User Based Car Sharing Relocation Supporting Strategy

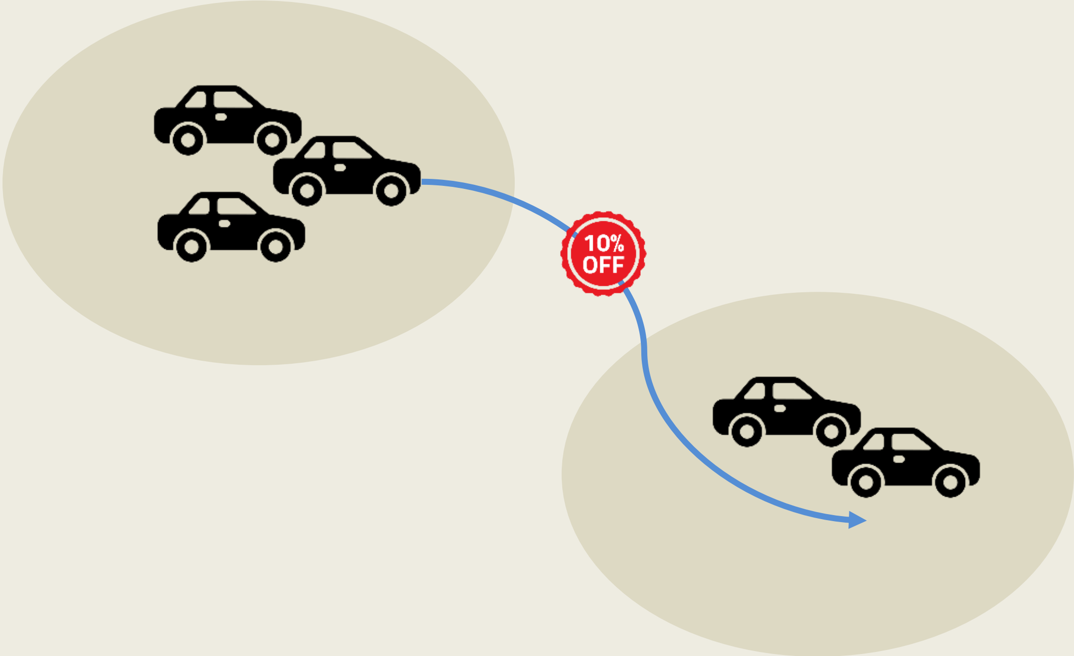
Abstract

- In a free floating car sharing system, the vehicle disposition may become imbalance during opertaion time.
- This could leads to shortages in some areas and surpluses in others. This not only increases the cost of dispatching personnel but also reduces overall operational efficiency. To address this issue, an innovative vehicle dispatch algorithm has been developed that leverages user incentives.

User based Relocation

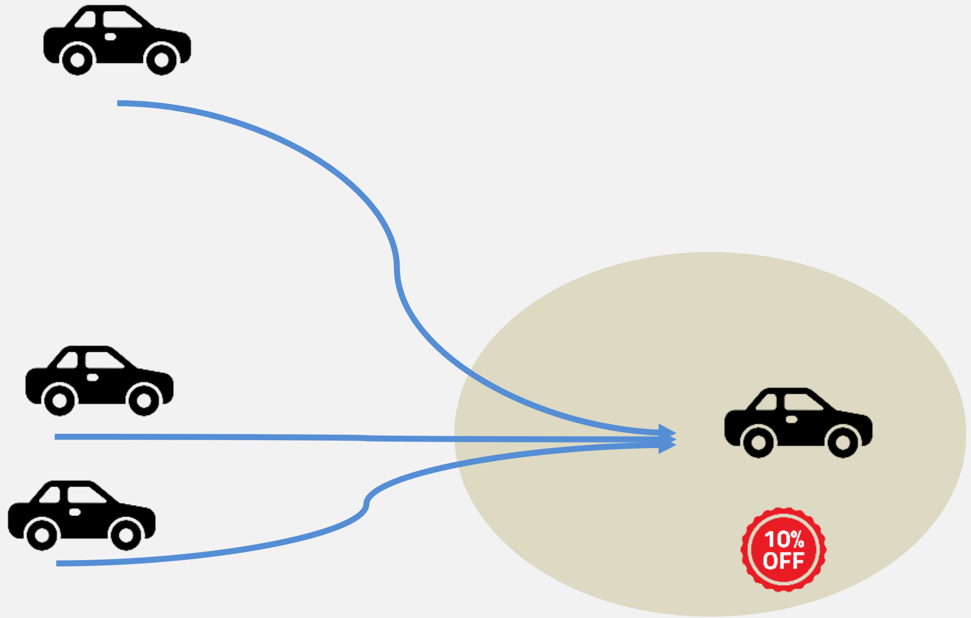
Complete rebalancing

- Move cars from A to B



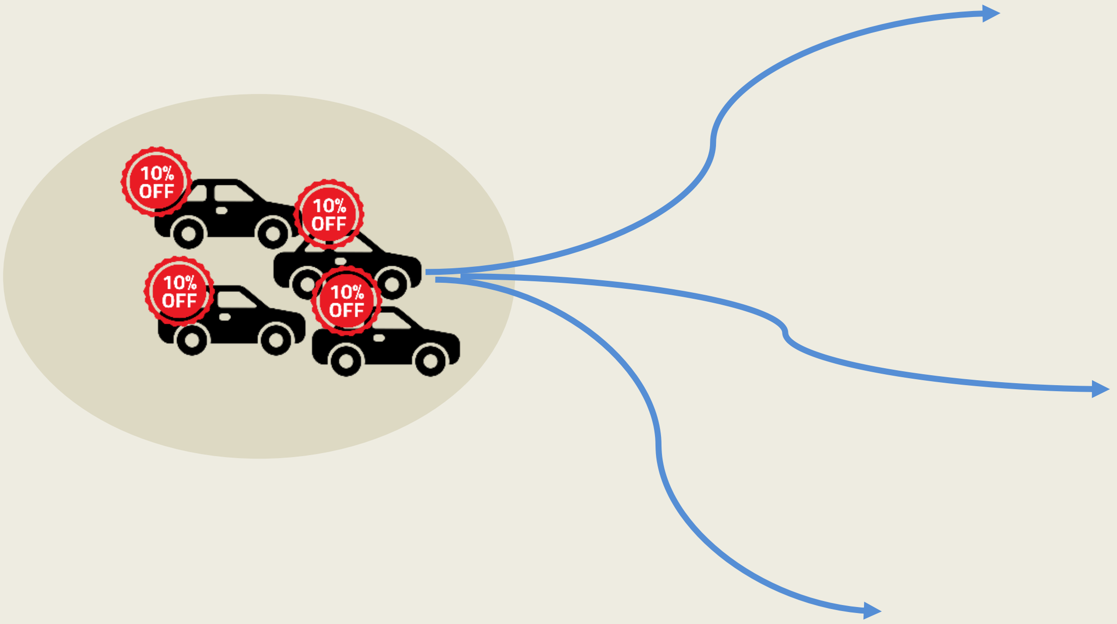
Pulled rebalancing

- Move cars to high-demand areas



Pushed rebalancing

- Move cars that are not be used for a period time to other areas



Method

- Use low-cost simulation and algorithm to get feasibility of applying incentive mechanism.
- The purpose is to assist in the dispatch of rental cars, ultimately achieving higher revenue and shareability.

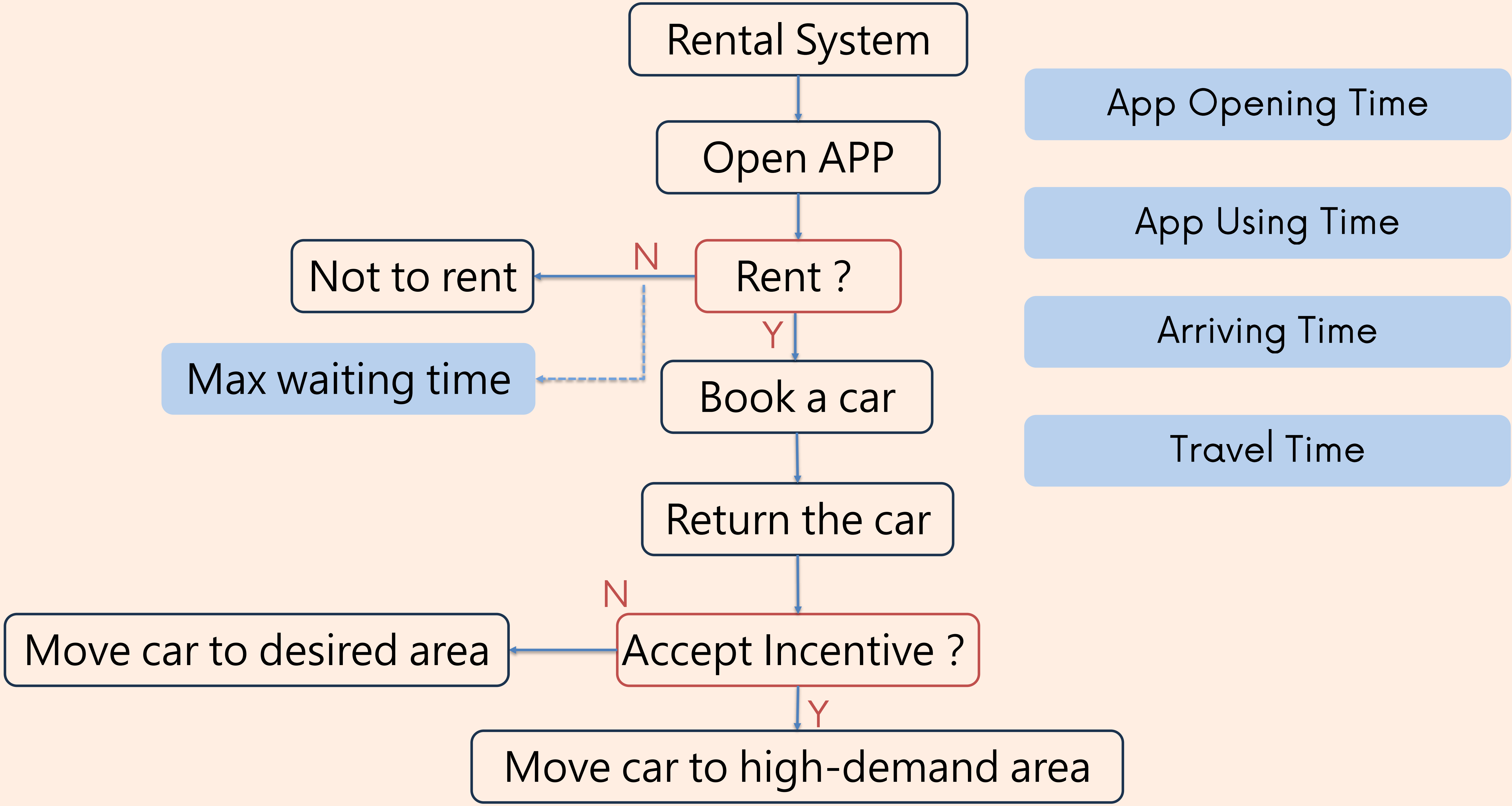
Stage 1

A **discrete event simulation model** is developed using historical data

Stage 2

The simulation model is utilized from Phase 1 to apply an algorithm aimed at **maximizing revenue**

Stage 1 Discrete Event Simulation – Rental Flowchart



Stage 1 Discrete Event Simulation

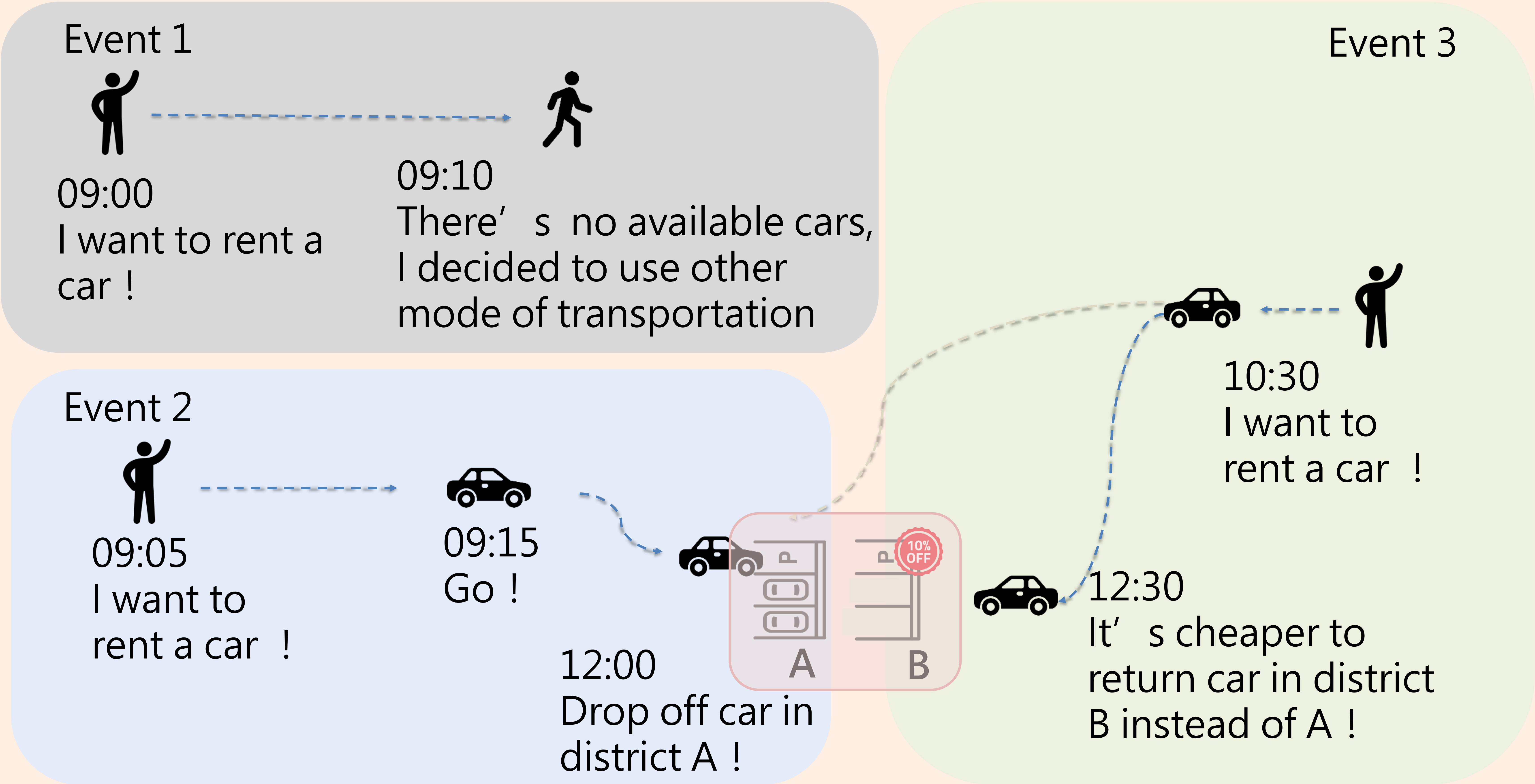
- Rental behavior
- Independent and identically distributed

Discrete Event Simulation?

Use historical data and statistics to simulate realistics

Stage 1

Discrete Event Simulation – Model Scenarios



Stage 1 Discrete Event Simulation Model

Car rental behavior consists of a series of stochastic actions in real world, which is difficult to explain through a mathematical model (MIP). In order to capture stochasticity, a large amount of historical data is used to simulate the real situation statistically

<u>System</u>

Free-floating car renting system (can be seen as a queuing system)

<u>Discrete Event</u>

User who open the app and is intend to rent a car is independent can be seen as a stochastic event. Users are independent and can be seen as stochastic discrete events

<u>Simulation</u>

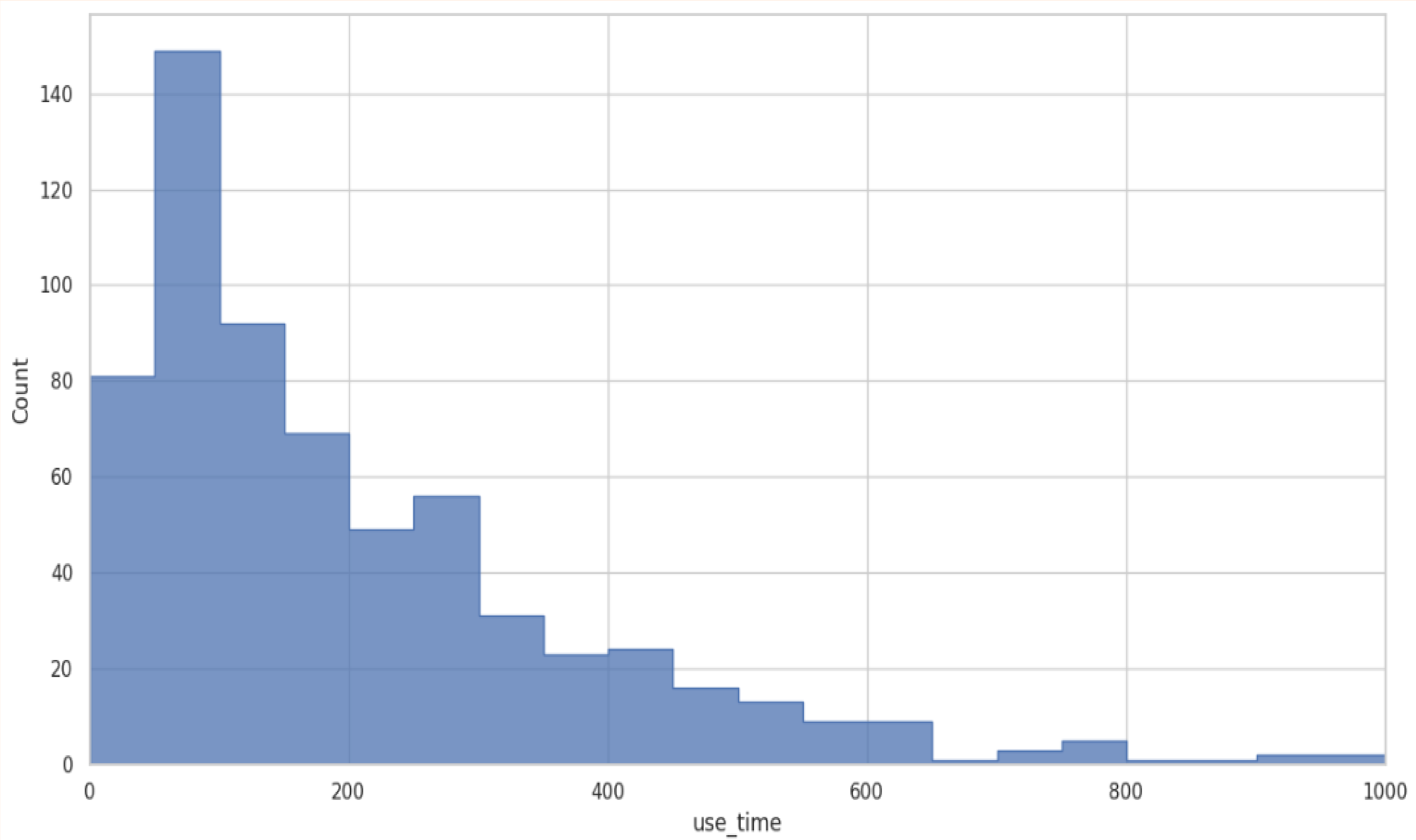
Simulate how many people will rent a car, how car will be moved, how much money could be earned, and so on during the operation time

Stage 1 Discrete Event Simulation – Distributions

probability distribution which is followed by

App Opening Time	Follow exponential distribution
App Using Time	Follow exponential distribution
Arriving Time	Follow exponential distribution
Travel Time	Follow log-normal distribution

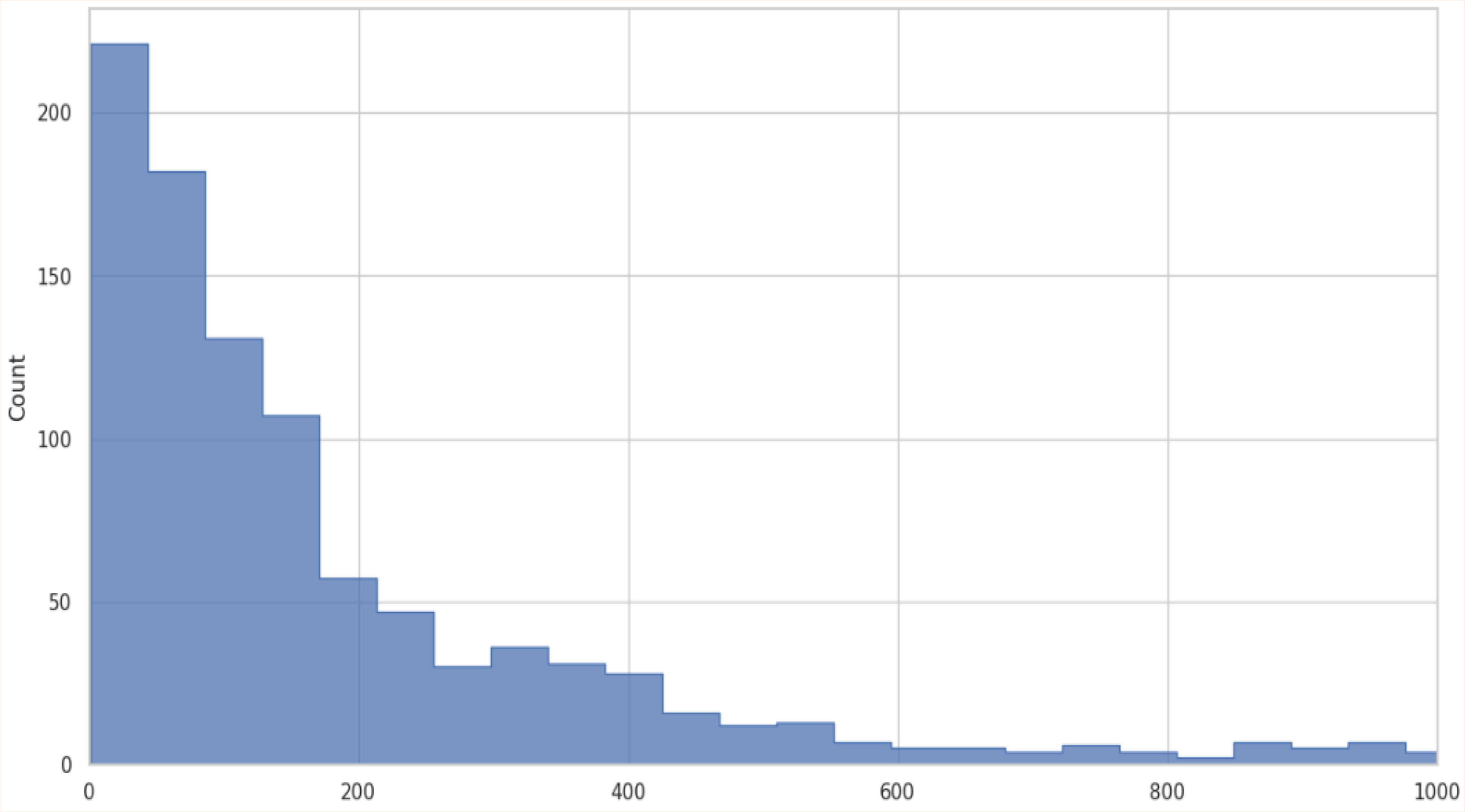
Stage 1 Discrete Event Simulation – Travel Time Proof



Histogram of real travel time distribution

Describe	Travel Time
mean	248.67
std	419.50
25%	71.55
50%	154.40
75%	288.73

Stage 1 Discrete Event Simulation – Travel Time Proof



Describe	Travel Time
mean	238.83
std	407.57
25%	49.09
50%	115.78
75%	262.65

Histogram of sampled travel time distribution

Stage 1 Discrete Event Simulation – Incentive Level Definition

Urgency = current car number / optimal car number (by area)
Incentive Level (max_incentive_level = 4)
0 < Urgency <= 0.25 -> Incentive Level = 1
0.25 < Urgency <= 0.50 -> Incentive Level = 2
0.50 < Urgency <= 0.75 -> Incentive Level = 3
Urgency > 0.75 -> Incentive Level = 4

max_accept_probability: maximum value of the probability that a user accepts to drop off a car to a new destination

I_relocation: Indicator if there's shortage in an area (1 / 0)

Incentive Accepting Rate = I_relocation * (Incentive Level / max_incentive_level) * max_accept_probability

e.g.
area 1 needs relocation, the urgency is 0.72, and max_accept_probability is 0.5
accept_probability = 1 * (3/4) * 0.5 = 0.375

Stage 1 Discrete Event Simulation – Pseudo Code

Main functions

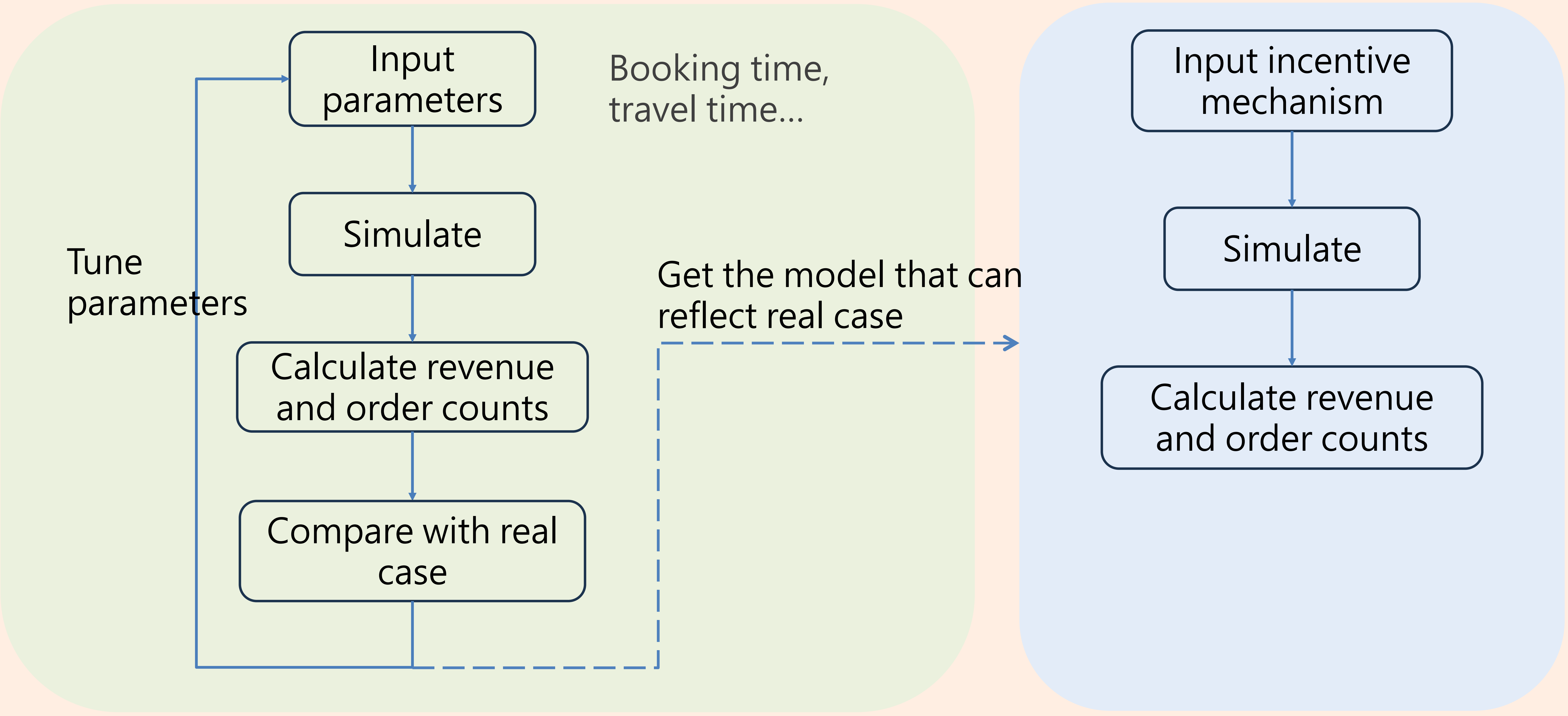
```
class CarSharingSystem():
    def sampling_travel_time()
    def _get_incentive_accepting_rate()
    def _set_dispatch_scenario():
        _get_incentive_accepting_rate
    def drop_off():
        _set_dispatch_scenario
        sampling_travel_time
    def rent_process():
        drop_off
    def generate_arrivals():
        rent_process
    def run_simulation():
        generate_arrivals
```

Pseudo Code

```
run_simulation() :
    environment set up
    generate arrivals
    start rent process:
        decide to dispatch or not
        decide to give incentive or not
        a customer to rent a vehicle or not
        if  a customer  rent a vehicle:
            decide origin and pick up a vehicle
            generate travel time
            if  give incentive:
                get incentive accepting  rate
            decide destination and drop off a vehicle
        else:
            leave the system
```

Stage 1 Discrete Event Simulation – Model Tuning

Objective Revenue



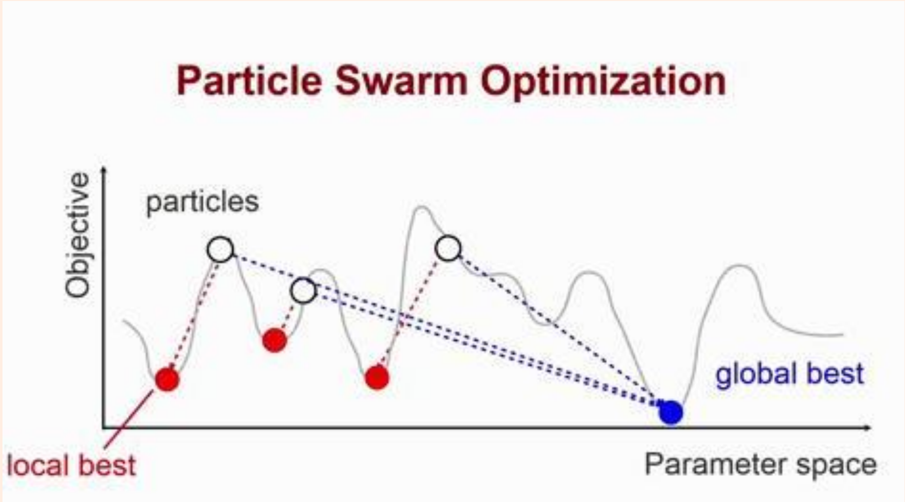
Stage 2 Particle Swarm Optimization, PSO

Mathematical Expression

$O_{Inertia}$ Feasible solution $O_{LocalOptimal}$ $O_{GlobalOptimal}$

$$\begin{matrix} P_{1,t_0} & P_{1,t_1} & \dots & P_{1,t_n} \\ P_{2,t_0} & P_{2,t_1} & \dots & P_{2,t_n} \\ P_{3,t_0} & P_{3,t_1} & \dots & P_{3,t_n} \end{matrix}$$

Objective：Maximize revenue ➡ Optimal car allocation



$$O_{x,t} = P(x, t)$$

Car allocation stands for decision variable x
t stands for iteration

$$P = w \cdot x_{inertia} + c_1 R_1 \cdot x_{LocalOptimal} + c_2 R_2 \cdot x_{GlobalOptimal}$$

Stage 2

Particle Swarm Optimization, PSO

Pseudo Code

for t in iteration:

    for p in particle:

        generate x

        calculate Objective value by discrete event simulation model

        update local optimal

    update global optimal

    #update particle

$$P = w \cdot x_p + c_1 R_1 \cdot x_{LocalBest} + c_2 R_2 \cdot x_{GlobalBest}$$

Parameters

particle = *10*

$c_1, c_2 = 1$

R_1, R_2 = *random number between 0 and 1*

w = *1*

Case Study

Boundary

Zhongshan District, Taipei – Divide into areas with a radius of 500 meters (a total of 86 areas)

Period

weekday / weekend x

Day peak	(08:00~17:00)
Night peak	(17:00~22:00)
Off peak	(22:00~08:00)

Main parameters

Renting probability, OD Metrix (weight is decided by times of history data)

Data

App Log (Open App、Slide App、Renting Operation)、Order detail(OD Location、Order Price)

註：資料時間為2024年7月

Case Study

Steps

1. Use PSO to find the best car allocation
2. Use Discrete event simulation model to simulate and calculate revenue and order counts

Compute Resource

Databricks all-purpose Cluster
1-8 Workers (1 is used)
16-128 GB Memory, 4-32 Cores

1 Driver
16 GB Memory, 4 Cores

Runtime14.3.x-scala2.12

Ref: Compute | Databricks on AWS

Compute Time

PSO: 100 repetitions
Discrete Event Simulation: 1000 repetitions

weekday	period	Discrete Event Simulation(s)	PSO(s)
weekend	day peak	17	17
weekend	night peak	9	9
weekend	off peak	11	12
weekday	day peak	12	12
weekday	night peak	8	8
weekday	off peak	8	7

Appendix Tools and Methods

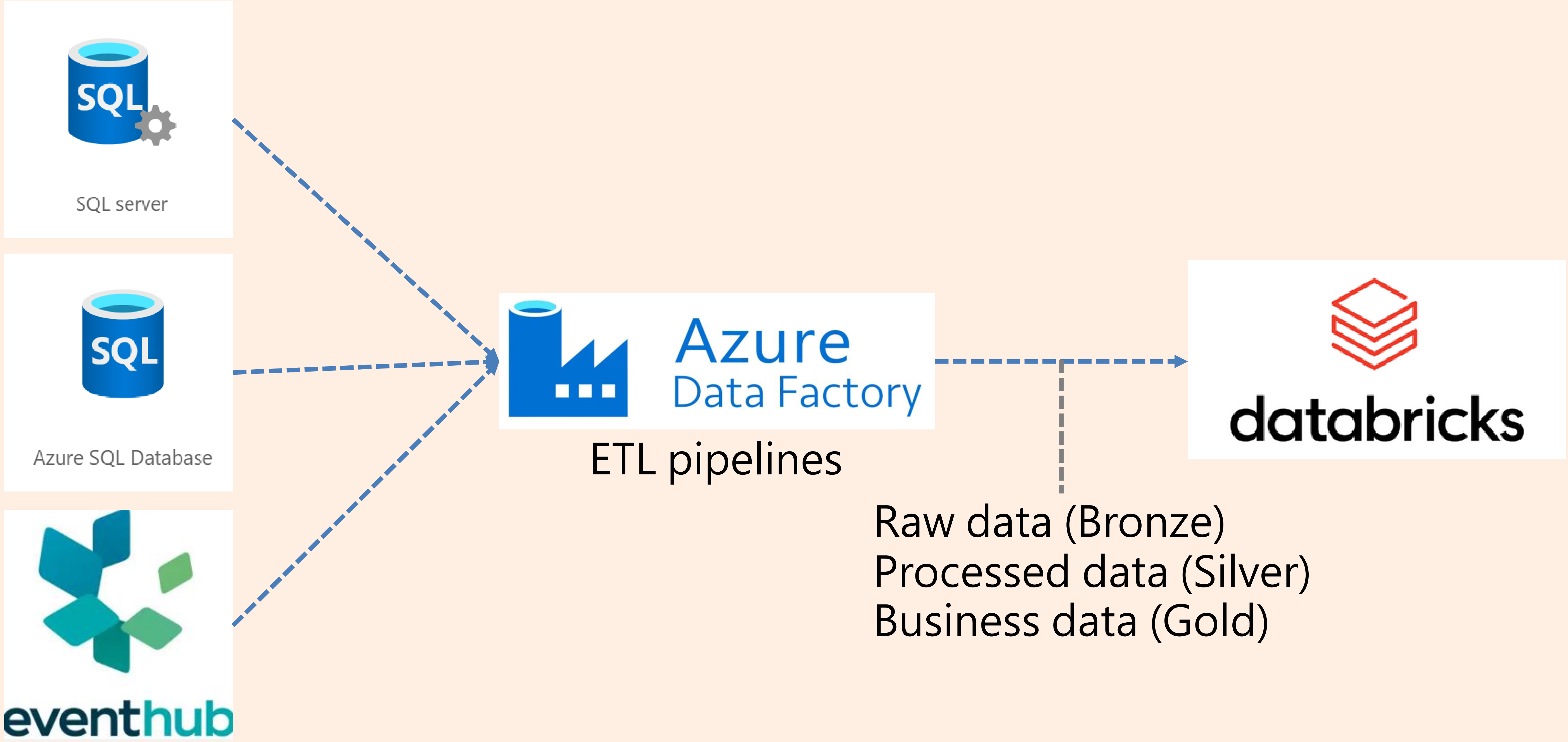
- Academic basis：Monica, C & Maria, PF et al. (2018) 'A Decision Support System for User-Based Vehicle Relocation in Car Sharing Systems'. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, VOL. 48, NO. 8, AUGUST 2018

- Methodology：Discrete event simulation (simpy)、Metaheuristics(PSO)

- Cloud-based platform：Azure Databricks

- AI tools：Azure Databricks Assistant

Appendix Data flow



ETL pipelines

Raw data (Bronze)
Processed data (Silver)
Business data (Gold)

Simulation model
Algorithm

Transaction data
App use log

Appendix Reference

- A decision support system for interactive decision making-Part I: model formulation | IEEE Journals & Magazine | IEEE Xplore
- Discrete-event simulation – Wikipedia
- Overview — SimPy 4.1.1 documentation
- Swarm intelligence - Wikipedia
- Lognormal distribution of daily travel time and a utility model for its emergence – ScienceDirect
- Dynamic Pricing for User-Based Rebalancing in Free-Floating Vehicle Sharing: A Real-World Case | SpringerLink