

# Large-Scale and Multi-Structured Databases

## *Project Design*

## *LondonSafeTravel*

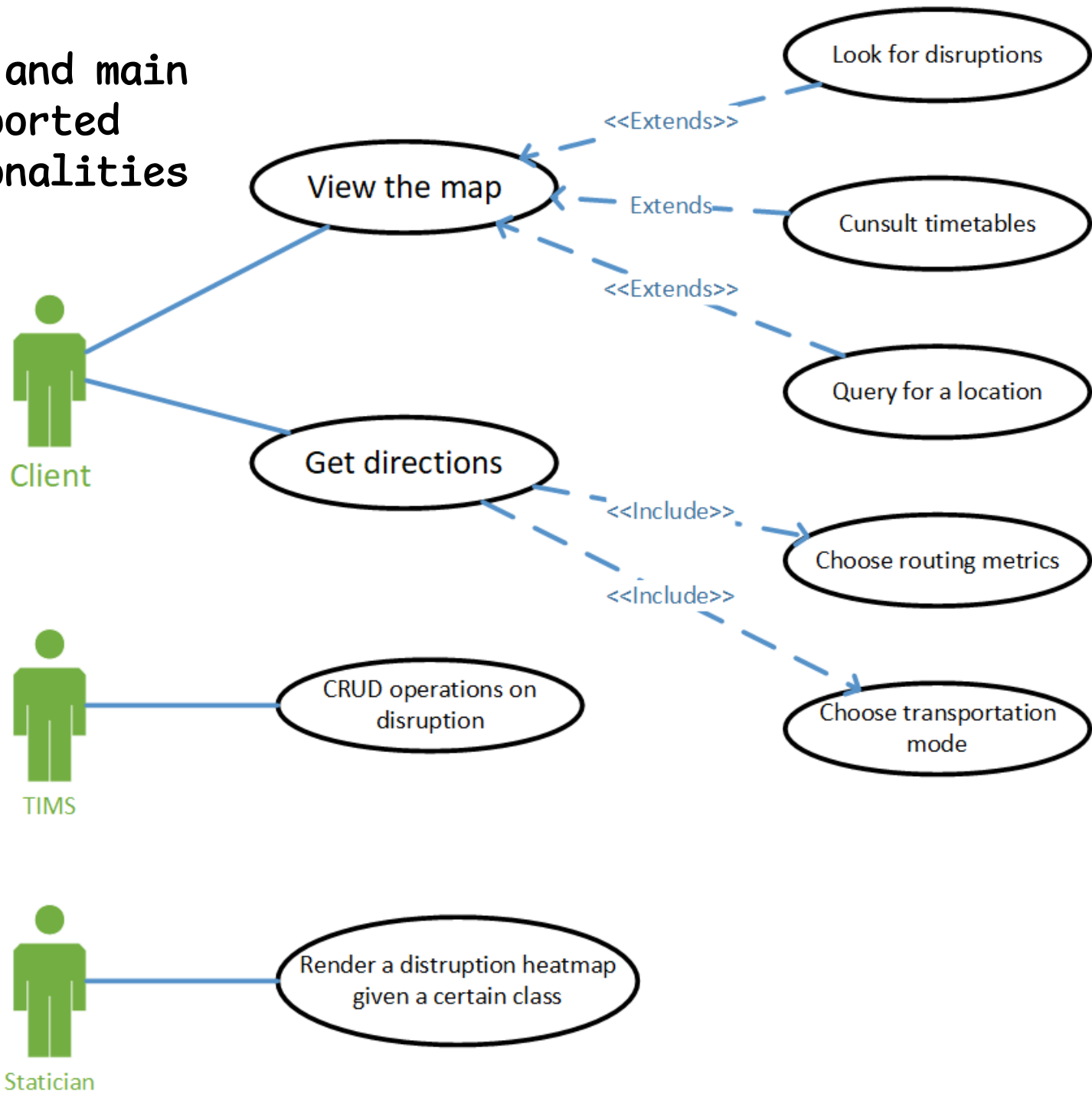
Ricky Marinsalda  
Federico Frati  
Dario Pagani

# Application Highlights

Our on-line service will allow its users to explore, search and visualize a geographical **map** of London and its public transport **network**.

- ➔ **Search** for places such as addresses and *POIs*
- ➔ **View** transit's timetables
- ➔ **Compute** a travel route between two points
  - On foot
  - By car
  - By public transport
- ➔ View and **avoid** disruptions on the network
  - Vehicles' accidents
  - Road works and other road closures

# Actors and main supported functionalities



# Dataset Description

## Sources:

*OpenStreetMap* for the road network

*Transport for London* for the timetables and routes

*Transport for London's TIMS API* for disruptions

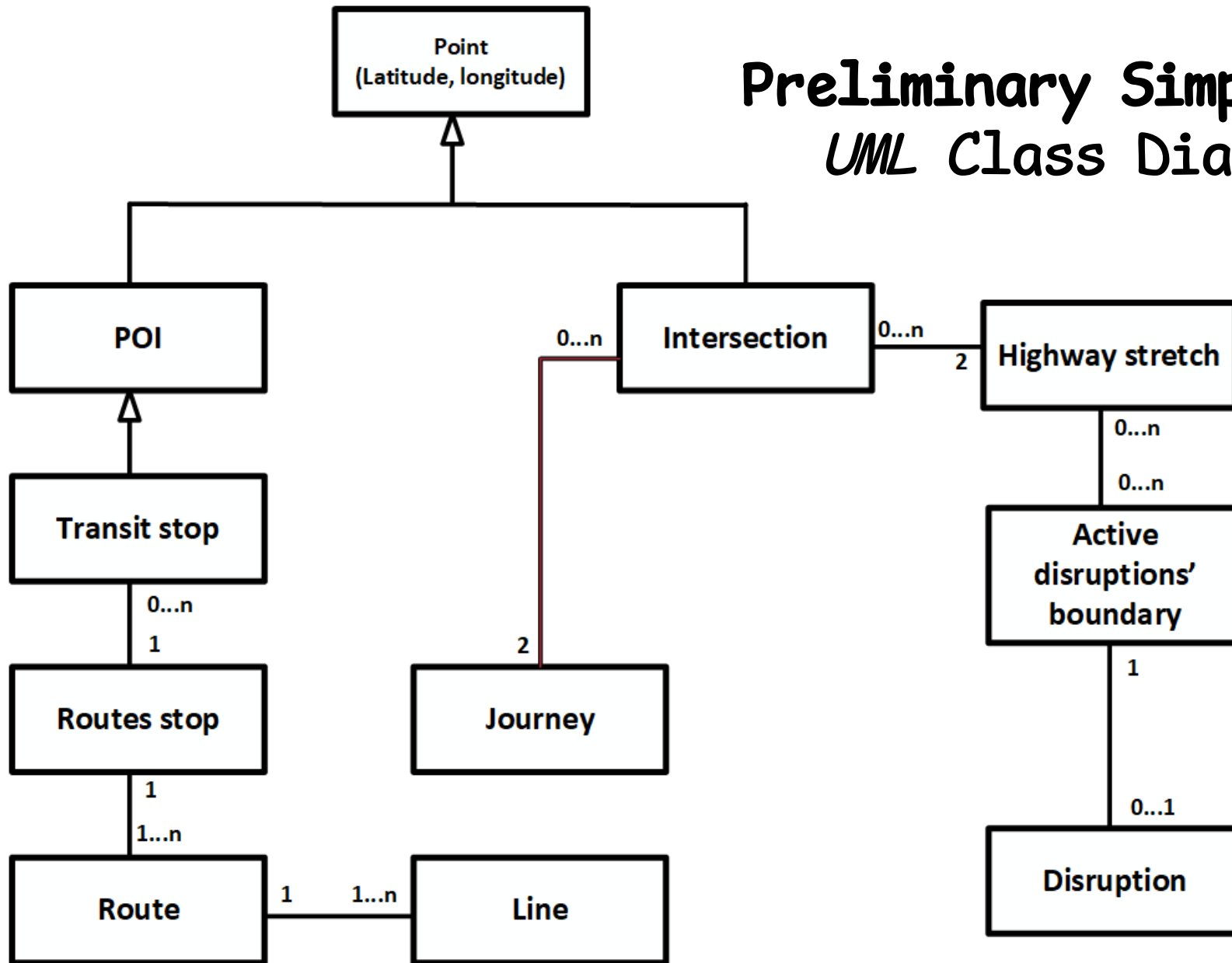
**Description:** A road and transit's network

**Volume:** circa 200MiB

**Variety:** CSVs files, JSONs files and OSM.PBFs files

**Velocity/Variability:** *TIMS* data are like a **live feed**, older data are moved into an archive for future **analytics**

# Preliminary Simplified UML Class Diagram



# Requirements and Entities handled by Document DB

## Entities

- *POIs* (transit stops, landmarks etc..)
- All entities related to public transportation
- Disruptions and their boundaries

## Queries

- List *POIs* in a given area [*map view*]
- Routing between two public transport's stops [*simplified*]
- Build a heatmap of a certain class of disruption
  - *for broken traffic lights, vehicles collisions, burst water pipes, etc..*
- Find the most common disruption in a given area

# Requirements and Entities handled by Key-Value DB

COMPUTE\_ROUTE(  
    SOURCE: WGS84 coords,  
    DESTINATION: WGS84 coords):

IF <SOURCE, DESTINATION> IN CACHE:  
    INCREASE HITS  
    RESET TTL  
    RETURN *CACHED ROUTE*

ELSE  
    ROUTE ← COMPUTE\_ROUTE(...)  
    CACHE ROUTE FOR N MINUTES  
    RETURN ROUTE

Journey

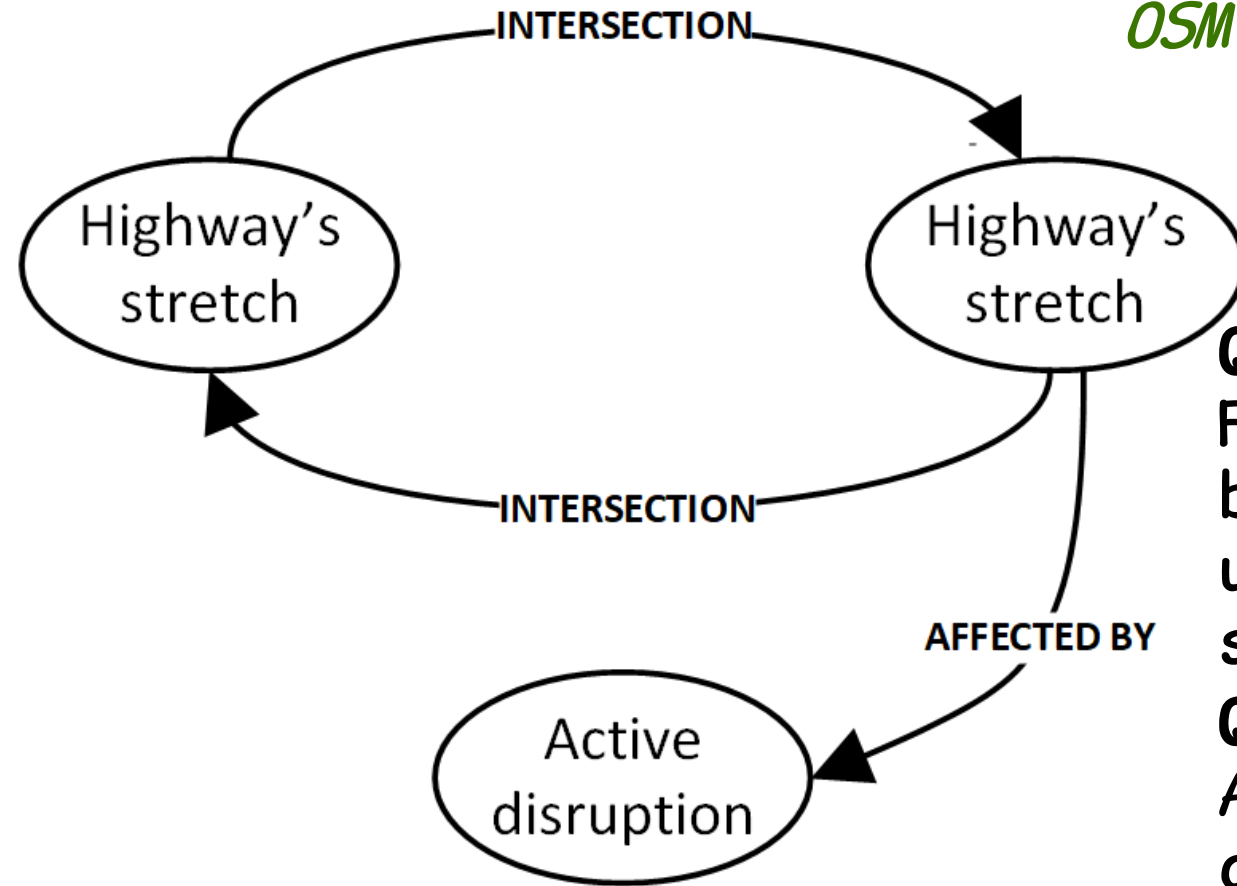
**KEY:** <SOURCE, DESTINATION>  
**VALUE:** <ROUTE, HITS>

**Another query:** find the most common recent destinations

# Requirements and Entities handled by Graph DB

A possible schema:

OSM vertexes → edges  
OSM edges → vertexes



**Query:**

Find a route between two points using an algorithm such as **A\*** or **ARA\***

**Query:**

All elements in an area [map view]

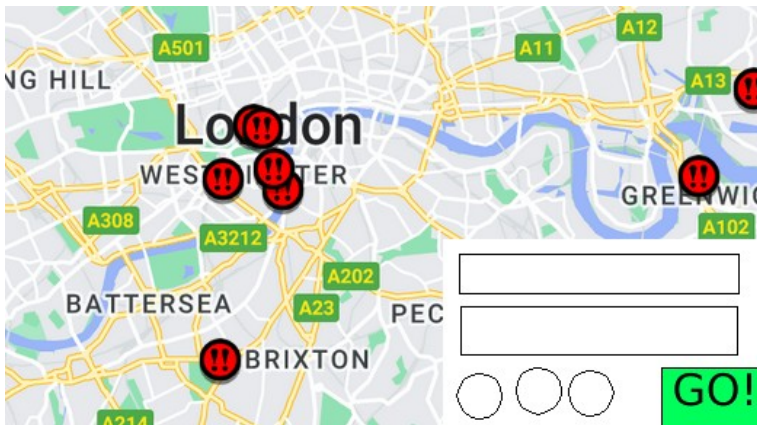


# Software Architecture Preliminary Idea

## DBMSes

- *MongoDB* for transit and TIMS' data
- *Neo4j* for road network's data
- *Redis* for routing paths' cache

## Mockup



## Technologies

One of the following:  
(yet to decide)

- Java & TomCat (web client)
- Python & Flask (web client)
- Java Swing (Java client)