

May 09, 2023

Computer Engineering,
Chulalongkorn University

IoT Healthcare

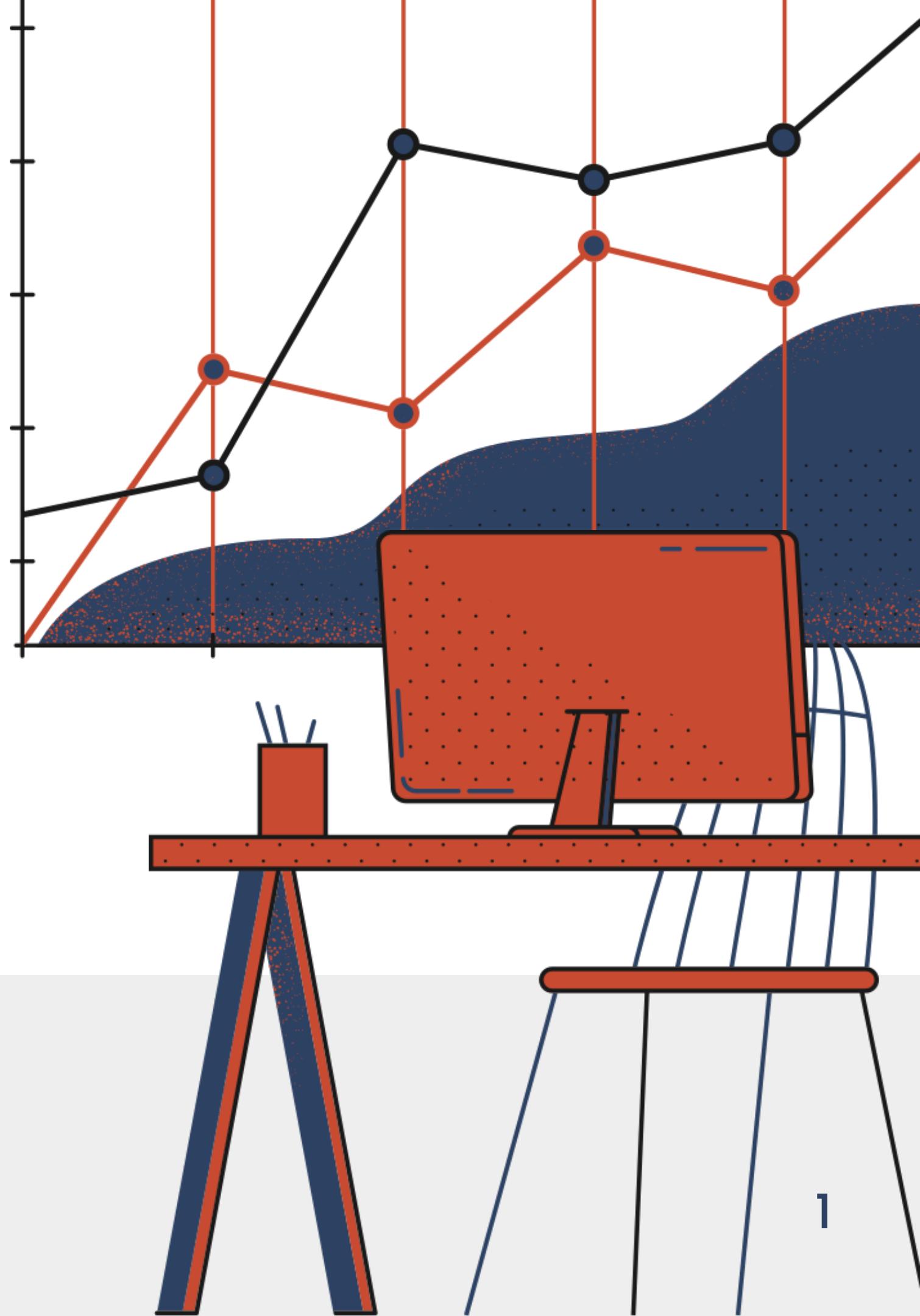
Final presentation

ADVISOR

Assoc. Prof. Kultida Rojviboonchai

Expert (SnapLogic)

Thanawut	Ananpiriyakul
Tanapat	Ruengsatra
Kanya	Kiatsakdawong



Member



Nitchakran Chaipojjana
6231322621



Siwagarn Jitwarodom
6231363321



Marineya Tajoparung
6231352421



Nattapat Jaruchaisittkul
6230177821

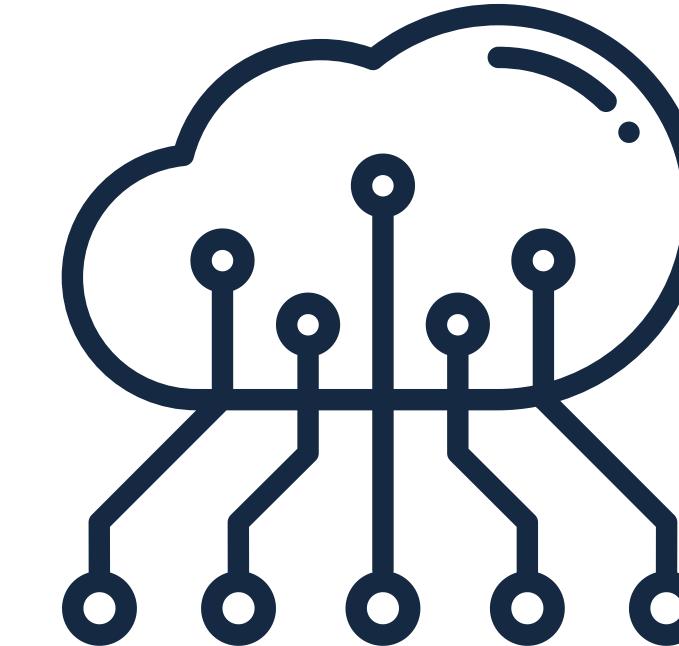
Table of Contents

- I Project Overview
- II Requirement
- III To be system
- IV Testing
- V Summary

Project Overview



Open Healthcare IoT Platform



Requirement



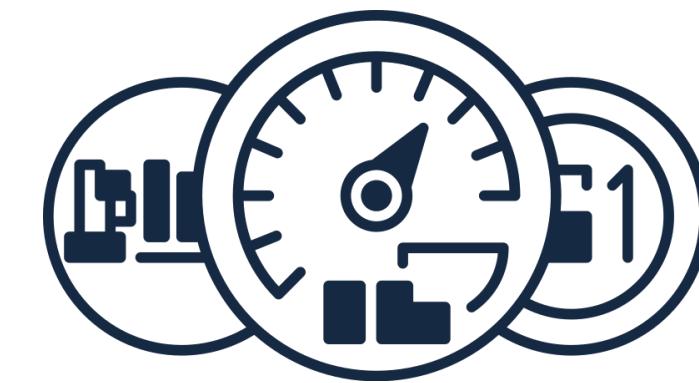
Systems



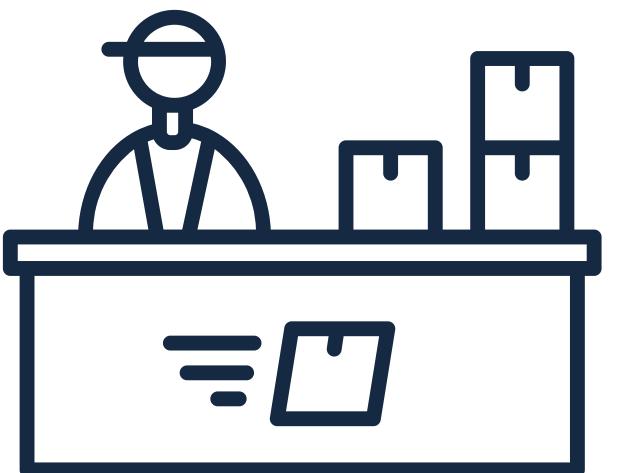
Data Receiving



Data Management



Developer Service



Admin Service



Logging

Systems



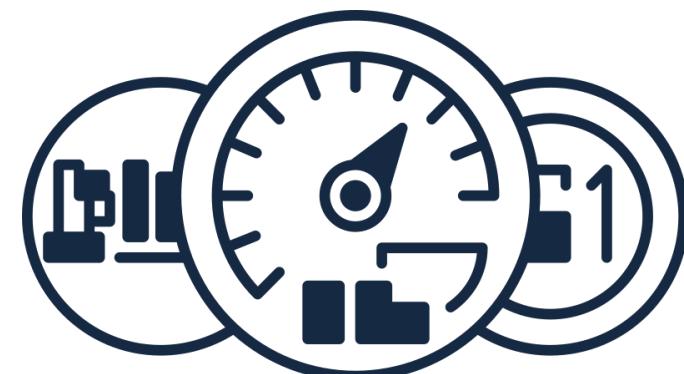
Data Receiving

- MQTT
- Kafka



Data Management

- Database: MongoDB
- Data Lake & Data Warehouse
- Dashboard



Developer Service

- Data Transmission (real-time & historical)
- Dashboard: application & subscriptions
- application's customer authentication & authorization

Systems



Admin Service

- Dashboard: device, developer, developer's subscriptions

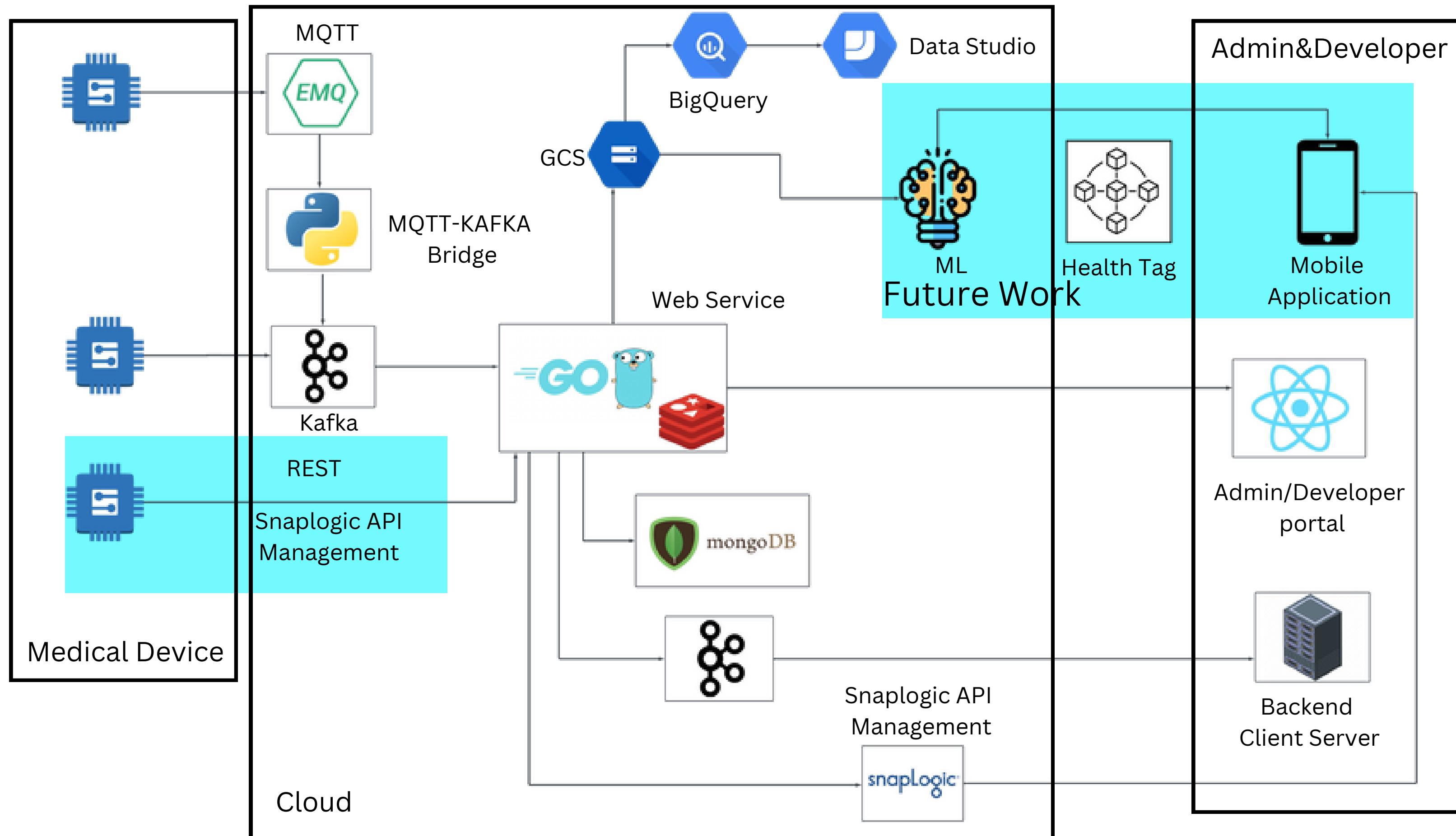


Logging

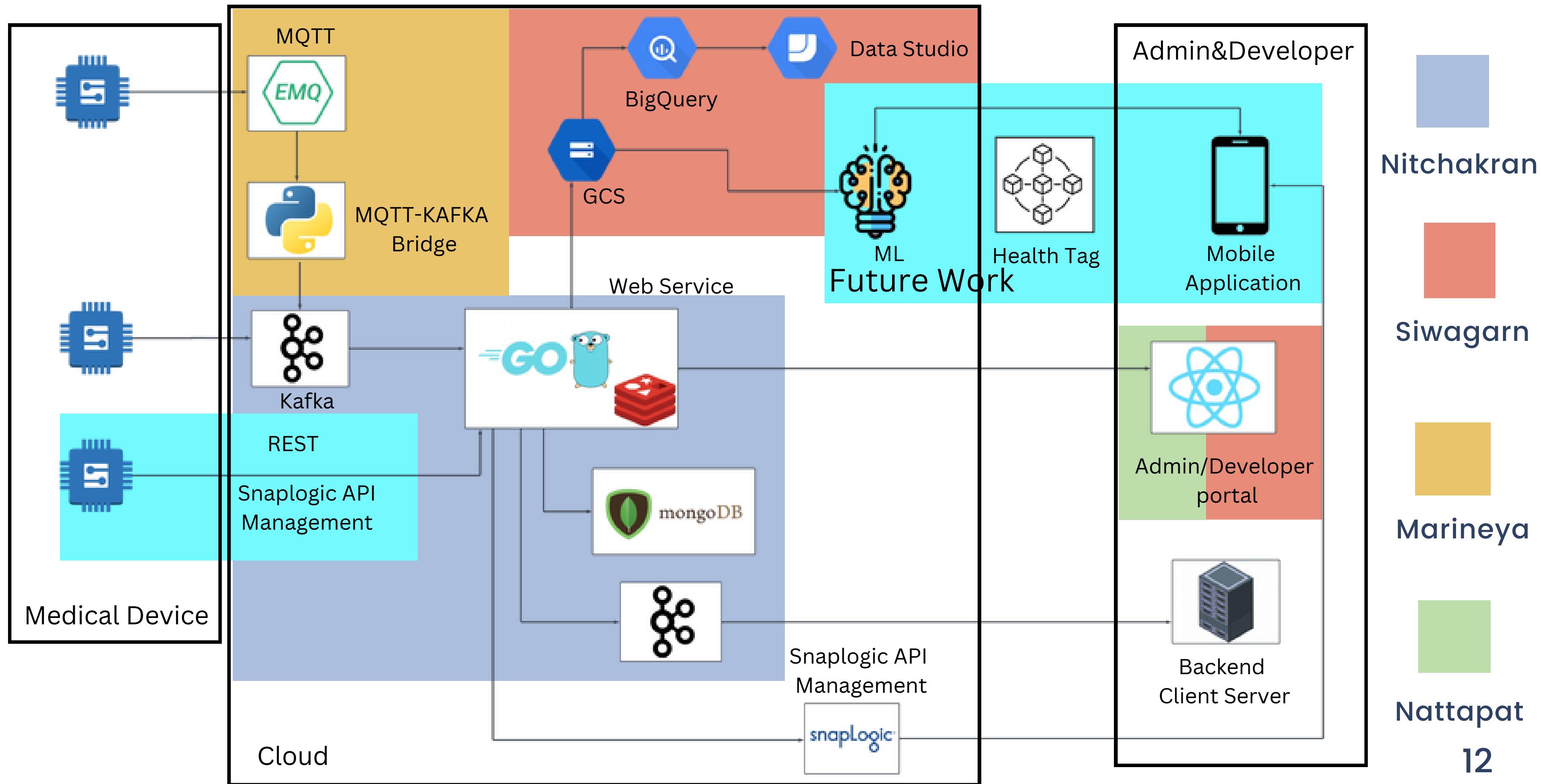
- Log of uploading to Data Lake
- api log
- system monitoring
- dashboard log

To be system





III To be system: Responsibility



Receiving Part

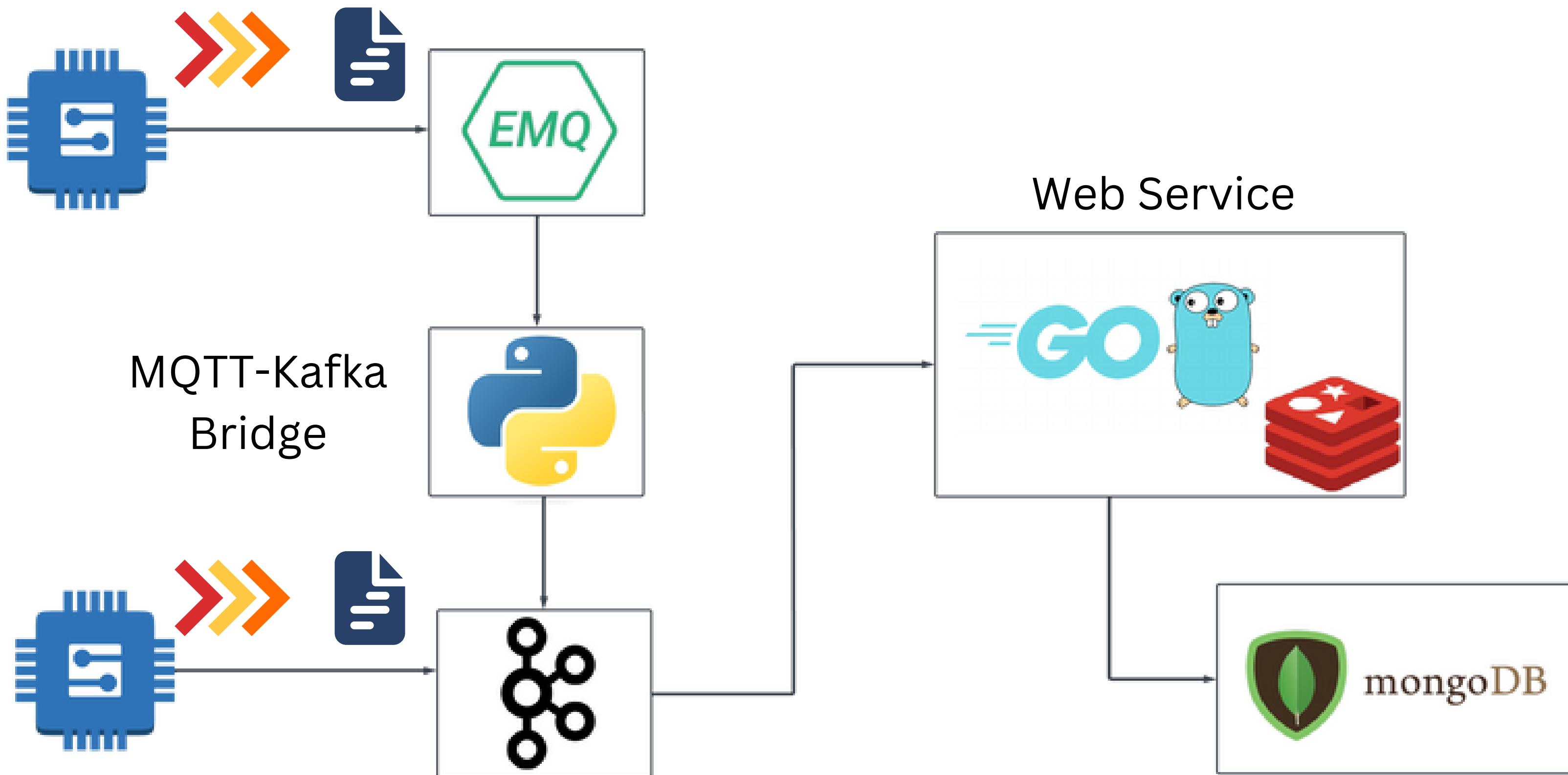
Data Flow

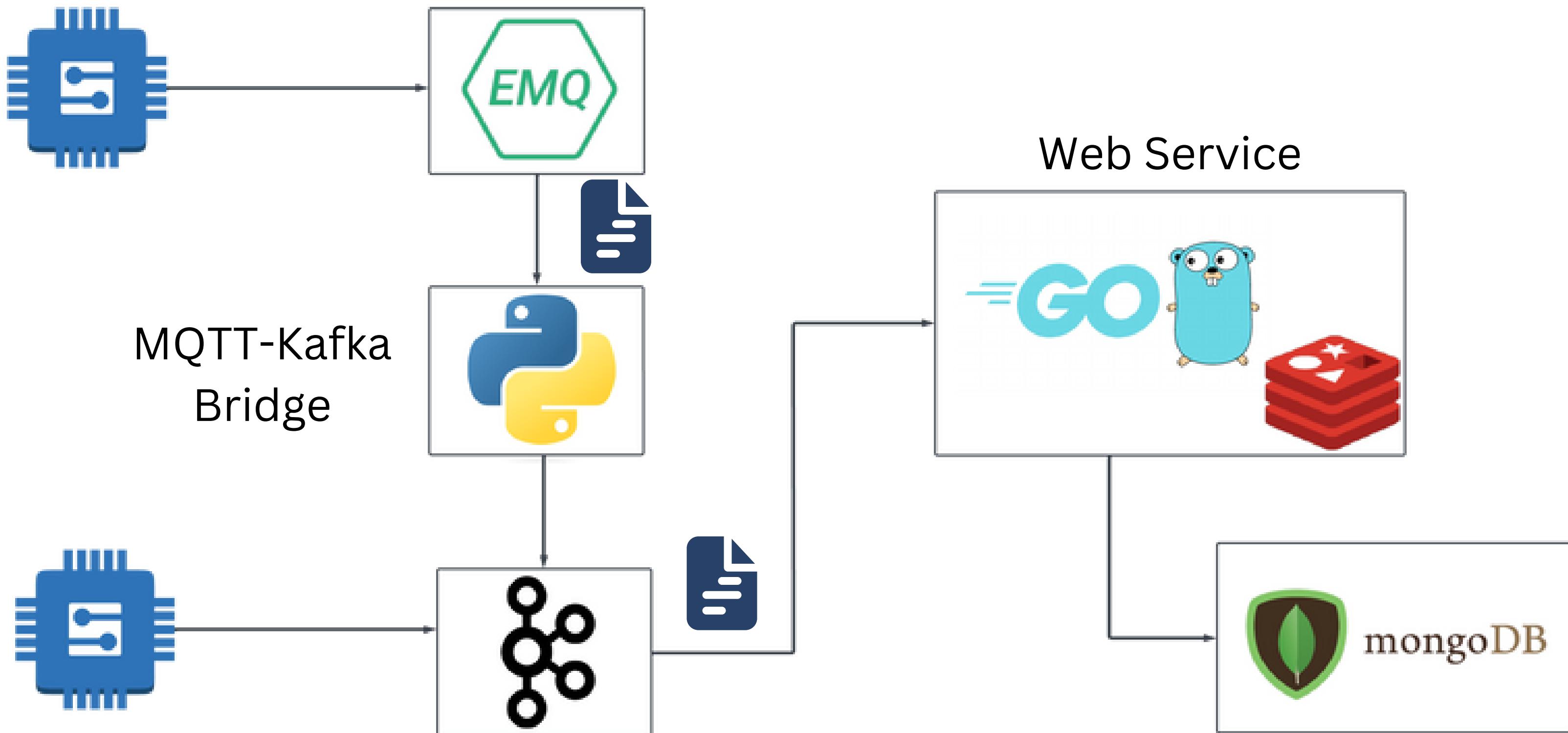


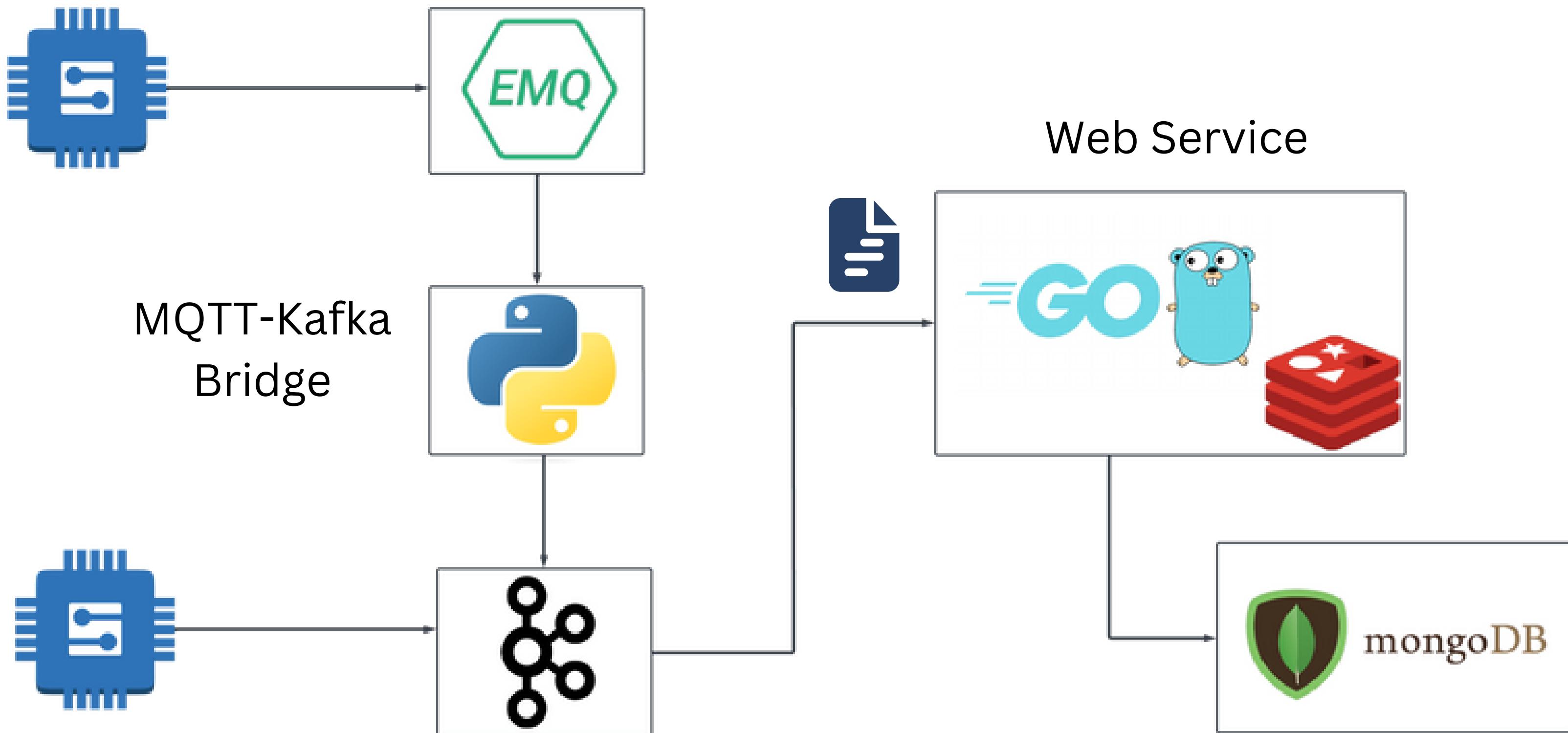
Transmittion Part

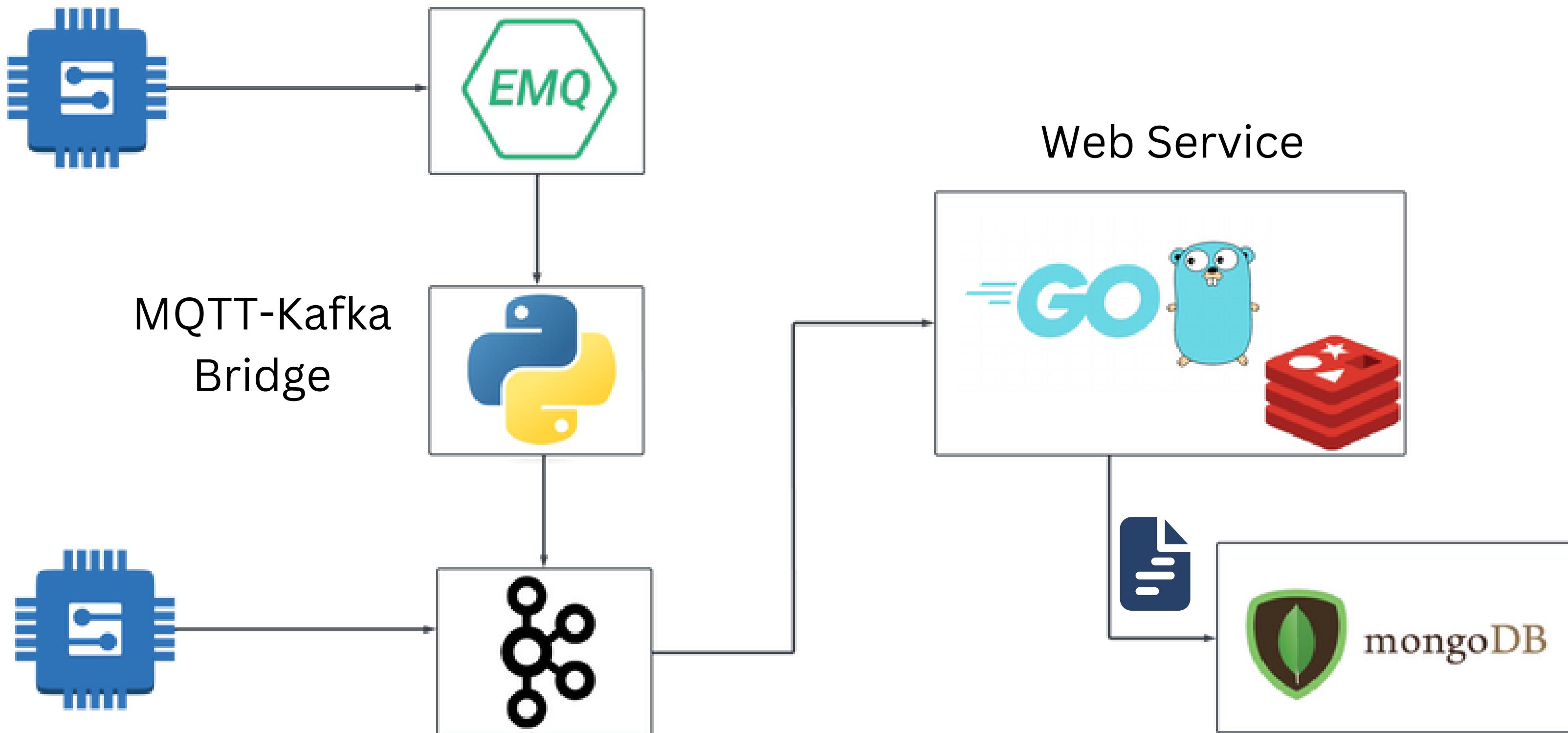
Receiving Part











Transmission part



Providing data to the application

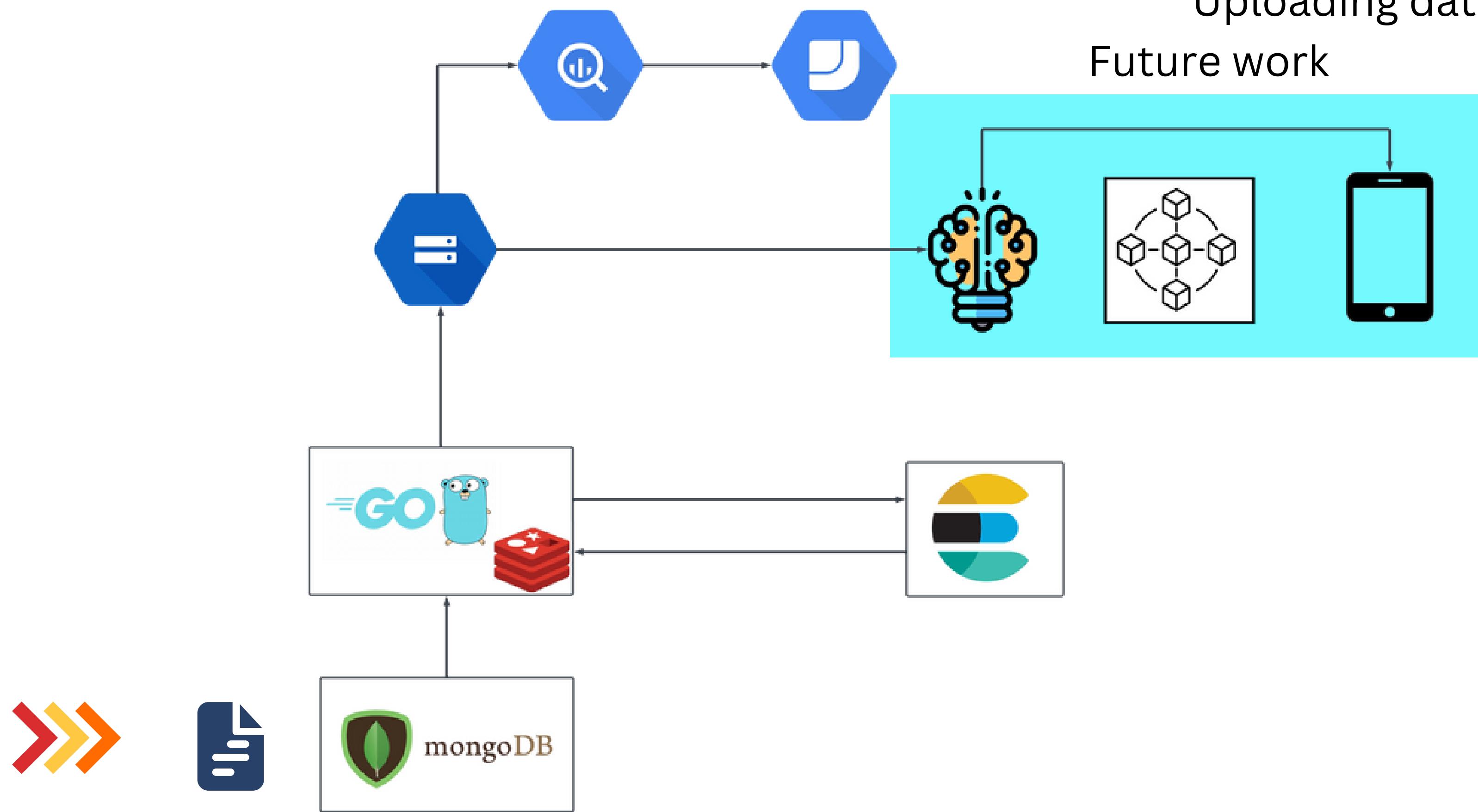


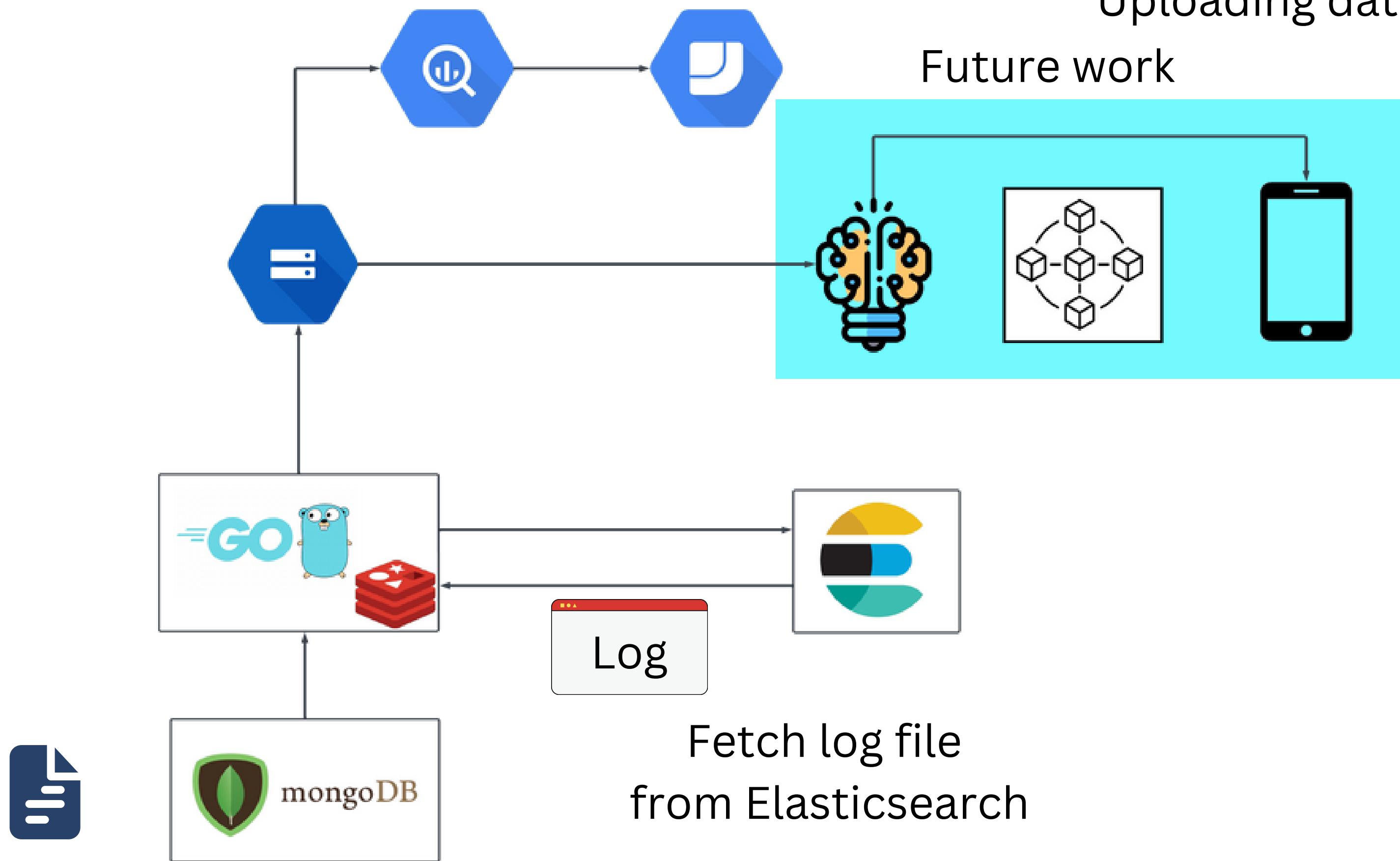
Transmission part

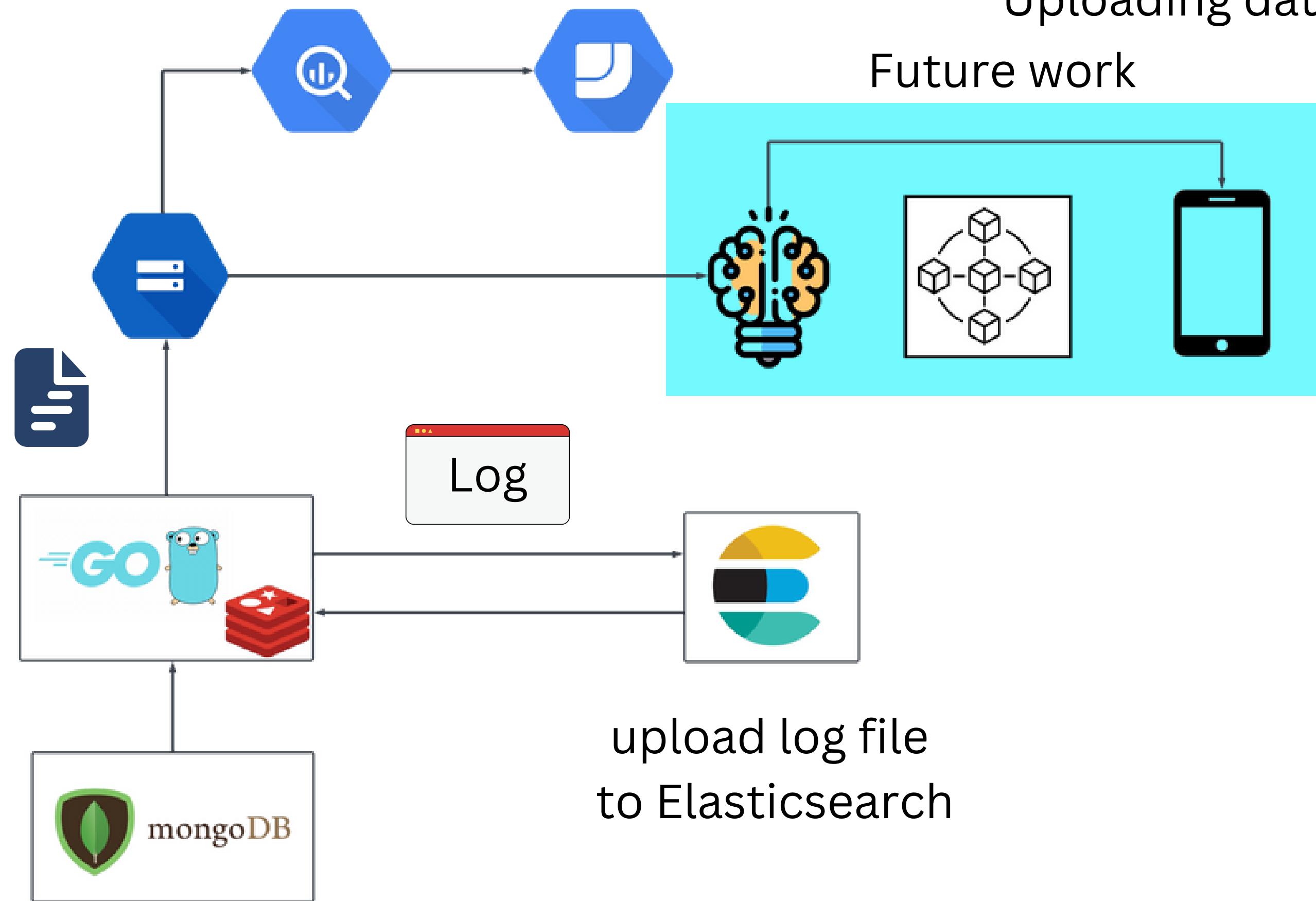
Uploading data to GCS
for data analysis

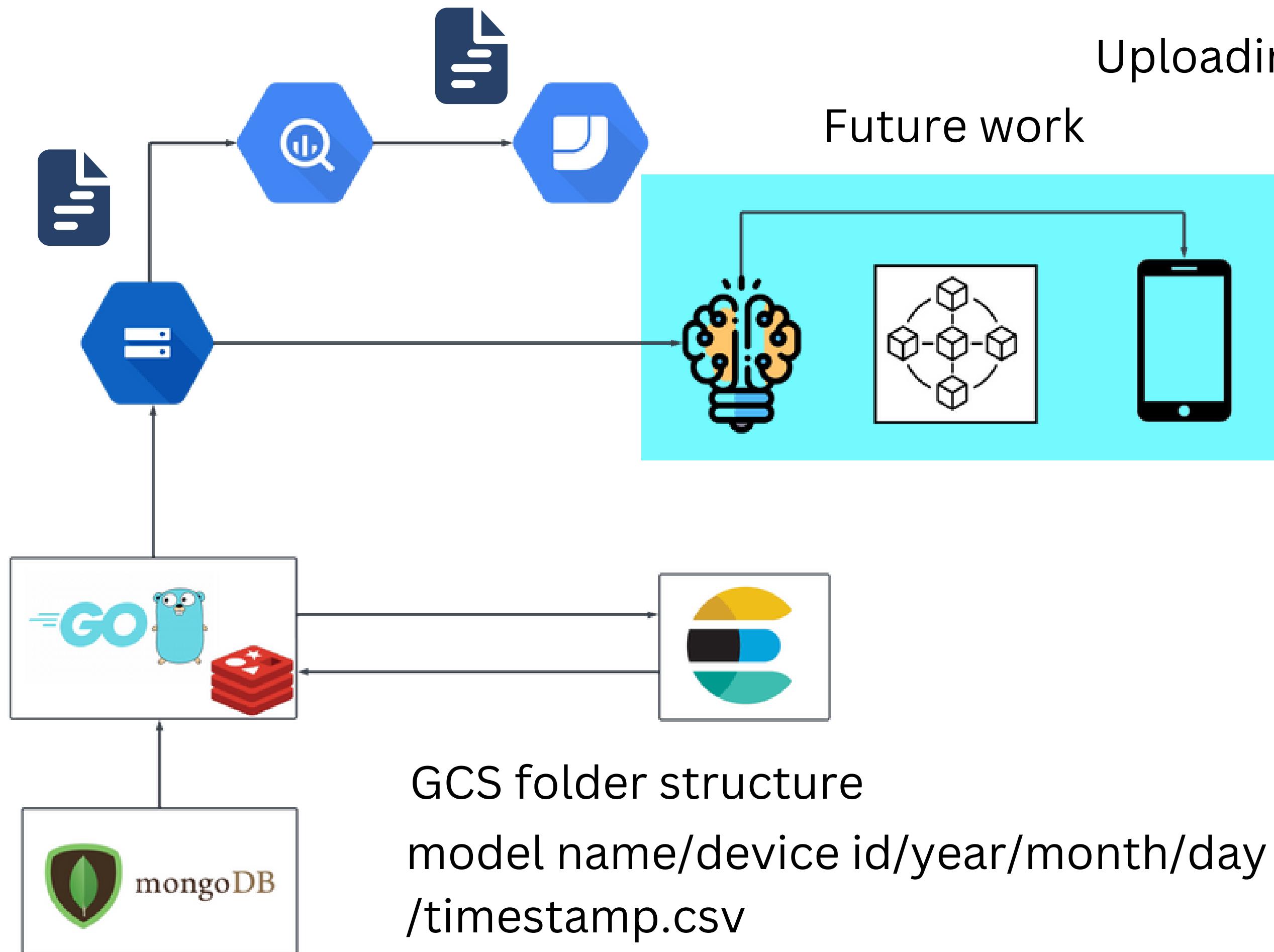
Uploading data to GCS for data analysis



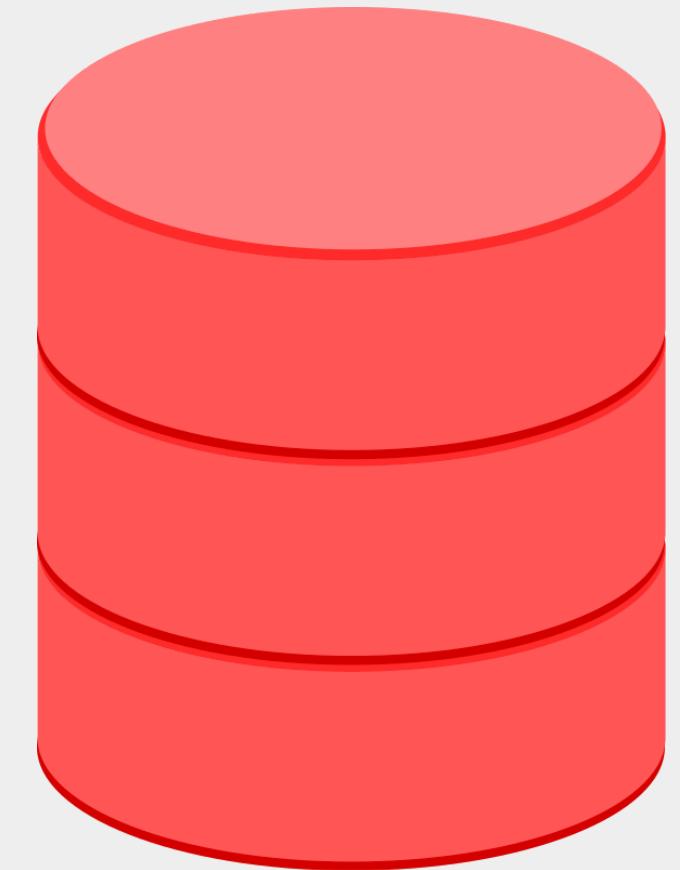


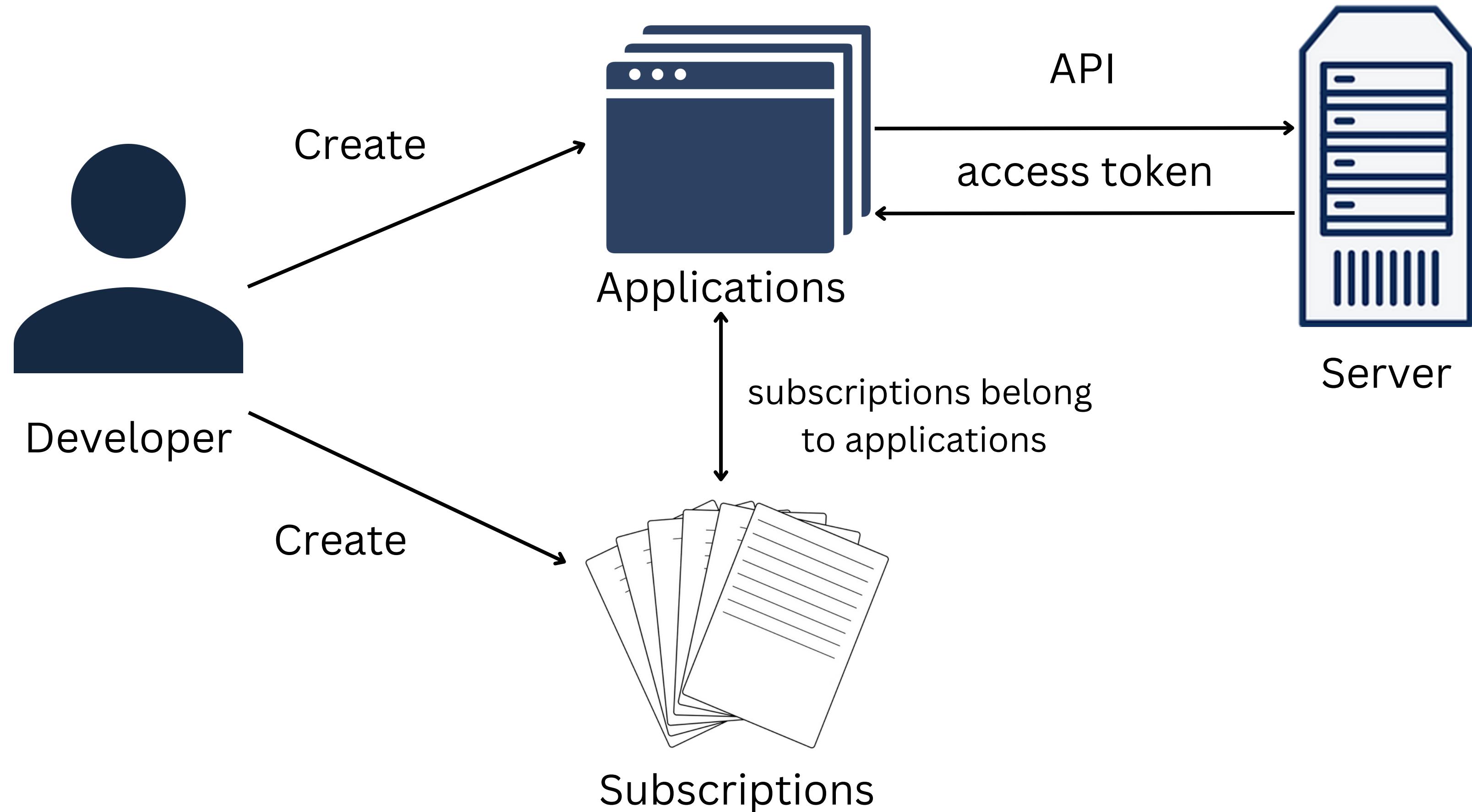


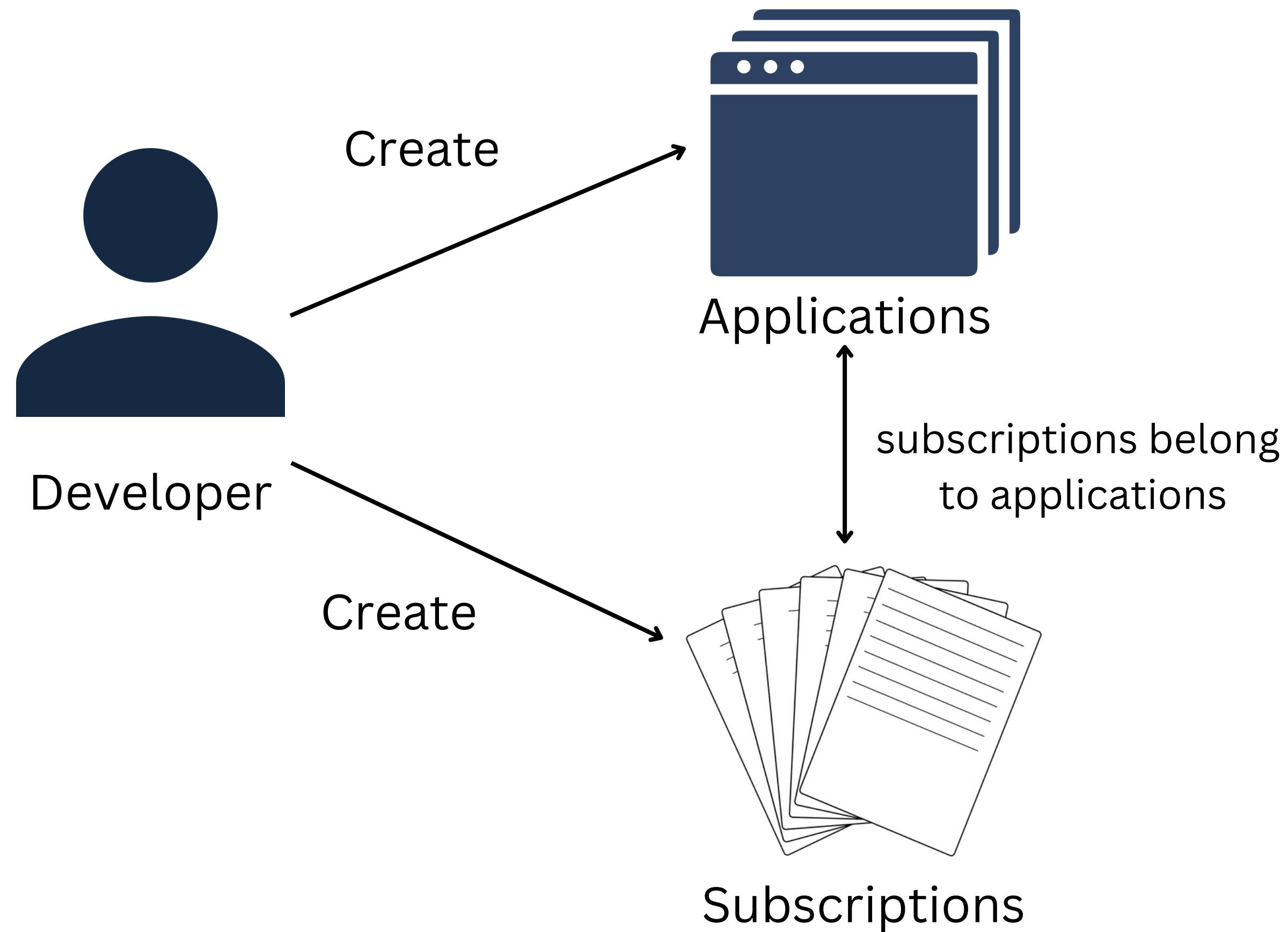


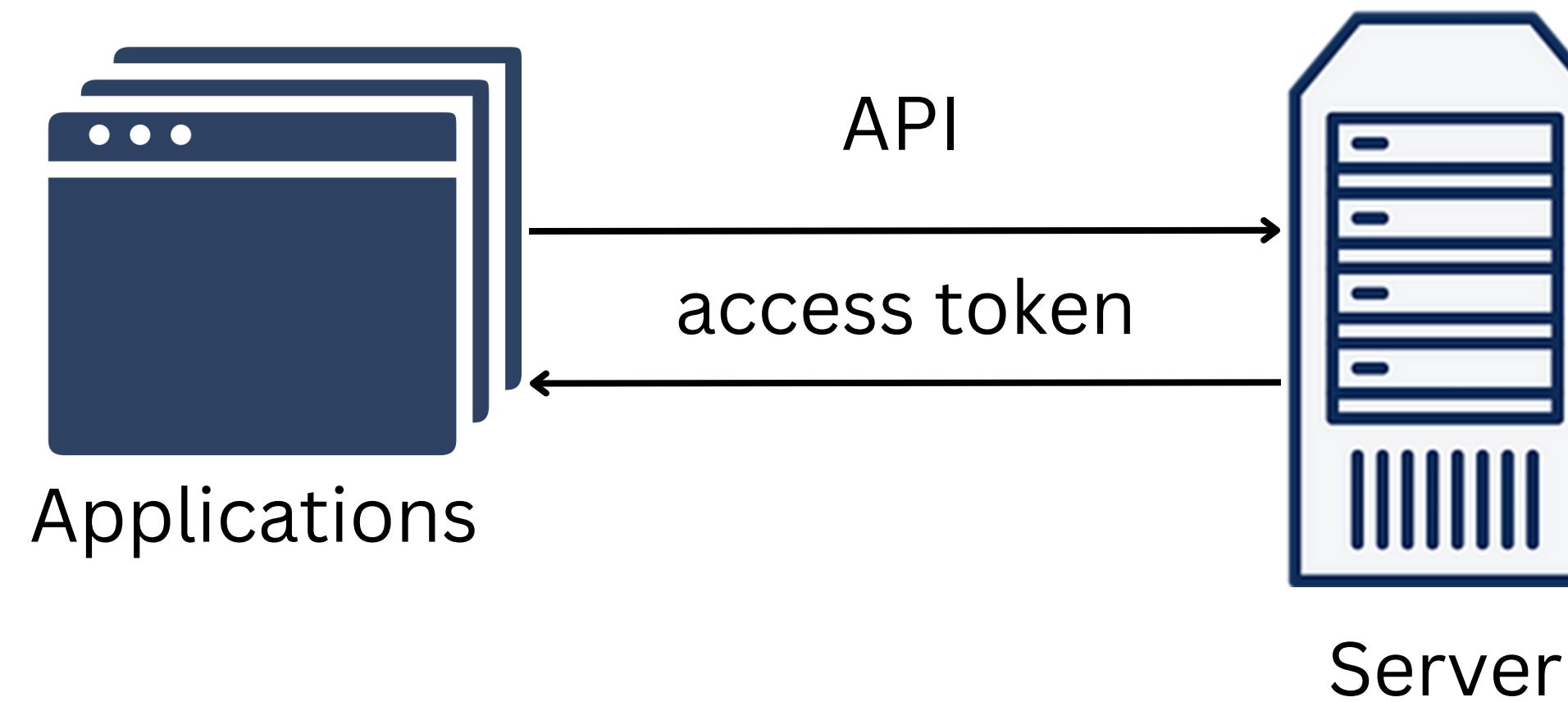


Providing data to the application







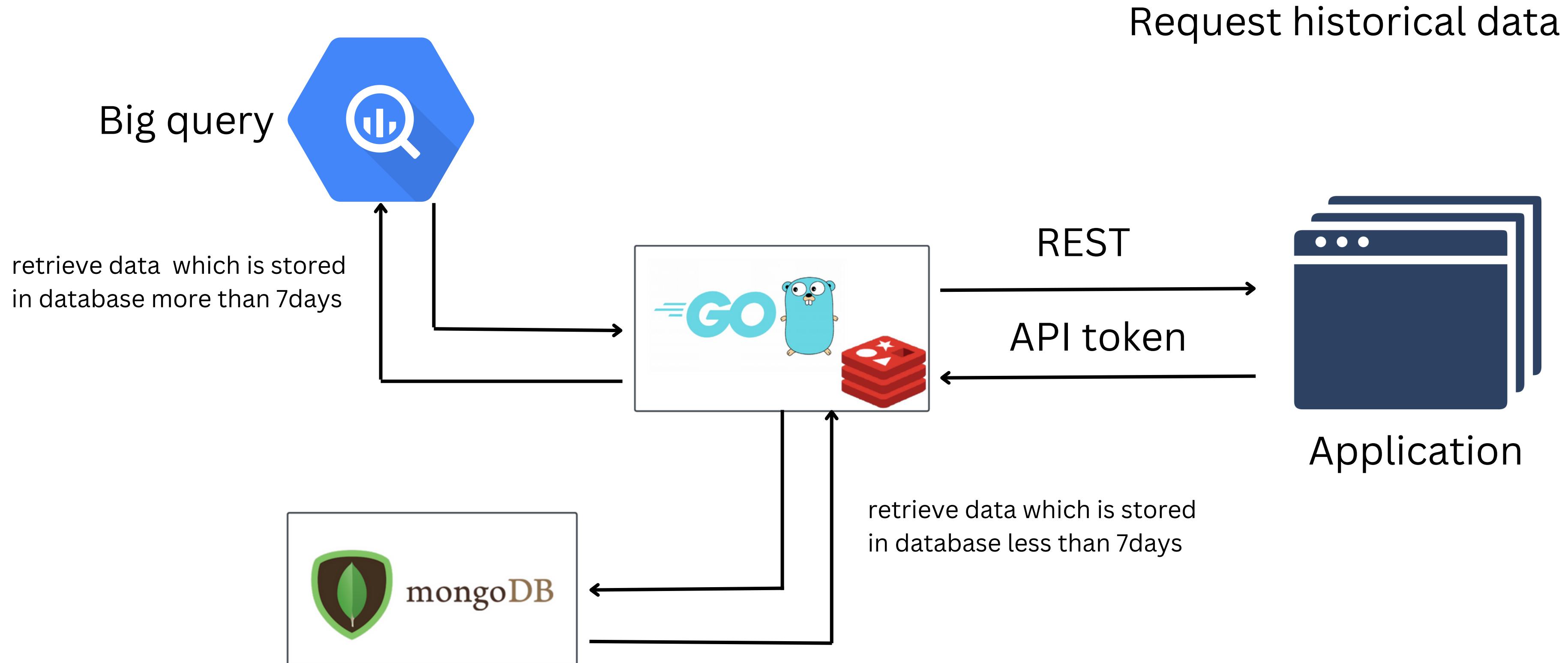


1

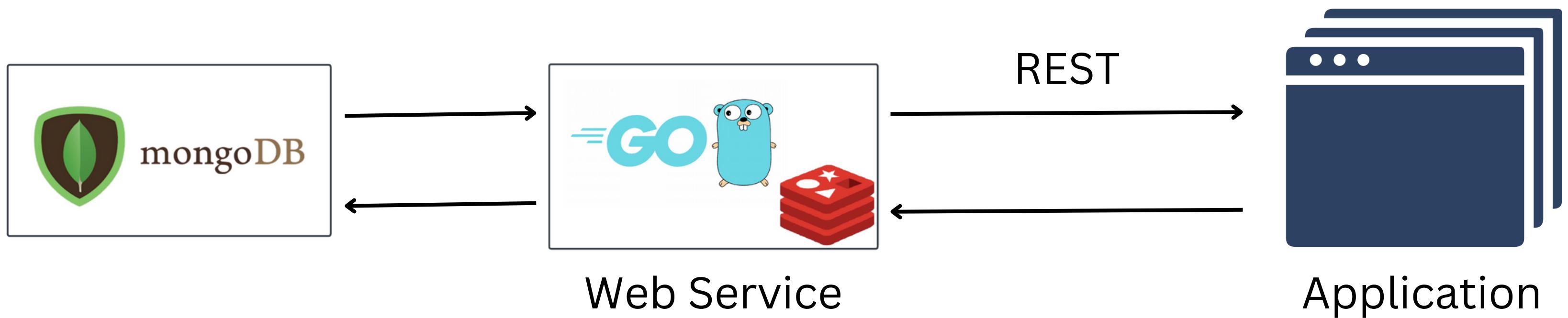
Request historical data

2

Request real-time data

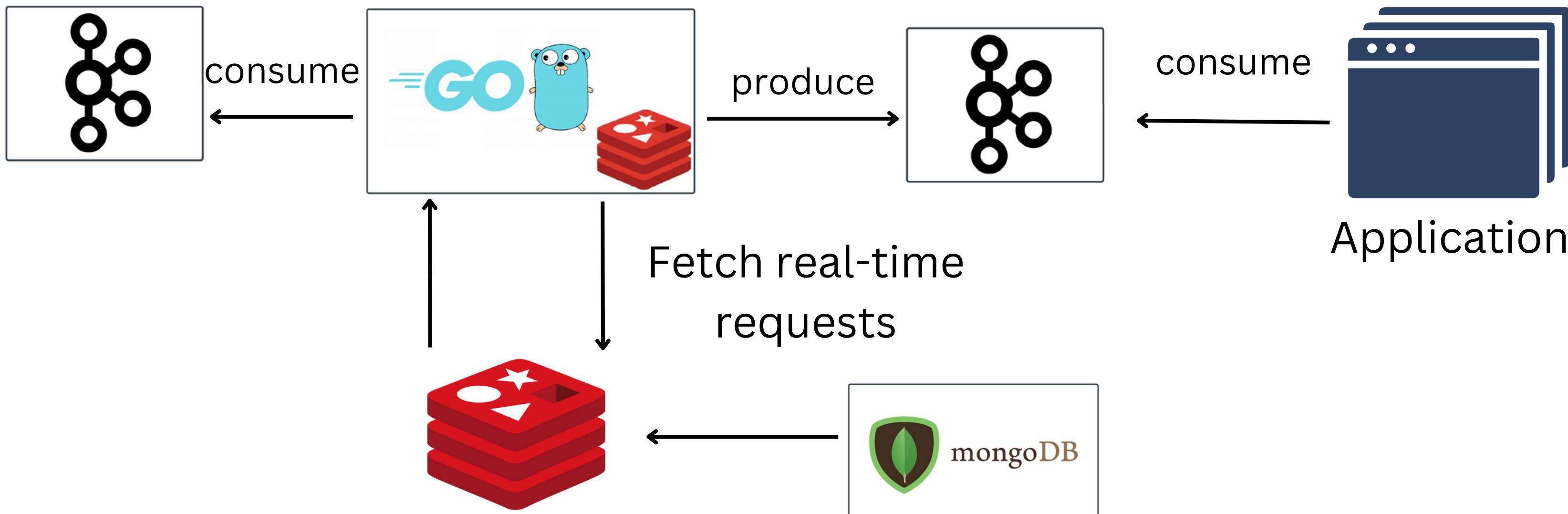


Request real-time data

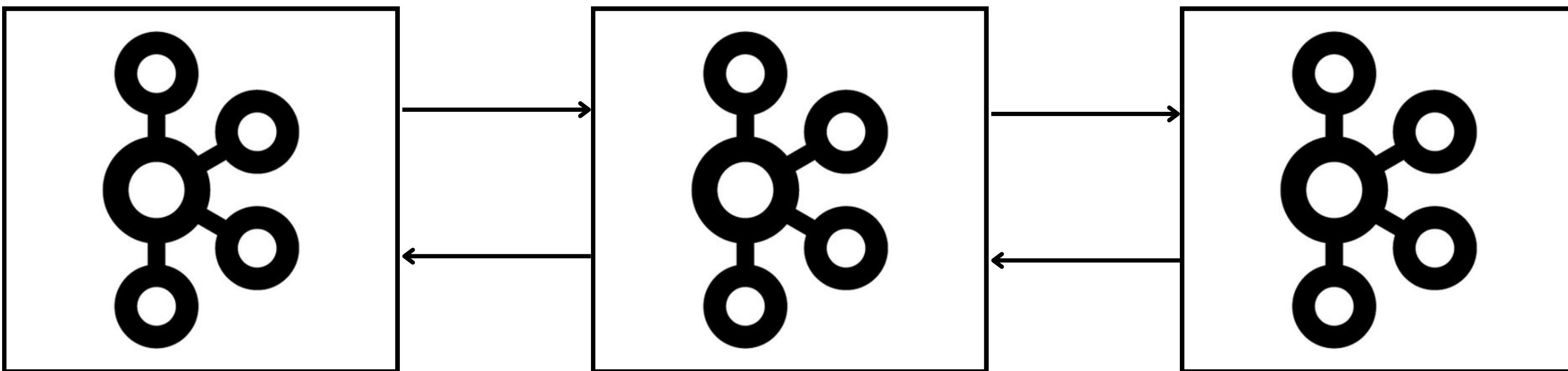


Request real-time data

Kafka



Kafka clusters

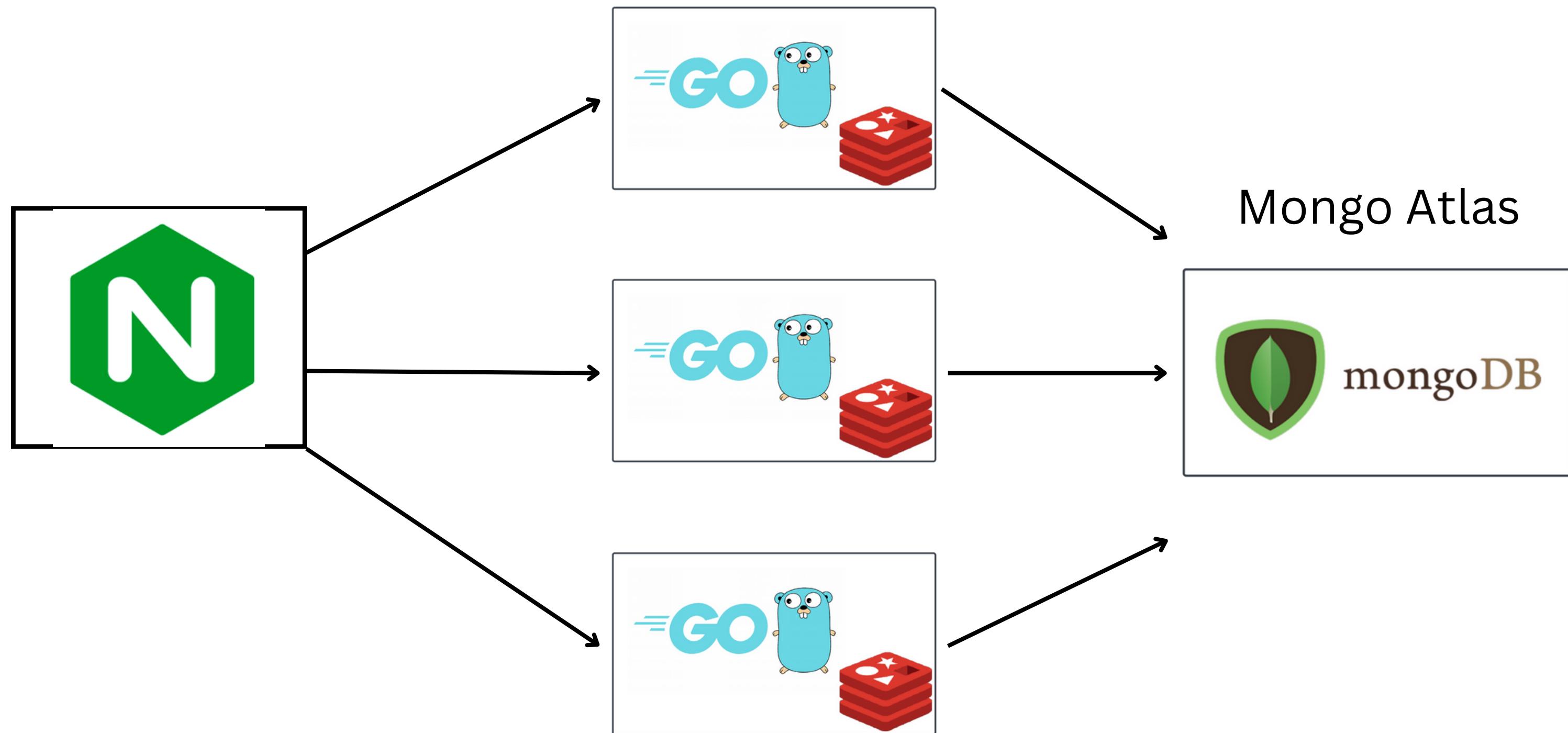


Each instances:

1-broker, 1-zookeeper

Security configuration: SASL_SSL

Deployment



Non-functional requirements



Backup site



Provide data based on
user's authorization

Non-functional requirements



For historical data requests, the system should pass the load test within following criteria

- response 95% of requests within 3 seconds
- average data size = 6000 dp
 - throughput 100 requests/sec
 - 100 devices โดย 1 devices 1 ชั่วโมงจะผลิตข้อมูลมา 60 dp



For future data requests, the system should pass the load test within following criteria

- response 95% of processing requests within 15 seconds
- throughput = 80 message/s

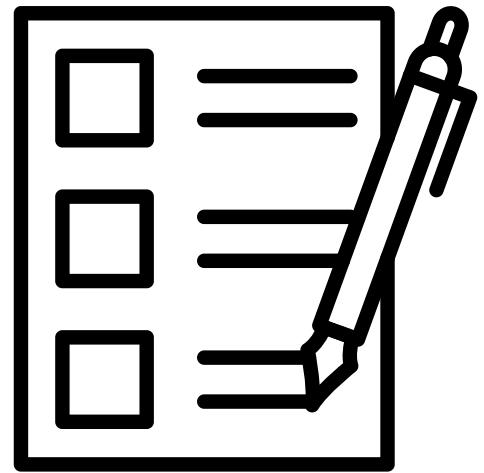
Demo



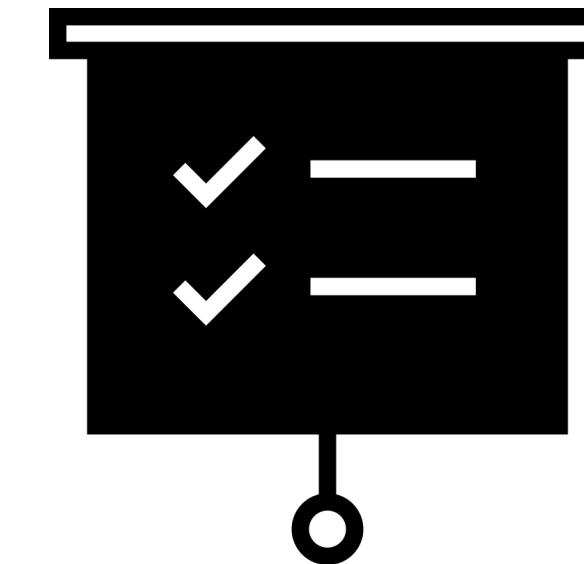
Testing



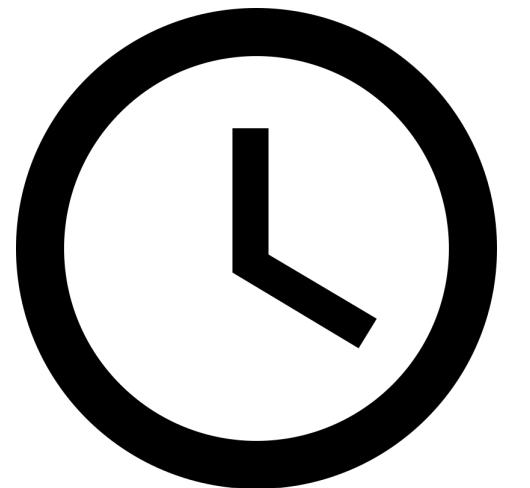
Testing



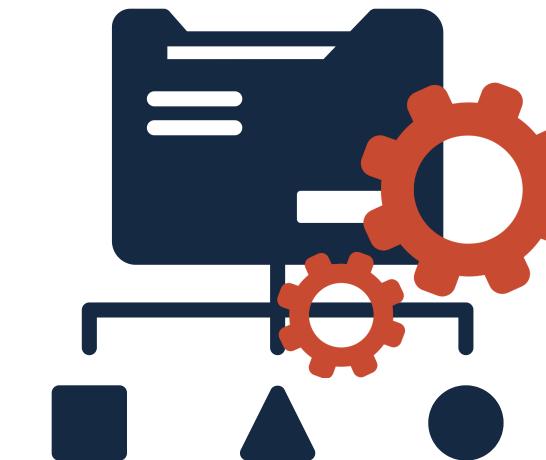
Fundamental Testing



High Availability Testing

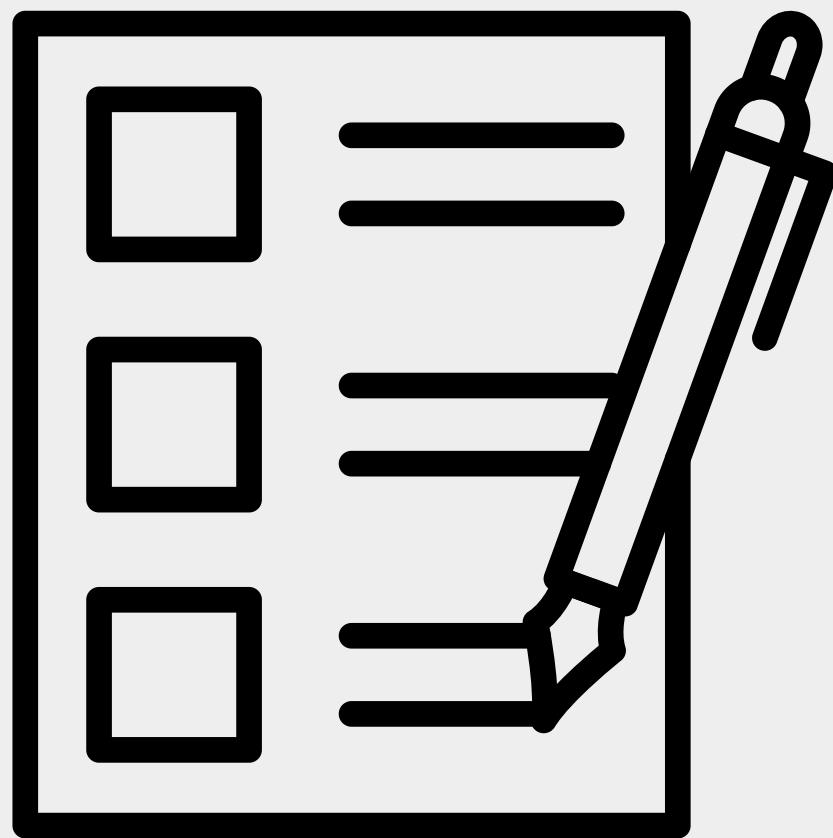


Load testing



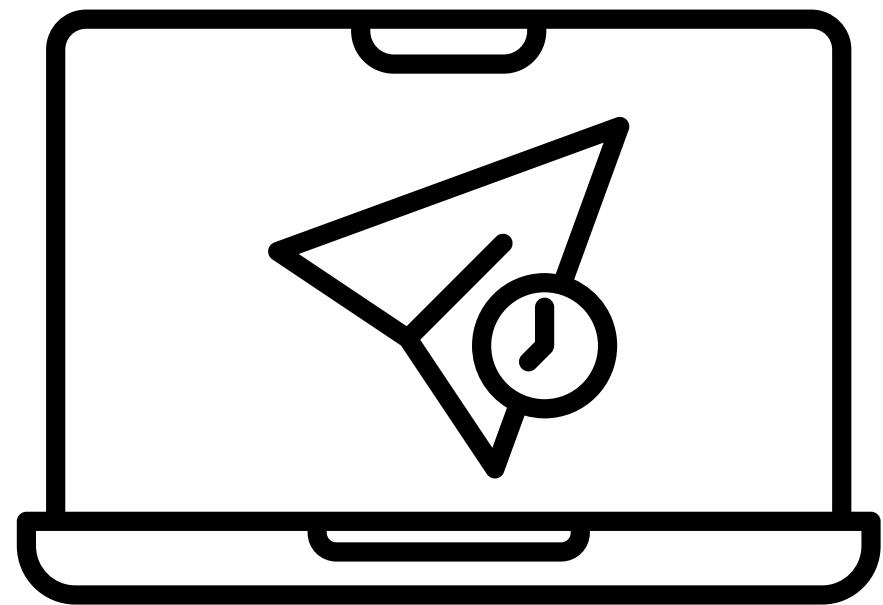
Kafka cluster security

Fundamental Testing

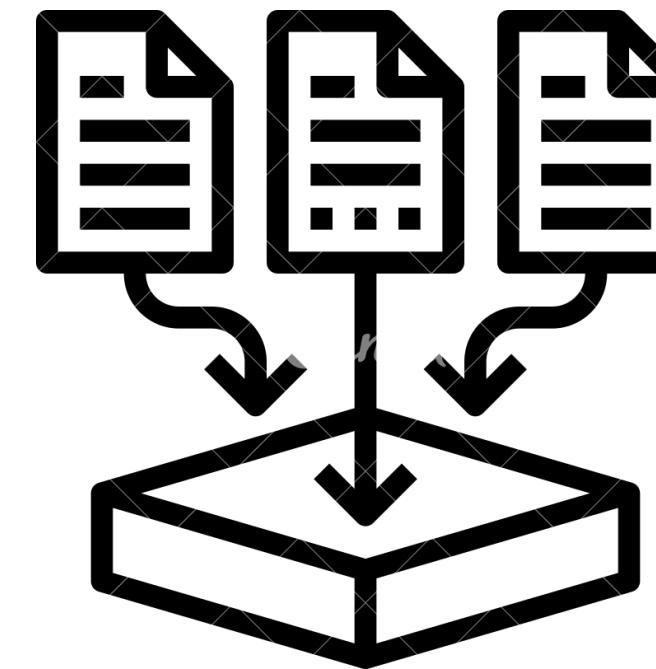


Fundamental testing: MQTT-Kafka bridge

Test scenario



normal case



trigger case

Fundamental testing: MQTT-Kafka bridge

Testing Result

ID	use case	Input	Expected output
1	send 10 data	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 10 differently	-
2	send 100 data	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently	-
3	send 10 data	Topic: newDevice2 { "device_id": "SN-2005-0009", "temperature": 10 } * 10 differently	Topic: newDevice2 { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently
4	send 100 data	Topic: newDevice2 { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently	Topic: newDevice2 { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently

```
ipdMonitoring4 b'{\n    "device_id": "SN-2005-0009",\n    "temperature": 1\n}\n\nipdMonitoring4 b'{\n    "device_id": "SN-2005-0009",\n    "temperature": 2\n}\n\nipdMonitoring4 b'{\n    "device_id": "SN-2005-0009",\n    "temperature": 3\n}\n\nipdMonitoring4 b'{\n    "device_id": "SN-2005-0009",\n    "temperature": 4\n}\n\nipdMonitoring4 b'{\n    "device_id": "SN-2005-0009",\n    "temperature": 5\n}\n\nipdMonitoring4 b'{\n    "device_id": "SN-2005-0009",\n    "temperature": 6\n}\n\nTINFO: 127.0.0.1:55255 "GET /trigger HTTP/1.1" 307 Temporary Redirect
```

Fundamental testing: MQTT-Kafka bridge

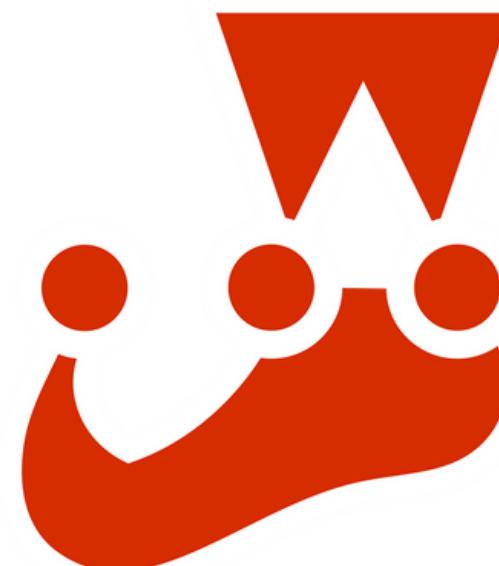
Testing Result

5	send 10 data + trigger 2	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 10 differently	-
6	send 10 data + trigger 8	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 10 differently	-
7	send 100 data + trigger 5	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently	-
8	send 100 data + trigger 10	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently	-
9	send 100 data + trigger 15	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently	-
5	send 10 data + trigger 2	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 10 differently	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 10 differently
6	send 10 data + trigger 8	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 10 differently	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 10 differently
7	send 100 data + trigger 5	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently
8	send 100 data + trigger 10	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently
9	send 100 data + trigger 15	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently	Topic: wrongTopic { "device_id": "SN-2005-0009", "temperature": 10 } * 100 differently

```
INFO: 127.0.0.1:55355 -> /trigger HTTP/1.1 307 TEMPORARY_REDIRECT
['ipdMonitoring4', 'newDevice2', 'ipdMonitoring', 'ipdMonitoring2', 'abc', 'patientMonitor', 'smartPulz', 'ipdMonitoring20', 'ipdMonitoring20
nitoring200', 'smartPulz2', 'smartPulz2', 'ipdMonitoring200', 'ipdMonitoring2', 'ipdMonitoring2', 'eiei', 'haha', 'qqq', 'ipdMonitoring14']
INFO: 127.0.0.1:55355 - "GET /trigger/ HTTP/1.1" 200 OK
haha b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 3\n)\\n'
ipdMonitoring14 b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 1\\n)\\n'
ipdMonitoring14 b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 2\\n)\\n'
ipdMonitoring14 b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 3\\n)\\n'
ipdMonitoring14 b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 4\\n)\\n'
ipdMonitoring4 b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 1\\n)\\n'
ipdMonitoring4 b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 2\\n)\\n'
ipdMonitoring4 b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 3\\n)\\n'
ipdMonitoring4 b'(\n    "device_id": "SN-2005-0009",\n    "temperature": 4\\n)\\n'
```

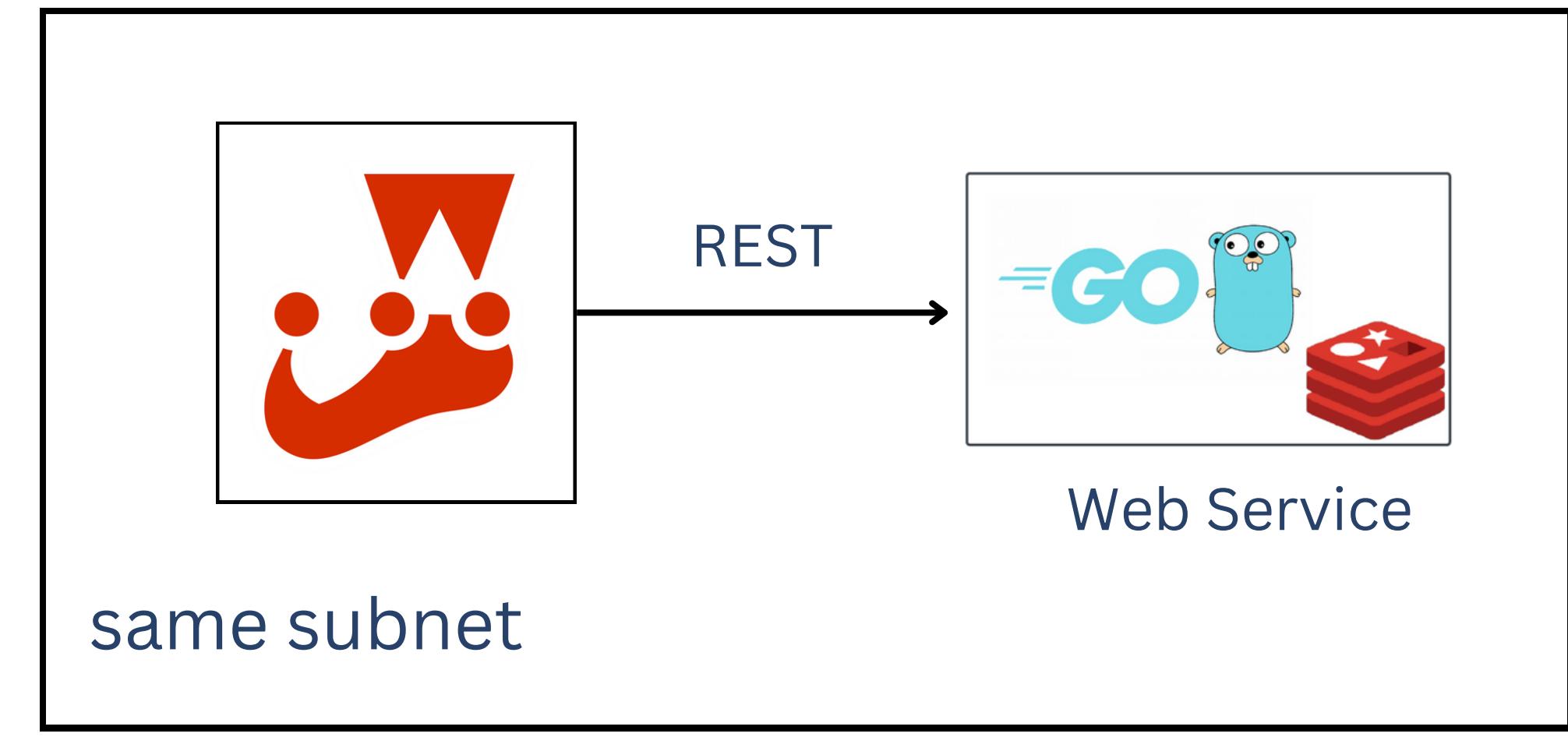
Fundamental testing: request medical data

Testing tool



Jest

Test environment



Test scenario: request medical data

Historical data	Real-time data
start-time or end-time is not allow	start-time or end-time is not allow
modelName or devicId is not allow	modelName or devicId is not allow
Unauthorized [Invalid token]	Unauthorized [Invalid token]
Successfully request data	Successfully request data
Data not found	

Test result: request medical data

```
yarn run v1.22.19
$ jest --detectOpenHandles
PASS  src/test/historical-data.spec.ts (7.241 s)
  Historical Data Testing
    ✓ Forbidden [start-time or end-time is not allowed] (458 ms)
    ✓ Forbidden [modelName or deviceId is not allowed] (354 ms)
    ✓ Unauthenticated [invalid token] (19 ms)
    ✓ Data Not found (530 ms)
    ✓ Successfully request data (713 ms)
  Request Real-time Data Testing
    ✓ Forbidden [start-time or end-time is not allowed] (363 ms)
    ✓ Forbidden [modelName or deviceId is not allowed] (374 ms)
    ✓ Unauthenticated [invalid token] (16 ms)
    ✓ Successfully request data (539 ms)

Test Suites: 1 passed, 1 total
Tests:       9 passed, 9 total
Snapshots:  0 total
Time:        7.424 s
Ran all test suites.
Done in 8.90s.
```

Fundamental testing (ui)

- Developer
 - My Application
 - My Subscription
 - My Account
- Admin
 - Device Management
 - Developer Management
 - Subscription Management
- Authentication
 - Login-Logout
 - Protected Route



Fundamental testing: Authentication

- Login-Logout
- Protected Route

Description	Input	Expected output	Pass/Fail
email or password (required) is empty	Email: "" Password: ""	ask user to enter email and password	Pass
email format is wrong and password is not empty	Email: 'abc' Password: 'password'	warning email is wrong format	Pass
email format is correct and password is correct	Email: 'abc@def.g' Password: 'password'	successfully login	Pass
email format is correct but not exists	Email: 'a@b.com' Password: 'password'	invalid email or password	Pass
email is correct and password is not correct	Email: 'abc@def.g' Password: 'wrongpassword'	invalid email or password	Pass
logout	-	clear token and redirect to login page	Pass
not login	-	Warning and redirect to login page	Pass



Fundamental testing: Developer

Description	Input	Expected output	Pass/Fail	Description	Input	Expected output	Pass/Fail																
get all of my applications	-	list all of my applications	Pass	get all of my subscriptions	-	list all of my subscriptions	Pass																
new application required fields(appname) are empty	AppName: "" Description: ""	ask user to enter app name	Pass	new subscription some of required fields are empty	application = "" model = "" device= [] starttime = "" endtime = "" mode = ""	ask user to input those fields	Pass																
new application app name is repetitive	AppName: 'my-app' Description: "	warning app name must be unique	Pass	new subscription	application = 'my-app' model = 'ipdMonitoring' device= ['SN-2005-0005'] starttime = '10/4/2023 23:00' endtime = '11/4/2023 23:00' mode = 'REST'	successfully create new subscription	Pass																
new application all required fields are complete	AppName: 'my-app-2' Description: 'this is my wonderful app'	successfully create a new app	Pass	new subscription all required fields are not empty	device= 'SN-2005-0005' model = 'ipdMonitoring' starttime = '10/4/2023 23:00' endtime = '11/4/2023 23:00' mode = 'REST' status = 'pending' application = 'my-app'	successfully create new subscription	Pass																
edit app info edit app name to be the name that is already used	AppName: 'my-app'	warning app name must be unique	Pass	filter subscriptions		show all subscriptions that match ter filter	Pass																
edit app info edit app name to be the name that isn't already used	AppName: 'my-app-2'	successfully update application info	Pass	ask user to enter new password	Pass																		
edit app info edit description	Description: 'this is my wonderful app'	successfully update application info	Pass	ask user to enter re-new password	Pass																		
renew client secret	-	client secret updated	Pass	warning the passwords aren't match	Pass																		
delete an app	-	successfully delete app	Pass	warning old password is invalid	Pass																		
list all subscription of this app	<table border="1"> <thead> <tr> <th>Description</th> <th>Input</th> </tr> </thead> <tbody> <tr> <td>get account info</td> <td>-</td> </tr> <tr> <td>change password old password is empty</td> <td>OldPassword: ""</td> </tr> <tr> <td>change password new password is empty</td> <td>NewPassword: ""</td> </tr> <tr> <td>change password re-new password is empty</td> <td>ReNewPassword: ""</td> </tr> <tr> <td>change password new password and re-new password aren't match</td> <td>NewPassword: "newpass" ReNewPassword: "newpassword"</td> </tr> <tr> <td>change password wrong password</td> <td>OldPassword: "wrong" NewPassword: "newpassword" ReNewPassword: "newpassword"</td> </tr> <tr> <td>change password correct password</td> <td>OldPassword: "password" NewPassword: "newpassword" ReNewPassword: "newpassword"</td> </tr> </tbody> </table>	Description	Input	get account info	-	change password old password is empty	OldPassword: ""	change password new password is empty	NewPassword: ""	change password re-new password is empty	ReNewPassword: ""	change password new password and re-new password aren't match	NewPassword: "newpass" ReNewPassword: "newpassword"	change password wrong password	OldPassword: "wrong" NewPassword: "newpassword" ReNewPassword: "newpassword"	change password correct password	OldPassword: "password" NewPassword: "newpassword" ReNewPassword: "newpassword"	successfully update password	Pass				
Description	Input																						
get account info	-																						
change password old password is empty	OldPassword: ""																						
change password new password is empty	NewPassword: ""																						
change password re-new password is empty	ReNewPassword: ""																						
change password new password and re-new password aren't match	NewPassword: "newpass" ReNewPassword: "newpassword"																						
change password wrong password	OldPassword: "wrong" NewPassword: "newpassword" ReNewPassword: "newpassword"																						
change password correct password	OldPassword: "password" NewPassword: "newpassword" ReNewPassword: "newpassword"																						



- My Application
- My Subscription
- My Account

Fundamental testing: Admin

- Device Management
- Developer Management
- Subscription Management

	page	Description	Input	Expected output	Pass/Fail
1	device management page	get all medical models		list all of medical models	Pass
2	device management page	delete a medical model		successfully delete medical model	Pass
3	device management page	Create new medical model. All fields are empty.	ModelName: "" ModelKey: [] NewKeys: []	ask user to enter model name and model key	Pass
4	device management page	Create new medical model. Model Name are empty. Model Keys are chosen New Keys are not added	ModelName: "" ModelKey: ["heartRate", "bloodPressure"] NewKeys: []	ask user to enter model name	Pass
5	device management page	Create new medical model. Model Name are correct Model Keys are empty New Keys are not added	ModelName: "SN-1005-2" ModelKey: [] NewKeys: []	ask user to enter model key	Pass
6	device management page	Create new medical model. Model Name are correct Model Keys are chosen New Keys are not added	ModelName: "SN-1005-2" ModelKey: ["heartRate", "bloodPressure"] NewKeys: []	successfully create new medical model	Pass
7	device management page	Create new medical model. Model Name are correct Model Keys are chosen. New Keys are added and empty.	ModelName: "SN-1005-2" ModelKey: ["heartRate", "bloodPressure"] NewKeys: [""]	ask user to enter new model key	Pass
8	device management page	Create new medical model. Model Name are correct Model Keys are chosen New Keys are added and correct	ModelName: "SN-1005-2" ModelKey: ["heartRate", "bloodPressure"] NewKeys: ["o2"]	successfully create new medical model	Pass

	page	Description	Input	Expected output	Pass/Fail
9	developer management page	get all developers		list all of developers	Pass
10	developer management page	delete a developer		successfully delete developer	Pass
11	developer management page	Create new developer Email is empty	Email: "" Name: "developer" AssociatedName: "hospital"	ask user to enter Email	Pass
12	developer management page	Create new developer Email has incorrect format	Email: "developer" Name: "developer" AssociatedName: "hospital"	ask user to enter Email in format	Pass
13	developer management page	Create new developer Email has correct format and is duplicate	Email: "developer@gmail.com" Name: "developer" AssociatedName: "hospital"	ask user to enter Email that not already create	Pass
14	developer management page	Create new developer Name are empty	Email: "developer001@gmail.com" Name: "developer" AssociatedName: "hospital"	ask user to enter Name	Pass
15	developer management page	Create new developer AssociatedName are empty	Email: "developer001@gmail.com" Name: "developer" AssociatedName: "hospital"	ask user to enter AssociatedName	Pass
16	developer management page	Create new developer All field are correct	Email: "developer001@gmail.com" Name: "developer" AssociatedName: "hospital"	successfully create new developer	Pass

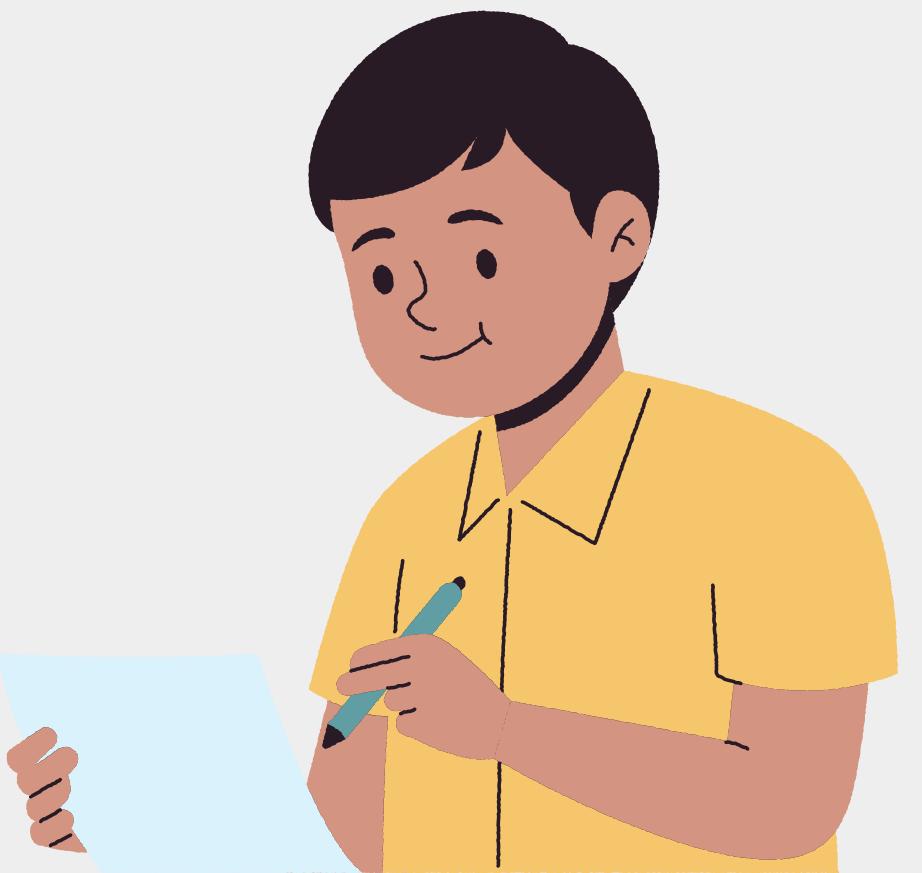
	page	Description	Input	Expected output	Pass/Fail
17	subscription management page	List all subscriptions that are pending and accepted		list all subscriptions that are pending and accepted	Pass
18	subscription management page	Accept subscription		successfully accept subscription	Pass
19	subscription management page	Reject subscription		successfully reject subscription	Pass
20	subscription management page	Revoke subscription		successfully revoke subscription	Pass
21	subscription management page	Filter by status (Revoked)	status = "REVOKED"	list all subscriptions that are revoked	Pass



Demo



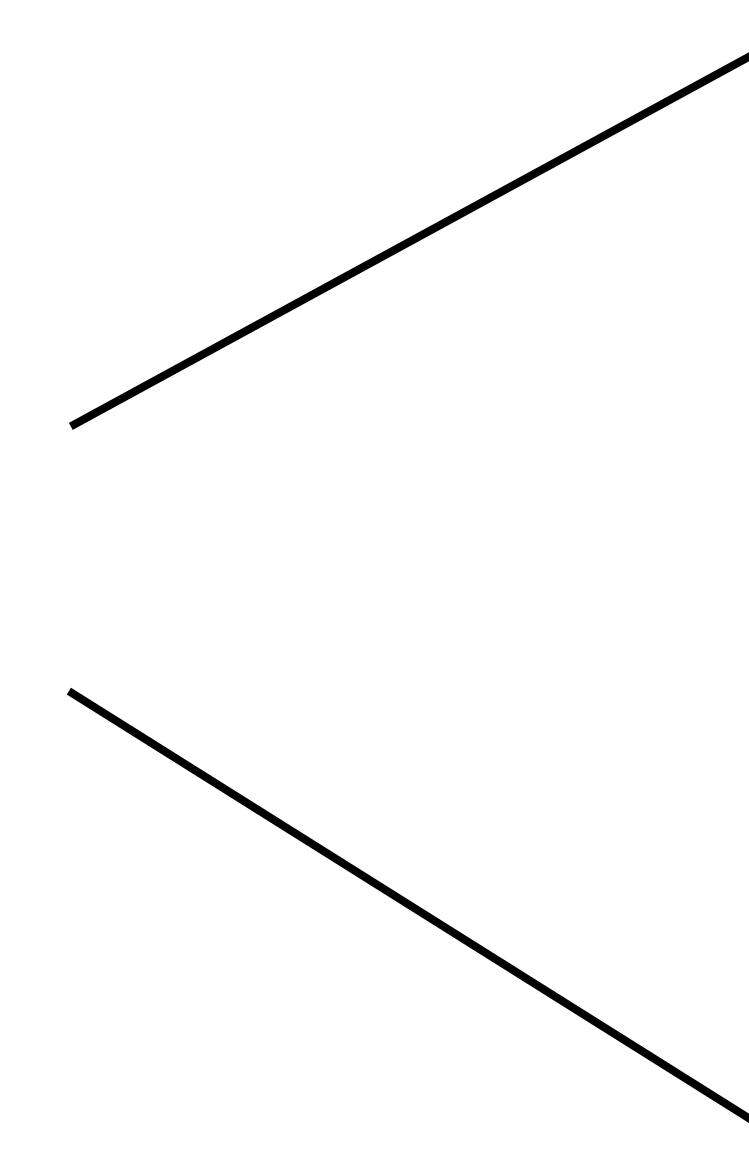
Load Testing



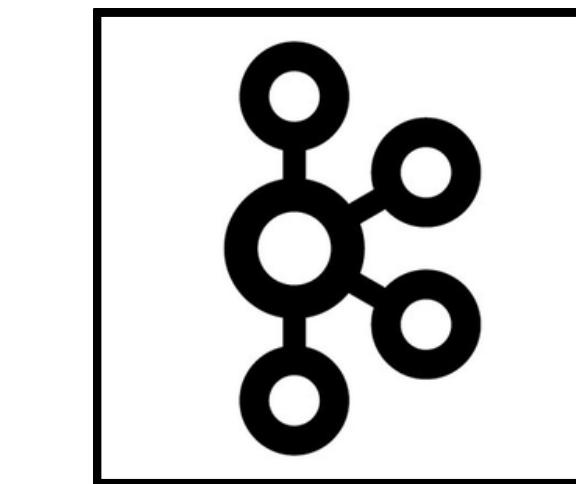
Load testing



Applications



Request historical data



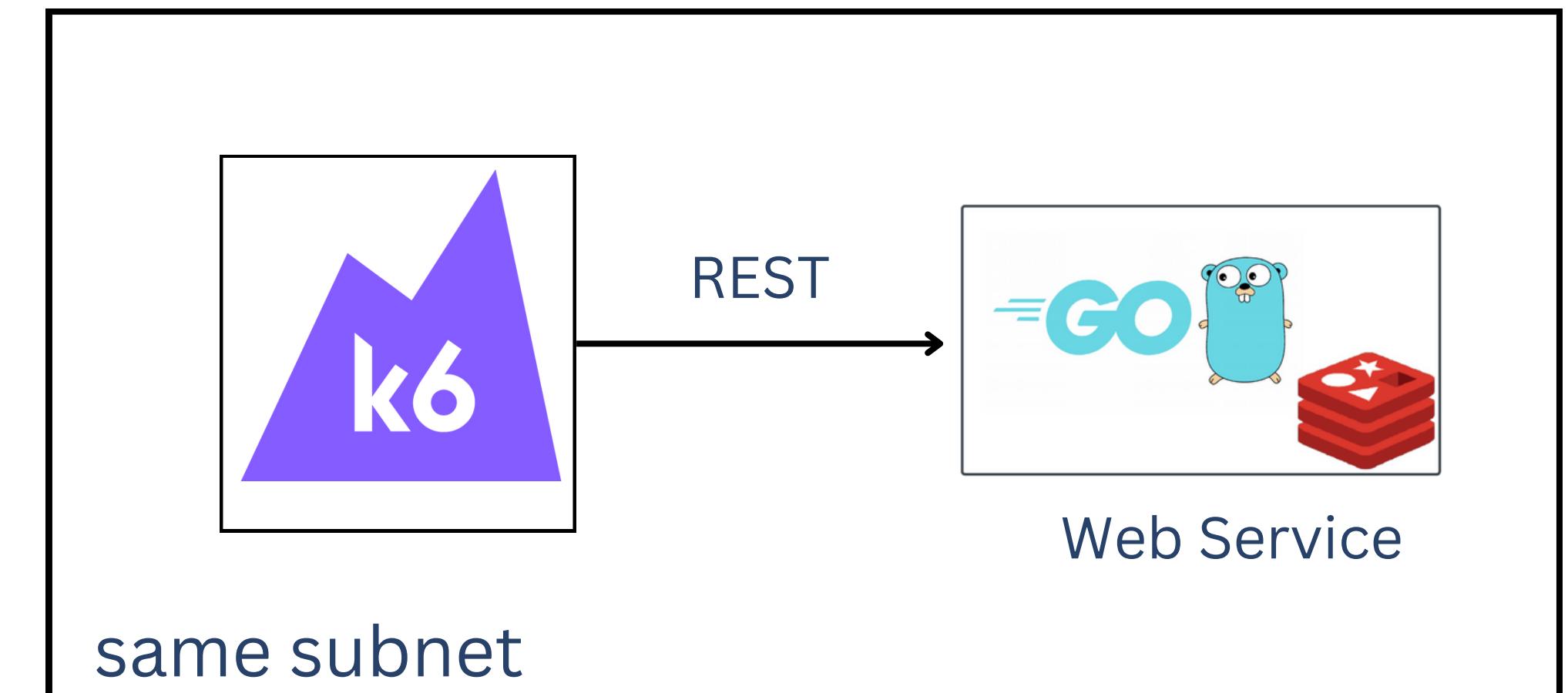
Request real-time data

Request historical data

Testing tool



Test environment



Test scenario

Throughput
100 requests/sec



size of response body
60 dp



Throughput = 100 requests/sec & Response body = 60 dp



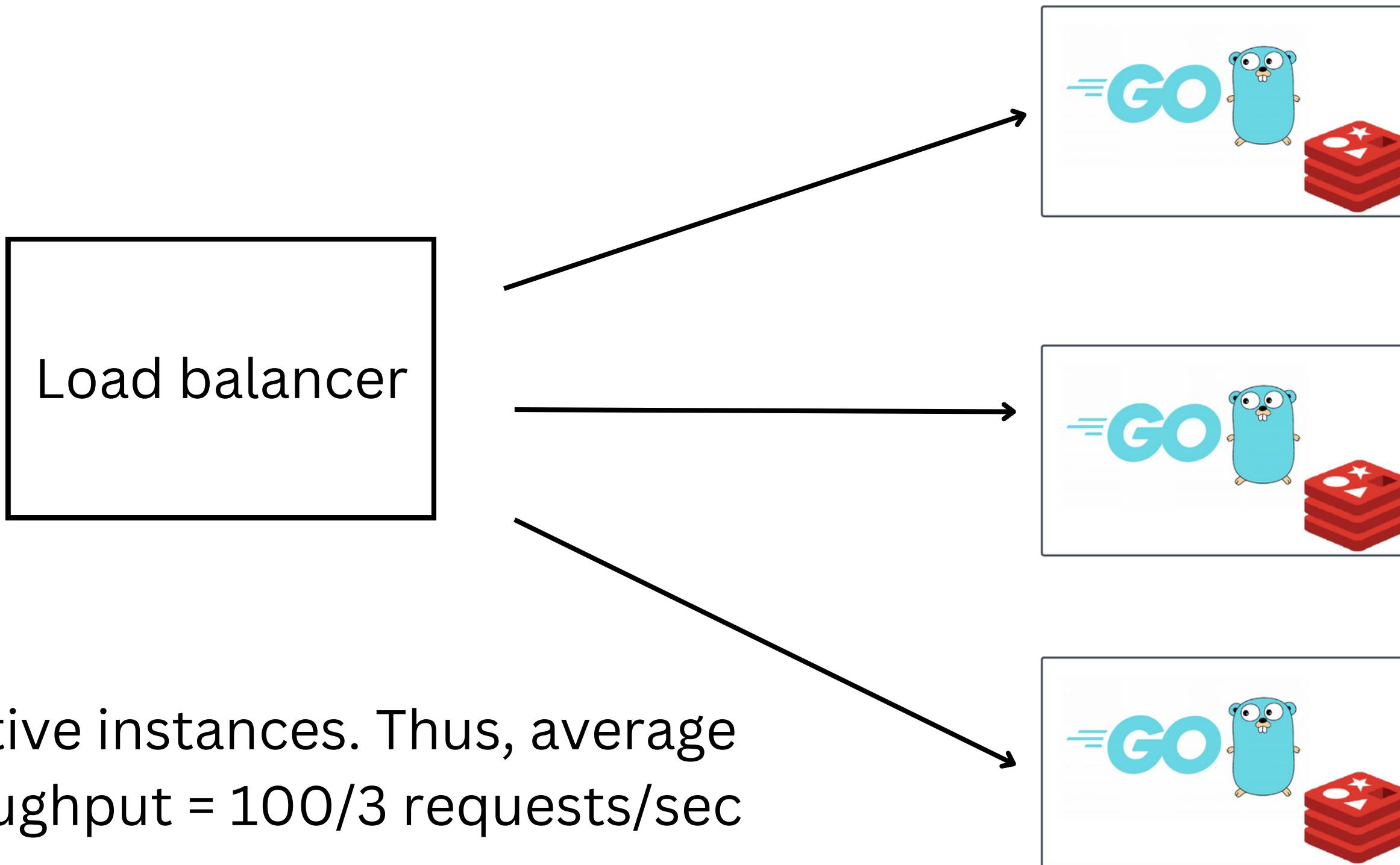
100 devices

- Most of devices produce data 60 dp/hour
- Recommend developer to paginate data ~ 60 dp/page
- Assume there are 100 active applications in the system
[1 application per 1 devices]



Throughput = 100 requests/sec
Response body = 60 dp

Throughput = 36.4 requests/sec



Expected result



response 95% of requests within 3 seconds

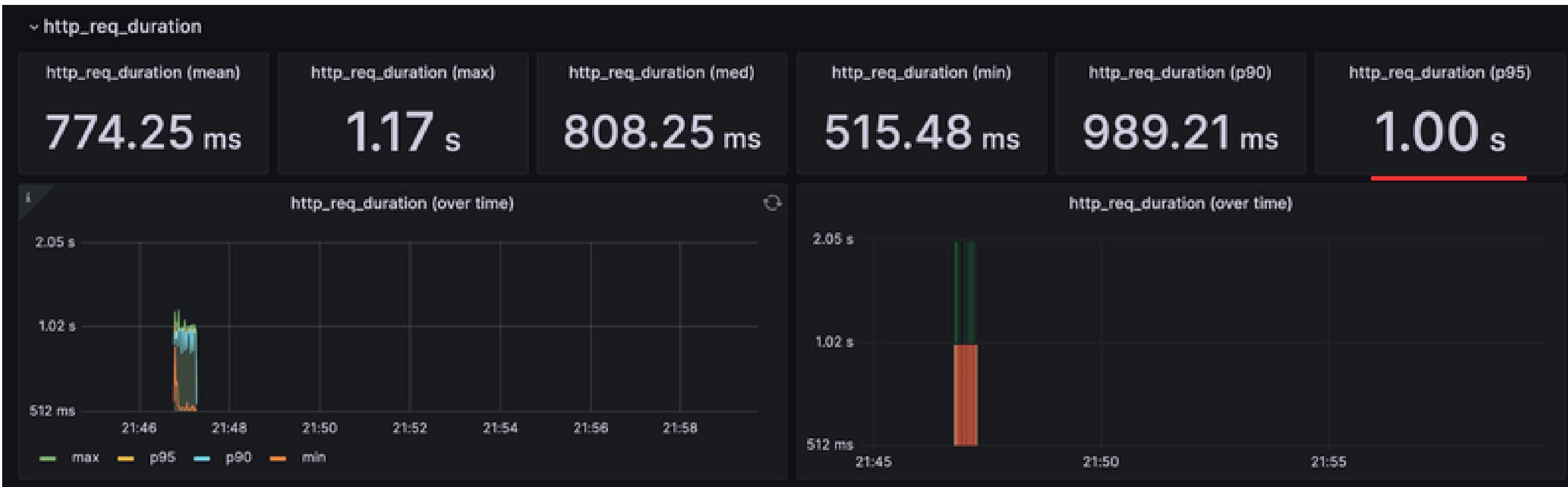
Test Result

```
✓ response code was 200

checks.....: 100.00% ✓ 1129 x 0
data_received.....: 18 MB 585 kB/s
data_sent.....: 549 kB 18 kB/s
http_req_blocked.....: avg=15.96µs min=1.5µs med=3.35µs max=2.01ms p(90)=5.19µs p(95)=8.44µs
http_req_connecting.....: avg=9.65µs min=0s med=0s max=1.91ms p(90)=0s p(95)=0s
http_req_duration.....: avg=774.24ms min=515.48ms med=808.24ms max=1.17s p(90)=989.21ms p(95)=1s
  { expected_response:true }....: avg=774.24ms min=515.48ms med=808.24ms max=1.17s p(90)=989.21ms p(95)=1s
http_req_failed.....: 0.00% ✓ 0 x 1129
http_req_receiving.....: avg=840.67µs min=42.81µs med=181.26µs max=34.6ms p(90)=1.41ms p(95)=3.26ms
http_req_sending.....: avg=36.16µs min=10.01µs med=27.29µs max=1.04ms p(90)=43.09µs p(95)=74.53µs
http_req_tls_handshaking.....: avg=0s min=0s med=0s max=0s p(90)=0s p(95)=0s
✓ http_req_waiting.....: avg=773.36ms min=514.81ms med=807.51ms max=1.17s p(90)=987.5ms p(95)=1s
  http_reqs.....: 1129 36.41723/s
vus.....: 29 min=28 max=29
vus_max.....: 30 min=30 max=30
```

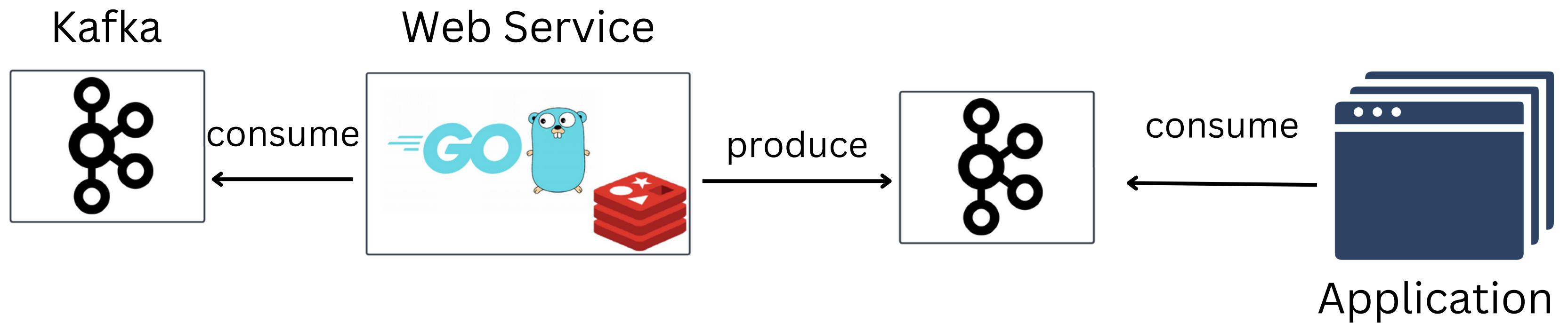
Test Result

Http request duration

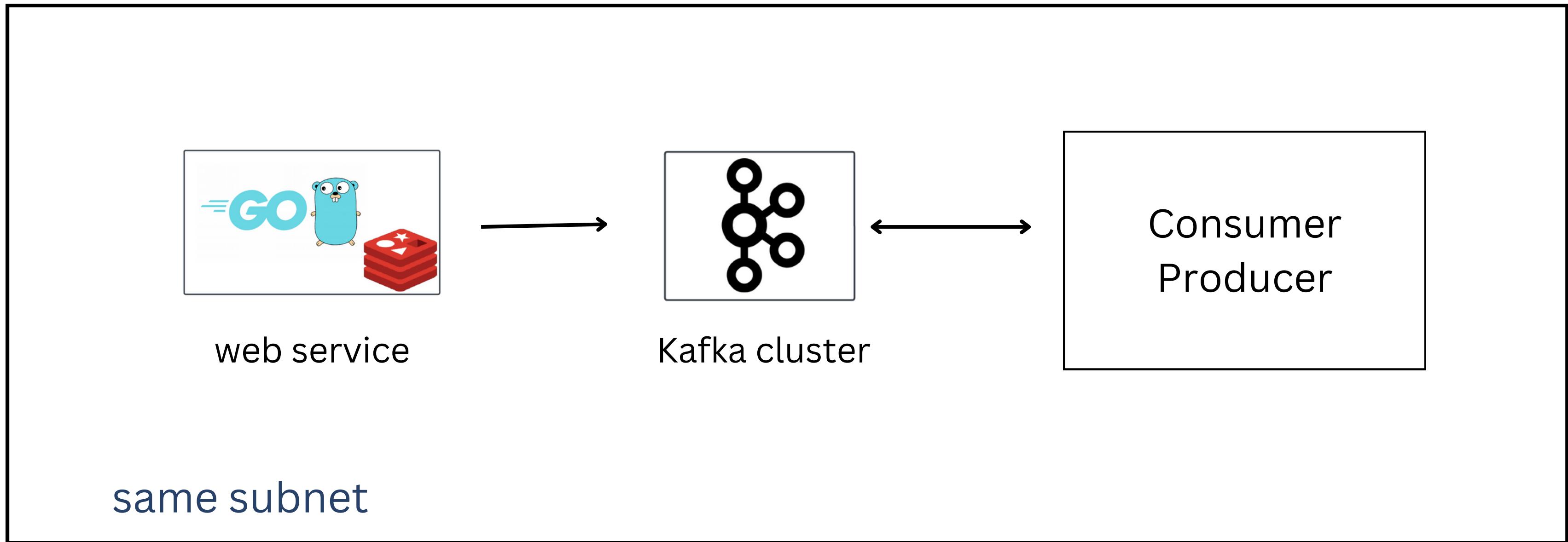


Request real-time data

Data flow



Request real-time data



Test environment

Test scenario

1

Fetch all apps

2

Grant permission for each apps

400 active requests

Running 3 producers in the background

3

Call POST /medical-data endpoint

4

Randomly choose one app as a consumer

5

Measure waiting time to get new message

Active application = 400 applications

- 40 ICU rooms / hospital
- 5 IoT healthcare devices / room
- $40 \times 5 = 200$ IoT healthcare devices in a hospital
- Assume 1 requests per 1 device
- Total requests = 200 requests
- Bound up value (multiply by 2) = $200 \times 2 = 400$ applications



Test Result

ครั้งที่	Average waiting time
1	1.051 s
2	1.048 s
3	1.028 s
4	1.051 s
5	1.048 s

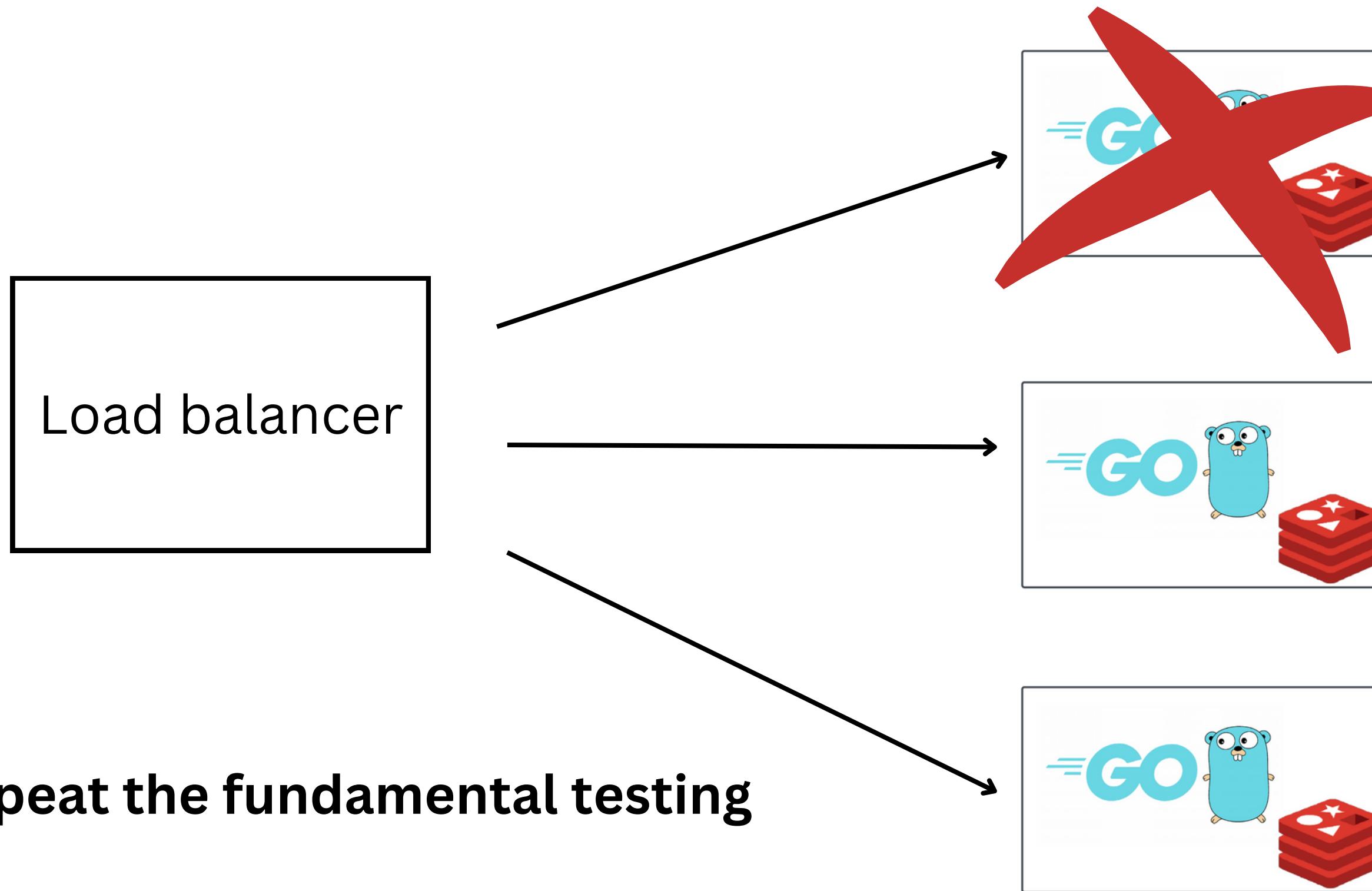
ครั้งที่	Average waiting time
6	1.045 s
7	1.026 s
8	1.048 s
9	1.026 s
10	1.026 s

Average: 1.040 s

High Availability Testing



Test scenario



Test result [Initial scenario]

<input type="checkbox"/>	web-server0	i-030824dc397ac3f44	🕒 Running	⌚⌚	t2.micro	🕒 2/2 checks passed	No alarms	+	us-west-1b
<input type="checkbox"/>	web-server2	i-0268f9c16a63ba03b	🕒 Running	⌚⌚	t2.micro	🕒 2/2 checks passed	No alarms	+	us-west-1b
<input type="checkbox"/>	web-server1	i-01e7c4b7b5d723c89	🕒 Running	⌚⌚	t2.micro	🕒 2/2 checks passed	No alarms	+	us-west-1b

```
PS D:\university\4th_year\capstone\selenium> python .\my-app.py
● get all of my applications: Pass
  required fields(appname) are empty: Pass
  app name is repetitive: Pass
  all required fields are complete: Pass
  ===Edit application===
  edit app name to be the name that isn't already used: Pass
  edit description: Pass
  ===client secret===
  renew client secret: Pass
  ===Delete app ===
  delete an app: Pass
  ===View subscription by app===
  list all subscription of this app: Pass
```

```
PS D:\university\4th_year\capstone\selenium> python .\my-sub.py
get all of my subscriptions: Pass
  ===New subscription===
● required fields(appname) are empty: Pass
  all required fields are complete: Pass
  PS D:\university\4th_year\capstone\selenium> python .\logout.py
  logout: Pass
● PS D:\university\4th_year\capstone\selenium> python .\my-account.py
● get account info: Pass
  ===Change password===
  required fields(appname) are empty: Pass
  new password and re-new password aren't match: Pass
  wrong password: Pass
  correct password: Pass
○ PS D:\university\4th_year\capstone\selenium> █
```

```
● PS D:\university\4th_year\capstone\selenium> python .\login.py
  D:\university\4th_year\capstone\selenium\login.py:17: DeprecationWarning: use options instead of chrome_options
    driver = webdriver.Chrome(service=s,chrome_options=options)
  email or password (required) is empty: Pass
  email format is wrong and password is not empty: Pass
  email format is correct but not exists: Pass
  email is correct and password is not correct: Pass
  email format is correct and password is correct: Pass
○ PS D:\university\4th_year\capstone\selenium> █
```

Test result [After stop 1 instance]

<input type="checkbox"/>	web-server0	i-030824dc397ac3f44	Running	QQ	t2.micro	2/2 checks passed	No alarms	+	us-west-1b
<input type="checkbox"/>	web-server2	i-0268f9c16a63ba03b	Stopped	QQ	t2.micro	-	No alarms	+	us-west-1b
<input type="checkbox"/>	web-server1	i-01e7c4b7b5d723c89	Running	QQ	t2.micro	2/2 checks passed	No alarms	+	us-west-1b

```
≡ login.txt
1 email or password (required) is empty: Pass
2 email format is wrong and password is not empty: Pass
3 email format is correct but not exists: Pass
4 email is correct and password is not correct: Pass
5 email format is correct and password is correct: Pass
6
```

```
≡ my-account.txt
1 get account info: Pass
2 ===Change password===
3 required fields(appname) are empty: Pass
4 new password and re-new password aren't the same: Pass
5 wrong password: Pass
6 correct password: Pass
7
```

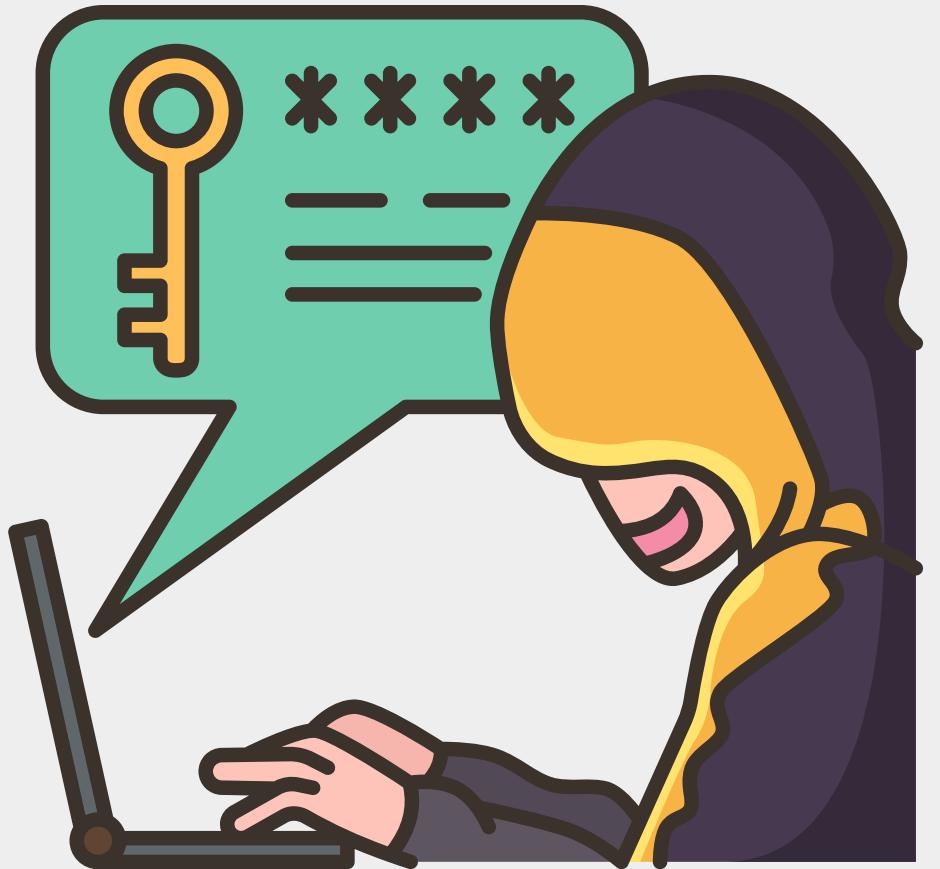
```
≡ unlogin.txt
1 not login (my application): Pass
2 not login (my subscription): Pass
3 not login (my account): Pass
4
```

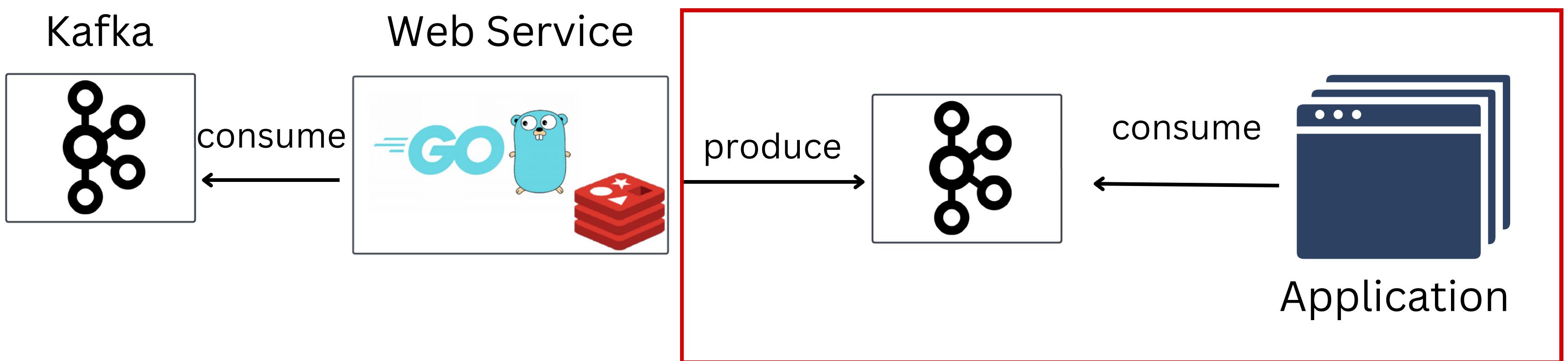
```
≡ my-app.txt
1 get all of my applications: Pass
2 ===New application===
3 required fields(appname) are empty: Pass
4 app name is repetitive: Pass
5 all required fields are complete: Pass
6 ===Edit application===
7 edit app name to be the name that is already used: Pass
8 edit app name to be the name that isn't already used: Pass
9 edit description: Pass
10 ===client secret===
11 renew client secret: Pass
12 ===Delete app ===
13 delete an app: Pass
```

```
≡ logout.txt
1 logout: Pass
2
```

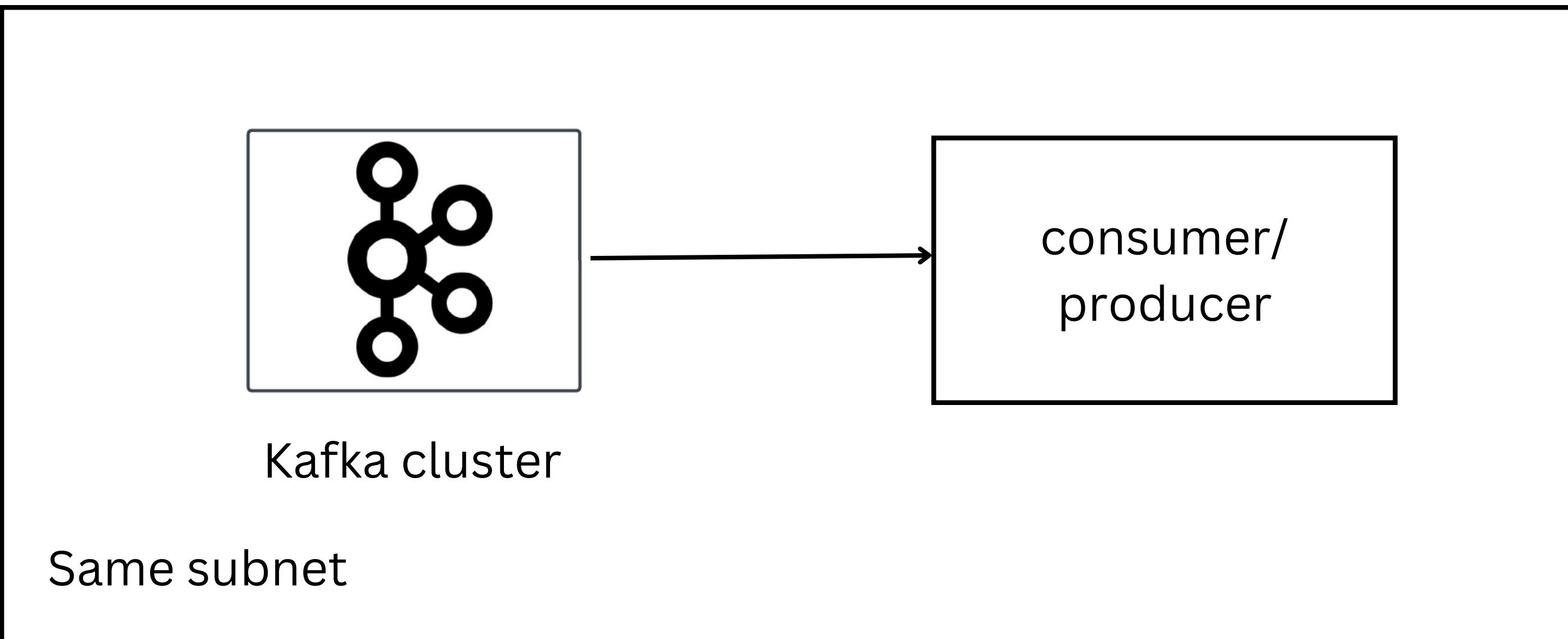
```
≡ my-sub.txt
1 get all of my subscriptions: Pass
2 ===New subscription===
3 required fields(appname) are empty: Pass
4 all required fields are complete: Pass
5
```

Kafka cluster security testing

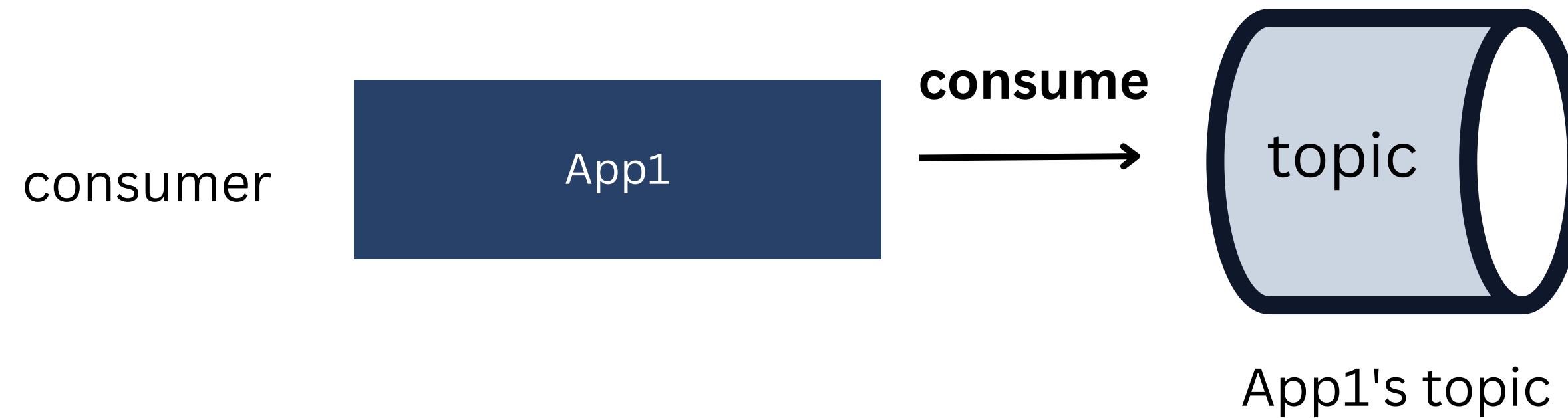




Test environment

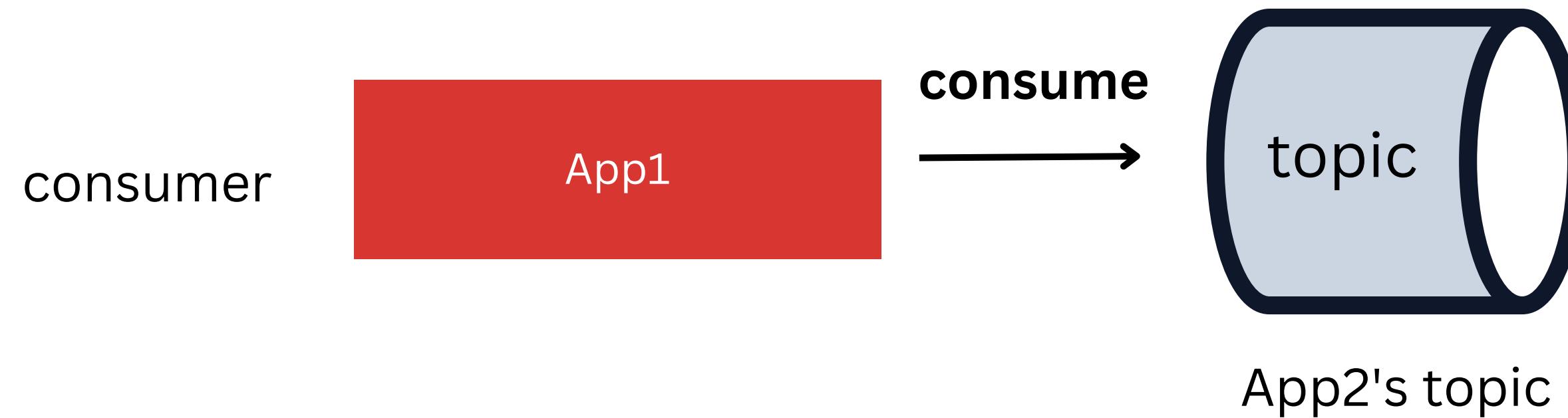


Test scenario [Normal Case]



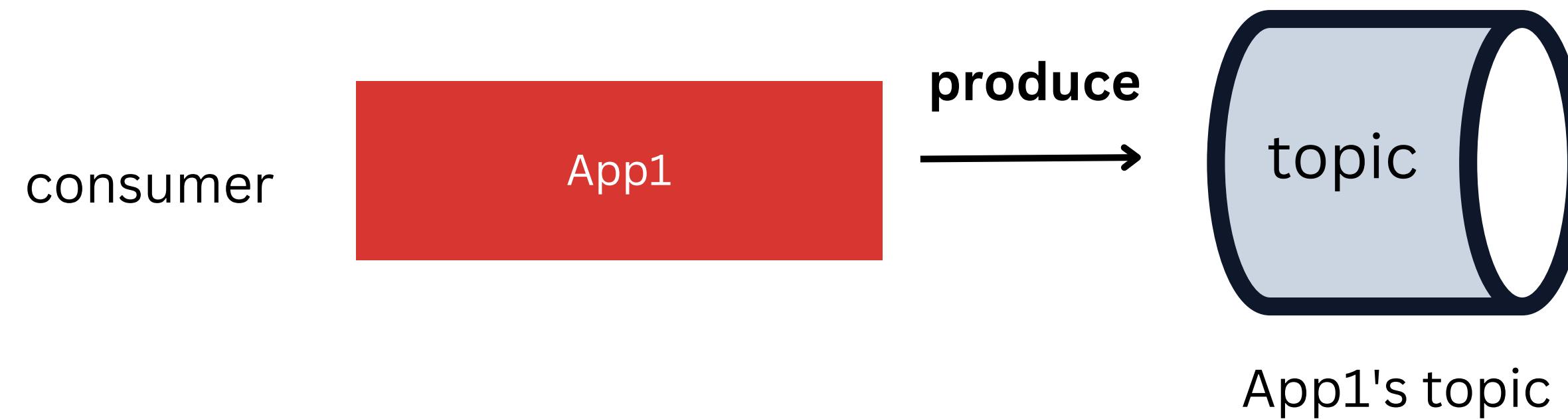
App1 try to consume data from App1's topic

Test scenario [Corner case]



App1 try to consume data from App2's topic

Test scenario [Corner case]



App1 try to data data to App1's topic

Test scenario [Corner case]



App1 try to data data to App2's topic

Test result

```
● ubuntu@ip-172-31-4-216:~/go-consumer$ ./test-script.sh
test1 consumer with permission
pass
test2 consumer without permission
pass
test3 producer produces data to its topic
fail
test4 producer produces data to other topics
pass
○ ubuntu@ip-172-31-4-216:~/go-consumer$ 
```

Demo



summary



Task	Responsibility	2022					2023			
		AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR
1 Project Topic										
2 Project Structure										
2.1 AWS IoT Core	ผู้ดูแล									
2.2 MQTT	ผู้ดูแล									
2.3 Kafka	ผู้ดูแล									
2.4 InfluxDB	ผู้ดูแล									
2.5 Flutter	ผู้ดูแล									
2.6 HealthTag	ผู้ดูแล									
3 Data receiving system										
3.1 learn confluent	ผู้รับผิดชอบ									
3.2 confluent	ผู้รับผิดชอบ									
3.3 learn zookeeper and kraft	ผู้รับผิดชอบ									
3.4 learn confluent kafka bridge	ผู้รับผิดชอบ									
3.5 docker compose file for confluent	ผู้รับผิดชอบ									
3.6 EMQX bridge kafka using EMQX Enterprise	ผู้รับผิดชอบ									
3.7 docker compose file for EMQX Enterprise	ผู้รับผิดชอบ									
3.8 learn MQTT Kafka client library	ผู้รับผิดชอบ									
3.9 build an MQTT Bridge to Kafka(hard code)	ผู้รับผิดชอบ									
3.10 learn pymongo	ผู้รับผิดชอบ									
3.11 connect bridge to mongoDB	ผู้รับผิดชอบ									
3.12 thread and coroutine concept	ผู้รับผิดชอบ									
3.13 trigger function when notice new modelName	ผู้รับผิดชอบ									
3.14 Build an MQTT Bridge to Kafka to get all data from DB	ผู้รับผิดชอบ									
3.15 MQTT Bridge authen to Kafka	ผู้รับผิดชอบ									
3.16 config EMQX cluster to Kafka	ผู้รับผิดชอบ									
3.17 Kafka - configure SASL_SSL	ผู้รับผิดชอบ									
4 Data management system										
4.1 send data to influxDB and mongoDB	ผู้รับผิดชอบ									
4.2 send data to datalake and data warehouse	ผู้รับผิดชอบ									
4.3 time series database clustering	ผู้รับผิดชอบ									
4.4 upload data from mongodb to cloud storage	ผู้รับผิดชอบ									
4.5 receiver - implement receiver consumer group with new library	ผู้รับผิดชอบ									
5 Developer service system										
5.1 Dashboard - Login	ผู้รับผิดชอบ									
5.2 Dashboard - My Application (Get)	ผู้รับผิดชอบ									
5.3 Dashboard - My Application (New)	ผู้รับผิดชอบ									
5.4 Dashboard - My Application (Update)	ผู้รับผิดชอบ									
5.5 Dashboard - My Application (RenewSecret)	ผู้รับผิดชอบ									
5.6 Dashboard - My Application (Delete)	ผู้รับผิดชอบ									
5.7 Dashboard - My Subscription (Get&Filter)	ผู้รับผิดชอบ									
5.8 Dashboard - My Subscription (New)	ผู้รับผิดชอบ									
5.9 Dashboard - My Account (Get)	ผู้รับผิดชอบ									
5.10 Dashboard - My Account (Update)	ผู้รับผิดชอบ									
5.11 Developer - Authenticate developers (Login)	ผู้รับผิดชอบ									
5.12 Developer - Create developer	ผู้รับผิดชอบ									
5.13 Developer - Update developer	ผู้รับผิดชอบ									
5.14 Developer - Get all developer	ผู้รับผิดชอบ									
5.15 Application - Create	ผู้รับผิดชอบ									
5.16 Application - Get app by developer id	ผู้รับผิดชอบ									
5.17 Application - Update app	ผู้รับผิดชอบ									
5.18 Application - Update application secret	ผู้รับผิดชอบ									
5.19 Subscription - Create subscription	ผู้รับผิดชอบ									
5.20 Subscription - Get all subscription according to application and developer	ผู้รับผิดชอบ									
5.21 Application - Delete app	ผู้รับผิดชอบ									
5.22 Application api - Authenticate application api	ผู้รับผิดชอบ									
5.23 Application api - Request historical data (code improvement)	ผู้รับผิดชอบ									
5.24 Application api - Request real time data	ผู้รับผิดชอบ									
5.25 Application api - [real-time data] sender consumer group [implement with new library]	ผู้รับผิดชอบ									

	Task	Responsibility	2022					2023				
			AUG	SEP	OCT	NOV	DEC	JAN	FEB	MAR	APR	
6	Admin service system											
6.1	Dashboard - Get Developer's Subscription	Developer										
6.2	Dashboard - Update Developer's Subscription	Developer										
6.3	Dashboard - Delete Developer's Subscription	Developer										
6.4	Dashboard - Get Developer's Account	Developer										
6.5	Dashboard - Create Developer's Account	Developer										
6.6	Dashboard - Delete Developer's Account	Developer										
6.7	Dashboard - Get Device	Developer										
6.8	Dashboard - Create Device	Developer										
6.9	Dashboard - Delete Device	Developer										
6.10	Admin - Authentication (Login)	Developer										
6.11	Developer - Create developers	Developer										
6.12	Developer - Get all developers in the system	Developer										
6.13	Developer - Delete developers	Developer										
6.14	Medical model - Get all model in the systems (code improvement)	Developer										
6.15	Medical model - Get all model and device id list	Developer										
6.16	Medical model - Create medical model (code improvement)	Developer										
6.17	Medical model - Update medical model	Developer										
6.18	Medical model - Delete medical model	Developer										
6.19	Subscription - Update subscription status	Developer										
7	Log system											
7.1	Logging gcs upload status	Developer										
7.2	Logging all request	Developer										
7.3	Uptime monitoring	Developer										
7.4	CPU/Mem monitoring	Developer										
7.5	Kibana	Developer										
7.6	Improve log when file was uploaded on google cloud storage	Developer										
8	Deployment											
8.1	Learned how to write nginx config file	Developer										
8.2	Write nginx config file and create docker image for application	Developer										
8.3	Write docker compose file for web service cluster	Developer										
8.4	Deploy on AWS (1 instance 1 for nginx server and others for web service)	Developer										
8.5	Create docker file for web service (need to have java runtime)	Developer										
8.6	Configure Kafka with SASL_SSL with 3 zookeeper and 3 broker using 3 instance	Developer										
9	Testing											
9.1	Unit data receiving test case	Developer										
9.2	Unit admin dashboard test case	Developer										
9.3	Unit developer dashboard test case	Developer										
9.4	Fundamental testing - Data receiving system	Developer										
9.5	Fundamental testing - Developer service system (not include application api)	Developer										
9.6	Fundamental testing - Application api	Developer										
9.7	Fundamental testing - Admin service system	Developer										
9.8	Web service high availability testing	Developer										

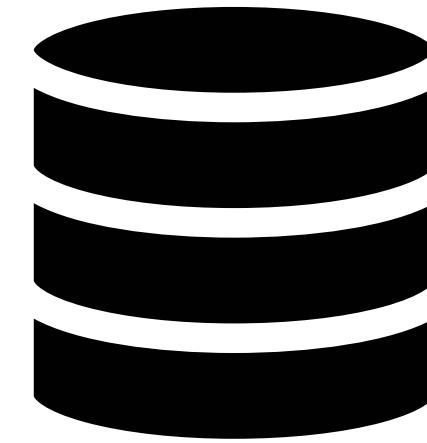
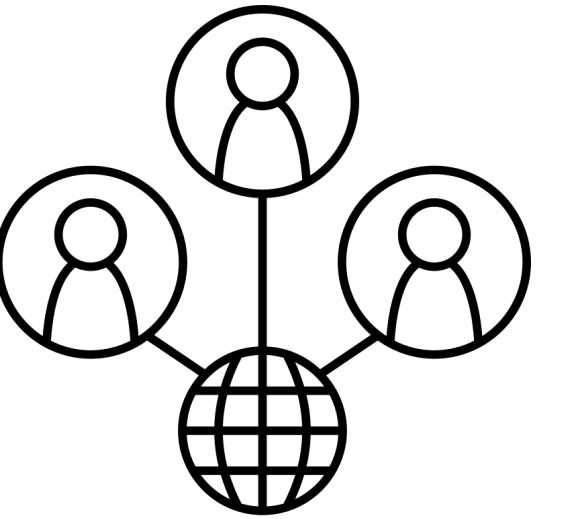
Discussion



Hospital network

- Patient Monitor (ALLSTAR-01)
 - Backup
 - 7-30 days
- On premise cloud

Discussion



Data Request

- from hospital network: private IP
- from outside network: public IP

Database: Mongodb

The End

**Thank you
for listening**

