何为优秀的架构

1. 对象拥有特定明确的角色
2. 简单的数据流
3. 不依赖某个架构或者服务
4. 足够灵活容易理解
5. 可测试性

名词解析

TDD方法

DRY 原则

#### 组合设计模式

#### iOS流行的架构模式

1. MVC
2. MVVM
3. Viper

MVC

开始MVC是适用于web开发的开发模式，但是iOS高度耦合性的viewcontroller使 view 和controller紧紧结合在一起，并不能单独使用controller。更谈不上复用，测试也非常麻烦这是mvc的模式的弊端。

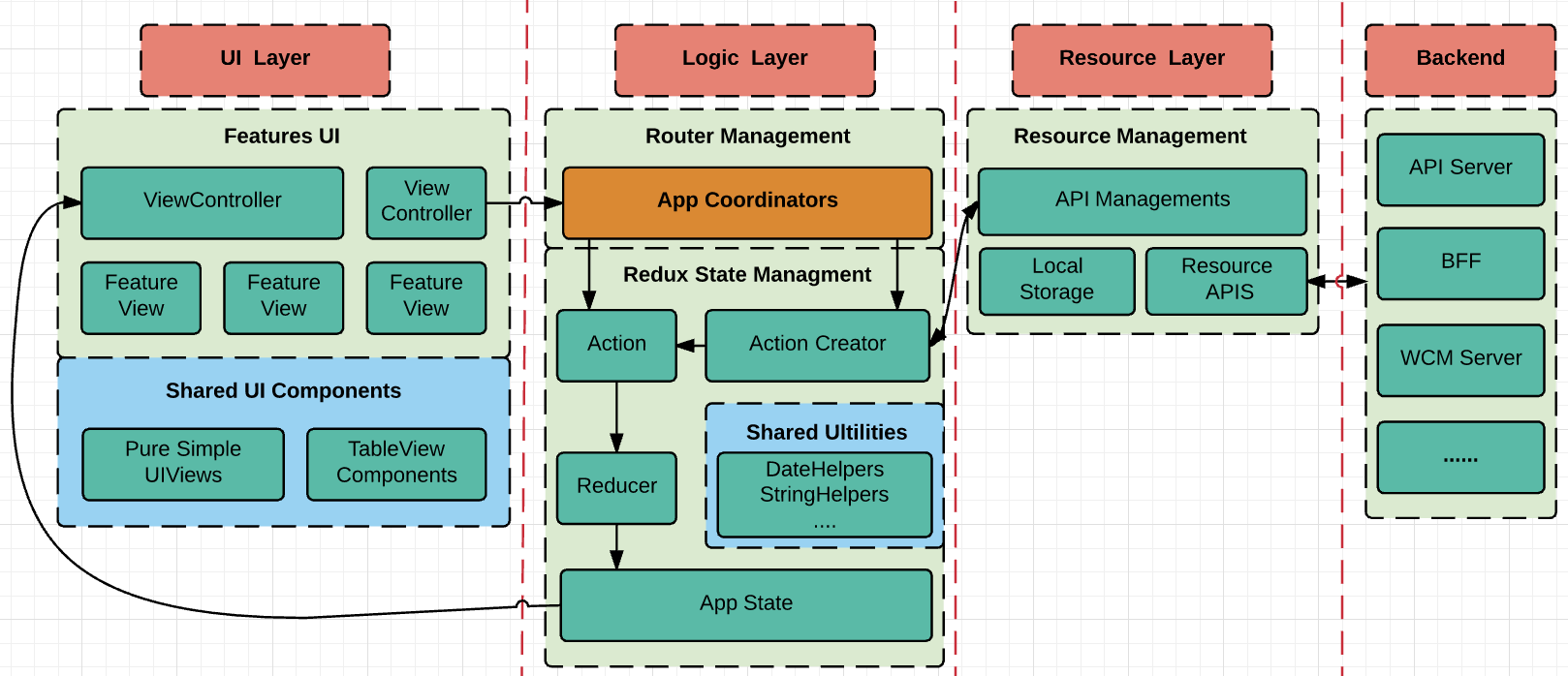
Viper

* 交互器 (interactor) 负责业务逻辑。
* 路由 (router)，它允许您为视图和视图控制器配置不同的展示上下文，以便您能够在多个地方当中展示相同的 UI
* 展示器 (presenter) 是一个包含 UIKit 独立展示逻辑的类。它本身不直接与 UIKit 进行交互，但是它会格式化链接，并执行类似的操作。

ReSwift

* **App State**: 在一个时间点上，应用的所有状态. 只要App State一样，应用的展现就是一样的。
* **Store:** 保存App State的对象，其还负责发送Action更新App State.
* **Action:** 表示一次改变应用状态的行为，其本身可以携带用以改变App State的数据。
* **Reducer:** 一个接收当前App State和Action，返回新的App State的小函数。

iOS 结构图



跳转接口

[JLRoutes](https://github.com/joeldev/JLRoutes)，[routable-ios](https://github.com/clayallsopp/routable-ios)，[MGJRouter](https://github.com/meili/MGJRouter)

