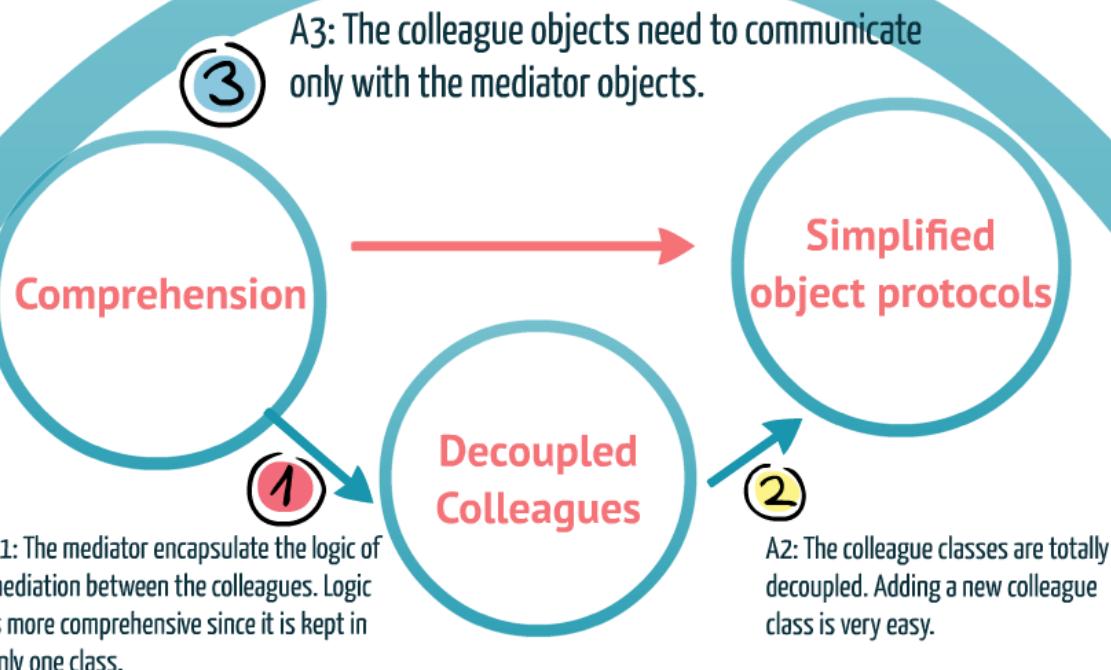
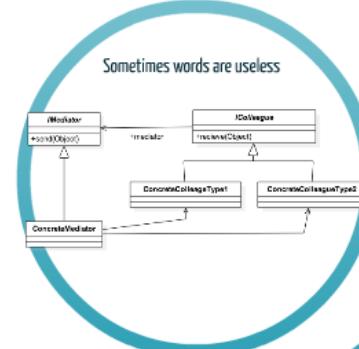
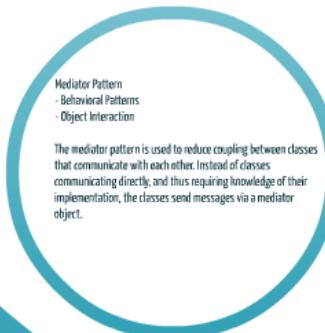


Welcome to the Mediator Pattern prezi



Pattern

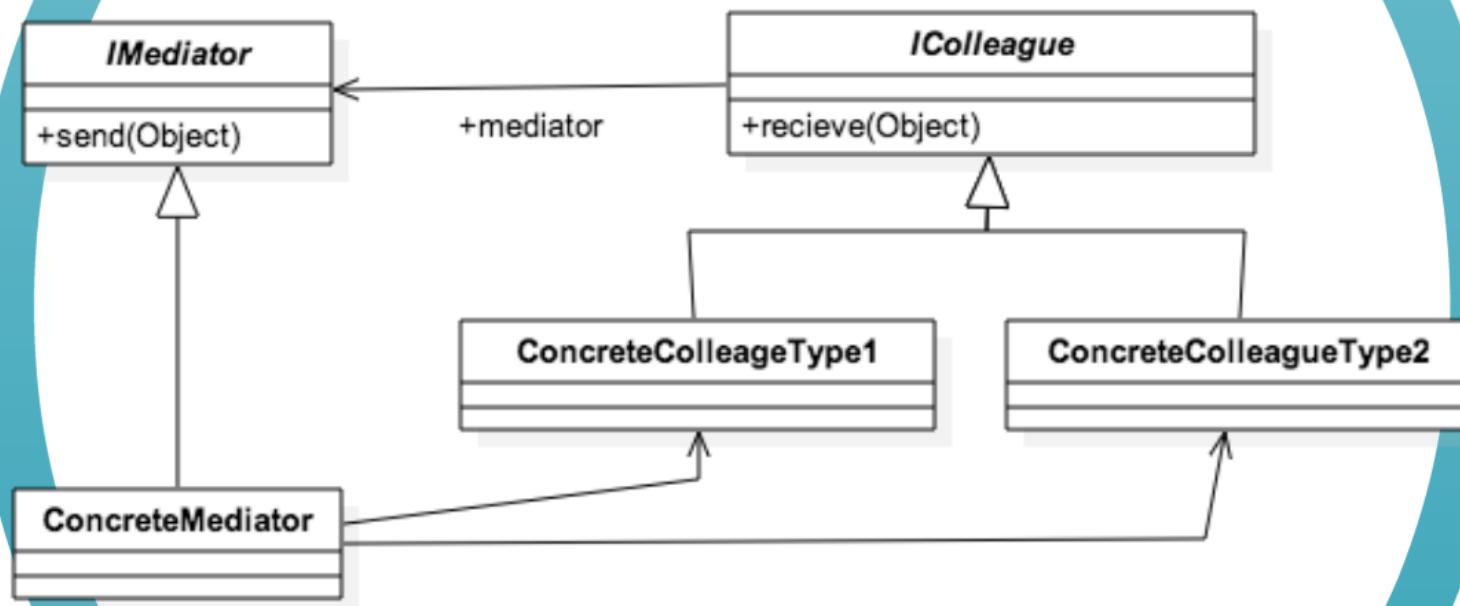


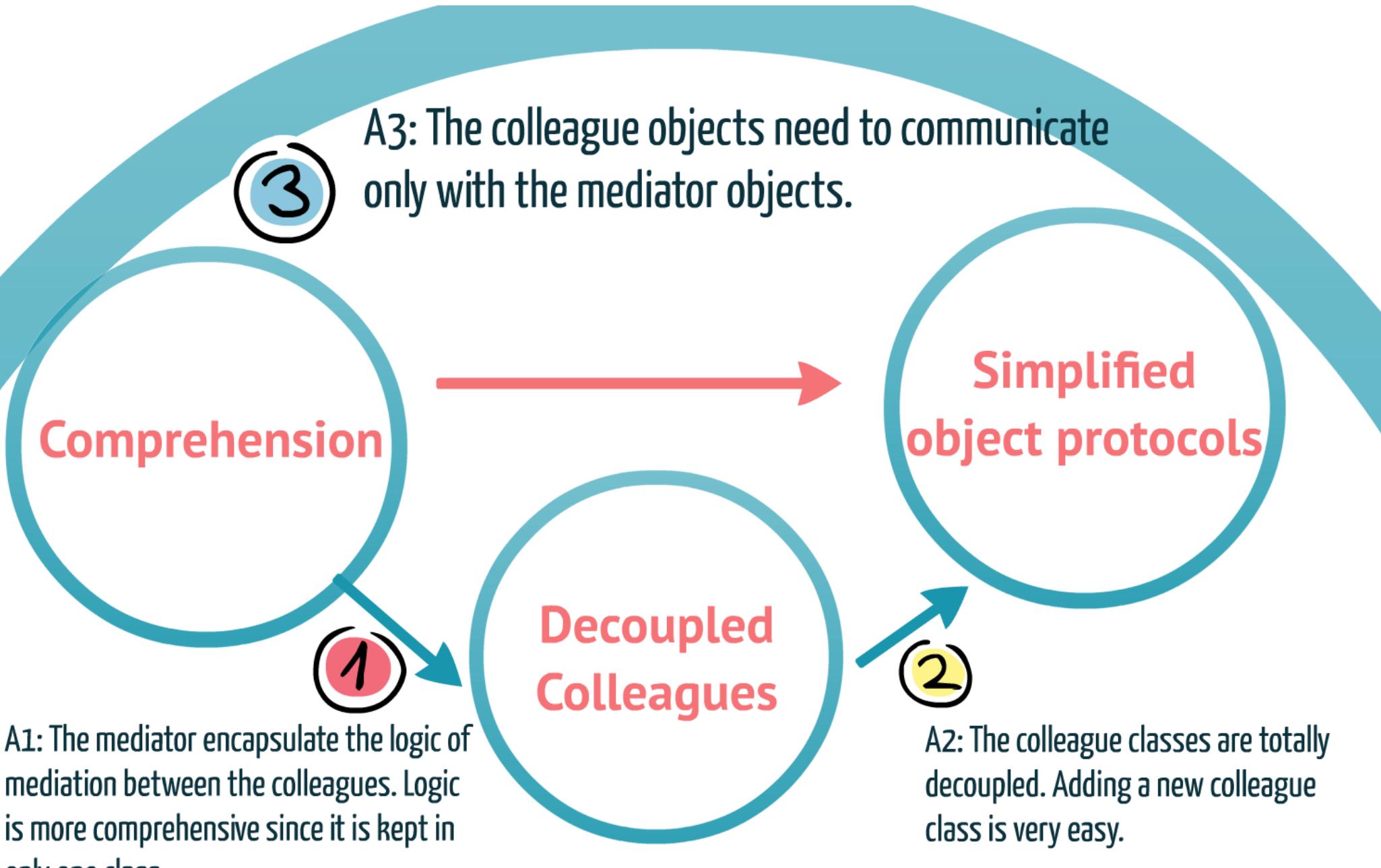
Mediator Pattern

- Behavioral Patterns
- Object Interaction

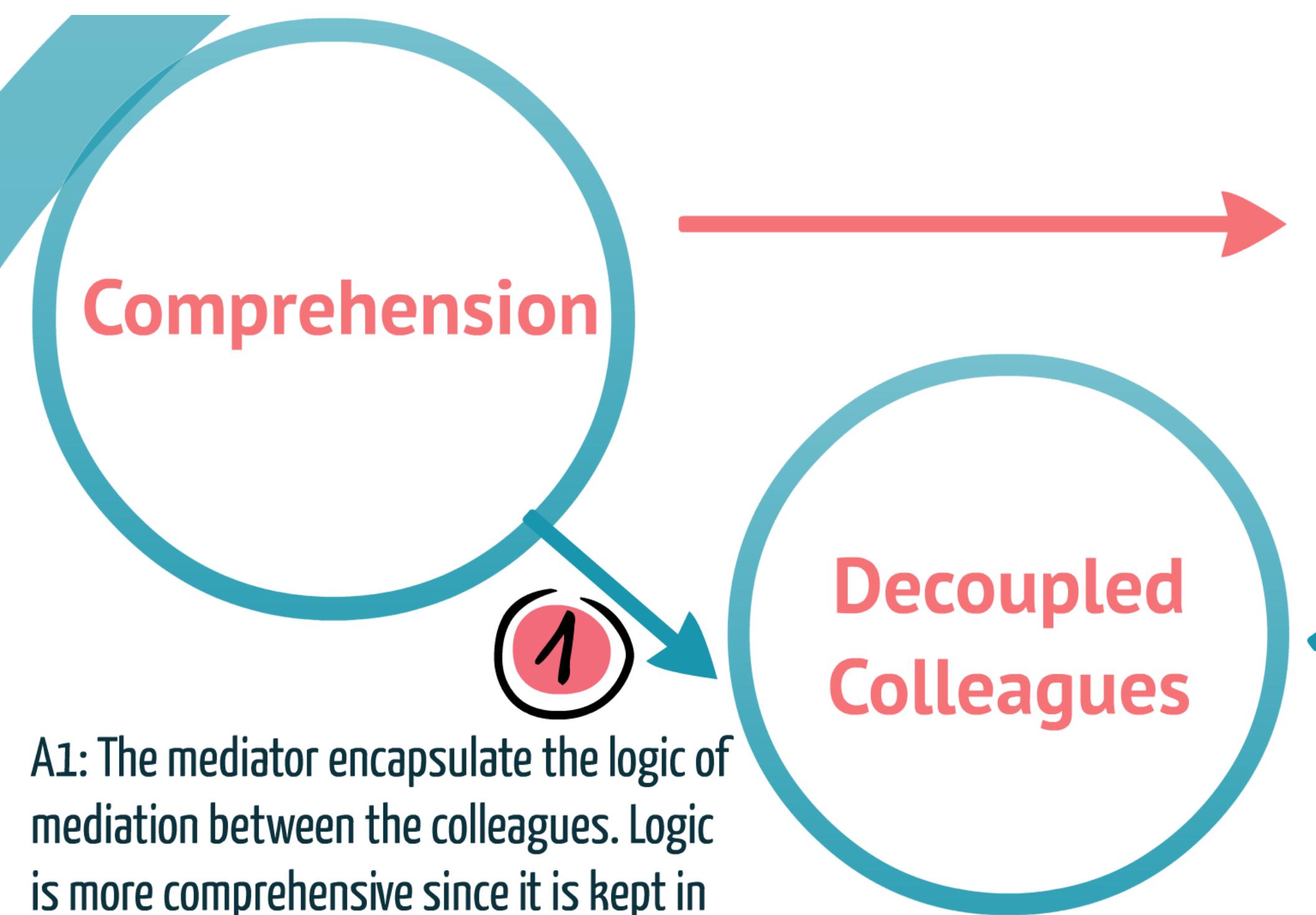
The mediator pattern is used to reduce coupling between classes that communicate with each other. Instead of classes communicating directly, and thus requiring knowledge of their implementation, the classes send messages via a mediator object.

Sometimes words are useless





Pattern



Comprehension

Decoupled
Colleagues

A1: The mediator encapsulate the logic of mediation between the colleagues. Logic is more comprehensive since it is kept in only one class.

in the mediator objects.

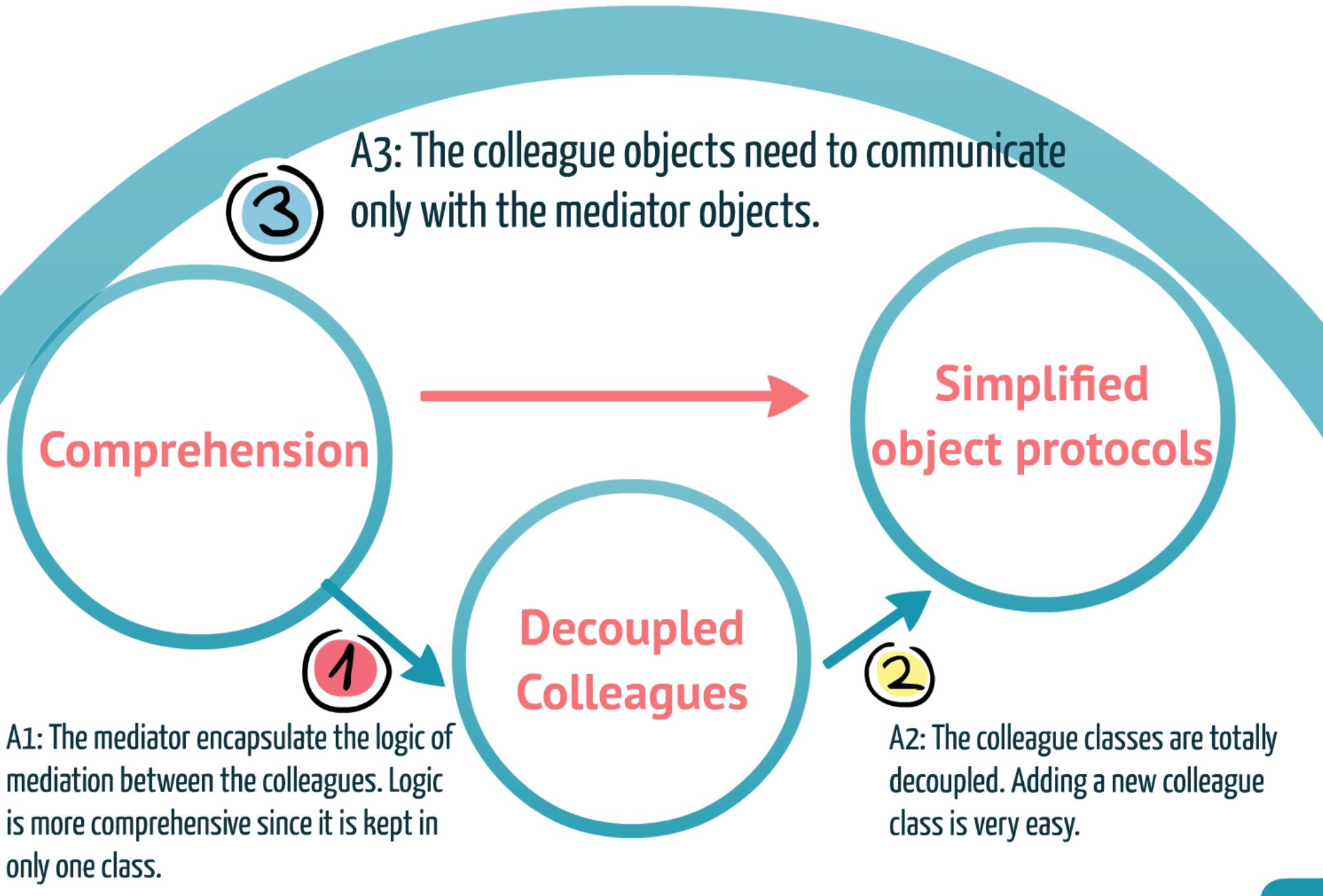
Decoupled
Colleagues



Simplified
object protocols



A2: The colleague classes are totally decoupled. Adding a new colleague class is very easy.



Circulation Problem

Imagine a piece of road with 

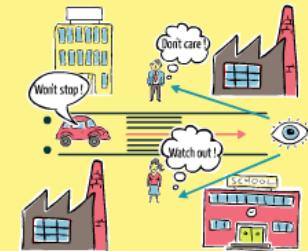
- trains,
- cars,
- pedestrians.

*Let's mediate all of this
and see what happens...*

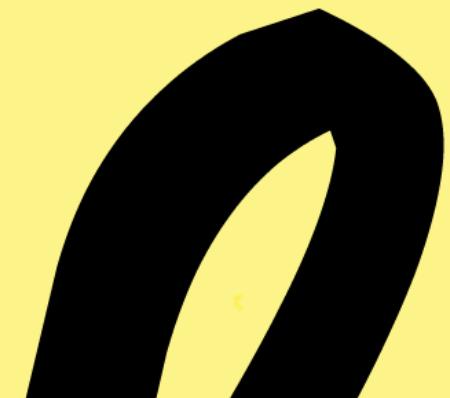
Circulation Problem

Imagine a piece of road with :

- trains,
- cars,
- pedestrians.



*Let's mediate all of this
and see what happens...*





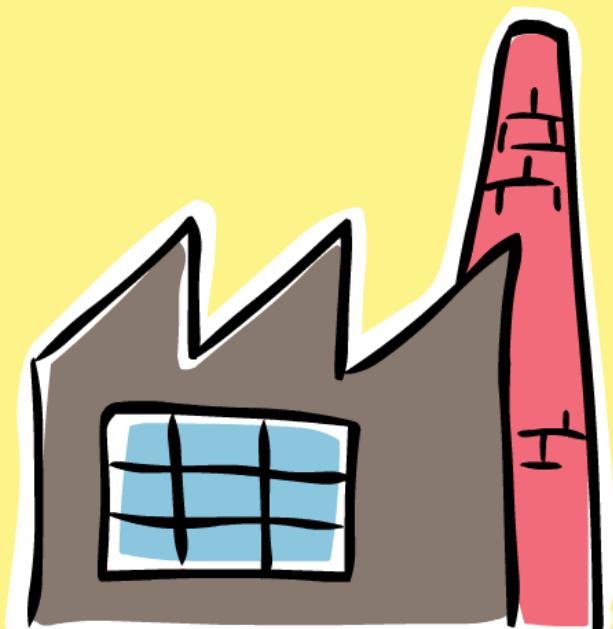
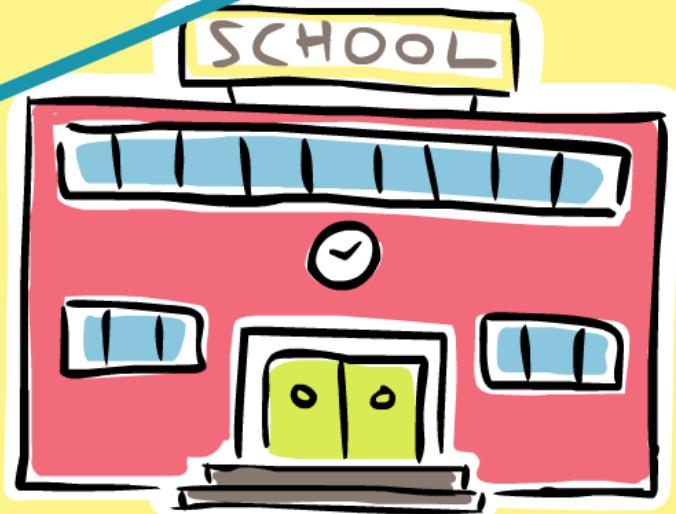
Won't stop!

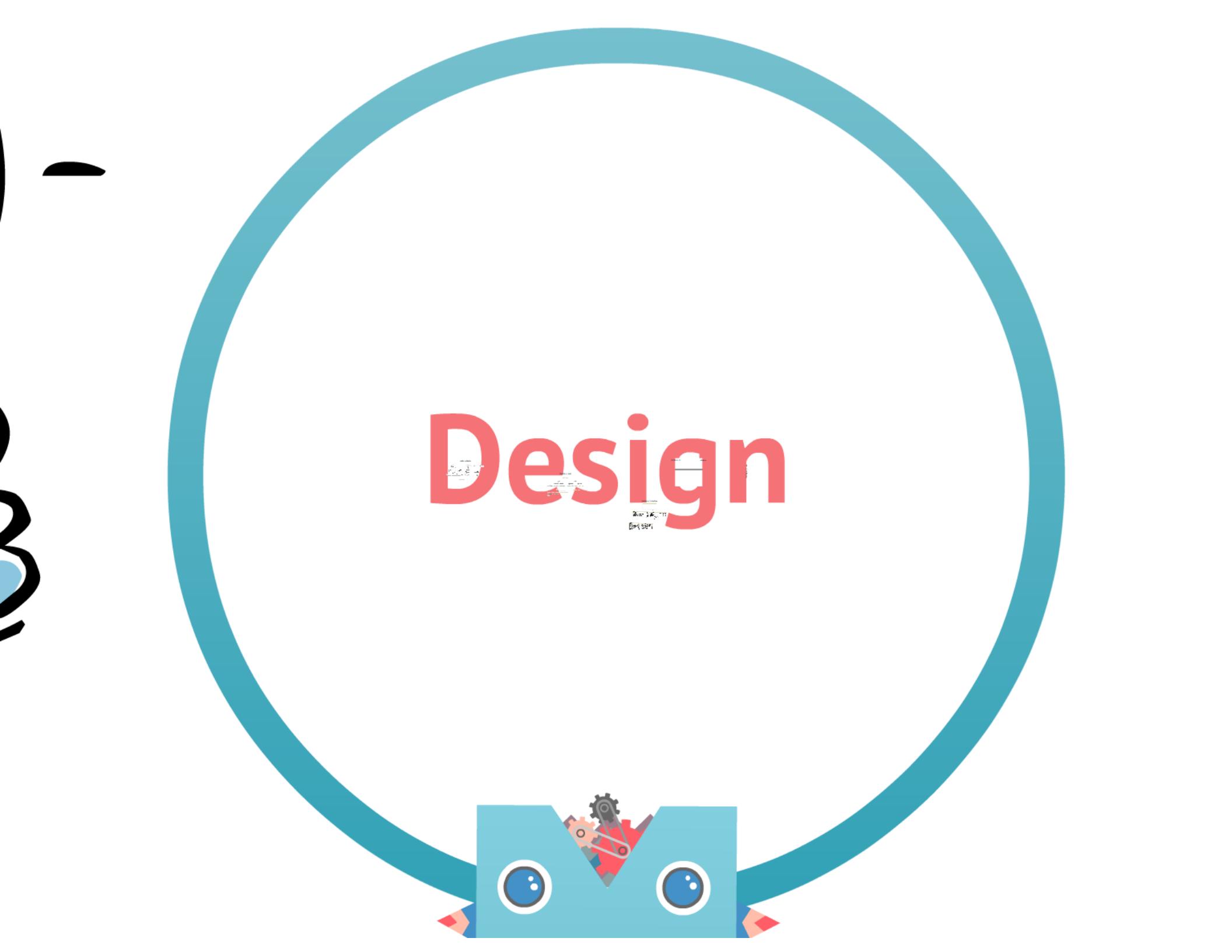


Don't care!



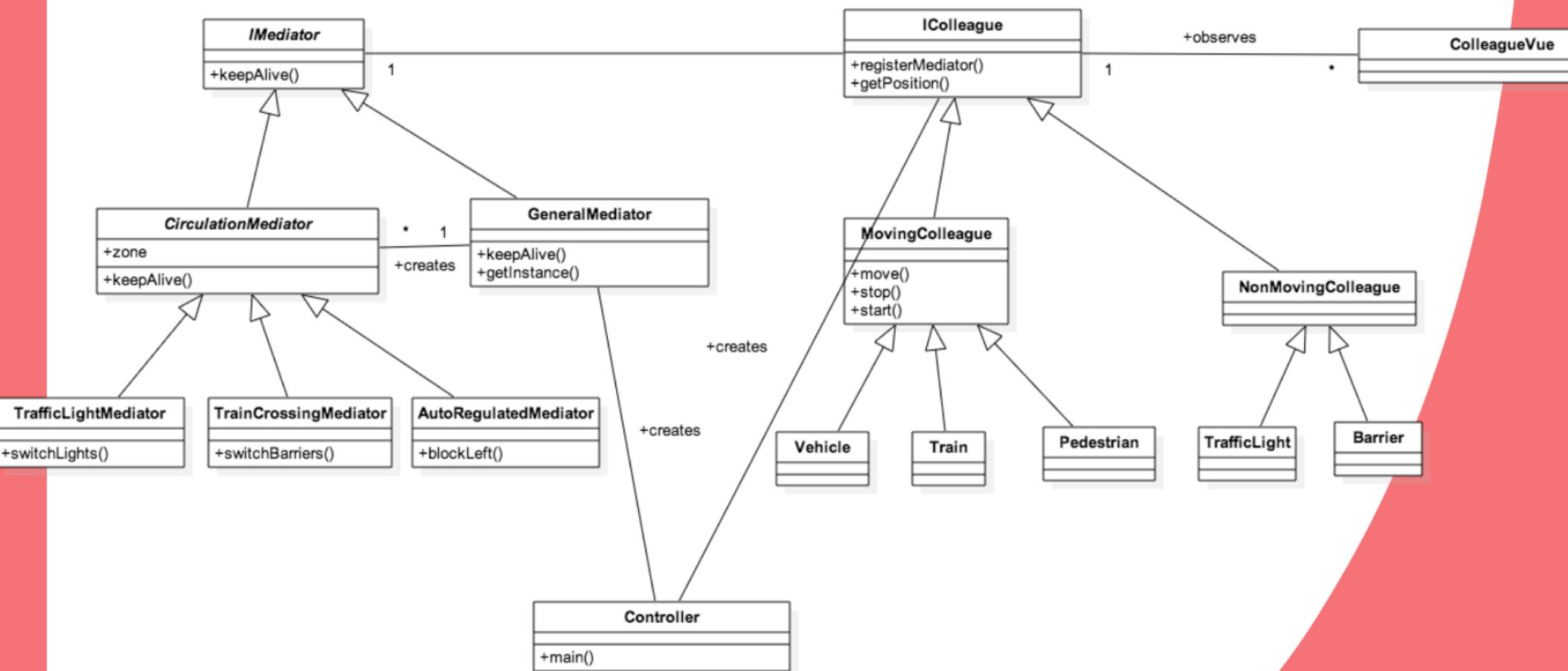
Watch out!



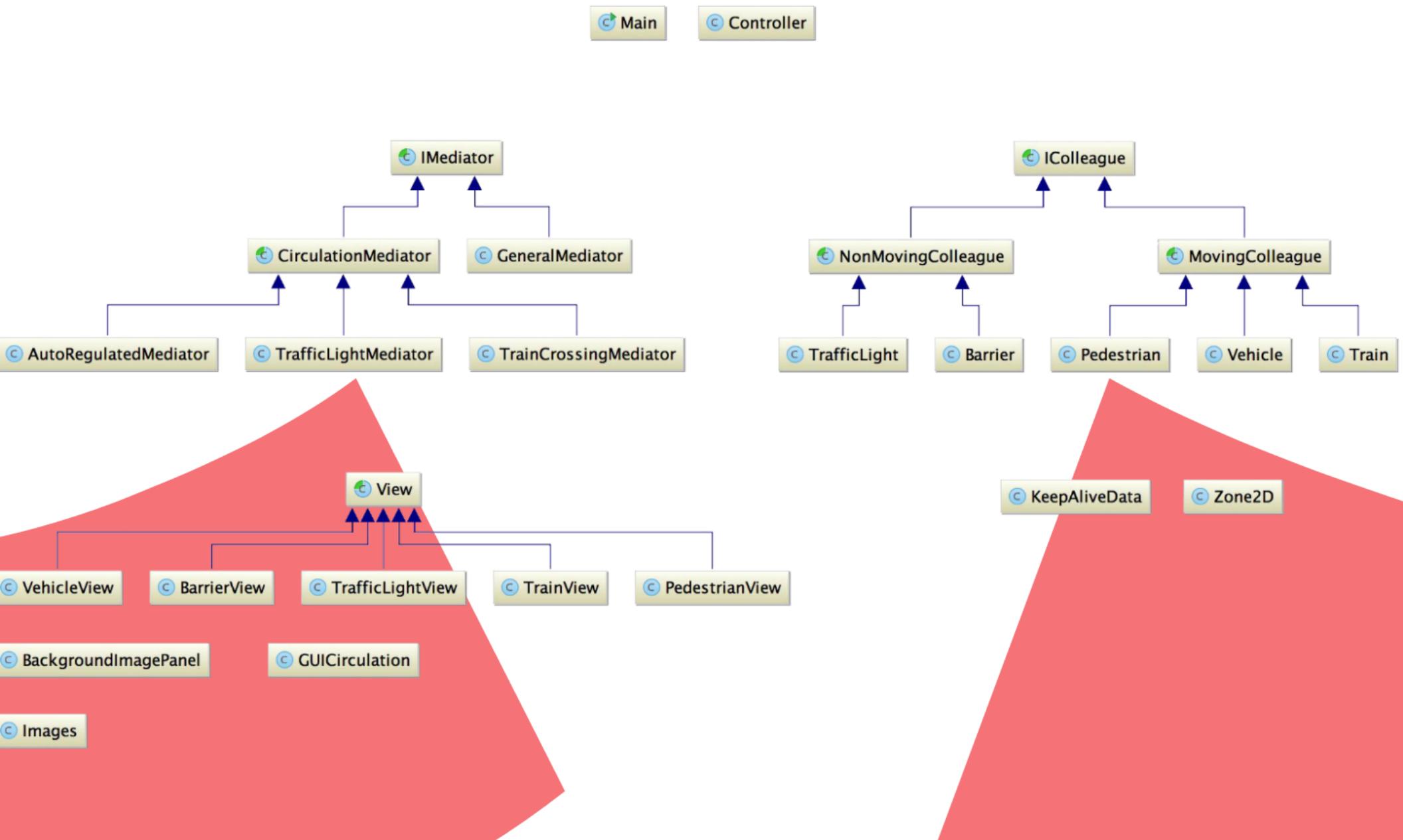


Design

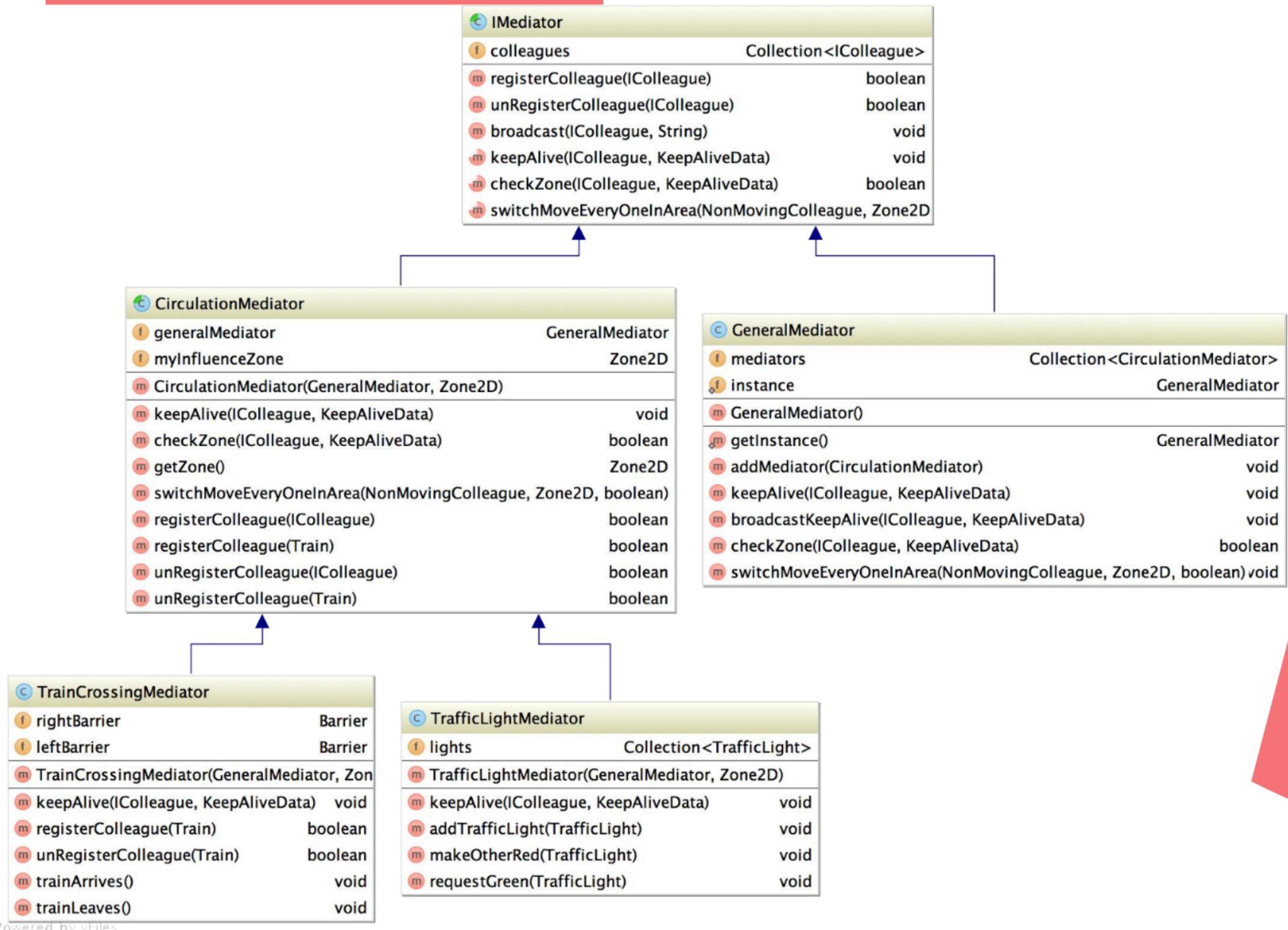
Solution Architecture



Class Diagram - Global

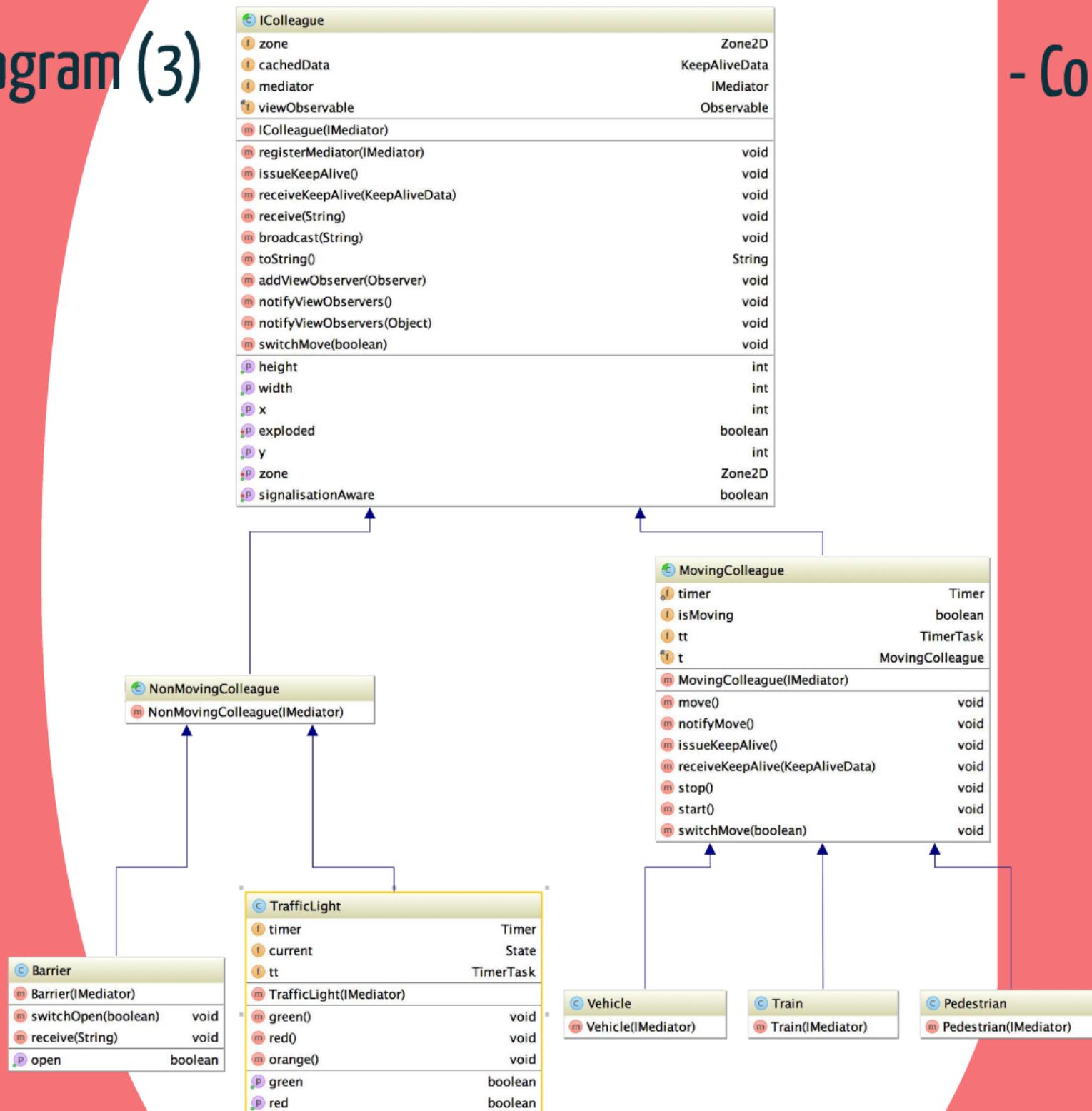


Class Diagram (2) - Mediators

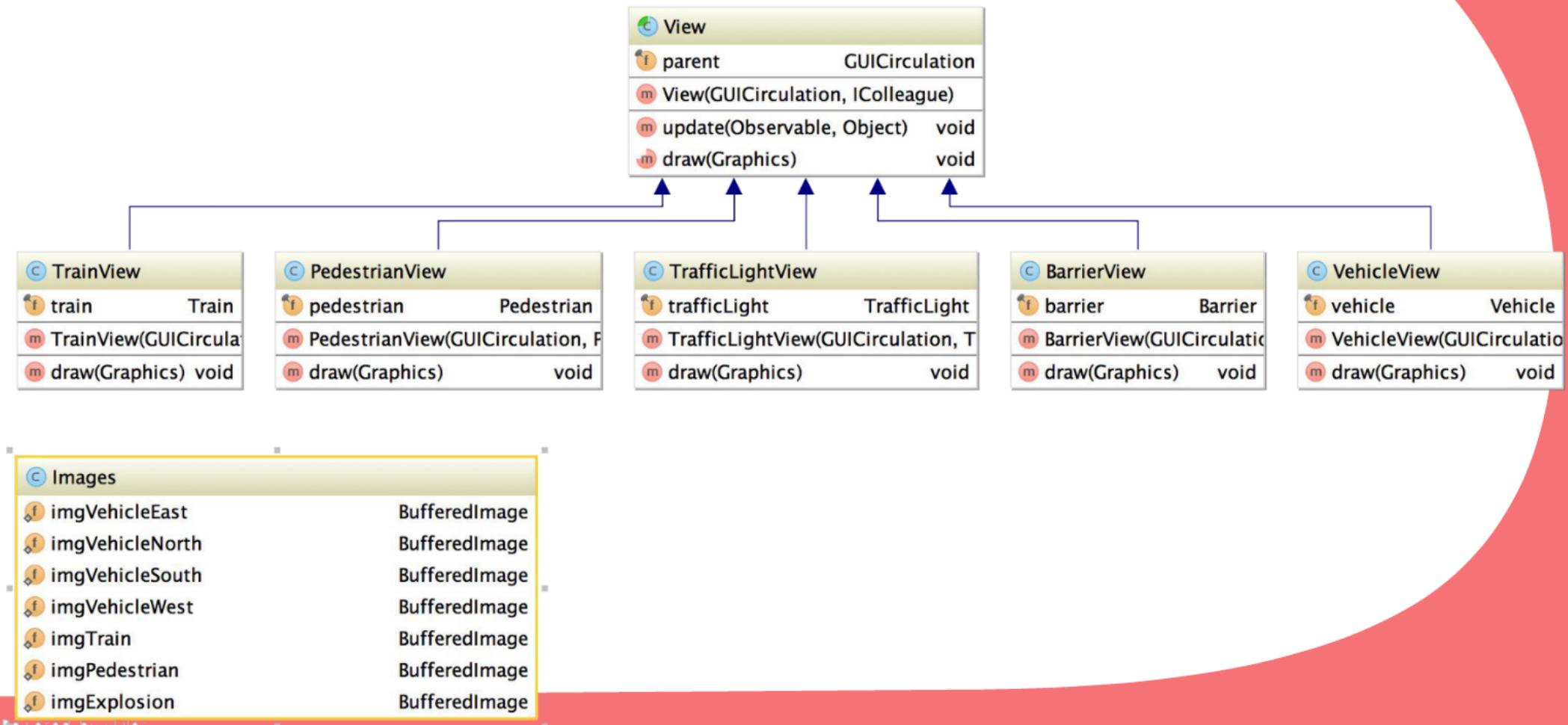


Class Diagram (3)

- Colleagues



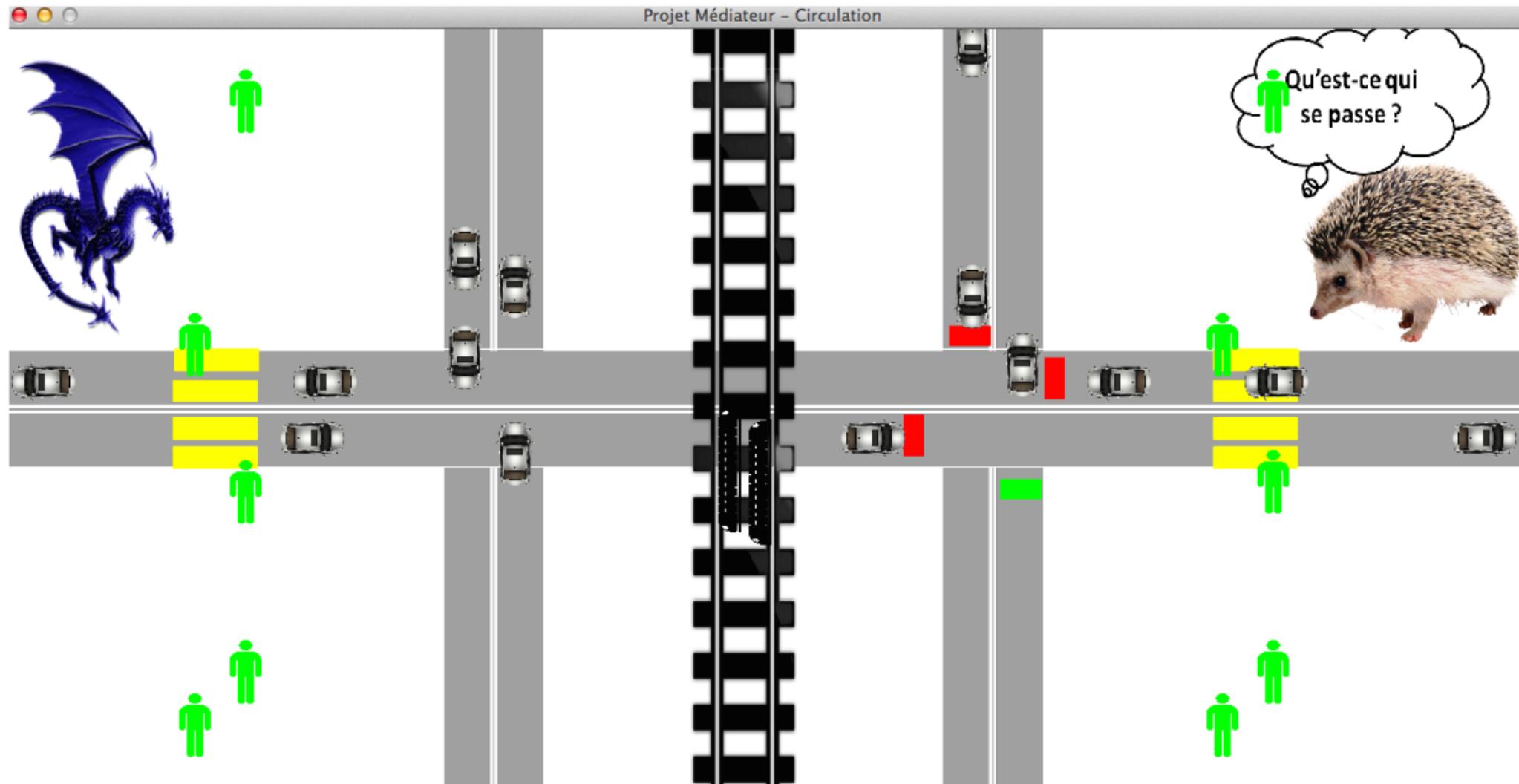
Class Diagram (4) - Views





Result

And, in practice



Conclusion

Overall conclusion

- The problem was a good example to implement the mediator design pattern.
- There is no "game" in our app (other than deactivating the reaction to messages and see vehicle explode... an extension would be to manually stop vehicle in this situation to avoid collisions).
- Good parallel of real life: colleagues are not aware of each other until they are close enough (typically, in front of you where you are going).

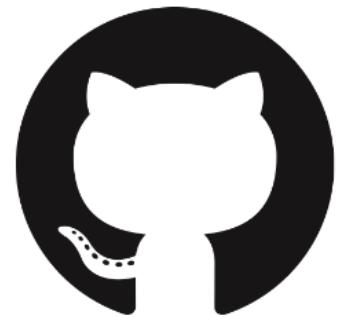
Technical conclusion

- The "keepalive" messages are working. We can see that vehicles are friendly: they stop when they have someone on their road, and they start again later, even if they have no clue of each other, neither the pedestrian and trains.
- The "traffic light" specific protocol is working. When someone arrives, a light request the green and tries to turn red all others. When greened, a light can start all vehicles, and stop them when turning red, even if they have no idea what is a vehicle and who they are.

Bugs, errors

- Unfortunately, the barrier protocol and the "right priority" protocol have not been implemented, but everything is ready to welcome them, and the pattern.
- Accidents happen a few times (it's a known bug, but it's similar as in real life: two people arrive at high speed and they don't have the time to see each other)
- Swing causes troubles again (blinking). We promised our moms we'll never swing again!

GitHub



Source code available at :

<https://github.com/ValentinMinder/CirculationMediator.git>

*All collected donations will be forwarded to the
"soutien des nuits blanches d'étudiants" association...*

