

Liuchang

教育经历

华东交通大学	交通运输、土木工程（双专业） 工科学士学位 一本 统招	2011/09 -- 2015/07	

工作经历

富登信贷 Java 工程师	2021/02 -- 2022/09
快牛智能 Java 工程师	2020/06 -- 2020/12
有利网 Java 工程师	2019/02 -- 2020/02
旺小宝科技 Java 工程师	2017/02 -- 2019/02

专业技能

- 1、掌握密码学常用技术，如对称、非对称加密、分组模式、签名等
- 2、掌握区块链区块结构、交易结构、共识、钱包、utxo 等相关技术
- 3、掌握 solidity 语言，掌握以太坊智能合约的编写部署
- 4、掌握 web3 开发，掌握 metamask、remix、infura 等开发工具的使用，掌握开发框架 truffle、hardhat
- 5、掌握 geth 和私有链的搭建、掌握 ganache 等工具的使用
- 6、掌握常见智能合约攻击手段和防御方案 (<https://github.com/lc1993929/smart-contract-attack>)
- 7、掌握代理合约 (<https://github.com/lc1993929/proxy-contract>)
- 8、掌握oracle智能合约
(<https://github.com/lc1993929/goProjects/tree/master/src/myOracleDemo>)
- 9、掌握智能合约常用框架，如chainlink、openZeppelin等
- 10、熟悉 evm，熟悉 ERC20、ERC721 等规范，熟悉去中心化交易所原理
- 11、熟悉2层链Polygon，熟悉乐观、zk汇总方案，熟悉 zkSync(<https://github.com/lc1993929/goProjects/tree/master/src/zkSyncDemo>)
- 12、熟悉 ipfs及其常用网关，如pinata等
- 13、熟悉前端 js、ts、vue、react、next、node.js 等技术
- 14、了解FlashLoan，了解MEV和FlashBot，了解JSON-RPC
- 15、了解安卓相关开发技术

- 1、掌握 java 基础、容器、并发、jvm 以及 java1.8 中的新特性
- 2、掌握 mysql、oracle 等关系型数据库，熟悉 mycat 等分库分表技术
- 3、掌握 redis，熟悉 mongodb 等非关系型数据库
- 4、掌握企业级高并发、高性能、高可用相关技术架构
- 5、掌握 23 种常用设计模式，熟悉常用设计原则
- 6、掌握常用数据结构和算法
- 7、掌握 kafka、rocketMq 等 mq 技术
- 8、掌握计算机网络相关知识
- 9、掌握分布式微服务以及 RPC 等相关技术
- 10、掌握 zookeeper、eureka 等常用服务中心
- 11、掌握 tomcat、nginx 等常用中间件技术
- 12、掌握 spring 等 java 常用框架技术
- 13、掌握 go 基础、并发、常用标准库、指针等
- 14、掌握 beego 等 go 常用框架
- 15、熟悉 eleasticsearch、logstash、kibana
- 16、熟悉 k8s+istio，熟悉 docker
- 17、熟悉 linux 相关知识
- 18、了解撮合引擎、永续合约等交易所技术

项目经验

自有NFT项目

白名单介绍:

Github: <https://github.com/lc1993929/Whitelist-Dapp>

访问地址: <https://whitelist-dapp-blue-chi.vercel.app/>

nft在发售前给前期支持者们提供一个白名单，支持者们可以自行加入白名单，在nft发售时，白名单中的支持者拥有优先购买nft的权利。

NFT介绍:

Github: <https://github.com/lc1993929/NFT-Collection>

访问地址: <https://nft-collection-kappa-ten.vercel.app/>

提供一个页面让客户可以使用eth购买nft，在发售前5分钟，只允许白名单中的地址购买nft，5分钟后允许所有地址自由购买nft。购买nft后，购买者可以在opensea查看自己的nft。

TOKEN介绍:

Github: <https://github.com/lc1993929/ICO>

访问地址: <https://ico-lovat.vercel.app/>

token发售页面，如果连接的地址拥有关联的NFT，那么每一个NFT都会给客户自动发送10个TOKEN。其它地址可以自由的使用eth购买token。

DAO介绍:

Github: <https://github.com/lc1993929/DAO-Tutorial>

访问地址: <https://dao-tutorial-opal.vercel.app/>

基于nft的dao组织，拥有nft的地址可以发起提案，在规定的时间内，拥有nft的地址可以投票决定是否支持提案。在投票时间结束后根据投票情况决定是否执行该提案。

Defi-exchange介绍:

Github: <https://github.com/lc1993929/defi-exchange>

访问地址: <https://defi-exchange-pearl.vercel.app/>

仿照uniswap的去中心化交易所，但是只能交易我发行的token和eth。客户可以选择将自己闲置的token和eth添加到交易所中增加流动性，在客户退出流动性的时候可以获得对应的奖励。客户也可以通过该交易所进行token和eth之间的交易，每一次交易都会收取客户1%的手续费。在刚开始增加该交易通

道，没有流动性的时候，客户第一次增加流动性，可以不添加eth，只添加token即可增加流动性。后续再增加流动性的时候必须按照比例提供对应的token和eth，否则不允许增加流动性。

使用技术：

solidity、next、ethers、metamask、hardhat、openezppelin、

众筹平台

代码仓库：<https://github.com/lc1993929/goProjects/tree/master/src/funding>

项目介绍：

众筹核心逻辑放在以太坊的智能合约中，解决了传统 web2 的众筹模式中项目方跑路或众筹平台跑路的问题。众筹项目方首先发起众筹项目，众筹支持者通过转账给智能合约账户支持对应的众筹项目。项目方通过智能合约发起支付申请，支持者通过群体投票决定是否同意支付请求。如果众筹失败，将智能合约中的支持者的金钱退回给对应的支持者。

前端页面通过 react 框架展示，使用web3 来与智能合约进行交互。使用 metamask 等插件获取到 web3 的provider。使用 npm 做依赖包的管理。

使用技术：

solidity、react、web3、metamask

富登信贷

审批、贷后、催收系统

审批项目介绍：

该项目主要是公司内部使用的审批系统，在客户进件后，会根据进件的产品和金额大小分别进行自动审批或由公司的审批人员进行人工审批。审批完成后将审批结果推送给进件系统。

贷后系统介绍：

在放款后，需要针对进件的还款情况进行跟踪处理，以便能收回放贷的金额。

催收系统介绍：

在用户逾期还款时，及时提醒用户进行还款。在用户拒绝还款时可以进行委外催收或提起诉讼等操作催促用户进行还款。

审批模块介绍：

审批流：针对单个的进件可能会由不同部门、不同工种、不同级别的审批人员进行审批。所以需要有一个审批流进行审批流程的管理。公司使用的审批流是由供应商根据 activity 魔改后的审批系统。

自动审批：一些符合预定条件的进件不需要人工进行详细信息的审批，那么就由系统根据风控和相关的业务规则进行自动审批。

人工审批：一些进件由于某些特点（例如金额较大等）需要审批人员校对进件的详细细节进行审批，那么

就需要一个审批页面进行审批。人工审批时，需要将贷前系统的进件数据进行同步，并且需要保留审批记录。在计算审批通过金额时，需要将详细的计算过程展示给审批人员。

互金数据同步：使用 canal 监听互金系统的数据库写操作，同步 binlog 到 rocketmq 中，审批系统通过解析 binlog 同步互金数据落库到审批系统的数据库中。

crm 数据同步：crm 系统将客户数据同步到中间库中，审批系统通过 oracle 数据库的存储过程将中间库的数据同步到审批系统的数据库中。

贷后模块介绍：

每日 emi 还款：每天都要跑定时任务去看今日是否到了还款计划中的规定还款日，如到了还款日，那么就需要去到用户的银行卡进行金额的划扣来还款。

入催：在划扣用户金额不成功或金额不足时，用户就产生了逾期。那么需要将对应的进件推送给催收系统进行催收。

展期管理：在客户无力偿还贷款时，可与业务人员商议延期或再次拆分多期还款等。此时可进行展期调整来调整还款计划。

熔断管理：通过计算保证收取用户金额不会超过政策规定的上限额度。

每日跑批：每天凌晨会计算每一单的还款金额、利息等数据。定时任务使用的是 java 自带的 timer 执行。计提、核销等

催收模块介绍：

AI 电话短信催收：根据业务规则对接三方供应商进行电话或短信方式提醒用户还款。

委外：在难以收回欠款时，将资产卖给外部收款服务商。

诉讼：在难以收回欠款时，去到法院提起诉讼申请以便收回欠款。

电催、直催：每日跑批生成催收任务推送到供应商自动拨打客户电话进行电话催收。

展期、划扣等

主要负责：

负责整个项目的管理和交付质量，并承担部分开发任务。

使用技术：

java、spring、mysql、rocketmq、oracle、canal

有利网

K 利贷贷前系统

项目介绍：

该项目主要做贷款服务，用户可通过此项目发起贷款申请，提交进件。

在进件前需要对用户做人脸识别、银行卡审核、合同签署、个人信息的录入以及审核等一系列审核校验，并将通过后的进件提交给贷中系统和贷后系统分别进行人工审核和贷款发放以及后续还款追踪等。

模块介绍:

进件服务: 是平台用户借款的入口服务, 主要功能为用户信息收集, 借贷风险控制, 用户差异化运营, 借款额度管理, 合同流程管理(签署合同并留底), 借还款流程管理等功能, 并可以实现将不满足进件资格的用户引流至三方合作渠道, 在通过风险检查之后, 推送到贷中和贷后系统是贷款业务的核心处理系统。使用 redis 来做分布式锁防止并发问题。

用户服务: 用户的来源可通过公司的老项目提供用户数据, 通过老项目的用户数据计算出一个用户白名单, 在用户进入本项目时, 通过查询用户白名单来确定用户是否有资格进行贷款, 如果已有用户数据则将用户数据同步至本服务。后期由于信贷行业政策变化, 新增用户只可通过自行注册并自行填写用户数据。服务之间的调用使用 istio 来处理。

任务服务: 用户在进入首页时会自动检查用户的初始化任务是否生成, 以及任务完成状态是否正确的检查。在用户的使用过程中, 会根据用户提交的数据以及相关校验产生不同的任务引导用户去完成不同的操作流程。在完成任务和生成任务的同时会执行一些操作来修改用户的相关权限和数据。

额度服务: 管理用户的借款额度, 主要功能是处理用户额度的赋予, 冻结, 扣减, 管理用户的额度状态。

人脸服务: 人脸识别的第三方引入以及人脸识别整体流程实现, 还要将人脸的相关数据进行保存以及根据人脸数据修改用户的个人信息和任务情况等数据。

风控服务: 第三方引入以及人脸识别整体流程实现, 管理人脸识别数据, 实现风险控制, 权限管理。

支付中心服务: 支付中心主要负责管理用户的银行卡、支付密码等支付相关数据。

流量分发: 将不满足进件资格的用户引流至三方合作渠道。根据用户的相关风险程度和个人情况分别引流到不同的三方合作渠道。

贷中贷后推送: 在推送进件数据之前, 先在风控中心进行风控校验, 通过后使用 mq 将进件数据分别推送给贷中系统和贷后系统。

主要负责:

负责在推送到贷中放款前的所有流程的相关业务功能的开发任务。

使用技术:

java、springboot、k8s+istio、redis、mysql、kafka