# **­EcoTraffic: smart urban mobility for a greener future**

* 1. **The problem**

Two urgent global concerns are environmental sustainability and climate change; because of air pollution and greenhouse gas emissions, transportation—especially urban commuting— contributes to worsening those issues.

* 1. **The Goal**

We want to dynamically modify the duration of traffic lights on the main roads in the city depending on the directions from where we observe the main traffic movements. For instance, if, at a certain point in time, we observe that the traffic flow on a certain road A is significantly higher than in the crossing roads, then we may decide to extend, for instance, for one hour, the duration of green lights on A (and, consequently, extend the duration of red lights in the crossing roads).   
We want to analyze the daily traffic patterns and identify possible optimizations in terms of one-way roads, traffic lights configuration, and public transport schedule.   
We want to collect information about the planning of events attracting large crowds (e.g., important sport events, concerts, fairs) and define event-specific configurations for traffic lights, roads and public transport schedules.

* 1. **Stakeholders**

Drivers: for reduced waiting times stuck in the traffic

Citizen: for air pollution, public transport

Urban Traffic manager: for optimization of city viability

Events planner: for a better management of events attracting large crowds

**2 Requirement Analysis**

**2.1.1 Human Actors**

Drivers: benefit from the EcoTraffic system

Citizen: benefit from the EcoTraffic system

Urban Area manager: approve or rejects modifications to one-way roads, traffic lights configuration, and public transport schedule.

**2.1.2 Non-Human Actors**

Traffic Lights: get its state set from the ET system for a determined time period.

Sensor Infrastructure: send sensor information to the ET system via data bus.

Public Transport Microservice: sends public transport schedules to the ET system via function calls.

News Channel: transmits city events information to the ET system.

* 1. **Use Cases**
     1. **Scenarios (to modify)**

**Traffic light duration adjustment**

1. During peak traffic hour, cars take more than the necessary to cross a particular intersection coming from a busy road, while the crossing road are less used.
2. The system sensor measures the times taken and publish them on the data bus.
3. EcoTraffic retrieves the data from the data bus and stores them into a database.
4. EcoTraffic tries to catch misbehaviour.
5. If a misbehaviour is detected, then EcoTraffic compute the new routines time for the traffic light which must be modified.
6. EcoTraffic connects to the traffic light control system providing serial numbers of the traffic lights which must modify the routine with the amount of time interval to be changed.
7. EcoTraffic writes into a log file the modifications done.

**Daily Analysis and optimization**

1. Ten minutes after midnight, EcoTraffic retrieve daily data from the database.
2. EcoTraffic analyse this data to detect daily traffic pattern and to find possible optimization in term of one-way roads, traffic lights configuration, and public transport schedule.
3. EcoTraffic performs the getScheduleByStreet and the getScheduleByLine offered by the microservice.
4. EcoTraffic tries to reorganize the schedules to optimize the public transport.
5. If possible optimizations are found, EcoTraffic store them into the database.
6. Once the UAM access the system, it presents the optimizations found and wait for the answer of the UAM.
7. EcoTraffic writes into a log file the answer for yearly reporting.

**Traffic Zone Optimization**

1. The Urban Area Manger asks to EcoTraffic to verify if a certain zone is optimized in terms of one-way roads, traffic lights configuration, and public transport schedule.
2. EcoTraffic retrieve from the database the traffic patterns of the zone.
3. EcoTraffic analyses the data retrieved and tries to minimize the medium amount of time taken to cross an intersection into the zone.
4. EcoTraffic performs the getScheduleByStreet and the getScheduleByLine offered by the microservice to get all the bus lines that has a stop into the area to optimize.
5. After the information arrived, EcoTraffic tries to reschedule the timetable of the bus lines in such a way to minimize the traffic. (i.e if in the zone there’s a school, then the system could anticipate the timetable of the stops near that to avoid the students to arrive late or could suggest increasing the number of buses in the area).
6. EcoTraffic presents to the UAM the recommendations and waits till the UAM decide to accept or to reject the recommendation and store the decision.
7. EcoTraffic writes into a log file the answer for yearly reporting.

**Event-specific configurations**

1. News channel publishes information about upcoming event with the expected attendance.
2. EcoTraffic receives the event information via integration with news channel.
3. EcoTraffic automatically categorizes the event by the attendance.
4. EcoTraffic analyzes historical traffic patterns from similar events.
5. EcoTraffic retrieves public transport schedules via microservice using getScheduleByStreet and getScheduleByLine operations.
6. EcoTraffic generates event-specific configuration recommendations for traffic lights and roads.
7. The UAM accesses to EcoTraffic and reviews the suggestions proposed and decides to accept or to reject them.
8. EcoTraffic writes into a log file the answer for yearly reporting.

**System monitors traffic during special event (forse da togliere, altrimenti si deve modificare l’architettura)**

1. EcoTraffic detect the presence of a special event happening today.
2. Traffic sensors detect traffic flow.
3. Traffic sensors publish data to message bus.
4. EcoTraffic receives sensor data from message bus.
5. EcoTraffic retrieve the data in the area near the event.
6. EcoTraffic implements temporary traffic light timing changes if needed.
7. EcoTraffic logs all adjustments and their effectiveness
8. EcoTraffic updates event-specific configuration data based on observations for future similar events

**Citizen views public traffic reports**

1. Citizen accesses EcoTraffic public portal.
2. EcoTraffic presents options for viewing reports.
3. Citizen selects the preferred option.
   1. Citizen selects "Daily Traffic Reports" and choose the date and time.
      1. EcoTraffic retrieves daily report data from database service.
      2. EcoTraffic displays report showing:
         1. Average traffic flow on main roads.
         2. Visualization of peak congestion periods.
         3. List of actions taken automatically.
         4. Traffic prediction for tomorrow.
   2. Citizen selects "Yearly Reports" option.
      1. EcoTraffic displays yearly report options.
      2. EcoTraffic retrieves yearly report data from database service.
      3. EcoTraffic displays comprehensive report showing:
         1. Suggested actions that were accepted.
         2. Suggested actions that were rejected.

**UAM views optimization to approve or reject**

1. UAM access to EcoTraffic public portal.
2. EcoTraffic displays all the suggested changes not already approved or rejected.
3. The UAM analyse each suggestion and decide to approve or reject choosing an option displayed.
4. EcoTraffic writes into a log file the answer for yearly reporting.
   * 1. **Use case diagrams (to modify)**
        1. **Traffic light duration adjustment**

Actors: Sensors, Traffic lights, Database.

Entry condition: After sensors measures new crossing times, new data arrives on the bus.

Flow of events:

* EcoTraffic reads data from the message bus.
* EcoTraffic stores in the database the data received.
* EcoTraffic compares the crossing times of the roads in the crossings.
* If EcoTraffic detects traffic load imbalance, the system computes the green light duration to optimize vehicle flow in the more congested direction.

Exit condition: The system sends the adjusted times to the traffics lights control system.

* + - 1. **Daily Analysis and Optimization**

Actors: Webservice, Database

Entry condition: It’s ten past midnight.

Flow of events:

* EcoTraffic queries the database to get all the crossing times measured in the day before.
* EcoTraffic analyses the data retrieved to obtain traffic pattern.
* EcoTraffic analyses the traffic pattern to find possible optimization in terms of one-way roads and traffic lights configuration.
* EcoTraffic retrieves public transport schedules via microservice using getScheduleByStreet and getScheduleByLine operations.
* EcoTraffic tries to find better schedules for the public transport line.

Exit condition: The system write into the database the possible optimization found.

* + - 1. **Traffic zone optimization**

Actors: Urban area manager, Public Transport Microservice, Database

Entry condition: UAM accesses EcoTraffic public portal and decide to request optimization of a certain area.

Flow of events:

* EcoTraffic queries the database to obtain the crossing times of the zone indicated by the UAM
* EcoTraffic analyses the data to find possible optimization in terms of one-way roads and traffic lights configuration.
* EcoTraffic retrieves public transport schedules via microservice using getScheduleByStreet and getScheduleByLine operations.
* EcoTraffic tries to find better schedules for the public transport line.
* The system sends the new scheduling and configurations proposal to the UAM for approval.
* The system waits for the answer.
* When the decision is taken, the answer is written into a log file.

Exit condition: Successful write of the log.

* + - 1. **Event-specific configurations**

Actors: News channel, UAM, Public transport microservice

Entry condition: News channel publishes information about upcoming events.

Flow of events:

* EcoTraffic categorizes the scale of the event by attendance.
* EcoTraffic investigates the database for historical traffic patterns and past decision tskrn in similar events in previous log files.
* EcoTraffic retrieves public transport schedules via microservice using getScheduleByStreet and getScheduleByLine operations.
* Using the collected data, EcoTraffic generates event-specific configurations that are optimized in terms of one-way roads, traffic lights configuration and public transport schedules.

Exit condition: The system write into the database the possible optimization found.

* + - 1. **System monitors traffic during special event**

Actors: Sensors, traffic lights, Public transport microservice

Entry condition: EcoTraffic detects an event on that specifi day from the News channel.

Flow of events:

* The system looks for historical traffic patterns in the area and at the time before/during/after the event.
* The system retrieves public transport schedule via microservice.
* The system periodically collects data from the bus.
* The system periodically compute an optimization for the traffic lights on the roads impacted by the event using the collected data.
* The system updates event-specific configuration data based on observations for future similar events.
* The system logs all adjustments and their effectiveness.

Exit condition: Successful write of the log.

* + - 1. **Citizen views public traffic reports**

Actors: Citizen

Entry condition: Citizen accesses EcoTraffic public portal.

Flow of events:

* The system presents options for viewing reports.
* The citizen selects the type of information that they want to be displayed.
* The system retrieves daily or yearly data from database service.
* The system elaborates the data to facilitate interpretation.
* The system displays a report with the elaborated data.
* The citizen chooses between seeing more information or exiting the portal.

Exit condition: The citizen exits the portal.

* + - 1. **UAM views optimization to approve or reject**

Actors: UAM

Entry condition: UAM accesses EcoTraffic public portal and decide to view the optimization waiting for approval.

Flow of events:

* EcoTraffic queries the database for the optimization proposals waiting for approval.
* EcoTraffic elaborates the data to facilitate interpretation.
* EcoTraffic displays all the decision to be taken.
* EcoTraffic waits for the UAM to answer to each request.
* Every time a decision is taken EcoTraffic stores it into a log file.

Exit condition: All the decision have been taken.

* 1. **Domain assumption**

1. The sensor infrastructure works correctly and with low latency 24/7.
2. The traffic lights are not faulty and set their state correctly in time from the ET system.
3. Drivers behave accordingly to the traffic light state.
4. No car can obstruct the passage in the crossing no matter the reason.
5. Events planners always report to the news channel up to date events in the city.
6. The public Transport Microservice always returns the right timetable given a line or the name of a street.
   1. **Requirements**
      1. **Functional Requirements (to modify put id)**

* In any circumstance the system must not allow two orthogonal traffic lights to be green at the same time.
* The system must modify the duration of the green lights to reduce traffic.
* The system shall process and aggregate traffic data to identify traffic flow patterns.
* The system should send adjustment commands to the traffic light control system.
* The system should be able to write to a log file what needed.
* The system shall generate optimization suggestions for one-way road and traffic light configurations.
* The system shall generate optimization suggestions for public transport schedules.
* The system shall continuously receive data from both the message bus and the news channel.
* The system must gather data from the microservice.
* The system must assess the potential traffic impact of planned events.
* The system shall generate event-specific suggestions for public transport adjustments.
* The system shall present suggestions to urban area managers for review.
* The system shall record and apply the acceptance or rejection of the proposal by the urban area managers.
* The system shall generate daily reports on average traffic flow.
* The system shall generate yearly reports on suggestions proposed and their outcome.
* The system shall publish reports for public access.

**2.4.2 Non-Functional Requirements**

* The system shall process sensor data in real time.
* The system should be available 24/7.
* The system shall implement traffic light adjustments in 15 seconds.
* The system shall generate reports within 1 hour after midnight.
* The system should maintain data consistency during communication with external systems.
* The system should be scalable regarding the addition of sensors and bus lines.

**2.4.3 Constraints (to modify, insert something to use the external services)**

* The setLight function must be atomical.
* The call to the microservice must be done using an API REST.