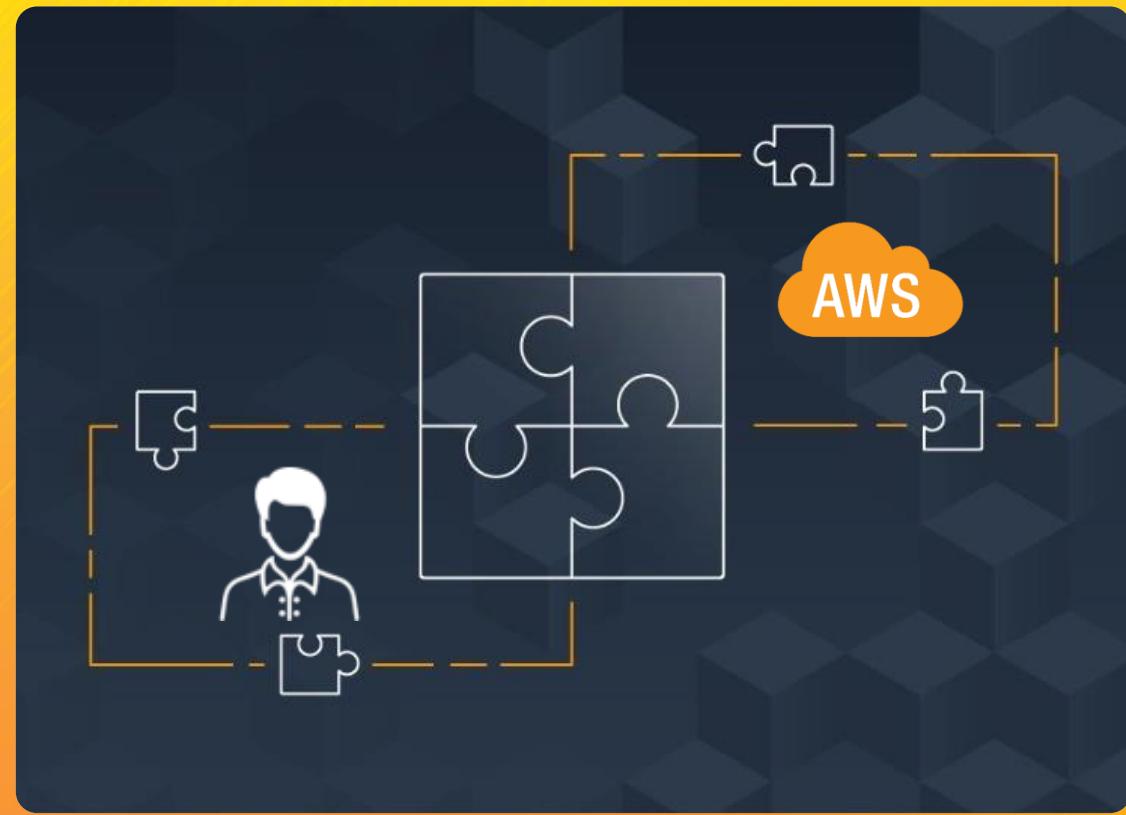




Security in Cloud





Shared Responsibility Model

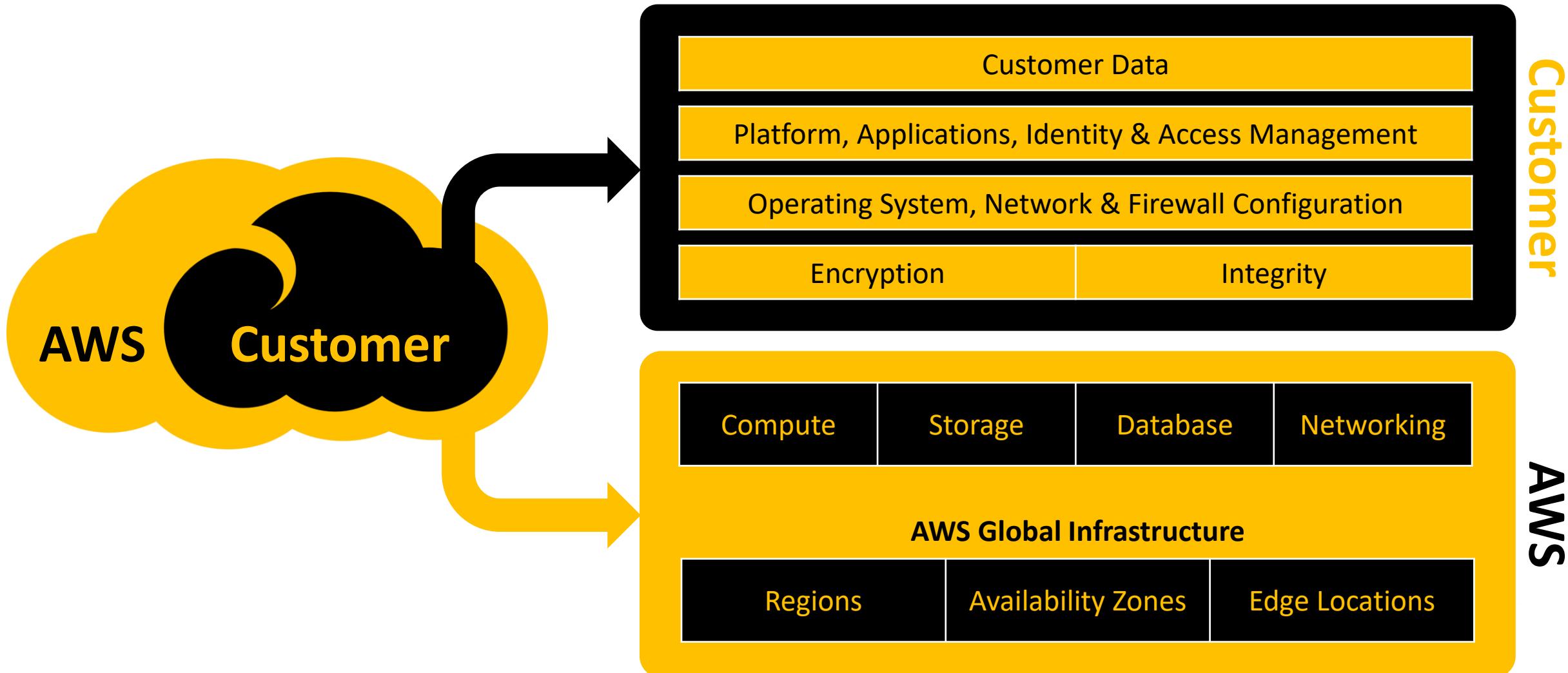
Shared Responsibility for my apartment

	Building Management Responsibility	My Responsibility
	Door Phone	Always verify before remotely opening door
	Camera	Ensure its operational
	Access Card	Keep it safe
	Lock and Key	Always lock
	Fire Exit	Keep it clear

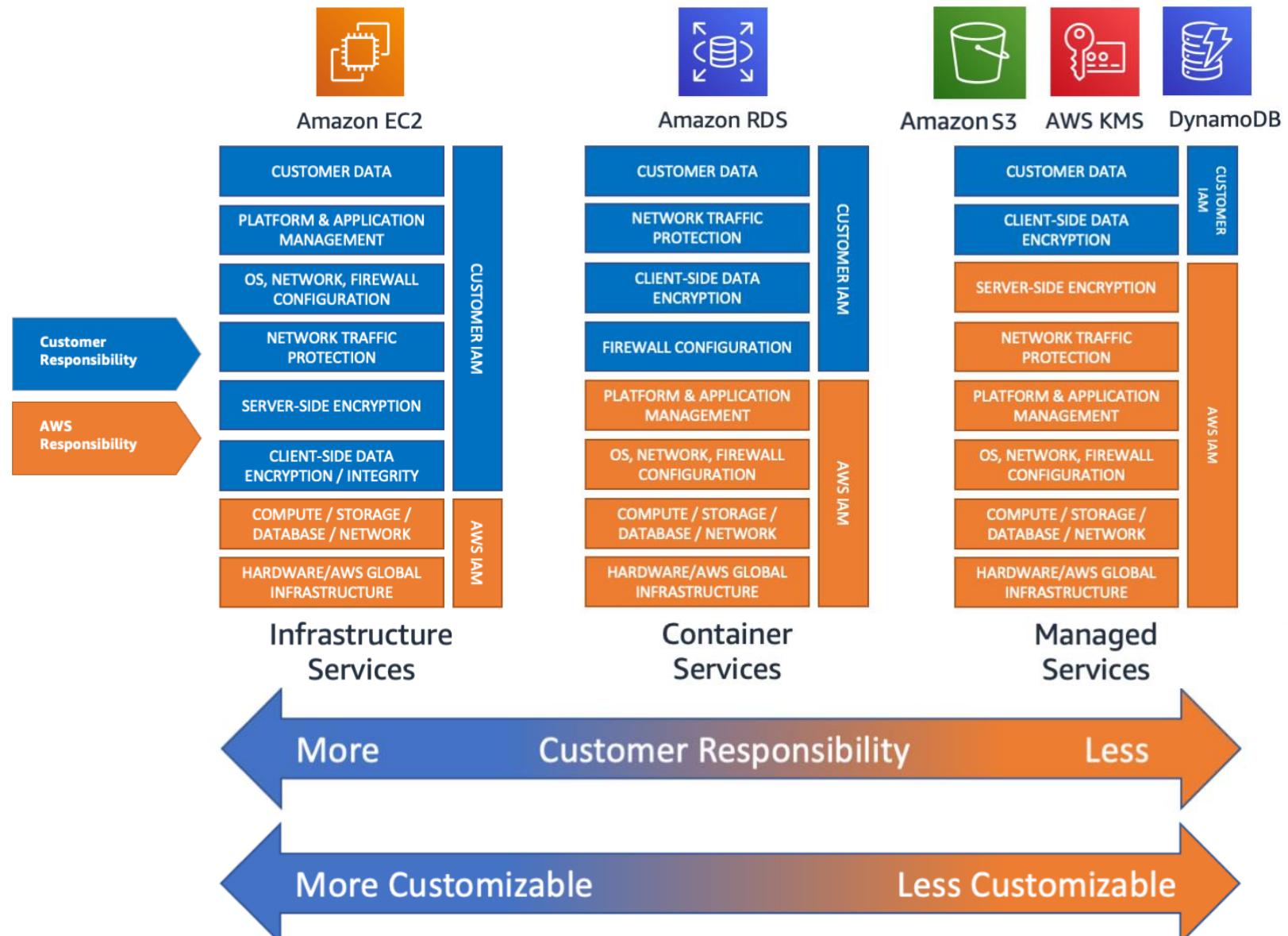
AWS Shared Responsibility Model

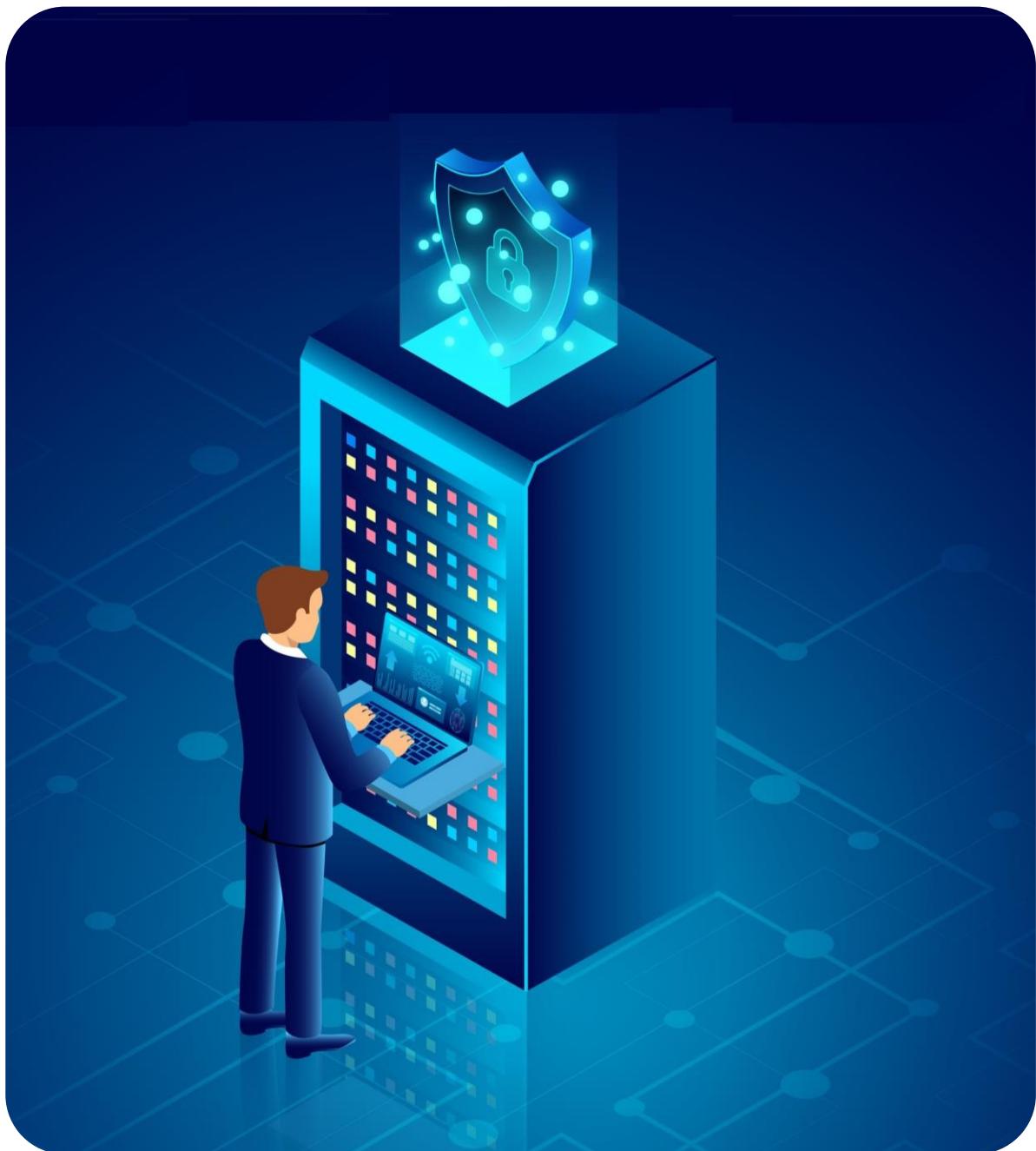


AWS Shared Responsibility Model



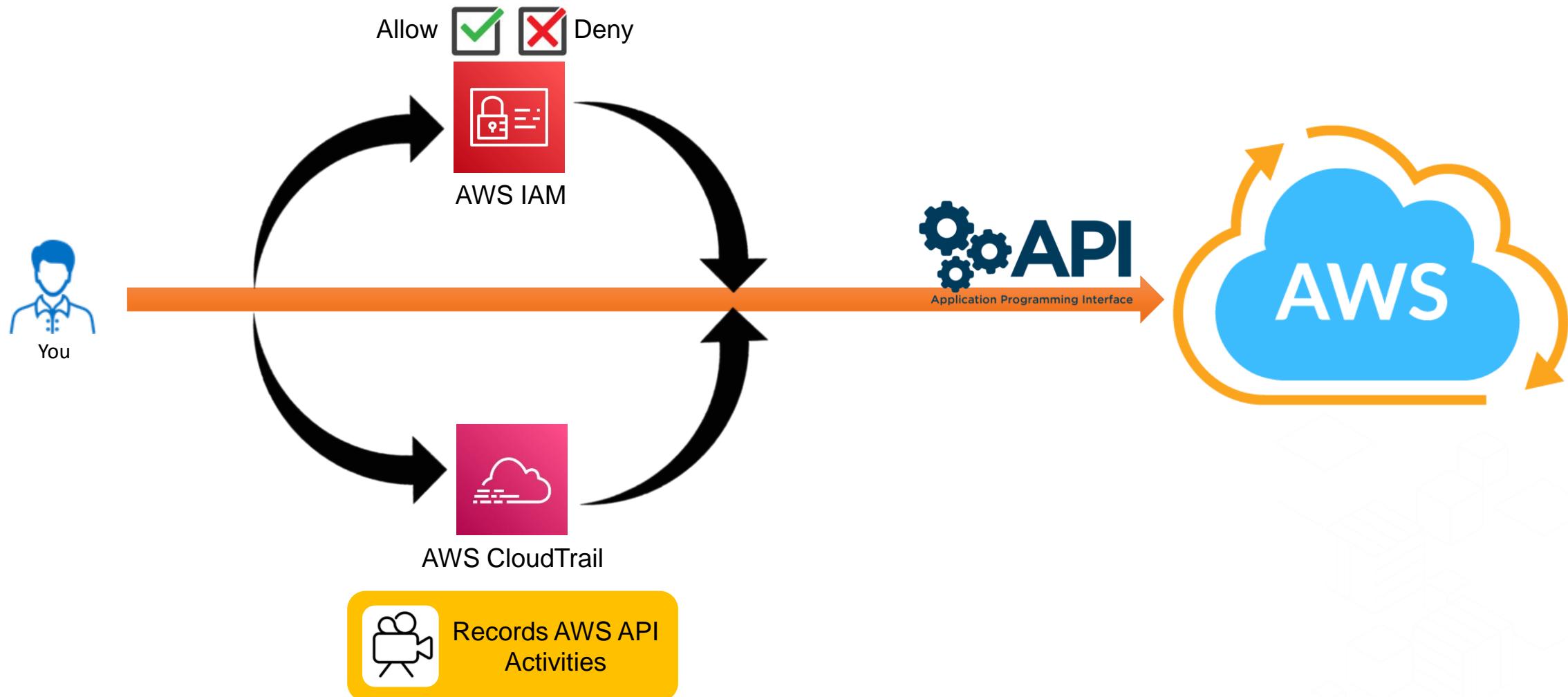
AWS Shared Responsibility Model





AWS Identity & Access Management (IAM)

Flow of a request to AWS

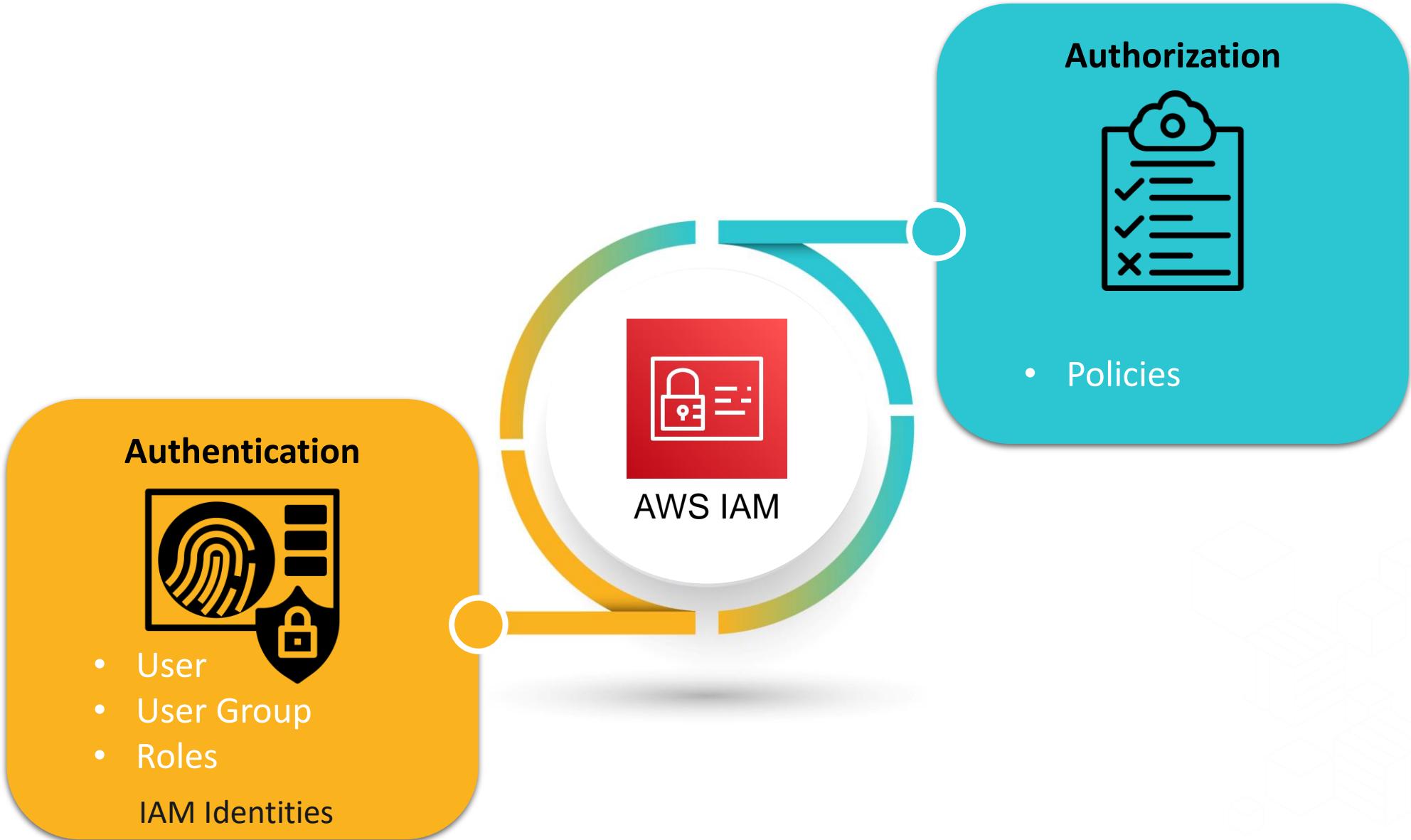


Traveling to another country



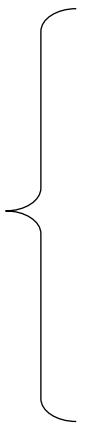
- Visa (Tourist visa / Work visa)
What can you do?
(Authorization)
 - Passport
Who you are?
(Authentication)

AWS IAM



Accessing AWS Services

IAM Username
and Password

A screenshot of the AWS Management Console sign-in page. It includes fields for 'Account' (12 Digit Account ID), 'User Name' (IAM Username), 'Password', 'MFA Code' (Optional MFA), and checkboxes for 'I have an MFA Token' and 'Sign In'. A note at the bottom says 'Sign-in using root account credentials'.

AWS
Management
Console

Access Key and
Secret Access Key



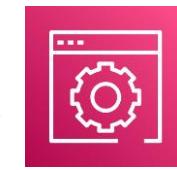
```
C:\Users>aws configure
AWS Access Key ID [None]: Access Key
AWS Secret Access Key [None]: Secret Access Key
Default region name [None]: Region
Default output format [None]: JSON / Text / Table
```



AWS
Command
Line Interface

A screenshot of a code editor showing the AWS CLI configuration file ('credentials') in the '.aws' directory. It contains a section named '[default]' with 'aws_access_key_id = Access Key' and 'aws_secret_access_key = Secret Access Key'.

AWS Tools
and SDKs



AWS
Management
Console



AWS
Command
Line Interface



AWS Tools
and SDKs

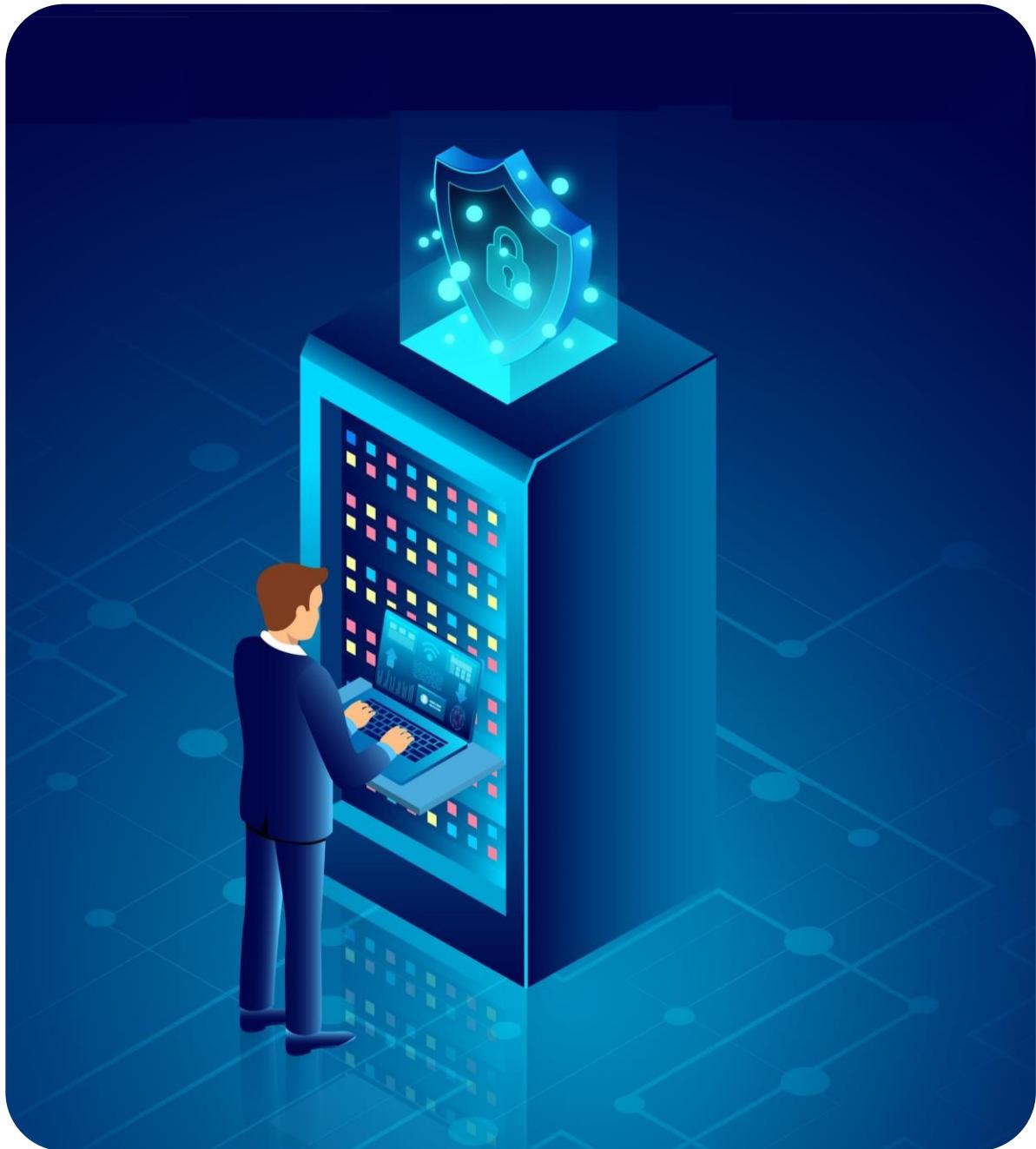


AWS
Command
Line Interface



API
Application Programming Interface





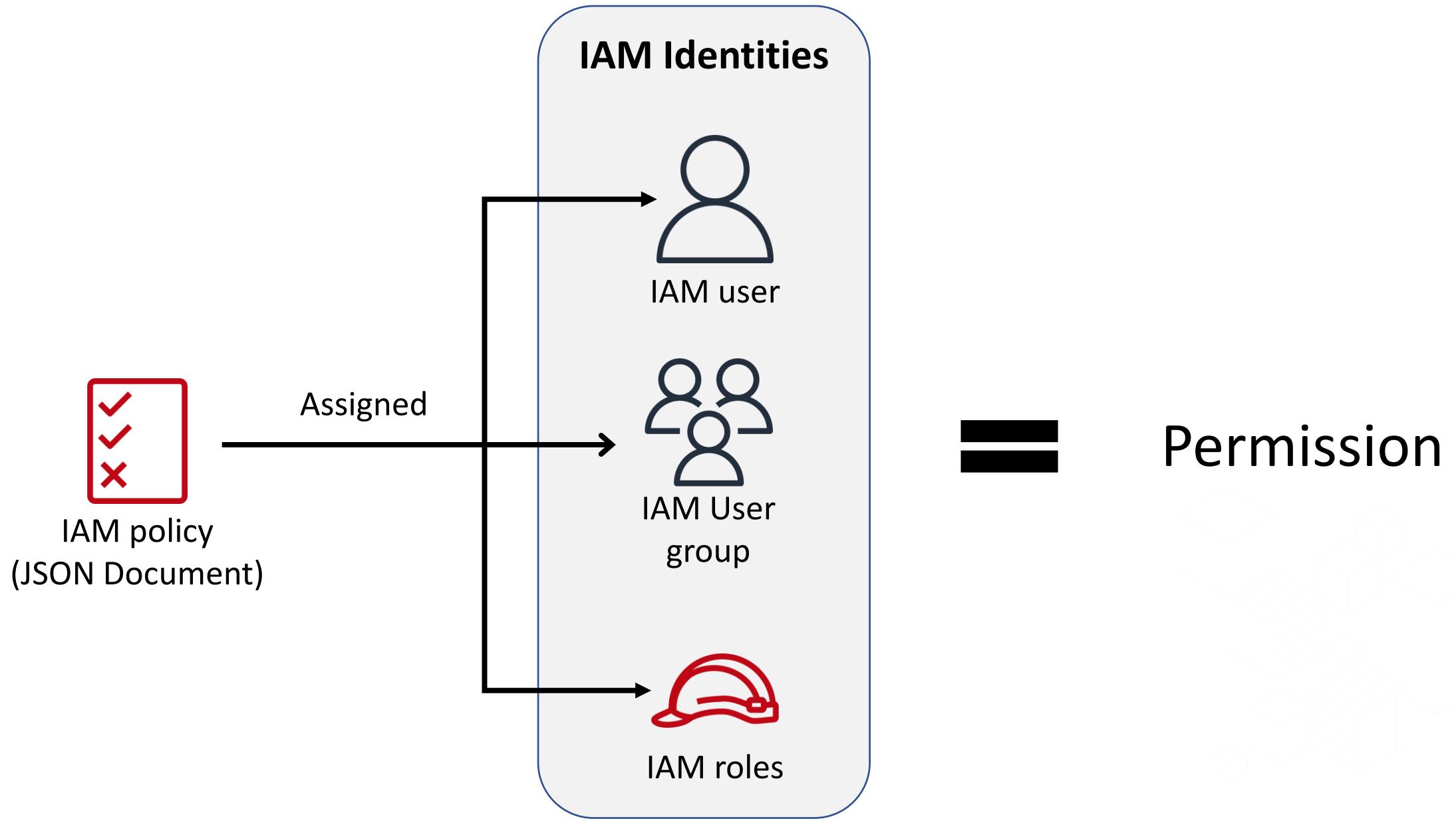
User, UserGroup
and Policies

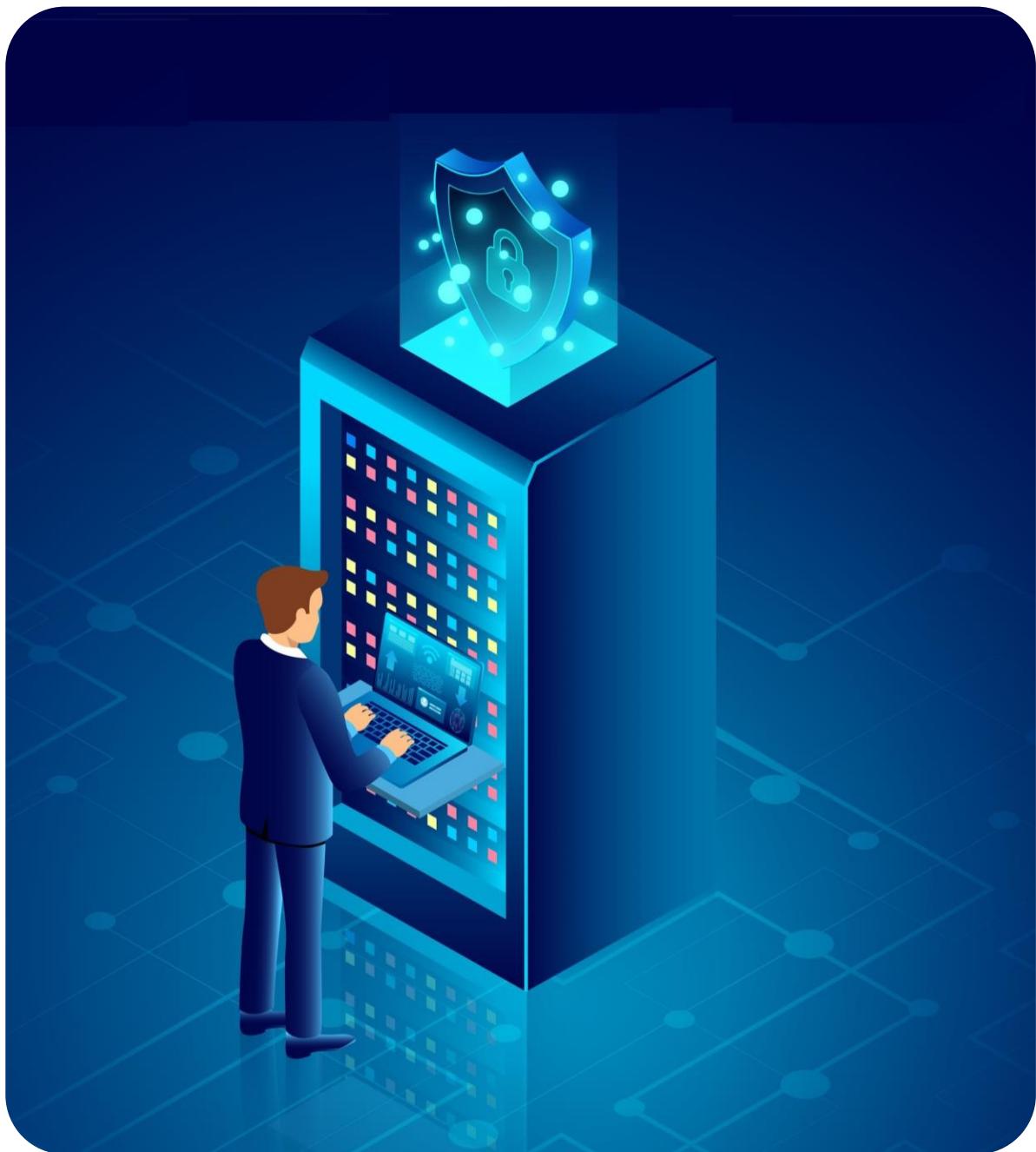
Using IAM

- Business Requirement

User Group	User	Permission
Developers	D1 and D2	<ul style="list-style-type: none">• Can use Amazon S3 service• Can use Amazon EC2 service
QA	Q1 and Q2	<ul style="list-style-type: none">• Can only read Amazon S3 objects• Can only deploy t2.micro EC2 instance in Ireland region
	Admin007	<ul style="list-style-type: none">• Full Admin Permission

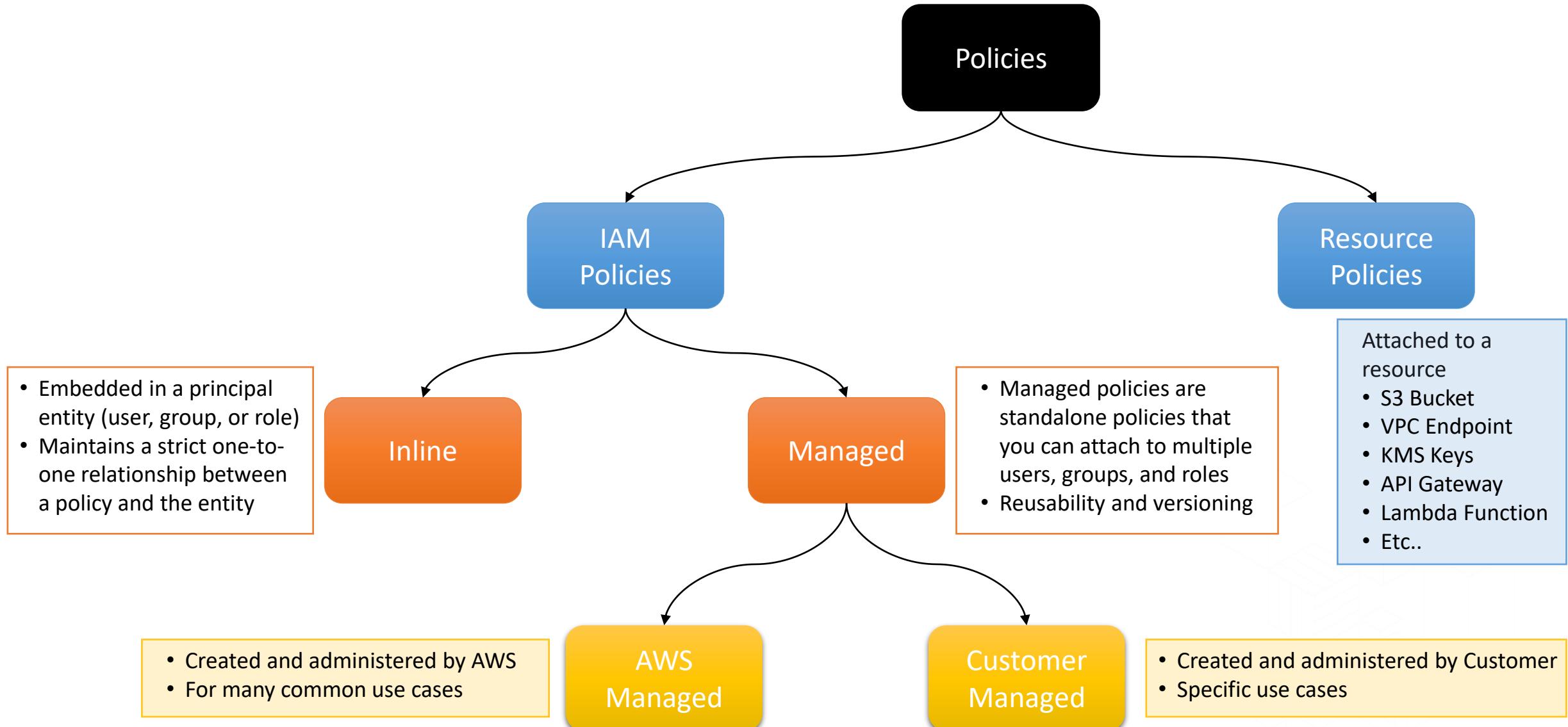
Permission





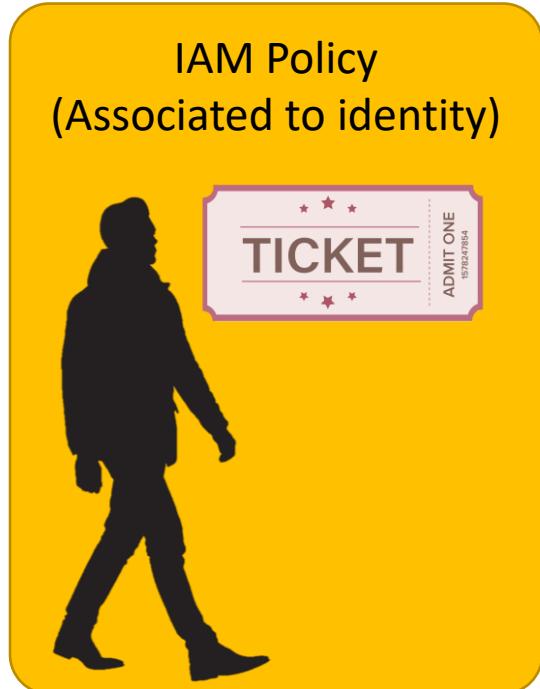
Policies

IAM and Resource Policies



Going to a theater

- To enter either I carry a ticket or have my name on Guest List



Policies

Resource Policy - S3 Bucket Policy

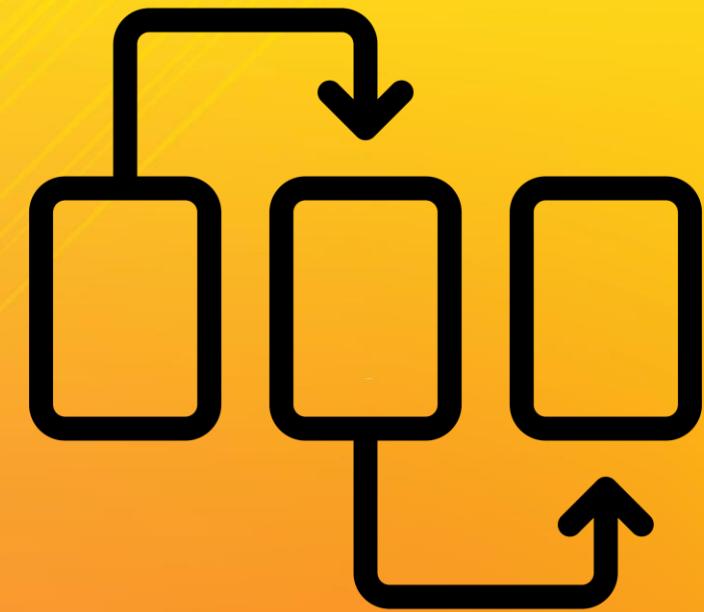
```
{  
  "Id": "Policy1676316462609",  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "Stmt1676316460592",  
      "Action": [  
        "s3:GetObject"  
      ],  
      "Effect": "Allow",  
      "Resource": "arn:aws:s3:::bucket/*",  
      "Principal": "*"  
    }  
  ]  
}
```

Resource Policy - API Gateway Policy

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": [  
          "arn:aws:iam::account-id-2:user/Alice",  
          "account-id-2"  
        ]  
      },  
      "Action": "execute-api:Invoke",  
      "Resource": [  
        "execute-api:/stage/GET/pets"  
      ]  
    }  
  ]  
}
```

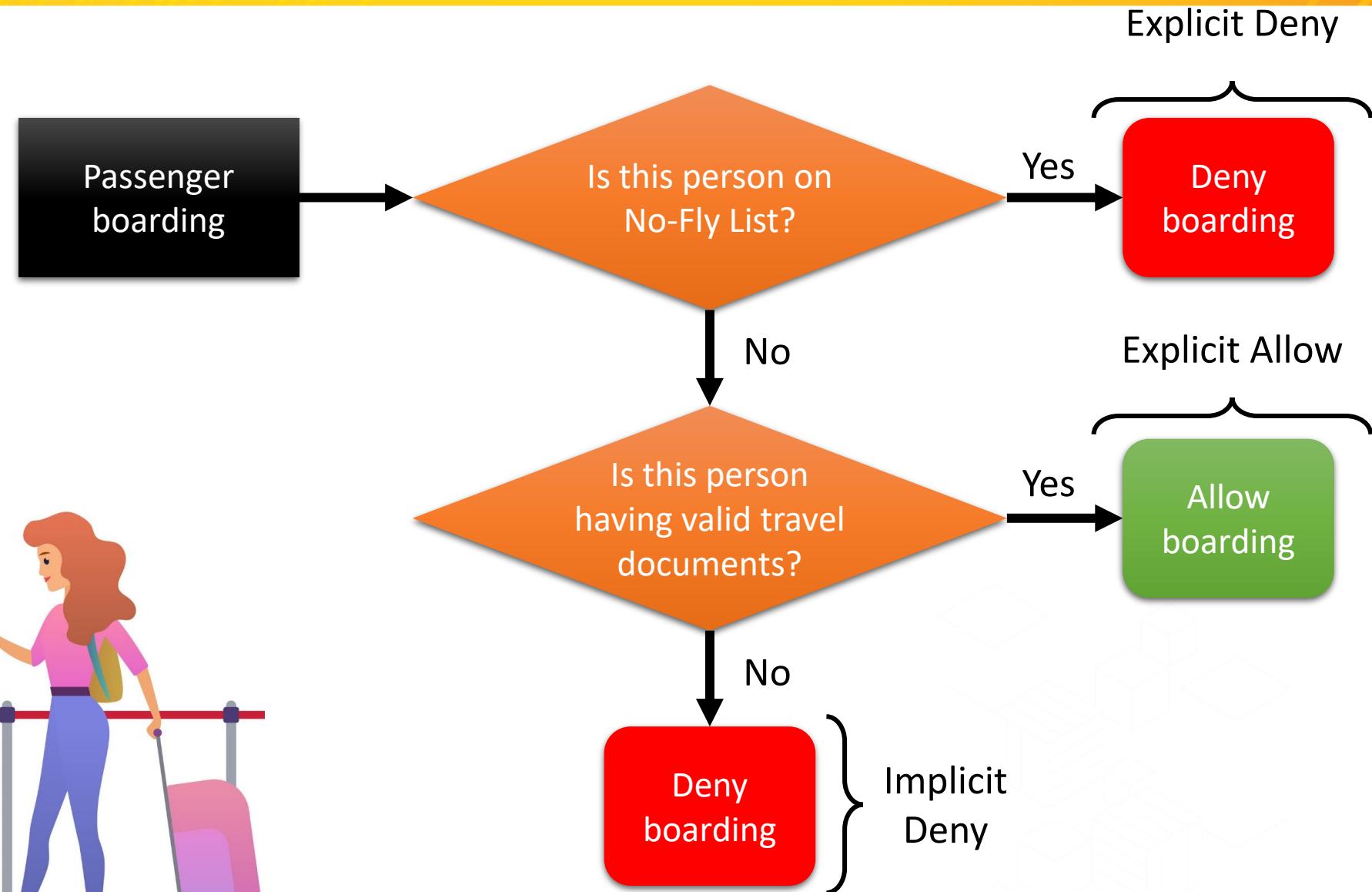
IAM Policy - AdministratorAccess

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": "*",  
      "Resource": "*"  
    }  
  ]  
}
```

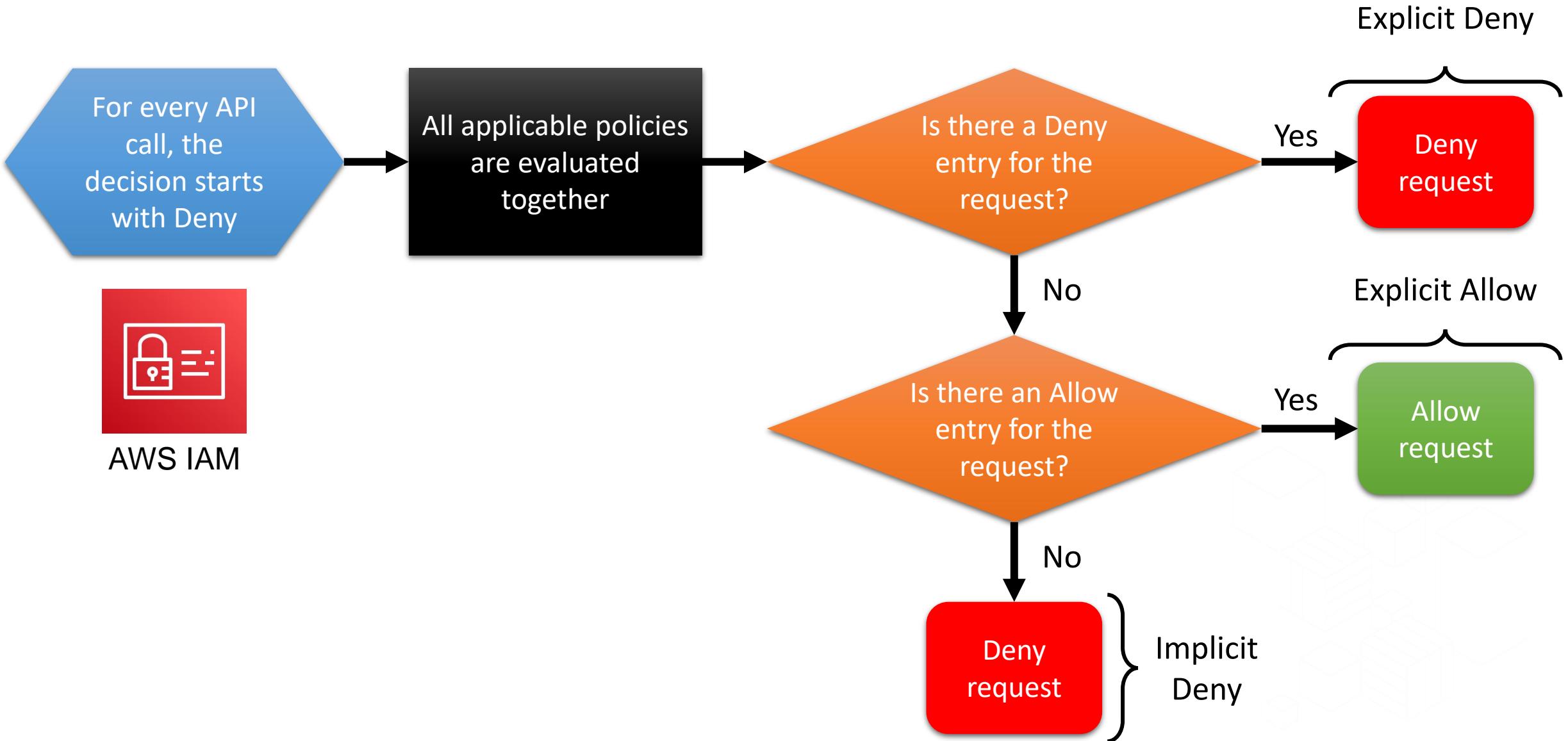


Policies Evaluation Logic

Boarding a Flight



IAM Policy Evaluation Logic



Policy Evaluation - Example

- Scenario 1

Group	IAM Policy	Bucket Policy	Can QA1 read S3 object?
QA	Allow – S3 Full Access	No bucket policy	✗
Member – QA1	Deny – S3	No bucket policy	Explicit Deny

- Scenario 2

Group	IAM Policy	Bucket Policy	Can QA1 read S3 object?
QA	No S3 Policy	No bucket policy	✗
Member – QA1	No S3 Policy	No bucket policy	Implicit Deny

- Scenario 3

Group	IAM Policy	Bucket Policy	Can QA1 read S3 object?
QA	No S3 Policy	Allow read for QA Group	✓
Member – QA1	No S3 Policy	No bucket policy	Explicit Allow

Policy Evaluation Logic

Step 1

- For every API call all applicable policies are combined.

Step 2

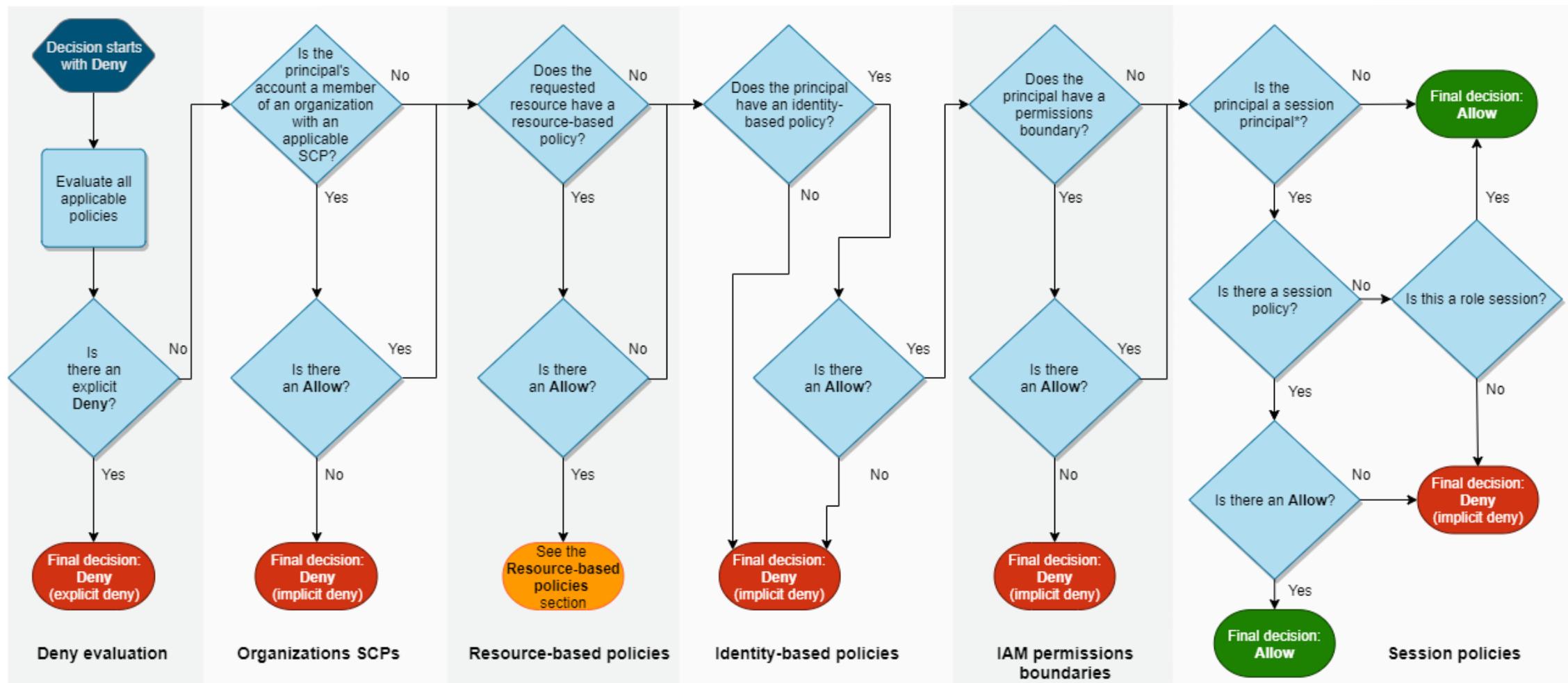
- If there is a matching Deny statement – decision is Deny.

Step 3

- If there is a matching Allow statement – decision is Allow.

Step 4

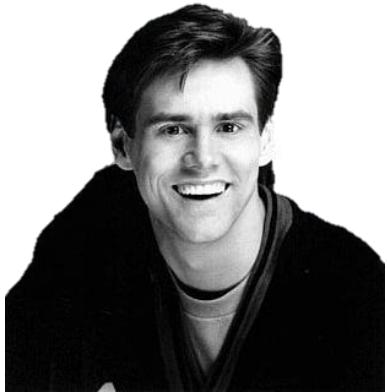
- If there is no matching statement – decision is Deny.





Roles

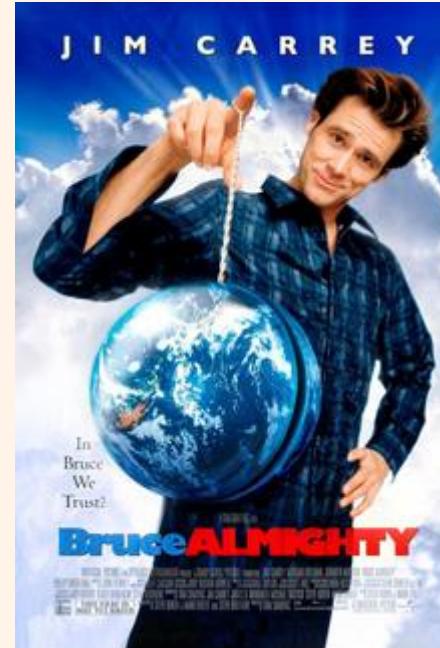
Understanding Role



Jim Carrey

Canadian-American
Actor

Bruce Almighty



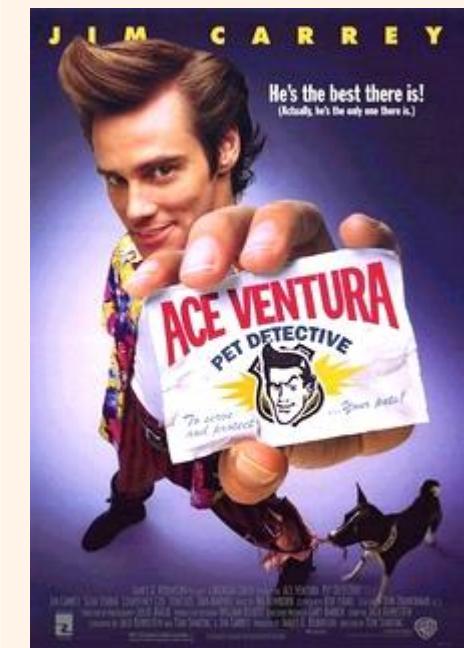
Role
God

The Mask



Role
The Green Thing

Ace Ventura



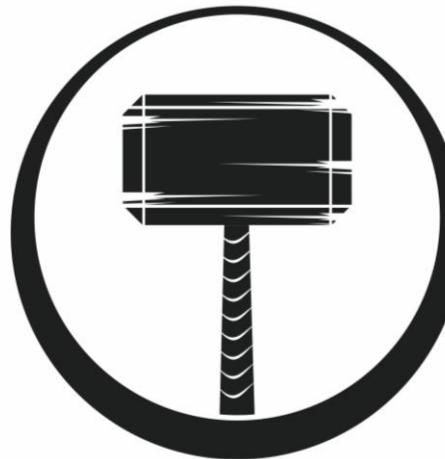
Role
Pet Detective

Assuming a role



Trust Policy

- Mjölnir - can only be lifted by Thor. It trusts Thor to lift it.



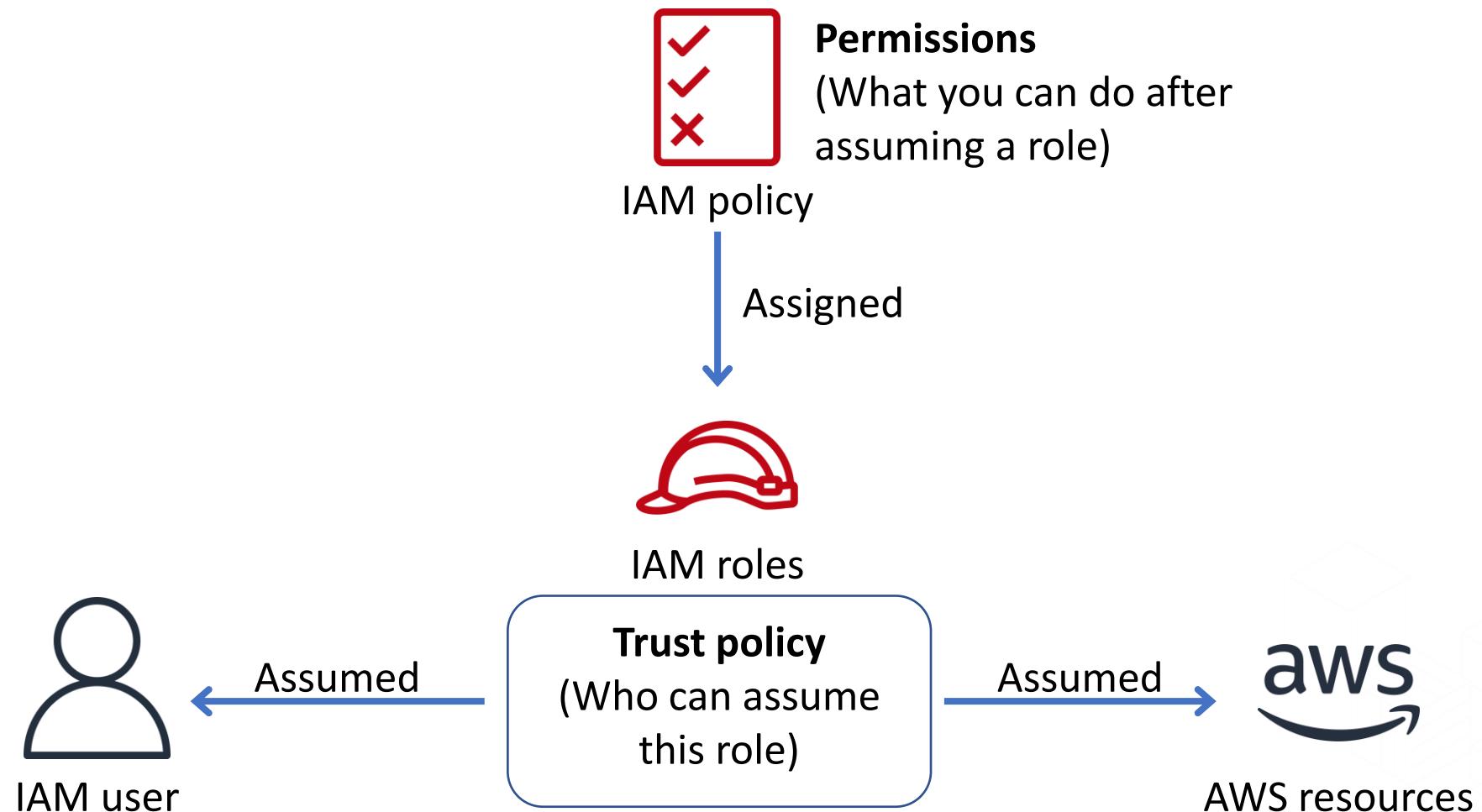
Mjölnir trusts Thor



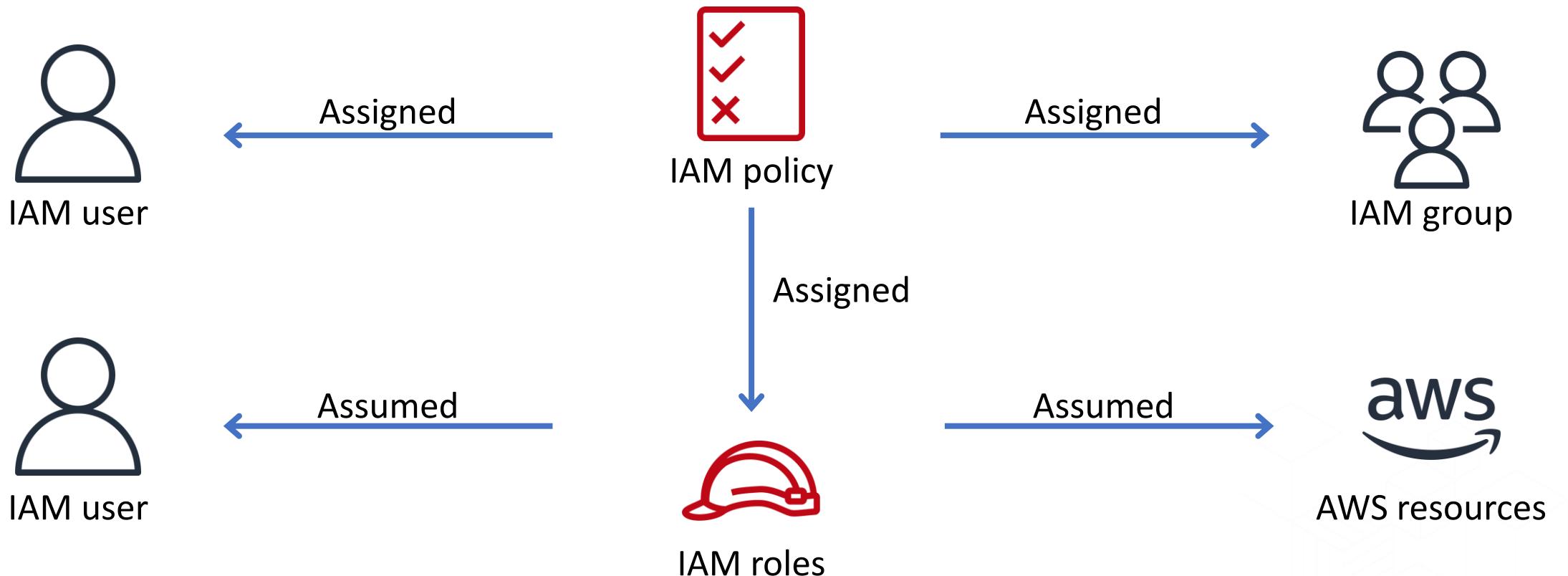
Business Requirement

- Allow Q1 user to assume a role, giving him/her permission equivalent to a developer user
- Three Step Process:
 1. Create a Role
 2. Add Q1 as a trusted principal to assume the role
 3. Add permissions to Role which are equivalent to a developer user

IAM Role – Trust Policy



IAM Role



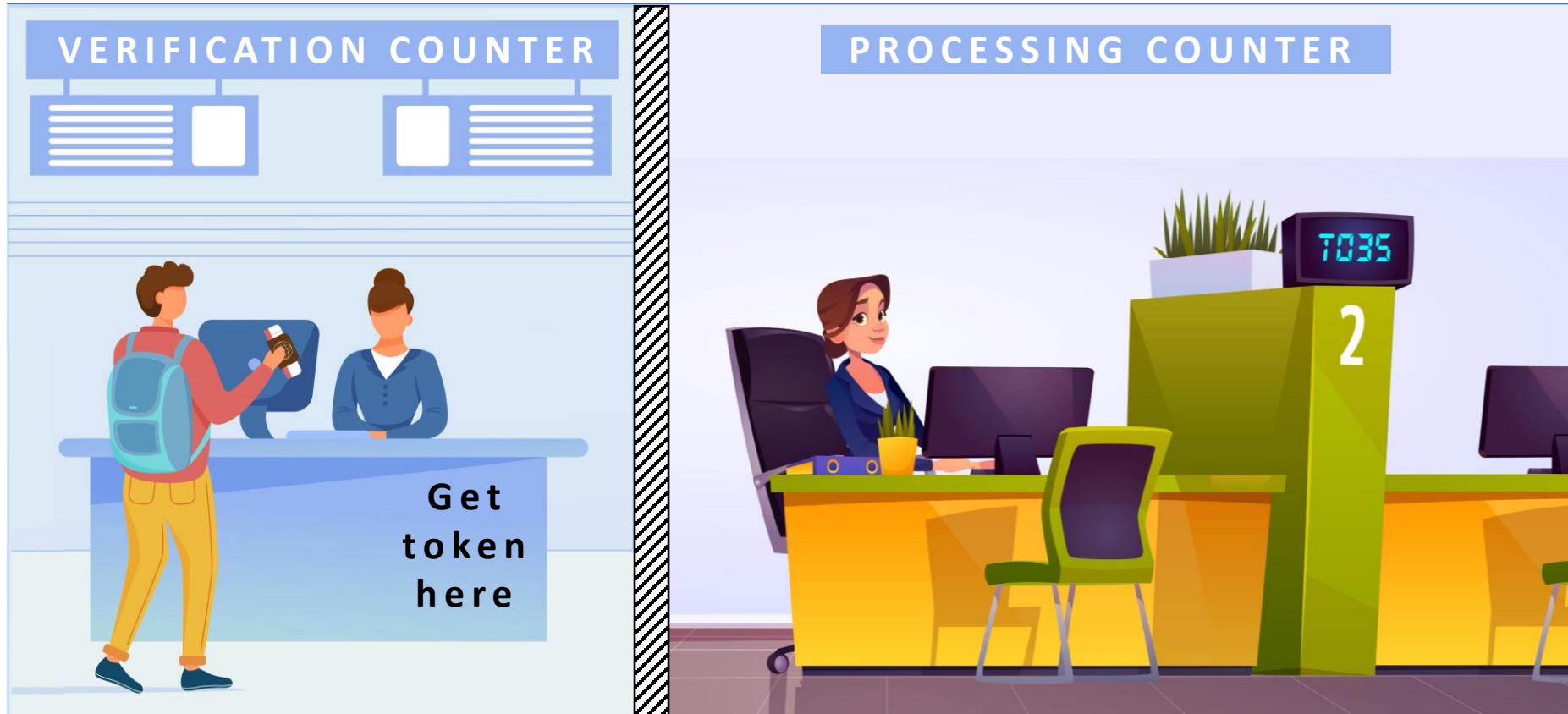
Facts to remember

- Roles can be assumed within AWS account or across AWS accounts.
- At a time only one role can be assumed.
- Trusted Entities to assume a role can be external (SAML / Web Identity).
- Behind the scene AWS Security Token Service (STS) generates a set of temporary security credentials that you can use to access AWS resources.

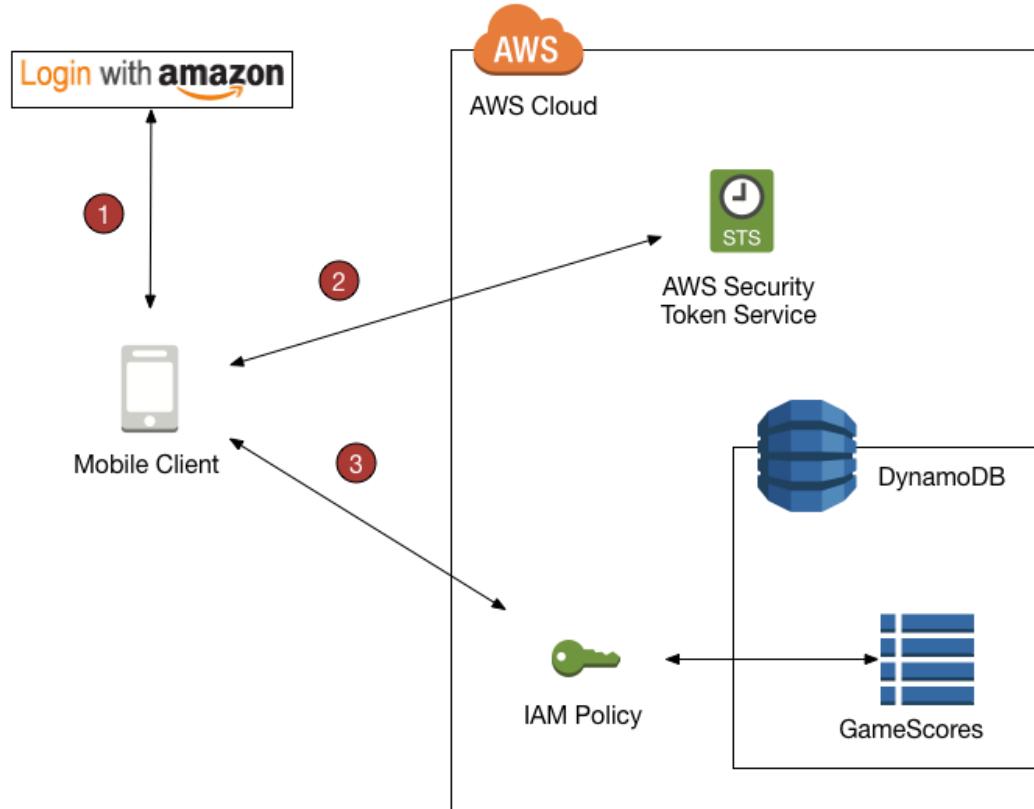
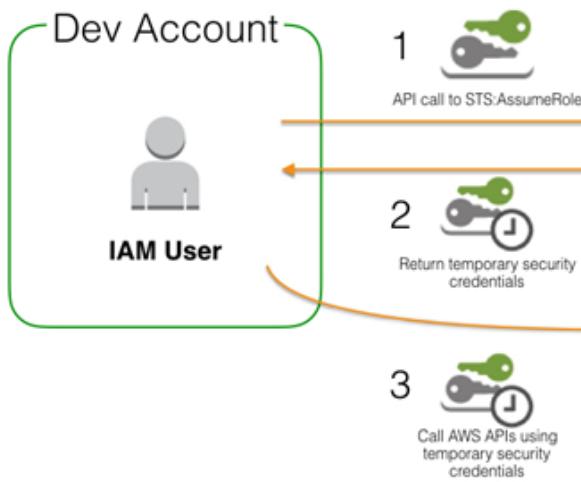
Try Web Identity Federation Playground - 

Applying for a visa

- After preliminary document verification get a token(number)
- Wait and when called go to processing counter



Amazon Security Token Service



Reference:

[Documentation](#)

What?

- AWS Security Token Service (AWS STS) enables you to request temporary, limited-privilege credentials for AWS IAM users or for federated users.
- Temporary security credentials are short-term and are not stored with the user but are generated dynamically and provided when requested. They can be configured to last for a few minutes or for several hours.

Category:

Security,
Identity, and
Compliance

Why?

- If you use AWS STS, you do not have to distribute or embed long-term AWS security credentials with an application.
- You can provide access to your AWS resources to users without having to define an AWS identity for them.

When?

- Temporary credentials are useful in scenarios that involve identity federation, delegation, cross-account access, and IAM roles. Temporary credentials are the basis for roles and identity federation.
- The temporary credentials have a limited lifetime, so you do not have to rotate them or explicitly revoke them when they're no longer needed.

Where?

- By default, AWS STS is available as a global service, and all AWS STS requests go to a single endpoint that maps to the US East (N. Virginia).
- You can use Regional AWS STS endpoints instead of the global endpoint to reduce latency, build in redundancy, and increase session token validity.
- No matter which Region your credentials come from, they work globally.

Who?

- You can use temporary security credentials to make requests for AWS resources using the AWS CLI or AWS API (using the AWS SDKs).
- When (or even before) the temporary security credentials expire, the user can request new credentials, as long as the user requesting them still has permissions to do so.

How?

- The AWS STS API operations create a new session with temporary security credentials that include an access key pair (consists of an access key ID and a secret key) and a session token. Users (or an application that the user runs) can use these credentials to access your resources.

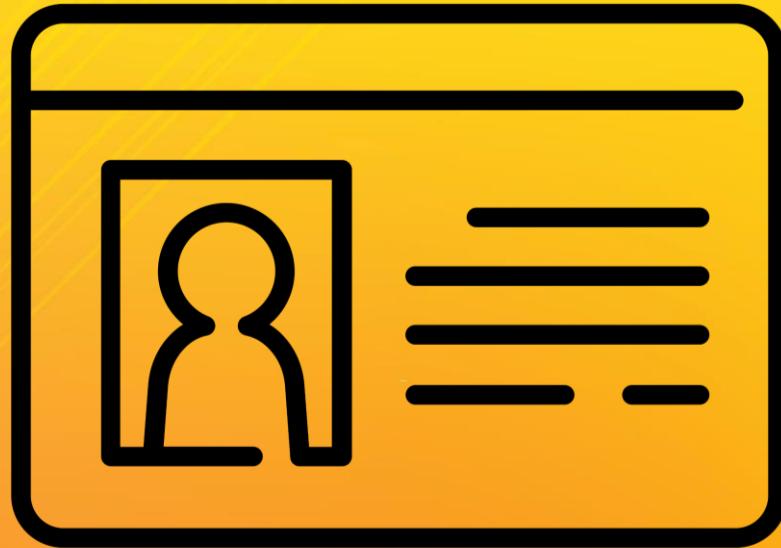
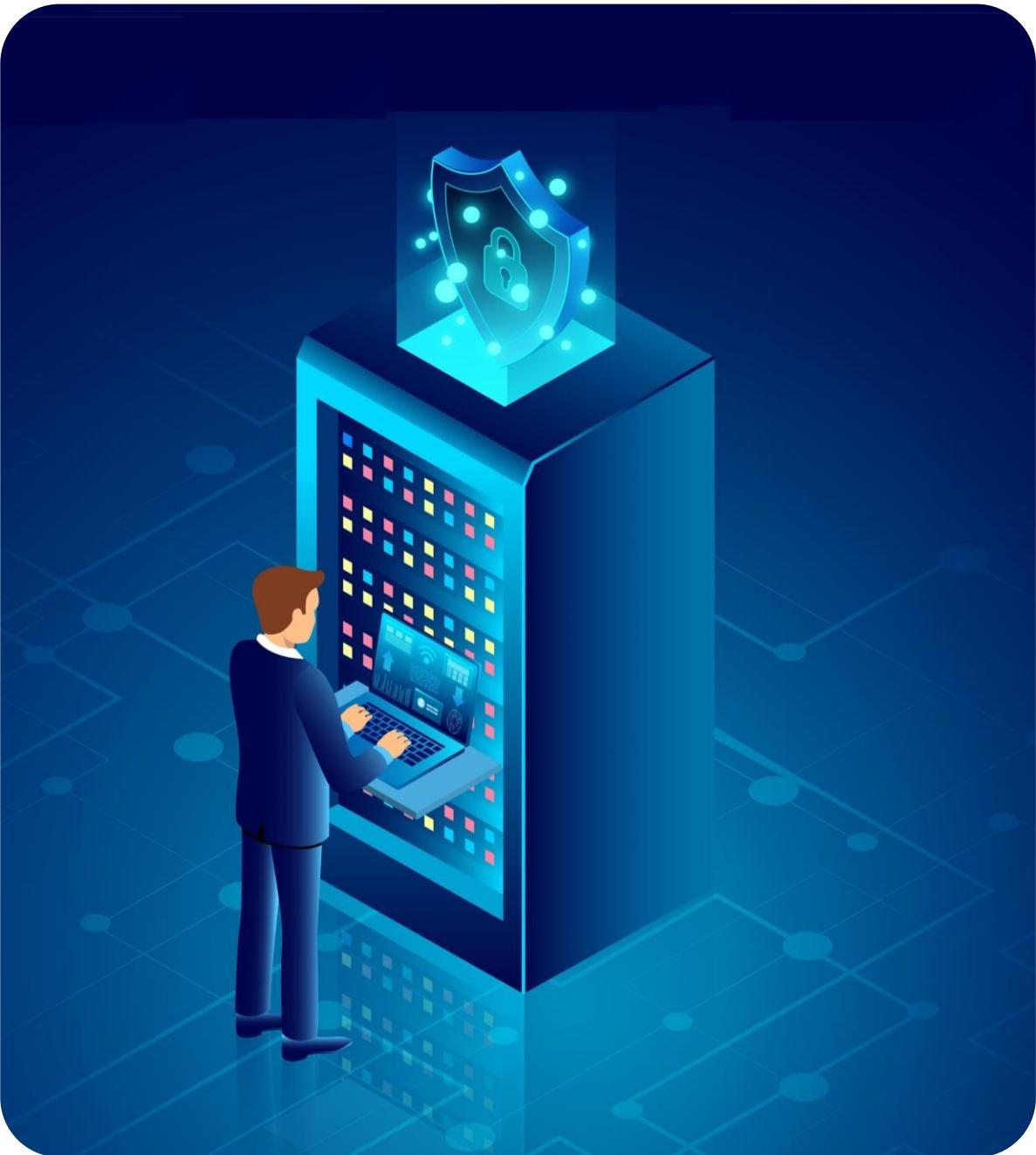
How
much?

- There is no additional cost to use AWS STS.

Created by:

[Ashish Prajapati](#)



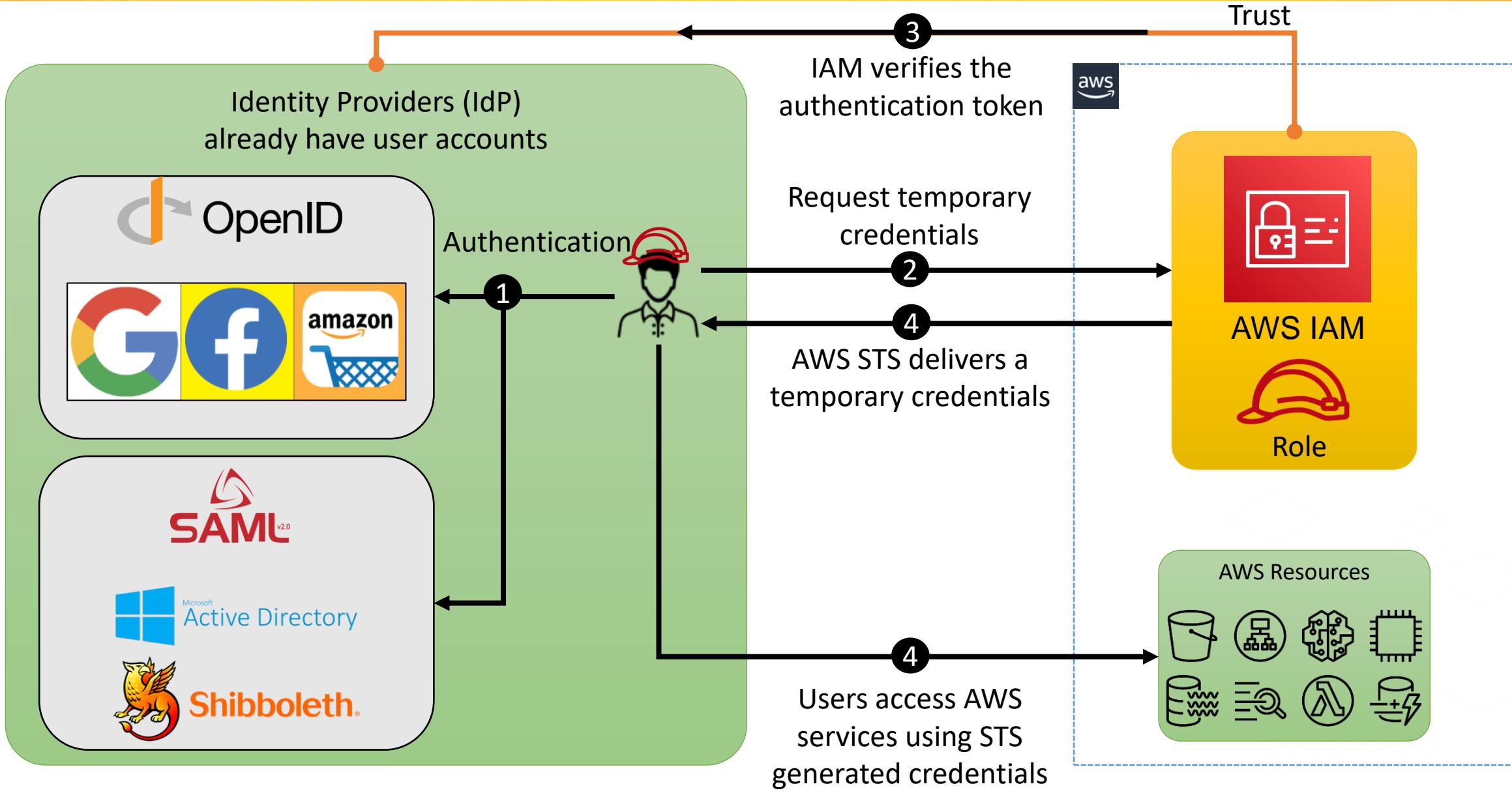


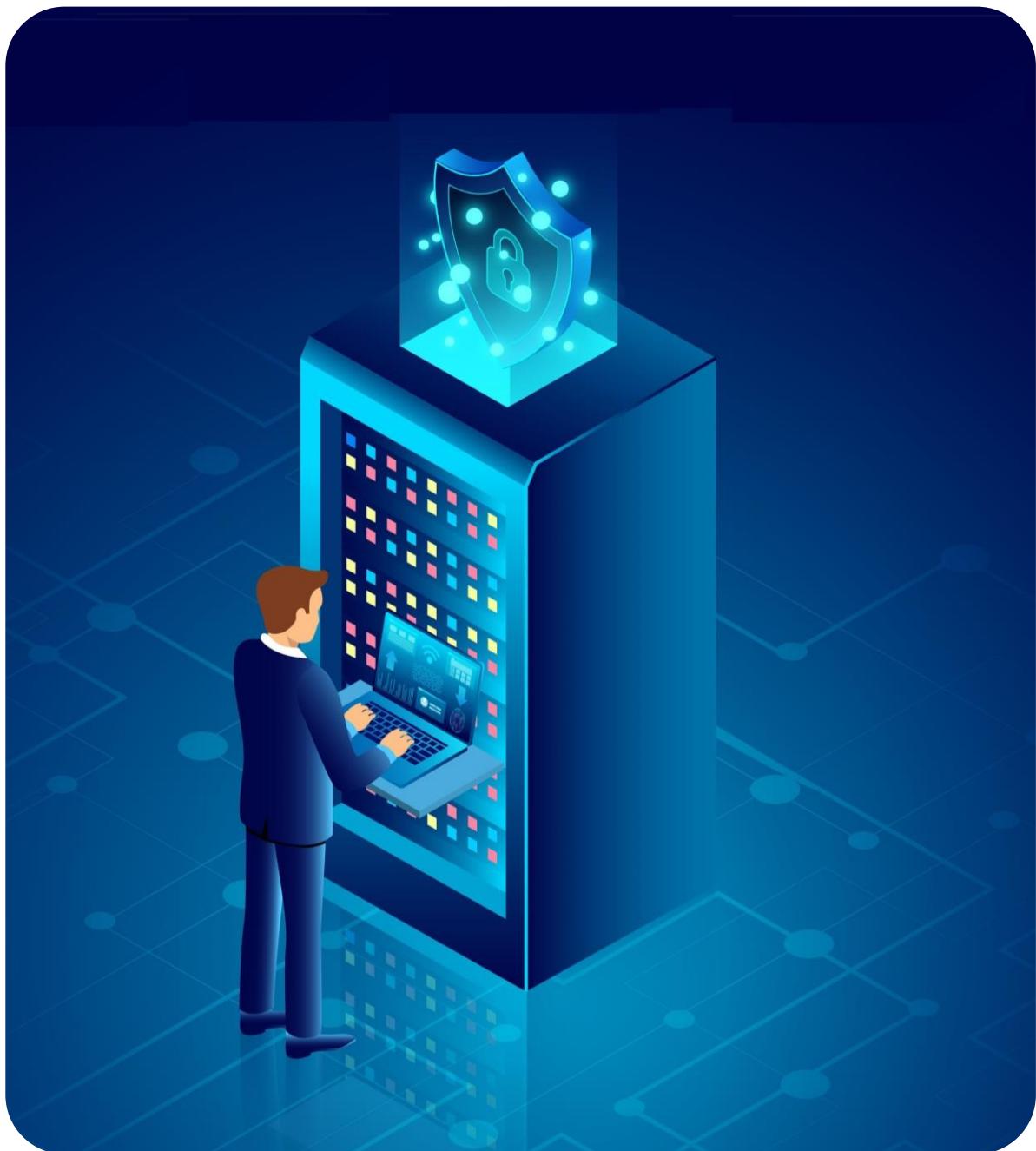
Identity Provider

Identity Providers

- IAM identity providers (IdPs) allow you to manage your identities outside of AWS.
- When you want to configure federation with an external IdP, you create an IAM identity provider to inform AWS about the external IdP and its configuration. This establishes "trust" between your AWS account and the external IdP.
- An external IdP can provide identity information to AWS using either OpenID Connect (OIDC) or Security Assertion Markup Language (SAML). Examples of well-known OIDC identity providers are: Login with Amazon, Facebook, and Google. Examples of well-known SAML identity providers are: Shibboleth and Active Directory Federation Services.

Identity Providers - Federating existing users





AWS IAM Summary

AWS IAM Summary

