CERTIFIED HYBRID MULTI-CLOUD RED TEAM SPECIALIST

CHMRTS

Agenda [Day -1]

1. Introduction to Hybrid Multi Cloud Red Teaming ?
2. Introduction to AWS Cloud Red Teaming ?
3. Introduction to Azure Cloud Red Teaming ?
4. Q/A ?

Agenda [Day -2]

1. Introduction to Azure Cloud Red Teaming ?
2. Introduction to Google Red Teaming ?
3. Q/A ?

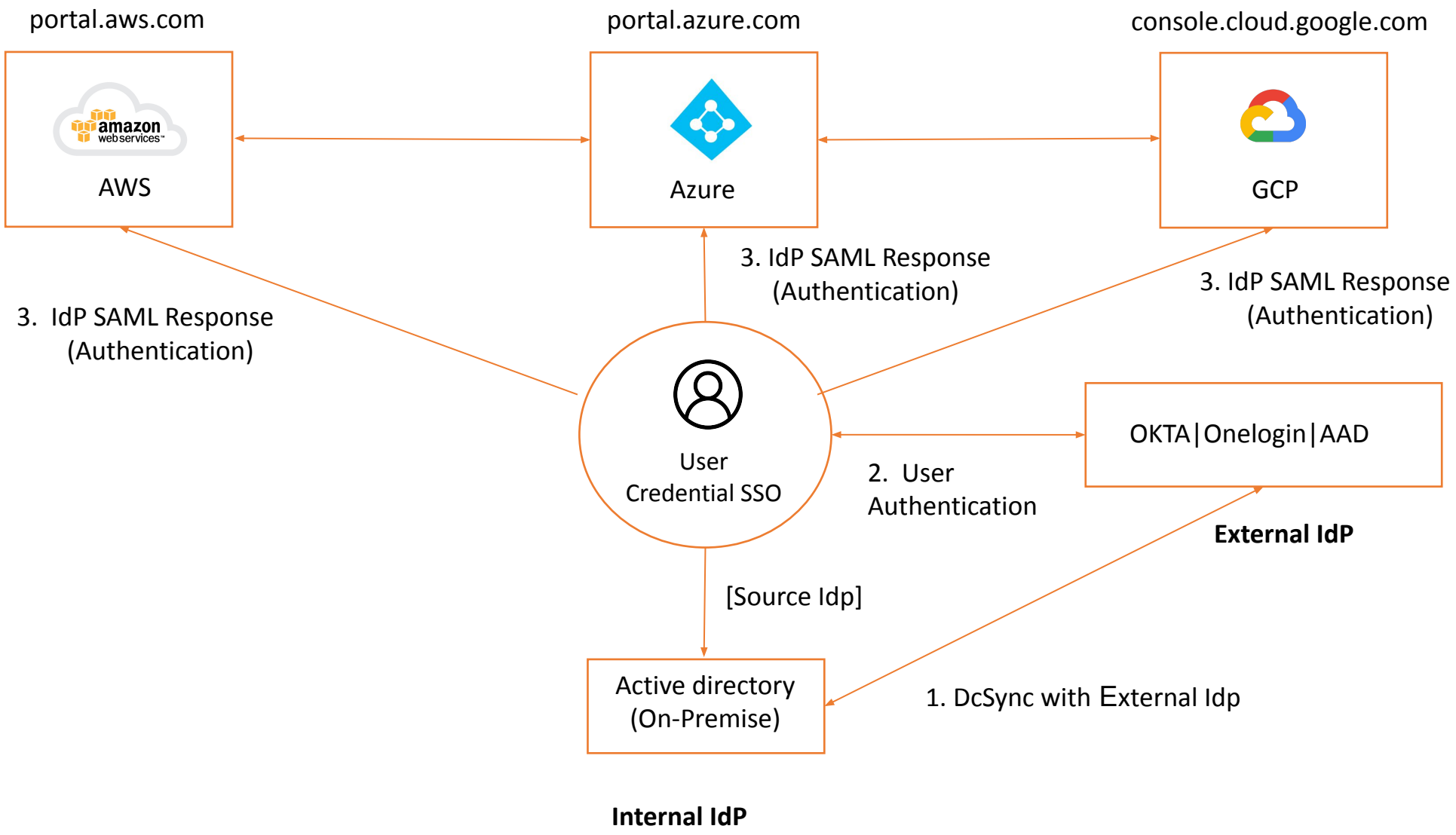
1. Introduction to Hybrid Multi Cloud Red Teaming

Hybrid Multi Cloud Environment Overview

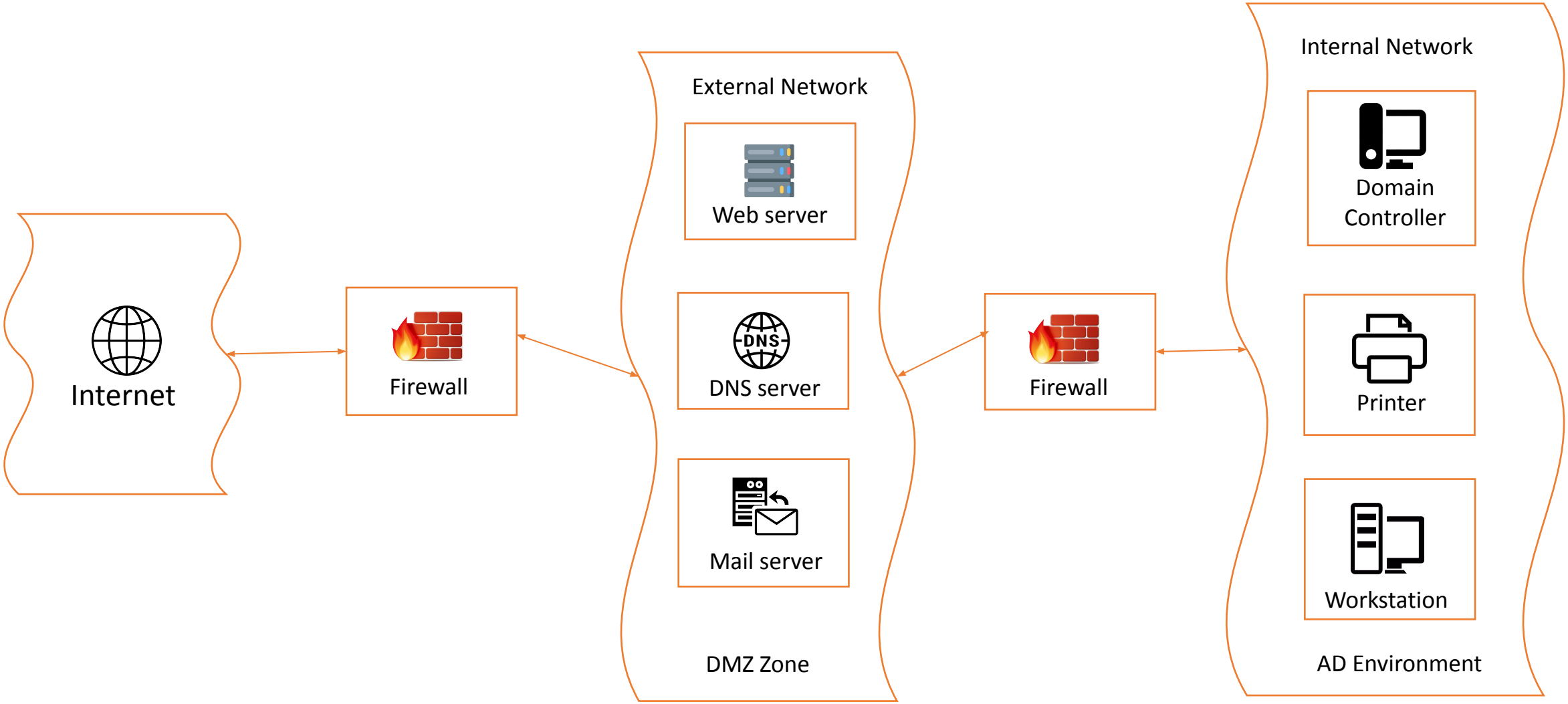
Hybrid Multi Cloud Environment is combination of On-premise and Multi Cloud Environment

- On-Premise Environment
- AWS Cloud
- Azure Cloud
- Google Cloud [GCP]

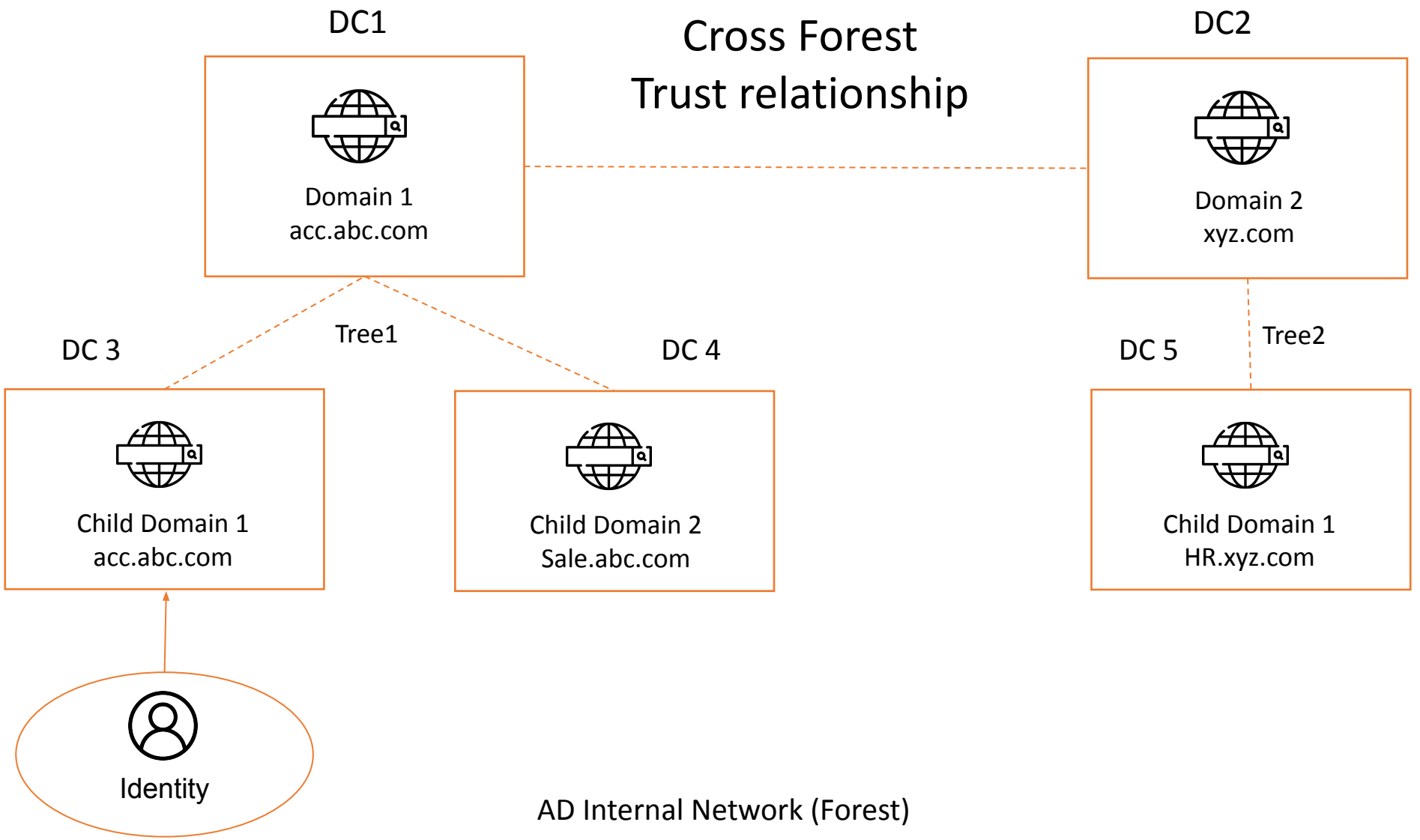
Hybrid Multi Cloud Environment



1.1 On-Premise AD Architecture



Network Architecture of Active Directory Environment



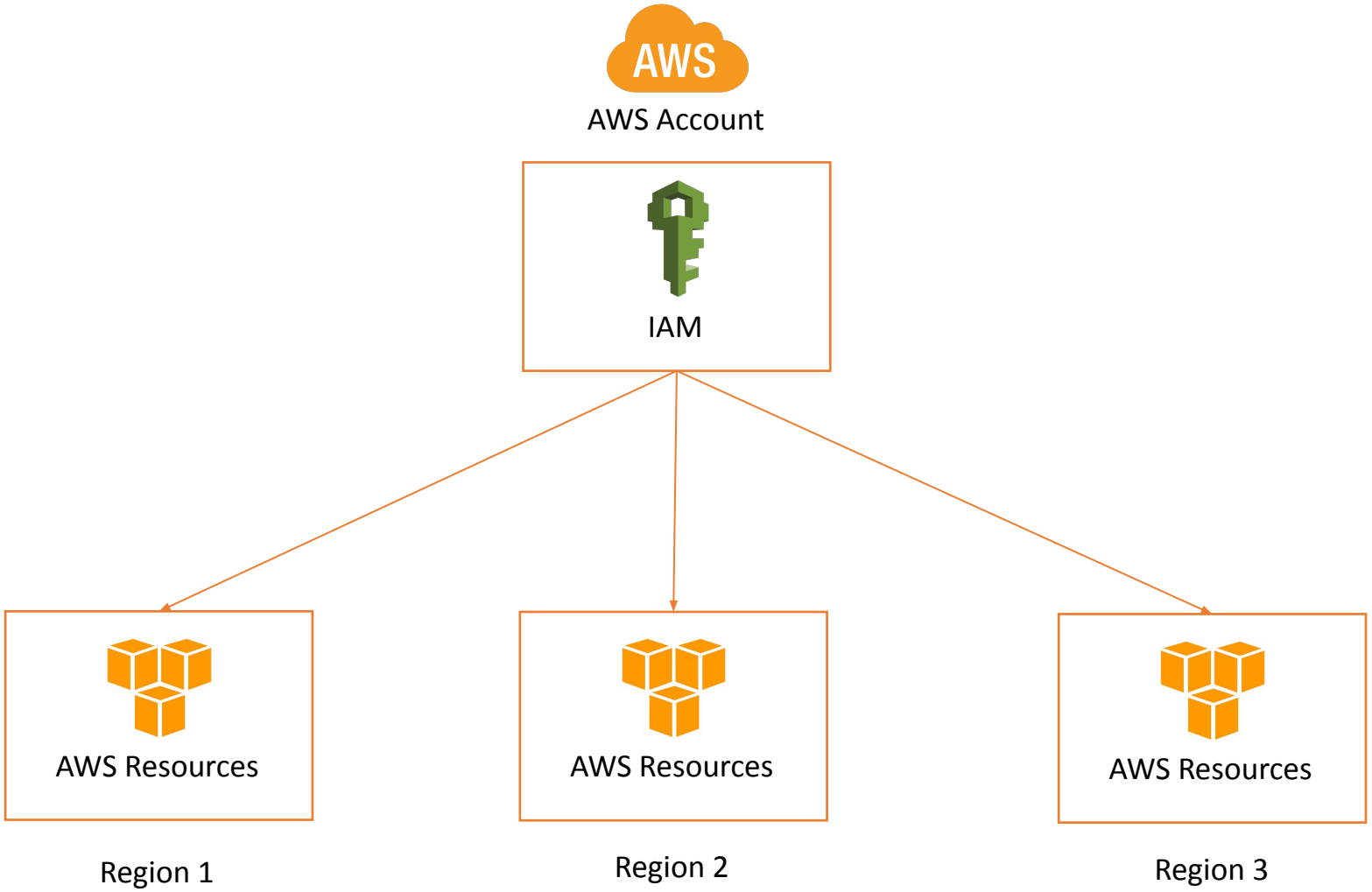
1.2 Multi Cloud Architecture

- A multi cloud environment is one where an enterprise uses more than one cloud platform.

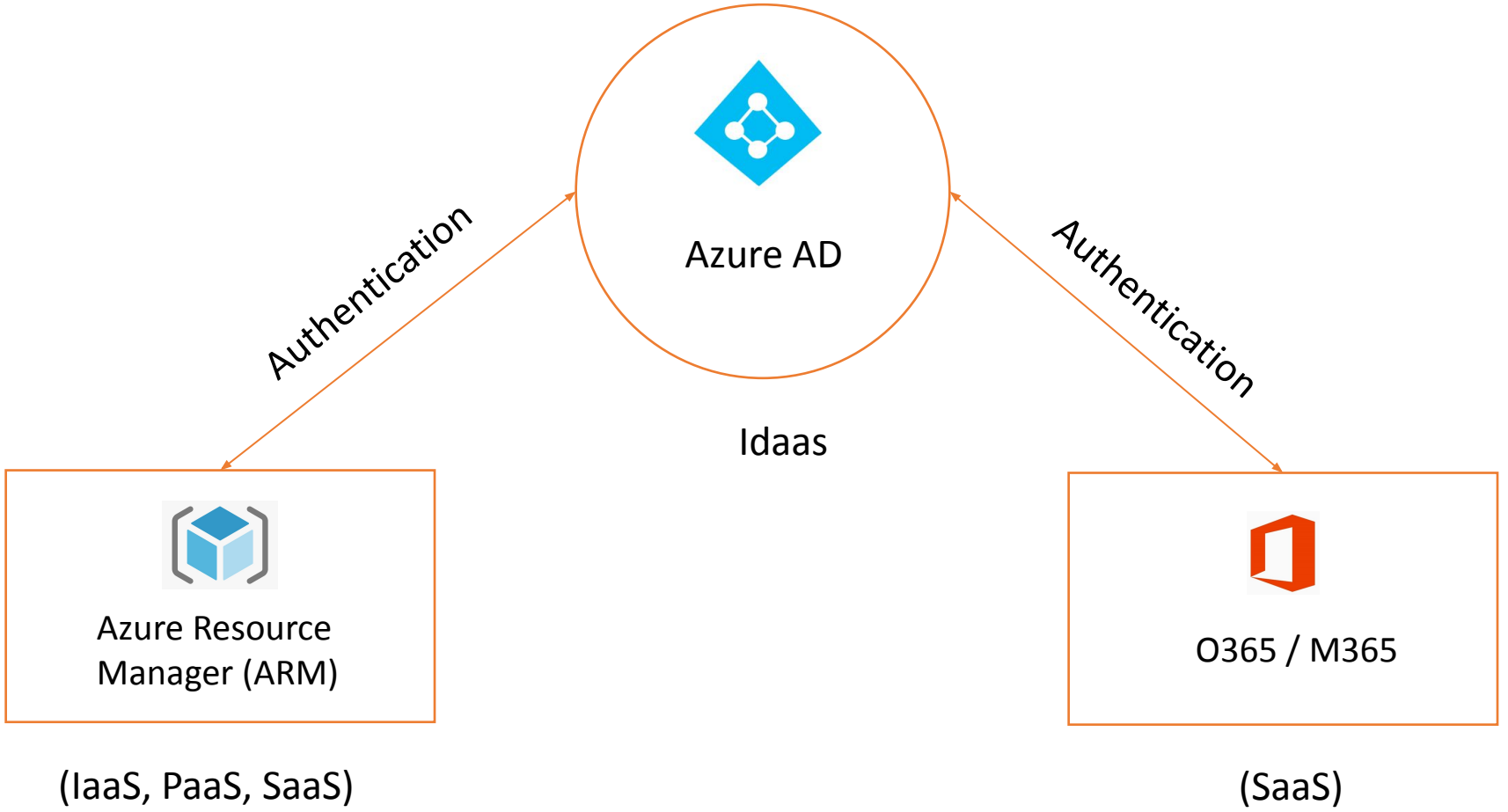
- A multicloud can be comprised of public, private, and edge clouds to achieve the enterprise's end goals.

- Public cloud is an IT model where on-demand computing services and infrastructure are managed by a third-party provider and shared with multiple organizations using the public Internet.
  - Amazon Web Service [AWS]
  - Microsoft Azure
  - Google Cloud Platform [GCP]
  - IBM Cloud
  - Oracle Cloud

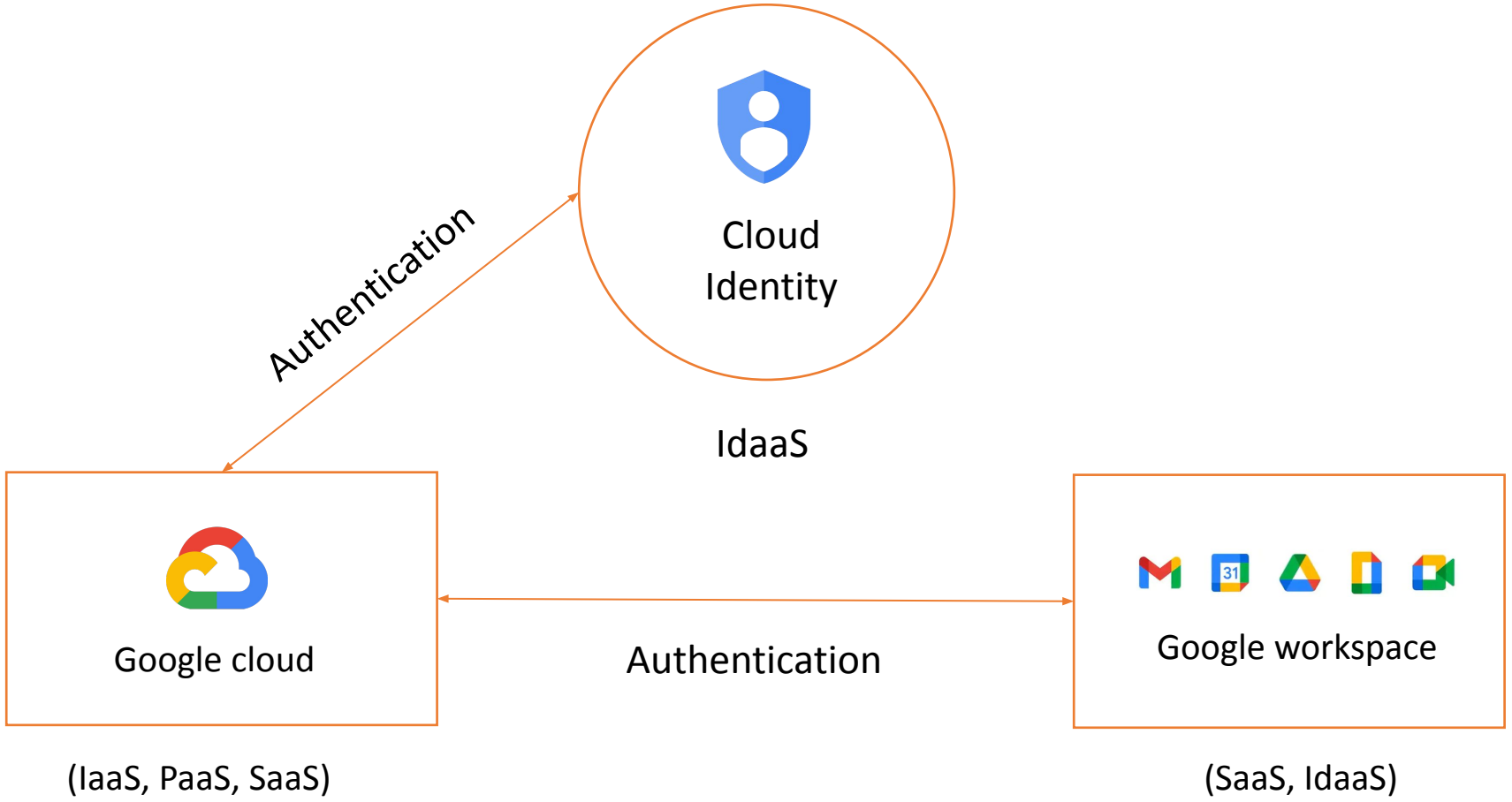
AWS Single Account Architecture



Azure Working Model



GCP Working Model

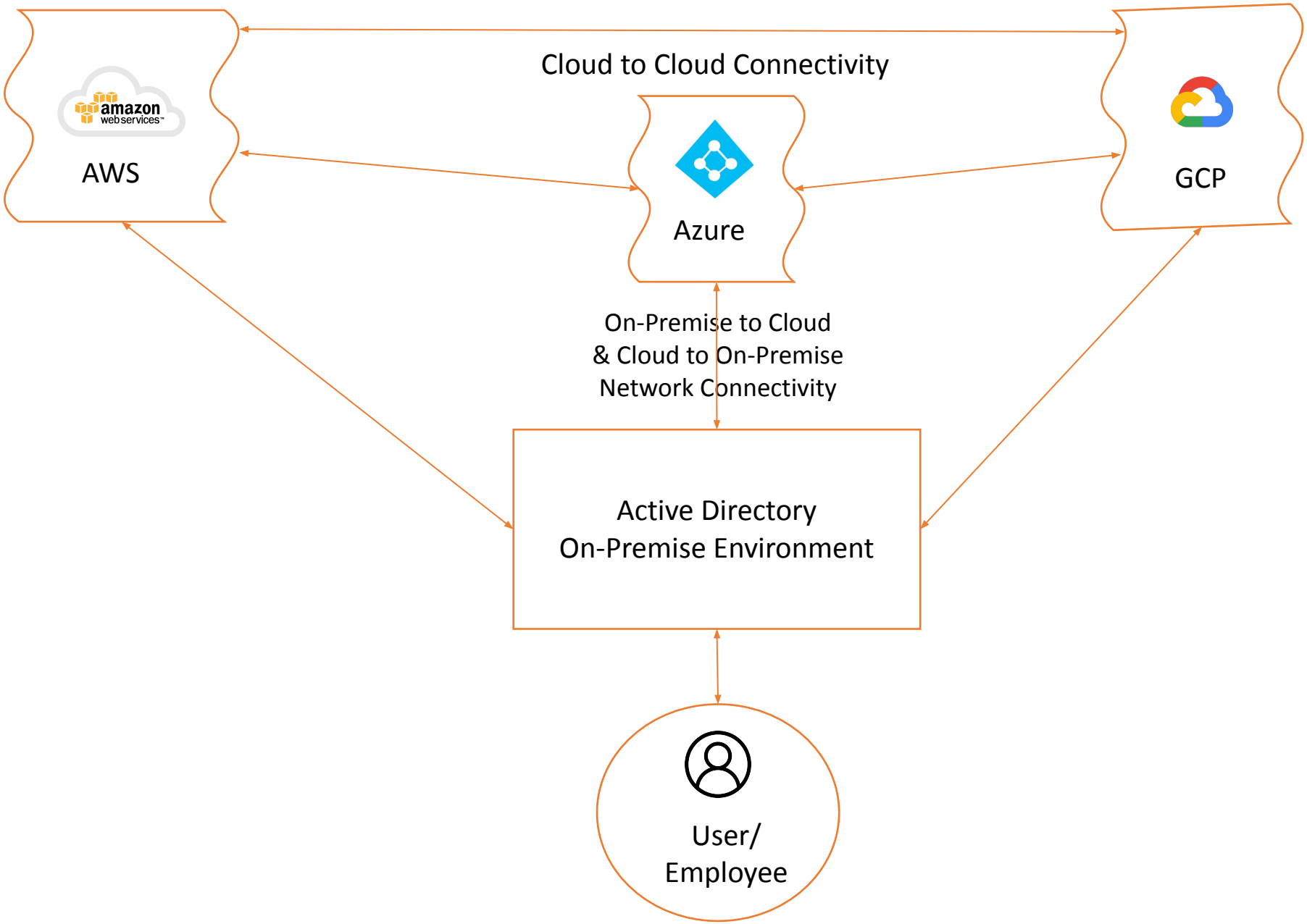


1.3 Hybrid Multi Cloud Architecture

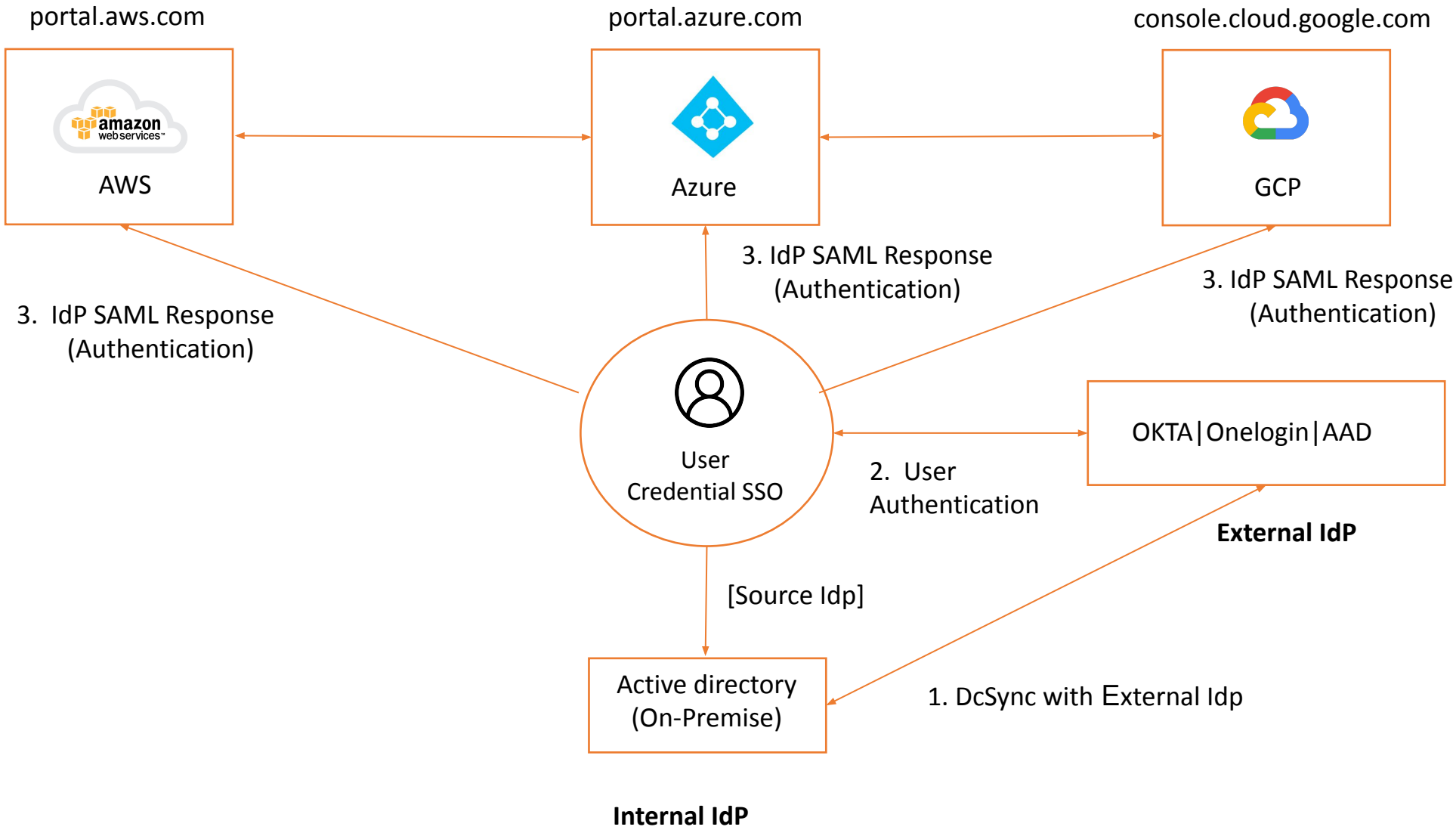
- A hybrid cloud becomes multi-cloud when there are more than one public cloud service combined with on-premise environment.

- An organization use service in hybrid multi cloud environment -
  - On-Premise
    - Active Directory
  - AWS
    - AWS SSO
    - AWS Cloud
  - Azure
    - Azure Active Directory
    - Azure Resource Manager
    - O365
  - GCP
    - Cloud Identity
    - Google Cloud
    - Google Workspace / G-Suite

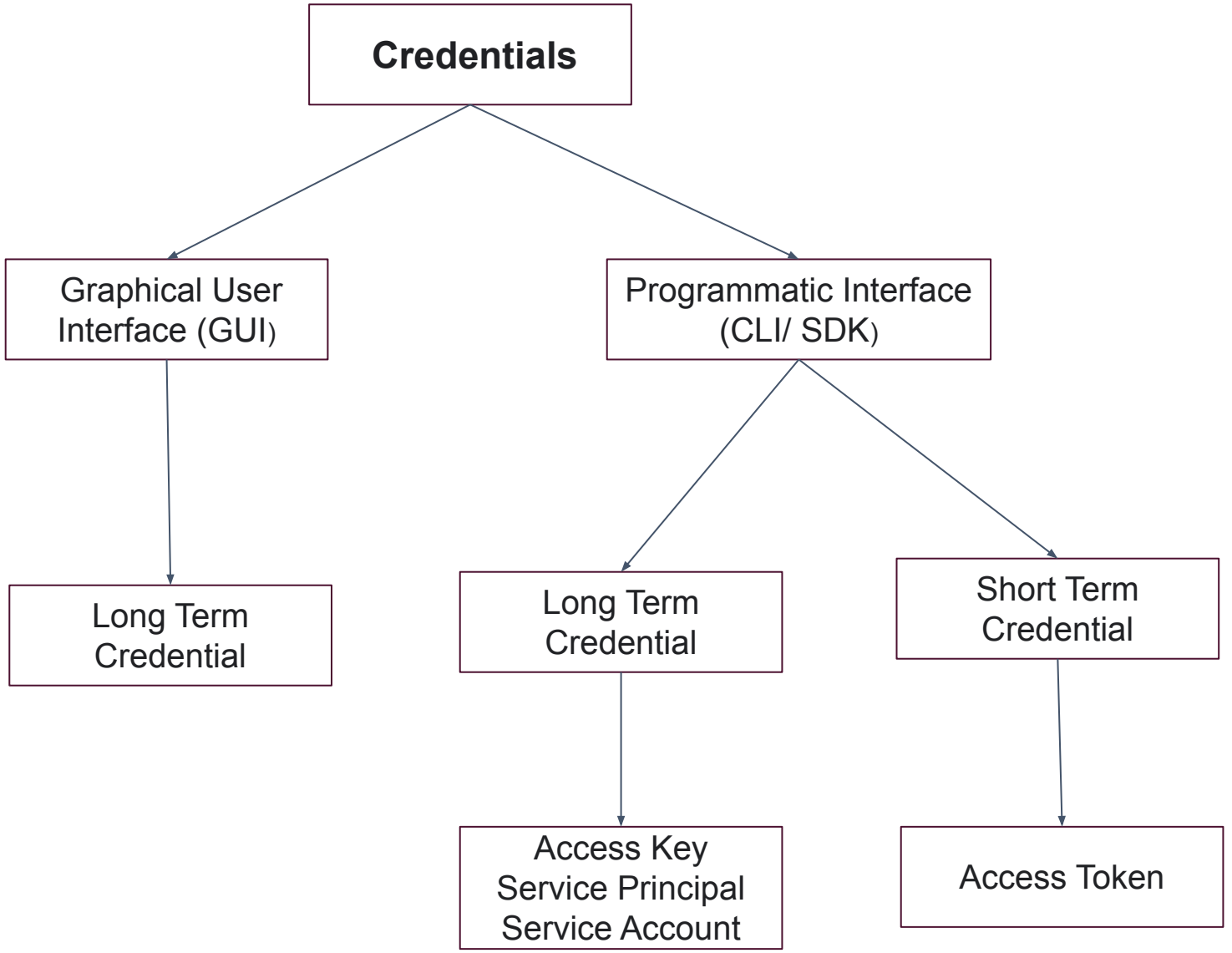
Network Connectivity between Cloud & On-Premise



Identity Federation from On-Premise to Cloud



Credentials in Hybrid Multi Cloud Environment



2. Introduction to AWS Cloud Red Teaming

2.1 Overview of AWS Cloud

Introduction:

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings.

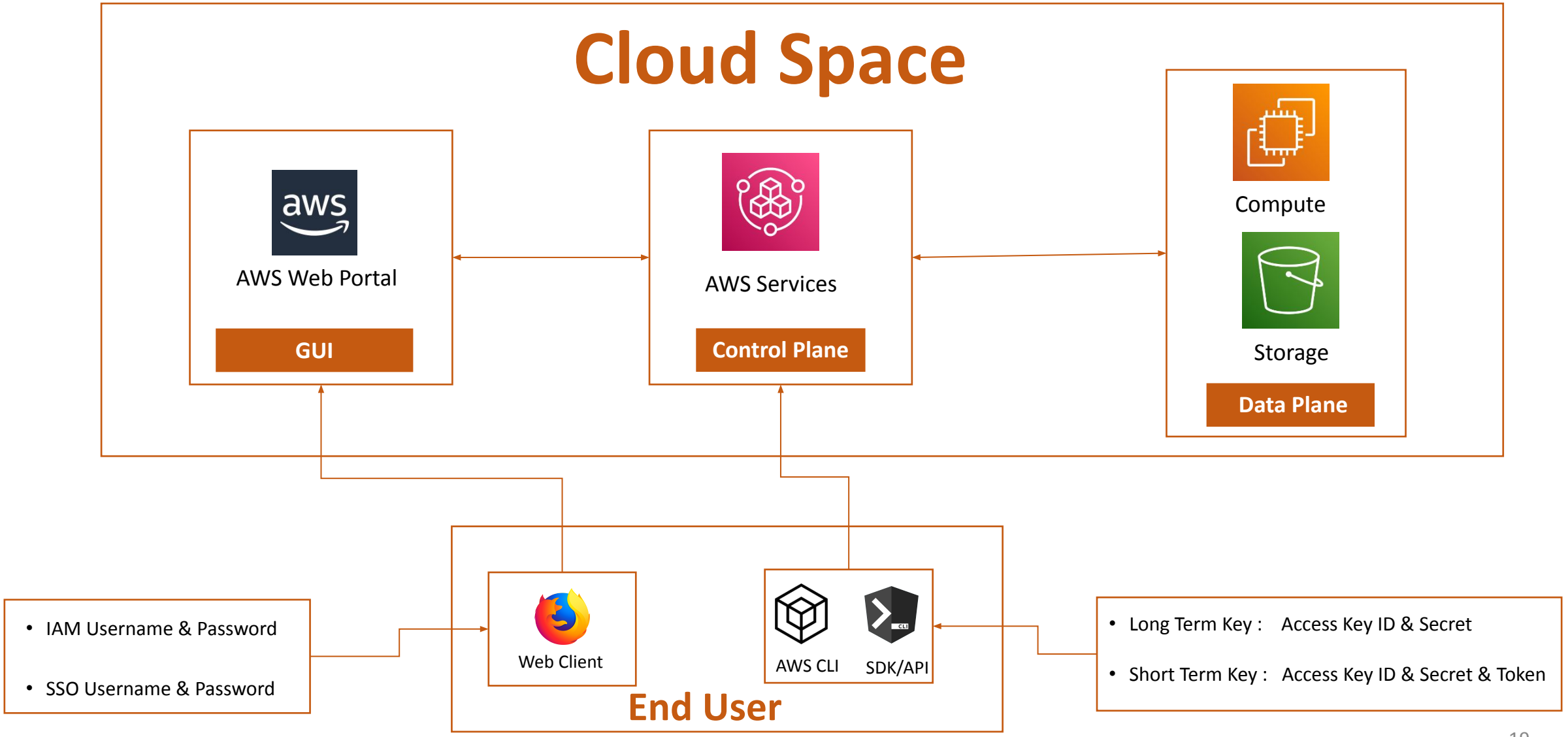
Regions:

AWS has the concept of a Region, which is a physical location around the world where aws have cluster data centers.

Availability Zones:

Region is further divided into logical data centers, which is called availability zones. AWS have

*AWS have 77 Availability Zones within 24 geographic regions around the world.

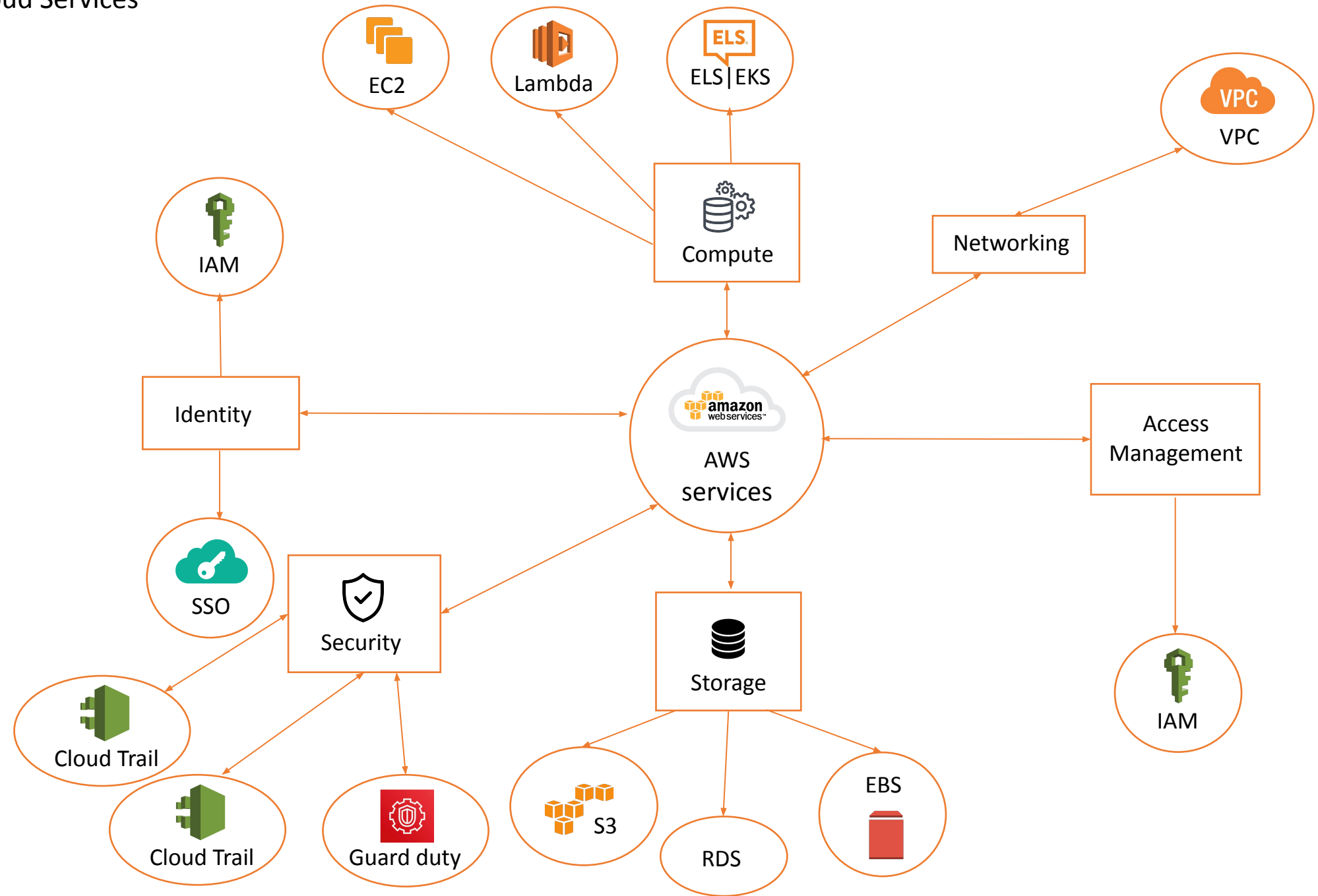

Cloud Space
AWS Web Portal
GUI
AWS Services
Control Plane
Compute
Storage
Data Plane
Web Client
AWS CLI
SDK/API
End User
• IAM Username & Password
• SSO Username & Password
• Long Term Key : Access Key ID & Secret
• Short Term Key : Access Key ID & Secret & Token

AWS Cloud Authentication Credentials -

- Console Access -
  - IAM Username & Password
  - SSO USername & Password

- Programmatic Access -
  - Access Key & Secret Key
  - Access Key, Secret Key & Access Token

AWS Cloud Services



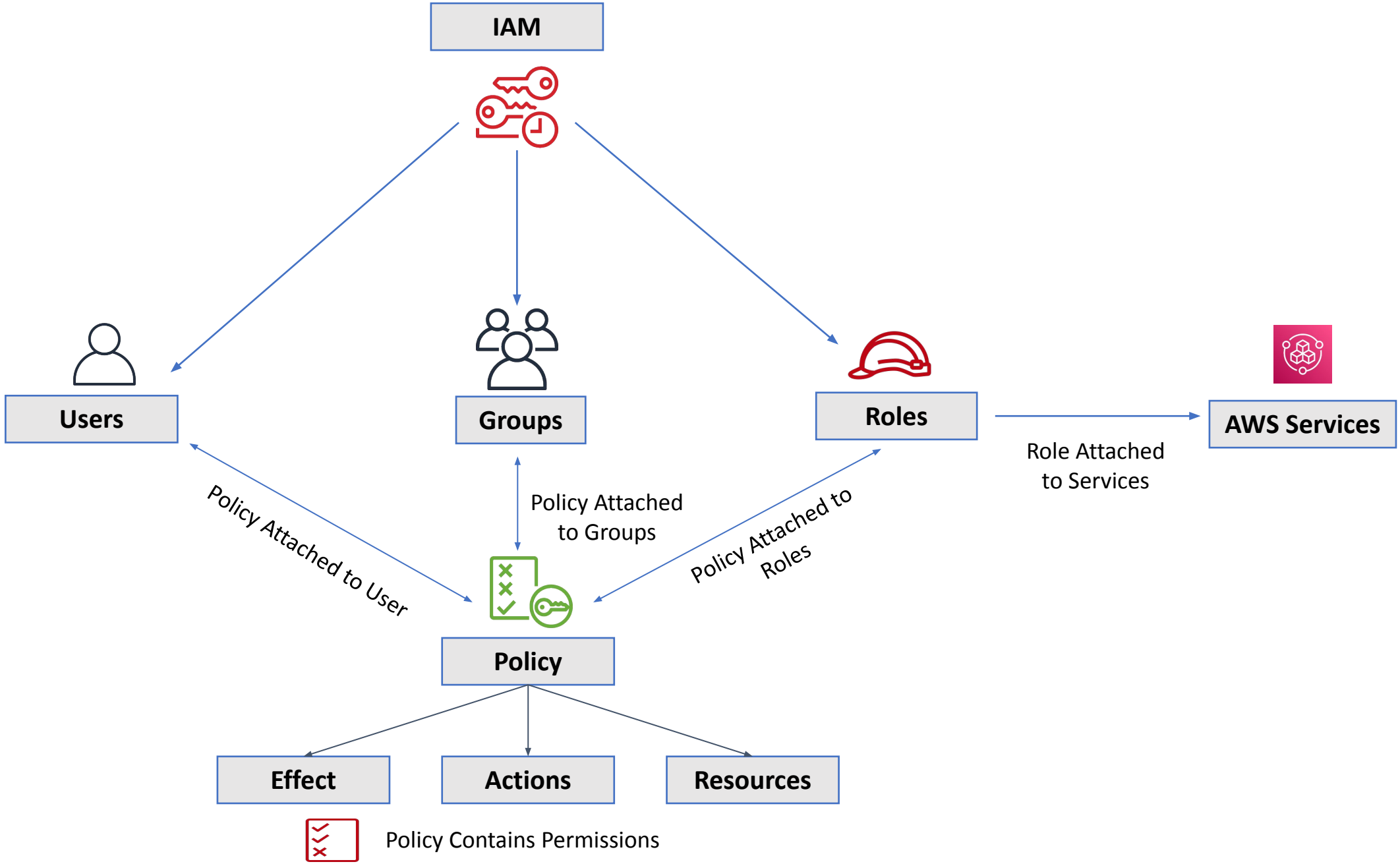
2.2 Identity and Access Management

IAM :

- AWS Identity and Access Management (IAM) enables you to manage access to AWS services and resources securely.
- IAM allow you can create and manage AWS users and groups and use permissions to allow and deny their access to AWS resources.

AWS IAM allows:

1. Manage IAM users, groups and their access.
2. Manage IAM roles and their permissions.
3. Manage federated users and their permissions.


IAM
Users
Groups
Roles
AWS Services
Role Attached to Services
Policy Attached to Groups
Policy Attached to User
Policy Attached to Roles
Policy
Effect
Actions
Resources
Policy Contains Permissions

A. Users

- An AWS Identity and Access Management (IAM) user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS.
- A user in AWS consists of a name and credentials.

AWS Services Access Type :

1. Programmatic access
   - Access key ID
   - Secret access key

2. AWS Management Console access
   - Username
   - Password

B. Groups

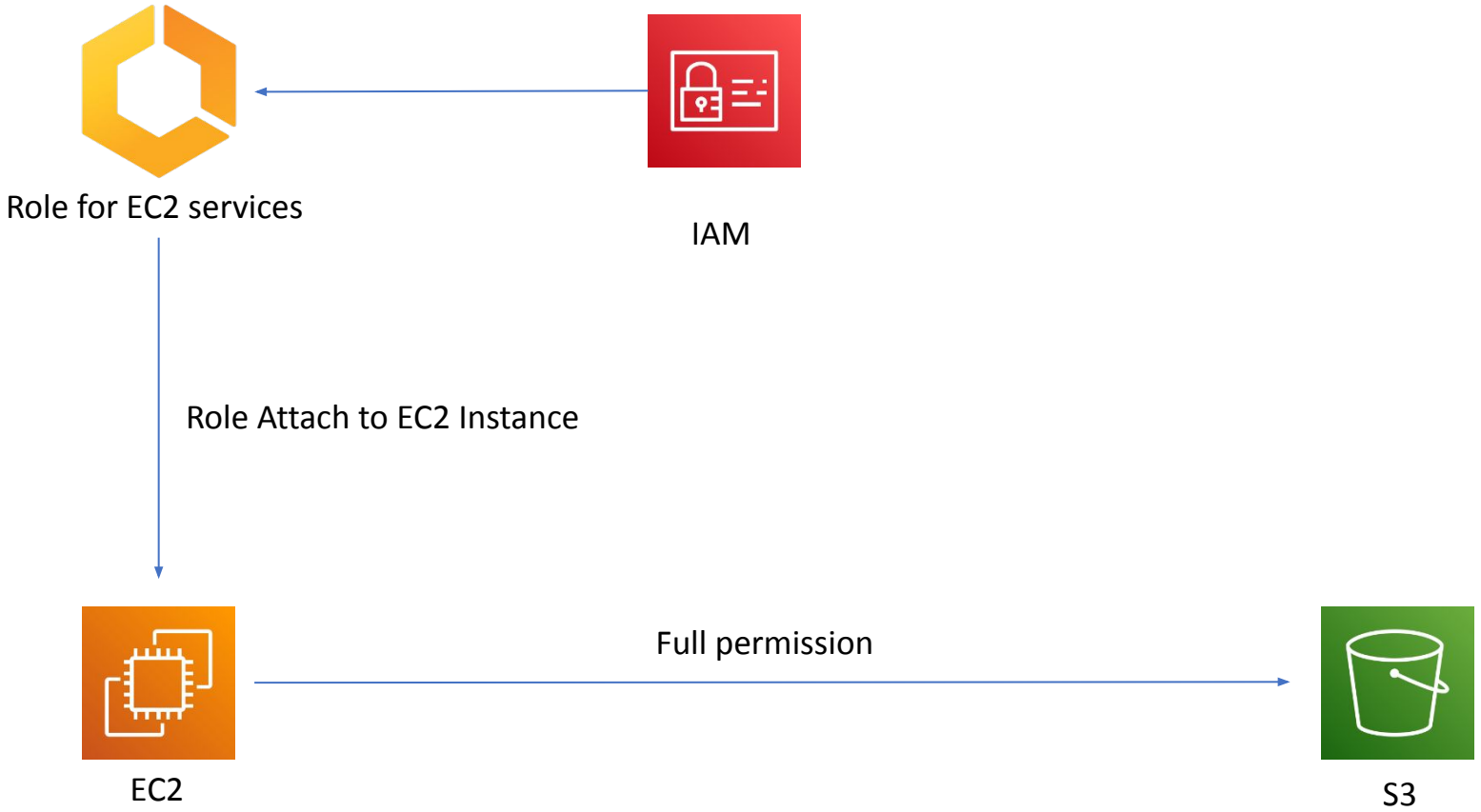
An IAM group is a collection of IAM users. Groups let you specify permissions for multiple users, which can make it easier to manage the permissions for those users

Following are some important characteristics of groups:

1. A group can contain many users, and a user can belong to multiple groups.
2. Groups can't be nested; they can contain only users, not other groups.

C. Roles

- An IAM role is an IAM entity that defines a set of permissions for making AWS service requests.
- IAM roles are associated with AWS services such as EC2, RDS etc.
- IAM roles are a secure way to grant permissions to entities that you trust. Examples of entities include the following:
  - IAM user in another account
  - Application code running on an EC2 instance that needs to perform actions on AWS resources
  - An AWS service that needs to act on resources in your account to provide its features
- IAM roles issue keys that are valid for short durations, making them a more secure way to grant access.



EC2 Instance can access S3 Bucket

D. Policies

- IAM policies define permissions for an action to perform the operation.
- For example, if a policy allows the GetUser action, then a user with that policy can get user information from the AWS Management Console, the AWS CLI, or the AWS API.
- Policies can be attached to IAM identities (users, groups or roles) or AWS resources.

Policy Data :

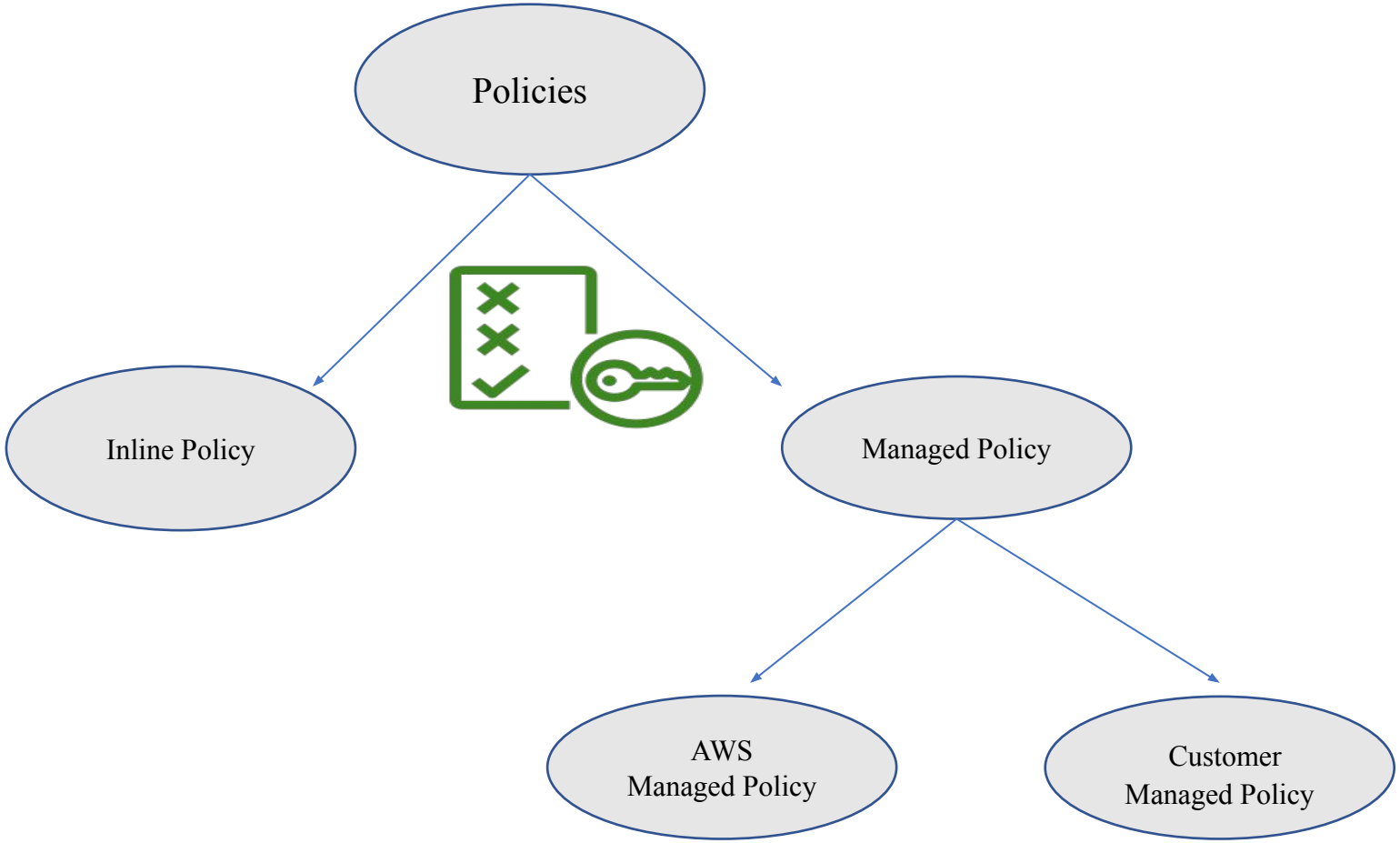
1. Effect - Use to Allow or Deny Access
2. Action - Include a list of actions (Get, Put, Delete) that the policy allows or denies.
3. Resource - A list of resources to which the actions apply

Policy types:

1. Inline Policies - An inline policy is a policy that's embedded in an IAM identity (a user, group, or role)

2. Managed Policies -

    - AWS Managed Policies
    - Customer Managed Policies



E. Security Token Service (STS)

- AWS Security Token Service (AWS STS) is a web service that enables you to request temporary, limited-privilege credentials for AWS IAM users or for users that you authenticate (federated users).
- STS allow user to give temporary access of aws resource using token.

Temporary credentials Contains :

1. Access key ID
2. Secret access key
3. Security token (session token)

Temporary credentials

1. STS - https://sts.amazonaws.com (Users OR Other AWS Account Users / Services)

   STS Endpoint:

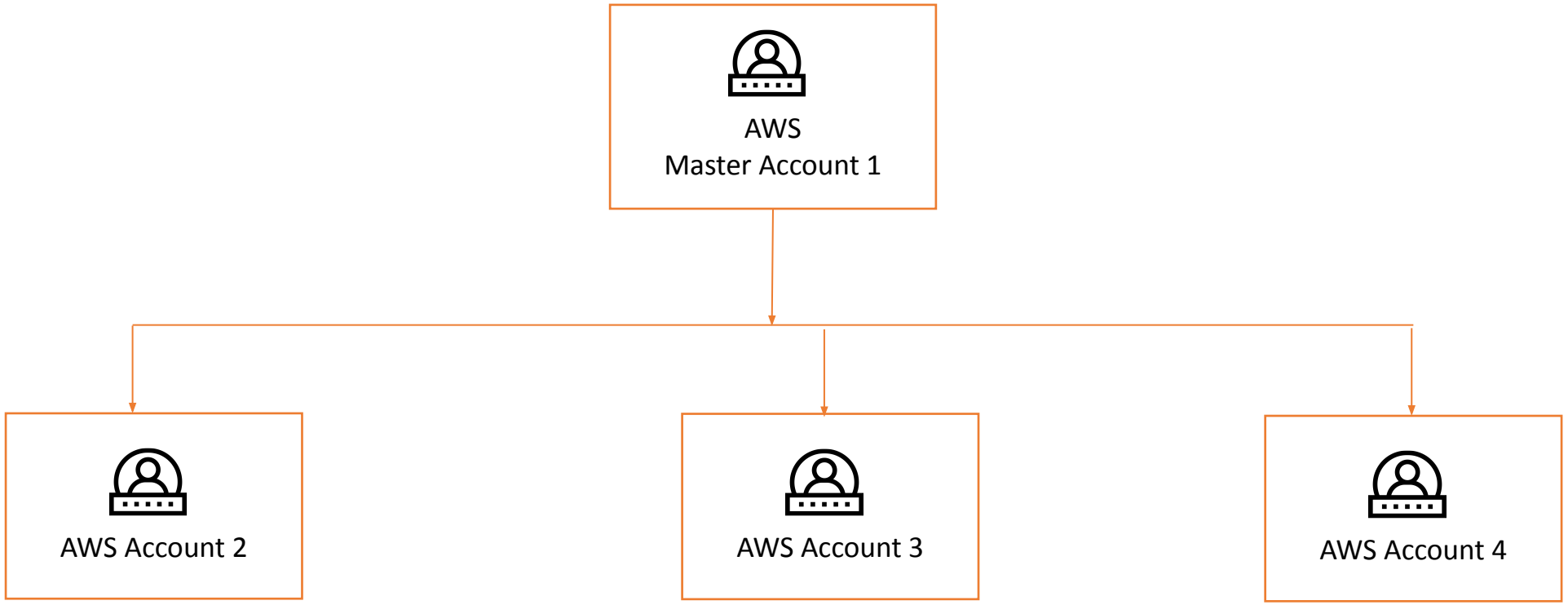
   - Global STS Endpoint - https://sts.amazonaws.com
   - Regional STS Endpoint - https://sts.**Region-Name**.amazonaws.com

2. AWS Metadata - http://169.254.169.254 (IAM Roles - Same AWS Account)

2.3 AWS Organization

- AWS Organizations is an account management service which allows to manage multiple AWS accounts centrally.

- AWS Organizations helps you to centrally manage billing, control access, compliance, security, and share resources across your AWS accounts.



AWS Organizations Components :

Master Account

- A master account is the AWS account you use to create your organization.
- Using master account, you can create other accounts in your organization and invite other accounts to join your organization, and remove accounts from your organization.
- There is only one master/root

Member Account

- A member account is an AWS account, other than the master account in the organization.

Organization Unit (OU)

- An organizational unit is a group of AWS accounts within an organization.

Service Control Policy (SCP)

- SCP is a document that describes controls to be attached to the entire organization, OUs, or individual AWS accounts.

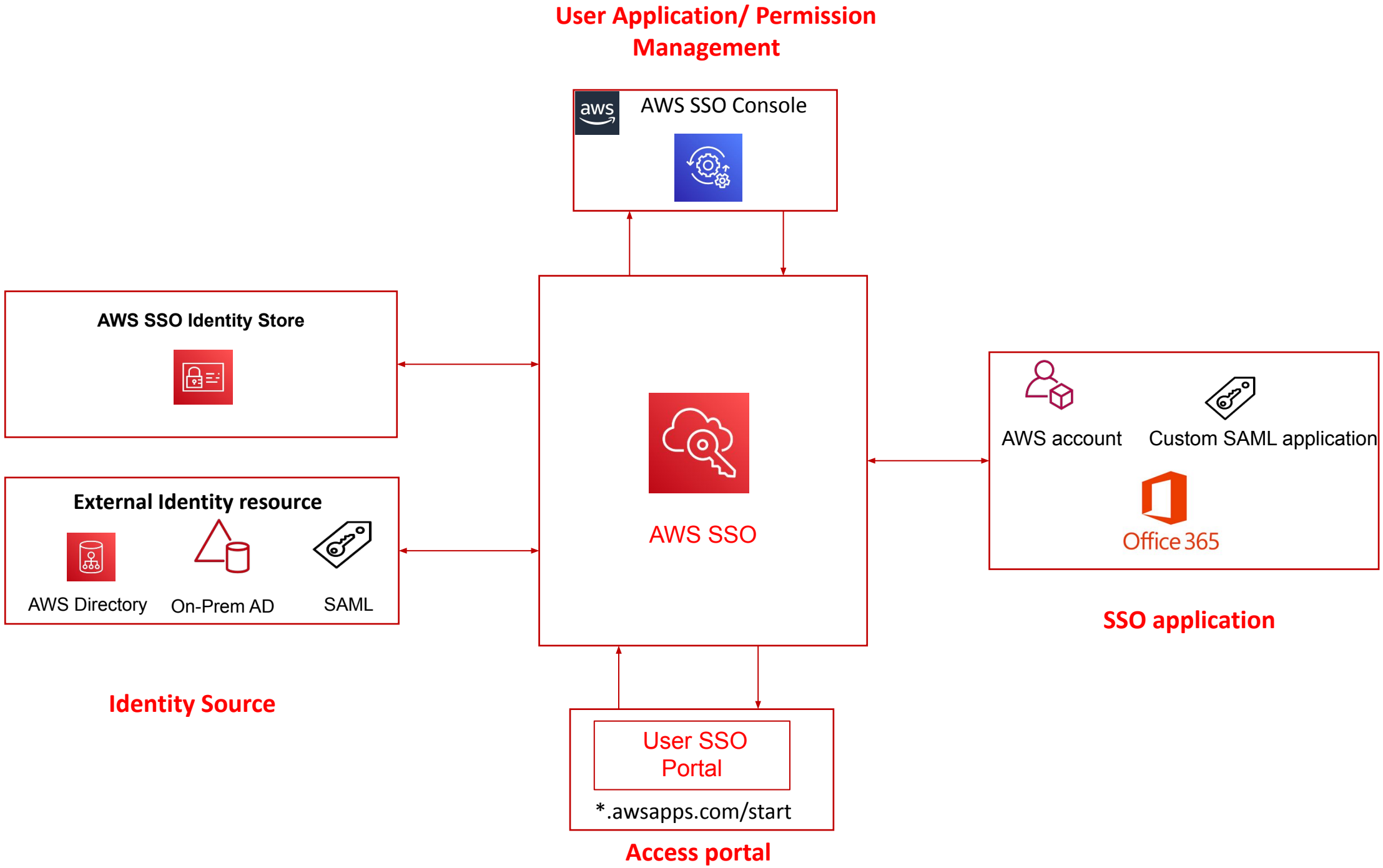
- Policy defines the services and actions that users or a role can perform.

2.4 AWS Single Sign On [SSO]

- AWS Single Sign-On (SSO) makes it easy to centrally manage access to multiple AWS accounts and business applications and provide users with single sign-on access to all their assigned accounts and applications from one place.

- With AWS SSO, we can easily manage access and user permissions to all your accounts in AWS Organizations centrally.

- AWS SSO also includes built-in integrations to many business applications, such as Salesforce, Box, Office 365, Azure, GCP and Custom SAML Applications.


User Application/ Permission Management
AWS SSO Console
AWS SSO Identity Store
External Identity resource
AWS Directory
On-Prem AD
SAML
Identity Source
AWS SSO
AWS account
Custom SAML application
Office 365
SSO application
User SSO Portal
*.awsapps.com/start
Access portal

User Identity Source :

- AWS SSO’s identity store
- External identity store
    - Microsoft Active Directory (Customer Managed)
    - AWS Managed Active Directory
    - Okta Universal Directory
    - Azure Active Directory (Azure AD)
    - SAML 2.0 IdP

Permission Sets :

Permission sets define the level of access that users have to their assigned AWS accounts.

Applications :

It’s includes cloud applications and any custom applications that support identity federation with SAML 2.0

2.5 Cross Account Access [Assume Role]

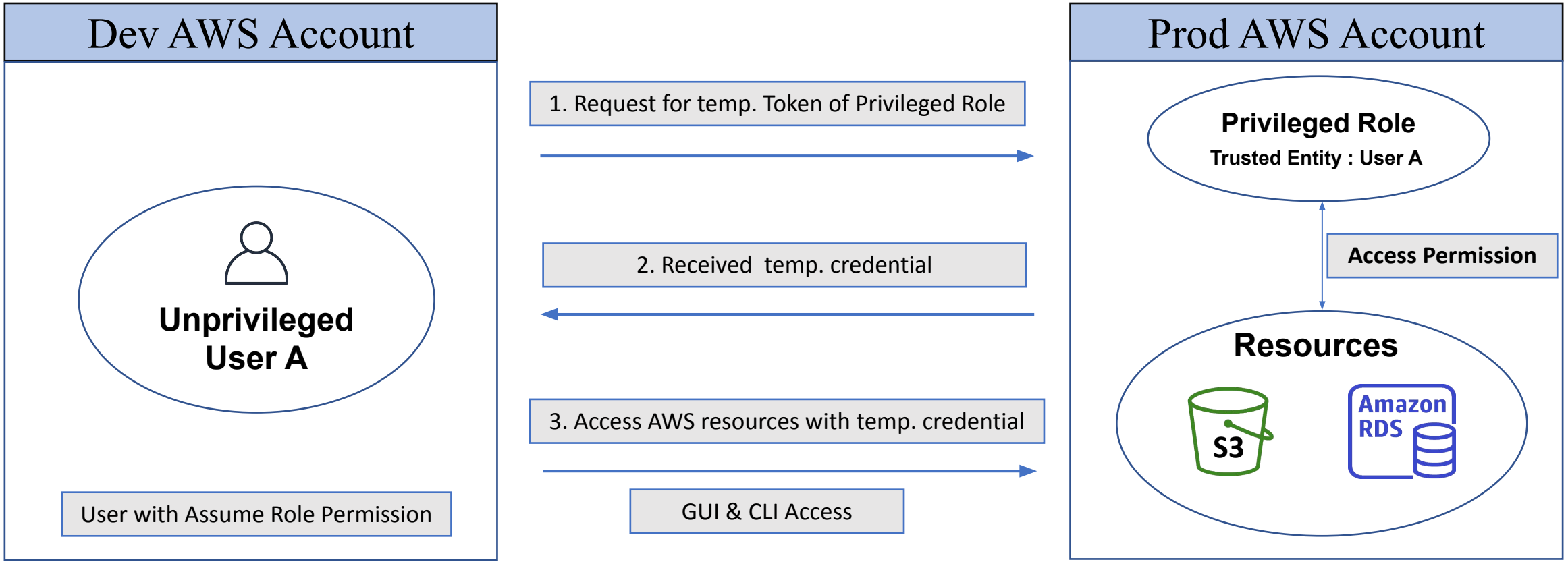


Fig : Cross AWS Account Resource Access With Assume Role

2.6 AWS Enumeration

- AWS Organization
- IAM
- Compute
- Storage
- Networking

Users:

List of IAM Users :

```
aws iam list-users
```

List the IAM groups that the specified IAM user belongs to :

```
aws iam list-groups-for-user --user-name user-name
```

List all manages policies that are attached to the specified IAM user :

```
aws iam list-attached-user-policies --user-name user-name
```

Lists the names of the inline policies embedded in the specified IAM user :

```
aws iam list-user-policies --user-name user-name
```

Groups :

List of IAM Groups:

```
aws iam list-groups
```

Lists all managed policies that are attached to the specified IAM Group :

```
aws iam list-attached-group-policies --group-name group-name
```

List the names of the inline policies embedded in the specified IAM Group:

```
aws iam list-group-policies --group-name group-name
```

Roles :

List of IAM Roles :

```
aws iam list-roles
```

Lists all managed policies that are attached to the specified IAM role :

```
aws iam list-attached-role-policies --role-name role-name
```

List the names of the inline policies embedded in the specified IAM role :

```
aws iam list-role-policies --role-name role-name
```

Policies:

List of IAM Policies :

```
aws iam list-policies
```

Retrieves information about the specified managed policy :

```
aws iam get-policy --policy-arn policy-arn
```

Lists information about the versions of the specified manages policy :

```
aws iam list-policy-versions --policy-arn policy-arn
```

Retrieved information about the specified version of the specified managed policy :

```
aws iam get-policy-version --policy-arn policy-arn --version-id version-id
```

Retrieves the specified inline policy document that is embedded on the specified IAM user / group / role :

```
aws iam get-user-policy --user-name user-name --policy-name policy-name
aws iam get-group-policy --group-name group-name --policy-name policy-name
aws iam get-role-policy --role-name role-name --policy-name policy-name
```

2. Introduction to Azure Cloud Red Teaming

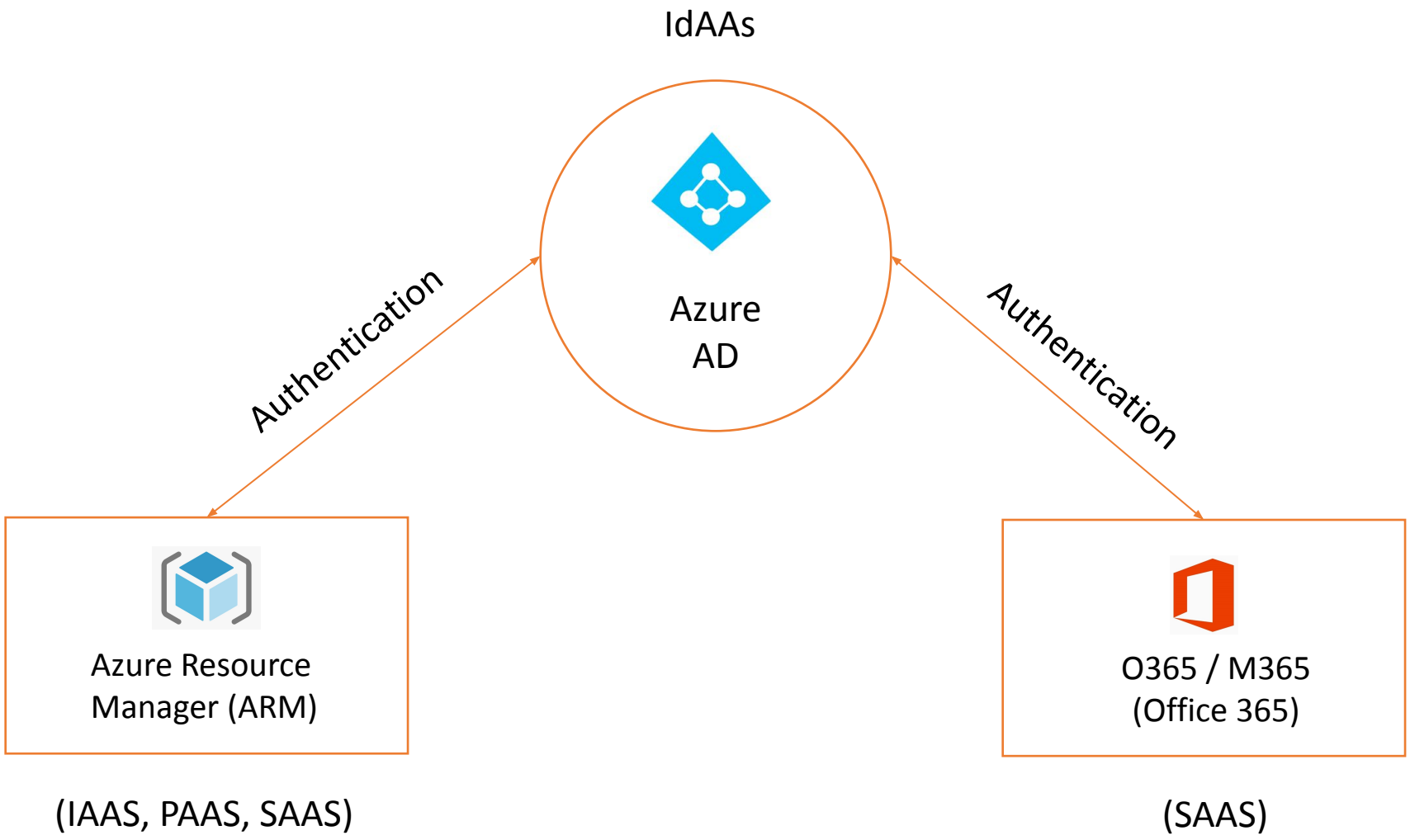
3.1 Azure Cloud Overview

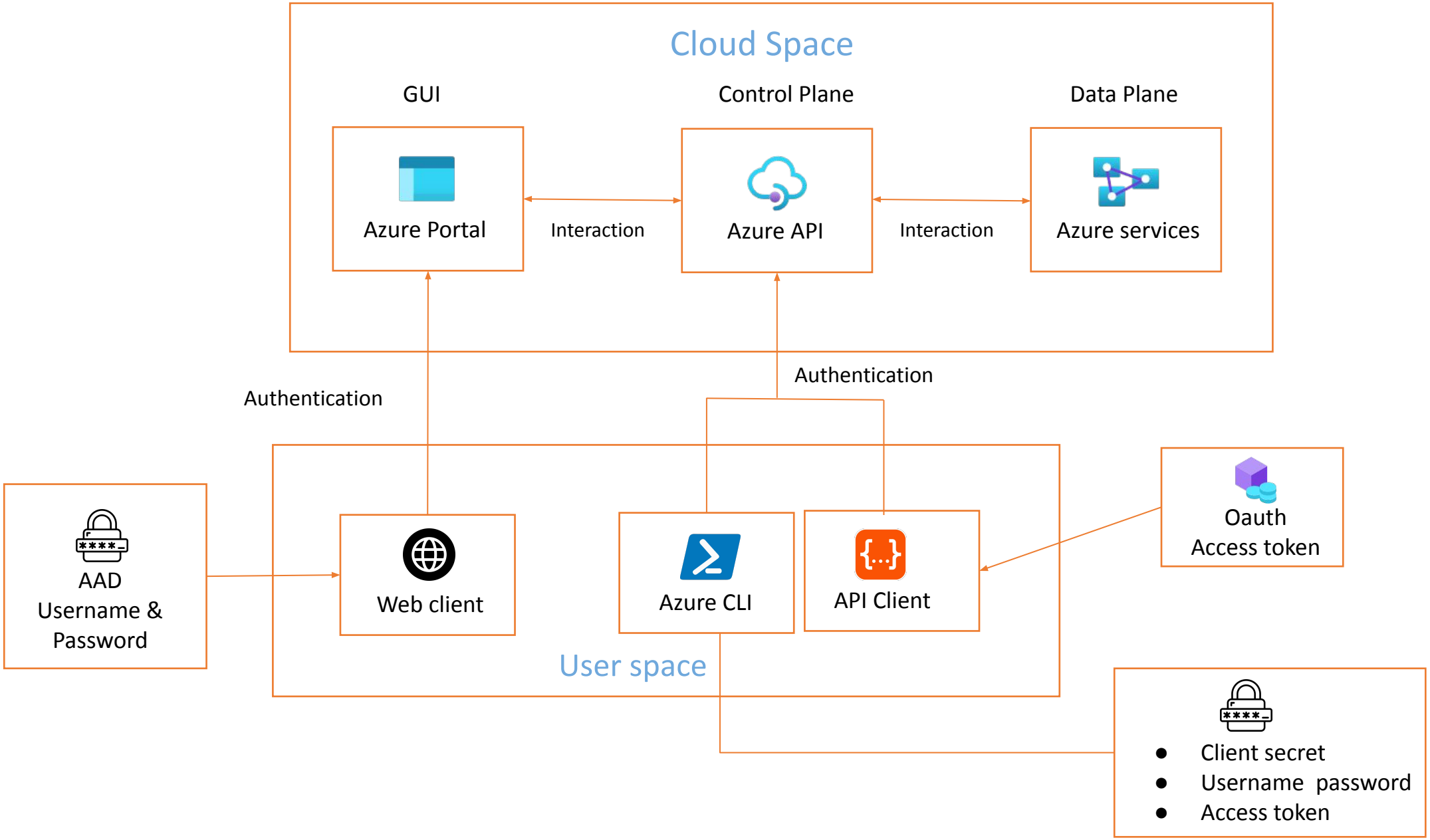
Introduction:

Microsoft Azure, commonly referred to as Azure, is a cloud computing service created by Microsoft for building, testing, deploying, and managing applications and services through Microsoft-managed data centers.

Three Main Components of Azure Cloud -

- Azure Active Directory [AAD] -
    - Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management service, which helps the employees sign in and access resources in cloud and on-premise.
- Azure Resource Manager [ARM] -
    - Azure Resource Manager (ARM) is **the native platform for infrastructure as code (IaC) in Azure**. It enables you to centralize the management, deployment, and security of Azure resources
- Office 365 [O365] -
    - Office 365 is a cloud-based suite of productivity & collaboration apps.


IdAAs
Azure
AD
Authentication
Authentication
Azure Resource
Manager (ARM)
O365 / M365
(Office 365)
(IAAS, PAAS, SAAS)
(SAAS)


Cloud Space
GUI
Control Plane
Data Plane
Azure Portal
Interaction
Azure API
Interaction
Azure services
Authentication
Authentication
AAD
Username &
Password
Web client
Azure CLI
API Client
Oauth
Access token
User space
Client secret
Username password
Access token

Azure Cloud Authentication Credentials :

- Console Access - [GUI]
  - Username & Password

- Programmatic Access - [CLI / API]
  - Username & Password
  - Client ID & Secret / Certificate
  - OAuth Access Token

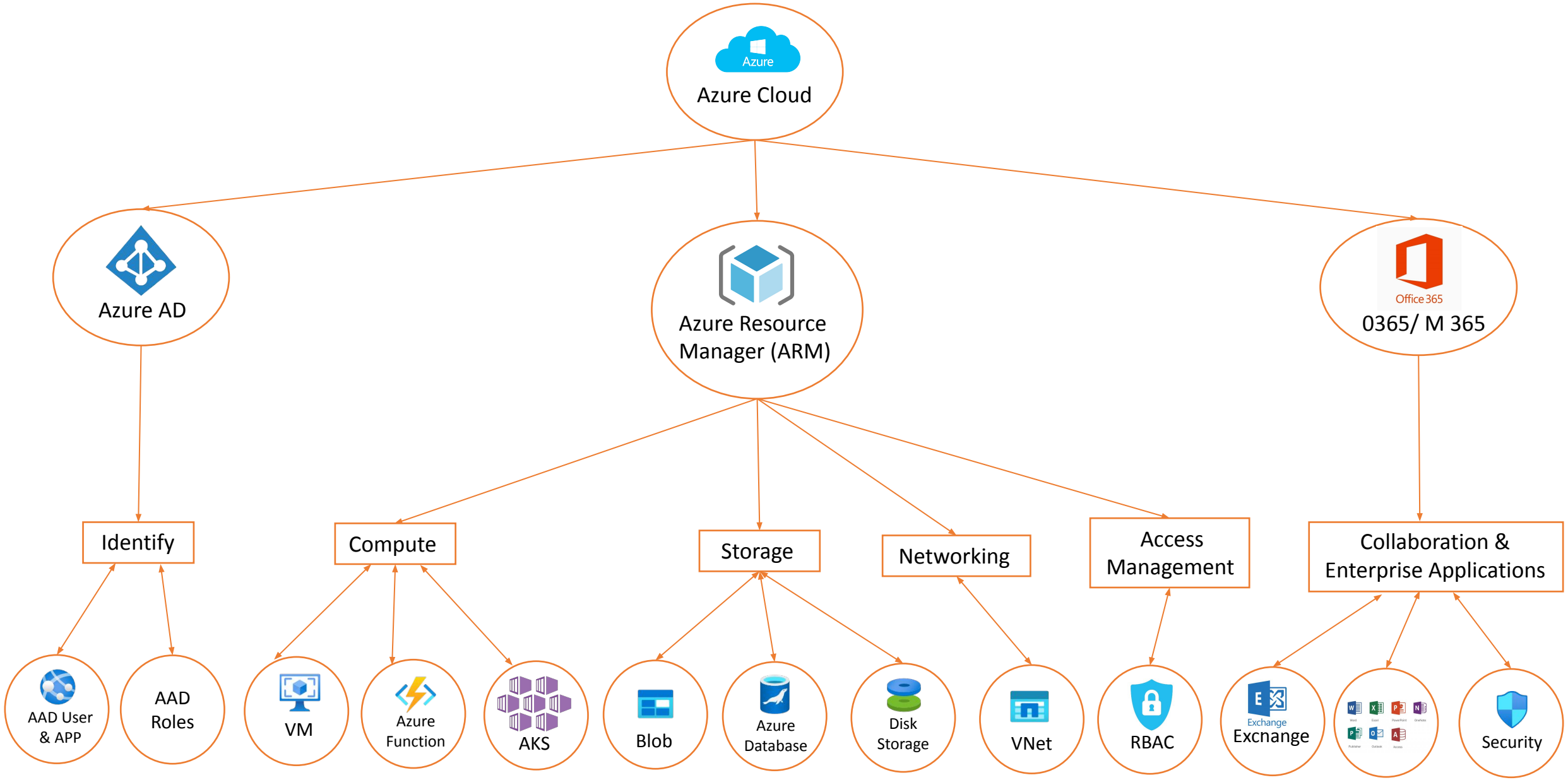
Azure + Azure AD + O365 Authentication Client :

- Azure Active Directory [AD]
  - Portal [aad.portal.azure.com]
  - Azure AD Module
  - Msol Module
  - Microsoft Graph API [graph.microsoft.com]

- Azure Resources [ARM]
  - Portal [portal.azure.com]
  - Az Cli Module
  - Az PowerShell Module
  - Azure Management Rest API [management.azure.com]

- Office 365 / Microsoft 365
  - User Portal [portal.office.com]
  - Admin portal [admin.microsoft.com]
  - Azure AD Module
  - Msol Module
  - O365 Management API [manage.office.com]
  - O365 Mail API [outlook.office.com]
  - Microsoft Graph API [graph.microsoft.com]

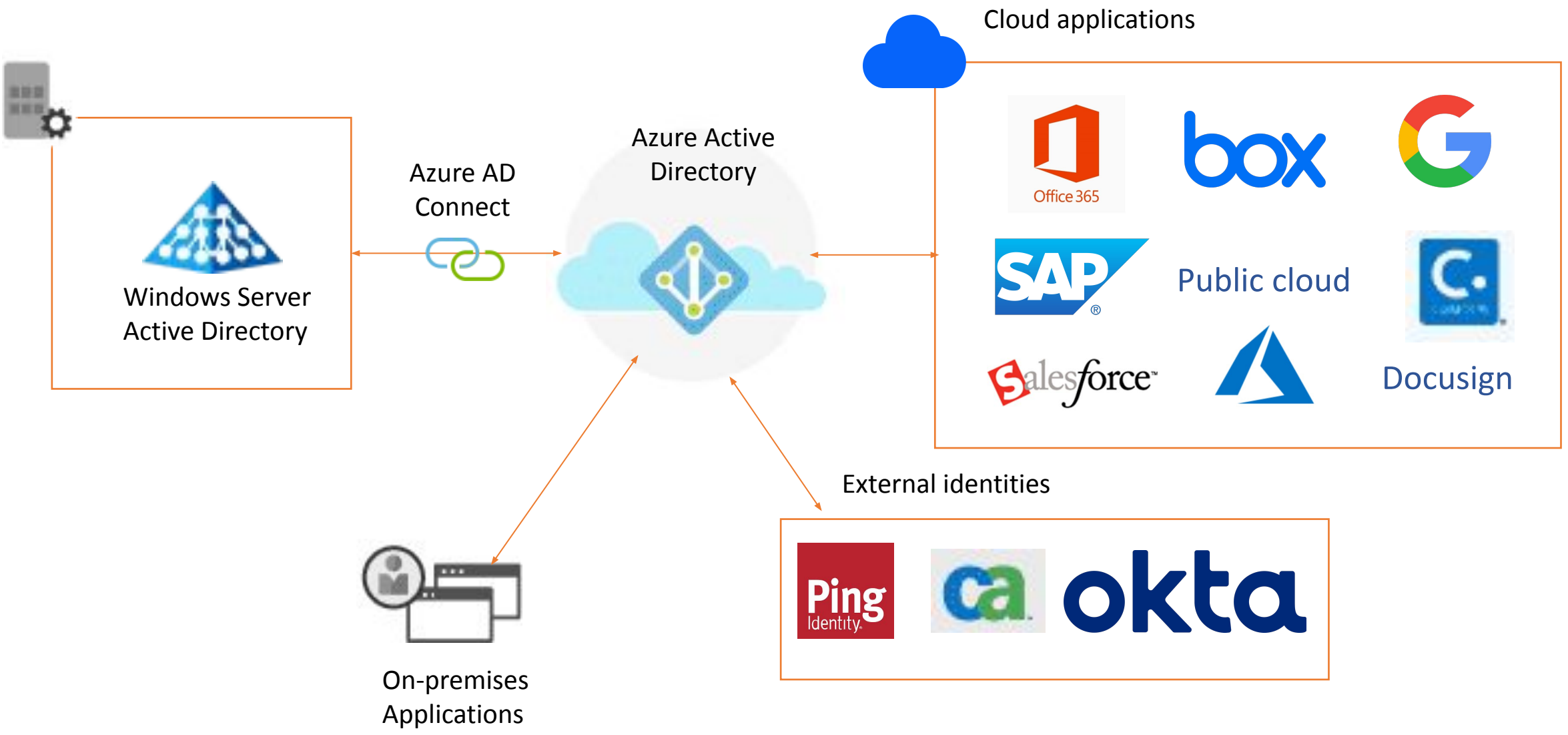
3.2 Azure Cloud Services



3.3 Azure Active Directory

- Azure Active Directory (Azure AD) is Microsoft's enterprise cloud-based identity and access management (IAM) solution.

- Azure AD is the backbone of the Office 365 system, and it can sync with on-premise Active Directory and provide authentication to other cloud-based systems via OAuth.


Cloud applications
Windows Server
Active Directory
Azure AD
Connect
Azure Active
Directory
Office 365
box
Public cloud
Docusign
Salesforce
External identities
Ping
Identity
okta
On-premises
Applications

Authentication Methods with Azure AD -

A. Portal

   https://aad.portal.azure.com

B. PowerShell

   - Azure-AD Module
   - Msol Module

C. CLI

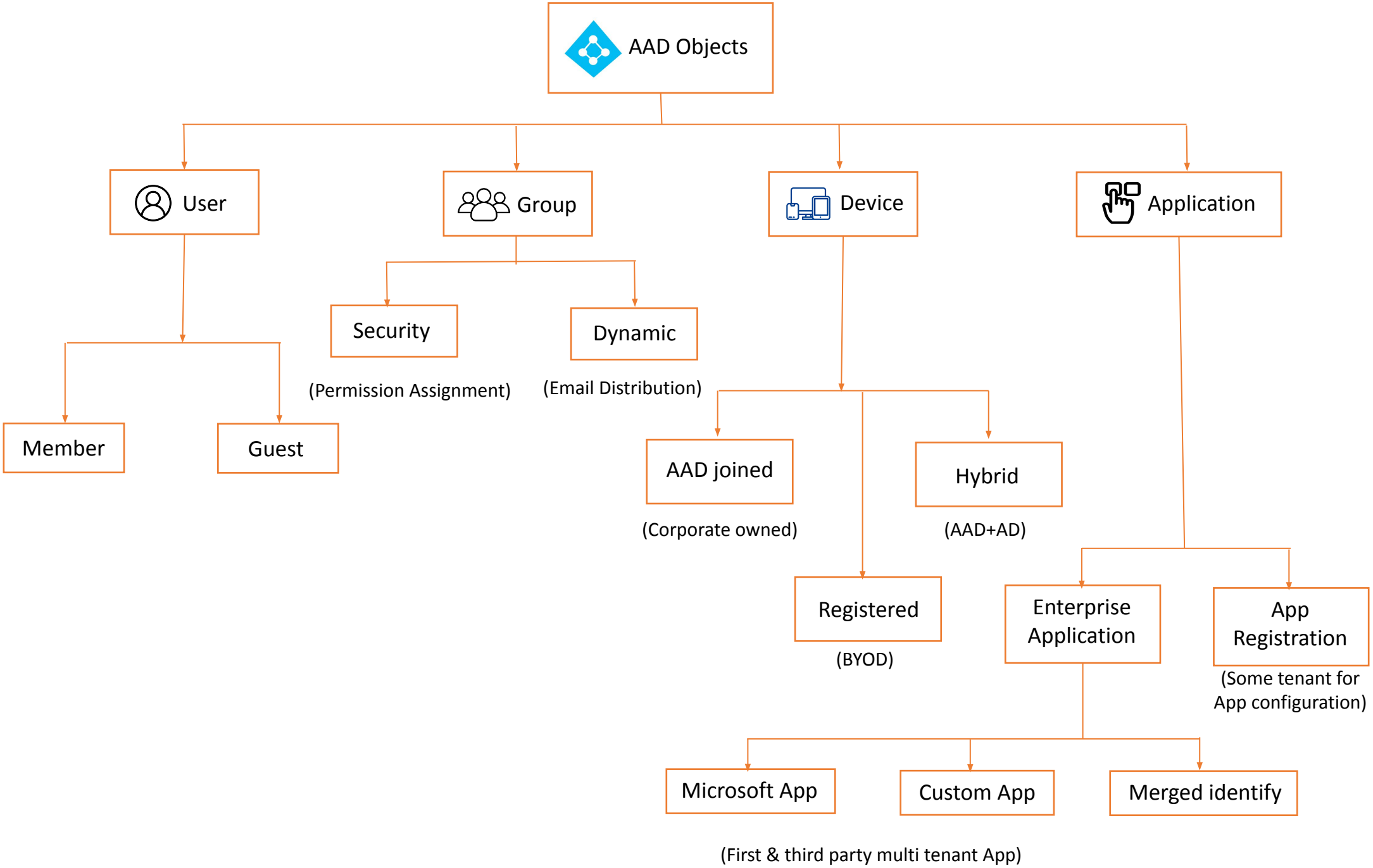
   - Az Module

D. API

   - Microsoft Graph API [graph.microsoft.com]

Azure AD Objects -

- Each azure ad object has an unique id associated with it, called object id.
- Each aad object has its own property.
- List of aad objects -
  - Users
  - Groups
  - Devices
  - Applications


AAD Objects
User
Group
Device
Application
Member
Guest
Security
(Permission Assignment)
Dynamic
(Email Distribution)
AAD joined
(Corporate owned)
Hybrid
(AAD+AD)
Registered
(BYOD)
Enterprise Application
App Registration
(Some tenant for App configuration)
Microsoft App
Custom App
Merged identify
(First & third party multi tenant App)

- **Users**

  - User Type
    - Member
      - User is a primary member of customer tenant.
      - Member have two type of security principal in aad -
        - username@domain-name.onmicrosoft.com
        - username@fqdn-domain-name
    - Guest -
      - Guest user can be part of multiple tenant.
      - Guest user has security principal in aad -
        - username#EXT#@domain.onmicrosoft.com

  - Identity Source
    - Azure Active Directory
    - Window Server AD
    - External Azure Active Directory

- **Groups**
    - Security Groups -
        - It's used to assign permissions to members of a group
        - Membership can be static or dynamic.
        - Group owner can manage security group.

    - Microsoft Groups -
        - Microsoft 365 Groups are used for collaboration between users, both inside and outside of company.

- **Devices**

  - Registered -
    - Personally owned corporate enabled
    - Authentication to the device is with a local id or personal cloud id
    - Authentication to corporate resources using a user id on AAD.

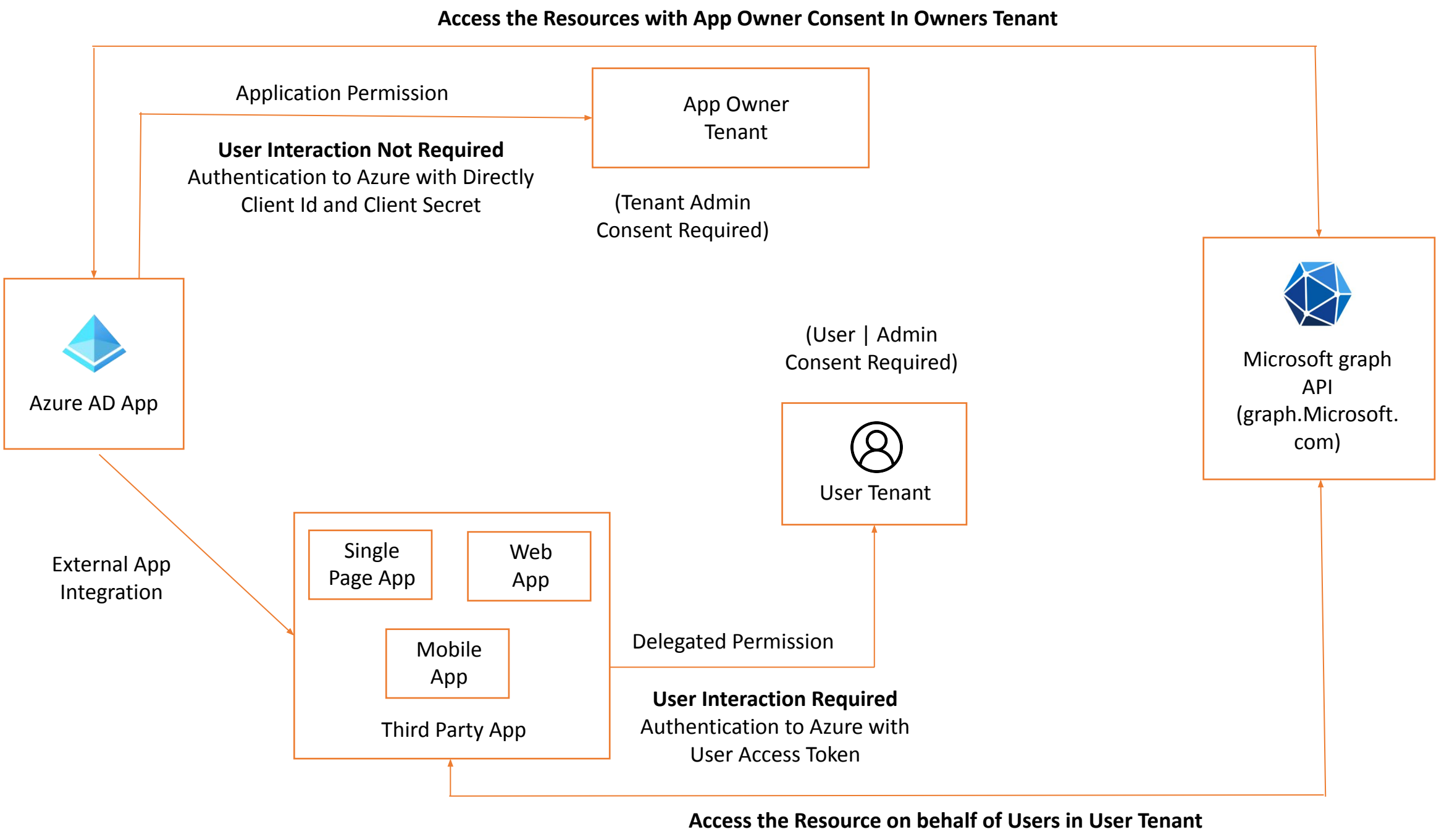
  - Azure AD Joined –
    - Corporate owned and managed devices
    - Authenticated using a corporate id that exists on Azure AD.
    - Authentication is only through AAD

  - Hybrid Joined (AAD + On-Premise AD) -
    - corporate owned and managed devices
    - Authenticated using a corporate user id that exists at local AD & on AAD.
    - Authentication can be done using both: On-Prem AD & Azure AD.

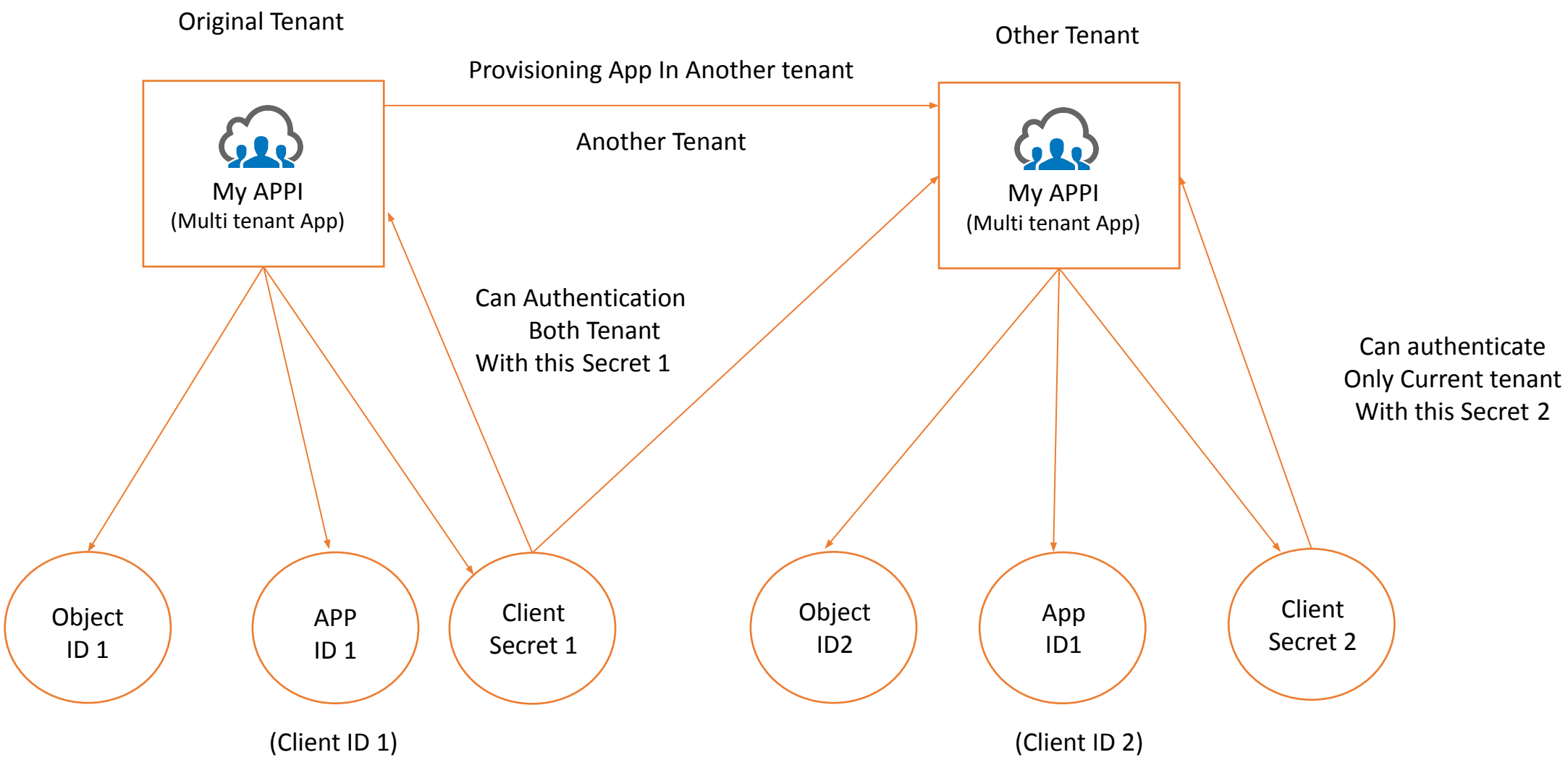
Applications

- **Application Object**
  - It comes under "**App Registration**" blade in AAD
  - "App registration" contains apps which are registered in the same tenant
  - This object acts as the template where you can go ahead and configure various things like API Permissions, Client Secrets, Branding, App Roles, etc.
  - The application object describes three aspects of an application:
    - How the service can issue tokens in order to access the application
    - Resources that the application might need to access
    - The actions that the application can take.
  - When we register an application in aad, its automatically create two objects -
    - Applications Object - Object ID : A unique identifier for each register application
    - Service Principal Object - Application ID / Client ID [Same as in enterprise application]
  - Application Attributes -
    - Owner - Owner of the registered application
    - API Permissions
      - Delegated Permission - User Interaction Required [Access the azure resources on the behalf of a user]
      - Application Permission- Permissions are assigned to the applications, User interaction not required. .
    - Client Secrets & Certificate
    - App Roles - It's used to assign permissions to the users to managed the registered application.
  - Consent -
    - Consent is the process of a user granting authorization to an application to access protected resources on their behalf.
    - Type of consent
      - Admin Consent - Admin consent flow is when an application developer directs users to the admin consent endpoint with the intent to record consent for the entire tenant (All Users).
      - User Consent - User consent flow is when an application developer directs users to the authorization endpoint with the intent to record consent for only the current user (Single User).

- **Service Principal Object**
  - It comes under "**Enterprise Application**" blade in AAD
  - A service principal is a concrete instance created from the application object and inherits certain properties from that application object
  - Service principal object defines -
    - What the app can actually do in the specific tenant
    - Who can access the app
    - What resources the app can access
  - In Enterprise Application there are two type of ID are there -
    - Object ID - A unique identifier for each service principal
    - Application ID - Service Principal Object [Same as in app registration]
  - "Enterprise Application" contains app which are registered in same tenant and app which are published by other companies [Other Tenants]
  - A service principal is created in each tenant where the application is used and references the globally unique app object.
  - Service Principal -
    - Service principal is unique identity belong to the same tenant or other tenant [e.g., Microsoft accounts etc.]
    - An Azure service principal is an identity created for use with applications, hosted services, and automated tools to access Azure resources.
    - This access is restricted by the roles assigned to the service principal, giving you control over which resources can be accessed and at which level.


Access the Resources with App Owner Consent In Owners Tenant
Application Permission
App Owner
Tenant
User Interaction Not Required
Authentication to Azure with Directly
Client Id and Client Secret
(Tenant Admin
Consent Required)
Microsoft graph
API
(graph.Microsoft.
com)
Azure AD App
(User | Admin
Consent Required)
User Tenant
External App
Integration
Single
Page App
Web
App
Mobile
App
Third Party App
Delegated Permission
User Interaction Required
Authentication to Azure with
User Access Token
Access the Resource on behalf of Users in User Tenant

Azure Multi Tenant Application

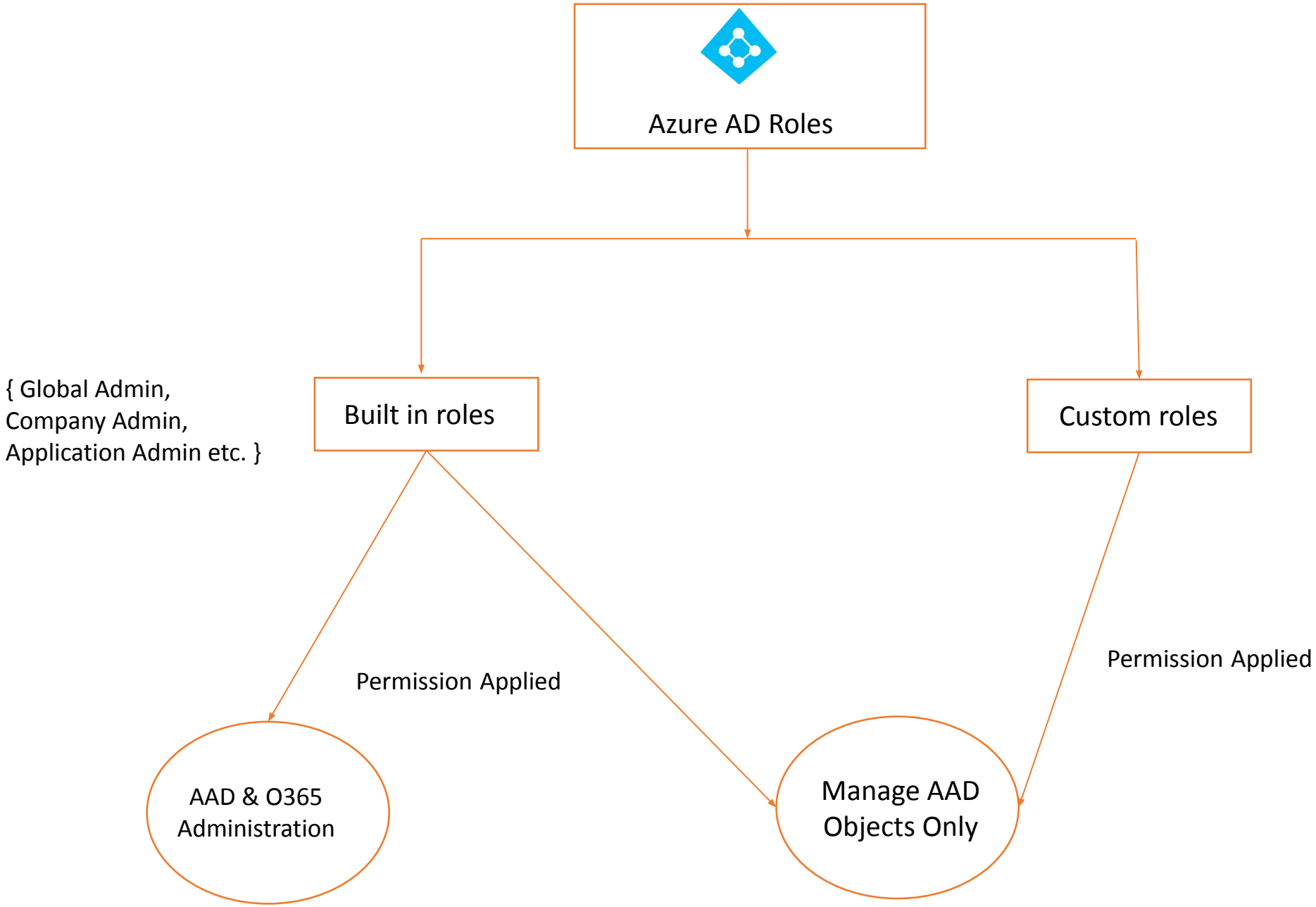


- **Roles**

  - Administrator or non-administrator needs to manage Azure AD resources, you assign them an Azure AD role that provides the permissions they need.

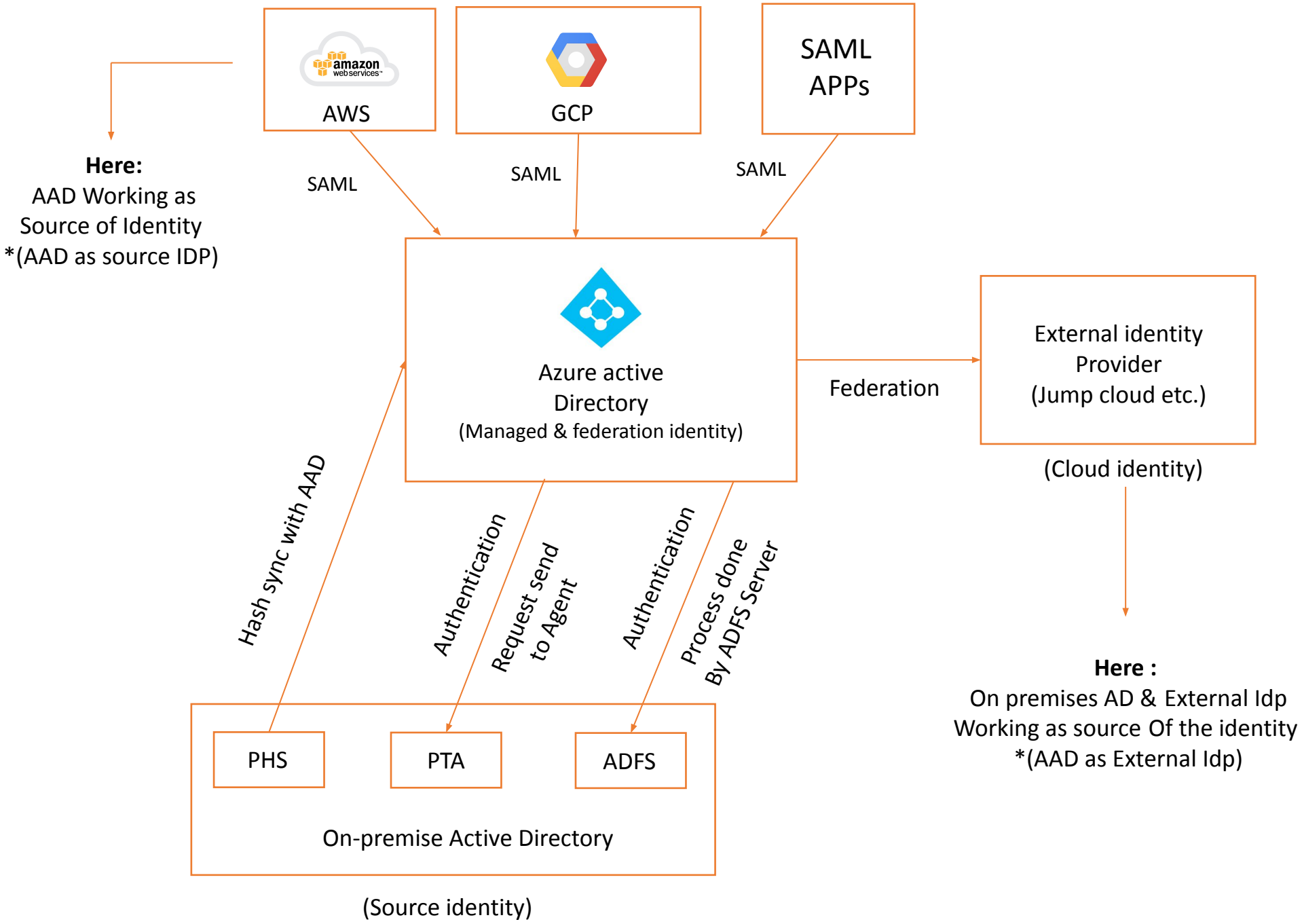
  - For example, you can assign roles to allow adding or changing users, resetting user passwords, managing user licenses, or managing domain names.

  - Types of AAD Roles :
    - Built-In Roles
      - Global Administrator - Can manage all aspects of Azure AD and Microsoft services that use Azure AD identities.
      - Application Administrator - Can create and manage all aspects of app registrations and enterprise apps.
      - Cloud Application Administrator - Can create and manage all aspects of app registrations and enterprise apps except App Proxy.
      - Global Readers - Can read everything that a Global Administrator can, but not update anything.
      - Directory Writers - Can read and write basic directory information. For granting access to applications, not intended for users.
      - Security Administrator - Can read security information and reports and manage configuration in Azure AD and Office 365.
    - Custom Roles


Azure AD Roles
{ Global Admin,
Company Admin,
Application Admin etc. }
Built in roles
Custom roles
Permission Applied
Permission Applied
AAD & O365
Administration
Manage AAD
Objects Only

- **Custom Domains**
  - Verified custom domain so user can use these domain while login in azure portal.
  - Customer can have more than one verified domains in aad.
  - Default domain provided by Microsoft : **CustomerID**.onmicrosoft.com

- **Integration with On-Premise AD**
  - Azure AD Connect - Azure Tool to sync on-premise AD information to Azure AD
    - PHS - [Password Hash Synchronization]
      - A hash of each password hash is being sent instead.
      - Two accounts are automatically created by Azure AD Connect:
        - MSOL_deeb213ff4bb in the Active Directory.
        - Sync_DCHostName_deeb213ff4bb in Azure AD.
    - PTA - [Pass Through Authentication]
      - Password hashes of Active Directory users do not transit over the network.
      - Pass through authentication agent is running on on-premise server
    - Seamless SSO [Single Sign On]
      - Azure Active Directory Seamless Single Sign-On (Azure AD Seamless SSO) automatically signs users in when they are on their corporate devices connected to your corporate network.
      - When enabled, users don't need to type in their passwords to sign in to Azure AD, and usually, even type in their usernames.
      -
  - Federation -
    - ADFS - [Active Directory Federation Service]
      - ADFS makes use of claims-based Access Control Authorization model to ensure security across applications using federated identity.
      - Claims-based authentication is a process in which a user is identified by a set of claims related to their identity. The claims are packaged into a secure token by the identity provider.
    - Federation with External Identity Provider [SAML]
      - Federation with external identity providers, Okta etc.


AWS
GCP
SAML APPs
Here:
AAD Working as
Source of Identity
*(AAD as source IDP)
SAML
SAML
SAML
Azure active
Directory
(Managed & federation identity)
Federation
External identity
Provider
(Jump cloud etc.)
(Cloud identity)
Hash sync with AAD
Authentication
Request send
to Agent
Authentication
Process done
By ADFS Server
PHS
PTA
ADFS
On-premise Active Directory
(Source identity)
Here :
On premises AD & External Idp
Working as source Of the identity
*(AAD as External Idp)

- Authentication using Azure Management Portal

Azure ARM (Azure Resource Manager) Portal URL:

```
https://portal.azure.com/
```

Azure ASM (Azure Service Manager) Portal URL:

```
https://manage.windowsazure.com/
```

Credential to access Azure Portal :

A. Username

B. Password

C. AccessToken (x-ms-RefreshTokenCredential)

- Authentication using Cross Platform Azure CLI

Login to azure using azure cli :

1. Interactive UI (Username + Password)

```
az login
az login --use-device-code
```

2. Service Principal (App ID + Password or Certificate)

```
az login --service-principal --username AppID --tenant TenantID --password ClientSecret
```

3. Managed Service Identity (VM Identity [System + User])

```
az login --identity
```

4. Access Token (Account ID + AccessToken)

```
Connect-AzAccount –AccessToken AccessToken
```

Logout to azure using azure cli :

```
az logout
```

Stored Secrets Location :

```
Folder - %USERPROFILE%\.azure (W) & $HOME/.Azure  (L)
File   - accessTokens.json
```

Operations :

Get the details about a default account

```
az account show
```

Get the details about a specified subscription

```
Az account show --subscription SubscriptionID
```

Get a list of active logged-in account

```
az account list
```

Set another logged-in account as a default account

```
az account set --subscription SubscriptionID
```

Get an access token to access azure api

```
Az account get-access-token --subscription SubscriptionID
```

Clear all active accounts from the CLI's local cache

```
az account clear
```

- Authentication using PowerShell Az Module

Login :

1. Interactive UI (Username + Password)

```
Connect-AzAccount
Connect-AzAccount -UseDeviceAuthentication
```

2. Service Principal (App ID + Password or Certificate)

```
$cred = Get-Credential [ Where, Username = Application ID & Password = Client Secret ]
Connect-AzAccount -ServicePrincipal -Tenant TentantID -Credential $cred
```

3. Managed Service Identity (VM Identity)

```
Connect-AzAccount -Identity
```

4. Access Token (Account ID + AccessToken)

```
Connect-AzAccount –AccessToken
```

Logout :

Logout to azure cloud :

```
Disconnect-AzAccount
```

Operations :

Get the currently logged in user context

```
Get-AzContext
```

List of all available contexts

```
Get-AzContext -ListAvailable
```

Change active subscription for a session

```
Set-AzContext -Subscription SubscriptionID
```

Select a subscription and account to have it persist between sessions

```
Select-AzContext -Name ContextName
```

List subscriptions

```
Get-AzSubscription
```

Select a specific subscription

```
Select-AzSubscription -SubscriptionID "SubscriptionID"
```

Export a context file

```
Save-AzContext -Path "C:\StolenToken.json"
```

Import a context file

```
Import-AzContext -Profile "C:\AzureAccessToken.json"
```

Cloud Secrets :

Stored Secrets Location :

`Folder / Directory -`

- Linux : `%USERPROFILE%\.azure\`
- Windows : `%USERPROFILE%\AppData\Local\.IdentityService\`

`Context File`

- Json File : `AzureRmContext.json`

`Token Cache File`

- Access Token : `TokenCache.dat` **OR** `masl.cache`

- Authentication using PowerShell Azure-AD Module

Login to azure using azure Azure-AD :

1. Interactive UI (Username + Password)

```
Connect-AzureAD
```

2. Service Principal (App ID + Password or Certificate) – Only certificate-based authentication is available

```
Connect-AzureAD -ApplicationId AppID  -TenantId TenantID -CertificateThumbprint CertThumID
```

3. Access Token

```
Connect-AzureAD -AadAccessToken
```

Logout to azure using azure Azure-AD :

```
Disconnet-AzureAD OR *Close the PowerShell Window for Disconnect
```

Stored Secrets Location :

```
*Secrets doesn't store on the hard disk. ( Only PowerShell Memory Cache )
```

Operations:

Get logged-in session information

```
Get-AzureADCurrentSessionInfo
```

- Authentication using PowerShell MsOnline Module

Login to azure using azure msonline :

1. Interactive UI (Username + Password)

```
Connect-MsolService
```

2. Service Principal (App ID + Password)

```
$cred = Get-Credential [ Where, Username = Application ID & Password = Client Secret ]
Connect-MsolService -Credential $Credential
```

3. Access Token (Azure AD / Microsoft Graph)

```
Connect-MsolService -AdGraphAccessToken AccessToken / -MsGraphAccessToken AccessToken
```

Logout to azure using azure cli :

```
*Close the PowerShell Window for Disconnect
```

Stored Secrets Location :

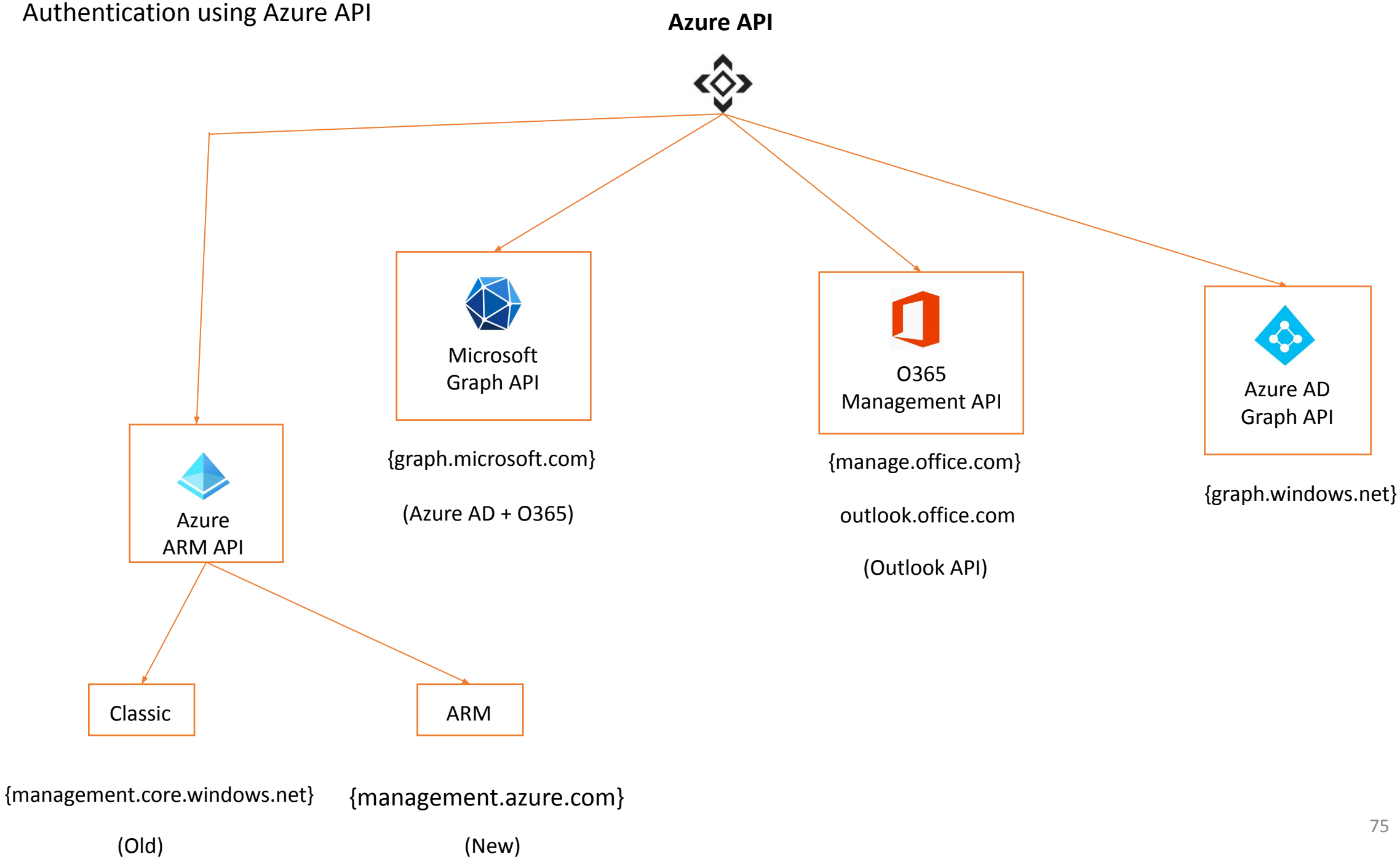
```
*Secrets doesn't store on the hard disk. ( Only PowerShell Memory Cache )
```

Operations:

List the Company Information

```
Get-MSolCompanyInformation
```

- Authentication using Azure API



Azure AD + Office 365 API :

Microsoft Graph API :

```
{HTTP method} https://graph.microsoft.com/{version}/{resource}?{query-parameters}
```

Azure AD Graph API :

```
{HTTP method} https://graph.windows.net/{version}/{resource}?{query-parameters}
```

O365 API : [management, outlook and other applications]

```
{HTTP method} https://*.office.com/{version}/{resource}?{query-parameters}
```

Azure Resources API :

ARM API :

```
{HTTP method} https://management.azure.com/{version}/{resource}?{query-parameters}
```

ASM API [Classic] :

```
{HTTP method} https://management.core.windows.net/{version}/{resource}?{query-parameters}
```

Authentication & Authorization to Graph API :

a. Register your app with Azure AD.
b. Get authorization Code. [Client ID, Redirect URI]
c. Get an access token. [Client ID, Client Secret, Authorization Code]
d. Call Azure API with the access token.
e. Use a refresh token to get a new access token

HTTP Request Header :

```
Authorization : Bearer AccessToken
```

Tools :

Microsoft Graph Explorer [https://developer.microsoft.com/graph/graph-explorer]
Postman

Azure AD Enumeration -

Check if target organization is using azure ad as a Idp

```
https://login.microsoftonline.com/getuserrealm.srf?login=Username@DomainName&xml=1
```

Azure AD valid user enumerations

```
o365creeper.py -f FileContainsEmail.txt
```

Password spray attack against Azure Ad users

```
Invoke-PasswordSprayEWS -ExchHostname outlook.office365.com -UserList FileContainsEmail.txt
-Password PasswordForSpray
```

Get currently logged-in session information

```
Get-AzureADCurrentSessionInfo
```

Get azure ad tenant information

```
Get-AzureADTenantDetail
```

Get a lists of domains in azure ad

```
Get-AzureADDomain
```

Get a list of all directory roles

```
Get-AzureADDirectoryRole
```

Get a list of members of a directory roles

```
Get-AzureADDirectoryRoleMember -ObjectId DirectoryObjectID
```

Get a lists of application owned by logged in user

```
az ad signed-in-user list-owned-objects
```

Get a lists of users in azure ad

```
Get-AzureADUser -All
```

Get a lists of groups in azure ad

```
Get-AzureADGroup -All
```

Get the owner of a group

```
Get-AzureADGroupOwner -ObjectId GroupObjectID
```

Get a lists of applications in azure ad

```
Get-AzureADApplication
```

Get the owner of an application

```
Get-AzureADApplicationOwner -ObjectId AppObjectID
```

Get a lists of service principal in azure ad

```
Get-AzureADServicePrincipal
```

Get the owner of a service principal

```
Get-AzureADServicePrincipalOwner -ObjectId ServicePrincipalObjectID
```

Get azure ad role membership of a service principal

```
Get-AzureADServicePrincipalMembership -ObjectId ServicePrincipalObjectID
```

Get service principal delegation api permission with user or admin consent

```
Get-AzureADServicePrincipalOAuth2PermissionGrant -ObjectId ServicePrincipalObjectID
```

Get service principal application api permission with admin consent only

```
Get-AzureADServiceAppRoleAssignedTo -ObjectId ServicePrincipalObjectID
```

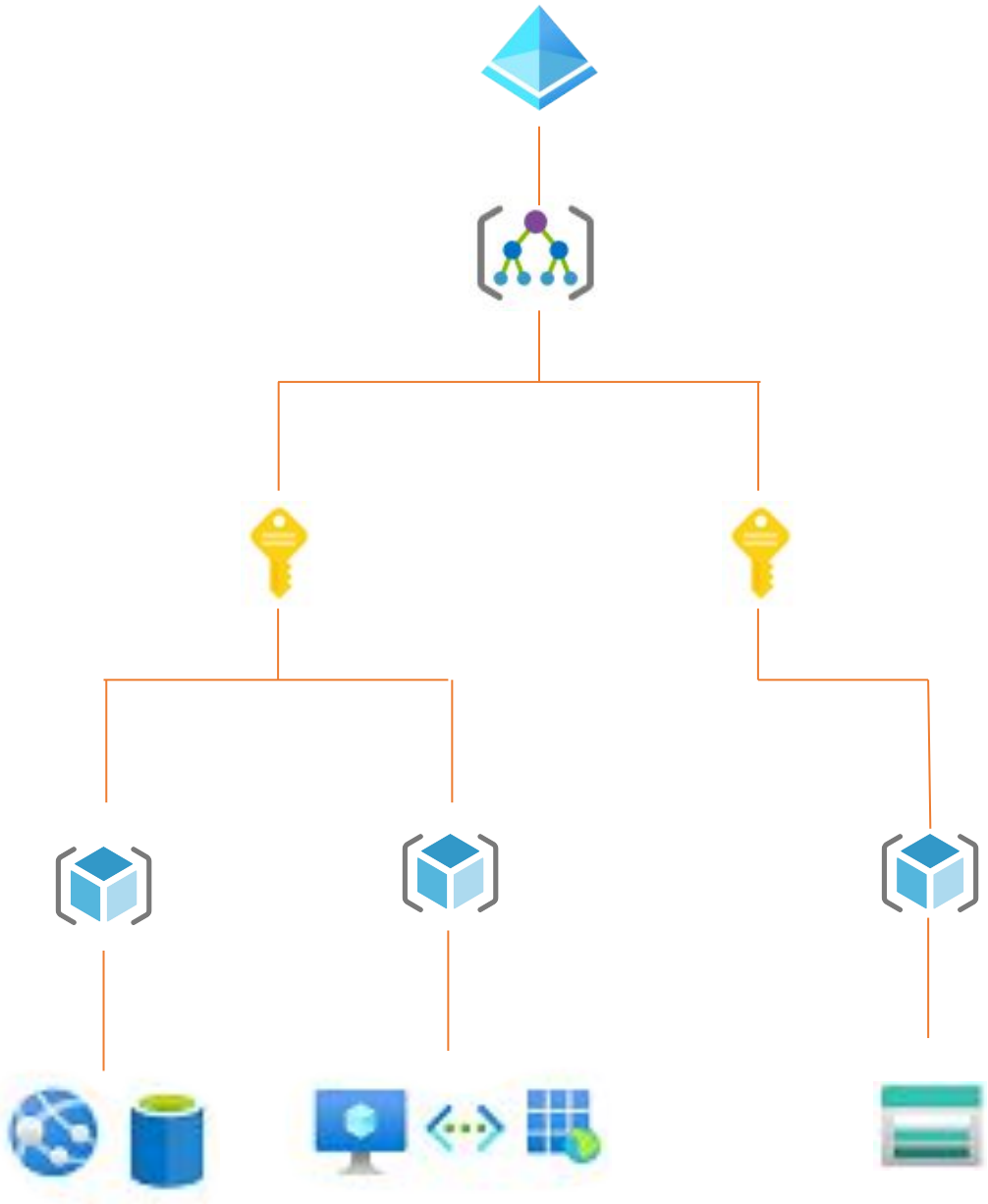
Retrieves the object(s) specified by the objectIds

```
Get-AzureADObjectByObjectId -ObjectIds ObjectID
```

3.4 Azure Resource Manager [ARM]

- Azure Resource Manager (ARM) is the native platform for infrastructure as code (IaC) in Azure.
- It enables us to centralize the management, deployment, and security of Azure resources.
- It provides Infrastructure as a Service [IaaS], Platform as a Service [PaaS] and Software as a Service [SaaS].
- Azure ARM manage access control by “Role Based Access Control [RBAC]”.

Enterprise Global Account



AAD Tenant

Management Group

Subscription

Resource Group

Resource

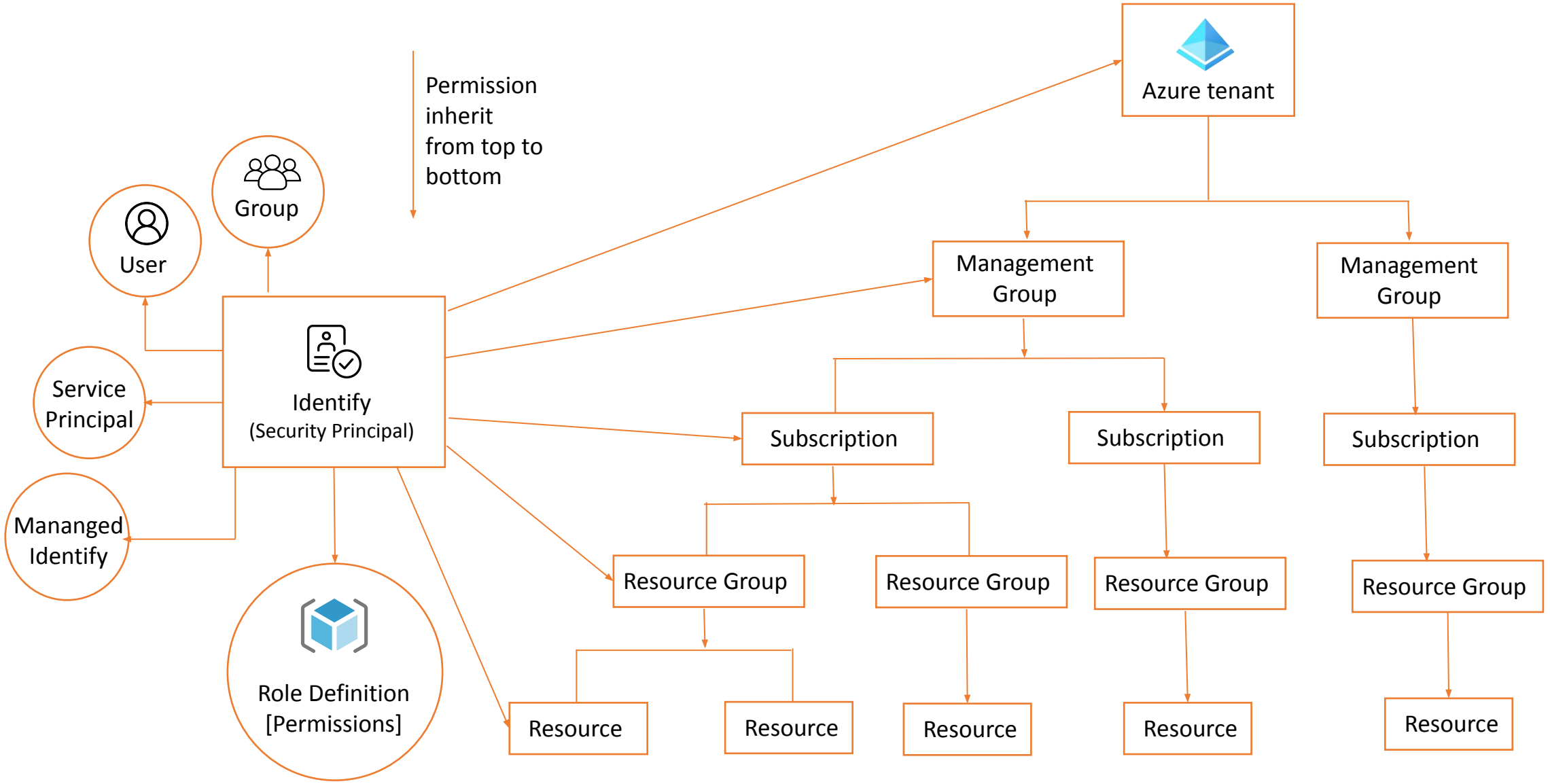
Azure Cloud Building Block :

- **Enterprise**
  - This represents the Azure global account. It’s the unique identity that the business owns and allows access to subscriptions, tenants, and services.
- **Tenant**
  - Tenants are instances of Azure for the Enterprise. An Enterprise can have multiple tenants.
  - Access to one tenant in an enterprise does not give access to another tenant. An analogy is that tenants are similar to Forests in Active Directory.
- **Management Groups**
  - Azure management groups provide a way for an organization to control and manage access, compliance, and policies for their subscription within their tenant.
- **Subscriptions**
  - Subscriptions are how you gain access to Azure services (Azure itself, Azure AD, Storage, etc). Subscriptions are often broken out into uses for the businesses, e.g. a subscription for production web apps, another subscription for development web apps, etc.
- **Resource Groups**
  - Resource groups are the containers that house the resources.
- **Resources**
  - Resources are the specific application, such as SQL servers, SQL DBs, virtual networks, run-books, accounts, etc.

Role Based Access Control (RBAC)

- Azure RBAC is an authorization system built on Azure Resource Manager (ARM) that provides fine-grained access management of Azure resources.

- Role Based Access Control [RBAC] Components -
  - Security principal
  - Scope
  - Roles Definition
  - Role Assignment


Permission
inherit
from top to
bottom
Azure tenant
Group
User
Management
Group
Management
Group
Service
Principal
Identify
(Security Principal)
Subscription
Subscription
Subscription
Mananged
Identify
Resource Group
Resource Group
Resource Group
Resource Group
Role Definition
[Permissions]
Resource
Resource
Resource
Resource
Resource

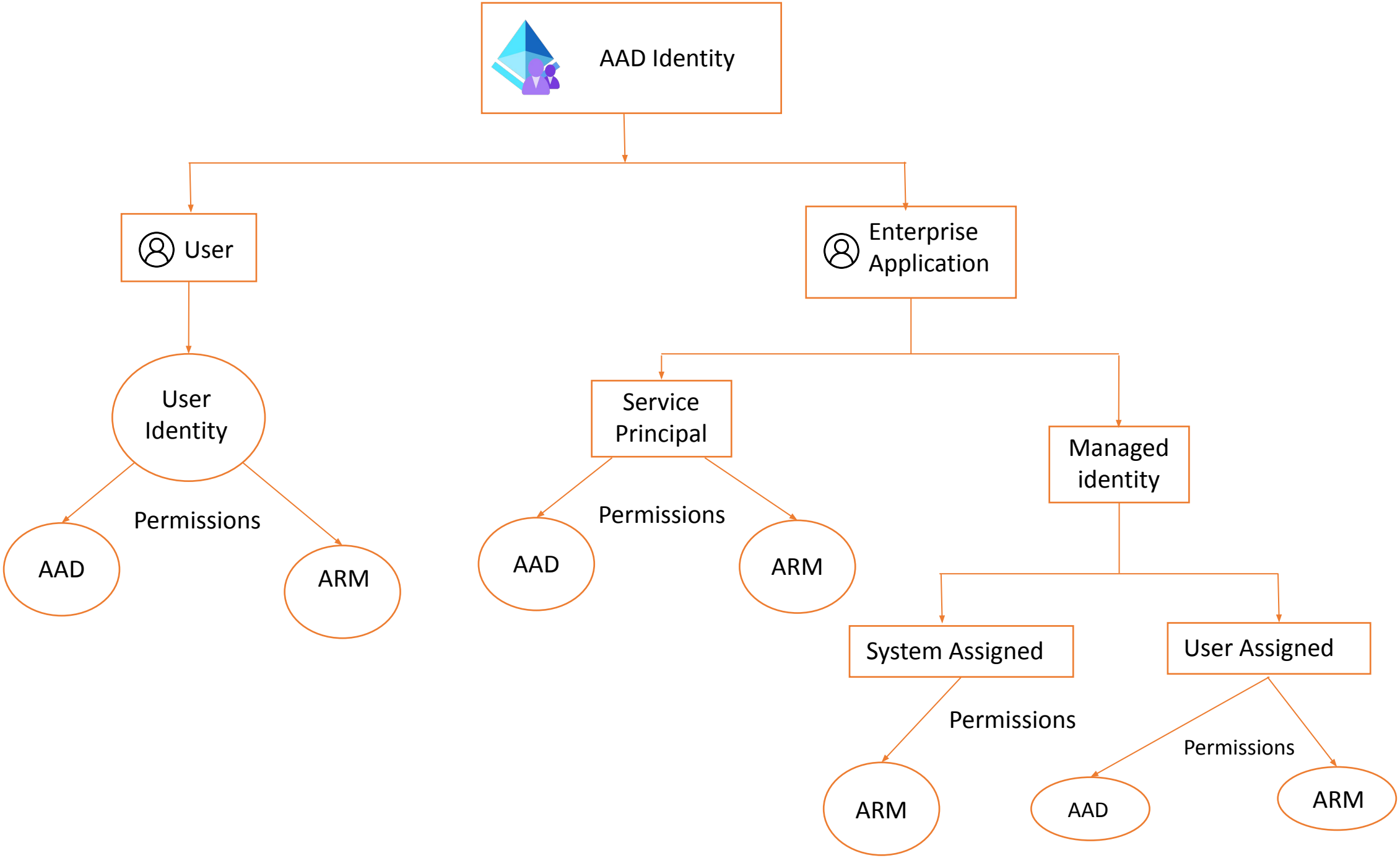
Security Principal -

- A security principal is an object that represents a user, group, service principal, or managed identity that is requesting access to Azure resources. You can assign a role to any of these security principals.

  - User Identity
    - Identity for a users
    - User Identity can have permission on both azure ad and azure resources.

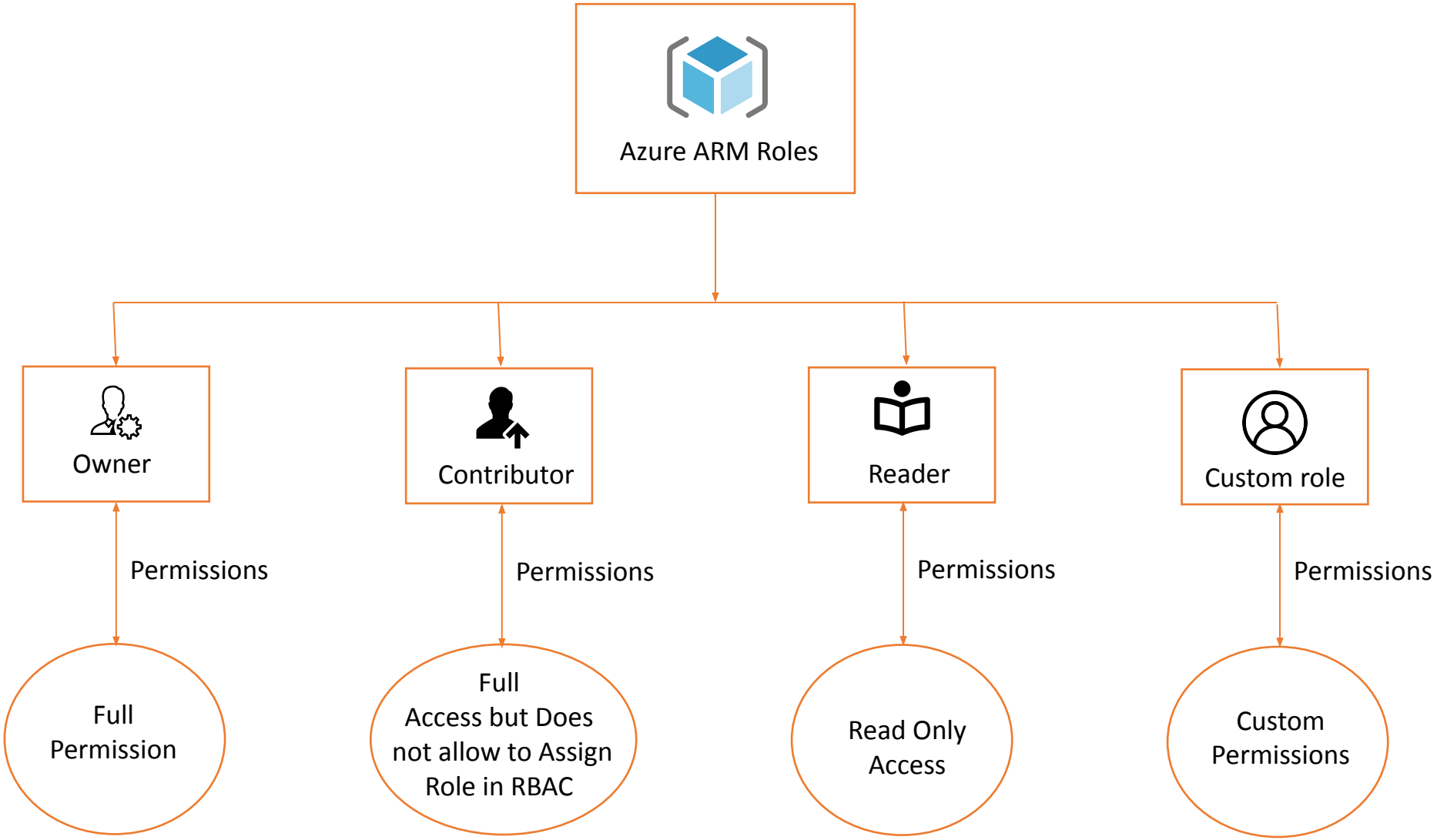
  - Service Principal Identity
    - Identity for azure applications / automation account
    - Service principal Identity can have permission on both azure ad and azure resources.

  - Managed Identity –
    - Identity only attached to an azure resources
    - Type of Managed Identity
      - System-assigned managed identity
      - User-assigned managed identity
    - Managed Identity can only have permission on azure resources not azure ad.


AAD Identity
User
Enterprise Application
User Identity
Service Principal
Managed identity
Permissions
AAD
ARM
Permissions
AAD
ARM
System Assigned
User Assigned
Permissions
ARM
Permissions
AAD
ARM

Role Definition -

- A role definition is a collection of permissions. It's typically just called a role. A role definition lists the operations that can be performed, such as read, write, and delete. Roles can be high-level, like owner, or specific, like virtual machine reader.
  - Owner
  - Contributor
  - Reader
  - Other Built-in Roles
  - Custom Roles


Azure ARM Roles
Owner
Contributor
Reader
Custom role
Permissions
Permissions
Permissions
Permissions
Full Permission
Full Access but Does not allow to Assign Role in RBAC
Read Only Access
Custom Permissions

Scope -

- Scope is the set of resources that the access applies to. When you assign a role, you can further limit the actions allowed by defining a scope.
  - Management Group Level
  - Subscription
  - Resource Group
  - Individual Resource

Role assignments

- A role assignment is the process of attaching a role definition to a user, group, service principal, or managed identity at a particular scope for the purpose of granting access.
- Access is granted by creating a role assignment, and access is revoked by removing a role assignment.

1. Security Principal

Marketing group

"Actions": [
"*"
],
"NotActions": [
"Auth/*/Delete",
"Auth/*/Delete",
"Auth/*/elevate...

Contributor

Pharma-sales
Resource group

Role Assignment

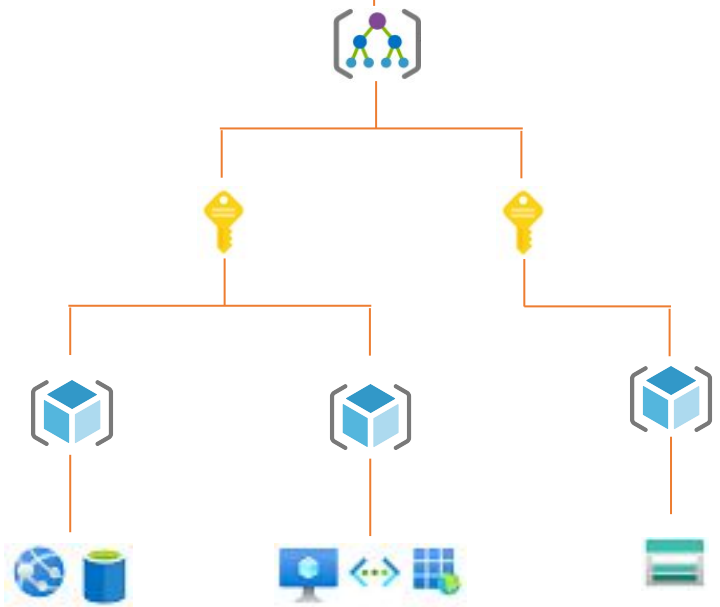
Owner
Contributor
Reader
Backup Operator
Security reader
User access Administrator
Virtual machine contributor

Built-in Role

Reader support tickets
Virtual machine operator

Custom Role

2. Role Definition



3. Scope

RBAC Role V/s Azure AD Role

- RBAC Role -
  - RBAC roles, allows administrator to define and restrict the fine-grained permissions on azure resources. So, Security principal can manage the resources on azure.
  - Azure roles control access to Azure resources such as virtual machines or storage using Azure Resource Management

- Azure AD Role -
  - AAD roles, allow administrator to define and restrict the fine-grained permissions on azure ad. So, Security principal can manage authentication and authorization on azure ad.
  - Azure AD roles control access to Azure AD resources such as users, groups, and applications using Graph API

Azure ARM Enumeration -

Get details about currently logged in session

```
az account show
```

Get a lists of role assigned to an identity [user, service principal, identity] in current subscription and inherited to all it's resource or group

```
az role assignment list --assignee ObjectID/Sign-InEmail/ServicePrincipal --all
```

Get the list of all available subscriptions

```
az account list --all
```

Get the details of a subscription

```
az account show -s Subscription-ID/Name
```

Get the list of available resource group in current subscription

```
az group list -s Subscription-ID/Name
```

Get the list of available resource group in a specified subscription

```
az group list -s Subscription-ID/Name
```

Get the list of available resources in a current subscription

```
az resource list
```

Get the list of available resources in a specified resource group

```
az resource list --resource-group ResourceGroupName
```

Lists of roles assigned in current subscription [Role Assignment]

```
az role assignment list
```

Lists of roles assigned in current subscription and inherited to all it's resource or group [Role Assignment]

```
az role assignment list -all
```

Lists of roles assigned in specified subscription [Role Assignment]

```
az role assignment list --subscription Subscription-ID/Name
```

Lists of roles with assigned permission [Role Definition - For Inbuilt and Custom Role]

```
az role definition list
```

Lists of custom role with assigned permissions

```
az role definition list --custom-role-only
```

Get the full information about a specified role

```
az role definition list  -n RoleName
```

3.5 Office 365 / Microsoft 365

Office 356 [O365]:

- Office 365 is a cloud-based suite of productivity apps.
- Office 365 is a line of subscription services offered by Microsoft.
  - Personal
  - Business
- Lists of enterprise app includes in office 365
  - Microsoft Exchange Online
  - Microsoft SharePoint Online
  - Office for the web: https://outlook.office365.com
  - Microsoft Skype for Business Online
  - Microsoft OneDrive
  - Microsoft Team : https://teams.microsoft.com/
  - Microsoft Intune : https://endpoint.microsoft.com/

Office 365 vs Microsoft 365 :

- Office 365 is a cloud-based suite of productivity apps, while Microsoft 365 is a package of services which includes Office 365, alongside other business tools

Office 365:

- Microsoft Exchange Online
- Microsoft SharePoint Online
- Office for the web
- Microsoft Skype for Business Online
- One Drive
- Microsoft Intune

Microsoft 365:

- O365
- Window 10 Enterprise License
- Cloud Based Security & Device Management

Management Portal Access :

- Web Portal :

  O365 / M365 Admin Center : https://admin.microsoft.com, https://portal.microsoft.com

- API :

  Microsoft Graph API :

  ```
  {HTTP method} https://graph.microsoft.com/{version}/{resource}?{query-parameters}
  ```

  O365 API : [management, outlook and other applications]

  ```
  {HTTP method} https://*.office.com/{version}/{resource}?{query-parameters}
  ```

Identity & Access Management

- Only O365 / M 365 Admin can access the "**Admin Center**" Portal & API.
- User can access the "**O365 / M365 Admin Center**" resources based on Role Associated with them.

Office 365 Admin Roles

- Office 365 roles are subset of Azure AD roles.
- Lists of Office 365 Administrator -
  - Global Administrator
  - Global Reader
  - Exchange Administrator
  - SharePoint Administrator
  - Dynamics 365 Administrator
  - Teams Administrator
  - User Administrator
  - Application Administrator
  - Helpdesk Administrator
  - Service support Administrator

User Portal Access

- Portal :

  User Access : https://portal.office.com

  SSO Portal : https://myapps.microsoft.com

- API :

  Microsoft Graph API :

  ```
  {HTTP method} https://graph.microsoft.com/{version}/{resource}?{query-parameters}
  ```

  O365 API : [management, outlook and other applications]

  ```
  {HTTP method} https://*.office.com/{version}/{resource}?{query-parameters}
  ```

Business Application

- Outlook
- Skype
- OneDrive
- SharePoint
- Team
- Calendar
- Other Apps

User Profile

- MyAccount portal is used to manage and retrieve information about logged in user.
- User's personal information e.g., organization name whom user belongs to, registered devices, assigned license, contact information etc.

User's Personal Profile : https://myaccount.microsoft.com/
User's Office Profile : https://apc.delve.office.com/

Office 365 Enumeration -

Check if target organization is using azure ad as a Idp

```
https://login.microsoftonline.com/getuserrealm.srf?login=Username@DomainName&xml=1
```

Check if target organization is using O365's outlook service [Exchange Online]

```
Organization DNS Record : MX - *.mail.protection.outlook.com
```

Get the information about the company

```
Get-MsolCompanyInformation
```

Get the information about services available in the current license

```
Get-MsolAccountSku | Select -ExpandProperty ServiceStatus
```

Get the information about all available license for an organization

```
Get-MsolAccountSku
```

Get a lists of domains in azure ad

```
Get-MsolDomain
```

Get a lists of users in azure ad

```
Get-MsolUser -All
```

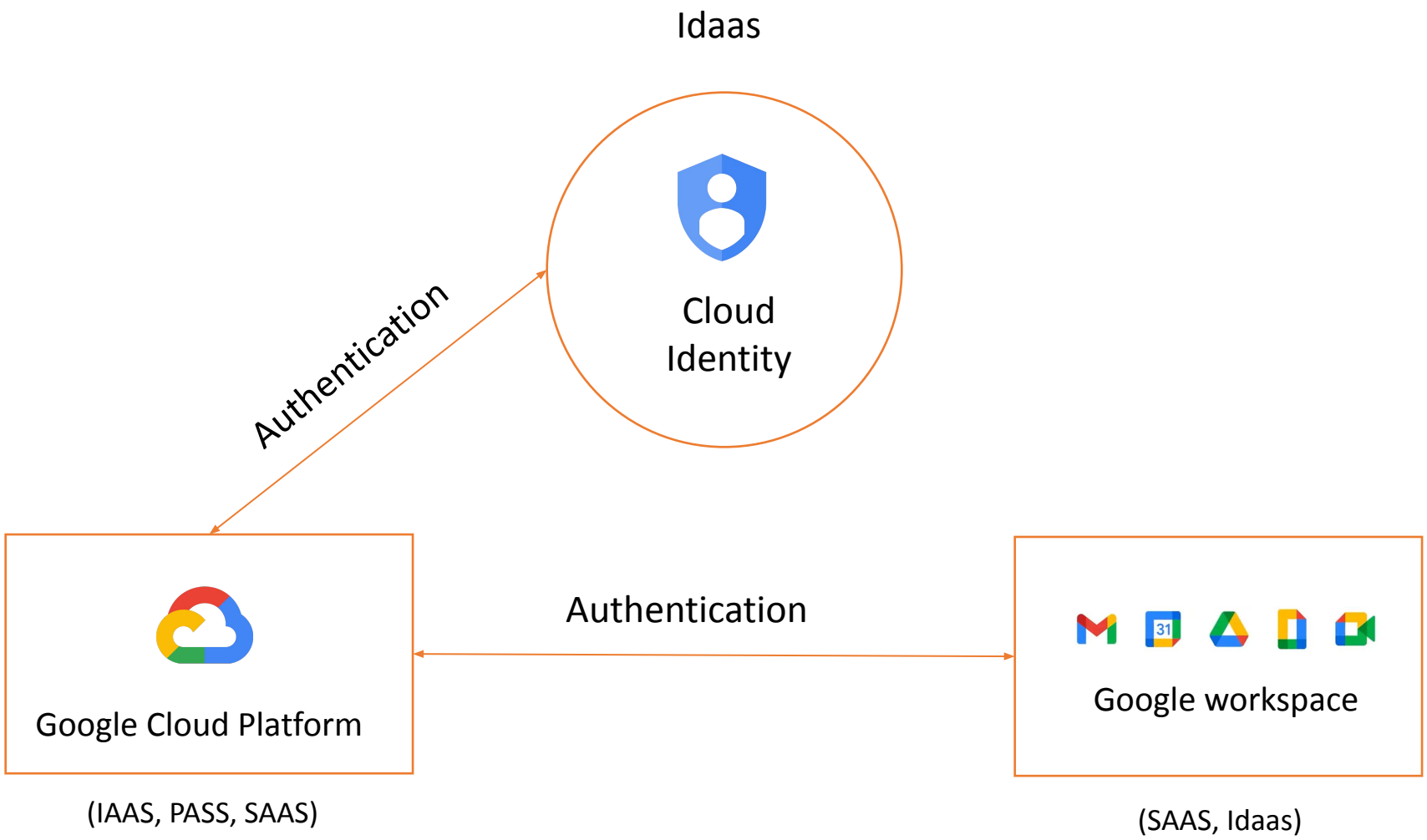
Get an Administrative roles assigned to a user in azure ad

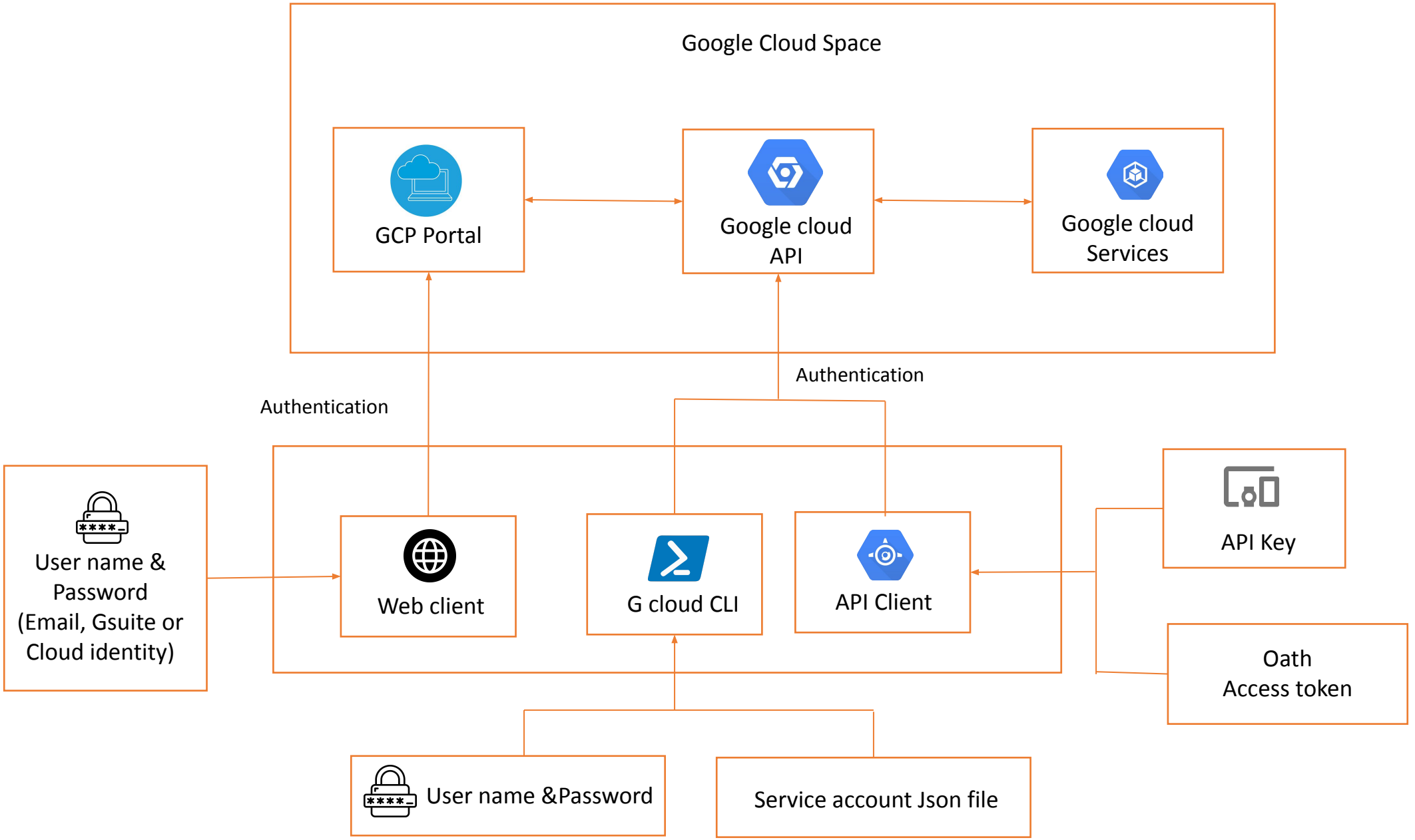
```
Get-MsolUserRole -UserPrincipalName UserEmailAddress
```

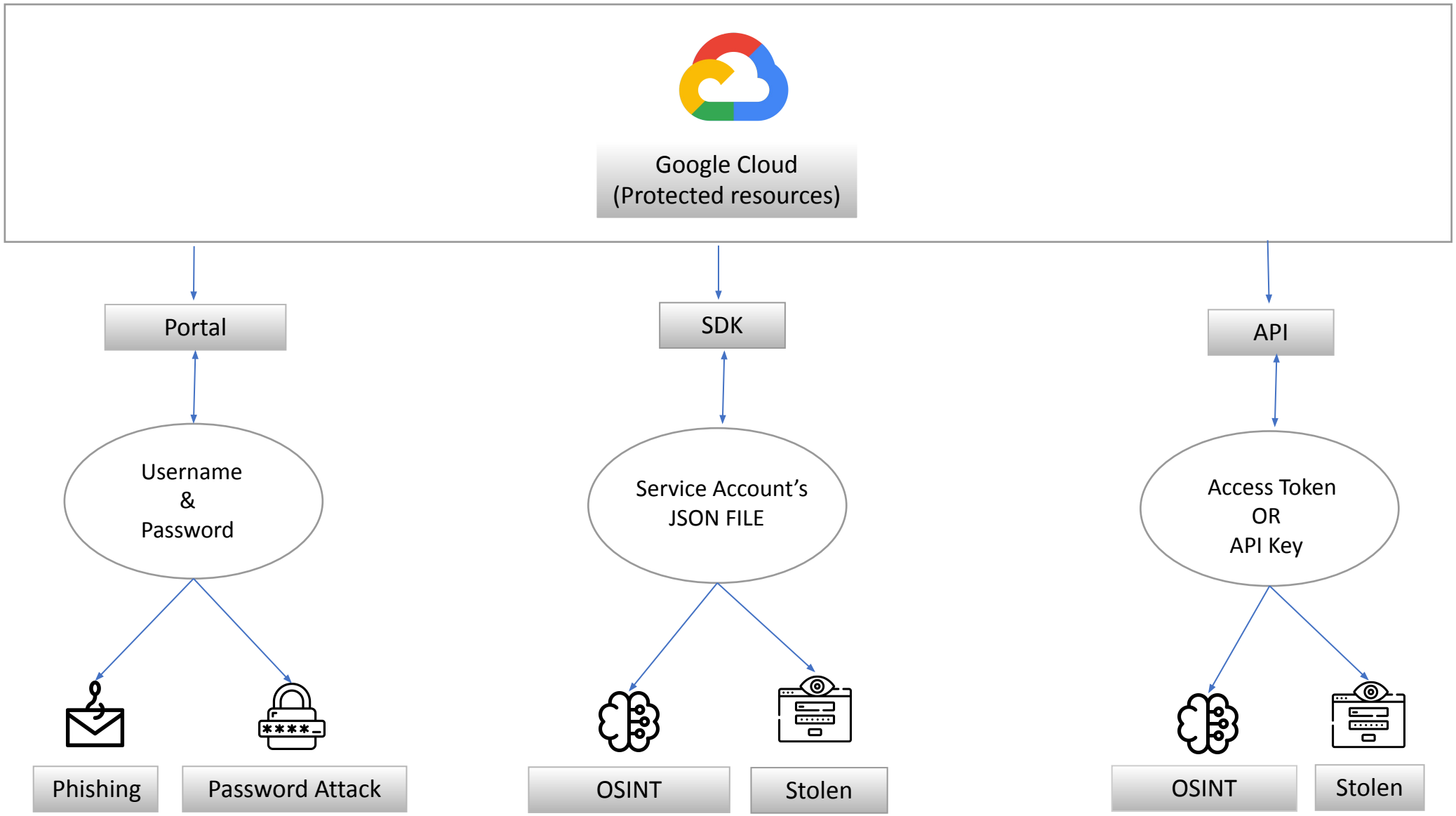
4.1 Google Cloud Overview

Three Main Components of Google Cloud -

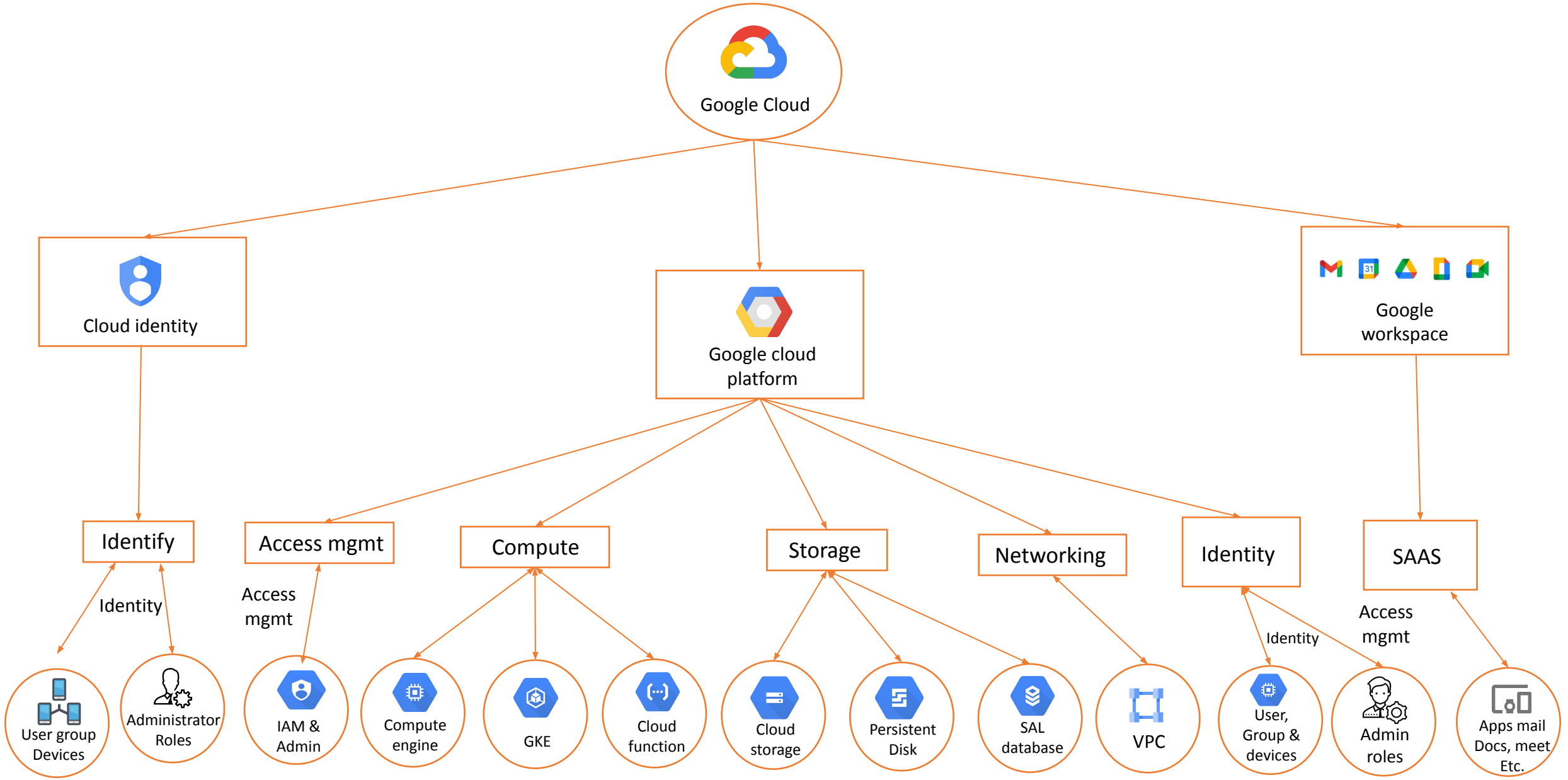
- **Cloud Identity**
  - Cloud Identity is an Identity as a Service (IDaaS) solution that centrally manages users, groups and devices.
  - We can configure Cloud Identity to federated identities between Google and other identity providers, such as Active Directory and Azure Active Directory.
  - Cloud Identity also gives you more control over the accounts that are used in your organization.
  - Cloud identity allow administrator to crate Cloud Identity account for each of users and groups in a organization.
  - We can then use Identity and Access Management (IAM) to manage access to Google Cloud resources for each Cloud Identity account.
- **Google Workspace [G-suite]**
  - Google Workspace (formerly G Suite) secure collaboration and productivity apps for businesses. Includes Gmail, Drive, Meet and more.
  - Google Workspace have integrated identity as a service in it.
  - We can use google workspace as identity source for google cloud platform.
- **Google Cloud Platform [GCP]**
  - Google Cloud Platform is a suite of public cloud computing services offered by Google.
  - The platform includes a range of hosted services for compute, storage and application
  - We can use cloud identity, google workspace or external identity as source of identity for GCP.


Idaas
Cloud
Identity
Authentication
Authentication
Google Cloud Platform
(IAAS, PASS, SAAS)
Google workspace
(SAAS, Idaas)


Google Cloud Space
GCP Portal
Google cloud API
Google cloud Services
Authentication
Authentication
User name & Password (Email, Gsuite or Cloud identity)
Web client
G cloud CLI
API Client
API Key
Oath Access token
User name &Password
Service account Json file


Google Cloud
(Protected resources)
Portal
SDK
API
Username
&
Password
Service Account's
JSON FILE
Access Token
OR
API Key
Phishing
Password Attack
OSINT
Stolen
OSINT
Stolen

4.2 Google Cloud Services



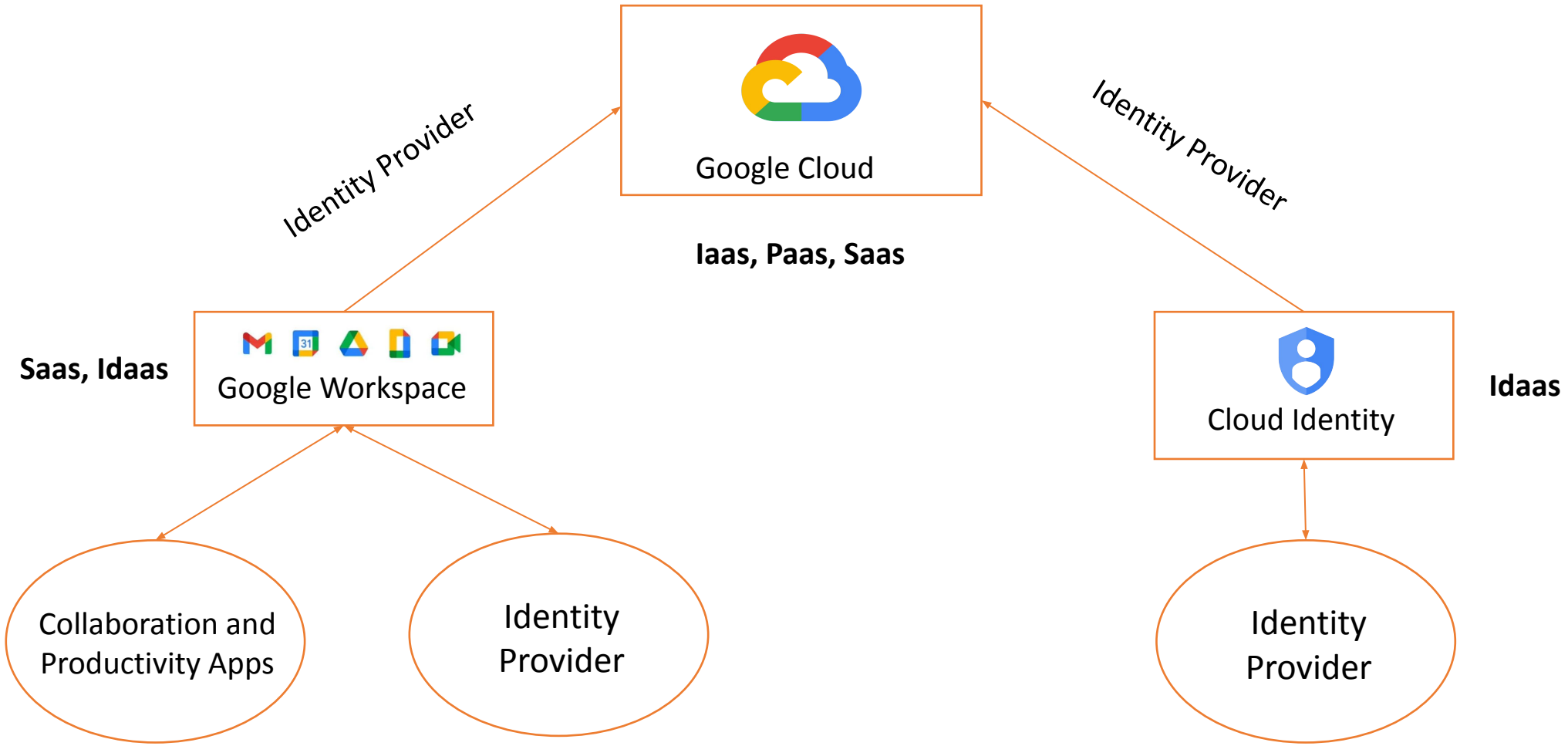
4.3 Cloud Identity & Google Workspace

Cloud Identity :

- Identity Provider
  - Cloud Identity is an Identity as a Service (IDaaS) solution that centrally manages users and groups.
  - You can configure Cloud Identity to federated identities between Google and other identity providers, such as Active Directory and Azure Active Directory.
  - Cloud Identity API : https://cloudidentity.googleapis.com ----- Organization Admin [Gcloud Role]

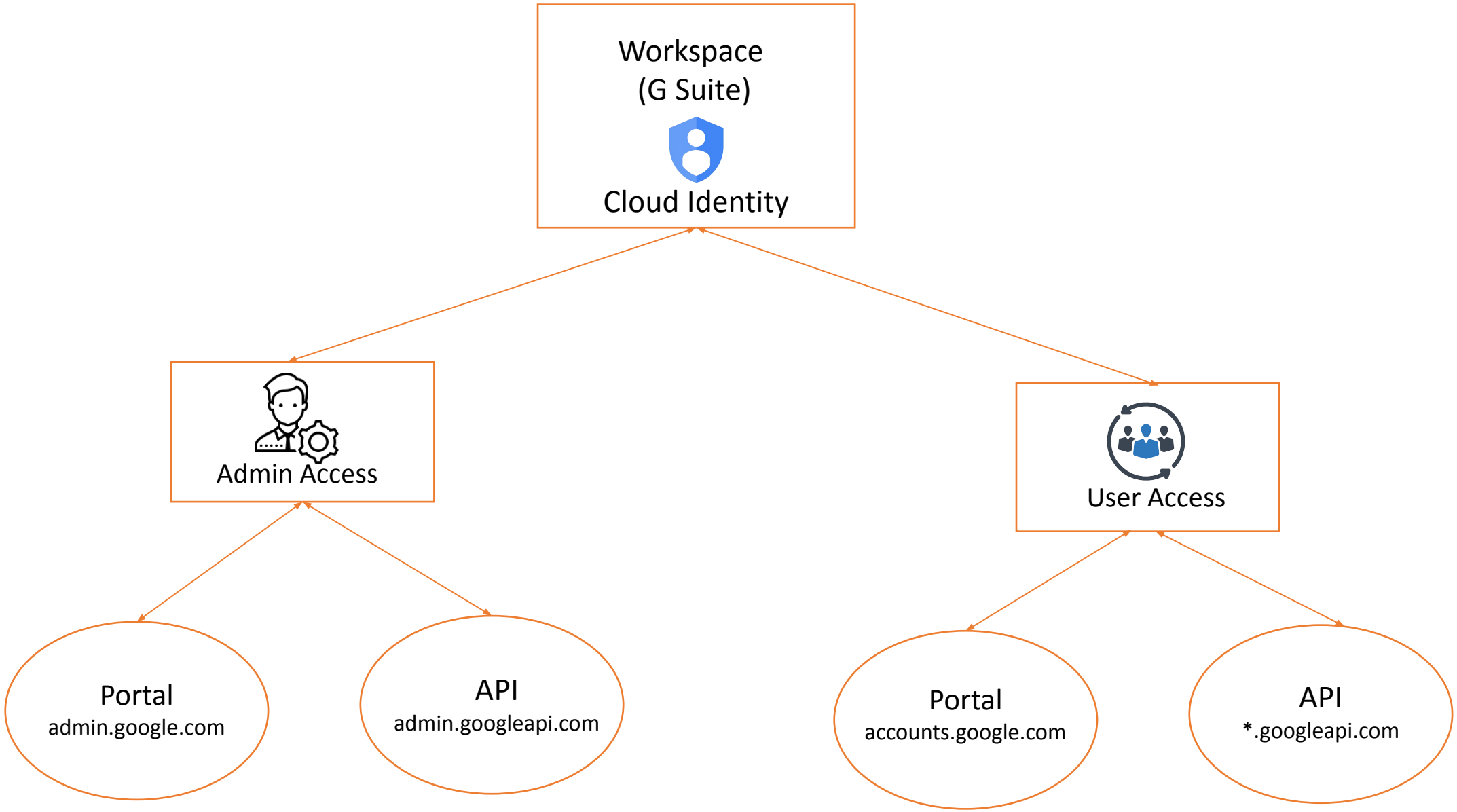
Google Workspace [Formerly known as G Suite] :

- Identity Provider
  - Google Workspace have inbuilt Idaas solution for accessing SAAS Applications and GCP Resource.
- Collaboration SAAS Application
  - Google Workspace plans provide a custom email for your business and includes collaboration tools like Gmail, Calendar, Meet, Chat, Drive, Docs, Sheets, Slides, Forms, Sites, and more.
  - Google Workspace API : https://www.googleapis.com/
    - Mail API : https://mail.googleapis.com/*
    - Drive API : https://drive.googleapis.com/*
    - Calendar API : https://calendar.googleapis.com/*


Google Cloud
Identity Provider
Identity Provider
Iaas, Paas, Saas
Saas, Idaas
Google Workspace
Cloud Identity
Idaas
Collaboration and Productivity Apps
Identity Provider
Identity Provider

Google Workspace V/S Cloud Identity

- Admin Portal [https://admin.portal.com] - Common for cloud identity & google workspace
- Admin API [https://admin.googleapis.com/*] - Common for cloud identity & google workspace
- Default Identity provider for google cloud is "**cloud identity**".
- Cloud Identity subscription only provide identity services and it's free for all users.
- Google workspace subscription provides identity service along with collaboration tool and it's paid per users.


Workspace
(G Suite)
Cloud Identity
Admin Access
User Access
Portal
admin.google.com
API
admin.googleapi.com
Portal
accounts.google.com
API
*.googleapi.com

Admin Portal Access

- **Domain :** Custom domain linked to cloud identity or google workspace for an organization.

- **Directory :**
  - Users
  - Groups
  - Organizational Units

- **Devices :**
  - End Point Devices
  - Network Devices

- **Apps :**
  - G-Suite Application
  - External Application

- **Roles :** Google workspace roles, which allows member to manage access control in google workspace / cloud identity for an organization.
  - Predefined Roles - Super Admin, Groups Admin, User Management Admin, Help Desk Admin, Services Admin, Mobile Admin etc.
  - Custom Roles

- **Console & API Access URL : [Only Cloud Identity Admin / Google Workspace Can Access]**
  - Admin Console Access URL : https://admin.google.com
  - Admin API Access URL : https://admin.googleapis.com/*

- **Integration (Federation) / External Identity :** Import external identity to google cloud identity.

User Portal Access

- Google Workspace (G-Suite) App Services
    - Gmail
    - Drive
    - Calendar
- Console & API Access URL
    - Console Access : https://accounts.google.com
    - API Access : https://**Service**.googleapis.com/*

Domain Wide Delegation

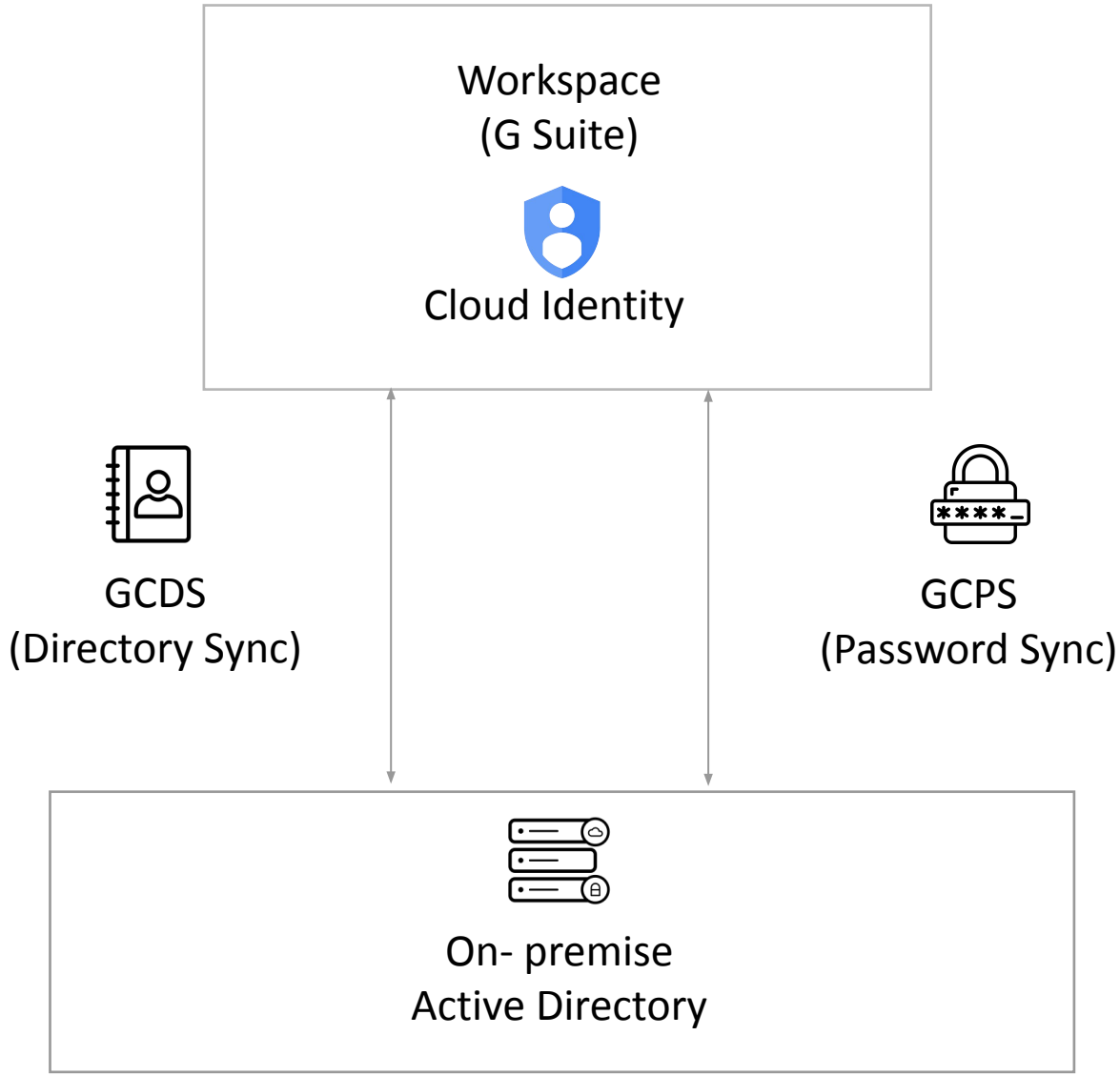
- Domain-wide delegation is a powerful feature that allows apps to access users' data across your organization's Google Workspace environment.
- Domain wide delegation can only be enabled for service account.
- Service account can access all user's data in google workspace (gsuite) .
- Domain wide delegation should be enabled bi-directional [Google Cloud and Google Workspace].

Domain Wide Delegation

On-premise to Google Cloud Identity Sync -

- Google Cloud Directory Sync (GCDS)
  - Google Cloud Directory Sync enables administrators to synchronize users, groups and other data from an Active Directory/LDAP service to their Google Cloud domain directory.
  - It doesn’t sync password from on-premise to google cloud.

- G Suite Password Sync (GSPS)
  - G Suite Password Sync (GSPS) automatically keeps your users' Google Workspace and Cloud Identity passwords in sync with their Microsoft Active Directory passwords. Whenever a user's Active Directory password is changed, GSPS immediately pushes the change to their managed Google Account.
  - GSPS never changes Active Directory passwords; it only syncs Active Directory password changes to your organization's Google Account.
  - G Suite Password Sync (GSPS) is available to Google Workspace and Cloud Identity administrators.


Workspace
(G Suite)
Cloud Identity
GCDS
(Directory Sync)
GCPS
(Password Sync)
On- premise
Active Directory

Download, Install and configure the Google Administrator Management Tool [GAM] :

```
Github Link :  https://github.com/jay0lee/GAM
```

Currently logged in user information :

```
gam info user
```

Organization custom domain information :

```
gam info domain
```

Get information about Configured Oauth Access Token's Scope :

```
gam oauth info
```

Lists of users in an organization :

```
gam print users
```

Get the information about a specified user :

```
gam info user UserName
```

Lists of groups in an organization :

```
gam print groups
```

Get the information about a specified group :

```
gam info group GroupName
```

Lists of roles in an organization

```
gam print roles
```

Lists of cloud identity admin / Google workspace admin in an organization :

```
gam print admins
```

Lists of cloud identity / google workspace licences :

```
gam print licences
```

Organization custom domain information :

```
gam info domain
```

4.4 Google Cloud Platform

Google Cloud Platform (GCP), offered by Google, is a suite of cloud computing services that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search, Gmail, file storage, and YouTube.

Regions -

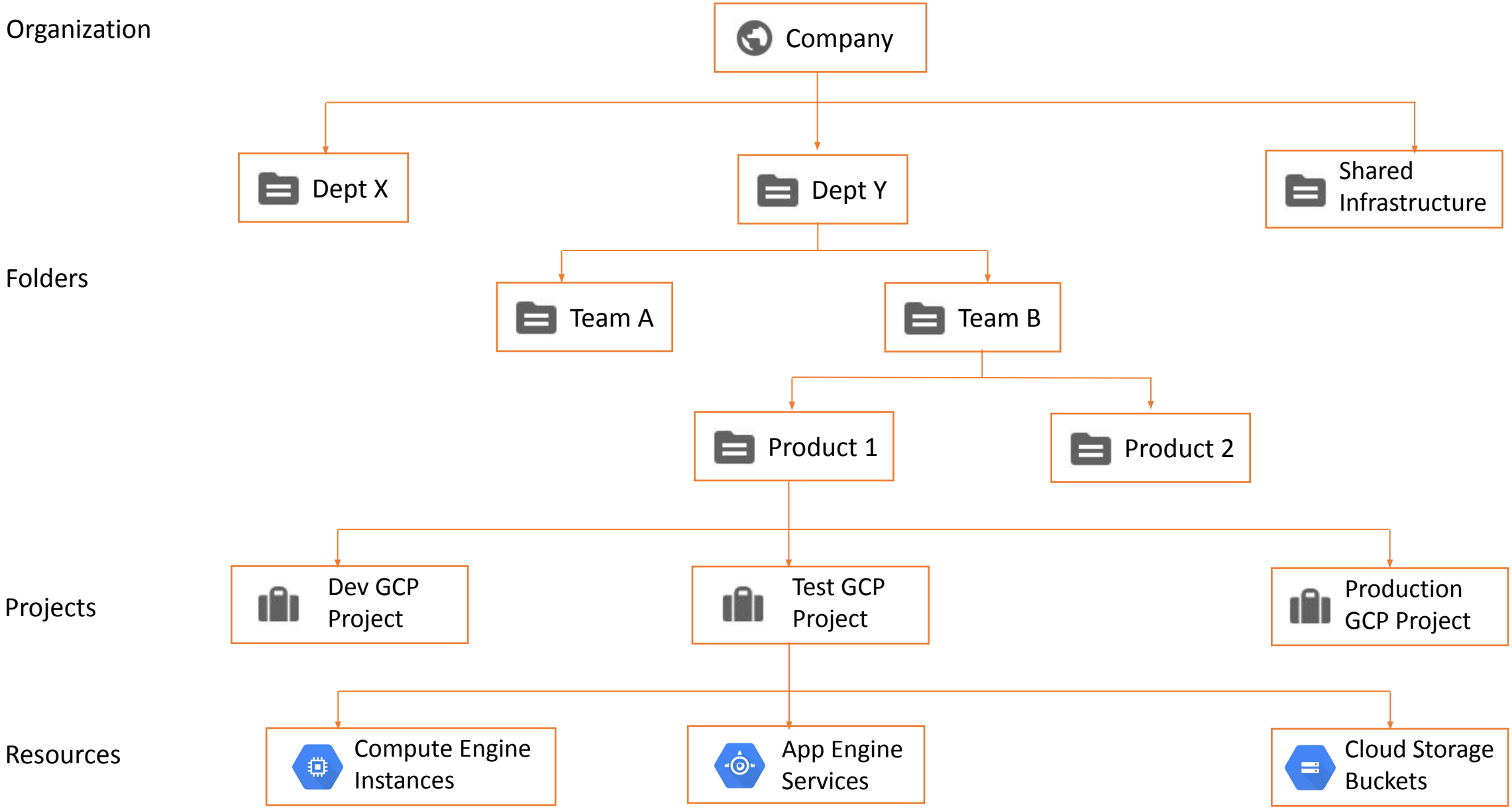
- Regions are independent geographic areas that consist of zones. Means Regions are collections of zones.
- There are around 24 regions in of google cloud.

Zones -

- A zone is a deployment area for Google Cloud resources within a region. Zones should be considered a single failure domain within a region
- There are around 73 zones within 24 regions in google cloud.

API -

- They are a key part of Google Cloud Platform, allowing us to easily manage everything from computing to networking to storage to machine-learning-based data analysis to our applications with programmatic access.


Organization
Company
Dept X
Dept Y
Shared Infrastructure
Folders
Team A
Team B
Product 1
Product 2
Projects
Dev GCP Project
Test GCP Project
Production GCP Project
Resources
Compute Engine Instances
App Engine Services
Cloud Storage Buckets

Resource Manager -

- Resource manager help manage resource containers such as organizations, folders, and projects that allow you to group and hierarchically organize other GCP resources

Resource Container -

- Resource container contains other gcp resources.

Organization

- Organization resource is the root node in the Google Cloud resource hierarchy and is the hierarchical super node and ancestor of project resources and folders.
- Organization administrators have central control of all resources and can view and manage all of the company's projects
- IAM access control policies applied to the Organization resource apply throughout the hierarchy on all resources in the organization.

Folders -

- Folders are an additional optional grouping mechanism on top of projects and provide isolation boundaries between projects.
- Folders can be used to model different legal entities, departments, teams, and environments within a company

Projects

- Projects are a core organizational component of GCP
- A project is required to use Google Cloud and forms the basis for creating, enabling, and using all Google Cloud services, managing APIs, enabling billing, adding and removing collaborators, and managing permissions.
- Each project has a name and a unique project ID across Google Cloud

Resources

- GCP provides resource like compute, networking, storage & access management.

Cloud IAM [Identity & Access Management]

- Identity and Access Management (IAM) lets administrators authorize who can take action on specific resources, giving you full control and visibility to manage Google Cloud resources centrally.

- In IAM follows Resource based policy instead of Identity based policy.

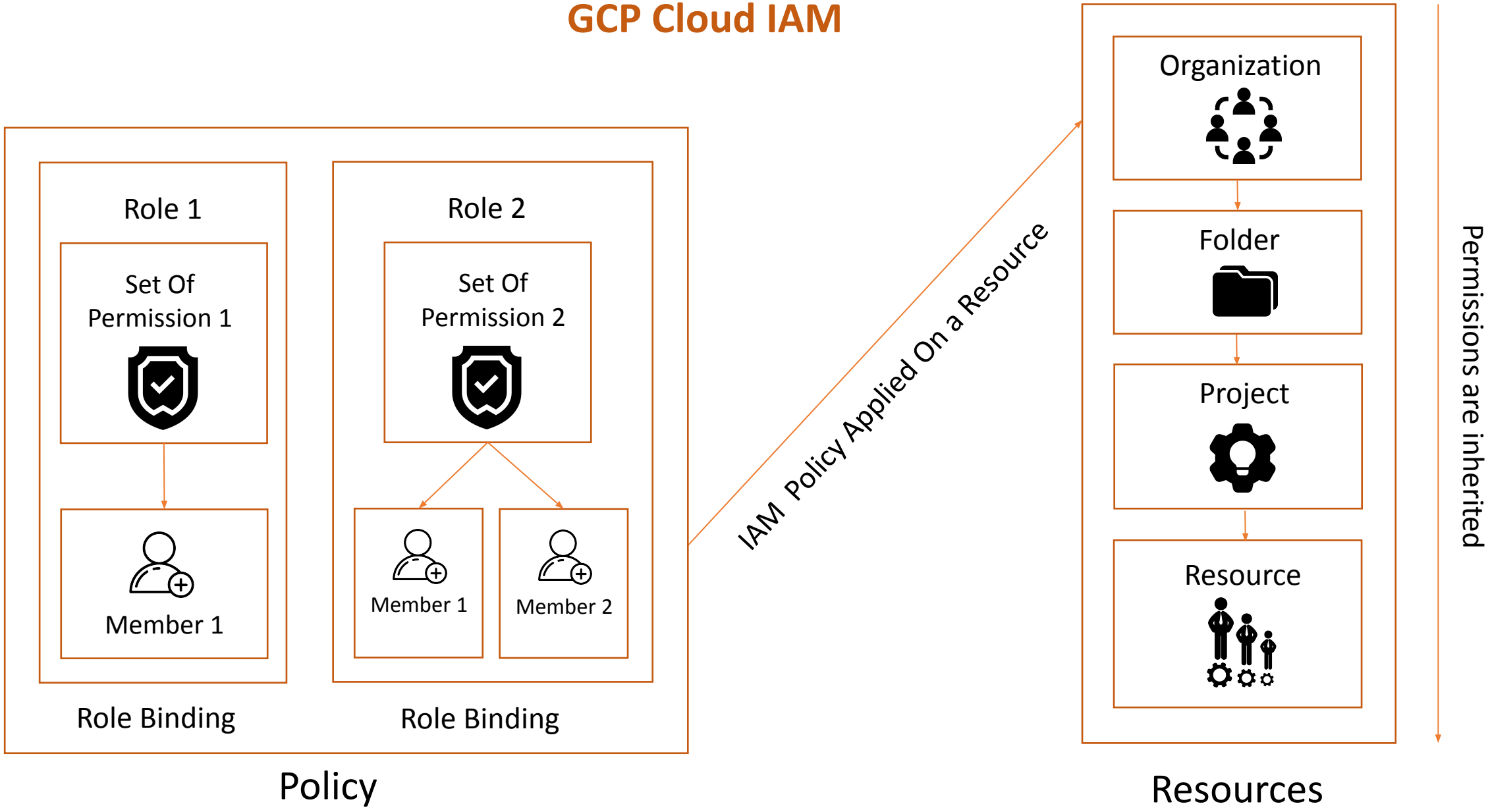
- In IAM policies are attached to resources not identities.

- In IAM we can't directly identify what permissions does an identity contains but we can enumerate what permission an identity have on a specific resource.

- In IAM, permission to access a resource isn't granted directly to the end user. Instead, permissions are grouped into roles, and roles are granted to authenticated members

- GCP IAM Entities :
    - Resources
    - Roles
    - Members

GCP Cloud IAM



Resources:

- Compute Engine virtual machine instances, Google Kubernetes Engine (GKE) clusters, and Cloud Storage buckets are all Google Cloud resources. The organizations, folders, and projects that you use to organize your resources are also resources.
- In IAM, permission can be grant at organization, folder, project and even resource level.
- In IAM, permission are inherited in the gcp hierarchy.
- **Resource hierarchy :**

  Google Cloud resources are organized hierarchically:

  - The organization is the root node in the hierarchy.
  - Folders are children of the organization.
  - Projects are children of the organization, or of a folder.
  - Resources for each service are descendants of projects.

- This policy inheritance is transitive; in other words, resources inherit policies from the project, which inherit policies from folders, which inherit policies from the organization. Therefore, the organization-level policies also apply at the resource level.

Identity [Members] :

- A member can be a Google Account (for end users), a service account (for apps and virtual machines), a Google group, or a Google Workspace or Cloud Identity domain that can access a resource.

- The identity of a member is an email address associated with a user, service account, or Google group; or a domain name associated with Google Workspace or Cloud Identity domains.

Type of member in GCP:

- Google Account
- Service account
- Google group
- Google Workspace domain
- Cloud Identity domain
- All authenticated users
- All users

Roles:

- A role is a collection of permissions. Permissions determine what operations are allowed on a resource. When you grant a role to a member, you grant all the permissions that the role contains.

Type of roles in GCP

  - **Basic roles:** Roles historically available in the Google Cloud Console. These roles are Owner, Editor, and Viewer.
  - **Predefined roles:** Roles that give finer-grained access control than the basic roles.
  - **Custom roles:** Roles that you create to tailor permissions to the needs of your organization when predefined roles don't meet your needs.

- Role is specified in the form of **roles/service.roleName**

Permission:

- Permissions determine what operations are allowed on a resource.
- In the IAM world, permissions are represented in the form of service.resource.verb

Policy:

- The *IAM policy* binds one or more members to a role. When you want to define who (member) has what type of access (role) on a resource, you create a policy and attach it to the resource
- In Policy, there always one role and multiple members.
- Policy always going to attached to a resource.
- An IAM policy is represented by the IAM Policy object.
- An IAM Policy object consists of a list of bindings.
- A Binding binds a list of members to a role.

IAM Policy Structure :

```
{
  "bindings": [
                {"role": "roles/storage.objectAdmin",
                  "members": [
                        "user:user1@example.com",
                        "user:user2@example.com",
                        "serviceAccount:my-other-app@appspot.gserviceaccount.com",
                        "group:admins@example.com",
                        "Domain:google.com"]  },


                {"role": "roles/storage.objectViewer",
                  "members": [
                    "user:user3@example.com"]  }

          ]

}
```

- Authentication & Enumeration using Google Cloud Portal

Google Cloud Management Portal URL :

`https://console.cloud.google.com/`

Google Workspace [G-Suite] Admin Portal URL :

`https://admin.google.com/`

Google Workspace [G-Suite] Users Portal URL :

`https://myaccount.google.com/`

Credential to access GCP Portal :

A. `Username`
B. `Password`
C. `Cookies`

Credential (Cookies) Stored :

Google cloud Platform Cookie : OSID, HSID, SID, SSID, APISID, SAPISID, LSID

Host : .google.com & console.cloud.google.com

- Authentication & Enumeration using Google Cloud CLI (gcloud)

Google Cloud Login :

1. User Account (Username + Password)

```
gcloud auth login
```

2. Service Account

   a. Custom Service Account (App ID + Certificate P12 **OR** JSON Key File)

```
gcloud auth activate-service-account  --key-file KeyFile
```

   b. Default Service Account (GCP Compute Instance)

```
gcloud auth activate-service-account
```

Google Cloud Logout :

```
gcloud auth revoke
```

Google Cloud Secrets :

Directory -

```
Windows : C:\Users\UserName\AppData\Roaming\gcloud\
Linux   : /home/UserName/.config/gcloud/
```

Files -

access_tokens.db : access_tokens(**T**) - account_id, access_token, token_expiry, rapt_token (**C**).

credentials.db : credentials (**T**) - account_id, value (**C**).

- Authentication & Enumeration using Google API [Cloud + Workspace]

Google Cloud API URL :

`https://www.googleapis.com/GCPServiceName/Version`

`https://GCPServiceName.googleapis.com/Version/`

G-Suite Admin API URL :

`https://admin.googleapis.com/`

HTTP Request Parameter :

`Apikey : APIKEY`

`Authorization : Bearer AccessToken`

Google API Authentication Methods :

- API Key
  - API Key
- Access Token
  - OAuth Client ID [User Consent]
  - Service Account

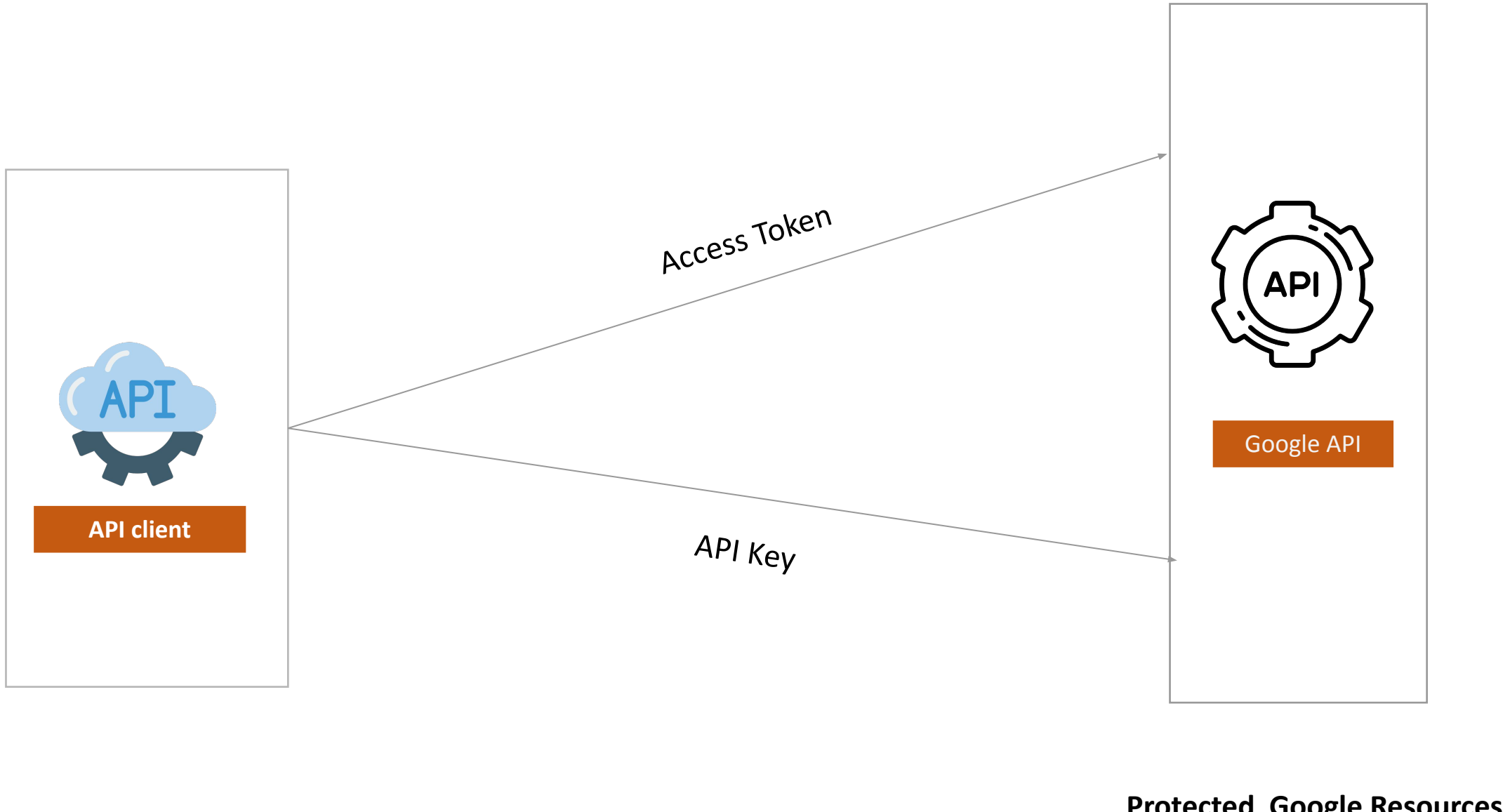
Validating Access Token :

`curl https://www.googleapis.com/oauth2/v1/tokeninfo?access_token=AccessToken`

Tools :

Google API Explorer [https://developers.google.com/apis-explorer/]

Postman


API
API client
Access Token
API Key
API
Google API
Protected Google Resources

List of active User / Service accounts :

```
gcloud auth list
```

Active configuration [user / service account + project] :

```
gcloud config list
```

List of organization in gcp account :

```
gcloud organizations list
```

Lists of iam policy attached to the specified organization :

```
gcloud organizations get-iam-policy OrganizationsID
```

Lists of folder in an organization :

```
gcloud resource-manager folders list --organization OrganizationsID
```

Lists of iam policy attached to the specified folder :

```
gcloud resource-manager folders get-iam-policy FolderID
```

List of projects in an organization :

```
gcloud projects list
```

Lists of iam policy attached to the specified project :

```
gcloud projects get-iam-policy ProjectID
```

List all of service accounts in a project : [Project name is specified using gcloud configuration]

```
gcloud iam service-accounts list
```

Get the IAM policy for a service account :

```
gcloud iam service-accounts get-iam-policy ServiceAccountEmailID
```

Get metadata for a service account in a project:

```
gcloud iam service-accounts describe ServiceAccountEmailID
```

Lists of roles in an origination / project :

```
gcloud iam roles list
```

Lists of permissions in a specified role :

```
gcloud iam roles describe RoleName
```

4. Q/A ?

Thank you

For group / team & enterprise enrollment please contact
“**info@cyberwarfare.live**” for quotation.

<3 from CyberWarFare Labs Team

Follow us on:


info@cyberwarfare.live


www.cyberwarfare.live