# Determining Policies for Internal and External Consumption of APIs

**Reza Salehi**
MCSE(CLOUD PLATFORM AND INFRASTRUCTURE), MCT, MCPD

@zaalion linkedin.com/in/rezasalehi2008

## Overview

**Understanding APIM policies**

**Configuring a policy**

- Sections (Inbound, backend, outbound, on-error)
- Policy scope

**Policies are powerful! A few use cases**

**Demo: configuring a new APIM policy**

**Protecting your APIs using policies**

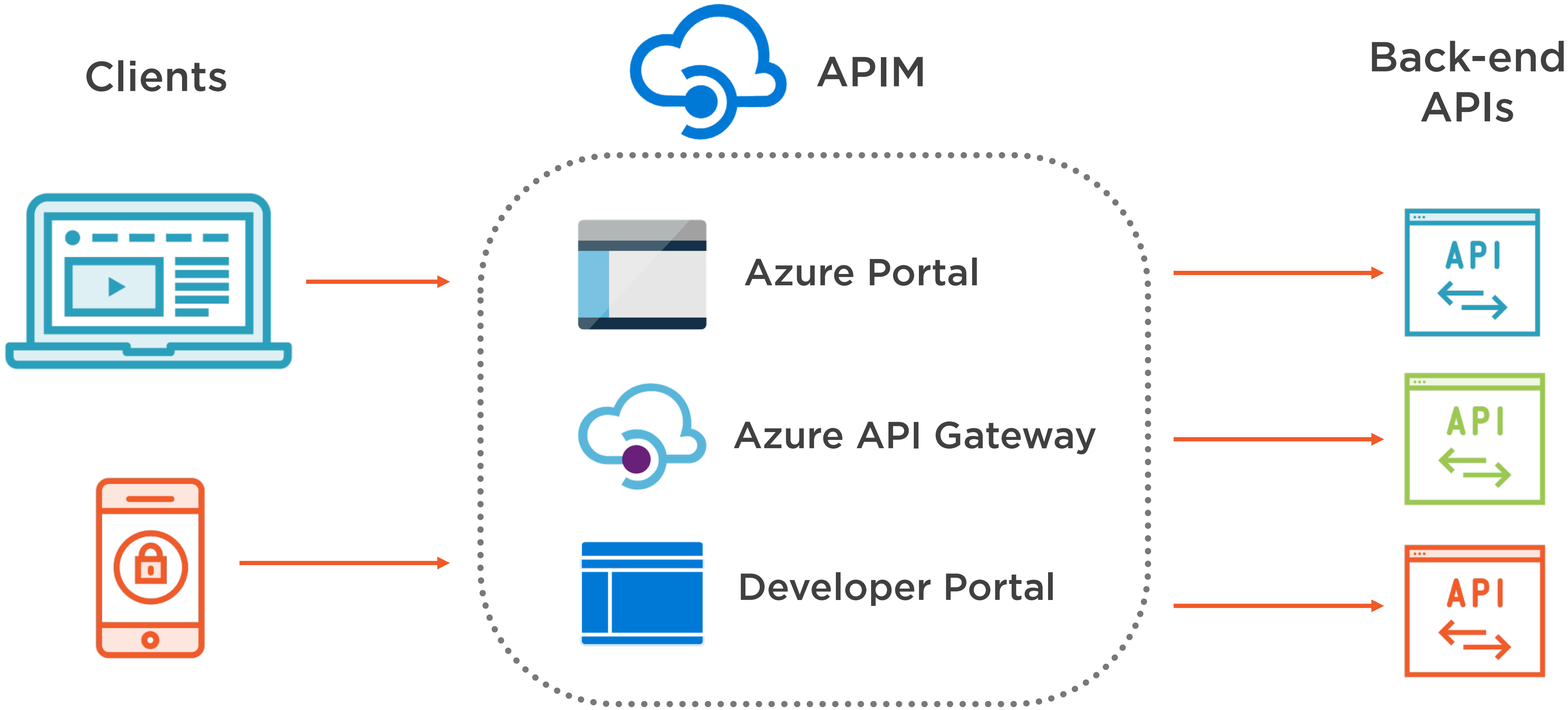
- Limit call rate, hide back-end API, prevent direct calls, filter caller IPs

**Demo: Protecting your APIs with policies**

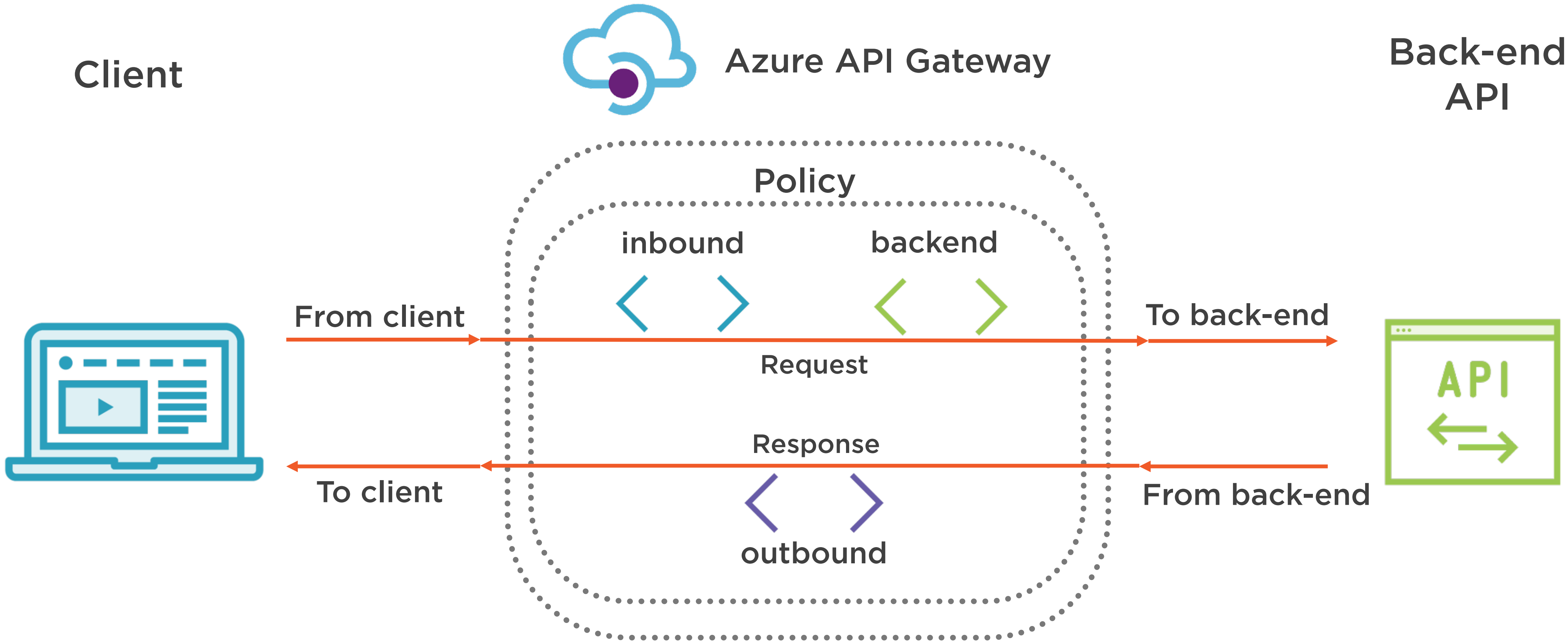
**Internal vs. external consumption**

## Configuring APIM Policies

# What Is an APIM Policy?



# What Is an APIM Policy?



Policies are a collection of Statements that are executed sequentially on the request or response of an API.

# Policy Sections

<inbound>

Statements to be applied to the request

<backend>

Statements to be applied before the request is forwarded to the back-end

<outbound>

Statements to be applied to the response

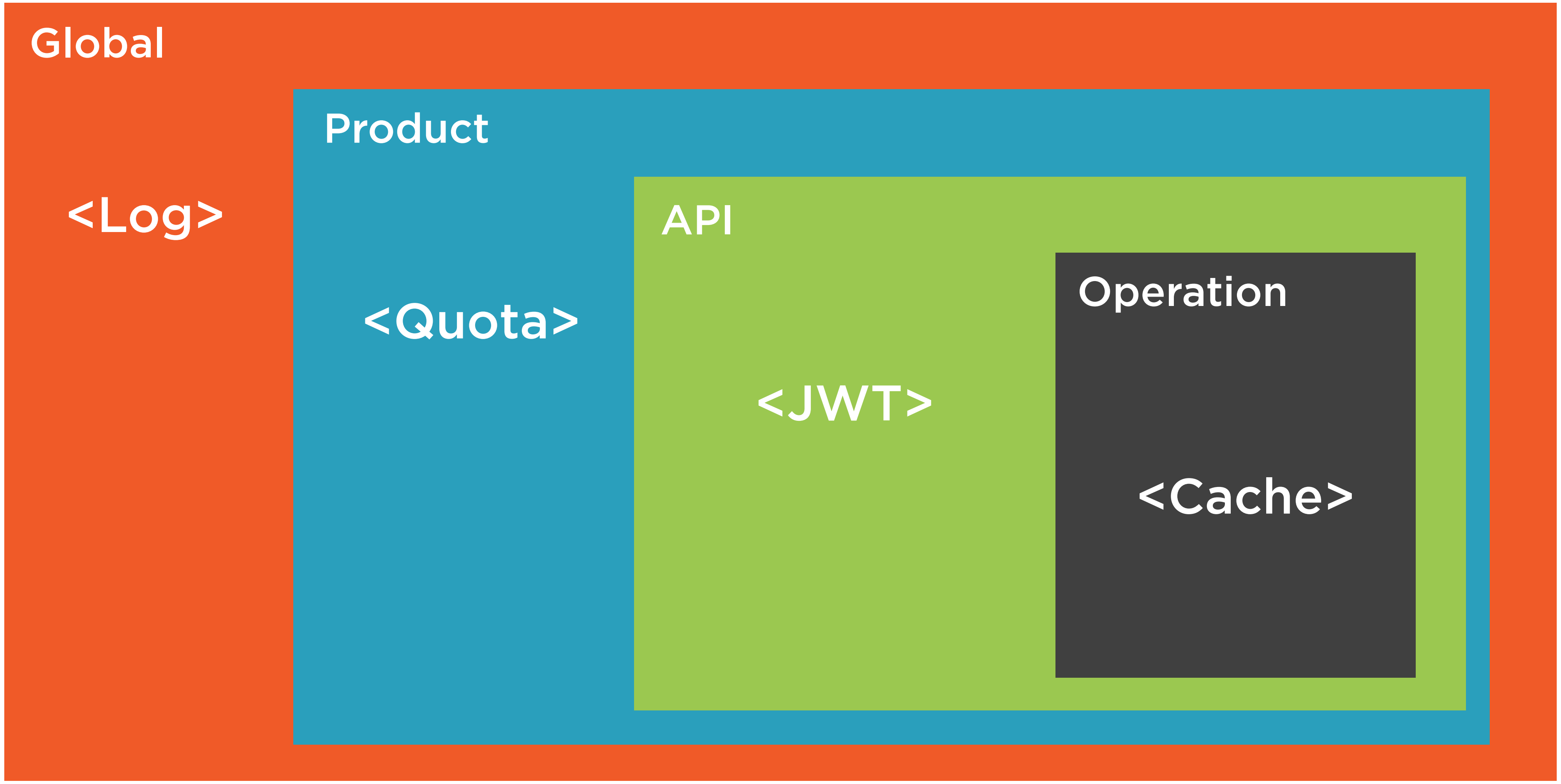
<on-error>

Statements to be applied if there is an error

# Policy Sections

```xml
<policies>
  <inbound>
    <!-- statements to be applied to the request go here -->
  </inbound>
  <backend>
    <!-- statements to be applied before the request is forwarded to
         the backend service go here -->
  </backend>
  <outbound>
    <!-- statements to be applied to the response go here -->
  </outbound>
  <on-error>
    <!-- statements to be applied if there is an error condition go here -->
  </on-error>
</policies>
```

# Policy Scope



# Policies Are Powerful! A Few Use Cases

**check-header**

Enforces existence and/or value of a HTTP header

**ip-filter**

Filters (allows/denies) calls from specific IP addresses and/or ranges

**return-response**

Aborts pipeline execution and returns the specified response to the caller

**Caching policies**

Enable response caching

**json-to-xml**

Converts request or response body from JSON to XML

## Where Do Policies Live?

Policies are configured in APIM > “Azure Portal”

Policies are executed in APIM > “Azure API “Gateway”

## Demo

Reviewing Azure Portal policy editor

Configuring caching and JSON to XML policies for the APIM instance

Examine different policy scopes

Test the configured policies

Reviewing the list of available policies

## Protecting APIs Using Policies

# Protecting APIs Using Policies

Limit client calls to specific bandwidth quota

Whitelist/Blacklist caller IP address

Protect back-end APIs from direct calls

Limit call numbers per subscription

Hiding back-end API URL from the caller

```
<quota calls="number" bandwidth="kilobytes" renewal-period="seconds">

    <api name="API name" id="API id" calls="number" renewal-
        period="seconds">

        <operation name="operation name" id="operation id"
            calls="number" renewal-period="seconds" />

    </api>

</quota>
```

## Limit Client Calls to Specific Bandwidth Quota

**The <quota/> policy enforces a renewable or lifetime call volume and/or bandwidth quota, on a per subscription basis.**

```
<rate-limit calls="number" renewal-period="seconds ">

    <api name="API name" id="API id" calls="number" renewal-
        period="seconds">

        <operation name="operation name" id="operation id"
            calls="number" renewal-period="seconds" />

    </api>

</rate-limit>
```

## Limit Call Numbers per Subscription

**The <rate-limit/> policy prevents API usage spikes on a per subscription basis by limiting the call rate to a specified number per a specified time period.**

```
<ip-filter action="allow | forbid">
    <address>address</address>
    <address-range from="address" to="address" />
</ip-filter>
```

## Whitelist/Blacklist Caller IP Address

**The <ip-filter/> policy filters (allows/denies) calls from specific IP addresses and/or address ranges.**

```
<find-and-replace from=“find" to=“replace"/>
```

## Hiding Back-end API URL from the Caller

**Using <find-and-replace/> policy, we can scan the response for original back-end API URL and replace it with the APIM instance URL.**

```
<set-header exists-action="override" name="key">
   <value>security token/key</value>
</set-header>
```

## Protect Back-end APIs from Direct Calls

**The <set-header/> policy can be used to inject a new security header into the request before passing it to the back-end.**
**The back-end then can only allow requests with this header present.**

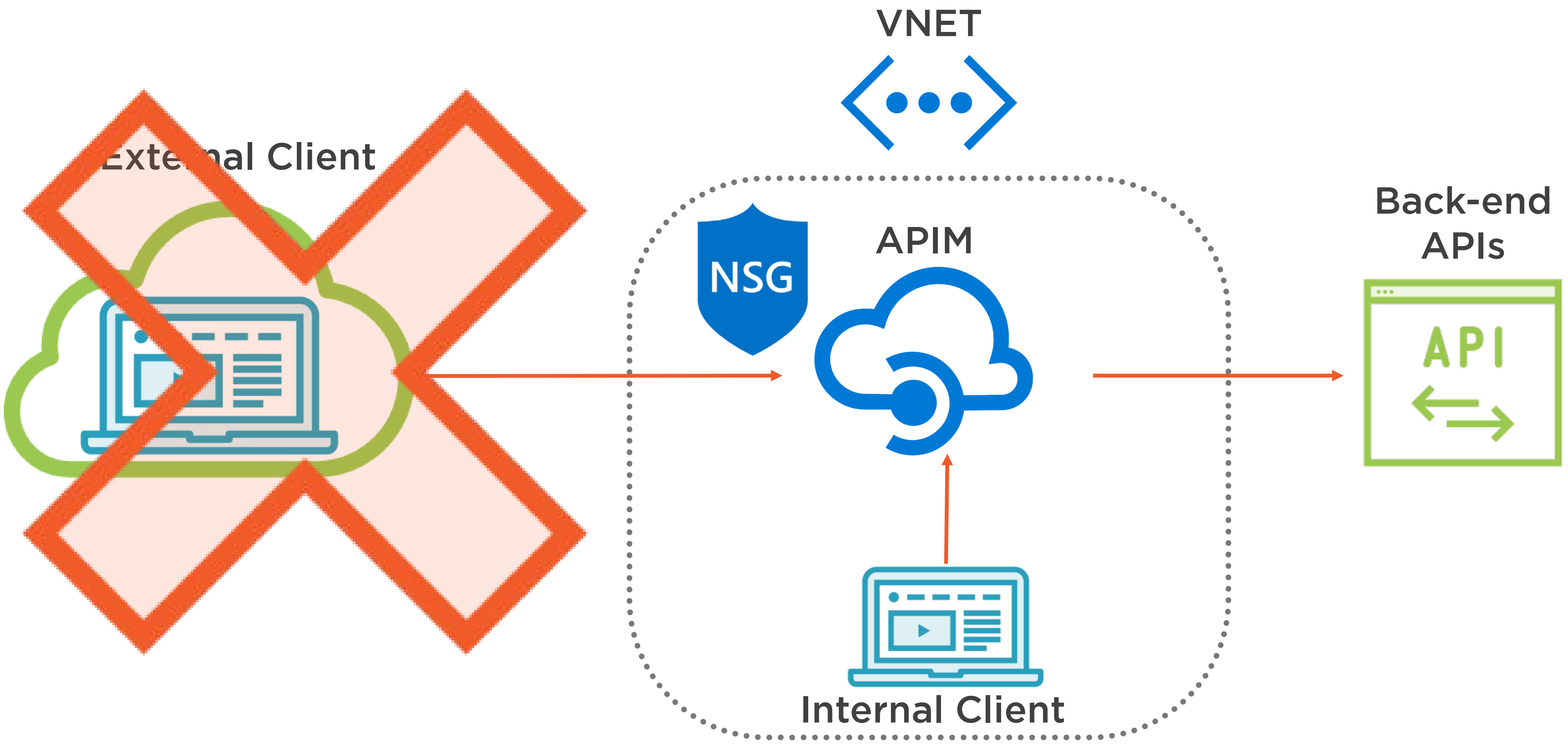
Demo

## Protecting your APIM instance using policies

- Setting a bandwidth quota
- Limiting call numbers
- Whitelist/Blacklist caller based on IP
- Hiding back-end API URL from the caller
- Protect back-end APIs from direct calls

## Internal vs. External Consumption

# Internal vs. External Consumption



Demo

Configuring the APIM instance to allow internal callers only

## Summary

Introduced APIM policies and explored a few use cases

Policy sections

Policy scope

Demo: configuring a new APIM policy

Protecting your back-end APIs using policies

Demo: Protecting your APIs with policies

Internal vs. External consumption