MEC-based Stable Topology Prediction for Vehicular Networks

- Asif Mehmood,
- Afaq Muhammad,
- Wang-Cheol Song,
- Taekyung Lee

Jeju National University, ATTO Research

AKRAINO

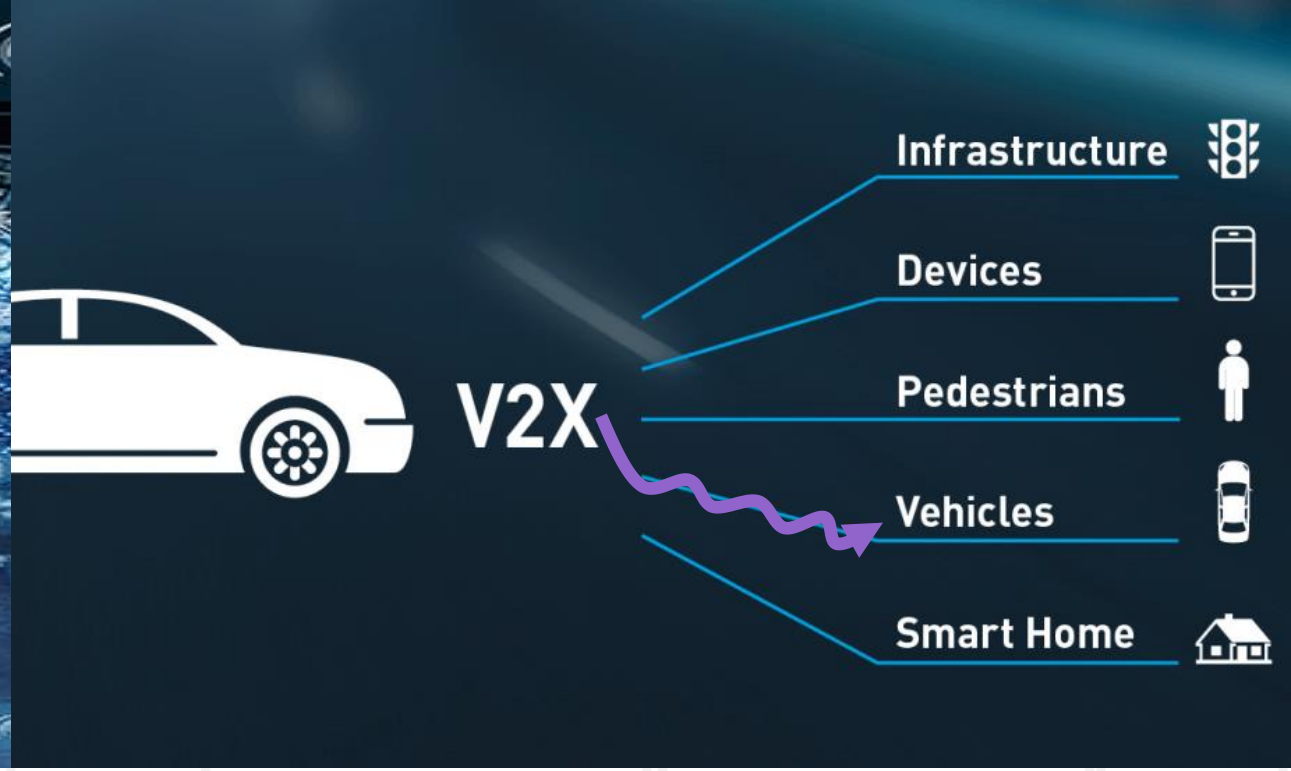
JEJU 1952
제주대학교
JEJU NATIONAL UNIVERSITY

Objectives of Blueprint

- Problem:
  - Vehicle-OBUs have different processing latency in sensors.

- Solution:

  - Predict the locations using Kalman filter, GPS add.
  - Rectify the location coordinates using the road-information (currently researching ways for designing the ERD which will store road-information at the edge).

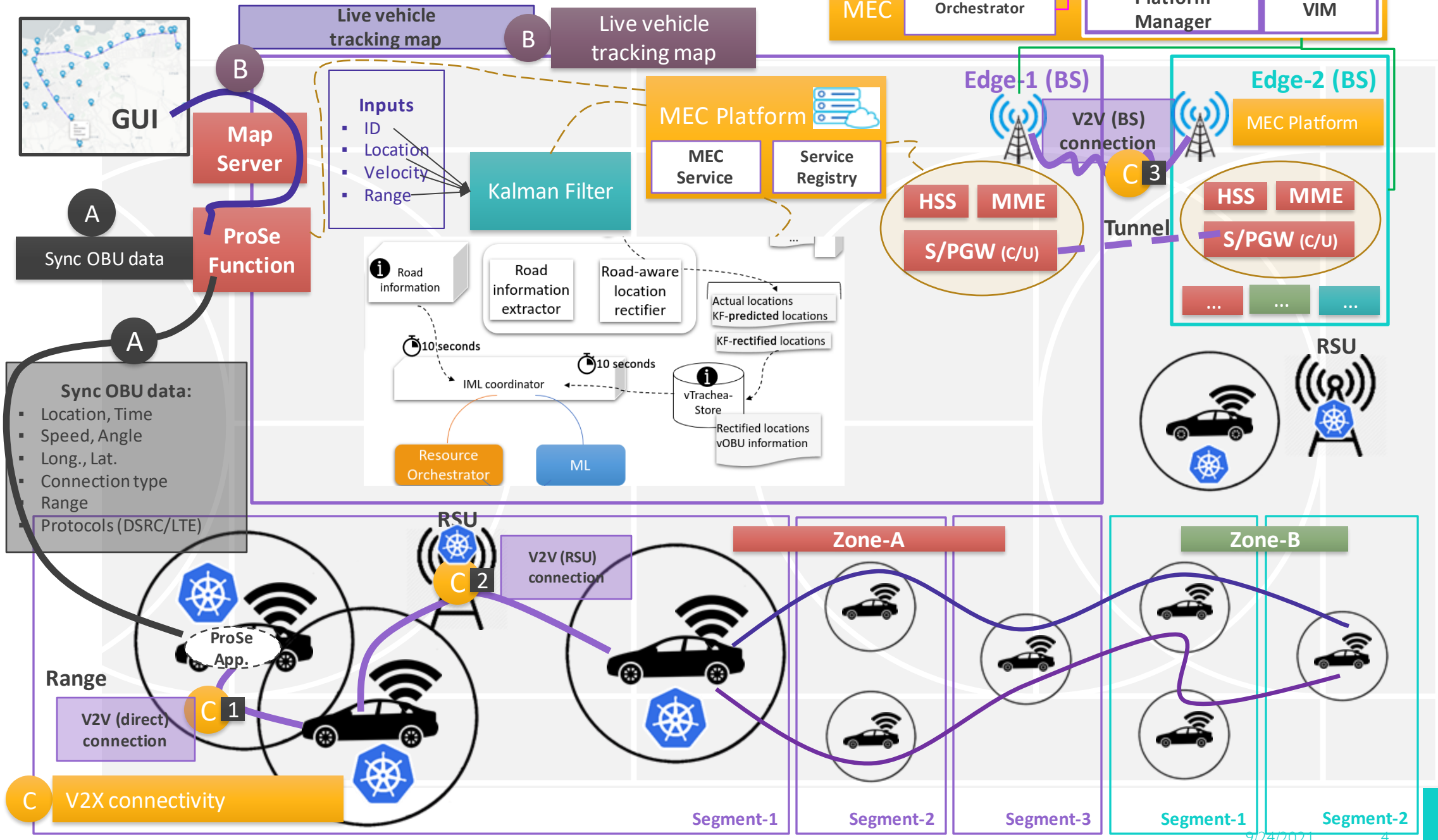


Automotive

New cellular networks like 5G enables additional applications such as:

- Autonomous driving
- V2X communications

INTRODUCTION



Data-set generation

Kalman filter prediction

Vehicle to edge push mechanism

API exposure (on which vehicle-info will be pushed)

Road-info extraction mechanism

ERD design for storing road-info

Rectification procedure

Machine Learning Model (ML)

Resource Orchestration (RO)

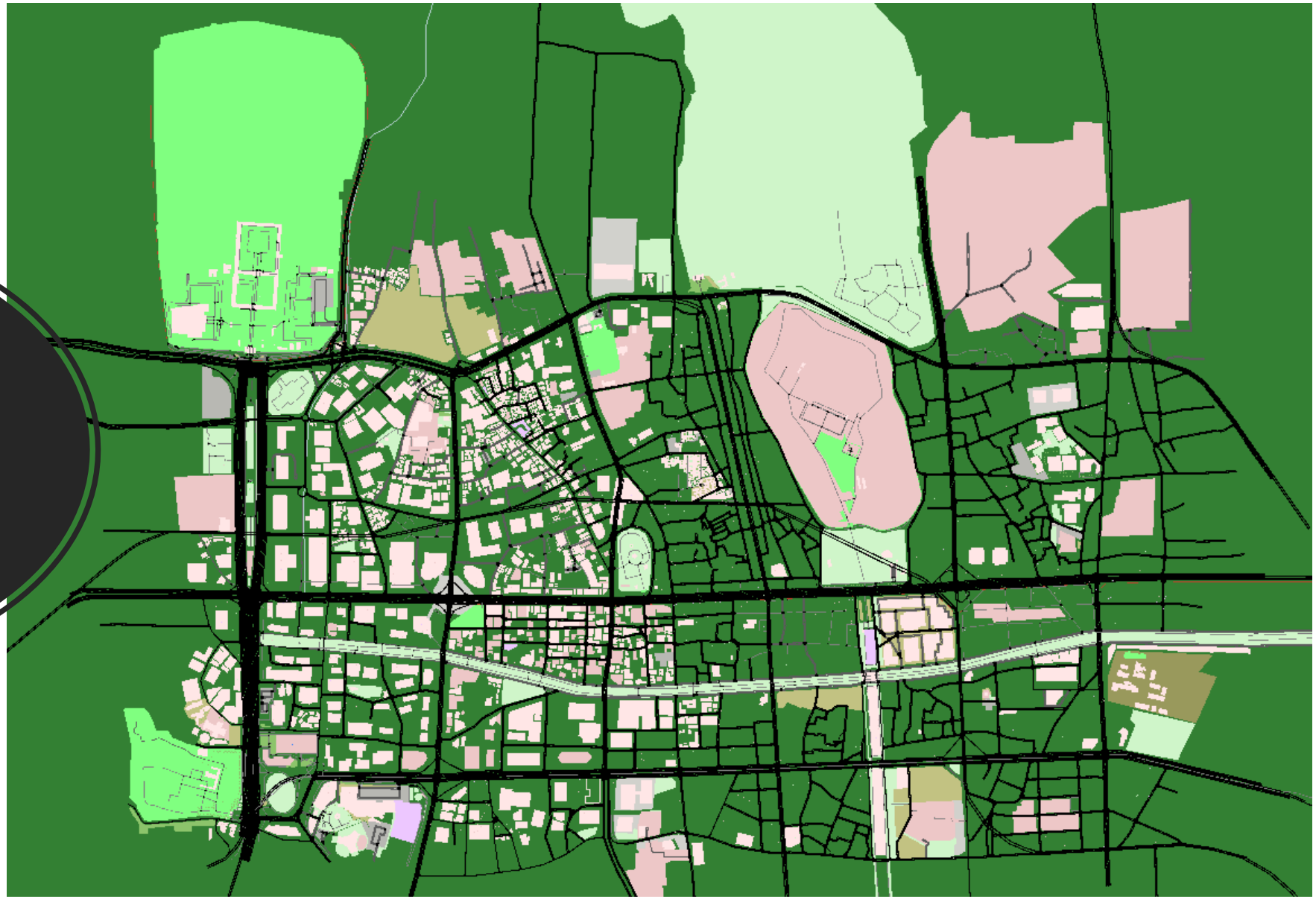
Trying to adopt as much standards

ETSI MEC
3GPP
NGMN

Much more to do

Data-set on Map of

Seoul, South Korea



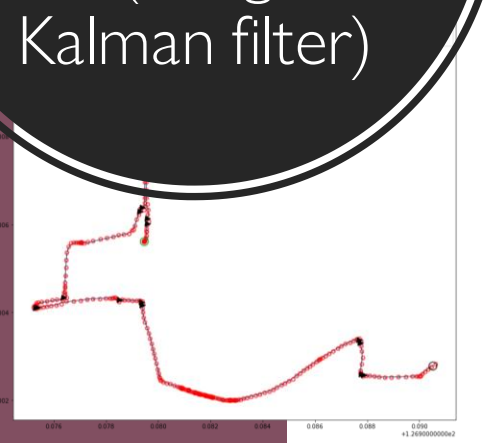
Dataset / API-info

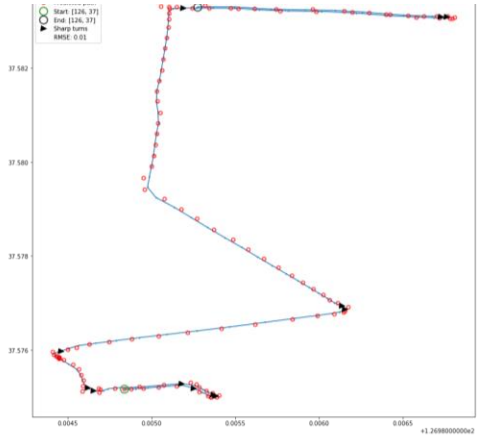
Kalman filter model API information			
Model	**Parameter type**	**Parameter name**	**Description of parameter**
Kalman filter	Inputs	vehicleid	the unique identifier of a vehicle.
		time	current timestamp 't'.
		longitude	Geographic coordinate that specifies the east–west position of a vehicle on the Earth's surface.
		latitude	Geographic coordinate that specifies the north–south position of a point on the Earth's surface.
		angle	Angle 0-359.99 to approximate the direction of a vehicle.
		speed	Speed of the vehicle at time 't'.
		lane	Lane number on which the vehicle is running.
		pos	Position of vehicle on which the vehicle is running on.
	Outputs	longitude	Predicted longitude.
		latitude	Predicted latitude.
Note: The API has not yet been developed but is a work in progress.			

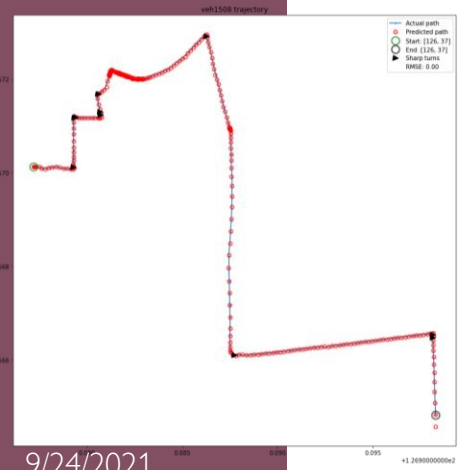
Inputs are the fields taken from the dataset.

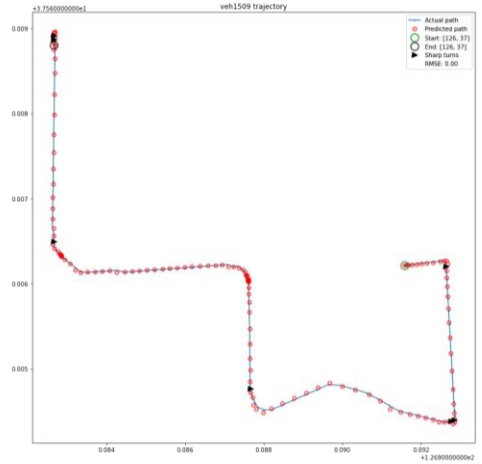
Vehicle trajectory prediction (using Kalman filter)

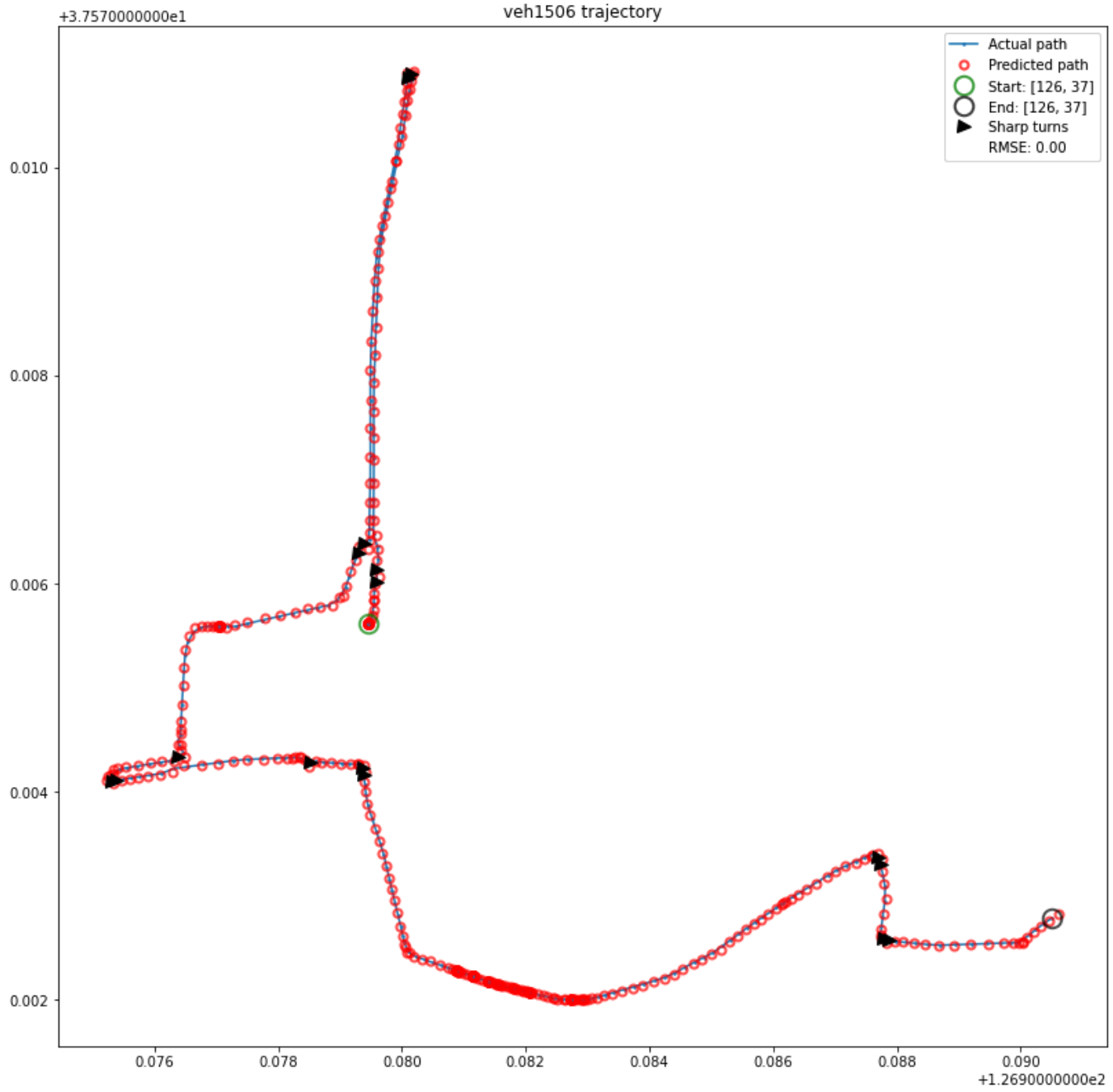
RMSE: 0.004







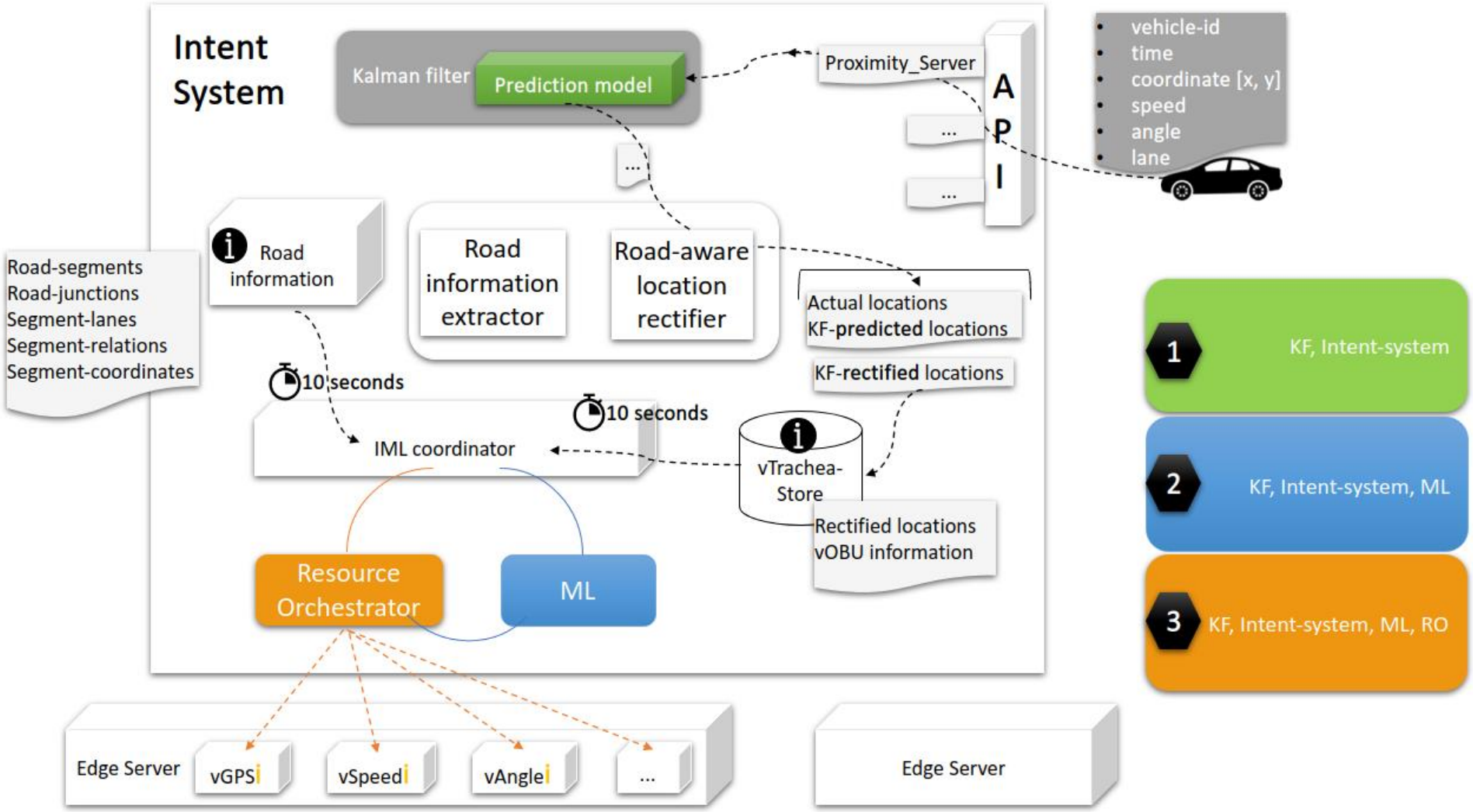




For	Name	Version/Info
Running Notebooks Kalman filter predcitions =>	Conda:	4.9.2
	Python:	3.8.10
	Jupyter Core:	4.7.1
	Jupyter Notebook	6.4.0
	Conda Environment File:	env_kf_model (file)
Pushing CD Logs Jenkins	Jenkins:	2.303.1
	Docker-hub image link:	mehmoodasif/jenkins
	pip3	20.0.2
	lftools:	0.35.10
Running Containers Docker	Docker:	20.10.8
	Docker build:	3967b7d
Map and Data-set Generation Data-set Generation =>	SUMO:	1.10.0
	TraceExporter.py	traceExporter (file)
	Netedit:	Netedit - SUMO
	Netconvert:	Netconvert - SUMO
Note:		

Software

Next…


Intent System
Kalman filter
Prediction model
Proximity_Server
API
vehicle-id
time
coordinate [x, y]
speed
angle
lane
Road information
Road information extractor
Road-aware location rectifier
Road-segments
Road-junctions
Segment-lanes
Segment-relations
Segment-coordinates
Actual locations
KF-predicted locations
KF-rectified locations
10 seconds
10 seconds
IML coordinator
vTrachea-Store
Rectified locations
vOBU information
Resource Orchestrator
ML
Edge Server
vGPS
vSpeed
vAngle
Edge Server
1
KF, Intent-system
2
KF, Intent-system, ML
3
KF, Intent-system, ML, RO

References

- ATTO Research
- Website - Akraino
- Wiki - Akraino
- Gerrit - Akraino
- Mailing lists - Akraino
- Blueprints - Akraino
- Calendars - Akraino