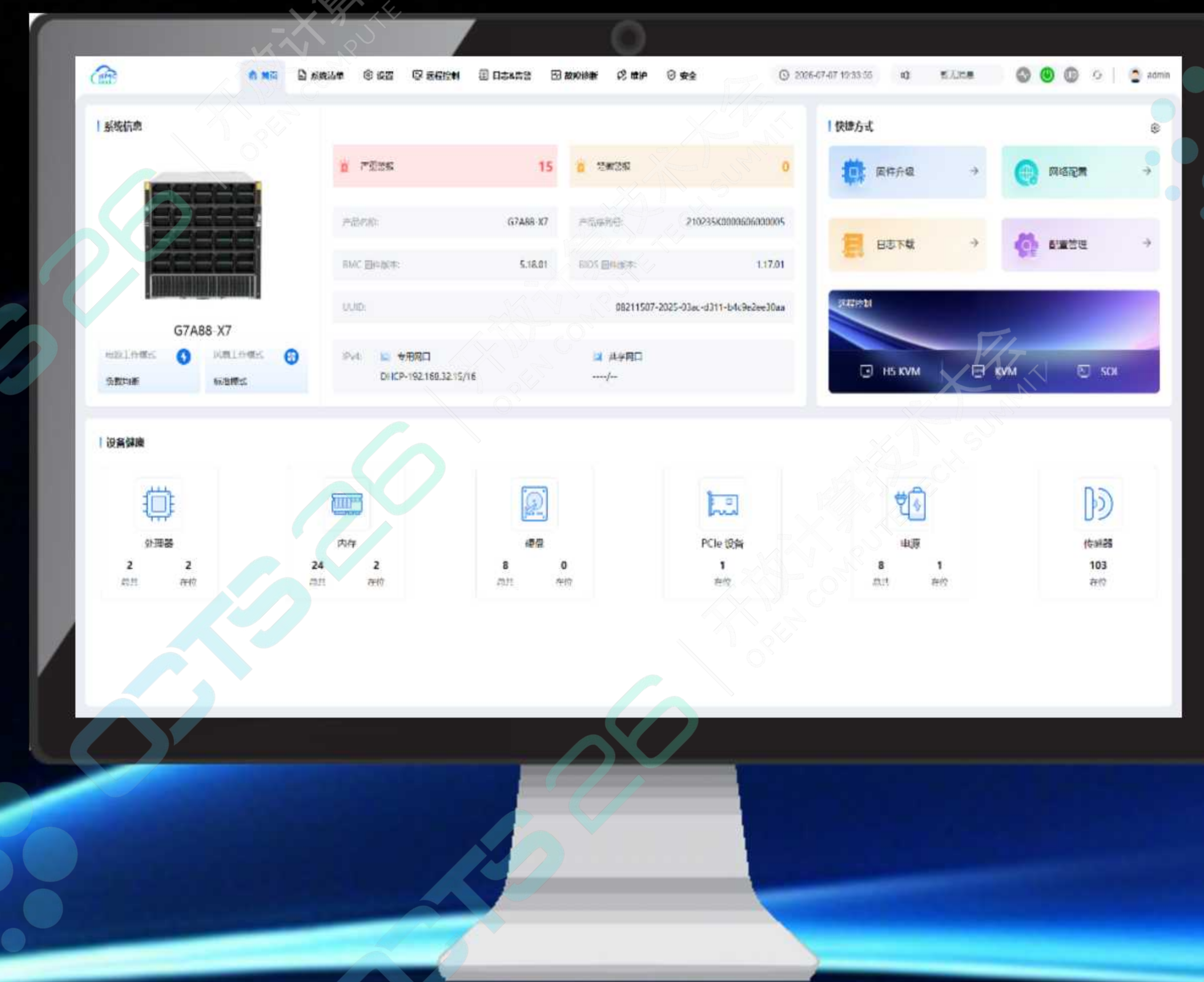


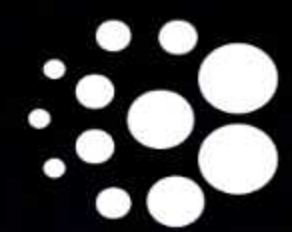
CNIT CloudBMC在架构优化和产品归一化设计实践

Architectural Optimization and Product Unification Design Practices of CNIT CloudBMC

云尖信息技术股份有限公司

魏岳伦/Allen





OCTS



固件产业技术创新联盟

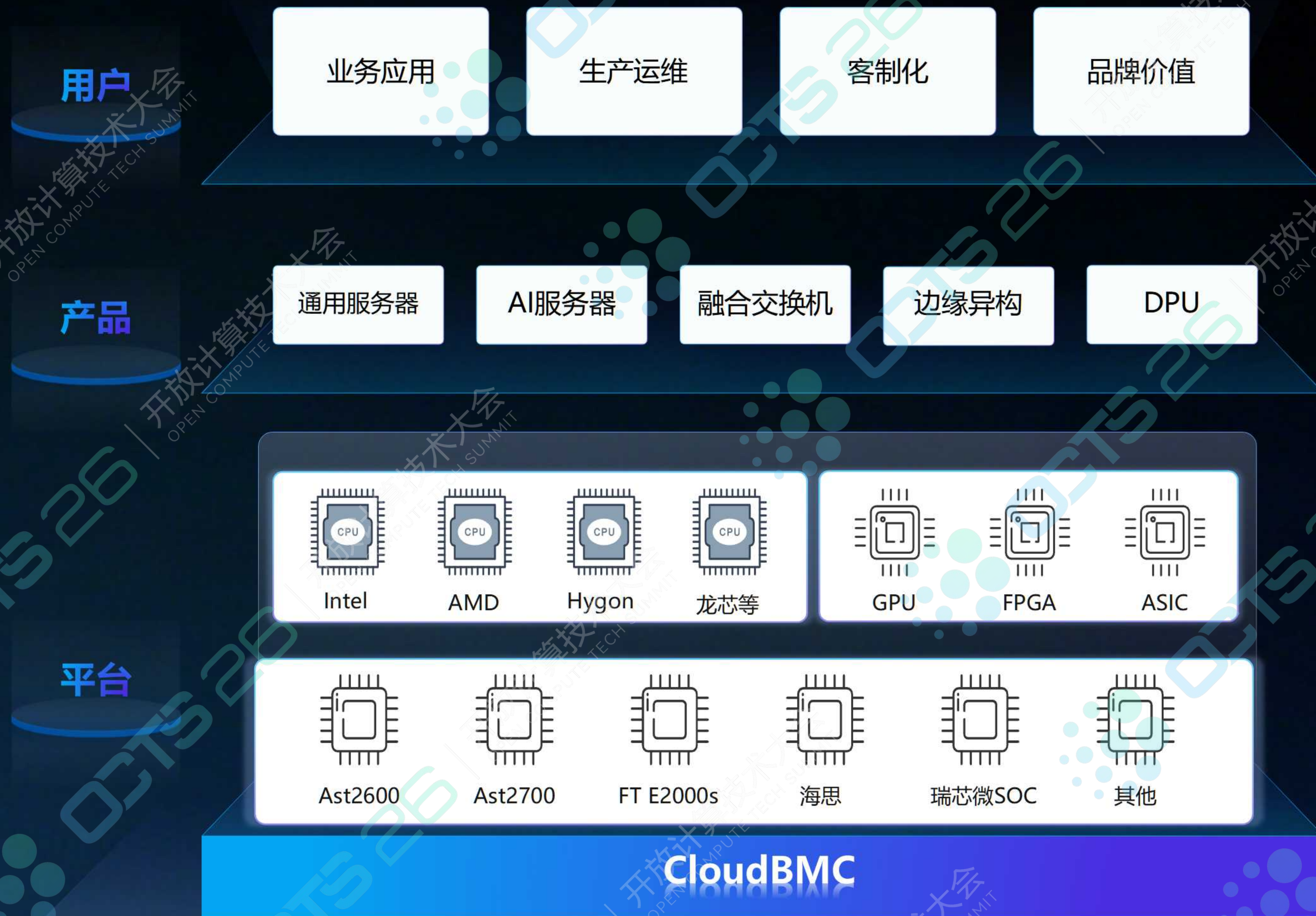


芯尖信息



CNIT CloudBMC平台架构

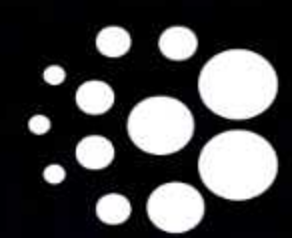
CloudBMC业务体系架构



- ✓ 支撑业务应用、生产运维多元使用场景
- ✓ 多维度客制化，打造用户全方位品牌价值

- ✓ 支持多产品形态
- ✓ 全面覆盖各类型产品业务需求

- ✓ 向下兼容BMC SOC芯片
- ✓ 适配多CPU平台特性
- ✓ 适配各类GPU/FPGA/ASIC等芯片产品
- ✓ 功能丰富，高度集成



OCTS

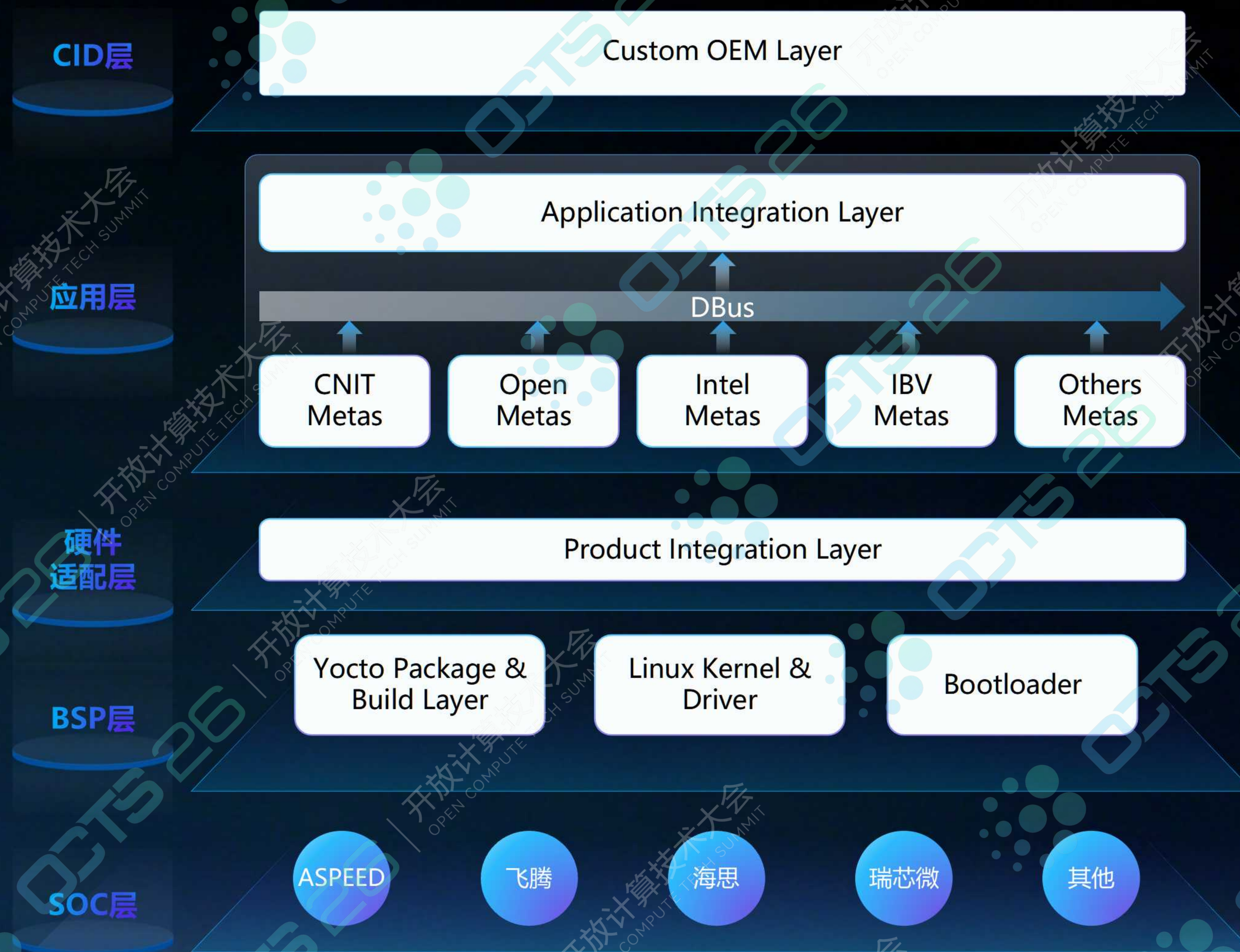


固件产业技术创新联盟



芯尖信息

CloudBMC软件层次架构



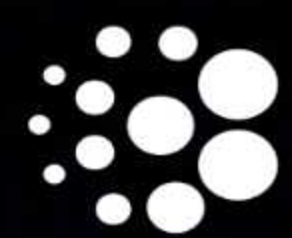
✓ 支撑产品和客户定制化内容

- ✓ 各应用APP, APP北向接口统一
- ✓ 支持第三方应用APP/插件平行兼容扩展

✓ 对应用层屏蔽产品硬件差异

- ✓ 源码包管理一致性
- ✓ Kernel/Driver/Boot

✓ 不同BMC SOC支撑



OpenBMC预构建+编译工具链优化

OpenBMC原始构建和工具链痛点



01

编译时间长

- ✓ **编译时间长**: 编译时间长, 通常编译时长1h+。
- ✓ **编译失败率高**: 源码编译对网络及编译环境要求高, 易导致编译失败。



02

Layer层级依赖严重

- ✓ **Layer层级相互依赖**: Layer层级recipes菜单相互依赖, 无法独立编译Layer层。
- ✓ **目标组件和Native组件依赖**: 编译目标组件时, 需要先编译所依赖的Native组件, 向下依赖导致单一组件编译时间长。

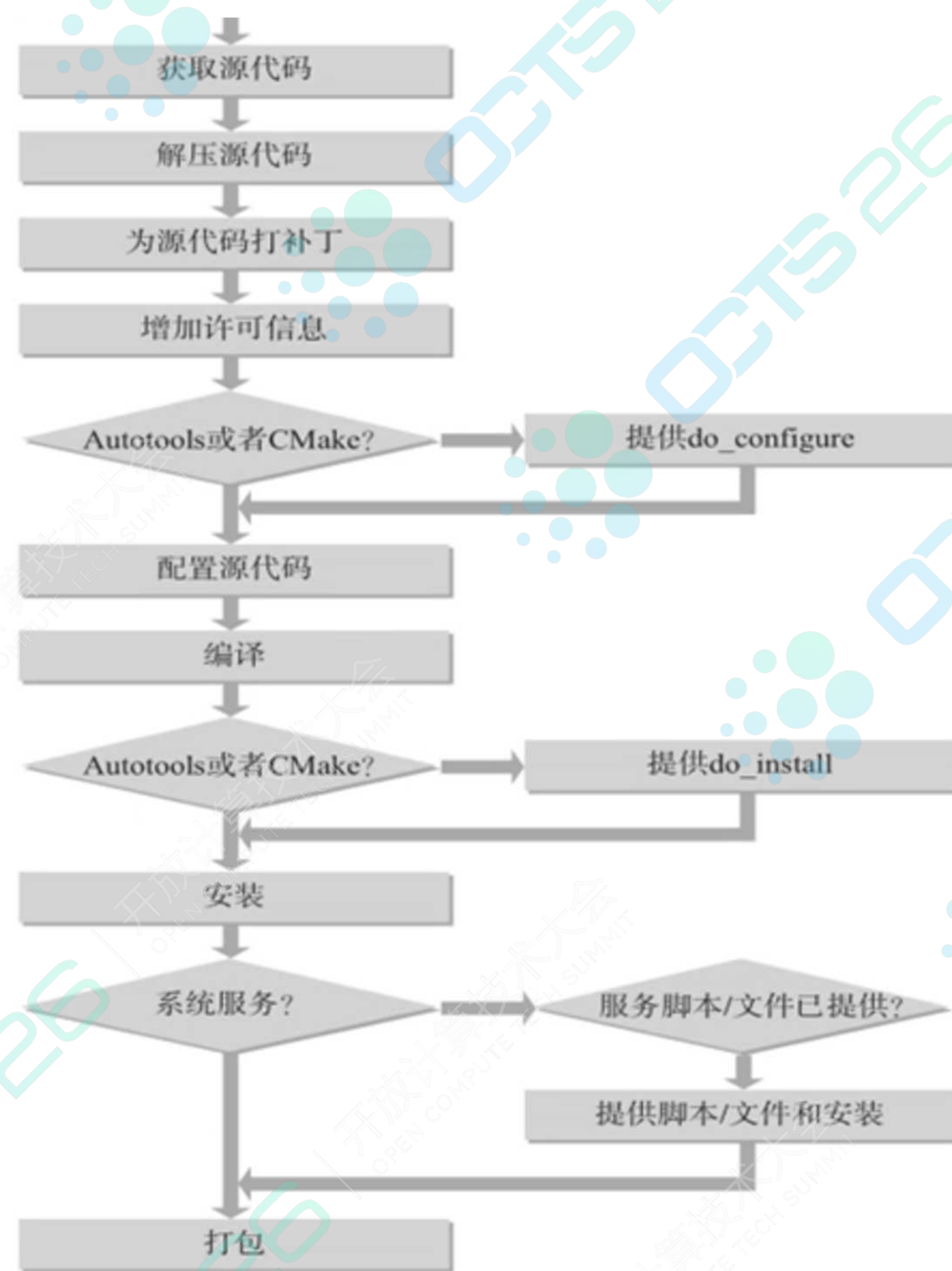


03

重复构建交叉工具链

- ✓ **重复构建交叉编译工具链**: 每次在业务软件编译时, 都需要重复编译工具链, 耗时长。
- ✓ **不同SOC平台编译工具链差异大**: SOC平台包括ARM和RISC-V指令集, 编译工具链差异大。

Yocto编译低效原因分析



- Yocto框架中包含大量的第三方Packages包和交叉编译工具链SDK，占用编译时间较长；
- Yocto为分布式编译框架，代码构建分成获取源代码、解压源代码、打补丁、增加许可信息、配置源代码、编译、安装和打包环节，构建链条过长，每个环节耗时大，进而导致整个工程代码构建时间长。

优化方案

第三方lib库RPM包引用

在第三方lib库.bb文件中，修改SRC_URI直接获取RPM包，通过bitbake bin_package类方法解析rpm包，并安装到work工作目录下。

```
SUMMARY = "Package the Proprietary Software"
DESCRIPTION = "MISC Proprietary library packages."
DEPENDS = "rpm-native mctp-misc"
RDEPENDS:${PN} = "mctp-misc"
LICENSE = "CLOSED"
ALLOW_EMPTY:${PN} = "1"
INSANE_SKIP:${PN} = "dev-so already-stripped libdir dev-deps file-rdeps ldflags"
PACKAGES = "${PN} ${PN}-dev"

RM_WORK_EXCLUDE += "storagecore"

S="${WORKDIR}/rpm"

SRC_URI = " \
    file://rpm \
    "

inherit bin_package
```

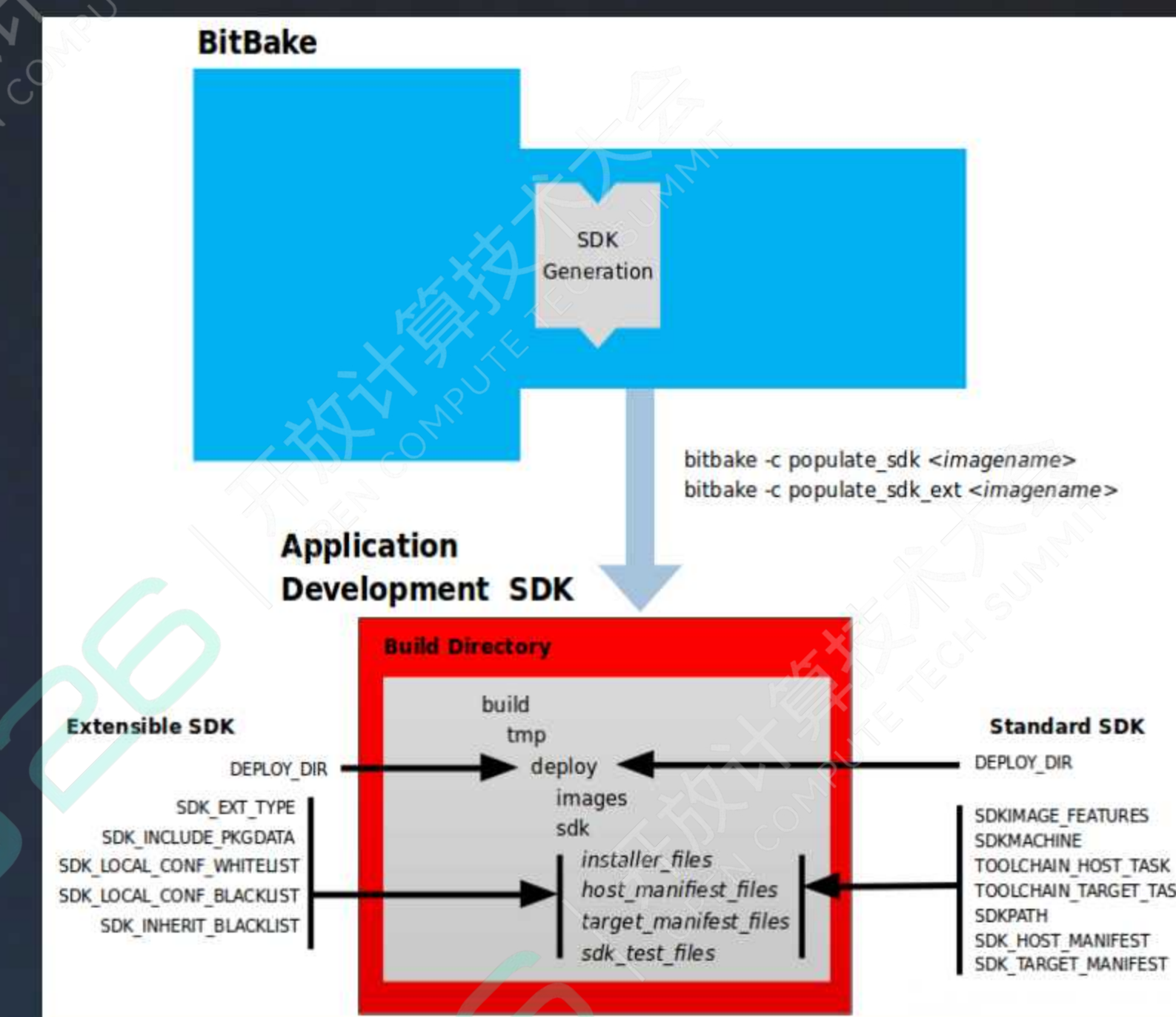
Pre-build编译工具链

输入外部Pre-build工具链，该工具链通过openbmc yocto工具生成，支持.sh格式发布。

- Part of meta-linaro layer:
 - `git://git.linaro.org/openembedded/meta-linaro.git`
- `TCMODE = "external-linaro"`
- `EXTERNAL_TOOLCHAIN = "/path/to/linaro"`
- Latest binaries are ARM hardfp:
 - `ELT_TARGET_SYS ?= "arm-linux-gnueabi"`
 - Otherwise version agnostic - `ELT_VER_*`

Pre-build native eSDK

输入第三方lib库native sdk包，通过eSDK模式编译成.sh格式，支持安装到规定的目录下。

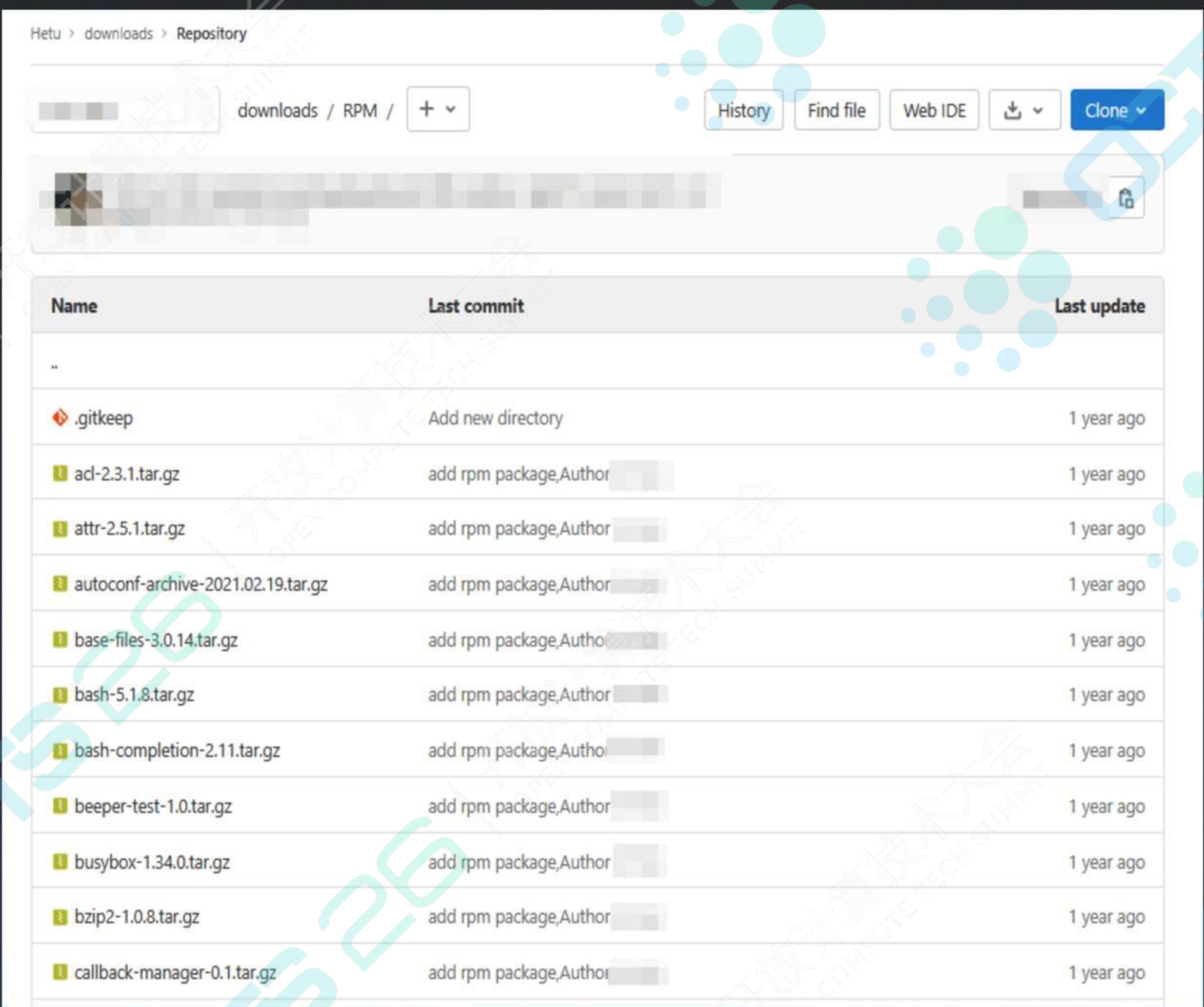


Git仓配置及规则

3rd lib 管理

- ✓ 独立存储空间: downloads/RPM文件夹下存放各lib的预编译RPM包;
- ✓ 编译构建: 通过编译命令参数, 选择对应rpm.bb编译文件, 加载对应lib的RPM包;
- ✓ lib更新:
 - 修改lib包源码, 重新构建RPM包;
 - 更新downloads/RPM下的RPM包;
 - 重新编译BMC

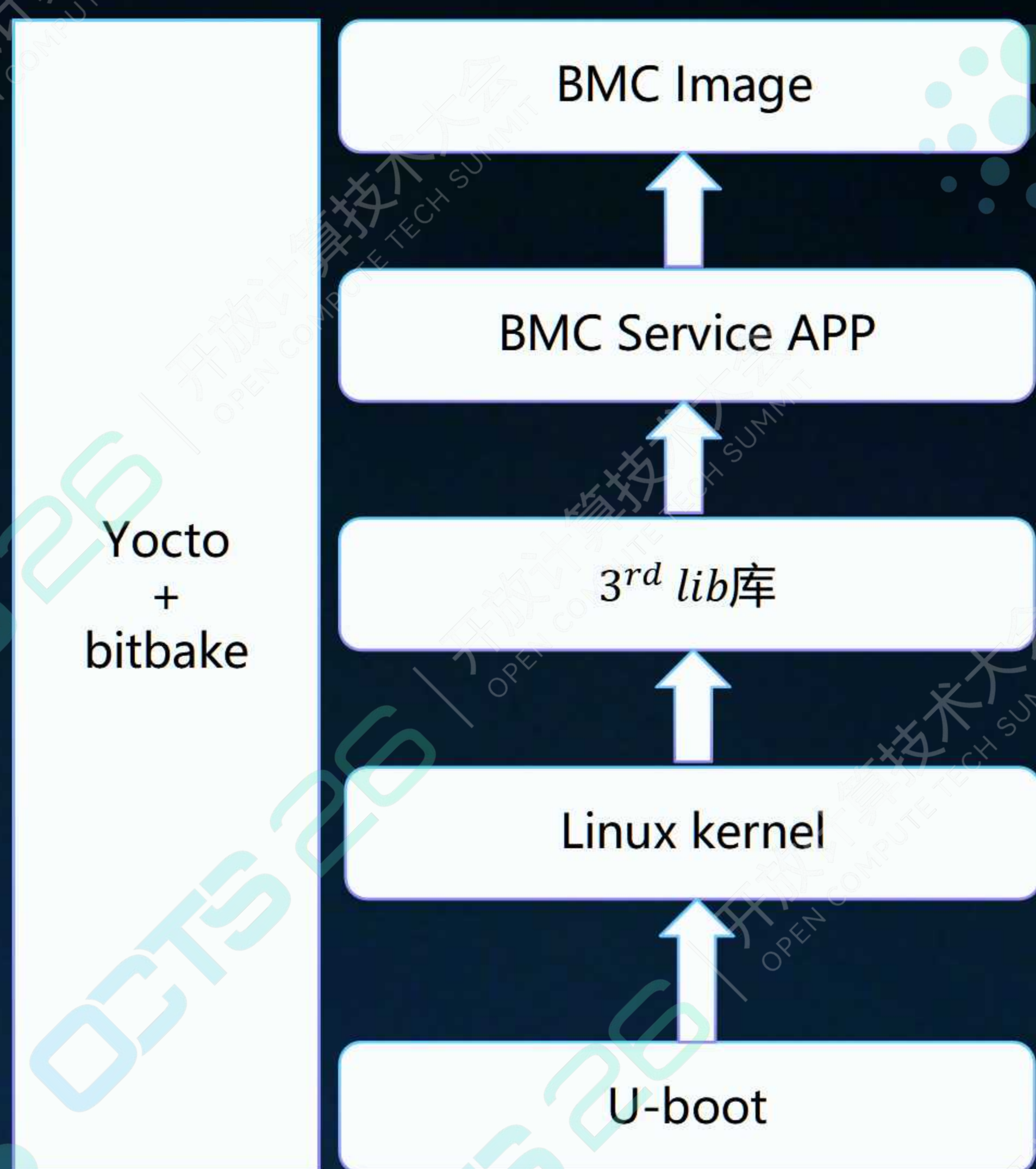
RPM git仓独立配置



Name	Last commit	Last update
..		
.gitkeep	Add new directory	1 year ago
acl-2.3.1.tar.gz	add rpm package,Author	1 year ago
attr-2.5.1.tar.gz	add rpm package,Author	1 year ago
autoconf-archive-2021.02.19.tar.gz	add rpm package,Author	1 year ago
base-files-3.0.14.tar.gz	add rpm package,Author	1 year ago
bash-5.1.8.tar.gz	add rpm package,Author	1 year ago
bash-completion-2.11.tar.gz	add rpm package,Author	1 year ago
beeper-test-1.0.tar.gz	add rpm package,Author	1 year ago
busybox-1.34.0.tar.gz	add rpm package,Author	1 year ago
bzip2-1.0.8.tar.gz	add rpm package,Author	1 year ago
callback-manager-0.1.tar.gz	add rpm package,Author	1 year ago

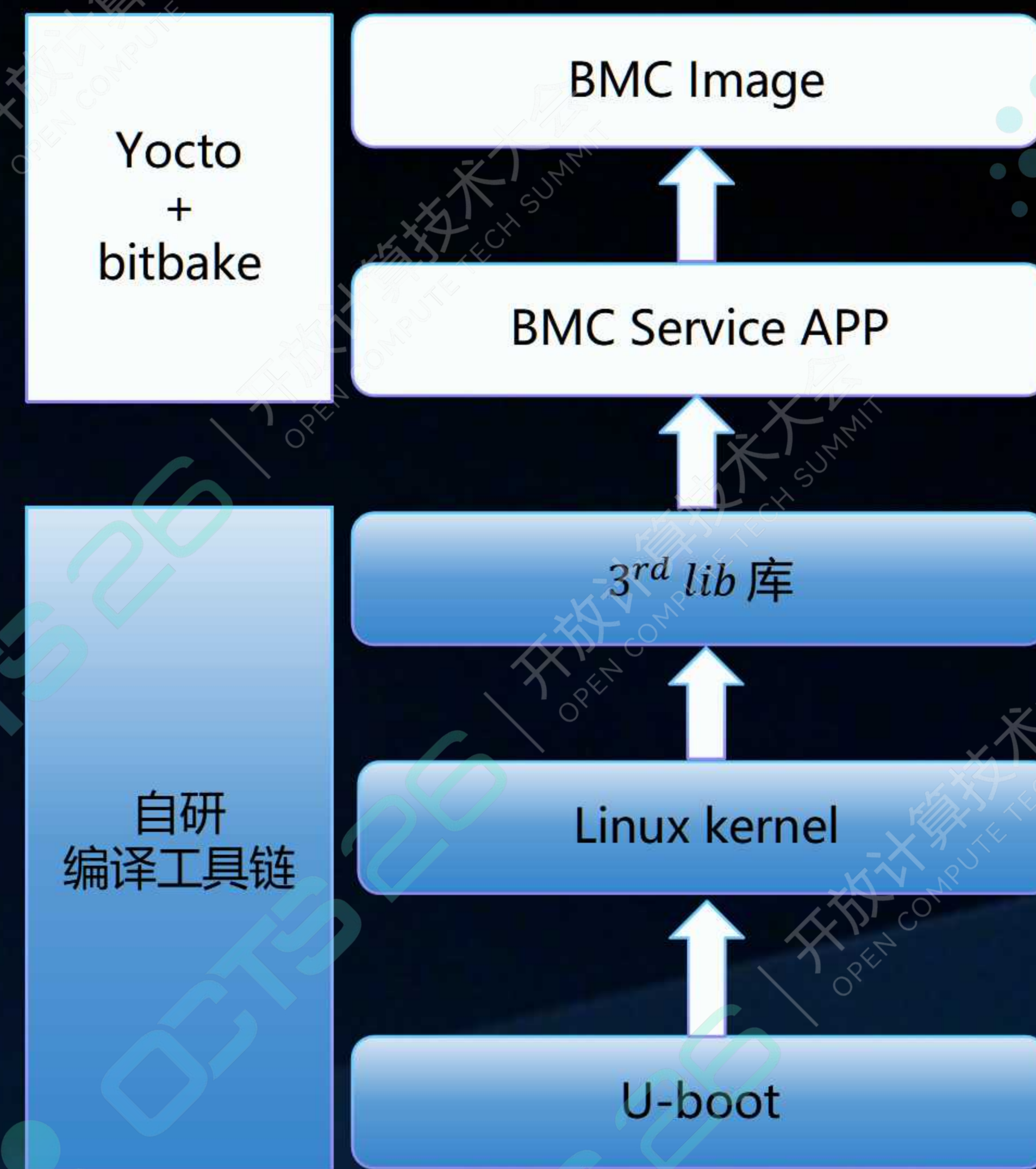
优化后收益

总结：原生openBMC构建框架中，BMC业务层和平台层严重耦合，只能通过yocto框架进行串行顺序构建，构建效率低下；通过自研编译工具链，实现CloudBMC平台U-boot/Linux kernel/第三方lib库预构建，和BMC业务层解耦，提升构建效率80%+



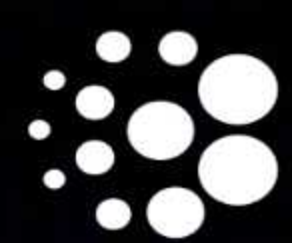
优化前流程

串行
顺序构建



优化后流程

并行
预构建



OCTS



固件产业技术创新联盟



芯尖信息



产品归一化——Onelimage

产品开发过程中痛点



工程文件互相独立

每个项目工程代码、固件交付件各一套，维护困难和投入成本高



新产品适配工作重复

新产品软件适配工作内容高度重复



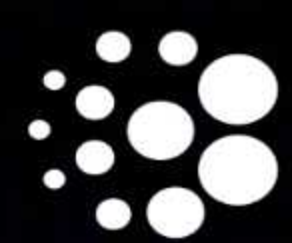
功能开发和移植麻烦，重复测试

功能代码移植对开发和测试的消耗极大



分支多，维护成本高

按分支维护项目代码，分支越来越多，同一个问题往往反复同步



OCTS



固件产业技术创新联盟



芯尖信息

Onelmage优化1-软硬件约束



BOARD ID要求

硬件为每个机型提供不同的
BOARD ID供BMC区分



I2C设计约束

关键器件的I2C拓扑设计在
各机型上保持一致



IO归一化设计

各机型上BMC关键IO pin脚
的功能效果保持一致



器件选型约束

关键器件如flash、
emmc、phy等选用型号在
规定清单内

OnelImage优化2-软件方案

01

u-boot层

定义产品ID

添加DTB映射

02

kernel层

修改dts设备树

根据硬件链路定义节点

03

应用层

配置产品ID

加载对应配置文件

Onelimage优化2-软件方案

✓ 硬件初始化加载

1. 启用fitimage特性，BOOT阶段根据产品ID动态加载DTB文件；
2. 优化社区镜像打包脚本，支持镜像编译打包多分DTB文件。

✓ 业务初始化加载

1. 业务配置信息根据产品ID动态加载；
2. 优化文件挂载顺序，实现动态文件拷贝初始化；
3. 整改业务流程，关键数据配置化处理；
4. 支持传感器、FRU依据产品动态加载显示。

✓ 启动顺序优化

1. 调整原app启动顺序，保障业务可以正确获取硬件配置信息；
2. BMC启动时间小于90s。



Onelmage收益

INTEL

ICX / EGS / BHS

AMD

Milan / Genoa / Turin

各类机型复用

覆盖主流产品矩阵，满足多样化部署需求

Hygon

飞腾

龙芯

Others

多形态覆盖

服务器、融合交换机、DPU卡、UBB等全品类支持

适配机型

60+

统一基线维护

同一基线、同一固件文件兼容全部产品，降低维护复杂度

未来计划

平台扩展计划

- Intel & AMD同国产CPU平台进一步合
- ARM和Risc-V CPU纳入兼容

AI+Agent赋能

- 深度融合AI coding
- 实现新机型智能适配
- 平台版本开发测试一体化
- 软件交付件智能输出

构建智能、可持续演进的统一BMC平台

THANKS

THANKS

THANKS

THANKS