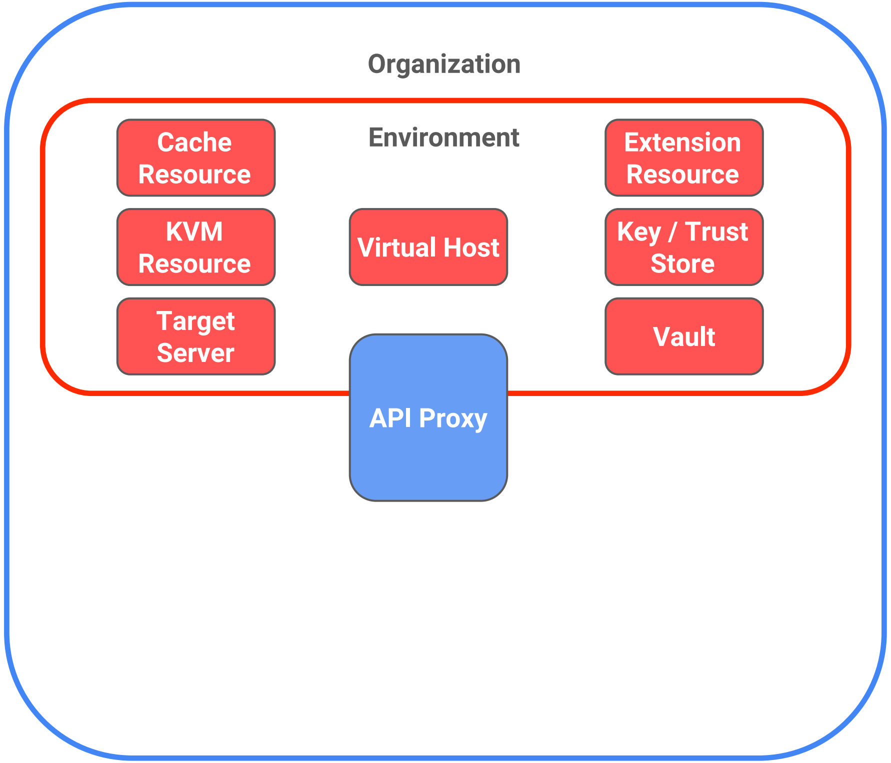
Anatomy of an API Proxy

Platform Entities

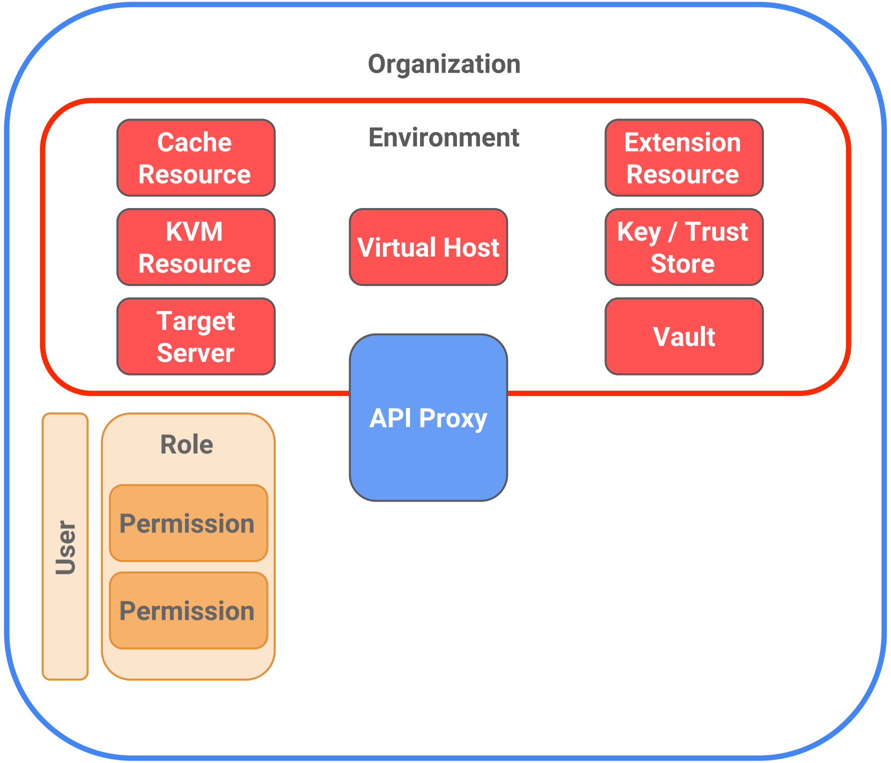
Platform Entities

Organization

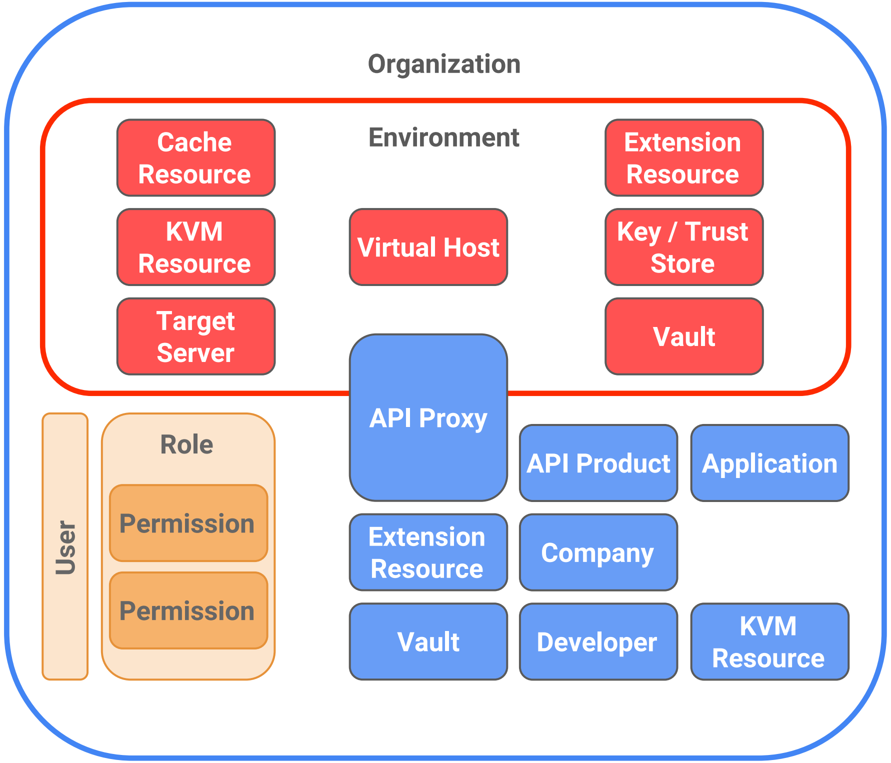
Platform Entities



Platform Entities

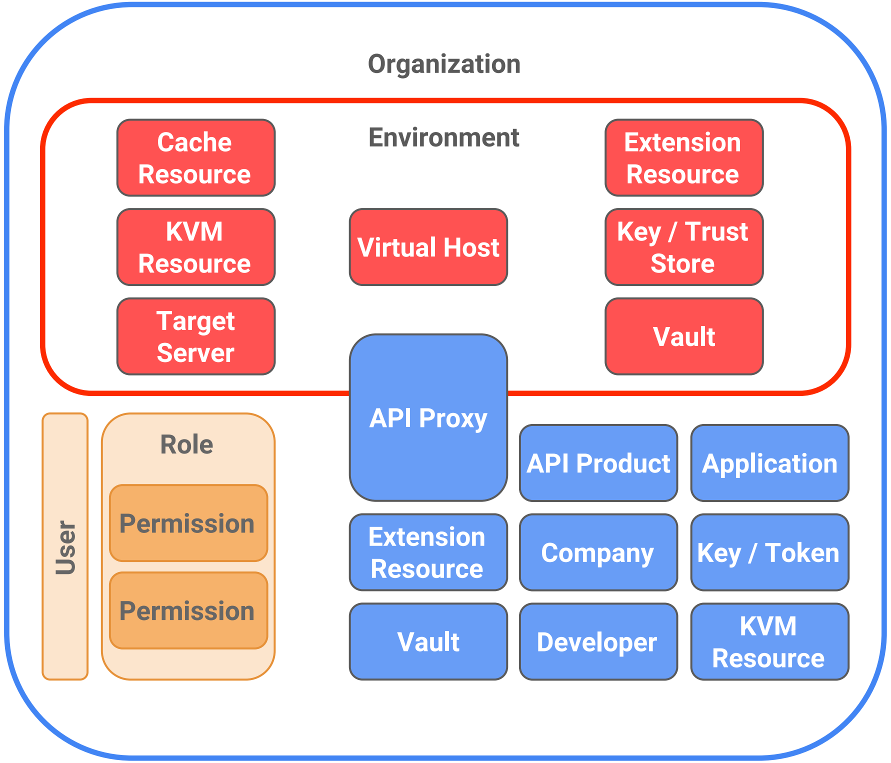


Platform Entities

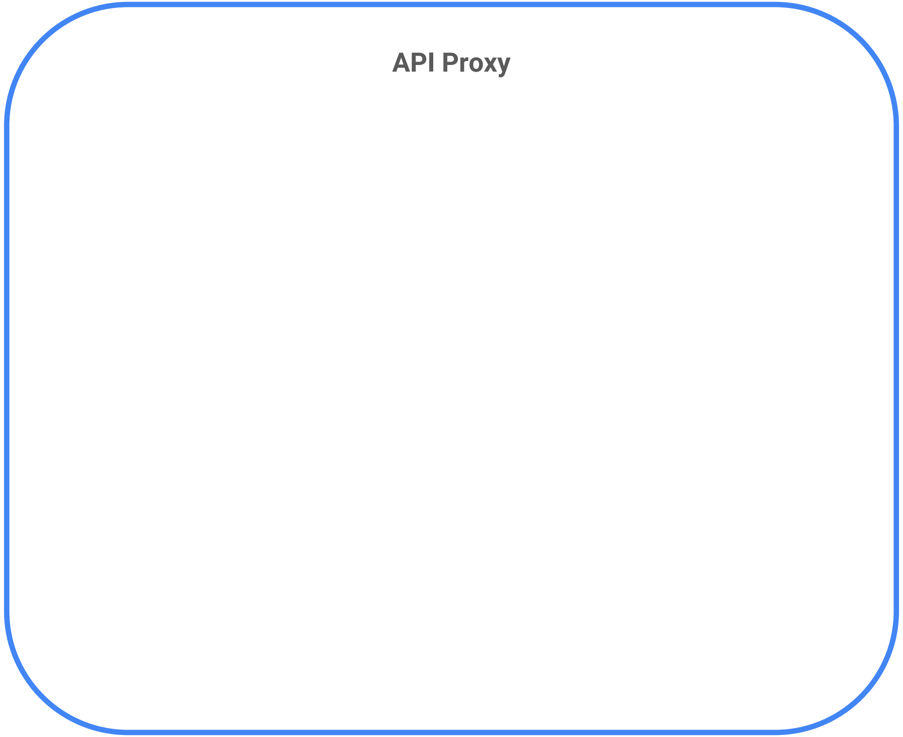


apigee

Platform Entities

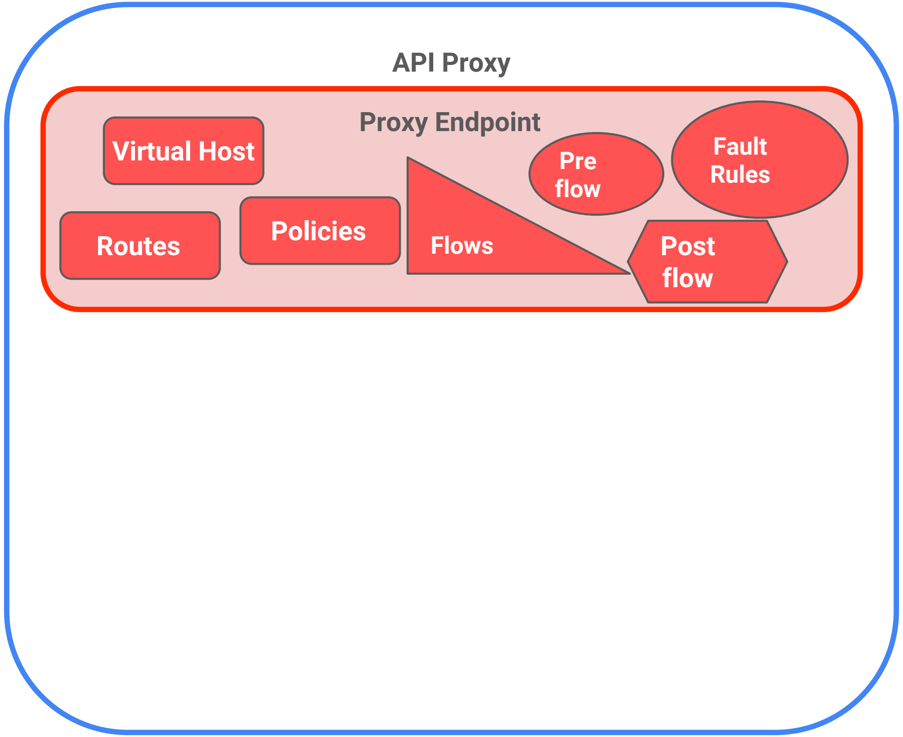


API Proxies

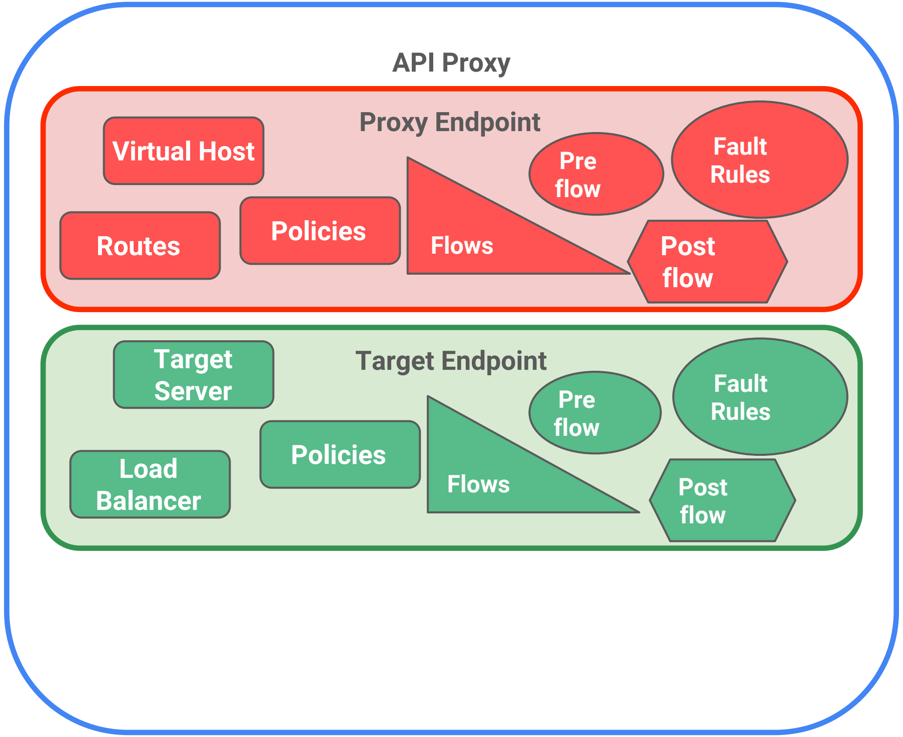


API Proxy

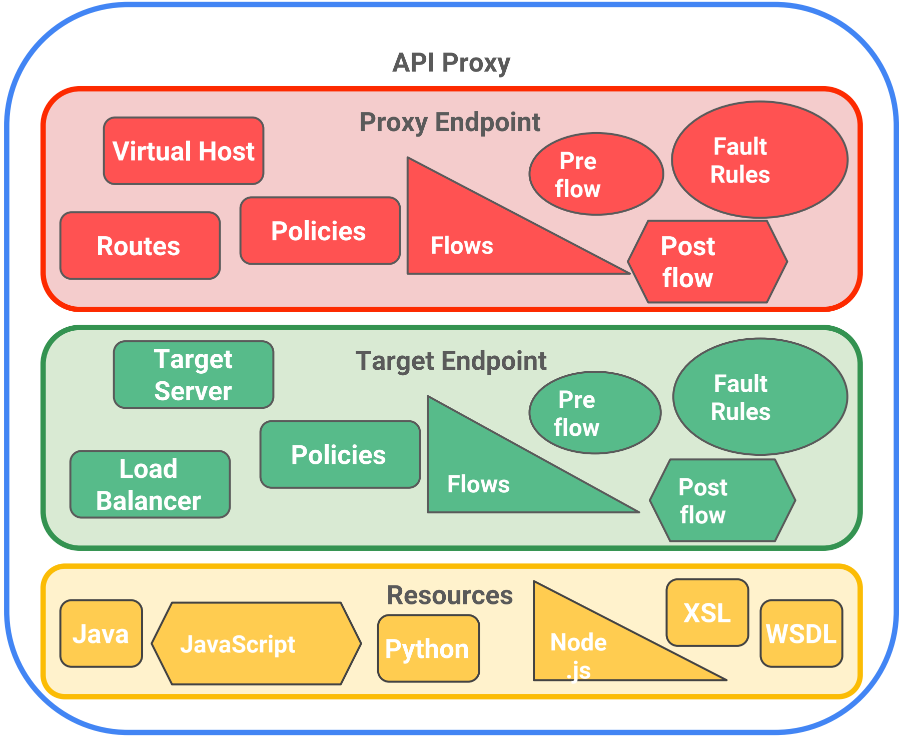
API Proxies


API Proxy
Proxy Endpoint
Virtual Host
Pre flow
Fault Rules
Routes
Policies
Flows
Post flow

API Proxies

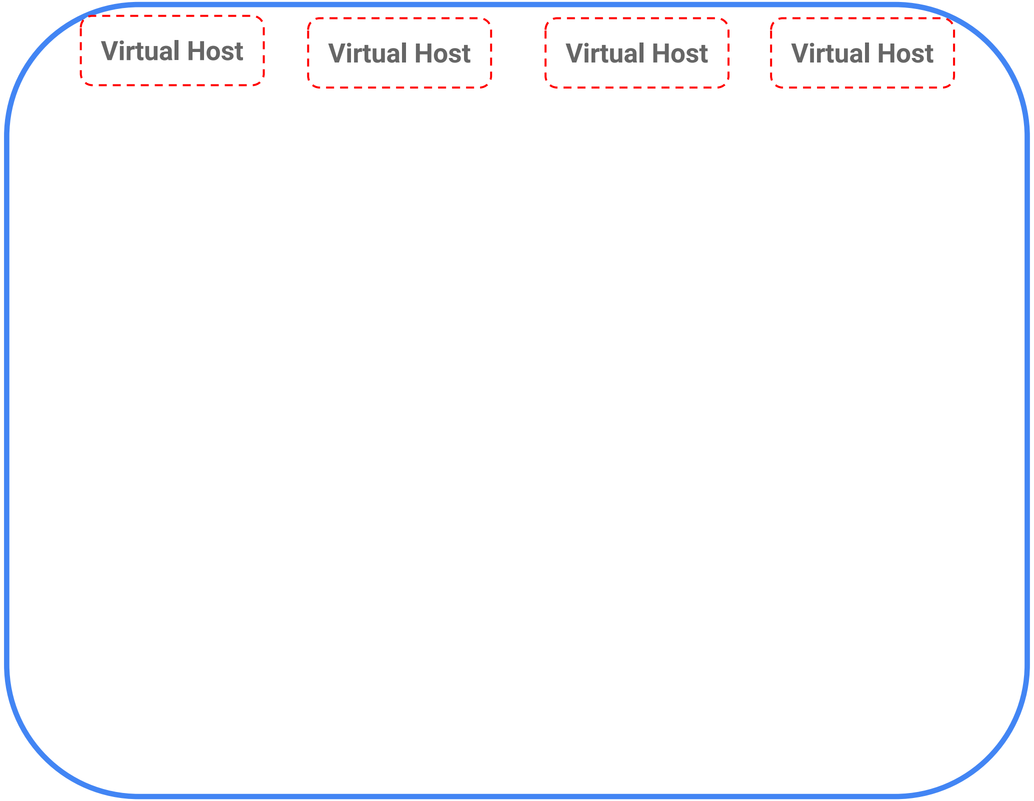

API Proxy
Proxy Endpoint
Virtual Host
Routes
Policies
Flows
Pre flow
Fault Rules
Post flow
Target Endpoint
Target Server
Load Balancer
Policies
Flows
Pre flow
Fault Rules
Post flow

API Proxies


API Proxy
Proxy Endpoint
Virtual Host
Routes
Policies
Flows
Pre flow
Fault Rules
Post flow
Target Endpoint
Target Server
Load Balancer
Policies
Flows
Pre flow
Fault Rules
Post flow
Resources
Java
JavaScript
Python
Node .js
XSL
WSDL

Proxy Endpoint

Proxy Endpoint



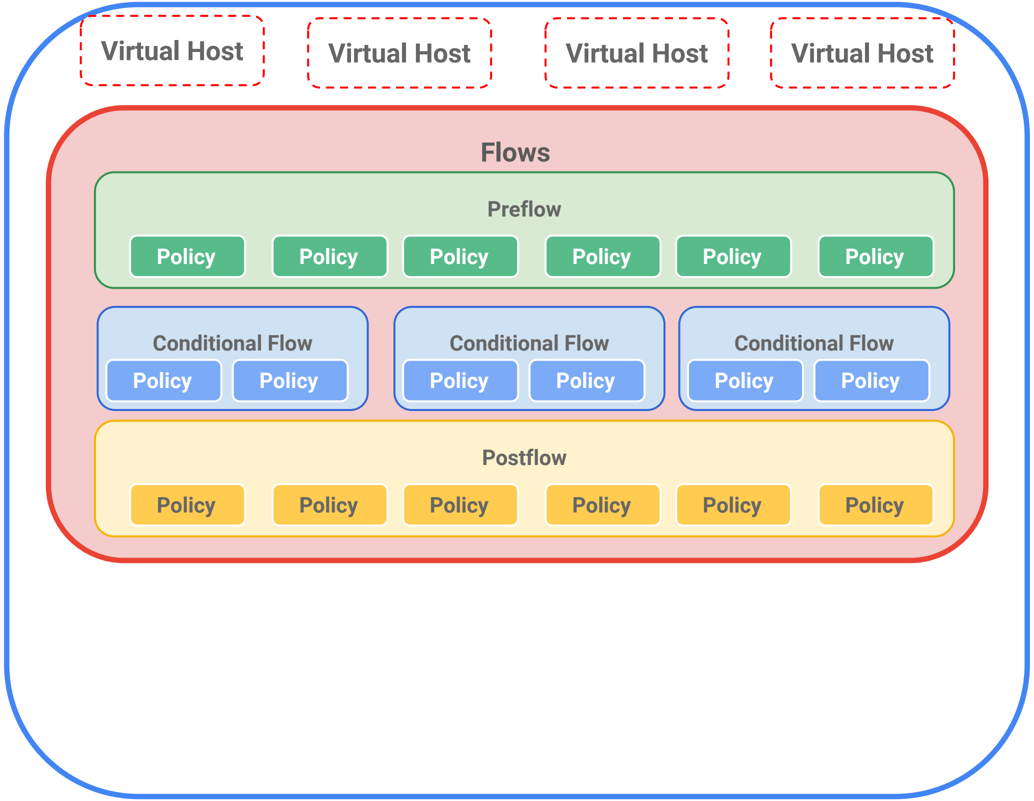
Virtual Host

Virtual Host

Virtual Host

Virtual Host

Proxy Endpoint



Virtual Host

Virtual Host

Virtual Host

Virtual Host

Flows

Preflow

Policy Policy Policy Policy Policy Policy

Conditional Flow

Policy Policy

Conditional Flow

Policy Policy

Conditional Flow

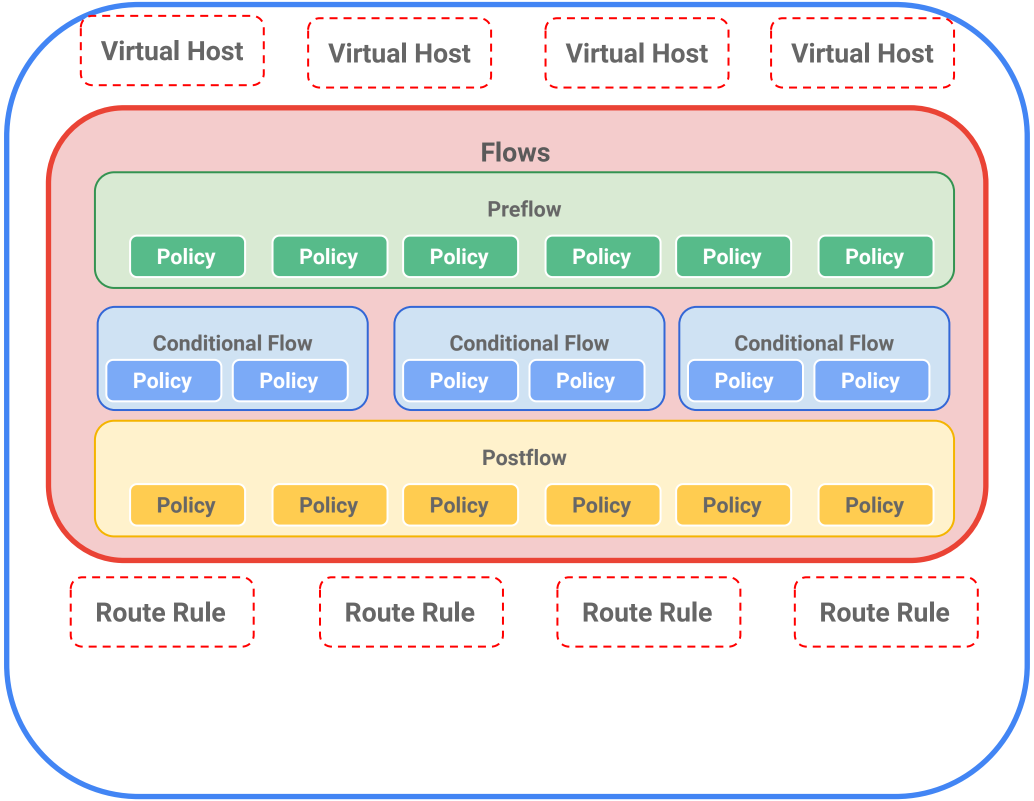
Policy Policy

Postflow

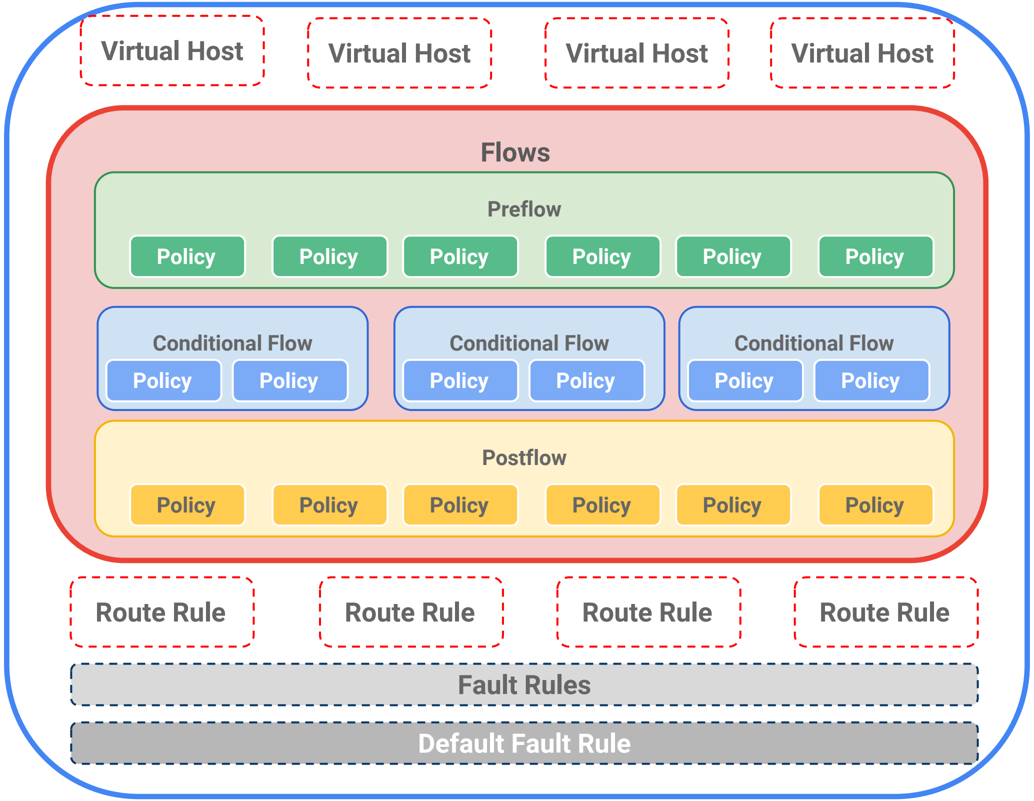
Policy Policy Policy Policy Policy Policy

apigee

Proxy Endpoint



Proxy Endpoint


Virtual Host
Virtual Host
Virtual Host
Virtual Host

Flows

Preflow

Policy Policy Policy Policy Policy Policy

Conditional Flow
Policy Policy

Conditional Flow
Policy Policy

Conditional Flow
Policy Policy

Postflow

Policy Policy Policy Policy Policy Policy

Route Rule
Route Rule
Route Rule
Route Rule

Fault Rules

Default Fault Rule

Proxy Endpoint Example

```
<ProxyEndpoint name="default">
    <Description/>
    <DefaultFaultRule name="generic_fault_handler">
        <Step>
            <Name>RF_GenericFault</Name>
            <AlwaysEnforce>true</AlwaysEnforce>
        </Step>
    </DefaultFaultRule>
    <FaultRules>
        <FaultRule name="invalid_api_key_fault">
            <Step>
                <Name>RF_InvalidKey</Name>
                <Condition>fault.name="InvalidApiKey"</Condition>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow"/>
    <Flows/>
    <PostFlow name="PostFlow"/>
    <HTTPProxyConnection>
        <BasePath>/v1/someAPI</BasePath>
        <Properties/>
        <VirtualHost>default</VirtualHost>
    </HTTPProxyConnection>
    <RouteRule name="default">
       <TargetEndpoint>default</TargetEndpoint>
    </RouteRule>
</ProxyEndpoint>
```

Proxy Endpoint Example

```
<ProxyEndpoint name="default">
    <Description/>
    <DefaultFaultRule name="generic_fault_handler">
        <Step>
            <Name>RF_GenericFault</Name>
            <AlwaysEnforce>true</AlwaysEnforce>
        </Step>
    </DefaultFaultRule>
    <FaultRules>
        <FaultRule name="invalid_api_key_fault">
            <Step>
                <Name>RF_InvalidKey</Name>
                <Condition>fault.name="InvalidApiKey"</Condition>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow"/>
    <Flows/>
    <PostFlow name="PostFlow"/>
    <HTTPProxyConnection>
        <BasePath>/v1/someAPI</BasePath>
        <Properties/>
        <VirtualHost>default</VirtualHost>
    </HTTPProxyConnection>
    <RouteRule name="default">
       <TargetEndpoint>default</TargetEndpoint>
    </RouteRule>
</ProxyEndpoint>
```

Fault Rules

Proxy Endpoint Example

```
<ProxyEndpoint name="default">
    <Description/>
    <DefaultFaultRule name="generic_fault_handler">
        <Step>
            <Name>RF_GenericFault</Name>
            <AlwaysEnforce>true</AlwaysEnforce>
        </Step>
    </DefaultFaultRule>
    <FaultRules>
        <FaultRule name="invalid_api_key_fault">
            <Step>
                <Name>RF_InvalidKey</Name>
                <Condition>fault.name="InvalidApiKey"</Condition>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow"/>
    <Flows/>
    <PostFlow name="PostFlow"/>
    <HTTPProxyConnection>
        <BasePath>/v1/someAPI</BasePath>
        <Properties/>
        <VirtualHost>default</VirtualHost>
    </HTTPProxyConnection>
    <RouteRule name="default">
       <TargetEndpoint>default</TargetEndpoint>
    </RouteRule>
</ProxyEndpoint>
```

Fault Rules

Flows

Proxy Endpoint Example

```
<ProxyEndpoint name="default">
    <Description/>
    <DefaultFaultRule name="generic_fault_handler">
        <Step>
            <Name>RF_GenericFault</Name>
            <AlwaysEnforce>true</AlwaysEnforce>
        </Step>
    </DefaultFaultRule>
    <FaultRules>
        <FaultRule name="invalid_api_key_fault">
            <Step>
                <Name>RF_InvalidKey</Name>
                <Condition>fault.name="InvalidApiKey"</Condition>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow"/>
    <Flows/>
    <PostFlow name="PostFlow"/>
    <HTTPProxyConnection>
        <BasePath>/v1/someAPI</BasePath>
        <Properties/>
        <VirtualHost>default</VirtualHost>
    </HTTPProxyConnection>
    <RouteRule name="default">
       <TargetEndpoint>default</TargetEndpoint>
    </RouteRule>
</ProxyEndpoint>
```

Fault Rules

Flows

VirtualHost / BasePath

apigee

Proxy Endpoint Example

```
<ProxyEndpoint name="default">
    <Description/>
    <DefaultFaultRule name="generic_fault_handler">
        <Step>
            <Name>RF_GenericFault</Name>
            <AlwaysEnforce>true</AlwaysEnforce>
        </Step>
    </DefaultFaultRule>
    <FaultRules>
        <FaultRule name="invalid_api_key_fault">
            <Step>
                <Name>RF_InvalidKey</Name>
                <Condition>fault.name="InvalidApiKey"</Condition>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow"/>
    <Flows/>
    <PostFlow name="PostFlow"/>
    <HTTPProxyConnection>
        <BasePath>/v1/someAPI</BasePath>
        <Properties/>
        <VirtualHost>default</VirtualHost>
    </HTTPProxyConnection>
    <RouteRule name="default">
       <TargetEndpoint>default</TargetEndpoint>
    </RouteRule>
</ProxyEndpoint>
```

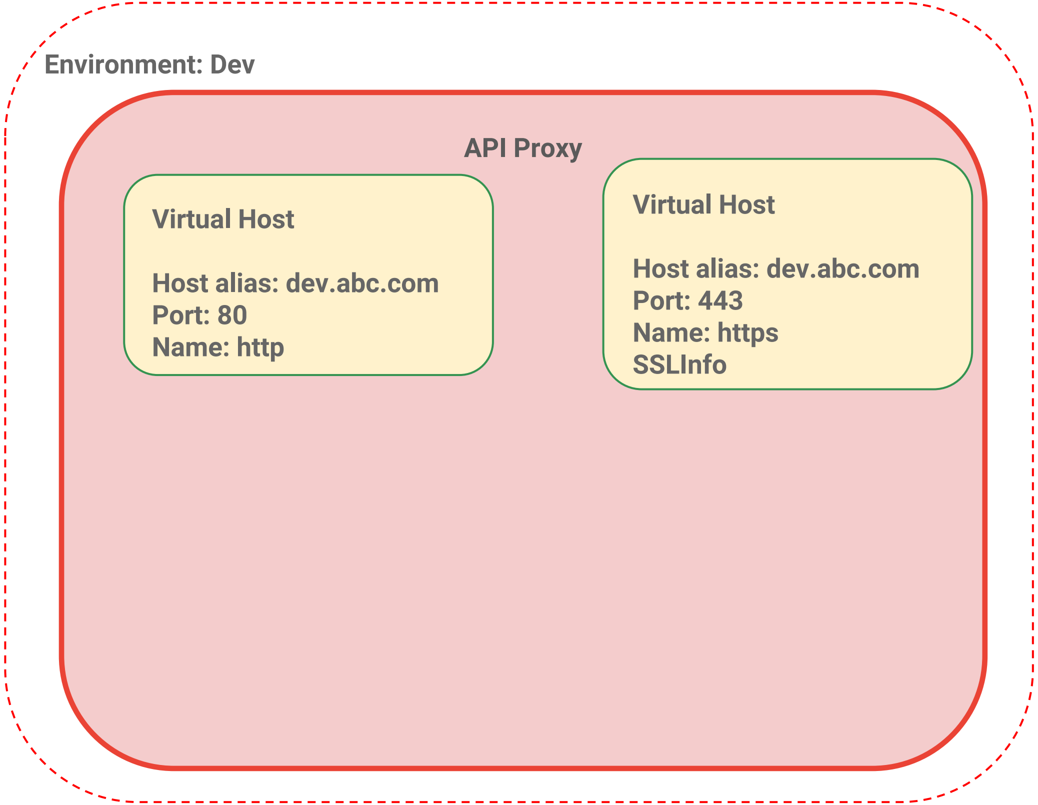
Fault Rules

Flows

VirtualHost / BasePath

RouteRule

Virtual Host



Environment: Dev

API Proxy

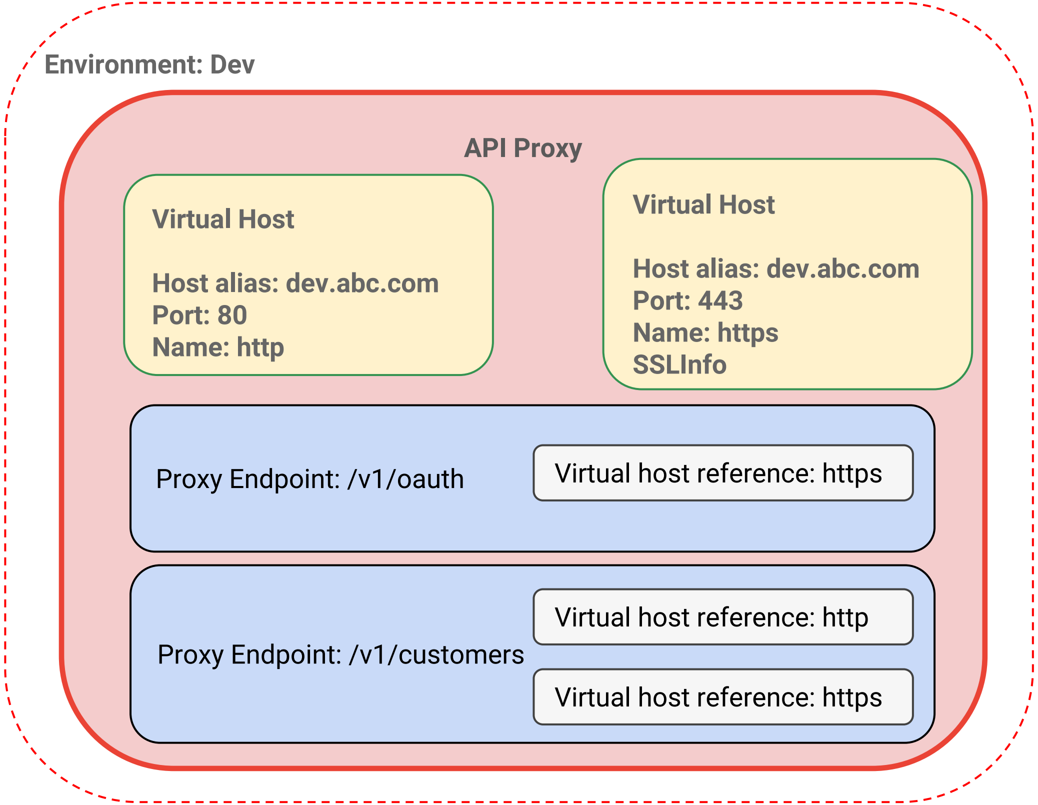
Virtual Host

Host alias: dev.abc.com
Port: 80
Name: http

Virtual Host

Host alias: dev.abc.com
Port: 443
Name: https
SSLInfo

Virtual Host



Environment: Dev

API Proxy

Virtual Host

Host alias: dev.abc.com
Port: 80
Name: http

Virtual Host

Host alias: dev.abc.com
Port: 443
Name: https
SSLInfo

Proxy Endpoint: /v1/oauth

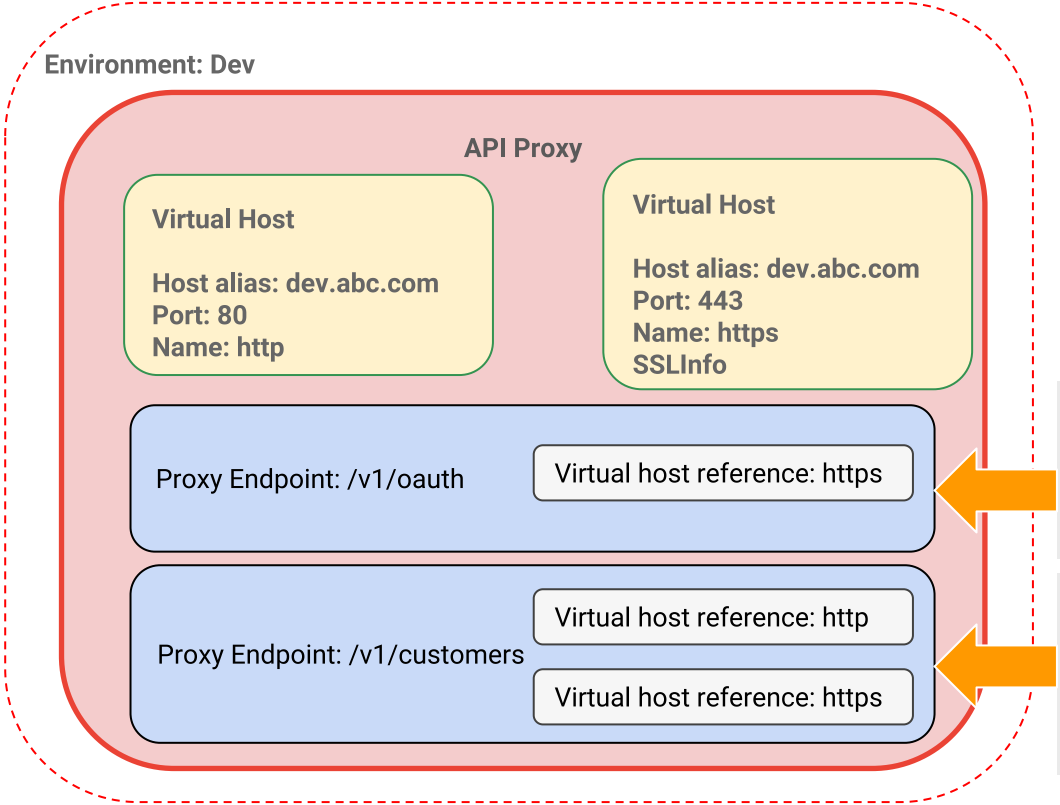
Virtual host reference: https

Proxy Endpoint: /v1/customers

Virtual host reference: http

Virtual host reference: https

Virtual Host



Environment: Dev

API Proxy

Virtual Host

Host alias: dev.abc.com
Port: 80
Name: http

Virtual Host

Host alias: dev.abc.com
Port: 443
Name: https
SSLInfo

Proxy Endpoint: /v1/oauth

Virtual host reference: https

Proxy Endpoint: /v1/customers

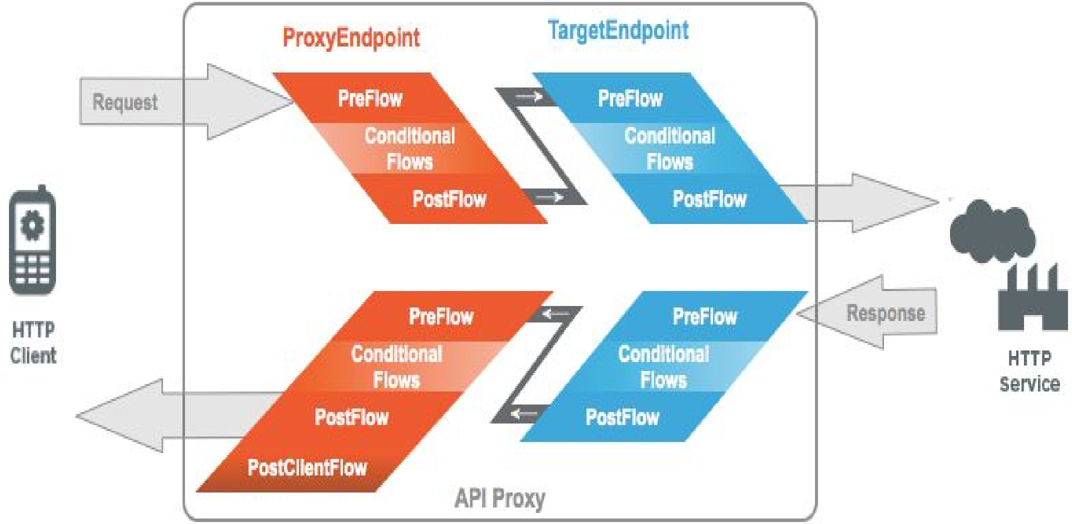
Virtual host reference: http

Virtual host reference: https

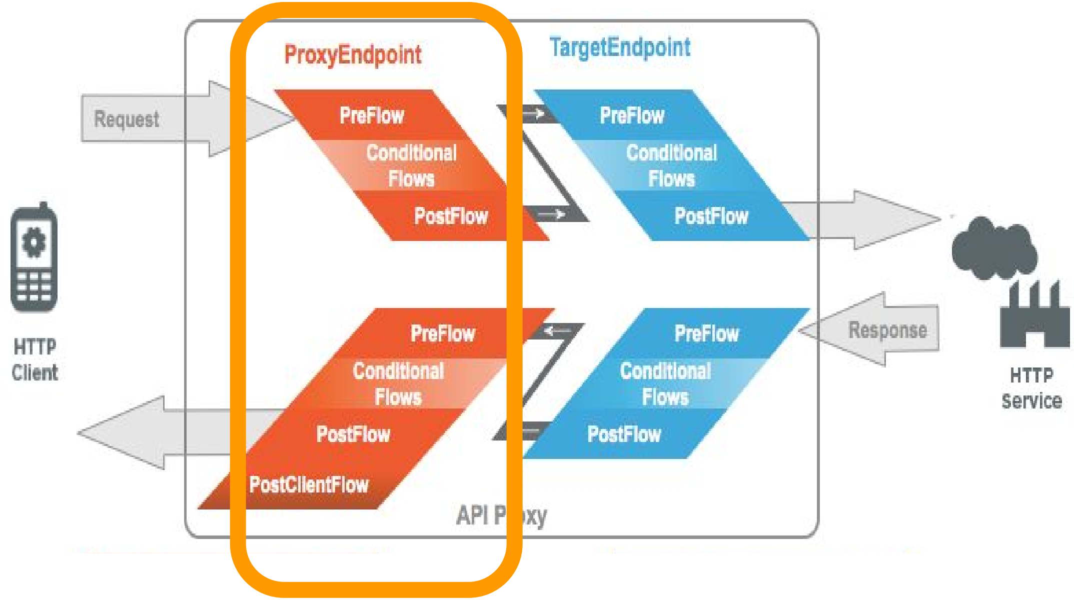
```
<HTTPProxyConnection>
   <BasePath>/v1/oauth</BasePath>
   <VirtualHost>https</VirtualHost>
</HTTPProxyConnection>
```

```
<HTTPProxyConnection>
   <BasePath>/v1/customers</BasePath>
   <VirtualHost>http</VirtualHost>
   <VirtualHost>https</VirtualHost>
</HTTPProxyConnection>
```

Policy Flows

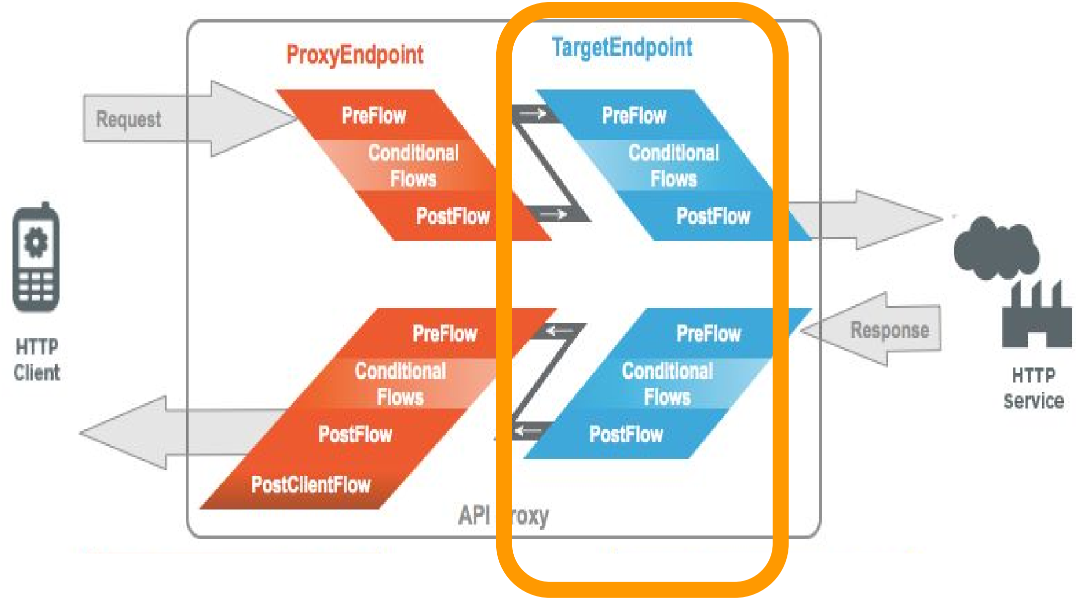


Policy Flows



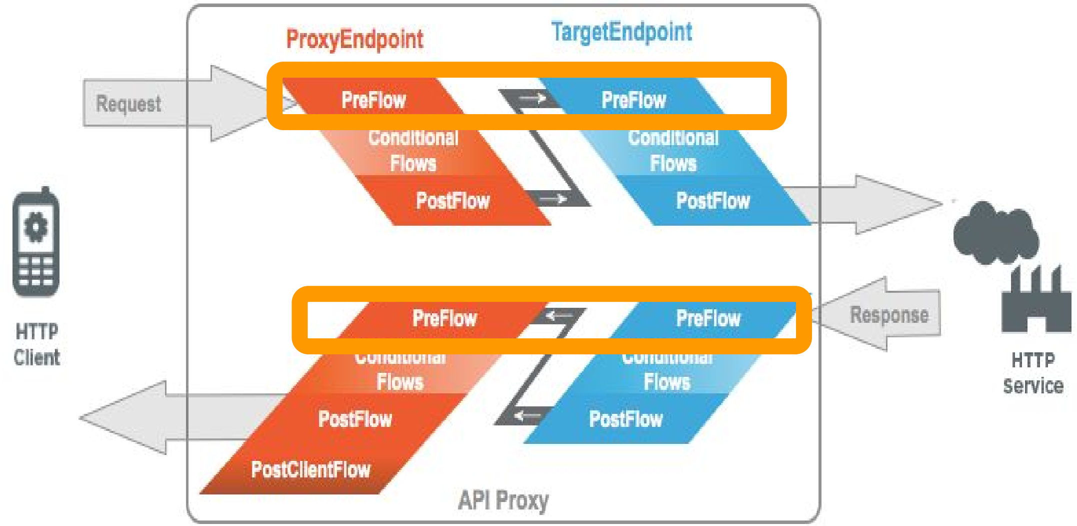
apigee

Policy Flows

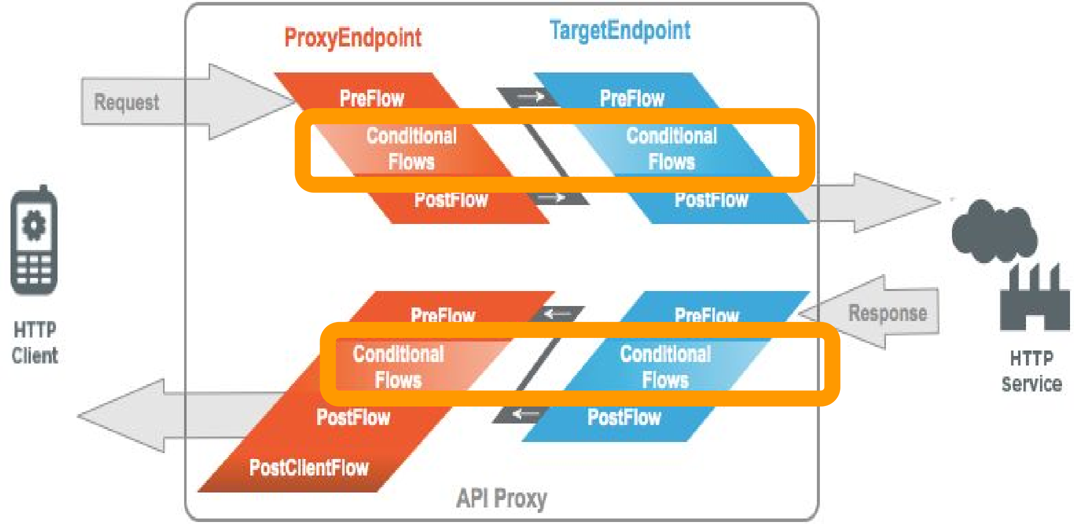


apigee

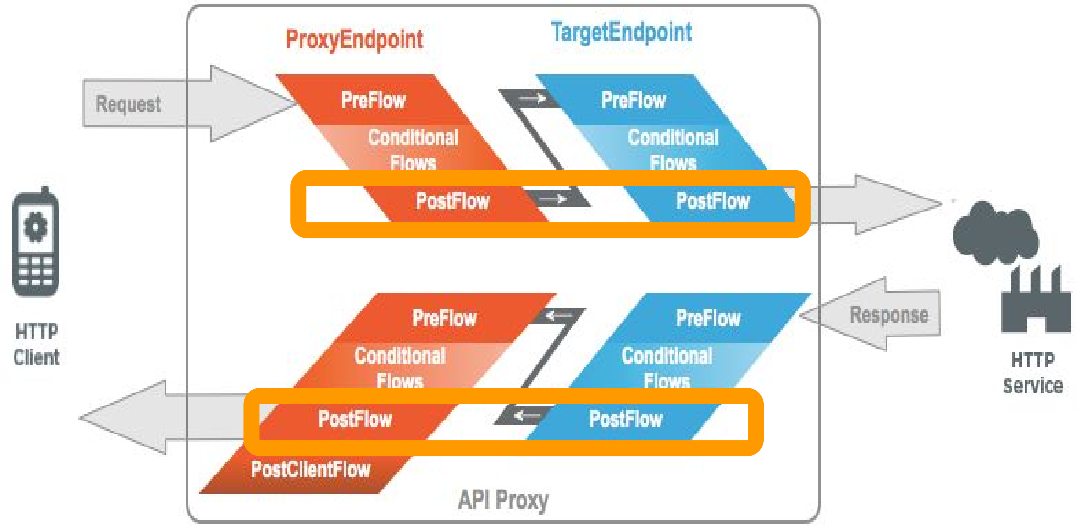
Policy Flows



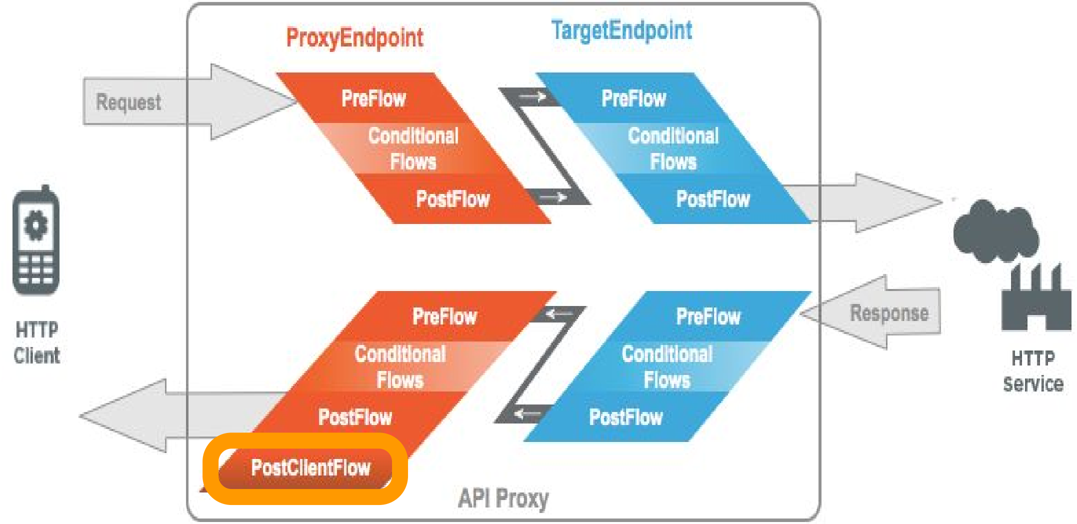
Policy Flows



Policy Flows



Policy Flows



Policy Flows

Global Flows

```
<PreFlow name="PreFlow">
    <Request>
        <Step>
            <Name>Verify-API-Key</Name>
        </Step>
    </Request>
    <Response/>
</PreFlow>
<PostFlow name="PostFlow">
    <Request/>
    <Response>
     <Step>
           <Name>Custom-Analytics</Name>
     </Step>
    </Response>
</PostFlow>
```

Policy Flows

Conditional Flow

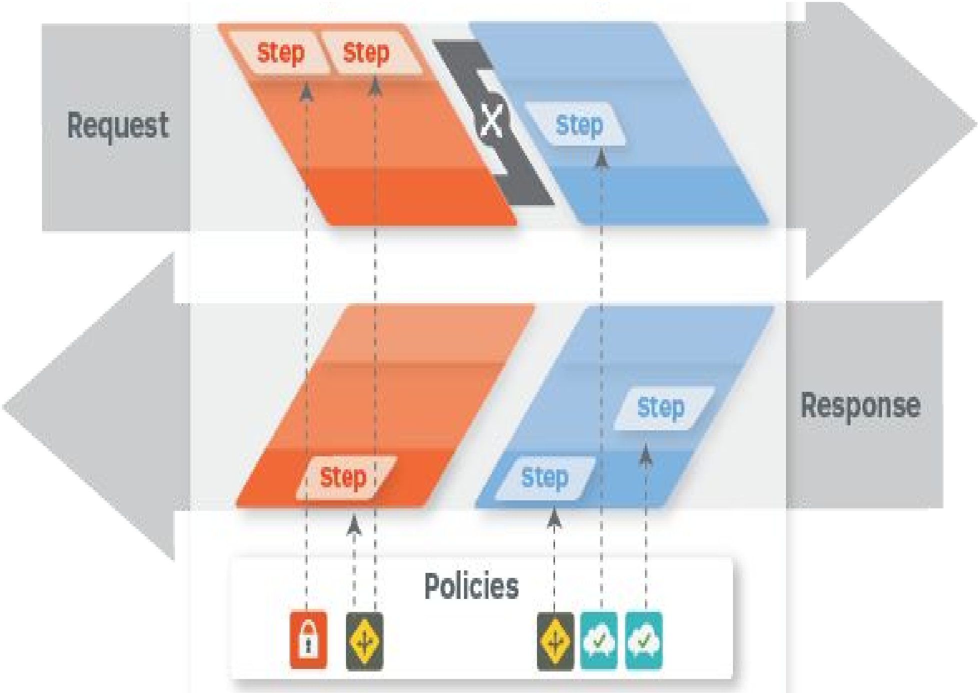
```
<Flows>
  <Flow name="GetResource">
  <Description>Get Resource</Description>
  <Request>
    <Step>
      <Name>Quota</Name>
    </Step>
  </Request>
  <Response/>
  <Condition>(proxy.pathsuffix MatchesPath 
"resource") and (request.verb="GET")</Condition>
  </Flow>
  ....
</Flows>
```

Policy Flows

PostClient Flow

```
<PostClientFlow>
  <Response>
    <Step>
      <Name>ML-Send-Audit</Name>
    </Step>
  </Response>
</PostClientFlow>
```

Policies



Route Rules

URL

```
<RouteRule name="default">
  <URL>http://api.mycompany.com/myAPI</URL>
</RouteRule>
```

Single Target

```
<RouteRule name="default">
  <TargetEndpoint>default</TargetEndpoint>
</RouteRule>
```

Conditional Targets

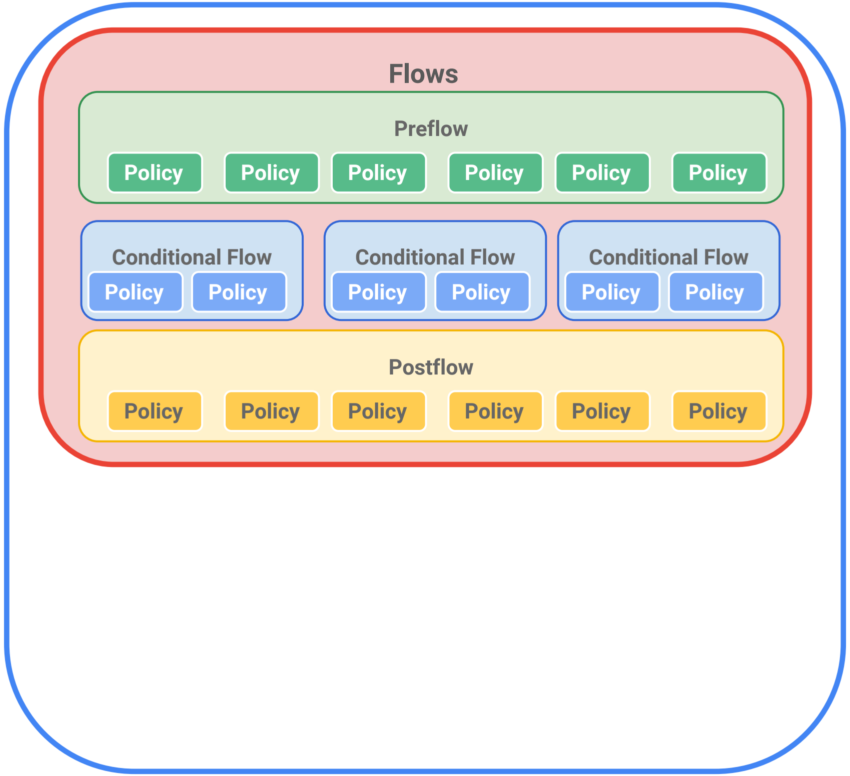
```
<RouteRule name="myRoute">
<Condition>request.header.mock="true"</Condition>
  <TargetEndpoint>mock</TargetEndpoint>
</RouteRule>

<RouteRule name="default">
 <TargetEndpoint>default</TargetEndpoint>
</RouteRule>
```

Null Targets

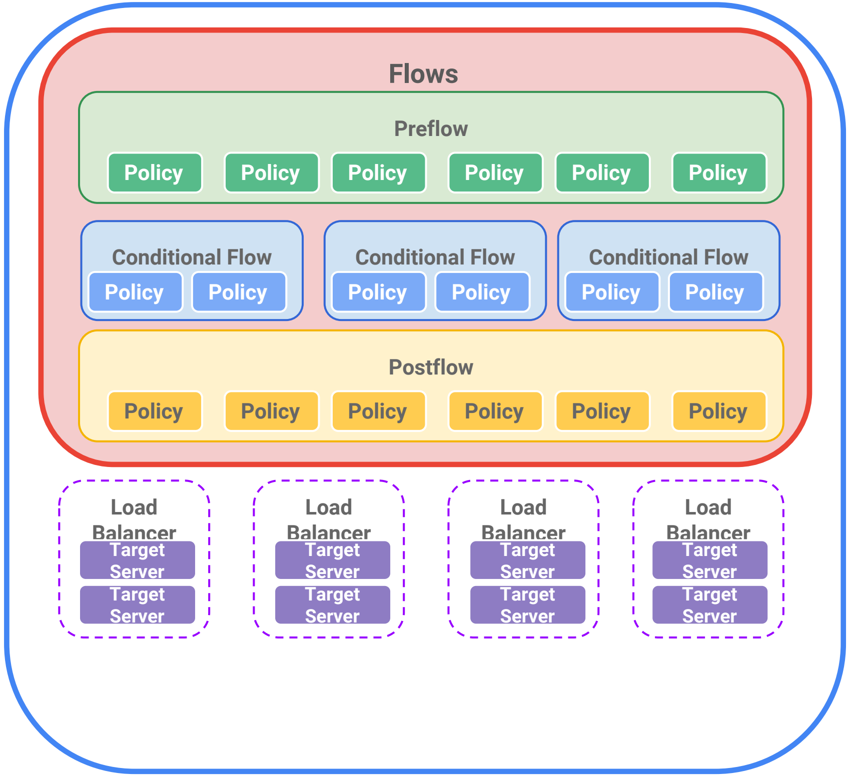
```
<RouteRule name="GoNoWhere"/>
```

Target Endpoint

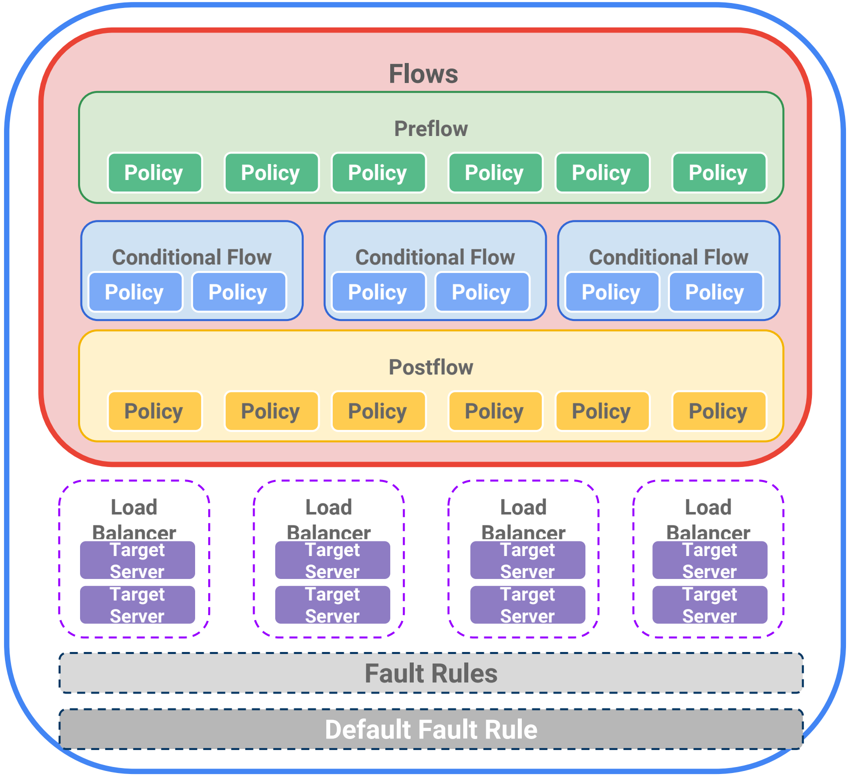
Target Endpoint



Target Endpoint

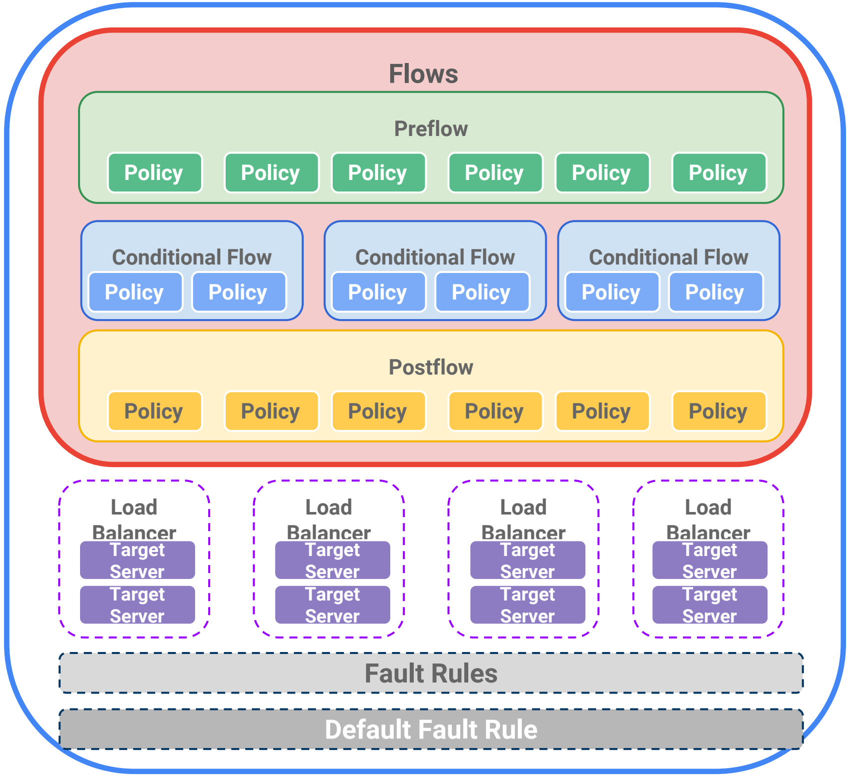


Target Endpoint



apigee

Target Endpoint



apigee

Target Endpoint Example

```
<TargetEndpoint name="default">
    <Description/>
    <FaultRules>
        <FaultRule name="catch_all">
            <Step>
                <Name>RF_GenericFault</Name>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow">
      <Request>
          <Step><Name>JS-set-target-url</Name></Step>
      </Request>
      <Response/>
    </PreFlow>
    <PostFlow name="PostFlow"/>
    <Flows/>
    <HTTPTargetConnection>
        <LoadBalancer>
          <Server name="target1" />
          <Server name="target2" />
        </LoadBalancer>
        <Path>/test</Path>
    </HTTPTargetConnection>
</TargetEndpoint>
```

Target Endpoint Example

```
<TargetEndpoint name="default">
    <Description/>
    <FaultRules>
        <FaultRule name="catch_all">
            <Step>
                <Name>RF_GenericFault</Name>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow">
       <Request>
          <Step><Name>JS-set-target-url</Name></Step>
       </Request>
       <Response/>
    </PreFlow>
    <PostFlow name="PostFlow"/>
    <Flows/>
    <HTTPTargetConnection>
        <LoadBalancer>
          <Server name="target1" />
          <Server name="target2" />
        </LoadBalancer>
        <Path>/test</Path>
    </HTTPTargetConnection>
</TargetEndpoint>
```

Fault Rules

Target Endpoint Example

```
<TargetEndpoint name="default">
    <Description/>
    <FaultRules>
        <FaultRule name="catch_all">
            <Step>
                <Name>RF_GenericFault</Name>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow">
      <Request>
          <Step><Name>JS-set-target-url</Name></Step>
      </Request>
      <Response/>
    </PreFlow>
    <PostFlow name="PostFlow"/>
    <Flows/>
    <HTTPTargetConnection>
        <LoadBalancer>
          <Server name="target1" />
          <Server name="target2" />
       </LoadBalancer>
       <Path>/test</Path>
    </HTTPTargetConnection>
</TargetEndpoint>
```

Fault Rules

Flows

Target Endpoint Example

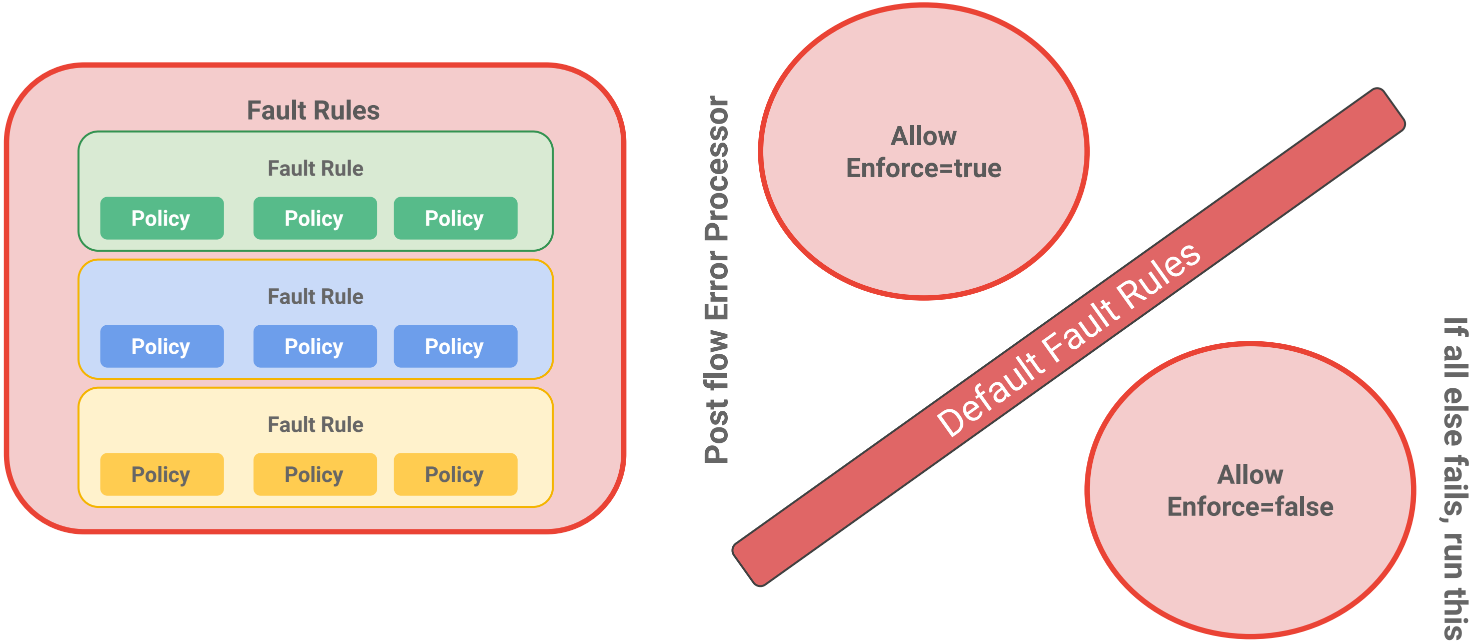
```
<TargetEndpoint name="default">
    <Description/>
    <FaultRules>
        <FaultRule name="catch_all">
            <Step>
                <Name>RF_GenericFault</Name>
            </Step>
        </FaultRule>
    </FaultRules>
    <PreFlow name="PreFlow">
       <Request>
          <Step><Name>JS-set-target-url</Name></Step>
       </Request>
       <Response/>
    </PreFlow>
    <PostFlow name="PostFlow"/>
    <Flows/>
    <HTTPTargetConnection>
        <LoadBalancer>
          <Server name="target1" />
          <Server name="target2" />
        </LoadBalancer>
        <Path>/test</Path>
    </HTTPTargetConnection>
</TargetEndpoint>
```

Fault Rules

Flows

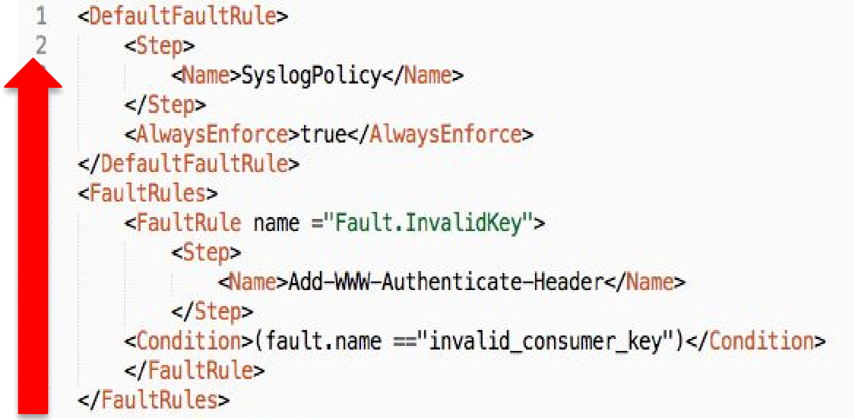
Load Balancer

Fault Rules


Fault Rules
Fault Rule
Policy
Policy
Policy
Fault Rule
Policy
Policy
Policy
Fault Rule
Policy
Policy
Policy
Post flow Error Processor
Allow
Enforce=true
Default Fault Rules
Allow
Enforce=false
If all else fails, run this

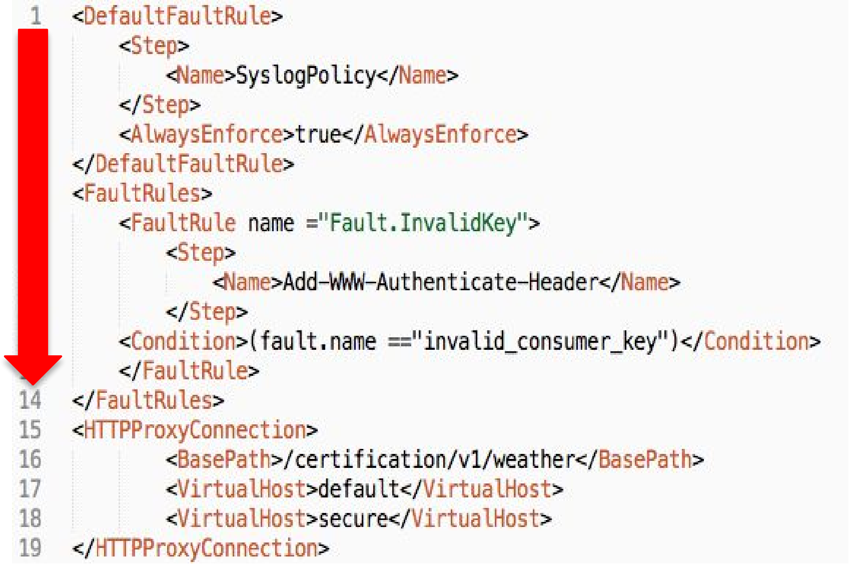
Fault Rules

Proxy Endpoint



```xml
<DefaultFaultRule>
    <Step>
        <Name>SyslogPolicy</Name>
    </Step>
    <AlwaysEnforce>true</AlwaysEnforce>
</DefaultFaultRule>
<FaultRules>
    <FaultRule name ="Fault.InvalidKey">
        <Step>
            <Name>Add-WWW-Authenticate-Header</Name>
        </Step>
    <Condition>(fault.name =="invalid_consumer_key")</Condition>
    </FaultRule>
</FaultRules>
```

Target Endpoint



```xml
<DefaultFaultRule>
    <Step>
        <Name>SyslogPolicy</Name>
    </Step>
    <AlwaysEnforce>true</AlwaysEnforce>
</DefaultFaultRule>
<FaultRules>
    <FaultRule name ="Fault.InvalidKey">
        <Step>
            <Name>Add-WWW-Authenticate-Header</Name>
        </Step>
    <Condition>(fault.name =="invalid_consumer_key")</Condition>
    </FaultRule>
</FaultRules>
<HTTPProxyConnection>
        <BasePath>/certification/v1/weather</BasePath>
        <VirtualHost>default</VirtualHost>
        <VirtualHost>secure</VirtualHost>
</HTTPProxyConnection>
```

Advanced Endpoint Properties

ProxyEndpoint <HTTPProxyConnection>

- **request.streaming.enabled -** request is streamed as-is
- **response.streaming.enabled -** response is streamed as-is

TargetEndpoint <HTTPTargetConnection>

- **connect.timeout.millis -** target connection timeout
- **io.timeout.millis -** timeout while reading request/response
- **request.streaming.enabled -** request is streamed as-is
- **response.streaming.enabled -** response is streamed as-is
- **success.codes -** explicit definition of success codes like 2xx, 3xx, 4xx

Thank You

Fault Rules

```
<DefaultFaultRule name="generic_fault_handler">
    <Step>
        <Name>RF_GenericFault</Name>
        <AlwaysEnforce>true</AlwaysEnforce>
    </Step>
</DefaultFaultRule>
<FaultRules>
    <FaultRule name="catch_all">
        <Step>
            <Name>RF_GenericFault</Name>
        </Step>
    </FaultRule>
    <FaultRule name="missing_key_rule">
        <Step>
            <Name>RF_MissingKey</Name>
            <Condition>fault.name="MissingApiKey"</Condition>
        </Step>
    </FaultRule>
    <FaultRule name="invalid_key_rule">
        <Step>
            <Name>RF_InvalidKey</Name>
            <Condition>fault.name="InvalidApiKey"</Condition>
        </Step>
    </FaultRule>
</FaultRules>
```