



ORGANIZING CLOUD-NATIVE JAVA BACKEND

Soham Dasgupta

SPECIAL THANKS TO..



Tijn van den Bergh

Mad genius



Kevin Pors

Tech guru

ABOUT ME

Soham Dasgupta

Cloud Solution Architect @ Microsoft

Father and Football lover

Tech Enthusiast/Programmer

Speaker & Blogger

Twitter : @iamssoham

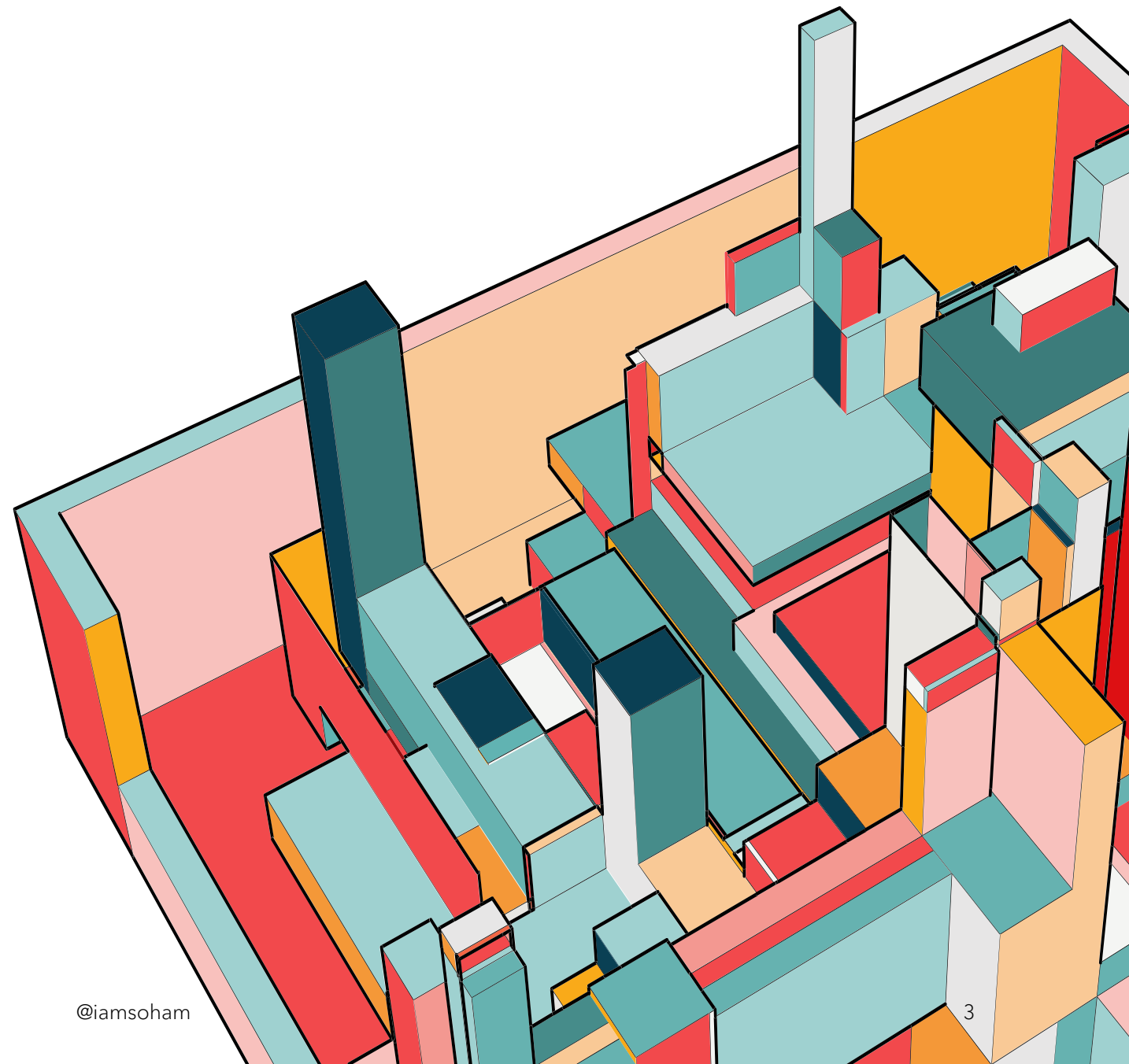
LinkedIn: dasguptasoham

Github: sohamda

Medium: @iam.soham

JSpring'23

@iamssoham



LANDSCAPE

Workflow/Pipeline

QA on PR/MR
OWASP exclusion list

Repo Management

Branching strategy
Versioning
Branch protection rule
Dir. vs Branches
App & config

Maven

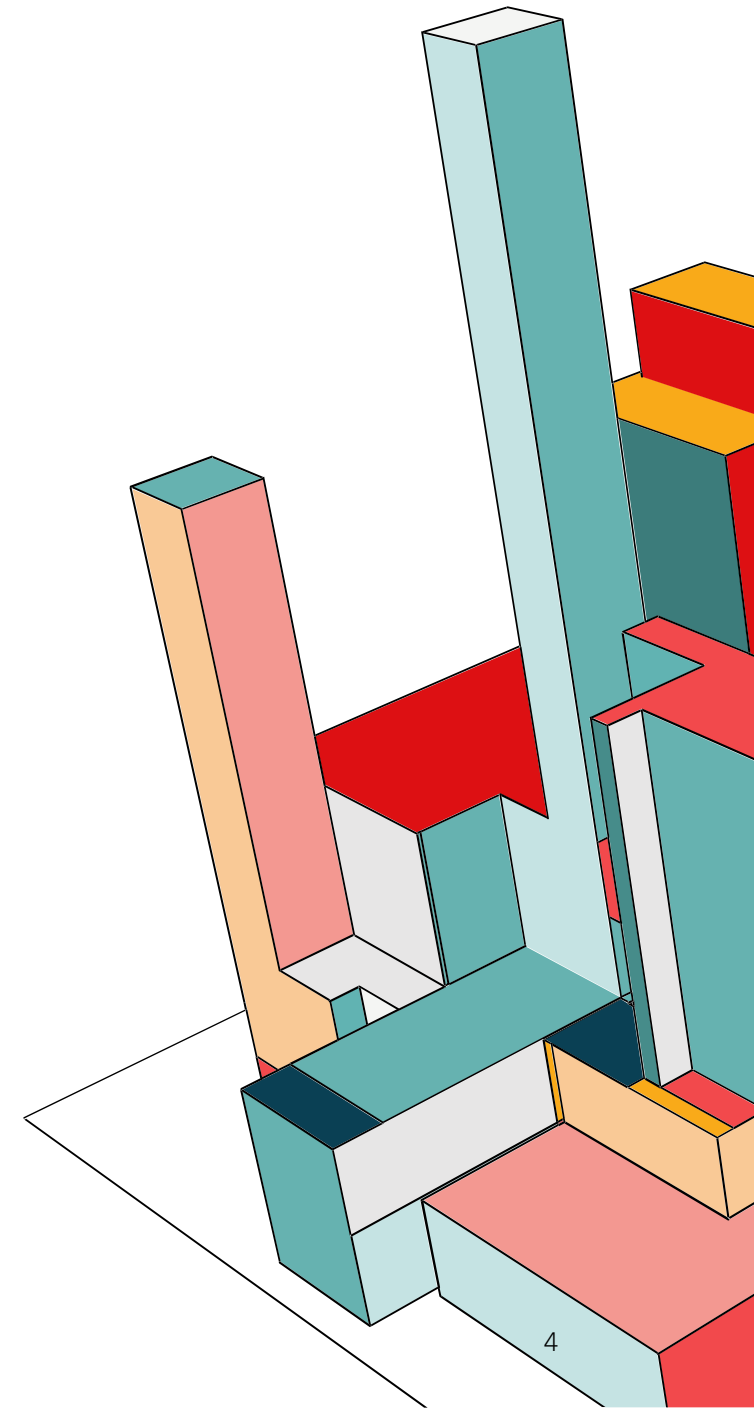
Structuring POMs
Enforcer

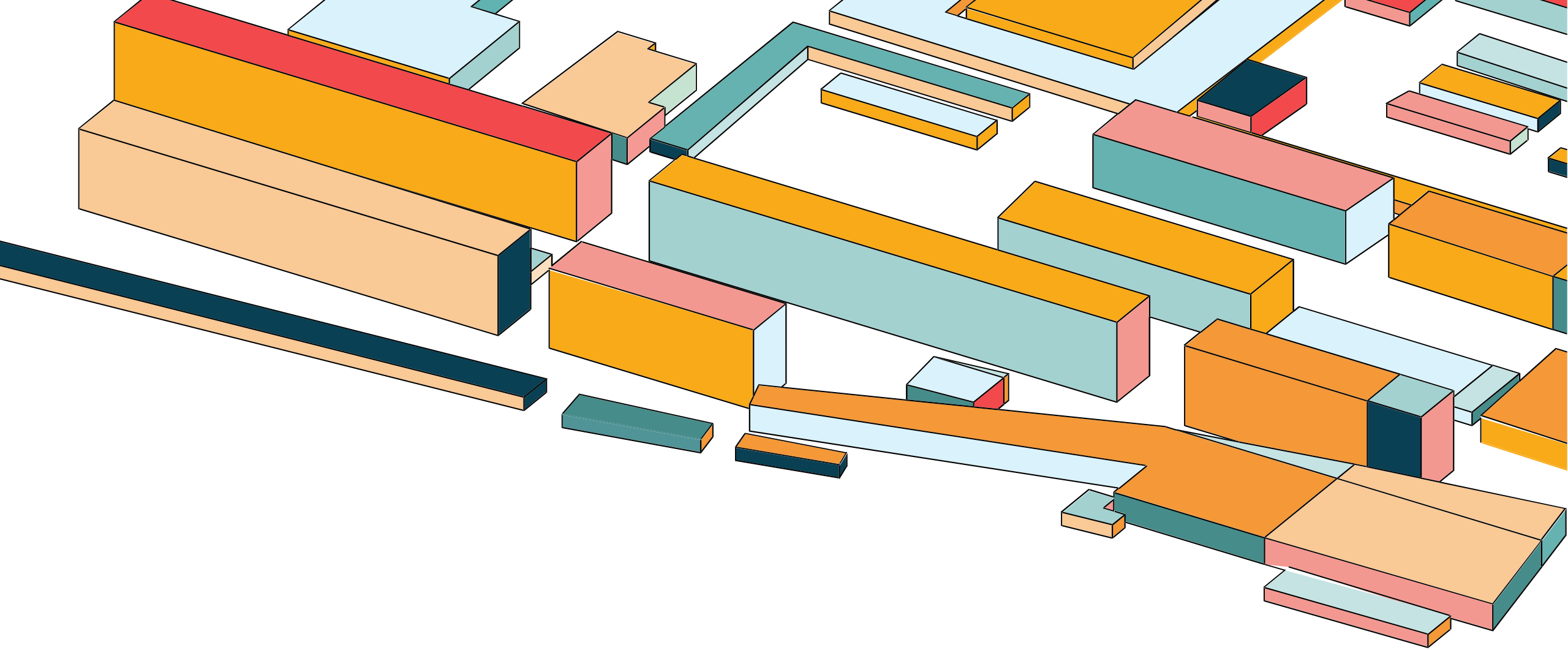
API servers & client

OpenAPI
Generation servers/clients
Sharing

Handy Stuff

Kafka Schema Management
Unit tests as Natural Language
Awaitility
Pi-test





WORKFLOW/PIPELINES



WORKFLOW/PIPELINE

Repeatable steps

Repeatable steps were put on a centralized repo.
Separation of concerns

Same actions/steps for Deployable modules & Libraries

Identify and centralize actions/steps
Nurture the reusability from the beginning

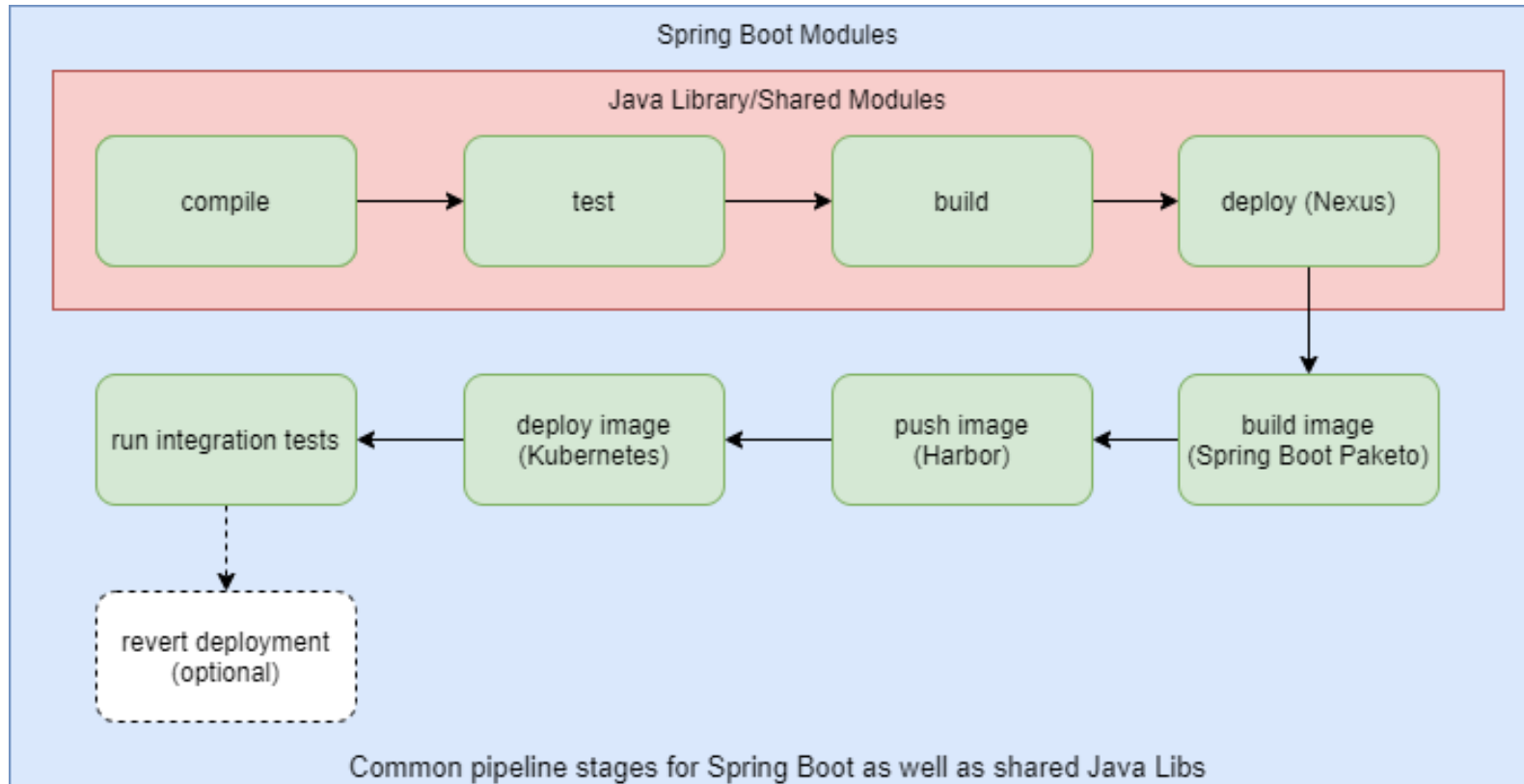
Streamline Branching & Release strategy

Centrally manage versioning, release and deploy

Easiness in Onboarding new repos

Simpler YAML files.
Not everyone has to be an expert on Github actions/Gitlab pipelines

REUSABLE ACTIONS/COMMON PIPELINE



REDUCE WORKFLOW EXECUTION TIME



Separate by
operation/actions/
types

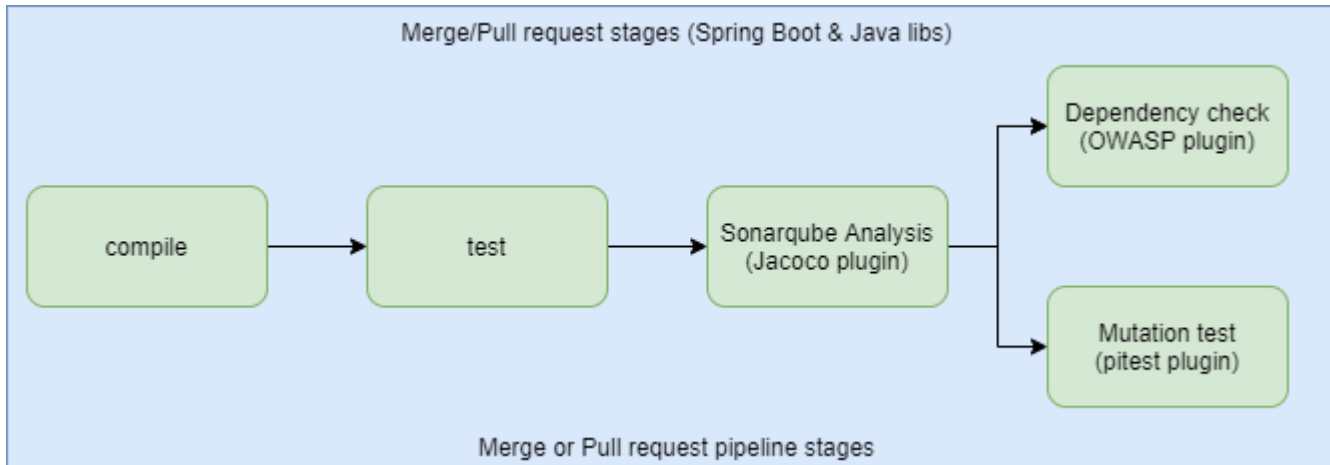


Identify feature
branches and env
specific branches



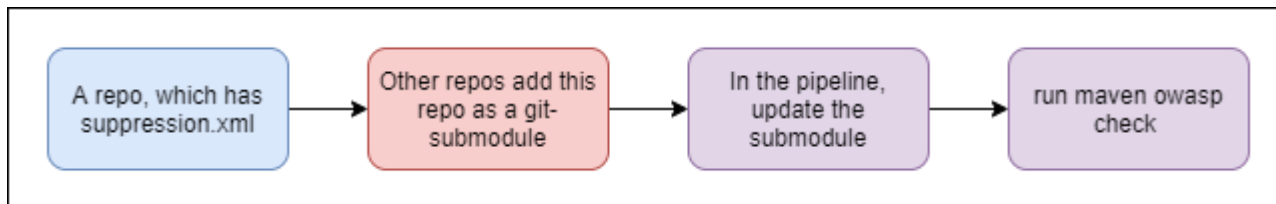
Developer
productivity

QA ON MERGE/PULL REQUEST



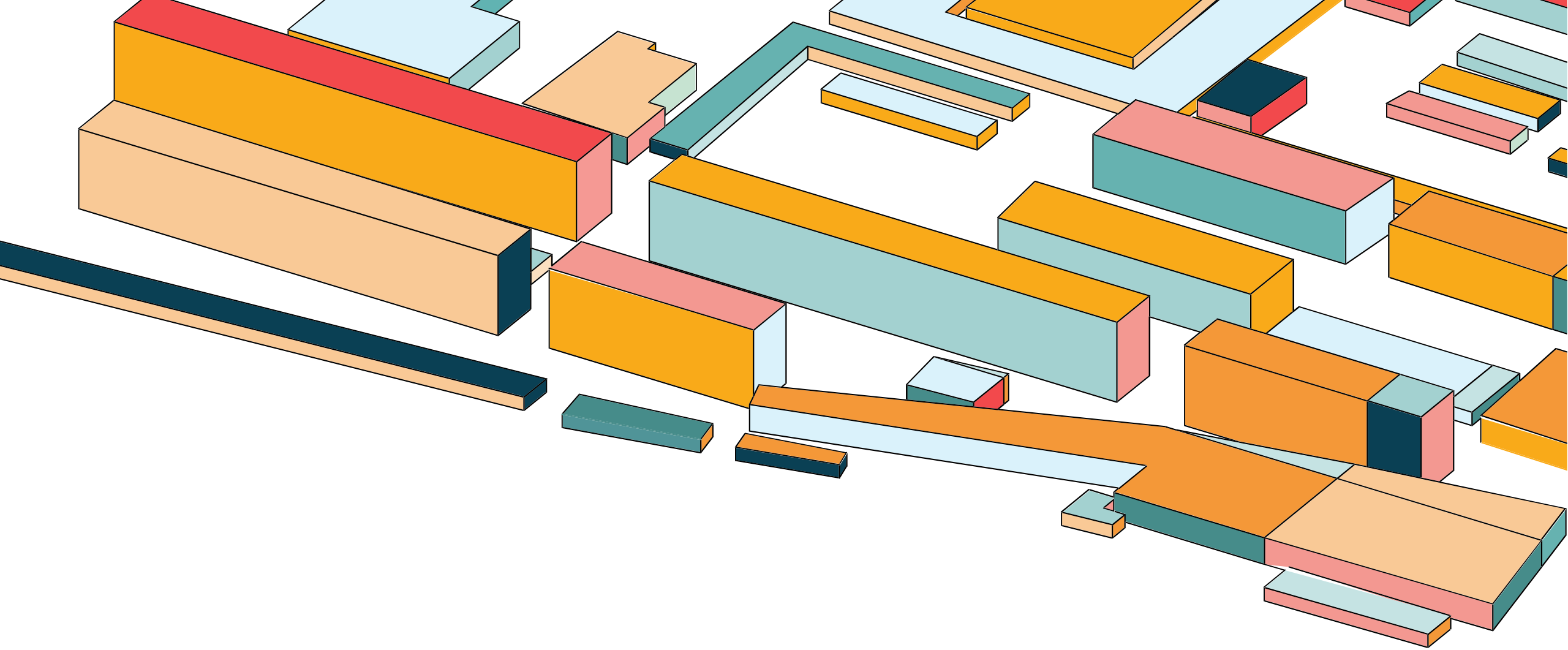
Static Code Analysis
OWASP dependency check
Mutation tests
Container scanning

CENTRALIZED OWASP EXCLUSION LIST



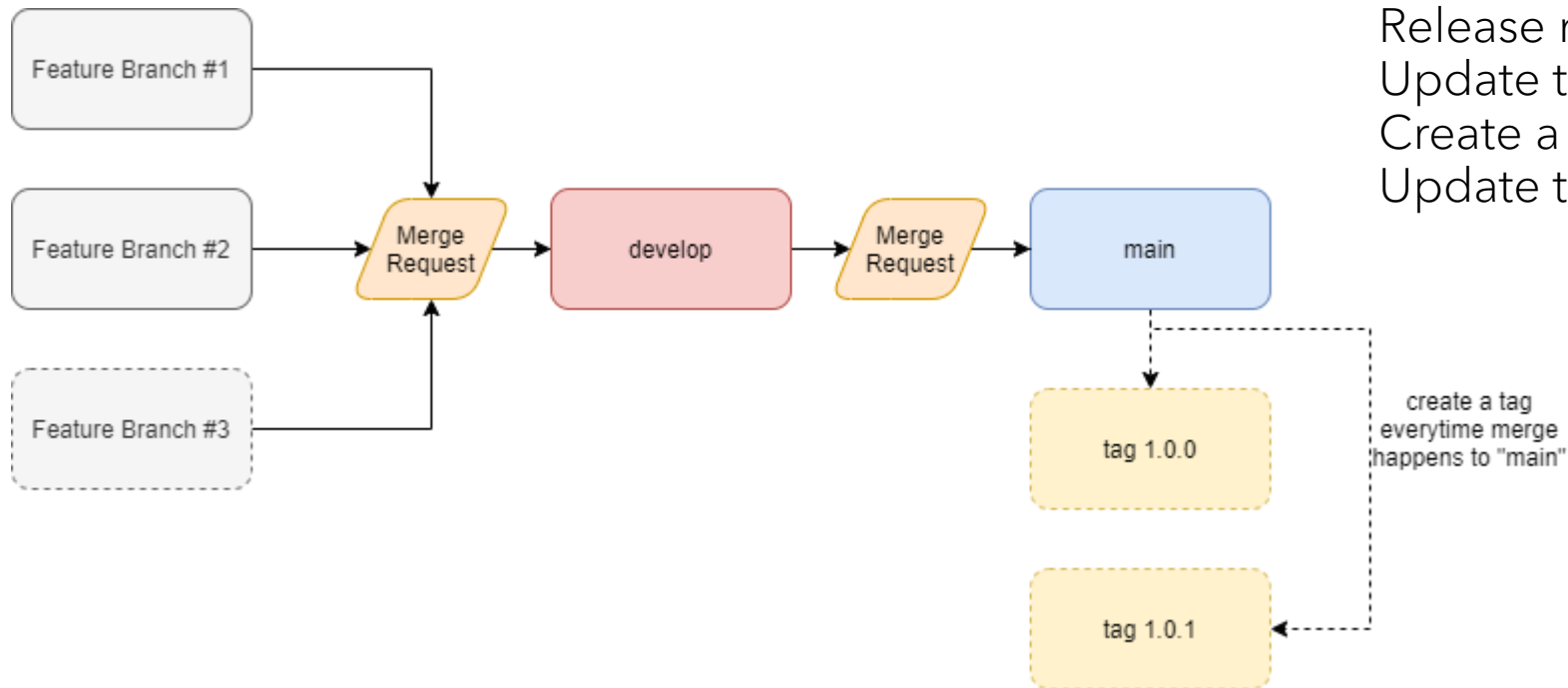
A central list for the all the repos.
Added as a git-submodule to every repo.

```
<?xml version="1.0" encoding="UTF-8"?>
<suppressions xmlns="https://jeremylong.github.io/DependencyCheck/dependency-suppression.1.3.xsd">
  <suppress>
    <notes>Only applicable for spEL injection on @Query and @Aggregation</notes>
    <filePath regex="true">.*\bspring-boot-starter-data-mongodb-2.7.*.jar</filePath>
    <cve>CVE-2022-22980</cve>
  </suppress>
</suppressions>
```



REPO MANAGEMENT

BRANCHING STRATEGY

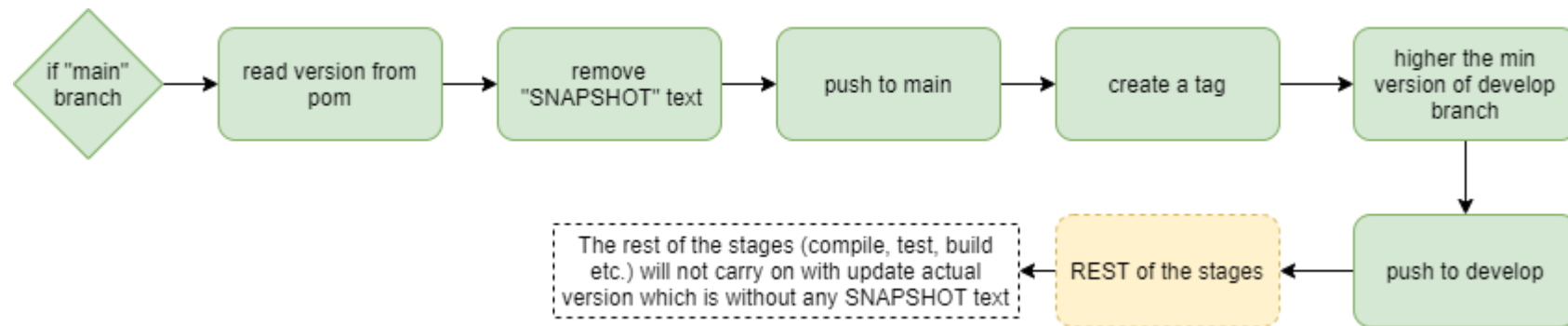


Release meant merging to the main branch.
Update the version in the main pom.
Create a tag from the version.
Update the snapshot version of develop.



AUTOMATIC VERSIONING

Release meant merging to the main branch.
Update the version in the main pom.
Create a tag from the version.
Update the snapshot version of develop.

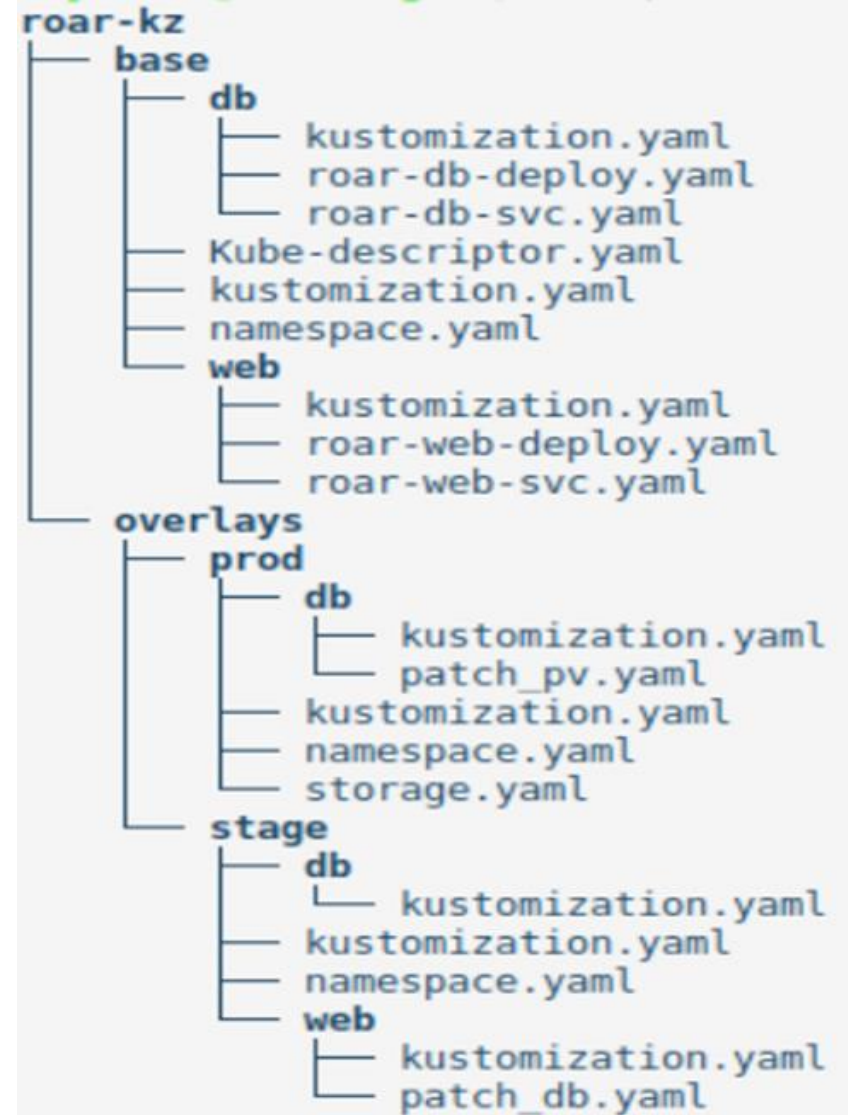


BRANCH PROTECTION RULE

- Non-deletable/Lock branch
- Review rule
- Review Merge rule
- Resolve Comments rule
- Restrict Push to branches



DIRECTORIES VS BRANCHES



APP & CONFIG SEPARATE

Application code and App Config separate
CI and CD separate
Desirably Async
App pipeline ends on container push
Config/deploy pipeline begins on new image

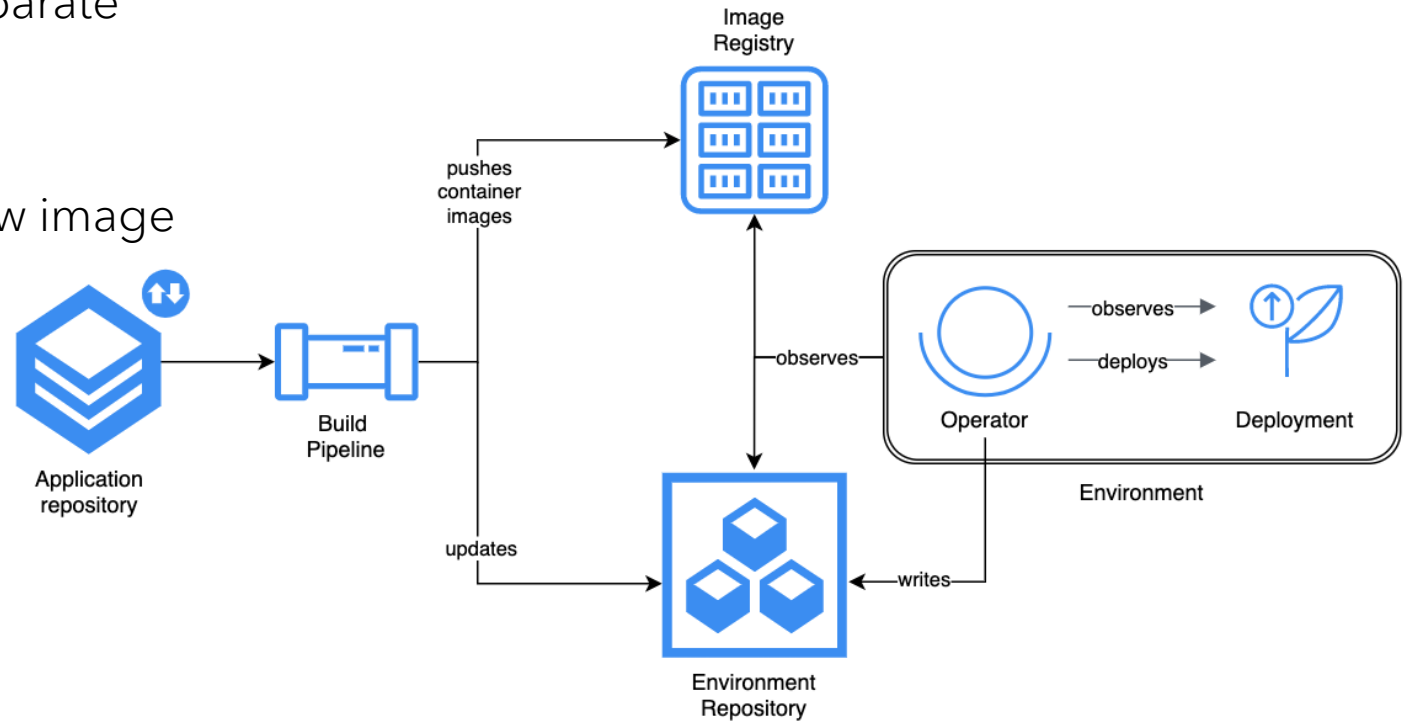
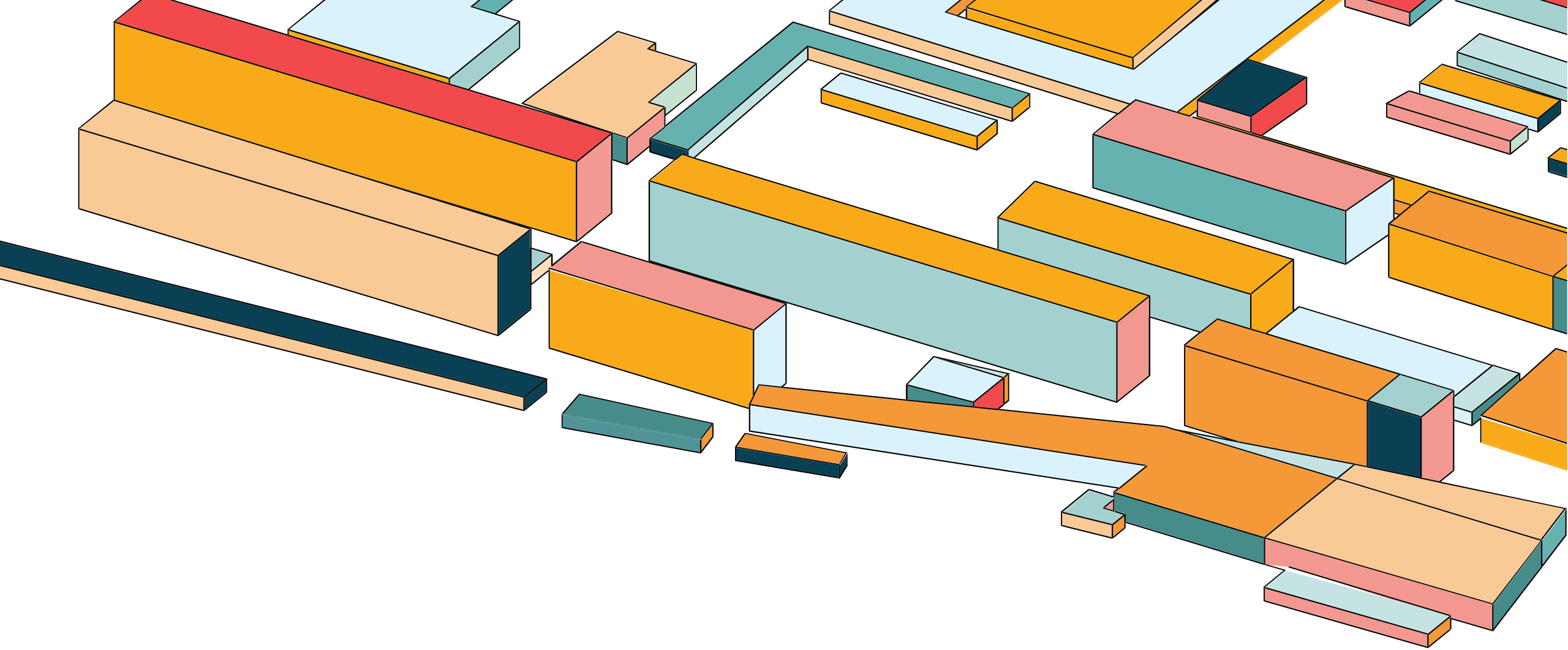


Image from: [GitOps - DevOps for Infrastructure Automation - DZone](#)



MAVEN TRICKS

PARENT POM

Parent POM

Defines a Spring Boot version as Parent.

Defines external library versions as properties.

Defines dependency management for external libs.

Defines dependencies which are automatically added to children.

Defines plugins needed for pipeline stages.

Child POM

Defines Parent POM as Parent.

Defines external library versions as properties.

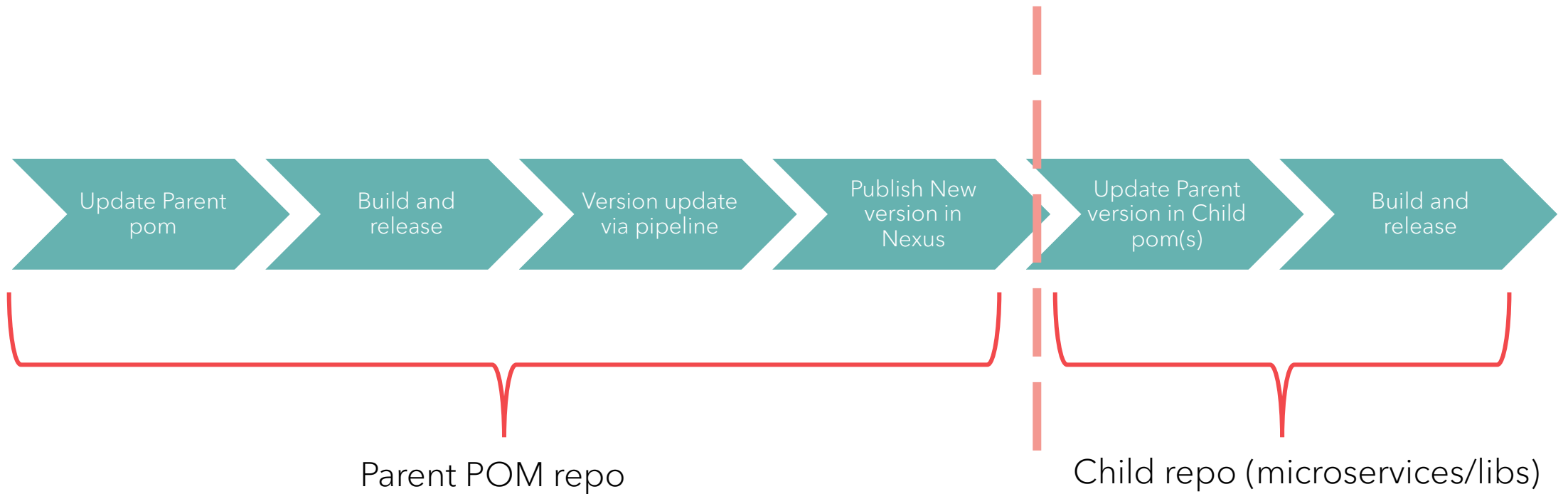
Defines dependencies without version, versions are inherited from Parent POM.

Defines plugins needed for code generations. (OpenAPI)

Optionally, override plugins to tweak execution.

Dependency management, versions
Plugins
Profiles

PARENT POM

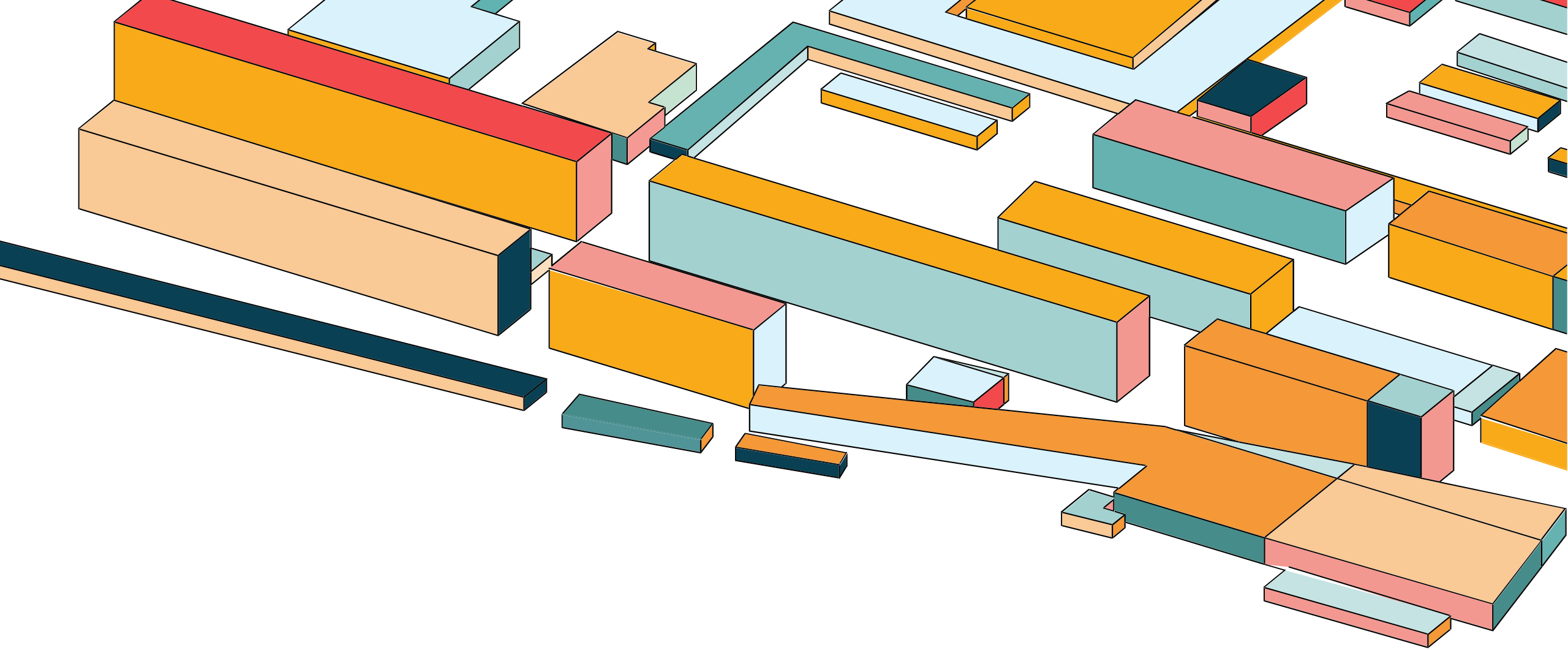


ENFORCER PLUGIN

```
<plugin>
  <groupId>org.apache.maven.plugins</groupId>
  <artifactId>maven-enforcer-plugin</artifactId>
  <version>3.3.0</version>
  <configuration>
    <rules>
      <dependencyConvergence/>
      <requireUpperBoundDeps/>
      <banTransitiveDependencies/>
      <banDuplicatePomDependencyVersions/>
      <bannedDependencies>
        <excludes>
          <exclude>org.apache.commons:commons-lang3</exclude>
        </excludes>
      </bannedDependencies>
    </rules>
  </configuration>
</plugin>
```

Duplicate dependencies
Dependency Convergence
Ban Transitive Dependencies
Banned Dependencies
Require Upper Bounds
Dependencies





API DEFINITION MANAGEMENT

```

bearerFormat: JWT
schemas:
#-----
# Abstract class with discriminator 'User'
#-----
'User':
  type: object
  discriminator:
    propertyName: '@type'
    mapping:
      MANAGER: '#/components/schemas/ManagerUser'
      EMPLOYEE: '#/components/schemas/EmployeeUser'
  required:
    - firstname
    - lastname
    - age
    - department
  properties:
    'firstname':
      type: string
    'lastname':
      type: string
    'department':
      type: string
    'age':
      type: integer
#-----
# Concrete classes
#-----
ManagerUser:
  allOf:
    - $ref: "#/components/schemas/User"
    - type: object
  required:
    - managingDepartment
    - directs
  properties:
    'managingDepartment':
      type: string
    'directs':
      type: array
      items:
        type: object
        $ref: "#/components/schemas/EmployeeUser"
EmployeeUser:

```

OpenAPI Specification with polymorphism 1.0.0 OAS 3.0

Servers

https://example-service/ - dev namespace

Authorize

User

GET

/user/{username}

DELETE

/user/{username} Deletes user based on username

Schemas

User >

ManagerUser >

EmployeeUser >

OPENAPI

CODE GENERATION

OpenAPI generator - Server

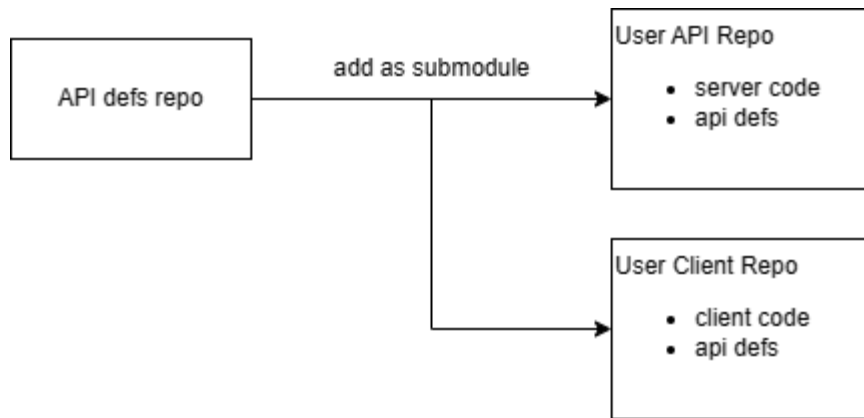
```
<plugin>
  <groupId>org.openapitools</groupId>
  <artifactId>openapi-generator-maven-plugin</artifactId>
  <version>${openapi-generator-maven-plugin.version}</version>
  <executions>
    <execution>
      <goals>
        <goal>generate</goal>
      </goals>
      <configuration>
        <inputSpec>
          ${project.basedir}/src/main/resources/petstore_api.yml
        </inputSpec>
        <generatorName>spring</generatorName>
        <apiPackage>com.company.generated.petstore.controller</apiPackage>
        <modelPackage>com.company.generated.petstore.model</modelPackage>
        <configOptions>
          <apiDocs>true</apiDocs>
          <delegatePattern>true</delegatePattern>
          <interfaceOnly>true</interfaceOnly>
          <serializableModel>true</serializableModel>
          <ensureUniqueParams>false</ensureUniqueParams>
        </configOptions>
      </configuration>
    </execution>
  </executions>
</plugin>
```

Kiota- Client

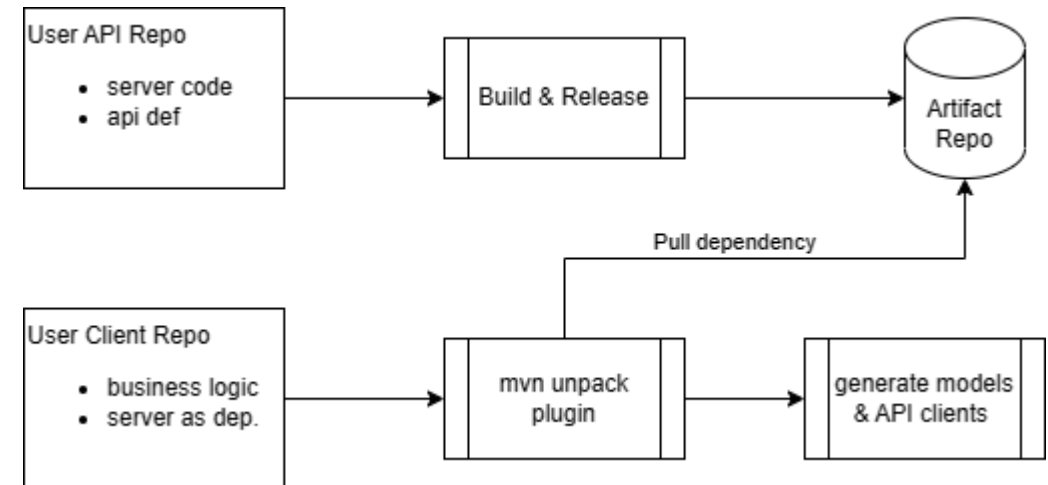
```
<plugin>
  <artifactId>kiota-maven-plugin</artifactId>
  <groupId>com.redhat.cloud</groupId>
  <version>0.0.4</version>
  <executions>
    <execution>
      <goals>
        <goal>generate</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <file>${project.basedir}/src/main/resources/apiSpecs.yml</file>
  </configuration>
</plugin>
```

GIT SUBMODULE VS MAVEN UNPACK

Submodule



Unpack



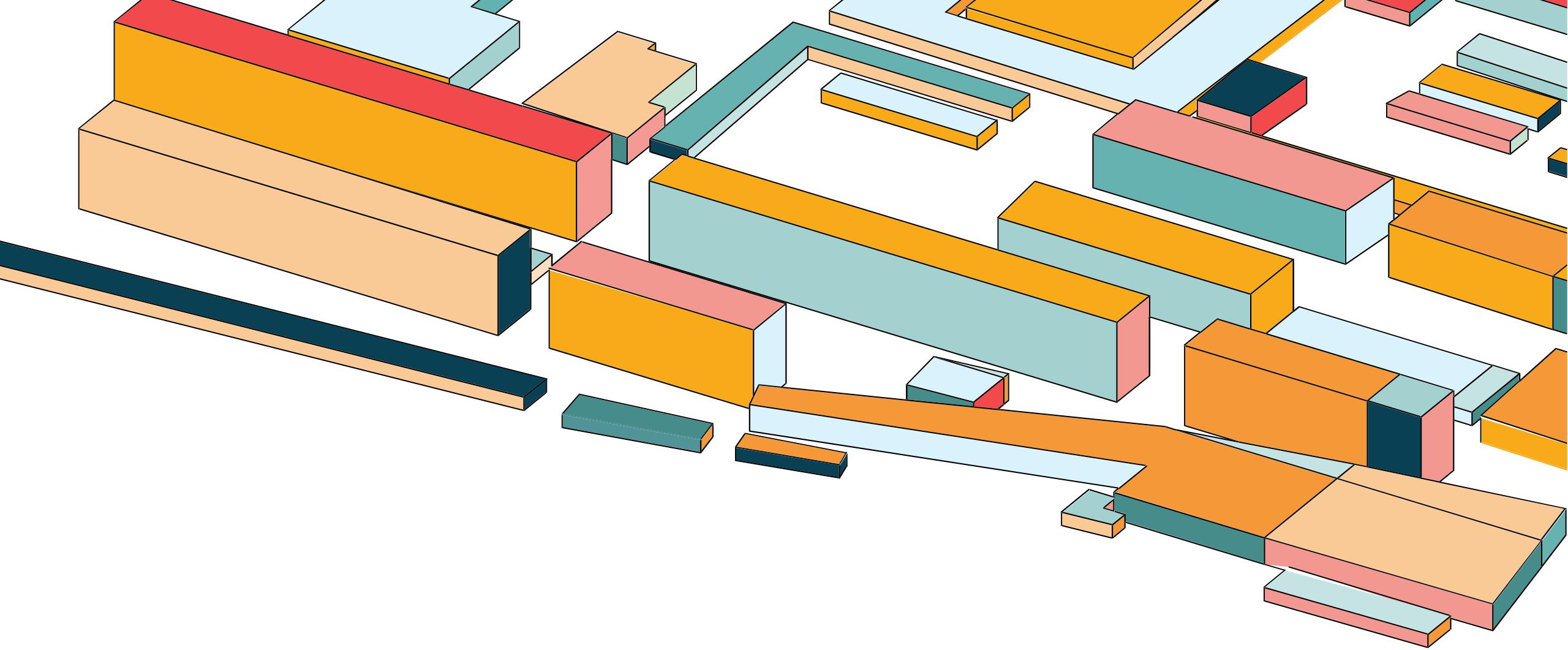
API DEF. CONSUMPTION

OpenAPI generator

```
<groupId>org.openapitools</groupId>
<artifactId>openapi-generator-maven-plugin</artifactId>
<version>${openapi-generator-maven-plugin.version}</version>
<executions>
  <execution>
    <id>pet-store-api</id>
    <goals>
      <goal>generate</goal>
    </goals>
    <configuration>
      <inputSpec>
        ${project.basedir}/petstore_api.yml
      </inputSpec>
      <generatorName>java</generatorName>
      <addCompileSourceRoot>true</addCompileSourceRoot>
      <apiPackage>com.company.generated.pestore.api</apiPackage>
      <modelPackage>com.company.generated.pestore.model.rest</modelPackage>
      <generateApis>true</generateApis>
      <generateModels>true</generateModels>
      <generateModelDocumentation>false</generateModelDocumentation>
      <generateModelTests>false</generateModelTests>
      <generateApiTests>false</generateApiTests>
      <library>feign</library>
      <configOptions>
        <dateLibrary>java8</dateLibrary>
        <sourceFolder>swagger</sourceFolder>
        <interfaceOnly>false</interfaceOnly>
        <java8>true</java8>
      </configOptions>
    </configuration>
  </execution>
</executions>
```

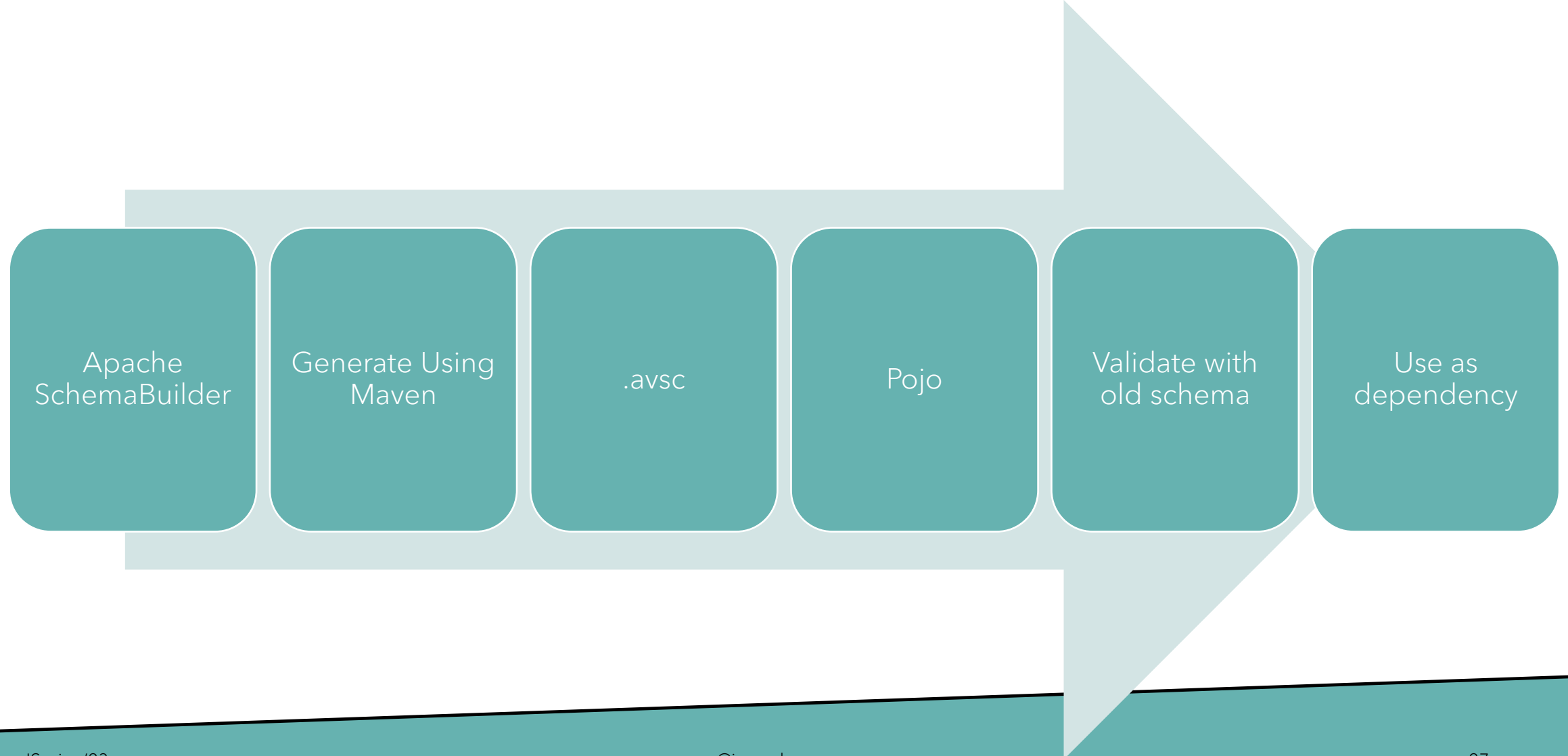
Kiota

```
<plugin>
  <artifactId>kiota-maven-plugin</artifactId>
  <groupId>com.redhat.cloud</groupId>
  <version>0.0.4</version>
  <executions>
    <execution>
      <goals>
        <goal>generate</goal>
      </goals>
    </execution>
  </executions>
  <configuration>
    <file>${project.basedir}/src/main/resources/apiSpecs.yml</file>
  </configuration>
</plugin>
```



HANDY STUFFS

KAFKA SCHEMA MGMT.



UNIT TESTS AS NATURAL LANGUAGE

Scenario: User api invoked for A
Given User API called with name A
When Response received
Then Last name is B
And DOB is 25-06-1983

Cucumber
Engine

Unit test
classes with
Mocks/TC

Publish them
to Wiki

Generate
HTML/JSON
report

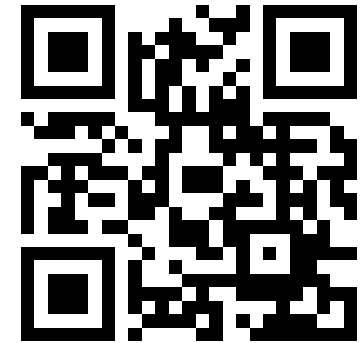
Happy Stakeholders
Readable Unit test report for
Business Users
Make your team a true T/M shape

AWAITILITY

```

@Test
public void updatesCustomerStatus() {
    // Publish an asynchronous message to a broker (e.g. RabbitMQ):
    messageBroker.publishMessage(updateCustomerStatusMessage);

    // Awaitility lets you wait until the asynchronous operation completes:
    await().atMost(5, SECONDS).until(customerStatusIsUpdated());
    ...
}
```



[Awaitility](https://awaitility.org/)

PITEST

```
public Long doSomeComplexMath(Long number1, Long number2, Long number3) {  
    if(number3 == 0) {  
        throw new RuntimeException("number3 cannot be zero");  
    }  
    Long firstDoSum = doSum(number1, number2);  
    Long division = doDivision(firstDoSum, number3);  
    return doSum(doMultiplication(number2, division), constantFactor);  
}
```

```
@Test  
public void testNonZeroNumber3() {  
    Throwable exception = assertThrows(RuntimeException.class, () -> {  
        calculator.doSomeComplexMath(2l, 3l, 0l);  
    });  
    assertEquals("number3 cannot be zero", exception.getMessage());  
}  
  
@Test  
public void checkCalculation() {  
    Long result = calculator.doSomeComplexMath(3l, 4l, 7l);  
    assertEquals(26l, result);  
}
```

Calculator.java

```
1 package com.pitest.demo;
2
3 public class Calculator {
4
5     private final Long constantFactor = 22L;
6
7     public Long doSomeComplexMath(Long number1, Long number2, Long number3) {
8         if(number3 == 0)
9             throw new RuntimeException("Division by zero");
10        }
11        Long firstDoSum = doSum(number1, number2);
12        Long division = doDivision(number1, number2);
13        return doSum(doMultiplication(number1, number2), firstDoSum);
14    }
15
16    private Long doSum(Long number1, Long number2) {
17        return number1 + number2;
18    }
19
20    private Long doDivision(Long number1, Long number2) {
21        if(number1 > number2)
22            return number1 / number2;
23        } else {
24            return number2 / number1;
25        }
26    }
27
28    private Long doMultiplication(Long number1, Long number2) {
29        return number1 * number2;
30    }
31}
```

Mutations

- 8 1. negated conditional → KILLED
- 13 1. replaced Long return value with 0L for com/pitest/demo/Calculator::doSomeComplexMath → KILLED
- 17 1. Replaced long addition with subtraction → KILLED
2. replaced Long return value with 0L for com/pitest/demo/Calculator::doSum → KILLED
- 21 1. changed conditional boundary → SURVIVED
2. negated conditional → SURVIVED
- 22 1. Replaced long division with multiplication → NO_COVERAGE
2. replaced Long return value with 0L for com/pitest/demo/Calculator::doDivision → NO_COVERAGE
- 24 1. Replaced long division with multiplication → KILLED
2. replaced Long return value with 0L for com/pitest/demo/Calculator::doDivision → KILLED
- 29 1. Replaced long multiplication with division → SURVIVED
2. replaced Long return value with 0L for com/pitest/demo/Calculator::doMultiplication → KILLED



PITEST

```
<plugin>
  <groupId>org.pitest</groupId>
  <artifactId>pitest-maven</artifactId>
  <version>${pitest-maven.version}</version>
  <configuration>
    <verbose>true</verbose>
    <maxDependencyDistance>1</maxDependencyDistance>
    <timeoutFactor>0.7</timeoutFactor>
    <threads>6</threads>
    <coverageThreshold>80</coverageThreshold>
    <mutationThreshold>80</mutationThreshold>
    <excludedClasses>
      <param>com.demo.generated.*</param>
      <param>org.springframework.jms.config.*</param>
      <param>org/slf4j/Logger.*</param>
      <param>com.demo.*.configuration.*</param>
    </excludedClasses>
    <avoidCallsTo>
      <avoidCallsTo>com.demo.generated</avoidCallsTo>
      <avoidCallsTo>org/springframework/jms/config</avoidCallsTo>
      <avoidCallsTo>org/slf4j/Logger</avoidCallsTo>
      <avoidCallsTo>com.demo.*.configuration</avoidCallsTo>
    </avoidCallsTo>
  </configuration>
  <dependencies>
    <dependency>
      <groupId>org.pitest</groupId>
      <artifactId>pitest-junit5-plugin</artifactId>
      <version>${pitest-junit5-plugin.version}</version>
    </dependency>
  </dependencies>
</plugin>
```

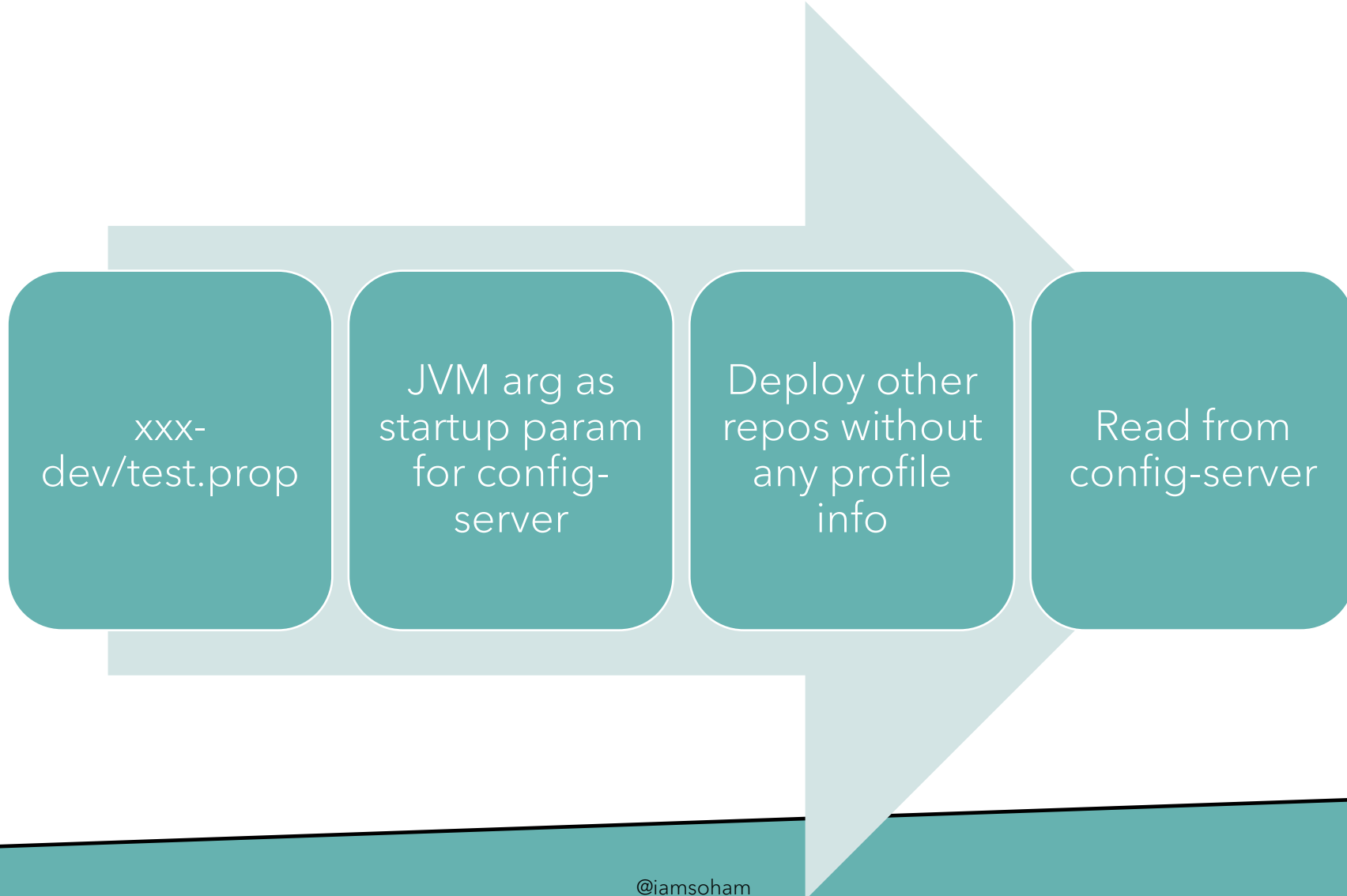

SPRING PROFILE

xxx-
dev/test.prop

JVM arg as
startup param

Read from
config-server
with activated
profile

SPRING PROFILE



THANK YOU

Code and Presentation

