**PhD Workshop ‘24**

# Data Management for mobile applications dependent on geo-located data

Luís M. Silva

02 Out 2024

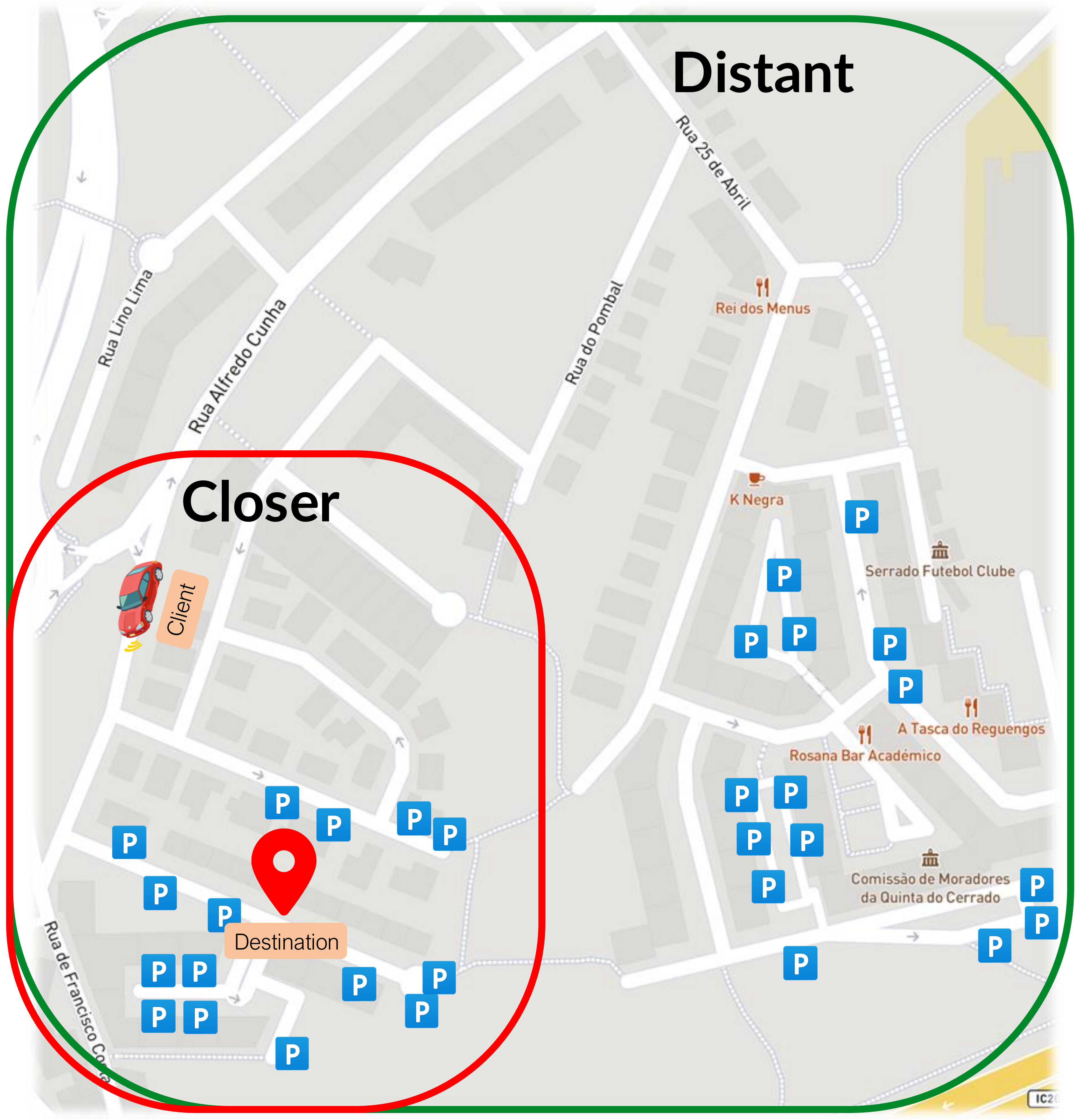
# Motivation

- Mobile devices have become the preferred platform for deploying new applications.
- Some applications need to share and manipulate location-dependent data.
- Examples of those applications are vehicular applications or mobile multiplayer games.

# Motivation

- The user's interest in data directly correlates to its distance to that data.

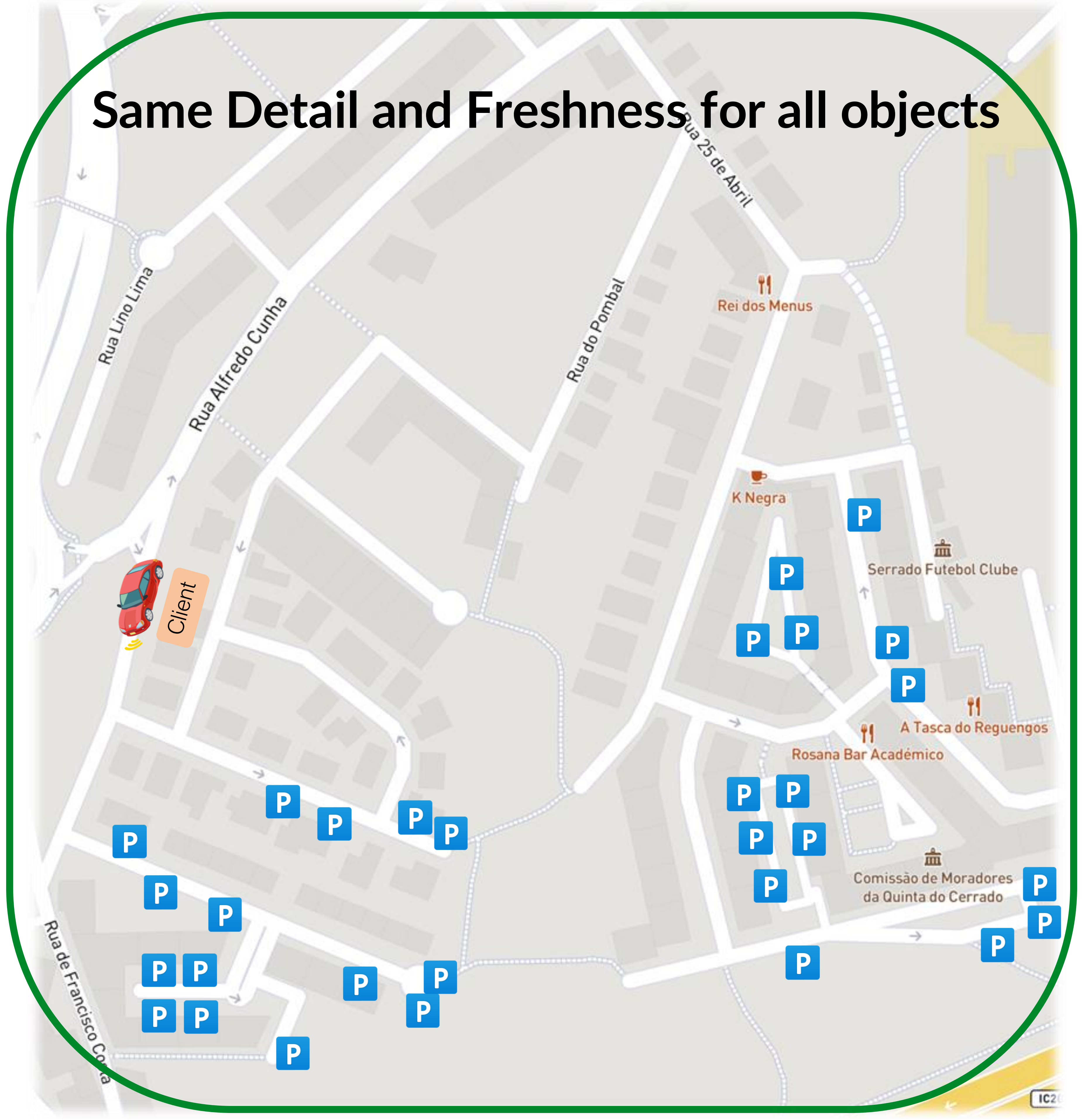
- Storage maintains a single data model, and consistency is uniform for all data.



## Motivation

- The user's interest in data directly correlates to its distance to that data.

- Storage maintains a single data model, and consistency is uniform for all data.

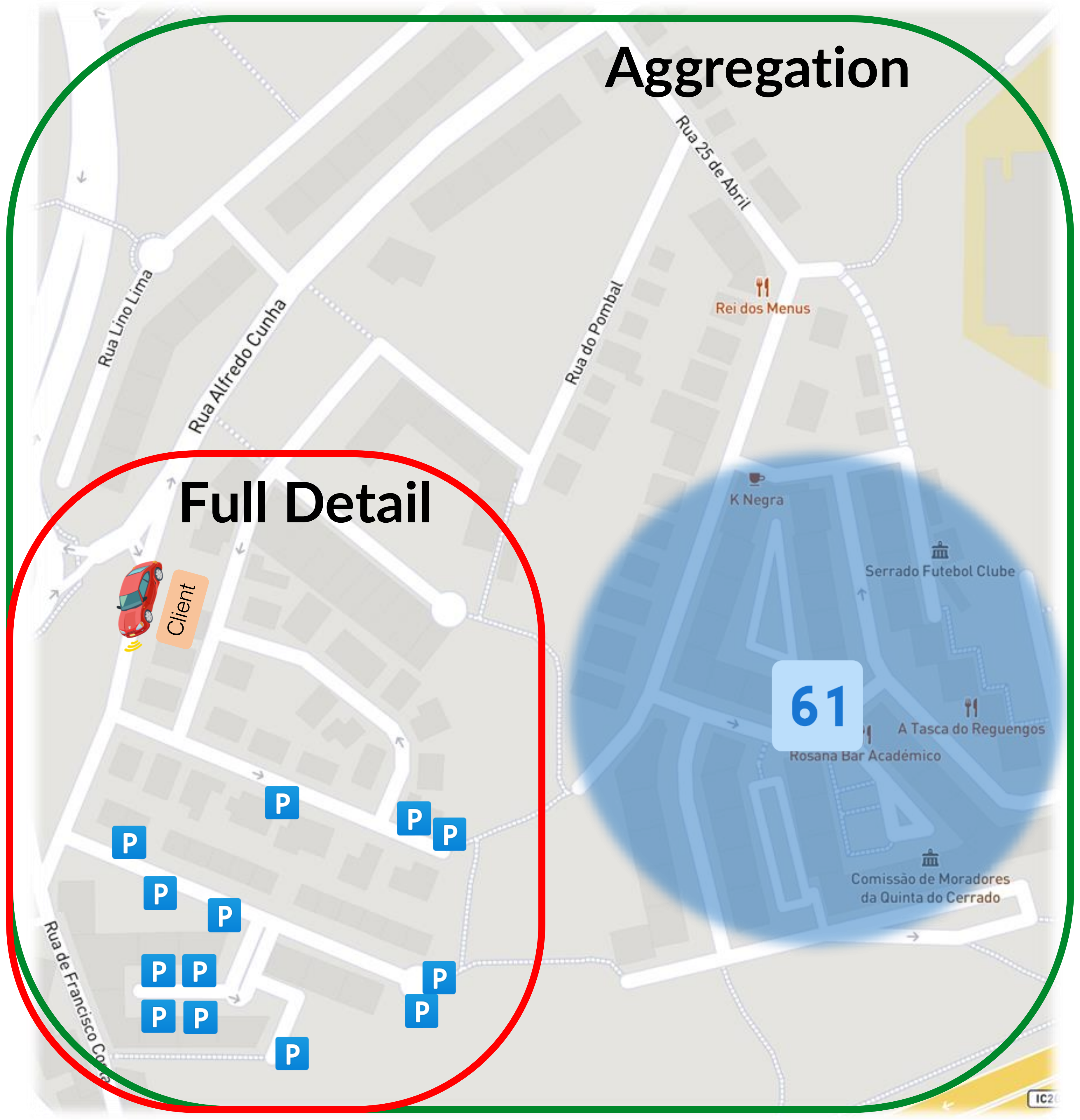


## The Problem

- No data management system supports both dynamic data models and tunable consistency constraints on a mobile environment for location-dependent data.

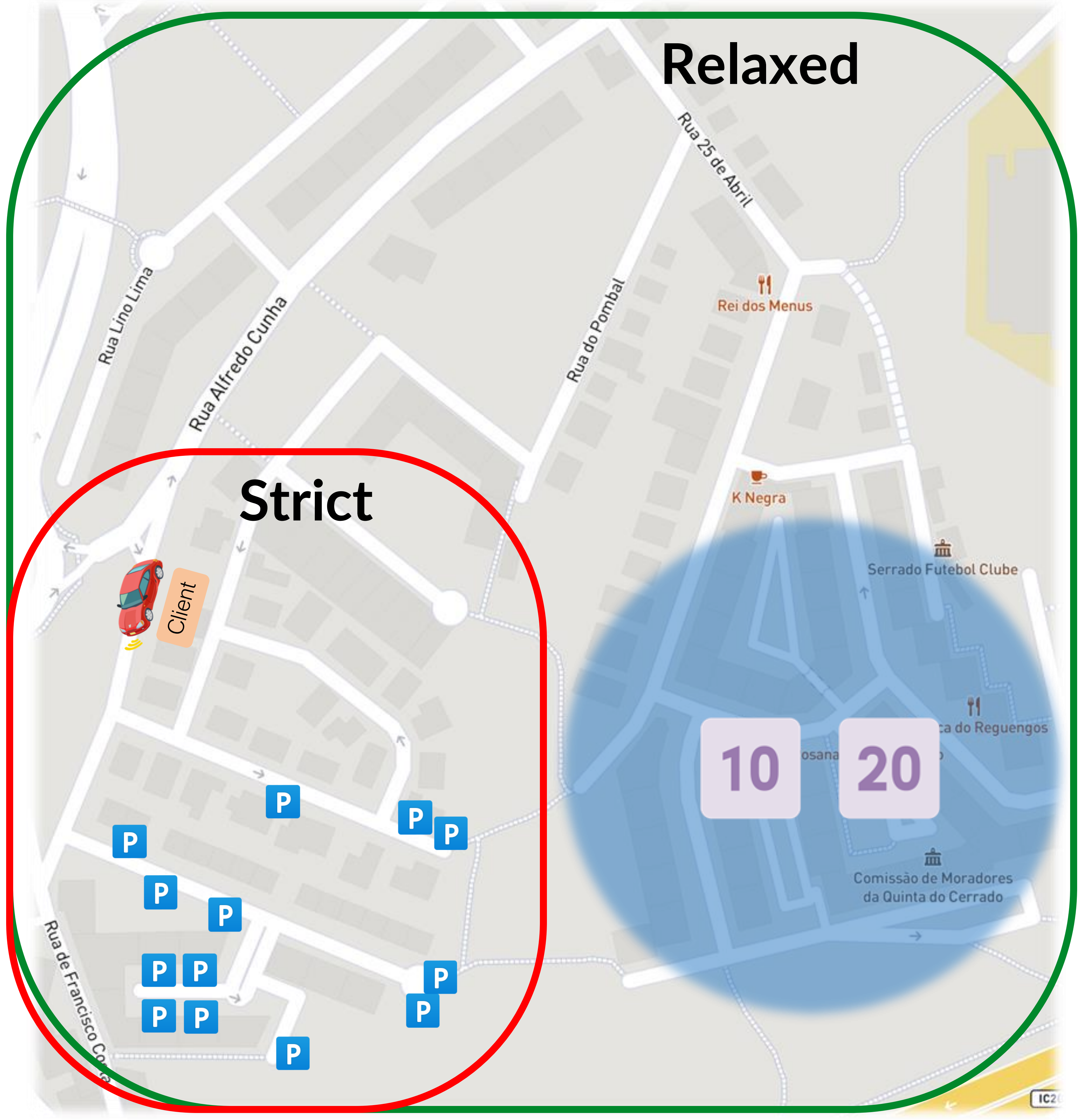
# Desired properties

- **Adaptive Data Detail Exposure**
- Exploiting all the system components' locations for performance gains.
- As the distance between a user and data increases, the detail is intended to diminish gradually and gracefully.
- It is a requisite to provide a flexible non-uniform data model.



# Desired properties

- **Dynamic Inconsistency Management**
- The data model allows consistency models with fine-grained control
- The system can adjust the frequency of updates to manage the degree of inconsistency.



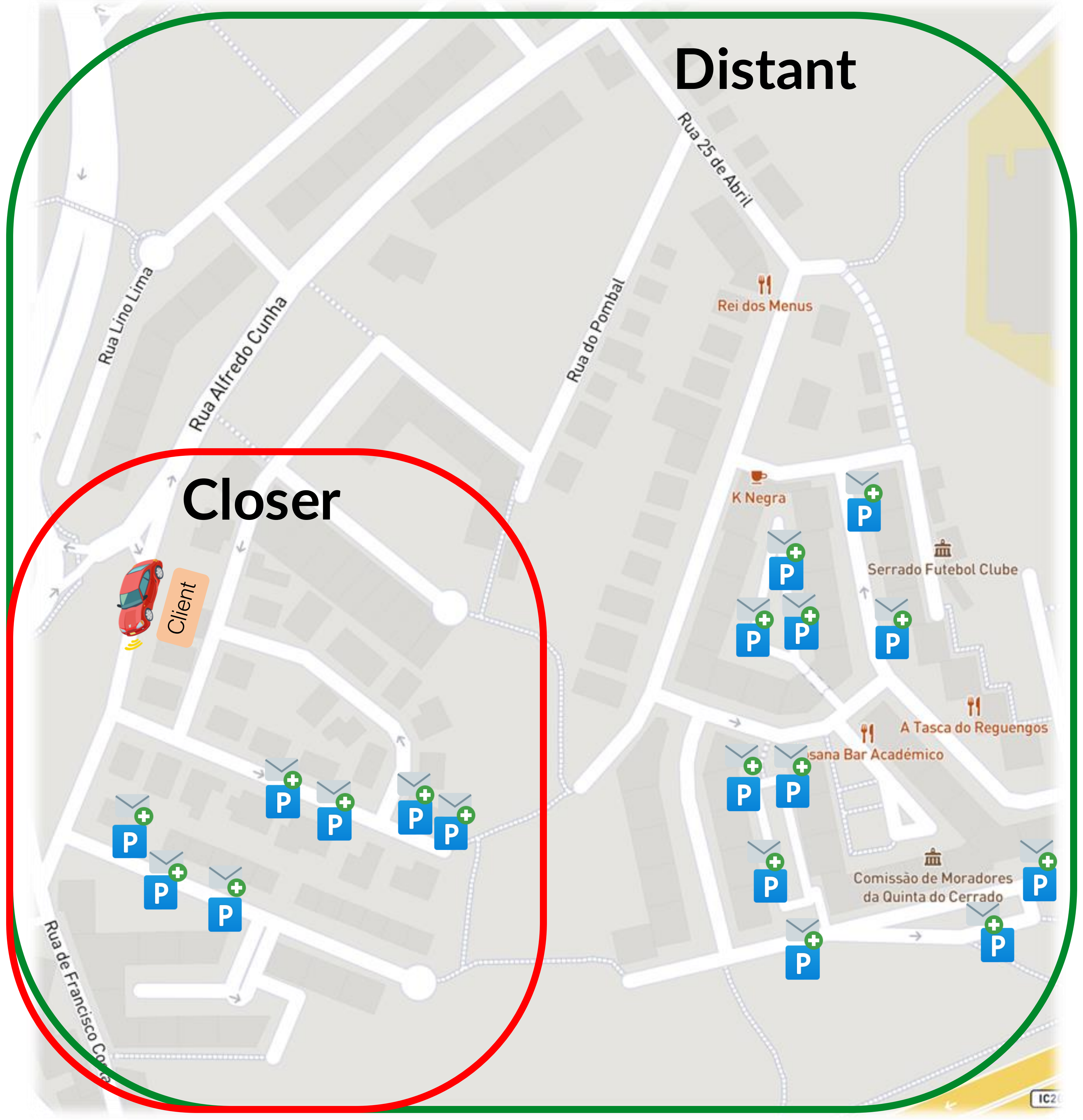
## Summary

- Introduced FocusDB: a system for geo-located data in mobile environments
- Data model that accounts for objects and client locations
- Levels of detail based on client location
- Adaptive consistency guarantees based on client interest
- Drawbacks: increased metadata traffic and cloud storage usage

Thank You!

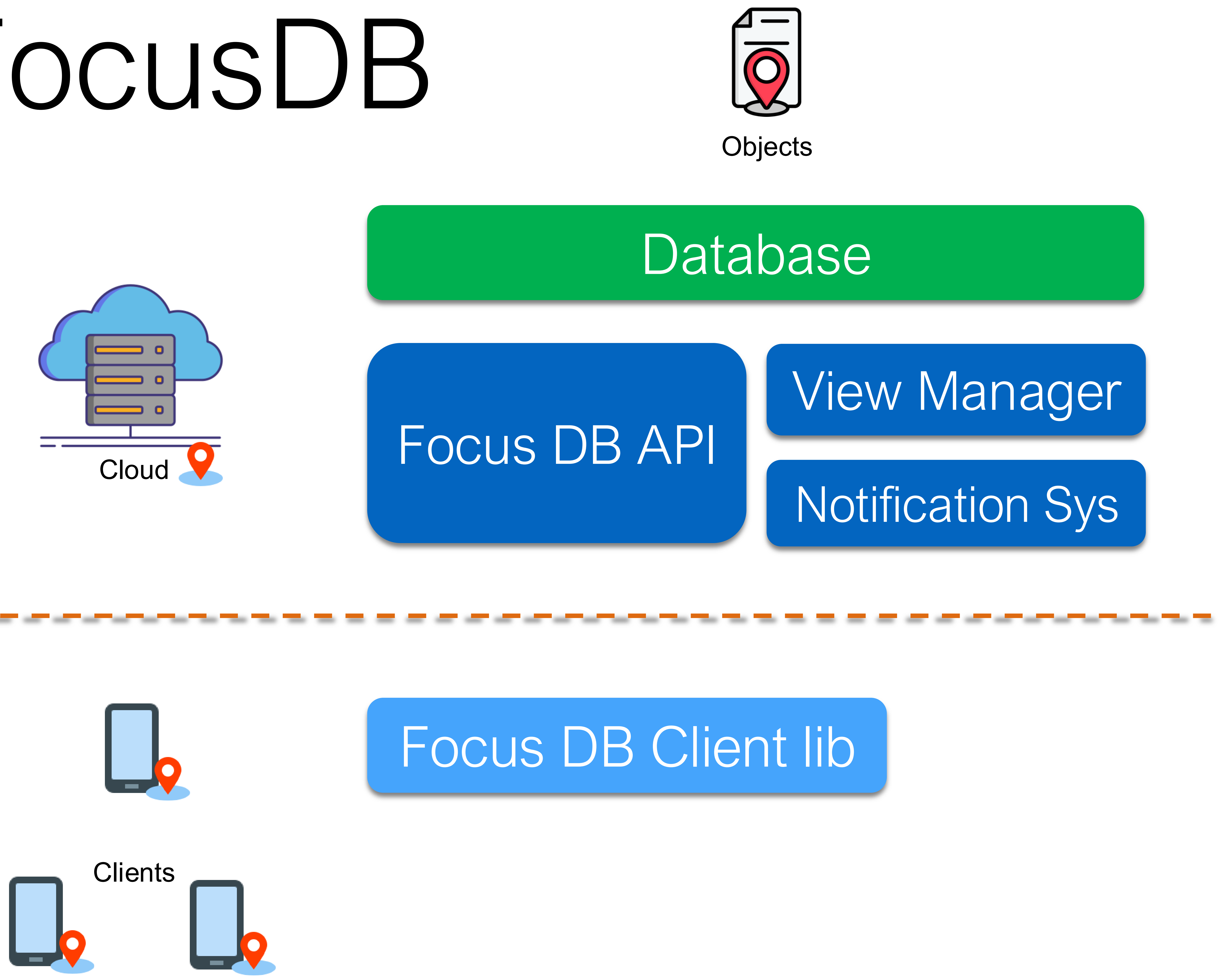
# Desired properties

- **Proximity-based Interest** and **Explicit User Preferences**
- The system is not aware of the user's intentions.
- We need a mechanism that informs the system of the user's interest.



# FocusDB

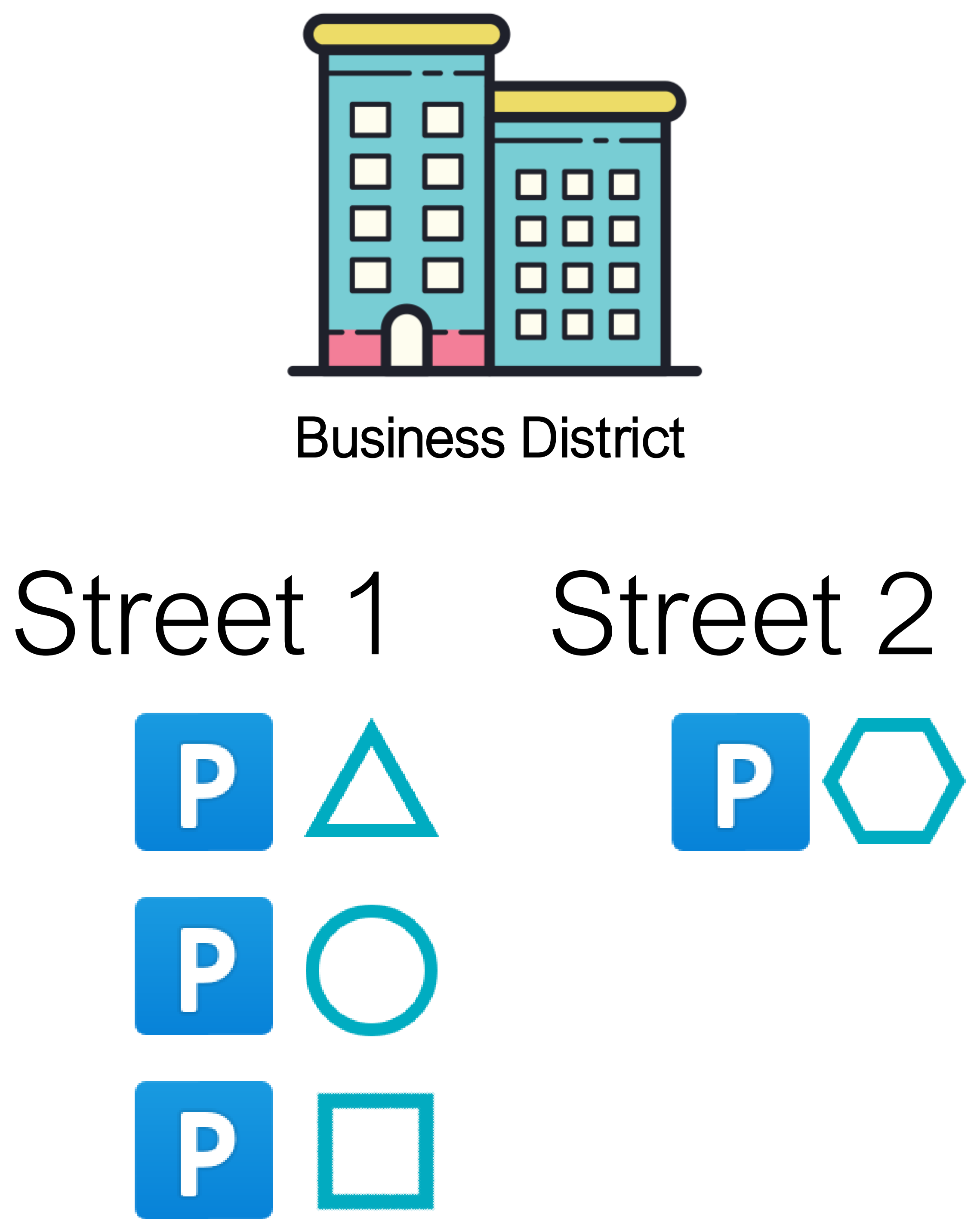
- **Cloud Side** responsible for allowing data access and manipulation methods
- **Client Side** interacts with the cloud using a library



# Data Model

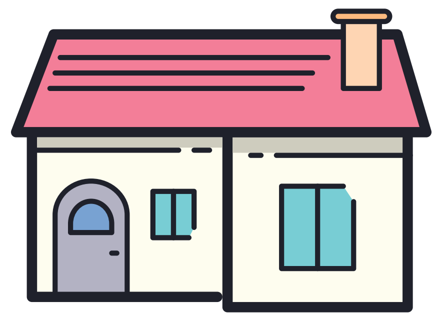
| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |

(Street_1, 3)
(Street_2, 1)

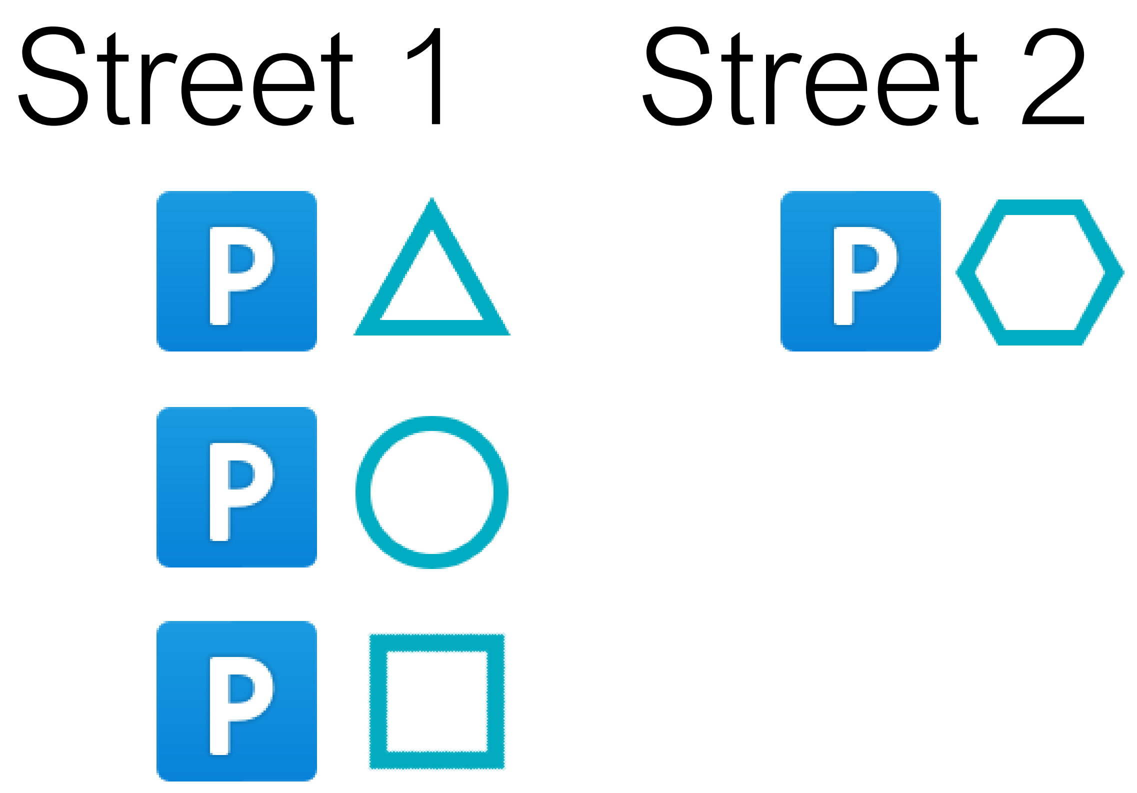


# Example

| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |

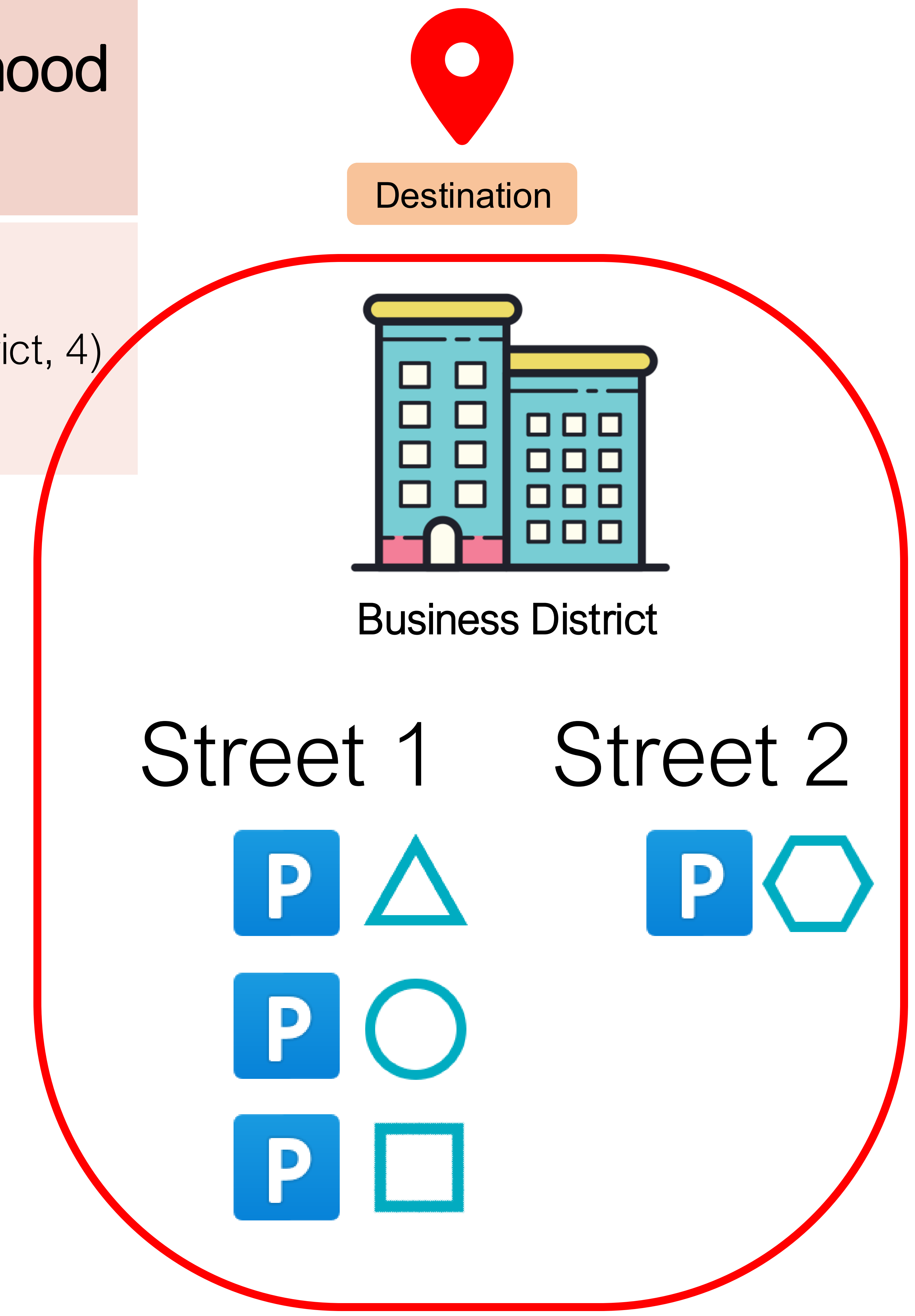


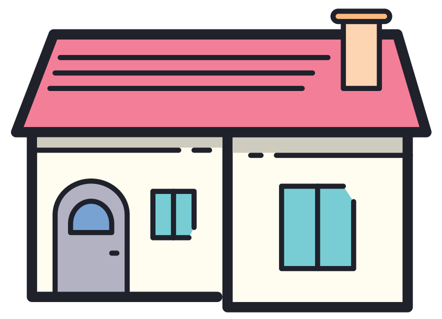
Client



## Example

| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |





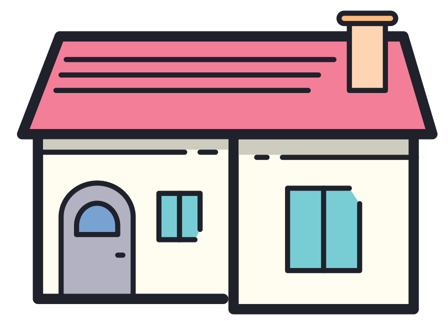
Client

| Spots | Streets View | Neighborhood View |
|---|---|---|
| | | |

# Example

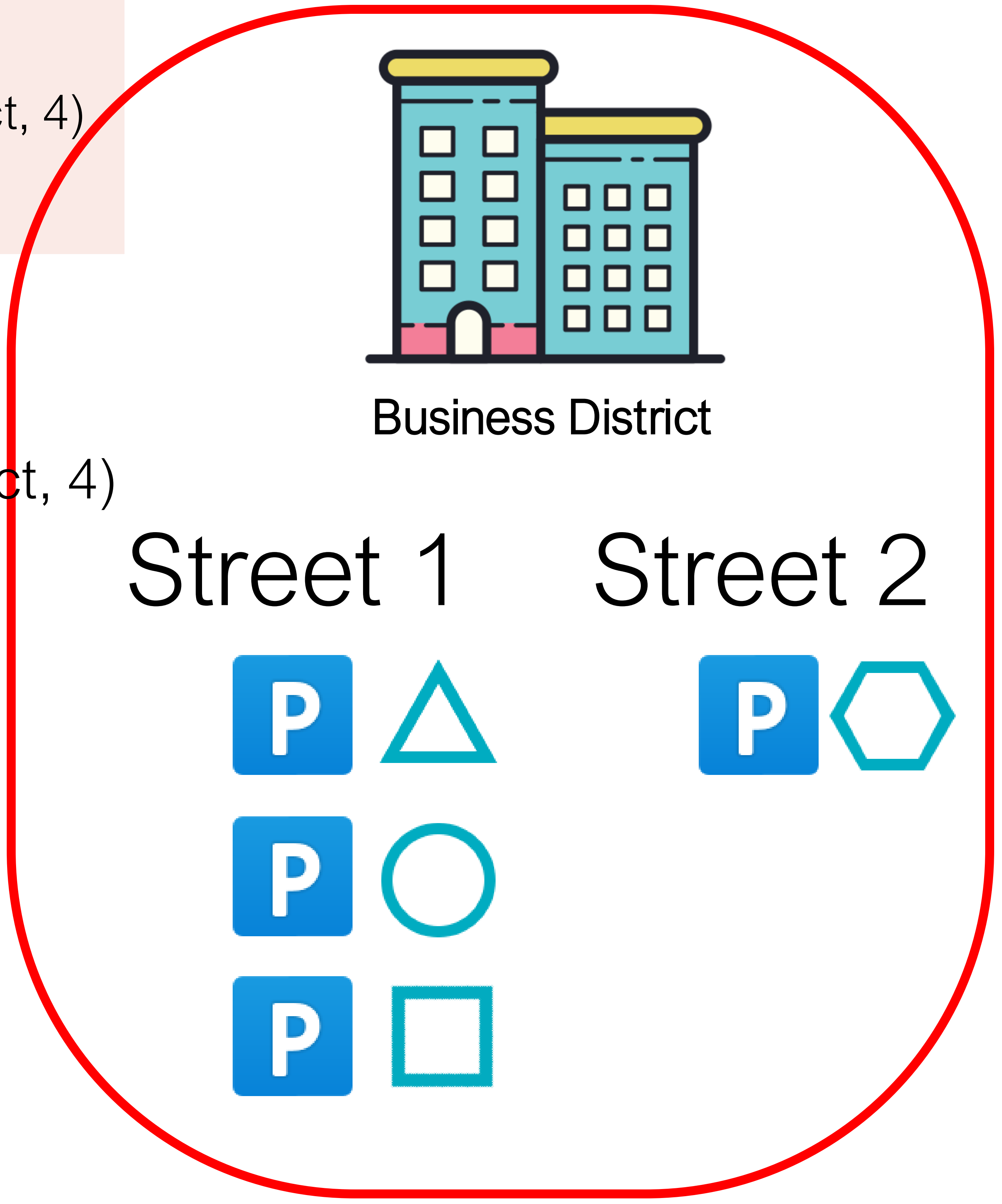
| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |

(Business District, 4)



Client

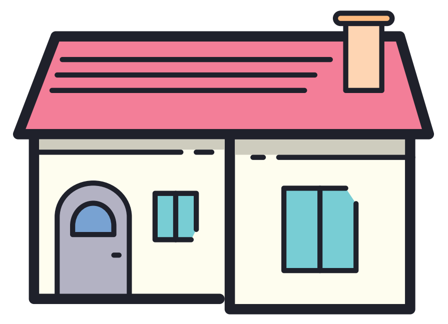
| Spots | Streets View | Neighborhood View |
|---|---|---|
| | | |



## Example

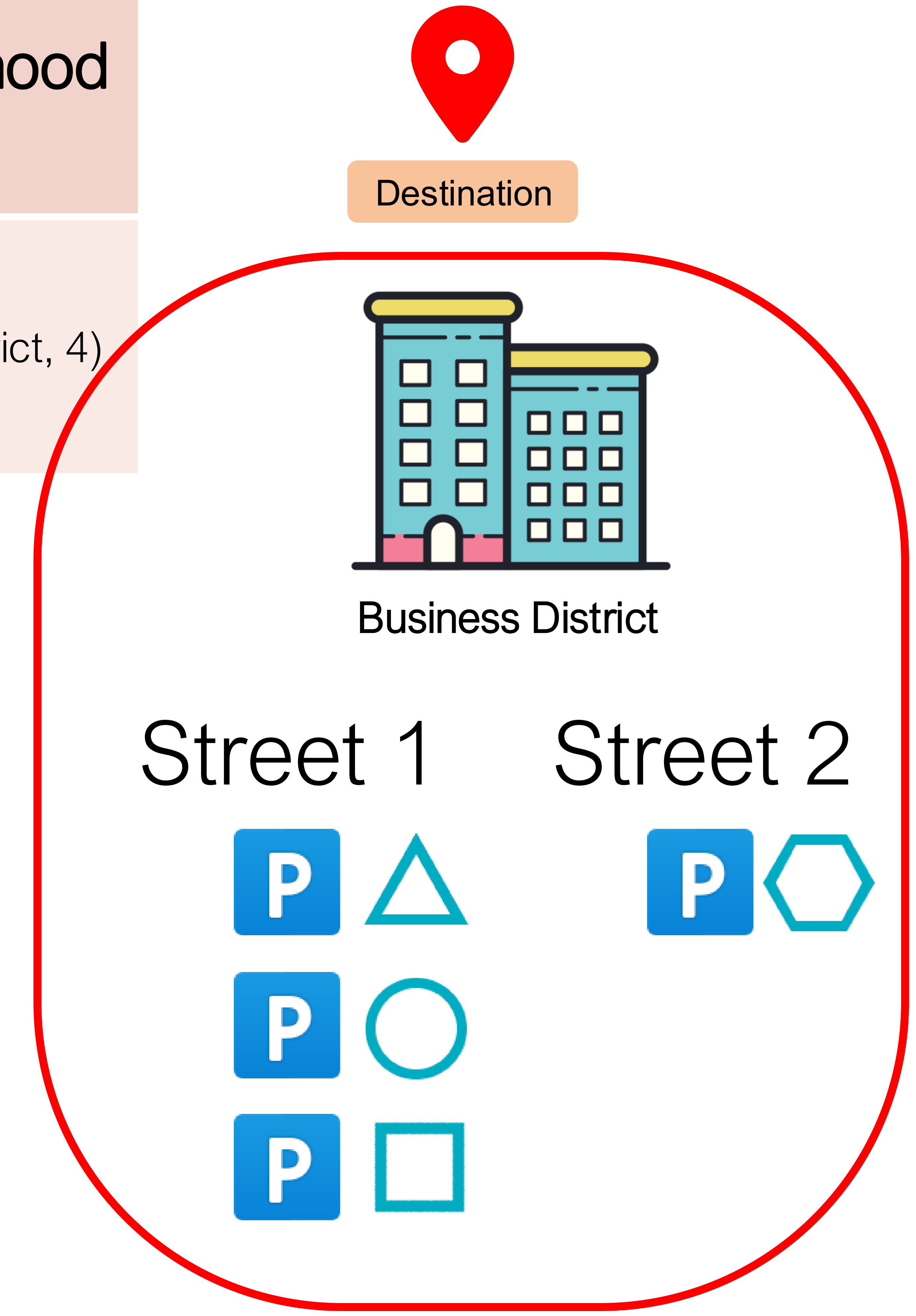
| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |

(Street 1, 3)
(Street 2, 1)



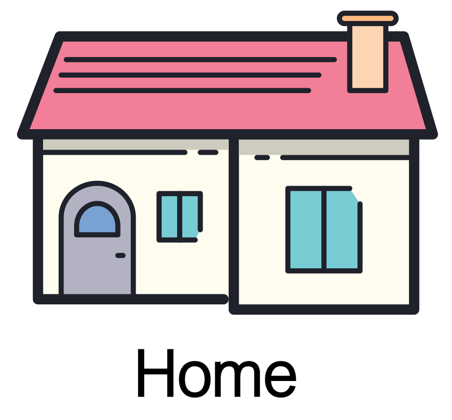
Client

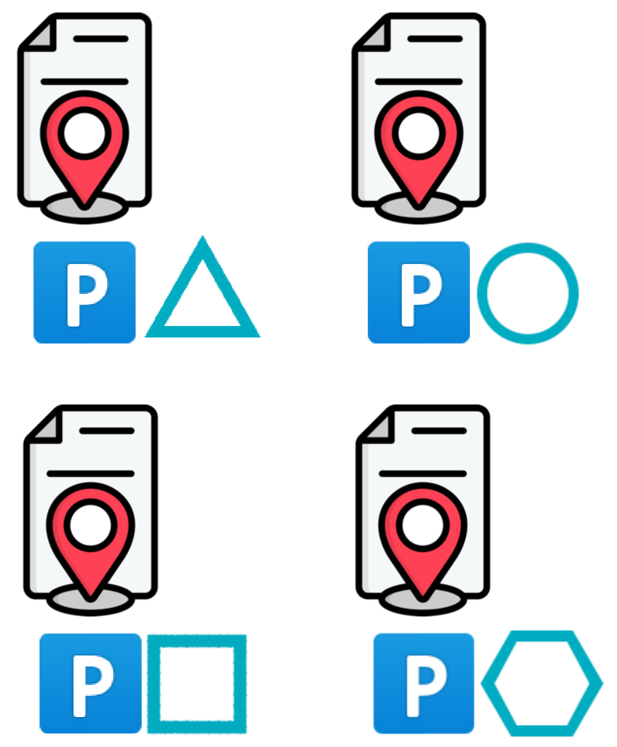
| Spots | Streets View | Neighborhood View |
|---|---|---|
| | | (Business District, 4) |

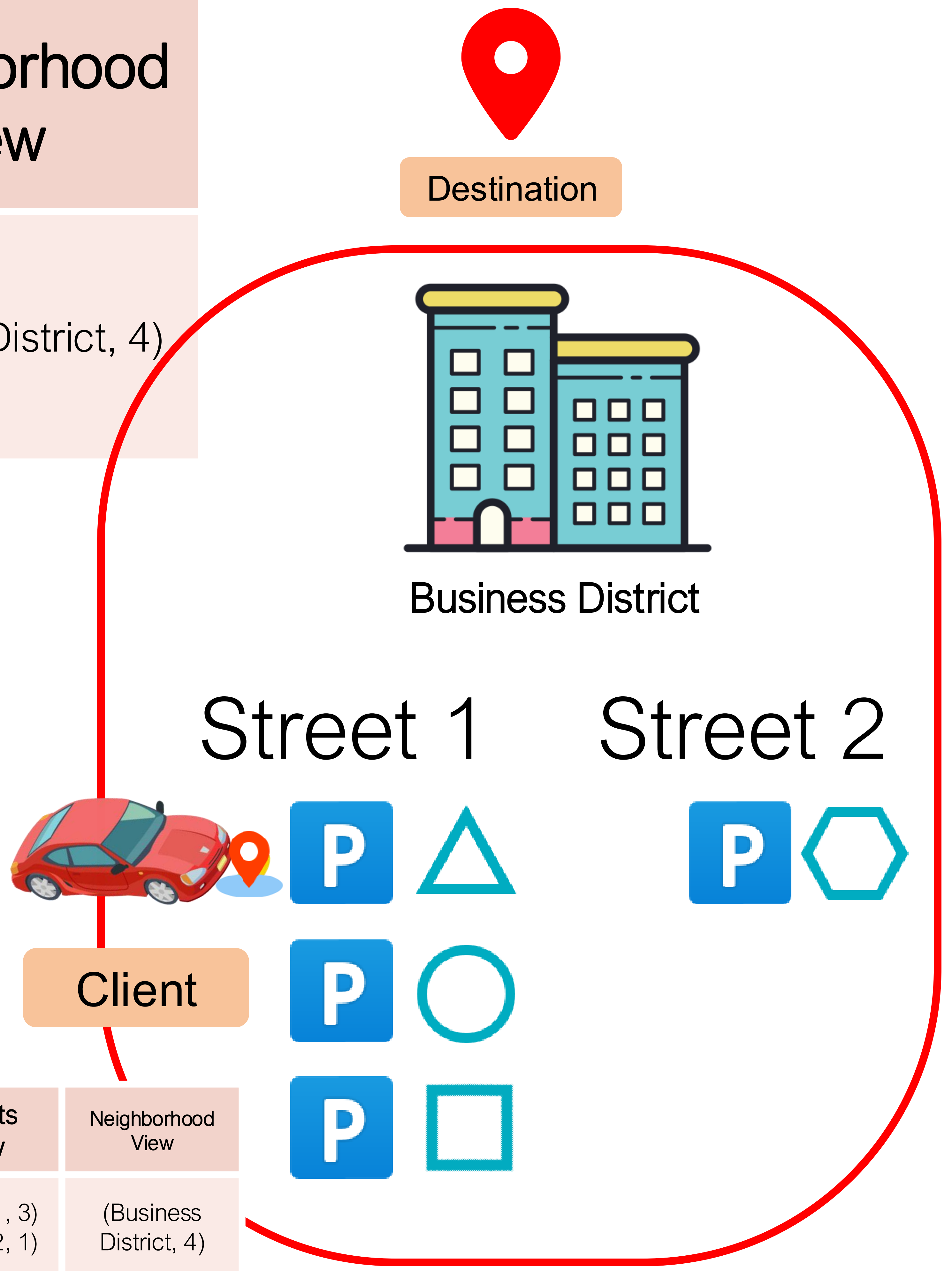


# Example

| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |







| Spots | Streets View | Neighborhood View |
|---|---|---|
| | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |

# Data Freshness

| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |

Street 1

P △

P ○

P □

Street 2

P ⬡

Client

| Neighborhood View |
|---|
| (Business District, 4) |

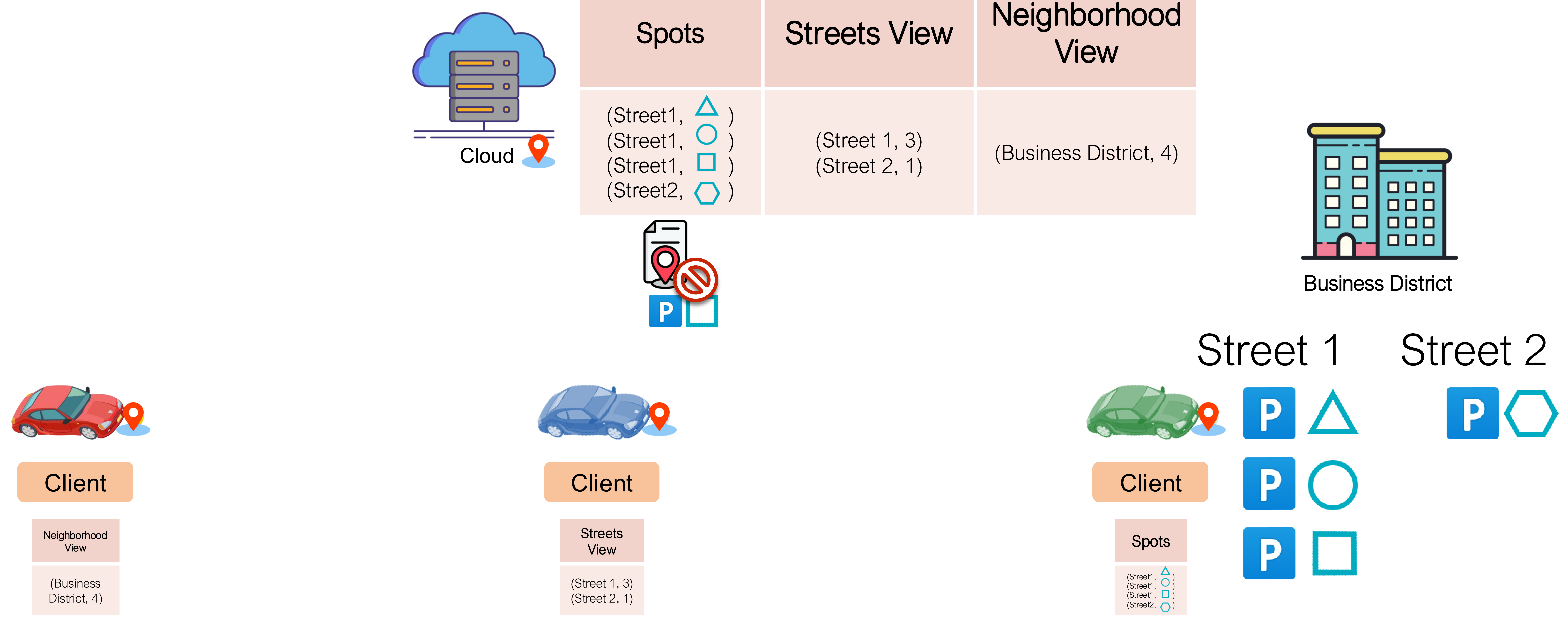
Client

| Streets View |
|---|
| (Street 1, 3)<br>(Street 2, 1) |

Client

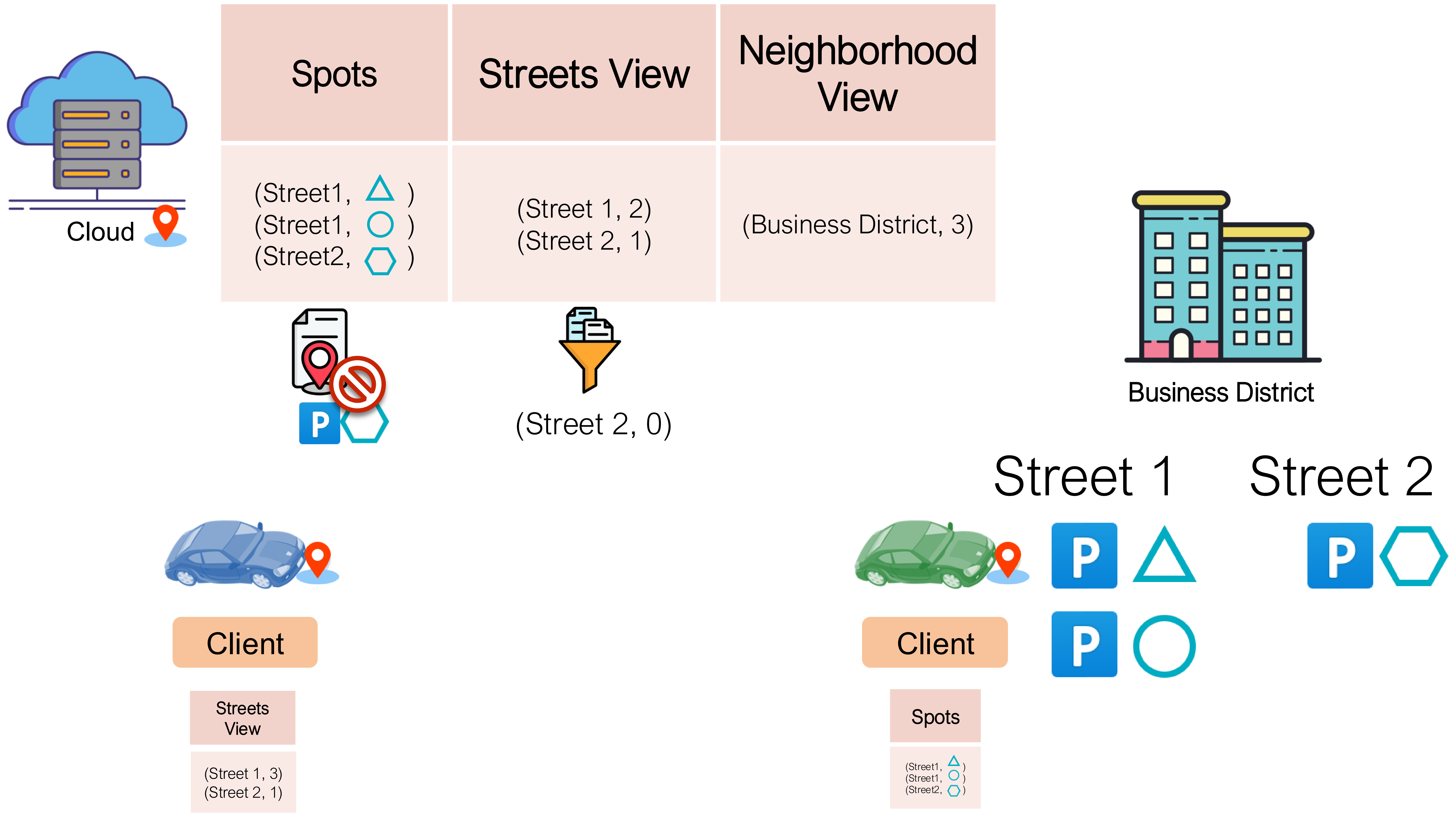
| Spots |
|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) |

# Data Freshness



| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street1, □ )<br>(Street2, ⬡ ) | (Street 1, 3)<br>(Street 2, 1) | (Business District, 4) |

# Data Freshness



| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street2, ⬡ ) | (Street 1, 2)<br>(Street 2, 1) | (Business District, 3) |

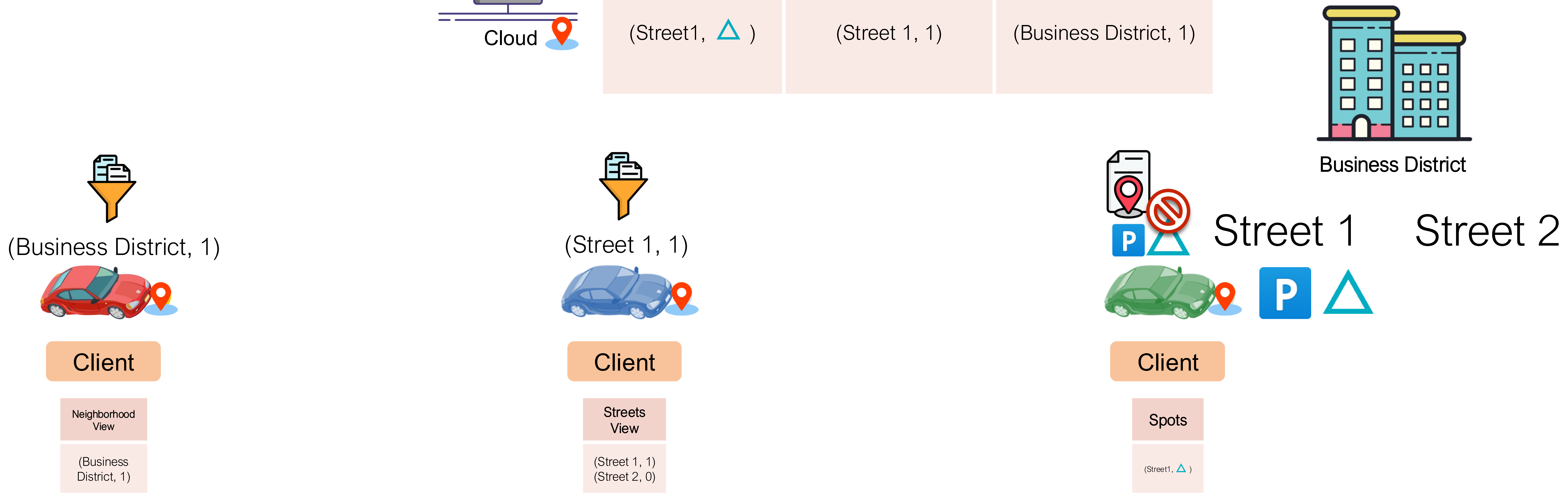
| Neighborhood View |
|---|
| (Business District, 4) |

| Streets View |
|---|
| (Street 1, 3)<br>(Street 2, 1) |

| Spots |
|---|
| (Street1, △ )<br>(Street1, ○ )<br>(Street2, ⬡ ) |

# Data Freshness

| Spots | Streets View | Neighborhood View |
|---|---|---|
| (Street1, △ ) | (Street 1, 1) | (Business District, 1) |



| Neighborhood View |
|---|
| (Business District, 1) |

| Streets View |
|---|
| (Street 1, 1) (Street 2, 0) |

| Spots |
|---|
| (Street1, △ ) |

## Evaluation

- While this work is still a work in progress, we've already conducted some preliminary experiments and evaluations.
- Does using a flexible non-uniform data model translate into benefits of performance?

## Evaluation

- Where to Park? Application
- Built a workload generator that mimics multiple mobile clients
- Prototype of FocusDB, plus two additional architectures
- Evaluated throughput and latency as a function of the number of clients and the total number of client requests

## Evaluation

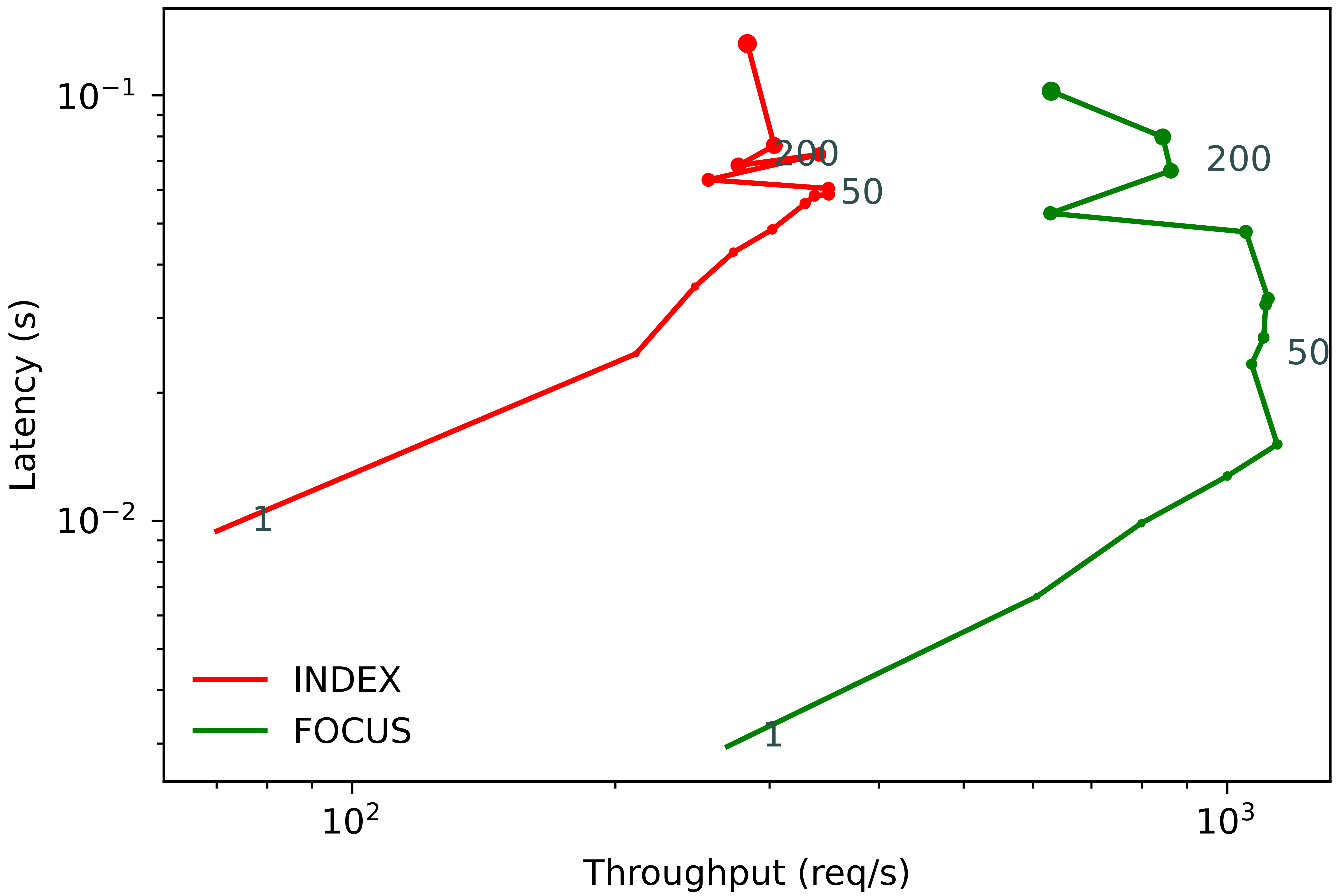
- Two datasets
  - one for client movement
  - the other for object placement (Parking Spots)
- Clients follow a path with Parking Spots in that route.

## Evaluation

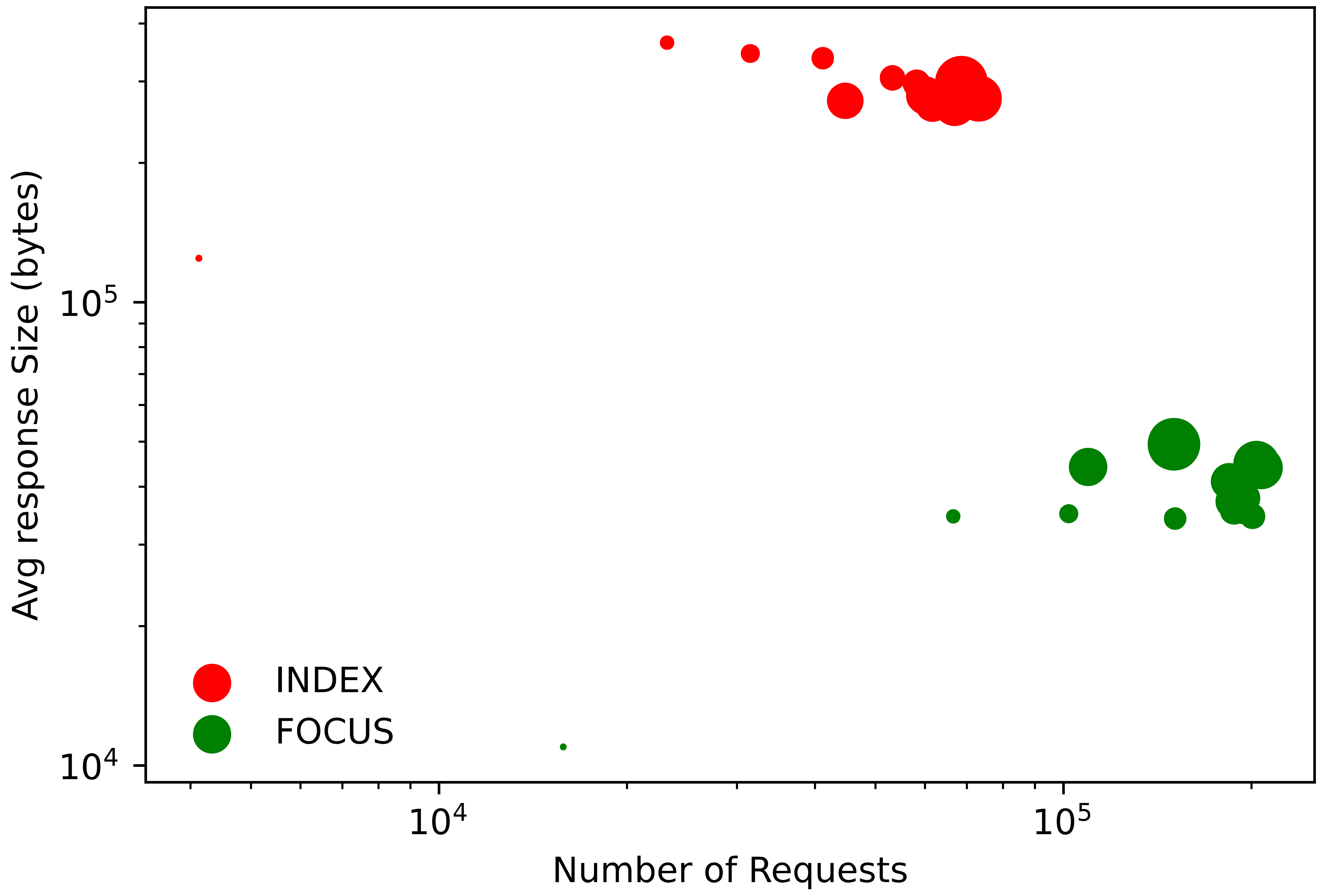
- A client initiates a request by specifying the end location of the trip, which indicates the interest set regarding parking spots
  - Client receives 3 levels of detail during the trip
    - Neighbourhood view -> Multiple street view -> Street view
- 5 Trips per Client per Test
- 80% aggregations and 20% full objects

## Evaluation

- Two different implementations
  - FocusDB prototype (FOCUS)
  - MongoDB Queries with indexes on relevant attributes (INDEX)



$10^{-1}$
$10^{-2}$
Latency (s)
$10^{2}$
$10^{3}$
Throughput (req/s)
200
50
1
200
50
1
INDEX
FOCUS


Avg response Size (bytes)
$10^5$
$10^4$
Number of Requests
$10^4$
$10^5$
INDEX
FOCUS

## Future Work

- Formalise the consistency model.
- Continue to explore location as a property for implementing efficient replication protocols.
- Expand the system model to a hierarchical cloud – edge.

# Summary

- Introduced FocusDB: a system for geo-located data in mobile environments
- Data model that accounts for objects and client locations
- Levels of detail based on client location
- Adaptive consistency guarantees based on client interest
- Drawbacks: increased metadata traffic and cloud storage usage

Thank You!