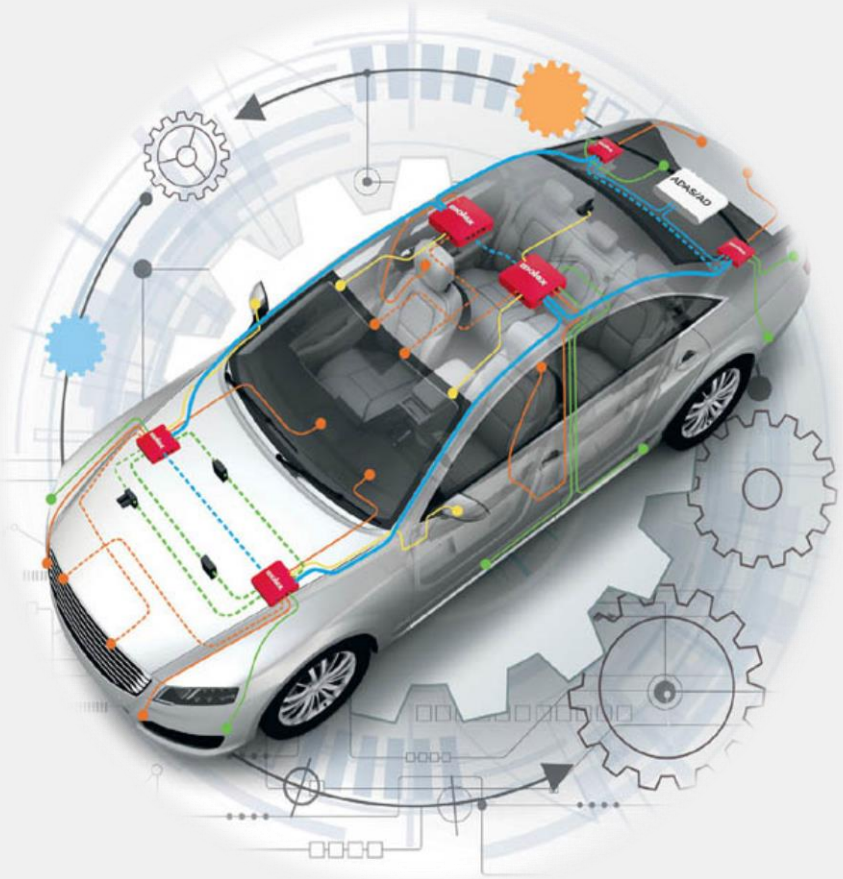
Control System of Self Driving Cars

Simulation in ROS
(Robot Operating System)

Learning Objectives

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **Challenges in Robotics**
- **Solution to the Challenges**
- **ROS (Robot Operating System)**
  - Definition
  - Why and Why not ROS?
  - Main Features
  - ROS Philosophy
  - Applications Using ROS
  - ROS Components & Concepts
  - Supported Operating Systems
- **ROS Concepts**
  - ROS Nodes
  - ROS Services
  - ROS Packages
- **ROS in Self Driving Cars**
- **Summary**
- **References**



Challenges in Robotics

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

Integration of multiple components is very complex.

Solution is ROS

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **ROS Provides:**
  - **Communication Infrastructure**
  - **Robot Specific Features**
  - **Elimination of Programming Language Barrier**
  - **Diagnostic Tools**
  - **Advance Simulation Capabilities**

ROS (Robot Operating System)

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- ROS is meta operating system
    - Collection of Frameworks, SDKs, and software tools.
- Launched in 2008 by Willow Garage.
- Almost 9th version released.
- Maintained by OSRF (Open Source Robotics Foundation)

ROS (Robot Operating System)

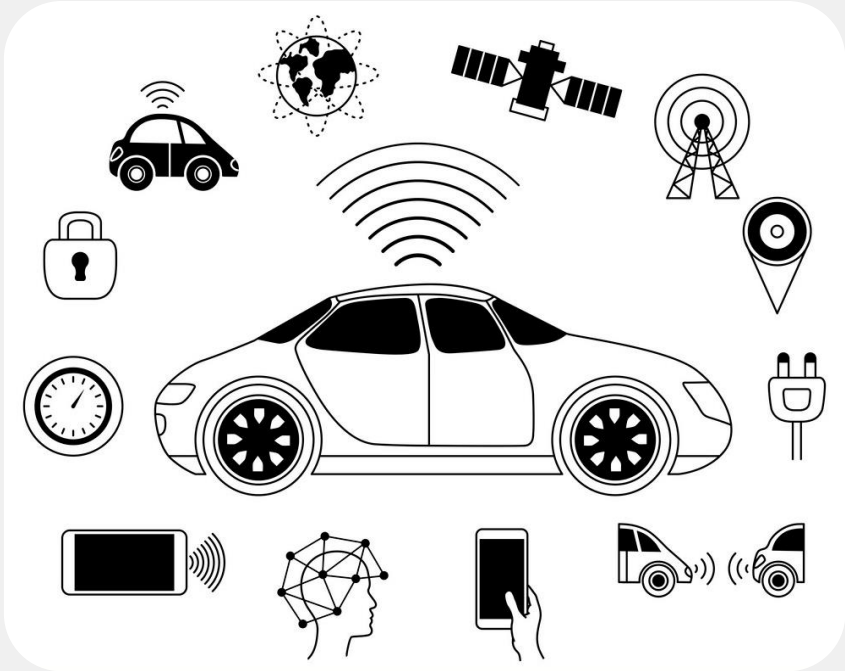
AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- ROS is a set of software libraries and tools that help you build robot applications.
- From drivers to state-of-the-art algorithms, and with powerful developer tools.
- ROS has what you need for your next robotics project.

ROS – Without ROS

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

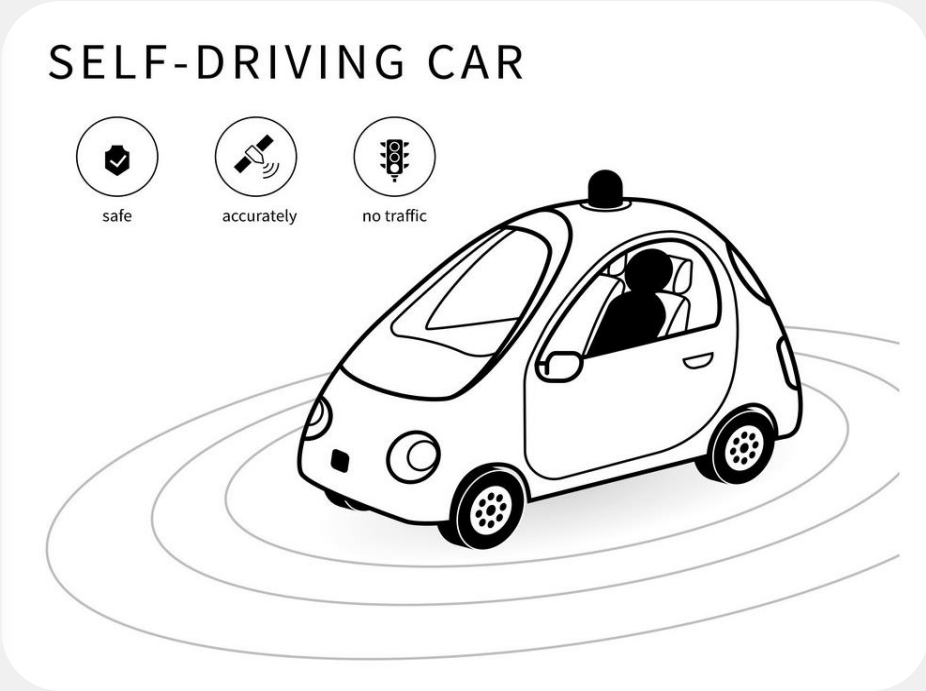
- Build device drivers
- Build a communication framework
- Write algorithms for perception, navigation, and motion planning
- Implementing logging, control, & error handling



ROS – With ROS

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

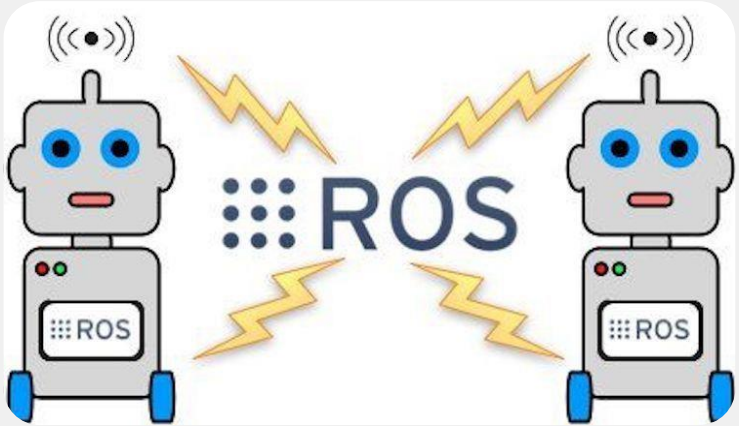
- **Logging, error handling, communication framework, drivers for standard devices**
- **Algorithms for perception, navigation, & motion planning**
- **Tools for visualization, simulation & analysis**



ROS – Main Features

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

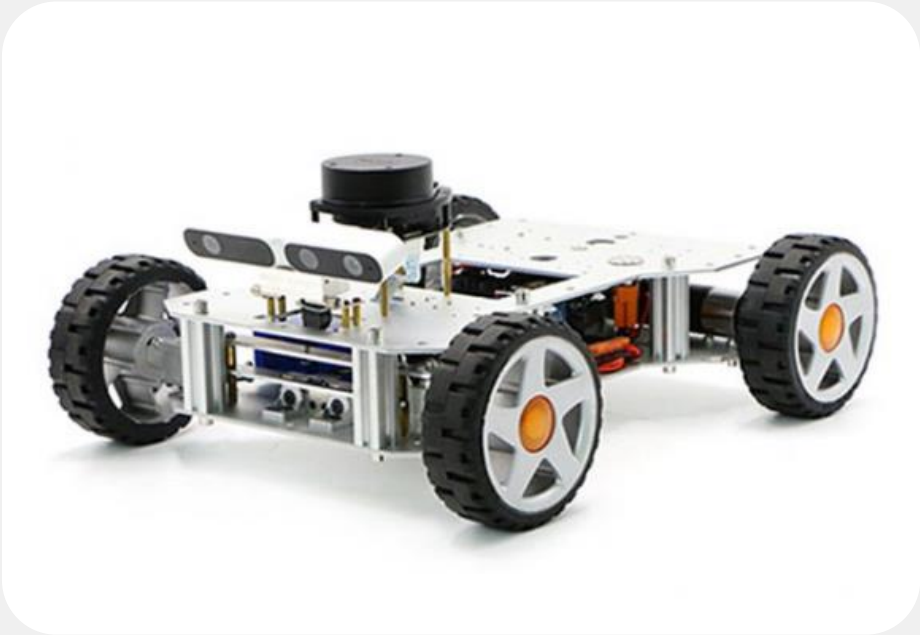
- **Operating System Side**
  - **Provides standard OS services.**
- **Suite of User Contributed Packages**
  - **Implement common robot functionality such as SLAM.**



ROS – Philosophy

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- Peer to Peer
- Tool-Based
- Multi-Lingual
- Thin
- Free & Open Source, Community-Based, Repositories



ROS Platforms

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

Jade

Indigo

Hydro

Groovy Galapagos

Fuerte Turtle


Electric

Diamondback

C Turtle

Box Turtle

These are the some example of ROS.

ROS – Applications

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

Fraunhofer IPA
Care-O-bot

Aldebaran Nao

Willow Garage
PR2

Merlin miabotPro

Clearpath
Robotics Husky

Videre Erratic

Lego NXT

iRobot Roomba

AscTec Quadrotor

Clearpath
Robotics
Kingfisher

TurtleBot

Shadow Hand

Robotnik Guardian

CoroWare Corbot

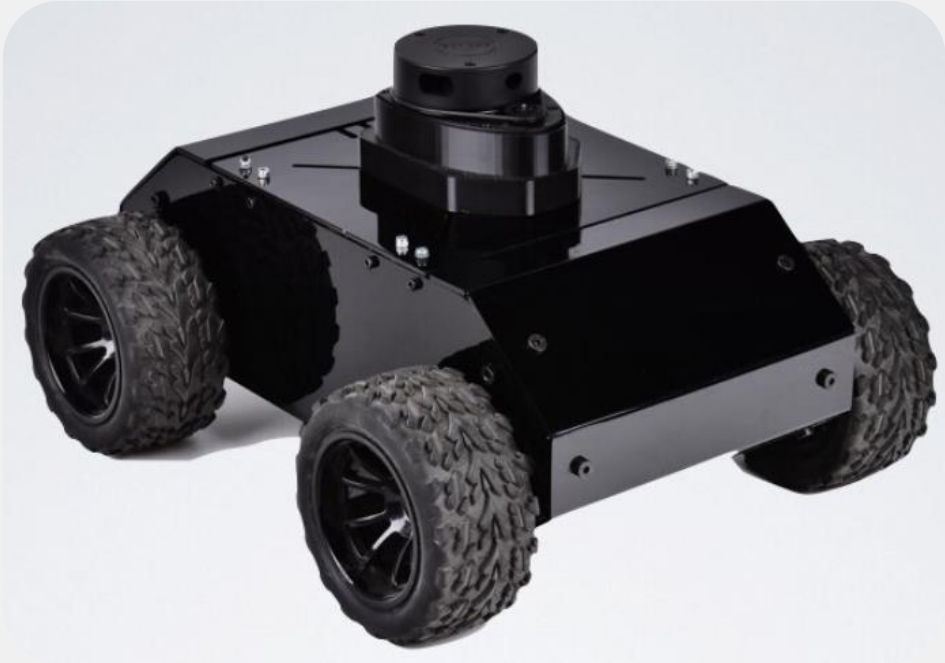
Festo Didactic
Robotino

ROS – Concepts & Components – Client Libraries

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **Main Client Libraries**
  - **Python**
  - **C++**
  - **Lisp**

- **Experimental Client Libraries**
  - **Java (with Android Support)**
  - **Lua**



ROS – Concepts & Components – Supported OS

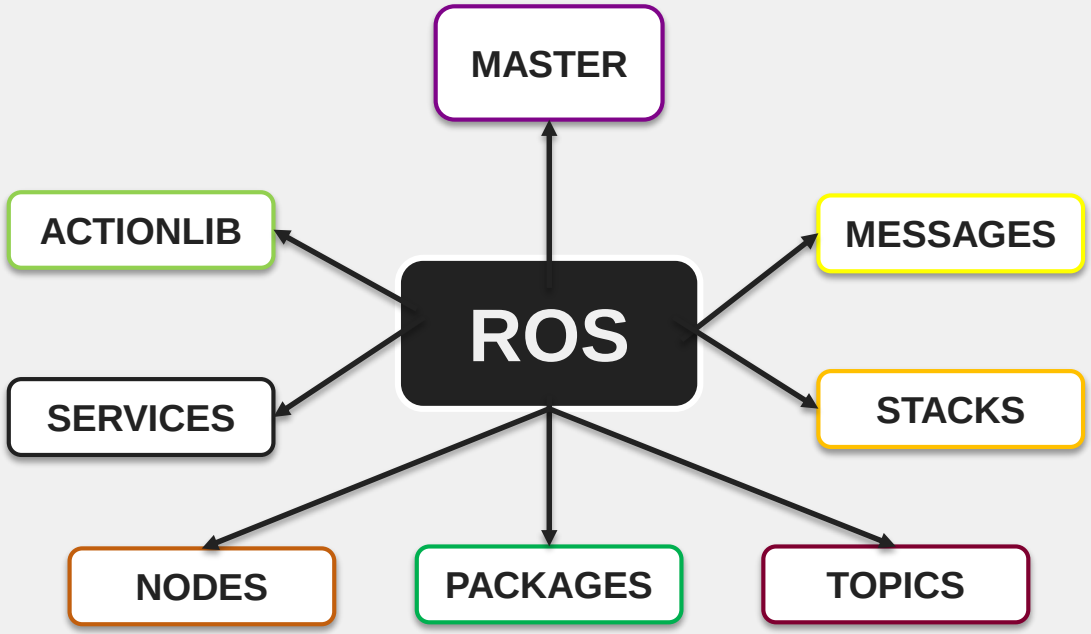
AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- Supported OS
  - Ubuntu (14+ LTS + ROS Indigo)
- Experimental
  - Arch
  - Debian
  - Fedora
  - Gentoo
  - Mac OS X
  - OpenSuse
  - Windows

ROS Core Concepts

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

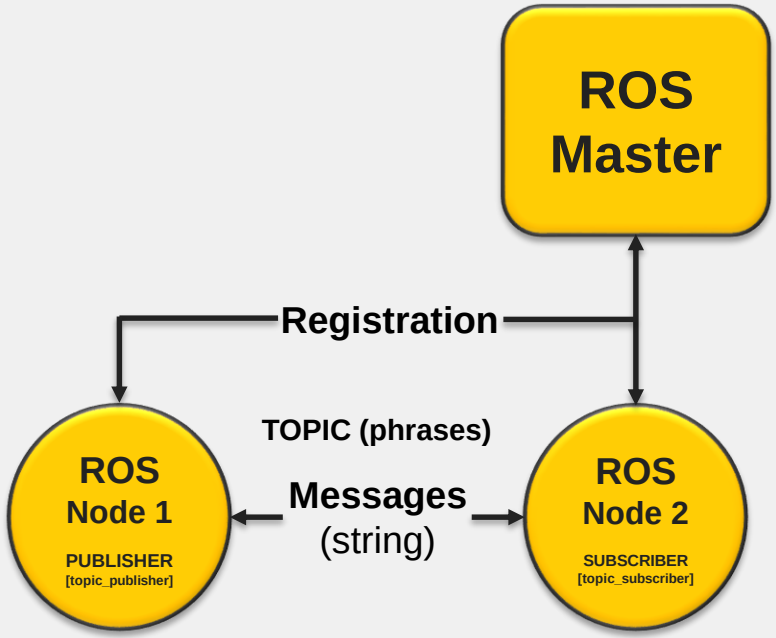
- Nodes
- Messages & Topics
- Services
- Actions
- ROS Master
- Parameters
- Packages & Stacks



ROS Core Concepts – Nodes

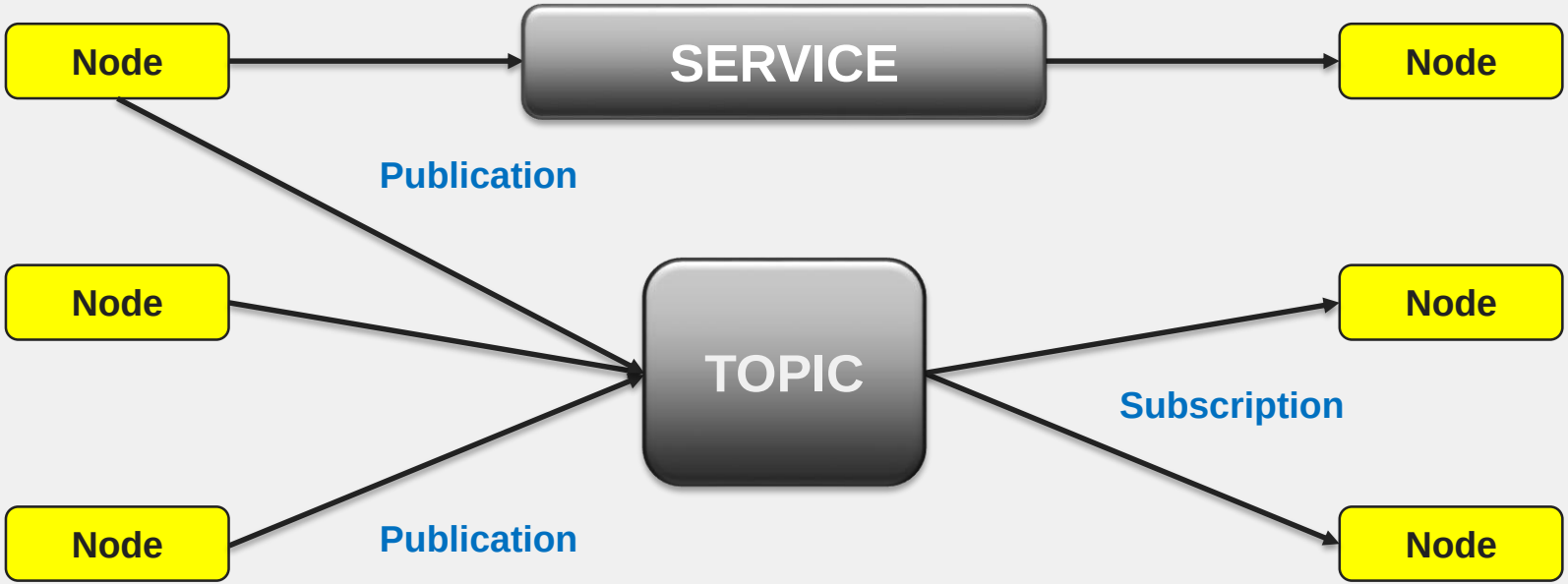
AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **Single Purpose Executable Programs**
  - **E.g. Sensor drivers, actuator drivers etc.**
- **Individually Compiled**
- **Written using ROS Client Library**
- **Provide or use Service or Action**



ROS Core Concepts – Nodes

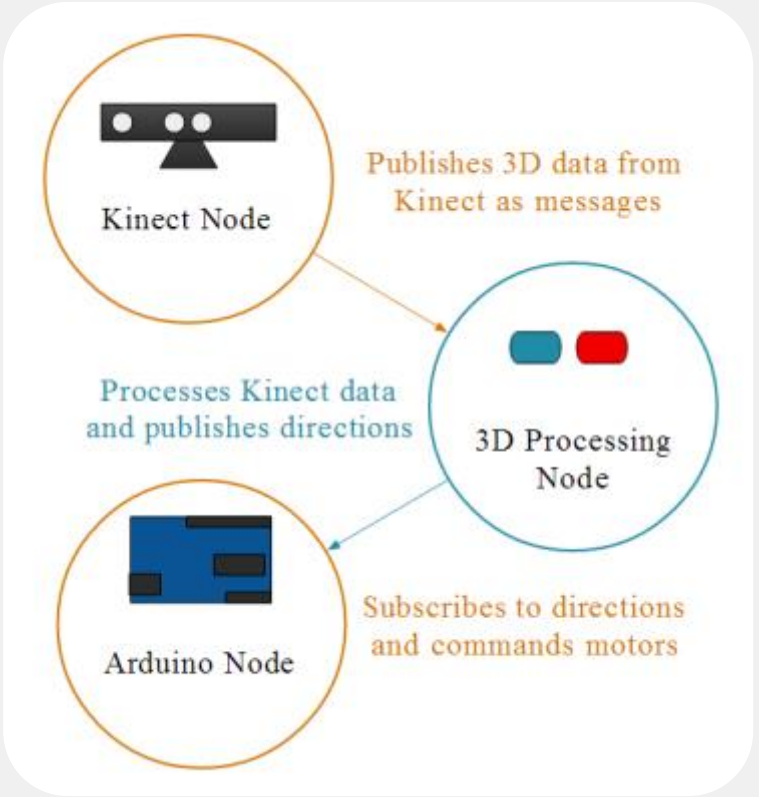
AUGMENTED STARTUPS
Computer Vision | AI | Robotics



ROS Concepts – Topics & Messages

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

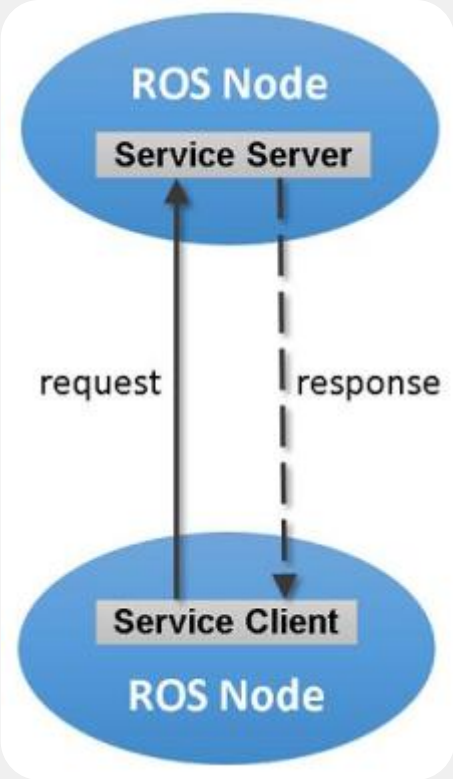
- **Topic: Stream of messages with a defined type**
    - **Data from a range-finder might be sent on a topic called scan, with a message of type *LaserScan.***
- **Messages: Strictly-typed data structures for inter-node communication**



ROS Concepts – Services

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

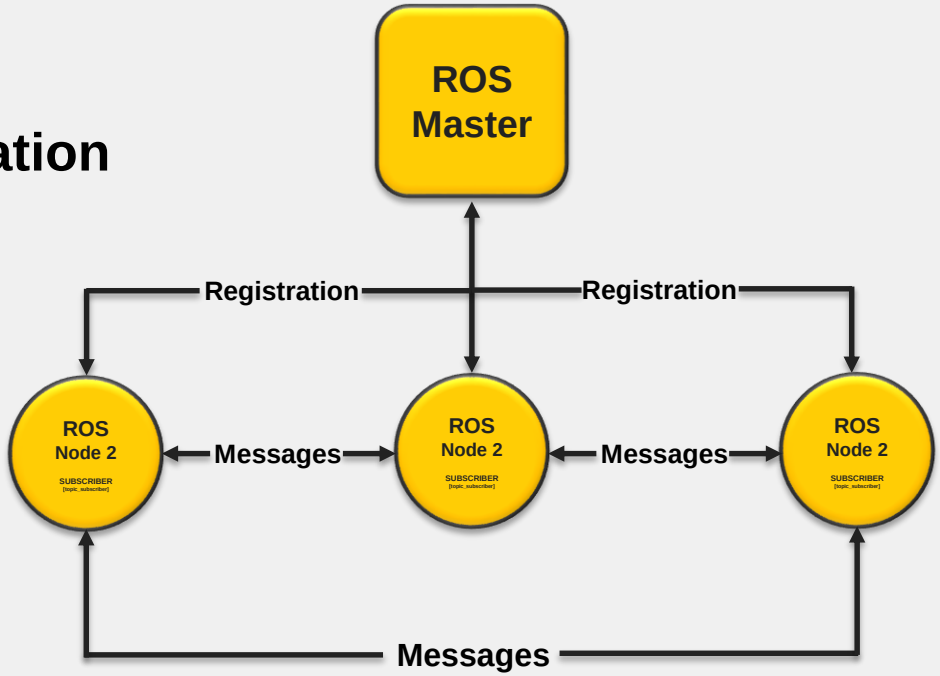
- Synchronous inter-node transactions (blocking RPC)
- Service/Client Model:
  - 1-to-1 request-response
- Service Roles:
  - Carryout remote computation
  - Trigger functionality/behavior
  - Map_server/static_map



ROS Concepts – ROS Master

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

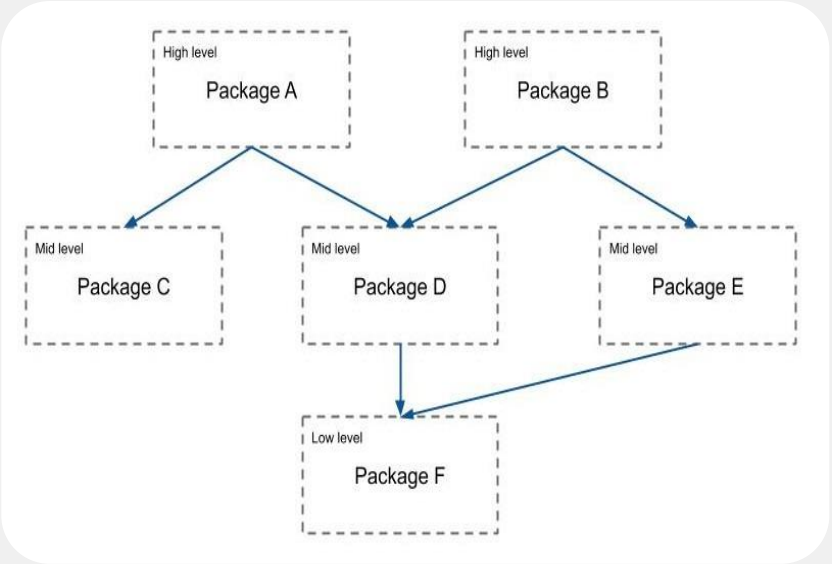
- Provides connection information to nodes so that they can transmit messages to each other.
- Every node connects to a specified master to register.



ROS Concepts – ROS Packages

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **Software in ROS is organized of packages.**
- **Contains one or more nodes, documentation, & provide a ROS interface.**
- **Most of ROS packages are hosted in GitHub.**



Installation of ROS

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **Setup sources.list**
  - *$ sudo sh-c' echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release-sc) main" > /etc/apt/sources.list.d/ros-latest.list'*
- **Setup Keys**
  - *$ sudo apt-key adv –keyserver hkp://pool.sks-keyservers.net –recv-key oxBo1FA116*
- **Install ROS Desktop-Full, & standalone tools**
  - *$ sudo apt-get update*
  - *$ sudo apt-get install ros-indigo-desktop-full*
  - *$ sudo resdep init*
  - *$ rosdep update*
- **Setup sources.list**
  - *$ echo \source /opt/ros/indigo/setup.bash" >> /.bashrc*
  - *$ /.bashrc*

Setting up ROS Environment for New User

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **Type in the following commands – Remember that spaces are necessary, & Linux is case sensitive!**
  - *Echo ``source /opt/ros/indigo/setup.bash”>>~/.bashrc*
  - *Source ~/.bashrc*
  - *$ mkdir –p ~/catkin_ws/src*
  - *$ cd ~/catkin_ws/src*
  - *$ catkin_init_workspace*
  - *$ cd ~/catkin_ws/*
  - *$ catkin_make*
  - *$ echo ``source ~/catkin_ws/devel/setup.bash”>>~/.bashrc*
  - *$ source ~/.bashrc*
  - *Echo $ROS_PACKAGE_PATH*

ROS Filesystem – Catkin Workspace

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- *workspace_folder/* -- WORKSPACE
- *build/* -- BUILD SPACE CMake is invoked to build the catking packages in the source space
- *devel/* -- DEVEL SPACE where build targets are placed prior to being installed
- *src/* -- SOURCE SPACE
- *CMakeLists.txt* -- 'Toplevel' CMake file, provided by catkin
- *package_1/*
- *CMakeLists.txt* -- CMakeLists.txt file for package_1
- *package.xml* -- Package manifest for package_1
- *…*
- *package_n/*
- *CMakeLists.txt* -- CMakeLists.txt file for package_n
- *package.xml* -- Package manifest for package_n
- *meta_package/* -- Collections of packages
- *sub_package_1/*
- *CMakeLists.txt* -- CMakeLists.txt file for sub_package_1
- *package.xml* -- Package manifest for sub_package_1
- *…*
- *sub_package_n/*
- *CMakeLists.txt* -- CMakeLists.txt file for sub_package_n
- *package.xml* -- Package manifest for sub_package_n
- *meta_package/*
- *package.xml* -- Package manifest indicating the meta_package

ROS Filesystem – Package Example

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

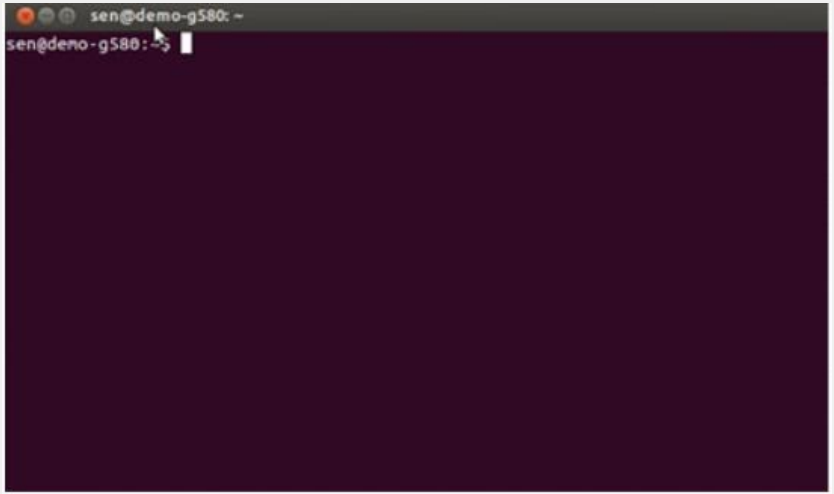
- **Hypothetical Package myPkg/**

- ***CMakeLists.txt:*** CMake build settings for package myPkg
- ***package.xml:*** metadata and dependencies required by package
- ***mainpage.dox:*** doc information of package myPkg
- ***include/myPkg:*** c++ header files
- ***src/:*** source code directory
- ***launch/:*** where launch files are stored (if needed)
- ***msg/:*** message (.msg) types
- ***srv/:*** service (.srv) types
- ***scripts/:*** executable scripts

robash – ROS Command Line Tools

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **Open up a terminal**
- Press \windows" key, then type \terminal", then press \Enter or use shortcut Ctrl+Alt+T



robash – ROS Command Line Tools

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **rospack: ROS package management tool**
  - *$ rospack list*
  - *$ rospack find turtlesim*
  - *$ rospack depends turtlesim*
  - *$ rospack profile*
- **roscd: change directory command for ROS**
  - *$ roscd*
  - *$ roscd turtlesim*
  - *$ ls (standard linux shell command)*
- **rosls: allows you to list the contents of a ROS package**
  - *$ roscd (return to workspace directory)*
  - *$ rosls turtlesim*

rosnode

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **The current list of supported commands are**
  - ***$ rospack list*** kill a running node
  - ***$ rosnode list*** list active nodes
  - ***$ rosnode machine*** list nodes running on a machine
  - ***$ rosnode ping*** test connectivity to node
  - ***$ rosnode info*** print information about node

rostopic

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **The current list of supported commands are**
  - ***$ rostopic bw*** display bandwidth used by topic
  - ***$ rostopic echo*** print messages to screen
  - ***$ rostopic find*** find topics by type
  - ***$ rostopic hz*** displays publishing rate of topic
  - ***$ rostopic info*** print information about active topic
  - ***$ rostopic list*** print information about active topics
  - ***$ rostopic pub*** publish data to topic

ROS Development Procedures

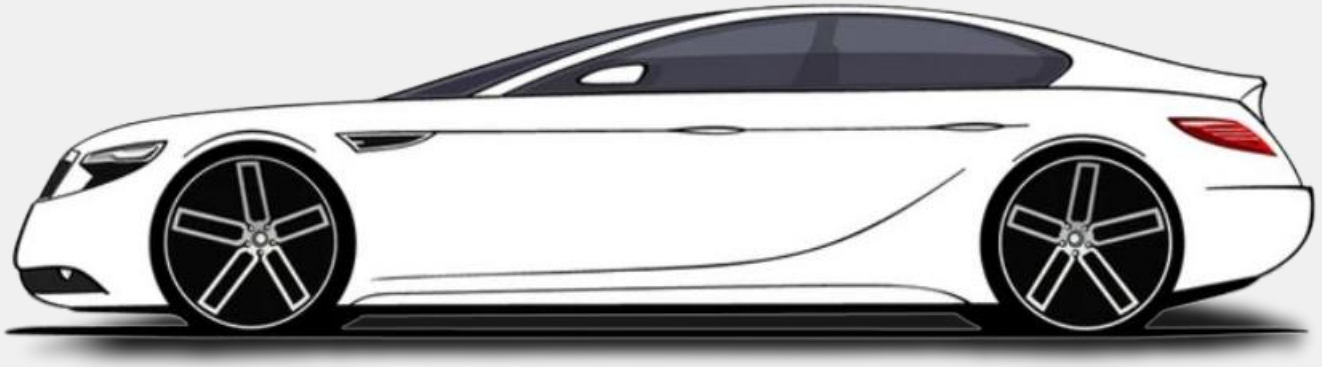
AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- Create a new catkin workspace
- Create a new ROS package
- Download and configure Eclipse
- Create Eclipse project file for your package
- Import package into Eclipse
- Write the Code
- Update the make file
- Build the package

How ROS Work in SDC?

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

PERCEPTION + DECISION MAKING + ACTUATION

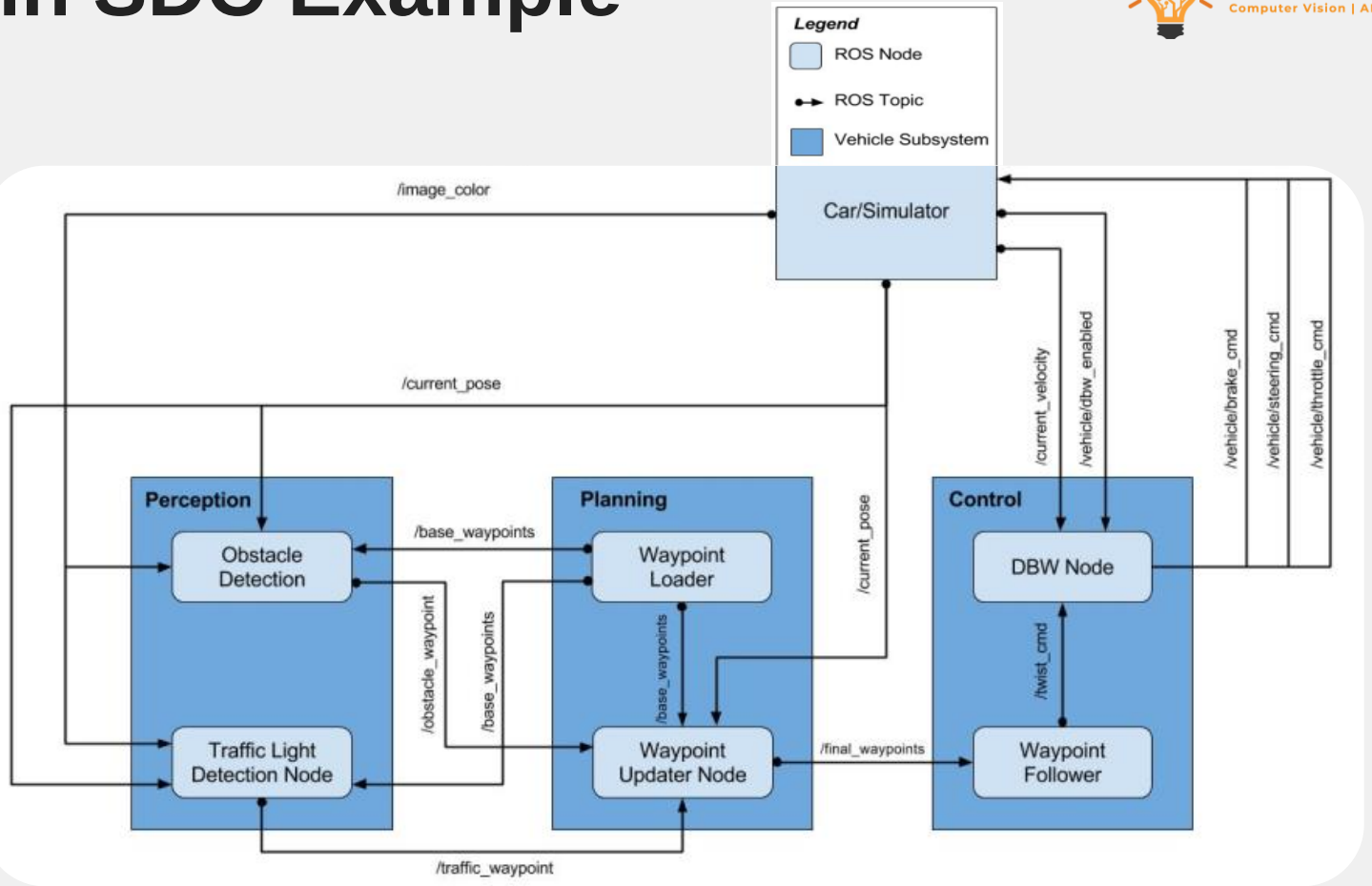


ROS Nodes in SDC

- **PERCEPTION:**
  - **Camera, GPS, LIDAR, Wheel Encoder, & Radar**
- **DECISION MAKING:**
  - **Path Planning, Trajectory Sampling, & Deep Learning**
- **ACTUATION:**
  - **Steering, Brakes, & Throttle**

ROS in SDC Example

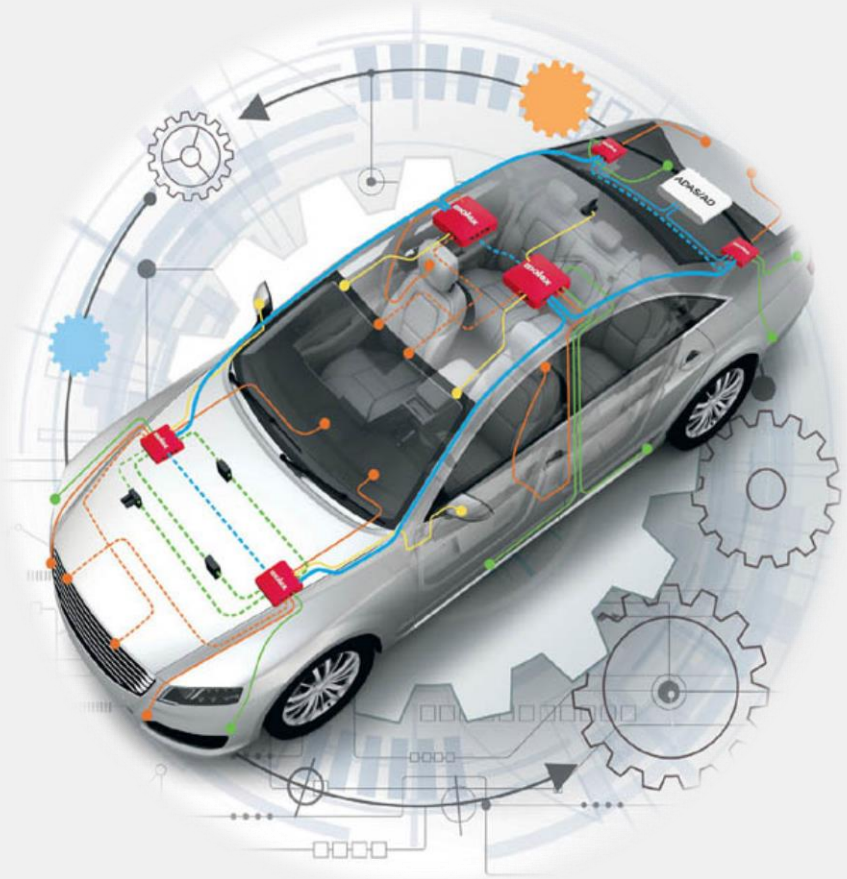
AUGMENTED STARTUPS
Computer Vision | AI | Robotics



Summary

AUGMENTED STARTUPS
Computer Vision | AI | Robotics

- **Challenges in Robotics**
- **Solution to the Challenges**
- **ROS (Robot Operating System)**
  - Definition
  - Why and Why not ROS?
  - Main Features
  - ROS Philosophy
  - Applications Using ROS
  - ROS Components & Concepts
  - Supported Operating Systems
- **ROS Concepts**
  - ROS Nodes
  - ROS Services
  - ROS Packages
- **ROS in Self Driving Cars**



Reference

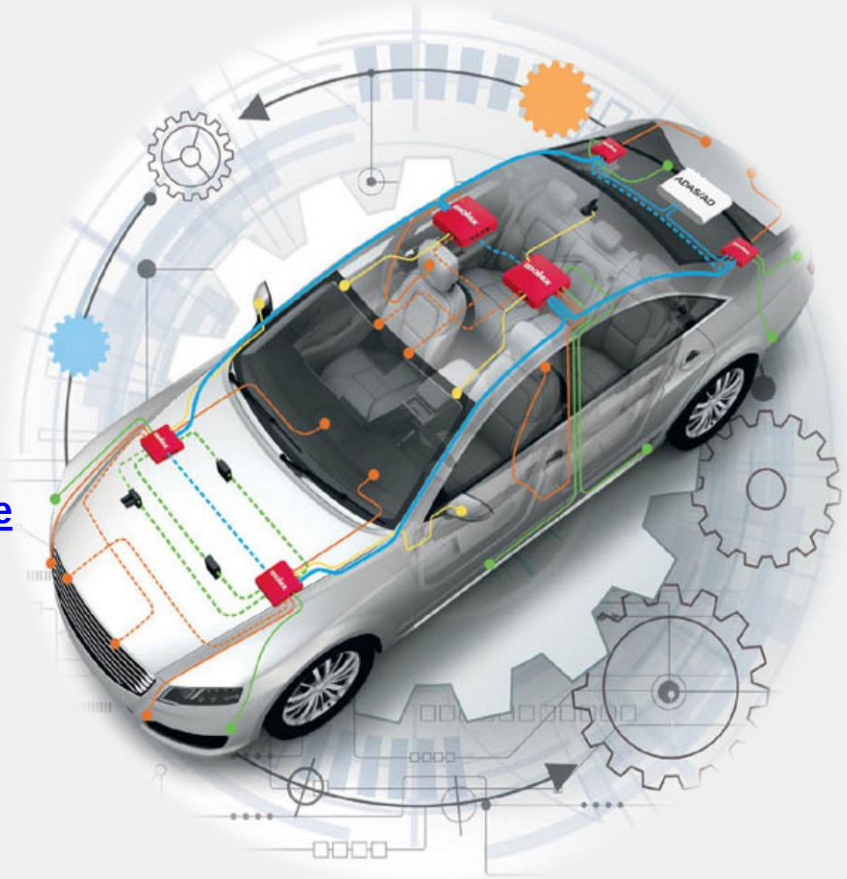
AUGMENTED STARTUPS
Computer Vision | AI | Robotics

1. https://www.theconstructsim.com/start-self-driving-cars-using-ros/

2. https://nl.mathworks.com/help/ros/ug/call-and-provide-ros-services.html

3. https://www.semanticscholar.org/paper/ROS-As-a-Service%3A-Web-Services-for-Robot-Operating-Koub%C3%A2a/b921aa54b96497c727c2c6cc73ed5108de4a9166

4. http://wiki.ros.org/ROS/Introduction#:~:text=Next-,What%20is%20ROS%3F,between%20processes%2C%20and%20package%20management.



Thanks