RT-Thread

嵌入式设备：

从实时操作系统到微内核操作系统

目录

01

实时操作系统

实时性，可确定性

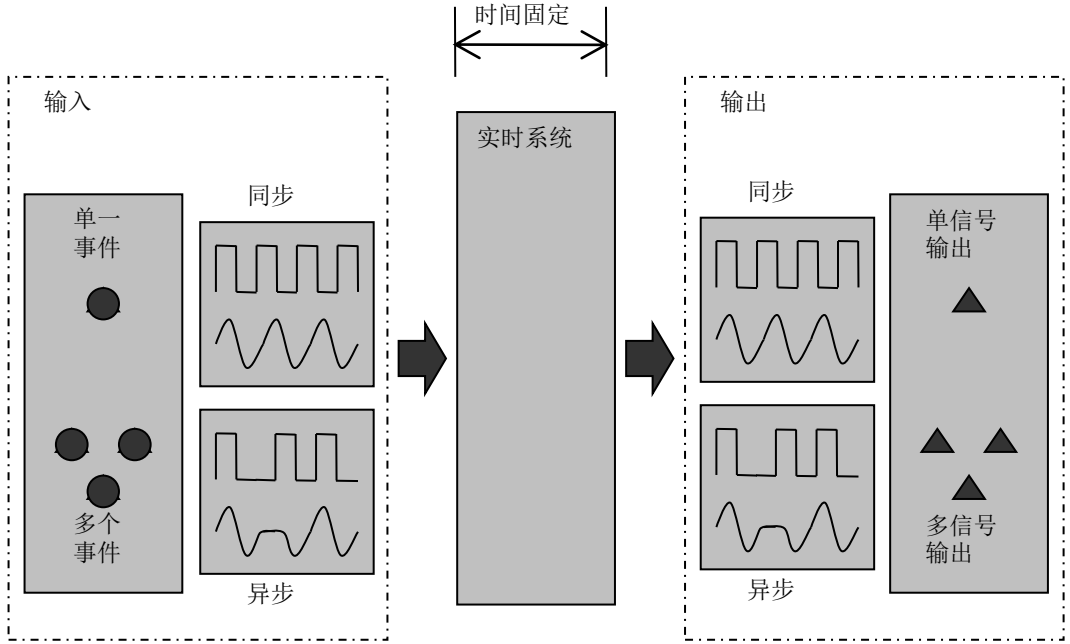
RT-Thread

实时操作系统

- 实时性，可确定性

- 小型化操作系统，多数只包括一个硬实时多任务内核：
  - 多任务调度，管理；任务间互斥、通信
  - 静态内存池，动态内存堆
  - 中断管理、定时器管理

- 大量应用于微控制器（MCU，Microcontroller Unit）中
  - 资源（Flash、SRAM）有限
  - 多用于控制场合



实时操作系统

- µC/OS-II, µC/OS-III
  - 2003年邵贝贝译《嵌入式实时操作系统µC/OS-Ⅱ》，后续近10年流行于国内

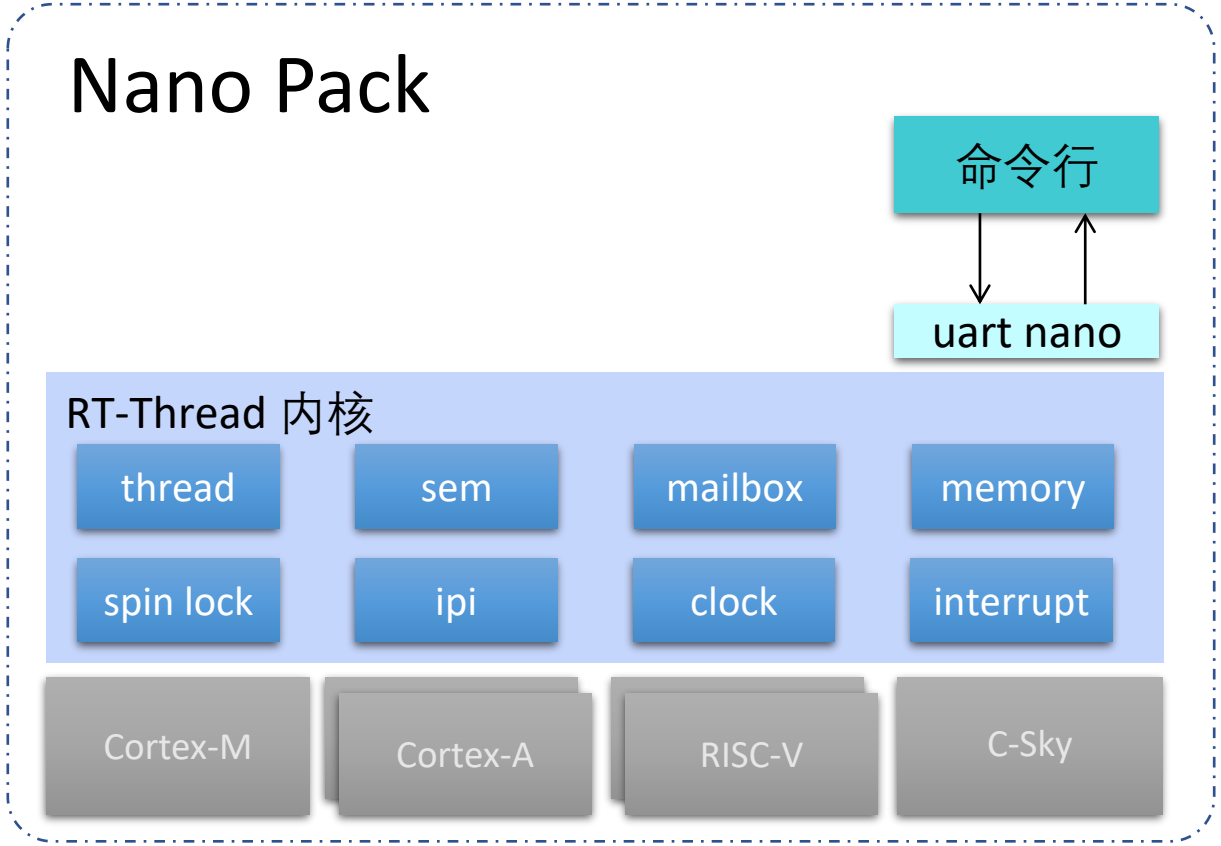
- FreeRTOS开源、许可宽松获得芯片厂商支持
  - ST、NXP、Xilinx
  - ARM Cortex-M

- RT-Thread Nano
  - 衍生自RT-Thread标准版本

RT-Thread Nano

RT-Thread Nando是面向MCU的深度低资源占用版本：

- Flash占用低至3kB Flash， RAM占用低至1.2kB；
- 获得ARM官方认可，以MDK Pack方式内置于Keil MDK中
- 携带Device API接口的UART nano驱动，无缝切换到RT-Thread标准版本；
- 丰富的IDE支持：***Keil MDK, CDK, GNU MCU Eclipse***



02

嵌入式操作系统

功能性、网络特性

RT-Thread

嵌入式操作系统

- 面向嵌入式系统（专用目的）的操作系统
  - 一般也是实时系统（优先级系统）

- 强调功能性的专用性操作系统:
  - 实时内核
  - 文件系统（数据存储）
  - 网络协议栈
  - POSIX API接口，C++编程
  - USB stack
  - 人机交互命令行

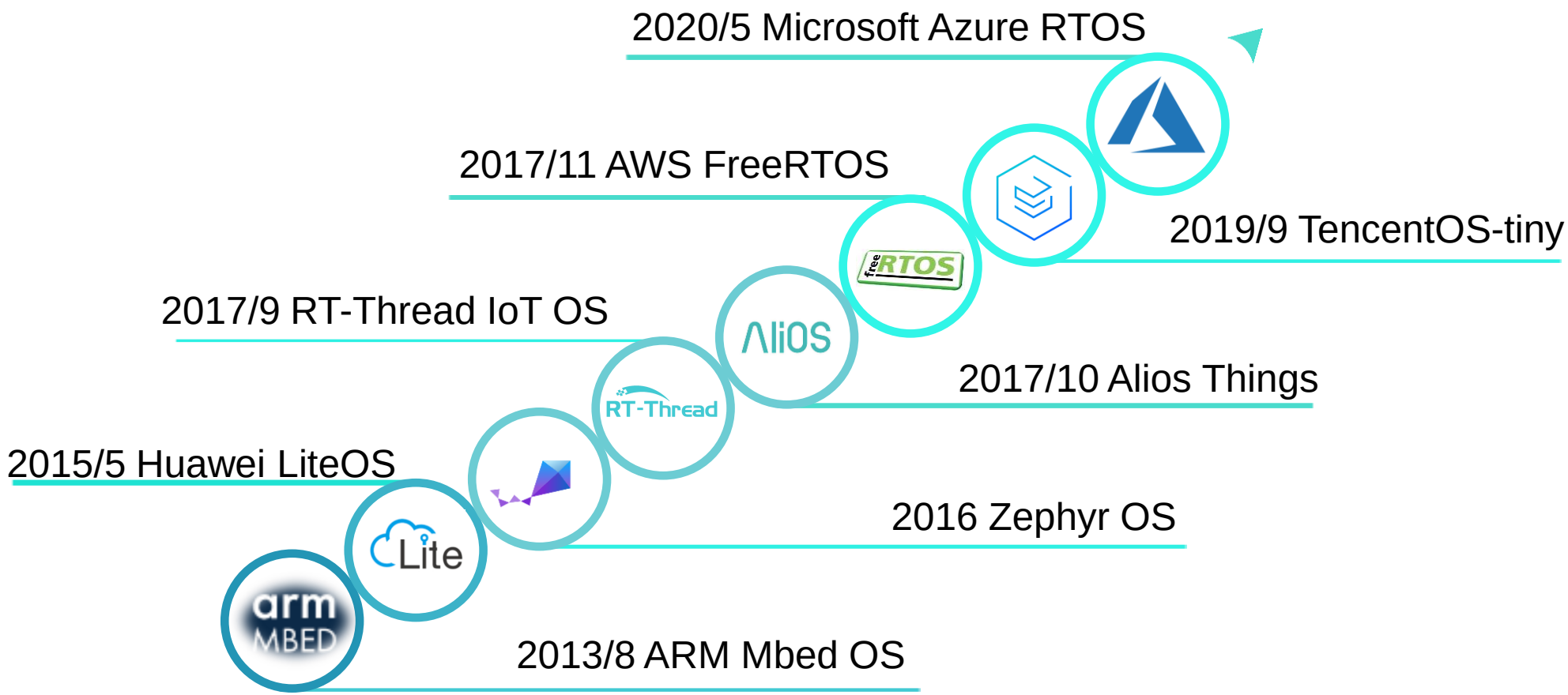
- 资源占用低，可裁剪性、可配置性

嵌入式操作系统

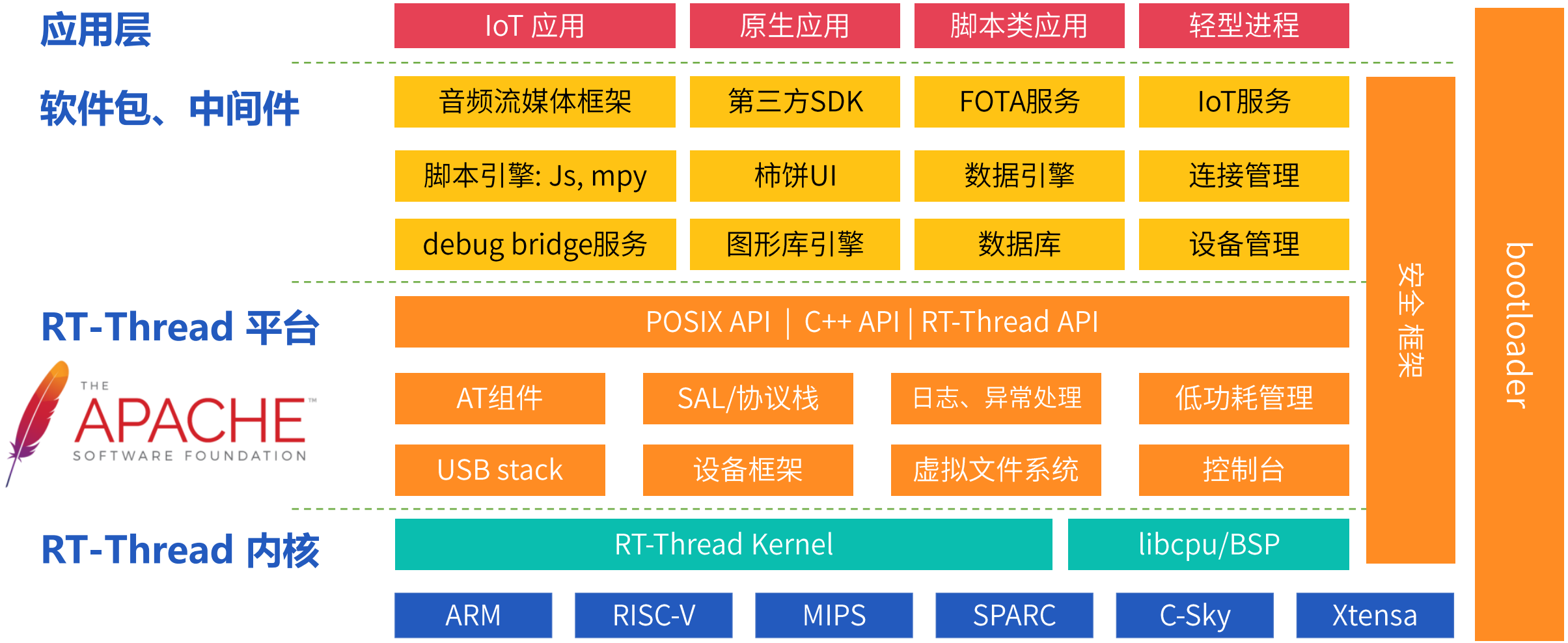
- 老牌的开源嵌入式操作系统：
  - eCos，1997由Cygnus Solutions开发，后被RedHat收购；RedHat收缩这部分业务后，独立出来成立公司，提供相关维护、服务。
  - RTEMS，1993年初始放于FTP上，导弹系统的实时执行程序。
  - 与编译器绑定严重，耦合性高，可裁剪性一般；

- ARM Cortex-M的发展，让更多设备使用嵌入式操作系统
  - 更高的集成度
    - SoC，把相关外设集成起来，Flash、SRAM集成到芯片中
    - 更易于使用的中断
  - 32位MCU芯片
    - 摩尔定律，Flash、SRAM、主频都在迅速提升
    - 集成更多的外设，功能逐渐复杂，裸机需要调整到使用操作系统的方式

IoT领域，巨头们的军备竞赛



RT-Thread标准版本



应用领域

- 加速产品上市
- 良好的用户体验

消费物联网




RT-Thread






行业物联网






- 可靠、稳定
- 坚如磐石

03

微内核操作系统

面向中高端终端领域

RT-Thread

复杂场合：中高端设备

启动快

系统在300ms内启动，有图像显示

兼容Linux生态

01

03

安全性高

内核稳定可靠，应用易升级

应用相互之间不受影响，安全性高

低成本

低成本低功耗

低资源占用，Flash/DRAM占用量

02

04

实时性强

强实时性

强实时的图形显示

启动时间小于1s，迅速响应

面向中高端的“复杂”系统

- 系统与应用分离的操作系统
  - 拆解方便，易于升级；
  - 团队协作开发更容易，维护性好，应用的安全性强

- 一些传统需求依然需要考虑
  - 实时性情况；
  - 资源占用低，资源占用 == 成本；
  - POSIX环境，易于现有应用程序移植；

- 启动迅速，轻型化、面向设备端的操作系统
  - 微内核操作系统
    - QNX、Fuchsia等

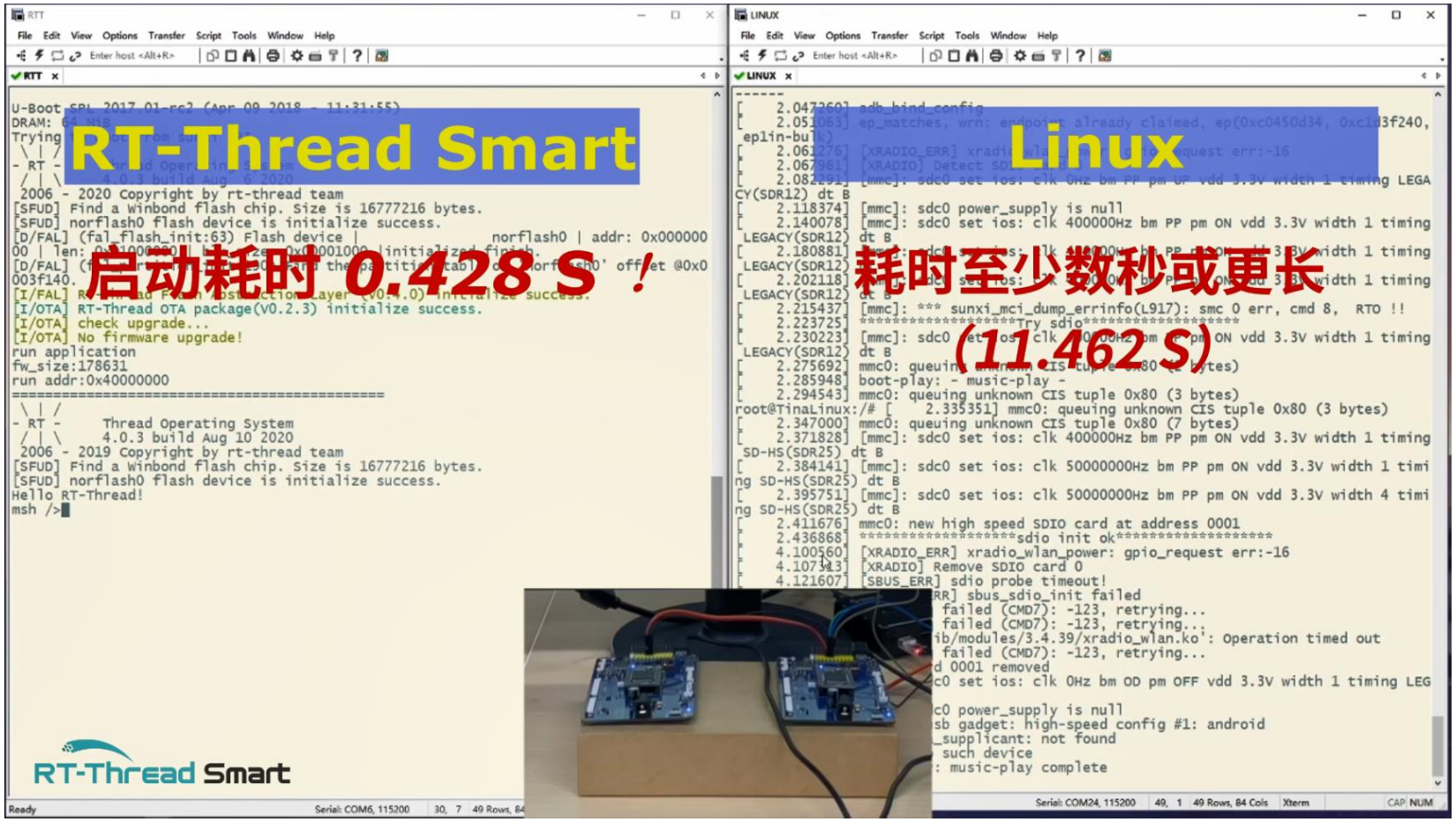
RT-Thread Smart

RT-Smart	内核 504kB，压缩后217kB；根文件系统低至128kB，DRAM占用1.9MB	<500ms启动时间
Linux	压缩内核后3.57MB，根文件系统5MB，DRAM占用17.4MB	5 – 10秒启动时间

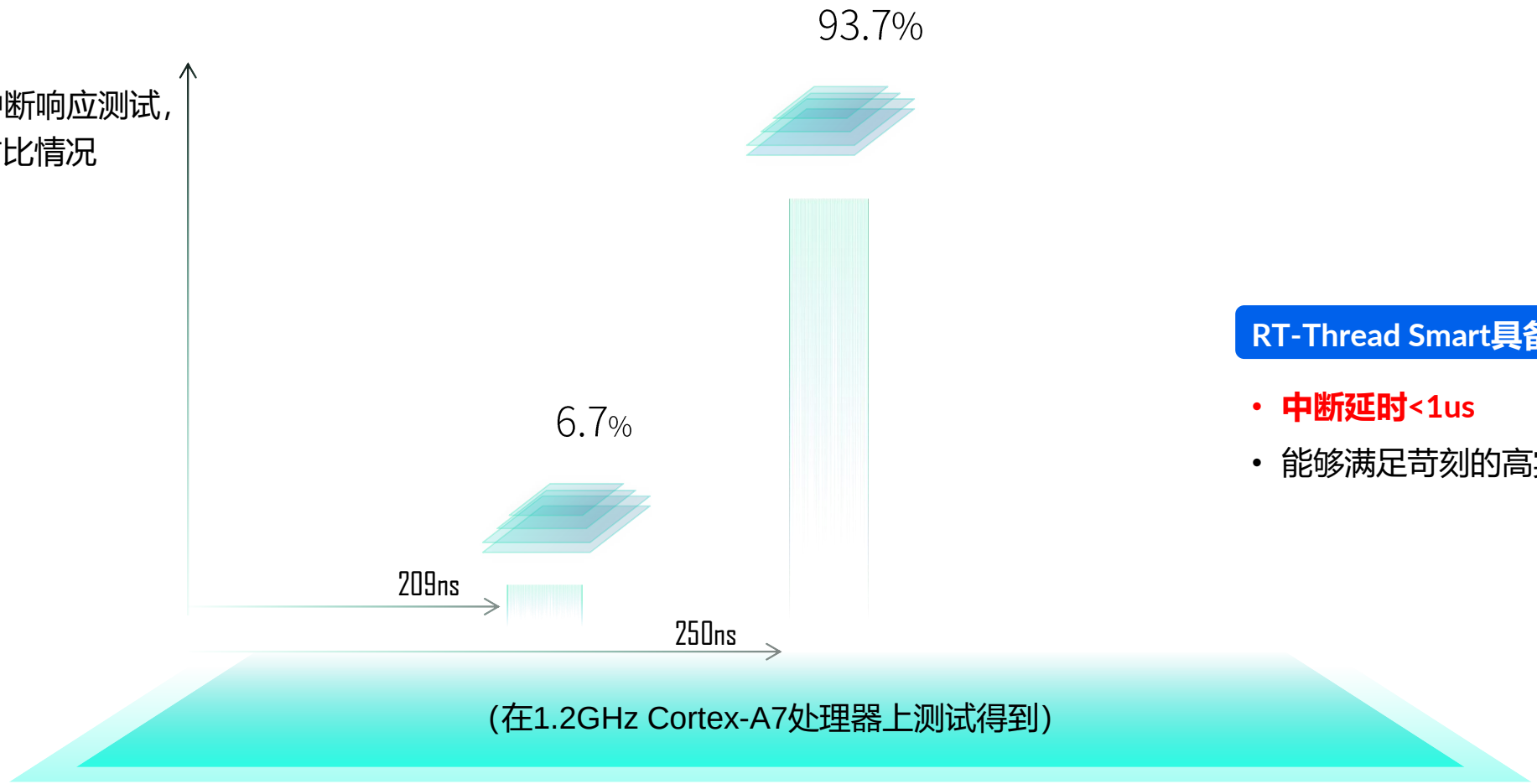
启动时间对比视频

同时开机对比

- 硬件条件完全相同



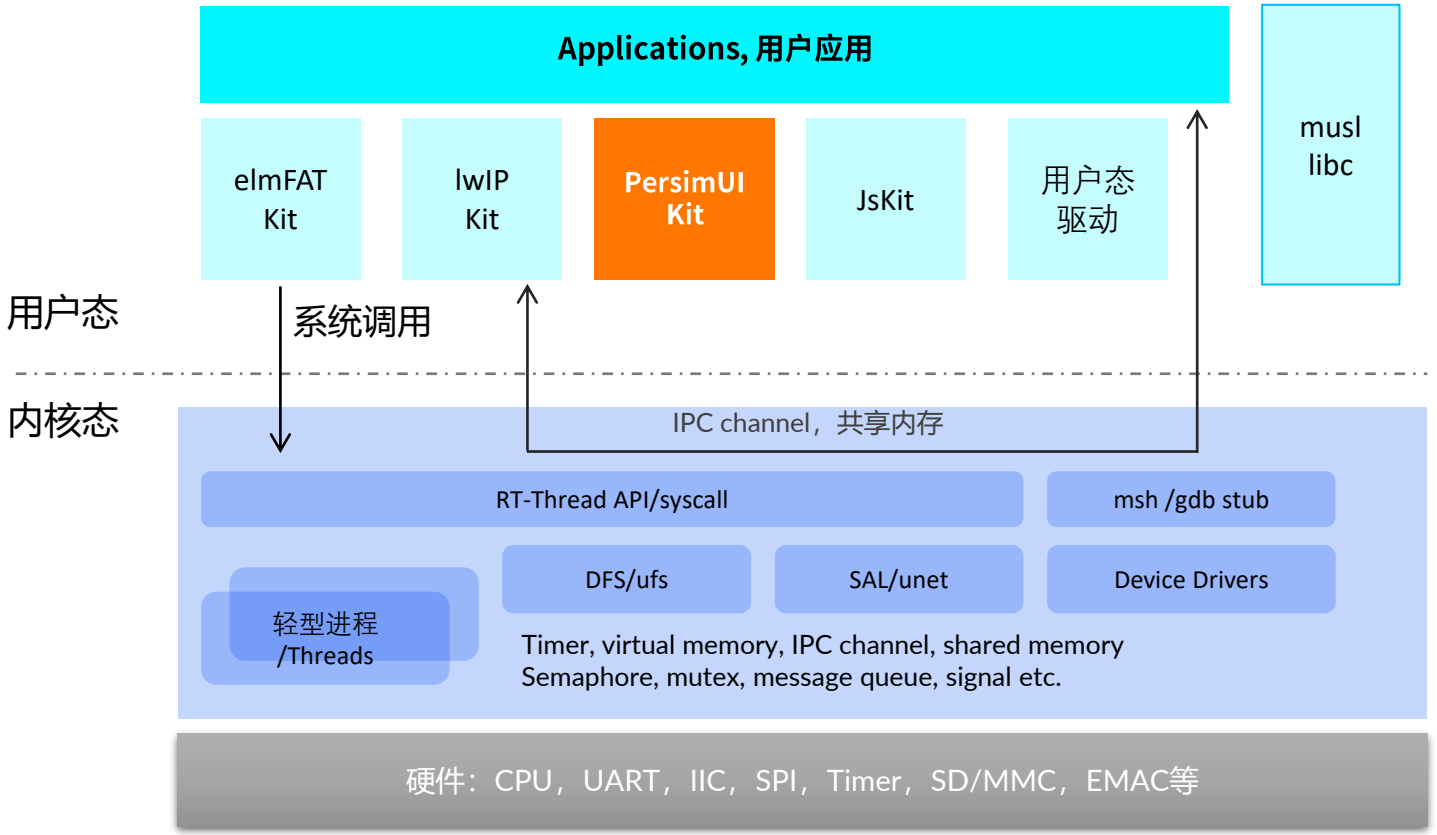
RT-Thread Smart的实时性



RT-Thread Smart具备优异的实时性能

- **中断延时<1us**
- 能够满足苛刻的高实时性场合

微内核架构



01. 内核轻型化

- 低至500kB内核尺寸
- 只包含基本功能，同时也可定制（调整系统服务位置）

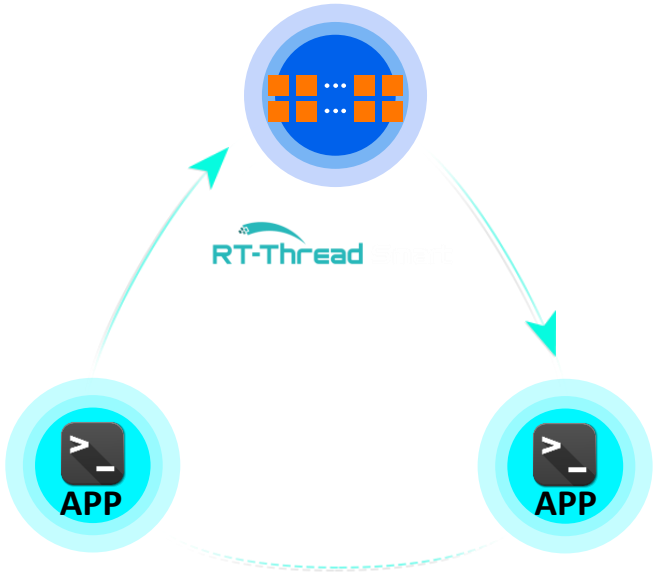
02. 用户态系统服务

- 可拆卸，可重启
- 每个应用进程具备独立的地址空间，相互隔离，安全性好

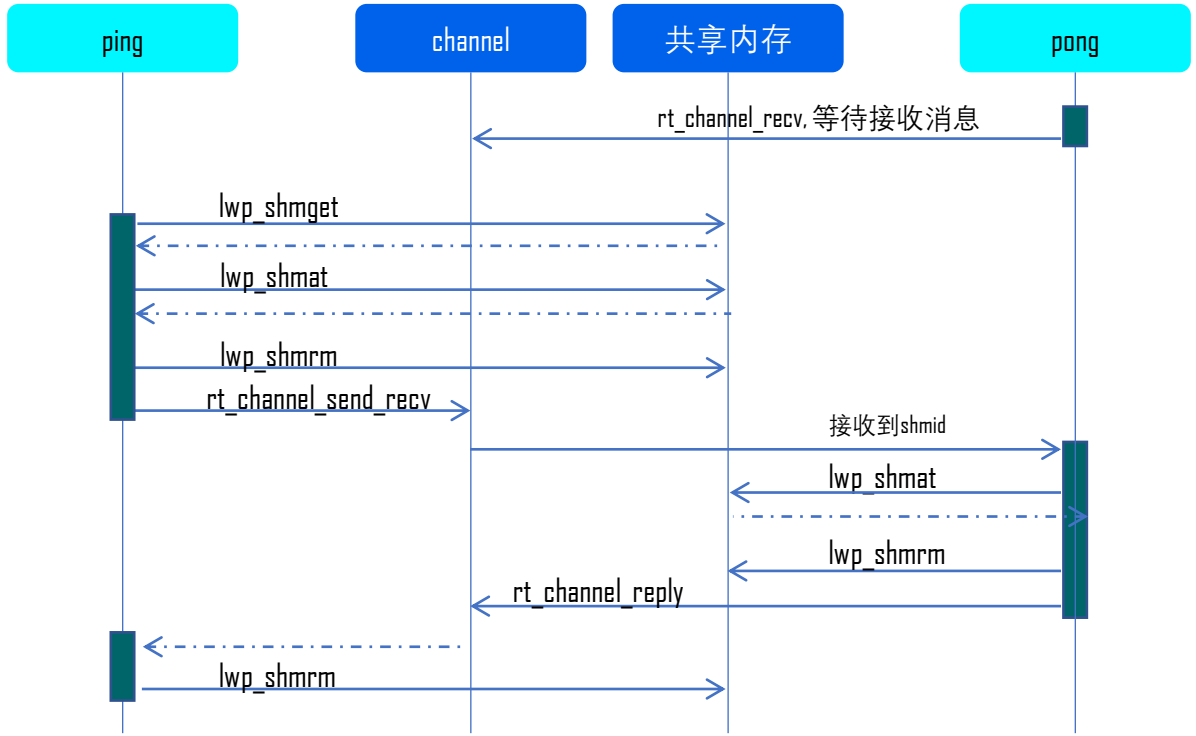
03. 相同的API风格

- 应用与内核都可延续RT-Thread API；
- 用户态扩展性强

RT-Thread Smart：进程间通信



- 进程间以消息方式传递数据，仅消息句柄，速度快、效率高；
- 消息数据放于共享内存空间中；
- 以共享内存方式解决进程与进程间数据地址的问题，同时免去数据复制开销；



RT-Thread Smart的用户态系统

POSIX环境

- POSIX.1
  - 部分进程相关API
  - POSIX signals
  - File I/O，Pipe etc
- POSIX threads – POSIX.1c

musl libc

- 开源的libc库，采用MIT license.
- 小型化libc库，用在多种场合
- 标准C库/POSIX库，完全一致性及鲁棒性；

RT-Thread CRT

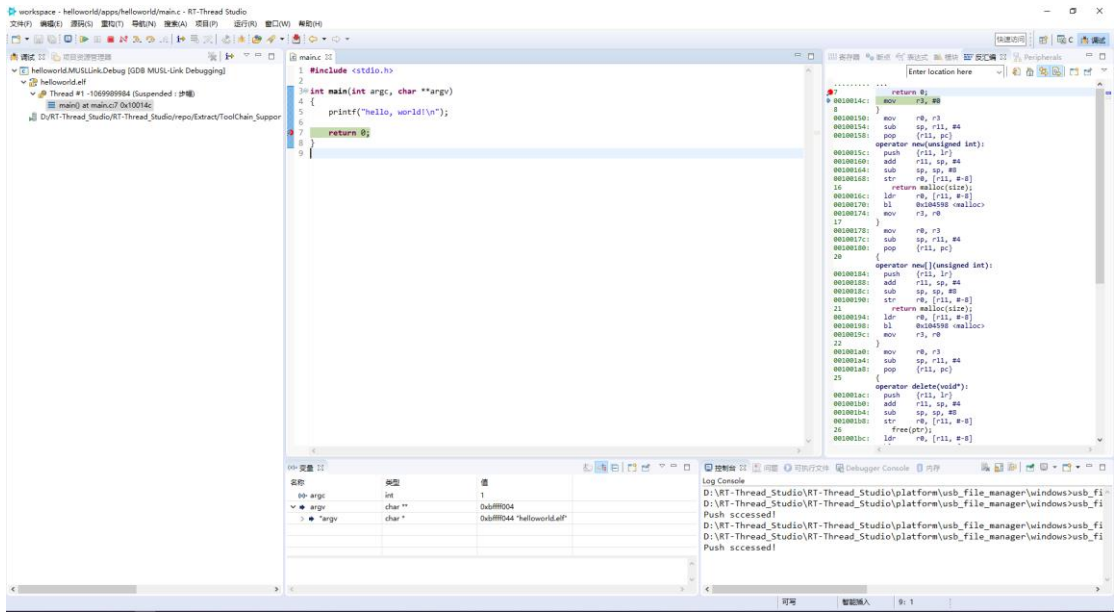
- 覆盖RT-Thread原有的API；
- 可以使用scons构建，并对接RT-Thread在线软件包；
- **延续RT-Thread原有生态**

RT-Thread Studio开发环境

与RT-Thread Studio集成

一站式工具

降低使用门槛
建立工程，代码编辑，甚至重构
下载、调试等完整功能



提升工作效率

磨刀不误砍柴工，
生产效率是企业的核心诉求

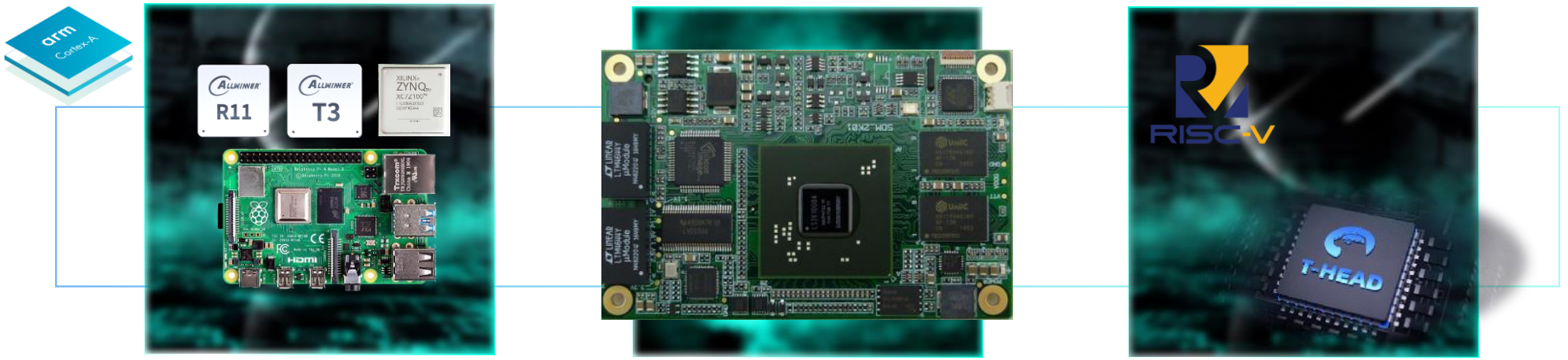
Web化的IDE

连接云端软件包
Web前端、浏览器 + node.js

软件定义终端开发

内置指令模拟器，在无硬件的条件下也可以进行调试、开发

RT-Thread Smart的硬件支持情况



ARM

全志Cortex-A、Xilinx Zynq、树莓派

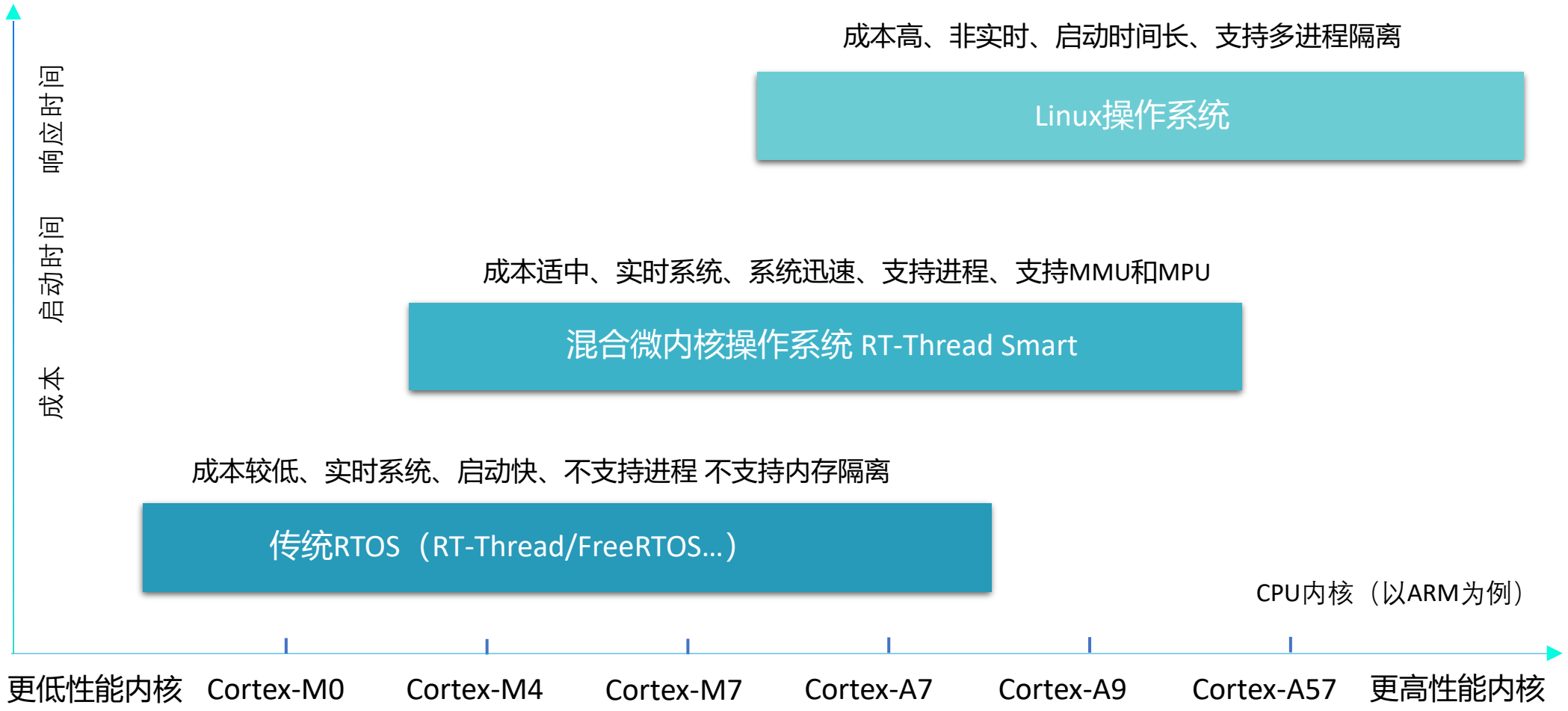
MIPS/龙芯

龙芯2K，64位处理器

更多.....

CK810、RISC-V

RT-Thread面向更广泛的IoT端侧场景



RT-Thread

· 谢谢聆听 ·

网站：www.rt-thread.org　　微信：RTThread