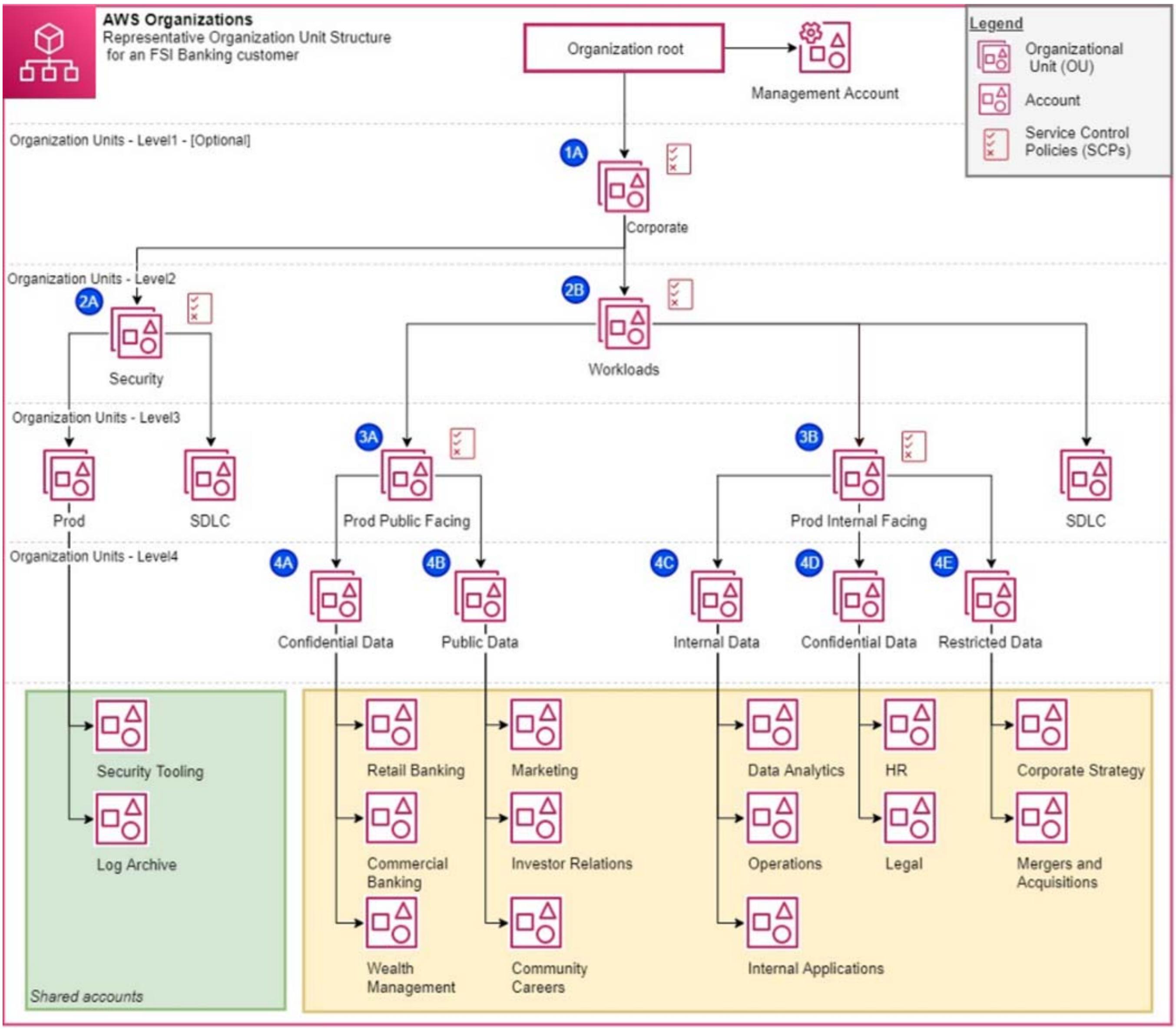
Multi-Account Best Practices

Implementing multi-account best practices is an ongoing effort. The following links and references provide resources so that you can learn more about the subject. Remember that the goal of this course is not to dive deep into each approach. Instead, the goal of this course is to maintain knowledge at a Solutions Architect - Associate level by highlighting the services that are used for the most common approaches.

Multi-account environments

The *AWS for Industries Blog* includes a blog post called Best Practices for AWS Organizations Service Control Policies in a Multi-Account Environment. This blog post talks about core concepts of AWS Organizations and service control policies (SCPs). It also provides suggestions about using different organizational unit (OU) structures for different use cases (such as a corporate OU, a production public-facing OU, a production internal-facing OU, and a security OU). At the end, you might have a complex structure that looks like the following diagram:



Organizing Your AWS Environments Using Multiple Accounts is an AWS Whitepaper that was published on July 26, 2022. It talks about the AWS Well-Architected Framework and much of the content that you learned about in this course. It’s a good read!

The *AWS Organizations* website also has another reference for further reading, called Establishing your best practice AWS environment.

Tag policies and SCPs

With Organizations, you can set tag policies, in addition to defining SCPs. You can use tag policies to maintain standardized tags for AWS resources that are used with Organizations accounts. For example, the following example tag policy defines a tag with a key of *Environment* and a value of *Production*, and this tag is enforced for Amazon Elastic Compute Cloud (Amazon EC2) instances.

```
{
  "tags": {
    "Environment": {
      "tag_key": {
        "@@assign": "Environment"
      },
      "tag_value": {
        "@@assign": [
          "Production"
        ]
      },
      "enforced_for": {
        "@@assign": [
          "ec2:instance"
        ]
      }
    }
  }
}
```

This tag policy prevents users from changing this tag on *existing* Amazon EC2 instances. *However, it doesn't prevent a user from launching new instances with non-compliant tags, or no tags.* This tag policy might be adequate if you use infrastructure as code (IaC) to provision environments. With IaC, environments will be created from a CloudFormation template, and you can thus embed the tags in the template.

However, if you want to prevent the creation of new AWS resources that aren't tagged, you need to *use SCPs.* You could use the following example SCP to make sure that the AWS resources are created only if a certain tag is present. This SCP requires specific tags on specified created resources, and it uses explicit deny statements.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyCreateSecretWithNoProjectTag",
      "Effect": "Deny",
      "Action": "secretsmanager:CreateSecret",
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:RequestTag/Project": "true"
        }
      }
    },
    {
      "Sid": "DenyRunInstanceWithNoProjectTag",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:volume/*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/Project": "true"
        }
      }
    },
    {
      "Sid": "DenyCreateSecretWithNoCostCenterTag",
      "Effect": "Deny",
      "Action": "secretsmanager:CreateSecret",
      "Resource": "*",
      "Condition": {
        "Null": {
          "aws:RequestTag/CostCenter": "true"
        }
      }
    },
    {
      "Sid": "DenyRunInstanceWithNoCostCenterTag",
      "Effect": "Deny",
      "Action": "ec2:RunInstances",
      "Resource": [
        "arn:aws:ec2:*:*:instance/*",
        "arn:aws:ec2:*:*:volume/*"
      ],
      "Condition": {
        "Null": {
          "aws:RequestTag/CostCenter": "true"
        }
      }
    }
  ]
}
```

Do you see how SCPs and tag policies can—and should—be used together? For more information, see Require a tag on specified created resources in the *AWS Organizations User Guide*.