
Exploring Smart Vehicles in the Virtual World

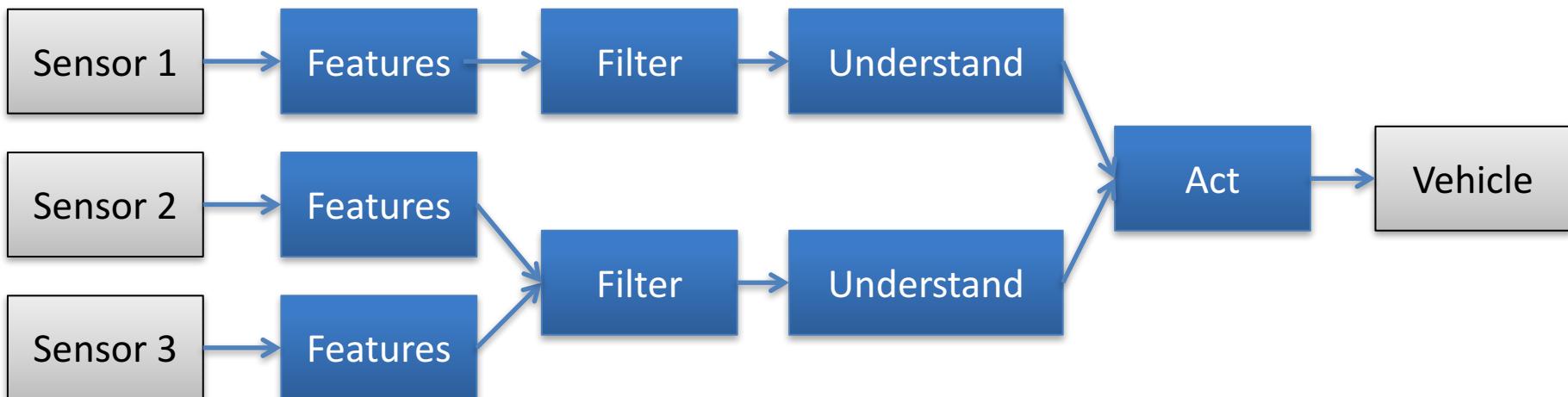
OpenDaVINCI + OpenDLV

OpenDaVINCI is a lean realtime-capable software development and runtime environment written entirely in standard C++ running on a variety of POSIX-compatible OS and hardware platforms.

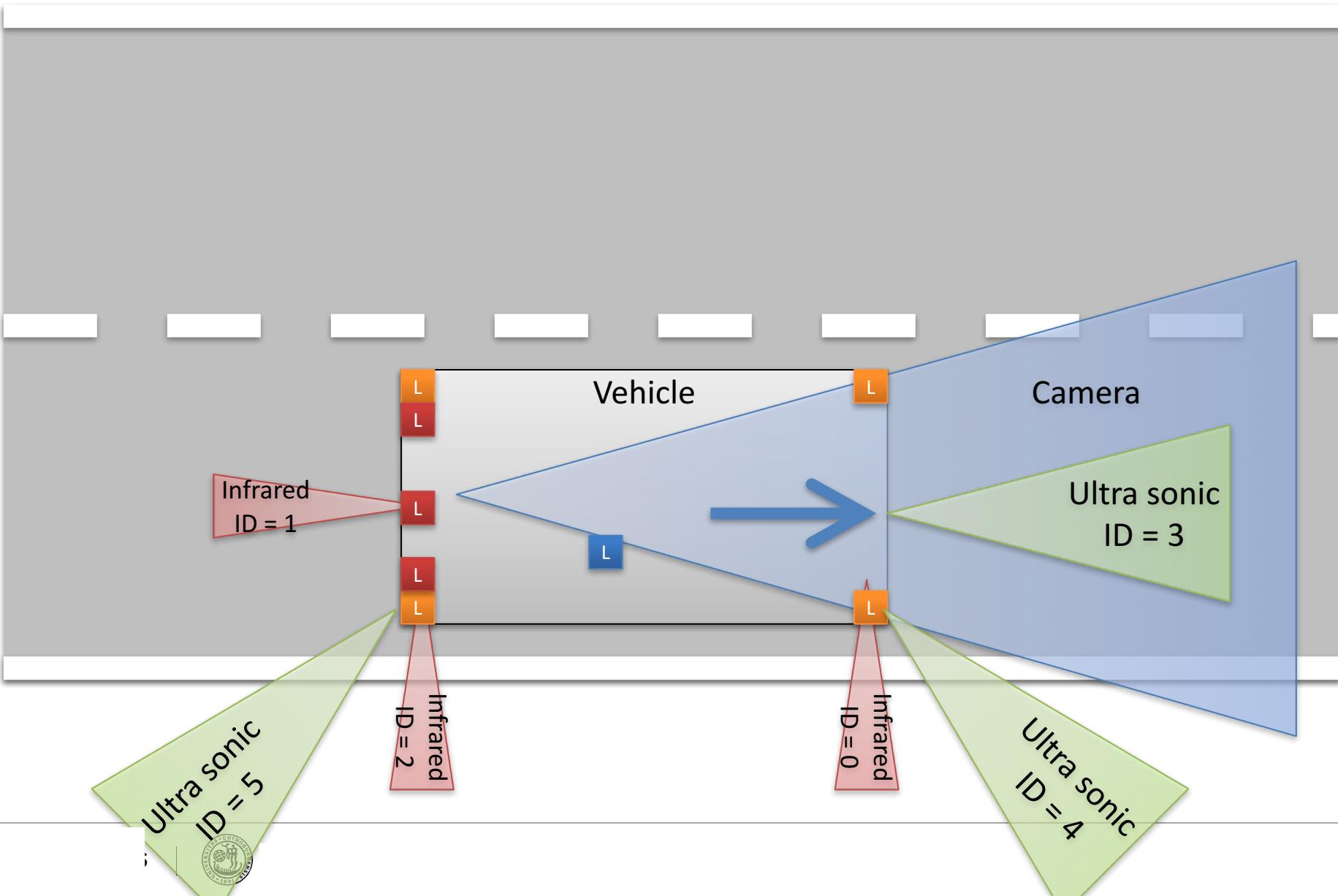
Features:

- Distributed data processing software modules in realtime environments.
- Time- or data-triggered software modules.
- Centrally maintained dynamic software module configuration protocol (DMCP).
- Publish/subscribe and directed communication.
- Transparent monitoring, logging, and replaying - including unstructured BLOBs like video streams.
- Built-in supervised execution with transparent control of communication, time, and scheduling.

OpenDLV provides low-level HW/SW interfaces to sensors and actuators and contains biologically inspired algorithms for self-driving vehicles.



Sensor Layout – Configuration



Sensor Layout – Configuration

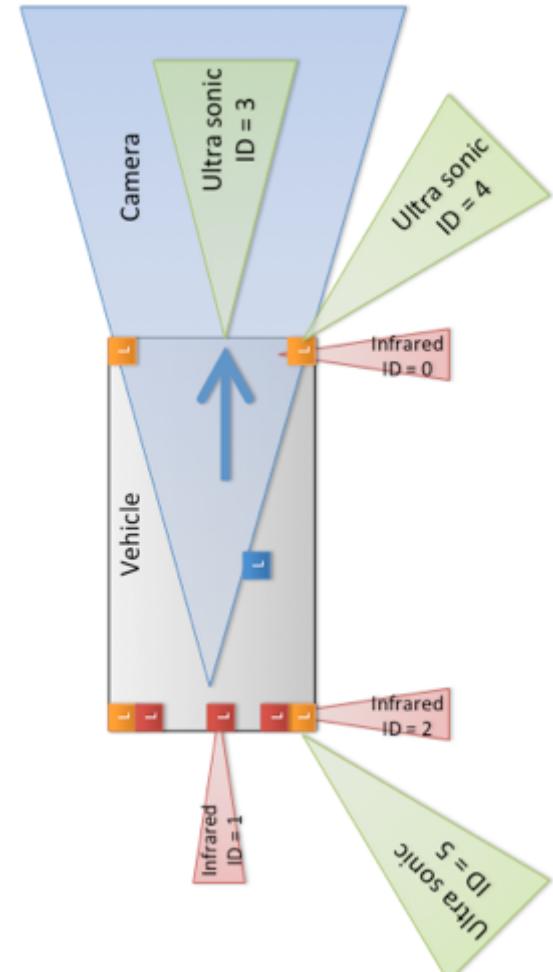
```
global.scenario = file://Parking-boxes-1.scnx
```

```
odsimirus.numberOfSensors = 6  
odsimirus.showPolygons = 1
```

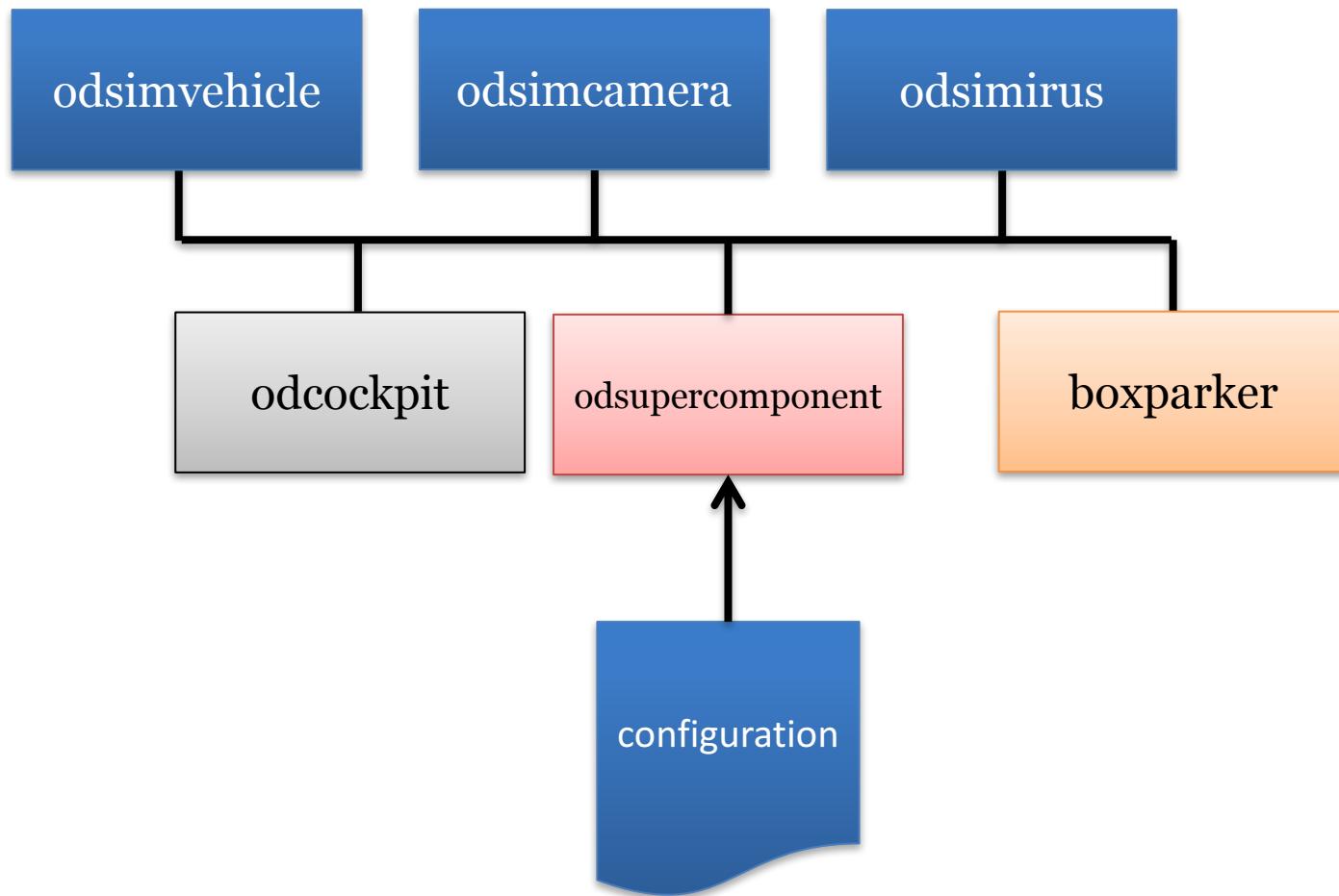
```
odsimirus.sensor0.id = 0  
odsimirus.sensor0.name = Infrared_FrontRight  
odsimirus.sensor0.rotZ = -90  
odsimirus.sensor0.translation = (1.0;-1.0;0.0)  
odsimirus.sensor0.angleFOV = 5  
odsimirus.sensor0.distanceFOV = 3  
odsimirus.sensor0.clampDistance = 2.9  
odsimirus.sensor0.showFOV = 1
```

```
odsimirus.sensor1.id = 1  
odsimirus.sensor1.name = Infrared_Rear  
odsimirus.sensor1.rotZ = -180  
odsimirus.sensor1.translation = (-1.0;0.0;0.0)  
odsimirus.sensor1.angleFOV = 5  
odsimirus.sensor1.distanceFOV = 3  
odsimirus.sensor1.clampDistance = 2.9  
odsimirus.sensor1.showFOV = 1
```

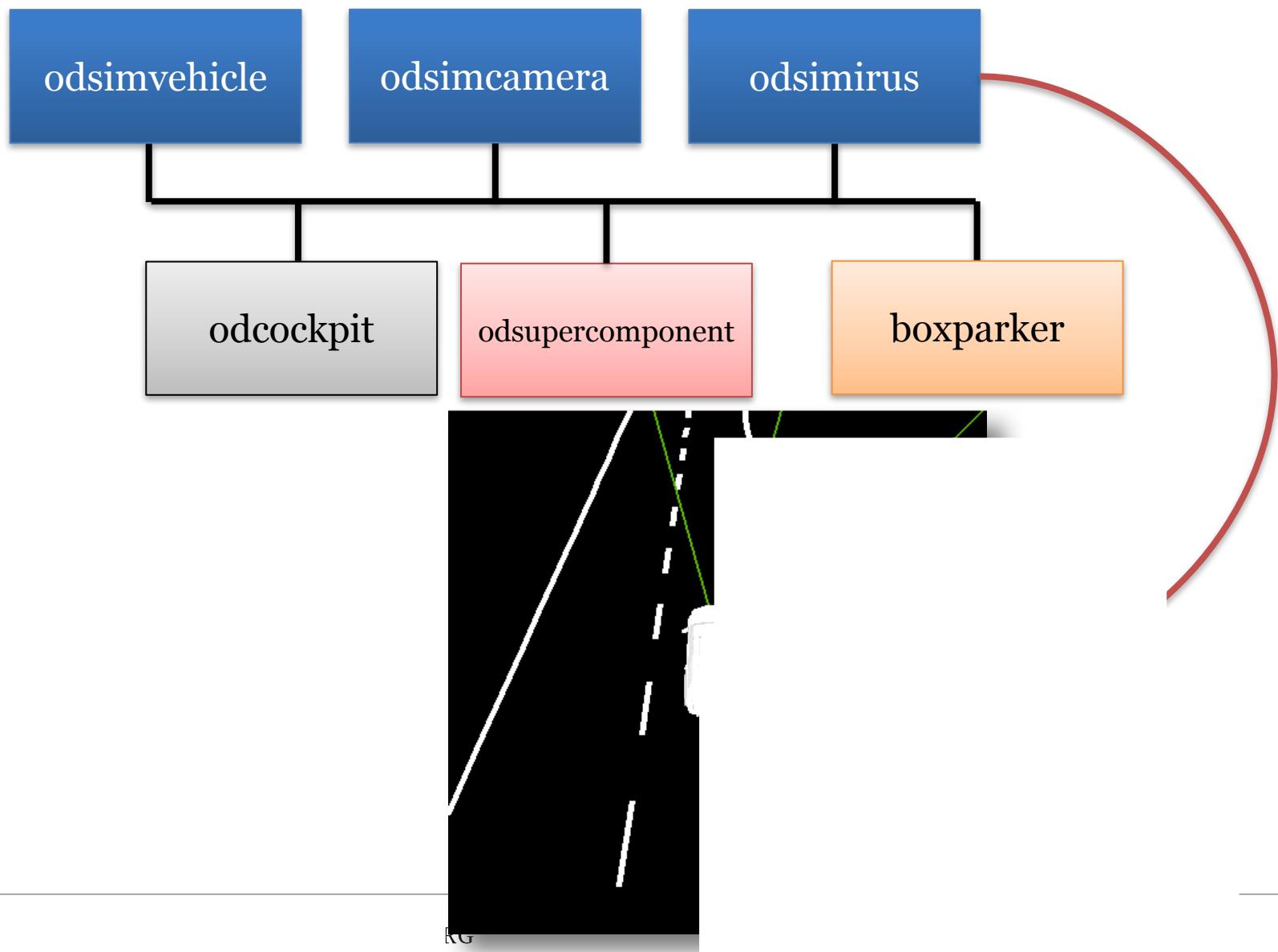
```
...
```



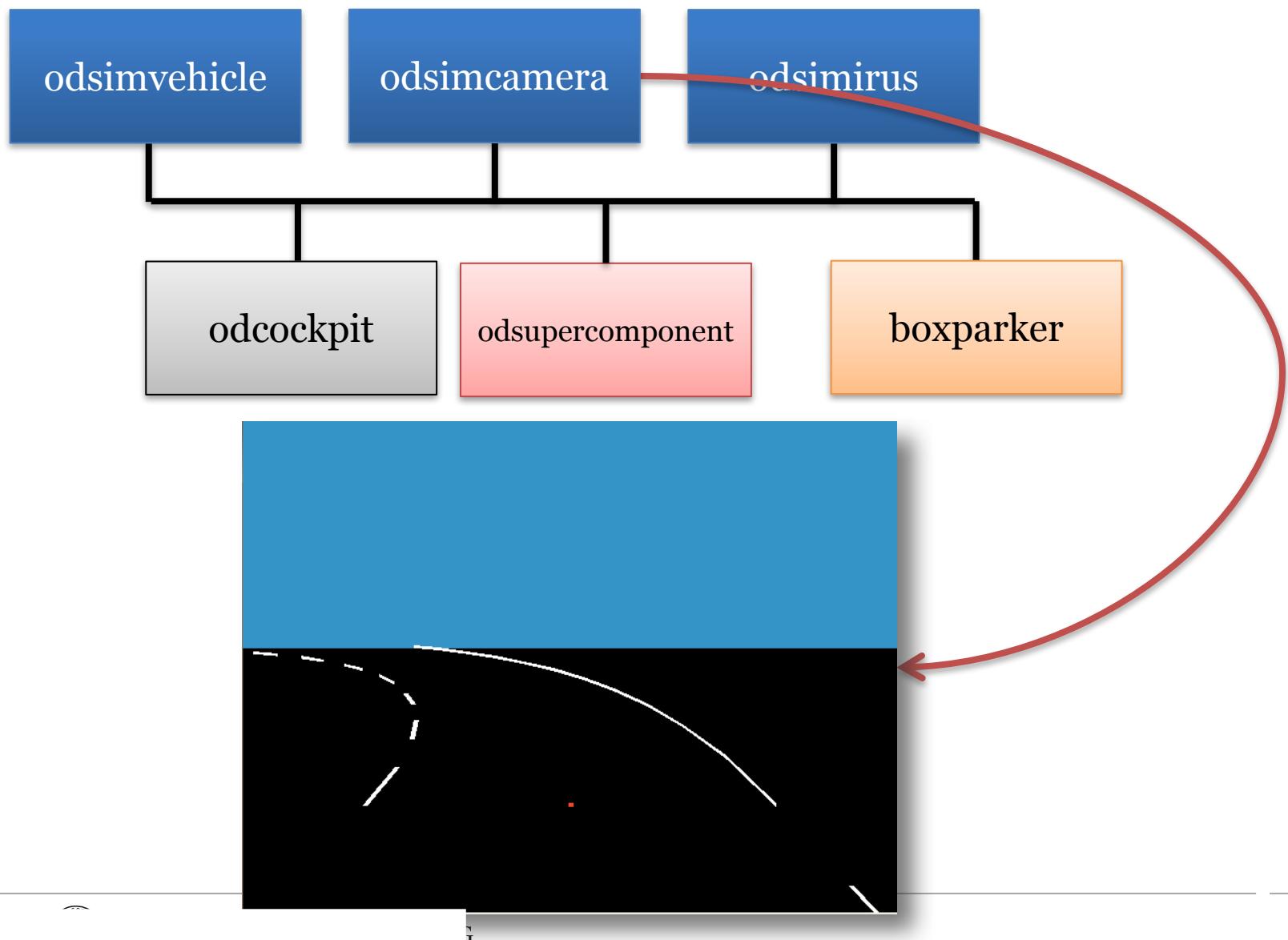
OpenDaVINCI + OpenDLV



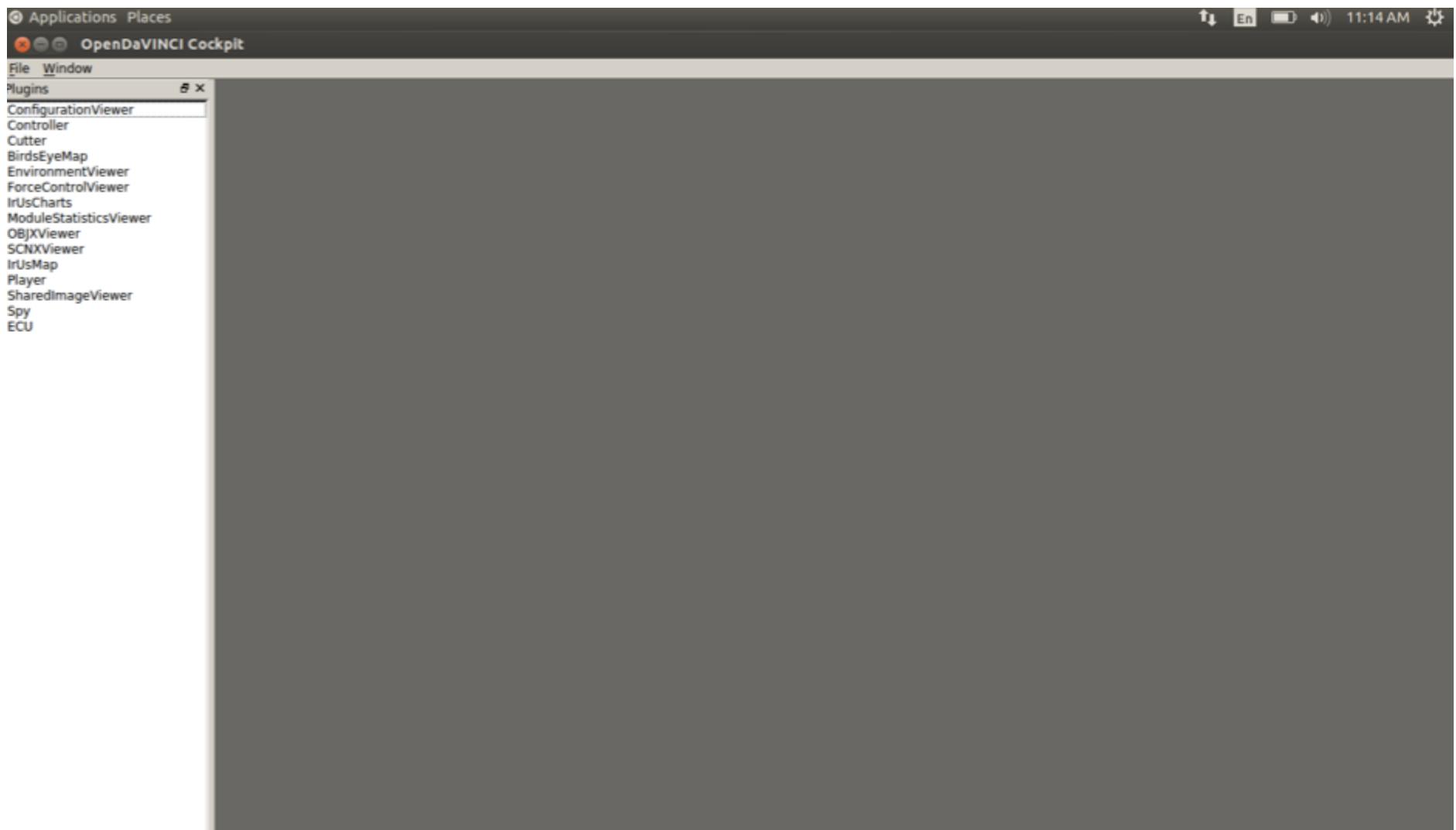
OpenDaVINCI + OpenDLV



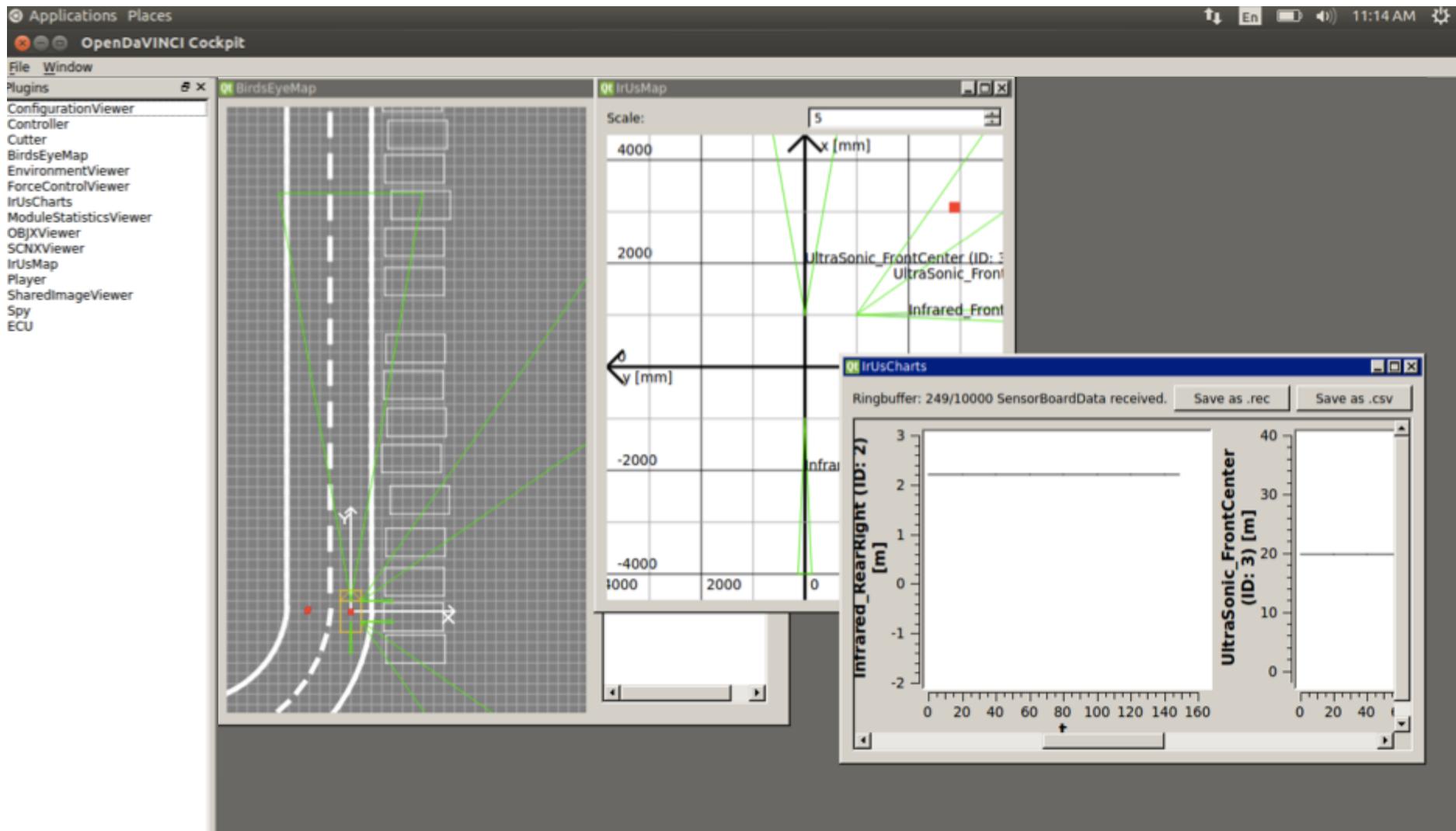
OpenDaVINCI + OpenDLV



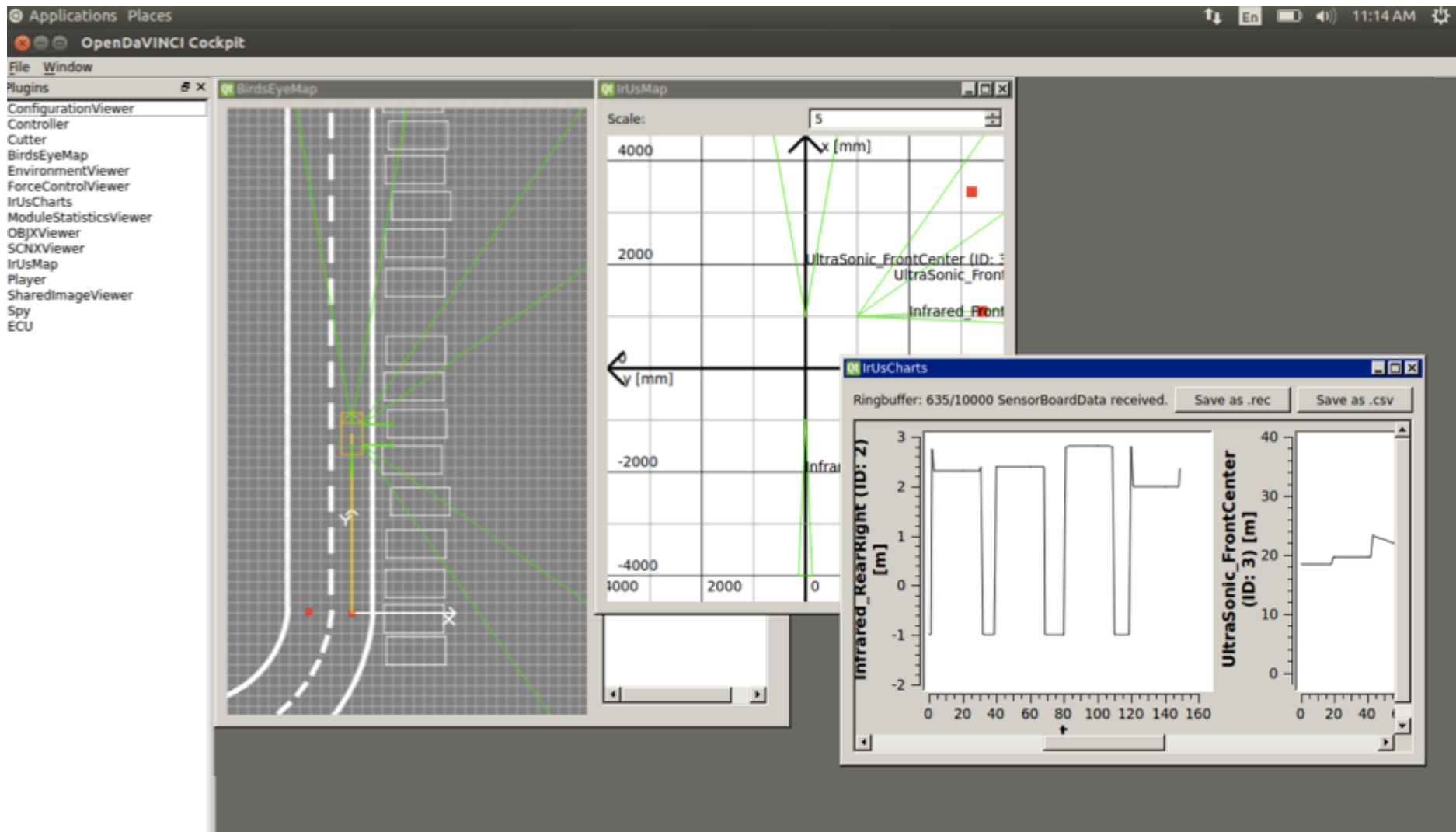
odcockpit



odcockpit



odcockpit



OpenDaVINCI + OpenDLV

Component	Simulation /Real Car	Parameters Simulation	Parameters Real Car
odsupercomponent	both	--cid=111	--cid=111
odsimirus	simulation	--cid=111 --freq=10	-
odsimcamera	simulation	--cid=111 --freq=10	-
odsimvehicle	simulation	--cid=111 --freq=10	-
odcockpit	both	--cid=111	--cid=111
Your Code	both	--cid=111 --freq=10	--cid=111 --freq=10
OpenDLV.core/proxy-camera	real car	-	--cid=111 --freq=10

Testing the Example

Downloading the scenario and configuration:

```
$ cd && wget http://www.cse.chalmers.se/~bergerc/DIT-168.zip
```

Unpack to \$HOME/DIT-168:

```
$ cd && unzip DIT-168.zip
```

Install Docker to your Linux system:

<https://docs.docker.com/engine/installation/linux>

Get our latest Docker image (**our Docker images will be updated at least weekly!**):

```
$ docker pull sereresearch/opendavinci-ubuntu-16.04-complete:latest
```

odsupercomponent

Run odsupercomponent:

```
$ docker run -ti --rm --net=host -v $HOME/DIT-168:/opt/configuration -w /opt/configuration sereresearch/opendavinci-ubuntu-16.04-complete:latest /opt/od4/bin/odsupercomponent --cid=111 --verbose=1
```

```
Creating multicast UDP receiver at  
225.0.0.111:19751.(supercomponent) Parsing configuration  
file...(supercomponent) Server information: IP: 0.0.0.0,  
Port: 19866, managedLevel: 5  
(supercomponent) Creating discoverer server...  
Creating multicast UDP receiver at  
225.0.0.111:19750.(supercomponent) Creating connection  
server...  
Creating multicast UDP receiver at  
225.0.0.111:12175.(supercomponent) Ready - managed level 0
```

odsimvehicle

Run odsimvehicle:

```
$ docker run -ti --rm --net=host seresearch/opendavinci-ubuntu-16.04-complete:latest /opt/od4/bin/odsimvehicle --cid=111 --freq=10
```

```
No runtime frequency set. Assuming a frequency of 1 Hz.  
Creating multicast UDP receiver at 225.0.0.111:12175.  
Creating multicast UDP receiver at 225.0.0.111:19751.  
(ClientModule) discovering supercomponent...  
(ClientModule) supercomponent found at IP: 10.0.2.15, Port: 19866, managedLevel: 5  
(ClientModule) connecting to supercomponent...  
(DMCP-ConnectionClient) sending configuration request...IP: 10.0.2.15, Port: 19866, managedLevel: 5  
(DMCP-Client) Received Configuration  
global.buffer.memorysegmentsize=2800000  
global.buffer.numberofmemorysegments=20  
global.car=Scenarios/Models/FordEscape.objx  
global.scenario=file://Parking-boxes-1.scnx  
global.showgrid=0  
odsimvehicle.headingdeg=90  
...
```

odsimirus

Run odsimirus:

```
$ docker run -ti --rm --net=host -v $HOME/DIT-  
168:/opt/configuration -w /opt/configuration  
seresearch/opendavinci-ubuntu-16.04-complete:latest  
/opt/od4/bin/odsimirus --cid=111 --freq=10
```

```
Creating multicast UDP receiver at 225.0.0.111:12175.  
Creating multicast UDP receiver at 225.0.0.111:19751.  
(ClientModule) discovering supercomponent...  
(ClientModule) supercomponent found at IP: 10.0.2.15, Port:  
19866, managedLevel: 5  
(ClientModule) connecting to supercomponent...  
(DMCP-ConnectionClient) sending configuration request...IP:  
10.0.2.15, Port: 19866, managedLevel: 5  
(DMCP-Client) Received Configuration  
global.buffer.memorysegmentsize=2800000  
global.buffer.numberofmemorysegments=20  
global.car=Scenarios/Models/FordEscape.objx  
global.scenario=file://Parking-boxes-1.scnx  
global.showgrid=0  
odsimirus.numberofsensors=6  
...
```

odsimirus + noise data

Adjusting configuration for odsimirus to include noise:

```
odsimirus.numberOfSensors = 6
odsimirus.showPolygons = 1

odsimirus.sensor0.id = 0
odsimirus.sensor0.name = Infrared_FrontRight
odsimirus.sensor0.rotZ = -90
odsimirus.sensor0.translation = (1.0;-1.0;0.0)
odsimirus.sensor0.angleFOV = 5
odsimirus.sensor0.distanceFOV = 3
odsimirus.sensor0.clampDistance = 2.9
odsimirus.sensor0.showFOV = 1
odsimirus.sensor0.showFOV = 1
```

odsimirus + noise data

Adjusting configuration for odsimirus to include noise:

```
odsimirus.numberOfSensors = 6
odsimirus.showPolygons = 1

odsimirus.sensor0.id = 0
odsimirus.sensor0.name = Infrared_FrontRight
odsimirus.sensor0.rotZ = -90
odsimirus.sensor0.translation = (1.0;-1.0;0.0)
odsimirus.sensor0.angleFOV = 5
odsimirus.sensor0.distanceFOV = 3
odsimirus.sensor0.clampDistance = 2.9
odsimirus.sensor0.showFOV = 1
odsimirus.sensor0.showFOV = 1
odsimirus.sensor0.faultModel.skip = 0.1
odsimirus.sensor0.faultModel.noise = 0.2
```

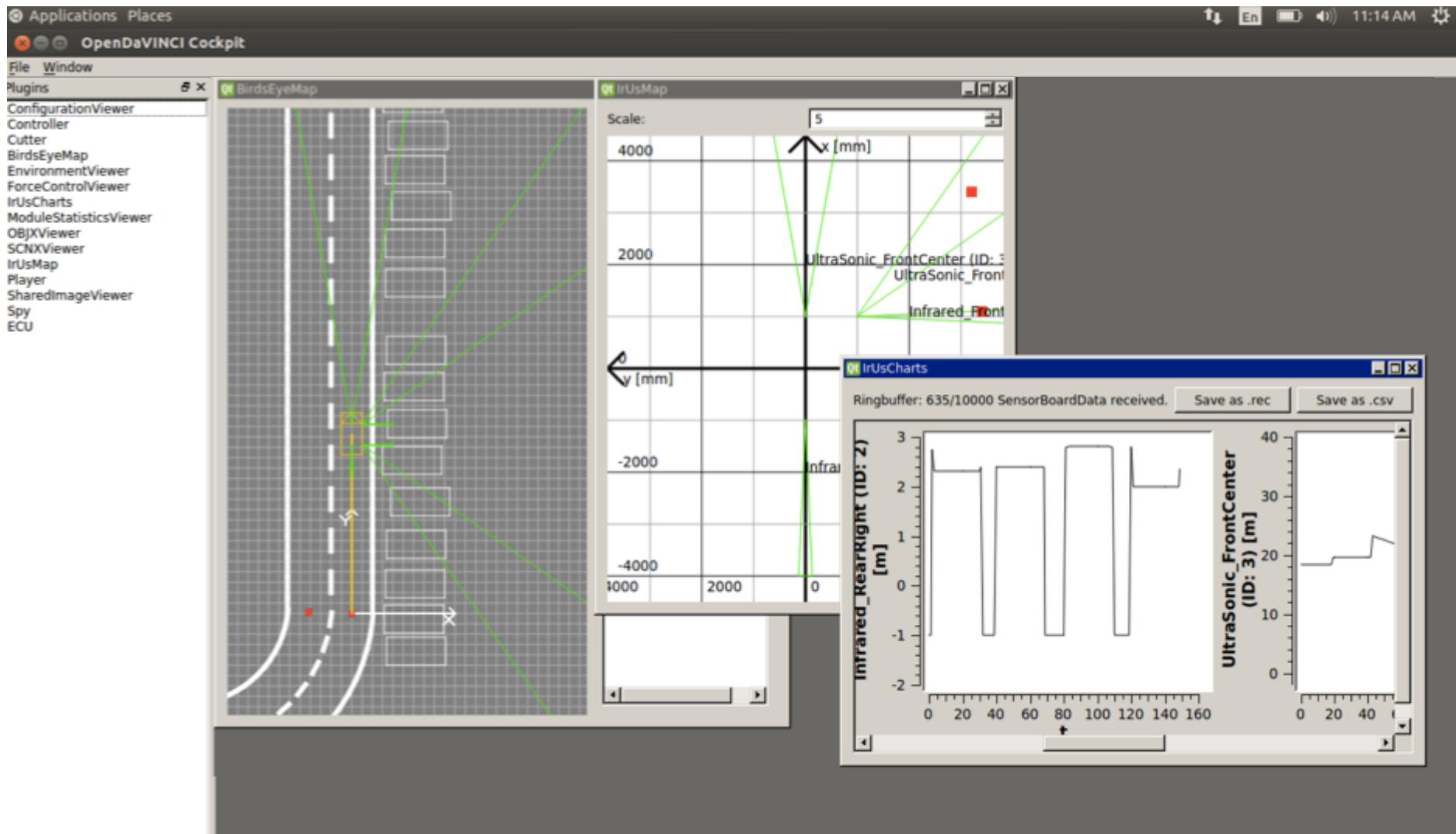
odcockpit

Run odcockpit:

```
$ xhost +  
$ docker run -ti --rm --net=host --ipc=host -e DISPLAY=$DISPLAY -v  
/tmp/.X11-unix:/tmp/.X11-unix -v $HOME/DIT-168:/opt/configuration -w  
/opt/configuration seresearch/opendavinci-ubuntu-16.04-complete:latest  
/opt/od4/bin/odcockpit --cid=111
```

```
No runtime frequency set. Assuming a frequency of 1 Hz.  
Creating multicast UDP receiver at 225.0.0.111:12175.  
Creating multicast UDP receiver at 225.0.0.111:19751.  
(ClientModule) discovering supercomponent...  
(ClientModule) supercomponent found at IP: 10.0.2.15, Port: 19866,  
managedLevel: 5  
(ClientModule) connecting to supercomponent...  
(DMCP-ConnectionClient) sending configuration request...IP: 10.0.2.15,  
Port: 19866, managedLevel: 5  
(DMCP-Client) Received Configuration  
global.buffer.memorysegmentsize=2800000  
global.buffer.numberofmemorysegments=20  
global.car=Scenarios/Models/FordEscape.objx  
global.scenario=file://Parking-boxes-1.scnx  
global.showgrid=0  
...
```

odcockpit



boxparker

Run boxparker example:

```
$ docker run -ti --rm --net=host seresearch/opendavinci-ubuntu-16.04-complete:latest
/opt/od4/bin/miniature/boxparker --cid=111 --freq=10

Creating multicast UDP receiver at 225.0.0.111:12175.
Creating multicast UDP receiver at 225.0.0.111:19751.(ClientModule) discovering supercomponent...
(ClientModule) supercomponent found at IP: 10.0.2.15, Port: 19866, managedLevel: 5
(ClientModule) connecting to supercomponent...
(DMCP-ConnectionClient) sending configuration request...IP: 10.0.2.15, Port: 19866, managedLevel: 5
(DMCP-Client) Received Configuration
global.buffer.memorysegmentsize=2800000
global.buffer.numberofmemorysegments=20
global.car=Scenarios/Models/FordEscape.objx
global.scenario=file://Parking-boxes-1.scnxglobal.showgrid=0
(ClientModule) connecting to supercomponent...done - managed level: 5
Existing ContainerConference disabled and ContainerListener redirected to localContainerConference.
Size = 0.403339
Size = 0.806678
Size = 1.21002
Size = 1.00835
Size = 0.504174
Size = 0.705844
Size = 0.605009
Size = 3.52922
Calling disposal service...
Cleaning up regular removal queue... done.
Cleaning up final removal queue... done....
```

odsimcamera

Run od़simcamera (if needed):

```
$ xhost +  
$ docker run -ti --rm --net=host --ipc=host -e DISPLAY=$DISPLAY -v  
/tmp/.X11-unix:/tmp/.X11-unix -v $HOME/DIT-168:/opt/configuration -w  
/opt/configuration seresearch/opendavinci-ubuntu-16.04-complete:latest  
/opt/od4/bin/odsimcamera --freq=10 --cid=111
```

```
Creating multicast UDP receiver at 225.0.0.111:12175.  
Creating multicast UDP receiver at 225.0.0.111:19751.  
(ClientModule) discovering supercomponent...  
(ClientModule) supercomponent found at IP: 10.0.2.15, Port: 19866,  
managedLevel: 5  
(ClientModule) connecting to supercomponent...  
(DMCP-ConnectionClient) sending configuration request...IP: 10.0.2.15,  
Port: 19866, managedLevel: 5  
(DMCP-Client) Received Configuration  
global.buffer.memorysegmentsize=2800000  
global.buffer.numberofmemorysegments=20  
global.car=Scenarios/Models/FordEscape.objx  
global.scenario=file://Parking-boxes-1.scnx  
global.showgrid=0  
odsimirus.numberofsensors=6  
...
```

Under the Hood

Reading odometer data:

```
// 1. Get most recent vehicle data:  
Container containerVehicleData =  
getKeyValueDataStore().get(automotive::VehicleData::ID());  
VehicleData vd = containerVehicleData.getData<VehicleData>();
```

Interface:

```
double getHeading() const;  
double getAbsTraveledPath() const;  
double getRelTraveledPath() const;  
double getSpeed() const;
```

Source:

<https://github.com/se-research/OpenDaVINCI/blob/master/AutomotiveData.odvd#L67>

Under the Hood

Reading sensor data:

```
// 2. Get most recent sensor board data:  
Container containerSensorBoardData =  
getKeyValueDataStore().get(automotive::miniature::SensorBoardData::ID());  
SensorBoardData sbd = containerSensorBoardData.getData<SensorBoardData>()  
();
```

Interface:

```
uint32_t getNumberOfSensors() const;  
bool containsKey_MapOfDistances(const uint32_t &key) const;  
double getValueForKey_MapOfDistances(const uint32_t &key);
```

Source:

<https://github.com/se-research/OpenDaVINCI/blob/master/AutomotiveData.odvd#L108>

Under the Hood

Writing vehicle control:

```
VehicleControl vc;
```

```
...
```

```
// Interface:
```

```
vc.setSpeed(-2); // Positive → forward, negative → backward
```

```
vc.setSteeringWheelAngle(25); // Positive → to the right, negative → to the left
```

```
// Create container for finally sending the data.
```

```
Container c(vc);
```

```
// Send container.
```

```
getConference().send(c);
```

Source:

<https://github.com/se-research/OpenDaVINCI/blob/master/AutomotiveData.odvd#L80>

Thank you.



More material:

C. Berger, "*From a Competition for Self-Driving Miniature Cars to a Standardized Experimental Platform: Concept, Models, Architecture, and Evaluation*," in [Journal of Software Engineering for Robotics, vol. 5, no. 1, 2014, pp. 63-79 \(arXiv: 1406.7768\)](#).