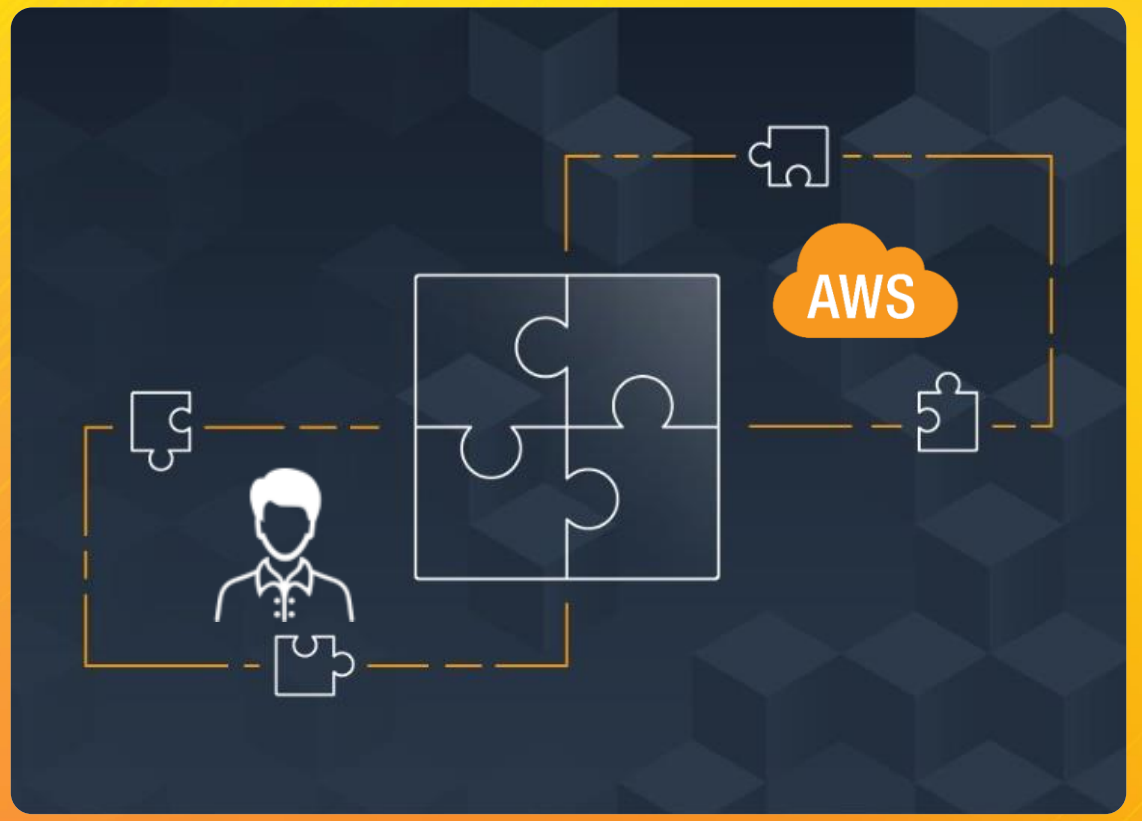
Security in Cloud



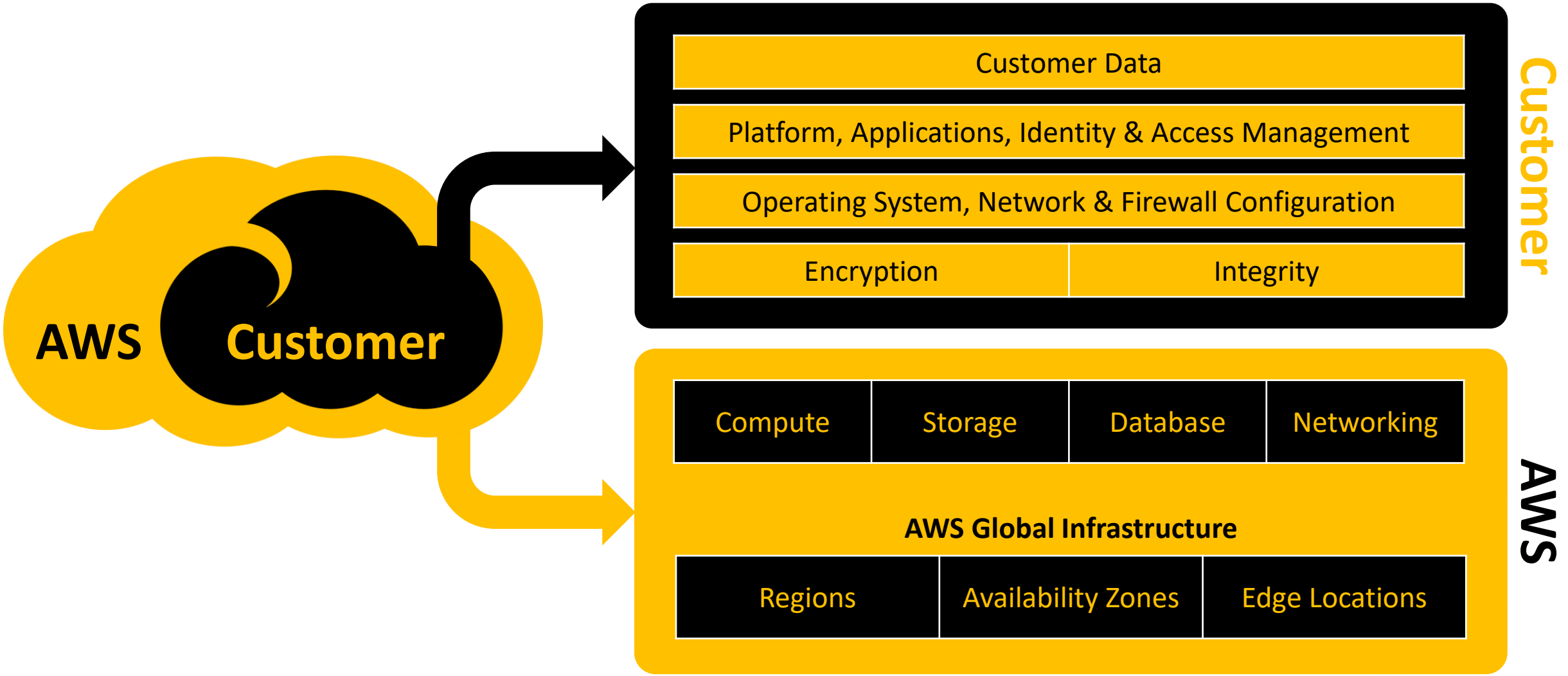
Shared Responsibility Model

Shared Responsibility for my apartment

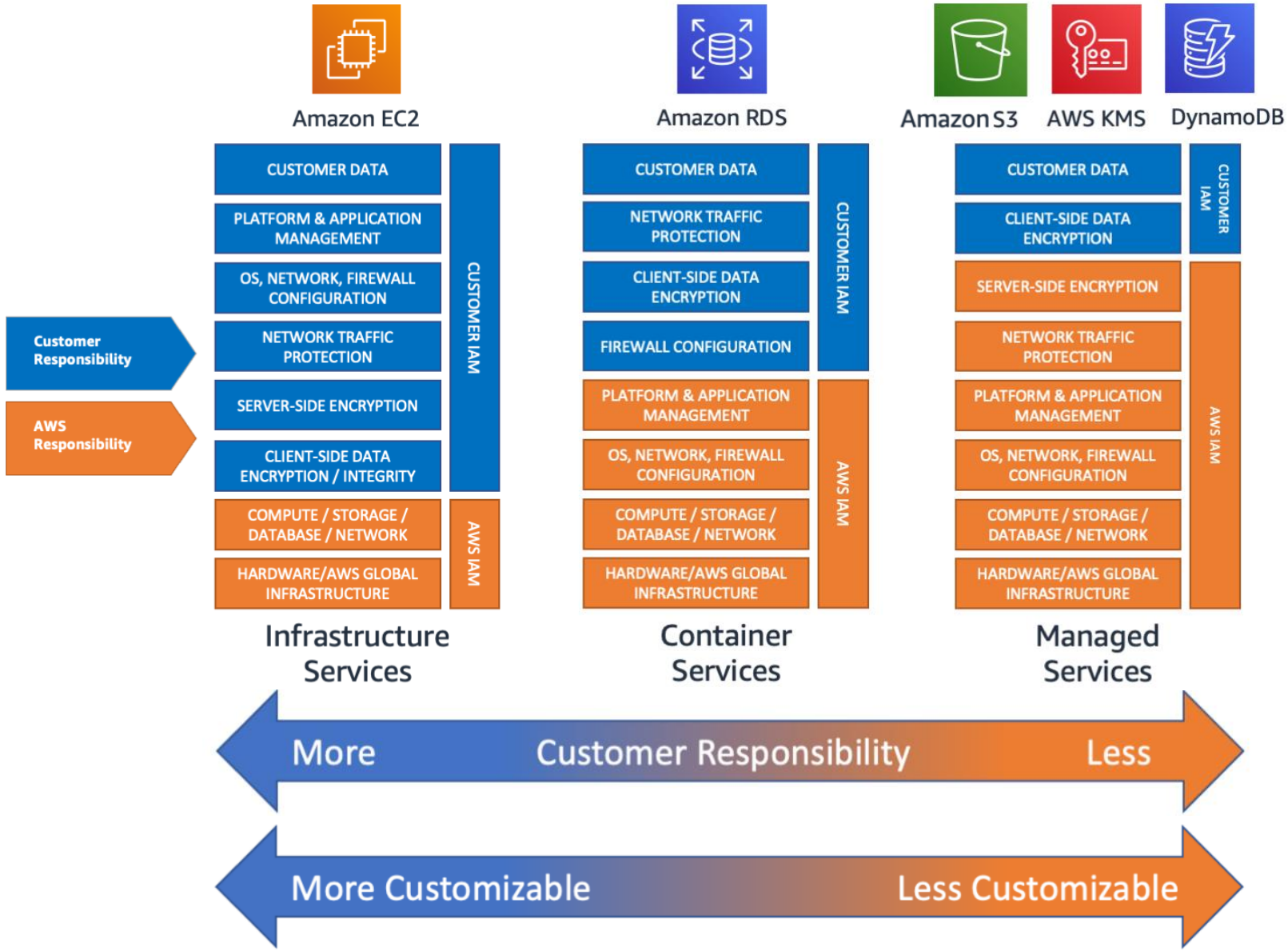
Building Management Responsibility	My Responsibility
Door Phone	Always verify before remotely opening door
Camera	Ensure its operational
Access Card	Keep it safe
Lock and Key	Always lock
Fire Exit	Keep it clear

AWS Shared Responsibility Model

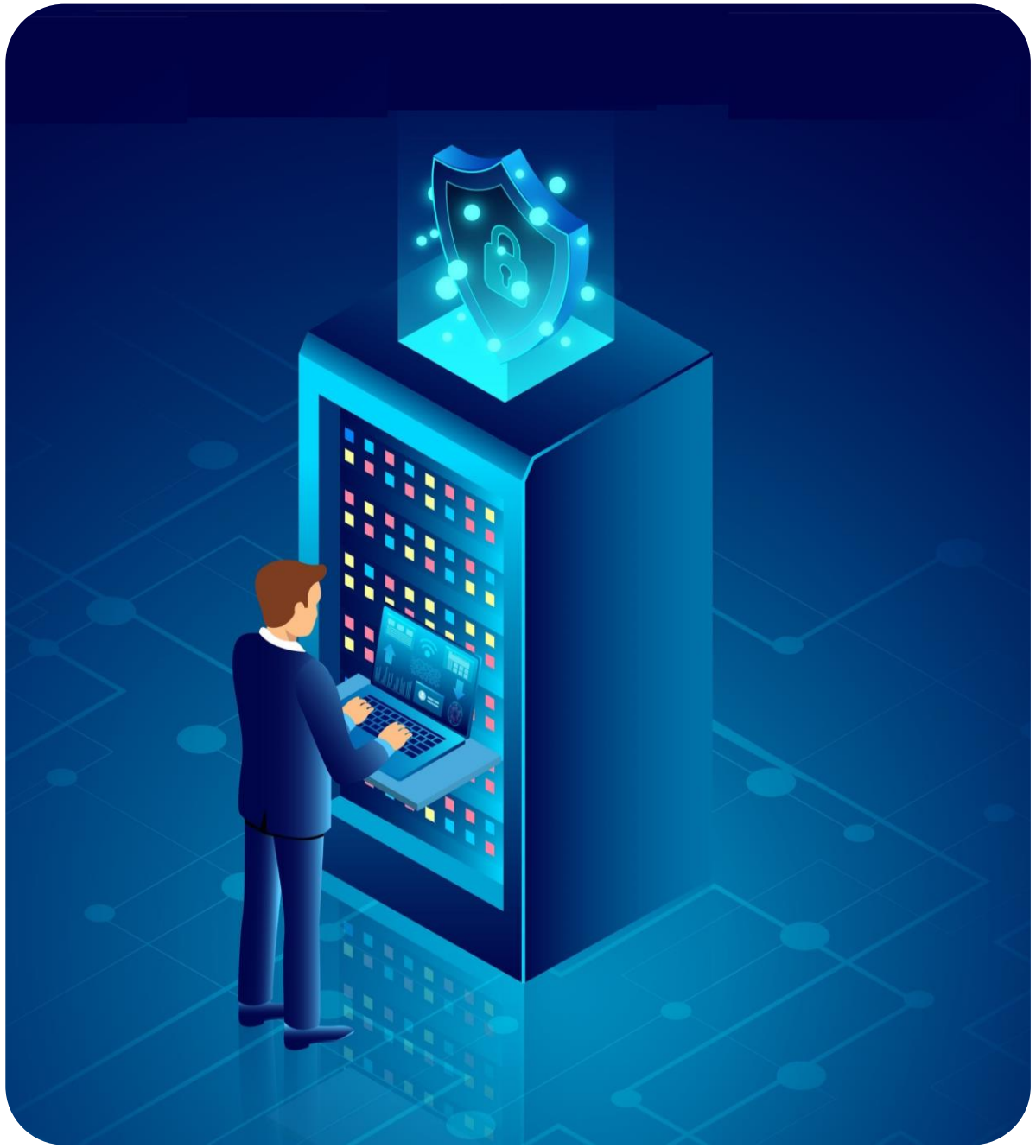
AWS Shared Responsibility Model



AWS Shared Responsibility Model

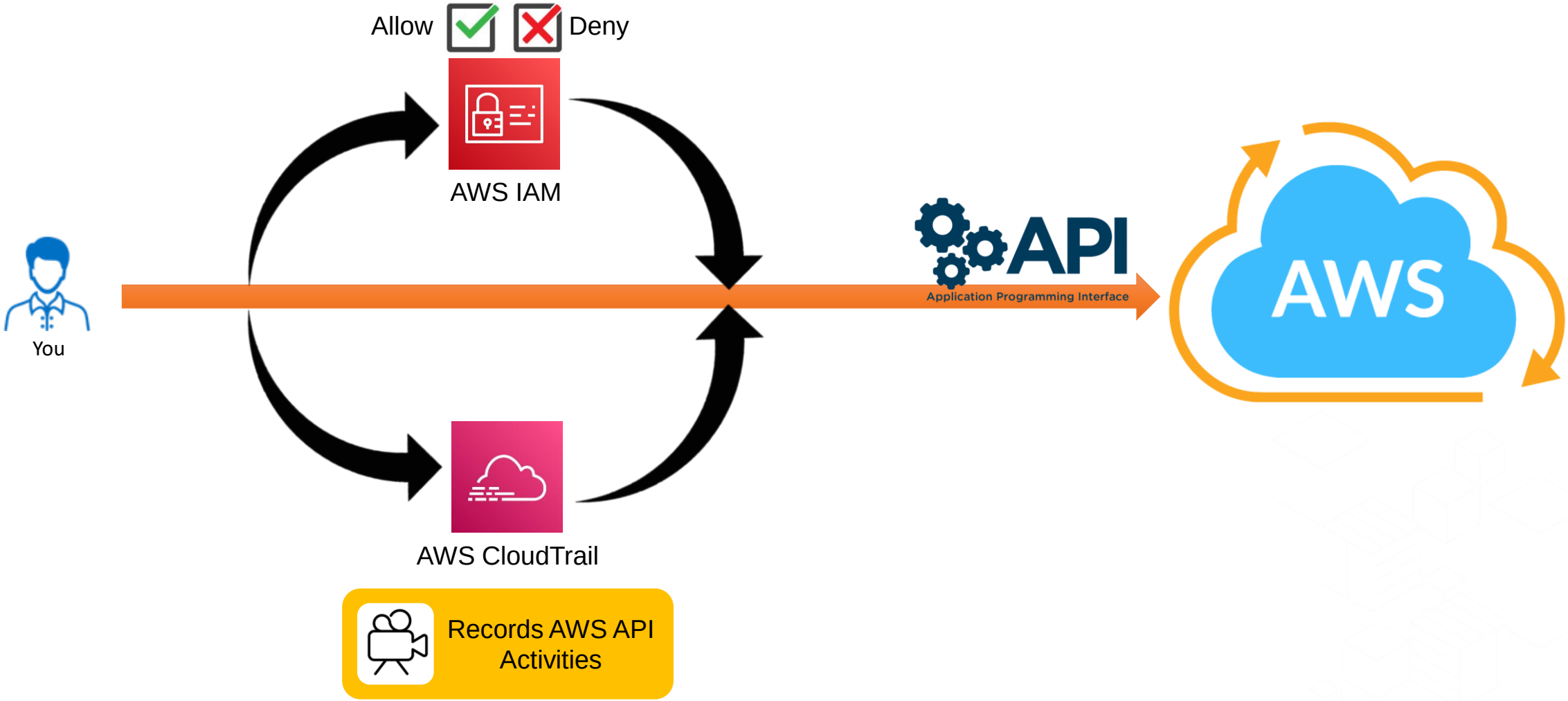


Ref: https://aws.amazon.com/blogs/industries/applying-the-aws-shared-responsibility-model-to-your-gxp-solution/



AWS Identity & Access Management (IAM)

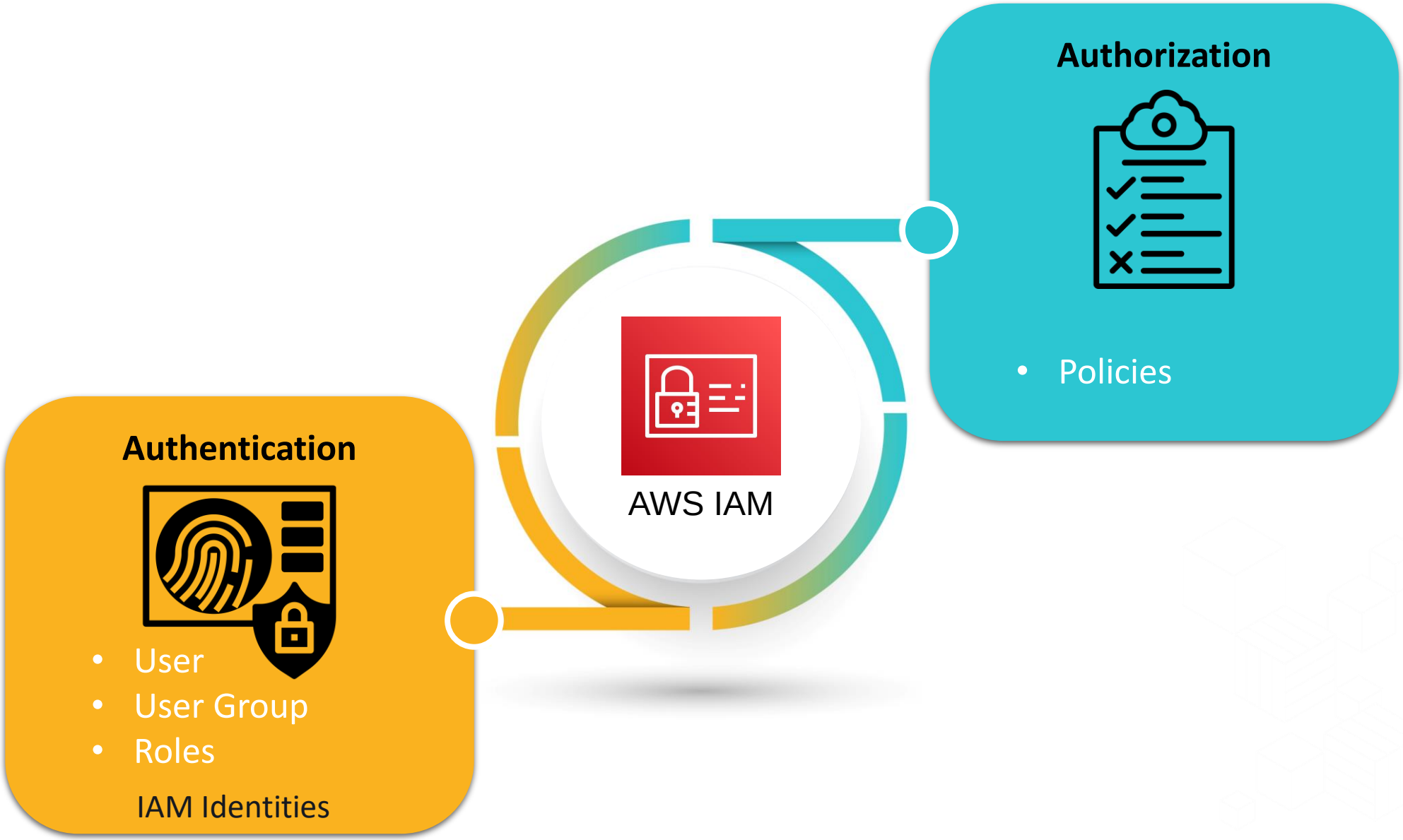
Flow of a request to AWS



Traveling to another country

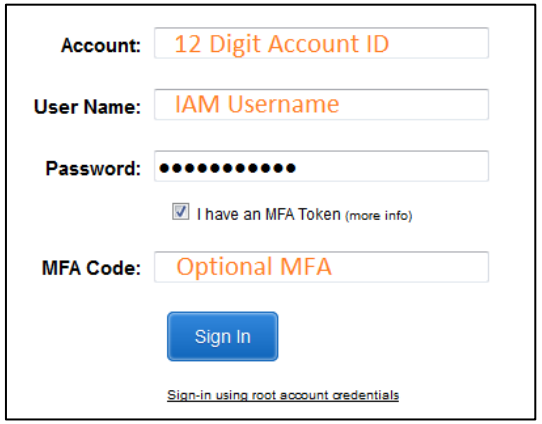
- Visa (Tourist visa / Work visa)
  - What can you do?
  - (Authorization)
- Passport
  - Who you are?
  - (Authentication)

AWS IAM


Authorization
Policies
AWS IAM
Authentication
User
User Group
Roles
IAM Identities

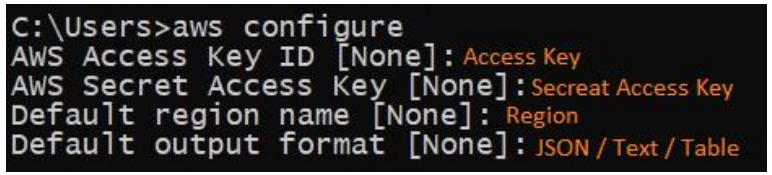
Accessing AWS Services

IAM Username and Password

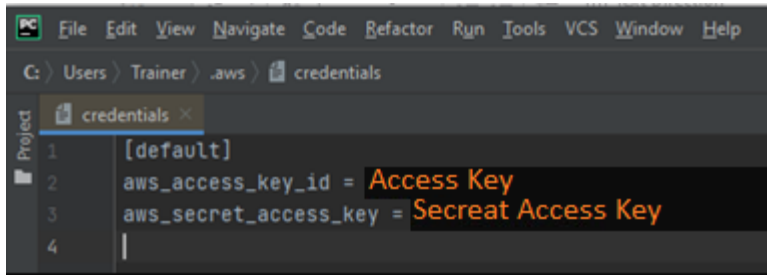


AWS Management Console

Access Key and Secret Access Key



AWS Command Line Interface



AWS Tools and SDKs

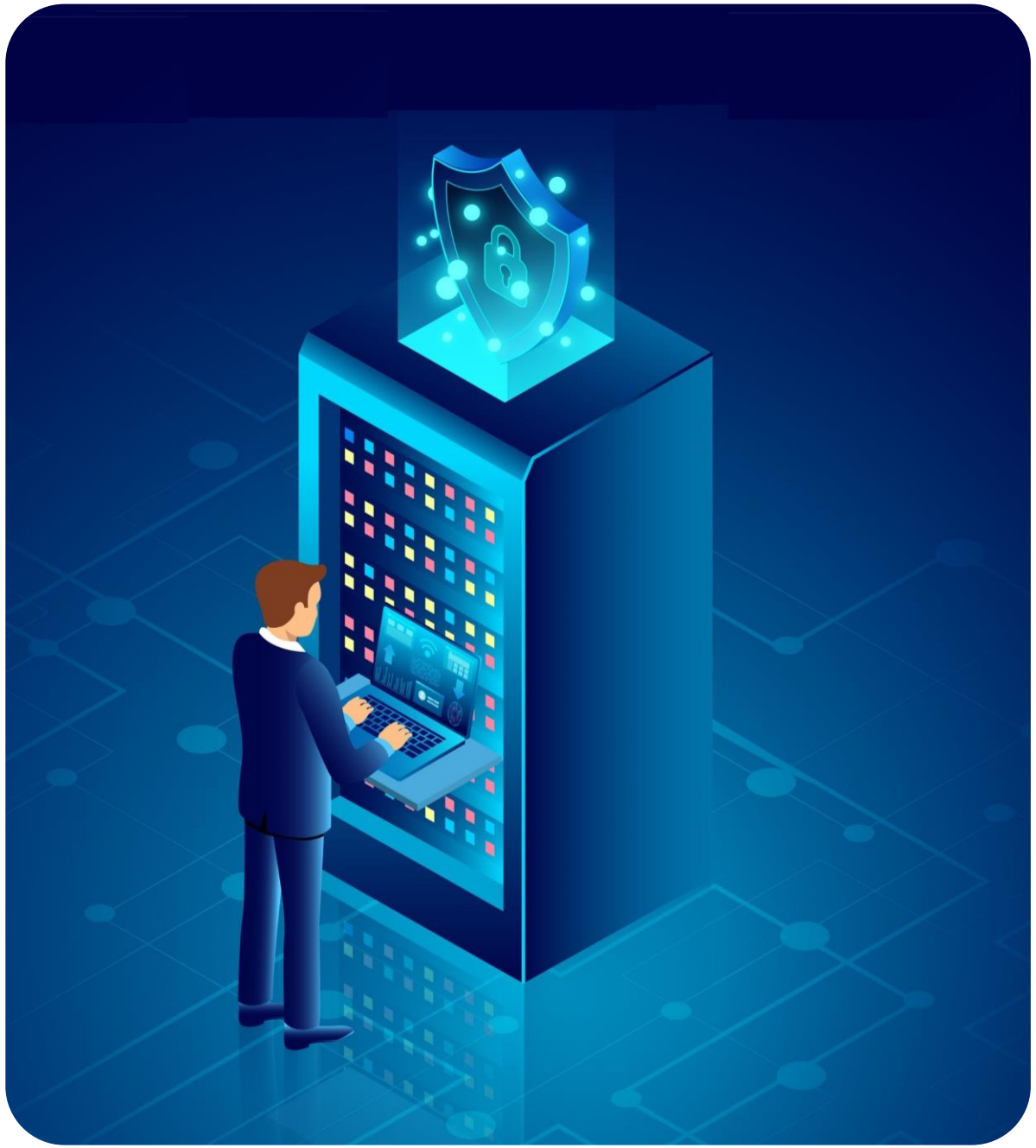
API
Application Programming Interface

AWS

Using IAM

- Business Requirement

User Group	User	Permission
Developers	D1 and D2	• Can use Amazon S3 service • Can use Amazon EC2 service
QA	Q1 and Q2	• Can only read Amazon S3 objects • Can only deploy t2.micro EC2 instance in Ireland region
	Admin007	• Full Admin Permission



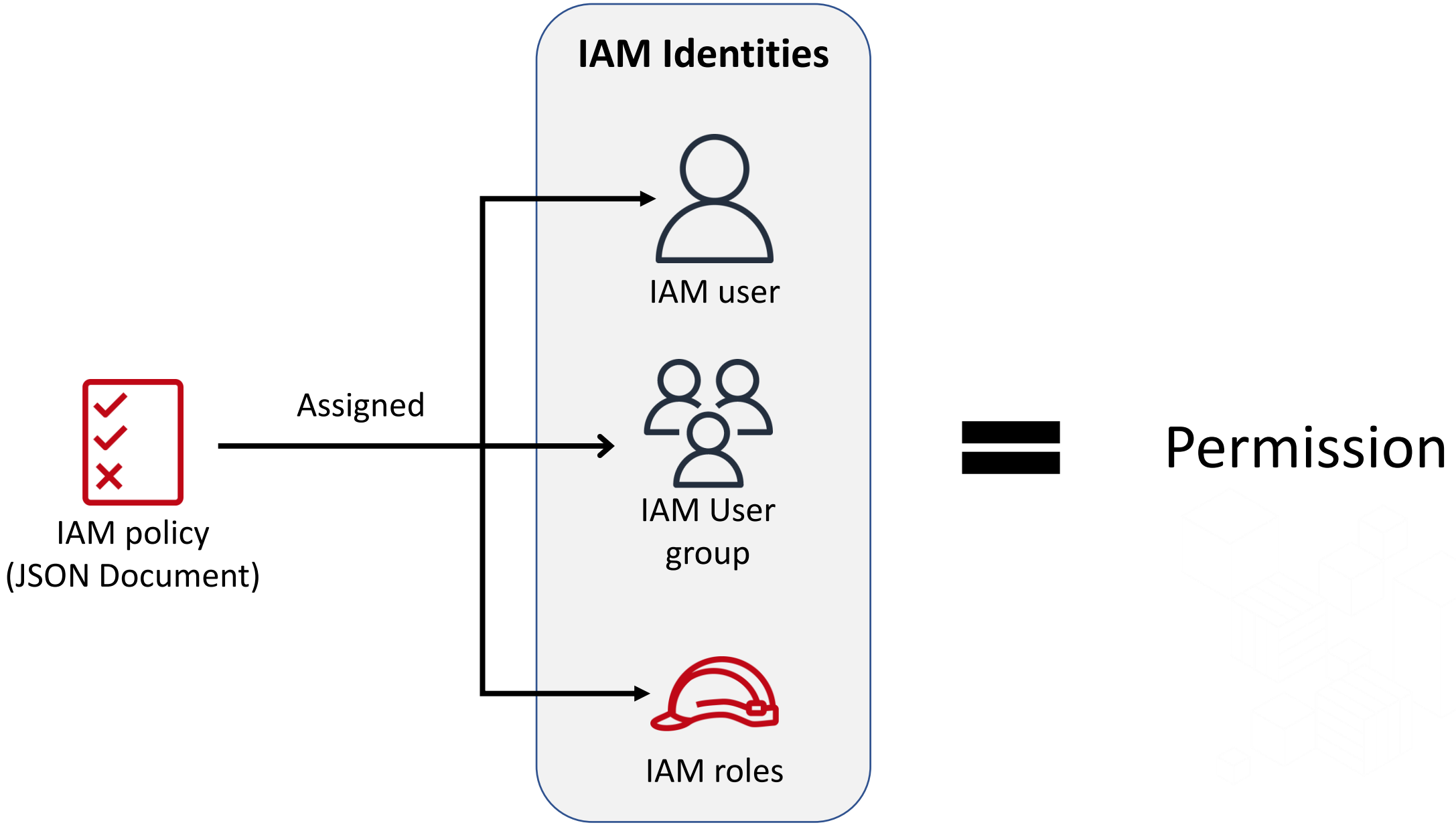
User, UserGroup
and Policies

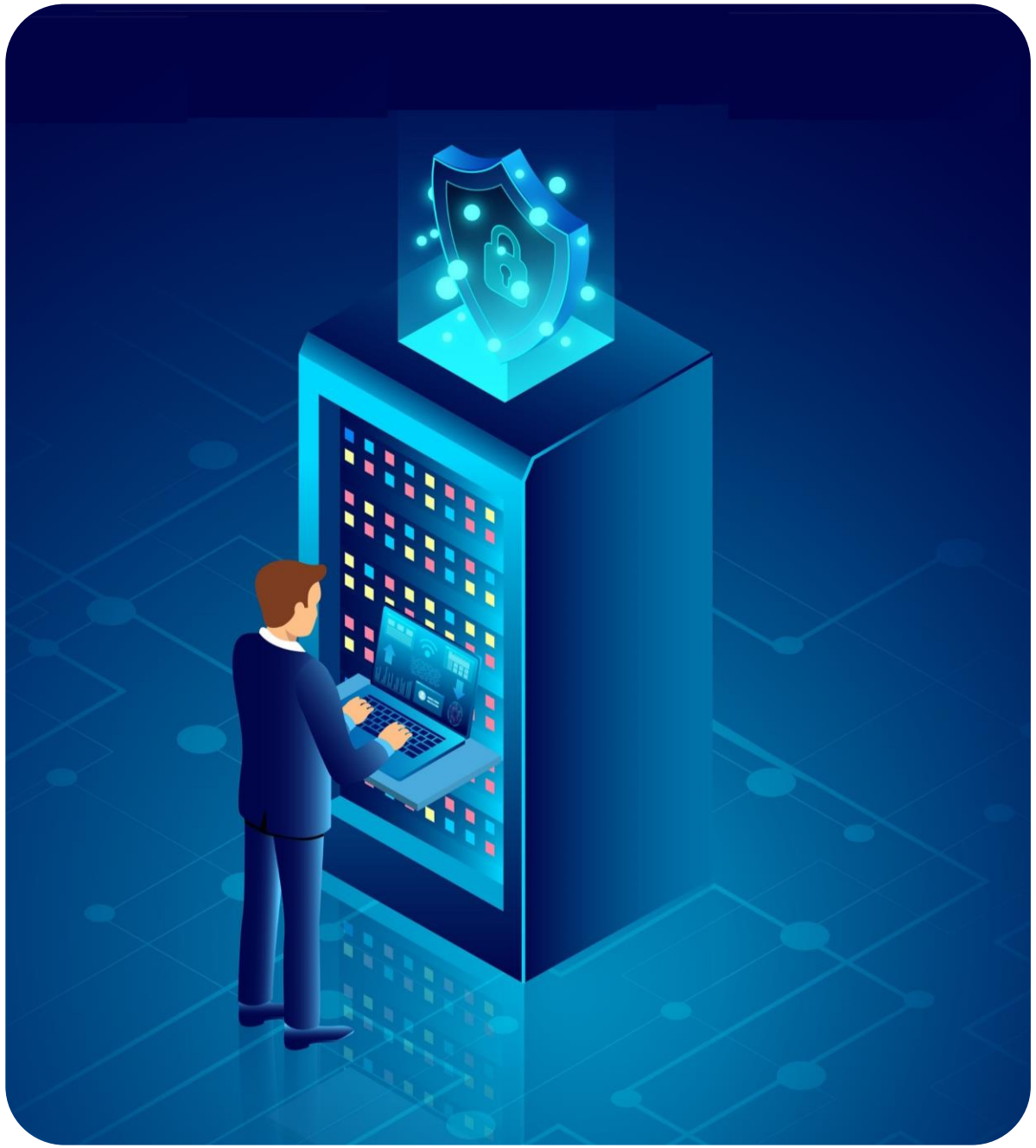
Using IAM

- Business Requirement

User Group	User	Permission
Developers	D1 and D2	• Can use Amazon S3 service • Can use Amazon EC2 service
QA	Q1 and Q2	• Can only read Amazon S3 objects • Can only deploy t2.micro EC2 instance in Ireland region
	Admin007	• Full Admin Permission

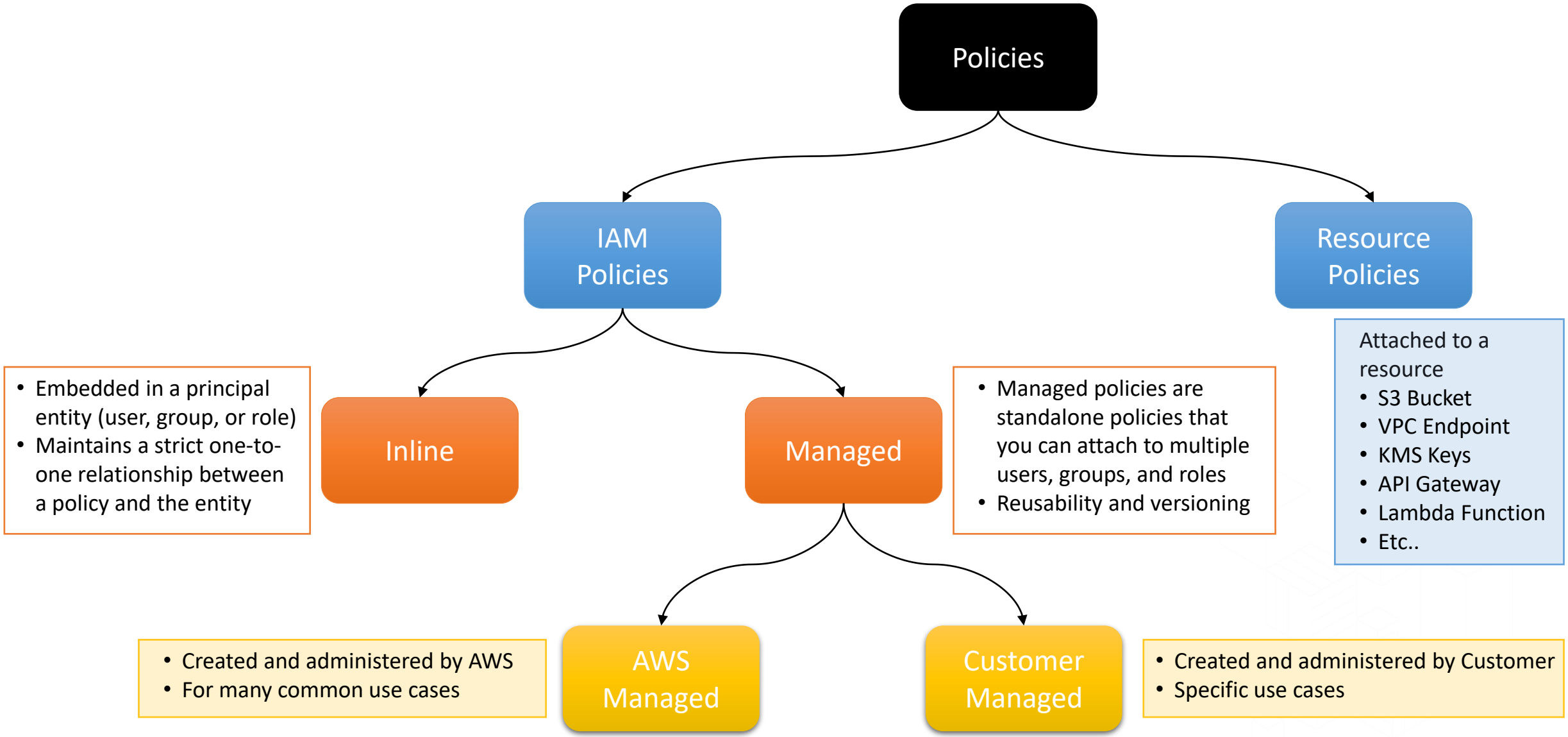
Permission


IAM Identities
IAM user
IAM User group
IAM roles
Assigned
IAM policy
(JSON Document)
Permission



Policies

IAM and Resource Policies



Going to a theater

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

Resource Policy
(Associated to resource)

IAM Policy
(Associated to identity)

Policies

Resource Policy - S3 Bucket Policy

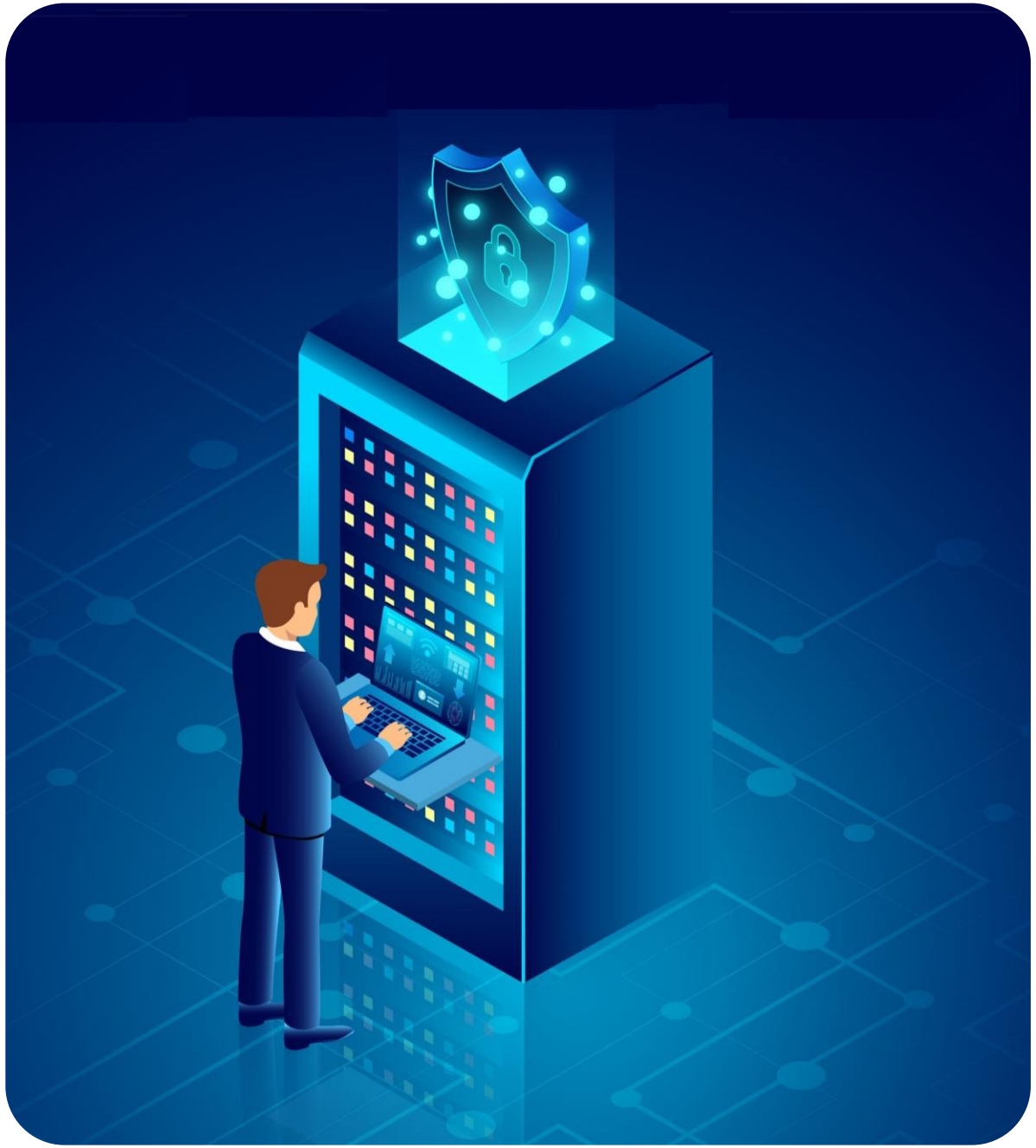
```
{
  "Id": "Policy1676316462609",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1676316460592",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::bucket/*",
      "Principal": "*"
    }
  ]
}
```

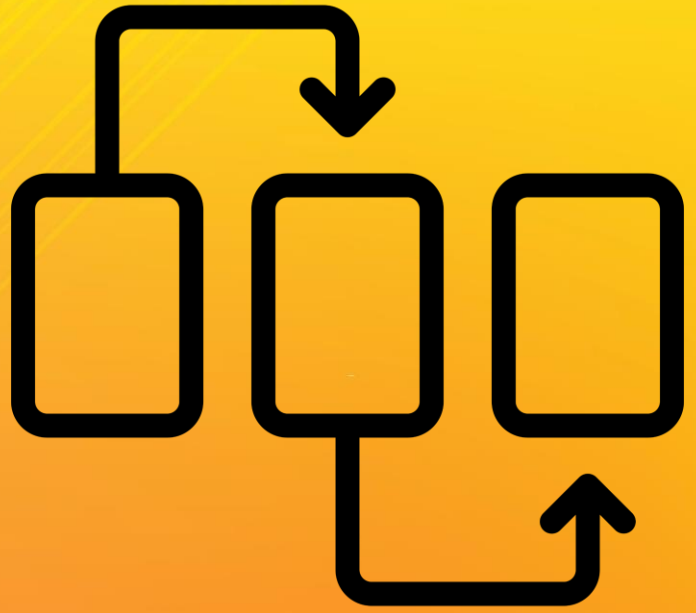
Resource Policy - API Gateway Policy

```
{
  "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

Passenger boarding

Is this person on No-Fly List?

Yes

Explicit Deny

Deny boarding

No

Is this person having valid travel documents?

Yes

Explicit Allow

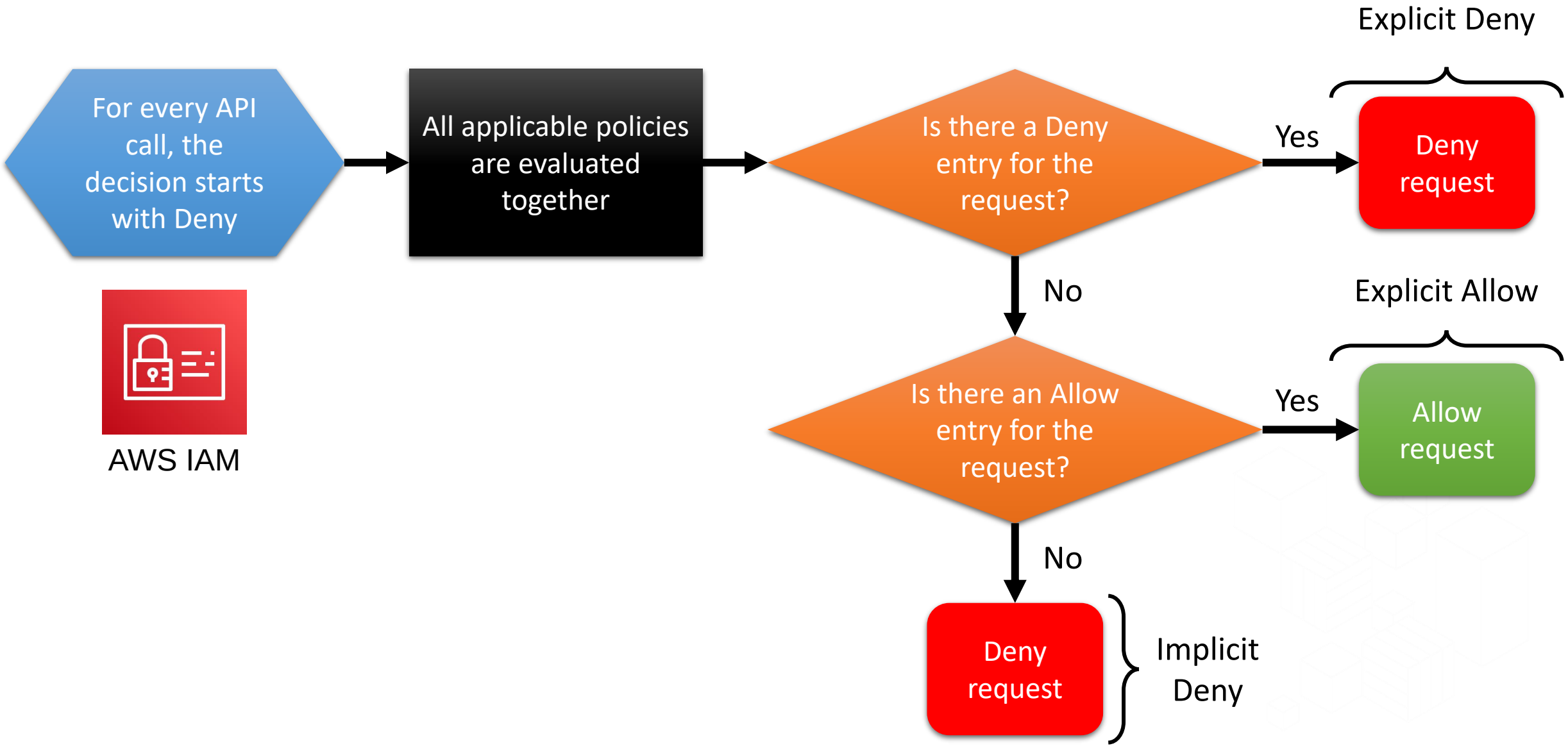
Allow boarding

No

Deny boarding

Implicit Deny

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

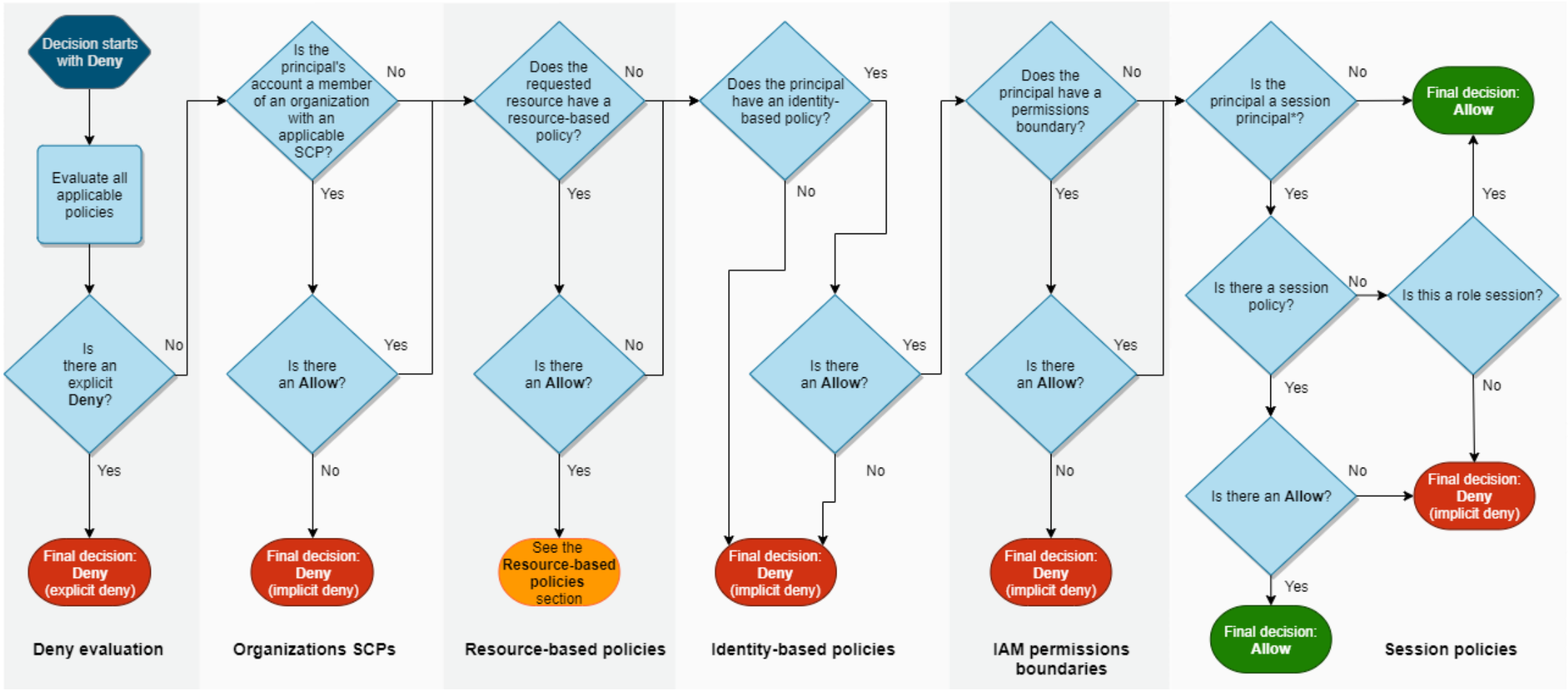
Policy Evaluation Logic

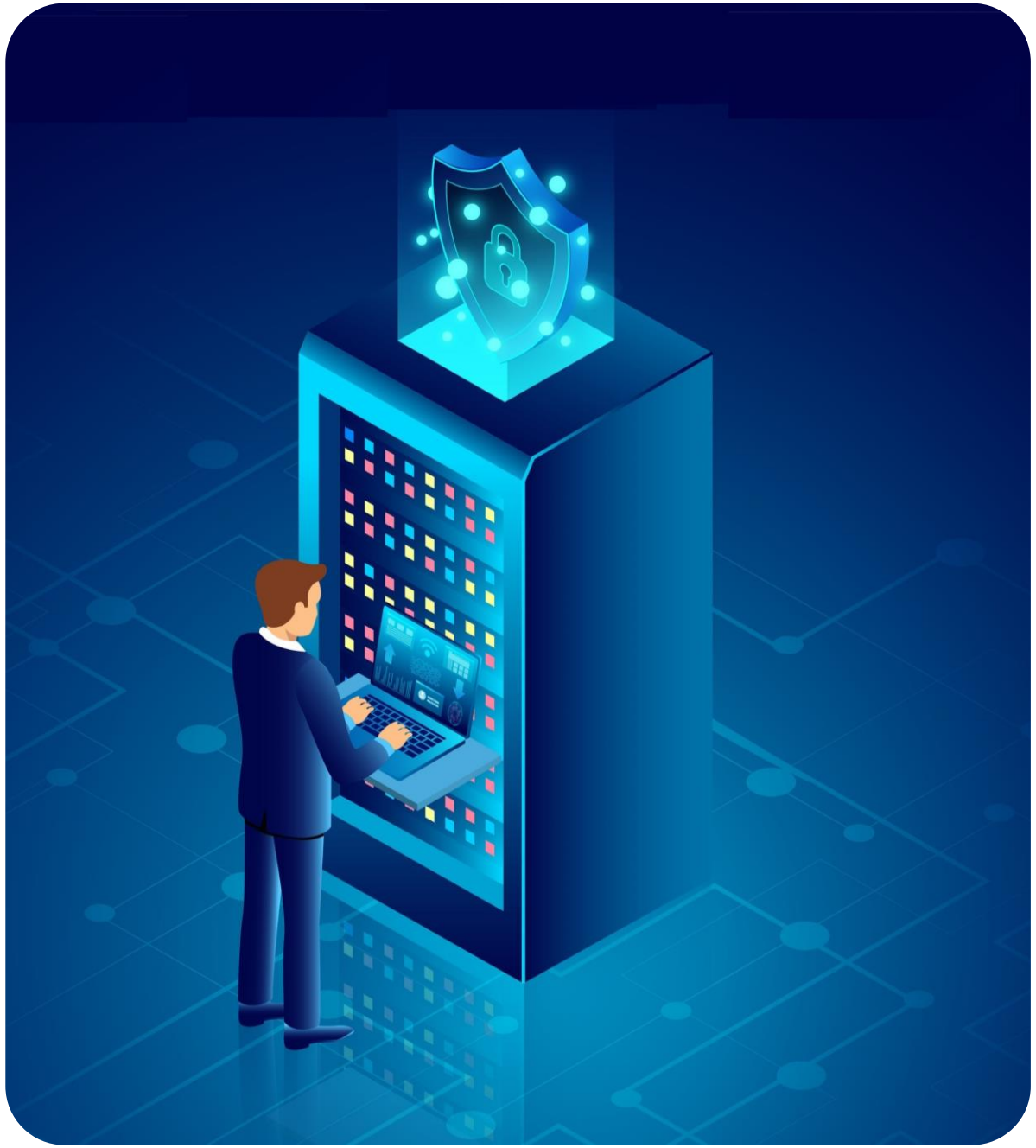
Step 1
- For every API call all applicable policies are combined.

Step 2
- If there is a matching Deny statement – decision is **Deny**.

Step 3
- If there is a matching Allow statement – decision is **Allow**.

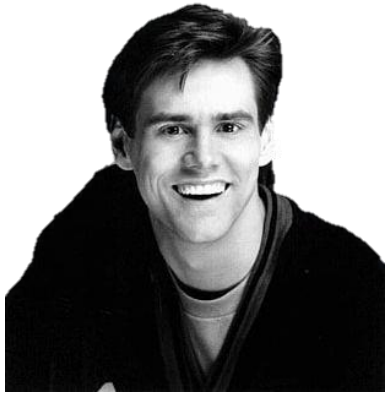
Step 4
- If there is no matching statement – decision in **Deny**.





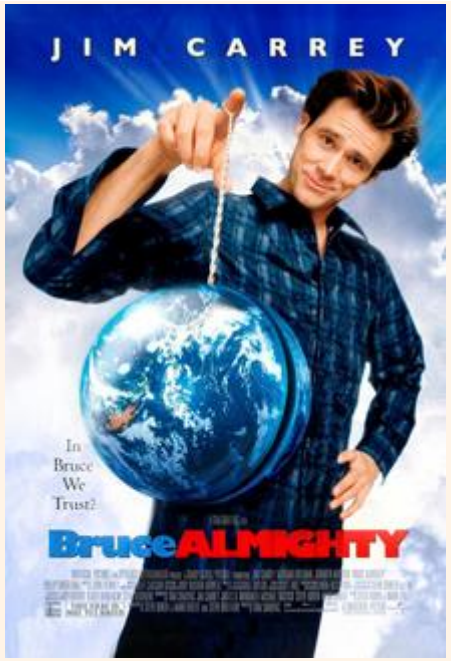
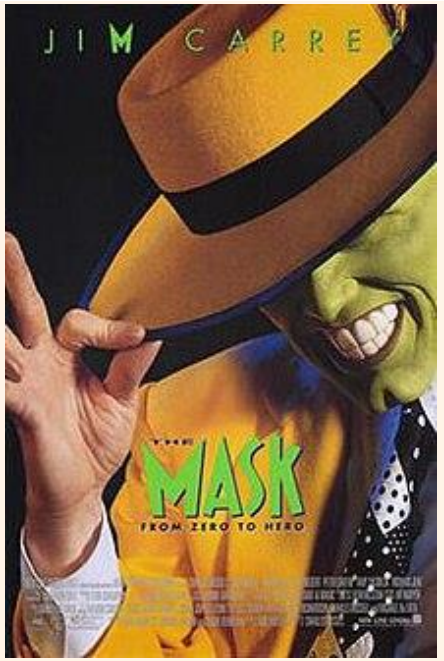
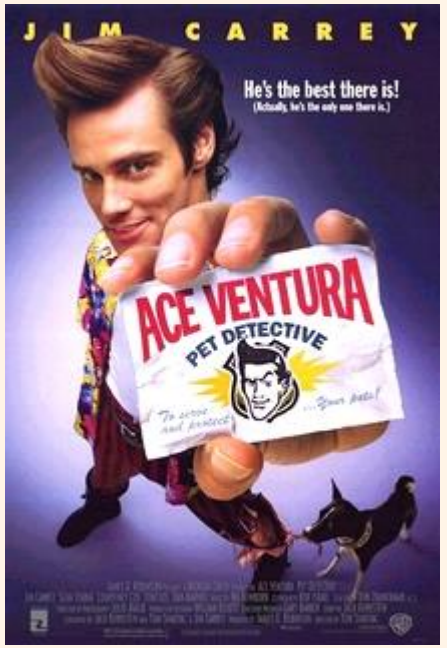
Roles

Understanding Role



Jim Carrey

Canadian-American Actor

Bruce Almighty	The Mask	Ace Ventura
		
Role **God**	Role **The Green Thing**	Role **Pet Detective**

From: https://en.wikipedia.org/wiki/Jim_Carrey

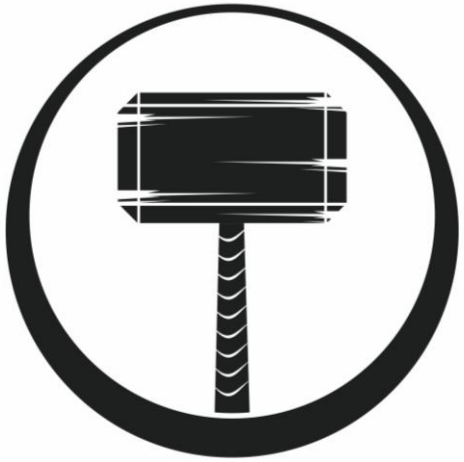
Assuming a role

https://cinespia.org/event/the-mask/

https://www.youtube.com/watch?v=GEVfiRbYA8o

Trust Policy

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



Mjölnir trusts Thor

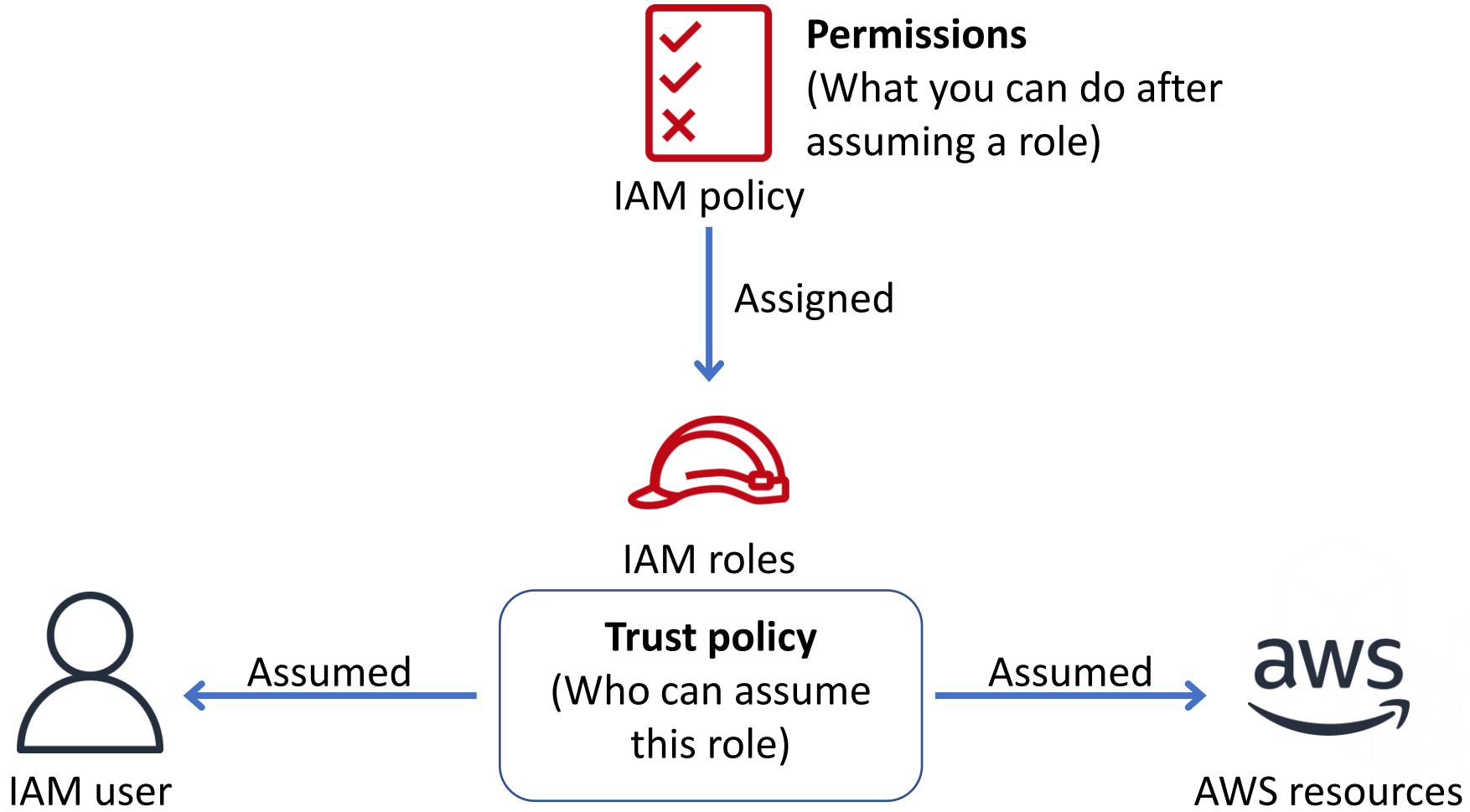
Business Requirement

- Allow Q1 user to assume a role, giving him/her permission equivalent to a developer user

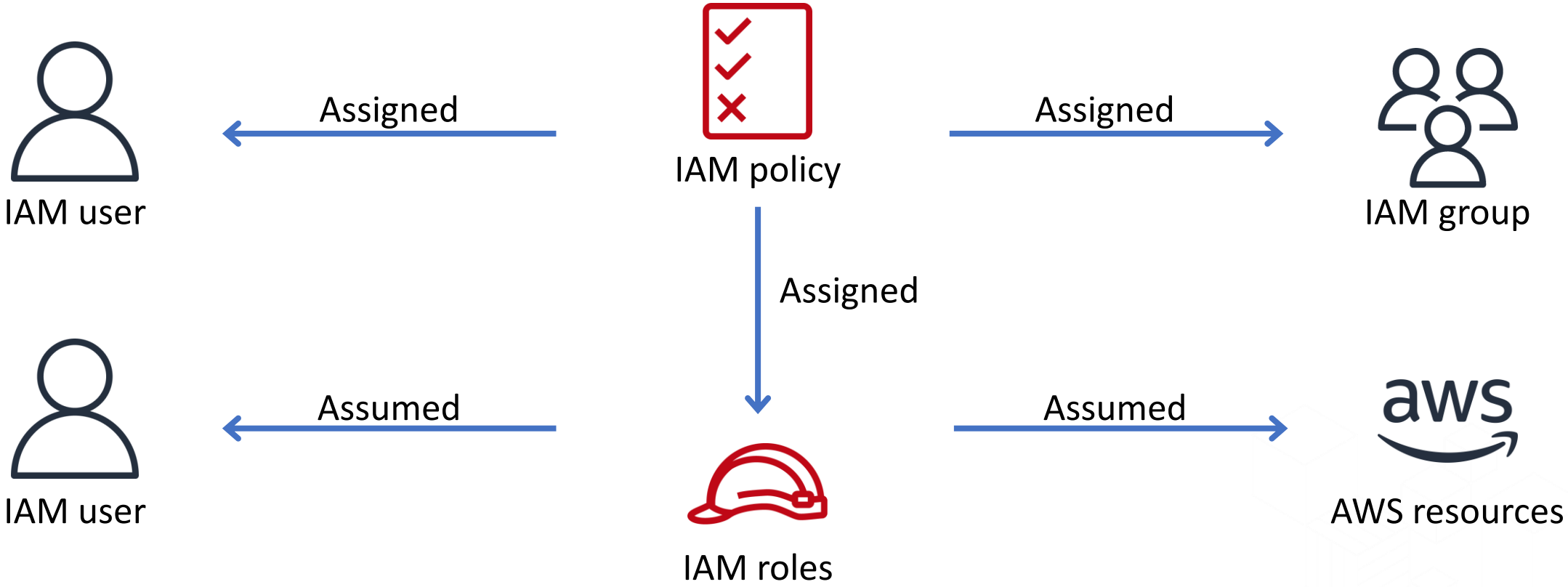
- Three Step Process:

    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


IAM user
Assigned
IAM policy
Assigned
IAM group
Assigned
IAM user
Assumed
IAM roles
Assumed
aws
AWS resources

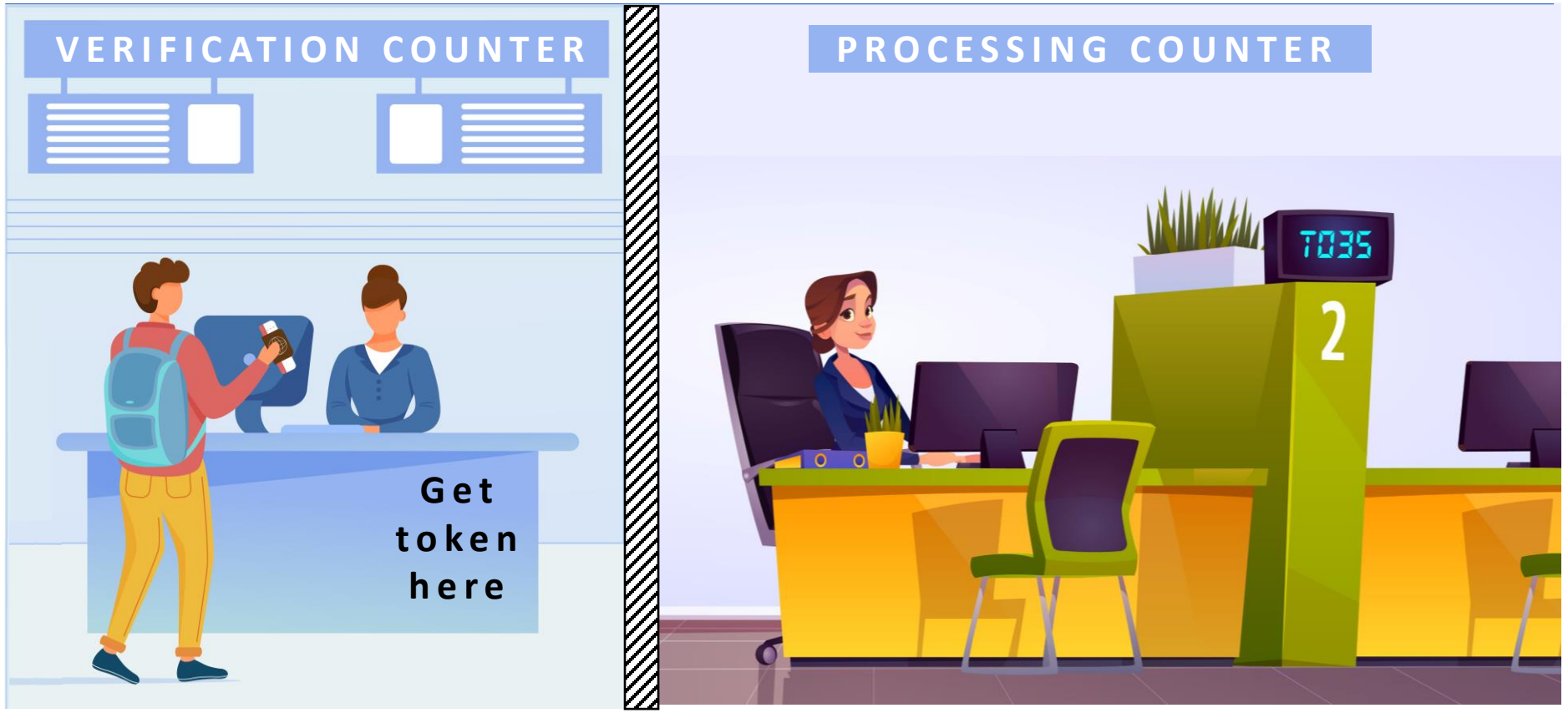
Facts to remember

- 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.

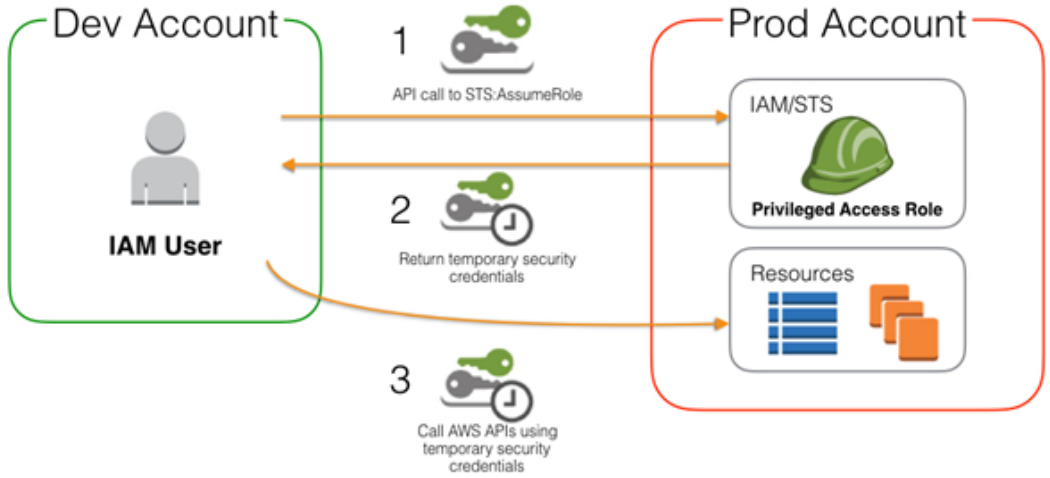
Try Web Identity Federation Playground - 

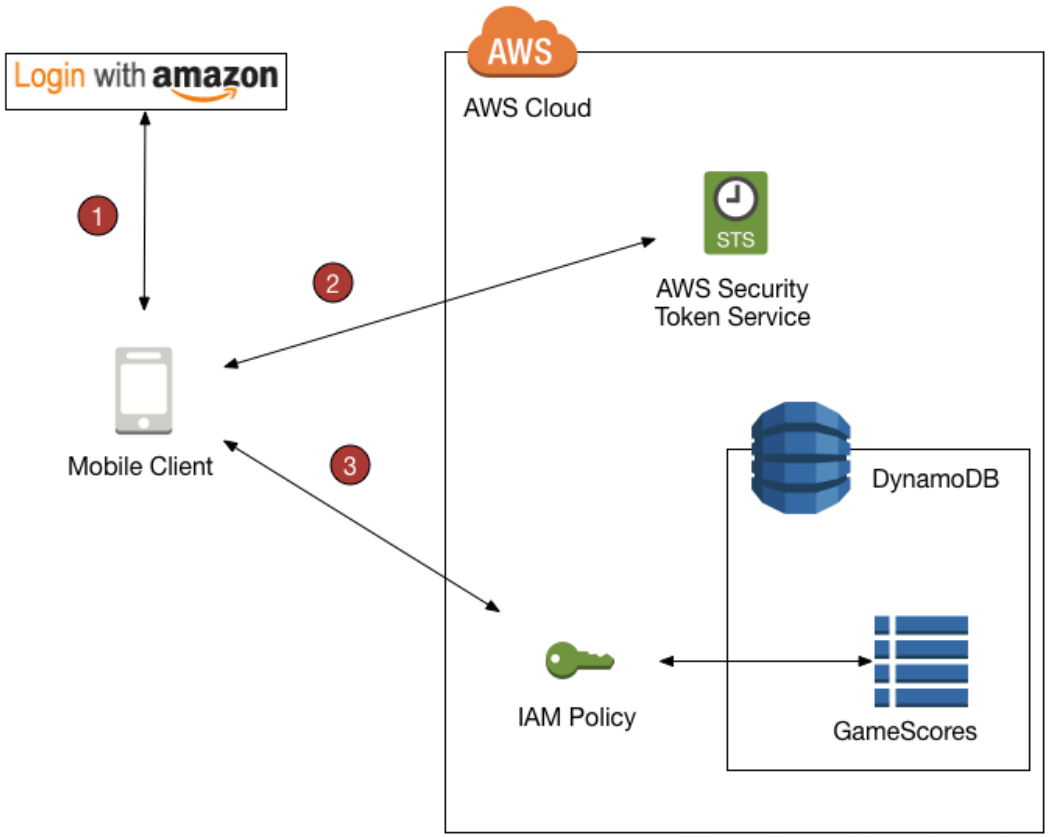
Applying for a visa

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



Amazon Security Token Service





Reference:

Documentation

Category:

Security, Identity, and Compliance

AWS Security Token Service (AWS STS)

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.

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?

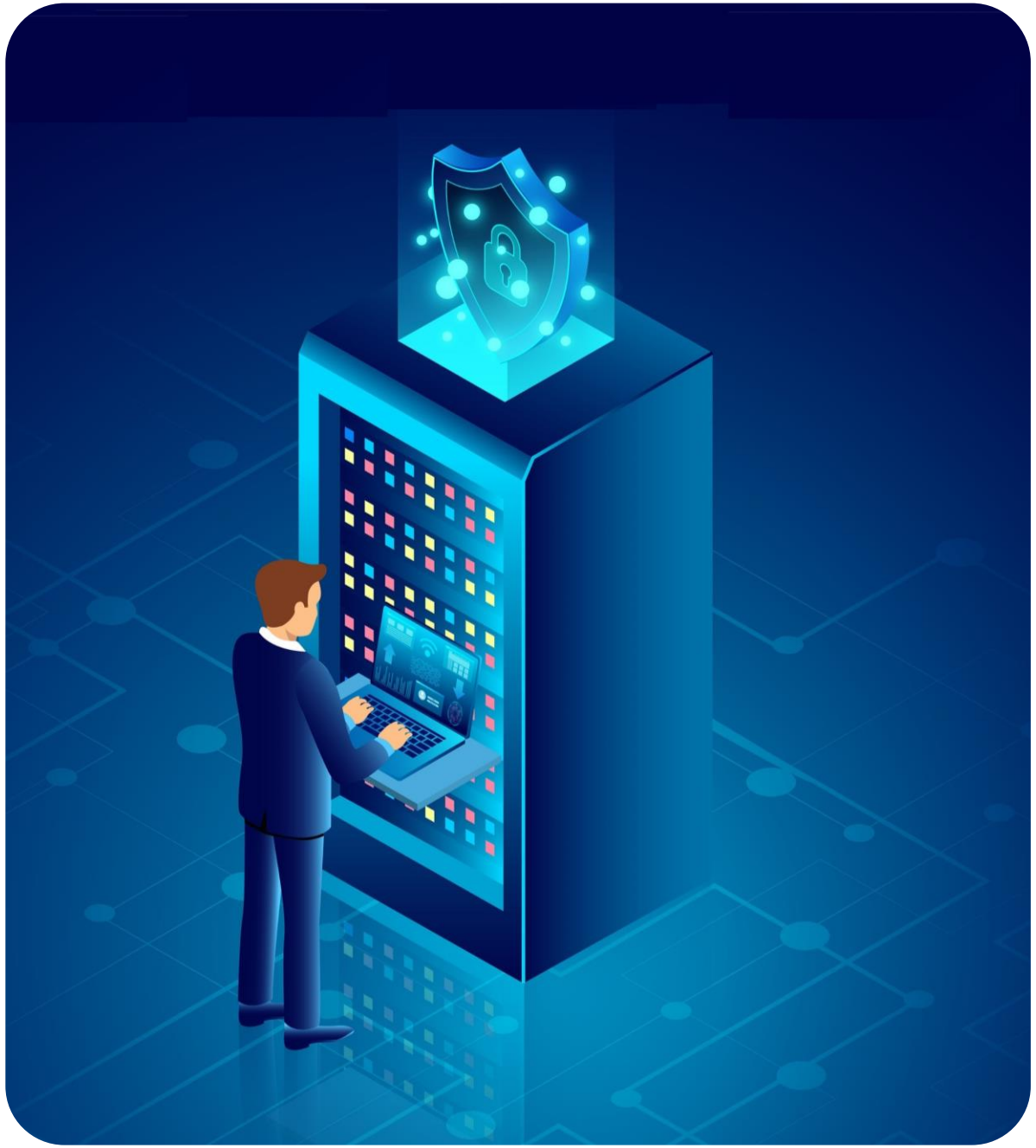
- The AWS STS API operations create a new session with temporary security credentials that include an access key pair (consists of an access key ID and a secret key) and a session token. Users (or an application that the user runs) can use these credentials to access your resources.

Created by:

Ashish Prajapati

How much?

- There is no additional cost to use AWS STS.



Identity Provider

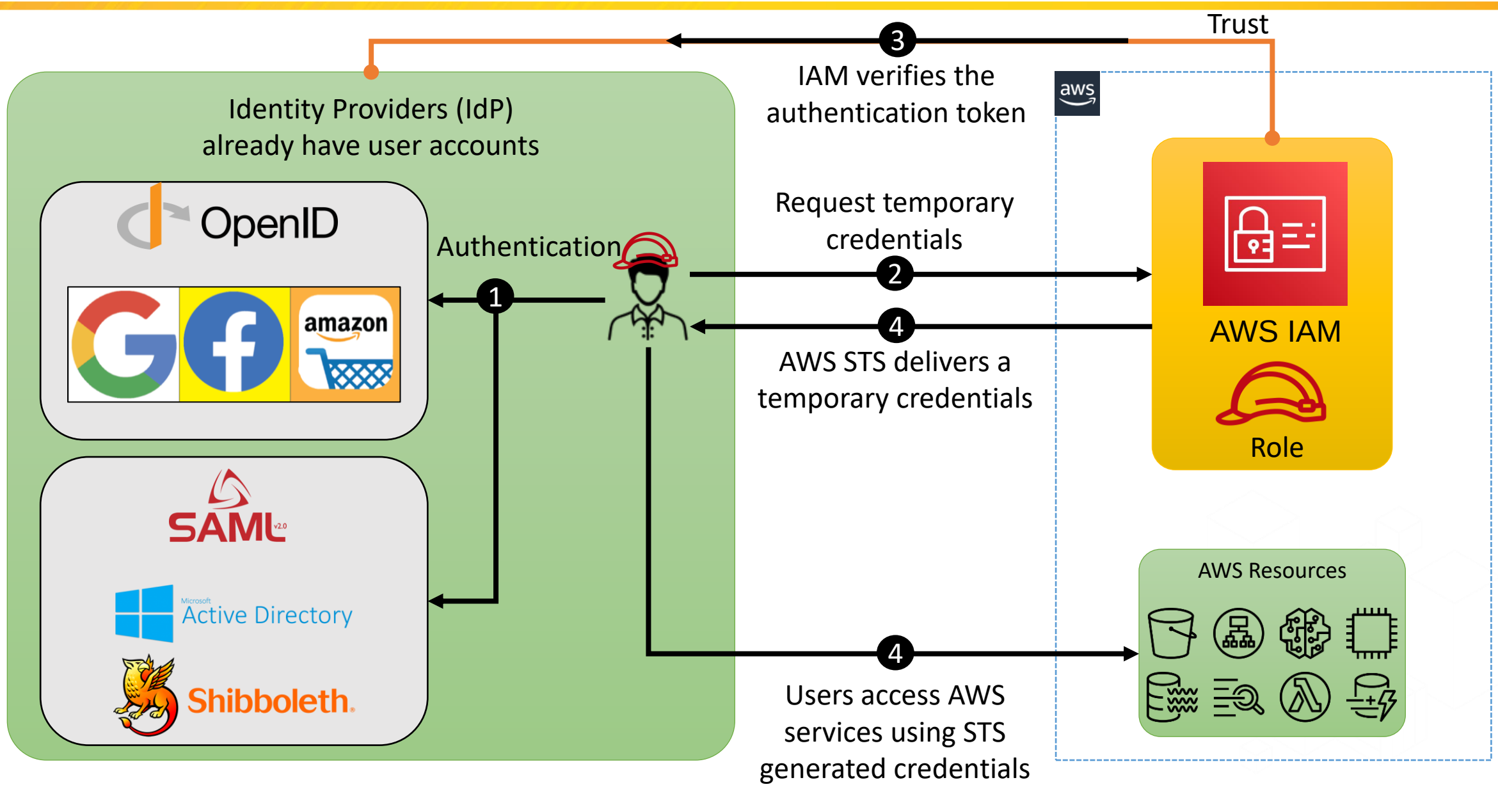
Identity Providers

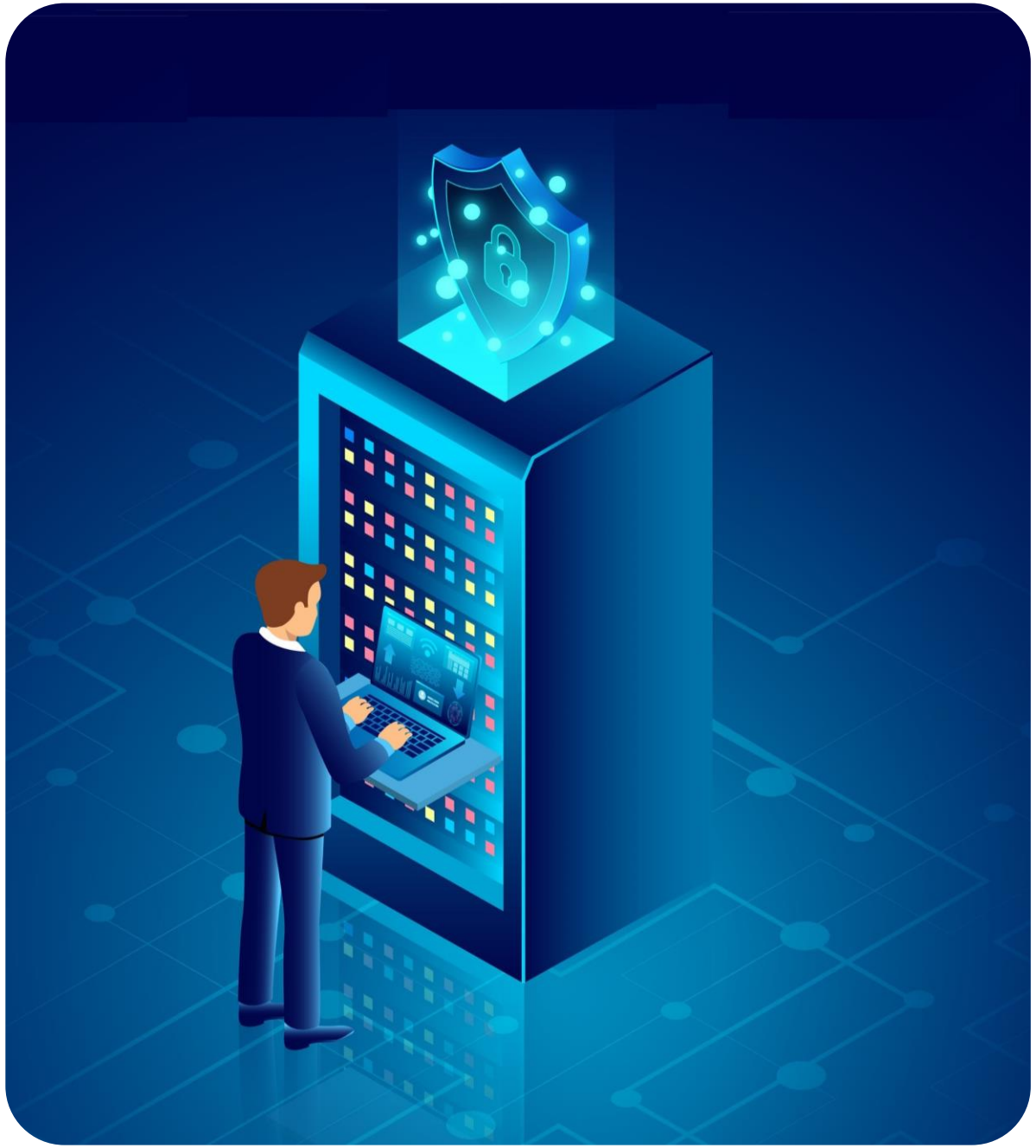
- IAM identity providers (IdPs) allow you to manage your identities outside of AWS.

- 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