

A decorative graphic on the left side of the slide. It features a large green circle with a white outline, a smaller yellow circle with a white outline, a blue circle, a pink circle, a red circle, and a small light blue circle.

# 基于Jetpack和MVVM的新闻客 户端实战训练营

Allen

2019.11

# 享学讲师团队



**Alvin老师**  
曾就职于三星、  
小米，项目经理



**Leo 老师**  
某创业公司技术总监，  
网易特约讲师



**King 老师**  
曾就职于招行、58同  
城



**Allen老师**  
国防科大研究生毕业，  
全球首批Android开发者



**Zero老师**  
前阿里P7移动架构师，  
曾就职于Nubia等一线  
互联网公司。



**Lance老师**  
某游戏公司主程，  
前爱奇艺高程。

1、大家使用的开发方法

2、Android App的特点、要求和原则

3、APP架构

4、页面架构MVC、MVP、MVVM

5、架构实战

6、Android移动架构师知识体系搭建

# 1、大家使用的开发方法

1

一个模块打天下？

2

一个文件打天下？

3

一个模式打天下？



1、大家使用的开发方法

2、Android App的特点、要求和原则

3、APP架构

4、页面架构MVC、MVP、MVVM

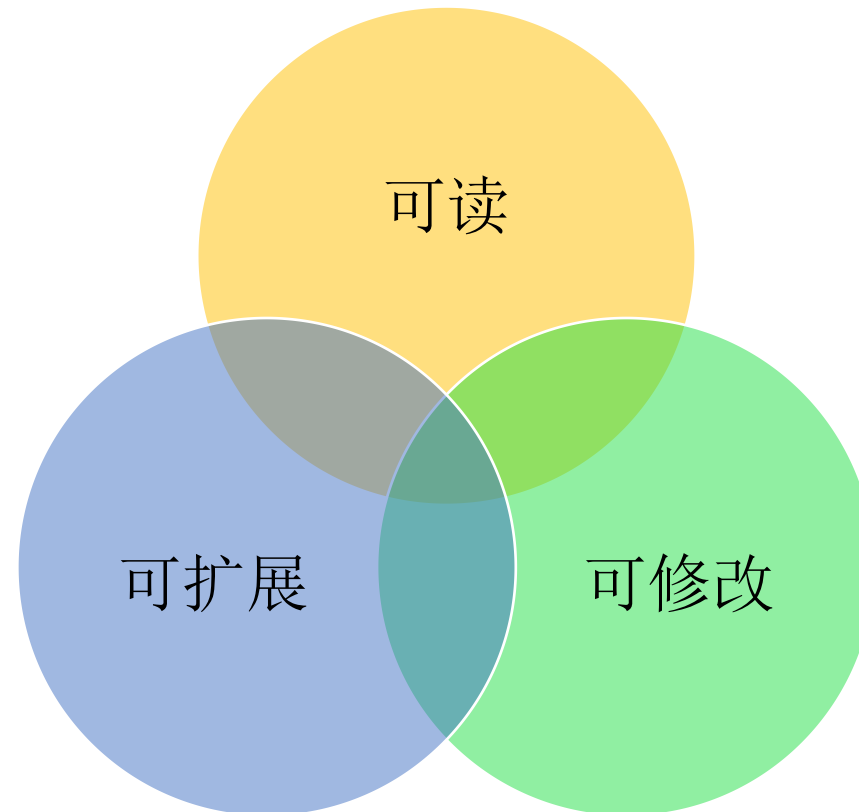
5、架构实战

6、Android移动架构师知识体系搭建

## 2.2、Android应用的特点、要求



直面用户



程序员视角

## 2.3、应用开发原则

### 推荐

- 遵循面向对象的SOLID原则
- 视图、数据、逻辑分离

### 避免

- 认为网络总是顺畅的
- 不要仅仅只考虑能在自己的手机上运行

1、大家使用的开发方法

2、Android App的特点、要求和原则

3、APP架构

4、页面架构MVC、MVP、MVVM

5、架构实战

6、Android移动架构师知识体系搭建



### 3、App架构设计



1、大家使用的开发方法

2、Android App的特点、要求和原则

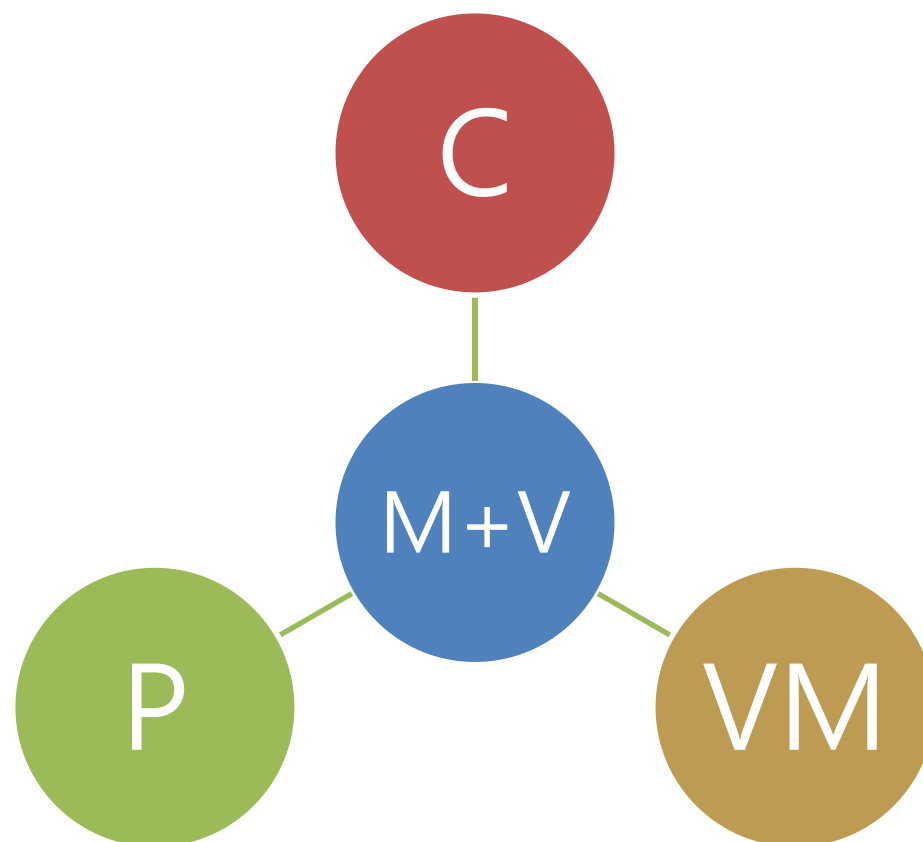
3、APP架构

4、页面架构MVC、MVP、MVVM

5、架构实战

6、Android移动架构师知识体系搭建

## 4、MVC、MVP、MVVM



# 4.1 MVx区别



类型	创建过程	A/F	特点	缺点	应用建议
MVC	C->M + V	C	分离了Model和Controller	Controller变得越来越复杂	简单的、不大修改的页面
MVP	V -> P -> M	V	在MVC的基础上通过Interface彻底分离了View和Model	Presenter与View的交互会琐碎而复杂	核心、复杂、需求变更快页面
MVVM	V -> VM -> M	V	在MVP的基础上增加了Data Binding, 代码量更小	XML中包含代码	核心、复杂、需求变更快页面

1、大家使用的开发方法

2、Android App的特点、要求和原则

3、APP架构

4、页面架构MVC、MVP、MVVM

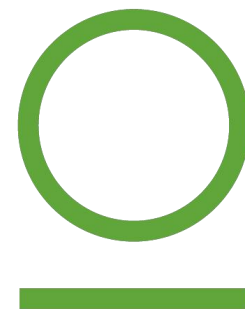
5、架构实战

6、Android移动架构师知识体系搭建

## 5.1 项目初始化

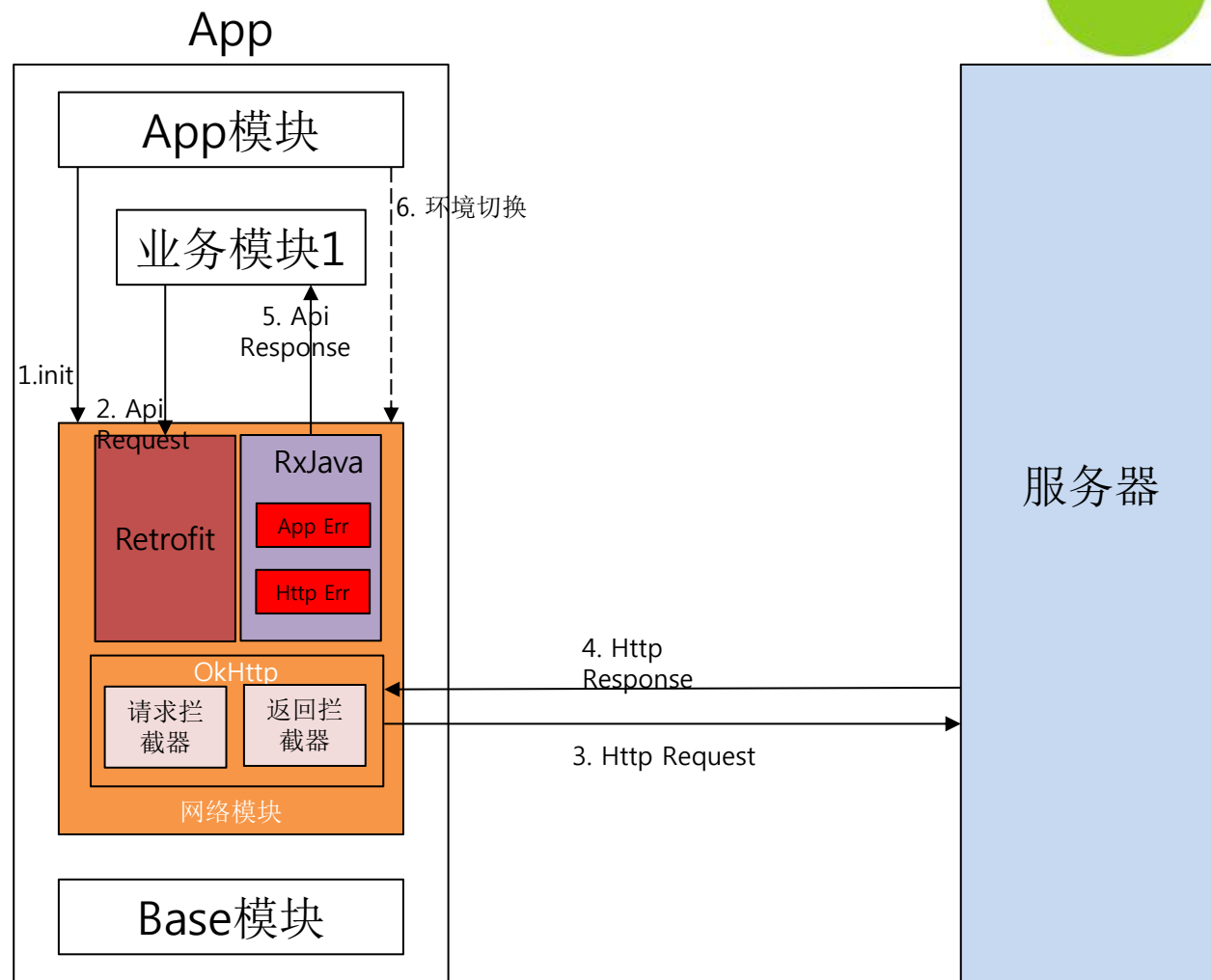
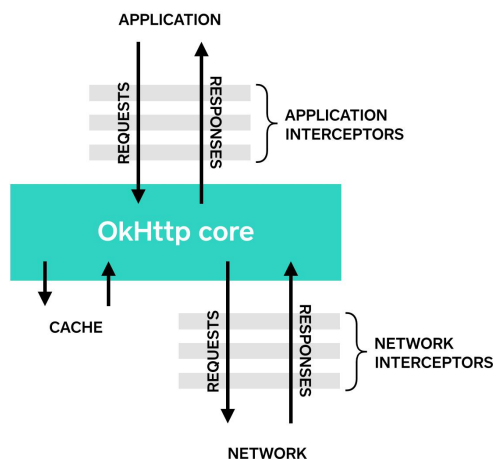
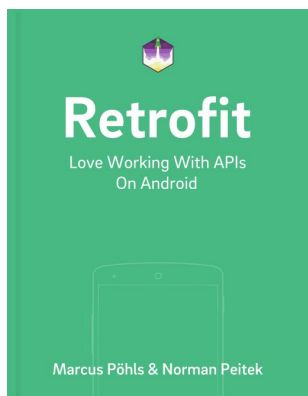


**Tips and Tricks**  
for Android



**JFrog Artifactory**

# 5.2 网络模块架构



## 5.3 腾讯新闻源



<https://market.cloud.tencent.com/products/4798>

全国新闻查询接口

每10分钟刷新。新闻API接口，包括多个频道的最新新闻，有国内焦点、国际焦点、军事焦点、财经焦点、互联网焦点、房产焦点、汽车焦点、体育焦点、娱乐焦点、游戏焦点等等。



# 5.4 自定义view



1、继承View重写onDraw方法	该方法主要用于实现一些不规则的效果，即这种效果不方便通过布局的组合方式来达到，往往需要静态或动态的显示一些不规则的图形，需要通过重写onDraw方法实现，需要提供对wrap_content和padding的支持。
2、继承View的子类 ( TextView,Button )	一般用于扩展已有的控件的功能，比如EditText实现烟花的效果，这种方式不需要手动处理wrap_content和padding支持。
3、继承ViewGroup的子类 ( RelativeLayout,LinearLayout )	比如几个View组合在一起的时候，可以通过该方法实现，采用该方法不需要自己处理测量和布局两个过程。 一般该方法可以实现的，下面的方法也可以实现，这种稍微简单，下面的更接近view的底层。
4、继承ViewGroup	主要用于实现自定义布局，需要处理ViewGroup的测量，布局两个过程，同时需要对子元素的测量和布局做处理。

# 5.5 提取自定义view基类



Type variable

General Definitions

Type parameter

Generic type declaration

■ Parametric polymorphism

Parameterised methods

Type argument

```
public class NonGenericBox<T> {  
    private T object;  
  
    public void set(final T object) {  
        this.object = object;  
    }  
  
    public T get() {  
        return this.object;  
    }  
}  
  
public void useOfNonGenericBox() {  
    final NonGenericBox<String> aNonGenericBox = new NonGenericBox<>();  
    aNonGenericBox.set(new String());  
    final String myString = (String) aNonGenericBox.get();  
    System.out.println(myString);  
}
```

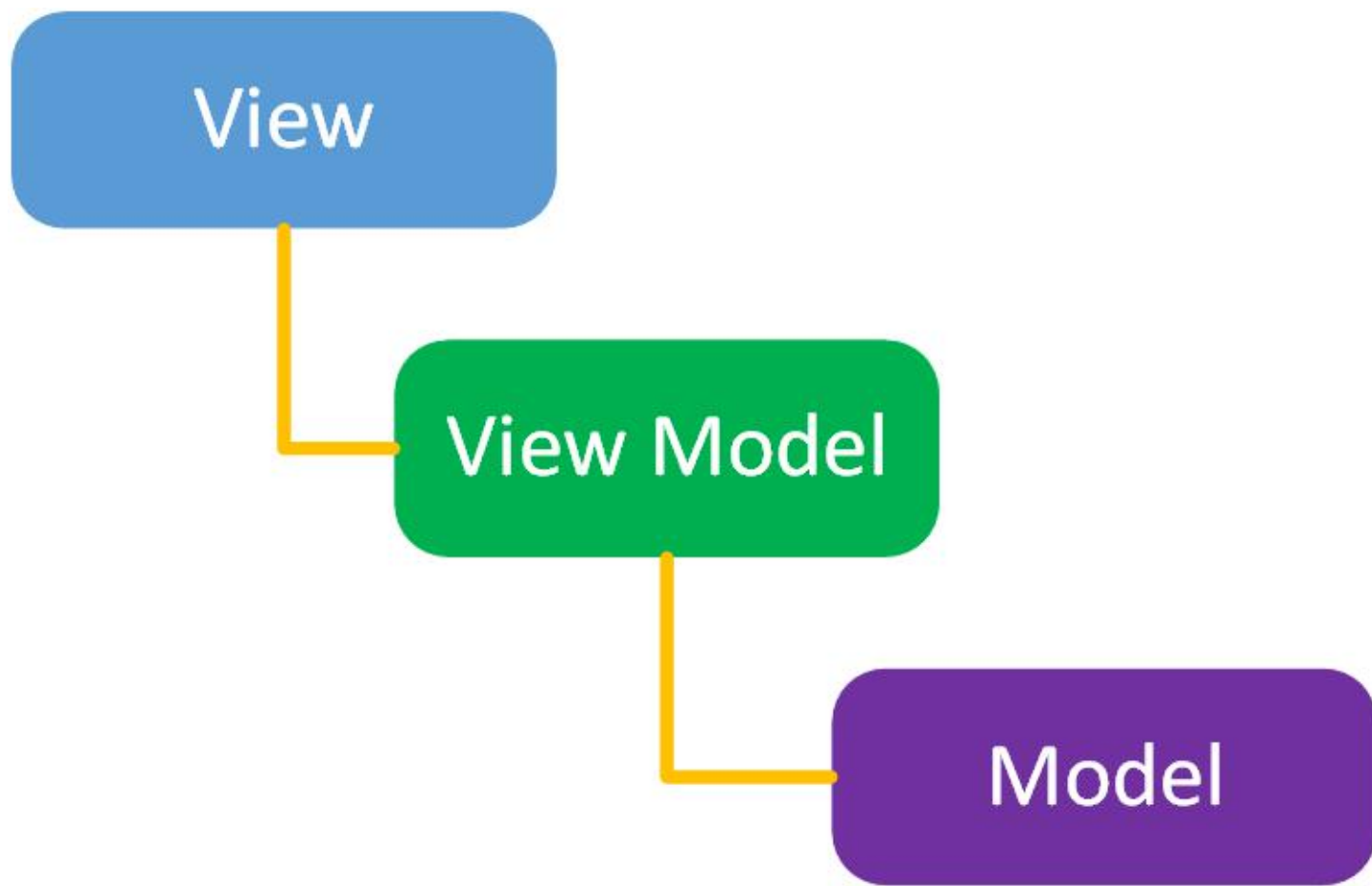


We use Java vocabulary in the following

48/171

泛型类、泛型接口、泛型方法

## 5.6 Mvvm中Model的主要功能

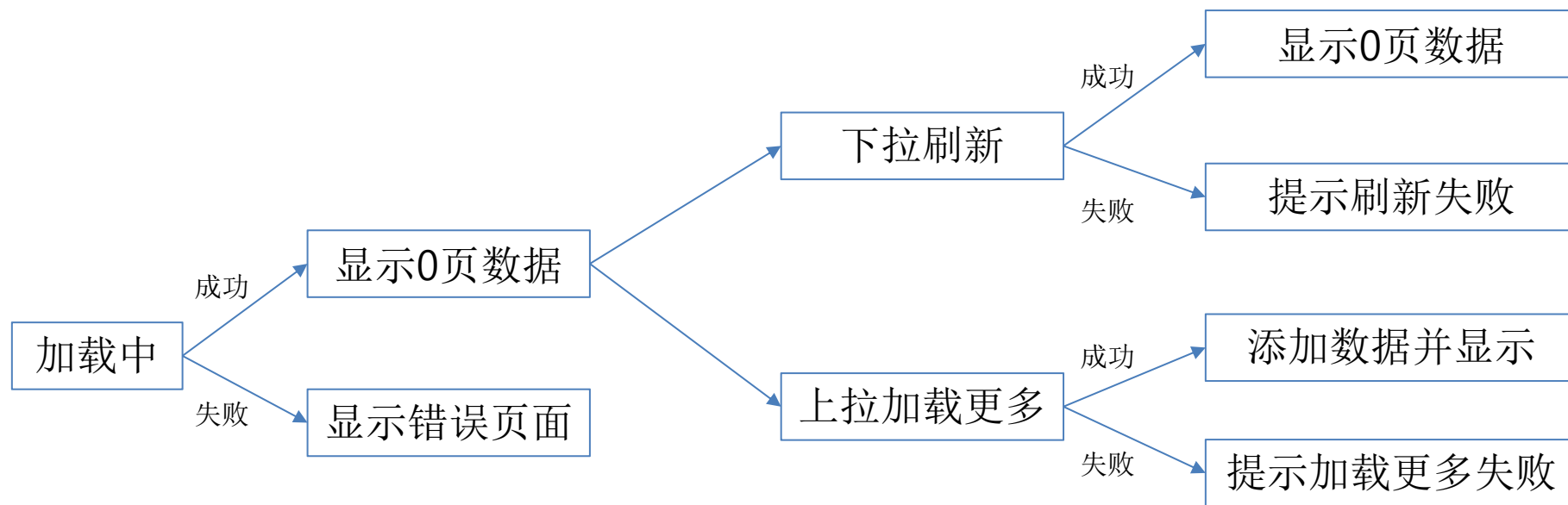


## 5.7 Model需要考虑的问题

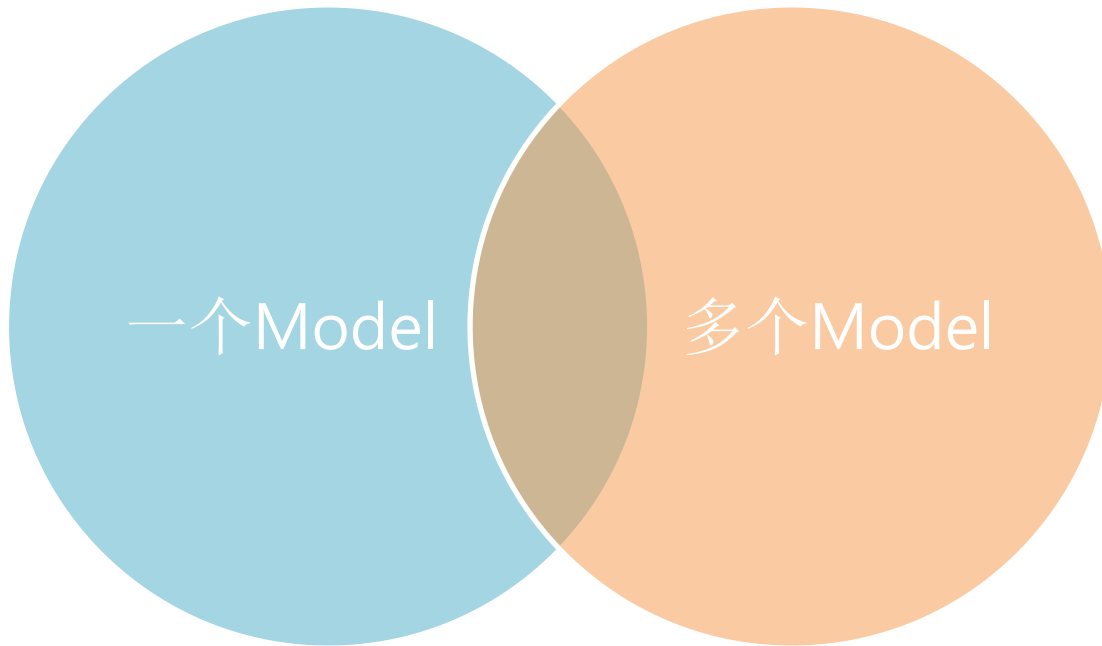


存什么？

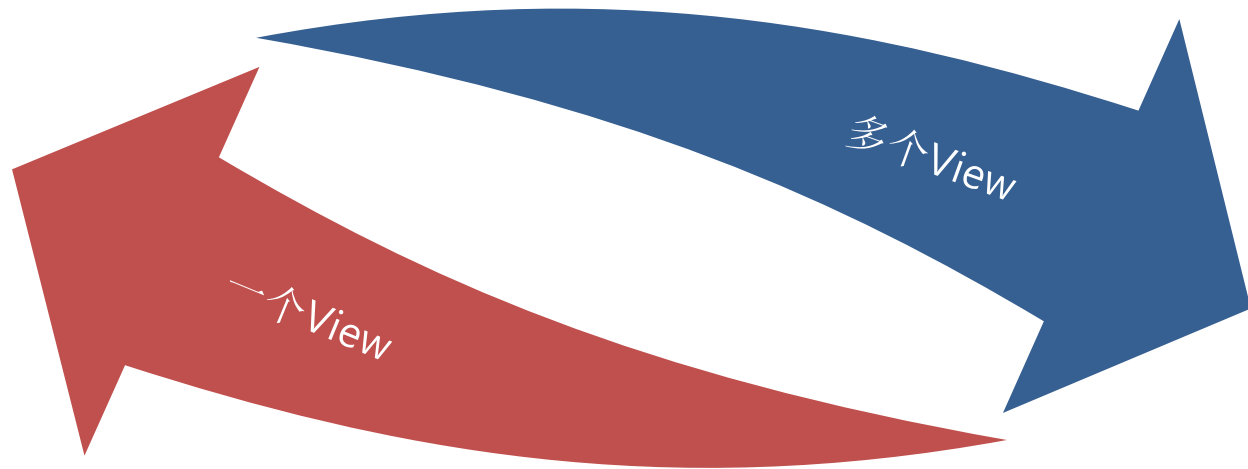
## 5.8 分页状态机



## 5.9 ViewModel怎么实现？



## 5.10 View怎么实现？



- 一个RecyclerView
- RecyclerView的ItemView使用自定义view

# 4.11 错误处理怎么办？



## 每个页面自己布局或者在基类设置布局

基类创建一个布局，然后把子类的布局放置在基类指定的布局



## 统一处理

错误页面统一在基类处理完成，派生类只需要设置替换的View即可



## 4.12 最后的但不是最不重要的



1、大家使用的开发方法

2、Android App的特点、要求和原则

3、APP架构

4、页面架构MVC、MVP、MVVM

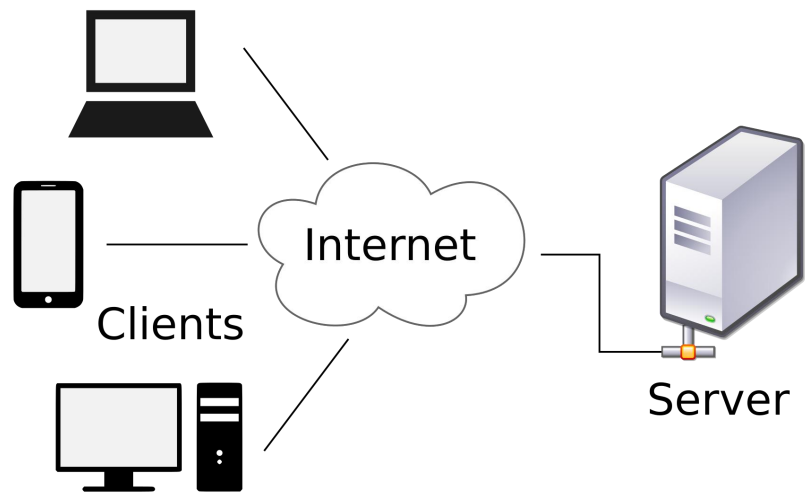
5、架构实战

6、Android移动架构师知识体系搭建

# 6.1 架构师



## 6.2 移动架构师



客户端/服务器架构

谢谢大家

