



开放原子开源基金会 OpenHarmony开发者大会 2023

OpenHarmony 新开发框架，开发更便捷



强波

OpenHarmony PMC 成员 & 应用框架SIG Leader

目录 Contents

01 OpenHarmony 应用框架：范围与内容

02 OpenHarmony 3.2 应用框架新增特性

- 模块化开发与应用包结构
- Stage应用开发模型
- ArkUI改进

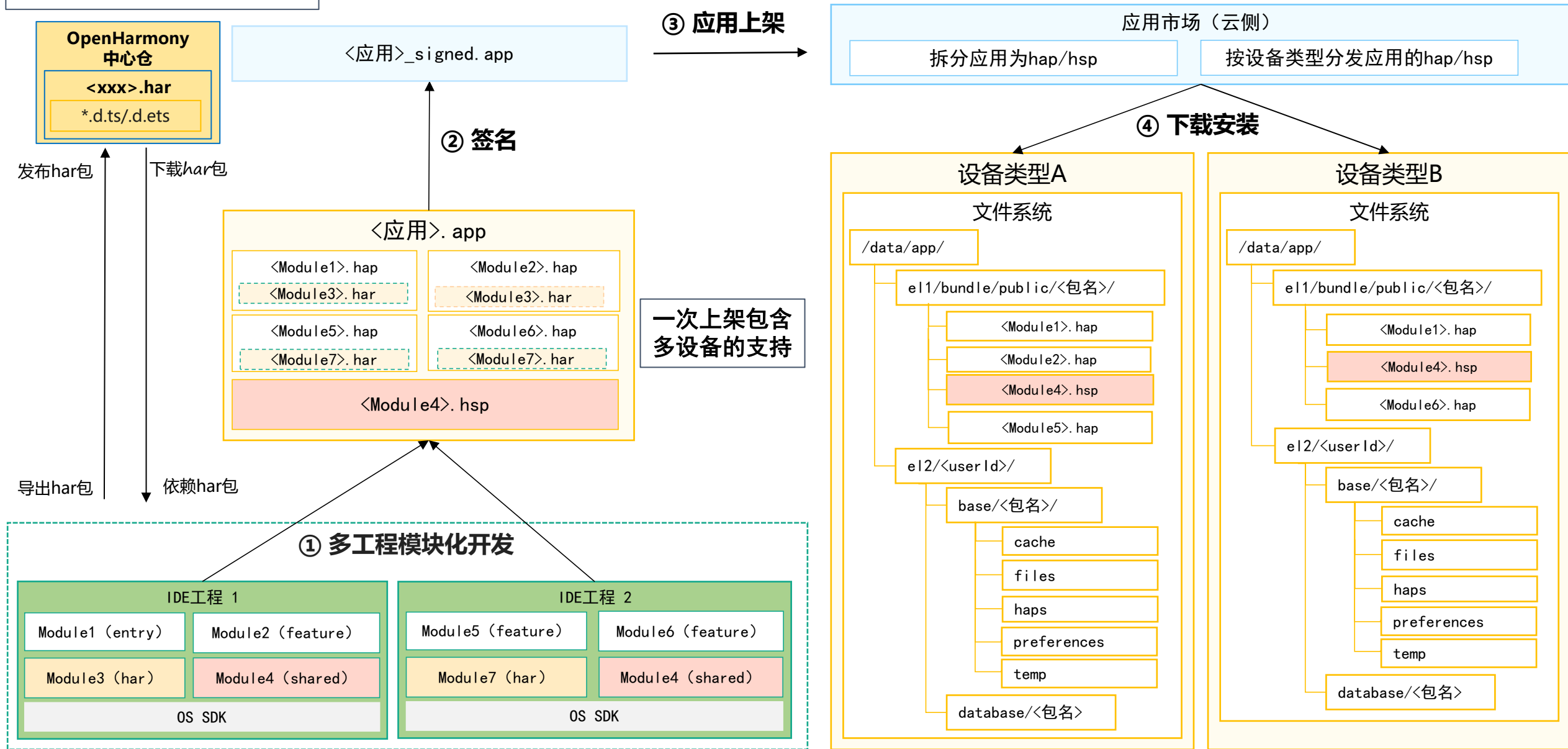
03 OpenHarmony 与三方框架

OpenHarmony 应用框架：范围与内容



OpenHarmony应用程序从开发到安装整体视图

中心仓承载了三方扩展能力





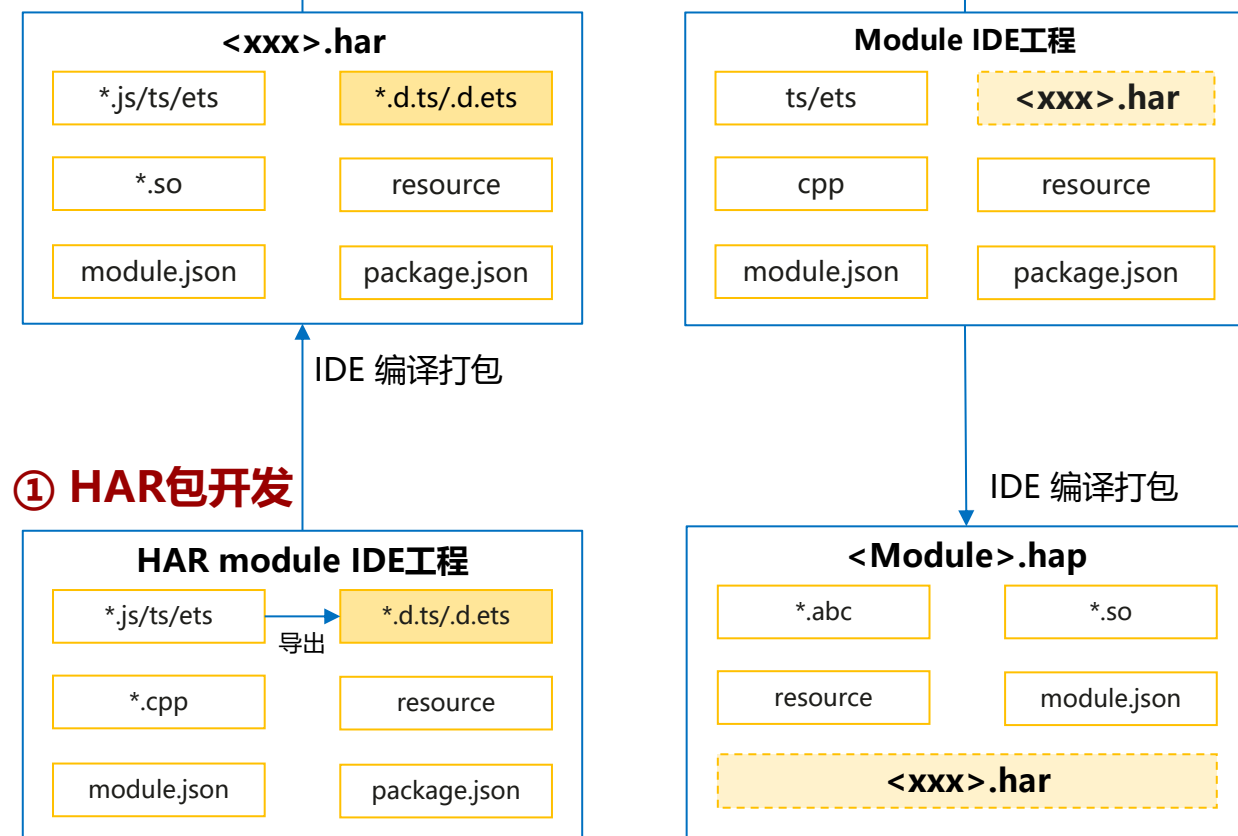
OpenHarmony 应用包结构关键概念

名称	使用场景	业务规则
HAP（Harmony Ability Package）	应用安装与更新的最小单元	<div><div>✓</div> 必须包含UIAbility或者ExtensionAbility</div> <div><div>✓</div> 可以包含资源</div> <div><div>✓</div> 可以包含*.so文件</div> <div><div>✓</div> 可以在设备上独立安装/运行</div>
HAR（Harmony Archive）	静态共享包，编译态复用	<div><div>✗</div> 共享包不包含UIAbility或者ExtensionAbility</div> <div><div>✓</div> 可以包含资源</div> <div><div>✓</div> 可以包含*.so文件</div> <div><div>✗</div> 不能在设备上单独安装/运行</div> <div><div>✓</div> 可发布到OpenHarmony中心仓供其他应用使用</div>
<div><div>★</div> HSP（Harmony Shared Package）</div>	动态共享包，运行时复用	<div><div>✗</div> 共享包不包含UIAbility或者ExtensionAbility</div> <div><div>✓</div> 可以包含资源</div> <div><div>✓</div> 可以包含*.so文件</div> <div><div>✗</div> 不能在设备上单独安装/运行</div>
App Pack（Application Package）	应用上架格式，可以包含多设备的编译产物（HAP、HAR、HSP）	应用上架格式

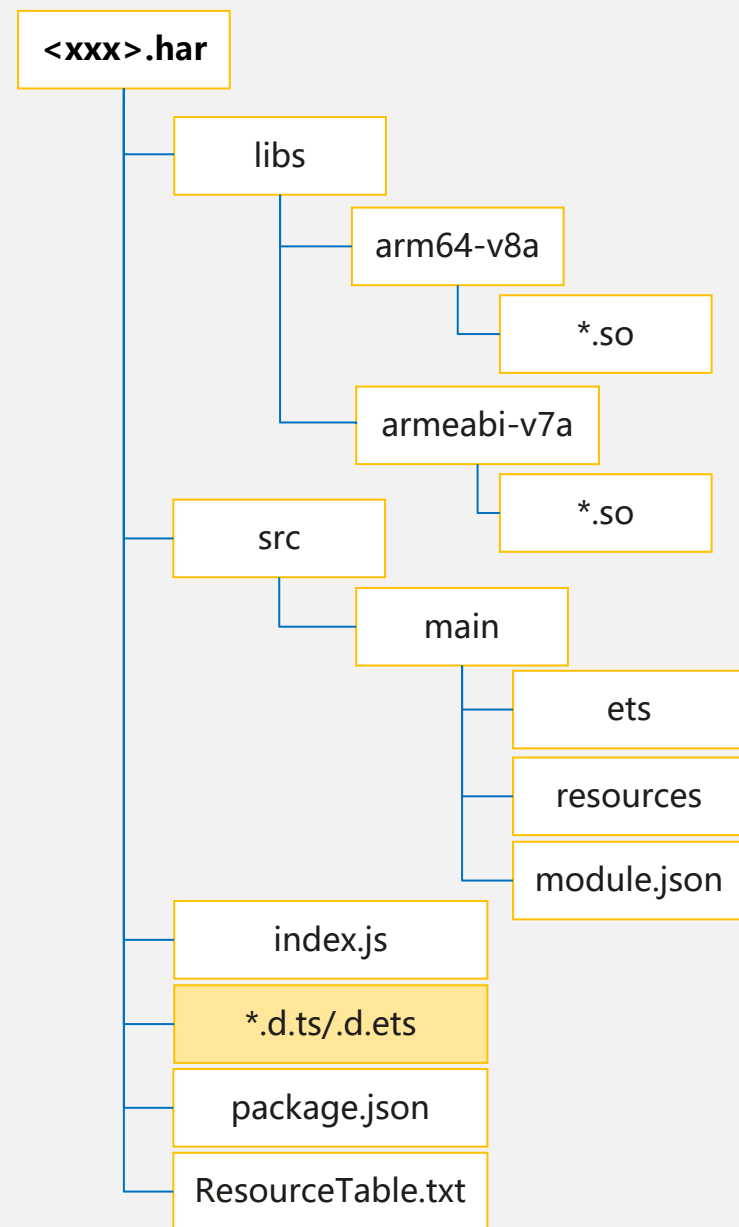
Harmony ARchive (HAR)包介绍

② HAR包发布

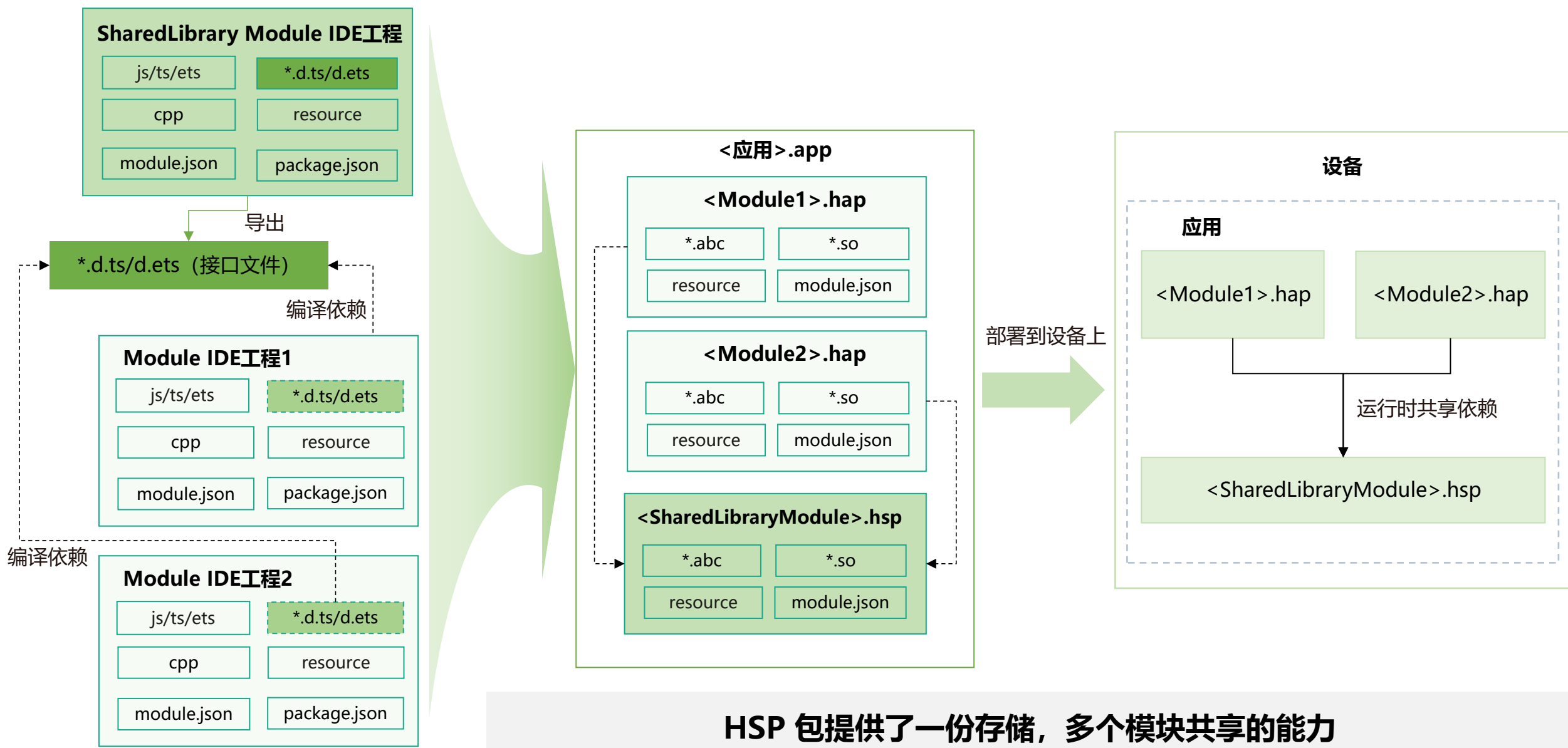
③ HAR开发依赖



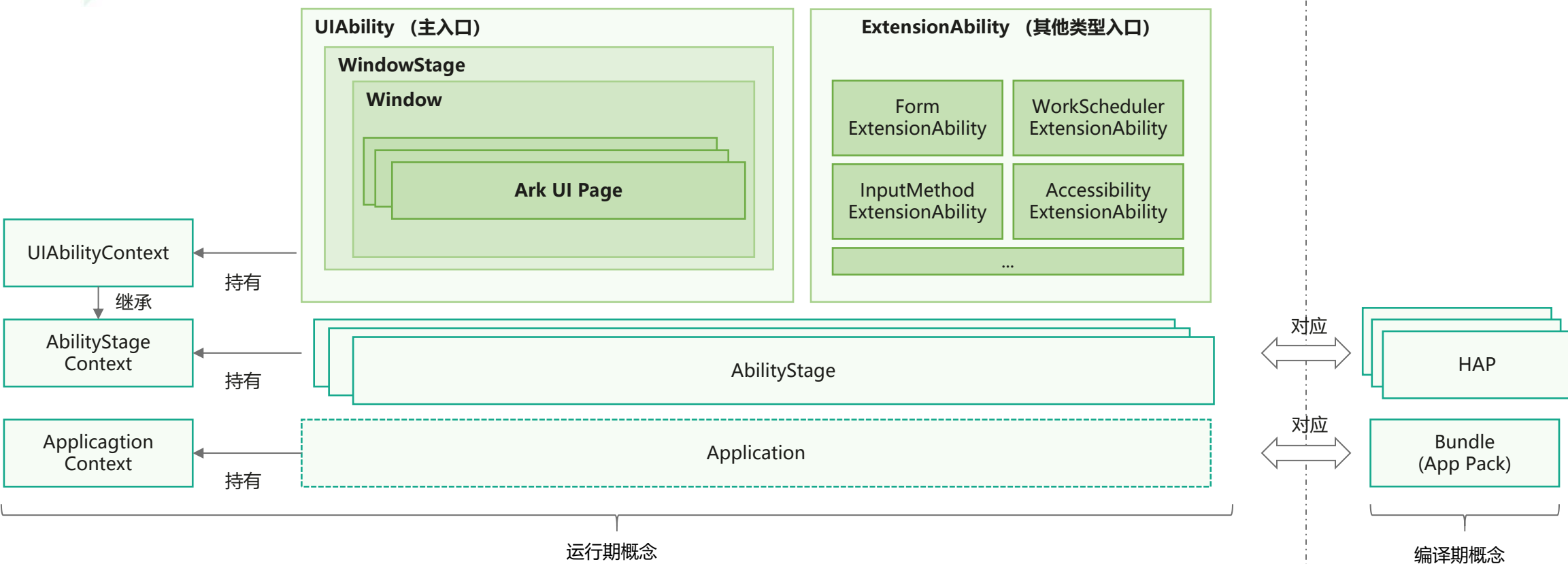
HAR包结构



Harmony Shared Package (HSP) 包介绍

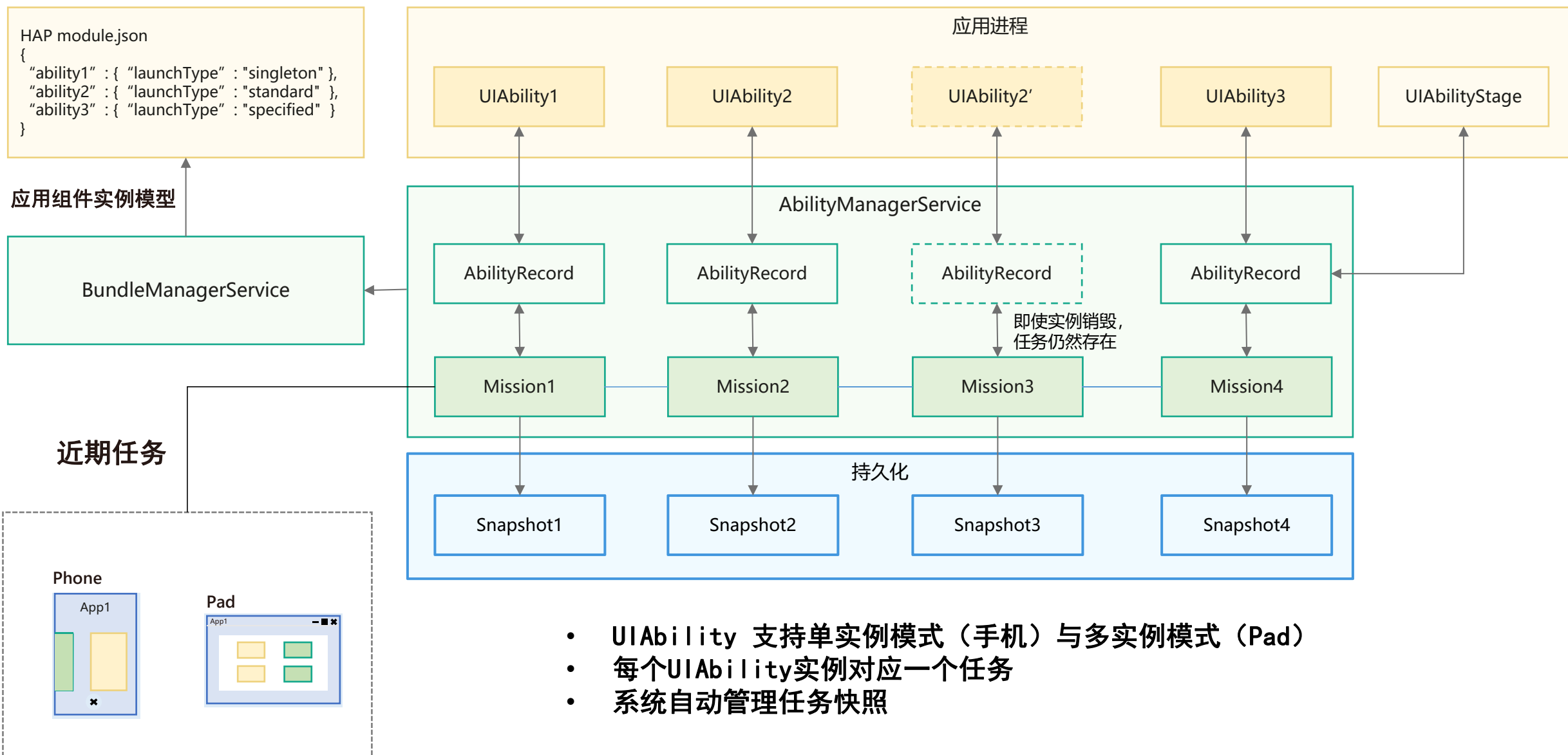


Stage模型 - 主要结构

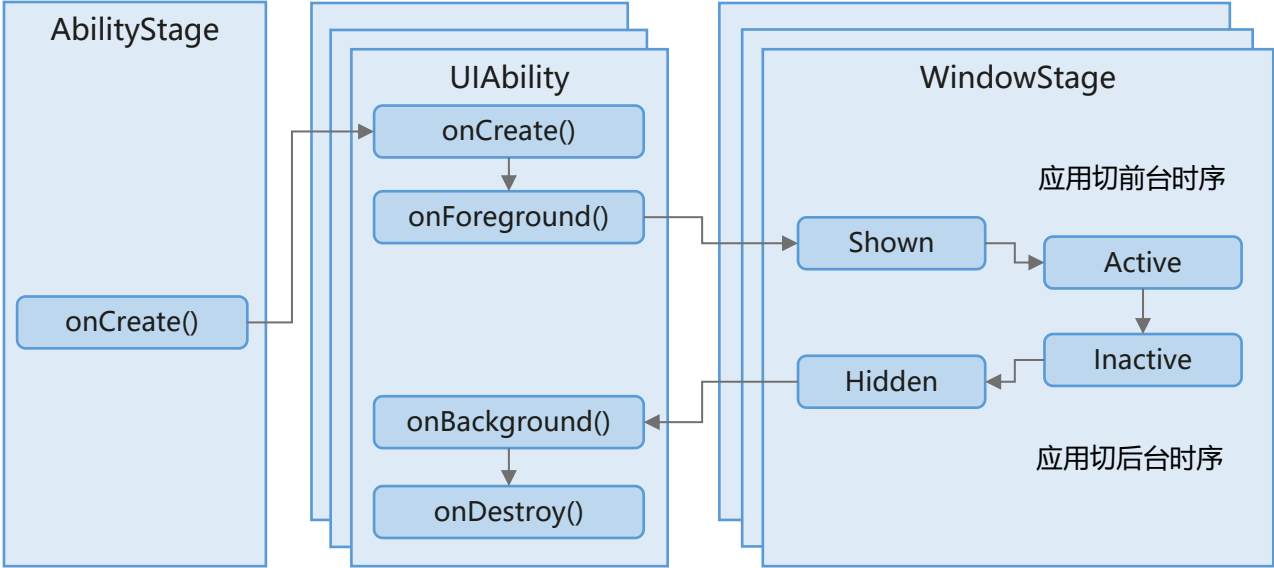


概念	描述
AbilityStage	对应HAP的运行期类，在HAP首次加载到进程中时创建
UIAbility	提供应用组件生命周期回调，持有UIAbilityContext
ExtensionAbility	基于场景的服务扩展能力，持有 XXXExtensionAbilityContext
XXXContext	提供对应类型开发者可以调用的各种能力
WindowStage/Window	窗口生命周期回调及窗口管理能力
Ark UI Page	用户界面内容

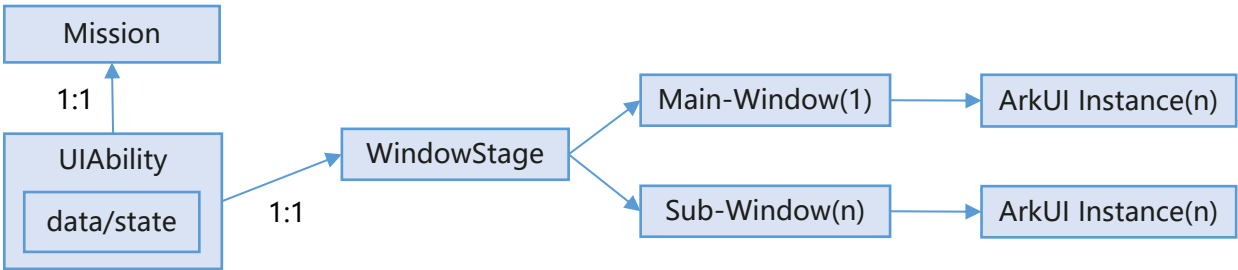
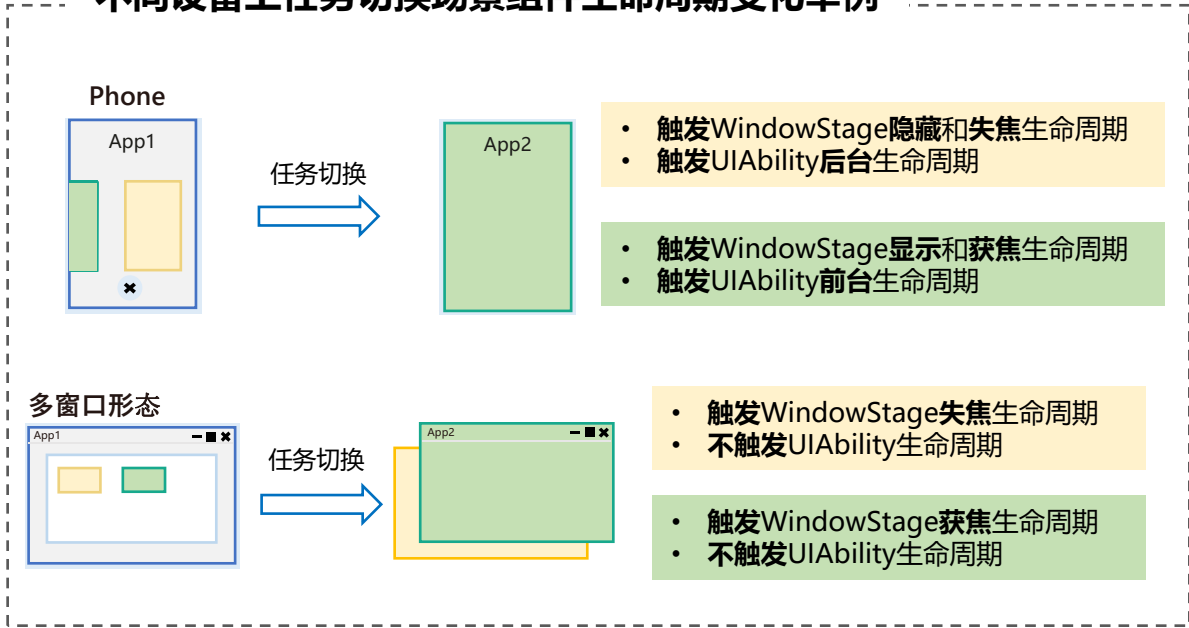
Stage模型 - UIAbility组件实例与任务



应用组件与窗口组件生命周期分离

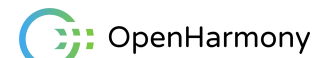


不同设备上任务切换场景组件生命周期变化举例



Stage模型通过应用组件和窗口的生命周期解耦，统一了多设备形态下的开发模型

OpenHarmony 3.2版本ArkUI新增特性一览



- 既存
- 新增
- 改进

LoadingProgress	Canvas	Shape	ImageAnimator	TextInput	Search	Row	Divider
AlphabetIndexer	Hyperlink	Badge	Progress	ActionSheet	Polyline	Swiper	
TimePicker	StepperItem	Column	QRCode	ColumnSplit	Flex	Slider	
Blank	RelativeContainer	Select	Text	Navigation	Polygon	Rect	
TextArea	Refresh	Grid	Button	LazyForEach	CheckboxGroup	DataPanel	Web
Counter	ListItem	Navigator	Panel	TabContent	Video	AlertDialog	Stack
Gauge	TextTimer	Tabs	Checkbox	TextClock	RichText	List	Span
Stepper	XComponent	ForEach	GridContainer	GridRow	Rating	PatternLock	
DatePicker	RowSplit	Marquee	GridItem	Radio	Circle	SideBarContainer	Line
TextPicker	Scroll	Path	Toggle	Image	ScrollBar	Ellipse	GridCol

组件总数 76

改进组件 26

新增组件 5

XComponent: 支持C++/ArkTS混合开发, 满足自渲染场景

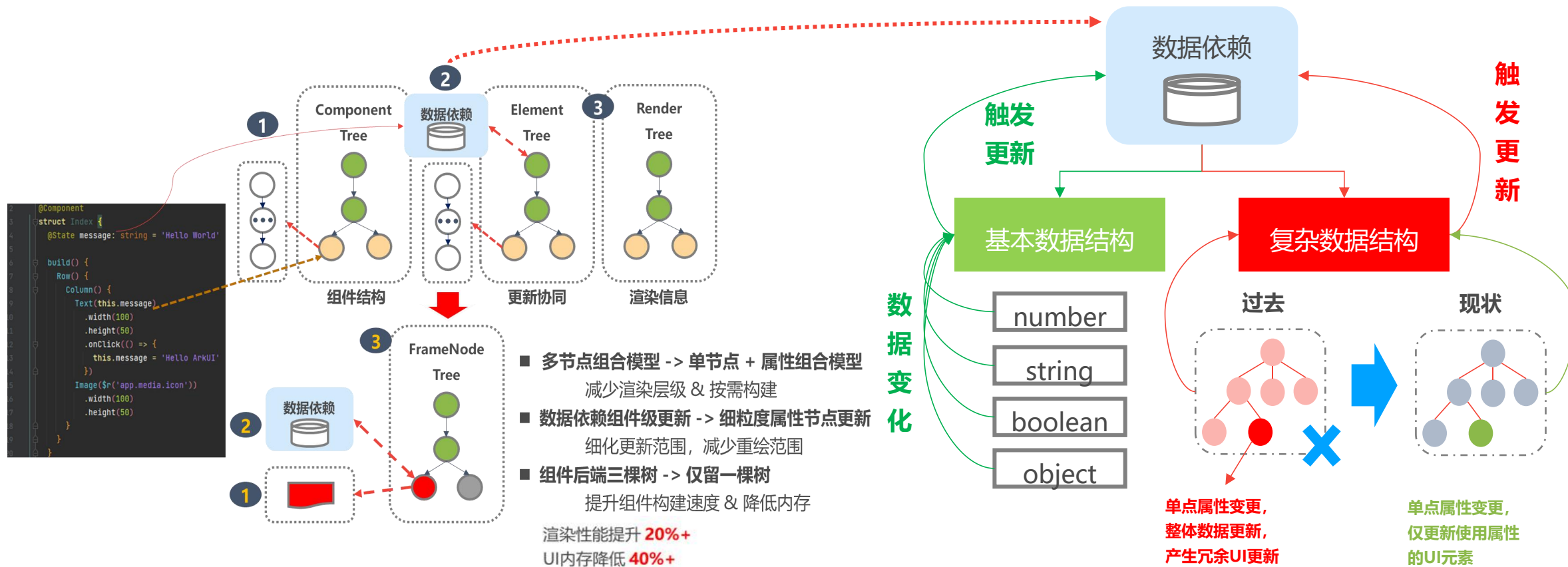
Canvas: 提供自绘制能力, 满足开发者自绘制场景

RelativeContainer: 相对布局能力, 降低页面元素对齐嵌套层级

GridRow: 提供页面栅格布局属性设置

GridCol: 用于栅格布局中设置子布局的大小、位置、顺序

最小化更新技术提高布局性能、降低应用功耗

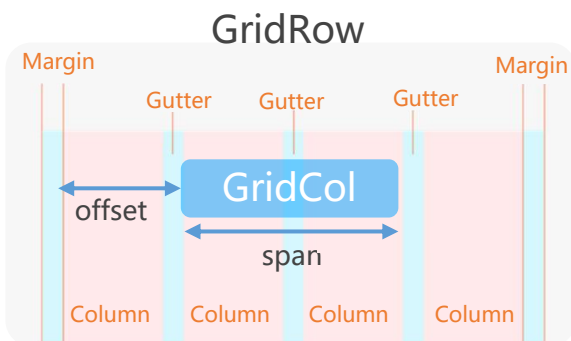


通过降低状态管理数据复杂度, 进一步减少UI更新范围, 提升渲染性能 ~20%

自适应栅格化布局能力



功能介绍



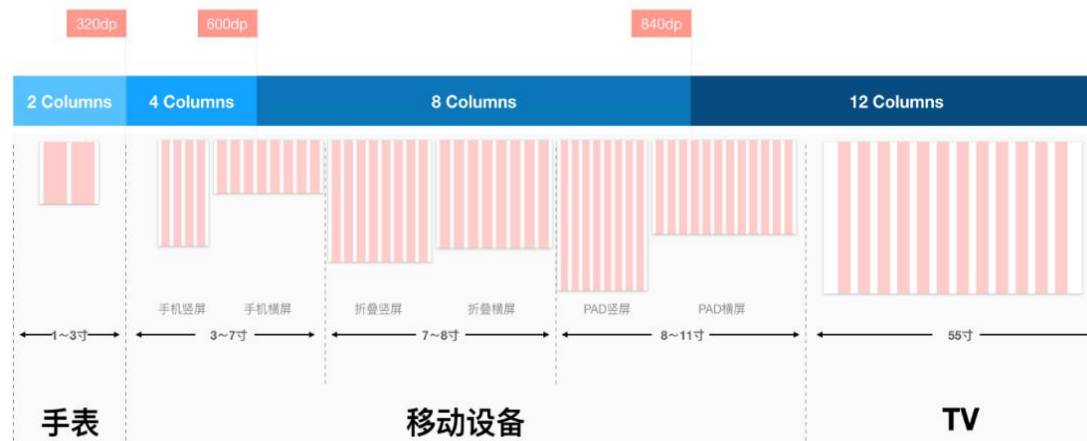
GridRow参数:

gutter	栅格布局间距
columns	设置栅格数量
breakpoints	设置栅格布局不同断点的尺寸
direction	栅格布局排列方向

GridCol参数:

span	GridCol宽度占用栅格数
offset	GridCol相对于前一个组件偏移数
order	GridCol排列序号, 整体按升序排列

设备分类与用法推荐



不同类型窗口推荐的栅格布局参数

水平宽度 (vp)	SizeType类型	Columns数量	默认Margin	默认Gutter	典型设备
[0, 320)	XS	2	12vp	12vp	智能手表
[320, 600)	SM	4	24vp	24vp	手机
[600, 840)	MD	8	32vp	24vp	平板竖屏、折叠屏展开形态
[840, +∞)	LG	12	48vp	24vp	平板横屏、PC

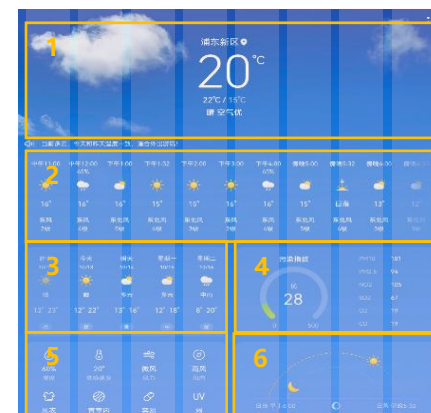
- 利用sizeType代替设备类型, 降低设备种类过多的困扰
- 针对SizeType设置属性, 实现一套代码描述多种布局样式

效果示例

手机 (4栅格)



折叠屏 (8栅格)



```
GridRow ( { columns: { sm: 4, md: 8, lg: 12 } } ) {
  GridCol ( { span: { sm: 4, md: 8, lg: 12 } } ) {
    //1 城市概览
  }
  GridCol ( { span: { sm: 4, md: 8, lg: 8 } } ) {
    //2 每小时天气
  }
  GridCol ( { span: 4, } ) {
    //3 每日天气
  }
  .....
}
```

OpenHarmony 3.2版本已支持大规模复杂应用开发

OpenHarmony 应用，卡片

程序框架

应用组件管理

- AbilityStage
- UIAbility
- XXXExtensionAbility

窗口管理

- WindowStage
- UI Pages
- 用户界面与业务逻辑解耦
- 支持多设备的统一生命周期

包管理

- 多工程、模块化 (module) 开发
- 静态 (har)、动态 (hsp) 共享包
- 一次上架，多端部署 (App Pack)

UI框架

ArkUI

- 组件增强
- 自渲染
- 栅格化布局
- 一多自适应
- 性能提升

OpenHarmony 拥抱开源社区，拥抱三方框架

三方框架 & 技术



OpenHarmony



欢迎开源社区一起共建OpenHarmony生态

https://gitee.com/openharmony/community/blob/master/sig/sig-appframework/sig-appframework_cn.md

THANK YOU



扫描二维码 关注官方公众号

【官网网址】 www.openharmony.cn