


## 도로 API 구현 시나리오

## 초기 구현 형태 - 1

BackEnd  
(SSE)



<input type="checkbox"/>	idx	xy_date	car_id	x_point	y_point
<input type="checkbox"/>	1	2023-10-06 13:41:31	1	14131825.049445724	4506461.050685771
<input type="checkbox"/>	2	2023-10-06 13:41:33	1	14132866.820321498	4506151.196376818
<input type="checkbox"/>	3	2023-10-06 13:41:36	1	14133936.747166885	4505672.330626618
<input type="checkbox"/>	4	2023-10-06 13:41:38	1	14135429.013556506	4505775.61539627
<input type="checkbox"/>	5	2023-10-06 13:42:09	2	14132115.994465088	4504921.168665519
<input type="checkbox"/>	6	2023-10-06 13:42:11	2	14132979.444199963	4504939.947714547
<input type="checkbox"/>	7	2023-10-06 13:42:13	2	14133730.270056376	4504489.250537888
<input type="checkbox"/>	8	2023-10-06 13:42:15	2	14135044.215305097	4504564.366733998
<input type="checkbox"/>	9	2023-10-06 13:42:18	2	14135813.811807921	4504864.831518437
<input type="checkbox"/>	10	2023-10-06 13:43:01	3	14135626.105343819	4508470.408931706

FrontEnd

SSE 기술을 통하여  
좌표 값을 실시간으로 받아온 후  
그 좌표값을 전달.

(조회 시 마지막 idx를 저장하여  
다음 조회 시에는 마지막 idx 뒤에 있는  
값들만 조회 가능)

## 초기 구현 형태 - 2

FrontEnd

<input type="checkbox"/>	idx	xy_date	car_id	x_point	y_point
<input type="checkbox"/>	1	2023-10-06 13:41:31	1	14131825.049445724	4506461.050685771
<input type="checkbox"/>	2	2023-10-06 13:41:33	1	14132866.820321498	4506151.196376818
<input type="checkbox"/>	3	2023-10-06 13:41:36	1	14133936.747166885	4505672.330626618
<input type="checkbox"/>	4	2023-10-06 13:41:38	1	14135429.013556506	4505775.61539627
<input type="checkbox"/>	5	2023-10-06 13:42:09	2	14132115.994465088	4504921.168665519
<input type="checkbox"/>	6	2023-10-06 13:42:11	2	14132979.444199963	4504939.947714547
<input type="checkbox"/>	7	2023-10-06 13:42:13	2	14133730.270056376	4504489.250537888
<input type="checkbox"/>	8	2023-10-06 13:42:15	2	14135044.215305097	4504564.366733998
<input type="checkbox"/>	9	2023-10-06 13:42:18	2	14135813.811807921	4504864.831518437
<input type="checkbox"/>	10	2023-10-06 13:43:01	3	14135626.105343819	4508470.408931706

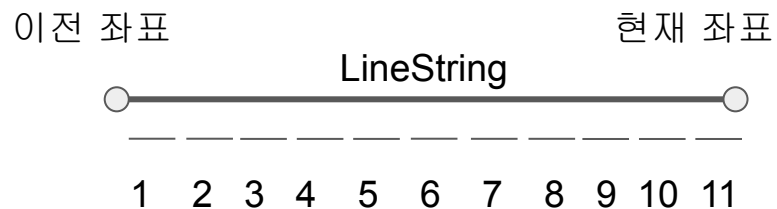
좌표를 전달 받은 프론트엔드는  
반복문을 통해 좌표 덩어리를 처리한다.  
반복문안에서는  
이전 좌표 + 이번 좌표를 서로 이어  
LineString 형태로 변환

이전 좌표

현재 좌표



### 초기 구현 형태 - 3

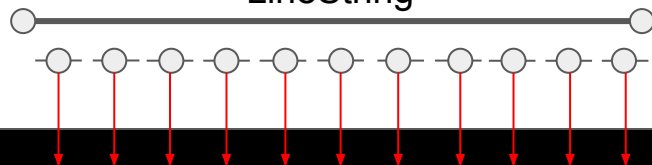


이전 좌표와 현재 좌표로 이어진 `LineString` 객체를  
10미터 단위로  $n$ 분으로 나누어  
 $n$ 개 각자 `LineString`으로 생성

## 초기 구현 형태 - 4

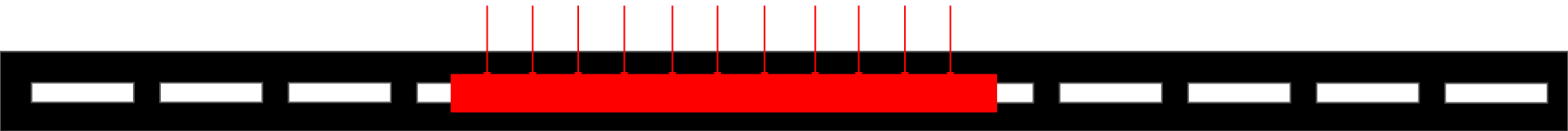
이전 좌표                      현재 좌표

LineString



**n개로 나뉜 세그먼트 LineString 들을  
가운데 기준으로 가장 가까운 도로의 수선의 발을 구한다**

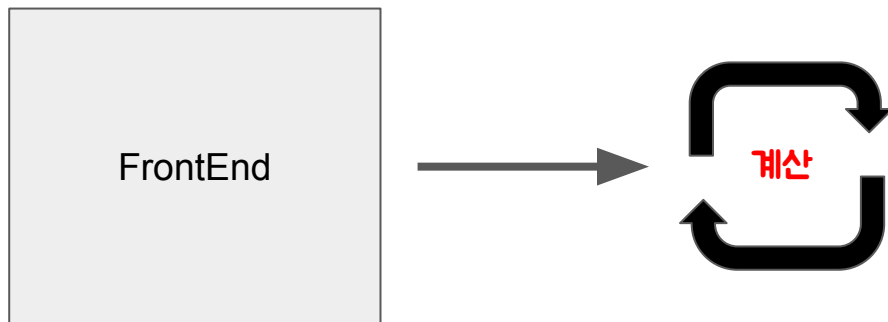
## 초기 구현 형태 - 5



수선의 발로 구해진 좌표들을  
모두 LineString 객체로 차례대로 이어준다.

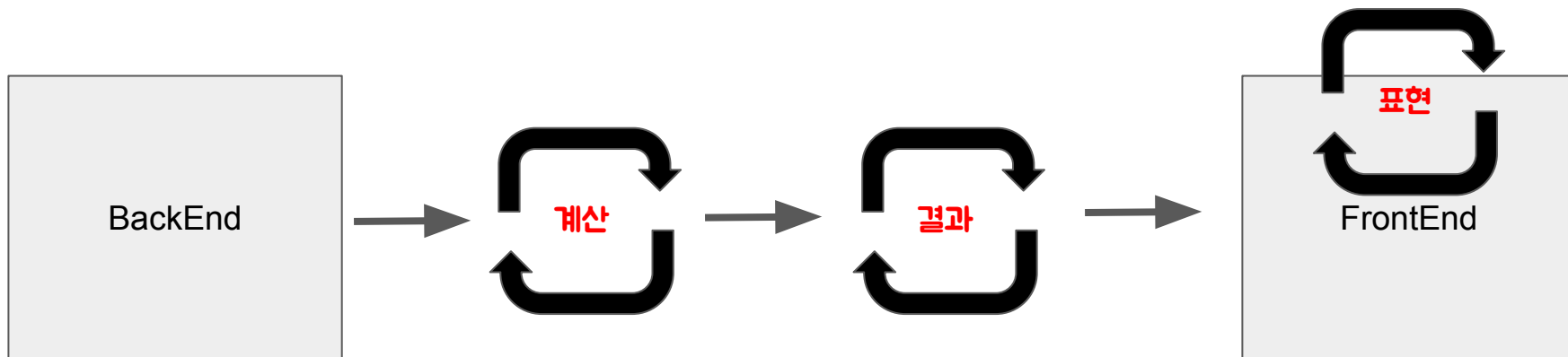
이렇게 구현하게 된다면 좌표가 빗나가도  
만들어진 도로 Vector 레이어 위에서만 그려지게 된다.

## 초기 구현의 문제점



프론트엔드에서  
단독 계산 작업을 모두 하다보니  
좌표가 많이 저장되어있을 경우  
초기 로딩시에 계산 작업이 매우 많아  
페이지 멈춤 현상 발생

## 초기 구현의 문제점 대책안

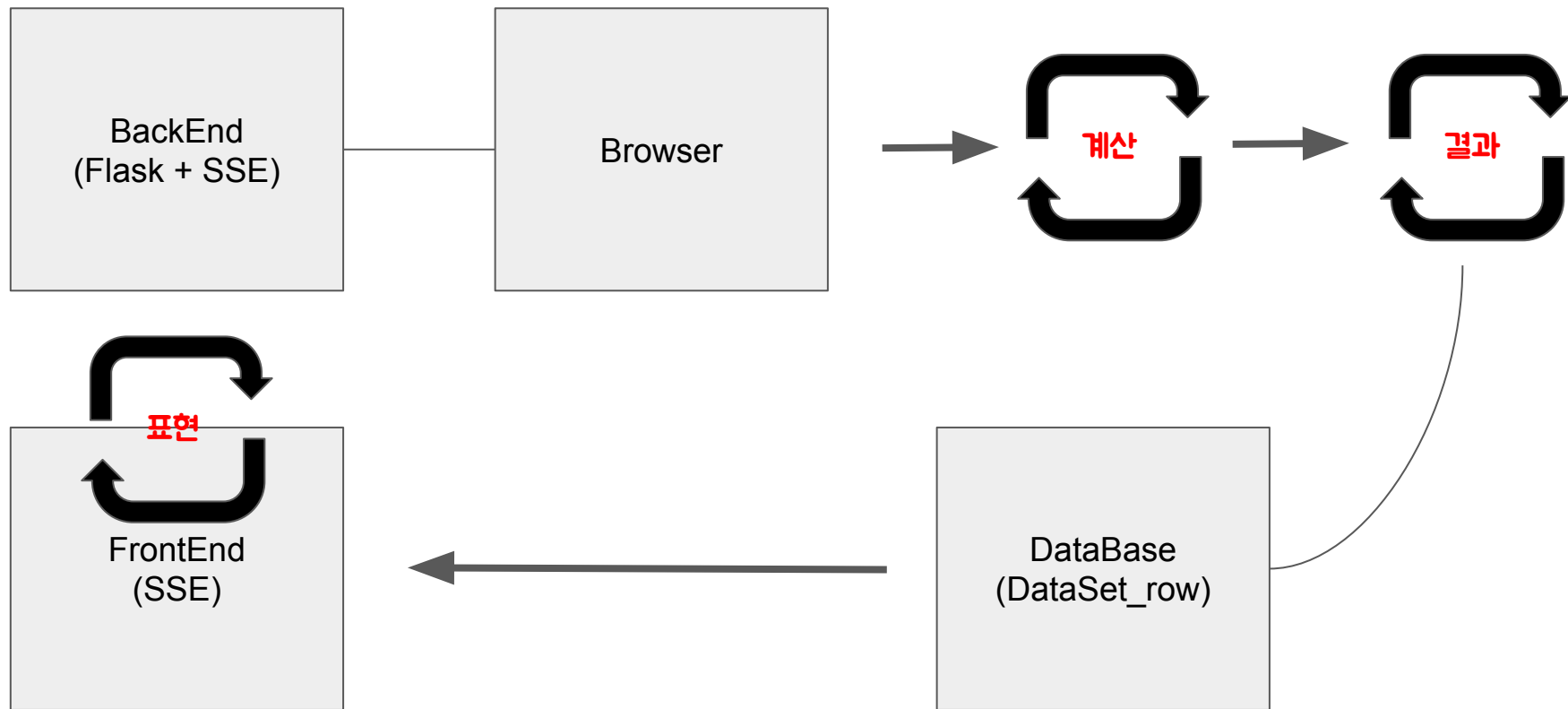


백엔드에서 미리 좌표들을 계산 후  
세그먼트 LineString들의 수선의 발들의 좌표를  
모두 DB에 저장

프론트엔드는 계산이 완료된 수선의 발 좌표만 표현하여  
도로 표현



## 최종 구현 형태

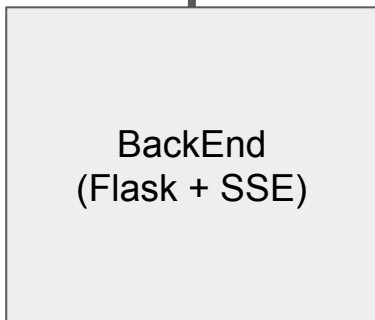


## 최종 구현의 DataBase 처리

차량이 실제 GPS 값 넣어주는 Table



실시간으로 데이터 조회

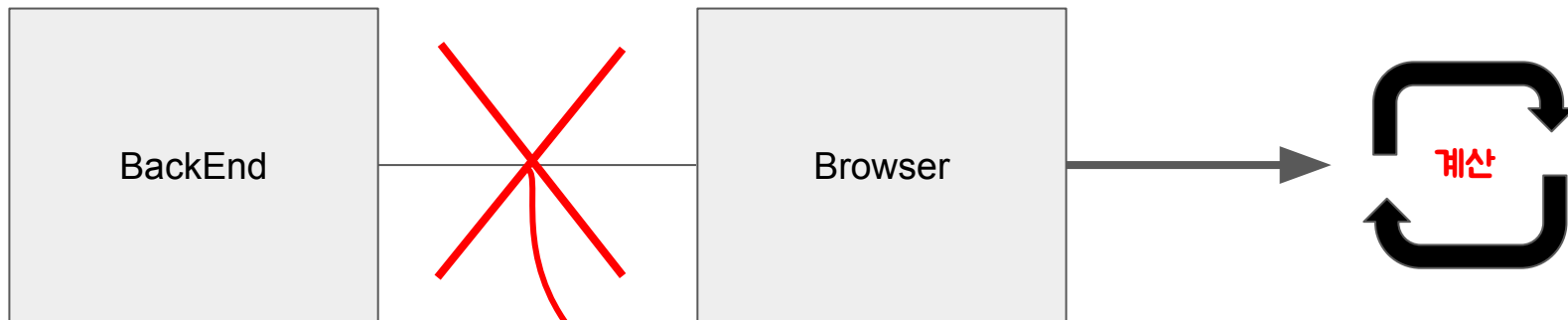


프론트엔드가 표현할 실제 값 Table



계산이 완료된 수선의 발 좌표 데이터 입력

## 대책안(최종구현)의 문제점



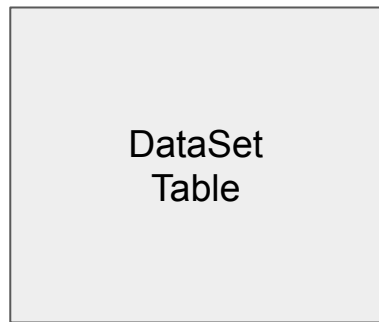
아직 백엔드에서 렌더링을 하지않고  
Openlayers 기능을 통해  
계산 할 수 있는 방법을 찾지 못하였고

현재는 꼼수로 백엔드에서  
브라우저 하나가 계산 처리를 하도록 구현한 상태

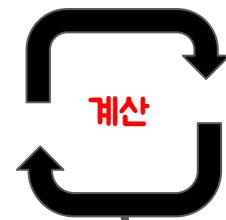
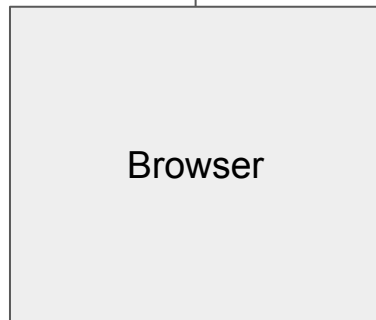
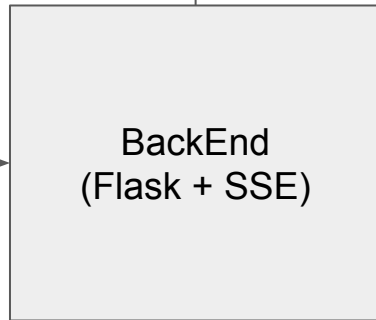
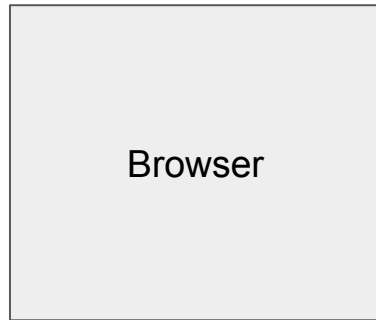
현 구현상태는 브라우저가 꺼지면  
좌표 계산이 동작하지 않아 문제 발생

추후 개선 문제점

차량이 실제 GPS 값 넣어주는 Table

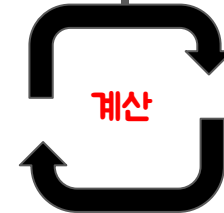


BackEnd  
(Flask + SSE)



같은 데이터 중복 저장

DataSet\_row  
Table



차량이 실제 GPS 값 넣어주는 Table



서버 사이드 세션을 이용하여  
서버에서 단 하나의 브라우저 손님만  
허용되도록 구현 예정