

Mail, Calendar, and Contacts REST APIs and Hybrid Exchange

Deepak Singh

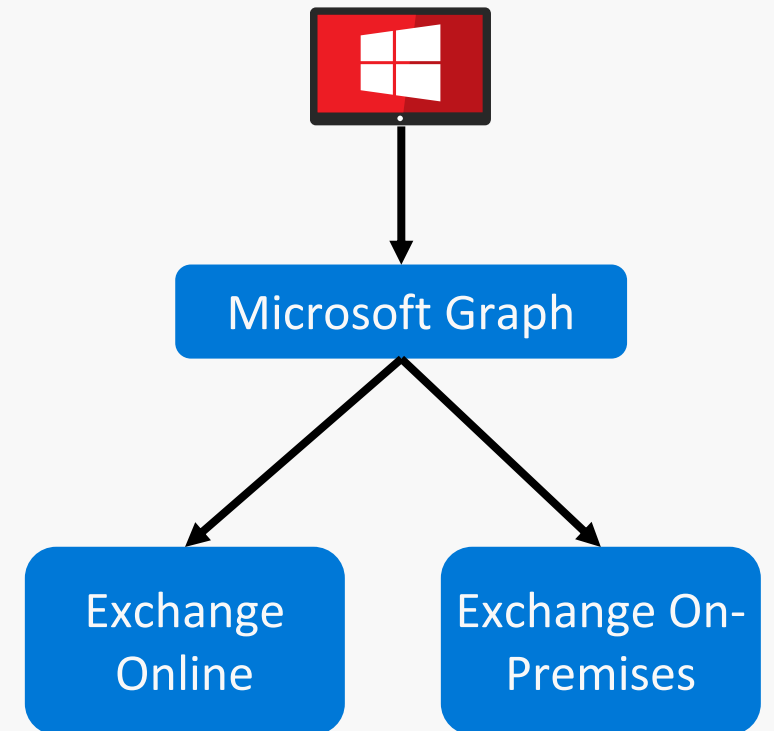
Agenda

• Session objective

- What is Microsoft Graph?
- What are the benefits of Microsoft Graph supporting hybrid Exchange deployments?
- Architecture overview
- Configuring/managing on-premises Exchange 2016 for Microsoft Graph
- Sneak-peek into what is coming

• Key takeaways

- **Simple:** One API endpoint & one auth mechanism to reach online & on-premises mailboxes!
- **Secure:** Industry standard OAuth 2 flow that respects your authentication policies.
- **Easier:** REST APIs that are much easier than EWS SOAP APIs to build apps on your platform of choice.
- **Available:** Now offered in Preview for you to use and give us feedback!



Overview

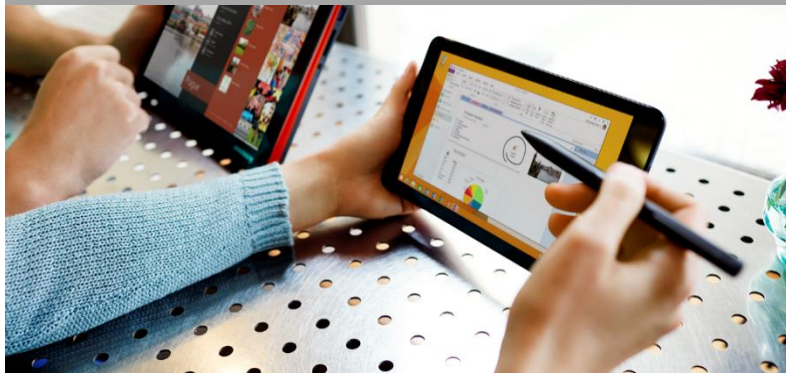
Office Platform

WHAT CAN I BUILD?



ADD-INS AND WEB PARTS:

Make your solution a native part of the modern Office



WEB AND DEVICE APPS:

Build smarter apps by connecting to Office services



VOICE, VIDEO, CONNECTORS, AND BOTS:

Create the next generation of productivity solutions



Why do the REST APIs matter?

- 100's of millions of users

- Office 365 & Outlook.com for production apps & services
- Hybrid Exchange deployment support in preview
- On-premises Exchange on our roadmap



- Many of these are highly engaged Outlook users.

- Powerful, intuitive way to build apps & services

- Granular, tightly scoped permissions to access data

- Based on open standards – OAuth 2.0, JSON & OData 4.0



API Landscape

• Modern APIs

- Microsoft Graph
 - The API to use to build apps & services integrating w/ Office services for cloud & hybrid users
- Outlook REST API
 - Outlook/Exchange specific API endpoint to use if Microsoft Graph doesn't have the APIs yet.

• Older APIs

- Exchange Web Services (EWS) SOAP APIs
 - Available since Exchange 2007 SP1 for integration with Exchange. No granular access.
- Exchange Active Sync (EAS)
 - Used by mobile device manufacturers for built-in messaging client. Requires licensing.
 - Not intended for any other integrations including line of business apps & services

Microsoft Graph

What is Microsoft Graph?

Single API for:

1. Accessing data

/me, /users, /groups, /messages, /drive, ...

2. Traversing data

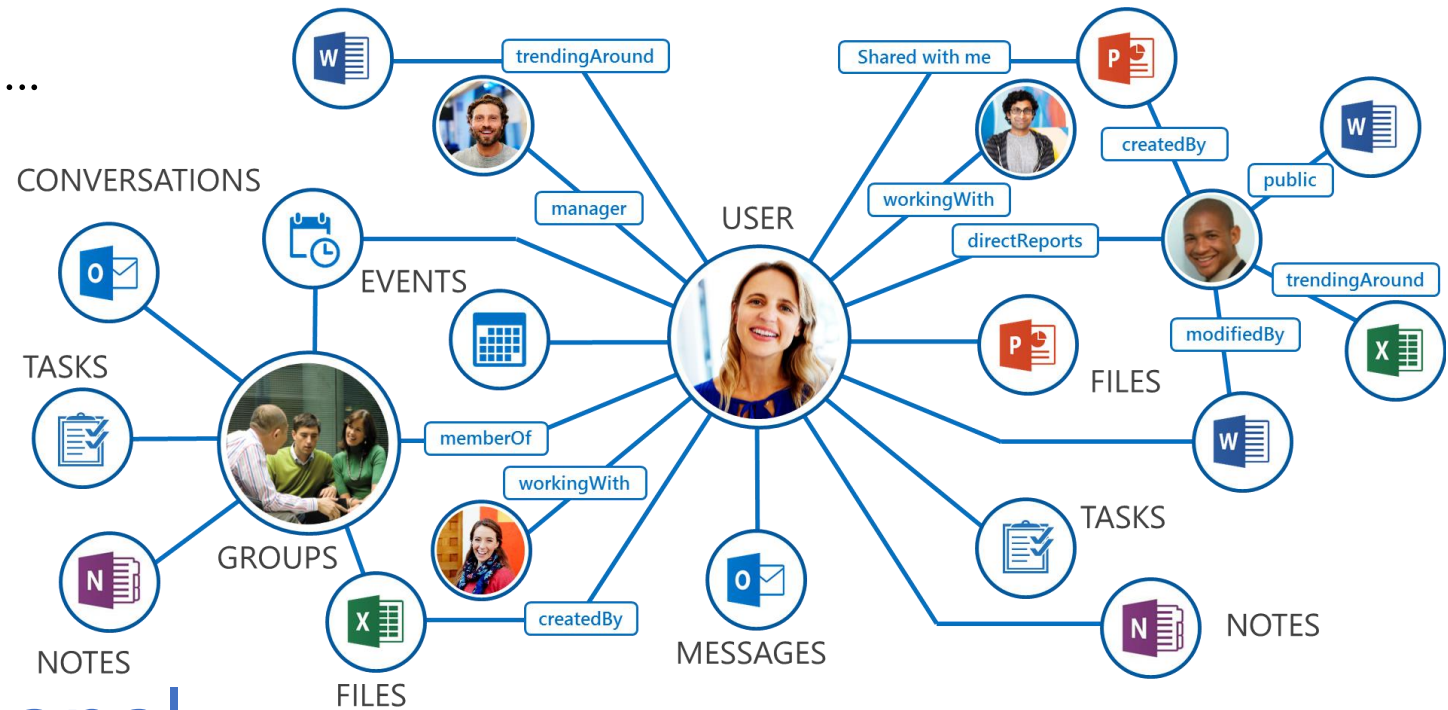
/drive/items/<id>/lastmodifiedByUser

3. Accessing insights

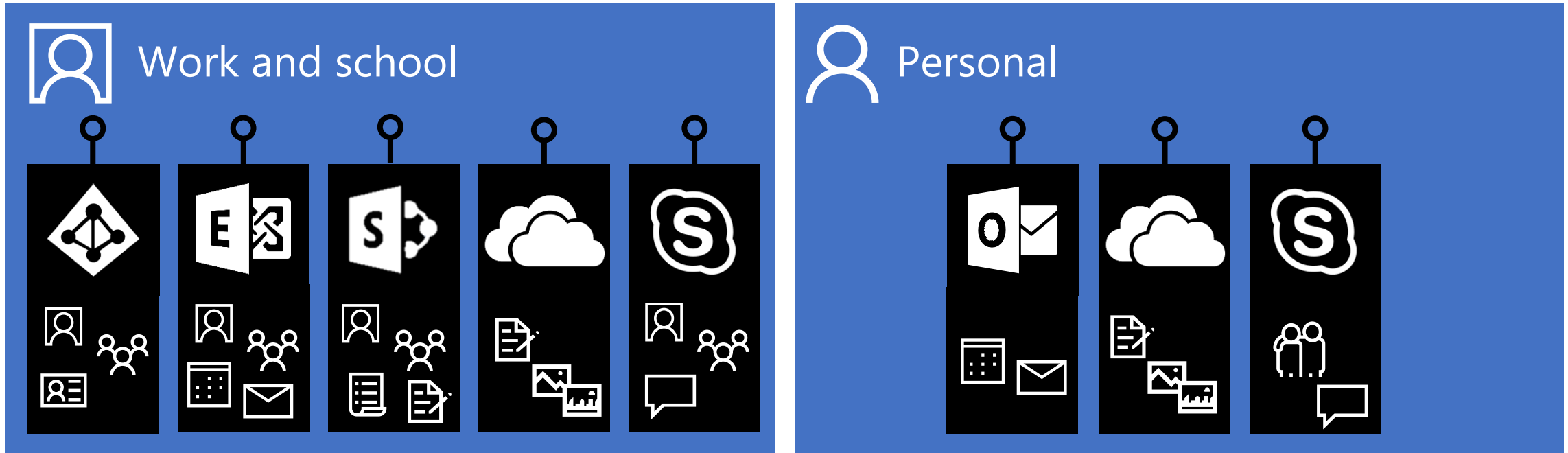
/insights/trending

4. Work/School and Personal

<https://graph.microsoft.com/>

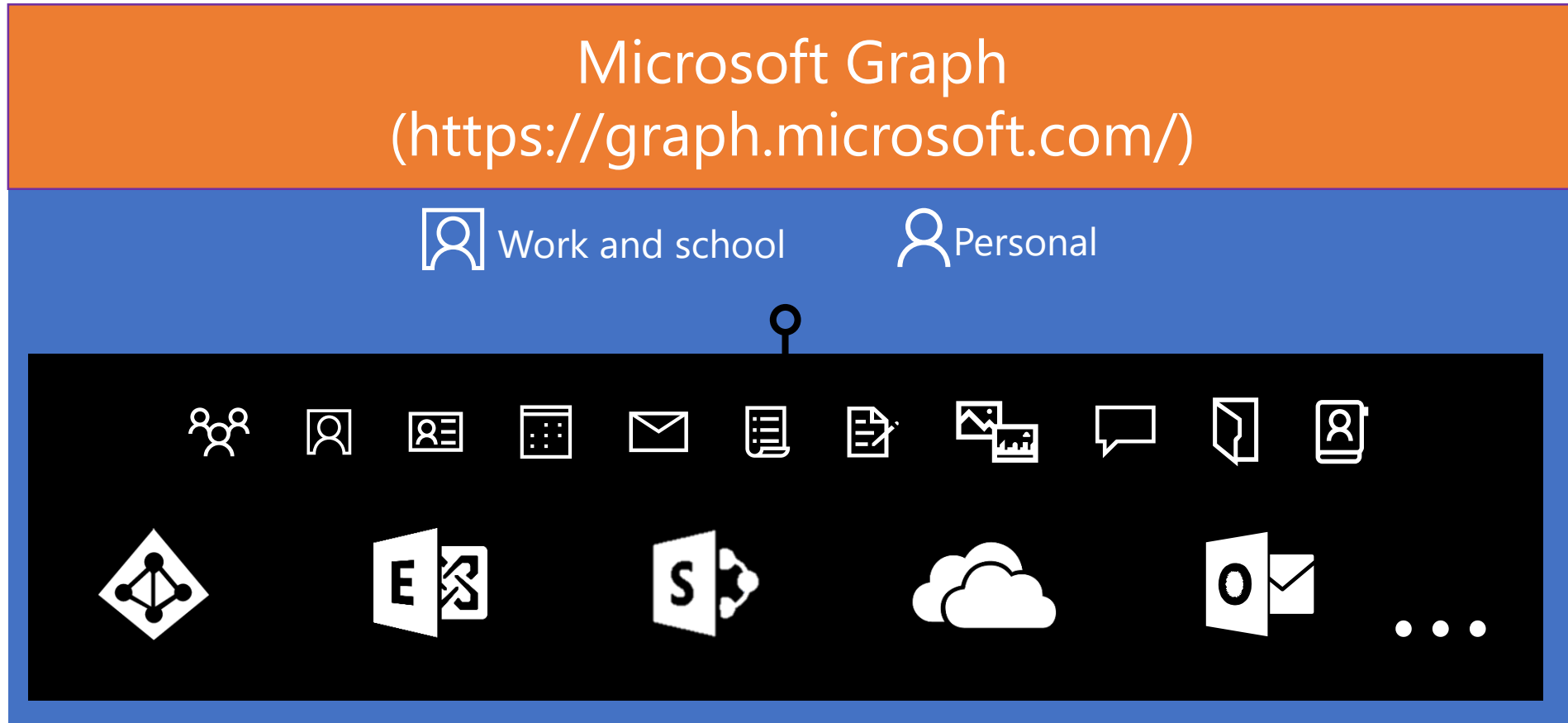


State of the world before Microsoft Graph



- Many different APIs to access data
- Separate auth stacks for work and personal

Today's world with Microsoft Graph



Unified API style – single endpoint

<https://graph.microsoft.com>  

Operation

GET my profile

GET my mail

GET my calendar

GET my contacts

GET my photo

GET my files

GET my manager

GET last user to modify file foo.txt

GET users in my organization

GET group conversations

GET people related to me

GET my tasks

GET my notes

GET files trending around me

Service endpoint

<https://graph.microsoft.com/v1.0/me>

<https://graph.microsoft.com/v1.0/me/messages>

<https://graph.microsoft.com/v1.0/me/calendar>

<https://graph.microsoft.com/v1.0/me/contacts>

[https://graph.microsoft.com/v1.0/me/photo/\\$value](https://graph.microsoft.com/v1.0/me/photo/$value)

<https://graph.microsoft.com/v1.0/me/drive/root/children>

<https://graph.microsoft.com/v1.0/me/manager>

<https://graph.microsoft.com/v1.0/me/drive/root/children/foo.txt/lastModifiedByUser>

<https://graph.microsoft.com/v1.0/users>

<https://graph.microsoft.com/v1.0/groups/<id>/conversations>

<https://graph.microsoft.com/beta/me/people>

<https://graph.microsoft.com/beta/me/tasks>

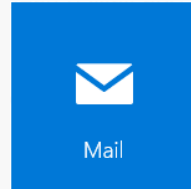
<https://graph.microsoft.com/beta/me/notes/notebooks>

<https://graph.microsoft.com/beta/me/insights/trending>

Outlook Entities & Capabilities on [Graph /v1.0](#)

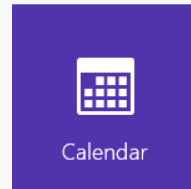
- Mail

- Mail folders
- Messages including Event Message
- Item & file attachments



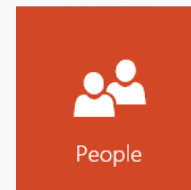
- Calendar

- Calendar groups & Calendars
- Events & Calendar view
- Reminders & Reminder view



- Contacts

- Contact folders & Contacts



- Groups

- Conversations, threads, posts
- Calendar, events



- Profile

- User profile per AAD
- Profile picture



- Mailbox Settings

- Automatic replies
- Language & Time zone



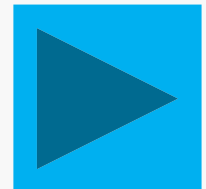
- Data extensibility

- Extensions
- Extended properties



- Capabilities

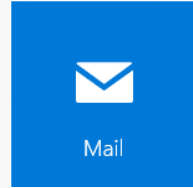
- CRUD
- Webhooks



Additional Outlook Entities on [Graph /beta](#)

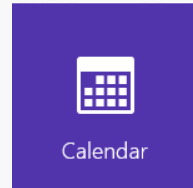
- Mail

- @Mentions
- Event Message Request
- Reference Attachment



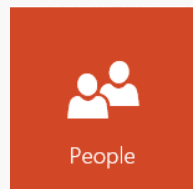
- Calendar

- Find Meeting Times API
- Forward Event API



- People

- Person (read-only)



- Profile

- Profile pics of many sizes



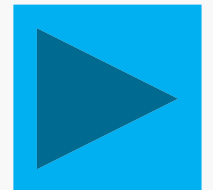
- Mail Tips

- Automatic replies



- Capabilities

- Delta sync

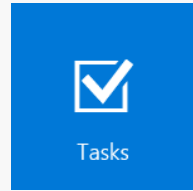


Additional Entities & Capabilities on Outlook API

/v2.0

- Outlook Tasks

- Task groups
- Task folders
- Tasks



- Capabilities

- Sync folder hierarchy
- Delta sync



/beta

- Capabilities

- Streaming notifications
- Batching API when signed in as user



What's Next

APIs Promoted from Beta to Production

- **Microsoft Graph (beta → v1.0)**

- **MailboxSettings API** to get/set a user's automatic reply, time zone and language.
- **Extensions** for message, event, contact and posts.
- **Extended Properties** for message, event, contact, post, mail folder, contact folder & calendar.
- **Outlook task APIs** – Task groups, task folders and tasks
 - Tasks.Read and Tasks.ReadWrite permissions

- **Outlook REST API (beta → v2.0)**

- **Message delta sync APIs** to sync incremental changes to mail in a specific folder.
- **Mail Folder hierarchy sync APIs** to sync mail folder hierarchy.

New APIs Added in Beta

• Microsoft Graph

- **MailTips API** to retrieve mail tips leveraged by outlook, especially to check if a recipient is out of office.
- **@Mentions API** to get/set users @mentioned in a message
- New **EventMessageRequest type** derived from Event Message to represent a message invite. Provides info on changes from an older invite, for updated meeting invites
- **Forward event API** to forward an event to another user

• Outlook REST API

- **Streaming notifications APIs** for client apps that don't have a server endpoint to receive webhooks.

Delegate & Shared Mailbox Access w/ User Consent

• Request example

- Signed in as meganb@contoso.com
- **X-AnchorMailbox:** BenW@Contoso.com
- GET <https://graph.microsoft.com/v1.0/users/BenW@Contoso.com/calendar/events>

• Permissions

- **Mail:** Mail.Read.Shared & Mail.ReadWrite.Shared
- **Calendar:** Calendars.Read.Shared & Calendars.ReadWrite.Shared
- **Contacts:** Contacts.Read.Shared & Contacts.ReadWrite.Shared
- **Tasks:** Tasks.Read.Shared & Tasks.ReadWrite.Shared

• Enumerate list of calendars shared by others with a user

- Signed in as meganb@contoso.com
- **X-AnchorMailbox:** MeganB@Contoso.com
- GET <https://graph.microsoft.com/beta/me/calendars>
- **Sharing properties:** CanShare, CanViewPrivateItems, IsShared, IsSharedWithMe, CanEdit, Owner

Microsoft Graph support Exchange Hybrid

Before Microsoft Graph Supported Hybrid Users

- **Step 1: User provides email address**

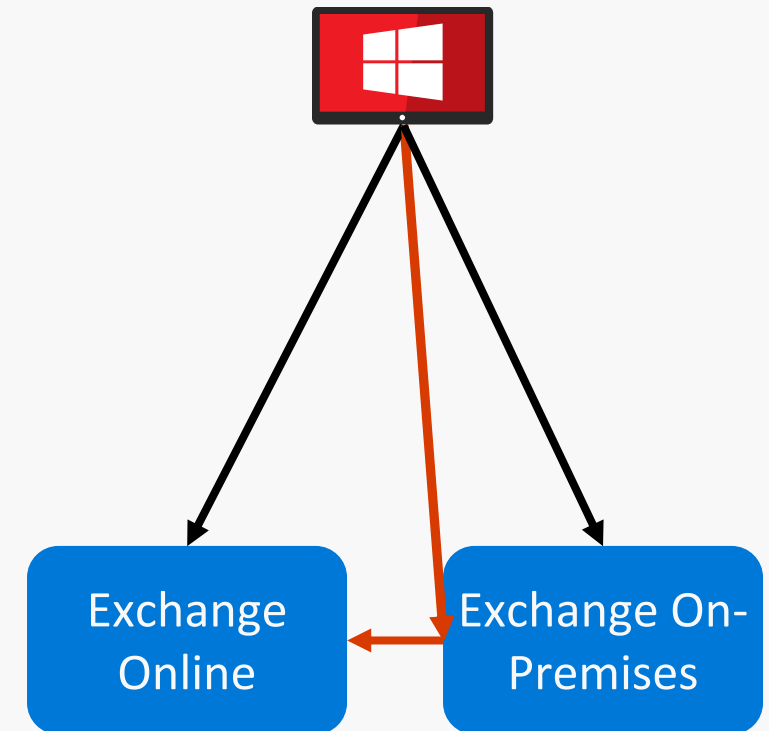
- ... And usually their credentials as well to sign in.

- **Step 2: App determines EWS endpoint**

- User signs into Autodiscover endpoint.
- Autodiscovery is complicated if app isn't built on .NET or Java!
- Need to cache API endpoint, since Autodiscover can take time.

- **Step 3: App uses EWS to connect to mailbox**

- Using Exchange Web Services (EWS) SOAP API from platforms other than .NET and Java is hard!!
- All-or-nothing read/write access to a mailbox.



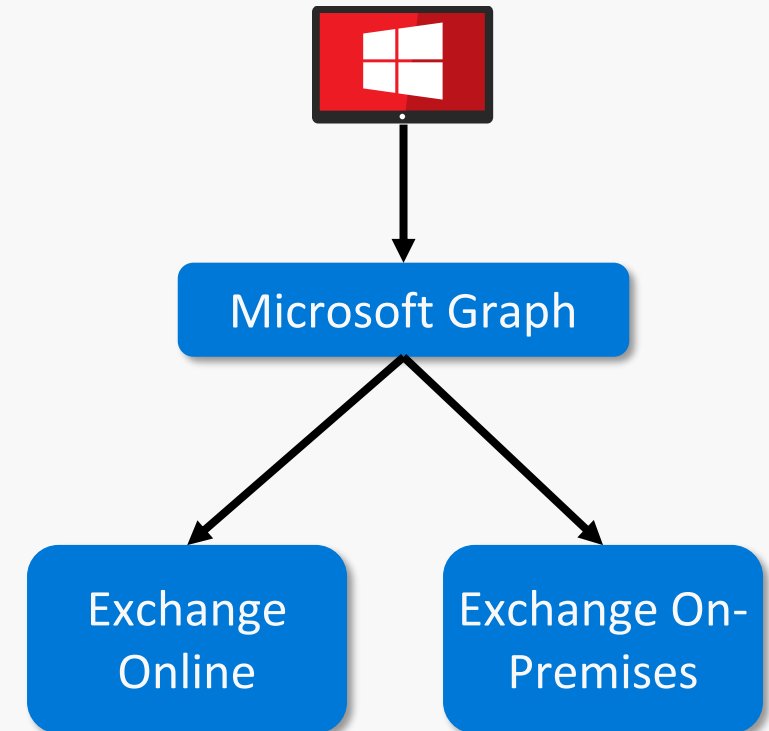
With Microsoft Graph Supporting Hybrid ...

- **Step 1:** User provides email address

- No need to provide app with user credentials.

- **Step 2:** App calls Microsoft Graph

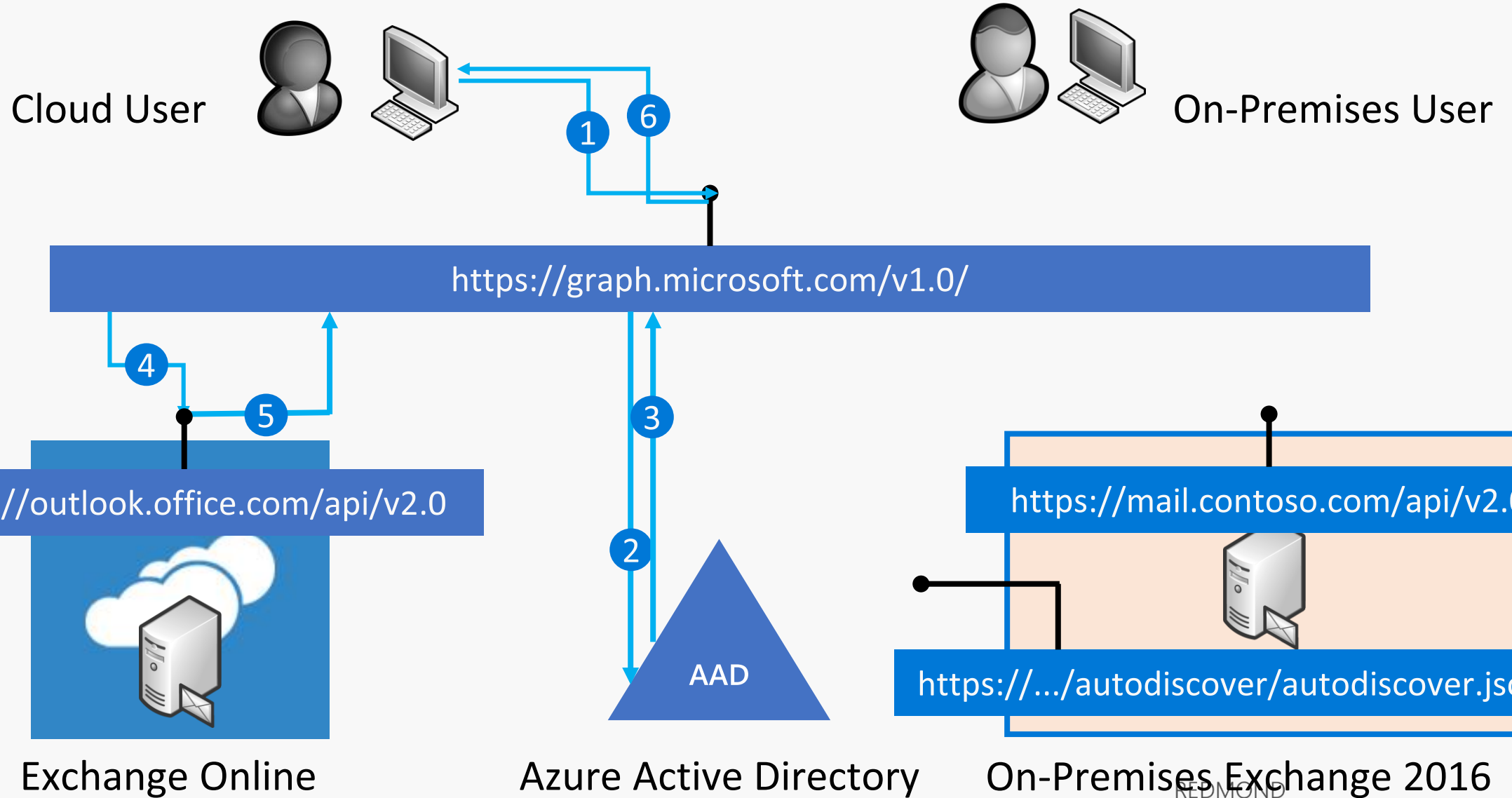
- Microsoft Graph checks if user is in cloud or on-premises.
- User signs in, grants consent and app gets a token.
- App connects to mailbox with easy-to-use REST APIs!
- Granular permission(s) ensure app has access to only what is approved!!



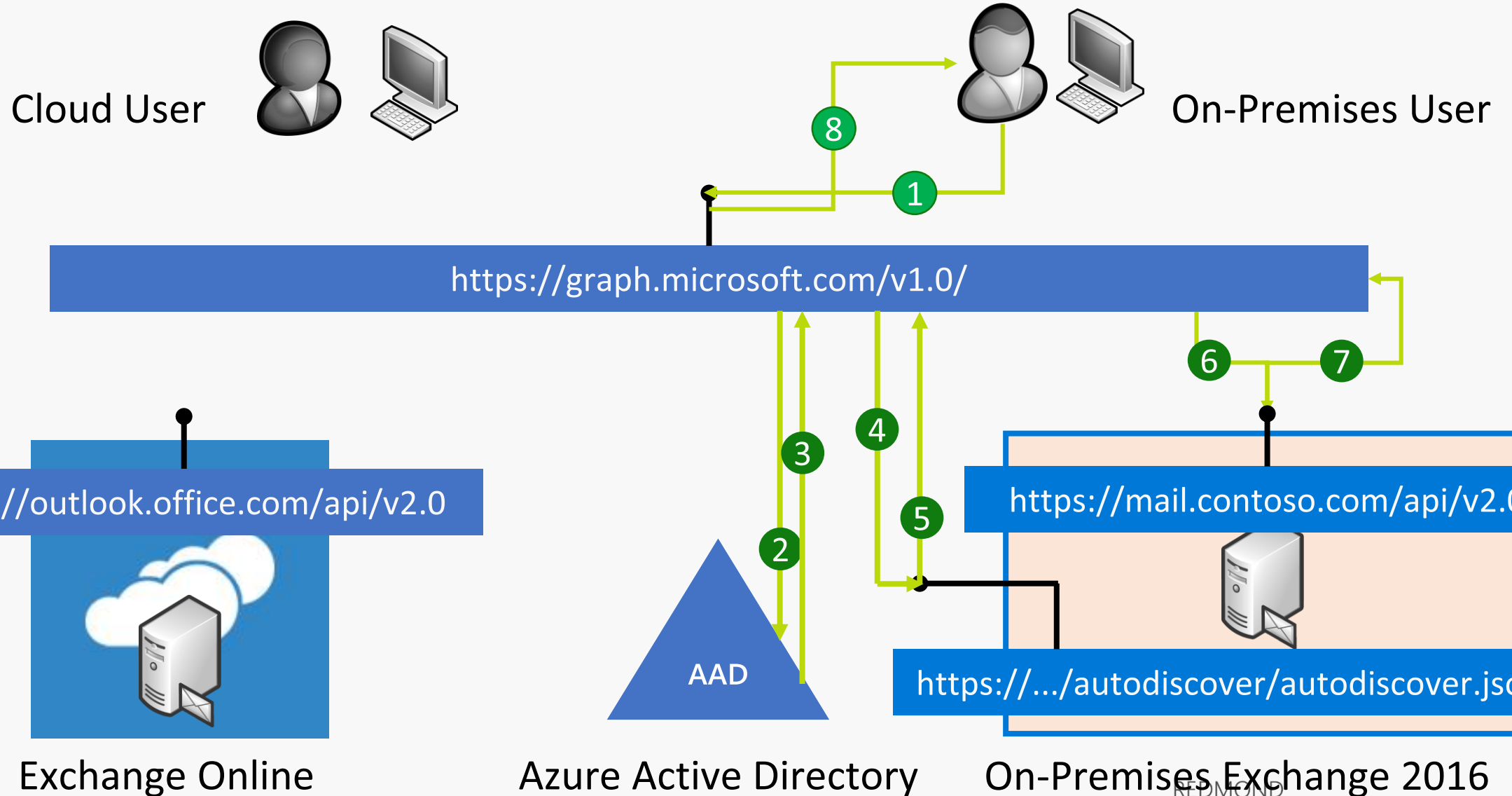
App doesn't need to find out or remember user-specific API endpoints!

Architecture Overview

Cloud User Flow



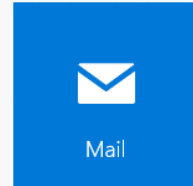
On-Premises User Flow



Graph Entities & Capabilities for On-Premises Users

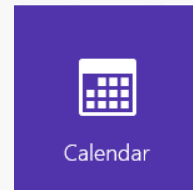
- Mail

- Mail folders
- Messages
- Item & file attachments



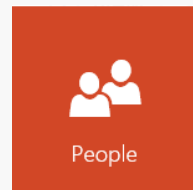
- Calendar

- Calendar groups & Calendars
- Events & Calendar view
- Reminders & Reminder view



- Contacts

- Contact folders
- Contacts



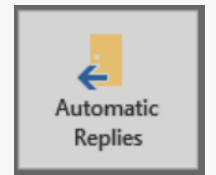
- Profile

- User profile per AAD
- Profile picture



- Mailbox Settings

- Automatic replies
- Language & Time zone



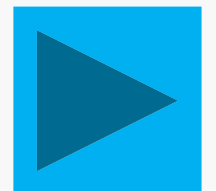
- Data extensibility

- Extended properties



- Capabilities

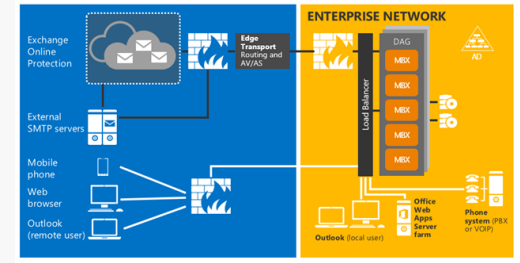
- CRUD



Configuration & Management

Setting up Exchange 2016 for Graph

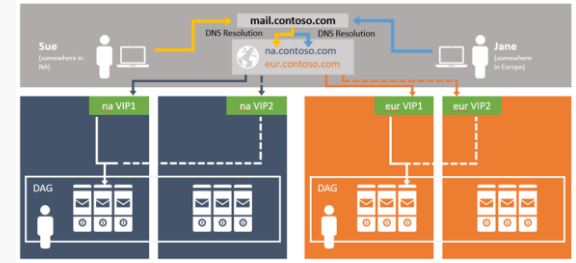
- Upgrade to Exchange 2016 [Cumulative Update 3 \(CU3\)](#).



- Run [/PrepareAD](#) explicitly.
 - Registers Microsoft Graph as a trusted partner application.
 - Admins can disable Graph support by removing it from trusted partner list
- On-premises Active Directory must be fully synced w/ AAD

Network Infrastructure Requirements

- Internet DNS records required for:
 - On-premises Exchange namespace
 - Autodiscover namespace for all accepted domains
- Firewall or application gateway must allow access to:
 - /api virtual directory for REST API requests
 - /autodiscovery/autodiscover.json virtual directory for API endpoint discovery
- Only Exchange 2016 (CU3 or later) servers allowed in load balanced array for Exchange namespace



What's Next

We Need Your Feedback & Suggestions!

- **Management**

- What user experience or service challenges are you facing?
- How hard is this to set up and maintain?
- What additional points of friction are you discovering?
- Are you

- **Developers**

- Are you
- Any feedback
- Are you
- Are there

We need your help to make this work great for you!

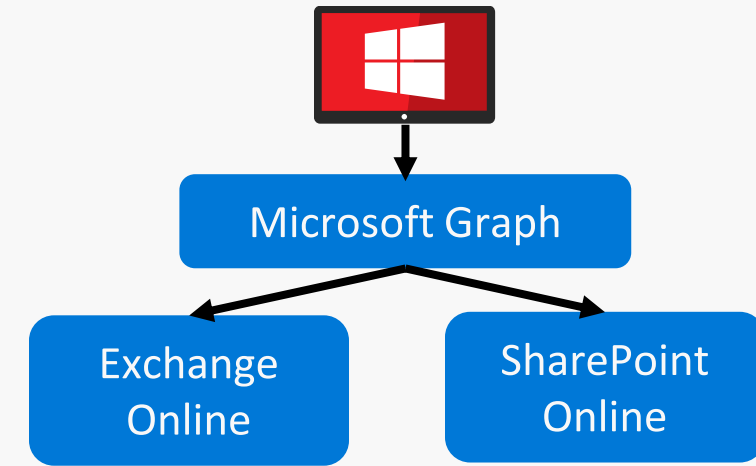
- **What more is needed to blend cloud/on-premises seams?**

Exit Preview for Hybrid Exchange Deployment

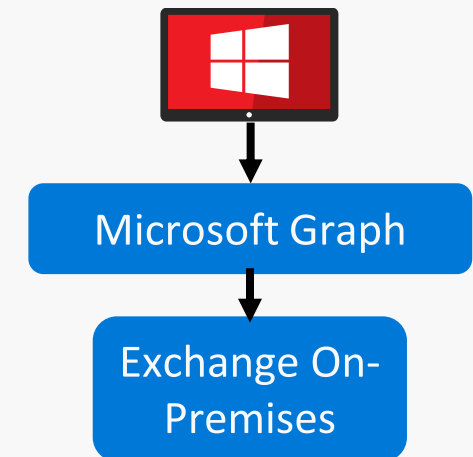
- Hybrid Configuration Wizard (HCW) update
- Querying per-user version & capabilities
- API improvements
 - Ability to use /me instead of /users/<id>/
 - Support for notifications
 - Additional entities for on-premises users e.g. Extensions
 - Microsoft Graph SDK support
 - End user [Graph explorer](#) support for on-premises
- **Incorporate your feedback and suggestions!**

Beyond Hybrid Exchange Deployments

- How do we extend hybrid support to other Office 365 workloads



- How can we enable an on-premises Exchange 2016 deployment to leverage Microsoft Graph?



Resources

- IT administration

- On-premises architectural requirements for REST API:
<https://blogs.technet.microsoft.com/exchange/2016/09/26/on-premises-architectural-requirements-for-the-rest-api>

- Development for hybrid deployments

- Overview: http://graph.microsoft.io/en-us/docs/overview/hybrid_rest_support
- Microsoft Graph documentation: <http://graph.microsoft.io/en-us/docs>

Wrap Up

Build LOB Apps using REST APIs

• Key Takeaways

- **Powerful:** APIs to create powerful applications & services to drive productivity.
- **Secure:** Industry standard OAuth flow that respects your authentication policies.
- **Easy to Use:** REST APIs that are easy to use on any platform you choose.
- **Strategic:** THE APIs to use for Office 365, Hybrid Exchange 2016 & Outlook.com.
- **Production Ready:** Available now to build production apps & services at scale

• We would love to hear from you!

API Endpoint	Area	URL
Microsoft Graph	Dev how-to questions or issues	http://stackoverflow.com/questions/tagged/microsoftgraph
	API doc issues	https://github.com/OfficeDev/microsoft-graph-docs/issues
	User Voice for everything else	https://officespdev.uservoice.com/
Outlook REST API	Dev how-to questions or issues	http://stackoverflow.com/questions/tagged/outlook-restapi
	API doc issues	
	User Voice for everything else	https://officespdev.uservoice.com/

Thank You!

Questions?