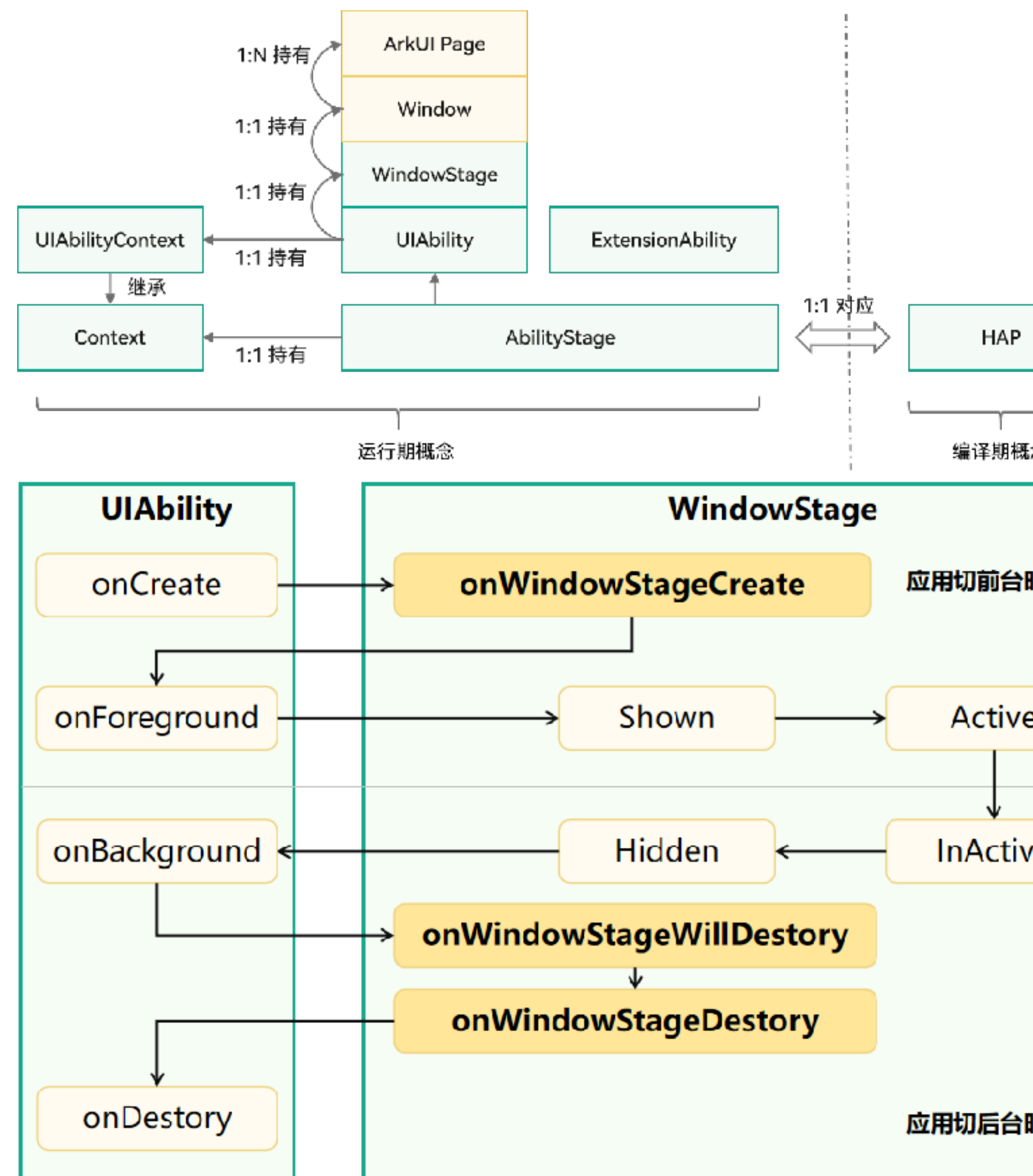
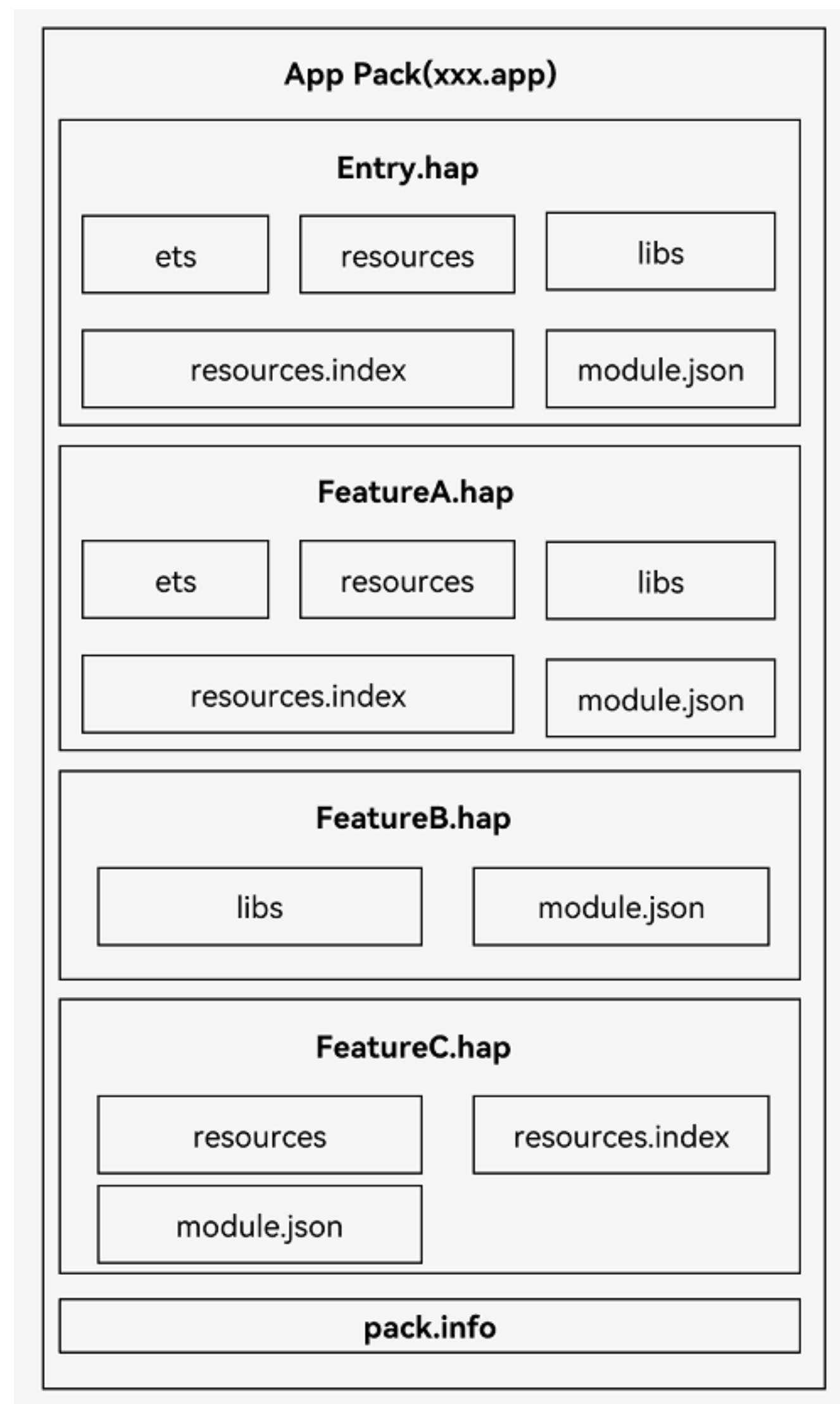


复习

快速入门

- App结构
- Stage模型
- WindowStage

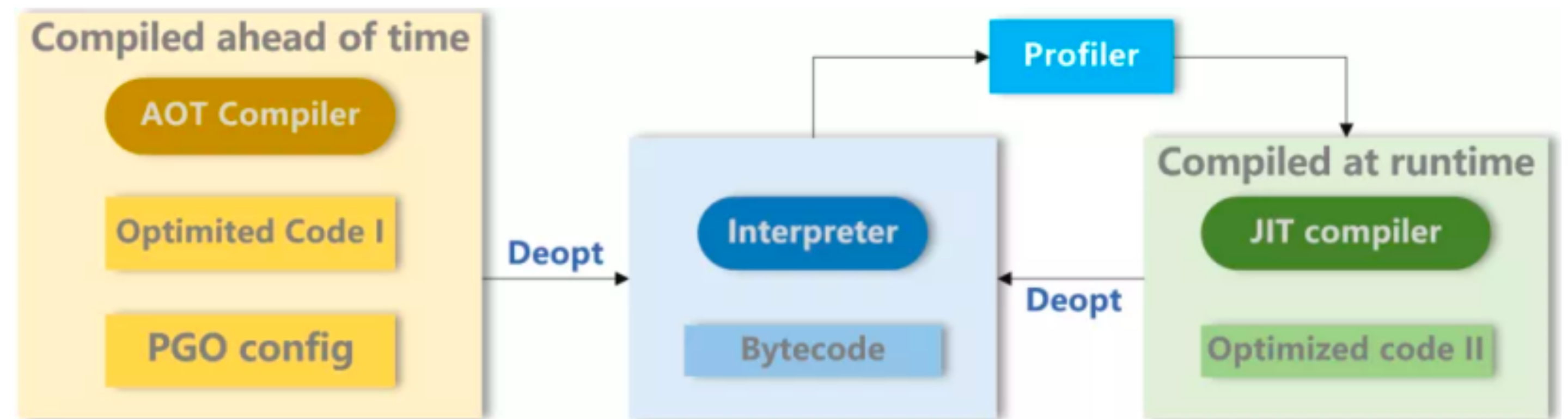
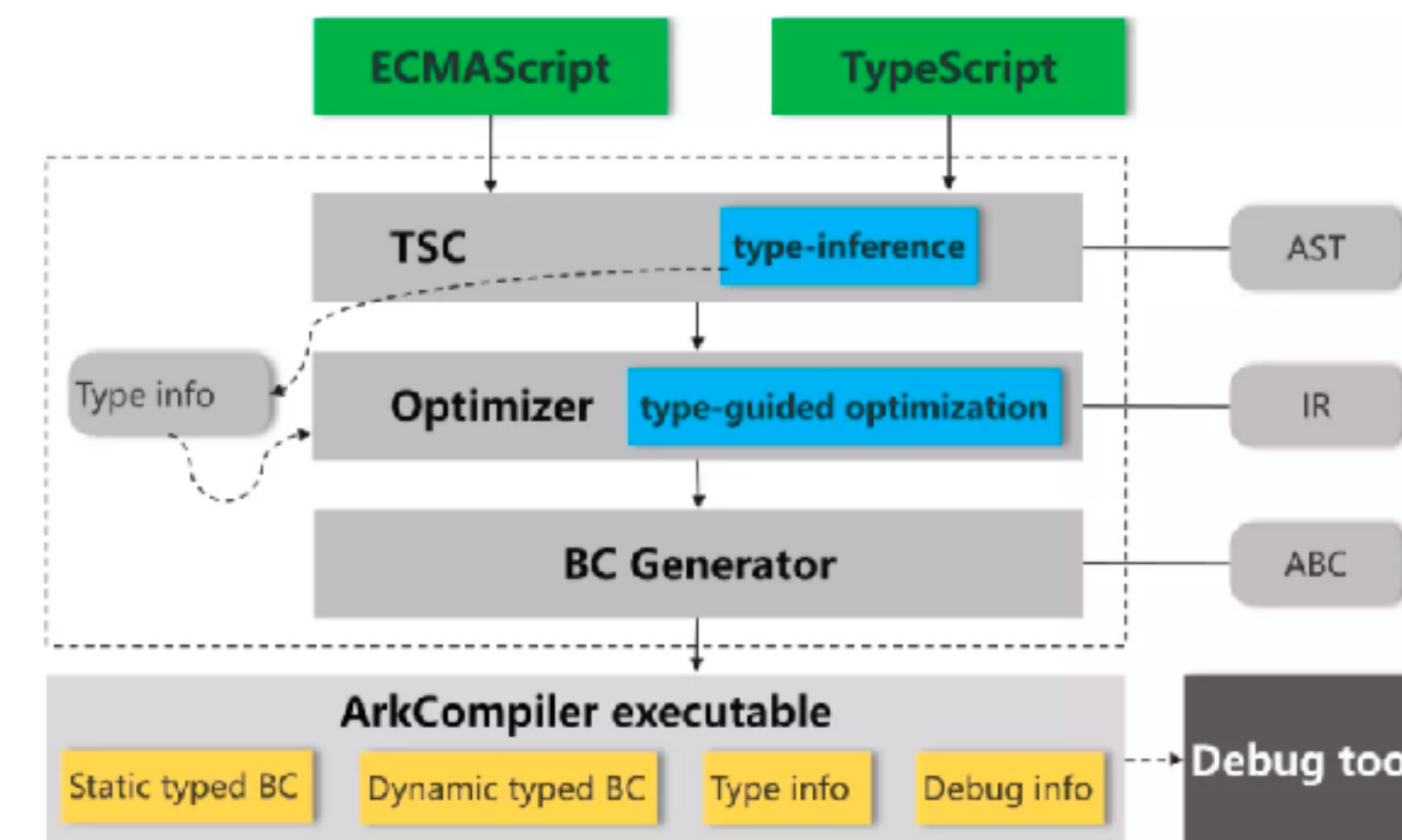
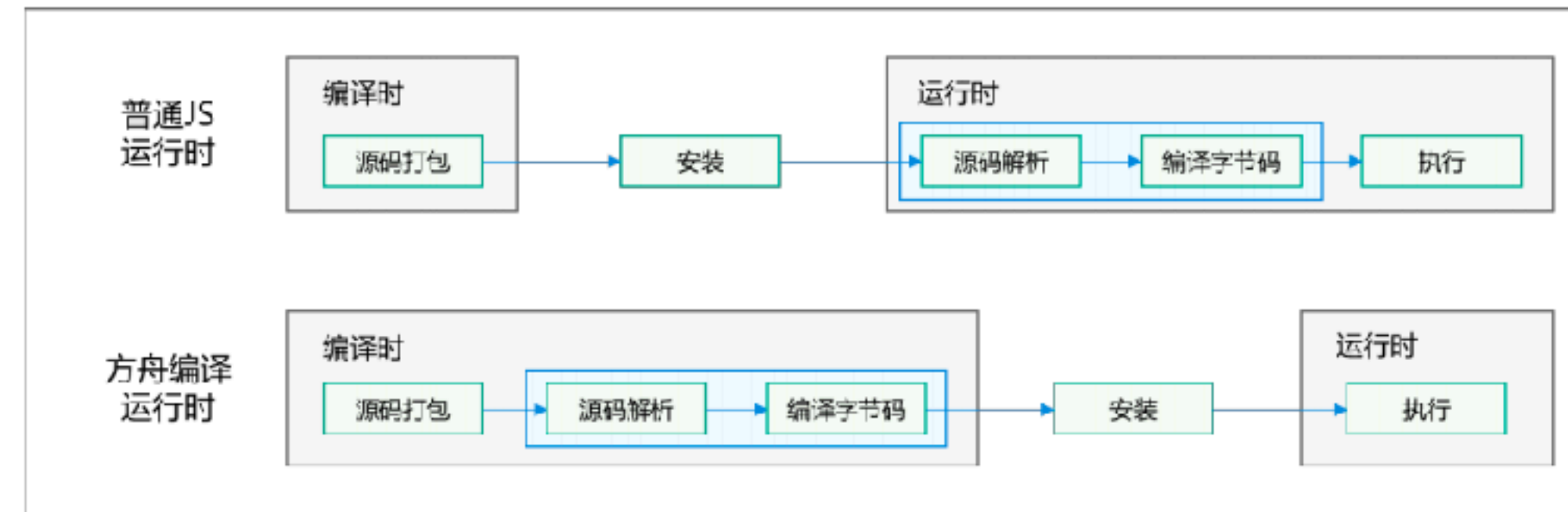


快速入门

- 启动模式
- Ability
- Want
- HAP、HSP、HAR区别

ArkCompiler和ArkRuntime

- 方舟编译运行时
- 类型推导
- AOT和JIT区别



ArkUI

- ArkUI框架

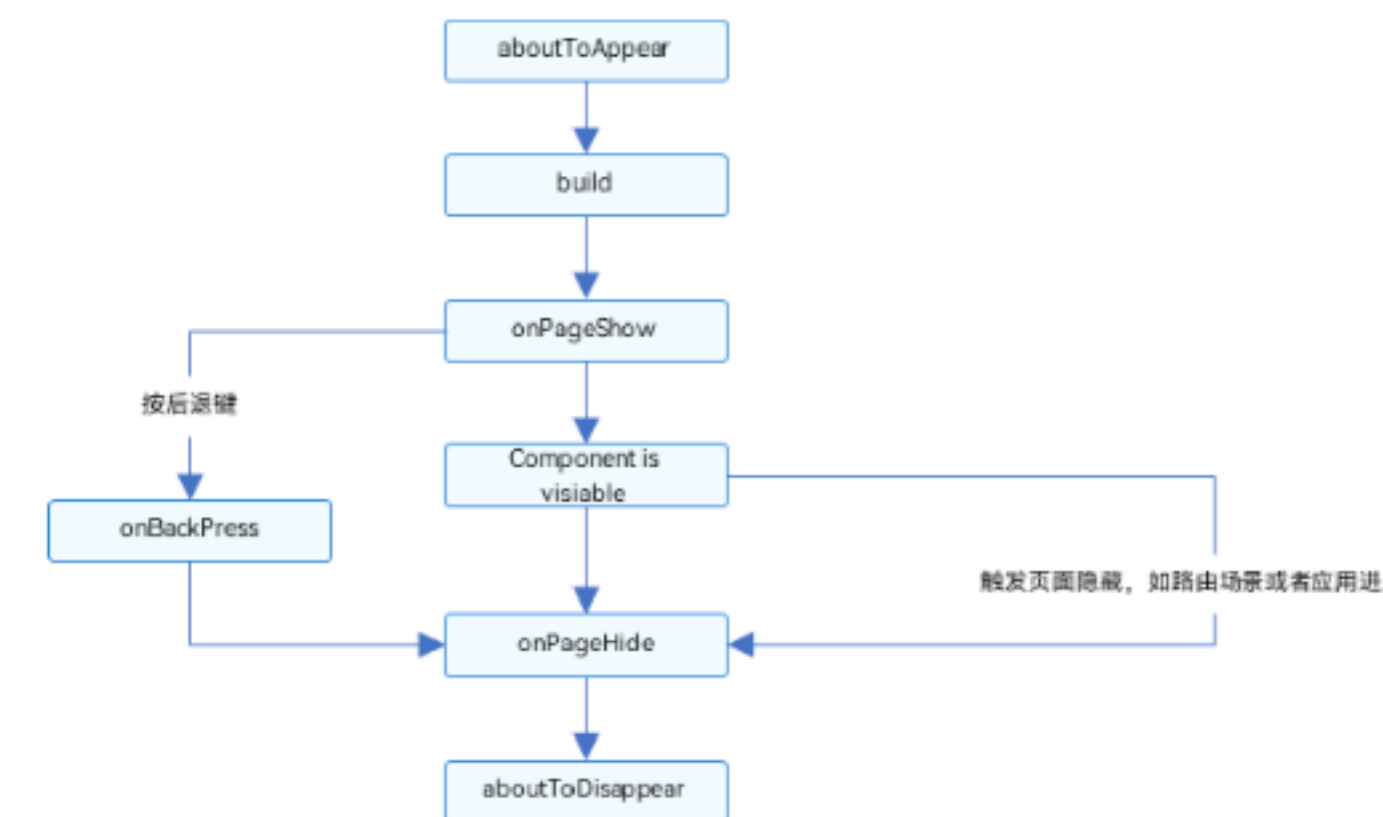
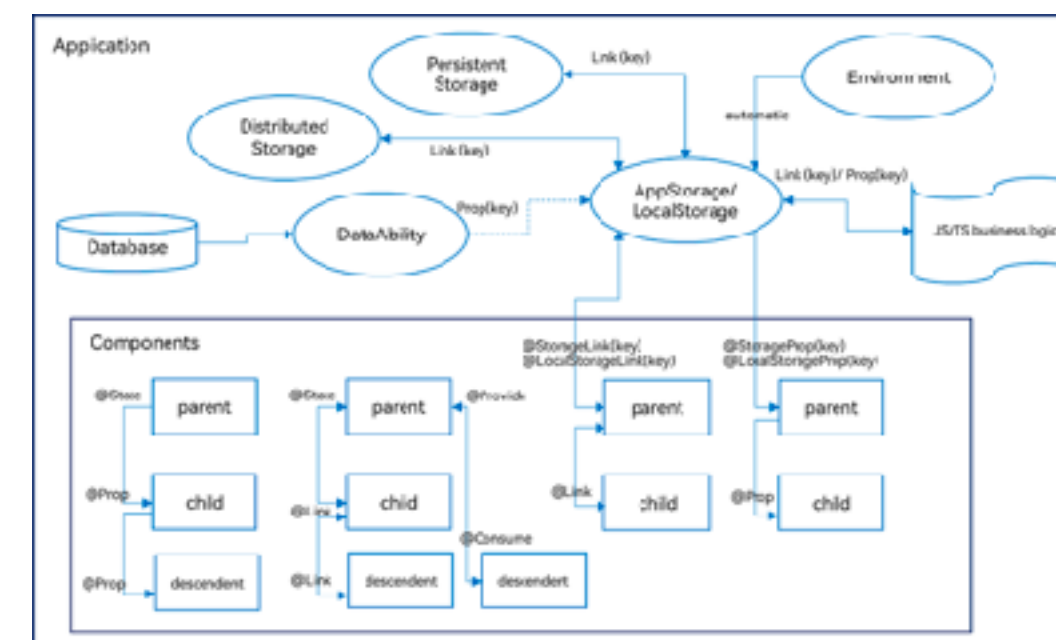
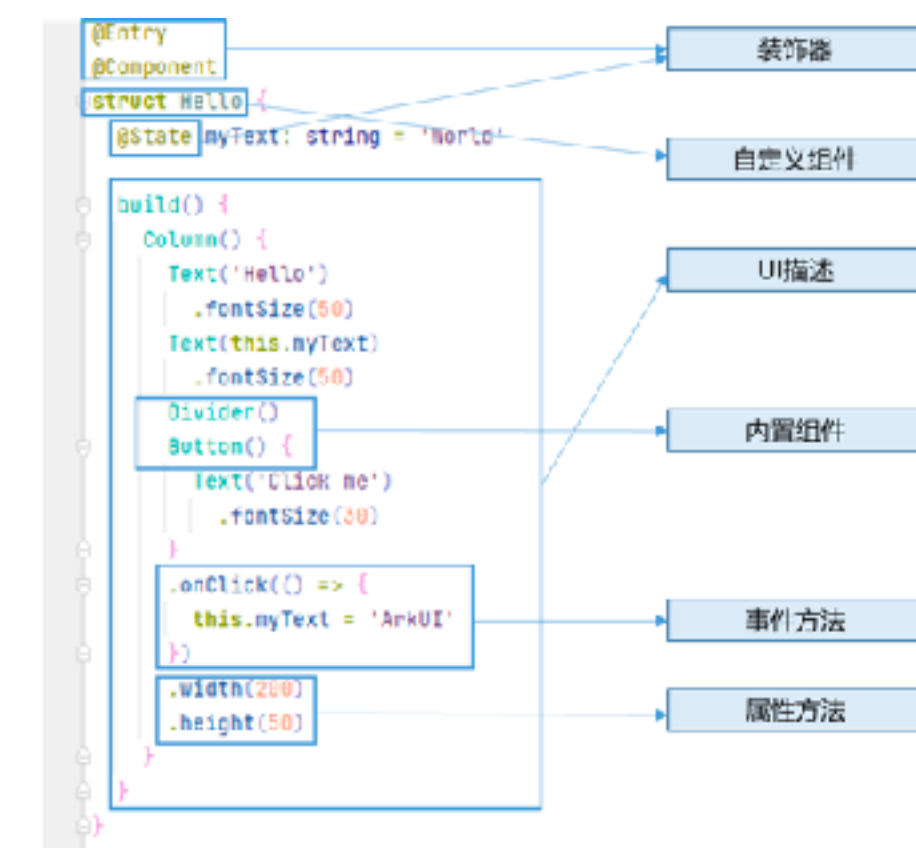
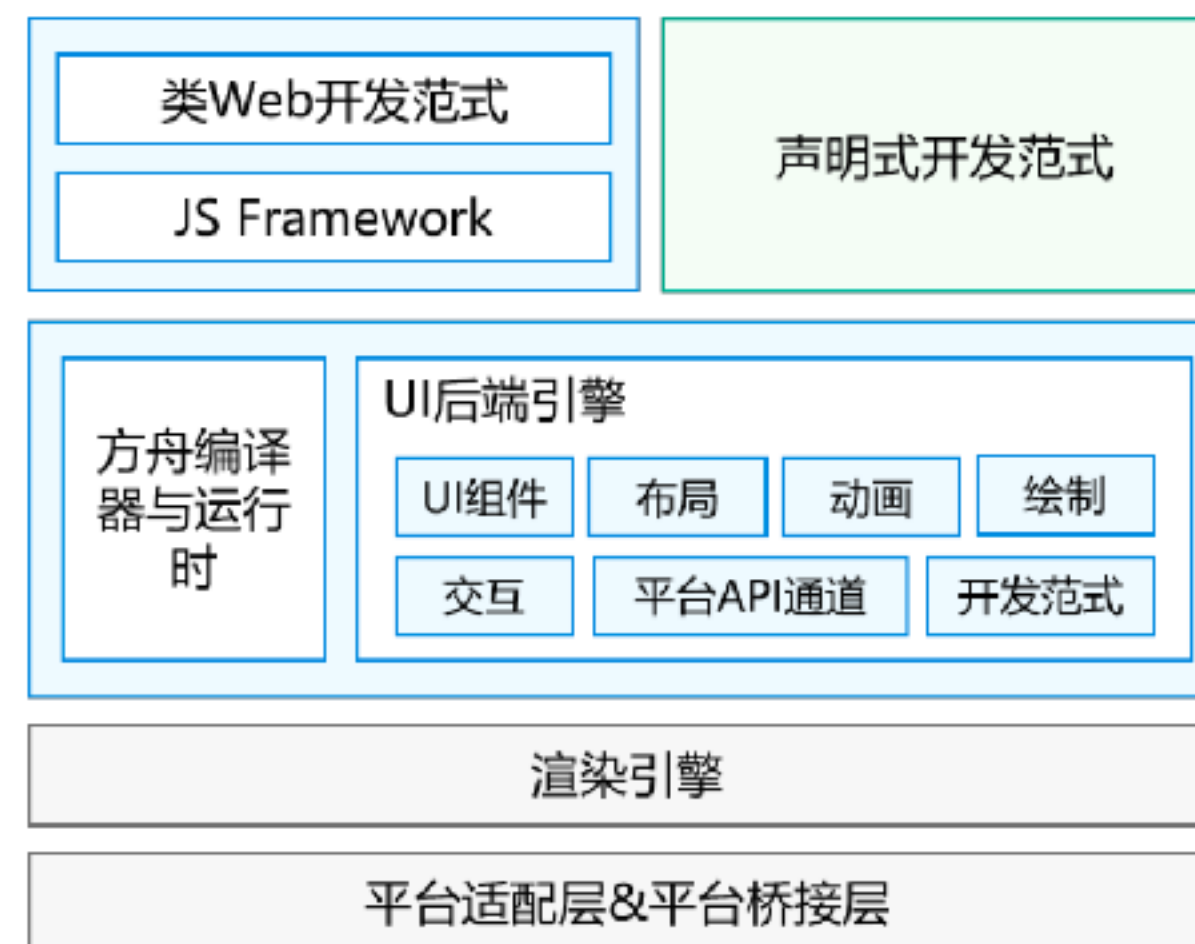
- UI示例

- 页面生命周期、组件生命周

- 状态管理

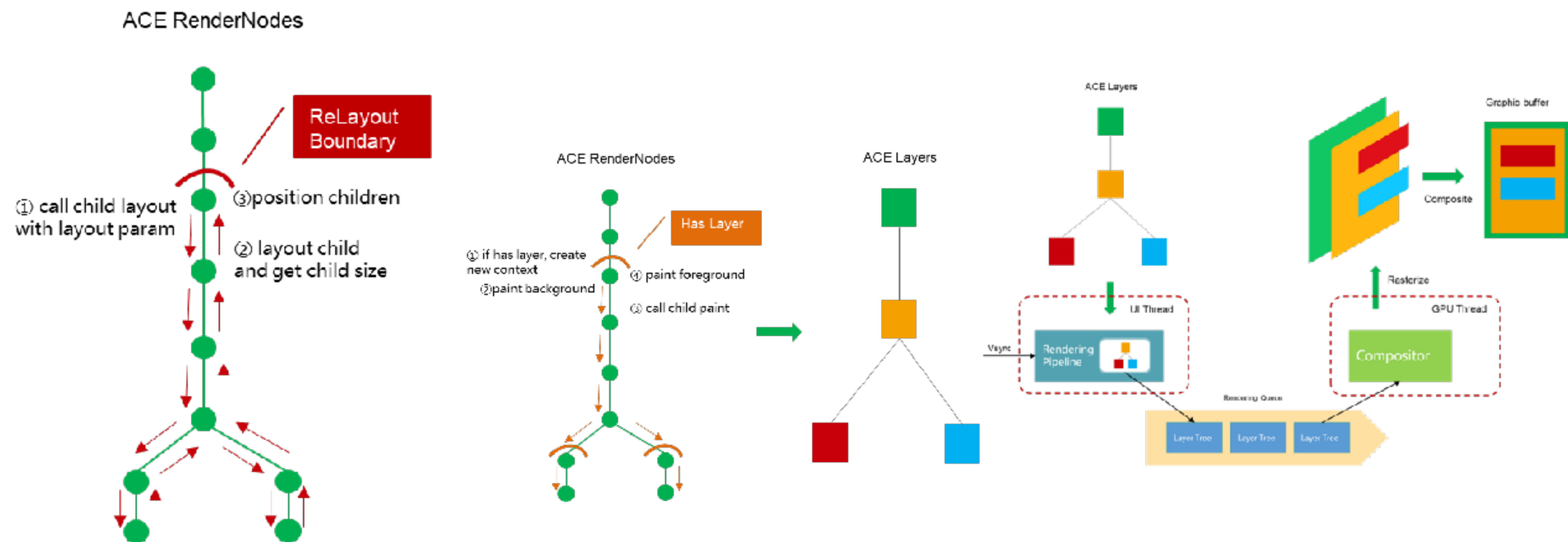
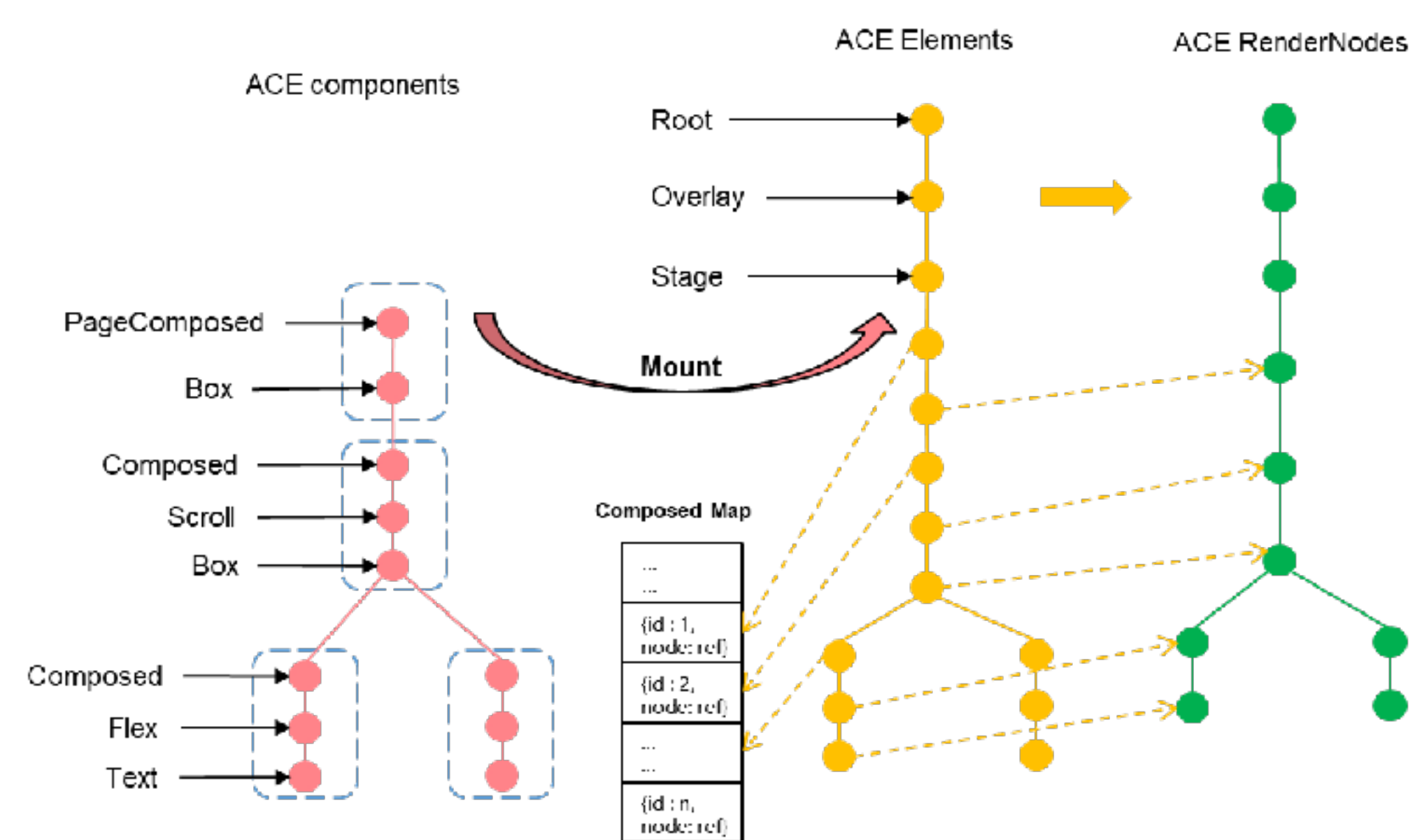
- 渲染控制

- 布局结构



ArkUI

- UI渲染
- Component树、Element树、Render树
- 布局的步骤
- 绘制
- 光栅化合成机制



ArkUI

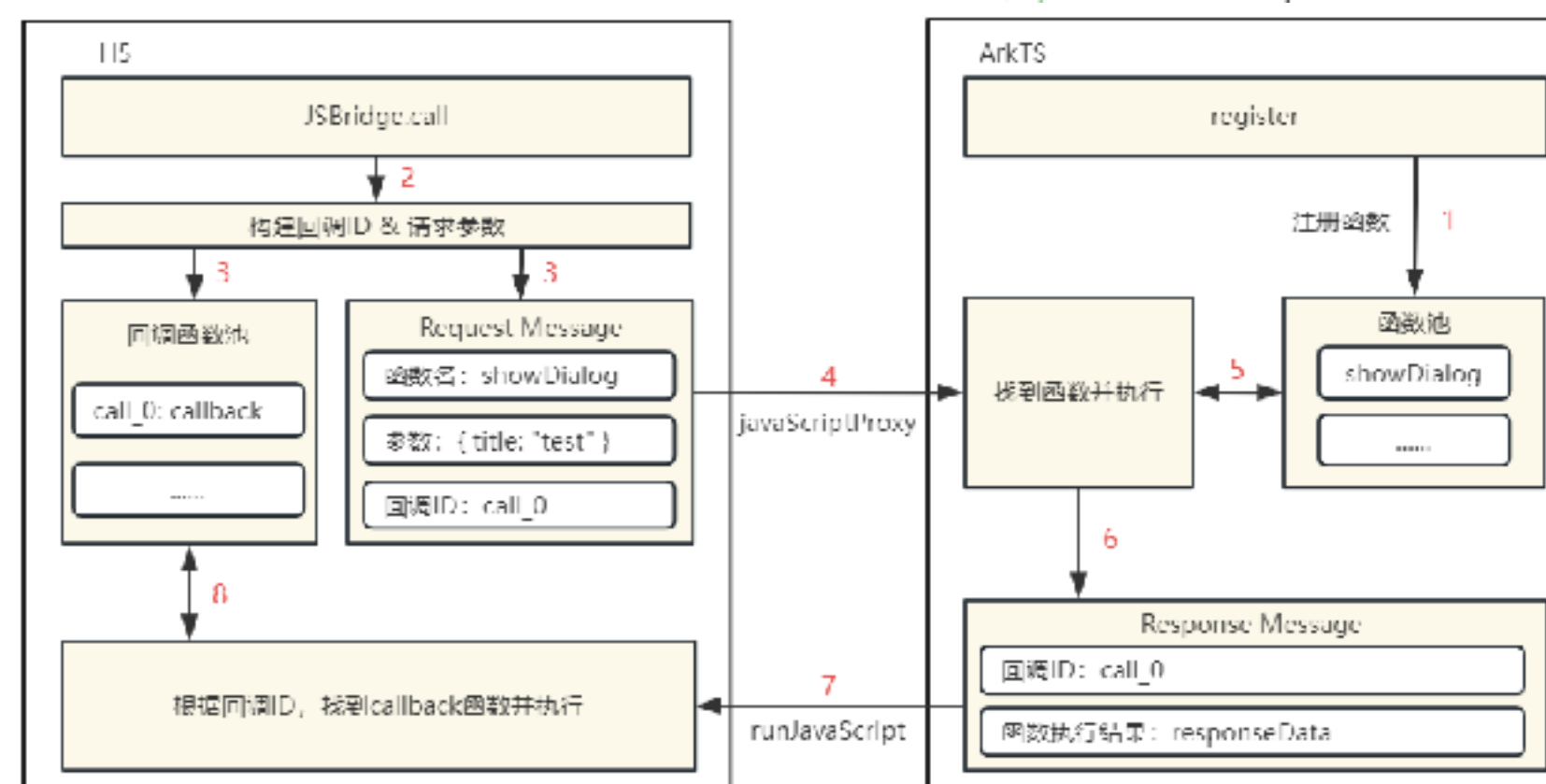
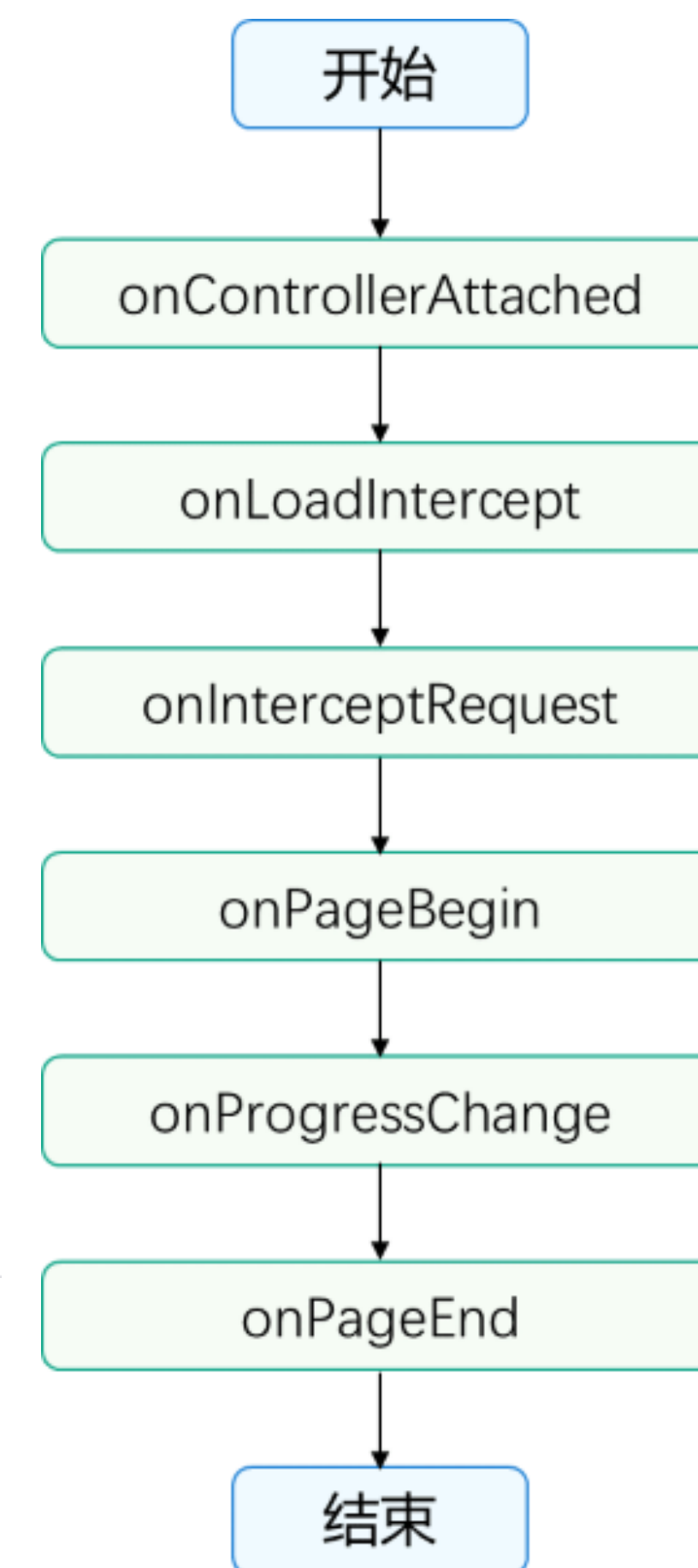
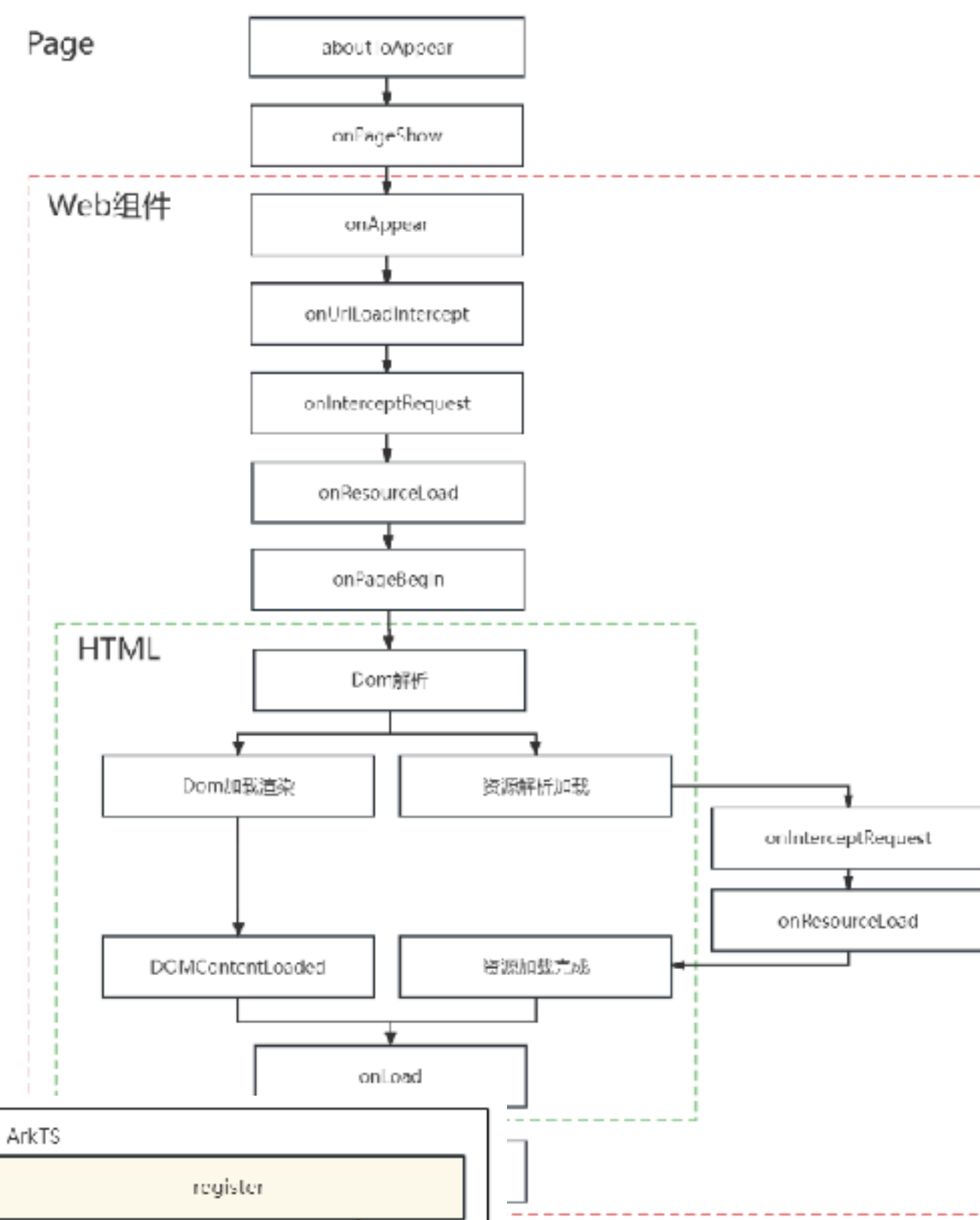
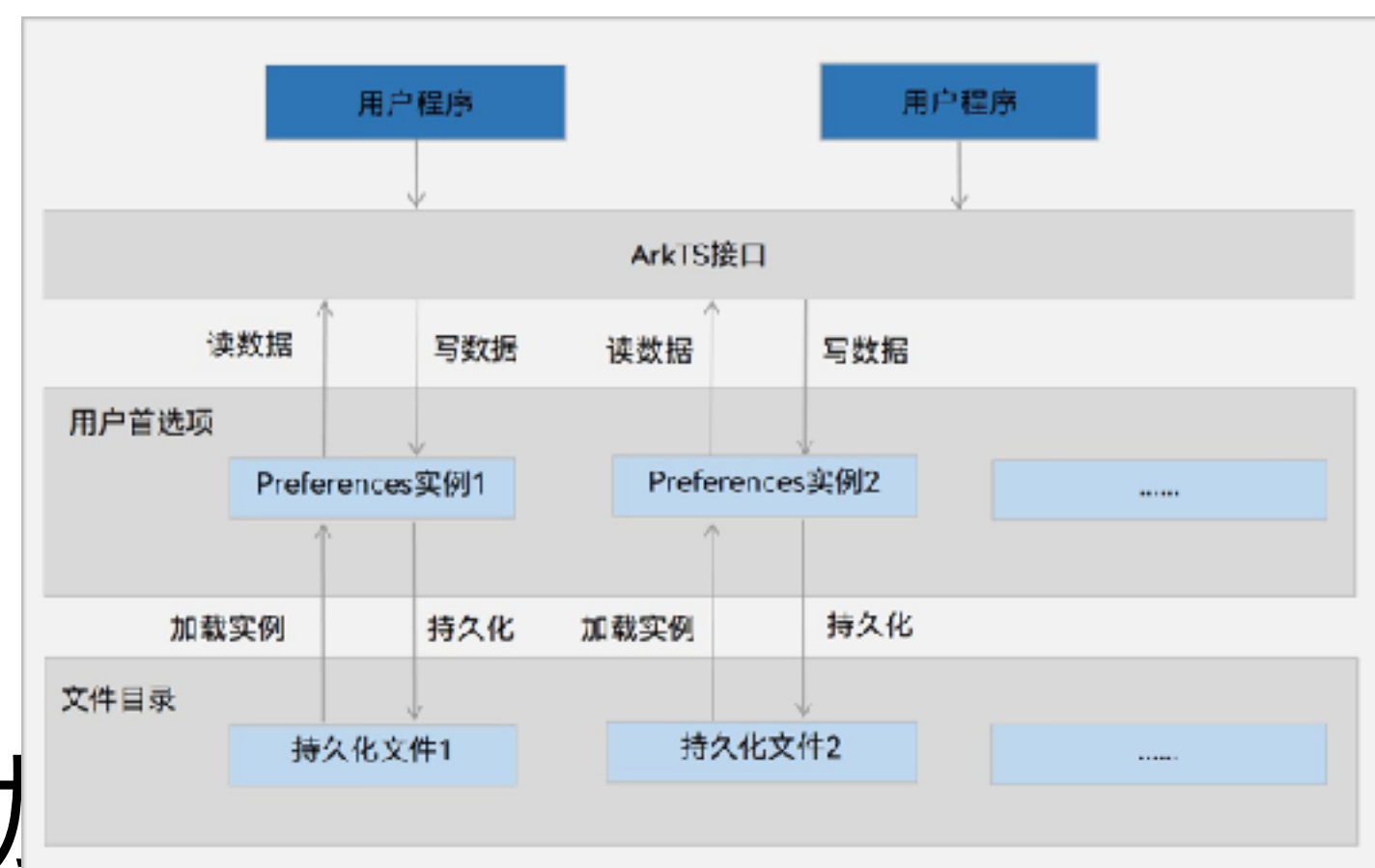
- 大前端框架演进
- React、Flutter、ArkUI对比

需求分析

- 层次性
- 涉众
- 需求获取
- 关注点
- 需求的组织

AbilityKit、Network Kit、ArkData和ArkWeb

- 启动UIAbility
- 冷启动和热启动
- 用户首选项原理
- Web组件生命周期
- H5页面和ArkTS交互



架构设计视角

- 业务、应用、数据、技术
- 整体横向分层抽象，局部纵向贯穿分解

7 总结

架构总结

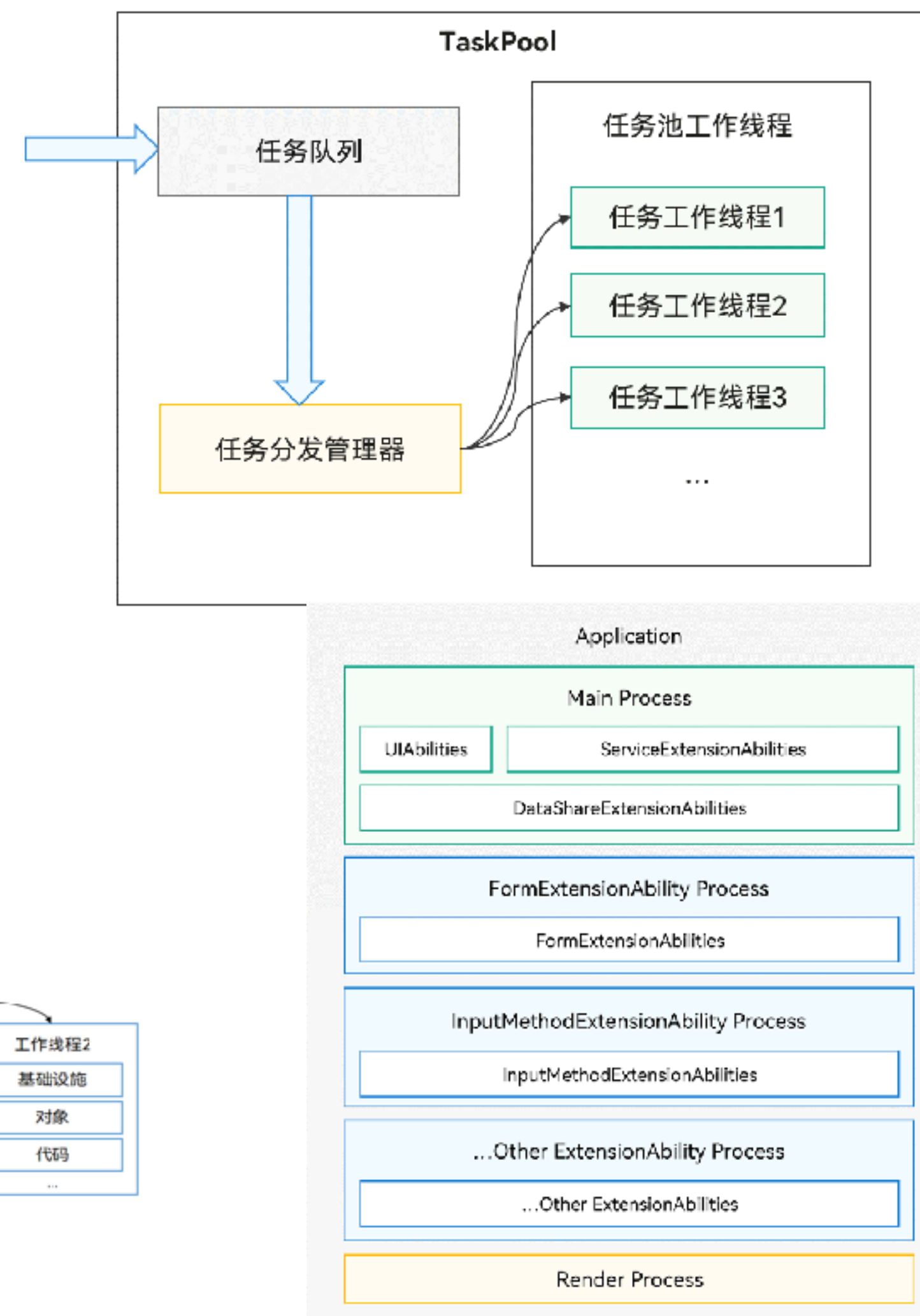
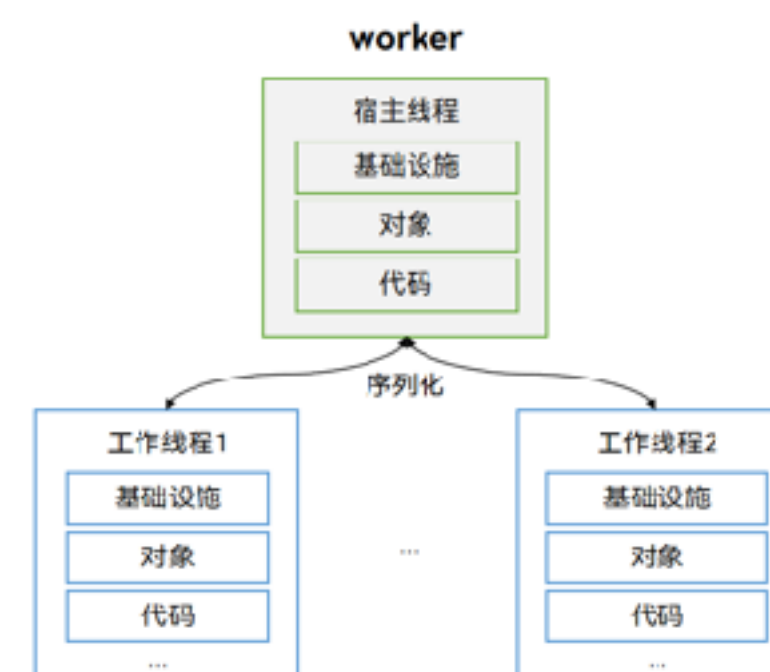
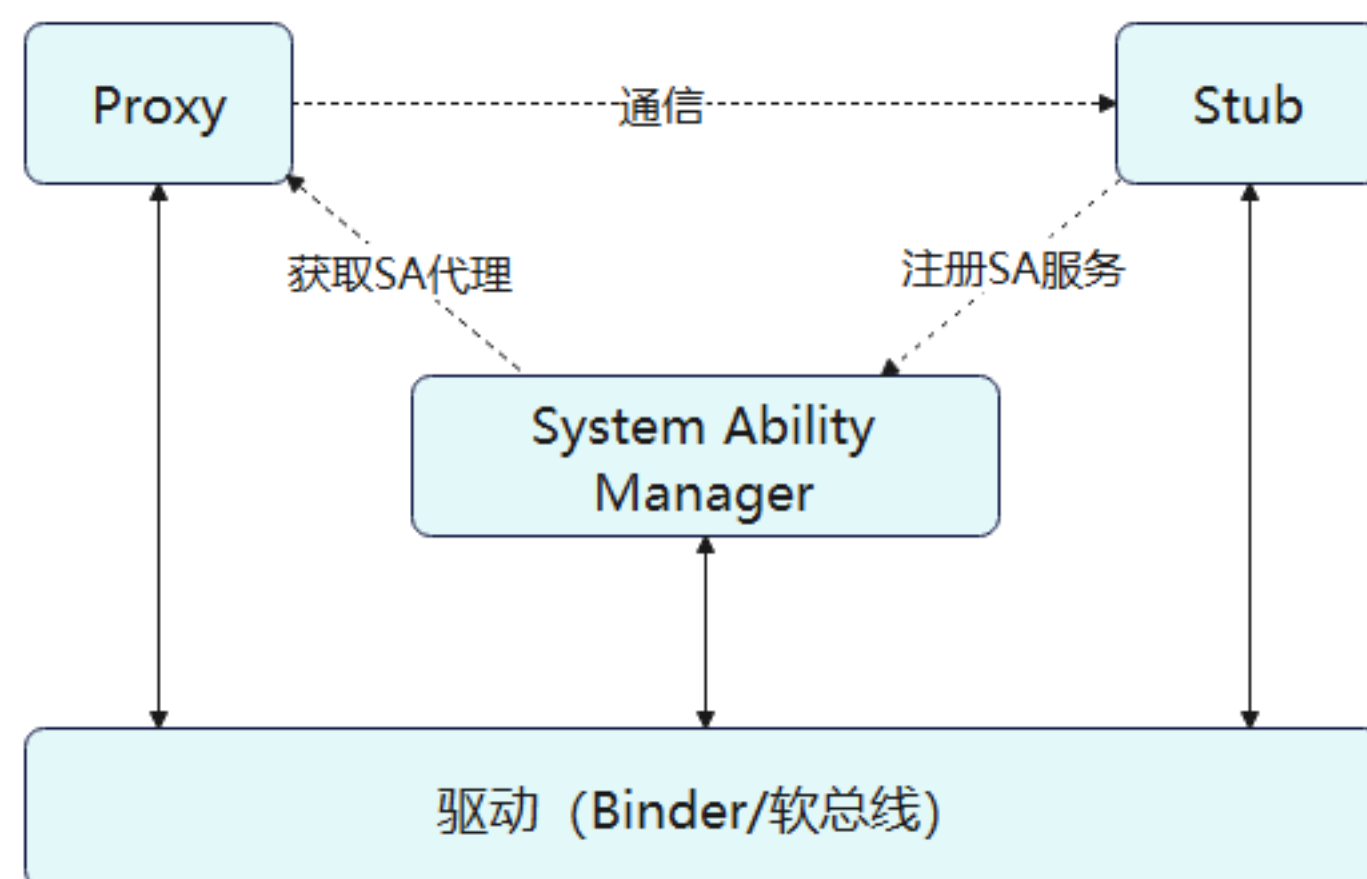
	解耦/拆分	抽象	集成	复用	治理
业务	1. 电商业务域 2. 核心、非核心业务 3. 主流程、辅流程 4. 业务规则分离		1. 跨业务域调用异步 2. 非核心业务异步	1. 基础业务下沉，可复用	1. 厘清业务边界、作用域
应用	1. 应用集群水平扩展 2. 按业务域分离应用 3. 按功能分离应用 4. 按稳定性分离应用	1. 服务抽象，服务调用不依赖实现细节 2. 应用集群抽象，应用位置透明	1. 易变依赖稳定 2. 流程服务依赖基础服务 3. 非核心应用依赖核心应用	1. 复用粒度是有业务逻辑的抽象服务	1. 服务自治 2. SLA 3. 可水平扩展 4. 可限流 5. 服务可降级 6. 容错设计 7. 服务白名单
数据	1. 读写分离 2. 按业务域分库 3. 分库分表 4. 冷热数据分离	1. 数据库抽象。应用只依赖逻辑数据库	1. 数据库只能通过服务访问 2. 统一的元数据管理 3. 统一的主数据管理		1. 重要数据做主备 2. 合理利用缓存容灾 3. 双写要做补偿
技术	1. 功能开发与运维分离 2. 业务子网 3. 分离功能、非功能型需求	1. 服务器资源抽象。应用只依赖虚拟化资源	1. 同步调用时，设置超时和任务队列长度 2. 利用回调异步化 3. 利用MQ、缓存、中间件异步化	1. 代码提共通，可复用 2. 非功能性服务，可复用 3. 基础配置、基础软件复用	1. N+1设计 2. 灰度部署 3. 版本可回滚 4. 可监控 5. 可容灾

高可用 高并发

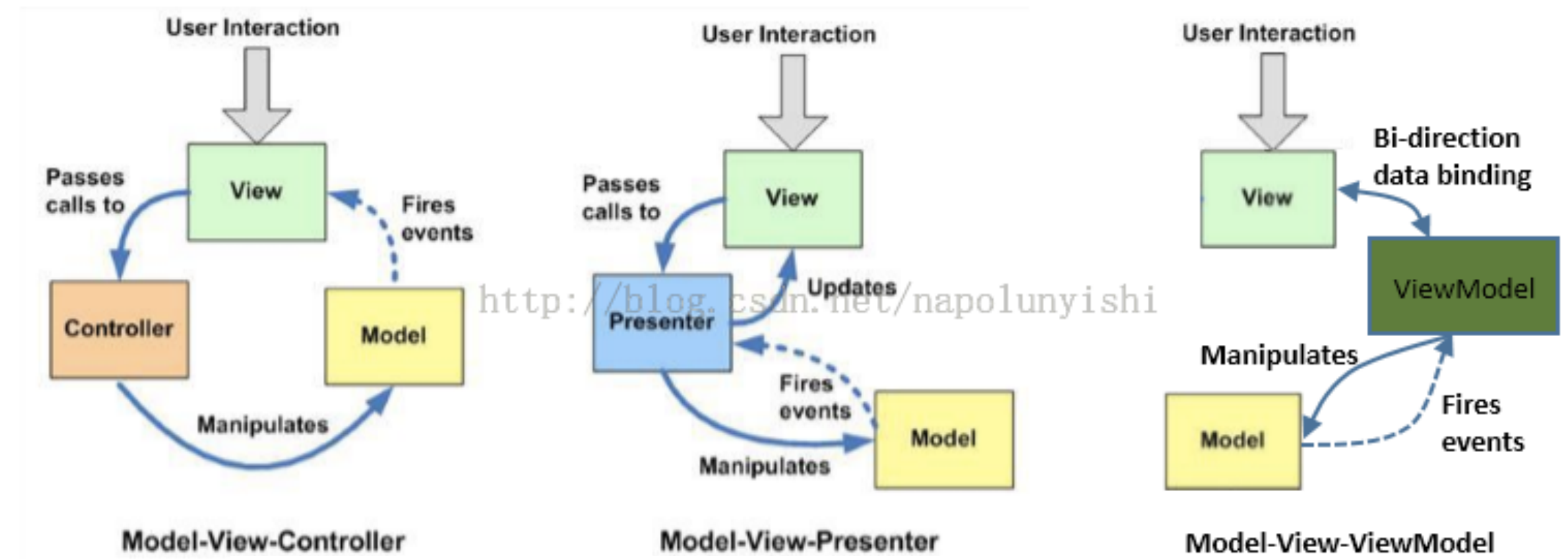
- 大型网站架构演化历程
- 高可用
 - 负载均衡与反向代理
 - 隔离
 - 限流
 - 降级
 - 超时与重试机制
 - 回滚
- 压测和预案
- 高并发
 - 应用级缓存
 - HTTP缓存
 - 连接池
 - 异步并发
 - 扩容
 - 队列

进程与线程

- 异步并发 (Promise和async/await)
- TaskPool机制和Worker机制
- 进程模型
- IPC机制

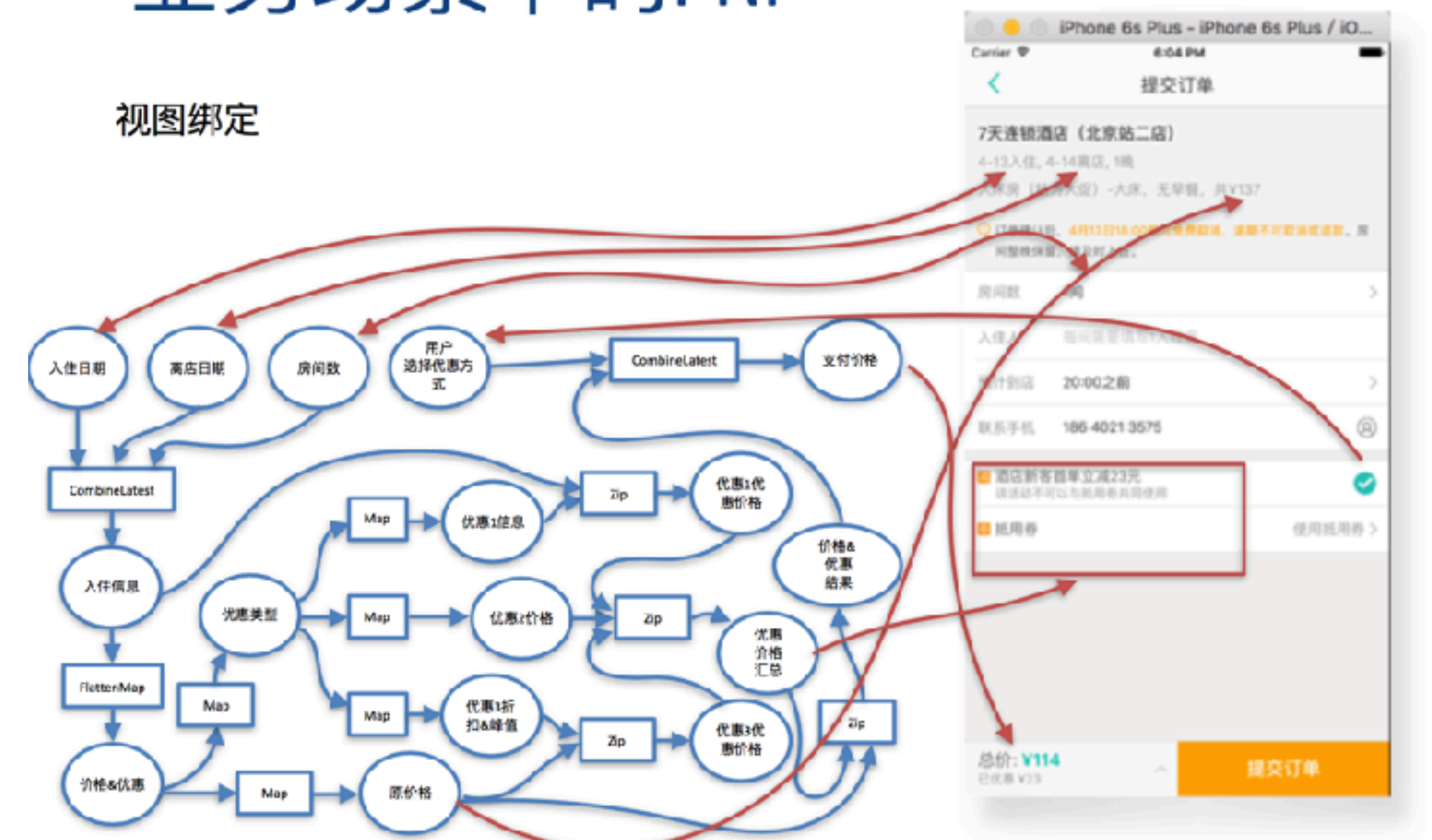


客户端架构



- MVC vs MVP vs MVVM
- FRP+MVVM

业务场景下的FRP

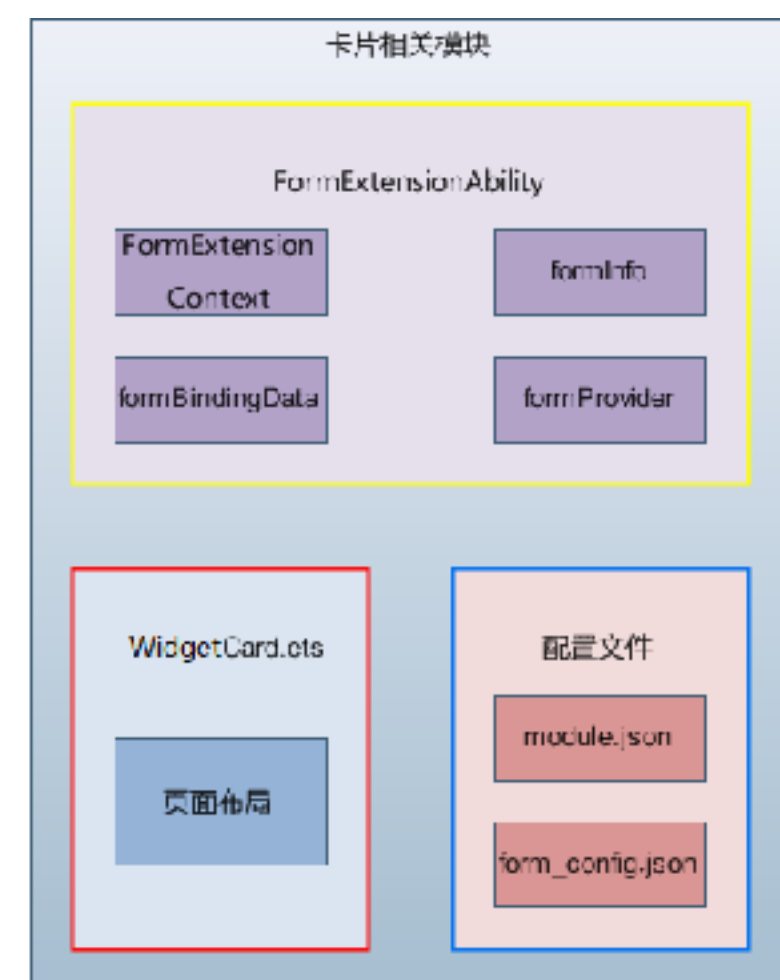
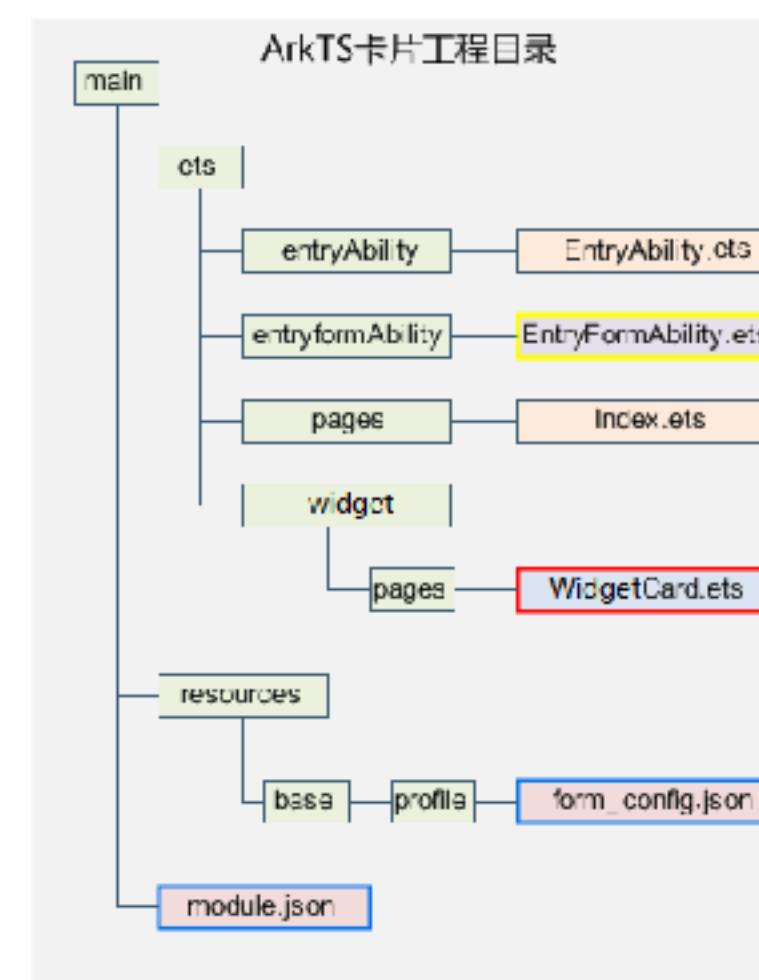
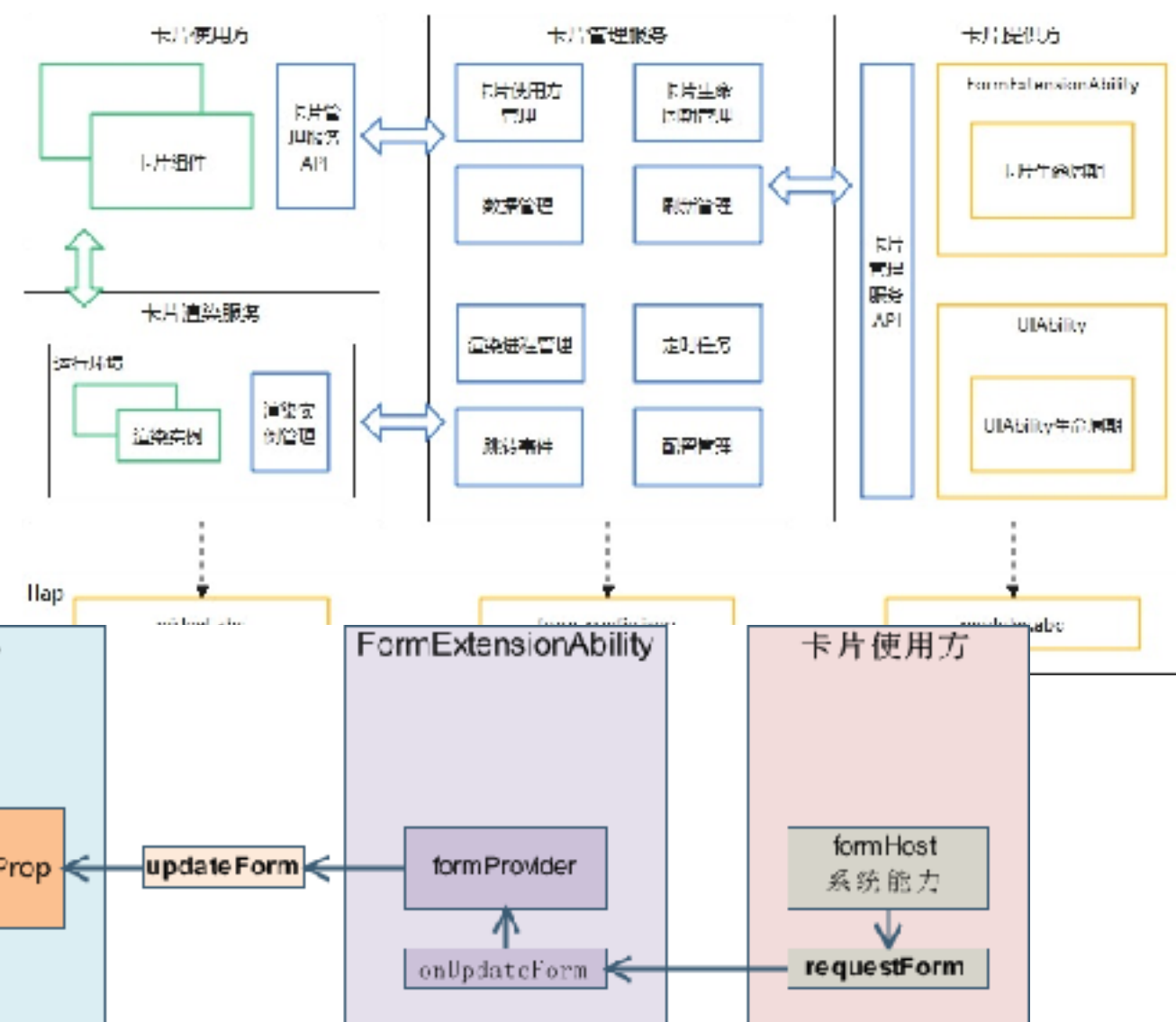
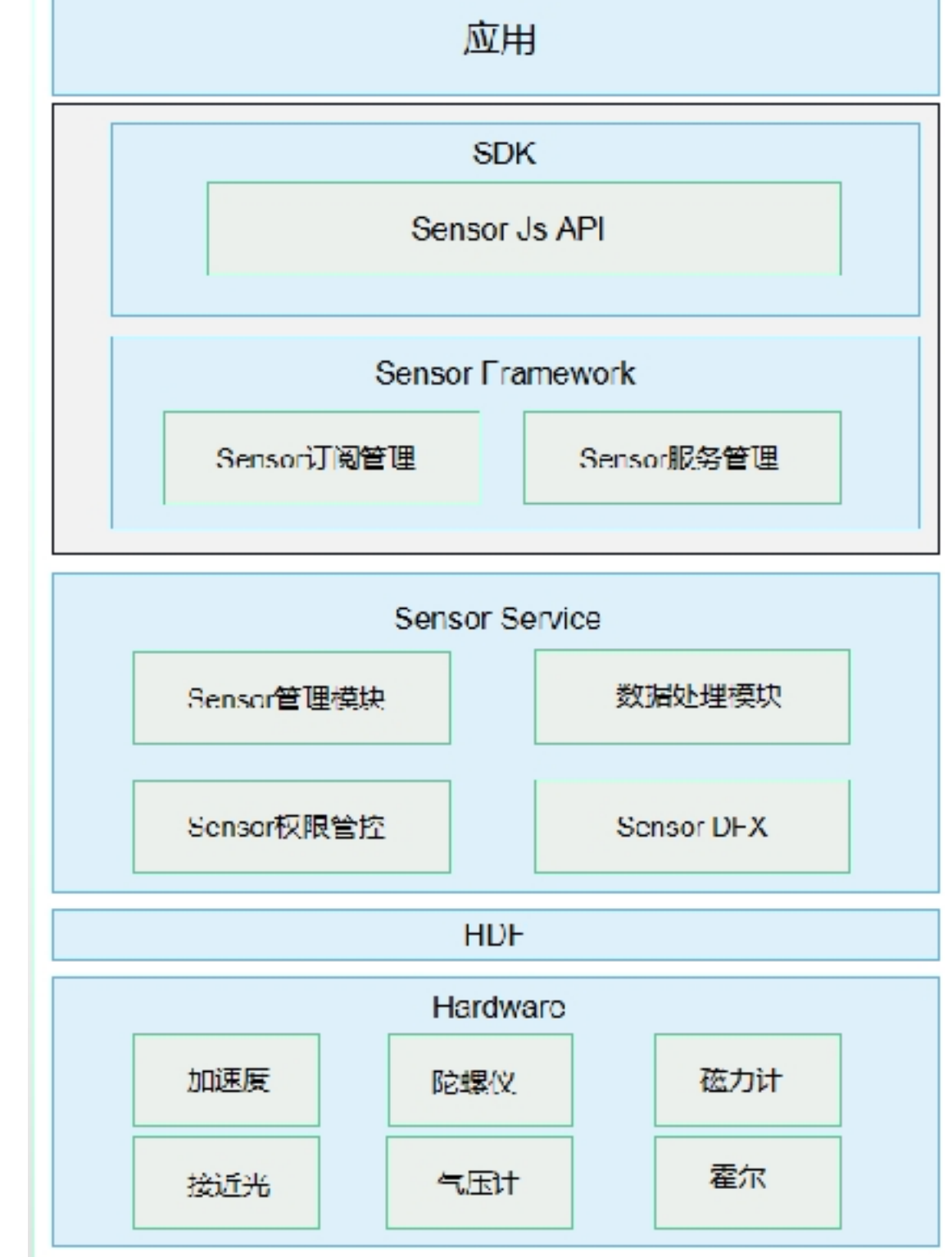
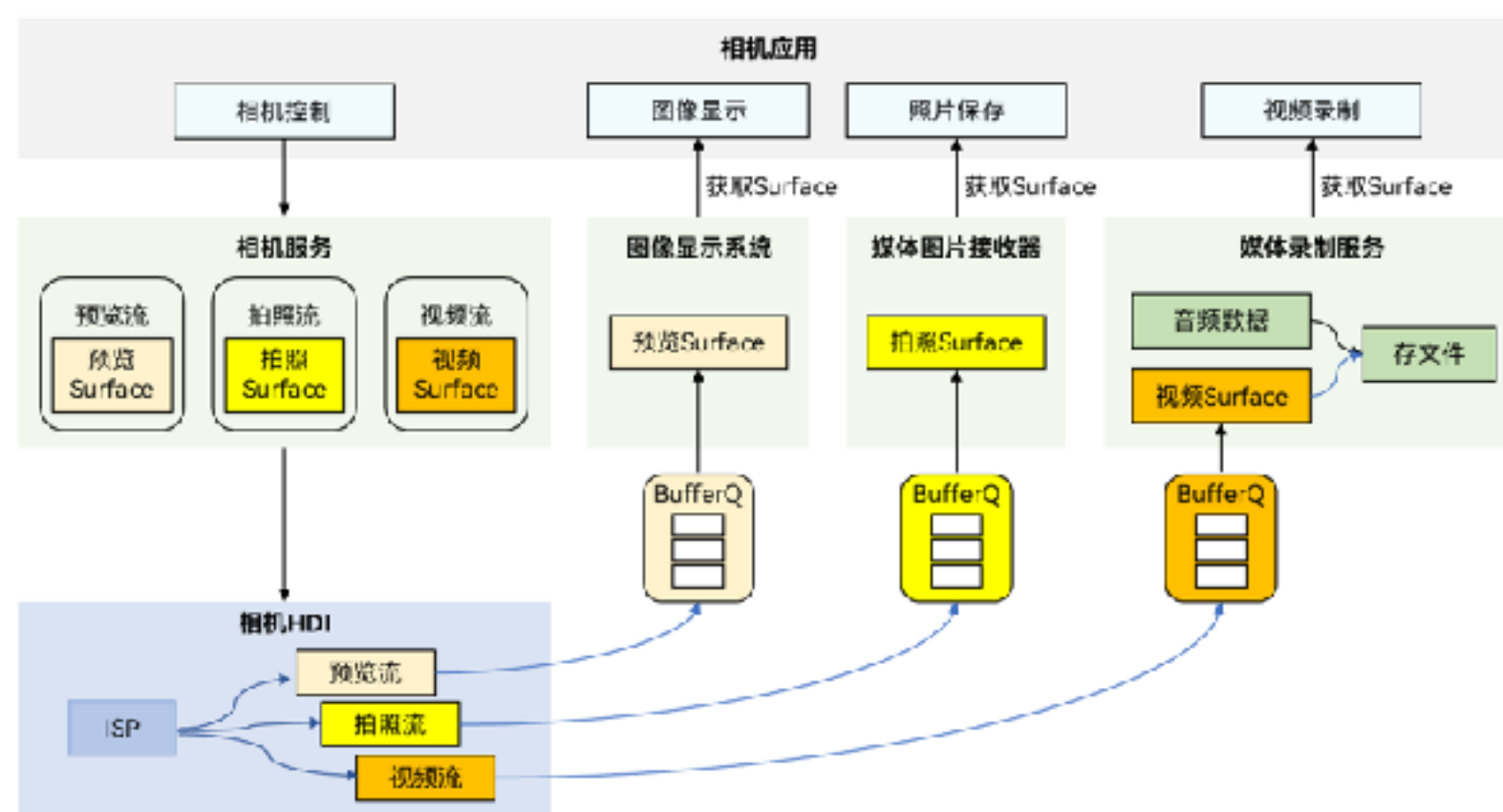


关注点

- 秒杀系统
- 可视化全链路日志追踪

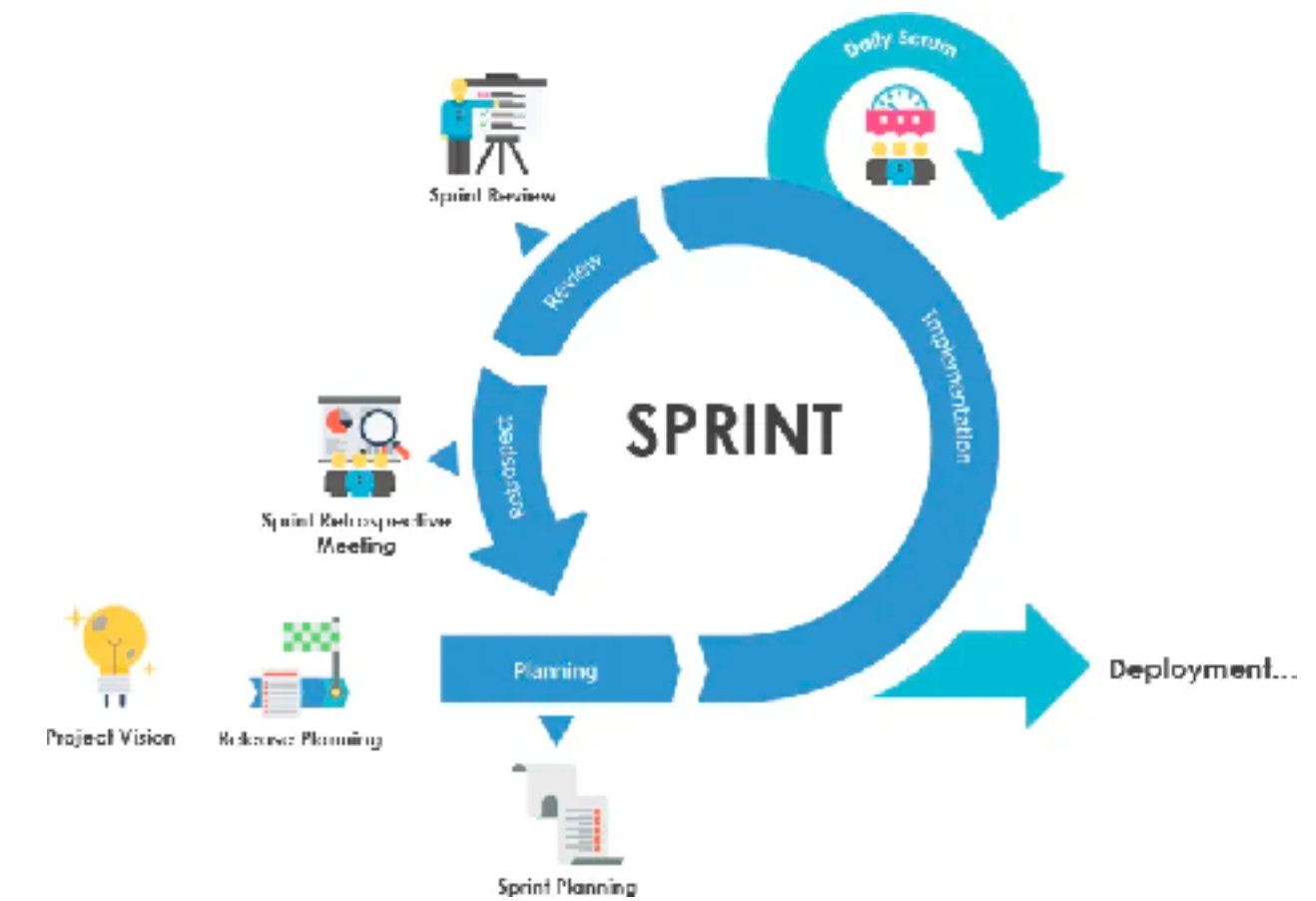
系统工具库

- 相机开发服务
- 传感器运作机制
- 卡片实现原理
- 卡片渲染原理
- 卡片数据交换



敏捷软件开发

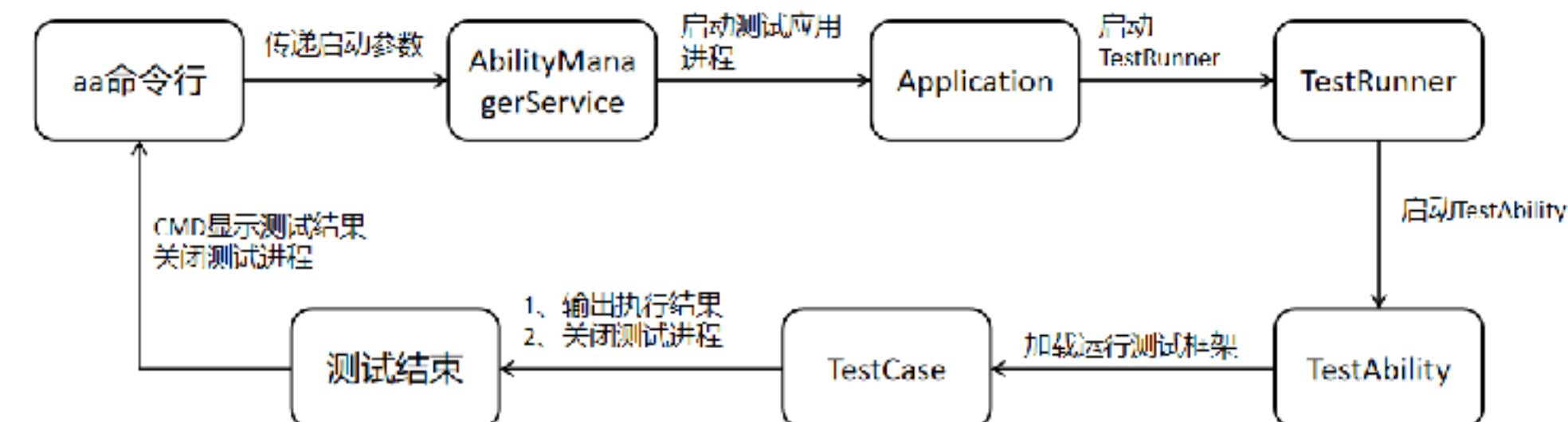
- Scrum
- 看板中的WIP
 - WIP太高，工作闲置
 - WIP太低，人员闲置



从微服务到云原生

- 微服务架构
- Docker、K8s、Istio架构
- Serverless
- 云原生

高级特性



- 脚本基础流程运行图
- UI测试用例
- NAPI对比JNI

