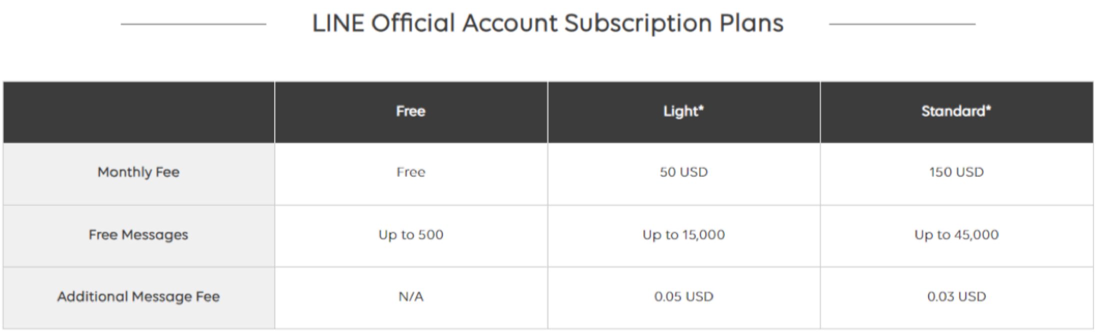
Q1: How is your project architecture related to the theory taught in the lecture?

According to our project, it actually uses the cloud computing technology and like a small distributed system which through the network to achieve sharing of hardware resources, file, database, information and so on. The distributed system is the one which hardware and software components located at networked computers then communicate and coordinate their actions by message passing such as API. As for our project, we push our project code in the Heroku cloud computing platform as a service which can register and run our project code in the cloud by using virtual hardware and software components. Moreover, in our project, we also use some API to connect with several servers and databases in order to get some services form them. Those reflect the concept of distributed system and cloud computing technology. For example, in this project, we use Linebot API which is provided by the official Line to create a program on the server to receive messages, process messages, and return messages. In this way, we cannot to design and develop our own chatbot server but use cloud to connect other existing server which have already created good software for using. For another thing, we use Redis to connect with this database so that we can achieve the function of recording user's news reading. The redis might contains many networked computers to achieve its various services but we can simply use cloud to make use of those services. Furthermore, in the project, we also use GoogleMap API to connect with Google system to make use of its map function for receiving some hospital location directly and automatically.

Take GoogleMap API as a example to illustrate the distributed system models. For instance, GoogleMap API uses HTTP requests to access driving, cycling, walking and public transportation routes, which represents the communication entities that are web services in the architecture model in the distributed system. Normally, form a system perspective, the entities that communicate in a distributed system are typically process, but from a programming perspective, more problem-oriented abstractions have been proposed such as web services that use web standards to represent and discover services. As for the concurrency and synchronous problems, our project uses redis to address these two things to achieve high concurrency and non-blocking. The Redis server is single-threaded for command processing, but at the I/O level it can simultaneously provide services to multiple clients concurrently, and the conversion from concurrent to internal single-threading is achieved through a multiplexing framework. I/O multiplexing is actually managing multiple I/O streams by recording and tracking the status of each socket (I/O stream) in a single thread.

Q2: Can you demonstrate, with some screen cap, how to increase capacity of your chat bot service?

As for Linebot, our chatbot design only use message function so that we can still stay at free account plan and no need to upgrade as Figure 1.

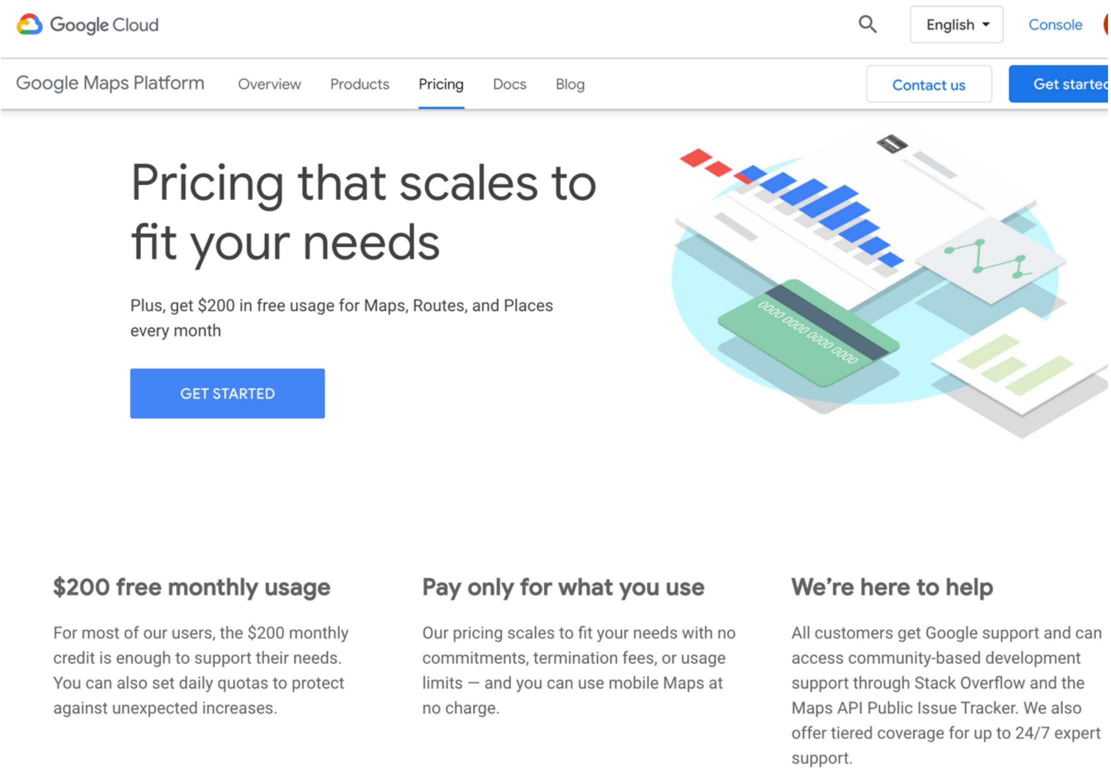


LINE Official Account Subscription Plans

	Free	Light*	Standard*
Monthly Fee	Free	50 USD	150 USD
Free Messages	Up to 500	Up to 15,000	Up to 45,000
Additional Message Fee	N/A	0.05 USD	0.03 USD

Figure 1- Line official account subscription plans

As for Google Maps API, we use it for find nearest hospital function. Figure 2 shows that we can use whatever we want, and the price is scalable to fit our needs. When lots of users use the function, the number of request will increase so that the charge will increase.



Pricing that scales to fit your needs

Plus, get $200 in free usage for Maps, Routes, and Places every month

$200 free monthly usage

For most of our users, the $200 monthly credit is enough to support their needs. You can also set daily quotas to protect against unexpected increases.

Pay only for what you use

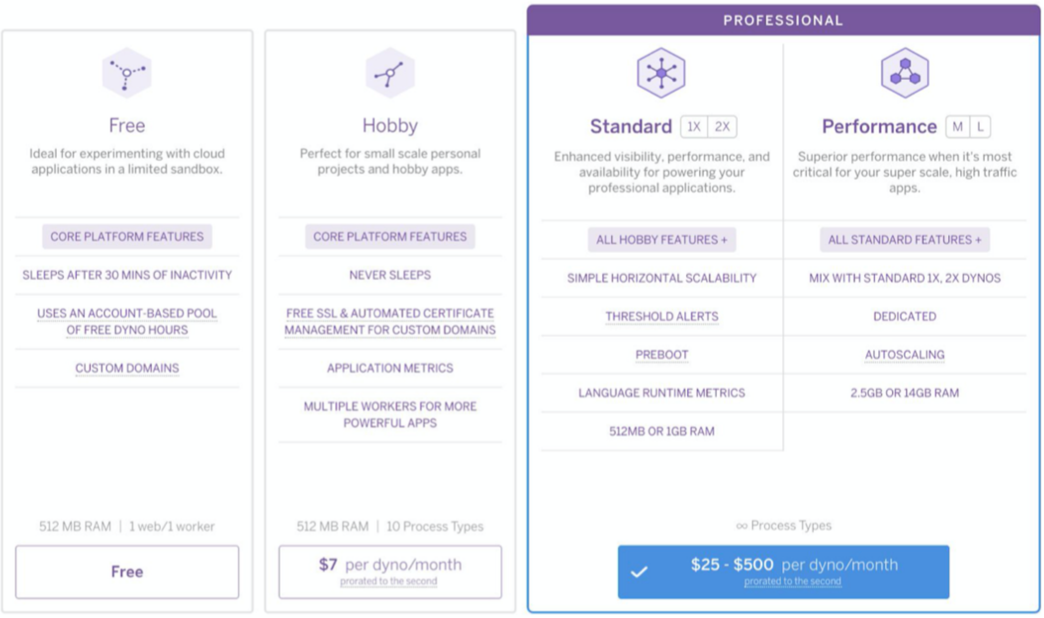
Our pricing scales to fit your needs with no commitments, termination fees, or usage limits — and you can use mobile Maps at no charge.

We’re here to help

All customers get Google support and can access community-based development support through Stack Overflow and the Maps API Public Issue Tracker. We also offer tiered coverage for up to 24/7 expert support.

Figure 2 - Google Maps API charge

As for Heroku, we can use Upgrade Heroku CPU, RAM, Auto Scalability, Dyno Linux container and relocate cloud server region to the closest data center function when the number of user or the data become larger and larger as figure3.



PROFESSIONAL

Free

Ideal for experimenting with cloud applications in a limited sandbox.

CORE PLATFORM FEATURES

SLEEPS AFTER 30 MINS OF INACTIVITY

USES AN ACCOUNT-BASED POOL OF FREE DYNO HOURS

CUSTOM DOMAINS

512 MB RAM | 1 web/1 worker

Free

Hobby

Perfect for small scale personal projects and hobby apps.

CORE PLATFORM FEATURES

NEVER SLEEPS

FREE SSL & AUTOMATED CERTIFICATE MANAGEMENT FOR CUSTOM DOMAINS

APPLICATION METRICS

MULTIPLE WORKERS FOR MORE POWERFUL APPS

512 MB RAM | 10 Process Types

$7 per dyno/month
prorated to the second

Standard 1X 2X

Enhanced visibility, performance, and availability for powering your professional applications.

ALL HOBBY FEATURES +

SIMPLE HORIZONTAL SCALABILITY

THRESHOLD ALERTS

PREBOOT

LANGUAGE RUNTIME METRICS

512MB OR 1GB RAM

Performance M L

Superior performance when it's most critical for your super scale, high traffic apps.

ALL STANDARD FEATURES +

MIX WITH STANDARD 1X, 2X DYNOS

DEDICATED

AUTOSCALING

2.5GB OR 14GB RAM

∞ Process Types

✓ $25 - $500 per dyno/month
prorated to the second

Figure 3 – Heroku plans

Q3: Can you identify if you bot is one of the examples of PaaS, IaaS, SaaS?

Our bot is one of the examples of SaaS. In our chatbot, it can provide some services for users such as searching latest news service, providing popular science service and getting hospital location in map service. For instance, in the main-services box, when users click "popular science", the chatbot will return another menu which precaution and more knowledge options. And when user click them respectively, user will receive some information automatically. Moreover, when user send their location to the chatbot, it will immediately search the nearest hospital location and show them in google map then return this to the user. In this way, user can easily and conveniently get a nearest hospital location without searching by themselves. In this way, our bot is a sample of SaaS.