openEuler Embedded介绍以及最新进展

openEuler : 面向数字基础设施的开源操作系统

openEuler 是?

openEuler 是由开放原子开源基金会
（OpenAtom Foundation）孵化及运营的开源项目

openEuler 愿景

为世界提供数字基础设施的开源操作系统

openEuler 使命

为世界提供开源软件
为开源软件提供指导、虚拟协作空间、创新平台和服务
在社区开发、管理和孵化开源软件，并且与其他许多开源社区合作

谁在贡献 openEuler

100+
SIG组

970+
成员单位

14,900+
社区贡献者

战略捐赠人 HUAWEI

白金捐赠人 麒麟信安 KYLINSEC、KYLINSOFT 麒麟软件、统信软件技术有限公司

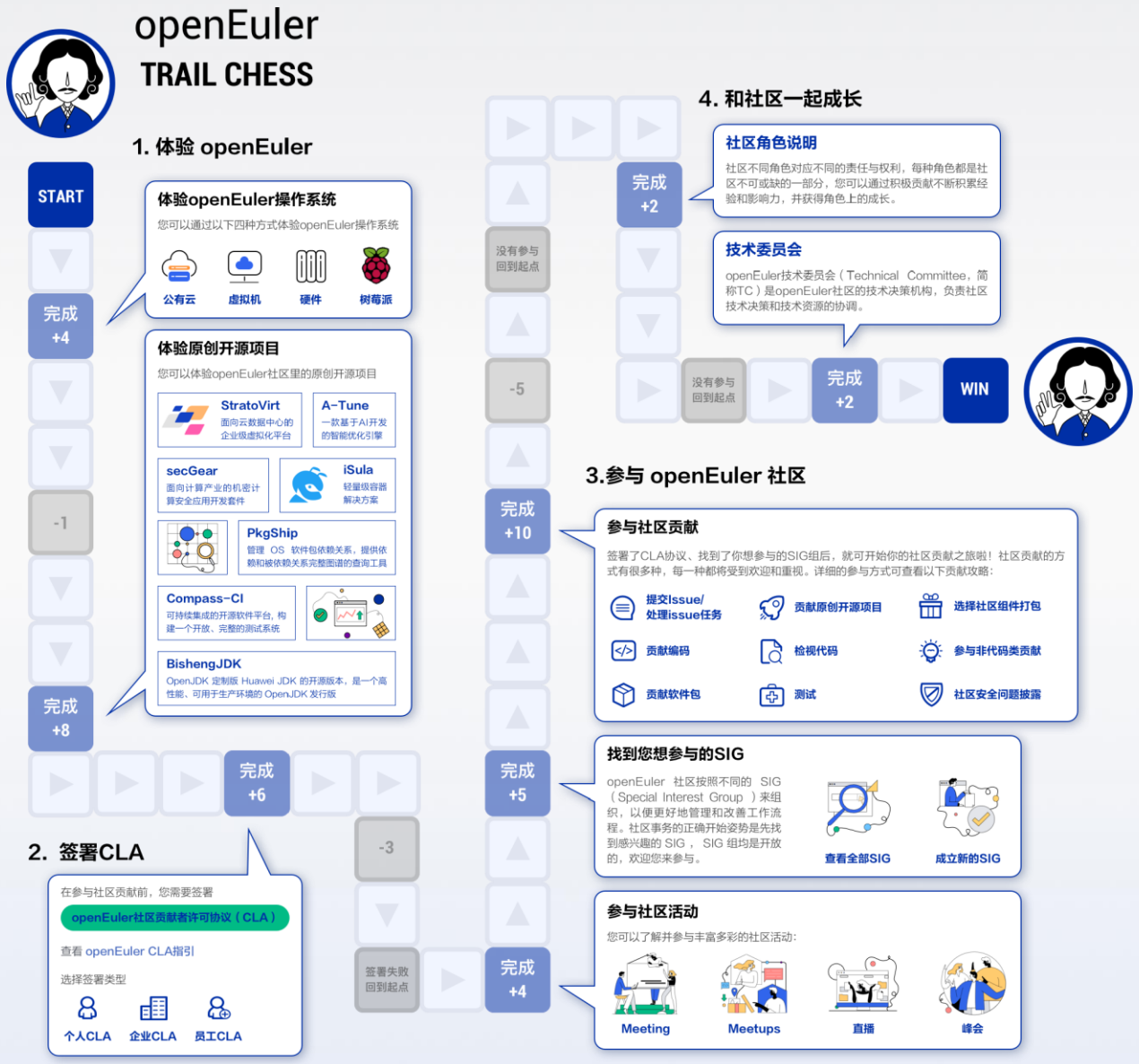
黄金捐赠人 FUSION、HOPERUN 润和软件、SOFTSTONE 软通动力、天翼云、China unicom 中国联通、Unicom Digital Tech 联通数科、中国移动 China Mobile

白银捐赠人 H3C、turbolinux、CS&S 中软国际

青铜捐赠人 BES 宝兰德 BESSYSTEM、Phytium 飞腾、CETC 普华基础软件、中盈优创 China Unitechs、Aspire 卓望

学术机构和非营利组织 ISCAS

加入 openEuler 社区



立即体验

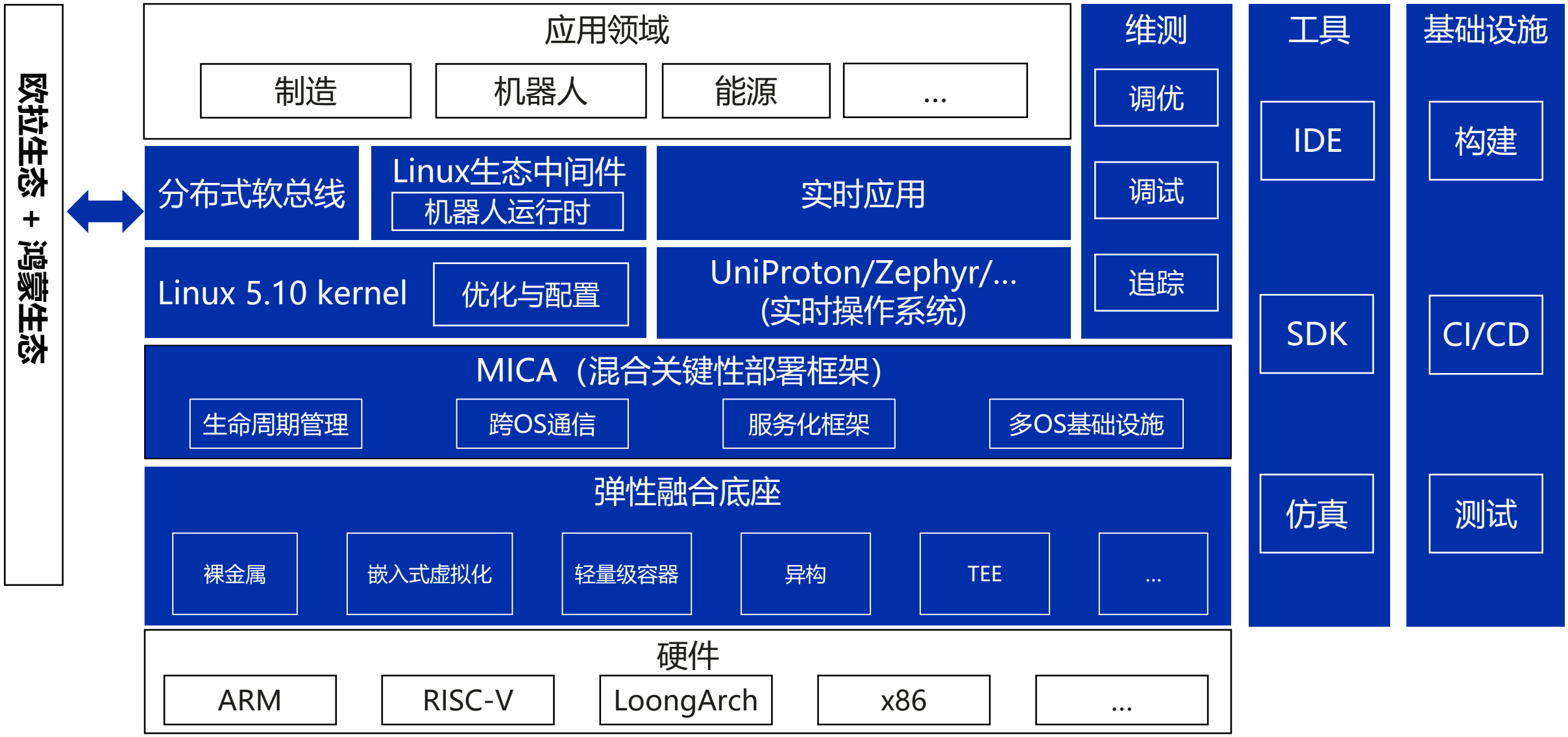
1 openEuler Embedded介绍及进展

2 新特性：嵌入式ROS2运行时介绍

openEuler Embedded
以Linux为中心的开放综合嵌入式系统软件平台

- openEuler Embedded适用于任何**需要Linux的嵌入式系统**
- Linux作为整个星系的中心，提供**丰富生态与功能, 混合关键性系统，分布式软总线，基础设施**等特性吸引其他行星
- 不同的行星提供各具特色的生态: **硬实时(实时操作系统)，信息安全 (TEE)，极致性能(裸金属)，混合关键性(嵌入式虚拟化)**

openEuler Embedded总体架构



openEuler Embedded关键技术特性（1+X+1）

OpenEuler

Linux基础框架

依托openEuler和华为嵌入式能力积累打造高质量嵌入式Linux基础框架

容器 | 图形 | 包管理 | 追踪

核心软件包
(C库, busybox, utils,…)

调优

调试

面向嵌入式场景的优化
(PREEMPT-RT, 裁剪配置)

高质量Linux内核
(5.10, 6.x, CVE)

不断孵化中的新的特性

面对新的技术、新的趋势、新的场景不断孵化新的特性

可信 | 更好的南北向生态 | 嵌入式AI | RUST | ……

轻量机器人运行时

面向机器人场景（工业/智能）构建实时、可靠的轻量级机器人运行时

ROS2 | 工业机器人软件栈 | micro-ROS | 定制化 | 实时

混合关键性系统

以openEuler Embedded为中心实现多运行时高效混合部署,同时满足功能、实时、安全等多目标

通信 | 服务化 | 隔离 | 管理

混合关键性部署 | 弹性融合底座

UniProton

分布式软总线

实现多节点跨通信介质高效互联，分布式能力共享，互通鸿蒙、欧拉生态，达到“1+1>2”效果

分布式能力集

组网 | 发现 | 连接

传输

认证 | 系统参数 | IPC

基础设施

依托openEuler构建强大的基础设施，保障QA，易用性，生态，增强用户粘性

文档 | 课程

SDK | 仿真

嵌入式构建系统
oebuild

CI&CD

门禁

构建

测试

Compass CI | 代码管理

首批基于 openEuler 嵌入式能力的商业发行版发布

OpenEuler

菁蓉联创

中天鲲鹏操作系统 V1.0（欧拉版）

GBA NCTI 粤港澳大湾区国家技术创新中心 GREATER BAY AREA NATIONAL CENTER OF TECHNOLOGY INNOVATION

工业嵌入式操作系统V1.0

麒麟信安 KYLINSEC

麒安嵌入式操作系统V3

openEuler 23.03 Embedded

openEuler Embedded 23.09

OpenEuler

基础设施

- **yocto升级4.0 LTS**
- oebuild完善：基线快照, 镜像定制，快速运行
- CI/CD: sstate cache镜像，compass CI对接
- 测试框架: 测试套完善、测试框架

Linux框架

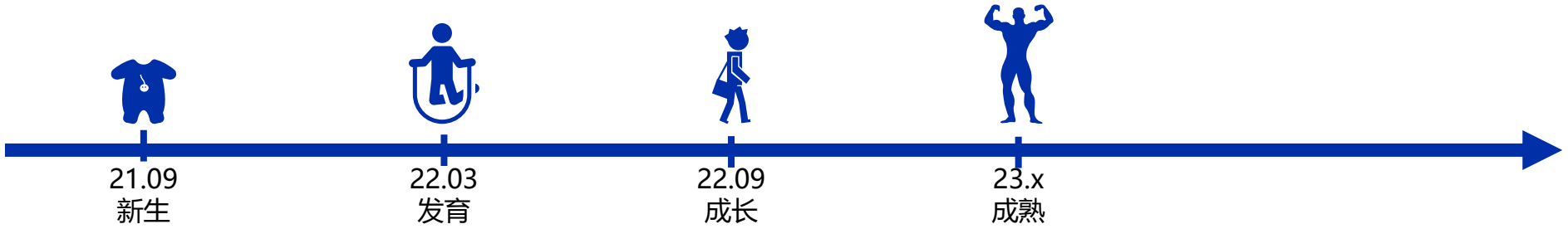
- 内核与社区紧密保持同步，为6.6内核做准备
- 竞争力构建：**小型化、快速启动、Preempt-RT优化**
- 总体软件包数量**400+**
- 嵌入式图形，包管理，systemd, python的优化
- Multi-lib、Multi-toolchain

关键特性

- 混合关键性框架（MICA)
  - System DTS的完善
  - **虚拟化、openAMP的融合**
  - 多OS、多工具链的融合
  - **UniProton，跨OS调试**
- 嵌入式弹性底座的探索
  - **Jailhouse的完善**
  - **轻量级容器(iSula)全面支持**
- **轻量级机器人运行时**
  - **ROS2 Humble**
  - 定制SDK

南北向生态

- 北向生态
  - 工业现场总线
  - PLC运行时
  - Demo
- 南向BSP
  - **更完善的RISCV支持**
  - ARM64: **RK3588**， Hi3093,...
  - ARM32:



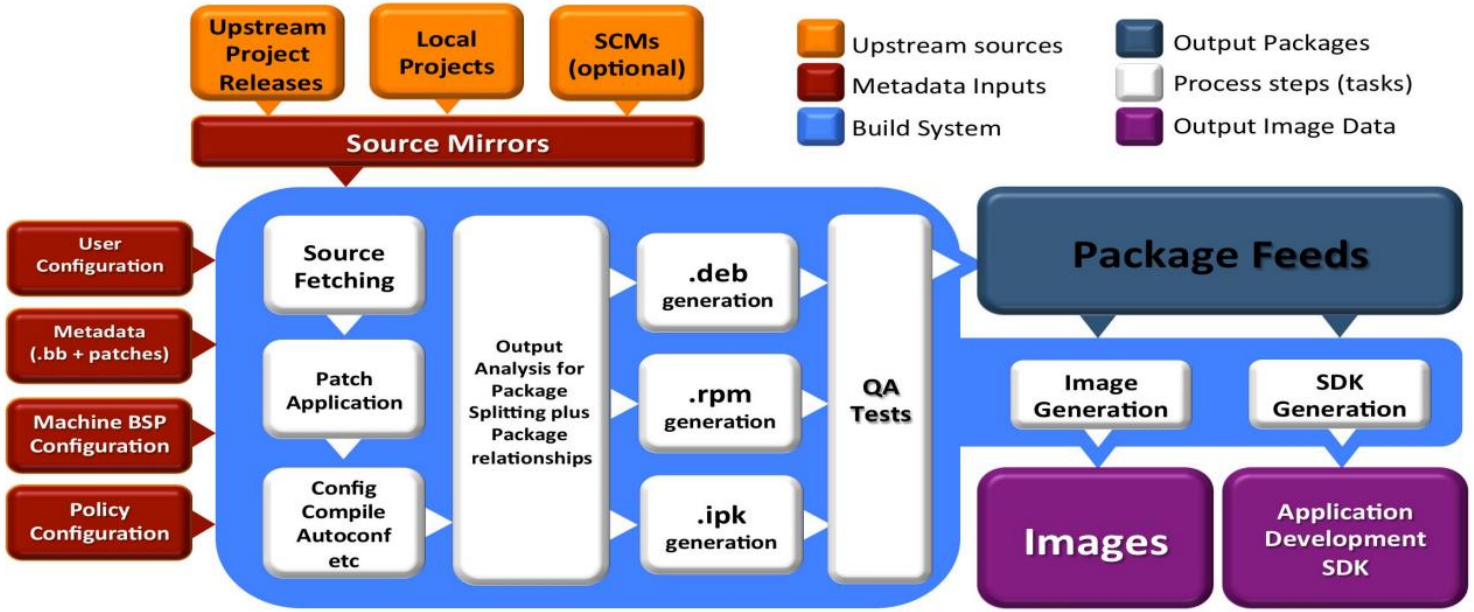
openEuler Embedded基础设施核心：Yocto Project

Yocto Project：为构建定制化的嵌入式Linux发行版提供一系列模板、工具和方法

- 不是一个Linux发行版，而是用来构建一个定制化的Linux发行版
- 不只是一个开源项目，而是一个开发生态

主要特点：

- **广泛应用于整个行业**：Intel, Xilinx, Huawei, Simens, ARM, Microsoft
- **多种架构支持**：arm, x86, riscv, MIPS
- **镜像和代码移植容易**：直接修改脚本
- **灵活**：基于Python带来无限可能
- **面向嵌入式和物联网设备**：裁剪和定制化
- **执行机制大于策略**：bb文件由你定
- **支持层模型**：可以复用别人的成果
- **支持部分组件构建**：选择你想要的组件
- **丰富的个人组织和生态**：meta-xxx
- **二进制可再现性**：拿到配方，既可复现
- **许可证清单**：生成各个组件的License



Yocto架构与构建流程

openEuler Embedded选择Yocto的理由

	Yocto	Buildroot	Openwrt	OBS
面向领域	泛嵌入式 (AGL, WRLinux)	嵌入式设备的固件	网络设备(路由器)	服务器、PC
交叉编译	支持	支持	支持	不支持
支持包的数量	8000+	2500+	3500+	10000+
可扩展性	优(python)	一般	一般	差
镜像大小/构建速度	场景依赖, 速度适中	小/快	小/快	大/慢
包管理	支持(rpm,deb,ipk)	不支持	支持(ipk)	支持
分布式构建	不支持	不支持	不支持	支持

Buildroot → 功能更强大，易扩展 → Yocto ← 易扩展,交叉编译 ← OBS

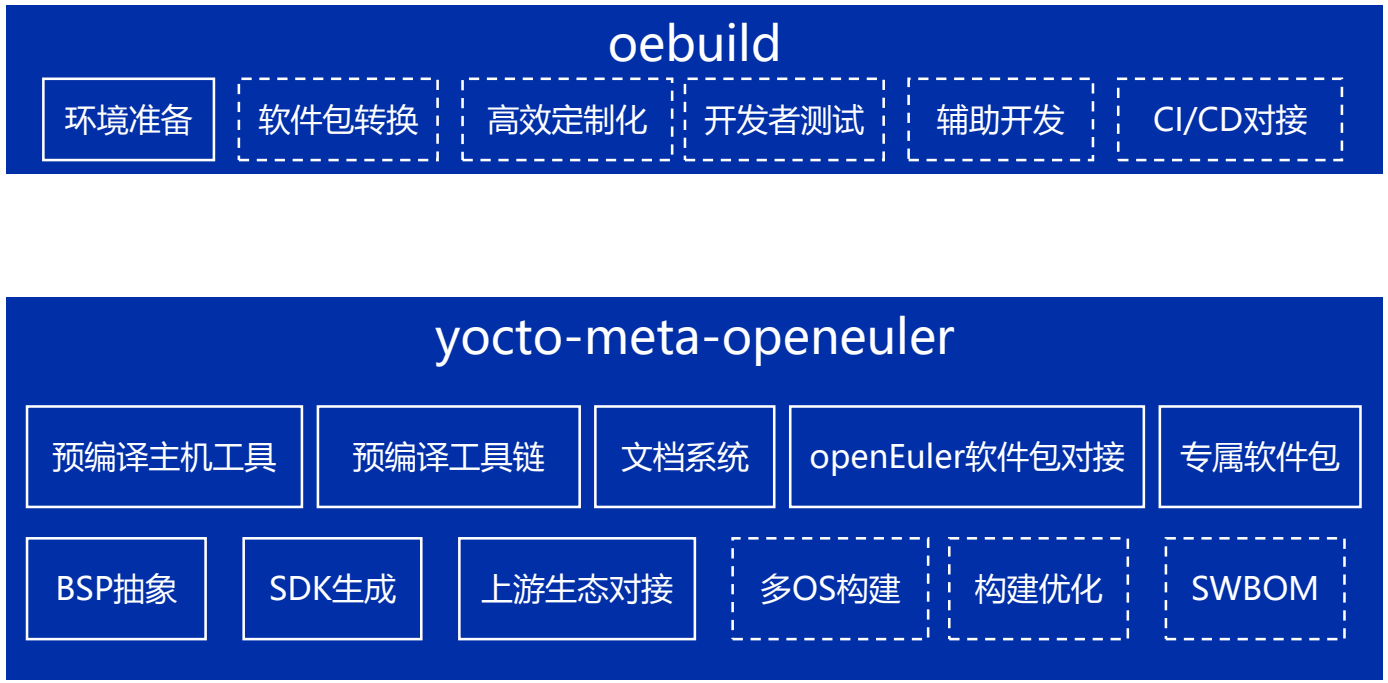
- 适合构建简单的镜像，不适合嵌入式发行版
- 没有包管理系统
- Makefile可扩展性不足、复杂操作实现困难
- Kconfig虽然简单，但表达性不足

- 基于python，获得了强大基础生态和灵活性
- 面向构建的DSL, 提升构建的效率
- recipe和layer的设计，使能代码复用
- 业界的广泛支持：Huawei, Intel, ARM...
- **缺点**:
  - 学习曲线陡峭
  - 过于灵活
  - 不支持分布式构建

- 构建环境搭建复杂，个人难以使用
- 不支持交叉编译，无法满足嵌入式场景需求
- 不支持细粒度的裁剪和构建，无法满足嵌入式场景需求

Yocto十分适合用于构建嵌入式Linux发行版

openEuler Embedded构建系统架构

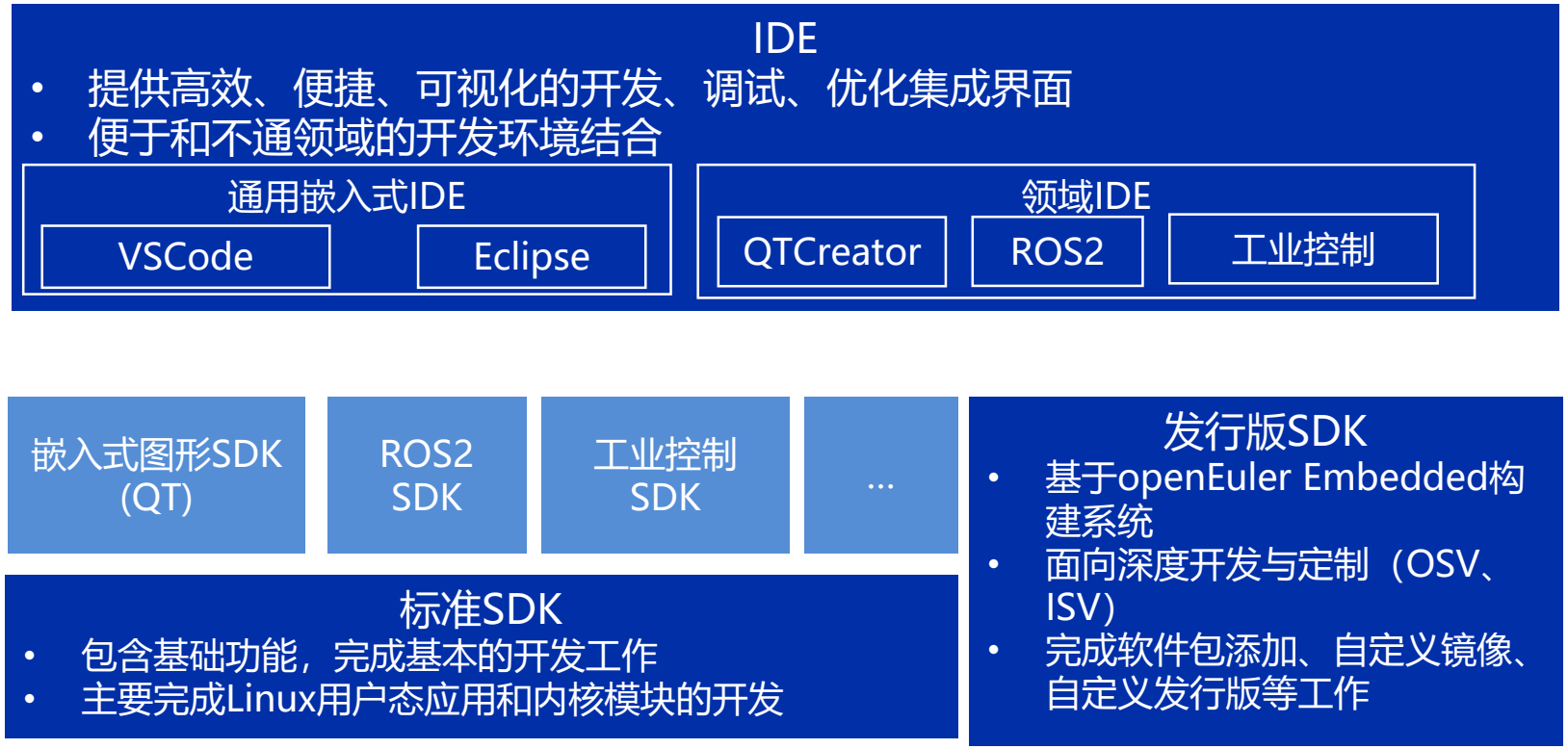


以python为主辅以shell的高效开发者工具，以实现简化yocto配置与开发，使能高效定制，多流程集成等功能

openEuler Embedded核心构建层，既兼容上游社区，又坚持自身特点：软件包同源，预编译工具，容器化构建，多OS构建

充分利用yocto/openembedded生态，避免无意义的重复造轮子

openEuler Embedded的工具体系结构



- IDE层(View):提供高效的可视化界面，不同的应用领域有专门的IDE
- SDK层(Controller)：核心工具和组件的融合，基于CLI可以完成全部开发工作，不同领域会有不同领域的SDK
- 单点工具层（Model)：包含原子性的单点工具和基本组件

分层抽象使整个架构非常灵活，有利于解耦、扩展、替换

灵活、高效、细粒度、端到端的分层定制能力

OpenEuler

- 镜像：镜像是领域OS最终体现，针对不同应用，不同场景，不同要求定制专用镜像，例如实时镜像，ROS2镜像，PLC镜像，容器镜像
- 软件包组：相关的软件包归纳成组，例如QT相关的软件包，ssh相关的软件包，DFX相关的软件包，不同软件包组件可能会有重复，同一个软件包组可以不同镜像复用
- 软件包：软件包细粒度的定制，运行时与开发文件分离，最小粒度可以到单个文件，自定义子软件包，
- 内核：按照需求定制化内核，内核kconfig，内核代码补丁，内核模块，最终形成不同的内核软件包
- 工具：定制化工具链，SDK，不同编译器(gcc,llvm)，不同的C库(glibc, muslc)，不同的组合

1 openEuler Embedded介绍

2 新特性：嵌入式ROS2运行时介绍

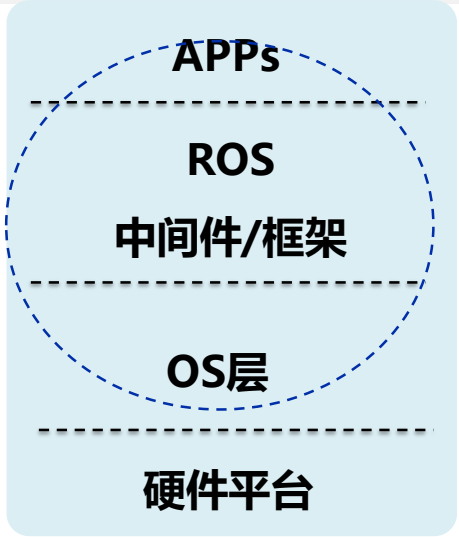
什么是ROS

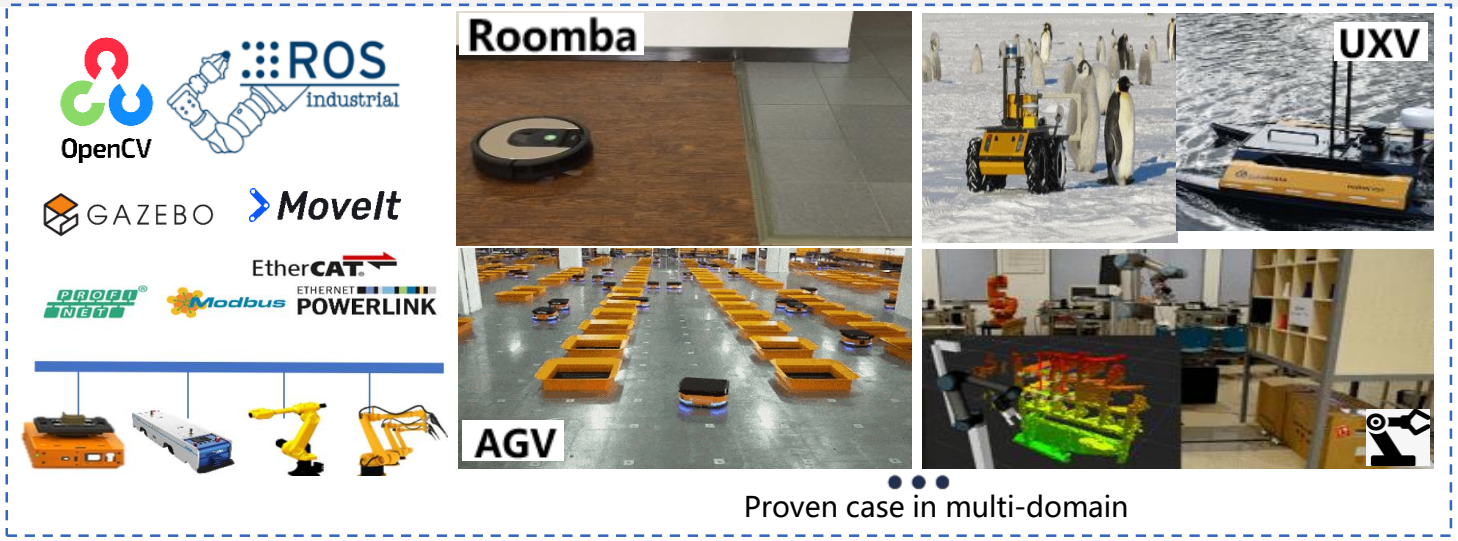
ROS（Robot Operating System）

计算机角度看ROS：底层基础OS之上的一套开源友好的机器人中间件/框架，与底层OS共同提供一套机器人产品开发&运行环境

核心：不重复造轮子的平台理念——**分布式、可重用、高协作**

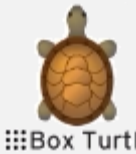
大量贡献者、组件齐全：提供了分布式通信框架，集成了从**硬件驱动**到**工具库和协议**等诸多软件，发展至今，具有大量应用案例，被用于众多领域。





ROS的诞生、发展

诞生 2007 斯坦福STAIR项目



正式开源 2010 ROS 1.0发布

科研 工业

机器人开源基金会接管 2013 OSRF接管

架构演进 2017年底 ROS 2.0发布

最后一个ROS1版本 2020 ROS Noetic发布

ROS 2 LTS版本 ~

关于ROS1和ROS2

- ROS1的假设/限制：需要良好的网络、足够快的硬件...使其适合科研、开发、演示而非实际产品
- ROS2的设计定位：面向**可靠性、嵌入式、实时、产品化**
- ROS1在2020发布最后一个版本，后面全部转向ROS2

openEuler社区面向机器人领域长期规划：逐步打造自主可控机器人操作系统（A-ROS）

OpenEuler

2022-2023	2023-2024	2025+
在openEuler社区引入&适配ROS软件包	ROS2核心包的替换	构建国内机器人操作系统社区（A-ROS）

openEuler+ROS 完整可用

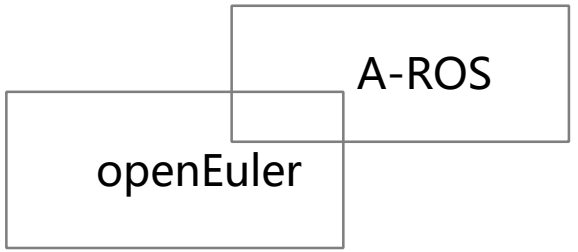
- ROS1软件包移植：1500+核心包、工具包、应用功能包、第三方功能包。
- ROS2软件包移植：900+核心包、工具包、应用功能包

对ROS2核心包做抽屉式替换和增强，满足特定行业场景要求

- DDS通信框架
- 关键软件包的替换（导航、SLAM、MoveIT等）
- RVIZ可视化工具
- Gazebo仿真工具
- ……

构建国内机器人操作系统根社区

- 与openEuler社区协同发展，共同打造国内机器人操作系统根社区

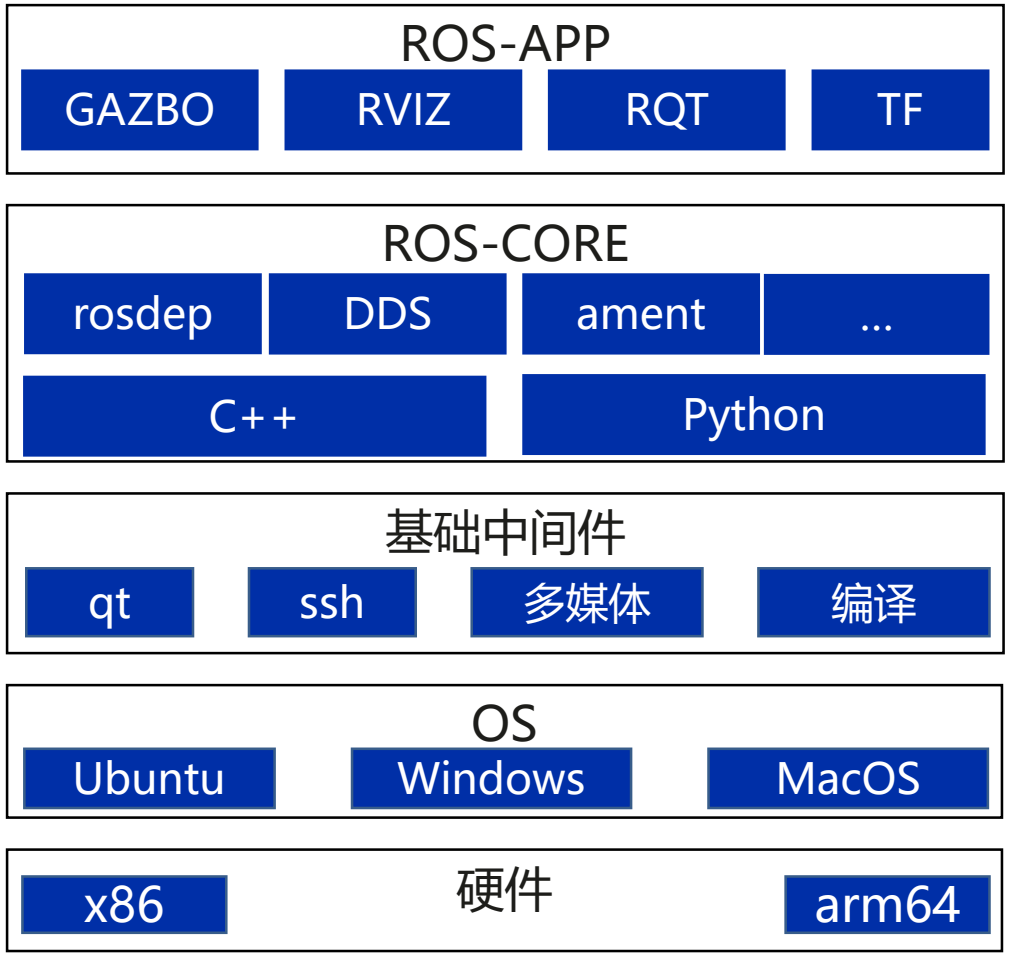


ROS2软件栈分析

OpenEuler

开发态软件栈(云、桌面、服务器)

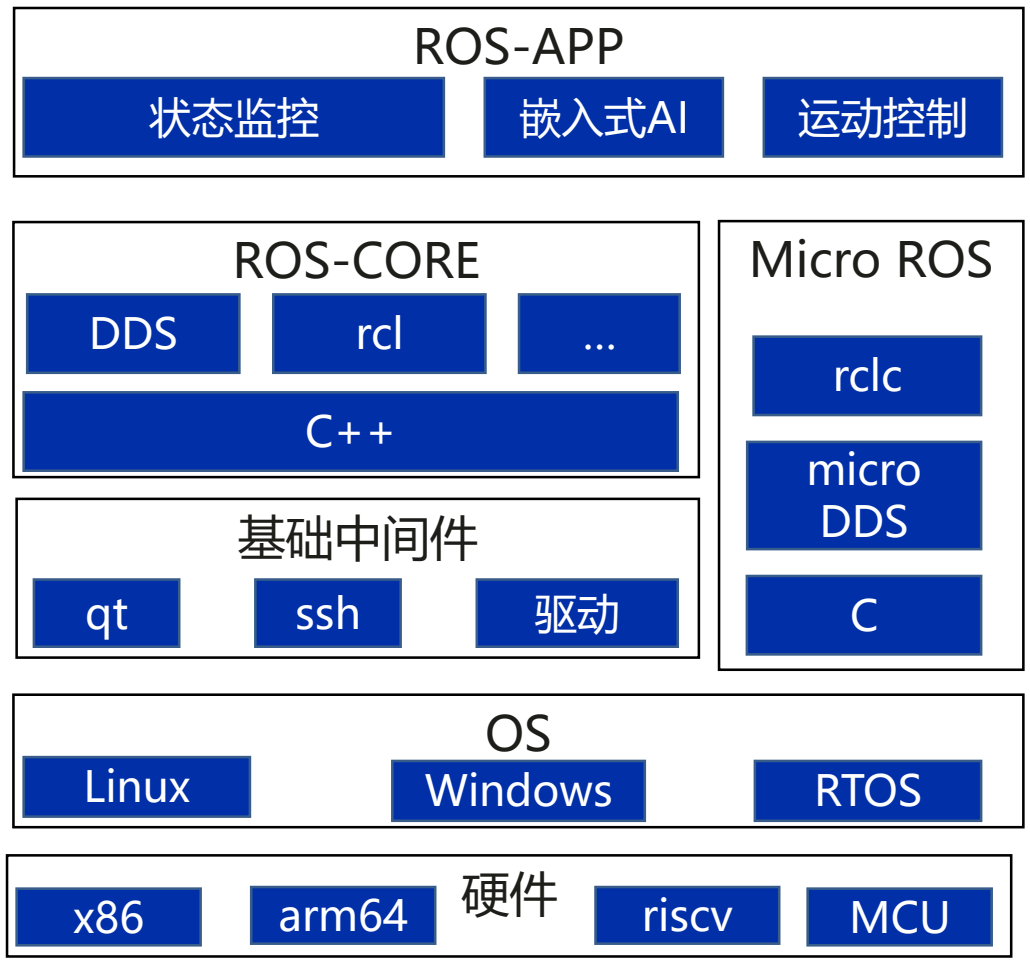
要求：易用性、高性能、图形支持、仿真支持、无明显短板



openEuler Embedded重点突破

运行态软件栈(边缘、嵌入式)

要求：高效、稳定、实时、安全、差异化竞争力



嵌入式ROS2的定位和挑战

OpenEuler

嵌入式场景特点

- 嵌入式硬件
- 有限资源，内存、算力、功耗
- 多目标约束：成本、实时、安全、可靠

场景挑战

- 多样性的硬件，迥异的架构
- 交叉编译与开发效率
- 量体裁衣，优化与定制化，满足多目标约束

业界项目情况

- 嵌入式集成

meta-ros

- **主要参与者**：LG，openEmbedded
- **目的**: 使能ROS在高度定制化的嵌入式Linux运行，而非绑定在Ubuntu之上
- **问题**：依旧存在耦合，裁剪定制难。Yocto+ROS维护门槛高、开发者有限，从2022年6月起LG不再维护meta-ROS

- 实时

ROS-RealTime

- **主要参与者**：Apex.AI, Bosch, Open Robotics, WindRiver
- **目的**: 端到端改善ROS2的实时性，主要包括：Linux with Preempt-RT，QNX/Vxwork替换Linux
- **问题**：Linux with Preempt-RT只能实现软实时性，强实时性无法保证，RT-Linux/Xenomai等方案与架构生态冲突

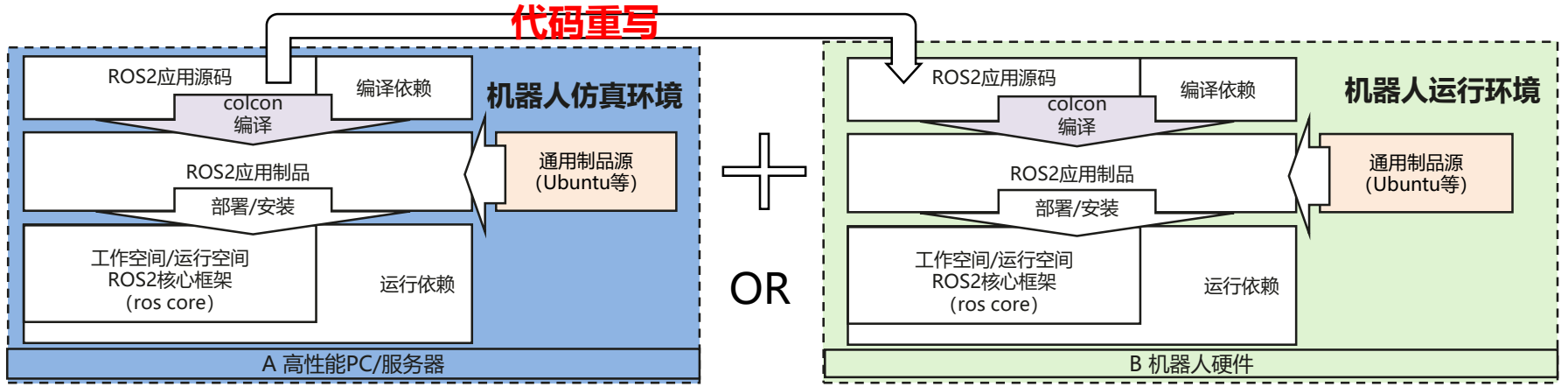
- 微型

Micro-ROS

- **主要参与者**：LG， ePROsima，Bosch, WindRiver
- **目的**：让ROS2能够运行在微控制器(实时操作系统）上，支持Zephyr/Nuttx/FreeRTOS
- **问题**：受限于RTOS的功能，Micro-ROS比较简单，往往无法独立实现任务功能

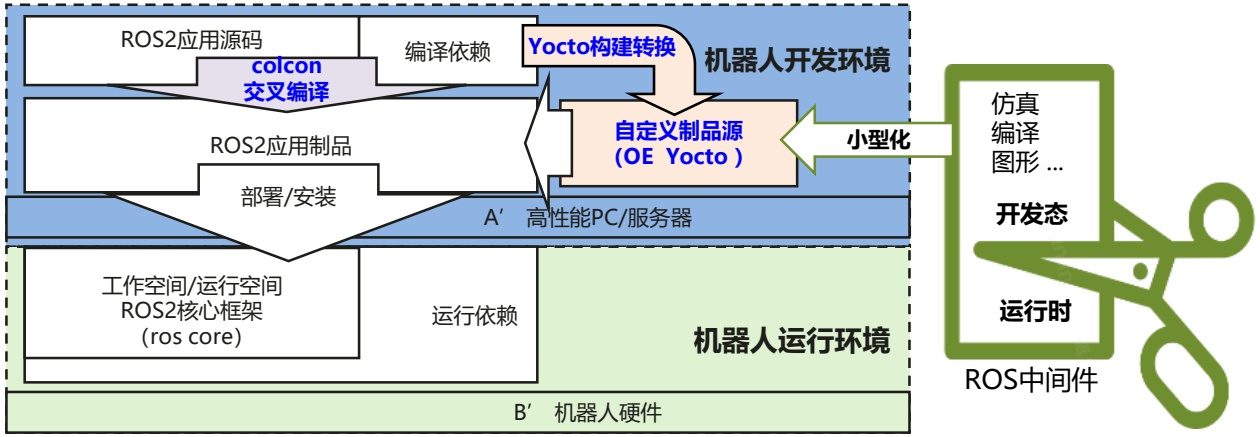
openEuler Embedded ROS2轻量级运行时

AS-IS



开发模式	说明	问题
B	直接在机器人运行环境开发	● 强绑定Ubuntu等系统，后期产品化需重新考虑裁剪镜像定制 ● 需要额外配置以满足开发编译，硬件规格成本高，编译时长不可控 ● 不支持自定义制品源（自有应用拓展难）
A+B	为加速B过程，先基于高性能PC仿真， 重写后在实际机器人环境重新编译验证	● 仿真完成后需要重写、移植到机器人环境 ● 其他问题同开发模式B

TO-BE



- 解耦ubuntu绑定，产品化裁剪定制无忧
- 机器人环境小型化（开发态、运行时解耦）
- 可拓展、自定义制品源（ROS应用转换框架/Yocto构建转换）
- 仿真、交叉编译、部署一体，减少重写移植工作（ colcon交叉编译支持）


OpenEuler