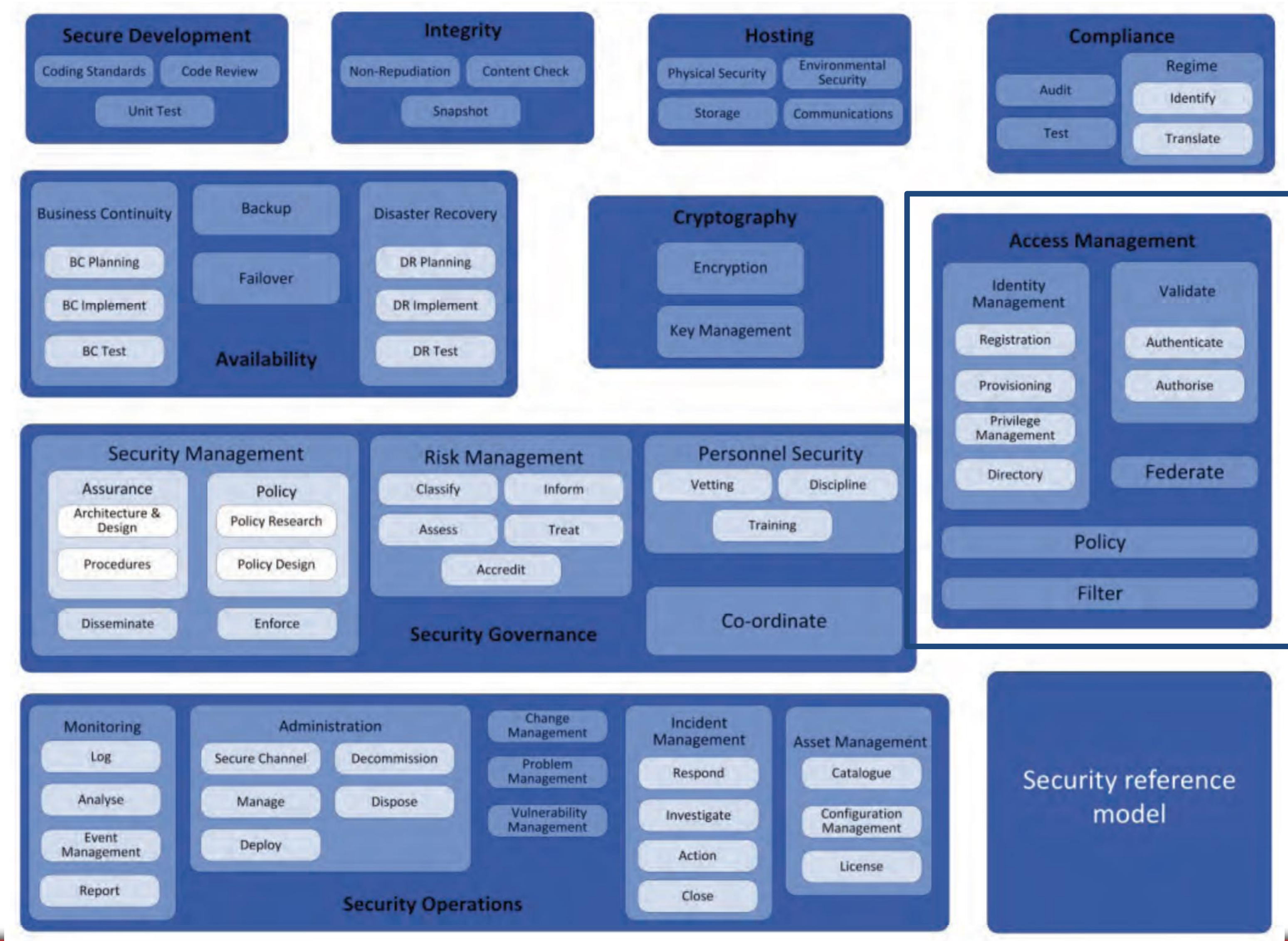
COMPX527 Week 5 1& 2

Identity and Access Management

Identity and Access Management



Secure Development
- Coding Standards
- Code Review
- Unit Test

Integrity
- Non-Repudiation
- Content Check
- Snapshot

Hosting
- Physical Security
- Environmental Security
- Storage
- Communications

Compliance
- Audit
- Test
- Regime
  - Identify
  - Translate

Availability
- Business Continuity
  - BC Planning
  - BC Implement
  - BC Test
- Backup
- Failover
- Disaster Recovery
  - DR Planning
  - DR Implement
  - DR Test

Cryptography
- Encryption
- Key Management

Access Management
- Identity Management
  - Registration
  - Provisioning
  - Privilege Management
  - Directory
- Validate
  - Authenticate
  - Authorise
- Federate
- Policy
- Filter

Security Governance
- Security Management
  - Assurance
    - Architecture & Design
    - Procedures
  - Policy
    - Policy Research
    - Policy Design
  - Disseminate
  - Enforce
- Risk Management
  - Classify
  - Inform
  - Assess
  - Treat
  - Accredit
- Personnel Security
  - Vetting
  - Discipline
  - Training
- Co-ordinate

Security Operations
- Monitoring
  - Log
  - Analyse
  - Event Management
  - Report
- Administration
  - Secure Channel
  - Decommission
  - Manage
  - Dispose
  - Deploy
- Change Management
- Problem Management
- Vulnerability Management
- Incident Management
  - Respond
  - Investigate
  - Action
  - Close
- Asset Management
  - Catalogue
  - Configuration Management
  - License

Security reference model

Top Threat to Cloud Computing

Identity and Access Management

If entities are not properly identified and what they are permitted to do is not clearly defined, then the chance of a data or system breach increases dramatically.

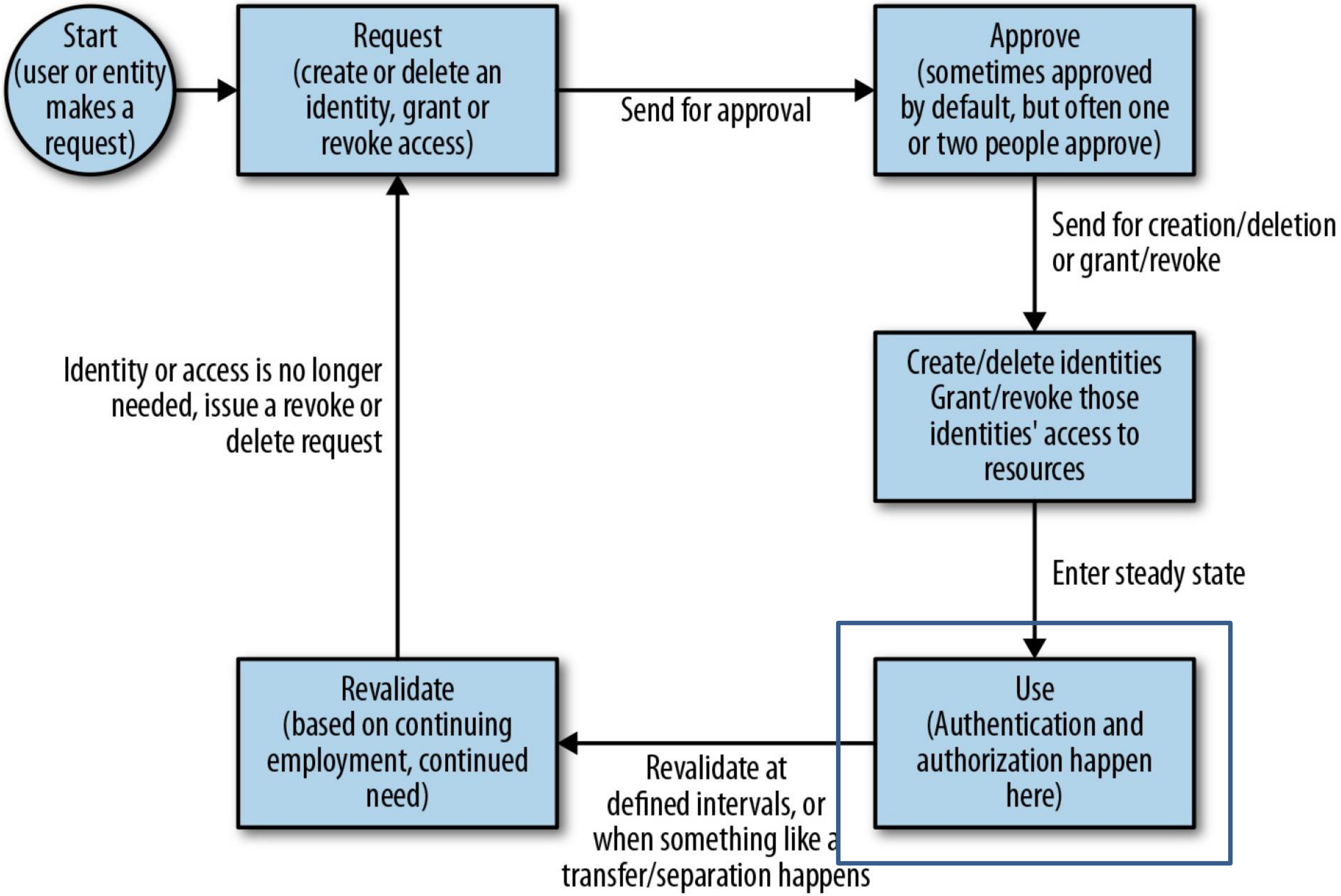
Identity and Access Management

- Identity and access management (IAM) are often discussed together, but it’s important to understand that they are two distinct concepts:
  - Each entity (such as a user, administrator, or an application) needs an identity. The process of verifying that identity is called **authentication**
  - Access management is about ensuring that entities can perform only the tasks they need to perform. The process of checking what access an entity should have is called **authorisation**

….but wait

- We do Authentication and Authorization in non-cloud environments too.
- In cloud environments
  - No Perimeter
    - No physical controls, and no perimeter firewalls
  - Improper access control on a service exposes it to the entire world
  - In addition to authenticating users and employees for your business, you also authenticate to the cloud services.

Identity Life Cycle



Start (user or entity makes a request)

Request (create or delete an identity, grant or revoke access)

Send for approval

Approve (sometimes approved by default, but often one or two people approve)

Send for creation/deletion or grant/revoke

Create/delete identities Grant/revoke those identities' access to resources

Enter steady state

Use (Authentication and authorization happen here)

Revalidate at defined intervals, or when something like a transfer/separation happens

Revalidate (based on continuing employment, continued need)

Identity or access is no longer needed, issue a revoke or delete request

Authentication

- Three levels of authentication in cloud
  - Authenticating your organization’s employees with your cloud providers
    - generically business-to-business, and often called something like “Cloud IAM” by cloud providers
  - Authenticating your organization’s employees with your own applications
    - Called business-to-employee
  - Authenticating your organization’s customers with your own applications
    - Called business-to-consumer

Business-to-Consumer & Business-to-Employee

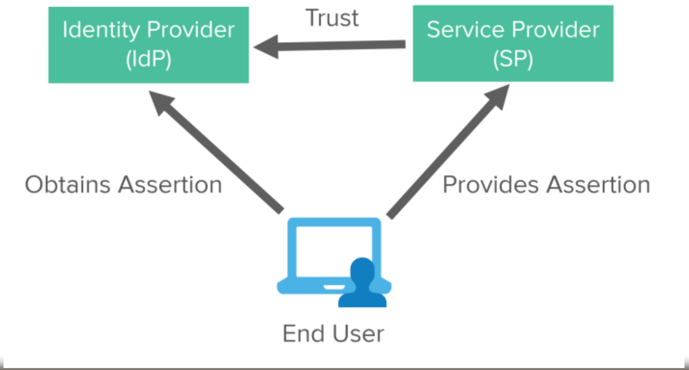
- Setup your own ID Management System
  - Users may have to juggle yet another set of credentials
  - Sometimes necessary

- Use an external identity service. This may be an internal identity service for your employees or your customer’s employees. For end customers, it may also be an external service such as Facebook, Google, or LinkedIn.

- Use customer identities specific to your application, and use a cloud service to manage these customer identities.

External Identity providers

- An identity provider (IdP) is a service that stores and manages digital identities
- You may want to give your users the freedom to choose which Identity Provider they use to sign in to your application
- Helps avoid password fatigue for the users
- Greater security by leveraging resources of experts
- Must learn how to use the APIs securely!
- Must understand how the IdP validates identity.

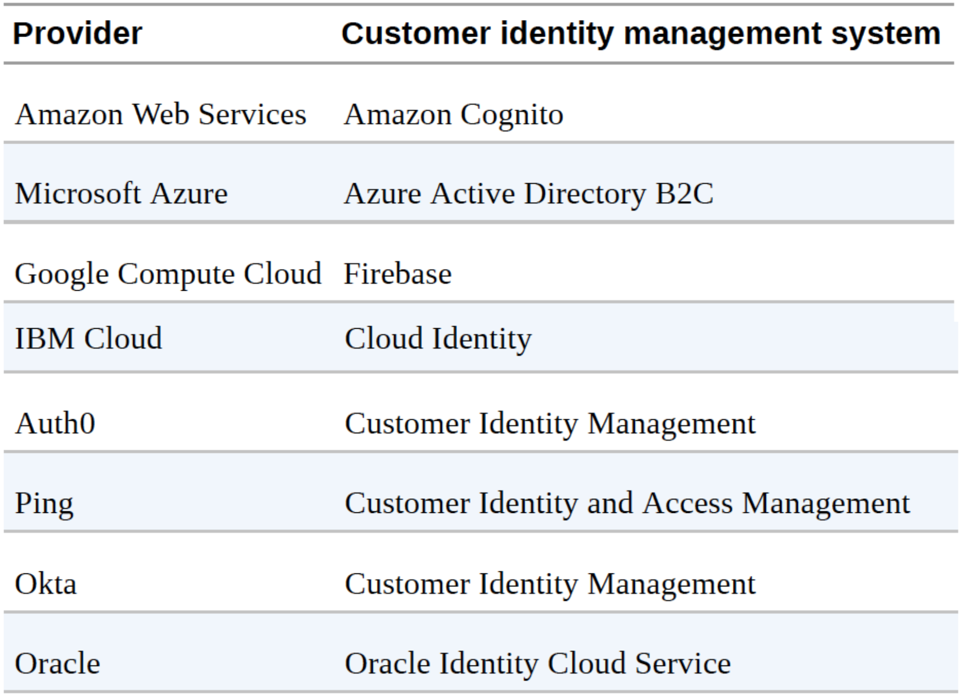


Cloud IAM

- Many cloud providers offer IAM services at no additional charge for accessing their cloud services.
- These systems allow you to have one central location to manage identities of cloud user and administrators in your organization
- Enables you to manage the access that you have granted those identities to all of the services that cloud provider offers.

ID as a service

- IDaaS services manage user identities and provide integration with cloud services



Provider	Customer identity management system
Amazon Web Services	Amazon Cognito
Microsoft Azure	Azure Active Directory B2C
Google Compute Cloud	Firebase
IBM Cloud	Cloud Identity
Auth0	Customer Identity Management
Ping	Customer Identity and Access Management
Okta	Customer Identity Management
Oracle	Oracle Identity Cloud Service

Authenticating Applications

- Four different ways
  - Approach 1
    - Store credentials in the deployment system, using features designed to hold secrets.
    - Tightly control access to the deployment system
    - Nobody sees the credentials by default, and only authorized individuals have the technical ability to view or change them in the deployment system.

Authenticating Applications

- Approach 2
  - Use a credentials server to hold credentials.
  - Either the deployment server or the deployed application contacts the credentials server to get the necessary credentials and use them.
  - In many cases the credentials are still visible in the configuration files of the running application after deployment, so operations personnel may be able to easily view them.
- Approach 3
  - Deployment server only able to get a one time token and pass it to the application, which then retrieves the credentials and holds them in memory.
  - This protects you from having the credentials to the secrets server or the application credentials themselves intercepted.
- Approach 4
  - The cloud provider provides trusted identity documents and metadata that the credentials server can use to decide which credentials to provide to each application.

Authorization

- Once you’ve completed the authentication phase and you know who your users are, it’s time to make sure they are limited to performing only the actions they are supposed to perform. Some examples of authorization may be
  - Permission to access an application at all
  - Permission to access an application with write access
  - Permission to access a portion of the network
  - Permission to access the cloud console etc.
- The most important concepts to remember for authorization are least privilege and separation of duties

Least Privilege

- Least Privilege
  - The principle of least privilege simply states that people or applications should be able to access only what they need to do their jobs, and no more.
  - A practical application of least privilege often means that your access policies are deny by default.

Separation of Duties

- Separation of duties
  - Ensuring that one individual does not have all necessary permissions to be able to complete a malicious action.
    - this could be an action such as using a key to access and decrypt data which that user should not normally have access to
  - Practically it means that more than one person is needed to complete a task that involves sensitive information such as granting or revoking access to services.

Policy

- Access Control is performed using policies
- A policy defines **who** can perform what action on **which resource** and under **what condition**
- Policies often have two parts
  - Specification: define the access policy
  - Enforcement: evaluate and implement the policy

Types of Policies

- Identity based policies
- Resource based policies
- Role based policies
- Attribute based policies

- CSPs often have variations of these as well as other types of policies specific to their cloud

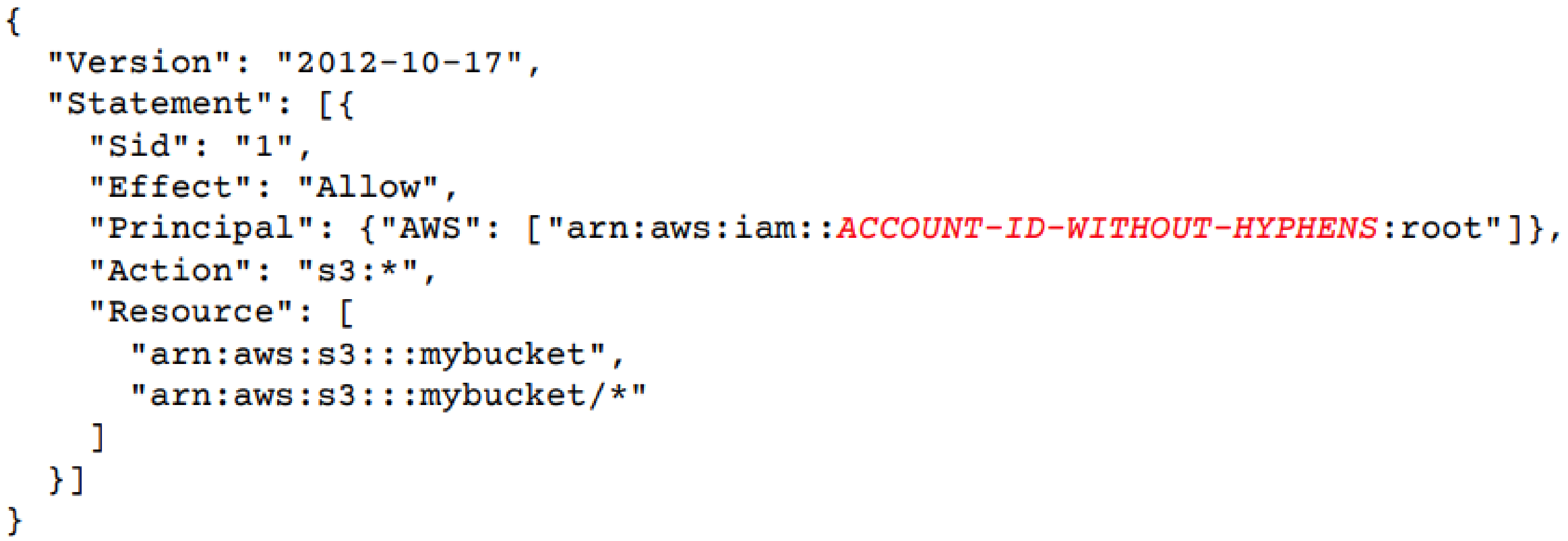
Identity based policies

- Identity-based policies control what actions an identity (users, groups of users, and roles) can perform, on which resources, and under what conditions

```
{
    "Version": "2025-8-6",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:ListAllMyBuckets",
            "Resource": "*",
            "Condition": {
                "DateGreaterThan": {"aws:CurrentTime": "2025-04-01T00:00:00Z"},
                "DateLessThan": {"aws:CurrentTime": "2026-06-30T23:59:59Z"}
            }
        }
    ]
}
```

Resource Based Policies

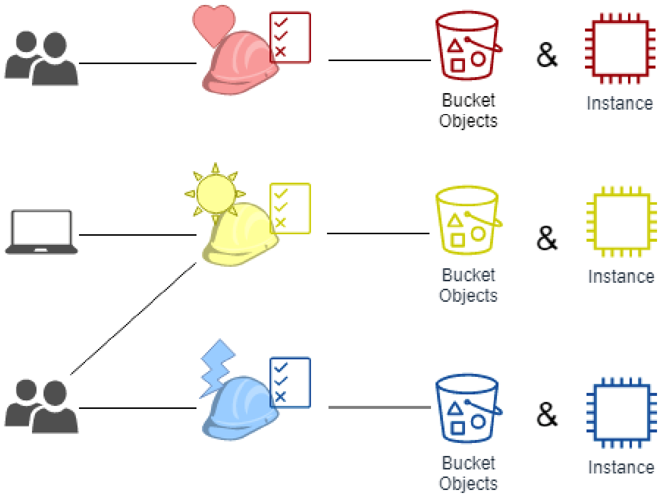
- Resource-based policies are attached to specific resources such as an Amazon S3 bucket. These policies grant the specified principals permissions to perform specific actions on that resource and defines under what conditions this applies.



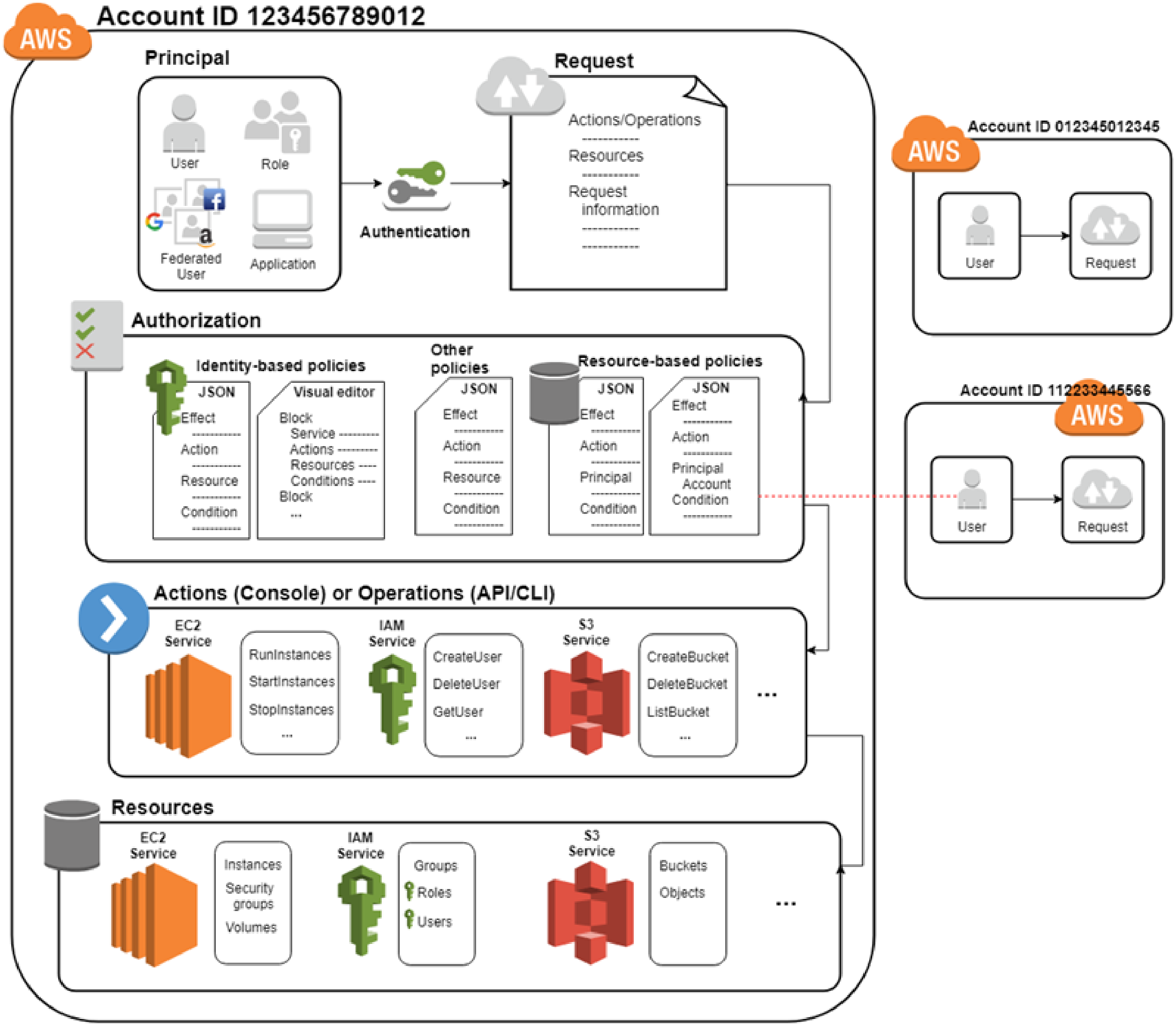
```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "1",
    "Effect": "Allow",
    "Principal": {"AWS": ["arn:aws:iam::ACCOUNT-ID-WITHOUT-HYPHENS:root"]},
    "Action": "s3:*",
    "Resource": [
      "arn:aws:s3:::mybucket",
      "arn:aws:s3:::mybucket/*"
    ]
  }]
}
```

Role Based Access Control

- Create Roles and apply permissions to roles rather than individual identities
- Assign roles to identities
- Many to many relationship between roles and identities



IAM infrastructure



Terminology

- AWS Account
  - Has a 12-digit account ID
  - Controls its created resources
  - Pays for AWS activity for those resources
- Principal
  - A user, service, or account that can make a request for an action on a resource
  - Used in resource-based policies

Terminology

- IAM User
  - – Entity created inside an account
  - – Can be used by another person or an application
  - – Has distinct security credentials, including username and password.
- IAM Roles
  - – Can be used to give a user in a different account temporary access to your resources
  - – Includes a trust policy that specifies which principals are allowed to use the role

Terminology

- AWS Organisation
  - Groups together accounts into organisation units
  - Management account can create policies that apply to member accounts.
  - Consolidated billing

Using Roles

Create role

Role access

Switch roles

- A role is created with a *trust* policy and another *policy* that specifies what actions the role is allowed to perform.
- The role is shared with the users that are permitted to use the role.
- Then those users can assume the role.

Trust policies

- When you create a role you must include a trust policy.
- This specifies who is allowed to assume the role.

```
{
  "Version": "2012-10-17",
  "Statement": {
    "Effect": "Allow",
  "Action": "sts:AssumeRole"
    "Principal":
{"Service": "ec2.amazonaws.com"},

  }
}
```

User permissions

- To assume a role you need permission first:

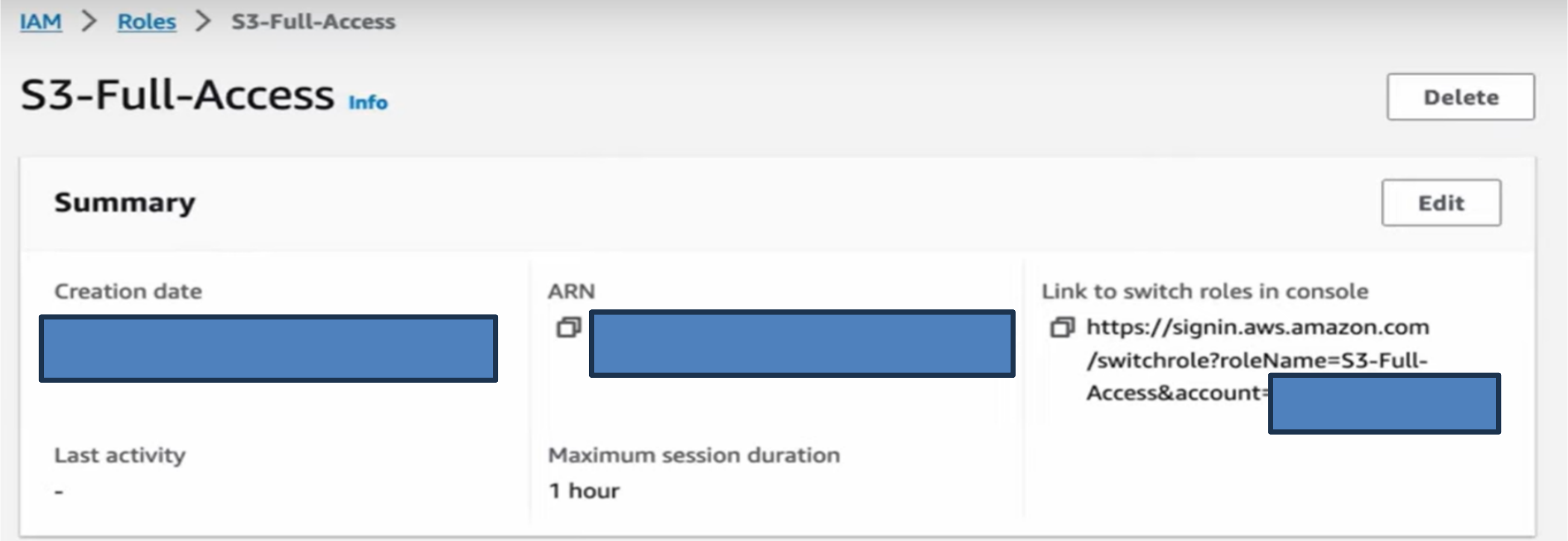
```
{
  "Version": "2025-7-10",
  "Statement": {
    "Effect": "Allow",
    "Action": "sts:AssumeRole",
    "Resource": "arn:aws:iam::account-id:role/*"
  }
}
```

- You also require permission to pass a role to another user or service:

```
{
    "Version": "2025-7-10 7",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "iam:GetRole",
            "iam:PassRole"
        ],
        "Resource": "arn:aws:iam::account-id:role/*"
    }]
}
```

Assuming a role

- The user can then switch to the role in the browser or using CLI.
- This gives them the permissions of the role.



IAM > Roles > S3-Full-Access

S3-Full-Access Info

Summary

Creation date
[redacted]

ARN
[redacted]

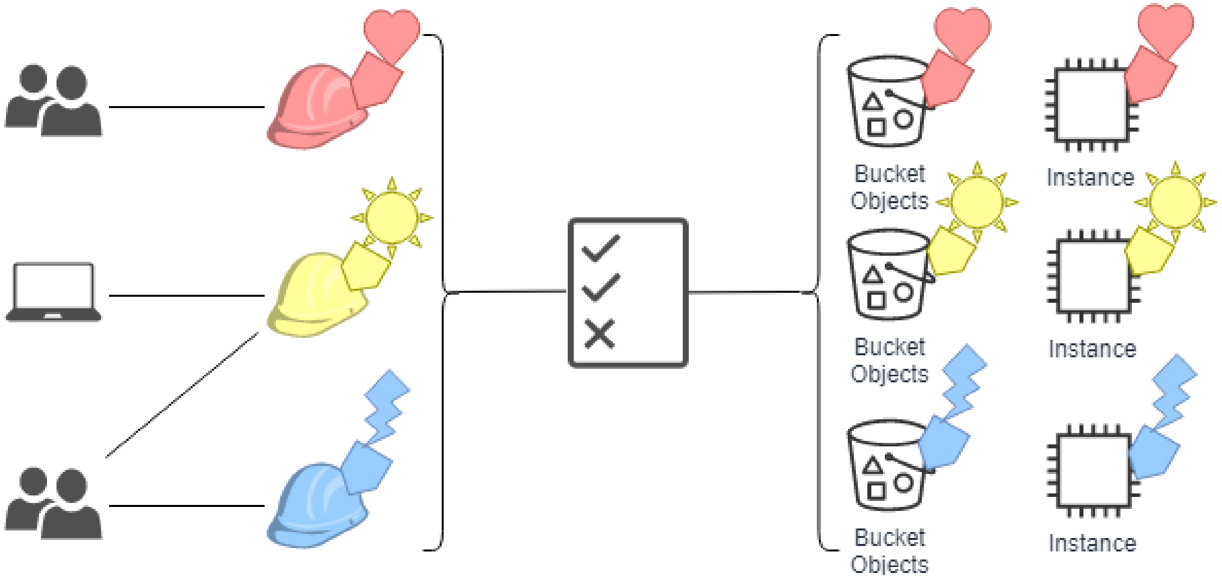
Link to switch roles in console
https://signin.aws.amazon.com/switchrole?roleName=S3-Full-Access&account=[redacted]

Last activity
-

Maximum session duration
1 hour

Attribute Based Policies

- These are policies that give permission based on attributes.
- You can apply tags to entities (users, roles) and resources (buckets, instances, etc).



Tags

- Tags have a tag key and tag value.

  team = frontend

- You can apply multiple tags and each tag can be used in multiple places.
- You cannot apply two tags with the same key on the single resource at once.

  ~~team = frontend~~

  ~~team = security~~

- This would overwrite the value of the team tag to the new value.

Tag Policies

- In the conditions of the policy you can get the tags on the resource using:

```
aws:ResourceTag/key-name:"tag-value"
```

- For example:

```
"Condition": {
       "StringEquals": {"aws:ResourceTag/Owner": "${aws:username}"}
           }

"Condition": {
           "StringEquals": {
               "aws:PrincipalTag/team": [
                   “frontend",
                   “security"
               ]
           }
```

Questions?

- https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_policies_condition-keys.html
- https://docs.aws.amazon.com/IAM/latest/UserGuide/reference_aws-services-that-work-with-iam.html

- Q and A