Google Cloud

Edge API Lifecycle and Tools

Maven Dependency Plugin

Prerequisites

1. Java
2. Maven
3. Use of Maven in CI tooling
4. Maven deploy plugin

Dependency Management

The problem:

- Uncontrolled duplication of code becomes a maintenance nightmare.
- Use of Maven promotes DRY (Don’t Repeat Yourself).

Dependency Management

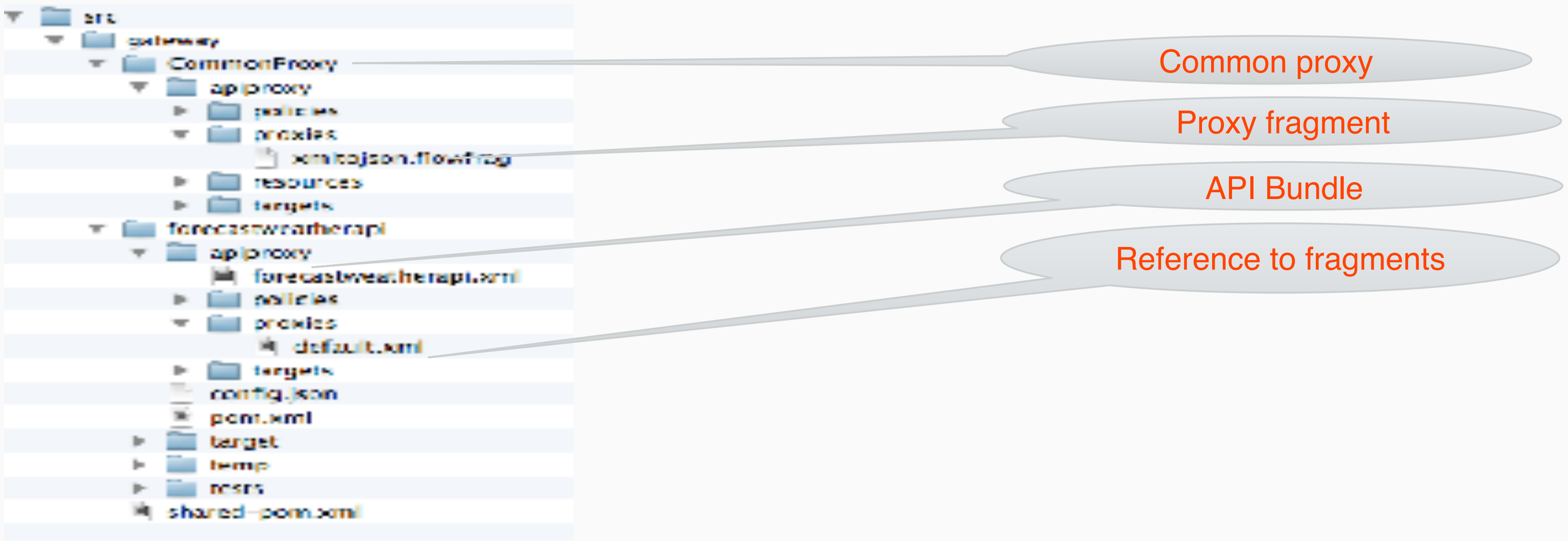
- Supports Two kinds of dependency resolution:
  - **Policy resolution** - An API proxy's proxy endpoint or target endpoint can reference policies that are common. These referenced policies are copied by the plugin to the policies directory of the API proxy.
  - **Flow resolution -** Flow resolution is supported by defining a new type of file called flowfrag(Flow fragment) file. Flow fragment is a text file containing a list of reusable steps with extension *.flowfrag*
- Flow fragment can be referenced by proxy or target endpoint using the format #flow-fragment-base-filename# without the .flowfrag extension.
- At build time, such reference results in macro style substitution of the content of the flow fragment file, followed by policy resolution of the all the policies referenced by the fragment

```
<Step>
           <Name>policy1</Name>
</Step>
<Step>
           <Name>policy2</Name>
</Step>
```

Dependency Management

Basic Concepts:

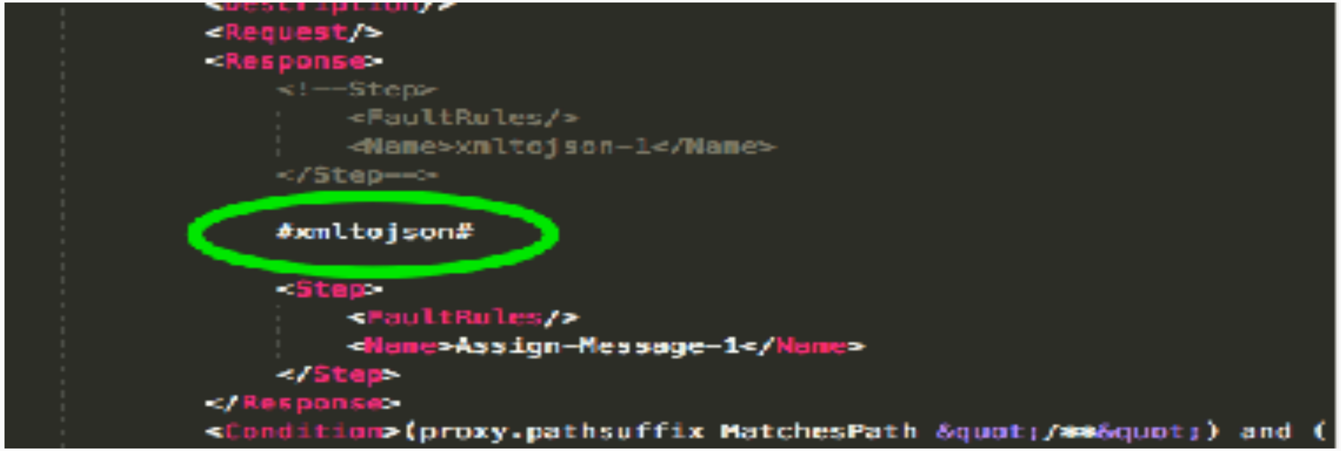
Common Fragments shared across multiple API Bundles



```
src
  gateway
    CommonProxy
      apiproxy
        policies
        proxies
          xmltojson.flowfrag
        resources
        targets
    forecastweatherapi
      apiproxy
        forecastweatherapi.xml
        policies
        proxies
          default.xml
        targets
      config.json
      pom.xml
      target
      temp
      tests
    shared-pom.xml
```

Dependency Management

Fragment reference from proxies/default.xml



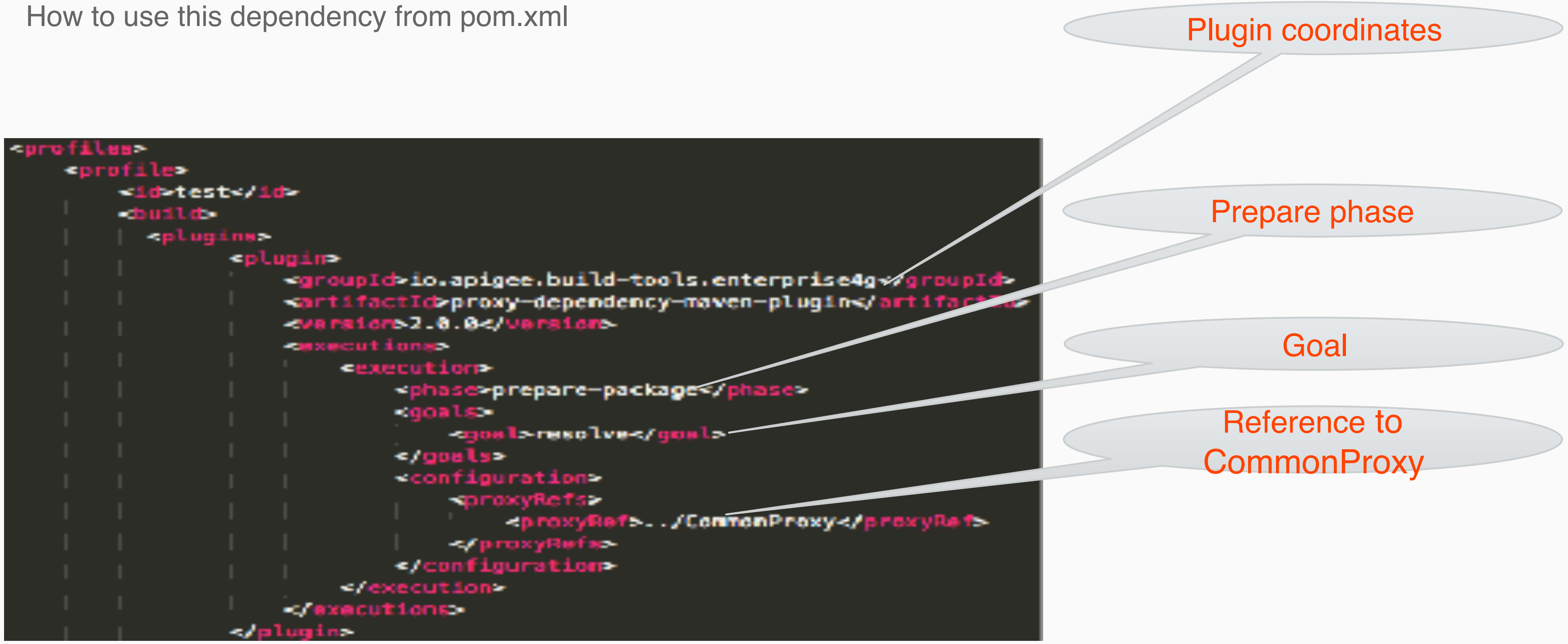
```
<Description/>
<Request/>
<Response>
    <!--Step>
        <FaultRules/>
        <Name>xmltojson-1</Name>
    </Step-->

    #xmltojson#

    <Step>
        <FaultRules/>
        <Name>Assign-Message-1</Name>
    </Step>
</Response>
<Condition>(proxy.pathsuffix MatchesPath &quot;/**&quot;) and (
```

Dependency Management

How to use this dependency from pom.xml



```
<profiles>
    <profile>
        <id>test</id>
        <build>
          <plugins>
                <plugin>
                    <groupId>io.apigee.build-tools.enterprise4g</groupId>
                    <artifactId>proxy-dependency-maven-plugin</artifactId>
                    <version>2.0.0</version>
                    <executions>
                        <execution>
                            <phase>prepare-package</phase>
                            <goals>
                                <goal>resolve</goal>
                            </goals>
                            <configuration>
                                <proxyRefs>
                                    <proxyRef>../CommonProxy</proxyRef>
                                </proxyRefs>
                            </configuration>
                        </execution>
                    </executions>
                </plugin>
```

Lab

API dev best practices

Use version control (git) as single source of truth for common code and proxy

Know who made a change to the common code

Know when a change was made

Plug this as part of automation

Benefits

- *Code re-use*
- *Avoids/Reduces manual tasks in promoting API from one env to another*
- *Avoids/Reduces manual tasks in moving API from one org to another*

Disclaimer

- Open source project github project
  https://github.com/apigee/proxy-dependency-maven-plugin
- Contribute and join the community
- Use community.apigee.com to ask questions
- File issues in github.com to report bugs or enhancements

THANK YOU