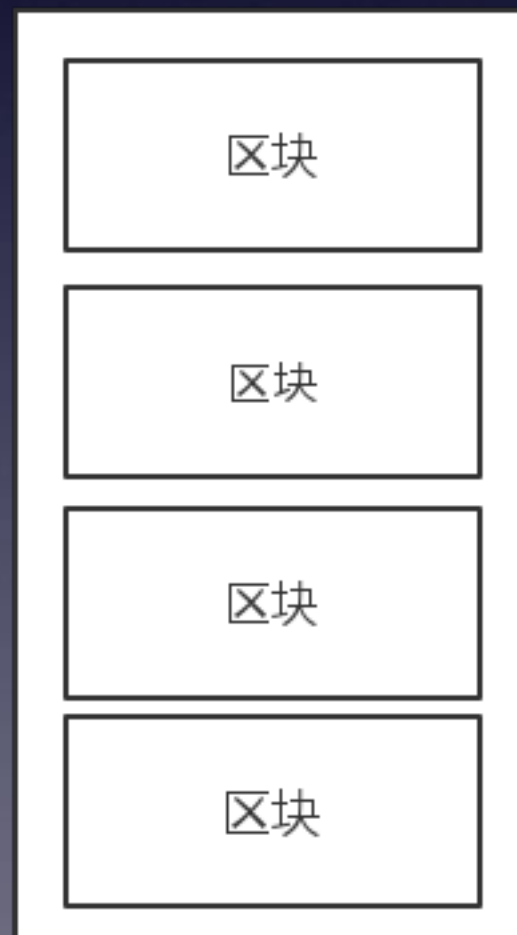


UI模板引擎

思考

页面构成

ListView/TableView

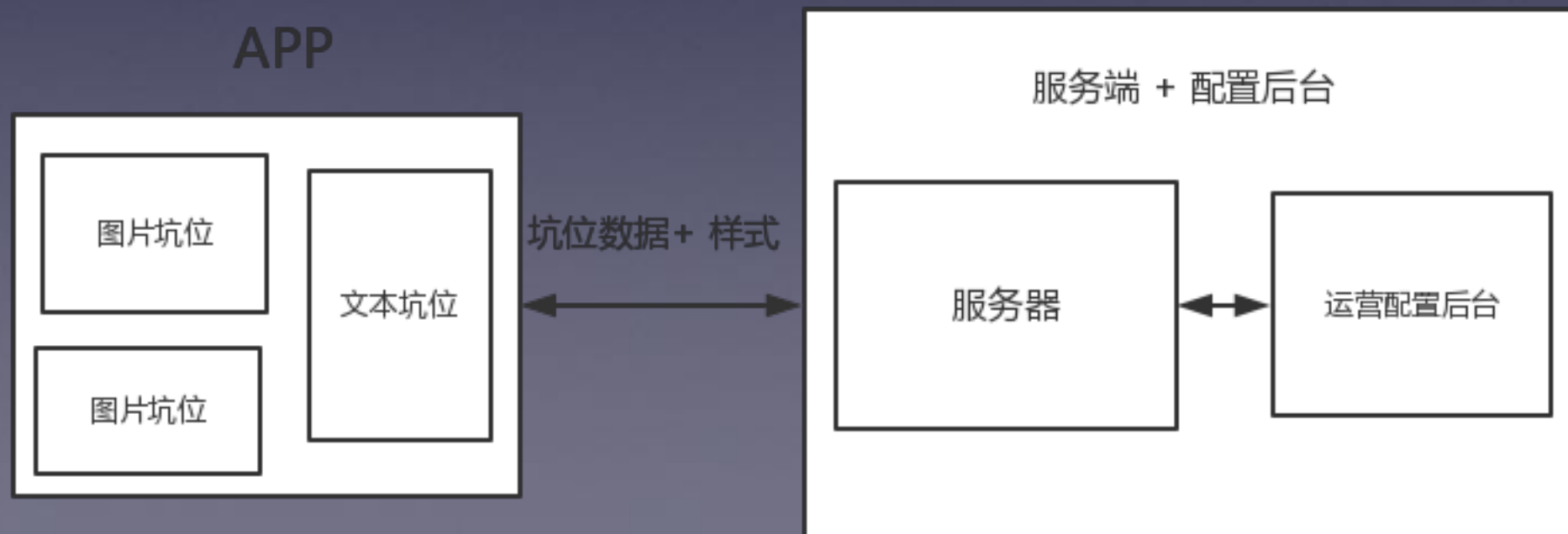


区块开发模式

1、固定模式



2、坑位模式



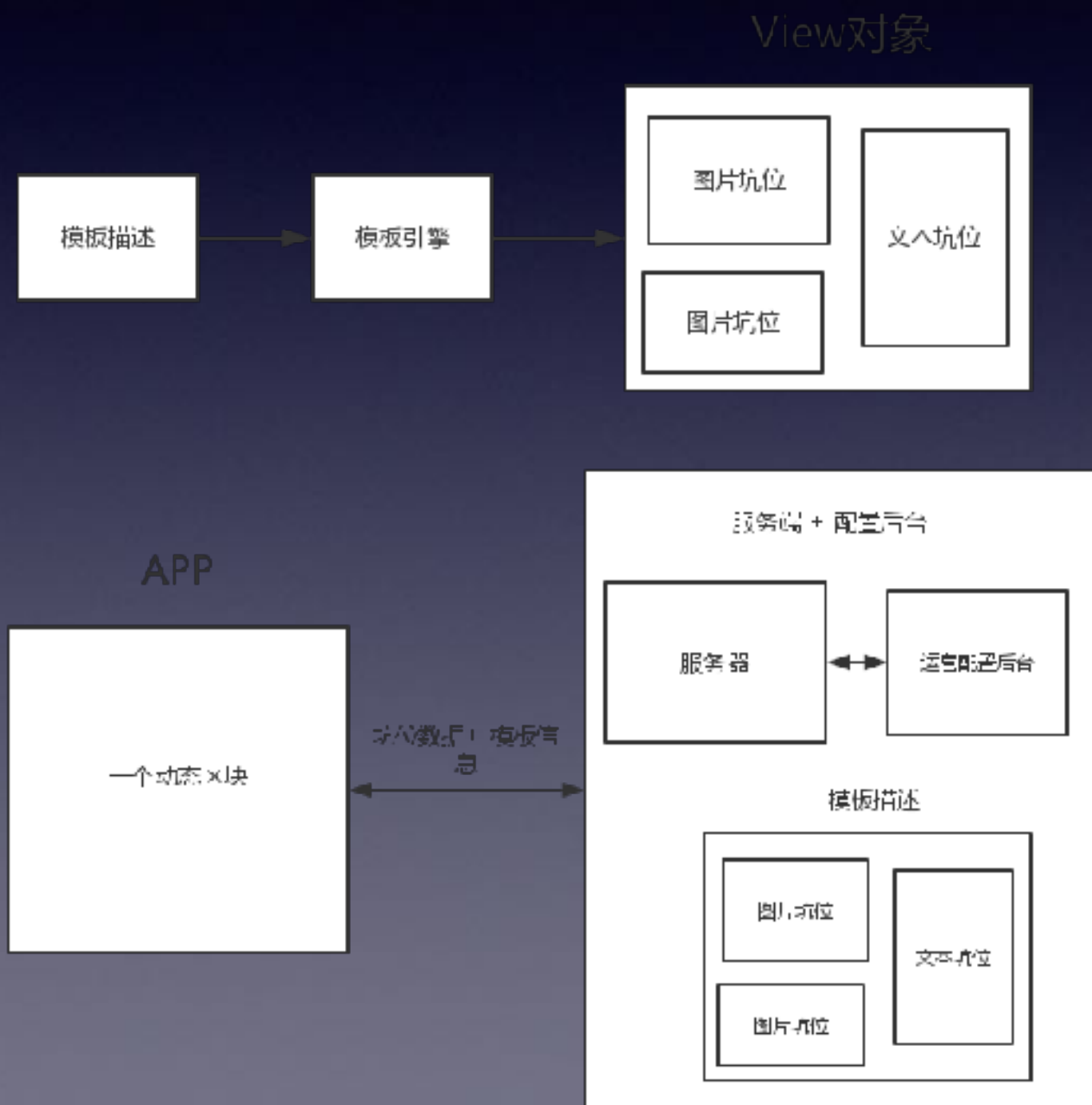
面临的问题

- 1、受限于Native机制，无法动态更新UI布局，无法快速应对运营需求
- 2、书写UI比较费力，缺少实时预览，开发效率低。

现有解决方案

如何应对挑战

UI模板引擎



如何构建一个好UI模板引擎

- 1、简单
- 2、灵活
- 3、强大
- 4、可扩展

UI模板引擎

基本布局

```
<View  
  id="标示符"  
  x="坐标"  
  y="坐标"  
  width="坐标"  
  height="坐标"  
  visible="true|false"  
  background="16进制颜色值,如#FFFFFFF"  
  alpha="透明度"  
  onclick="事件处理函数"/>
```


UI模板引擎

发挥平台布局特性

```
<LinearLayout  
    orientation="vertical|horizontal">  
    <View width="100%" weight="1"/>  
    <View width="100%" weight="1"/>  
</LinearLayout>
```

UI模板引擎UI

高级布局特性 支持100%及calc表达式

```
<View  
  y ="20"  
  width="100%"  
  height="calc(100%-20)"/>
```

UI模板引擎

数据绑定功能

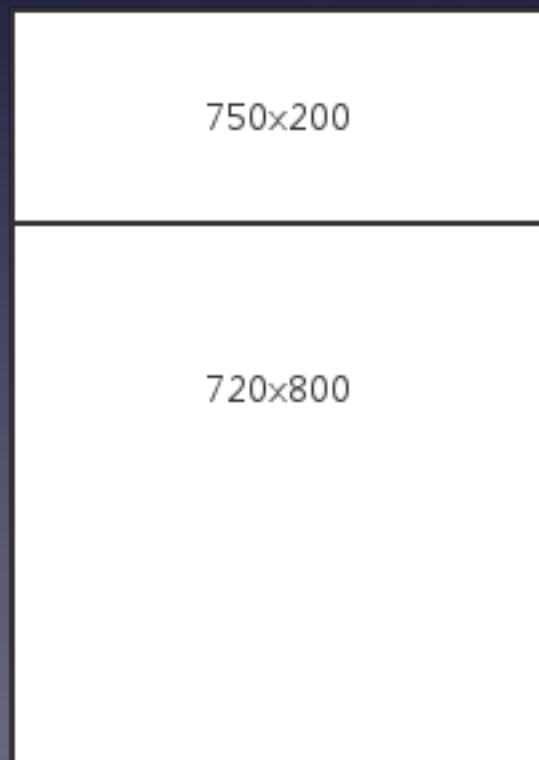
```
<TextView  
    text = “${card[0].desc}”/>
```

```
<ImageView  
    imageUrl = “${card[0].imageUrl}”/>
```

UI模板引擎

多屏幕适配、方便开发、高保真与设计稿统一

设计尺寸



模板书写

```
<View
  height="100%"
  screenUnit="750">
  <View
    width="100%"
    height="200"/>
    <View
      y="200"
      height="calc(100%-200)"/>
    </View>
  </View>
```

