

保障 API 优雅与稳定的多平台实践

岑裕

融云 - 工程效率首席架构师

极客邦科技 会议推荐2019

ArchSummit 深圳

全球架构师峰会

大会：7月12-13日
培训：7月14-15日

ArchSummit 北京

全球架构师峰会

大会：12月6-7日
培训：12月8-9日

5月

QCon 北京

全球软件开发大会

大会：5月6-8日
培训：5月9-10日

QCon 广州

全球软件开发大会

培训：5月25-26日
大会：5月27-28日

6月

GTLC
GLOBAL
TECH LEADERSHIP
CONFERENCE

上海

技术领导力峰会

时间：6月14-15日

GMTC 北京

全球大前端技术大会

大会：6月20-21日
培训：6月22-23日

7月

QCon 上海

全球软件开发大会

大会：10月17-19日
培训：10月20-21日

10月

GMTC 深圳

全球大前端技术大会

大会：11月8-9日
培训：11月10-11日

AiCon 北京

全球人工智能与机器学习大会

大会：11月21-22日
培训：11月23-24日

11月

12月

自我介绍

岑裕

在融云担任工程效率首席架构师、音视频核心技术负责人

工作集中在 IM、音视频与工程效率

对 0 → 1 过程中质量与效率的提升比较有兴趣

之前，曾在三星通信研究院从事 Android 系统及 RCS 融合通信 framework 研发



目录

- 重要性与愿景
- 问题与挑战
- API 的范围
- 实践讨论
- 演进示例

API 的重要性

计算机领域有句名言 ——

计算机科学领域的任何问题都可以通过增加一个间接的中间层来解决。

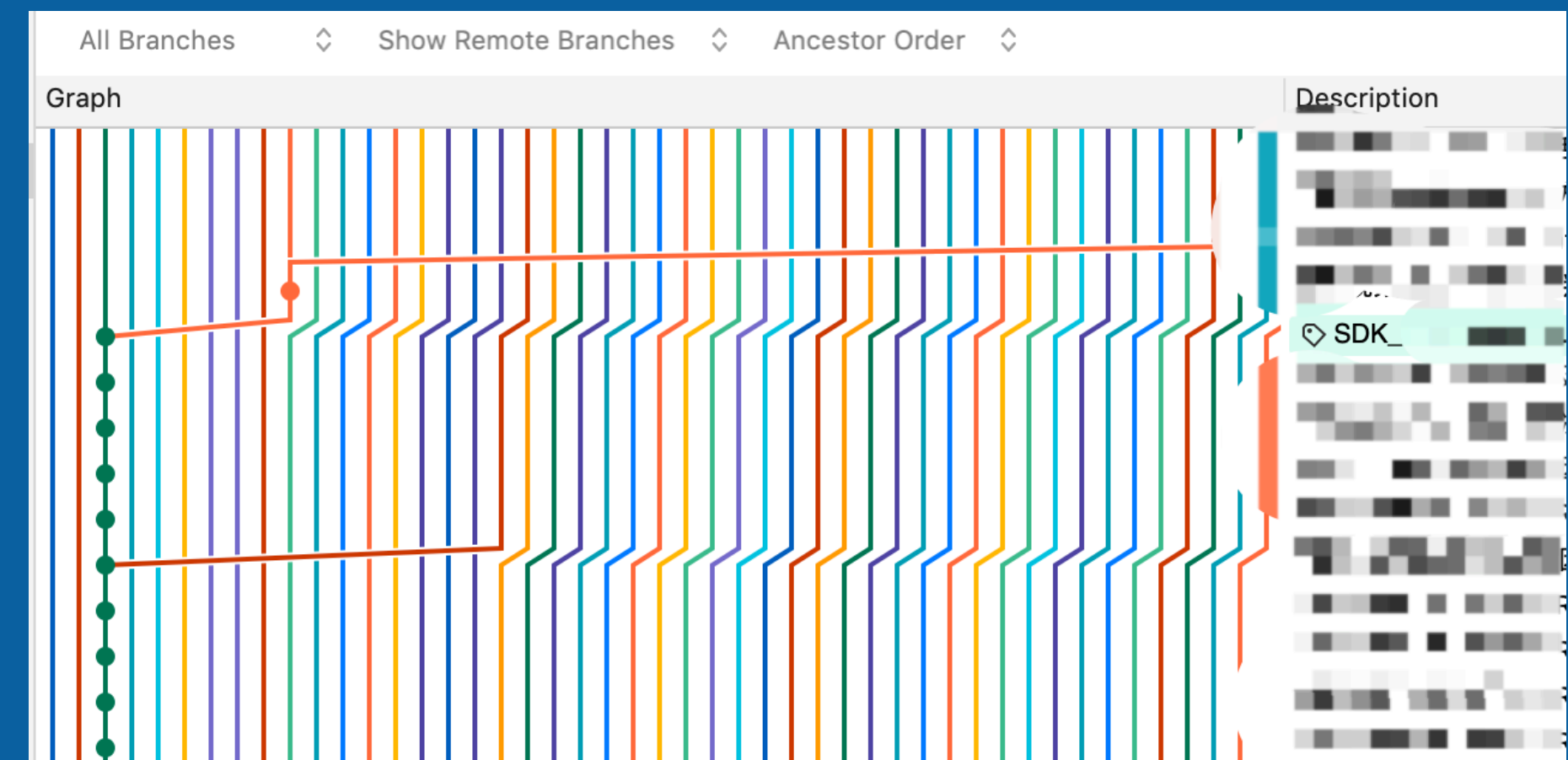
- 积木式编程，扩宽业务范围
- 降低交流成本，身心愉悦

愿景

- 优雅
- 简单
- 健壮
- 稳定

问题与挑战

- 用户群与使用方式不明确
- 不同用户的需求不一致，功能与流程不断变化
- 多种平台风格不统一，变更不可预见
- 历史包袱与向后兼容



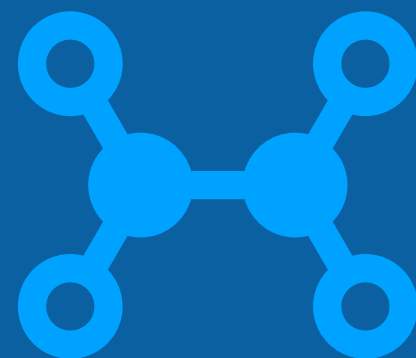
API 的范围

- 如何被使用？
协议调用(HTTP/RPC)，语言调用(Native SDK)
- 如何运行？
配置文件，环境变量，加载库，内存，线程
- 谁在使用？
LSUD(Larget Set of Unkown Developers)
SSKD(Small Set of Kown Developers)
- 相关方都有谁？
人，需要可理解性 文档/Demo/工具/Log
机器，需要 向后兼容

实践讨论：明确的 Workflow



范围定义



设计



编码



Review



发布



反馈



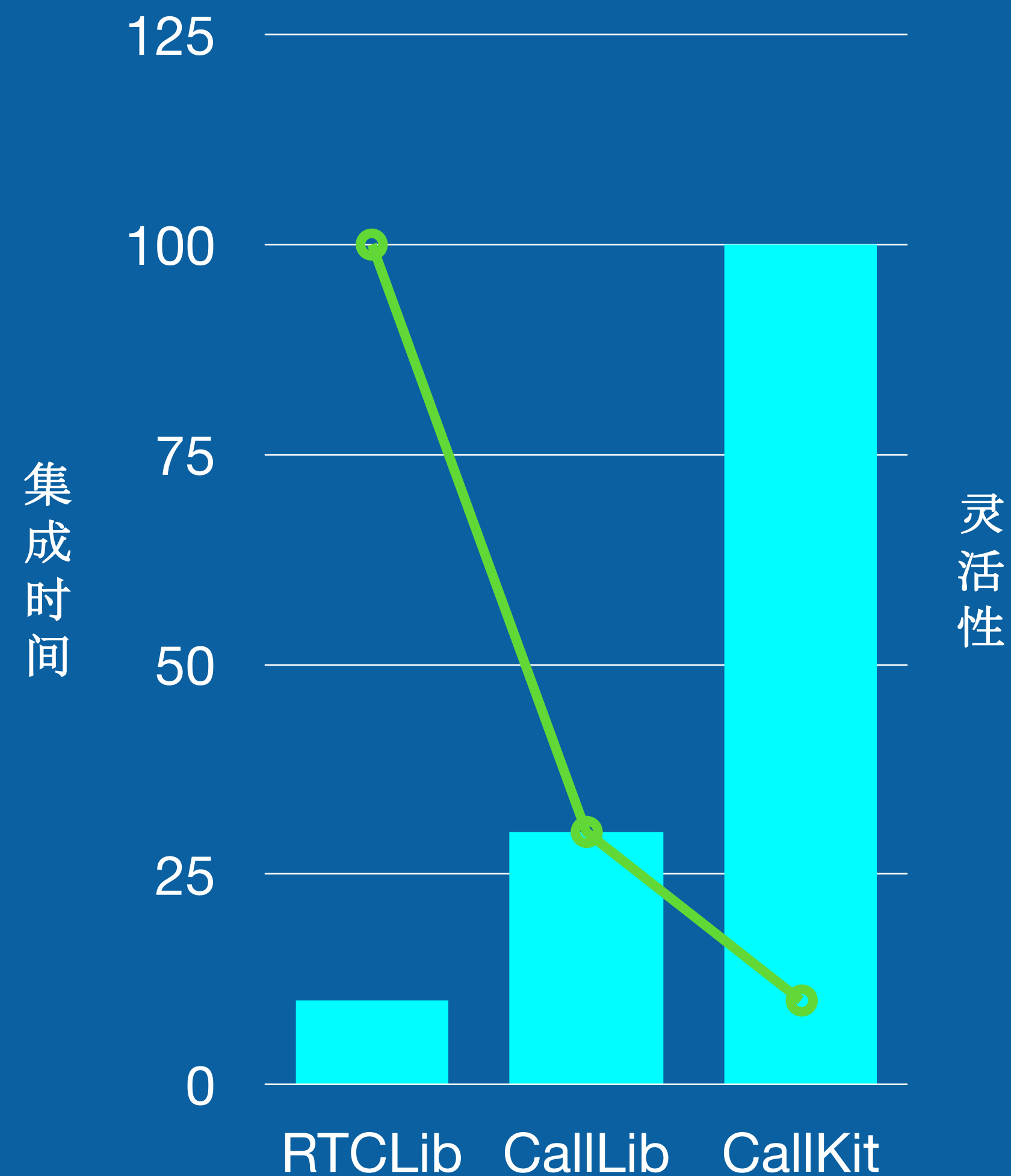
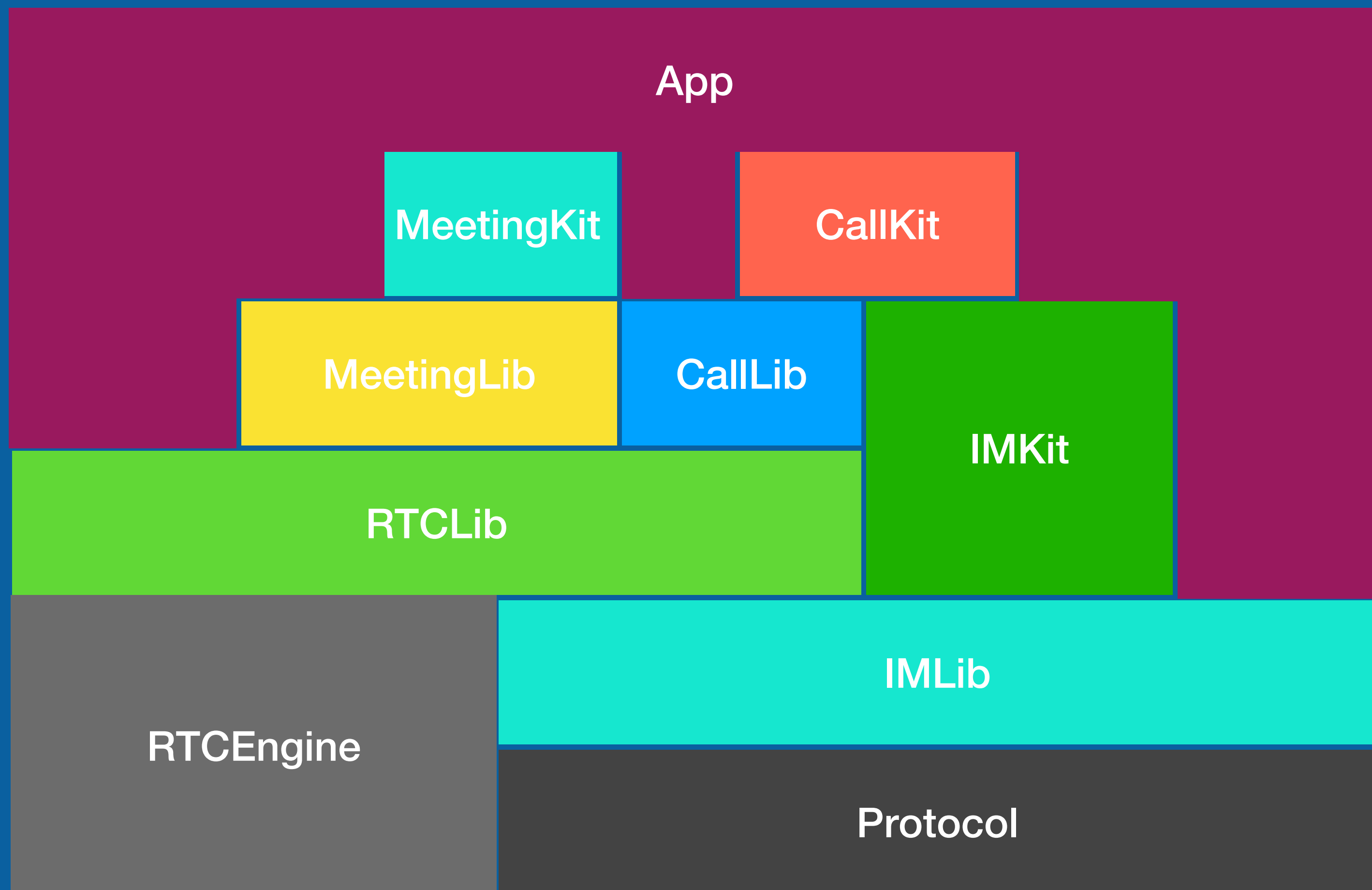
演进

- 量化指标和使用工具提升稳定性
- 避免设计压力集中于 1~2 个人

实践讨论： 范围定义

- 明确服务对象
- 明确用户的使用场景
- Eating your own dog food
- 针对不同需求可以采用分层 API 模式

范围定义：分层 API 模式



实践讨论：优雅的设计

- 简单：一个 API 只做一件事情
- 先抽象模型，再进行多平台设计
- 平台风格 → 语言风格 → 业务抽象 → 团队风格
- 命名细节：时态、单复数、介词、大小写 等等
- 经验式编程：所见即所得

```
func move(from start: Point, to end: Point)  
x.move(from: x, to: y)
```

```
a.move(toX: b, y: c)  
a.fade(fromRed: b, green: c, blue: d)
```

```
a.moveTo(x: b, y: c)  
a.fadeFrom(red: b, green: c, blue: d)
```

优雅的设计：经验式编程

- 免注释式命名
- 行为自解释
- 区分基础与高级 API
- 避免副作用

```
- (void)startSingleCall:(NSString *)targetId mediaType:(RCCallMediaType)mediaType;  
  
- (void)startMultiCall:(RCConversationType)conversationType  
    targetId:(NSString *)targetId  
    mediaType:(RCCallMediaType)mediaType;
```

```
- (RCCallSession *)startCall:(RCConversationType)conversationType  
    targetId:(NSString *)targetId  
    to:(NSArray *)userIds  
    mediaType:(RCCallMediaType)type  
    sessionDelegate:(id<RCCallSessionDelegate>)delegate  
    extra:(NSString *)extra;
```

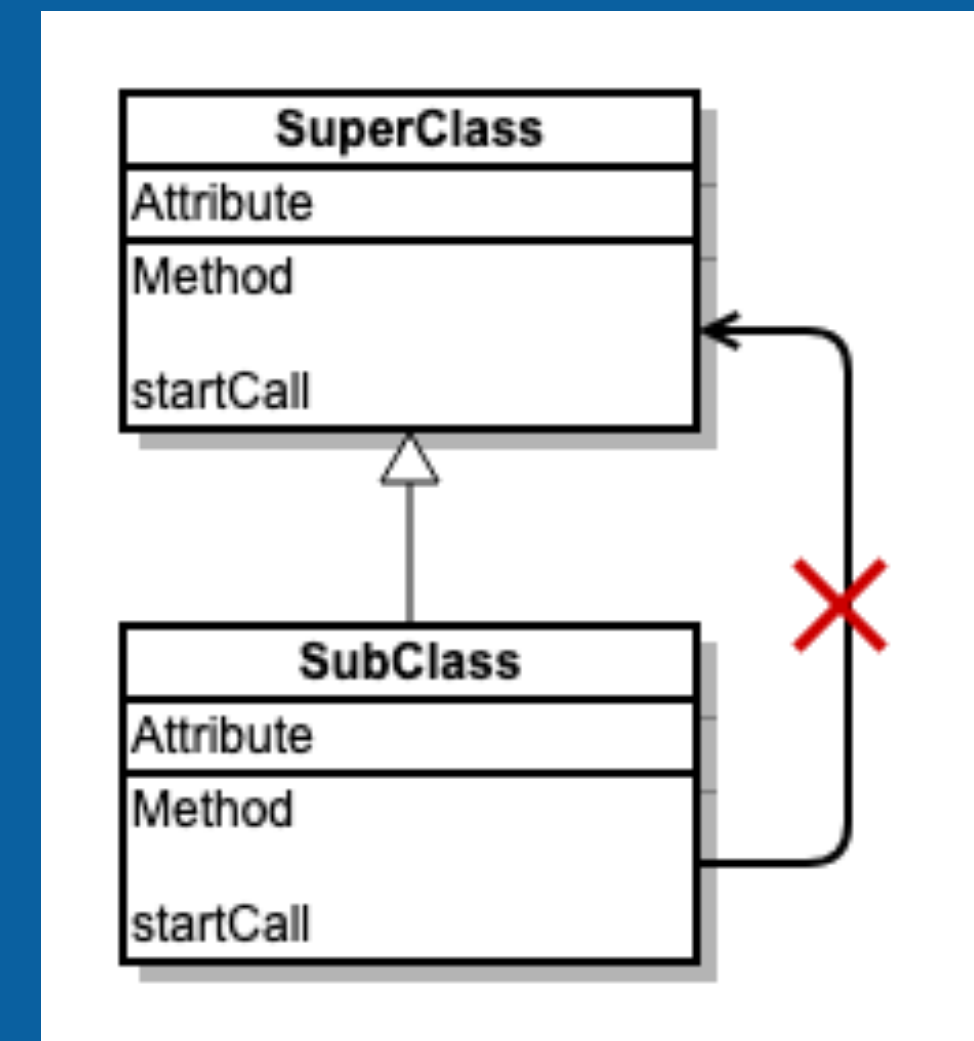
```
- (void)accept:(RCCallMediaType)type;  
  
- (void)hangup;
```

```
- (void)inviteRemoteUsers:(NSArray *)userIds mediaType:(RCCallMediaType)type;
```

<https://www.programmableweb.com/>

实践讨论：稳定的设计

- 少即是多
 - 信息是否必须要用？ http://api.example.com/cgi-bin/get_user.php?user=1
 - 如果只保留1个接口，2个接口？
 - 避免扩散：Interface/ Final
- 向后兼容



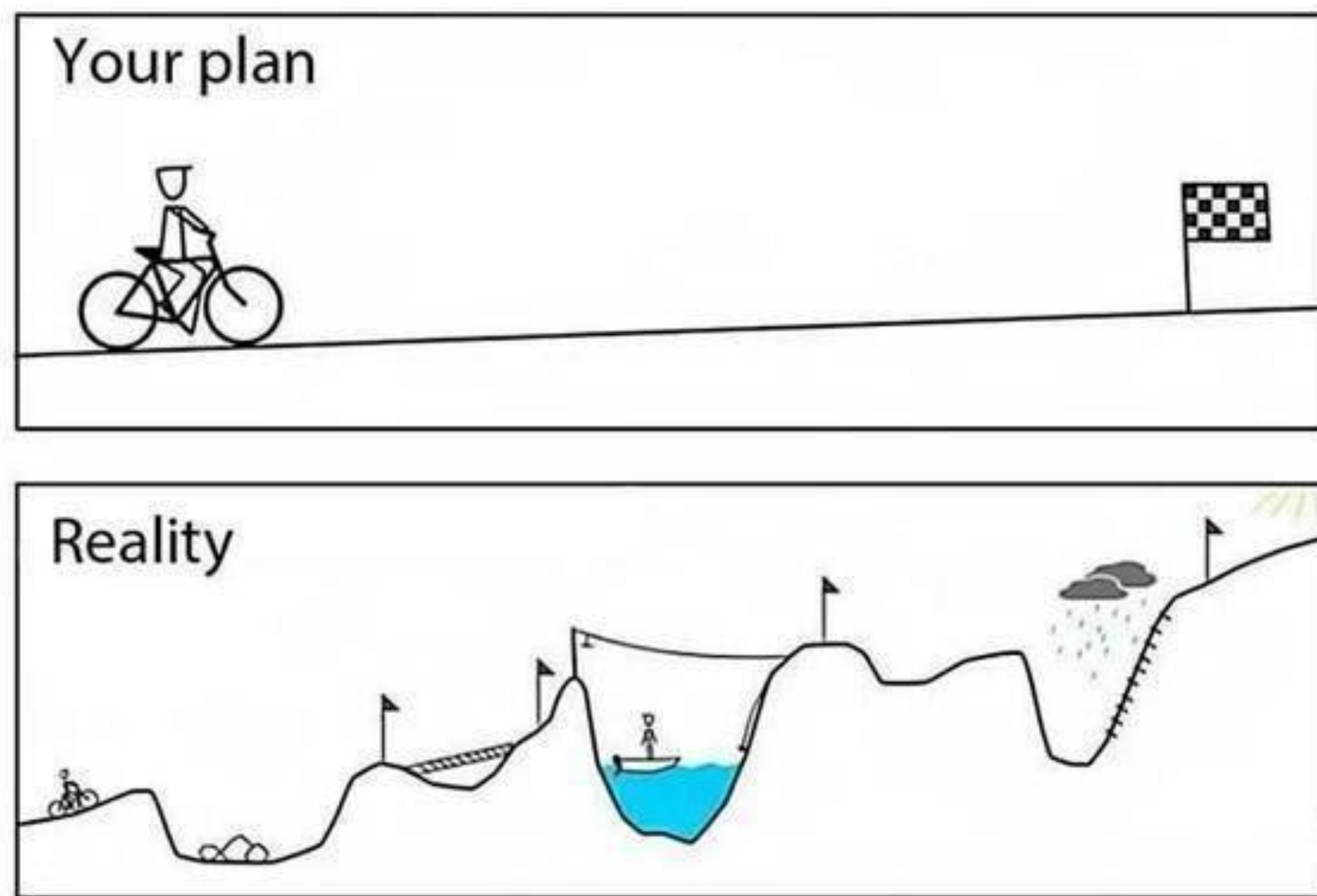
Java8: Interface
Java7: Final Class

Objective-C:
__attribute__((objc_subclassing_restricted))
__attribute__((objc_requires_super))

稳定的设计：向后兼容

- 100% 向后兼容是可能的吗？

否



稳定的设计：向后兼容

- 接口适配层，为业务预留空间
- URI 中预置版本号
 - `http://api.example.com/v1/users/12345?fields=name,age`
- 预定义错误码
 - 410 Gone
- 小版本之间直接迁移
- 使用新 API 重新实现旧 API，并标为废弃
- 抽象模型不同时：警惕同时维护两套 API

实践讨论： 编码

- 防御性编程
- LifeCycle 注册
- 检查版本号
- 运行库命名空间

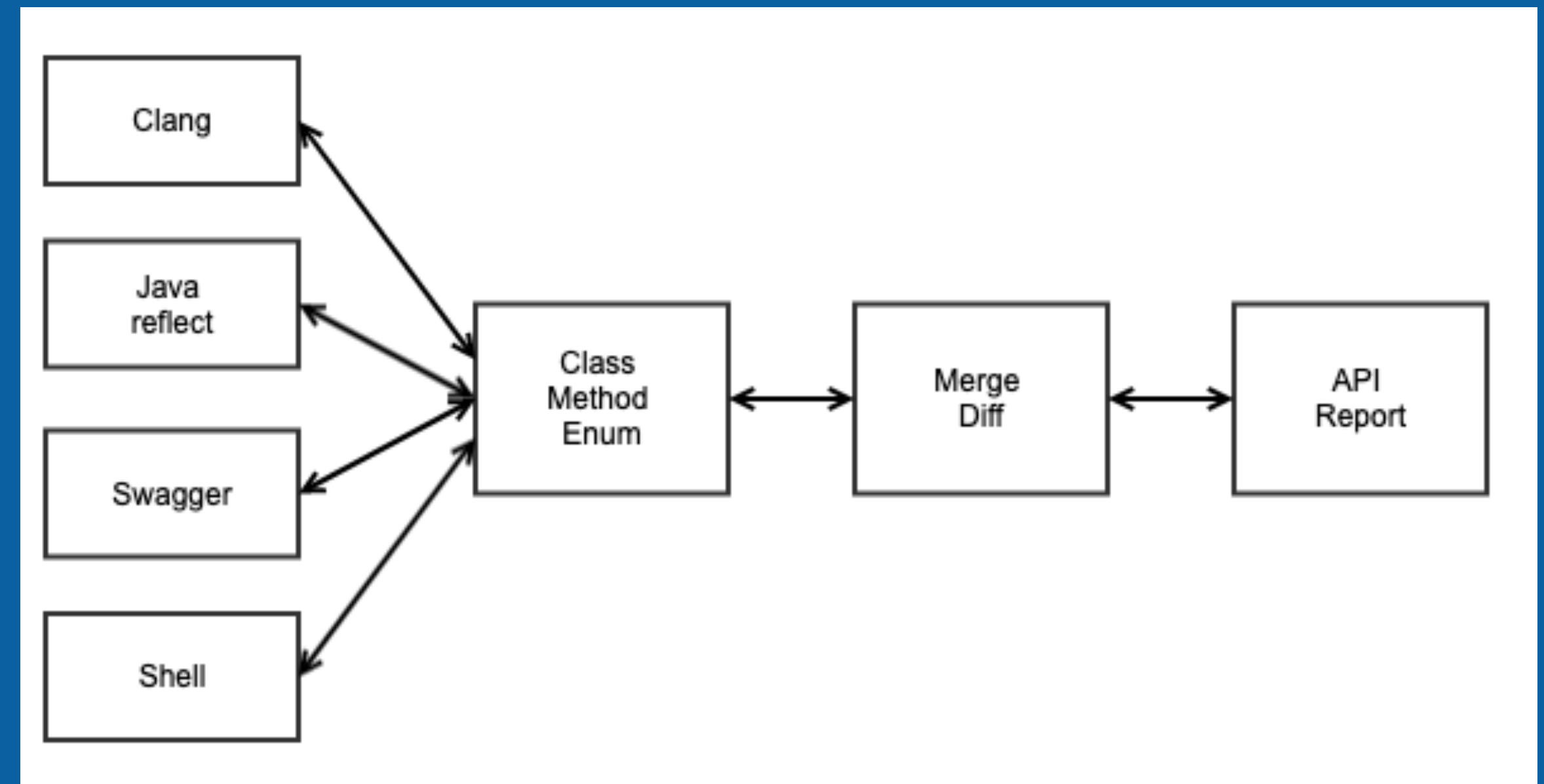
```
public abstract class API {
    public static final int VERSION = 1;
    protected API() {
        System.err.println("Initializing version " + VERSION);
        init(API.VERSION);
        System.err.println("Properly Initialized: " + this);
    }
    protected abstract void init(int version) throws IllegalStateException;
}

public class Impl extends API {
    @Override
    protected void init(int version) throws IllegalStateException {
        if (version != API.VERSION) {
            throw new IllegalStateException("API version error!")
        }
    }
}
```

```
$cxx -MMD -MF $out.d $defines $includes $cflags $cflags_cc;
$cflags_pch_cc -c $in -o $out;
$objcopy --redefine-syms $redefine_symbols_in_files $out;
```

实践讨论: Review

- Code Review 是基础
- 自动化工具的参考报告
 - 静态分析：命名/复杂度
 - 接口变更统计
 - 接口 Unit Test
 - 文档自动生成
- 团队评审



实践讨论：发布与反馈

- 区分版本：dev/stable
- A/B Test
- 接口使用统计：APM
- Log：级别与回调

演进示例：一次爬坑的实践

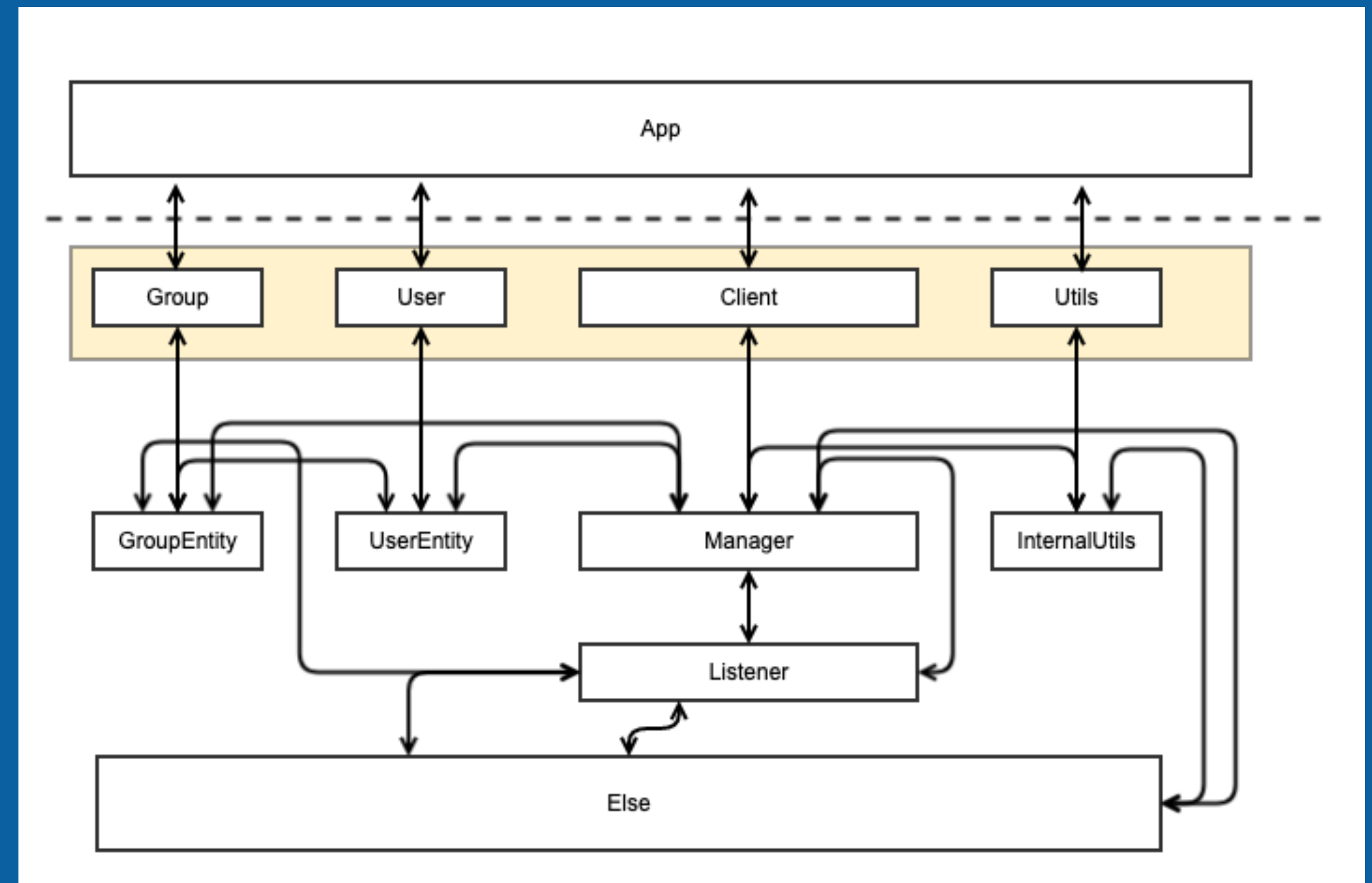
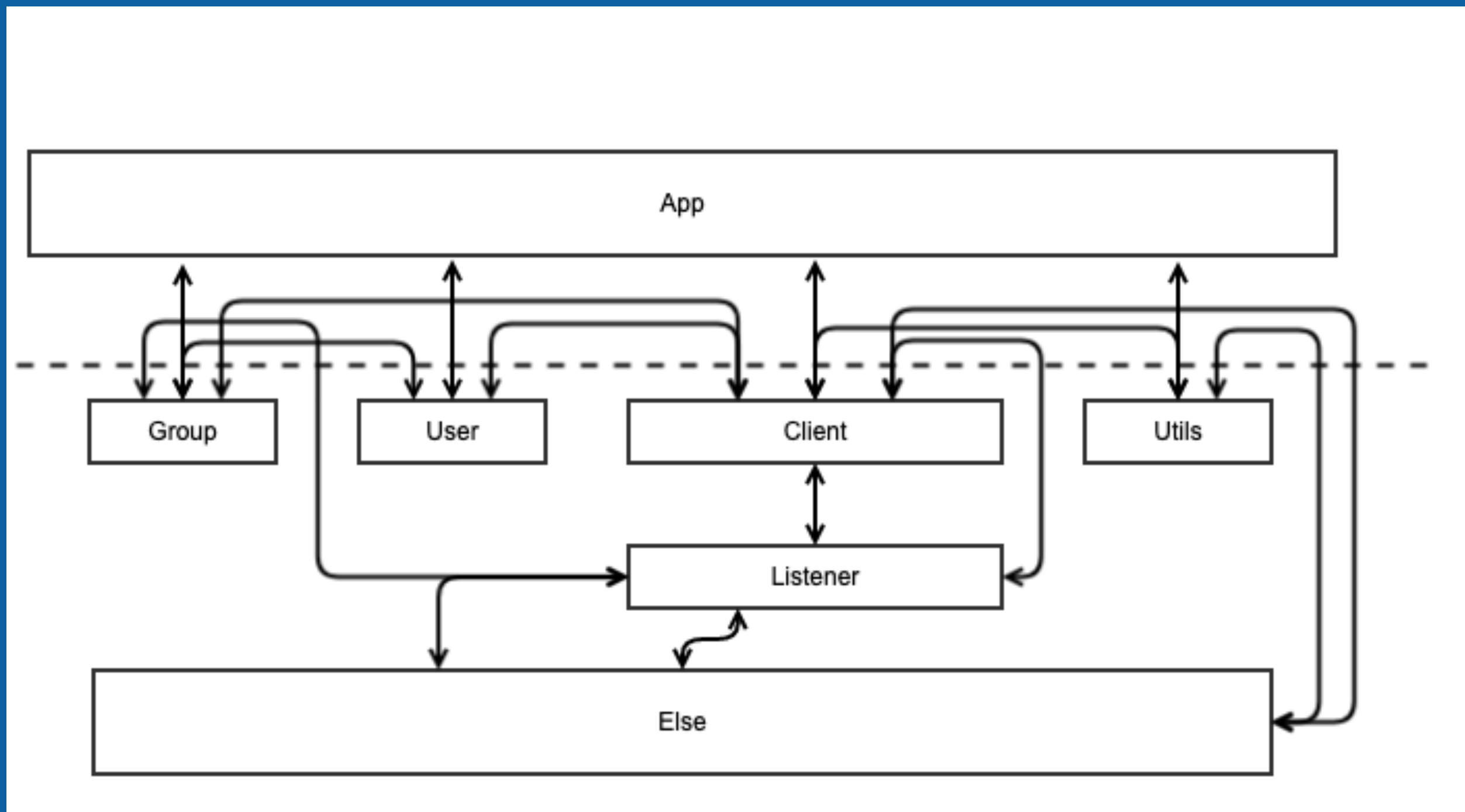
1. 假设拯救灾难项目：停下新功能的发布
2. 加入反馈：API 使用统计，A/B Test
3. 建立 API 评审的流程
4. Dog Food：写一个 App，审视 API 是否合理

起跑线

演进示例：一次爬坑的实践

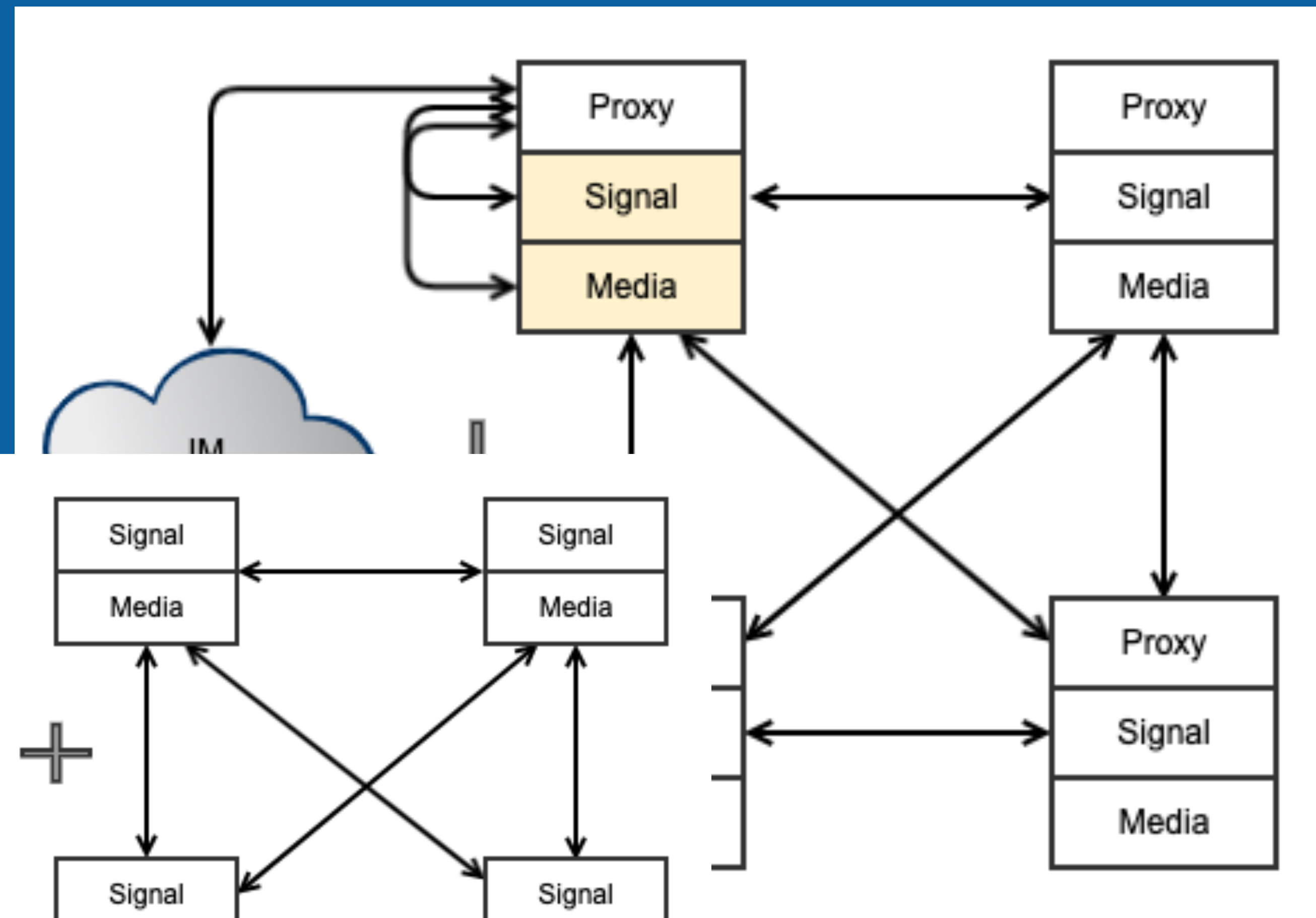
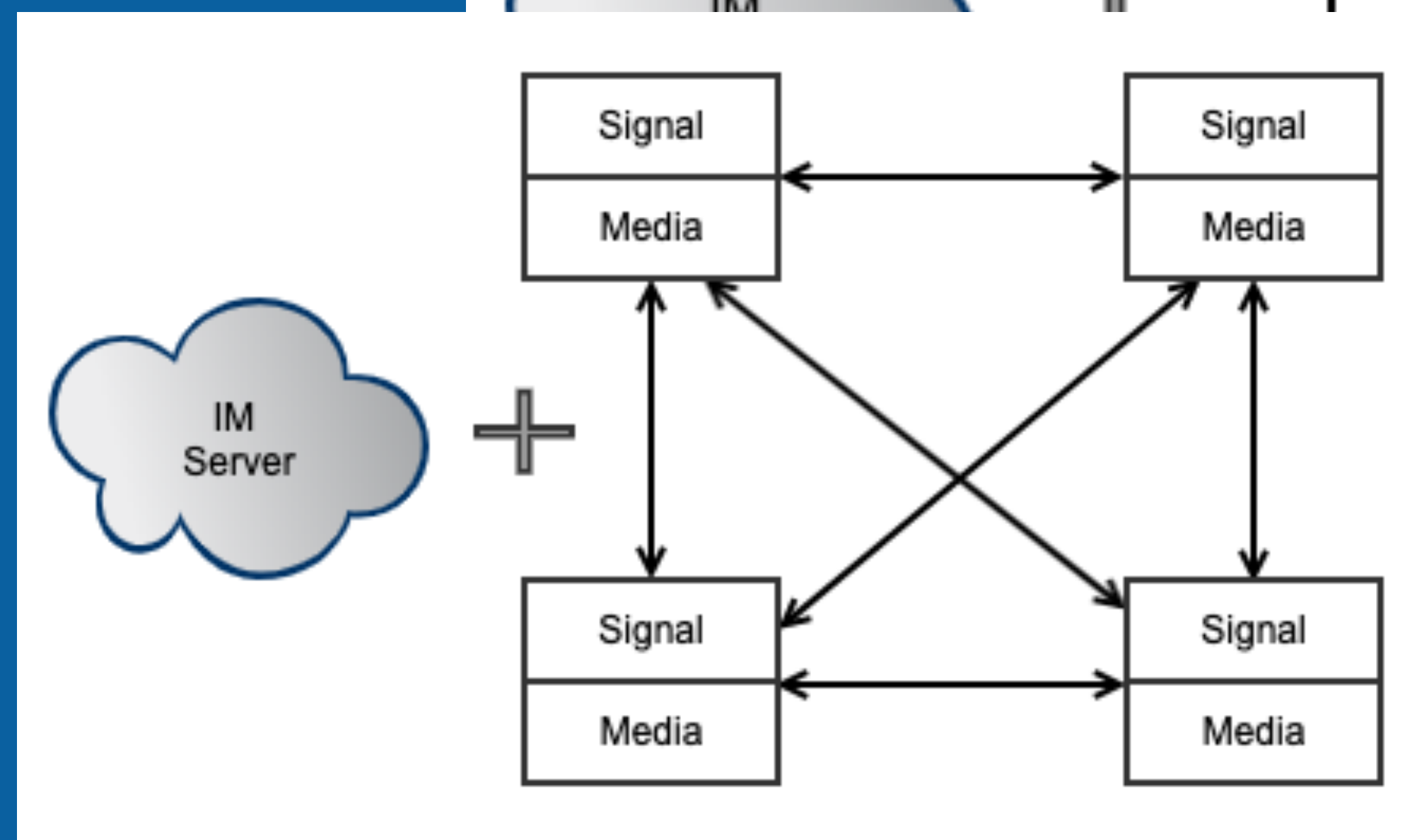
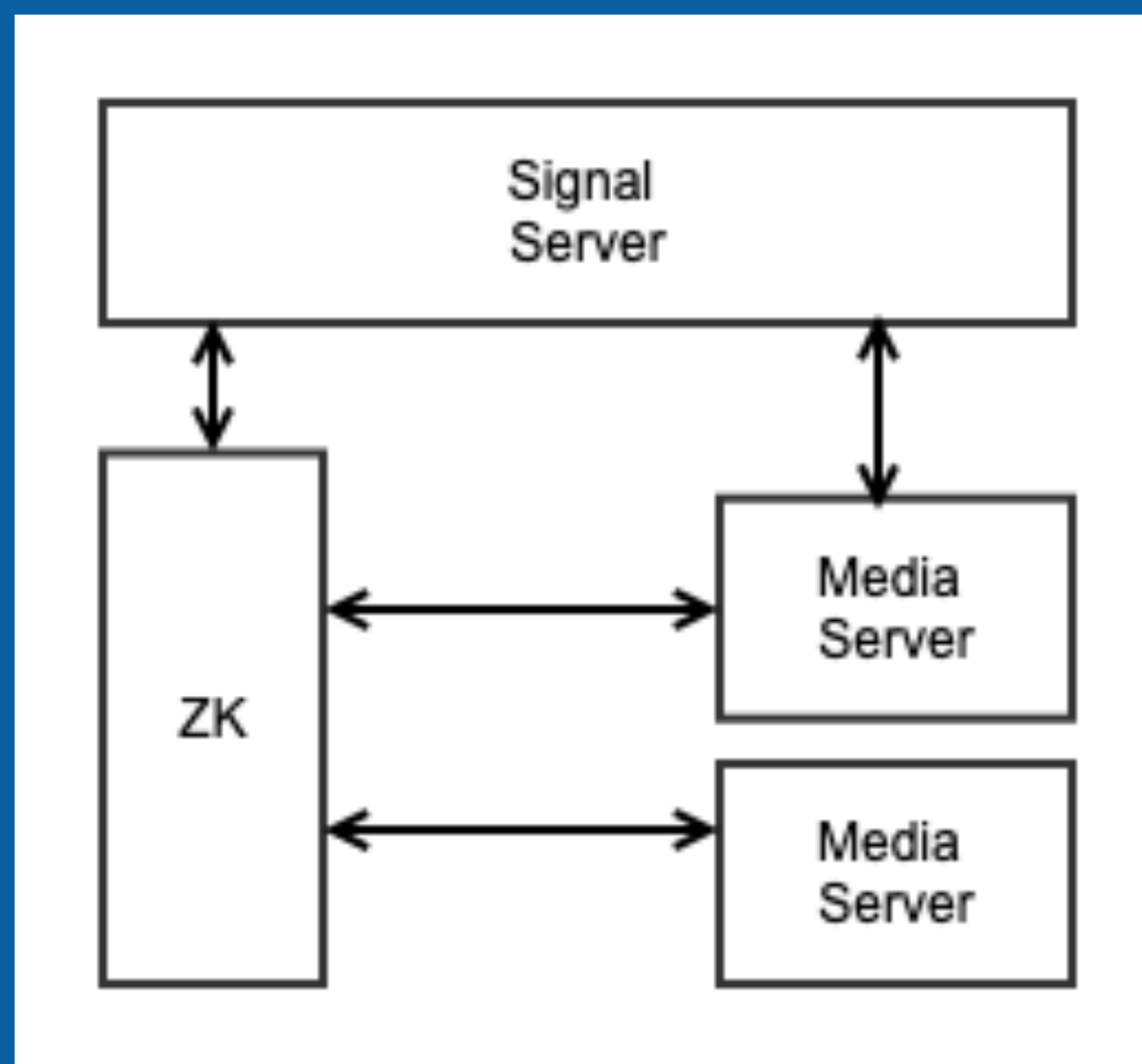
5. API 声明变更，interface/ final/ Getter/ Setter

6. 向下抽象适配层，为 API 调整预留空间



演进示例：一次爬坑的实践

7. 业务重构，用新的业务实现旧 API



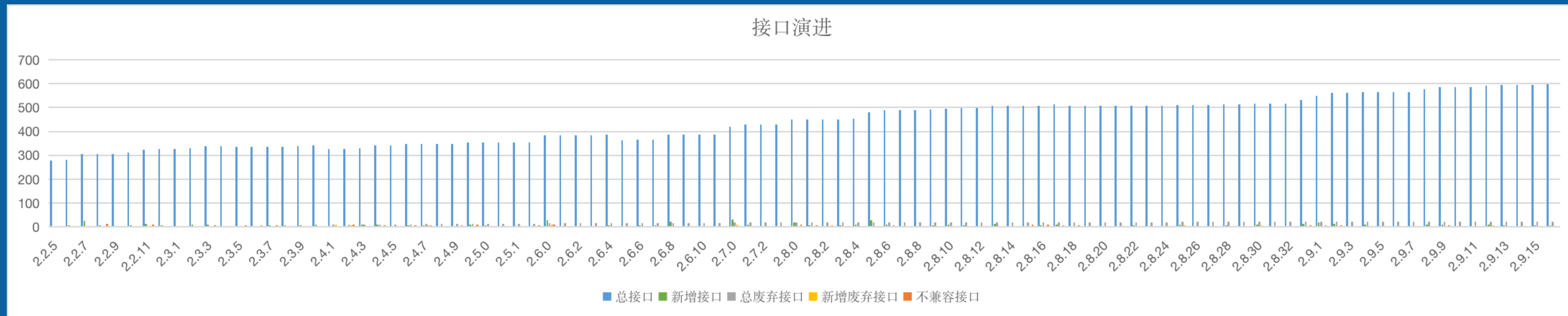
演进示例：一次爬坑的实践

- 8. 用新 API 重写 Demo , 尝试简单和复杂的功能
- 9. 发布 dev 版本并收集反馈
- 10. 发布 stable 版本更新

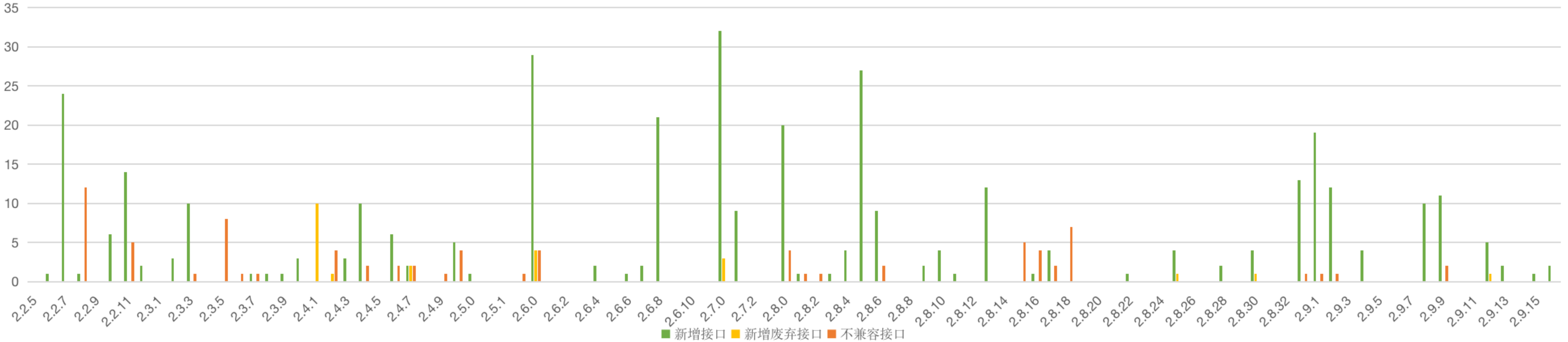
演进示例：数据统计展示

- API: 融云 IMLib iOS SDK
- 版本: 2.2.5 ~ 2.9.16 / 98
- 时间: 2015/07/27 ~ 2019/05/14

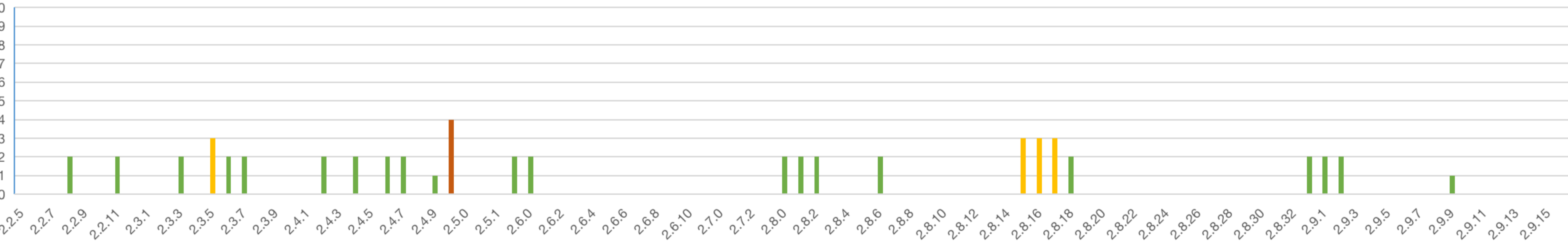
接口演进



变更接口



变更风险



总结

- 明确 API 的用户群体
- 建立 API Workflow 与度量指标
- 收集反馈
- 改进细节

TGO 鲲鹏会

汇聚全球科技领导者的高端社群

 全球12大城市

 850+高端科技领导者

使命
Mission

为社会输送更多优秀的
科技领导者

愿景
Vision

构建全球领先的有技术背景
优秀人才的学习成长平台



扫描二维码，了解更多内容

THANKS

QCon
全球软件开发大会