Akka Sensor Platform

Fangqing Wu

About Product

- Sensors to monitor information during the runtime
- Event-oriented, automatic job scheduler, allow dynamically custom programmed and deployed
- Alert when meet the condition given by user

- Environments:
  - Java 8
  - Maven
  - Akka
  - Akka Quartz Scheduler
  - 1.0.snapshot
- Repository:
  `https://github.com/fwqfwq/sensor-platform`
- Original Repository:
  `https://github.com/JIMsZHOU/sensor-platform`

Contents

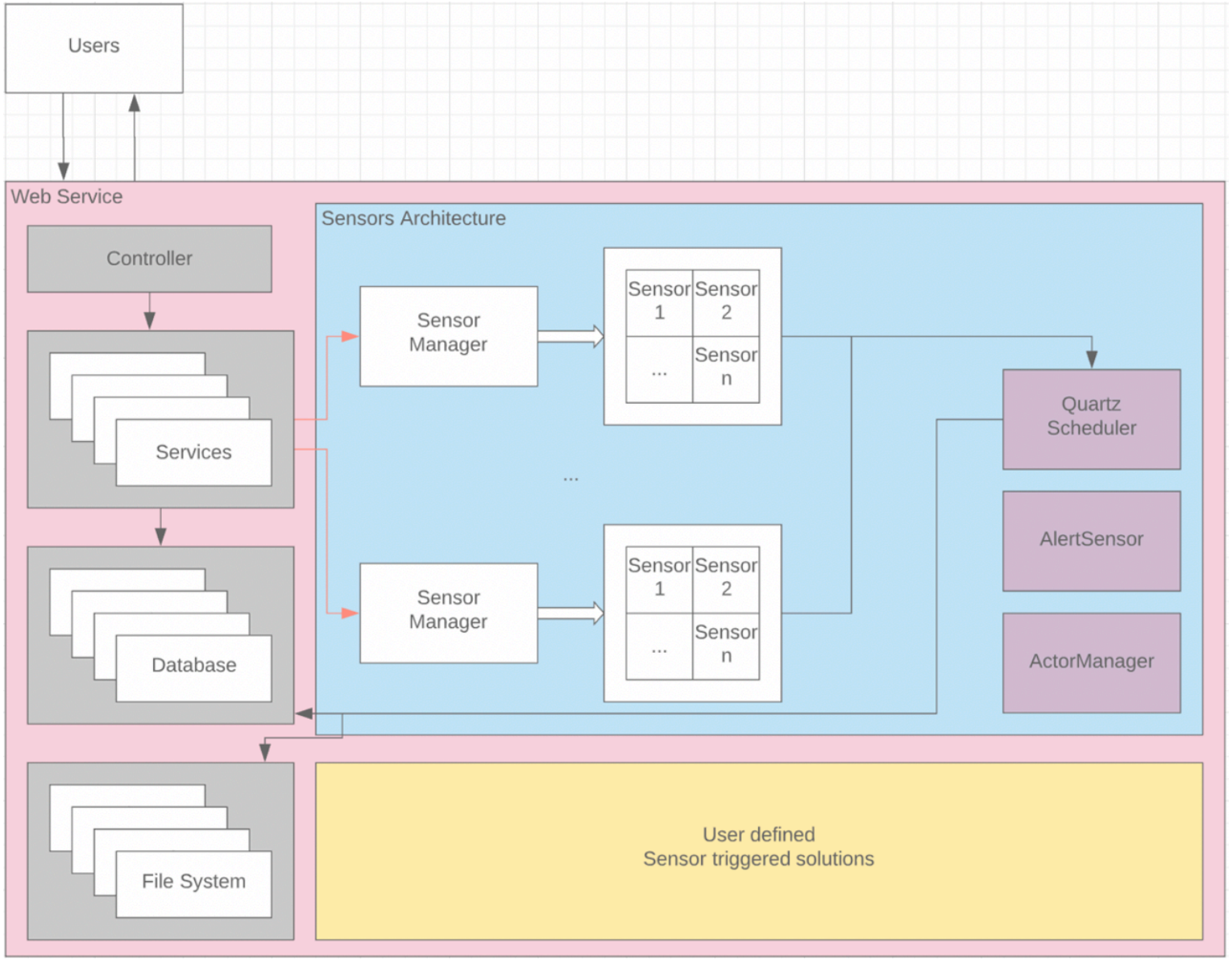
1. Structure
2. Workflow
3. Main Functions
4. Demo

Structure

When user requests web services, the controller passes the requests and different types of sensors would be provoked to deal with specific jobs.

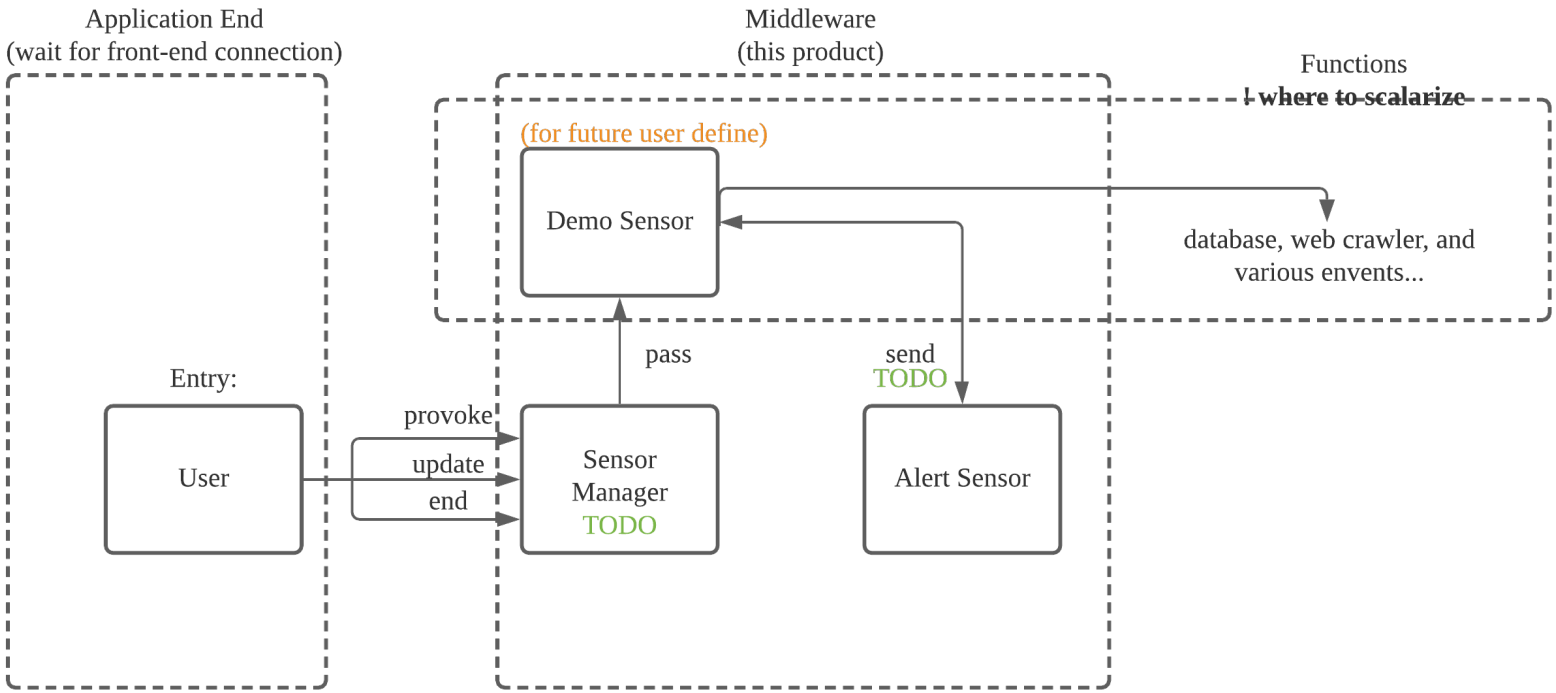
- Base: **Spring Boot**
  Use Spring to build RESTful web services, with Controllers and Services .
- Main tool: **Akka**
  Keep services concurrent, scalable, fault-tolerent and responsive in real-time, with JobManagers.
- Optimizer: **Quartz**
  For a better performance in job scheduling.
- Endpoint: **Swagger api**
- Future Deployment in cloud

Structure


Users
Web Service
Sensors Architecture
Controller
Services
Database
File System
Sensor Manager
Sensor Manager
Sensor 1
Sensor 2
...
Sensor n
Sensor 1
Sensor 2
...
Sensor n
Quartz Scheduler
AlertSensor
ActorManager
User defined
Sensor triggered solutions

Workflow

Take a Demo Sensor for example.



Main Functions

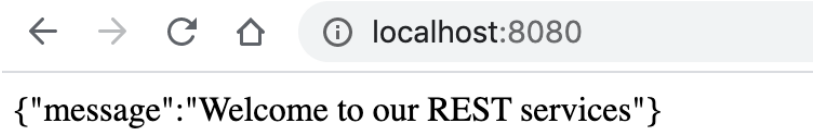
1. Message sensor, listen specified information to specific task.
2. Crawler sensors, listen different signal to trigger different crawler processes.
3. Automated Order Dispatcher, define customized rules to send out orders.

Demo

Demo Sensor

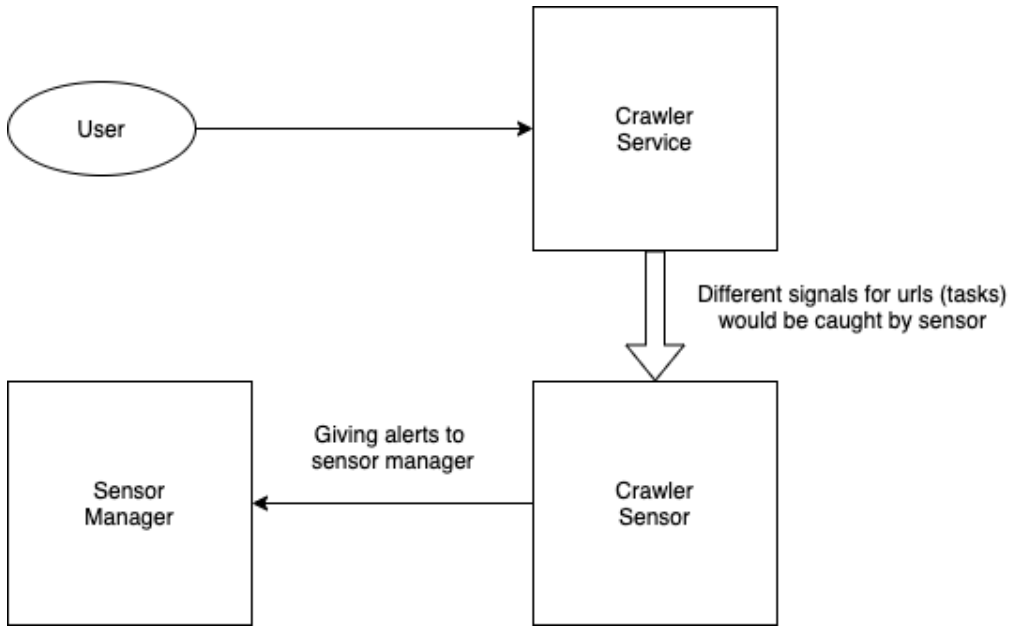
Running Demo:

```java
@RequestMapping(value = "/")
public ResponseEntity<String> index() {
    HttpHeaders httpHeaders = new HttpHeaders();
    httpHeaders.set("Content-Type", "application/json");
    String ans = "{\"message\":\"Welcome to our REST services\"}";
    return ResponseEntity.ok().body(ans);
}
```



Demo

Crawler Sensor



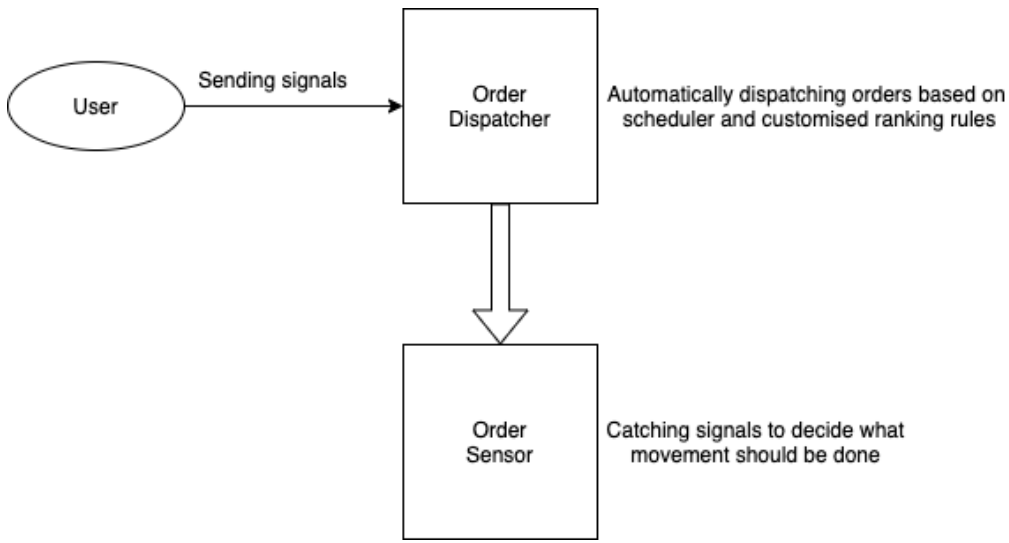
Demo

Crawler Sensor

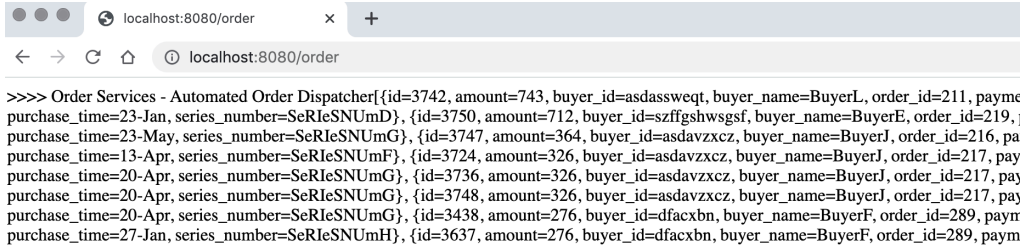
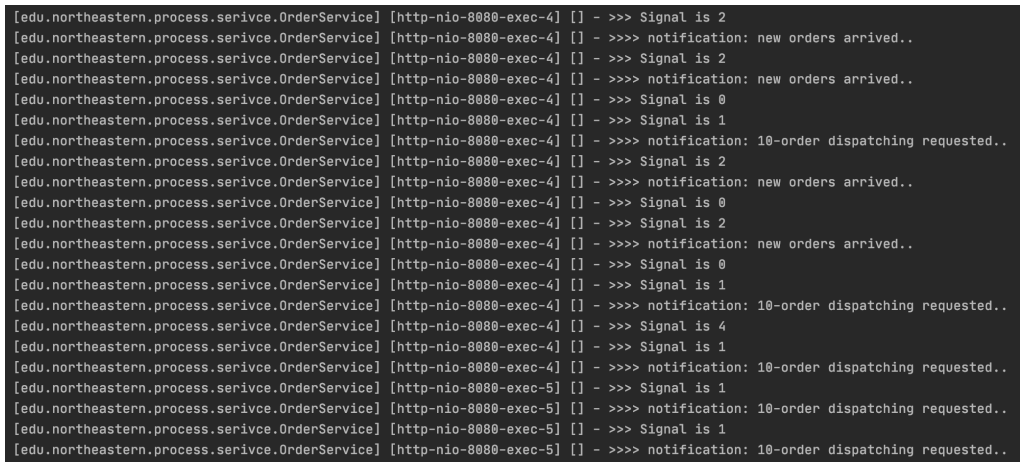
Running:

```
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>>> processing crawler 2...
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>>> done
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>> Signal is 0
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>> Signal is 4
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>> Signal is 7
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>> Signal is 6
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>>> processing crawler 2...
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>>> done
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>> Signal is 5
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>>> processing crawler 1...
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-2] [] - >>>> processing crawler 1...
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>>> done
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>> Signal is 8
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>> Signal is 8
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>> Signal is 6
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-2] [] - >>>> done
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-2] [] - >>> Signal is 8
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-2] [] - >>> Signal is 8
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-2] [] - >>> Signal is 6
[edu.northeastern.process.serivce.CrawlerService] [http-nio-8080-exec-3] [] - >>>> processing crawler 2...
```

Order Dispatcher Sensor



Demo

Order Dispatcher Sensor

Running: