

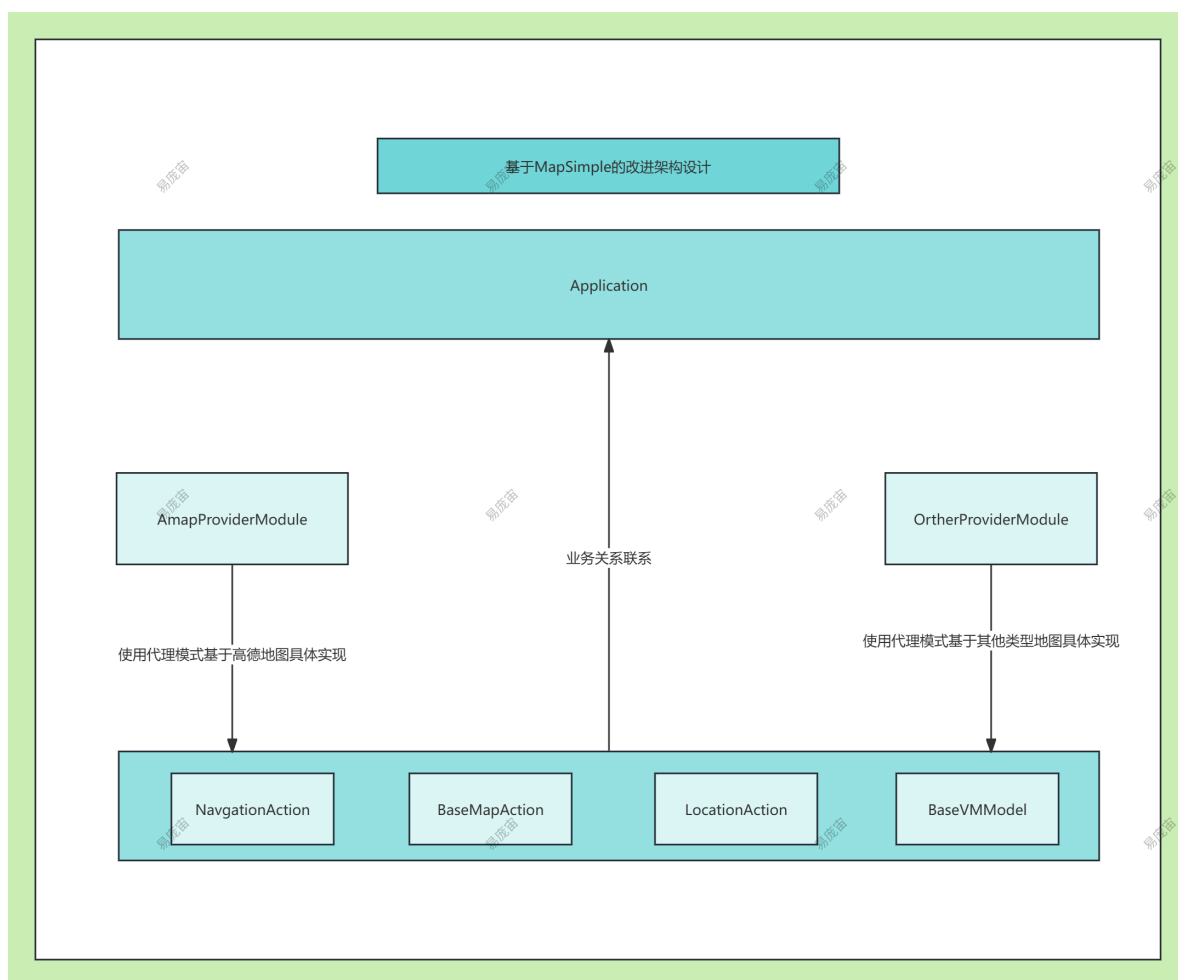
# MapSimple 编码思路及实现

## 抽离具体抽象业务逻辑 及 具体实现设计

1. **IMapAction**: 负责业务包含简单地图操作抽象接口
2. **IMapFragment**: 负责继承 **IMapAction** 及 **Fragment** 看做抽象层和图层 UI 的抽象集合体
3. **AMapFragment**: 基于高德地图 SDK 及实现 **IMapFragment** 具体业务动作图层 UI 对象
4. **NavHelperAction**: 抽离高德地图 **sdk** 导航相关动作的具体实现类
5. **InstanceMapActionVM**: 贯穿整个 **App** 的 **VM model**, 通过这个 **model** 感知具体的数据变化进行相应的 **UI** 控制及部分逻辑控制, 解耦部分模块耦合

## 改进优化

如果需要运用这个项目 **simple** 作为商用技术或者变为整体通用技术仓库或者模块可以尝试使用一下设计思路:



## 地图抽象层设计模块 及 代理实现层

- **BaseMapAction**: 地图图层常用操作动作接口
- **NavigationAction**: 地图导航抽象动作接口
- **LocationAction**: 定位用户或者地图定位抽象接口
- **BaseVMModel**: 地图抽象层设计模块基准数据 VM 模块

基于上述4个抽象或加上其他业务抽象动作设计为一个通用业务型地图 **BaseMapActionModule**，例如基于 高德地图SDK 及 **BaseMapActionModule** 实现为 **AMapProviderModule** 通过引入 **BaseMapActionModule**、**AMapProviderModule**，编写开发过程中声明使用 **BaseMapActionModule** 实际上运用为 **AMapProviderModule** 提供具体实现。当后续需要更换地图框架 SDK 可通过重新实现新 **MapProviderModule** 去更换旧的 **MapProviderModule**，从而避免上层 App UI层面出现重大的变化。