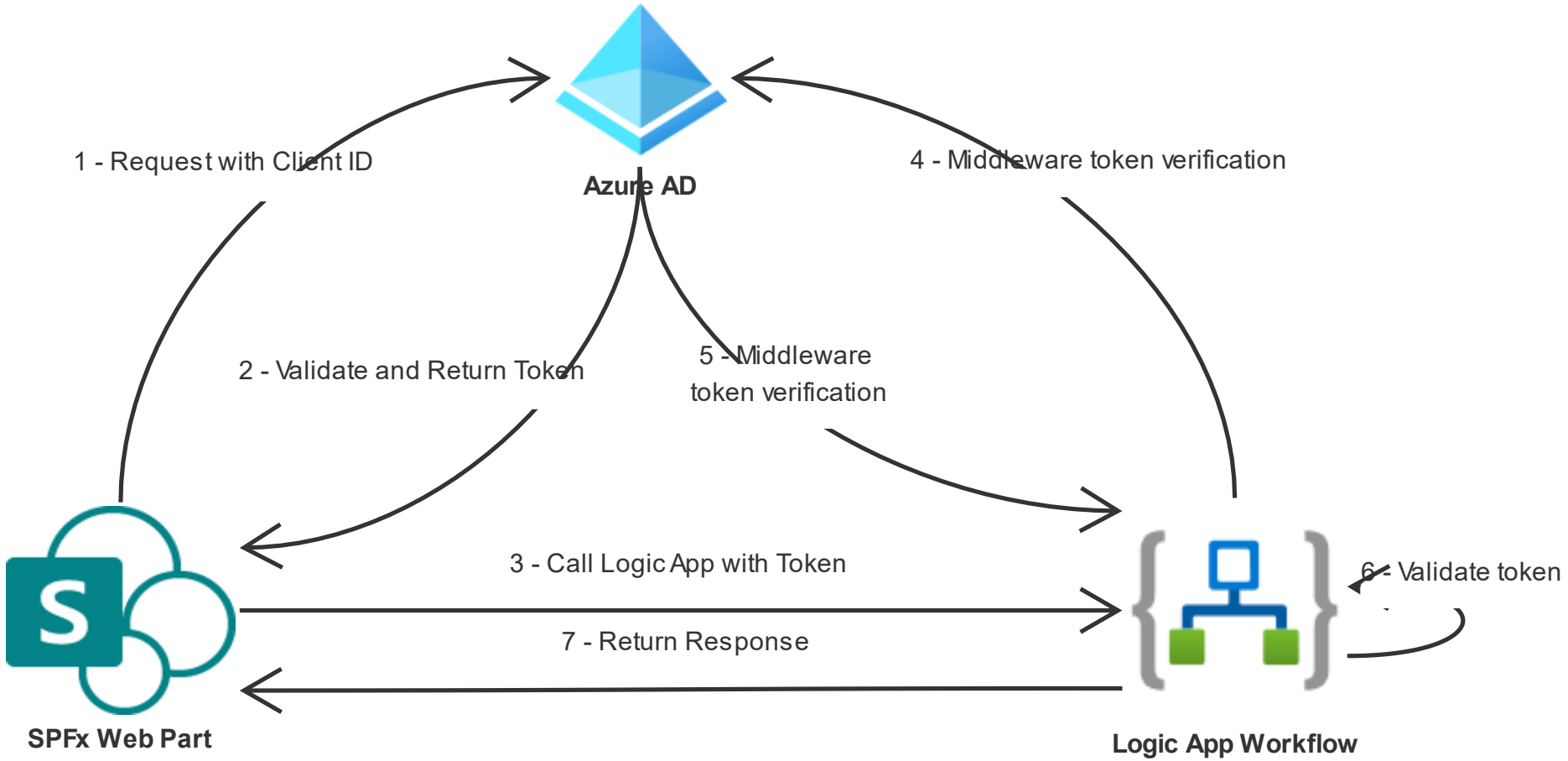
Security Through Obscurity

- **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

Using Logic Apps or Power Automate as a Secure API

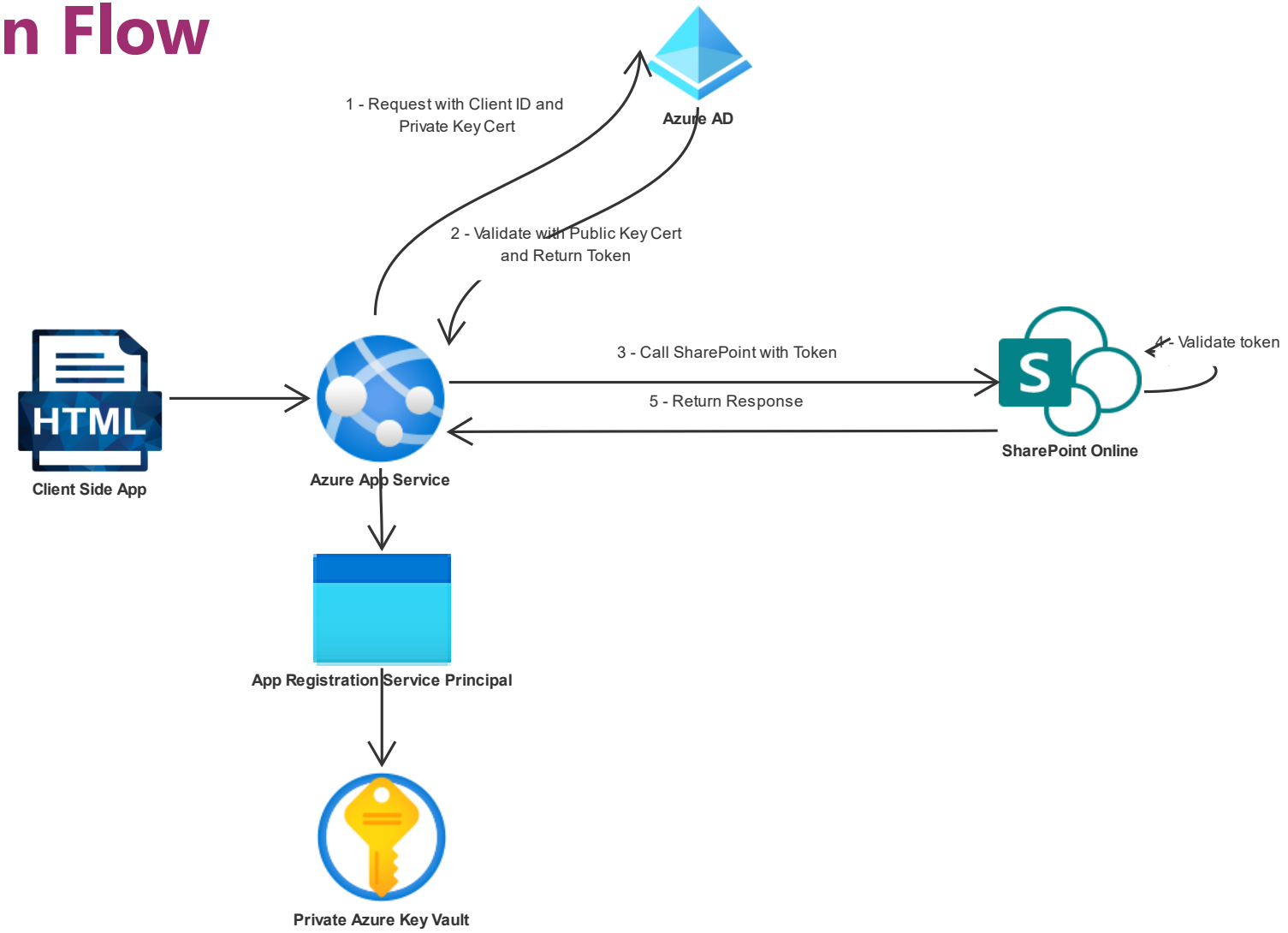
- **Trigger for the workflow is an HTTP POST REST method**
  - Could also be a GET, PUT, PATCH, or DELETE
- **Logic Apps / Flow generates a URL**
  - Includes a Shared Access Signature secret
- **Can also configure Azure AD authentication**
  - Register an app in Azure AD
  - Record the Client ID in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

OAuth2 SPFx to Logic App Authentication Flow

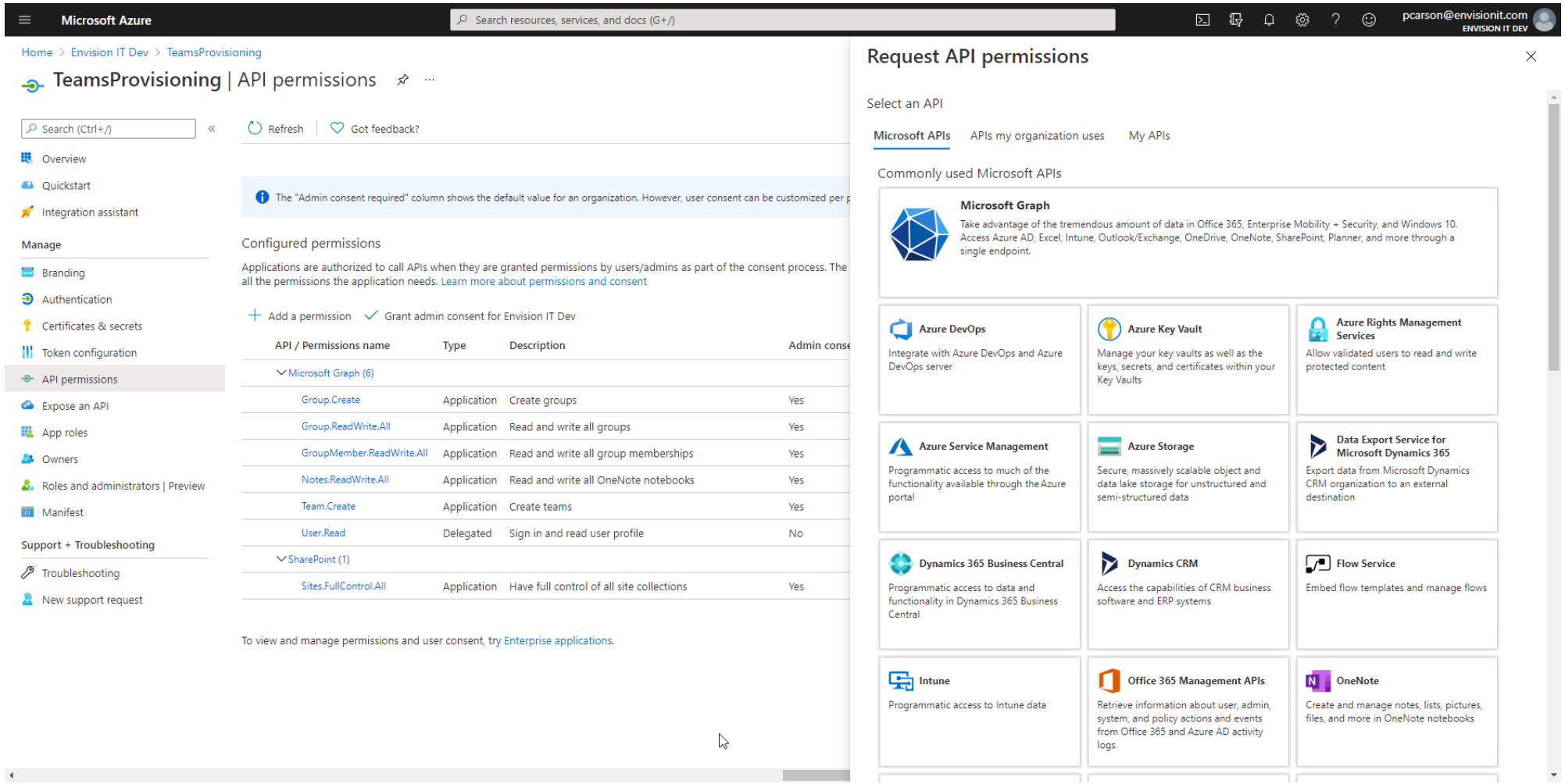


https://docs.microsoft.com/en-us/azure/active-directory/develop/v2-oauth2-auth-code-flow

OAuth2 Azure App Service to SharePoint Authentication Flow



API Permissions



API Permissions

Delegated

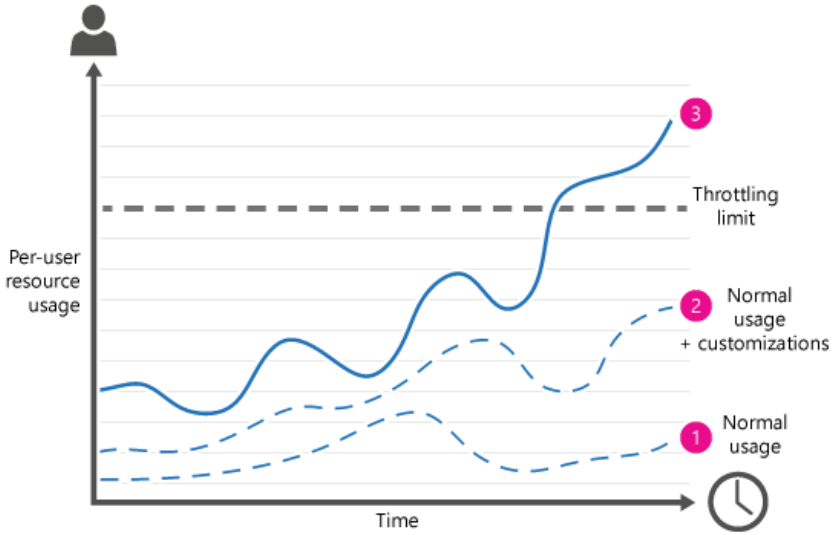
- **Need a user token to request a delegated token**
- **Typically only applicable for interactive applications**
- **Permissions are applied to the app registration**
- **Delegated user also needs appropriate permissions**
- **Some APIs only support delegated permissions**
  - Planner Tasks

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
  - Scoping by site collection is in preview

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



Recommendations

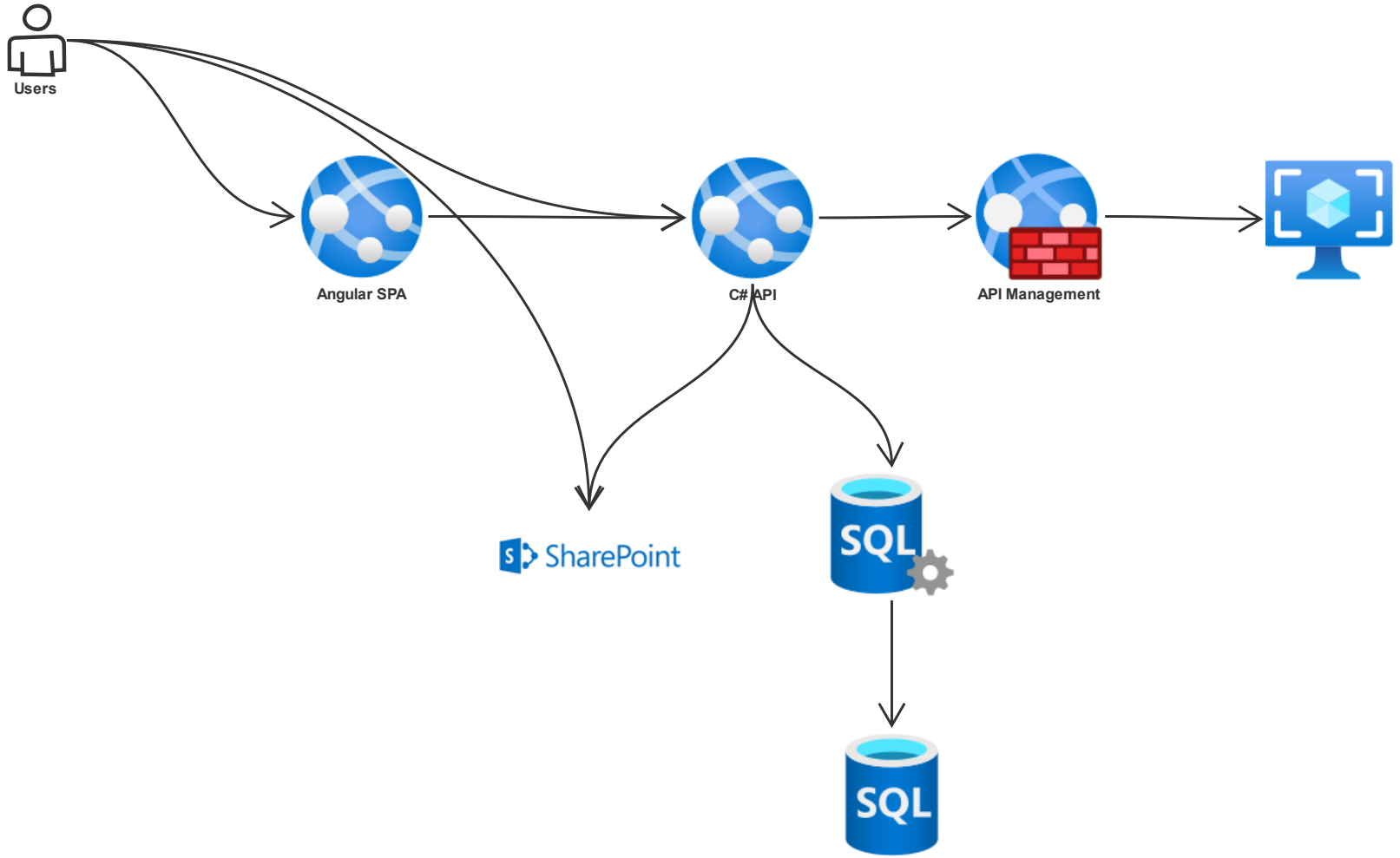
- Decorate your traffic
- Respect retry recommendations

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
  - Different threshold than user requests, but also 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



Renewals

- **Azure Automation RunAs certificates need to be renewed annually**
- **Logic App and Power Automate API connections**
  - Need to be periodically re-authorized
  - Can be a regular user account or a specific “service” account
  - Re-authorize is an interactive action, so MFA is supported

Part 2 Webinar

Microsoft 365 SDLC Best Practices (Part 2 of 2)
May 4, 2021
12 pm – 1 pm EST

Register for all upcoming events at http://eum.co/resources/events

Additional Webinars

Integrating Multiple Planner Boards into a Power BI Dashboard
May 20, 2021
12 pm – 1 pm EST

Register for all upcoming events at http://eum.co/resources/events

Thank you!

Questions?