사용자 인증 기능을 포함한
채팅 서버의 구현

작성일	2023. 10. 13
작성자	도현승
연락처	010-9661-3680 mabytion@live.com

1. 목표

   A. 인가된 사용자에 한하여 서비스 제공
   B. 토큰 기반 인증을 활용한 사용자 인증
   C. 분산 서버 구조를 이용한 고가용성 시스템 구현

2. 구성

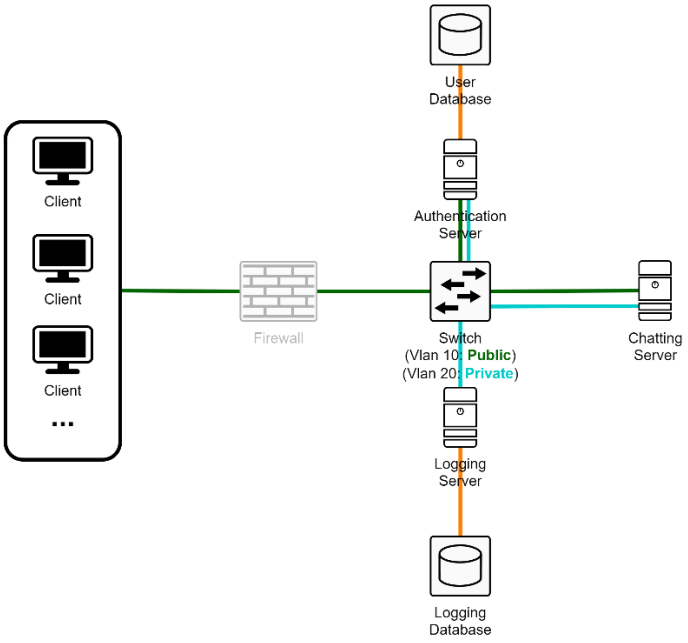
   A. 시스템 구성

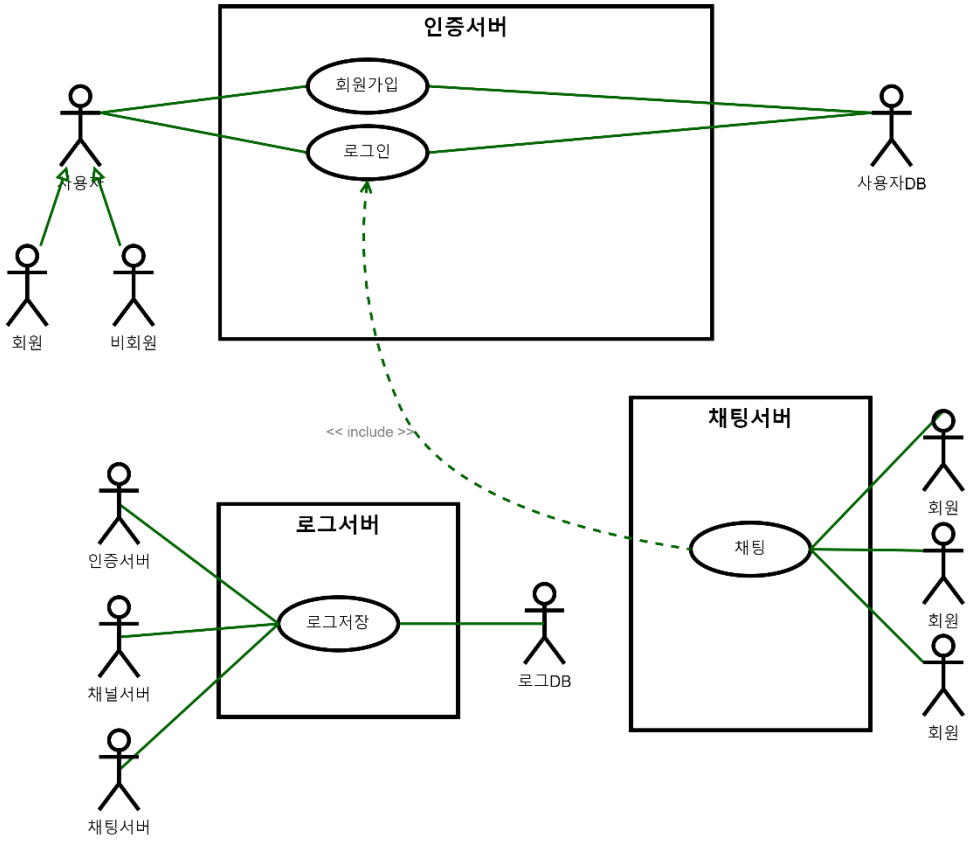
Public	
Log	192.168.75.11
Chat	192.168.75.12
Auth	192.168.75.14

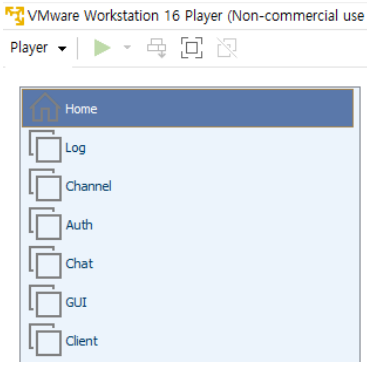
Private	
Log	192.168.123.11
Chat	192.168.123.12
Auth	192.168.123.14

OS	
ALL	CentOS 7.9

Database	
User	MariaDB 10.4.29
Log	MongoDB 5.0.17







➢ 구현물에선 가상머신을 이용함

B. JWT

Access Token

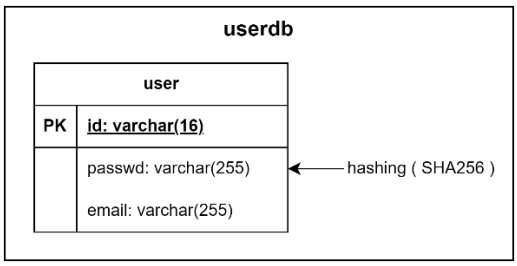
Header
{
"alg": "HS256",
"typ": "JWT"
}

Payload
{
"iss": "Auth",
"user": $User_ID,
"token": "access"
"exp": iat + 30min,
"iat": now()
}

Verify
{
HMACSHA256(
base64encode(Header) +
"," +
base64encode(Payload),
key)
}

C. 데이터베이스

i. User Database



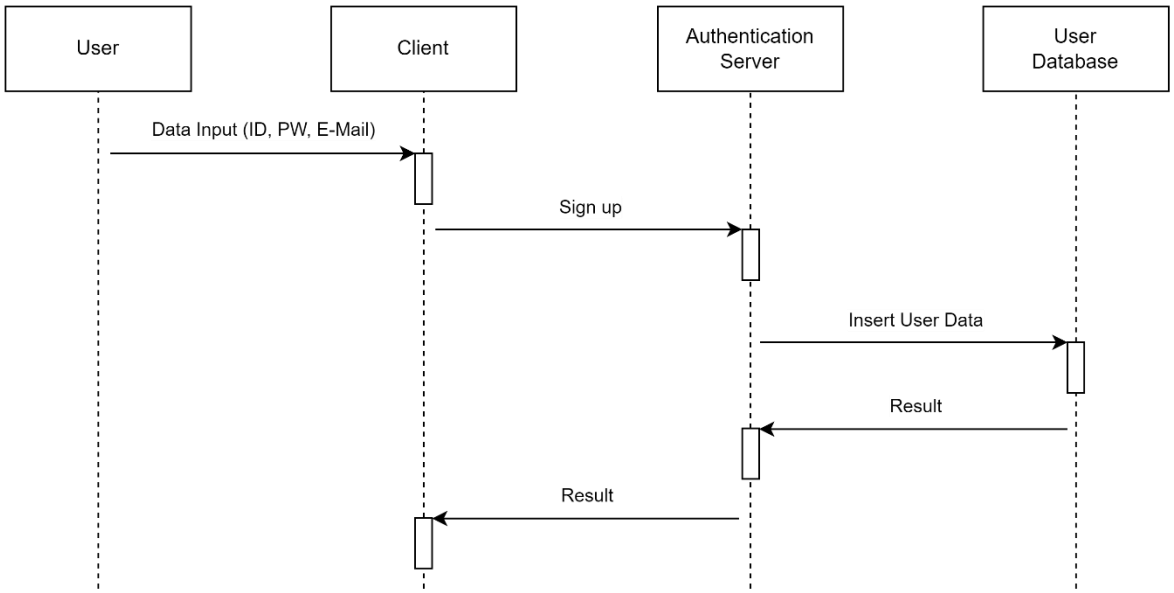
ii. Logging Database

Logdata
+ datetime
+ server
+ category
+ id
+ data

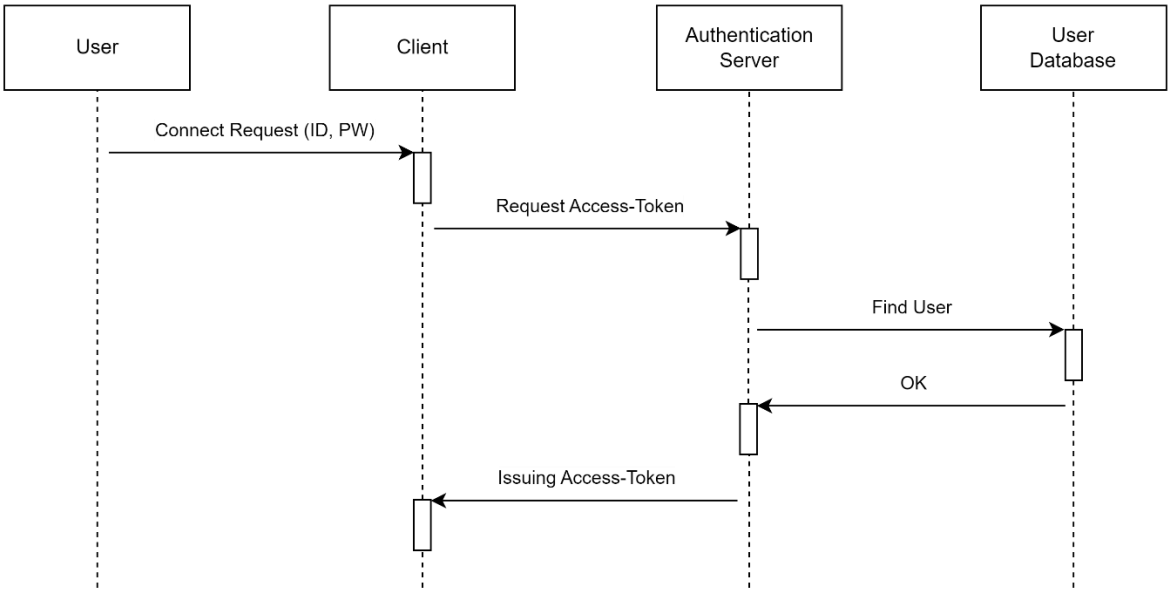
3. 기능

A. 인증서버

i. 사용자 생성



ii. 사용자 인증(JWT 토큰 발급)



iii. API 명세

Name	HTTP Method	Content Type	URL	Request	Response

Login	POST	JSON	http://192.168.75.14:8081/auth/login	{
				"id":"(id)",
				"passwd":"(passwd)"
				}
					{
					"token":
					"(access_token)"
					}

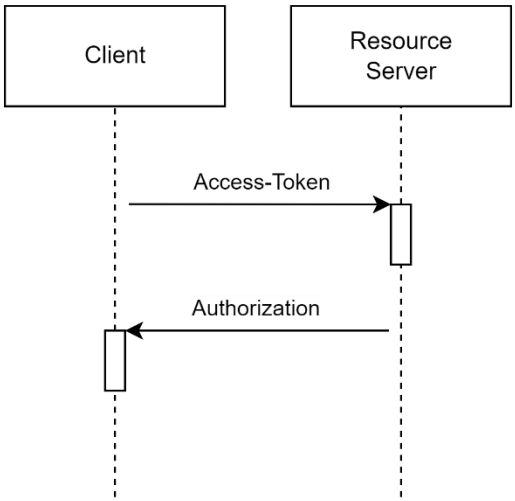
Signup	POST	JSON	http://192.168.75.14:8081/auth/signup	{
				"id":"(id)",
				"passwd":"(passwd)",
				"email":"(email)"
				}
					{
					"message": "signup successful"
					}

Getuser	GET	JSON	http://192.168.75.14:8081/auth/getuser/:id		{
					"id":"(id)"
					}

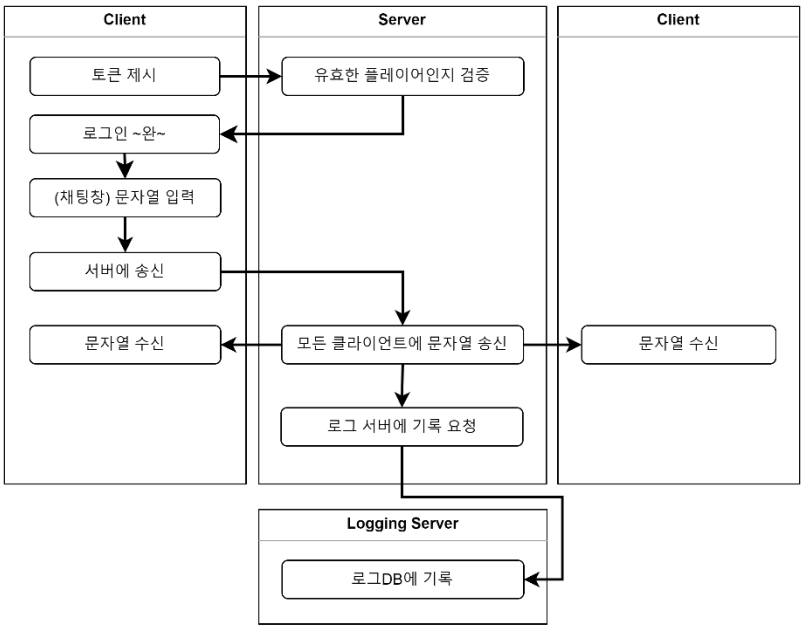
Health	GET	JSON	http://192.168.75.14:8081/auth/health		{
					"message": "alive"
					}

B. 채팅서버

i. 인가



ii. 채팅 기능 제공




```
[root@Client client]# ./client
ID: test1
PW: 123123
ok

connected.

dk
test1 : dk
test3 : 123123
test2 : 123123
123123123123
test1 : 123123123123
```

```
[root@Client client]# ./client
ID: test2
PW: 123123
ok

connected.

test3 : 123123
123123
test2 : 123123
test1 : 123123123123
```

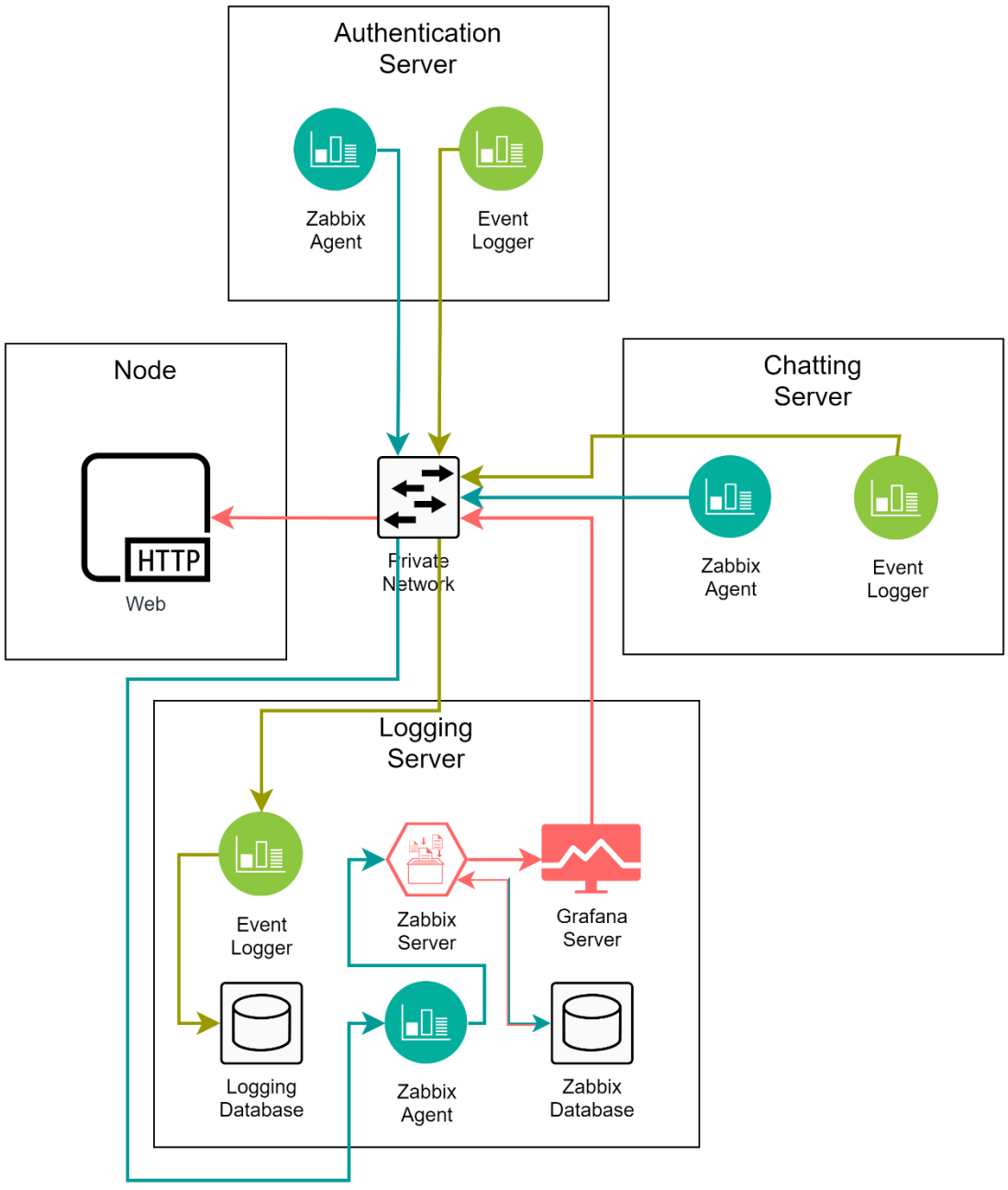
```
[root@Client client]# ./client
ID: test3
PW: 123123
ok

connected.

123123
test3 : 123123
test2 : 123123
test1 : 123123123123
```

C. 로그서버

i. 시스템 구조



ii. 시스템 자원 모니터링

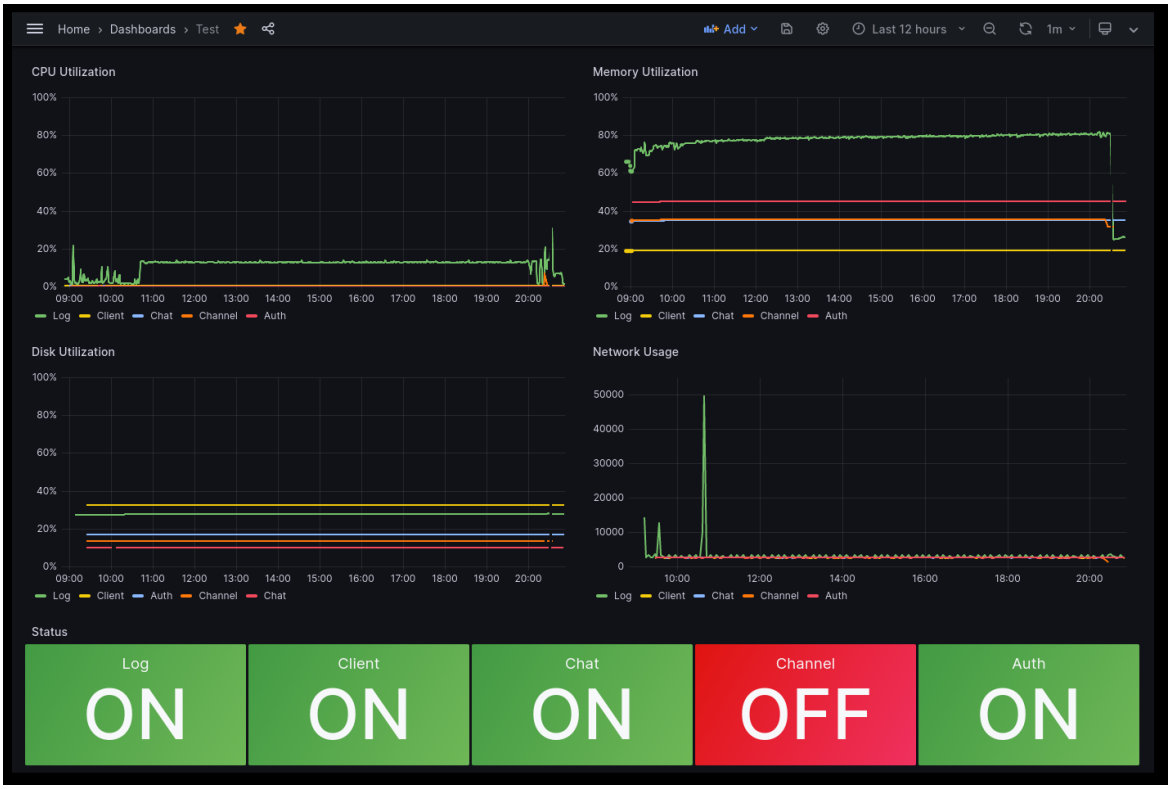
❖ 사용한 솔루션

1) Zabbix (시스템 자원 현황 수집 도구)

2) Grafana (데이터 시각화 도구)

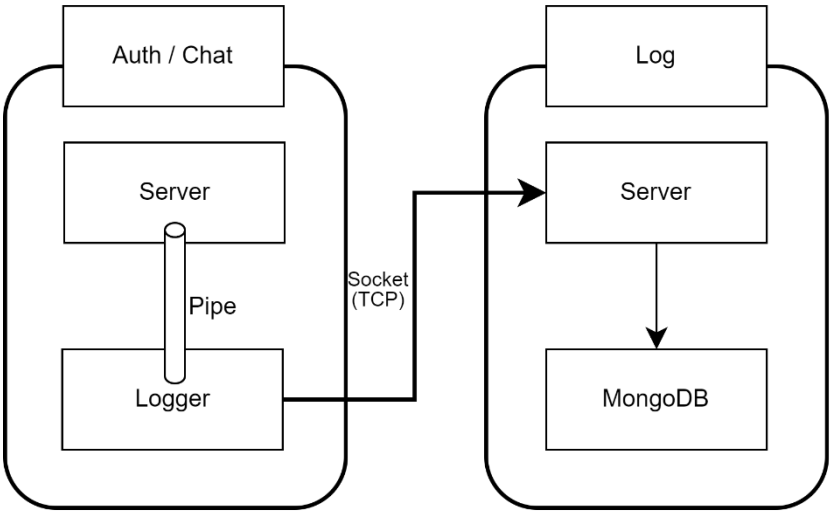
3) 시각화 항목

1) CPU 사용량	2) 메모리 사용량
3) 디스크 사용량	4) 네트워크 사용량
5) Agent 상태	



iii. 이벤트 로그 수집

❖ 이벤트 로그 수집 기능



① 서버 프로세스 복제 (fork)

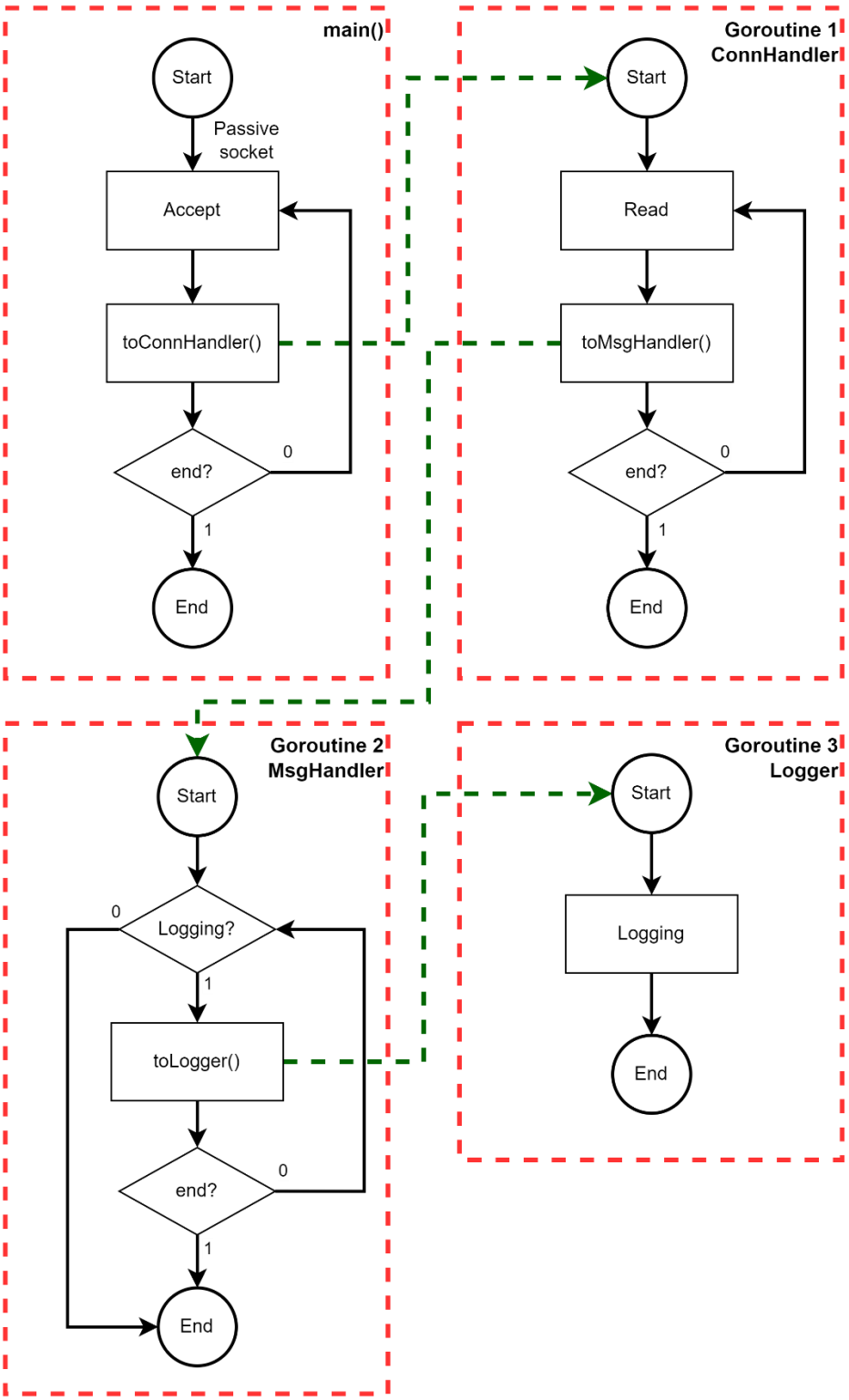
② 복제된 프로세스를 Logger 바이너리로 교체 (exec)

③ 서버에서 특정 이벤트 발생 시 일정한 형식의 바이트 스트림을 생성하고 Pipe를 통해 Logger에 전달

④ Logger는 받은 바이트 스트림을 Log Server에 릴레이

⑤ Log Server는 전달받은 바이트 스트림을 파싱 하여 데이터베이스에 저장

❖ Log Server 내부 기능



❖ 데이터 입력

❖ db.Logdata.insertOne({ datetime:"(datetime)", server: "(server_name)", category: "(category)", data: "(dataform)" })

❖ string: 0/(server)/(category)/(id)/(data)

❖ 데이터 형식

1) datetime

❖ YYYY-MM-DDTHH:MM:SSZ

2) format

server	category	id	data
auth	sign{up | in}	{userid}	{Success | Fail}
chat	chat	({nickname})	{chat-contents}

➢ 실제로 저장되는 로그

```
{
  _id: ObjectId("65286232409a523615639592"),
  date_time: '2023-10-13T06:16:20Z',
  server: 'auth',
  category: 'signin',
  id: 'testaccounts0',
  data: 'fail'
},
{
  _id: ObjectId("65286235409a523615639593"),
  date_time: '2023-10-13T06:16:34Z',
  server: 'auth',
  category: 'signin',
  id: 'testaccounts1',
  data: 'success'
},
{
  _id: ObjectId("65286237409a523615639594"),
  date_time: '2023-10-13T06:16:37Z',
  server: 'auth',
  category: 'signin',
  id: 'testaccounts2',
  data: 'success'
},
{
  _id: ObjectId("6528623a409a523615639595"),
  date_time: '2023-10-13T06:16:39Z',
  server: 'auth',
  category: 'signin',
  id: 'testaccounts5',
  data: 'fail'
}
```

```
{
  _id: ObjectId("64f4de9a0c98d8890431e003"),
  date_time: '2023-09-04T04:29:30Z',
  server: 'chat',
  category: 'chat',
  id: 'testaccounts2',
  data: '2'
},
{
  _id: ObjectId("64f4de9a0c98d8890431e004"),
  date_time: '2023-09-04T04:29:30Z',
  server: 'chat',
  category: 'chat',
  id: 'testaccounts2',
  data: '3'
},
{
  _id: ObjectId("64f4de9a0c98d8890431e005"),
  date_time: '2023-09-04T04:29:30Z',
  server: 'chat',
  category: 'chat',
  id: 'testaccounts2',
  data: '4'
},
{
  _id: ObjectId("64f4de9f0c98d8890431e006"),
  date_time: '2023-09-04T04:29:30Z',
  server: 'chat',
  category: 'chat',
  id: 'testaccounts2',
  data: '안녕하세요'
}
```