
collabdays | lisbon
October 10th, 2020
Mastering Modern Authentication and Authorization Techniques for SharePoint, Office 365, and Azure
by Eric Shupps, MVP

Our Sponsors

DIAMOND

COMMUNITY

About Me

Eric Shupps

Office Servers & Services MVP

- @eshupps
- sharepointcowboy
- slideshare.net/eshupps
- linkedin.com/in/eshupps
- github.com/eshupps

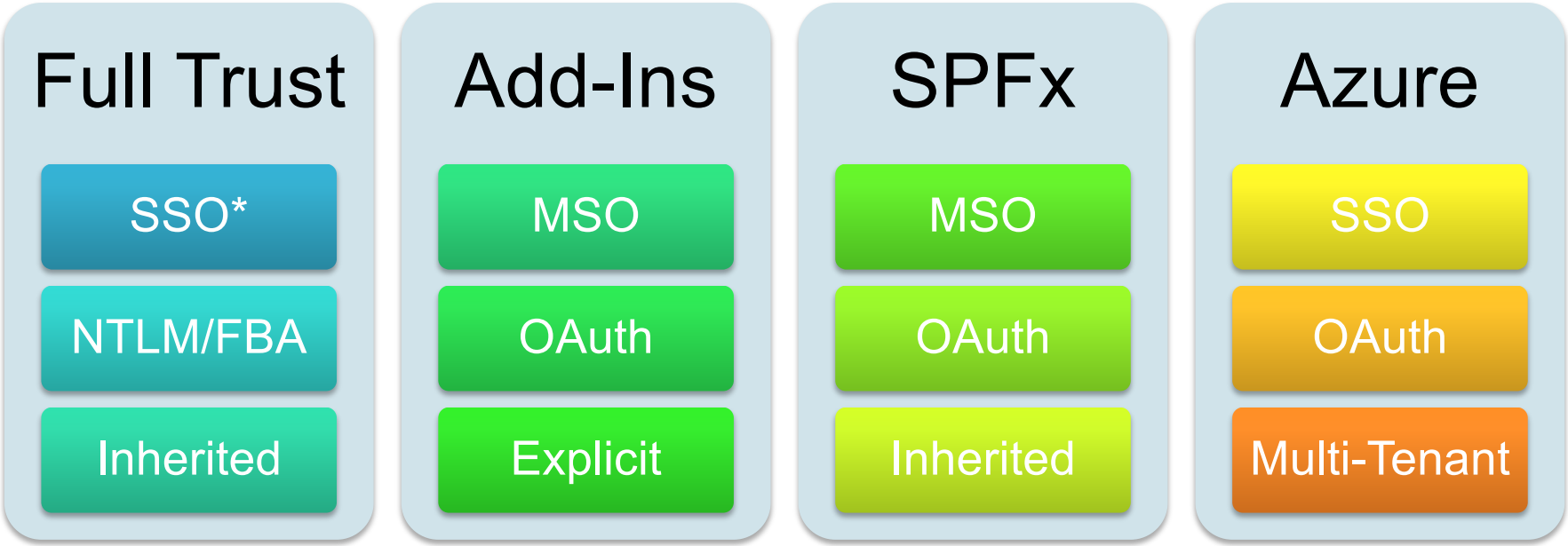
collabdays

Agenda

- Introduction
- Authentication
- Authorization
- Permissions

Introduction

Development Models



Challenges

- User authentication on-premises and in the cloud
- Managing identities across different application types
- Obtaining, storing and processing authorization tokens
- Traversing applications within a single tenant context
- Deploying applications to multiple tenants
- Supporting mobile devices and apps

Authentication

Add-Ins

- Wide range of supported authentication types
  - On-Premises: NTLM, FBA, Other
  - Cloud: FBA, Azure AD, Other
  - (NOTE: Anonymous with app-only permissions)
- Distributed identity management
- Instance-based consent framework
- Multiple contexts
  - Explicit (PHA – JSOM, REST, Graph)
  - Implicit (SHA - JSOM)

SPFx

- Inherits authentication type from parent
- Centralized identity management (out of scope)
- Instance-based consent framework
- Inherited context
- Limited native connectivity (GraphHttpClient, SPHttpClient, AadHttpClient)

Azure AD

- Single sign on across all cloud applications
  - Singe tenant
  - Multi-tenant
- Centralized API-accessible identity management
- User or tenant consent framework (with optional group assignment)
- Explicit context per endpoint
  - SharePoint requires certificate exchange

Single Sign-On in an Azure AD Web Application

DEMO

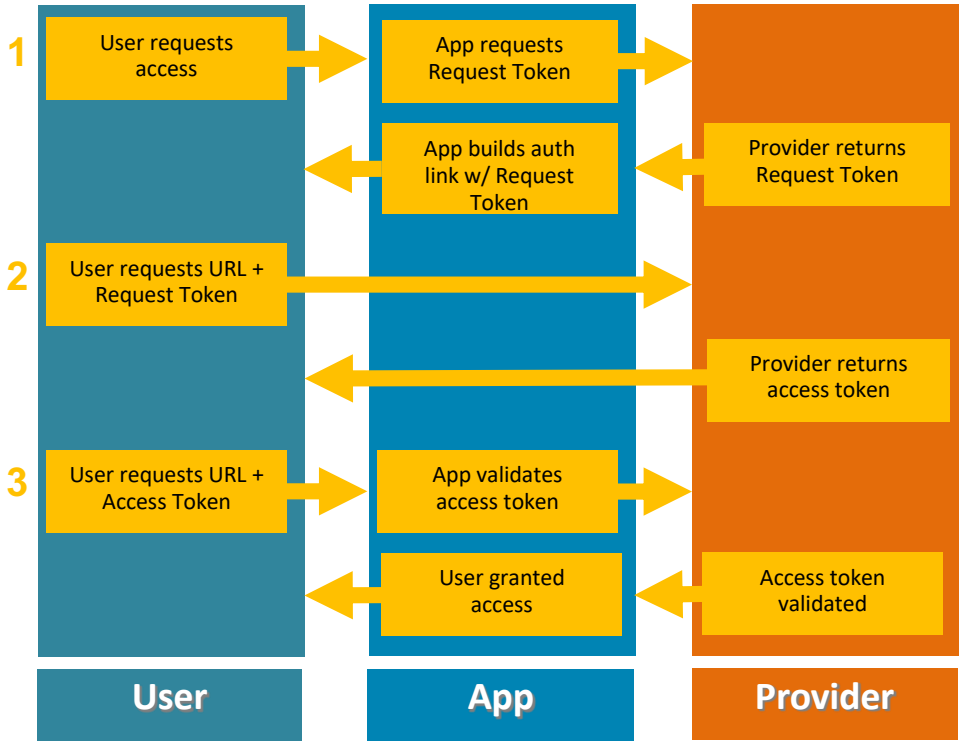
Authorization

OAuth

- Open standard for app integration and authorization
- Authentication independent
- Token-based scheme for enforcement of explicit trust relationships
- Permission scopes (grants) define access boundaries
- Explicit or inherited authorization context

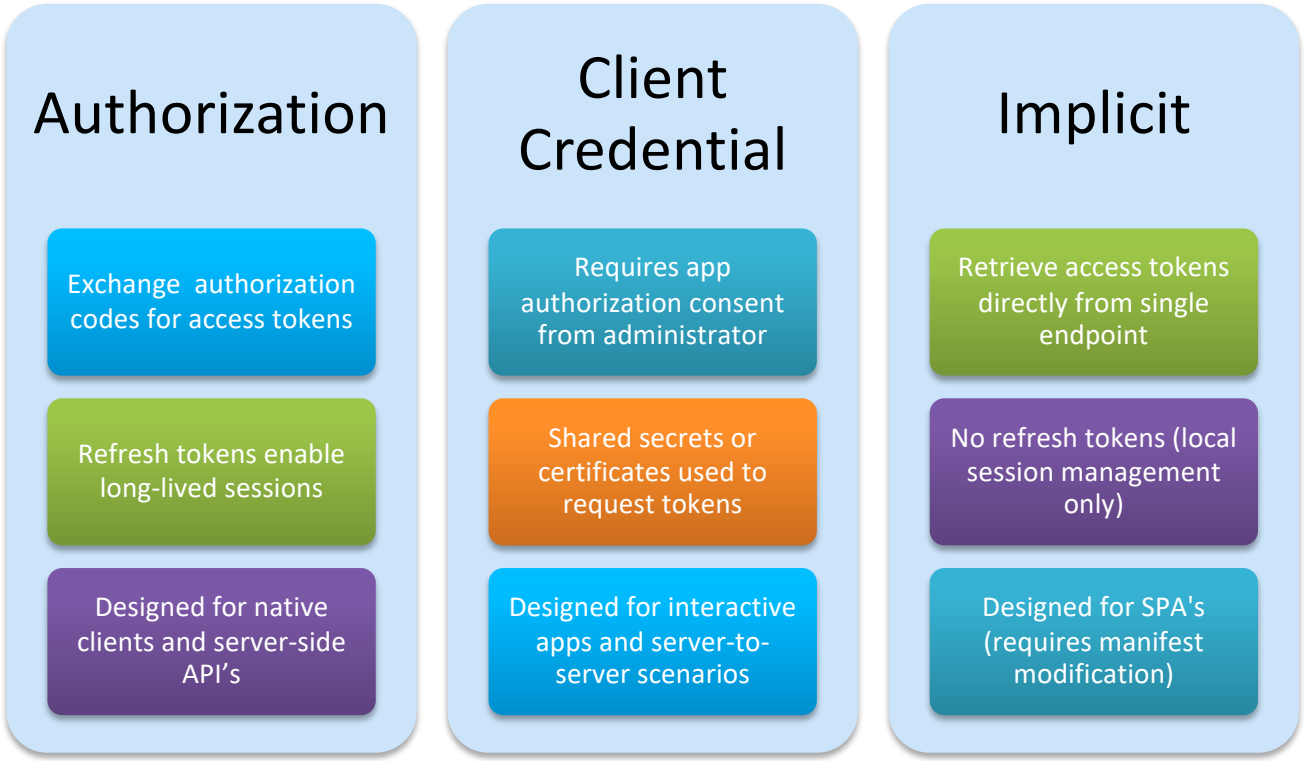
OAuth Flow

- Authorization Endpoint
- Application Registration
- User/Admin Consent
- Authorization Code
- Access Token
- Refresh Token



Reference: https://oauth.net/2/

Authorization Flows



Token Management

- Use authorization/request tokens to obtain short-lived access tokens
- Include access tokens in resource calls
- Store refresh tokens to obtain new access tokens upon expiration
- Track tokens by tenant (multi-tenant), app or user
- Force token expiration to prompt authentication
- Utilize client secret only in confidential client apps

Token Configuration

Property	Policy String	Affects	Default	Minimum	Maximum
Access Token Lifetime	AccessTokenLifetime	Access tokens, ID tokens, SAML 2 tokens	1 hour	10 minutes	1 day
Refresh Token Max Inactive Time	MaxInactiveTime	Refresh tokens	90 days	10 minutes	90 days
Single-Factor Refresh Token Max Age	MaxAgeSingleFactor	Refresh tokens (for any users)	Until revoked	10 minutes	Until revoked
Multi-Factor Refresh Token Max Age	MaxAgeMultiFactor	Refresh tokens (for any users)	Until revoked	10 minutes	Until revoked
Single-Factor Session Token Max Age	MaxAgeSessionSingleFactor	Session tokens (persistent and non-persistent)	Until revoked	10 minutes	Until revoked
Multi-Factor Session Token Max Age	MaxAgeSessionMultiFactor	Session tokens (persistent and non-persistent)	Until revoked	10 minutes	Until revoked

NOTE: Configurable tokens not supported in SPO. Access = 1 hour, Refresh = 90 days

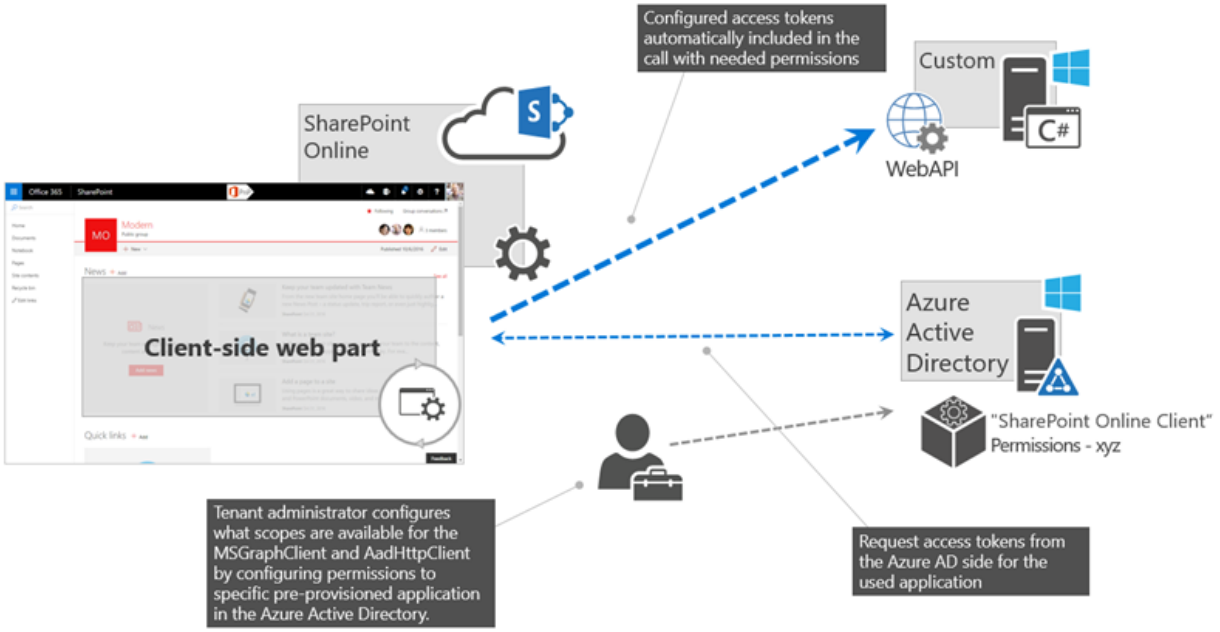
Reference: http://bit.ly/2IUuJNo

Client Credential Authorization Flow in Azure AD Web Application with Microsoft Graph

DEMO

SPFx Authorization

- ADAL.js* or msal.js AadHttpClient & MSGraphClient
- Manage permissions in Azure portal via SharePoint Online Client Extensibility app
- Include scope requests in package-solution.json
- Tenant-wide admin consent
- Manageable via O365 admin center
  - SharePoint > Advanced > API Access
- Permissions do not affect app provisioning or deployment
- EACH PAGE IS AN APP!

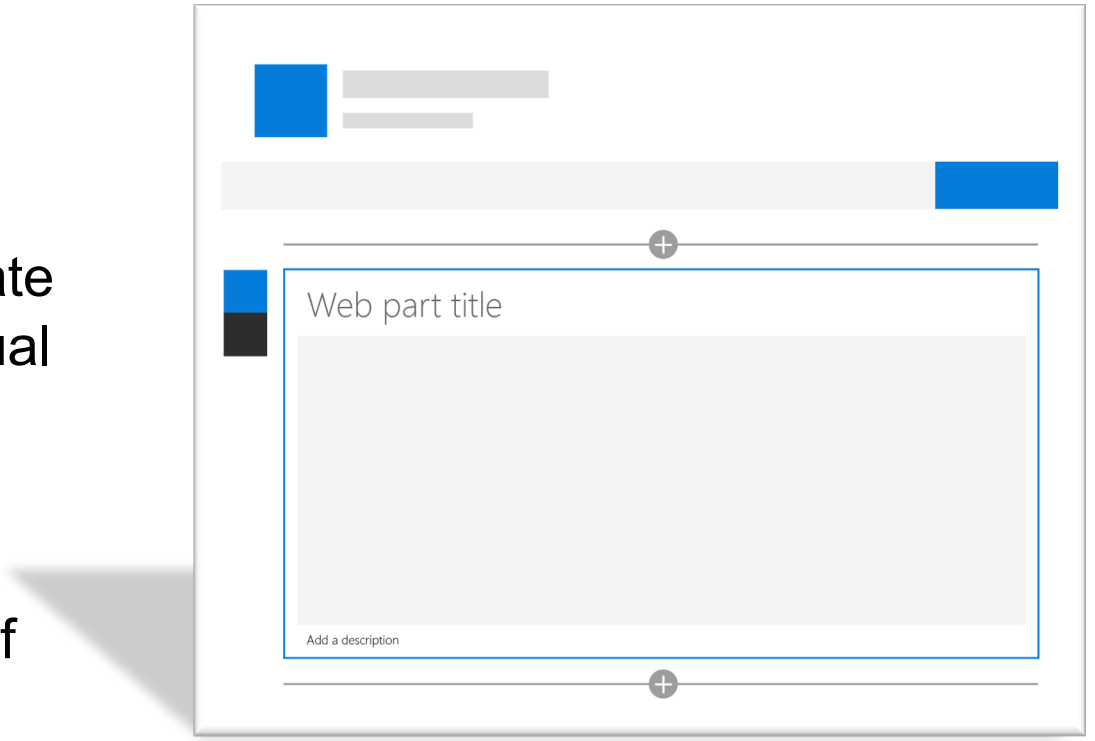


* - Version 1.4.0 or lower. See https://docs.microsoft.com/en-us/sharepoint/dev/spfx/use-aadhttpclient for more info.

Reference: http://bit.ly/2IUcqId

Web Parts

- ADAL.js implements a single configuration instance on the page
- Patch required to isolate auth tokens to individual web parts
- Specify web part ID in configuration
- Page URL's are part of trusted contract with Azure AD



collabdays

Mobile Applications

- Register app in Azure
- Configure redirect URI
- Configure manifest with oauth2AllowImplicitFlow = true
- Assign permission grant
- Use ADAL.js for authorization
- Leverage the Authenticator for SSO across applications

Reference: http://bit.ly/2rTEzW1

Calling an AAD-Secured API from SPFx

Managing App Permissions in Azure & Office 365

DEMO

Permissions

Add-In Permission Scopes

Scope URI	Available Rights
http://sharepoint/content/tenant	Read, Write, Manage, FullControl
http://sharepoint/content/sitecollection	Read, Write, Manage, FullControl
http://sharepoint/content/sitecollection/web	Read, Write, Manage, FullControl
http://sharepoint/content/sitecollection/web/list	Read, Write, Manage, FullControl
http://sharepoint/bcs/connection	Read
http://sharepoint/search	QueryAsUserIgnoreAppPrincipal
http://sharepoint/social/tenant	Read, Write, Manage, FullControl
http://sharepoint/social/core	Read, Write, Manage, FullControl
http://sharepoint/social/microfeed	Read, Write, Manage, FullControl
http://sharepoint/taxonomy	Read,Write

Reference: http://bit.ly/2wW4lOr

SPFx Permissions

- Graph (v2.0)
  - Specify resource and scope in the package-solution.json file (dynamic consent)
- Azure AD
  - Permission requests must be approved by tenant admin
  - Manage permission scopes via Azure portal or O365 Admin Center
  - Grants are TENANT SCOPED and apply to all SPFx solutions
  - Removing a solution DOES NOT revoke permission grant(s)
  - All grants associate with **SharePoint Online Client Extensibility** service principal

JSON

```
{
 "$schema": "https://dev.office.com/json-schemas/spfx-build/package-solution.schema.json",
 "solution": {
   "name": "spfx-graph-client-side-solution",
   "id": "5d16587c-5e87-44d7-b658-1148988f212a",
   "version": "1.0.0.0",
   "includeClientSideAssets": true,
   "skipFeatureDeployment": true,
   "webApiPermissionRequests": [
     {
       "resource": "Microsoft Graph",
       "scope": "Calendars.Read"
     },
     {
       "resource": "Microsoft Graph",
       "scope": "User.ReadBasic.All"
     }
   ]
 },
 "paths": {
   "zippedPackage": "solution/spfx-graph.sppkg"
 }
}
```

```
PS C:\> Enable-SPOTenantServicePrincipal
```

Reference: http://bit.ly/2IUcqId

collabdays

Azure AD Permission Grants

- Types
  - Application
  - Delegated
- Administrative Level
  - Minimum: “Sign in and read user profile”
  - Beware permission level restrictions for consent
- Resources

• Exchange	Yammer	Azure AD
• SharePoint Online	Power BI	Azure Management
• O365 Management	Skype	

Reference: http://bit.ly/2rV8FII

Consent Framework

- User must agree to accept defined permissions
  - Consent applies only for that specific user
  - Prompt identifies permission scopes
  - Avoid unnecessary scopes
  - Blocked from consent if admin scopes present
- Admin can agree on behalf of entire tenant
  - Prevents user prompts
- Scope changes require new consent

Dynamic Consent

- Azure AD v2 endpoints
- Permission scopes defined in application manifest
- User provides consent “on the fly”
  - Requires user interaction
- Not compatible with implicit flow
- Beware admin consent scopes

Reference: http://bit.ly/2XuEhUs

Specifying Permissions for Azure AD Web Applications

DEMO

THANK YOU!

https://www.slideshare.net/eshupps

https://www.github.com/eshupps

Our Sponsors

DIAMOND

COMMUNITY


collabdays | lisbon