# Architecture - Week 39

>> Two use cases, Bob & Alice :

> *Use Case 1 - Bob :*

* Bob is a lambda student who wants to move
* Bob need to transport his bed from is parents house (Nice) to is new student appartment (Sophia)
* Bob is a smart guy so he decide to use BlablaMove
* He log in on BlablaMove : he has the right amount of points.
* He fill a form where he give some informations : the start point of the things he want to move, the arrival point, the size of his bed, the weight of his bed, when he wants to move (range ?) and the maximum number of points he want to spend.
* The system give him a list of results who answer his need. (Number of points/Date/ Hours/...)
* Bob chose a ride for his bed
* The system answer him with a recap
* Bob confirm
* He receive a confirmation mail from BlablaMove : Charlie can help him to move his things
* --- Ellipse ---
* At the chosen date, Charlie goes to Bob house and take his bed
* Charlie goes to Sophia
* Bob receive an notification that confirm the delivery of his bed
* Bob can now confirm the transaction to BlaBlaMove
* After the confirmation from Bob, the BlablaMove collects the points that were needed for this transaction

> *Use Case 2 - Alice :*

* Alice is a student who lives in Nice and goes to Sophia in car every day for her studies
* She decide to create an account to BlablaMove to help others student to move their things
* She create her account and specify the type of car (5 places, medium) and her disponibilities.
* On BlaBlaMove she offers to transport things between Nice and Sophia every day, between 7:30 am (Nice) and 8:30 am (Sophia).
* When she makes the offer, BlaBlaMove suggests an amount of point she should charge for the delivery. (Based on the number of points that are usually charged for this distance. )
* She can choose to charge the amount BlaBlaMove suggest her, or she can make a new offer.
* One day she receives a mail from BlaBlaMove : Dimitri wants her to transport a box from Nice to Sophia at a certain date.
* She agrees to do it and confirm on BlablaMove.
* --- Ellipse ---
* At the chosen date, she goes to Dimitri's house and take his box in her car
* She goes to Sophia and leave the box where Dimitri told her to.
* She confirm on BlablaMove that she delivered the box.
* She receives points for the delivery.

# Architecture - Week 39

*> Use Cases variations to handle :*

* Bob choose Alice to move his bed but, in the end, Alice isn't available and refuse (She cancel on BlabalMove).
* Bob choose Alice but want to cancel the transfert after confirmation.
* Bob or Alice aren't at the meeting point.
* The package is not delivered or is broken during the transfert.
* Bob want to move something but he don't have enough points.
* One car isn't enough to get the box from A to B. (Connection necessary)

---

>> Focus on billing :

The calculation of our billing will be calculated according to the following parameters :

* Different prices depending on the distance to cover, the weight and size of the package(s).
* When a user wants to make a reservation, the more he will be close to the date of the move, the more he will spend points on the delivery. ( If he want to make a reservation for the next day, it will be more expensive than for the next week.)
* BlaBlaMove will take 10% of the points that Bob will pay for Alice to move his things. ( If Bob pay 100 points, Alice will receive 90 points and BlablaMove 10 points.)
* Optionally, when he will fill the form to do the research at the start of the process, Bob will be able to say how many points he want to spend on his move. (If he has 300 points, maybe he want to spend only 100.So if an offer cost more than 100 points, it won’t appear in his research.)

# Architecture - Week 39

The use cases that we defined this week (Alice & Bob, see above) leads us to this first architecture approach.
We defined three components classes that we will need later : one subsystem for the road composition, one for the management of users account and the last for the points management. Each one of this systems will interact with the others and with a database.
Below you can see, a diagram of a first approach to our architecture: