CLOUD COMPUTING
(Undergraduate Course)

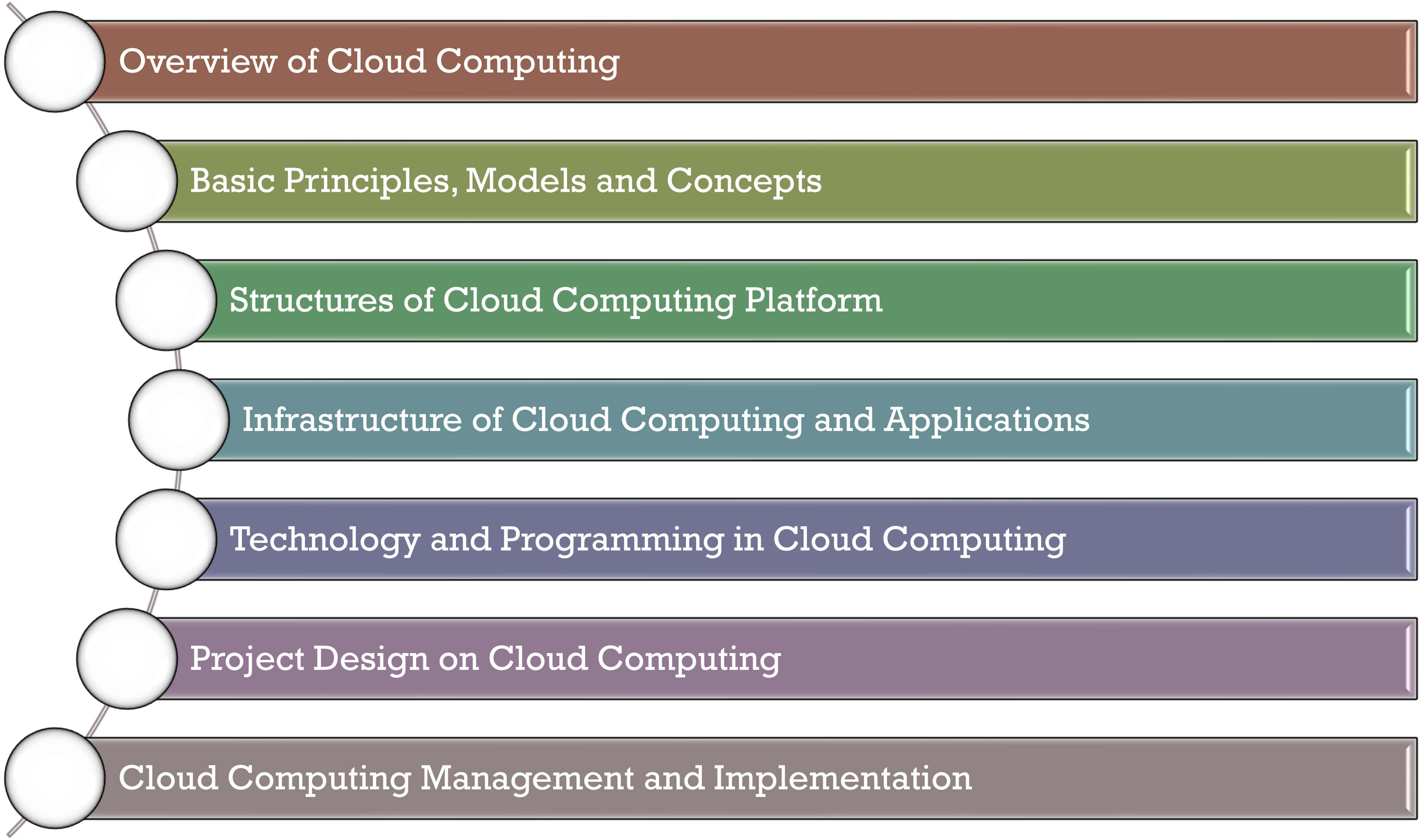
PRACTICES
Practice 2 – Cloud RESTfull, API, REST API

Presenter: **Dr. Nguyen Dinh Long**

Email: dinhlonghcmut@gmail.com

Oct. 2022

Outline



- Overview of Cloud Computing
- Basic Principles, Models and Concepts
- Structures of Cloud Computing Platform
- Infrastructure of Cloud Computing and Applications
- Technology and Programming in Cloud Computing
- Project Design on Cloud Computing
- Cloud Computing Management and Implementation

References

Main:

- Thomas Erl, Zaigham Mahmood, and Ricardo Puttini. 2013. *Cloud Computing Concepts, Technology & Architecture*. Prentice Hall.
- Michael J. Kavis. 2014. *Architecting the Cloud: Design Decisions for Cloud Computing Service Models*. Wiley
- Arshdeep Bahga, and Vijay Madisetti. 2013. *Cloud Computing: A Hands-On Approach*. CreateSpace Independent Publishing Platform

More:

- Rajkuma Buyya, Jame Broberg and Andrzej Goscinski. 2011. *Cloud Computing –Principles and paradigms*, Wiley
- Nick Antonopoulos, and Lee Gillam. 2010. *Cloud Computing - Principles, Systems and Applications*, Springer-Verlag London Limited.

- Slides here are modified from several sources in Universities and Internet.

Cloud Computing: Practices

NLU – DH20HM
Course: Cloud Computing

PRACTICES – Google Cloud Platform (GCP)

Levels: Beginning (3 weeks) – Intermediate (3 weeks) – Advanced (3 weeks)

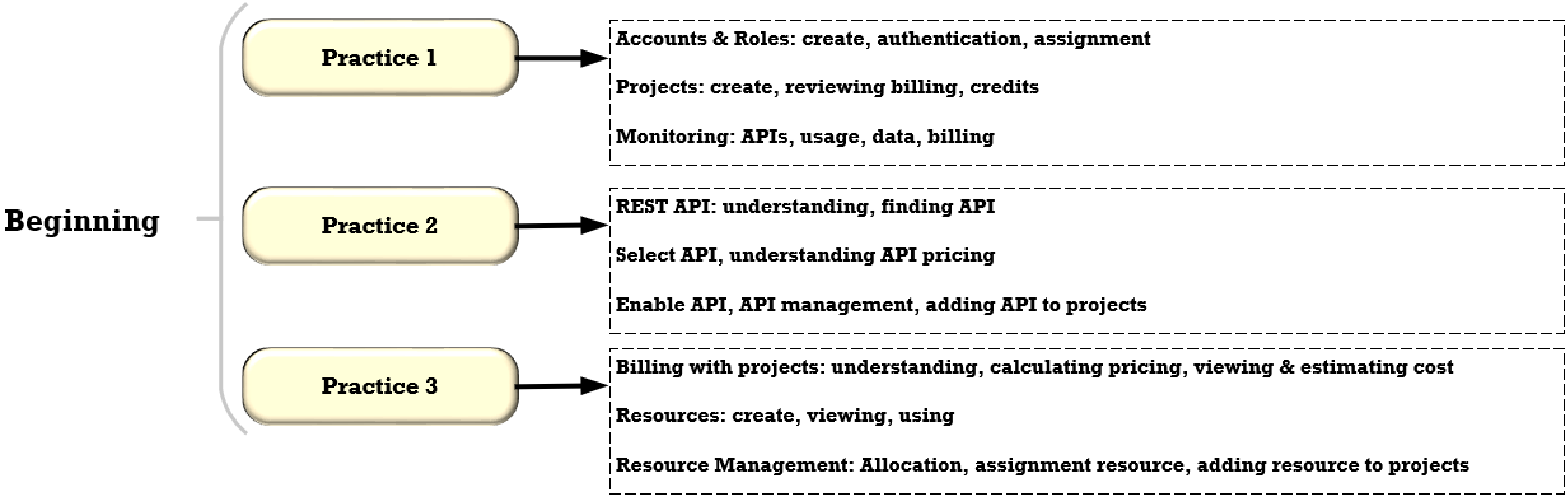
Groups: 9 with 5 person/group

Practice: submit a report for each group, submit to our Google Classroom

Cloud Computing: Practices

NLU – DH20HM
Course: Cloud Computing

PRACTICES – Google Cloud Platform (GCP)



Beginning

Practice 1

- Accounts & Roles: create, authentication, assignment
- Projects: create, reviewing billing, credits
- Monitoring: APIs, usage, data, billing

Practice 2

- REST API: understanding, finding API
- Select API, understanding API pricing
- Enable API, API management, adding API to projects

Practice 3

- Billing with projects: understanding, calculating pricing, viewing & estimating cost
- Resources: create, viewing, using
- Resource Management: Allocation, assignment resource, adding resource to projects

Content of Practice 2

1. Cloud RESTfull, REST API, API

2. Cloud API: finding, enabling, using, managing and monitoring

3. Practice Cloud APIs for Project development

Cloud RESTfull-API and Applications

REST and API:

- REST (Representational State Transfer) is a style of software architecture that provides a convenient and consistent approach to requesting and modifying data.
- REST works over the HTTP protocol that provides communication between client-server. REST is a transfer method used in software based on service-oriented architecture.

- REST is an API that defines a set of functions that programmers can use to send requests and receive responses using the HTTP protocol methods such as GET and POST. In the context of Google APIs, it refers to using HTTP verbs to retrieve and modify representations of data stored by Google.

- In a RESTful system, resources are stored in a data store; a client sends a request that the server perform a particular action (such as creating, retrieving, updating, or deleting a resource), and the server performs the action and sends a response, often in the form of a representation of the specified resource.

- Rest services:
  - They are platform-independent.
    (It doesn’t matter if the client is Windows, and the server is Linux)
  - They are language independent.
  - They work over HTTP.
  - They are flexible and can be extended very easily.

Cloud RESTfull-API and Applications

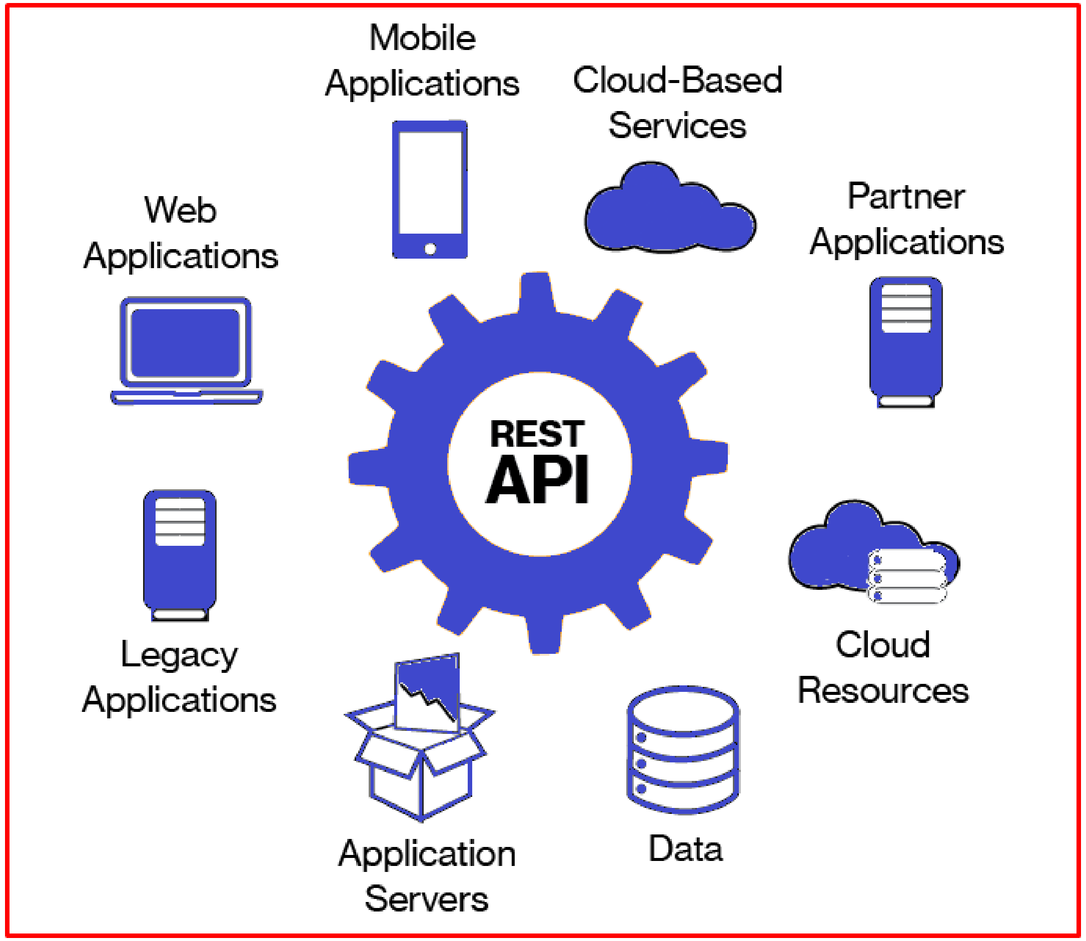
❑ REST and API:

- REST is an API that defines a set of functions that programmers can use to send requests and receive responses using the HTTP protocol methods such as GET and POST. In the context of Google APIs, it refers to using HTTP verbs to retrieve and modify representations of data stored by Google.

API (Application Program Interface) is an agreed way to send and receive data between computers

In Google's RESTful APIs, the client specifies an action using an HTTP verb such as POST, GET, PUT, or DELETE.

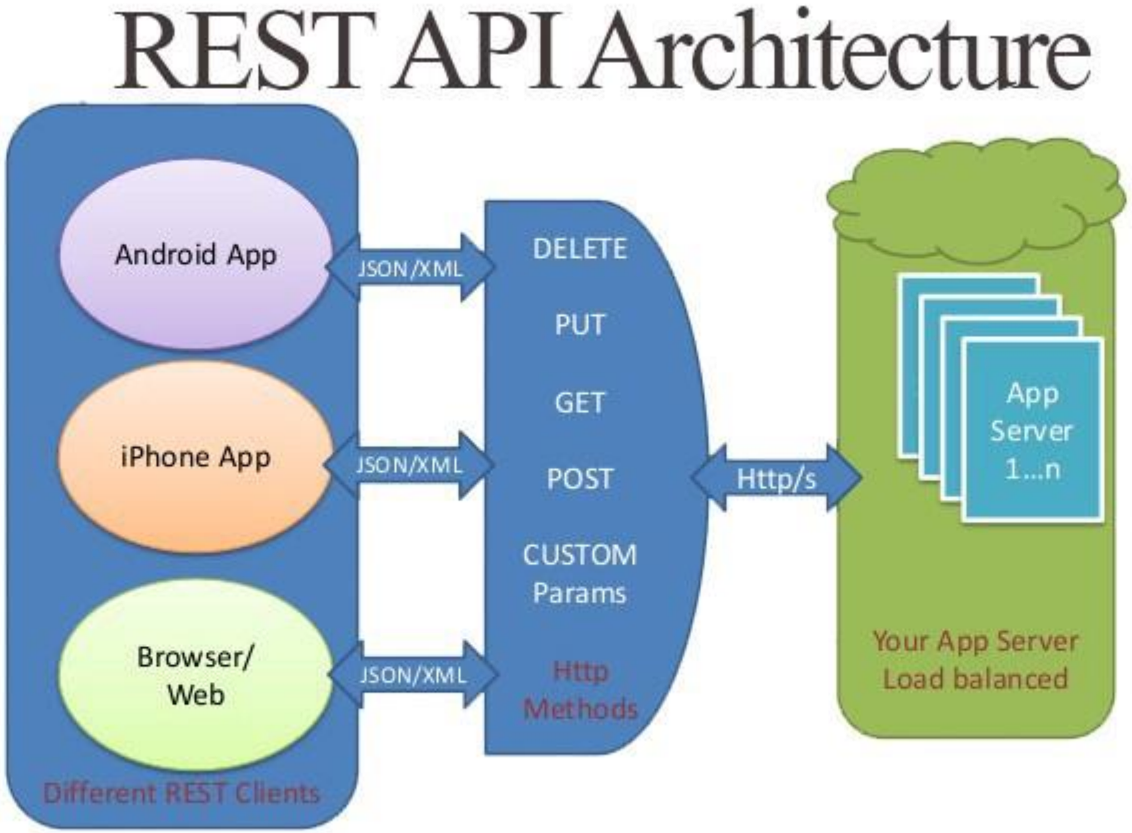
Because all API resources have unique HTTP-accessible URIs, REST enables data caching and is optimized to work with the web's distributed infrastructure.



Cloud RESTfull-API and Applications

❑ **REST API:**

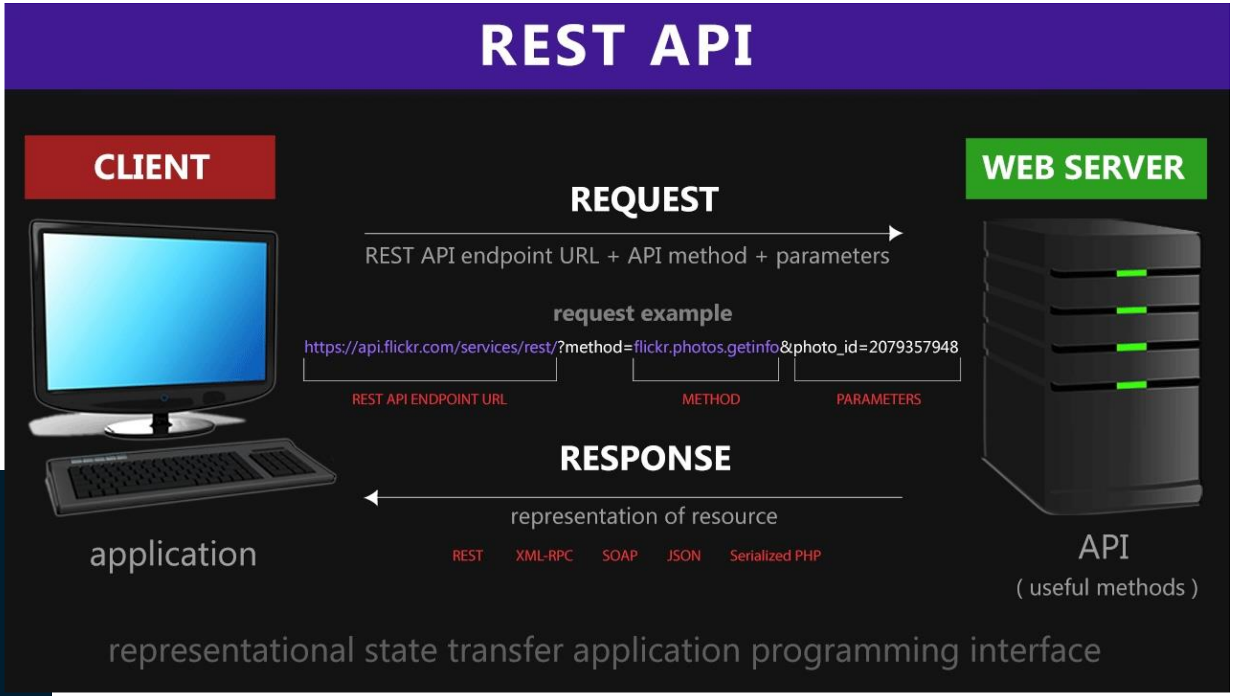
▪ REST API can be used by any site or application no matter what language it is written in because the requests are based on the universal HTTP protocol, and the information is usually returned in the JSON format that almost all of the programming languages can read.

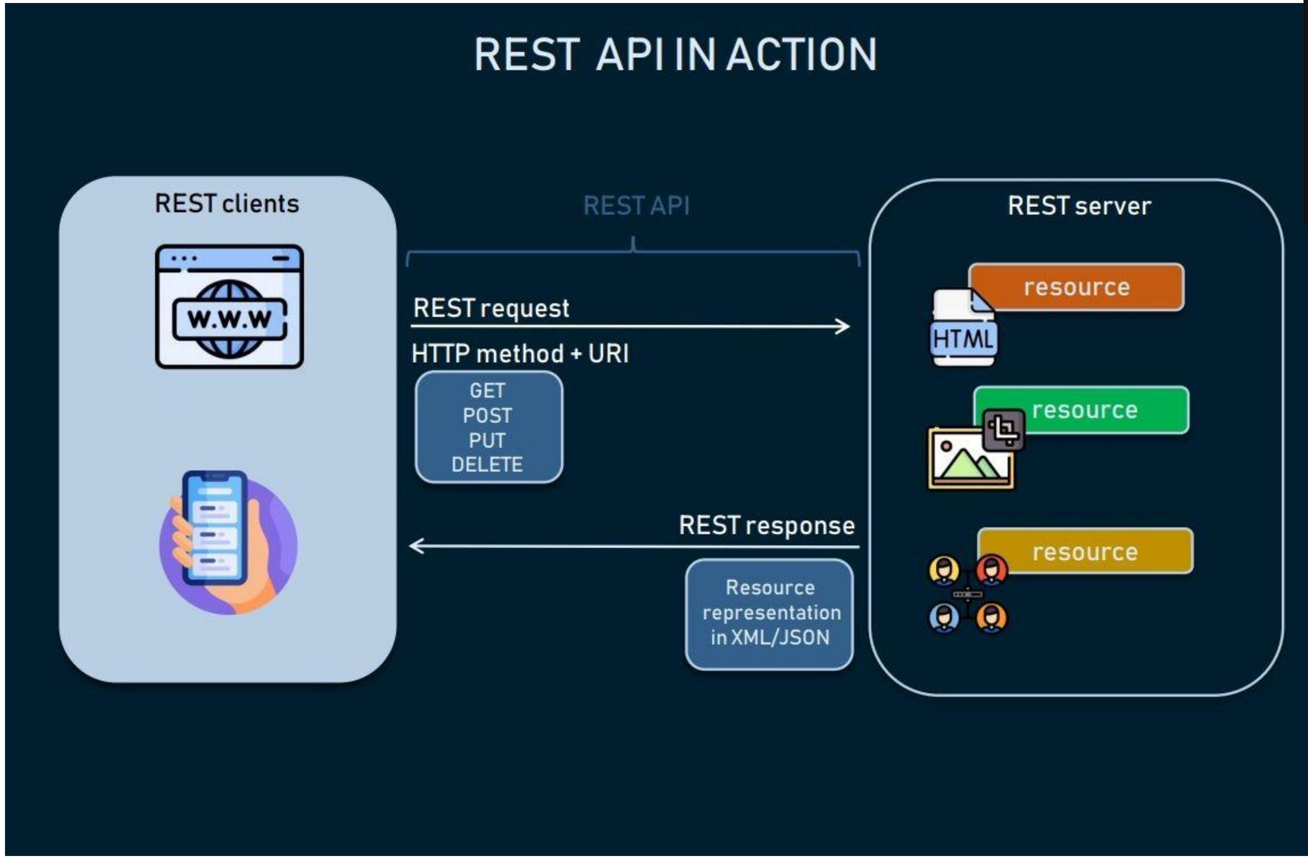


Cloud RESTfull-API and Applications

❑ **REST, API, REST API:**

.

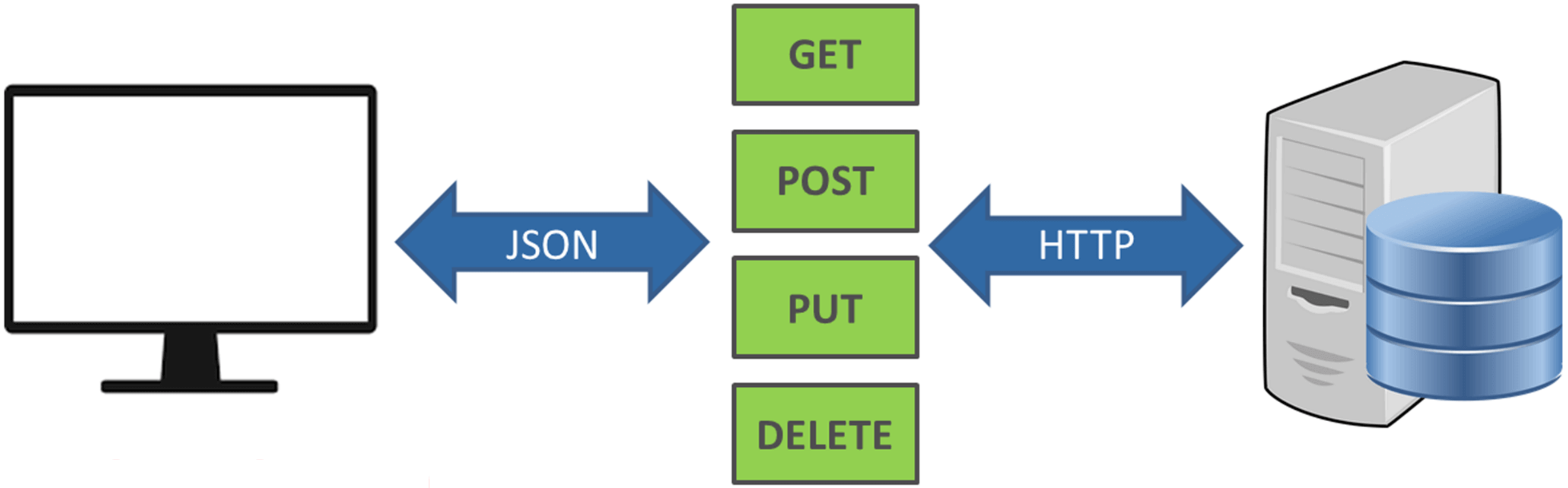

REST API
CLIENT
WEB SERVER
REQUEST
REST API endpoint URL + API method + parameters
request example
https://api.flickr.com/services/rest/?method=flickr.photos.getinfo&photo_id=2079357948
REST API ENDPOINT URL
METHOD
PARAMETERS
RESPONSE
representation of resource
REST
XML-RPC
SOAP
JSON
Serialized PHP
application
API
(useful methods)
representational state transfer application programming interface



Cloud RESTfull-API and Applications

❑ REST API:

- REST API can be used by any site or application no matter what language it is written in because the requests are based on the universal HTTP protocol, and the information is usually returned in the JSON format that almost all of the programming languages can read.



JSON

GET

POST

PUT

DELETE

HTTP

Client sends a **request**

HTTP methods

Server sends a **response**

Cloud RESTfull-API and Applications

- **Get started with REST for Google Photos APIs:**

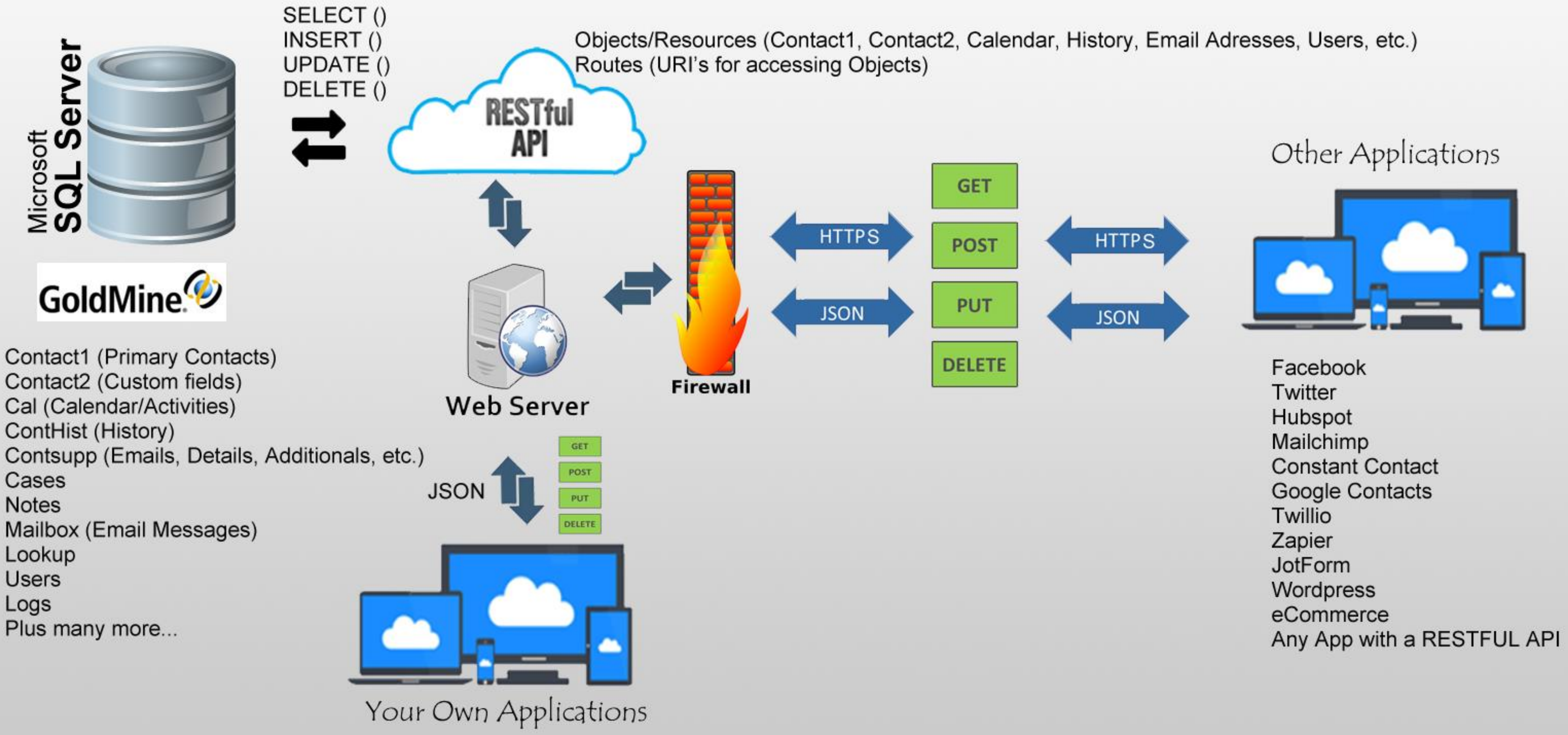
- To start using the Google Photos Library API, configure your project by enabling the API via the Google API Console and setting up an OAuth 2.0 client ID.
- Your application interacts with Google Photos on behalf of a Google Photos user. For instance, when you create albums in a user's Google Photos library or upload media items to a user's Google Photos account, the user authorizes these API requests via the OAuth 2.0 protocol.
- The OAuth 2.0 client ID allows your application users to sign in, authenticate, and thereby use the Library API. The Library API does not support service accounts; to use this API, users must be signed in to a valid Google Account.

Configure your app

- ✓ Enable the API
- ✓ Request an OAuth 2.0 client ID
- ✓ Testing

https://developers.google.com/photos/library/samples

Cloud RESTfull-API and Applications



Microsoft
SQL Server

GoldMine

Contact1 (Primary Contacts)
Contact2 (Custom fields)
Cal (Calendar/Activities)
ContHist (History)
Contsupp (Emails, Details, Additionals, etc.)
Cases
Notes
Mailbox (Email Messages)
Lookup
Users
Logs
Plus many more...

SELECT ()
INSERT ()
UPDATE ()
DELETE ()

RESTful
API

Objects/Resources (Contact1, Contact2, Calendar, History, Email Adresses, Users, etc.)
Routes (URI's for accessing Objects)

Web Server

Firewall

HTTPS

JSON

GET
POST
PUT
DELETE

HTTPS

JSON

Other Applications

Facebook
Twitter
Hubspot
Mailchimp
Constant Contact
Google Contacts
Twillio
Zapier
JotForm
Wordpress
eCommerce
Any App with a RESTFUL API

JSON

GET
POST
PUT
DELETE

Your Own Applications

Google Cloud APIs

- Google Cloud APIs are programmatic interfaces to Google Cloud Platform services. They are a key part of Google Cloud Platform, allowing you to easily add the power of everything from computing to networking to storage to machine-learning-based data analysis to your applications.

- Cloud APIs are exposed as network API services to customers, such as Cloud Pub/Sub API. Each Cloud API typically runs on one or more subdomains of googleapis.com, such as pubsub.googleapis.com, and provides both JSON HTTP and gRPC interfaces to clients over public internet and Virtual Private Cloud (VPC) networks.
- Clients can send HTTP and gRPC requests to Cloud API endpoints directly or by using client libraries.

- To see available Cloud APIs, see Google Cloud APIs in the Google Cloud console API Library.

Google Cloud APIs

- **Accessing Cloud APIs:**
- You can access Cloud APIs from server applications with our client libraries in many popular programming languages, from mobile apps via the Firebase SDKs, or by using third-party clients. You can also access Cloud APIs with the Google Cloud CLI tools or Google Cloud console.

- If you are new to Cloud APIs, see Getting Started on how to use Cloud APIs. https://cloud.google.com/apis/docs/getting-started

Google Cloud APIs

- **API Design Guide:**

- Regardless of the interface type, all Cloud APIs use resource-oriented design principles as described in our API Design Guide, which ensures Cloud APIs to have a simple and consistent developer experience. https://cloud.google.com/apis/design
- You can reference our API Design Guide to have a better understanding of Cloud APIs.

If you want to study the interface definition of Cloud APIs, you can visit the Google APIs repository on GitHub.

https://github.com/googleapis/googleapis

Google Cloud APIs

- **Capping your API usage:**

- Cloud APIs are shared among millions of developers and users. To ensure fair usage and minimize abuse risks, all Cloud APIs are enforcing rate limits and resource quotas on usage, commonly known as quotas.
- You can also use these quotas to control your spending on Google Cloud products by reducing your own quota limits.
- If you need more quotas than the default limits, you need to file quota increase requests.

For more information, see Capping API usage.

https://cloud.google.com/apis/docs/capping-api-usage

Google Cloud APIs

- **Capping your API usage:**
- Depending on the API, you can explicitly cap requests by limiting the requests per day, requests per minute, or requests per minute per user.
- You might want to limit the billable usage by setting caps. For example, to prevent getting billed for usage beyond the free courtesy usage limits, you can set requests per day caps.

- Limiting requests per user
- To prevent individual users from using up your API quota, some APIs include a default per user per minute limit. If such a default limit exists, you can modify that value as described in the previous section to limit the quota available to each user.

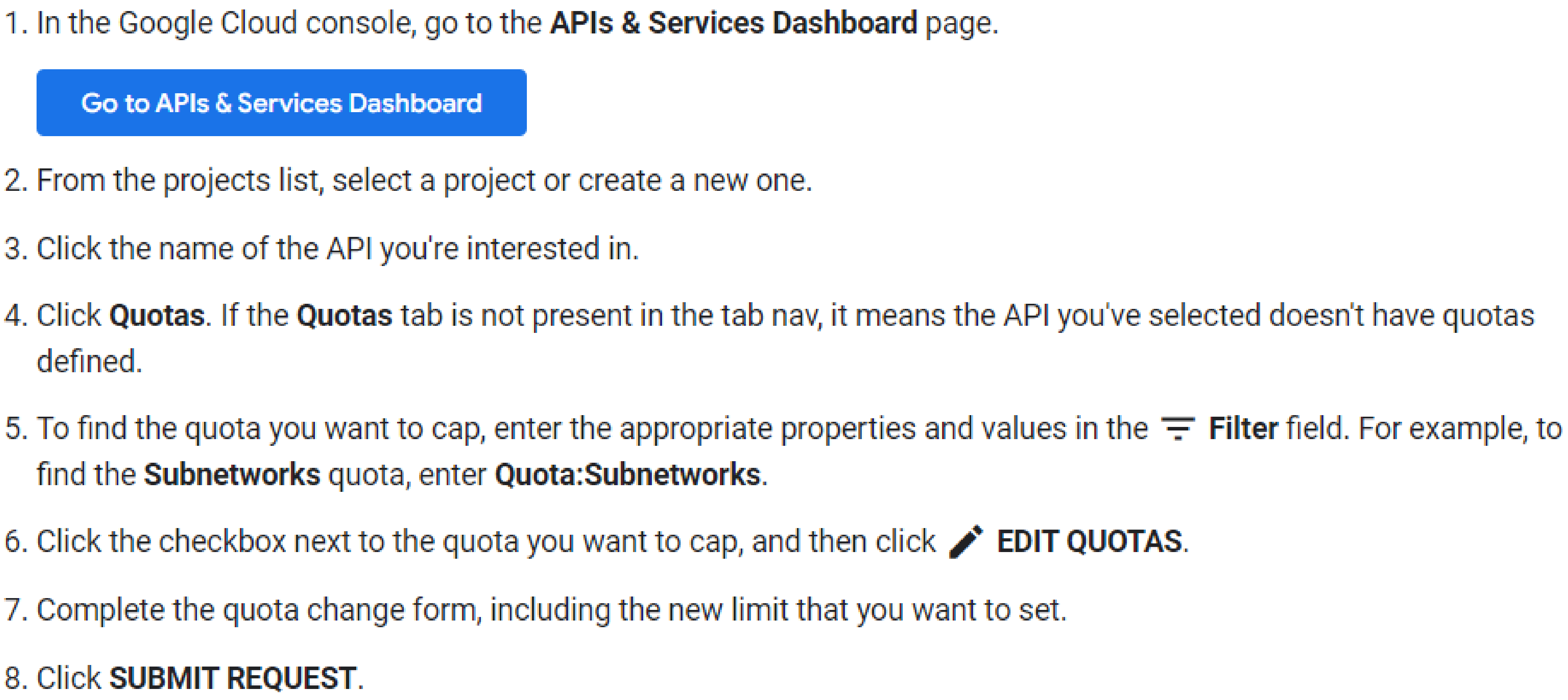
- Individual users are identified by a unique string. If you're creating a server-side application (where the calling code is hosted on a server that you own) that makes requests on behalf of users, your requests must include the quotaUser parameter.

Google Cloud APIs

❑ Capping your API usage:

- Modify the limits on the number of requests

You can set limits to all requests to any billable API. Most APIs set default limits, but you can change that limit up to a maximum specified by Google. Some APIs set a low limit until you enable billing on your project.



1. In the Google Cloud console, go to the **APIs & Services Dashboard** page.

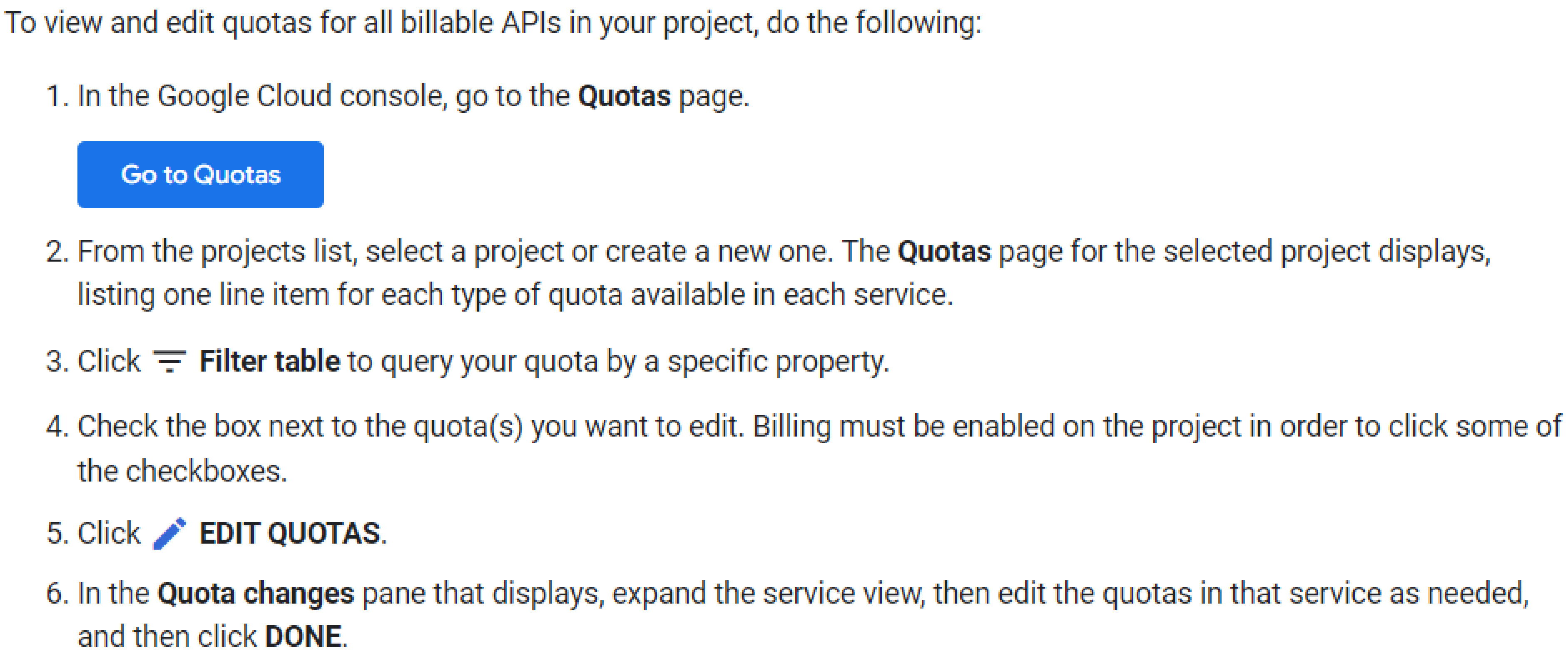
   Go to APIs & Services Dashboard

2. From the projects list, select a project or create a new one.
3. Click the name of the API you're interested in.
4. Click **Quotas**. If the **Quotas** tab is not present in the tab nav, it means the API you've selected doesn't have quotas defined.
5. To find the quota you want to cap, enter the appropriate properties and values in the **Filter** field. For example, to find the **Subnetworks** quota, enter **Quota:Subnetworks**.
6. Click the checkbox next to the quota you want to cap, and then click **EDIT QUOTAS**.
7. Complete the quota change form, including the new limit that you want to set.
8. Click **SUBMIT REQUEST**.

Google Cloud APIs

- **Capping your API usage:**
  - View and edit all quotas for APIs in a project

You can review, edit, and request higher quota limits for all billable APIs in a given project by viewing the Quotas page in the IAM & Admin section of Google Cloud console.



To view and edit quotas for all billable APIs in your project, do the following:

1. In the Google Cloud console, go to the **Quotas** page.

2. From the projects list, select a project or create a new one. The **Quotas** page for the selected project displays, listing one line item for each type of quota available in each service.
3. Click **Filter table** to query your quota by a specific property.
4. Check the box next to the quota(s) you want to edit. Billing must be enabled on the project in order to click some of the checkboxes.
5. Click **EDIT QUOTAS**.
6. In the **Quota changes** pane that displays, expand the service view, then edit the quotas in that service as needed, and then click **DONE**.

Monitoring API usage

- **Using the API Dashboard:**
- The simplest way to view your API metrics is to use the Google Cloud console's API Dashboard. You can see an overview of all your API usage, or you can drill down to your usage of a specific API.

- To see an overview of your API usage:

Visit Cloud console's APIs and Services section. The main API Dashboard is displayed by default. In this page you can see all the APIs you currently have enabled for your project, as well as overview charts for the following metrics:

1. **Traffic**: the number of requests per second made by or about your project to enabled APIs
2. **Errors**: the percentage of requests to enabled APIs that resulted in errors
3. **Median latency**: the median latency for requests to enabled APIs, if available".

Monitoring API usage

- **Using Cloud Monitoring:**
- If you use Cloud Monitoring, you can dive deeper into available metrics data using the Metrics Explorer to give you greater insight into your API usage. Cloud Monitoring supports a wide variety of metrics, which you can combine with filters and aggregations for new and insightful views into your application performance.

To see API metrics in Metrics Explorer, select Consumed API as the resource type, then select one of the *serviceruntime* metrics. Then use the filter and aggregation options to refine your data. After you've found the API usage information you want, you can use Cloud Monitoring to create custom dashboards and alerts that will help you continue to monitor and maintain a robust application. You can find out how to do this in the following pages:

- Building charts
- Introduction to alerting
- Managing alerting policies

For more information, see Metrics Explorer.

https://cloud.google.com/monitoring/charts/metrics-explorer

Google Cloud APIs

- **Best practices:**

While API metrics are an extremely useful tool, there are issues you need to consider to make sure they provide useful information, particularly when setting up alerts based on metric values. The following best practices will help you get the most from API metrics data.

- Is latency causing a problem?

While some services are quite latency-sensitive, for others scale and reliability matter more. With data from API metrics, you can learn what your users need from a given service.

- Look for changes from the norm

Before you decide to alert on a particular metric value, consider what actually counts as unusual behavior. Looking at your API metrics can show you that latency results for most services fall within a normal distribution: a big hump in the middle, and outliers on either side.

→ look for sustained changes from the norm that are correlated with observed issues in your application.

- Traffic rate

API metrics are most useful where you have a high volume of traffic going to the API. If you call a service only intermittently, your API metrics won’t be statistically valid and won’t give you meaningful triage information.

Google Cloud APIs

- **Client libraries explained:**

Client libraries make it easier to access Google Cloud APIs from a supported language. While you can use Google Cloud APIs directly by making raw requests to the server, client libraries provide simplifications that significantly reduce the amount of code you need to write.

This document explains the different types of client libraries that Google provides for Cloud APIs. You can also find out more about the available libraries for your product or language of choice in the product or language's documentation.

- Cloud Client Libraries

Cloud Client Libraries are the recommended option for accessing Cloud APIs programmatically, where available. Cloud Client Libraries use the latest client library model and:

  - Provide idiomatic code in each language to make Cloud APIs simple and intuitive to use.
  - Provide a consistent style across client libraries to simplify working with multiple Cloud services.
  - Handle all the low-level details of communication with the server, including authenticating with Google.
  - Can be installed using familiar package management tools such as npm and pip.
  - In some cases, give you performance benefits by using gRPC. For more information, see the gRPC APIs.

You can find installation instructions and reference material for the Cloud Client Library on the Client Libraries page for the Cloud APIs you're using. You can find links to get started with the Cloud Client Libraries and their reference documentation on the Cloud Client Libraries page.

Cloud Monitoring - Report

TASK I: Google API Cloud

How to find a API for your Project

- Hiển thị giao diện “Marketplace” (chợ Cloud) and Marketplace for API” (chợ API) để tìm và sử dụng API cho Project
- Cách thứ 2 để vào trực tiếp “chợ API” từ Menu “API & Services”

☺ Trình tự các bước thực hiện (từ giao diện chính Dashboard …)

☺ Ảnh chụp minh chứng sau khi enabled các APIs

Cloud Monitoring - Report

TASK II: API Library

How to find a right API for your Project

- ☺ Cách thứ 2 để vào trực tiếp "chợ API" từ Menu "API & Services
  Trình tự các bước thực hiện (từ giao diện chính Dashboard …)

- ☺ Hiển thị danh mục các APIs thuộc "Category": Maps, Databases
  Ảnh chụp minh chứng theo mỗi loại "Category"

- ☺ Tạo bảng phân loại các APIs này theo "Category":

  Google Earth Engine API, Places API, Geolocation API, Datastream API, Compute Engine API, Cloud Run API, Cloud Healthcare API, Cloud Datastore API, Google Drive API, YouTube Data API

Cloud Monitoring - Report

TASK III: API Enabling

Finding and Enabling APIs: Map JavaScript API, Cloud Storage API and BigQuery API

- ☺ Trình tự các bước thực hiện (tính từ giao diện chính Dashboard …)
- ☺ Mỗi API đề cập thuộc “Category” nào trong API Library
- ☺ Mục đích sử dụng 3 APIs này trong Project
- ☺ Ảnh chụp minh chứng sau khi enabled các APIs

Cloud Monitoring - Report

TASK IV: Capping your API usage

Management and Limitation Quotas: APIs usage

- ☺ Hiển thị giao diện “quotas” của 3 APIs (Task III)
  Ảnh chụp minh chứng cho mỗi APIs

- ☺ Select a API and “modifi limit number of requests” for this API
  Ảnh chụp minh chứng

- ☺ Hiển thị giao diện show tất cả “quotas” của Project
  Trình tự thực hiện (từ giao diện chính Dashboard …)
  Ảnh chụp minh chứng

Cloud Monitoring - Report

TASK V: Monitoring API usage

Using công cụ “API metrics” để quản lý APIs của Project:

- ☺ Mục đích chính của “Monitoring API usage” là gì trong việc quản lý Project
- ☺ Trình tự các bước thực hiện (từ giao diện chính Dashboard …)
- ☺ Liệt kê bao nhiêu cách để truy cập vào tính năng quản lý API từ Google Cloud
- ☺ Ảnh chụp minh chứng các tính năng quản lý API