# Event Driven微服务

# **Spring-boot-axon-sample**

系统架构如下所示：



在Task类中定义Domain，核心代码如下所示：

*@Aggregate*

*public class Task {*

*@AggregateIdentifier*

*private String id;*

*//定义CommandHandler，根据CommandBus中的Command进行对应处理*

*@CommandHandler*

*public Task(CreateTaskCommand command) {*

*//直接发送Event到EventBus中*

*apply(new TaskCreatedEvent(command.getId(), command.getUsername(), command.getTitle()));*

*}*

*...*

*@CommandHandler*

*void on(UnstarTaskCommand command) {*

*apply(new TaskUnstarredEvent(command.getId()));*

*}*

*}*

在TaskEntryUpdatingEventHandler中定义EventHandler

*@Component*

*public class TaskEntryUpdatingEventHandler {*

*private final TaskEntryRepository taskEntryRepository;*

*@Autowired*

*public TaskEntryUpdatingEventHandler(TaskEntryRepository taskEntryRepository) {*

*this.taskEntryRepository = taskEntryRepository;*

*}*

*//定义EventHandler*

*@EventHandler*

*void on(TaskCreatedEvent event) {*

*//将数据写入到数据库中*

*TaskEntry task = new TaskEntry(event.getId(), event.getUsername(), event.getTitle(), false, false);*

*taskEntryRepository.save(task);*

*}*

*.....*

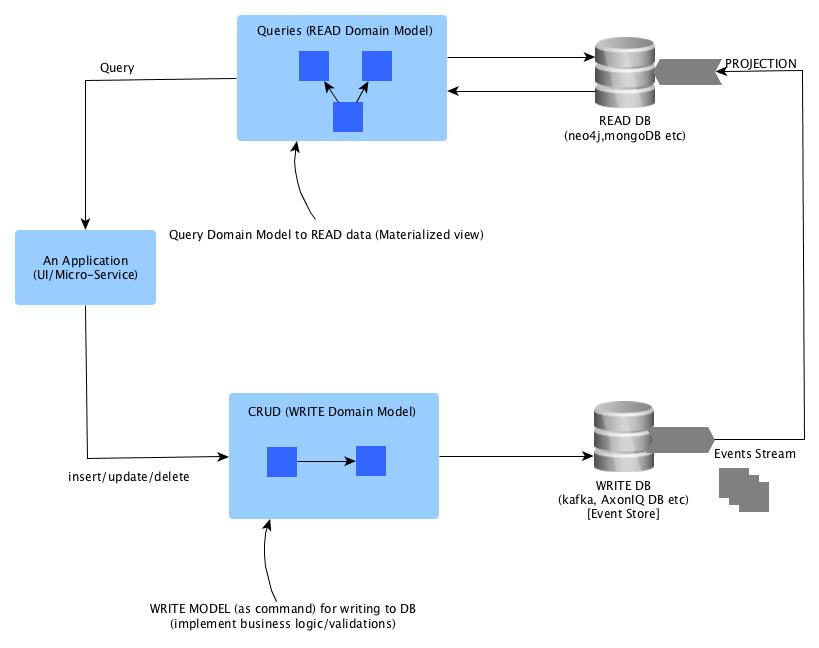
*}*

在例子上仅定义了Command的执行，并没有涉及到Query的处理。

https://github.com/avthart/spring-boot-axon-sample

# **AxonIQ Example**

在该示例中使用Event Driven架构，实现了读写模块分离，模块之间通过Event来交互，架构如下所示：



https://github.com/mgorav/EventDrivenMicroServiceUsingAxonIQ