COORDINATING EDGE AND CLOUD
FOR BIG DATA ANALYTICS

The CLASS Software Architecture: Edge Analytics and Use Cases

Roberto Cavicchioli, UNIMORE

HiPEAC Virtual venue, April 22, 2021

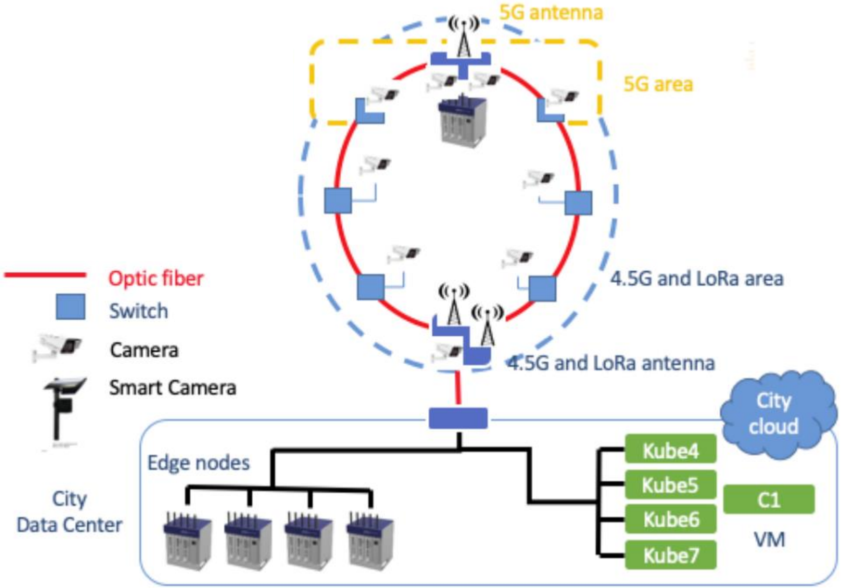
The CLASS project has received funding from the European Union's Horizon 2020 research and innovation programme under the grant agreement No 780622

Outline

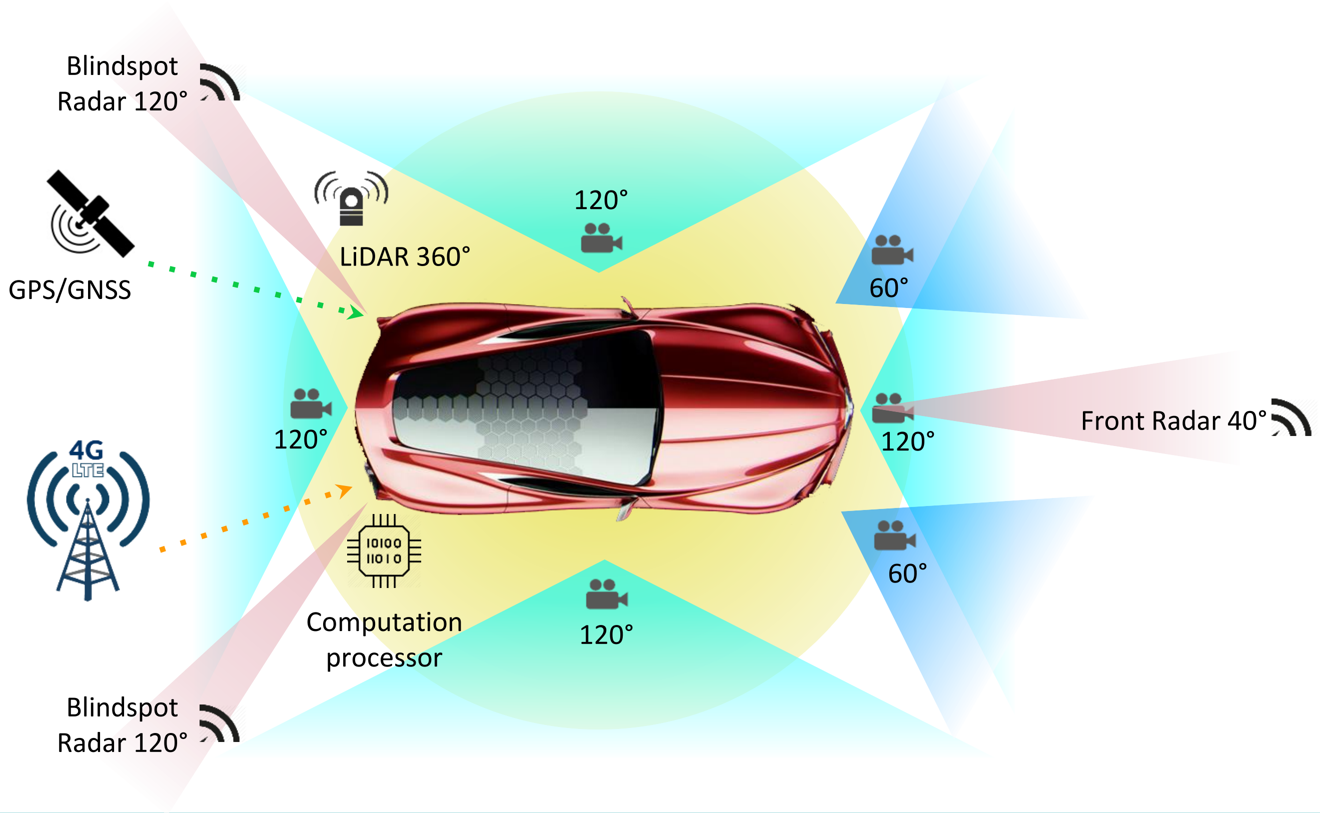
- The MASA infrastructure
- Use Cases
- Analytics

Infrastructure

Use Case Infrastructure – Modena Automotive Smart Area



On-board vehicle Components



Surroundings perception
- 6 Camera
- 3 Radar
- 1 LiDAR

Vehicle positioning
- GPS/GNSS receiver

V2I communication
- 4G-LTE antenna

On-board vehicle Components

Maserati Quattroporte MY19





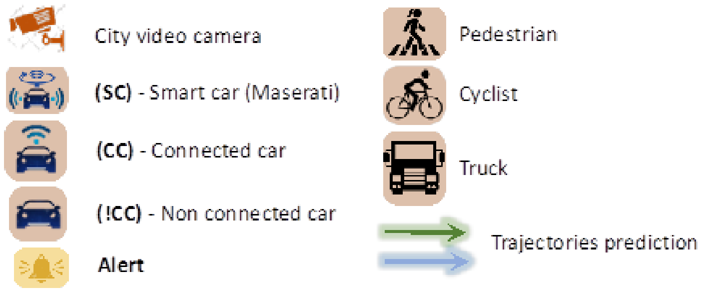
Units/car	Component	Model
1	Computation processor	Nvidia board – DRIVE AGX Pegasus
1	LiDAR (360°)	OUSTER OS2-128
2	Camera (60°)	Sekonix SF3325-10X
4	Camera (120°)	Sekonix SF3324-10X
1	GPS/GNSS	Xsens MTi-G-710-GNSSINS-2A8G4-DK

Use Cases

Obstacle Detection scenario

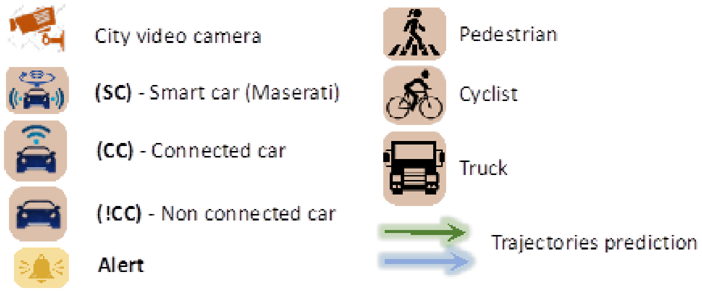
Scenario	Location in MASA	Actors	Functionality
1	Via Manfredo Fanti - Str. Attiraglio	City camera, Smart Car (SC), Connected Car (CC), and Non-Connected Car (!CC)	Virtual mirror
2	Str. Attiraglio	Two City cameras, Smart Car (SC), Truck, and Pedestrian	Virtual mirror
3	Via Manfredo Fanti (at the parking exit)	Two Smart Cars (SC), Connected Car (CC), and Cyclist	Virtual mirror
4	Via Maria Montessori - Via Pico della Mirandola (at the roundabout)	Two City cameras, Smart Car (SC), and Cyclist	Two sources of attention
5	Via Pico della Mirandola (at the parking exit)	City camera, Connected Car (CC), Non-Connected Car (!CC), and Pedestrian	Two sources of attention

Obstacle Detection: scenario 2
Virtual Mirror



The actors involved are: two city cameras at Attiraglio street that detect a pedestrian, a Maserati car (SC), and a truck that hides the view of the SC driver. The driver will receive an alert due to a hazard detection, since the pedestrian is crossing the street.

Obstacle Detection: scenario 5
2 sources of attention



The actors involved are: a connected car (CC) exiting the parking, and incorporating to the roundabout; a city camera detecting a non-connected car (!CC) and a pedestrian. The CC will receive an alert various hazard detection.

Analytics

Position of objects

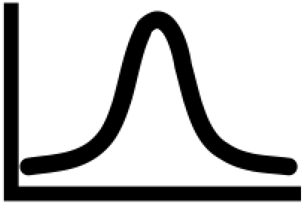
- Detection performed on camera → we obtain bounding boxes on the camera image
- GPS conversion → all the lower centers (pixel x, pixel y) of the bounding box are converted in (GPS latitude, GPS longitude)
- Meters → the GPS points (GPS latitude, GPS longitude) are finally converted in meters (x [m], y[m]) w.r.t an origin on order to feed the tracker with meters and not GPS coordinates and perform tracking from a map point of view and not from the camera itself.

DNN Performance

	Yolo v3 COCO (416x416)	Yolo v3 Berkeley (416x416)	Yolo v3 Berkeley (544x320)	Yolo v3 Berkeley (488x256)	Yolo v2 Berkeley (544x320)	Yolo v4 Berkeley (544x320)
FULL	6 fps	6 fps	5.5 fps	10 fps	**11.5 fps**	5.6 fps
HALF	7.5 fps	7.5 fps	7.5 fps	12 fps	**17 fps**	7.2 fps
Map	0,94	44,63	47,07	45,16	33,77	**51,8**
map person	**70,24**	45,81	51,54	47,61	32,04	59,11
map car	4,78	66,7	70,78	66,4	51,78	**76,5**
map truck	0,02	57,34	58,14	57,01	51,23	**64,0**
map bus	0,01	55,09	56,65	55,78	51,63	**64,2**
map bike	0,05	39,05	**41,41**	40,53	33,49	40,63
map motor	0,03	34,3	36,57	39,66	31,2	**41,01**

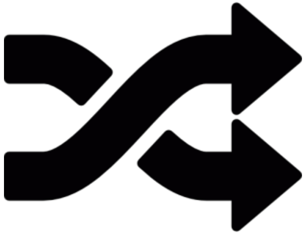
Tracking Overview

- Filter every object using the lower center of the bounding box
- Extended Kalman Filter
- Nearest neighbor technique to associate the bounding boxes of the same object in successive frames.
- One tracker for each object → multiple trackers



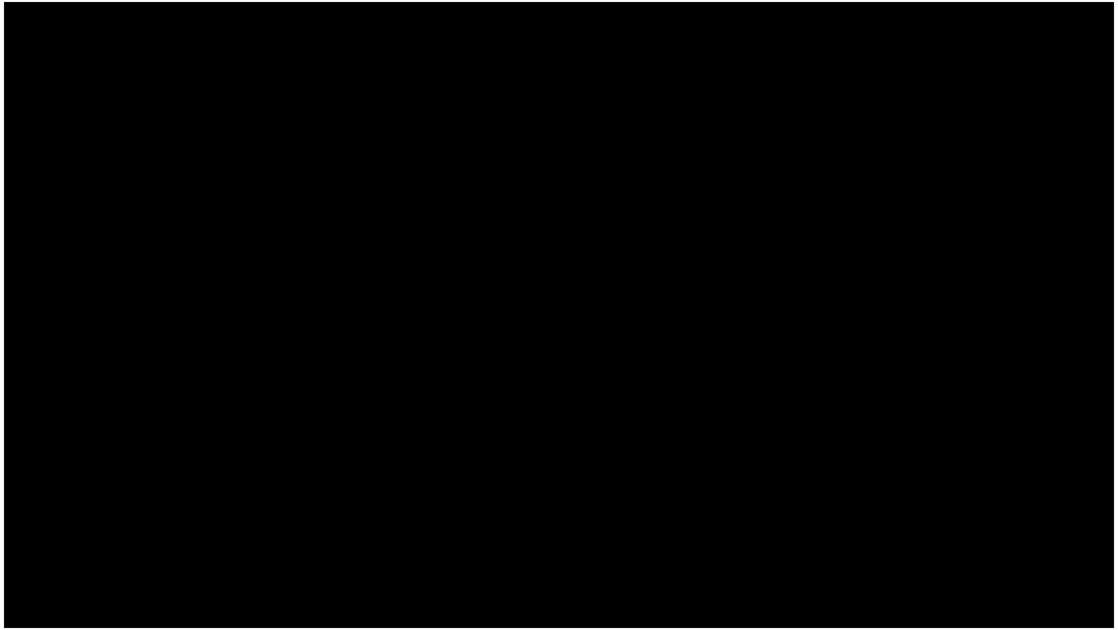
Extended Kalman Filter

Nearest Neighbor



Multiple trackers

Multiple trackers: view from camera



Multiple trackers: view from the top

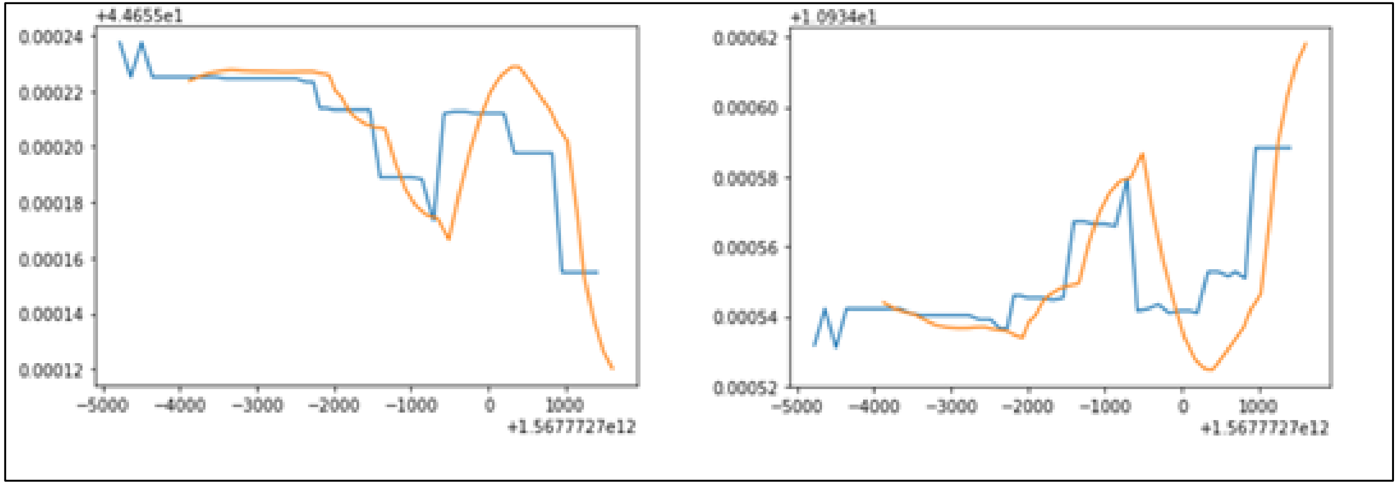
Trajectory prediction

TP Algorithm in CLASS is able to:

- Compute multiple predictions per object, in different future time points.
- Input samples are not equally apart in time.
- Predict trajectories for multiple objects at the same time.

The core trajectory prediction needs to be applied to each object in the MASA, and only when new location data is available for the object

Trajectory prediction



Example of X and Y outputs for the trajectory prediction using a range of 200 milliseconds. In blue the original path; in orange the predicted path

Collision detection

The core collision detection algorithm is applied to a pair of street objects (car, pedestrian, etc) from the DKB. The collision detection then computes predicted path intersections as following

- A quadratic regression is computed to get the functions for the two objects that are in the scope, to later get possible intersections
- When the two trajectory functions are defined, both are crossed to get the intersection points, if exists
- The next step is to get the timestamps for both objects when they get the intersection point. At this phase, a variable of 2 seconds has been defined to arise an alert if they converged at the intersection point with a difference of less than 2 seconds

Collision detection

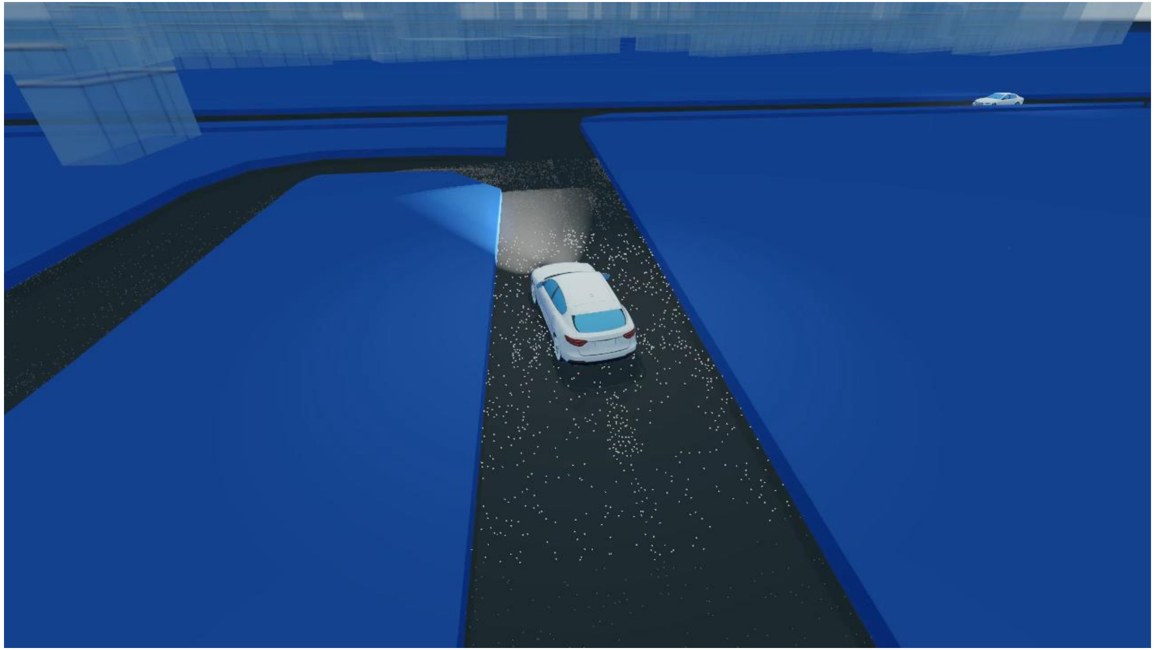
- At the end, if potential collisions are detected, they are sent with the information of the two objects that could crash, the X and Y coordinates and the timestamp associated

```
---------------------------------------
WARNING!!! Possible collision detected
  v_main: 2128793555 v_other: 1348026716
    x: 44.65553424126951 y: 10.934492820873857 t: 1567772745174.0
---------------------------------------
```

Example of an alert when a potential collision is detected

- Those collisions are finally sent to the next component of the workflow, in our case to the WA alert visualization

Warning visualization



Thank you!

Questions?

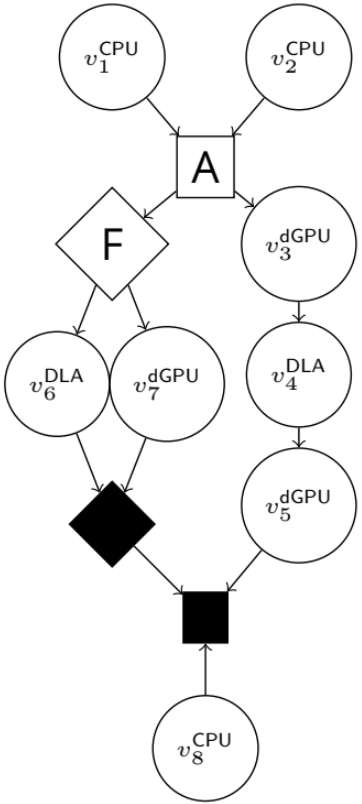
www.class-project.eu
Twitter: **@EU_CLASS**
LinkedIn: **http://bit.ly/CLASS-project**

Backup

Dynamic Work Distribution

DAG nodes

- **Alternative nodes**: to model alternative implementations of the same functionality on different computing engines to be selected off-line
- **Conditional nodes**: to model if-then-else branches to be selected at run-time



Scheduling analysis

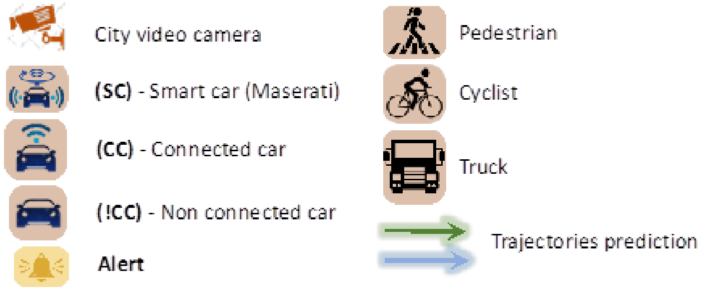
- Each engine has its own scheduler and a separate ready-queue.
  - Sub-tasks are allocated (partitioned) onto the available engines so that the system is schedulable.
- Meeting timing constraints of a concrete task depends on the allocation of the sub-tasks onto the different execution engines.
  - To reduce the complexity of dealing with precedence constraints directly, we impose intermediate offsets and deadlines on each sub-task.

Task Allocation

CLASS

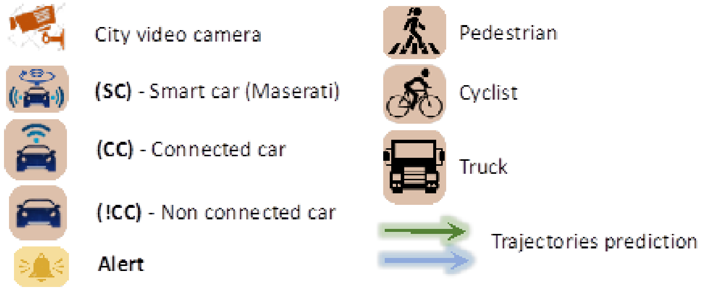
- ILP formulation of this problem fails to produce feasible solutions in an acceptable short time, therefore we propose a set of greedy heuristics to quickly explore the space of solutions.
  - Single-engine allocation
  - Best-fit/Worst-fit allocation
  - Serial/Parallel allocation

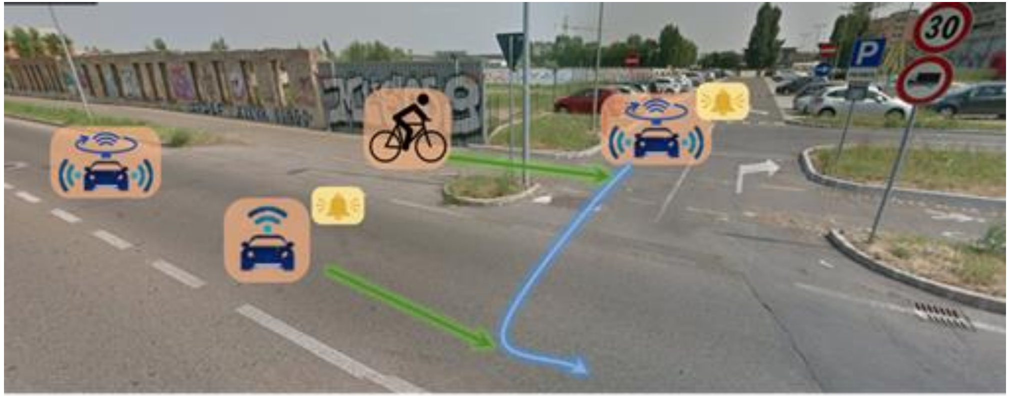
Obstacle Detection: scenario 1



The actors involved are: a city camera at Attiraglio street that detects a vehicle (!CC), and a Maserati car (SC) that detects a connected car (CC) at Via Manfredo Fanti, both reaching the intersection. The connected car (CC) will receive an alert due to a hazard detection, since the non-connected car will exit Attiraglio street.

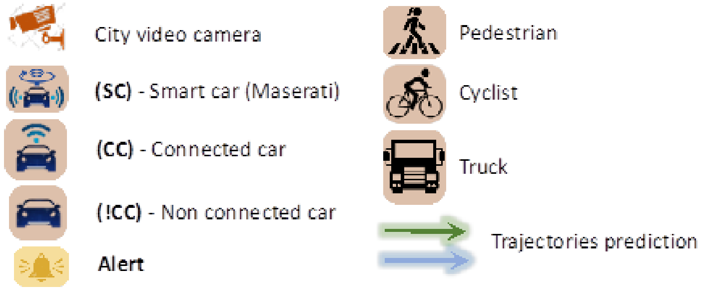
Obstacle Detection: scenario 3





The actors involved are: a Maserati car (SC 1) exiting the parking; and a second Maserati car (SC 2) that detects a connected car (CC) at Via Manfredo Fanti, and a cyclist, the three of them reaching the intersection with the parking exit. Both, the SC 1 and the CC, will receive an alert due to various hazard detection.

Obstacle Detection: scenario 4



The actors involved are: a connected car (CC) reaching the roundabout; and two city cameras that detect the CC and a Cyclist at the roundabout. The CC will receive an alert due to a hazard detection, since the cyclist may be crossing the street.

On-board vehicle Components (2/4)

Maserati Quattroporte MY18

Units/car	Component	Model
1	Computation processor	Nvidia board - DRIVE PX2 Autochauffeur
1	LiDAR (360°)	VLP-16 channel PUCK
2	Camera (60°)	Sekonix SF3325-10X
4	Camera (120°)	Sekonix SF3324-10X
1	GPS/GNSS	Xsens MTi-G-710-GNSSINS-2A8G4-DK

On-board vehicle Components (3/4)

Maserati Levante





Units/car	Component	Model
1	Computation processor	Nvidia board - DRIVE AGX Xavier
1	LiDAR (360°)	OUSTER OS1-64
2	Camera (60°)	Leopard Imaging LI-AR0231-GMSL
4	Camera (120°)	Leopard Imaging LI-AR0231-GMSL
1	GPS/GNSS	Xsens MTi-G-710-GNSSINS-2A8G4-DK