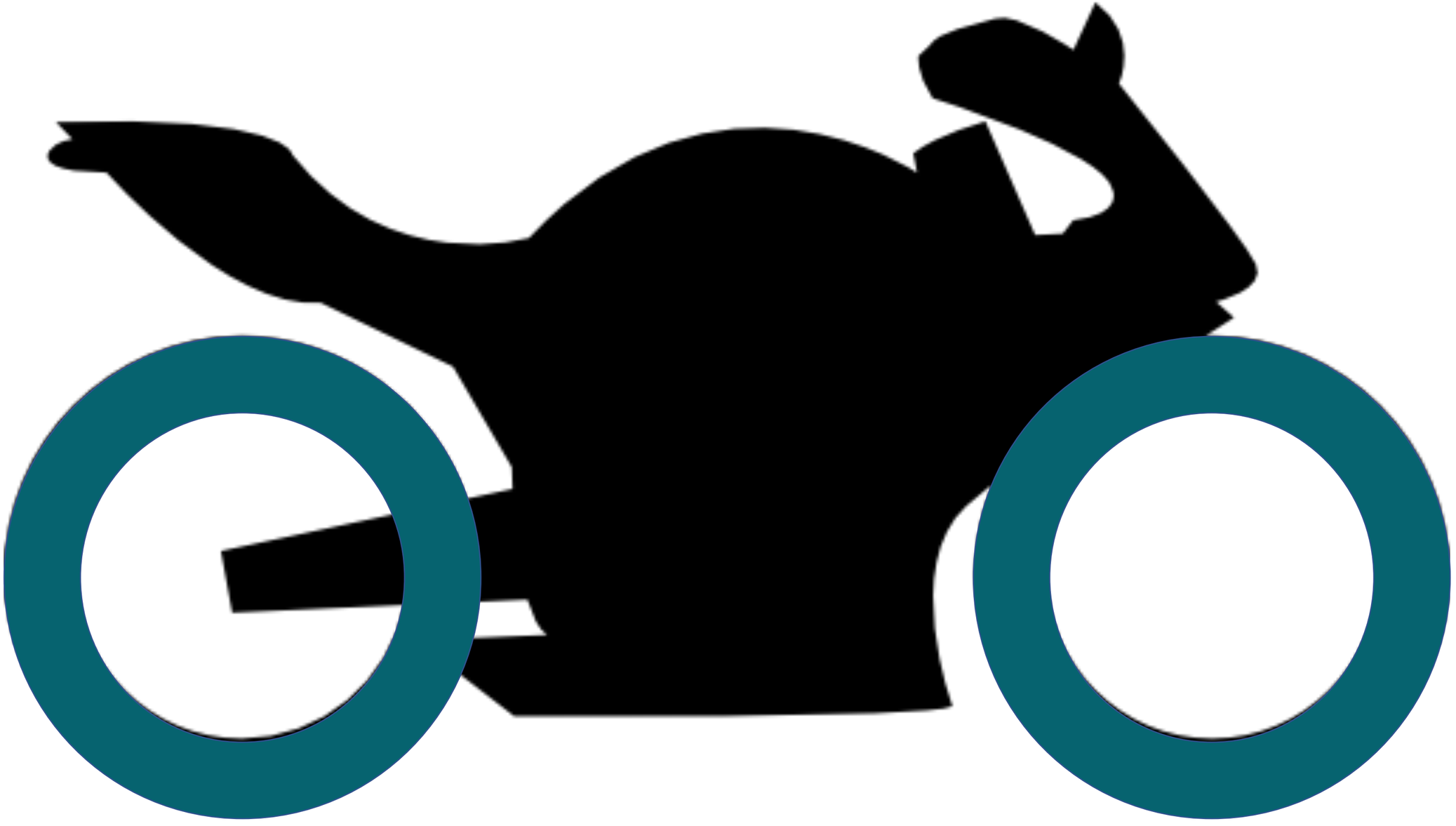
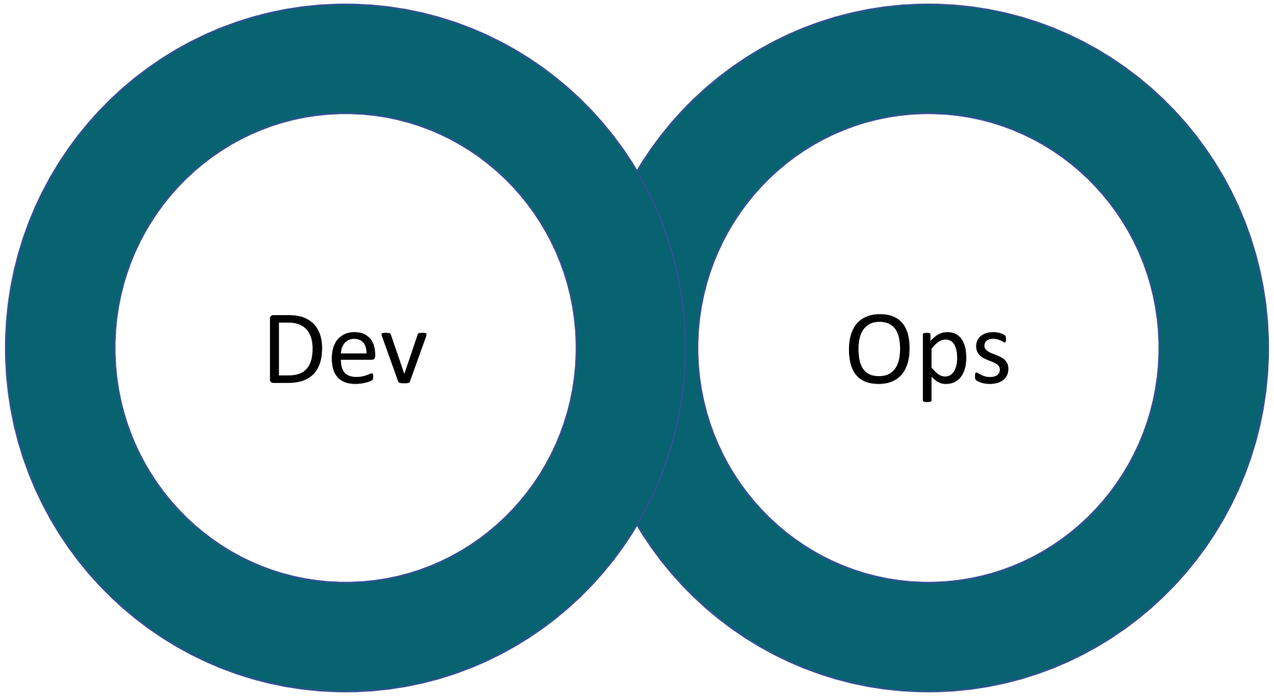
# Governance e Compliance: Approccio *shift-left* e *shift-right* nel mondo cloud

*La maggior parte della potenza frenante di una moto viene dal freno anteriore, esso aiuta l’entrata in curva. Anche se il freno posteriore fornisce una minore potenza frenante rispetto all'anteriore, ha un ruolo importante nel controllo della moto all’interno della curva*

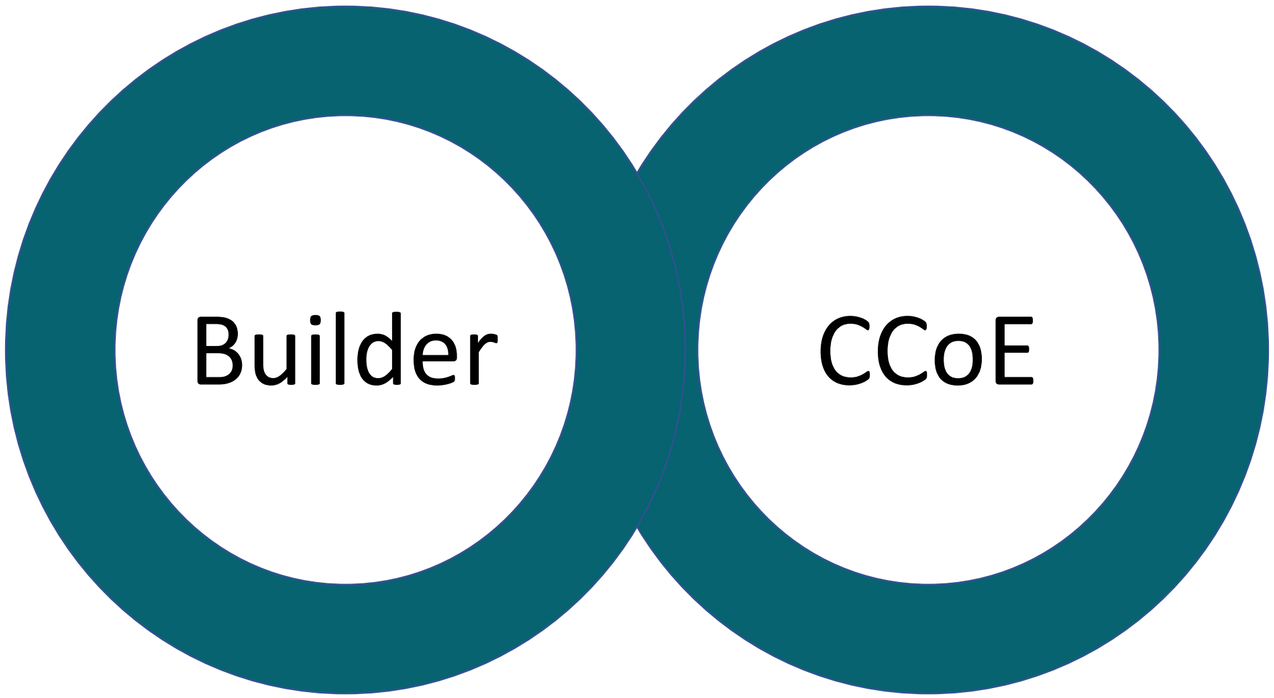
*Keith Code – "A Twist of the Wrist"*



Shift left and shift right are core testing concepts of the agile DevOps methodology. As part of the continuous cycle of progressive delivery, DevOps teams are also adopting Shift-Left (ensure software meets design) and Shift-Right (ensure performace, resilience, availability) principles to ensure software quality in these dynamic environments.



# Governance e Compliance
# Approccio *shift-left* e *shift-right* nel mondo cloud



# Paolo Latella

CEO/CTO ReCube

AWS Hero / AWS Authorized Instructor

BIKERS

https://www.linkedin.com/in/paololatella/

@LatellaPaolo

# Define the rules of the game – Choose your governance model

## Centralized Model

- The MSO/MSP work as service broker
- More work streams with a DevOps approach
- Work streams with compliance requirements related to both projects and organization
- Builders work on project’s requirement with limited autonomy
- Safety before Agility

## Decentralized Model

- The MSO/MSP work as facilitator
- More work streams with a DevOps approach
- Work streams with compliance requirements more related to the organization.
- The project's compliance requirements are delegated to builders
- Balance Agility and Safety

# Define the rules of the game – Choose your governance model



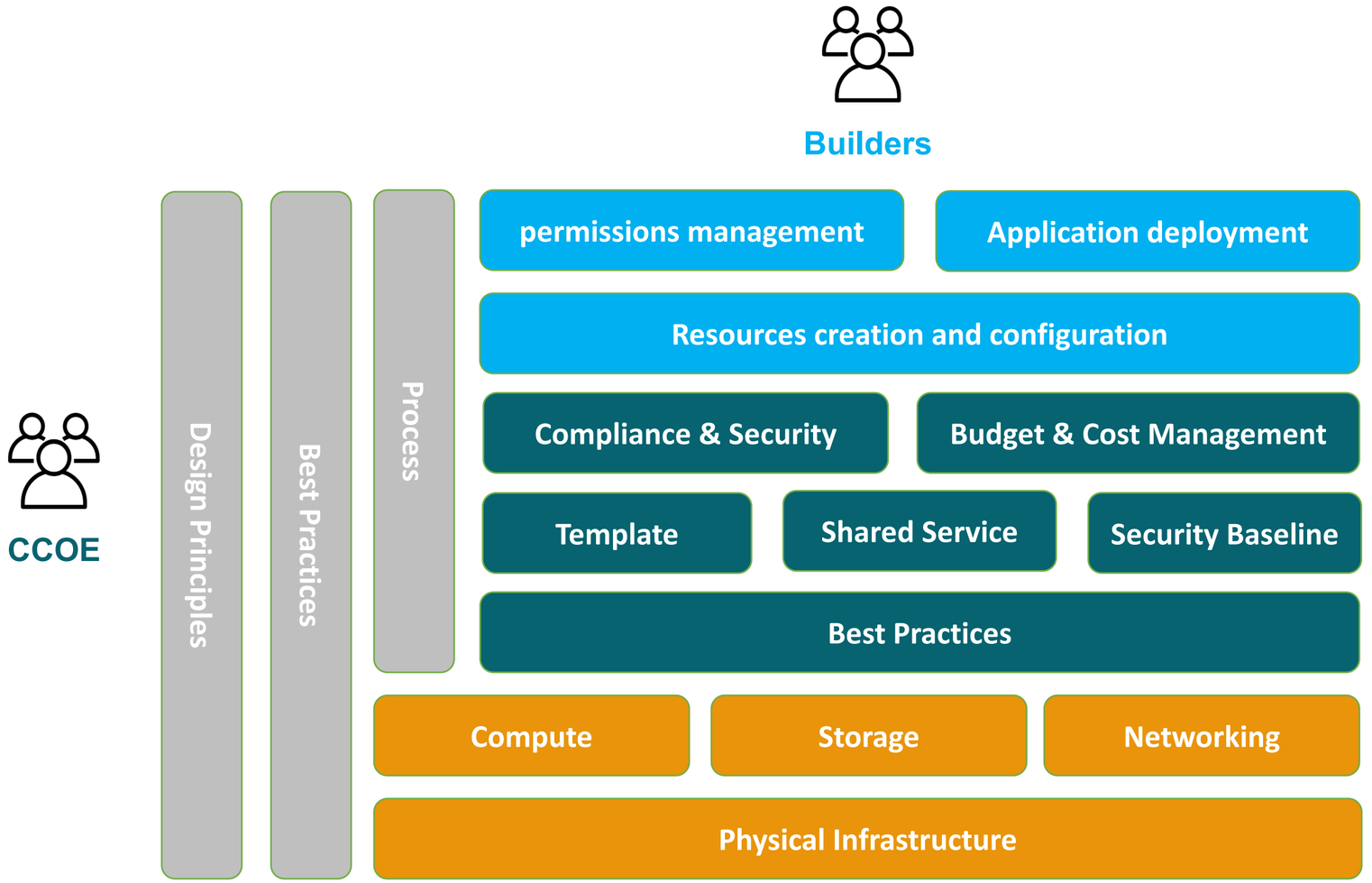
## Centralized Model

- The MSO/MSP work as service broker
- More work streams with a DevOps approach
- Work streams with compliance requirements related to both projects and organization
- Builders work on project’s requirement with limited autonomy
- Safety before Agility

## Decentralized Model

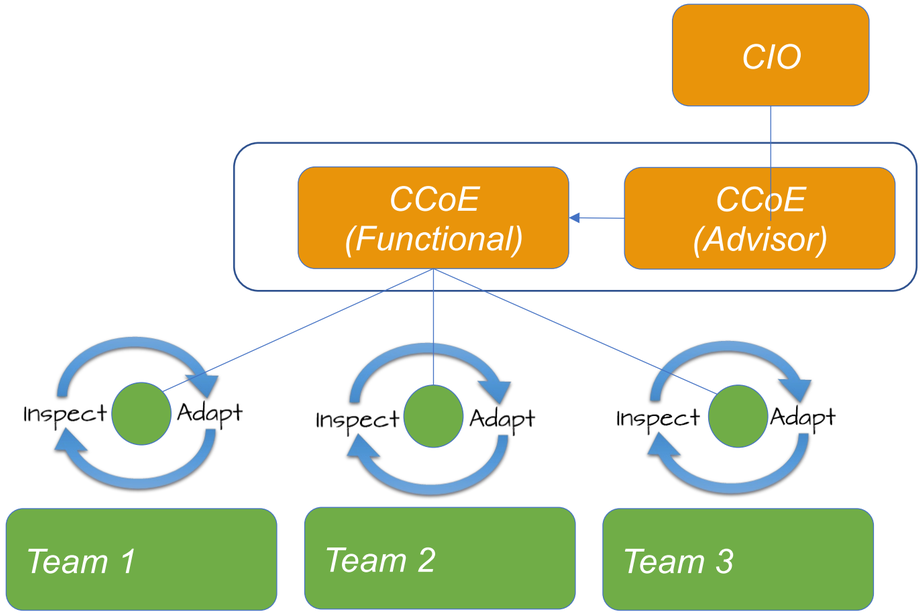
- The MSO/MSP work as facilitator
- More work streams with a DevOps approach
- Work streams with compliance requirements more related to the organization.
- The project's compliance requirements are delegated to builders
- Balance Agility and Safety

# Define the rules of the game - Decentralized Governance



# Define the rules of the game – Start a CCoE (Cloud Center of Excellence)

*A CCoE develop and implement the solutions under the processes and principles that Organization has defined*



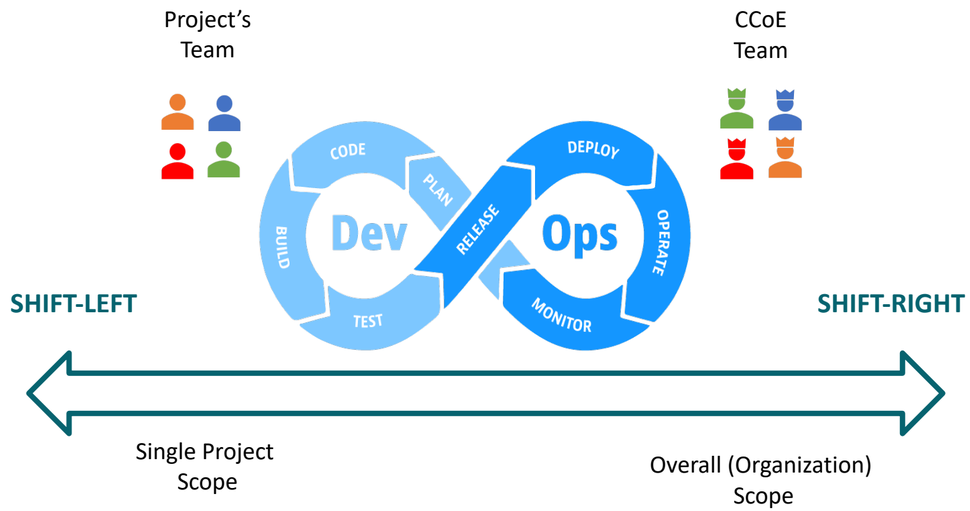

## Principles

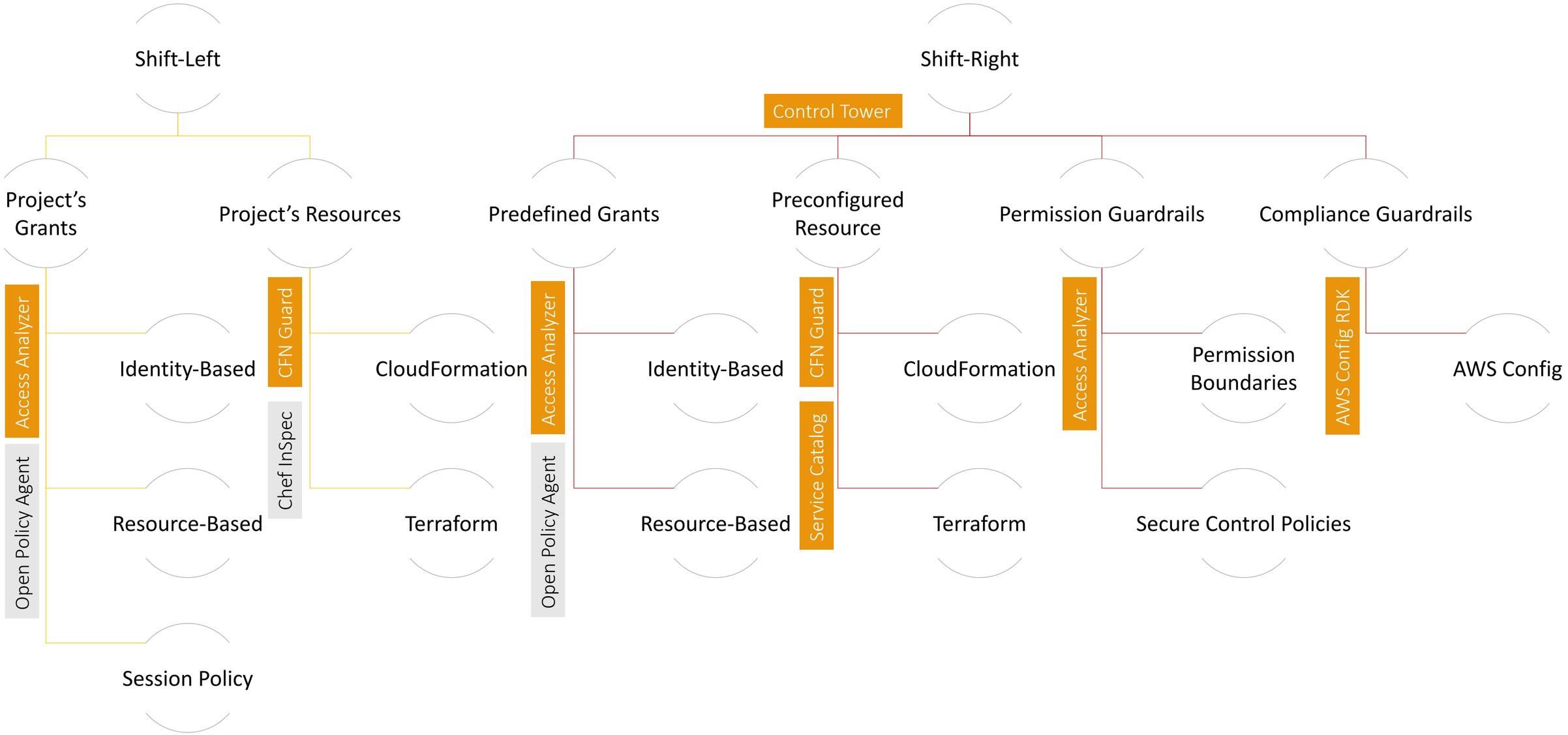
- Drive Cloud Culture
- Engage and Evangelize
- Scale and Re-Organize
- Build Reference Architectures
- Improve agility and security

*Let your builders to experiment and innovate quickly and safely. Adopt a decentralized governance model and encourage both Shif-Left and Shift-Right approach*

# Governance - Encourage both shift-left and shift-right approach

- Shift-Left
  - Detect compliance issues very early in the process
  - Our tests are more related to a specific workload.
  - Tests planned and executed by the project's team
  - More open-source and CSP agnostic tools
- Shift-Right
  - Require more effort to align with requirements
  - Requirements defined at the organization level.
  - The tests are planned and executed by a CCoE
  - Tools (services) provided directly by CSP




Shift-Left
Shift-Right
Control Tower
Project's Grants
Project's Resources
Predefined Grants
Preconfigured Resource
Permission Guardrails
Compliance Guardrails
Access Analyzer
Open Policy Agent
Identity-Based
Resource-Based
Session Policy
CFN Guard
Chef InSpec
CloudFormation
Terraform
Access Analyzer
Open Policy Agent
Identity-Based
Resource-Based
CFN Guard
Service Catalog
CloudFormation
Terraform
Access Analyzer
Permission Boundaries
Secure Control Policies
AWS Config RDK
AWS Config

# Shift-Left / Shift-Right - IAM Access Analyzer

IAM Access Analyzer continuously monitors and analyzes resource policy to help you understand the potential security implications. It guides you toward least privilege permissions.

- Identify resources in your organization and accounts that are shared with an external entity.
- Validates IAM policies against policy grammar and best practices.
- Generate IAM policies based on access activity in your AWS CloudTrail logs.

# IAM Access Analyzer - Validate

```
aws accessanalyzer validate-policy --policy-type RESOURCE_POLICY \
--profile CCMD-EUCOM-ThirdFleet-PaoloLatella \
--policy-document file://Workspaces/CCMDLabs/ValidatePolicy.json --region us-east-1
```

```
{
    "Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "NotPrincipal": {
                 "AWS": [
                "arn:aws:sts::444455556666:assumed-role/cross-account-read-only-role/",
                "arn:aws:iam::444455556666:role/cross-account-read-only-role",
                "arn:aws:iam::444455556666:root"
            ]},
        "Action": "s3:*",
        "Resource": [
            "arn:aws:s3:::Bucket_AccountAudit",
            "arn:aws:s3:::Bucket_AccountAudit/*"
        ]
    }]
}
```

RESOURCE_POLICY (Left&Right)
IDENTITY_POLICY (Left&Right)
SERVICE_CONTROL_POLICY (Right)

```
"findings":
        {
             "findingDetails": "Using Allow with NotPrincipal can be overly permissive.,
             "findingType": "SECURITY_WARNING",
             "issueCode": "ALLOW_WITH_NOT_PRINCIPAL",
             "learnMoreLink": "https://docs.aws.amazon.com/IAM/latest/UserGuide/...
         }
```

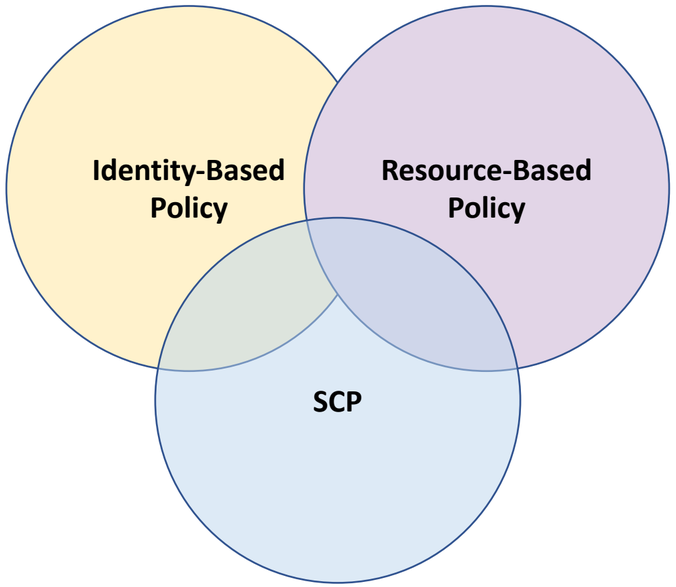
# IAM Access Analyzer - Generate

```
aws accessanalyzer start-policy-generation \
--policy-generation-details principalArn=arn:aws:iam::278014156012:user/paolo.latella \
--cloud-trail-details '{"trails":[…],"startTime":"2022-10-15T10:30:00.000Z","endTime":"2022-10-15T18:30:00.000Z"}'
--profile CCMD-EUCOM-ThirdFleet-PaoloLatella \
--region eu-west-1
```

```
aws accessanalyzer get-generated-policy \
--job-id jobid \
--profile CCMD-EUCOM-ThirdFleet-PaoloLatella \
--region eu-west-1
```

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "s3:Get*",
                "s3:List*"
            ],
            "Resource": [
                "arn:aws:s3:::my-bucket/shared/*"
            ]
        }
    ]
}
```

# Shift-Right - Service Control Policies



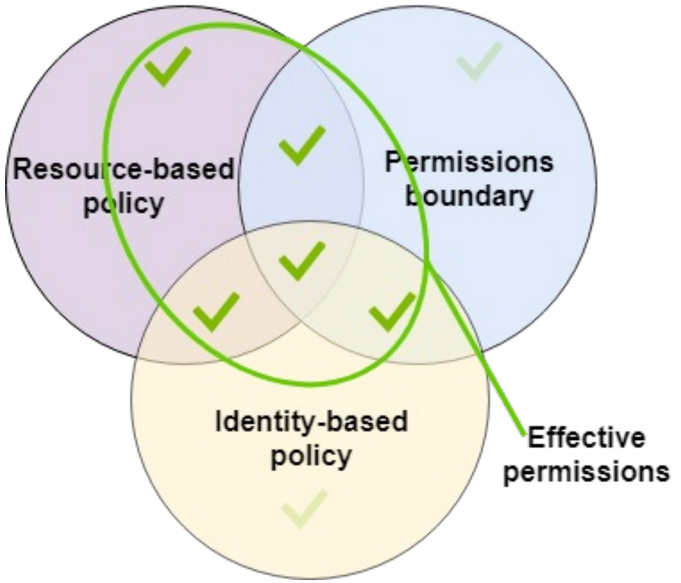

Deny List Strategy - Actions are allowed by default, and you specify what services and actions are prohibited

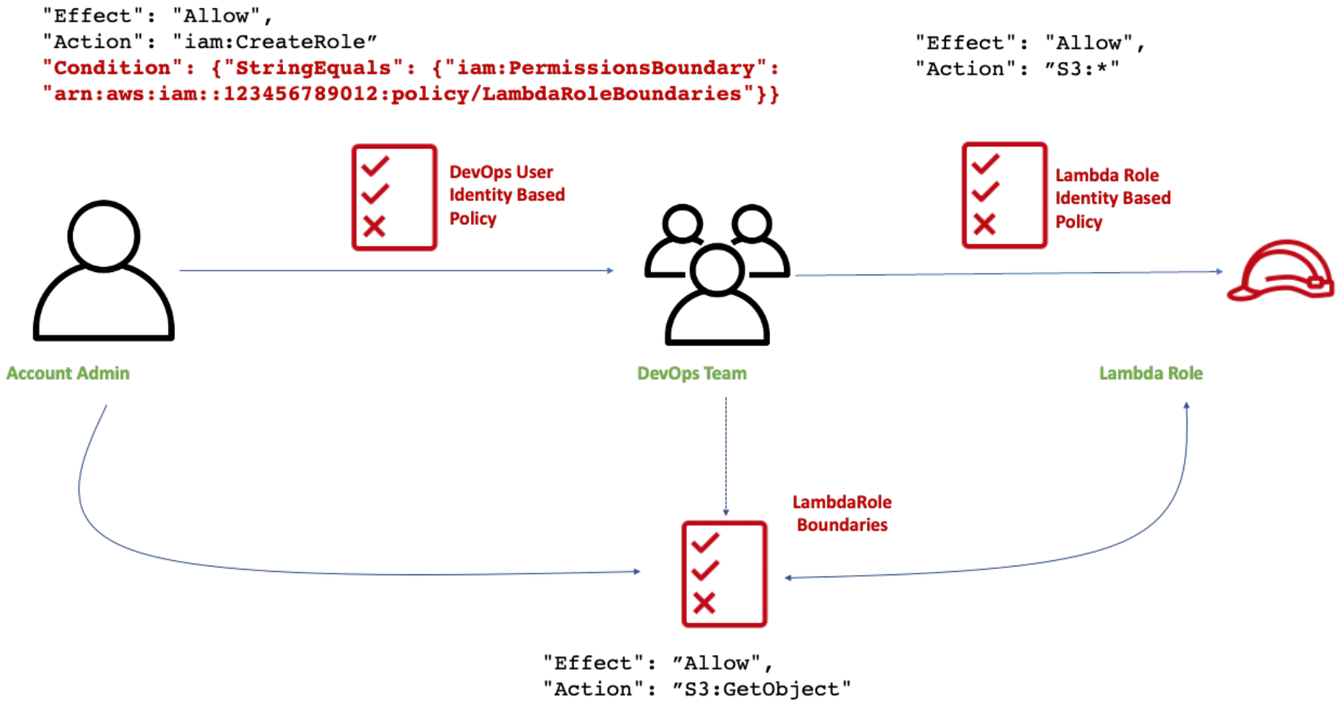
- **Require less maintenance**
- Policy require less space (max 5KB)

Allow List Stratey - Actions are prohibited by default, and you specify what services and actions are allowed

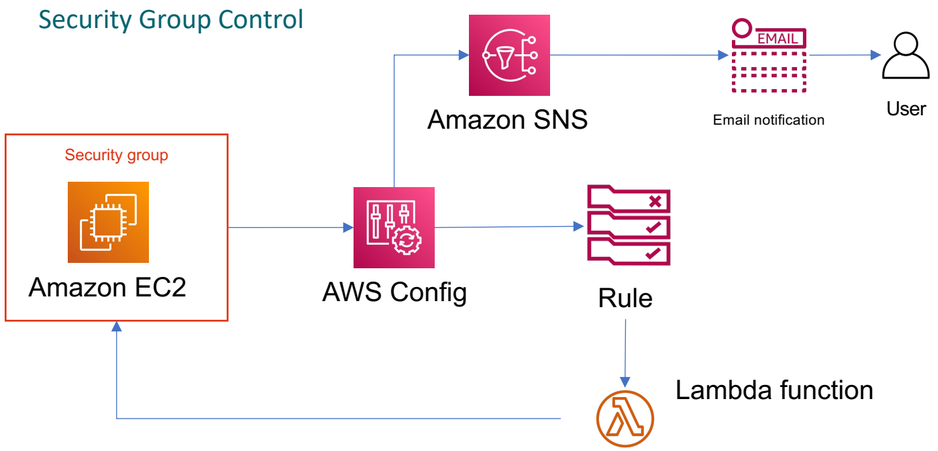
- The implicit deny impact on existing resources
- **Most aligned to principle of least privileges**

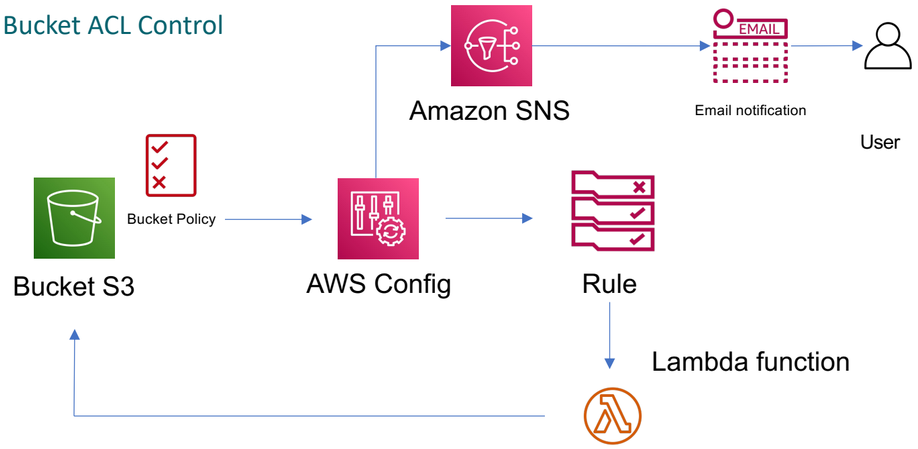
# Shift-Right - Permissions Boundaries

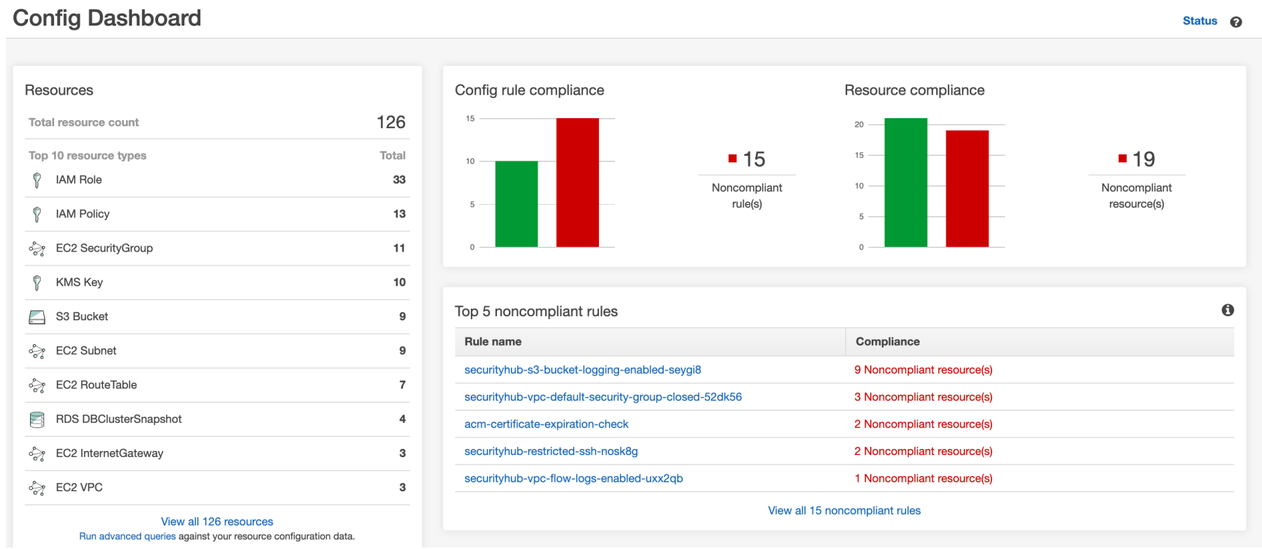


# Shift-Right – Config Rules




Centralized and Aggregated Dashboard

- multi-account
- mullti-regions

# Shift-Right – Config RDK (Rules Development Guide)

```
rdk create SecurityGroupIpCheck --runtime python3.7 --resource-types AWS::EC2::SecurityGroup --input-parameters
'{"allowedIP":"192.168.10.10"}'
Running create!
Local Rule files created.

rdk deploy SecurityGroupIpCheck
Running deploy!
Found Custom Rule.
Zipping SecurityGroupIpCheck
Uploading SecurityGroupIpCheck
Upload complete.
Creating CloudFormation Stack for SecurityGroupIpCheck
Waiting for CloudFormation stack operation to complete....
CloudFormation stack operation complete.
Config deploy complete.

rdk test SecurityGroupIpCheck
Running local test!
Testing SecurityGroupIpCheck
Debug!
test_sts_access_denied ... ok
test_sts_unknown_error ... ok
Ran 3 tests in 0.001s - OK
```

https://github.com/awslabs/aws-config-rdk

# Shift-Left / Shift-Right – CFN Guard and OPA

- AWS CloudFormation Guard is a open source policy-as-code evaluation tool
  - Guard validate JSON/YAML data against Guard DSL policy rules
  - Guard doesn't validate templates for valid syntax or allowed property values (use cfn-lint)
- The Open Policy Agent (OPA) is an open source, general-purpose policy engine that unifies policy enforcement across the stack.
  - A graduated project in the Cloud Native Computing Foundation (CNCF) landscape
  - OPA Policies written in Rego
  - Daemon, Library, CLI Tool

# Shift-Left / Shift-Right – Cloudformation Guard

```
cfn-guard validate --rules rules.guard --data template.json
```

rules.guard

```
#
# For all EC2 Instances specified in the Template verify Tags presence
# For all IAM Roles specified in the Template verify session duration

let roles = Resources.*[Type == "AWS::IAM::Role"]
let ec2 = Resources.*[Type == "AWS::EC2::Instance"]

rule verify_max_session_duration when %roles !empty
{
    %roles.Properties.MaxSessionDuration <= 3600
}

rule verify_tags_presence when %ec2 !empty
{
  %ec2.Properties.Tags !empty
}
```

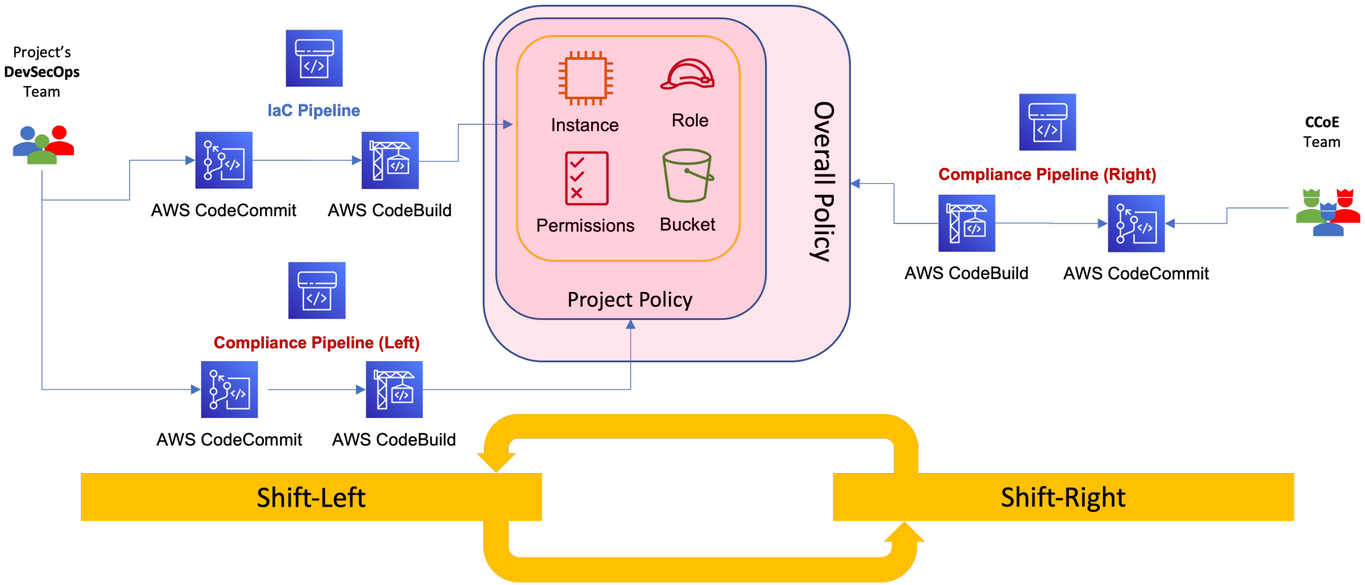
Query/Filter

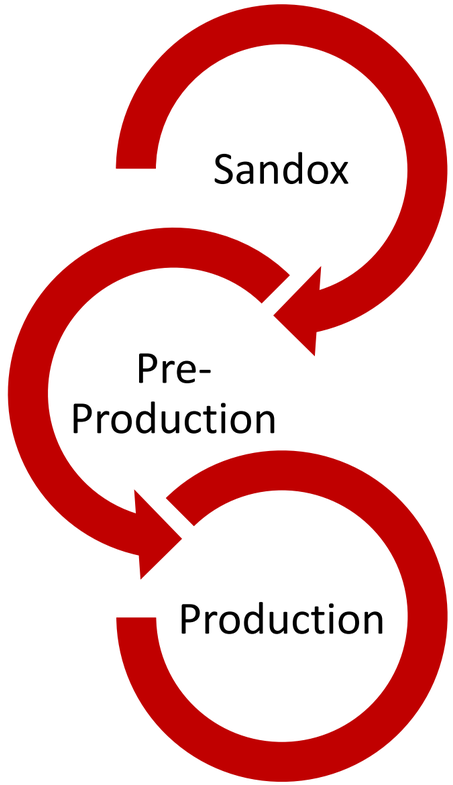
Rule

template.json

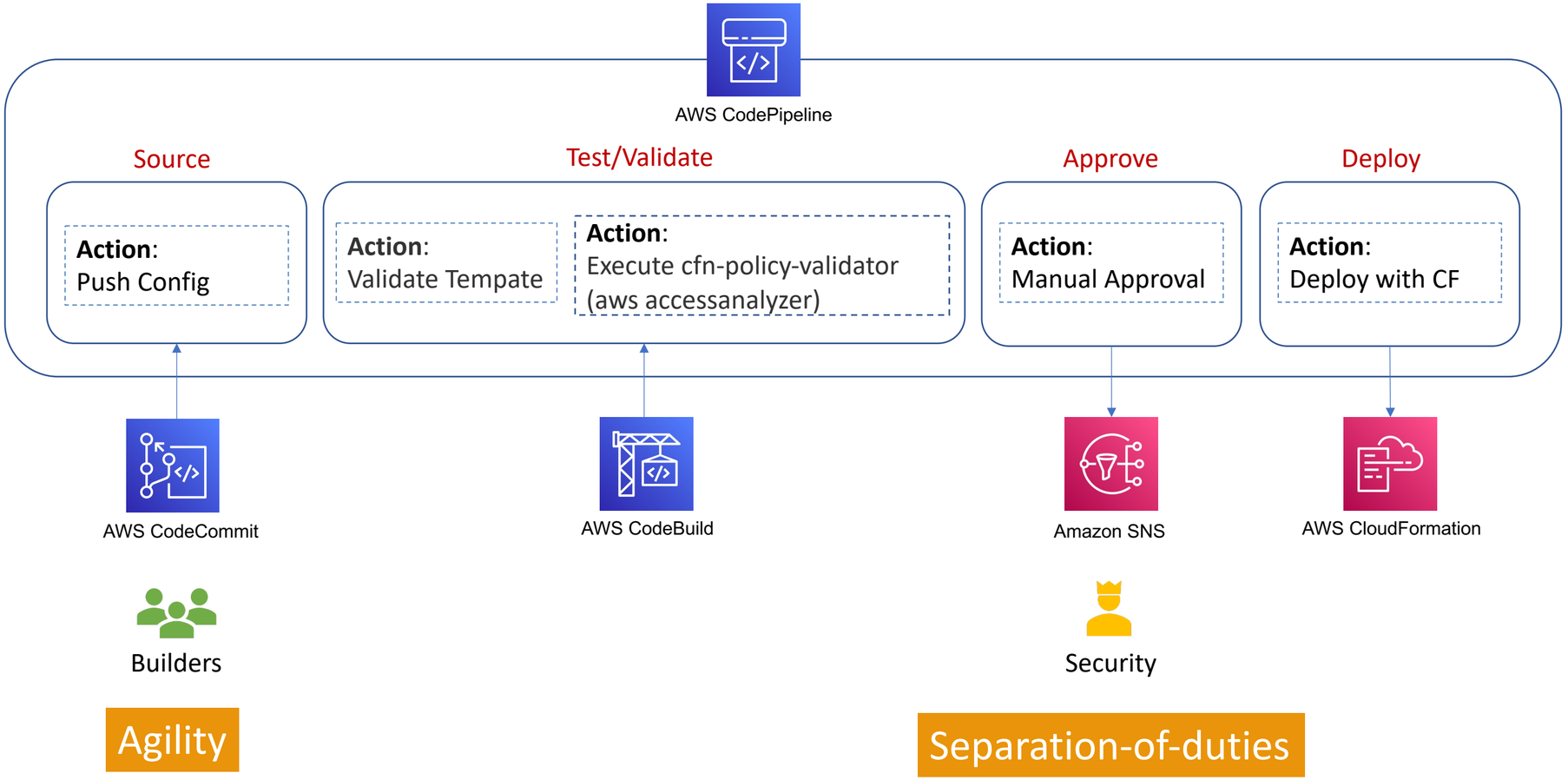
```
{
   "AWSTemplateFormatVersion": "2010-09-09",
   "Resources": {
    "MyTestRole": {
     "Type" : "AWS::IAM::Role",
     "Properties" : {
        "AssumeRolePolicyDocument" : "RomePolicy",
        "Description" : "Test Cloudformation Guard",
        "MaxSessionDuration" : 4600,
        "RoleName" : "TestRole"
      }
     },
    "MyTestInstance": {
     "Type" : "AWS::EC2::Instance",
     "Properties" : {
      "ImageId" : "ami-79fd7eee",
      "tags" : [
       {
        "Key" : "Environment",
        "Value" : "production"
       }
      ]
     ...
```

# Governance – CI/CD on Left and Right side

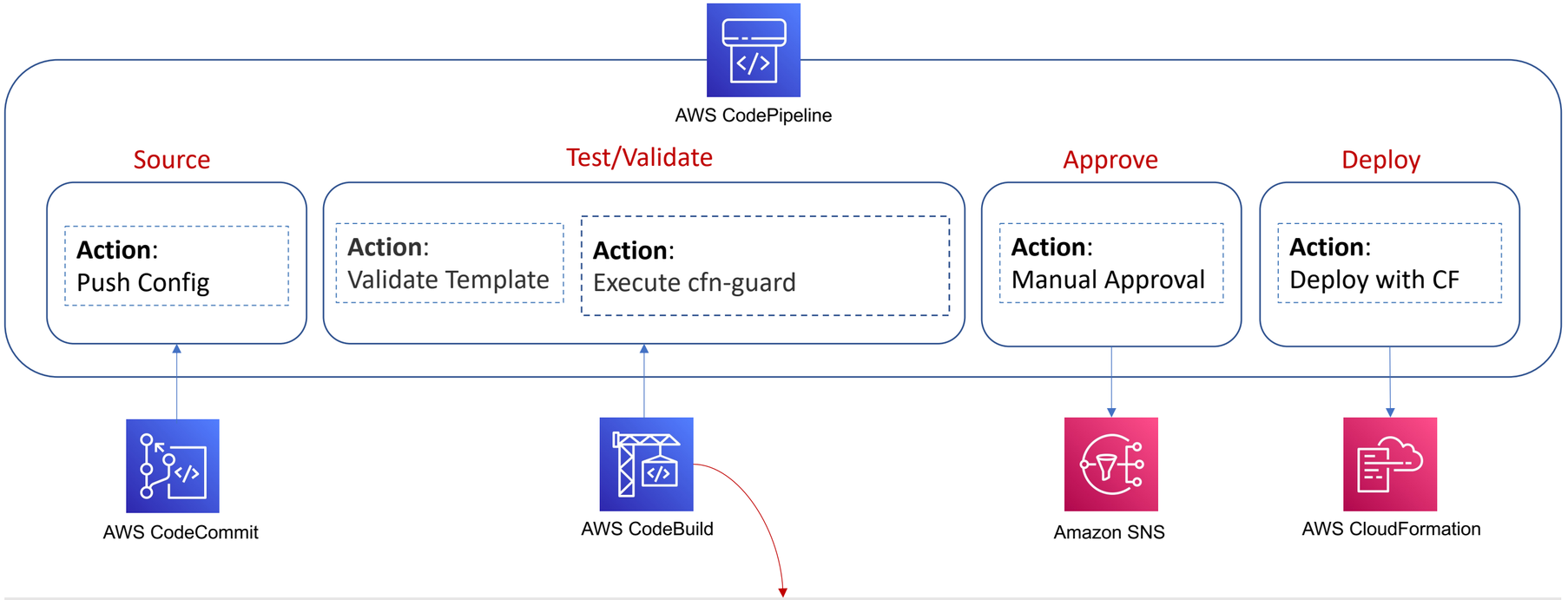


# Policy Analyzer in CI/CD Pipeline



# Cloudformation Guard in CI/CD Pipeline




```
version: 0.2
phases:
    install:
        commands:
        - curl https://raw.githubusercontent.com/aws-cloudformation/cloudformation-guard/main/install-guard.sh | sh
    build:
        commands:
         - cfn-lint template.json
         - cfn-guard validate --rules rules --data template.json
```

# Takeaways

- Adoption of a decentralized model permits to strike the right balance between Agility and Safety
- In a decentralized model you must work on Left side and Right side of AWS organization
- Automate security best practices in creating secure architectures
- Implements controls that are defined and managed as code in version-controlled templates.

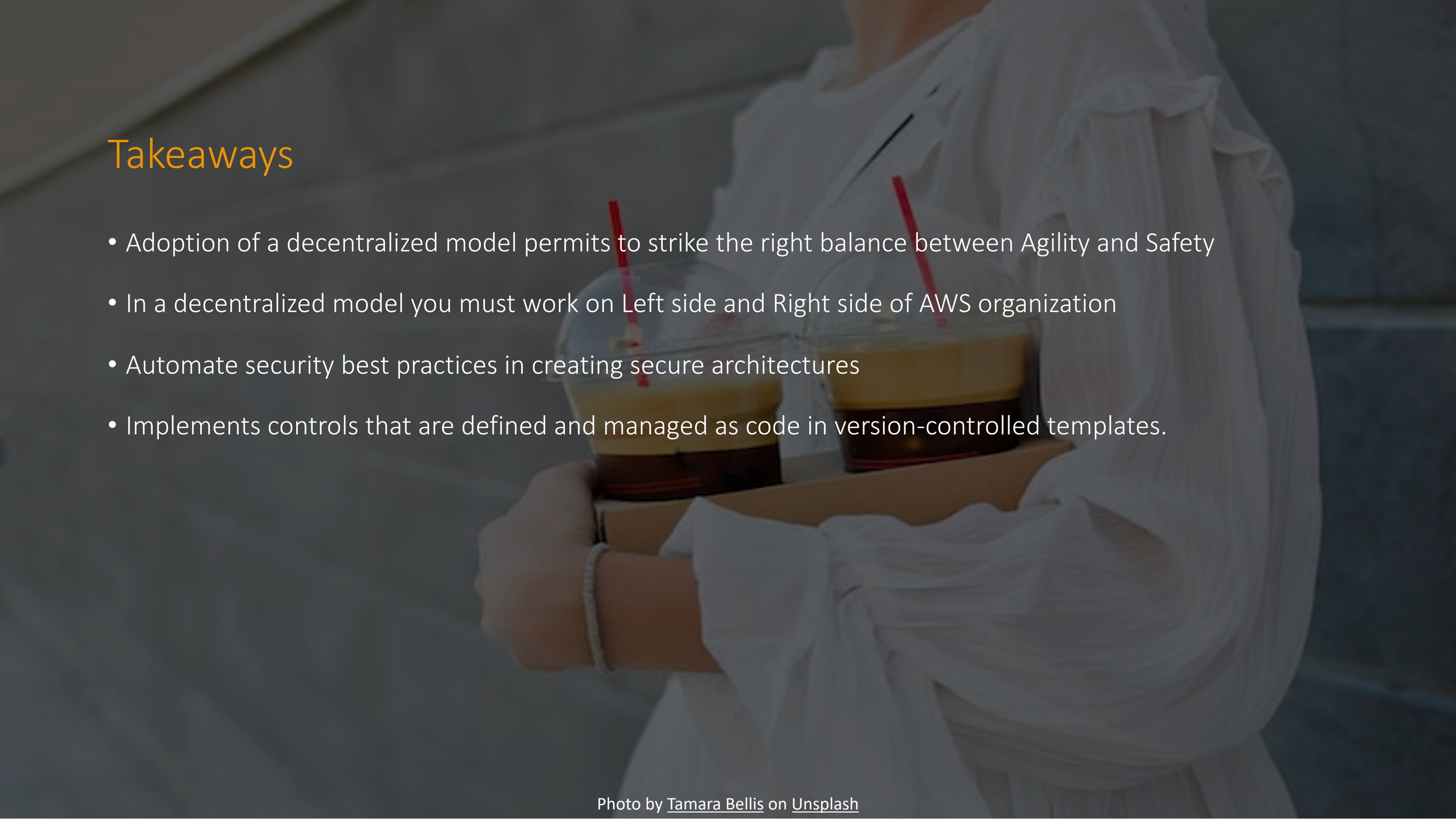


Photo by Tamara Bellis on Unsplash