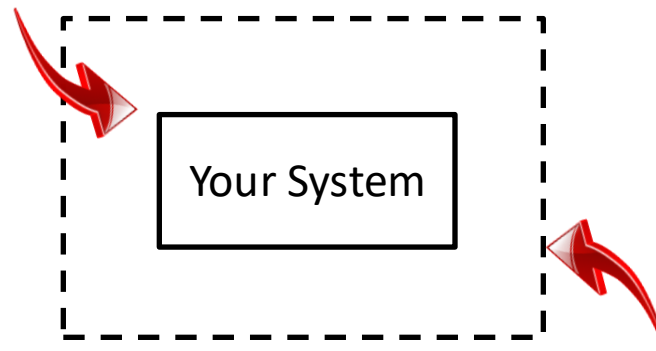


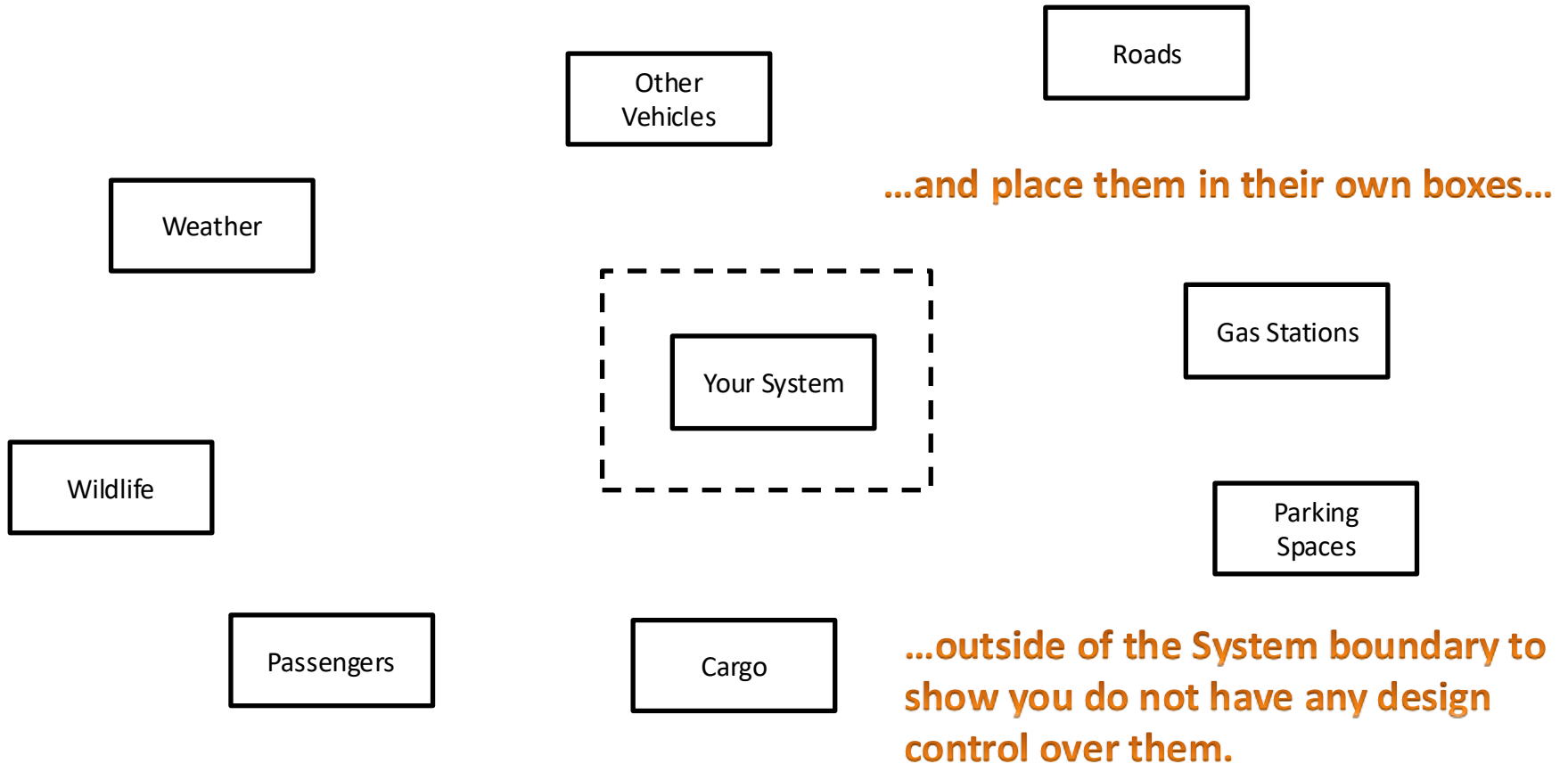
**Start with your “System” in a box**



**And place a dashed box around it to symbolize your System's boundary**

**For the purposes of this example, we'll assume your System is some kind of car-like vehicle**

Then start listing things your “System” has to interact with...



Capitalizing the things' names in the boxes is also common practice

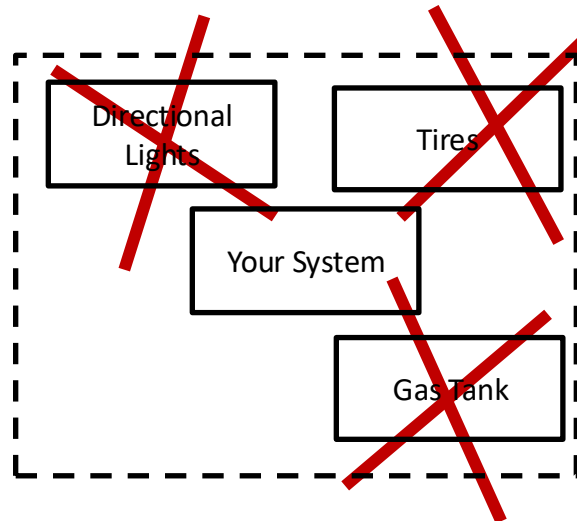
Other  
Vehicles

**Don't place other boxes in the system boundary,  
even if they're a part of your System!**

Roads

Weather

Wildlife



Gas Stations

Parking  
Spaces

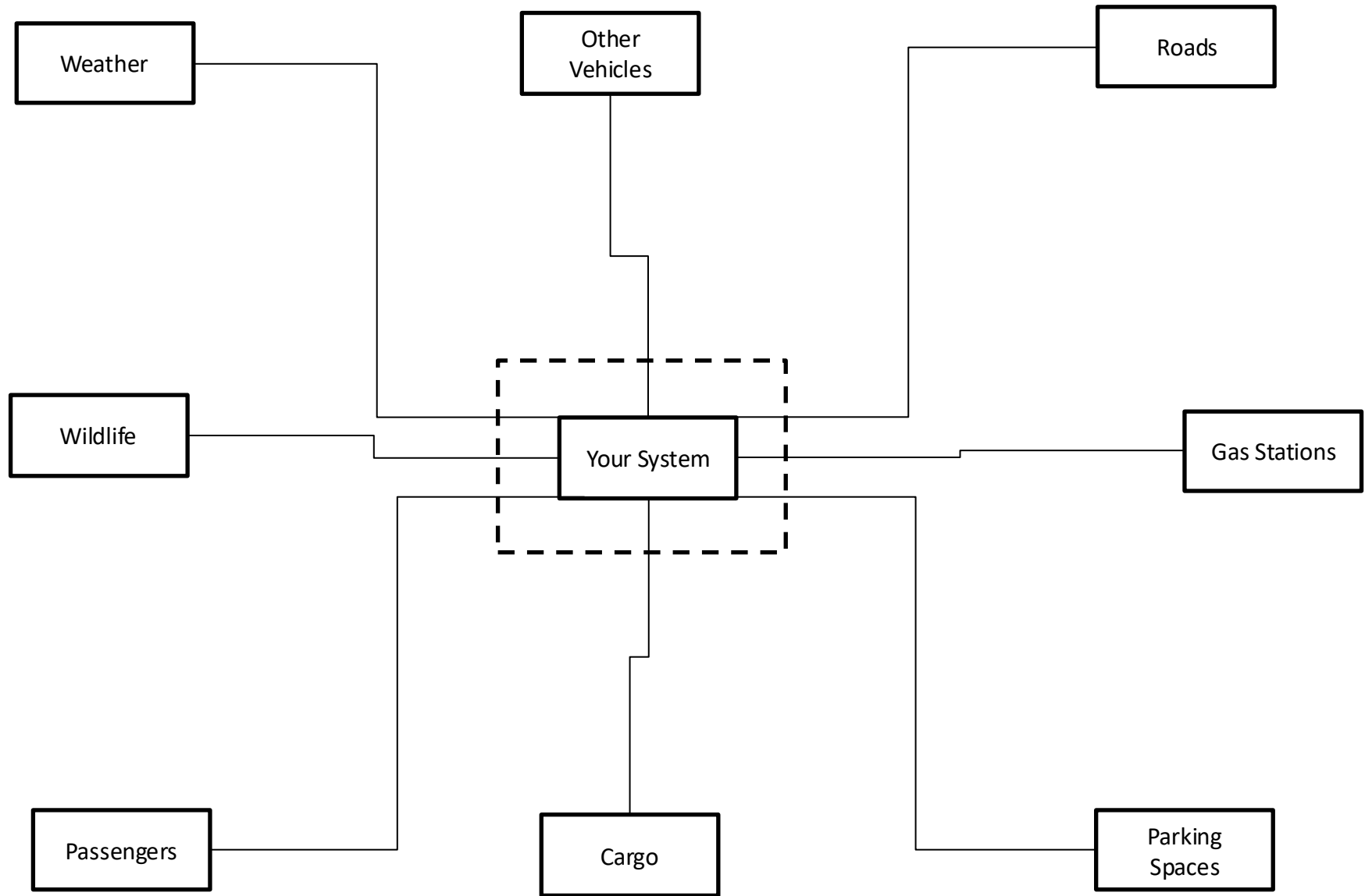
Passengers

Cargo

**This will help you stay focused on the interactions  
with other things without committing to what  
your System is/has structurally**

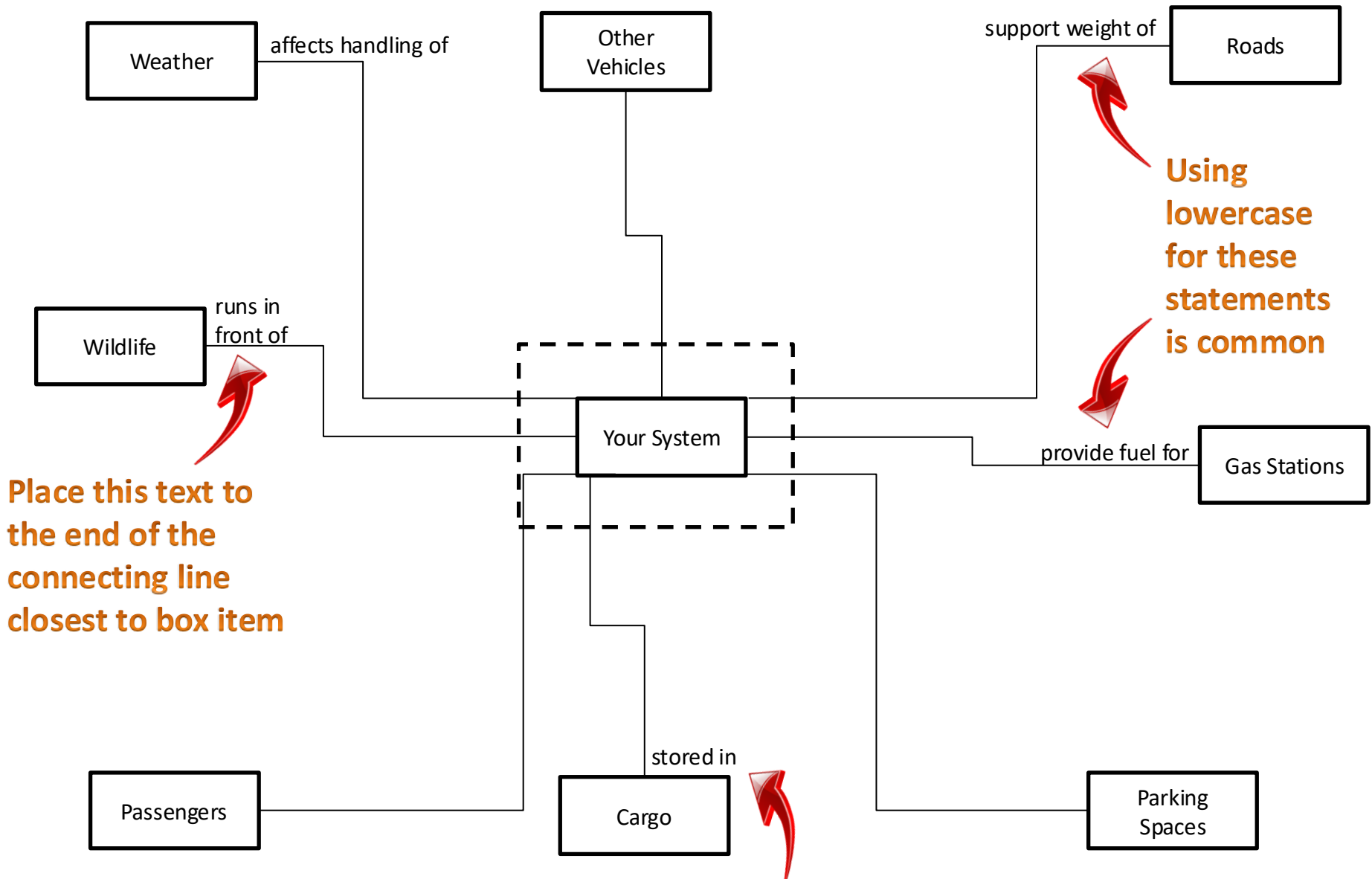
**Try trusting the tool and learn about its  
value, before trying to break the tool.**

**Draw (rectangular) lines connecting the outer boxes with your system**



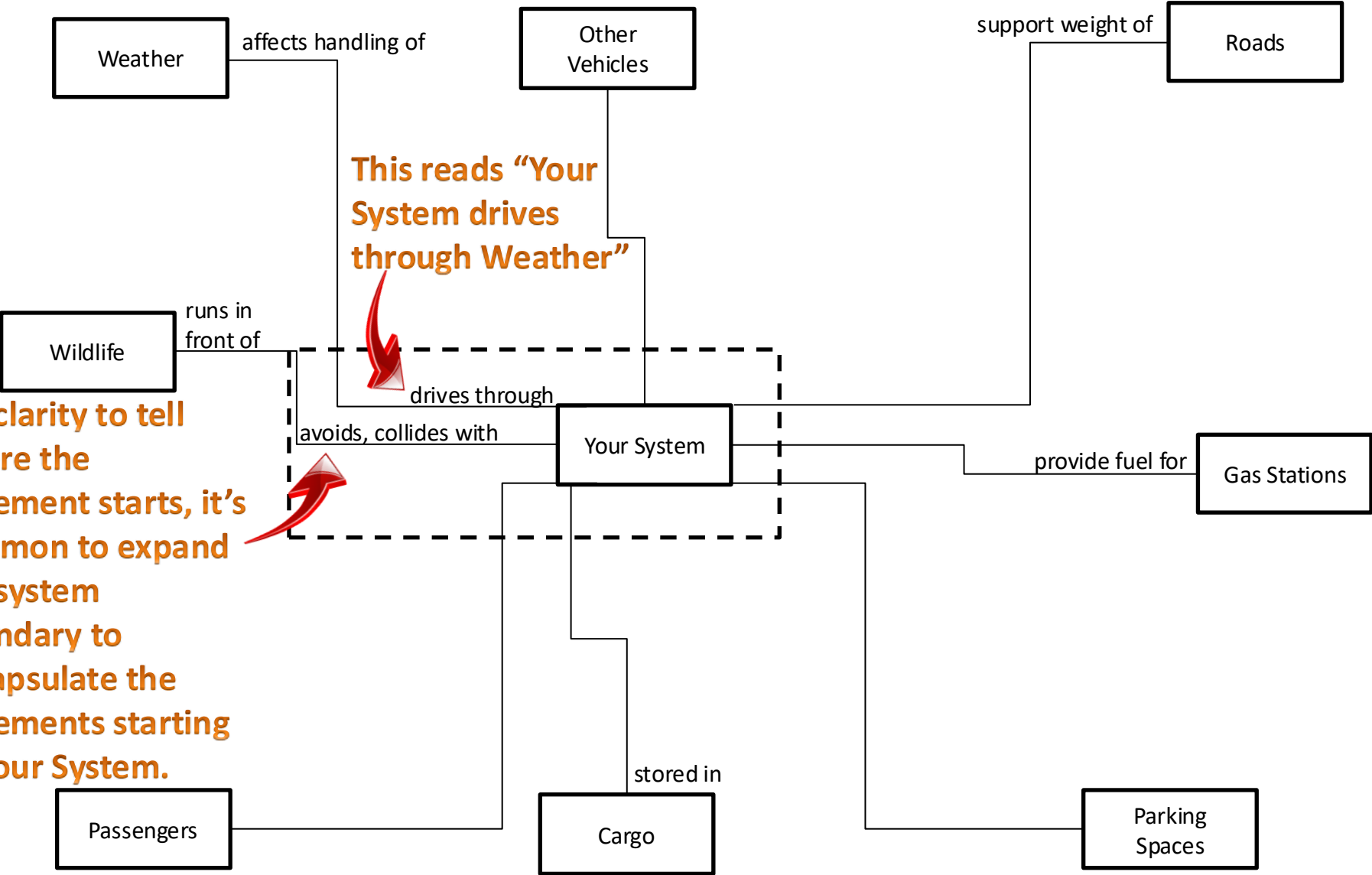
**It's helpful in the next step, if you space your boxes out around the System box**

Start to label the lines with text about how the box item interacts with the System

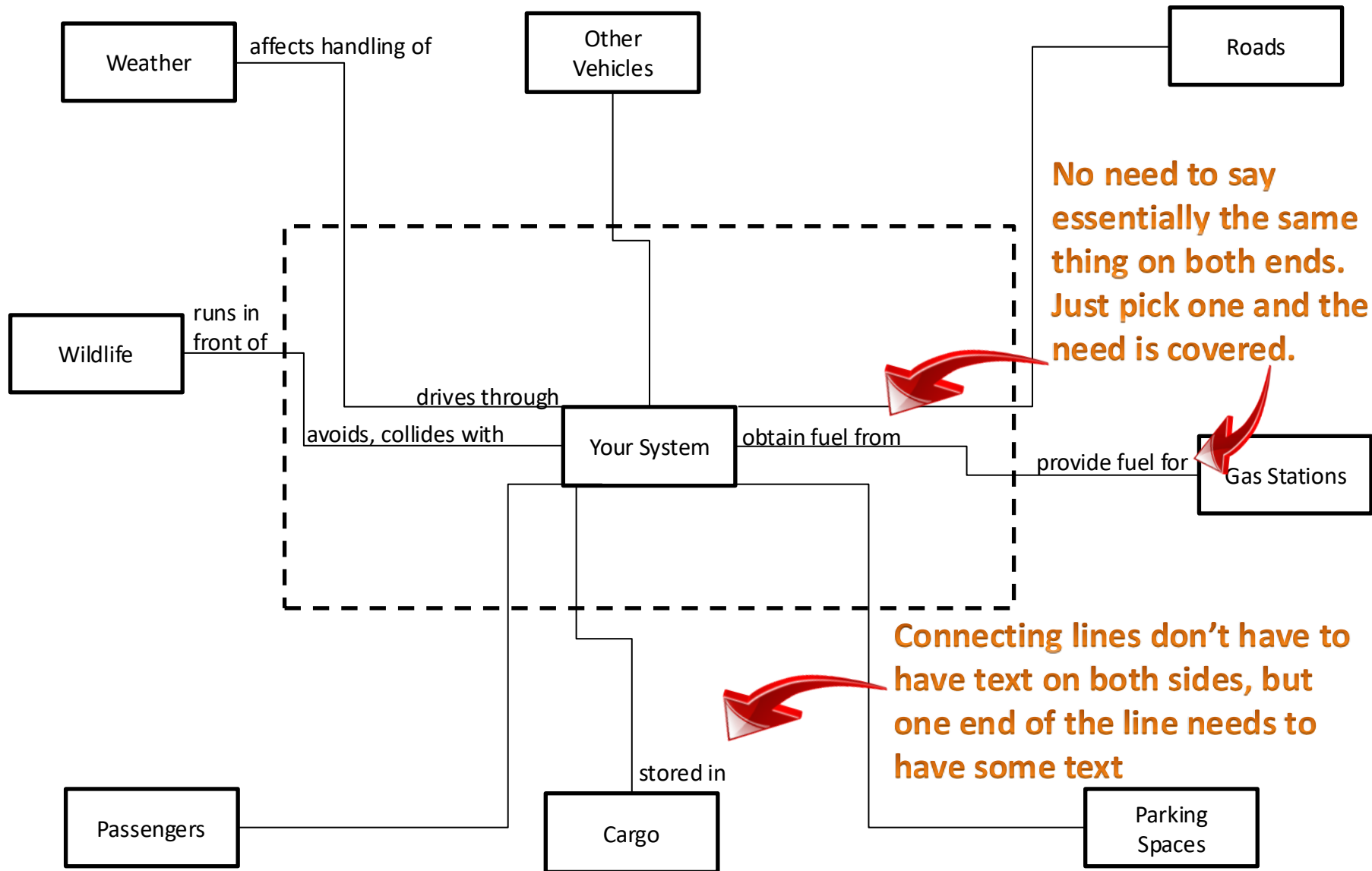


Write the text as if you were writing a short statement that started with the name of the box item and ends with your System: Cargo is “stored in” Your System

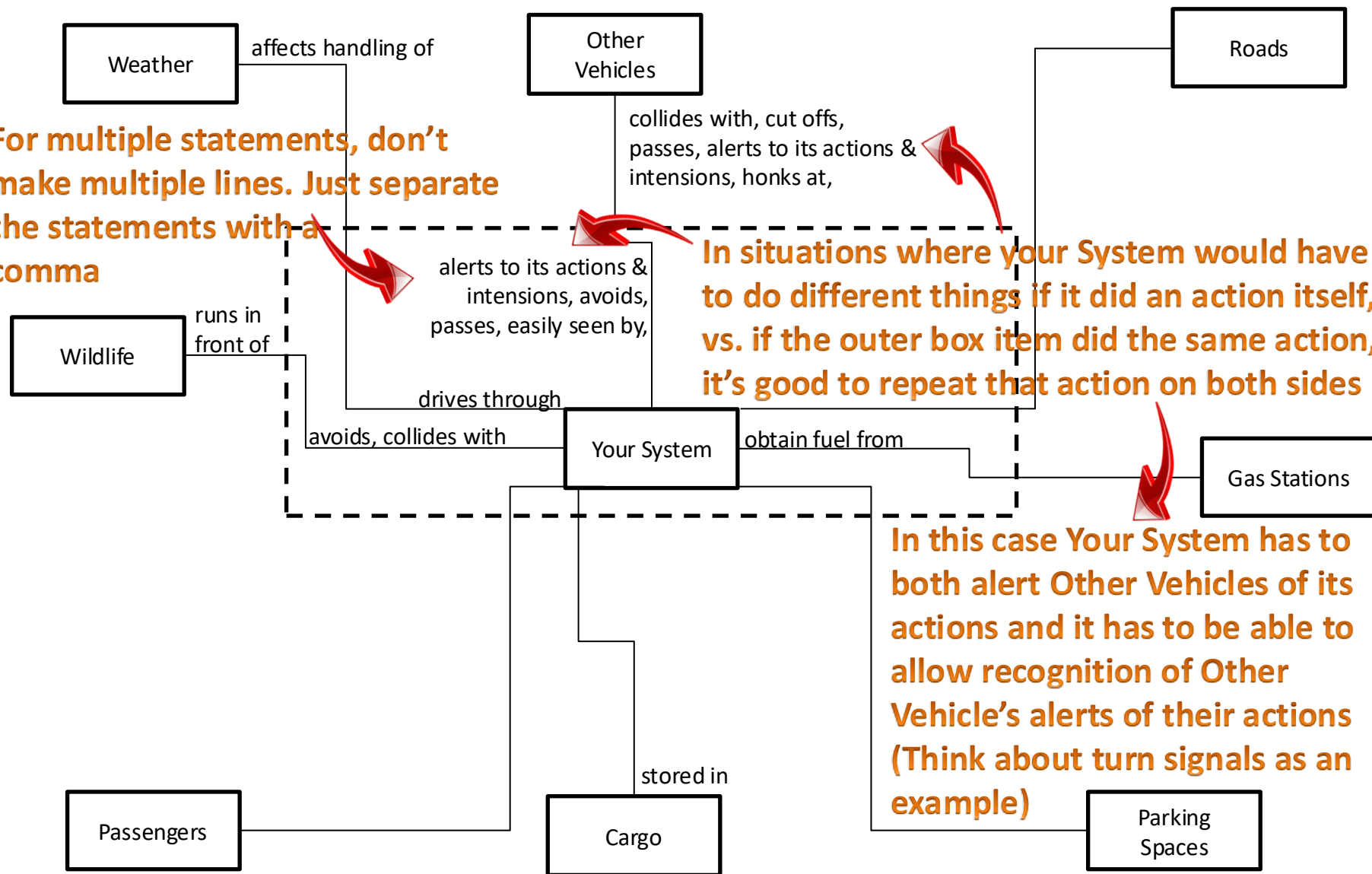
And at the same time label the lines with text about how your System interacts with the box items...

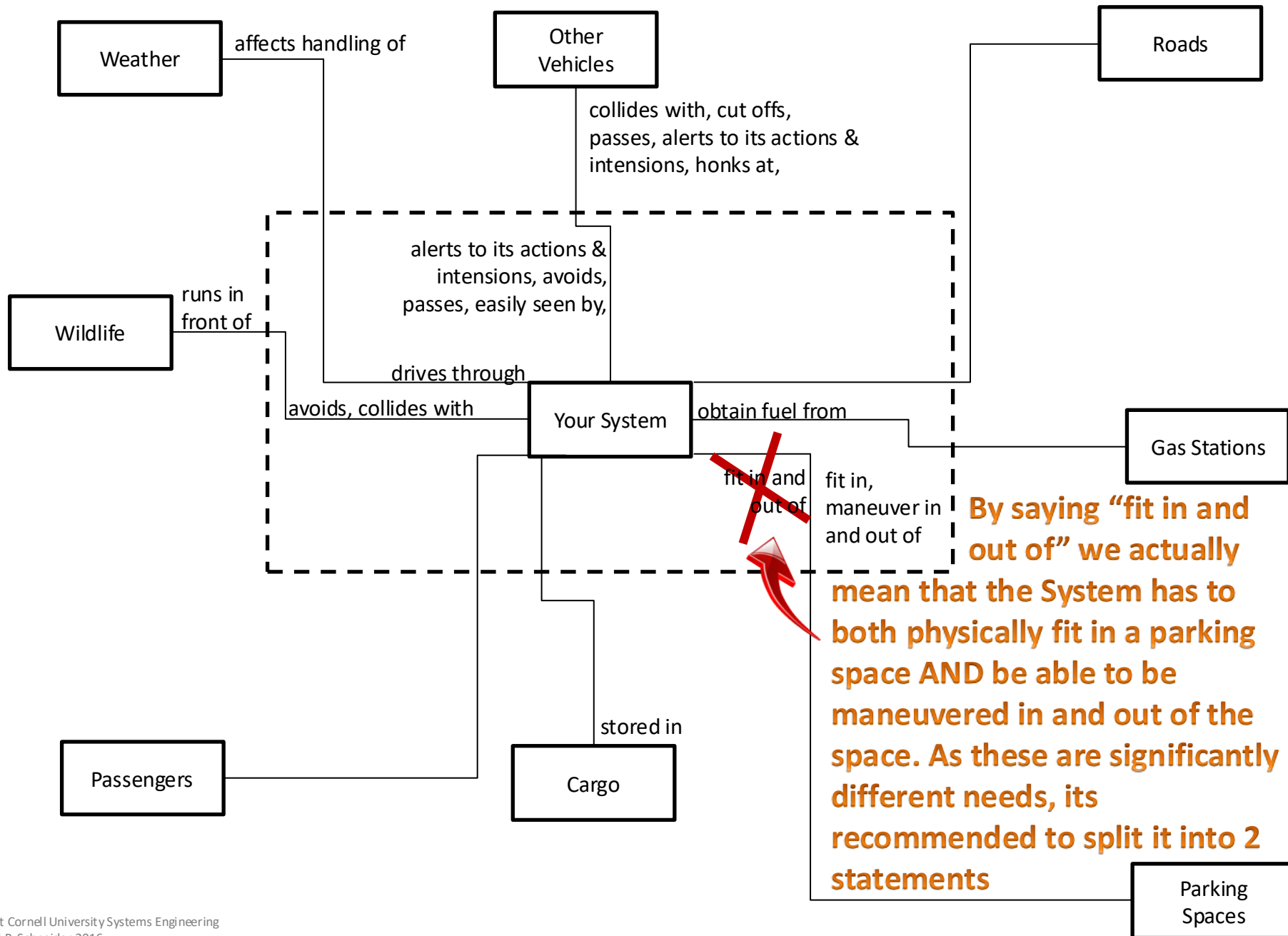


For clarity to tell where the statement starts, it's common to expand the system boundary to encapsulate the statements starting at your System.

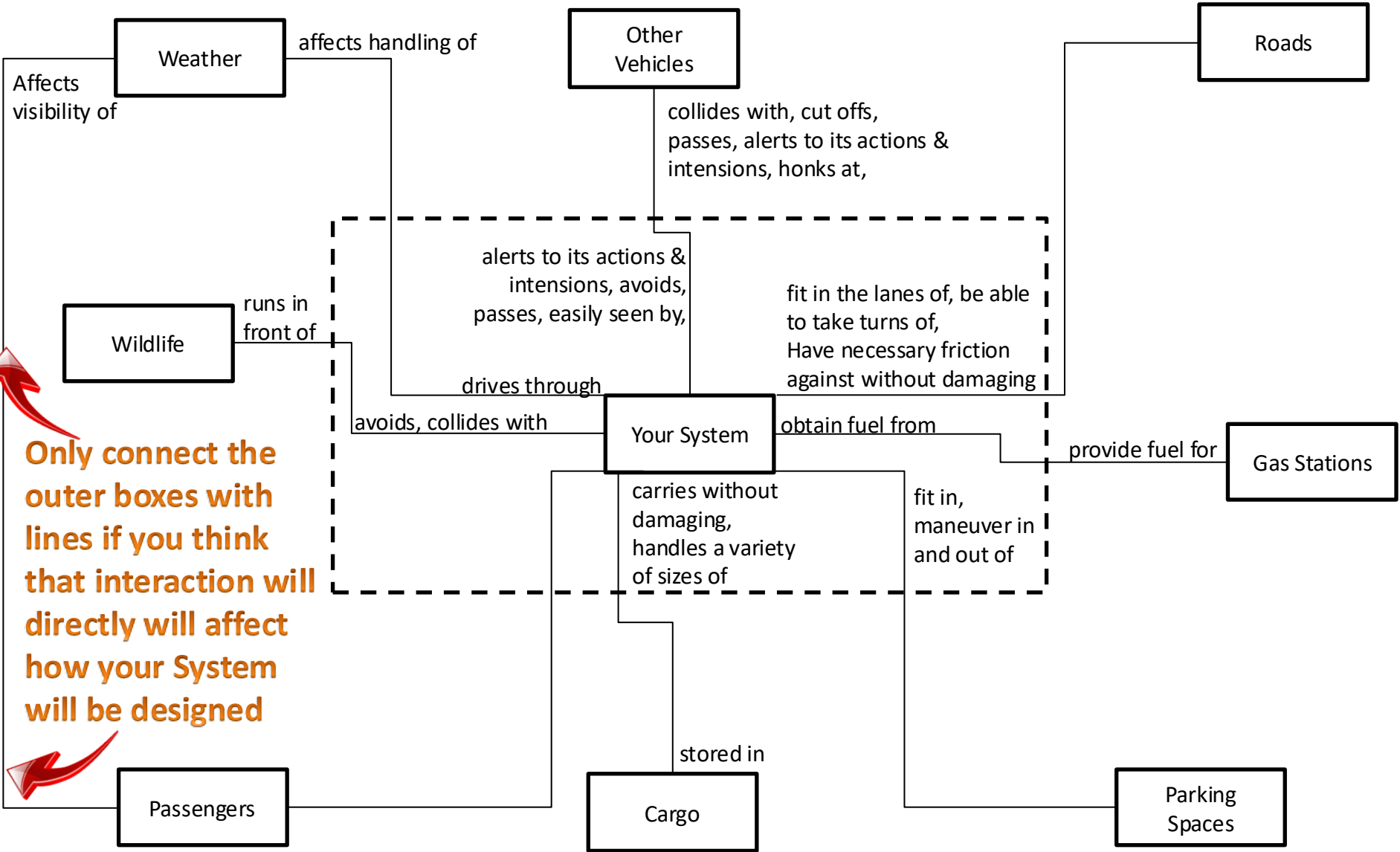


**For multiple statements, don't make multiple lines. Just separate the statements with a comma**

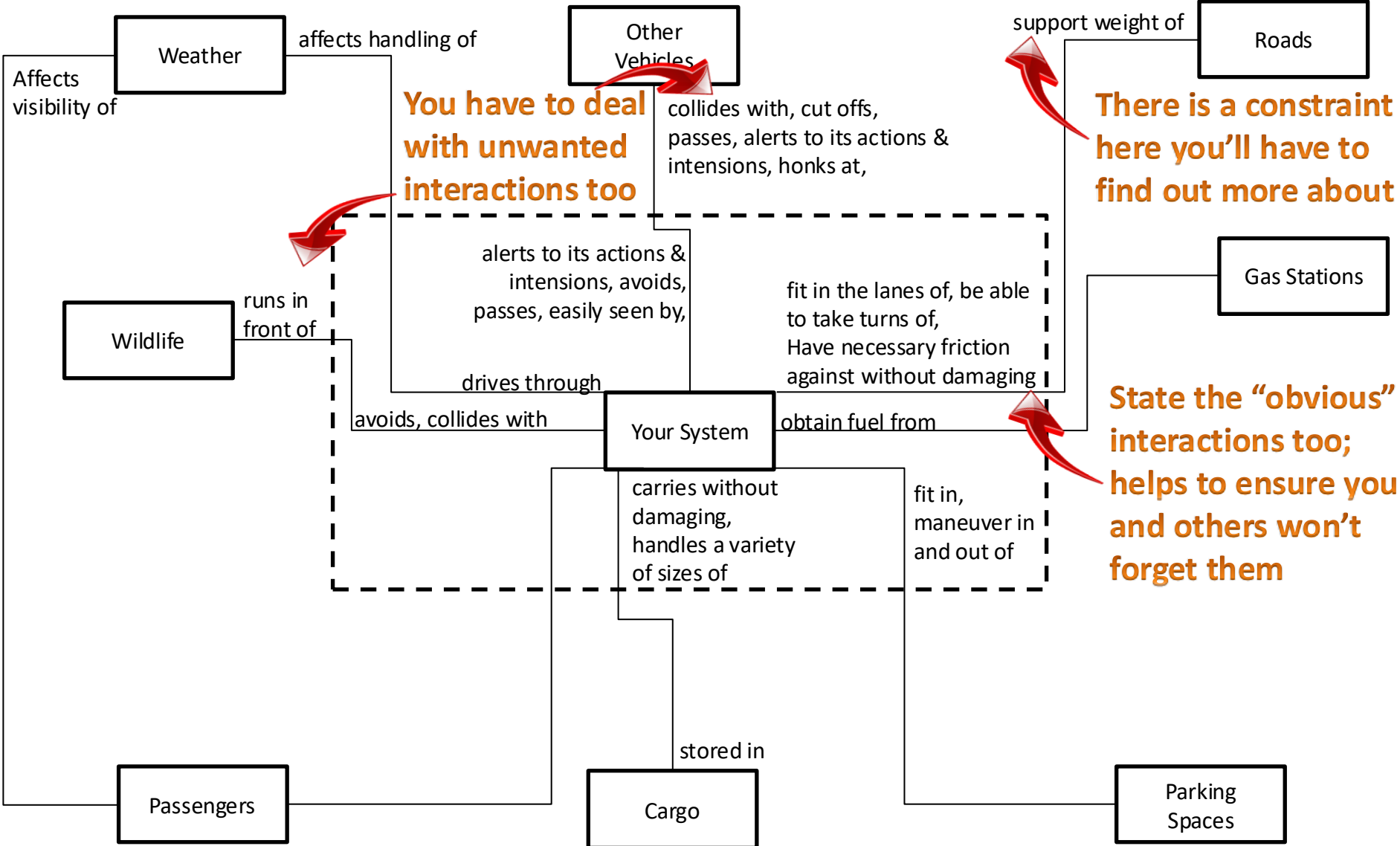




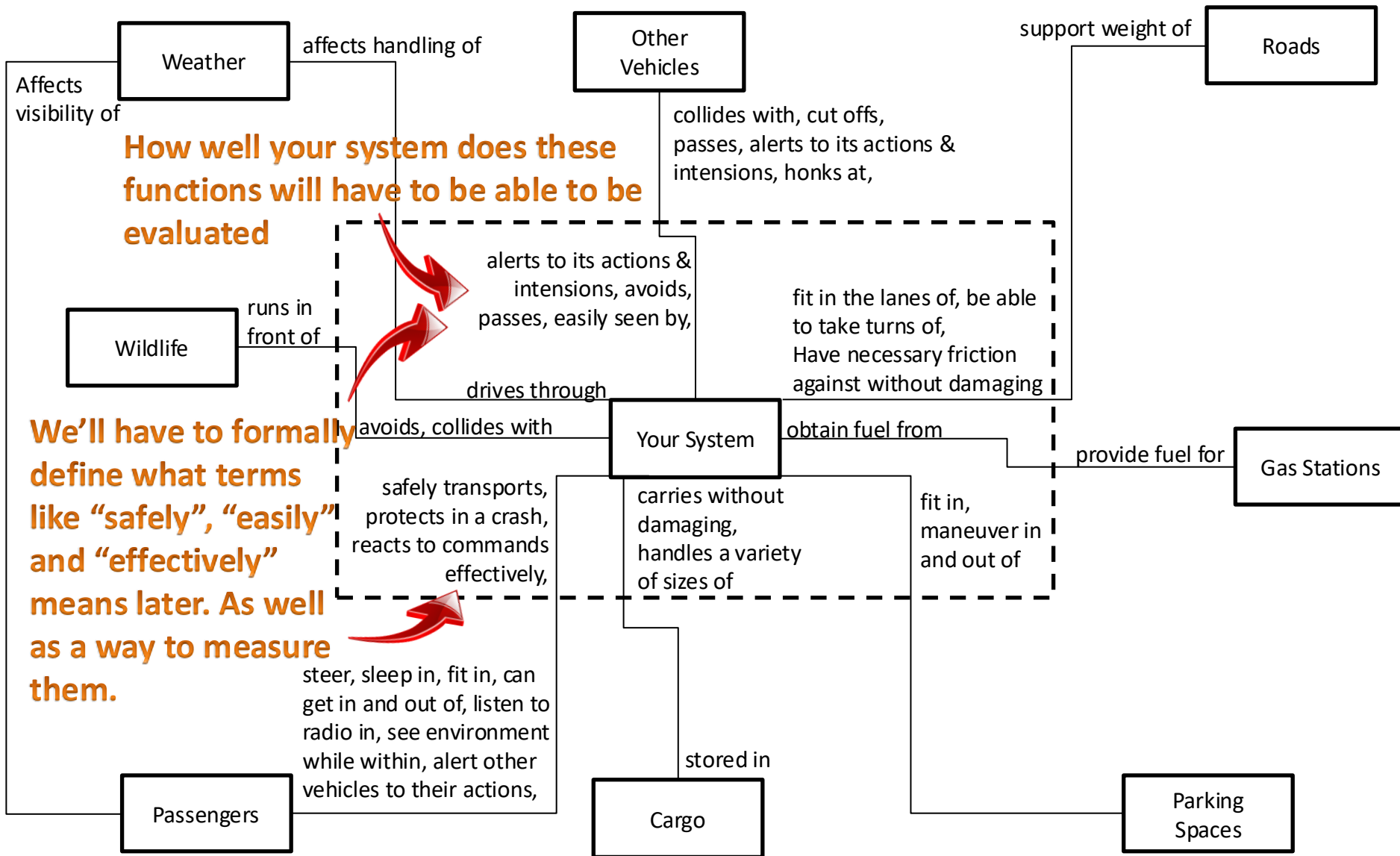
At the same time you can also write text on how the System interacts with the box items

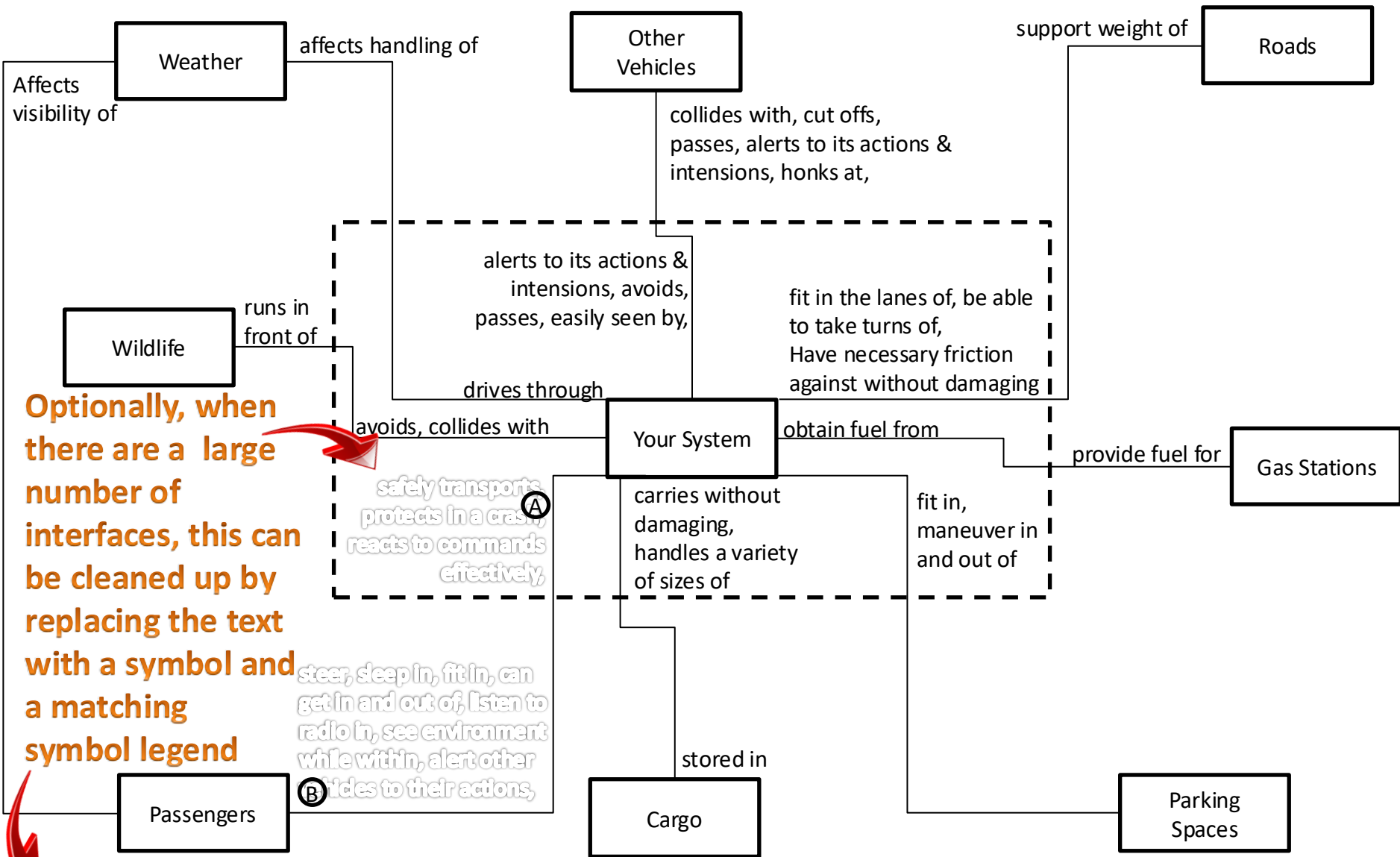


As you may have noticed, the way the outer boxes connect to your System informs needs that your System must meet...



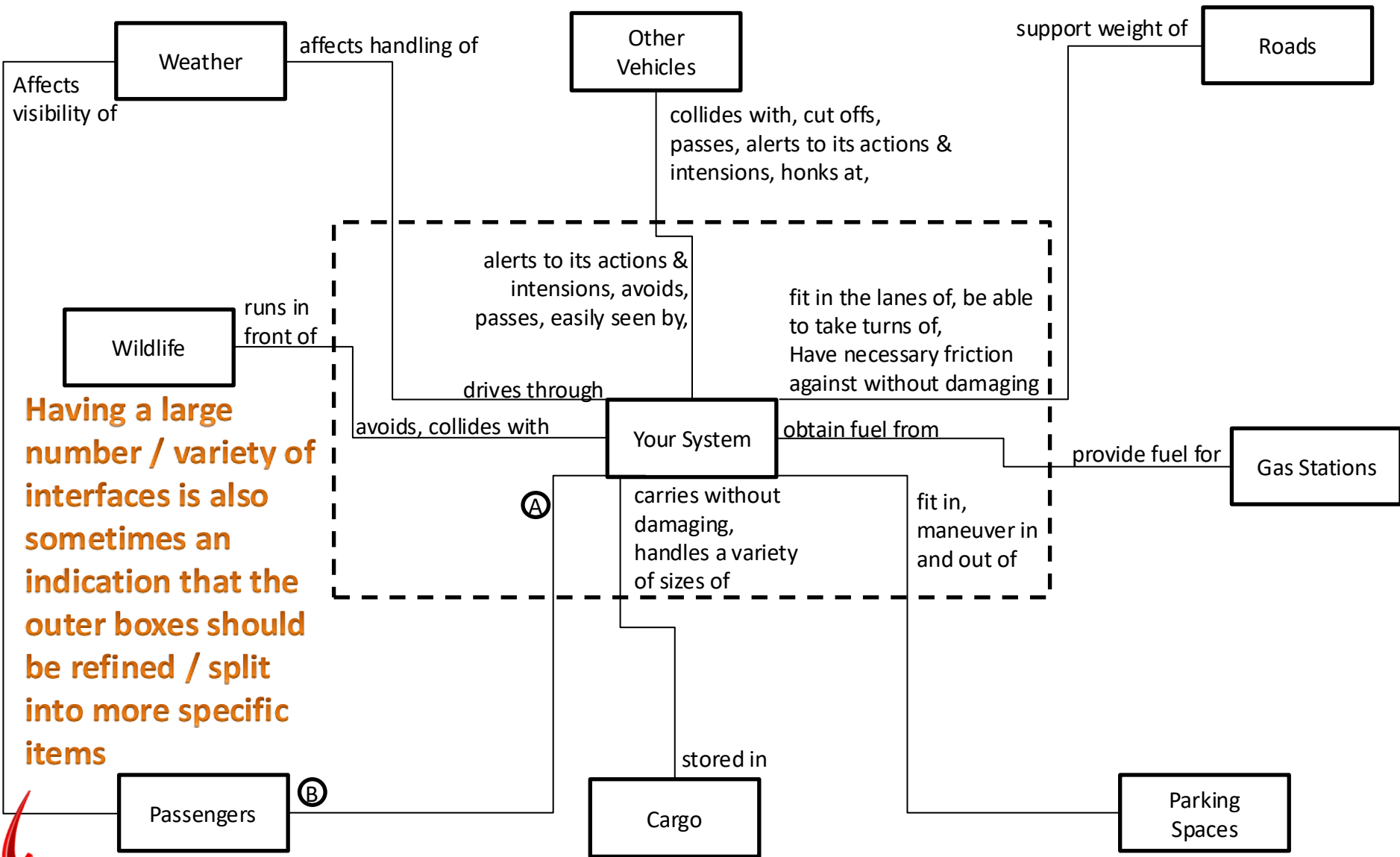
## Connections also help identify potential Performance Criteria...





Ⓐ safely transports, protects in a crash, reacts to commands effectively

Ⓑ steer, sleep in, fit in, can get in and out of, listen to radio in, see environment while within, alert other vehicles to their actions

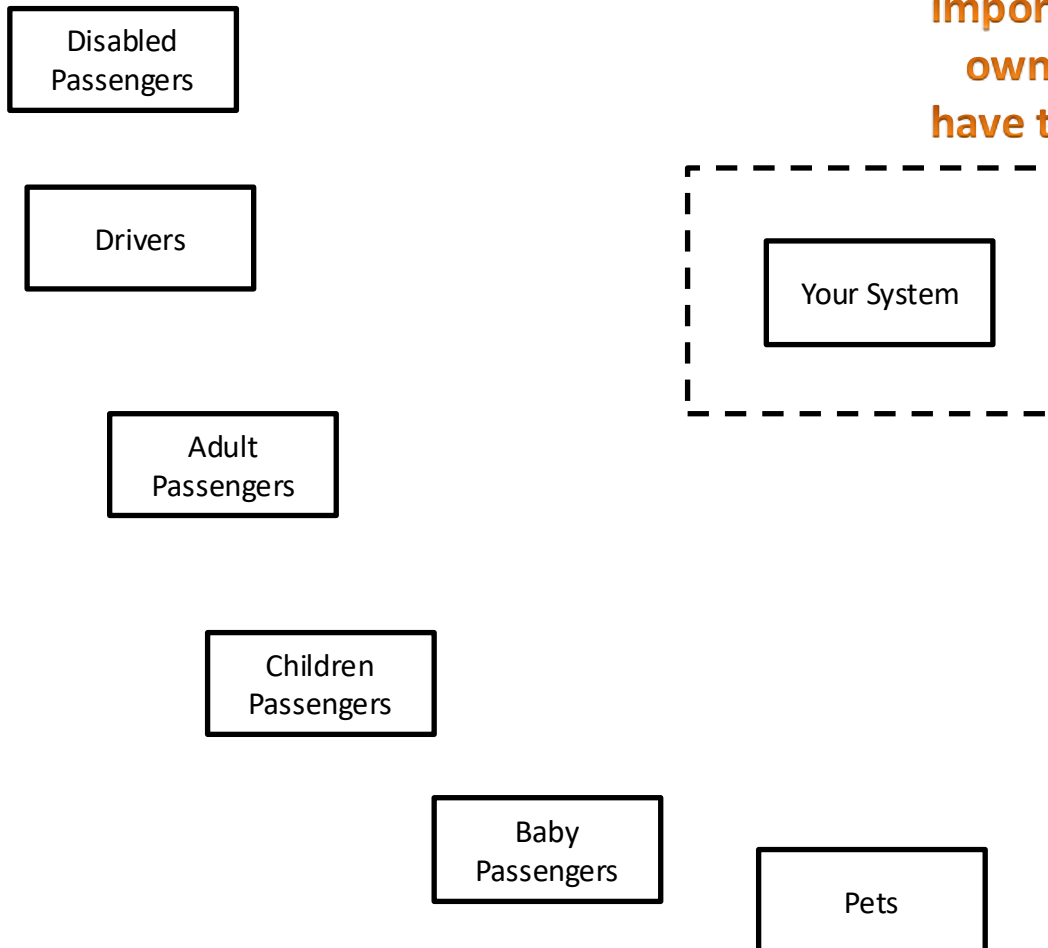


Ⓐ safely transports, protects in a crash, reacts to commands effectively

Ⓑ steer, sleep in, fit in, can get in and out of, listen to radio in, see environment while within, alert other vehicles to their actions

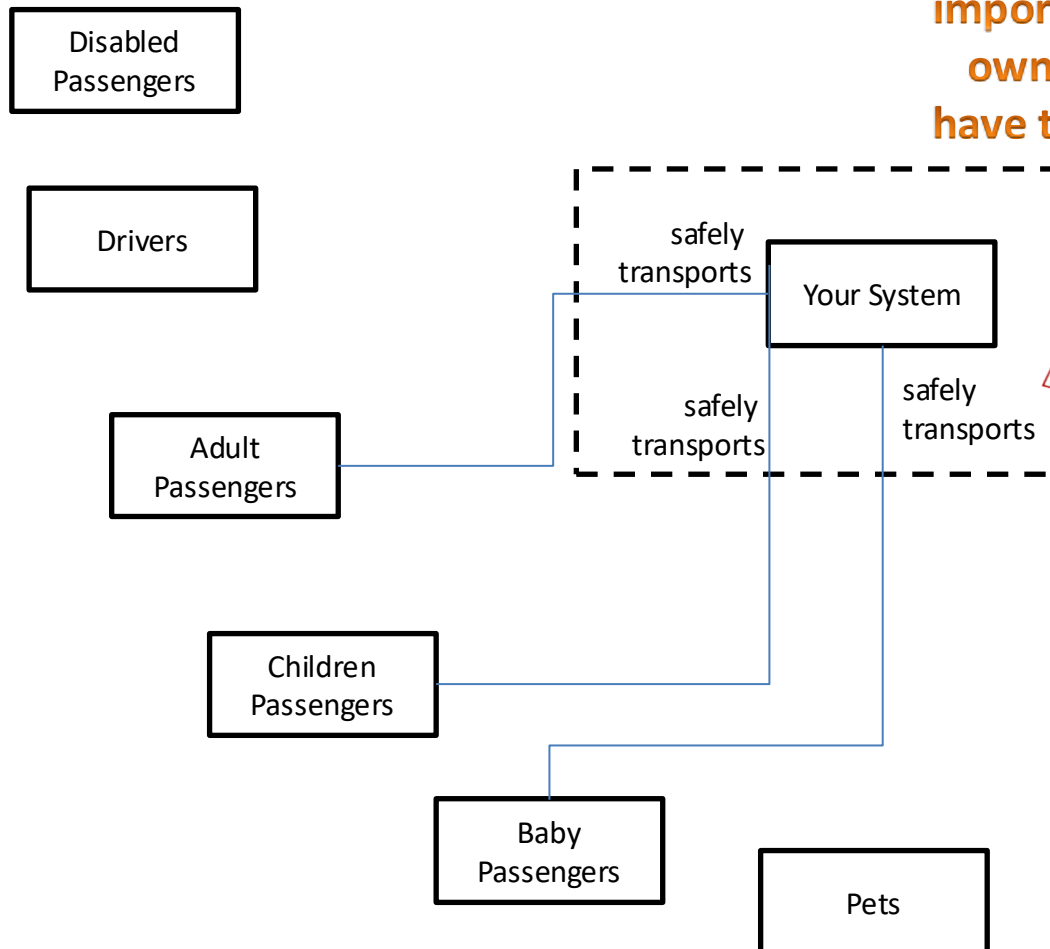
**Taking a step back to think more about these interactions reveals there are a number of important kinds of passengers**

**What makes different kinds of passengers important enough to separate into have their own box, is that each kind of passenger will have to be handled differently by the system.**



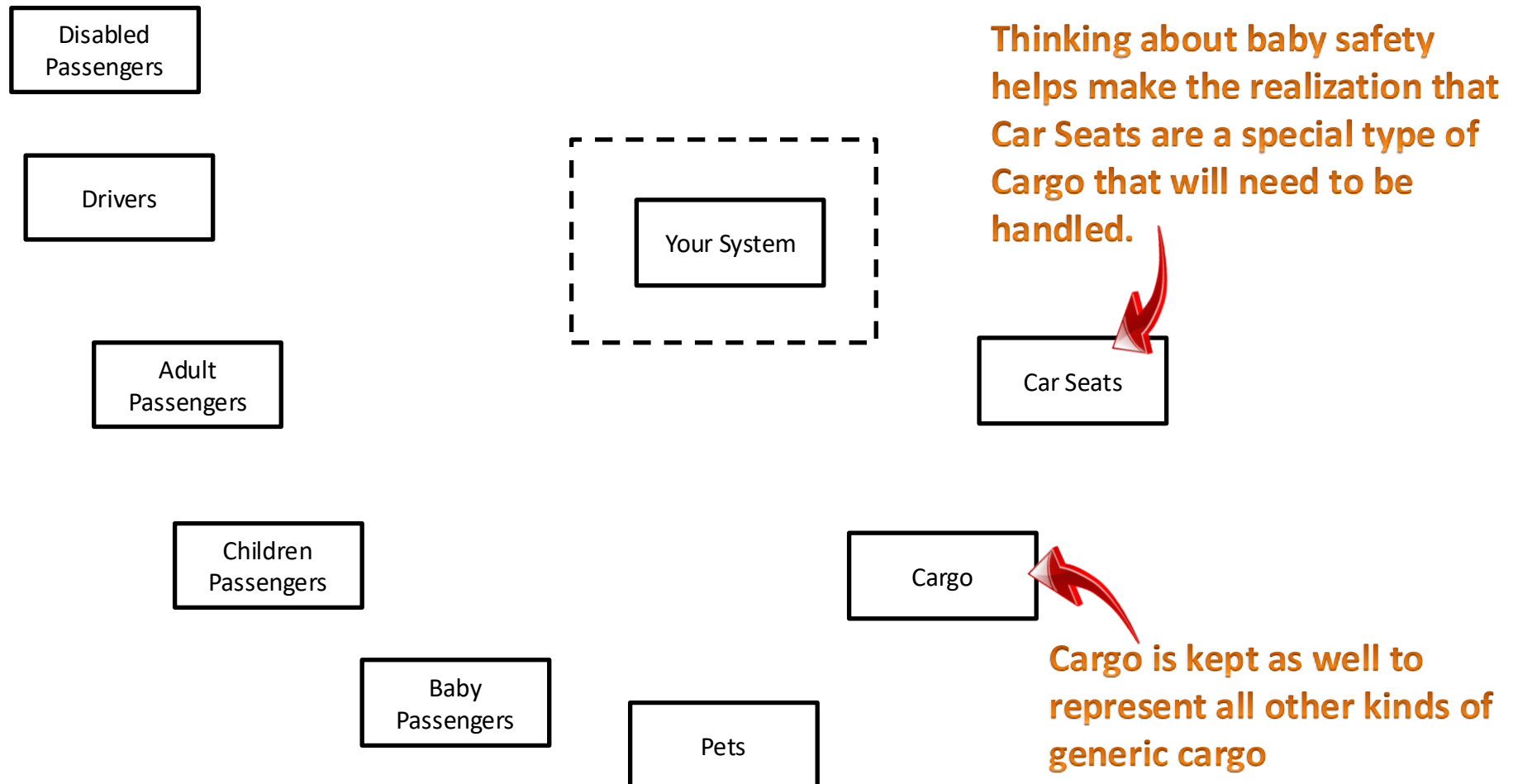
**Taking a step back to think more about these interactions reveals there are a number of important kinds of passengers**

**What makes different kinds of passengers important enough to separate into have their own box, is that each kind of passenger will have to be handled differently by the system.**



**One example interaction is shown although there are many. In order to safely transport each of these passenger types, the System will have to do something special.**

## This triggers further brainstorming...



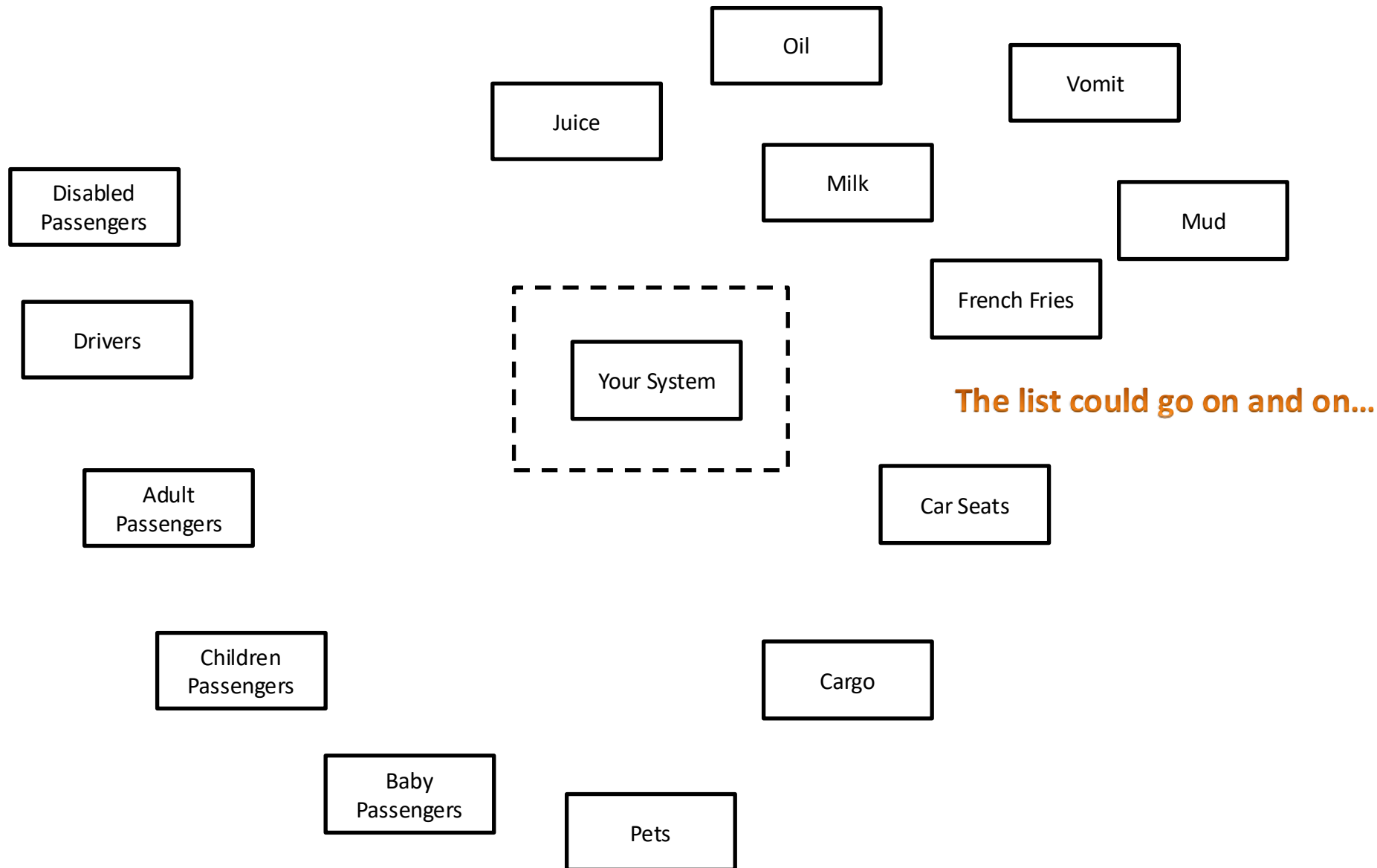
Thinking about baby safety helps make the realization that Car Seats are a special type of Cargo that will need to be handled.

Car Seats

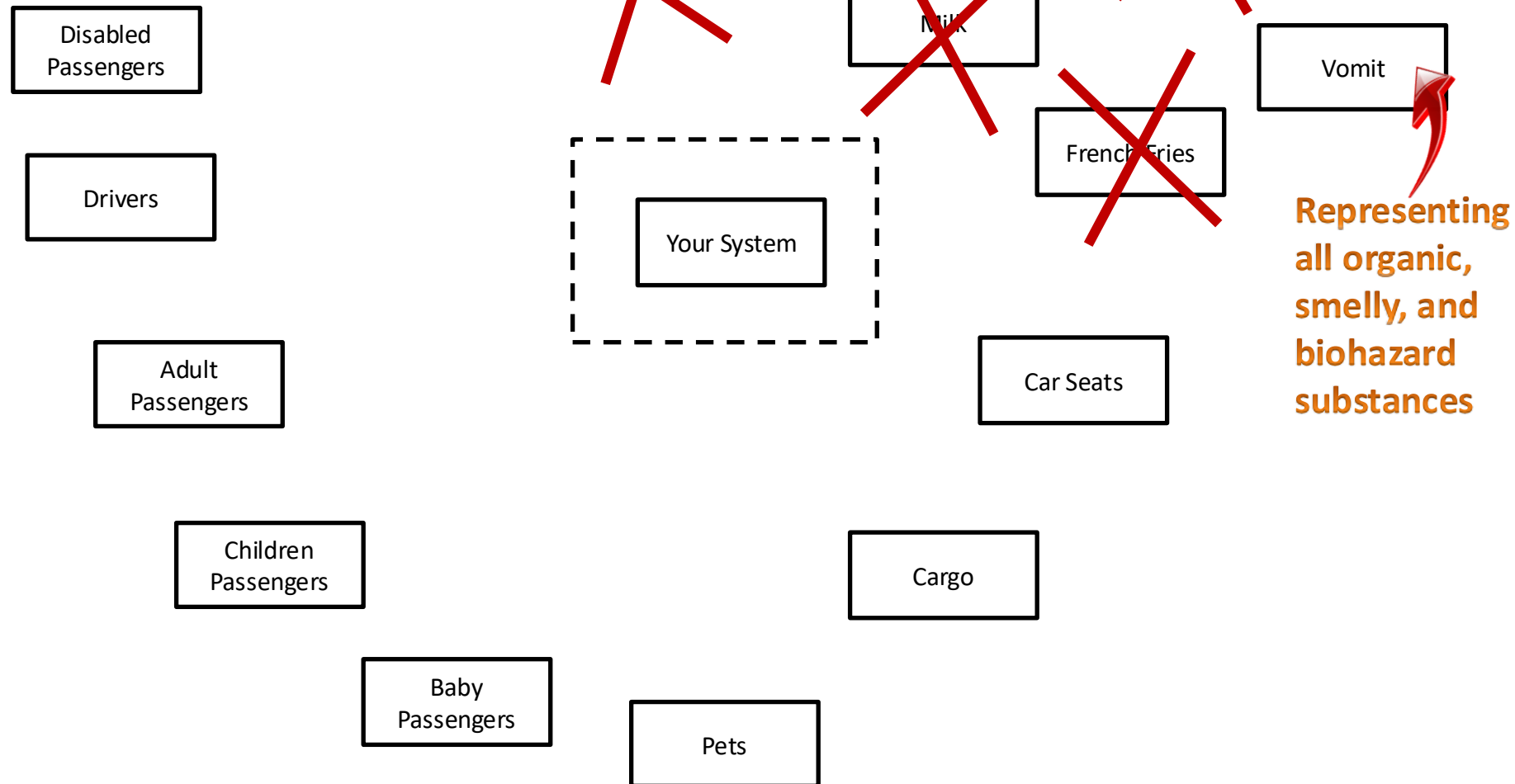
Cargo

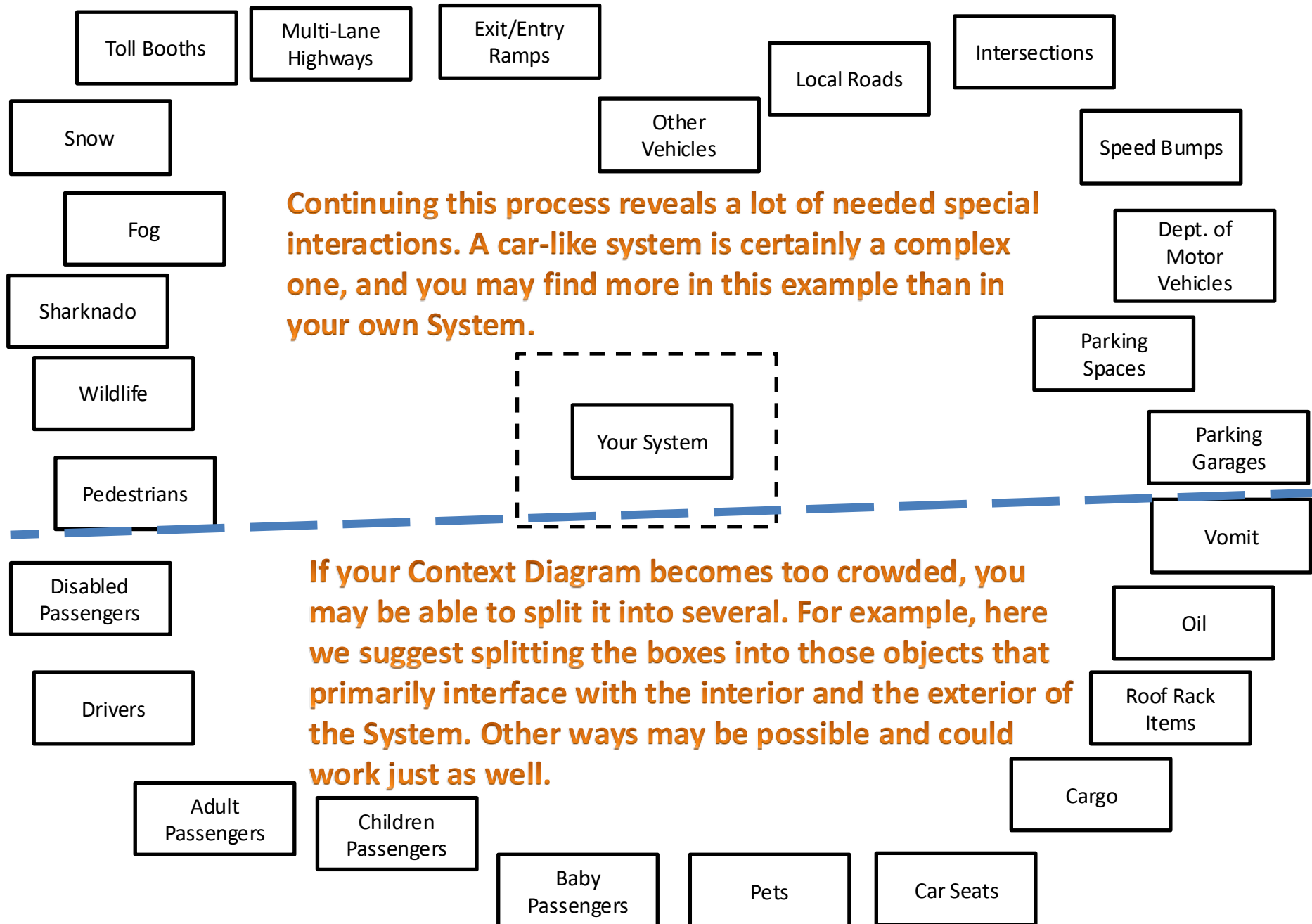
Cargo is kept as well to represent all other kinds of generic cargo

**Baby's make messes and so do pets, so  
we better think about all of the kinds of  
mess sources**



So instead of writing them all, we pick a subset such that if the System can handle that subset, it should be able to handle all the rest.





**Being thorough in your Context Diagram can be time consuming and even feel tedious if you're not used to using this tool. But there are common ways to break up the work**

**After you have done some initial work (as shown in slide 12) and you want to explore parts of the diagram further, this becomes a great time to start to split up the work with your teammates. Each person can then create their own context diagram(s) for the parts they explored.**

**You can try to combine them all together later if you want, but at least recognize which outer boxes show up in multiple teammate's work and compare the interactions discussed in all teammate's work to help everyone have a clear idea of the interaction needs.**