Simplifying your AWS IAM policy using federated identity attributes

Yuri Duchovny
Solutions Architect, AWS

Eran Medan
Sr. Experiences Consultant, AWS

January 27th, 2020

> *“Every program and every user of the system should operate using the least set of privileges necessary to complete the job.”*

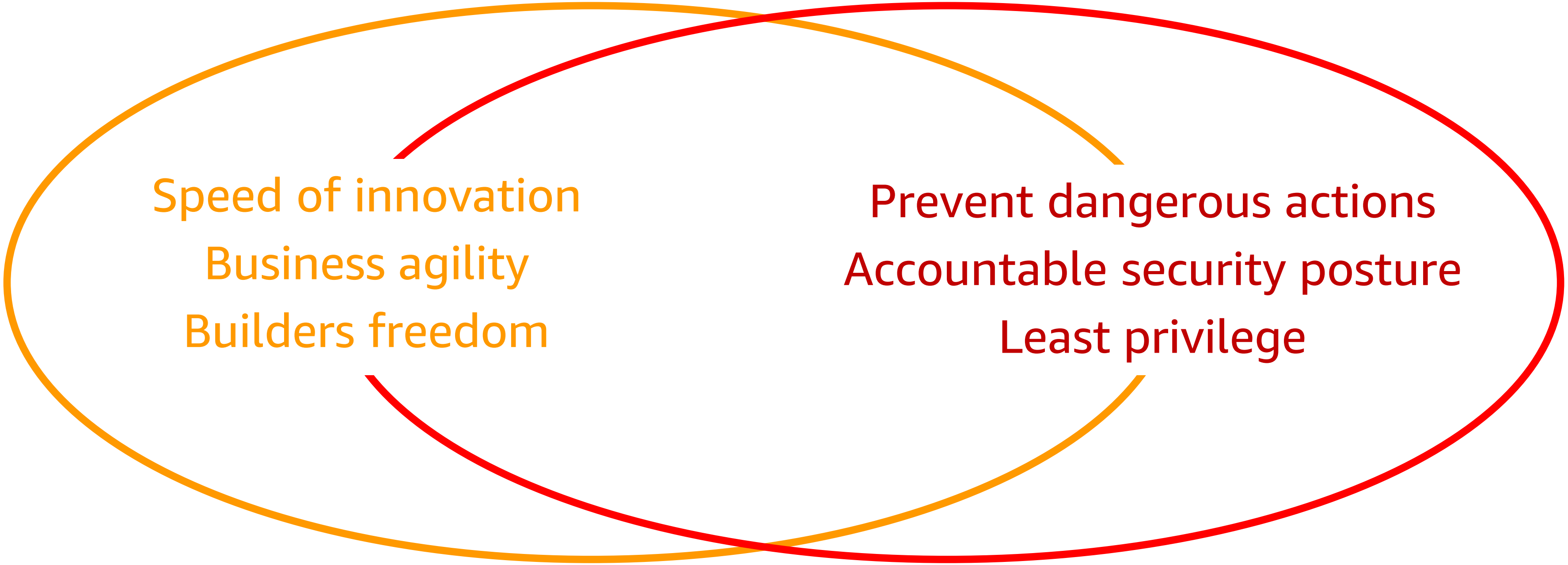
Saltzer, Jerome H. & Schroeder, Michael D. "The Protection of Information in Computer Systems." *Proceedings of the IEEE 63*, 9

September 1975

Access control confidence

What your builders want..

What security needs…



Speed of innovation

Business agility

Builders freedom

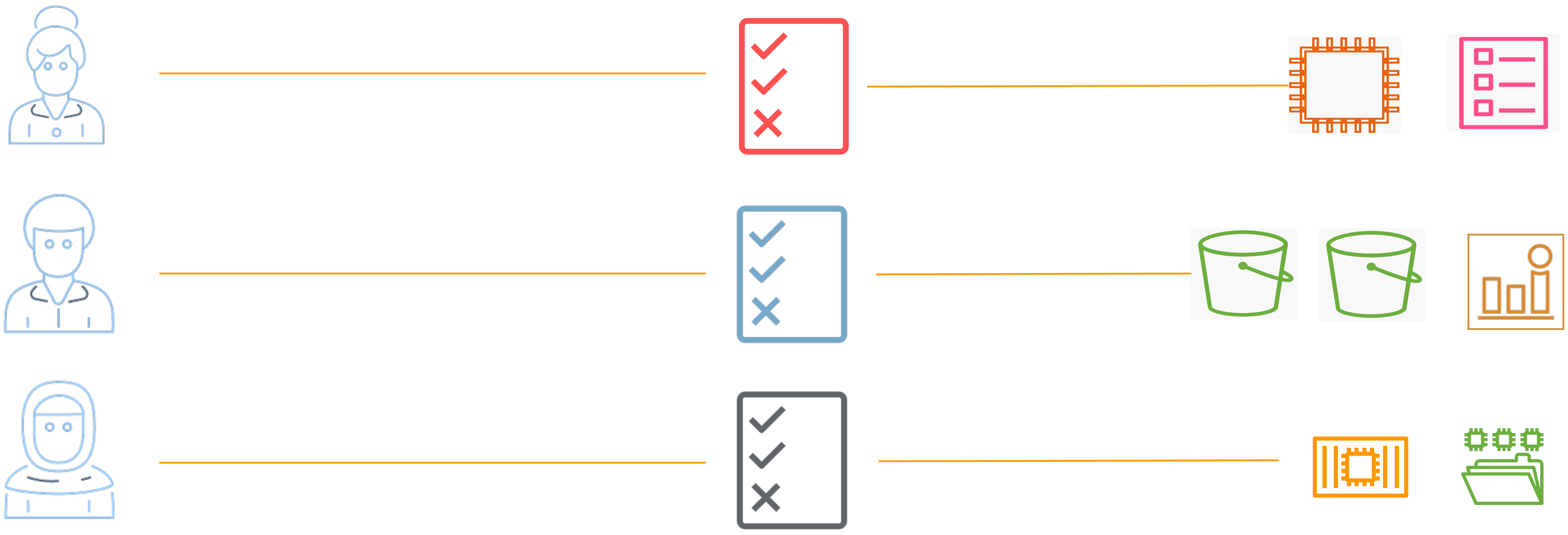
Prevent dangerous actions

Accountable security posture

Least privilege

Role-based access control and Attribute-based access control

Role-based access control (RBAC)

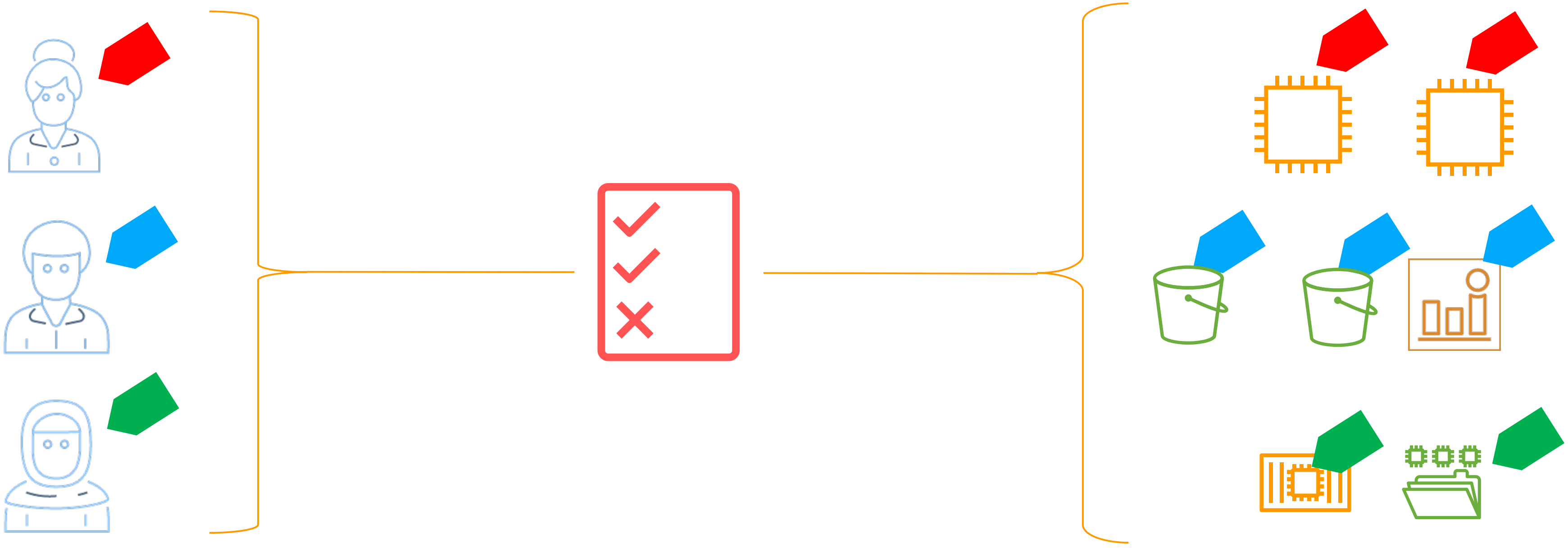


Workforce users

Permissions

Resources

A scalable permissions model based on attributes



Workforce users

Permissions

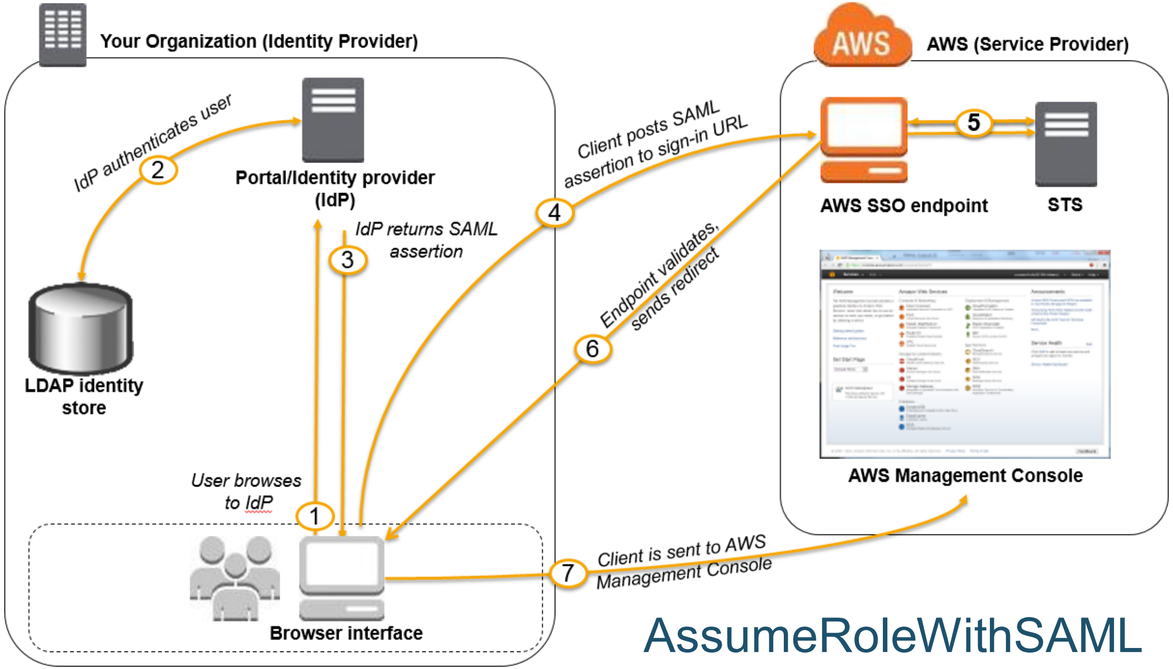
Resources

Examples of attribute-based permissions

- Grant developers read and write access to their project resources
- Require developers to assign their project to new resources
- Grant developers read access to resources that are common to their team
- Manage only the resources that you own

AWS IAM and federated users

SAML 2.0 – based federated users



Your Organization (Identity Provider)
Portal/Identity provider (IdP)
LDAP identity store
Browser interface
User browses to IdP
1
IdP authenticates user
2
IdP returns SAML assertion
3
Client posts SAML assertion to sign-in URL
4
AWS (Service Provider)
AWS
AWS SSO endpoint
STS
5
Endpoint validates, sends redirect
6
AWS Management Console
Client is sent to AWS Management Console
7

AssumeRoleWithSAML

Ping Identity

Auth0

RSA

ForgeRock

okta

IBM Security

onelogin

IAM policies enable granular access controls

```
{
 "Statement":[{
   "Effect":"effect",
   "Principal":"principal",
   "Action":"action",
   "Resource":"arn",
   "Condition":{
     "condition":{
       "key":"value" }
      }
    }
  ]
 }
```

Principal: The entity that is allowed or denied access

"Principal":"AWS":"arn:aws:iam::123456789012:user/username"

Action: Type of access that is allowed or denied

*"Action":"*secretsmanager:GetSecretValue*"*

Resource: The Amazon resource(s) the action will act on

"Resource":"arn:aws:secretsmanager:xx-xxxx-xx:xxx:secret:xxx"

Condition: The conditions that are valid under the access defined

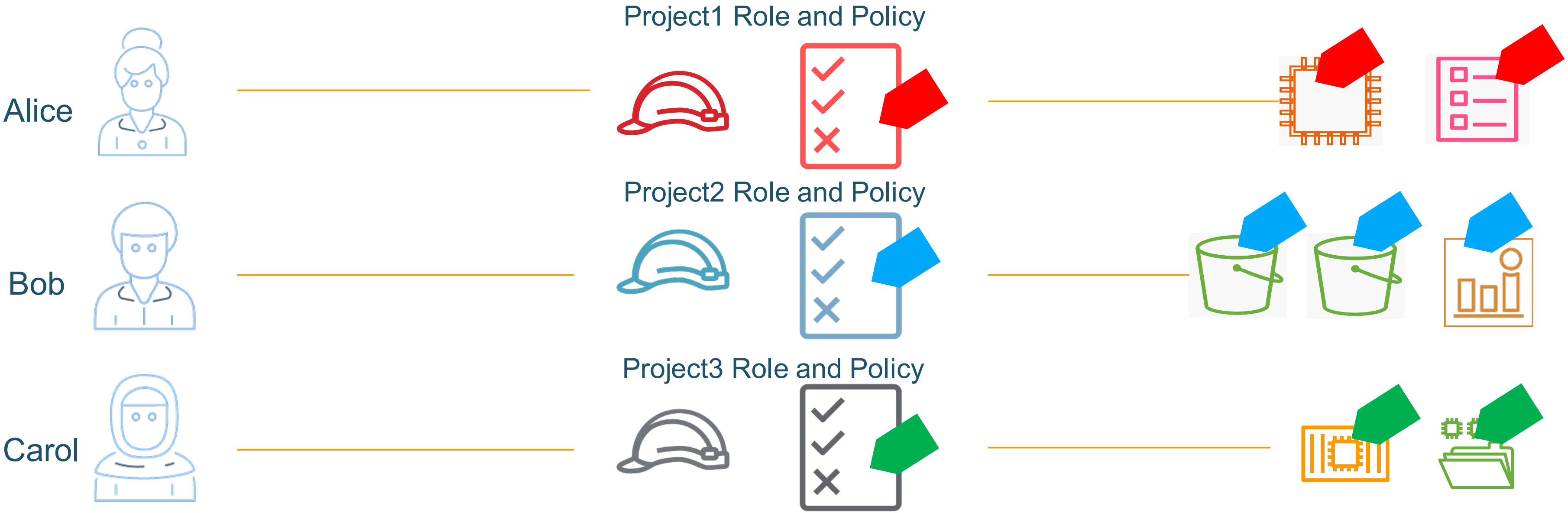
"StringEqua": {"secretsmanager:ResourceTag/Project": "Project1"}

The road to ABAC

Control access explicitly

```
{
  "Effect": "Allow",
  "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:PutSecretValue",
        "secretsmanager:ListSecretVersionIds",
        "secretsmanager:UpdateSecret"
  ],
  "Resource": "arn:aws:secretsmanager:<region>:<acc-id>:secret:<secret-id>"
}
```

ResourceTag/tag-key: tag-value



Project1 Role and Policy

Alice

Project2 Role and Policy

Bob

Project3 Role and Policy

Carol

Workforce users

Permissions

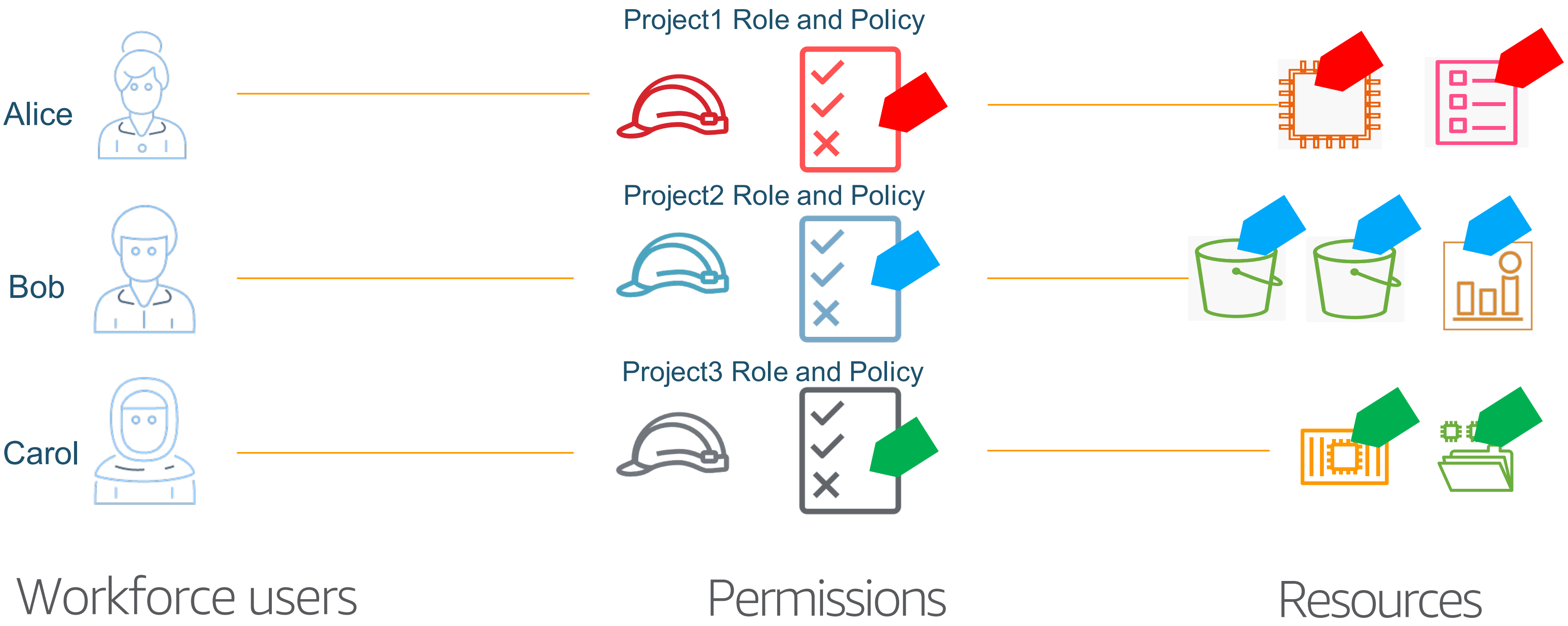
AWS IAM roles & policies

Resources

Access control using ResourceTag

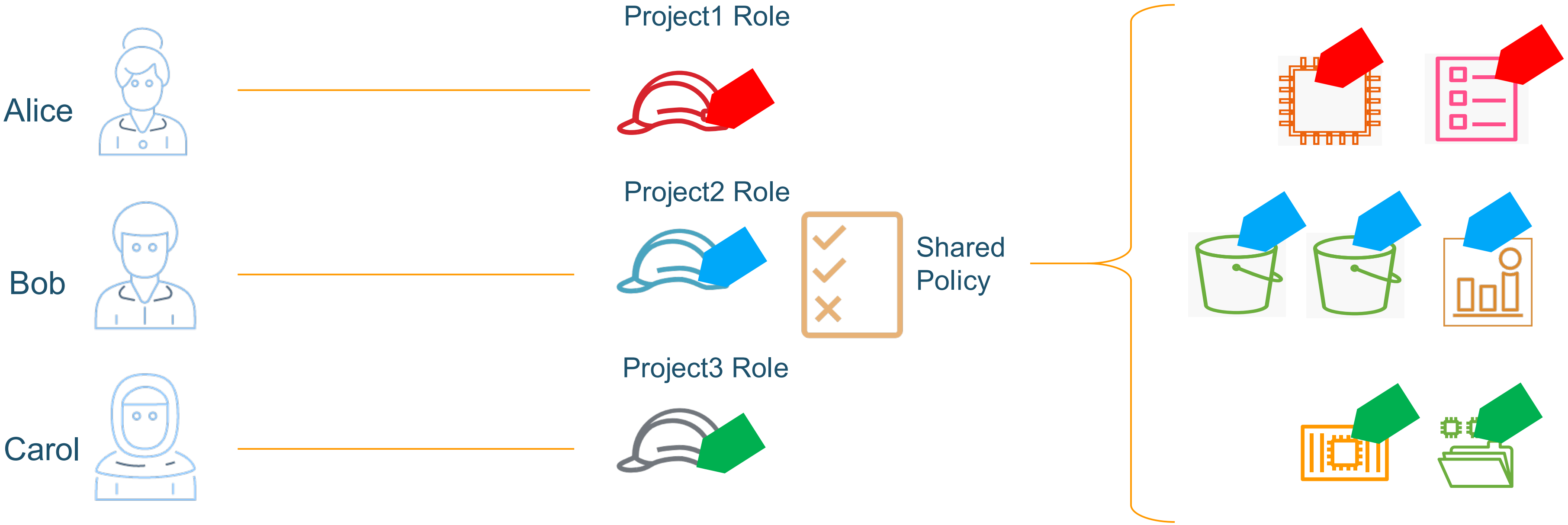
```
{
 "Effect": "Allow",
 "Action": [
       "secretsmanager:GetSecretValue",
       "secretsmanager:DescribeSecret",
       "secretsmanager:PutSecretValue",
       "secretsmanager:ListSecretVersionIds",
       "secretsmanager:UpdateSecret"
 ],
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "secretsmanager:ResourceTag/Project": "Project1"
  }
 }
}
```

ResourceTag/tag-key: tag-value



AWS IAM roles & policies

ResourceTag/tag-key: PrincipalTag/tag-key



Alice

Bob

Carol

Project1 Role

Project2 Role

Project3 Role

Shared Policy

Workforce users

Permissions

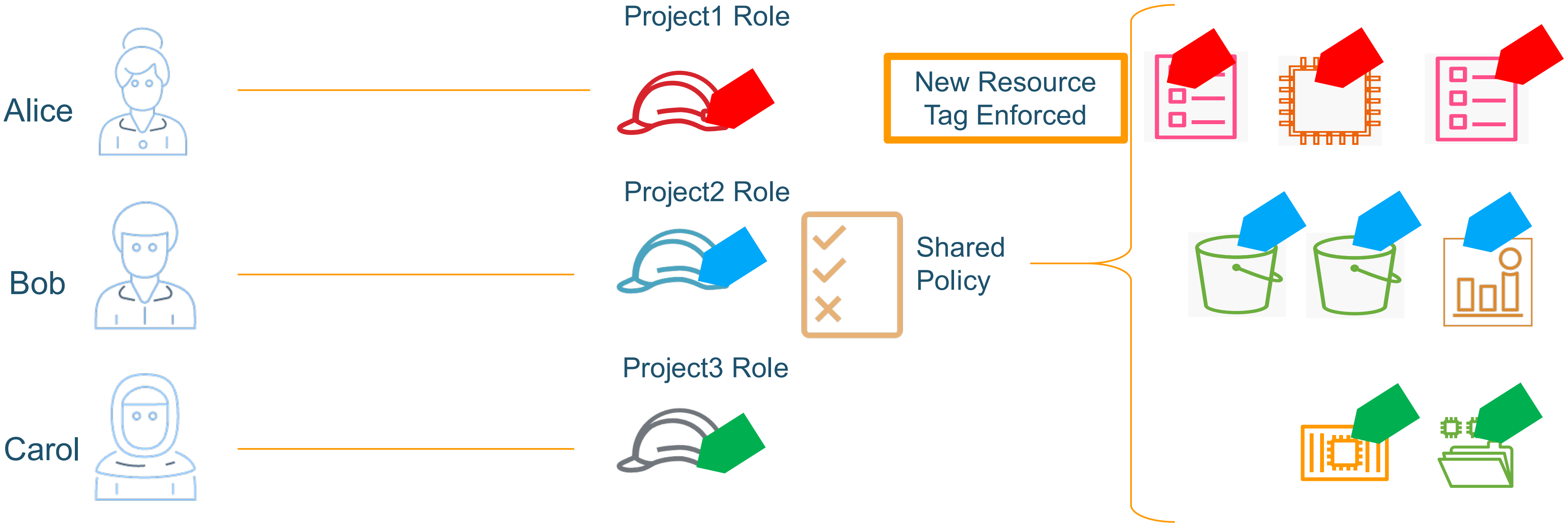
AWS IAM roles & policies

Resources

Access control using ResourceTag and PrincipalTag

```
{
  "Effect": "Allow",
  "Action": [
        "secretsmanager:GetSecretValue",
        "secretsmanager:DescribeSecret",
        "secretsmanager:PutSecretValue",
        "secretsmanager:ListSecretVersionIds",
        "secretsmanager:UpdateSecret"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "secretsmanager:ResourceTag/Project": "${aws:PrincipalTag/Project}"
    }
  }
}
```

RequestTag/tag-key: PrincipalTag/tag-key TagKeys:



Workforce users

Permissions

AWS IAM roles & policies

Resources

Enforcing tag value on create using aws:RequestTag

```
{
  "Effect": "Allow",
  "Action": [
    "secretsmanager:GetSecretValue",
    ...
    "secretsmanager:CreateSecret",
    "secretsmanager:TagResource"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "secretsmanager:ResourceTag/Project": "${aws:PrincipalTag/Project}"
    },
    "StringEqualsIfExists": {
        "aws:RequestTag/Project": "${aws:PrincipalTag/Project}"
    }
  }
}
```

Enforcing allowed tag keys using aws:TagKeys

```
{
 "Effect": "Allow",
 "Action": [
  "secretsmanager:GetSecretValue",
  ...
  "secretsmanager:CreateSecret",
  "secretsmanager:TagResource"
 ],
 "Resource": "*",
 "Condition": {
  "StringEquals": {
   "secretsmanager:ResourceTag/Project": "${aws:PrincipalTag/Project}"
  },
  "StringEqualsIfExists": {
    "aws:RequestTag/Project": "${aws:PrincipalTag/Project}"
  },
  "ForAllValues:StringEquals": {
    "aws:TagKeys": [
      "Project",
      "Name"
    ]
  }
 }
}
```

Enforcing naming convention

```
{
  "Effect": "Allow",
  "Action": [
    "secretsmanager:GetSecretValue",
    ...
    "secretsmanager:CreateSecret",
    "secretsmanager:TagResource"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "secretsmanager:ResourceTag/Project": "${aws:PrincipalTag/Project}"
    },
    "StringEqualsIfExists": {
        "aws:RequestTag/Project": "${aws:PrincipalTag/Project}"
    },
    "ForAllValues:StringEquals": {
        "aws:TagKeys": [
            "Project",
            "Name"
        ]
    },
    "StringLikeIfExists": {
        "secretsmanager:Name": "${aws:PrincipalTag/Project}-*",
        "aws:RequestTag/Name": "${aws:PrincipalTag/Project}-*"
    }
  }
}
```

Session tags for ABAC

Session tags for ABAC

New!

Identity provider is the source of truth

Pass in user attributes as tags specific to each federated AWS session

Permissions automatically apply

Access adjusts as user attributes change or new users are added to your directory

Track user activity

AWS logs attributes in AWS CloudTrail, enabling you to track the user identity for a role session

Ping Identity

Auth0

RSA

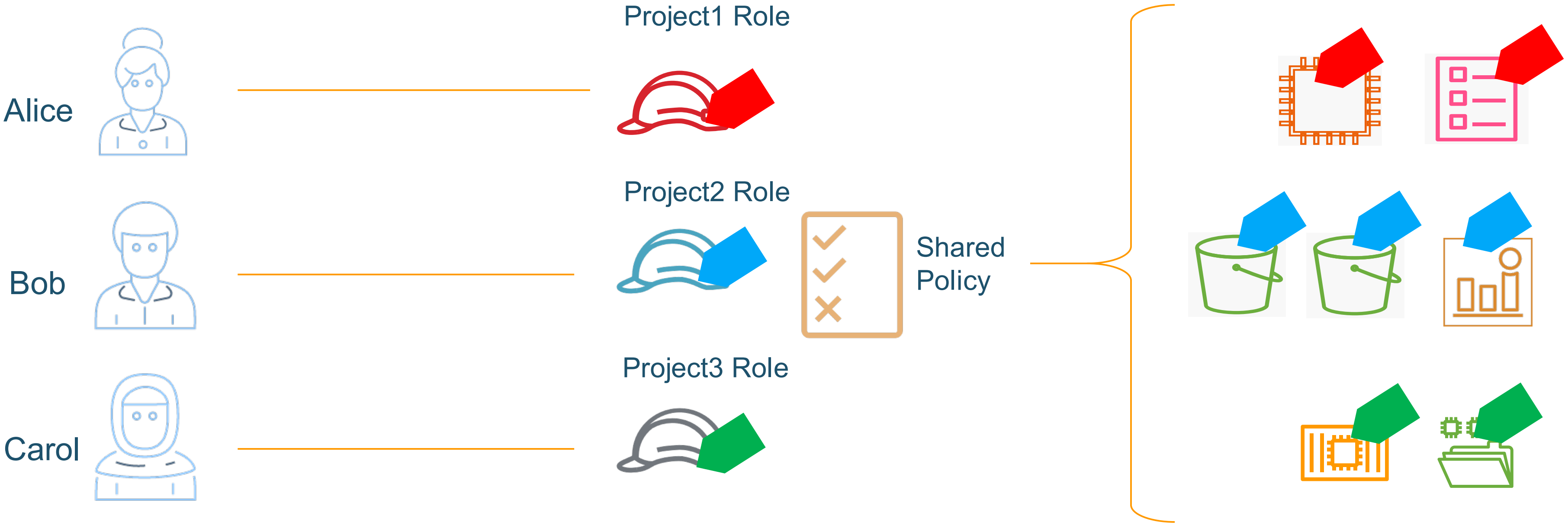
ForgeRock

okta

IBM Security

onelogin

ResourceTag/tag-key == PrincipalTag/tag-key



Alice

Bob

Carol

Project1 Role

Project2 Role

Project3 Role

Shared Policy

Workforce users

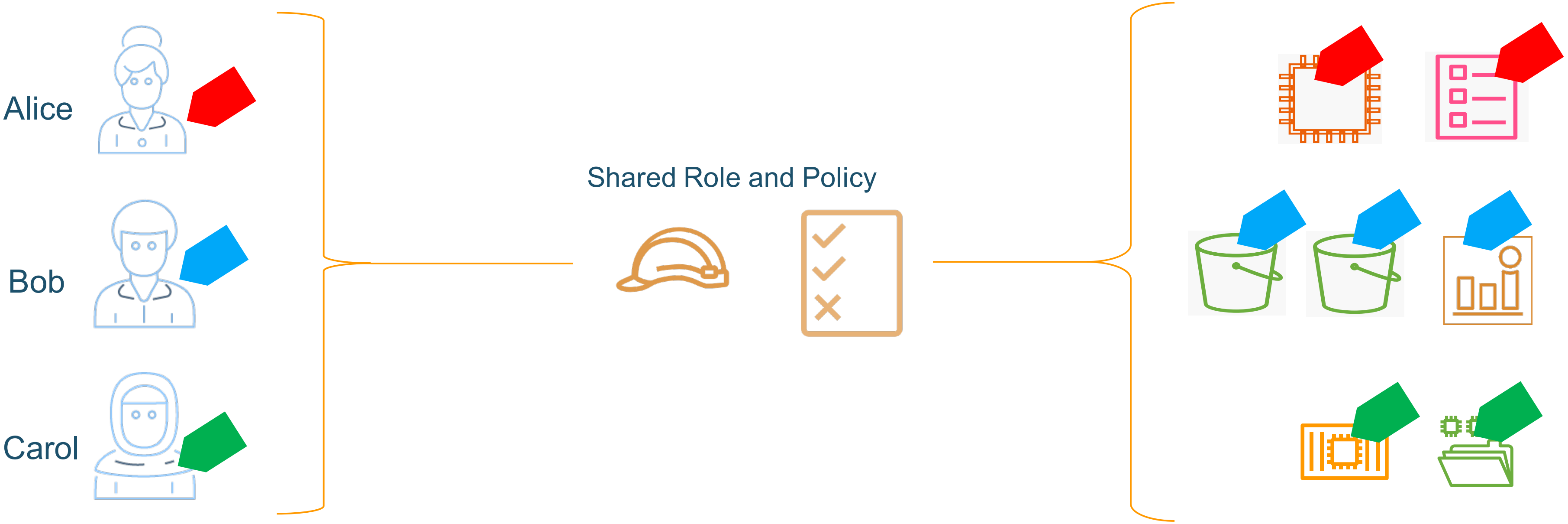
Permissions

AWS IAM roles & policies

Resources

Employee Attributes passed as Session Tags

New!



Workforce users

Permissions

AWS IAM roles & policies

Resources

Trust policy to require specific session tags

```
{
  "Effect": "Allow",
  "Principal": {
    "Federated": "arn:aws:iam::xxxxx:saml-provider/Okta"
  },
  "Action": [
    "sts:AssumeRoleWithSAML",
    "sts:TagSession"
  ],
  "Condition": {
    "StringEquals": {
      "SAML:aud": "https://signin.aws.amazon.com/saml"
    },
    "StringLike": {
      "aws:RequestTag/Project": "*"
    }
  }
}
```

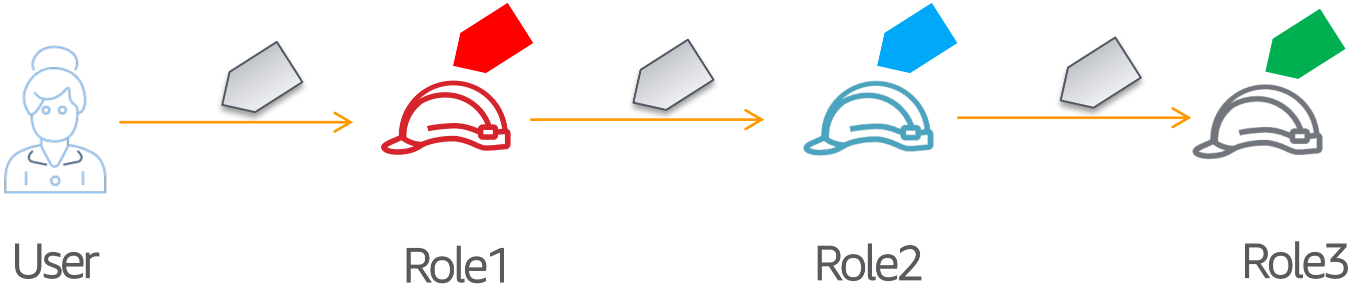
Example SAML assertion to pass in new attributes

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:Project">
    <AttributeValue>Project1<AttributeValue>
</Attribute>
```

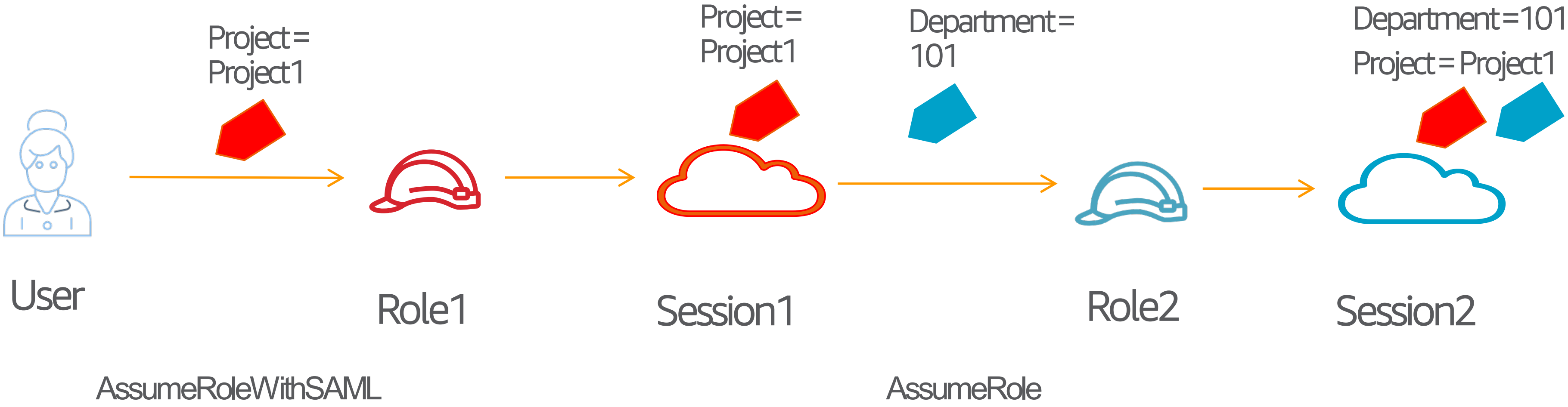
Session attributes in AWS CloudTrail

```
"requestParameters":
       {
       "sAMLAssertionID": ”xxxxx_lbUwCxxxxxx",
       "roleSessionName": "username",
       "principalTags": {
              "project": ”Project1"
        },
       "durationSeconds": 3600,
       "roleArn": "arn:aws:iam::xxxxxx:role/ASM-ABAC",
       "principalArn": "arn:aws:iam::xxxxxx:saml-provider/Okta"
},
```

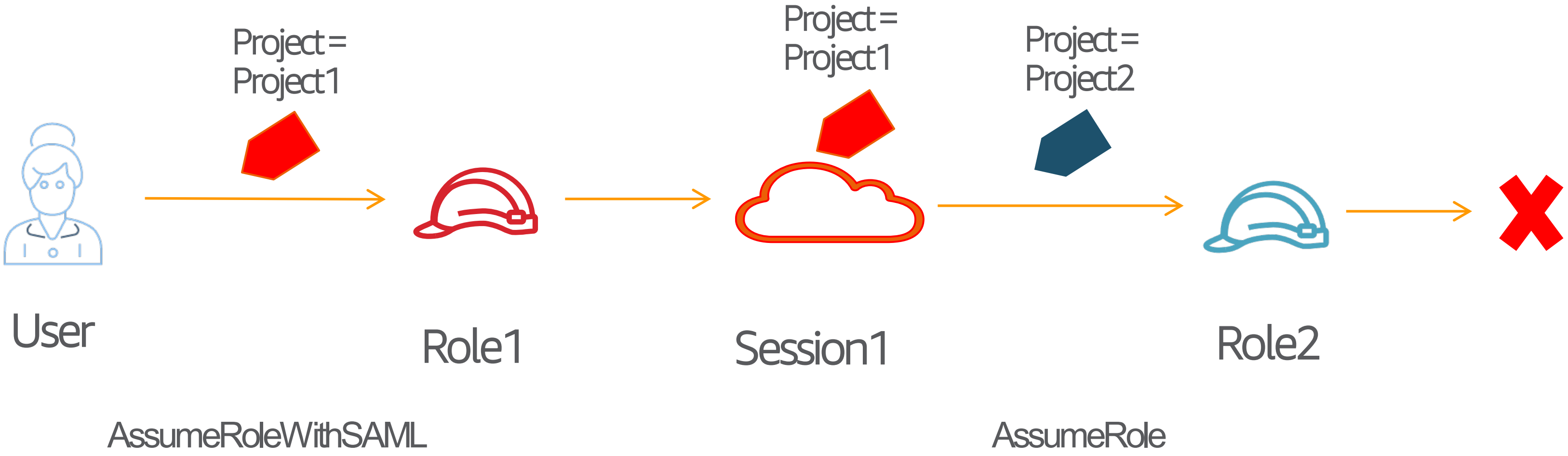
Role Chaining



Role Chaining with Transitive Session Tags



Role Chaining with Transitive Session Tags

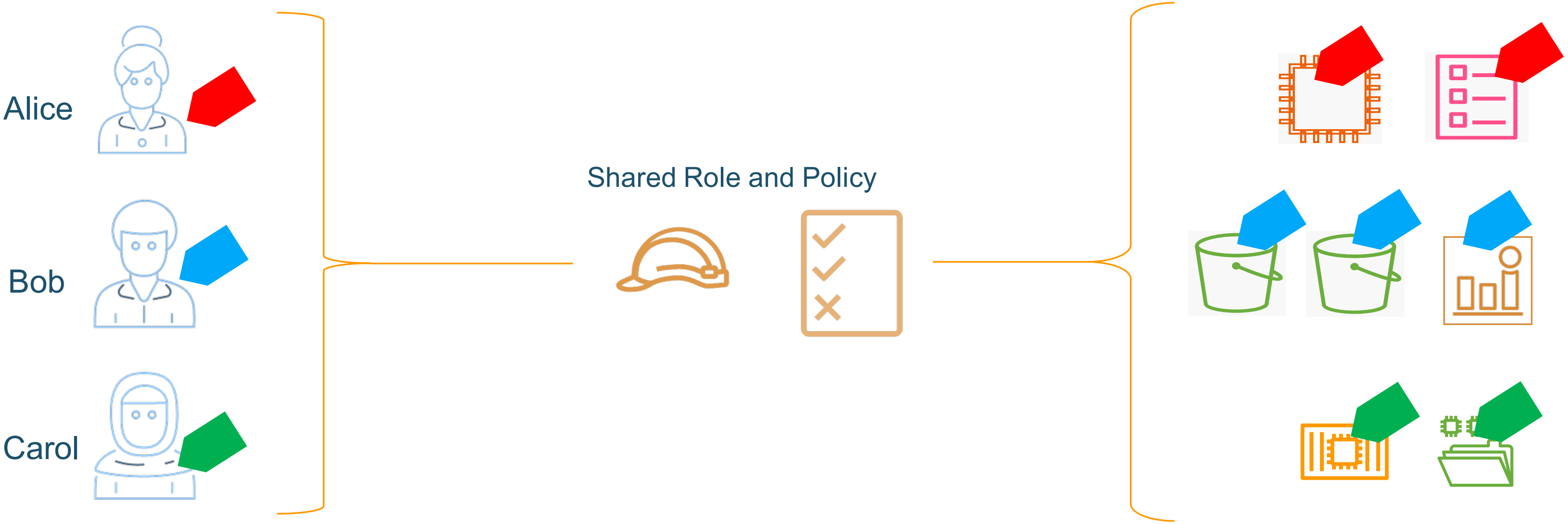


Example SAML assertion to pass in new attributes

```
<Attribute Name="https://aws.amazon.com/SAML/Attributes/PrincipalTag:Project">
    <AttributeValue>Project1<AttributeValue>
</Attribute>

<Attribute Name="https://aws.amazon.com/SAML/Attributes/TransitiveTagKeys">
    <AttributeValue>Project<AttributeValue>
</Attribute>
```

Employee Attributes passed as Session Tags



Workforce users

Permissions

AWS IAM roles & policies

Resources

Additional resources

https://aws.amazon.com/blogs/security/rely-employee-attributes-from-corporate-directory-create-fine-grained-permissions-aws/

https://docs.aws.amazon.com/IAM/latest/UserGuide/tutorial_attribute-based-access-control.html

https://aws.amazon.com/blogs/security/working-backward-from-iam-policies-and-principal-tags-to-standardized-names-and-tags-for-your-aws-resources/

https://docs.aws.amazon.com/IAM/latest/UserGuide/id_session-tags.html#id_session-tags_role-chaining

https://saml-doc.okta.com/SAML_Docs/How-to-Configure-SAML-2.0-for-Amazon-Web-Service

https://www.okta.com/blog/2019/11/okta-and-aws-partner-to-simplify-access-via-session-tags/

https://github.com/oktadeveloper/okta-aws-cli-assume-role

Thank you!

Yuri Duchovny
Solutions Architect, AWS
dyuri@amazon.com

Eran Medan
Sr. Experiences Consultant, AWS
eranmeda@amazon.com