



Grand Tour of Azure API Management

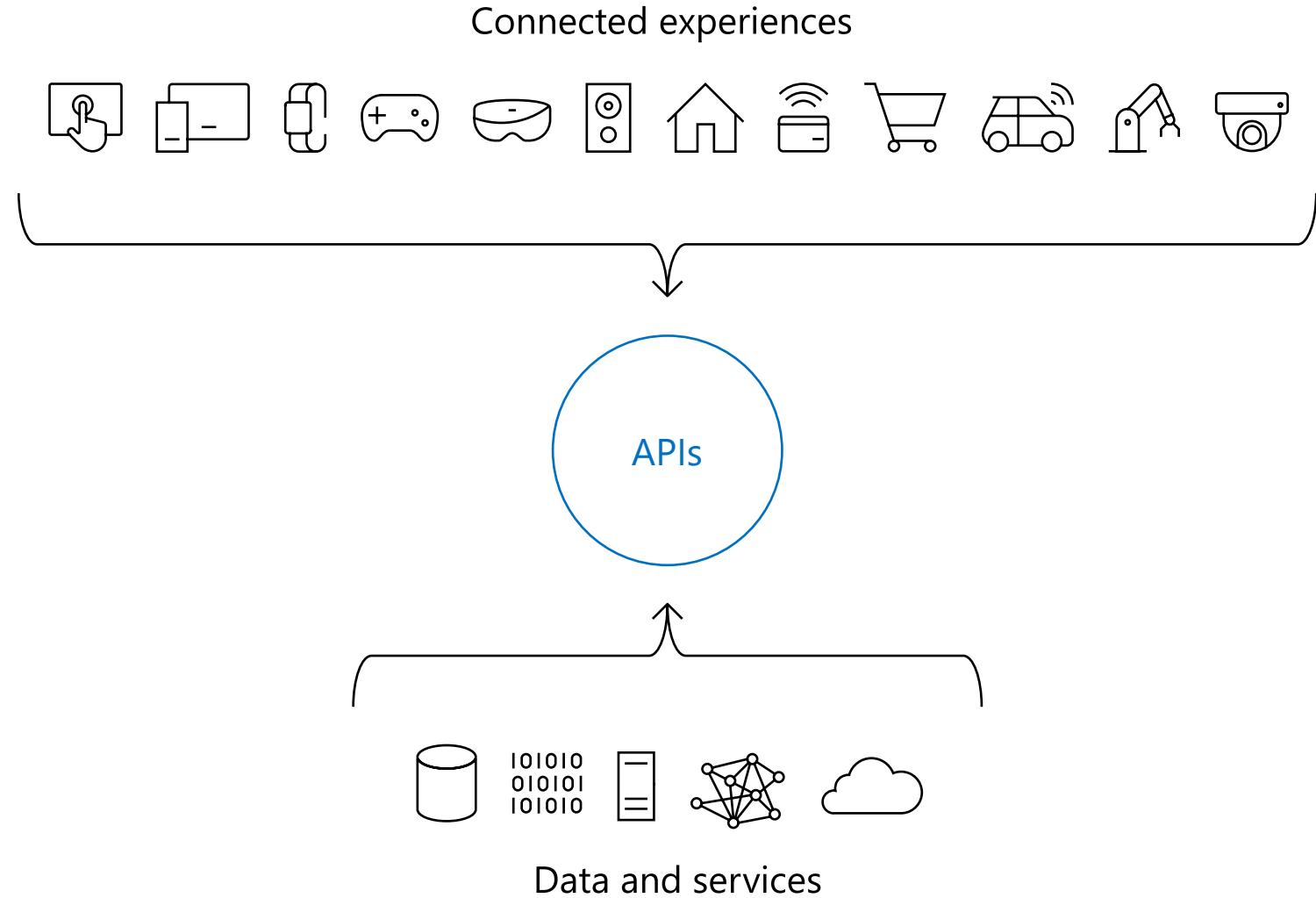


Agenda

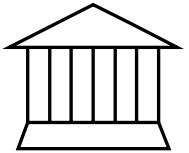
Azure API Management overview
In-depth look at the API life cycle phases

Design → Develop → Secure → Publish → Scale → Monitor → Analyze

Digital transformation is built on APIs



API governance and usage defines success



Façade

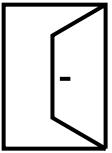
Abstraction

Aggregate or slice

Normalize or modernize

Decouple life cycle

Mock



Front door

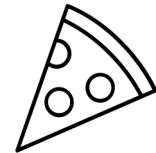
Control

Route and accelerate

Secure and protect

Transform

Observe



Frictionless consumption

Onboarding

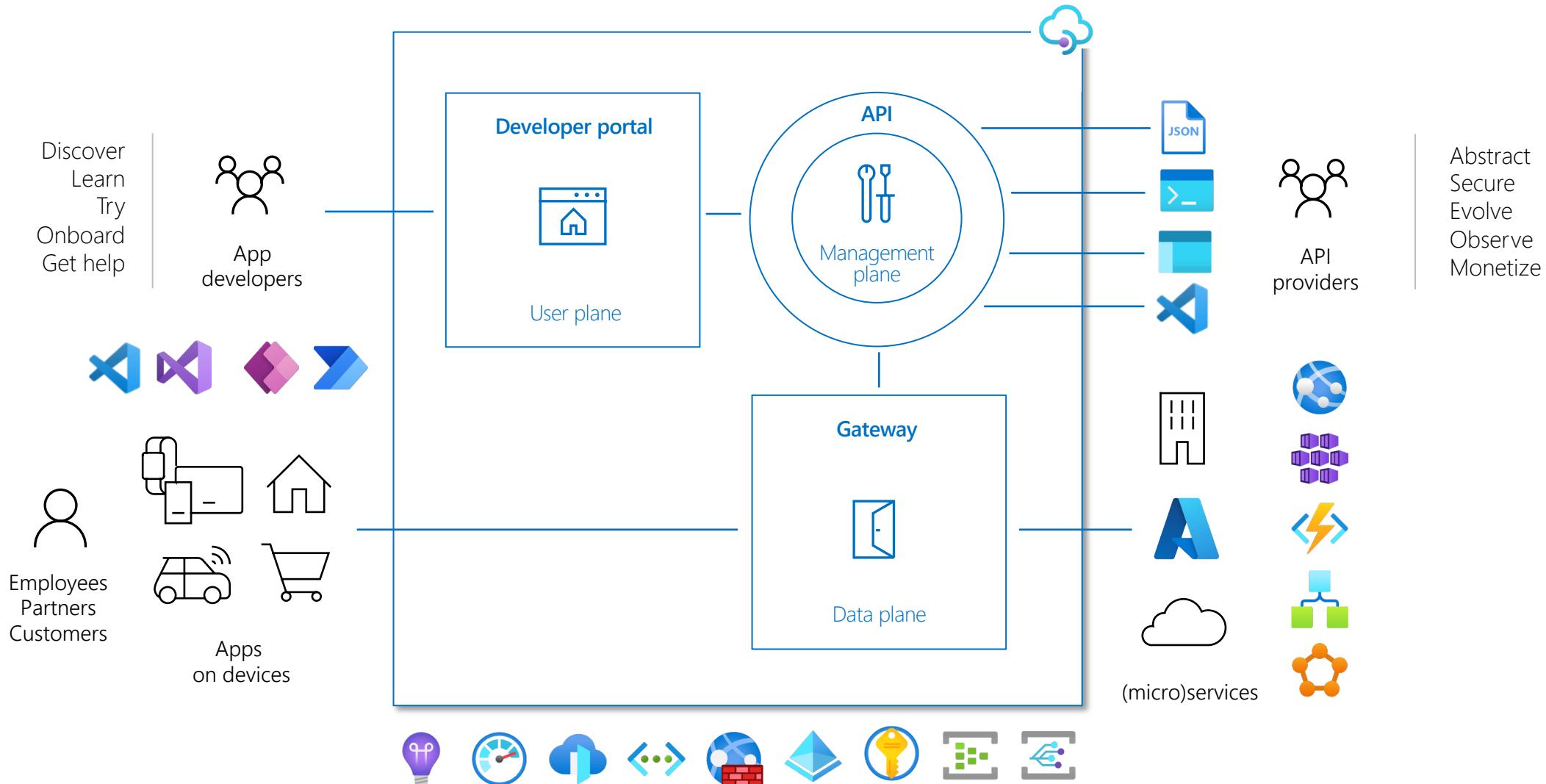
Discover and learn

Try

Obtain access

Get started

Azure API Management



Fully-managed serverless and dedicated tiers

Consumption tier

No infrastructure to provision or manage

Built-in auto-scaling down to zero

Consumption-based micro billing

Variable, usage-based monthly cost

No reserved capacity

Shared management plane

On-demand activation

Curated set of [features](#) and usage [limits](#)

Developer | Basic | Standard | Premium tier

No infrastructure to provision or manage

Manual scaling or external auto-scaling

Billing based on reserved capacity

Constant, predictable monthly cost

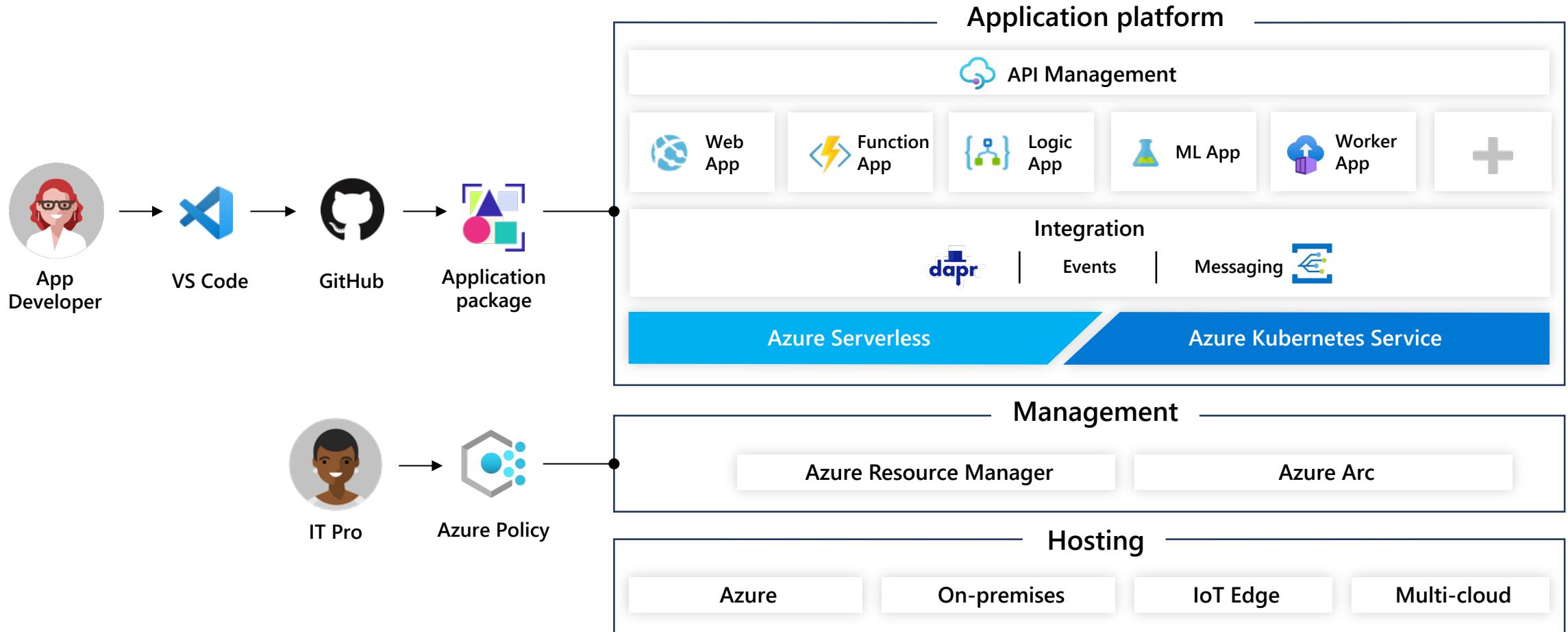
Reserved capacity

Dedicated management, user, and data planes

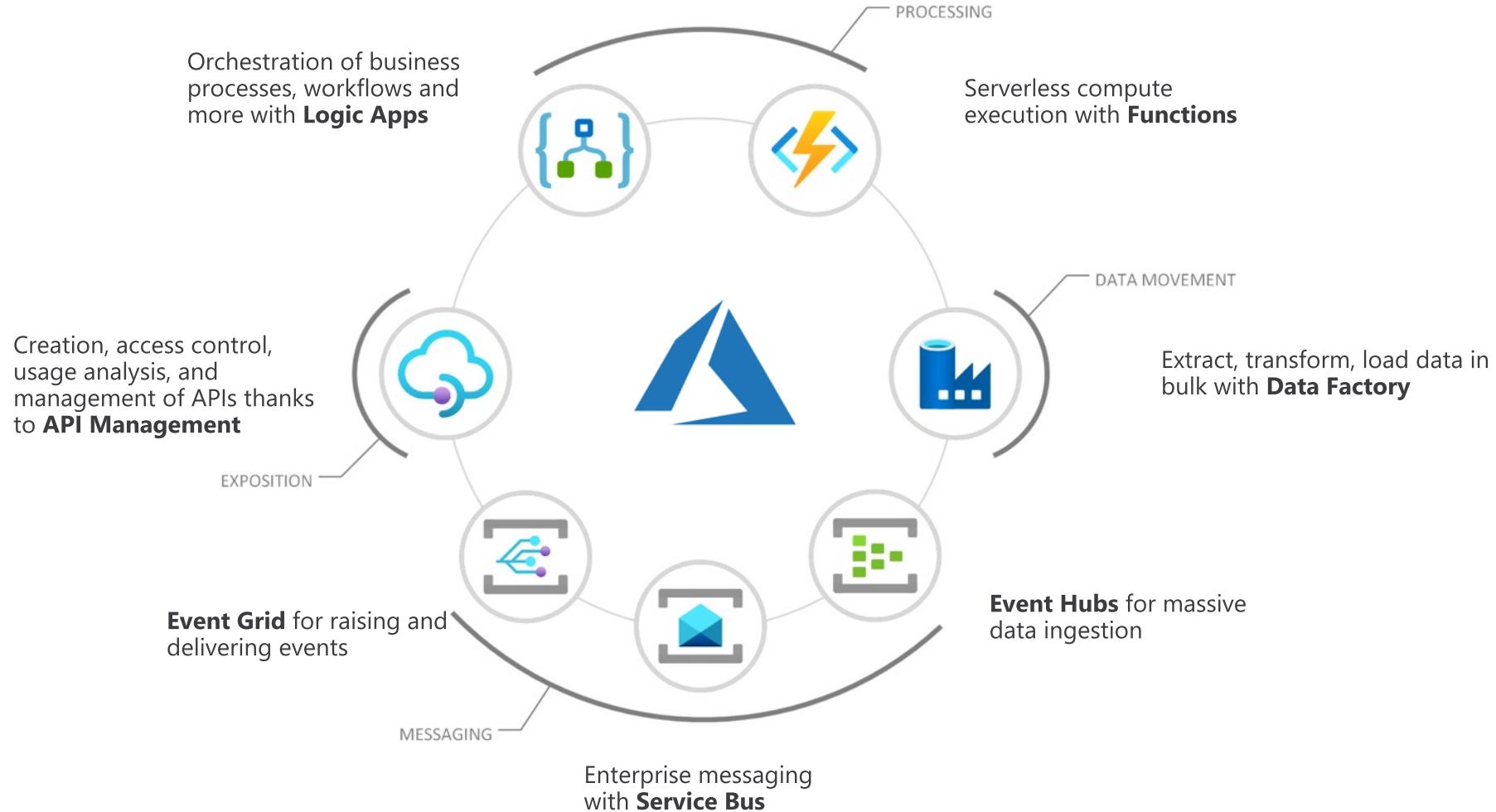
Always on

Full set of features. Not governed.

Azure Application Platform



Azure Integration Services – Enterprise iPaaS



[Gartner named Microsoft a leader in 2021 Gartner Magic Quadrant for Enterprise iPaaS](#)

API management is key in digital business ecosystem

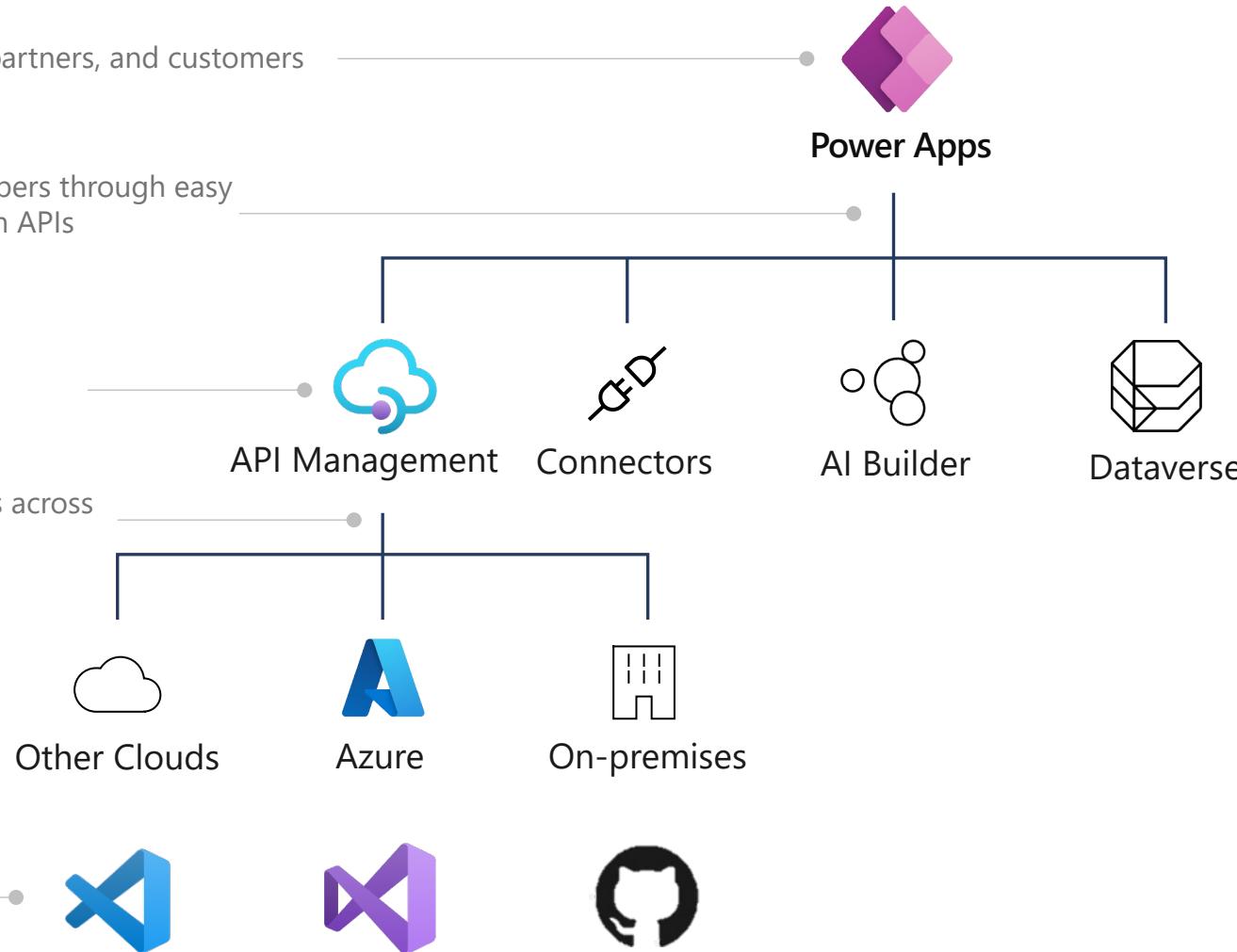
Marketplace of apps for employees, partners, and customers

Extend innovation with citizen developers through easy discovery and consumption of custom APIs

Enterprise marketplace for API-based innovation

Unified management for custom APIs across clouds and on-premises

Integrated early on into the API development process



End users

More productive and satisfied employees, partners, and customers



Citizen developers

Able to accelerate app creation



IT pros

Able to govern all APIs and Apps



Professional developers

Able to amplify skills by maximizing API reuse

Value proposition

-  Mature full life cycle API management solution
-  Trusted by thousands of enterprise customers
-  Abstract, secure, observe, and make APIs discoverable in minutes
-  One solution for APIs across clouds and on-premises
-  Dependable, secure, scalable, and performant
-  DevOps and developer-friendly
-  Azure-native and integrated with other Azure services
-  Globally available and supported
-  Low-barrier-to-entry pricing

4.65T

API calls per annum
87% YoY growth

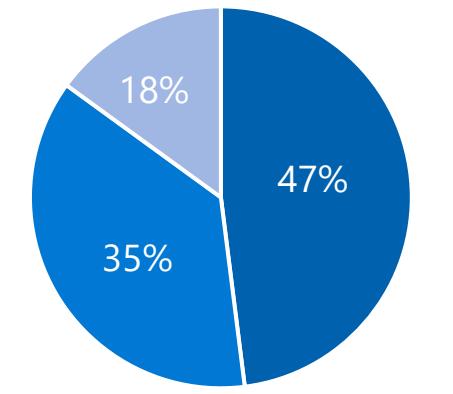
840K

APIs under management
72% YoY growth

18K

Customers
38% YoY growth

54 regions worldwide

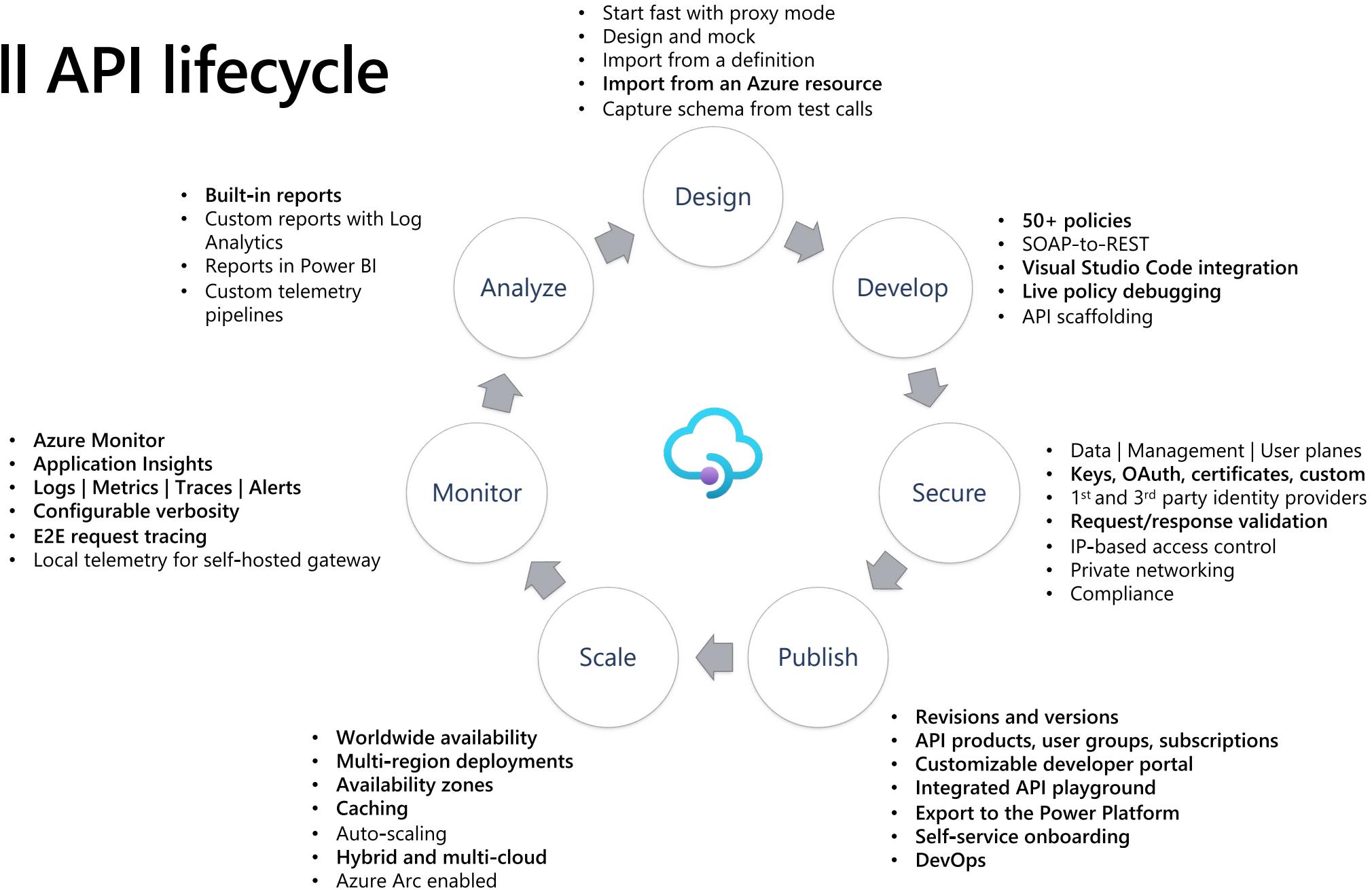


■ Americas ■ EMEA ■ APAC

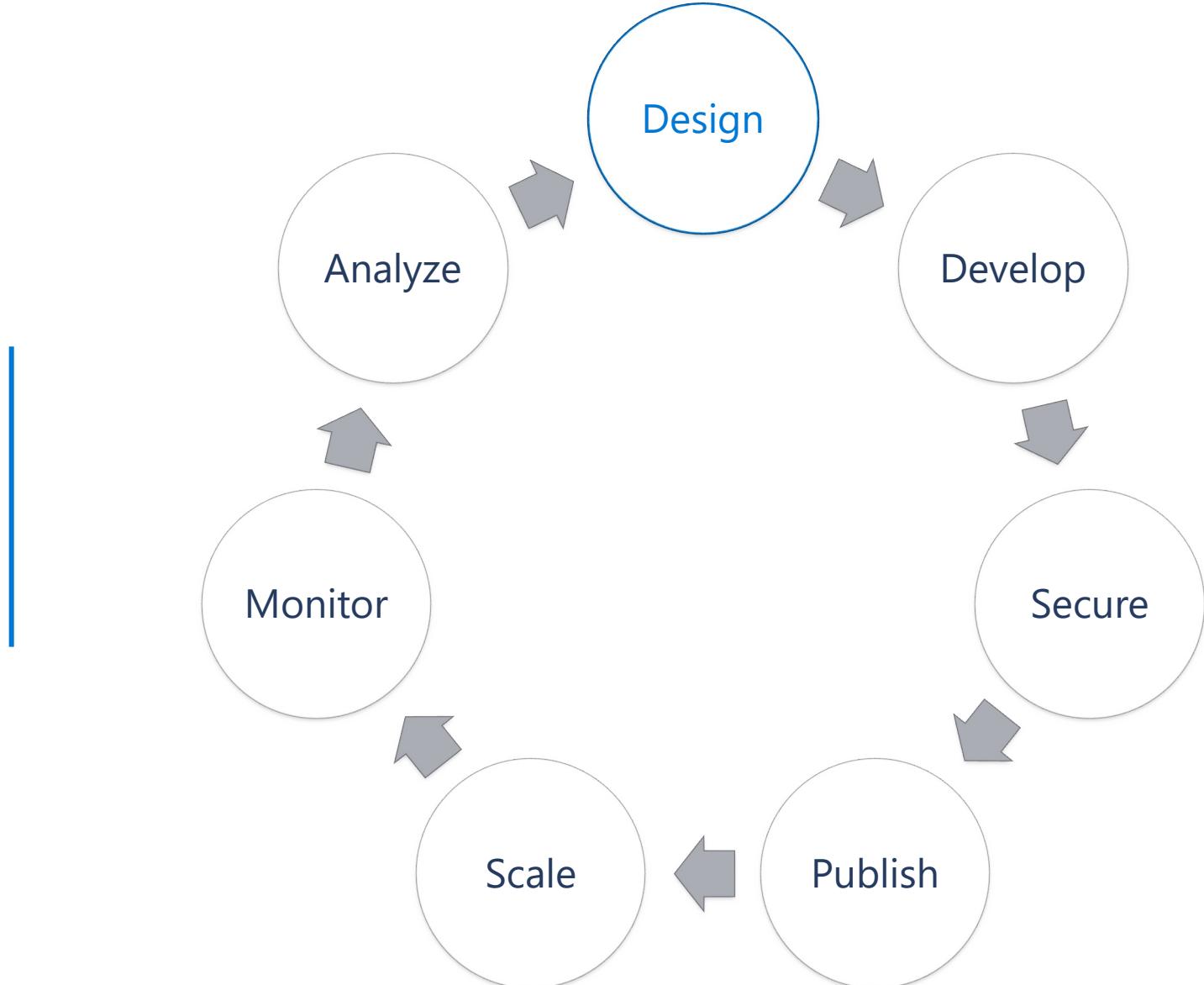
Azure API Management Customers



Full API lifecycle



API life cycle: design



Code- and design-first approaches to building APIs

API Management supports both approaches to building APIs:

Code-first approach

Implement the API and generate the API specification as an afterthought (i.e. with Swashbuckle)

Benefits:

- More convenient for API developers
- The only option for existing APIs

Design-first approach

Create an API specification, review it with stakeholders, and implement the API

Kickstart development by scaffolding the code from the API specification

Benefits:

- Better API consumer experience thanks to the deliberate API design
- Reduced risk thanks to the API review processes

Create an API

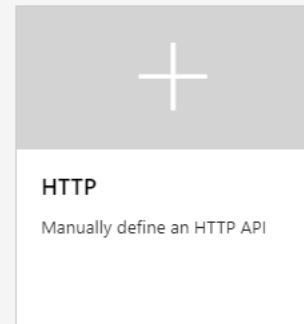
Support for SOAP, REST,
WebSocket and GraphQL APIs

Import an API from OpenAPI (1,
2, or 3), WADL, or WSDL files

Import an API from App Service,
Logic App, Function App, or
Container App

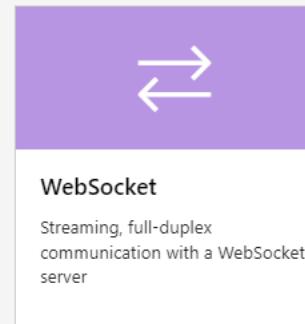
Create a blank API

Define a new API



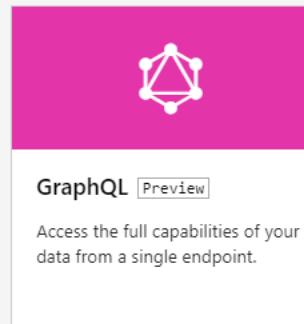
HTTP

Manually define an HTTP API



WebSocket

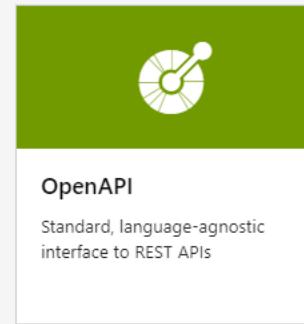
Streaming, full-duplex communication with a WebSocket server



GraphQL Preview

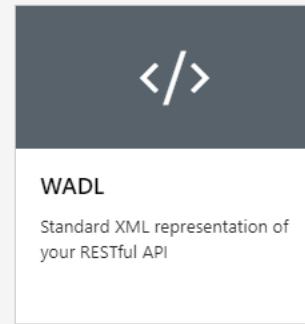
Access the full capabilities of your data from a single endpoint.

Create from definition



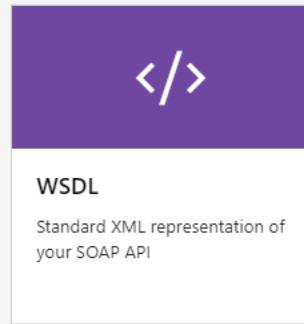
OpenAPI

Standard, language-agnostic interface to REST APIs



WADL

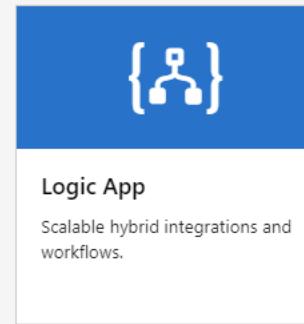
Standard XML representation of your RESTful API



WSDL

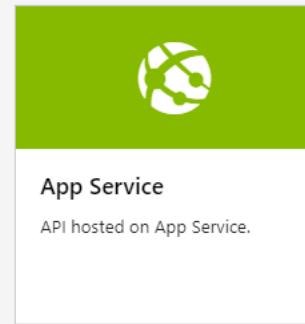
Standard XML representation of your SOAP API

Create from Azure resource



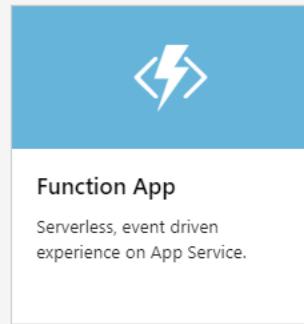
Logic App

Scalable hybrid integrations and workflows.



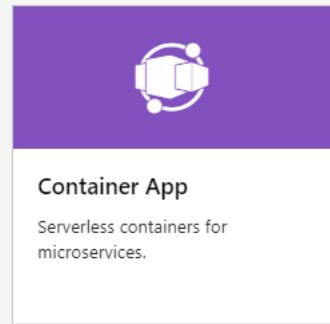
App Service

API hosted on App Service.



Function App

Serverless, event driven experience on App Service.



Container App

Serverless containers for microservices.

Code-first approach – use the wildcard proxy mode

Wildcard “*” proxy mode

Use to route all requests through API Management if an accurate API specification doesn't exist

Use built-in API design features to improve the specification

The screenshot shows the Microsoft Azure API Management service interface. The top navigation bar is blue with the Microsoft Azure logo and the text "Microsoft Azure". Below it, the breadcrumb navigation shows "Dashboard > fabrikam". The main area has a header "fabrikam | APIs" with a "Developer portal" link. On the left, there's a sidebar with "All APIs" (including "Demo Conference API" and "Wildcard API") and a "+ Add API" button. The "Wildcard API" is selected and highlighted in light blue. The main content area has tabs "Design", "Settings", "Test", "Revisions", and "Change log", with "Design" being the active tab. Under "Design", there's a "Frontend" section with fields for "Display name" (set to "Catch all"), "Name" (set to "catch-all"), and "URL" (set to "GET /*"). There's also a "Description" field which is currently empty.

Design the API

Define the API with form-based or text-based editors in the Azure portal or the Visual Studio Code extension

Test the API in the Azure portal and generate schemas from the API responses

The screenshot shows the Microsoft Azure API Management service interface for the 'fabrikam' service. The top navigation bar includes 'Dashboard > fabrikam', the service logo 'fabrikam | APIs', and 'API Management service'. Below the navigation is a breadcrumb trail: 'Developer portal' and 'Developer portal (legacy)'. The main content area displays the 'Demo Conference API > OpenAPI specification' in JSON format. A context menu is open over the JSON code, specifically over the 'paths' section. The menu items include: [traceClick] = "Swagger editor > Insert" > Insert Operation - /sessions, [traceClick] = "Swagger editor > Insert" > Insert Response - /sessions - get, [traceClick] = "Swagger editor > Insert" > Insert Request Body - /sessions - get, Edit Information Object, Add Operation Object, Add Definition Object, Add Tag Object, and [traceClick] = "Swagger editor > Insert" > Insert Request Body - /sessions - post. The JSON code itself is a detailed definition of an API, including paths like '/sessions', operations like 'get', and security definitions involving 'apiKeyHeader' and 'apiKeyQuery'.

```
1 {  
2   "swagger": "2.0",  
3   "info": {  
4     "title": "Fabrikam Conference API",  
5     "version": "1.0",  
6     "description": "API for managing conference sessions."  
7   },  
8   "host": "fabricademosession.azurewebsites.net",  
9   "schemes": ["https"],  
10  "securityDefinitions": {  
11    "subscriptionKey": {  
12      "type": "apiKey",  
13      "name": "Subscription-Key",  
14      "in": "header"  
15    },  
16    "apiKeyQuery": {  
17      "type": "apiKey",  
18      "name": "subscription-key",  
19      "in": "query"  
20    }  
21  },  
22  "security": [{  
23    "apiKeyHeader": []  
24  }, {  
25    "apiKeyQuery": []  
26  }],  
27  "paths": {  
28    "/sessions": {  
29      "get": {  
30        "description": "A list of sessions. Optional parameters work as filters to refine the results.",  
31        "operationId": "GetSessions",  
32        "summary": "GetSessions",  
33        "parameters": [{  
34          "name": "sneakername",  
35          "type": "string",  
36          "in": "query",  
37          "description": "Filter by sneaker name."  
38        }]  
39      }  
40    }  
41  }  
42}
```

At the bottom of the screen are two buttons: 'Save' and 'Discard'.

[»](#) Design Settings Test Revisions Change log[+ Add operation](#)

Demo Conference API > GetSpeakers > Frontend

[🔗 OpenAPI specification View](#)**All operations**

GET GetSession ...

* Display name
GetSpeakers

GET GetSessions ...

* Name
GetSpeakers

GET GetSessionTopics ...

* URL
GET /speakers

GET GetSpeaker ...

Description
Test test

GET GetSpeakers ...

Markdown

Tags
e.g. Booking

GET GetSpeakerSes...

[Template](#) [Query](#) [Headers](#) [Request](#) [Responses](#)

GET GetSpeakerTop...

Query parameters

Define additional query parameters.

NAME	DESCRIPTION	TYPE	VALUES	REQUIRED	
dayno	Format - int32.	integer		<input type="checkbox"/>	
speakername		string		<input type="checkbox"/>	
+ Add parameter					

GET GetTopicSpeak...

[Operations](#)[Definitions](#)[Save](#)[Discard](#)

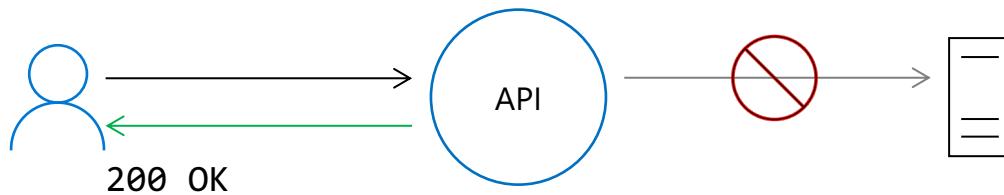
Design-first approach – mock the API

Unblock front-end teams by mocking API responses

Use an example defined in the API definition

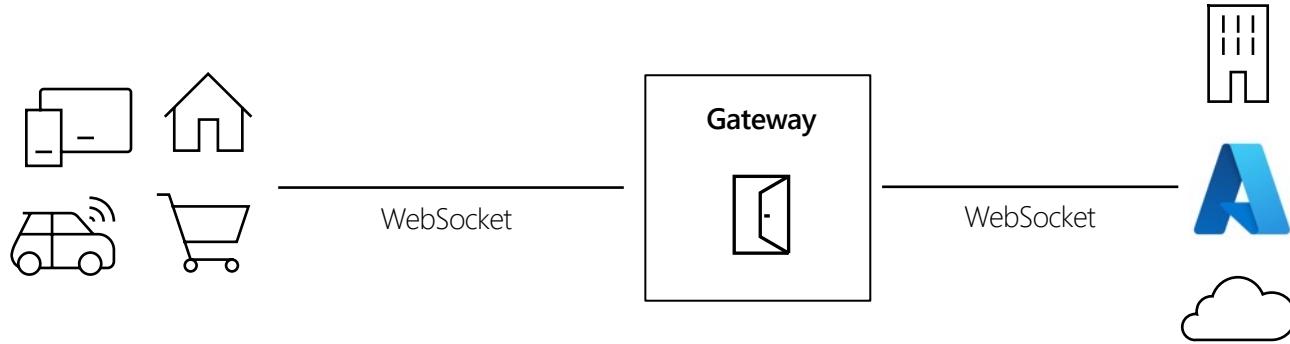
Configure with a single-line policy

```
<inbound>
  <base />
  <mock-response status-code="200" content-type="application/json" />
</inbound>
```



```
{
  "message": "example"
}
```

WebSocket API support



Passthrough support for WebSocket APIs

Client applications establish WebSocket connections with APIM

API Management establishes WebSocket connections with backend services

API Management proxies WebSocket messages

Features

- CRUD WebSocket APIs

- Apply policies to handshake requests

- Browse WebSocket APIs in the Developer portal

- Test WebSocket APIs in the Azure and Developer portals

- Azure Monitor metrics and logs

GraphQL API support (Public preview)

Passthrough support for GraphQL APIs

CRUD existing GraphQL APIs via Azure portal and management API

Explore the schema and run test queries in the Azure and developer portals

Apply existing access control policies

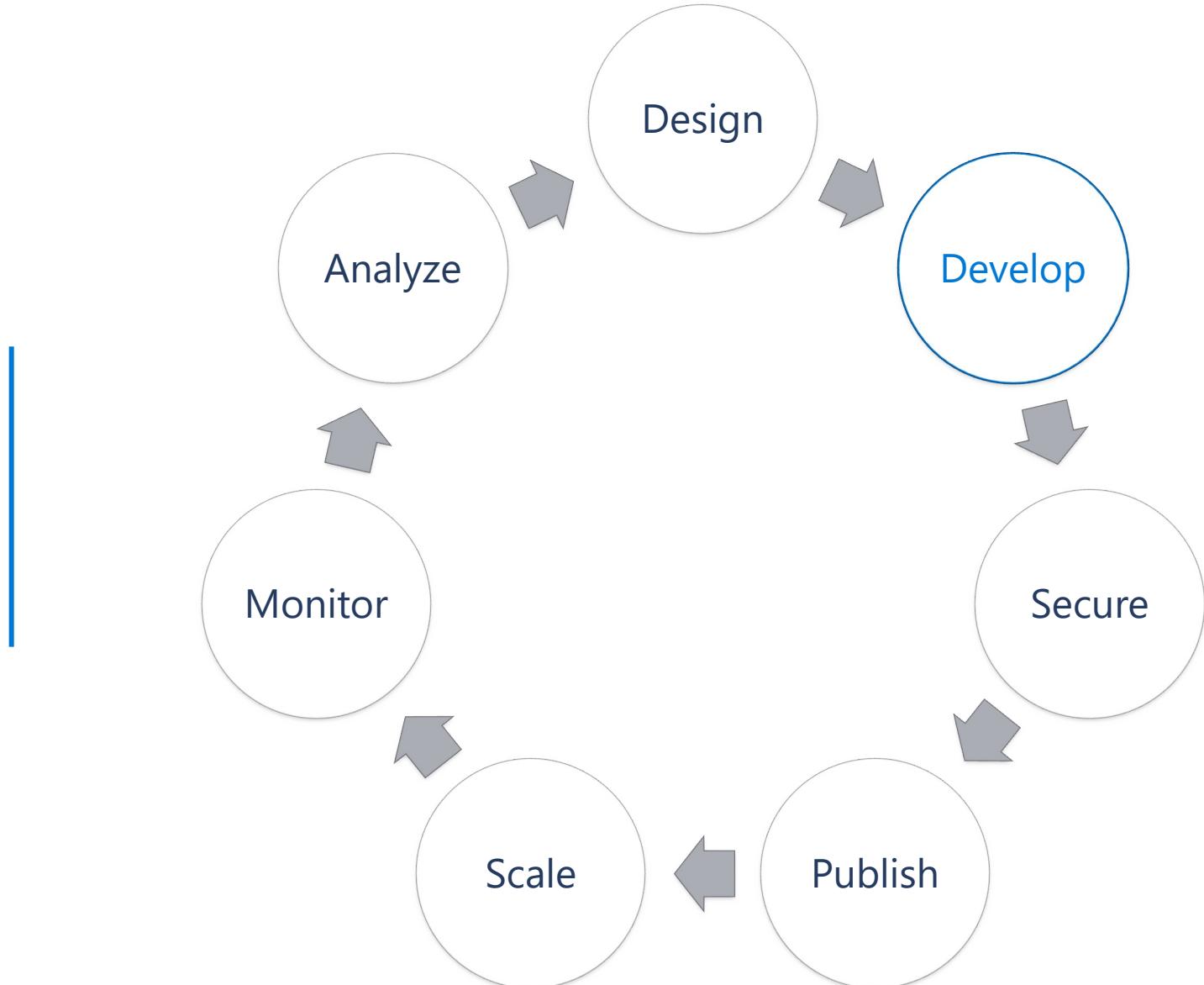
Apply a new 'validate-graphql-request' policy to protect against GraphQL-specific attacks

- Query validation

- Field-based authorization

- Query depth and size restriction

API life cycle: develop



Cross domain policies

+ Allow cross domain calls

+ CORS

+ JSONP

Authentication policies

+ Authenticate with Basic

+ Authenticate with client certificate

Access restriction policies

+ Check HTTP header

+ Limit call rate per key

+ Limit call rate per subscription

+ Restrict caller IPs

+ Set usage quota per key

+ Set usage quota per subscription

+ Validate JWT

There's a policy for that

Encapsulate common API management functions

Access control, Protection, Transformation, Caching, ...

Mutate request context or change API behavior

E.g. add a header or throttle

Set in the inbound and outbound directions

Apply at a variety of scopes or on error

Scope determines which APIs are affected

Can define custom scopes in addition to four available by default

Compose into a pipeline from effective scopes

Degree of control over inheritance of scopes, i.e. <base/> element

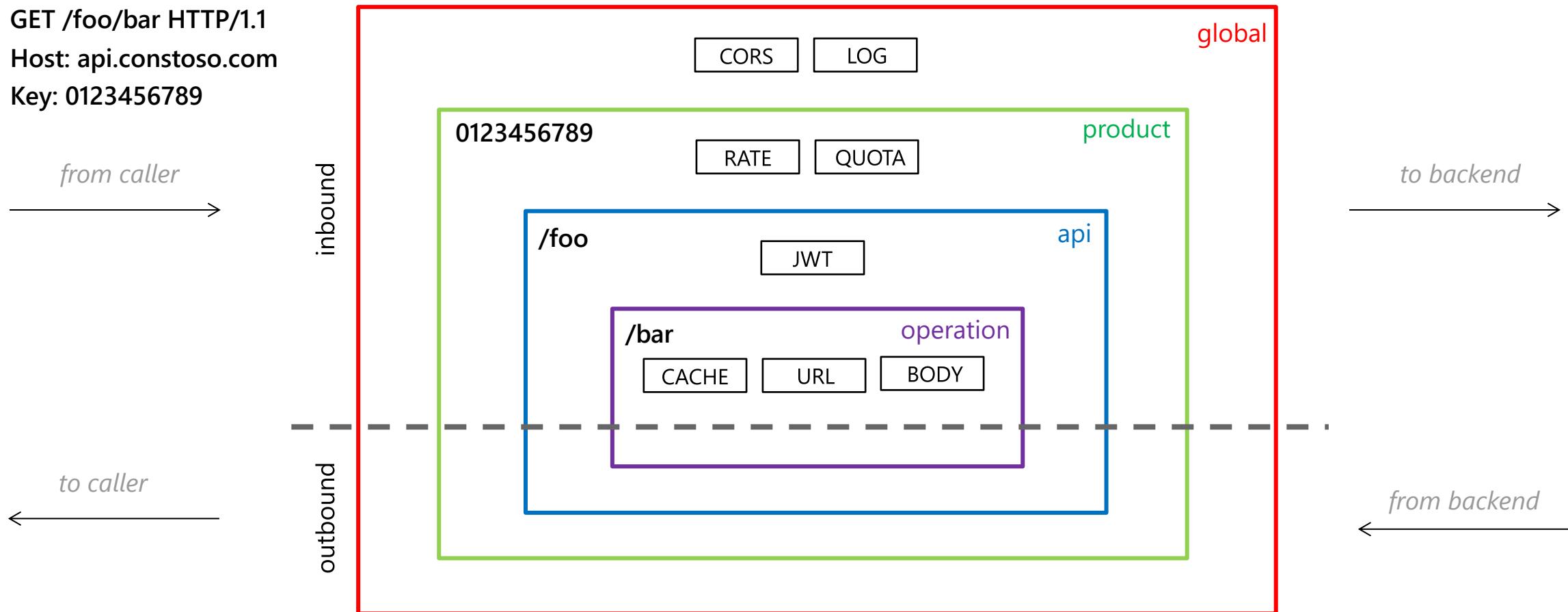
Don't delete <base/> inadvertently

<http://aka.ms/apimpolicyexamples>

Calculate effective policy

Policy scopes

GET /foo/bar HTTP/1.1
Host: api.constoso.com
Key: 0123456789



Policy expressions

C# “snippets” used with policies

Have read-only access to the request context

Use only whitelisted .NET types

Used to configure and conditionally execute policies

Named values

Scoped to an APIM service instance

Keep secrets and “magic” strings out of policies

Provide environment-specific values

Add semantics, if named well

Enable a single point of change

Integrate with Azure Key Vault for an additional layer of protection and access management

```
1 ...
2 <inbound>
3   <base/>
4   <set-variable name="content-length" value="@(context.Request.Headers["Content-Length"]?[0])" />
5   <choose>
6     <when condition="@int.Parse(context.Variables.GetValueOrDefault<string>("content-length")) > {{max-content-length}}">
7       <rewrite-uri template="{{alternate-path-and-query}}"/>
8       <set-backend-service base-url="{{alternate-host}}"/>
9     </when>
10    </choose>
11  </inbound>
12 ...
```

53 policies out of the box

Access restriction	Transformation	Advanced	Dapr integration
<ul style="list-style-type: none">• Check HTTP header• Limit call rate by subscription• Limit call rate by key• Restrict caller Ips• Set usage quota by subscription• Set usage quota by key• Validate client certificate• Validate JWT	<ul style="list-style-type: none">• Convert JSON to XML• Convert XML to JSON• Find and replace string in body• Mask URLs in content• Set backend service• Set body• Set HTTP header• Set query string parameter• Rewrite URL• Transform XML using XSLT	<ul style="list-style-type: none">• Send one way request• Send request• Set HTTP proxy• Set variable• Set request method• Set status code• Control flow• Emit metric• Log to Event Hub• Trace• Mock response• Forward request• Limit concurrency• Return response• Retry• Wait	<ul style="list-style-type: none">• Send request to a service• Send message to a pub/sub topic• Trigger output binding
Authentication	Caching	Cross Domain	Validation policies
<ul style="list-style-type: none">• Authenticate with basic• Authenticate with client certificate• Authenticate with managed identity	<ul style="list-style-type: none">• Get from cache• Store to cache• Get value from cache• Store value from cache• Remove value from cache	<ul style="list-style-type: none">• Allow cross-domain calls• CORS• JSONP	<ul style="list-style-type: none">• Validate content• Validate parameters• Validate headers• Validate status code• Validate GraphQL request

Integration policies

<send-request/>

Response composition (or [gateway aggregation](#))

One client request -> multiple backend requests

Data lookup, complex content transformation, payload or credential validation

Typical pattern:

1. externalize logic as an HTTP endpoint
2. make a call
3. cache the result

<send-one-way-request/>

Traffic mirroring

Coordinate callouts with <wait> for all or any outstanding requests

<log-to-eventhub/>

Event Hub is widely supported within Azure

Custom reporting, batch analytics, archiving, audit

Customer has full control over what is logged, when it is logged and owns the data

We employ buffering (e.g. 200MB per node in Premium)

Delivery is not guaranteed – comprehensive set of metrics is available

It's crucial to adequately scale the target Event Hub

Co-location in the region is highly recommended

Request forwarding

<forward-request/>

Usually inherited from the global scope via <base/>

No policy, no forwarding

Timeout can be set to 30 sec – 10 min (default is 5 min)

Can be configured to follow redirects or (default) return them to caller

<retry/>

Most often used with <forward-request/> but can be used with other policies

Retry is triggered when specified expression evaluates as true

Choice of fixed, linear or exponential back off interval

Optional fast first retry

Does NOT retain a copy of the request automatically

<limit-concurrency/>

Caps the number of concurrent requests forwarded to the backend

Can be used with other policies - limits the number of requests entering enclosed policies

<set-backend-service>

Change backend service during runtime

Can be [configured](#) with conditional policies for blue/green deployment

Caching

Distributed Redis cache hosted as part of service instance (not available in the Consumption tier)

- Shared among all units within a region

- Not persistent and thus gets lost during service updates

- No preloading

<cache-lookup/> and <cache-store/>

- Caches response if it's smaller than 2MB

- Acts as server of origin – ignores cache control headers from backend and replaces them with own

- With expressions possible to use cache control settings sent from backend

- `vary-by-developer` and `vary-by-group` provide additional scope control

- Can be configured to cache requests with Authorization header

- Properly handles conditional requests (e.g. if-match, if-modified-since)

- Cache hit ratio is provided as a metric

<cache-lookup-value> & <cache-store-value>

- Entity to cache and a key are specified by expressions

- Invalidation

- TTL or LRU

- Any policy change invalidates cache entries at that scope

- <cache-remove-value/> removes an entry with a specified key

Bring your own cache

Add externally provisioned, Redis-compatible cache

- Full control over cache configuration and size

- Ability to preload and purge cache content

- Ability to independently scale cache

Only cache option in the Consumption tier

Cache policies are extended to work with external cache

- Added `cache-preference` attribute

- Can be set to "internal", "external", (default) "prefer-external"

```
<cache-lookup downstream-caching-type="private" must-revalidate="true" cache-preference="external" >
|...<vary-by-query-parameter>version</vary-by-query-parameter>
</cache-lookup>
```

Can use different cache types at different scopes

Throttling

Accuracy of (distributed) throttling policies is limited by synchronization latency

<rate-limit-by-key/>

Number of calls allowed in short interval (usually 1 sec)

Enforced per region

Key expression specifies throttling semantics, e.g. caller IP, subscription ID, developer ID

Uses sliding time window, i.e. last 5 seconds

Counts every request or only the ones that meet specified condition, e.g. only 200 OK

Different requests can be weighted differently, e.g. based on cost to the backend

Legacy <rate-limit/> == <rate-limit-by-key/> with subscription ID as a key

<quota-by-key/>

Total number of calls and/or bytes per time period (usually hour, day, week, month)

Enforced per service instance

Key expression specifies throttling semantics, e.g. caller IP, subscription ID, developer ID

Uses calendar time

Counts every request or only the ones that meet specified condition, e.g. status < 400

Different requests can be weighted differently, e.g. based on value provided to the caller

Legacy <quota/> == <quota-by-key/> with subscription ID as a key

Authentication

Authentication using subscription keys is supported out-of-the-box without configuring policies

<validate-jwt>

- validates JSON Web Tokens
- Supports JWS and JWE (RSA256 and HS256)
- Supports Open ID Configuration endpoint
- Can also check specific claims
- Can be configured at any policy scope

<validate-client-certificate>

Enforce that a certificate presented by a client matches the specified validation rules and claims, such as subject, thumbprint, or issuer

```
<validate-client-certificate>
  validate-revocation="true"
  validate-trust="true"
  validate-not-before="true"
  validate-not-after="true"
  ignore-error="false"
  <identities>
    <identity
      thumbprint="BEFC6215108D7CA1DAD7A01AFBDC74F13E8681BC" />
  </identities>
</validate-client-certificate>
```

Transformation

<set-header> and <set-query-parameter>

Add/remove/modify headers and query parameters of incoming and outgoing requests

<set-body>

Set the payload of incoming and outgoing requests

<rewrite-url>

Convert request URL from its public form to the form expected by the backend service

<xml-to-json> and <json-to-xml>

Convert payload of incoming and outgoing requests between XML and JSON

<find-and-replace>

Find and replace substrings in the payload of incoming and outgoing requests

<xsl-transform>

Applies XSL transformation to XML in the payload of incoming and outgoing requests

Validation

<validate-content>

Validates the size or JSON schema of a request or response body against the API schema

<validate-parameters>

Validates the header, query, or path parameters in requests against the API schema

<validate-headers>

Validates the responses headers against the API schema

<validate-status-code>

Validates the HTTP status codes in responses against the API schema

<validate-graphql-request>

Validates and authorizes a request to a GraphQL API

Visual Studio Code

Designed to increase productivity

Convenient resource explorer

Advanced policy editor

Policy debugging

Syntax check and IntelliSense

Embedded REST client for testing

Integrated with automation tools

Command palette support



Azure API Management

Microsoft | 91,656 installs | (9) | Free

An Azure API Management extension for Visual Studio Code.

[Install](#)

[Trouble Installing?](#)

[Overview](#)

[Version History](#)

[Q & A](#)

[Rating & Review](#)

Visual Studio Marketplace v1.0.1

installs 91.66K

Azure Pipelines

succeeded

license MIT

Azure API Management Extension for Visual Studio Code

Use the Azure API Management extension to perform common management operations on your Azure API Management service instances without switching away from Visual Studio Code.

[Azure API Management](#) is a fully managed service that helps customers to securely expose their APIs to external and internal consumers. API Management serves as a facade and a front door for the API implementation, which enables their frictionless consumption by developers. Visit [this page](#) for more information and resources about Azure API Management.

Live policy debugging in Visual Studio Code

Postmortem debugging

Rely on logs after requests are processed

Live debugging

Follow the processing of requests in real time

Features

Initiate live debugging session from VS Code

Single-step through policies

Set breakpoints at individual policies

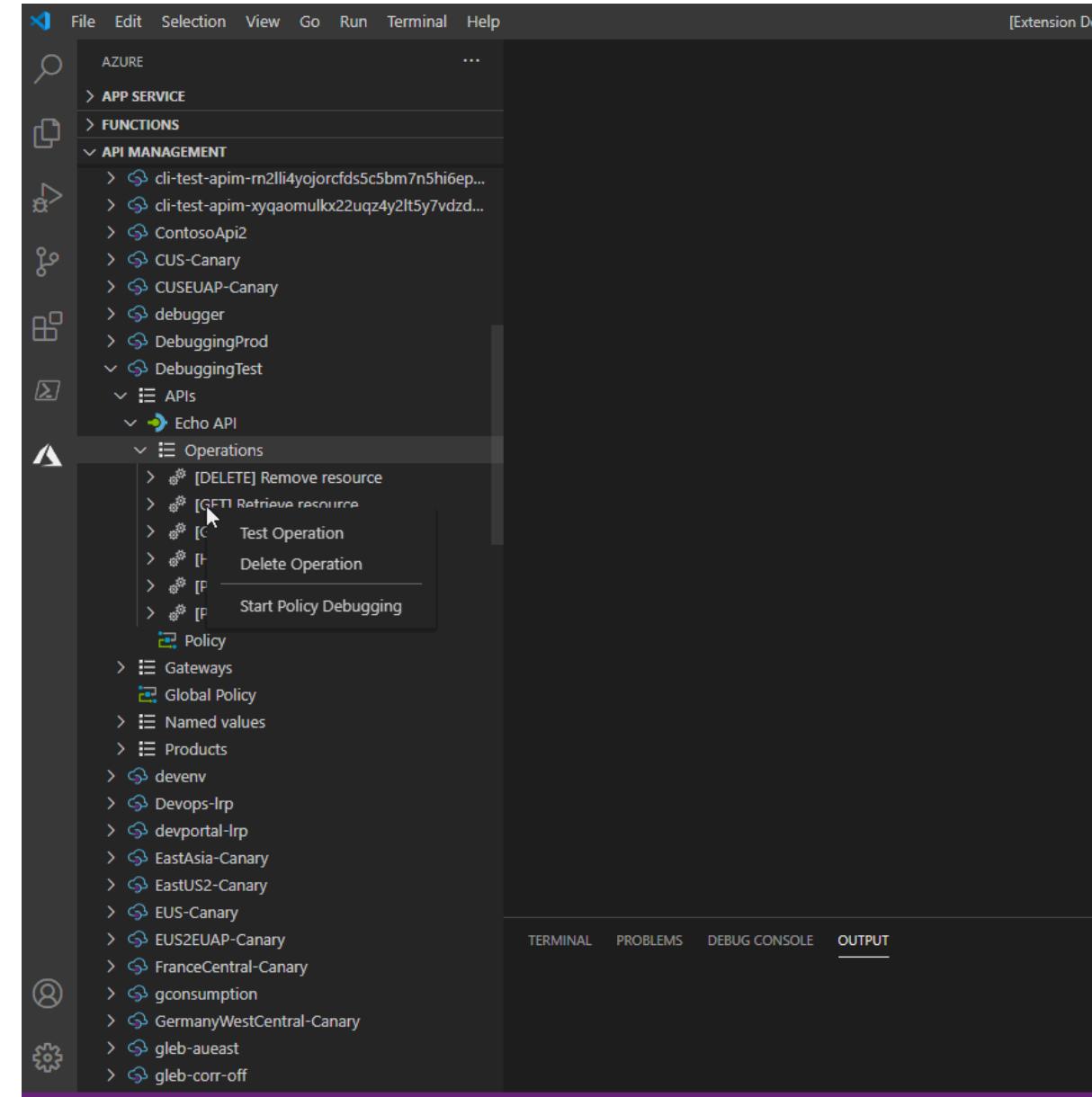
Inspect system-created and user-created variables

Examine errors

Restrictions & limitations

Developer-tier only

One debugging session per instance

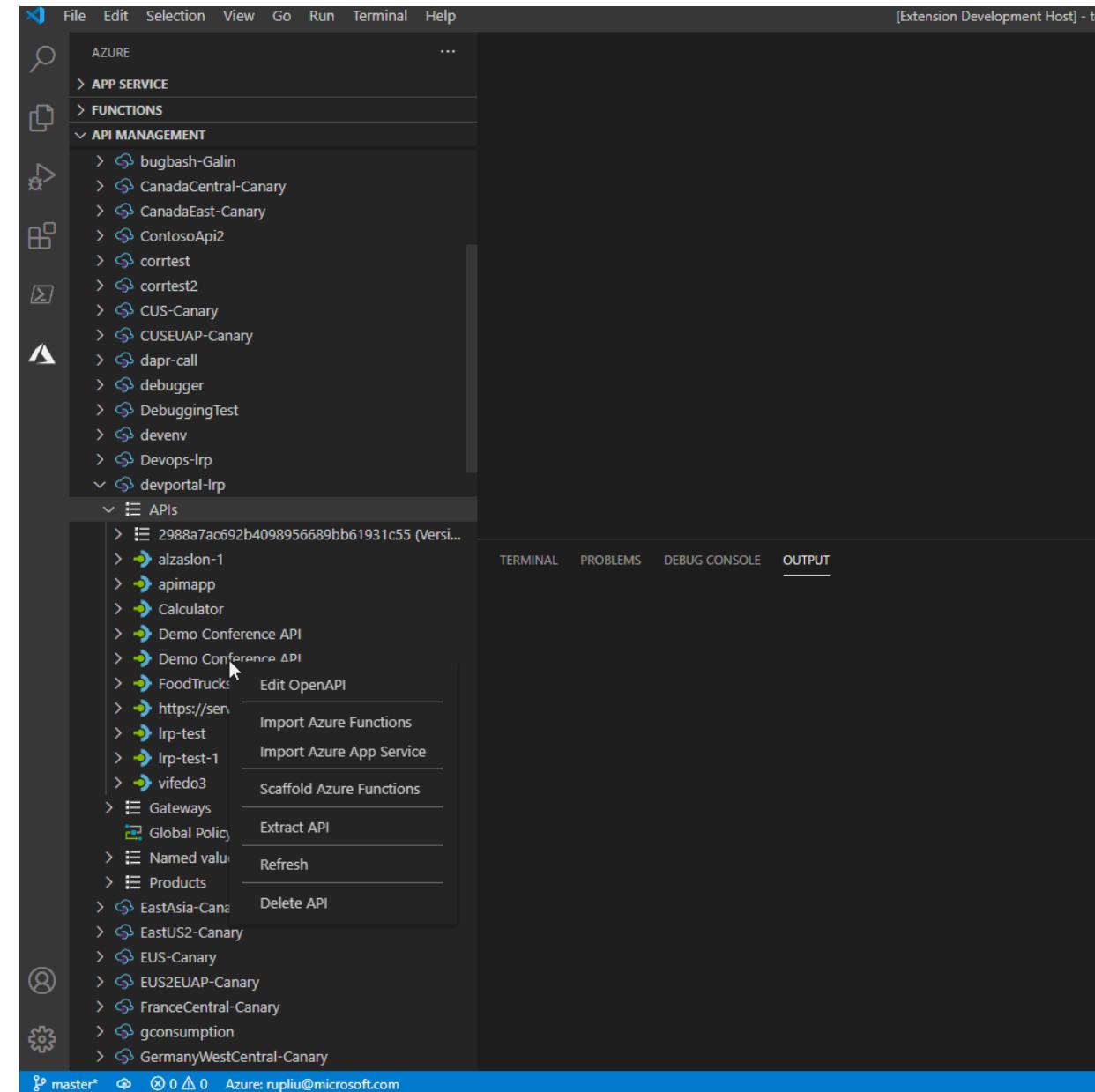


Design-first API development

Design an API with OpenAPI spec
Mock API responses to unblock front-end developers

Scaffold Azure Functions in VS Code
Fill in the business logic
Supported languages

C#
Java
Python
TypeScript



Automate API Management deployments

Context

Multiple deployment environments, e.g. development, QA, production

Some of the environments are shared, e.g. production

Many API development teams each responsible for one or more APIs

Problems

Automate deployment of APIs into API Management

Migrate configurations from one environment to another

Avoid interference between development teams

There is no one-size-fit-all solution

Deployment options



- APIs
- PowerShell Cmdlets
- Azure CLI
- Resource Manager Templates
- Bicep
- Terraform
- SDKs

SOAP-to-REST

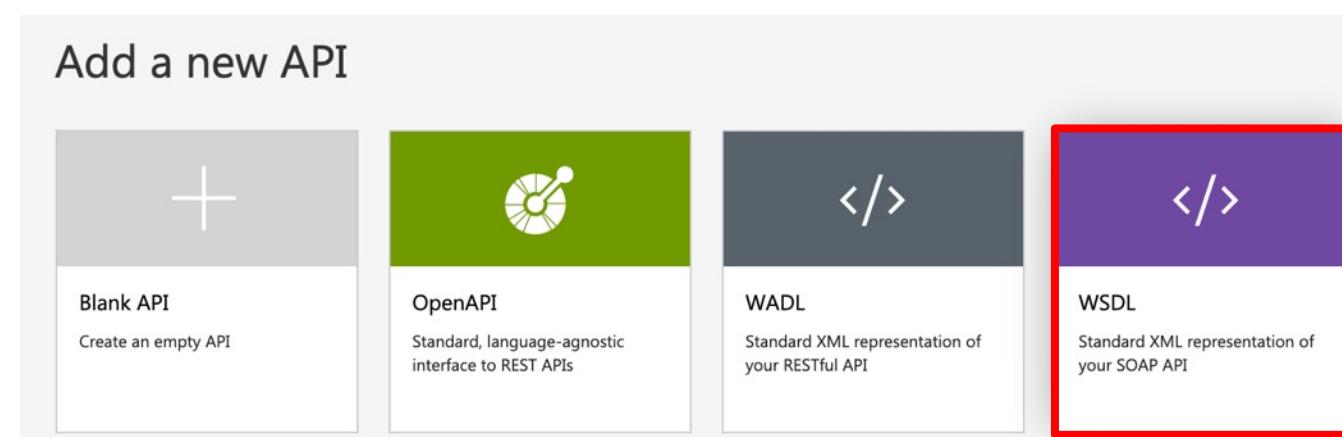
One-click modernization of legacy services

Import a WSDL, get a REST API façade instantly

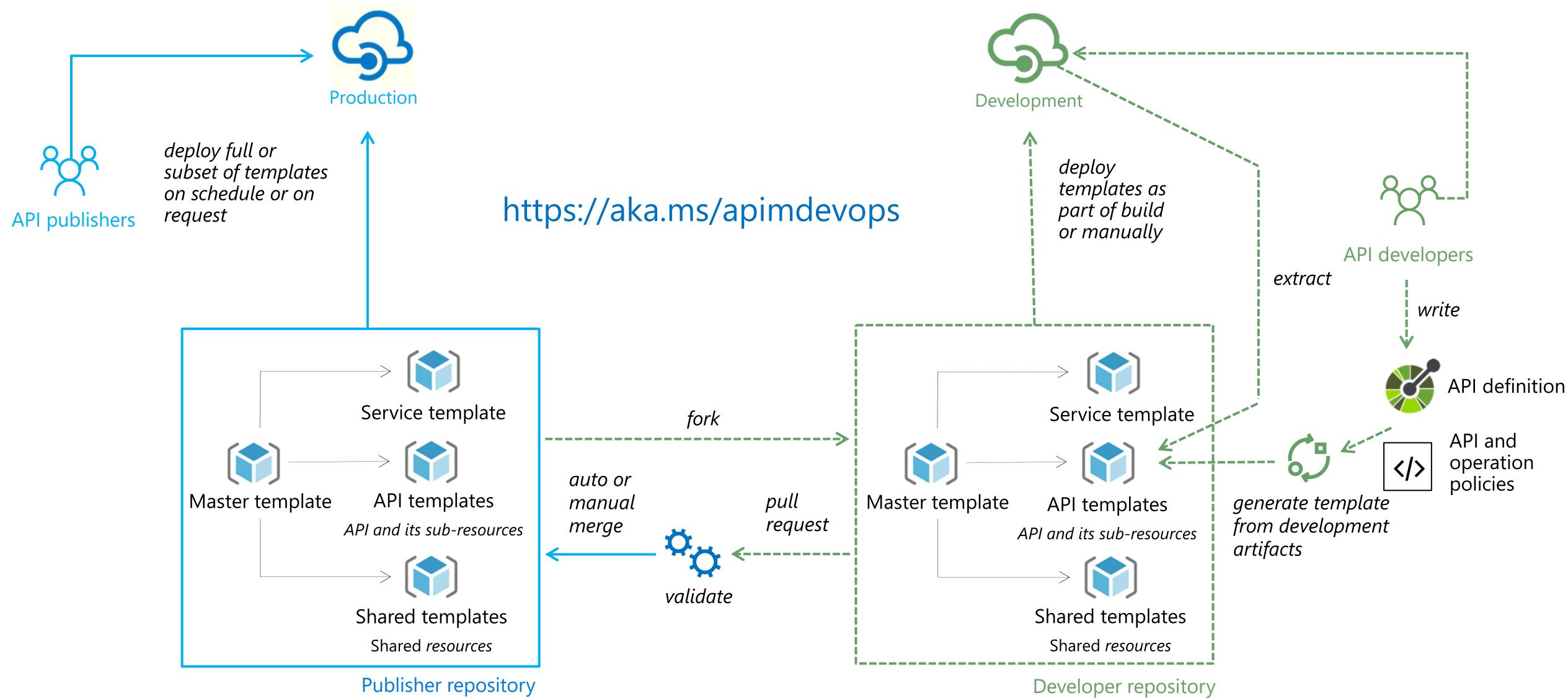
APIIM does all the conversions using heuristics

Customers have full control of the conversions through policies

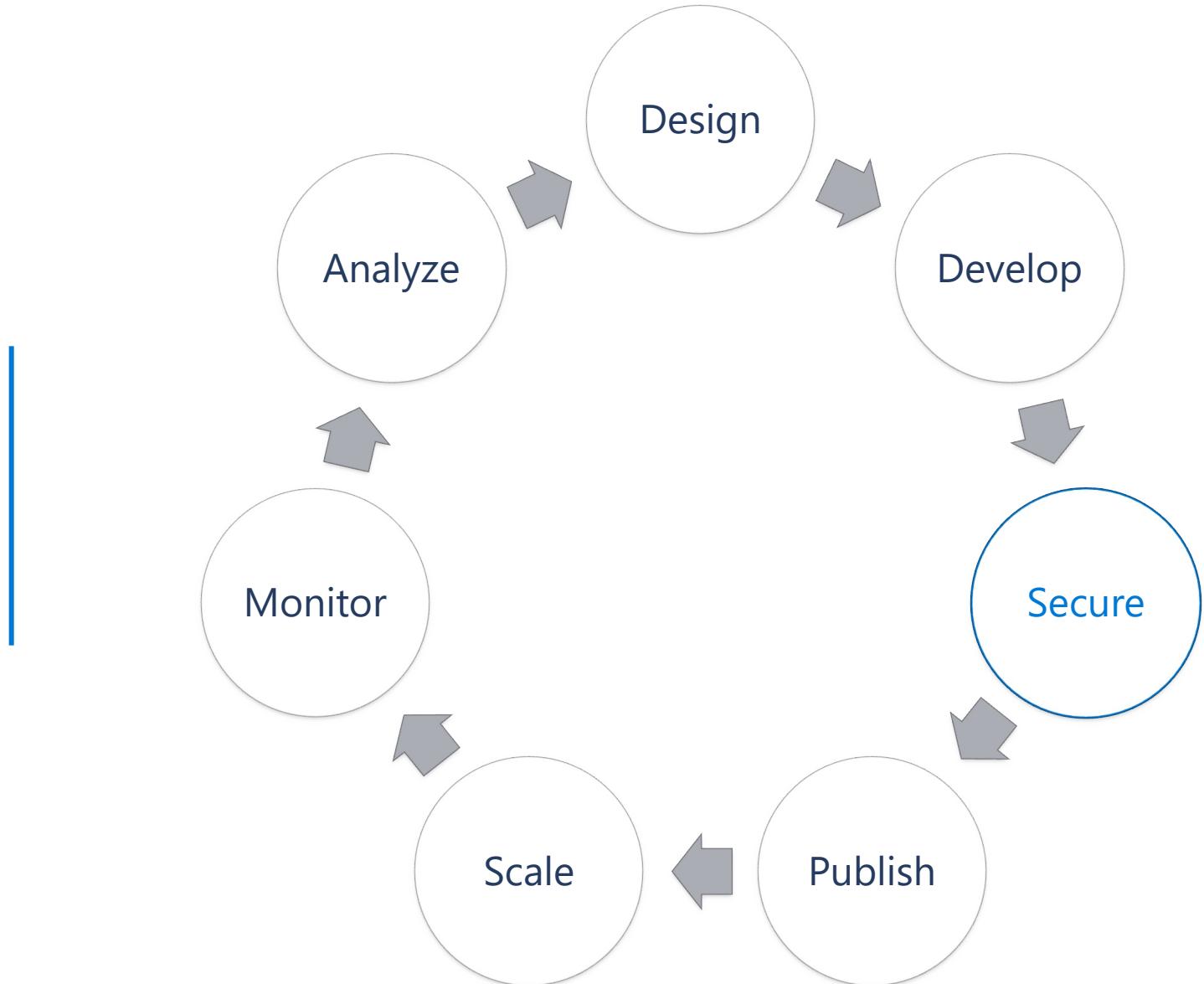
Known [restrictions](#)



Recommended approach



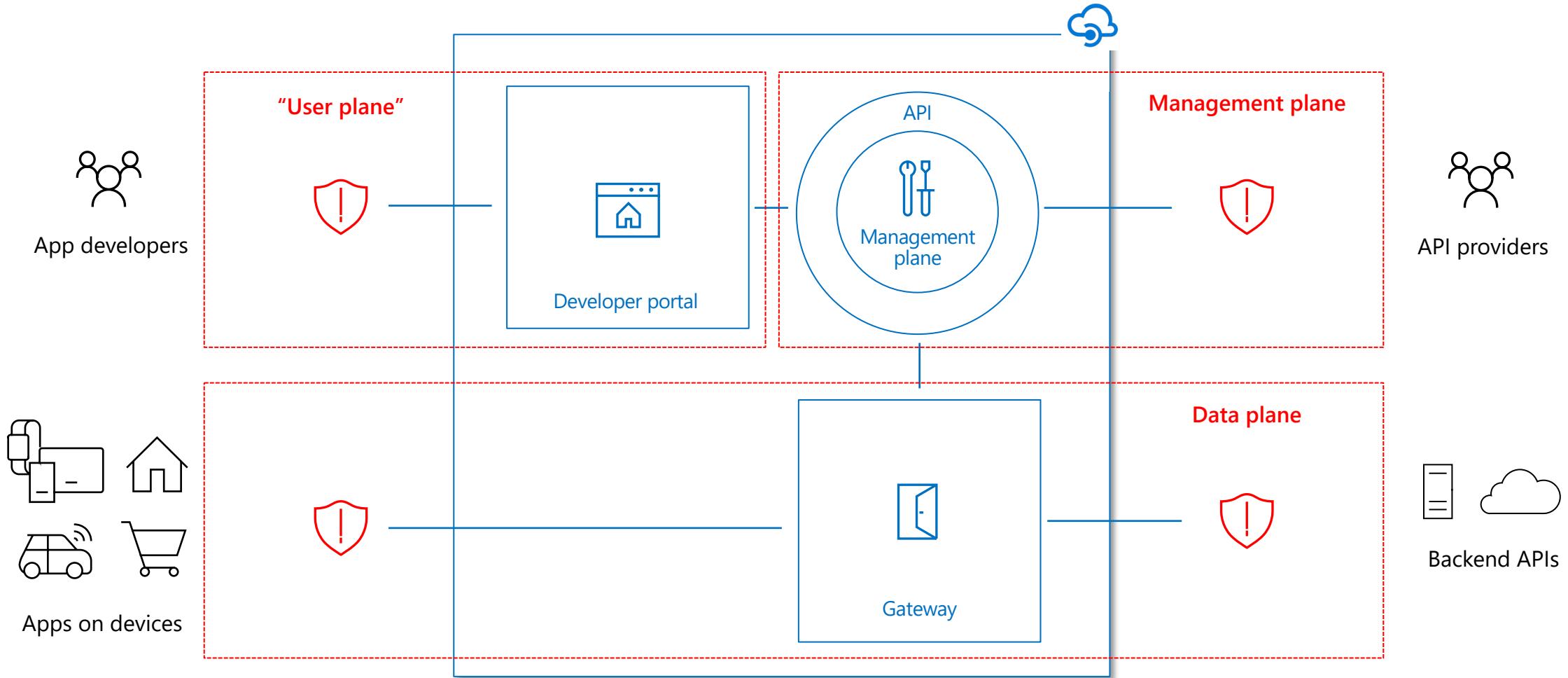
API life cycle: secure



API Management to the rescue

#	OWASP API Top 10 (2019)	Mitigations and preventive measures in API Management
1	Broken Object Level Authorization	
2	Broken Authentication	Key/token/certificate-based authentication Request transformation
3	Excessive Data Exposure	Filtering or masking sensitive data Request and response validation
4	Lack of Resources & Rate Limiting	Throttling and quota limit Backend concurrency
5	Broken Function Level Authorization	Key/token-based authorization Custom authorization
6	Mass assignment	Request and response validation
7	Security misconfigurations	TLS enforcement and configuration CORS Sanitization of response headers and error messages Ciphers and protocols management Coming soon: security configuration recommendations
8	Injection	Request and response validation
9	Improper Assets Management	Up-to-date API catalog API lifecycle management
10	Insufficient logging and monitoring	Logging

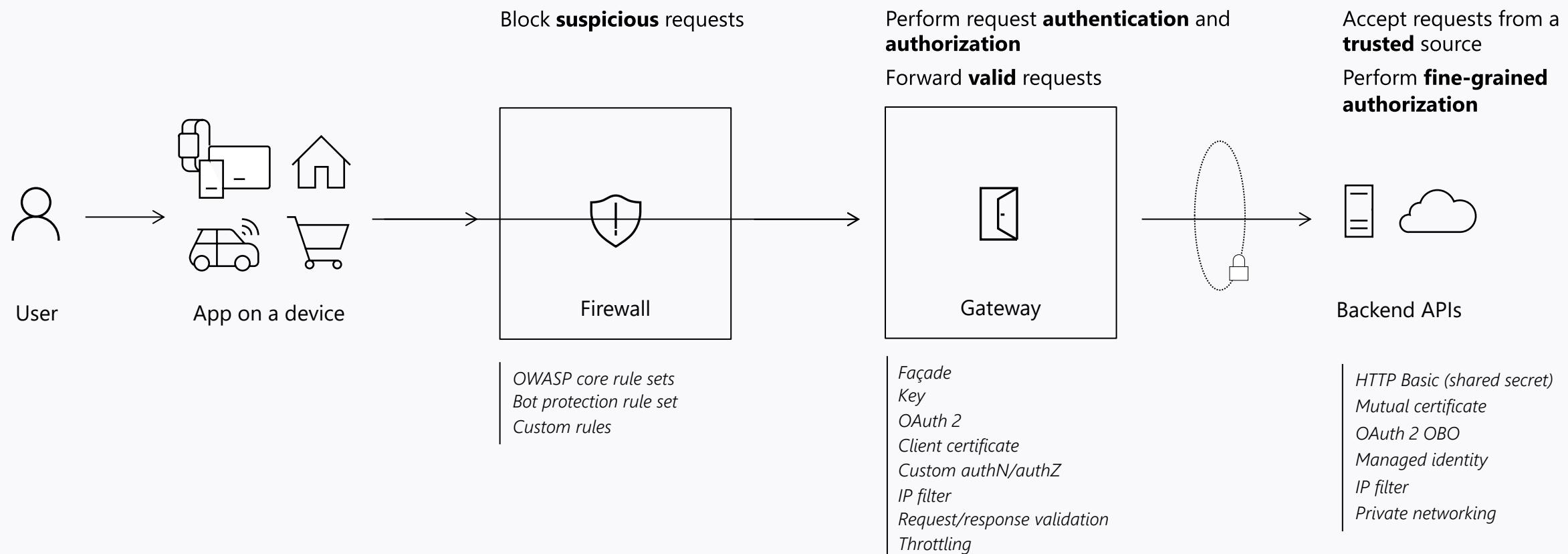
Secure all points of interaction



Data plane security

Secure and protect backend APIs

Layered defense
Separation of responsibilities between the layers



Façade

- Expose selected backend APIs
- Allow chosen HTTP methods and routes
- Enforce TLS and its configuration
- Define CORS rules
- Restrict client IPs

Keys

- Turned on and UUID by default
- Can be rotated and set to custom values
- Identify developer and app
- Roughly equivalent to HTTP Basic security-wise

JWT



Signed (JWS) and encrypted (JWE)

Validate via policy and expressions

Enforce claims

Require signatures and expiration time

Provide keys inline or via a metadata endpoint

Client certificates

- Issued by trusted and untrusted CA
- Use the validate-client-certificate policy
- Require certificate on per host basis
- Check or ignore revocation lists

Custom authentication and authorization

Integrate with a bespoke or unsupported identity or authorization system

Call out to an external HTTPS endpoint

Cache the result for efficiency

Throttling

Rate limit

Approximate

Per region

Key expression defines throttling semantics

Can count requests with specific status code

Variable increment count

Quota

Calls and data transfer

Approximate

Per service

Key expression defines throttling semantics

Can count requests with specific status code

Variable increment count

Concurrency limit

Precise

Per node

Response sanitization

- Filter or mask confidential data
- Standardize error messages
- Remove sensitive headers

Request and response validation

Use request and response validation policies to protect your APIs from vulnerabilities



Excessive data exposure

OWASP API Top 10



Mass assignment

OWASP API Top 10



Injection

OWASP API Top 10



DoS large payload attack

Validation policies

Four policies

- Validate content - validates the size or JSON schema of a request or response body against the API schema
- Validate parameters - validates the request header, query, or path parameters against the API schema
- Validate headers - validates the response headers against the API schema
- Validate status code - validates the HTTP status codes in responses against the API schema

Prevention and detection modes

Granular overrides for child elements

Logging of errors to a context variable

Use the tracing policy to send logs to Application Insights

Performance implications and limits

Max body size: 100 kB

Max schema size: 4 MB

The larger the API schema size, the lower the throughput

The larger the payload in a request or response, the lower the throughput

The size of the API schema has a larger impact on performance than the size of the payload

Validation against an API schema that is several megabytes in size may cause request or response timeouts

Mass assignment

Attackers modify object properties they are not supposed to

Usually caused by binding client-provided data (e.g., JSON) to data models, without explicit filtering of properties

Attackers explore other API endpoints, read documentation, or blindly guess additional object properties

Attackers inject additional object properties into request payloads

Mitigation

Set the "additionalProperties" option of request objects' JSON schemas to false

Precisely define request object schemas in the API specification and enforce them with the validate-content policy

```
<validate-content unspecified-content-type-action="prevent" max-size="102400" size-exceeded-action="prevent">
    <content type="application/json" validate-as="json" action="prevent" />
</validate-content>
```

Injection

Malicious data in a request executes unintended commands or accesses data without proper authorization

For example, SQL or NoSQL injection

Mitigation

Provide format properties, like regex for text fields, in the API specification's object schemas and enforce them with the validate-content policy

```
<validate-content unspecified-content-type-action="prevent" max-size="102400" size-exceeded-action="prevent">
    <content type="application/json" validate-as="json" action="prevent" />
</validate-content>
```

Excessive data exposure

API responses surface sensitive or excessive data

Developers tend to expose all object properties without considering their individual sensitivity
They rely on clients to perform the data filtering before displaying it to the user

Mitigation

Set the "additionalProperties" option of response objects' JSON schemas to false
Precisely define response object schemas in the API specification and enforce them with the validate-content policy
Define all allowed response status codes in the API specification and enforce them with the validate-status-code policy
Precisely define all allowed response headers in the API specification and enforce them with the validate-headers policy

```
<validate-headers specified-header-action="prevent" unspecified-header-action="prevent"/>
```

```
<validate-status-code unspecified-status-code-action="prevent" />
```

DoS large payload attack

Large-payload requests cause API outages

Malicious requests block the API traffic on system's bottlenecks

They occupy networking resources and consume excessive computing power

Mitigation

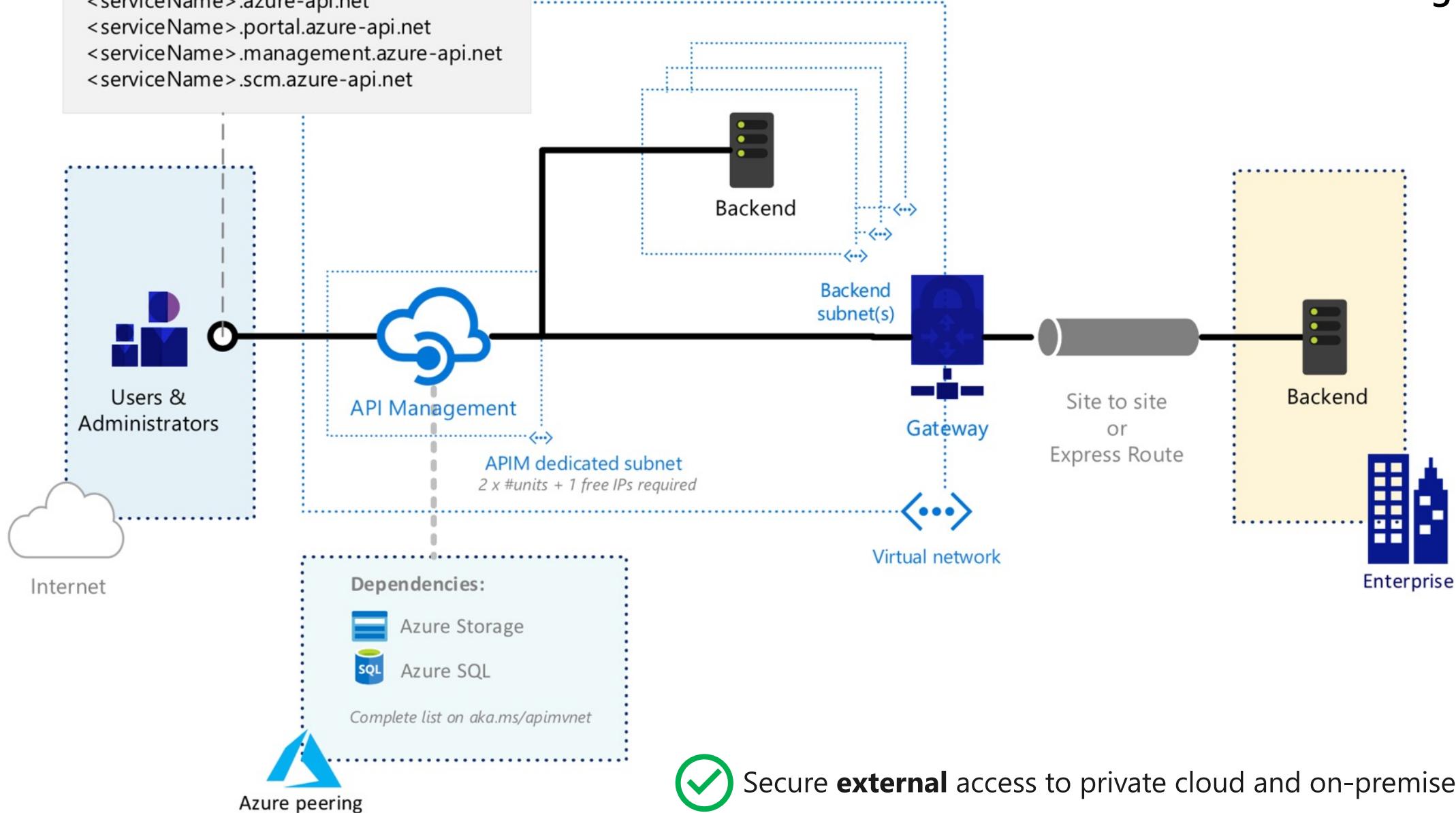
Enforce maximum request content size with the content-validation policy

```
<validate-content max-size="102400" size-exceeded-action="prevent"  
unspecified-content-type-action="prevent" />
```

Private networking and upstream security

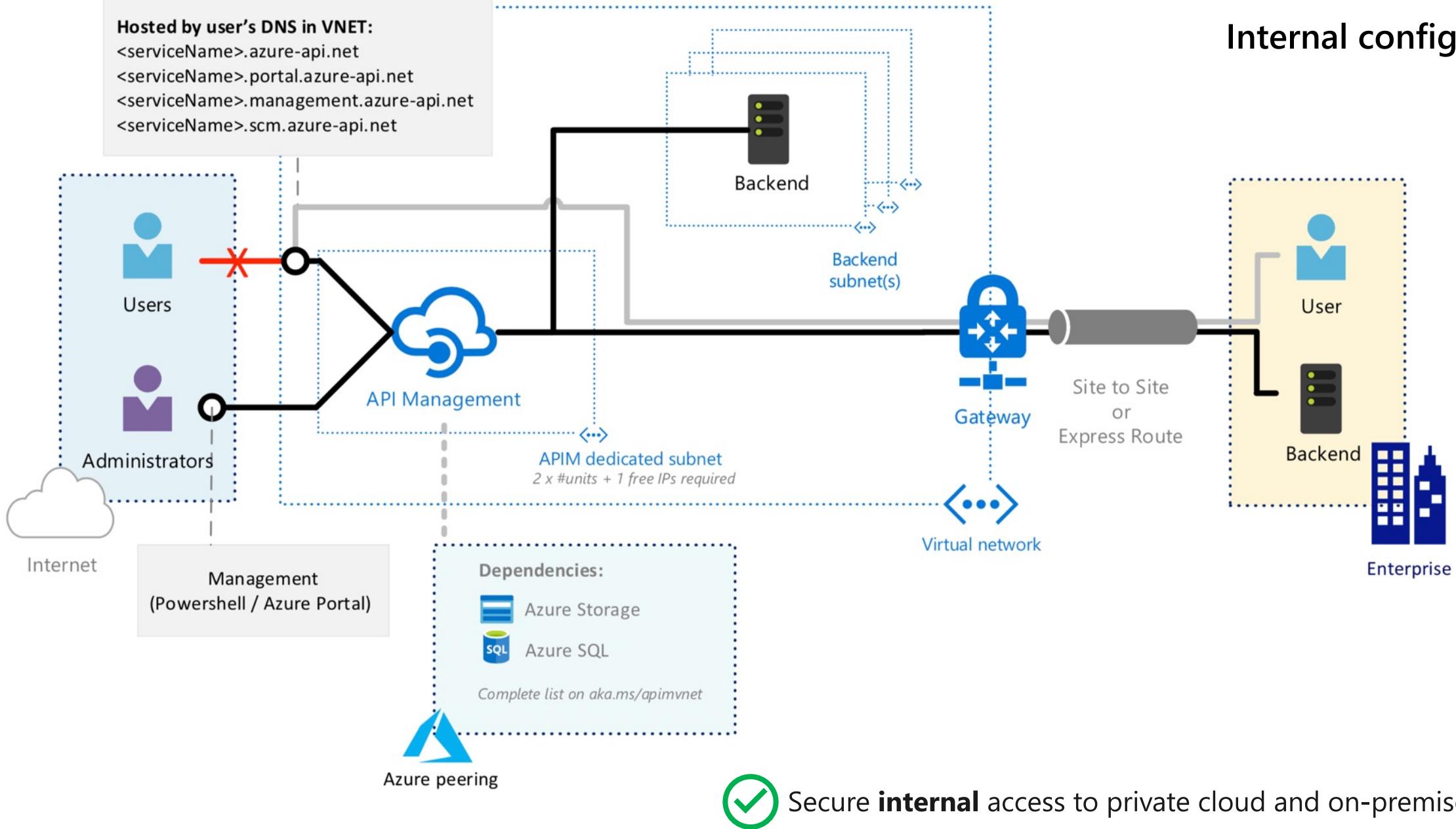
External configuration

Hosted by Azure DNS:
<serviceName>.azure-api.net
<serviceName>.portal.azure-api.net
<serviceName>.management.azure-api.net
<serviceName>.scm.azure-api.net



Secure **external** access to private cloud and on-premises endpoints

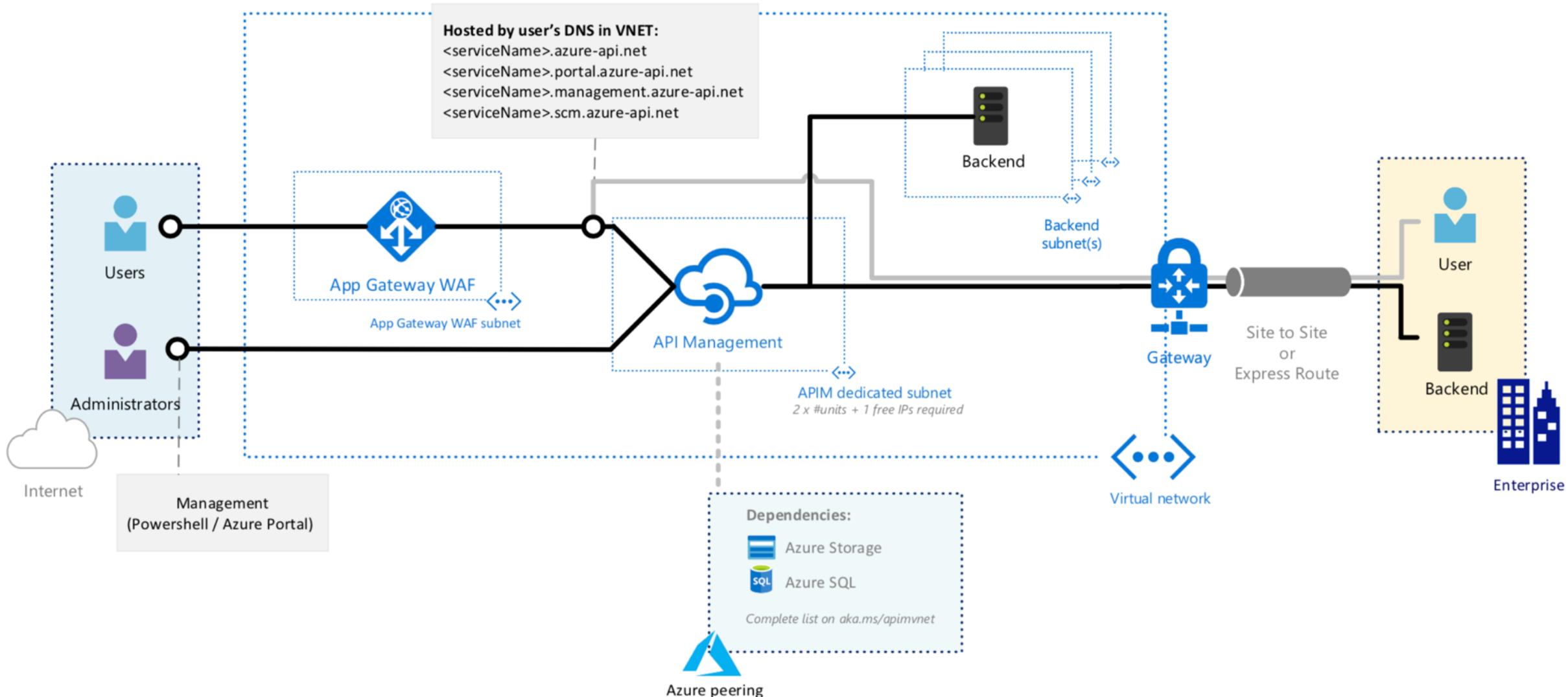
Internal configuration





More secure **external** access to private and on-premises endpoints
Secure **internal** access to private cloud and on-premises endpoints

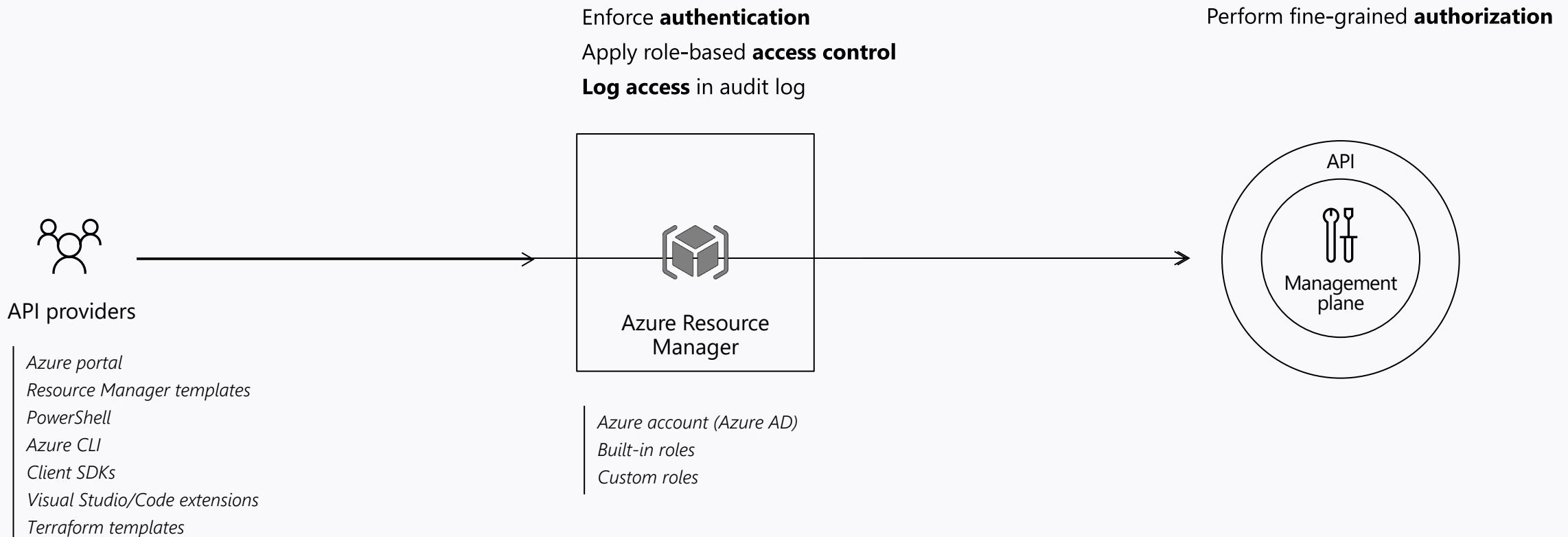
Internal configuration with WAF



Management plane security

Manage and enforce permissions

Access only by authenticated users
Fine-grained permissions based on roles
Audit log



“User plane” security

Manage visibility and access to APIs

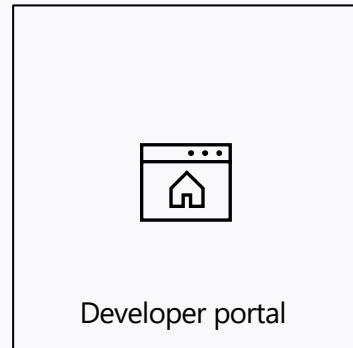
Enforce authentication rules
Present different APIs to various groups of users
Impose onboarding rules

Regulate user onboarding to APIs



App developers

Employee developers
Partner developers
Customer developers
Public developers



Work and internet accounts
Integration with Azure AD B2B/B2C
Custom (delegated) authentication
Native and Azure AD groups
Self-service or invite-only onboarding to API products
Auto or manual approval of subscriptions
Limits on the number of subscriptions
Subscription suspension and revocation

Compliance

Meets a multitude of global, regional, country and industry specific regulations

ISO 27001

PCI DSS

HIPAA

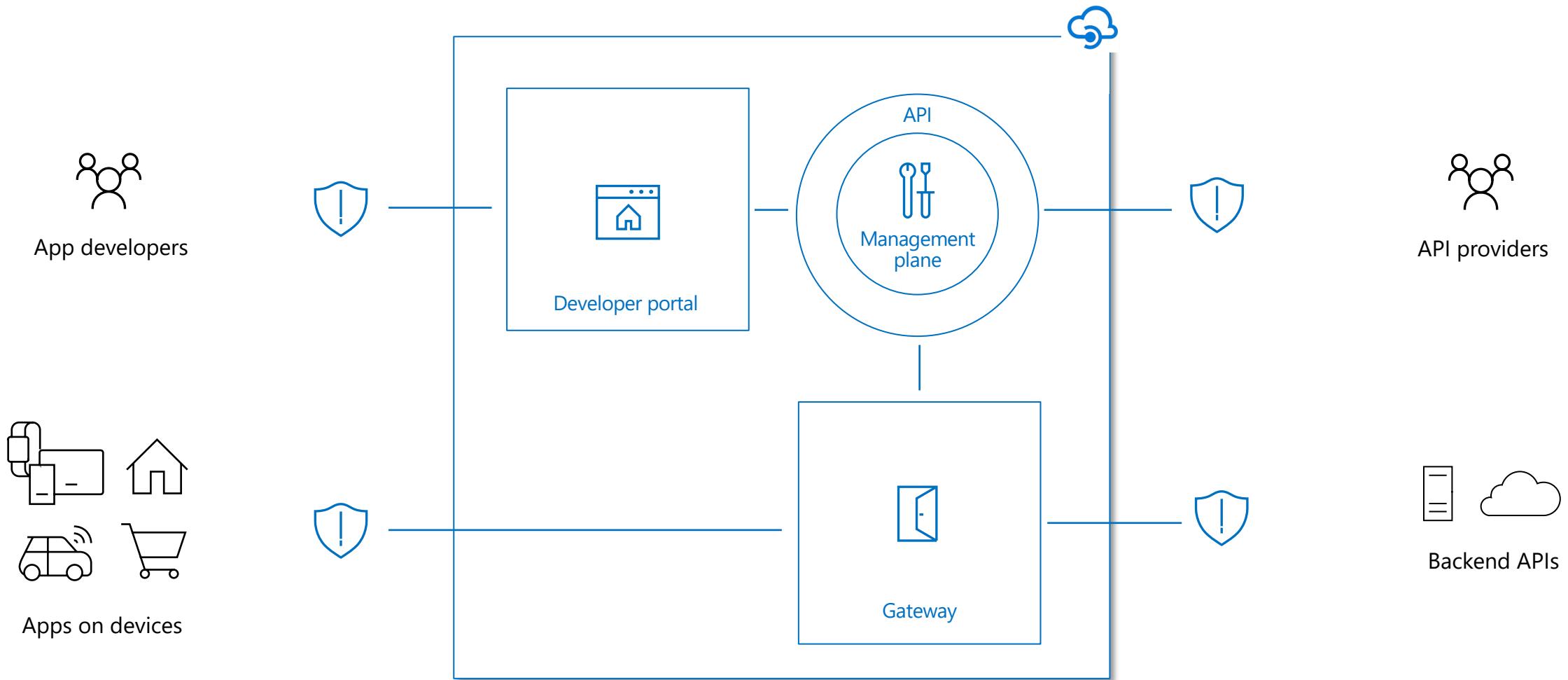
FedRAMP High

GDPR

...

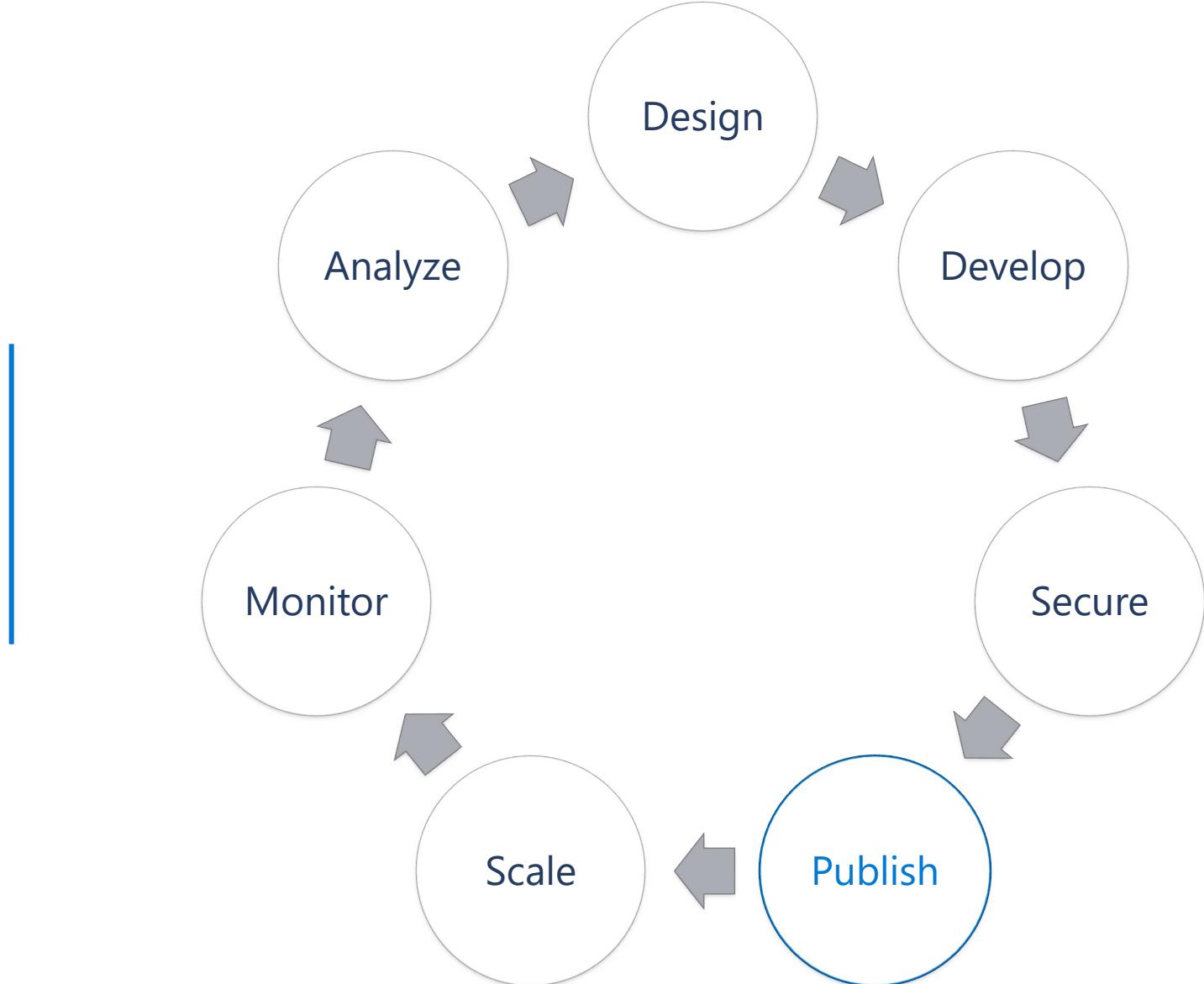
Full list and documentation available on <https://aka.ms/apim/trusted>

End-to-end security and compliance



Azure platform + built-in capabilities + Azure services

API life cycle: publish



Developer portal is a discovery and self-onboarding point for application developers

Built-in developer portal lets API consumers

[Discover APIs](#)

[Learn how to use them](#)

[Test them out with interactive console](#)

[Create and manage accounts](#)

[Request and manage API access](#)

[Analyze API usage](#)

Developer portal is...

Built-into API Management

Open the portal within seconds; updates are on us.

Fast go-to-market

Rely on default styling and content to minimize customizations.

Easily customizable

Author content and brand the portal with a drag-and-drop visual editor.

Open-source

Browse the codebase and engage with the community on GitHub.

Extensible

Extend the codebase with custom logic and self-host the resulting portal.

Automatable

Automate deployments via APIs.

[master](#)[5 branches](#)[39 tags](#)[Go to file](#)[Add file](#)[Code](#)

ygrik Contrast colors fix for try button and focus selection (#1239)	✓ 9f8cc5d 6 hours ago	513 commits
.github/ISSUE_TEMPLATE Update issue templates (#323)		17 months ago
.vscode Added end-to-end and unit-test scaffolds. Fixed issue with request he...		7 months ago
community/widgets/document-details Fixed several accessibility issues. (#1195)		21 days ago
examples Fixed several accessibility issues. (#1195)		21 days ago
js Enhanced HipCaptcha initialization in absence of jQuery. (#357)		16 months ago
readme New cover image (#258)		2 years ago
scaffolds/widget Upgraded paperbits libraries to 0.1.382. (#1174)		last month
scripts.v2 Changed conflict destToken -> destKey (#1164)		2 months ago
scripts.v3 Uncommented /portalRevisions endpoint to enable publishing. (#1236)		4 days ago
scripts Remove unused PC image and add missing contoso black logo (#1069)		4 months ago
src Contrast colors fix for try button and focus selection (#1239)		6 hours ago
tests Added end-to-end and unit-test scaffolds. Fixed issue with request he...		7 months ago
.gitattributes Added source files		2 years ago
.gitignore Excluded .vs folder from github and vscode tracking		2 years ago
CONTRIBUTIONS.md Add contributions guidelines links (#421)		15 months ago

About



Developer portal provided by the Azure API Management service.

aka.ms/apimlove

[microsoft](#) [azure](#) [api-management](#)

[developer-portal](#)

[Readme](#)

[MIT License](#)

Releases 39

[2.8.0](#) Latest
3 days ago

[+ 38 releases](#)

Contributors 25



[+ 14 contributors](#)

Portal customizations

Create content with the drag-and-drop visual editor without writing any code

Use widgets to connect to the API Management service (i.e., to retrieve the list of APIs or sign in a user)

Customize the portal in a dedicated style guide panel

The screenshot shows a web-based visual editor interface for API Management. On the left, there's a vertical toolbar with icons for list items, colors, and fonts. The main content area contains several heading levels (Level 3, Level 4, Level 5, Level 6) and a code snippet for a PictureModel class. A quote by Robert Bringhurst is also present.

Heading level 3

Heading level 4

Heading level 5

Heading level 6

```
class PictureModel {  
    sourceKey: string;  
    caption: string;  
}
```

Typography is the craft of endowing human language with a durable and expressive form.
- Robert Bringhurst

• List item 1
• List item 2
• List item 3
 ◦ Nested list item 1
 ◦ Nested list item 2

1. List item 1
2. List item 2
3. List item 3
 1. Nested list item 1
 2. Nested list item 2

COLORS

Background - black Background - white Featured Formatted text Bg Links Links hover Primary Primary +2 Primary -2 Text - black

Background - grey Text - white Transparent

Add color

Icons at the bottom include a file, user, and navigation symbols.

Search operations

Group by tag

pet

POST Add a new pet to the store**DEL** Deletes a pet**GET** Find pet by ID**GET** Finds Pets by status**GET** Finds Pets by tags**PUT** Update an existing pet**POST** Updates a pet in the store with for...**POST** uploads an image

store

DEL Delete purchase order by ID**GET** Find purchase order by ID**POST** Place an order for a pet**GET** Returns pet inventories by status

user

POST Create user**POST** Creates list of users with given input**DEL** Delete user**GET** Get user by user name**GET** Logs out current logged in user session**GET** Logs user into the system**PUT** Update user

Swagger Petstore

API definition

This is a sample Pet Store Server based on the OpenAPI 3.0 specification.

Find pet by ID

Returns a single pet

pet

Request

```
GET https://mibudz-private.azure-api.net/petstore/pet/{petId}
```

Request parameters

Name	In	Required	Type
petId	template	true	integer

Response: 200 OK

successful operation

application/xml application/json

PetRequest-xml

{}

Name	Required	Type	Description
id	false	int64	

Authorization

Subscription key

subscription key

Parameters

petId

value

+ Add parameter

Headers

Cache-Control

no-cache

Remove

+ Add header

HTTP

Curl

C#

Java

JavaScript

PHP

Python

Ruby

Objective C

HTTP request

Copy

```
GET https://mibudz-private.azure-api.net/petstore/pet/{petId} HTTP/1.1
```

Cache-Control: no-cache

Send

Last hour

Today

Last 7 days

Last 30 days

Last 90 days

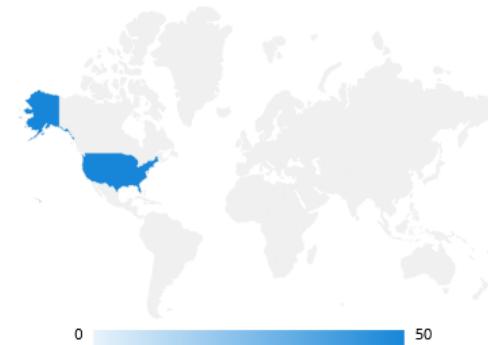
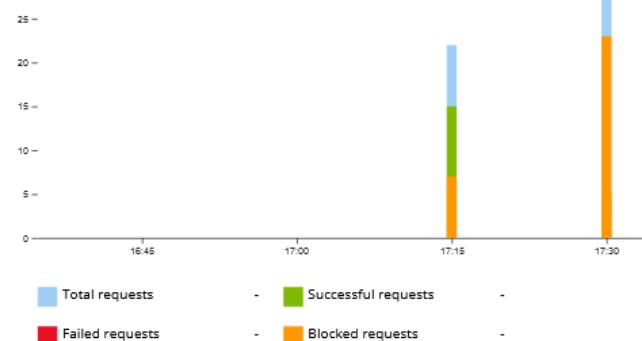
API usage reports

Application developers explore their usage of APIs in the developer portal

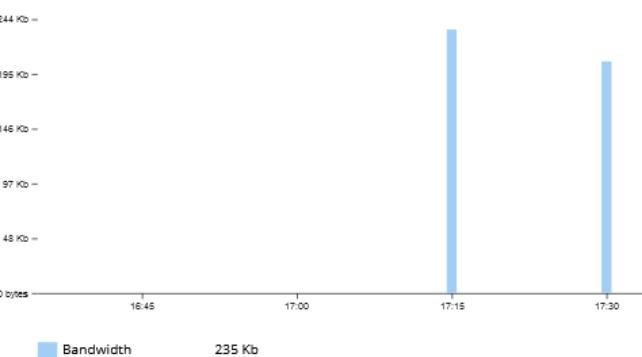
API providers analyze the usage in the Azure portal

Reports are grouped by time, response type, bandwidth, products, subscription keys, APIs, and API operations

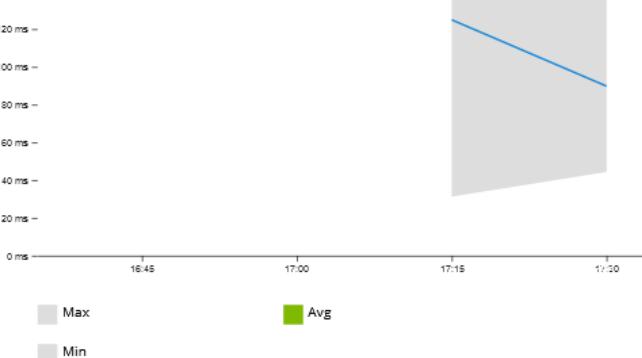
API calls



Data transfer



API response times



Extensibility of the developer portal

If the out-of-the-box capabilities are insufficient, you can:

- Request a feature on GitHub
- Contribute code on GitHub
- **Fork the repository, extend the code base, and self-host the portal**

Self-hosting the portal is simple and efficient

Portal generates static files for hosting in the cloud or on premises

Recommended hosting with Azure Storage Account

Developer portal

Welcome to Contoso!

We provide industry-leading APIs.

[Sign up](#)[Explore APIs](#)

99.95% availability

Our APIs can be used for mission-critical systems.

25 million API calls daily

Our APIs define the industry's standards.

1 million active users

Millions of people trust us.

Welcome to Contoso!

We provide industry-leading APIs.

[Sign up](#)[Explore APIs](#)

99.95% availability

Our APIs can be used for mission-critical systems.

25 million API calls daily

Our APIs define the industry's standards.

1 million active users

Millions of people trust us.

API versioning

Revisions

For non-breaking changes

Providers choose when to deploy

API requests default to current revision

Test by specifying revision ID, then promote

Versions

For breaking changes

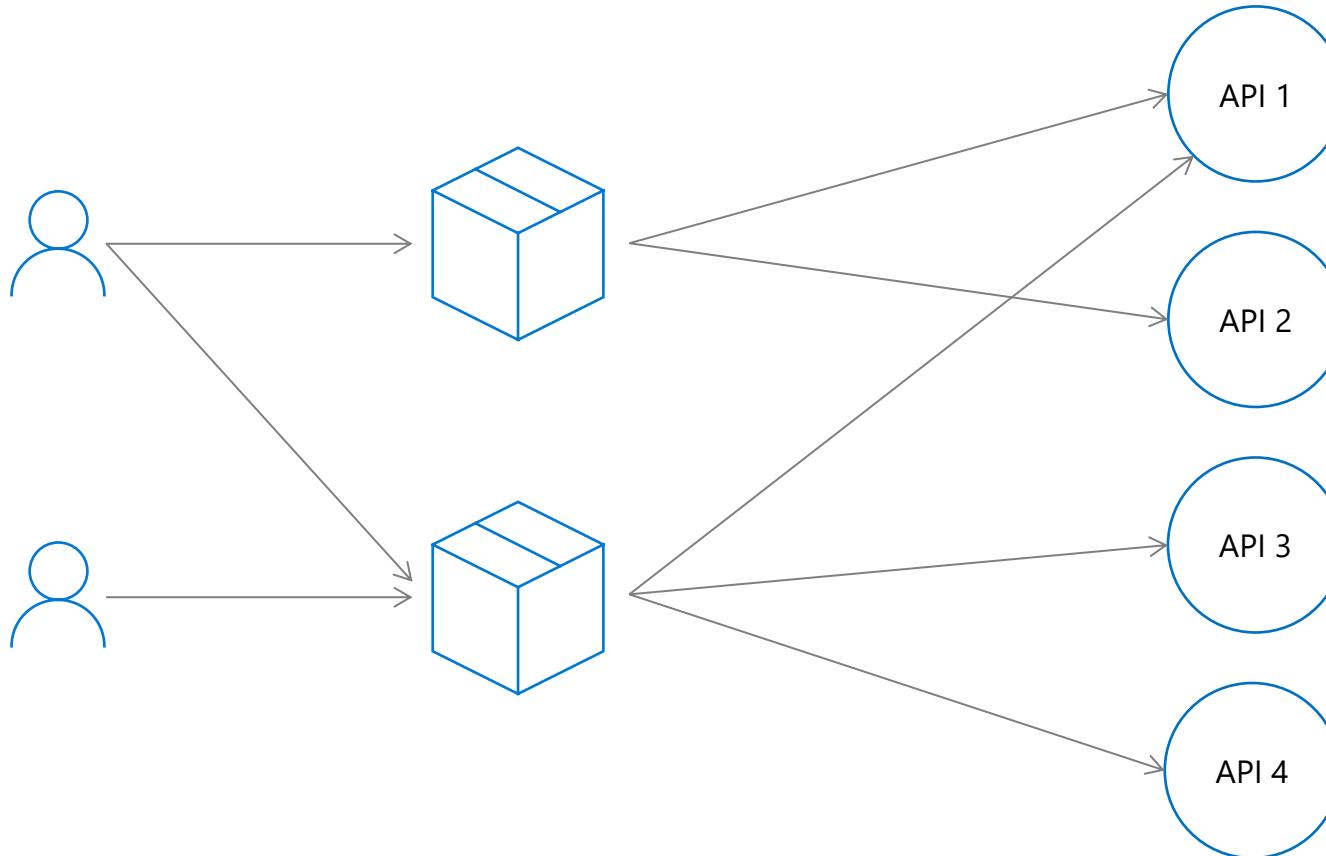
Consumers choose when to adopt

Specify with URL path, query, or header param

Versions and revisions

Domain	API	Versions	Operations	Revisions	
<code>https://example.org/</code>	<code>foo</code>	<code>/v1</code> <code>/speakers</code> <code>/sessions</code> <code>/days</code> <code>/v2</code>	<code>online</code>	<code>;rev=1</code> <code>;rev=2</code> <code>;rev=3</code> <code>;rev=4</code>	<code>/events</code>
			<code>offline</code>	<code>;rev=1</code> <code>;rev=2</code>	<code>/speakers</code>
			<code>current</code>		<code>/sessions</code> <code>/venues</code>

Bundle APIs with products



Developer portal

Browse products and associated APIs
Subscribe to products
Manage subscriptions and keys



Management plane

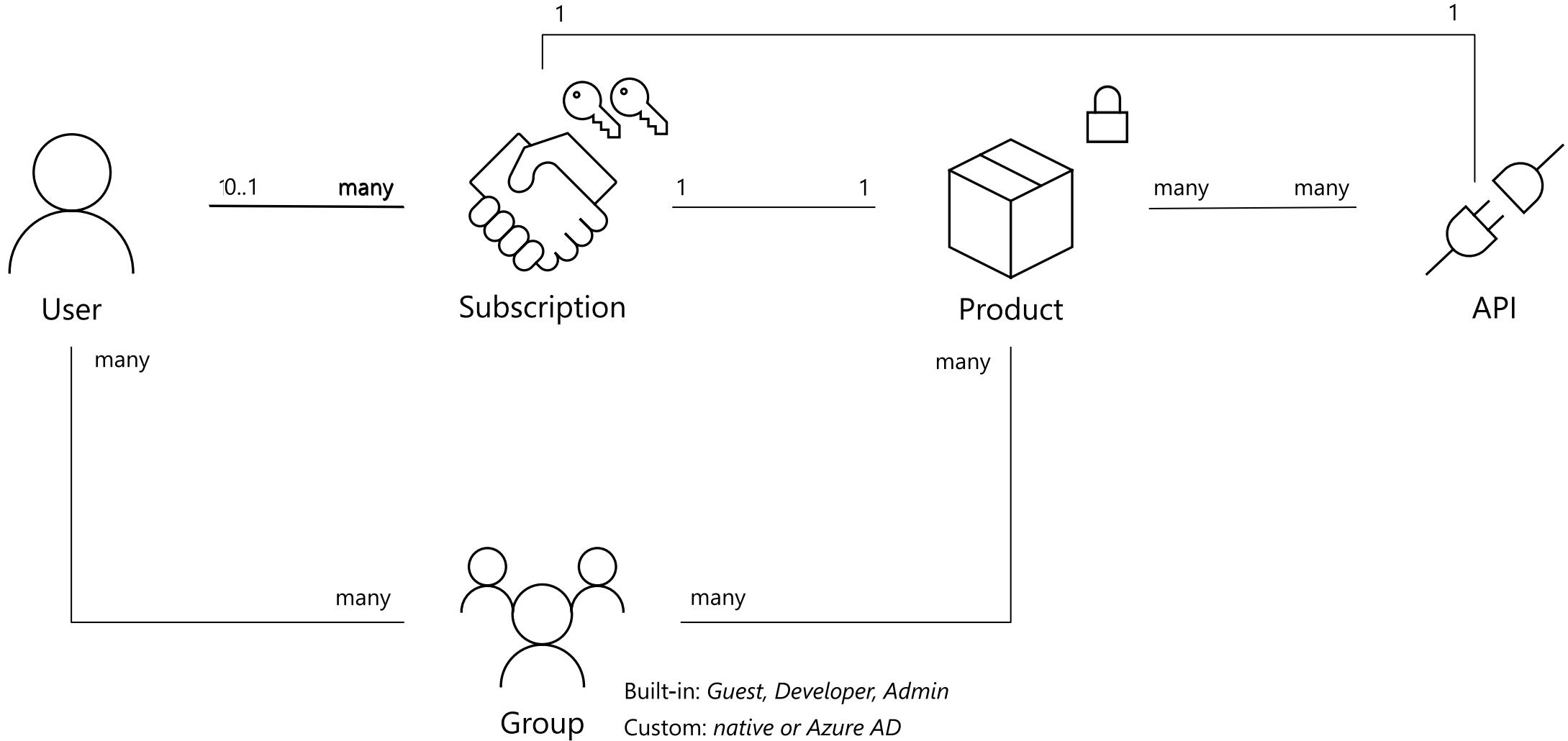
Manage products and API associations
Define product-scoped policies
Approve and manage subscriptions
Collect and analyze usage data
Monetize access



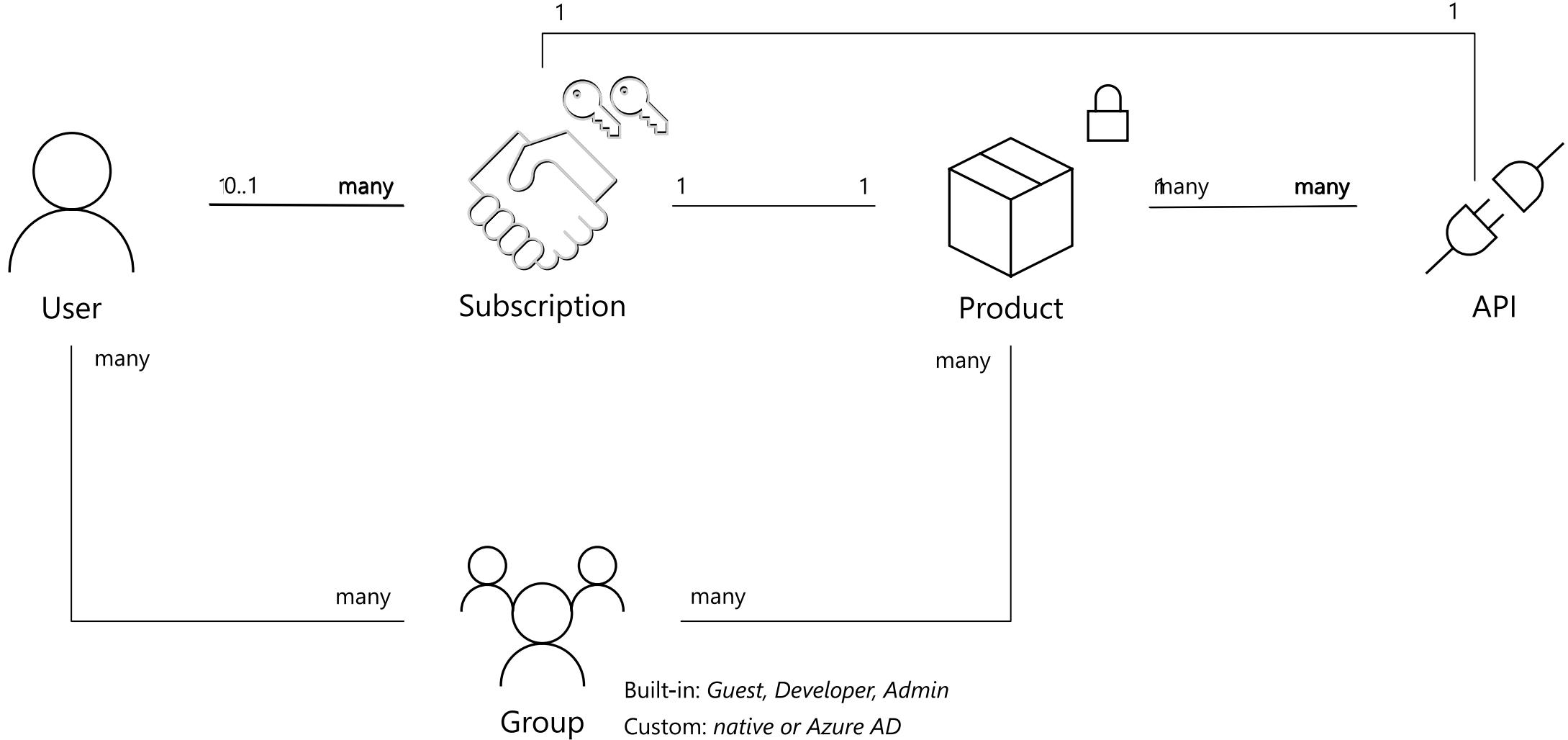
Gateway

Authenticate API requests with keys
Execute product-scoped policies

Users, groups, products, APIs, and subscriptions



Products not requiring subscriptions



API Portal

Standalone modern API documentation portal

Customize it through a drag-and-drop, no-code visual editor

Contains REST API reference pages, code samples, and interactive console

Relies on the same technology as the Azure API Management's developer portal

GitHub-based API ecosystem for communication and collaboration

Track source code changes

Automate portal deployments with GitHub Actions

Host the site for free with GitHub Pages

Sample use cases

Enterprise-wide API catalog for discoverability, deduplication of assets, and business efficiency

Branded API documentation portal for partners or external consumers for discoverability and self-onboarding

<https://aka.ms/ApiPortal>

Monetization

Support for common monetization models

Subscriptions with call quotas

Per call fee

Pre-paid calls with overages

API Management collects the data to support these models

Subscription billing – list of active subscriptions in a billing period

Metered billing - # of requests per subscription in the billing period

Customers are responsible for integration with payment providers

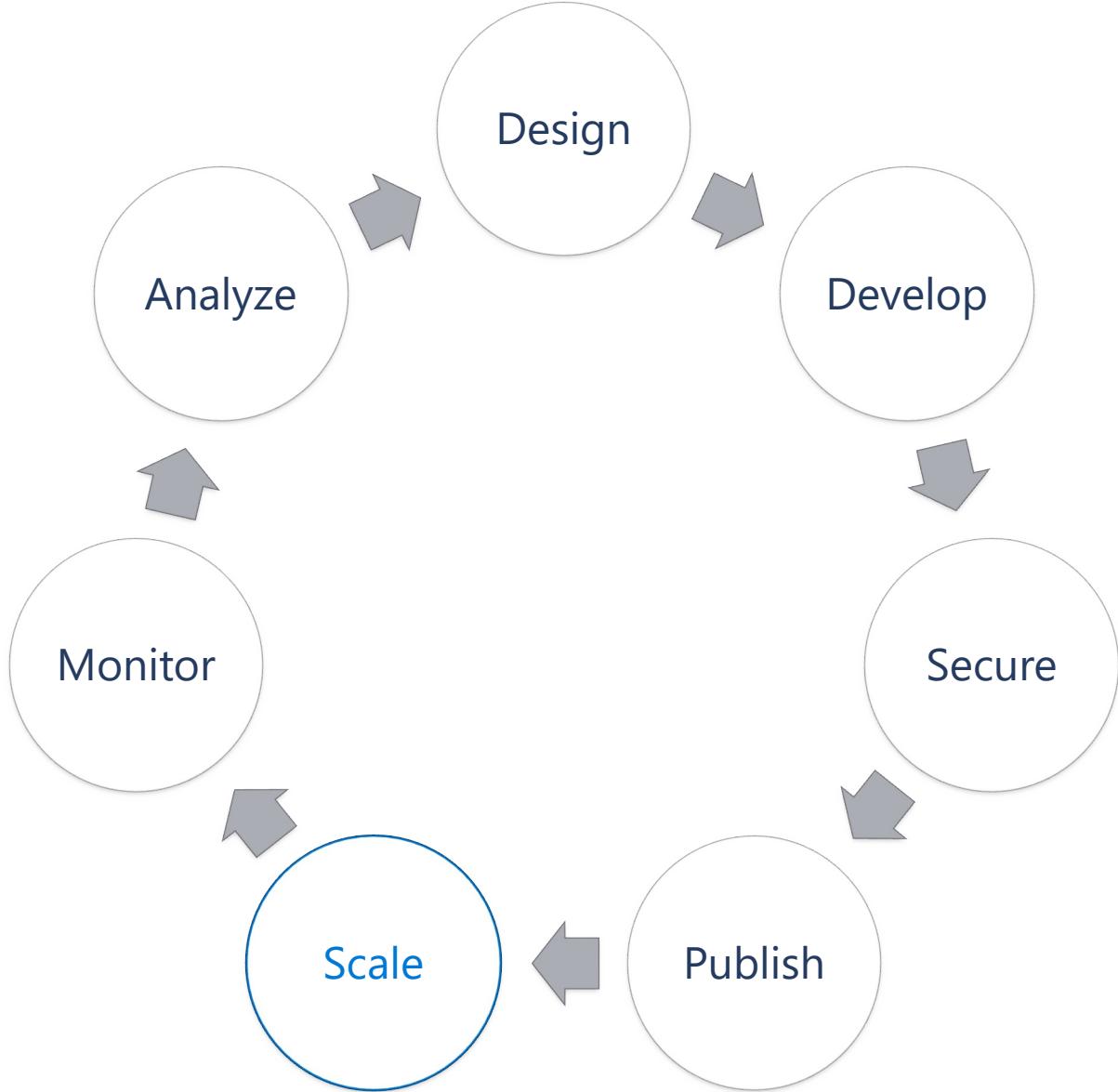
[Sample solution](#) for Stripe and Ayden

Integration mechanisms

Subscription delegation on the developer portal

Management API

API life cycle: scale



Worldwide presence

44 public regions in Americas, Europe, Asia, Australia, Africa

6 US Government regions

4 regions in China

[Browse all available regions](https://azure.microsoft.com/regions/) on azure.microsoft.com

Higher availability with multi-region feature

Improved availability of the data plane – 99.99% vs 99.95% SLA

Reduced latency of API calls

Single Premium instance can be scaled across multiple regions

- Additional units can be deployed into the Primary or other Secondary region

- Regions can have a different number of units

- Regions and units come at an additional cost

Primary region hosts all the components

- Gateway, developer portal, management API, ...

- Developer portal and management API are inaccessible if Primary region becomes unavailable

Secondary region hosts gateway only

- Secondaries can operate on a last received configuration while the Primary region is unavailable

- They periodically try to reconnect and catch up

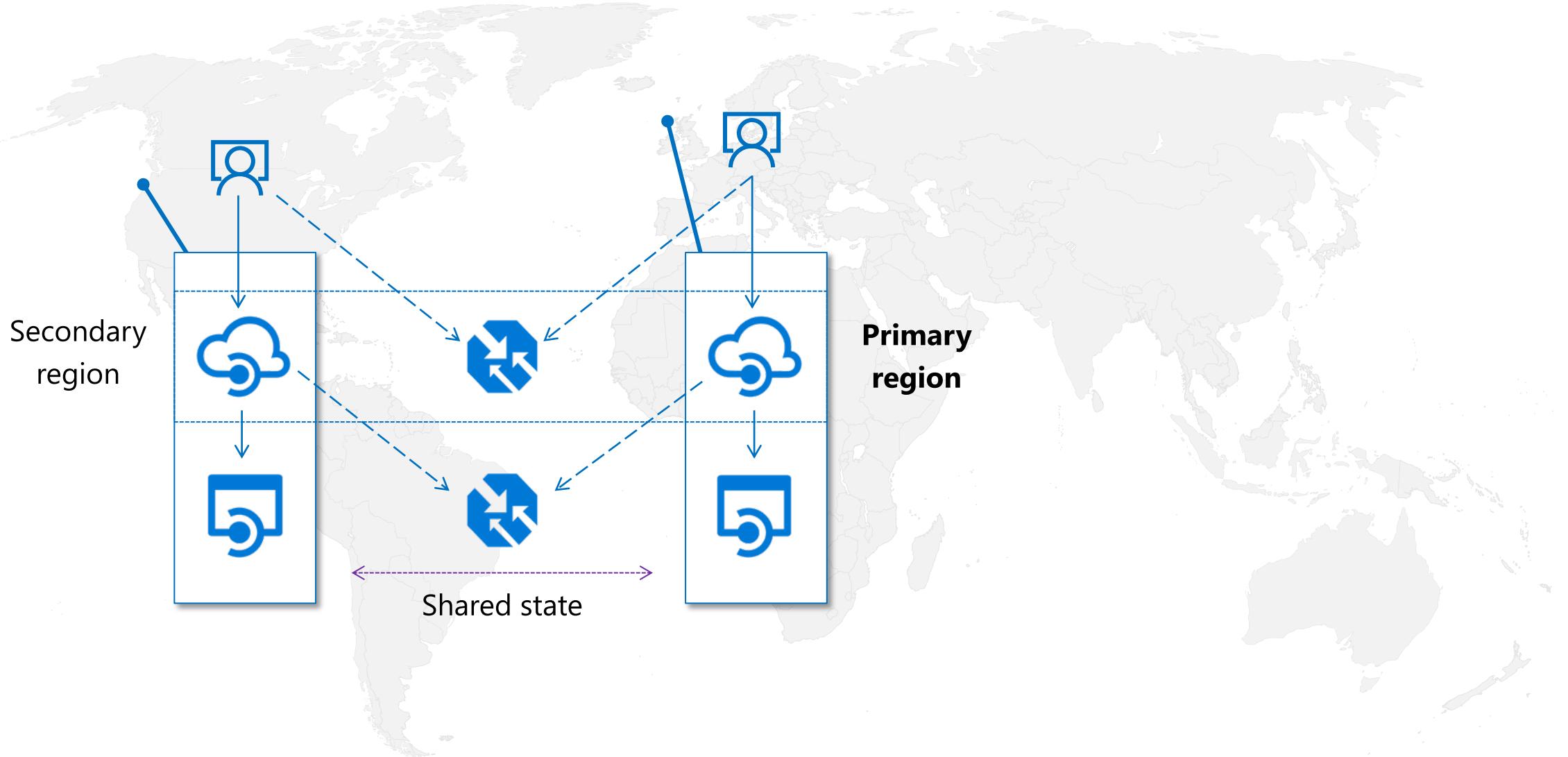
All APIs are available in every region

Requests are routed to the closest available region by Azure Traffic Manager

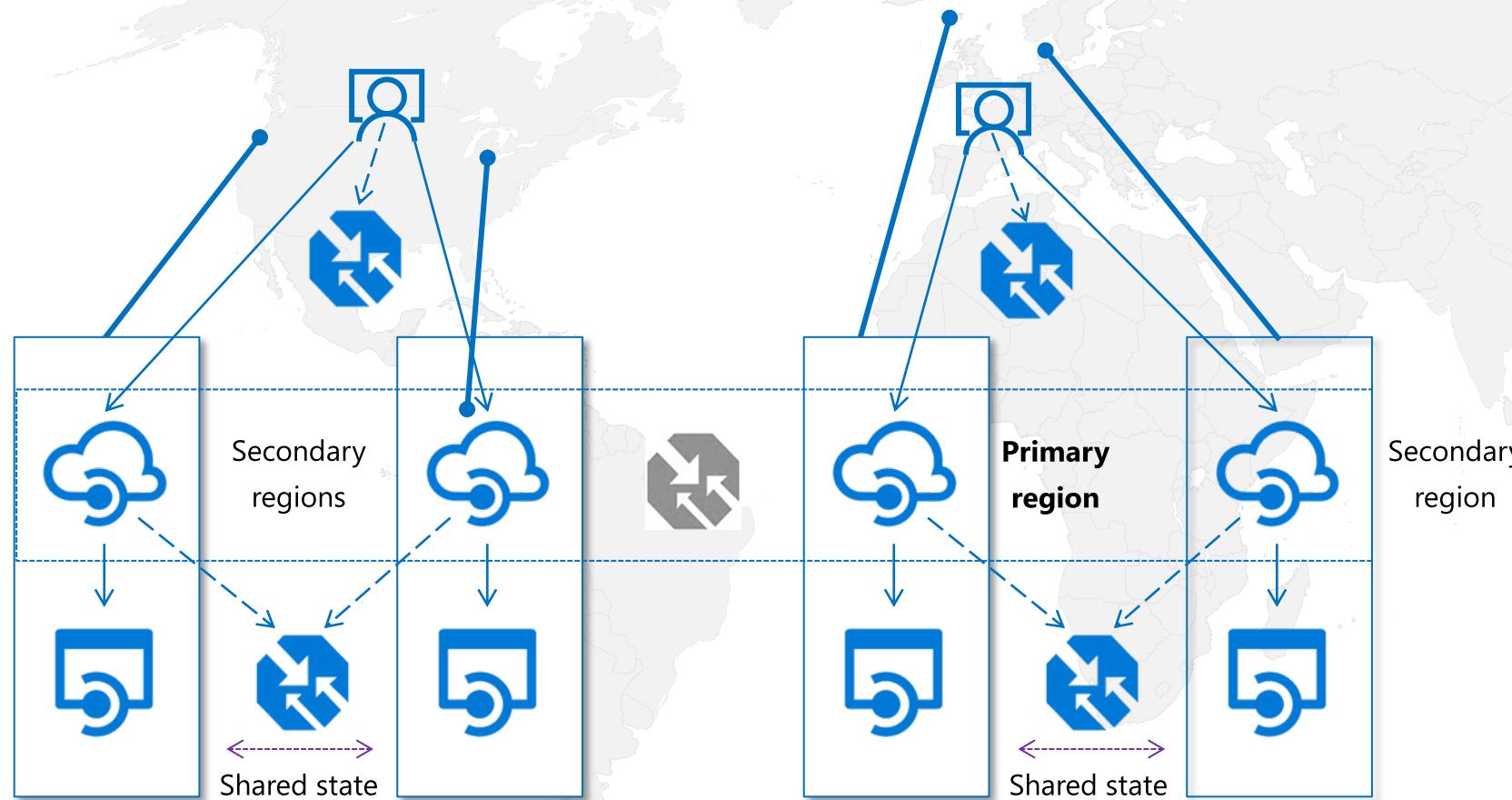
- Uses [Traffic Manager's performance routing](#) with 5min TTL

Regional endpoints enable custom traffic management, for example for data sovereignty

Default multi-region topology



Custom multi-region topology



Availability Zones

Obtain 99.99% SLA with two (or more) zones in a single region

Improve resiliency of the primary region in a multi-region deployment

Each unit contains all API Management components

Units must be evenly distributed across zones

Available in the Premium tier in every AZ-enabled Azure region

Self-hosted API gateway



Deployable to on-premises or cloud

Functionally equivalent to the managed gateway
Packaged as a Linux-based Docker container image
Available from the Microsoft Container Registry



Managed and observed from Azure

Requires only outgoing connectivity to Azure on port 443
Connects to a “parent” API Management service
Pulls down configuration and pushes up telemetry



Simple to provision and operate

Just a single container
Easy to evaluate on a laptop with Docker Desktop or Minikube
Kubernetes provides availability, scaling, rolling upgrades, and more

Self-hosted gateway pricing

Developer tier

Pre-production environments

Unlimited gateway locations

Single node per location

No additional charge

Premium tier

Production environments

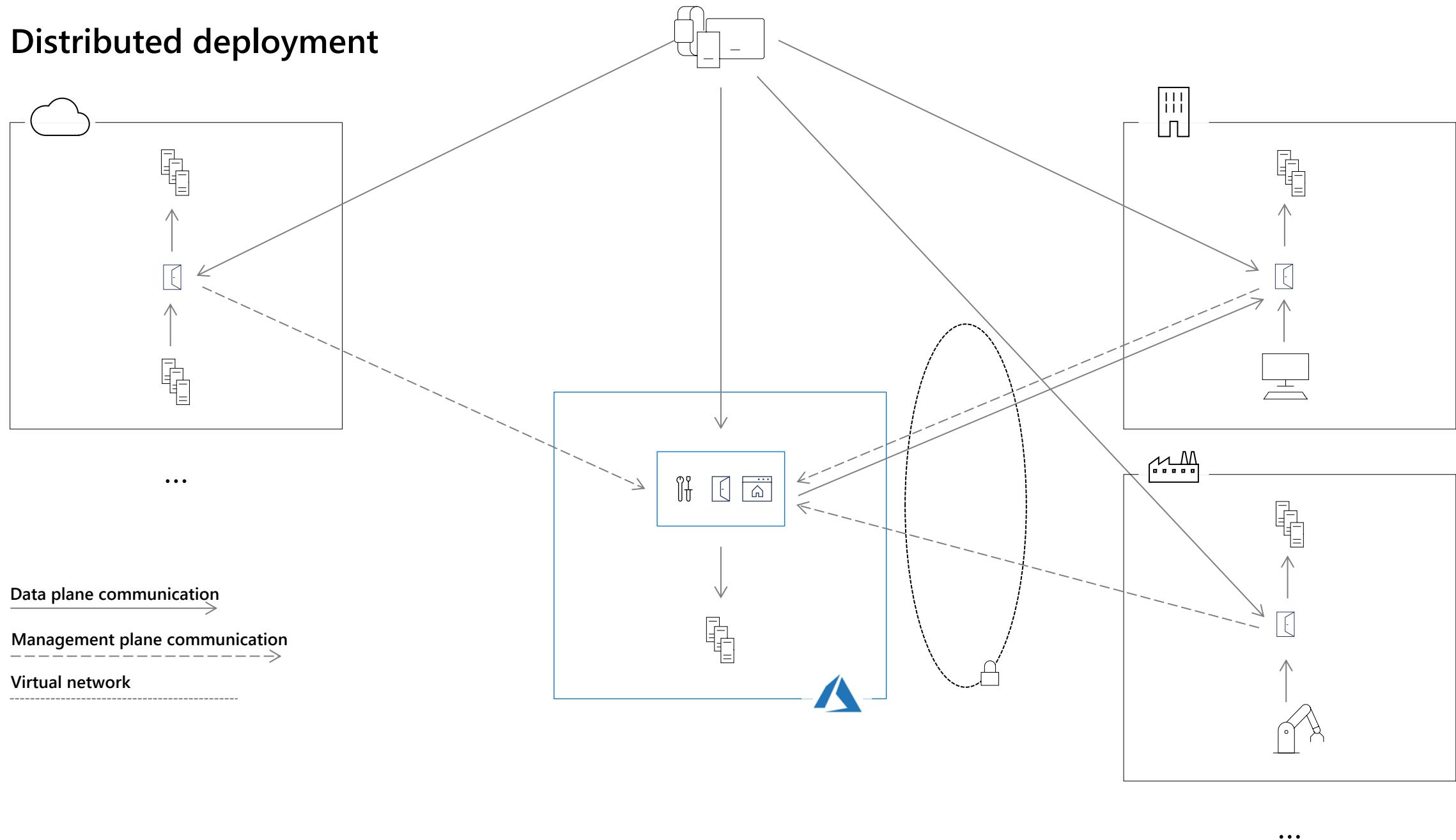
Unlimited gateway locations

Unlimited nodes per location

Paid add-on

Nodes in a gateway location share configuration – e.g., APIs, domain names, certificates

Distributed deployment

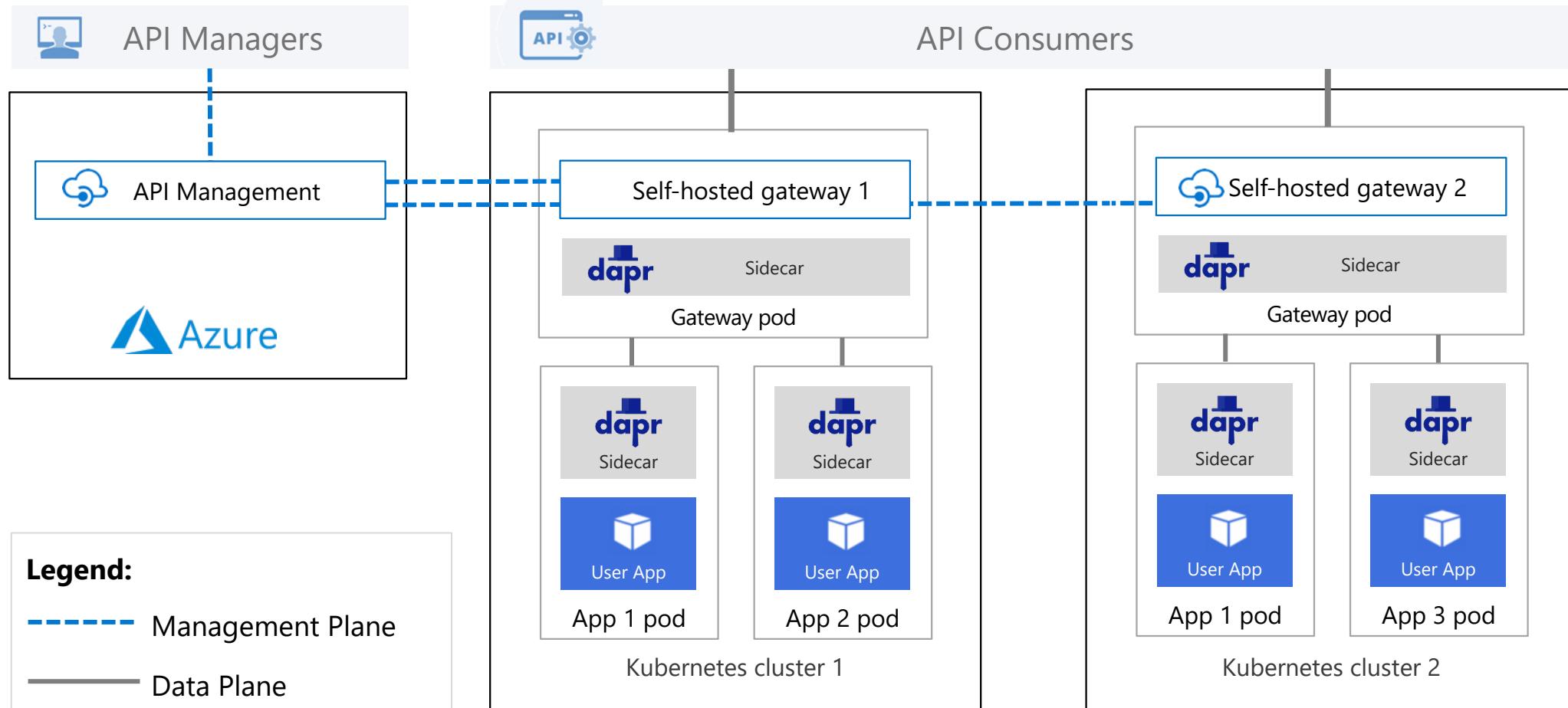


Dapr integration policies

Invoke a service

| Send a message to a pub/sub topic

| Trigger an outbound binding



Isolated SKU

Same capabilities as the Premium SKU

Ensures [compute isolation](#)

Meets US Department of Defense IL5 [requirements](#)

In Public Preview

Price TBA, contact support to provision

	Consumption	Developer	Basic	Standard	Premium	Isolated ^{Preview}
Purpose	<u>Lightweight</u> and serverless version of API Management service, billed per execution	Non-production use cases and evaluations	Entry-level production use cases	Medium-volume production use cases	High-volume or enterprise production use cases	Enterprise production use cases requiring high degree of isolation

Backup and restore for disaster recovery

Backup

Usually takes around 10 min

Captures everything but reports and custom domain settings in a blob

Service configuration operations (e.g., scaling, upgrades) are blocked while backup is in progress

Changes applied after backup starts are not included in the backup

Restore

Could take as long as 30 min or more depending on the size

Instance is not available while restore is in progress

Custom domain configuration need to be re-applied manually

Standby failover instance can reduce RTO

Create backup instance in a different region in advance

Configure custom domain identically to the active instance

Sync configuration with the active instance periodically to achieve desired RPO

To fail over update the CNAME to reference backup instance

Scale up if and as required

Troubleshooting and support

SLA

99.95% in all tiers

99.99% in the Premium tier with multi-region configured

Self-troubleshoot

Built-in automated troubleshooting experiences in the Azure portal

[Extensive documentation on Azure Docs](#)

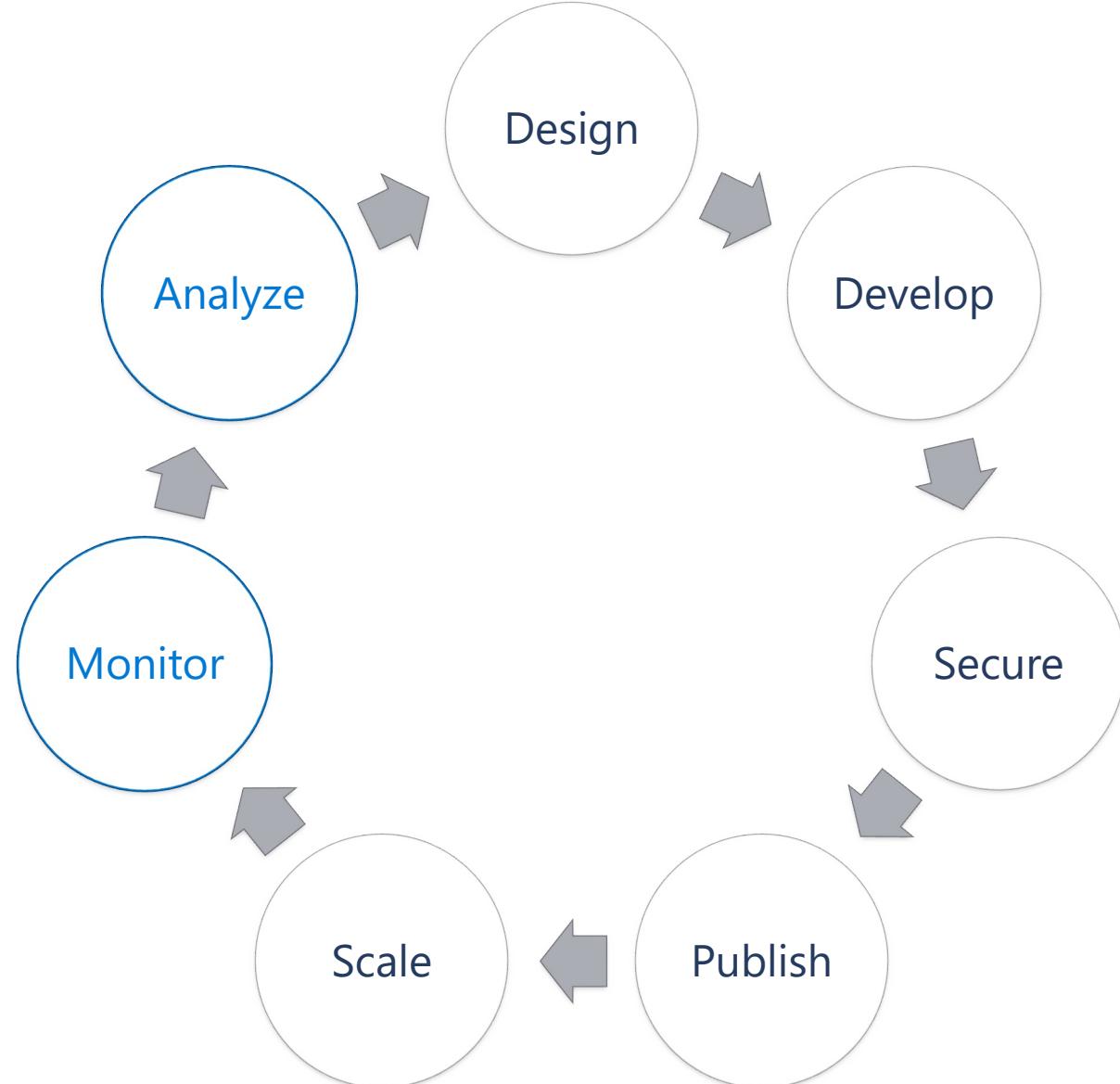
Supported by Azure Support

Requires support plan

Available worldwide in nine languages: English, Spanish, French, German, Italian, Portuguese, Traditional Chinese, Korean, and Japanese

24x7 in English for severity A and B and in Japanese for severity A

API life cycle: monitor & analyze



Monitor and analyze features

Tech	Reporting	Monitoring	Debugging	Data lag	Retention	Sampling	Data schema	Data kind	Enabled
API inspector	-	-	Good	Instant	Last 100 traces	Turned on per request	Fixed can be extended	Request trace	Always
Built-in reports	Good	-	-	Minutes	Unspecified	100%	Fixed	Reports Logs via API	Always
Azure Monitor Metrics	Basic	Good	-	Minutes	93 days export to extend	100%	Fixed	Metrics	Always
Azure Monitor Logs	Good	Good	Good	Minutes	31 day (5GB) upgrade to extend	100% adjustable	Fixed can be extended	Logs	Optional
Application Insights	Good	Good	Good	Seconds	90 days (5GB) upgrade to extend	Custom	Choice of presets can be extended	Logs, metrics	Optional
Log to Event Hub	Custom	Custom	Custom	Seconds	User managed	Custom	Custom	Logs	Optional

API Inspector

Request scoped trace

Turned on per request

Fixed schema (can be extended)

```
    "traceId": "379249f9-577d-47b4-9c19-30954fa6d5ce",
    "traceEntries": {
      "inbound": [ ... ], // 3 items
      "backend": [
        {
          "source": "forward-request",
          "timestamp": "2020-05-21T02:56:27.5664235Z",
          "elapsed": "00:00:00.0055591",
          "data": { ... } // 2 items
        },
        {
          "source": "forward-request",
          "timestamp": "2020-05-21T02:56:27.6451773Z",
          "elapsed": "00:00:00.0789389",
          "data": {
            "response": {
              "status": {
                "code": 200,
                "reason": "OK"
              },
              "headers": [
                {
                  "name": "Connection",
                  "value": "keep-alive"
                },
                {
                  "name": "Access-Control-Allow-Origin",
                  "value": "*"
                },
                {
                  "name": "Access-Control-Allow-Credentials",
                  "value": "true"
                },
                {
                  "name": "Content-Length",
                  "value": "0"
                },
                {
                  "name": "Content-Type",
                  "value": "text/html; charset=utf-8"
                },
                {
                  "name": "Date",
                  "value": "Thu, 21 May 2020 02:56:27 GMT"
                },
                {
                  "name": "Server",
                  "value": "gunicorn/19.9.0"
                }
              ]
            }
          }
        }
      ]
    }
  }
```

Azure Monitor metrics

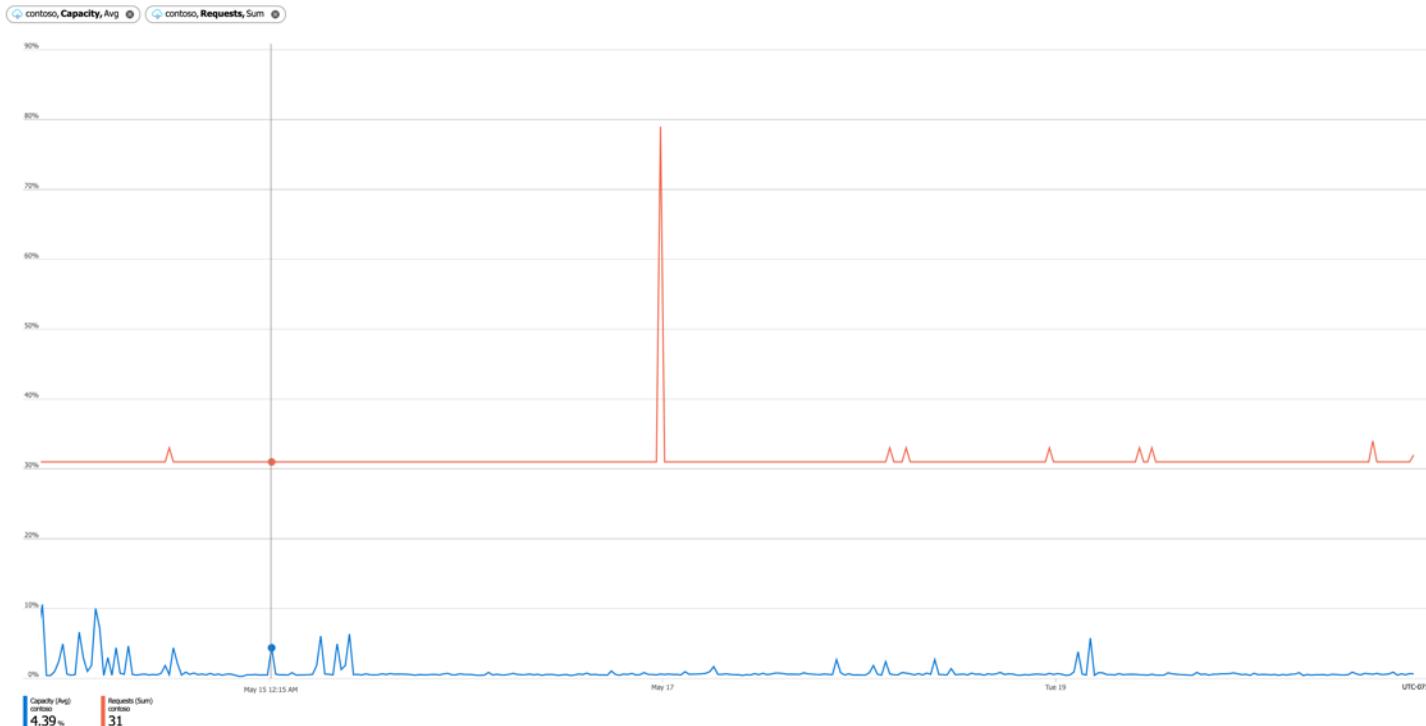
Aggregated metrics

Always-on

Samples all requests

93-day retention

Alerts and notifications



▶ Run Time range : Last 24 hours | Save Copy link + New alert rule Export

```
ApiManagementGatewayLogs  
| where BackendResponseCode != 200  
| summarize count() by bin(TimeGenerated, 1d)
```

Results Chart | Columns ▾ | Display time (UTC+00:00) ▾ Group columns

Completed. Showing results from the last 24 hours.

TimeGenerated [UTC]	count_
5/20/2020, 12:00:00.000 AM	4,270
5/21/2020, 12:00:00.000 AM	636

Azure Monitor logs

Request scoped logs

Opt-in

Adjustable sampling

Fixed schema (can be extended)

31-day retention (5GB)

Built-in query experience

Application Insights

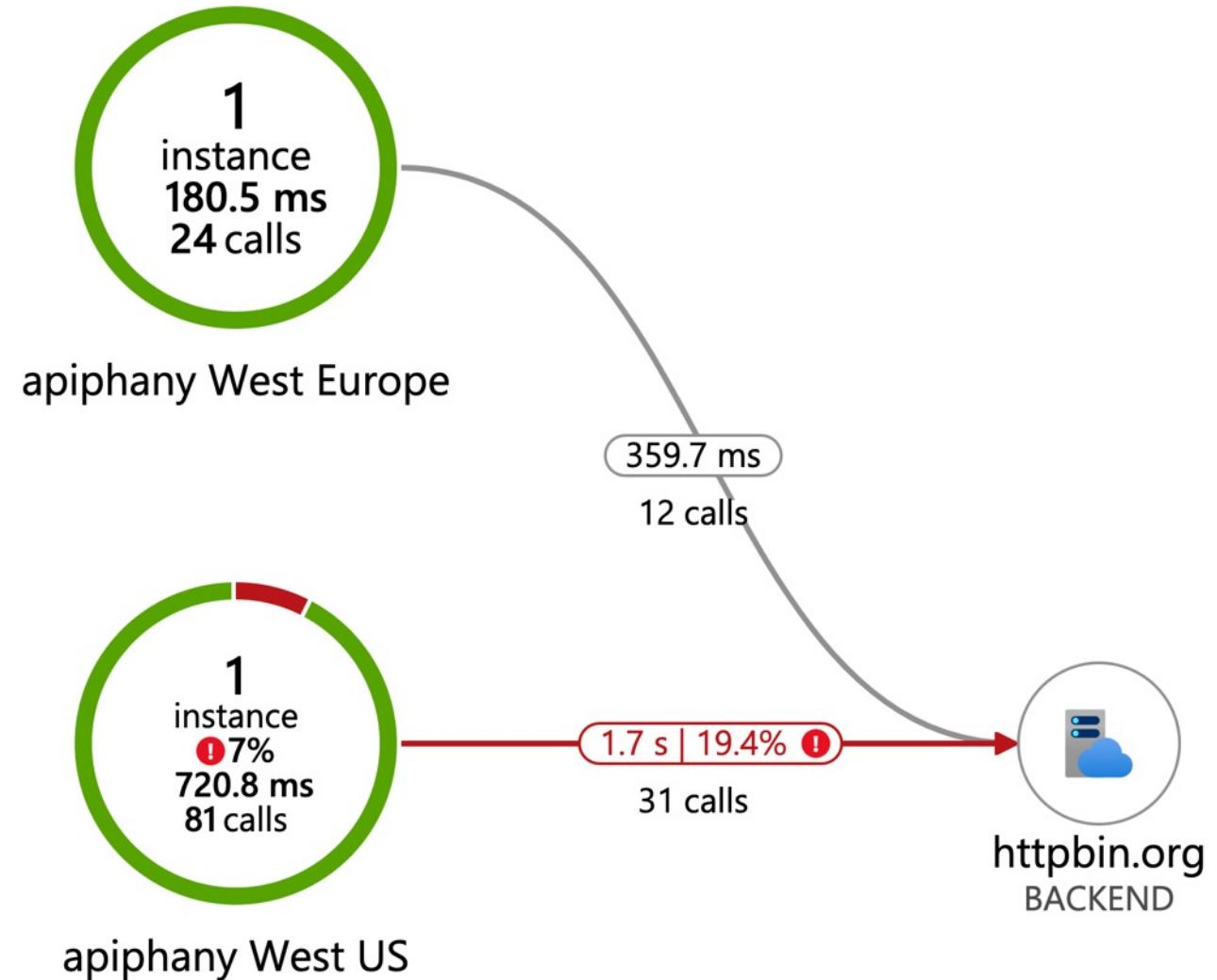
Request scoped traces

Opt-in

Adjustable sampling

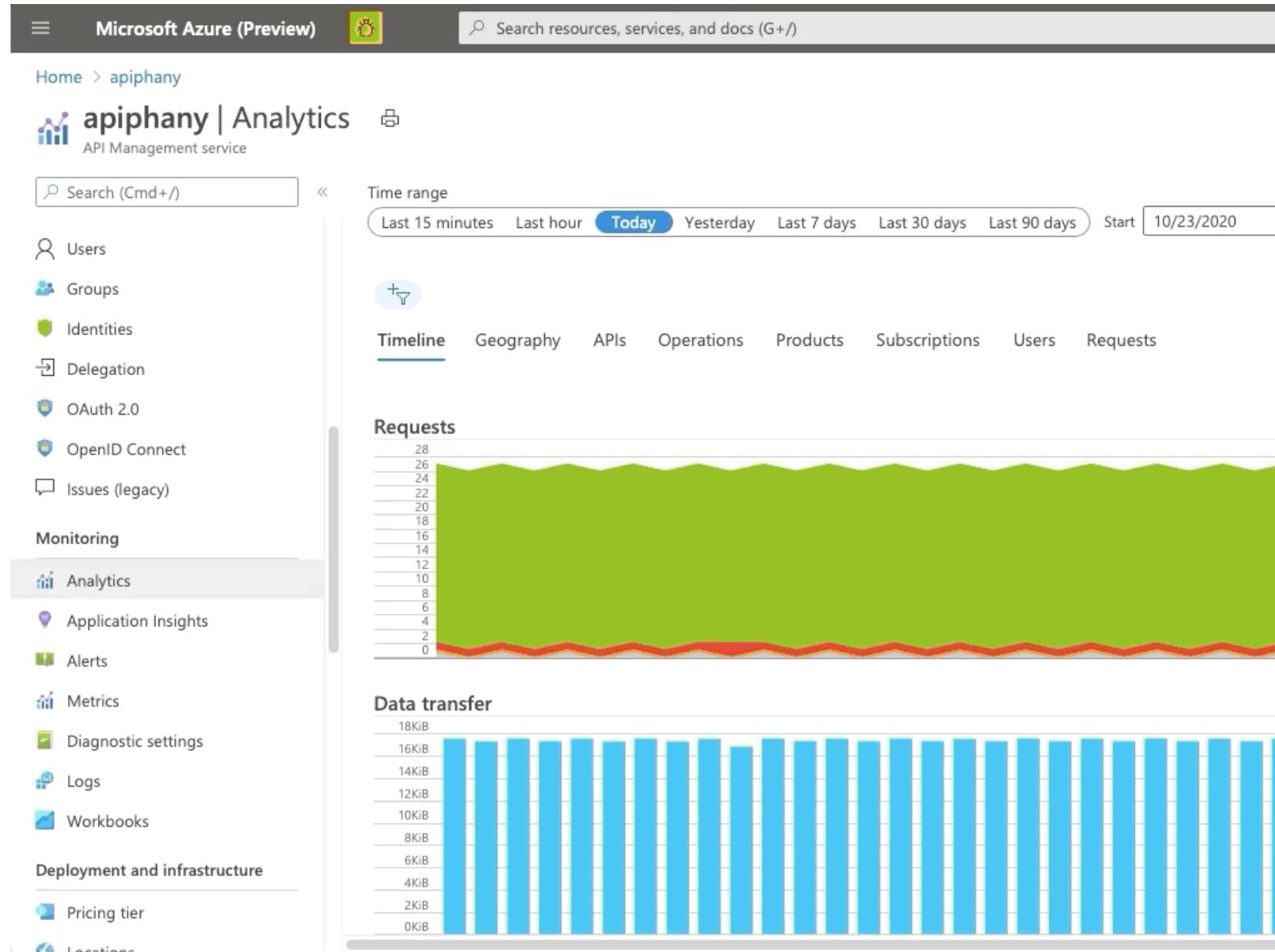
90-day retention (5GB)

Distributed tracing

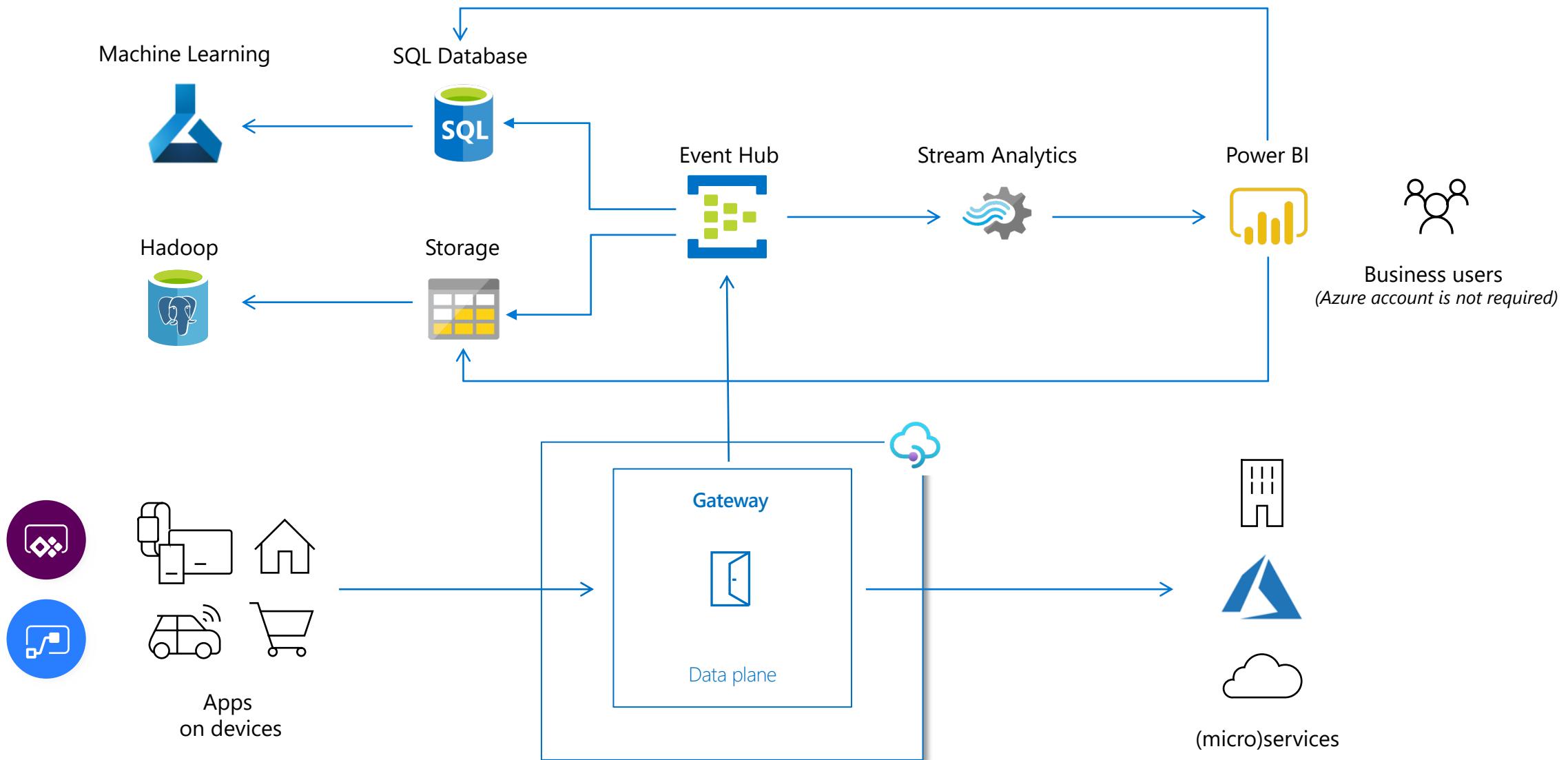


Built-in reports

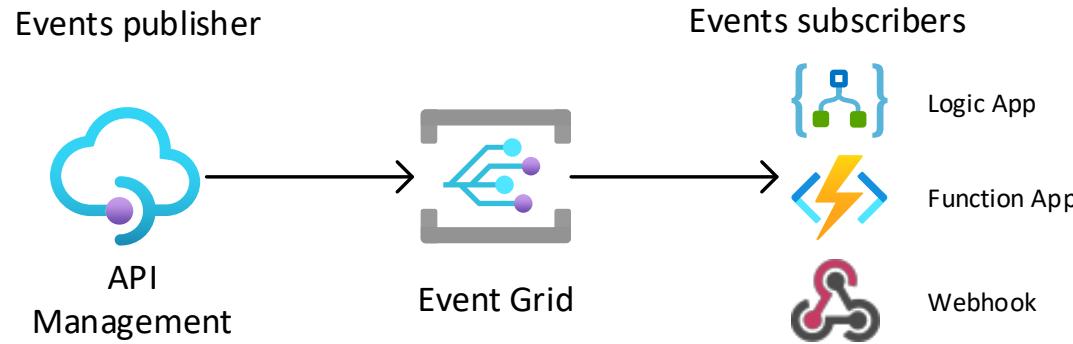
Out-of-the-box
Always-on
Rich report types
Access via Azure portal or API



Custom analytics and reporting



Event Grid integration



Integration with Event Grid

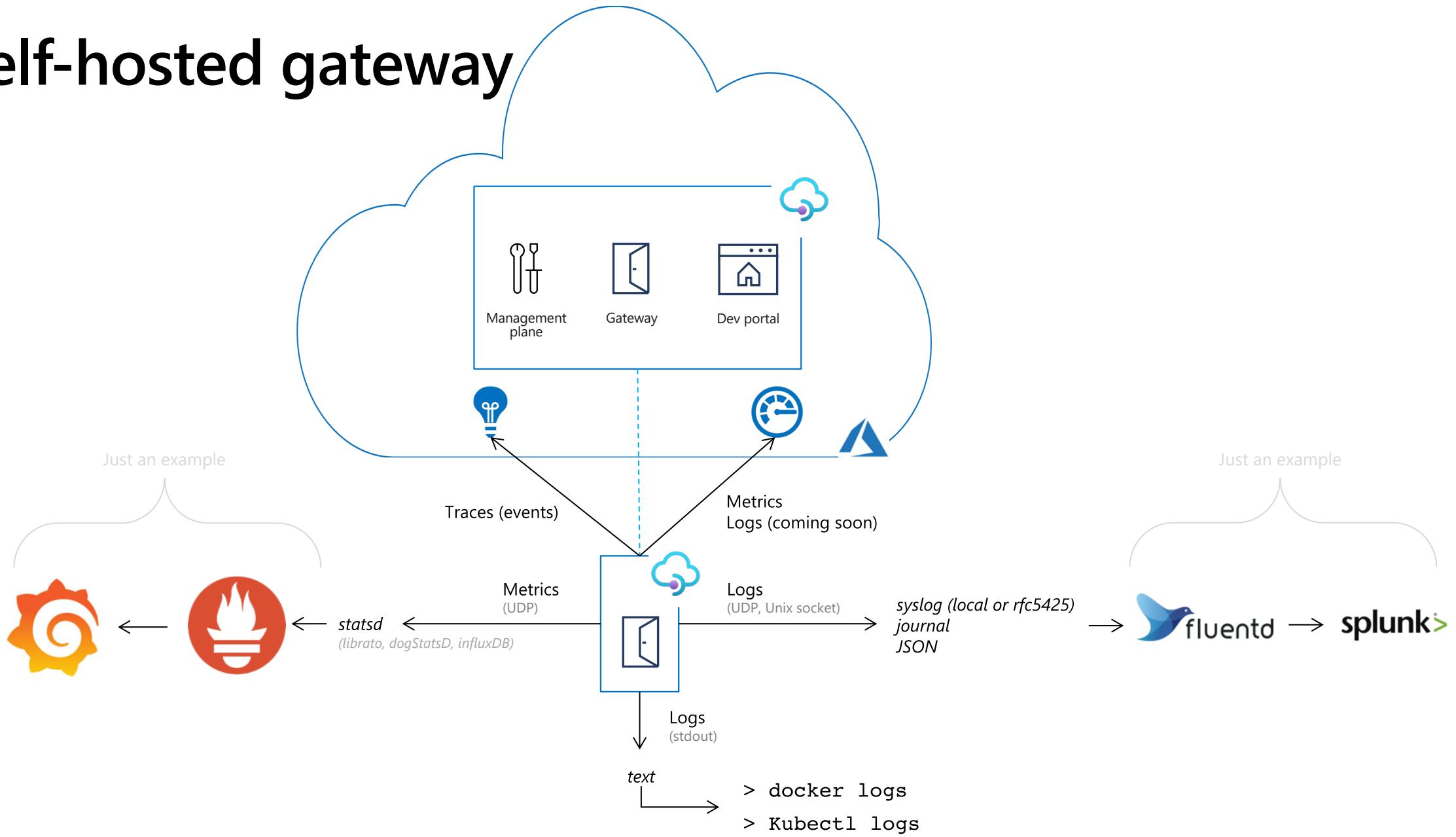
Send event notifications to Event Grid system topic of type Microsoft.ApiManagement

Trigger downstream processes on Azure Logic App, Azure Functions or via Webhook

Published events are CRUD of API, Product, Release, Subscriptions, User *

* At the time of GA (Nov. 21)

Self-hosted gateway



Azure API Management

Mature **full life cycle** API management solution

Trusted by thousands of enterprise customers

Abstract, secure, observe, and make APIs discoverable in minutes

One solution for APIs across clouds and on-premises

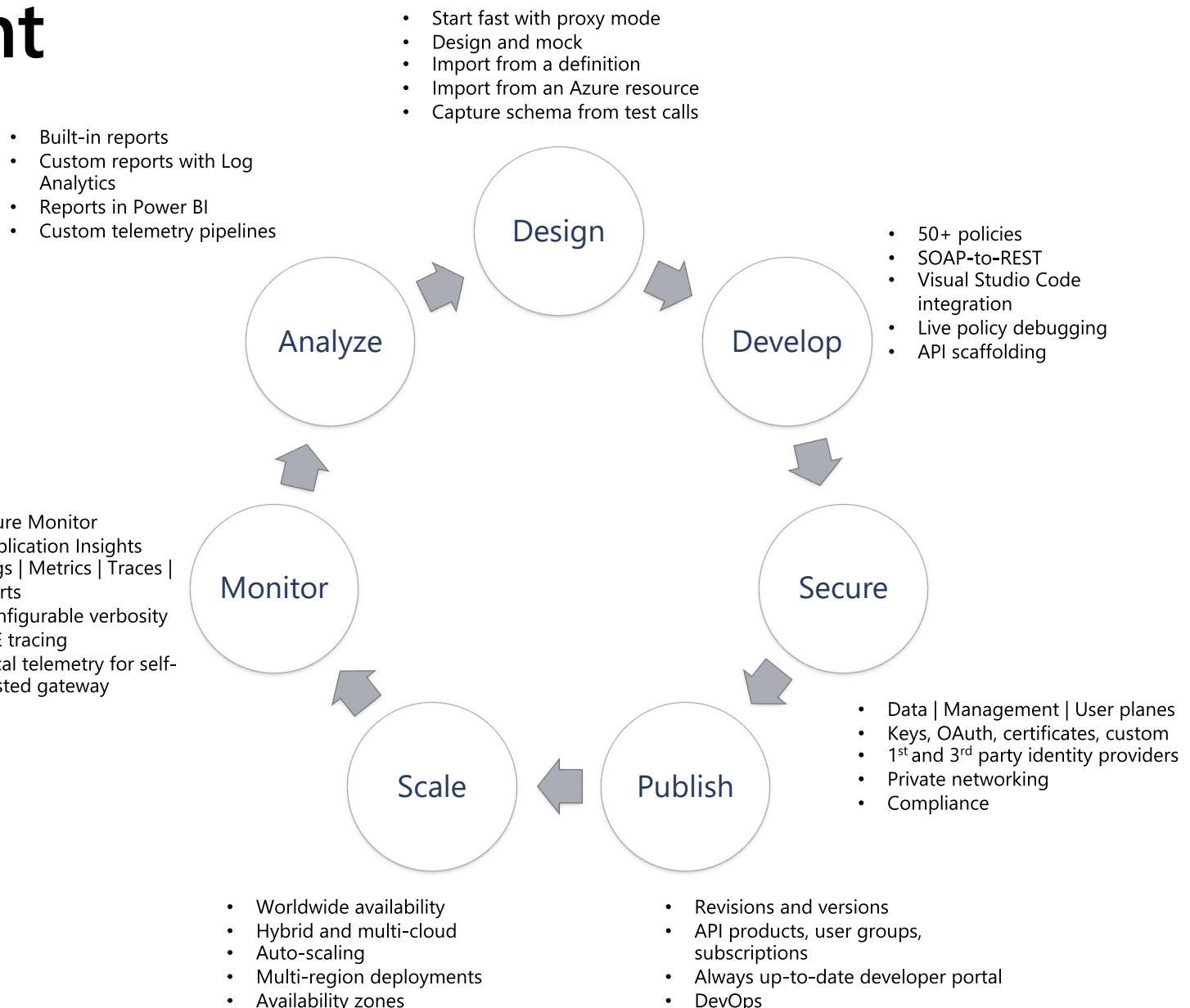
Dependable, secure, scalable, and performant

DevOps- and developer-friendly

Azure-native and integrated with other Azure services

Globally available and supported

Low-barrier-to-entry pricing



Resources

<https://aka.ms/apimlove>

Questions



