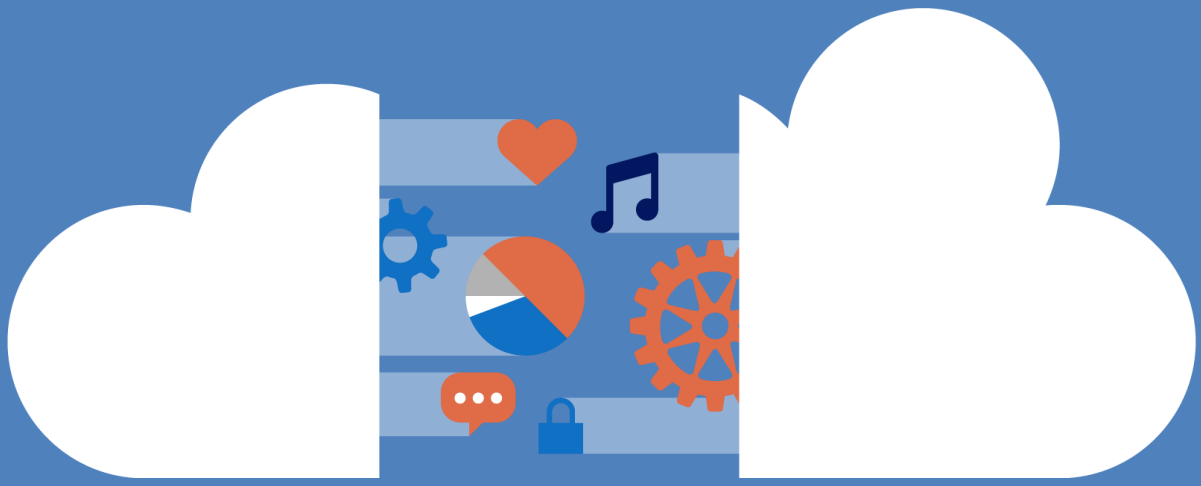
Azure Active Directory Data Security Considerations



Version: 2.02

Published: June 2020

For the latest information, please see

https://aka.ms/aaddatawhitepaper

Microsoft

Table of contents

Version history

Version	Changes	Date
1.0	Initial release	June 2018
1.01	Minor errors fixed	June 2018
1.02	Broken URLs fixed	January 2019
1.03	Minor errors fixed	March 2019
2.0	PIM and Managed Identity information added	May 2019
2.01	Removal of previous legacy authentication service per service evolution.	October 2019
2.02	User Provisioning, B2C, Password reset added, Authenticator app	June 2020

Acknowledgements

- Adrian Drumea – Partner Group Software Engineering Manager
- Alex Simons – Corporate Vice President, Identity Program Management
- Arturo Lucatero – Senior Program Manager
- Arun Nanda – Partner Architect
- Arvind Harinder – Program Manager
- Avi Carmon – Partner Group Engineering Manager
- Ian Farr – Senior Program Manager
- Igor Sakhnov – Partner Director of Engineering
- Kamen Moutafov – Partner Group Engineering Manager
- Keith Brintzenhofe – Partner Group PM Manager
- Marcus Carvalho – Senior Program Manager
- Martin Coetzer – Principal PM Manager
- Maxim Yaryn – Principal Software Engineer
- Modi Gendelman – Principal Software Engineering Manager
- Murli Satagopan – Technical Fellow
- Nadim Abdo – Partner Director of Engineering
- Pavle Anicic – Senior Software Engineer
- Qi Cao – Principal Engineering Manager
- Rahul Tewari – Principal Director Software Engineering
- Raman Chikkamagalur – Principal Group Engineering Manager
- Ramiro Calderon – Principal Program Manager
- Riley Kotchorek – Senior Software Engineer
- Sumit Parikh – Senior Program Manager
- Svyatoslav Trukhanov – Principal Software Engineer
- Tarek Dawoud – Principal PM Manager
- Vijayshree Chowdhary – Principal Software Engineering Manager

Introduction

This document explains the following aspects of Azure Active Directory:

- **Azure AD Components**: What are the different components of Azure AD. This will help you to understand the later sections of the document.
- **Core Data and Location**: What customer data is used by Azure AD and where is it located.
- **Data Protection**: How is the directory data protected at transit and at rest.
- **Data Flow**: How data from various sources such as on premises directories and applications flows to and from Azure AD.
- **Data and Operations**: What data and operational procedures are used by the Azure AD engineering team to manage the service.

The target audience of this document is enterprise security evaluators, identity, and access management (IAM) architects, policy makers and regulators, as well as customers with compliance requirements or regulated environments.

Azure Active Directory Components

The following major components will be outlined.

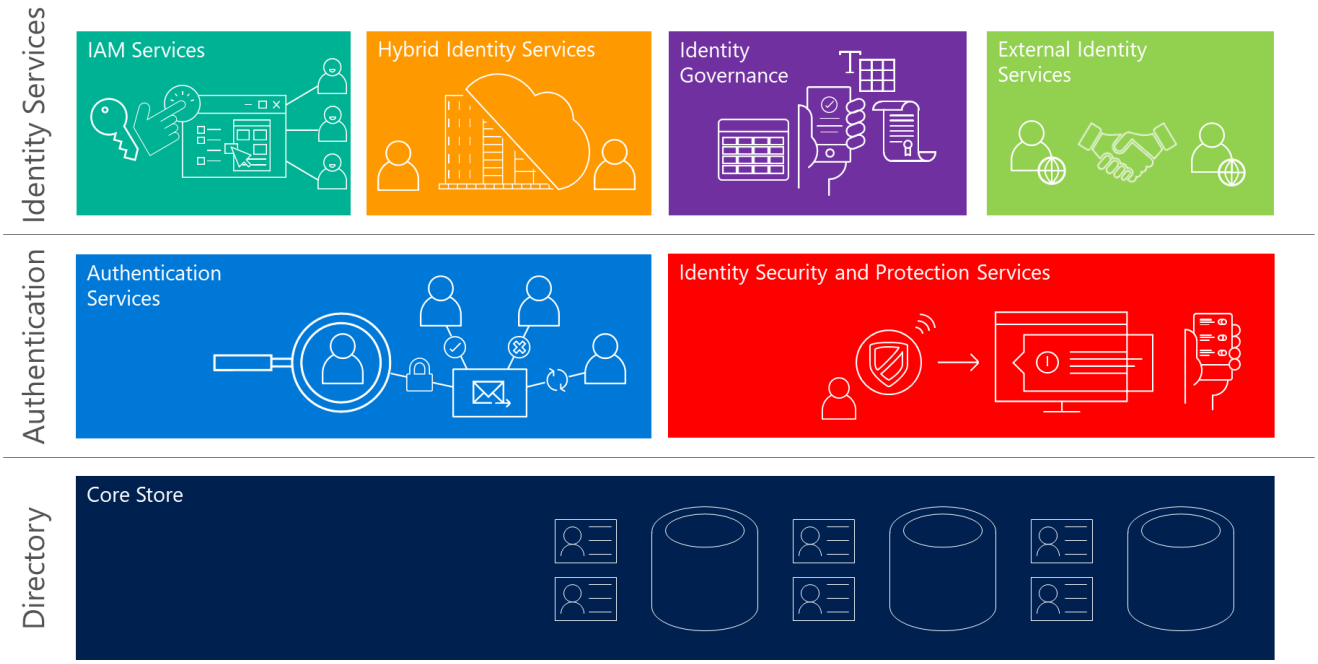


Figure 1 - Azure AD High-Level Components

As shown in the figure above, Azure AD is composed of the following high-level components:

- **Directory Data** is the data stored for your directory system. The directory data is created from the identity and access data provided by your organization to populate the service. This data includes the following entities and their attributes: Users, groups and group memberships, devices, applications, and roles. Directory Data also includes the metadata necessary to represent the relationships between these objects; some of which is provided by the customer and some of which is created by Azure AD services based on user actions such as registering applications, joining devices, etc.
- **Core Store** is the complete set of an organization's Directory Data is stored in a logical container (a "tenant") in a specific scale unit in the Azure AD distributed data store. The Azure AD storage is divided into scale units, and each unique scale unit holds multiple tenants. The Azure AD core store also provides the directory data access interfaces to other services.
- **Authentication Services**: Processes user input, validates credentials, and implements the authentication flows, endpoints, and security tokens required by the different industry standards supported by the system. The industry standards define the format and exchange patterns for issuing, renewing, canceling, and validating security tokens provided by the authentication services as a security token service (STS).
- **Identity Security and Protection Services**: Provides identity-driven protection to users when interacting with the system such as Azure Multifactor Authentication (MFA), Azure AD Identity Protection, and Conditional Access.
- **Identity and Access Management (IAM) Services**: Provides advanced identity management features such as self-service password reset, self-service group management, dynamic group

membership, automated app assignment, provisioning for third-party services, management interfaces, and reporting capabilities.

- **Azure AD Services**: Provides customers the infrastructure necessary to integrate existing on-premises infrastructure to Azure AD.
  - **Azure AD Connect** provides synchronization of on-premises directory users to the cloud.
  - **Azure AD Connect Health** provides monitoring and analytics for synchronization, federation, and domain services.
  - **Azure AD Application Proxy** enables secure publishing of on-premises web applications for remote access.
  - **Azure AD Domain Services** Provides managed domain services, such as domain join, group policy, LDAP, Kerberos, and NTLM authentication. These services are fully compatible with Windows Server Active Directory.
- **Azure AD Identity Governance**: Provides customers governance capabilities such as Azure AD Privileged Identity Management (PIM) Just In time (JIT) access to privileged roles, access certification, attestation campaigns, alerting, and reporting.
- **Azure AD External Identities**: Provides authentication services for external identities, such as users in partner organizations or consumers.

Azure AD and Data

Core Store

Azure AD is an Identity as a Service (IDaaS) solution that enables customers to store and manage Identity and Access data in the cloud and use it to enable and manage access to cloud services, achieve mobility scenarios, and secure their organization. An instance of the Azure AD service for a customer, called a tenant, is an isolated set of Directory Object Data provisioned and owned by the customer.

Data operations, such as updates or retrievals, in the Azure AD core store, are scoped to a single tenant based on the user's security token, to achieve tenant isolation.

As indicated above, the core store is made up of individual tenants. Tenants are stored in scale units, each of which contains multiple tenants. The Azure AD service replicates each scale unit to all the physical data centers of a logical region for resiliency and performance.

Azure AD currently has the following logical regions: North America, Europe Middle East, and Africa (EMEA), Australia, China, Germany, US Government, and Worldwide. Each of these handle directory data based on usability, performance, residency and/or other unique requirements. Residency means that Microsoft provides assurance that the data will not be persisted outside of the geographic region.

The Azure AD service replicates each tenant through its scale unit among multiple data centers based on the following criteria:

- Reduce latency to assure fast login times for users by storing Directory Data in data centers closest to the locations where they reside.
- Assure availability if there are unforeseen geological events by storing Directory Data in data centers that are geographically isolated from each other.
- Compliance with data residency or other unique requirements for specific customers and countries/regions.

When a tenant is created (for example, signing up to Office 365 or Azure, or when creating additional instances of Azure AD through the Microsoft Azure portal) the user selects a country or region as the primary location. Azure AD automatically maps the selection to a logical region and a single scale unit within it. Tenant administrators cannot change a tenant's location after it is created.

Azure AD Cloud solution models

Azure AD offers various cloud solutions models, based on infrastructure, data location, and operation sovereignty, as described in the table below.

Model	Logical regions in this model	Data Location	Operations Personnel	Customer Support	How to get a tenant on this model?
Regional (3)	North America EMEA (1) Australia	At rest in the target geographical region. Exceptions by service or feature	Operated by Microsoft. Globally Microsoft's datacenter personnel must pass a background check.	Microsoft, Globally	Creating the tenant through the general signup experience and choosing the country within the desired residency.
Worldwide (3)	Worldwide		Operated by Microsoft. Globally Microsoft's datacenter personnel must pass a background check.	Microsoft, Globally	Creating the tenant through the general signup experience and choosing a country that does not offer a Regional model.
National Clouds	Germany (1) US Government China	At rest in the target country/ region. No exceptions	Operated by Data Custodian (2) Screening requirements and practices tailored to specific requirements.	Microsoft, Country/Region	Each national cloud instance has its own sign up experience.

NOTE:
(1) **EMEA region and Germany region differences**: Signing up with the regular experience (portal.azure.com or office.com) and choosing Germany as the country region will result in an Azure AD tenant created in the EMEA Azure AD logical region. To create a tenant in the Germany national Cloud, please refer to Microsoft Cloud Deutschland

(2) **Data Custodians**: Data centers in Microsoft's Worldwide region are operated by Microsoft. In China, Azure AD is operated through a partnership with 21Vianet. In Germany, Azure is available through a data trustee model with Deutsche Telekom.
Learn more: Microsoft Trust Center - Microsoft National Clouds

(3) **Authentication Data**: Tenants outside of the National Clouds have authentication information at rest in the continental US.

Learn more:

- Azure Active Directory – Where is your data located?
- Understand Azure Active Directory architecture
- Which Azure region is right for me?
- Microsoft Trust Center - Microsoft National Clouds

Data Location across Azure AD Components

Aside from authentication service data (described in more detail in the table in the "Data storage across Azure AD Components" section below), Azure AD components, and service data is located or stored inside the region of the Azure AD edition.

Learn more: Azure Active Directory Features and Capabilities

NOTE:
To understand the data location of additional services (such as Exchange Online, or Skype for Business), please refer to the corresponding service's documentation.

Data Storage across Azure AD Components

Below, is a summary of the data storage per Azure AD Components.

Azure AD Authentication Services: The current Azure AD Authentication Service is stateless. The data needed to perform authentication resides in the Azure AD core store. It has no Directory Data on its own.

The Authentication Service generates log data that is stored in Azure storage in the data center where the service instance is running. When users attempt to authenticate using Azure AD, they are automatically routed to an instance at the geographically nearest data center, which is part of its Azure AD logical region.

Azure AD IAM Services:

- **User Experience and Management Experience**: The Azure AD management experience is entirely stateless and has no Directory Data of its own. It generates log and usage data, which is stored in Azure table storage.
- **Identity Management Business Logic and Reporting Services**: These services have locally cached data storage for groups and users.

These services also generate log and usage data that is stored in Azure table storage, Azure SQL and in Microsoft's Elastic Search reporting services.

Azure MFA: Refer to Data residency and customer data for Azure Multi-Factor Authentication for details on the data storage and retention of MFA operations.

The Azure MFA service logs the username (UPN) and telephone number for voice calls and SMS challenges. For challenges sent to mobile app modes, the service logs the username and a unique device token. The Azure MFA service and the logs it creates are currently stored in data centers in the North America region.

Azure MFA Server: The on-premises MFA Server uses the backend infrastructure of the Azure MFA described above to send phone calls, SMS messages, and push notifications. Therefore, the following data will leave their current location because it is required to perform MFA; the data is also necessary to provide the MFA Server Activity Report:

- **Unique ID** (either username or on-premises Multi-Factor Authentication Server ID)
- **First and Last Name** (optional)
- **Email Address** (optional)
- **Phone Number** (when using a voice call or SMS authentication)
- **Device Token** (when using mobile app authentication)
- **Authentication Mode**
- **Authentication Result**
- **Multi-Factor Authentication Server Name**
- **Multi-Factor Authentication Server IP**
- **Client IP** (if available)

Azure AD Domain Services: Regions where Azure AD Domain Services is published at: Products available by region. The service holds system metadata globally in azure tables, and it does not contain any personal data.

Azure AD Connect Health: Azure AD Connect Health generates alerts and reports in Azure Table Storage and blob storage.

Azure AD Dynamic Membership for Groups/ Azure AD Self-Service Group Management: The definition of dynamic membership rules is stored in Azure Table storage.

Azure AD Application Proxy: Azure AD Application Proxy stores metadata about the tenant, connector machines, and configuration data in SQL Azure.

Azure AD Password Reset: Azure AD Password Reset backend service uses Redis Cache to track session state.

Azure AD Password Writeback: During the first configuration Azure AD Connect generates an asymmetric keypair, using the RSA cryptosystem, and sends the public key to the SSPR cloud service. The SSPR cloud service then performs two operations:

- It creates two Azure Service Bus relays for the Azure AD Connect on-premises service to securely communicate with the SSPR service
- It generates an AES key (K1)

The Service Bus relay locations, the corresponding listener keys and a copy of the AES key (K1) is sent back to Azure AD Connect in the response. All future communication between SSPR and Azure AD Connect will now take place over this newly created Service Bus channel and is encrypted using SSL.

During operation, when new password resets are submitted, the passwords are encrypted with the RSA public key that was generated by the client during the onboarding. Only the private key on the Azure AD Connect machine can decrypt them. This prevents any other subsystems in the pipeline having access to the plaintext password.

The whole message payload (encrypted passwords + additional data + metadata) is then encrypted with the AES key (K1) that both Azure AD Connect and SSPR have. This prevents malicious ServiceBus operators or attackers from tampering with the payload, even with full access to the internal ServiceBus channel.

The keys / data needed by Azure AD Connect for password writeback include:

- The AES key (K1) that is used to encrypt the payload of the reset / change requests that go from the SSPR Service to Azure AD Connect via the ServiceBus pipeline.
- The private key from the asymmetric key pair used to decrypt the encrypted passwords sent in reset / change request payload.
- The ServiceBus listener keys.

The AES key and the asymmetric keypair are rotated at a minimum of every 180 days and can also be changed under certain onboarding / offboarding configuration events. An example of this would be when a customer disables and then re-enables password writeback. This may also occur during component upgrade during servicing and maintenance.

All the keys and data needed for writeback are stored the in Azure AD Connect database, and are encrypted via DPAPI (CALG_AES_256). The resultant key is known as the master ADSync encryption key and is stored in the Windows Credential Vault in the context of the AdSync on-premises service account. The Windows Credential Vault supplies automatic re-encryption of secrets as the password of the service account changes. Resetting the service account password invalidates any secret kept in the Windows Credential Vault for the given service account. Changing to a new service account manually will also completely invalidate all stored secrets.

By default, the ADSync service runs in the context of a Virtual Service Account. The account may be customized at install time to any least privileged domain service account or a Managed Service Account (MSA / gMSA). While both virtual and managed service accounts provide automatic password rotation, the customer must manage password rotation for a custom provisioned domain account. As noted above, resetting the password will result in immediate loss of all stored secrets.

Azure AD Device Registration Service: Provides lifecycle management of computers and devices in the directory, which enable scenarios such as device-state conditional access, and mobile device management.

Azure AD provisioning: The Azure AD provisioning service handles creating (provisioning)/updating and removing users in other systems, such as SaaS applications. It also manages the creation of users

in Azure AD and on-premises AD from cloud HR sources, like Workday. The service stores its configuration in an Azure Cosmos DB. The Cosmos DB stores the group membership data for the user directory it keeps. Cosmos DB replicates the database to multiple datacenters inside the same region as the tenant, to isolate them according to the Azure AD cloud solution model. Replication provides high availability and multiple reading and writing endpoints.

Cosmos DB provides encryption on the database information, and the encryption keys are stored in the secrets storage for Microsoft (outlined later in this document).

Azure AD B2B Collaboration: Azure AD B2B Collaboration does not have Directory Data of its own. It is important to note that users and other directory objects participating in a B2B relationship with another tenant will result in copying user data in other tenants, which may have data residency implications.

Azure AD Identity Protection: Azure AD Identity Protection uses real-time user login data along with multiple signals from company and industry sources to feed to its machine learning systems to detect anomalous logins. Personal data is scrubbed from this real-time login data before it is passed into the machine learning system, along with the remaining login data used to identify users and logins that are potentially risky. After the analysis is complete and the data is sent to Microsoft's reporting systems, the risky logins and usernames are flagged so administrators can generate reports.

Azure AD PIM: The Azure AD PIM service consists of two main components: a database stored in SQL Azure and the background service responsible for orchestrating events and tasks necessary to satisfy role assignment requests.

Production Azure AD PIM uses two separate Azure SQL databases to store tenant settings. The first database is used exclusively for European tenants. The first database is in and replicated between the European datacenters (North Europe and West Europe) for redundancy and availability purposes. Azure AD stores all other tenant settings in a SQL Azure database in US datacenters (Central US and East US 2). National clouds have their own database instances for PIM settings. PIM stores the role definitions for each tenant in the database. A role definition, for example, for Global Administrators, include role settings like the duration limits, multi-factor authentication requirements, and the role assignments eligible member and active member. The SQL Azure database also keeps current and past request information. Request information includes role ID, user ID, type of request and start/end time. Furthermore, Azure AD PIM generates and stores alerts against violation of best practice rules, such as too many permanent administrators in the database. Azure AD PIM logs requests and assignments in the Azure AD audit log.

Azure AD Managed Identities for Azure resources: With managed identities systems can authenticate to Azure services, without storing credentials in their system. Rather than using username and password, managed identities authenticates to Azure services with certificates.

The service writes certificates it issues in Azure Cosmos DB in the East US region, which can be failed over to another region. Azure Cosmos DB provides geo-redundancy by globally replicating the data. Replicating the database provides a read-only copy in each Azure region Azure AD managed identities runs. Microsoft isolates each Cosmos DB instance in a specific Azure AD Cloud solution model.

The resource provider, such as the virtual machine host, will also store the certificate to use for authentication and identity flows with other Azure services.

The service stores its master key for accessing the Azure Cosmos DB in a datacenter secrets management service and the master encryption keys are stored in Azure Key Vault.

Azure Active Directory B2C: Is an identity management service that enables you to customize and control how customers sign up, sign in, and manage their profiles when using your applications.

B2C use the same core store as Azure AD, to store identity information for user's identity information. The core store database follows the same storage, replication, deletion and data residency rules as outlined in other places in this whitepaper.

The B2C also uses an Azure Cosmos DB system to store policies and secrets related to the service. Cosmos DB provides encryption and replication services on the database information, and its encryption key is stored in a secrets storage for Microsoft (outlined later in this document). Microsoft isolates each Cosmos DB instance in a specific Azure AD Cloud solution model.

Data Protection Considerations

Access Control

As shown in the diagram below, services store and retrieve Directory Object Data through a role-based access control authorization layer that calls the internal directory data access layer. This ensures that the data requested is allowed for the user requesting it:

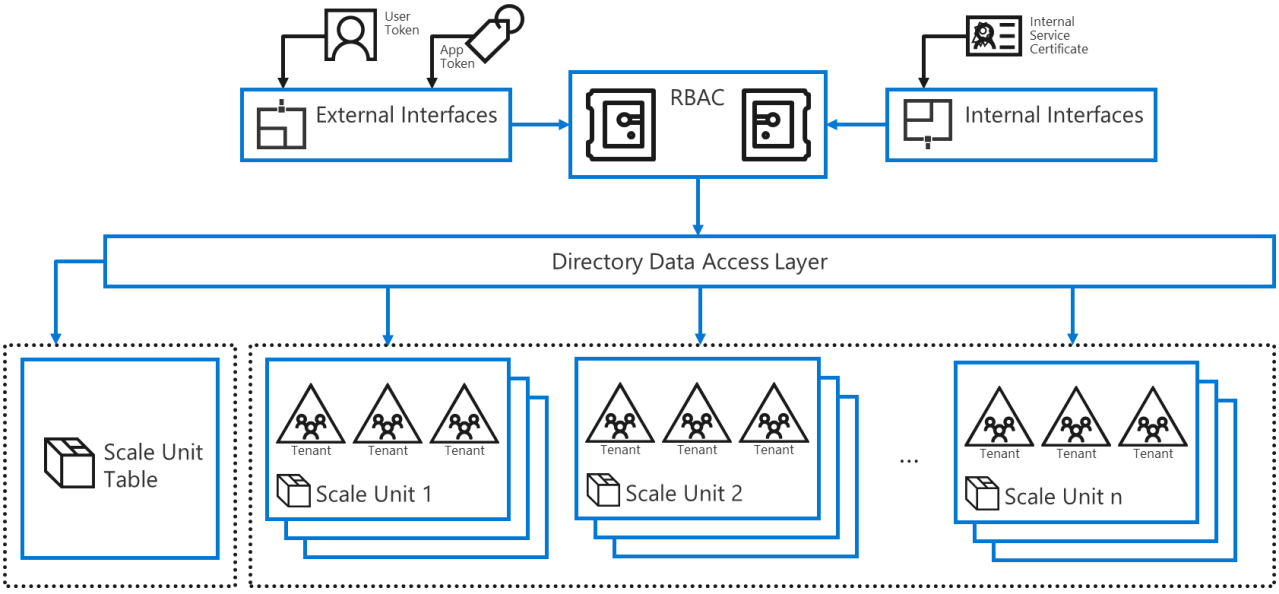


Figure 2 – Tenant Isolation

Azure AD Internal Interfaces Access: Azure AD internal interfaces are used for service-to-service communication with other Microsoft services, such as Office 365. Those interfaces authorize the service's callers using client certificates.

Azure AD External Interfaces Access: Azure AD external interface prevents data leakage by employing role-based access controls. When a security principal, such as a user, makes an access request to read information through Azure AD's interfaces, a security token must accompany the request that contains claims about the principal making the request.

The security tokens are issued by the Azure AD Authentication Services. Information on the user's existence, enabled state, and role is used by the authorization system to determine whether the requested access to the target tenant is authorized for this user in this session.

Application Access: Since applications can access the Application Programming Interfaces (APIs) directly (without any user context), the access check also includes information about the user's application as well as the scope of access requested (for example read only, read/write, etc.). For example, many applications use OpenID Connect or OAuth to obtain tokens to access the directory on behalf of the user. These applications must be explicitly granted access to the directory or they will not receive a token from the Security Token Service (STS), and they can only access data from within the specific scope they were granted.

Auditing: Access is audited. For example, authorized actions such as create user and password reset create an audit trail that can be used by a tenant administrator to manage compliance efforts or investigations. Tenant administrators can generate audit reports at any time using the Azure AD audit API.

Learn more: Azure Active Directory audit report events

Tenant Isolation: Enforcement of security in Azure AD's multi-tenant environment involves two primary elements:

1. Preventing data leakage and access across tenants. This means that data belonging to Tenant 1 cannot in any way be obtained by users in Tenant 2 without explicit authorization by Tenant 1.
2. Resource access isolation across tenants. This means that operations performed by Tenant 1 cannot in any way impact access to resources for Tenant 2.

The information below outlines tenant isolation:

- Each tenant is secured using Role-Based Access Control (RBAC) policy to ensure data isolation.
- To enable access to a specific tenant, a principal (for example a user or application) needs to be able to authenticate against Azure AD to obtain context and has explicit permissions defined in the tenant. If a principal is not authorized in the tenant, the resulting token will NOT carry any permissions and, as a result the RBAC system will reject any requests in this context.
- RBAC ensures that access to a tenant is performed by a security principal that is explicitly defined and authorized within the tenant. Access across tenants is only possible where a tenant administrator explicitly creates a security principal representation in the same tenant. Examples of these representations are configuring a federation or provisioning a guest user account using B2B collaboration. This is because each tenant is an isolation boundary; existence in one tenant does not equate to existence in another tenant unless the administrator allows it.
- Azure AD data for multiple tenants is stored in the same physical server and drive for a given partition. Isolation is ensured because access to the data is protected by the RBAC authorization system.
- A customer application cannot access Azure AD without proper authentication. The request is rejected if not accompanied by the proper credentials as part of the initial connection negotiation process. This is core to preventing unauthorized access to a tenant by neighboring tenants. Only user credential's token (or Security Assertion Markup Language (SAML) token) is brokered as part of a federated trust, and therefore validated by Azure AD based on the shared keys configured by the Global Administrator of your Azure AD tenant.
- Since there is no application component that can execute directly from within the Core Store itself, it is not possible for one tenant to forcibly breach the integrity of a neighboring tenant.

Data Security

Encryption in Transit: To assure data security, Directory Data in Azure AD is signed and encrypted while in transit between data centers within a scale unit. The data is encrypted and unencrypted by the Azure AD core store tier, which resides inside secured server hosting areas of the associated Microsoft data centers.

Customer-facing web services are secured with the Transport Layer Security (TLS) protocol.

Secret Storage: Azure AD service back-end utilizes secret stores for storing of sensitive material for service use such as certificates, keys, credentials, and hashes using technology that is proprietary to Microsoft. The specific store used depends on the service, the operation, the scope of the secret (user-wide or tenant-wide), and other requirements.

These stores are operated by a security-focused group for the Azure fabric/service (the Azure Security Team) via established automation and workflows, including certificates' request, renewal, revocation, and destruction.

There is auditing of activities related to these stores/workflows/processes and there is no 'standing access'. Access is request- and approval-based, and only for a limited amount of time.

For more information about "Secret encryption at rest" see the table below.

Algorithms: Below is a table that documents-specific cryptography algorithms used by various Azure AD components. As a cloud service, Microsoft is constantly reassessing and improving the cryptography based on security research findings, internal security reviews, key strength against hardware evolution, etc.

Data / Scenario	Crypto used	Comments
Password hash sync Cloud account passwords	Hash: Password Key Derivation Function 2 (PBKDF2), using HMAC-SHA256 @ 1000 iterations	For password hash sync, the on-premises account password hash is salted and rehashed. Cloud account passwords are salted and hashed. The resulting one-way hash derived from this operation is encrypted at rest (see the "Secret encryption at rest" row of this table for details). It is important to note that only this derivative is stored in the cloud service.
Directory data in transit between data centers	Signed and encrypted using a random session key using AES-256-CTS-HMAC-SHA1-96	Directory Data in Azure AD is signed and encrypted while in transit between data centers within a scale unit. The data is encrypted and unencrypted by the Azure AD Core Store that resides inside secured server hosting areas of the associated Microsoft data centers.

Data / Scenario	Crypto used	Comments
Pass-through authentication user credential flow	RSA 2048 Public/Private key pair	When a new pass-through authentication agent is installed, a new client certificate for the agent is generated. When a user provides credentials in the Azure AD user interface, the authentication service encrypts them with the public keys of the certificates for agents registeredin the tenant. Typically, the certificate is rolled over every three to six months. It can be rolled over on demand, based on operational and security requirements. Learn more: Understand Azure AD Application Proxy connectors
Seamless single sign-on service account password	When Azure AD Connect is configured to use seamless single sign-on, a service account credential is needed for Kerberos authentication. The credentials are stored at rest per the details of the "Secret encryption at rest" row in this table. Learn more: Azure AD Connect: Single Sign-on	

Data / Scenario	Crypto used	Comments
Self-service password reset password writeback – Cloud to on-premises communication	RSA 2048 Private/Public key pair AES_GCM (256-bits key, 96-bits IV size)	When Azure AD Connect Is configured, a new private/public key is generated. The cloud backend only knows the public key and the Azure AD Connect keeps the private key. In addition to this, a AES_GCM symmetric key is exchanged for use at runtime. The key is 32 bytes (256-bit) key, 12 bytes (96-bit) nonce, 16 bytes (128-bit) tag. The requests from the cloud service include the new password (encrypted with the public key described above), as well as metadata. Then, the request information is encrypted with AES_GCM as described above and then sent on-premises via Azure Service Bus.
Self-service password reset password writeback – Security question answers	Hash: SHA256	The answers of the security questions are hashed. The resulting one-way hash is encrypted at rest (see the "Secret encryption at rest" row of this table for details).

Data / Scenario	Crypto used	Comments
SaaS application outbound provisioning credentials	The credentials used for Azure AD for outbound provisioning to third-party SaaS applications are stored at rest per the details of the "Secret encryption at rest" row in this table. In addition, a certificate is exchanged between the provisioning service and the secret store API. When the provisioning service requests to decrypt credential, the Secret Store decrypts the data from the storage and then re-encrypts the credential with the certificate's public key, as described above. The provisioning service then decrypts the credential provided by the secret store API and uses it to authenticate to access SaaS applications provisioning APIs.	
SSL certificates for Azure AD Application Proxy published applications	AES 256-bit symmetric key	The SSL certificate is stored and encrypted at stored in SQL Azure, using a symmetric key that is typically rotated every 70 days. The symmetric key is in turn stored using the secret store per the "Secret encryption at rest" row in this table.
Disk Level Encryption	AES 128-bit	The data volumes of the servers that run the Azure AD Core store are encrypted at rest using Microsoft BitLocker drive encryption technology. A unique key is generated for each server. Learn more: BitLocker Drive Encryption. How we secure your data in Azure AD – Enterprise Mobility and Security Blog

Data / Scenario	Crypto used	Comments
Secret encryption at rest	AES 128-bit symmetric key – CBC	In addition to disk level encryption, when at rest, secrets stored in the directory are encrypted using the Distributed Key Manager(DKM). The encryption keys are stored in Azure AD core store and in turn are encrypted with a scale unit key. The key is stored in a container that is protected with directory ACLs, for highest privileged users and specific services. The symmetric key is typically rotated every six months. Access to the environment is further protected with operational controls and physical security. See sections Operator Security and Physical Security for more details. The higher-level services that consume the secret store call the encrypt/decrypt operations through a dedicated interface, locked down to these specific services. The key material used for these operations does not leave the Azure AD Core Store. The encryption operations must also define a scope that can be the tenant, user, or service principal.
Azure AD Managed Identities	Data encryption keys (DEK): AES Master key encryption keys (KEK): A256KW	Data encryption keys are 256-bits in size There is a KEK per Azure region. Each region's KEK is shared with other regions to support failover and cross-region user assigned identity flows.

Data / Scenario	Crypto used	Comments
Microsoft Authenticator app	Passwordless key (2048 bits). Does not expire. Key ID (256 bits) using AES-256 Hashing use SHA-512	The device generates a cryptographic RSA key pair that is used to support Passwordless protocols. The public key is stored in as a user property in the Azure AD data store. For back up, the device requests a Key ID from Microsoft to encrypt secure data in a JSON Web encryption blob. The blob is hashed and added to the encryption to protect against tampering. The resulting data is uploaded to Microsoft's cloud storage provider on Android device and to iCloud on iOS devices.

Deleting data from Azure AD

When an administrator or the Azure AD service deletes a user object, it is first moved into the recycle bin where the data remains intact. This gives a customer the ability to easily recover the user object if objects are accidentally deleted. After 30 days, the user object becomes a deleted object (also known as tombstone) where the attribute data is removed from Azure AD, except for the subset of unique identifier data that is required for replicating deletions among Microsoft data centers. At this point, no personal data remains. After another 30 days, the user object is removed from the Azure AD scale unit.

Azure AD also provides an option for customers to accelerate deletion of user object by turning user objects in the recycle bin to tombstone objects. Currently, the recycle bin functionality is only supported for Directory user objects. When you delete other Directory objects, they become tombstone immediately, and they are removed from the Azure AD scale unit after 30 days.

The same logic applies to scenarios where customers are using Azure AD Connect to populate their Azure AD tenants using on-premises AD objects. If the on-premises AD user objects are deleted, the corresponding user objects in Azure AD are moved into the recycle bin. For other on-premises AD objects such as groups and contacts, they become tombstone immediately.

Data Flow Considerations

This section provides details on the data exchanged between different systems and Azure AD per the diagram below:

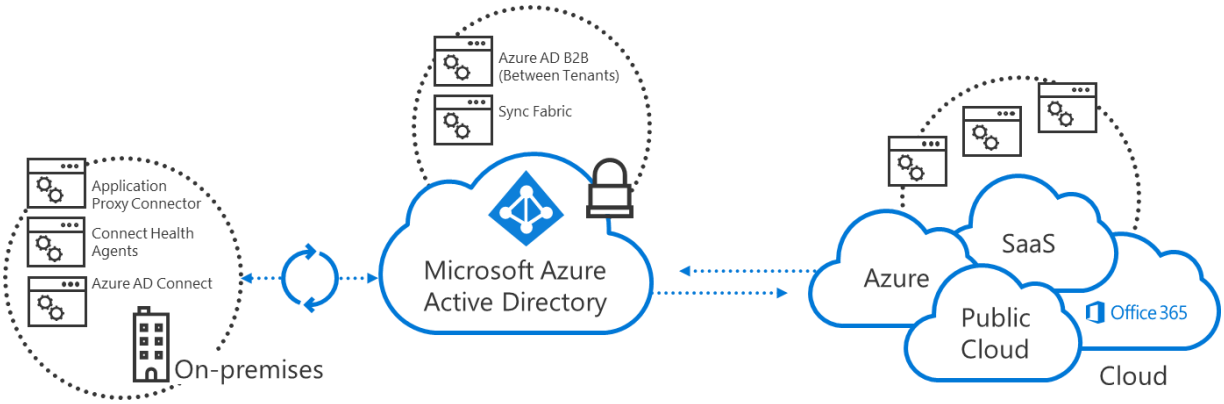


Figure 3 – Azure AD Data Flows

Azure AD Connect

Azure AD Connect allows synchronization between on-premises directories to Azure AD:

Directory Data: Whenever a directory object is created in Azure AD, it is automatically assigned a globally unique identifier (GUID). This GUID is used to reference the object throughout the Azure AD system, but it intentionally contains no Directory Data, and no Directory Data is used to generate the GUID.

Learn more: Attributes synchronized by Azure AD Connect

Attribute filtering: Customers can customize the synchronization rules to filter out the attributes that are not approved for synchronization based on their constraints.

Learn more: Azure AD Connect sync: How to make a change to the default configuration

NOTE:
Some applications such as Exchange Online require specific attributes to be synchronized.
Learn more: Azure AD Connect sync: Attributes synchronized to Azure Active Directory .

Service accounts: Azure AD Connect Servers require on-premises accounts. They create cloud accounts and grant the required permissions to securely perform the synchronization tasks.

Learn more: Azure AD Connect: Accounts and permissions

Exporting on-premises changes to the cloud: Azure AD Connect server obtains an Azure AD token using the auto-generated service account as described on the "service accounts" section. Then, Azure AD Connect establishes a client TLS connection to the synchronization web service in the cloud and perform SOAP calls to export the changes, providing the Azure AD token to authorize the service account.

Password writeback network channel: The network channel used for password writeback operations (for example password reset) is initiated from the Azure AD Connect computer on-premises to the cloud service using Azure Service Bus; this technology uses bi-directional sockets to enable the operations at runtime.

Learn more: Service Bus messaging overview

Pass-through authentication network channel: The network channel used for pass-through authentication is initiated from the Azure AD Connect server on-premises to the cloud service using Azure Service Bus. This service bus uses bi-directional sockets to enable the operations at runtime.

Learn more: Service Bus messaging overview

Azure AD Connect Health Agents

Azure AD Connect Health agents upload information to the cloud service, encrypting in transit using a client TLS connection. When the agents are configured, a certificate is generated on-premises to authenticate the agent to the cloud endpoints.

Agent Installation Locations:

- The Agent for Sync installed on the Azure AD Connect server.
- The Agent for AD FS installed on the AD FS servers and Web application proxies.
- The Agent for AD DS installed on the Active Directory domain controllers.

Agent Data Collection: The Agents collect the following information

- Error event logs
- Activity or audit logs
- Server performance counters
- Results of explicit checks performed by Connect Health agents
- Server metadata and topology data

Learn more: Monitor your on-premises identity infrastructure in the cloud

Azure AD Application Proxy Connectors

Connector Installation Location: The App Proxy connectors are installed on the on-premises corporate network for secure remote access of internal applications.

Learn more:

- Understand Azure AD Application Proxy connectors
- Security considerations when accessing apps remotely using Azure AD Application Proxy

Azure AD provisioning services

Azure AD provisioning services interoperate with third-party SaaS applications to provision users, groups, and roles calling the APIs supported by the SaaS applications. The service captures credentials for programmatic access per the mechanism described in the Cryptography section in this document.

Learn more: Automated SaaS App User Provisioning in Azure AD

Office 365 considerations

When assigning Office 365 service plans to users, relevant users, groups, devices, and application data are replicated from Azure AD into the Office 365 services.

NOTE:
The encrypted and hashed passwords are not copied into the Office 365 cache and are stored in Azure AD. For more information about see "Data Encryption" for more details.

For example, when a customer adds a subscription for Office 365 that includes Exchange Online, it is recorded in the Microsoft commerce systems, which trigger Azure AD to begin synchronizing data to the applicable Office 365 services based on the customer's country code and the Office 365 license plan they have selected.

Data Operational Considerations

Log Files

Azure AD generates log files for auditing, investigation, and debugging purposes for a wide variety of actions and events in the service. Log files contain data about usernames, groups, devices, and apps. Log files are originally created and stored in Azure storage in the data center where the Azure AD service runs.

Log files are used for local debugging, usage analysis, and system health monitoring purposes, as well as for service-wide analysis. Prior to any system-wide analysis, log files are first scrubbed of personal data, which is tokenized. These logs are then copied over a secure SSL connection to Microsoft's reporting machine learning systems, which are contained in Microsoft owned data centers in the Continental United States.

NOTE:
Personal data being any information relating to a person that can be used to directly or indirectly identify who they are, including the physical, physiological, genetic, mental, economic, cultural, or social identity of said individual.

Usage Data

Usage data is metadata generated by the Azure AD service that indicates how the service is being used. This metadata is used to generate administrator and user facing reports and is also used by the Azure AD engineering team to evaluate system usage and identify opportunities to improve the service. This data is generally written to log files, but in some cases, is collected directly by our service monitoring and reporting systems. personal data is stripped out of Microsoft's usage data prior to the data leaving the originating environment.

Operator Security

Access to Azure AD by Microsoft personnel, contractors, and vendors (system admins) is highly restricted. Wherever possible, human intervention is replaced by an automated, tool-based process, including routine functions such as deployment, debugging, diagnostic collection, and restarting services.

Admin access is limited to a subset of qualified engineers. Admin access requires the completion of a multi-factor authentication (MFA) challenge. System access and update functions are assigned to roles that are managed by Microsoft's just-in-time (JIT) privileged access management system. System administrators request elevation using the JIT system. The JIT system then routes the request for manual or automated approval. Upon approval, the JIT system elevates the operator’s account.

Requests for elevation, approvals, elevations into roles, and removals from roles are logged in case they are needed for future debugging or investigations.

Microsoft personnel can only execute operations from a secure access workstation, which use an internal isolated strong authentication identity platform. Access to other Microsoft identity systems cannot grant access to the security access workstation because the identity platform runs separately from all other Microsoft identity systems.

Physical Security

Physical access to servers that comprise the Azure AD service and direct access to Azure AD's back-end systems is restricted. Azure AD customers have no access to physical assets or locations, and therefore it is not possible for them to bypass the logical RBAC policy checks stated in the **Access Control** section. Personnel with operator access are authorized to run only approved workflows for maintenance purposes.

Change Control Process

The Azure AD team defines deployment rings to gradually roll out changes to the service across data centers. There are strict exit criteria before applying a change to the next deployment ring. The amount of time to roll a change across rings is defined by the operations team, based on its potential impact, typically taking between 1 to 2 weeks. Critical changes such as security fixes or hot fixes are deployed faster.

If the change does not meet the exit criteria when applied to a given deployment ring, it is rolled back to the prior stable state.

Additional Resources

- Microsoft Service Trust Documents
- Microsoft Azure Trust Center - Azure
- Where is my data? – Office 365 documentation
- Compliance Program now publicly available for financial services customers - Office Blogs