OpenApi Gateway

기술기능 분석 설계 자료

|  |  |  |
| --- | --- | --- |
| 작성자 |  |  |
| 검토자 |  |  |
| 승인자 | 정기봉(인) |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

개 정 이 력

| 버전 | 작성일 | 변경내용[[1]](#footnote-1) | 작성자 | 승인자 |
| --- | --- | --- | --- | --- |
| 1.0 | 2017.11.24 | 최초작성 |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

목차

[1. 개요 4](#_Toc500244070)

[1.1. 문서 개요 4](#_Toc500244071)

[1.1.1. 목적 4](#_Toc500244072)

[1.1.2. 기술 개요 4](#_Toc500244073)

[1.1.3. 기술 범위 4](#_Toc500244074)

[2. 분석 5](#_Toc500244075)

[2.1. 요구사항 분석 5](#_Toc500244076)

[2.1.1. 기술 요구사항 5](#_Toc500244077)

[2.1.2. 구현 범위 및 방법 요구사항 5](#_Toc500244078)

[2.1.3. 리소스 요청 및 응답 요구사항 5](#_Toc500244079)

[2.2. 요구사항 기능분석 5](#_Toc500244080)

[2.3. 요구기능 설명 6](#_Toc500244081)

[2.3.1. Api Gateway 6](#_Toc500244082)

[2.3.1.1. Kong (Open Source Api Geteway) 개요 6](#_Toc500244083)

[2.3.1.2. Kong (Open Source Api Geteway) 구성 7](#_Toc500244084)

[2.3.1.3. Kong (Open Source Api Geteway) 특징 7](#_Toc500244085)

[2.3.2. Vert.x 8](#_Toc500244086)

[2.3.2.1. Vert.x 개요 8](#_Toc500244087)

[2.3.2.2. Vert.x 특징 8](#_Toc500244088)

[2.3.2.3. Vert.x 구성 8](#_Toc500244089)

[2.3.2.4. Vert.x Event Loop 및 Event Bus프로세스 9](#_Toc500244090)

[2.3.2.5. Vert.x Multi instance 처리 9](#_Toc500244091)

[2.3.2.6. Vert.x 에서의 Hazelcast 특징 10](#_Toc500244092)

[3. 설계 11](#_Toc500244093)

[3.1. 구성도 11](#_Toc500244094)

[3.2. Vert.x 11](#_Toc500244095)

[3.3. Kong (Open Source API Gateway) 11](#_Toc500244096)

[3.4. Kong 과 inbound Restful api 연계 12](#_Toc500244097)

1. 개요
   1. 문서 개요
      1. 목적

* 본 문서는 PHR 인터페이스 연계 방안의 분석 문서이다.
  + 1. 기술 개요
* 의료기기 사용자 데이터 분석결과 데이터와 인터페이스 제공
* 인터페이스를 위하여 HTTP 통신 기반의 ReSTFul 서비스 방식과 JSON포맷을 이용하여 OPEN API를 제공함
* Open API 의 키 관리를 위한 OAuth 2.0 활용키 관리 서버를 개발함
  + 1. 기술 범위
* 본 문서의 범위는 의료기기 사용자 데이터 분석결과 데이터와 인터페이스 제공 대한 내용으로 한정되어 있다.

1. 분석
   1. 요구사항 분석
      1. 기술 요구사항

* 본 사업에서 사용되는 인터페이스 시스템을 통한 데이터 인터페이스 편의 제공
* 인터페이스는 HTTP 통신 기반의 RESTful 서비스 방식과 JSON 포멧을 이용하여 Open API 제공
* Open API의 키 관리를 위한 OAuth 2.0 활용 키 서버 개발
  + 1. 구현 범위 및 방법 요구사항
* 인터페이스 모듈은 외부 시스템과의 API 연동을 위한 RESTful 서비스 및 내부 DAO 트랜잭션과, 배치 스케쥴을 처리함
* Open API 및 배치 스케쥴에 적절한 데이터를 전-후 처리 하도록 한다. 또한 연속적인 대량 커넥션에 대비하기 위한 Vert.x 와의 연동을 위해 Vert.x 임베디드 모듈 및 Vert.x 플랫폼간을 연결하는 Haxelcast Data grid 탐재
* Interface Block은 메시지 데이터 처리에 최적화된 동적관리 게이트웨이 개발 모델로 다양한 inbound 모듈 및 outbound 모듈을 정의하고 데이터 처리 및 프로세스 관리를 위한 핵심 기반을 제공함
  + 1. 리소스 요청 및 응답 요구사항
* 인요청 측에서 HTTP 리퀘스트를 전송하고 응답을 파싱 할 수 있도록 시스템에서 사용 가능한 API 명세화를 제공함
* REST는 시스템에서 클아이언트는 HTTP의 문법인 POST, GET, PUT, DELETE를 각 명세에 따라 사용하도록 정의함
* 시스템의 Interface API는 요청결과를 가급적 JSON 포멧으로 리턴하도록 함
  1. 요구사항 기능분석
* HTTP 통신 기반의 RESTful ( POST, GET, PUT, DELETE ) 방식의 Open API 기능
* JSON 포멧을 이용한 요청결과 리턴 기능
* OAuth 2.0 을 이용한 인증기능
* 적절한 데이터 전후 처리
* Vert.x 와 Haxelcast Data grid 를 이용하여 연속적인 대량 커넥션 대비
* API Gateway 를 이용하여 댜양한 inbound 모듈 및 outbound 모듈을 정의하고 데이터 처리 및 프로세스 관리를 위한 핵심 기반을 제공함
* API 명세화 제공
  1. 요구기능 설명
     1. Api Gateway
* 유저별 API별 인증/인가
* 대표 URL (예, api.myservice.com)에서 내부 API Resources(서버) 요청 라우팅
* 로드 밸런싱
* Logging : API 사용 오류 분석 및 Scalability, Metering에 이용
* QoS 조정 (Throttling or Rate Limiting).
  + - 1. Kong (Open Source Api Geteway) 개요

Kong은 확장 가능한 오픈 소스  **API Gateway**입니다. Kong은 RESTful API 앞에서 실행되며 핵심 플랫폼에 플러그인 방식의 추가 기능과 서비스를 제공합니다.

기본적인 api gateway 의 기능을 모두 제공하고 있으며, 현재 Open Source Api Gateway 중 편리한 사용성과 가장 활발한 업데이트 및 확장성을 가지고 있음

* + - 1. Kong (Open Source Api Geteway) 구성

|  |
| --- |
|  |

* + - 1. Kong (Open Source Api Geteway) 특징
* **확장성** : Kong은 단순히 더 많은 기계를 추가함으로써 수평 적으로 쉽게 확장 할 수 있습니다. 즉 플랫폼은 대기 시간을 낮게 유지하면서 거의 모든 부하를 처리 할 수 ​​있습니다.
* **모듈형** : RESTful Admin API를 통해 쉽게 구성 할 수있는 새로운 플러그인을 추가하여 Kong을 확장 할 수 있습니다. (Oauth 2.0 플러그인 지원)
* **모든 인프라** 에서 실행됩니다 : Kong은 어디에서나 실행됩니다. Kong을 단일 또는 다중 데이터 센터 설정과 공개, 비공개 또는 초대 전용 API를 비롯하여 클라우드 또는 사내 구축 환경에 배포 할 수 있습니다.
  + 1. Vert.x
       1. Vert.x 개요
* JVM에서 동작하는 폴리글랏(Polyglot, 여러가지언어로 개발가능하다), 넌 블로킹(Non-Blocking), 이벤트기반(Event-Driven) 비동기 서버 프레임워크다.
* 2011년 팀 폭스(Tim Fox)에 의해 시작되어VMware의 지원속에 발전하다가 지금은 Eclipse Foundation에서 관리하고있다.
* Node.js에서 출발한 프로젝트로 Node.js와 같이 이벤트 기반이지만 싱글쓰레드로 동작하는 Node.js와달리 멀티 스레드로 동작한다.
  + - 1. Vert.x 특징
* 내부 컴포넌트 구성
  + Apache Netty를 서버 엔진으로 사용하고 있고, Vert.x 노드간의 통신을 지원하기 위해서, 데이타그리드 솔루션인 HazelCast를 사용함
* 다양한 언어 지원 (Node.js는 Javascipt만 됨.)
  + Verticle은 JVM 위에서 구동하는 여러가지 언어로 구현이 가능함. Java, JavaScript, Python, Groovy, Scala 를 지원하고 있음
* 클러스터링 지원
  + Vert.x는 클러스터링 지원이 가능
  + 내장된 HazelCast 기능을 사용하면, 인스턴스간의 데이타 공유 구현은 가능하다.

* + - 1. Vert.x 구성

JVM

Netty

Event

Bus

(Hazelcast)

Worker

Thread

Pool

Polyglot

java

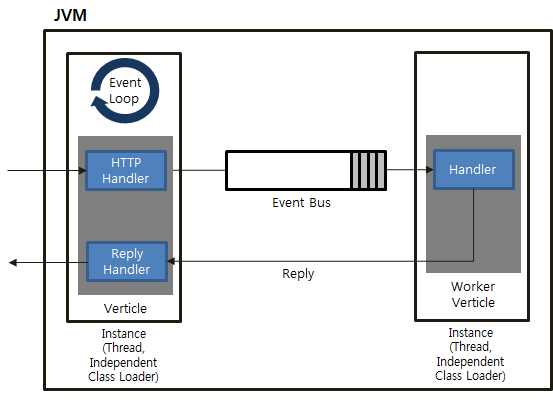
javascript

Python

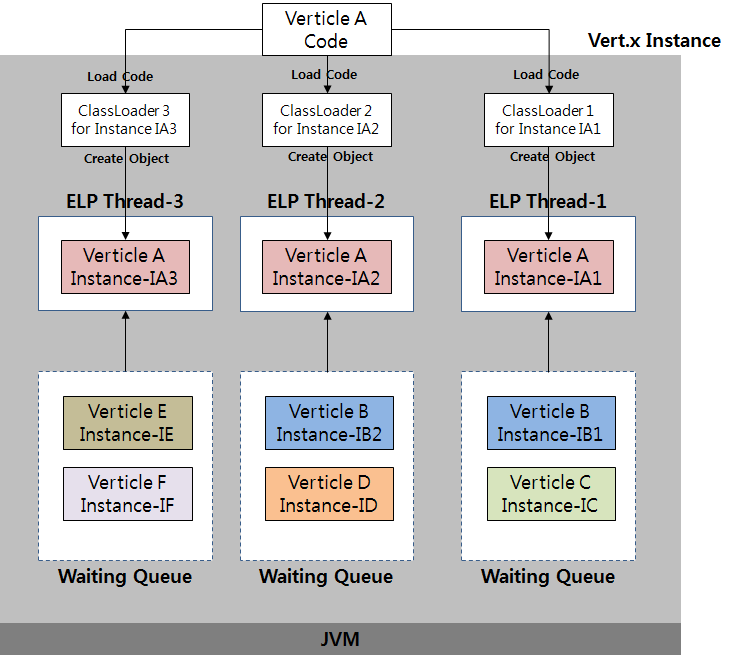
Groovy

Scala

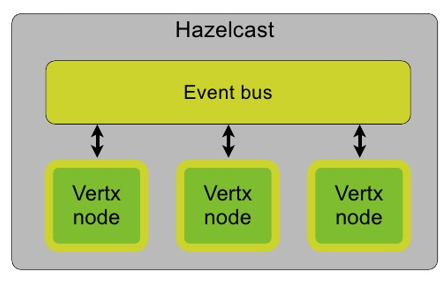
* + - 1. Vert.x Event Loop 및 Event Bus프로세스



* + - 1. Vert.x Multi instance 처리

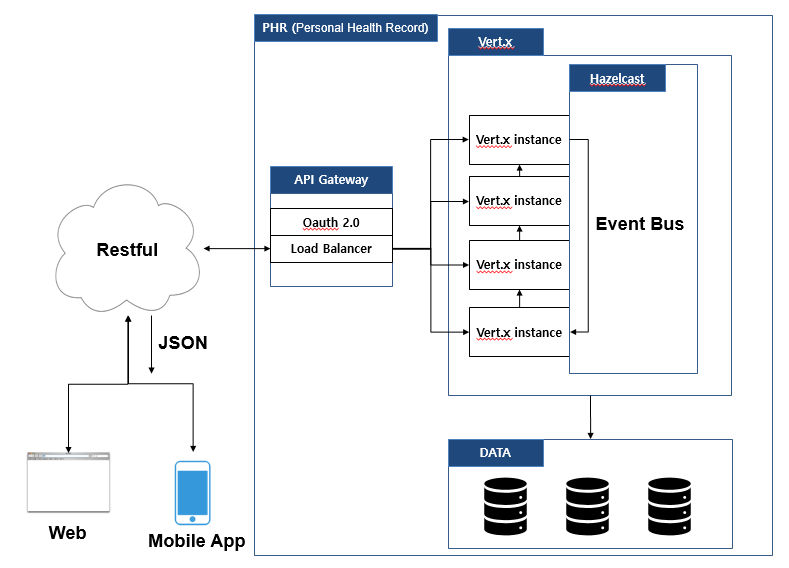


* + - 1. Vert.x 에서의 Hazelcast 특징



* Hazelcast는 분산 In-Memory Data Grid
* Vert.x의 여러 verticle 에서 Hazelcast 를 이용하여 데이터 공유가 가능
* Hazelcast API는 사용자에게 직접 공개되지 않고 Vert.x에서 사용
* Vert.x가 시작되면, Hazelcast는 임베디드 요소 로 시작됨

1. 설계
   1. 구성도



* 1. Vert.x
     1. Vert.x 설치
* <http://vertx.io/download/> 에서 해당 OS 에 맞는 Vert.x 를 다운로드 한다
* Ubuntu 에 적당한 경로에 압축해제를 한다
* Vert.x의 bin 디렉토리 환경변수를 PATH 에 추가한다.
* > vertx version 으로 설치와 설치버전을 확인한다
  + 1. Vert.x Rest 구현
* Vertx-web 모듈을 import 하여 웹서버와, Restful, json 형태의 결과 리턴이 가능하다
  1. Kong (Open Source API Gateway)
     1. Kong Download
* <https://konghq.com/install/> 에서 OS를 선택하여 다운로드
  + 1. Kong install

$ sudo apt-get update

$ sudo apt-get install openssl libpcre3 procps perl

$ sudo dpkg -i kong-community-edition-0.11.2.**\***.deb

* + 1. DataBase install
* Kong 의 설정정보를 관리한다
* [PostgreSQL 9.4+](http://www.postgresql.org/) and [Cassandra 3.x.x](http://cassandra.apache.org/) 의 데이터베이스를 설치한다
  + 1. DataBase 사용자와 데이터베이스 생성
* **CREATE** **USER** kong;
* **CREATE** **DATABASE** kong
* **OWNER** kong;
  + 1. Start Kong
* $ kong start **[**-c /path/to/kong.conf]
  1. Kong 과 api 연계
     1. API를 Gateway Admin 에 추가

$ curl -i -X POST \

--url http://localhost:8001/apis/ \

--data 'name=example-api' \

--data 'hosts=example.com' \

--data 'upstream\_url=http://mockbin.org'

* + 1. API의 KEY 플러그인 추가

$ curl -i -X POST \

--url http://localhost:8001/apis/example-api/plugins/ \

--data 'name=key-auth'

* + 1. API의 사용자 추가

$ curl -i -X POST \

--url http://localhost:8001/consumers/ \

--data "username=Jason"

* + 1. API의 사용자에게 KEY 추가

$ curl -i -X POST \

--url http://localhost:8001/consumers/Jason/key-auth/ \

--data 'key=ENTER\_KEY\_HERE'

1. 변경 내용: 변경이 발생되는 위치와 변경 내용을 자세히 기록(장/절과 변경 내용을 기술한다.) [↑](#footnote-ref-1)