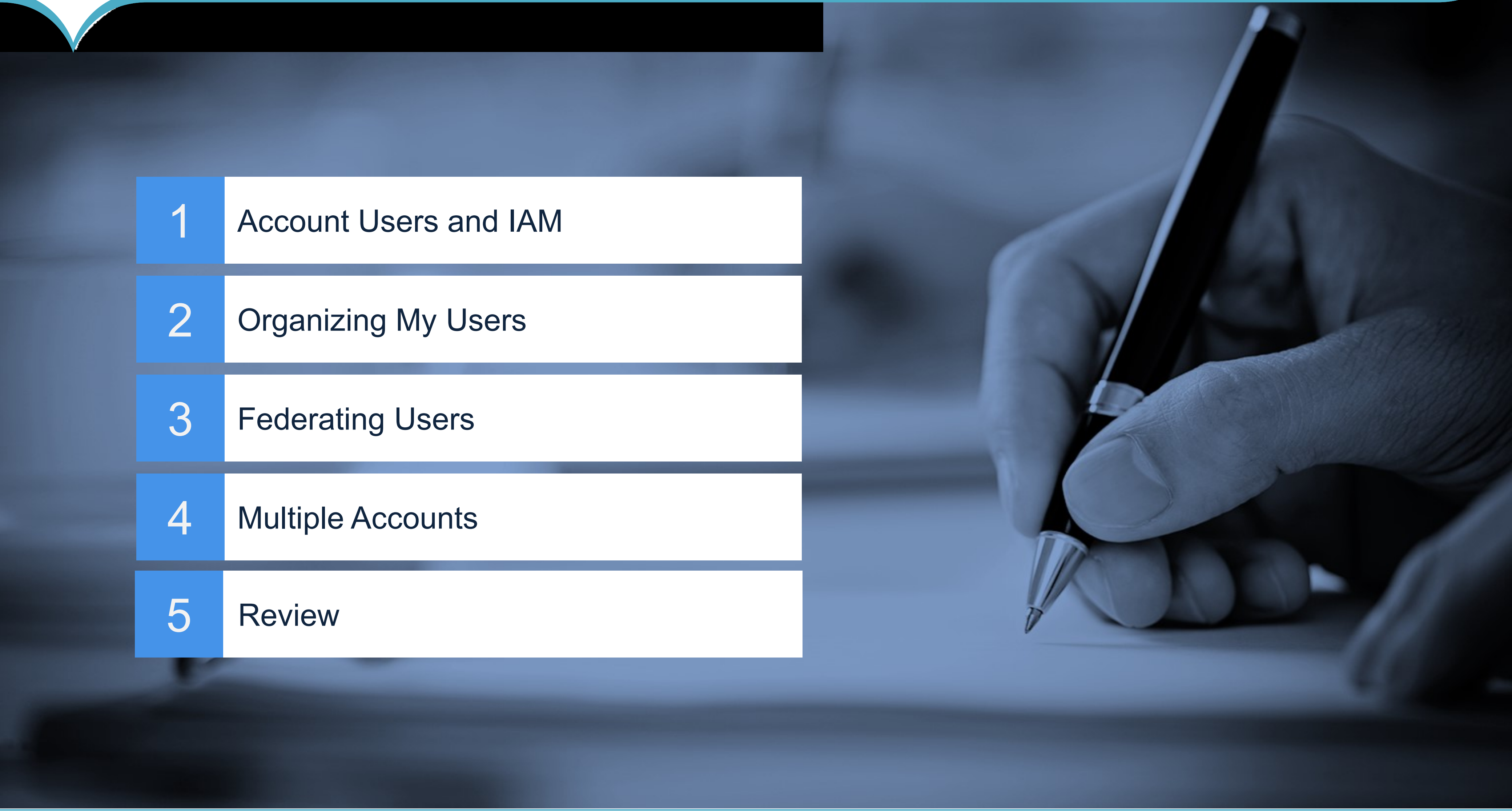
Capgemini
CONSULTING.TECHNOLOGY.OUTSOURCING

AWS Identity Access and Management (IAM)

2019

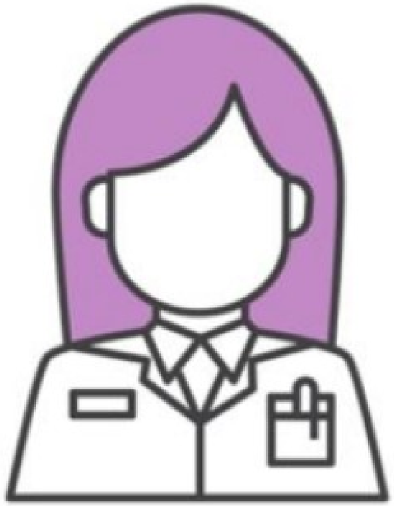
People matter, results count.

Agenda



Account Users and IAM

AWS Account Root User

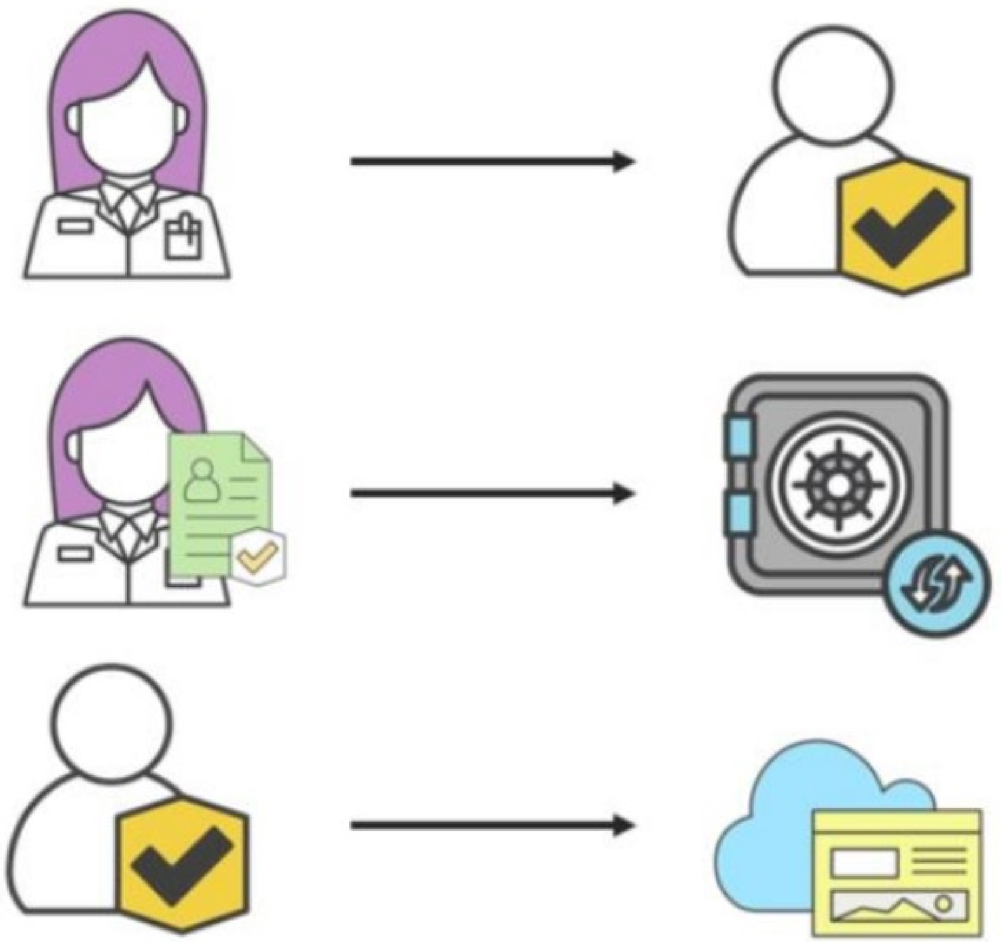


This account has **full** access to **all** AWS services and resources.

- Billing information
- Personal data
- Your entire architecture and its components

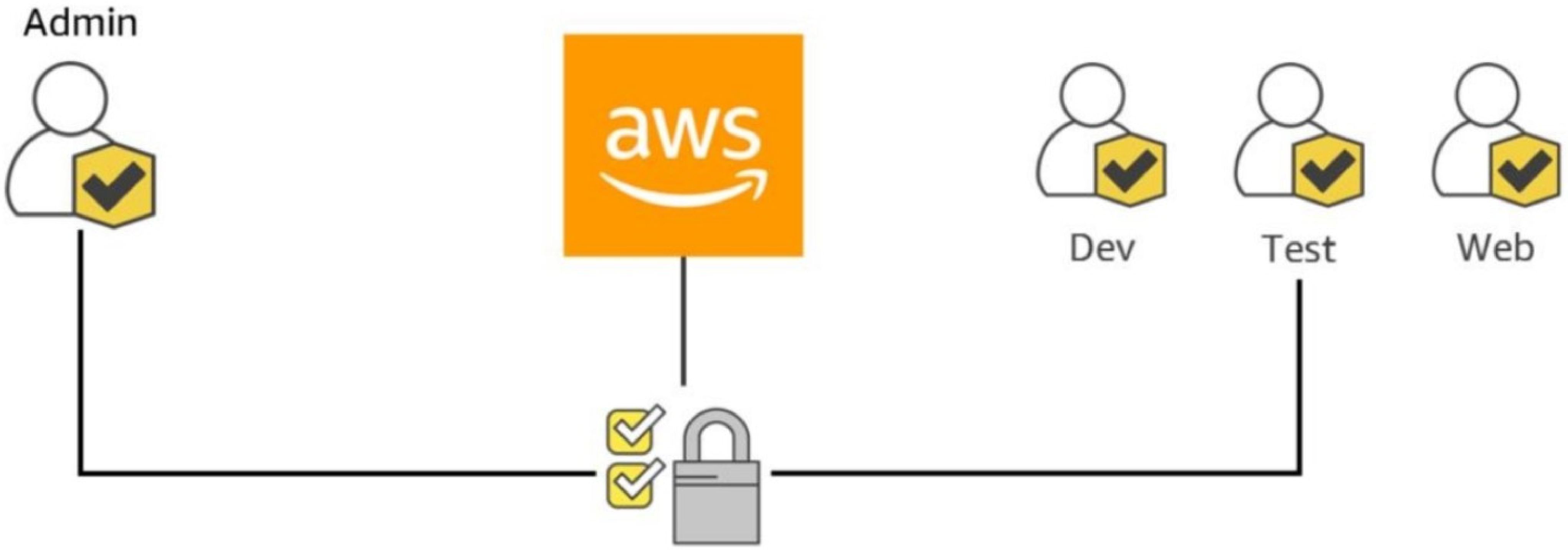
The AWS account root user has *extreme power and* cannot be limited

A Safer Way to a Administer

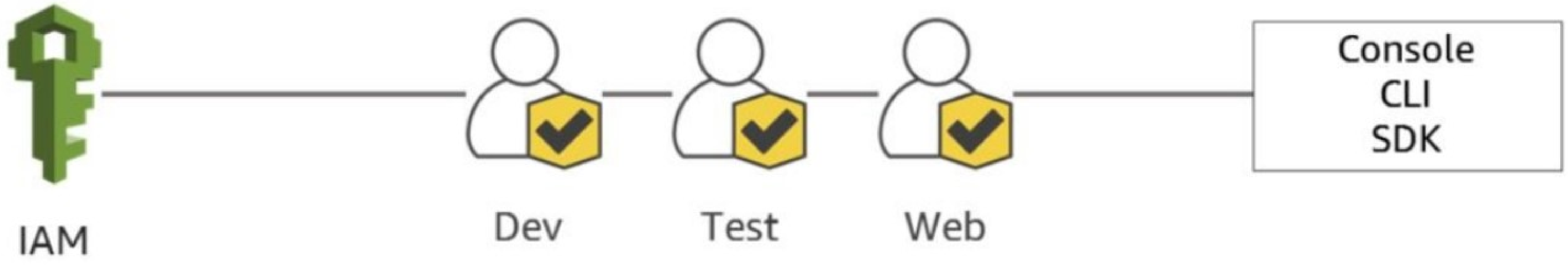


Everybody Wants to Rule the World

Problem: You need to be able to restrict **granularly**

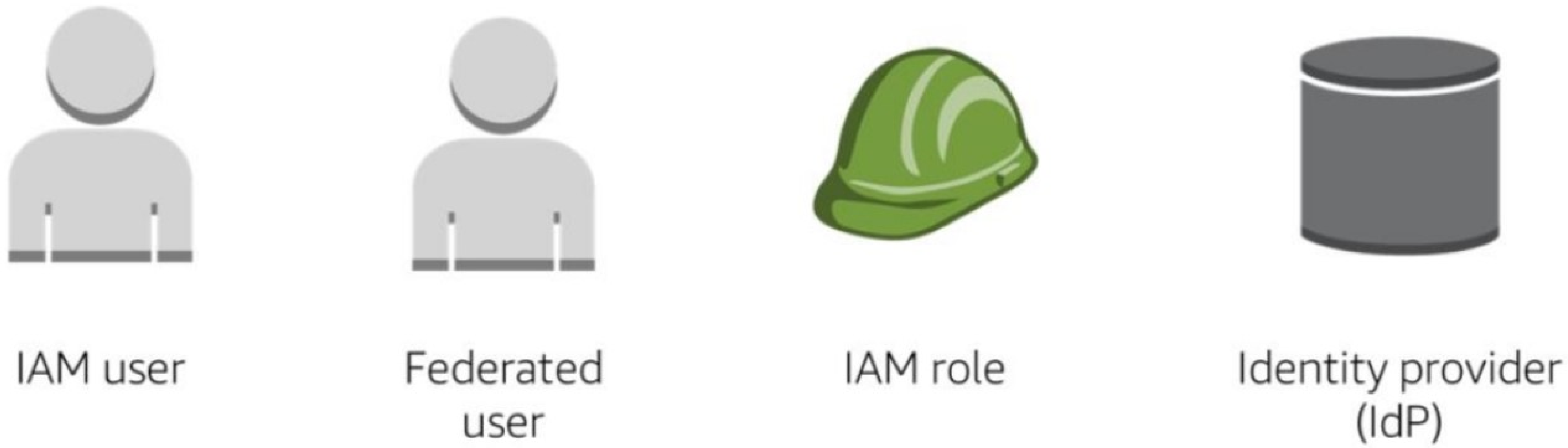


AWS Identity and Access Management



- **Integrates** with other AWS services
- Federated **identity management**
- Secure access for **applications**
- **Granular** permissions

IAM Principals



IAM Users

IAM User

Account ID or alias
IAM user name
Password

IAM users are not separate AWS accounts; they are users **within your account.**

Each user has their **own credentials.**

IAM users are authorized to perform specific AWS actions based on their **permission**.

The Birth of an IAM Users

There are **no default permissions.**

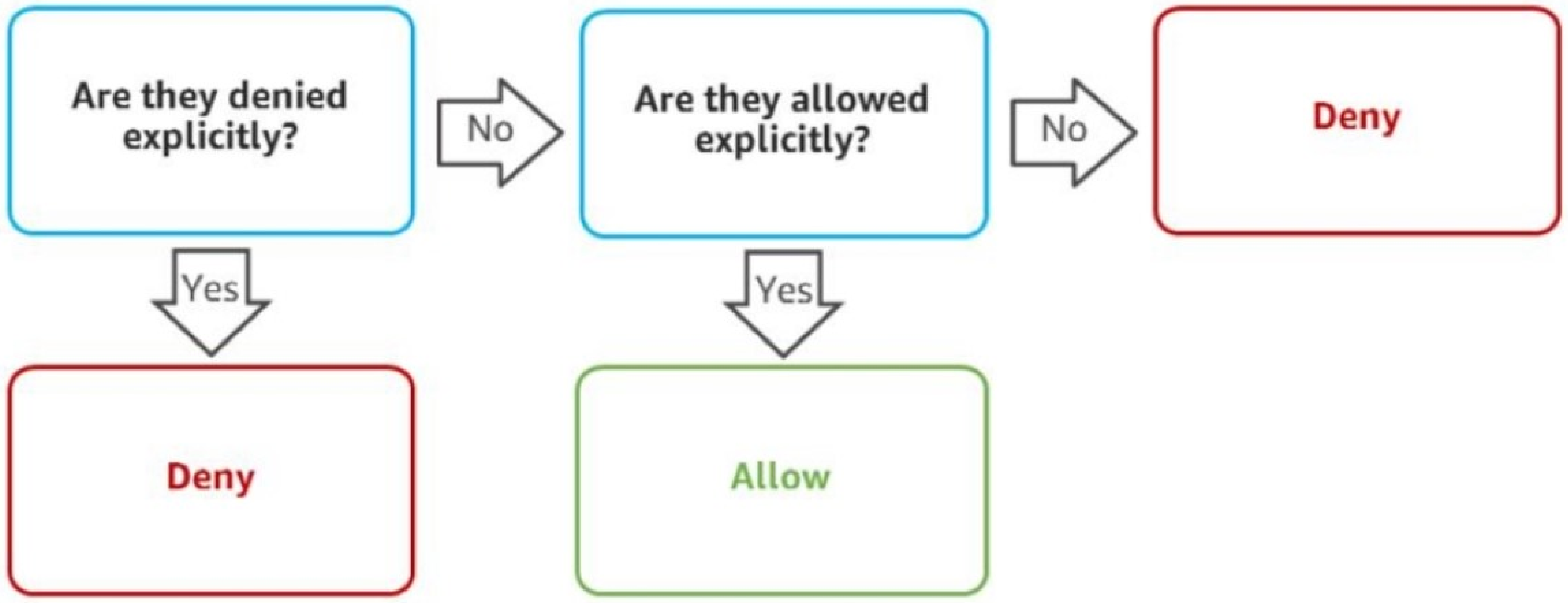
Access to the AWS Management Console or CLI must be **explicitly** granted.

Granting Permissions

- A formal declaration of **one or more permissions**
- Evaluated at the **time of request**
- IAM policies ONLY control access to **AWS services**
- IAM has **no visibility** above the hypervisor

IAM Permissions

How IAM determines permissions:



Granting Permission

- Resource-Based-Attached to an AWS resource
- Identity-Based-Attached to an IAM principal

Identity-Based Policy

Identity-based policy

Attached to:

- User
- Group
- Role

Control:

- Actions performed
- Which resources
- What conditions are required

Types of Policies:

- AWS-managed
- Customer-managed
- Inline

Resource-Based Policies

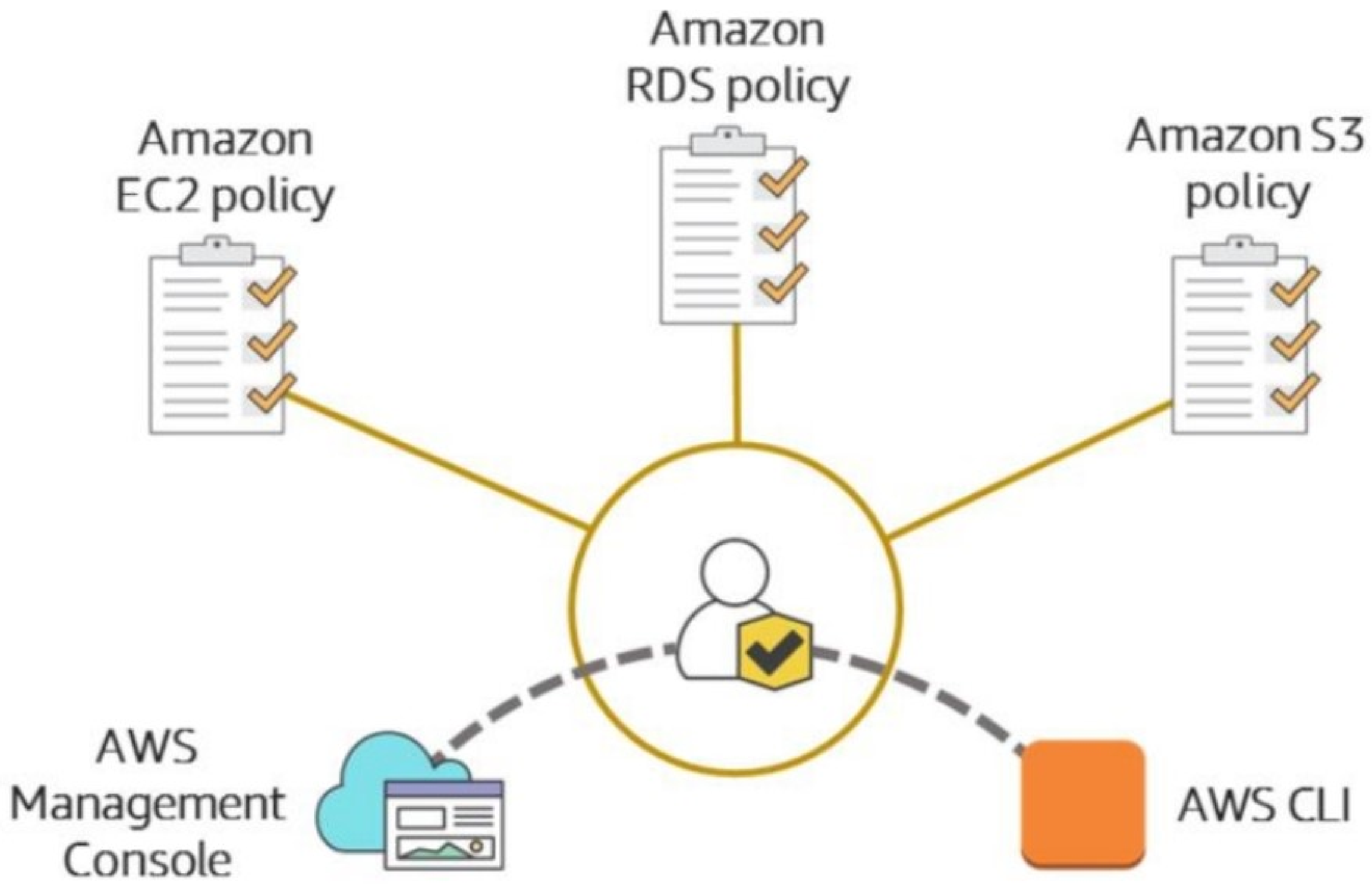
Attached to:

- AWS resources such as Amazon S3, Amazon Glacier and AWS KMS

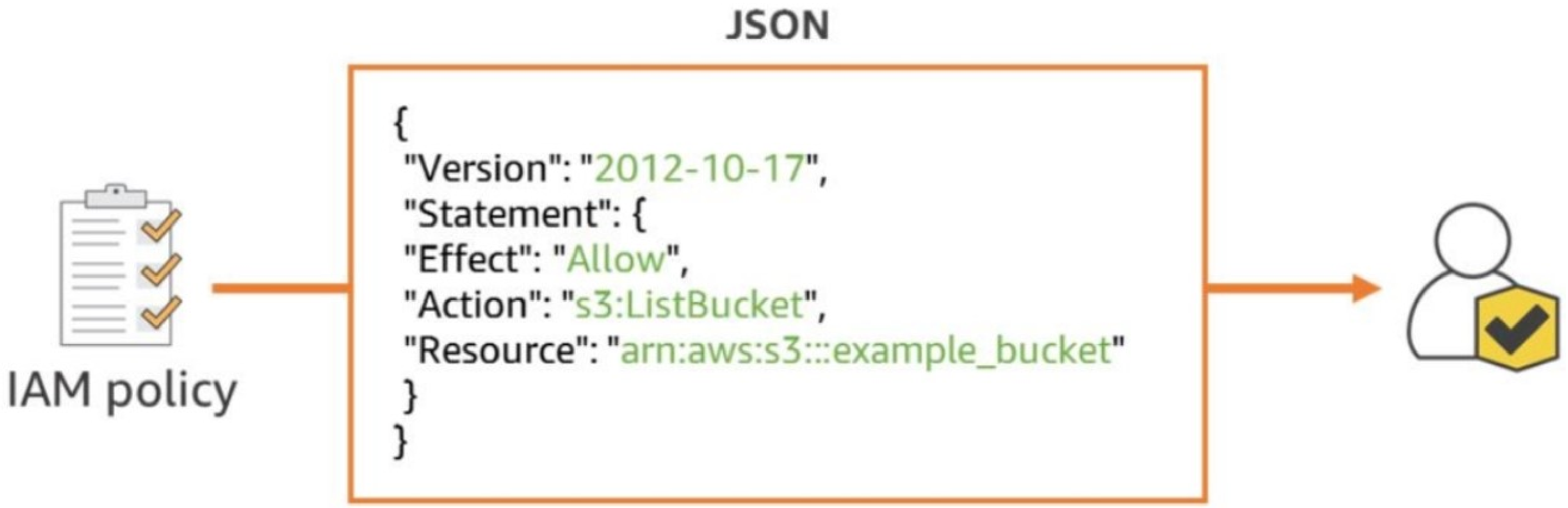
Control:

- Actions allowed by specific principal
- What conditions are required
- Are always inline policies
- No AWS-managed resource-based policies

Identities with Attached Permissions

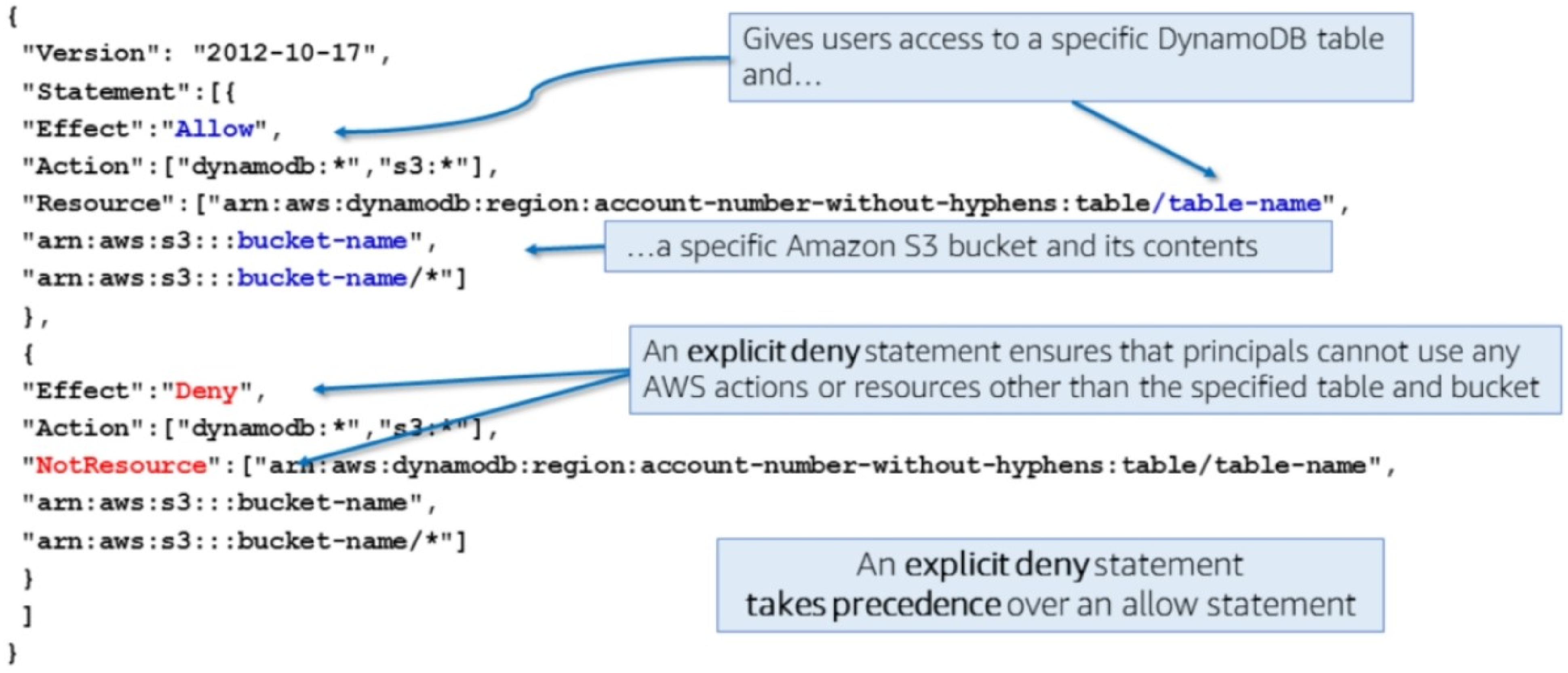


Applying Permissions



```
{
 "Version": "2012-10-17",
 "Statement": {
 "Effect": "Allow",
 "Action": "s3:ListBucket",
 "Resource": "arn:aws:s3:::example_bucket"
 }
}
```

IAM Policy Example



```
{
 "Version": "2012-10-17",
 "Statement":[{
 "Effect":"Allow",
 "Action":["dynamodb:*","s3:*"],
 "Resource":["arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name",
 "arn:aws:s3:::bucket-name",
 "arn:aws:s3:::bucket-name/*"]
 },
 {
 "Effect":"Deny",
 "Action":["dynamodb:*","s3:*"],
 "NotResource":["arn:aws:dynamodb:region:account-number-without-hyphens:table/table-name",
 "arn:aws:s3:::bucket-name",
 "arn:aws:s3:::bucket-name/*"]
 }
 ]
}
```

Gives users access to a specific DynamoDB table and…

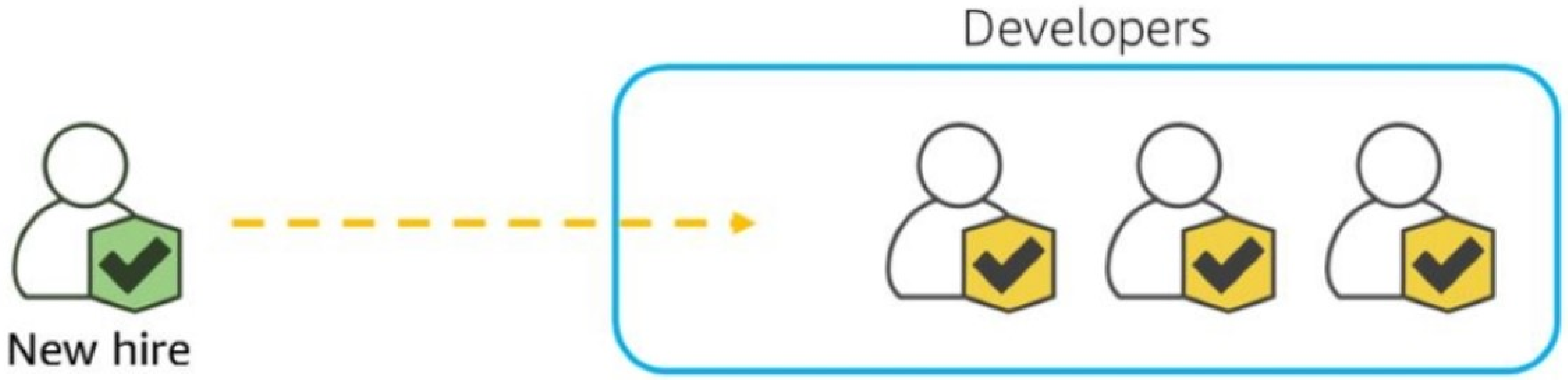
…a specific Amazon S3 bucket and its contents

An **explicit deny** statement ensures that principals cannot use any AWS actions or resources other than the specified table and bucket

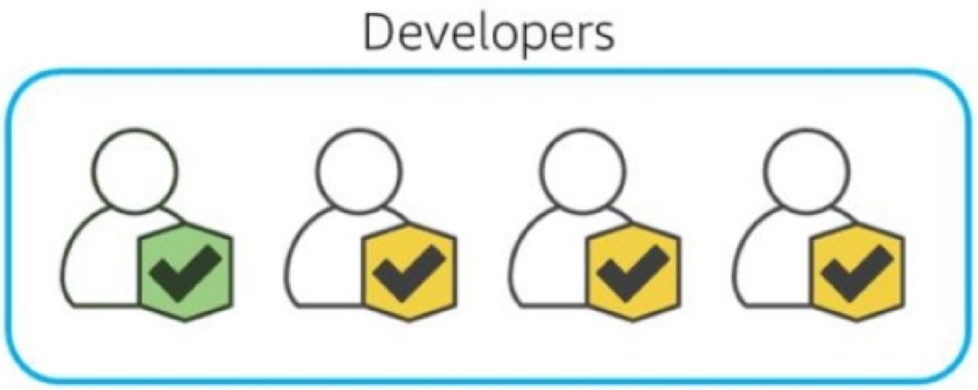
An **explicit deny** statement **takes precedence** over an allow statement

Organizing My Users

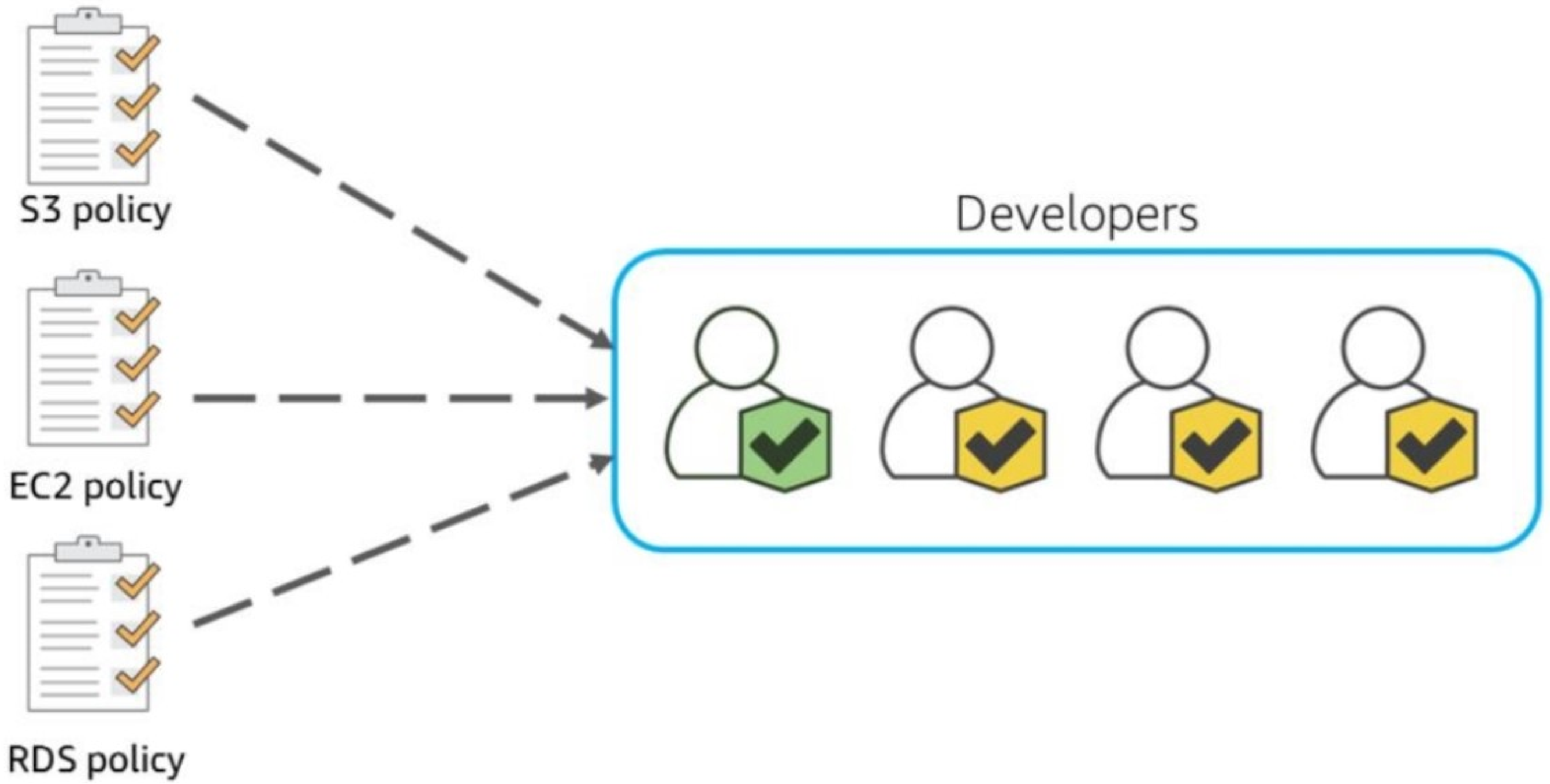
IAM User Group



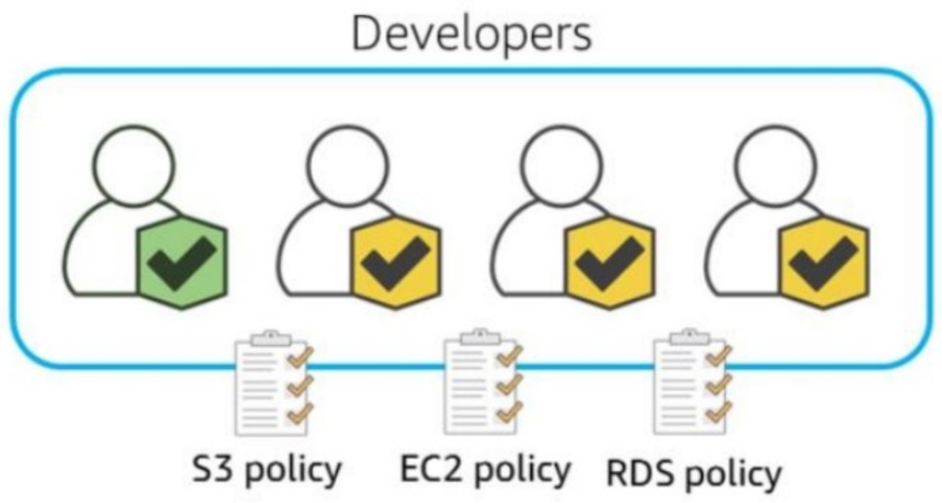
IAM User Group



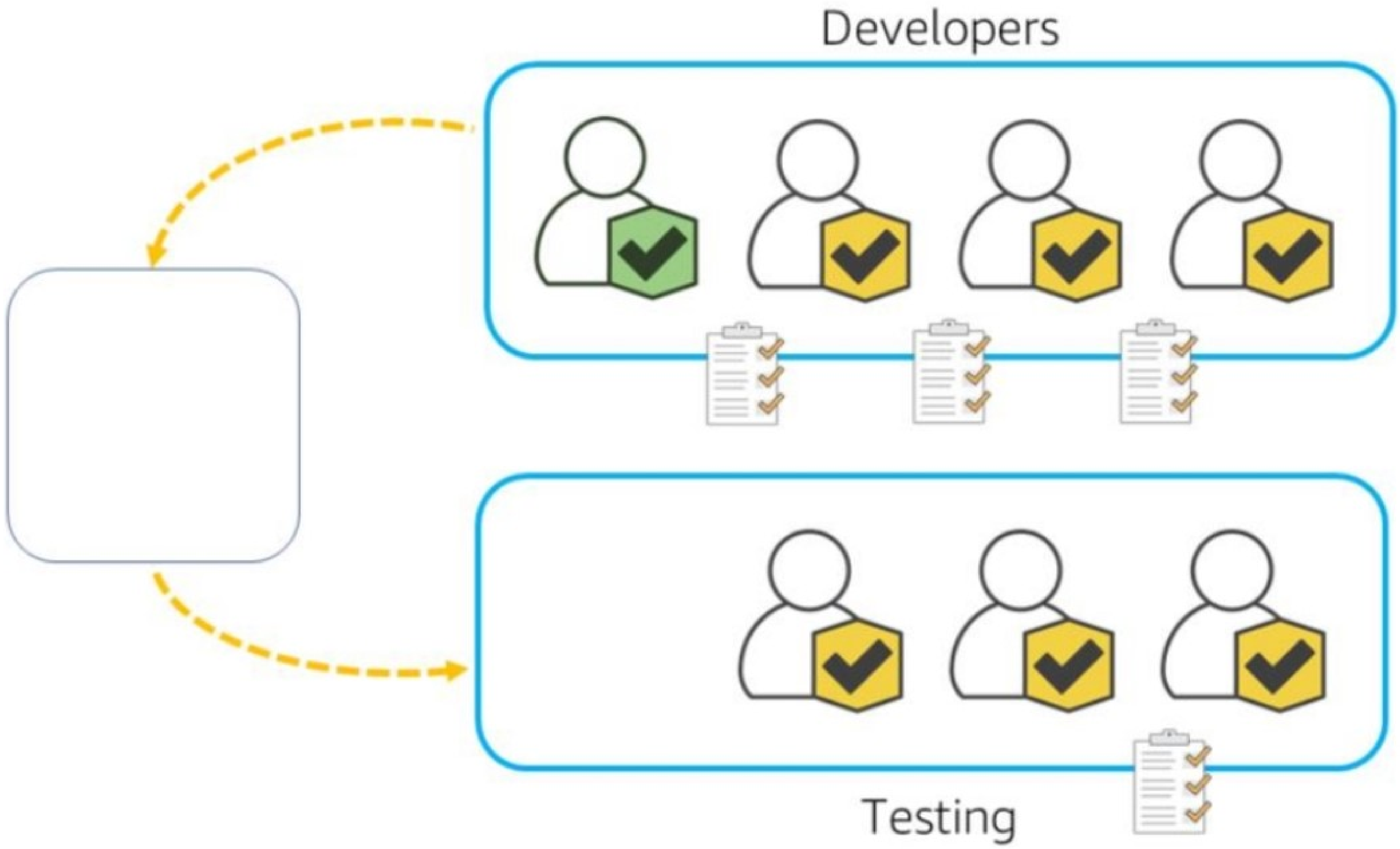
IAM User Group



IAM User Group



IAM User Group



What if the User Needs to Test for Only a Day?

What if I don't want to keep pushing and pulling the user between different groups myself?

What if I don't want to give permanent credentials to someone or something?

Federating Users

IAM Roles

A role lets you define a set of permissions to access the resources that a user or service needs.

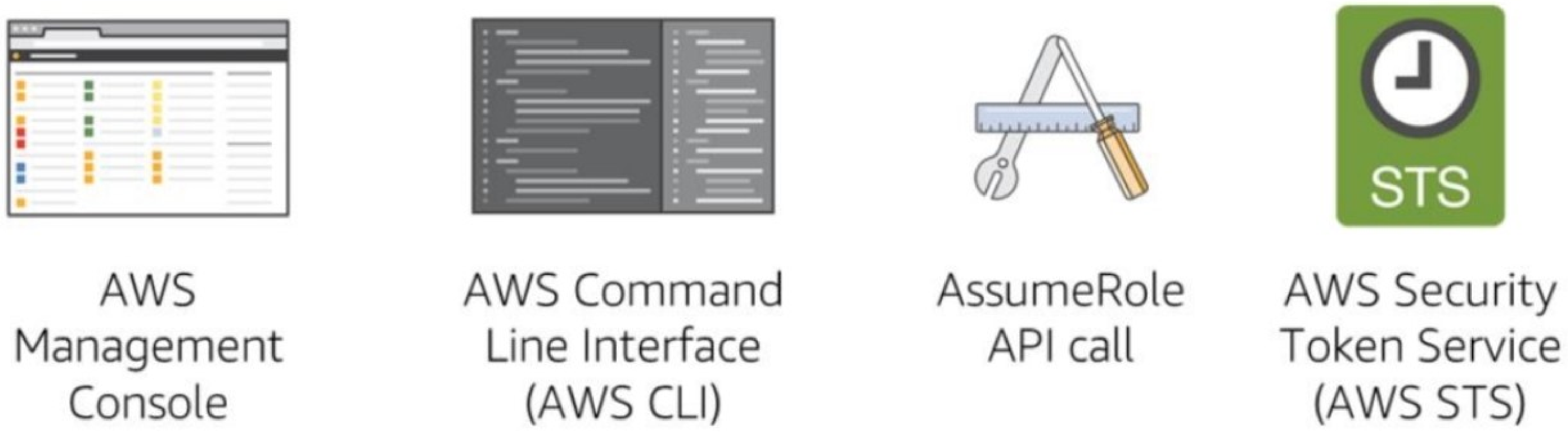
- The permissions are not attached to an IAM user or group.
- The permissions are attached to a role and the role is assumed by the user or the service.

IAM Roles

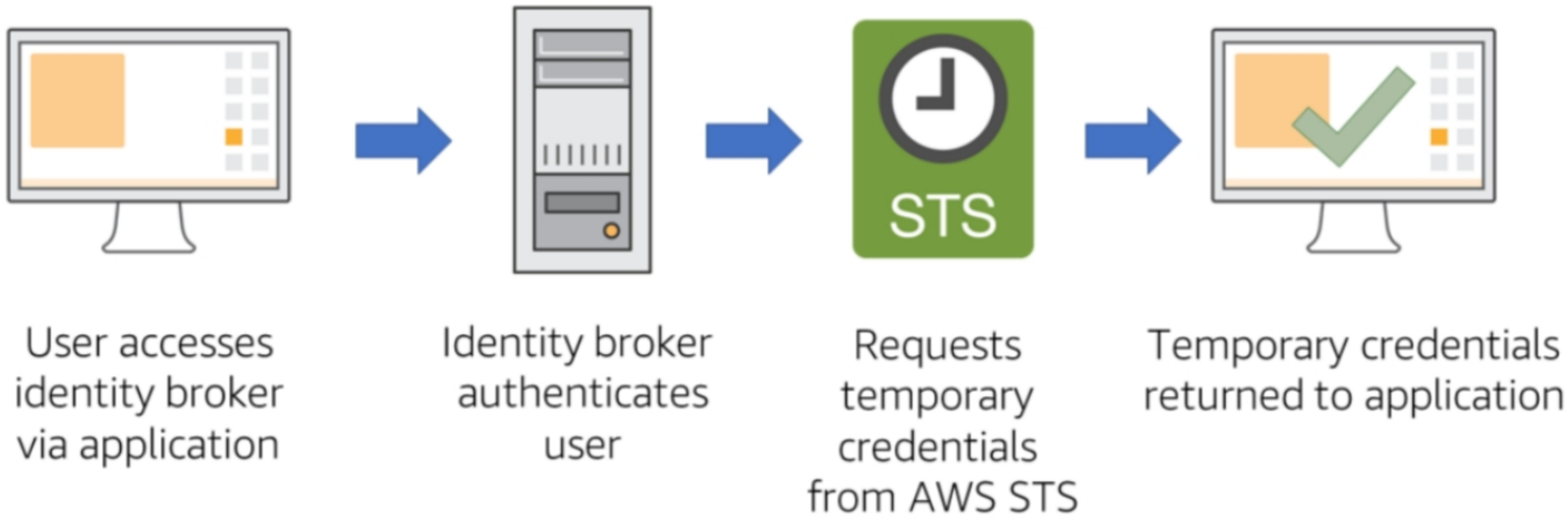
Use Cases:

- Provide AWS resources with access to AWS services
- Provide access to externally authenticated users
- Provide access to third parties
- Switch roles to access resources in:
  - Your AWS account
  - Any other AWS account (cross-account access)

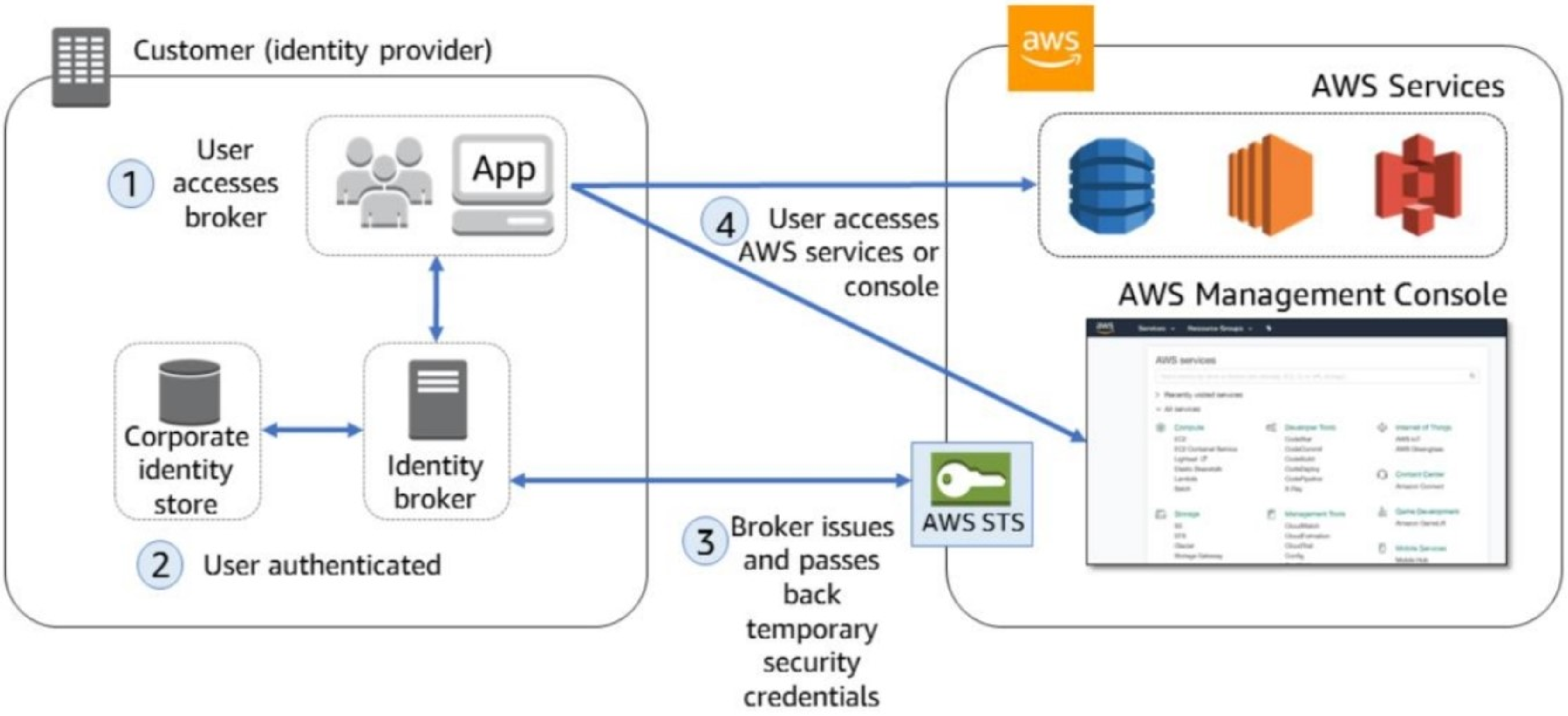
Assume Role


AWS Management Console
AWS Command Line Interface (AWS CLI)
AssumeRole API call
STS
AWS Security Token Service (AWS STS)

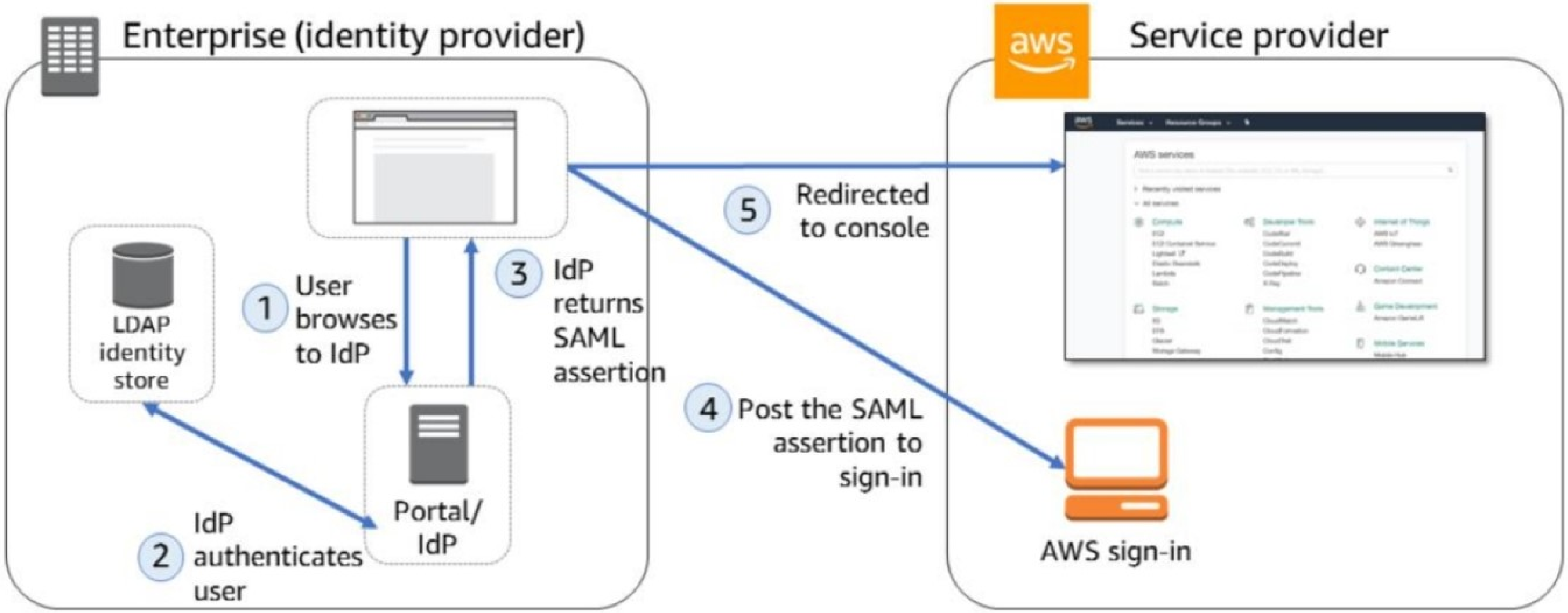
STS Identity Broker Overview


STS
User accesses identity broker via application
Identity broker authenticates user
Requests temporary credentials from AWS STS
Temporary credentials returned to application

STS Identity Broker Process



SAML

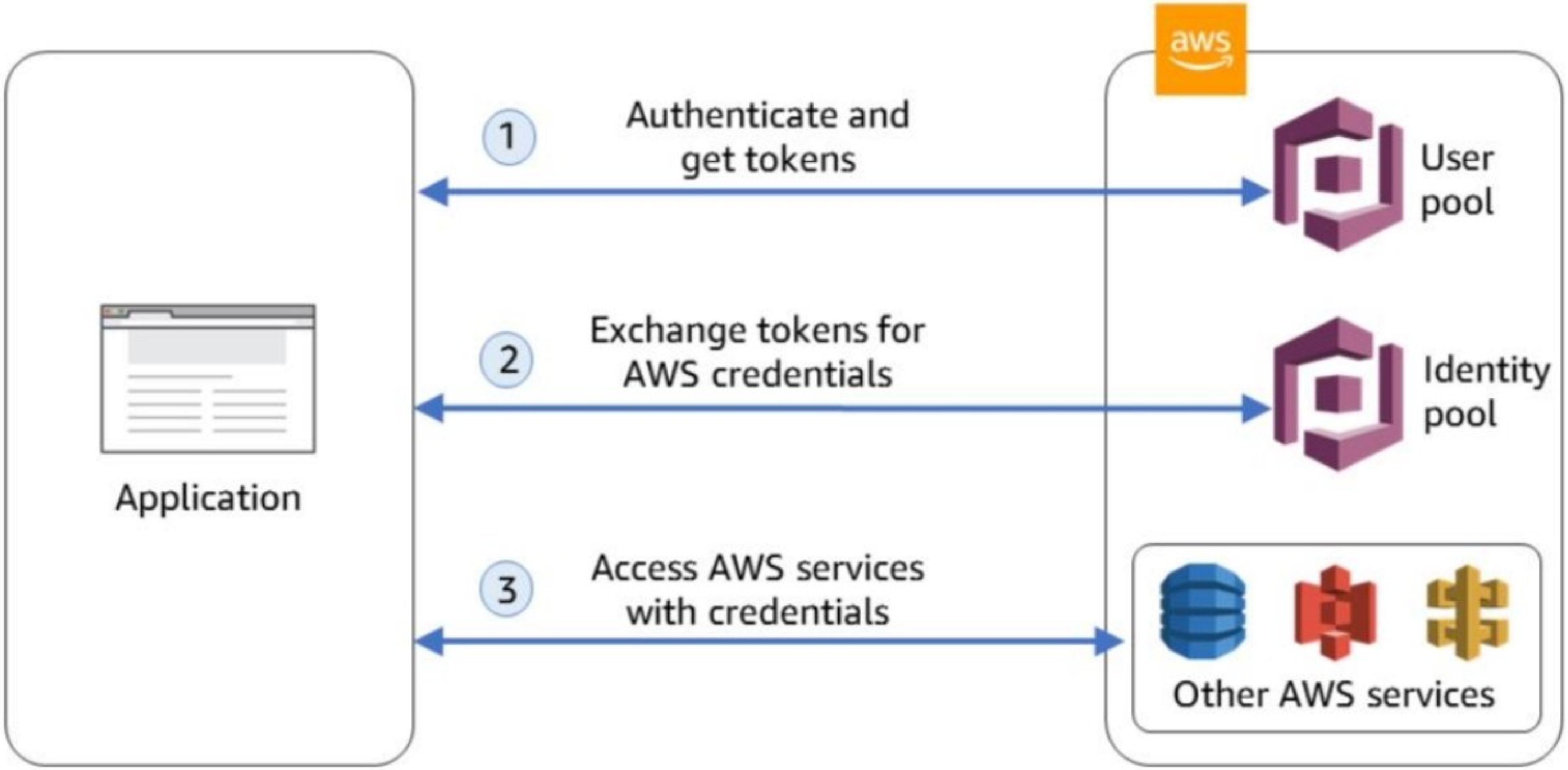

Enterprise (identity provider)
Service provider
aws
LDAP identity store
1 User browses to IdP
2 IdP authenticates user
3 IdP returns SAML assertion
4 Post the SAML assertion to sign-in
5 Redirected to console
Portal/ IdP
AWS sign-in
AWS services

Amazon Cognito

Fully managed service that provides authentication, authorization and user management for web and mobile apps.

- User pools
- Identity pools

Amazon Cognito Example


aws
1
Authenticate and get tokens
User pool
2
Exchange tokens for AWS credentials
Identity pool
Application
3
Access AWS services with credentials
Other AWS services

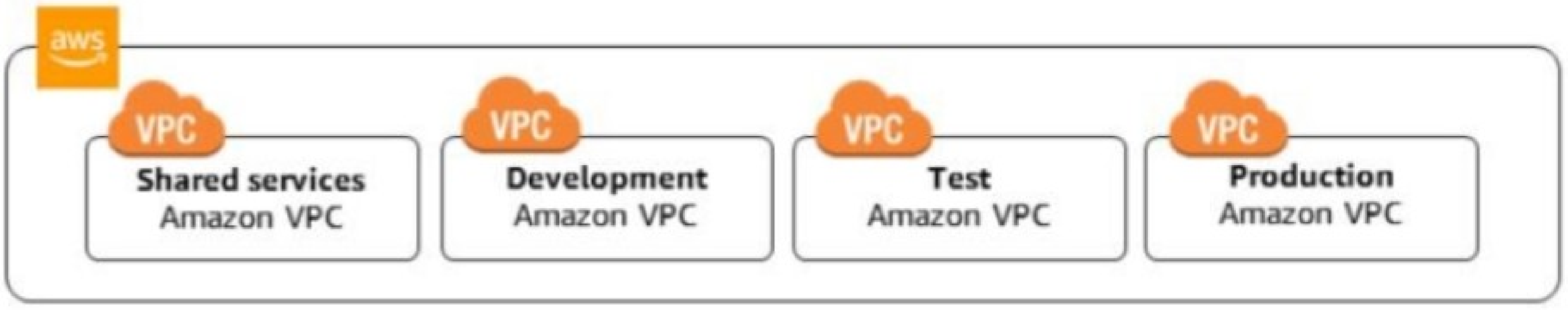
Multiple Accounts

AWS "In the World"

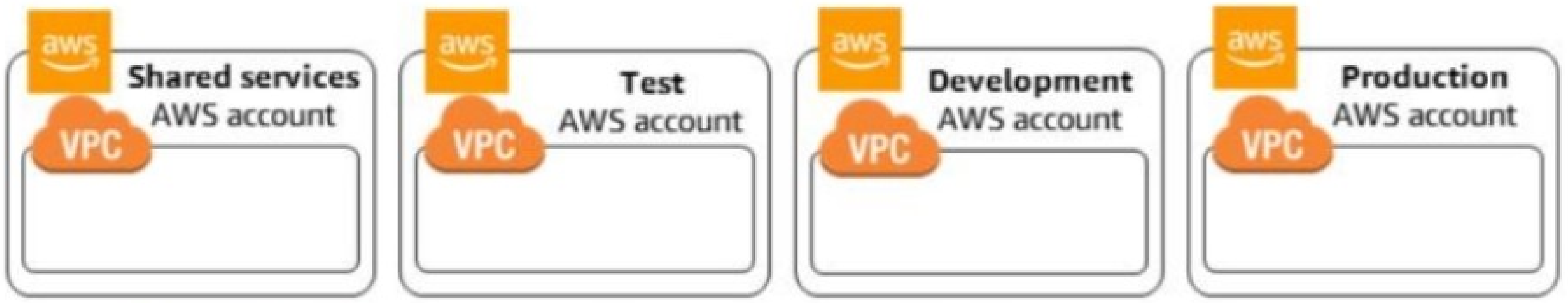
How many AWS Accounts does your organization need?

aws
Dev
aws
Test
aws
Production

AWS Recommendations



One account – multiple VPCs



Multiple accounts – One VPC per account

Multiple AWS Accounts

Can be leveraged for **isolation**:

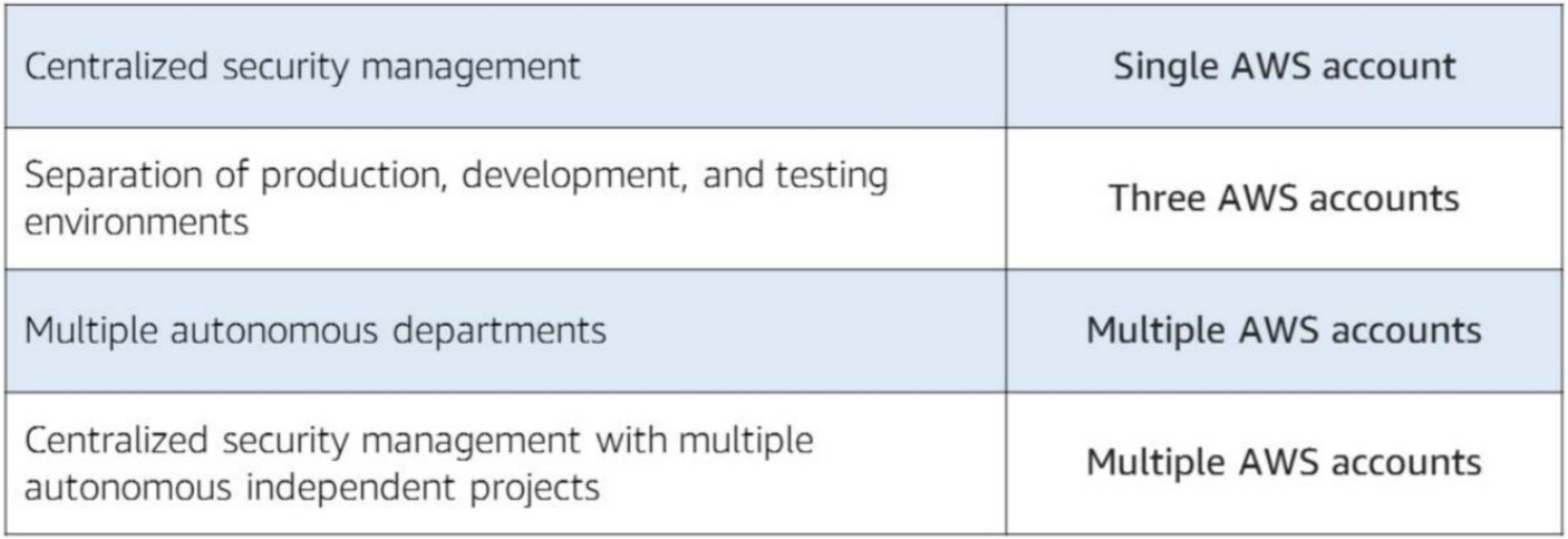
- Separate business units, dev/test/production environments

Can be leveraged for **security**:

- Separate accounts for regulated workloads, different geographical locations, governing other accounts

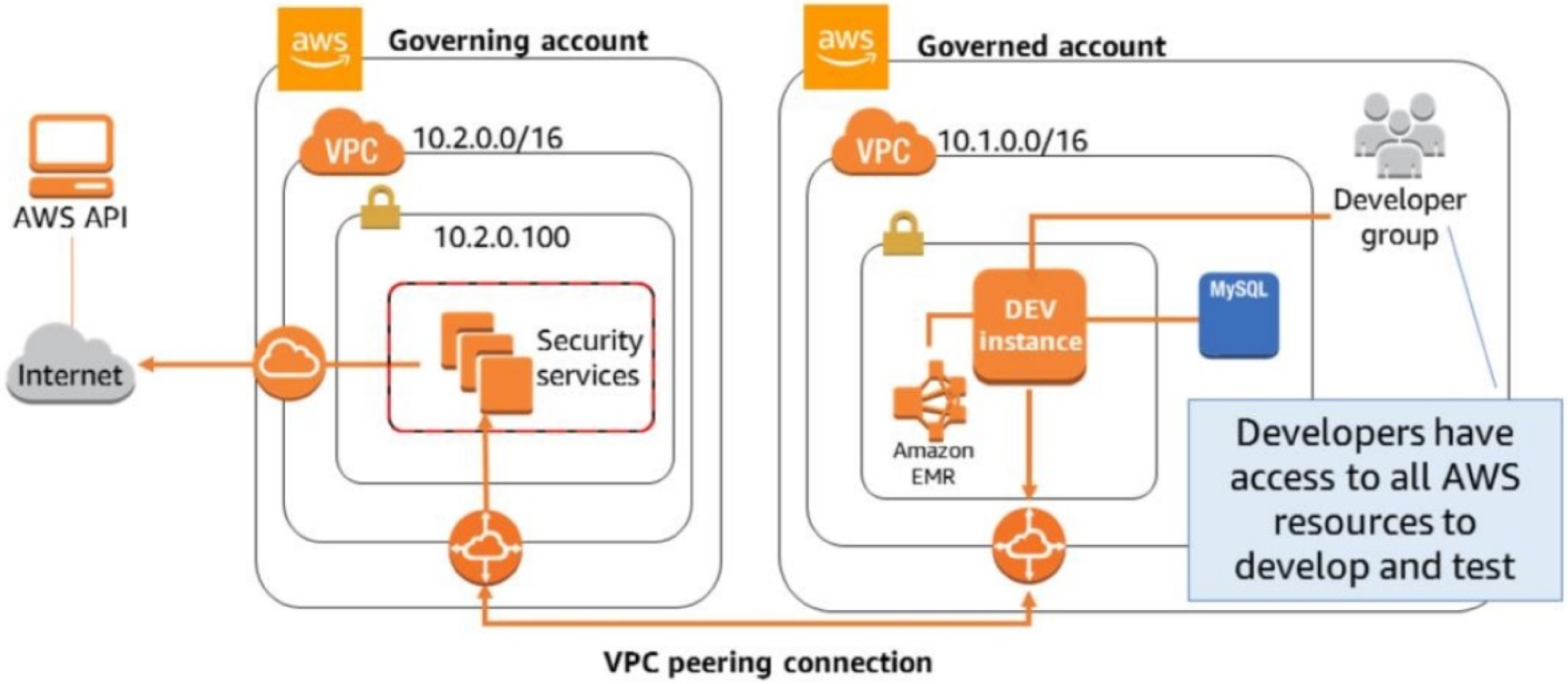
Cross-account access is not enabled by default

Strategies for Using Multiple AWS Accounts



Centralized security management	Single AWS account
Separation of production, development, and testing environments	Three AWS accounts
Multiple autonomous departments	Multiple AWS accounts
Centralized security management with multiple autonomous independent projects	Multiple AWS accounts

Using Multiple Accounts for Governance

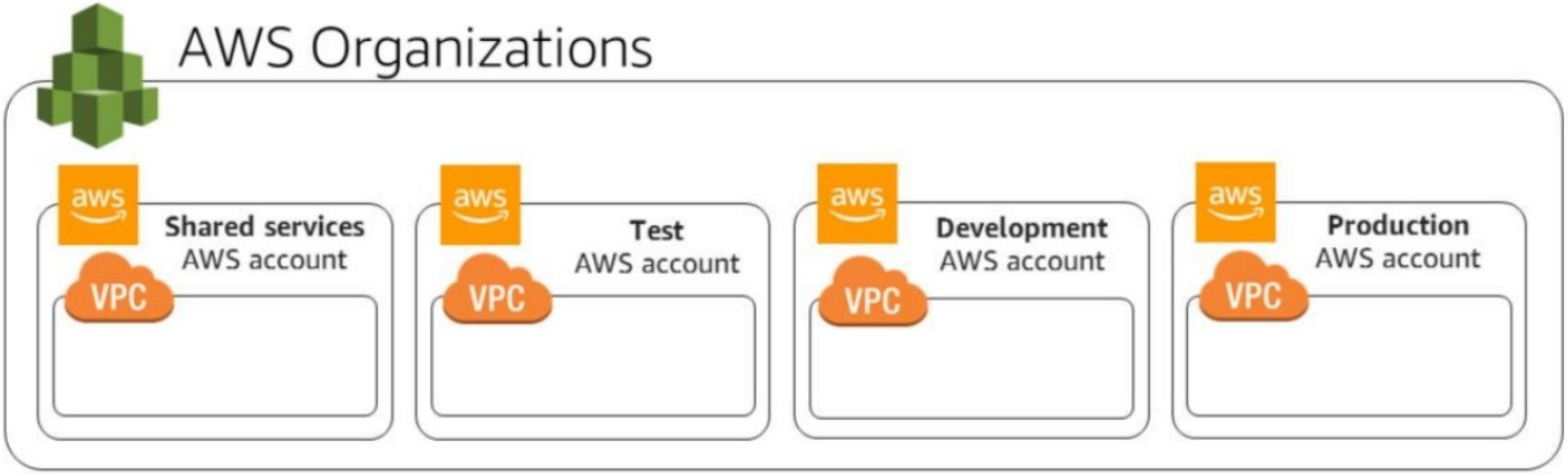


How Do I Manage All These Accounts?

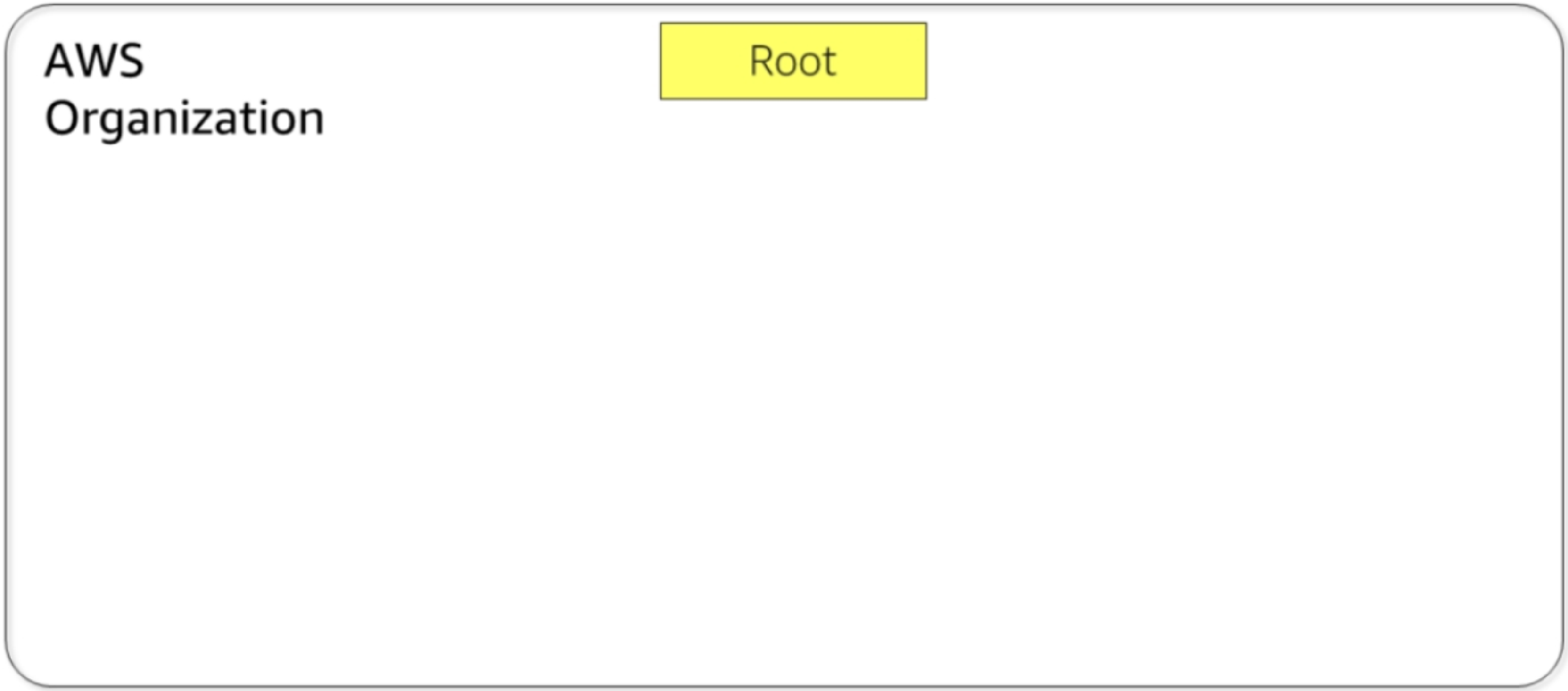
Centralized account management

- Group-based account management
- Policy-based access to AWS services
- Automated account creation and management
- Consolidated billing
- API-based

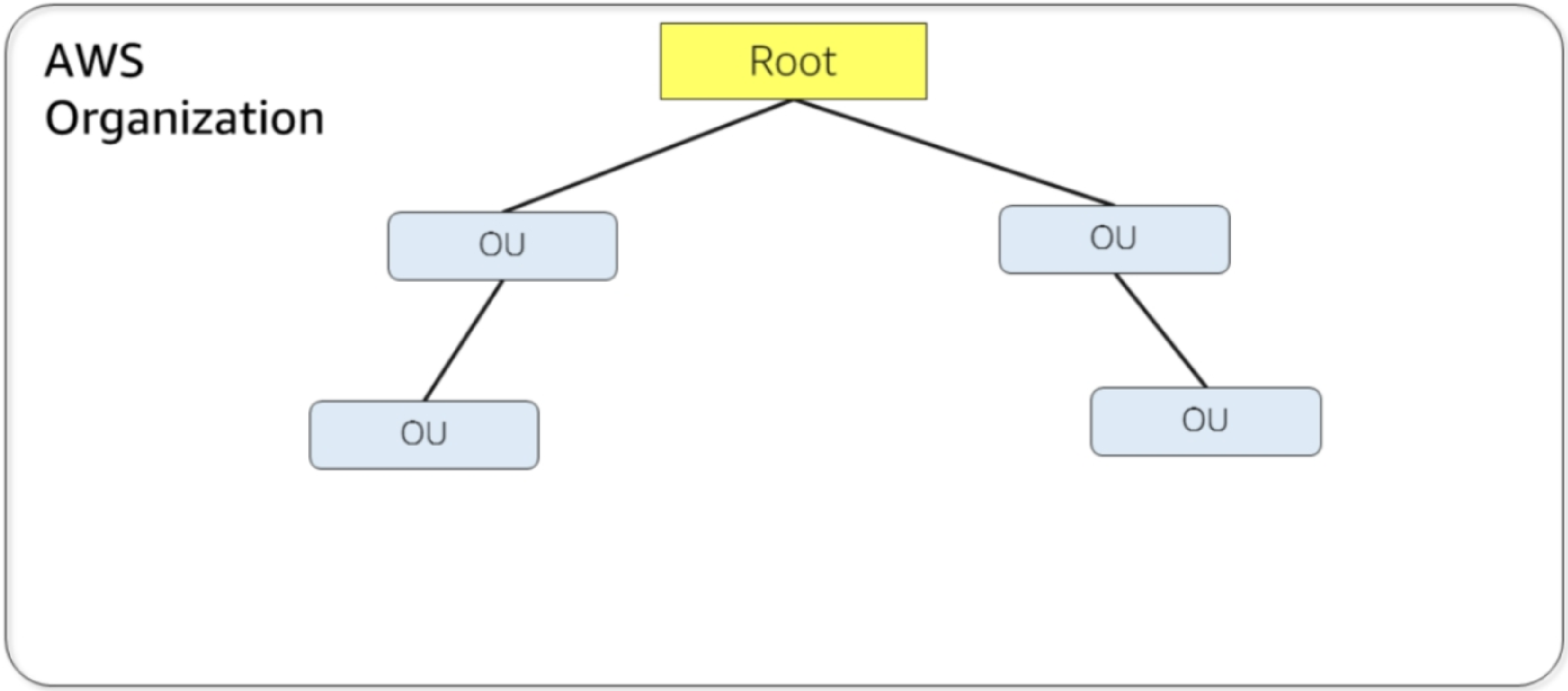
AWS Recommendations



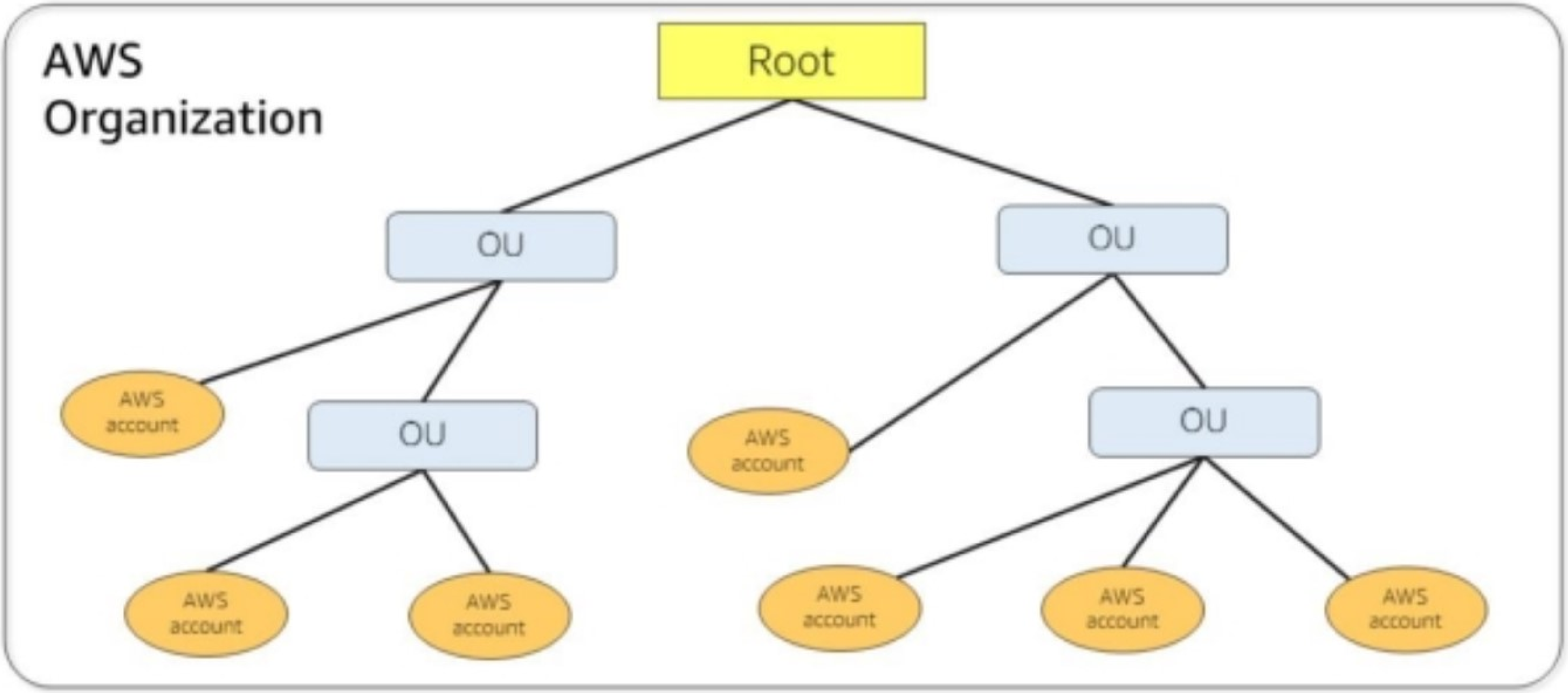
AWS Organizations: Illustrated



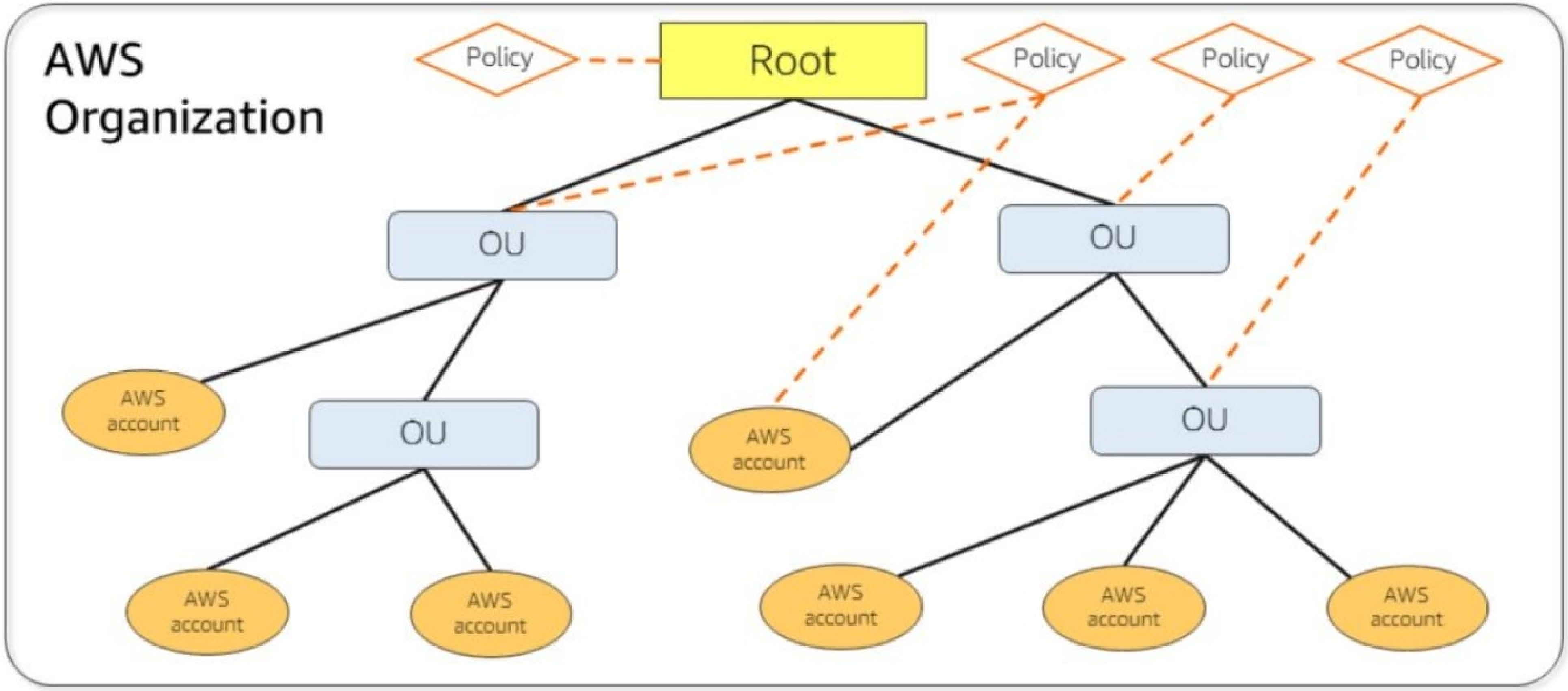
AWS Organizations: Illustrated



AWS Organizations: Illustrated



AWS Organizations: Illustrated



Review

Review

If you need to grant temporary permissions

to a resource, what would you use?

Review

If you need to grant temporary permissions

to a resource, what would you use?

IAM Role

Review

One of your users can't access an S3 bucket. What should you check to identify the cause of the problem?

Review

One of your users can't access an S3 bucket. What should you check to identify the cause of the problem?

The policies attached to the user and to the bucket

Review

1. You have created a **mobile application** that makes calls to **DynamoDB** to fetch data.
2. The application is using the **DynamoDB SDK** and the **AWS account root user access/secret access key** to connect to DynamoDB from the mobile app.
3. With respect to the best practice for **security** in this scenario, how should this be fixed?

Review

First: **Stop** using the AWS account root user in production!

Then, if possible, have the app use an **IAM role** with **web identity federation.**

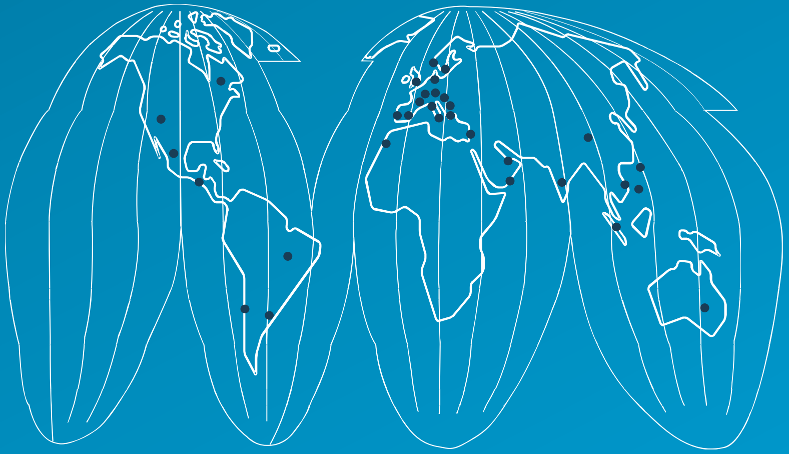
CONSULTING.TECHNOLOGY.OUTSOURCING

People matter, results count.

About Capgemini

With more than 145,000 people in 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2014 global revenues of EUR 10.5 billion.
Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

Rightshore® is a trademark belonging to Capgemini



www.capgemini.com