API Gateway Architecture

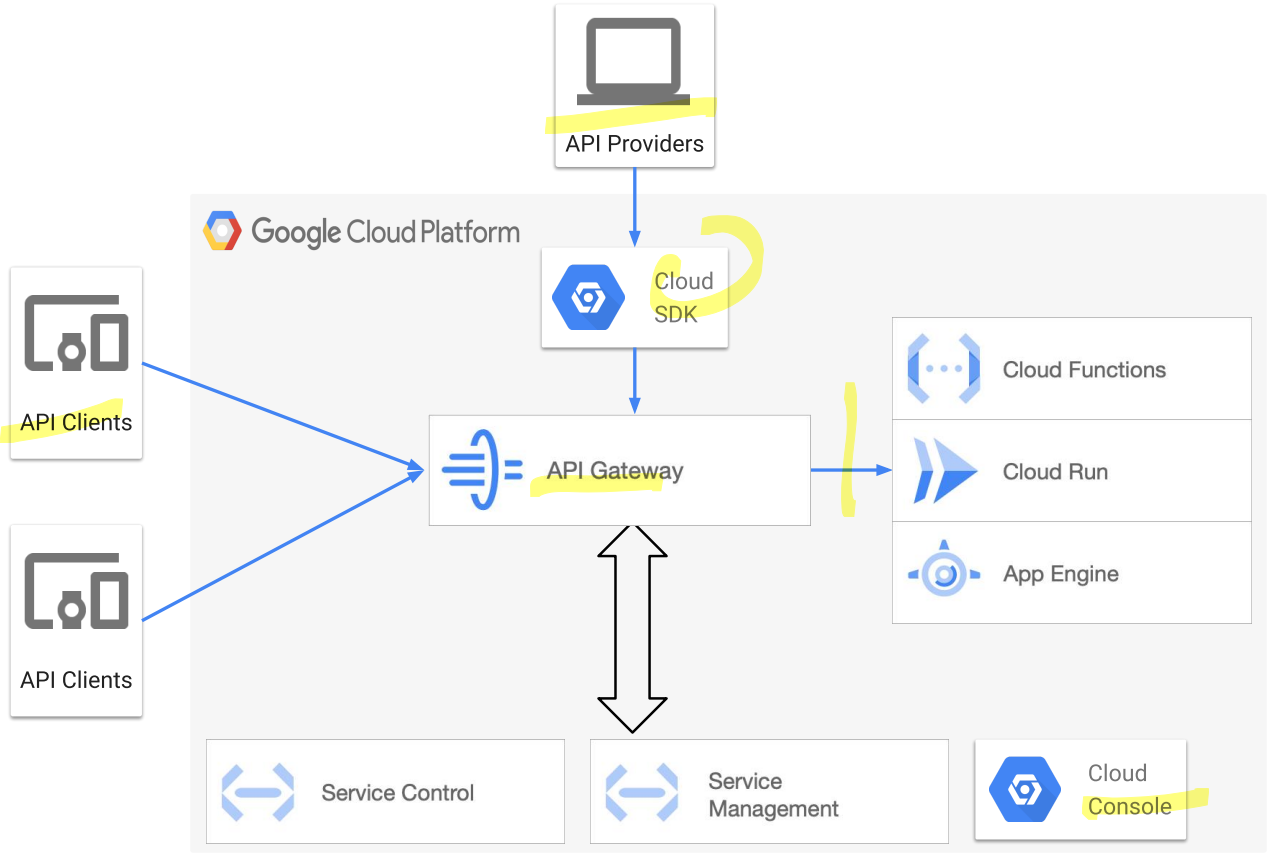
cloud.google.com/api-gateway/docs/architecture-overview

API Gateway is an API management system that provides management, monitoring, and authentication for your APIs. The components that make up API Gateway include:

- **API Gateway**: for managing all aspects of a deployed API
- **Service Control**: for applying API management rules
- **Service Management**: for managing API configurations
- **Cloud SDK**: for deploying and managing your APIs
- **Google Cloud Console**: for logging, monitoring and sharing

Architecture

Below is a high level diagram of the major components involved in API Gateway:



In this diagram:

- **The API provider** is responsible for creating and deploying an API on API Gateway. Each API is defined by a file written as an OpenAPI 2.0 spec.

  The OpenAPI spec defines the publicly facing URL of the REST endpoint for the API, the backend service accessed by the API, and any other characteristics of the API such as authentication, data format, and response options.

- **The API client** makes a REST request to an API hosted on API Gateway to access backend services. An API client can be any app capable of making a REST call, such as a browser, mobile app, or web app.

  The API client only needs to know the URL of the API, the request verb (such as `GET`, `PUT`, `POST`, `DELETE`), any authentication requirements, and the format of any data sent to or received from the API.

  The API client does not need to know anything about the backend implementation. In fact, a single API hosted on API Gateway can be configured to access different backends based on information passed in the request.

API Gateway components

API Gateway

API Gateway provides a fully managed, pay-per-use solution for hosting your APIs. API Gateway provide secure access to your backend services through a well-defined REST API that is consistent across all of your services, regardless of the service implementation.

API Gateway is integrated with Google Cloud so that you can use the same development, monitoring, logging, and trace tools that you use with any other Google Cloud product.

If you are connecting to a backend service hosted outside of Google Cloud, you can still take advantage of all Google Cloud services, including the authentication and authorization services used to control access to your APIs.

Service Control API

Service Control API applies API management rules at runtime, such as API key authentication, monitoring, and logging. Service Control provides the following methods:

- **Check**: verifies authentication and API keys, and indicates whether a call should be permitted
- **Report**: notifies the systems of record for logging and monitoring

Service Management API

You use the OpenAPI specification to define your API. You then use the Cloud SDK to upload the OpenAPI spec to Service Management, which creates the API config. Other configuration-related tasks also happen here, such as sharing your API with other developers, enabling or disabling the API in different projects, and generating API keys.

Cloud SDK

The Cloud SDK provides the gcloud command-line tool that you can use to make calls to various Google Cloud services. You use the `gcloud` command-line tool to upload your OpenAPI spec, which creates the API config, and then to deploy the API config to API

Gateway.

Cloud Console

Google Cloud Console is the graphical user interface for Google Cloud. Use the Google Cloud Console to expose monitoring and logging data recorded by Service Control, to configure authentication and authorization, and for developers to generate API keys to call the API.

Handling an API request

In an API configuration, there are two types of endpoints:

- **API endpoint**: defines the publicly available endpoint that clients use to consume your APIs.
- **Backend endpoint**: defines the endpoint that the API uses to connect to your backend service. Security settings, such as HTTP or HTTPS access, are defined by the implementation of the backend service.

Requests to your API endpoint are passed to the backend endpoint, including any data passed as part of the request. Responses from the backend service, including any data returned by the service, are passed back to the client.

Request routing

When a request is received:

1. API Gateway creates a trace token for Cloud Trace.

   Note: For the Beta release, trace is disabled.
2. API Gateway matches the path of the incoming requests with the target API. After finding a matching route, API Gateway performs any authentication steps for the specified API.

3. If JWT validation is necessary, API Gateway validates the authentication using the appropriate public key for the signer, and validates the audience field in the JWT. If an API key is required, API Gateway calls the Service Control API to validate the key.

4. Service Control looks up the key to validate it, and ensures that the project associated with the key has enabled the API. If the key isn't valid or the project hasn't enabled the API, the call is rejected and it is logged via the Service Control API.

5. If Service Control successfully validates the key, the request along with all original headers, plus a JWT validation header, if appropriate, is forwarded to the backend.

6. When a response is received from the backend, API Gateway returns the response to the caller and sends the final timing information to Trace. The call points are logged by the Service Control API, which then writes metrics and logs to their appropriate destinations.

What's next

API Gateway Deployment Model

Rate and review