

# Akka Sensor Platform

Concepts / Structure / Demo

Fangqing Wu

## About Product

- Sensors to monitor information during the runtime
- Event-oriented, 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

### ► Original Repository:

<https://github.com/JIMsZHOU/sensor-platform>

### ► Updating Repository:

<https://github.com/fwqfwq/sensor-platform>

# Contents

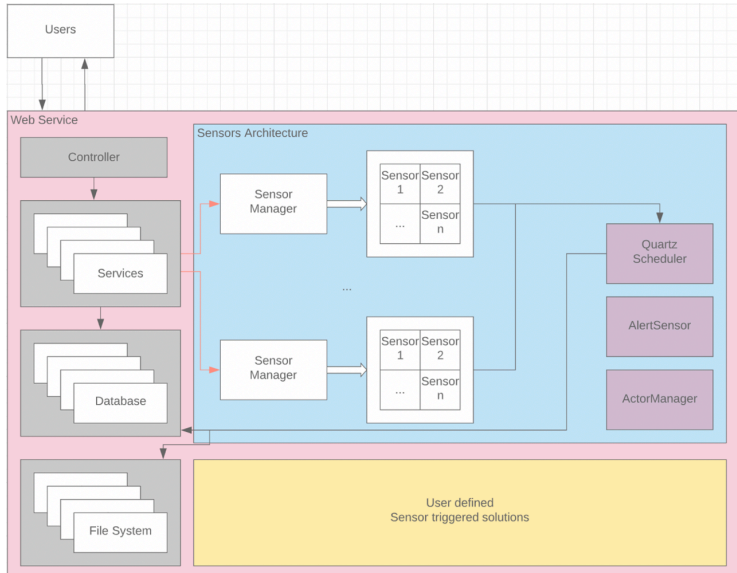
1. Structure
2. Workflow
3. TODO Tasks
4. Demo / Updates

# Structure

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

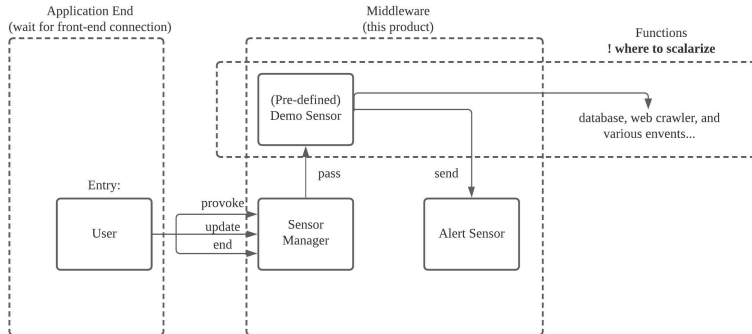
- ▶ Base: **Spring Boot**  
Use Spring to build RESTful web services, with Controllers and Services .
- ▶ Main tool: **Akka**  
Keep services concurrent, scalable, fault-tolerant and responsive in real-time, with JobManagers.
- ▶ Endpoint: **Swagger api**
- ▶ Future Deployment in cloud

# Structure



# Workflow

Take a Demo Sensor for example.



# TODO Tasks

1. Create much more various types of event sensors.
2. Optimize sensors' process and lifecycle, set deadlines.

# Demo / Updates

## Demo Sensor

### Running Demo:

demo-controller Demo Controller	
GET	/ index
HEAD	/ index
POST	/ index
PUT	/ index
DELETE	/ index
OPTIONS	/ index
PATCH	/ index
GET	/start demoProcess
event-controller Event Controller	
POST	/event/add createEvent
GET	/event/delete/{eventId} deleteEvent
POST	/event/testToJson testToJson
POST	/event/update updateEvent
user-controller User Controller	
GET	/user/add-{userId} addUser
GET	/user/getAll getUsers
GET	/user/getById-{userId} getById
GET	/user/read-{userId} readUser
GET	/user/send/{userId}-{message} sendMessage

```
@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);
}
```



localhost:8080

```
{"message": "Welcome to our REST services"}
```



# Demo / Updates

## Demo Sensor - Updates

### TODO tasks:

- ▶ Modify the DemoService
- ▶ Crack the bottleneck issue in SensorManager (\*SensorManager.java)

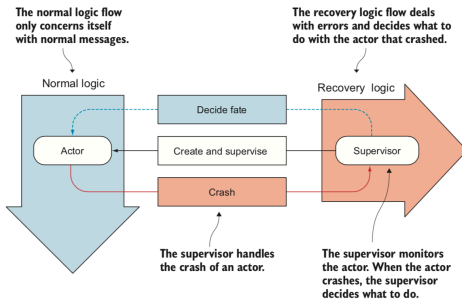


Figure: cr. Akka in Action

## Demo Sensor - Updates

- ▶ `SensorManager.java`

Adding `preStart()` `postStop()` functions

## Web Crawler Sensor

New Service added.

Updated:

```
+-- process
|   +- controller
|   |   +- CrawlerController.java
|   +- beans
|   |   +- CrawlerEntity.java
|   |   +- //CrawlerConfiguration.java
|   |   +- //CrawlerEntity.java
|   +- dao
|   |   +- CrawlerRepo.java
|   +- services
|   |   +- CrawlerService.java
|   +- 
```

Ongoing...