

# UAVs autonomously navigating dynamic environments

Borcherd van Brakell & Vic Segers



**Bedrijfsvoorstelling**

**1**

**Projectvoorstelling**

**2**

**Implementatie  
architecture**

**3**

**Demo architecture**

**4**

# INHOUDSOPGAVE



**5 Onderzoeksvragen**

**6 Implementatie  
showcase**

**7 Demo showcase**

**8 Reflectie**

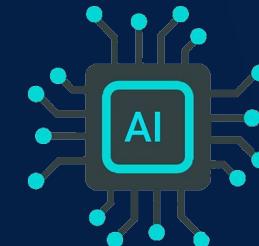
# 1

## Bedrijfsvoorstelling

---



# Bedrijfsvoorstelling



Robotics



VR/AR



A complex network graph is displayed against a dark blue background. The graph consists of numerous cyan-colored nodes (dots) connected by a web of white lines representing edges. The nodes are distributed across the frame, with larger clusters on the left and right sides and more isolated nodes towards the center and bottom. Some nodes have multiple edges connecting them to others, while others are part of small local clusters.

2

# Projectvoorstelling

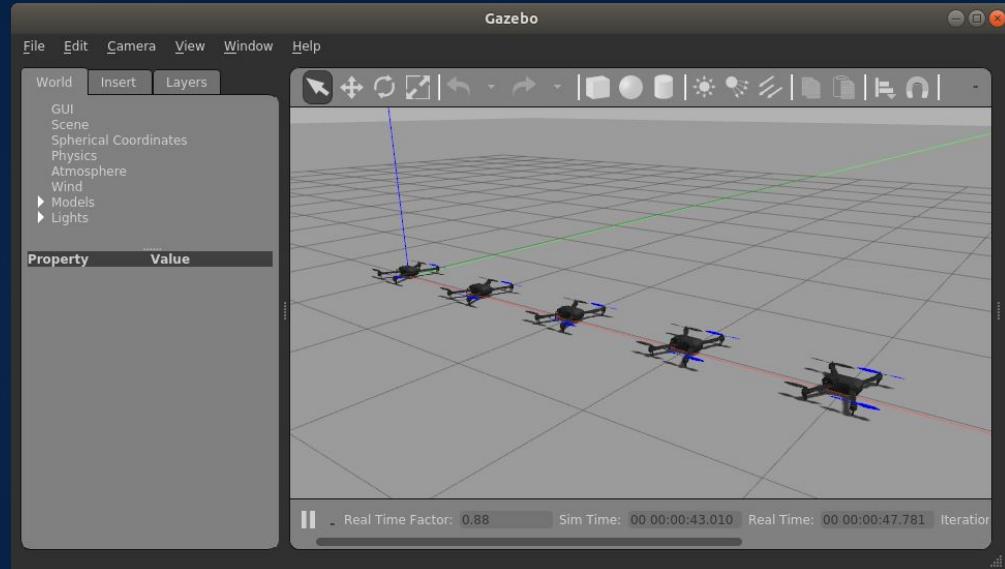
---

# Projectvoorstelling

## ARCHITECTURE

### Vereisten:

- Git
- Docker
- Ubuntu
- Mesa-utils
- 6.5 GB vrije ruimte

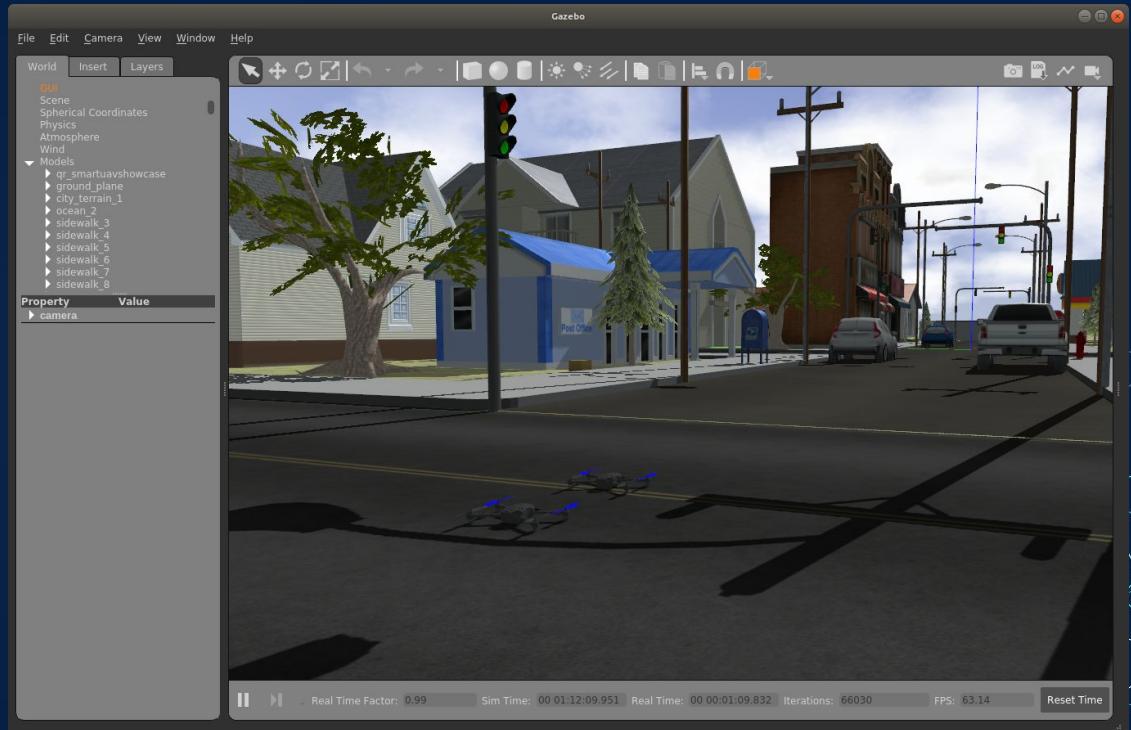


# Projectvoorstelling

## Showcase

### Vereisten:

- Architecture
- 500 MB vrije ruimte

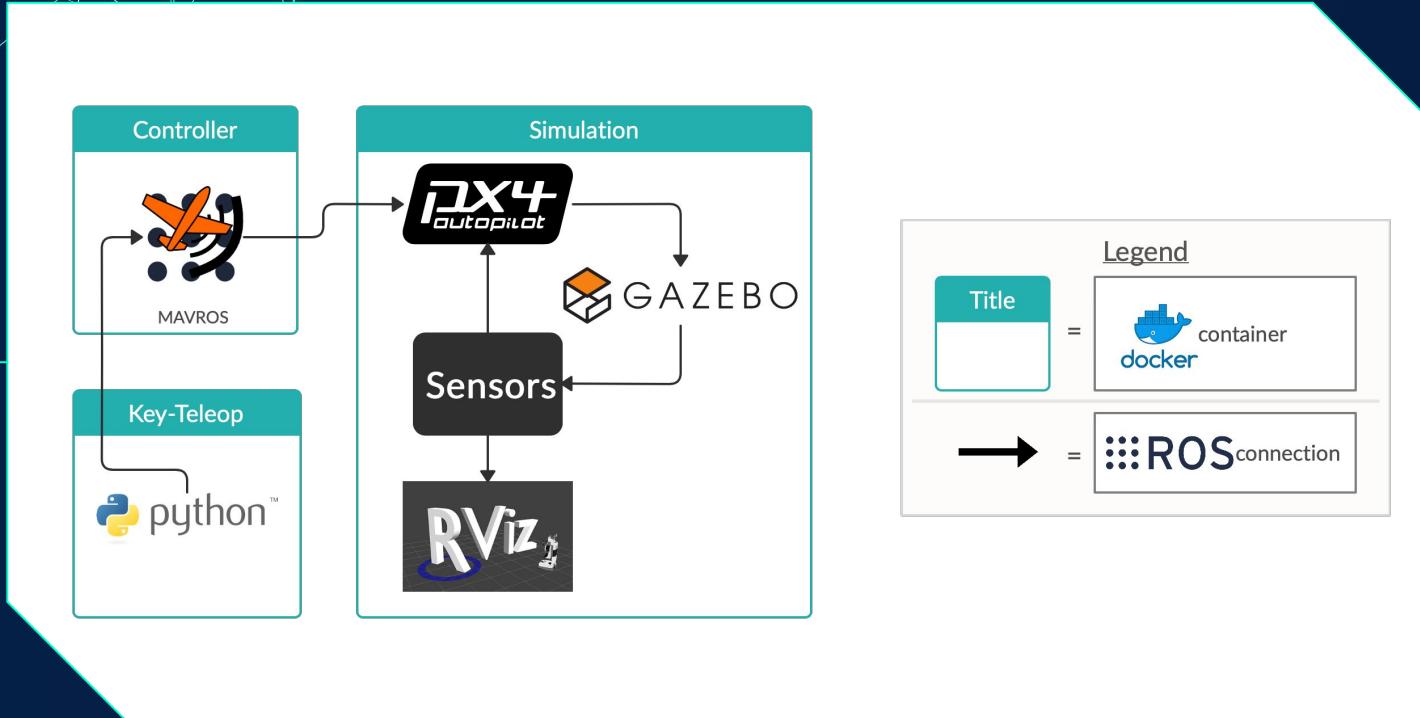


# 3

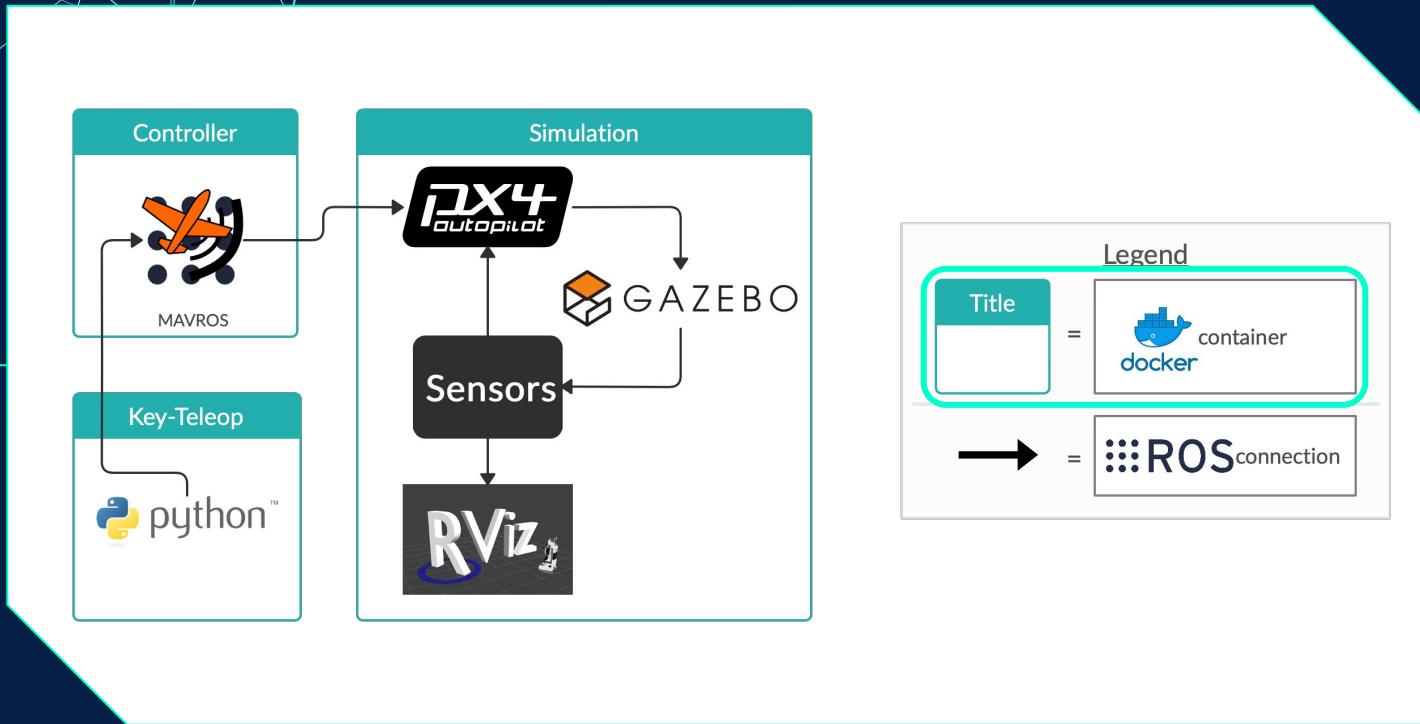
## Implementatie architecture



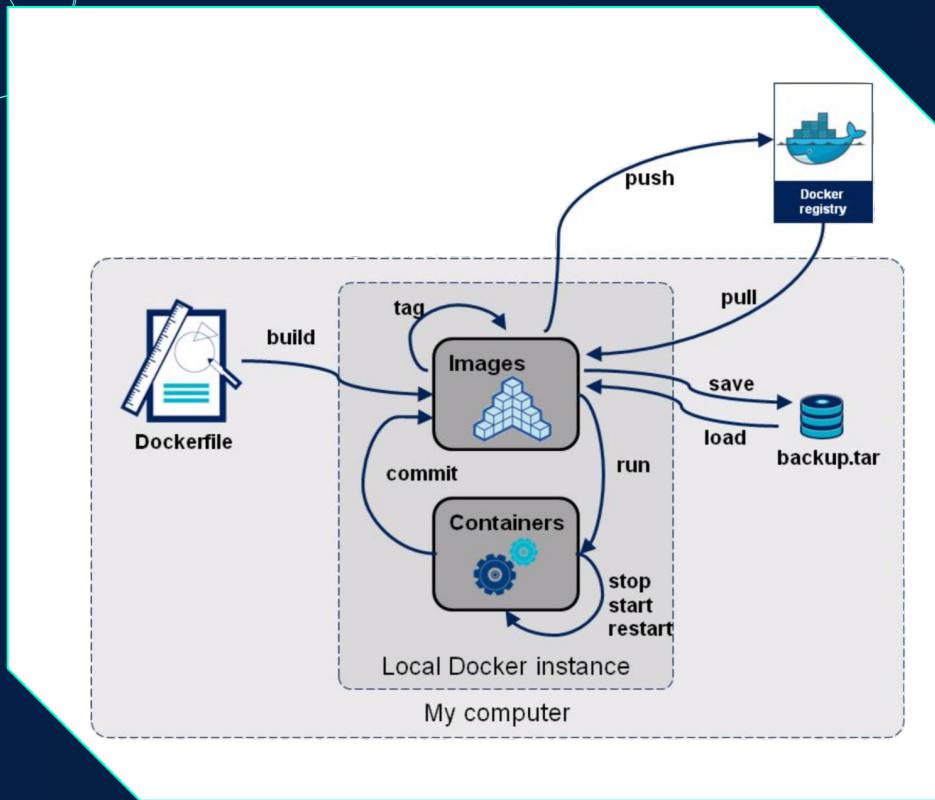
# Implementatie architecture



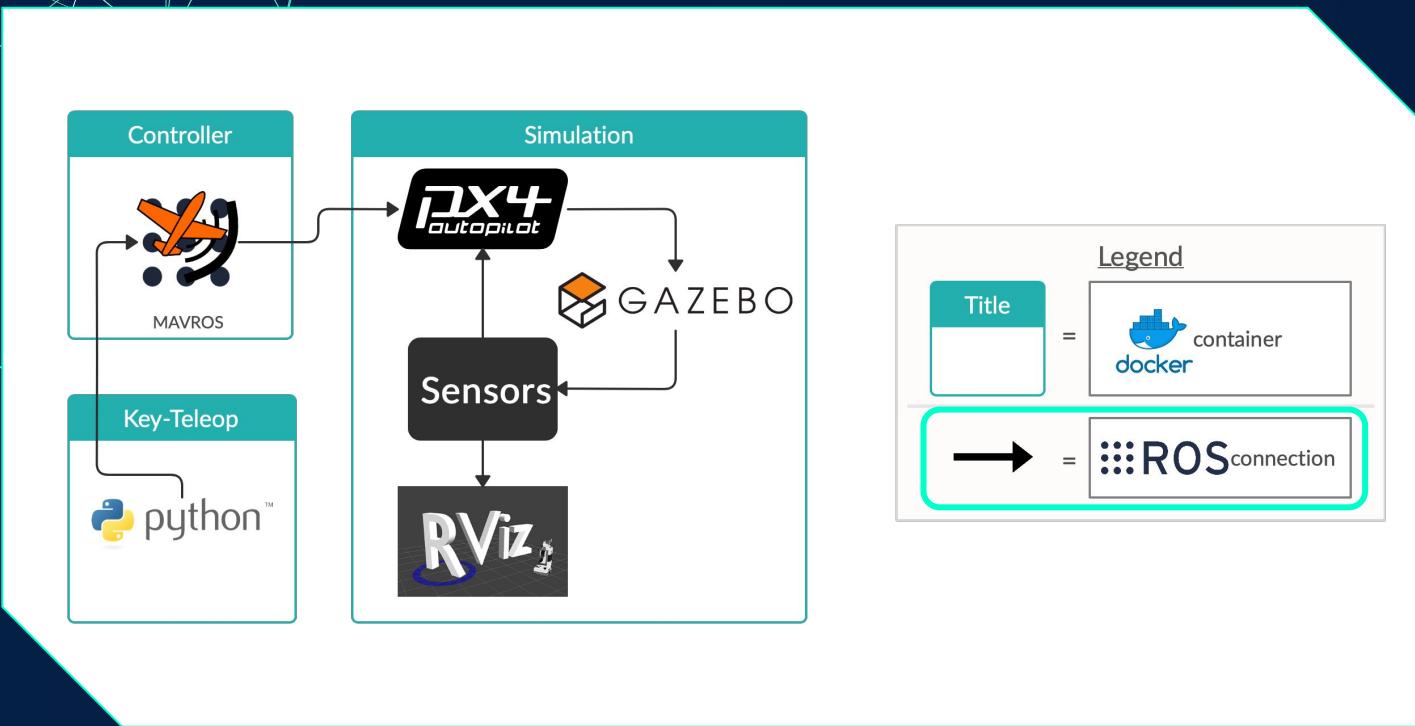
# Implementatie architecture



# Docker



# Implementatie architecture

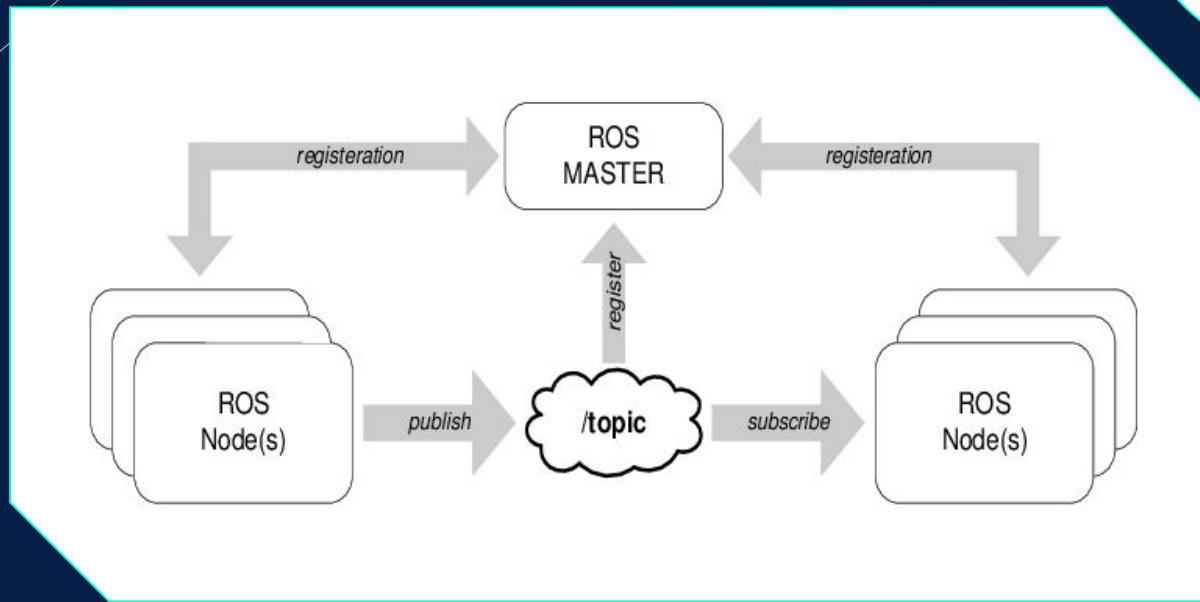




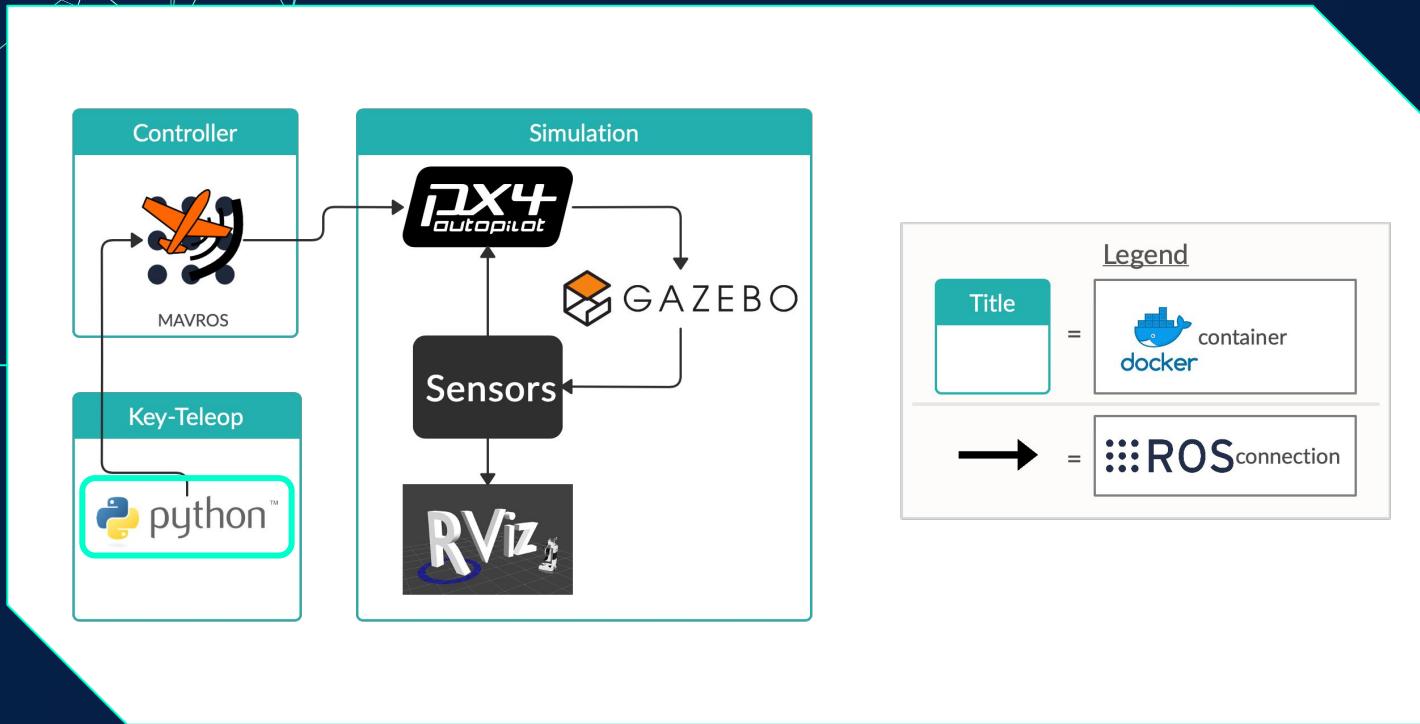
# ROS



# ROS



# Implementatie architecture

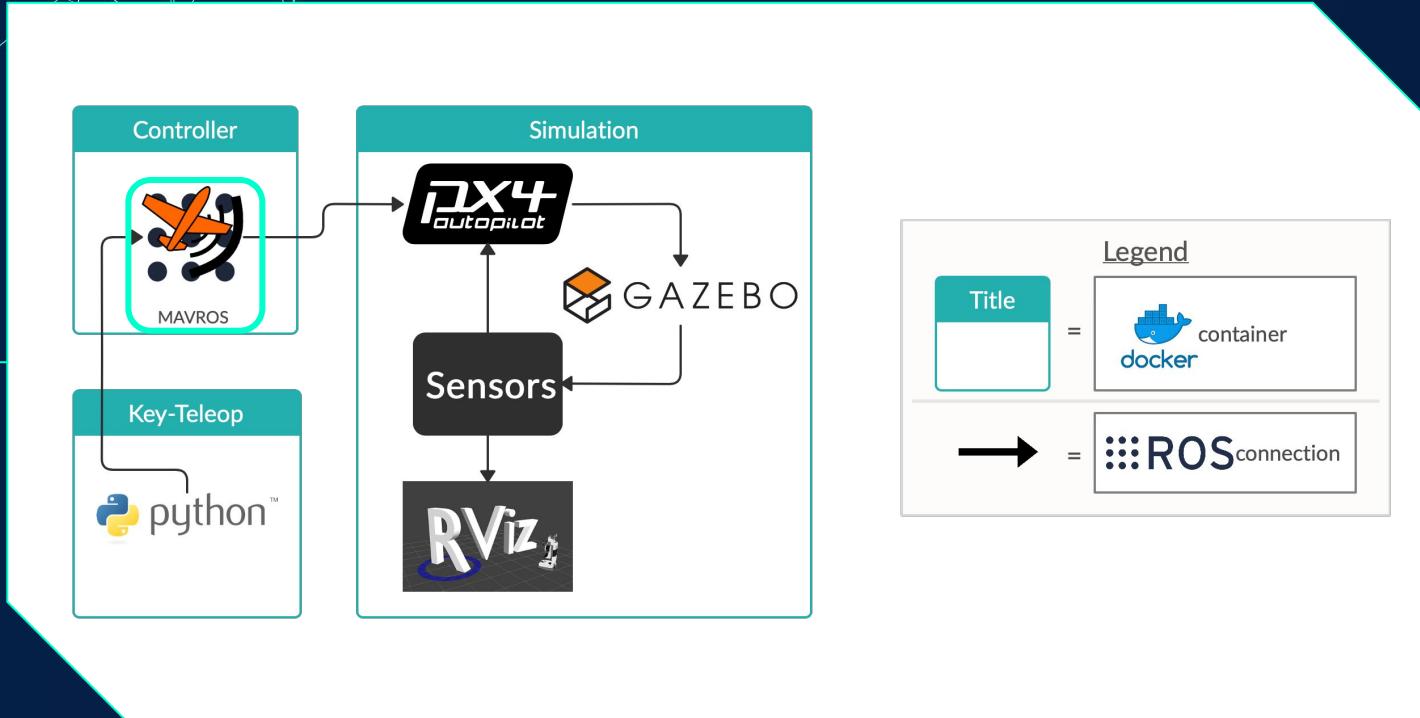


# Python



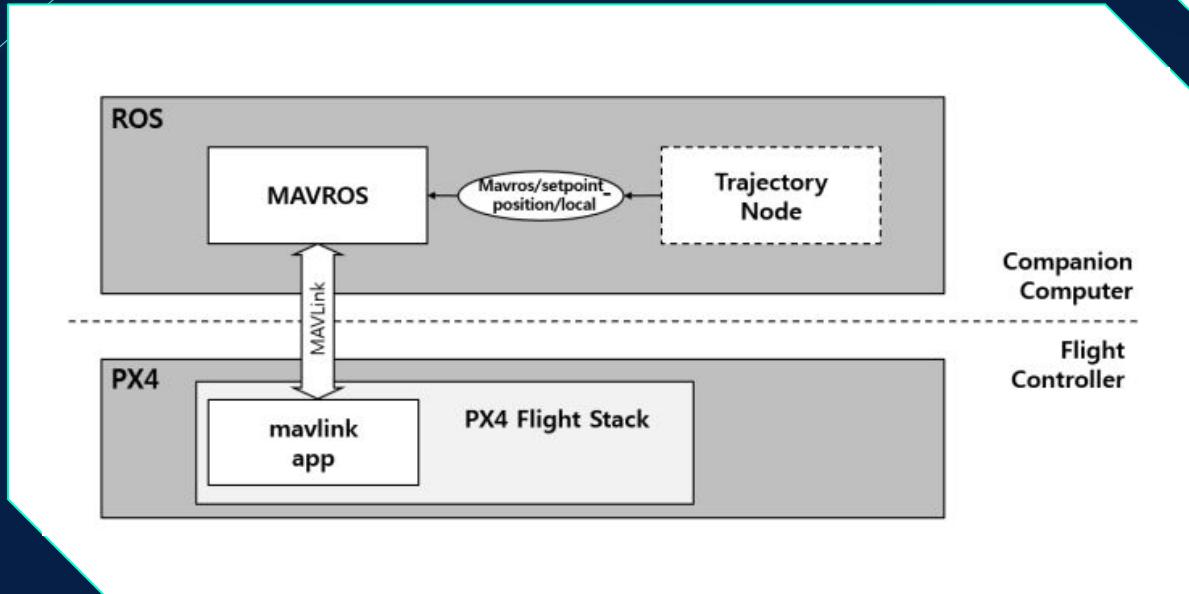
```
33     self.logduplicates = True
34     self.debug = debug
35     self.logger = logging.getLogger(__name__)
36     if path:
37         self.file = open(os.path.join(path, 'fingerprint.log'), 'a')
38         self.file.seek(0)
39         self.fingerprints.update(self._read_file())
40
41     @classmethod
42     def from_settings(cls, settings):
43         debug = settings.getbool('SUPERVISOR_DEBUG')
44         return cls(job_dir(settings), debug)
45
46     def request_seen(self, request):
47         fp = self.request_fingerprint(request)
48         if fp in self.fingerprints:
49             return True
50         self.fingerprints.add(fp)
51         if self.file:
52             self.file.write(fp + os.linesep)
53
54     def request_fingerprint(self, request):
55         # ...
```

# Implementatie architecture

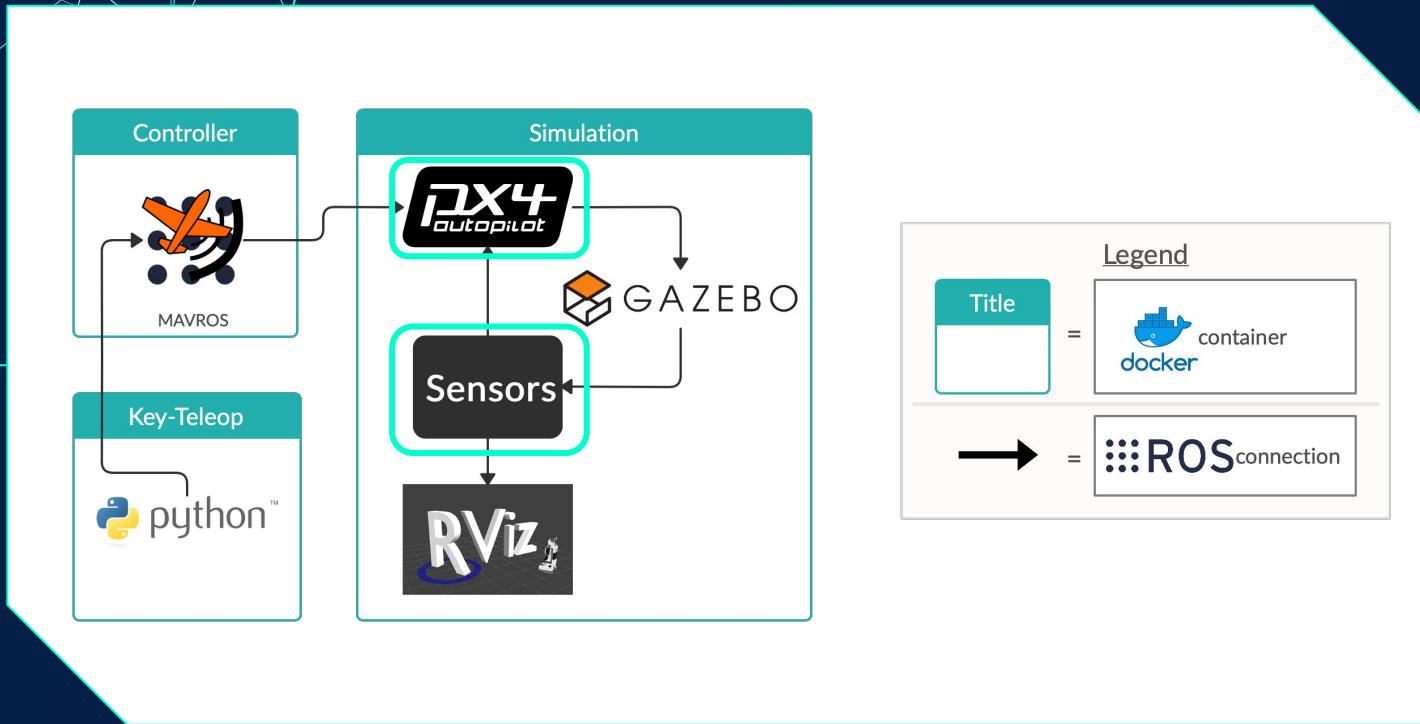




# MAVROS



# Implementatie architecture

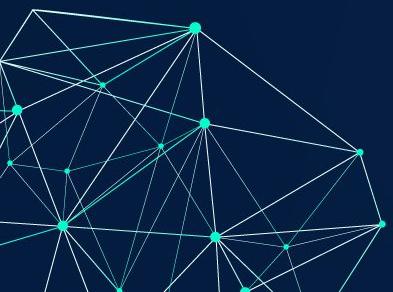




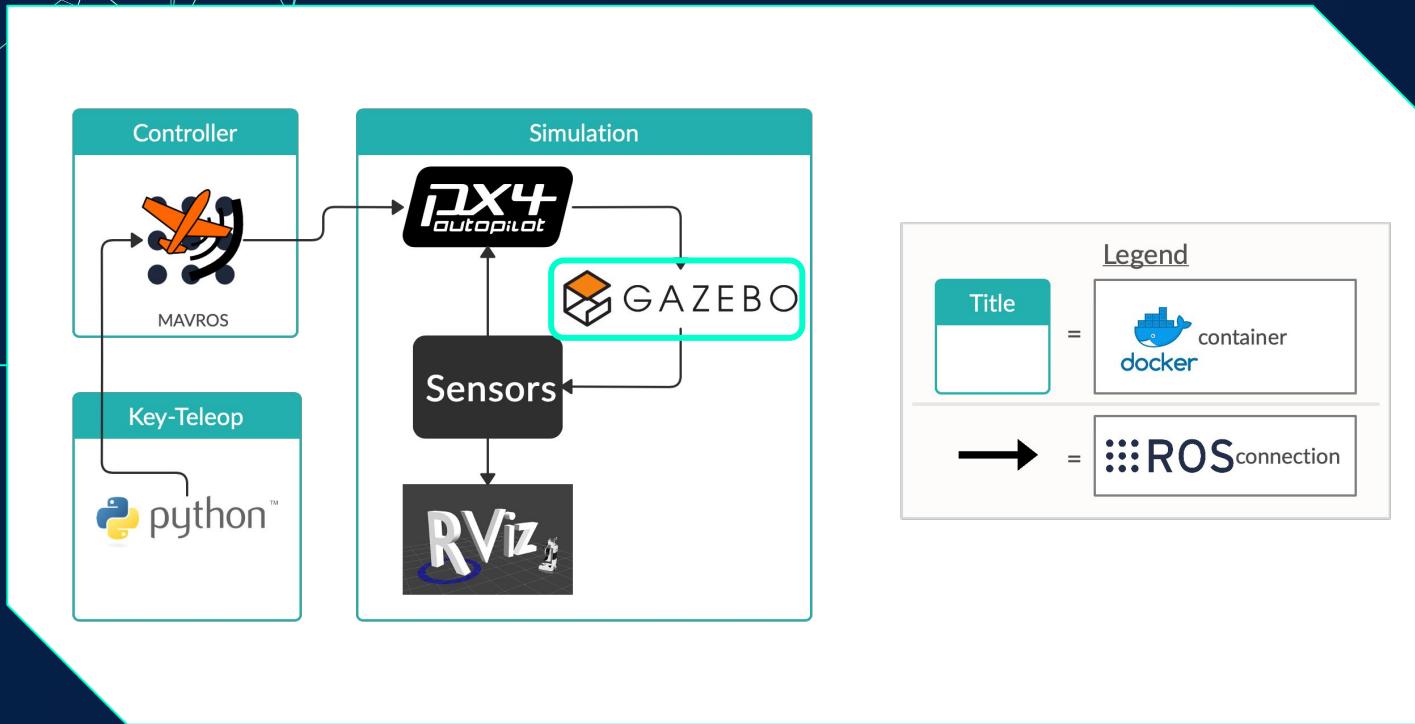
# PX4 autopilot



# Sensoren



# Implementatie architecture

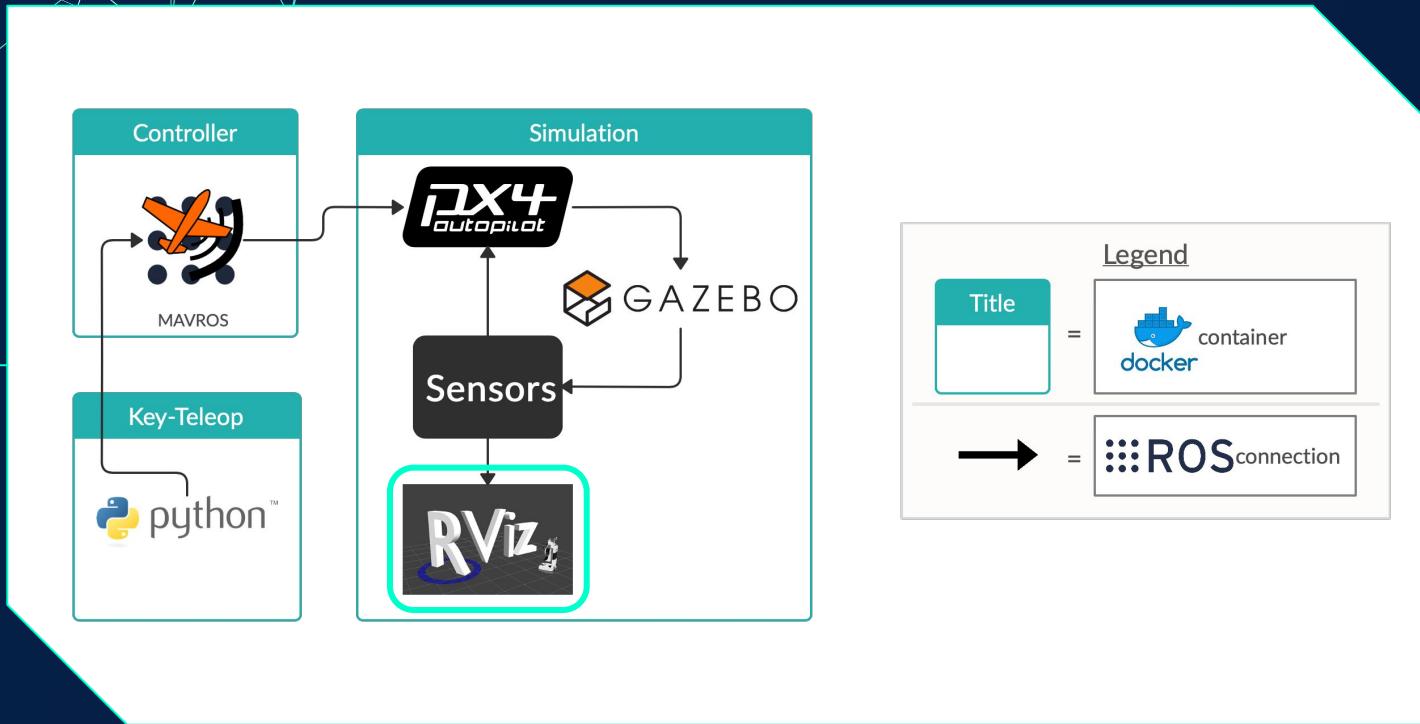




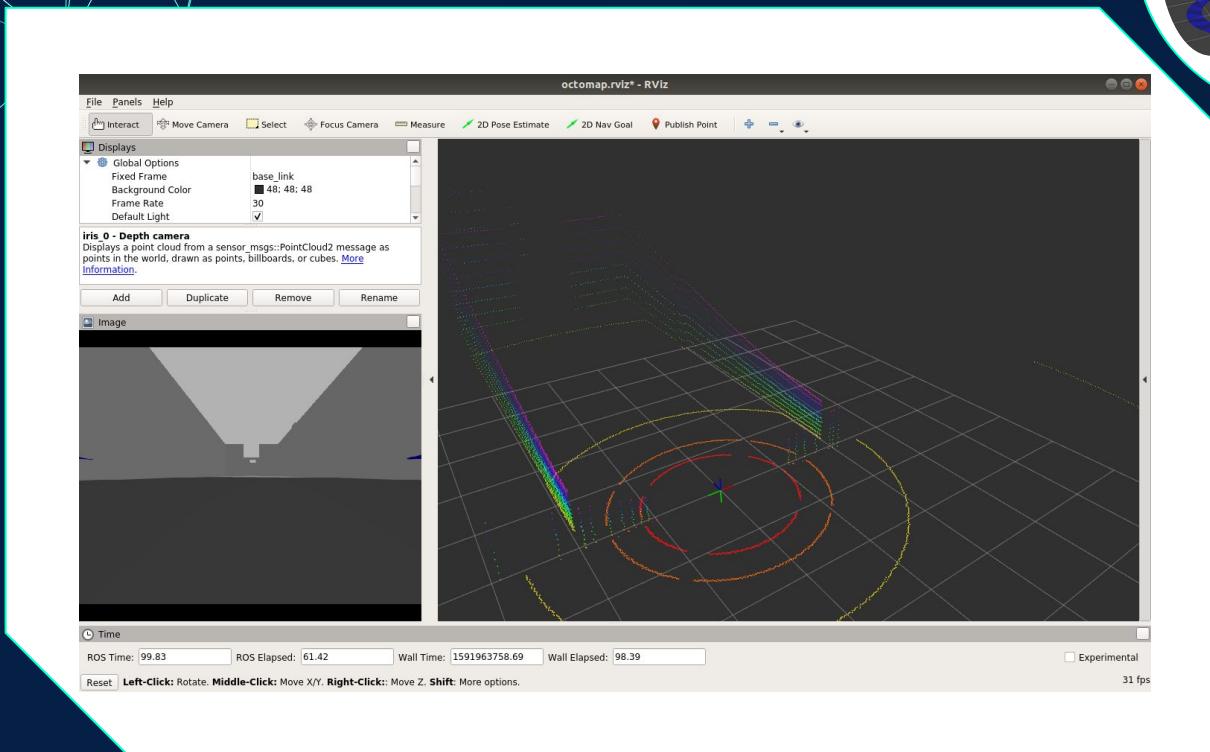
# Gazebo



# Implementatie architecture



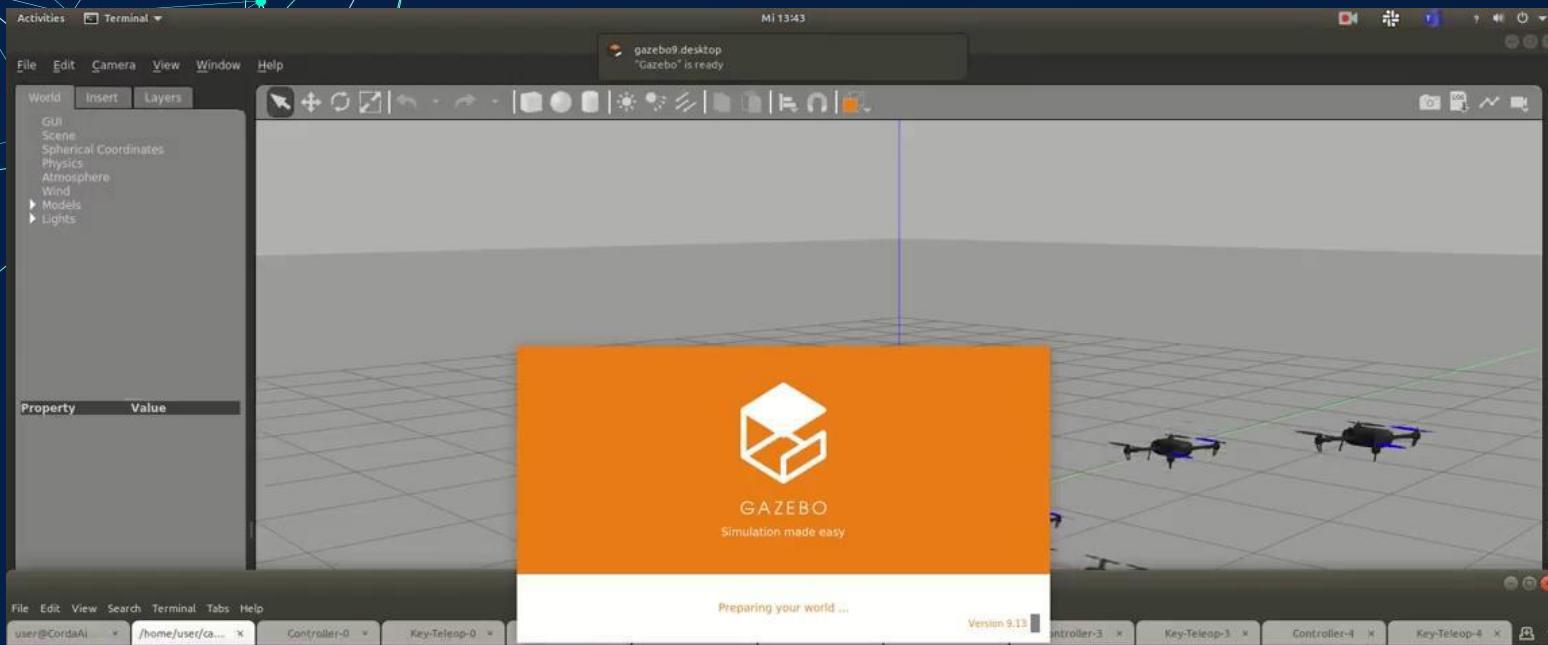
# RViz



# 4

## Demo architecture

# Demo architecture



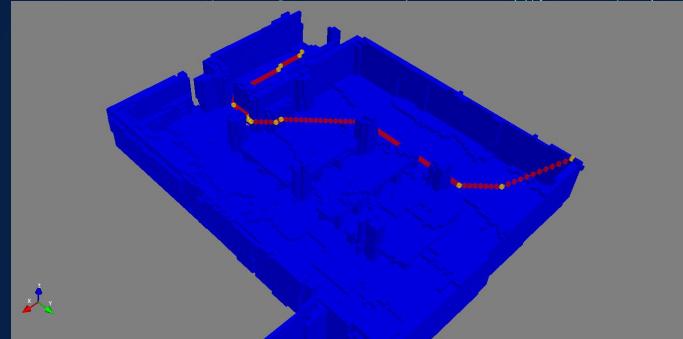
# 5

## Onderzoeksvragen

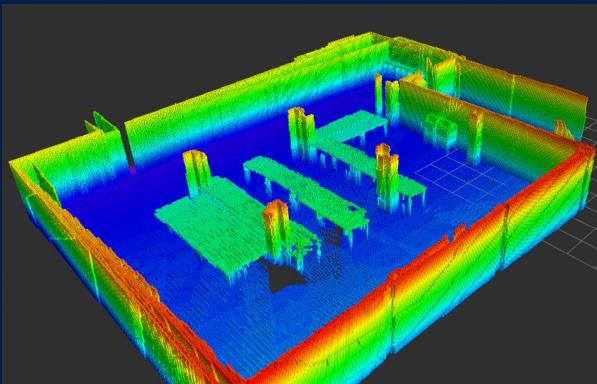
# Hoe kan een UAV autonoom navigeren in een dynamische omgeving?



Dynamische omgeving

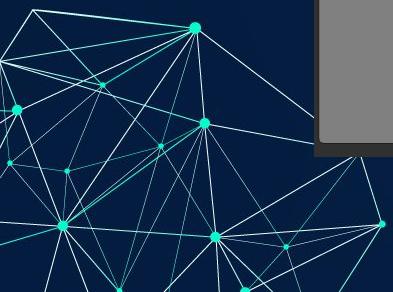
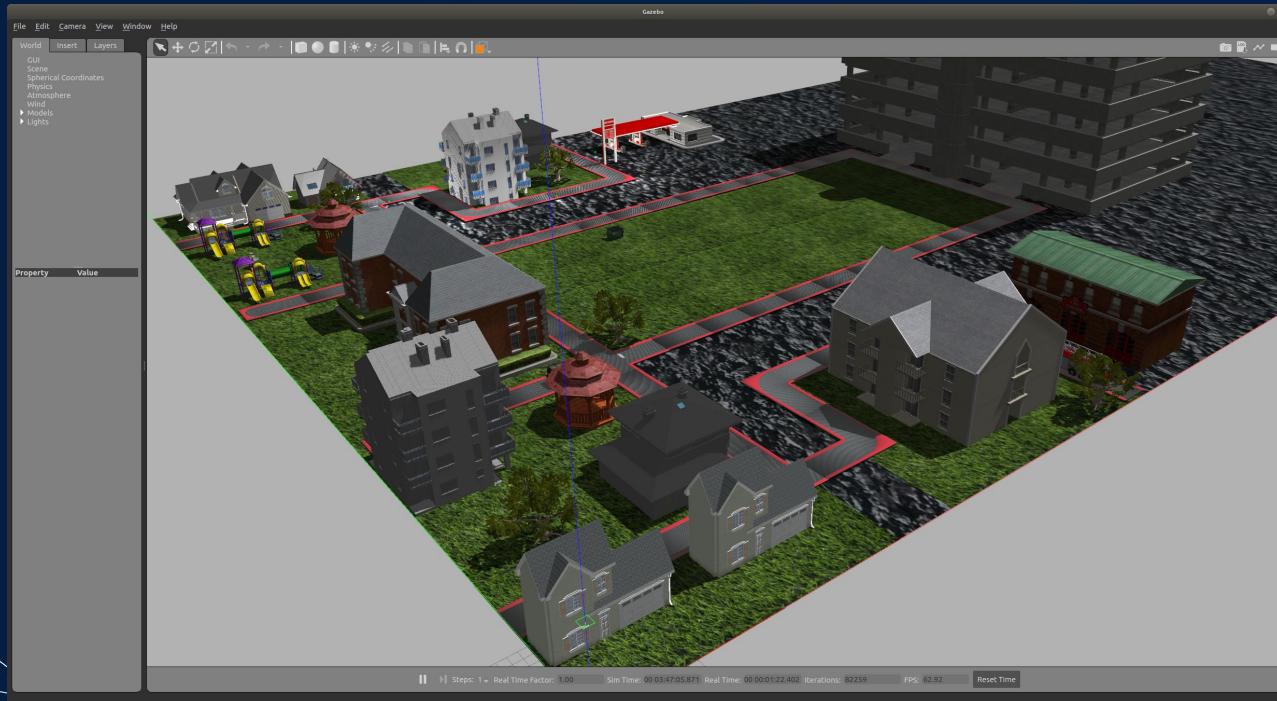


Pad planning en uitvoering

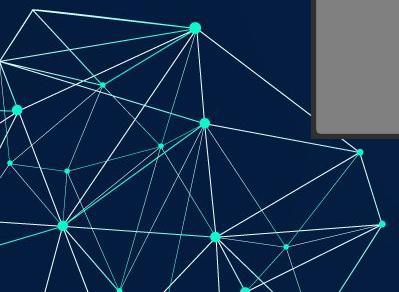


Oriëntatie

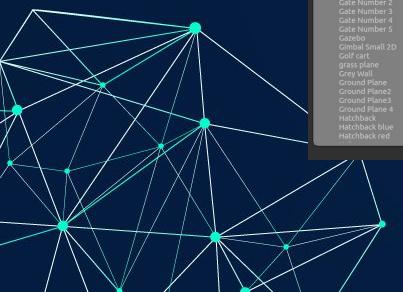
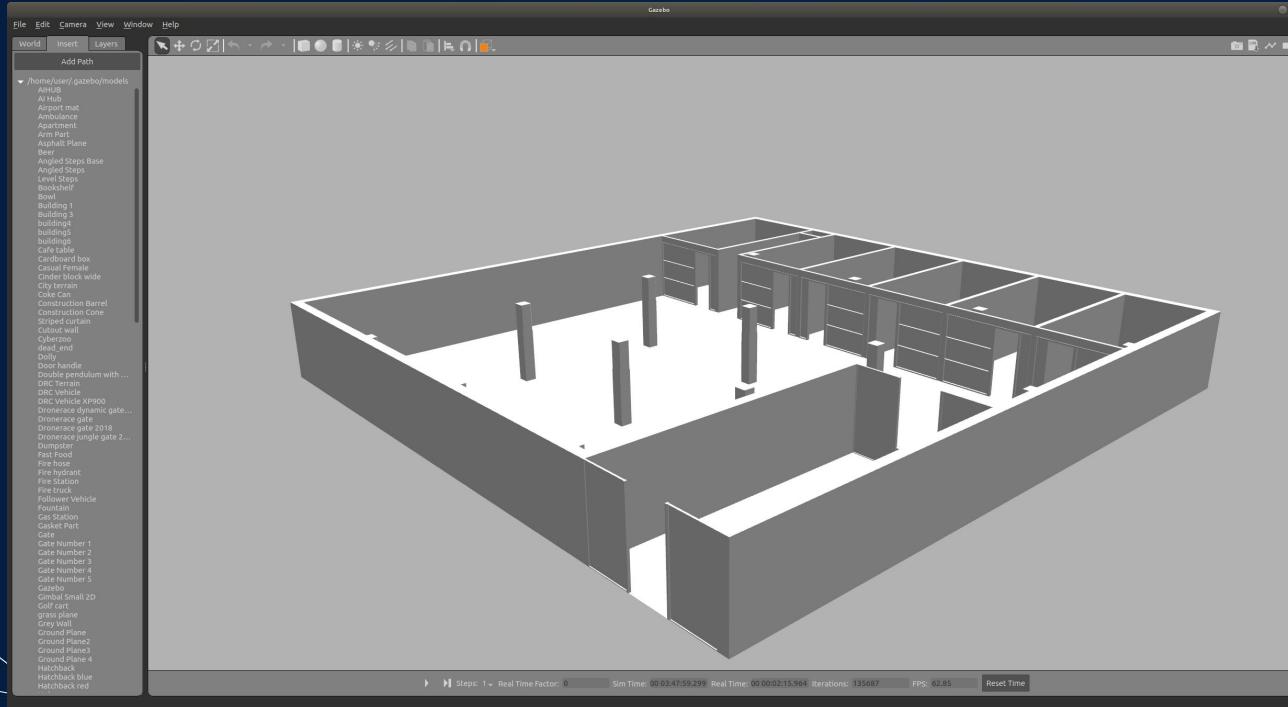
# Gazebo werelden



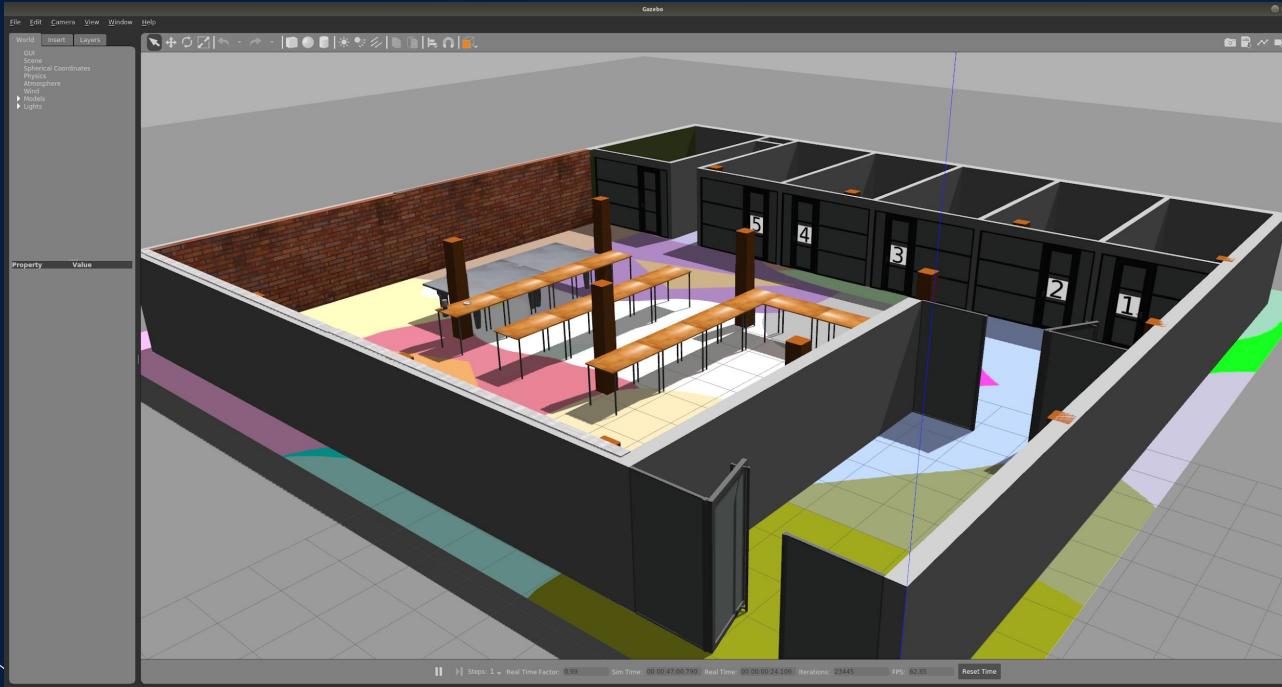
# Gazebo werelden



# Gazebo werelden



# Gazebo werelden



# Dynamische omgeving

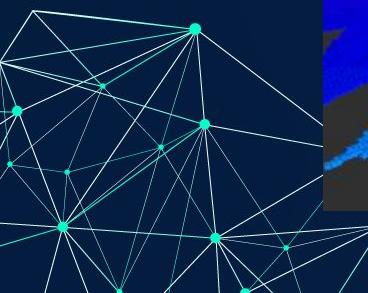
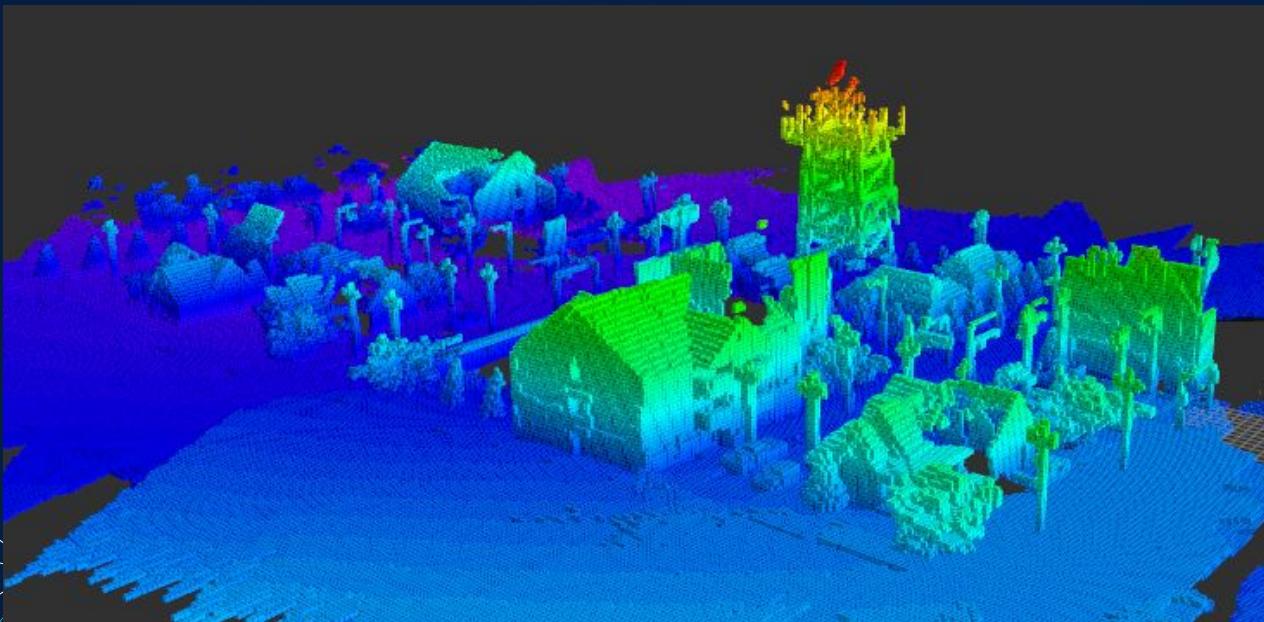


# Oriëntatie

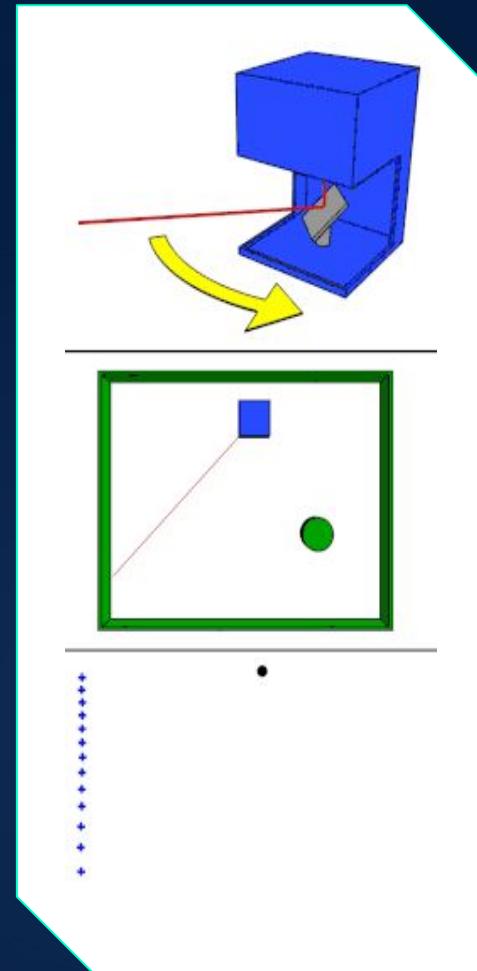
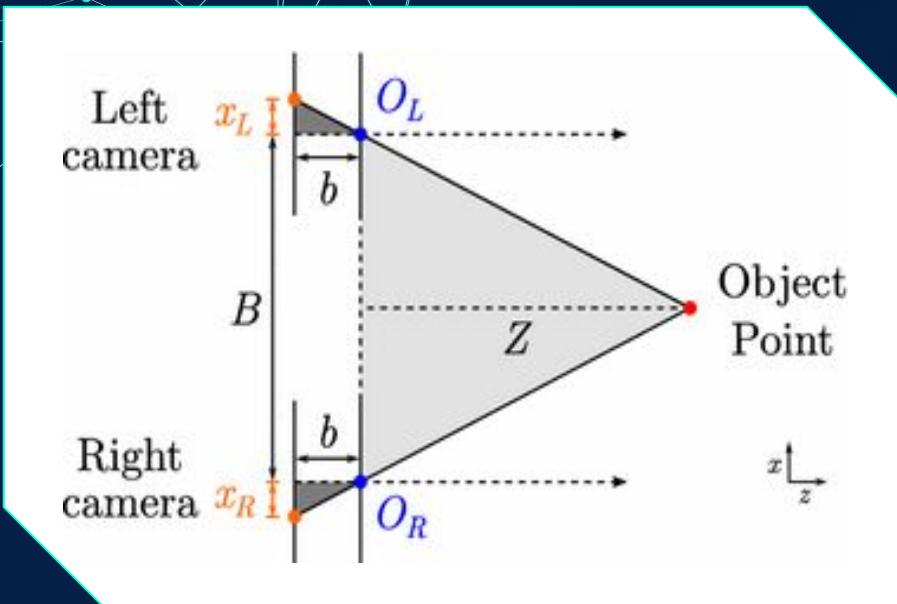
**Observatie**

**Mappen en lokaliseren**

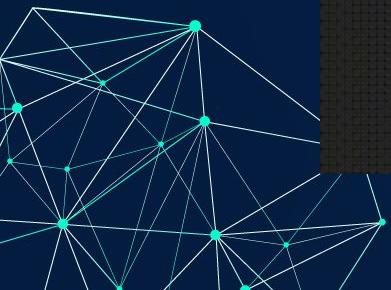
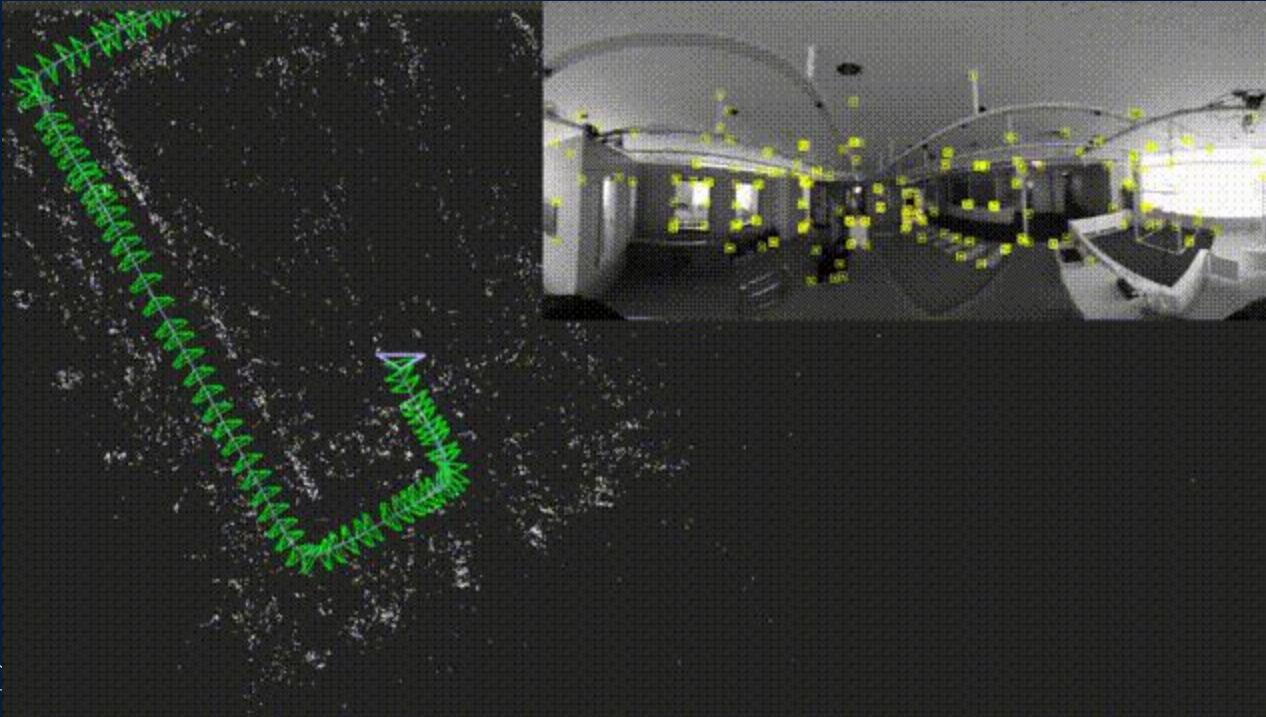
**SLAM algoritmen**



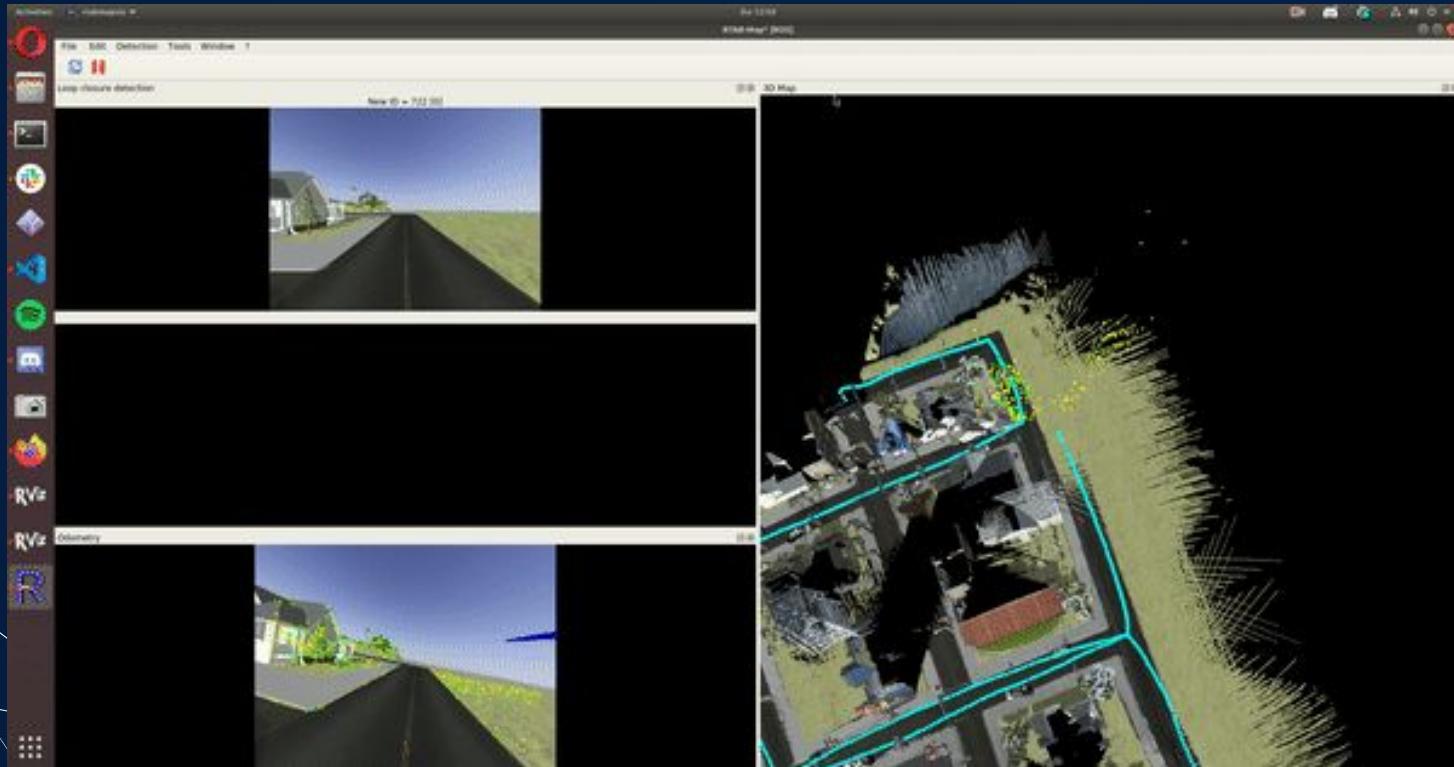
# Observatie



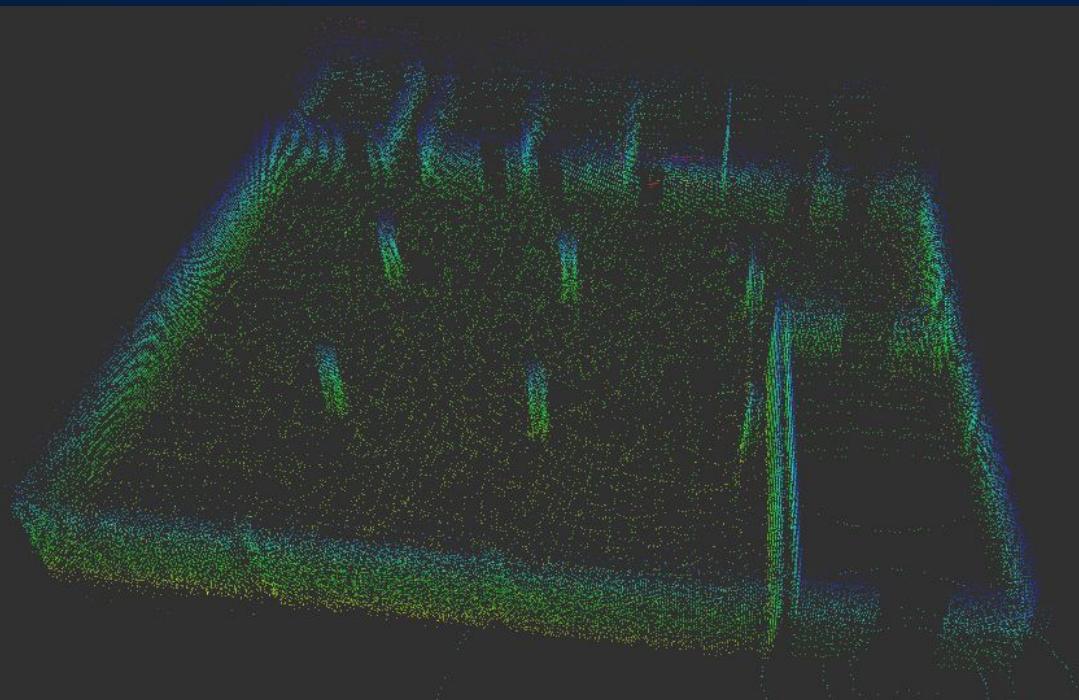
# Mappen en lokaliseren



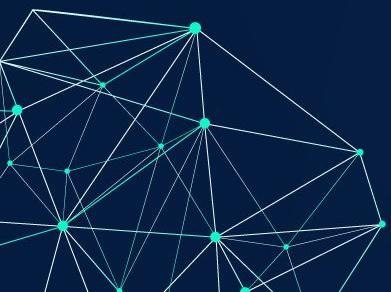
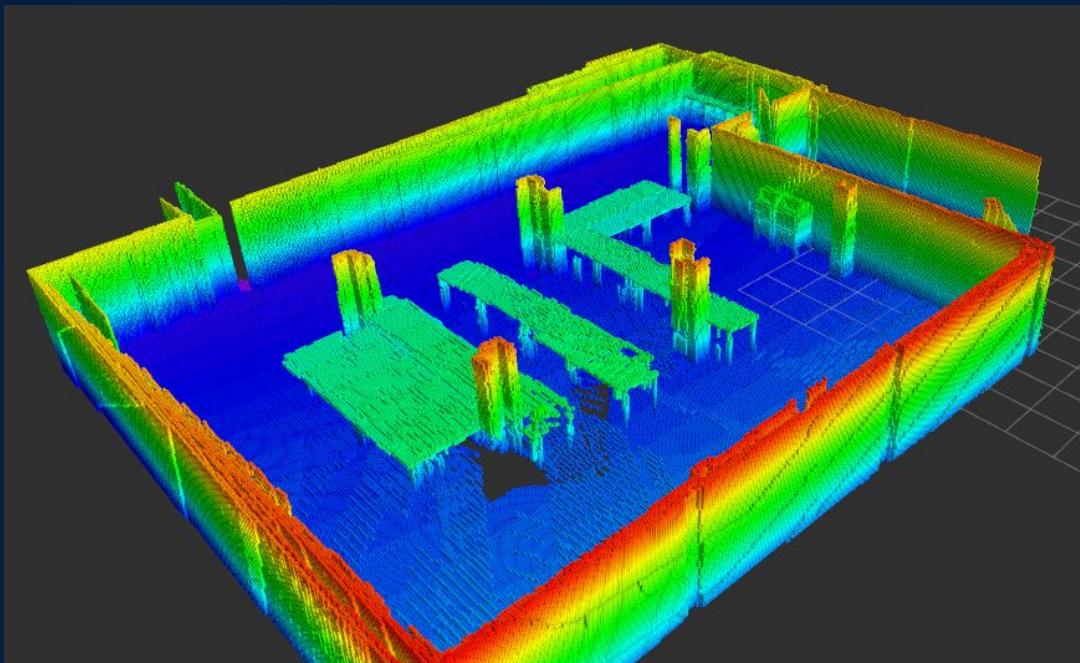
# Loop closure



# Point cloud



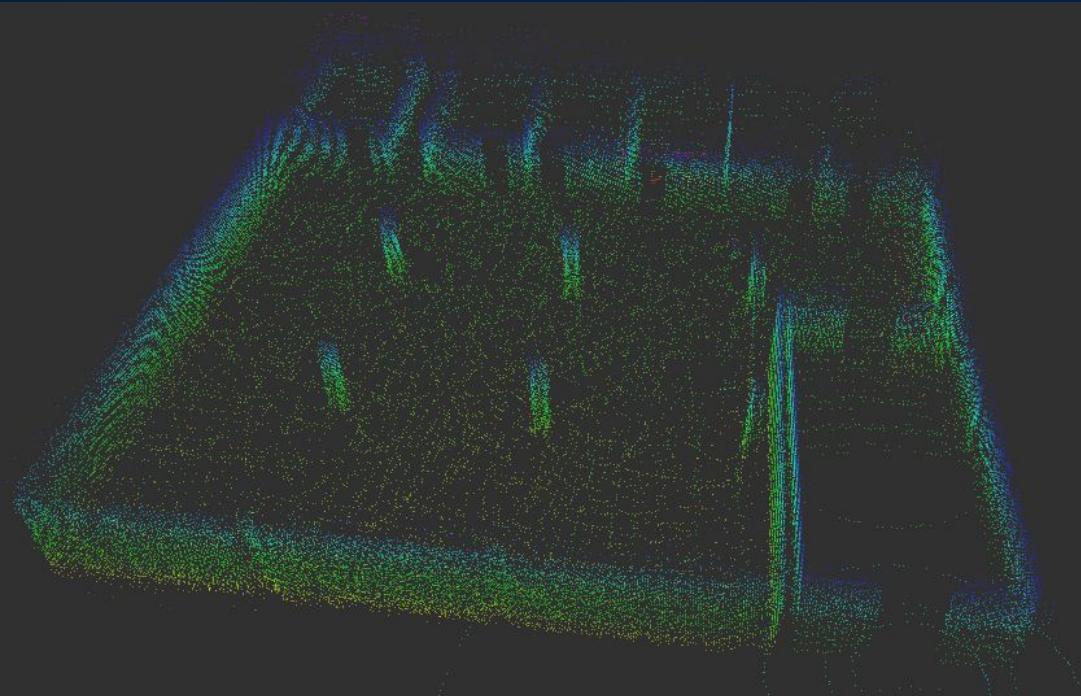
# Octomap



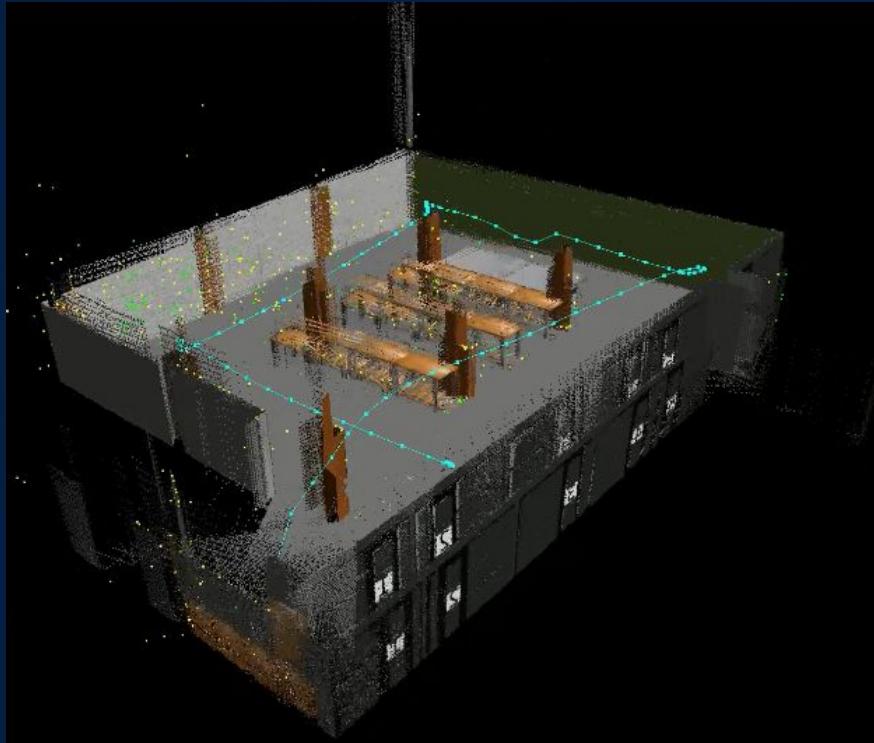
# SLAM algoritmen

Algorithm	Sensors	Point cloud	Dimentional
ORB-SLAM2	Monocular camera Stereo camera RGB-D camera	sparse	3D
Cartographer	Multiple configurations	dense	2D & 3D
BLAM!	3D LiDAR	dense	3D
hdl_graph_slam	3D LiDAR	dense	3D
RTAB-Map	Stereo camera RGB-D camera 3D LiDAR	dense	3D

# hdl\_graph\_slam

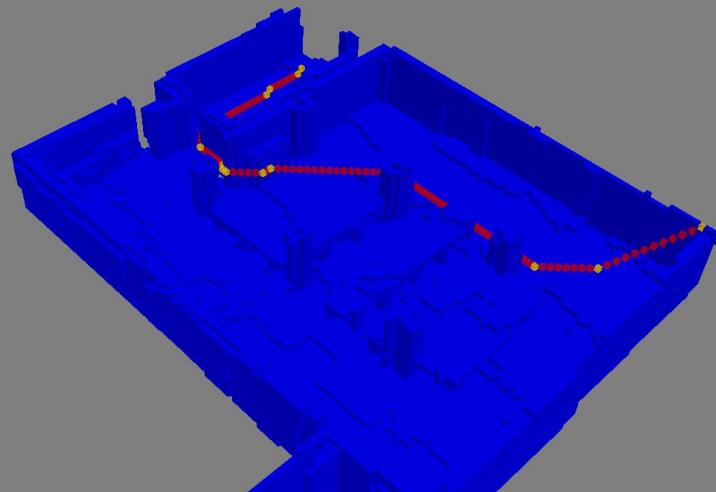


# RTAB-Map

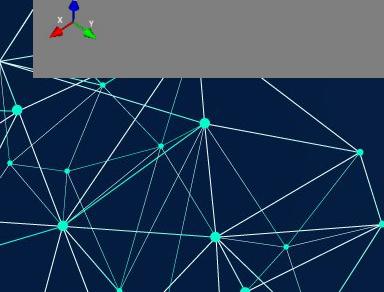


# Pad planning en uitvoering

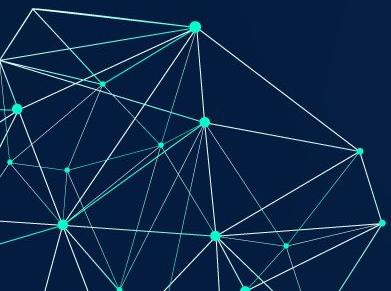
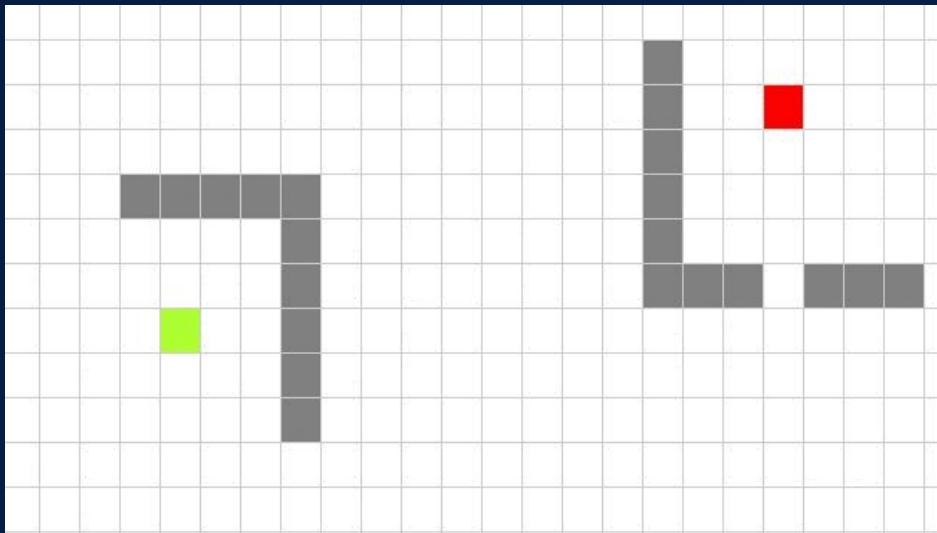
Uitvoerbaar pad



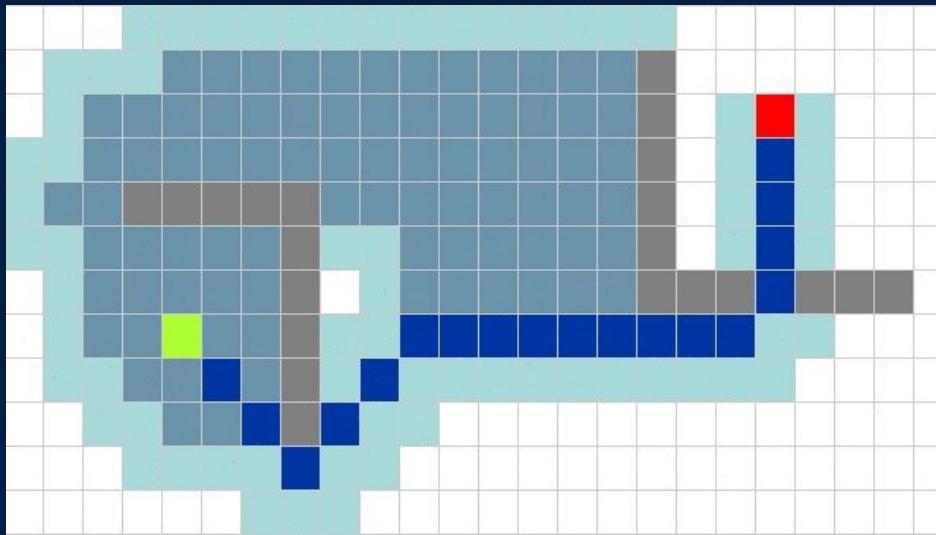
Doel



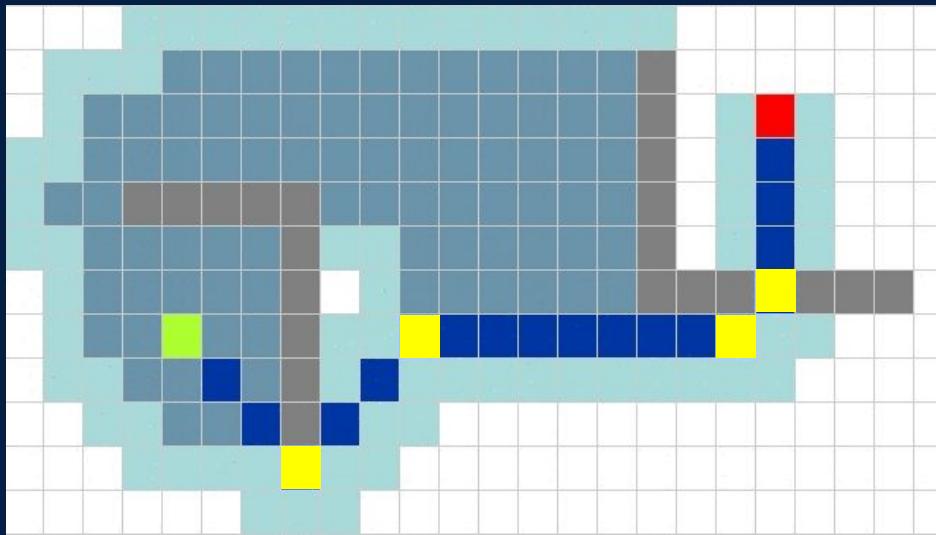
A\*



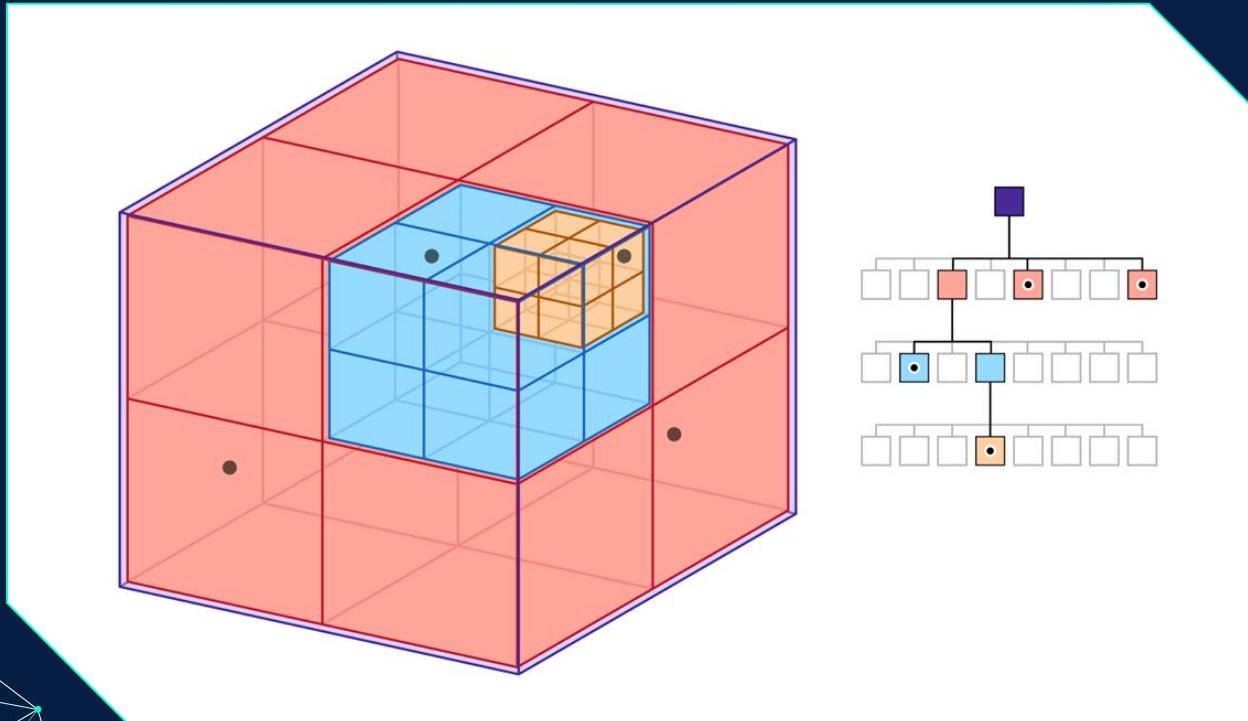
# Uitvoerbaar pad



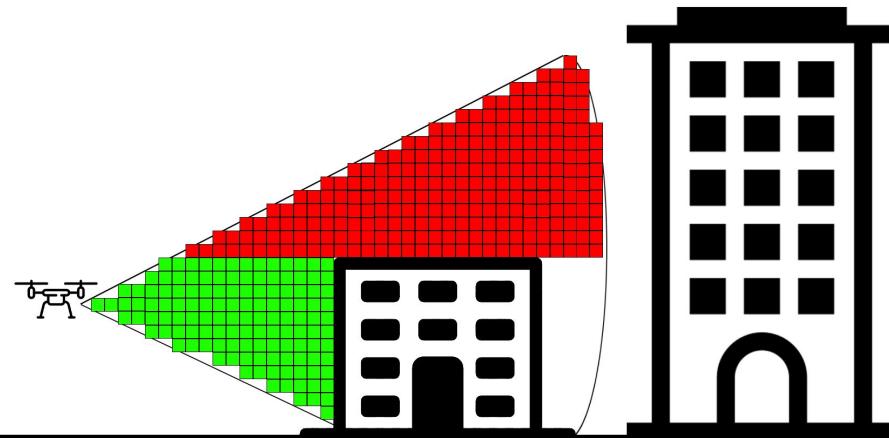
# Uitvoerbaar pad



# Octree



# Outdoor vrije punten



# Doe<sup>l</sup>

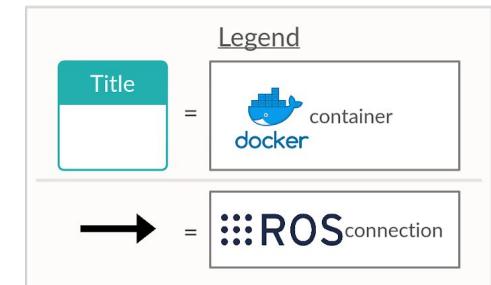
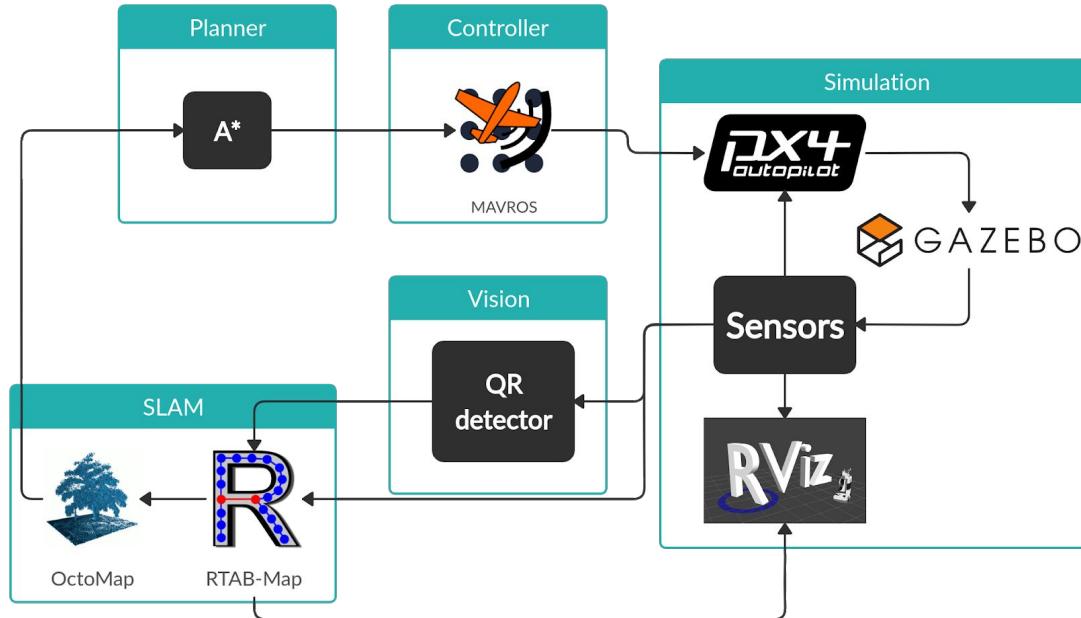


6

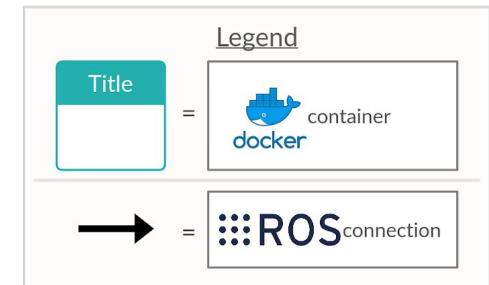
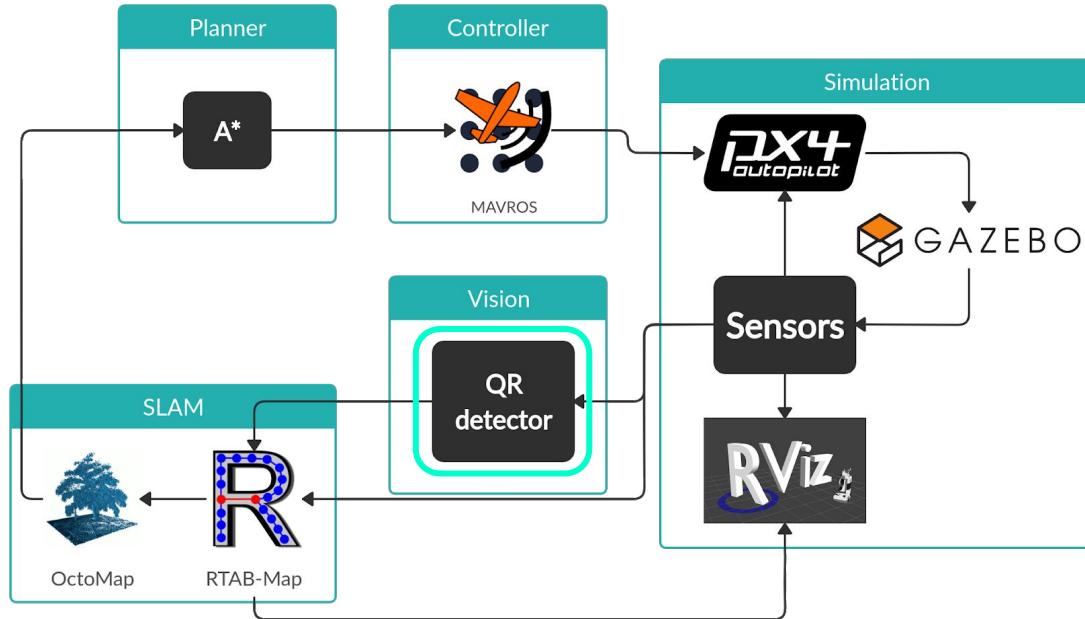
# Implementatie showcase

---

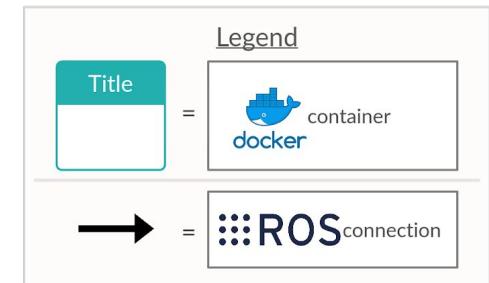
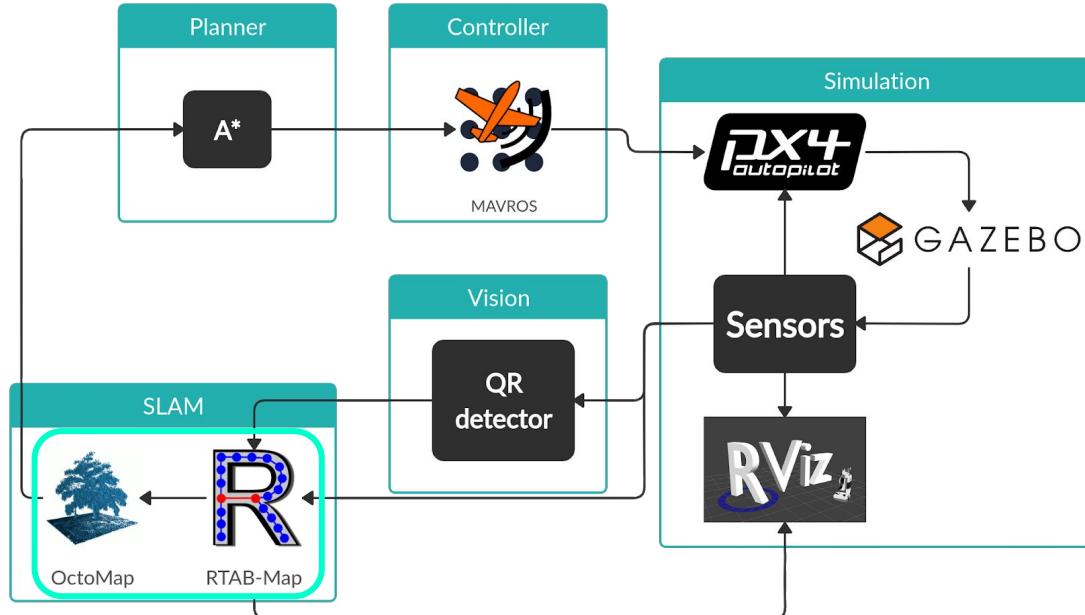
# Implementatie showcase



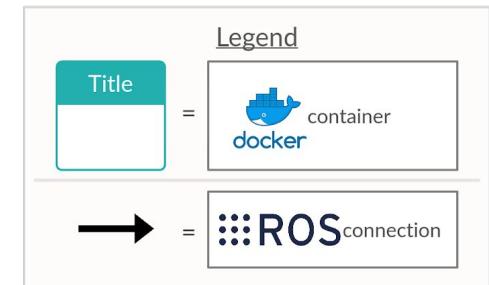
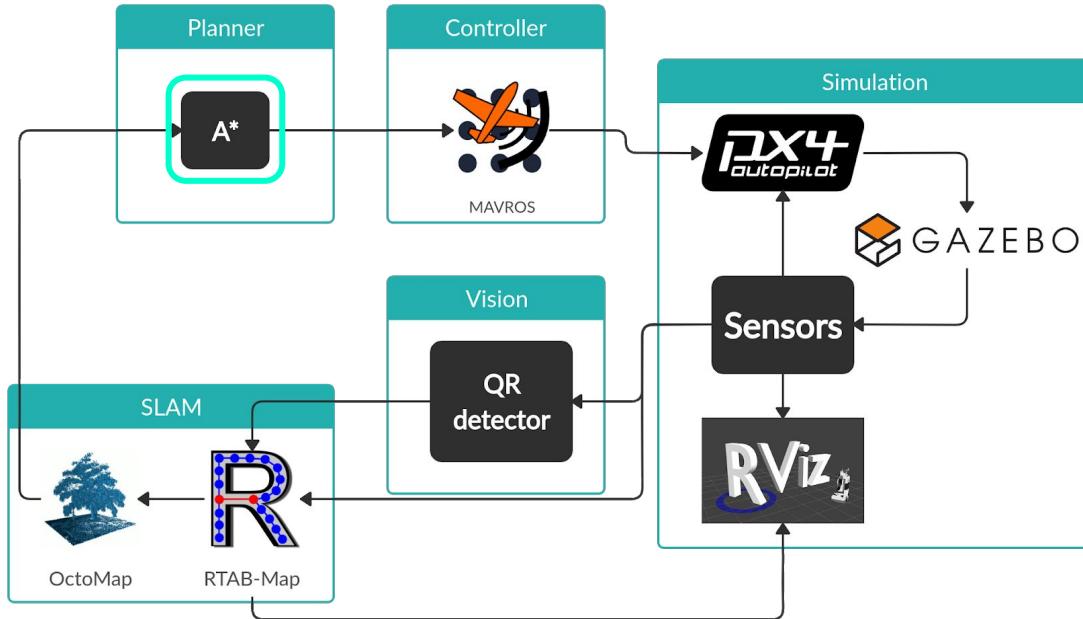
# Implementatie showcase



# Implementatie showcase



# Implementatie showcase



# 7

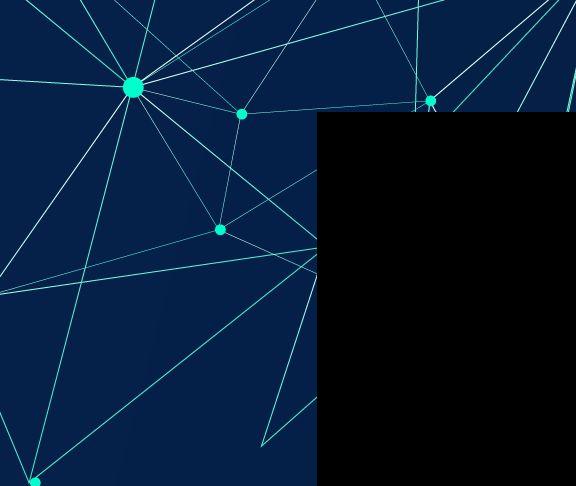
## Demo showcase

---

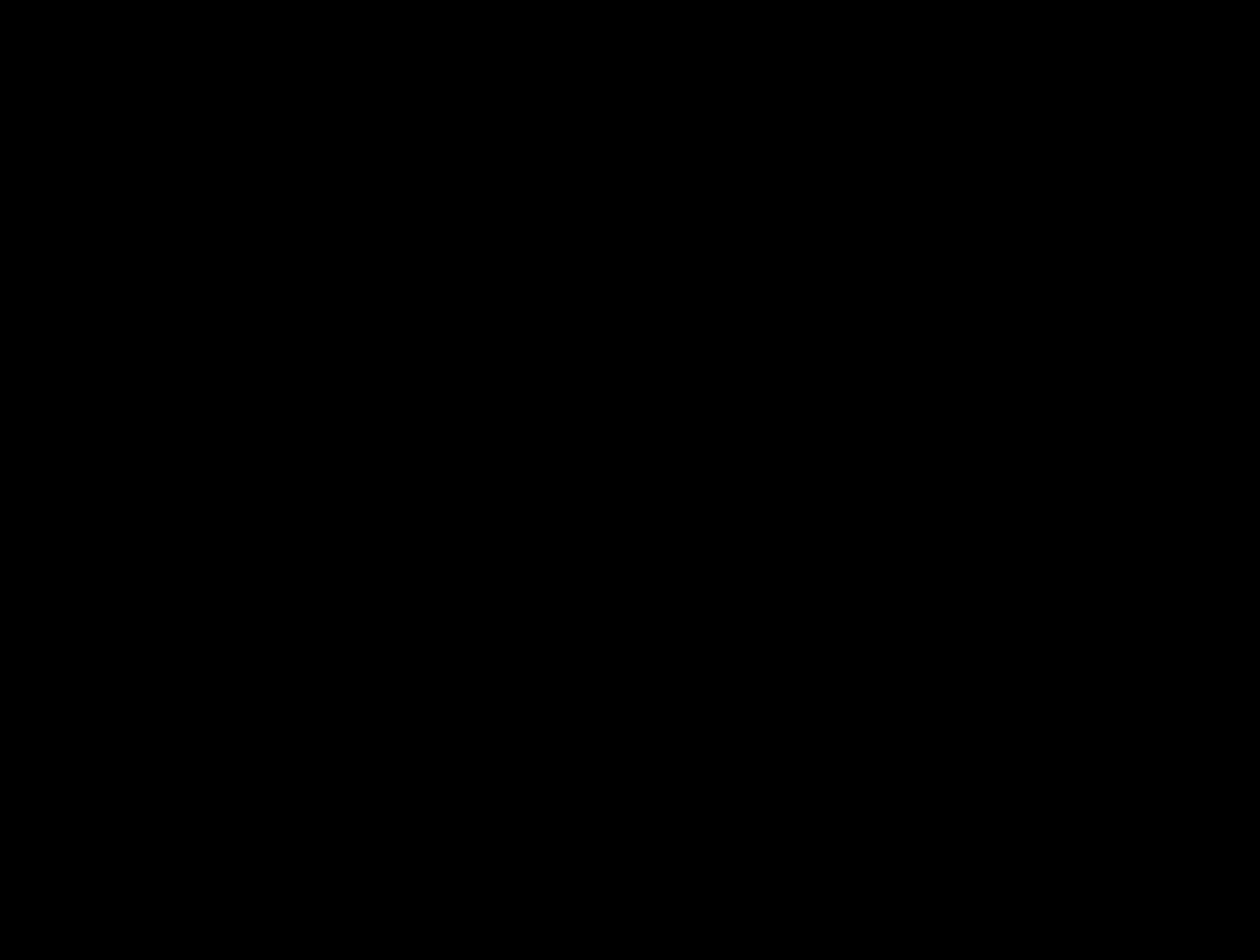




# Demo showcase indoor



# Demo showcase outdoor



# 8

## Reflectie

---





**Bedankt voor jullie  
aandacht!**