Microsoft

Azure Fundamentals

Identity

Abstract and learning objectives

In this whiteboard design session, you will learn how to implement different components of a hybrid identity solution that integrates an Active Directory forest with an Azure Active Directory tenant and leverages a number of Azure Active Directory features, including pass-through authentication with Seamless Single Sign-On, Multi-Factor Authentication, Self-Service Password Reset, Azure AD Password Protection for Windows Server Active Directory, Hybrid Azure AD join, Windows Hello for Business, Microsoft Intune automatic enrollment, Azure AD Conditional Access, Azure AD Application Proxy, Azure AD B2B, and Azure AD B2C.

Step 1: Review the customer case study

Outcome

Analyze your customer needs.

Timeframe

15 minutes

Customer situation: Contoso

- A medium size financial services company:
  - headquarters in New York
  - branch office in San Francisco
  - operating on-premises, with infrastructure running on the Windows platform
- Contoso is facing challenges related to increased workforce mobility:
  - Contoso management is concerned about property costs:
    - considering implementing a flexible work arrangement policy to allow employees to work from home, using either corporate- and employee-owned devices
  - Contoso's Information Security is concerned about:
    - lack of controls that would prevent access from unauthorized or non-compliant systems
    - using traditional VPN technologies or DirectAccess, which provide excessive access to on-premises infrastructure

Current environment: Contoso

- A single domain Active Directory forest implemented over a decade ago
- The AD domain uses a non-routable DNS name contoso.local
  - Directory Services did not implement domain name change due to potential negative implications of such change
  - Contoso owns a publicly routable DNS domain name contoso.com
- The AD domain has been recently upgraded to Windows Server 2016
- Contoso is in the process of migrating desktops from Windows 7 to Windows 10
- Majority of servers are running either Windows Server 2012 R2 or 2016

Customer objectives

- Transition operations into an Internet-open model:
  - support for mobile workforce and integration with business partners,
  - support for current security and manageability controls
    - given current dependency on Active Directory and migration to Windows 10 devices, Contoso intends to evaluate Azure Active Directory and Microsoft Intune as identity and management solutions
- Implement new identity capabilities:
  - step-up authentication
  - per-application permissions based on the properties of users' accounts and the state of these users' devices
  - minimized or even eliminated persistent assignments of privileged roles for identity management (while accounting for break-glass scenarios, allowing for a non-gated emergency use of privileged accounts)
  - monitoring and auditing of privileged accounts

Customer objectives (continued)

- Minimize the use of passwords in lieu of more secure authentication methods.
- In situations where passwords are required, users should also be able to both change and reset them without having to rely on HelpDesk services.
- Honor any on-premises Active Directory user account restrictions and settings:
  - allowed sign-in hours
  - expired accounts
- Preserve existing Active Directory password policies:
  - In addition, Information Security wants to prevent the use of common terms in passwords
- Optimize end-user experience, especially in environment where users might be using several different devices.
  - User-defined settings, such as accessibility or app customization should be automatically synchronized across all devices.

Customer objectives (continued)

- Expand customer base through partnership with other financial institutions:
  - Contoso established business relationship with Fabrikam
  - Fabrikam manages an extensive portfolio of mortgage related products
  - Contoso intends to provide Fabrikam with access to its internal WIA-based web applications to facilitate integration with the existing Fabrikam's products
  - Fabrikam has already moved its operations almost entirely to Azure
- Provide direct access to its services to external clients:
  - Contoso started developing web and mobile client apps
  - The effort of managing customer identities should be minimized

Customer objectives (continued)

- Maximize resiliency and SLAs
  - Emphasized by the management team of Contoso, including its CIO, Andrew Cross
- Minimize infrastructure requirements

Customer needs

- Remote users must be able to sign into their devices by using their AD credentials.
- Existing AD user sign-in hours and password policies must be preserved (although password complexity can be further increased).
- User sign-in experience should be optimized by:
  - minimizing the number of sign-in prompts
  - limiting the use passwords in lieu of more secure authentication methods
- Users device configuration should be simplified by:
  - leveraging a mobile device management solution
  - roaming user-specific settings across multiple devices
- Access of users to applications and resources should be based on:
  - users group membership
  - state of the users devices
  - dynamically evaluated risk based on comprehensive security-related telemetry

Customer needs (continued)

- Users must be allowed to reset their own passwords.
- Designated users should be able to temporarily elevate their privileges to manage other user accounts. All elevation events must be edited.
- Contoso remote users must be able to access on-premises WIA-based applications.
- Fabrikam users must be able to access on-premises WIA-based applications.
- Apps developed by Contoso must be made available to external customers with minimum overhead associated with identity management.
- Resiliency must be maximized whenever possible.
- Infrastructure requirements must be minimized.

Customer objections

- Our Active Directory domain is using a non-routable domain name. We cannot risk renaming it in order to implement single sign-on with Azure Active Directory.

- We have heard that it is not possible to run simultaneously multiple instance of Azure AD Connect. All identity services components in our environment must provide resiliency and support failover.

- If we decide to integrate our Active Directory environment with Azure Active Directory, this must be performed in stages. This is likely to be complex, considering that users in each stage would be members of different Active Directory groups and their accounts might reside in different Active Directory organizational units.

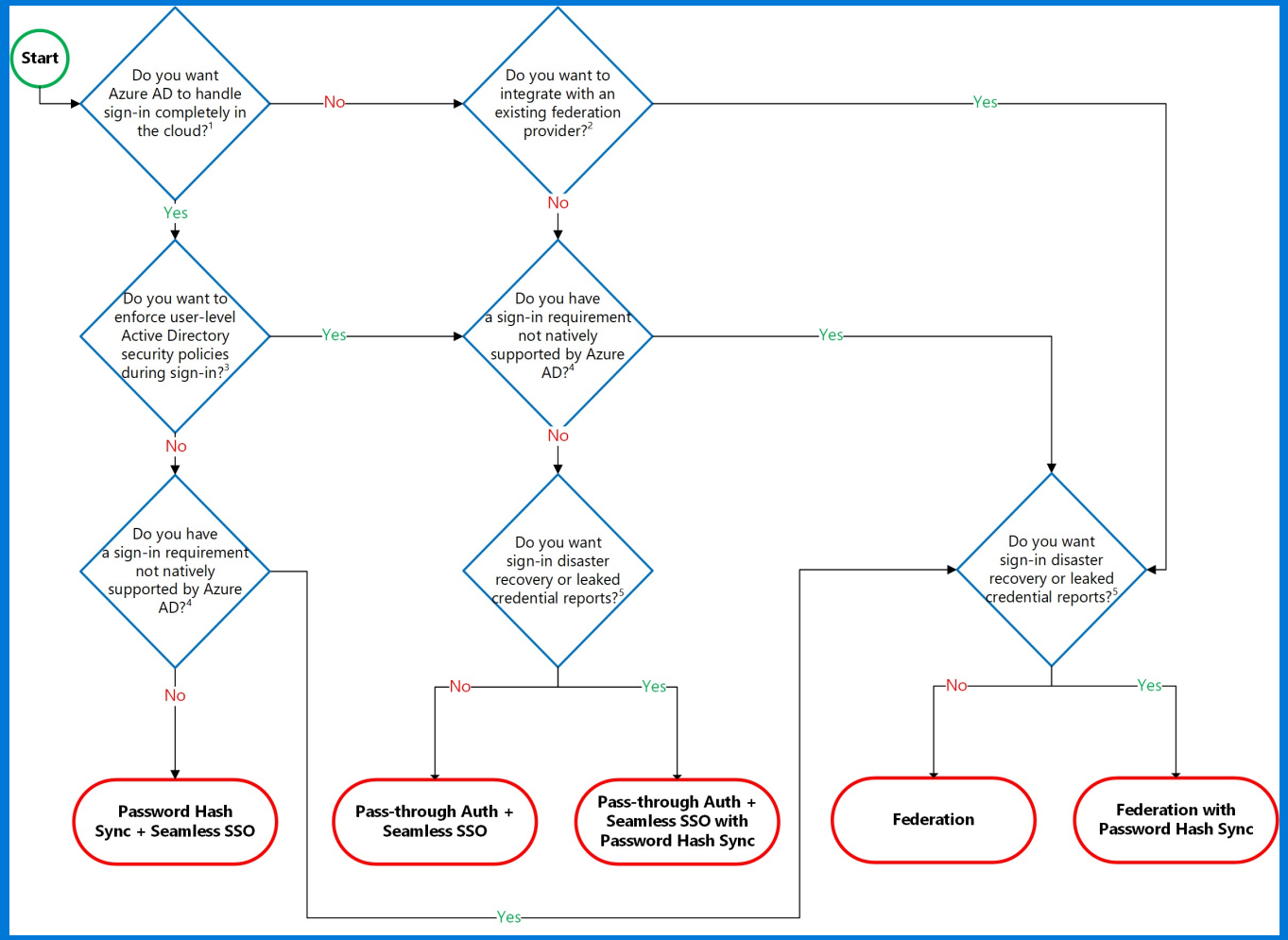
Customer objections (continued)

- Synchronizing our Active Directory accounts with Azure AD accounts makes the former vulnerable to malicious or accidental lockouts that affect the latter. This would effectively expose our on-premises environment to external attacks.

- A number of critical web applications running in our on-premises environment rely on Kerberos-based Windows Integrated Authentication. Microsoft states that Azure Active Directory does not support Kerberos. Doesn't this mean that remote users authenticating to Azure Active Directory and our business partners will not be able to properly authenticate and access these applications?

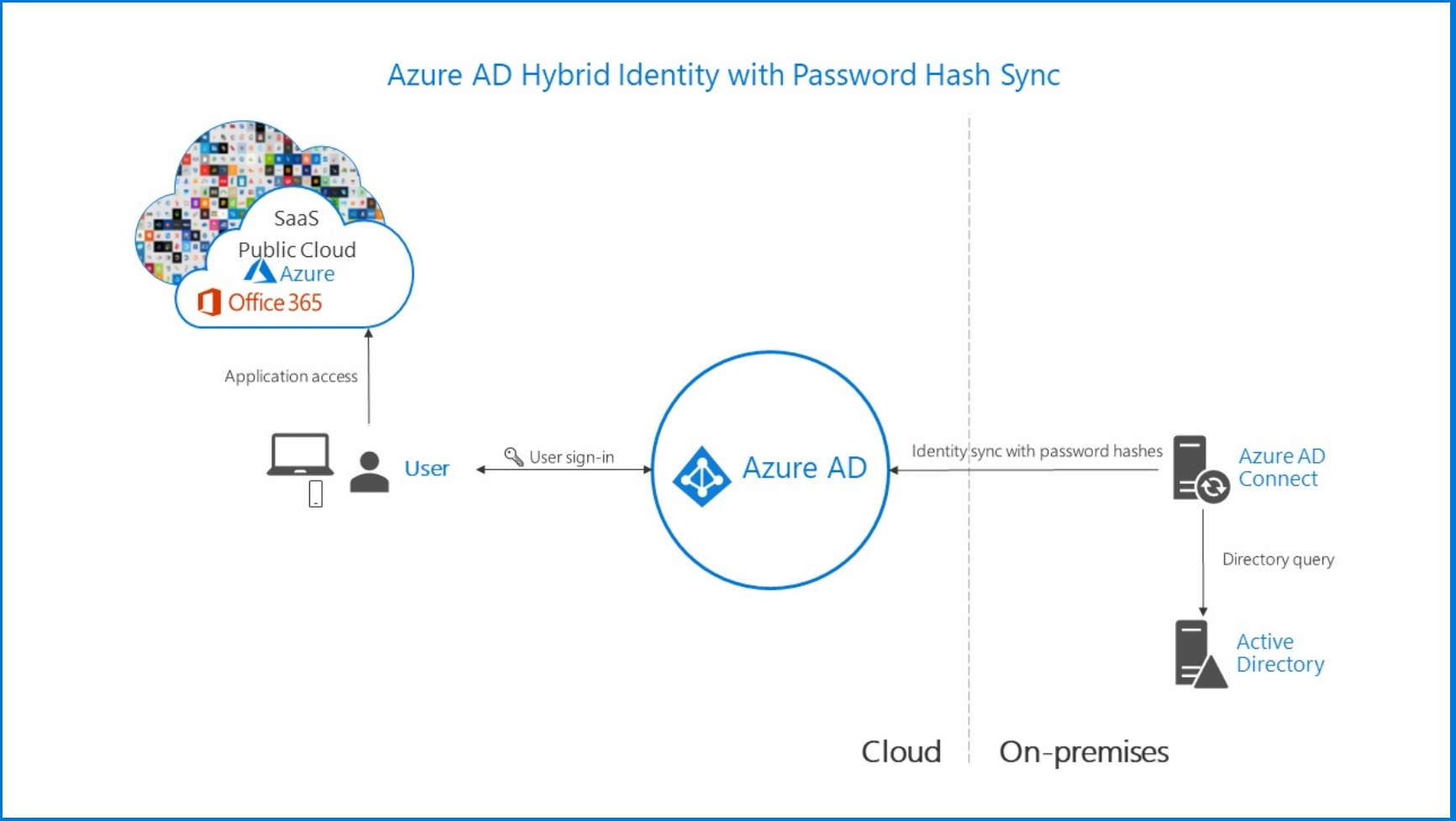
Common scenarios

Hybrid identity authentication decision workflow:



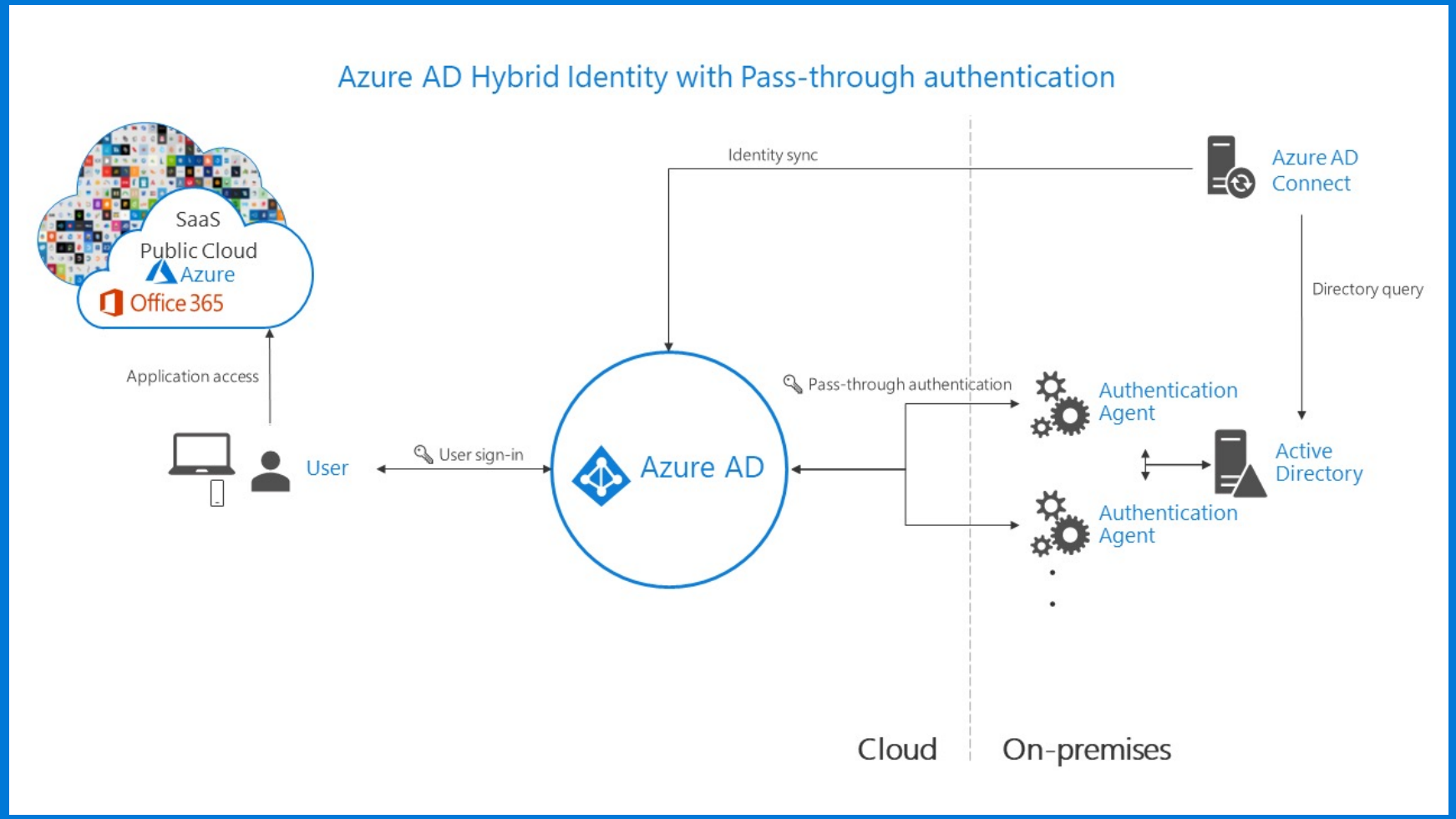
Common scenarios

Azure AD hybrid identity password-hash synchronization



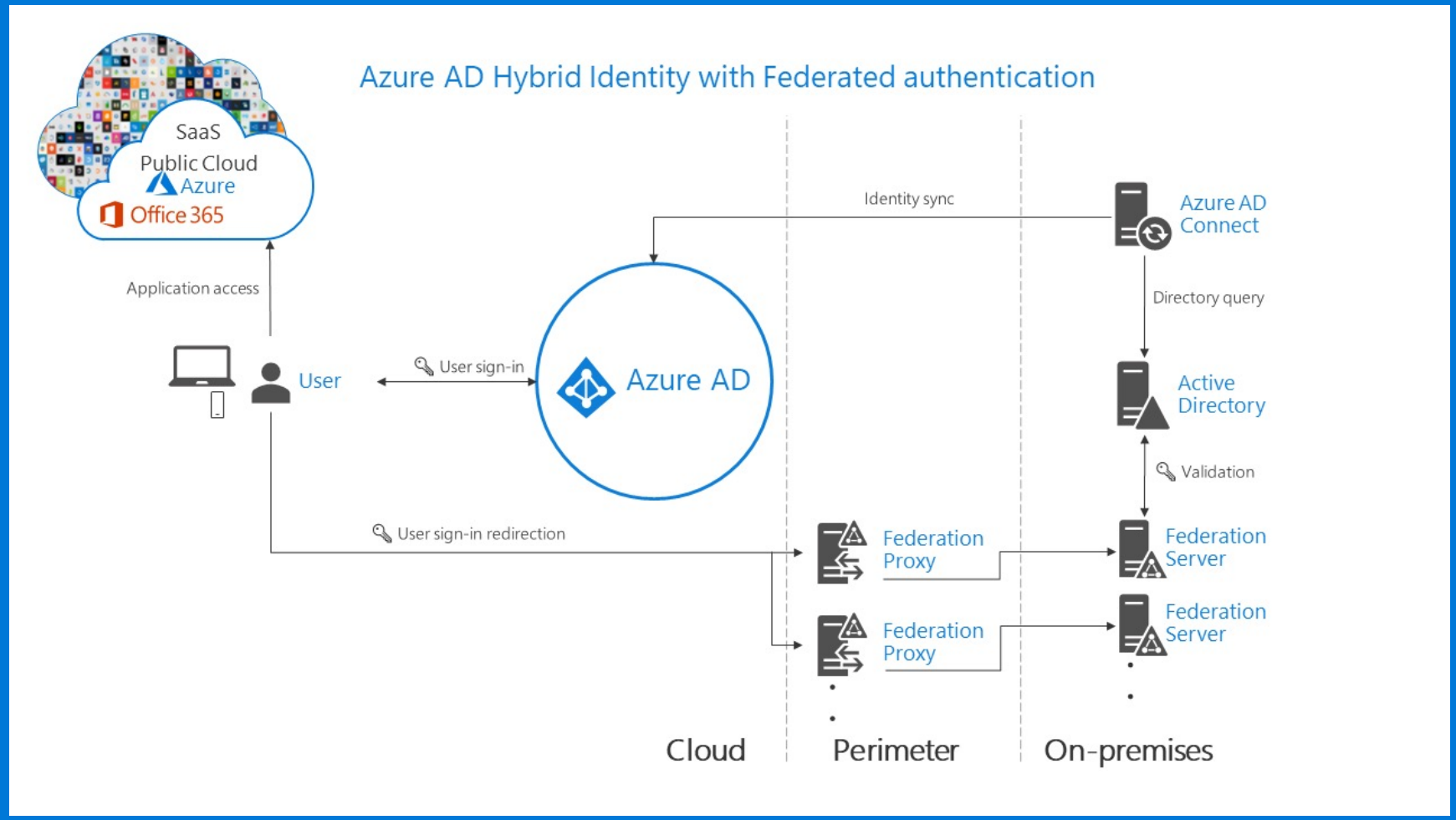
Common scenarios

Azure AD hybrid identity pass-through authentication



Common scenarios

Azure AD hybrid identity federation



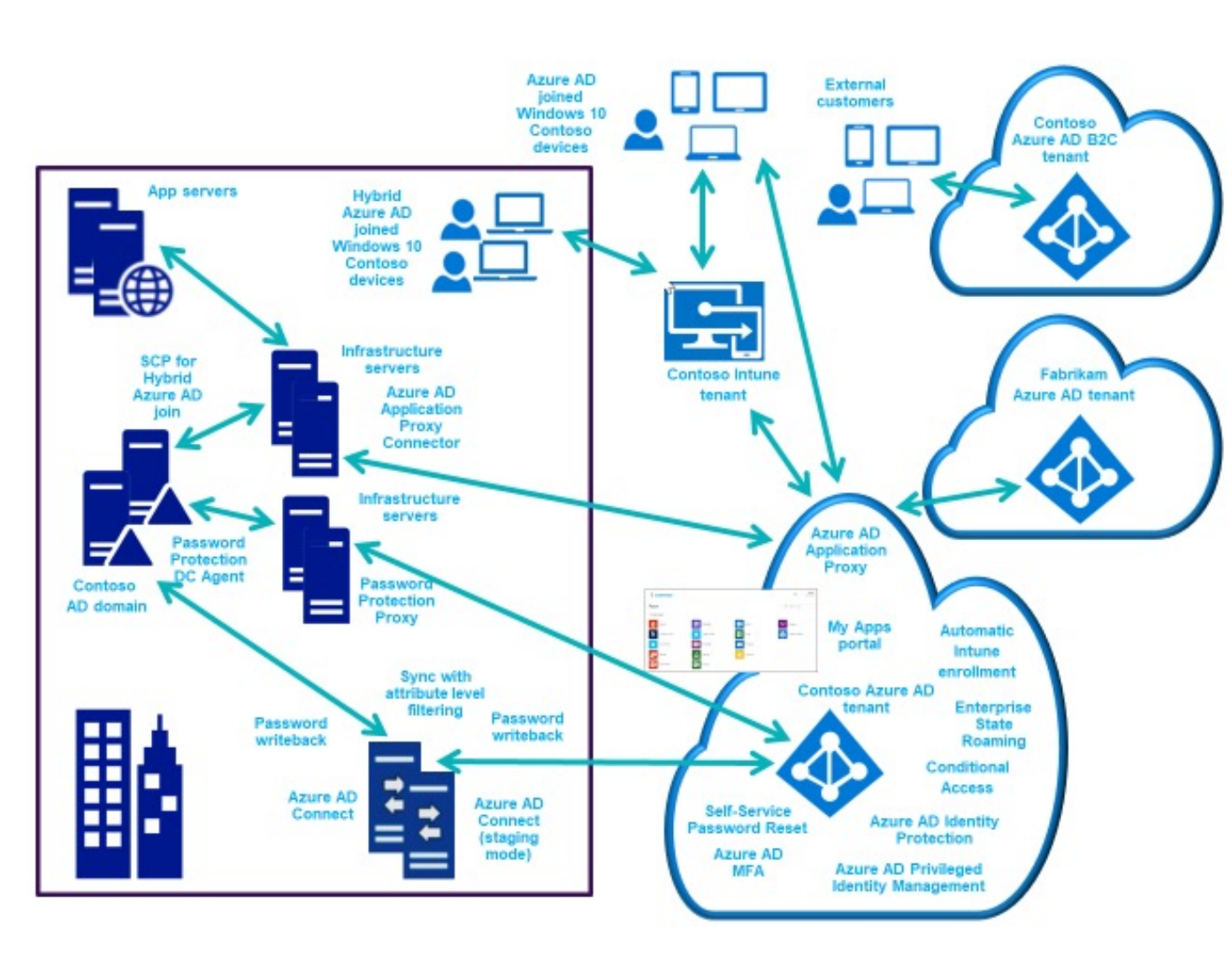
Requirements recap

- Remote users must be able to sign in to their devices by using their AD credentials.
- Existing AD user sign-in hours and password policies must be preserved (although allowed password values could be further restricted).
- User sign-in experience should be simplified by minimizing the number of sign-in prompts and limiting the use passwords in lieu of more secure authentication methods.
- Users device configuration should be simplified by leveraging a mobile device management solution and roaming user-specific settings across multiple devices.
- Control access of users to applications and resources by relying on a combination of multiple conditions, including users group membership, state of the users devices, and dynamically evaluated risk based on comprehensive security-related telemetry.

Requirements recap (continued)

- Users must be allowed to reset their own passwords.
- Designated users should be able to temporarily elevate their privileges to manage other user accounts. All elevation events must be audited.
- Contoso remote users must be able to access on-premises WIA-based applications.
- Fabrikam users must be able to access on-premises WIA-based applications.
- Applications developed by Contoso must be made available to external customers with minimum overhead associated with identity management.
- Resiliency must be maximized whenever possible.
- Infrastructure requirements must be minimized.

Preferred solution (high-level design)



Preferred solution (Azure AD)

- Integration of on-premises AD with Azure AD:
  - Uses pass-through authentication with seamless single sign-on
  - Relies on Azure AD Connect on an on-premises Windows Server with direct connectivity to Active Direct domain controllers and Azure AD
  - Drives a number of other configuration choices, including Hybrid Azure AD join, Self-Service Password Reset with password writeback, Azure AD Password Protection for Windows Server Active Directory, Azure AD Multi-Factor Authentication, Azure AD Privileged Identity Management, Azure AD Conditional Access with Azure AD Identity Protection-based risk assessment, and Azure AD Application Proxy.
- Replacing passwords with more secure authentication methods:
  - Relies on Windows Hello for Business on Windows 10 domain-member computers
  - Uses Hybrid Key trust-based model (available for Hybrid Azure AD join devices)
  - Requires infrastructure changes, including deployment of an internal CA

Preferred solution (Azure AD B2B and B2C)

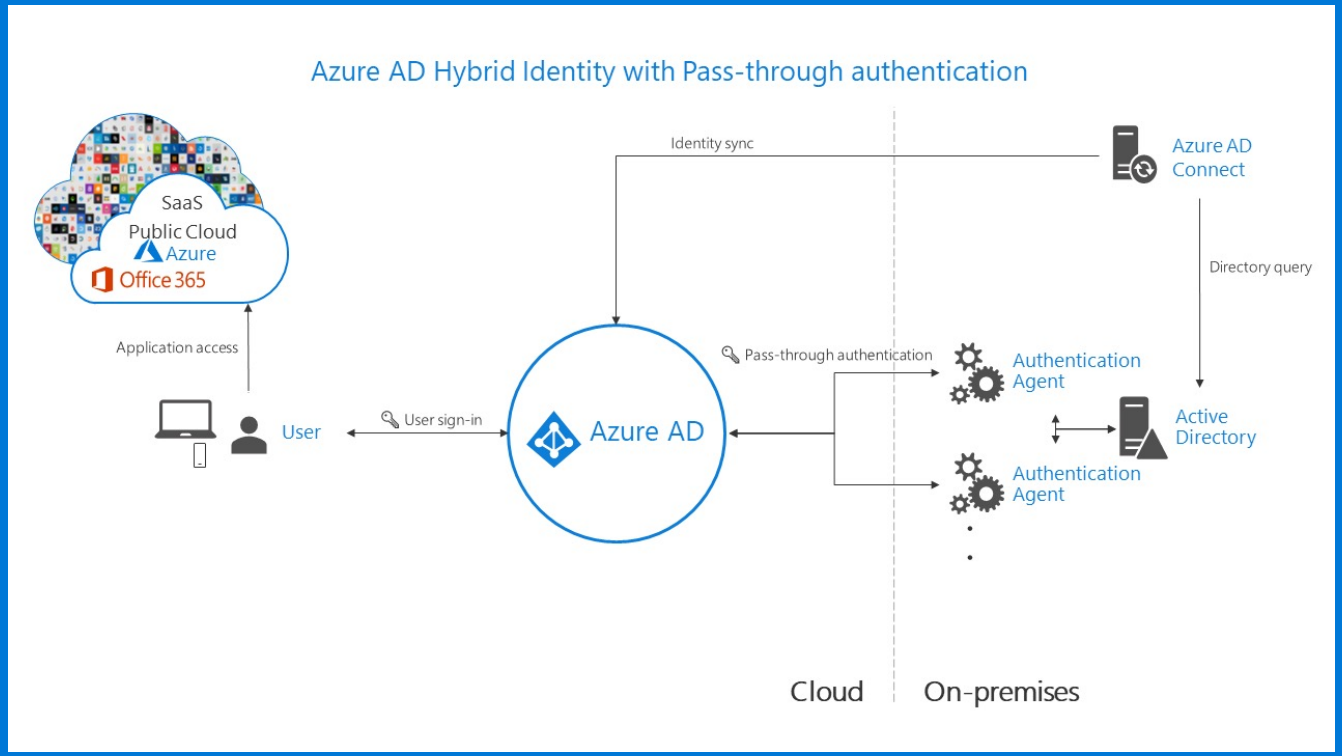
- Access to on-premises applications to business partners:
  - Leverages Azure AD B2B capabilities and Azure AD Application Proxy
- Access to in-house developed mobile and web apps for customers:
  - Leverages a dedicated Azure AD B2C tenant

Preferred solution (prerequisites)

- A new Azure Active Directory tenant with a custom, publicly routable domain name
- Azure AD Connect for integration between AD and Azure AD
- Azure AD Premium P2 licenses, which provide:
  - Azure AD Privileged Identity Management
  - Azure AD Identity Protection
  - Conditional Access (available with Azure AD Premium P1 and P2)
  - Multi-Factor Authentication (available with Azure AD Premium P1 and P2)
  - Azure AD Application Proxy (available with Azure AD Premium P1 and P2)
  - Password Protection for Windows Server Active Directory (available with Azure AD Premium P1 and P2)
  - Self-service password reset/change/unlock with on-premises writeback (available with Azure AD Premium P1 and P2)

Preferred solution (authentication method)

- Pass-through authentication with seamless single sign-on (SSO):
  - Preserves on-premises AD user account restrictions (such as allowed sign-in hours):
    - Not available with Azure AD password hash synchronization
    - Available with federated auth
  - Minimizes infrastructure footprint:
    - Like AD password hash sync
    - Unlike federated auth
  - Streamlines user sign-on:
    - Like AD password hash sync with seamless SSO
    - Like federation-auth



Preferred solution (Windows Hello for Business)

- Benefits:
  - Replaces passwords with strong two-factor authentication on Windows 10 devices
    - authentication uses credential tied to the user's device (biometric or PIN)
  - Lets user authenticate to an Active AD or Azure AD
  - In hybrid deployments, leverages Hybrid Azure AD joined computers
- Implementation:
  - Hybrid Azure AD join of on-premises Windows 10 computers
  - Hybrid key trust deployment model
    - No dependency on federated authentication (unlike other models)

Implementing a hybrid identity solution

- Azure AD configuration:
  - a new Azure AD tenant
  - a publicly routable DNS domain name as a verified, custom DNS domain name of the newly provisioned Azure AD tenant.
  - Enterprise Mobility + Security E5 licenses

Implementing a hybrid identity solution

- Active Directory configuration:
  - The UPN suffix matching the Azure AD custom DNS domain name assigned as the suffix of the userPrincipalName attribute of all AD user accounts to be integrated with Azure AD
  - Recycle Bin:
    - Allows restoring deleted AD user object
    - AD user object restore automatically restores the soft-deleted Azure AD user object during the next synchronization cycle.

Implementing a hybrid identity solution

- Azure AD Connect configuration:
  - Authentication method (pass-through authentication with Seamless SSO):
    - Identify one or more (three recommended) non-dedicated on-premises servers:
      - OS: Windows Servers 2012 R2 or newer
      - Network: direct access to AD domain controllers and outbound access to internet
    - Enable TLS 1.2 on each server
    - Install lightweight Authentication Agents (one per server):
      - The first is installed automatically on the Azure AD Connect server
      - Additional can be installed interactively or via an unattended deployment script

Implementing a hybrid identity solution

- Azure AD Connect configuration:
  - Filtering:
    - Attribute-based:
      - configurable via Synchronization Rules Editor (included in Azure AD Connect)
      - applied when importing objects from AD into to the metaverse (inbound)
      - in Contoso, checks the value of the userPrincipalName attribute of AD user objects
        - looks for match of the UPN suffix and the Azure AD custom DNS domain name
        - this value will be set for individual AD user objects as part of staged deployment
      - objects to be synchronized to Azure AD must have the metaverse attribute cloudFiltered ***not*** set to TRUE

Implementing a hybrid identity solution

- Azure AD Conditional Access
  - Configurable directly from the Azure portal
  - Grants or blocks access to Azure AD integrated resources based on:
    - User identity, group membership, or role assignment
    - Specific cloud application being accessed
    - Device platform, compliance status, or state (e.g. Hybrid Azure AD join)
    - Network location or IP address
    - Client apps being used to access cloud applications
    - Sign-in risk (Requires Identity Protection)
    - Approved client application
  - Supports enforcing MFA and session-level (rather than gated) restrictions

Implementing a hybrid identity solution

- Azure AD Multi-Factor Authentication
  - Three main implementation steps:
    - configuring the MFA registration method
    - configuring the MFA authentication method
    - designating scenarios in which MFA is required

Implementing a hybrid identity solution

- Azure AD Multi-Factor Authentication : registration
  - Three main implementation methods
    - Conditional Access:
      - Offers the most flexibility
      - Conditional Access policies can be used to force users to complete registration at first sign-in (depending on customizable conditions)
    - Modifying the user state:
      - Effectively forces users to use MFA during every sign-in
      - Overrides Conditional Access policies
      - Might be preferred due to lower licensing costs
    - Azure AD Identity Protection

Implementing a hybrid identity solution

- Azure AD Multi-Factor Authentication : registration (Azure AD Identity Protection)
  - Requires Azure AD Premium P2 licensing
  - Includes MFA registration policy:
    - prompt users to register during next interactive sign-in
  - Provides automated risk detection in Conditional Access policies:
    - forces password changes in case of a threat of compromised identity
  - Requires MFA when sign-in is deemed risky in response to such events as:
    - leaked credentials
    - sign-ins from anonymous IP addresses
    - impossible travel to atypical locations
    - sign-ins from unfamiliar locations, infected devices, suspicious IP addresses

Implementing a hybrid identity solution

- Azure AD Multi-Factor Authentication : authentication method
  - Admins need to specify the authentication methods available to users
    - It is recommended to allow more than one authentication method
    - Multiple methods provide backup in case the primary method is unavailable
  - The methods include:
    - Notification through mobile app
    - Verification code from mobile app
    - Call to phone
    - Text message to phone

Implementing a hybrid identity solution

- Azure AD Self-Service Password Reset (SSPR) and password writeback:
  - To configure SSPR, select one or more authentication methods:
    - Mobile app notification
    - Mobile app code
    - Email
    - Mobile phone
    - Office phone
    - Security questions
  - It is recommended to allow more than one authentication method
  - To enable password writeback, use Azure AD Connect:
    - This effectively allows users to reset passwords of their AD accounts

Implementing a hybrid identity solution

- Azure AD password protection for Windows Server Active Directory:
  - Facilitates eliminating easily guessed passwords based on a custom list of prohibited words
  - Relies on two primary components:
    - Azure AD password protection DC agent software
      - Installed on a domain controller running Windows Server 2012 or newer
      - should be installed on all domain controllers to guarantee password protection
      - applies to password changes/resets processed by that domain controller.
    - Azure AD Password Protection Proxy service
      - installed on any domain-joined Windows Server 2012 R2 or newer
      - requires .NET 4.7 and connectivity to internet
      - forwards password policy download requests from domain controllers to Azure AD
      - return responses from Azure AD to the DC Agent service

Implementing a hybrid identity solution

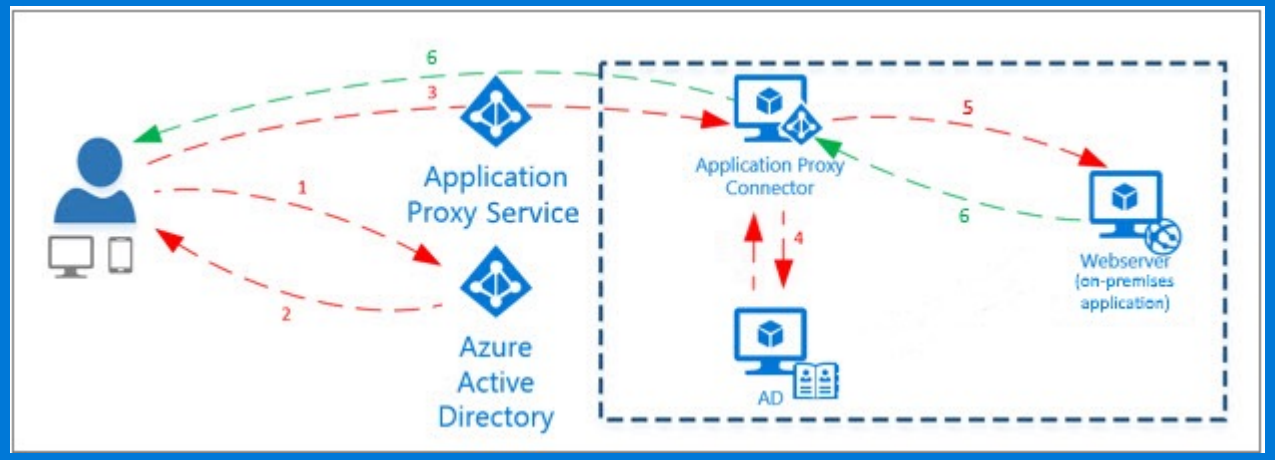
- Smart Lockout
  - Can be included in hybrid deployments using password hash sync or pass-through authentication to protect AD user accounts from being locked out by attackers
  - When using smart lockout with pass-through authentication, make sure that:
    - The Azure AD lockout threshold is less than the AD account lockout threshold.
      - AD account lockout threshold should be at least two times longer
    - The Azure AD lockout duration should be longer than the AD *reset account lockout counter after* duration.
      - Note that, when using graphical interface tools, the Azure AD duration is set in seconds, while the AD duration is set in minutes.

Implementing a hybrid identity solution

- Azure AD Application Proxy:
  - provides access to on-premises web apps via an external URL or an internal portal
  - offers single sign-on experience and consistent interface regardless of the app location
  - includes support for:
    - Web applications that use Windows Integrated Authentication
    - Web applications that use form-based or header-based authentication
    - Web APIs that you want to expose to rich applications on different devices
    - Applications hosted behind a Remote Desktop Gateway
    - Rich client apps integrated with the Active Directory Authentication Library

Implementing a hybrid identity solution

- Azure AD Application Proxy:
  - Relies on the following components:
    - **Endpoint**: a URL or a portal via which users access on-premises apps
    - **Azure AD tenant**: authenticates users



    - **Application Proxy service**: a cloud service hosted by Azure AD that passes sign-in tokens from users to Application Proxy Connector
    - **Application Proxy Connector**: a lightweight agent managing communication between on-premises apps and the Application Proxy service via an outbound connection
      - Eliminates the need for opening inbound ports on perimeter firewalls
      - Requires on-premises Windows Server 2012 R2 or newer with TLS 1.2 enabled
        - domain joined for SSO to WIA apps (to allow Kerberos Constrained Delegation)
    - **Active Directory**: performs authentication required to access on-premises apps

Assessing resiliency of a hybrid identity solution

- Network connectivity between Active Directory and Azure Active Directory:
  - Should be highly available and performant:
    - Failure of network connectivity will result in pass-through authentication failures
  - To mitigate the risk of failures:
    - Implement password hash synchronization as a failover authentication mechanism
    - Extend AD environment to Azure by:
      - Implementing an Azure virtual network
      - Establishing a S2S VPN or ExpressRoute connection between AD and the Azure virtual network
      - Deploying additional AD domain controllers into the Azure virtual network
      - Installing additional pass-through authentication agents on Azure VMs in the same Azure virtual network.

Assessing resiliency of a hybrid identity solution

- Azure AD Connect synchronization engine:
  - Should be highly available and performant:
    - Failure of the engine or network connectivity will prevent synchronization of changes:
      - This includes password and AD object changes (e.g. disabling of user accounts)
      - Downtime has lesser impact in pass-through and federated authentication scenarios
  - To mitigate, install Azure AD Connect on another server in the staging mode:
    - The staging mode option is supported directly by the installation wizard
    - Failover can be performed by re-running the wizard on the staging server
    - In the staging mode, the sync engine imports and synchronizes data, but does not export it
    - Password sync and password writeback are disabled while in staging mode

Assessing resiliency of a hybrid identity solution

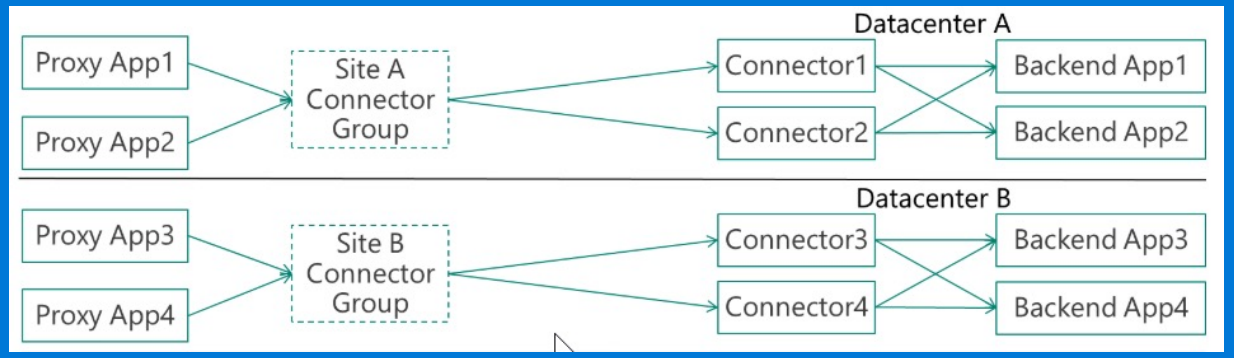
- Azure AD Connect pass-through authentication agent:
  - Should be highly available and performant:
    - Failures will result in pass-through authentication failures
  - To mitigate:
    - Install two or more (three are recommended) authentication agents
    - Installing multiple agents ensures high availability, but not deterministic load balancing
    - To determine the number, consider the peak and average load of sign-in requests
    - An agent can handle 300-400 authentications/sec on a 4-core CPU, 16-GB RAM server

Assessing resiliency of a hybrid identity solution

- Azure AD Password Protection for Windows Server Active Directory:
  - Should be highly available and performant:
    - Proxy servers must be available to download new password policies from Azure
    - DC Agents calls proxy servers in round-robin fashion and skip non-responding ones
  - To mitigate:
    - Deploy two proxy servers to ensure availability
      - The DC Agent maintains a local cache of the most recently downloaded password policy and enforces it even if proxy servers are not available.
      - Brief outages of the proxy servers do not impact significantly
        - Update frequency for password policies is usually days, not hours or less.

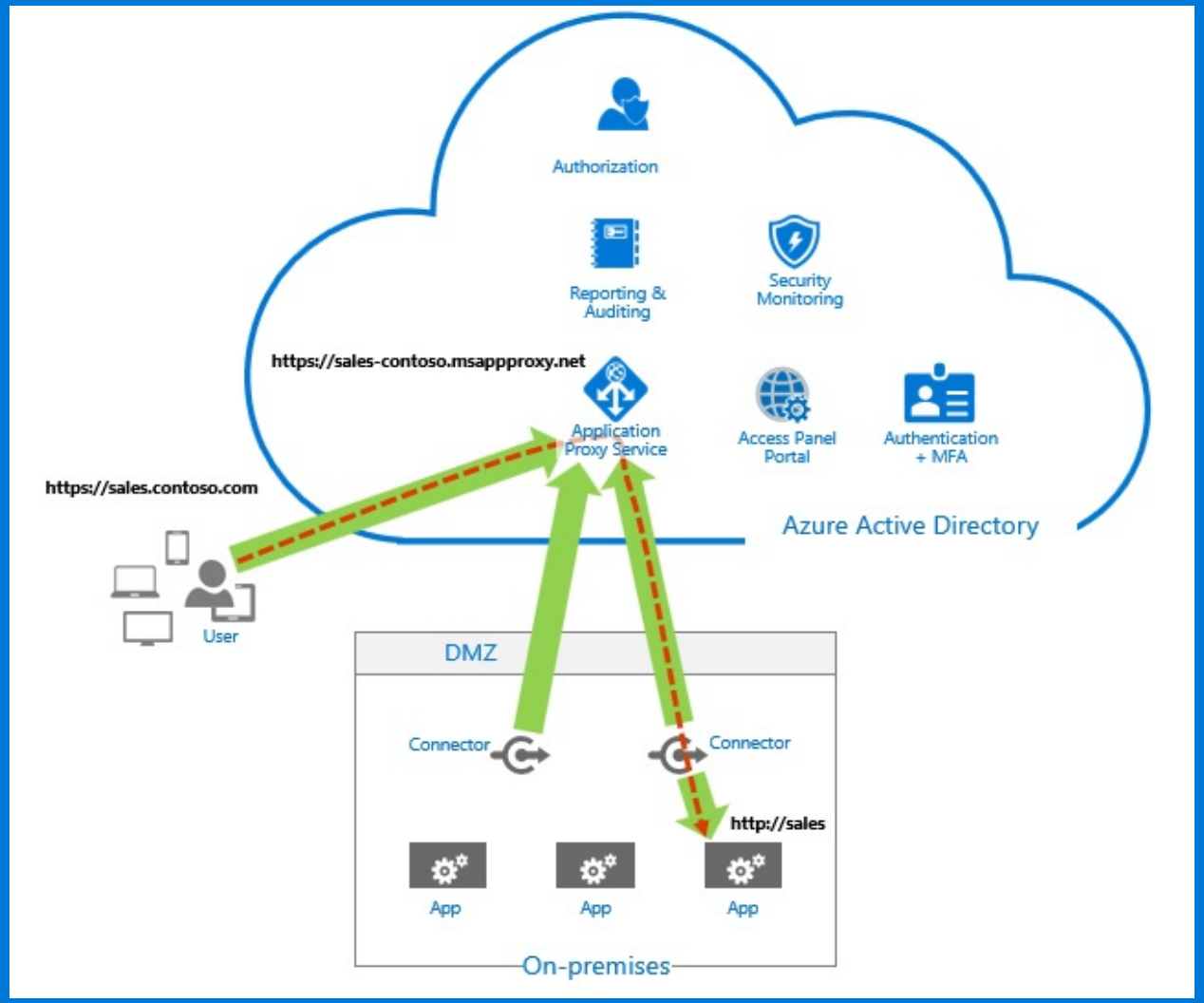
Assessing resiliency of a hybrid identity solution

- Azure AD Application Proxy connector
  - Should be highly available and performant
    - Failures will affect access to applications
  - To mitigate:
    - Install two or more connectors and organize them into connector groups:
      - Each group handles traffic to specific set of applications.
      - Each group should have at least two connectors to provide high availability
      - Groups can improve latency when accessing apps in different regions
      - For connector sizing info, refer to https://docs.microsoft.com/en-us/azure/active-directory/manage-apps/application-proxy-connectors
  - Azure AD Password Protection Proxy and Application Proxy should not be installed on the same server (different versions of the Microsoft Azure AD Connect Agent Updater service)



Component failover in a hybrid identity solution

- Azure AD Application Proxy connector
  - No explicit failover is required:
    - Application requests are dynamically load balanced across all available connectors
    - Each new request is routed to one of currently available connectors
    - If a connector becomes temporarily unavailable, it is excluded from request distribution



Component failover in a hybrid identity solution

- Network connectivity between Active Directory and Azure Active Directory:
  - Failover to password hash synchronization-based authentication is not automatic
  - To perform a switch, re-run Azure AD Connect to reconfigure the authentication method
  - You must have at that point connectivity to AD domain controllers and Azure AD
    - This requirement limits the viability of such failover.
    - Instead, consider a resilient infrastructure design (e.g. a hybrid architecture with additional AD domain controllers and pass-through authentication agents in an Azure virtual network).

Component failover in a hybrid identity solution

- Azure AD Connect synchronization engine:
  - Failover to the redundant server running in the staging mode is not automatic
  - To perform a failover, re-run the Azure AD Connect installation wizard to change the mode of the staging instance:
    - Disable the option labeled *Enable staging mode*. *When selected, synchronization will not export any data to AD or Azure AD* on the ***Ready to configure*** page of the wizard
  - Either deprovision the former active instance or switch it to the staging mode:
    - There must be only a single server which is actively exporting data

Component failover in a hybrid identity solution

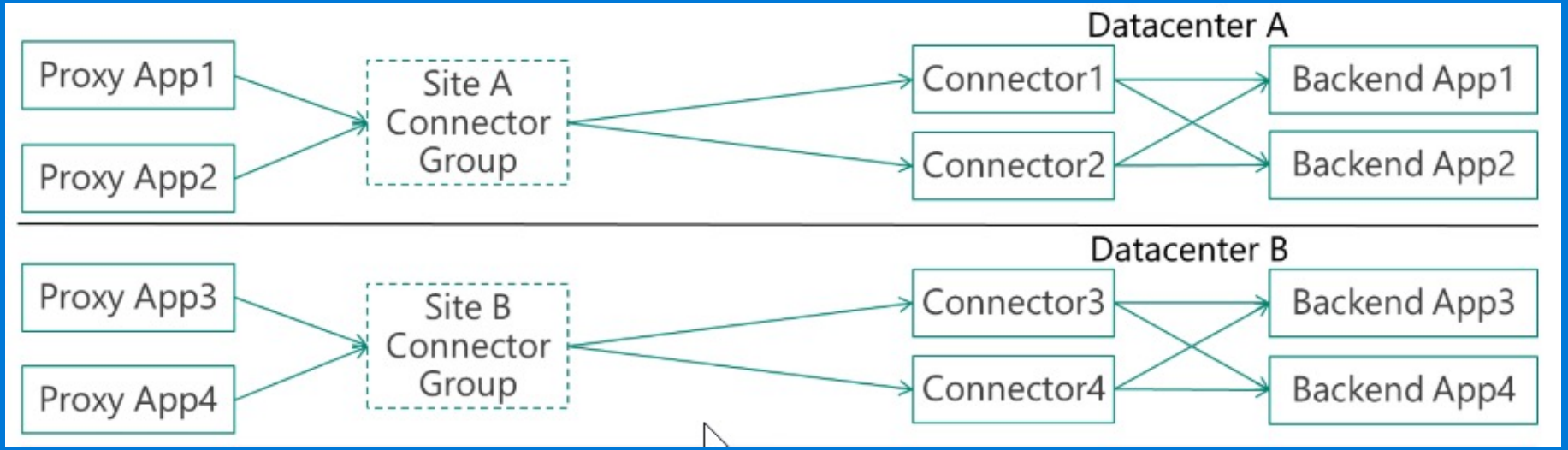
- Azure AD Connect passthrough authentication agent:
  - No explicit failover is required
  - Authentication requests are dynamically distributed across all available agents

Component failover in a hybrid identity solution

- Azure AD Password Protection for Windows Server Active Directory
  - DC Agents
    - No explicit failover is required:
      - Agents should be installed on all domain controllers, which provides high availability
  - Azure AD Password Protection Proxy service agents
    - No explicit failover is required:
      - DC Agents calls proxy servers in round-robin fashion and skip non-responding ones

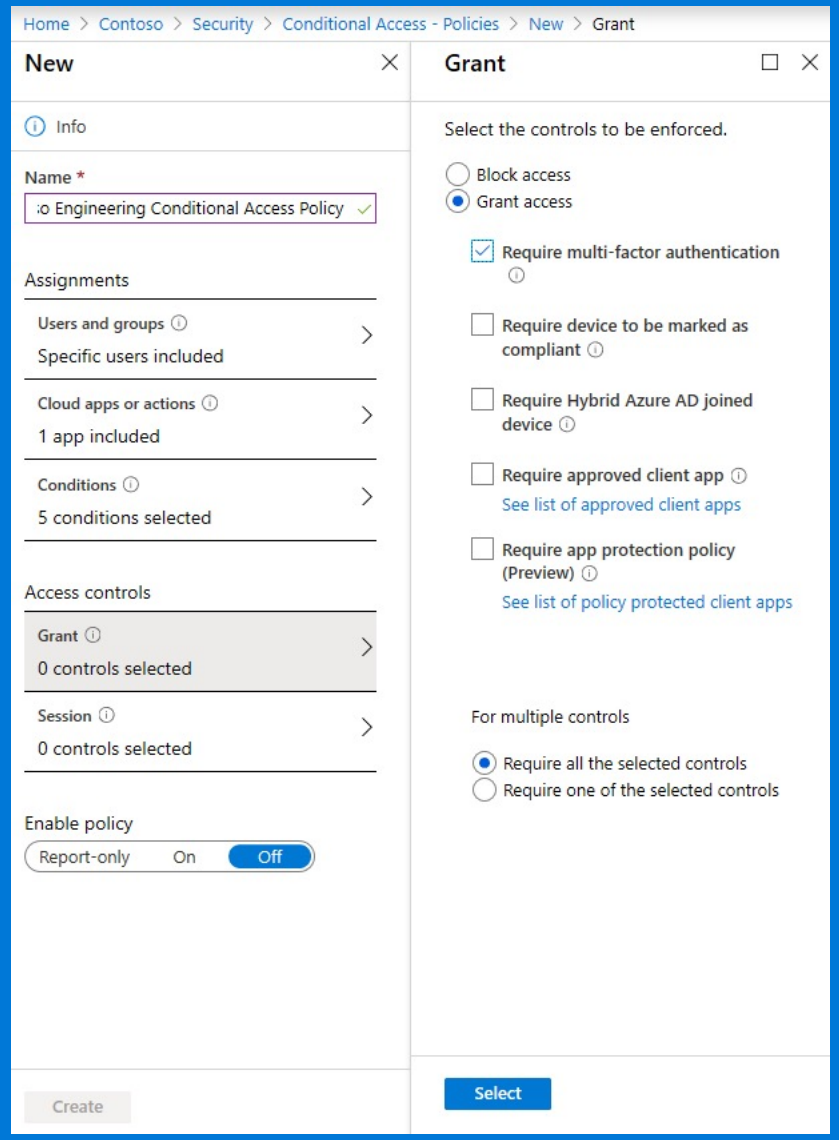
Component failover in a hybrid identity solution

- Azure AD Application Proxy connector
  - No explicit failover is required:
    - Application requests are dynamically load balanced across all available connectors
    - Each new request is routed to one of currently available connectors
    - If a connector becomes temporarily unavailable, it is excluded from request distribution



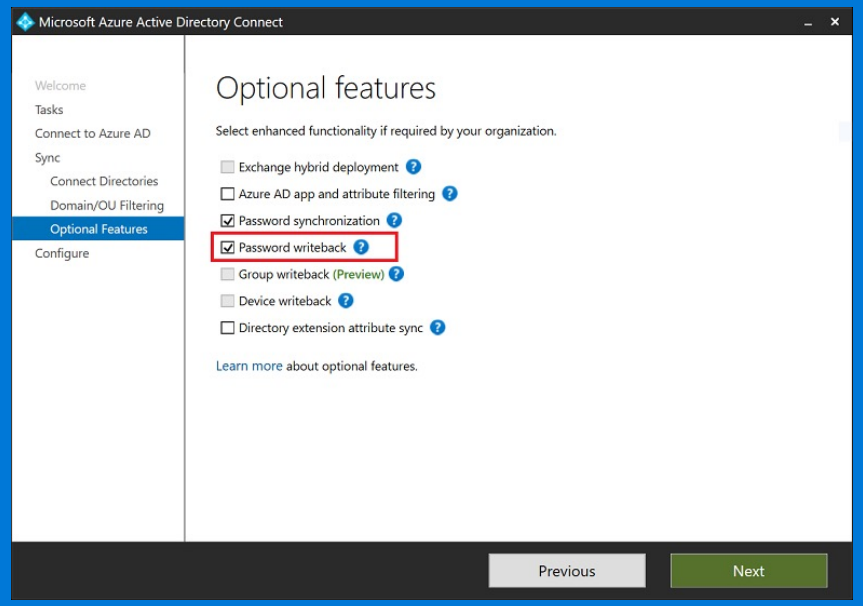
Optimizing authentication configuration

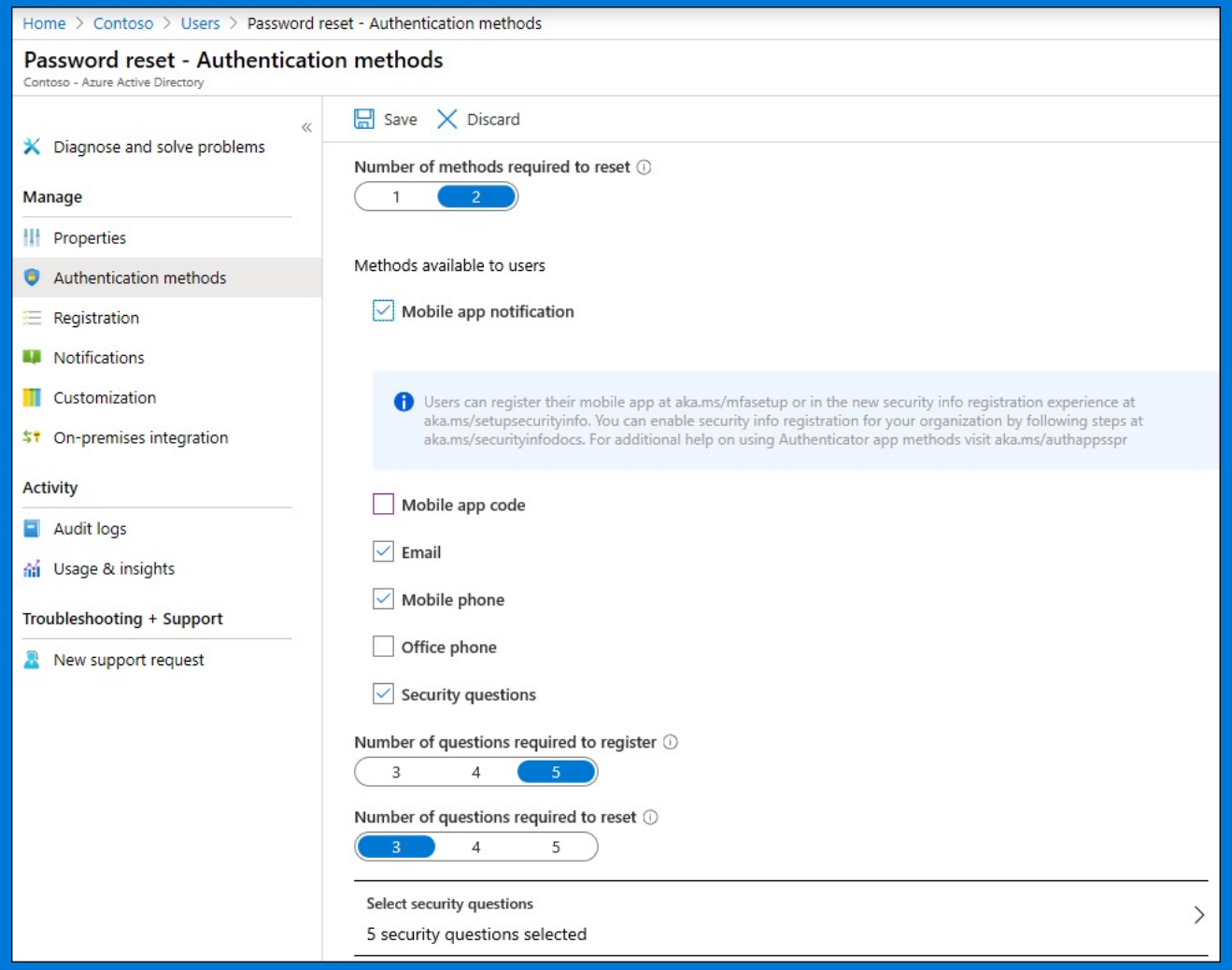
- Multi-Factor Authentication
  - Use it in combination with Conditional Access and Azure AD Identity Protection
  - Leverage it when configuring Self-Service Password Reset (SSPR)



Optimizing authentication configuration

- Self-Service Password Reset:
  - Combine Self-Service Password Reset with password writeback:
    - Allows users to reset passwords of their AD accounts.





Optimizing authentication configuration

- Windows Hello for Business:
  - Use in Windows 10 to replace passwords with strong two-factor authentication
  - Leverage Azure AD and Azure AD Connect in hybrid scenarios.
  - Implement in combination with Azure AD pass-through authentication in the Hybrid Azure AD joined Key Trust Deployment model (eliminates dependency on AD FS)
  - Use Azure AD Connect to register Windows 10 client devices in Azure AD:
    - The Azure AD Connect wizard configures the AD SCPs for device registration

Optimizing authorization configuration

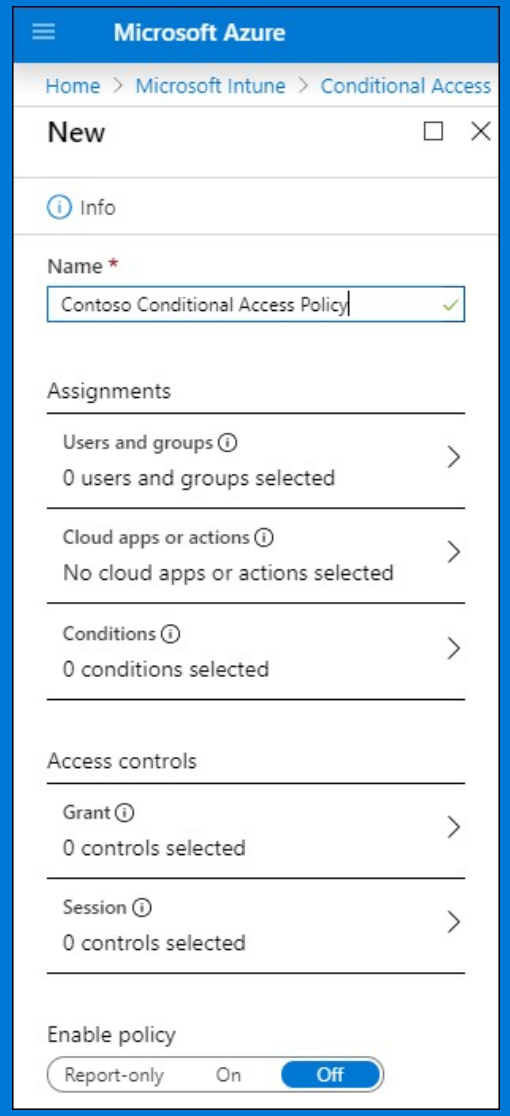
- Privileged Identity Management:
  - Leverage PIM to:
    - Provide just-in-time privileged access to Azure AD and Azure resources
    - Assign time-bound access to resources
    - Require approval to activate privileged roles
    - Enforce multi-factor authentication to activate any role
    - Ensure that a rationale is provided as part of elevation approval process
    - Configure notifications triggered by activation of privileged roles
    - Facilitate access reviews to validate whether users should be eligible for role assignment
    - Track elevation events

Optimizing access control and management

- Mobile Device Management
  - Use an MDM solution, such as Microsoft Intune, that supports hybrid identity
  - Leverage Intune Azure AD integration to automatically enroll Azure AD joined devices
  - Take advantage of a wide range of Intune device and application management features:
    - configuration and compliance policies
    - software deployment
    - remote administration
    - support for multi-identity (which separates corporate and personal data)
  - Benefit from bundled licensing offers:
    - Microsoft Intune licensing is part of EMS E5, which includes Azure AD Premium P2

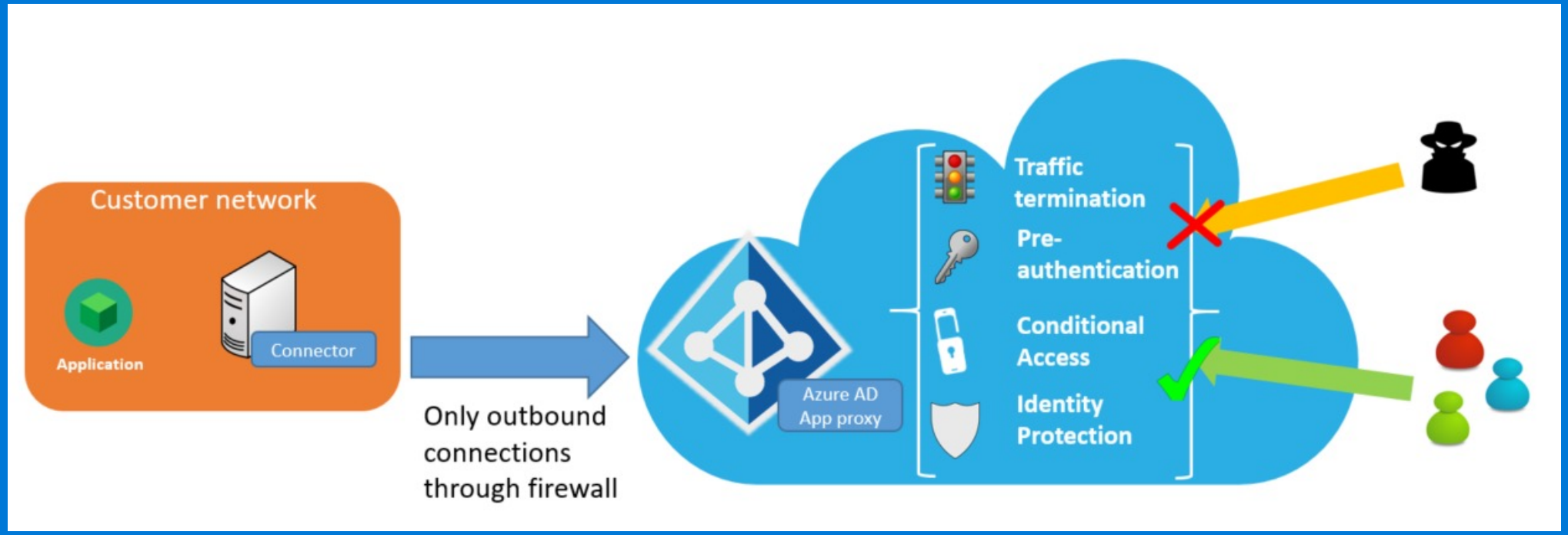
Optimizing access control and management

- Azure AD Conditional Access
  - Restrict access to resources integrated with Azure AD based on:
    - Azure AD group membership
    - target resource
    - device platform and state (e.g. Hybrid Azure AD join)
    - network location
    - client application being used to access the resource
    - sign-in risk
      - evaluated by relying on Azure AD Identity Protection
    - device compliance:
      - evaluated by Intune compliance policies



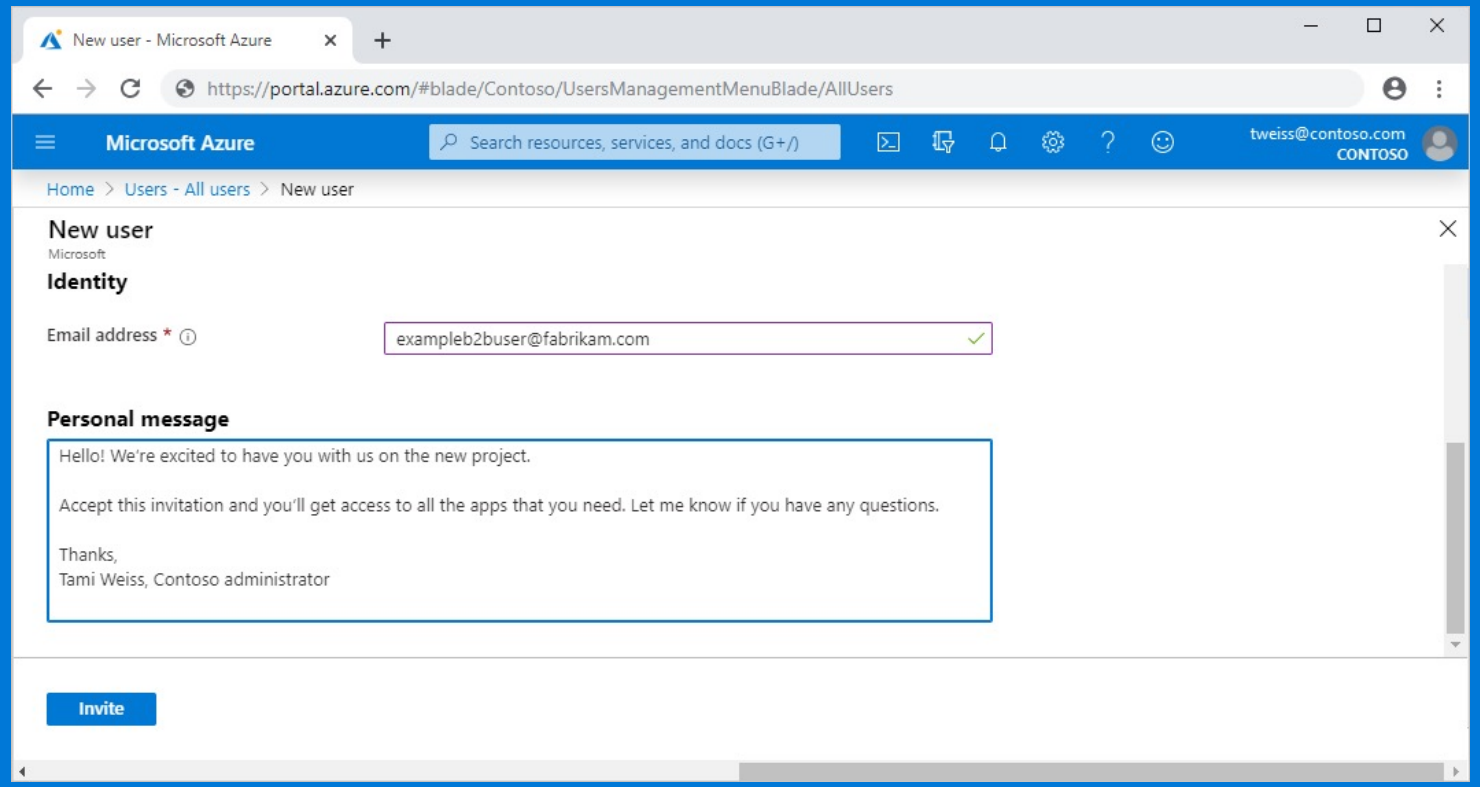
Optimizing access control and management

- Azure AD Application Proxy
  - Implement seamless, secure access to on-premises LOB applications
  - Provide consistent user experience when accessing cloud and on-premises apps



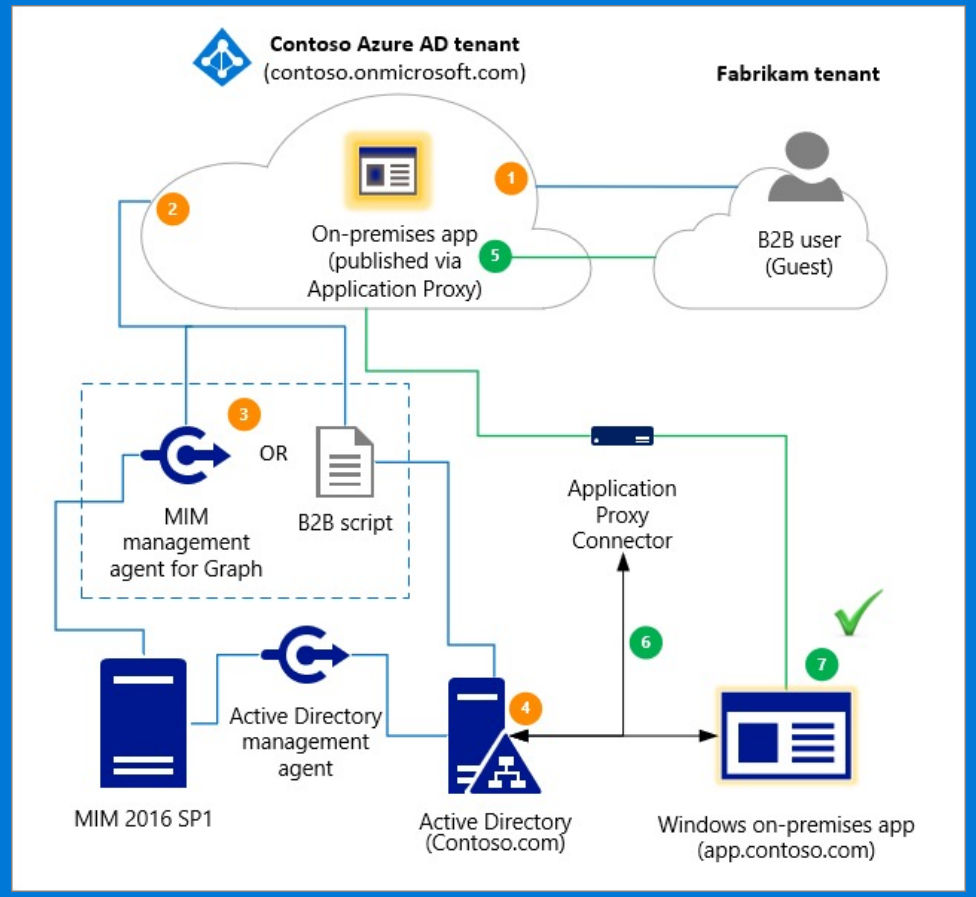
Optimizing access control and management

- Azure AD B2B
  - Grant access to apps and services to business partners by using Azure AD integration
  - Provision guest accounts via a straightforward invitation and redemption process
  - Allow invitees use their existing credentials to authenticate
  - Identify partner guest accounts based on the userType attribute set to Guest



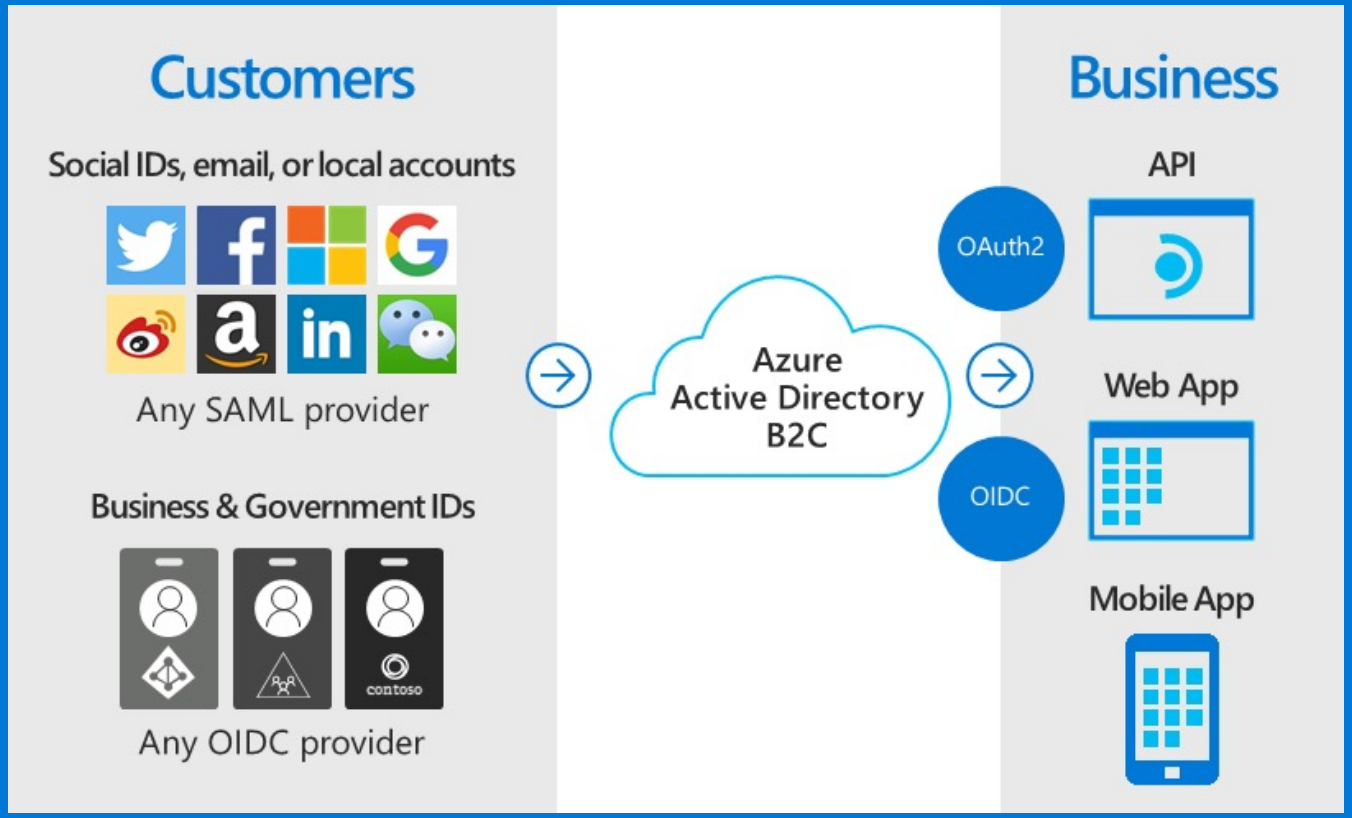
Optimizing access control and management

- Azure AD B2B
  - Grant guest accounts access to on-premises apps:
    - For SAML-based authentication, make apps available directly from the Azure portal:
      - Add them to Azure AD based on the non-gallery application template
      - Publish them via Azure AD Application Proxy
    - For IWA and KCD apps, in addition, create AD user accounts for B2B guest users via:
      - Microsoft Identity Manager (MIM) and the MIM agent for Microsoft Graph.
      - A PowerShell script: a lightweight solution that does not require MIM.



Optimizing access control and management

- Azure AD B2C
  - Allow customers to use their preferred identities to authenticate when accessing apps and APIs offered by your organization:
    - Create a B2C Azure AD tenant separate from the one used by your organization's users and apps
    - Register apps and APIs in the B2C tenant to make them available to customers



Preferred objections handling

Our Active Directory domain is using a non-routable domain name. We cannot risk renaming it in order to implement single sign-on with Azure Active Directory.

Potential Answer:

Contoso does not have to rename their Active Directory domain in order to integrate with an Azure Active Directory tenant. Such integration is possible regardless of the DNS name of the Active Directory domain. What's important in order to ensure single sign-on experience for Active Directory users accessing cloud-based resources is to ensure that there is a match between the userPrincipalName in Active Directory and Azure AD. This is the Microsoft's recommended approach.

Preferred objections handling

We have heard that it is not possible to run simultaneously multiple instance of Azure AD Connect. All identity services components in our environment must provide resiliency and support failover.

Potential Answer:

While Azure AD Connect cannot operate in the active/active mode, it is possible to setup an additional server hosting the sync engine operating in the staging mode. In this mode, the sync engine imports and synchronizes data the same way as the active instance, but it does not export anything to Azure AD or AD. Password sync and password writeback features of Azure AD Connect are disabled while in staging mode. Since a server in the staging mode continues to receive changes from Active Directory and Azure AD, it can quickly take over the responsibilities of a failed active server. The switch involves simply re-running the Azure AD Connect installation wizard.

Preferred objections handling

If we decide to integrate our Active Directory environment with Azure Active Directory, this must be performed in stages. This is likely to be complex, considering that users in each stage would be members of different Active Directory groups and their accounts might reside in different Active Directory organizational units.

Potential Answer:

Azure AD Connect supports a number of different filtering options that determine the scope of synchronized Active Directory objects. While organizational unit based filtering is the most straightforward to configure option, the scope can be based on a value of individual Active Directory attributes, which offers object-level granularity.

Preferred objections handling

Synchronizing our Active Directory accounts with Azure AD accounts makes the former vulnerable to malicious or accidental lockouts that affect the latter. This would effectively expose our on-premises environment to external attacks.

Potential Answer:

Azure AD offers the Smart Lockout functionality, which can be integrated with hybrid deployments, using password hash sync or pass-through authentication to protect on-premises Active Directory accounts from being locked out by attackers. By setting smart lockout policies in Azure AD appropriately, attacks can be filtered out before they reach on-premises Active Directory.

Preferred objections handling

A number of critical web applications running in our on-premises environment rely on Kerberos-based Windows Integrated Authentication. Microsoft states that Azure Active Directory does not support Kerberos. Doesn't this mean that remote users authenticating to Azure Active Directory and our business partners will not be able to properly authenticate and access these applications?

Potential Answer:

Azure AD provides the ability to access to on-premises web applications by relying on Azure AD Application Proxy via an external URL or an internal application portal. Azure AD Application Proxy offers a single sign-on experience and consistent user interface regardless of the location of the target app. For example, Application Proxy can facilitate access to on-premises line of business (LOB) applications, Office 365, or any other SaaS-based application integrated with Azure AD. This eliminates the need for VPN infrastructure and integrates with other Azure AD features, such as Conditional Access and Multi-Factor Authentication.

Customer quote

"Azure AD offers a wide range of new opportunities that will allow us to extend our identity and access management far beyond our existing on-premises Active Directory environment, facilitating support for our remote users and providing secure and easy to manage integration platform for our business partners and customers."

---Andrew Cross, CIO, Contoso

Microsoft