

# 同程艺龙小程序前端框架

## TheNew

### 及其配套体系建设

# 王硕

@同程艺龙住宿研发中心

小程序前端框架  
监控和数据平台

# 1 设计思路和架构

# 曾经面临的问题

体系分化

功能重复

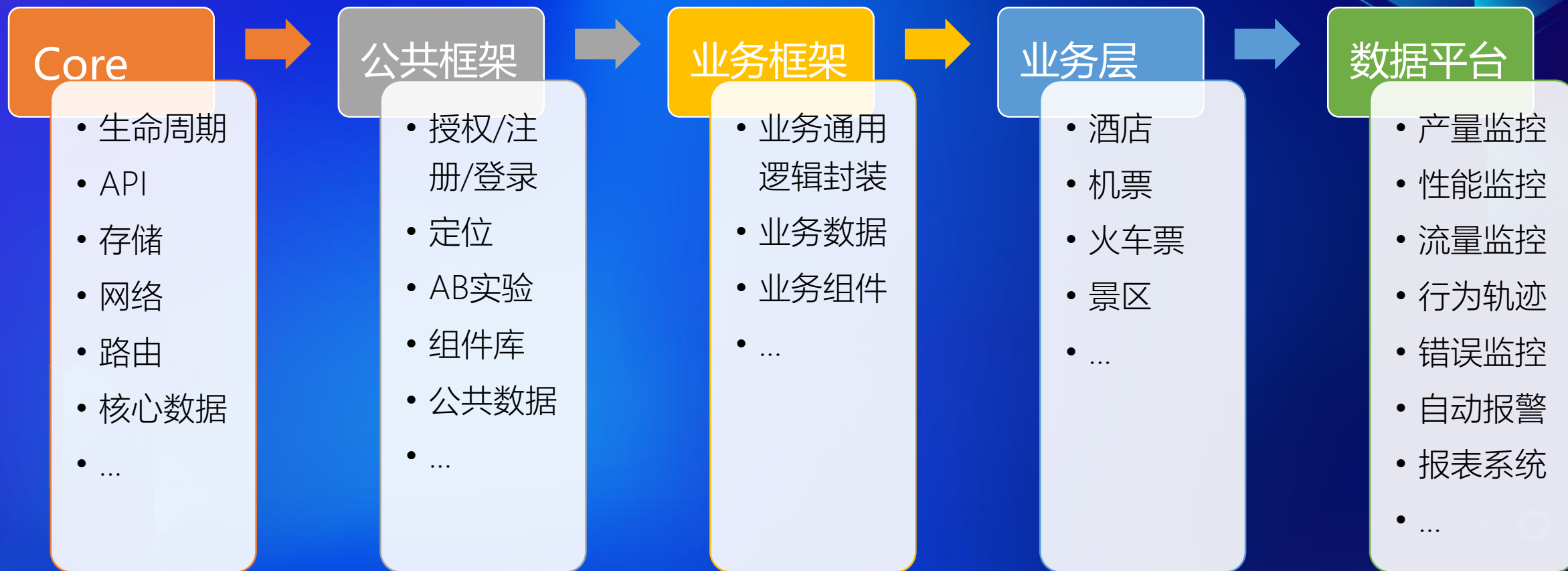
性能不佳

监控薄弱

# 架构

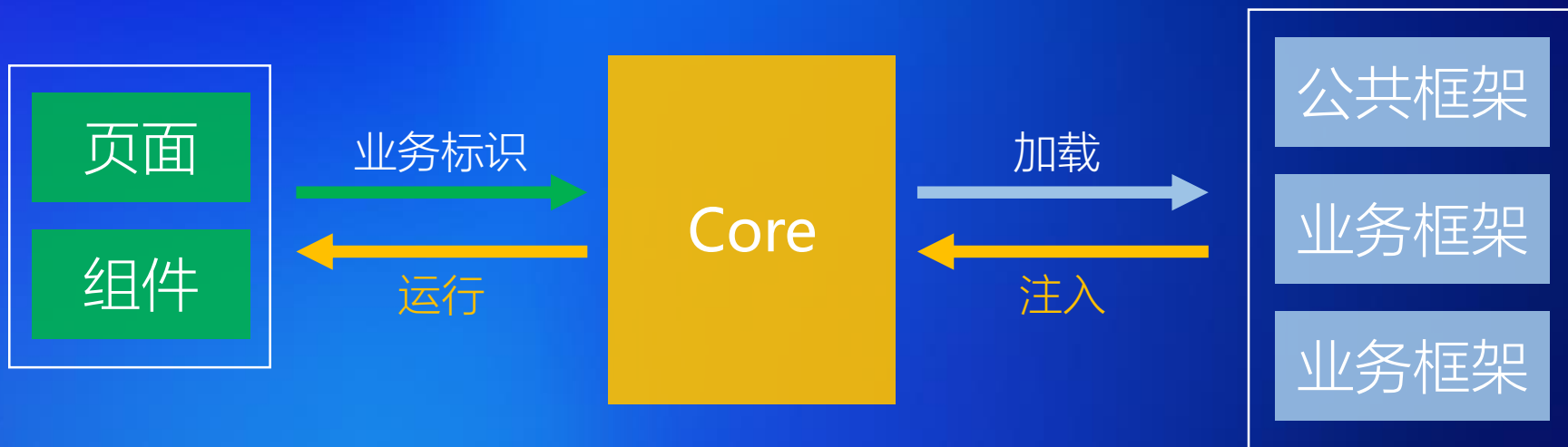


# 架构



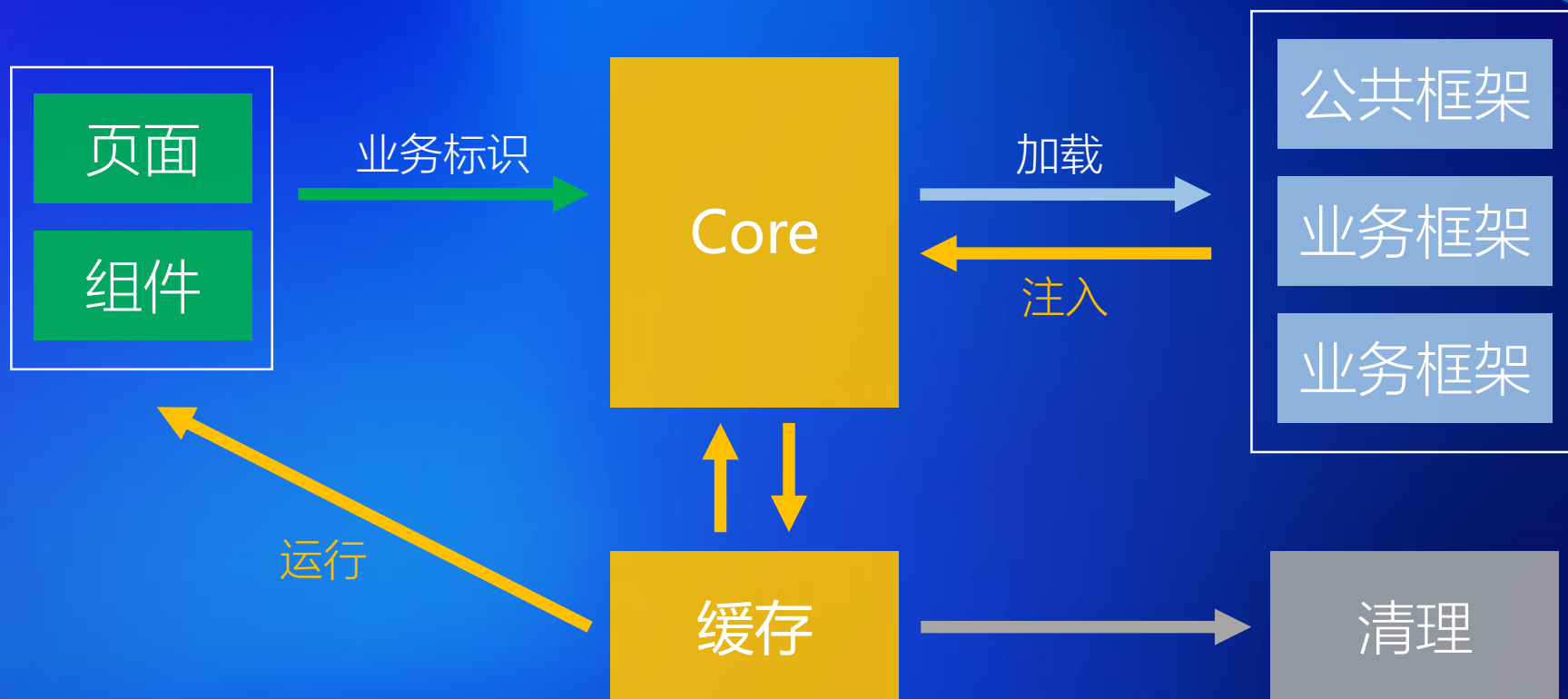
## 2 运行机制

# 运行原理





# 业务框架的缓存和回收



# 动态调用机制

```
class Demo extends wx.tn.BasePage {  
  // ...  
  customFunc1 () {  
    this.$api.setCache('foo', 0)  
  }  
  
  customFunc2 () {  
    this.$api.setCache('bar', 1)  
  }  
}
```

存入

取出

处理依赖

缓存

自动清理

取出

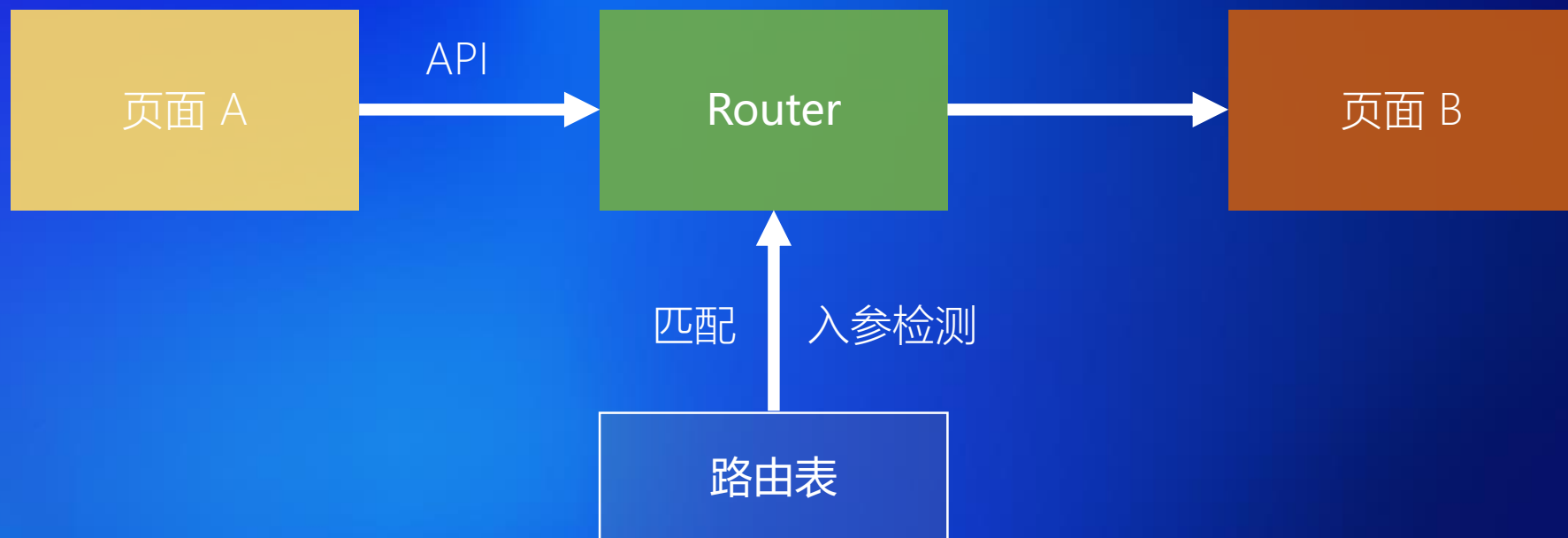
```
class Demo1 extends wx.tn.BasePage {  
  // ...  
  customFunc1 () {  
    this.$api.setCache('foo', 0)  
  }  
}
```

### 3 功能增强和性能优化

# 路由管理

- 希望给页面传递复杂的入参
- 页面路径发生变化，但跳转操作散落在其他上百个页面之中
- 不知道页面入参是否合法

# 路由管理



# 路由管理

```
export default {
  basePath: '/page/hotel',
  routes: [{
    name: 'hotel-list',
    path: '/list',
    query: {
      city: {
        type: [String, Number],
        required: true
      },
      foo: Object
    }
  }]
}
```

路由配置

```
this.$router.push({
  name: 'hotel-list',
  query: {
    city: 0101,
    foo: {
      bar: 1
    }
  }
})
```

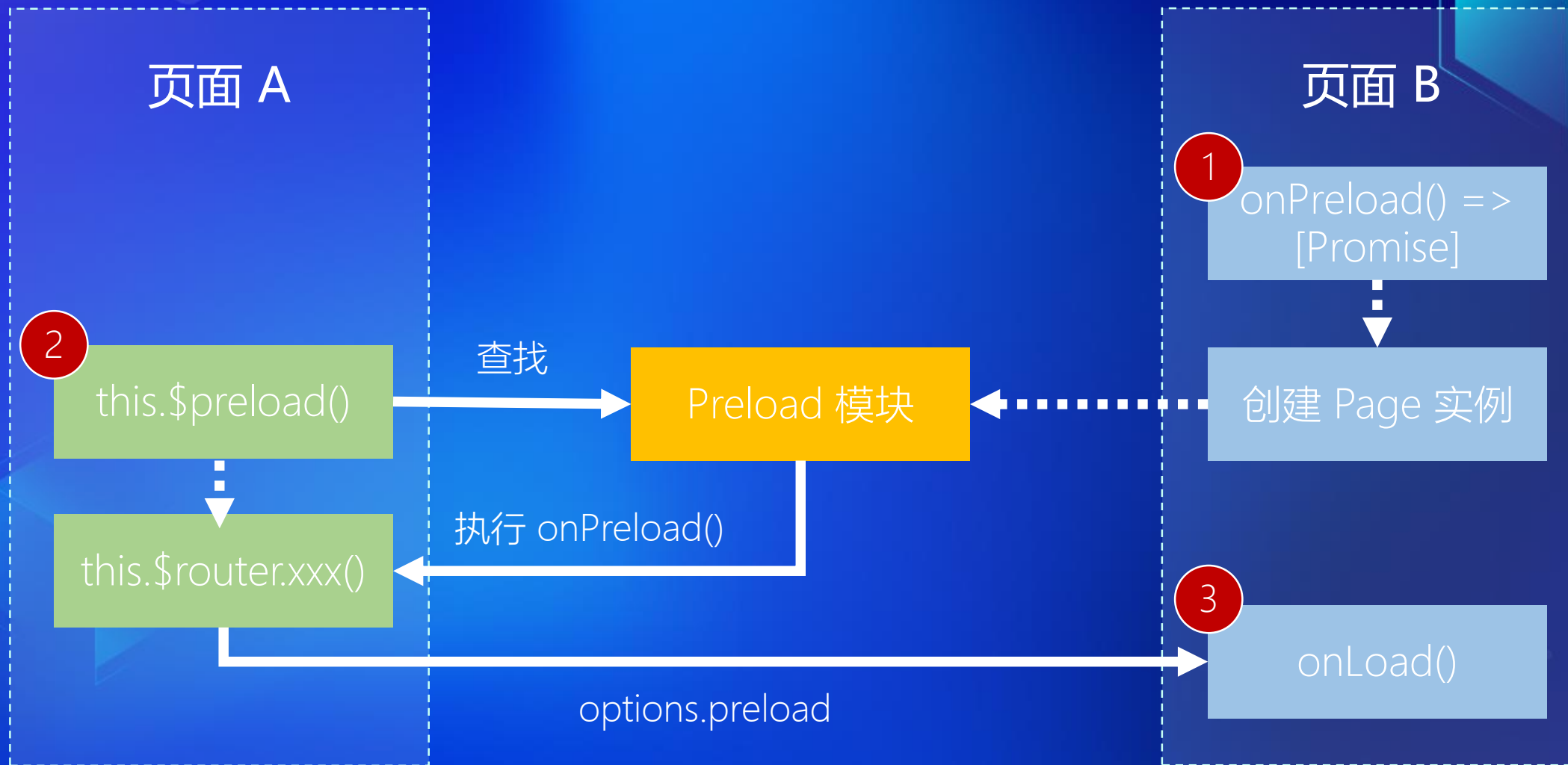
路由跳转

# 预加载机制





# 预加载机制





# 网络请求管理

业务线配置

Cookies

队列管理

优先级管理

超时管理

自动重试管理

请求终止

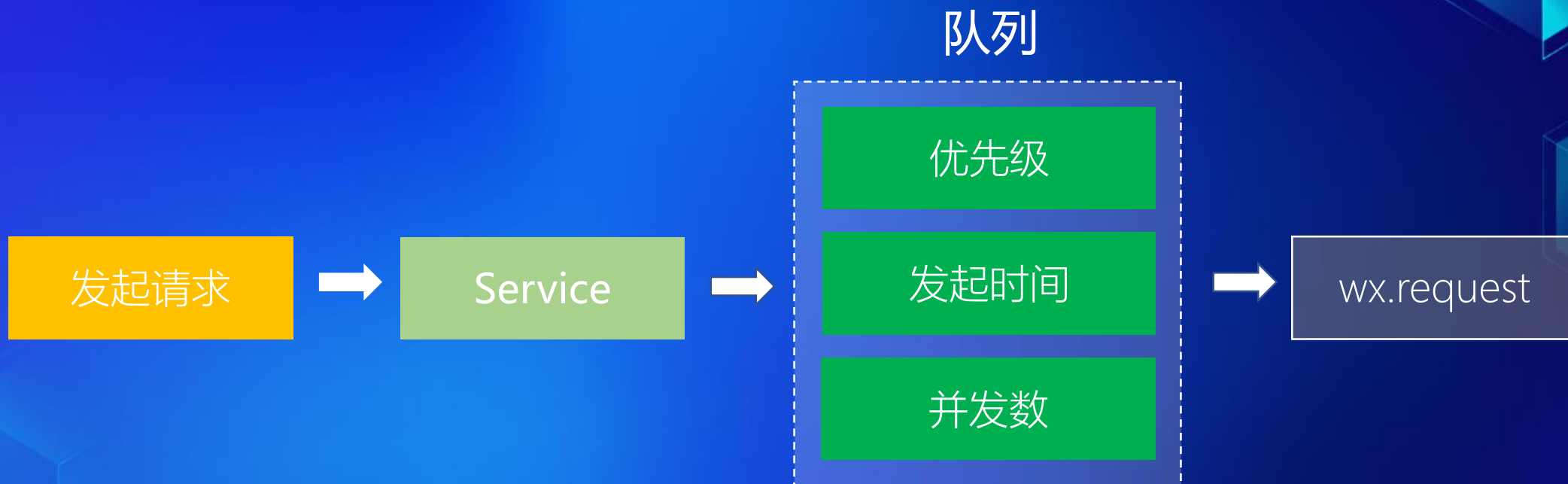
缓存

Mock

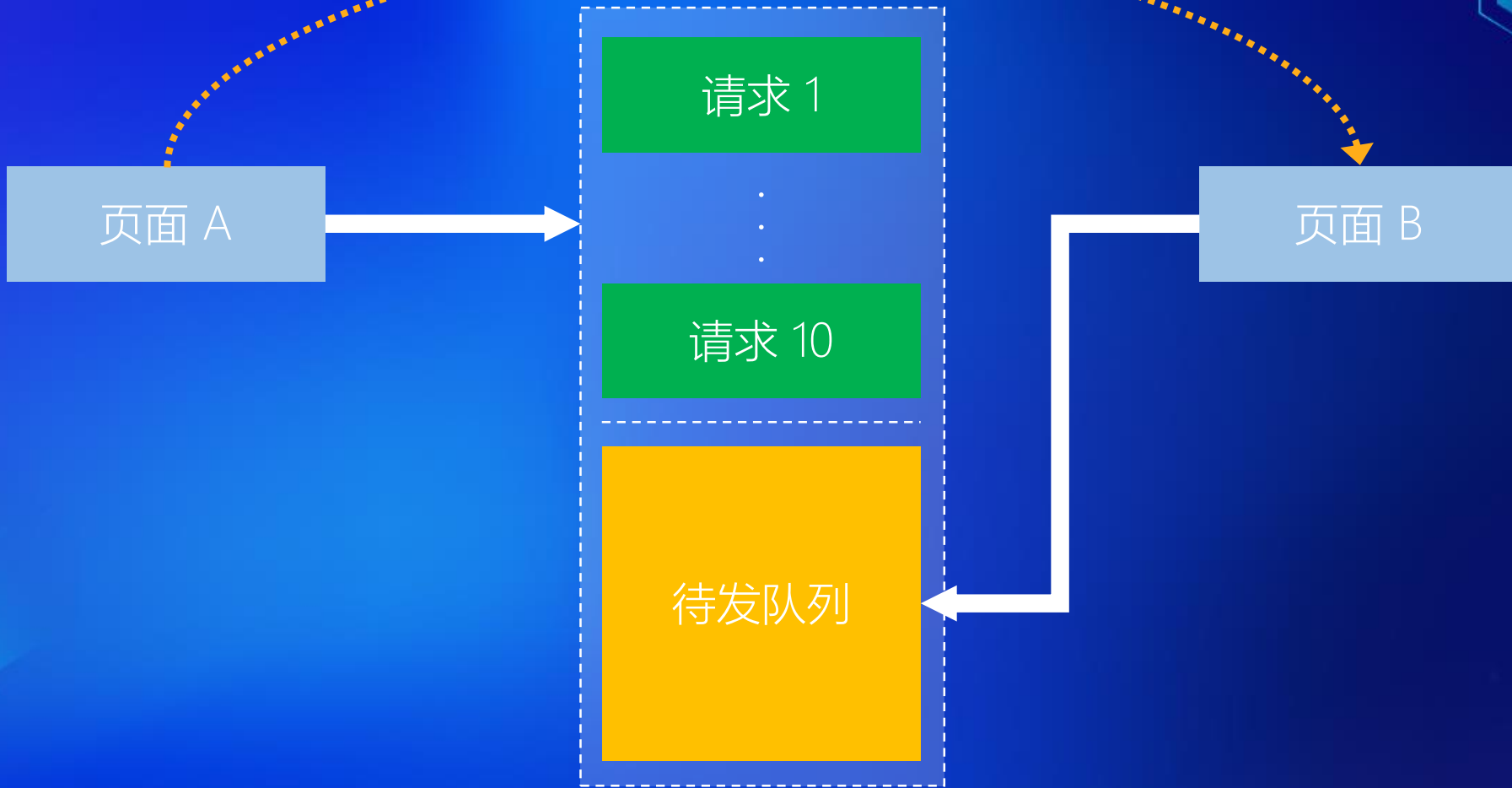
拦截器

请求结果校验

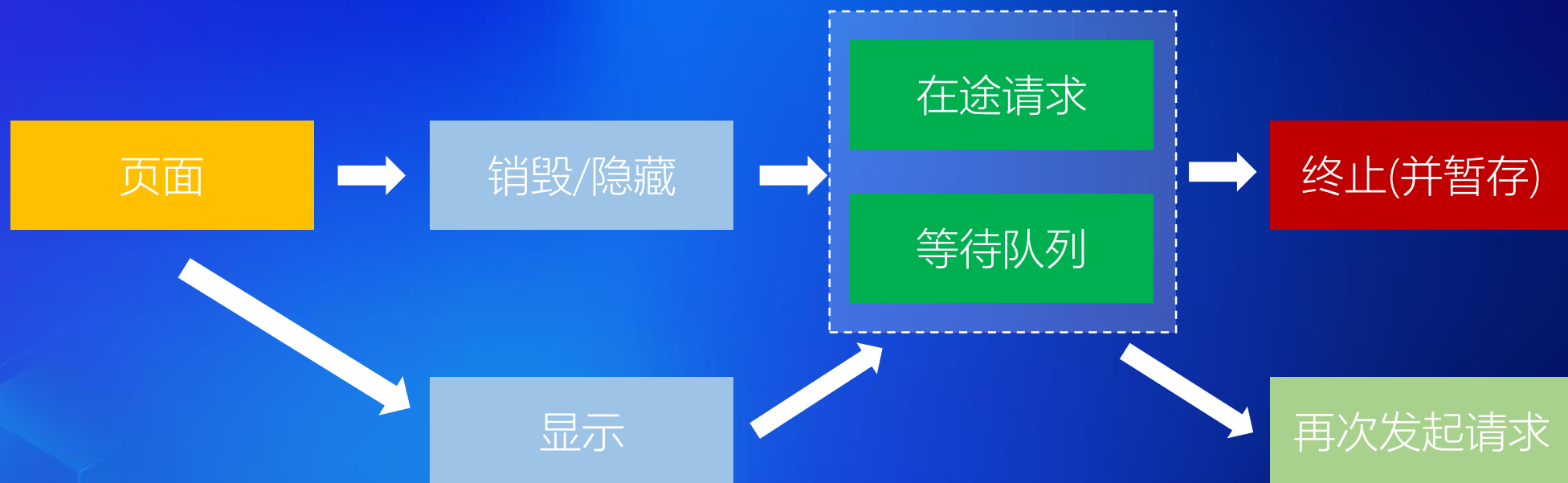
# 队列和优先级管理



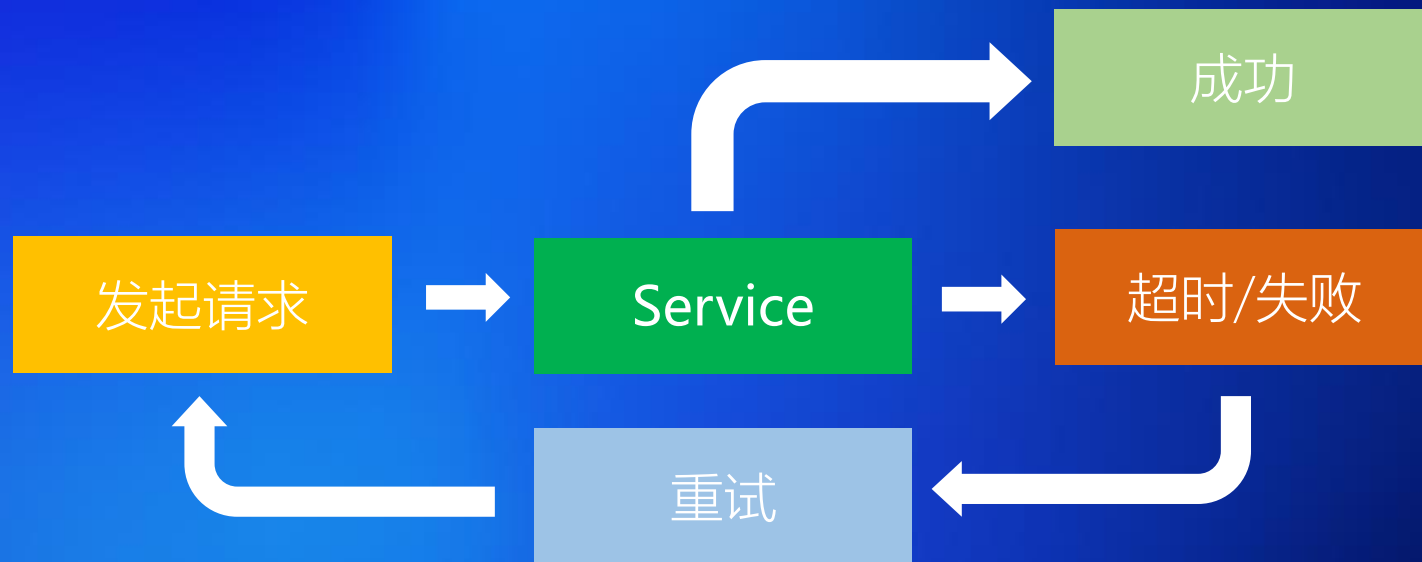
跳转



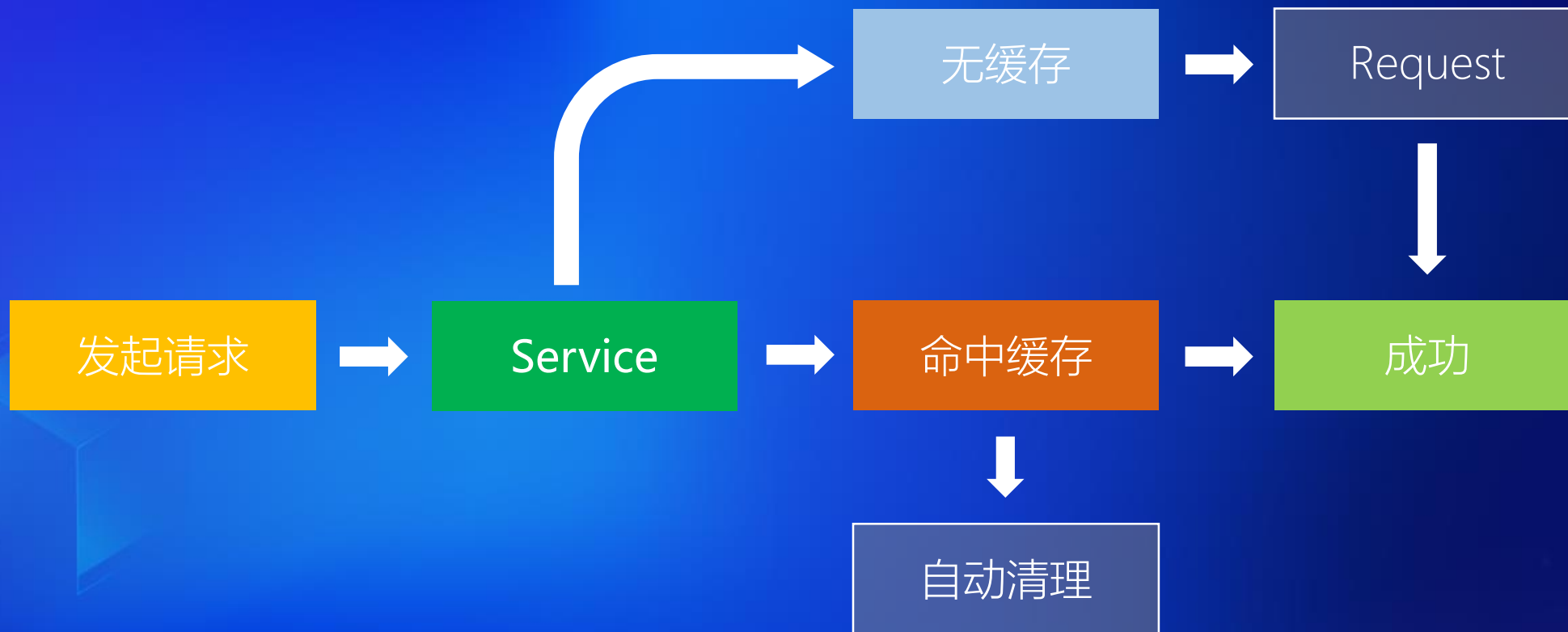
# 请求终止策略



# 超时/失败和自动重试



# 请求缓存机制



# 请求配置示例

- 获取酒店列表数据
- 优先级为 3
- 3 秒内未返回结果即为超时
- 超时或出错后，最多重试 3 次
- 请求成功后，缓存 30s 内有效
- 页面隐藏后，可以终止

```
this.service.get('/getHotelList', {  
  priority: 3,  
  timeout: 3000,  
  retry: 3,  
  cache: 30 * 1000,  
  onHideAbort: true  
})
```



# SetData 优化





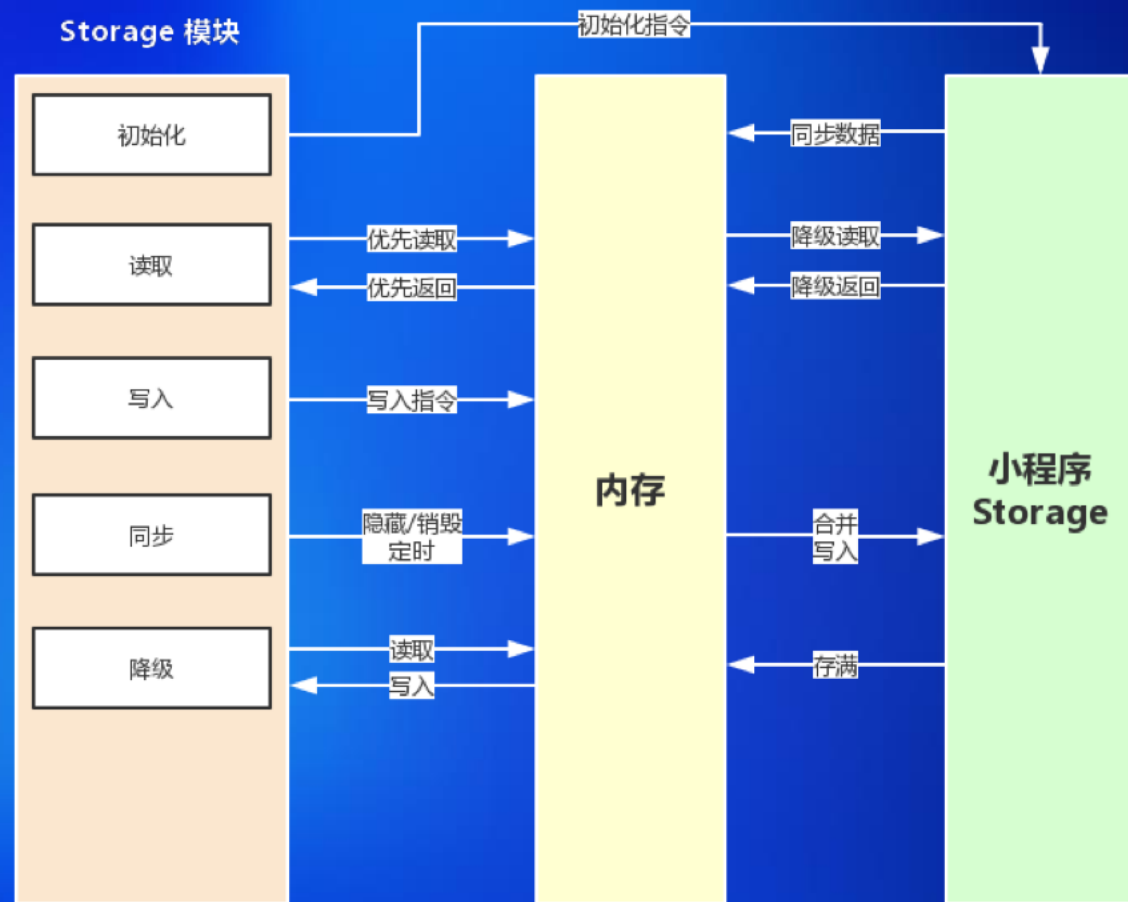
# setData 优化原理



# Storage 的性能问题

| 机型                      | 浏览器    | 微信小程序   | 性能差距 |
|-------------------------|--------|---------|------|
| Android<br>(MI 2)       | 1243ms | 21610ms | 约17倍 |
| iOS<br>(iPhone 6s Plus) | 141ms  | 1439ms  | 约10倍 |

# Storage 存储优化



# Storage 存储优化

## 本地存储

本页面展示了本地存储的相关功能和性能优化后的效果。其中 API 测试方式为写入 100 个长度为 100 byte 的随机字符串。

### 小程序原生 API

|                      |      |
|----------------------|------|
| wx_setStorageSync    | 点击测试 |
| wx_getStorageSync    | 点击测试 |
| wx_removeStorageSync | 点击测试 |
| wx_clearStorageSync  | 点击测试 |
| wx_setStorage        | 点击测试 |
| wx_getStorage        | 点击测试 |
| wx_removeStorage     | 点击测试 |
| wx_clearStorage      | 点击测试 |

### the-new API

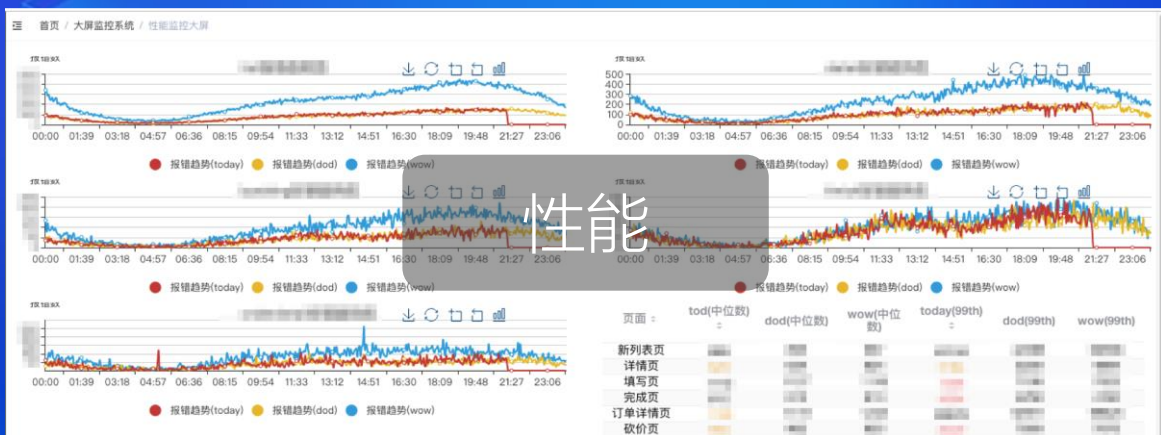
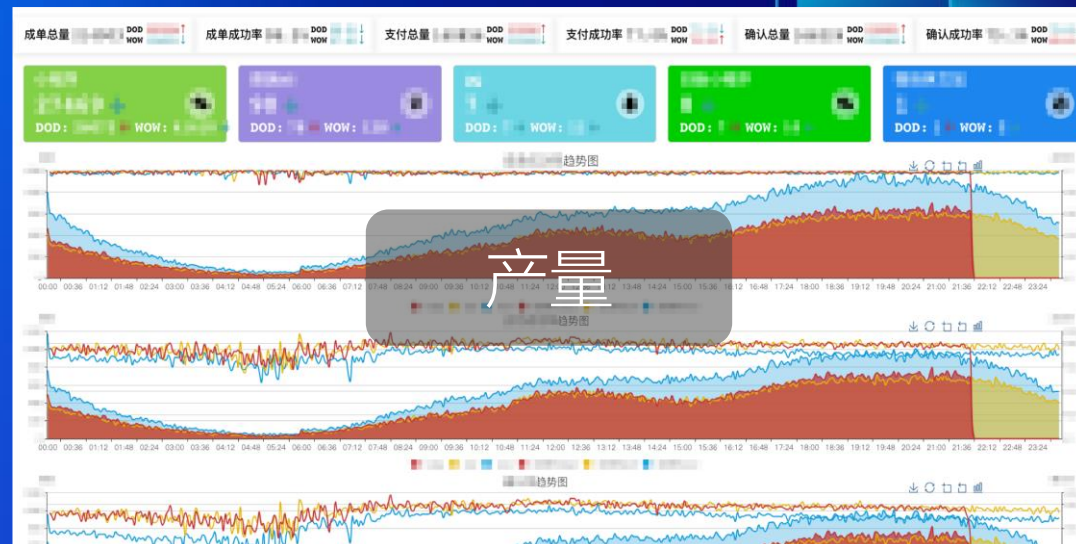
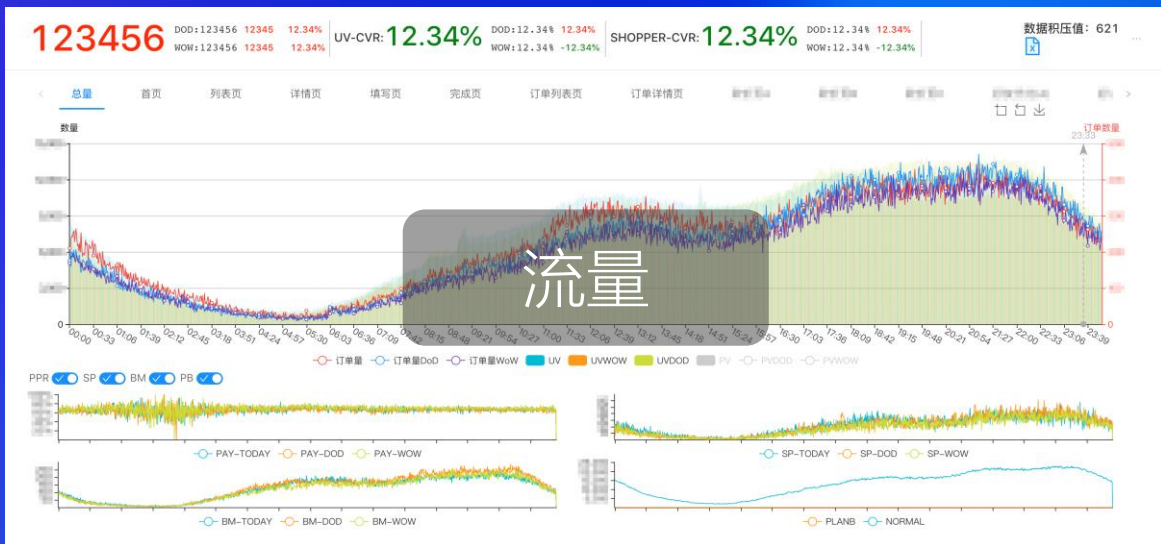
|               |      |
|---------------|------|
| setStorage    | 点击测试 |
| getStorage    | 点击测试 |
| removeStorage | 点击测试 |

4

## 数据采集和监控平台



# 数据监控平台



# 数据采集

业务层

业务数据

公共层

性能

个性化

核心层

用户操作

错误

流量/产量

上报和补发

# 自动采集用户行为数据

点击

长按

重复操作

截屏

多指操作

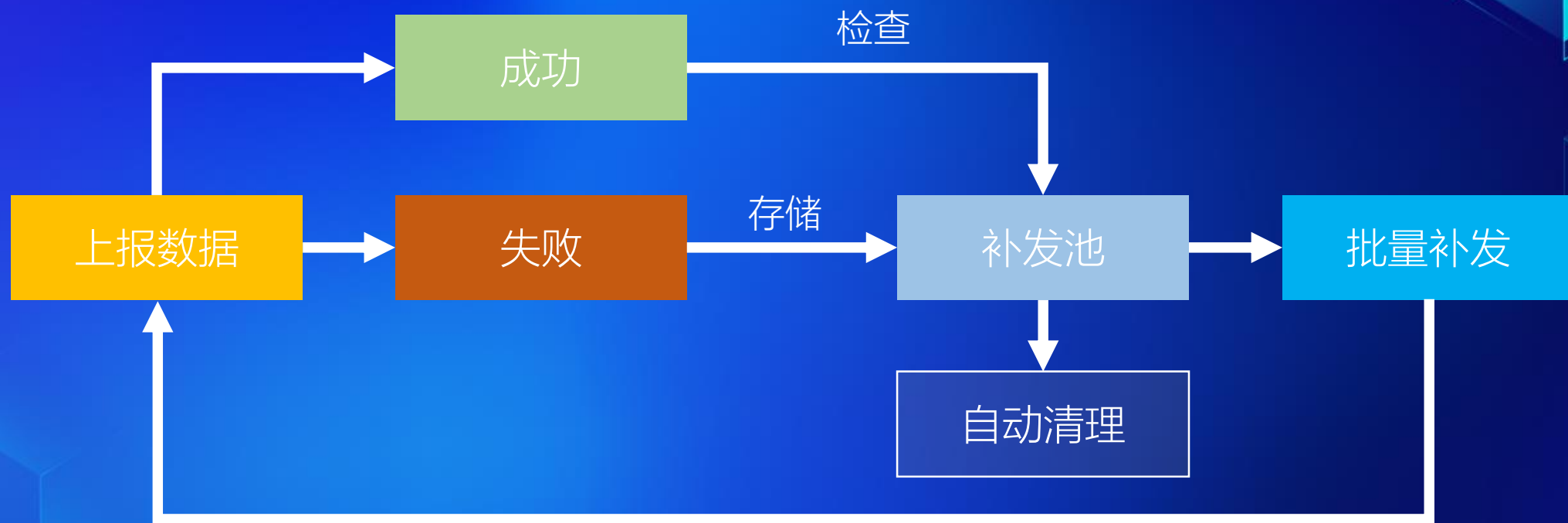
切换到后台

连续点击

滑动



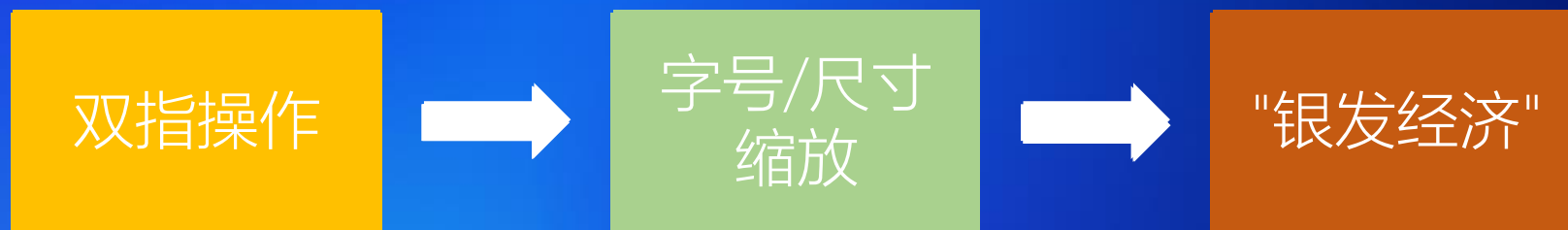
# 数据补发机制



# 5

## 对用户行为数据的思考

# 对用户行为数据的思考



## 6 工程化

# 工程体系建设

代码管理

GitLab

Git Submodules

持续集成

工单系统

自动打包

自动发版

工具化

打包和压缩

体积检测和统计

依赖分析

规范化

体验规范

代码规范

Git 管理规范

数据格式规范

自动化测试

单元测试

UI 测试

黑盒测试

# 谢谢



本PPT来自2019携程技术峰会  
更多信息请关注“携程技术中心”微信公众号~