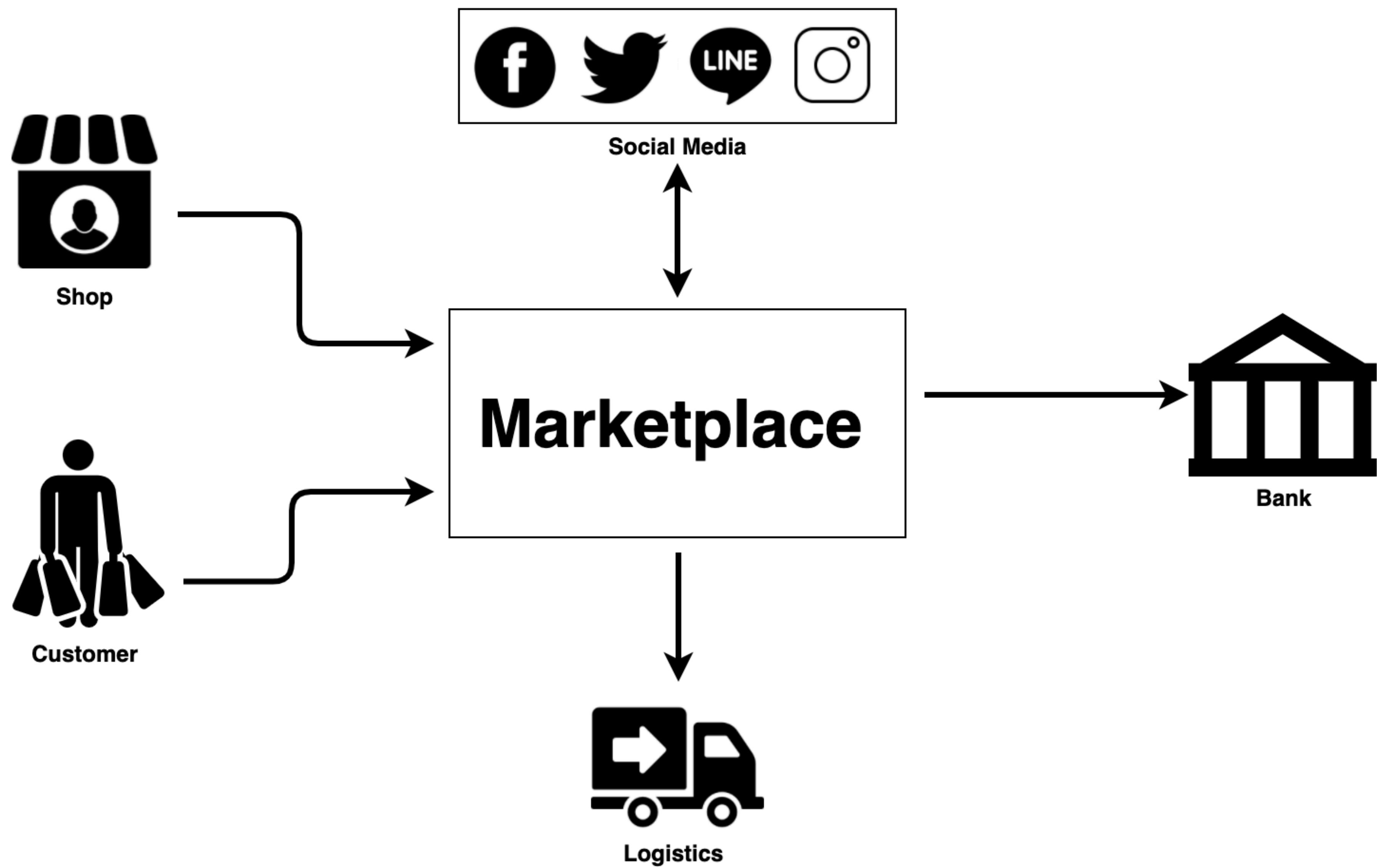


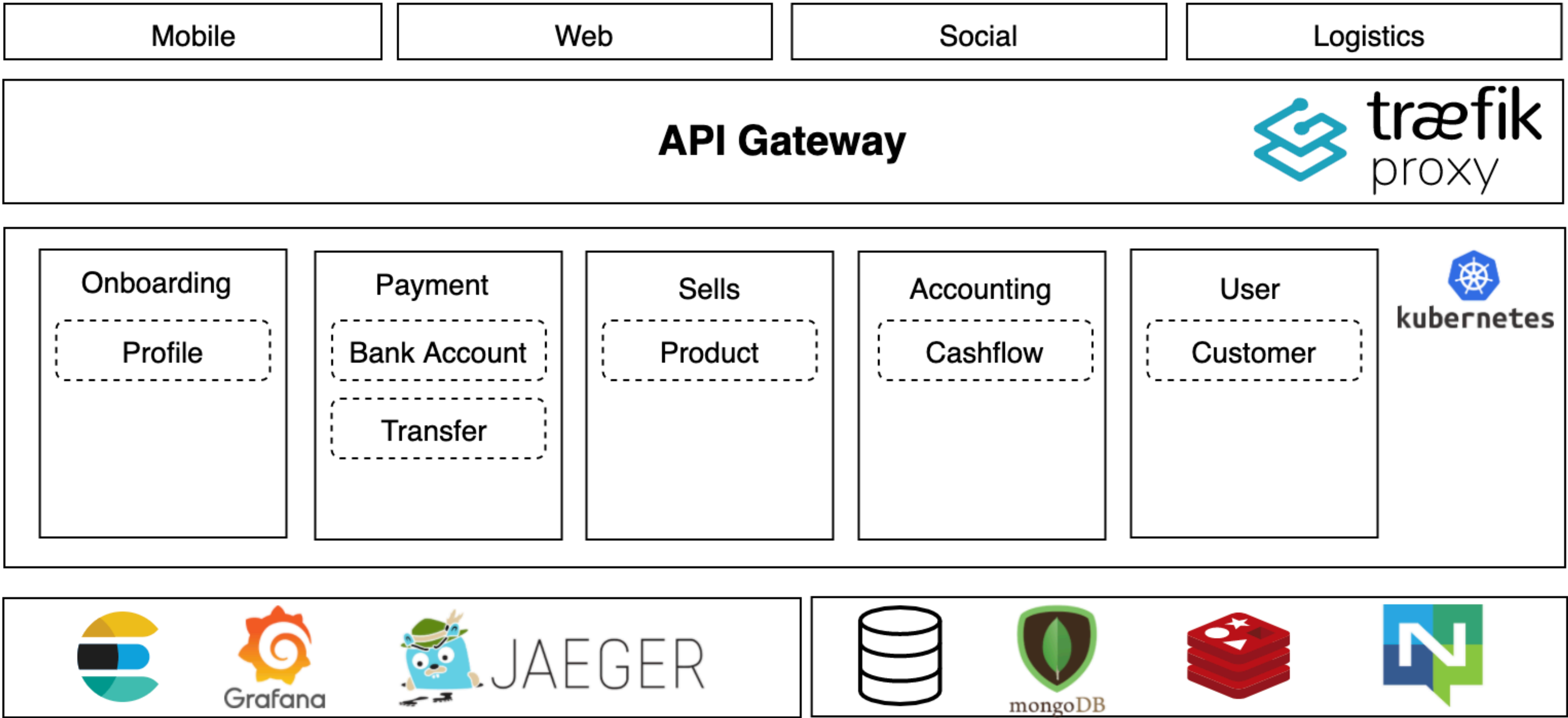
Social Marketplace

By Pradist Keawborisut

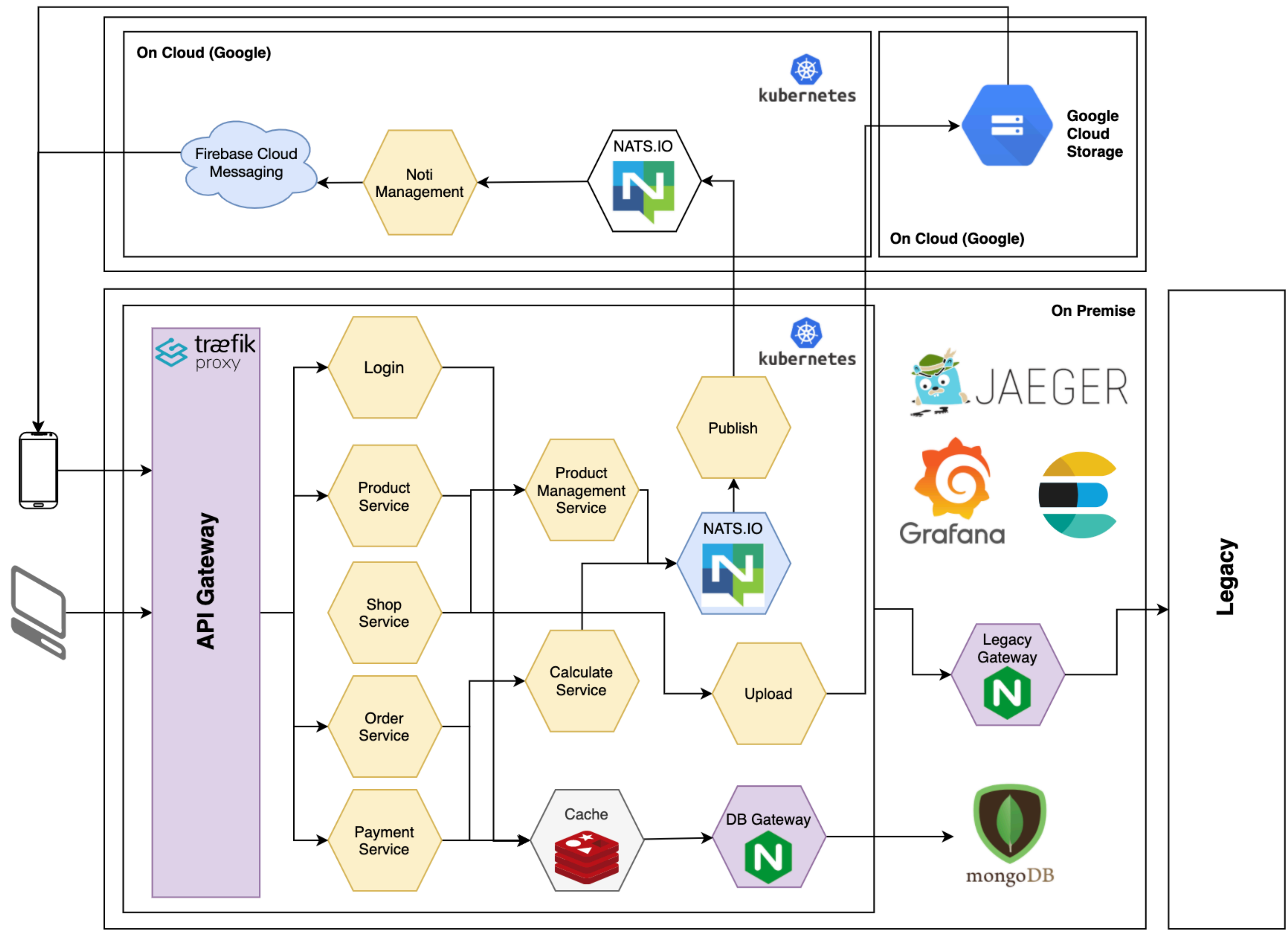
System Context Diagram



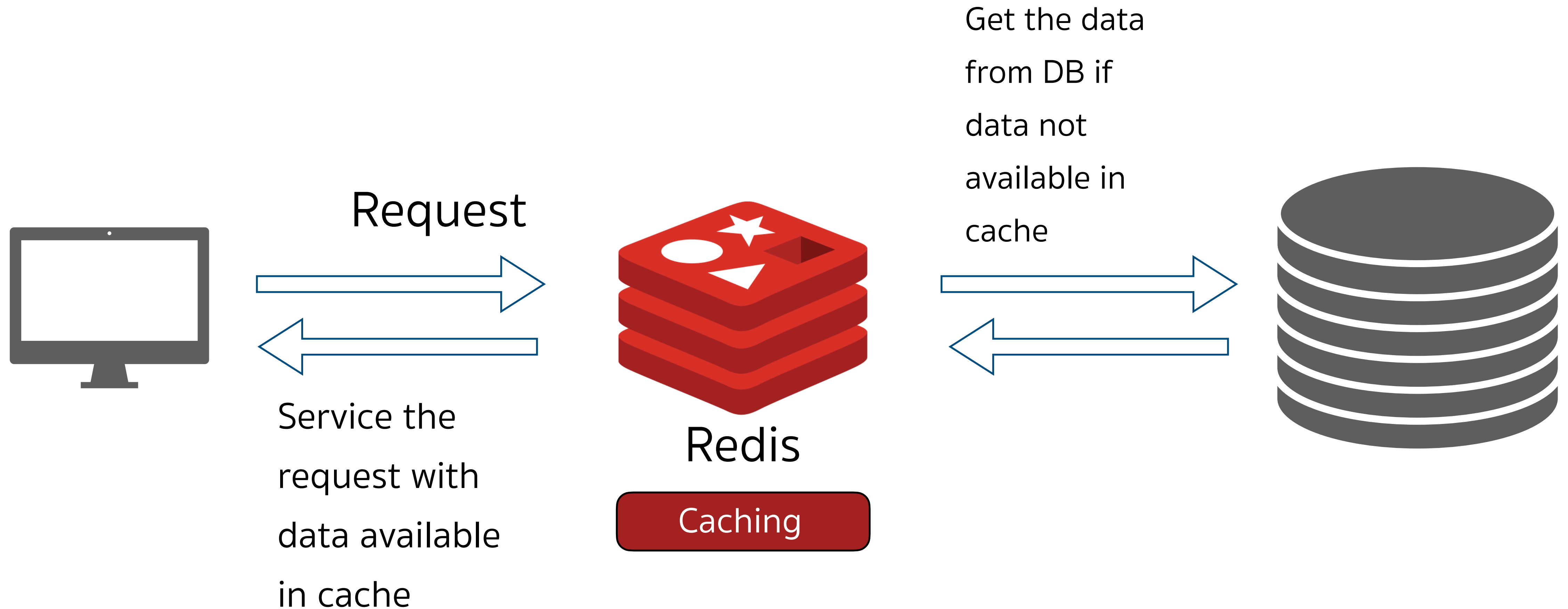
Architecture



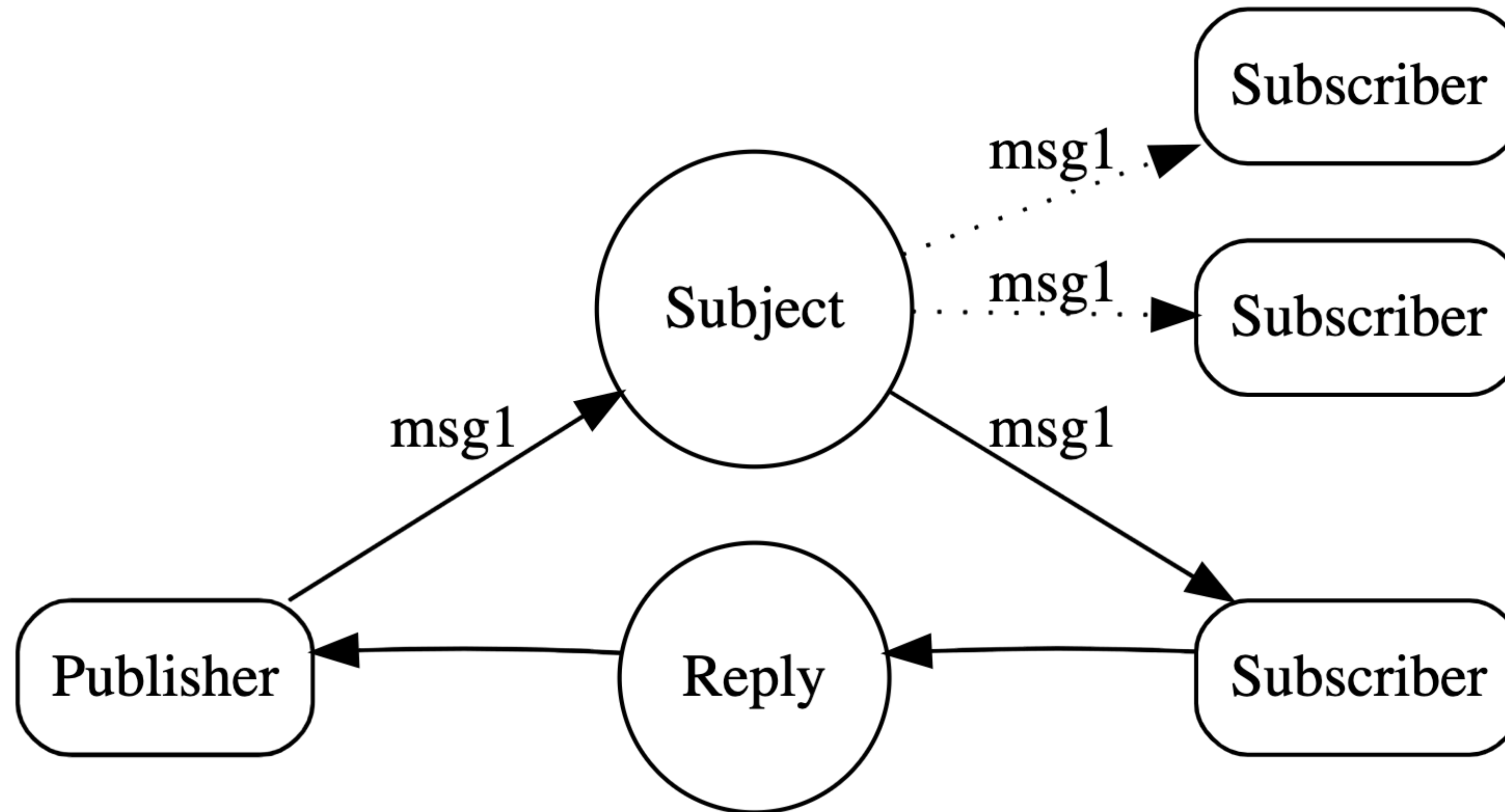
Architecture



Architecture



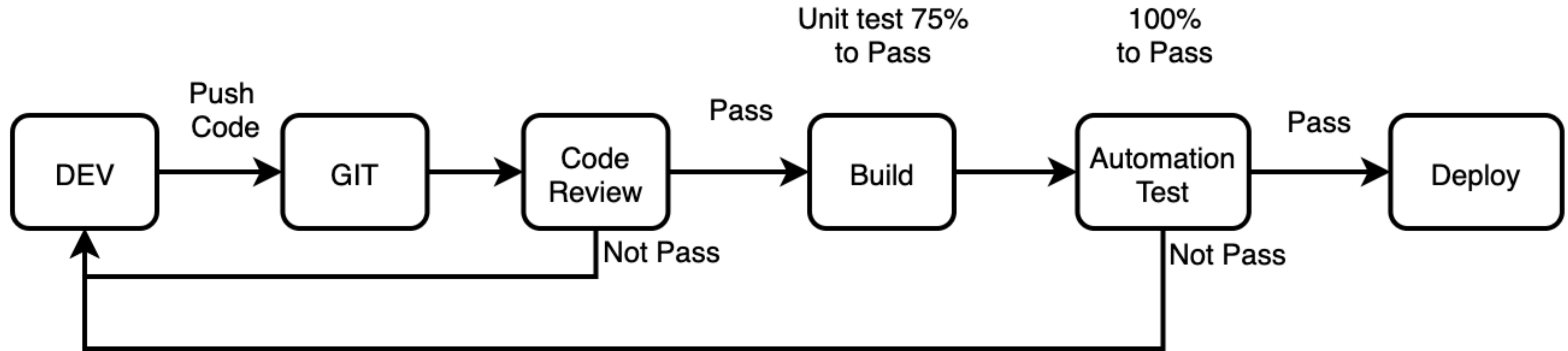
Architecture



Technology Selection

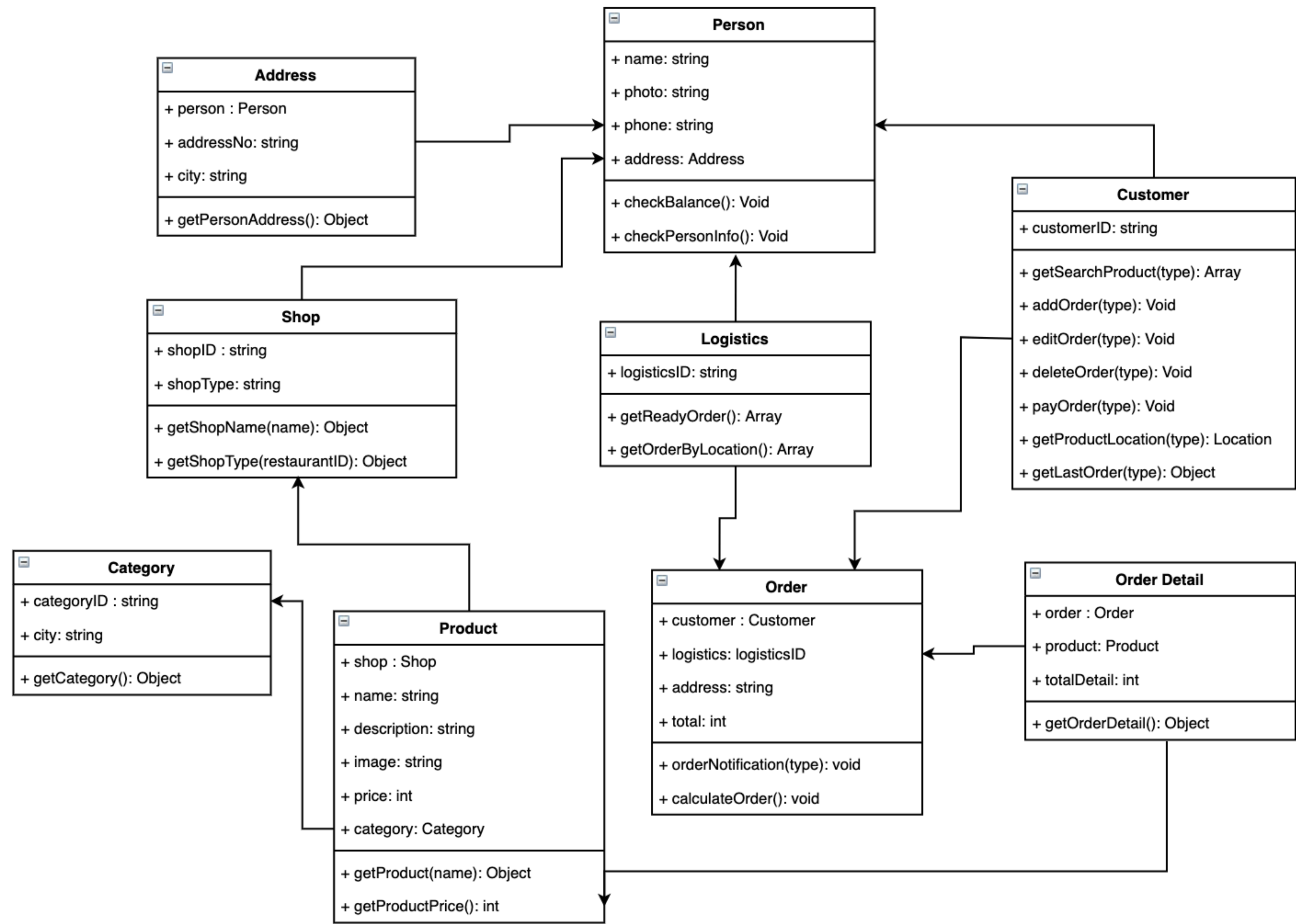
Technology	Description
Frontend	Mobile: Flusters / Web: React
Backend	Base on golang (Can use another language as appropriate)
Server	Kubernetes (Container Base)
Database	MongoDB
Cache	Redis
API Gateway	Traefik Proxy
Outbound Gateway	Nginx
Message Queue	nats.io streaming
Monitor Log	Elasticsearch, Gafana, Jeager

Deployment Process



- Jenkins
- Postman (Newman)

ER Diagram



API Specification

🔍 Search...

Authentication

user >

product >

payment >

order >

shop >

Documentation Powered by ReDoc

K Marketplace Product (1.0.0)

Download OpenAPI specification: [Download](#)

This is a K Marketplace Swagger.

Authentication

bearerAuth

Security Scheme Type	API Key
Header parameter name:	Authorization

API Specification

🔍 Search...

Authentication

user

POST

Register user

POST

login

product

payment

order

shop

Documentation Powered by ReDoc

...

REQUEST BODY SCHEMA: application/json

Register user object

username

string

[3 .. 255] characters

email

string <email>

[3 .. 255] characters

mobile

string

10 characters

Responses

> 200

successful operation

— 500

operation fail

Request samples

Payload

Content typeapplication/json

CopyExpand allCollapse all

{
 "username": "string",
 "email": "user@example.com",
 "mobile": "stringstri"
}

Response samples

200

Content typeapplication/json

CopyExpand allCollapse all

{
 "Time": "1999-12-31T23:59:59Z",
 "result": "Ok",
 - "resource": [
 + { ... }
]
}

API Specification

🔍 Search...

Authentication

user >

product ▼

POST

 uploads an image

POST

 Create product

PUT

 Update product

GET

 Get product

DEL

 Delete product

GET

 Search products

payment >

order >

shop >

Documentation Powered by ReDoc

product

🔗 uploads an image

AUTHORIZATIONS: bearerAuth

PATH PARAMETERS

→ product_id
required

integer <int64>
ID of product to update

REQUEST BODY SCHEMA: multipart/form-data

→ file

string <binary>
file to upload

Responses

> 200 successful operation

POST /product/{product_id}/uploadImage ▼

Response samples

200

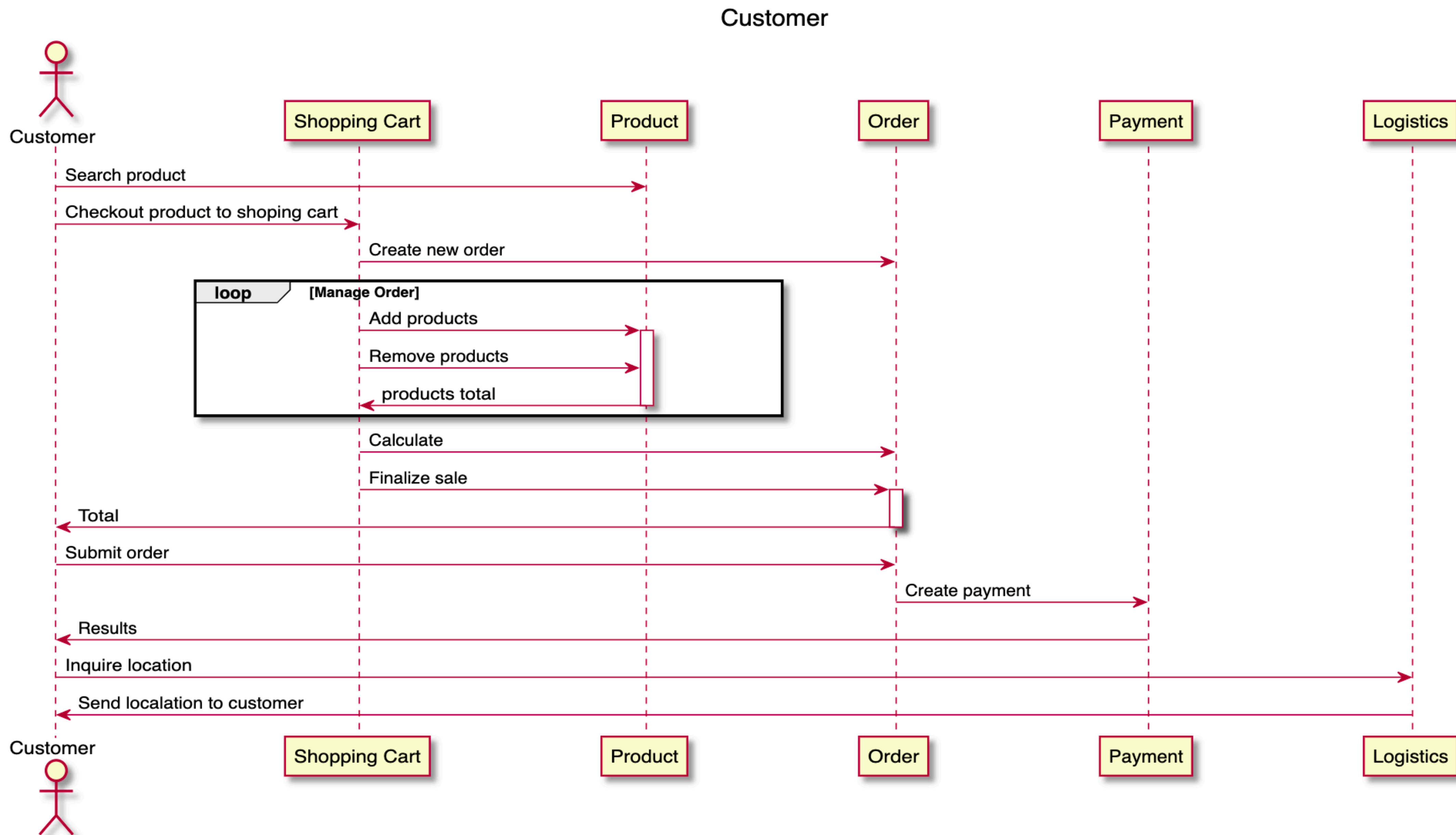
Content type

application/json

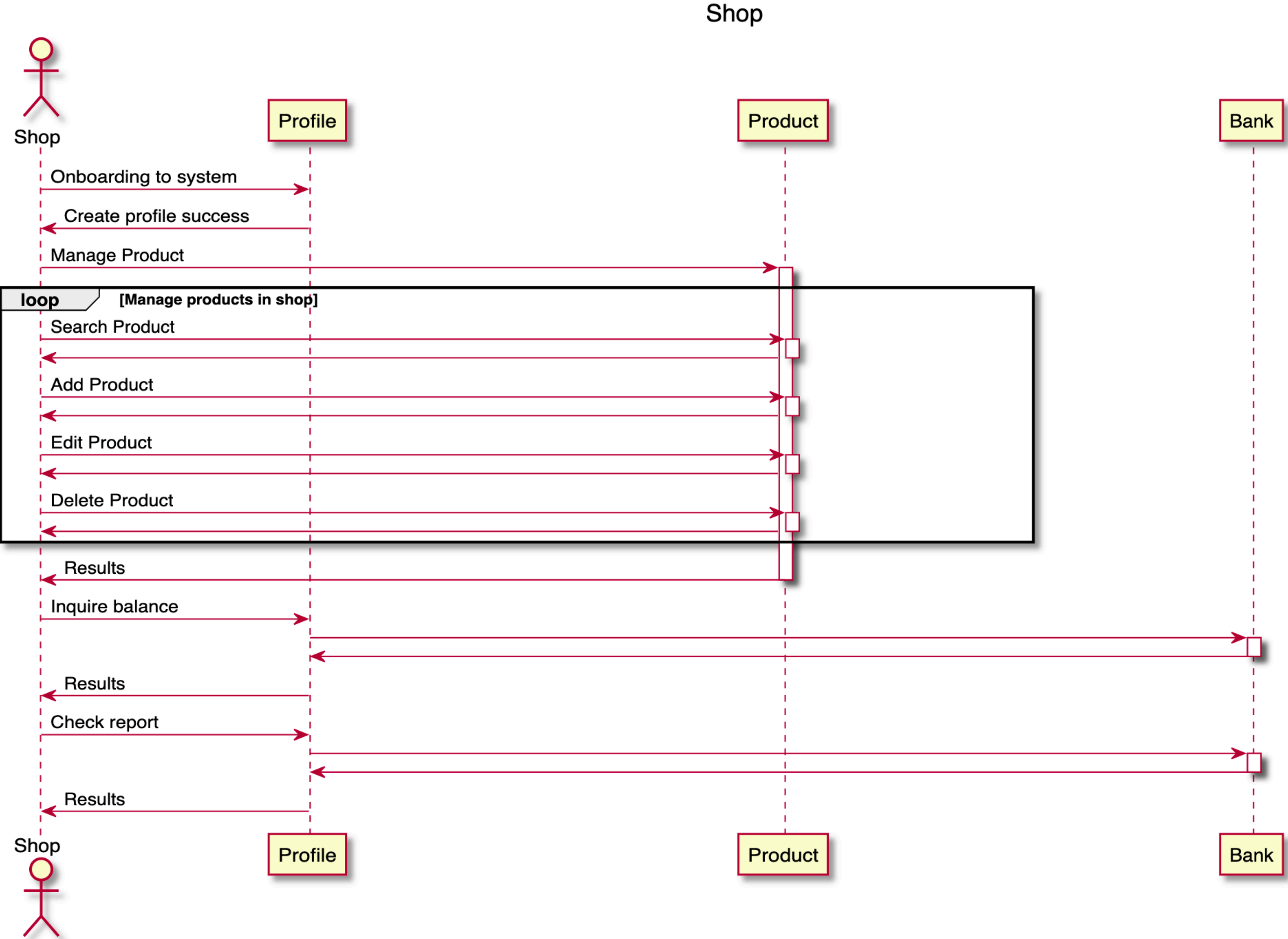
CopyExpand allCollapse all

```
{
  "id": 0,
  "name": "string",
  "description": "string",
  "category_id": 0,
  "shop_id": 0,
  "attribute_set_id": 0,
  "tax_id": 0,
  "price": 0,
  "currency": "str",
  "sku": "string",
  "stock": 0,
```

Sequence Diagram

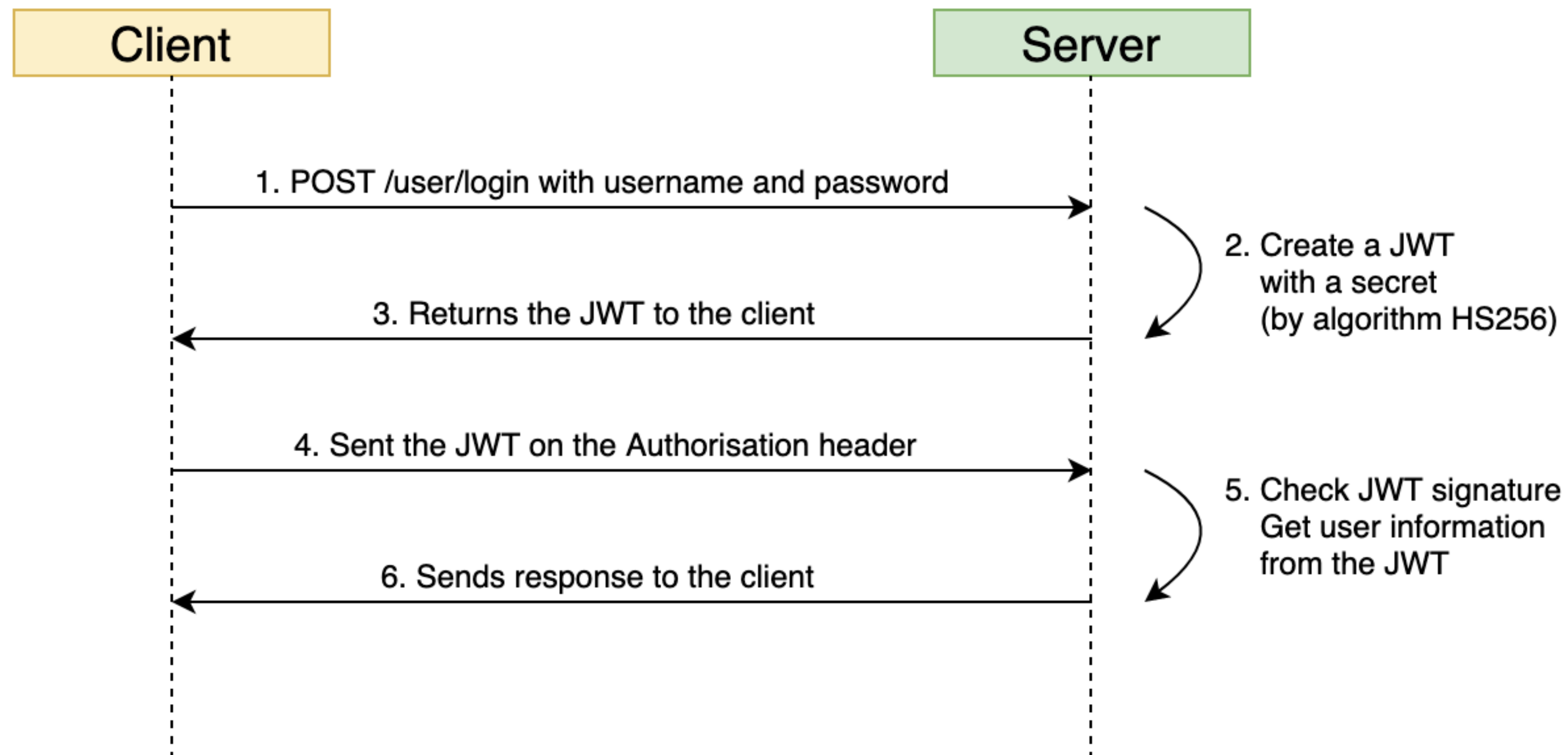


Sequence Diagram



Security Implementation

- Use TLS 1.2 for backend side.
- Use JWT by algorithm HS256
- Use Encryption and Decryption by "golang.org/x/crypto/scrypt"



***Reconcile Data**

Design Choices

Technology	Advantage	Disadvantages
Kubernetes	<ul style="list-style-type: none">- Easy to use- Can use command line to access- Free	<ul style="list-style-type: none">- Hard to setup
Redis	<ul style="list-style-type: none">- Easy to use- Fast- Have meal feature- Easy to scaling	<ul style="list-style-type: none">- Less support
NATS IO	<ul style="list-style-type: none">- Fast- Support height concurrent	<ul style="list-style-type: none">- Data loss

“Thank You.”

– Pradist Keawborisut