

项目总结报告

项目总结报告

日期：2023.9.10

组号	19	项目名称	Cealgull
编程语言	Golang/TypeScript/Python	开发平台和框架	Echo/React Native/Hyperledger Fabric

项目工作小结

1.是否实现了项目立项时的所有需求？列出实现的新增需求和未实现的需求。

基本实现了立项时的优先度高的所有需求。

- 1. 基本的论坛功能逻辑
- 2. 基本的基于论坛usecase的智能合约编写
- 3. 基本的基于IPFS的分布式存储

未实现需求：

- 1. 引入环签名算法后前端新架构的 CI/CD 工作流
- 2. 根据内容查找回复、建立群组等

2.采用哪种架构风格？哪些设计模式？

微服务架构

单例 职责链模式 工厂方法模式 装饰器模式

3.技术方案有哪些亮点？

- 1. 使用了IPFS的去中心化网络存储功能
- 2. 使用了hyperledger区块链技术
- 3. 前端使用混合开发框架 React Native 并实现了与主流移动端 Android 和 iOS 的兼容适配

4.是否做了单元测试？是否做了系统功能测试？是否做了性能测试？是否做了兼容性等其他非功能测试？

- 1. 完成功能测试
- 2. 完成性能测试
- 3. 完成了冒烟测试
- 4. 完成了兼容性测试

项目组成员对项目的贡献度（%）

注：项目管理包括项目计划、报告、沟通与协调等。

吴逸洋: 27%

徐轲： 25%

任柏俊： 25%

王浩丞： 23%

软件度量

软件代码行数（不包括注解行、空行和复用代码）：	20436
复用第三方代码行数：	408
类的个数：	148

经验、教训和建议

- 1. 数据库关联键选用 -- 由于触发合约与数据在业务节点的创建时间有先后性，所以不能使用常规的数据库的方法进行关联，仅能使用与时序无关的一些键例如hash值或者是uuid来进行唯一性标注。
- 2. fabric 部署 -- fabric虽然历经了多次的迭代已经有了良好的usecase历史，但是其文档过于繁杂，且存在过时文档，导致部署的时候经常出错。

3. 没有使用k8s进行集群部署 -- fabric是有k8s的 operator 但是由于业务开发的周期较为紧张导致最终部署时有些许困难。
4. 数据库读写性能有些问题，考虑后期增加缓存层并将业务数据库的链下数据进行读写分离
5. 提前评估开发风险，做好风险预防和风险应对工作。本次项目开发过程中，前端的核心技术栈 React Native 仍在快速迭代，导致我们不得不重新配置构建系统，影响项目的稳定性。
6. 谨慎对待开源框架，不应过分依赖。大量引入活跃的开源框架可能导致项目的稳定性下降，同时会破坏代码的可拓展性。如果开源项目停止维护（archive），很可能显著提升项目维护成本，浪费时间和人力。
7. 充分进行项目规划，对每个阶段的迭代目标应该有心理预期。对于分布式开发场景，利用版本控制工具持续化集成和部署是十分重要的。

项目组各成员签字：王浩丞 吴逸洋



徐轲