

Power EnJoy

DD

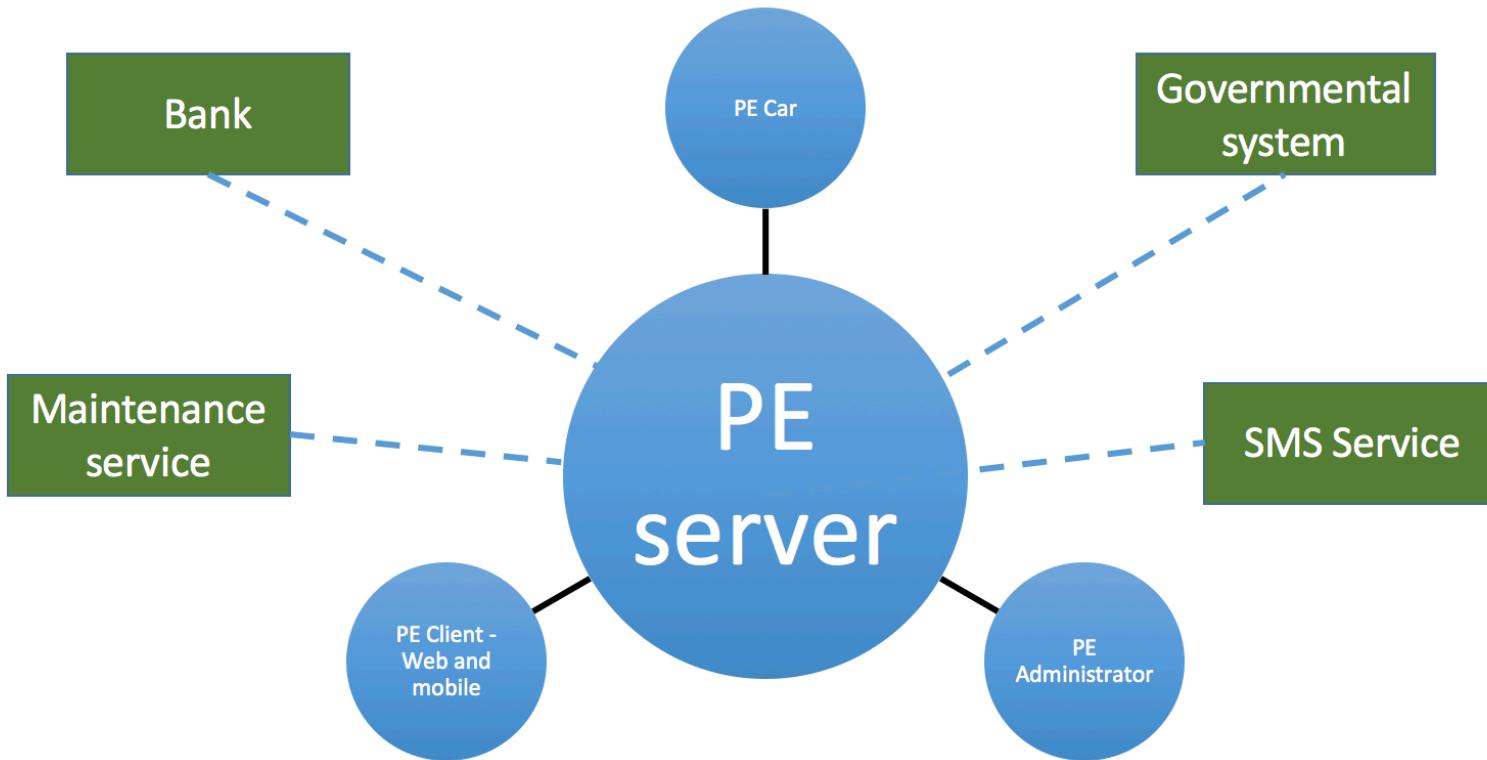
Design Document

Authors:

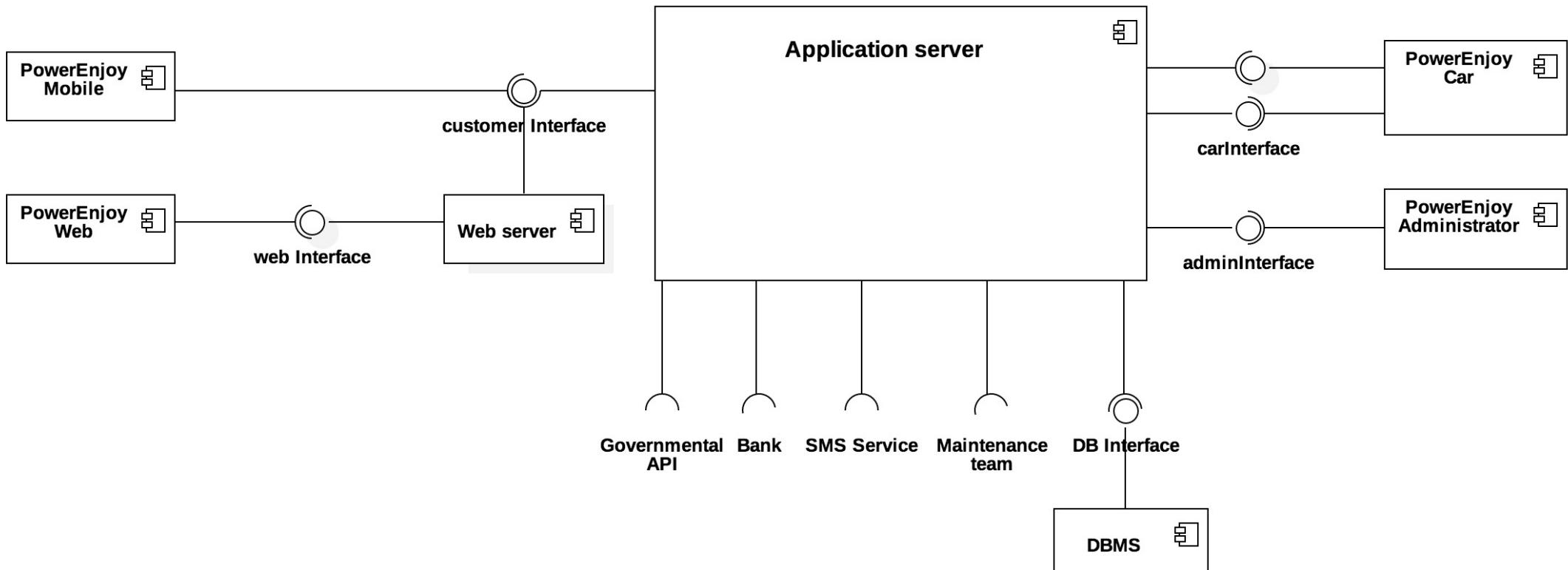
Andrea Battistello : 873795
William di Luigi : 864165

14/12/16

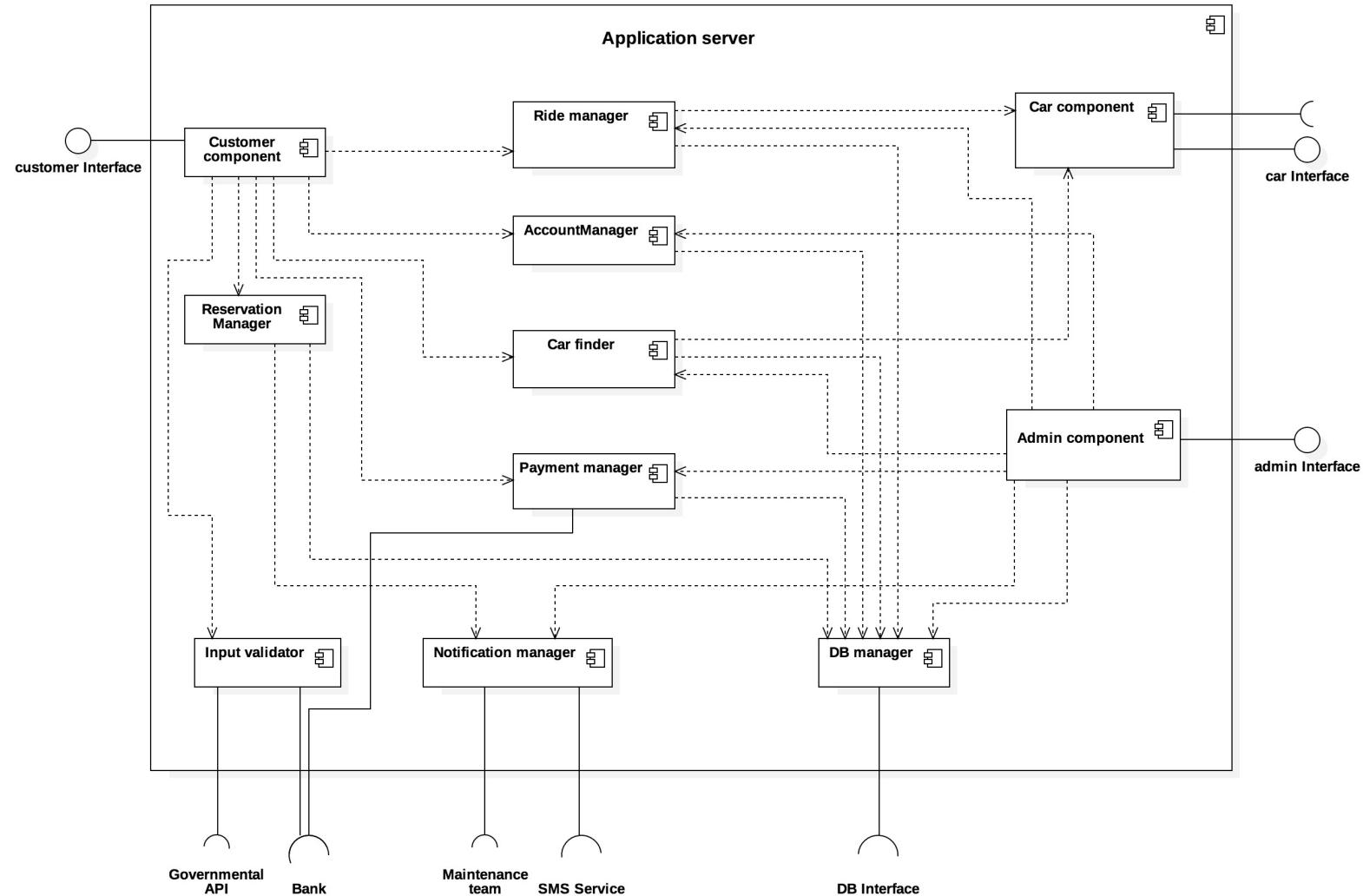
Overview



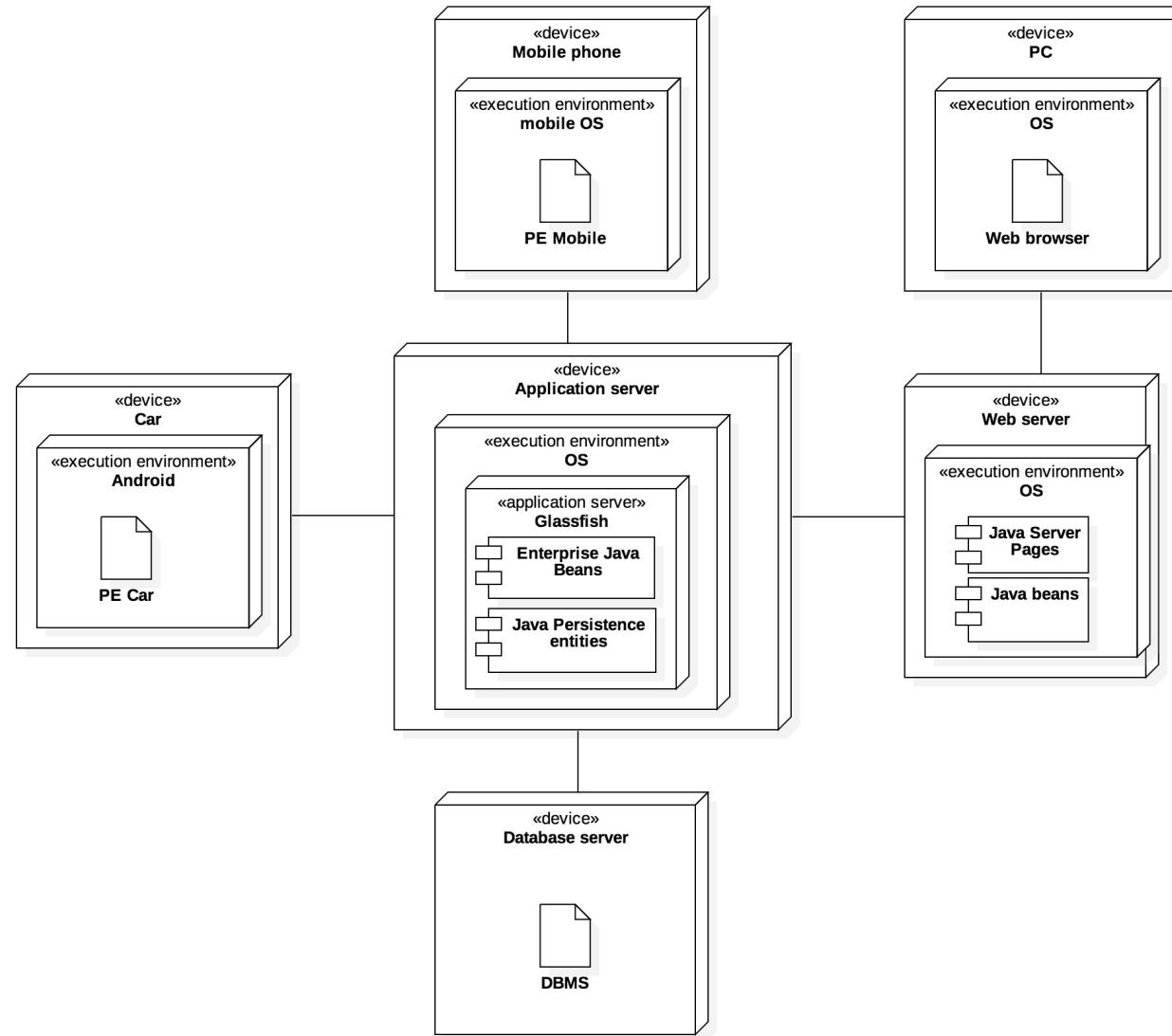
Component view



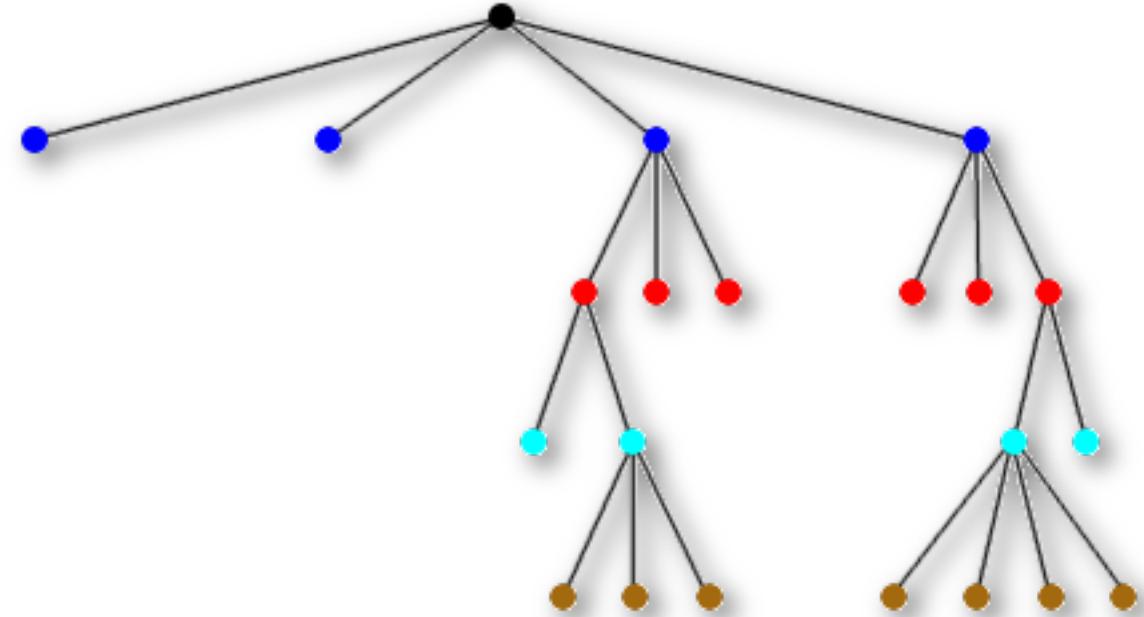
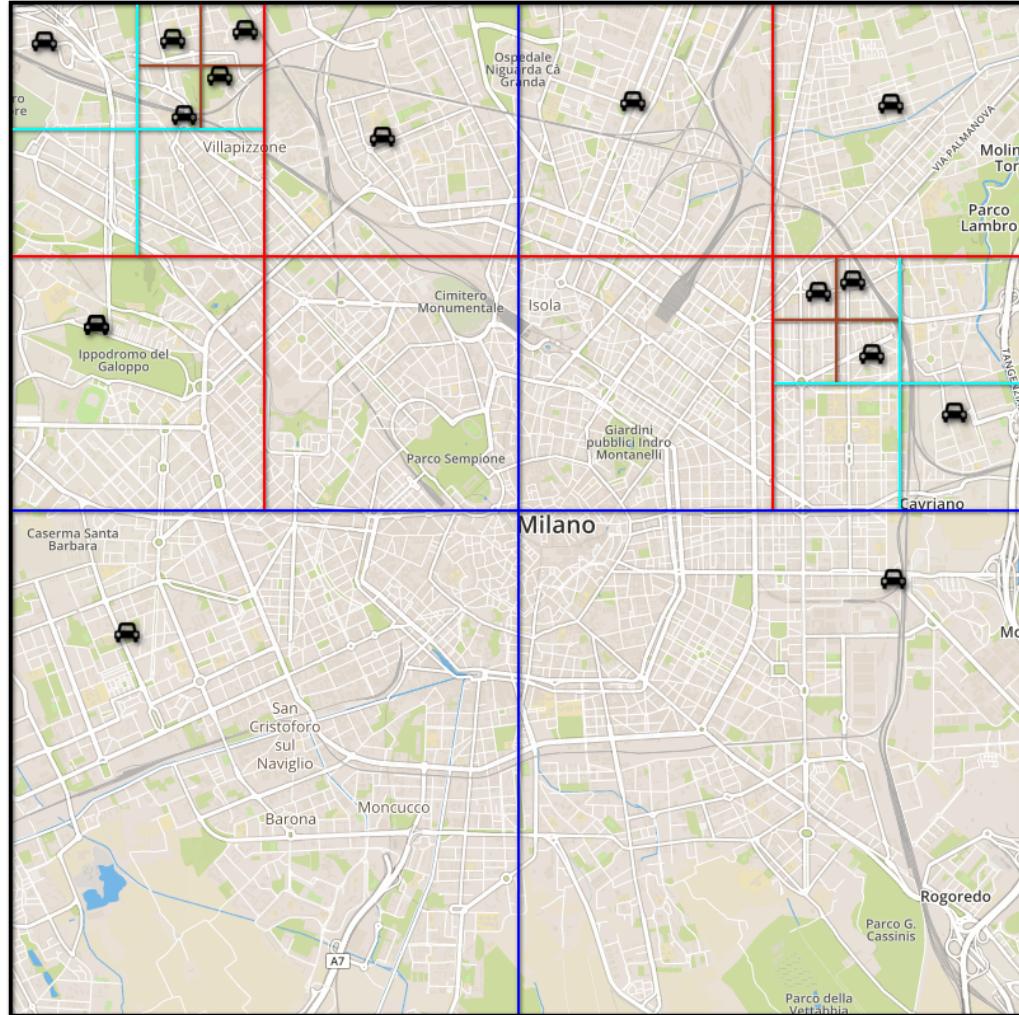
Application server



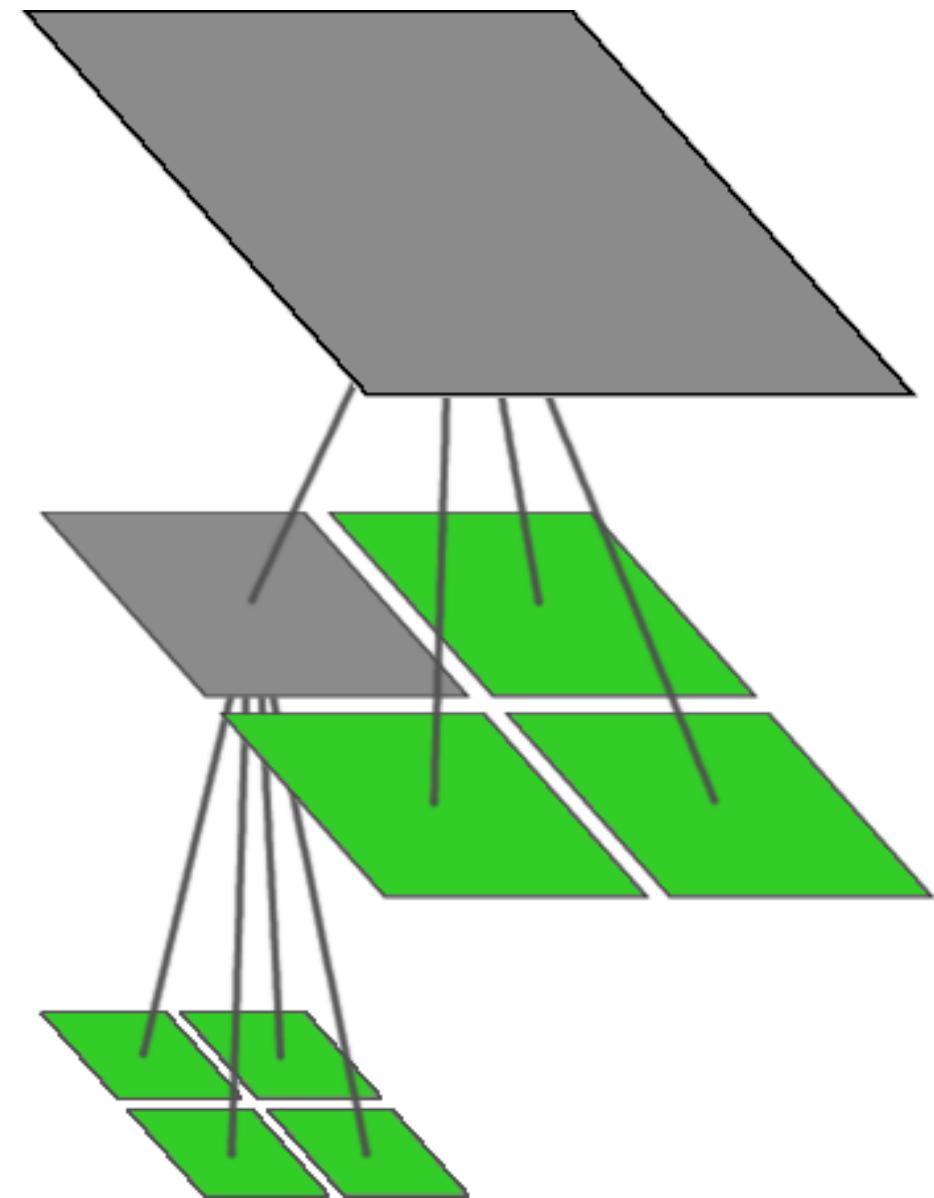
Deployment view



Quad trees



Insert



```
procedure QuadInsert(C, X)
    // Try to insert car C at node X in quadtree
    // By construction, each leaf will contain either
    // 1 or 0 cars
    if the subtree rooted at X has more than 1 car then
        determine in which child Y of node X the car C is in
        // Y is either the top left, top right, bottom left,
        // or the bottom right "quadrant"
        QuadInsert(C, Y)

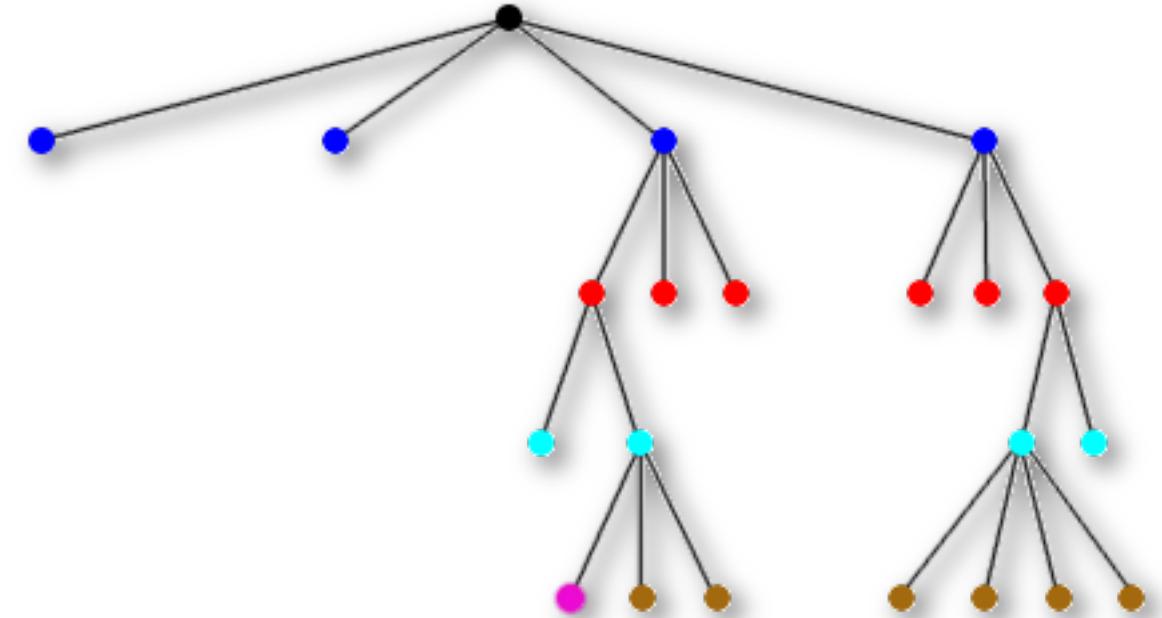
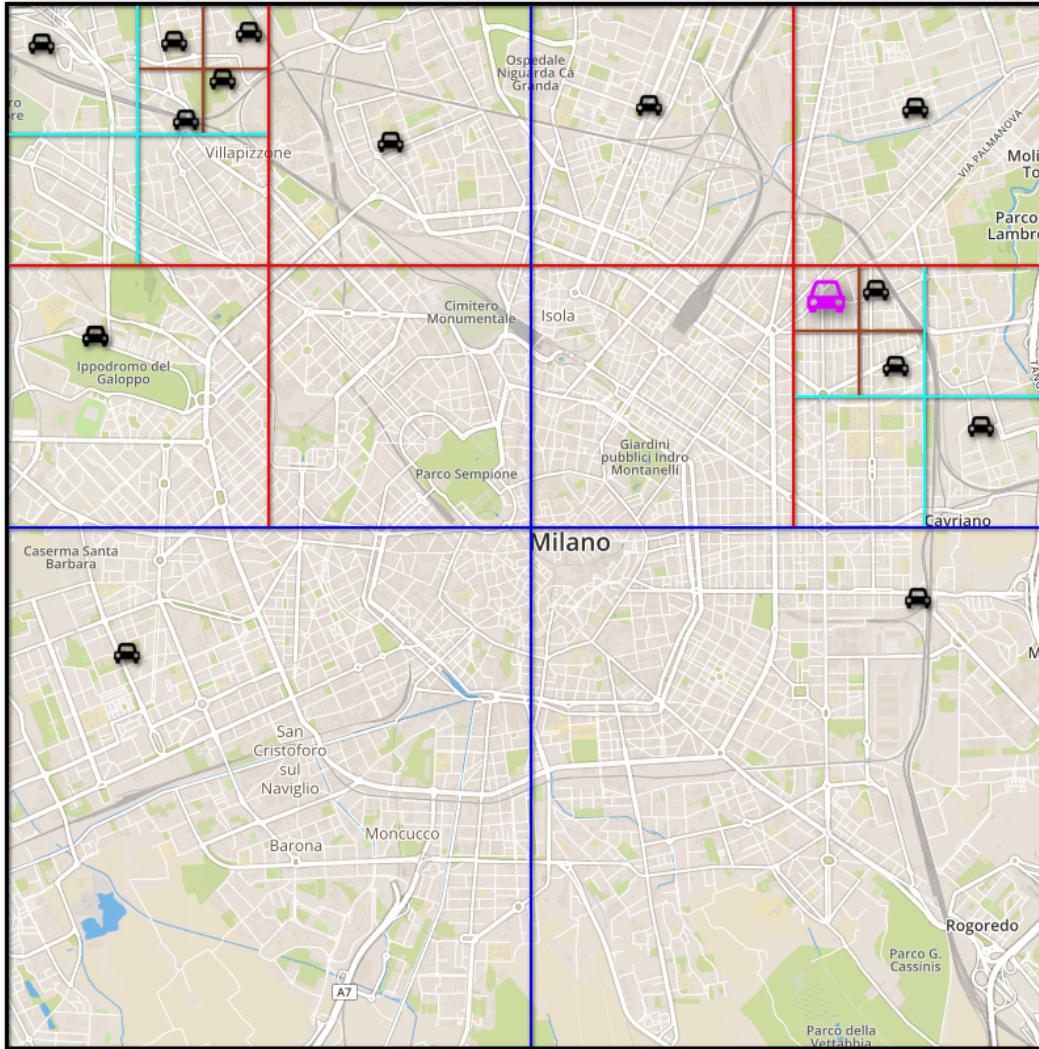
    else if the subtree rooted at X has exactly 1 car then
        // X is a leaf
        create four children for node X in the Quadtree
        // X is not a leaf anymore
        move the car in X into the child in which it lies

        let Y be child in which car C lies
        QuadInsert(C, Y)

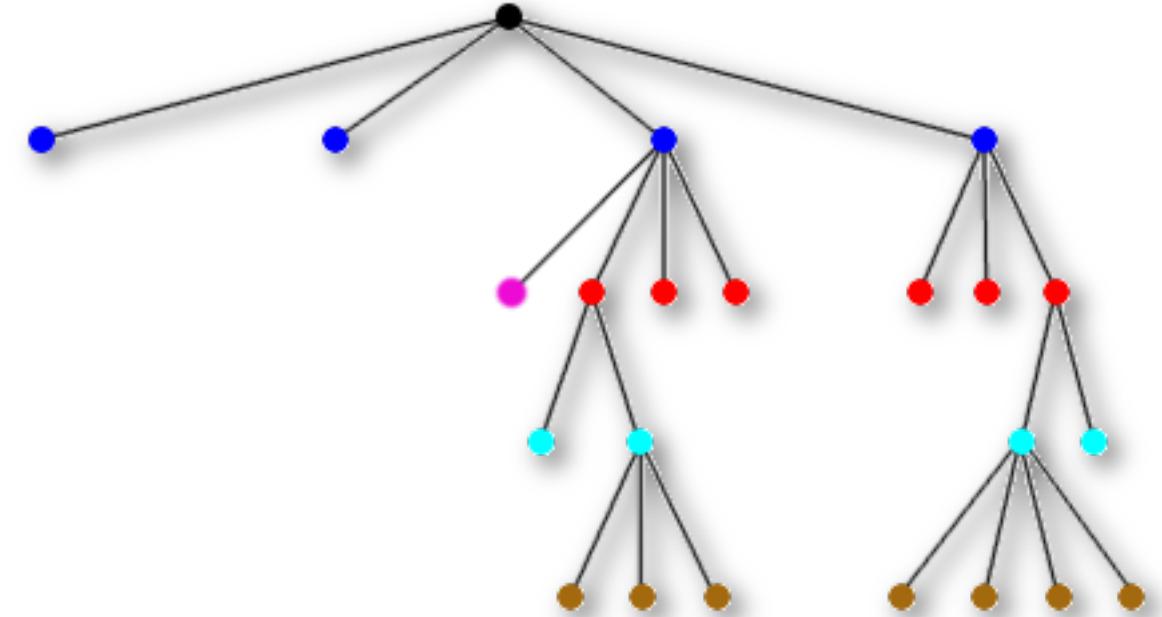
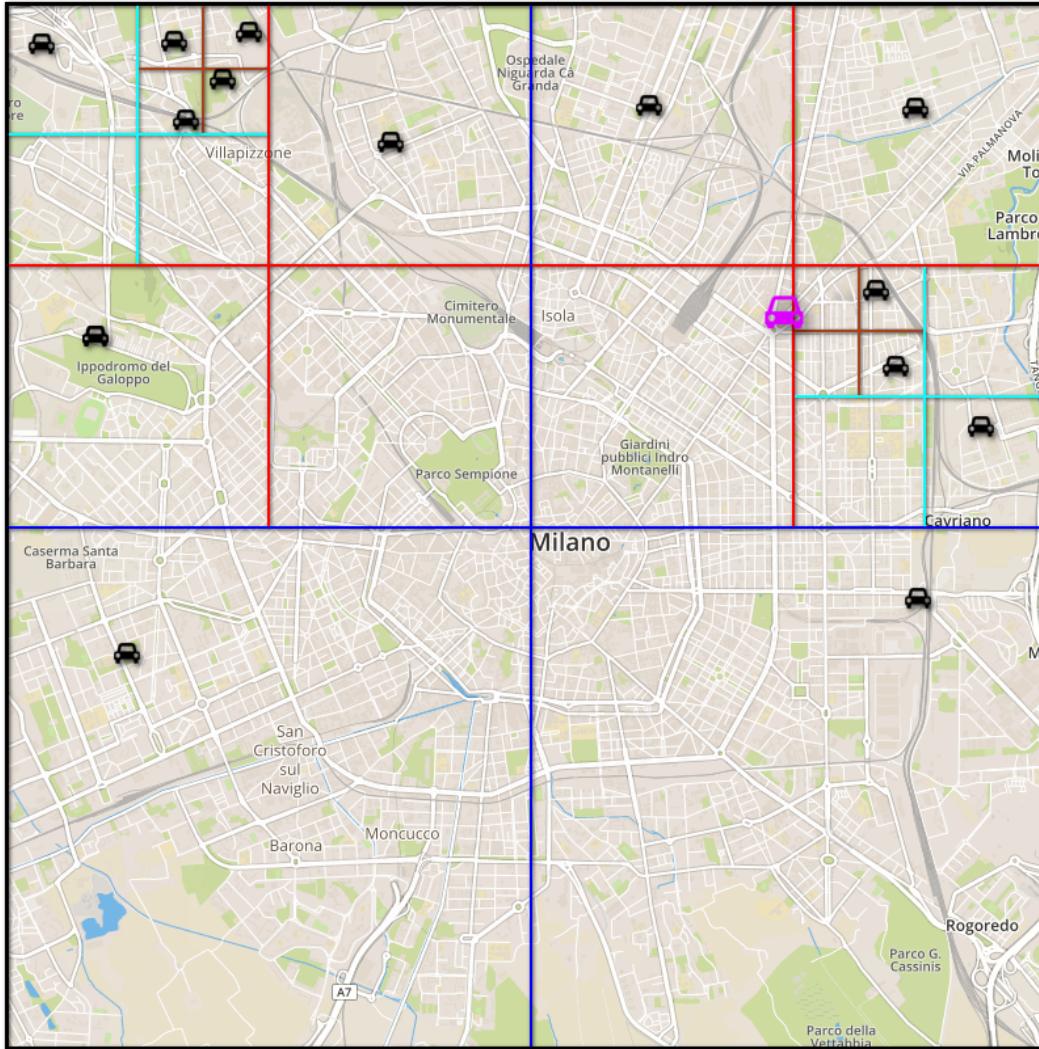
    else
        // X is a leaf
        store car C in node X
    endif

end
```

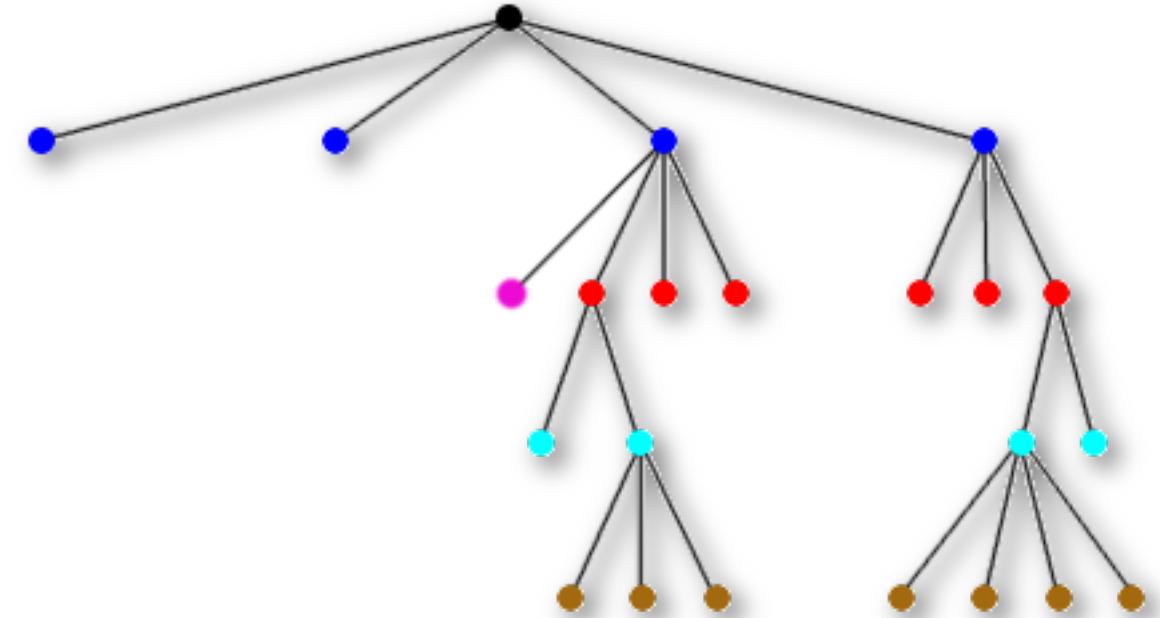
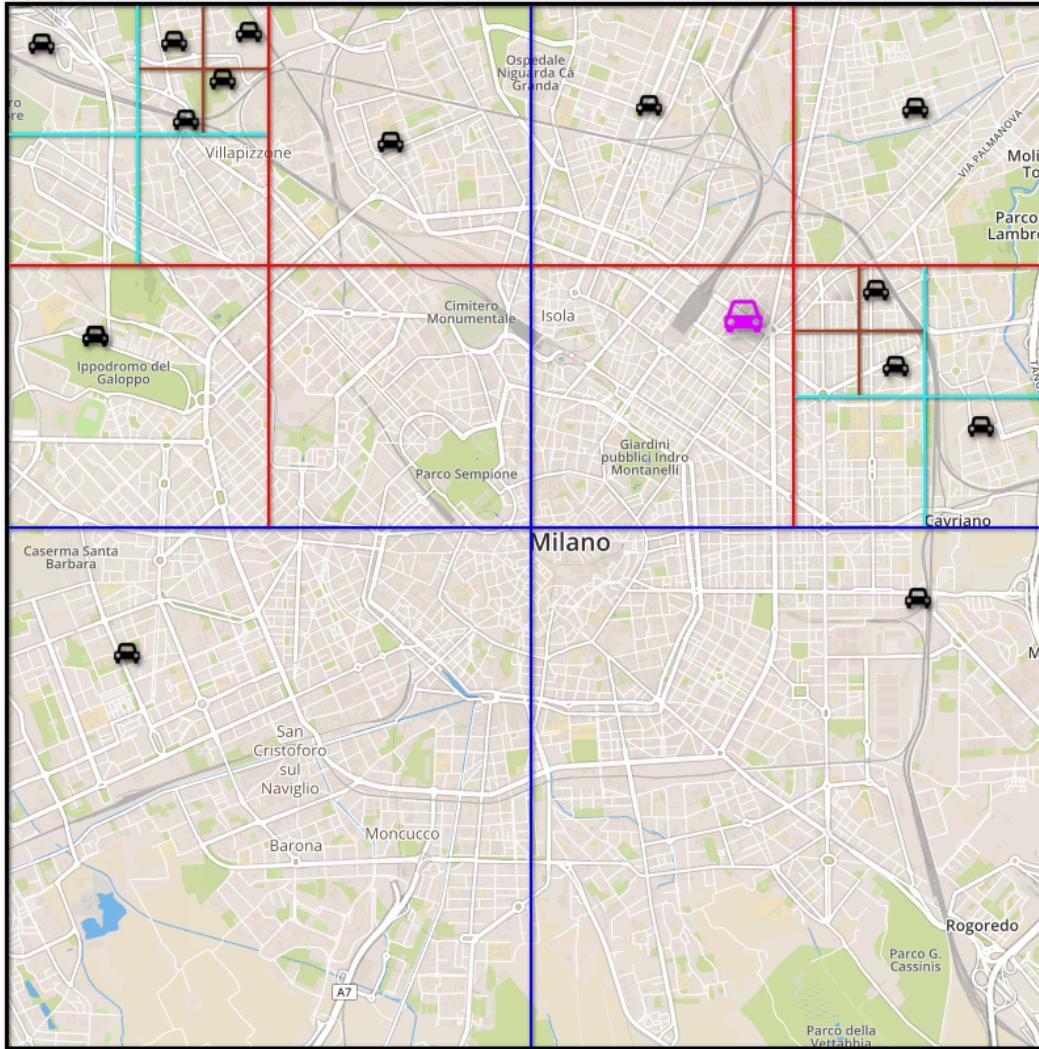
Update car position



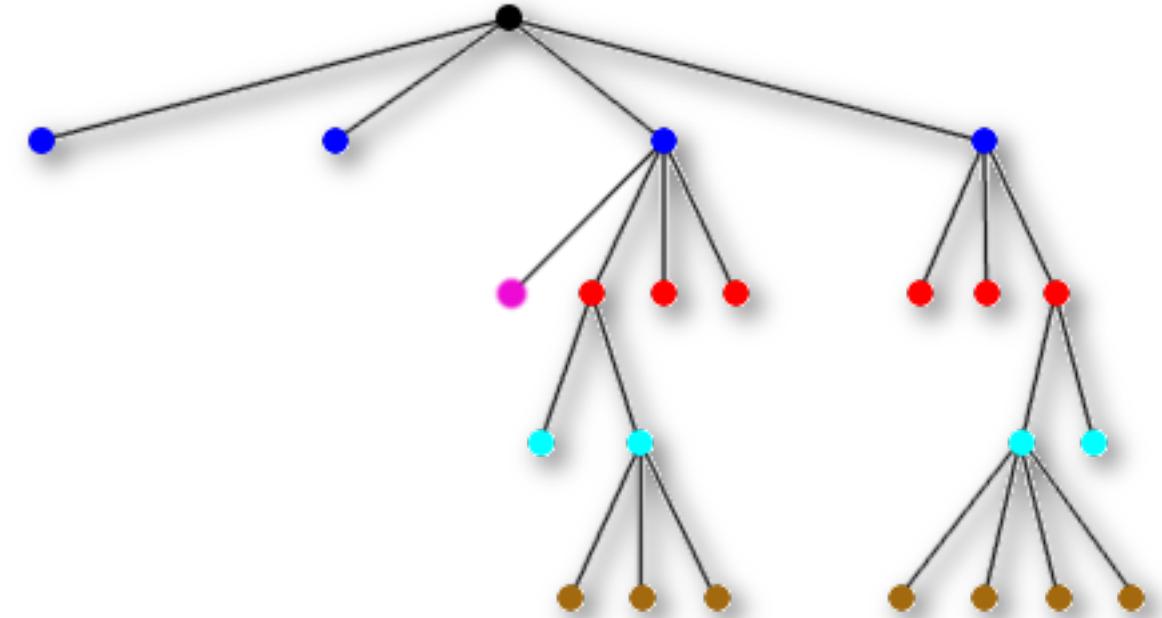
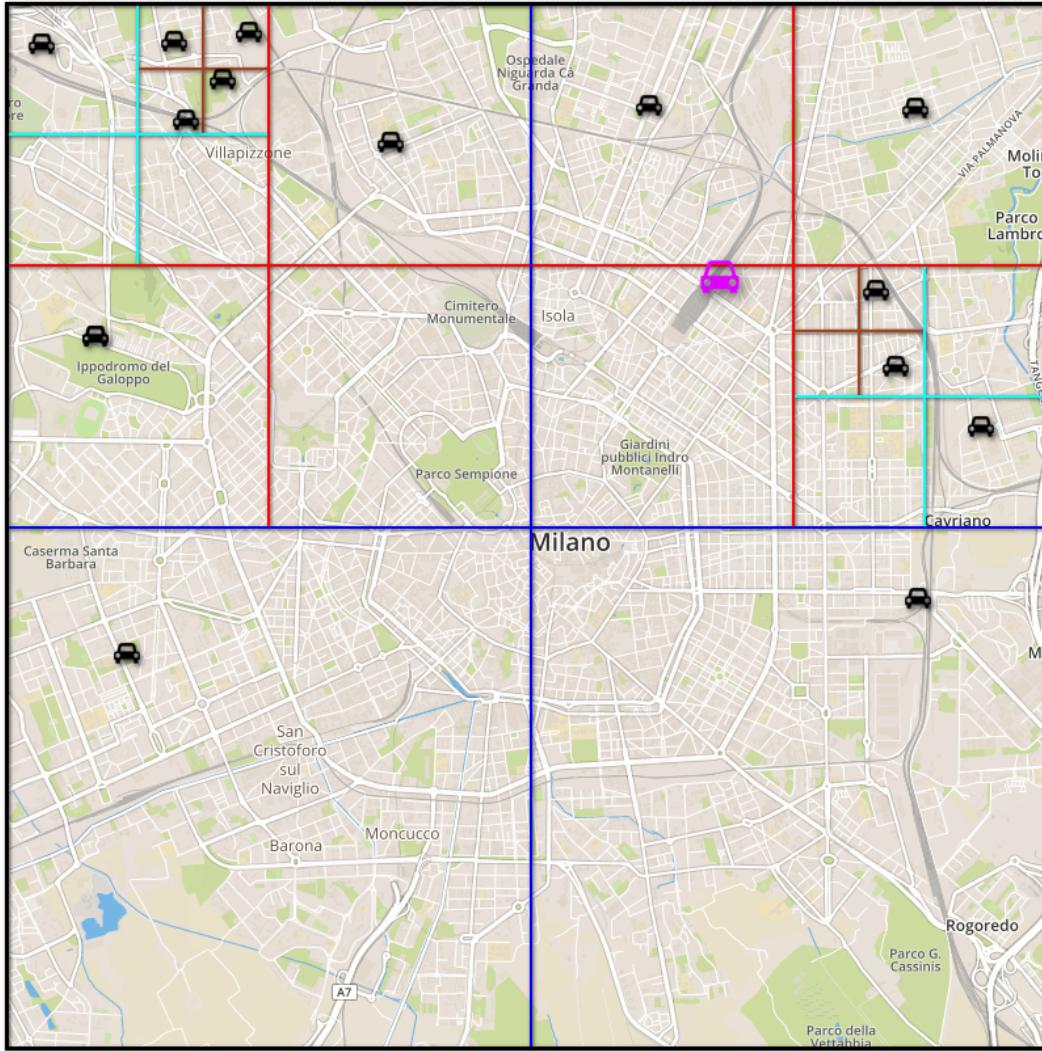
Update car position



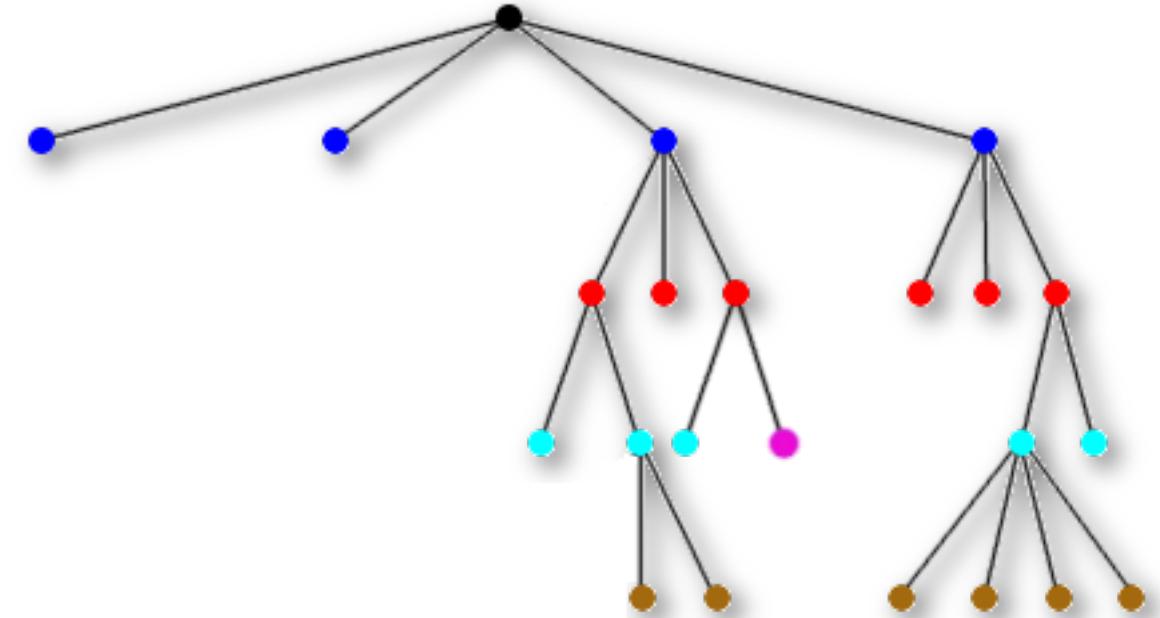
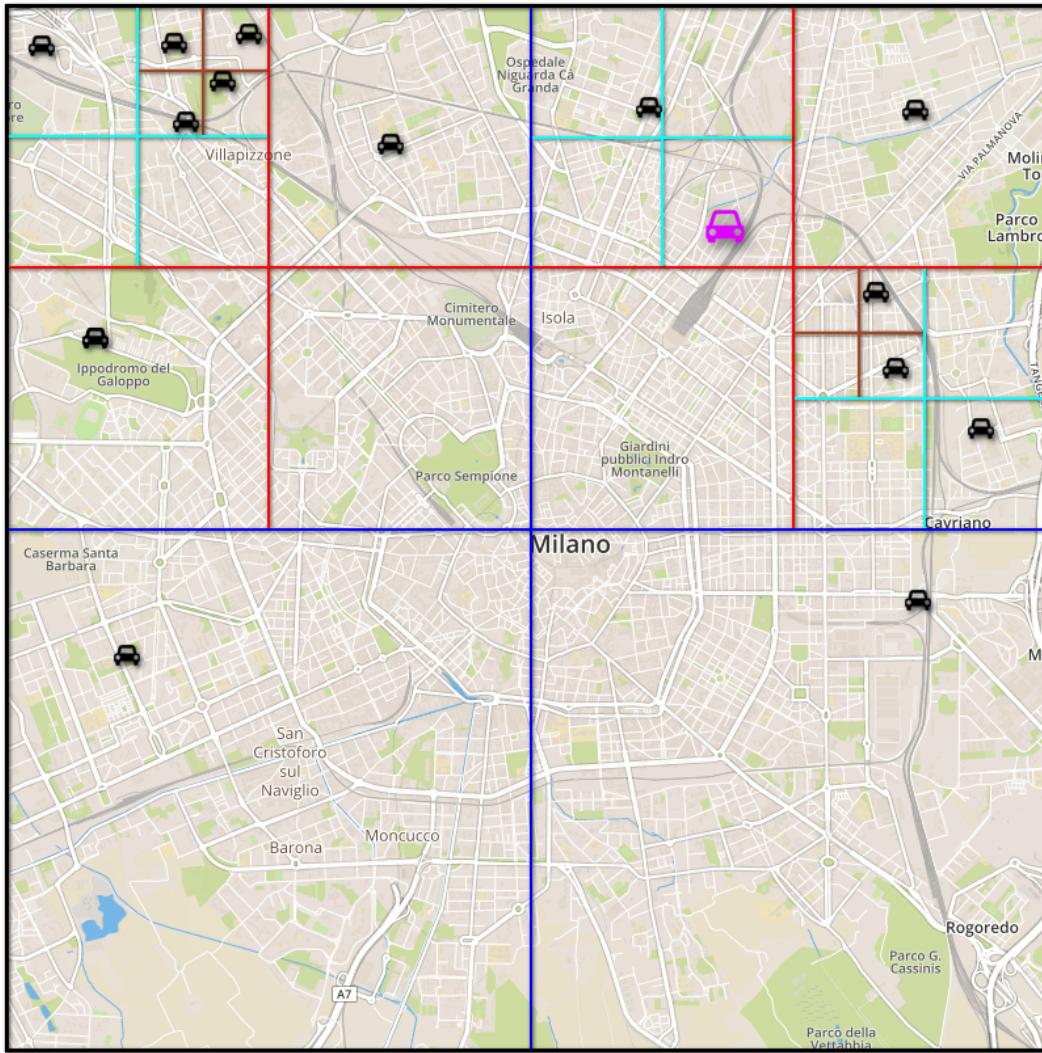
Update car position



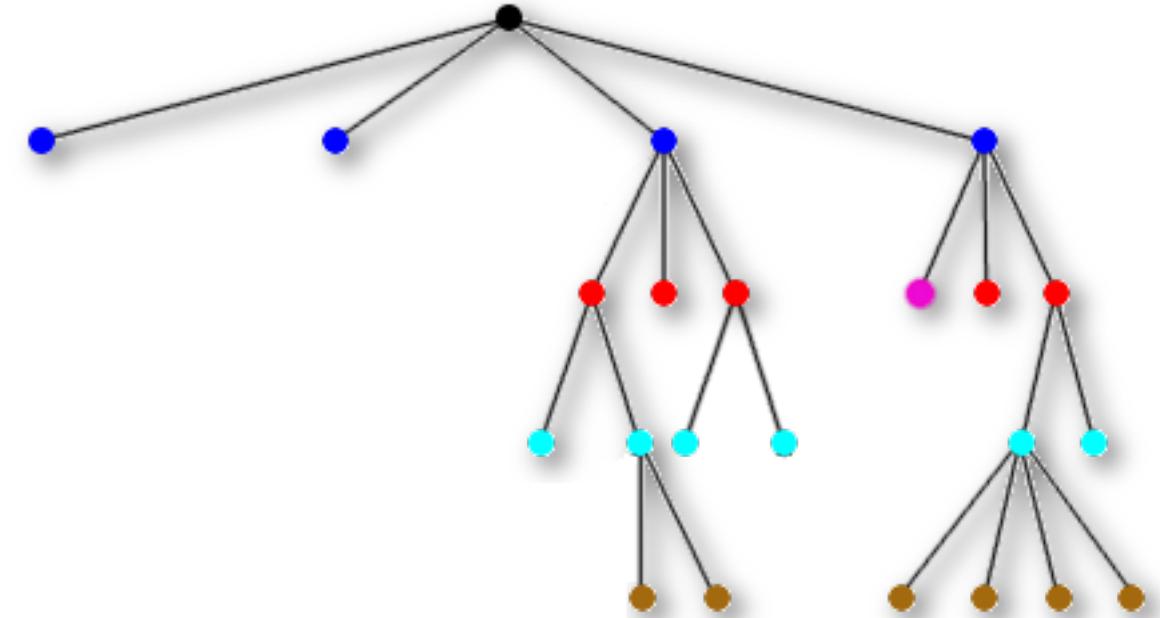
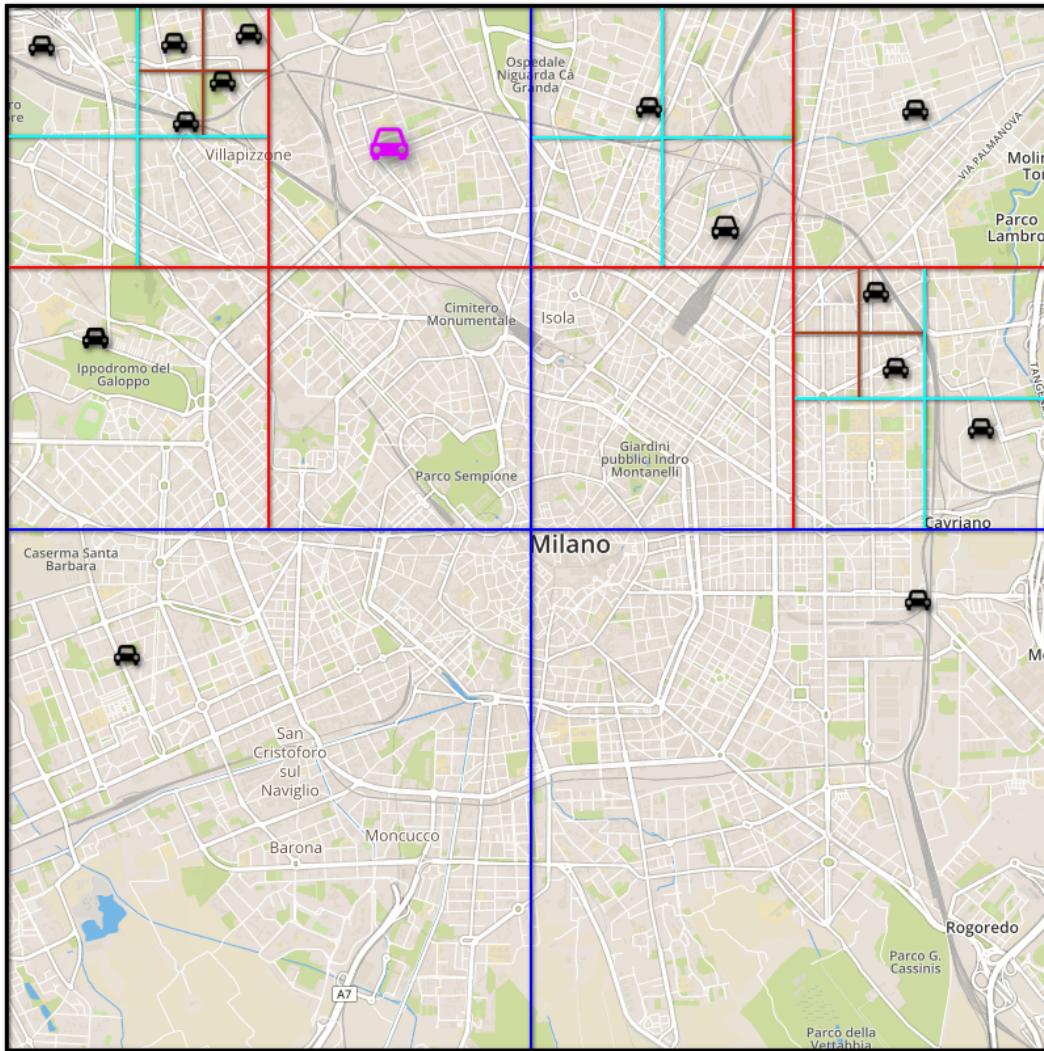
Update car position



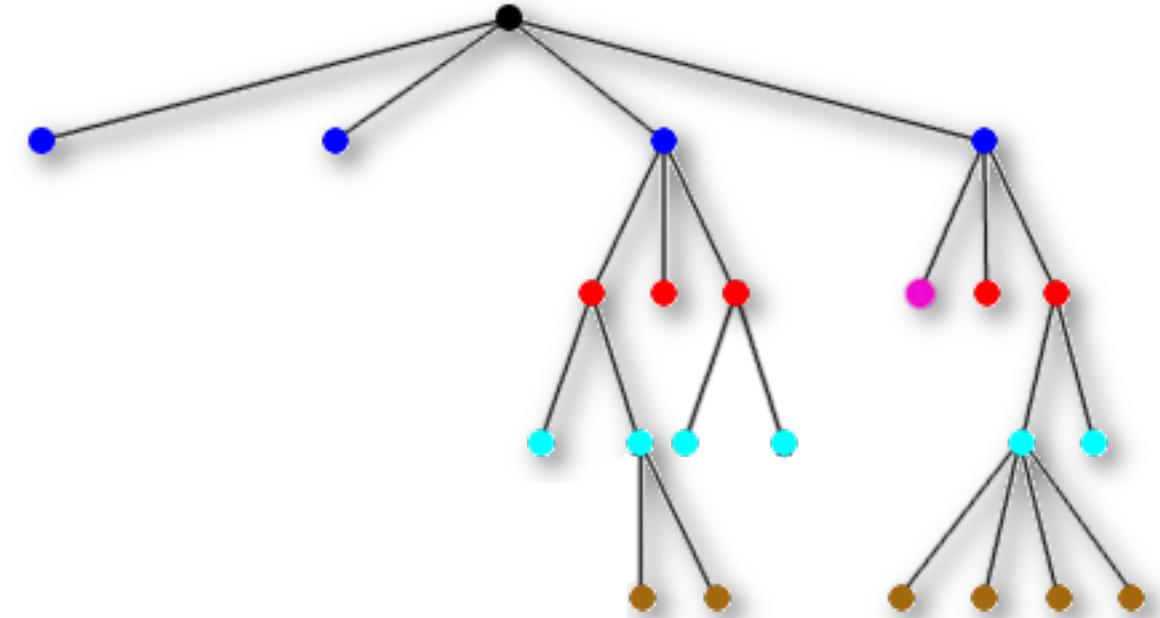
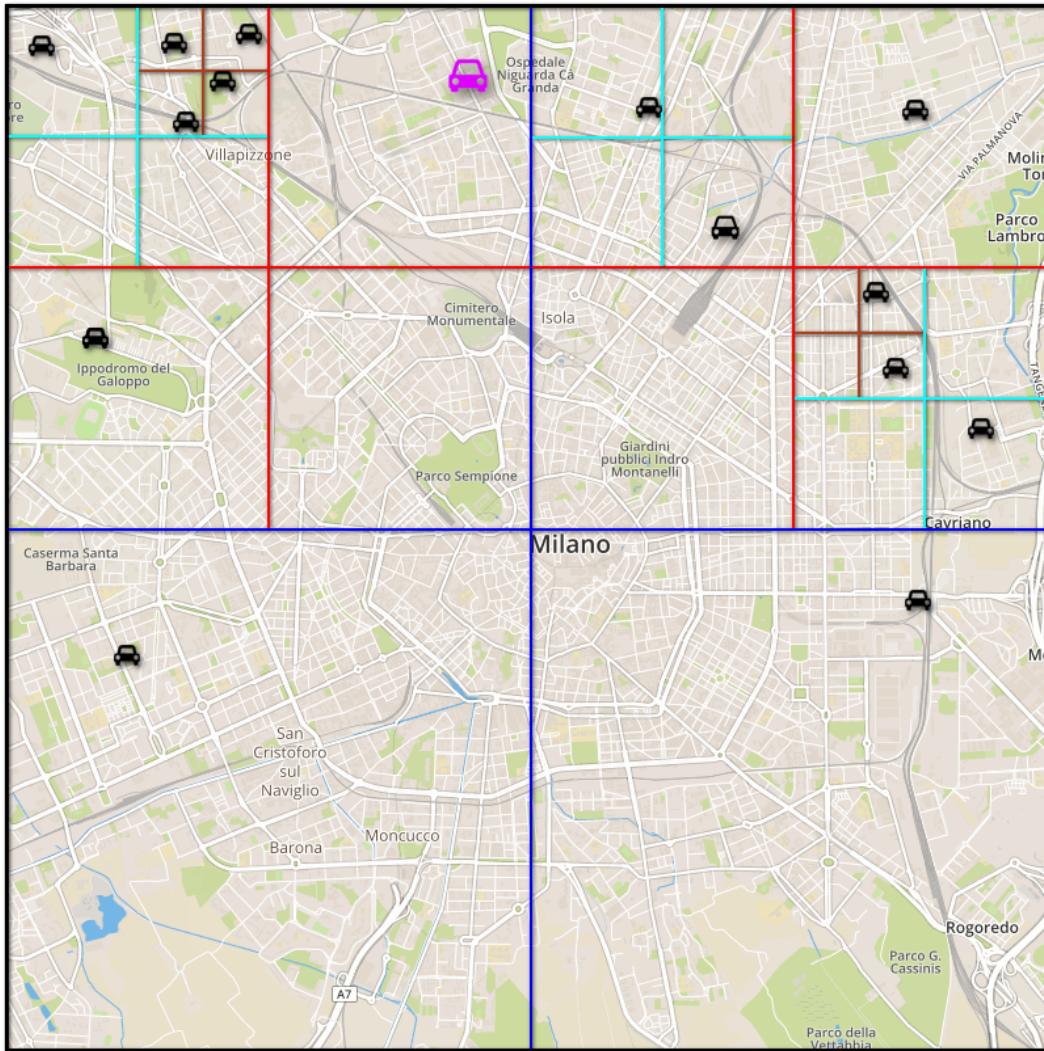
Update car position



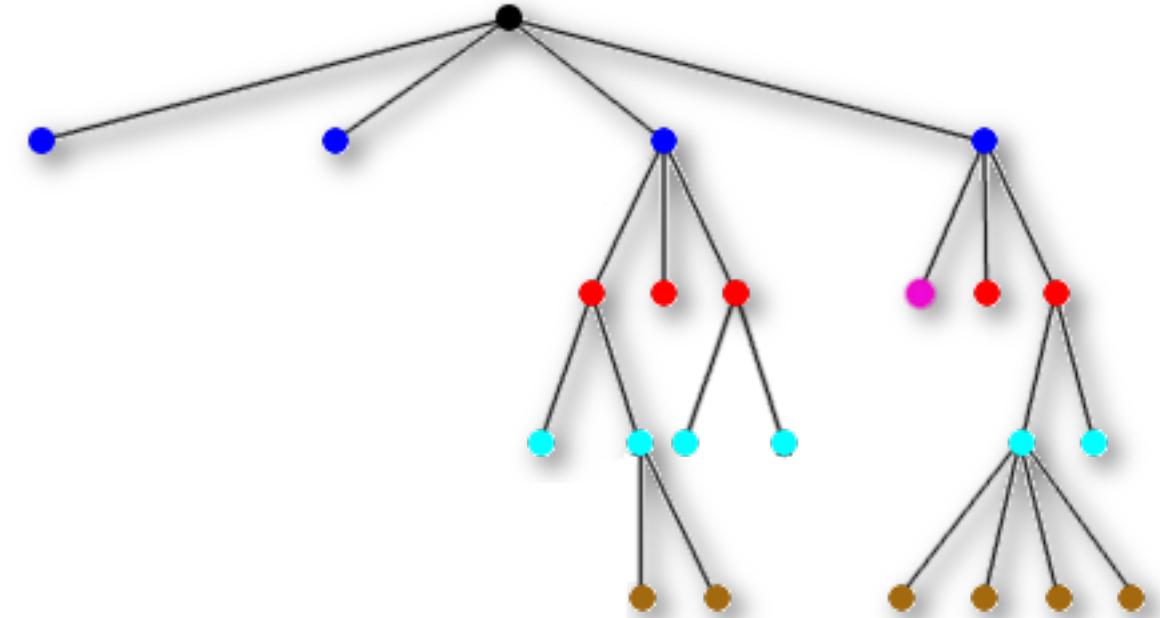
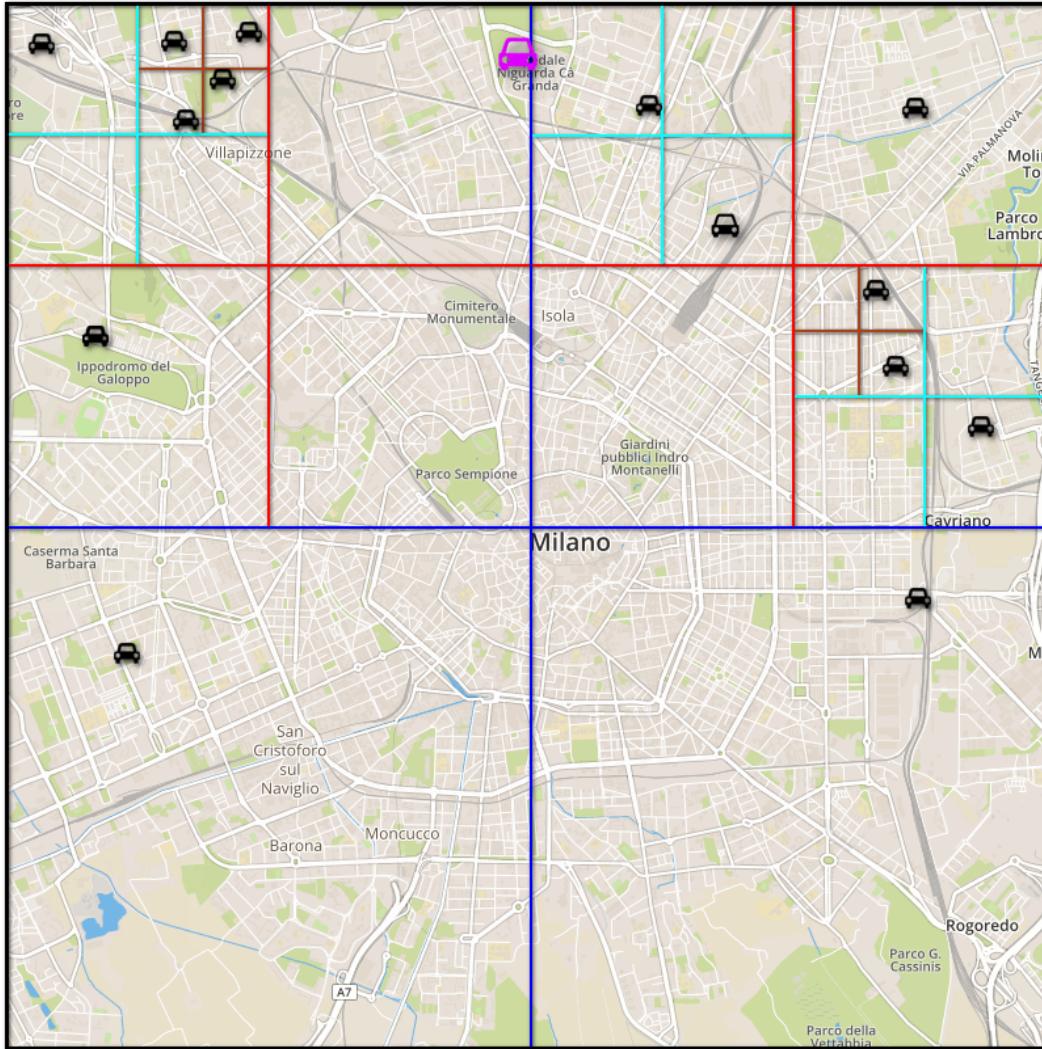
Update car position



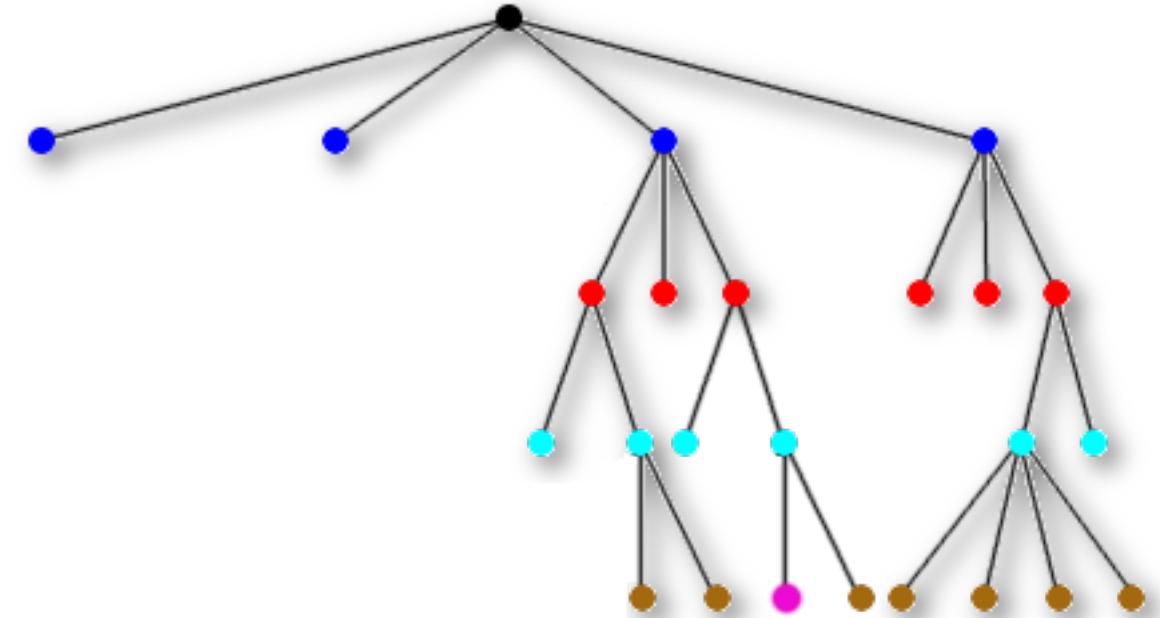
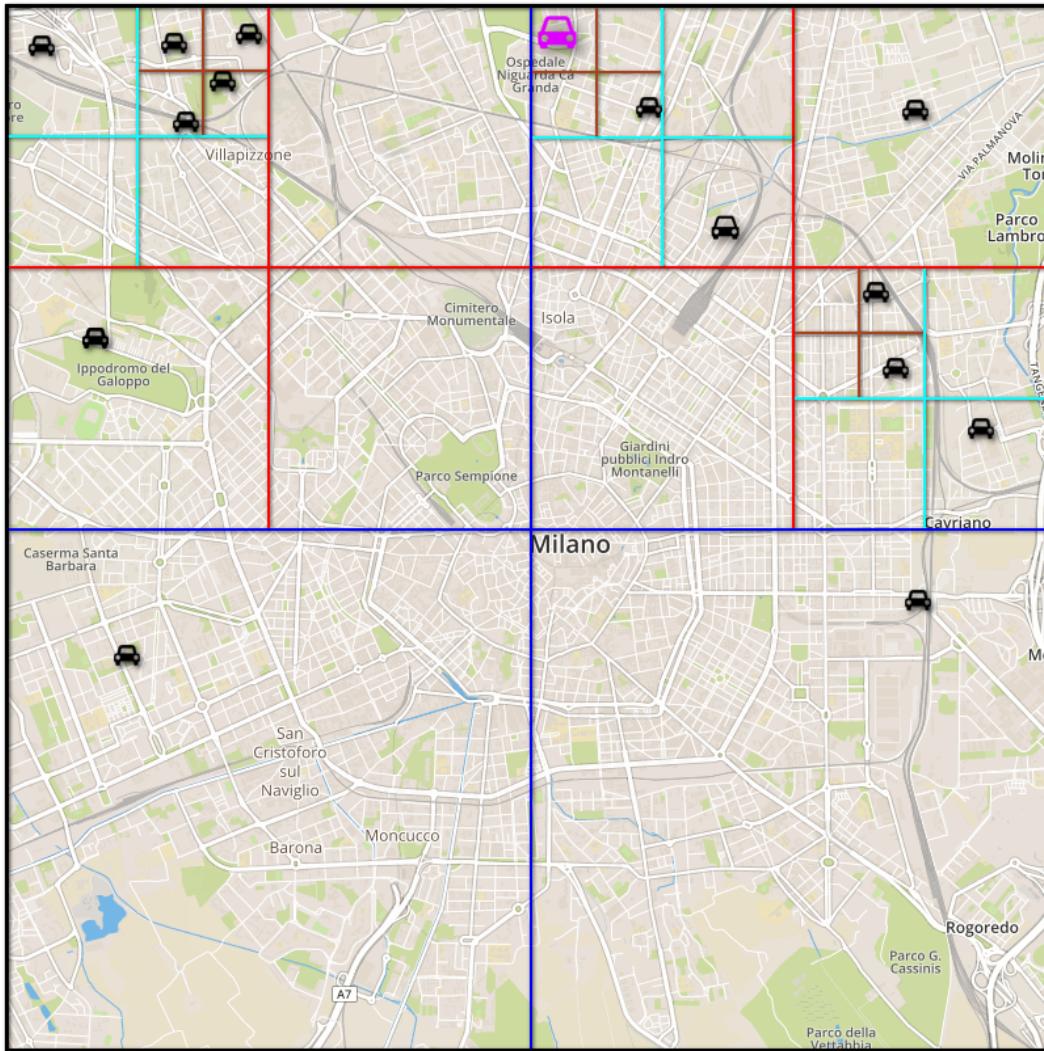
Update car position



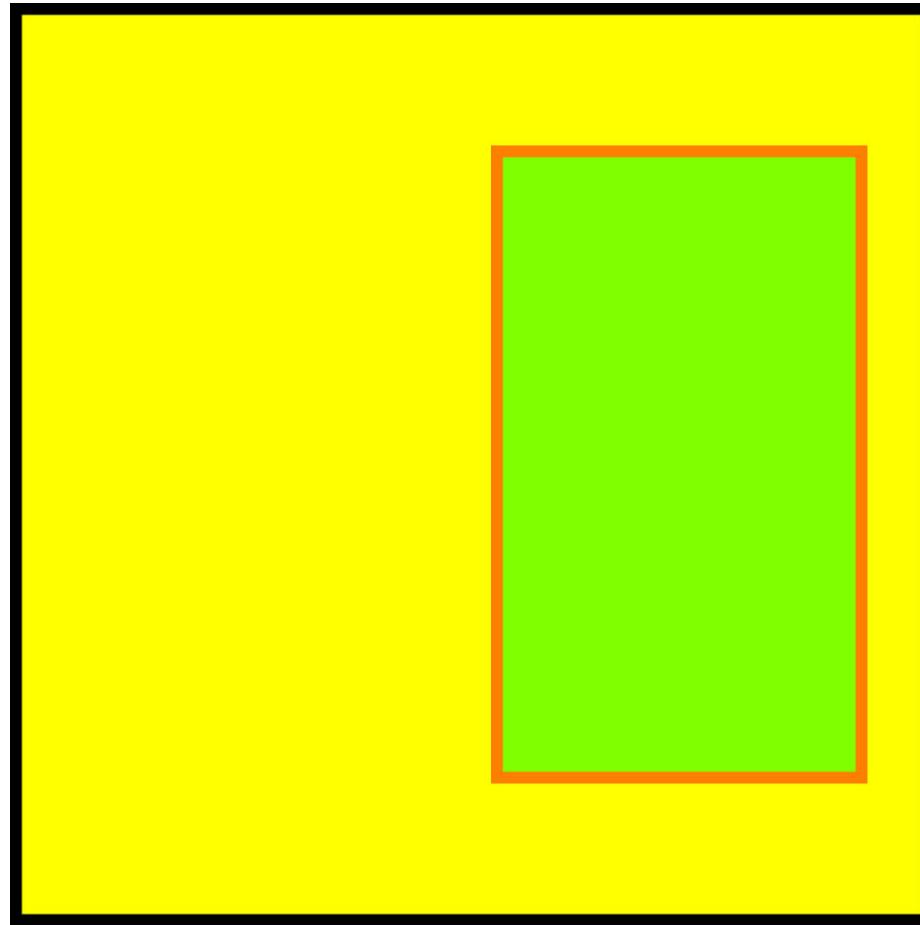
Update car position



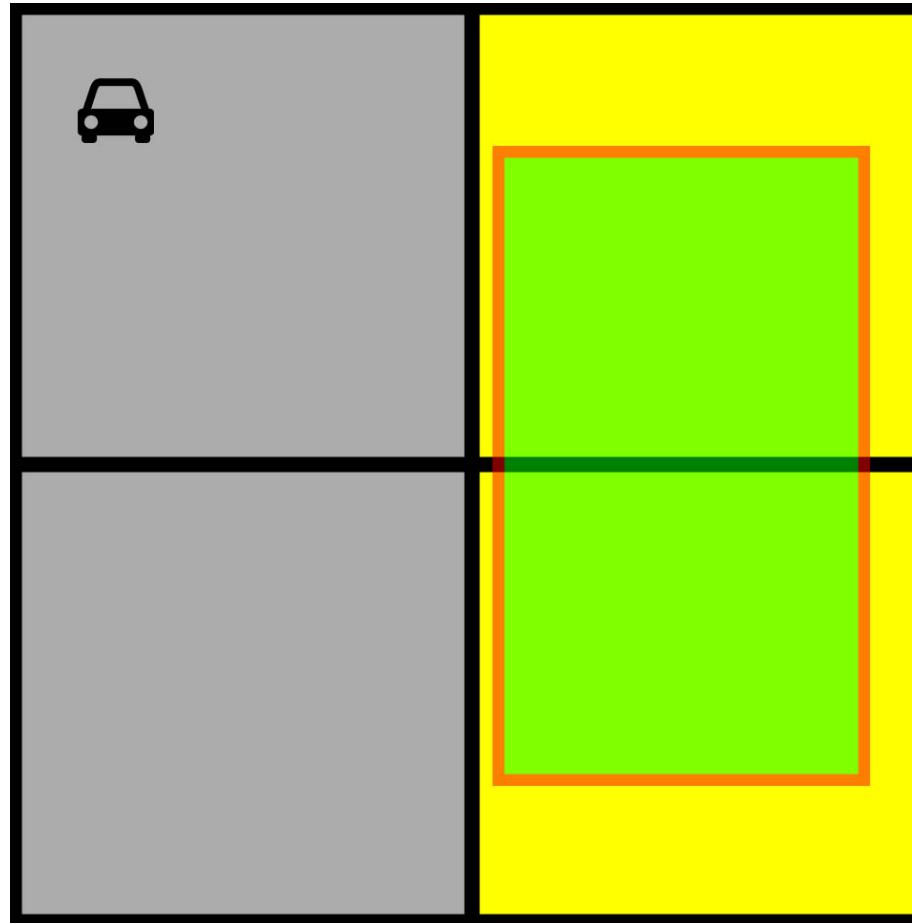
Update car position



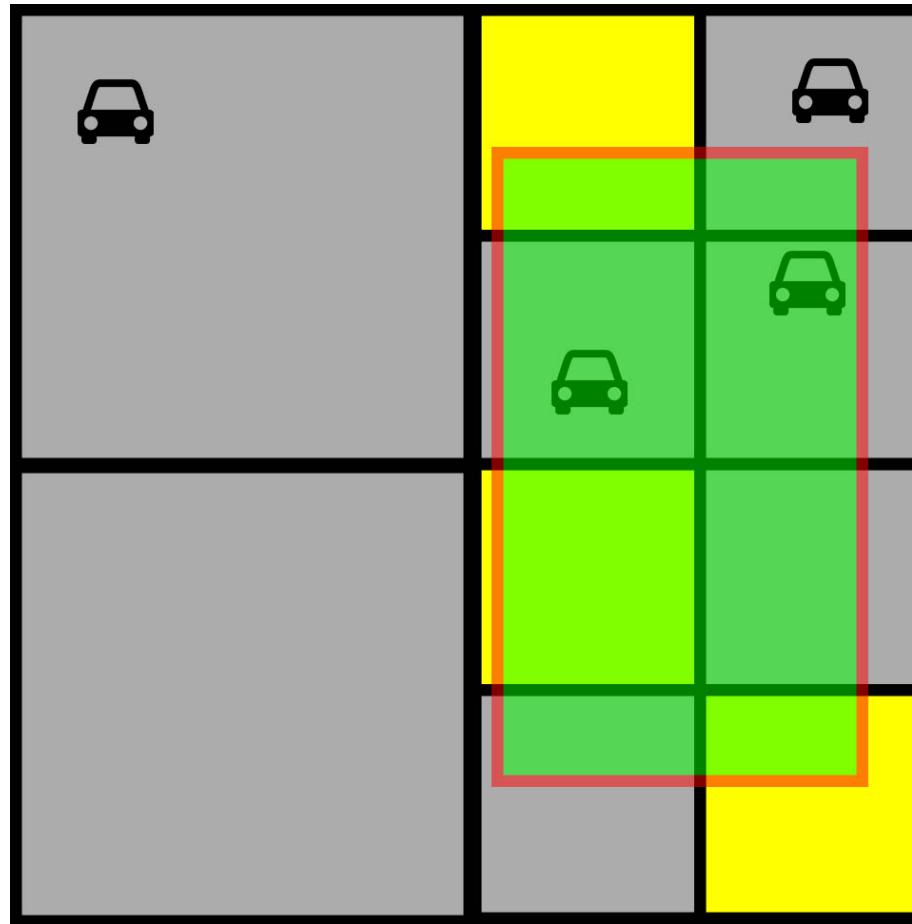
List all cars in a given area – Step 1



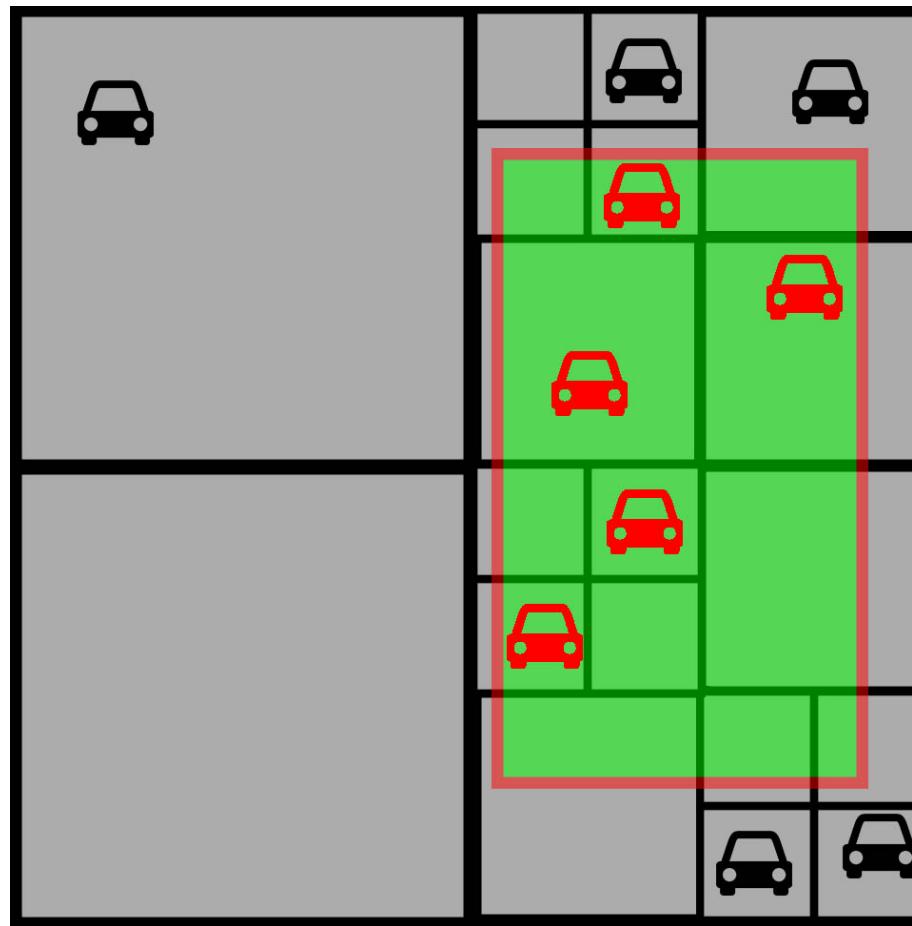
List all cars in a given area – Step 2



List all cars in a given area – Step 3



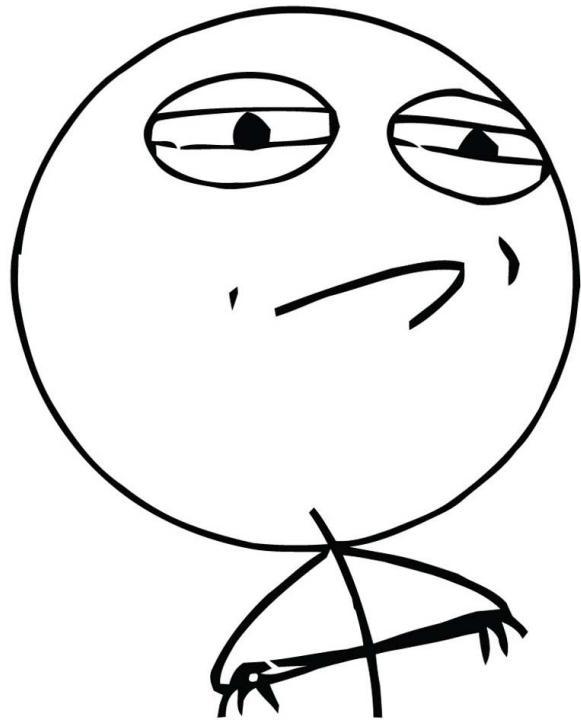
List all cars in a given area – Step 4



List all cars in a given area

```
procedure QuadList(S, X)
    // S is the “query square”, that is: the interesting area
    // X is the root node, initially is set to root
    answer = [] // empty list
    Y = S ∩ area(X) // intersection between S and the area of X
    if Y ≠ ∅ // non-empty intersection
        if X is a leaf
            answer += [all cars that are inside Y] // 0 or 1 car
        else
            answer += QuadList(S, top left of X)
            answer += QuadList(S, top right of X)
            answer += QuadList(S, bottom left of X)
            answer += QuadList(S, bottom right of X)
        endif
    endif
    return answer
```

We're done.



Any questions?