#CollabDaysBE

Become a Microsoft Graph Ninja

Paolo Pialorsi | @PaoloPia

CollabDays Belgium 2022 by BIWUG

Thanks to our sponsors!

Platinum

Kianda

Microsoft

Gold

acolad. DIGITAL

AvePoint

inetum realdolmen Positive digital flow

ORDINA

pointfire Multilingual SharePoint

SysKit

Quest

spikes

The Flow

Xylos

Silver

advantive innovative people valuable solutions

CTG

Docubird™

European SharePoint Office 365 & Azure Conference

SharePint

ShareGate: by GSoft

Community

EUROPEAN COLLABORATION SUMMIT

Organized by

BIWUG

ABOUT ME

- Solution Architect, Consultant, Trainer
  - PiaSys.com based in the USA and in Italy
- More than 50 Microsoft certification exams passed
  - MCSM – Charter SharePoint
  - MVP M365 Development + M365 Apps & Services
  - Microsoft 365 PnP Team Member
- Focused on SharePoint, Teams and Microsoft 365
- Author of many books about XML, SOAP, .NET, LINQ, SharePoint, and Microsoft 365
- Speaker at main IT conferences worldwide
- Follow me @PaoloPia
- Subscribe to: https://youtube.com/@PiaSysTechBites

Microsoft CERTIFIED Master

MVP Microsoft® Most Valuable Professional

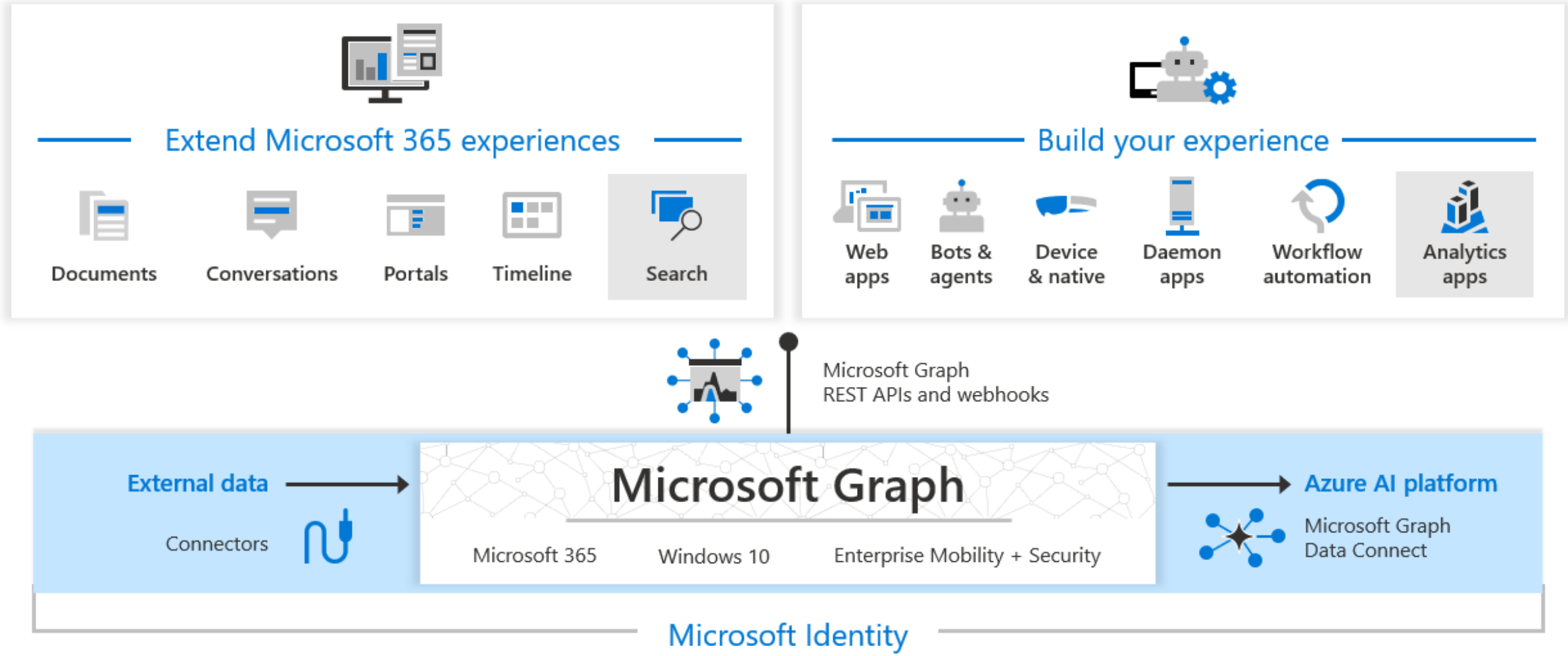
PnP

Microsoft Graph API

What is the Microsoft Graph API?

- A unique set of API to rule all the Microsoft services
- Unified endpoint
- Unique handshake and Access Token
- Unique development experience
- Unique consumption model: REST

Microsoft 365 Platform



Cross-platform technology

- You can consume the Microsoft Graph from:
  - Android
  - Angular
  - ASP.NET
  - Go
  - iOS
  - Node.js
  - PHP
  - Python
  - Ruby
  - REST
  - UWP
  - Xamarin
  - JavaScript
  - etc.
- You just need an HTTP client and support for OAuth 2.0

App registration and consumption model

Understanding the Microsoft Identity platform

Consumption Model

- Register an application in Azure AD
  - Choose the permission model and the token flow
- Make an OAuth 2.0 handshake with Azure AD
  - Get Access Token (and eventually Refresh Token)
- Invoke Microsoft Graph providing the Access Token
  - Authorization: Bearer …

Security Tokens

- ID Tokens
  - Are tokens for users' identification purposes
  - Allow the client to verify that user is who she/he claimed to be
  - Are JWT objects made of claims
  - By default they last 1 hour
- Access Tokens
  - Enable clients to securely consume Web APIs
  - Web APIs rely on access tokens to perform AuthN and AuthZ
  - Are JWT objects made of claims
  - They have a variable lifetime (60-90 mins)
- Refresh Tokens
  - When provided, are used to obtain a new access/refresh token when the current one expires
  - Their content is opaque
  - Lifetime is 24 hours for SPA and 90 days for all other scenarios
  - Can be revoked by the server because of change of credentials, user action, or admin action

OAuth 2.0 Token Flows (1/2)

- Auth code grant
  - Leverages browser-based authentication and redirection
  - Provides access token and refresh token, eventually id token, too
  - Used by web apps, SPA, desktop, mobile apps, etc.
- Client credentials grant
  - Authenticates application using ClientId and ClientSecret/Certificate
  - Used for daemons, server-to-server communication, etc.
- Device code grant
  - Requests user to browse to a web page, provide a code, and sign-in
  - Provides access token and refresh token
  - Used to authenticate on devices like Smart TV, IoT, etc.

OAuth 2.0 Token Flows (2/2)

- On-behalf-of grant
  - A user invokes a back-end Web API/service providing an access token
  - The back-end Web API/service needs to consume another back-end API (like Microsoft Graph) on behalf of (OBO) the front-end user
  - The back-end Web API/service needs client credentials to request the OBO token
- Implicit code grant
  - Allows retrieval of tokens with direct/implicit request to the authorization endpoint
  - Returns access token or id token, only
  - Does not return the refresh token
  - Try to stay away from this flow, sooner or later it will be deprecated
- Resource owner password credentials
  - Don’t use it, unless it is really needed and unless you really trust all the actors in your architecture
  - Sends username and password to the token endpoint
  - Provides access token and refresh token

Permissions Model

- Application
  - You consume Microsoft Graph with an application-only identity
  - Ideal for daemons, background processes, automated tasks
  - Permissions are those granted to the application
- Delegated
  - You consume Microsoft Graph with user and application identity
  - The user delegates the app to work under her/his own identity
  - Permissions are the intersection of those granted to user and application

Permission Consent

- Needed to grant resources access to the Azure AD Application
- Depending on the permissions it could be
  - Admin's consent
  - User's consent
- At the very first access to the application, you're automatically prompted
  - If you're using a regular user account
    - You can only grant user's consent
    - Or ask an admin to grant tenant-wide consent
  - If you're using an admin account
    - You can choose to consent permission tenant-wide (or still for your user only)
- Manual consent can be done by browsing to the following URL
  - https://login.microsoftonline.com/{tenant-id}/adminconsent?client_id={client-id}&redirect_uri={return_URL}
    - Or by using the "Grant Permissions" button in Azure AD management UI
  - Specific permissions for tenant-wide consent are required
    - Global Administrator, Privileged Role Administrator, Cloud Application Administrator, Application Administrator, or custom role

DEMO

Registering a Microsoft Graph consumer application

Consuming Microsoft Graph with Microsoft Graph SDK + MSAL

Throttling with Microsoft Graph

Understanding throttling

- When throttling threshold is exceeded
  - Response is 429 (Too many requests)
  - Request fails
  - Suggested wait time is provided in the response headers
- Best practices to avoid throttling
  - Reduce the number of operations per request
  - Reduce the frequency of calls
  - Avoid immediate retries, because all requests accrue against your usage limits
- Best practices to handle throttling
  - Trap 429 response code
  - Look for Retry-After (secs) response header and take it into account, if any
  - Use the Microsoft Graph SDK to mitigate throttling
  - Leverage batching of requests

Willing to learn more about throttling limits?

- https://learn.microsoft.com/en-us/graph/throttling-limits

Batching Microsoft Graph requests

Microsoft Graph batching requests

- Supported in v1.0 of Microsoft Graph
- Allows combining multiple requests into a single JSON request
  - Uses JSON batching
- Dedicated endpoint (POST): https://graph.microsoft.com/v1.0/$batch
- Correlation id in the requests to associate responses and to define dependencies (*dependsOn*)
  - If parent request fails, dependent requests fail too (424 – Failed Dependency)
- Supports images (Base64 encoded)
- Batching is also useful to bypass URL length limitations

DEMO

Batching with the Microsoft Graph SDK

Know Issues/Limitations

- No nested batch
- All individual requests must be synchronous
- No transactions
- Use relative URIs for batch items
  - So, you cannot mix v1.0 and beta endpoints
- Up to 20 requests limit
- Dependencies just on a single other request (Parallel, Serial, Same)
  - Fully sequential or fully parallel

Microsoft Graph Change Notifications

Microsoft Graph change notifications

- Leverages a webhook mechanism to deliver change notifications
- Based on a subscription model
  - Create a subscription for a resource
  - Receive notifications
  - Renew subscription before it expires (for up to 3 more days)
  - Keep on receiving notifications
- Supported change notifications
  - created
  - updated
  - deleted
- You can provide a custom *clientState* (up to 128 chars) while subscribing
- Some notifications can *includeResourceData* which will be encrypted

Required permissions

- Generally speaking, requires *Read* permission on the target resource
- Supported account types:
  - Delegated - Work or school account
  - Delegated - Personal Account (depends on the resource)
  - Application

Supported Resources

- Security alert
- Teams callRecord
- Teams channel
- Teams chat
- Teams chatMessage
- Teams conversationMember
- Teams team
- Group conversation
- OneDrive driveItem
- SharePoint list
- Outlook message, event, contact
- Directory user, group, others
- Presence
- Print printer
- Print printTaskDefinition
- ToDo task

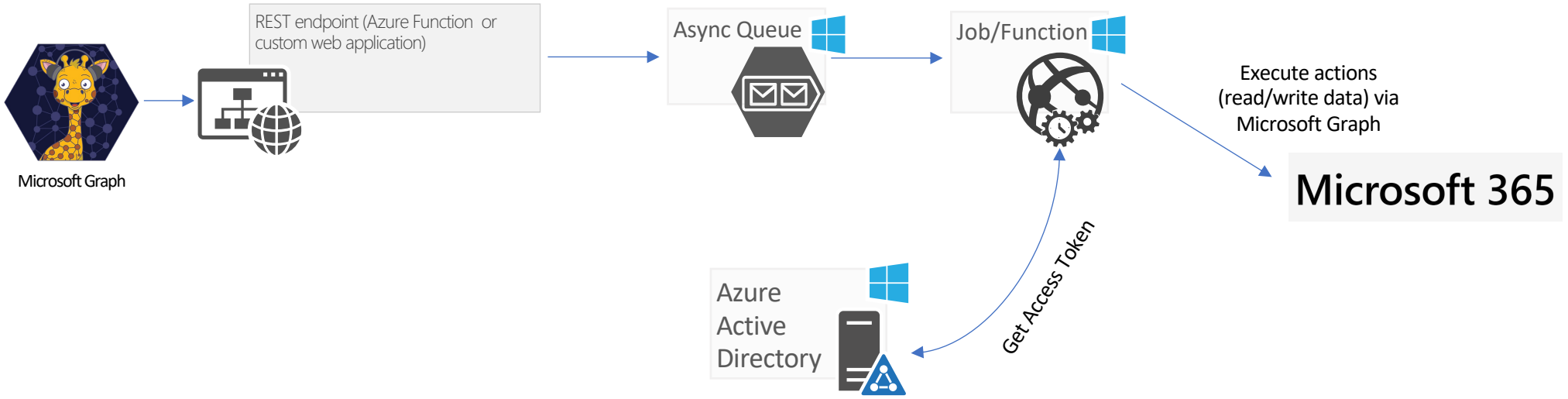
Notification endpoint validation

- Microsoft Graph sends a POST request to the notification URL
  - Including a *validationToken* in the querystring
- The notification endpoint replies within 10 seconds with:
  - 200 (OK)
  - Content-type: text/plain
  - The *validationToken* in the body of the response
- A subscription renewal doesn’t imply a validation process

Processing notifications

- Microsoft Graph sends a POST request to the notification endpoint with:
  - subscriptionId
  - subscriptionExpirationDateTime
  - clientState
  - changeType
  - resource
  - resourceData
- The notification endpoint replies within 30 seconds with:
  - 202 (Accepted)
- The notification endpoint must validate the *clientState* value

Suggested architecture



DEMO

Playing with the Microsoft Graph Change Notifications

Technical Limits

- For Azure AD resources (user, groups) there are some limits:
  - Per app: 50.000 total subscriptions
  - Per tenant: 1.000 total subscriptions across all apps
  - Per app and tenant combination: 100 total subscriptions
  - When limits exceeded:
    - 403 Forbidden + error message
- For Outlook resources (messages, events, or contacts)
  - UPN does not support apostrophes, use user ID instead
- Azure AD B2C tenants are not supported, either
- Some notifications do not provide resource details (OneDrive for example)
  - Use other techniques to retrieve actual data (for example /delta for OneDrive)

Microsoft Graph extensibility model

Graph Extensibility Options

- Extension attributes
  - For users and devices there are up to 15 extension attributes
  - Are just string values stored on the target object
- Directory extensions
  - Strongly-typed, discoverable, and filterable extensions
  - Available for directory objects: User, Group, AdministrativeUnit, Application, Device, Organization
- Open extensions
  - Un-typed, not discoverable, not filterable extensions
  - Easy to use and available for many more resources, rather than just directory objects
- Schema extensions
  - Conceptually similar to Directory extensions
  - Are world-wide defined and a good option for ISVs

Microsoft Graph Open Extensions

- You can add **untyped** properties to resources in Graph
  - Using a key/value pair approach
- Just some of the resources support Open Extensions
- Name of extensions should include reverse DNS name
  - For a TLD that you own, indeed
  - For example: *com.contoso.contactExtension*
- **Can be added, updated, deleted**
- **Resources cannot be filtered based on open extensions!**
- Supported methods: POST, GET, UPDATE, DELETE
- Required permission can vary based on the resource

Microsoft Graph Schema Extensions

- You can add **strongly-typed** custom data to resources in Graph
- Just some of the resources support Schema Extensions
- They are “schema based”
  - First you define the schema
  - Then you use it to enrich instances of resources
- Schema Extensions have a reference owner (i.e. the app that defined it)
- Cannot be created within an App-Only context
  - Requires delegated permission for: Directory.AccessAsUser.All
- If “InDevelopment” state can be: added, updated, deleted
- **Resources can be filtered based on schema extensions**

Schema Extensions (cont.)

- Name of schema extensions can follow any of the following rules
  - {*domainName*}_{*schemaName*} if you have a registerd TLD
    - Your organization must own the namespace piasys as a part of one of the verified domains
  - {name} and Graph will rewrite it to:
    - ext{*8-random-alphanumeric-chars*}_{*schema-name*}
- To create Schema Extensions an app requires *Directory.AccessAsUser.All* permission
- To read and write values of schema extensions, you need permissions to read or write the target resource

Schema Extensions Data Types

- Binary (up to 256 bytes)
- Boolean (not supported for contacts, messages, events and posts)
- DateTime (ISO 8601 format. Stored in UTC)
- Integer (32bit - not supported for contacts, messages, events and posts)
- String (up to 256 characters)

DEMO

Working with the Microsoft Graph Extensions

Microsoft Graph - Wrap up!

- Microsoft Graph is the present (and the future) of Microsoft 365 development
- Rely on the Microsoft Graph SDK for a better developer experience
- Learn about REST, OAuth 2.0, JSON batching, and the Graph
- Keep an eye on Microsoft Graph know limitations
  - https://docs.microsoft.com/en-us/graph/known-issues
- Play with the Microsoft Graph and have fun!

Code sample download

https://github.com/PiaSys/Conferences-Samples/tree/master/MSGraph-VSCode

https://collabdays.be

Please rate this session and have a chance to win an awesome prize

#CollabDaysBE

#CollabDaysBE