4+1 Architectuur model openpilot

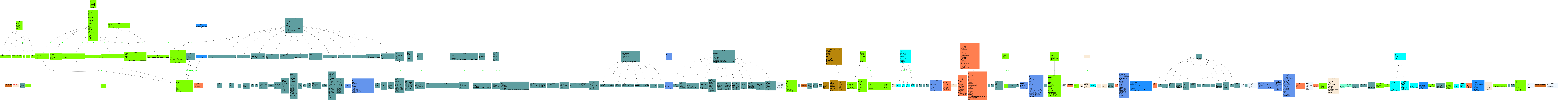
In dit verslag wordt naar de code van openpilot gekeken door middel van het 4+1 view model.

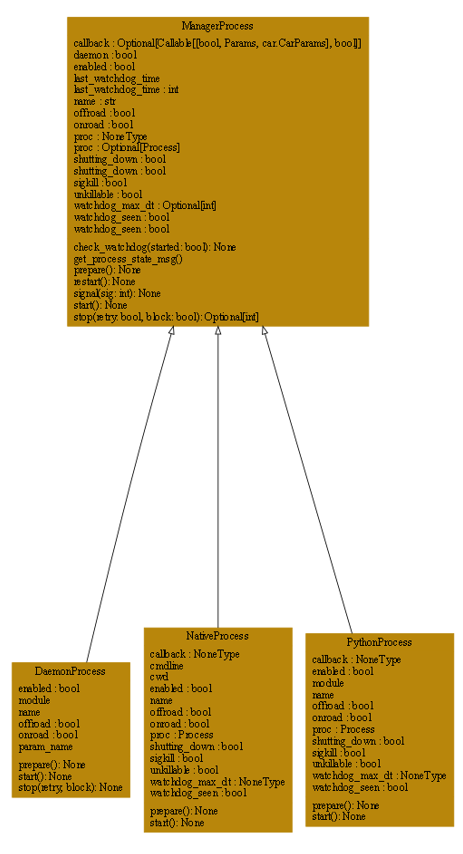
Hierbij wordt er in 5 verschillende views gekeken naar de code: logical, process, development, physical en scenarios.

# Logical view

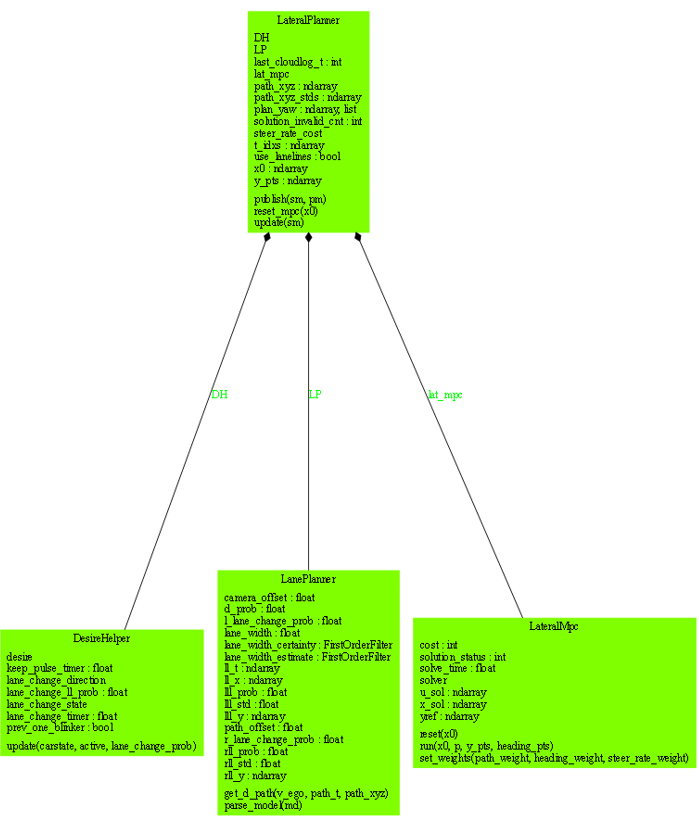
Dit hoofdstuk bevat het class diagram

Hieronder is het totaal overzicht te zien van de class diagram. Omdat deze extreem groot en onleesbaar is, zijn hieronder enkele ingezoomde elementen te zien welke het belangrijkste zijn voor de ontwikkeling van de virtuele rijcoach.



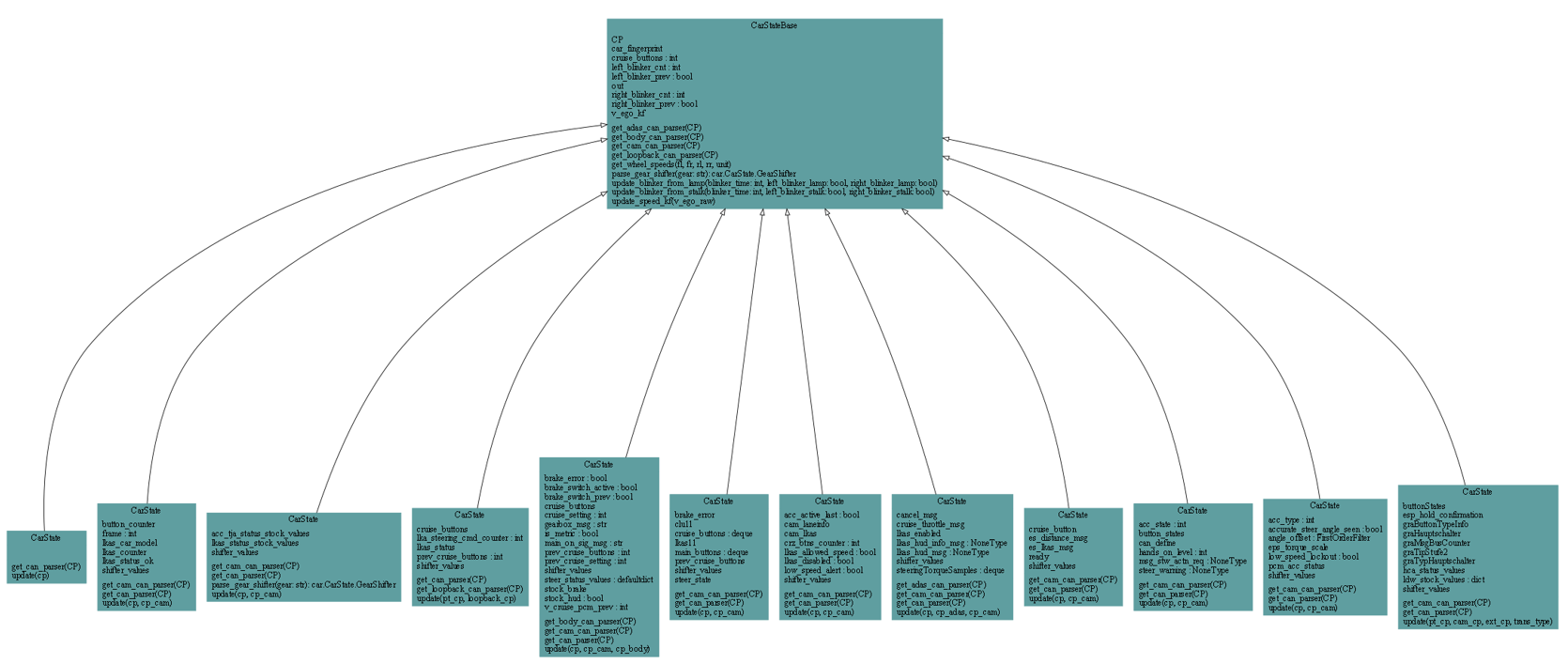


Afbeelding In dit diagram is te zien dat er 3 verschillende processen kunnen zijn welke gestart en gestopt kunnen worden door de ManagerProcess

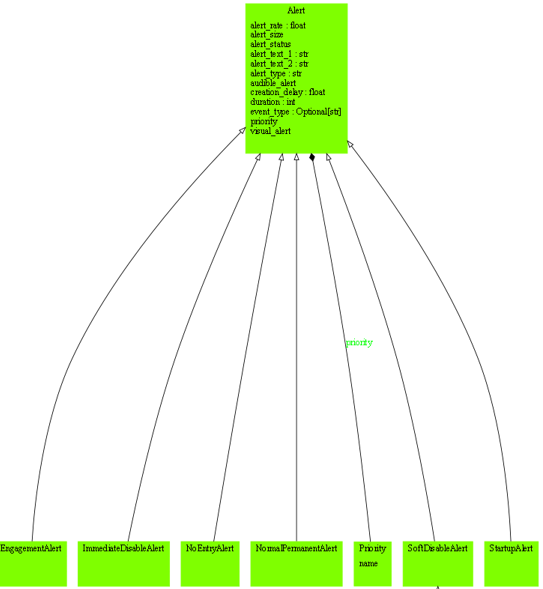


Afbeelding In dit diagram zijn de klassen te zien die plannen op welk baangedeelte de auto zicht moet (gaan) begeven

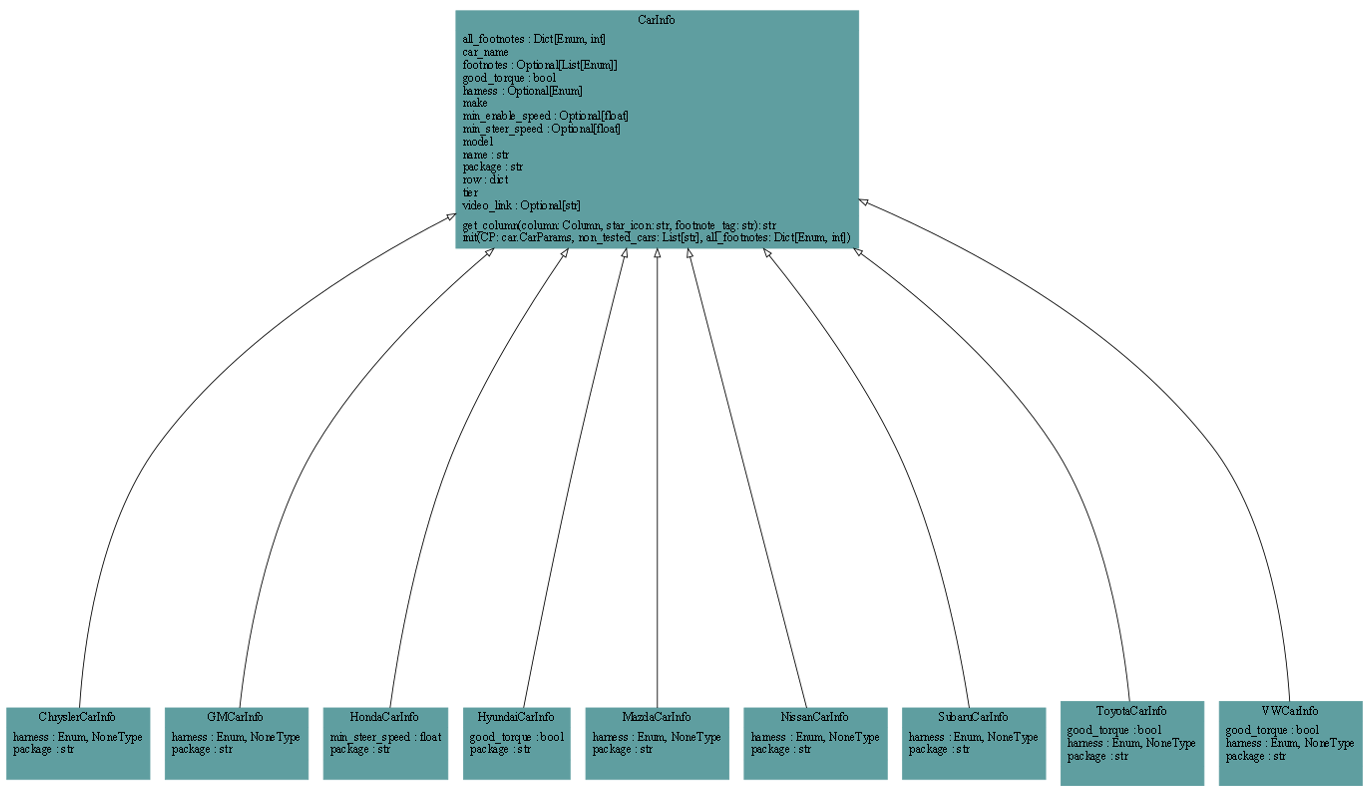
## State diagram



Afbeelding doordat Comma veel verschillende types auto's ondersteund moet er voor de meeste merken een andere CarState aangemaakt worden omdat dit vaak verschild per automerk



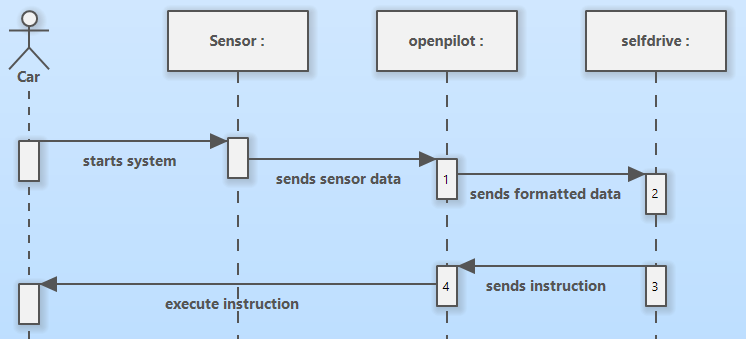
Afbeelding In dit diagram is te zien welke klassen verbonden zijn aan de Alert klasse, welke gebruikt kan worden voor alerts op de comma



Afbeelding Hier zijn de verschillende merken te zien die onderstuend worden door de comma

# Process view

Sequence diagram waarin te zien is hoe data van een sensor omgezet wordt in een instructie voor de auto.



1: De sensor stuurt data naar openpilot waar deze data wordt omgezet in bruikbare data voor het selfdrive gedeelte van de code.

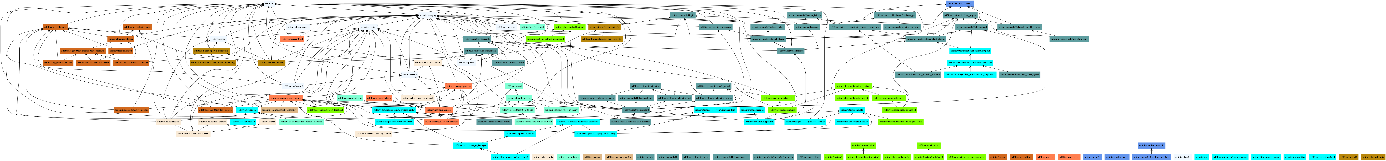
2: In selfdrive wordt de data ontvangen en geanalyseerd door middel de modellen.

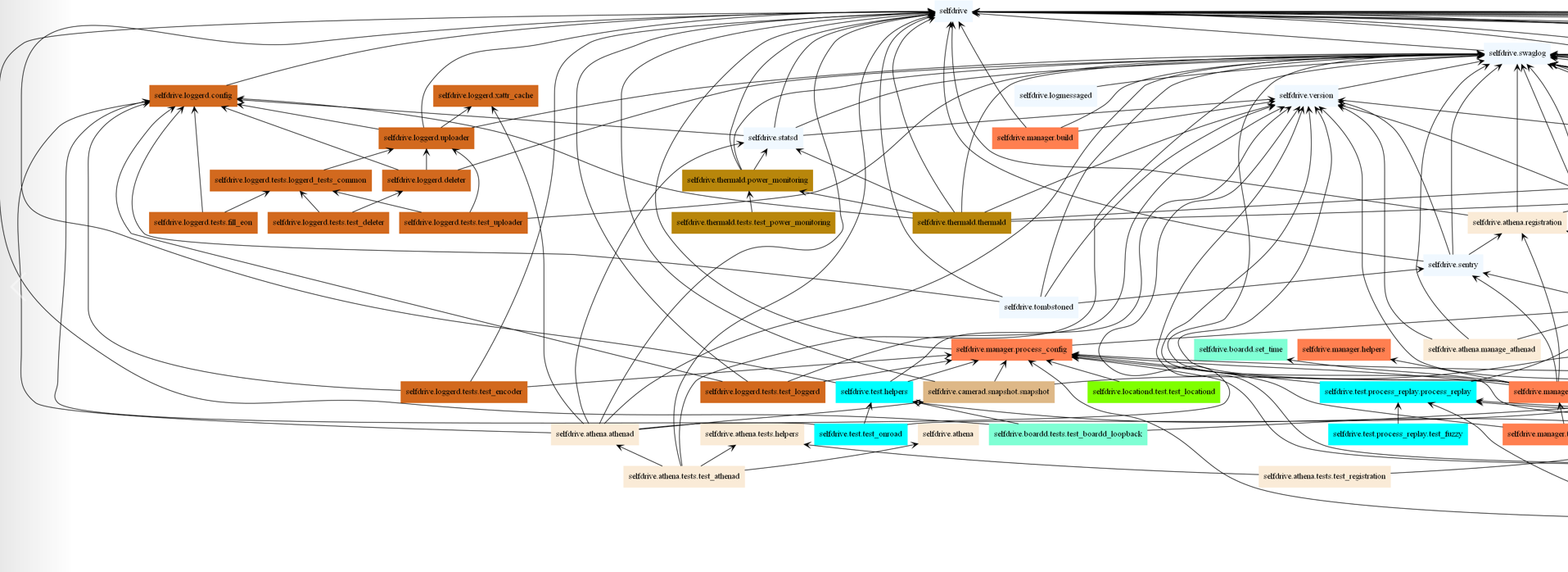
3: De modellen hebben de data geanalyseerd en een instructie bepaald die naar openpilot wordt gestuurd

4: Openpilot vertaald de instructie naar een instructie die geformatteerd is voor de auto en de auto wordt hierdoor aangestuurd.

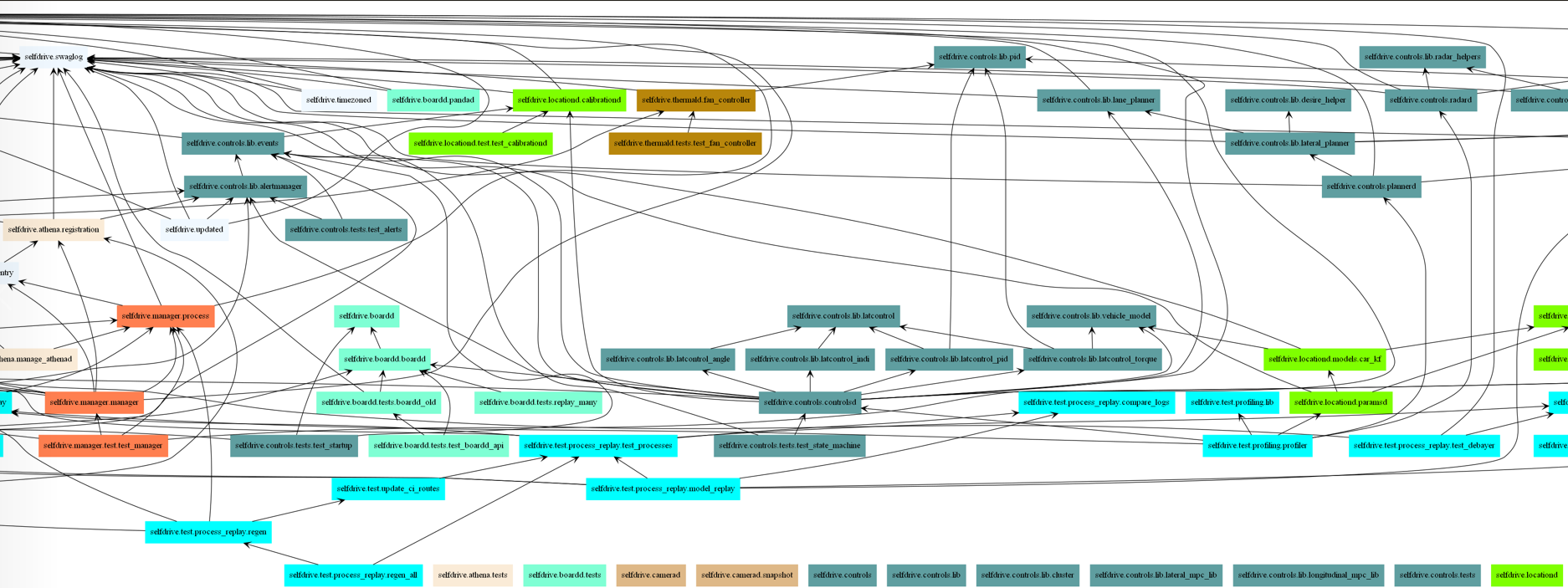
# Development view

Net als het klassendiagram in de logical view is ook het package diagram te groot om goed op een afbeelding weer te geven. Om toch een beetje duidelijkheid te geven over de packages in de map selfdrive is de afbeelding hieronder opgedeeld in 3 delen, van links naar rechts. Ook zijn de packageg Car, Debug en Hardware weggelaten om een overzichtelijker beeld te geven.

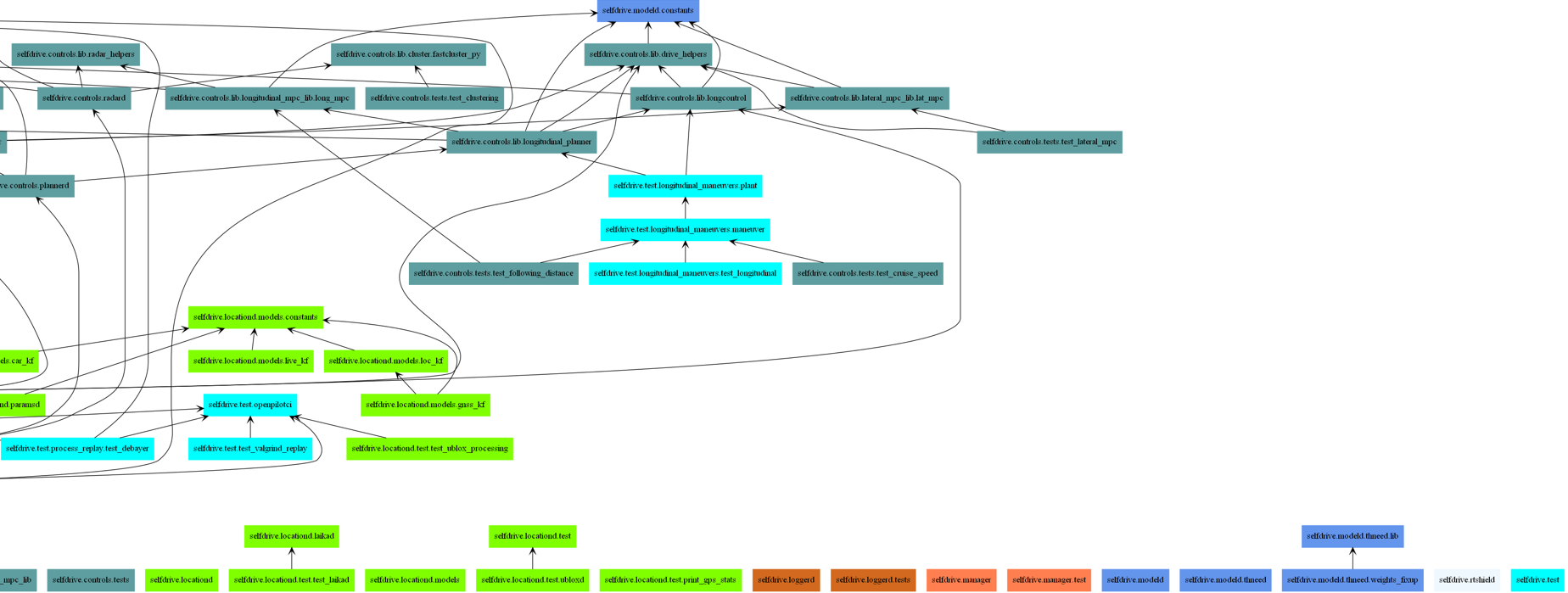




Afbeelding linker deel packagediagram

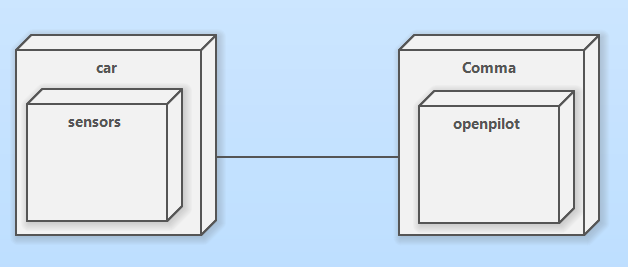


Afbeelding middelste deel packagediagram



Afbeelding rechter deel packagediagram

# Physical view



# Scenarios

Een usecase diagram met als actor de bestuurder van de auto

