

理论上，Flutter 拥有原生 App 的性能。在 Flutter 新赛道上，如何构建新的高可用体系是一个新的命题。本章将拆分为四个方面进行详细介绍，包括高可用标准、性能优化最佳实践、稳定性保障最佳实践和可持续发展的高可用体系。

4.1 Flutter 高可用标准

构建高可用体系的第一步是先制定标准，有了标准才能公平地衡量 App 的表现。为了更好地衡量检测和优化 App 性能和可用性，将高可用指标分为以下两类：

- 性能体验视角：这类指标衡量用户能否流畅地操作 App，能否快速地看到页面内容，如 App 启动速度、首屏显示时间、流畅度、CPU 使用率和请求接口响应速度等。
- 稳定性视角：这类指标衡量用户操作被异常中止的概率，如错误异常率、内存使用率和请求接口成功率等。

以下重点从首屏显示时间、流畅度、CPU 使用率、错误异常率和内存使用率几个角度进行分析。

4.1.1 首屏显示时间

首屏显示时间是指从用户在 A 页面点击等操作触发页面打开，到页面内容稳定显示的时间，如图 4-3 所示。在 Flutter 混合工程中，起点页面可能是 Flutter 页面，也可能是原生页面；中间可能会唤起新的 Flutter 容器，也可能直接通过 Dart 侧路由 SDK 直接唤起新页面的 Widget；新页面的内容可能是一个网络请求，也可能是多个网络请求，但首屏显示时间可以只关注起点和终点。

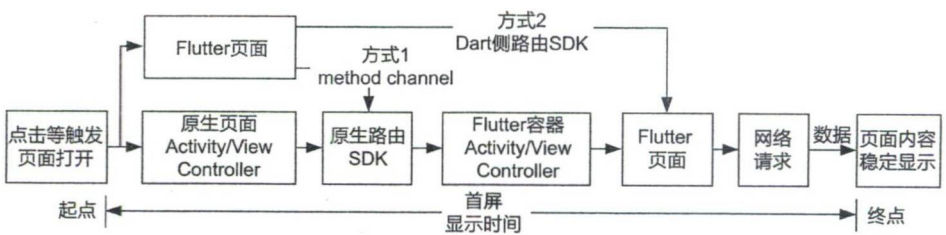


图 4-3 页面显示时间示意图

1. 起点定义

为避免业务层触发点的不确定性，定义路由接口调用入口为起始跳转时间点。在混合工程下，需要同时监听 Flutter 路由以及 Native 路由，以获取更合理的起始点。