

年度直播：

1/11

前端转管理

2/29

前端搞基建

3/28

前端搞搭建

4/11

前端搞规划

4/25

前端搞监控

5/16

联票群彩蛋

5/30

前端搞构建

6/27

前端搞跨端

7/25

前端搞部署

8/29

前端搞可视化

9/26

前端搞性能

10/31

前端搞安全

11/28

前端搞框架

12/26

前端搞报表

第三届前端早早聊大会

前端成长的新起点 | 前端如何搞搭建-3/28

| | | | | |
|----|---------------------|-----------------------|------|-------|
| 洛尘 | 鲁班/搭建页面 500+ | 如何设计实现 PC 站点搭建 Schema | 政采云 | 10:00 |
| 月飞 | 飞冰 iceluna/搭建页面 6k+ | 如何设计实现中后台搭建 PaaS 服务 | 淘宝 | 11:00 |
| 沐童 | MPM/搭建页面 1.9w+ | 如何设计实现 H5 页面搭建 数据模型 | 京东 | 13:00 |
| 妙净 | ImgCook/搭建页面 2w+ | 如何为搭建物料智能生成代码 自动编码 | 淘宝 | 14:00 |
| 墨冥 | 方舟/搭建页面 2w+ | 如何设计实现 toC 营销搭建 终端秒开 | 淘宝 | 15:00 |
| 步天 | 天马/搭建页面 100w+ | 如何设计实现跨端页面搭建 跨端模块 | 淘宝 | 16:00 |
| 奕纯 | 淘积木/搭建页面 200w+ | 如何设计实现 toB 商家搭建 海量部署 | 阿里妈妈 | 17:00 |

跟进大会 PPT/录播视频/讲稿文章
跟进下届大会新动态/讲师征集
跟进大会优秀团队/招聘岗位
大会报名跟进/扫码关注



掘金

如何设计实现 H5 页面搭建 数据模型

京东京喜前端团队 沐童



CATALOG

目录

- 01 MPM 整体介绍**
- 02 数据层面临的痛点**
- 03 页面模型设计**
- 04 请求模型设计**
- 05 总结**

01

MPM 整体介绍

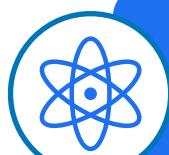




上线服务4年+, 系统迭代超3个版本



累计使用人数1400+, 搭建页面1.9w张



除日常活动外, 承担80+%的大促会场

Mart Page Maker

H5 卖场可视化搭建系统

| MPM能力概览

搭建物料丰富

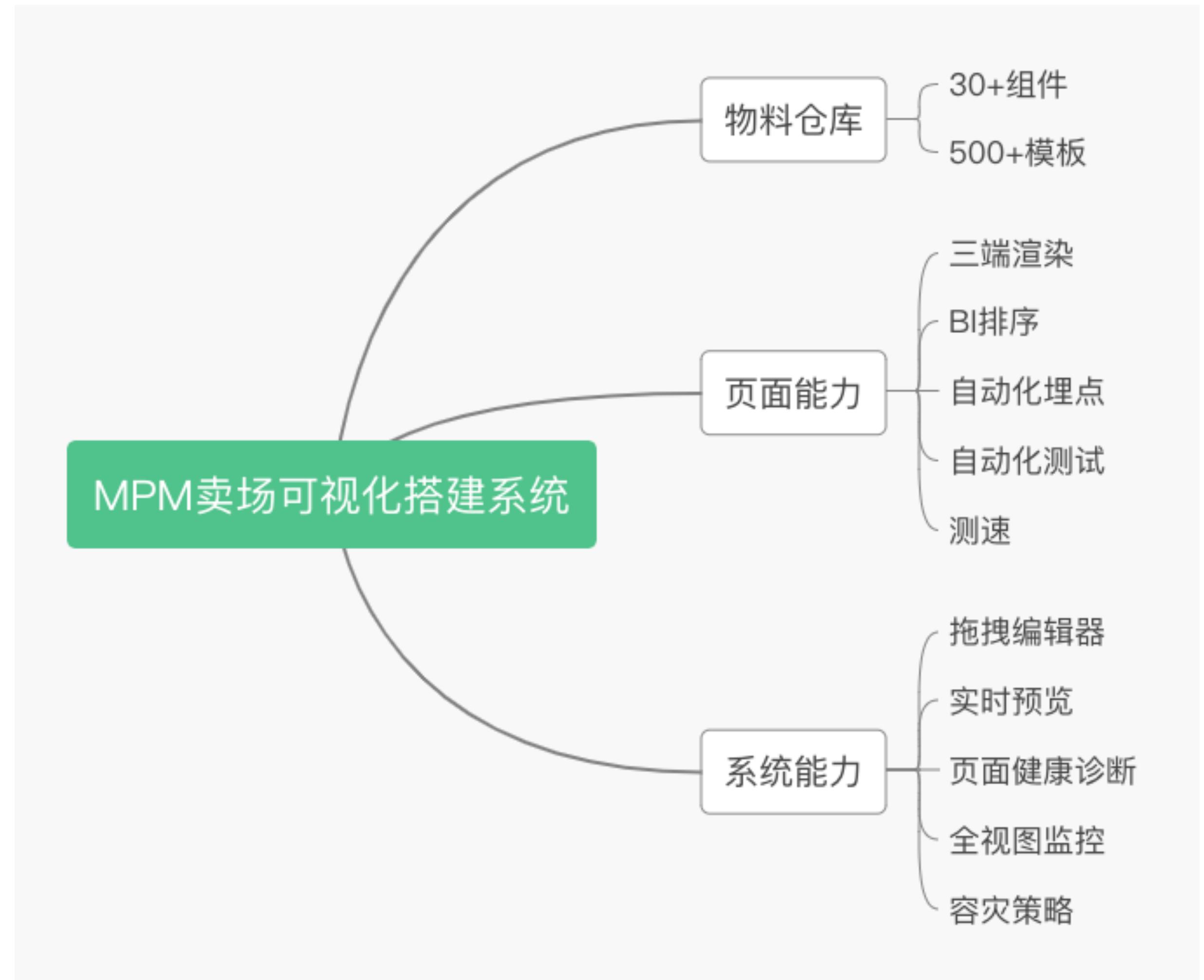
MPM 现有 30+ 个组件、500+ 个模板，业务能力覆盖商品、导购、营销等多个场景。

配置功能强大

三端渲染是 MPM 的强大能力，除此之外，MPM 还支持页面配置 BI 排序、自动化埋点、自动化测试、页面测速等。

系统能力全面

MPM 为页面保驾护航，不但配备了流畅的拖拽编辑器、实时预览和页面健康诊断能力，还对系统和页面做了全方面的监控和容灾降级方案。



MPM效果展示

1、从这里拖一个组件到预览区

MPM卖场页面制作系统

基本信息 其他信息 保存 预览 发布 更多

商品组件

- 猜你喜欢
- SKU商品

图文组件

- 秒杀
- 商品排行榜

导购内容

- 购物车降价
- 拼购商品

公共导航

- 智能卖场
- 商品组

营销活动

- 新品首发
- 品牌特卖

- 为您推荐
- 拼购榜单

- 智能榜单

选择样式

全部 1 普通样式 618样式

样式key:2204
1) 一行一个样式，拉取单买价和补贴价作差价展示；
2) 可配置展示「百亿补贴」角标；
3) 拉取「上角标」和「促销运营语」作为利益点。

样式key:2206
1) 一行两个样式，拉取单买价和补贴价作差价展示；
2) 可配置展示「百亿补贴」角标；
3) 拉取「上角标」和「促销运营语」作为利益点。

样式key:2210
1) 一行三个，展示拼购价、线价和返现信息

样式key:1732
1、功能：支持区分新老用户展示

样式key:1734
1、支持去参团
2、支持配券：拉取中台【拼...

样式key:2224
1) 返现商品展示「拼购价-返现价=返后价」，非返现商品...

样式key:1502
1、配置：【banner】 【标...

样式key:2044
1、配置：【背景图 710 x 1000px】 【今日必拼组件】 @分场次10

样式key:2124
1、配置：【背景图 710 x 1000px】 【今日必拼组件】 @分场次10

2、选择你需要的模板

MPM效果展示

The screenshot displays the MPM Marketing Platform's interface for creating promotional banners. On the left, a sidebar lists various components and sections. The main area shows a preview of a red-themed promotional banner titled '全民京喜日' (Everyone's Joy Day) featuring a '海底捞火锅底料3包13.9元' (Haidilao Hot Pot Base 3 bags 13.9 yuan) offer. Below this, sections include '签到抽惊喜' (Check-in to draw surprises), '新人专享' (Exclusive for new users), and '补贴爆款' (Subsidy best-sellers). The right side shows the configuration panel for a specific component, with tabs for '选择样式' (Select Style), '设置基本数据' (Set Basic Data), and '设置样式属性' (Set Style Properties). The '设置基本数据' tab is active, showing fields for '展示开始时间' (Display Start Time), '展示结束时间' (Display End Time), '可见用户' (Visible Users) with options for '全部可见' (All visible), '仅新用户可见' (Only new users visible), and '仅老用户可见' (Only old users visible); '展示环境' (Display Environment) with checked boxes for '微信H5' (WeChat H5), '小程序' (Mini Program), '手Q' (Hand Q), 'M站' (M Station), '京东APP' (JD APP), and '京喜APP' (Jingxi APP); '配置方式' (Configuration Method) with a selected radio button for '营销中台' (Marketing Platform); and input fields for '商品组id' (Product Group ID) set to '03603206', '商品数量' (Product Quantity) set to '30', and '默认展示数量' (Default Display Quantity). A large blue speech bubble in the bottom right corner contains the text '4、发布页面' (4. Publish Page).

MPM卖场页面制作系统

基本信息 其他信息 保存 预览 发布 更多

商品组件

猜你喜欢 SKU

秒杀 商品排行榜

购物车降价 拼购商品

智能卖场 商品组

新品首发 品牌特卖

为您推荐 拼购榜单

智能榜单

12.01 12.02-12.03 12.04
京喜开门红 全民拼购日 超市百货不过百 潮

全民京喜日 海底捞火锅底料3包13.9元

超市百货 服饰内衣 运动鞋包 母婴美妆

签到抽惊喜 12月10日-12月12日还有机会抽大红包哦！
活动未开始

12.1 12.2 12.3 12.4 12.5 12.6 12.7

新人专享 1元好货 仅新用户才有的福利

补贴爆款 今日可购买3件！羊毛薅起来！

超级百亿补贴 新奇士Sunkist澳洲一级脐橙
新品上市 高品质

开启sku置顶 是 否

拼购商品 com_1004

1. 功能描述：多样式的拼购商品展示
2. 数据源：营销中台商品组/icos智能卖场/秒杀
3. 配置说明：
4. 注意事项：
1) 使用营销中台收品时，拼购业务有2种模式，通天塔模板（分期模式）&非通天塔模板（标签模式），一般建议选择通天塔模板（分期模式），避免后续出现商品过期导致的空窗问题。
·分期模式特点：商品有效期与分期一致，若分期不过期，商品一直有效
·标签模式特点：校验商品有效期，若商品过期则前端展示空窗

选择样式 设置基本数据 设置样式属性

展示开始时间 选择日期时间

展示结束时间 选择日期时间

可见用户 全部可见 仅新用户可见 仅老用户可见

展示环境 微信H5 小程序 手Q M站 京东APP 京喜APP

配置方式 营销中台 智能卖场 秒杀 联合营销

商品组id 03603206 同步商品组数据

商品数量 30

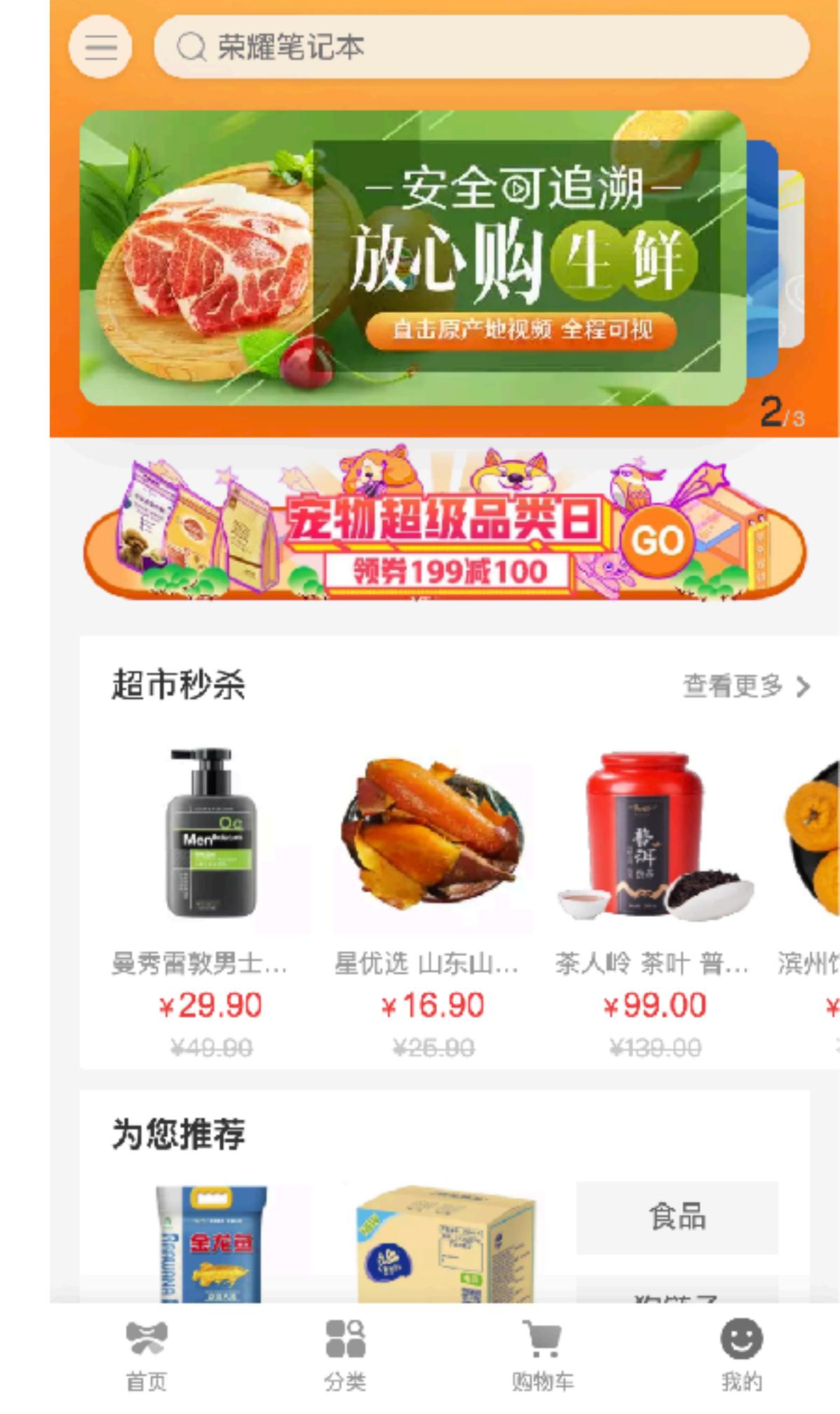
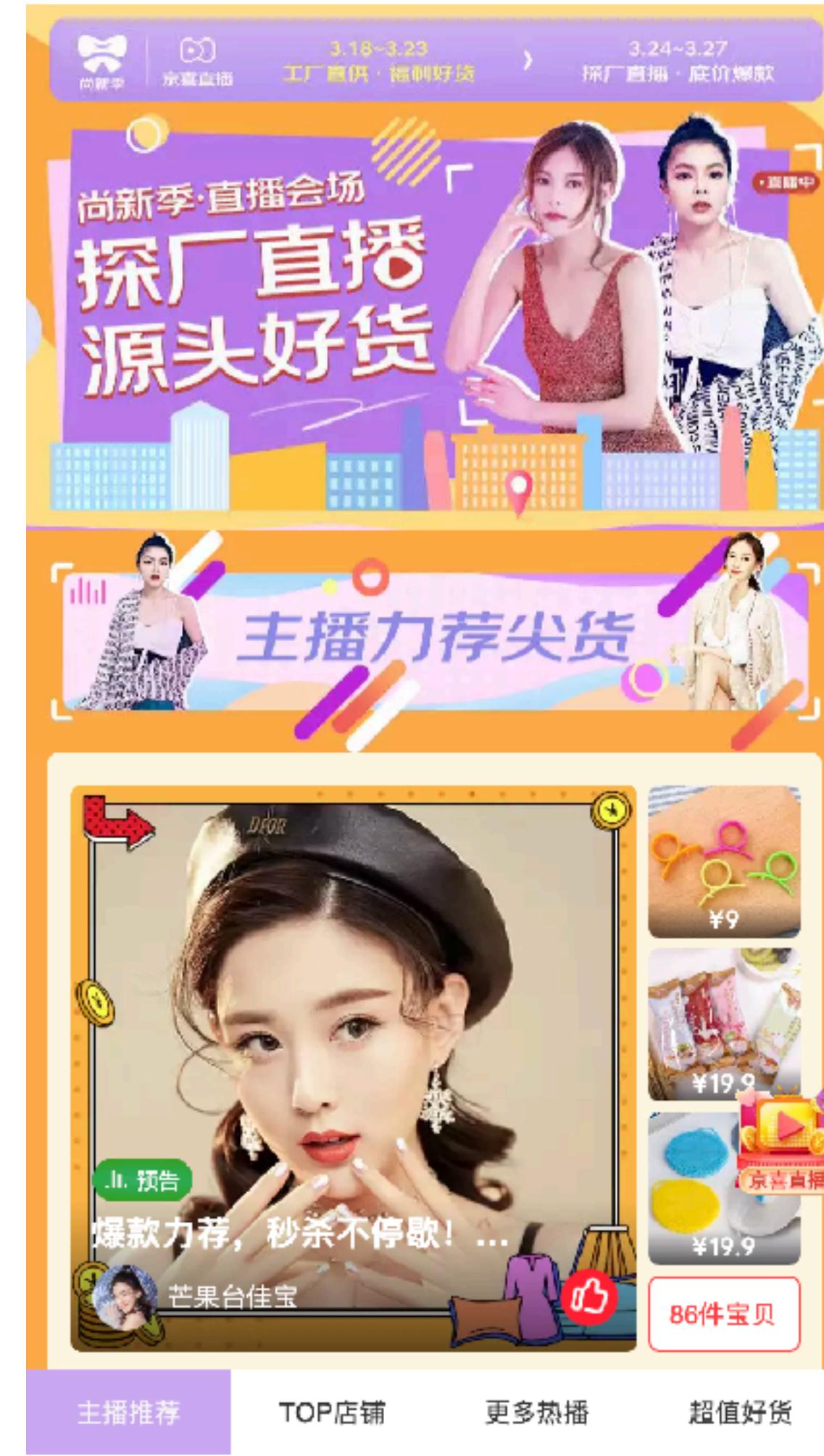
默认展示数量

开启sku置顶 是 否

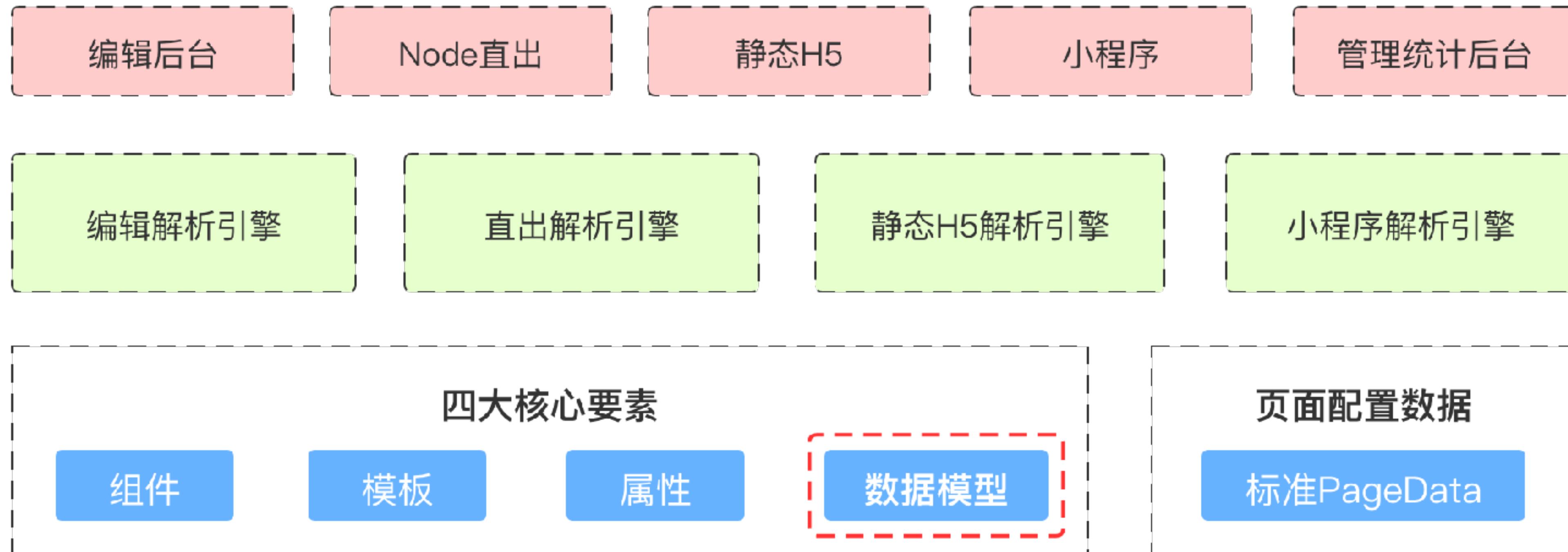
3、配置楼层组件 属性

4、发布页面

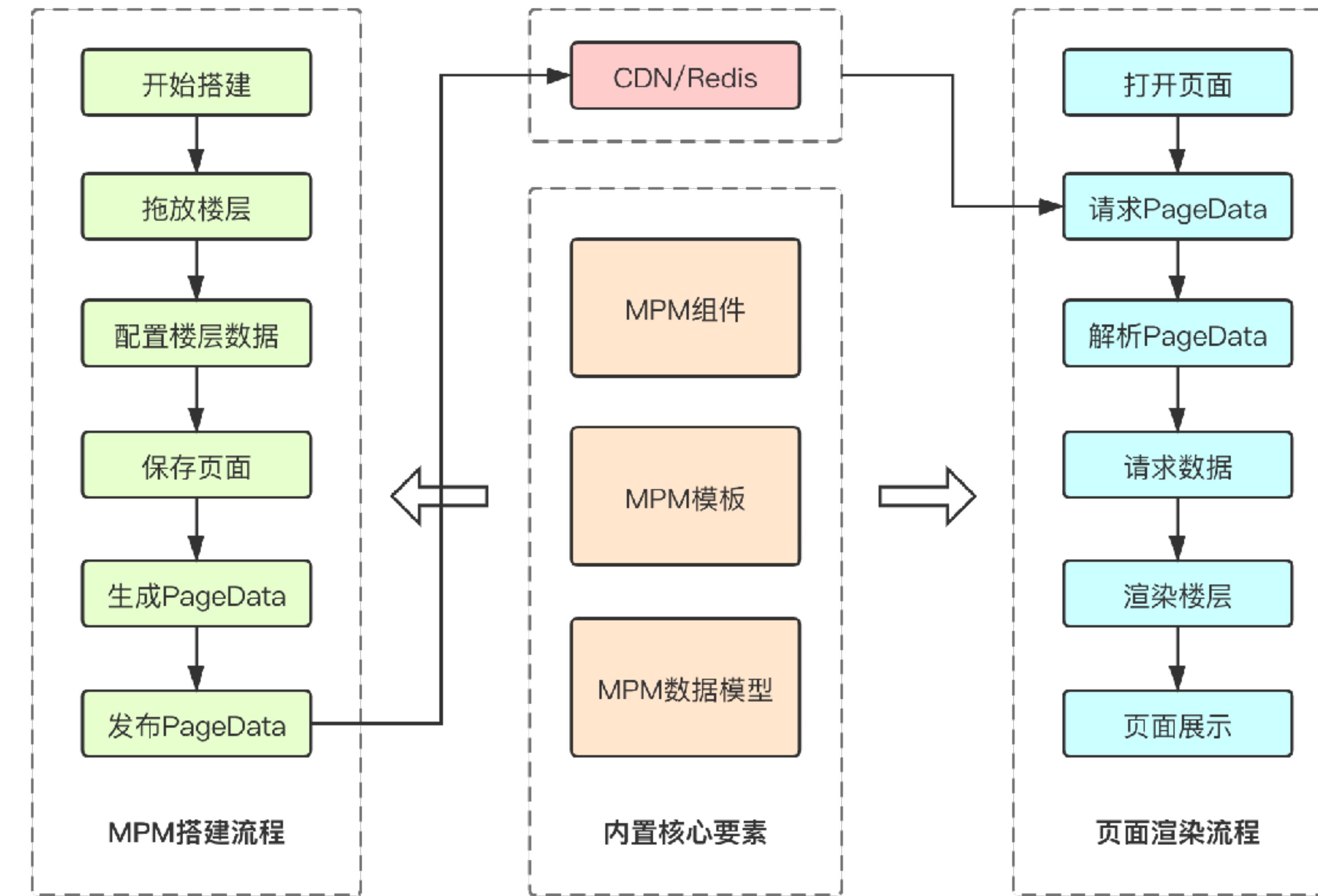
MPM效果展示



| MPM系统架构



MPM工作流程



02

数据层面临的痛点

| 解决场景

“

页面请求茫茫多，有时候想定位页面中某个请求来自哪个组件，可能得定位半天……

”

请求散乱无章

卖场承载各线业务，接口场景繁杂，如果简单地将请求交给组件自身各自发起和处理，维护起来十分麻烦。

| 解决场景

“

某个页面，多个组件都配置了
同一个预约 ID，导致页面发出
了 N 个一模一样的预约状态查
询请求……

”

多余的重大请求

缺乏数据请求统一管理的后果就是，这些无
效的重复请求将严重拖垮你的页面性能。

| 解决场景

“

商品接口支持批量请求，可我们页面中多个商品组件的接口请求并没有合理聚合，走批量调用……

”

接口压力大

同个页面经常会多次请求同个业务接口，这对接口服务造成了十分大的压力。

| 解决场景

“

商品组件下的各个模板，除请求商品之外，有些模板会拉取新人价，有些模板会拉取补贴价……

”

数据模型多变

组件对应多个模板，每个模板都可能对应了不同的数据模型。

| 解决场景

“

以 Vue 为例，我们习惯在组件
created 时请求数据，这在客
户端渲染时十分完美，但在直
出场景却完全行不通.....

”

三端同构诉求

Vue 支持 SSR，但却欠缺对异步数据获取的
同构支持，为了适配 MPM 的三端同构，数
据层设计必须考虑这个问题。

| 定义

为自搭建卖场打造的一套高效通用的**数据请求解决方案**

1

统一管理

维护请求秩序，优化
请求性能

2

自由组合

数据模型即插即用，
自由组合

3

适配三端

为三端同构提供统一
的数据请求方案

03

页面模型设计



| PageData

PageData 是页面的抽象描述层

PageData —————^{解析引擎}—> 真实页面

► 页面级配置

- 1) 一些基础配置，页面级别的请求，如[楼层 BI 排序查询](#)在这里发起。
- 2) 页面对用户身份的要求配置，用户级请求，如[用户身份查询](#)在这里发起。

► 组件级配置

包含了组件楼层的配置，因此这份配置决定了[组件楼层的业务数据获取](#)。

```
interface iPageData {  
    pageId: number; // 页面id  
  
    pageConfig: {  
        basic: iBasic; // 基础信息配置  
        search: iSearch; // 搜索配置  
        share: iShare; // 分享配置  
    };  
  
    userConfig: { // 用户配置  
        checkNewUser: boolean; // 1) 是否需要查询新人  
    },  
  
    componentConfig: iComData[] // 组件配置  
}
```

ComData 是 PageData 的核心部分

► 模板配置

指定使用某个模板，引擎将根据这个配置使用相应的组件模板进行渲染。

► 数据配置

楼层配置数据，这里包含了组件请求接口所需的参数。

► 组件关系

描述了组件的父子级对应关系。

```
interface iComData {  
    uid: number; // 组件id  
    source: string; // 数据源标识  
    data: object; // 组件配置数据  
    styleKey: string; // 组件模板id  
    parentUid: number; // 父组件id  
    children: iComData[]; // 子组件  
}
```

04

请求模型设计



•
•
•

数据源及请求优化策略

- 统一管理 + 自由组合 -



| 数据源

数据源是请求模型的基本单位，描述了一类请求动作

接口地址

每个数据源对应了一个接口

1

请求后置处理

请求响应后的一些通用的数据处理逻辑

3

聚合分发策略

描述了如何对多个同接口请求作请求聚合和响应分发

5

请求前置处理

发起请求前的参数组装处理

2

入参校验

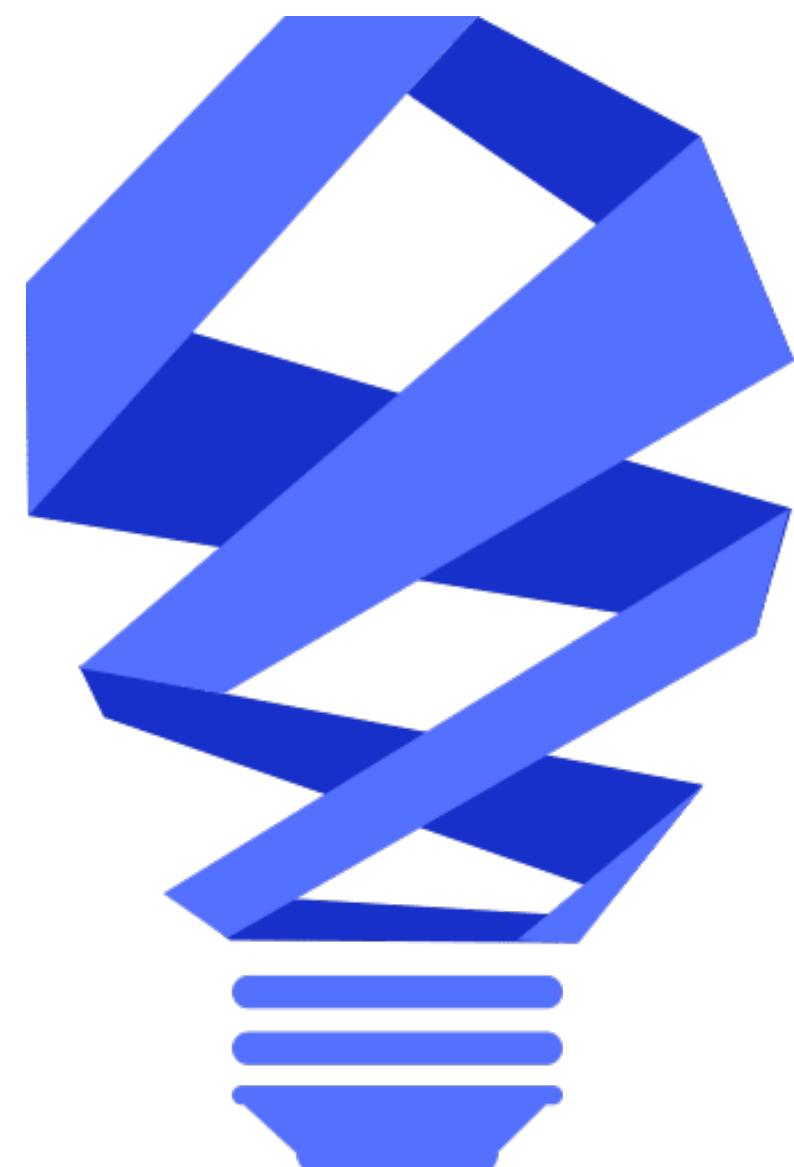
对入参进行校验，不合法入参将不会发起请求

4

监控统计配置

接口监控、统计相关的配置

6



数据源

数据源是一个 class，以配置数据为参，实例化后可以得到一个请求对象。

```
// 数据源模型
interface iDataSource {
  url: string; // 请求地址
  params: object; // 请求参数
  verify (params: object): boolean; // 参数验证
  beforeRequest (config: MPMComConfig): any; // 请求前置处理
  afterResponse (result: any): any; // 请求后置处理
  batch: iBatch; // 聚合分发策略
  monitor: iMonitor; // 监控统计配置
}
```

每个数据源都有自己的名称标识，调用层通过[指定数据源标识](#)，来选择调用某个数据源。

```
// 组件/模板直接调用数据源获取「商品」
dataServive.fetch({ source: 'skus', ...data })
```

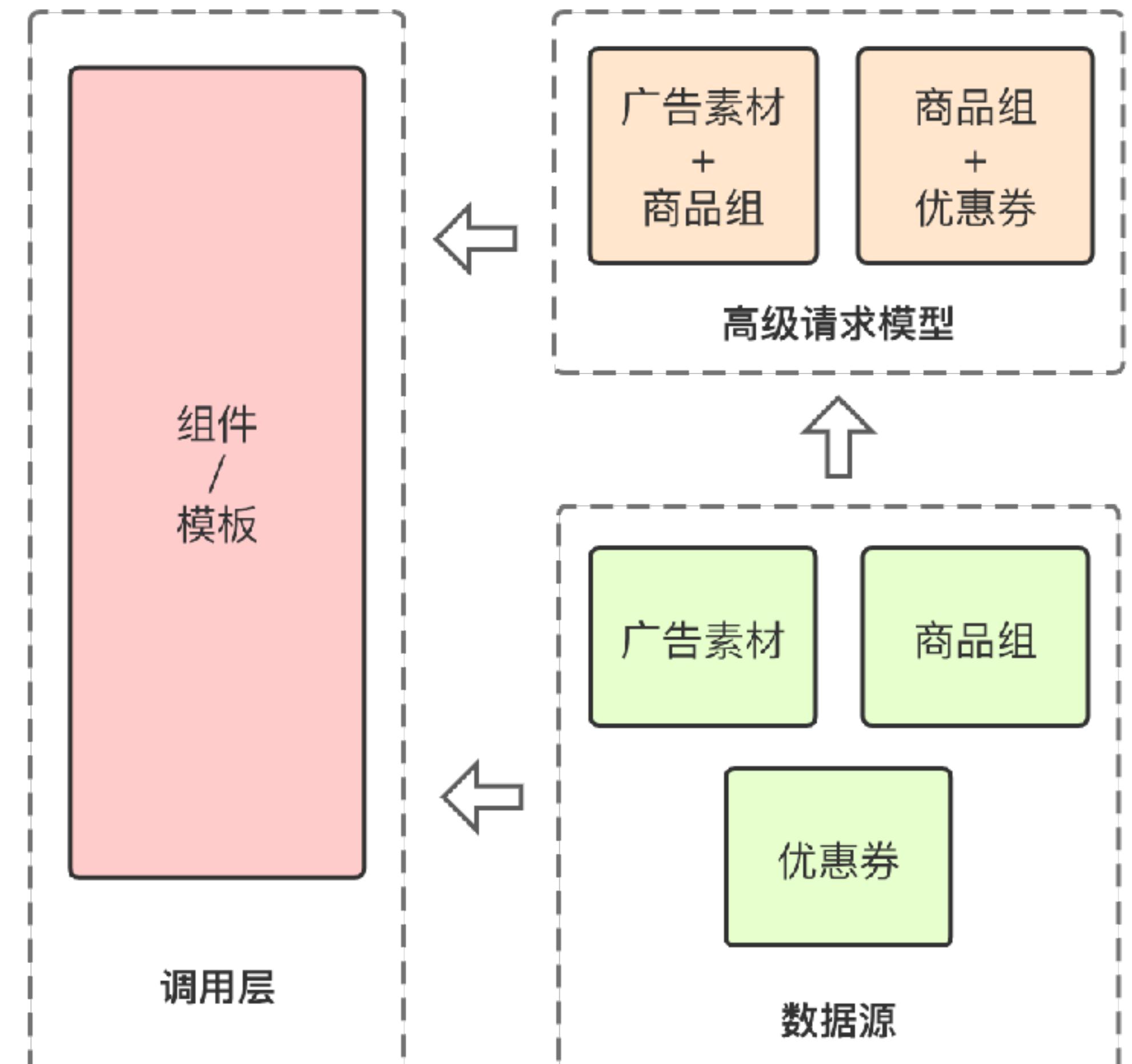
| 自由组合

1 数据源理应纯粹而专一

数据源应该只做一件简单的事，比如请求响应的一些通用处理，特殊业务逻辑不应该在数据源中出现。

2 高级请求模型由数据源组合而成

复杂的请求模型可以由多个数据源组合而成，调用层可以直接调用数据源，也调用封装好的高级请求模型。



| 自由组合

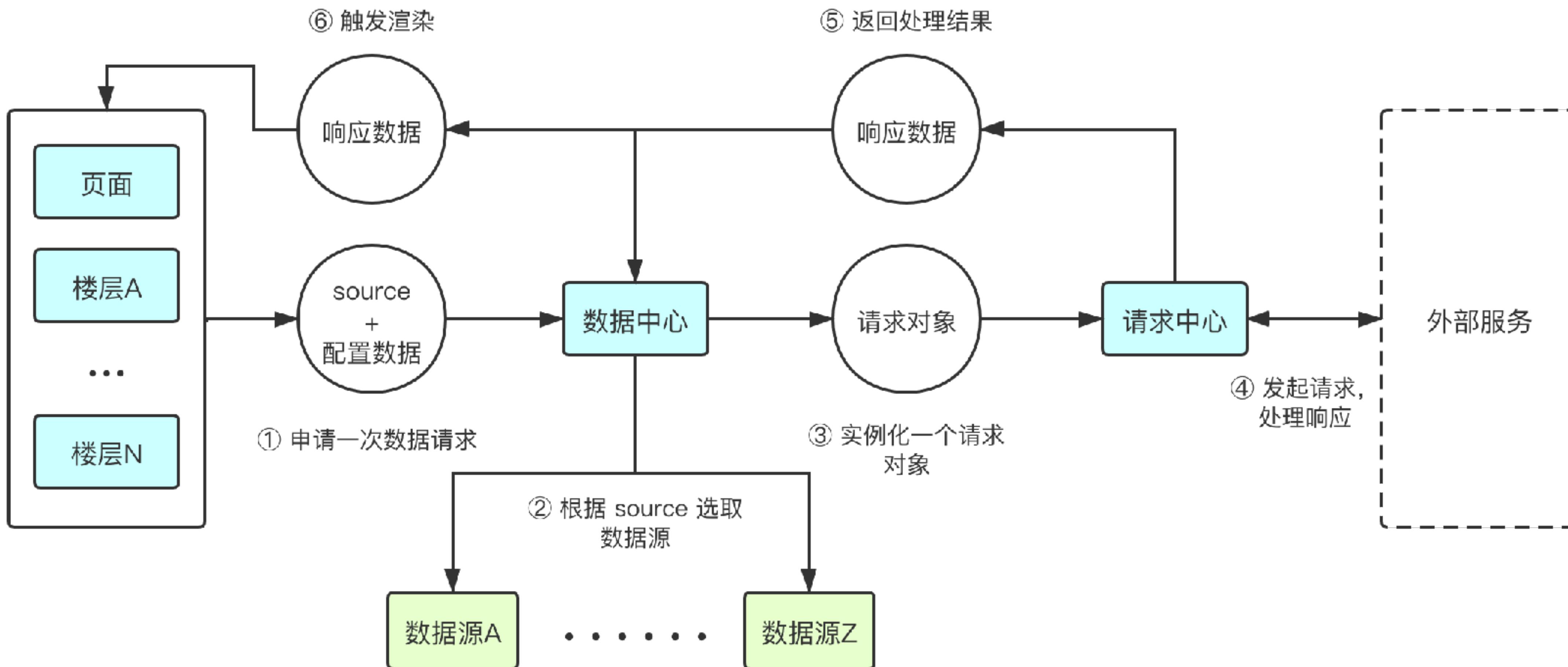
请求模型的组合采用最简单灵活的函数调用，这种组织方式对多变的请求模型十分有利。

向上，我们将调用方式对齐，对开发者来说，调用单一数据源还是高级请求模型，没什么区别。

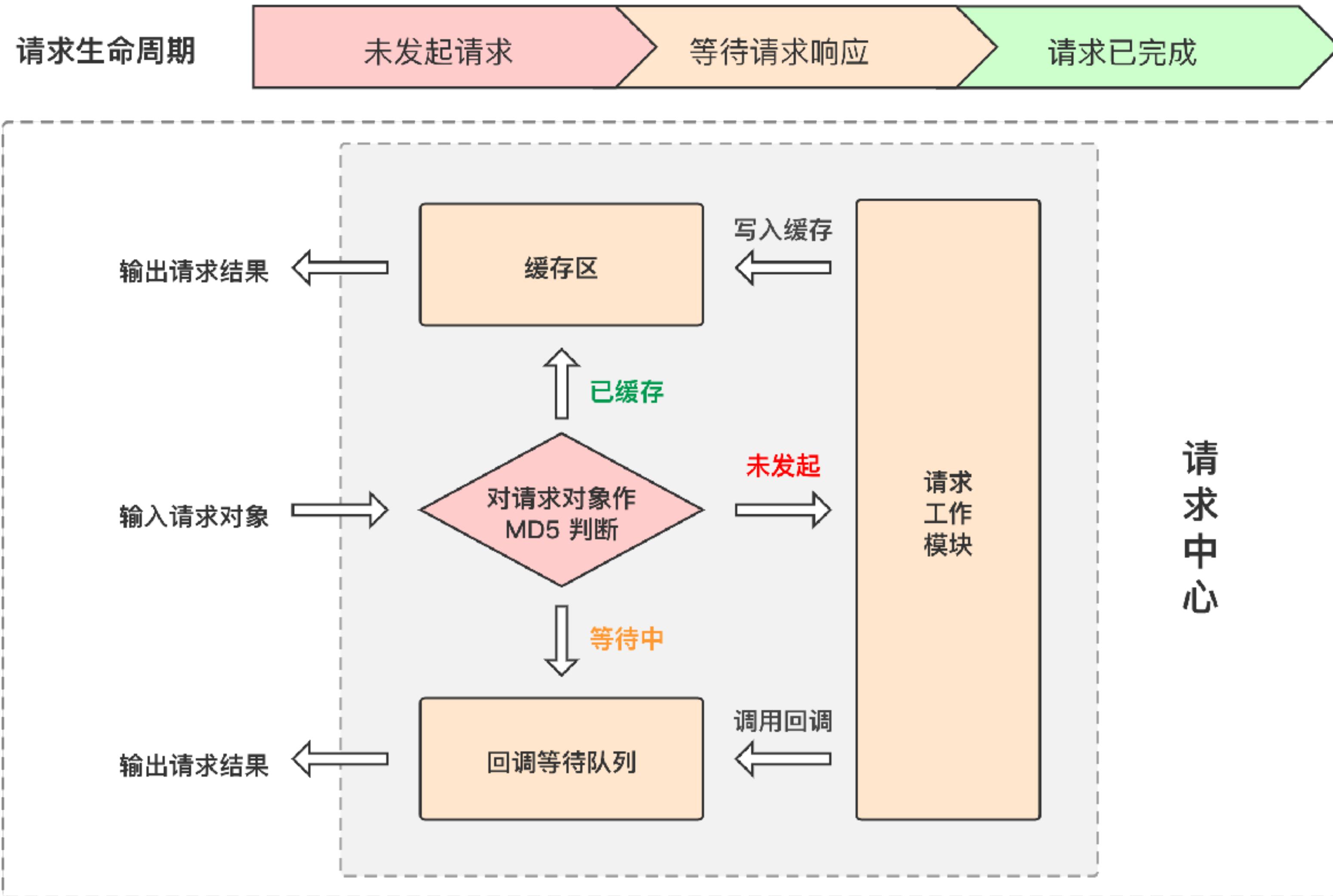
```
// 商品 + 优惠券
async skuWithCoupon (data) {
  // ...
  const skus = await dataServive.fetch({ source: 'skus' });
  const coupons = await dataServive.fetch({ source: 'coupon' });
  // ...
  return result;
}
```

```
// 组件/模板间接调用数据源获取「商品+优惠券」
dataServive.fetch({ source: 'skuWithCoupon', ...data });
```

| 统一管理



请求优化策略



避免重复请求

依靠一个简单的请求队列
和请求缓存，我们有效避免了
页面内发起重复请求

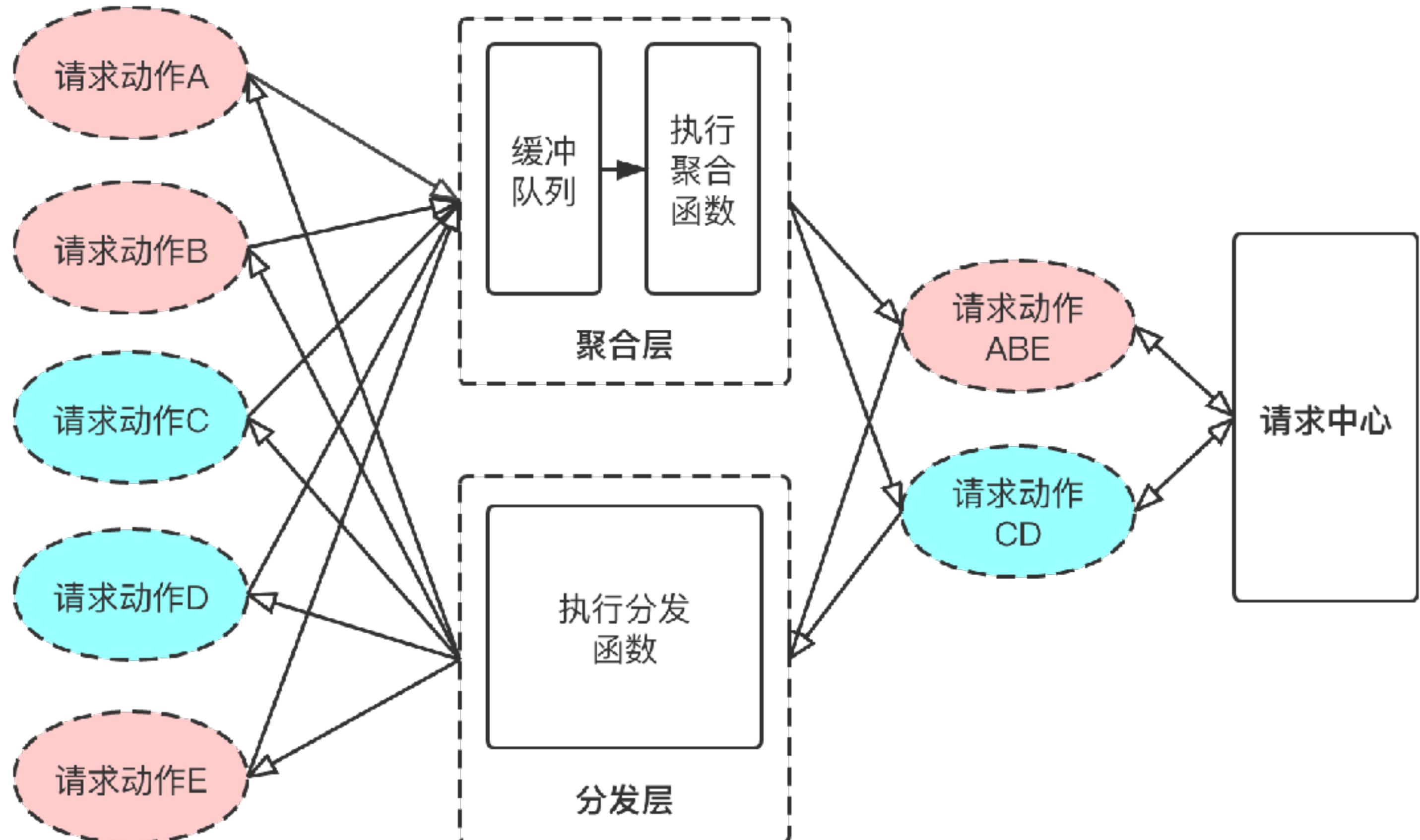
请求的聚合分发策略

```
interface iBatch {  
    limit: number  
    pack: (actions: Action[]) => Action  
    unpack: (result: any, actions: Action[]) => Record<ActionMd5, ActionData>  
}
```

- 1 数据源可制定聚合分发策略，使得同类请求对象在发出前经 pack 合并，响应后经 unpack 拆包分发

- 2 引擎以队列收集请求对象，会等到下一个 Tick 再发起请求，在这个 Tick 中，如果队列超限，就会提前发起

| 请求优化策略



合并同类请求

每个数据源可定制自己的
聚合分发策略，以此让同类
请求合多为一

初态函数

- 三端同构 -



灵魂拷问：前后端渲染有什么区别？

我想把客户端渲染那一套，照搬到直出端，为什么不行？

| 假如不考虑同构

客户端的渲染流程

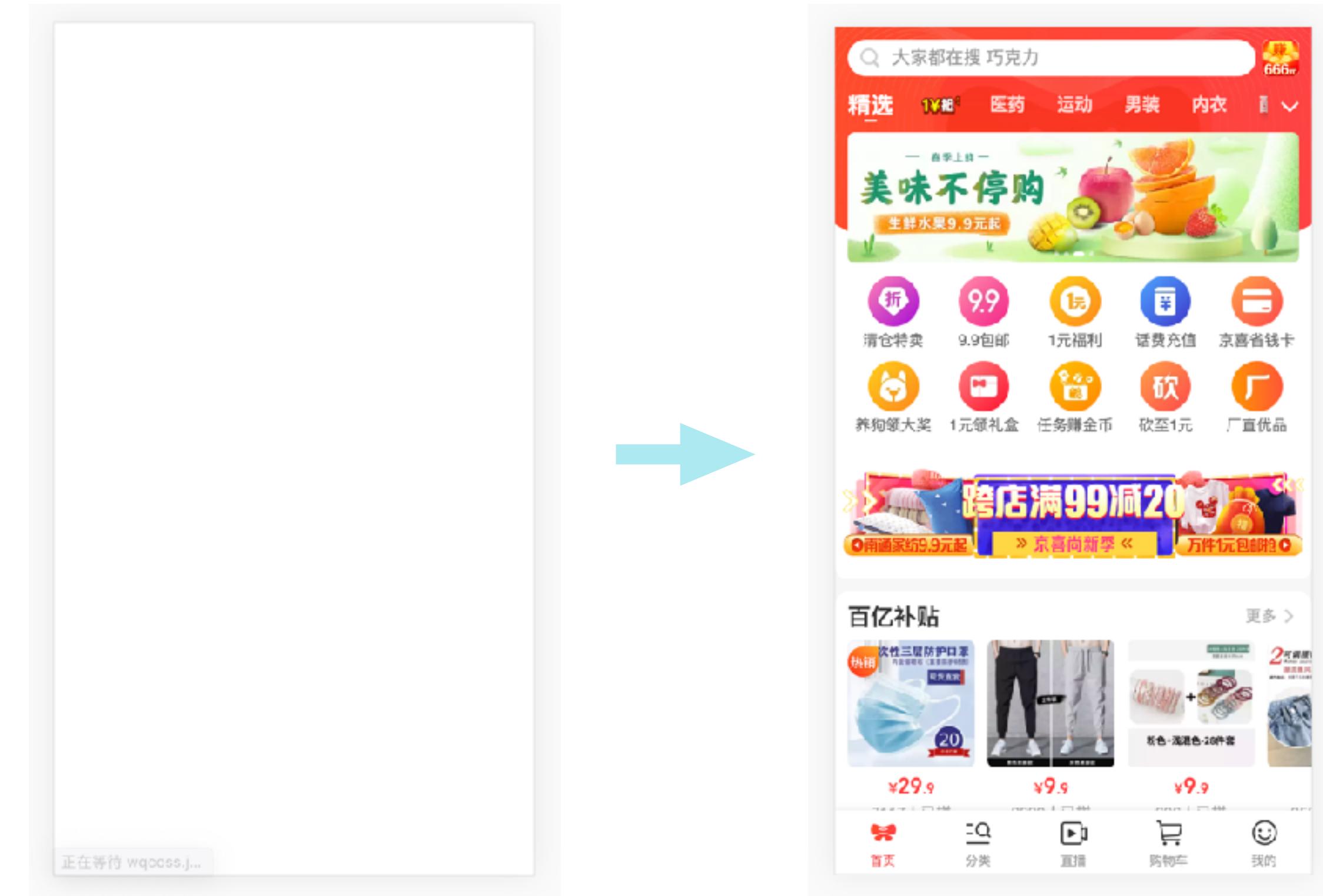
客户端允许存在多趟渲染，中间以骨架屏占位，渲染与请求没有严格的先后顺序



| 假如不考虑同构

直出端的渲染流程

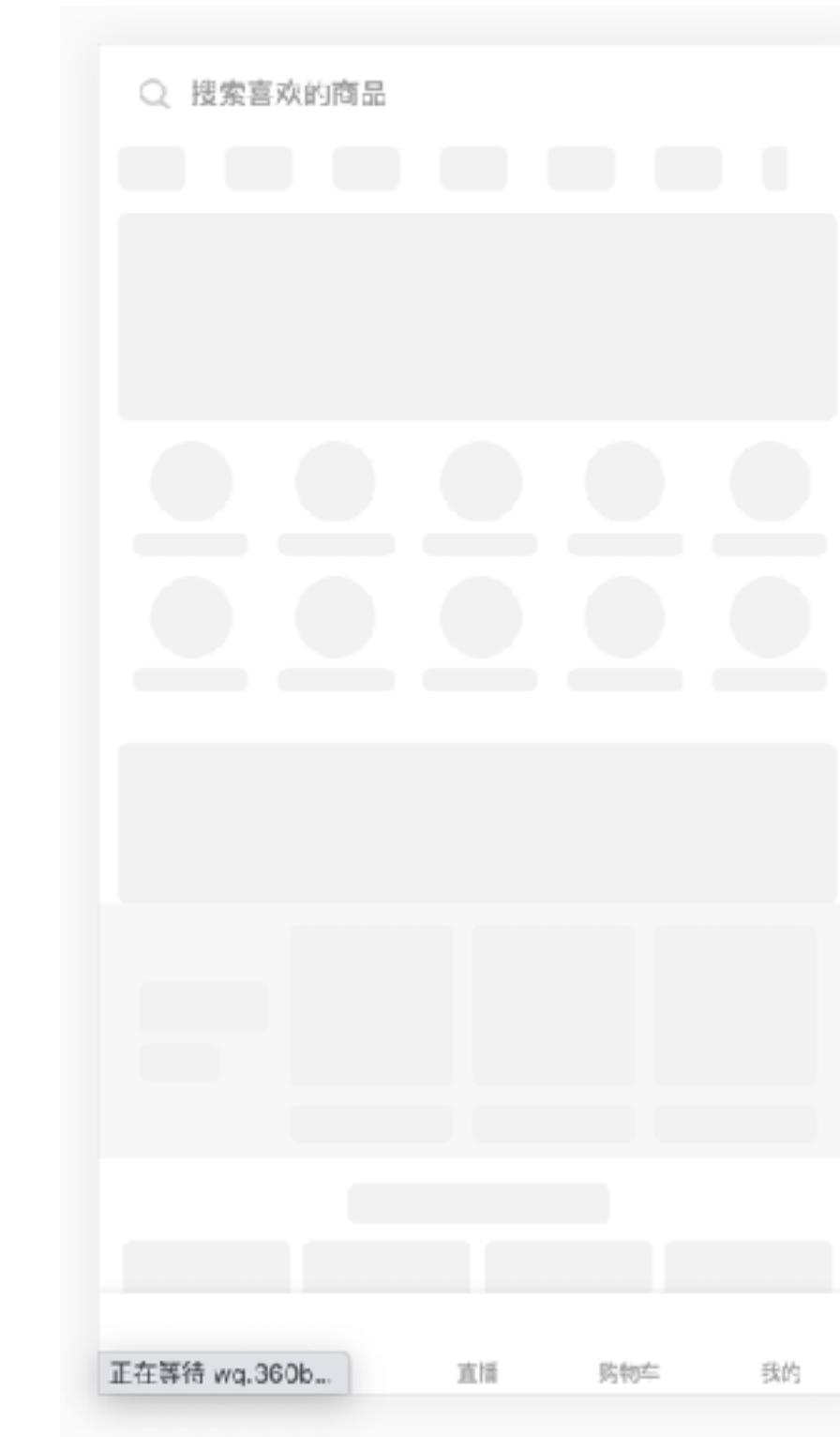
直出端只有一趟渲染，渲染前要求数据全部到位，所以请求必须在渲染之前完成



| 假如不考虑同构

结果你直出了一份骨架屏？

如果你把客户端渲染直接搬到直出端，很遗憾，你可能就只能直出一份骨架屏



| 基于三端同构的请求模型

1

同构的关键在于初态渲染

前后端同构的关键就是初态渲染，也就是页面的初始化渲染阶段。

2

补充适配直出端的生命周期

Vue 现有生命周期没有任何一个能够满足直出端的异步数据获取，要实现直出，数据模型就必须补充适配直出渲染的生命周期。

3

三端解析流程应保持统一

要实现三端同构，我们还需要规范解析流程，让三端解析渲染流程保持高度统一。



| 初态函数

位于组件生命周期之前的**异步函数**

```
interface iMPMComponent {  
    // ...  
    getInitialState (config: MPMComConfig): Promise<MPMComData>;  
    // ...  
}
```

每个 MPM 组件都有一个初态函数，初态函数以组件配置数据为入参，异步返回用于组件初始化渲染的初态数据。

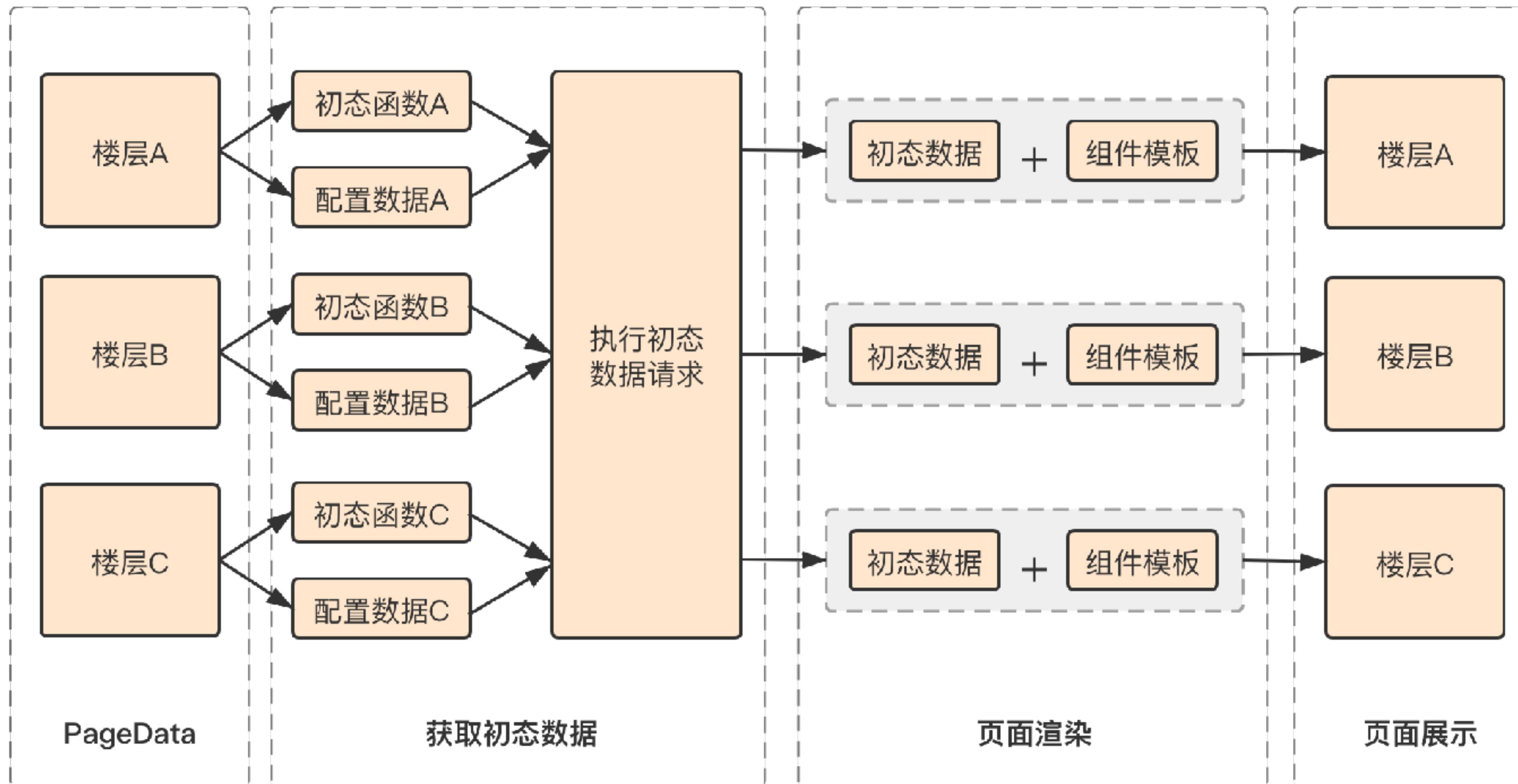
1

三端引擎在创建组件实例前，将先收集执行各组件的初态函数，并将函数的异步返回结果作为组件初始化渲染的数据。

2

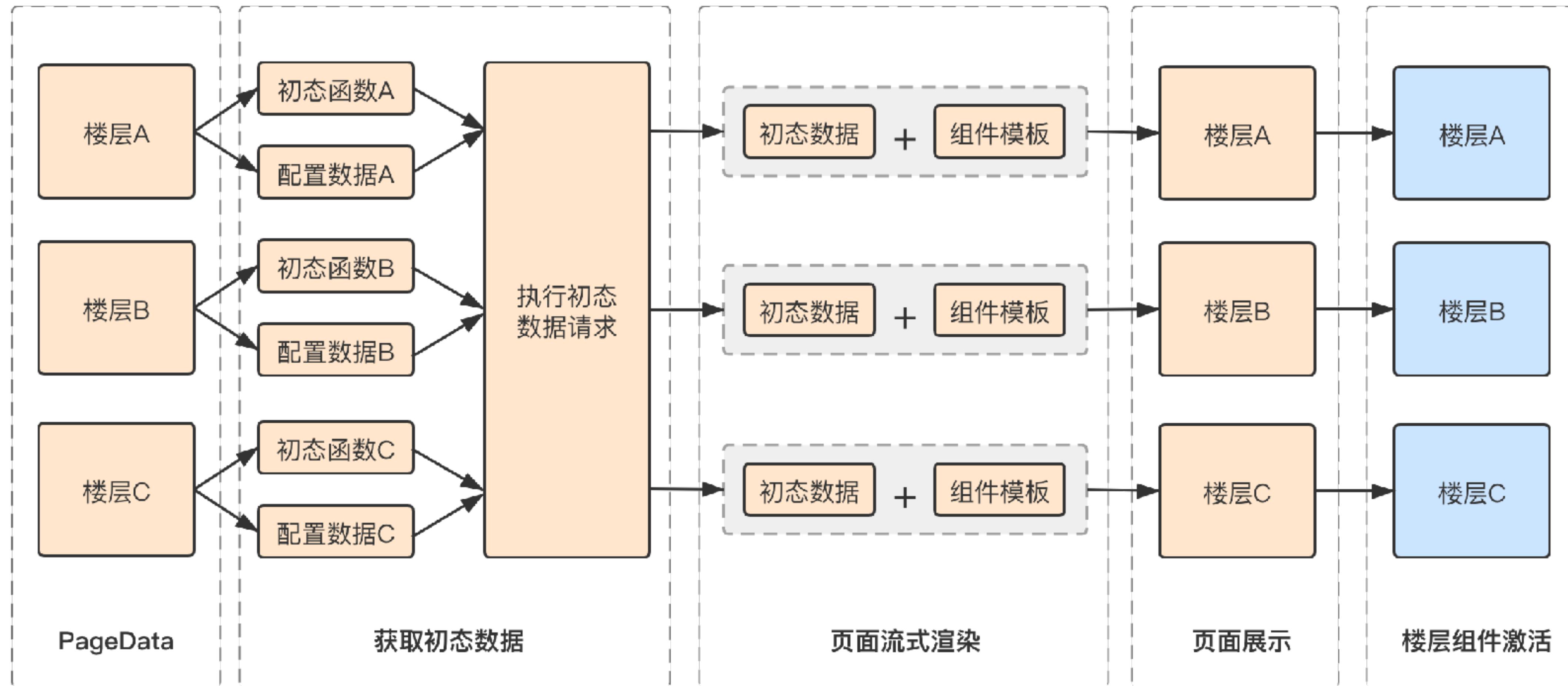
| 基于三端同构的解析流

静态 H5 / 小程序端



| 基于三端同构的解析流

直出端



刁钻场景：主次接口分端渲染

我有一个组件，串联请求了主、次两个接口，可是次接口内容没那么重要，为了提高直出效率，能不能只直出主接口，次接口等到了客户端再请求呢？

主次接口数据的分端请求渲染

主次接口串联时，主接口在直出端完成，次接口在客户端补充，提高直出速度

```
interface iMPMComponent {  
    // ...  
    getInitialState (config: MPMComConfig, callback: (data: MPMComData) => boolean): void;  
    // ...  
}
```

初态函数的第二个参数是一个回调
函数，传入初态数据并执行可以触
发引擎立刻渲染，因此，利用它可
以实现初态数据的分段渲染。

1

对于这类初态函数，直出端只会响应
第一个 callback，余下的 callback 将
直接忽略，等到了客户端之后再作执
行，补充初态数据。

2

05

总结





01

相比独立开发一个页面，搭建场景开发可能随处都要求着严谨的设计，任何你认为的微不足道都不应该被忽视

02

数据模型解决方案更多只是一套开发范式，没有规范约束的搭建系统是不可维护的

03

独立和统一并不矛盾，虽然搭建的目的是自由组合，但在设计开发时却必须足够重视统一的思想

THANKS
感谢您的观看

欢迎关注我们

—



年度直播：

1/11

前端转管理

2/29

前端搞基建

3/28

前端搞搭建

4/11

前端搞规划

4/25

前端搞监控

5/16

联票群彩蛋

5/30

前端搞构建

6/27

前端搞跨端

7/25

前端搞部署

8/29

前端搞可视化

9/26

前端搞性能

10/31

前端搞安全

11/28

前端搞框架

12/26

前端搞报表

第四届前端早早聊大会

前端成长的新起点 | 前端如何搞规划 - 4/11



| | | | | |
|----|-------------|---------------------|-----|-------|
| 堂主 | 高级前端专家 P8 | 《如何在初级到专家的路上打破成长瓶颈》 | 政采云 | 13:00 |
|----|-------------|---------------------|-----|-------|

| | | | | |
|----|-------------|----------------------|----|-------|
| 远舟 | 高级前端专家 P8 | 《如何做出专家级别的技术与技术产品规划》 | 税友 | 14:00 |
|----|-------------|----------------------|----|-------|

| | | | | |
|----|-------------|---------------------------|-----|-------|
| 城池 | 高级前端专家 P8 | 《如何对 P5/P6/P7 做职业规划和技术培养》 | 阿里云 | 15:00 |
|----|-------------|---------------------------|-----|-------|

| | | | | |
|----|-------------|----------------------|----|-------|
| 贵重 | 高级前端专家 P8 | 《如何在团队中培养出更多的前端技术专家》 | 钉钉 | 16:00 |
|----|-------------|----------------------|----|-------|

跟进大会 PPT/录播视频/讲稿文章

跟进下届大会新动态/讲师征集

跟进大会优秀团队/招聘岗位

大会报名跟进/扫码关注



掘金