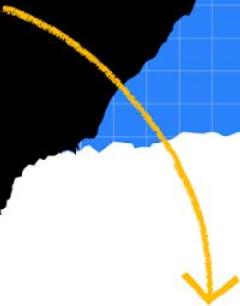


# devfest 2022

book-nav-toggle"  
dden="" fixed="" aria-label="Hide  
" Hide side navigation"



## Digital Modernization with APIfication using Google Apigee

 Google Developer Groups  
Gurugram, India



Anmol Krishan Sachdeva  
Hybrid Cloud Architect, 



**Follow Me**  
**@greatdevaks**

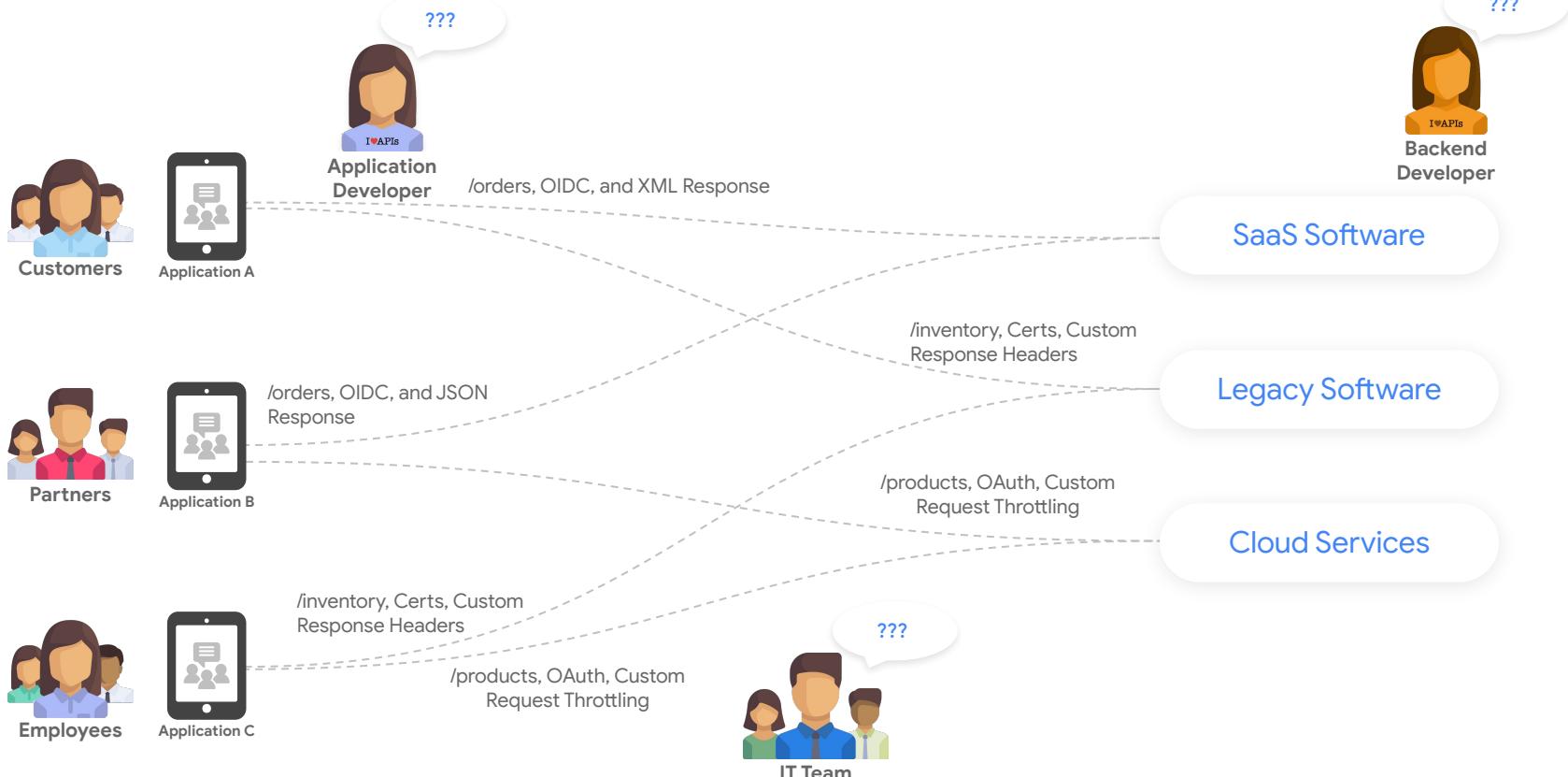


The content and views presented during the session are author's own and not of any organizations they are associated with.

## Disclaimer



# Life Before API Management

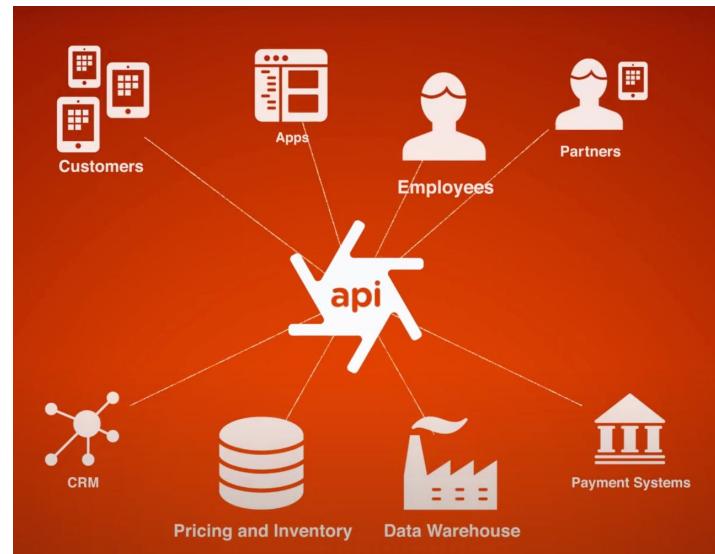




devfest 2022

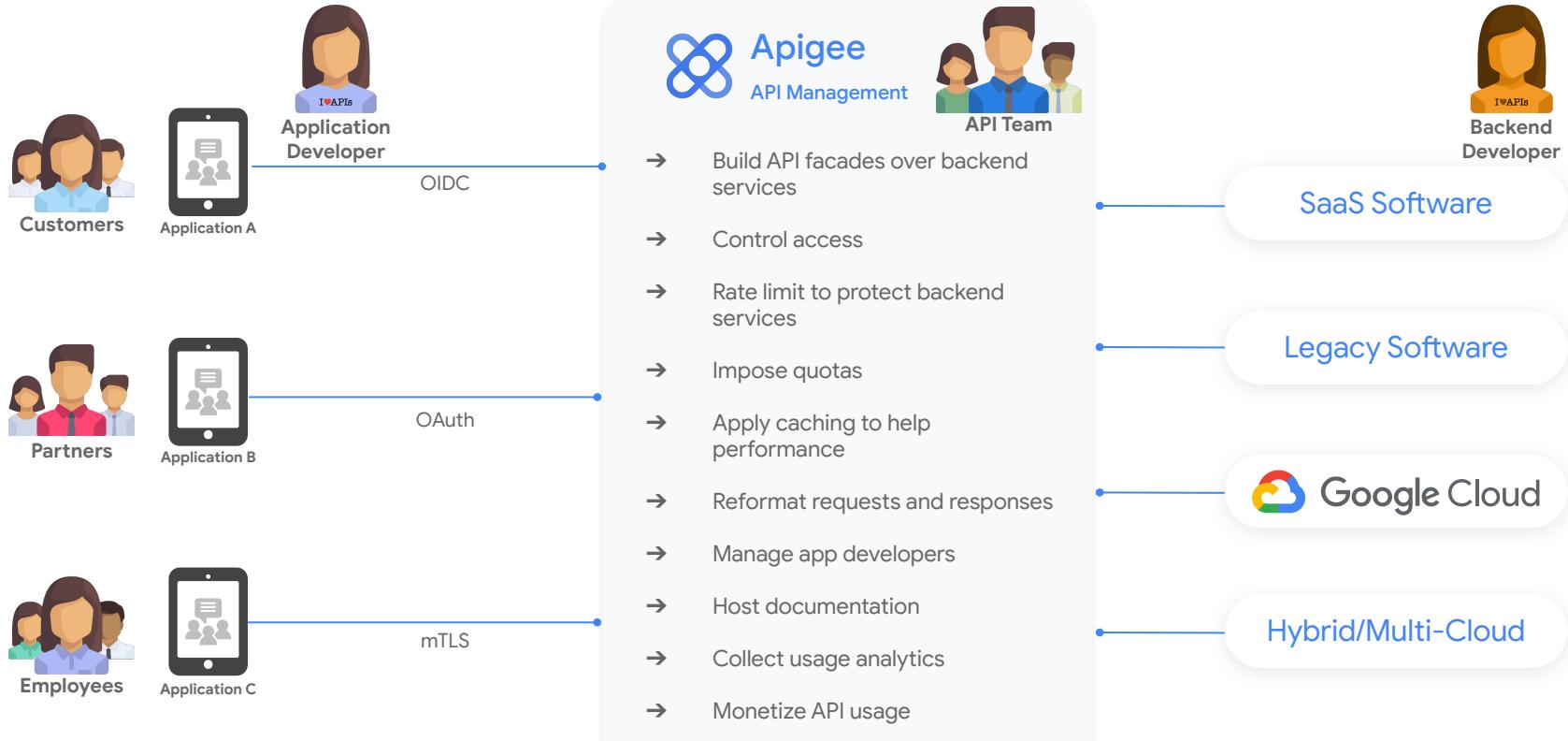


Digital Modernization with APIfication using Google Apigee  
Anmol Krishan Sachdeva (@greatdevaks) | Hybrid Cloud Architect, Google



Google Cloud |

# API First Pattern with Apigee



# Modernize with Ease



## Develop services faster

Bring new services to market faster by leveraging existing backends to build new offerings



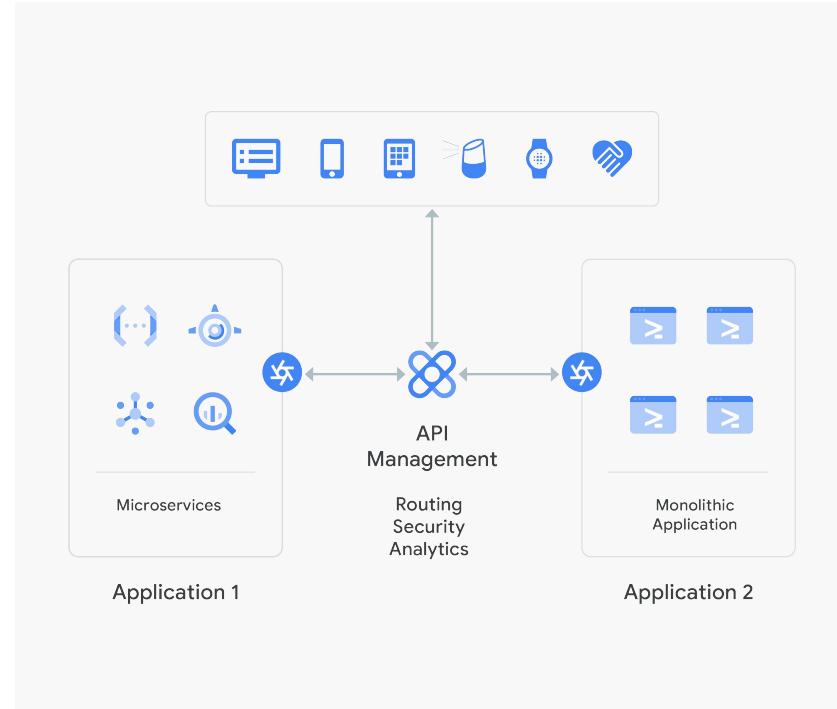
## Ensure business continuity

APIs ensure uptime and continuity by acting as a layer between backends and new services, reducing the risk of disruption



## Connect securely

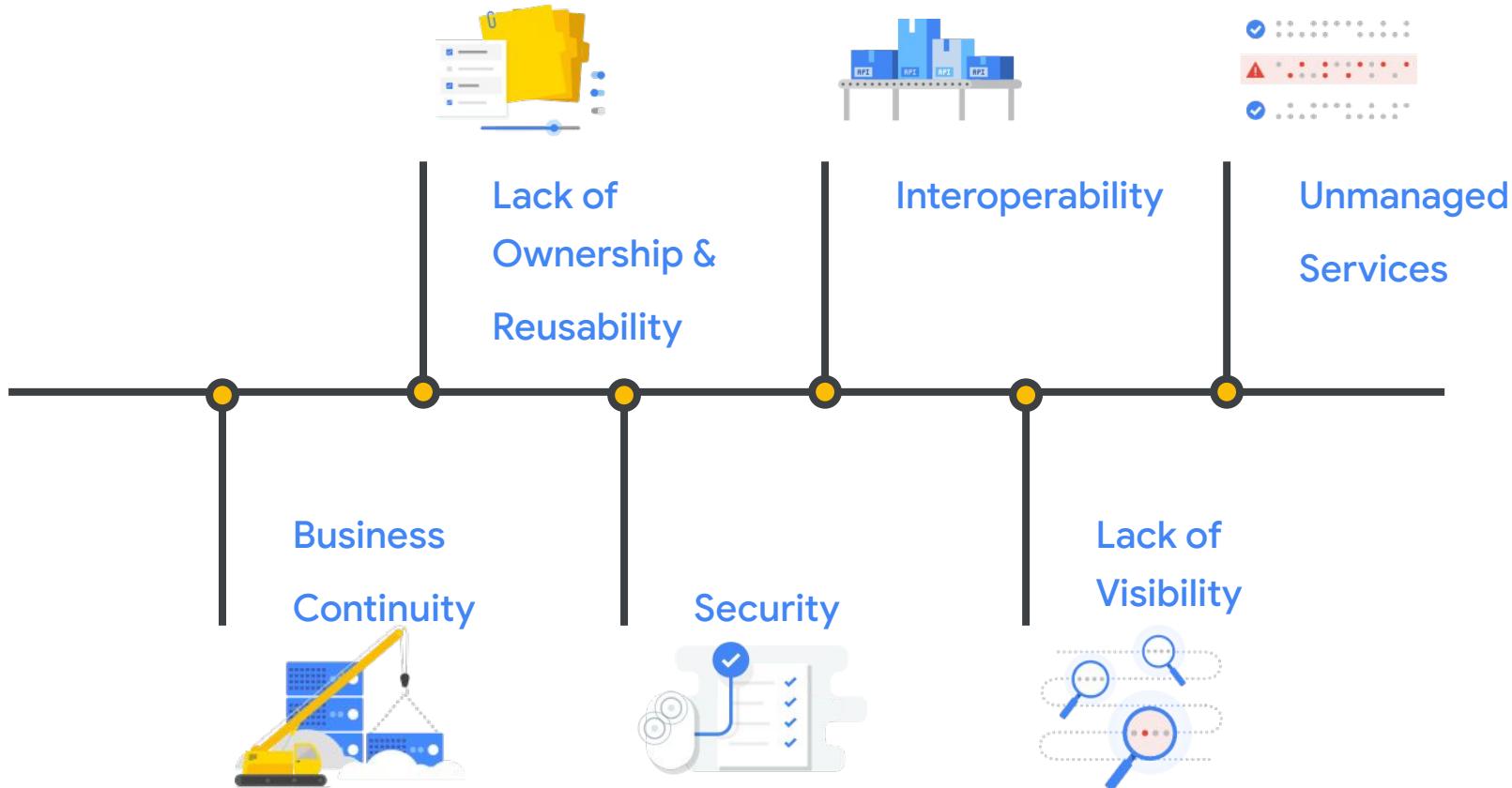
Use governance and security controls to securely expose sensitive legacy data



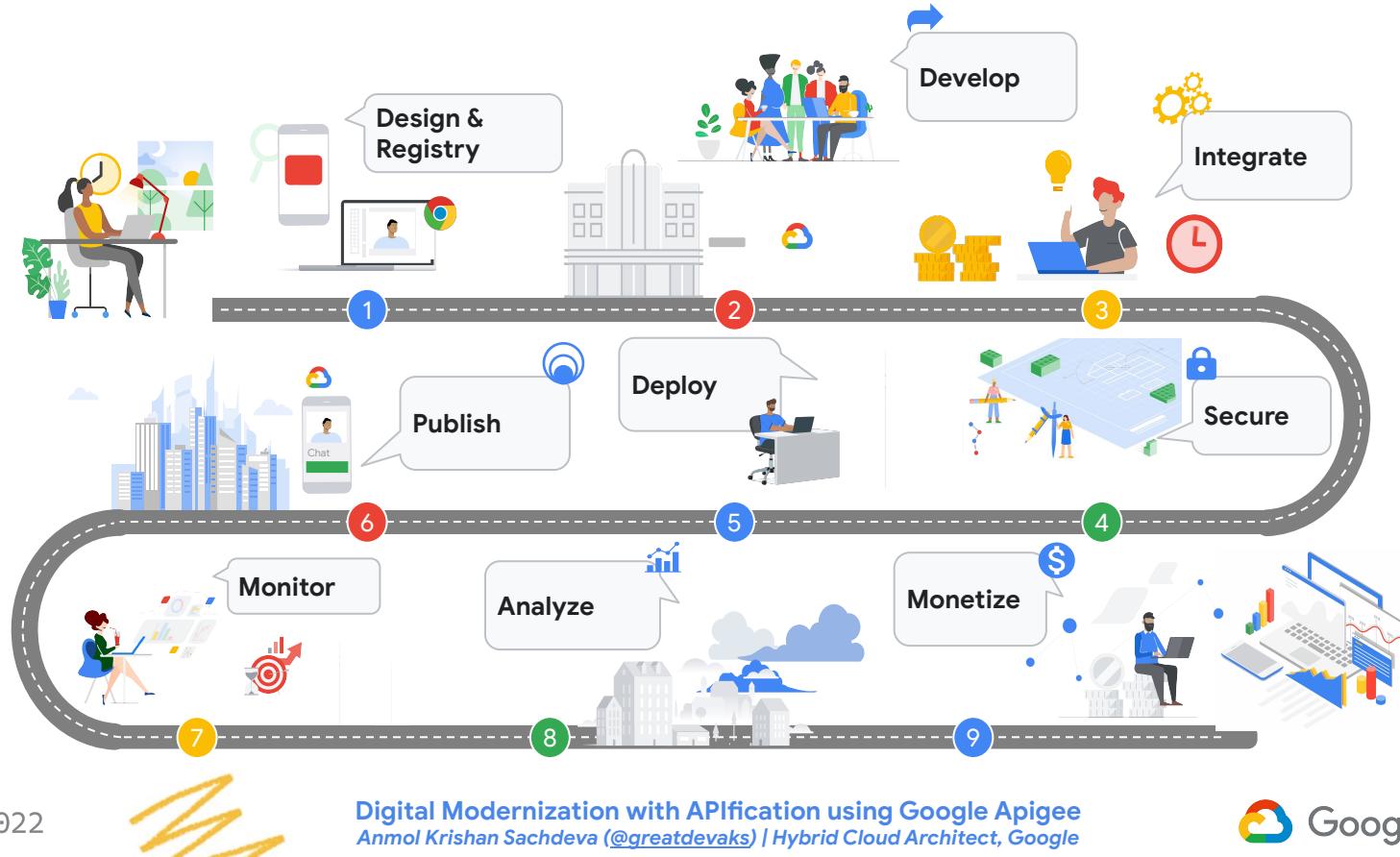
```
con class=" devsite-nav-toggle  
-haspopup="menu" hidden="" fixed="" aria-label="Hide  
navigation" data-title="Hide side navigation"  
-expanded="true">><span class="material-icons  
" data-bbox="180 10 270 100">menu</button>
```

# Key challenges that **APIfication** tries to address

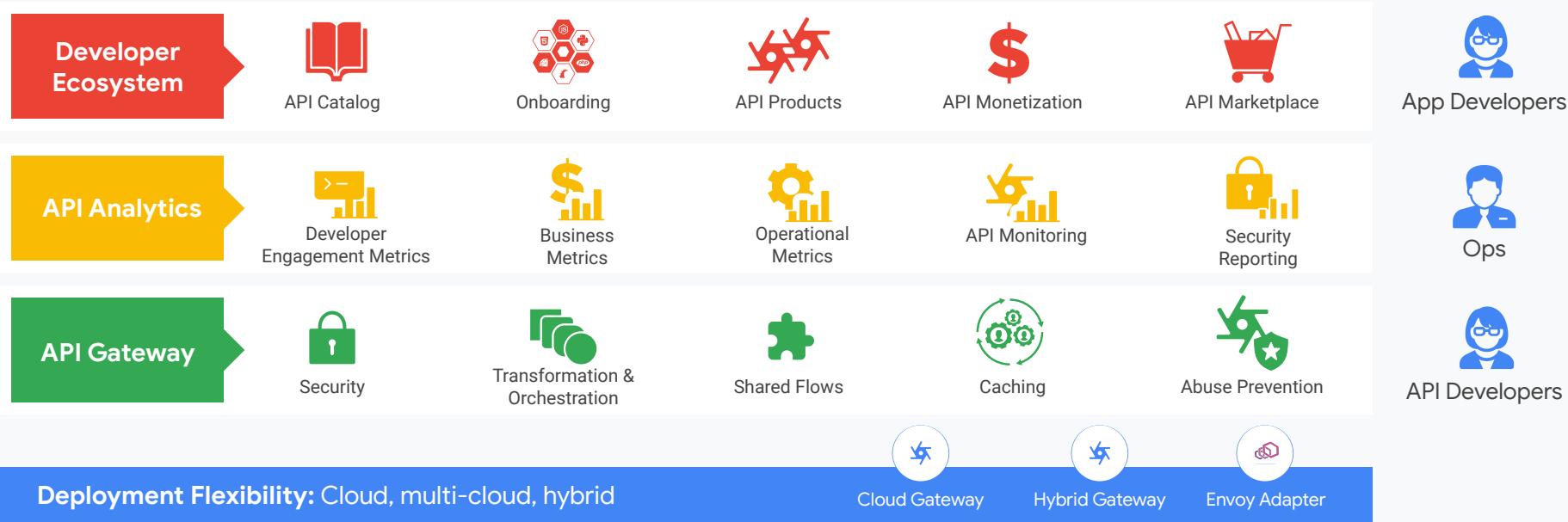




# The API Journey and Lifecycle



# The Difference



# 50+ Policies - Blazing Fast Development

**Manage** traffic and optimize API performance

TRAFFIC MANAGEMENT
Quota
Spike Arrest
Response Cache
Lookup Cache
Populate Cache
Invalidate Cache
Reset Quota

**Secure** APIs and protect back-end systems from attack

SECURITY
Basic Authentication
XML Threat Protection
JSON Threat Protection
Regular Expression Protection
OAuth v2.0
Get OAuth v2.0 Info
Set OAuth v2.0 Info
Delete OAuth v2.0 Info
Revoke OAuth v2.0
Verify API Key
Access Control
Generate SAML Assertion
Validate SAML Assertion
Generate JWT
Verify JWT
Decode JWT
Generate JWS
Verify JWS
Decode JWS
HMAC
CORS

**Transform**, translate, and reformat data for consumption

MEDIATION
JSON to XML
XML to JSON
Raise Fault
Monetization Limits Check
XSL Transform
OpenAPI Spec Validation
SOAP Message Validation
Assign Message
Extract Variables
Access Entity
Key Value Map Operations
Assert Condition
GraphQL

**Extend** with programming when you need it

EXTENSION
Java Callout
Python
JavaScript
Service Callout
Flow Callout
Message Logging
Data Capture
External Callout
Integration Callout
Set Integration Request
Trace Capture
Publish Message

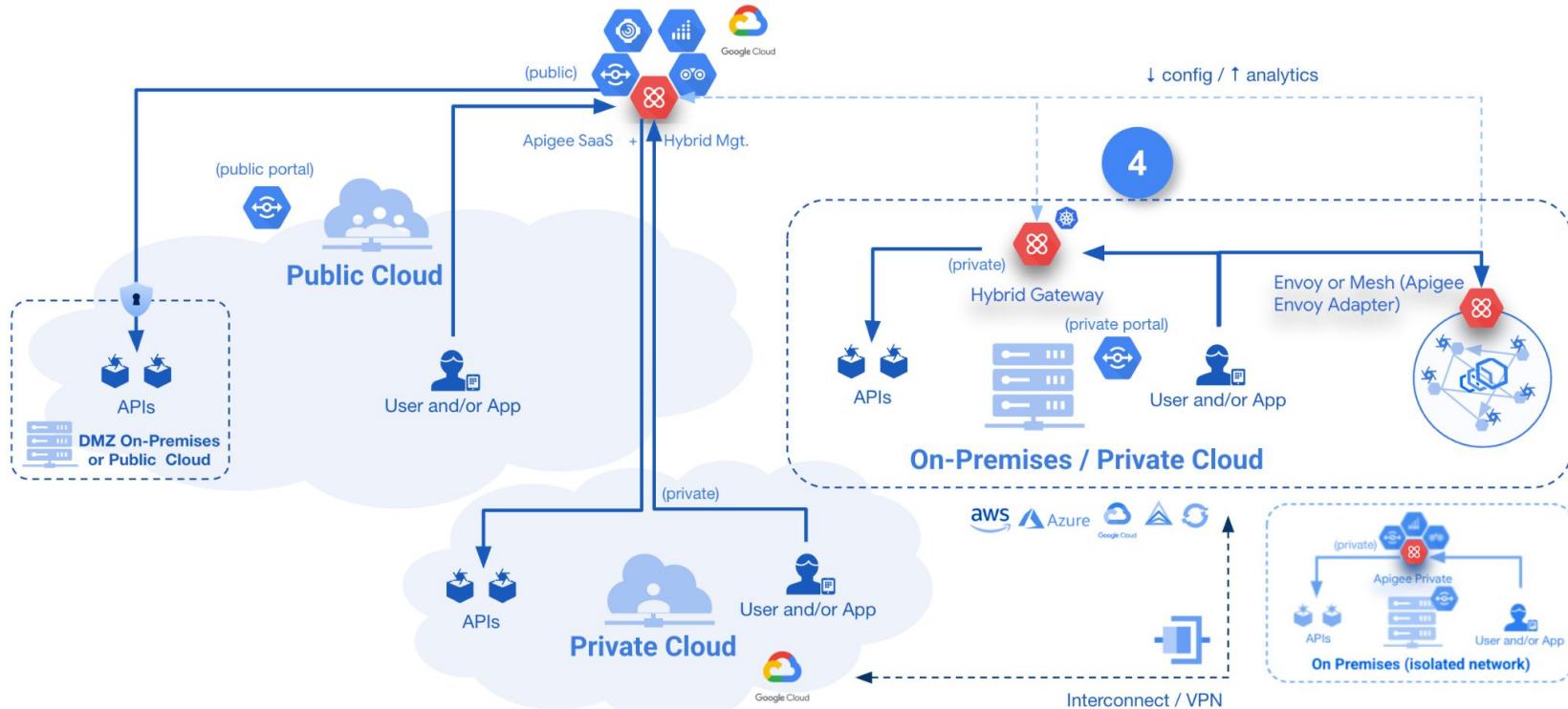


## Apigee Runtime Options

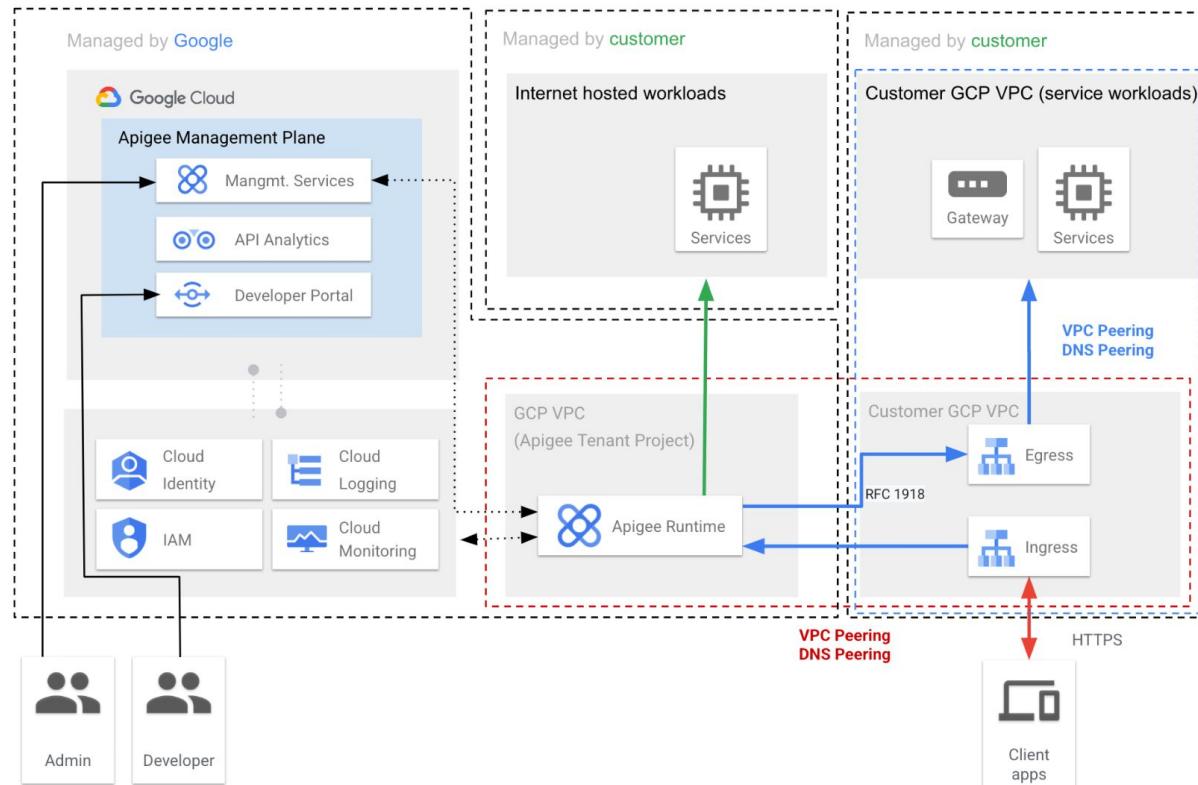
Ownership	Apigee X (SaaS)	Apigee Hybrid	Apigee Edge for Private Cloud (OPDK)
GCP-managed	<div style="display: flex; align-items: center; justify-content: space-around;"><div style="text-align: center;">Control Plane  Runtime Plane</div><div style="text-align: center;">Control Plane</div></div>		
Customer-managed		<div style="display: flex; align-items: center; justify-content: space-around;"><div style="text-align: center;">Runtime Plane</div><div style="text-align: center;">Control Plane  Runtime Plane</div></div>	<div style="display: flex; align-items: center; justify-content: space-around;"><div style="text-align: center;">Control Plane  Runtime Plane</div></div>



# The Big Picture - Deployment Options



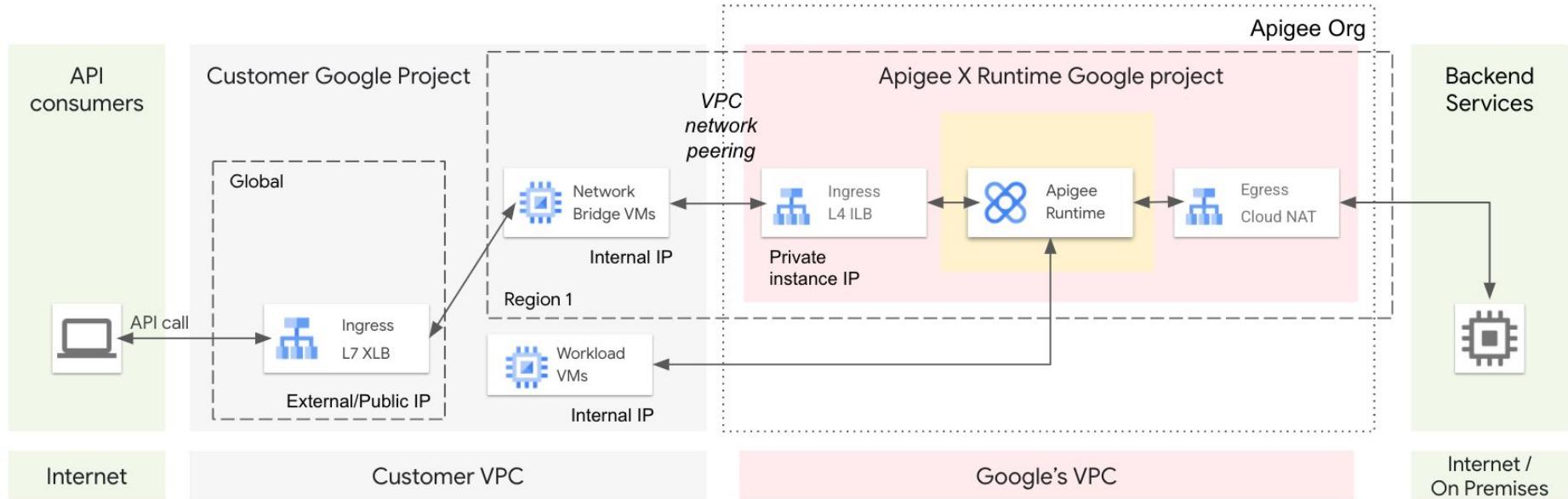
# Apigee X High-Level Architecture



- Apigee Management Plane**  
API Lifecycle Management Capabilities
- API Runtime Plane**  
Full featured enterprise API Gateway
- Managed Runtime**  
Automated updates and scaling



# Behind The Scenes - The Detailed Flow



# Getting Started with Apigee X

## [Apigee Provisioning Wizard](#)

Configure access to the 'eval-group' env group

Should 'eval-group' be accessible from the internet?

- No internet access: No external IP will be assigned from your internal network. Each of the regions will use a different internal IP address.
- Enable internet access: An HTTPS load balancer will be created and assigned a static external IP address. The load balancer will point to your selected domain to all regions.

Google Cloud Load Balancer      \$20-25 USD/month estimate

Use wildcard DNS service example: nlp.io

Domain \*

A load balancer and a VM instance will be created in your `willman-test1` project:

LOAD BALANCER      Internal IP: 10.0.0.1      External IP will be assigned

COMPUTE ENGINE      VM instance

us-west1 runtime      10.0.0.2

VM instance name: apigee-proxy      EDIT

Subnetwork \*

VM instances will use the subnet in region us-west1 and Apigee will enable Private Google Access for the subnet.

SSL Certificate (for)

You can provide a self-managed certificate or skip to have a Google-managed certificate created automatically for you, which can take up to 60 minutes

Signed certificate      BROWSE

RSA private key      BROWSE

SET ACCESS      CANCEL

## [Command Line Provisioning](#)

Step 2: Enable APIs

1. Apigee requires you to enable several Google Cloud APIs. Enable them by executing the following `gcloud services enable` command:

```
% gcloud services enable apigee.googleapis.com \
    servicenetworking.googleapis.com compute.googleapis.com \
    cloudresourcemanager.googleapis.com --project=PROJECT_ID
```

2. (Optional) To check your work, use the `gcloud services list` command to show all the enabled API services.

```
$ gcloud services list
```

The response shows all enabled services, including the APIs that you just enabled (Apigee, Service Networking, Cloud KMS, and Compute Engine).

Step 3: Create the Apigee service identity

1. Create the Apigee service identity:

```
% gcloud beta services identities create --service=apigee.googleapis.com \
    --project=PROJECT_ID
```

2. Verify that the agent was successfully created. The response should show the name of the agent in the following format: `service=PROJECT_NUMBER@sa-apigee.iam.gserviceaccount.com`. For example:

```
Service identity created: service=123456789012@sa-apigee.iam.gserviceaccount.com
```

Step 4: Configure service networking

Service Networking automates the private connectivity set up (using VPC Network Peering) between your network and Apigee. For more information and important considerations, see [Understanding peering ranges](#).

1. Create these environment variables:

```
% RANGE_NAME=$VPC_RANGE_NAME \
% NETWORK_NAME=$VPC_NETWORK_NAME
```

Where:

- `RANGE_NAME` is the name of the IP address range you are creating. You can name the range anything you want. For example: `google-apis`
- `NETWORK_NAME` is the name of the network resource in which the addresses should be reserved. Google creates a `default` network (named `default`) for each new project, so you can use that. However, Google does not recommend using the default network for anything other than testing.

\* Note: Google does not recommend using `defaults` unless you are testing this setup process. You should not use `defaults` with a `default` network in production.

## [Terraform Modules](#)

Sign up      [☰](#)

apigee / terraform-modules      Public

Terraform modules and ready to use end-to-end examples for Apigee.

Apache-2.0 License

### Apigee Terraform Modules

This repository provides terraform modules and ready to use end-to-end examples for Apigee.

#### Modules

Currently the following modules are available and can be used either as part of the end-to-end examples or as part of your own scripting:

- Apigee X Core Configures a complete Apigee X organization with multiple instances, environment groups, and environments.
- Apigee X Bridge MIG Configures a managed instance group of network bridge GCE instances (VMs) that can be used as a load balancer backend and forward traffic to the internal Apigee X endpoint.
- Apigee X mTLS MIG Configures a managed instance

## [Trial Provisioning Script](#)

```
curl -L
https://raw.githubusercontent.com/apigee/devrel/main/tools/apigee-x-trial-provision/apigee-x-trial-provision.sh | bash -
```





## Talk Material

[tinyurl.com/devfest2022-gurugram-apigee](https://tinyurl.com/devfest2022-gurugram-apigee)



devfest 2022



Digital Modernization with APIfication using Google Apigee  
Anmol Krishan Sachdeva (@greatdevaks) | Hybrid Cloud Architect, Google



# Thanks Everyone !



Anmol Krishan Sachdeva

Hybrid Cloud Architect, Google  
MSc Advanced Computing, University of Bristol, UK  
[LinkedIn](#) | [Twitter](#) (@greatdevaks)

