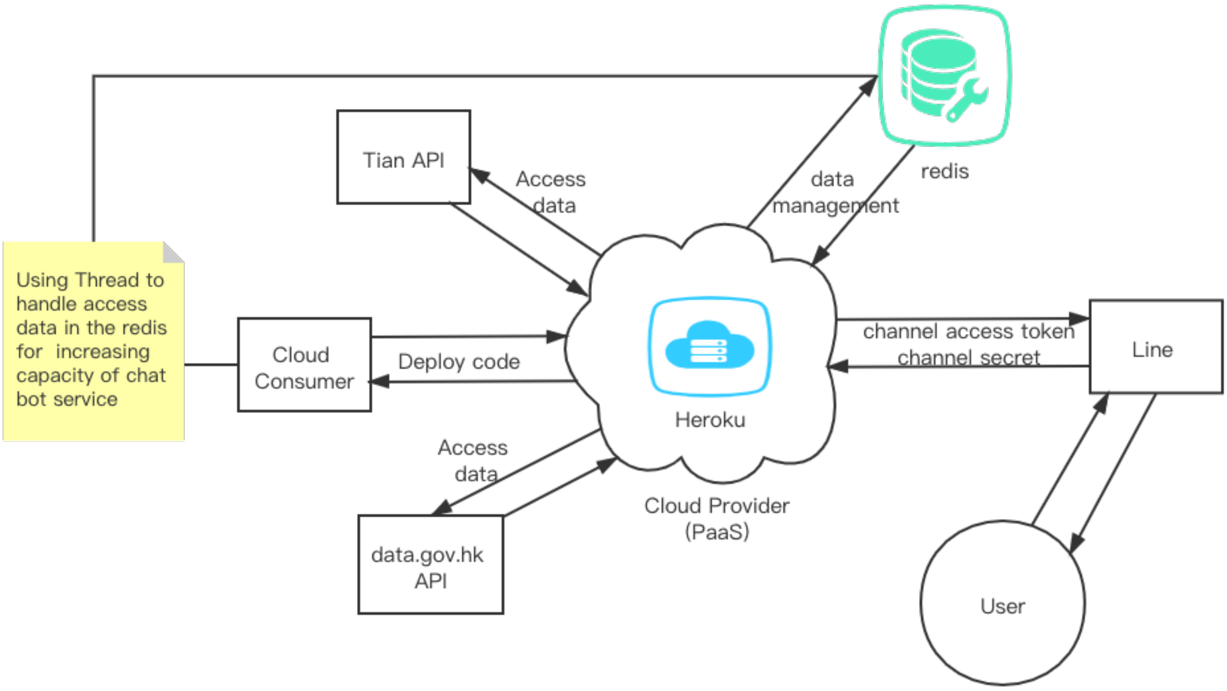
The project architecture is shown like the following figure. The Cloud Consumer is the code of the line chat bot.
1.Once the code is deployed to the cloud side(the cloud provider) which is the heroku.
2.According to the channel access token and channel secret, it would match to the line chatbot I already set.
3.Except some basic operation like the port or parse webhook body, it would connect to the redis and read and save the data in json format from two APIs which are Tian API and data.gov.hk API corresponding to the COVID-19 confirmed case in the world and in Hong Kong.
4. Then after user typing a district name in Hong Kong/ a country name / short code of a country, the line chatbot would reply the number of confirmed case in this area. At this time, I use Thread method to access two data set in redis simultaneously which is for increasing the capacity of chat bot service.

In my personal perspective, there are several way to for the increase of capacity of chat bot service. For the virtual server side, we can increase the configuration of the server to make sure the server can support the huge amount of users or internet access. The second method is to design algorithms to handle the logic/structure of code, such as the Hamiltonian. But at the chat bot, I use a simpler design to increase the capacity of service which is the thread for accessing the two different data set in redis.

Obviously, the line chat bot is one of example of PaaS. The infrastructure-centric IT resources prepared by the Heroku, we only need to do is to develop and deploy operation in this “ready to use” environment to control the application. In this case, we deploy the code in python to heroku git side to control or response for the event which comes from the client side.


Tian API
Access data
data management
redis
Using Thread to handle access data in the redis for increasing capacity of chat bot service
Cloud Consumer
Deploy code
Heroku
channel access token
channel secret
Line
Access data
data.gov.hk API
Cloud Provider (PaaS)
User