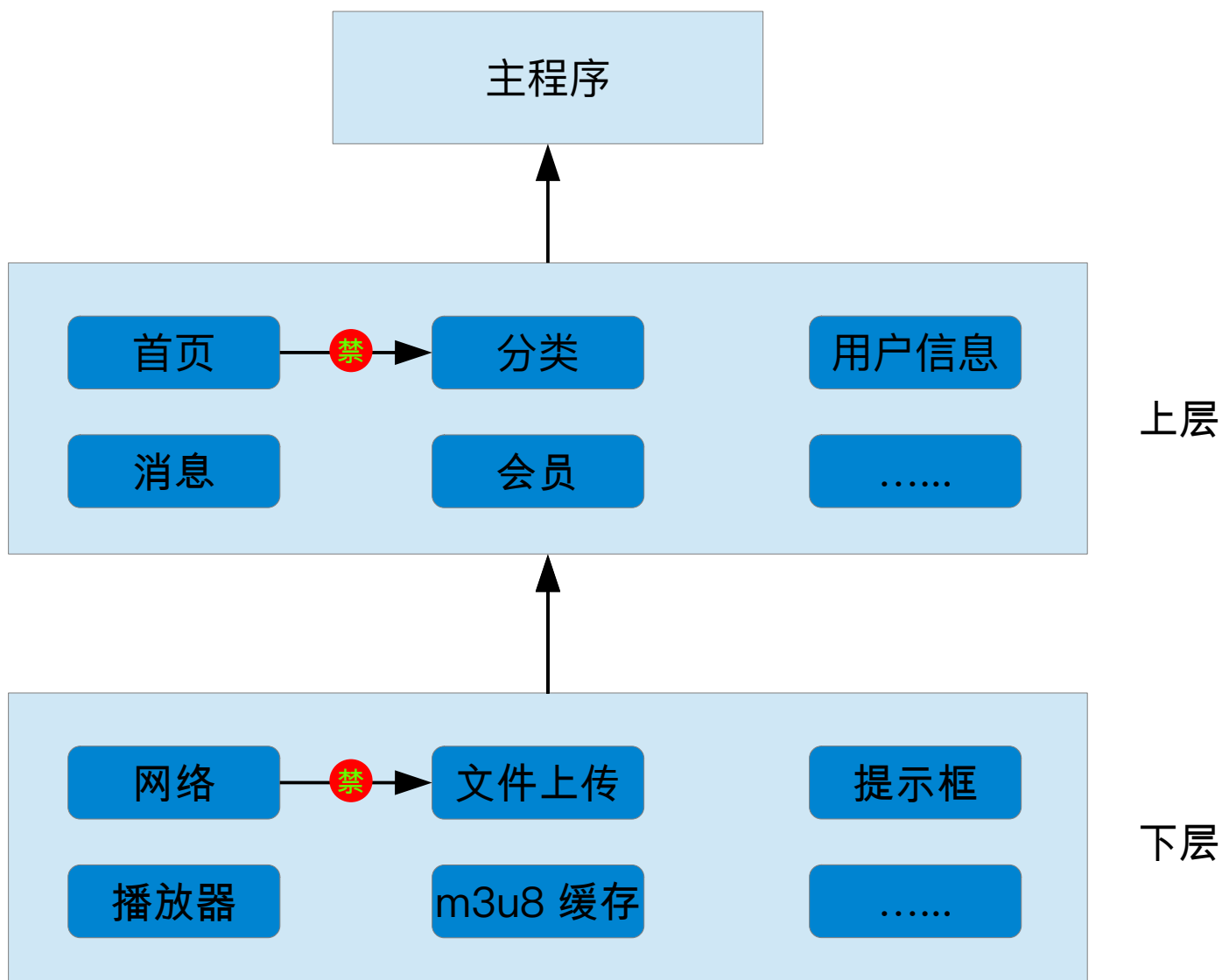


APP 架构



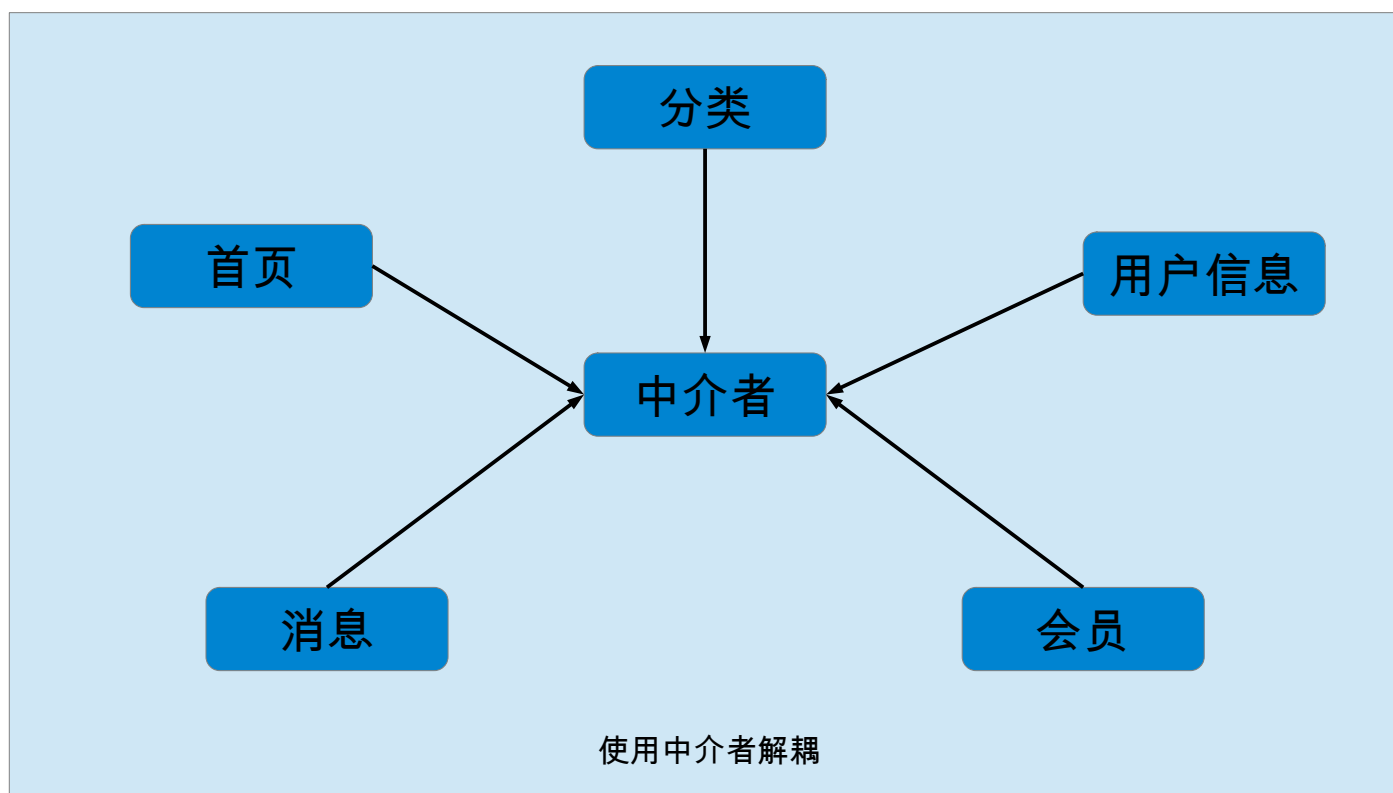
app 典型架构

1. 下层为上层提供服务；下层不可以调用上层。
2. 同层级模块之间不能相互引用。
3. 允许跨层访问。

模块间通信

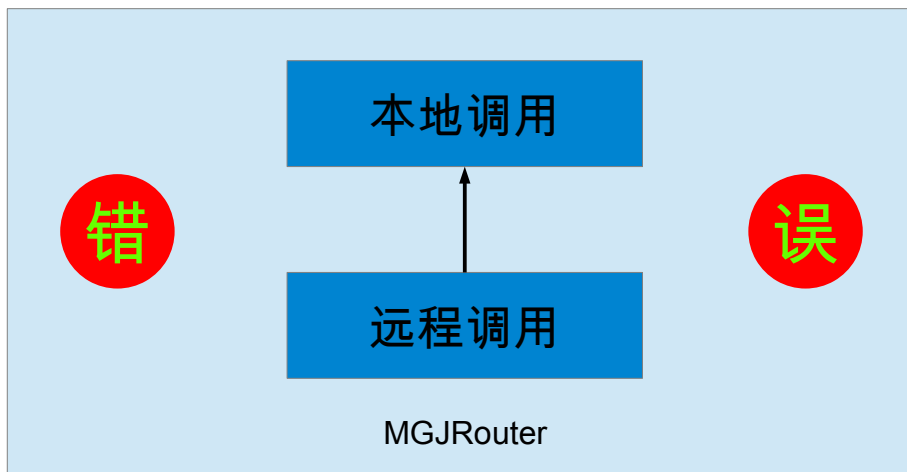


上层业务模块间难免会出现模块间相互调用，比如首页控制器 HomeVC 跳转到分类控制器 CategoryVC，如果直接在 HomeVC 中直接引入 CategoryVC，模块就耦合到一起了，这时候所做的就是解藕。

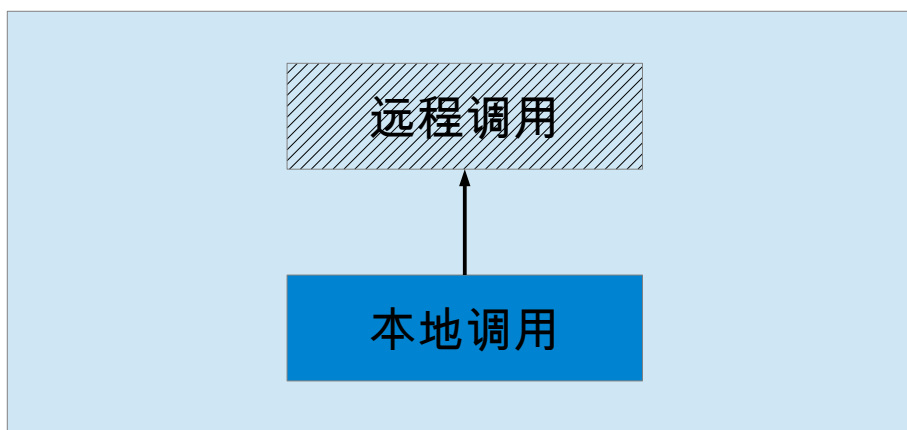


组件化

提到组件化，很多人就会想到 URL 方式实现；例如 JRRoutes、MGJRouter。
但 URL 方式不是组件化的必须方式！请记住：我们的目的是解耦。



蘑菇街有个思想上的错误：远程调用为本地调用提供服务。



正确的做法是：本地调用为远程调用提供服务。
但我们的目的是为了解耦，本地调用已经能够解耦，远程调用就没必要有了。

组件化解耦：XSCCommand

1. 代码量极少，核心代码 68 行；总代码 104 行。
2. 采用命令模式设计，面向对象
3. 支持自定义初始化方法，支持 block、delegate 回调