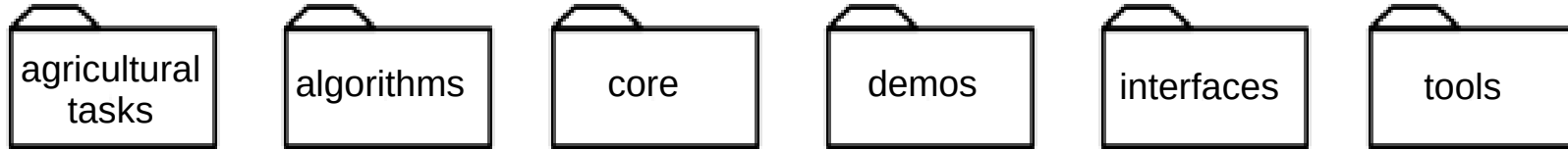
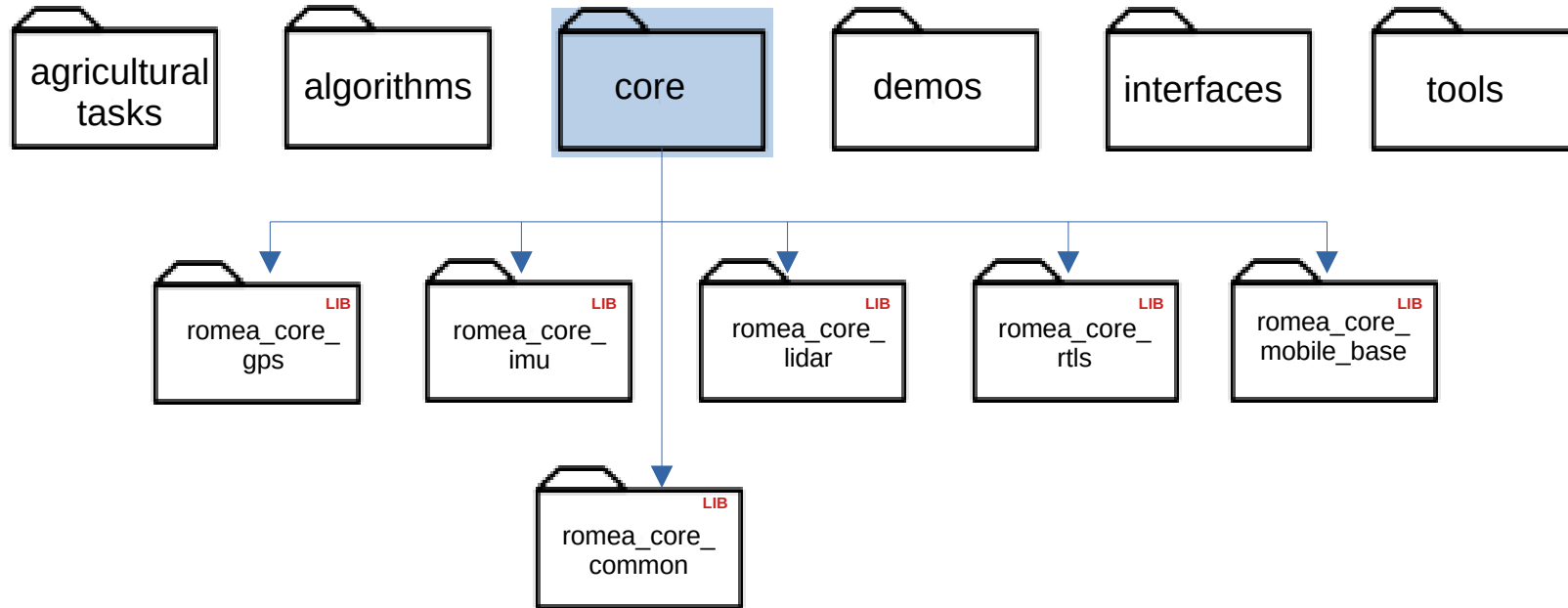


Workspace organization



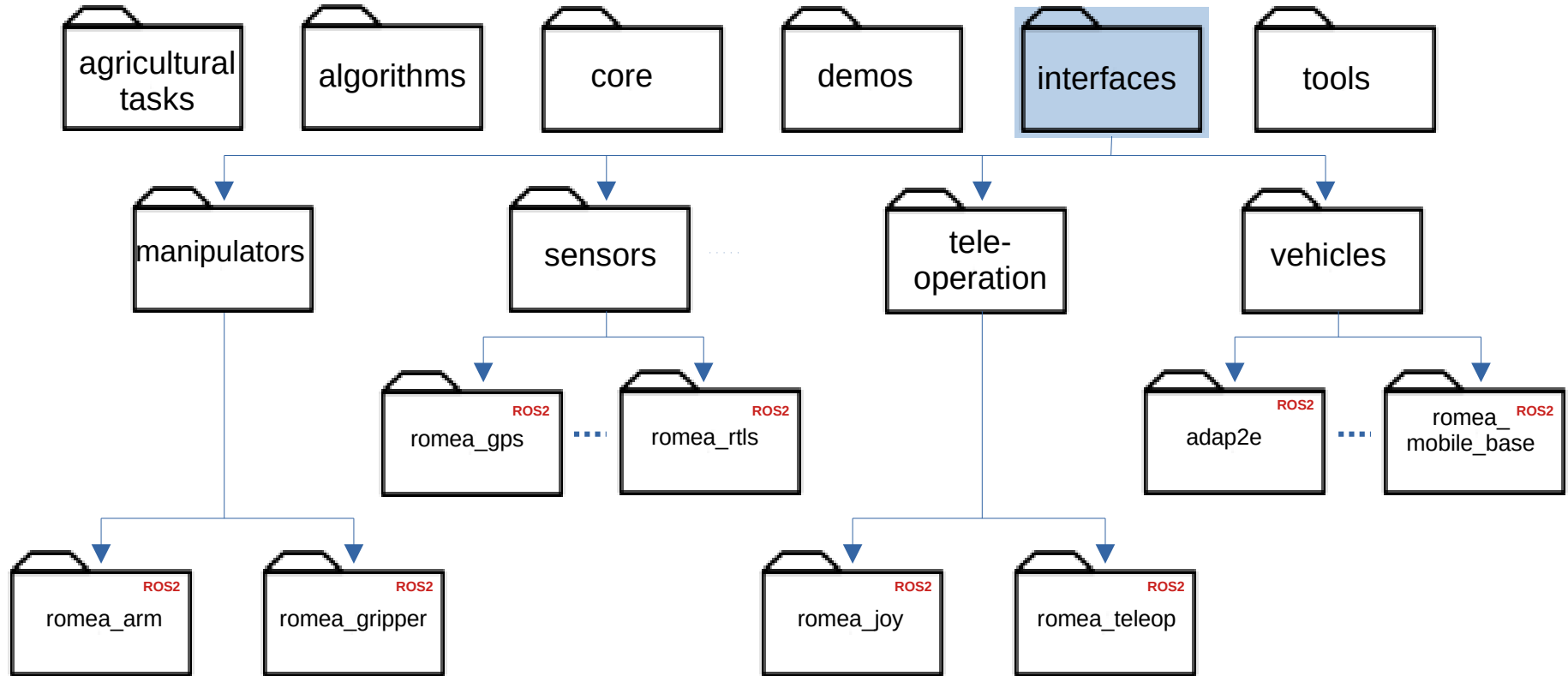
Workspace organization

Core librairies



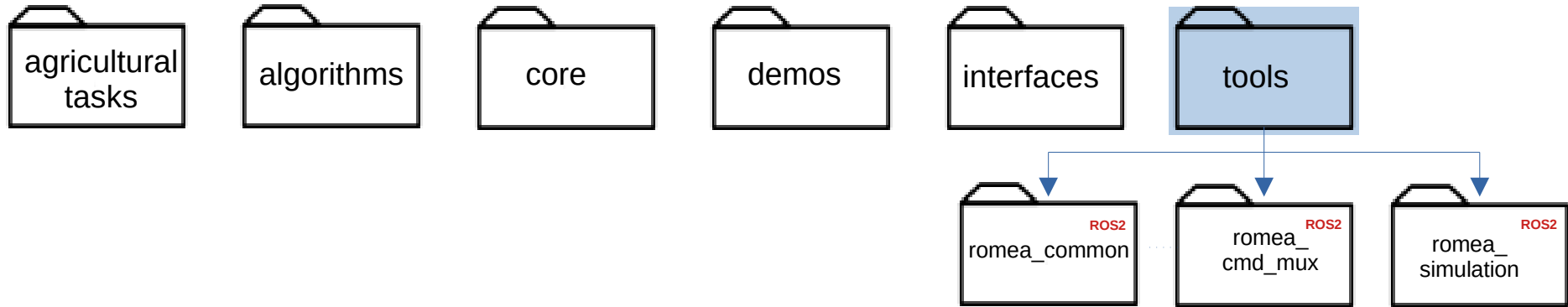
Workspace organization

Device interfaces packages



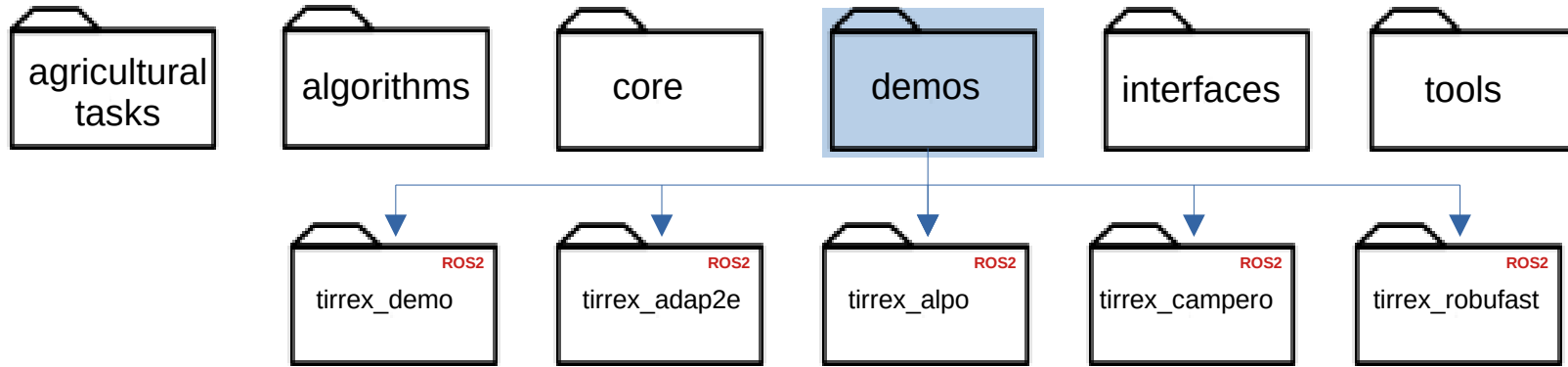
Workspace organization

Tools packages



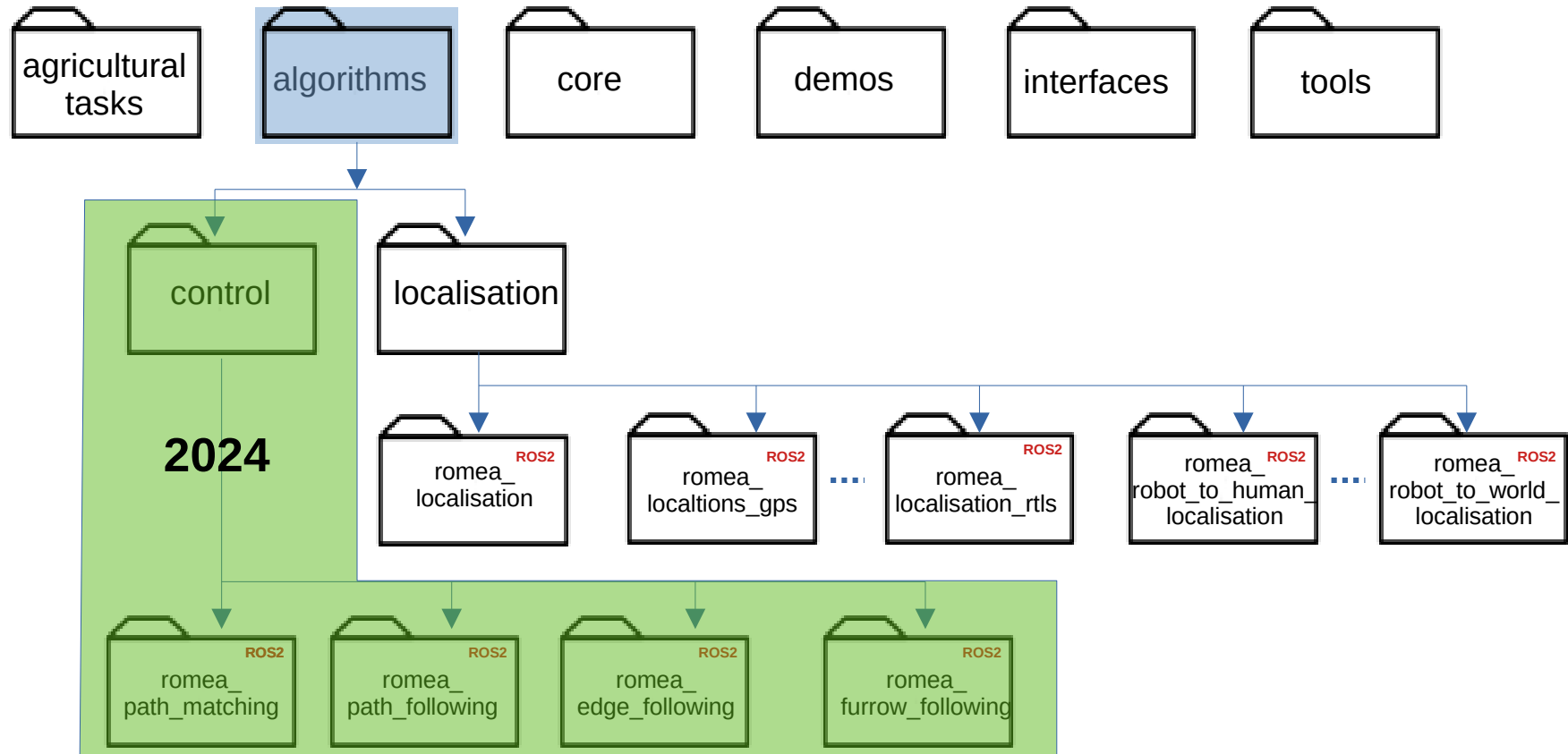
Workspace organization

Demo packages



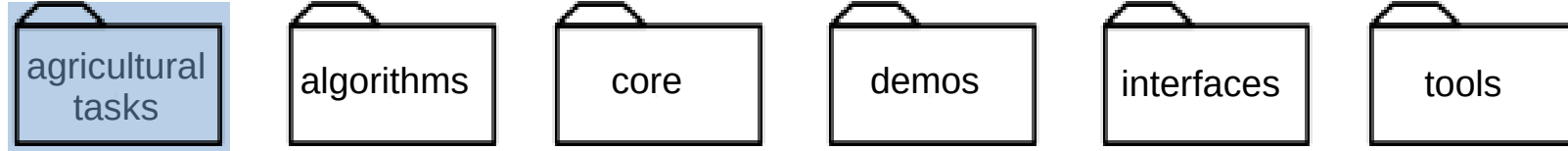
Workspace organization

Algorithms packages



Workspace organization

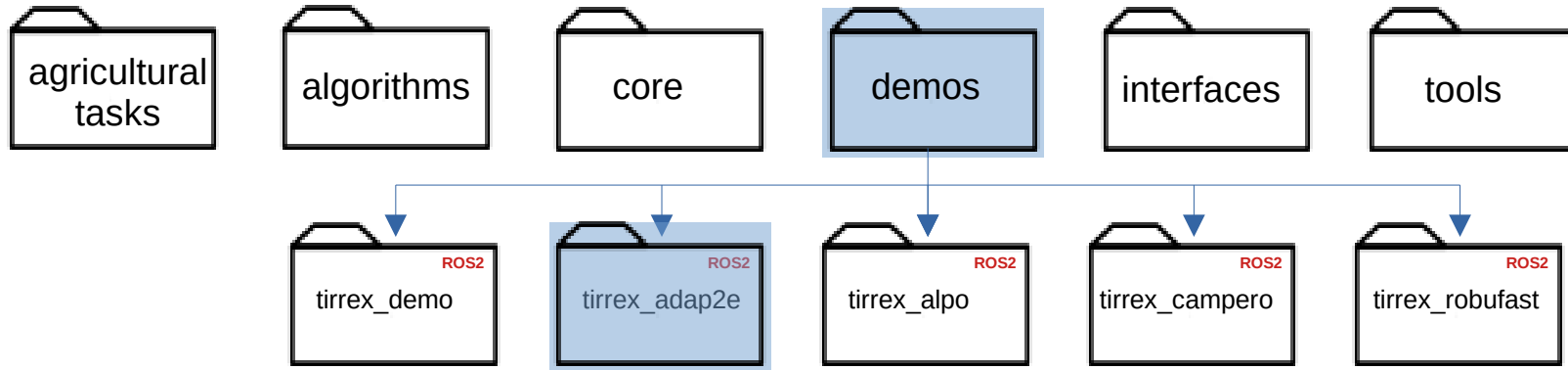
Agricultural tasks packages



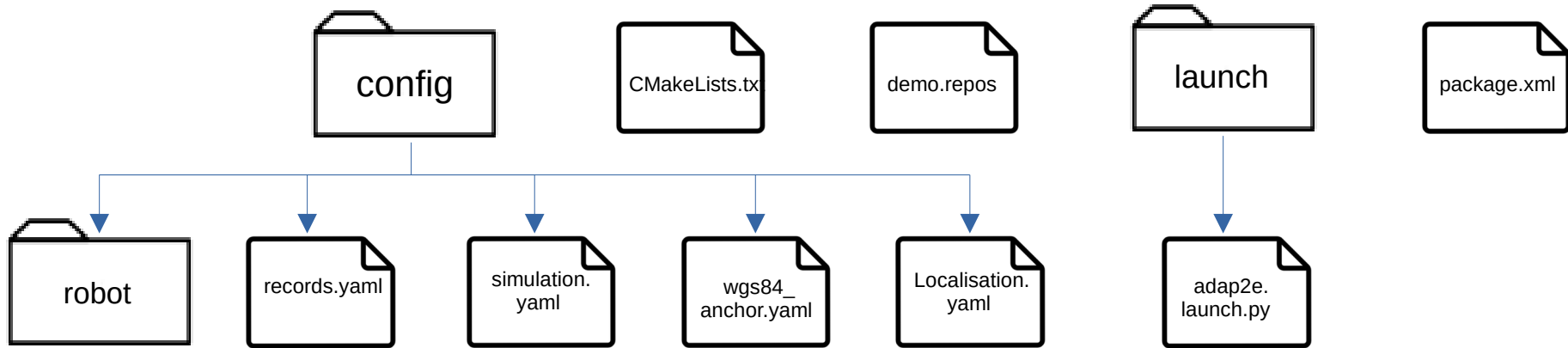
Contributions du projet Ninsar

Workspace organization

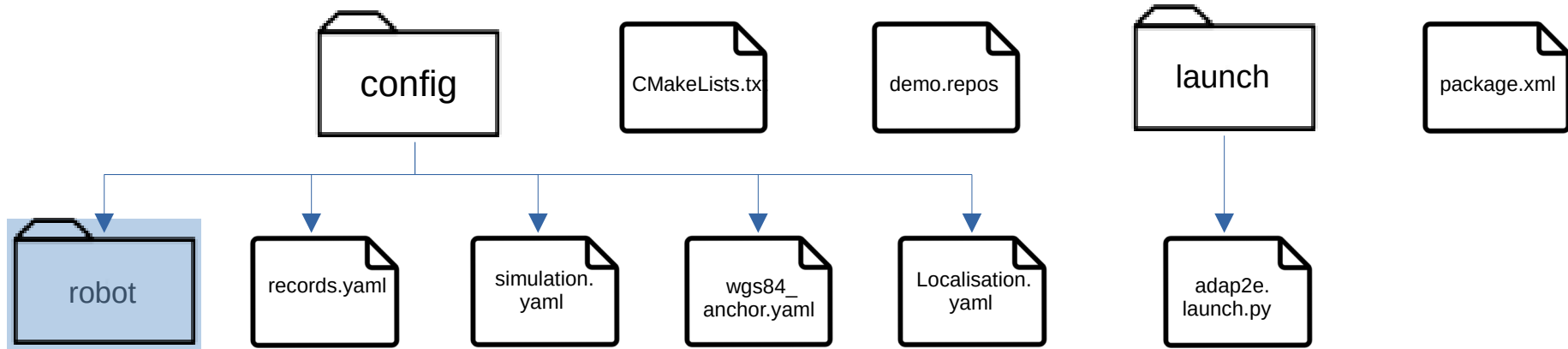
Demo packages



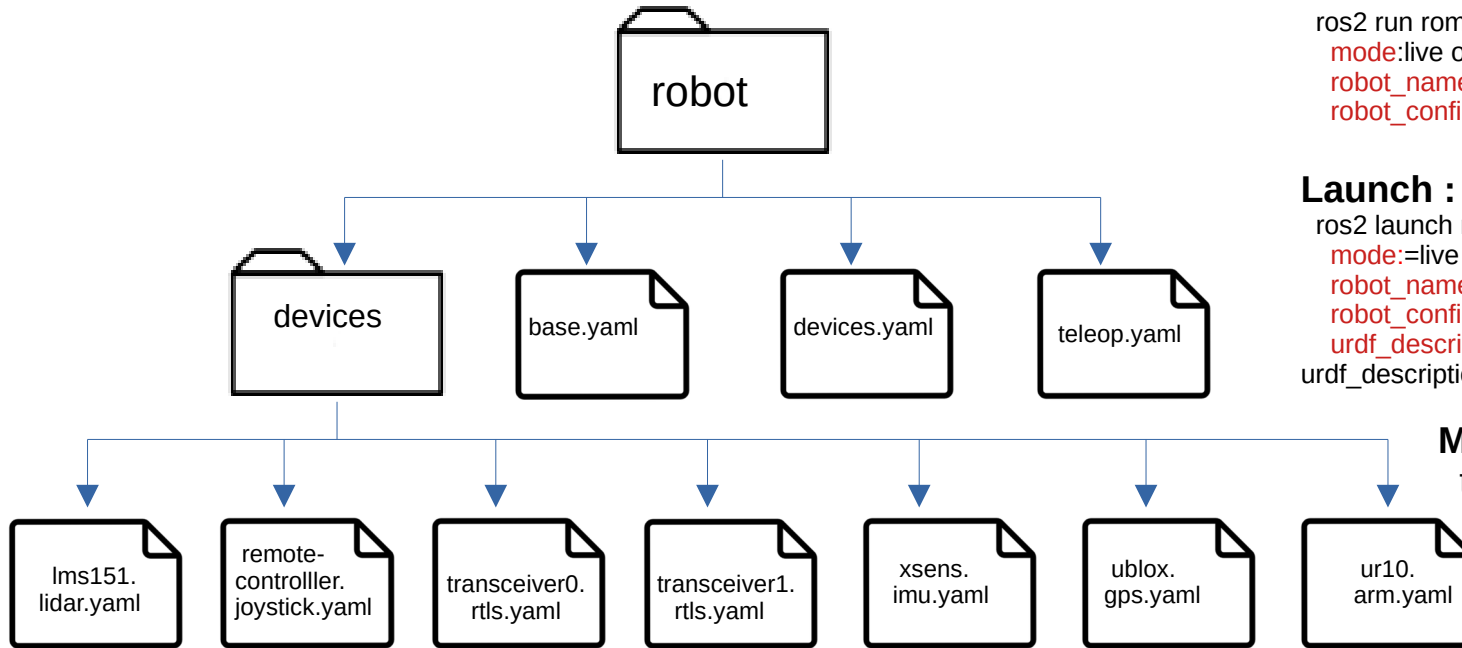
Demo organization



Demo organization



Robot configuration



URDF description :

`ros2 run romea_demo robot_description.py`

mode:live or simulation

robot_namespace:adap2e

robot_configuration_directory:robot_directory_path

Launch :

`ros2 launch romea_demo robot.launch.py`

mode:=live or simulation

robot_namespace:=adap2e

robot_configuration_directory:=robot_directory_path

urdf_description:= user_description (optional, by default

urdf_description parameter is the result of urdf description script)

More info :

tirrex_demo package ([gitlab](#),[github](#))

Examples :

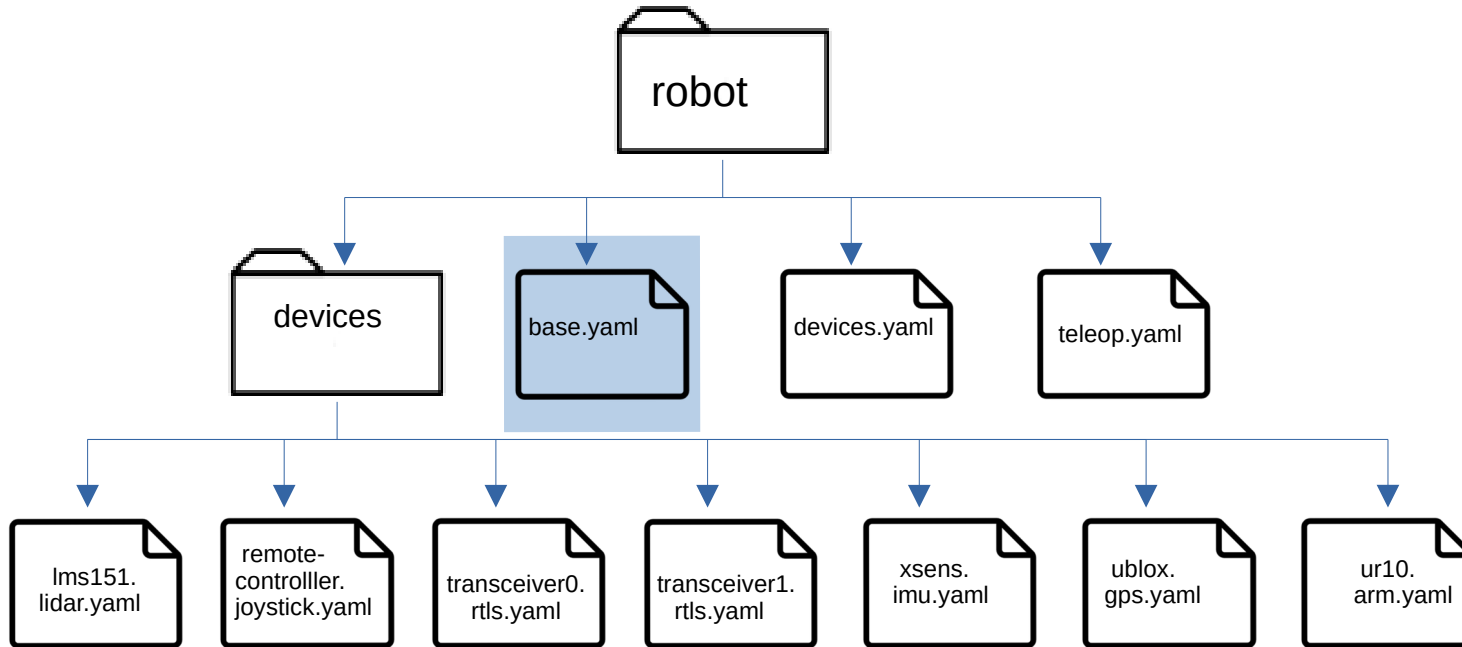
`-ros2 run tirrex_demo robot_description.py mode:live robot_namespace:adap2e robot_configuration_directory:path_to_demo/config/robot > robot.urdf`

`-ros2 launch tirrex_demo robot.launch.py mode:=live robot_namespace:=adap2e robot_configuration_directory:=path_to_demo/config/robot`

(warning in simulation mode gazebo must be launch before by using `ros2 launch gazebo_ros gazebo.launch.py`)

Robot configuration

Mobile base



Meta-description:

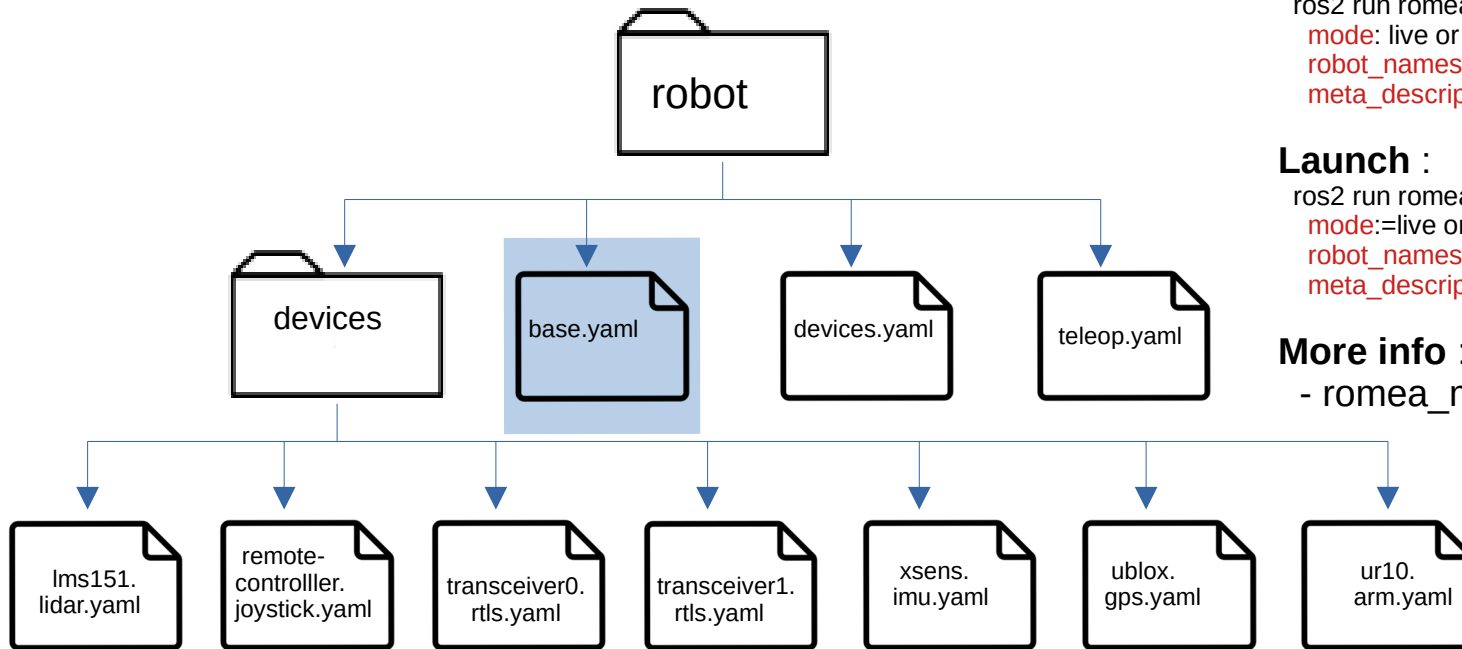
name: "base"
configuration:
 type: adap2e
 model: fat
records:
 joint_states : true
 controller/odom : true
 controller/odometry : true
 controller/kinematic : true
simulation:
 initial_xyz: [0.0, 0.0, 0.0] # meters
 initial_rpy: [0.0, 0.0, 0.0] # degrees

Supported robots :

- adap2e ([gitlab](#), [github](#))
- alpo ([gitlab](#))
- campero ([gitlab](#))
- robufast ([gitlab](#), [github](#))

Robot configuration

Mobile base



URDF Description:

`ros2 run romea_mobile_base_bringup urdf_description.py`
`mode`: live or simulation
`robot_namespace`: adap2e
`meta_description_file_path`: path to base meta-description file

Launch :

`ros2 run romea_mobile_base_bringup mobile_base.launch.py`
`mode`: =live or simulation
`robot_namespace`: =adap2e
`meta_description_file_path`: =path to base meta-description file

More info :

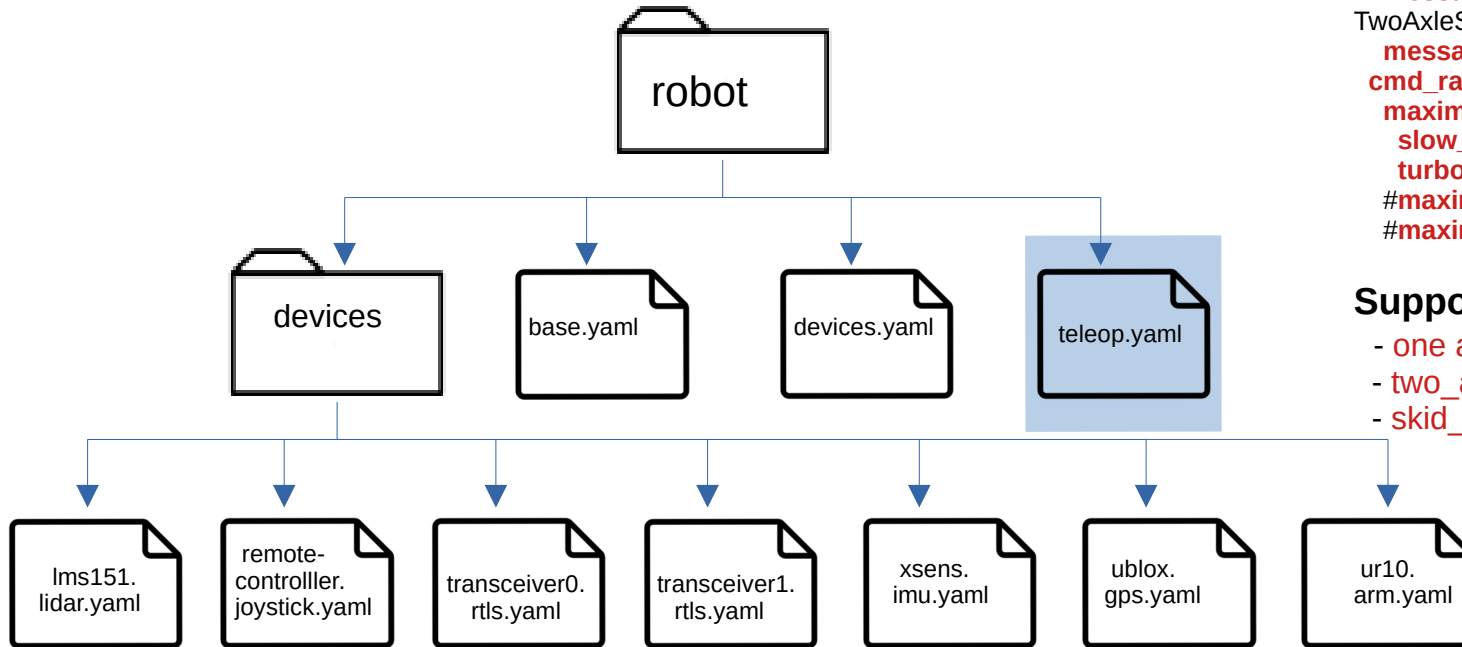
- romea_mobile_base stack ([gitlab](#), [github](#))

Examples :

`-ros2 run romea_mobile_base_bringup urdf_description.py mode:live robot_namespace:adap2e meta_description_file_path:path_to_demo/config/robot/base.yaml > base.urdf`
`-ros2 launch romea_mobile_base_bringup robot.launch.py mode:=live robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/base.yaml`
(warning in simulation mode gazebo must be launch before by using `ros2 launch gazebo_ros gazebo.launch.py`)

Robot configuration

Teleoperation



Meta-description:

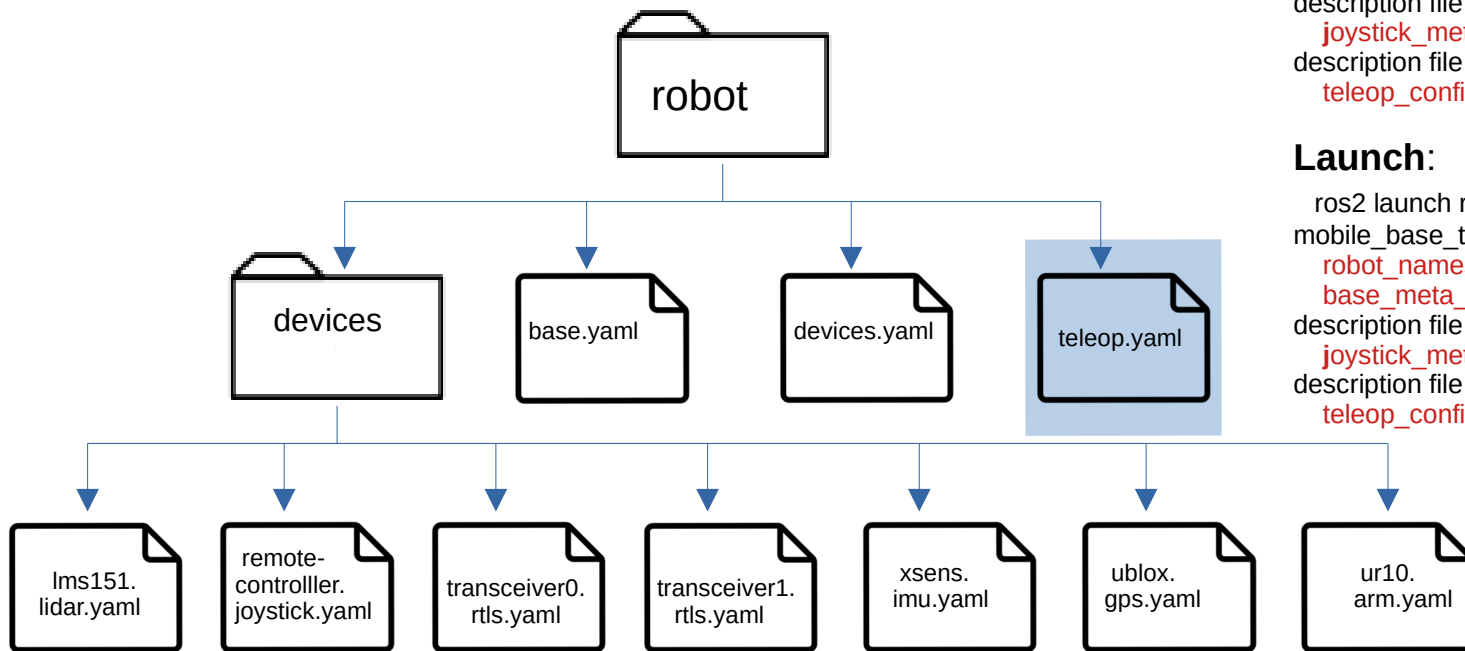
cmd_output:
message_type: romea_mobile_base_msgs/
TwoAxleSteeringCommand
message_priority: 100
cmd_range:
maximal_linear_speed:
slow_mode: 1.0 # m/s
turbo_mode: 2.0 # m/s
#maximal_front_steering_angle: 20.0 # degree, optional
#maximal_rear_steering_angle: 0.0 # degree, optional

Supported command type:

- one axle steering (alpo)
- two_axle_steering (robustfast, adap2e)
- skid_steering (campero, effibote3)

Robot configuration

Teleoperation



Joystick mapping (TODO) :

`ros2 run romea_teleop_bringup joystick_remapping.py`
`base_meta_description_file_path`: path to mobile base meta-description file
`joystick_meta_description_file_path`: path to joystick meta-description file
`teleop_configuration_file_path` : path to teleop configuration file

Launch:

`ros2 launch romea_teleop_bringup`
`mobile_base_teleop.launch.py`
`robot_namespace`: adap2e
`base_meta_description_file_path`: path to mobile base meta-description file
`joystick_meta_description_file_path`: path to joystick meta-description file
`teleop_configuration_file_path` : path to teleop configuration file

More info :

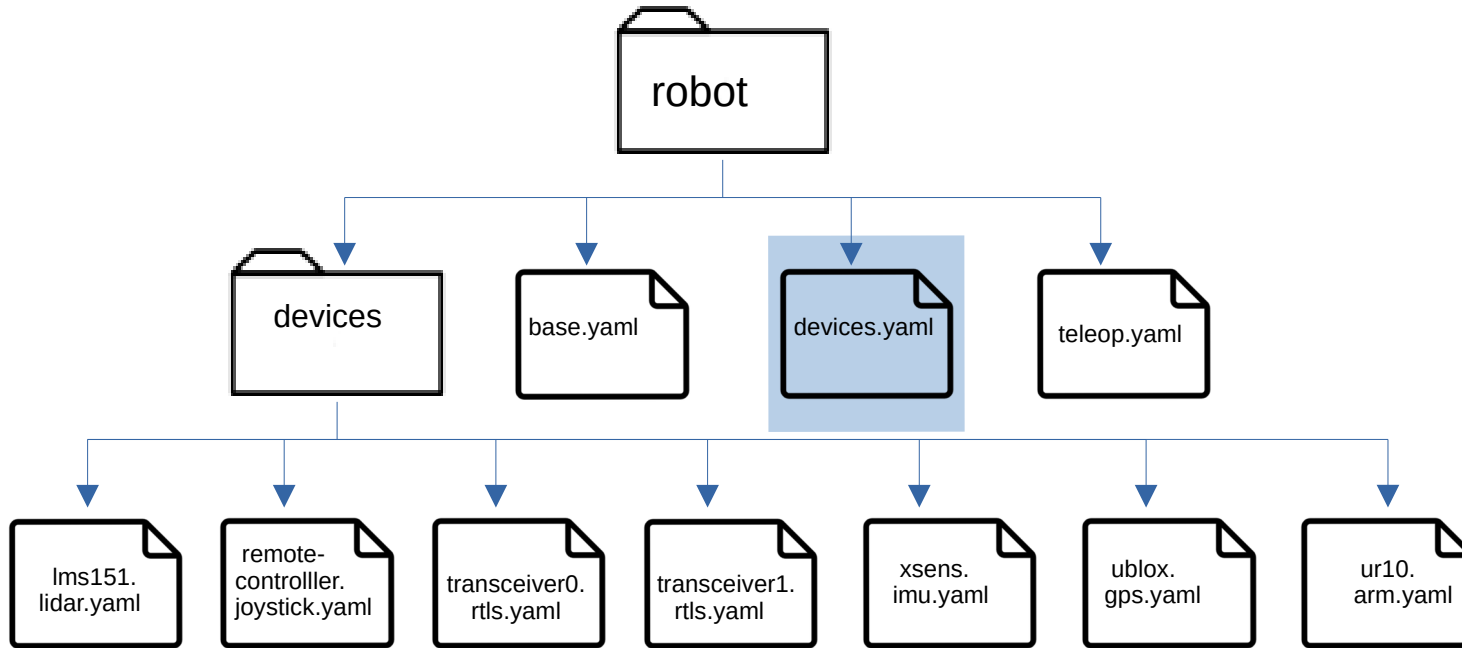
romea_teleop stack ([gitlab](#), [github](#))

Examples :

`-ros2 launch romea_teleop_bringup mobile_base_teleop.launch.py robot_namespace:=adap2e base_meta_description_file_path:=path_to_demo/config/robot/base.yaml joystick_meta_description_file_path:=path_to_demo/robot/devices/remote_controller.joytick.yaml teleop_configuration_file_path:=path_to_demo/config/robot/teleop.yaml`

Robot configuration

Devices selection



Devices configuration:

remote_controller:

type: joystick

available_mode: all

lms151:

type: lidar

available_mode: live

ublox:

type: gps

available_mode: none

xsens:

type: imu

available_mode: none

transceiver0:

type: rtls

available_mode: simulation

transceiver1:

type: rtls

available_mode: simulation

ur10:

type: arm

available_mode: simulation

More info :

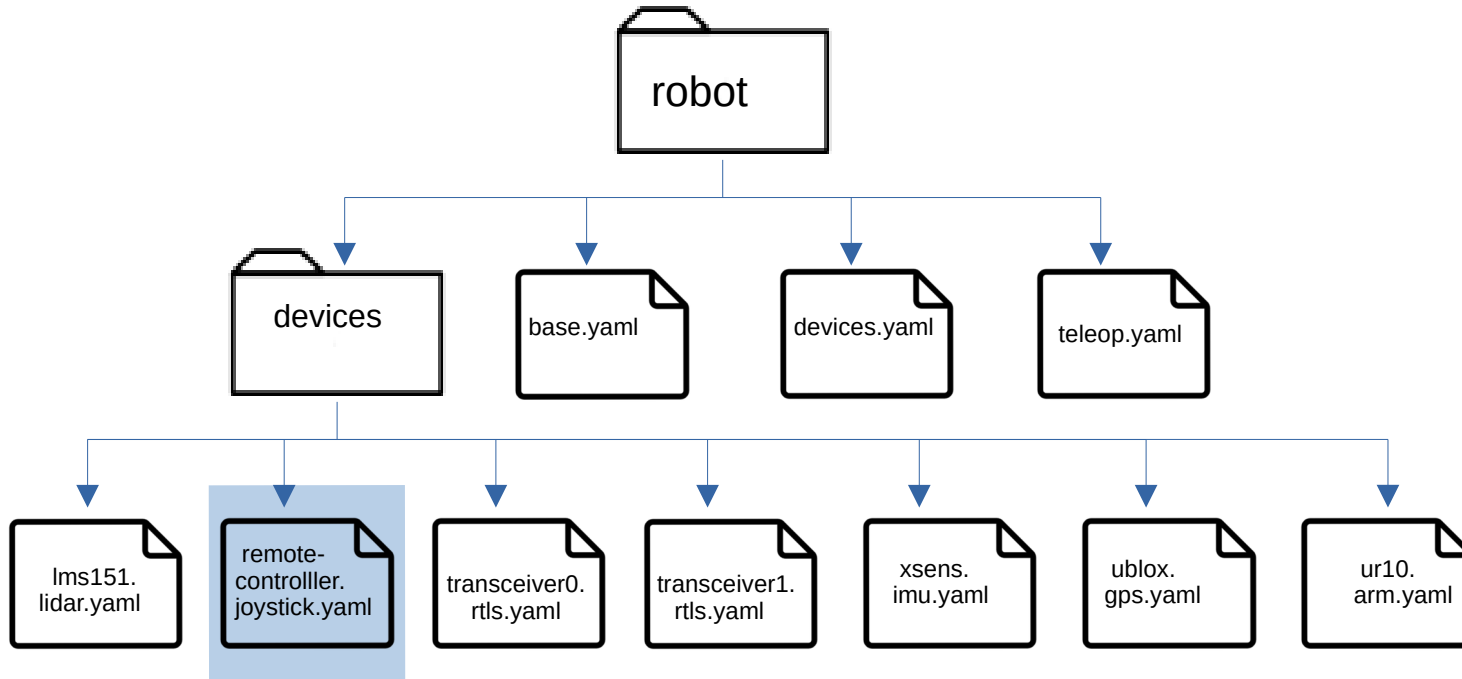
- tirrex_demo package ([gitlab](#), [github](#))

Examples :

-ros2 launch romea_demo_robot_devices.launch.py configuration_filename:=path_to_demo/config/robot/devices.yaml

Robot configuration

Joystick



Meta-description :

name: "joystick"
driver:
pkg: "joy"
device: "/dev/input/js0"
autorepeat_rate: 10.0
deadzone: 0.1
configuration:
type: xbox
records:
joy: true

Supported driver packages:

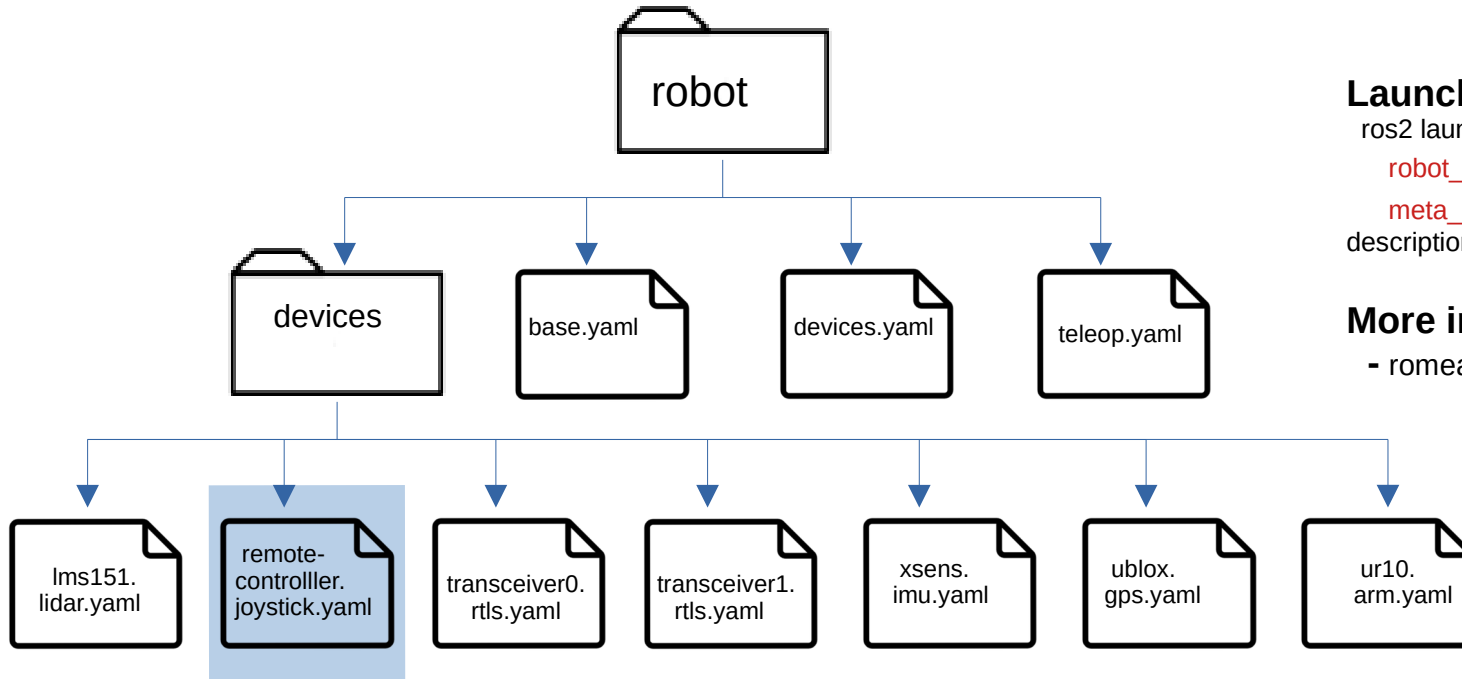
- joy
- ds4_driver

Supported joysticks:

- xbox (360, one)
- dualshock4

Robot configuration

Joystick



Launch :

```
ros2 launch romea_joy_bringup joystick_driver.launch.py  
  robot_namespace:=adap2e (optional, default = "")  
  meta_description_file_path:= path to joystick meta-  
  description file
```

More infos :

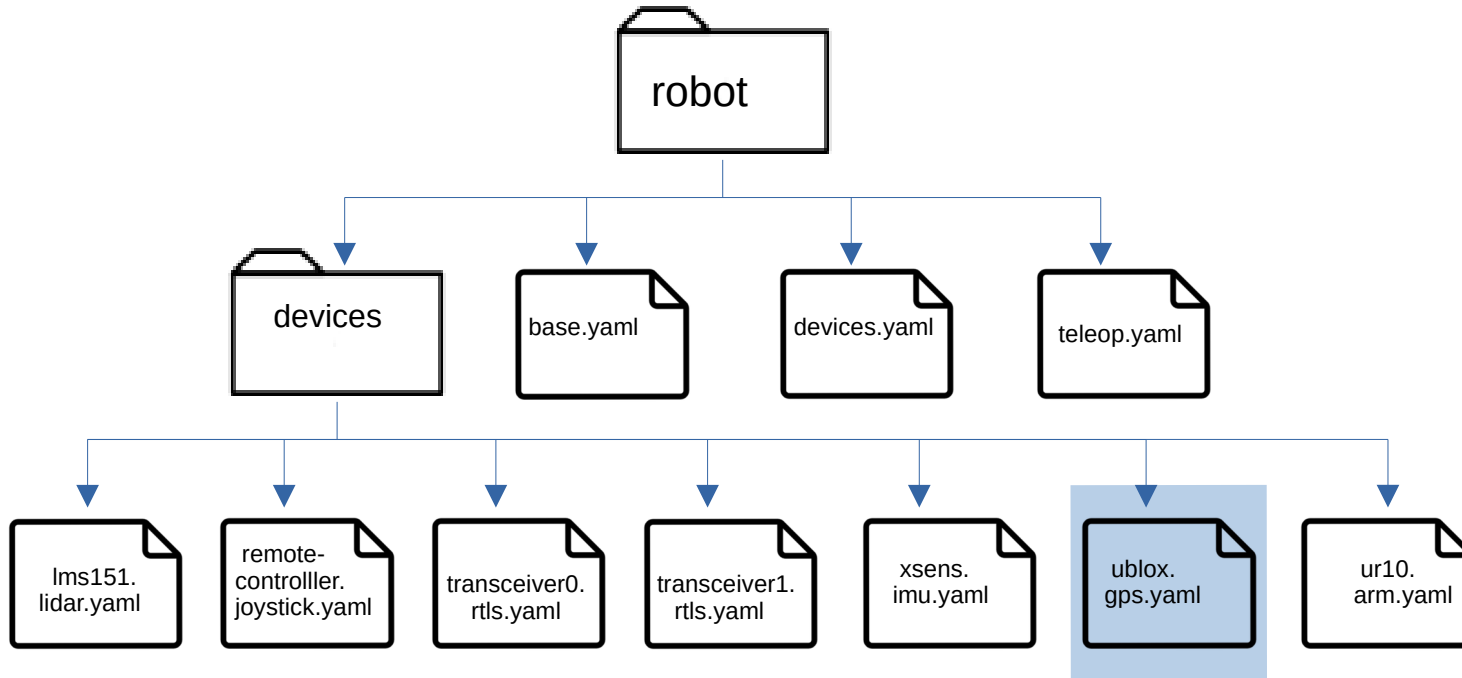
- romea_joy stack (gitlab, github)

Example :

```
-ros2 launch romea_joystick_bringup joystick_driver.launch.py robot_namespace:=adap2e meta_description_filename:=path_to_demo/robot/config/devices/remote-controller.joystick.yaml
```

Robot configuration

GPS



Meta-description :

name: "gps"
driver:
 pkg: "roмеa_ublox_driver"
 device: "/dev/ttyACM0"
 baudrate: 115200
ntrip: # optional
 pkg: "ntrip_client"
 host: caster.centipede.fr
 port: 2101
 username: centipede # optional
 password: centipede # optional
 mountpoint: MAGC
configuration:
 type: drotek
 model: fp9
 rate: 10 # hz
geometry:
 parent_link: "base_link"
 xyz: [0.0, 0.0, 1.5] #meters
records:
 nmea_sentence: true
 gps_fix: false
 vel: false

Supported driver packages :

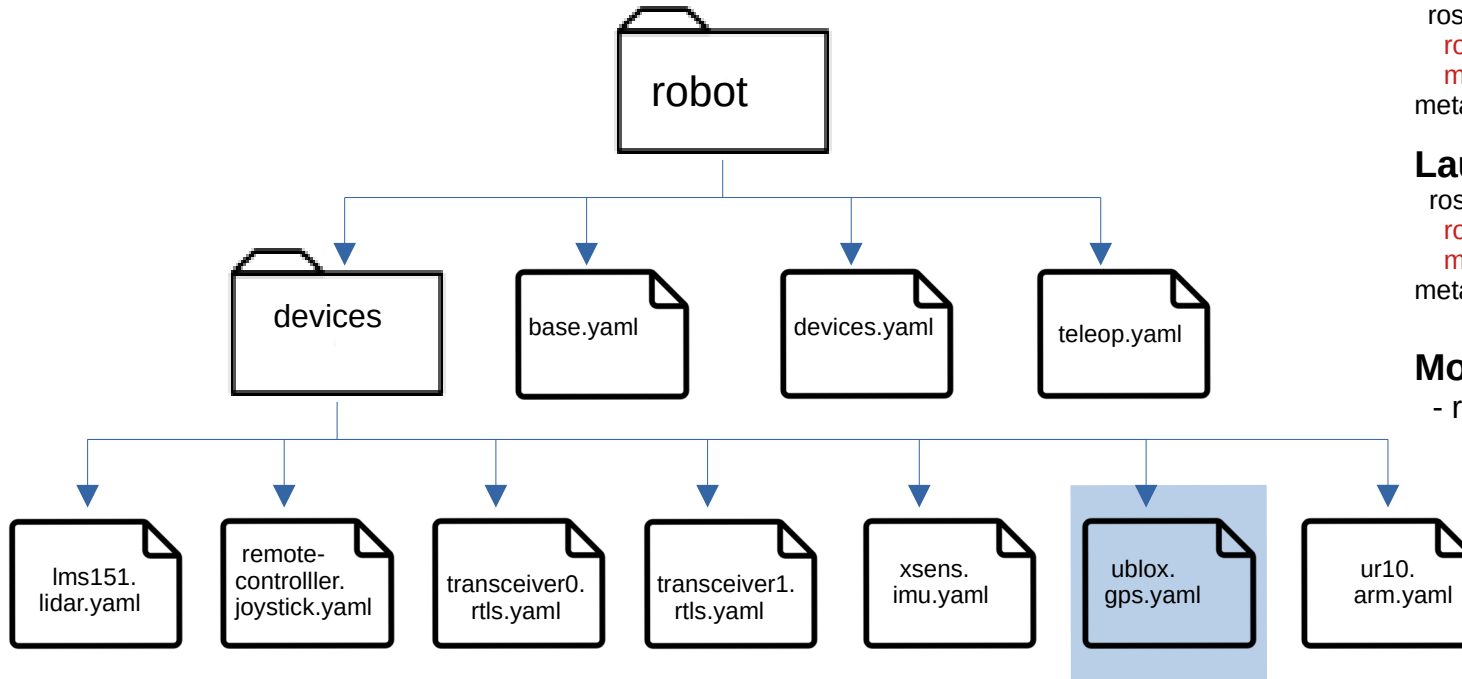
- [nmea_navsat_driver](#)
- [roмеa_ublox_driver](#)
- [ntrip_client](#)

Supported GPS receivers

- ublox (drotek fp9, ublox evk m8)
- ashtech proflex800

Robot configuration

GPS



URDF Description :

`ros2 run roma_gps_bringup urdf_description.py`
`robot_namespace:=adap2e`
`meta_description_file_path:=` absolute path to gps meta-description file

Launch :

`ros2 run roma_gps_bringup gps_driver.launch.py`
`robot_namespace:=adap2e`
`meta_description_file_path:=` absolute path to gps meta-description file

More info :

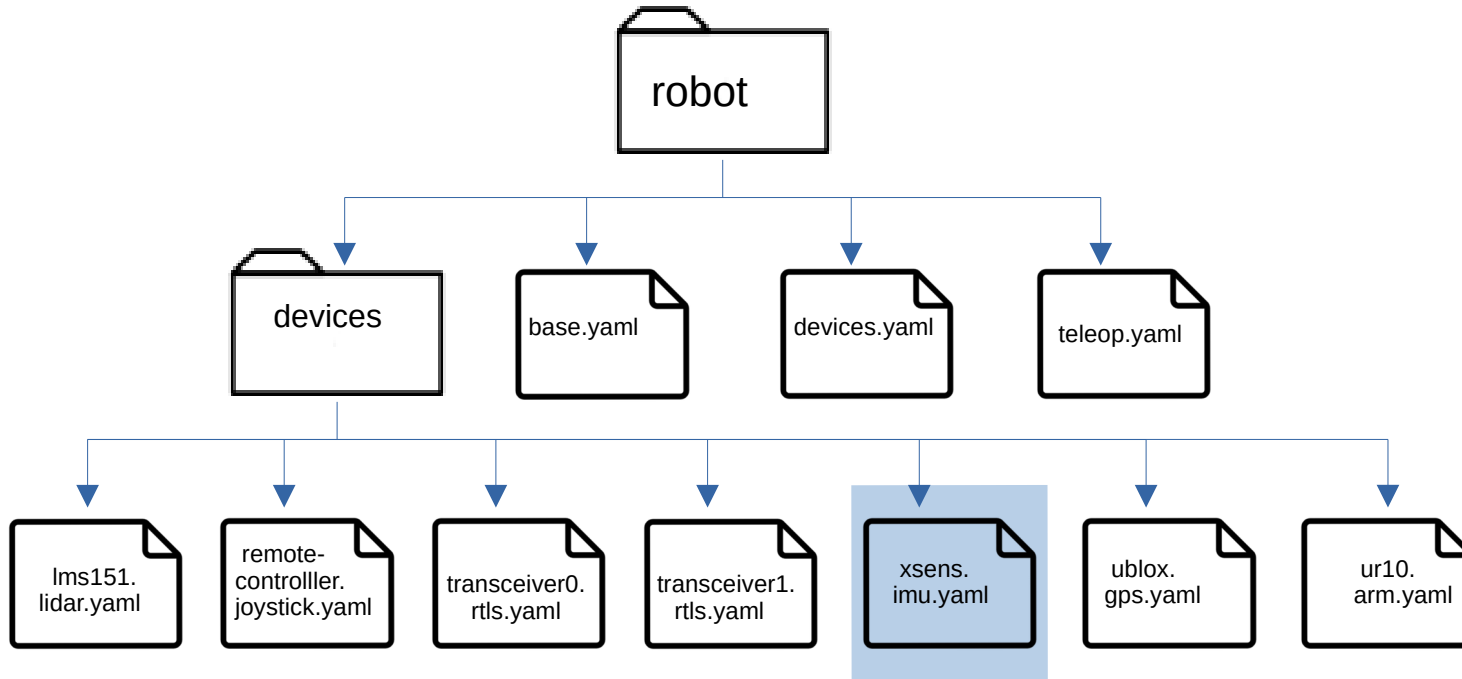
- romea_gps stack ([gitlab](#), [github](#))

Examples :

`-ros2 run romea_gps_bringup urdf_description.py robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/ublox.gps.yaml > ublox.urdf`
`-ros2 launch romea_gps_bringup gps_driver.launch.py robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/ublox.gps.yaml`

Robot configuration

IMU



Meta-description :

name: "imu"

driver:

pkg: "xsens_driver"

device: "/dev/ttyUSB0"

baudrate: 115200

configuration:

type: xsens

model: mti

rate: 100 #hz

geometry:

parent_link: "base_link"

xyz: [0.0, 0.0, 1.0] #meters

rpy: [0.0, 0.0, 0.0] #degrees

records:

data: true

Supported driver packages :

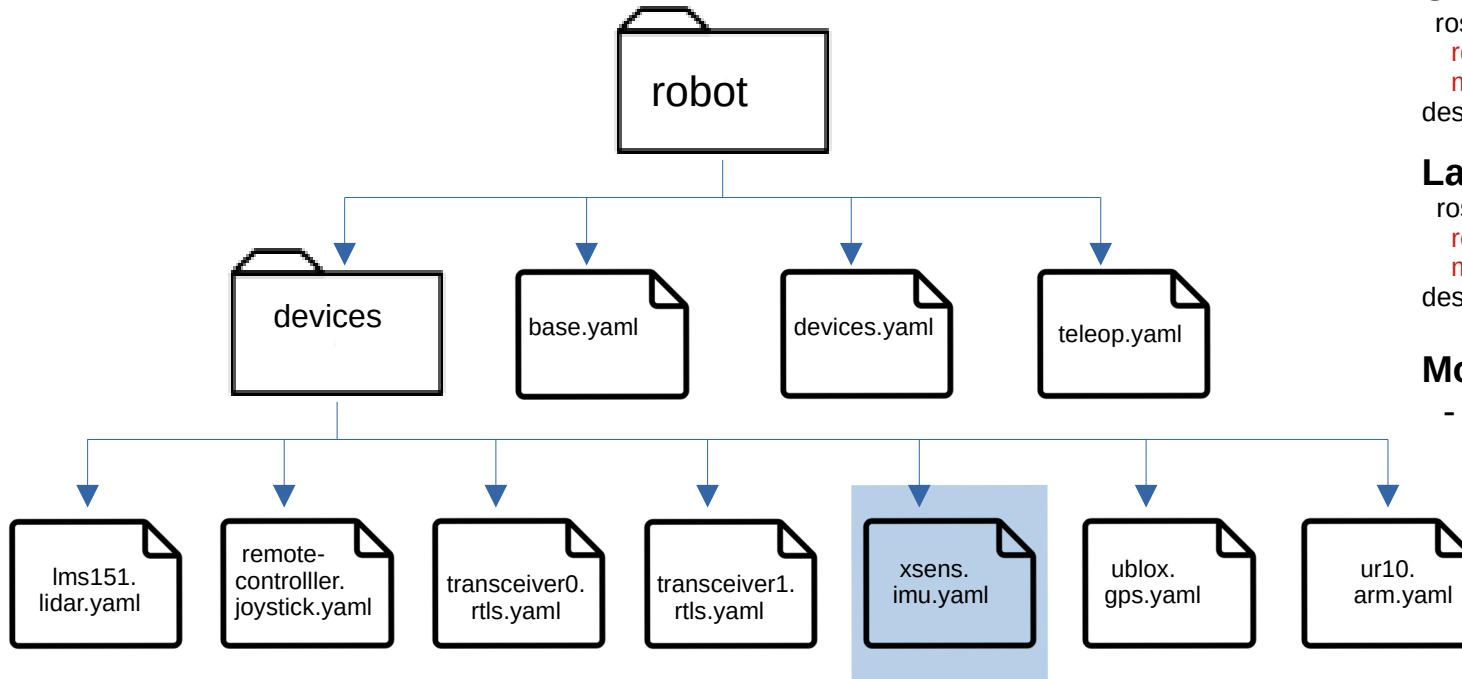
- [bluespace_ai_xsens_mti_driver](#)
- [xsens_driver](#)

Supported IMU sensors:

- xsens (models mti and mti6xx)

Robot configuration

IMU



URDF Description :

`ros2 run roma_imu_bringup urdf_description.py`
`robot_namespace:=adap2e`
`meta_description_filename:=path to imu meta-description file`

Launch :

`ros2 run roma_imu_bringup imu_driver.launch.py`
`robot_namespace:=adap2e`
`meta_description_filename:=path to imu meta-description file`

More info :

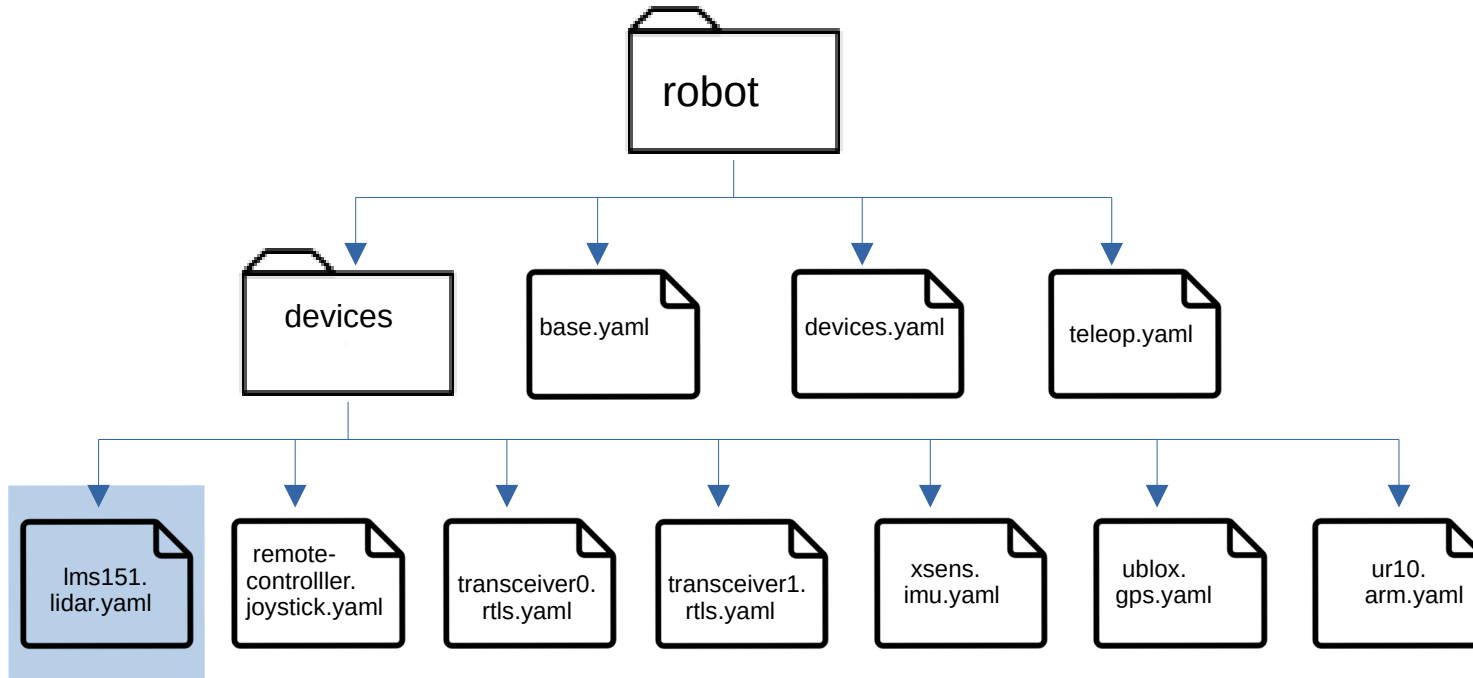
- romea_imu stack([gitlab](#),[github](#))

Examples :

`-ros2 run romea_imu_bringup urdf_description.py robot_namespace:=adap2e meta_description_filename:=path_to_demo/config/robot/devices/xsens.imu.yaml > xsens.urdf`
`-ros2 launch romea_imu_bringup imu_driver.launch.py robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/devices/xsens.imu.yaml`

Robot configuration

LIDAR



Meta-description:

name: "lidar"

driver:

pkg: "sick_scan"

ip: "192.168.1.112"

port: 2112

configuration:

type: sick

model: lms151

rate: 50 # hz

resolution: 0.5 # degree

geometry:

parent_link: "base_link"

xyz: [2.02, 0.0, 0.34] # meters

rpy: [0.0, 0.0, 0.0] # degrees

records:

scan: true

cloud: false

Supported driver packages :

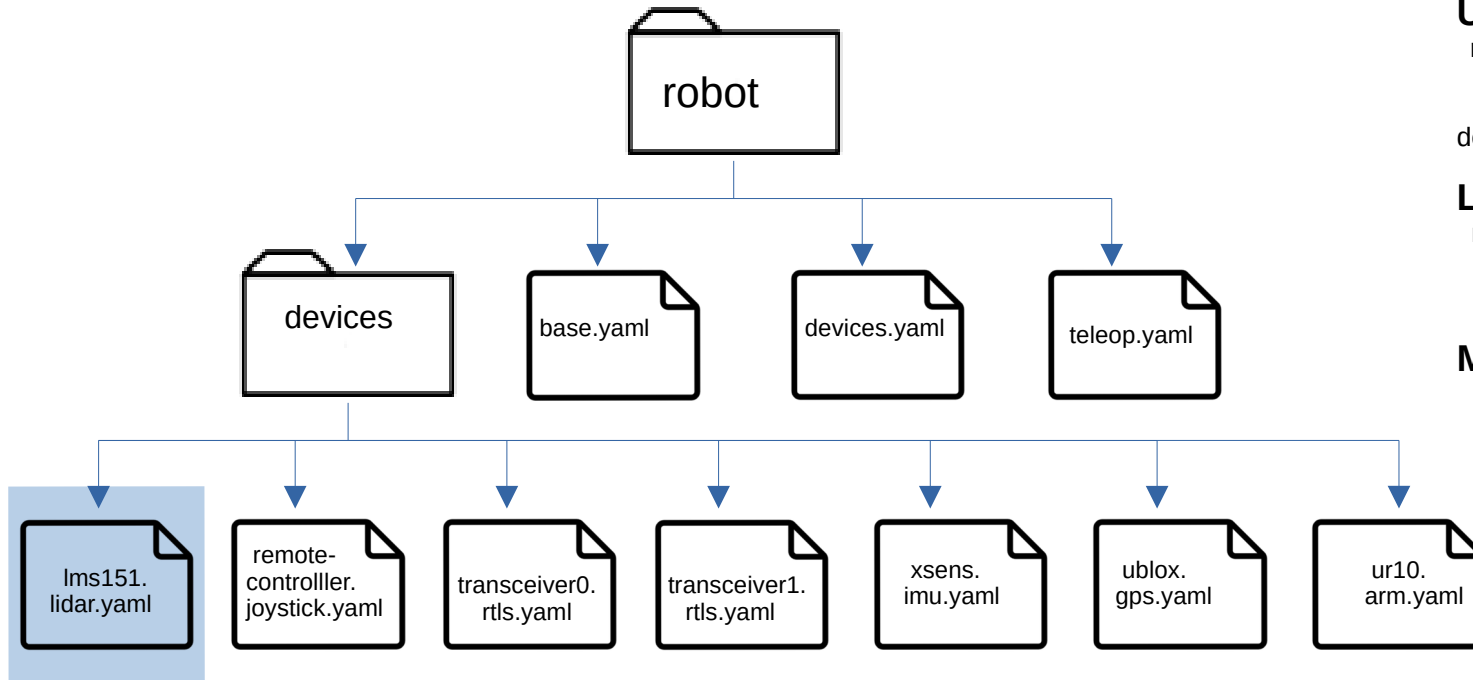
- **sick_scan**

Supported lidars :

- **sick (lms1xx,tim5xx)**

Robot configuration

LIDAR



URDF Description :

`ros2 run roma_lidar_bringup urdf_description.py`
`robot_namespace:=adap2e`
`meta_description_filename:=path to lidar meta-description file`

Launch :

`ros2 run roma_lidar_bringup lidar_driver.launch.py`
`robot_namespace:=adap2e`
`meta_description_filename:=path to lidar meta-description file`

More info :

- romea_lidar stack ([gitlab](#), [github](#))

Examples :

`-ros2 run romea_lidar_bringup urdf_description.py robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/lms151.lidar.yaml > lms151.urdf`
`-ros2 launch romea_lidar_bringup lidar_driver.launch.py robot_namespace:=adap2e meta_description_file_path:=path_to_demo/config/robot/lms151.lidar.yaml`

Robot configuration

Bridge (Campero, Alpo....)

Adap2e

Meta-description:

```
name: "lidar"
driver:
  pkg: "sick_scan"
  ip: "192.168.1.112"
  port: 2112
configuration:
  type: sick
  model: lms151
  rate: 50 # hz
  resolution: 0.5 # degree
geometry:
  parent_link: "base_link"
  xyz: [2.02, 0.0, 0.34] # meters
  rpy: [0.0, 0.0, 0.0] # degrees
records:
  scan: true
  cloud: false
```

Campero

Meta-description:

```
name: "front_lidar"
configuration:
  type: sick
  model: lms151
  rate: 50 # hz
  resolution: 0.5 # degree
geometry:
  parent_link: "base_link"
  xyz: [0.490, -0.300, 0.3513] # meters
  rpy: [180.0, 0.0, -45.0] # degrees
records:
  scan: true
bridge:
  scan: /campero_bridge/front_laser/scan
```



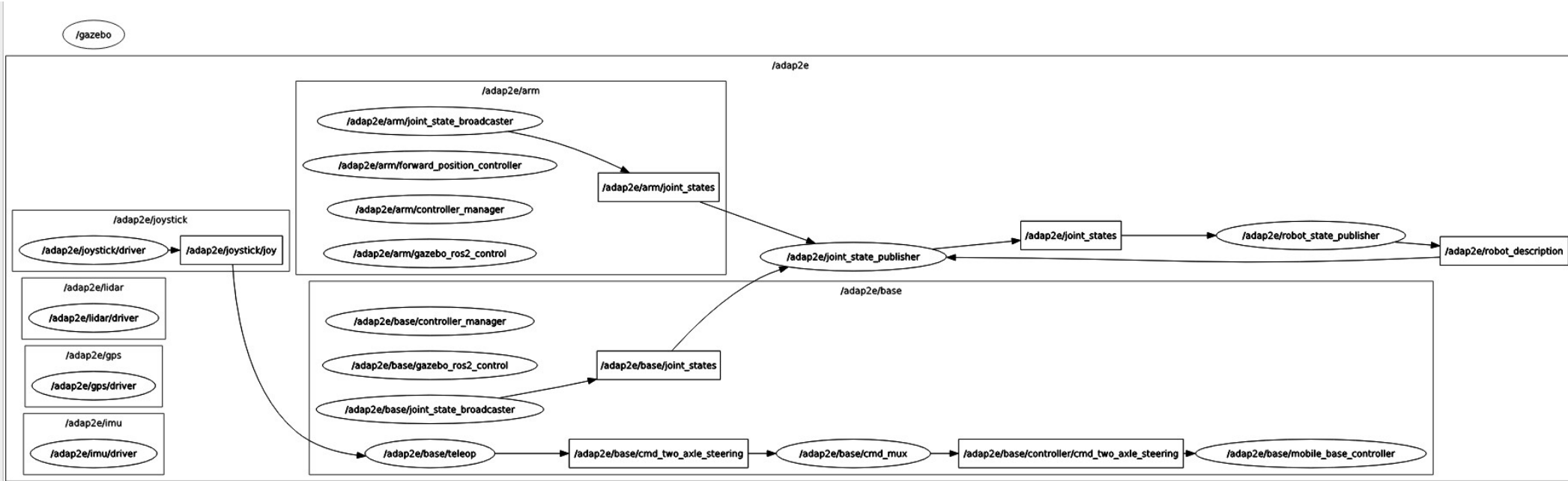
Complete remapping used by algorithms, applications...
/robot_name/front_lidar/scan : /campero_bridge/front_laser/scan



Only a driver or a bridge into a Meta-description

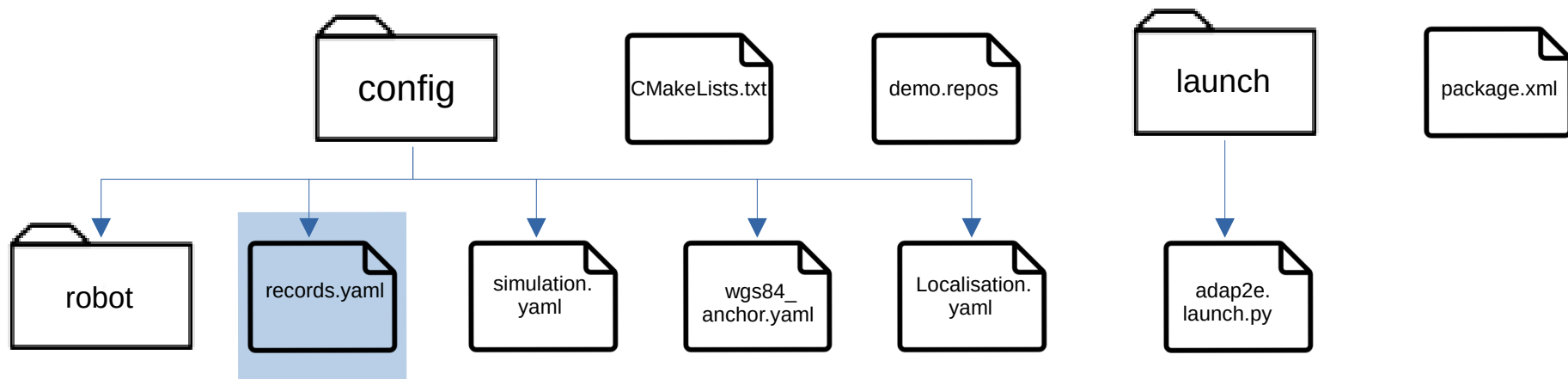
Robot Simulation

Nodes graph



Demo configuration

Record/Replay



Records configuration:

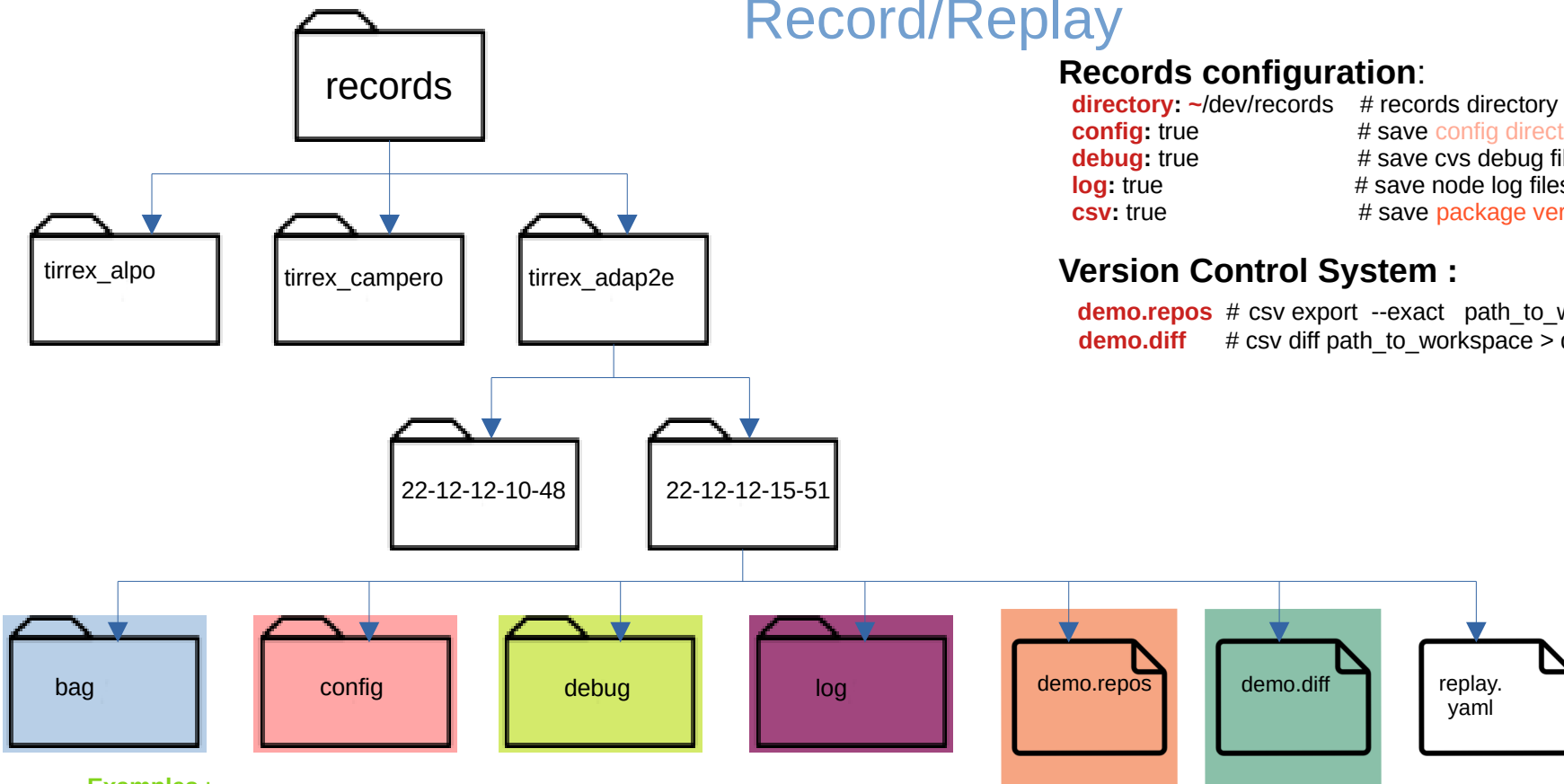
directory: ~/dev/records	# records directory
config: true	# save config directory
debug: true	# save cvs debug files into debug directory
log: true	# save node log files into log directory
csv: true	# save package versions and source codes diff

Examples :

- ros2 launch tirrex_adap2e adap2e.launch.py mode:=simulation record:= true

Demo configuration

Record/Replay



Records configuration:

directory: ~/dev/records # records directory
config: true # save config directory
debug: true # save cvs debug files into debug director
log: true # save node log files into log directory
csv: true # save package versions and source codes diff

Version Control System :

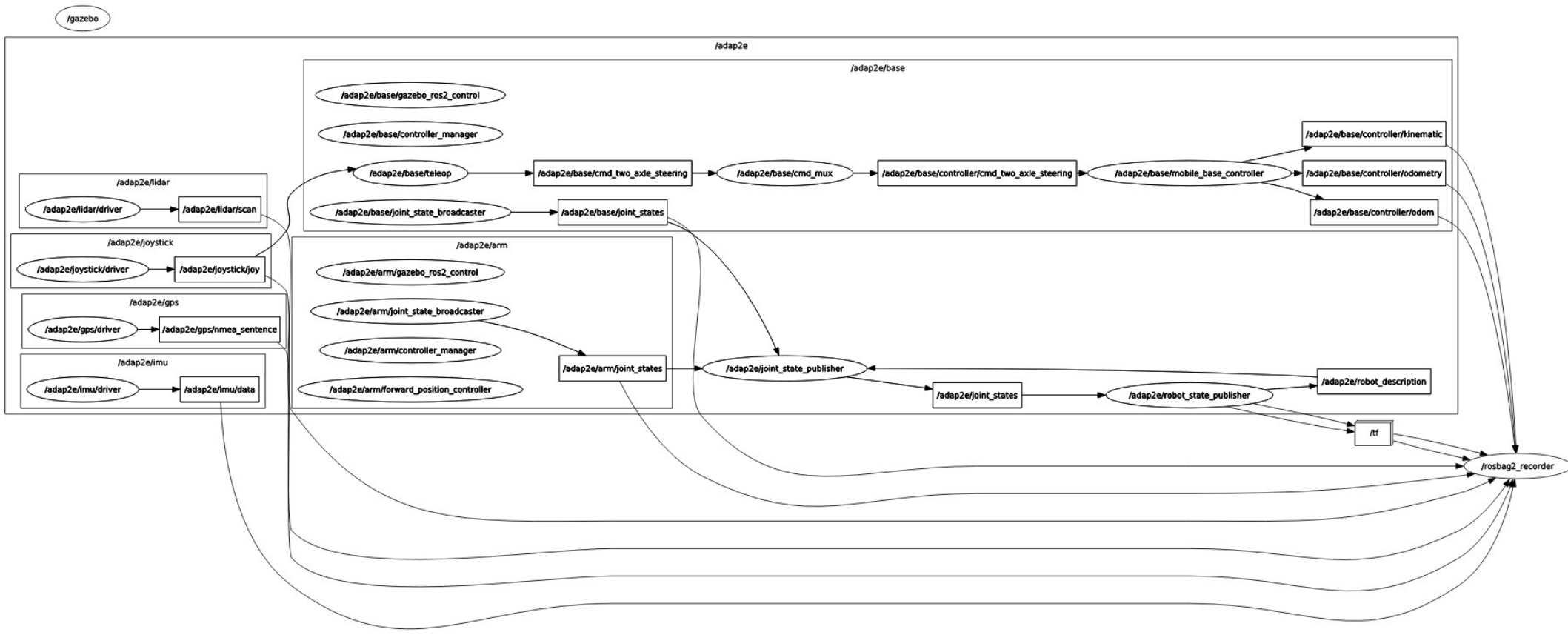
demo.repos # csv export --exact path_to_workspace > demo.repos
demo.diff # csv diff path_to_workspace > demo.diff

Examples :

- ros2 launch tirrex_adap2e adp2e.launch.py mode:=simulation record:= true
- ros2 launch tirrex_demo replay.launch.py replay_directory:=~/dev/records/tirrex_adap2e/22-12-12-15-51

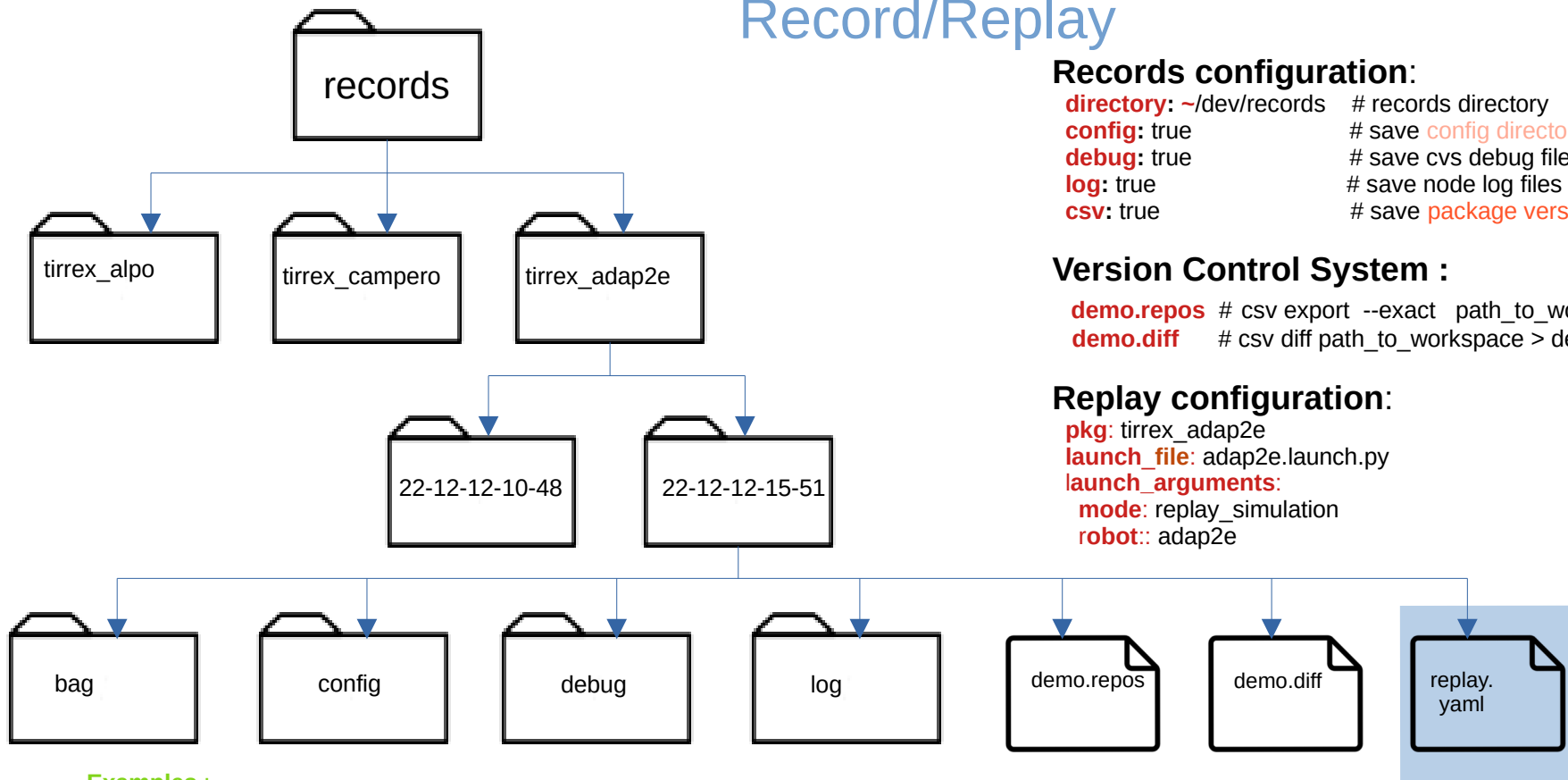
Demo configuration

Record/Replay



Demo configuration

Record/Replay



Records configuration:

directory: ~/dev/records # records directory
config: true # save config directory
debug: true # save cvs debug files into debug director
log: true # save node log files into log directory
csv: true # save package versions and source codes diff

Version Control System :

demo.repos # csv export --exact path_to_workspace > demo.repos
demo.diff # csv diff path_to_workspace > demo.diff

Replay configuration:

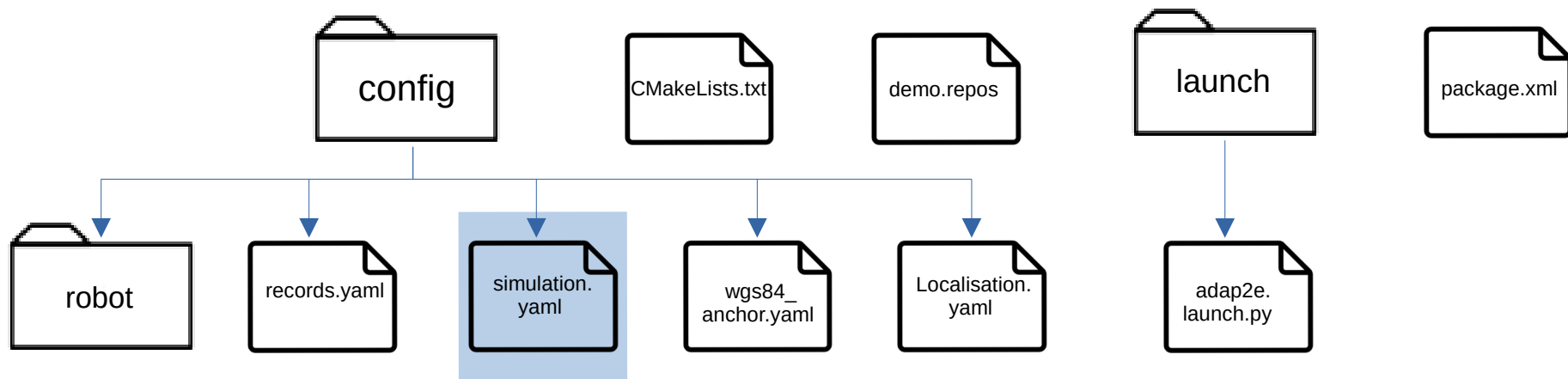
pkg: tirrex_adap2e
launch_file: adap2e.launch.py
launch_arguments:
mode: replay_simulation
robot:: adap2e

Examples :

- ros2 launch tirrex_adap2e adap2e.launch.py mode:=simulation records:= true
- ros2 launch romea_demo replay.launch.py replay_directory:~/dev/records/tirrex_adap2e/22-12-12-15-51

Demo configuration

Simulation configuration



Simulation configuration :

world_package: romea_simulation_gazebo_worlds

world_name: romea_small_vineyard.world

More Info :

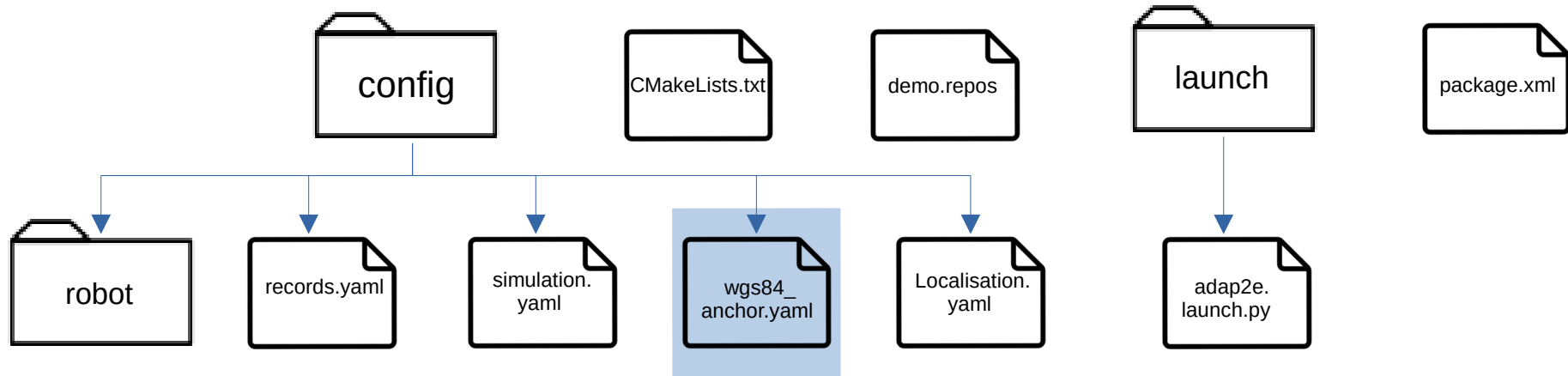
romea_simulation stack (gitlab, github)

Examples :

```
-ros2 launch romea_simulation_bringup simulator.launch.py simulation_configuration_file_path:=path_to_demo/config/simulation.yaml  
wgs84_anchor_file_path:=path_to_demo/config/wgs84_anchor.yaml simulator_type :=gazebo
```

Demo configuration

WGS84 reference point configuration

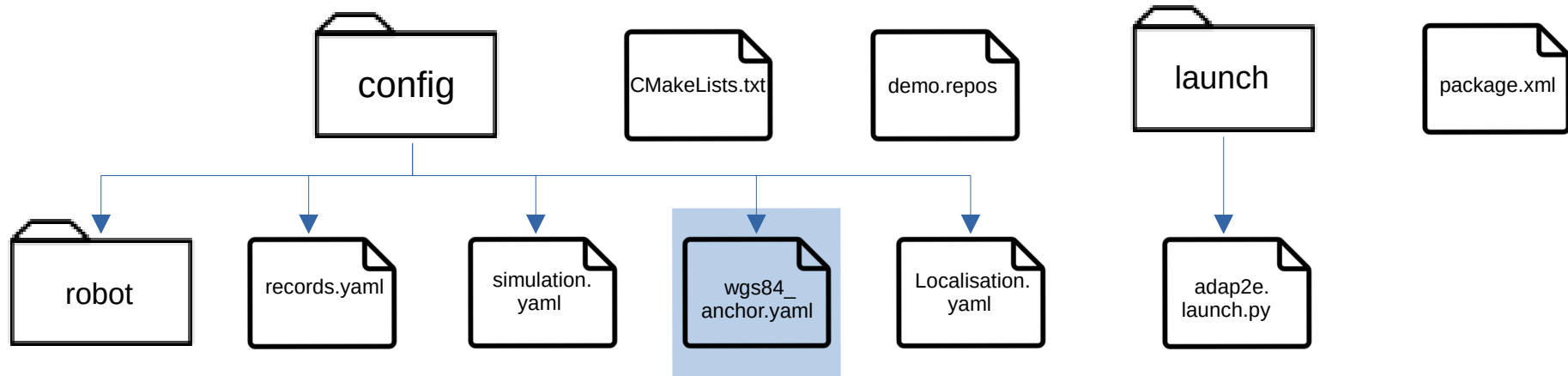


WGS84 anchor :

latitude: 45.76265802
longitude: 3.11000985
Altitude: 405.839

Demo configuration

Localisation configuration



WGS84 anchor :

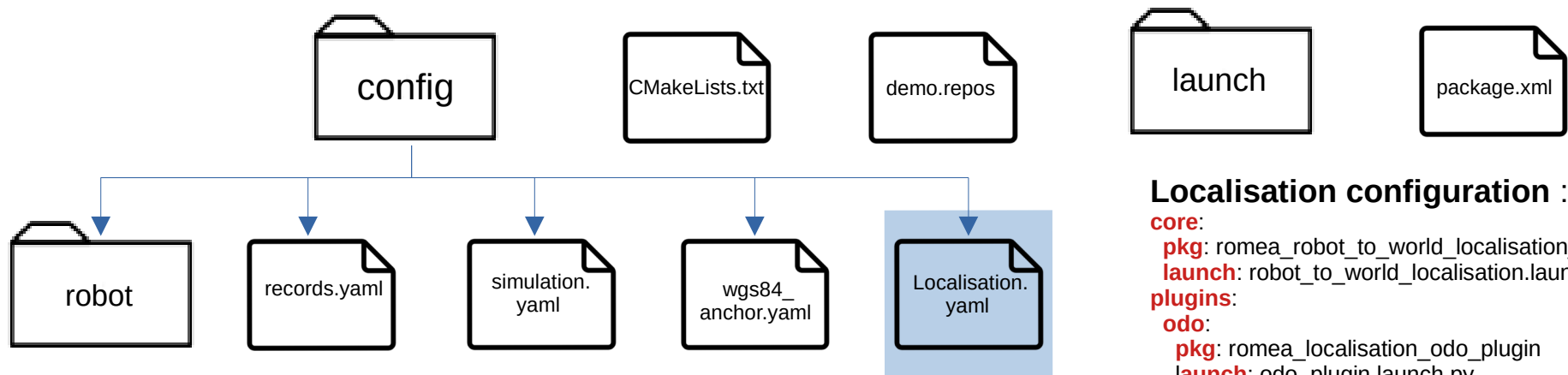
latitude: 45.76265802

longitude: 3.11000985

Altitude: 405.839

Demo configuration

Localisation configuration



Localisation configuration :

core:

pkg: romea_robot_to_world_localisation_core

launch: robot_to_world_localisation.launch.py

plugins:

odo:

pkg: romea_localisation_odo_plugin

launch: odo_plugin.launch.py

imu:

pkg: romea_localisation_imu_plugin

launch: imu_plugin.launch.py

input_sensor_meta_description_filename: xsens

gps:

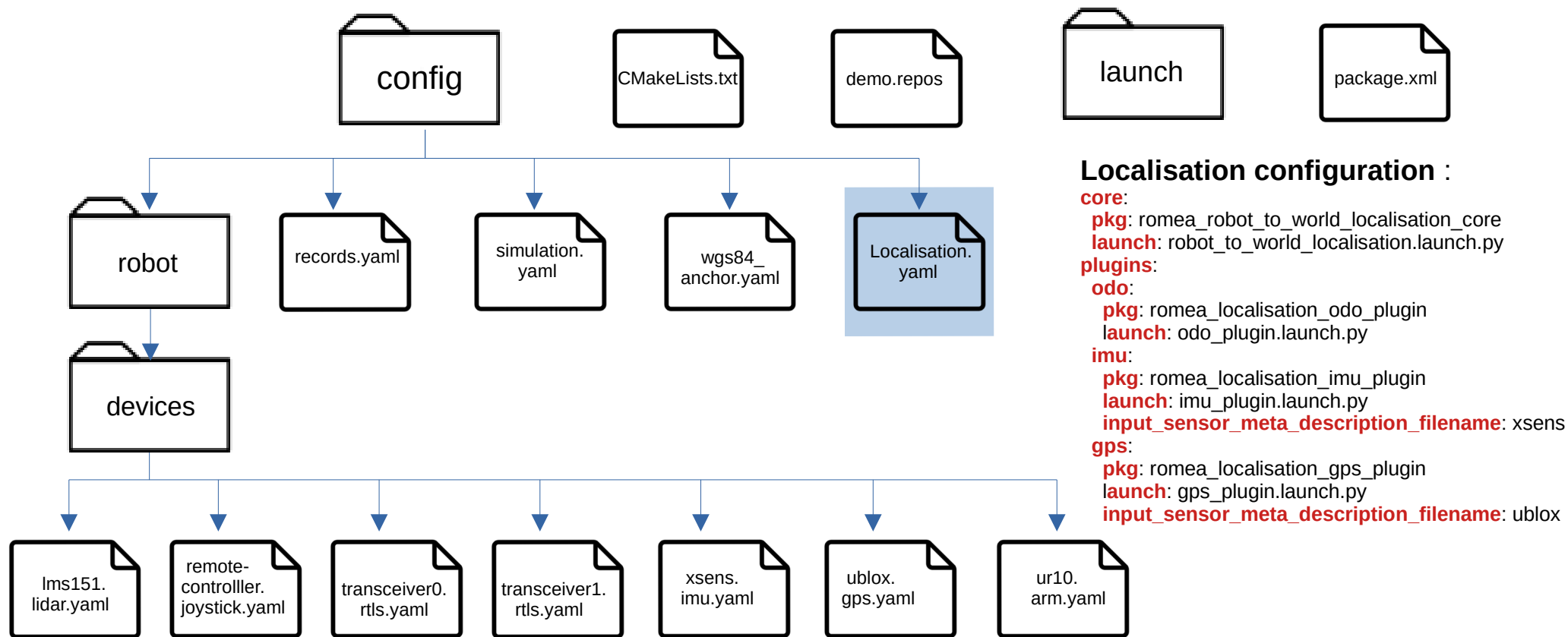
pkg: romea_localisation_gps_plugin

launch: gps_plugin.launch.py

input_sensor_meta_description_filename: ublox

Demo configuration

Localisation configuration



Demo configuration

Localisation configuration

