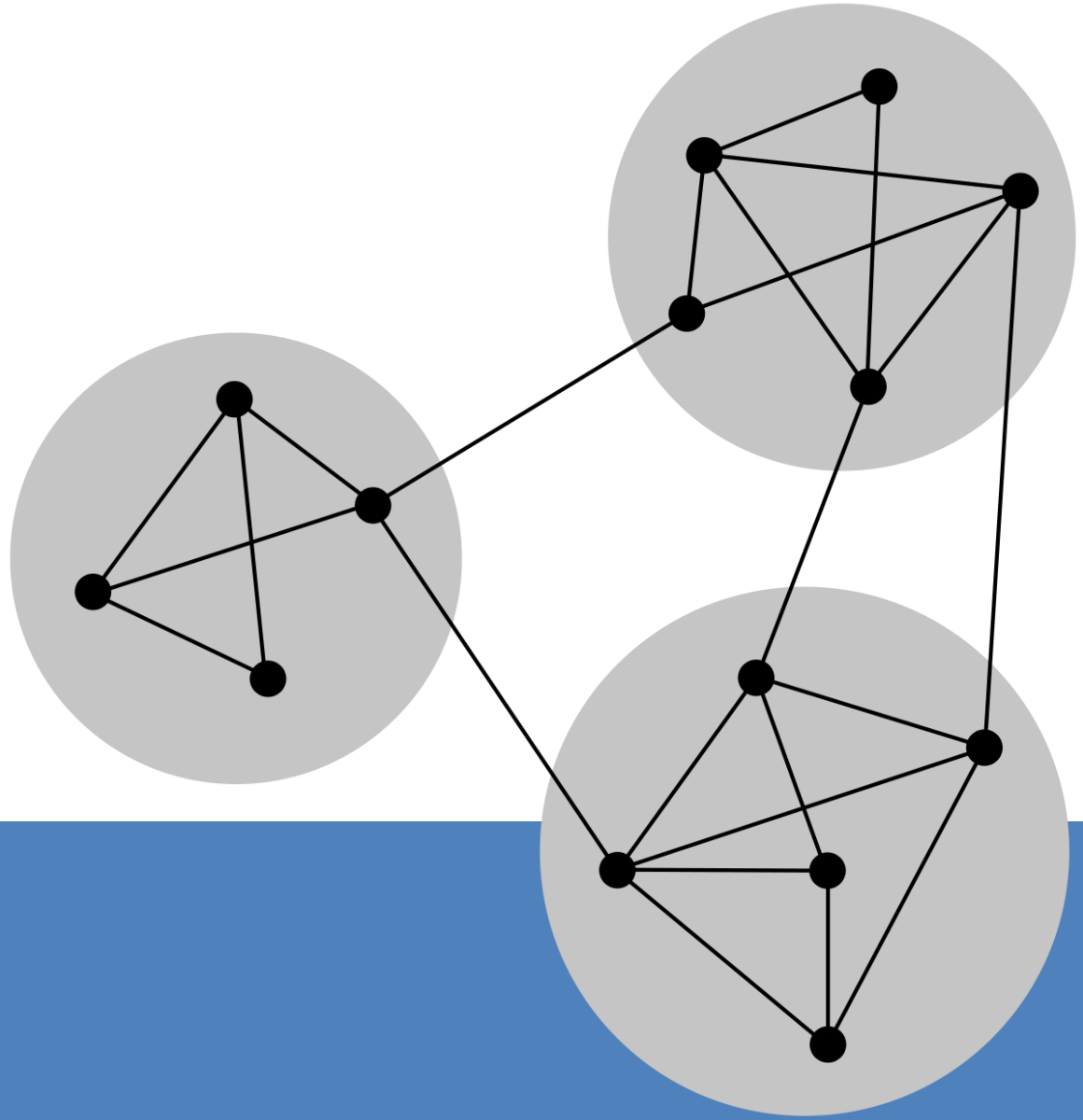
Global Microsoft 365 Developer Bootcamp

Lisbon, December 14th, 2019



Rise of the Graph

André Vala

Wi-Fi Code

msevent302db

André Vala

IT Deputy Director | Business Technology
Pestana Hotel Group

/in/andrevala
https://andrevala.com
@atomicvee
andre.vala@gmail.com

PESTANA
HOTEL GROUP

Agenda

What is the Microsoft Graph?

What's New in Microsoft Graph?

Microsoft Graph Capabilities

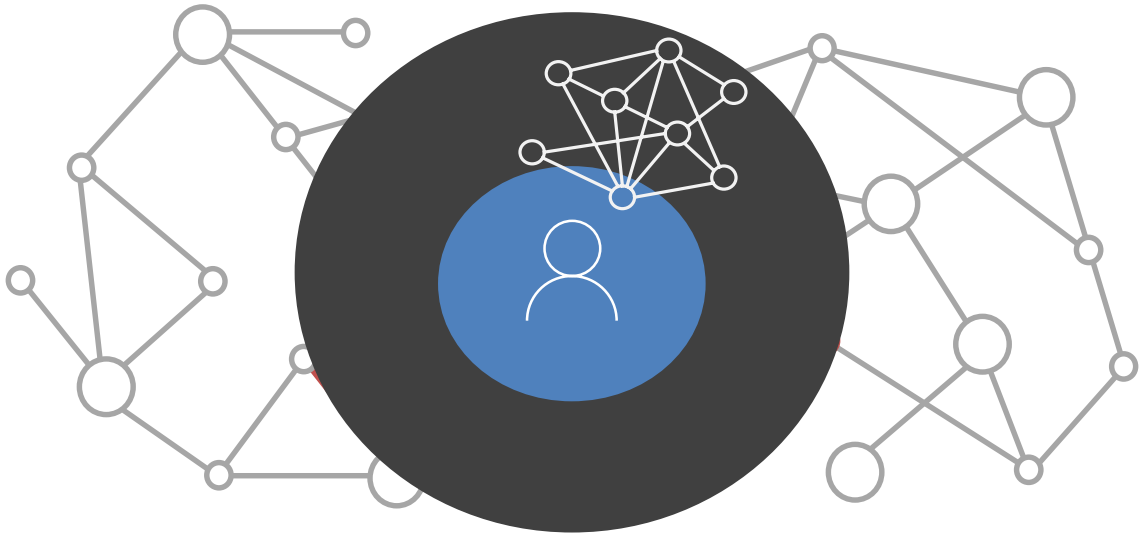
Microsoft Graph Data Connect

Microsoft Graph Toolkit

What is the Microsoft Graph?

Microsoft Graph is all about you

The fabric of all your data



It all starts with **`/me`**

Unified REST API for Microsoft 365

Azure Active Directory

Office 365

SharePoint, OneDrive, Outlook/Exchange, Microsoft Teams, OneNote, Planner, and Excel

Enterprise Security and Mobility

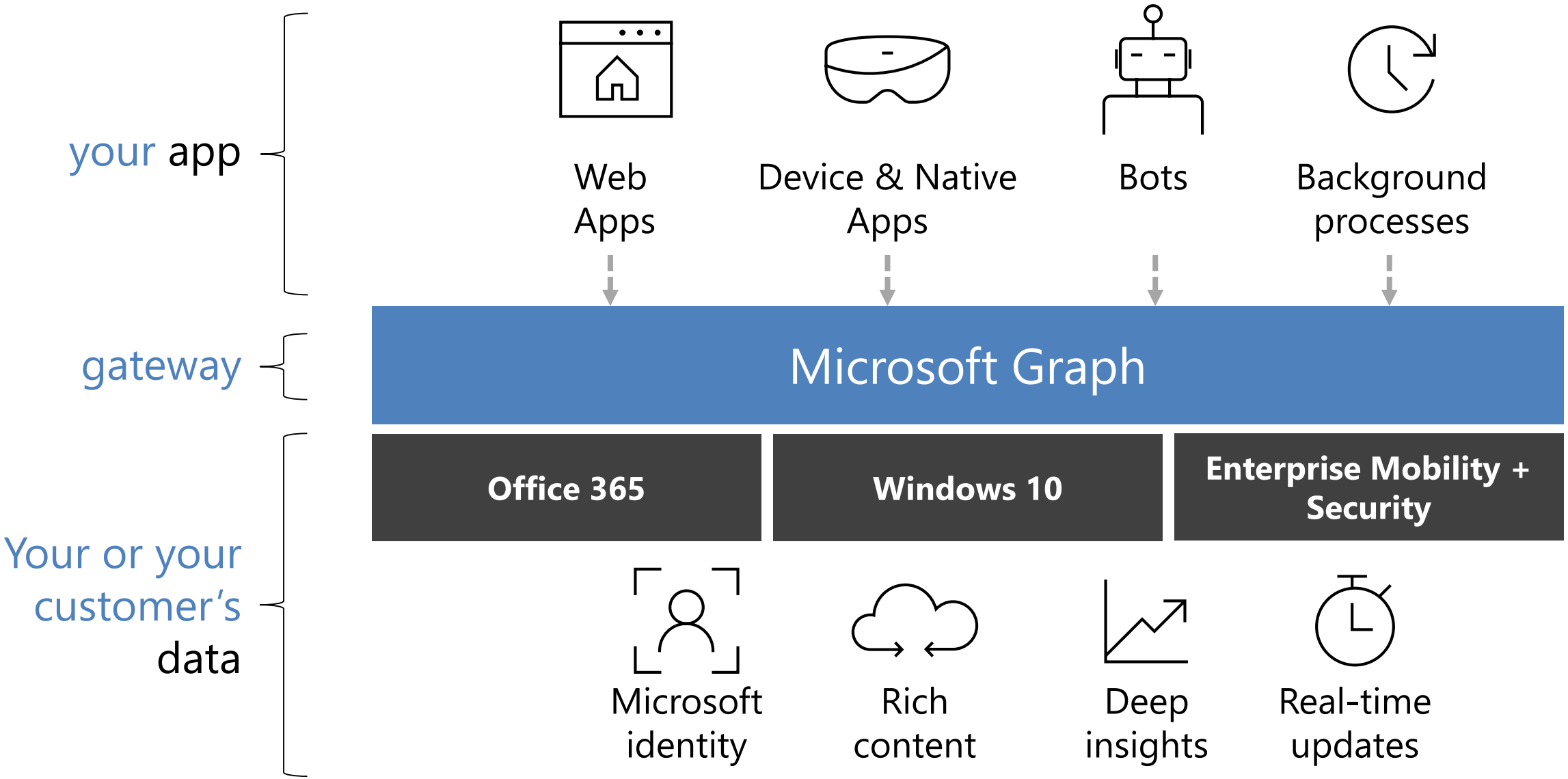
Identity Manager, Intune, Advanced Threat Analytics and Advanced Threat Protection.

Windows 10

Activities and Devices

Education

Gateway to you data in the Microsoft cloud



What's New in Microsoft Graph?

November 2019

Preview

Calendar: support for online meetings in outlook
Calendar: set properties of **`room`** and **`room list`**
Cloud Communication: new **`call`** resource type
Devices and Apps: several updates to Intune
Identity and Access: support for conditional access
Mail: change notifications on message resource
Notifications: support for web push
People: new profile resource
Search: new Microsoft Search API
Teamwork: get files associated with teams and channels
Users: new **`creationType`** property

Generally Available

Groups: list groups, read basic group properties, read/write the membership of groups
Groups: create groups with application permission
Identity and Access: register applications that authenticate with Azure AD
Mail: new **`conversationIndex`** property
Mail: get message or mail folder, track their changes and manage subscriptions for change notifications
Users: check group memberships

December 2019

Preview

Devices and Apps: several updates to Intune
Notifications: change notifications that include resource data
Identity and Access: new API to manage **`threatAssessementRequest`** resources
Teamwork: notifications for new/edited channel messages and chat messages

Generally Available

Insights: Insights APIs available in v1.0
Toolkit: version 1.1 released

Microsoft Graph Capabilities

Microsoft Graph Capabilities

- Delta queries
- Webhooks
- Custom data
- Working with insights
- Batch requests

Delta Queries

Optimize your queries by retrieving only what has changed since you last queried the Graph.

Optional query parameters

Paging with **nextLink**

Change tokens with **deltaLink**

Supported resources for delta queries

Optional Query Parameters

Value	Description	Example
`$count`	Retrieves the total count of matching resources	`/me/messages?$top=2&count=true`
`$expand`	Retrieves related resources	`/groups?$expand=members`
`$filter`	Filters results (rows)	`/users?$filter=startsWith(givenName,'J')`
`$orderBy`	Orders results	`/users?$orderBy=displayName desc`
`$search`	Returns results based on search criteria. Currently supported on **messages** and **person** collections	`/me/messages?$search=pizza`
`$select`	Filters properties (columns)	`/users?$select=givenName,surname`
`$skip`	Indexes into a result set, also used by some APIs to implement paging and can be used together with $top to manually page results	`/me/messages?$skip=11`
`$skipToken`	Retrieves the next page of results from result sets that span multiple pages.(Some APIs use $skip instead)	`https://graph.microsoft.com/v1.0/users?$skiptoken=X%27445370 ... 0000000%27`
`$top`	Sets the page size of results	`/users?$top=10`

$filter

Support for $filter varies across Microsoft Graph APIs, but the following are generally supported:

- Equals – **eq**
- Not equals – **ne**
- Greater than – **gt**
- Greater than or equals – **ge**
- Less than – **lt**
- Less than or equals – **le**
- And – **and**
- Or – **or**
- Not - **not**

Paging with nextLink

Paging with Microsoft Graph

Some queries return multiple pages of data

Page size is specified using **`$top`** query parameter

@odata.nextLink

Presence indicates there are additional pages of data

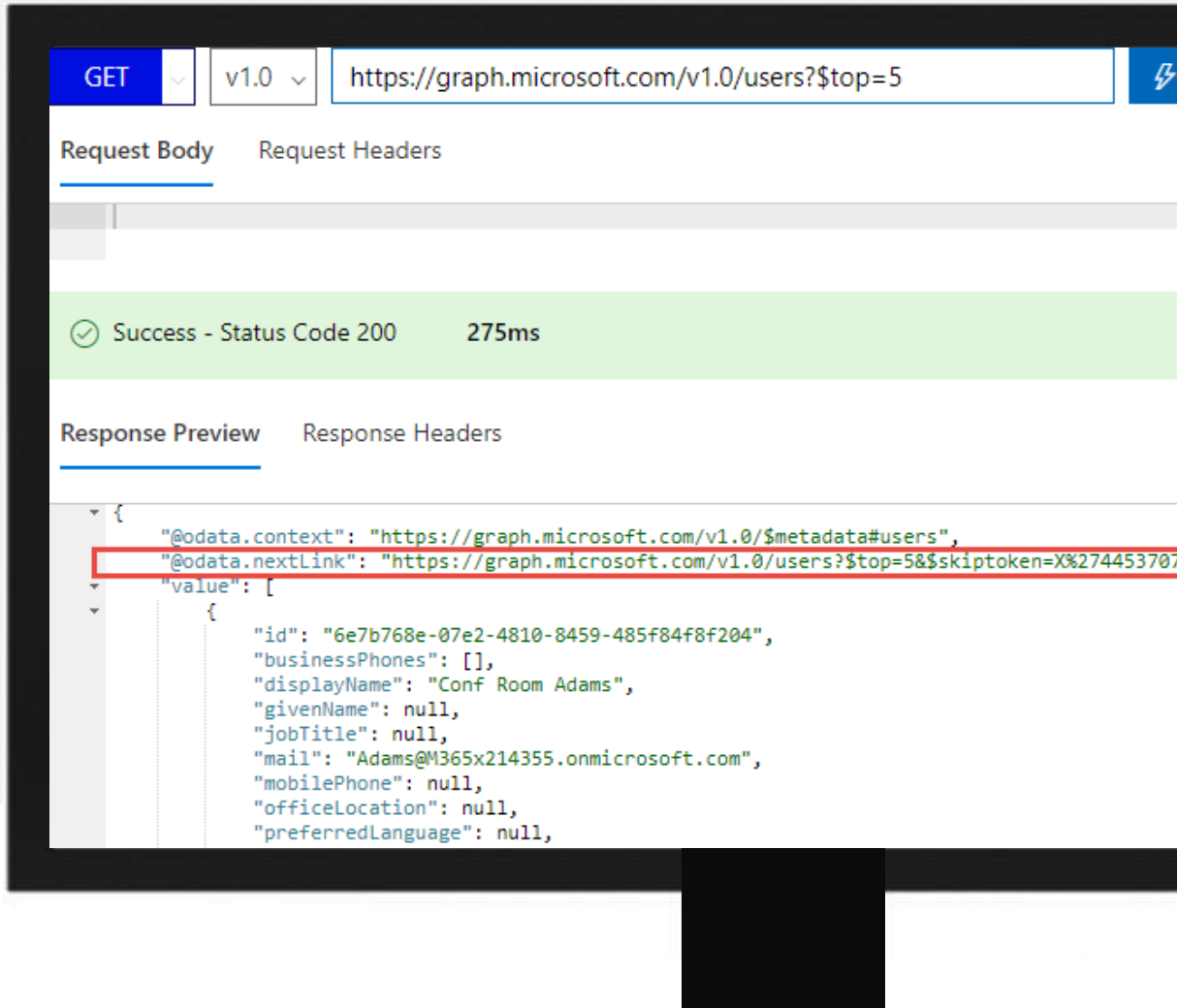
Provides full URL link to next page of data

API behavior

Not all APIs support paging

Different default and max page sizes

Some support backwards paging (**`&previous-page=true`**)



Change tokens with deltaLink

Delta queries

Discover newly created, updated, or deleted entities without a full read of the target resource

Useful for synchronizing changes to a local data store

Requires permission to read the requested resource

@odata.nextLink

The **`$select`** query parameter from the initial request is encoded into the **`nextLink`** URL

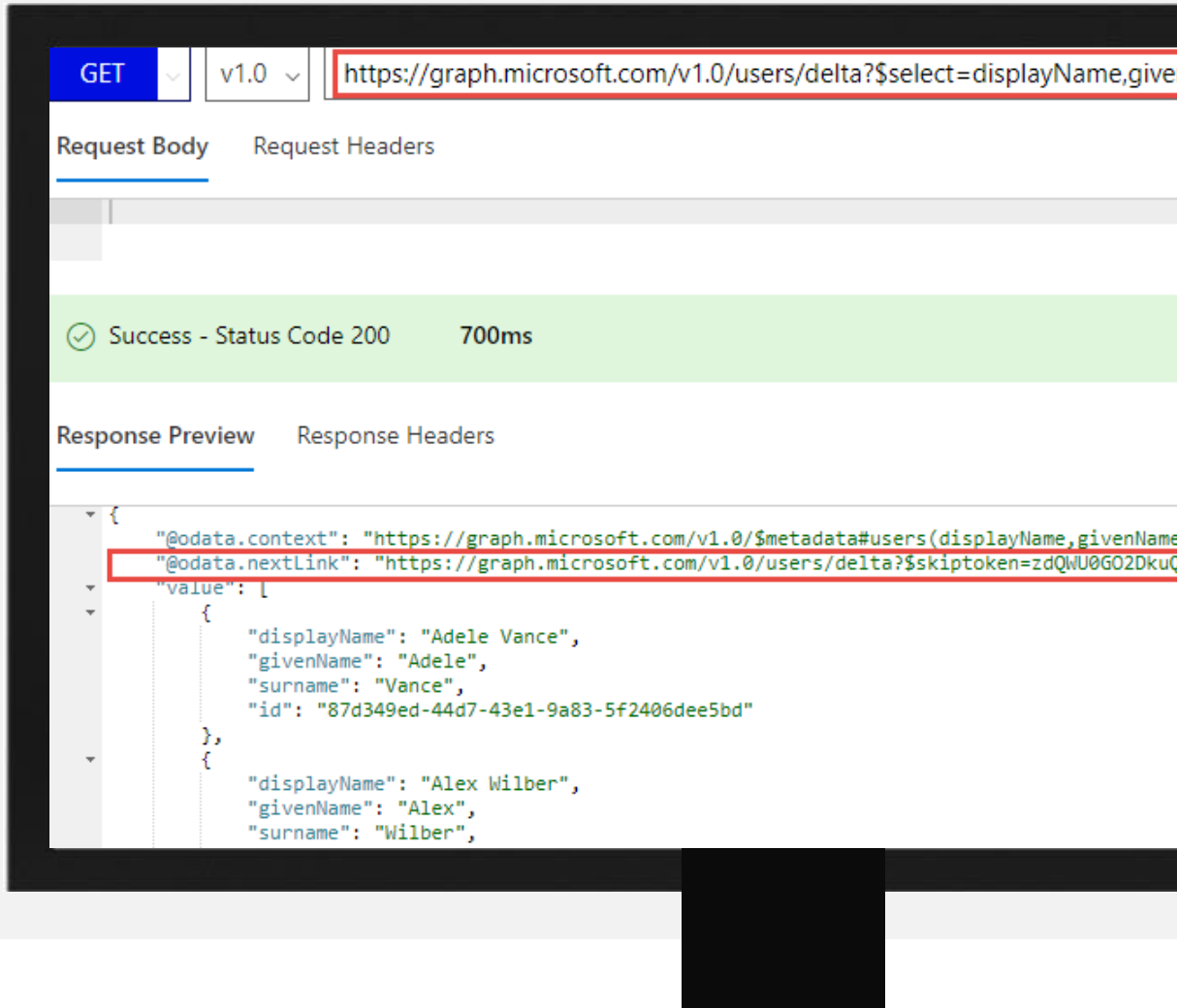
Presence of **`nextLink`** indicates more data is available

@odata.deltaLink

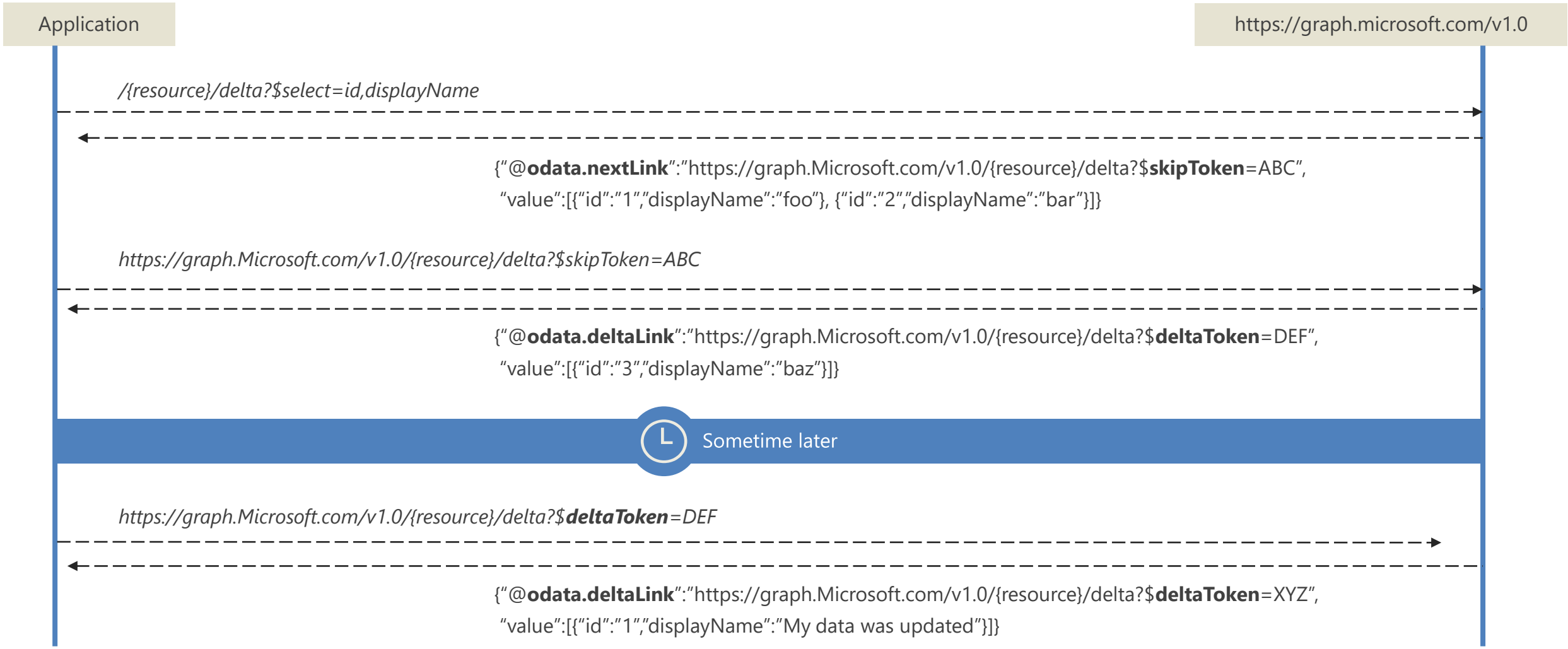
Presence of **`deltaLink`** indicates no more data to be returned

Contains **`deltaToken`**, save this for future queries

If no changes have occurred, the same **`deltaToken`** is returned with no results



Typical call pattern to track changes



Resources that support delta queries

Resource collection	API
Directory roles	Delta function of the **directoryRole** resource
Groups	Delta function of the **group** resource
Mail folders	Delta function of the **mailFolder** resource
Messages in a folder	Delta function of the **message** resource
Personal contact folders	Delta function of the **contactFolder** resource
Personal contacts in a folder	Delta function of the **contact** resource
Users	Delta function of the **user** resource
Drive items	* Delta function of the **driveItem** resource
Events in a calendar view (date range) of the primary calendar	Delta function of the **event** resource

https://docs.microsoft.com/en-us/graph/delta-query-overview?view=graph-rest-1.0

Delta Queries

Demo

https://github.com/microsoftgraph/msgraph-training-webhooks-customdata-insights/tree/master/Demos/01-user-changes

Webhooks

Let Microsoft Graph notify your app when specific events occur, instead of polling the Graph waiting for changes.

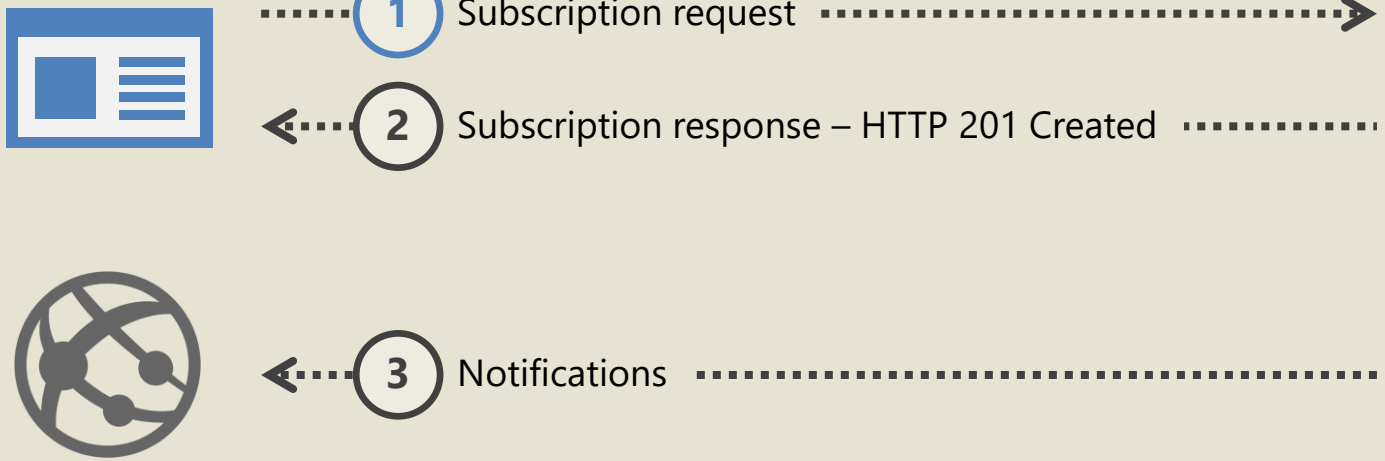
Webhook subscriptions

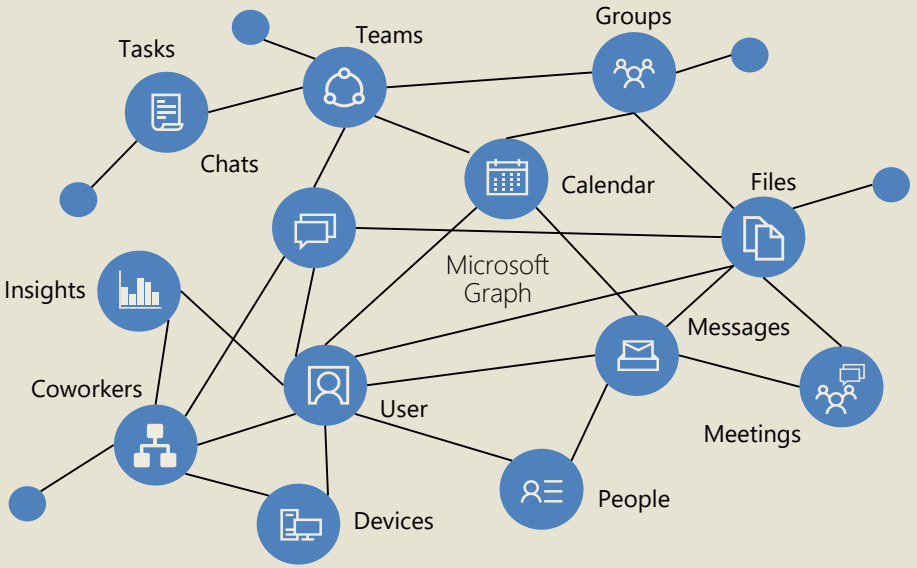
Understanding webhook apps

Configuring webhooks

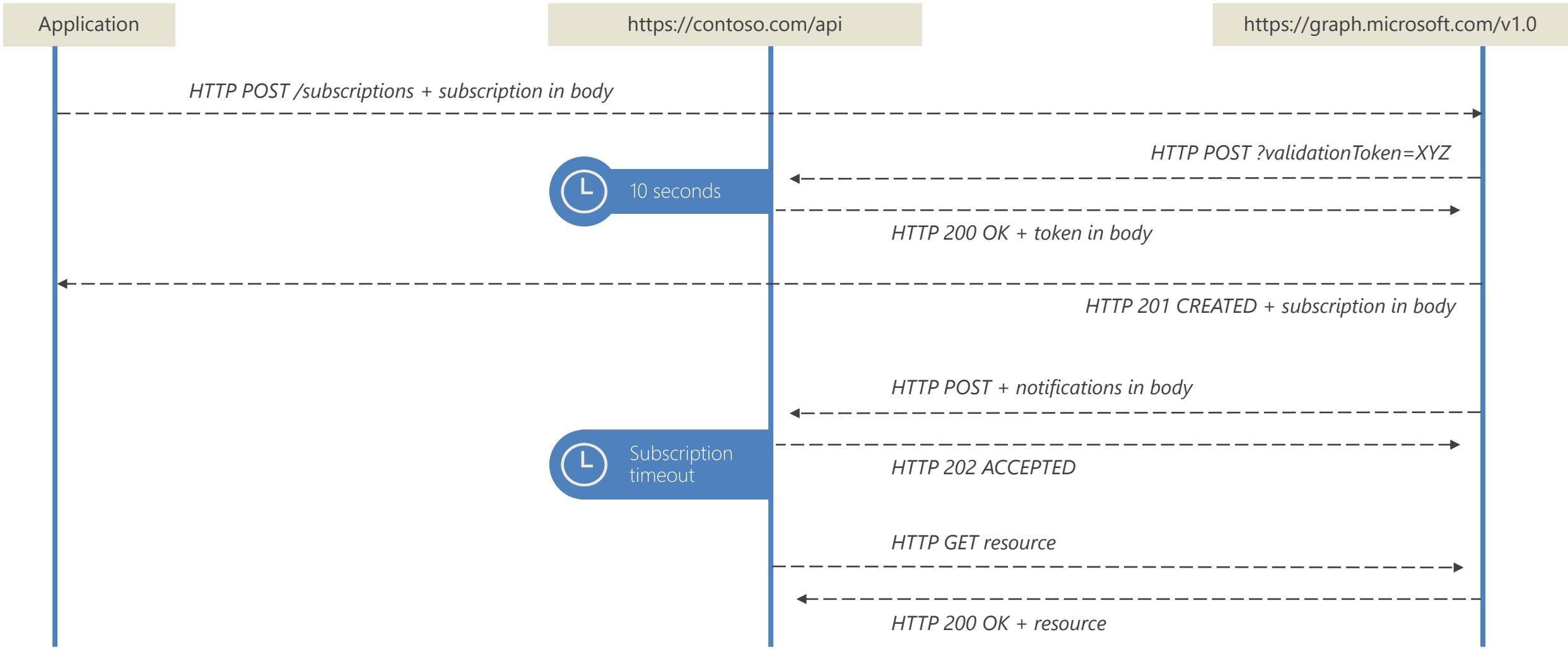
Retrieving changes from webhooks

Microsoft Graph webhook subscriptions





Token validation and notification responses



Understanding webhook apps

Subscriptions

Application sends a subscription request

Microsoft Graph responds with HTTP 201 – Created and a subscription object in the body

Microsoft Graph sends notifications to the HTTPS endpoint during the lifetime of the subscription

Subscriptions require read permission to the resource

Expiration

Subscriptions expire (time varies)

Applications must renew subscription requests

Applications can unsubscribe from subscriptions

Permission type	Supported resource types in Microsoft Graph v1.0
Delegated – work or school account	Contact, conversation, drive, event, message
Delegated – personal Microsoft account	None
Application	Contact, conversation, event, message

Maximum expiration times

Resource	Maximum expiration time
Users	4230 minutes (~70 hours)
Groups	4230 minutes (~70 hours)
Mail	4230 minutes (~70 hours)
Calendar	4230 minutes (~70 hours)
Contacts	4230 minutes (~70 hours)
Group conversations	4230 minutes (~70 hours)
Drive root items	86400 minutes (~ 60 days)

Webhooks

Demo

https://github.com/microsoftgraph/msgraph-training-webhooks-customdata-insights/tree/master/Demos/02-webhooks

Custom Data

Add you own data to resources in the Microsoft Graph.

Open Extensions

Schema Extensions

Extending Microsoft Graph

Add custom data to resources using extensions

Keep your app lightweight and store app-specific user profile data in Microsoft Graph

Keep your existing user profile store and add an app-specific identifier to Microsoft Graph

Add metadata to messages to provide new conversation capabilities

Enhance calendar events with contextual data

Two types of extensions

Open extensions – open and flexible

Schema extensions – more versatile: schema is discoverable and shareable, enables filtering

Open Extensions

Good way for developers to get started

Open types that offer a flexible way to add untyped app data directly to a resource instance

Accessible through the **`extensions`** navigation property of the resource instance

`extensionName` is the only pre-defined, writeable property, must be unique within the tenant

Use a reverse DNS naming system (i.e. Com.Contoso.ContactInfo), but do not use Microsoft (com.Microsoft or com.onMicrosoft).

Open Extensions Example

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-length: 420

{
    "id": "84b80893-8749-40a3-97b7-68513b600544",
    "displayName": "John Smith",
    "mail": "john@contoso.com",
    "extensions": [
        {
            "@odata.type": "#microsoft.graph.openTypeExtension",
            "extensionName": "com.contoso.roamingSettings",
            "id": "com.contoso.roamingSettings",
            "theme": "dark",
            "color": "purple",
            "lang": "Japanese"
        }
    ]
}
```

Schema Extensions

Versatility

Define schema extension definition and extend resource instances with strongly-typed custom data.

Discoverable by other apps via status property

Schema extensions are complex types, enabling use of HTTP verbs:

- POST to specify custom data when creating a new resource instance
- GET to read the custom data
- PATCH to add or update in an existing resource instance
- PATCH to set the complex type to null, deleting the custom data

Schema Extensions Example

GET https://graph.microsoft.com/v1.0/groups?$filter=graphlearn_courses/courseId eq ‘123’&
$select=displayName,id,description,graphlearn_courses

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-length: 326
{
  "value": [
    {
      "displayName": "New Managers March 2017",
      "id": "14429ae5-3e74-41a2-9fa8-028fbb984637",
      "description": "New Managers training course for March 2017",
      "graphlearn_courses": {
        "@odata.type": "#microsoft.graph.ComplexExtensionValue",
        "courseId":"123",
        "courseName":"New Managers",
        "courseType":"Online"
      }
    }
  ]
}
```

Supported Resources

Resource	Open extensions	Schema extensions
Administrative unit	Preview only	Preview only
Calendar event	GA	GA
Device	GA	GA
Group	GA	GA
Group calendar event	GA	GA
Group conversation post	GA	GA
Message	GA	GA
Organization	GA	GA
Personal contact	GA	GA
User	GA	GA

Schema Extensions Lifecycle

State	Lifecycle state behavior
InDevelopment	Initial state. Only the owner app can extend resource instances with this schema definition, and only in the same directory where the owner app is registered. Only the owner app can update the extension definition with additive changes or delete it. The owner app can move from InDevelopment to Available
Available	Available for use by all apps in any tenant. Any app can add custom data to instances of those resource types (as long as it has permission to the resource). Only the owner app can update the extension definition with additive changes. No app can delete the extension definition while in this state. The owner app can move from Available to Deprecated.
Deprecated	Schema extension definition can no longer be read or modified. No app can view, update, add new properties, or delete the extension. Apps can, however, still read, update, or delete existing extension property values. The owner app can move from Deprecated back to Available.

Permissions and Data Limits

Permissions

The same permissions that are required to read or write from a resource are required to read or write to any extensions on that resource.

To create and manage schema definitions, app must be granted *Directory.AccessAsUser.All*

Data limits

Open extension limits

Each open extension can have up to 2KB of data (including the extension definition itself)

An application can add up to two open extensions per resource instance

Schema extension limits

An application may create no more than five schema extension definitions

Directory resources (device, group, and user) limit total number of property extension values to be set on a resource instance to 100

Custom Data

Demo

https://github.com/microsoftgraph/msgraph-training-webhooks-customdata-insights/tree/master/Demos/03-custom-data

Working with Insights

Leverage relationships calculated by the Graph using advanced analytics and machine learning.

Understanding the Insights resource type

Trending

Used

Shared

Retrieving insights collections

Understanding the Insights resource type

Relationships calculated using advanced analytics and machine learning

Trending – documents from OneDrive and SharePoint sites trending around a user

Used – documents viewed and modified by a user

- Documents the user used in OneDrive for Business
- SharePoint
- Opened as email attachments

Shared – documents shared with a user as email attachments or OneDrive for Business links sent in emails

```
https://graph.microsoft.com/v1.0/me/insights/{api}
```

Permissions for Insights

Permission type	Permissions (least to most privileged)
Delegated (work or school account)	Sites.Read.All, Sites.ReadWrite.All
Delegated (personal Microsoft account)	Not supported
Application	Sites.Read.All, Sites.ReadWrite.All

Returned Data

Every insight is returned with both a **`resourceVisualization`** and a **`resourceReference`** complex value type

resourceReference – reference to the item

- `webUrl`
- `id`
- `Type`

resourceVisualization – visualization data, used to display the item

- `title`
- `type`
- `mediaType`
- `previewImageUrl`
- `previewText`
- `containerWebUrl`
- `containerDisplayName`
- `containerType`

Trending

Rich relationship connecting a user to documents that are trending around the user (are relevant to the user). OneDrive files, and files stored on SharePoint team sites can trend around the user.

```
{
  "id": "string",
  "weight": "double",
  "resourceVisualization": [{"@odata.type": "microsoft.graph.resourceVisualization"}],
  "resourceReference": [{"@odata.type": "microsoft.graph.resourceReference"}],

  "resource": [ { "@odata.type": "microsoft.graph.entity" } ]
}
```

Used

An insight representing documents used by a specific user. The insights returns the most relevant documents that a user viewed or accessed. This includes documents in:

- OneDrive for Business
- SharePoint

```
{
  "id": "string",
  "lastUsed": "usageDetails",
  "resourceVisualization": "resourceVisualization",
  "resourceReference": "resourceReference",

  "resource": [ { "@odata.type": "microsoft.graph.entity" } ]
}
```

Shared

An insight representing files shared with or by a specific user. The following shared files are supported:

- Files attached directly in an email or a meeting invite.
- OneDrive for Bussiness and SharePoint modern attachments - files stored in OneDrive for Business and SharePoint that users share as a links in an email.

```
{
  "id": "string",
  "lastShared": "sharingDetail",
  "resourceVisualization": "resourceVisualization",
  "resourceReference": "resourceReference",

  "resource": [ { "@odata.type": "microsoft.graph.entity" } ]
}
```

Retrieving Insights Collections

Get the items trending around me

```
https://graph.microsoft.com/v1.0/me/insights/trending
```

Get the resourceVisualization for the items trending around me

```
https://graph.microsoft.com/v1.0/me/insights/trending?$select=resourceVisualization
```

Get the resourceVisualization for my shared items that are attachments in emails

```
https://graph.microsoft.com/v1.0/me/insights/shared?$sharingType eq 'Attachment'&$select=resourceVisualization
```

Show the OneNote notebooks that I have used

```
https://graph.microsoft.com/v1.0/me/insights/used?$filter=resourceVisualization/type eq 'OneNote'
```

Batch Requests

Optimize multiple requests by using batches and sequence requests using dependencies.

Batch request

Batch response

Sequencing requests

Bypassing URL lenght limits

Batch Request

Request format

Batch requests are always sent using POST to the **`/$batch`** endpoint.

A JSON batch request body consists of a single JSON object with one required property: **`requests`**

The **`requests`** property is an array of individual requests. For each individual request, the **`id`**, **`method`**, and **`url`** properties are required.

The **`id`** property functions primarily as a correlation value to associate individual responses with requests.

The **`method`** and **`url`** properties are exactly what you would see at the start of any given HTTP request.

Individual requests can optionally also contain a **`headers`** property and a **`body`** property.

Batch Request Example

```
POST https://graph.microsoft.com/v1.0/$batch
Accept: application/json
Content-Type: application/json

{
    "requests": [
    {
      "id": "1",
      "method": "GET",
      "url": "/me/drive/root:/{file}:/content"
    },
    {
      "id": “2",
      "url": "/me",
      "method": "PATCH",
      "body": {
        "city" : "Redmond"
      },
      "headers": {
        "Content-Type": "application/json"
      }
    }
  ]
}
```

Batch Response

Response format

The property in the main JSON object is named responses as opposed to requests

Individual responses might appear in a different order than the requests

Individual responses have a **`status`** property. The value of **`status`** is a number that represents the HTTP status code.

The status code on a batch response is typically **`200`** or **`400`**. If the batch request itself is malformed, the status code is **`400`**.

A **`200`** status code on the batch response does not indicate that the individual requests inside the batch succeeded.

A **`nextLink`** property might be included in the batch response. To ensure that all individual responses have been received, continue to follow the **`nextLink`** as long as it exists.

Batch Response Example

```
200 OK
Content-Type: application/json

{
    "responses": [
    {
      "id": "1",
      "status": 302,
      "headers": {
        "location": "https://b0mpua-by3301.files.1drv.com/y23vmagahszhxzlcvhasdhasghasodfi"
      }
    },
    {
      "id": "4",
      "status": 204,
      "body": null
    }
  ]
}
```

Sequencing Requests

Sequencing requests with the dependsOn property

The **`dependsOn`** property is an array of strings that reference the **`id`** of a different individual request.

The values for **`id`** must be unique.

```
POST https://graph.microsoft.com/v1.0/$batch
Accept: application/json
Content-Type: application/json

{
    "requests": [
    {
      "id": "1",
      "method": "GET",
      "url": "..."
    },
    {
      "id": "2",
      "dependsOn": [ "1" ],
      "method": "GET",
      "url": "..."
    }
  ]
}
```

Bypassing URL length limitations

Versatility

An additional use case for JSON batching is to bypass URL length limitations.

In cases where the filter clause is complex, the URL length might surpass limitations built into browsers or other HTTP clients.

You can use JSON batching as a workaround for running these requests because the lengthy URL simply becomes part of the request payload.

Microsoft Graph Data Connect

Challenging scenarios with Microsoft Graph

Today's most interesting data-driven applications

- require reasoning & analysis over massive datasets
- built by extracting data from Office 365 via Microsoft Graph APIs

Microsoft Graph is optimized for simple, real-time, entity centric queries, not:

- Data Access in bulk - requires expensive engineering
- Data Privacy is at risk - given the current all-or-nothing consent models
- Data Security & Data Governance - entirely up to the experience author

What is Microsoft Graph Data Connect?

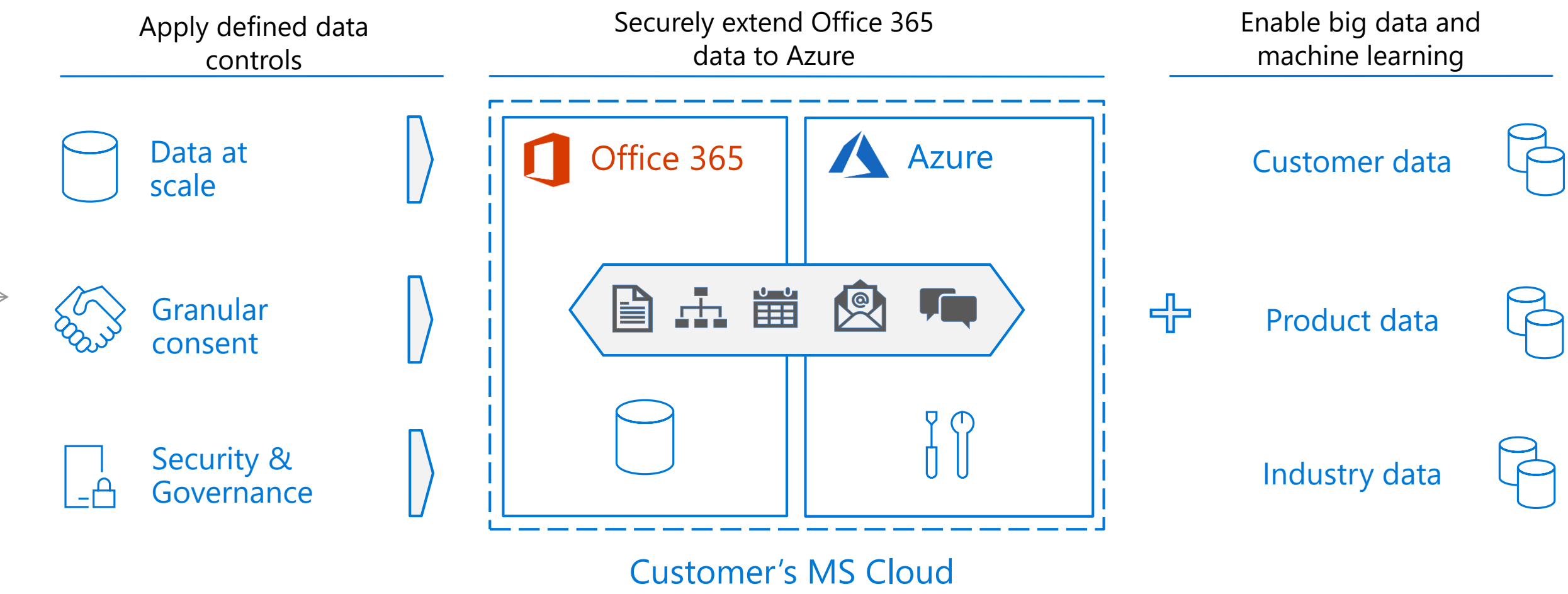
New model for data access & hosting applications within the customer's control

Allows developers to build intelligent applications and experiences using Office 365 data & Azure resources

Application access to customer data requires explicit customer approval

Microsoft Graph remains the best approach for enabling end-user task centric apps

Microsoft Graph Data Connect



Graph APIs vs Data Connect

	Graph APIs	Graph data connect
Access scope	Single use / entire tenant	Many users / groups
Access pattern	Real time	Recurrent schedule
Data operations	Operates on data master	Operates on cache of the data
Data protection	Data is protected while in Microsoft 365	Data protection is extended to the cache of data in your Azure subscription
User consent	Self; resource types	None
Admin consent	Entire organization; resource types	Select groups of users; resource types & properties; excludes users
Access tools	RESTful web queries	Azure Data Factory

Microsoft Graph Toolkit

Microsoft Graph Toolkit

Collection of reusable, framework-agnostic web components and helpers for accessing and working with Microsoft Graph.

Supported in React and Angular

Supported in all latest versions of browsers

https://docs.microsoft.com/en-us/graph/toolkit/overview

Microsoft Graph Toolkit Components

Person

Display a person or contact by using their photo, name and/or email address.

Person Card

Responsive component to display more information related to a person.

People

Display a group of people or contacts by using their photos or initials.

People Picker

Searches for a specified number of people and render the list of results.

Login

Button and flyout control to facilitiate Microsoft identity platform authentication.

Get

Make any GET query from Microsoft Graph directly in your HTML.

Agenda

Represents events in a user or group calendar.

Tasks

Enables the user to view, add, remove, complete, or edit tasks. It works with tasks in Microsoft Planner or Microsoft To-Do.

Labs

Lab #1

Building Microsoft Graph Applications

120 mins

In this lab

1. Build a .NET console application using Microsoft Graph
2. Build a Javascript application using Microsoft Graph
3. Build an Azure Function using Microsoft Graph
4. Build a mobile application with Xamarin using Microsoft Graph

https://github.com/microsoftgraph/msgraph-training-building-apps/blob/master/Lab.md

Lab #2

Authenticate and Connect to Microsoft Graph

120 mins

In this lab

1. Obtain tokens and connect with the Microsoft Graph using REST
2. Connecting with Microsoft Graph using OpenID Connect
3. Dynamic Permissions with the Azure AD v2.0 endpoint and Microsoft Graph

https://github.com/microsoftgraph/msgraph-training-authentication/blob/master/lab.md

Lab #3

Microsoft Graph Capabilities

120 mins

In this lab

1. Microsoft Graph delta queries
2. Microsoft Graph webhooks
3. Adding custom data to resources in Microsoft Graph
4. Developing insights with Microsoft Graph
5. Creating batch requests with Microsoft Graph

https://github.com/microsoftgraph/msgraph-training-webhooks-customdata-insights/blob/master/Lab.md

Lab #4

Build apps with the Microsoft Graph REST API

120 mins

In this lab

1. Create an Azure AD web application with the App Registration Portal
2. Working with the Microsoft Graph REST API in Postman
3. Create & configure an ASP.NET MVC web application & configure it for MSAL
4. Update the ASP.NET MVC application to leverage the Microsoft Graph REST API

https://github.com/microsoftgraph/msgraph-training-restapi/blob/master/Lab.md

Lab #5

Using Microsoft Graph data connect to analyze emails to find subject matter experts

120 mins

In this lab

1. Setup tenant & get data out of Office 365 with Graph Data Connect and Data Factory
2. Extract Office 365 data with Graph Data Connect
3. Process the exported data

https://github.com/microsoftgraph/msgraph-training-dataconnect/blob/master/Lab.md

Lab #6

Optimize data usage when using Microsoft Graph with query parameters

45 mins

In this lab

1. Demonstrate how to manipulate REST queries with query parameteres
2. Create queries that expand complex entities
3. Demonstrate how to search for content with Microsoft Graph
4. Optimize Microsoft Graph queries with batching

https://docs.microsoft.com/en-us/learn/modules/optimize-data-usage/

Lab #7

Optimize network traffic with Microsoft Graph

70 mins

In this lab

1. Explain how Microsoft Graph maintains resource health
2. Identify when Microsoft Graph throttles requests
3. Decide the appropriate pattern to address throttled requests
4. Create queries that mitigate throttling scenarios

https://docs.microsoft.com/en-us/learn/modules/optimize-network-traffic/

Questions?

/in/andrevala

https://andrevala.com

@atomicvee

andre.vala@gmail.com