



AWS
re:Invent

SEC402-R1

AWS Identity: Permission boundaries & delegation

Cameron Worrell

Solutions Architect
Amazon Web Services

Ilya Epshteyn

Principal Solutions Architect
Amazon Web Services

Workshop team



Faraz
Angabini



Rob
Barnes



Cassia
Martin



Yuval
Schaal



Alex
Tomic



Tatyana
Yatskevich



Cameron
Worrell



Ilya
Epshteyn

Question

Who would be comfortable giving developers permission to create IAM roles (e.g. for Lambda functions) in production accounts?

Problem: safe delegation of permission management

- Should use caution when granting permission to create users and roles
- But there are many situations where user and role creation is required
- So, we need a way to safely delegate permission management

Solution: permissions boundaries

- Safely delegate permission management
- Free up developers (get out of their way) and do so securely
- Also allow multiple teams in the same account to do permission management

Agenda

Basics

Demo

Mechanism

Resource Restrictions

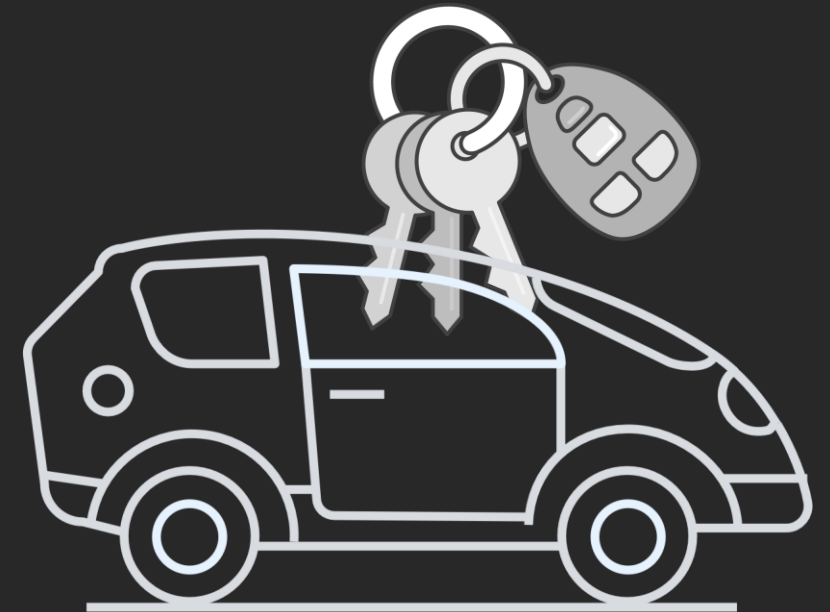
Hands on

Basics

Bob gives the car keys to his teenager

- Car keys give a lot of power: drive fast, drive anywhere, etc.
- You can set rules: don't speed, don't go beyond 20 mile range, etc.
- ...but, you can only verify that they followed your rules (check odometer, see if they got a speeding ticket or got into an accident.)

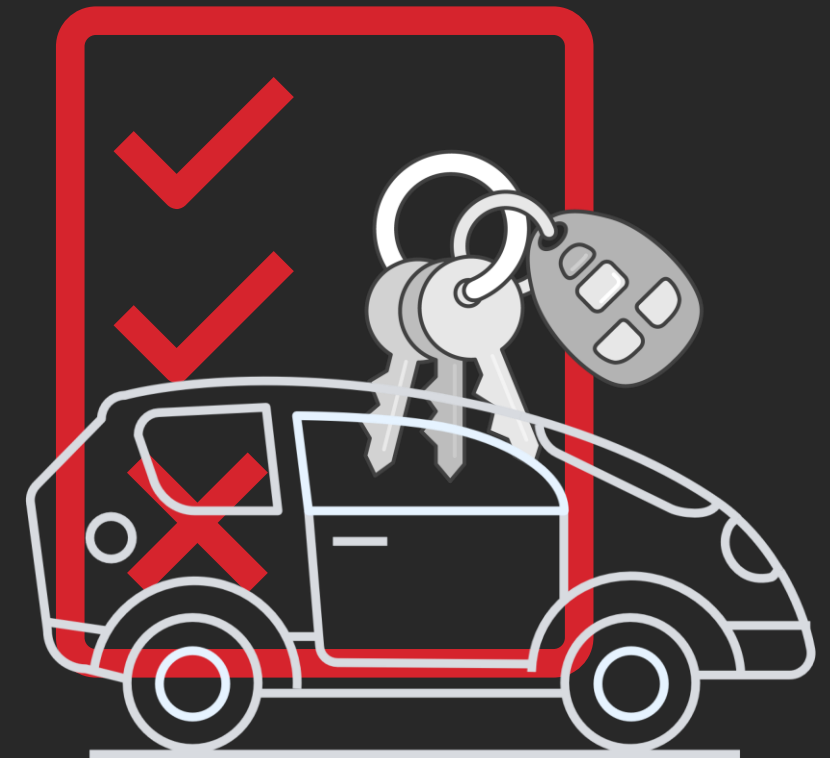
Once you have the car keys you can drive however you want.



Bob gives the car keys to his teenager

- Some cars have programmable keys so you can restrict certain parameters.
- Ability (permission) of the car key is the intersection between the desire of the driver and the settings you program. **Bound keys.**

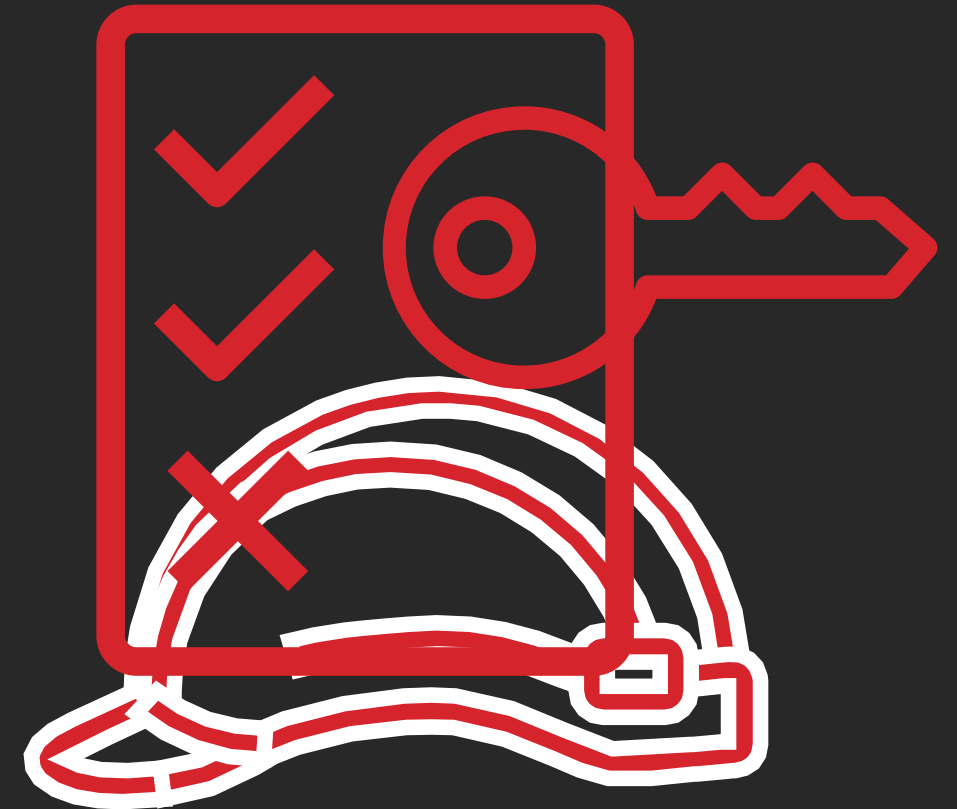
Key programming sets maximum ability of the key.



Bob gives permissions management to developers

- Permission to create users or roles provides a lot of power.
- Developer attaches policies (what they want a role to be able to do) but you also require a permissions boundary (like the programming on the car key).
- Effective permission of the role is the intersection of the two. **Bound roles.**

Permissions boundary sets maximum permissions of the role they create.

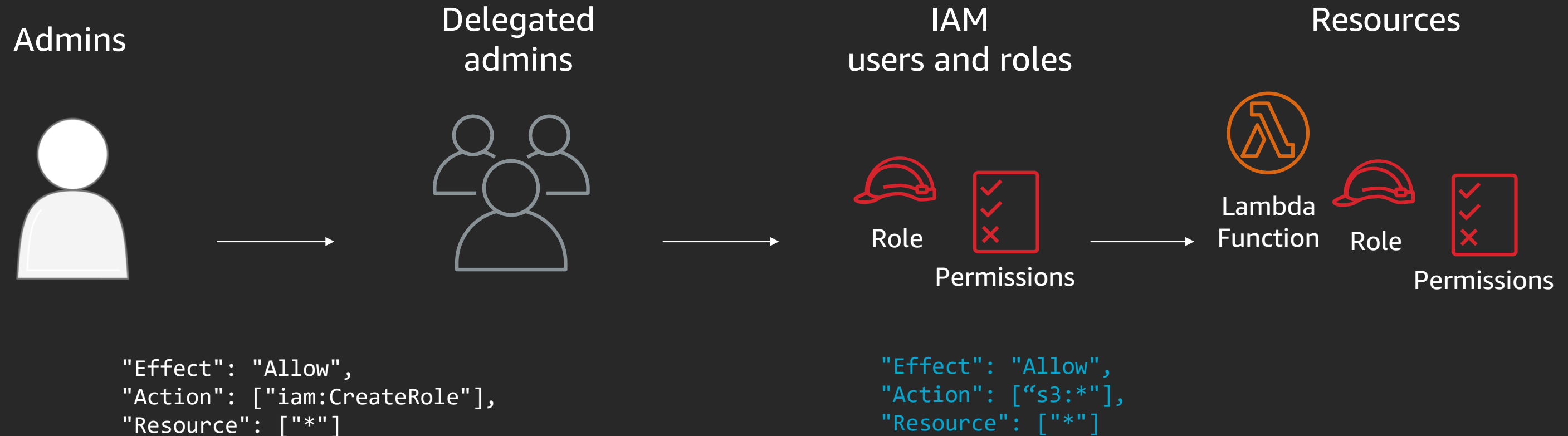


What are permissions boundaries?

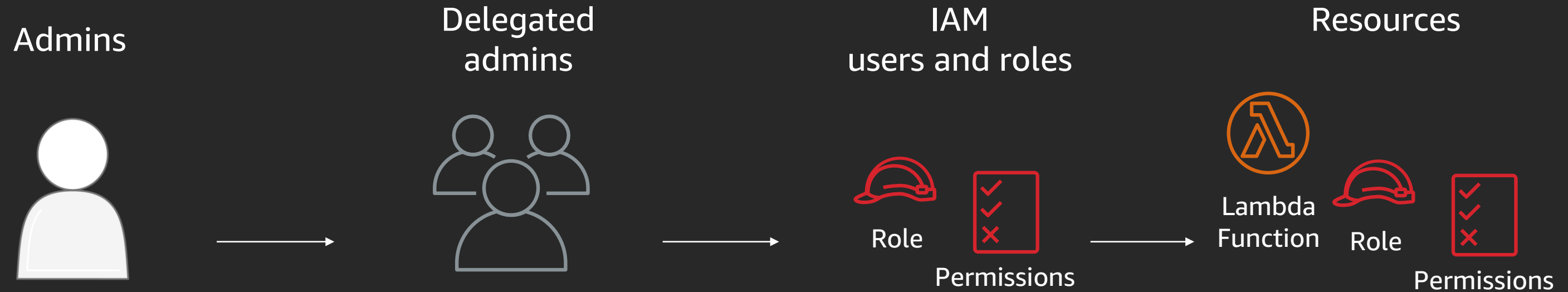
Allow you to **delegate** permission to **create users and roles** while preventing **privilege escalation** or **unnecessarily broad permissions**.

Permission boundaries control the **maximum permissions** of a user or role created by a delegated admin.

Before permission boundaries



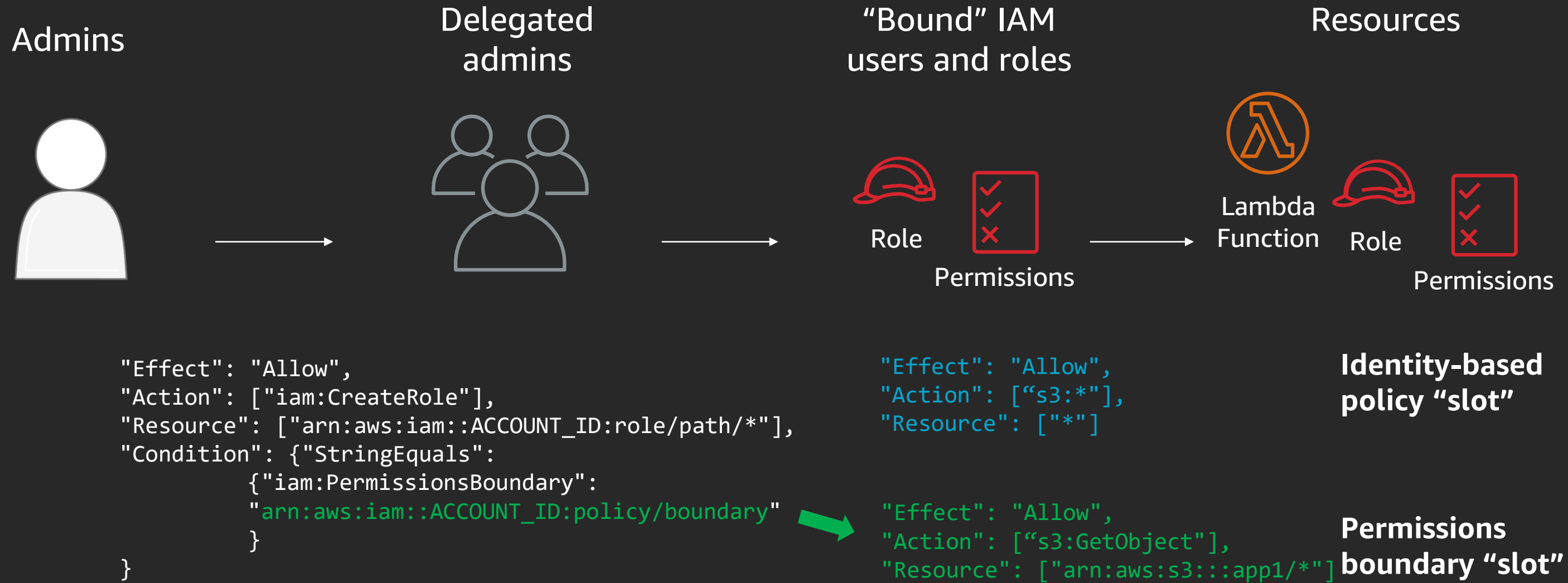
With permission boundaries



```
"Effect": "Allow",
"Action": ["iam:CreateRole"],
"Resource": ["arn:aws:iam::ACCOUNT_ID:role/path/*"],
"Condition": {"StringEquals":
  {"iam:PermissionsBoundary":
    "arn:aws:iam::ACCOUNT_ID:policy/boundary"
  }
}
```

```
"Effect": "Allow",
"Action": ["s3:*"],
"Resource": ["*"]
```

With permission boundaries



Permission policy “slots”

Before Permissions
Boundaries were
launched

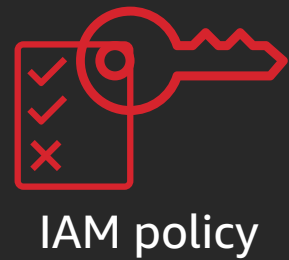


Identity-based policy “slot”



Permission policy "slots"

After Permissions
Boundaries were
launched



Identity-based policy "slot"



Identity-based
policy

Permissions boundary "slot"



Permissions
boundary



**IAM
role**

Permission policy “slots”

▼ Attach permissions policies

Choose one or more policies to attach to your new role.







Create policy

Identity-based policy slot

Filter policies ▼

Search

Showing 582 results

	Policy name ▼	Used as	Description
<input type="checkbox"/>	▶  AdministratorAccess	Permissions policy (9)	Provides full access to AWS services an...
<input type="checkbox"/>	▶  AlexaForBusinessDeviceSetup	None	Provide device setup access to AlexaFor...
<input type="checkbox"/>	▶  AlexaForBusinessFullAccess	None	Grants full access to AlexaForBusiness r...
<input type="checkbox"/>	▶  AlexaForBusinessGatewayExecution	None	Provide gateway execution access to Al...
<input type="checkbox"/>	▶  AlexaForBusinessReadOnlyAccess	None	Provide read only access to AlexaForBu...
<input type="checkbox"/>	▶ AllowAssumeDeleteDDBRole	None	
<input type="checkbox"/>	▶ AllowDeleteofDDbTable	Permissions policy (1)	
<input type="checkbox"/>	▶  AmazonAPIGatewayAdministrator	None	Provides full access to create/edit/delete...

▼ Set permissions boundary

Set a permissions boundary to control the maximum permissions this role can have. This is an advanced feature used to delegate permission management to others. [Learn more](#)

☒ Create role without a permissions boundary

☐ Use a permissions boundary to control the maximum role permissions

Permissions boundary slot

It's just a condition ...

```
"Condition": {"StringEquals":  
  {"iam:PermissionsBoundary":  
    "arn:aws:iam::ACCOUNT_ID:policy/permissionboundary"  
  }  
}
```

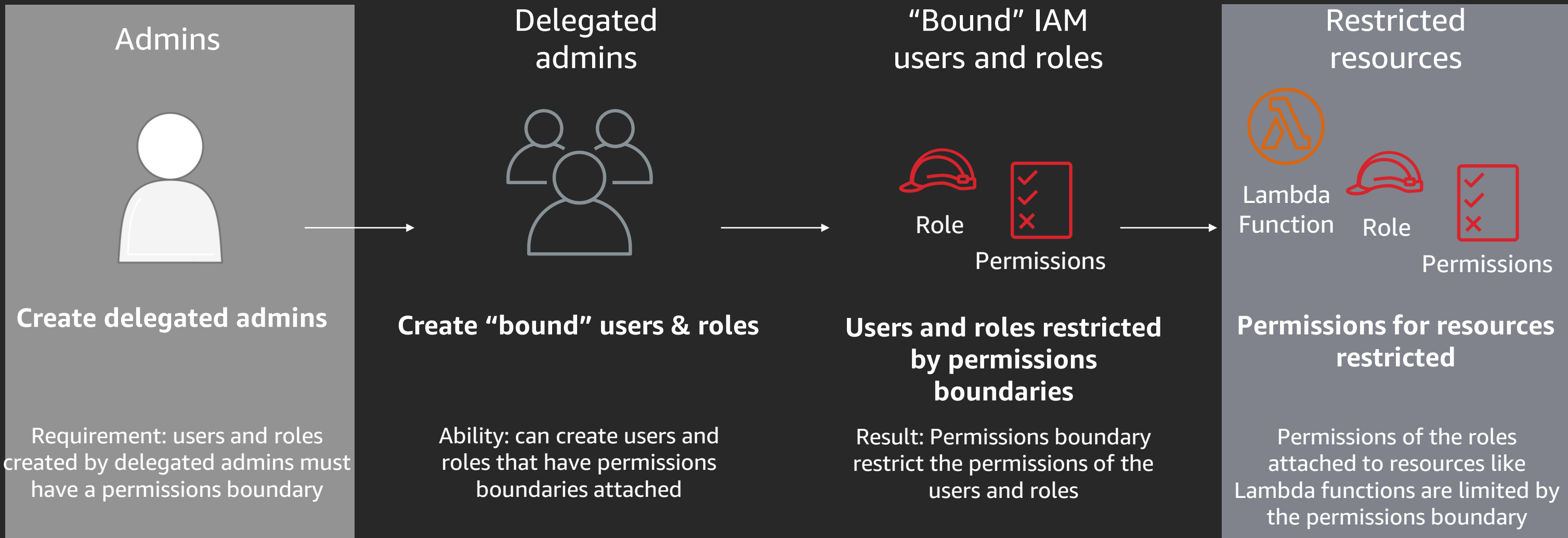
... applied to principal actions

```
"Effect": "Allow",  
"Action": ["iam:CreateRole"],  
"Resource": ["arn:aws:iam::ACCOUNT_ID:role/path/*"],  
"Condition": {"StringEquals":  
    {"iam:PermissionsBoundary":  
        "arn:aws:iam::ACCOUNT_ID:policy/permissionboundary"  
    }  
}
```

Condition key support

- AttachRolePolicy
- AttachUserPolicy
- CreateRole
- CreateUser
- DeleteRolePermissionsBoundary
- DeleteUserPermissionsBoundary
- DeleteRolePolicy
- DeleteUserPolicy
- DetachRolePolicy
- DetachUserPolicy
- PutRolePermissionsBoundary
- PutUserPermissionsBoundary
- PutRolePolicy
- PutUserPolicy

Permissions boundary end-to-end workflow



Developer experience changes little

Create role for a Lambda function

Step 1: Create role and attach permissions boundary

```
$ aws iam create-role --role-name Some_Role --path /Some_Path/  
--assume-role-policy-document file:///Some_Trust_Policy.json  
--permissions-boundary arn:aws:iam::<ACCOUNT_NUMBER>:policy/Permissions_Boundary
```

Step 2: Create identity-based policy

No change

Step 3: Attach identity-based policy

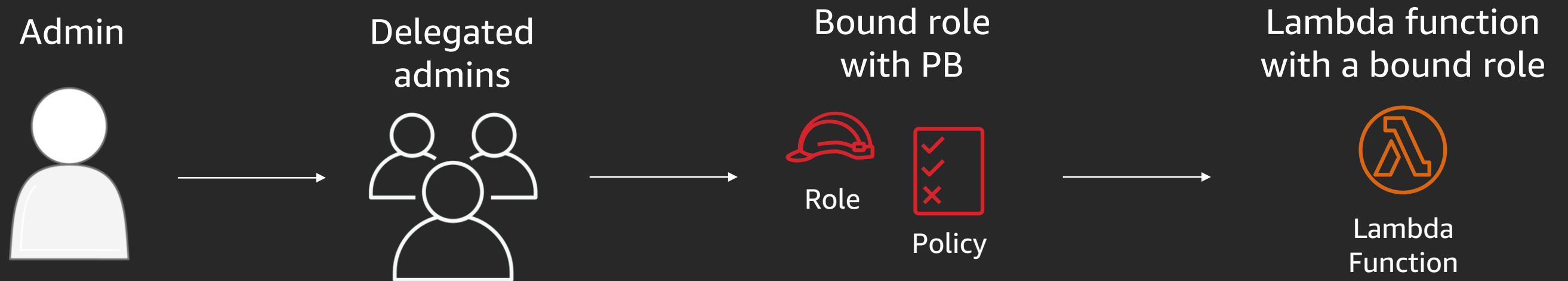
No change

Use cases

- Builders (e.g. creating roles for Lambda functions)
- Application owners creating roles for EC2 instances
- Admins creating users for particular situations
- Any others?

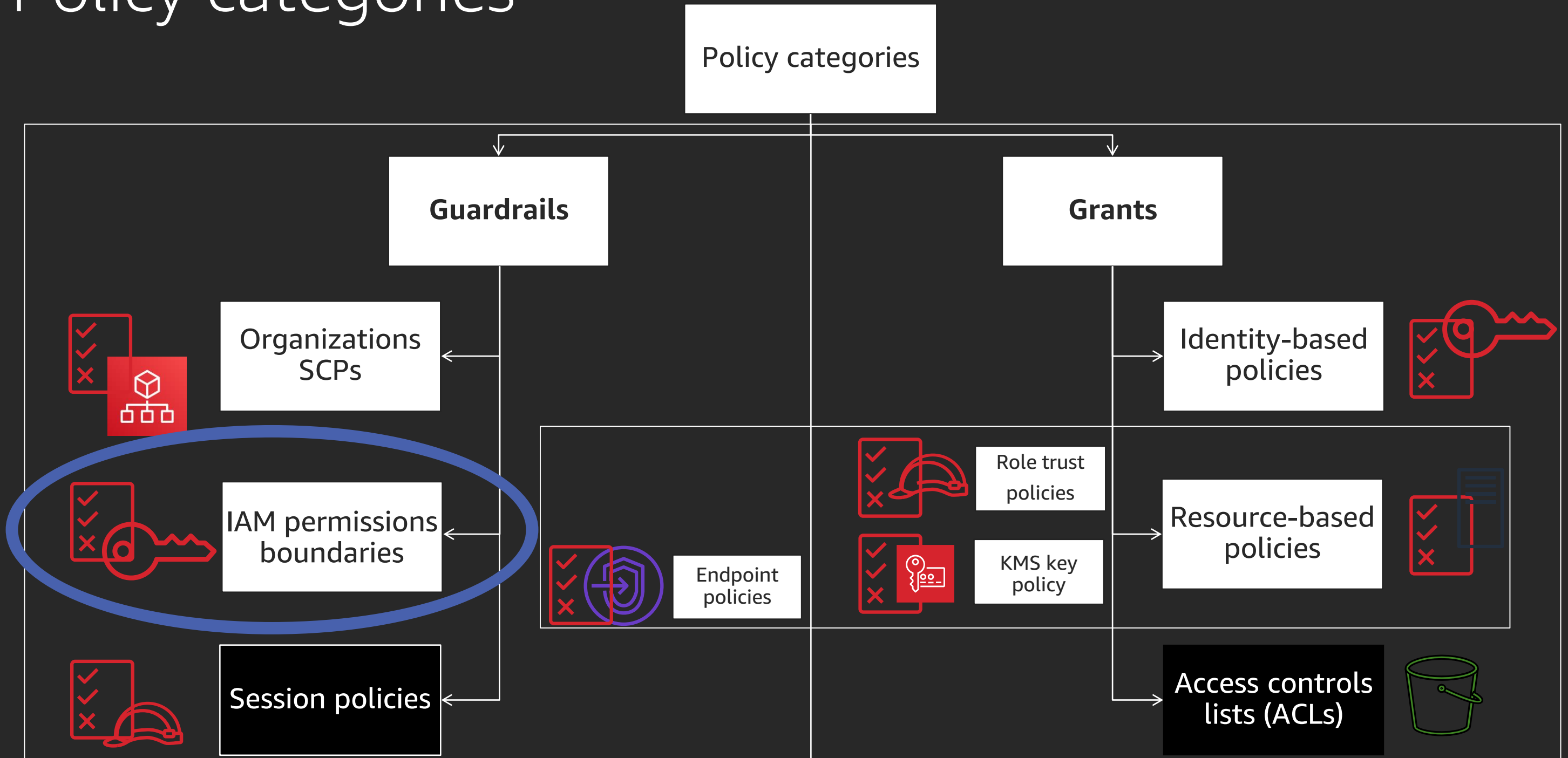
Demo

Demo

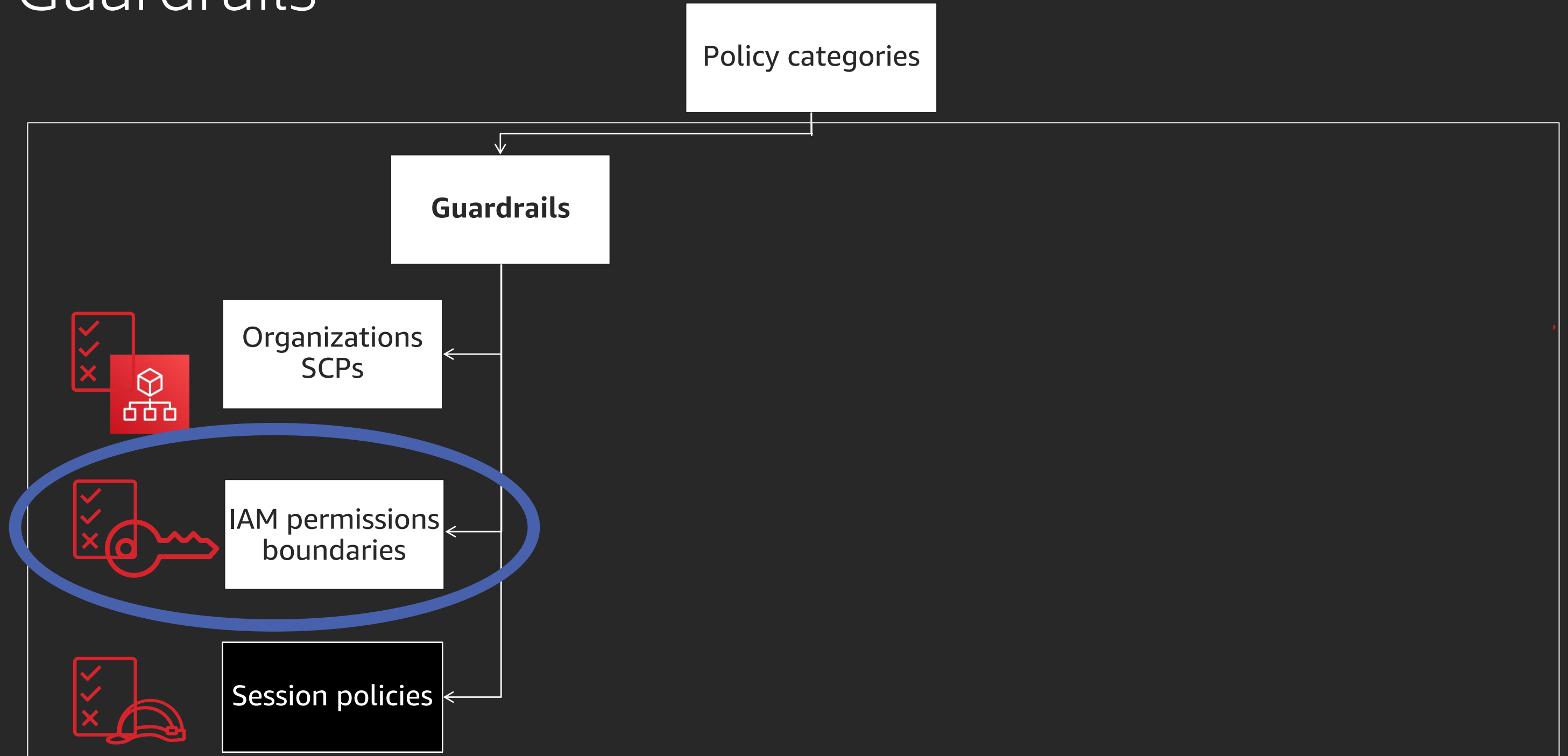


Categories

Policy categories

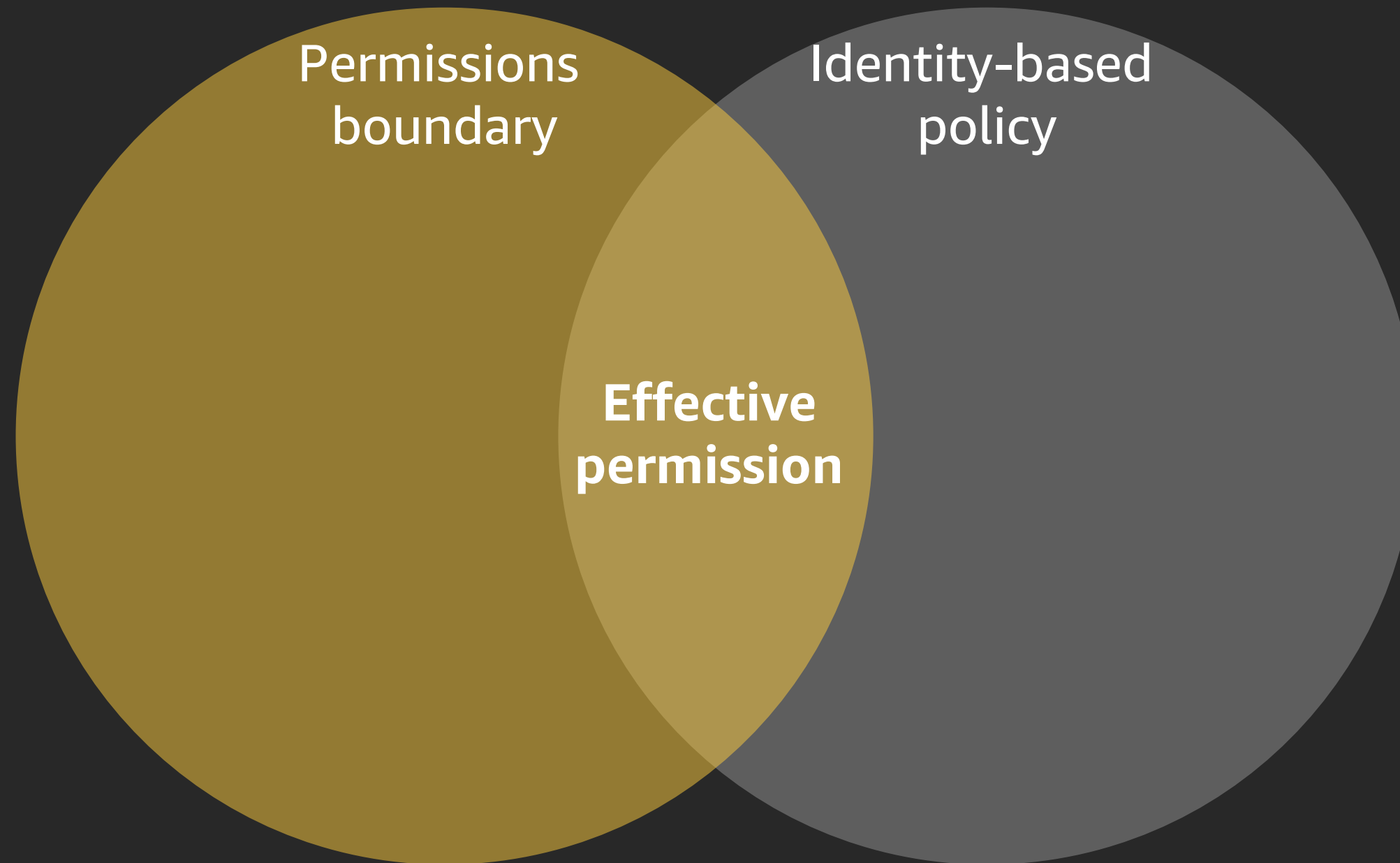


Guardrails

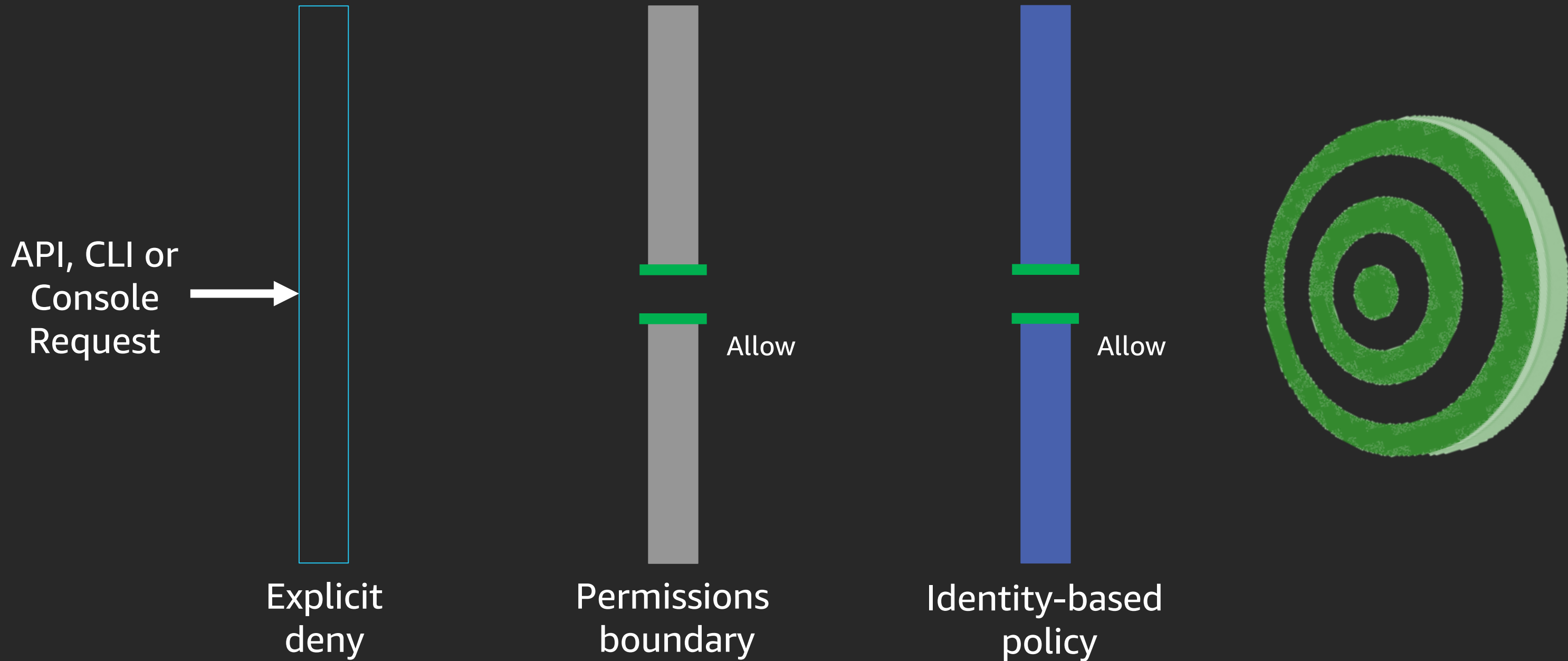


Mechanism

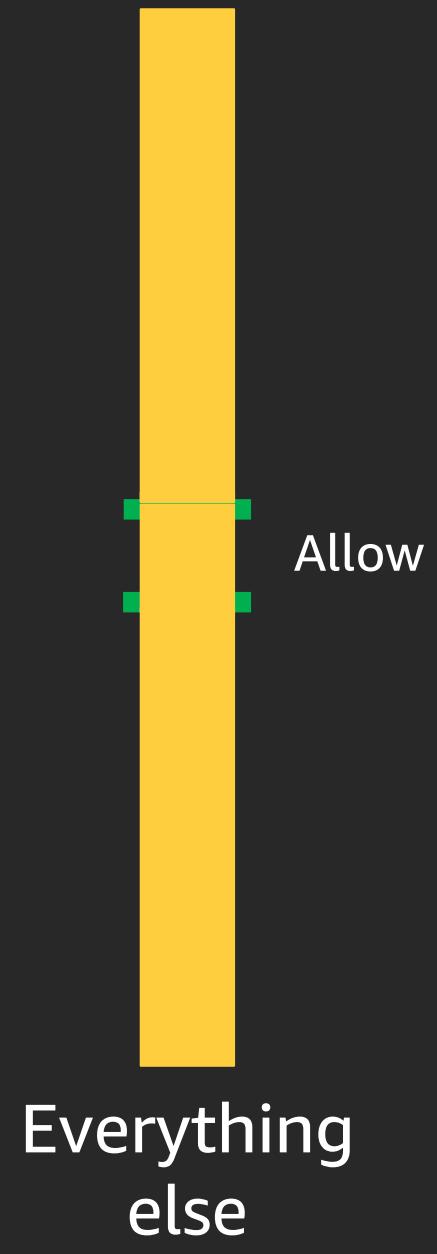
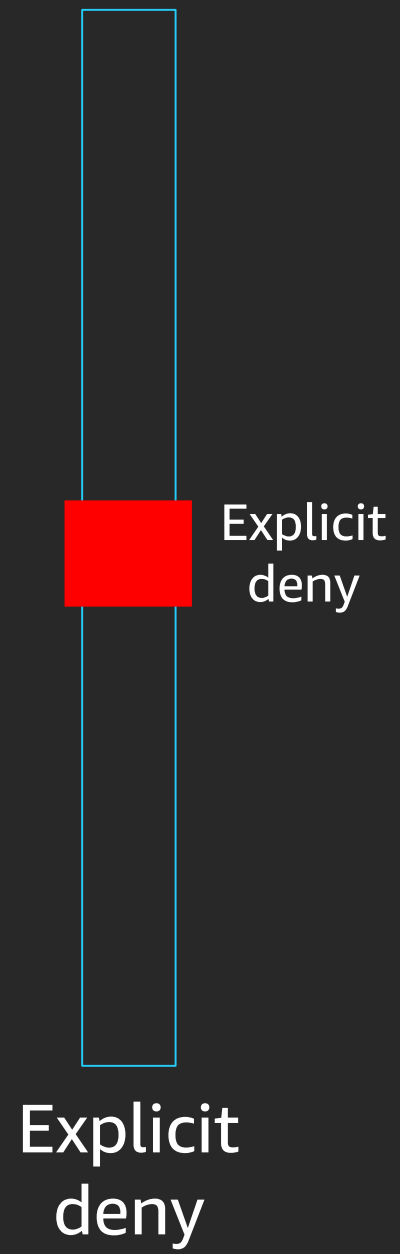
Policy evaluation – Venn diagrams



Trying to hit the target – must go through obstacles



Two types of obstacles



Effective permissions – scenario 1

Request: s3:GetObject / bucket name: example1

Permissions boundary

```
{
  "version": "2012-10-17",
  "statement": [
    {
      "effect": "Allow",
      "action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "resource": "arn:aws:logs:*:*:*"
    }
  ],
}
```

Identity-based policy

```
{
  "version": "2012-10-17",
  "statement": [
    {
      "effect": "Allow",
      "action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "resource": "*"
    },
    {
      "effect": "Allow",
      "action": ["s3:GetObject"],
      "resource": "arn:aws:s3:::example1/*"
    }
  ]
}
```

Effective permissions – scenario 1

Request:
s3:GetObject

API
Request

Request
denied

Implicit
Deny

Allow

Explicit
deny

Permissions
boundary

Identity-based
policy



Effective permissions – scenario 2

Request: s3:GetObject / bucket name: example1

Permissions boundary

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents"
      ],
      "Resource": "arn:aws:logs:*:*:*"
    },
    {
      "Effect": "Allow",
      "Action": ["s3:GetObject"],
      "Resource": "arn:aws:s3:::example1/*"
    }
  ]
}
```

Identity-based policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "logs:CreateLogGroup",
        "logs:CreateLogStream",
        "logs:PutLogEvents",
        "s3:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Effective permissions – scenario 2

Request:
s3:GetObject

API
Request

Explicit
deny

Permissions
boundary

Identity-based
policy

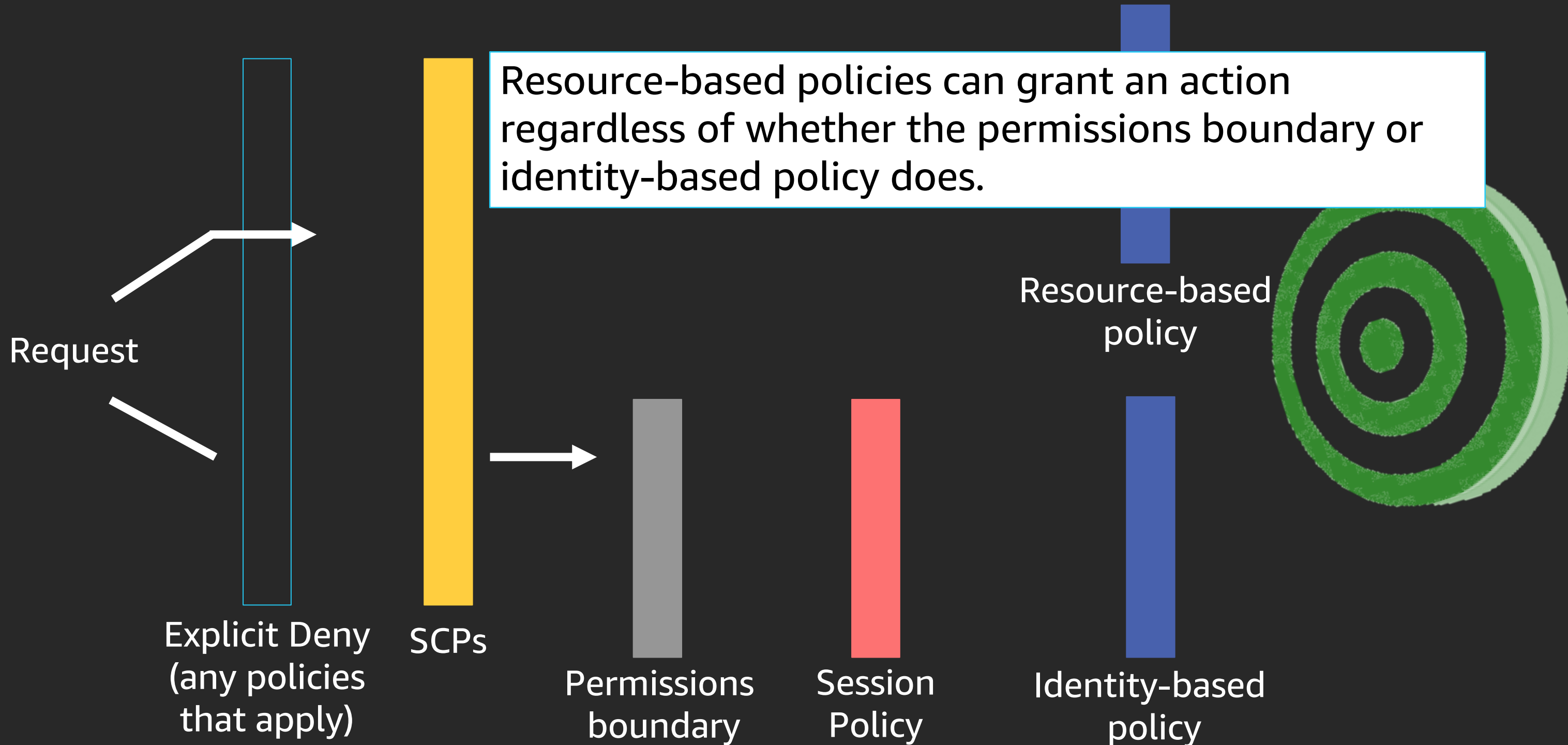
Allow

Allow

Request
allowed



Resource based policies – intra account



Resource restrictions

Resource Restrictions

- Goal: create a “walled garden” for the delegated admins
- Important since not all actions support the permissions boundary condition
- Also allows different teams to safely do delegated permissions management in the same account
- Pathing preferred (requires CLI). “Naming restrictions” can also be used. **Tags are also an option.**

Resource Restrictions - paths

Basic path example:

arn:aws:iam::123456789012:role/**webadmins/????**

Naming example:

arn:aws:iam::123456789012:role/**webadmins***

Resource Restrictions - paths

Role: arn:aws:iam::123456789012:role/**webadmins**

Role with a path: arn:aws:iam::123456789012:role/**namer/webadmins**

Role with paths: arn:aws:iam::123456789012:role/**namer/dept1/webadmins**

Permission:

"Effect": "Allow",

"Action": "iam:DeleteRole",

"Resource": "arn:aws:iam::123456789012:**/namer/dept1/***"

or "Resource": "arn:aws:iam::123456789012:**/namer/***"

Command:

```
aws iam create-role --role-name webadmin --path /namer/dept1/ --  
assume-role-policy-document file://policydoc
```

Pathed walled garden



AWS Account

Web Admins



Webadmins
Role

Create policies
and roles:

/namer/dep1/webadmins/*



Roles: /namer/dep1/webadmins/test-role



Policies: /namer/dep1/webadmins/test-policy

Rest of the account



Other roles: /namer/dep2/test-role



Other policies: admin-policy

Q & A

End of presentation questions

- What is the condition context key used for permissions boundaries?
- What are some of the advantages of using pathing for policies, users and roles?
- What are some permissions boundaries use cases?

Hands on

<https://dashboard.eventengine.run>



Who are you?

1. By using Event Engine for the relevant event, you agree to the [AWS Event Terms and Conditions](#) and the [AWS Acceptable Use Policy](#). You acknowledge and agree that are using an AWS-owned account that you can only access for the duration of the relevant event. If you find residual resources or materials in the AWS-owned account, you will make us aware and cease use of the account. AWS reserves the right to terminate the account and delete the contents at any time.
2. You will not: (a) process or run any operation on any data other than test data sets or lab-approved materials by AWS, and (b) copy, import, export or otherwise create derivate works of materials provided by AWS, including but not limited to, data sets.
3. AWS is under no obligation to enable the transmission of your materials through [AWS Event Engine] and may, in its discretion, edit, block, refuse to post, or remove your materials at any time.
4. Your use of the [event engine] will comply with these terms and all applicable laws, and your access to [AWS Event Engine] will immediately and automatically terminate if you do not comply with any of these terms or conditions.

Team Hash (e.g. abcdef123456)

This is the 12 digit hash that was given to you or your team.

✓ Invalid Hash



Who are you?

1. By using Event Engine for the relevant event, you agree to the [AWS Event Terms and Conditions](#) and the [AWS Acceptable Use Policy](#). You acknowledge and agree that are using an AWS-owned account that you can only access for the duration of the relevant event. If you find residual resources or materials in the AWS-owned account, you will make us aware and cease use of the account. AWS reserves the right to terminate the account and delete the contents at any time.
2. You will not: (a) process or run any operation on any data other than test data sets or lab-approved materials by AWS, and (b) copy, import, export or otherwise create derivate works of materials provided by AWS, including but not limited to, data sets.
3. AWS is under no obligation to enable the transmission of your materials through [AWS Event Engine] and may, in its discretion, edit, block, refuse to post, or remove your materials at any time.
4. Your use of the [event engine] will comply with these terms and all applicable laws, and your access to [AWS Event Engine] will immediately and automatically terminate if you do not comply with any of these terms or conditions.



This is the 12 digit hash that was given to you or your team.

✓ Proceed

Team Dashboard



Event

[Set Team Name](#)



[AWS Console](#)



[SSH Key](#)

Event: relInvent Test 2

Team Name: (Team Name Not Set Yet)

Event ID: 708d2049

Team ID: 22b20db8



Modules

PWW - Workshop



[Readme](#)

Outputs:

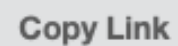
No outputs defined

Region is N. Virginia (us-east-1)

*

Remember to only use "US" as your region!

Login Link



Credentials

[illegible]

OK

Permissions boundaries workshop

Build phase (60 min)

<https://bit.ly/2CMjqmh>

Read through the **Overview**, then click on **Build Phase**:

Follow instructions under “Click here if you are *using your own AWS account ...*”

<https://identity-round-robin.awssecworkshops.com/permission-boundaries-advanced/>

Permissions boundaries workshop
Verify phase (15 min)

<https://bit.ly/2CMjqmh>

Click on **Verify Phase:**

<https://identity-round-robin.awssecworkshops.com/permission-boundaries-advanced/>

Final Q & A

End of workshop questions

- What is the risk of implementing permissions boundaries without resource restrictions?
- What do you attach the permissions boundary to?
- How does a permissions boundary differ from an IAM policy?

Summary

Safely delegate permission management

Let builders build without compromising on security

Also allow multiple teams in the same account to do permission management

Related breakouts

SEC207-L - Leadership session: AWS identity

SEC209-R - [REPEAT] Getting started with AWS identity

SEC316-R - [REPEAT] Access control confidence: Grant the right access to the right things

SEC217-R - [REPEAT] Delegate permissions management using permissions boundaries

Thank you!

Cameron Worrell
Ilya Epshteyn



Please complete the session
survey in the mobile app.