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

Structure

When user requests web services, the controller passes the requests and different types of sensors would be provoked to deal with specific jobs.
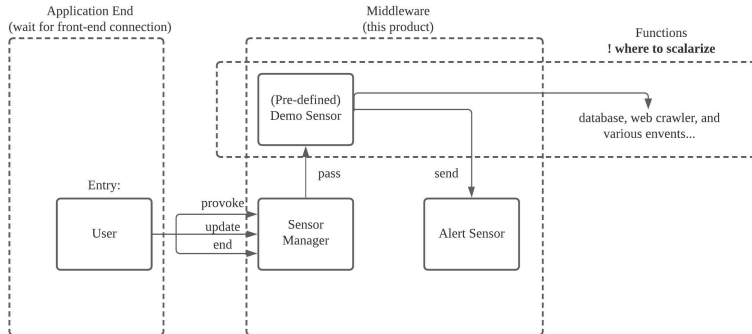
▶ Base: **Spring Boot**
  Use Spring to build RESTful web services, with Controllers and Services .

▶ Main tool: **Akka**
  Keep services concurrent, scalable, fault-tolerent and responsive in real-time, with JobManagers.

▶ Endpoint: **Swagger api**

▶ Future Deployment in cloud

Structure
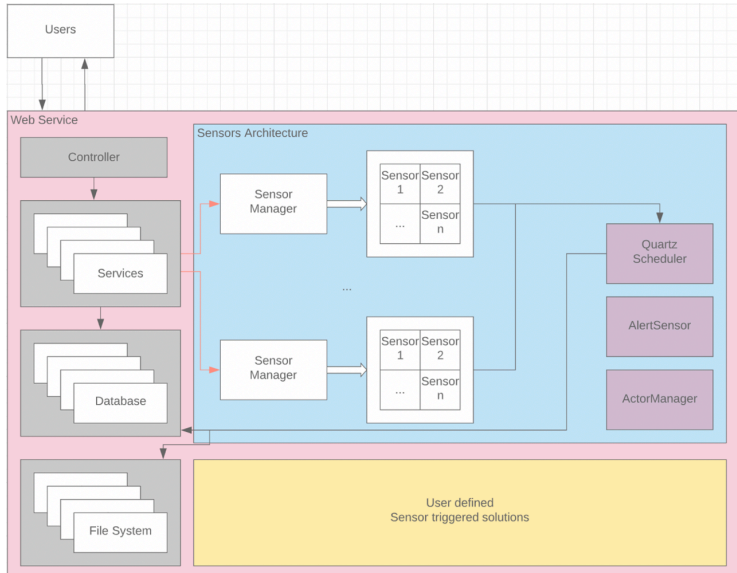
Workflow

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 Sensor - Updates

TODO tasks:
- ▶ Modify the DemoService
- ▶ Crack the bottleneck issue in SensorManager (*SensorManager.java)

Figure: cr. Akka in Action
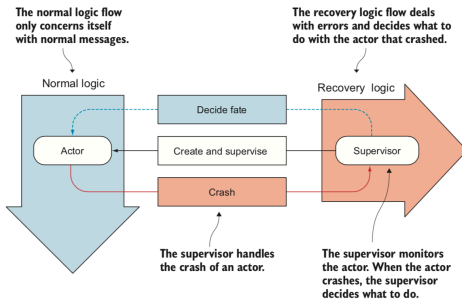
Demo Sensor - Updates

- SensorManager.java

Adding preStart() postStop() functions

Demo / Updates

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...