What is Azure Active Directory (and Why Should I care)?

Eric Kool-Brown (kool@uw.edu)
Software Engineer UW-IT Identity and Access Management
Presented to the Internet2 2018 Technology Exchange

UNIVERSITY of WASHINGTON

Subtitle: Is it a Desert Topping or an Oven Cleaner?

Microsoft's Office 365 includes Azure AD. Now what?

> Do you need on-premise Active Directory?
> How does identity data get into AAD?
> How does one manage AAD?
> What can AAD do?
> How much does AAD cost?

This talk will attempt to answer these questions and more. I will demonstrate how AAD fits into the identity mix at the UW and discuss the pros and cons of our architecture.

Please hold questions to the end

Agenda

- > Intro to Azure AD
- > Who uses it
- > How is it used
- > Licensing
- > AAD features
- > Graph API
- > OAuth Consent
- > AAD at the UW
- > MFA for AAD
- > AAD Governance
- > Developing for AAD
- > Summary
- > Questions
- > Resources and Glossary

* Personal views and not the views of the University of Washington

Intro – What is Azure AD? (part 1)

- A directory service – storing user and group objects and their attributes
  - AAD objects are accessed via RESTful APIs
  - Other object types are stored to support additional functionality

- A credential store – storing hashed passwords and certificates
  - So that AAD can be a stand-alone IdP and/or do cert auth

Intro – What is Azure AD (part 2)

- An identity provider – authenticating via OIDC, SAML and WS-Fed/WS-Trust
  - No multi-lateral federation
  - Federation via ADFS or other proxies possible
    - (Not straightforward)
  - AAD Business-to-Business (B2B) is not true federation
    - It creates shadow accounts for B2B “guests” but authenticates them using their home IdP
  - AAD Business-to-Consumer (B2C) creates a separate “tenant” to store these accounts

Intro – What is Azure AD (part 3)

- An authorization server (AS) – managing access to resources via OAuth scopes and Azure roles
  - Azure and Office 365 web APIs have AAD scopes
  - Scopes can be combined into roles which can be used by Azure policies
  - Developers can create AAD application objects
    - These are OAuth clients
    - Custom scopes and roles can be defined on them
- A licensing store
- A rapidly evolving IDaaS platform

Intro – AAD is not AD

Active Directory

- An LDAP directory
  - LDAP API and auth
  - Hierarchical namespace
  - Extensible LDAP schema
- Kerberos authentication
  - And NTLM
- Computer joining
  - Group Policy

Azure Active Directory

- Not an LDAP directory
  - No LDAP API
  - No hierarchical namespace
  - No LDAP schema
- No Kerberos auth
  - No NTLM (yay!)
- AAD device joining
  - Device Management (MDM) via InTune

Intro – More on AAD

> Limited support for custom attributes

> Syncing custom AD attributes to AAD not simple

AAD underpinnings

- Based on AD-LDS, modified for the usage and scaling
- Internally, it is called MSODS – Microsoft Operational Data Store
- Core of AD is the Jet database with an LDAP head above that DB
  - It would be possible to put a non-LDAP head above the Jet DB (but I don't know the details)
- Several Office workloads (EO, SPO) have shadow directories that have been extended with their own attribute needs
  - Back-end sync processes move changes from the master AAD to those shadow directories.

Who Uses AAD?

- Office 365 apps (online and thick client versions of Outlook, Word, Excel, SharePoint, etc.) use AAD for user authentication
- Azure workloads (VMs and other Azure services) can use AAD for autheNticateion/authoriZation
- Custom applications that need enterprise authN/authZ and identity information
- Third party “Gallery” apps use AAD, e.g. Salesforce

How Is It Used – 2 Basic Modes

- Stand-alone: all accounts created directly in AAD
  - Provision to AAD from Workday or using the Graph API
- Synced from on-premise AD
  - On-premise AD is the master but you can configure sync-back for changes made in AAD
  - AD sync has two modes of operation: password hash sync or no password hash sync
  - If password hashes are synced from AD, then authN can be done entirely in AAD
  - If password hashes are not synced, then federation must be configured to allow AAD to use an external IdP

AAD Licensing

- AAD stores user license assignments
  - Includes the licenses to use AAD and Office 365 features
  - Many advanced AAD features require a high level of licensing for all your AAD users
- An Office 365 license includes a basic AAD license
  - Covers the standard set of AAD features e.g. user authN/authZ
- MS licensing is complex and constantly changing

Licensing Levels

	FREE	BASIC	PREMIUM P1	PREMIUM P2	OFFICE 365 APPS
Common Features					
Directory Objects[1]	500,000 Object Limit	No Object Limit	No Object Limit	No Object Limit	No Object Limit
User/Group Management (add/update/delete)/ User-based provisioning, Device registration	✔	✔	✔	✔	✔
Single Sign-On (SSO)	10 apps per user[2] (pre-integrated SaaS and developer-integrated apps)	10 apps per user[2] (free tier + Application proxy apps)	No Limit (free, Basic tiers + Self-Service App Integration templates[5])	No Limit (free, Basic tiers + Self-Service App Integration templates[5])	10 apps per user[2] (pre-integrated SaaS and developer-integrated apps)
B2B Collaboration[7]	✔	✔	✔	✔	✔

From https://azure.microsoft.com/en-us/pricing/details/active-directory/ (only the top of a very long page)

AAD Features (part 1)

A wide variety of standard and optional features are available based on your level of licensing

- Tenant isolation: each AAD/O365 organization has a separate DNS namespace and entity ID (basic license)
  - DNS namespaces form the set of allowable UPN suffixes
  - Every Azure subscription must be bound to an AAD tenant
    - Multiple subscriptions can be bound to the same tenant
    - Allows you to segregate your Azure usage into different expense buckets

AAD Features (part 2)

- Conditional access
  - Set rules for what and how resources are accessed
  - MFA requires conditional access (P1 license for those users)
- Azure Identity Protection (AIP)
  - Machine learning is used to analyze access patterns such that unusual patterns can be flagged as suspicious (P2 license for all users)
- Reporting and auditing
  - Reports on activity and access can be viewed through both the GUI and via RESTful web API calls
  - More advanced reports require P1 licensing

AAD Features (part 3)

- Application publishing: develop an application and make it available to be used by any and all Azure/O365 users (basic license)
  - An Azure app is the anchor object for an OAuth client
    - It defines the client ID and the client secret
  - The app can be limited to your tenant or can be published in the app gallery for any tenant to use
  - Conditional access can be used to limit who has access to an application
    - E.g. only members of a specific group (P1 license for those users)

AAD Features (part 4)

- Device authentication
  - Devices can be "joined" to AAD to provide a higher level of assurance for user authN (premium license for some flavors)
  - This is a certificate-based process with the device's private key stored in its TPM (if it has one)
  - E.g. via conditional access, don't require MFA if logging in from a joined/trusted device
- Device management: Intune MDM (P1 license)
  - Join devices to AAD and manage the devices including configuration and remote wipe

AAD Features (part 5)

- AAD Domain Services
  - Provides LDAP, machine join, Kerberos, NTLM and Group Policy
  - It is not full AD; you are limited in what you can do
    - No schema modification
  - AD join Azure VMs so they can use Windows Integrated Auth
    - Use an Azure Virtual Network for the VMs and the AAD DS so that those ports are not wide open to the Internet
  - Open LDAPS (port 636) to the public internet for use by SaaS apps
  - The licensing cost is per AAD DS user/group account

(continued)

AAD Features (part 5 continued)

- AAD Domain Services
  - If your AAD is federated with your local AD then you must have AAD password hash sync enabled
  - The AAD DS domain is a stand-alone domain
    - There is no trust from it to your on-prem AD (but it does have SID history)
  - No domain admin privileges
    - It is a fully managed instance of AD

AAD Features (part 6)

- Business to Business (B2B) – not really federation
  - Creates shadow accounts for "guest" users
  - Defers account management and authN to the guest's IdP
  - Guests must be invited either interactively or programmatically – it isn't a formal IdP-to-IdP relationship
- Business to Consumer (B2C)
  - Creates separate "tenant" for you to hold consumer accounts you create (or that customers create themselves using your custom web app)
    - Effectively an IdP-of-last-resort
  - Can employ other IdPs such as Google and FB for authn

AAD Features (part 7)

- App Proxy
  - A service that allows AD-joined machines to use their AAD login token to be exchanged for a Kerberos service ticket
  - This extends OIDC SSO to Windows Integrated Authentication
  - Requires a “connector” server in your on-prem data center
- Privileged Identity Management (PIM)
  - Monitor, audit, and JIT approve use of roles that convey elevated access
- Group Management
  - Three type of groups – synced from AD, AAD native, and Office groups

Graph API

RESTful web API CRUD access to AAD and Office 365

> Two variations

- **Azure Graph** - the original, only manages AAD, reasonably comprehensive
- **Microsoft Graph** - manage both AAD and Office 365 workloads, not yet up to par with the Azure Graph WRT AAD

> Both Use

- "industry standard" **O-Data query language** but only implements a subset of the functionality
- **OAuth authentication** and its authorization scopes

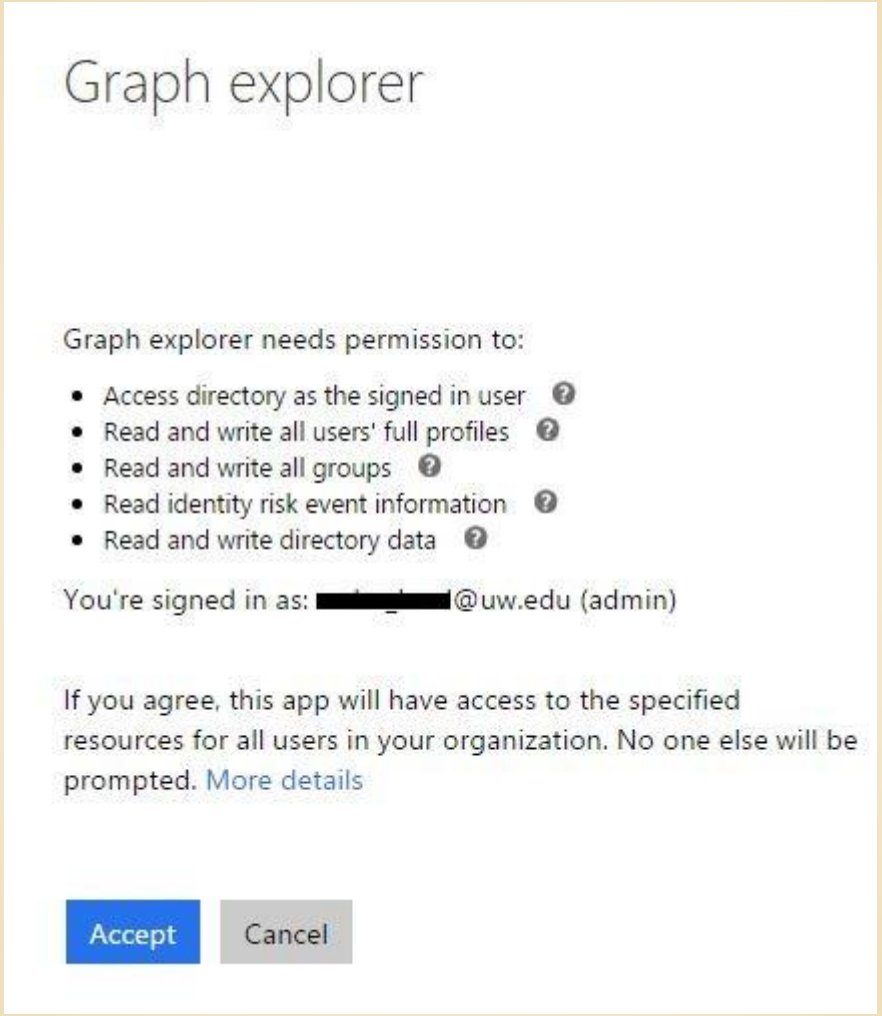
E.g. https://graph.windows.net/uw.edu/users/kool@uw.edu

OAuth Consent in AAD

- AAD as an OAuth AS also manages user consent
- It will prompt for consent on first use
  - For API access, consent must be granted through the Admin Portal
- It saves the response
- You can query the state of user consent
  - E.g. you can ask “what consent has user X granted?”
- Consent can be either per-user or admin consent for all users

Examples of consent screens follows

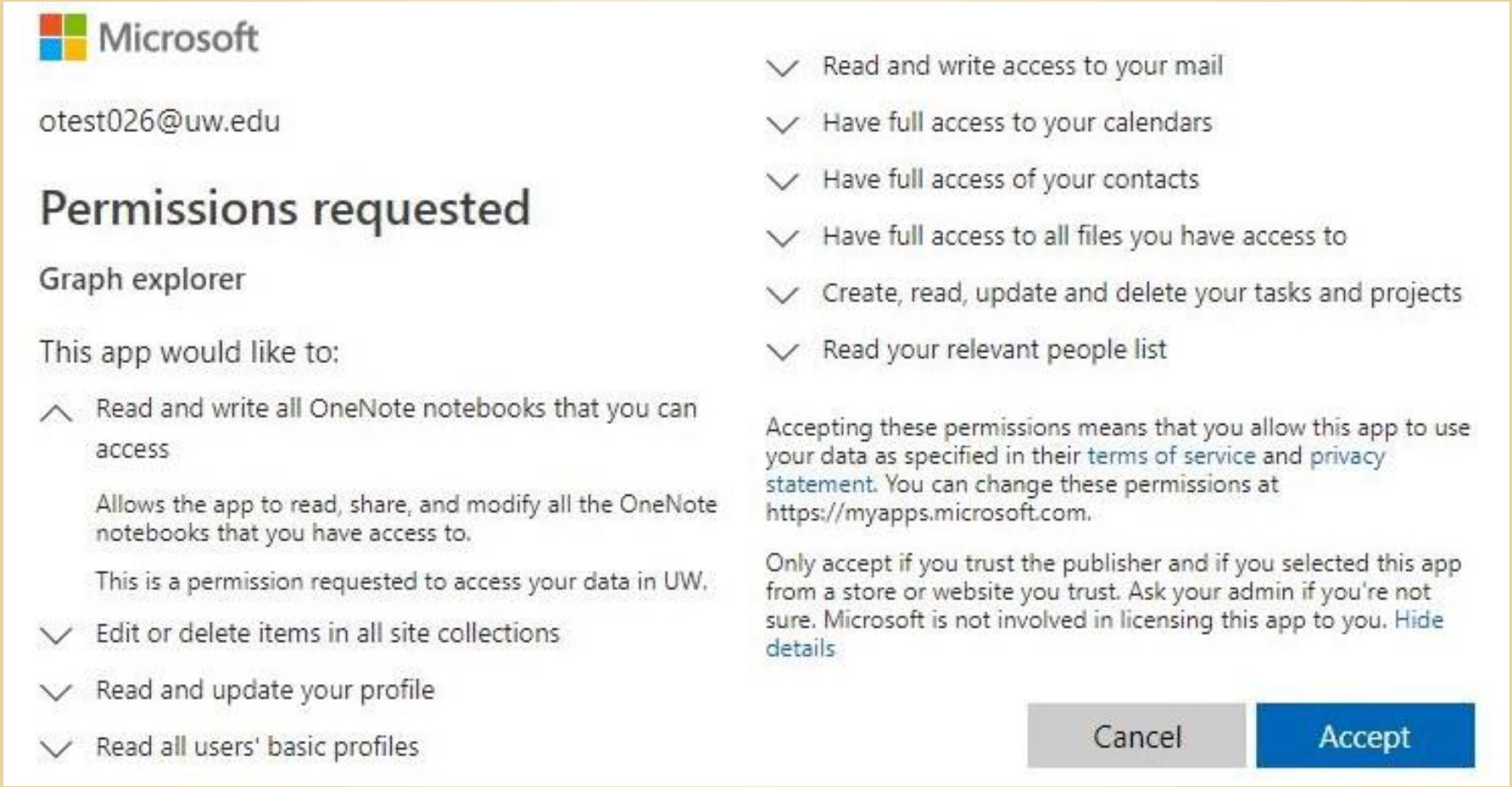
Azure Graph API Explorer Consent Dialog



This is a fairly old consent UI

UNIVERSITY of WASHINGTON

MS Graph Explorer Consent Dialog – recent version



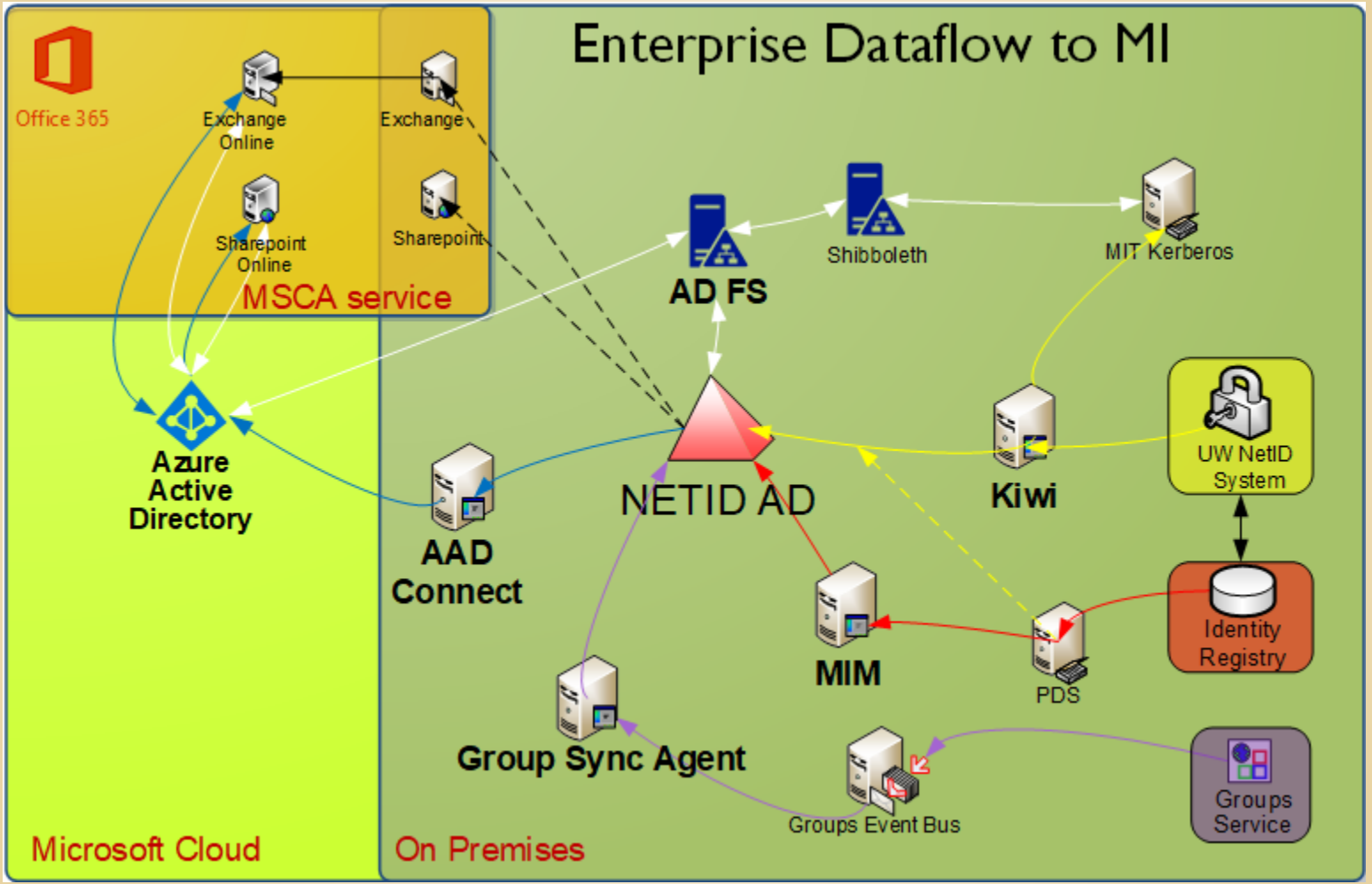
UNIVERSITY of WASHINGTON

AAD at the UW

UW NetID system provides identities to Linux, Main-frames, and Windows

- User Provisioning
  - SOR -> ID-Registry -> OpenLDAP/MIT-Kerberos -> AD -> AAD
    - UW AD provisioning via homebrew pub/sub system
    - AAD Connect used to sync AD changes to AAD
- Group Provisioning
  - Grouper -> AD -> AAD
    - Grouper changes posted to AWS event queue
    - A process listens for those events and updates AD
- Office/Azure Authentication
  - AAD -> ADFS -> Shibboleth

AAD at the UW Graphically



UNIVERSITY of WASHINGTON

AAD at the UW continued

The authentication flow is complicated with a lot of hops

> Upgrading ADFS to 4.0 was a huge undertaking
  - ADFS 4.0 removed features we had been using requiring us to engineer clumsy work-arounds
  - We had to modify Shibboleth to accept a non-standard SAML AuthnContextClassRef

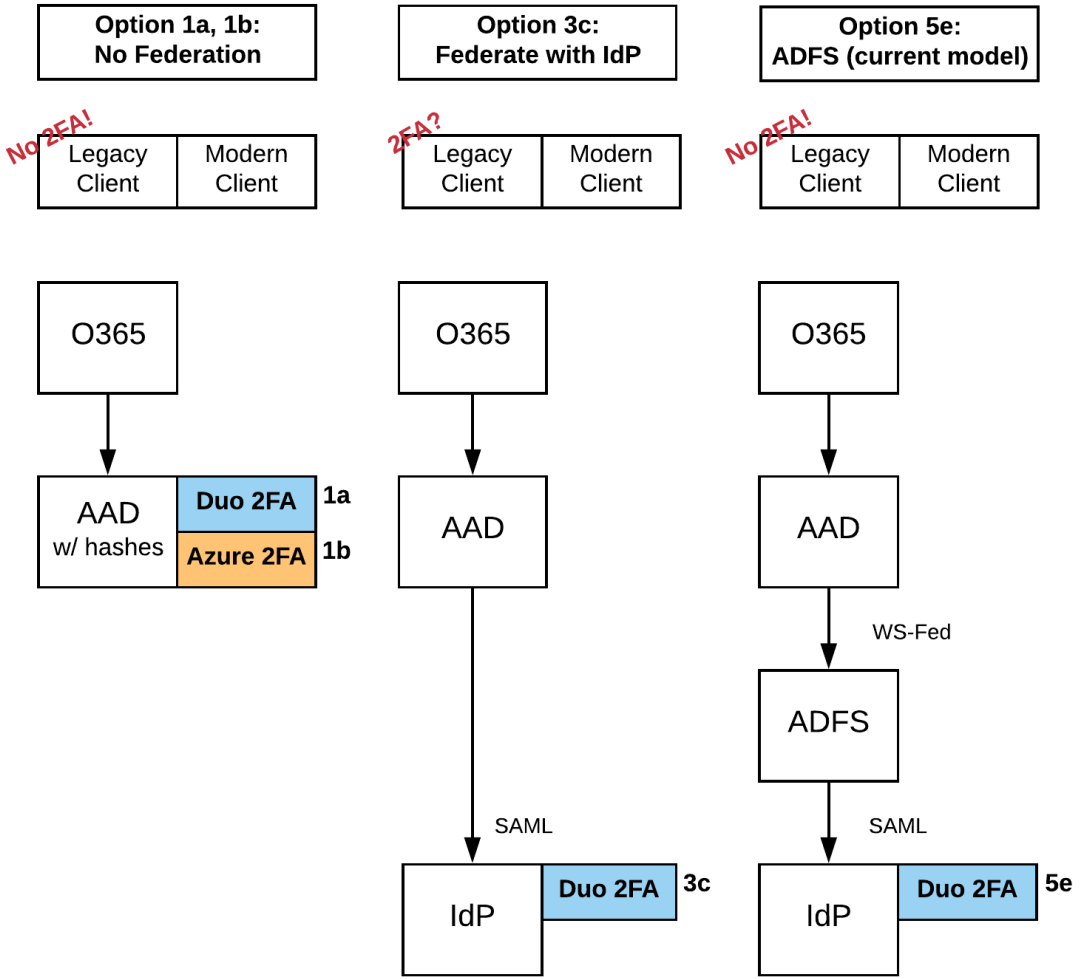
> Office 365 “Modern Auth” can break in many ways
  - Different versions of the Office thick clients have different auth behavior (2016 C2R vs. 2016 MSI vs. 2013)
  - Fiddler can be necessary to figure out what is going on

AAD at the UW – MFA

- UW currently using Duo with Shibboleth IdP
- Lots of options for AAD MFA, none simple or inexpensive
  - We've launched an analysis project
  - Options table with multiple rows and columns
    - Duo vs. AAD MFA
    - Duo in AAD vs. Duo in ADFS vs. Duo in Shib
    - PW hash sync to AAD and AAD MFA would be the simplest but there would be two different user experiences for login and MFA
    - Legacy clients are problematic (app passwords?)

MFA with AAD

These are the finalist options out of the 14 permutations that were initially identified



AAD Governance

- Inadequate Technical Controls – examples:
  - No group member privacy
  - Poor group naming control
  - Poor object ownership and lifecycle management
  - Misbehaving/compromised user accounts
  - Cumbersome e-discovery mechanisms
- Involve your data custodians and stakeholders
  - Create technical controls where possible
  - Create policy when necessary

AAD Futures at the UW

- Password hash sync
  - **Pro:** Would simplify the login flow (no ADFS or Shib)
  - **Pro:** More signal intelligence for AIP
  - **Con:** Can't use AAD self-service password reset
    - no simple way to reverse sync the new password to our NetID system
  - **Con:** User education and phishing – two different login experiences
  - Neutral: Required for AAD Domain Services
- Leveraging the Azure Platform
  - UW apps Azure hosted and authenticated via OIDC
  - UW web services using Azure OAuth versus using the new Shib or other AS/OP?
  - Hybrid networking by connecting the UW net to Azure (TBD)

Developing for AAD

- Building a Visual Studio IIS web site that uses AAD OIDC/OAuth is drop-dead simple
- It is “standard” OIDC/OAuth such that libraries for other languages should work
  - MS folks on the OpenID Foundation working groups may help ensure that the MS implementations adhere to the emerging profiles and their conformance tests
- I built a monitoring app that downloads audit events and other Graph objects
  - At the time there was limited library support, but it wasn't hard to code directly

Summary

Azure Active Directory is:

- Capable and complex (almost dizzyingly so!)
- Maturing but not mature (especially the documentation)
- Being enhanced on a rapid cadence
- Not cheap and can be quite expensive
  - > The security features being P1/P2 is troubling
- Mostly standards compliant
- MS recognizes the security weaknesses of its legacy protocols and is moving to an all-new model with web-friendly APIs and strong public-key-based processes

Questions?

Resources

- Doc entry point: https://docs.microsoft.com/en-us/azure/active-directory/
- Azure Licensing: https://azure.microsoft.com/en-us/pricing/details/active-directory/
- Workday provisioning: https://docs.microsoft.com/en-us/azure/active-directory/saas-apps/workday-inbound-tutorial
- UW AAD Architecture Diagram: https://itconnect.uw.edu/wares/msinf/design/arch/
- UW MFA Analysis: https://wiki.cac.washington.edu/x/pppABQ
- UW Group Sync code: https://bitbucket.org/uwitiam/group-sync
- EKB blog on cert auth for services: https://blogs.uw.edu/kool/

Glossary

- OAuth = a web-friendly authorization protocol
  - AS = Authorization Server, the server that issues OAuth tokens
  - Client = the web app/API that is protected by OAuth
  - Client ID and secret = the credentials of an OAuth client
- OIDC = OpenID Connect, an authentication protocol built on OAuth
  - OP = OIDC Provider, the OIDC equivalent of an IdP
- Modern Auth = Microsoft's term for OAuth/OIDC
- WS-Federation, WS-Trust = the protocols used by ADFS
- ADFS = Active Directory Federation Services, a locally run service that enables federated authentication
- NTLM = a very old and very insecure authentication protocol
- Microsoft Organizational Accounts (org accounts) – in an AAD tenant
- Microsoft Consumer Accounts – from Hotmail, Outlook, Live, a separate tenant
- JIT = Just In Time – a short term conveyance of privileges