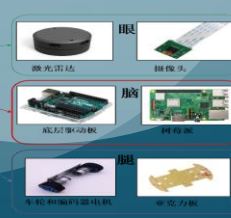
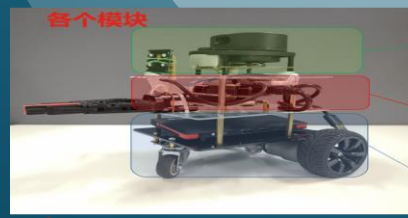
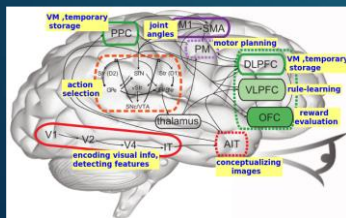
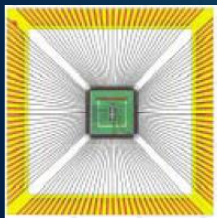


ROS机器人操作系统简介

杰克科技中央研究院



何为ROS?

ROS提供了一个管理控制机器人系统的软件框架，起到软件中间件的作用，即连接整合其他软件成分或应用，达到管理机器人系统的目标。

中间件特点：

- 可移植性：提供了一个跨语言和平台的共同编程模式
 - 可靠性：已经若干应用验证了的，经独立开发和测试的系统
 - 可控复杂度：底层软硬件由兼容了各类操作系统及硬件的专用库程序管理
- ◆ 机器人通常是一个高度复杂系统，集成了如各类传感器、马达等硬件和规划、控制算法等软件
 - ◆ 使用机器人中间件，可以为不同软硬部件之间交换数据提供一个通用管道；同时也为不同的应用间提供了同一个接口
 - ◆ 方便在不同项目之间共享、复用代码

ROS 版本: <http://wiki.ros.org/Distributions>

何为ROS（续）？

ROS (Robot Operating System) provides libraries and tools to help software developers create robot applications. It provides hardware abstraction, device drivers, libraries, visualizers, message-passing, package management, and more. ROS is licensed under an open source, BSD license. -- from wiki.ros.org

ROS提供了一个硬件抽象层：

- 设备驱动、各类软件库、可视化、消息传递、功能包管理，等等
- ROS为开源软件框架，采用BSD许可证



ROS = 您的STEM项目计划 + 开发工具 + 开发能力 + 开发生态系统

那么，ROS能够帮我们解决什么具体问题？

ROS可帮助解决什么问题？

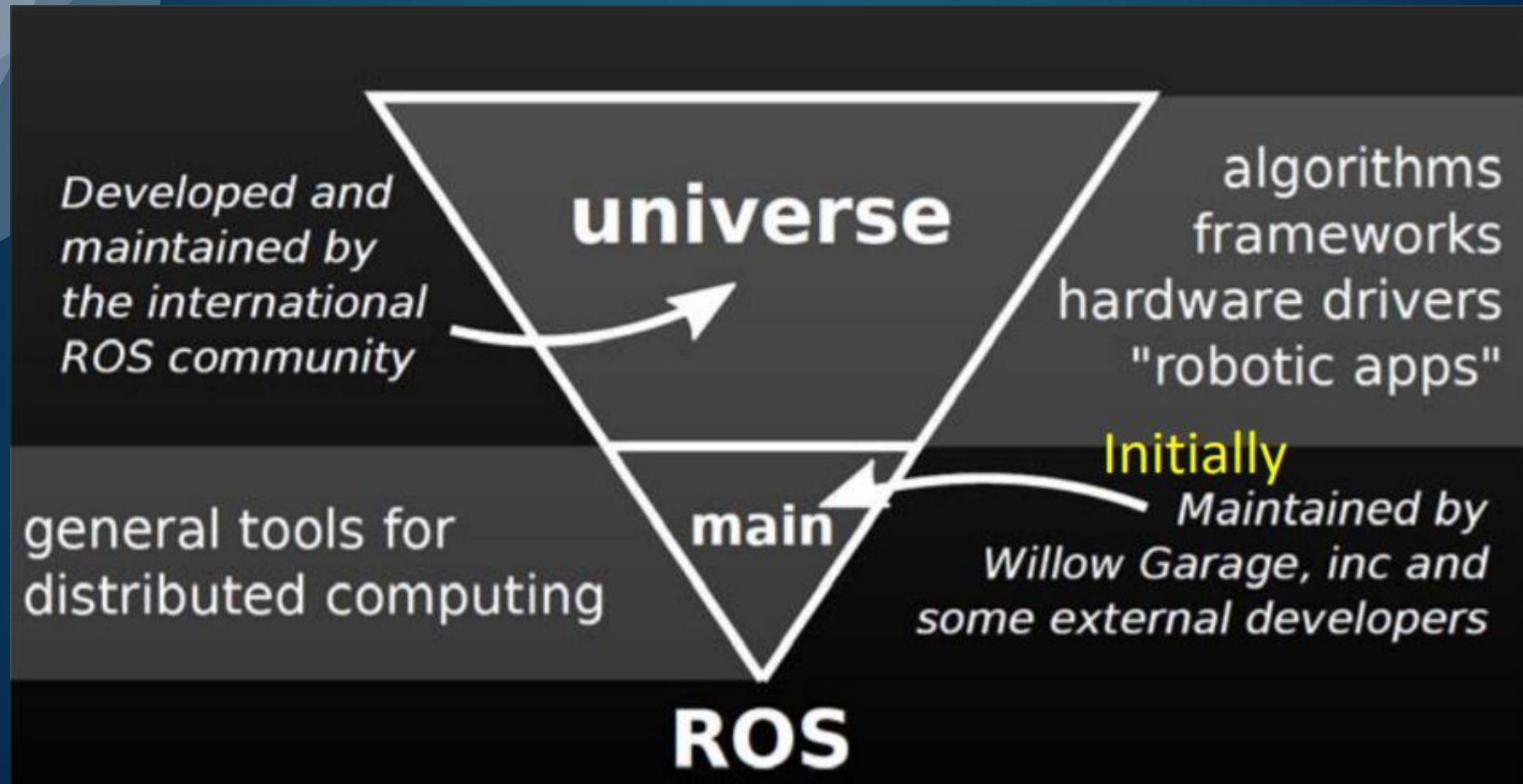
- 分布计算问题：现代众多复杂机器人项目依赖于跨程序、跨处理器及跨平台的软件系统的协同操作。
 - ROS解决了跨平台运行的不同处理程序之间的信息交互问题
 - ROS框架提供的消息传递机制已成为智能机器人软件协作的实际（**de facto**）工业标准
- 软件复用问题：随着智能机器人研发的飞速发展，出现了越来越多好的导航、运动控制与规划、建图等算法可供不同项目共享使用，而ROS的标准库和功能包则集成了很多此类算法，并经社区广泛使用，确保算法稳定性，免于逻辑或程序错误。
- 高效验证问题：机器人系统的测试验证通常会花费一个开发团队大量的时间，且易出错。
 - ROS提供机器人硬件系统的模拟器，方便工程师仅通过调试顶层应用程序（如决策、控制等）高效实现系统验证。
 - ROS也提供了记录回放各类传感器所记录数据的方法

总之，真实机器人、模拟系统及数据记录回放机制具有相同的接口机制

什么不属于ROS的范畴？

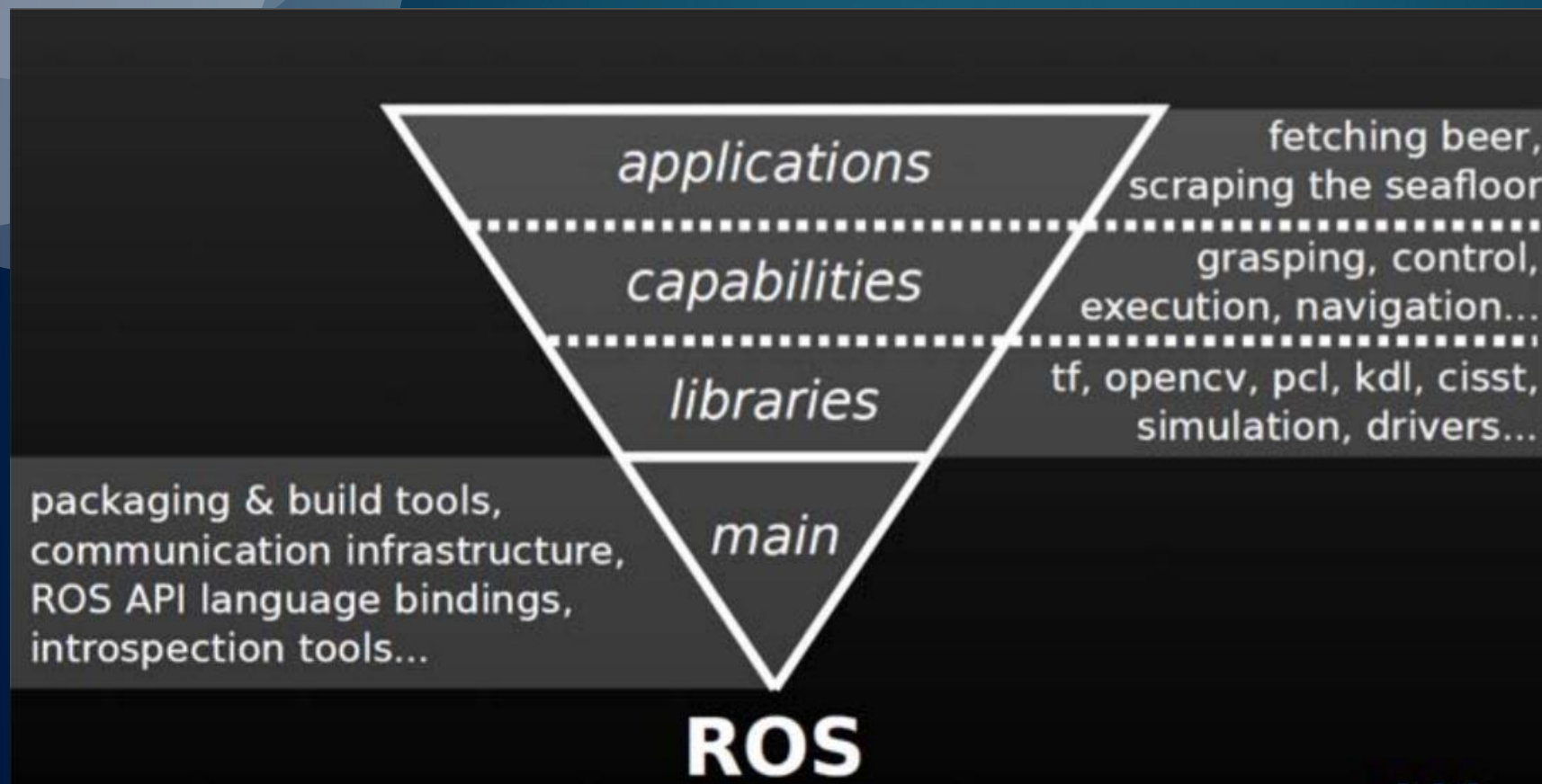
- ◆ **ROS不是一个实际的操作系统：** 尽管ROS提供了一个操作系统所提供的服务，如硬件抽象化、底层设备控制、常用功能模块化实现、功能包管理及消息传输机制，但它不是一个操作系统
- ◆ **ROS不是一个编程语言：** ROS程序由C++、Python等语言编写
- ◆ **ROS不是一个编程环境/IDE：** ROS程序通常可采用普通文本编辑器或命令行编写，不必使用IDE
- ◆ **ROS不是一个严格的实时系统：** 尽管ROS系统中，程序运行速度很快，可满足绝大多数机器人的实时操控需求，但不严格保证操作的实时性，不适用于对实时性要求极高的系统

ROS开发层级



ROS目前由OpenRobotics机构维护：
<https://www.openrobotics.org/>

ROS开发层级（续）



ROS核心要素:

1. 信息交换设施
2. 机器人相关功能包
3. 各类机器人开发调试工具