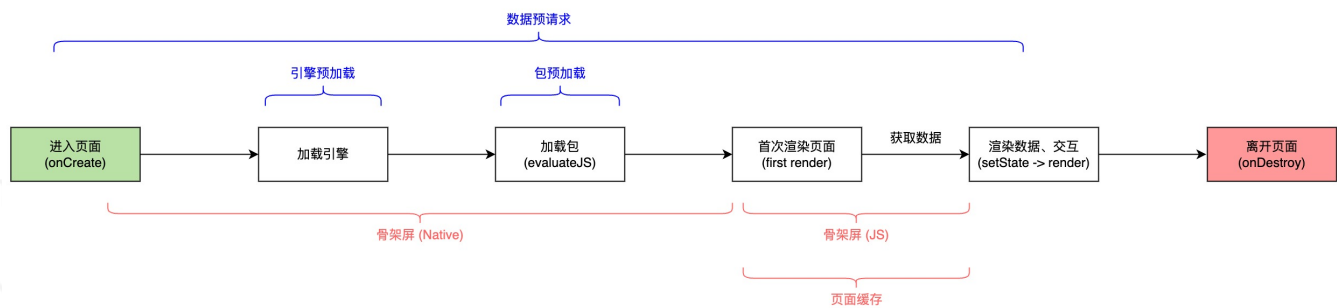


2.RN 加载体验提升

加载全流程



各阶段说明

分类	项目 / 方案	描述	是否已实现	数据提升预估	补充说明
降低页面加载时间	引擎预加载	预先初始化引擎，进入页面时直接使用已预热的引擎，降低加载耗时	是 框架内部实现		
	包预加载	预先把包读取到内存并执行 evaluate，进入页面的时候跳过此步，降低加载耗时	否 可业务实现	20ms 左右	这里主要是 IO 成本，在使用 V8 的时候，这一步会做「编译」的动作，把源代码编译成中间码，会比较耗时。目前使用 Hermes，bundle 包已经是中间码。
	数据预请求方案	进入页面的同时先请求当前页面的数据，等执行到页面 render 需要请求数据的时候直接获取本地已经请求成功的数据即可	否 可业务实现	<ul style="list-style-type: none">200ms 左右（前公司）线下测试数据：线下网络请求时间统计	这个数据是以前公司的数据统计，针对当前我们的，需要自己再统计下，使用 1.RN 监控 的「页面加载耗时」来评估
提高加载过程体验	骨架屏	通过骨架屏取代 Loading，优化加载过程中的体验	否 <ul style="list-style-type: none">骨架屏 (Native): 框架内部实现骨架屏 (JS): 可业务实现		
	页面快照缓存	记录上一次页面的快照（页面级缓存），下次首次先用快照显示，数据更新后再替换掉	否 框架内部实现		RN 技术栈上可以记录首次用数据渲染的各种 Action（如 CreateNode、CreateText 等），下次直接先用这些 Action 创建快照页面