AZ-305T00A
Designing Microsoft Azure Infrastructure Solutions

Design Authentication and Authorization Solutions

https://learn.microsoft.com/training/modules/design-authentication-authorization-solutions/

Learning Objectives

- Design for identity and access management
- Design for Microsoft Entra ID
- Design for Microsoft Entra B2B
- Design for Azure Active Directory B2C
- Design for conditional access
- Design for identity protection
- Design for access reviews
- Design service principals for applications
- Design for Azure key vault
- Case study
- Learning recap

AZ-305: Design Identity, Governance, and Monitoring Solutions (25-30%)

Design Authentication and Authorization Solutions

- Recommend an authentication solution
- Recommend an identity management solution
- Recommend a solution for authorizing access to Azure resources
- Recommend a solution to manage secrets, certificates, and keys

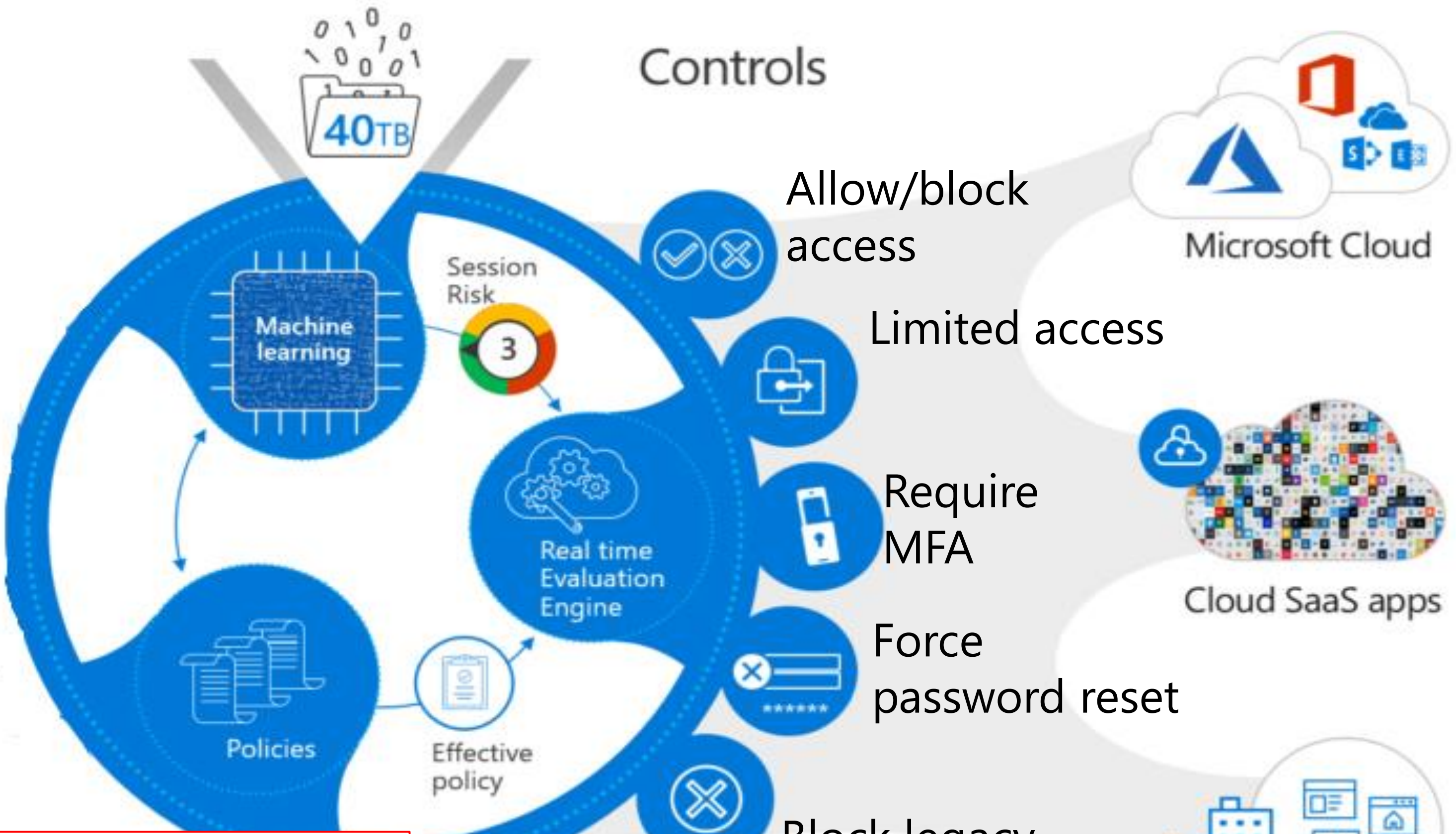
Design for identity and access management

Follow the Zero Trust model guidelines

Never trust, always verify.

- Employee and partner user and roles
- Trusted and compliant devices
- Physical and virtual location
- Client apps and authentication method

Conditions



Controls

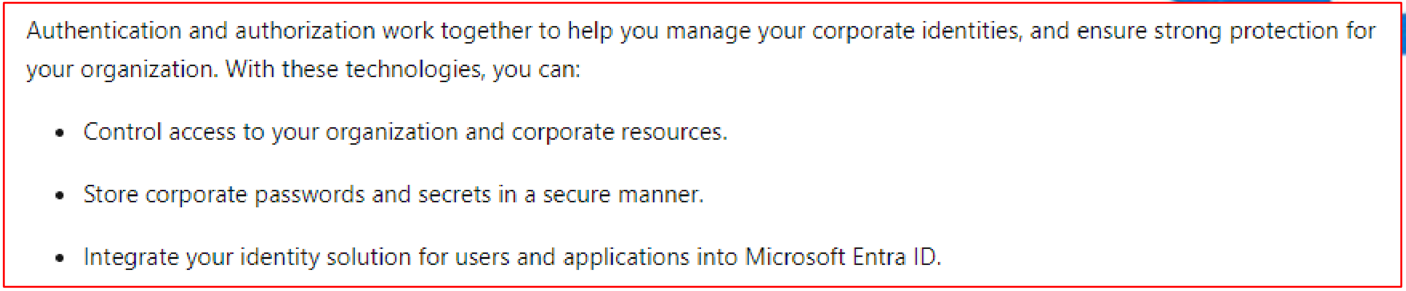
Allow/block access

Limited access

Require MFA

Force password reset

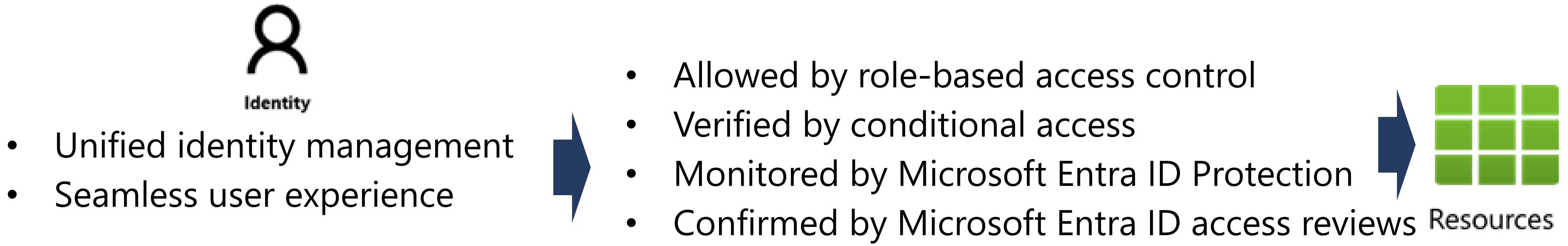
Block legacy authentication



Authentication and authorization work together to help you manage your corporate identities, and ensure strong protection for your organization. With these technologies, you can:

- Control access to your organization and corporate resources.
- Store corporate passwords and secrets in a secure manner.
- Integrate your identity solution for users and applications into Microsoft Entra ID.

What is identity and access management



Identity

- Unified identity management
- Seamless user experience

- Allowed by role-based access control
- Verified by conditional access
- Monitored by Microsoft Entra ID Protection
- Confirmed by Microsoft Entra ID access reviews

Resources

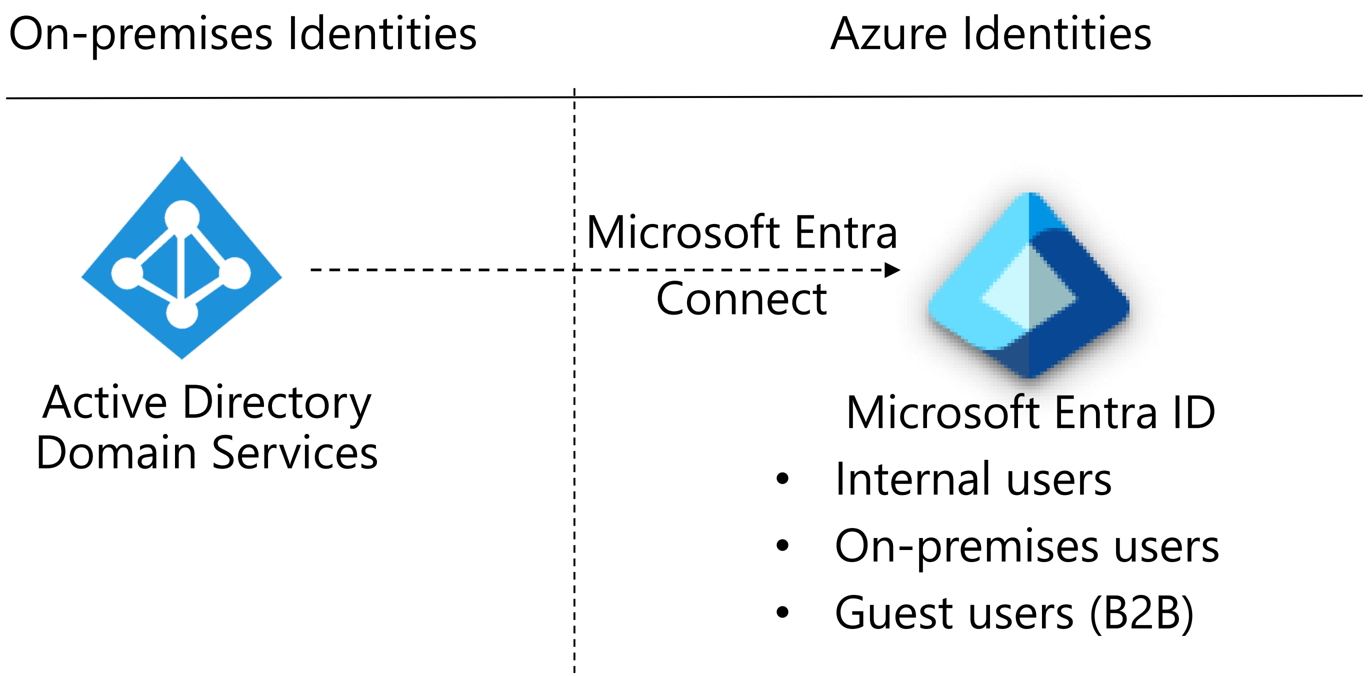
If you need this	Use this
Provide identity and access management for employees in a cloud or hybrid environment.	Microsoft Entra ID
Collaborate with guest users and external business partners like suppliers and vendors.	Microsoft Entra B2B
Control how customers sign up, sign in, and manage their profiles when they use your applications.	Azure AD Business to Consumer (B2C)

Design for Microsoft Entra ID

When to use Microsoft Entra ID

Microsoft Entra ID is a cloud-based solution for identity and access management. Microsoft Entra ID is a multitenant, cloud-based directory, and identity management service.

- Centralize identity management
- Establish a single Microsoft Entra tenant
- Use Microsoft Entra Connect Sync, or Microsoft Entra Cloud Sync for hybrid identity synchronization



On-premises Identities

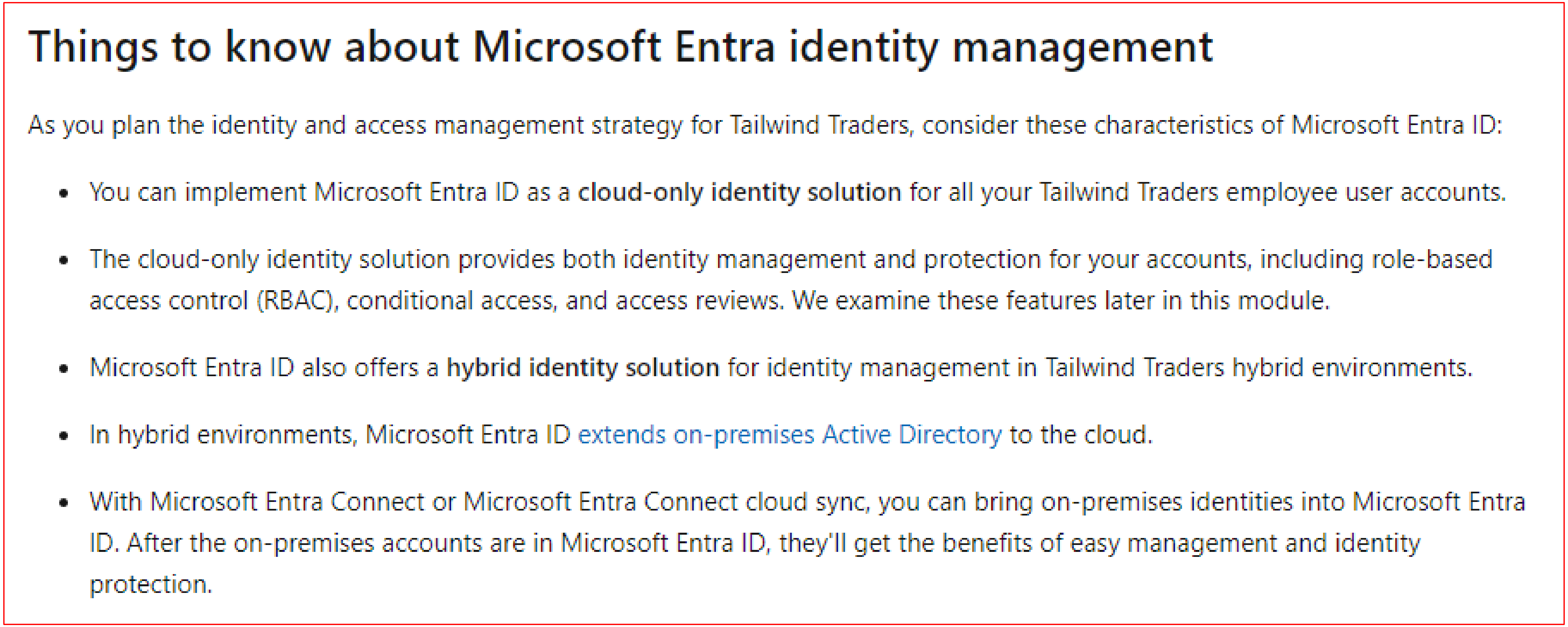
Azure Identities

Active Directory
Domain Services

Microsoft Entra
Connect

Microsoft Entra ID

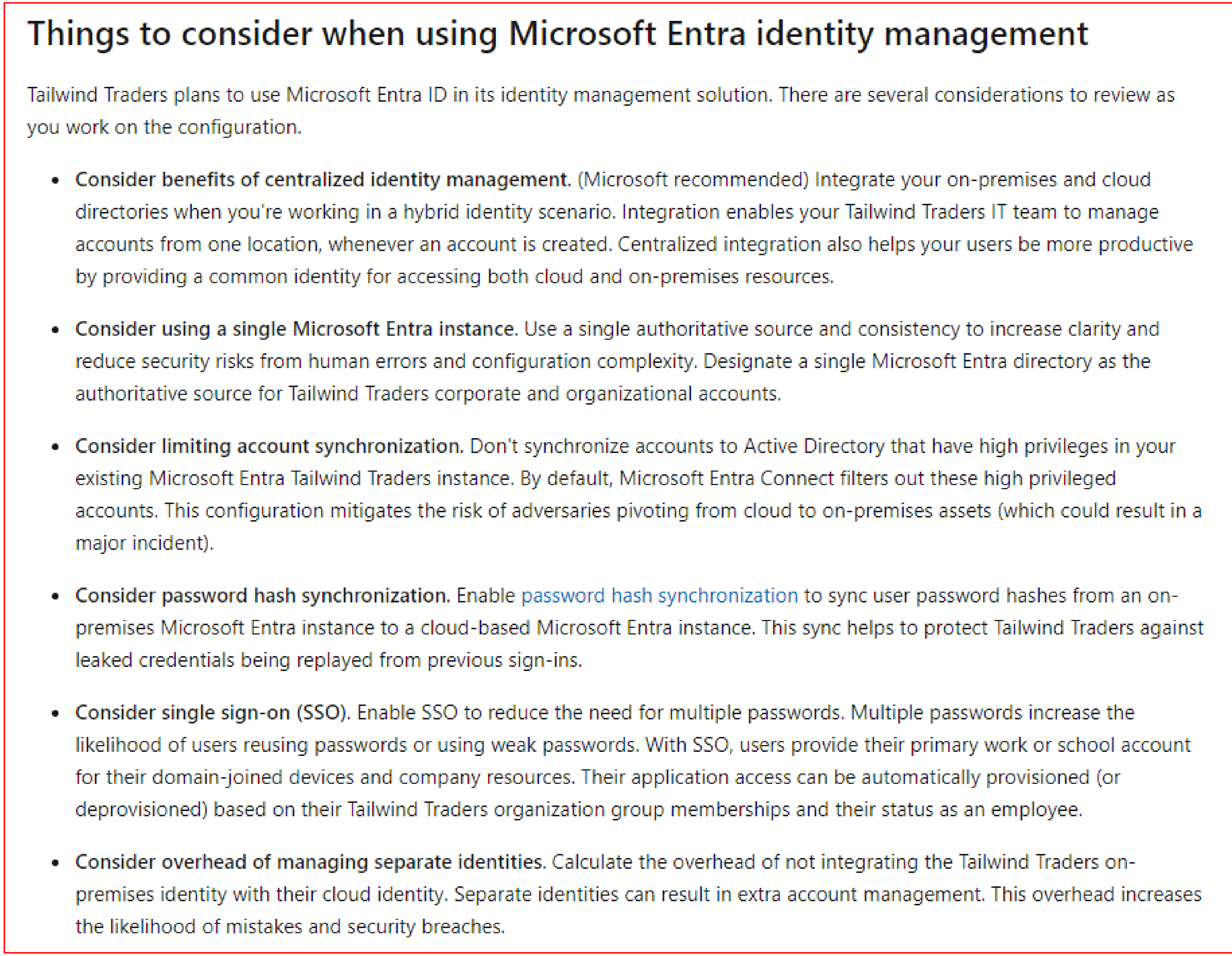
- Internal users
- On-premises users
- Guest users (B2B)



Things to know about Microsoft Entra identity management

As you plan the identity and access management strategy for Tailwind Traders, consider these characteristics of Microsoft Entra ID:

- You can implement Microsoft Entra ID as a **cloud-only identity solution** for all your Tailwind Traders employee user accounts.
- The cloud-only identity solution provides both identity management and protection for your accounts, including role-based access control (RBAC), conditional access, and access reviews. We examine these features later in this module.
- Microsoft Entra ID also offers a **hybrid identity solution** for identity management in Tailwind Traders hybrid environments.
- In hybrid environments, Microsoft Entra ID extends on-premises Active Directory to the cloud.
- With Microsoft Entra Connect or Microsoft Entra Connect cloud sync, you can bring on-premises identities into Microsoft Entra ID. After the on-premises accounts are in Microsoft Entra ID, they'll get the benefits of easy management and identity protection.



Things to consider when using Microsoft Entra identity management

Tailwind Traders plans to use Microsoft Entra ID in its identity management solution. There are several considerations to review as you work on the configuration.

- **Consider benefits of centralized identity management.** (Microsoft recommended) Integrate your on-premises and cloud directories when you're working in a hybrid identity scenario. Integration enables your Tailwind Traders IT team to manage accounts from one location, whenever an account is created. Centralized integration also helps your users be more productive by providing a common identity for accessing both cloud and on-premises resources.

- **Consider using a single Microsoft Entra instance.** Use a single authoritative source and consistency to increase clarity and reduce security risks from human errors and configuration complexity. Designate a single Microsoft Entra directory as the authoritative source for Tailwind Traders corporate and organizational accounts.

- **Consider limiting account synchronization.** Don't synchronize accounts to Active Directory that have high privileges in your existing Microsoft Entra Tailwind Traders instance. By default, Microsoft Entra Connect filters out these high privileged accounts. This configuration mitigates the risk of adversaries pivoting from cloud to on-premises assets (which could result in a major incident).

- **Consider password hash synchronization.** Enable password hash synchronization to sync user password hashes from an on-premises Microsoft Entra instance to a cloud-based Microsoft Entra instance. This sync helps to protect Tailwind Traders against leaked credentials being replayed from previous sign-ins.

- **Consider single sign-on (SSO).** Enable SSO to reduce the need for multiple passwords. Multiple passwords increase the likelihood of users reusing passwords or using weak passwords. With SSO, users provide their primary work or school account for their domain-joined devices and company resources. Their application access can be automatically provisioned (or deprovisioned) based on their Tailwind Traders organization group memberships and their status as an employee.

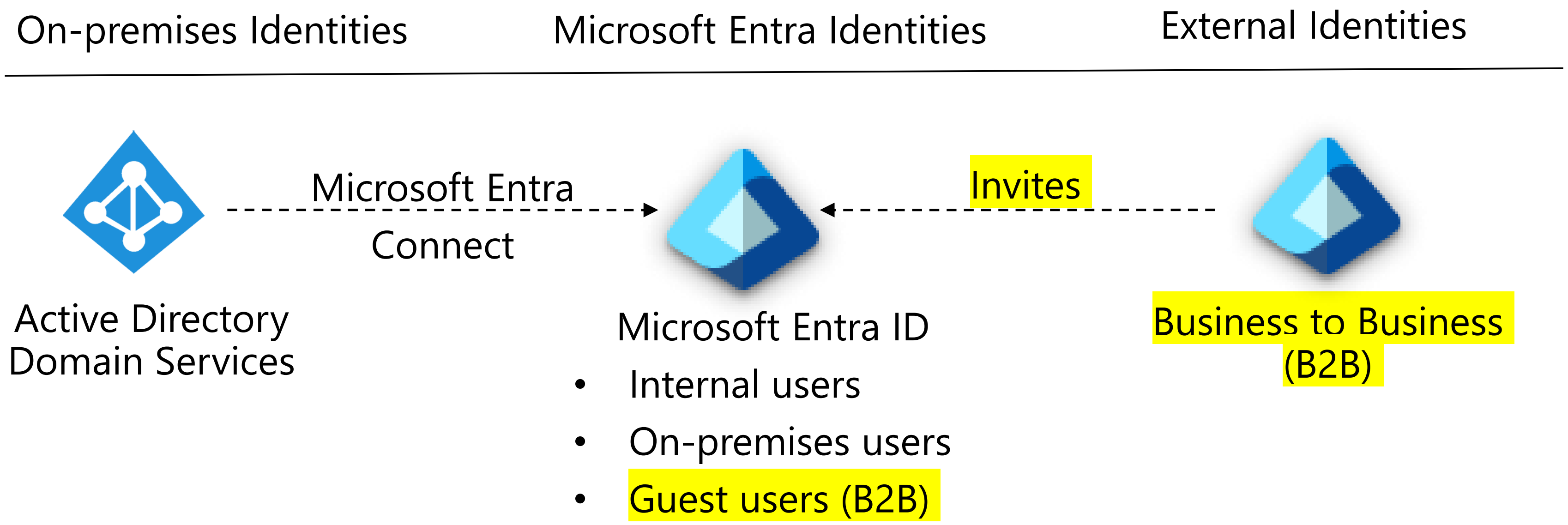
- **Consider overhead of managing separate identities.** Calculate the overhead of not integrating the Tailwind Traders on-premises identity with their cloud identity. Separate identities can result in extra account management. This overhead increases the likelihood of mistakes and security breaches.

Design for Microsoft Entra Business to Business

When to use Microsoft Entra Business to Business (B2B)

Microsoft Entra B2B enables you to securely collaborate with external partners.

- Integrate with identity providers
- Use conditional access policies to intelligently grant or deny access
- Require MFA for guest users



On-premises Identities

Microsoft Entra Identities

External Identities

Microsoft Entra Connect

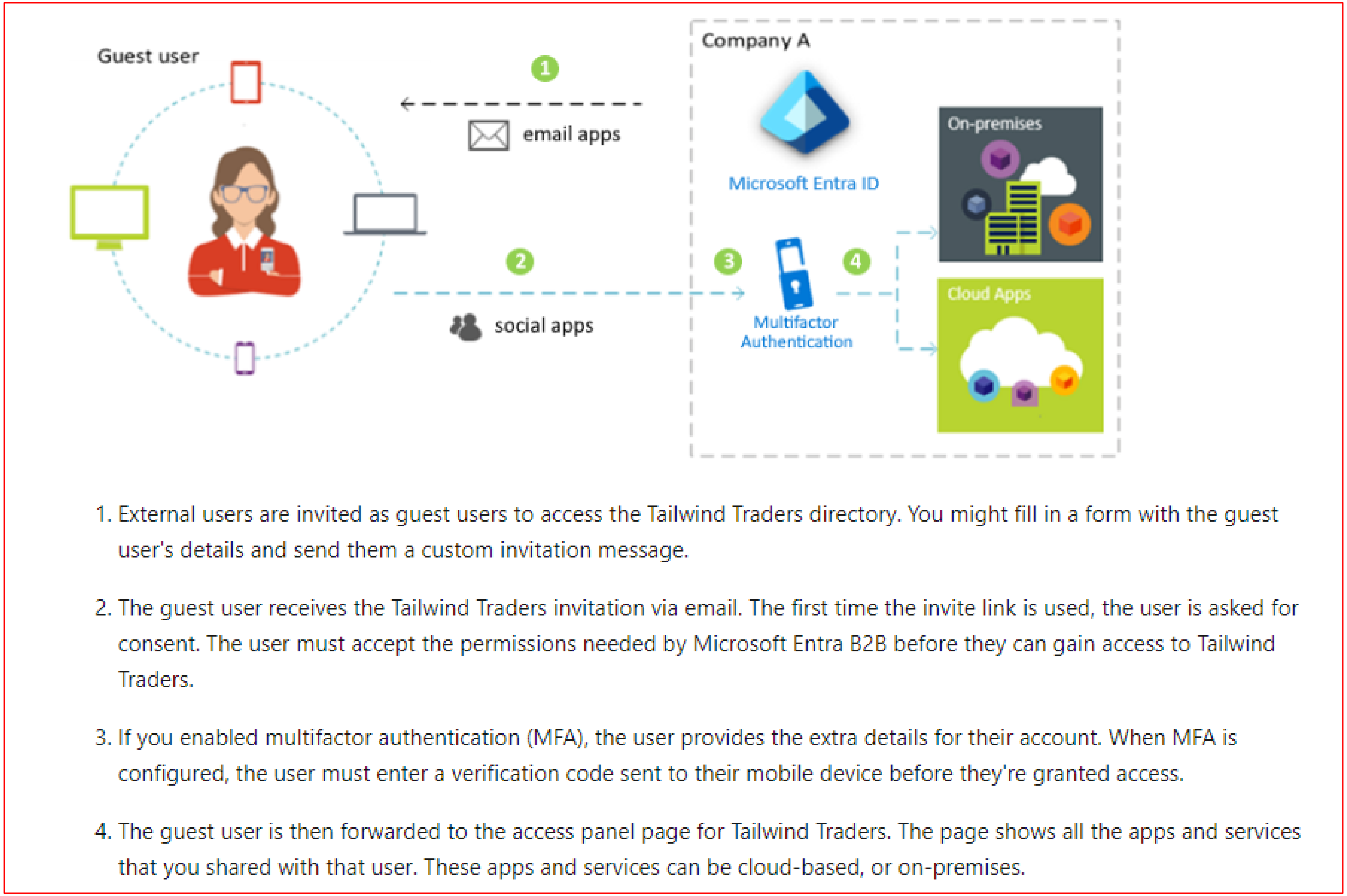
Invites

Active Directory Domain Services

Microsoft Entra ID

- Internal users
- On-premises users
- Guest users (B2B)

Business to Business (B2B)

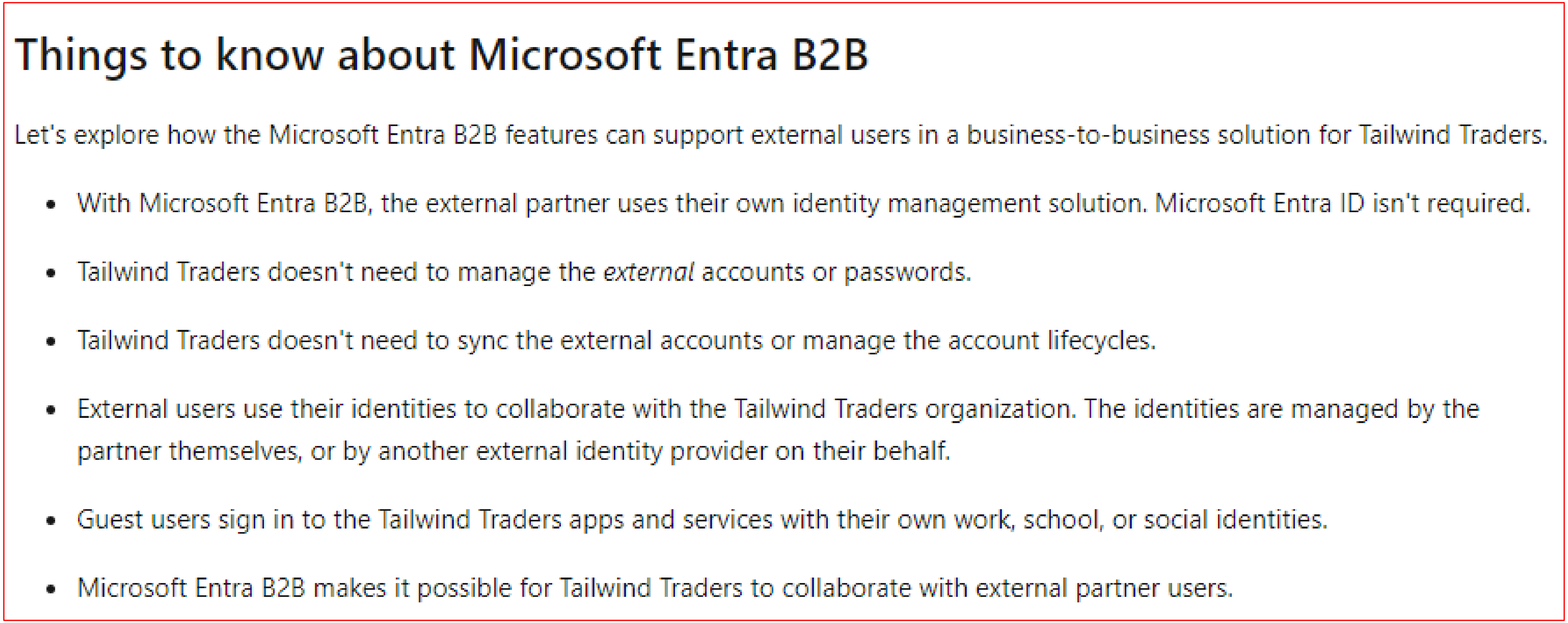


1. External users are invited as guest users to access the Tailwind Traders directory. You might fill in a form with the guest user's details and send them a custom invitation message.

2. The guest user receives the Tailwind Traders invitation via email. The first time the invite link is used, the user is asked for consent. The user must accept the permissions needed by Microsoft Entra B2B before they can gain access to Tailwind Traders.

3. If you enabled multifactor authentication (MFA), the user provides the extra details for their account. When MFA is configured, the user must enter a verification code sent to their mobile device before they're granted access.

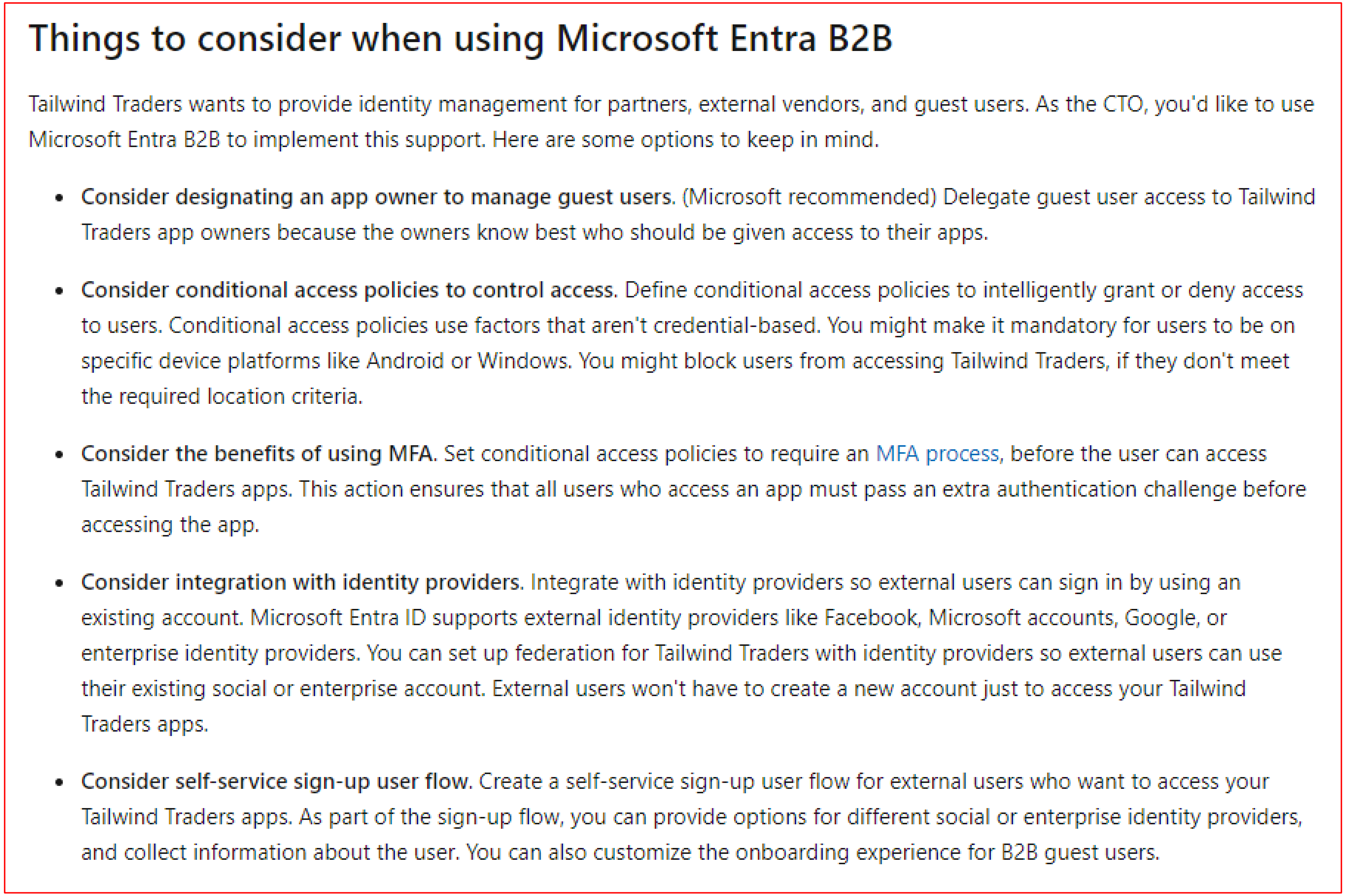
4. The guest user is then forwarded to the access panel page for Tailwind Traders. The page shows all the apps and services that you shared with that user. These apps and services can be cloud-based, or on-premises.



Things to know about Microsoft Entra B2B

Let's explore how the Microsoft Entra B2B features can support external users in a business-to-business solution for Tailwind Traders.

- With Microsoft Entra B2B, the external partner uses their own identity management solution. Microsoft Entra ID isn't required.
- Tailwind Traders doesn't need to manage the *external* accounts or passwords.
- Tailwind Traders doesn't need to sync the external accounts or manage the account lifecycles.
- External users use their identities to collaborate with the Tailwind Traders organization. The identities are managed by the partner themselves, or by another external identity provider on their behalf.
- Guest users sign in to the Tailwind Traders apps and services with their own work, school, or social identities.
- Microsoft Entra B2B makes it possible for Tailwind Traders to collaborate with external partner users.



Things to consider when using Microsoft Entra B2B

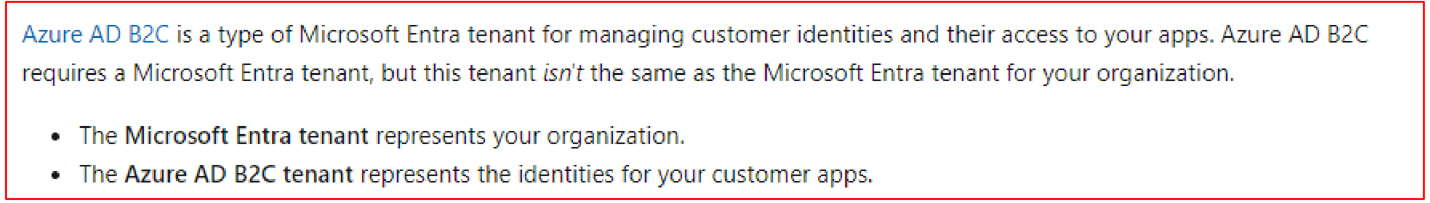
Tailwind Traders wants to provide identity management for partners, external vendors, and guest users. As the CTO, you'd like to use Microsoft Entra B2B to implement this support. Here are some options to keep in mind.

- **Consider designating an app owner to manage guest users**. (Microsoft recommended) Delegate guest user access to Tailwind Traders app owners because the owners know best who should be given access to their apps.
- **Consider conditional access policies to control access.** Define conditional access policies to intelligently grant or deny access to users. Conditional access policies use factors that aren't credential-based. You might make it mandatory for users to be on specific device platforms like Android or Windows. You might block users from accessing Tailwind Traders, if they don't meet the required location criteria.
- **Consider the benefits of using MFA**. Set conditional access policies to require an MFA process, before the user can access Tailwind Traders apps. This action ensures that all users who access an app must pass an extra authentication challenge before accessing the app.
- **Consider integration with identity providers.** Integrate with identity providers so external users can sign in by using an existing account. Microsoft Entra ID supports external identity providers like Facebook, Microsoft accounts, Google, or enterprise identity providers. You can set up federation for Tailwind Traders with identity providers so external users can use their existing social or enterprise account. External users won't have to create a new account just to access your Tailwind Traders apps.
- **Consider self-service sign-up user flow.** Create a self-service sign-up user flow for external users who want to access your Tailwind Traders apps. As part of the sign-up flow, you can provide options for different social or enterprise identity providers, and collect information about the user. You can also customize the onboarding experience for B2B guest users.

Design for Azure AD Business to Customer

When to use Azure AD Business to Customer (B2C)

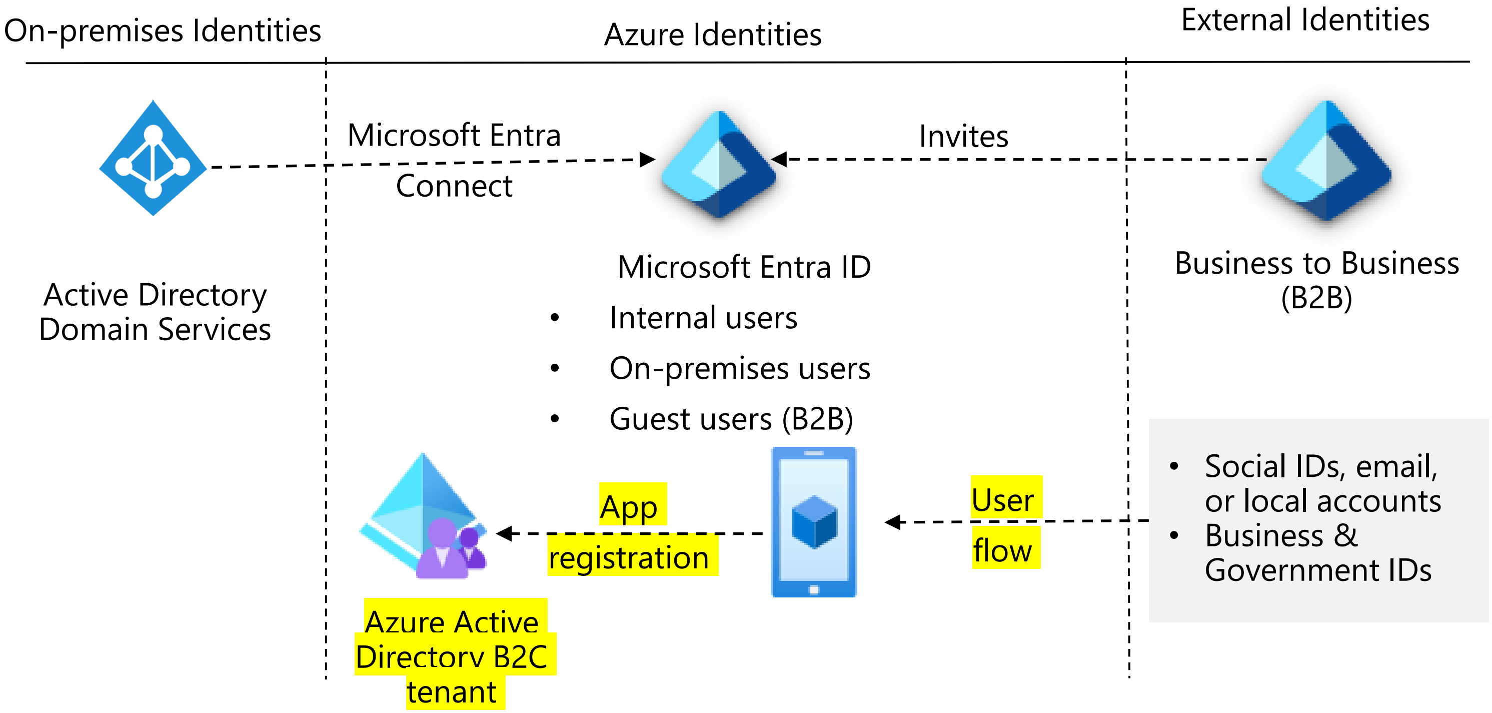
Azure AD B2C is a tenant to manage customer identities and their application access.



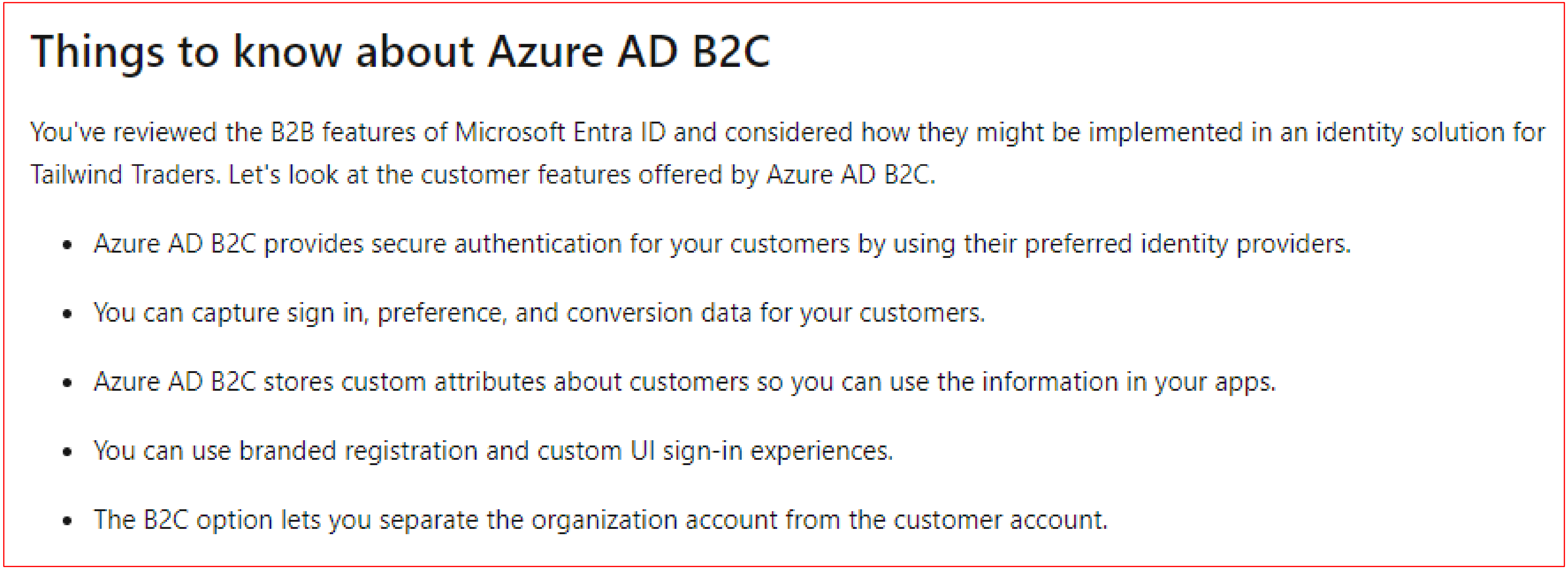
Azure AD B2C is a type of Microsoft Entra tenant for managing customer identities and their access to your apps. Azure AD B2C requires a Microsoft Entra tenant, but this tenant *isn't* the same as the Microsoft Entra tenant for your organization.

- The **Microsoft Entra tenant** represents your organization.
- The **Azure AD B2C tenant** represents the identities for your customer apps.

- Integrate with external user stores
- Provide single sign-on access with a user-provided identity
- Create a custom-branded identity solution
- Use policies to configure user journeys
- Use progressive profiling to gradual collect user information
- Pass user data to a 3rd party for validation



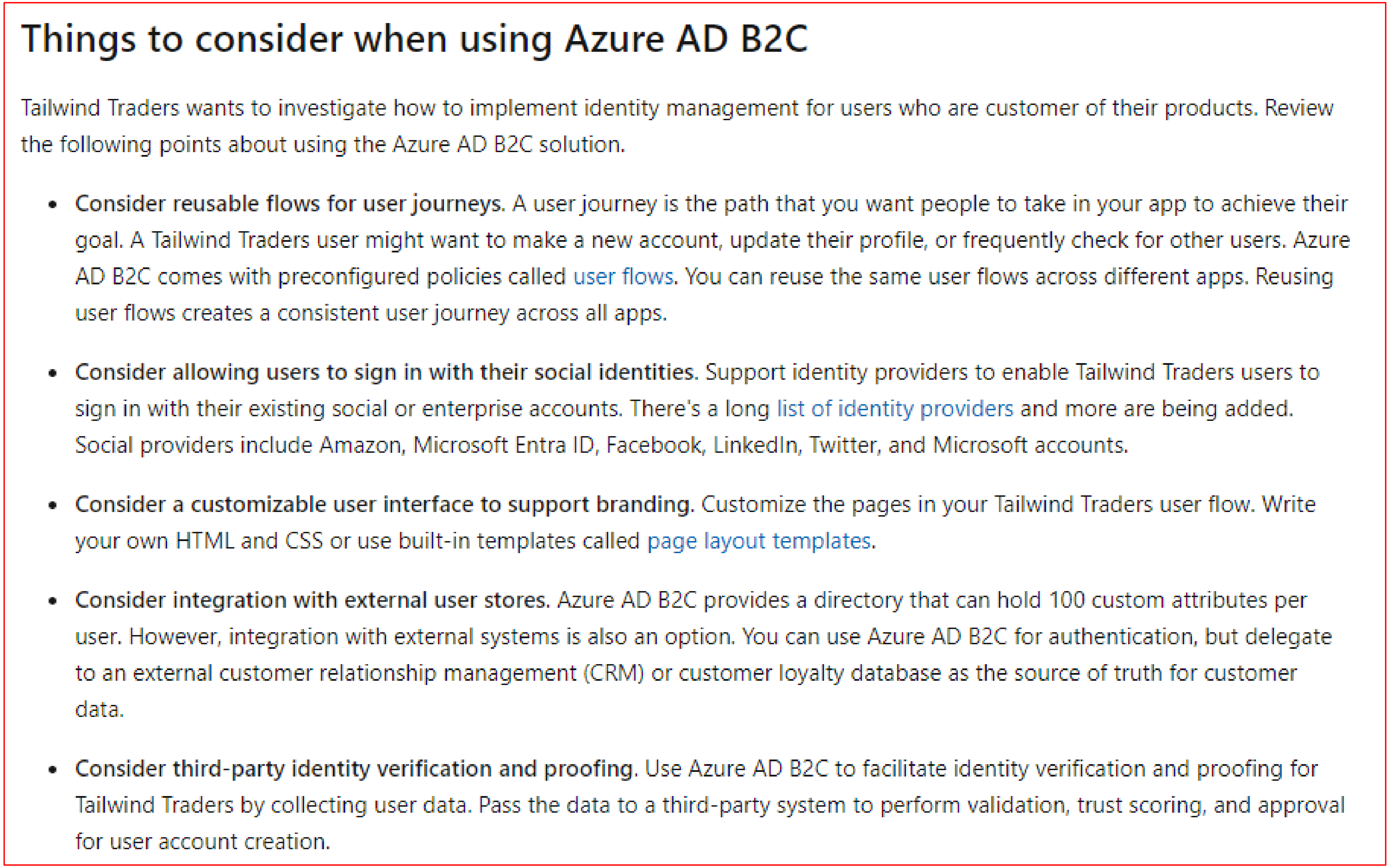
After you set up your Azure AD B2C tenant, you must register your app. You use user flows to manage things like user sign-ins and sign-ups. Your Azure AD B2C tenant lets you create multiple types of user flows.



Things to know about Azure AD B2C

You've reviewed the B2B features of Microsoft Entra ID and considered how they might be implemented in an identity solution for Tailwind Traders. Let's look at the customer features offered by Azure AD B2C.

- Azure AD B2C provides secure authentication for your customers by using their preferred identity providers.
- You can capture sign in, preference, and conversion data for your customers.
- Azure AD B2C stores custom attributes about customers so you can use the information in your apps.
- You can use branded registration and custom UI sign-in experiences.
- The B2C option lets you separate the organization account from the customer account.



Things to consider when using Azure AD B2C

Tailwind Traders wants to investigate how to implement identity management for users who are customer of their products. Review the following points about using the Azure AD B2C solution.

- **Consider reusable flows for user journeys**. A user journey is the path that you want people to take in your app to achieve their goal. A Tailwind Traders user might want to make a new account, update their profile, or frequently check for other users. Azure AD B2C comes with preconfigured policies called user flows. You can reuse the same user flows across different apps. Reusing user flows creates a consistent user journey across all apps.

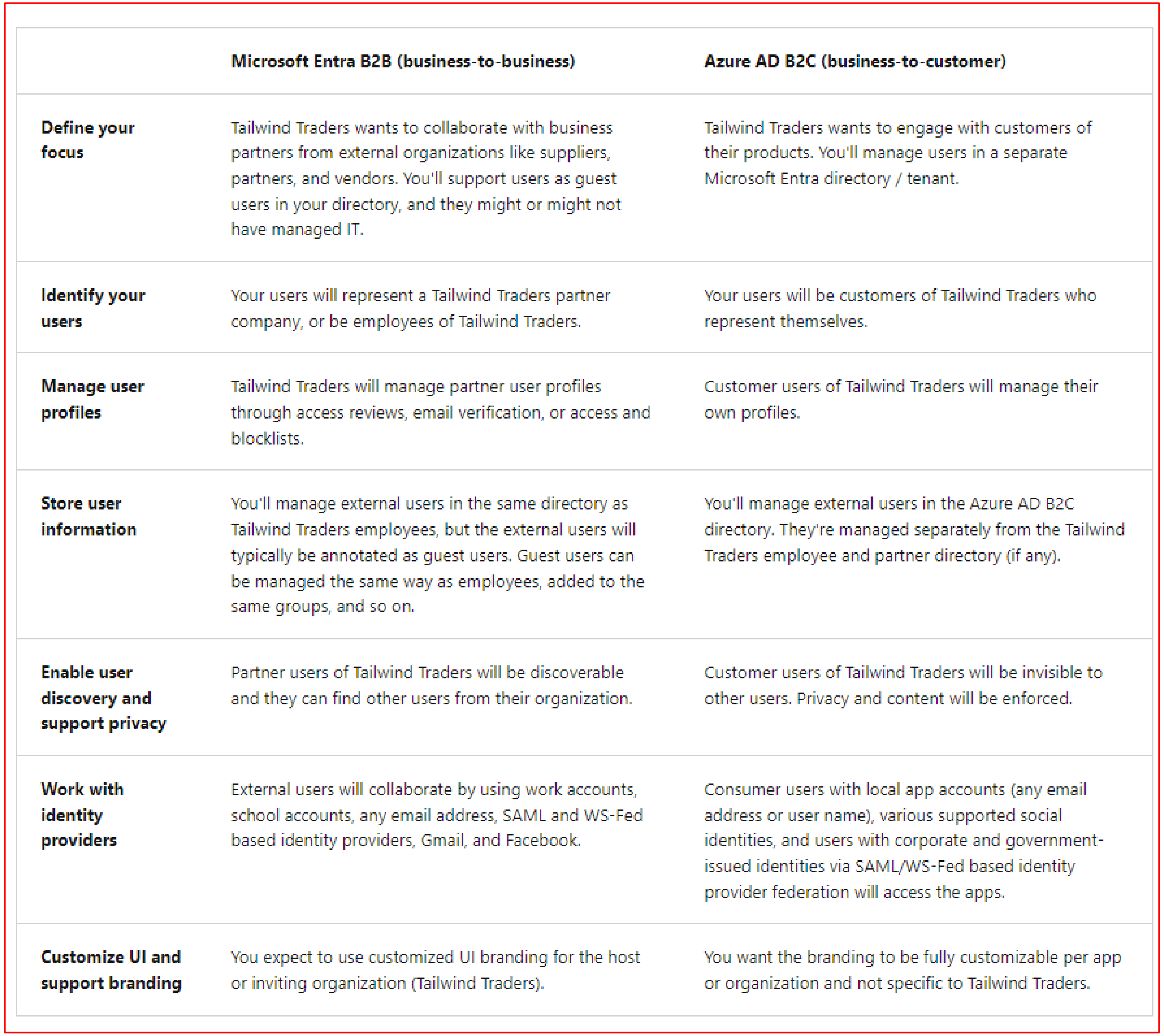
- **Consider allowing users to sign in with their social identities.** Support identity providers to enable Tailwind Traders users to sign in with their existing social or enterprise accounts. There's a long list of identity providers and more are being added. Social providers include Amazon, Microsoft Entra ID, Facebook, LinkedIn, Twitter, and Microsoft accounts.

- **Consider a customizable user interface to support branding.** Customize the pages in your Tailwind Traders user flow. Write your own HTML and CSS or use built-in templates called page layout templates.

- **Consider integration with external user stores.** Azure AD B2C provides a directory that can hold 100 custom attributes per user. However, integration with external systems is also an option. You can use Azure AD B2C for authentication, but delegate to an external customer relationship management (CRM) or customer loyalty database as the source of truth for customer data.

- **Consider third-party identity verification and proofing**. Use Azure AD B2C to facilitate identity verification and proofing for Tailwind Traders by collecting user data. Pass the data to a third-party system to perform validation, trust scoring, and approval for user account creation.

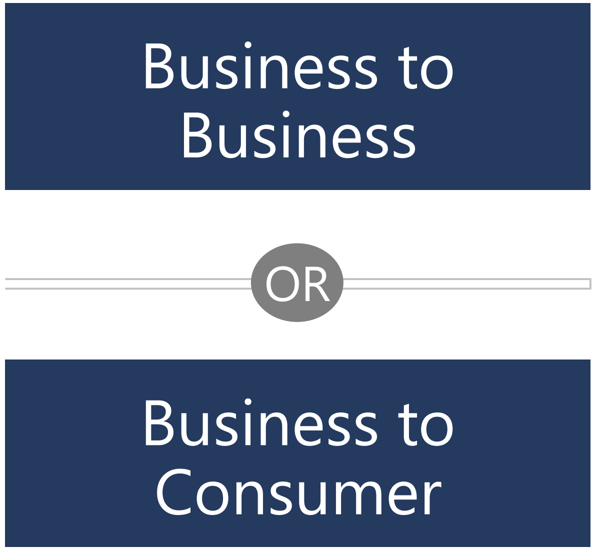
Compare B2B to B2C identity solutions



	Microsoft Entra B2B (business-to-business)	Azure AD B2C (business-to-customer)
Define your focus	Tailwind Traders wants to collaborate with business partners from external organizations like suppliers, partners, and vendors. You'll support users as guest users in your directory, and they might or might not have managed IT.	Tailwind Traders wants to engage with customers of their products. You'll manage users in a separate Microsoft Entra directory / tenant.
Identify your users	Your users will represent a Tailwind Traders partner company, or be employees of Tailwind Traders.	Your users will be customers of Tailwind Traders who represent themselves.
Manage user profiles	Tailwind Traders will manage partner user profiles through access reviews, email verification, or access and blocklists.	Customer users of Tailwind Traders will manage their own profiles.
Store user information	You'll manage external users in the same directory as Tailwind Traders employees, but the external users will typically be annotated as guest users. Guest users can be managed the same way as employees, added to the same groups, and so on.	You'll manage external users in the Azure AD B2C directory. They're managed separately from the Tailwind Traders employee and partner directory (if any).
Enable user discovery and support privacy	Partner users of Tailwind Traders will be discoverable and they can find other users from their organization.	Customer users of Tailwind Traders will be invisible to other users. Privacy and content will be enforced.
Work with identity providers	External users will collaborate by using work accounts, school accounts, any email address, SAML and WS-Fed based identity providers, Gmail, and Facebook.	Consumer users with local app accounts (any email address or user name), various supported social identities, and users with corporate and government-issued identities via SAML/WS-Fed based identity provider federation will access the apps.
Customize UI and support branding	You expect to use customized UI branding for the host or inviting organization (Tailwind Traders).	You want the branding to be fully customizable per app or organization and not specific to Tailwind Traders.

Compare solutions (activity)

- Customers cannot be viewed by other users
- Users are managed in a separate Microsoft Entra tenant
- Users need to be able to self-signup for accounts
- Users manage their own profiles
- Users can come from SAML and WS-Fed based identity providers



Business to Business

OR

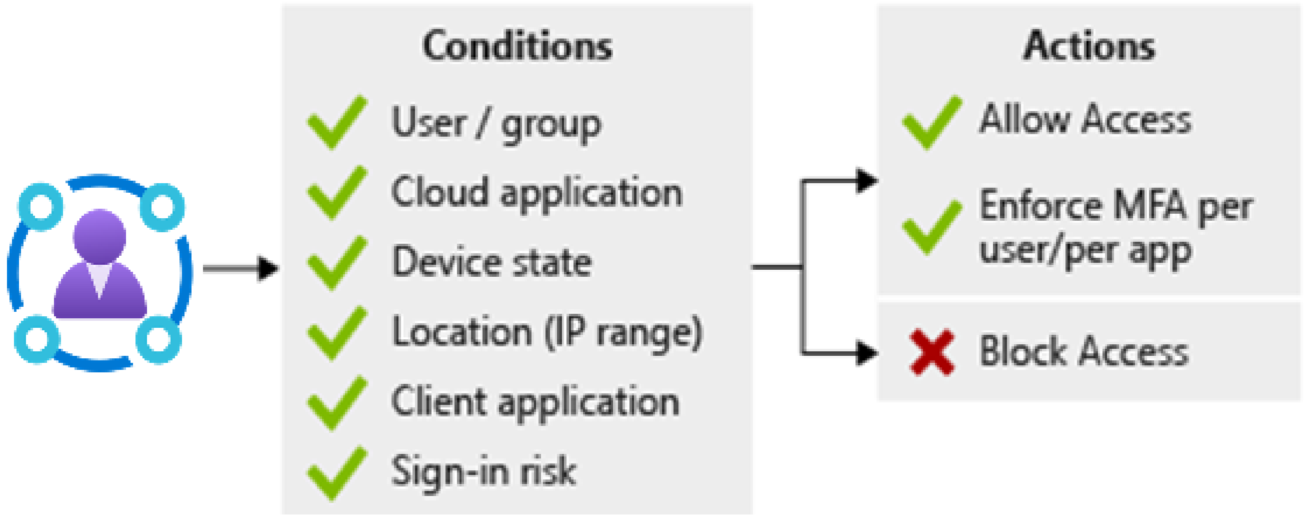
Business to Consumer

Design for conditional access

When to use conditional access

Conditional Access is a Microsoft Entra tool that allows (or denies) access to resources.

- Use to enable multifactor authentication
- Require managed devices
- Access only approved client applications
- Exclude countries from which you never expect a sign in
- Respond to potentially compromised accounts.
- Completely block access
- Block legacy authentication protocols.
- Test using the report-only mode



Conditions

✓ User / group

✓ Cloud application

✓ Device state

✓ Location (IP range)

✓ Client application

✓ Sign-in risk

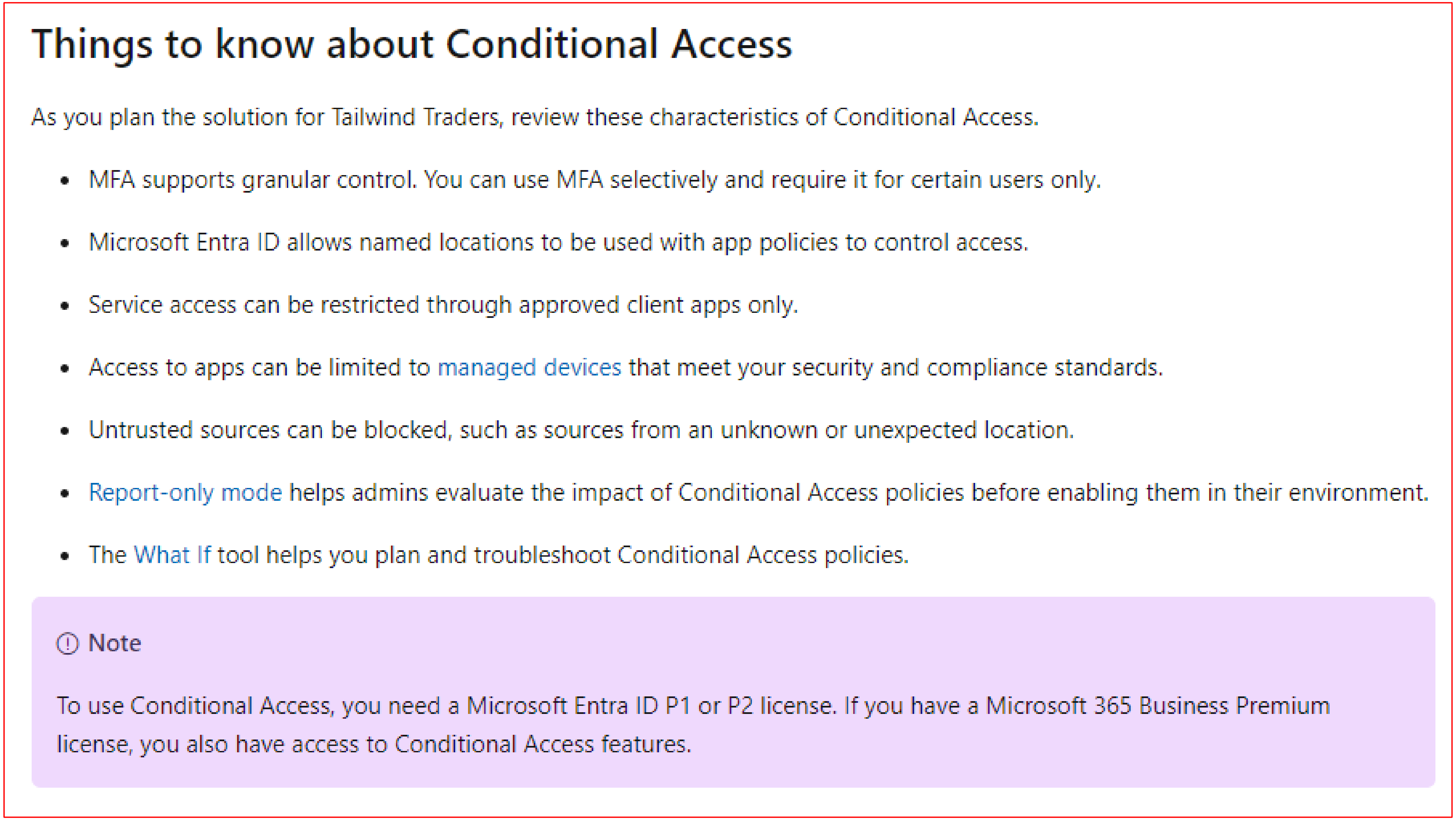
Actions

✓ Allow Access

✓ Enforce MFA per user/per app

✗ Block Access

When a user signs in, Conditional Access examines who the user is, where the user is, and from what device the user is requesting access. Based on these signals, Conditional Access can allow access, enforce multifactor authentication (MFA), or deny access.



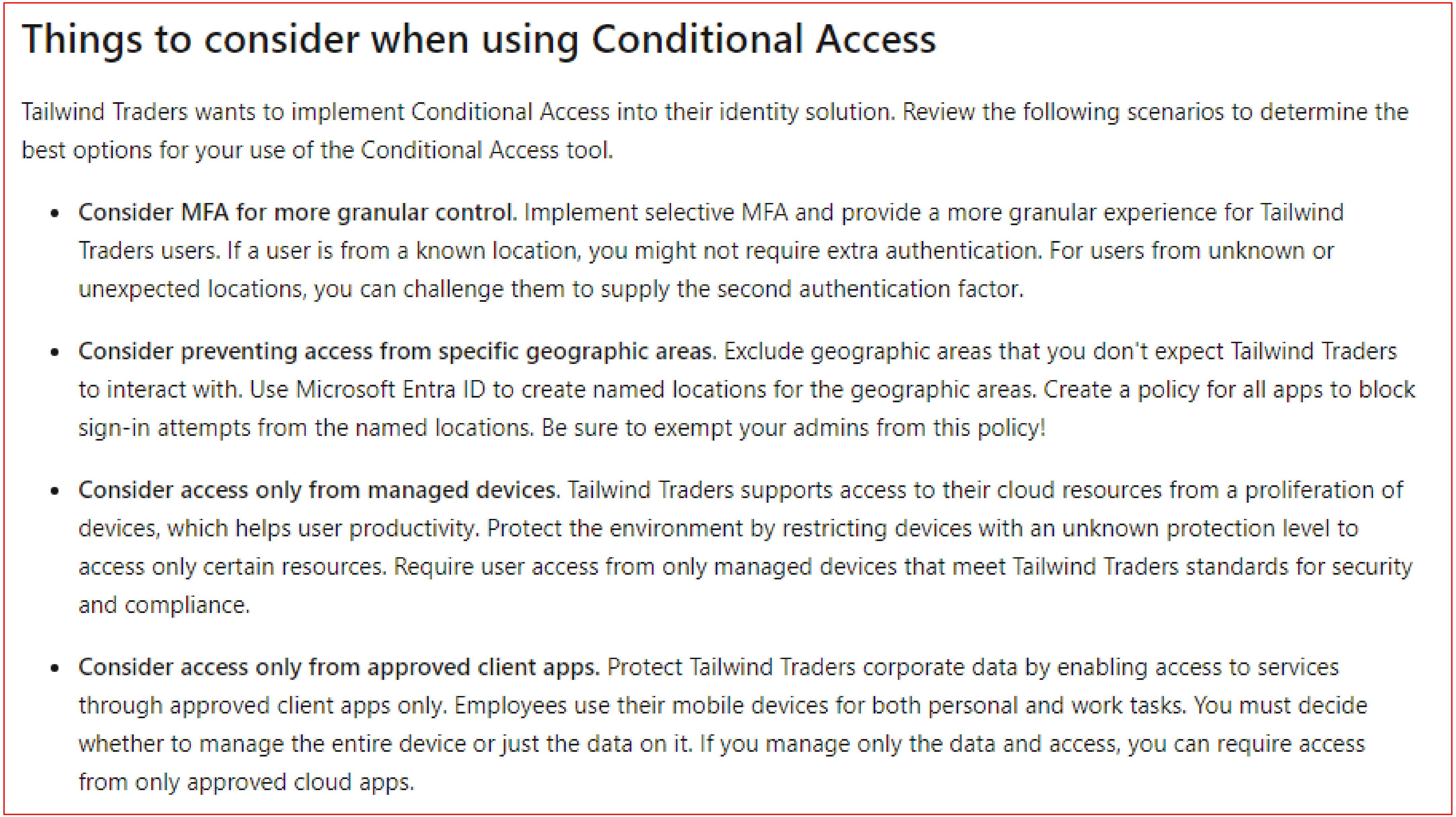
Things to know about Conditional Access

As you plan the solution for Tailwind Traders, review these characteristics of Conditional Access.

- MFA supports granular control. You can use MFA selectively and require it for certain users only.
- Microsoft Entra ID allows named locations to be used with app policies to control access.
- Service access can be restricted through approved client apps only.
- Access to apps can be limited to managed devices that meet your security and compliance standards.
- Untrusted sources can be blocked, such as sources from an unknown or unexpected location.
- Report-only mode helps admins evaluate the impact of Conditional Access policies before enabling them in their environment.
- The What If tool helps you plan and troubleshoot Conditional Access policies.

ⓘ Note

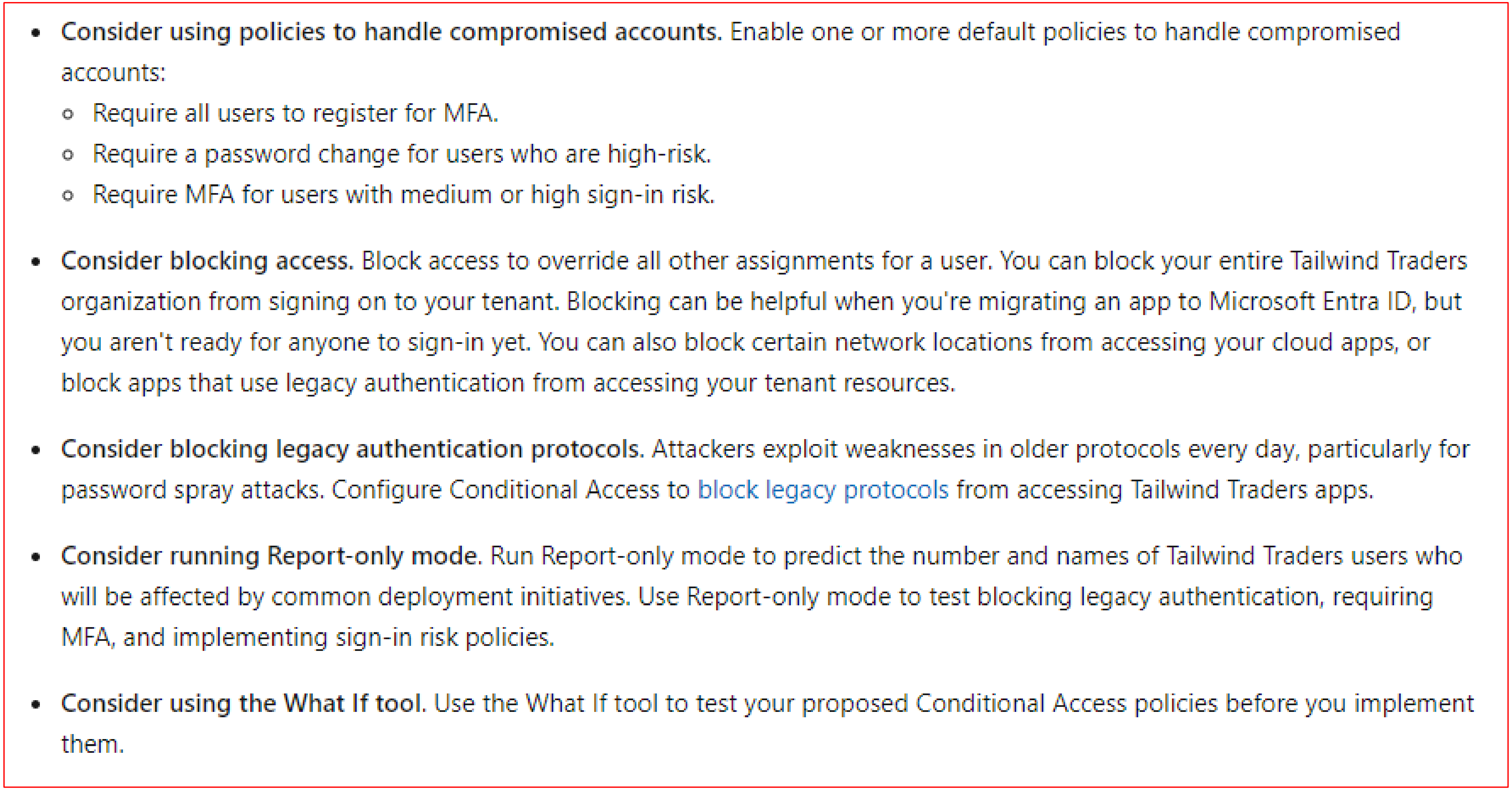
To use Conditional Access, you need a Microsoft Entra ID P1 or P2 license. If you have a Microsoft 365 Business Premium license, you also have access to Conditional Access features.



Things to consider when using Conditional Access

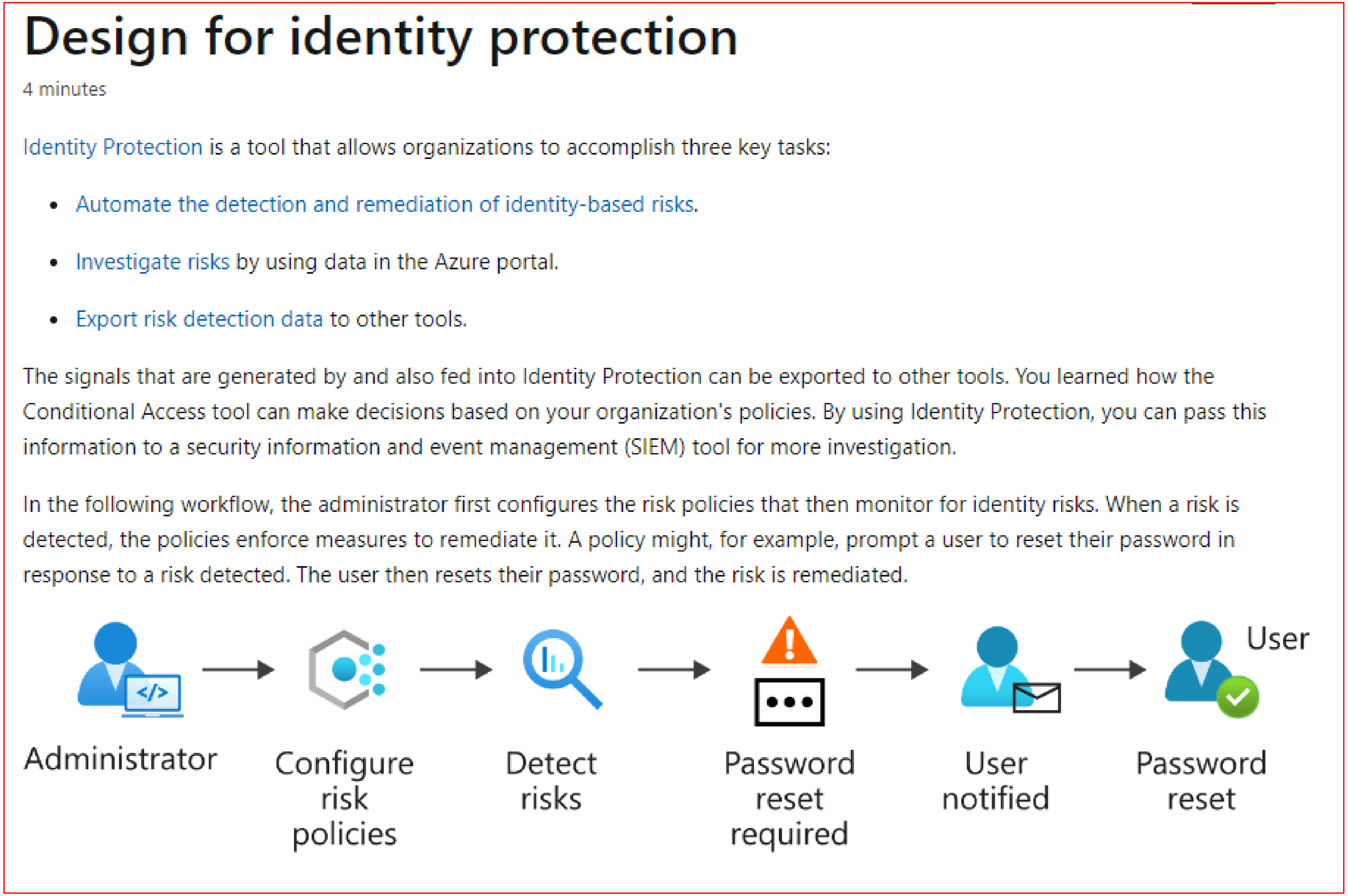
Tailwind Traders wants to implement Conditional Access into their identity solution. Review the following scenarios to determine the best options for your use of the Conditional Access tool.

- **Consider MFA for more granular control.** Implement selective MFA and provide a more granular experience for Tailwind Traders users. If a user is from a known location, you might not require extra authentication. For users from unknown or unexpected locations, you can challenge them to supply the second authentication factor.
- **Consider preventing access from specific geographic areas.** Exclude geographic areas that you don't expect Tailwind Traders to interact with. Use Microsoft Entra ID to create named locations for the geographic areas. Create a policy for all apps to block sign-in attempts from the named locations. Be sure to exempt your admins from this policy!
- **Consider access only from managed devices.** Tailwind Traders supports access to their cloud resources from a proliferation of devices, which helps user productivity. Protect the environment by restricting devices with an unknown protection level to access only certain resources. Require user access from only managed devices that meet Tailwind Traders standards for security and compliance.
- **Consider access only from approved client apps.** Protect Tailwind Traders corporate data by enabling access to services through approved client apps only. Employees use their mobile devices for both personal and work tasks. You must decide whether to manage the entire device or just the data on it. If you manage only the data and access, you can require access from only approved cloud apps.



- **Consider using policies to handle compromised accounts.** Enable one or more default policies to handle compromised accounts:
  - Require all users to register for MFA.
  - Require a password change for users who are high-risk.
  - Require MFA for users with medium or high sign-in risk.
- **Consider blocking access.** Block access to override all other assignments for a user. You can block your entire Tailwind Traders organization from signing on to your tenant. Blocking can be helpful when you're migrating an app to Microsoft Entra ID, but you aren't ready for anyone to sign-in yet. You can also block certain network locations from accessing your cloud apps, or block apps that use legacy authentication from accessing your tenant resources.
- **Consider blocking legacy authentication protocols.** Attackers exploit weaknesses in older protocols every day, particularly for password spray attacks. Configure Conditional Access to block legacy protocols from accessing Tailwind Traders apps.
- **Consider running Report-only mode.** Run Report-only mode to predict the number and names of Tailwind Traders users who will be affected by common deployment initiatives. Use Report-only mode to test blocking legacy authentication, requiring MFA, and implementing sign-in risk policies.
- **Consider using the What If tool.** Use the What If tool to test your proposed Conditional Access policies before you implement them.

Design for identity protection



Design for identity protection

4 minutes

Identity Protection is a tool that allows organizations to accomplish three key tasks:

- Automate the detection and remediation of identity-based risks.
- Investigate risks by using data in the Azure portal.
- Export risk detection data to other tools.

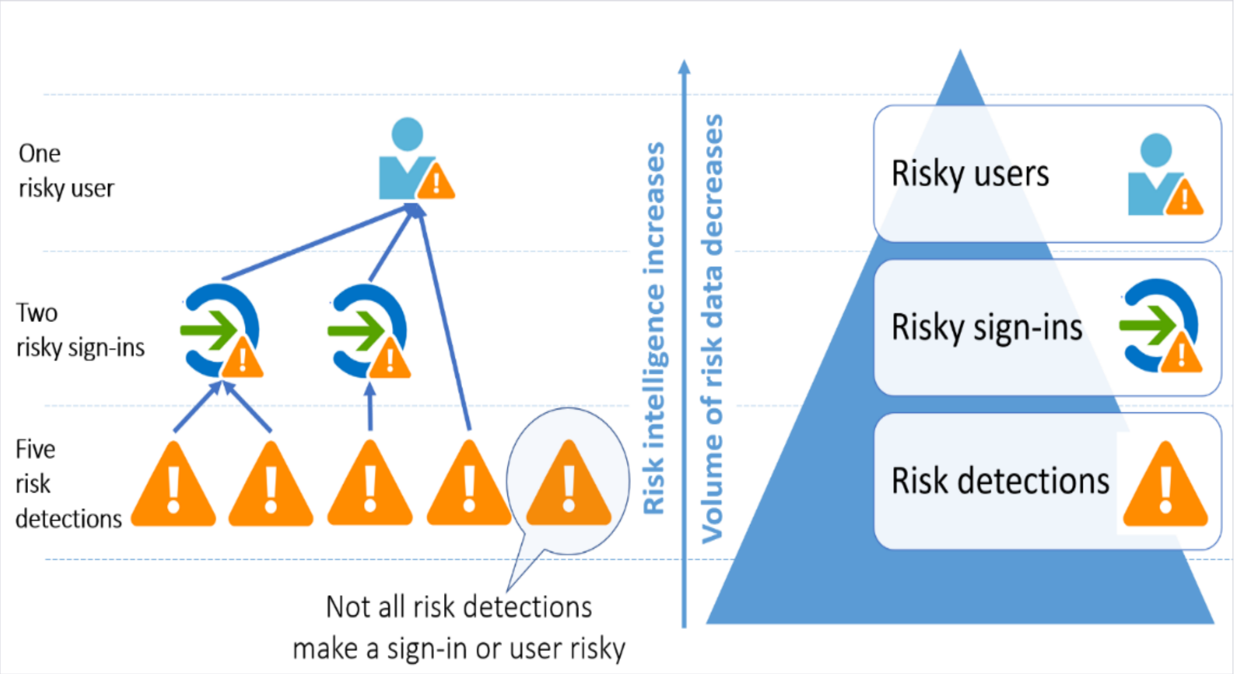
The signals that are generated by and also fed into Identity Protection can be exported to other tools. You learned how the Conditional Access tool can make decisions based on your organization's policies. By using Identity Protection, you can pass this information to a security information and event management (SIEM) tool for more investigation.

In the following workflow, the administrator first configures the risk policies that then monitor for identity risks. When a risk is detected, the policies enforce measures to remediate it. A policy might, for example, prompt a user to reset their password in response to a risk detected. The user then resets their password, and the risk is remediated.

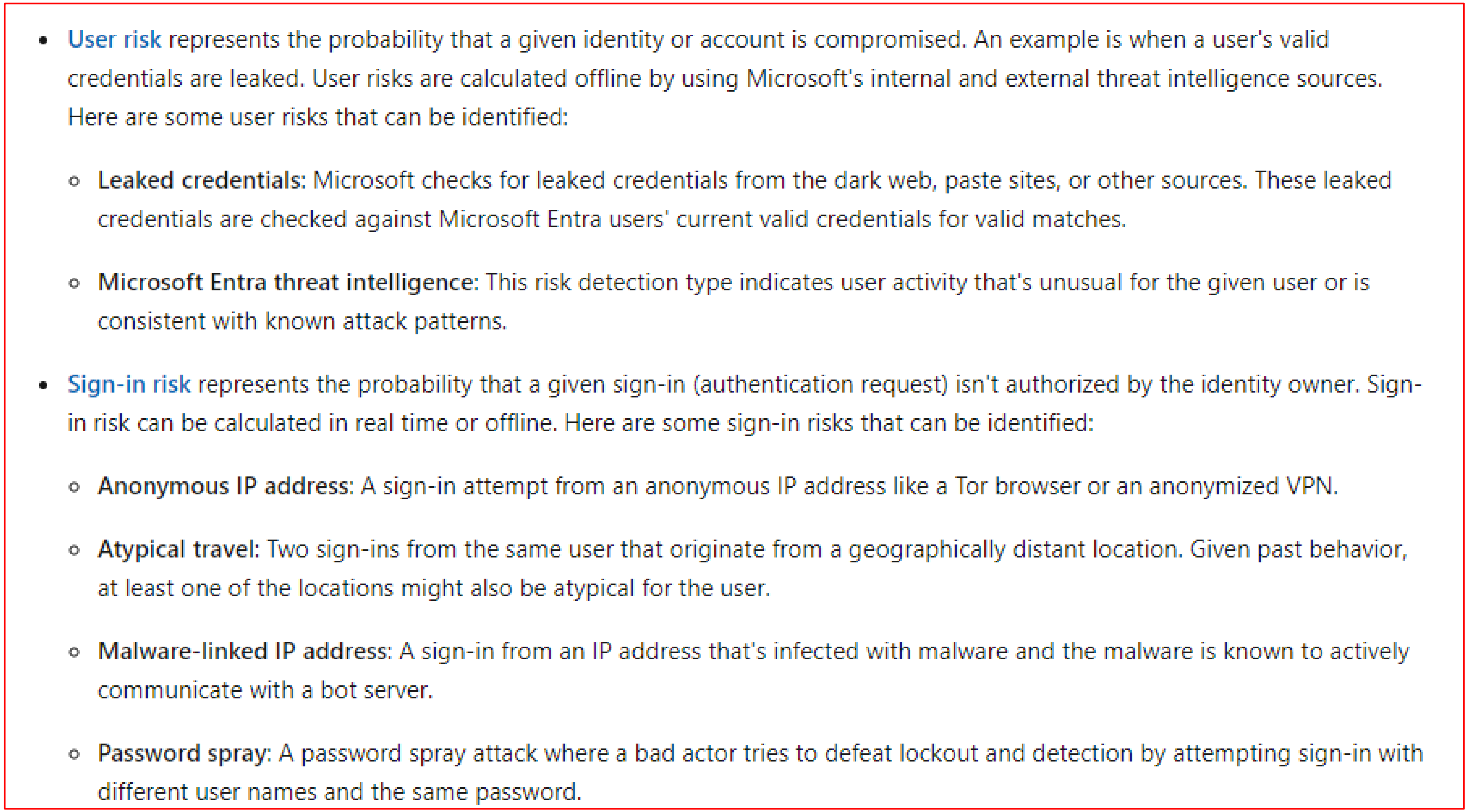
When to use identity protection

Identity protection is a Microsoft Entra tool that automates the detection and remediation of identity-based risks.

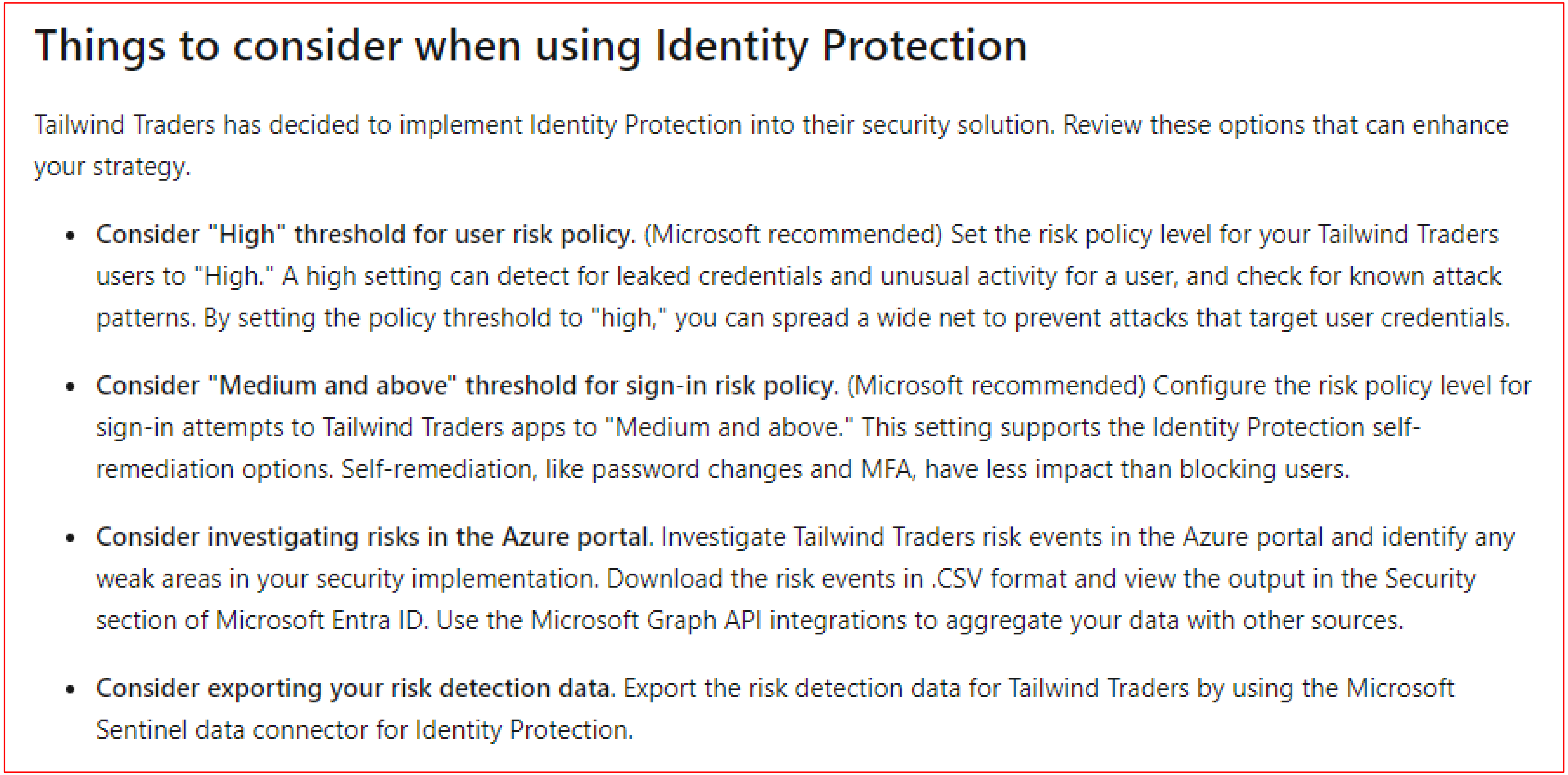
- Configure the policies and actively review the results
- Set the sign-in risk policy to Medium and above and allow self-remediation options
- Set the user risk policy threshold to High
- Allow for excluding users - emergency access or break-glass administrator accounts
- Send data to Conditional Access or other security information and event management (SIEM) tool


One risky user
Two risky sign-ins
Five risk detections
Not all risk detections make a sign-in or user risky
Risk intelligence increases
Volume of risk data decreases
Risky users
Risky sign-ins
Risk detections

Two risk policies are evaluated: user risk and sign-in risk:



- User risk represents the probability that a given identity or account is compromised. An example is when a user's valid credentials are leaked. User risks are calculated offline by using Microsoft's internal and external threat intelligence sources. Here are some user risks that can be identified:
  - **Leaked credentials:** Microsoft checks for leaked credentials from the dark web, paste sites, or other sources. These leaked credentials are checked against Microsoft Entra users' current valid credentials for valid matches.
  - **Microsoft Entra threat intelligence:** This risk detection type indicates user activity that's unusual for the given user or is consistent with known attack patterns.
- Sign-in risk represents the probability that a given sign-in (authentication request) isn't authorized by the identity owner. Sign-in risk can be calculated in real time or offline. Here are some sign-in risks that can be identified:
  - **Anonymous IP address:** A sign-in attempt from an anonymous IP address like a Tor browser or an anonymized VPN.
  - **Atypical travel:** Two sign-ins from the same user that originate from a geographically distant location. Given past behavior, at least one of the locations might also be atypical for the user.
  - **Malware-linked IP address:** A sign-in from an IP address that's infected with malware and the malware is known to actively communicate with a bot server.
  - **Password spray:** A password spray attack where a bad actor tries to defeat lockout and detection by attempting sign-in with different user names and the same password.



Things to consider when using Identity Protection

Tailwind Traders has decided to implement Identity Protection into their security solution. Review these options that can enhance your strategy.

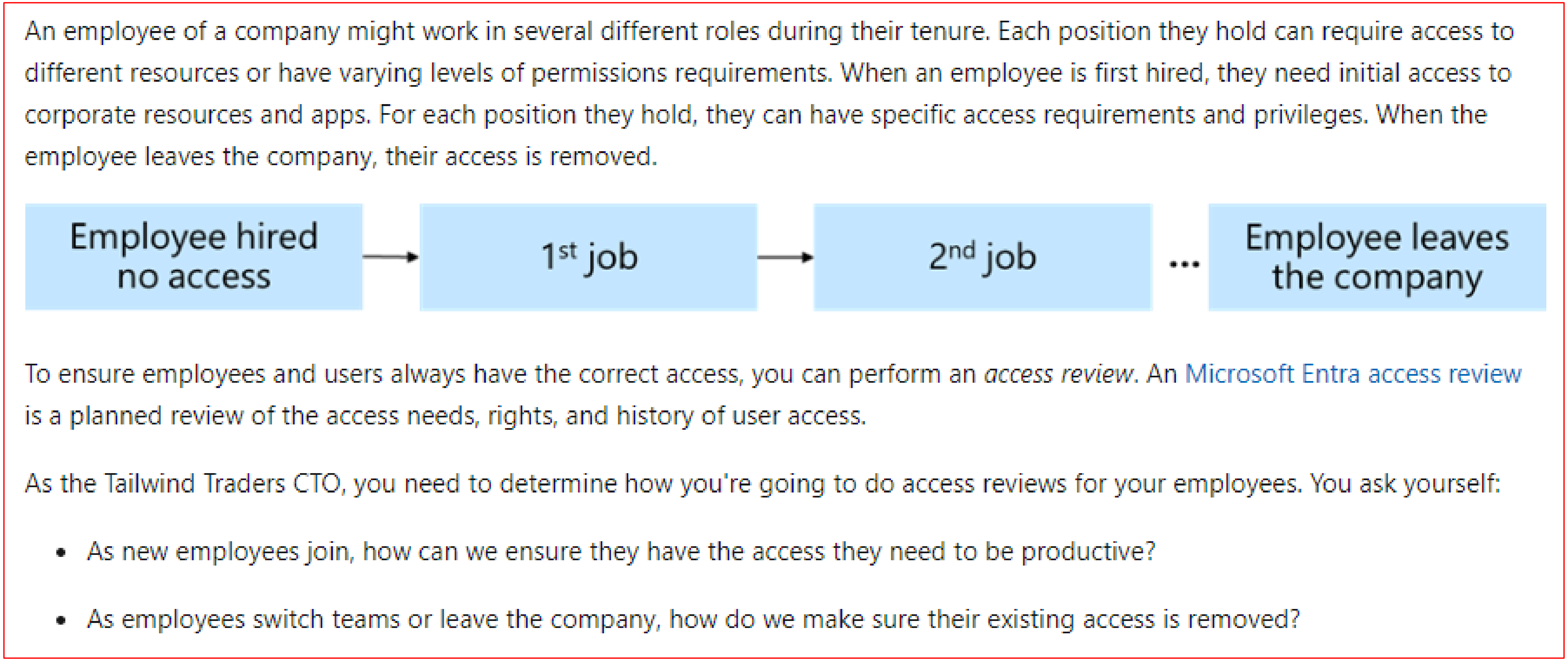
- **Consider "High" threshold for user risk policy.** (Microsoft recommended) Set the risk policy level for your Tailwind Traders users to "High." A high setting can detect for leaked credentials and unusual activity for a user, and check for known attack patterns. By setting the policy threshold to "high," you can spread a wide net to prevent attacks that target user credentials.

- **Consider "Medium and above" threshold for sign-in risk policy.** (Microsoft recommended) Configure the risk policy level for sign-in attempts to Tailwind Traders apps to "Medium and above." This setting supports the Identity Protection self-remediation options. Self-remediation, like password changes and MFA, have less impact than blocking users.

- **Consider investigating risks in the Azure portal.** Investigate Tailwind Traders risk events in the Azure portal and identify any weak areas in your security implementation. Download the risk events in .CSV format and view the output in the Security section of Microsoft Entra ID. Use the Microsoft Graph API integrations to aggregate your data with other sources.

- **Consider exporting your risk detection data.** Export the risk detection data for Tailwind Traders by using the Microsoft Sentinel data connector for Identity Protection.

Design for access reviews



An employee of a company might work in several different roles during their tenure. Each position they hold can require access to different resources or have varying levels of permissions requirements. When an employee is first hired, they need initial access to corporate resources and apps. For each position they hold, they can have specific access requirements and privileges. When the employee leaves the company, their access is removed.

To ensure employees and users always have the correct access, you can perform an *access review*. An Microsoft Entra access review is a planned review of the access needs, rights, and history of user access.

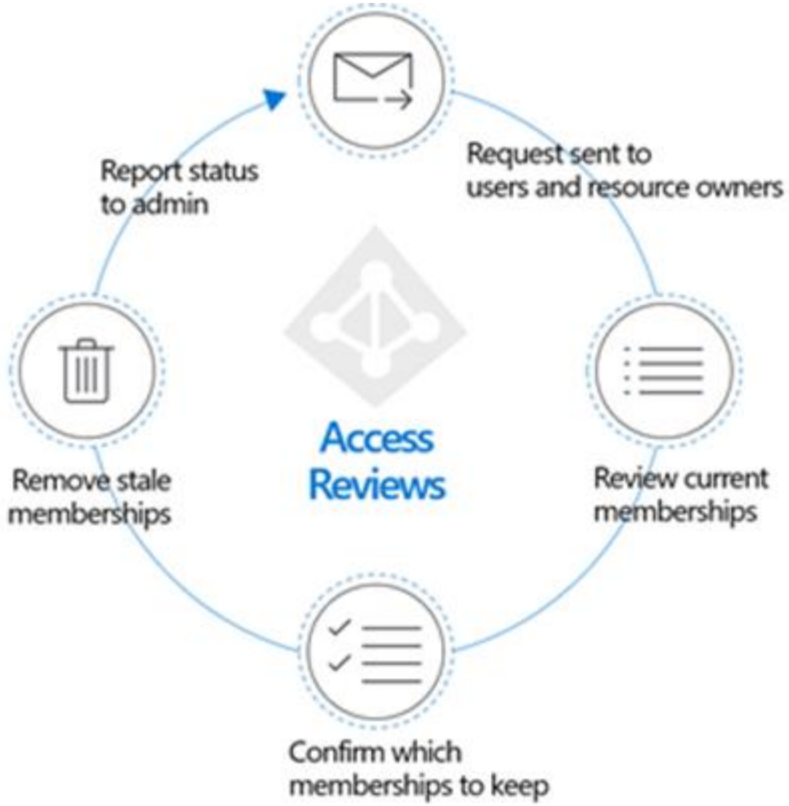
As the Tailwind Traders CTO, you need to determine how you're going to do access reviews for your employees. You ask yourself:

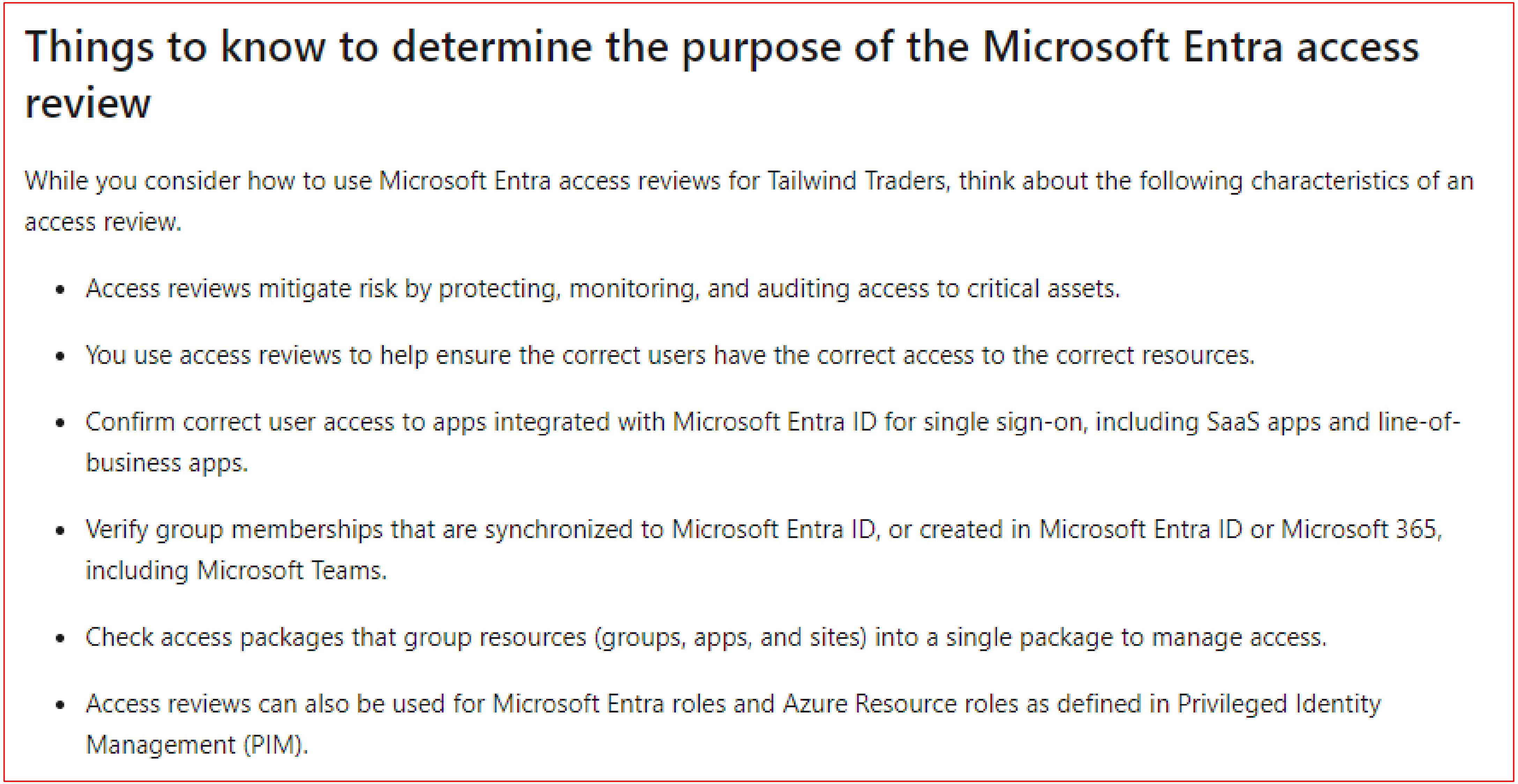
- As new employees join, how can we ensure they have the access they need to be productive?
- As employees switch teams or leave the company, how do we make sure their existing access is removed?

When to use access reviews

Access reviews are a Microsoft Entra tool to review user access and ensure they should have continued access to resources.

- Determine the purpose of the access review
- Engage the right stakeholders
- Create an access review plan
- Determine who will conduct the reviews
- Decide who can self-attest access
- Determine what resource types will be reviewed
- Start small – pilot your plan – keep people informed

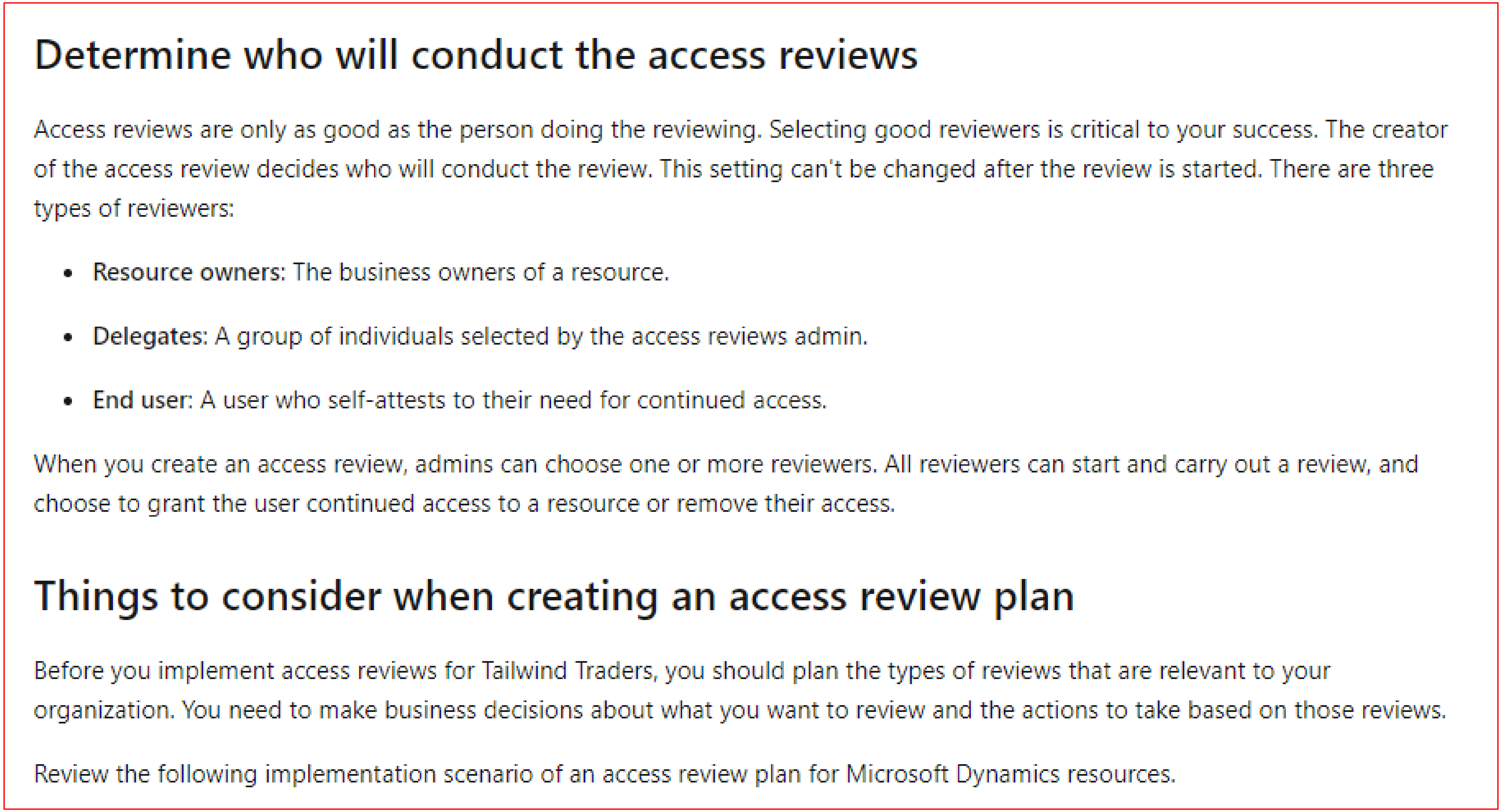

Request sent to users and resource owners
Review current memberships
Confirm which memberships to keep
Remove stale memberships
Report status to admin
Access Reviews



Things to know to determine the purpose of the Microsoft Entra access review

While you consider how to use Microsoft Entra access reviews for Tailwind Traders, think about the following characteristics of an access review.

- Access reviews mitigate risk by protecting, monitoring, and auditing access to critical assets.
- You use access reviews to help ensure the correct users have the correct access to the correct resources.
- Confirm correct user access to apps integrated with Microsoft Entra ID for single sign-on, including SaaS apps and line-of-business apps.
- Verify group memberships that are synchronized to Microsoft Entra ID, or created in Microsoft Entra ID or Microsoft 365, including Microsoft Teams.
- Check access packages that group resources (groups, apps, and sites) into a single package to manage access.
- Access reviews can also be used for Microsoft Entra roles and Azure Resource roles as defined in Privileged Identity Management (PIM).



Determine who will conduct the access reviews

Access reviews are only as good as the person doing the reviewing. Selecting good reviewers is critical to your success. The creator of the access review decides who will conduct the review. This setting can't be changed after the review is started. There are three types of reviewers:

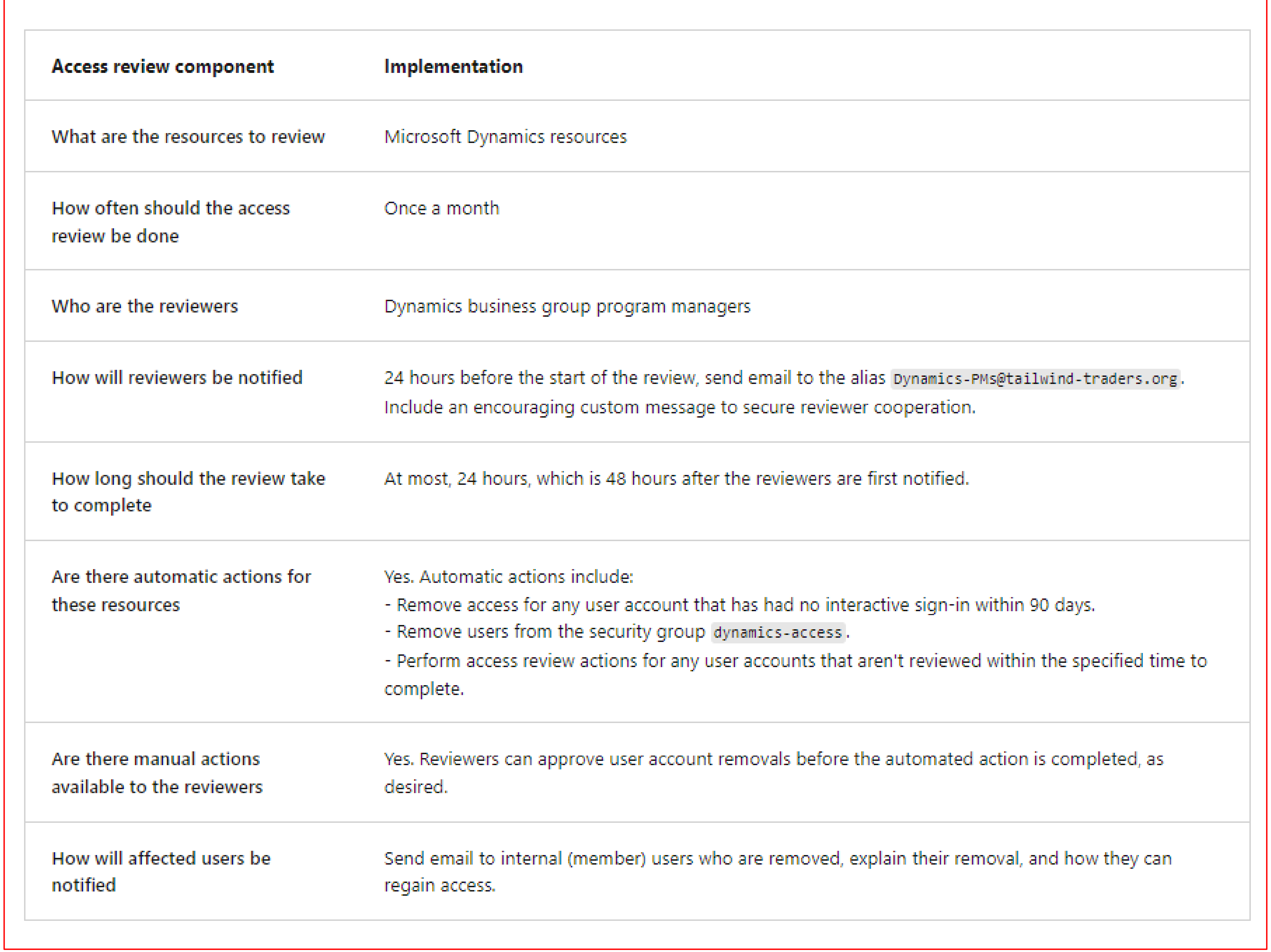
- **Resource owners:** The business owners of a resource.
- **Delegates:** A group of individuals selected by the access reviews admin.
- **End user:** A user who self-attests to their need for continued access.

When you create an access review, admins can choose one or more reviewers. All reviewers can start and carry out a review, and choose to grant the user continued access to a resource or remove their access.

Things to consider when creating an access review plan

Before you implement access reviews for Tailwind Traders, you should plan the types of reviews that are relevant to your organization. You need to make business decisions about what you want to review and the actions to take based on those reviews.

Review the following implementation scenario of an access review plan for Microsoft Dynamics resources.



Access review component	Implementation
What are the resources to review	Microsoft Dynamics resources
How often should the access review be done	Once a month
Who are the reviewers	Dynamics business group program managers
How will reviewers be notified	24 hours before the start of the review, send email to the alias `Dynamics-PMs@tailwind-traders.org`. Include an encouraging custom message to secure reviewer cooperation.
How long should the review take to complete	At most, 24 hours, which is 48 hours after the reviewers are first notified.
Are there automatic actions for these resources	Yes. Automatic actions include: - Remove access for any user account that has had no interactive sign-in within 90 days. - Remove users from the security group `dynamics-access`. - Perform access review actions for any user accounts that aren't reviewed within the specified time to complete.
Are there manual actions available to the reviewers	Yes. Reviewers can approve user account removals before the automated action is completed, as desired.
How will affected users be notified	Send email to internal (member) users who are removed, explain their removal, and how they can regain access.

Design service principals for applications

Design managed identities

Managed identities provide an identity for application authentication.

I want to build an application using

Source

- Azure VMs
- Azure App Service
- Azure Functions
- Azure Container Instances
- Azure Kubernetes Service
- Azure Logic Apps
- …

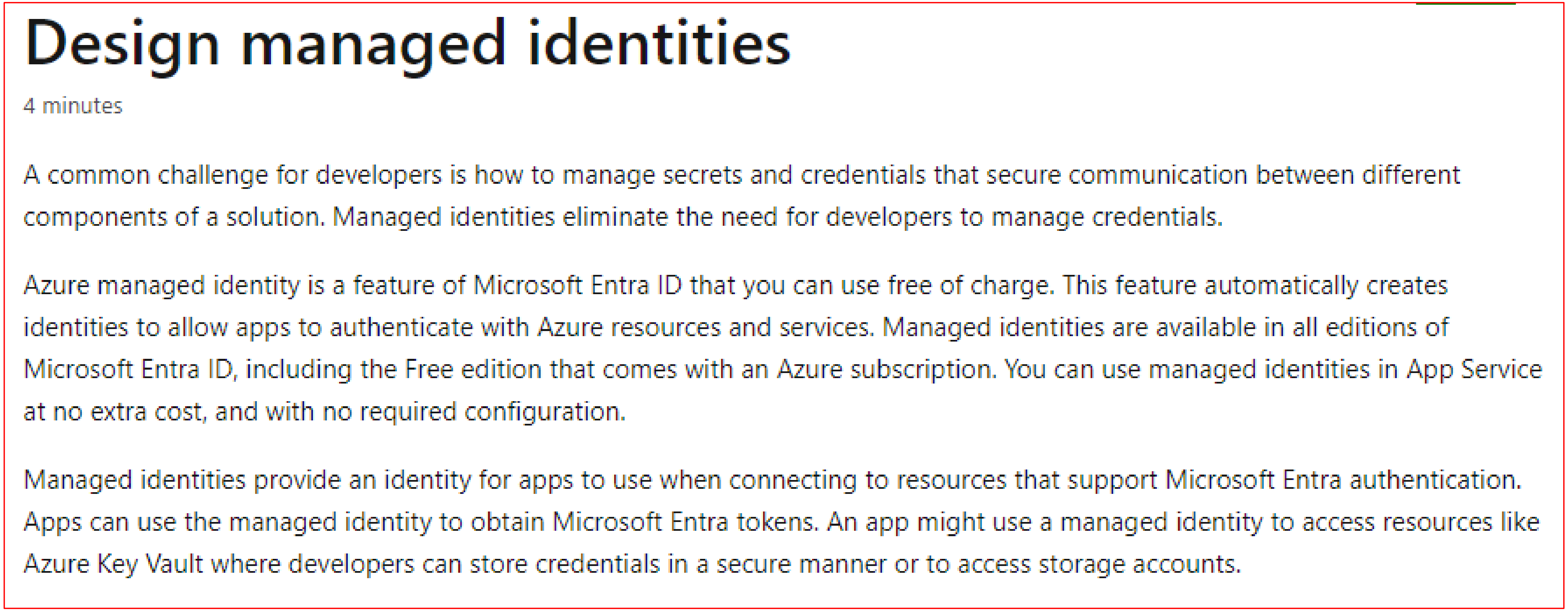
that accesses

Target

Any target that supports Microsoft Entra ID authentication

- Your applications
- Azure services (Azure Key Vault, Azure Storage, Azure SQL, …)

- The source is an Azure resource
- The target supports Microsoft Entra ID authentication and Azure RBAC
- No credential rotation or certificate management



Design managed identities

4 minutes

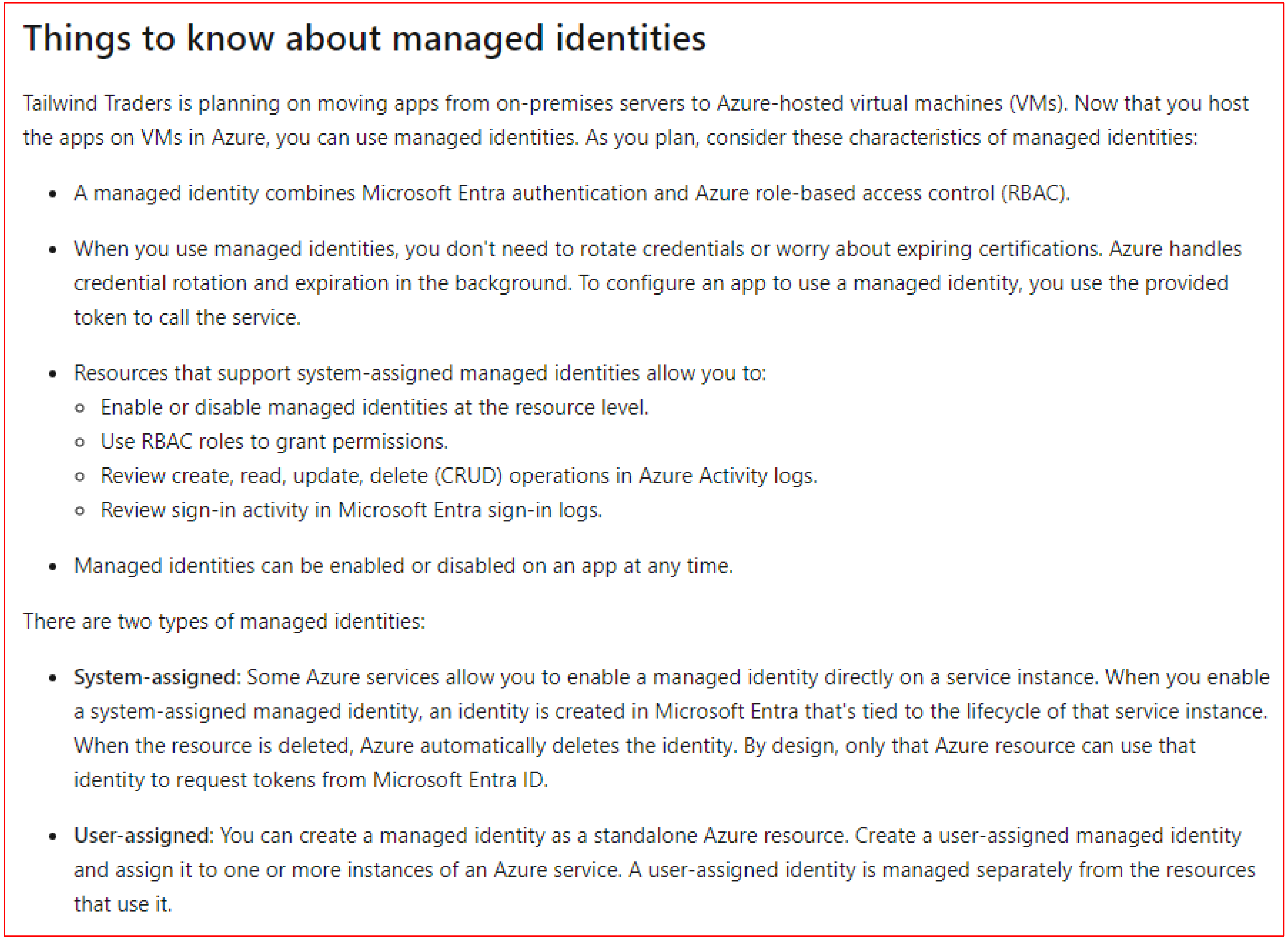
A common challenge for developers is how to manage secrets and credentials that secure communication between different components of a solution. Managed identities eliminate the need for developers to manage credentials.

Azure managed identity is a feature of Microsoft Entra ID that you can use free of charge. This feature automatically creates identities to allow apps to authenticate with Azure resources and services. Managed identities are available in all editions of Microsoft Entra ID, including the Free edition that comes with an Azure subscription. You can use managed identities in App Service at no extra cost, and with no required configuration.

Managed identities provide an identity for apps to use when connecting to resources that support Microsoft Entra authentication. Apps can use the managed identity to obtain Microsoft Entra tokens. An app might use a managed identity to access resources like Azure Key Vault where developers can store credentials in a secure manner or to access storage accounts.

Select managed identities

Property	System-assigned managed identity	User-assigned managed identity
Creation	• Created as part of an Azure resource	• Created as a stand-alone Azure resource
Life cycle	• Shared life cycle with the Azure resource	• Independent life cycle • Must be explicitly deleted
Sharing across Azure resources	• Cannot be shared • Can only be associated with a single Azure resource	• Can be shared • Can be associated with more than one Azure resource
Common use cases	• Workloads that are contained within a single Azure resource • Workloads for which you need independent identities. • For example, an application that runs on a single virtual machine	• Workloads that run on multiple resources and which can share a single identity • Workloads that need pre-authorization to a secure resource as part of a provisioning flow. • Workloads where resources are recycled frequently, but permissions should stay consistent.



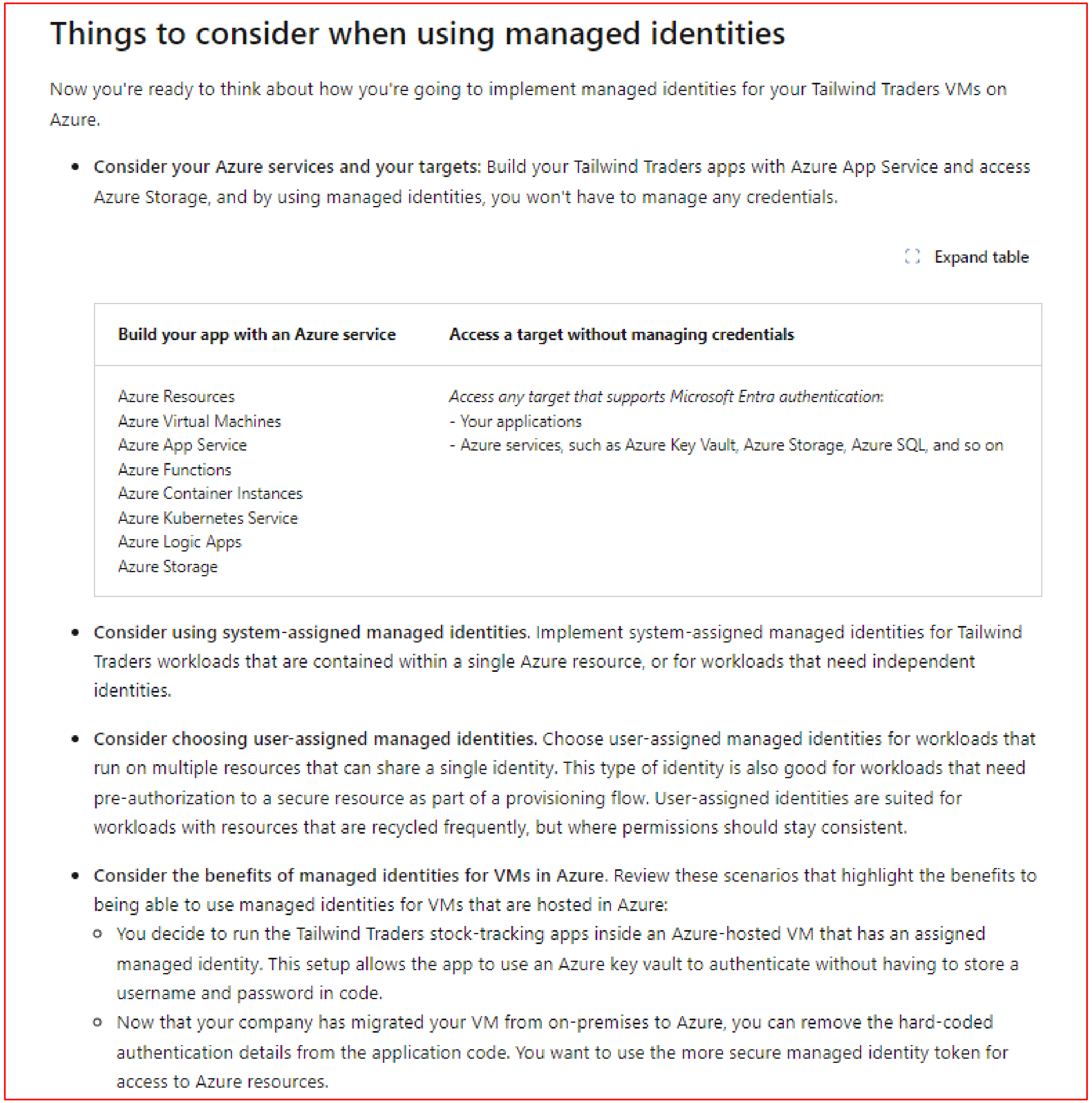
Things to know about managed identities

Tailwind Traders is planning on moving apps from on-premises servers to Azure-hosted virtual machines (VMs). Now that you host the apps on VMs in Azure, you can use managed identities. As you plan, consider these characteristics of managed identities:

- A managed identity combines Microsoft Entra authentication and Azure role-based access control (RBAC).
- When you use managed identities, you don't need to rotate credentials or worry about expiring certifications. Azure handles credential rotation and expiration in the background. To configure an app to use a managed identity, you use the provided token to call the service.
- Resources that support system-assigned managed identities allow you to:
  - Enable or disable managed identities at the resource level.
  - Use RBAC roles to grant permissions.
  - Review create, read, update, delete (CRUD) operations in Azure Activity logs.
  - Review sign-in activity in Microsoft Entra sign-in logs.
- Managed identities can be enabled or disabled on an app at any time.

There are two types of managed identities:

- **System-assigned:** Some Azure services allow you to enable a managed identity directly on a service instance. When you enable a system-assigned managed identity, an identity is created in Microsoft Entra that's tied to the lifecycle of that service instance. When the resource is deleted, Azure automatically deletes the identity. By design, only that Azure resource can use that identity to request tokens from Microsoft Entra ID.
- **User-assigned:** You can create a managed identity as a standalone Azure resource. Create a user-assigned managed identity and assign it to one or more instances of an Azure service. A user-assigned identity is managed separately from the resources that use it.



Things to consider when using managed identities

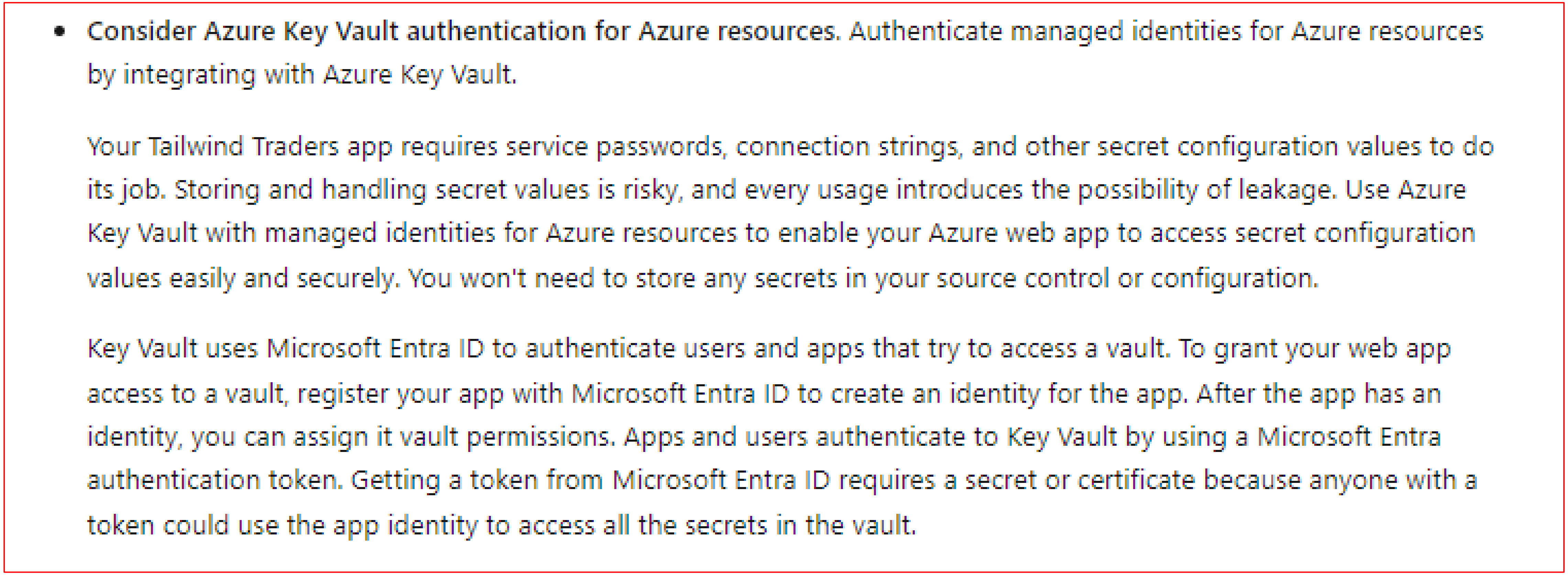
Now you're ready to think about how you're going to implement managed identities for your Tailwind Traders VMs on Azure.

- **Consider your Azure services and your targets:** Build your Tailwind Traders apps with Azure App Service and access Azure Storage, and by using managed identities, you won't have to manage any credentials.

Expand table

Build your app with an Azure service	Access a target without managing credentials
Azure Resources Azure Virtual Machines Azure App Service Azure Functions Azure Container Instances Azure Kubernetes Service Azure Logic Apps Azure Storage	*Access any target that supports Microsoft Entra authentication:* - Your applications - Azure services, such as Azure Key Vault, Azure Storage, Azure SQL, and so on

- **Consider using system-assigned managed identities.** Implement system-assigned managed identities for Tailwind Traders workloads that are contained within a single Azure resource, or for workloads that need independent identities.
- **Consider choosing user-assigned managed identities.** Choose user-assigned managed identities for workloads that run on multiple resources that can share a single identity. This type of identity is also good for workloads that need pre-authorization to a secure resource as part of a provisioning flow. User-assigned identities are suited for workloads with resources that are recycled frequently, but where permissions should stay consistent.
- **Consider the benefits of managed identities for VMs in Azure.** Review these scenarios that highlight the benefits to being able to use managed identities for VMs that are hosted in Azure:
  - You decide to run the Tailwind Traders stock-tracking apps inside an Azure-hosted VM that has an assigned managed identity. This setup allows the app to use an Azure key vault to authenticate without having to store a username and password in code.
  - Now that your company has migrated your VM from on-premises to Azure, you can remove the hard-coded authentication details from the application code. You want to use the more secure managed identity token for access to Azure resources.



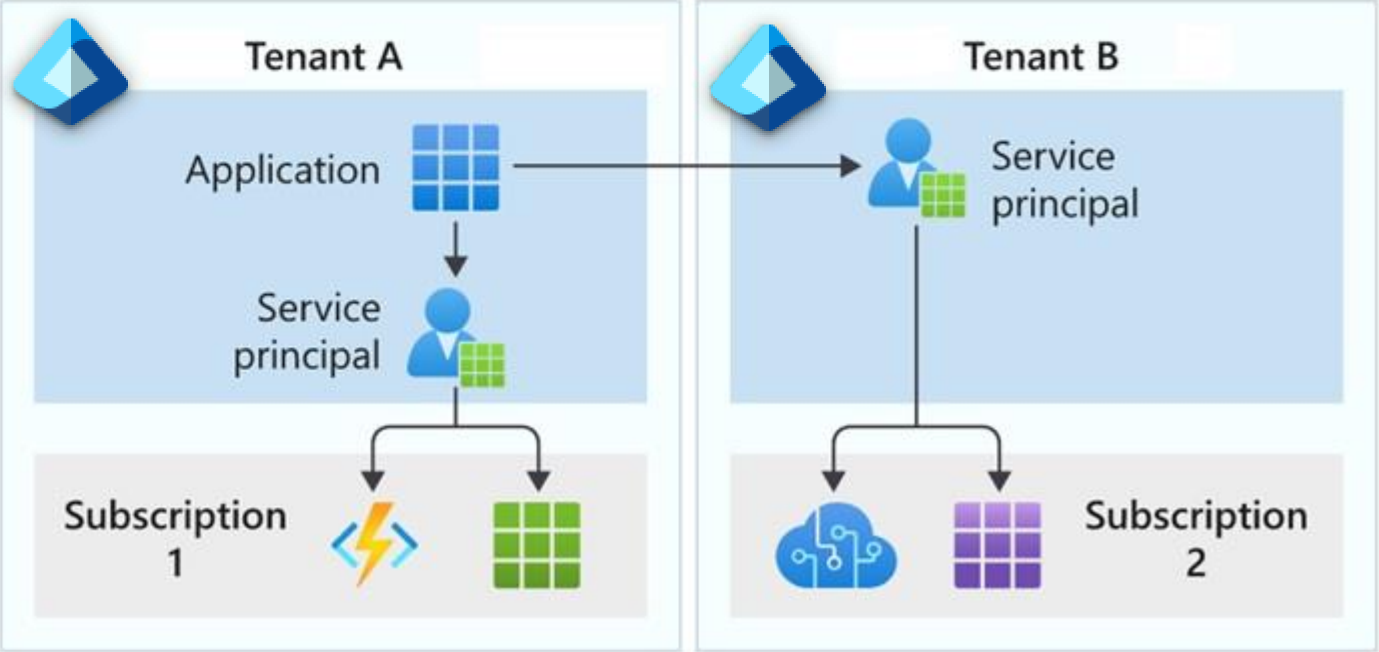
- **Consider Azure Key Vault authentication for Azure resources.** Authenticate managed identities for Azure resources by integrating with Azure Key Vault.

  Your Tailwind Traders app requires service passwords, connection strings, and other secret configuration values to do its job. Storing and handling secret values is risky, and every usage introduces the possibility of leakage. Use Azure Key Vault with managed identities for Azure resources to enable your Azure web app to access secret configuration values easily and securely. You won't need to store any secrets in your source control or configuration.

  Key Vault uses Microsoft Entra ID to authenticate users and apps that try to access a vault. To grant your web app access to a vault, register your app with Microsoft Entra ID to create an identity for the app. After the app has an identity, you can assign it vault permissions. Apps and users authenticate to Key Vault by using a Microsoft Entra authentication token. Getting a token from Microsoft Entra ID requires a secret or certificate because anyone with a token could use the app identity to access all the secrets in the vault.

Select application service principals

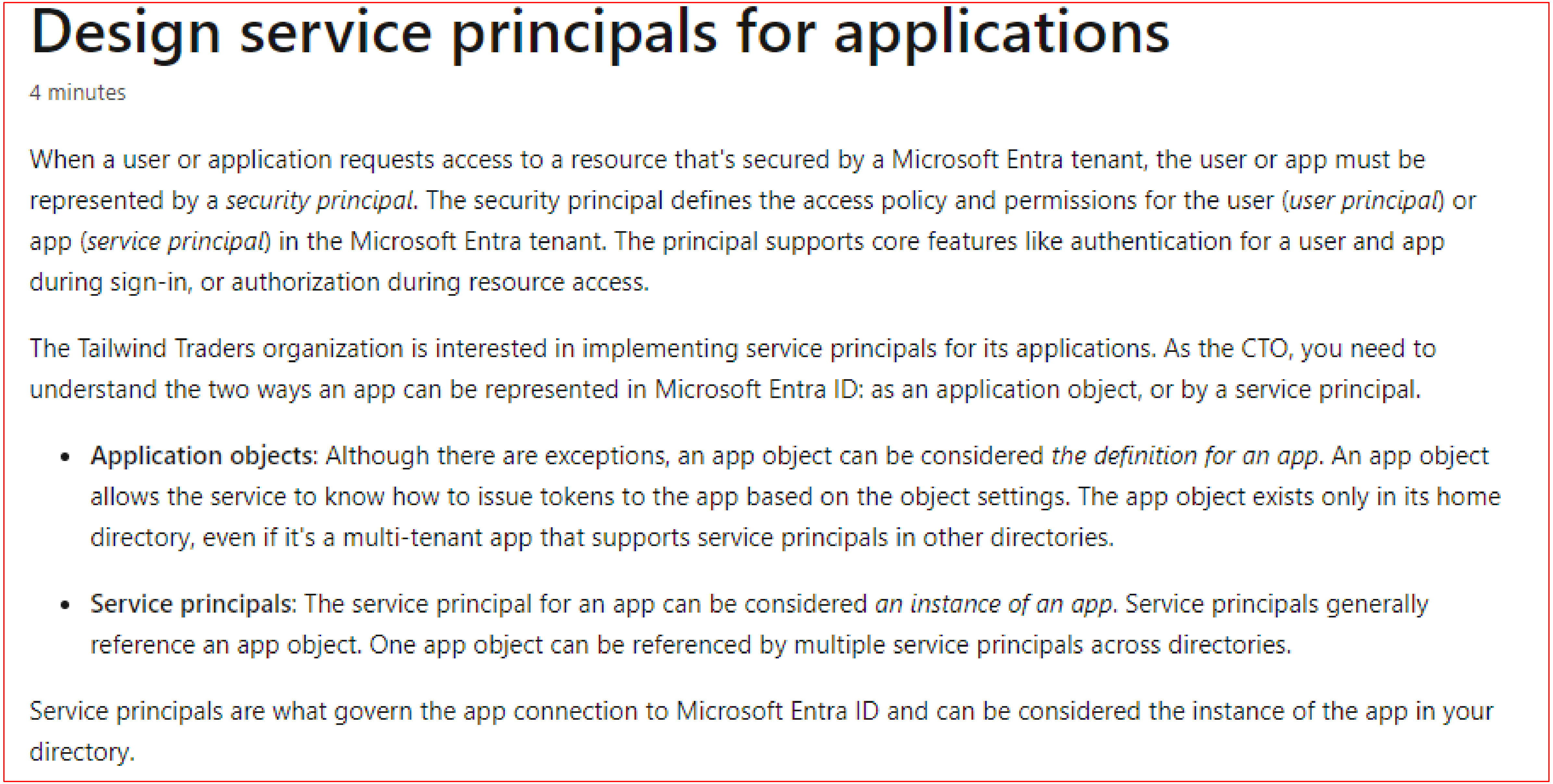
The local representation, or application instance, of an object in a single tenant or directory



Useful when Managed Identities cannot be used

Authentication is performed by the application using a secret or certificate

Often used to authenticate external applications to Azure resources



Design service principals for applications

4 minutes

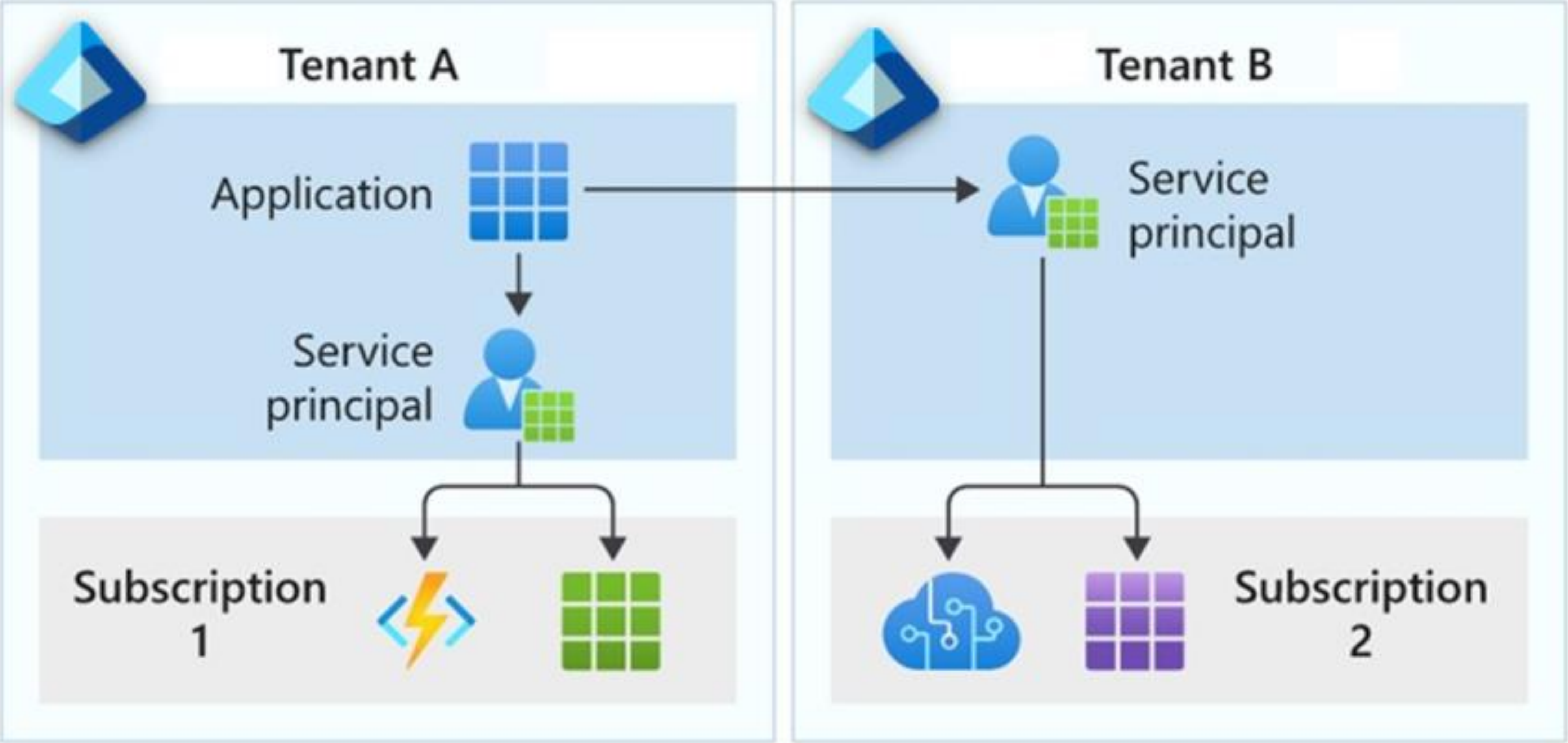
When a user or application requests access to a resource that's secured by a Microsoft Entra tenant, the user or app must be represented by a *security principal*. The security principal defines the access policy and permissions for the user (*user principal*) or app (*service principal*) in the Microsoft Entra tenant. The principal supports core features like authentication for a user and app during sign-in, or authorization during resource access.

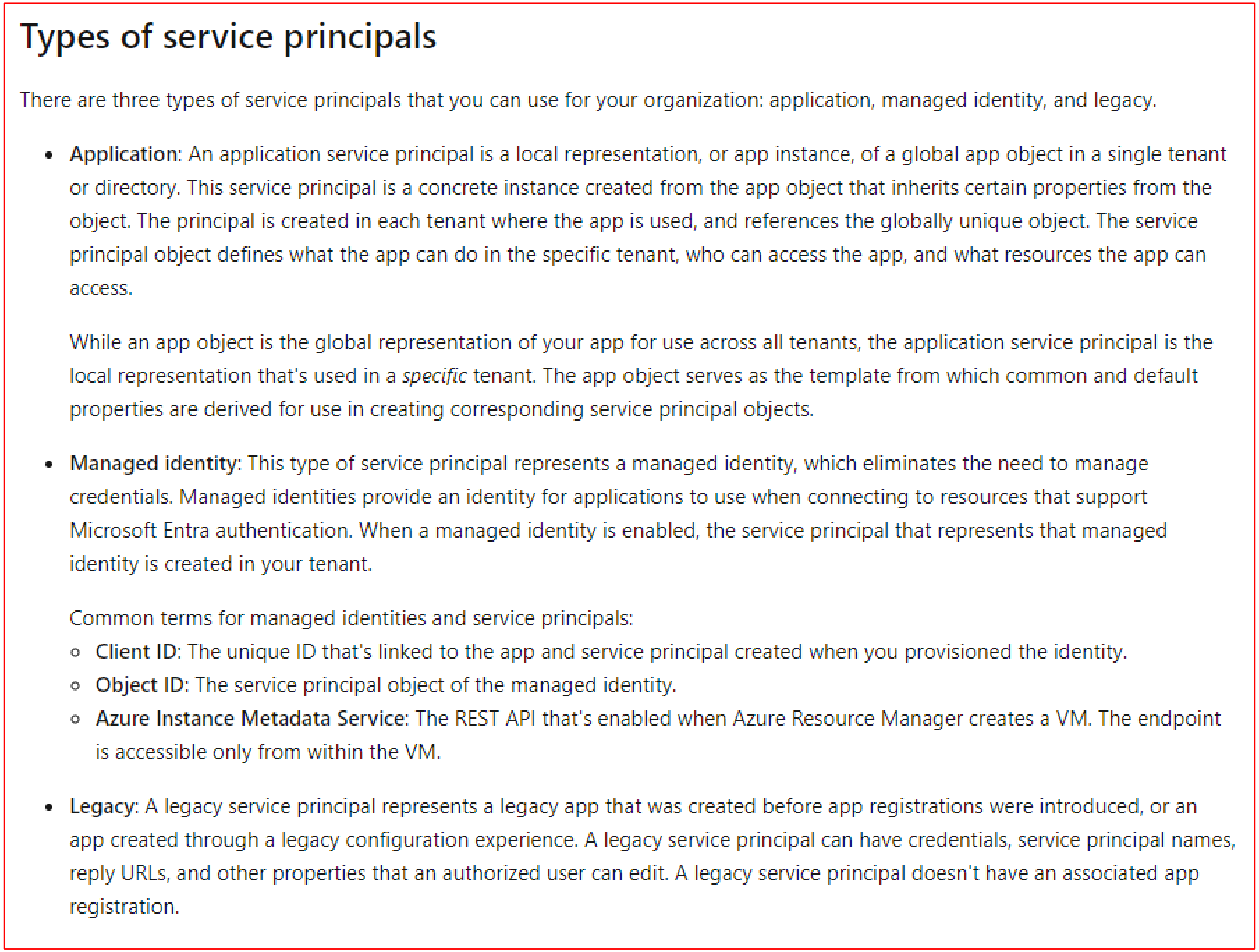
The Tailwind Traders organization is interested in implementing service principals for its applications. As the CTO, you need to understand the two ways an app can be represented in Microsoft Entra ID: as an application object, or by a service principal.

- **Application objects**: Although there are exceptions, an app object can be considered *the definition for an app*. An app object allows the service to know how to issue tokens to the app based on the object settings. The app object exists only in its home directory, even if it's a multi-tenant app that supports service principals in other directories.

- **Service principals**: The service principal for an app can be considered *an instance of an app*. Service principals generally reference an app object. One app object can be referenced by multiple service principals across directories.

Service principals are what govern the app connection to Microsoft Entra ID and can be considered the instance of the app in your directory.


Tenant A
Application
Service principal
Subscription 1
Tenant B
Service principal
Subscription 2



Types of service principals

There are three types of service principals that you can use for your organization: application, managed identity, and legacy.

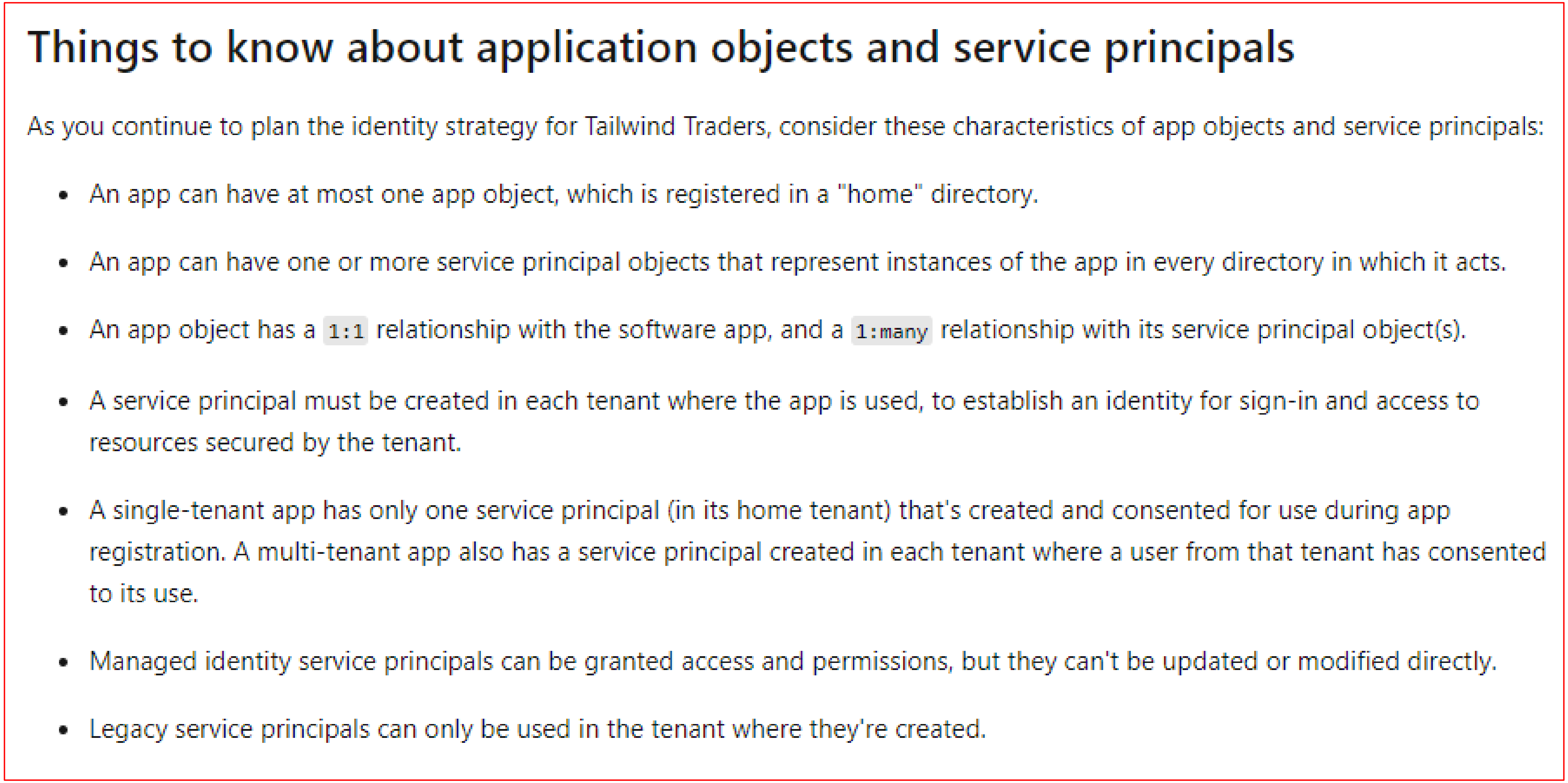
- **Application**: An application service principal is a local representation, or app instance, of a global app object in a single tenant or directory. This service principal is a concrete instance created from the app object that inherits certain properties from the object. The principal is created in each tenant where the app is used, and references the globally unique object. The service principal object defines what the app can do in the specific tenant, who can access the app, and what resources the app can access.

  While an app object is the global representation of your app for use across all tenants, the application service principal is the local representation that's used in a *specific* tenant. The app object serves as the template from which common and default properties are derived for use in creating corresponding service principal objects.

- **Managed identity**: This type of service principal represents a managed identity, which eliminates the need to manage credentials. Managed identities provide an identity for applications to use when connecting to resources that support Microsoft Entra authentication. When a managed identity is enabled, the service principal that represents that managed identity is created in your tenant.

  Common terms for managed identities and service principals:
  - **Client ID**: The unique ID that's linked to the app and service principal created when you provisioned the identity.
  - **Object ID**: The service principal object of the managed identity.
  - **Azure Instance Metadata Service**: The REST API that's enabled when Azure Resource Manager creates a VM. The endpoint is accessible only from within the VM.

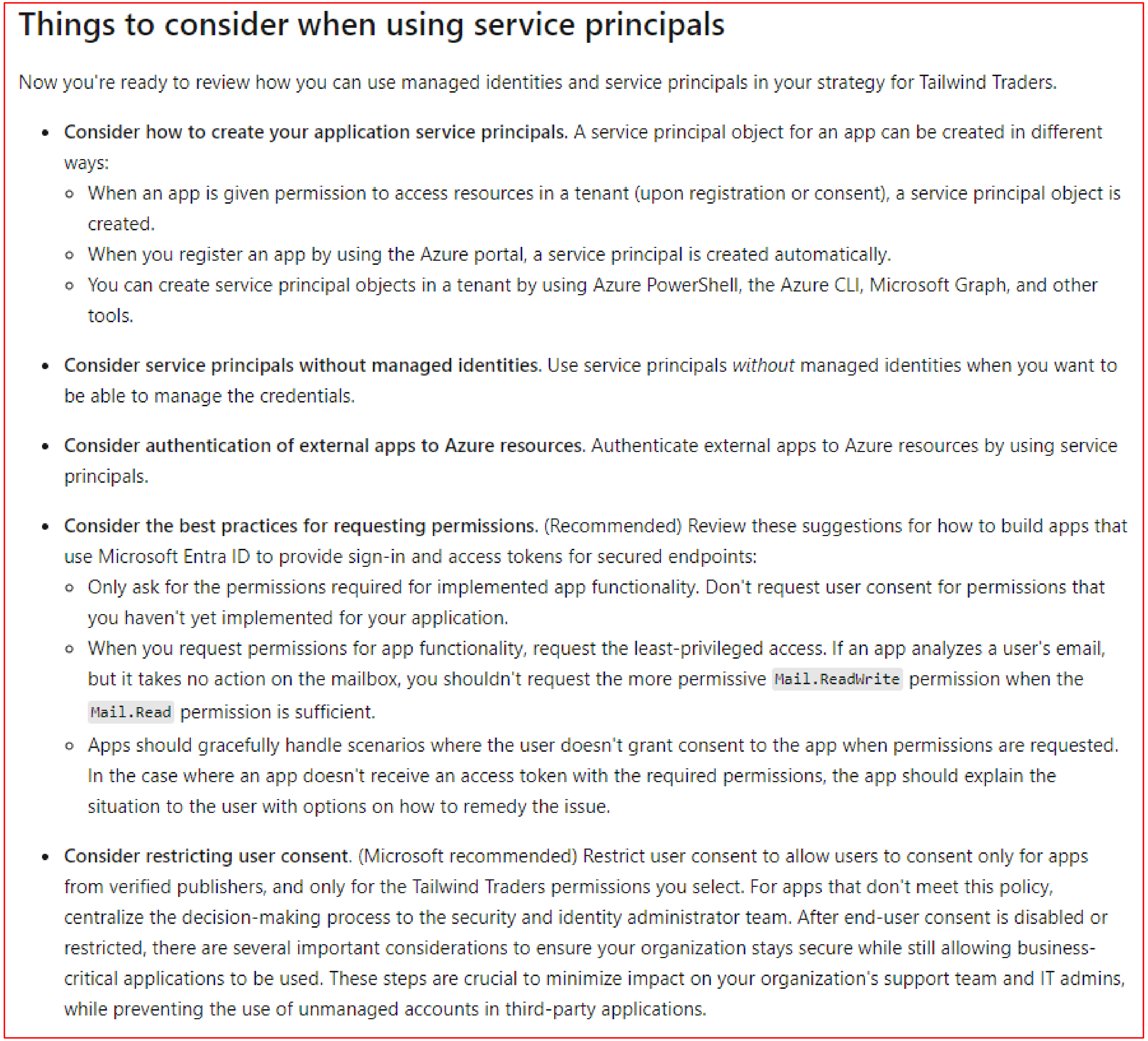
- **Legacy**: A legacy service principal represents a legacy app that was created before app registrations were introduced, or an app created through a legacy configuration experience. A legacy service principal can have credentials, service principal names, reply URLs, and other properties that an authorized user can edit. A legacy service principal doesn't have an associated app registration.



Things to know about application objects and service principals

As you continue to plan the identity strategy for Tailwind Traders, consider these characteristics of app objects and service principals:

- An app can have at most one app object, which is registered in a "home" directory.
- An app can have one or more service principal objects that represent instances of the app in every directory in which it acts.
- An app object has a `1:1` relationship with the software app, and a `1:many` relationship with its service principal object(s).
- A service principal must be created in each tenant where the app is used, to establish an identity for sign-in and access to resources secured by the tenant.
- A single-tenant app has only one service principal (in its home tenant) that's created and consented for use during app registration. A multi-tenant app also has a service principal created in each tenant where a user from that tenant has consented to its use.
- Managed identity service principals can be granted access and permissions, but they can't be updated or modified directly.
- Legacy service principals can only be used in the tenant where they're created.



Things to consider when using service principals

Now you're ready to review how you can use managed identities and service principals in your strategy for Tailwind Traders.

- **Consider how to create your application service principals.** A service principal object for an app can be created in different ways:
  - When an app is given permission to access resources in a tenant (upon registration or consent), a service principal object is created.
  - When you register an app by using the Azure portal, a service principal is created automatically.
  - You can create service principal objects in a tenant by using Azure PowerShell, the Azure CLI, Microsoft Graph, and other tools.

- **Consider service principals without managed identities.** Use service principals *without* managed identities when you want to be able to manage the credentials.

- **Consider authentication of external apps to Azure resources.** Authenticate external apps to Azure resources by using service principals.

- **Consider the best practices for requesting permissions.** (Recommended) Review these suggestions for how to build apps that use Microsoft Entra ID to provide sign-in and access tokens for secured endpoints:
  - Only ask for the permissions required for implemented app functionality. Don't request user consent for permissions that you haven't yet implemented for your application.
  - When you request permissions for app functionality, request the least-privileged access. If an app analyzes a user's email, but it takes no action on the mailbox, you shouldn't request the more permissive `Mail.ReadWrite` permission when the `Mail.Read` permission is sufficient.
  - Apps should gracefully handle scenarios where the user doesn't grant consent to the app when permissions are requested. In the case where an app doesn't receive an access token with the required permissions, the app should explain the situation to the user with options on how to remedy the issue.

- **Consider restricting user consent.** (Microsoft recommended) Restrict user consent to allow users to consent only for apps from verified publishers, and only for the Tailwind Traders permissions you select. For apps that don't meet this policy, centralize the decision-making process to the security and identity administrator team. After end-user consent is disabled or restricted, there are several important considerations to ensure your organization stays secure while still allowing business-critical applications to be used. These steps are crucial to minimize impact on your organization's support team and IT admins, while preventing the use of unmanaged accounts in third-party applications.

Best practices for requesting permissions

When building an app that uses Microsoft Entra ID to provide sign-in and access tokens for secured endpoints, there are a few good practices you should follow.

When registering an application in Microsoft Entra ID, consider business and security needs of admin consent versus user consent

Only ask for the permissions required for implemented app functionality. Don't request user consent for permissions that you haven't yet implemented for your application.

In addition, when requesting permissions for app functionality, you should request the least-privileged access.

Apps should gracefully handle scenarios where the user doesn't grant consent to the app when permissions are requested.

Design for Azure key vault

Design for Azure Key Vault

Azure Key Vault provides a secure storage area for managing all your app secrets so you can properly encrypt your data in transit or while it's being stored.

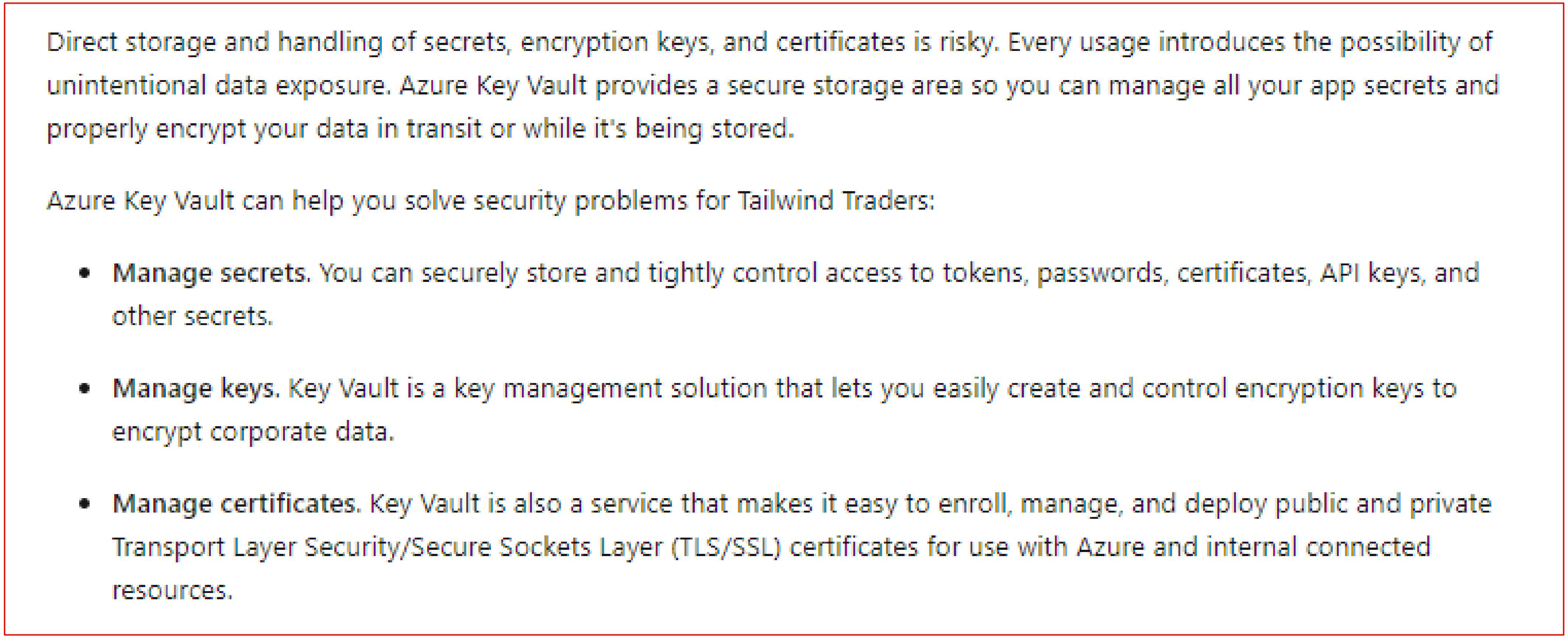
Why use Key Vault?

- Separation of sensitive app information from other configuration and code, reducing the risk of accidental leaks.
- Restricted secret access with access policies tailored to the apps and individuals that need them.
- Centralized secret storage, allowing required changes to happen in only one place.
- Access logging and monitoring to help you understand how and when secrets are accessed.
- Implementing Customer Managed Keys for Azure services

When to consider multiple Key Vaults:

- RBAC vs Policies
- Performance

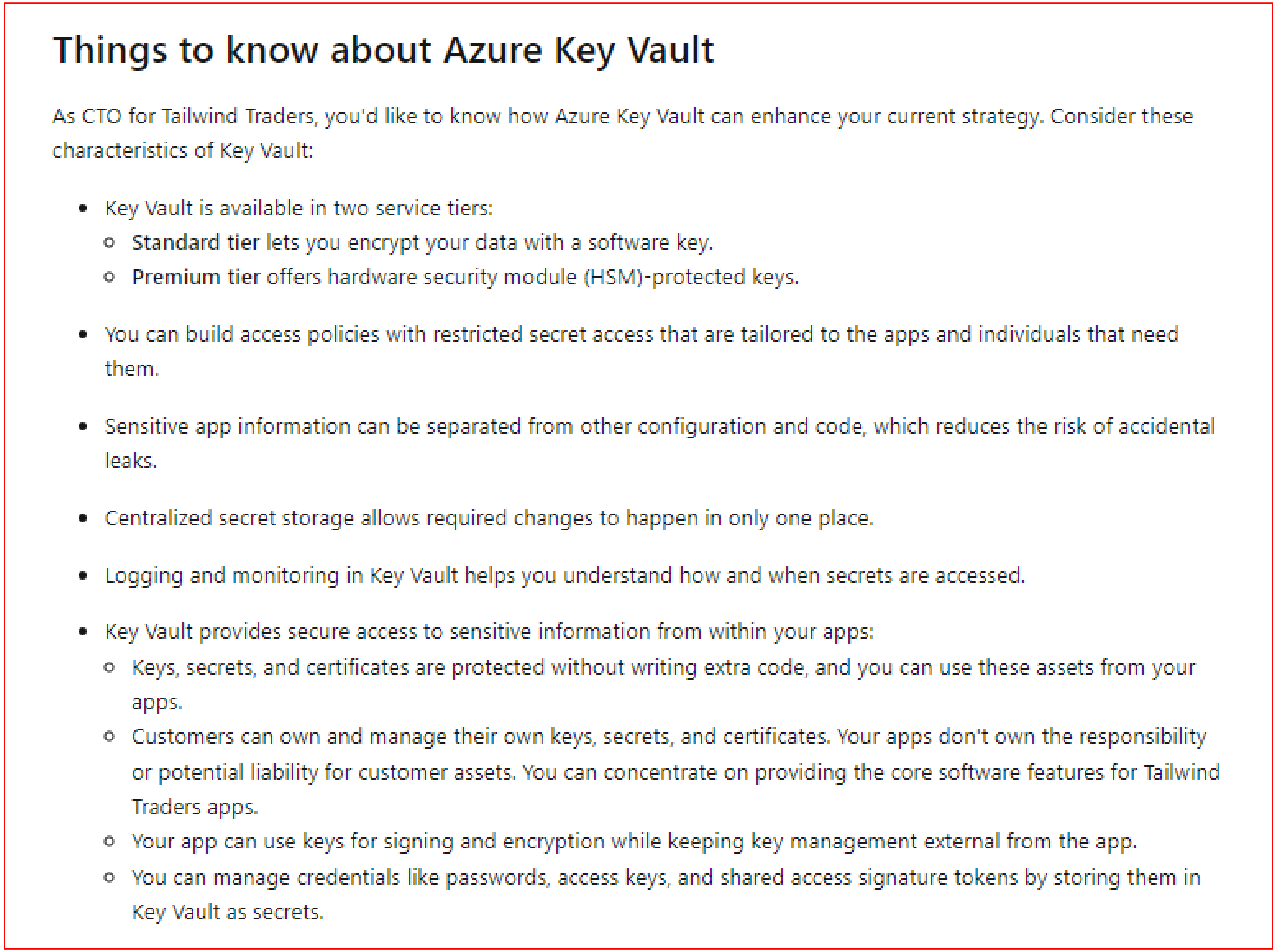
Design for Azure Key Vault



Direct storage and handling of secrets, encryption keys, and certificates is risky. Every usage introduces the possibility of unintentional data exposure. Azure Key Vault provides a secure storage area so you can manage all your app secrets and properly encrypt your data in transit or while it's being stored.

Azure Key Vault can help you solve security problems for Tailwind Traders:

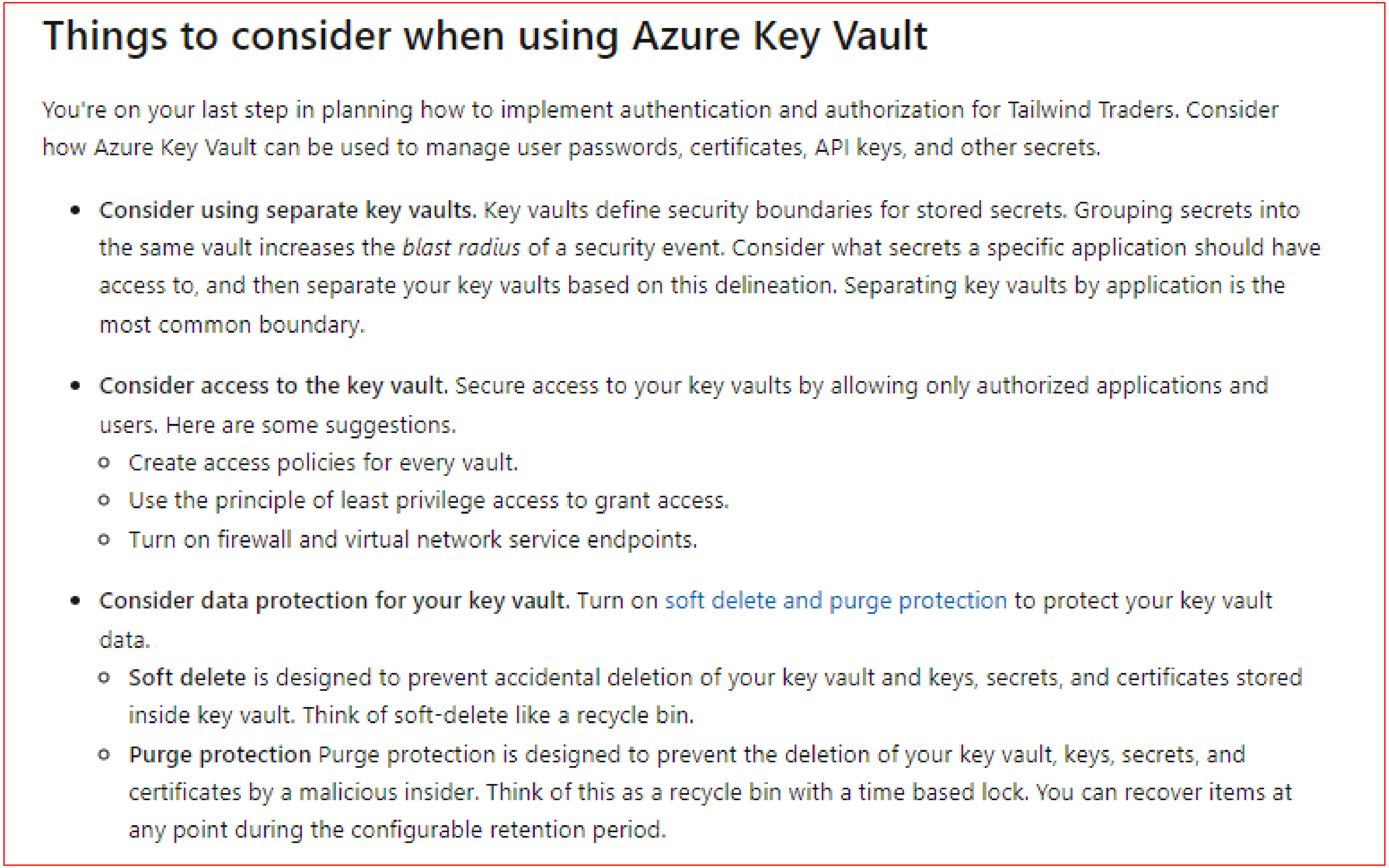
- **Manage secrets.** You can securely store and tightly control access to tokens, passwords, certificates, API keys, and other secrets.
- **Manage keys.** Key Vault is a key management solution that lets you easily create and control encryption keys to encrypt corporate data.
- **Manage certificates.** Key Vault is also a service that makes it easy to enroll, manage, and deploy public and private Transport Layer Security/Secure Sockets Layer (TLS/SSL) certificates for use with Azure and internal connected resources.



Things to know about Azure Key Vault

As CTO for Tailwind Traders, you'd like to know how Azure Key Vault can enhance your current strategy. Consider these characteristics of Key Vault:

- Key Vault is available in two service tiers:
  - **Standard tier** lets you encrypt your data with a software key.
  - **Premium tier** offers hardware security module (HSM)-protected keys.
- You can build access policies with restricted secret access that are tailored to the apps and individuals that need them.
- Sensitive app information can be separated from other configuration and code, which reduces the risk of accidental leaks.
- Centralized secret storage allows required changes to happen in only one place.
- Logging and monitoring in Key Vault helps you understand how and when secrets are accessed.
- Key Vault provides secure access to sensitive information from within your apps:
  - Keys, secrets, and certificates are protected without writing extra code, and you can use these assets from your apps.
  - Customers can own and manage their own keys, secrets, and certificates. Your apps don't own the responsibility or potential liability for customer assets. You can concentrate on providing the core software features for Tailwind Traders apps.
  - Your app can use keys for signing and encryption while keeping key management external from the app.
  - You can manage credentials like passwords, access keys, and shared access signature tokens by storing them in Key Vault as secrets.



Things to consider when using Azure Key Vault

You're on your last step in planning how to implement authentication and authorization for Tailwind Traders. Consider how Azure Key Vault can be used to manage user passwords, certificates, API keys, and other secrets.

- **Consider using separate key vaults.** Key vaults define security boundaries for stored secrets. Grouping secrets into the same vault increases the *blast radius* of a security event. Consider what secrets a specific application should have access to, and then separate your key vaults based on this delineation. Separating key vaults by application is the most common boundary.

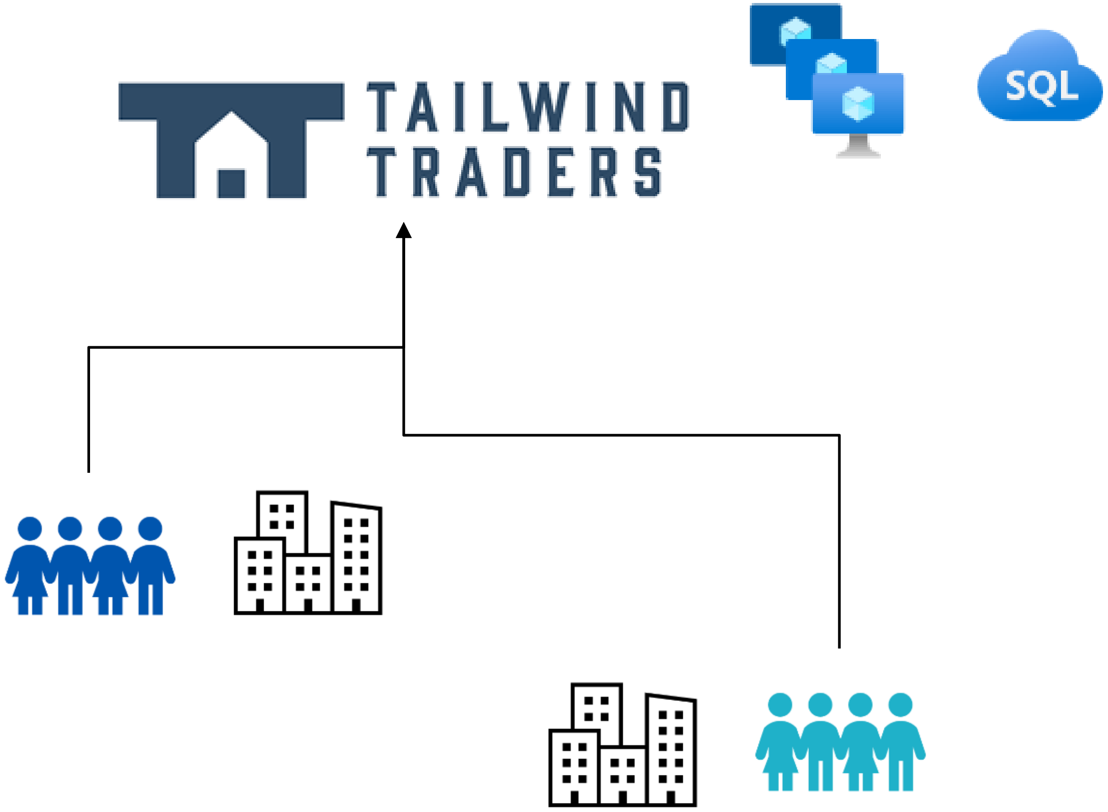
- **Consider access to the key vault.** Secure access to your key vaults by allowing only authorized applications and users. Here are some suggestions.
  - Create access policies for every vault.
  - Use the principle of least privilege access to grant access.
  - Turn on firewall and virtual network service endpoints.

- **Consider data protection for your key vault.** Turn on soft delete and purge protection to protect your key vault data.
  - **Soft delete** is designed to prevent accidental deletion of your key vault and keys, secrets, and certificates stored inside key vault. Think of soft-delete like a recycle bin.
  - **Purge protection** Purge protection is designed to prevent the deletion of your key vault, keys, secrets, and certificates by a malicious insider. Think of this as a recycle bin with a time based lock. You can recover items at any point during the configurable retention period.

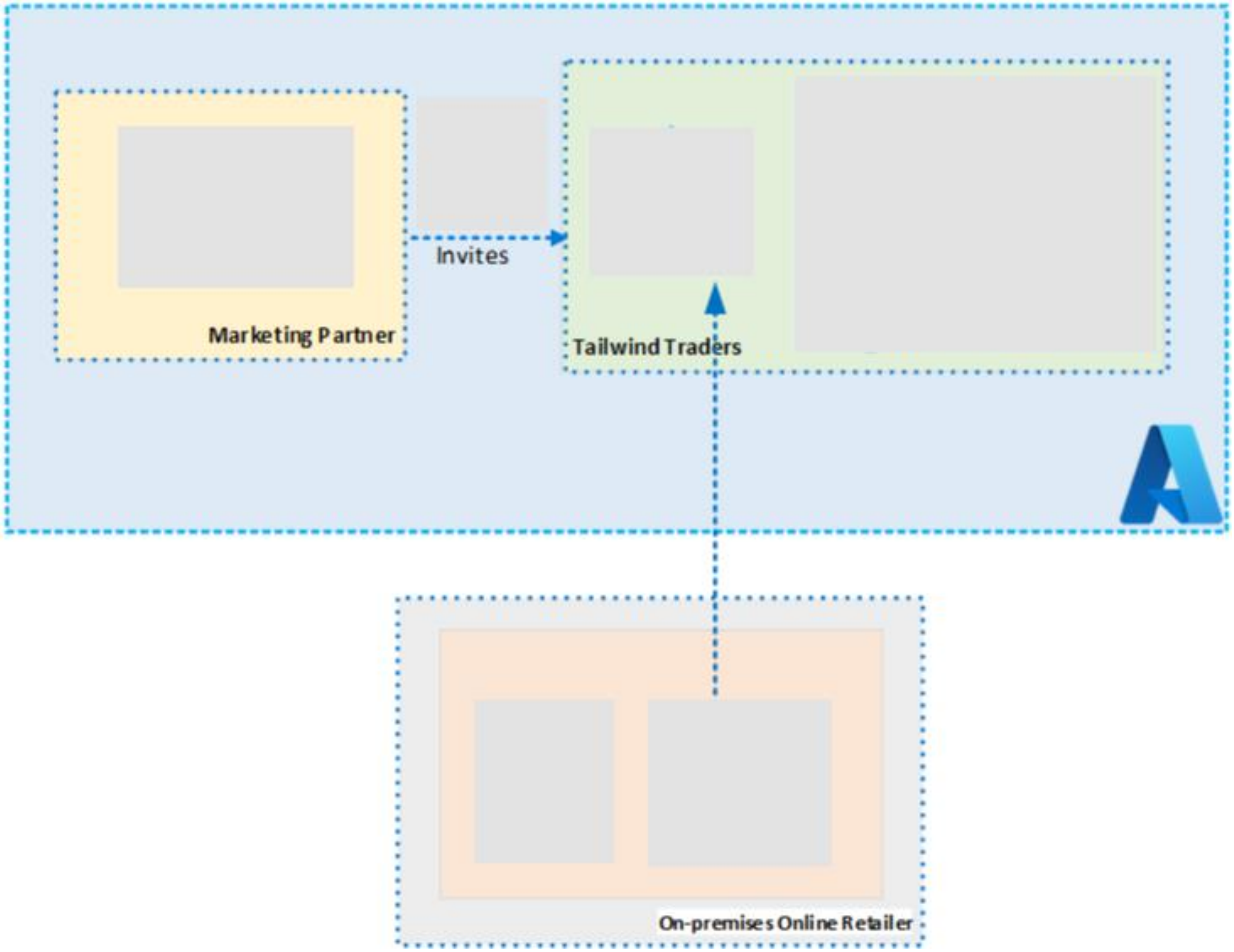
Case study and review

Case Study – Authentication and authorization

1. A company acquisition will add 75 employees – new user accounts
2. New employees are in different geographic regions – new identity protection policies
3. New application with a SQL database – access solution



New Employee Accounts



ADDS

Microsoft Entra Connect

Conditional Access

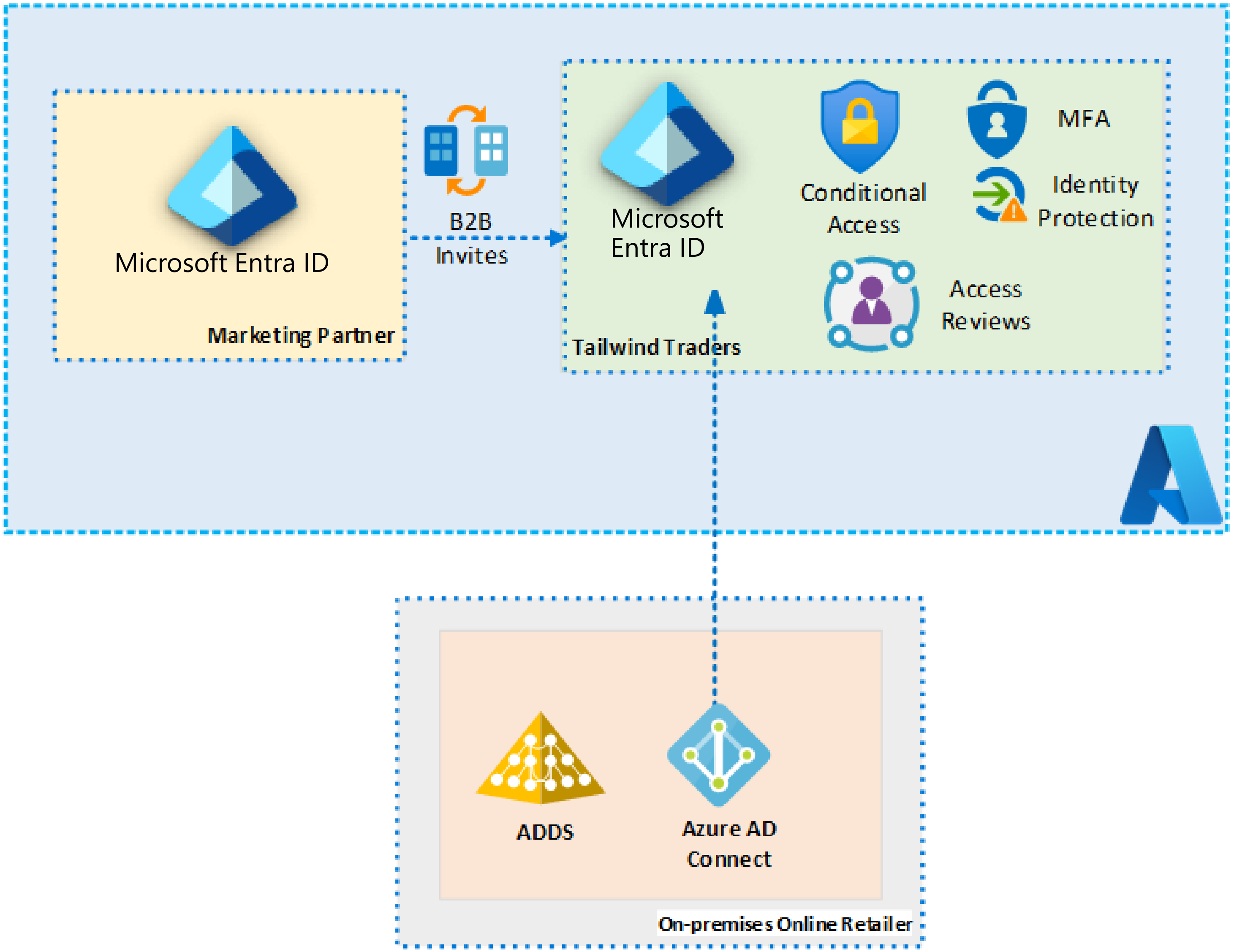
Identity Protection

B2B

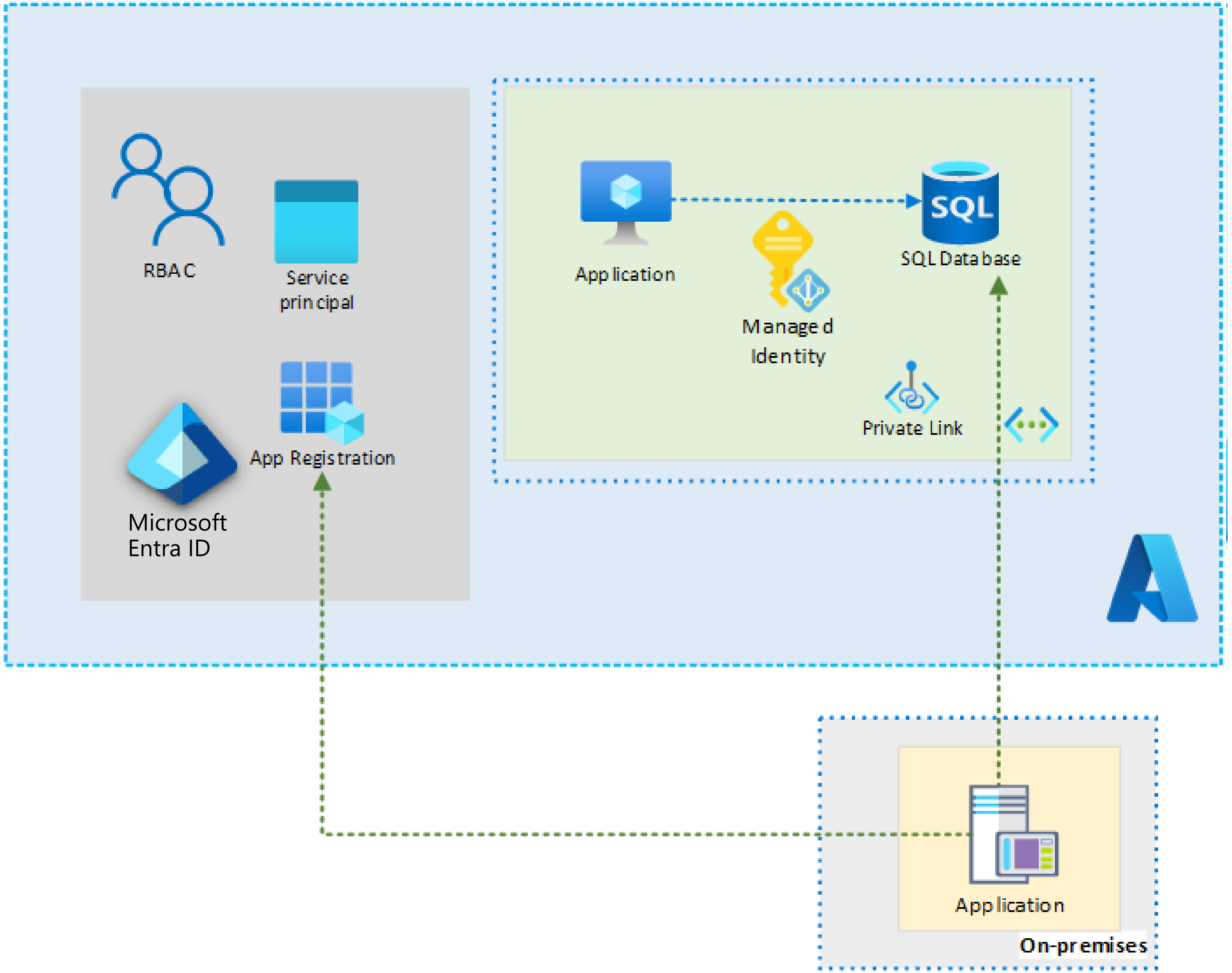
MFA

Microsoft Entra ID

New Employee Accounts (completed)



New Identity Solution Features


RBAC
Service principal
Microsoft Entra ID
App Registration
Application
Managed Identity
SQL Database
Private Link
Application
On-premises