组件化介绍及实践

- 组件化简介
- 术语
- 业务拆分及调用
- 示例代码

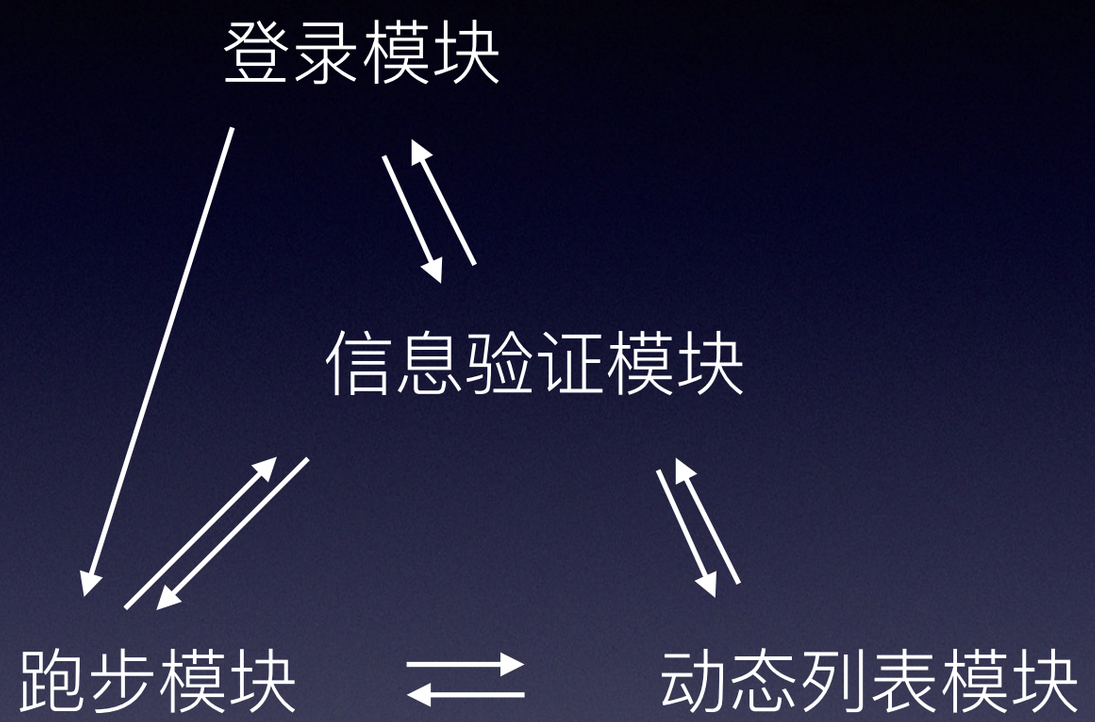
组件化简介

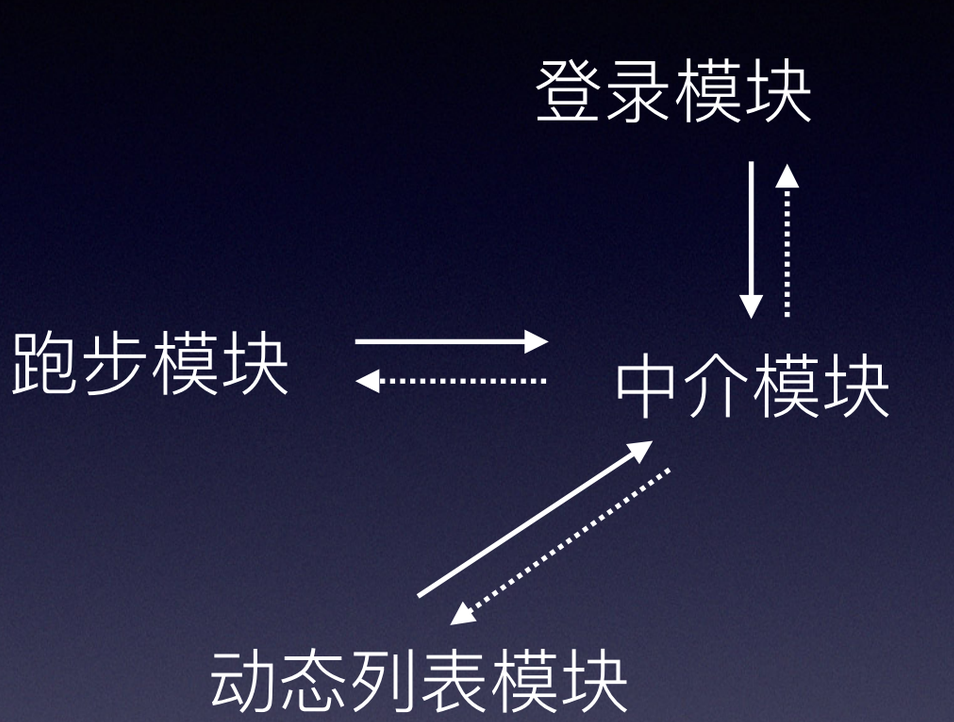
- 组件的划分
- 组件化的需求来源
- 组件化方案

组件的划分

- 代码模块根据业务相关性划分
  - 独立业务模块 ⟶ 存在相互之间的引用，处理业务
    - 登录模块、信息验证模块、跑步模块、动态列表模块、话题详情模块、私信模块
  - 非业务模块 ⟶ 与业务模块之间单向引用
    - 网络模块、数据库模块、照片获取及处理模块

组件化的需求来源





组件化前

代码耦合度高

相互依赖严重

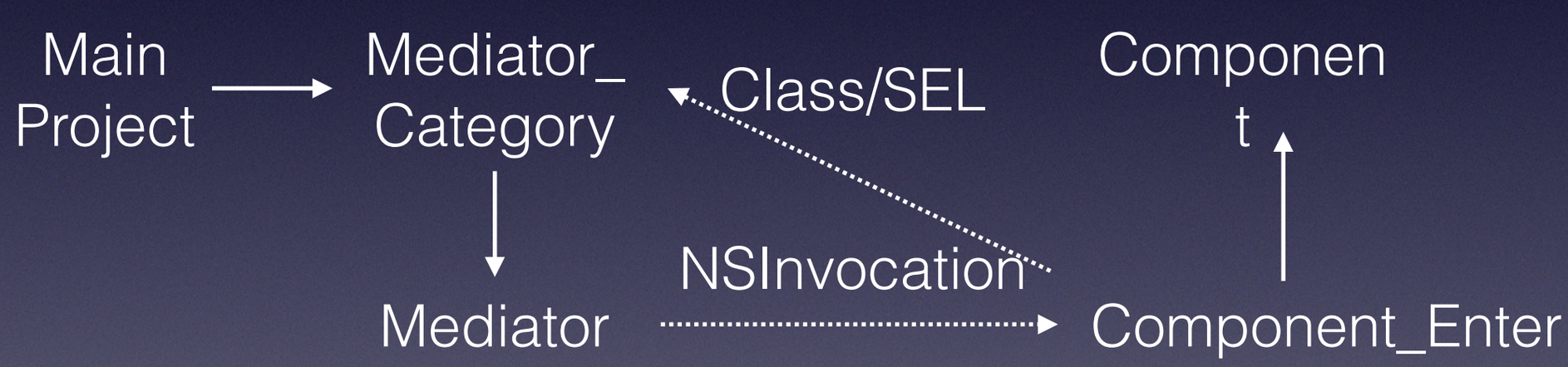
组件化后

代码解耦

分工明确

组件化方案

- 解除中介模块对其他模块的依赖
    - Runtime + Category



术语

- podspec
    - 组织创建一个Pod库所需的资源、代码、需引用的framework、第三方库等的文件
    - 证书：s.license = { :type => "MIT", :file => "FILE_LICENSE" }
    - 代码托管：s.source = { :git => "https://github.com/Suncoldx333/Component_A_Category.git", :tag => s.version.to_s }
    - 代码：s.source_files = “Component_A_Category/Little_A_Category/**/*.{h,m}"
    - 第三方：s.dependency “SDWebImage"
- 私有sepc库
    - 用于存放共享库的podspec文件的仓库

业务组件拆分

- 新建XCode项目，将原有项目的待拆分业务组件代码转移至该项目
- 建立私有Pod库，存放该项目
- pod spec create [项目名]，生成podspec文件，根据实际代码修改s.license、s.source、s.source_files、s.dependency等字段
- pod lib lint [项目名].podspec 进行验证
- pod repo push [私有sepc库] [项目名].podspec，将该项目的podspec文件加入私有sepc库

业务组件调用

- 业务组件拆分之后，可通过cocoapods调用，与调用常规第三方一致，需添加私有spec库的git地址（此案例使用cocoapods共有库）

```
source 'https://github.com/Suncoldx333/HelloSpec.git'
source 'https://github.com/CocoaPods/Specs.git'

target 'MainProject' do

  pod 'Component_A_Category'
  pod 'Component_A','~> 5'
  pod 'Texture','~>2.6'
end
```

- 示例项目：https://github.com/SuncoIdx333/ProjectAfterComponent.git

    - 中介模块使用CTMediator:https://github.com/casatwy/CTMediator.git

- 调用业务组件的方法必须带入参，因为CTMediator在使用NSInvocation调用的方法名经合成后，会添加一个冒号，如果不带入参，则[[[Class alloc] init] respondsToSelector:SEL],会返回NO。