Security and Compliance

Identity & Visibility are foundations in AWS

Identity options in AWS

Every AWS Cloud journey is *unique.*

Going all-in on cloud solutions across the organization.

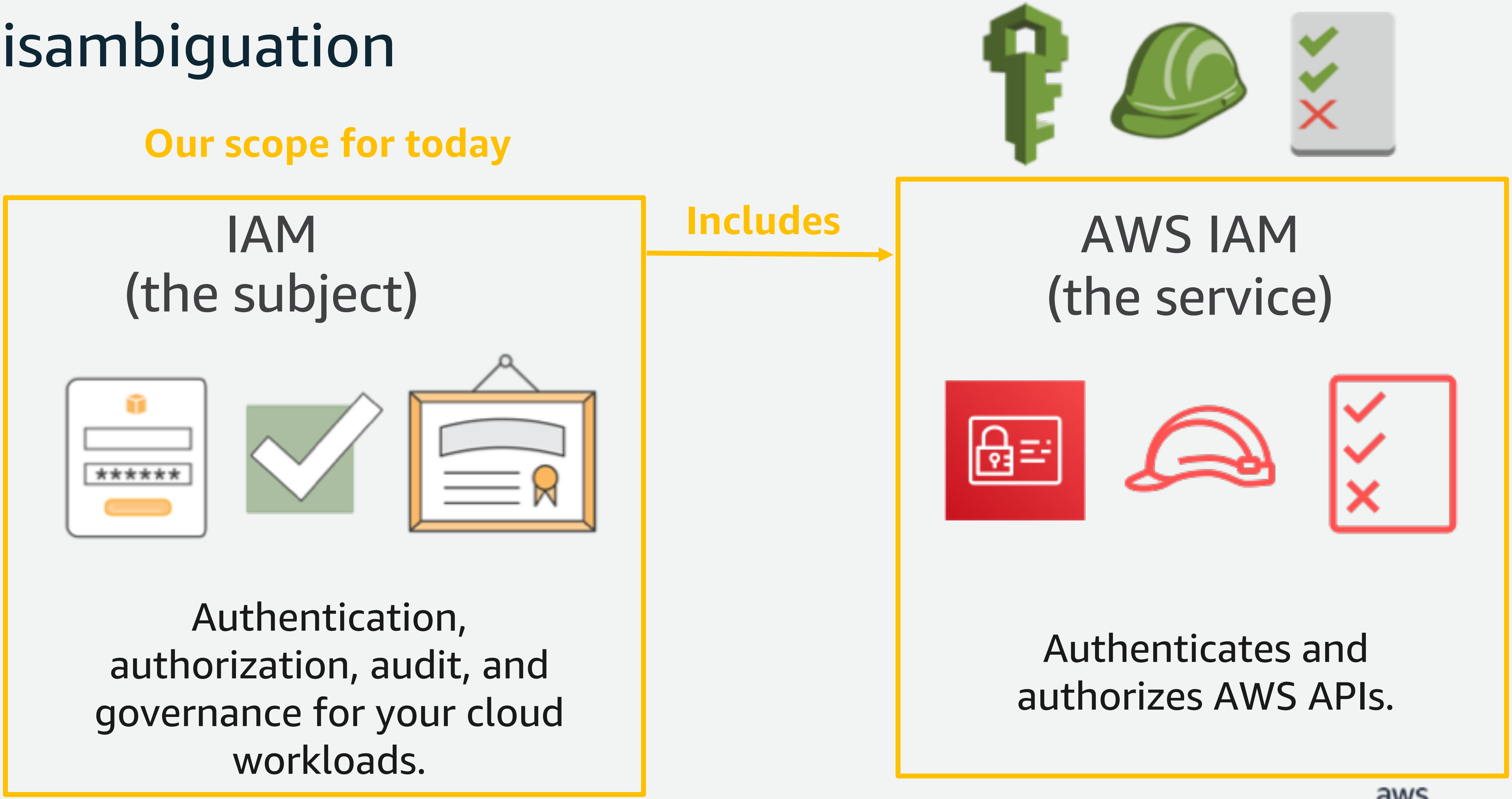
Migrating or extending existing infrastructure and applications.

Building customer facing cloud-native applications.

Using the scale of the AWS Cloud to solve new challenges.

Requiring *unique* identity and access management solutions.

Disambiguation



Our scope for today

IAM
(the subject)

Authentication, authorization, audit, and governance for your cloud workloads.

Includes

AWS IAM
(the service)

Authenticates and authorizes AWS APIs.

Identity and Access Management Means …

Authentication

Validate identities securely.

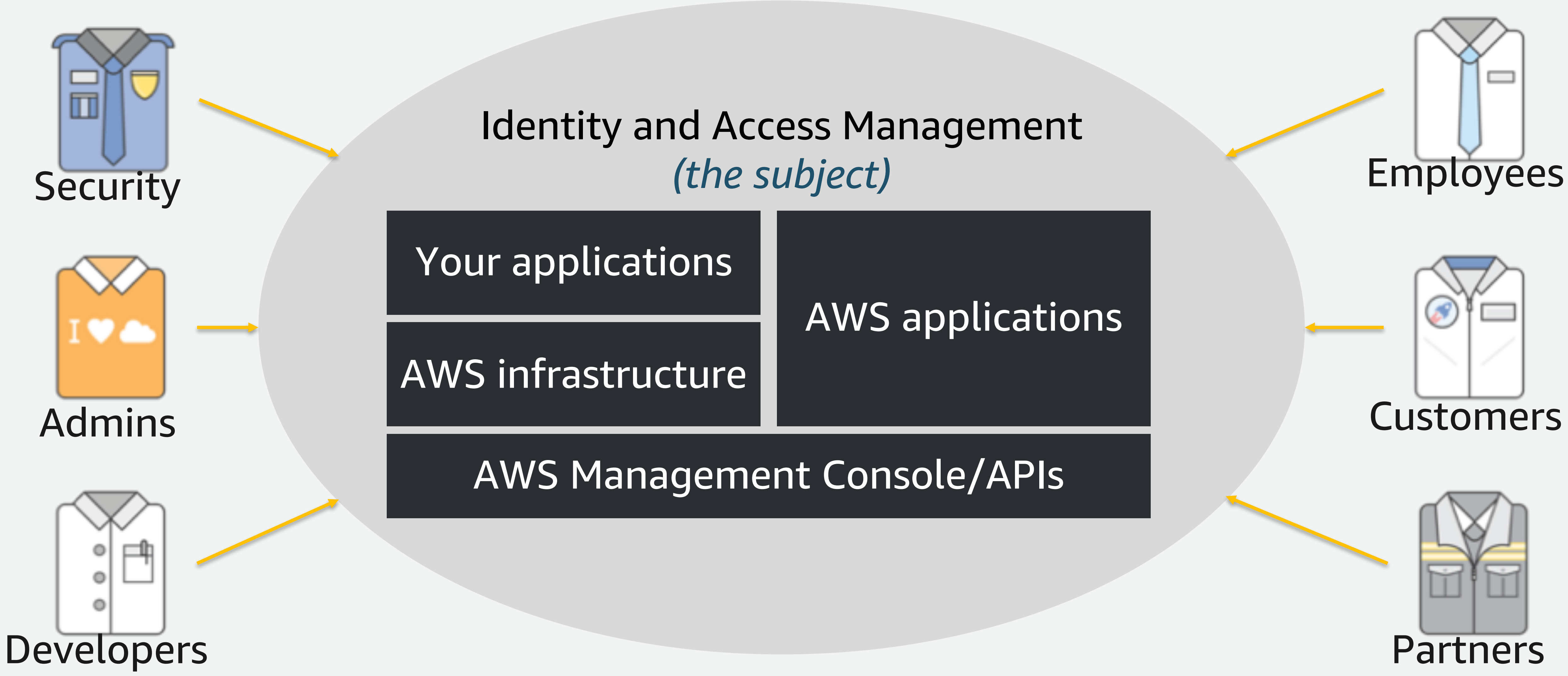
Authorization

Manage access using fine-grained policies.

Audit/Governance

Meet compliance requirements.

At All Levels



Security

Admins

Developers

Identity and Access Management

(the subject)

Your applications

AWS infrastructure

AWS applications

AWS Management Console/APIs

Employees

Customers

Partners

Foundational Services

- Identity on AWS

AWS Identity and Access Management (IAM)

Permissions

Role

- Integrating with your existing identity

AWS Directory Service

AWS SSO

SAML token

- Providing Visibility

AWS CloudTrail

Amazon CloudWatch

AWS Config

Additional Services

- Multiple Accounts

AWS Organizations

Account

Organizational unit

- Identity for applications

Amazon Cognito

- Credential Management

AWS Secrets Manager

Authentication and Authorization on AWS

Security, Governance, and Oversight

=

Authentication

+

Authorization

+

Audit/Log

AWS Identity and Access Management (IAM)

Permissions

AWS CloudTrail

AAA

Authenticate	Authorize	Audit
IAM Username/Password	IAM Policies	AWS CloudTrail
Access Key (+ MFA)	Resource Policies	AWS Config
Federation	IAM Roles	Customer IdP

IAM Principals & Policies

AWS Principals

Account Owner ID (Root Account)

- Access to all subscribed services
- Access to billing
- Access to console and APIs
- Access to Customer Support

IAM Users, Roles, Federated Users

- Access to specific services
- Access to console and/or APIs
- Access to Customer Support (Business and Enterprise)

Temporary Security Credentials for Applications

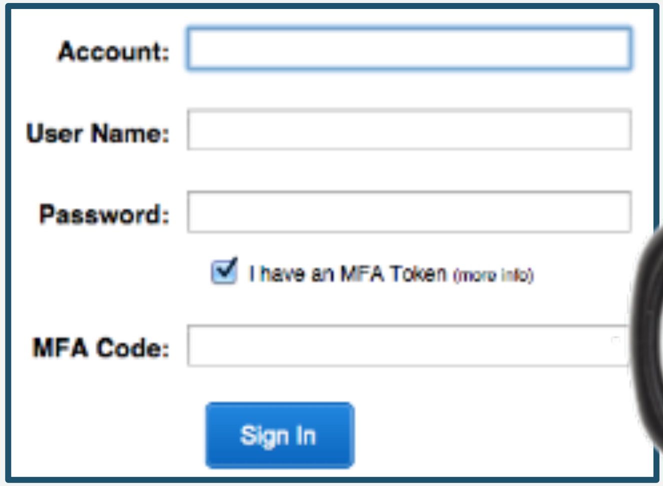
- Access to specific services
- Access to console and/or APIs

AWS Identity Authentication

How do we know you are who you say you are?

AWS Management Console

Login with **Username/Password** with optional **MFA** (recommended)



Account:	
User Name:	
Password:	
	☑ I have an MFA Token (more info)
MFA Code:	

API and CLI access

Access API using **Access Key + Secret Key**, with optional MFA

ACCESS KEY ID

Ex: `AKIAIOSFODNN7EXAMPLE`

SECRET KEY

Ex: `UtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY`

Multi-Factor Authentication device

AWS Authorization

What are you allowed to do?

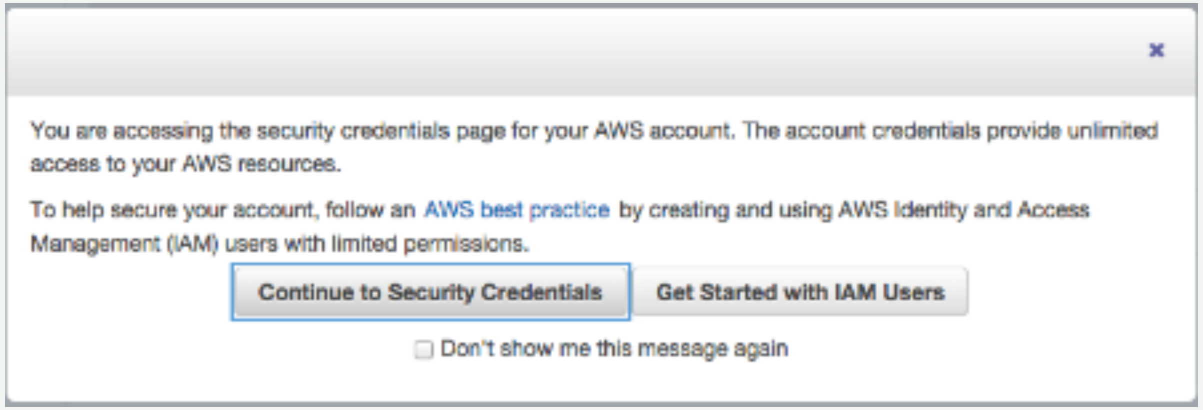
Account Owner (Root)

Privileged for all actions

Immediately turn on MFA for Root!

IAM and Resource Policies

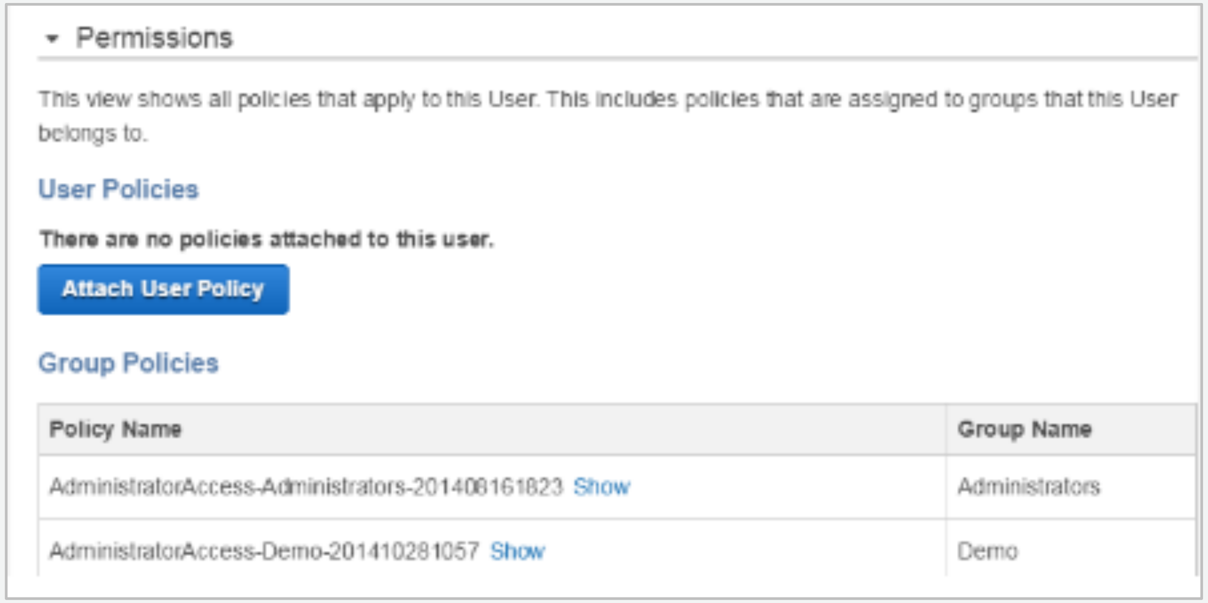
Privileges defined at User and Resource Level



You are accessing the security credentials page for your AWS account. The account credentials provide unlimited access to your AWS resources.

To help secure your account, follow an AWS best practice by creating and using AWS Identity and Access Management (IAM) users with limited permissions.

×
Continue to Security Credentials
Get Started with IAM Users
□ Don't show me this message again



▾ Permissions

This view shows all policies that apply to this User. This includes policies that are assigned to groups that this User belongs to.

User Policies

There are no policies attached to this user.

Group Policies

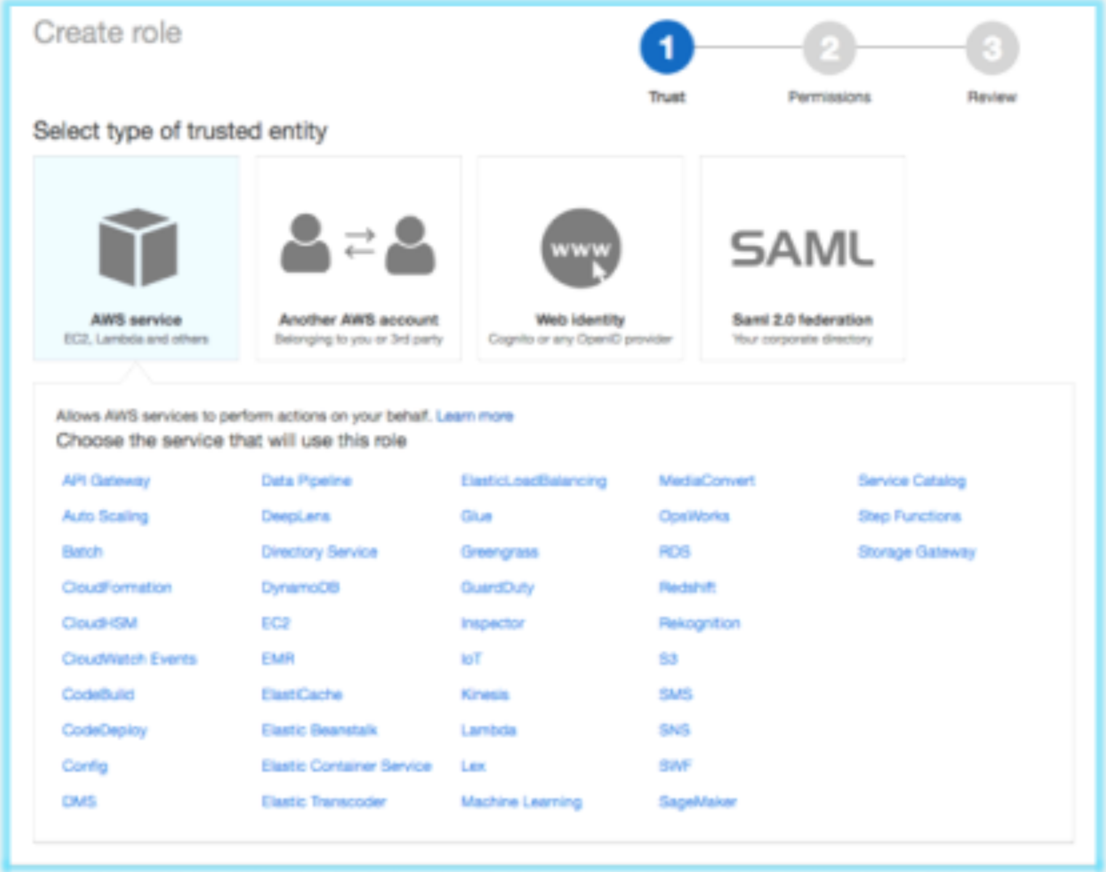
Policy Name	Group Name
AdministratorAccess-Administrators-201408161823 Show	Administrators
AdministratorAccess-Demo-201410281057 Show	Demo

Application Access to Data & Resources

Avoid hardcoding credentials in source code. You can use **IAM roles** instead.

- AWS distributes and rotates short-term credentials on your behalf automatically.
- IAM roles work with Amazon EC2, Amazon EC2 containers, and AWS Lambda functions.

You can define fine-grained permissions to AWS resources using IAM policies.



Select type of trusted entity

Allows AWS services to perform actions on your behalf. Learn more

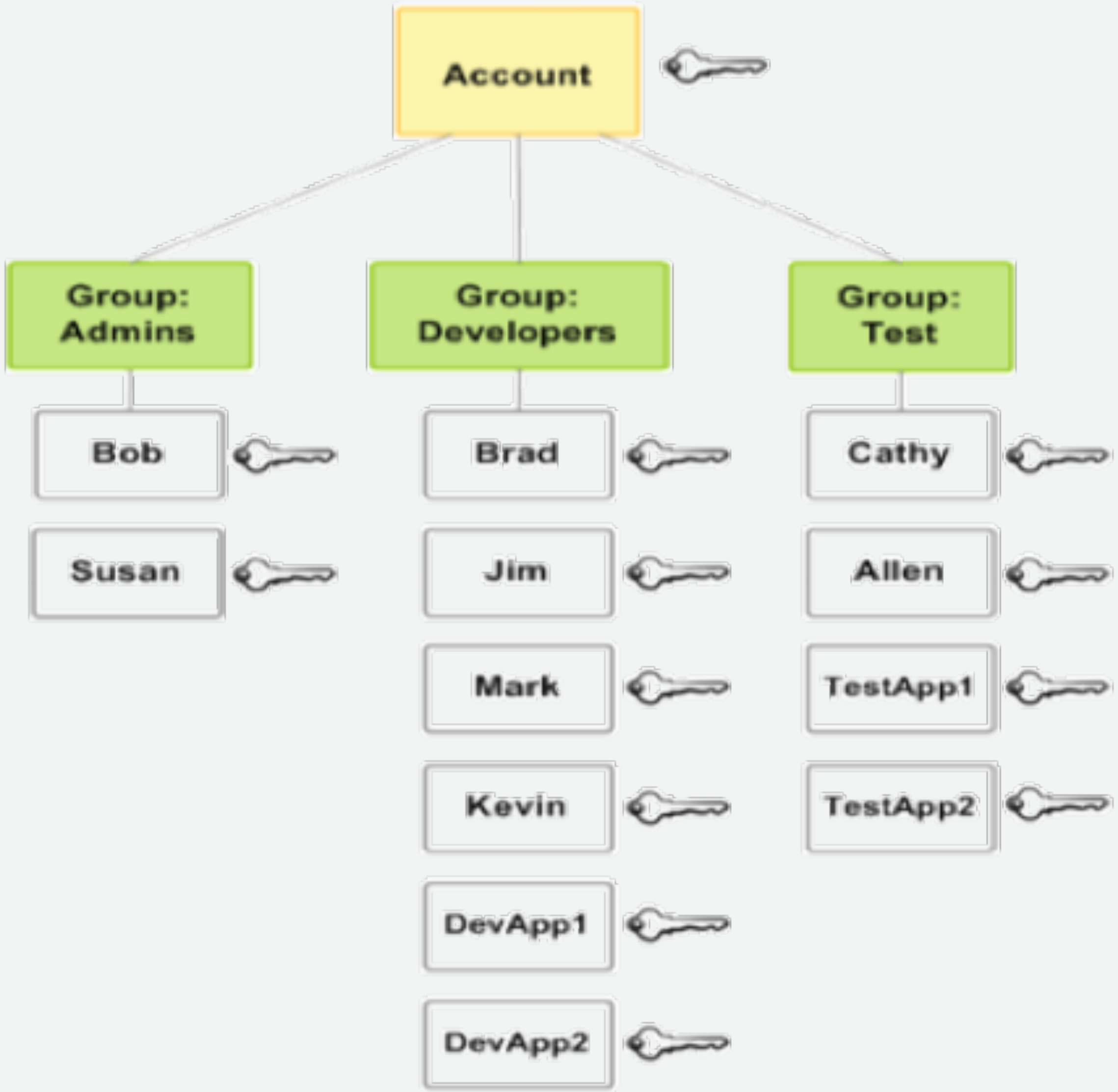
Choose the service that will use this role

API Gateway
Auto Scaling
Batch
CloudFormation
CloudHSM
CloudWatch Events
CodeBuild
CodeDeploy
Config
DMS
Data Pipeline
DeepLens
Directory Service
DynamoDB
EC2
EMR
ElastiCache
Elastic Beanstalk
Elastic Container Service
Elastic Transcoder
ElasticLoadBalancing
Glue
Greengrass
GuardDuty
Inspector
IoT
Kinesis
Lambda
Lex
Machine Learning
MediaConvert
OpsWorks
RDS
Redshift
Rekognition
S3
SMS
SNS
SWF
SageMaker
Service Catalog
Step Functions
Storage Gateway

AWS Identity and Access Management (IAM)

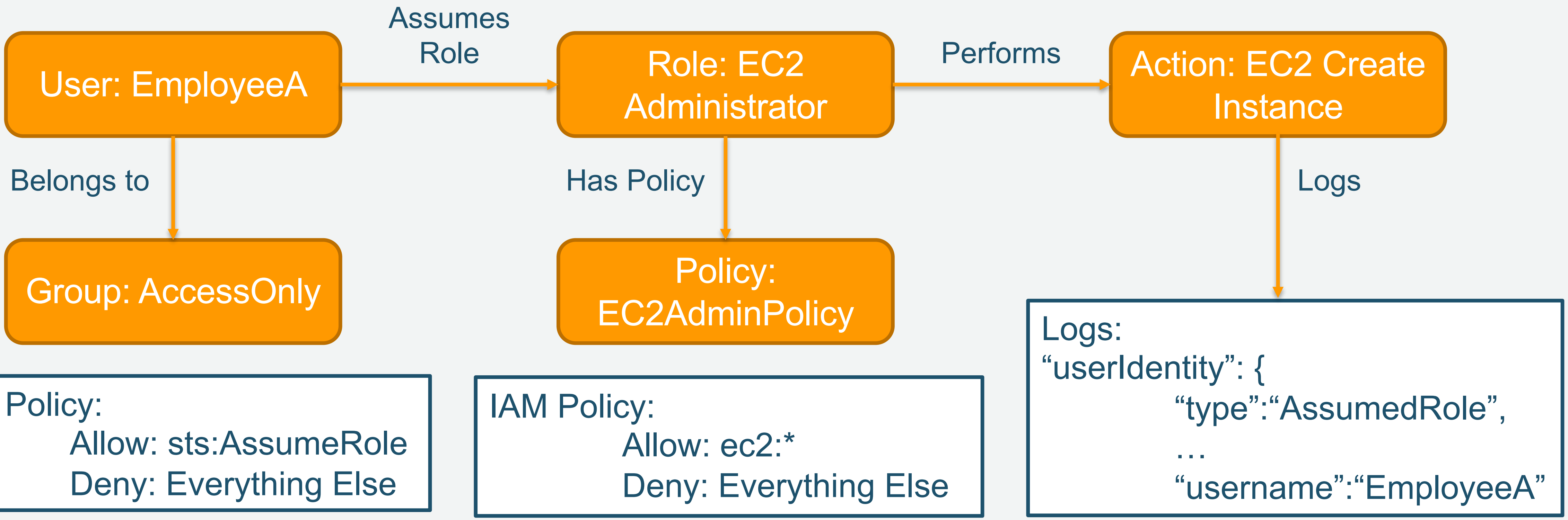
Securely control access to AWS services and resources for your users

- Username/User
- Manage groups of users
- Centralized Access Control



AWS Identity and Access Management (IAM)

Role-based Authorization



User: EmployeeA

Assumes Role

Role: EC2 Administrator

Performs

Action: EC2 Create Instance

Belongs to

Group: AccessOnly

Has Policy

Policy: EC2AdminPolicy

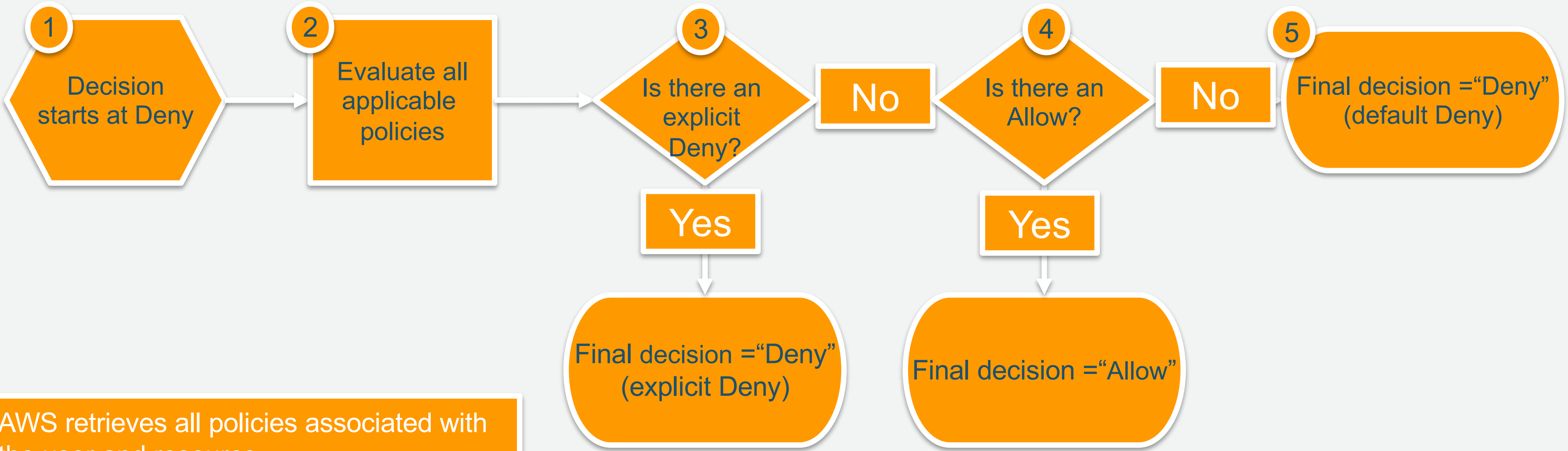
Logs

```
IAM Policy:
        Allow: sts:AssumeRole
        Deny: Everything Else
```

```
IAM Policy:
        Allow: ec2:*
        Deny: Everything Else
```

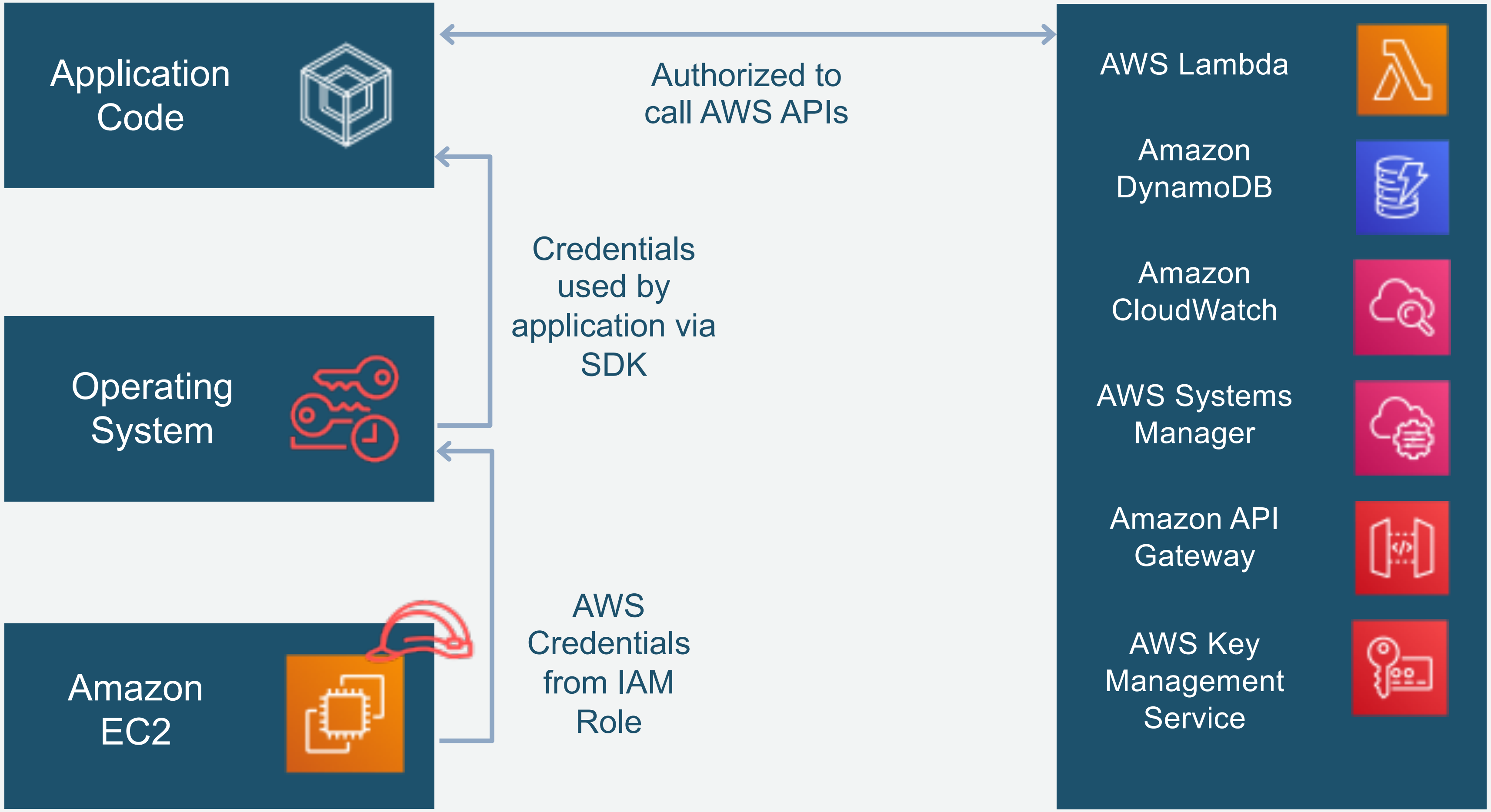
```
Logs:
“userIdentity”: {
        “type”:“AssumedRole”,
        …
        “username”:“EmployeeA”
```

Basic Policy Enforcement


1
Decision starts at Deny
2
Evaluate all applicable policies
3
Is there an explicit Deny?
No
4
Is there an Allow?
No
5
Final decision =“Deny” (default Deny)
Yes
Final decision =“Deny” (explicit Deny)
Yes
Final decision =“Allow”

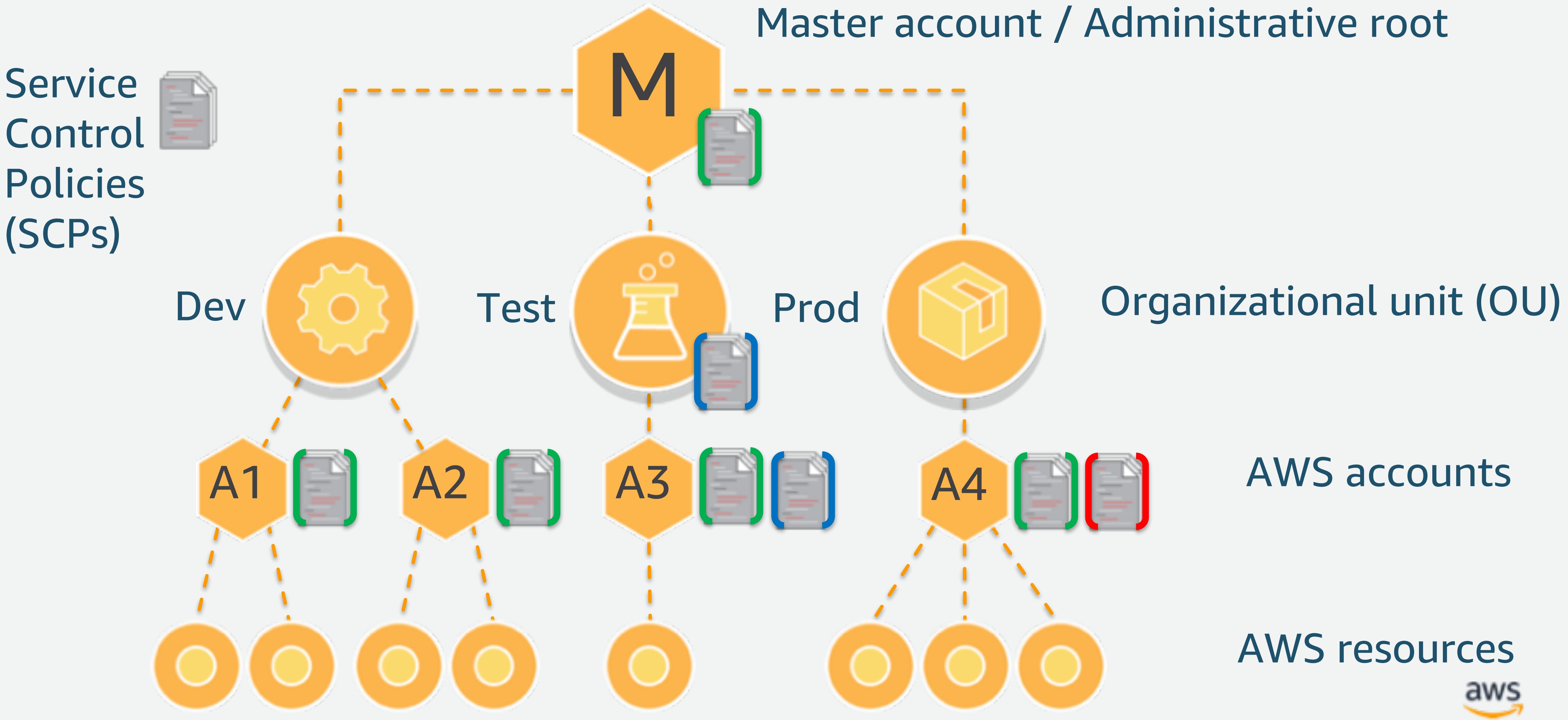
- AWS retrieves all policies associated with the user and resource.
- Only policies that match the action, resource and conditions are evaluated.

Applications built on AWS calling AWS resources

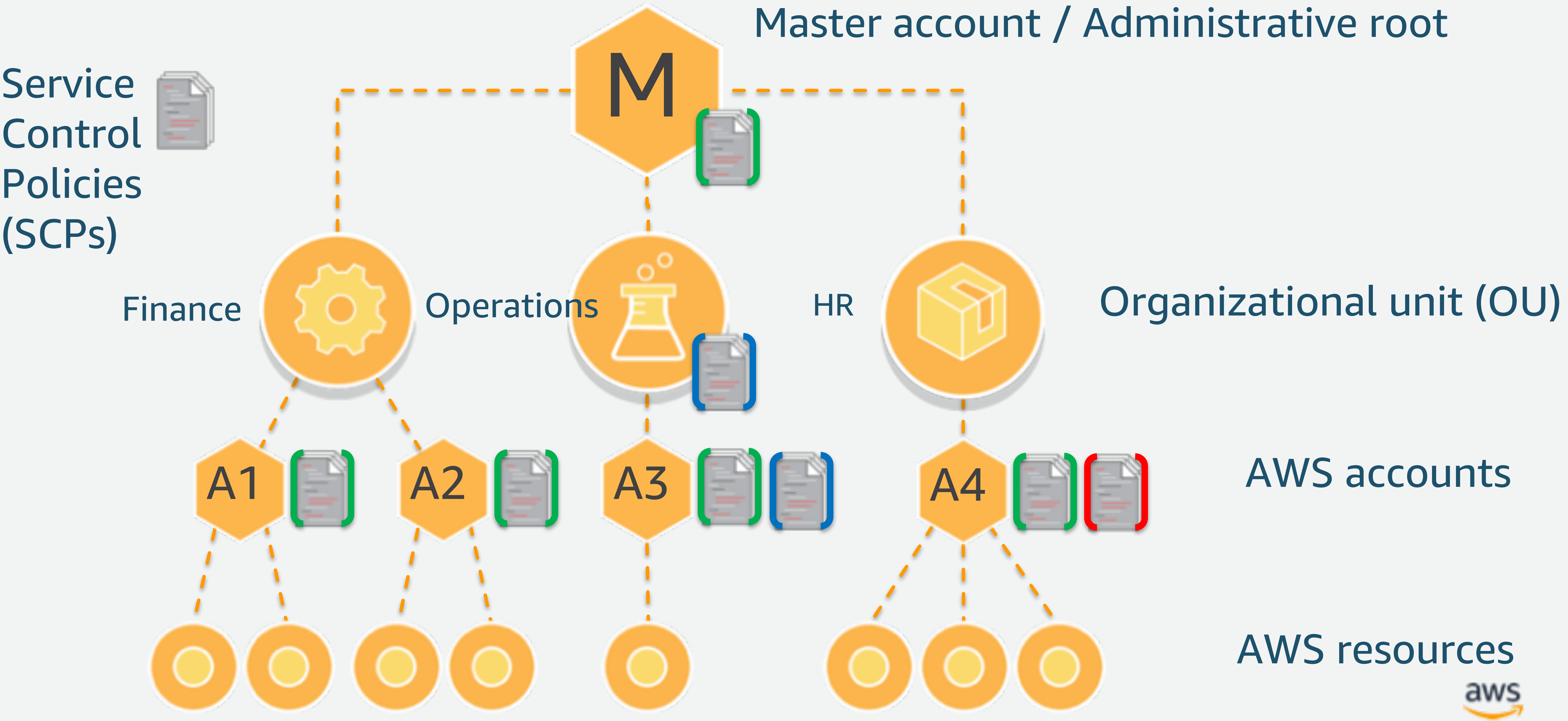

Application Code
Operating System
Amazon EC2
Authorized to call AWS APIs
Credentials used by application via SDK
AWS Credentials from IAM Role
AWS Lambda
Amazon DynamoDB
Amazon CloudWatch
AWS Systems Manager
Amazon API Gateway
AWS Key Management Service

AWS Organizations

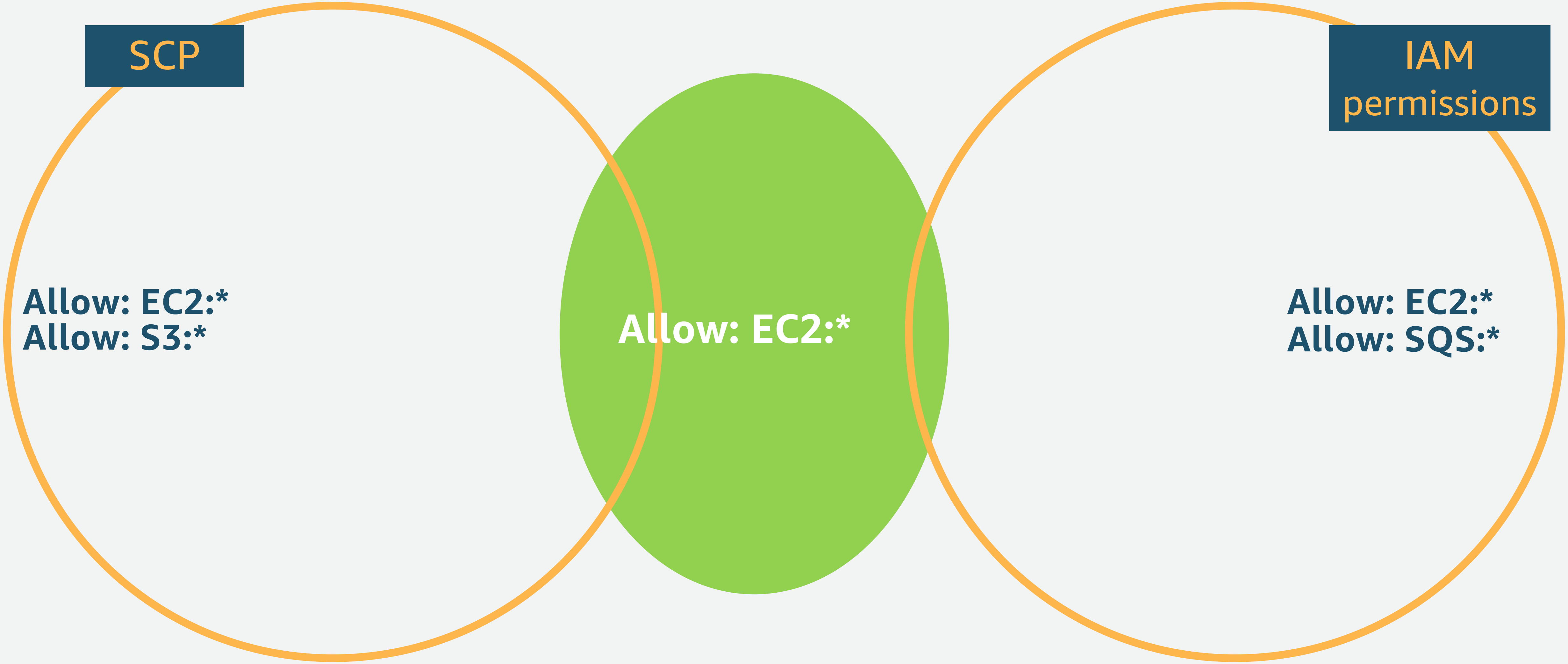
Example 1: OUs by environment



Example: OUs by Business Unit



AWS Organizations: Together with IAM



SCP

Allow: EC2:*
Allow: S3:*

Allow: EC2:*

IAM permissions

Allow: EC2:*
Allow: SQS:*

Wrap up

What did we do?

- Identity options – IAM to Organizations
- Visibility is easier – CloudTrail & Guard Duty
- IAM Principals & Policies – Humans & systems
- Service control policies – hierarchical control
- Auditing is easier – how can you use this?