

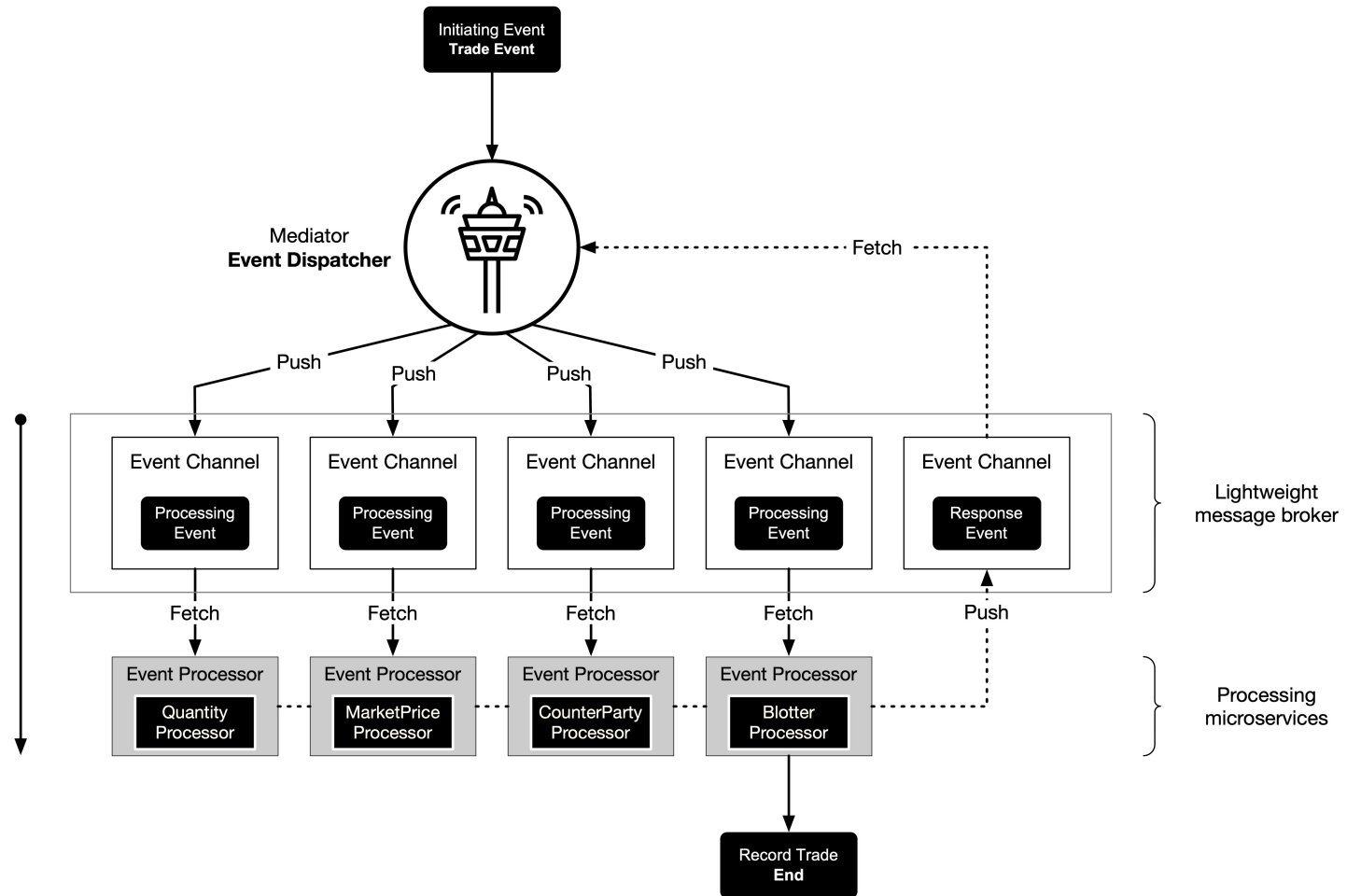


Trade Capture System Prototype

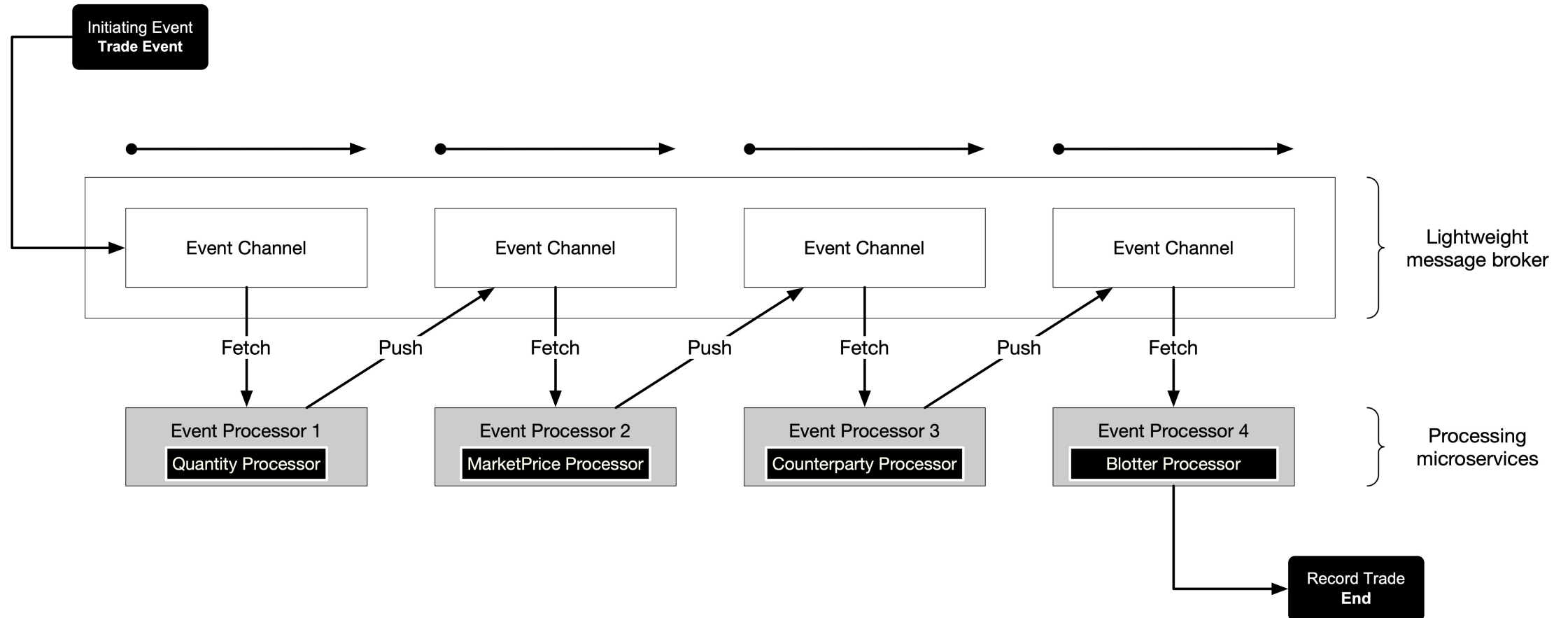
By Anastasia Chanbour



An Event-Driven Architecture: The Mediator Pattern



An Event-Driven Architecture: The Broker Pattern



Important considerations

In the mediator pattern, event processors run asynchronously while in the broker pattern, they run sequentially.

Event processors run concurrently, either as independent microservices or as separate threads.

Communication between the message broker and event consumers must be nonblocking.

In the broker pattern, event processors mutate the event object before passing it to the next link in the chain.

In the mediator pattern, event processors may return a response event to indicate to the event dispatcher.



Questions to the business team

The mediator pattern enables concurrent execution of multiple event processors at the cost of increased complexity. The broker pattern chains event processors and may block if an event processor takes too much time. **What is the trade-off between complexity and performance in real-world trading scenarios?**

Decoupling can be achieved at different levels, either locally with multithreading or at scale over the internet. Local threads offer a much lower latency while large cloud-native architectures over the internet are powerful. **Is lower latency or computing power more critical in trading applications?**

With multithreaded decoupling, there is no message broker middleware and shared memory queues are used to pass events instead of events channels. **How thread-safe queues and thread synchronization are implemented?**

In the current model, each event processor can only process one event at a time. In the ideal model, multiple events could either be balanced between multiple instances of the same processor or a new thread could be created for each new event. **What is the ideal model?**

While message brokers (such as RabbitMQ or Apache Kafka) are widely used in cloud architectures today, securing these infrastructures can be tricky. **In the case of events containing sensitive data, should these events be encrypted inside the message queues, and if yes, what would be the performance hit?**



Sources

- Chapter 6: Architectural Patterns in *Hands-On Design Patterns with Java*
- Chapter 2: Event-Driven Architectures in *Software Architecture Patterns*