Use case diagram (nodes and labels):

- **constructor**
- **President** — Deploy → constructor
- **request shipment**
- **Customer** — pickup location, destination, package dimensions → request shipment
- **drop off package at pickup location**
- Customer — confirm order → **pay for shipment**
- **generate optimized delivery route** — request shipment → generate optimized delivery route
- generate optimized delivery route — determine fees → pay for shipment
- **pick up package from pick up location**
- **drop off package at exchange location**
- **pay driver**
- **Driver1**
- **pick up package at exchange location**
- **request driving route**
- **Driver2** — current location, duration of availability , car dimensions → request driving route
- **drop off package at destination**

```
request  --->  optimization  --->  transport  --->  fullfilled

          driver requests          drivers          final driver
          route with location,     notified         drops off
          how long they are        of their         package
          available for, and       routes           at
          car dimensions                            destination

customer
requests
shipment
with origin,
destination,
dimensions

deploy

deployment
```

Janus
struct Location {int latitude, int destination, string address
struct Shipment { byte32 shipper_addr, int width, int height, int length, Location origin, Location destination }
struct Driver {byte32 driver_addr, int car_weight_capacity, int width, int height, int length, Location location}
struct Route { Driver driver, Shipment package, Location pickup_location, Location destination_location}
mapping(address=>Shipment) shipments
mapping(address=>List<Route>) routes
mapping(address=>Driver) drivers
modifier driverAvailable
modifier driverClose
modifier packageFits
modifier isCarrier
constructor()
requestShipment()
requestDrivingRoute()
pickup(Shipment)
dropoff(Shipment)