

## Taxi call service

### 구성원 객체

Drivers

Taxi Platform Operator

User

Payment

### RideRequest

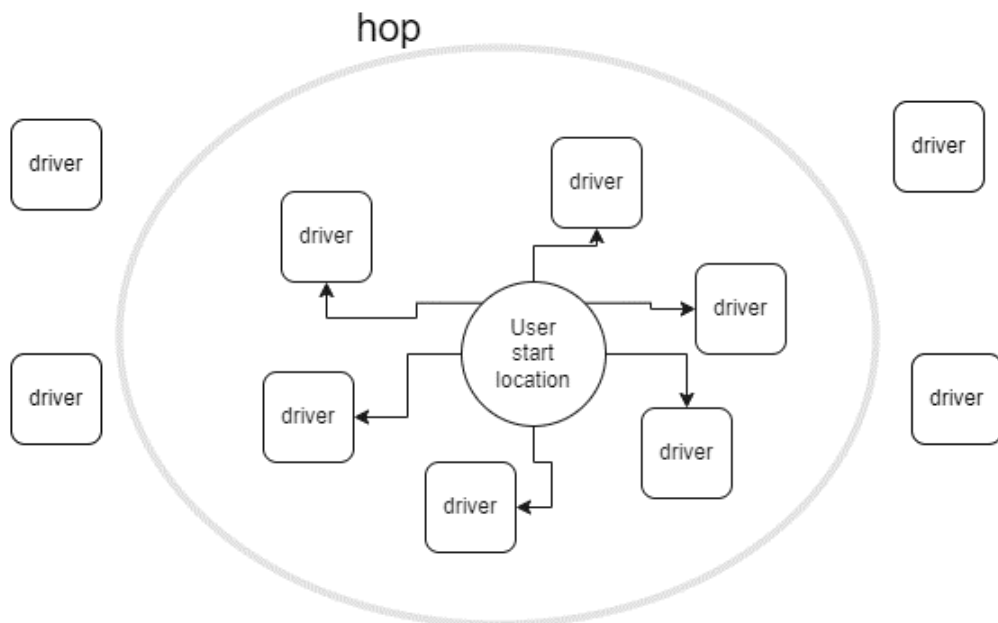
Taxi call Service를 이용하는 User은 platform에게 탑승 요청을 보낸다. 해당 요청은 확실히 송수신되었는지 확인되어야 함으로 request/reply로 처리한다. 작성된 current location 대신 start location으로 바꿔 처리하였다. 이를 통해 보다 명확한 장소 선정과 driver 등의 다른 노드 간의 current location을 헷갈리지 않을 수 있다.

{Timestamp, User-ID, start location, Destination}

## PickRequest

Platform은 User start location을 기준으로 일정 거리(hop)에 있는 모든 driver에게 topic direct로 수신한다. 개인정보 보호를 위해 driver의 판단을 위한 필수 정보(출발, 도착 지점)만 제공한다. 여건 상 driver 한 노드만 구현하였지만 실제 서비스에서는 다수의 driver들이 platform에게 정보를 받는다. Hop 안에 있는지 없는지 판단하는 기준은 location & status를 통해 지속적으로 push되는 driver의 current location를 따른다.

{Timestamp, start location, Destination}



## PickupRequestResponse

Request/reply

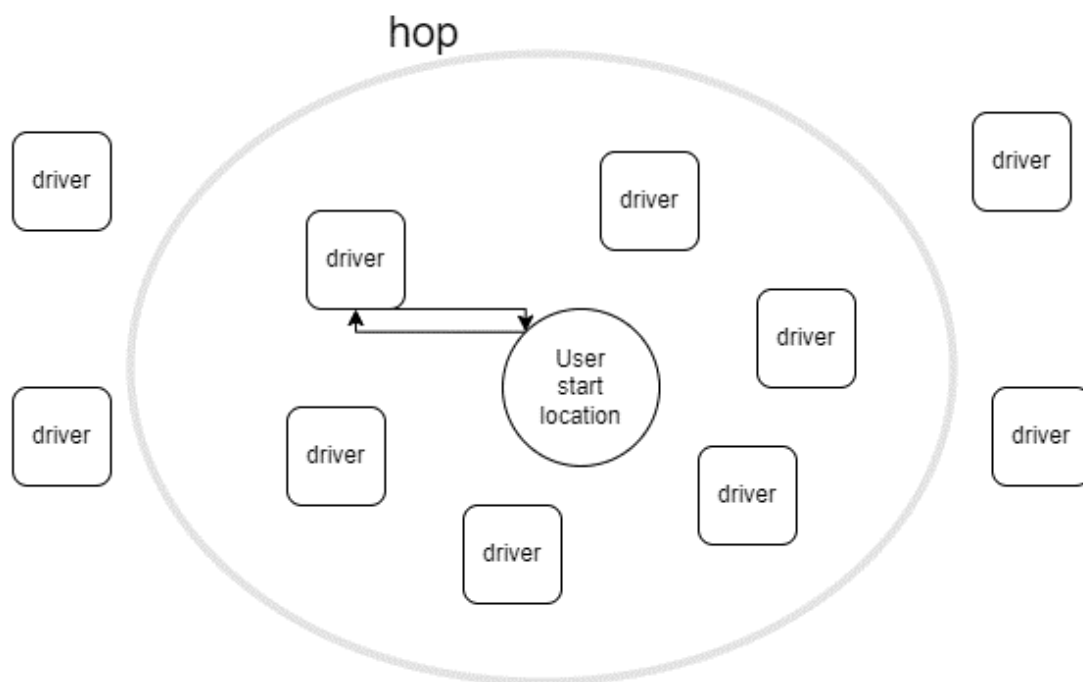
Result 값은 해당 운전 요청을 수락하는 True와 거절하는 False로 나눈다.

Request {Timestamp, Ride-ID, Result}

Reply {result}

반드시 한 driver에게만 배정해야 함으로 platform에게 가장 먼저 응답이 된 driver에게만 배정되었다는 뜻인 'True' reply를 보내고, 나머지는 배정에 실패했다는 False reply를 보낸다. True reply를 받은 driver만 session(connection)을 유지한다. 즉, 해당 driver만 다음 통신을 수행한다.

True reply를 받지 못한 driver은 timeout으로 해당 event 프로세스에서 내리고 데이터 삭제. Platform은 연결을 종료한다. 만약 일정 시간 동안 platform이 PickupResponse를 못 받았을 때 다시 PickRequest를 흘뿌린다



## RideRequestResponse

guaranteed

User를 운전할 driver이 결정되었을 경우 platform은 User에게 탑승에 필요한 정보를 제공한다.

Result 값은 True 또는 False로 기록한다.

{Timestamp, Estimated Time of Arrival, TaxiNumber}

## PickupComplete

guaranteed

해당 event를 수행하는 driver는 User의 start location과 driver의 current location이 동일할 때 platform에게 User 탑승이 완료되었다는 메시지를 platform에게 송신한다.

{Timestamp, Ride-ID, Location}

만약 platform이 일정거리(hop) 안의 최대 소요 시간 동안 받지 못했다면 해당 드라이버에게 pickupcomplete를 보내달라는 요청과 location & status를 물어본다.

## DropoffComplete

guaranteed

Driver의 current location == User의 Destination일 때 Driver이 platform과 payment에게 user 서비스 이용이 완료되었다는 메시지를 송신한다.

# 만약 platform이 일정거리(hop) 안의 최대 소요 시간 동안 받지 못했다면 해당 드라이버에게 DropoffComplete를 보내달라는 요청과 location & status를 물어본다.

{{Timestamp, Ride-ID, Location}}

## DropoffCompleteUser

request/reply

유저에게 제대로 도착하였는지 확인한다. Requester인 Platform이 User에게 잘 도착하였는지 물어보고 User는 User의 current location == User의 Destination일 때 잘 도착했다는 메시지를 송신한다. Result 값은 잘 도착했다는 True 값 또는 도착하지 못하였다는 False 값으로 나눈다. Ack는 'Did you arrive safely? I confirm your arrival.'문자열이다.

Request: {Timestamp, ask}

Reply: {Timestamp, Result}

## PaymentStart

guaranteed

Platform이 모두 확인한 이후 payment에게 user\_id와 ride\_id, cost를 송신한다.

## PaymentRequest

guaranteed

payment가 User에게 송신한다.

{timestamp, cost, accout}

---

## 소감

Platform이라 명칭 하였으나 판단을 내리고 노드를 연결지어주는 event broker의 역할에 더 가깝다.

코드 구현에 힘이 들었다.

실제로 고려해야 하는 상황이 매우 많았다. 기술적 구현의 부족으로 전부 실행시키기 못한 바가 아쉬울 따름이다.

taxi\_call\_service가 아닌 조금 더 pub/sub 구조에 적합한 다른 business model를 상상하고 구현해봤으면 한다.