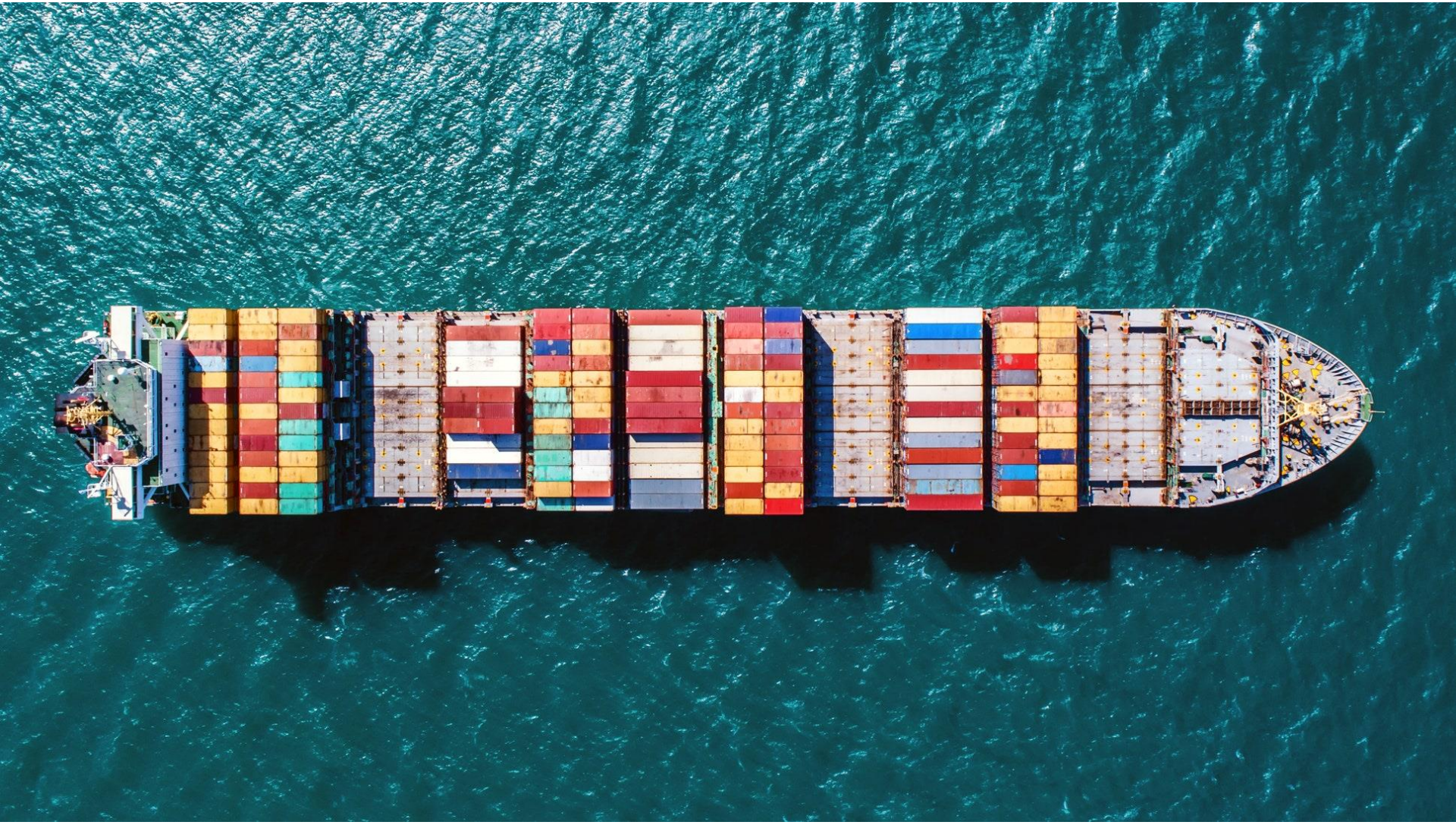
Spring DDD Cargo Tracker

Flying

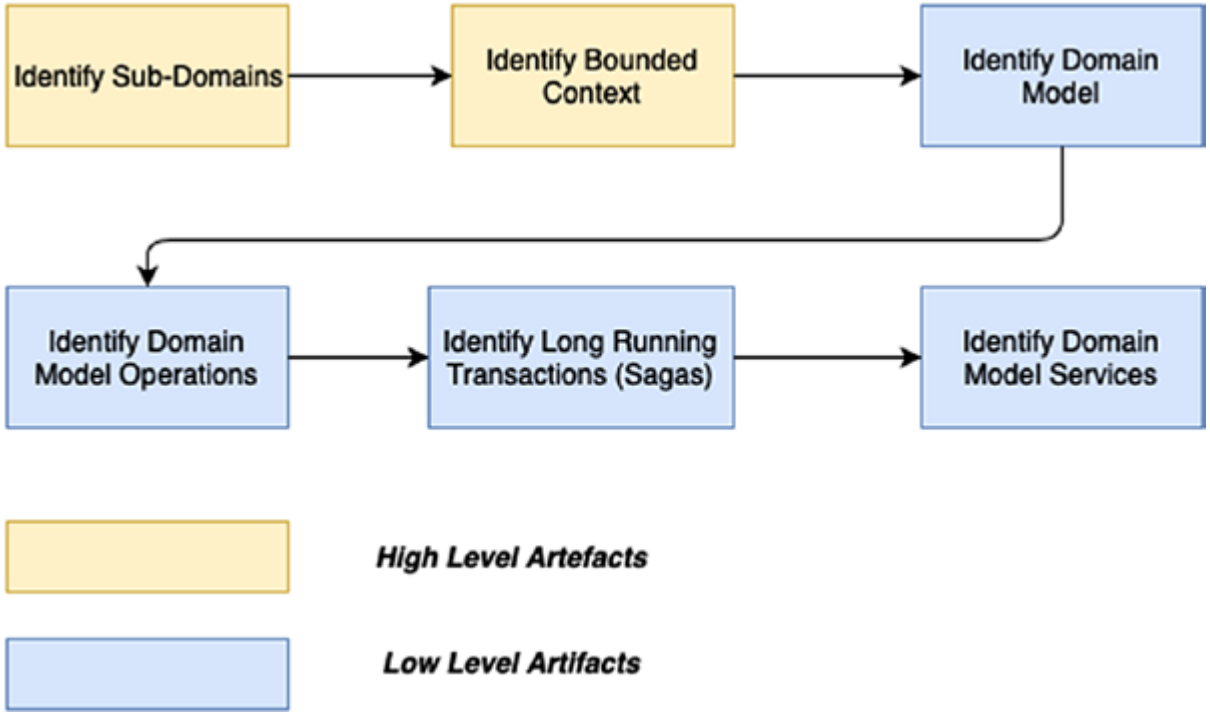
23 novembre 2022

Patrick Ceppi

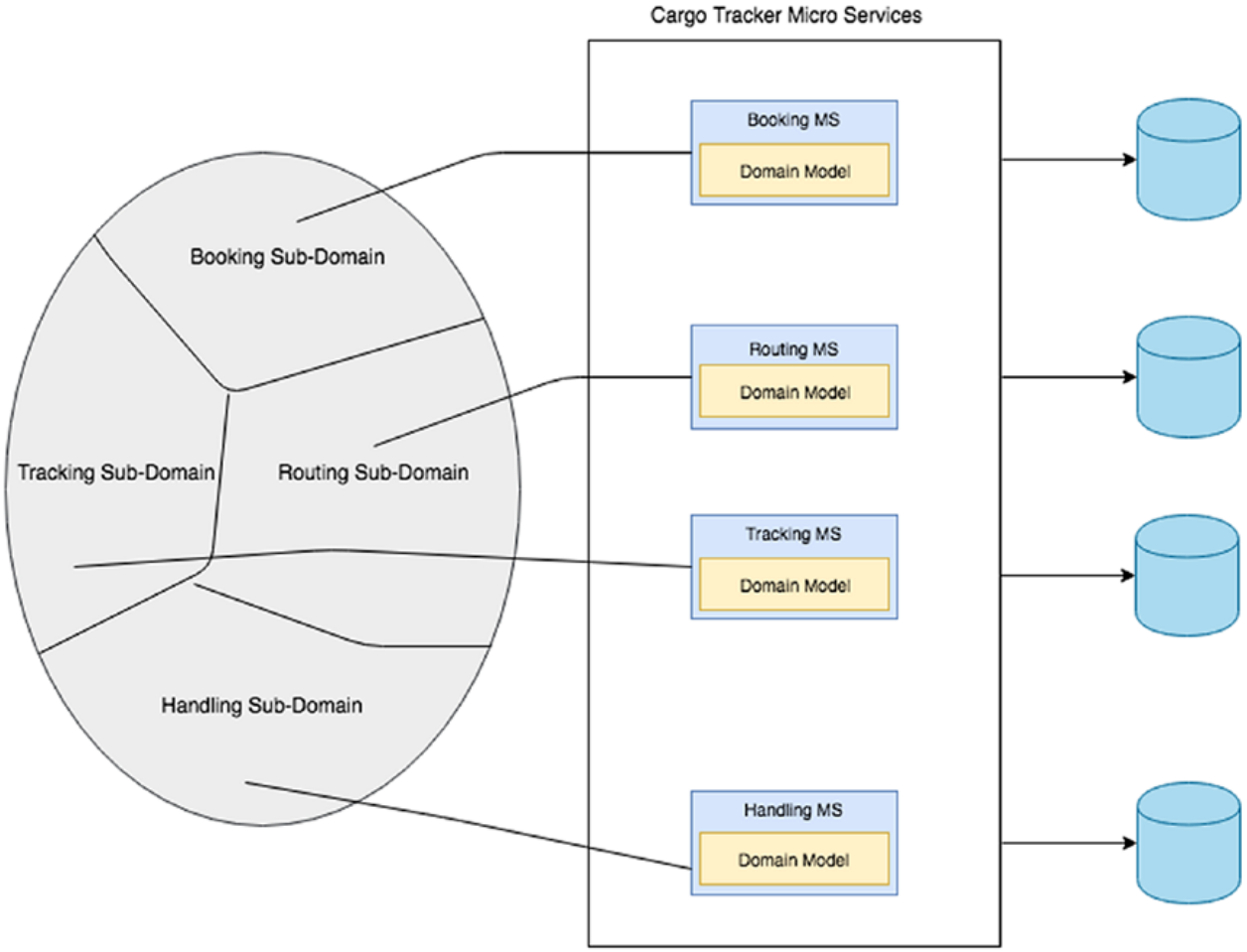
DDD Cargo tracker



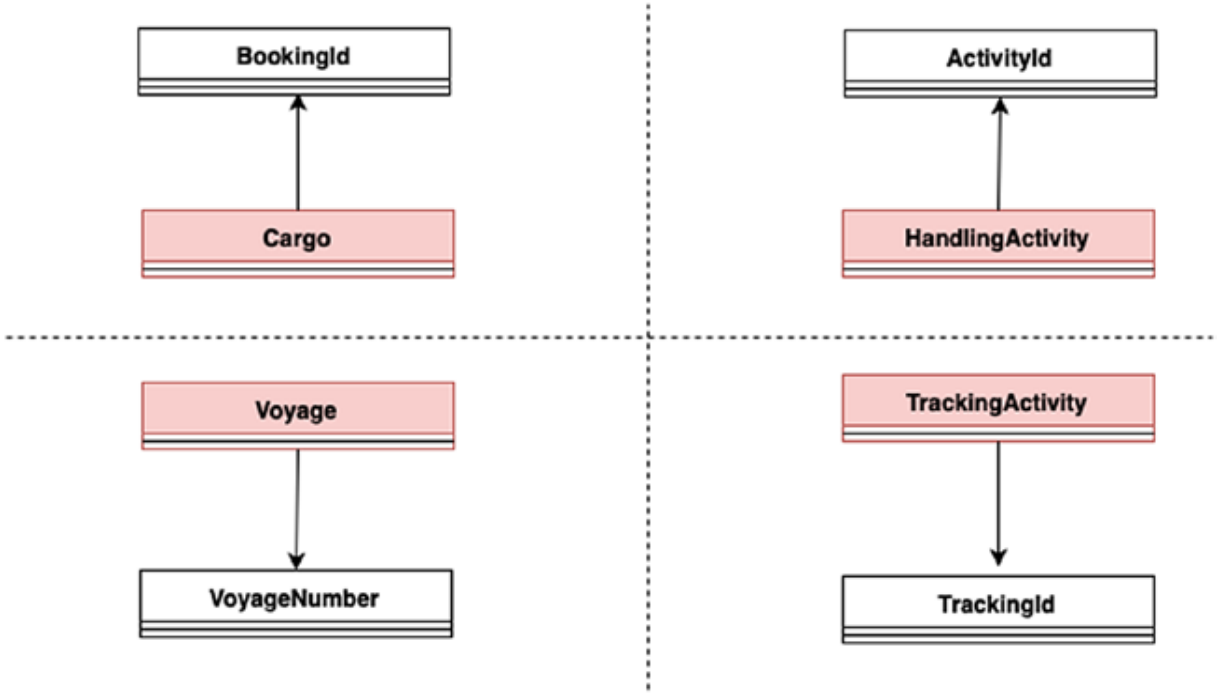
DDD Cargo tracker



Sub-domains/Bounded contexts

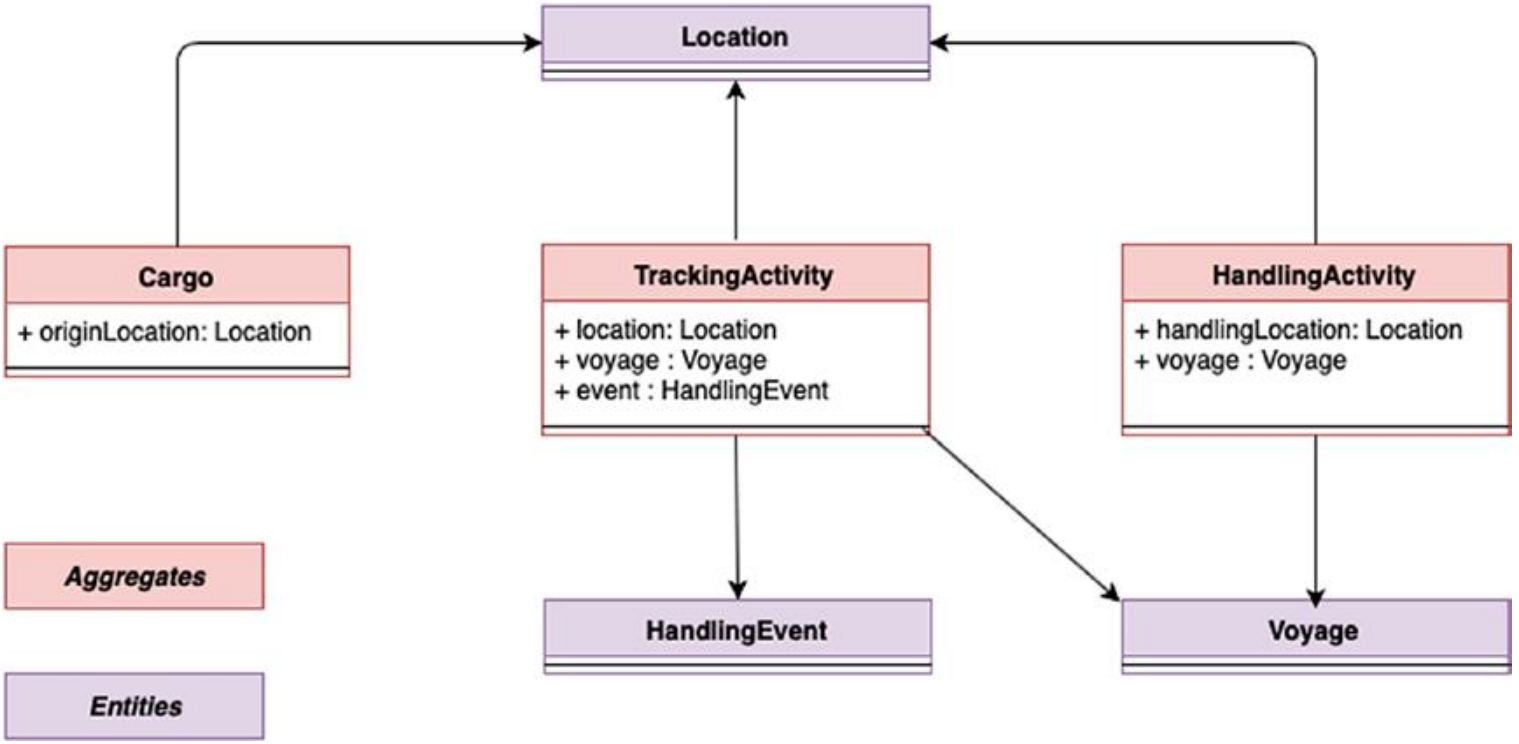


Aggregates and identifier

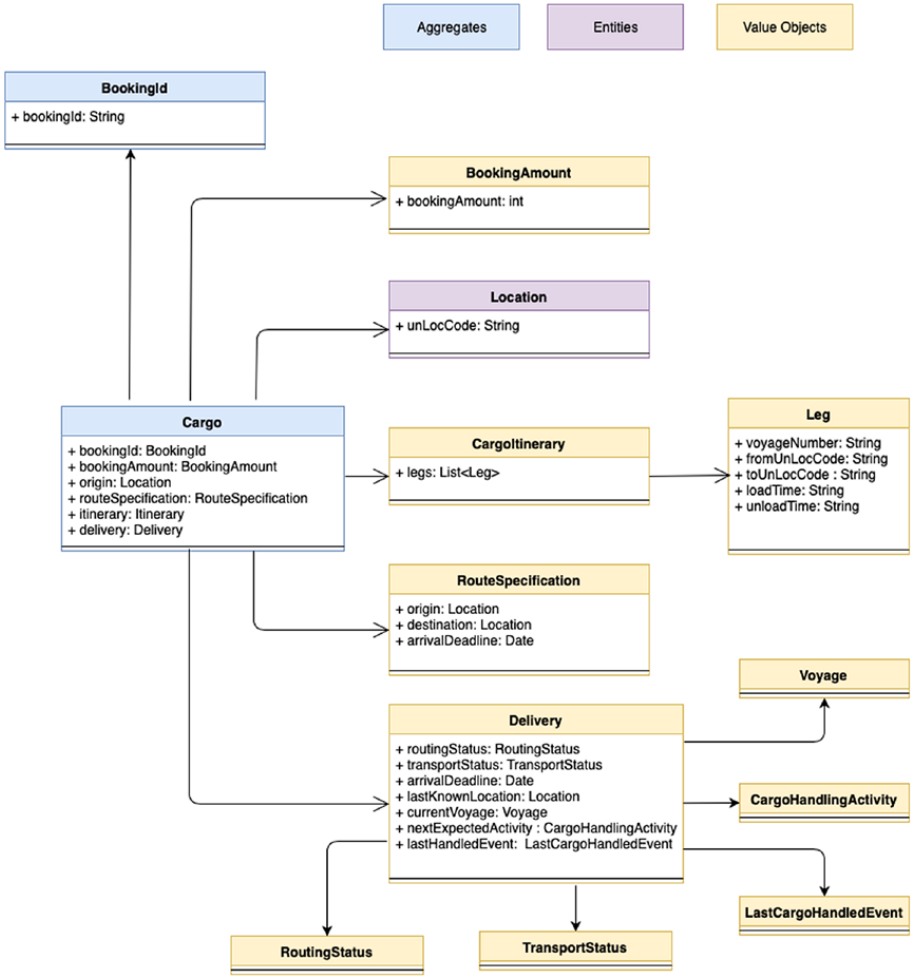


Domain model – Core Domain Model

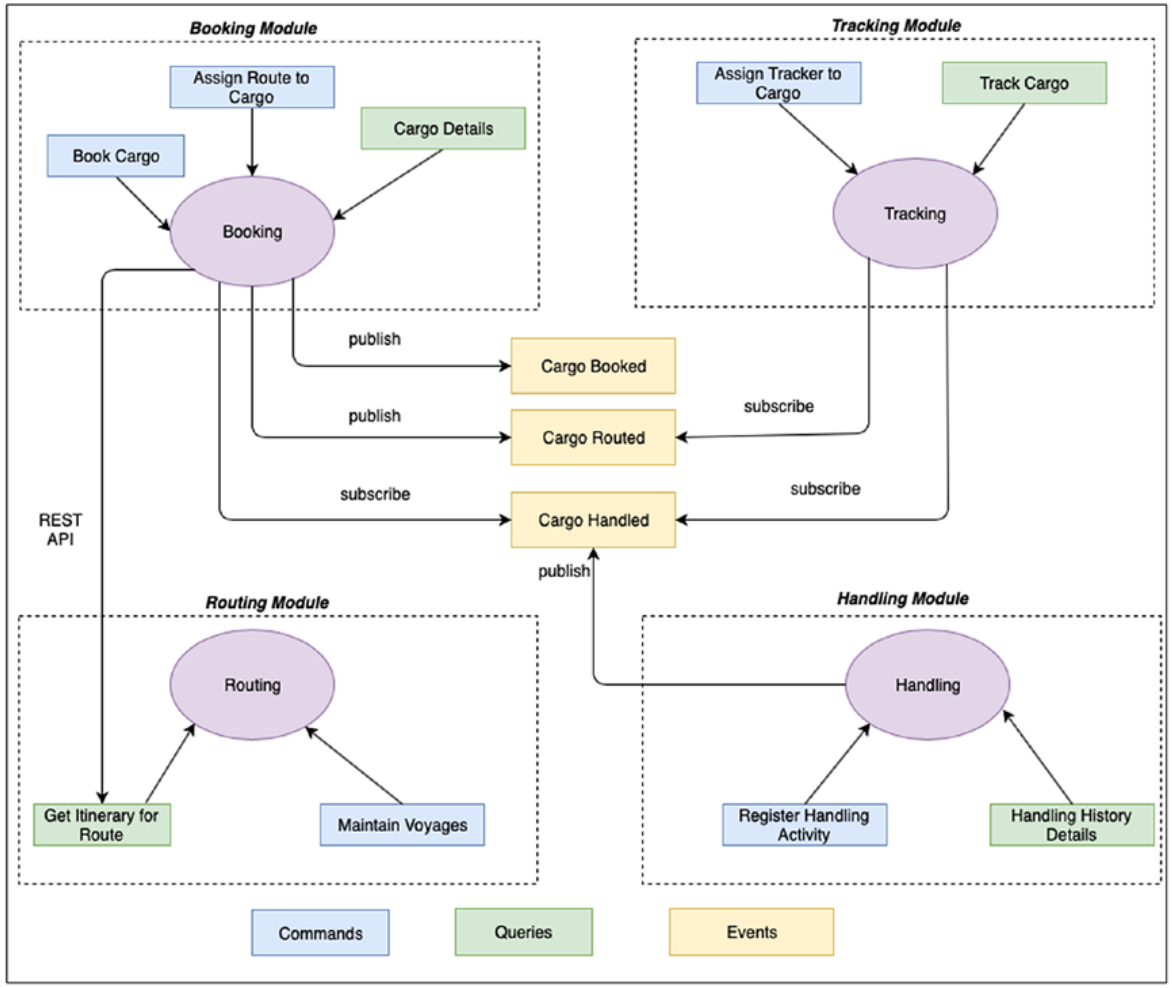
Entities



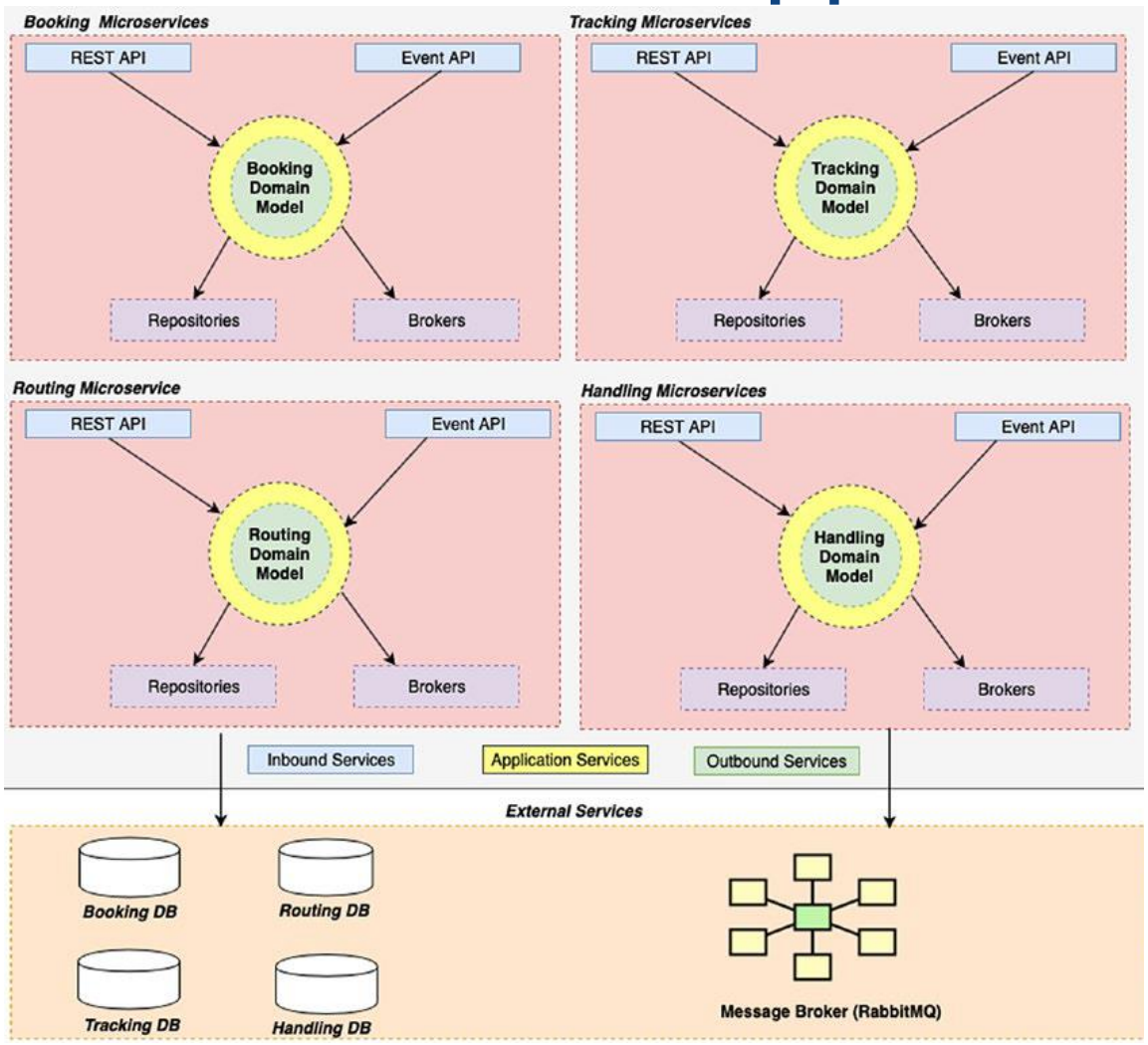
Value objects



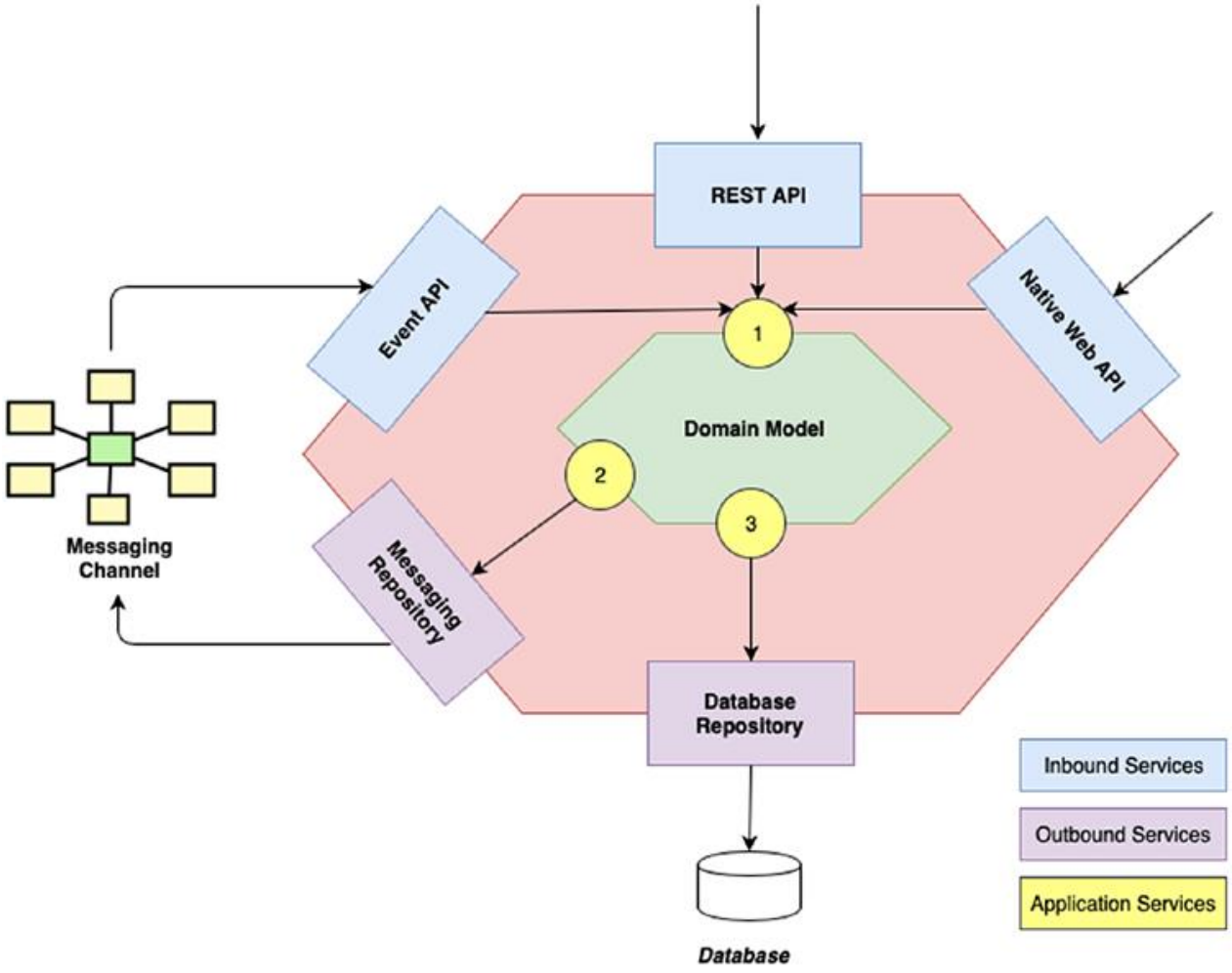
Commands, Queries, Events



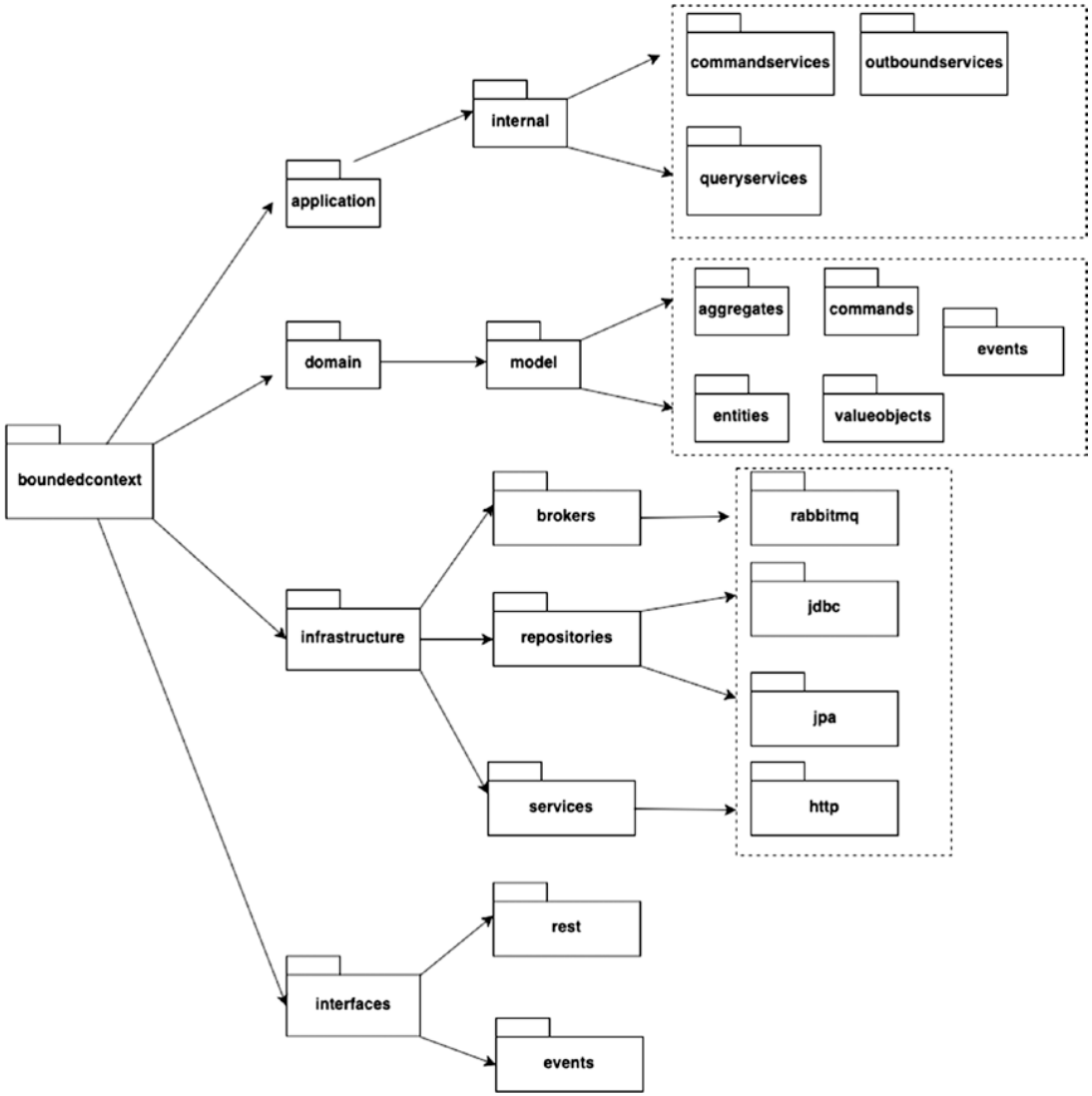
Inbound, Outbound, Application



Hexagonal architecture



Spring implementation 1/3



Spring implementation 2/3

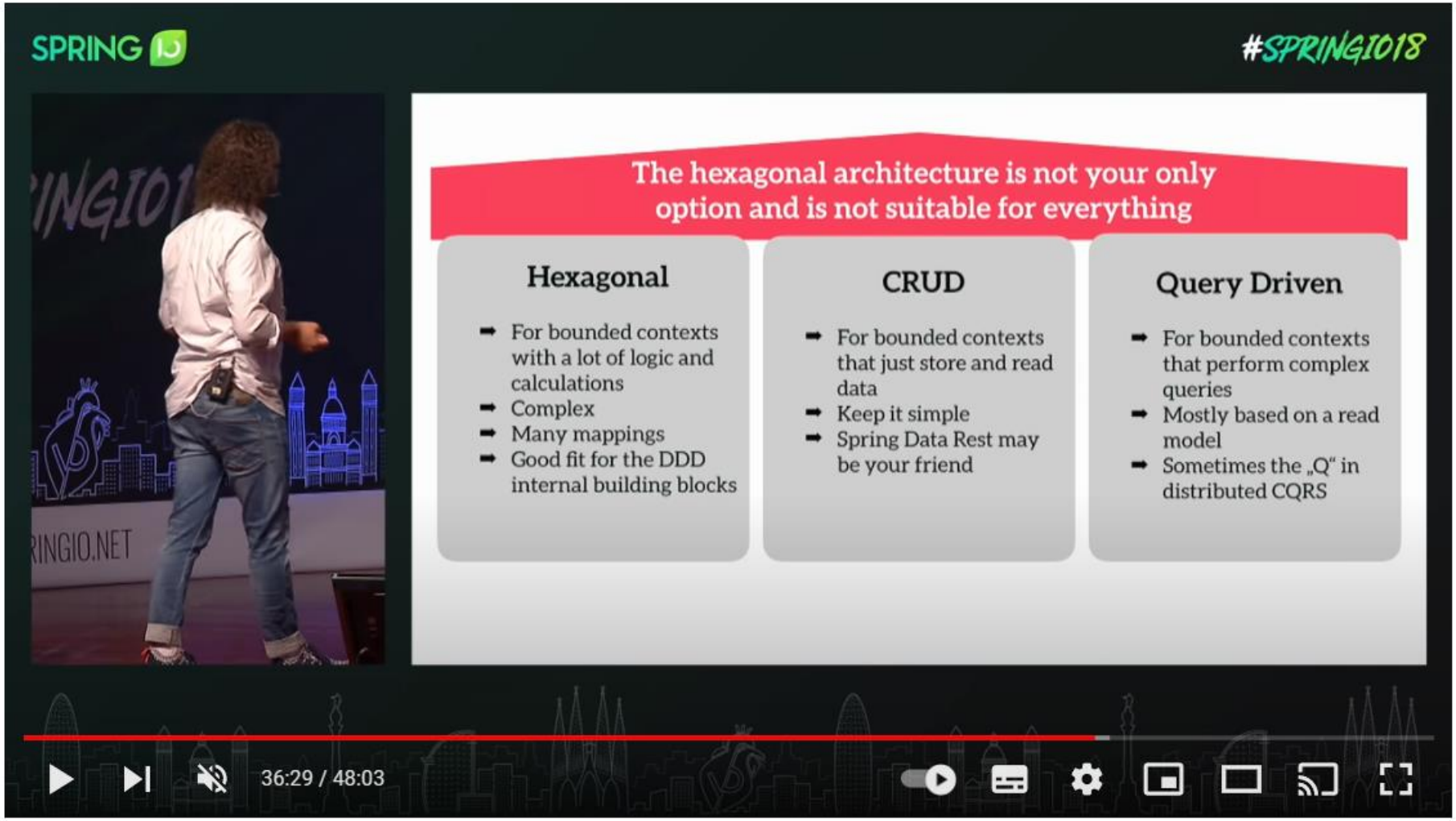
```
@Entity
public class Cargo {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Long id;
    @Embedded
    private BookingId bookingId; // Aggregate Identifier
    @Embedded
    private BookingAmount bookingAmount; //Booking Amount
    @Embedded
    private Location origin; //Origin Location of the Cargo
    @Embedded
    private RouteSpecification routeSpecification; //Route Specification of
    the Cargo
    @Embedded
    private CargoItinerary itinerary; //Itinerary Assigned to the Cargo
    @Embedded
    private Delivery delivery; // Checks the delivery progress of the cargo
    against the actual Route Specification and Itinerary
}
```

Spring implementation 3/3

```
/**
 * Constructor Command Handler for a new Cargo booking
 */
public Cargo(BookCargoCommand bookCargoCommand){
    this.bookingId = new BookingId(bookCargoCommand.getBookingId());
    this.routeSpecification = new RouteSpecification(
                new Location(bookCargoCommand.getOriginLocation()),
                new Location(bookCargoCommand.getDestLocation()),
                bookCargoCommand.getDestArrivalDeadline()
        );
    this.origin = routeSpecification.getOrigin();
    this.itinerary = CargoItinerary.EMPTY_ITINERAR
    since the Cargo has not been routed yet
    this.bookingAmount = bookingAmount;
    this.delivery = Delivery.derivedFrom(this.rout
            this.itinerary, LastCargoHandledEvent.
}
```

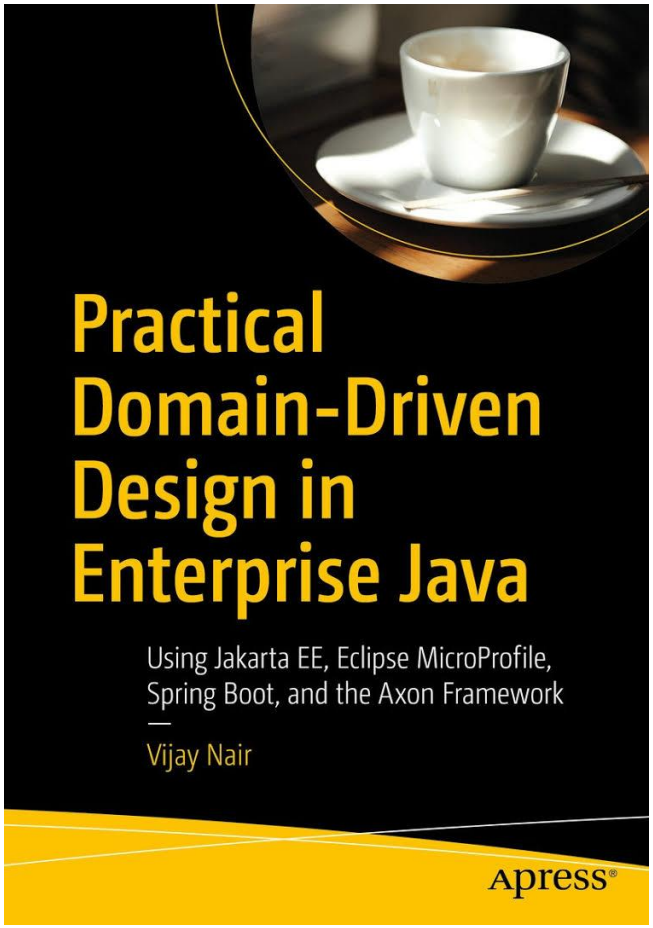
```
/**
 * Command Handler for the Route Cargo Command. Sets the state of the
 Aggregate and registers the
 * Cargo routed event
 * @param routeCargoCommand
 */
public void assignToRoute(RouteCargoCommand routeCargoCommand) {
    this.itinerary = routeCargoCommand.getCargoItinerary();
    // Handling consistency within the Cargo aggregate synchronously
    this.delivery = delivery.updateOnRouting(this.routeSpecification,
            this.itinerary);

}
```

Conclusioni



https://www.youtube.com/watch?v=a9dF7fnArq0

Fonte



By Vijay Nair

Pubblicato il 6 settembre 2019

https://github.com/practicalddd

Contattami

email patrick.ceppi@eoc.ch
github @pac19
twitter @pac_19

Grazie per l’attenzione

Domande?