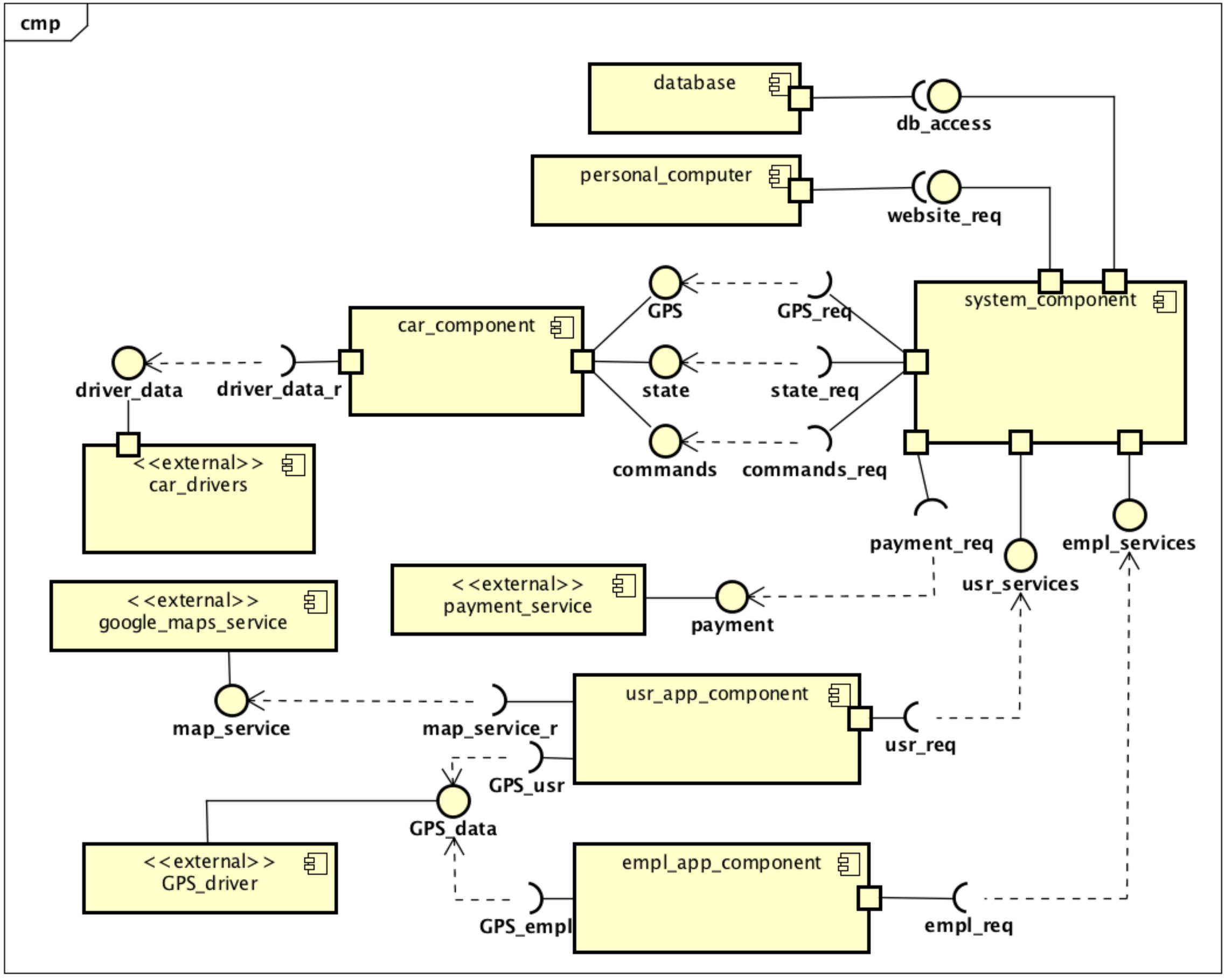
Design Document

PowerEnJoy

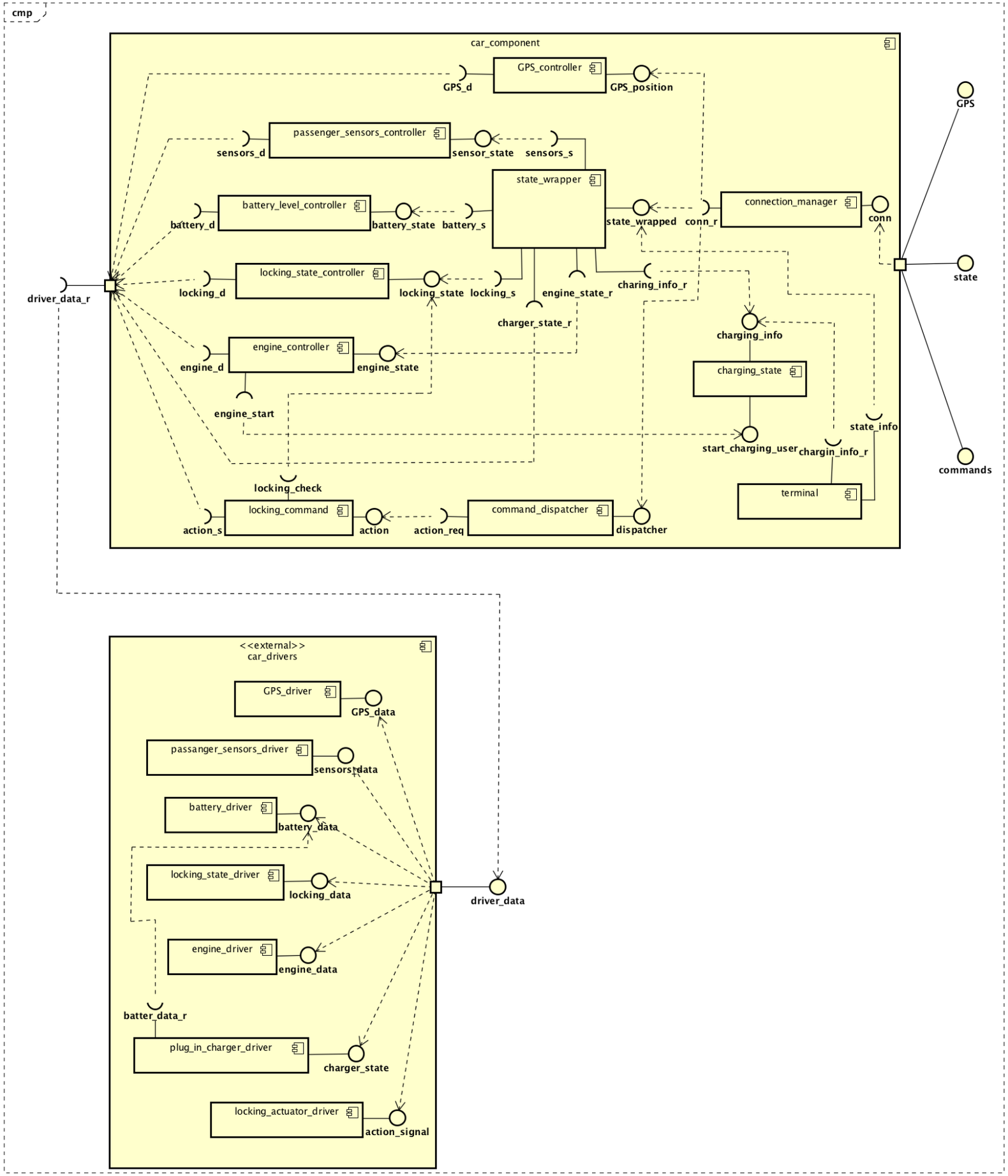
Peverelli Francesco
Reppucci Federico

EXTERNAL COMPONENTS INTERFACES

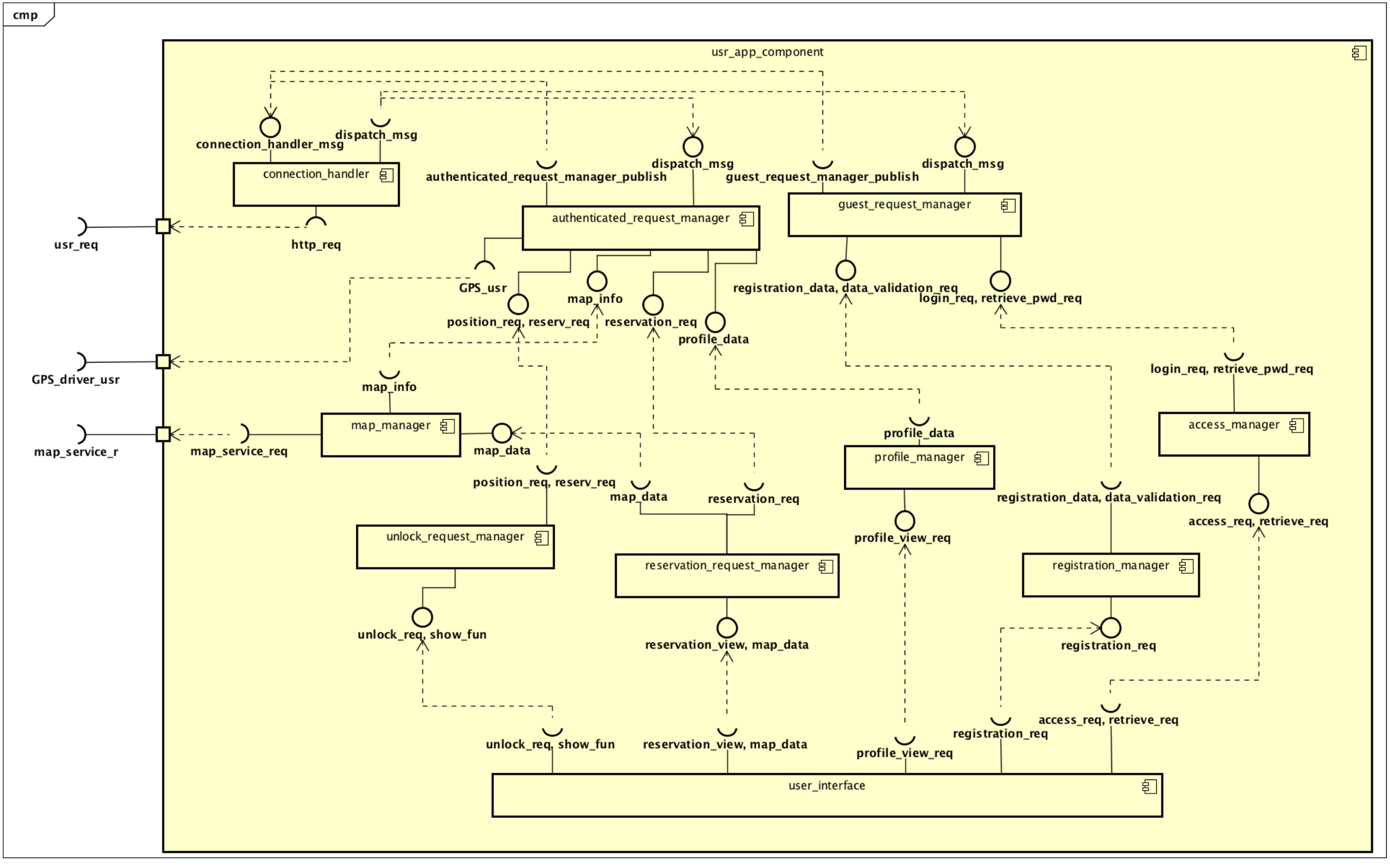
HIGH LEVEL COMPONENTS



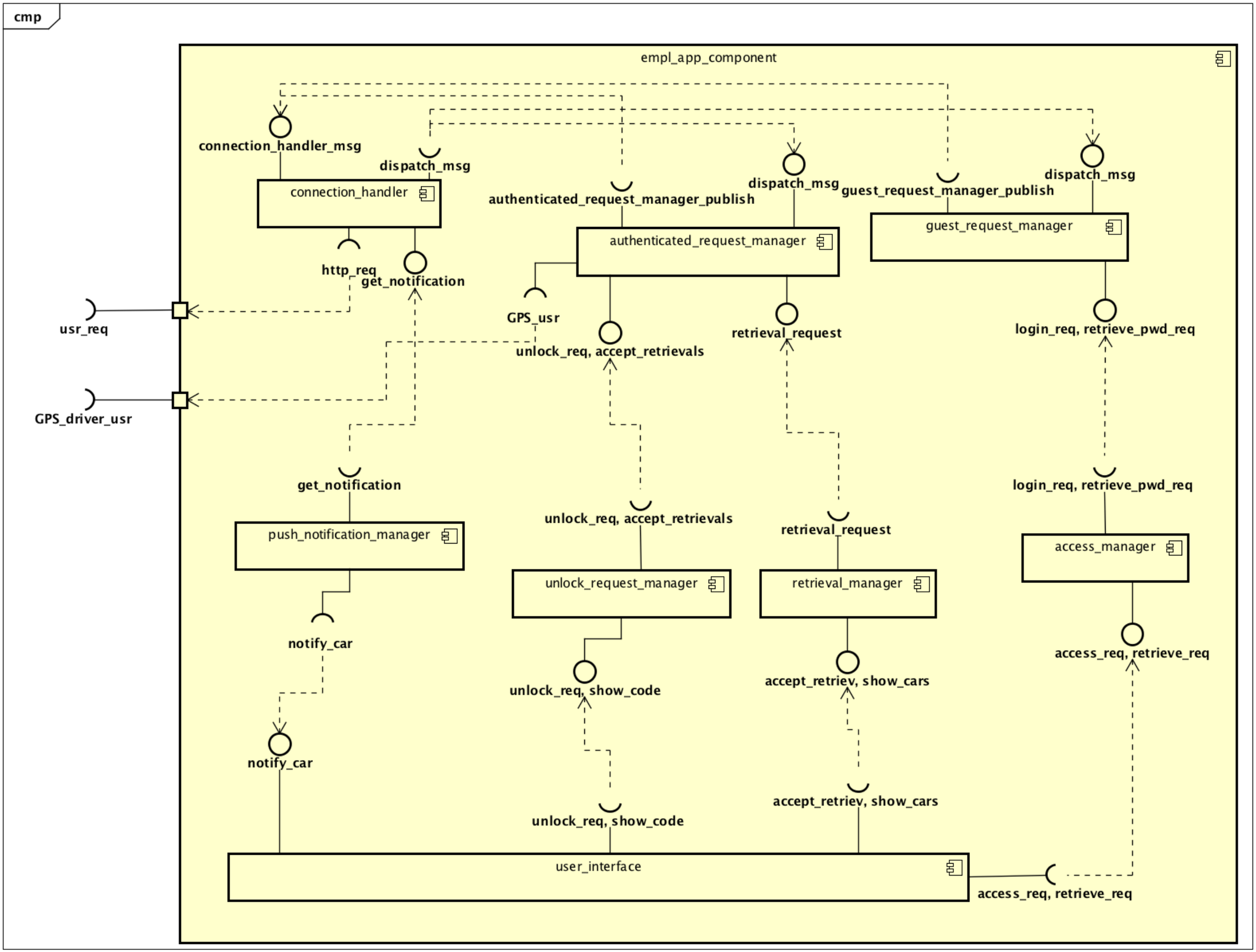
CAR COMPONENT



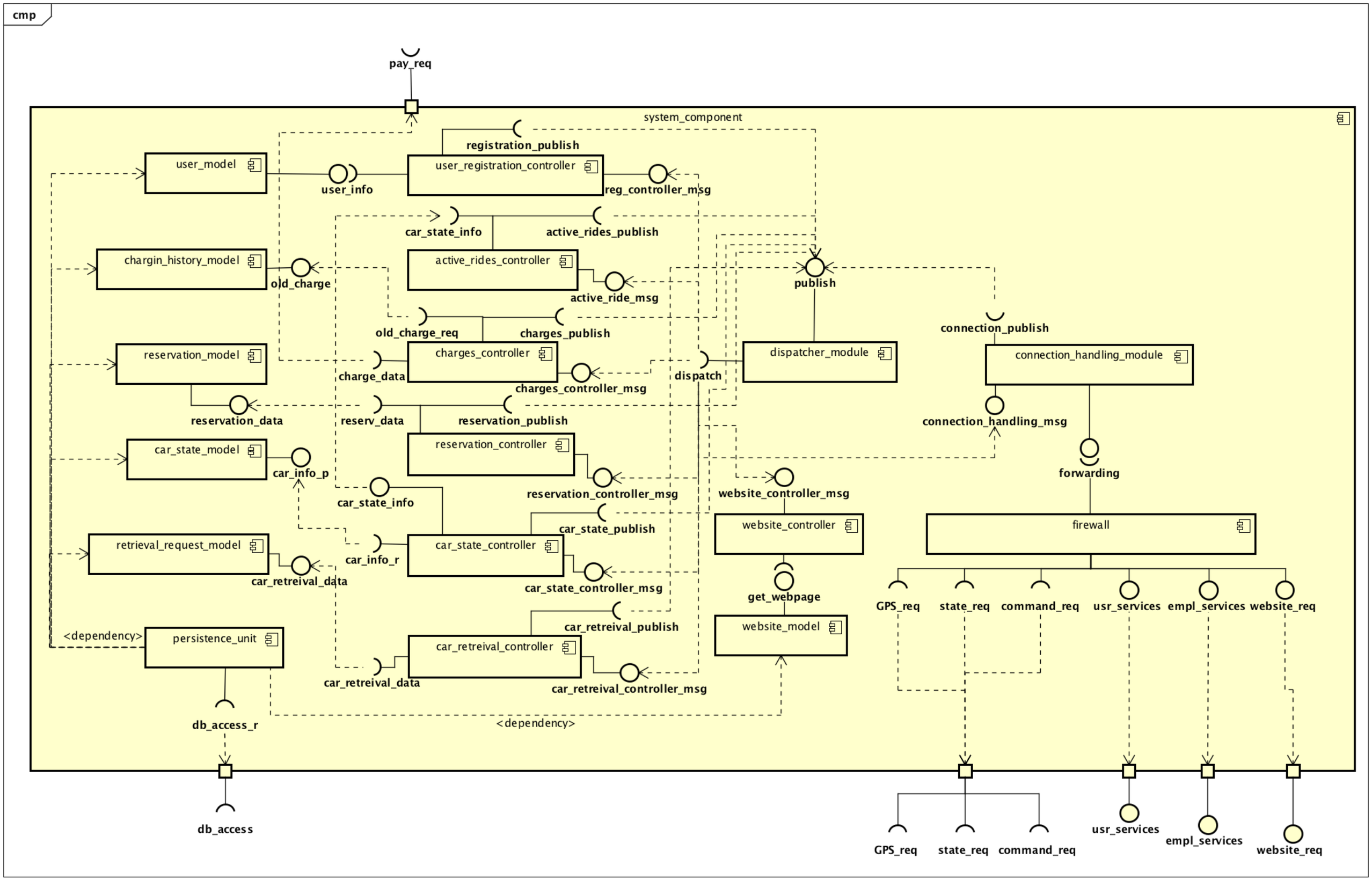
USER APPLICATION COMPONENT



EMPLOYEE APPLICATION COMPONENT

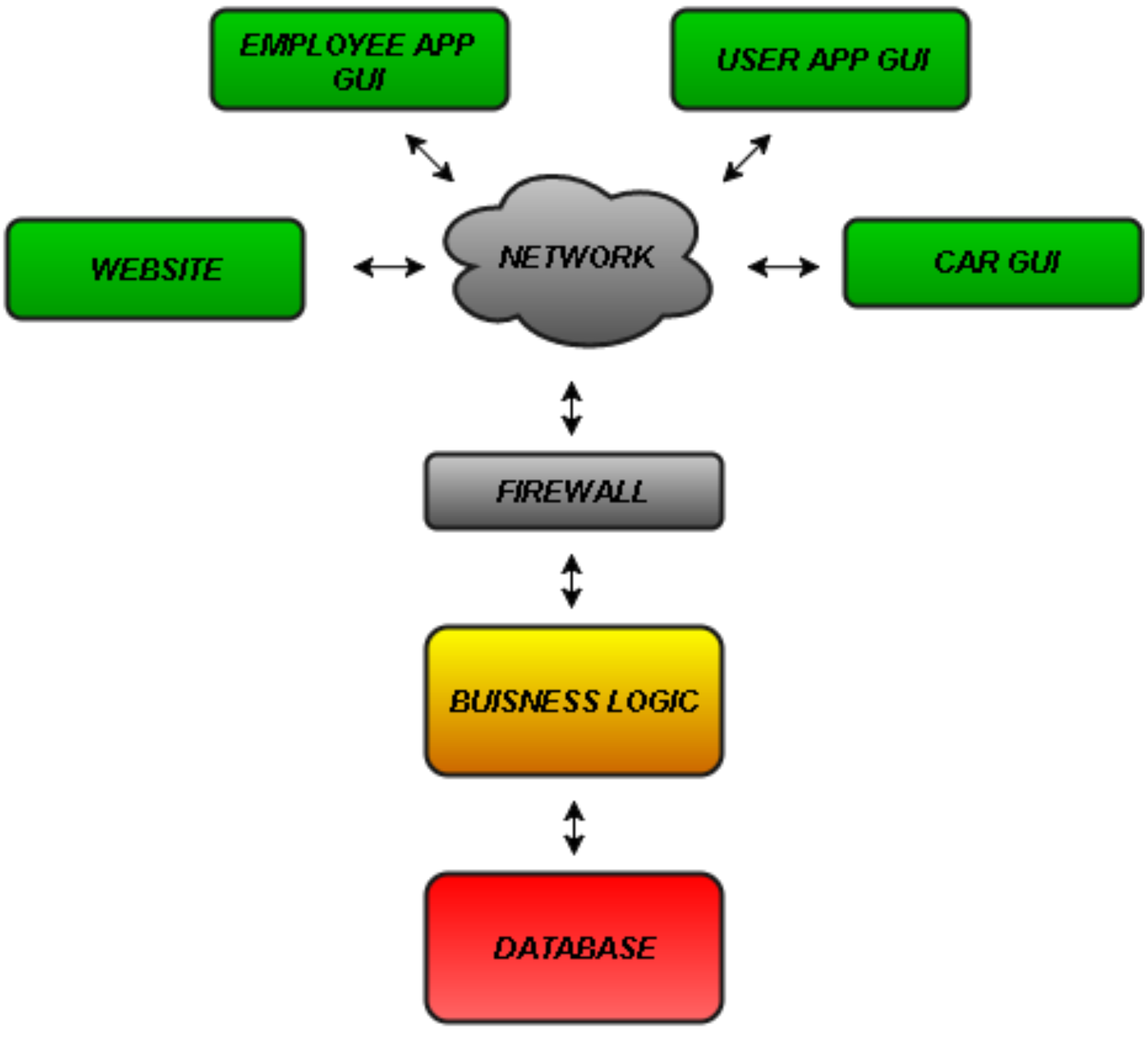


SYSTEM COMPONENT



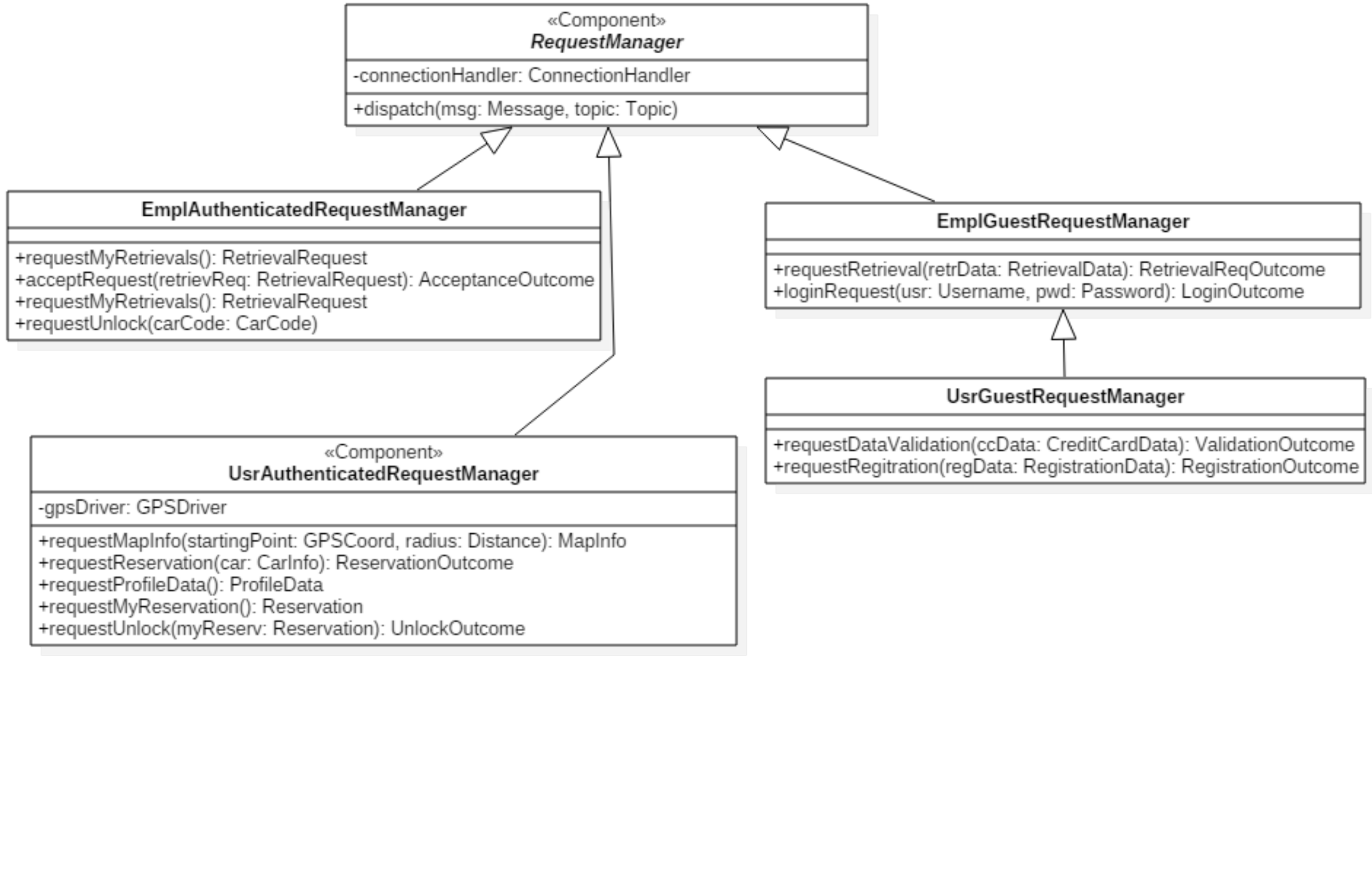
SOFTWARE ARCHITECTURE

The result is a **three-tier architecture**, where the presentation layer is located on the mobile applications and the car, the business logic is almost entirely on the server's side (although both the mobile apps and the software systems on the car contain some control logic, it is mostly used to formulate requests for the server to evaluate, or to pass on messages to act upon), and the persistency layer comes in the form of a database component.

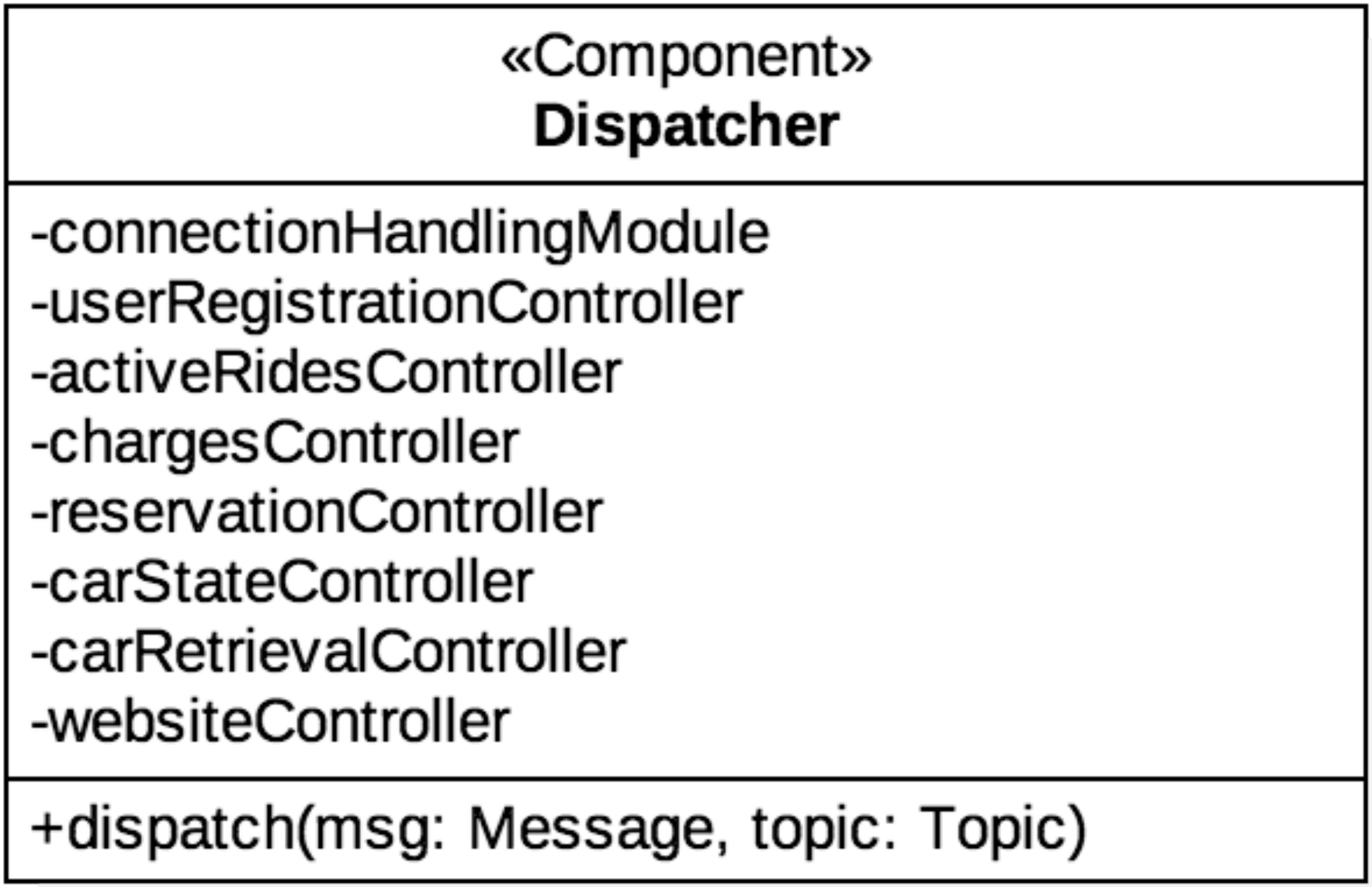


INTERFACES

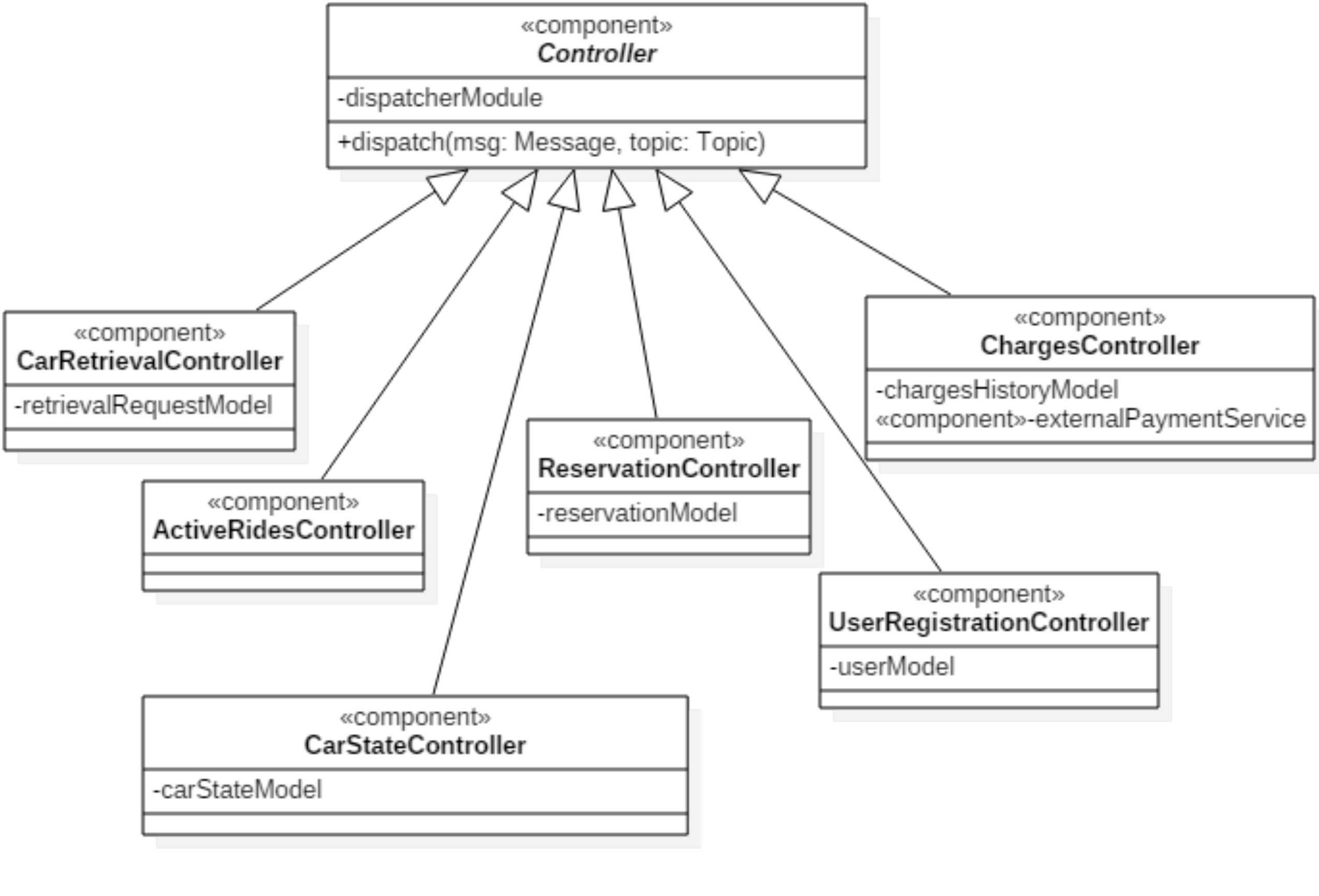
REQUEST MANAGER INTERFACE



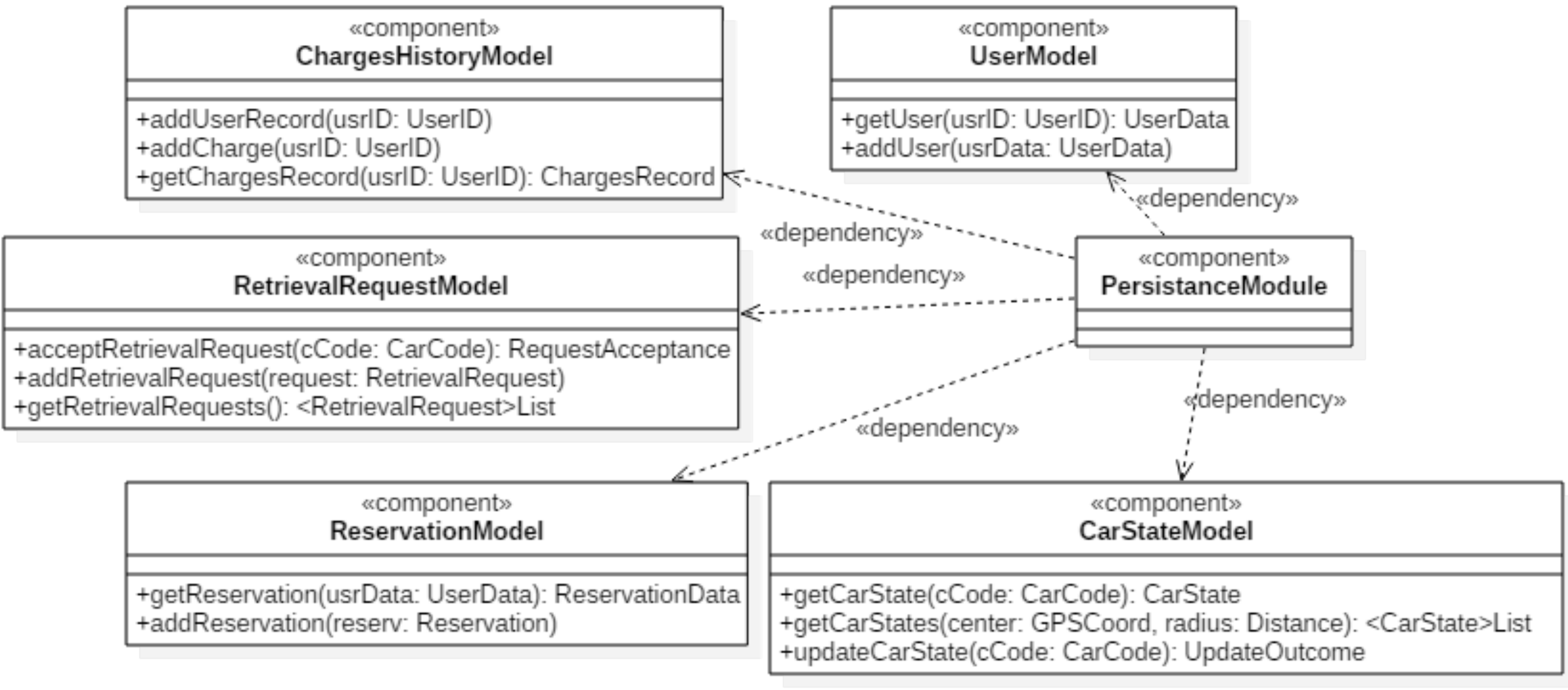
DISPATCHER INTERFACE



CONTROLLER INTERFACE

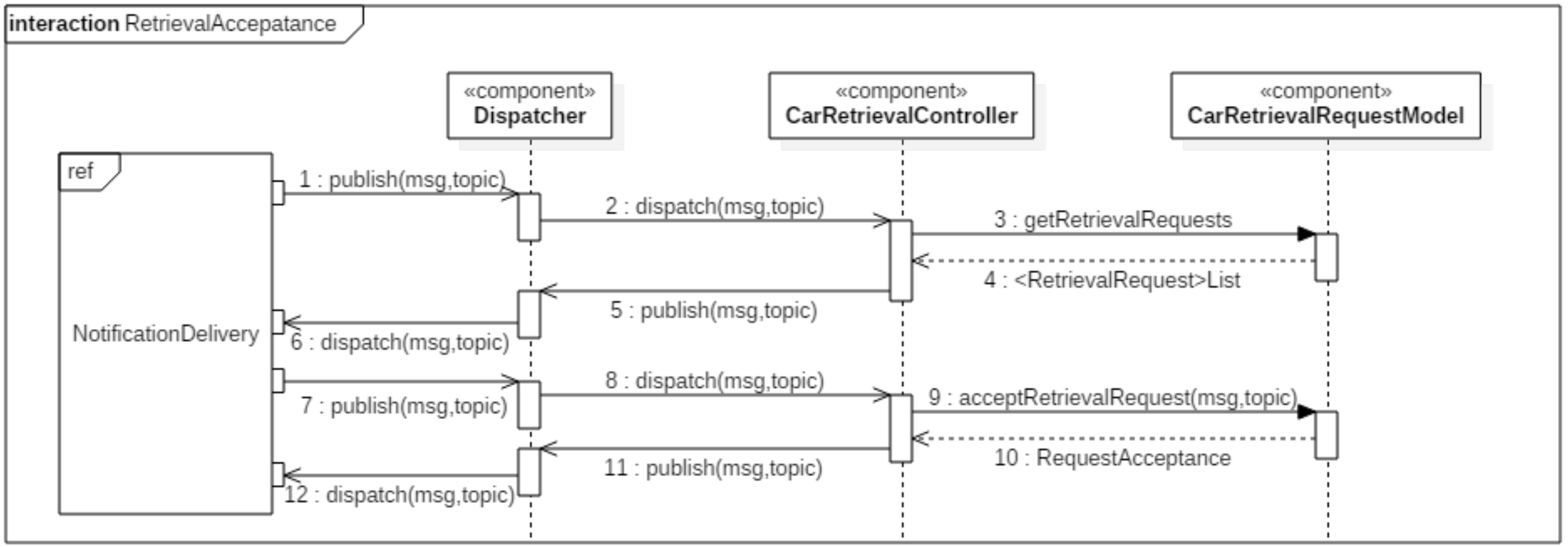


DATA MODEL INTERFACE

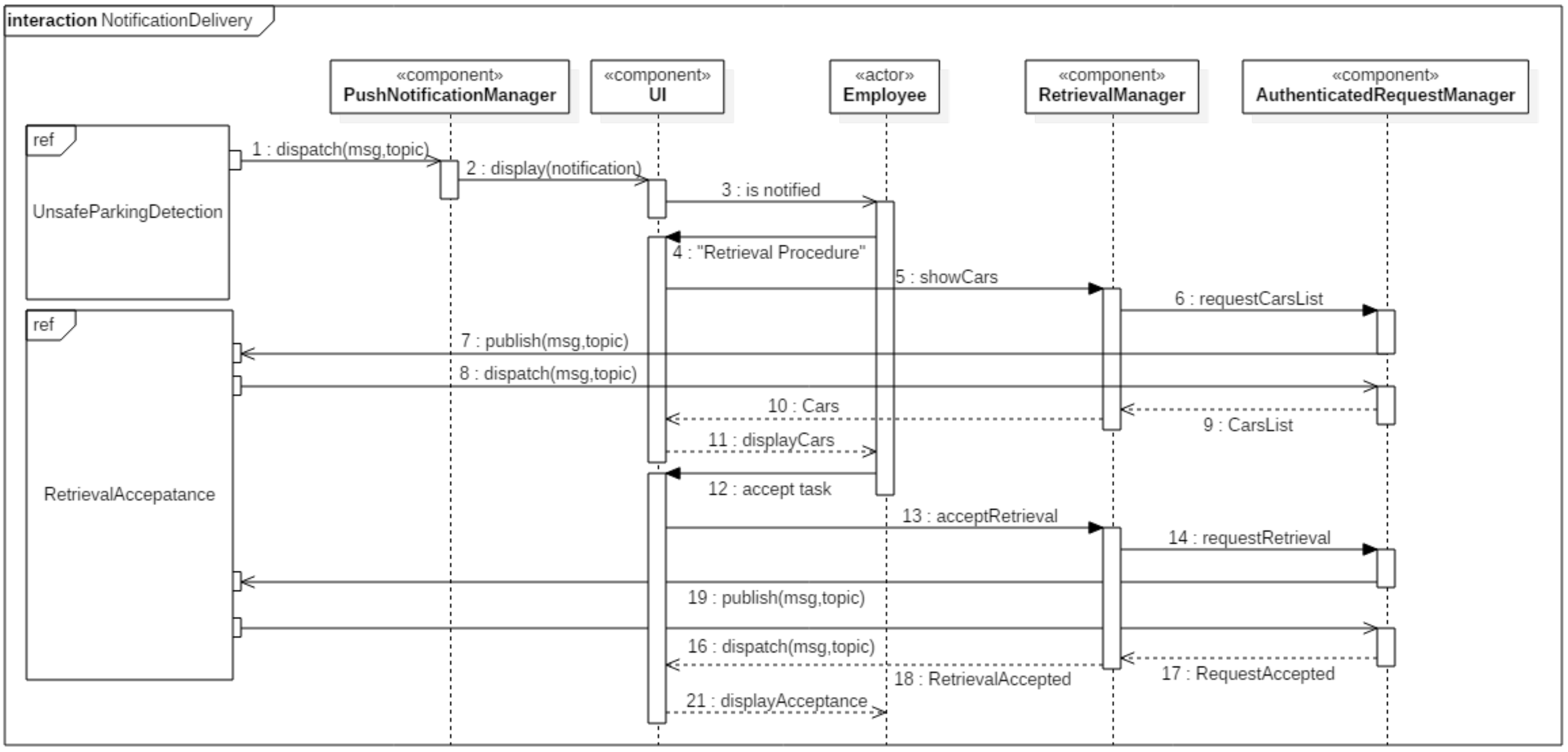


RUNTIME SEQUENCE DIAGRAM

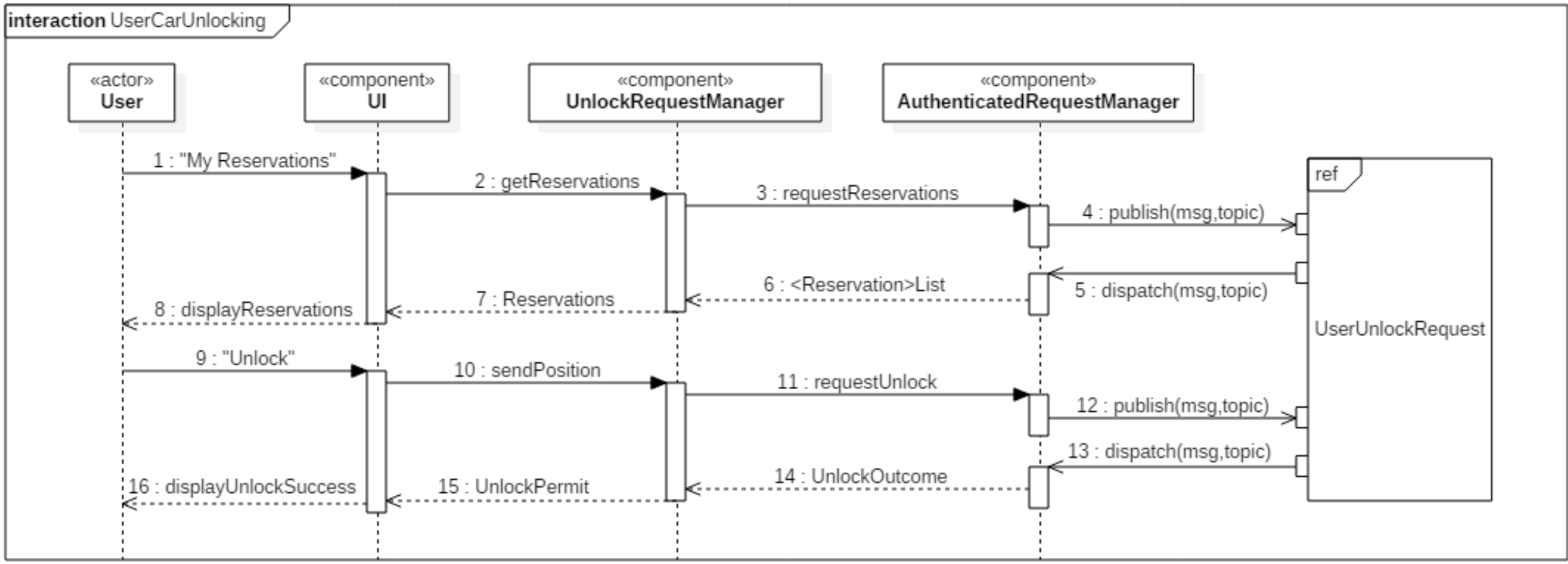
RETRIEVAL ACCEPTANCE PROCEDURE



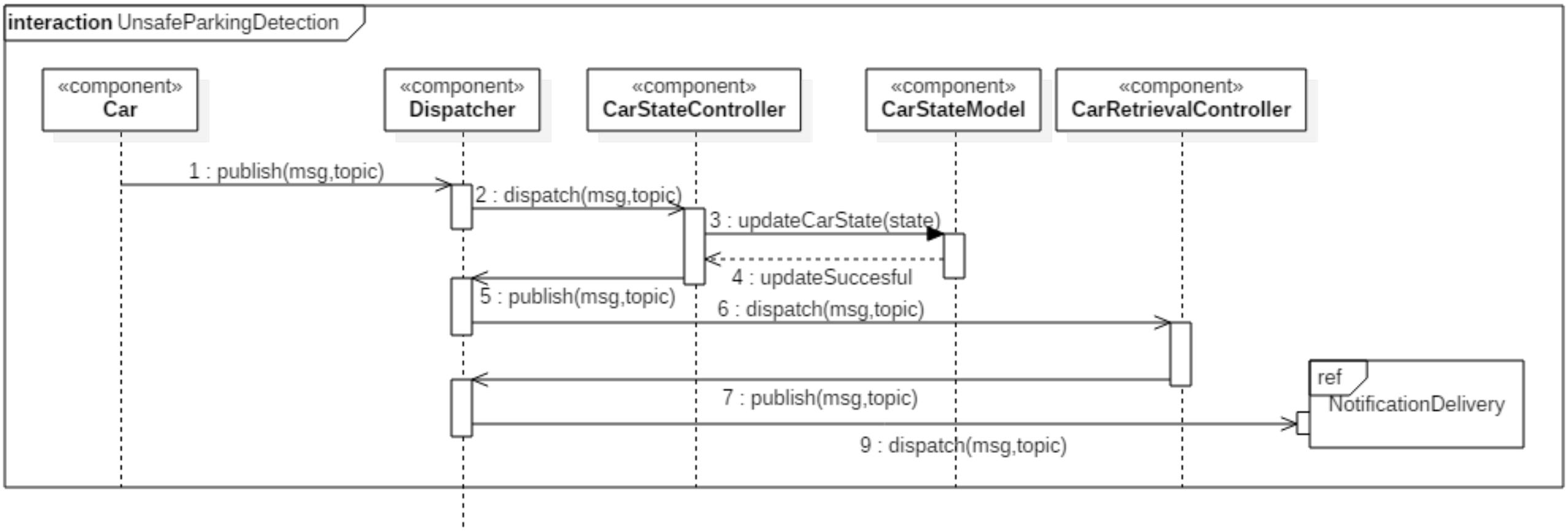
NOTIFICATION DELIVERY PROCEDURE



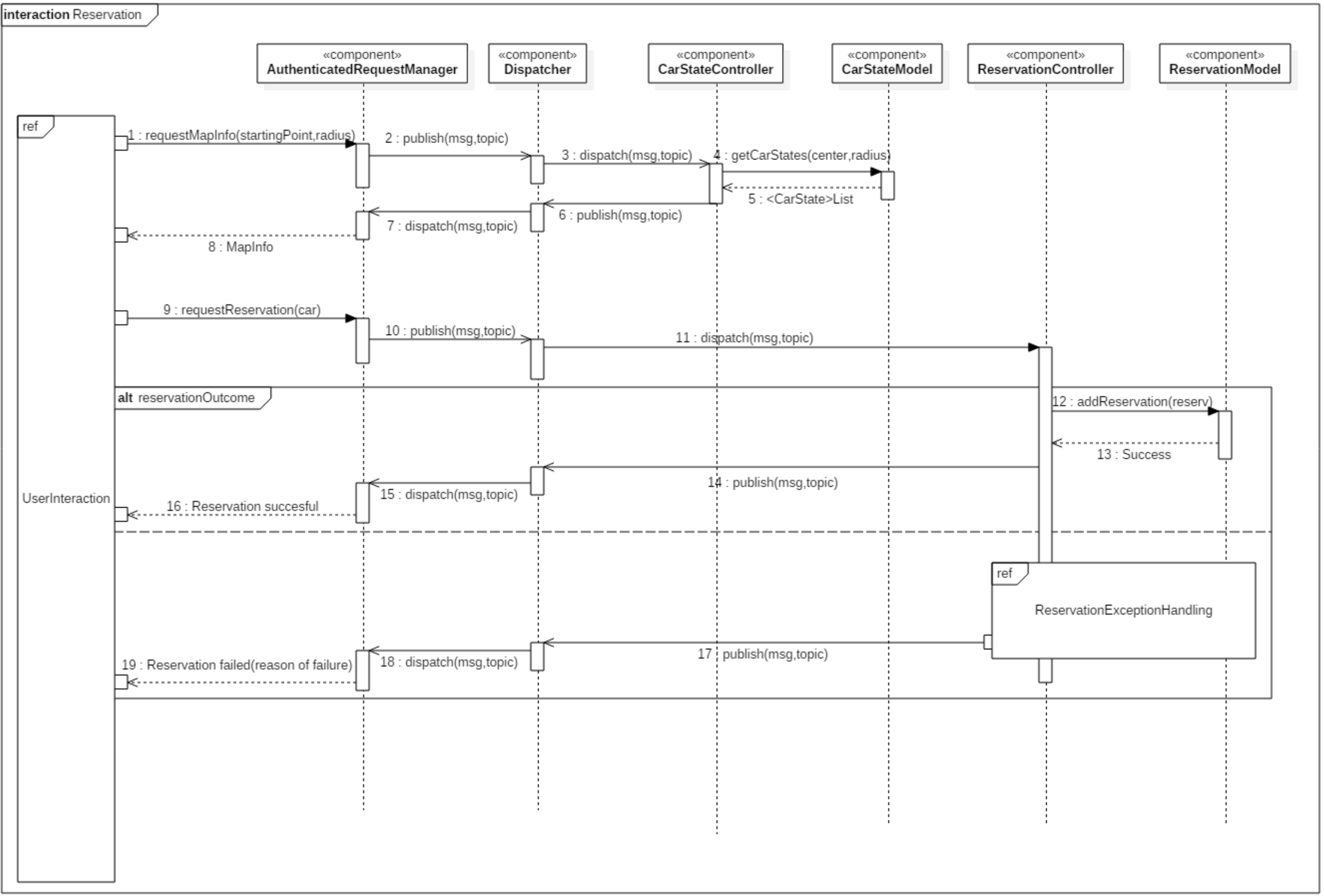
UNLOCKING USER PROCEDURE



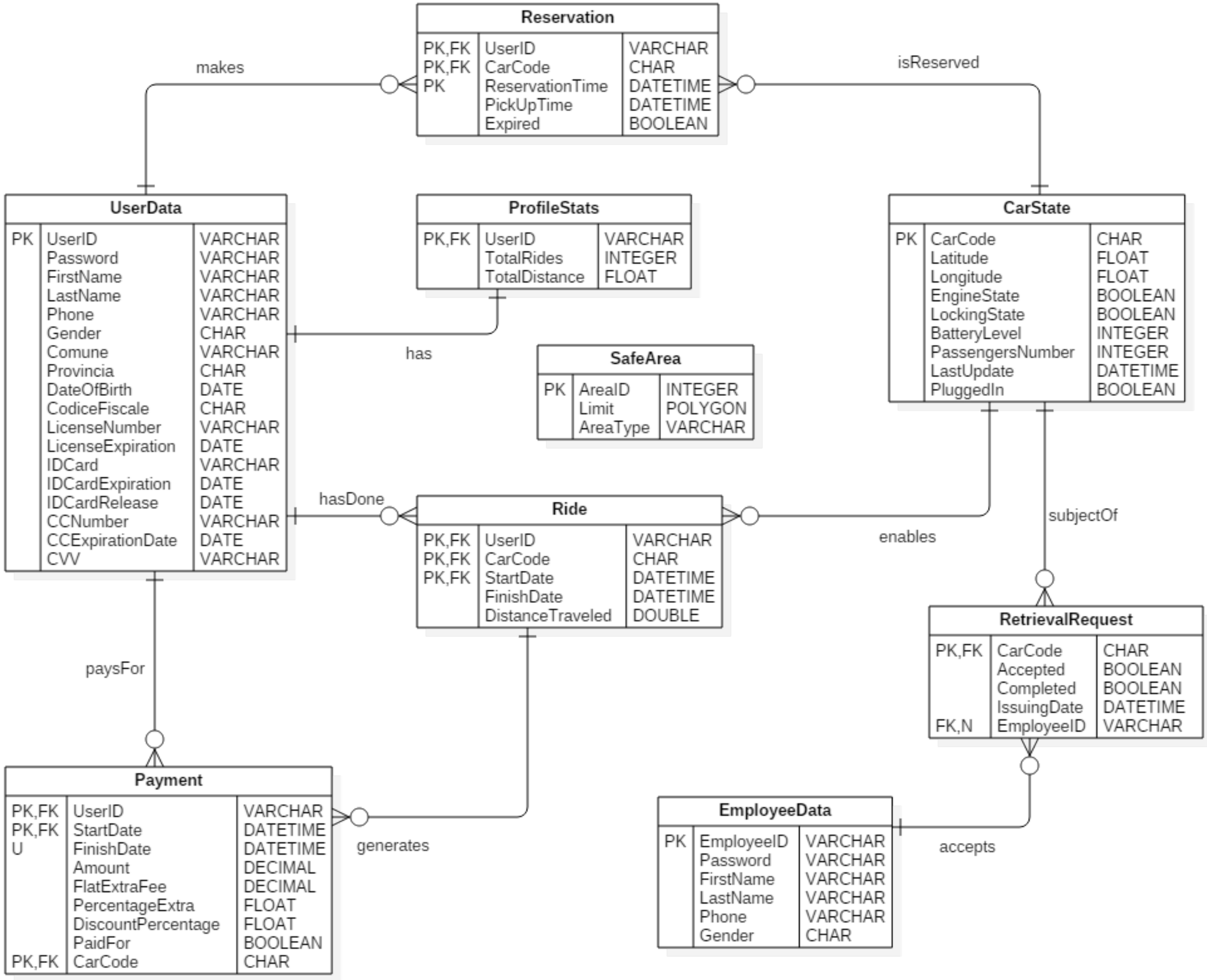
UNSAFE PARKING DETECTION PROCEDURE



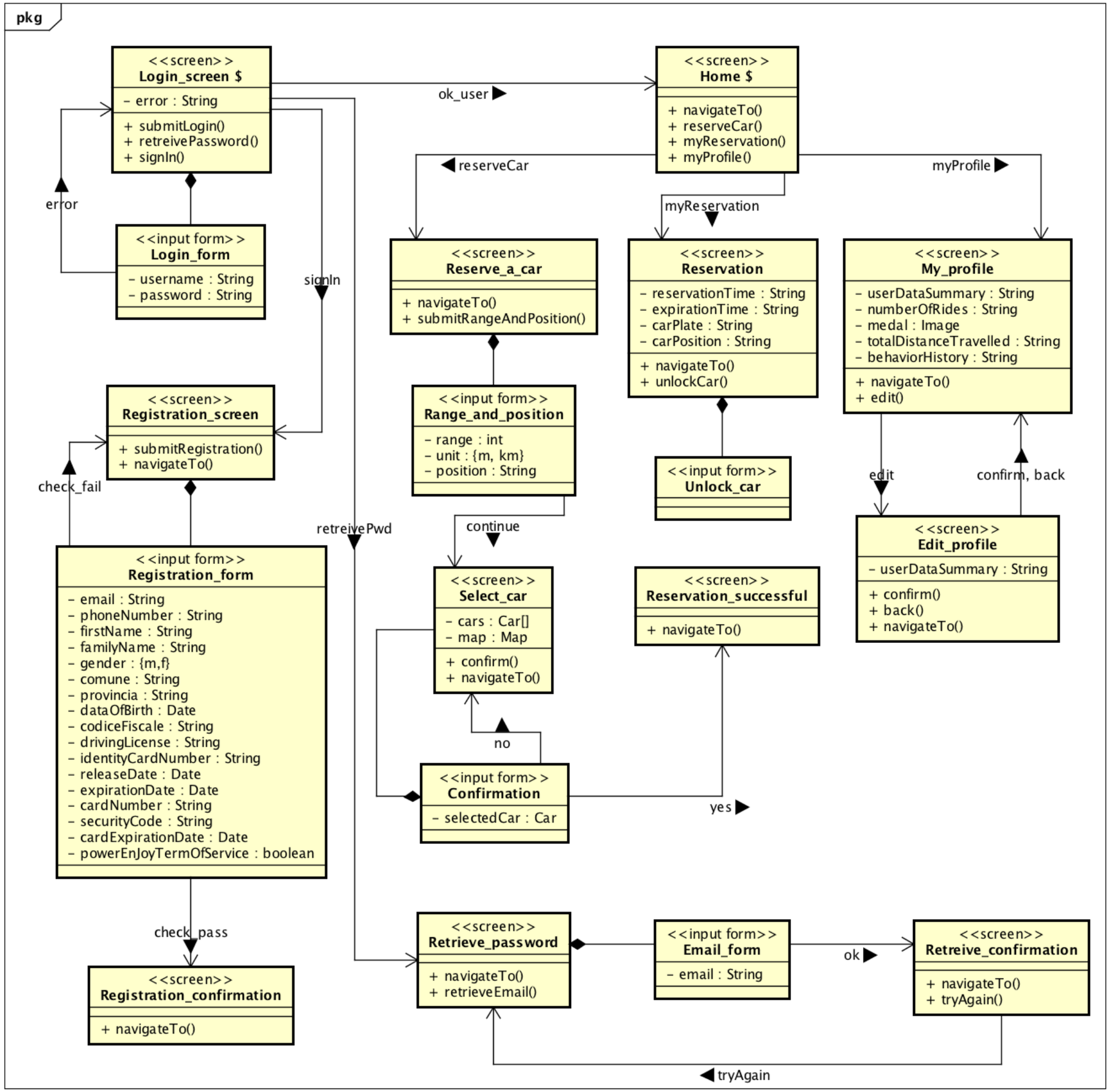
RESERVATION PROCEDURE



DATA MODEL


Reservation
PK,FK UserID VARCHAR
PK,FK CarCode CHAR
PK ReservationTime DATETIME
PickUpTime DATETIME
Expired BOOLEAN
makes
isReserved
UserData
PK UserID VARCHAR
Password VARCHAR
FirstName VARCHAR
LastName VARCHAR
Phone VARCHAR
Gender CHAR
Comune VARCHAR
Provincia CHAR
DateOfBirth DATE
CodiceFiscale CHAR
LicenseNumber VARCHAR
LicenseExpiration DATE
IDCard VARCHAR
IDCardExpiration DATE
IDCardRelease DATE
CCNumber VARCHAR
CCExpirationDate DATE
CVV VARCHAR
ProfileStats
PK,FK UserID VARCHAR
TotalRides INTEGER
TotalDistance FLOAT
has
SafeArea
PK AreaID INTEGER
Limit POLYGON
AreaType VARCHAR
CarState
PK CarCode CHAR
Latitude FLOAT
Longitude FLOAT
EngineState BOOLEAN
LockingState BOOLEAN
BatteryLevel INTEGER
PassengersNumber INTEGER
LastUpdate DATETIME
PluggedIn BOOLEAN
hasDone
Ride
PK,FK UserID VARCHAR
PK,FK CarCode CHAR
PK,FK StartDate DATETIME
FinishDate DATETIME
DistanceTraveled DOUBLE
enables
subjectOf
RetrievalRequest
PK,FK CarCode CHAR
Accepted BOOLEAN
Completed BOOLEAN
IssuingDate DATETIME
FK,N EmployeeID VARCHAR
paysFor
Payment
PK,FK UserID VARCHAR
PK,FK StartDate DATETIME
U FinishDate DATETIME
Amount DECIMAL
FlatExtraFee DECIMAL
PercentageExtra FLOAT
DiscountPercentage FLOAT
PaidFor BOOLEAN
PK,FK CarCode CHAR
generates
EmployeeData
PK EmployeeID VARCHAR
Password VARCHAR
FirstName VARCHAR
LastName VARCHAR
Phone VARCHAR
Gender CHAR
accepts

USER INTERFACE DESIGN



SELECTED TOOLS

Operating systems

SUSE Linux Enterprise 10: is the operating system running on the server machines

Microsoft Windows Embedded Automotive: is the operating system that the provided cars come with.

Windows, iOS and Android are identified as the operating systems for which the mobile application will be developed.

Application server

Payara: is an open-source application server derived from Glassfish, and although it's not Java EE certified it is effectively Java EE 7 compliant and offers far more stable patch releases, security fixes, production support and developer support. It also has a very responsive community.

Database

MySQL: among all the available DBMSs, MySQL stands out for its scalability and flexibility, which also comes along with good performances and availability. In addition, it is open source, just like Payara, and can run on multiple platforms. All these reasons contribute to make MySQL our DBMS of choice.

Frameworks

J2EE: is a solid framework which will be used to ease the development of the application logic and the presentation layer for both the website and the apps on the server side.

Windows Automotive Application Framework: is chosen as a native framework that our windows developers are familiar with.

Communication

On the server side JMS is used as the messaging API, and the messages are exchanged in a text-based XML format via SOAP Transport Protocol.

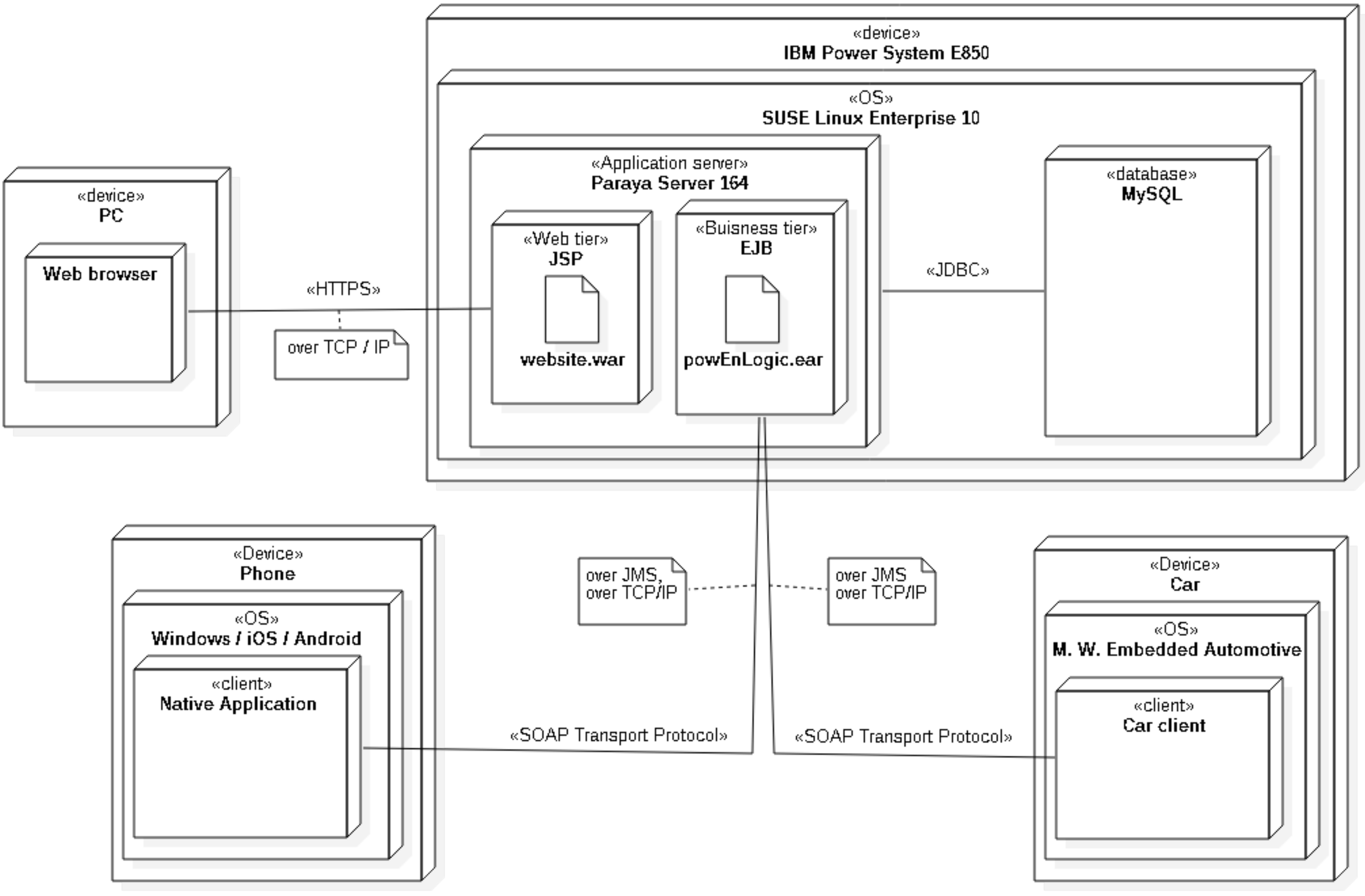
On the client side Kaazing WebSocket Gateway APIs are used to support JMS messaging.

Ides

The selected IDEs are NetBeans for the development of the server-side application, VisualStudio for the development of the software running on the cars, and AndroidStudio, XCode and VisualStudio are the integrated environments selected to develop for Android, iOS and Windows respectively.

DEPLOYMENT

USER INTERFACE DESIGN



ALGORITHM DESIGN

```
/**
 * This class controls whether the reservation is well-formed.
 * A reservation is well-formed if:
 *    1) the user hasn't already an active reservation, and
 *    2) the user hasn't some pending payment, and
 *    3) the user's credit card and identity card aren't expired,
 *    4) the selected car is actually available
 */
class ReservationController implements Publish, Subscriber{
    attribute ReservationModel reservationModel

    method void checkAndAddReservation(Reservation reservation) {
        try{
            doubleReservationCheck(reservation);
            lastPaymentCheck(reservation);
            validityUserDataCheck(reservation);
           carAvailabilityCheck(reservation);

            reservationModel.addReservation(reservation);

    } catch(DoubleCheckReservation or
     PendingPaymentException or
     InvalidUserDataException or
     NotAvailableCarException){

    throw ReservationException
    }
}
. . .
```

```
. . .
method void doubleReservationCheck(Reservation reservation) {
    user = reservation.getUser();
    reservationToCheck =
   reservationModel.getNotExpiredReservation(user);

    if (reservationToCheck != null)
        throw DoubleCheckReservation
}
```

```
method void doubleReservationCheck(Reservation reservation) {
    user = reservation.getUser();
    reservationToCheck =
   reservationModel.getNotExpiredReservation(user);

    if (reservationToCheck != null)
        throw DoubleCheckReservation
}
. . .
```

```
. . .
method void validityUserDataCheck(Reservation reservation){
    userData = reservation.getUser().getUserData();

    if (not userData.areValid())
        throw InvalidUserDataException
}
```

```
    method void carAvailabilityCheck(Reservation reservation){
        carState = reservation.getCar().getCarState();

        if (not carState.isAvailable())
            throw NotAvailableCarException
    }
}
```

REQUIREMENTS

TRACEABILITY

[R1.1]: The app is available for any person to download and run on his/her phone

This requirement will be achieved at a later stage.

[R1.2]: From the home page of the app any person can carry out the registration procedure

[R1.3]: The registration procedure requires a person's credentials and payment info to be carried out

[R1.4]: The registration procedure uses the external payment service to verify the validity of the provided payment info

[R1.5]: At the end of the registration procedure the person whose credentials were used is registered in the system

[R1.6]: At the end of the registration procedure the person receives an e-mail containing a password which he/she can use to access the system

[R1.7]: At the end of the registration procedure the person can ask the system to send another mail

COMPONENTS: UserInterface, RegistrationManager, GuestRequestManager, UserRegistrationController, UserModel

[RA1.1]: The app allows any person to log in by providing a valid e-mail and password

[RA1.2]: The app does not allow any person who does not provide a valid e-mail and password to log in

[RA2.1]: From the home page of the app the password retrieval procedure can be initiated by any person

[RA2.2]: If a person provides a valid e-mail address during the password retrieval procedure the system sends an e-mail to that address containing the associated password

COMPONENTS: UserInterface, AccessManager, GuestRequestManager, UserRegistrationController, UserModel

[RA3]: Access to the PowerEnJoy's website (a static page) is granted upon request by the system (no login required)

COMPONENTS: WebsiteController, WebsiteModel

[R2.1]: The "Reserve a car" function can be accessed by the user from the home page of

the app
[R2.2]: The "Reserve a car" function allows the user to select a range (distance)

[R2.3]: The system acquires the user's current position through the GPS coordinates of the user's phone

[R2.4]: The system tracks all available cars' current position through their GPS coordinates [R2.4.1]: The cars must possess a device which can be tracked via GPS

[R2.5]: The "Reserve a car" function allows the user to select a starting position for the search, which can be either their current location or a given address

[R2.6]: When the user confirms the inserted parameters the search is carried out and the "Reserve a car" function displays to the user the data of the search acquired from the system in a Google provided map

[R3.1]: The app allows the user to tap on any available car on the map displayed as the result of a search conducted through the "Reserve a car" function.

[R3.2]: When a user taps on a car the app generates a pop-up asking the user if he/she wants to confirm the reservation.

[R3.3]: As long as the car was not reserved by another user in the meantime, when the user confirms the car is marked as reserved by the system and the user can see the "Reservation successful!" message on the app.

[R3.4]: When the system marks a car as reserved any reservation request from any user is rejected by the system while the car is in the reserved state.

[R3.5]: A car is in reserved state for one hour from the moment it was marked as reserved.

[R3.6]: A car in reserved state is not signaled by the system during the "Reserve a car" procedure.

[R3.7]: After one hour from its reservation a car is no longer in reserved state.

[R3.8]: A car not in reserved state is considered available by the system only if it is parked in a safe area less than 3 km away from a power grid station and has more than 20% of its battery.

COMPONENTS: UserInterface, ReservationRequestManager, MapManager, AuthenticatedRequestManager, ReservationController, ReservationModel, CarStateController, CarStateModel, + StateWrapper and the controllers on the car providing status updates.

[R4.1]: One hour after a car has been reserved if it was never ignited the system charges for 1 EUR the user who reserved it

COMPONENTS: ReservationModel, ReservationController, ChargesController

[R5.1]: From the home page of the app the user can access the "My reservations" section

[R5.2]: In the "My reservations" section if the user has reserved a car less than an hour ago an active reservation is displayed with an "Unlock" button

[R5.3]: If the user is less than 10 meters away from the car and presses the unlock button the car unlocks

COMPONENTS: UserInterface, UnlockRequestManager, AuthenticatedRequestManager, ReservationController, ReservationManager, CarStateController, CarStateModel, CommandDispatcher, LockingCommand

[R6.1]: When a car is ignited the system starts charging the last user who reserved the car

[R6.2]: When the charging starts, the display on the car shows a "Current charge" field with a number representing the current total charge, which starts from 0

[R6.3]: Once a minute the "Current charge" value is incremented by a set amount

COMPONENTS: EngineController, Terminal, StateWrapper, ChargesController, ChargesHistoryModel, ChargingState

[R7.1]: When a car is stopped and the sensors in the car detect no one inside, if a user was being charged for the car the system stops charging him/her.

[R7.2]: One minute after a car has been stopped and the sensors in the car detect no one inside, the system locks the car.

COMPONENTS: PassengerSensorsController, LockingStateController, StateWrapper, ChargesController, ChargesHistoryModel

[R8]: The moment the car is stopped, if the sensor in the car detected two passengers the system records it as a possible discount of 10%

[R9]: The moment the car is stopped and the sensors in the car detect no one inside, if the car has more than 50% of its maximum battery the system records it as a possible discount of 20%.

[R10]: If before 2 minutes since the moment the car has been stopped and its sensors detected no one was inside the car is plugged in a power grid and its position is within a special parking

[R11.0]: The moment the car is stopped and the sensors in the car detect no one inside, if the safe area nearest to the car is more than 3km away from it, the system records an extra fee of 30%

[R11.1]: The moment the car is stopped and the sensors in the car detect no one inside, if the car has less than 20% of its maximum battery, the system records an extra fee of 30%.

[R12]: After two minutes since the car has been stopped and its sensors detected no one was inside, the system applies all the extra fees and if there are none it applies the highest discount among the possible ones to the cost of the ride.

COMPONENTS: GPS_Controller, PassengerSensorsController, BatteryLevelController, EngineController, StateWrapper, ChargesController, ChargingHistoryModel

[RA4]: If a user with a pending payment procedure tries to reserve a car, a pop-up lets the user know that he/she needs to pay for his/her last ride to be able to reserve a car and the app does not allow the user complete the reservation procedure.

COMPONENTS: ChargesController, ChargesHistoryModel + components for a reservation

[RA5]: If in the user's profile either the credit card or identity card expiration date has already passed, when the user tries to reserve a car a pop-up lets him/her know that the data in the user's profile need to be updated and the app prevents the reservation procedure from being completed

COMPONENTS: UserRegistrationController, UserModel + components for a reservation

[RA6.1]: From the home page of the app the user can access the "My profile" section

[RA6.2]: From the "My profile" section the user can use the "Edit profile" button to modify his/her credential and payment info

[RA6.3]: The system can check via the external payment service whether the payment info inserted are valid

[RA6.4]: If the inserted payment info is valid the user can save the changes by tapping the "Confirm" button.

COMPONENTS: UserInterface, ProfileManager, AuthenticatedRequestManager, UserRegistrationController, UserModel

[RA7]: If the user has already a reservation which is not expired yet when he/she tries to reserve a car, a pop-up lets the user know that he/she cannot reserve a car and the app does not allow the user complete the reservation procedure

COMPONENTS: ReservationController, ReservationModel + other components for a reservation

[RA8]: When a car is locked the system checks its GPS coordinates, and if they correspond to those of a non-safe area the last user who reserved the car is charged for a set extra fee.

COMPONENTS: GPSController, CarStateController, CarStateModel

[R13.1]: Each employee has access to an application, AdminPowerEnJoy, on their phone

[R13.2]: When a car is locked the system checks its GPS coordinates, and if they correspond to those of a non-safe area all employees are notified through AdminPowerEnJoy that the car needs to be retrieved

[R13.3]: AdminPowerEnJoy allows an employee to accept a retrieval request through the "Retrieval procedure" function

[R13.4]: If an employee has already accepted a retrieval request, the retrieval request can no longer be accepted

[R13.5]: After 12 hours, if an employee has accepted a retrieval request but has not retrieved the car, the request is issued again by the system and another employee can accept it

[R13.6]: When an employee is notified of a car to retrieve, the notification contains the information necessary to set up the navigator of the company's cars to find the position of the car to retrieve

[R13.7]: AdminPowerEnJoy allows an employee to unlock any car for which he/she has accepted a retrieval request

COMPONENTS: UserInterface, RetrievalManager, PushMotificationManager, UnlockrequestManager, AuthenticatedRequestManager, CarStateController, CarStateModel, CarRetrievalController, RetrievalRequestModel

[R14]: When the employee ignites a car which was opened through AdminPowerEnJoy the system does not initiate any charging procedure

COMPONENTS: UnlockRequestManager, AuthenticatedRequestManager, CarStateController, CarStateModel, CommandDispatcher, LockingCommand