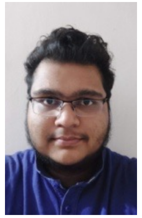
Computation Offloading In Dynamic Mobile Environment

Team W1

Capstone Project Phase-1

ESA Presentation

Team Composition

Dheemanth R Joshi	PES1UG20EC059	
Gautham Bolar	PES1UG20EC044	
Chennamsetti Sai Pranay	PES1UG20EC048	

Guide : Dr. Vamsi Krishna (ECE)

Introduction

- Task offloading in a highly dynamic environment is a challenging problem.
- Finds Applications in Automated navigation, and Intensive data processing such as object recognition
- Requires strict constraints for latency tolerance, bandwidth scheme and user privacy.
- The introduced scheme should be highly adaptive to the environment conditions.

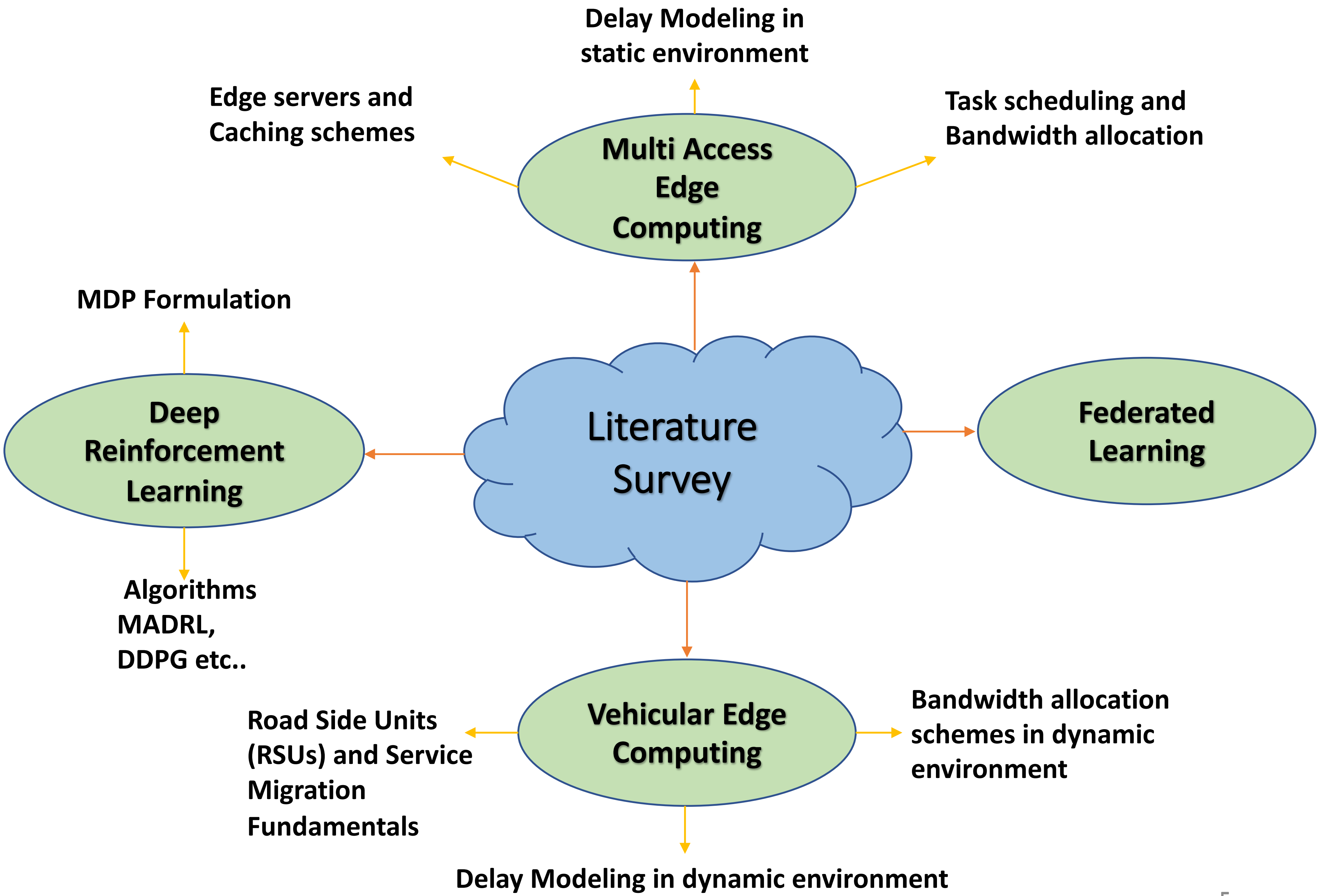
Motivation

- Enable Low Latency handling capability: To provide services to the customers within the promised delay constraints.

- Optimize Bandwidth Allocation: Provide more bandwidth to

- Improve Reliability: Robust system is required to handle these tasks which can be reliability

- Enhanced User Privacy and Security


Literature Survey
Multi Access Edge Computing
Delay Modeling in static environment
Edge servers and Caching schemes
Task scheduling and Bandwidth allocation
Deep Reinforcement Learning
MDP Formulation
Algorithms MADRL, DDPG etc..
Federated Learning
Vehicular Edge Computing
Road Side Units (RSUs) and Service Migration Fundamentals
Bandwidth allocation schemes in dynamic environment
Delay Modeling in dynamic environment

Y. Qi, Y. Zhou, Y. -F. Liu, L. Liu and Z. Pan, "Traffic-Aware Task Offloading Based on Convergence of Communication and Sensing in Vehicular Edge Computing," in *IEEE Internet of Things Journal*, vol. 8, no. 24, pp. 17762-17777, 15 Dec.15, 2021, doi: 10.1109/JIOT.2021.3083065.

State	**DRL Not used**
Action	**DRL Not used**
Reward	**DRL Not used**
Remarks	**1) Objective of the optimization problem was to partition data size and bandwidth such that the delay constraints are met and was solved by a TATO + binary search method**

W. Qi, X. Xia, H. Wang and Y. Xing, "A Task Partitioning and Offloading Scheme in Vehicular Edge Computing Networks," *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, Norman, OK, USA, 2021, pp. 1-5, doi: 10.1109/VTC2021-Fall52928.2021.9625369.

State	**DRL Not used**
Action	**DRL Not used**
Reward	**DRL Not used**
Remarks	**1) Utilized *contract-theory* to optimize RSU resource usage and delay requirements of vehicles.** **2) The *contact* was biased towards reducing the RSU's compute costs rather than delay sensitivity of vehicles.** **3) Does not utilize service-migration.**

Z. Zhu, S. Wan, P. Fan and K. B. Letaief, "Federated Multiagent Actor–Critic Learning for Age Sensitive Mobile-Edge Computing," in *IEEE Internet of Things Journal*, vol. 9, no. 2, pp. 1053-1067, 15 Jan.15, 2022, doi: 10.1109/JIOT.2021.3078514.

State	**1) Status of edge devices 2) Status of the buffers 3) Offloading bandwidth.**
Action	**1) Move 2) Execute Task 3) Offloading scheduling**
Reward	**Penalty given by measuring the age of the processed data**
Remarks	**1) State space of the MDP was not clear. 2) Doesn't deal with service migrations**

W. Zhan *et al.*, "Deep-Reinforcement-Learning-Based Offloading Scheduling for Vehicular Edge Computing," in *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5449-5465, June 2020, doi: 10.1109/JIOT.2020.2978830.

State	**1)Vehicle's task queue.** **2)Location** **3)Channel and transmission status based on location.**
Action	**3 Scheduling actions:** **1)Local execution** **2)Remote execution** **3)Hold (postpone task scheduling)**
Reward	**Minimization of Compute delay and energy consumed by vehicle.**
Remarks	**1) This paper considered task dependencies.** **2) Paper also considered energy optimization for tasks executed locally.** **3) Service migration was not utilized.**

Base Paper: Q. Yuan, J. Li, H. Zhou, T. Lin, G. Luo and X. Shen, "A Joint Service Migration and Mobility Optimization Approach for Vehicular Edge Computing," in *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9041-9052, Aug. 2020, doi: 10.1109/TVT.2020.2999617.

State (maintained by each vehicle)	**1)Source grid and destination grid** **2) number of vehicles in the grid** **3) service entities (VMs in our scenario)**
Action	**1) action of migration of the service entities** **2) action of routing (moving to next grid)**
Reward	**a positive reward if the vehicle reaches destination, negative reward if latency exceeds delay tolerance and cost migration magnitude**
Remarks	**1) Decision maker in the environment was not clear** **2) No considerations of bandwidth allocation for new vehicles coming into the RSU's contact after the beginning of the episode** **3) Fixed penalty in the reward function**

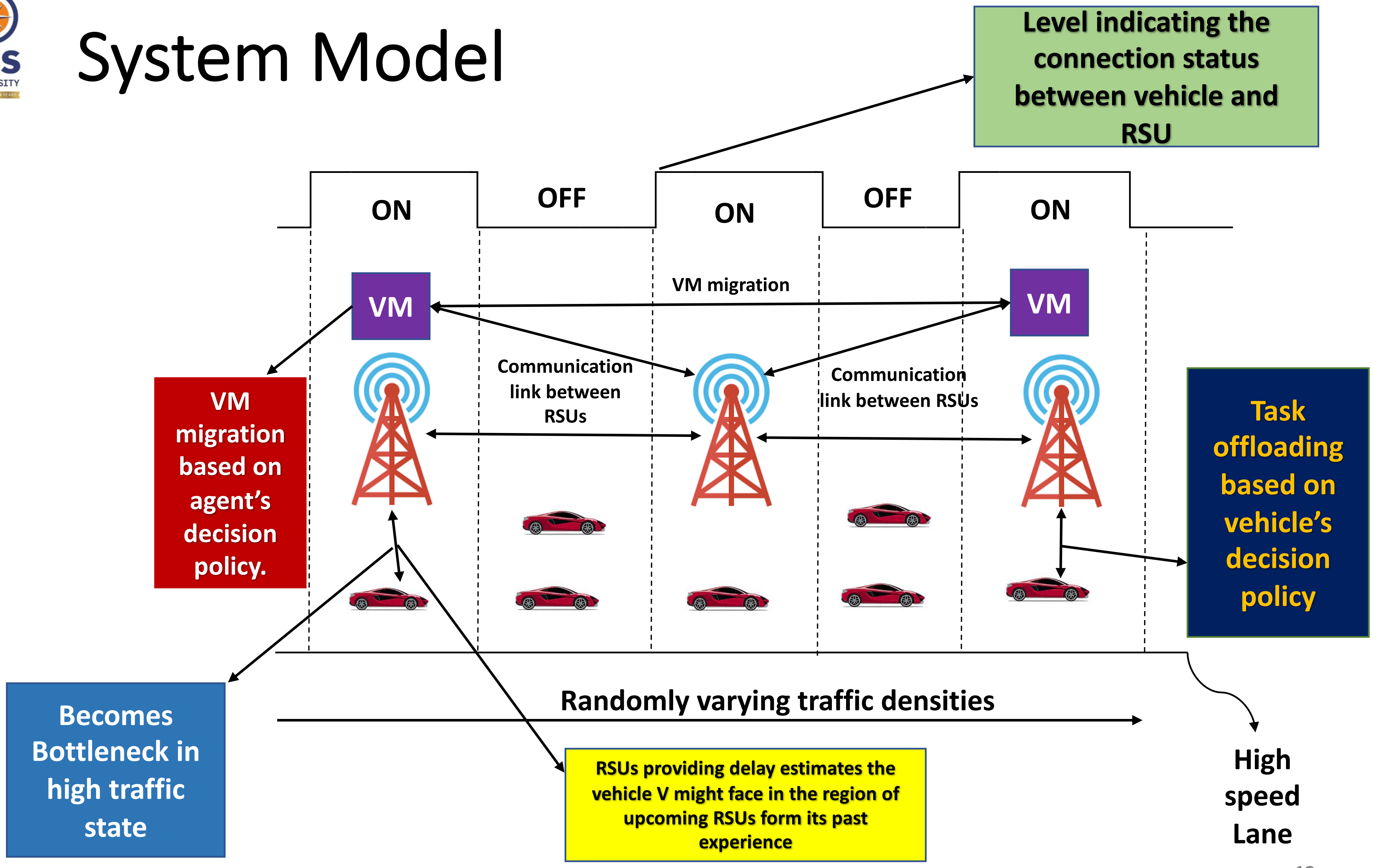
Main Research Gaps

- Channel bandwidth bottleneck.
- No limitation of compute resources at edge servers.
- Task partitioning and task dependency.
- Reward function modeling.
- Compute delay determines routing.

Finalized Problem Statement

- To collect the data from vehicles and build a model to understand their mobility pattern
- To assign optimal bandwidth and computing resources for the vehicles according to the mobility pattern
- Use the mobility pattern to perform service placement and service migration for the vehicles
- To build a multi-agent system which minimizes service migration cost of the vehicles

System Model

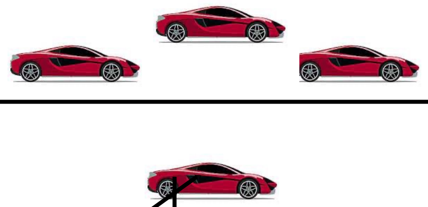


Methodology Flow

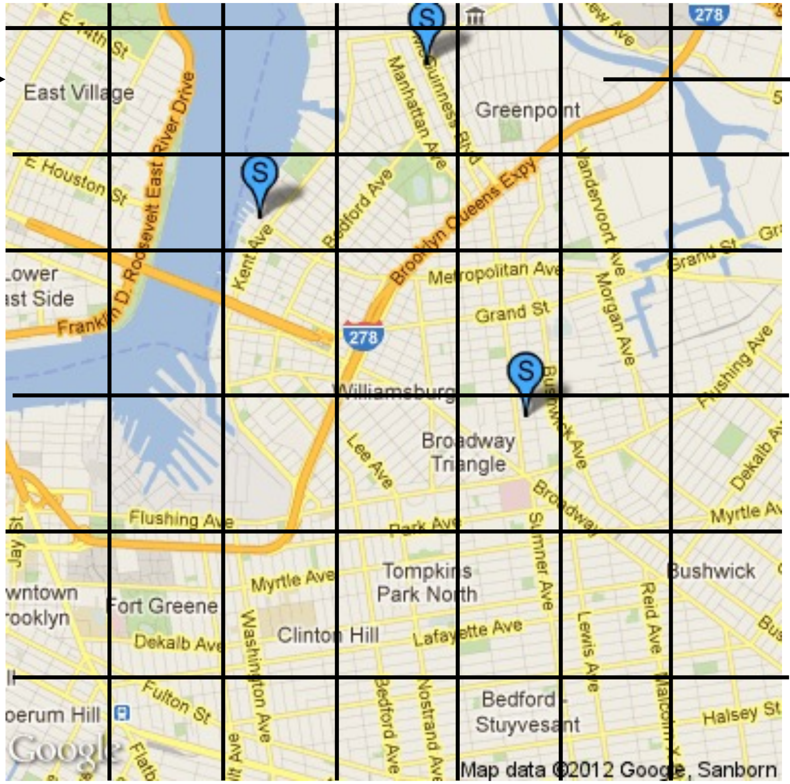
Environment Creation Synthetic Dataset Generation → **Formulation of Markov Decision Process (MDP)** → **Training The Deep Reinforcement Learning Model** → **Observations, Results and reports**

Environment Creation and Simulation

Vehicle Injection (Follows Poisson's Distribution)



The environment is divided into N grids as shown



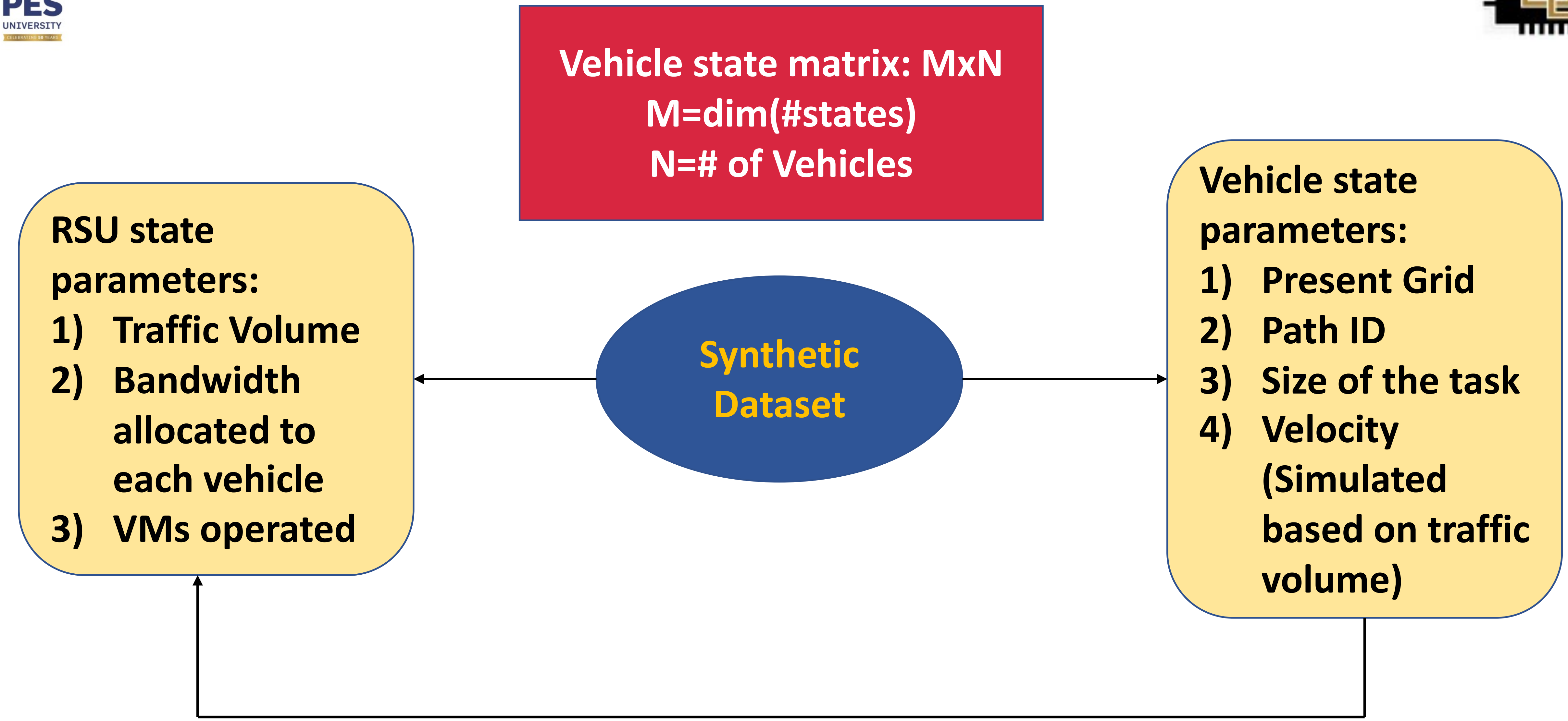
Grid Properties:
***Each Valid Grid Has a Road Side Unit (RSU)**
*** Vehicle density, average velocity, task arriving frequency and channel bandwidth are maintained in RSU.**

Pre-determined Source and destination grids, terminates the process once destination is reached

Each Vehicle takes its routing and offloading decisions based on the delay estimate provided by the present RSU

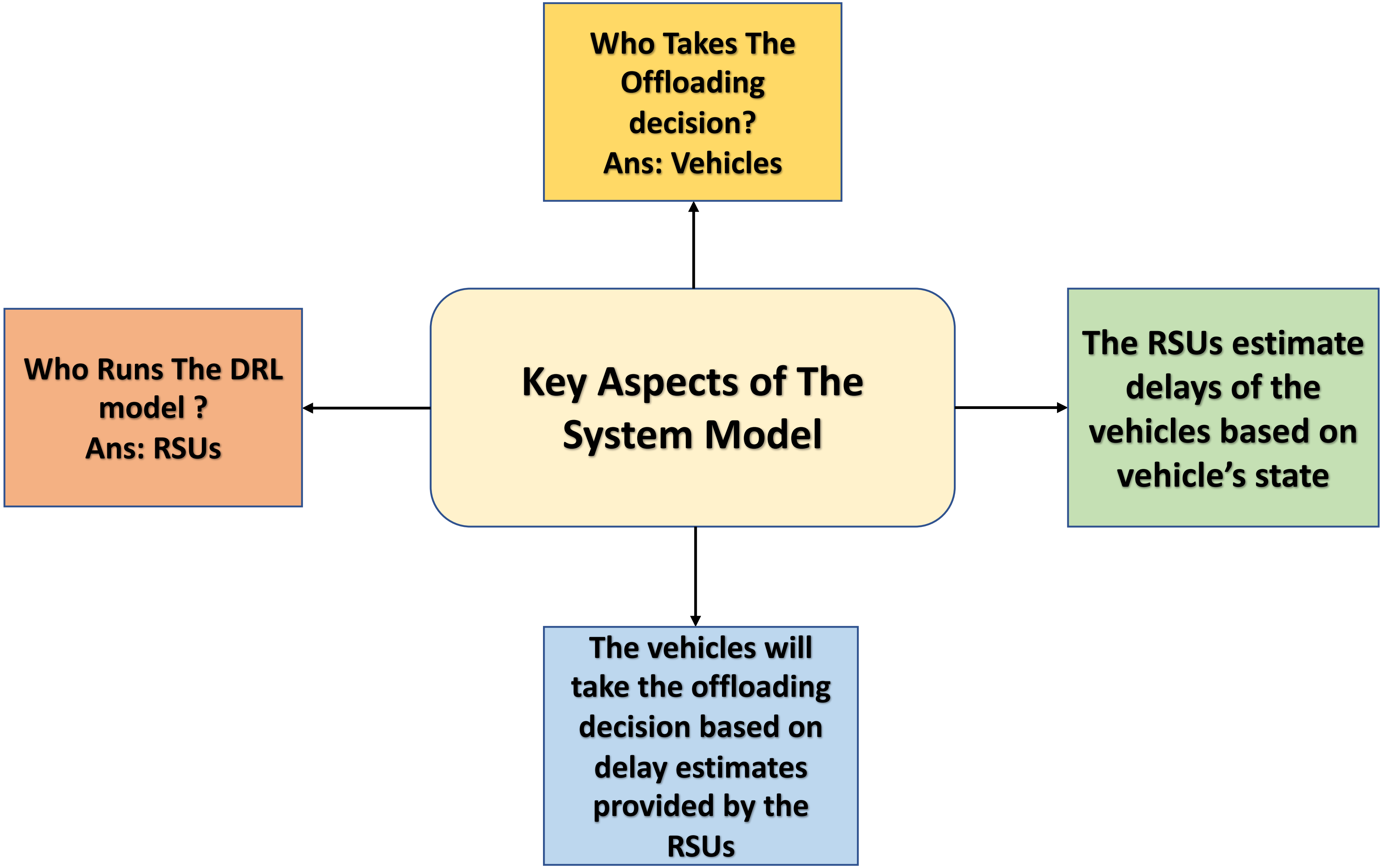
Taking this environment into consideration Synthetic dataset is generated

Each RSU communicates with its neighbors for important data exchange Ex: Service Migration


Vehicle state matrix: MxN
M=dim(#states)
N=# of Vehicles
RSU state parameters:
1) Traffic Volume
2) Bandwidth allocated to each vehicle
3) VMs operated
Synthetic Dataset
Vehicle state parameters:
1) Present Grid
2) Path ID
3) Size of the task
4) Velocity (Simulated based on traffic volume)

MDP Formulation and DRL Training

- Create an MDP by defining the states, allowed actions.
- MDP(s) with large state-action space are hard to solve.
- Must resort to using iterative methods.
- DRL is one such method, we make use of the data generated in the previous section to train the model.


Who Takes The Offloading decision?
Ans: Vehicles
Who Runs The DRL model ?
Ans: RSUs
Key Aspects of The System Model
The RSUs estimate delays of the vehicles based on vehicle's state
The vehicles will take the offloading decision based on delay estimates provided by the RSUs

Project Deliverables

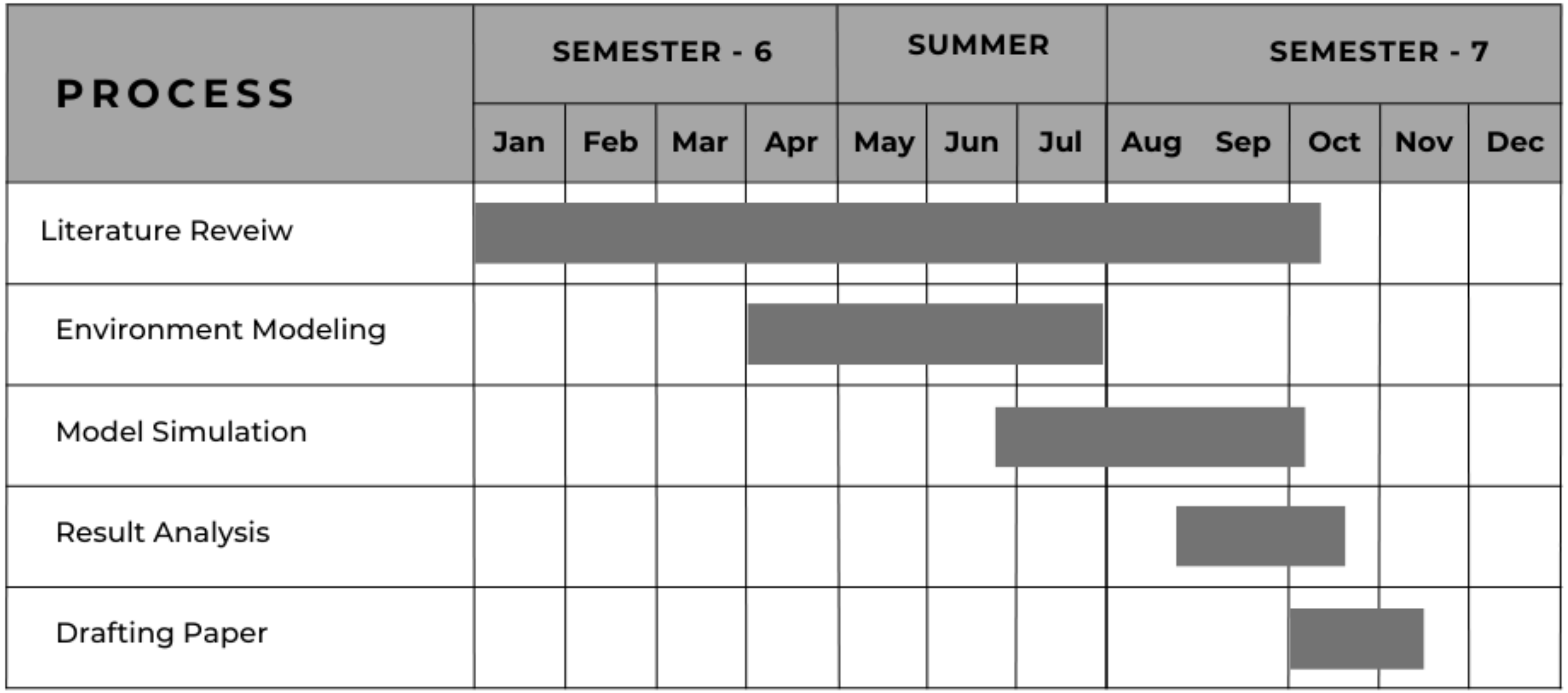
- Standard results reported in literature (eg: reward gained vs time)
- Comparative studies with existing literature.
- Handling of edge cases, research gaps unaddressed by Base paper. (eg: Compute resource limitations, No preference for routing).
- Behaviour of our model WRT to changes in environmental parameters, assessing bottlenecks, key factors related to offloading.

Software Requirements

- Python 3
- Modules used: PyTorch

Capstone Project TimeLine

----------2023



QNA Thanks