

基于开源固件与机架级管理技术的创新

林柏宏(Mike Lin) *General Manager @AMI*

孙国祥(Mike Sun) *Firmware R&D @AMI*

宋才栋(CD Song) *DCM @AMI*

Agenda

- **AMI拥抱开源并坚持开放标准**
- **基于UEFI开放标准对RISC-V固件的设计和开发**
- **从服务器到机架：AI 时代的机架级管理趋势与 AMI 的探索**

AMI拥抱开源并坚持开放标准

- 归一化
 - 同一个Codebase
 - 同一个核心架构
 - 同一个开发者体验
- 我们使命和目标
 - 去碎片化
 - 标准化
 - 携手合作

- AMI对开源的承诺

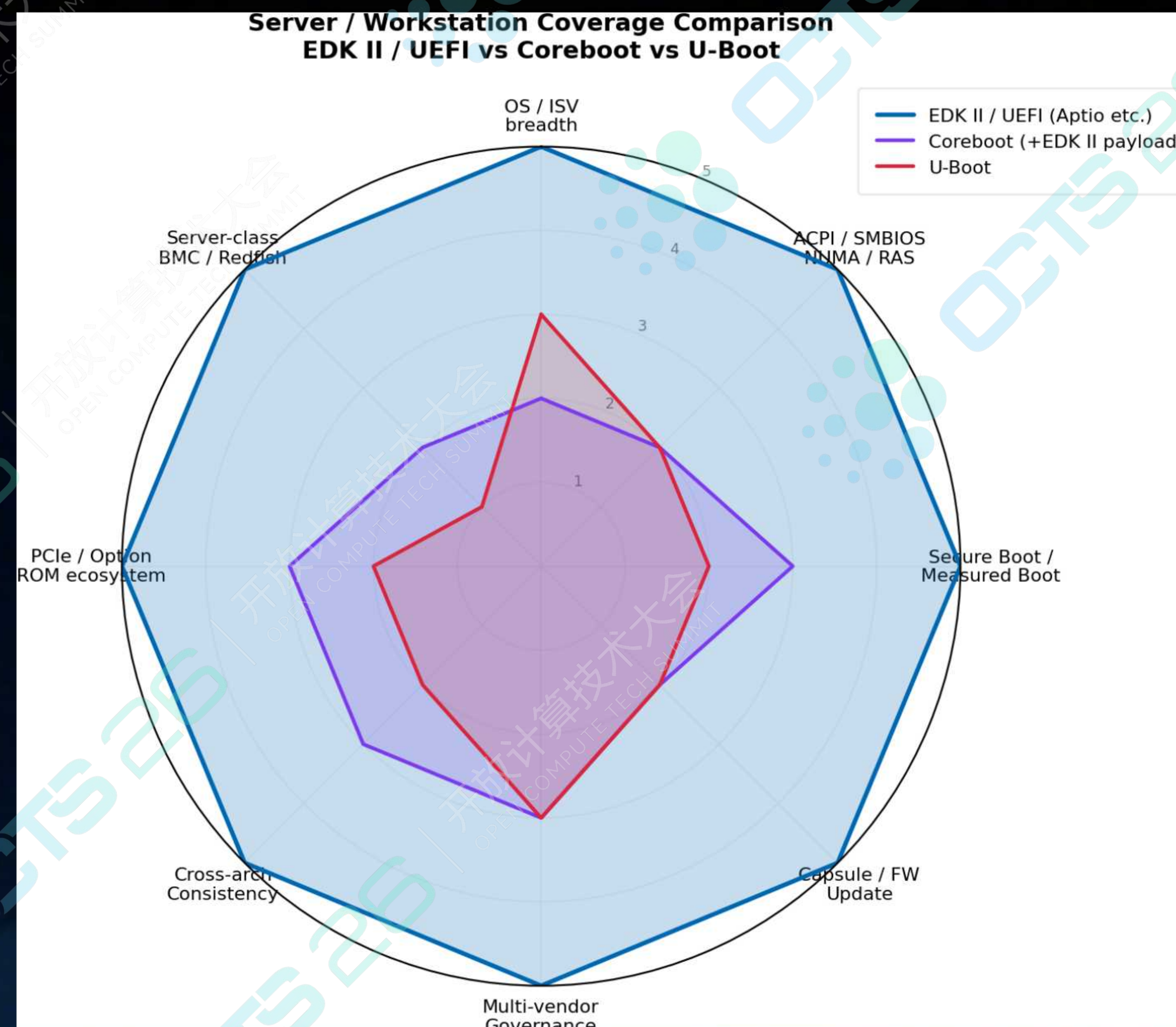
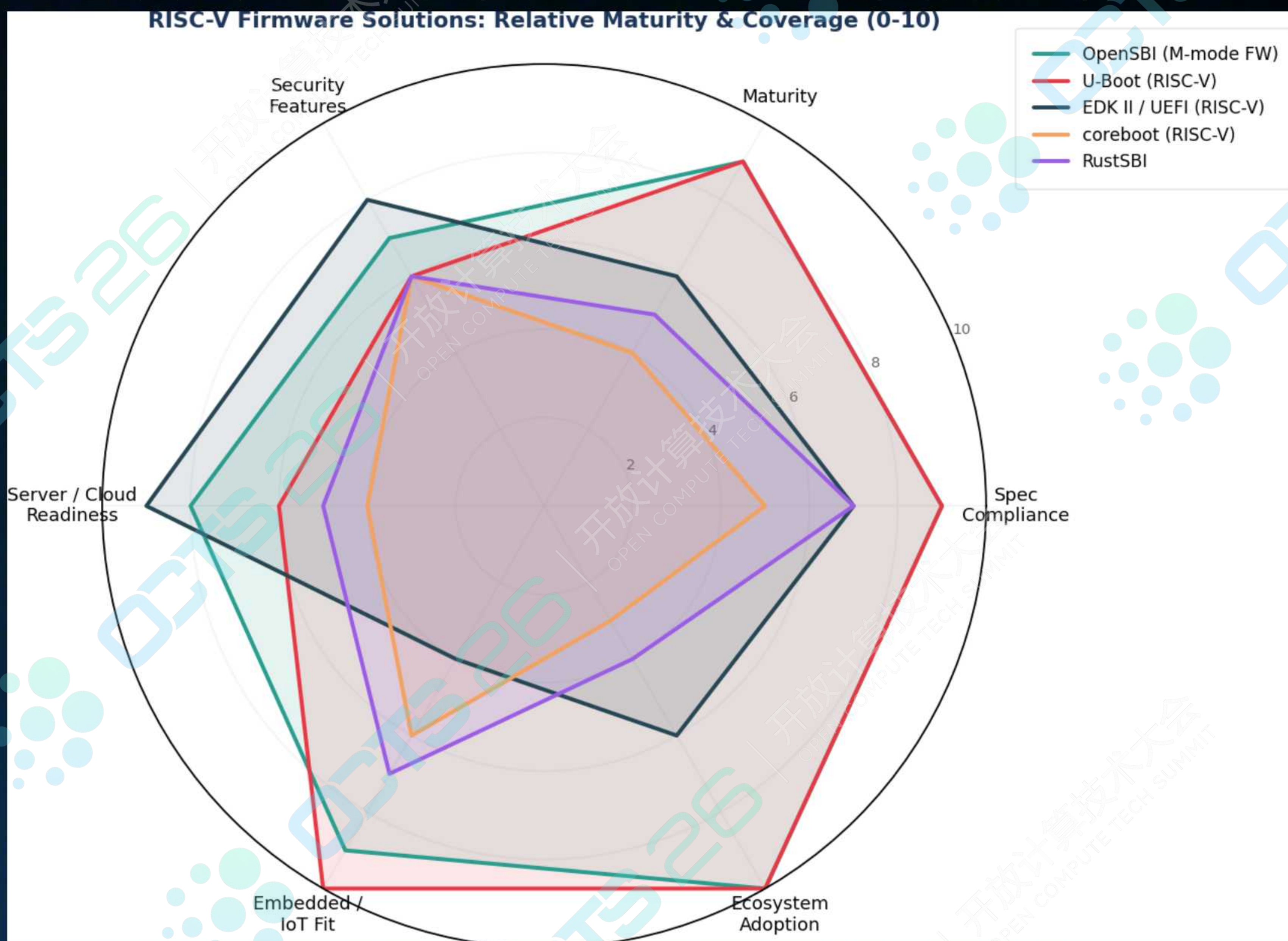
将致力于推动固件生态系统的转型，积极拥抱开放架构、开放标准和开源解决方案，以满足云服务提供商、ODM、OEM 以及更广泛的技术社区不断变化的需求。我们与OCP、DMTF,UEFI和其他领先的标准组织保持着深入的合作，以确保最先进的开放固件解决方案能够便捷地应用于整个生态系统。

基于UEFI开放标准对RISC-V固件的设计和开发

- 引言：RISC-V 固件生态现况
- RISC-V 的 UEFI 和 EDK II 现状
- RISC-V 固件开源开放式设计
- 基于 EDK2 的平台构建
- 基于 EDK2 的源码结构
- 总结与展望

引言：RISC-V 固件生态现状

- 在嵌入式和物联网领域RISC-V的U-Boot固件更成熟。
- 在服务器/工作站/个人电脑领域RISC-V EDK2固件更成熟。
- 这就是为什么需要基于EDK2的固件方案



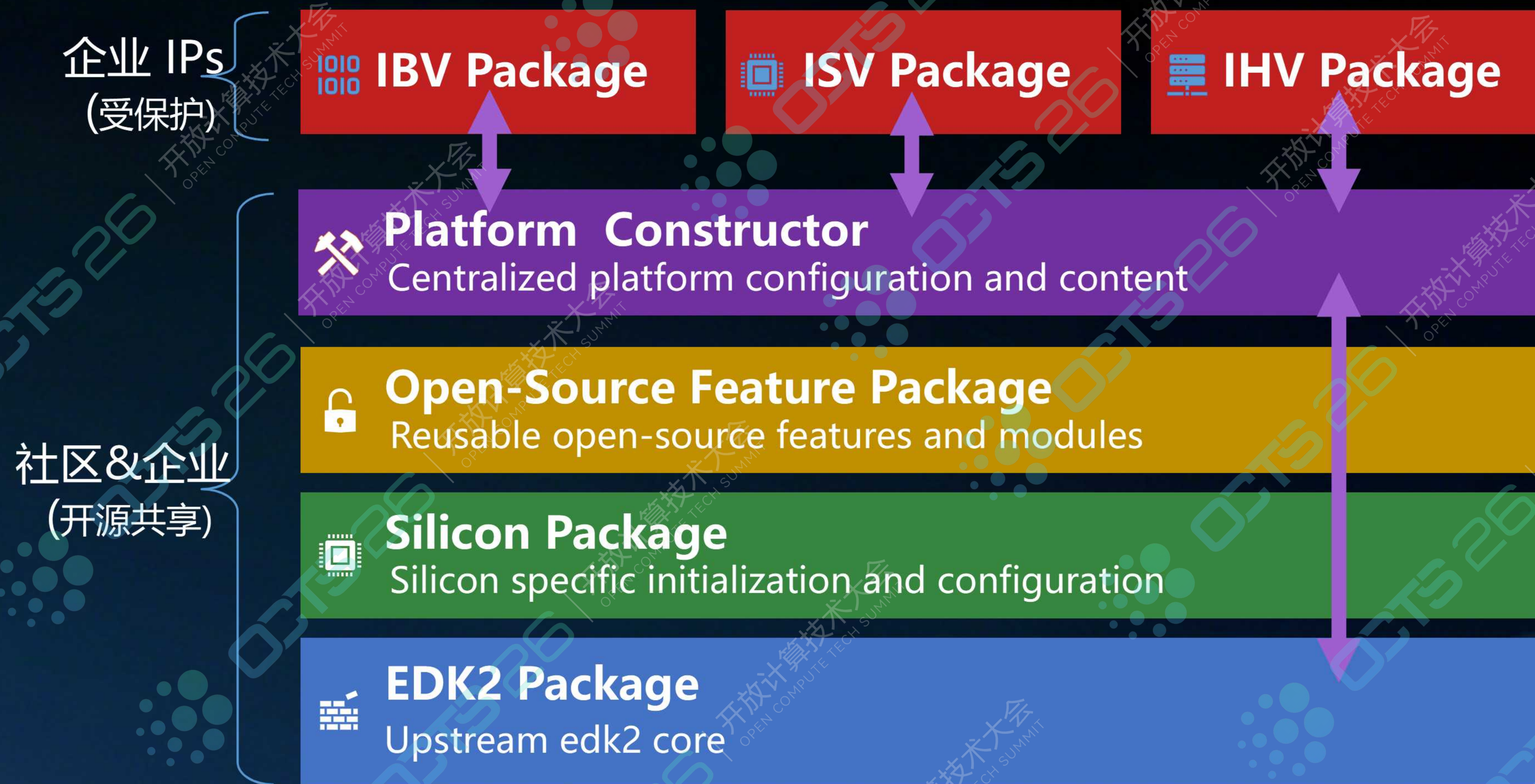
RISC-V 上 UEFI 和 EDK II 的现状

服务于RISC-V服务器、工作站及个人电脑的UEFI能力矩阵及EDK2的实现实例

UEFI Capability	状态	EDK2 Readiness
UEFI 2.10/RISC-V UEFI Protocol Spec 1.0	完成	MdePkg
RISC-V 64-bit	完成	UefiCpuPkg/CpuDxeRiscV64
Secure Boot/StandaloneMM	完成	SecurityPkg/StandaloneMMPkg
TPM 2.0	完成	SecurityPkg/CryptoPkg/Tcg2Dxe
PCIe	完成	MdeModulePkg/PciBusDxe
ACPI	完成	OvmfPkg/RiscVVirt/AcpiTables
SMBIOS	完成	MdeModulePkg/SmbiosDxe
PXE/HTTP Boot	完成	NetworkPkg/Boot Manager
UEFI Shell	完成	ShellPkg
Redfish	完成	RedfishPkg
更多的。。。	完成	。 。 。
更多的。。。	验证中。。。	。 。 。
更多的。。。	开发中。。。	。 。 。
更多的。。。	。 。 。	。 。 。

RISC-V 固件开源开放式设计

AMI提供全方位的解决方案，既有可使用的、经久验证的、成熟/闭源的codebase，亦有基于EDK2的社区开源版方案。如下开源方案所示：



- **企业版：**CE + IBV/ISV/IHV 功能IP

- **社区版：**EDK2 + 芯片/功能 开源包

基于 EDK2 的平台构建

使用 MS MU项目开源的Stuart 构建工作流，它是被贡献到EDK2的一套扩展工具，用于简化和自动化构建/维护 EDK2-based的UEFI固件体系。

➤ AmiPkg

AmiPkg包含用于标准 EDK2 软件包的通用 定义和配置文件，以及Stuart构建系统的核心。

➤ Stuart Invocables

Invocable是执行特定构建任务的单个命令行可执行文件

stuart_setup : 初始化并下载项目的必要依赖源代码。

stuart_update : 检查并更新现有依赖到最新版本。

stuart_build : 编译和构建。

stuart_ci_setup/_ci_build/_parse/_report : 持续集成/分析/汇报

➤ IBV/ISV/IHV受保护的IP

例如：IBV的专属功能 AxxRedfishBasicPkg, AxxRedfishHostPkg, AxxRedfishAdvancedPkg
AxxSecurityPkg, AxxServerPkg, AxxTsePkg, 等等。

基于 EDK2 的源码结构

● **Green:**
表示开源共享部分

● **Red:**
表示来自于BIOS、Silicon、OEM、ODM等的专属功能IP



实际案例

- > AmiFeatures
- > AmiPkg
- > Build
- > Ci
- > Conf
- > Edk2
- > Platform
- > Silicon
- > .gitignore
- > .gitlab-ci.yml
- > .gitmodules
- > README.md

总结与展望

基于EDK2:

x86 上用了 20 年构建了固件生态的基石

ARM 上用了 10 年追赶并站稳。

RISC-V 还需要再等 10 年吗?

不用再等10年啦!

Based on EDK2:

20 years → Building the foundation of the firmware ecosystem on x86

10 years → Arm catch up and establish the foundation

???years → Building the foundation of RISC-V firmware ecosystem.

Do not to wait another 10 years!

从服务器到机架：AI 时代的机架级管理趋势与 AMI 的探索

- 从服务器到机架：早期演进
- 功率跃升：AI 为何把管理单元推向机架
- 机架级管理的挑战
- 迈向标准化：OCP OpenRMC
- 软件机架管理器 (SRM) 架构
- 机架级管理器 (RSM) 分层架构
- 总结与展望

从服务器到机架：早期演进

2012–2013

“机架即计算机”

微软研究院设立“机架级计算”项目（2013 年 1 月）；研究者将机架定义为“新的部署单元”。

2014

Intel RSA

Intel 在 IDF 2014 系统阐述机架级架构（RSA）：把机架内 CPU、内存、存储解耦为可按需组合、统一管理的“池化系统”。

2015–2016

高速互连使解耦可行

100G 以太网、Intel Omni-Path、光互连与 NVMe-oF 走向成熟，让内存、存储跨机架池化在带宽与延迟上首次可行。

2016–2018

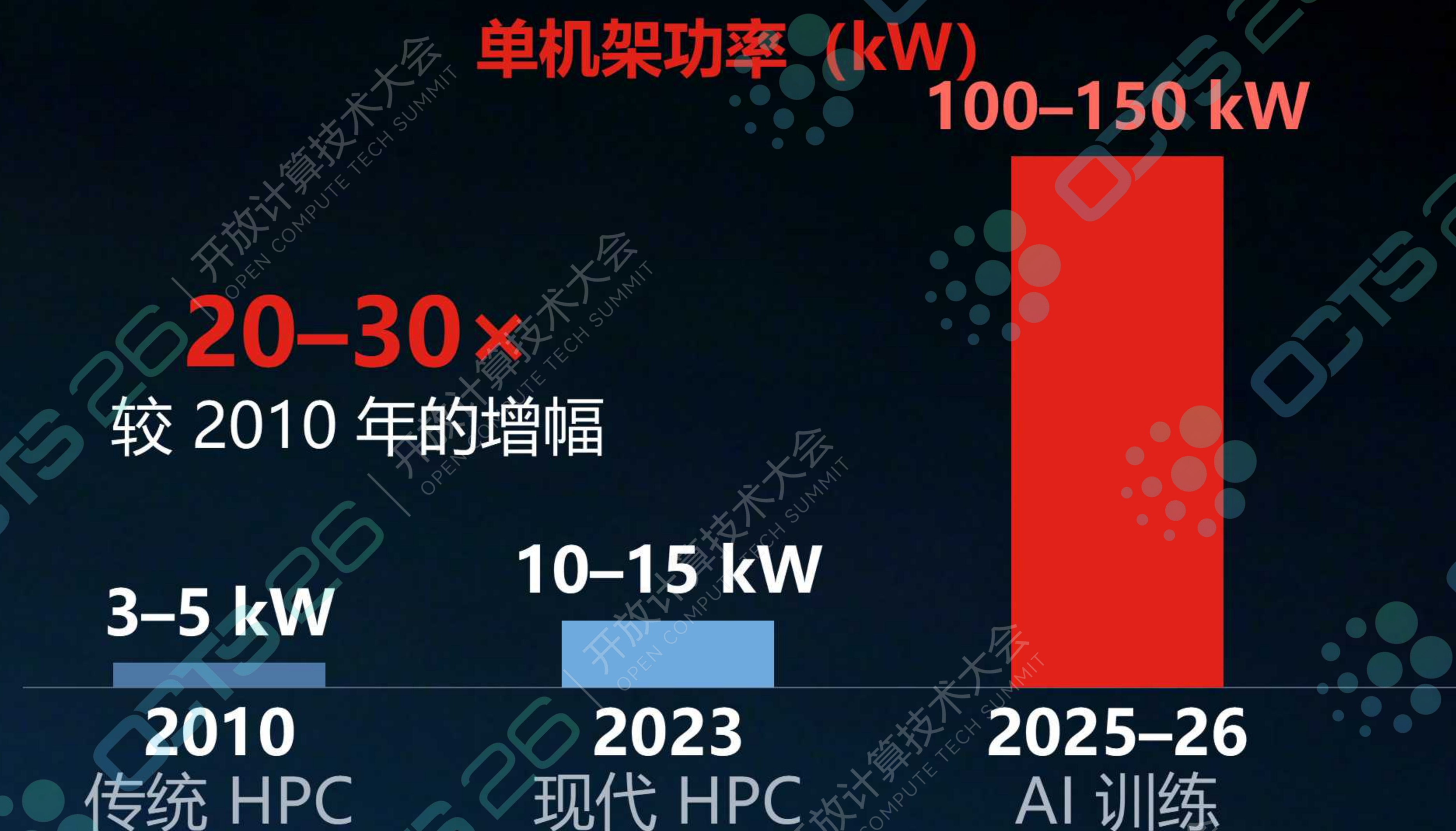
初期产品面世

Intel RSD 多轮迭代（v1.2 → v2.3），新增 NVMe 池化、FPGA 解耦与 NVMe-oF 支持；9+ 家 OEM 出货。

将机架视为单一管理单元的理念已逾十年——从最初概念，到机架级架构（RSA）与高速互连让解耦在管理与技术上同时成立，再到早期产品面世，共同为今天 AI 时代的机架级管理奠定基础。

功率跃升：AI 为何把管理单元推向机架

AI 将单机架功率推高 **20–30 倍**，供电、散热、安全在机架物理交汇，管理单元从单台服务器转向整机架。



供电链 · 高功率密度

供电机架级共享，须整机架协调

散热链 · 风冷失效

液冷回路机架共享，须整机架管理

安全链 · 漏液风险

亚秒级本地闭环，须机架内隔离

共同结论 · 以机架为管理单元
功率 · 散热 · 安全 在机架物理交汇

机架级管理的挑战

服务器 BMC

Redfish、IPMI

GPU 加速卡

Redfish (经 BMC)、厂商 API

Scale-Up 交换机

厂商 API、Redfish ...

Power Shelf 电源架

Redfish (Power Shelf Profile)

CDU / 液冷单元

Modbus、SNMP、Redfish

架顶交换机 ToR

gNMI、SNMP、厂商 CLI

机架管理控制器
RMC

翻译 · 归一化 · 聚合
 $N \times M$ 集成难题

统一 北向接口
(OpenRMC)

→ 数据中心管理层

设备众多、协议各异：暂无单一标准全覆盖。

迈向标准化：OCP OpenRMC

v1.0
2021

- 资产清单
- 功耗监控
- 温度监控
- 健康状态
- 电源管理
- 固件管理

v1.1
2023

- 节点分组操作
- 节点级固件升级
- 节点安全授权

v1.2
草案

- 遥测
- 可组合性

v1.3
草案

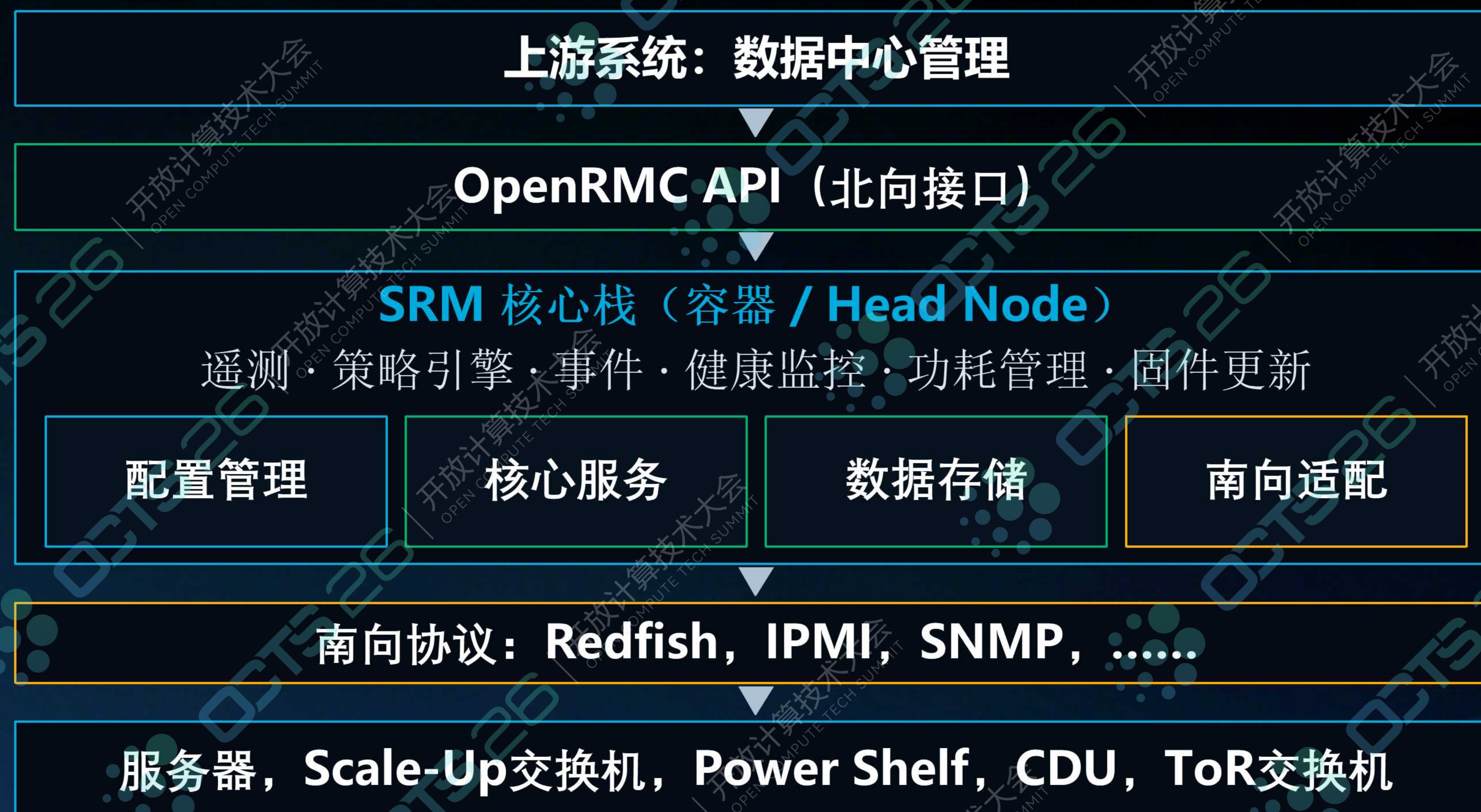
- 聚合
- 策略管理
- POD管理
- 扩展与连接

2.X
展望

- 下一代AI基础设施

软件机架管理器 (SRM) 架构概略图

从北向**API**到南向设备的最小闭环



特点

- **北向统一**：通过OpenRMC暴露资产、遥测、事件与动作。
- **部署灵活**：运行于Head Node或容器中，支持 x86 / ARM 平台。
- **本地闭环**：策略引擎就近处理漏液、功耗、健康告警，以及升级编排。
- **南向覆盖**：面向多类型、多厂商、多协议，统一接入多种设备。

机架级管理器 (RSM) 分层架构

- **分层自治**: 机架本地快速响应, 全局统一协同
- **云原生**: IM 微服务容器化部署 (Kubernetes)
- **灵活部署**: RM 可为固件 (FRM) 或软件 (SRM)

基础设施管理器 (IM)

全局编排 · 全局分析 · 全局策略

编排

可视化

可观测性

AI 分析

策略

报表

▼ OpenRMC API ▼

机架管理器 (RM)

单机架监控 · 本地闭环控制

发现

资产

遥测

事件

健康

功耗

固件

策略

▼ Redfish · IPMI · SNMP ▼

机架硬件

服务器

Power Shelf

CDU

ToR交换机

Scale-Up交换机

总结与展望



机架级管理正成为 AI 基础设施规模化发展的必然方向



开放标准 (OpenRMC) 与产业协作是关键使能因素



AMI 基于 MegaRAC OneTree + OpenRMC + DCM 的实践经验, 持续探索机架级管理



未来方向: 安全可靠 | 资源可组合 | AI 驱动智能运维



欢迎社区协作与反馈

AMI

OneTree

DCM

开放标准

OpenRMC

THANKS