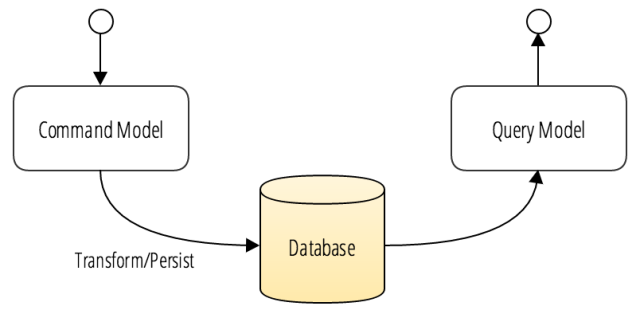
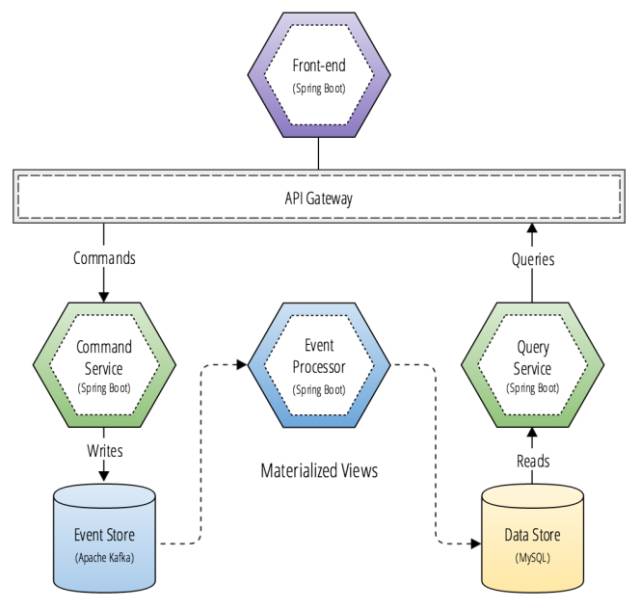
CQRS微服务实践

微服务是面向服务架构(SOA)架构风格的一种变体，将应用程序构建为一系列松散耦合的服务，在微服务体系结构中，服务是细粒度的，协议应该是轻量级的。将应用程序分解为不同的小型服务的好处是它提高了模块性，使应用程序易于理解、开发和测试。它使小型自主小组能够独立开发、部署和扩展各自的服务，从而实现并行发展，允许通过持续的重构来实现单个服务的体系结构，从而连续迭代和部署。但是随着服务数量的增加，服务的治理难度就会增加，这就带来了协议轻量级的概念。但是微服务开发事务性业务应用是一件具有挑战性的工作，CQRS(Command Query Responsibility Segregation)，命令查询职责分离的微服务开发方法可以实现可扩展且高效的查询，其是一种架构风格：



其基本理念是将数据读写的模型相互分离，CQRS在事件驱动架构领域比较热门的实现。使用Spring Boot实施CQRS的简单微服务框架如下图：



将微服务分为命令端、查询端和事件处理器，这个三个部分相互独立部署。

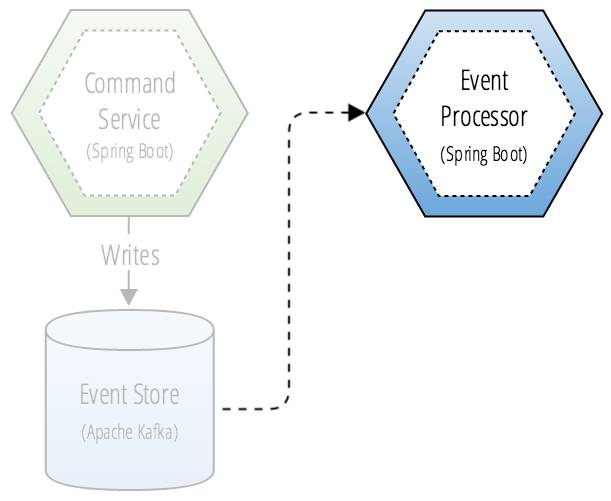
1. 命令端

提供REST API，接受通过HTTP发送的请求，请求采用命令的形式，可以驱动对微服务所拥有的领域数据状态进行更改。对领域数据的任何写入都将以命令形式以从API请求流出，处理导致数据库发生的更改的操作。

命令触发操作，操作触发领域事件

1. 事件处理器

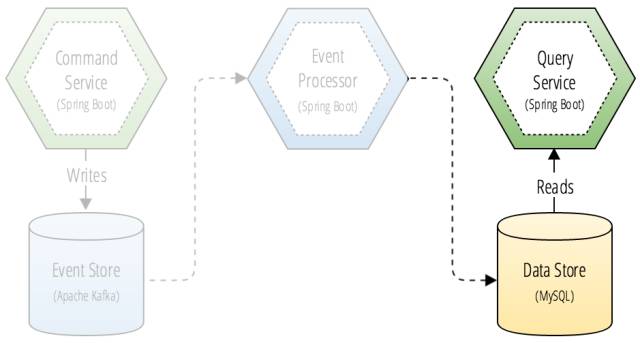
CQRS组件采用Worker应用的形式，负责获取领域事件，事件处理器是无状态的，并侦听来自事件存储的消息，对传入的事件消息采取操作



领域事件可以生成多个事件，这些事件可以发送到其他微服务，通过这种方法可以发布和订阅来自限界环境之外的应用的领域事件。

1. 查询端

事件处理器首先要负责应用可改变领域整合状态的领域事件，每个领域事件可以用来更新数据库记录，形成描述整合的增量实体化视图。反过来，查询端将提供REST API，允许HTTP客户端读取从已处理事件生成的实体化视图



查询端组件中的限制是领域数据是只读的。此系统中所有状态的更改都会从命令端流入，然后形成可在查询端读取的实体化视图。

# **Micro-Starter**

简单示例：

https://github.com/mgorav/EventDrivenMicroServiceUsingAxonIQ

相关术语：

领域模型

CQRS，命令查询分离模式

Axon , 事件驱动

微服务

Serverless框架

https://www.sohu.com/a/164970101\_747818

https://www.cnblogs.com/sky233/p/7795402.html

https://github.com/sky233/micro-starter

Cqrs-microservice-sampler:

https://github.com/benwilcock/cqrs-microservice-sampler