ROBOTIC MIDDLEWARES

ROBOTICS

POLITECNICO
MILANO 1863

MIDDLEWARE ORIGINS

goo.gl/DBwhhC

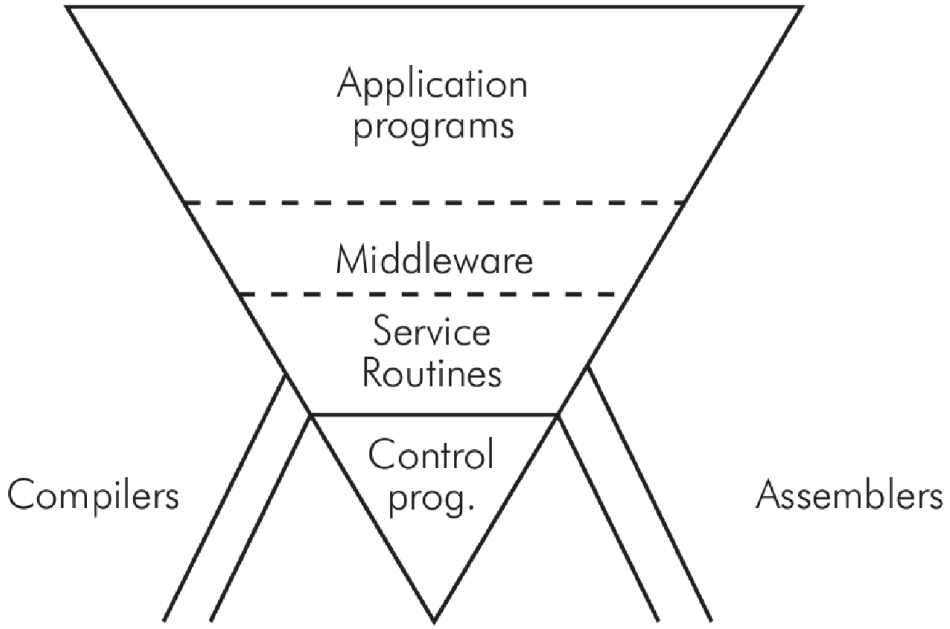
The origins

1968 introduced by d'Agapeyeff

80's wrapper between legacy systems and new applications

Nowadays: widespread in different domain fields (including Robotics)

Some (non robotics) examples: Android, SOAP, Web Services, …



MIDDLEWARE ORIGINS

goo.gl/DBwhhC

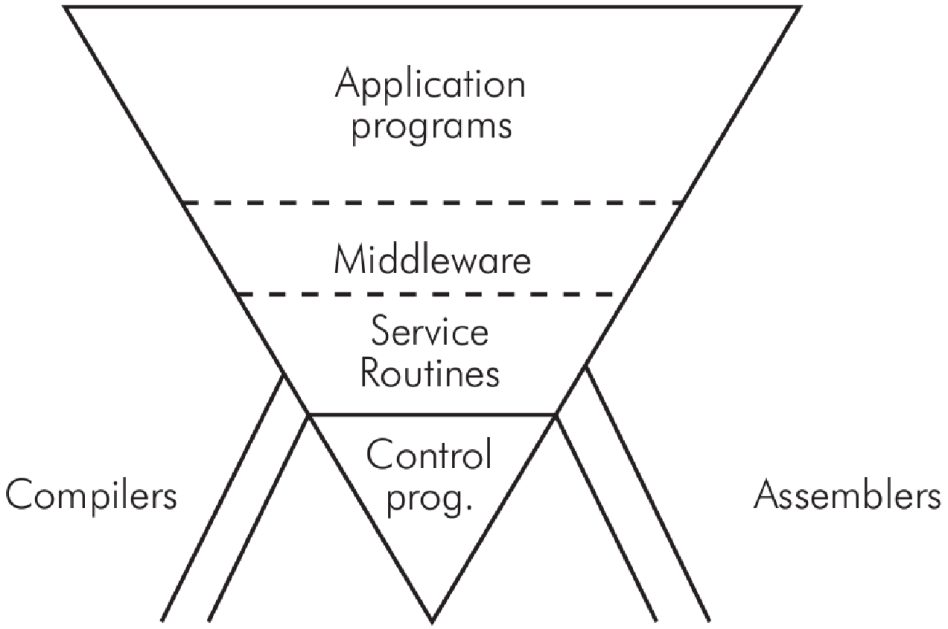
The Middleware idea

Well-known in software engineering

It provides a computational layer

A bridge between the application and the low-level details

It is not a set of API and library



MIDDLEWARE ORIGINS

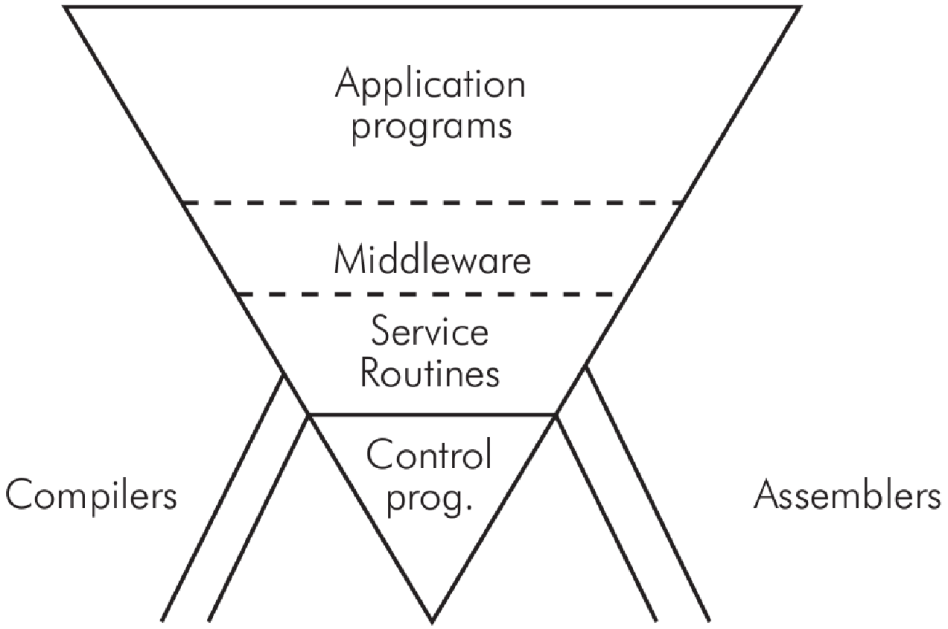
goo.gl/DBwhhC

Issues in developing real robots

Cooperation between hardware and software

Architectural differences in robotics systems

Software reusability and modularity

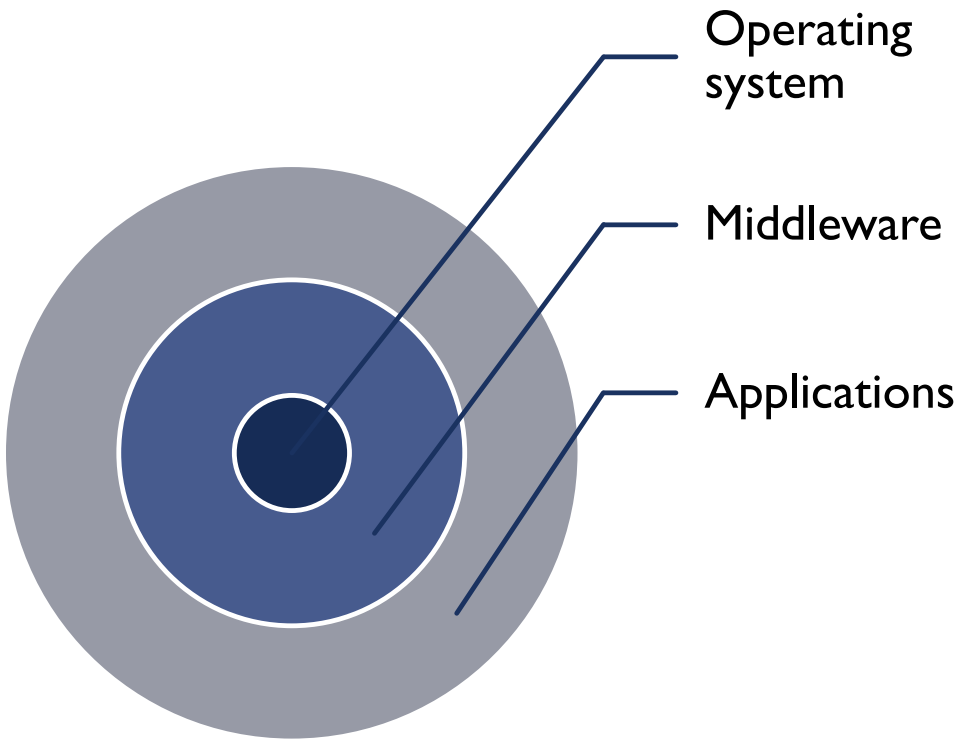


WHAT IS A MIDDLEWARE?

goo.gl/DBwhhC

*Software that **connects** different software components or applications:*

- Set of services that permits to several processes to interact
- Framework used to reduce the developing time in complex systems.



WHAT IS A MIDDLEWARE?

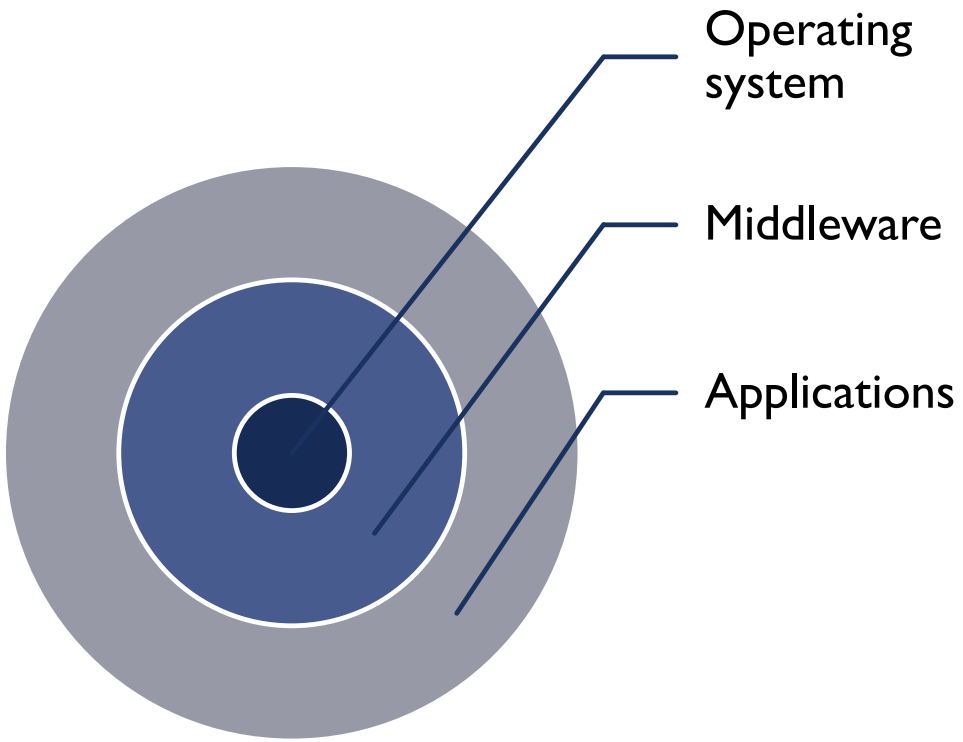
goo.gl/DBwhhC

Middleware vs. Operating System

The middleware stays between software and different operating systems.

The distinction between operating system and middleware is sometimes arbitrary.

Some features of a middleware are now integrated in operating systems (e.g., TCP/IP stack).



MIDDLEWARES MAIN FEATURES

goo.gl/DBwhhC

Portability: provides a common programming model regardless the programming language and the system architecture.

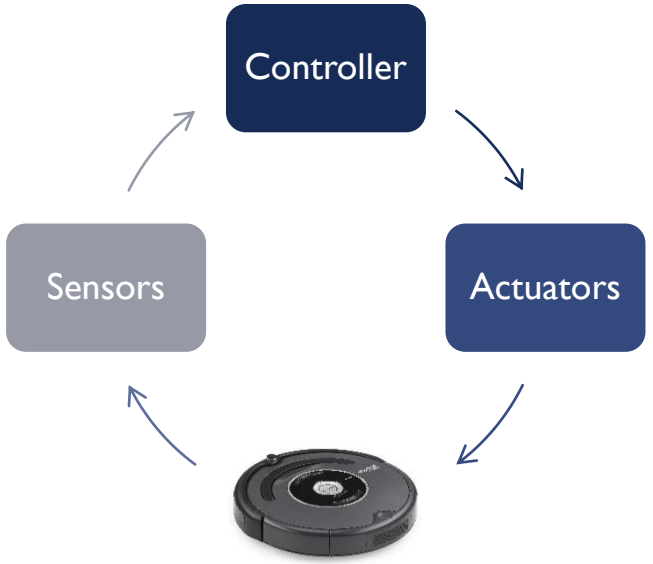
Reliability: middleware are tested independently. They permit to develop robot controllers without considering the low level details and using robust libraries.

Manage the complexity: low-level aspects are handled by libraries and drivers inside the middleware. It (should) reduce(s) the programming error and decrease the development time.

ROBOT DEVELOPMENT

goo.gl/DBwhhC

1. Modelling
   - Kinematic model
   - Differential kinematics
   - Dynamic model
2. Planning
   - Motion laws
   - Trajectory generation
3. Control
   - Translate the movement into motor commands
   - Several type of control: motion, force, etc.



Before the introduction of middleware

- Monolithic approach
- Little if any reuse of models or components
- Hard to maintain code and hard to integrate components

Some people believe the real issue with Robotics is **integration**!

ROBOT MIDDLEWARES: A LIST

goo.gl/DBwhhC

Several middleware have been developed in recent years:

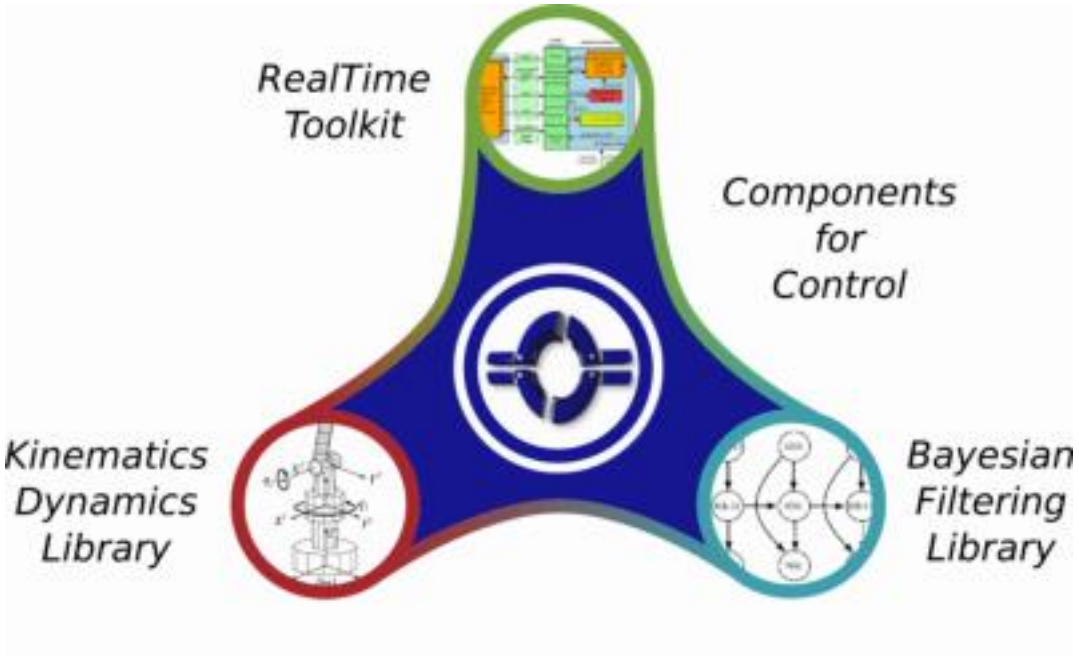
OROCOS	[Europe]
ORCA	[Europe]
YARP	[Europe / Italy]
BRICS	[Europe]
OpenRTM	[Japan]
OpenRave	[US]
ROS	[US]
…	

Let's see their common features and main differences

OROCOS: OPEN ROBOT CONTROL SOFTWARE

goo.gl/DBwhhC

The project started in December 2000 from an initiative of the mailing list EURON then it become an European project with 3 partners: K.U. Leuven (Belgium), LAAS Toulouse (France), KTH Stockholm (Sweden)

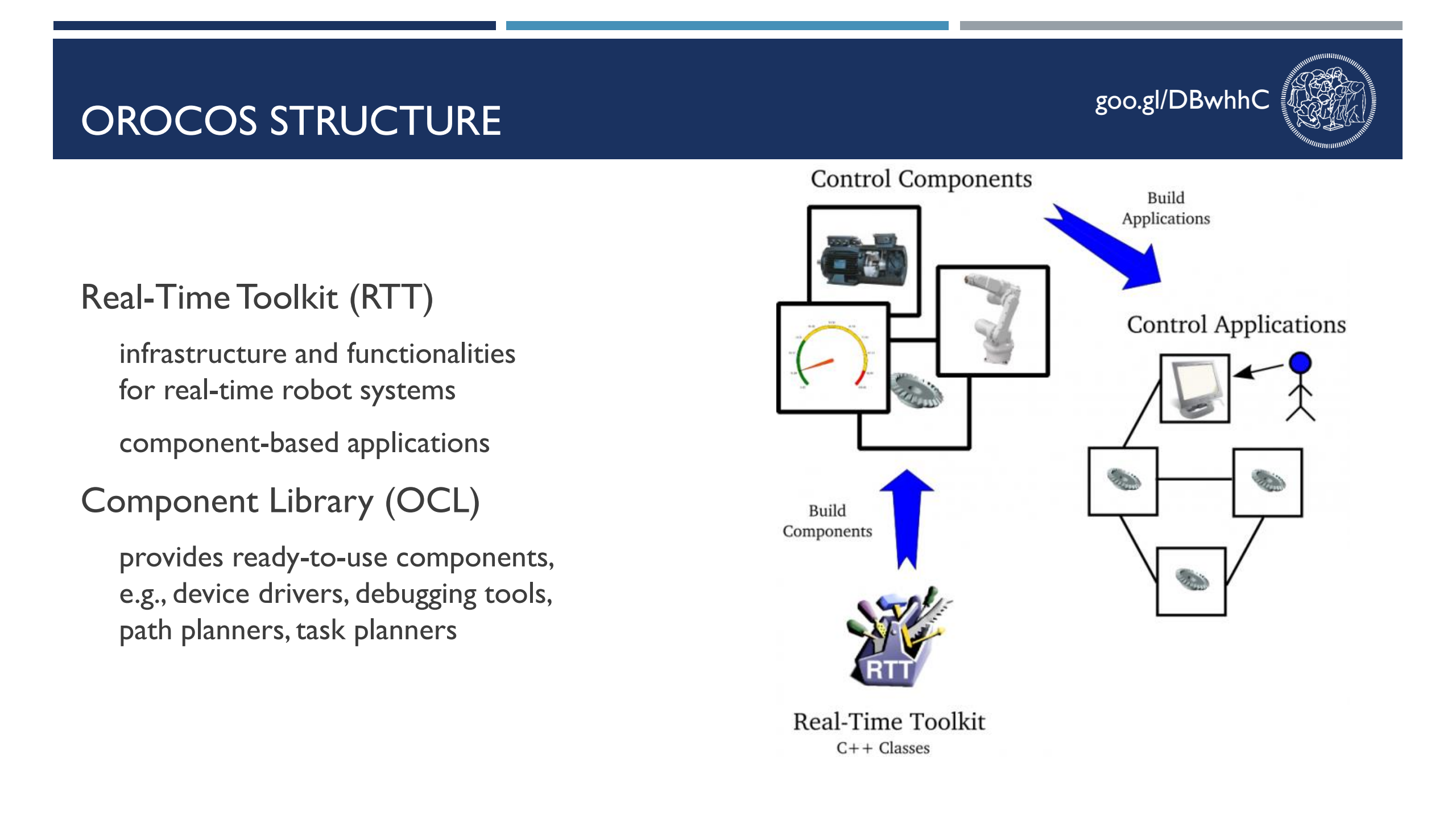


OROCOS requirements:

- Open source license
- Modularity and flexibility
- Not related to robot industries
- Working with any kind of device
- Software components for kinematics, dynamics, planning, sensors, controller
- Not related to a unique programming language

OROCOS STRUCTURE

goo.gl/DBwhhC



Real-Time Toolkit (RTT)

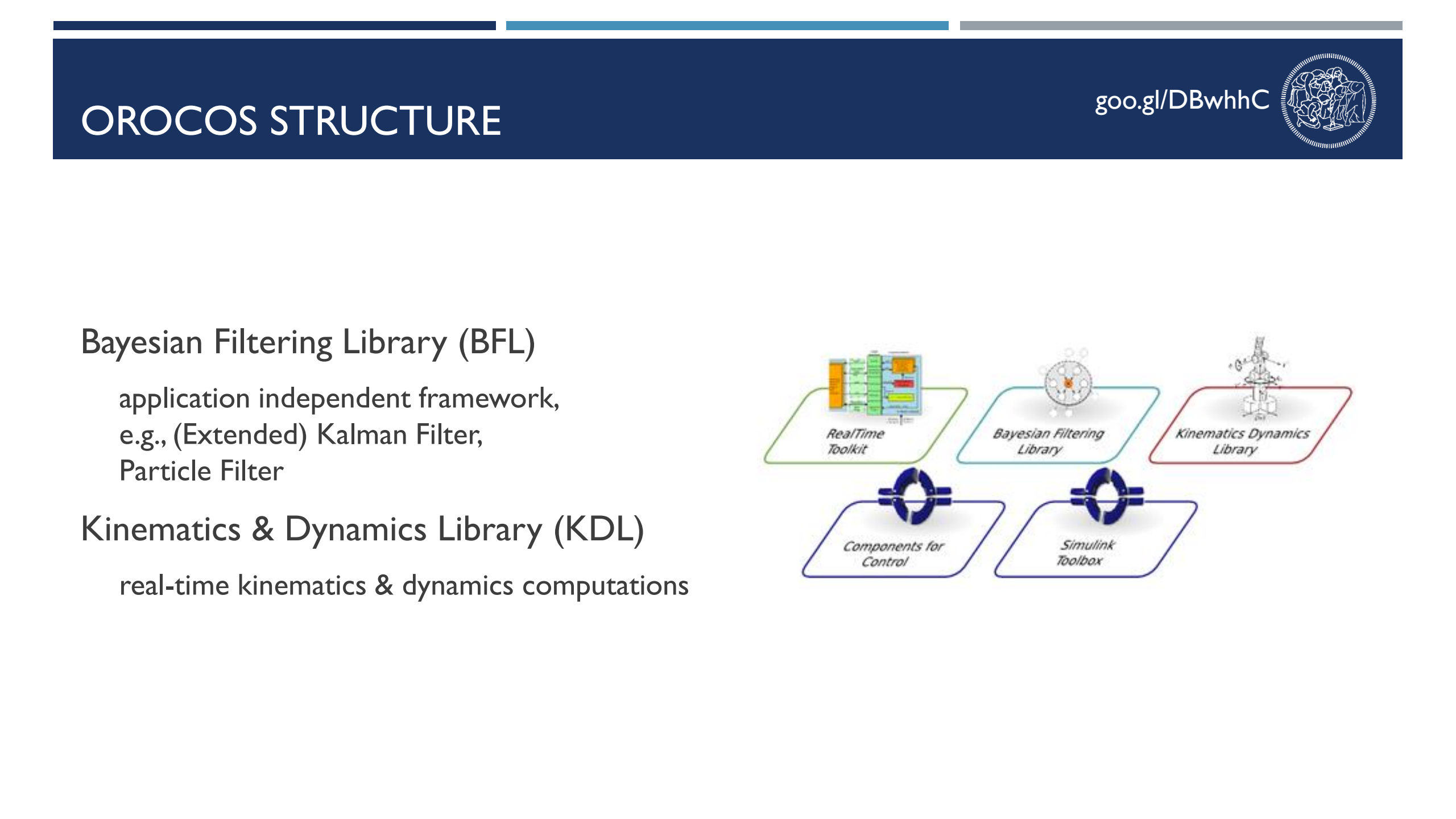
- infrastructure and functionalities for real-time robot systems
- component-based applications

Component Library (OCL)

- provides ready-to-use components, e.g., device drivers, debugging tools, path planners, task planners

OROCOS STRUCTURE

goo.gl/DBwhhC



Bayesian Filtering Library (BFL)

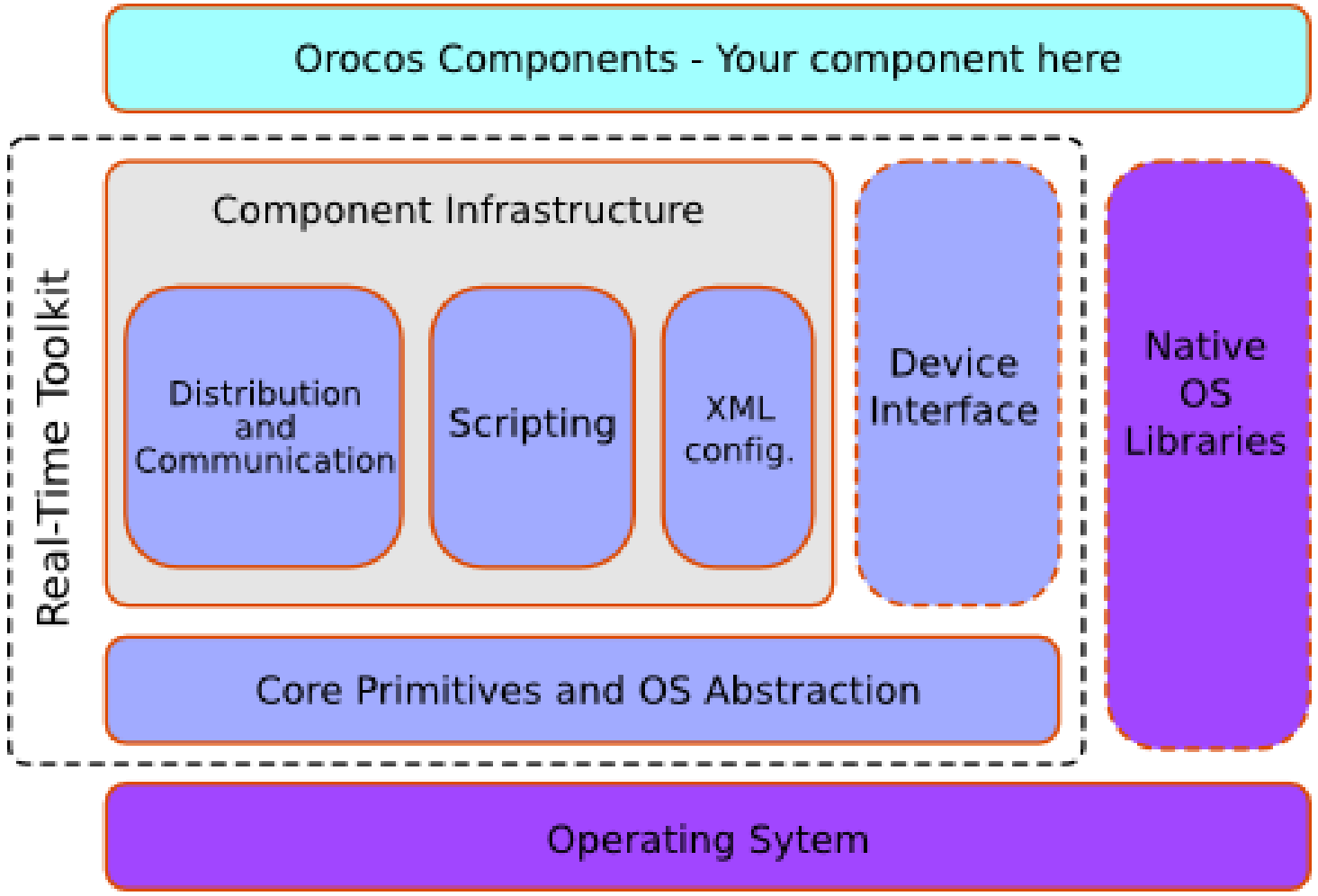
- application independent framework,
  e.g., (Extended) Kalman Filter,
  Particle Filter

Kinematics & Dynamics Library (KDL)

- real-time kinematics & dynamics computations

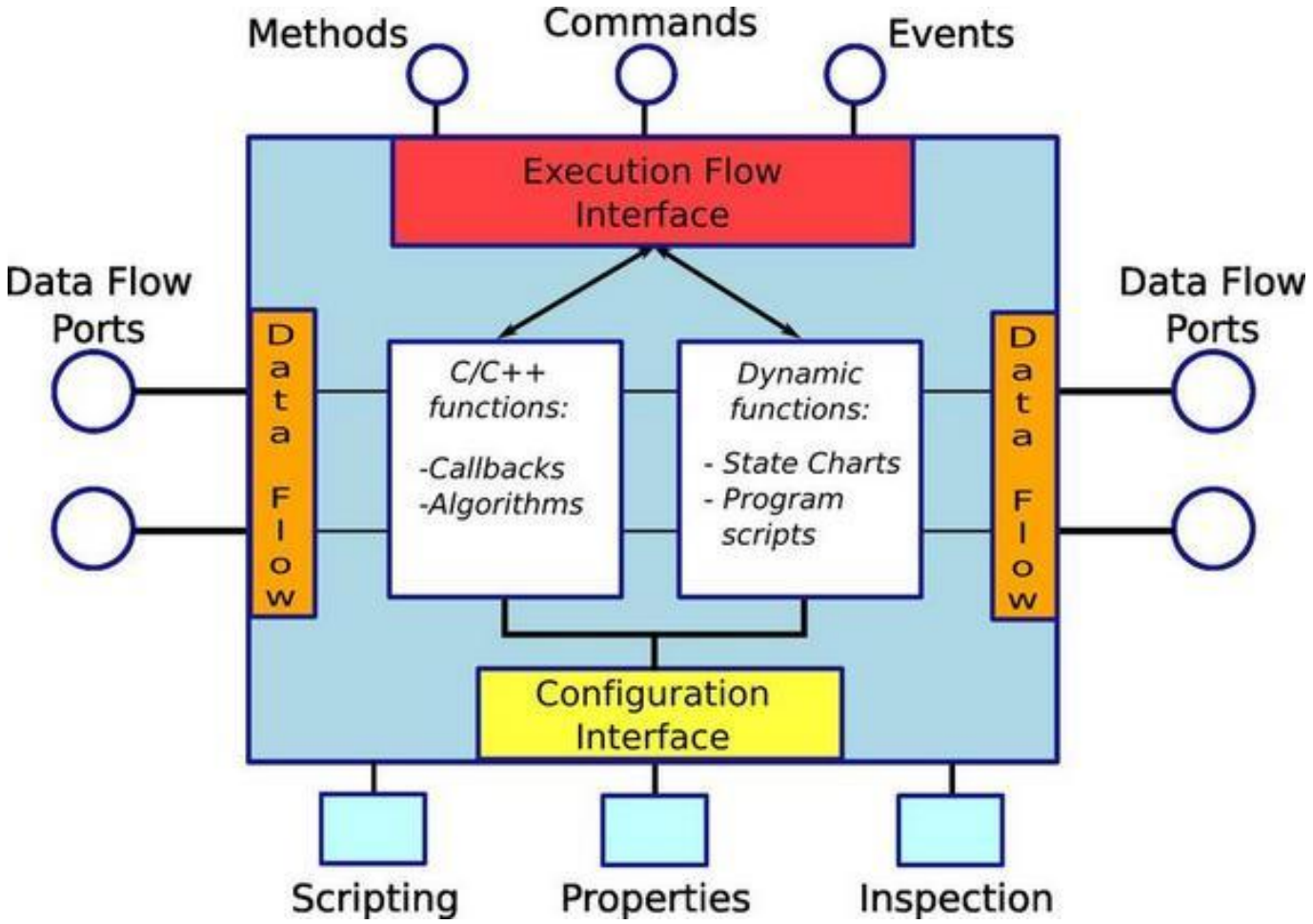
OROCOS RTT FRAMEWORK

goo.gl/DBwhhC



OROCOS COMPONENT

goo.gl/DBwhhC



ORCA: COMPONENTS FOR ROBOTICS

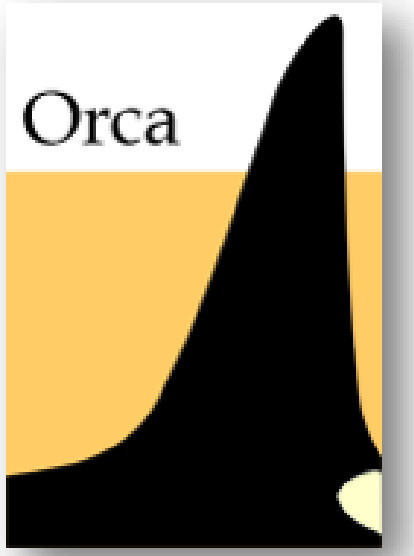
goo.gl/DBwhhC

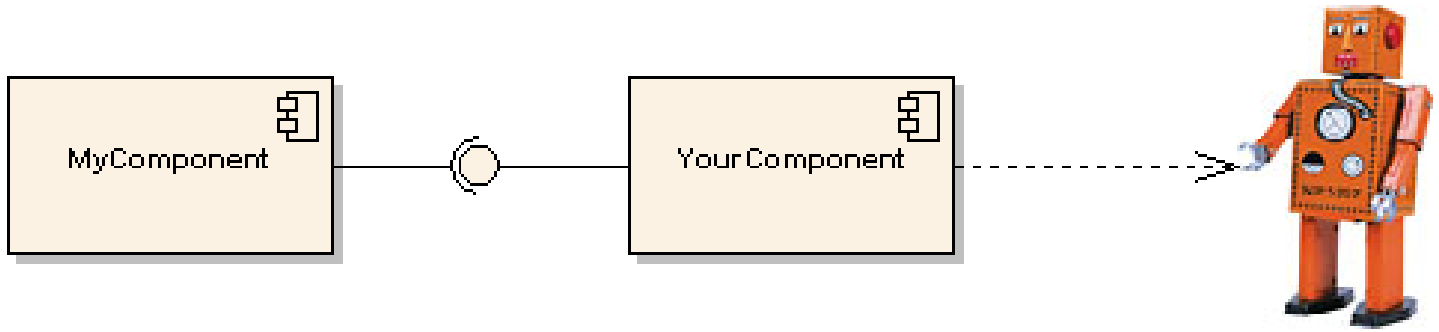
The aim of the project is to focus on software reuse for scientific and industrial applications

Key properties:

- commonly-use interfaces
- high-level libraries
- updated software repositories

ORCA defines itself as "unconstrained component-based system"





ORCA AND ICE

goo.gl/DBwhhC

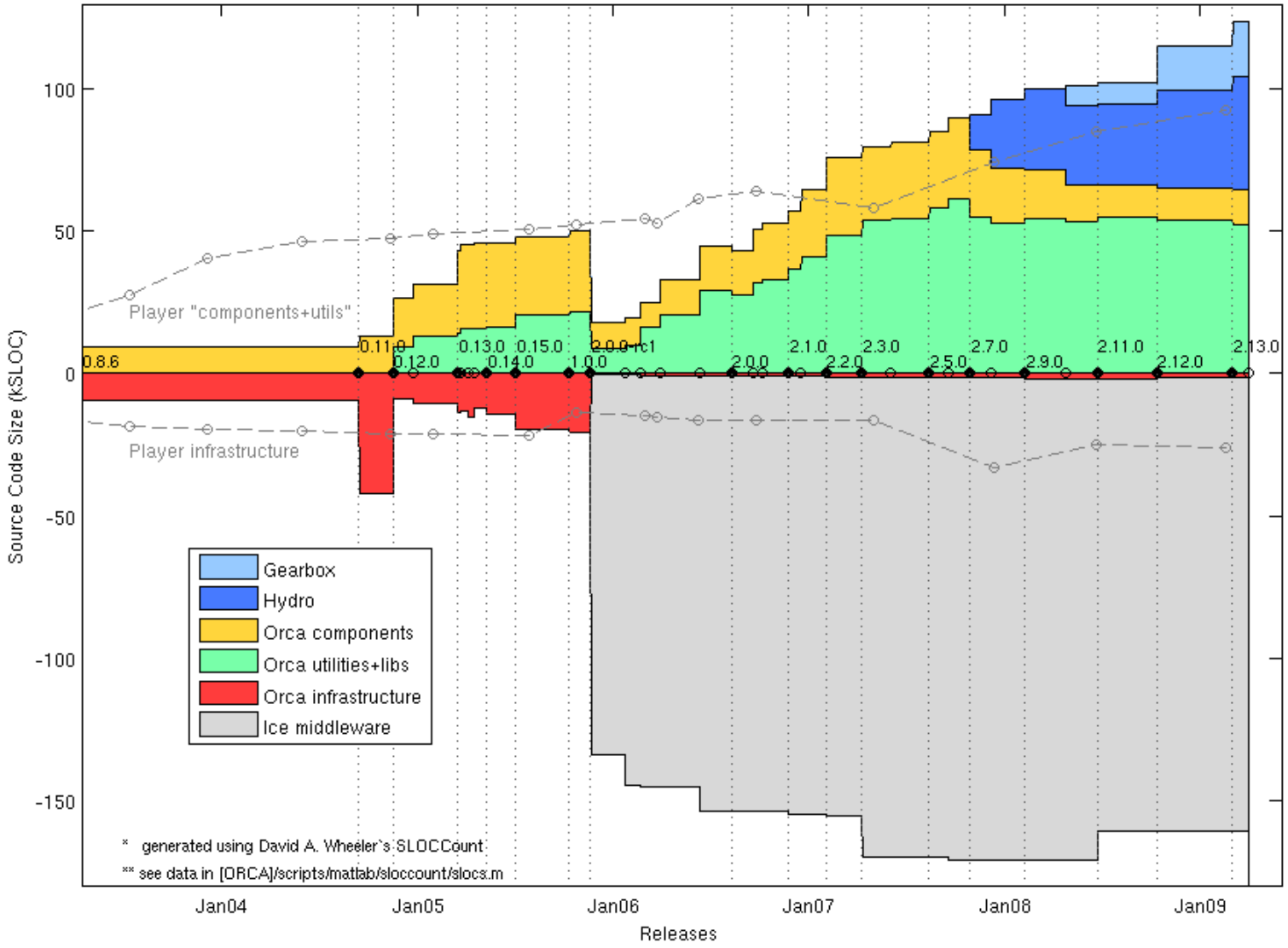
The main difference between OROCOS and ORCA is the communication toolkit; OROCOS uses CORBA while ORCA uses ICE

- ICE is a modern framework developed by ZeroC
- ICE is an open-source commercial communication system
- ICE provides two core services
  - IceGrid registry (Naming service): which provides the logic mapping between different components
  - IceStorm service (Event service): which constitute the publisher and subscriber architecture

"A component can find the other components through the IceGrid registry and can communicate with them through the IceStorm service."

ORCA: LIBRARY EVOLUTION

goo.gl/DBwhhC



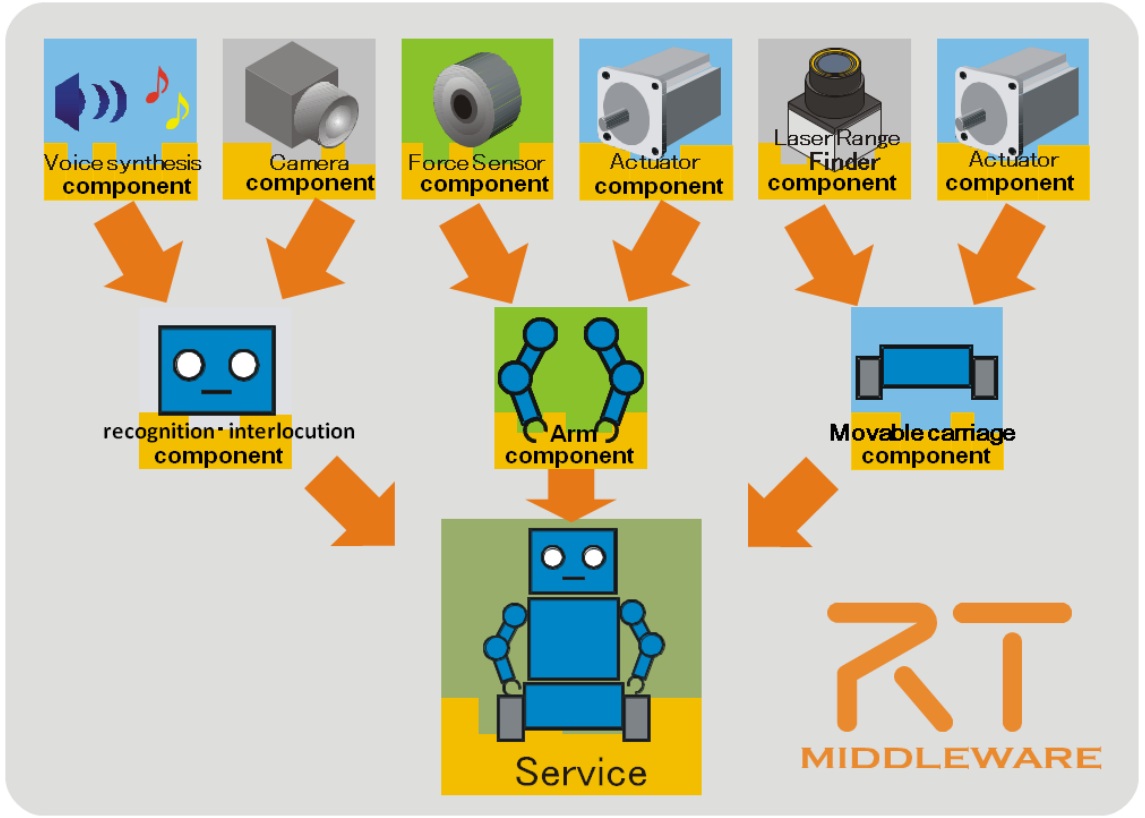
RT MIDDLEWARE

goo.gl/DBwhhC

RT-Middleware (RTM) is a common platform standard to construct the robot system by combining the software modules of the robot functional elements (RTC):

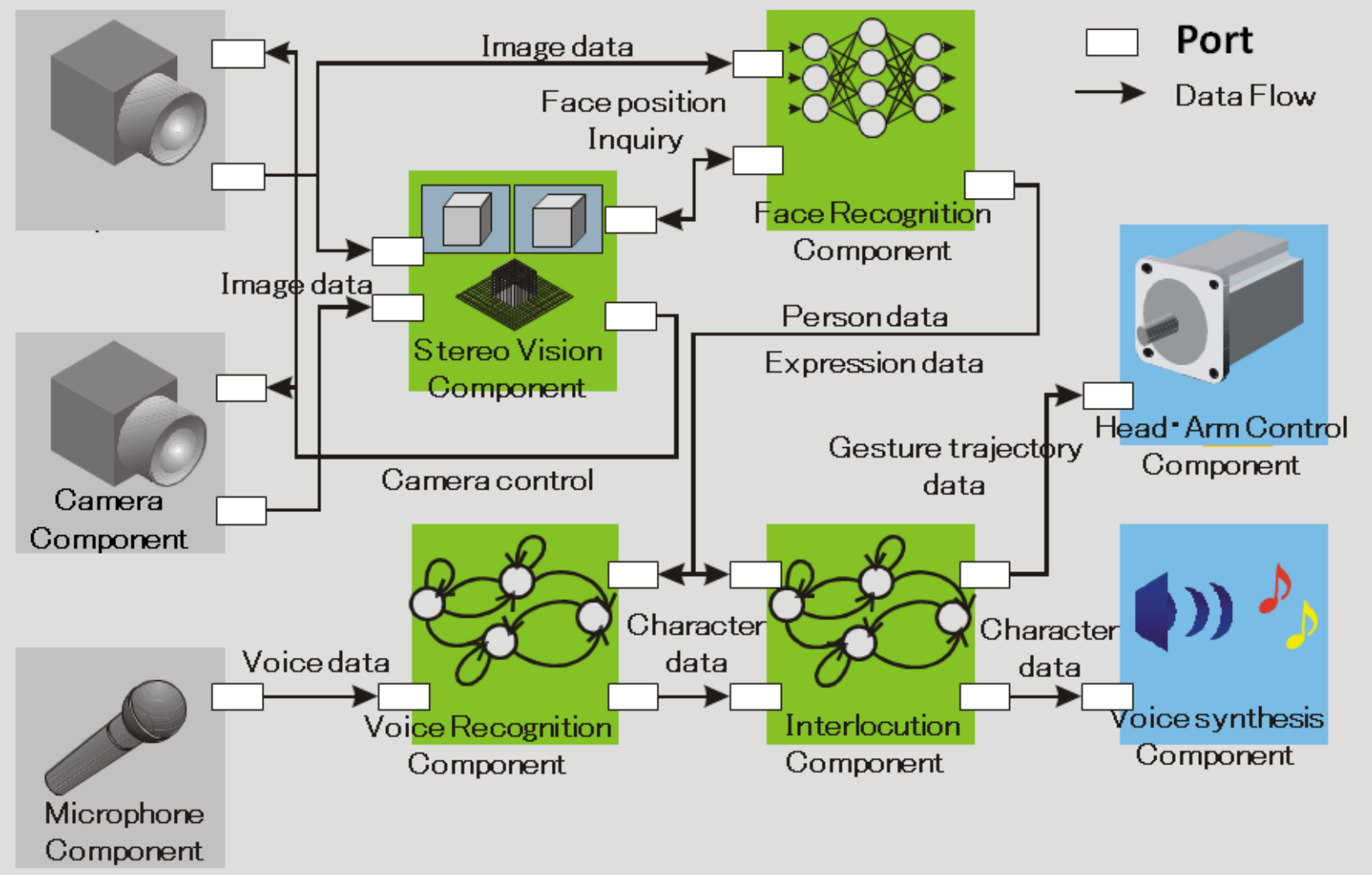
- Camera component
- Stereovision component
- Face recognition component
- Microphone component
- Speech recognition component
- Conversational component
- Head and arm component
- Speech synthesis component

OpenRTM-aist (Advanced Industrial Science & Technology) is based on the CORBA technology to implement RTC extended specification



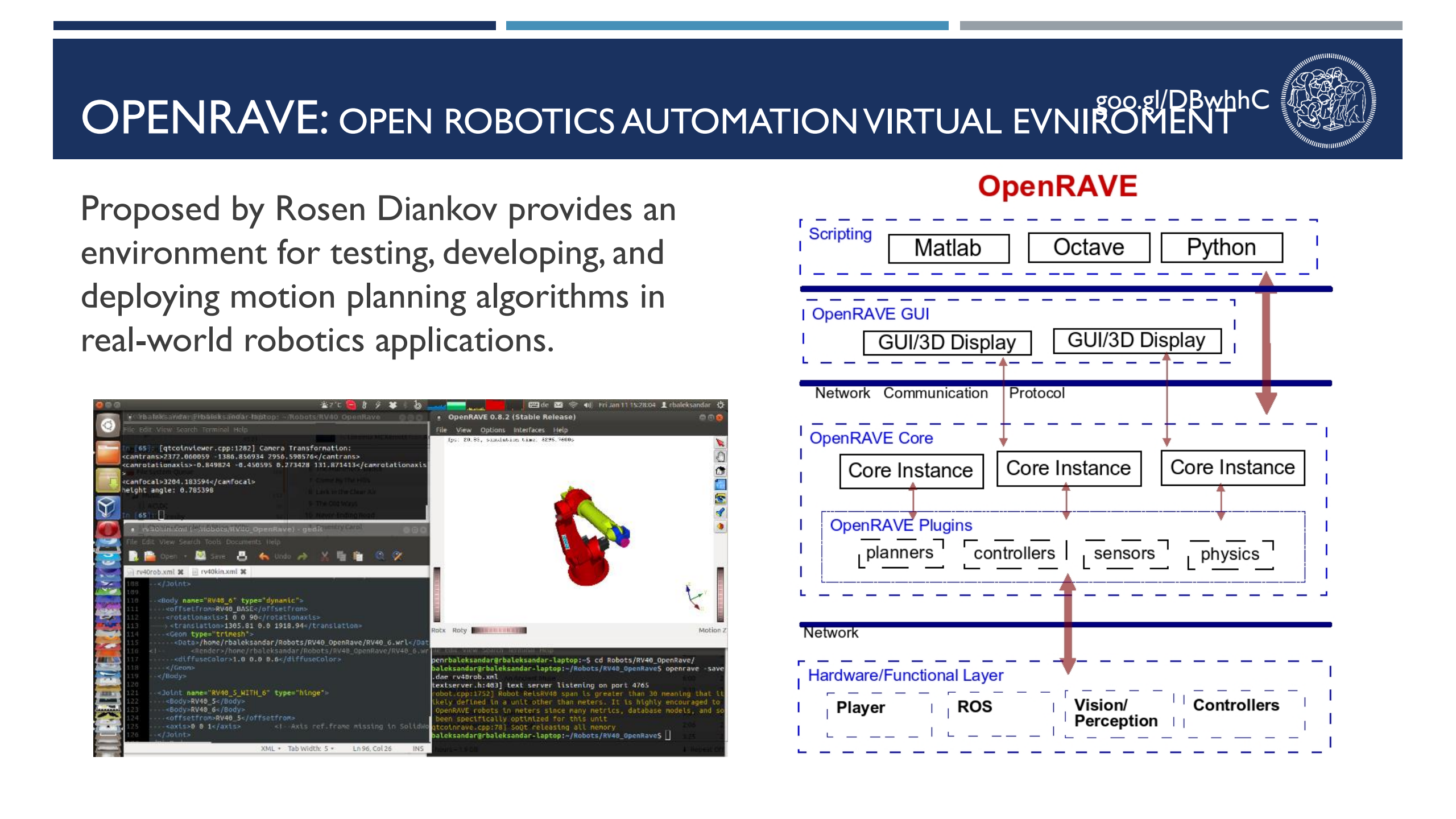
OPENRTM-AIST

goo.gl/DBwhhC



OPENRAVE: OPEN ROBOTICS AUTOMATION VIRTUAL EVNIROMENT

goo.gl/DBwhhC



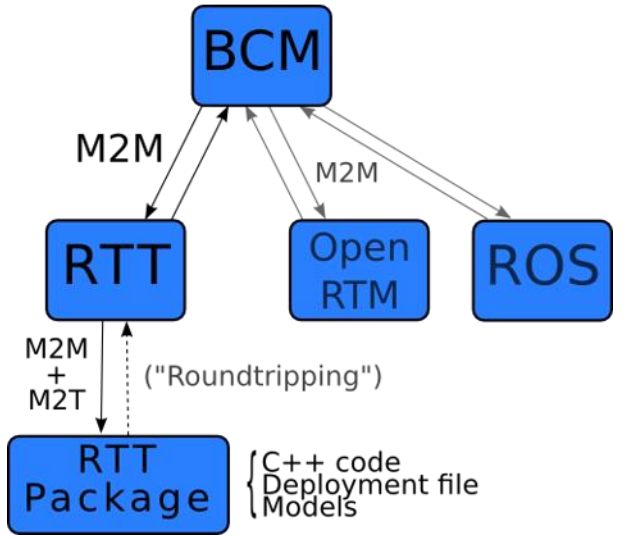
Proposed by Rosen Diankov provides an environment for testing, developing, and deploying motion planning algorithms in real-world robotics applications.

BRICS: BEST PRACTICES IN ROBOTICS

goo.gl/DBwhhC

Aimed at find out the "best practices" in the developing of the robotic systems:

- Investigate the weakness of robotic projects
- Investigates the integration between hardware & software
- Promote model driven engineering in robot development
- Design an Integrated Development Environment for robotic projects (BRIDE)
- Define showcases for the evaluation of project robustness with respect to BRICS principles.

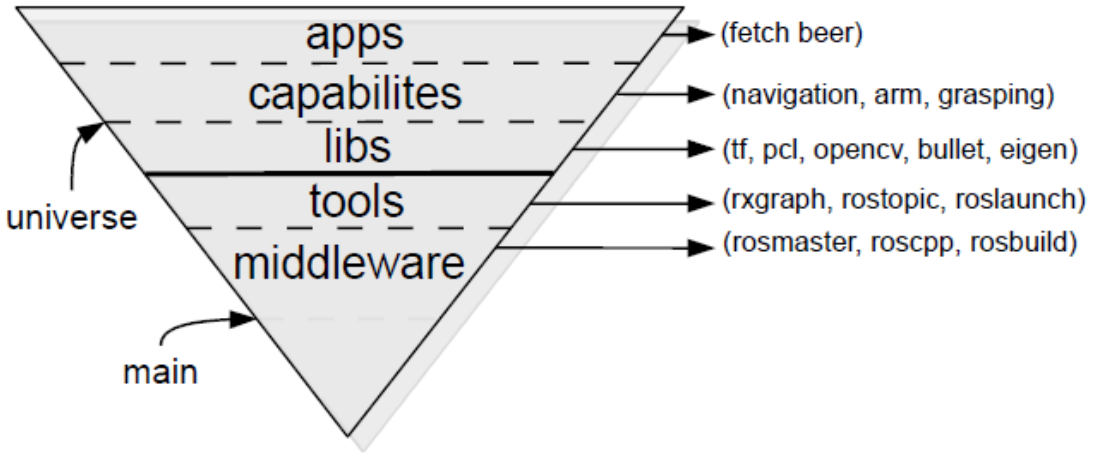


"The prime objective of BRICS is to structure and formalize the robot development process itself and to provide tools, models, and functional libraries, which help accelerating this process significantly."

ROS: ROBOT OPERATING SYSTEM

goo.gl/DBwhhC

Presented in 2009 by Willow Garage, is a meta-operating system for robotics with a rich ecosystem of tools and programs



CONCLUSIONS

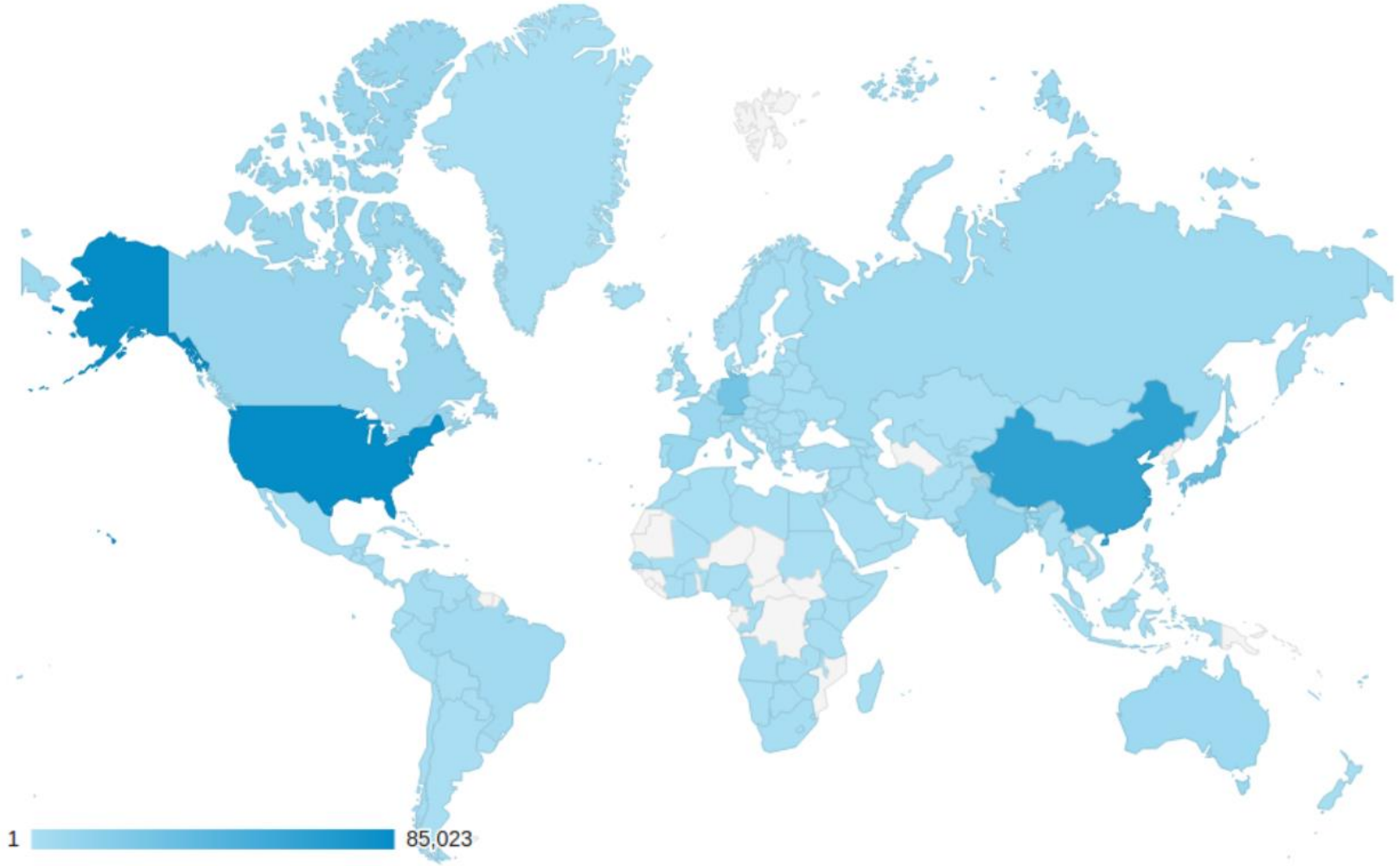
goo.gl/DBwhhC

Middlewares in Robotics :

- Are widely used
- Component-based
- Based on asynchronous communication
- Implement some form of messages exchange architecture
- Support different robot architectures (PR2, NAO, AIBO, ROOMBA, iCUB, etc..)
- Provide libraries of existing components
- Way too many…

WHY ROS?

goo.gl/DBwhhC



ROS has grown to include a large community of users worldwide

The community of developer is one of the most important characteristics of ROS

A LOT OF RESOURCES

goo.gl/DBwhhC

ROS Wiki

- Archive for the existing ROS component
- Installation and configuration guides
- Information about the middleware itself
- Lots of tutorials

ROS Q&A

- For specific problems
- Thousand of already answered questions
- Active community
- Like *Stack Overflow* for ROS

SOME NUMBERS

goo.gl/DBwhhC

ROS wiki:

- pages: 17058
- edits: 14,7/day
- views: 44794/day

ROS Q&A:

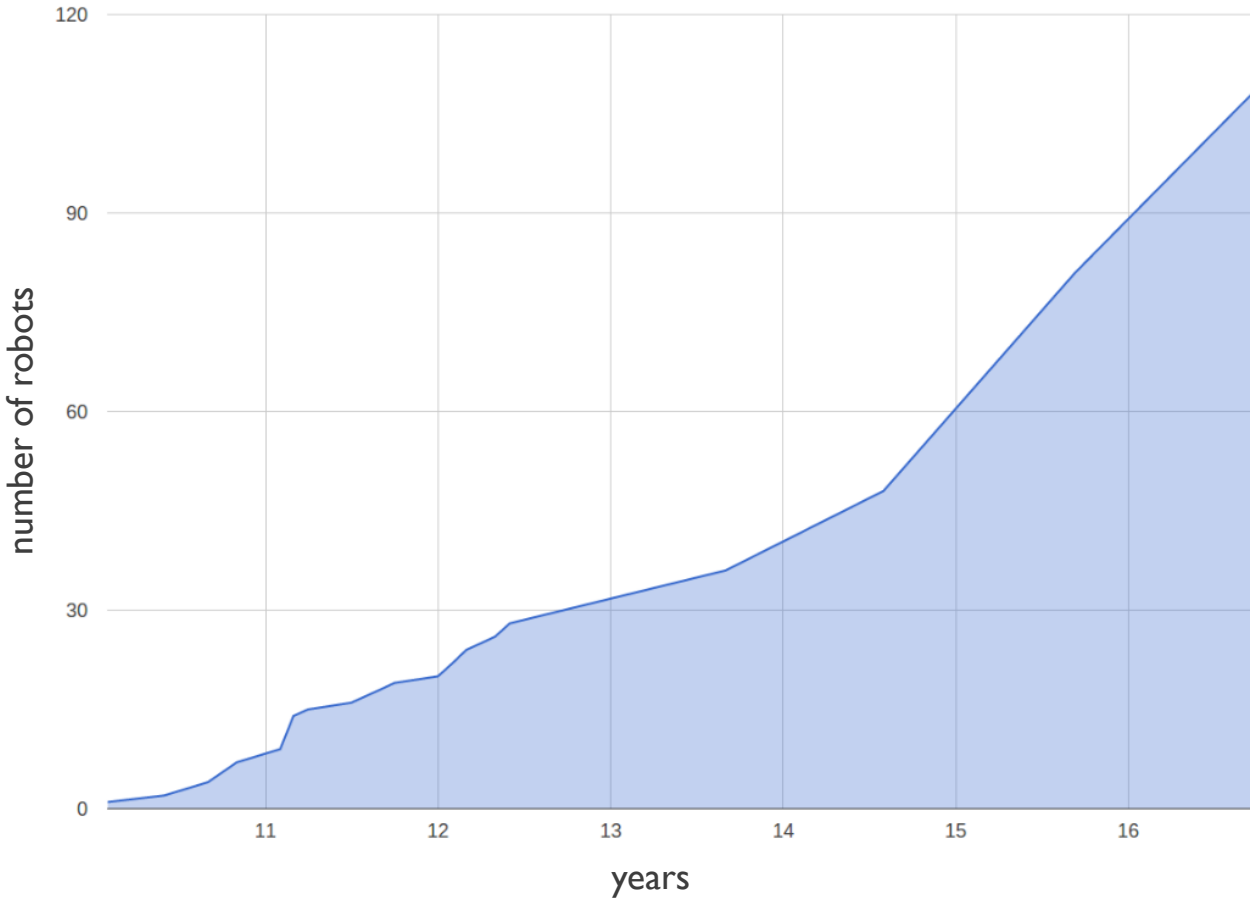
- total Q: 30243
- total A: 21697
- avg Q: 17,2/day

ROS deb:

- total DL: 8441279
- unique DL: 7582
- unique IP: 113345

ROBOT AND RESEARCH

goo.gl/DBwhhC



Total number of papers citing *ROS: an open-source Robot Operating System* (Quigley et al., 2009)

2683 (+46%)