

泛娱乐 iOS 架构图

- 1. 架构分为 UI 层，中间层，能力层
- 2. 能力层，提供基本能力，如 http，WS，hummer，thunder等
- 3. 中间层，net handle 检测网络情况，如 net check 网络状态，hummer 网络状态，thunder 网络状态，WS 网络状态等，统一在 net handle 中处理，然后通过注册的后，有问题就通知，冒泡回调，如果返回YES，就不回调了，按照优先级回调
- 4. 中间层，task bus 用于数据同步，从能力层获取的回调通知统一走消息总线，向上抛回调，在主线程回调；设置拦截器，用于判断是否要接收请求
- 5. UI 层，Login 是用于处理登录请求，包括 http 登录 和 hummer 登录，现在只有这两个。这个模块属于全局的
- 6. UI 层，MVCP 模式，model 用于数据处理；view 视图；presenter 发送 http 消息，hummer 动作等，所有与外界的联系都是 presenter 负责；Controller 处理业务逻辑，并与 view，model，presenter 互通交互；ControllerLifeLine，ViewController 的生命周期管理，集成于 ViewController，管理整个生命周期，Controller 继承 ControllerLifeLine；UserList 单独的模块，用于用户管理，现在是在房间内使用，可能不会用到。
- 7. 对于 LiveBG + LiveBGView 的处理，LiveBG 通过处理消息，把消息进行上抛回调给 delegate，上层接收者统一处理，LiveBGView 由上层管理者管理
- 8. 提示框统一管理，并设置是否超时时间
- 9. http 请求，下一步统一使用 block，不要在 success block 中直接请求下一步，有序请求，可以使用 PromiseKit
- 10. 日志问题，一定要有，要打印用户行为，并且不同 view 要打 不同 tag，能够快速定位问题

