Inline Text Wrapping Picture

北京邮电大学

硕士研究生学位论文开题报告

学 号: 2017180215

姓 名: 李超

学 院: 软件学院

专业(领域): 软件工程

研究方向:

导师姓名: 崔毅东

攻 读 学 位: 工程硕士

2018年12月6日

|  |  |  |  |
| --- | --- | --- | --- |
| 论文题目 | 聚合支付系统的设计与实现 | | |
| 选题来源 | 其他 | 论文类型 | 应用研究 |
| 开题日期 | 2018-12-04 | 开题地点 | 北京邮电大学 |
| **一、立题依据（包括研究目的、意义、国内外研究现状和发展趋势，需结合科学研究发展趋势来论述科学意义；或结合国民经济和社会发展中迫切需要解决的关键科技问题来论述其应用前景。附主要参考文献目录）（不少于800字）**  1.1.研究目的  目前市场上存在大量电商平台和类电商平台，这些平台具有多样的支付结算需求，通过调研市场实际情况，总结起来主要有以下痛点：  行业合规性：目前市场上多数的企业没有持有第三方支付牌照。一方面，对于企业来讲，为了更好的为用户提供优质服务，针对用户的支付结算需求是必要的；另一方面，每年市场上都会存在平台跑路现象，为了避免企业设立资金池、捐款跑路，关于“二清”监管是非常有必要的。由此，支付的合规性问题，一直是行业的要解决的一个痛点。  释放人力资源：传统模式下，当同一个集团旗下具有多个业务线时，每个业务线会分别与外部支付公司进行开发对接，这样造成重复对接、严重浪费技术资源；同时，对于财务而言，由于需要管理的系统较多，也因此产生了很多的财务管理工作。  解决信息孤岛：传统模式下，当同一个集团旗下具有多个业务线时，每个业务线会分别与外部支付公司进行开发对接，每一条业务线的用户数据分别存储于各自的数据库，用户信息散落在各个业务系统，无法做到资源共享，最大化利用。对于用户而言，同一个集团或同一个APP下的不同业务线需要分别绑定银行卡，用户体验极其不好；对于企业而言，由于各个业务线的用户数据是割裂的，因此不能对用户的数据资源进行有效利用。  提升业务效率：传统模式下，对于新的业务需要单独开发，不能灵活复用，造成项目上线周期长，错过市场时机；同时，一些小企业不具备开放能力，也因此上线支付能力遇到困难。  系统稳定性：由于多数企业开发能力、技术资源优先，通常系统的底层通道单一，也因此系统的稳定性过度依赖于底层单一通道；  总结下来，“聚合支付平台系统”解决了以下问题：1.平台仅过信息流，不做资金流，从而解决了“二清”的合规性问题；2.释放技术资源，避免重复工作、多次开发；3.实现集团企业多业务信息共享；4.提升支付结算效率；  1.2.研究意义  “聚合支付平台系统”是解决企业合规的开展支付业务，在不同渠道重复对账、独立统计的弊端，实现收款、结算、数据统计的全面聚合，打造统一接口、SDK管理，实现一次接入享受聚合平台所有通道，极大的简化了多次对接支付通道的繁琐流程、释放了多次开发对接的技术资源。  1.2.1.社会效益  “聚合支付平台系统”作为业务支付阶段的重要一环，承担着较大的社会责任。以某金融公司的聚合支付平台系统为例，主要具有以下社会意义：  （1）提高了用户提现、还款时的体验，更加多途径、更加便利的实现提现与还款操作。有效减小了用户的操作成本，对降低用户逾期率有显著效果，让金融回归“普惠”的本质；  （2）给合作商户提供更快速、便捷的入款途径，合理地避免了因账期问题导致的商户资金周转问题，为友商创造了良性的运营环境；  （3）方便金融监管，通过支付平台对金融各业务线支付订单账务统一管理，统一对小贷各业务的交易数据管理，方便统一上报，有效降低了监管成本；  1.2.2.经济效益  “聚合支付平台系统”担负着提升公司效益的作用，以某金融公司的聚合支付平台系统为例，给公司创造着经济效益：  提高用户还款成功率：  （1）用户多种途径还款，显著降低了用户的逾期率；  （2）多层多支付通道自动路由切换，智能选择成本最低的支付通道，有效降低了金融各业务的运营成本；  （3）优秀的用户体验，对用户留存做出了巨大贡献；  1.3.现状  目前市场上聚合支付平台近百家，行业野蛮发展，缺乏行业规范。其中一些平台并不能很好的满足监管的要求。 | | | |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **二、研究内容和目标（说明课题的具体研究内容，研究目标和效果，以及拟解决的关键科学问题。此部分为重点阐述内容）（不少于2500字）**  **北 京 邮 电 大 学**  **研 究 生 论 文 开 题 报 告**  学 号 2017180215  姓 名 李超  申请学位 工学硕士  专业名称 软件工程  研究方向 企业信息化  导师姓名 崔毅东  所在院系 软件学院  报告日期： 2018 年 12 月 3 日  目录  [1. 立题依据 4](#_Toc11351)  [1.1. 研究目的 4](#_Toc24767)  [1.2. 研究意义 4](#_Toc563)  [1.2.1. 社会效益 5](#_Toc12606)  [1.2.2. 经济效益 5](#_Toc15251)  [1.3. 现状 5](#_Toc8483)  [2. 研究内容和目标 5](#_Toc16890)  [2.1. 研究目标和效果 5](#_Toc3411)  [2.2. 鉴权签约 6](#_Toc31325)  [2.2.1. 目的 6](#_Toc31729)  [2.2.2. 需求描述 6](#_Toc15184)  [2.2.3. 流程图 7](#_Toc5714)  [2.3. 代付 8](#_Toc18845)  [2.3.1. 目的 8](#_Toc29629)  [2.3.2. 需求描述 8](#_Toc13960)  [2.3.3. 流程图 9](#_Toc17174)  [2.4. 扣款产品 10](#_Toc5458)  [2.4.1. 目的 10](#_Toc4826)  [2.4.2. 需求描述 10](#_Toc27066)  [2.4.3. 流程图 10](#_Toc12833)  [2.1. 收银台需求 12](#_Toc32646)  [2.1.1. 总体流程 12](#_Toc14971)  [2.1.2. 支付授权流程页面详述 13](#_Toc7006)  [2.1.3. 支付流程页面详述 18](#_Toc4577)  [2.2. 路由中心 20](#_Toc14210)  [2.2.1. 目的 20](#_Toc27928)  [2.2.2. 路由说明 20](#_Toc21274)  [2.2.3. 路由流程 21](#_Toc24114)  [2.3. 鉴权签约路由 21](#_Toc20798)  [2.3.1. 目的 21](#_Toc6163)  [2.3.1. 决策因子 21](#_Toc3289)  [2.3.2. 决策规则 22](#_Toc5925)  [2.3.3. 示例说明 22](#_Toc12571)  [2.4. 付款路由 23](#_Toc8348)  [2.4.1. 目的 23](#_Toc8733)  [2.4.2. 决策因子 23](#_Toc21759)  [2.4.3. 决策规则 24](#_Toc24532)  [2.4.4. 路由流程图 24](#_Toc14290)  [2.5. 扣款路由 25](#_Toc23958)  [2.5.1. 目的 25](#_Toc23941)  [2.5.2. 决策因子 25](#_Toc23218)  [2.5.3. 决策规则 25](#_Toc14890)  [2.5.4. 路由流程图 26](#_Toc27891)  [2.6. 收银台支付路由 26](#_Toc7799)  [2.6.1. 目的 26](#_Toc13312)  [2.6.2. 决策因子 27](#_Toc7701)  [2.6.3. 决策规则 27](#_Toc9464)  [2.7. 解决的主要问题 27](#_Toc12635)  [3. 研究内容和目标 28](#_Toc6079)  [3.1. 研究方法 28](#_Toc13789)  [3.2. 技术路线 28](#_Toc19409)  [3.3. 可行性分析 28](#_Toc19579)  [4. 课题特色 31](#_Toc2203)  [4.1. 少有人研究 31](#_Toc18640)  [4.2. 经济效益 31](#_Toc23974)  [4.3. 社会效益 32](#_Toc17713)  [5. 研究基础与工作条件 32](#_Toc26909)  [5.1. 软件工程知识 32](#_Toc18337)  [5.2. 工作基础 32](#_Toc32623)  [5.3. 开发环境 33](#_Toc1890)  [5.4. 研究思路 33](#_Toc9505)  [6. 论文计划 33](#_Toc8203) 立题依据研究目的 目前市场上存在大量电商平台和类电商平台，这些平台具有多样的支付结算需求，通过调研市场实际情况，总结起来主要有以下痛点：  行业合规性：目前市场上多数的企业没有持有第三方支付牌照。一方面，对于企业来讲，为了更好的为用户提供优质服务，针对用户的支付结算需求是必要的；另一方面，每年市场上都会存在平台跑路现象，为了避免企业设立资金池、捐款跑路，关于“二清”监管是非常有必要的。由此，支付的合规性问题，一直是行业的要解决的一个痛点。  释放人力资源：传统模式下，当同一个集团旗下具有多个业务线时，每个业务线会分别与外部支付公司进行开发对接，这样造成重复对接、严重浪费技术资源；同时，对于财务而言，由于需要管理的系统较多，也因此产生了很多的财务管理工作。  解决信息孤岛：传统模式下，当同一个集团旗下具有多个业务线时，每个业务线会分别与外部支付公司进行开发对接，每一条业务线的用户数据分别存储于各自的数据库，用户信息散落在各个业务系统，无法做到资源共享，最大化利用。对于用户而言，同一个集团或同一个APP下的不同业务线需要分别绑定银行卡，用户体验极其不好；对于企业而言，由于各个业务线的用户数据是割裂的，因此不能对用户的数据资源进行有效利用。  提升业务效率：传统模式下，对于新的业务需要单独开发，不能灵活复用，造成项目上线周期长，错过市场时机；同时，一些小企业不具备开放能力，也因此上线支付能力遇到困难。  系统稳定性：由于多数企业开发能力、技术资源优先，通常系统的底层通道单一，也因此系统的稳定性过度依赖于底层单一通道；  总结下来，“聚合支付平台系统”解决了以下问题：1.平台仅过信息流，不做资金流，从而解决了“二清”的合规性问题；2.释放技术资源，避免重复工作、多次开发；3.实现集团企业多业务信息共享；4.提升支付结算效率； 研究意义 “聚合支付平台系统”是解决企业合规的开展支付业务，在不同渠道重复对账、独立统计的弊端，实现收款、结算、数据统计的全面聚合，打造统一接口、SDK管理，实现一次接入享受聚合平台所有通道，极大的简化了多次对接支付通道的繁琐流程、释放了多次开发对接的技术资源。 社会效益 “聚合支付平台系统”作为业务支付阶段的重要一环，承担着较大的社会责任。以某金融公司的聚合支付平台系统为例，主要具有以下社会意义：   1. 提高了用户提现、还款时的体验，更加多途径、更加便利的实现提现与还款操作。有效减小了用户的操作成本，对降低用户逾期率有显著效果，让金融回归“普惠”的本质； 2. 给合作商户提供更快速、便捷的入款途径，合理地避免了因账期问题导致的商户资金周转问题，为友商创造了良性的运营环境； 3. 方便金融监管，通过支付平台对金融各业务线支付订单账务统一管理，统一对小贷各业务的交易数据管理，方便统一上报，有效降低了监管成本；  经济效益 “聚合支付平台系统”担负着提升公司效益的作用，以某金融公司的聚合支付平台系统为例，给公司创造着经济效益：   1. 提高用户还款成功率： 2. 用户多种途径还款，显著降低了用户的逾期率； 3. 多层多支付通道自动路由切换，智能选择成本最低的支付通道，有效降低了金融各业务的运营成本； 4. 优秀的用户体验，对用户留存做出了巨大贡献；  现状 目前市场上聚合支付平台近百家，行业野蛮发展，缺乏行业规范。其中一些平台并不能很好的满足监管的要求。 研究内容和目标研究目标和效果 支付通道的统一管理，包括信息与对接的管理，避免重复对接，释放资源，更好的支持各业务线的增量需求，支持新增业务线的快速对接，统一放支付通道的管理及使用。智能路由建立，根据费率、支付成功率，实现支付通道的用户无感知自动切换，同时起到容灾备份的作用。  项目主要功能：   1. 鉴权绑卡：用于核验用户提供的信息一致性，以及对用户审核进行核验； 2. 鉴权签约：用户通过请求并回填短信验证码与平台签约，授权平台进行扣款； 3. 代扣收款：根据与用户的约定，系统可以根据业务需要，对客户进行扣款； 4. 协议支付收款：用户完成短信签约后，系统可以根据业务需要，对客户进行扣款； 5. 收银台产品：具备前端页面，集优惠券营销、多种支付方式并存的支付方式，主要用于用户主动完成支付行为； 6. 代付产品：商户将自有资金结算给用户，如：现金贷业务，用户的提现操作； 7. 支付路由配置：根据业务需求，自动化路由最有的支付通道； 8. 后台管理：用于管理支付产品管理、业务线管理、用户管理、对账管理等；  鉴权签约目的 通过绑卡的形式进行用户信息一致性校验、鉴权与签约。免去后续支付时输入四要素。 需求描述  1. 鉴权绑卡规则：   在需要使用代收付的业务场景下，业务方发起银行卡绑定关系建立请求到支付平台，支付平台接收到交易请求后，在进行报文合法性和业务合法性校验后，建立请求到支付公司，接收支付公司返回结果后，存储订单信息并向业务方返回绑定交易的状态信息。  鉴权成功后，根据用户身份证号生成唯一支付用户ID，留存用户信息项目：姓名、身份证号、银行卡号、预留手机号、业务方编号、业务用户ID等。用户绑定的银行卡继承在业务方，即：各个业务方下的用户银行卡列表，是支付用户银行卡列表的子集。   1. 鉴权方式：   有短信验证，支付公司提供的需要用户短信授权的鉴权方式，有短验证需要配合使用“绑卡确认”、“发送短信验证码”接口；  无短信验证，支付公司提供的无需用户短信授权的鉴权方式，无短验证不需要使用“绑卡确认”、“发送短信验证码”接口。业务方可根据需要选择“有短”或“无短”鉴权方式，目前项目以及合作支付通道，均采用“无短”模式。   1. 绑定卡种选择：   如果绑定的是借记卡，请求报文主要包括：商户号（业务方）、商户订单号、业务用户ID、业务类型、卡用途、渠道编码、银行编码、卡号、卡种、证件号、证件类型、姓名、预留手机号；  如果绑定的是信用卡，求情报文主要包括：商户号（业务方）、商户订单号、业务用户ID、业务类型、卡用途、渠道编码、银行编码、卡号、卡种、证件号、证件类型、姓名、预留手机号、信用卡有限期、CVV2号；目前项目不支持信用卡。   1. 多渠道下绑卡规则：   用户的某个银行卡在进行首次鉴权时，为了节省成本，仅鉴权（认证绑卡）主通道。在执行支付路由，当第N个通道不能使用时，自动路由到第（N+1）个通道，如果第（N+1）个通道下该银行卡尚未进行鉴权，调起平台包装的“鉴权+代扣”组合产品，依次类推。 流程图  代付目的 完成资金出款，如用户的提现操作、给商户返佣的操作。 需求描述 由业务方发起交易请求，支付平台在通过报文合法性和业务合法性校验后，向支付公司发起的代付交易请求，银行渠道负责验证由支付公司发送的卡信息相关数据，通过风控规则后，进行资金的出款。  当同一个业务方多个支付通道时，支付平台负责筛选当前订单下的可用通道，并计算各可用支付通道成本费用和通过率（也可由业务方指定由哪一个支付通道出款），做出路由决策。（参见支付路由章节）  重要说明：同一支付订单号，保证出款成功一次。 流程图  扣款产品目的 完成资金入款。应用场景，如：现金贷用户主动操作还款、系统发起还款。 需求描述 由业务方发起交易请求，支付平台在通过报文合法性和业务合法性校验后，向支付公司发起的代付交易请求，银行渠道负责验证由支付公司发送的卡信息相关数据，通过风控规则后，进行资金的扣款。  当同一个业务方多个支付通道时，支付平台负责筛选当前订单下的可用通道，并计算各可用支付通道成本费用和通过率（也可由业务方指定由哪一个支付通道出款），做出路由决策。（参见支付路由章节）  重要说明：同一支付订单号，保证扣款成功一次。 流程图  收银台需求总体流程  * 流程图      * 重要节点说明  1. 当用户选择进件时“原绑定的代扣卡”时进行支付时，首先判定该卡是否已在快捷支付通道授权。如果没有授权，则进入填写四要素页面，且原代扣卡信息反显，待用户确认授权（为保证安全，仅手机号可以编辑修改）；如果已经授权，则进入下一个节点。 2. 是否已设置过密码，该支付用户是否已设置过支付密码。 3. 是否需要短验，底层支付公司对该订单是否需要进行短信验证码校验。一期连连快捷支付短验规则：首次支付一定需要短验；再次支付时，当订单金额大于5000元需要短验。 4. 用户选择“新绑定的快捷卡”则不需要再次授权，直接进入后续节点；  * 入口流程页面详述   该流程为用户选择“已绑定的代扣卡”进行首次支付的流程，即：使用“未授权”卡进行支付的流程；   * 入口页面     收银台的入口页面，即业务方页面。用户在业务方页面点击“还款”相应操作时，业务方生成订单并后端请求支付平台接口，支付平台生成“预支付订单”返回给业务方后端，业务方发起支付请求唤起支付平台收银台。出于安全性考虑，订单及用户银行卡信息应当避免通过前端传输。  该环节接口主要传递参数：业务订单号，业务订单时间，业务订单金额，业务商品名称，业务方编号，业务用户ID，支付用户ID，请求时间，回调地址，订单有效期，备注。 支付授权流程页面详述   该流程为用户选择“已绑定的代扣卡”进行首次支付的流程，即：使用“未授权”卡进行支付的流程；   * 收银台页面     收银台页面主要有两个状态：仅有一张代扣卡和已主动绑定过快捷卡。   * 规则说明：   用户点击“去支付”时，当该笔订单金额大于当前选中银行卡所支持的金额时，页面给出提示“已超出该银行卡限额，请使用其他银行卡支付”；  当用户存在多张银行卡时，按照该每张银行卡的单笔限额由大到小排列，限额相同时按照绑卡时间由近到远依次排列；   * 字段说明：  1. 账单名称：该笔订单名称，即业务商品名称，由业务方传入； 2. 待支付金融：该笔订单的金额，由业务方传入； 3. 单笔限额：该银行卡所支持快捷支付的单笔限额，取支付通道和支付平台的最小值； 4. 单日限额：该银行卡所支持快捷支付的单日限额，取支付通道和支付平台的最小值；  * 确认银行卡页     用户选择原绑定的代扣卡，进行首次支付时，支付平台底层更换了支付通道（由原代扣通道切换至快捷支付通道），需要用户对该银行卡重新在新的快捷支付通道绑卡授权。一期连连支付该环节不需要短信验证码校验，未来的快捷支付通道可能会有短信校验环节。   * 规则说明：  1. 为提升用户体验，该页面的要素信息有则反显； 2. 为了保证绑定银行卡为本人卡，该页面反显的要素信息仅预留手机号可以编辑修改，其余要素信息仅反显不可修改。为防止信息被篡改，其余要素信息由后端传输； 3. 用户需要勾选底部协议才可进行确认绑定； 4. 用户在提交信息前，前端页面对手机号信息进行有效性核验，不符合手机号规则时，前端给出相应的提示信息“您填写的手机号有误，请重新填写”； 5. 该页面的四要素信息，将有支付平台传输至底层支付公司进行核验；  * 字段说明：  1. 卡号：用户银行卡号； 2. 开户行：该银行卡所属银行； 3. 姓名：用户姓名； 4. 身份证号：用户身份证号； 5. 预留手机号：该银行卡在银行预留的手机号；  * 设置支付密码页     切入到支付场景中的设置支付密码页面。   * 规则说明：   6位数字密码，调起自有键盘；   1. 密码规则：不能为6位完全连续数字，不能为6位相同数字； 2. 用户在点击“确定”时，如果输入的密码不合法，前端给出相应的信息提示“您输入的密码格式有误，请重新输入”  * 确认支付密码   切入到支付场景中的确认支付密码页面。   * 规则说明：  1. 6位数字密码，调起自有键盘； 2. 用户在点击“确定”时，如果输入的密码不合法，前端给出相应的信息提示“您输入的密码格式有误，请重新输入”； 3. 用户在点击“确定”时，如果输入的密码与首次不一致，前端给出相应的信息提示“您两次输入的密码不一致，请重新输入”  * 输入短信验证码页     切入到支付场景中的输入短信验证码页面。   * 规则说明：  1. 该短信验证码由支付公司下发，发送条件可能未知且不可控； 2. 6位数字短信验证码，调起自有键盘，首条短验自动触发； 3. 重新获取时间间隔60s，该时间间隔由前端页面限制； 4. 短信验证码过期时间10min，该时效由支付公司限制； 5. 为防止恶意刷短信验证码，单位时间内获取次数限制由支付公司控制； 6. 用户在点击“确定”时，如果输入的短信验证码不合法或不正确，前端给出相应的信息提示“验证码不正确，请重新输入”； 7. 用户在点击“取消”时，隐藏蒙层，停留在当前页，即：“确认银行卡”页；  * 支付结果页     切入到支付场景中的支付等待和支付结果页。   * 规则说明：  1. 通常等待10s内能获得确切的支付结果； 2. 等待页面为在当前页面上蒙层等待，具体以设计稿为准；  支付流程页面详述   该流程为用户选择“已绑定的代扣卡”进行再次支付的流程，或用户选择“新绑定的快捷卡”进行支付的流程，即：使用“已授权”卡进行支付的流程；   * 收银台页面   同2.3.1；   * 输入支付密码     收银台的入口页面，即业务方页面。用户在业务方页面点击“还款”相应操作时，唤起支付平台收银台。   * 规则说明：  1. 6位数字密码，调起自有键盘； 2. 用户在点击“确定支付”时，如果输入的密码不合法，前端给出相应的信息提示“您输入的密码格式有误，请重新输入”； 3. 用户在点击“确定支付”时，如果输入的密码不正确，前端给出相应的信息提示“您输入的密码有误，请重新输入”  * 输入短信验证码页   校验短信验证码，同2.3.5；用户在支付时，短信验证码流程可能会有，可能没有；   * 支付结果页   支付等待及结果页同2.3.6； 路由中心目的 1.银行覆盖度，支持更多的银行；2.时效性，不同的支付通道支付时效性有差异，如代付是否实时到账；3.节省成本，每笔支付均有通道手续费，针对不同的订单选择最优的支付通道；4.容灾备份，保证系统稳定。简言之，基于已经接入的通道，选择合适的通道，尽最大能力、最优通道完成支付操作。  最大能力是指，只要已接入的支付通道能够完成该订单，那就一定要完成支付；最优通道是指，在所有已接入的可用通道中，选取最稳定、成本最小的通道进行支付。  基于市场上裸代扣产品受监管限制，在保证用户体验的情况下，扩展更多系统扣款通道。 路由说明 支付产品包装关系，决定了支付路由的设计，支付平台包装的支付产品：   1. 付款产品，底层主要使用第三方支付公司的代付产品、银企直连通道； 2. 收款产品，底层主要使用第三方支付公司的代扣产品和协议支付产品； 3. 收银台产品，底层主要使用第三方支付公司的协议支付和快捷支付产品；    路由流程 业务方发起鉴权绑卡请求，通过“安全机制”和“支付风控”校验后，按照支付平台的路由策略执行：  1.支付平台先执行“决策器”。“决策器”根据“决策规则”筛选出针对该笔订单的“最优通道”；  2.调起“执行器”。“执行器”根据“路由排序表”选择当前条件下的最优通道，通过该最优通道发起鉴权签约。  3.定时调起“查询器”。“查询器”定时对执行结果进行扫描，当收到确切的“绑卡成功”或“绑卡失败”结果时，将结果置为终态并记录通道成功率，通知业务方，完毕。 鉴权签约路由 此处的鉴权签约主要是指“协议支付鉴权签约api”，不包含收银台流程中的绑卡流程。如下图所示，所有支付通道代扣支持的银行的全集为鉴权路由所支持的银行卡列表。   目的 鉴权签约是支付的前提条件，为后续支付环节路由埋下伏笔。 决策因子  1. 支持卡行，支持当前订单的银行的支付通道； 2. 支持卡种，根据业务属性，当订单银行卡为信用卡时，支持信用卡的支付通道； 3. 交易限额，单笔限额的大小，支付公司下的银行层； 4. 综合指数，根据商务合作中的流量分配，支付公司的服务情况、技术成熟度、业内口碑、近期重大事件等因素人工预留优先级，该参数是支付公司下的“银行层”； 5. 时效性，根据支付（收款）后信息流回盘时间、资金流实际到账时间，对支付公司的到账时效性，分1、2、3级，支付公司下的银行层； 6. 成本费用，该订单在各个支付通道的手续费用，不区分支付通道下的各个银行； 7. 通道的QOS，失败率、掉单率，不区分支付通道下的各个银行； 8. 系统响应，网络响应时间等，不区分支付通道下的各个银行；  决策规则 优先级规则：可用通道（支持卡行、卡种账户类型、限额）>>综合指数>时效性>成本费用>通道稳定性>响应时间。  鉴权签约支持的银行，是针对该订单在当前状态下所有“可用通道”支持银行的合集。  当某个银行有多个可用通道时，依次按照综合情况、时效性、成本费用、通道稳定性、响应时间的优先级，筛选出最优的通道。 示例说明  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 通道 | 可用性 | | | 优先级 | | | | | | 是否支持该行 | 是否支持该卡种 | 该行单笔限额 | 该行综合指数 | 该行时效性 | 通道成本费用 | QOS | 系统响应 | | 甲 | 是 | 是 | 5000 | 0 | 1 | 2元/笔 |  |  | | 乙 | 是 | 是 | 10000 | 0 | 1 | 0.2% |  |  | | 丙 | 是 | 是 | 10000 | 0 | 1 | 3元/笔 |  |  | | 丁 | 是 | 是 | 10000 | 0 | 2 | 2.5元/笔 |  |  | | 戊 | 否 | - | - | 0 | - |  |  |  |   举例说明：如上表所示，订单a为招商银行借记卡，订单金额6000元，有甲、乙、丙、丁4家支付通道支持招商银行借记卡，且4家支付通道的招商银行单笔限额分别为5000元、10000元、10000元、10000元，手续费率分别为2元/笔、3元/笔、比例计费0.2%、2.5元/笔。  这时，在不拆单的情况下，针对该订单的可用通道有乙、丙、丁3家，而后依次按照综合指数、时效性、成本费用等优先级筛选出最优的通道。“综合指数”为默认值，则在该环节各个通道的优先级一致，优选“乙、丙、丁通道”；接着乙、丙、丁“时效性”等级分别为1、1、2，则在“时效性”过滤环节优选“乙、丙通道”；由于丙通道相对于乙通道成本更低，则在成本环节优选“丙通道”； 付款路由目的 容灾备份，系统稳定，银行覆盖度，节省成本是支付路由选择支付通道的最主要的规则。 在可用的通道范围内，优先选取最稳定、最省钱的通道。  提升支付产品的QOS，这体现在系统的可靠性、稳定性、性能和可用性上，通过屏蔽掉无法连接、不稳定、性能低的通道来提升这些指标；降低运营成本，一个设计良好的支付路由，可以大大降低运营投入。 决策因子  1. 支持卡行，支持的银行列表； 2. 支持卡种，根据业务属性，当订单银行卡为信用卡时，支持信用卡的支付通道； 3. 交易限额，单笔限额，超过这个额度，需要变换通道（本期不拆单）； 4. 备付金余额，业务方在支付渠道的备付金余额； 5. 账户类型，对公、对私； 6. 综合指数，根据商务合作中的流量分配，支付公司的服务情况、技术成熟度、业内口碑、近期重大事件等因素人工预留优先级，该参数是支付公司下的“银行卡层”； 7. 到账时效，实时、非实时或在某时间段实时（结合发起时间，有通道区分工作时间和非工作时间），分1、2、3级，支付公司下的银行层； 8. 成本费用，该订单在各个支付通道的手续费用，不区分支付通道下的各个银行；； 9. 通道的QOS，掉单率，不区分支付通道下的各个银行；； 10. 网络响应时间延迟信息，不区分支付通道下的各个银行；；  决策规则 优先级规则：通道可用（支持卡行、卡种、限额、备付金可用余额、账户类型）>>综合指数>时效>成本费用（成本、活动优惠）>通道稳定性>响应时间。 路由流程图  扣款路由目的 路由的主要目的：容灾备份，系统稳定，银行覆盖度，成本最低，体验最优。即： 在符合业务要求的通道范围内，优先选取最稳定、最省钱、体验最优的支付通道。  提升支付产品的QOS，这体现在系统的可靠性、稳定性、性能和可用性上，通过屏蔽掉无法连接、不稳定、性能低的通道来提升这些指标；降低运营成本，一个设计良好的支付路由，可以大大降低运营投入。 决策因子  1. 支持卡行，支持的银行列表； 2. 支持卡种，根据业务属性，当订单银行卡为信用卡时，支持信用卡的支付通道； 3. 交易限额，单笔限额，超过这个额度，需要变换通道（本期不拆单、不考虑单日单月） 4. 账户类型，对公、对私； 5. 综合指数，根据商务合作中的流量分配，支付公司的服务情况、技术成熟度、业内口碑、近期重大事件等因素人工预留优先级，该参数是支付公司下的“银行卡层”； 6. 到账时效，到账时效，扣款通常都是实时的； 7. 成本费用，该订单在各个支付通道的手续费用，一般为两种计费方式：单笔计费、按比例计费，不区分支付通道下的各个银行； 8. 通道的QOS，掉单率，不区分支付通道下的各个银行； 9. 网络响应时间延迟信息，不区分支付通道下的各个银行；  决策规则 优先级规则：通道可用（支持卡行、可用限额、卡类型、账户类型）>>综合指数>时效>成本费用（成本、活动优惠）>通道稳定性>响应时间。 路由流程图  收银台支付路由目的 1.为系统扣款扩展更多通道；2.银行覆盖度，支持更多的银行；3.节省成本，每笔支付均有通道手续费，针对不同的订单选择最优的支付通道；4.容灾备份，保证系统稳定。简言之，基于已经接入的通道，选择合适的通道，尽最大能力、最优通道完成支付操作。 决策因子  1. 支持卡行，支持的银行列表； 2. 支持卡种，根据业务属性，当订单银行卡为信用卡时，支持信用卡的支付通道； 3. 交易限额，单笔限额，超过这个额度，需要变换通道（本期不拆单、不考虑单日单月）； 4. 账户类型，对公、对私； 5. 综合指数，根据商务合作中的流量分配，支付公司的服务情况、技术成熟度、业内口碑、近期重大事件等因素人工预留优先级，该参数是支付公司下的“银行卡层”； 6. 到账时效，到账时效，扣款通常都是实时的； 7. 成本费用，该订单在各个支付通道的手续费用，一般为两种计费方式：单笔计费、按比例计费，不区分支付通道下的各个银行； 8. 通道的QOS，掉单率，不区分支付通道下的各个银行； 9. 网络响应时间延迟信息，不区分支付通道下的各个银行；  决策规则 当默认系统扣款卡开通的协议支付通道数小于3个时，用户在主动支付选择该默认卡进行主动支付时需要扩展新通道，让用户输入短信验证码完成签约，同时系统执行扣款；当系统默认扣款卡开通的协议支付通道数达到3个后，用户在主动支付选择该默认卡时时按照路由规则进行支付。  已有签约通道小于3个时，扩展新协议支付通道规则：在剩余未签约通道中，依照扣款扣款路由的优先级规则，进行签约+支付；  已有签约通道达到3个之后，扣款优先级规则：通道可用（支持卡行、可用限额、卡类型、账户类型）>>综合指数>时效>成本费用（成本、活动优惠）>通道稳定性>响应时间；  用户自主添加新的银行卡，签约通道优先级规则：通道可用（支持卡行、可用限额、卡类型、账户类型）>>时效>成本费用（成本、活动优惠）>通道稳定性>响应时间。 解决的主要问题 统一管理支付通道，更稳定可靠；  释放各业务人员开发时间。  解决目前业务方单一支付通道的痛点。  解决目前业务方手工对账的痛点。 研究内容和目标研究方法 支付系统是提供完善的支付服务平台，各个业务线的接入同意通过支付系统进行服务，由接口服务负责支付的相关操作， 如用户的鉴权、用户的提现，用户的扣款，用户支付密码，用户绑卡列表等：所有通过支付相关的功能对应的银行，支付系统会不断完善，补充做一个完善的支付平台；服务前期是通过内部服务进行完成调用，业务方请求数据到支付平台， 我们会对用户信息进行数据校验， 防止一些用户不正确的行为进行屏蔽过滤，并进行合理的扣款付款等服务 技术路线  1. 用户可以通过绑定银行卡进行鉴权； 2. 可以对用户的银行卡进行付款包括对公，对私，小银行等； 3. 代扣用户的款项的功能，进行自动还款； 4. 设置安全的支付密码； 5. 设计的性能指标； 6. 支持高并发，平均响应时间10秒以内； 7. 整个系统性能稳定，可以满足全公司支付的使用需要； 8. 所有支付的请求、展现的服务器监控，确保在点击异常和展示异常方面可以做到立即通知和及时修复； 9. 所有增删改操作需记录日志（操作时间+操作人+操作类型）；  可行性分析  * 技术问题：   dubbo服务  dubbo是一个分布式服务框架，一种中间层服务框架，其具有高可靠性、高性能、高并发等特性，支持跨平台，同时支持事件驱动、多协议、异步等功能，是公司整体架构的重要组成部分。  dubbo主要由4部分组成：  dubbo Server：服务容器，用于宿主开发人员所开发的dubbo服务，接收和处理来自客户端的请求。  dubbo Client：客户端，调用者就像在调用本地接口一样方便，同时还提供了负载均衡、容错等机制。客户端现支持 Java等主流平台。  dubbo 序列化：序列化支持多种序列号， dubbo序列化、 hessian2序列化、 json序列化、 java序列化等跨平台的二进制序列化解决方案。  dubbo Protocol：协议，分为传输协议和数据协议。传输协议用于进行数据传输，数据协议用于请求和响应结果的内部数据。  其架构图如图4所示：  dubbo Client：  负载均衡：该模块提供了对服务集群的均衡调度，坏节点自动移除检测等功能。通过请求队列长度值来平衡服务节点的权重，以及动态计算的请求超时时间（队列越长超时时间越短），避免了服务被压死或调用者被堵死的情况。  动态代理：该模块负责创建客户端的服务代理，拦截调用请求，将请求转换成 dubbo 协议通过网络通讯模块发送出去，同时将响应结果返回值返回给调用者。  网络通讯：该模块负责发送请求协议包和接受来自服务端的响应，该模块通过Session ID和等待窗口的机制实现了服务的异步调用。  dubbo 序列化（跨平台二进制序列化）：该组件的目标是高效、跨平台，序列化出来结果的字节少，使用简单。不同语言平台遵循该结构进行序列化和反序列化，进而实现跨平台。  dubbo Protocol（通讯协议）：分为传输协议和数据协议。传输协议用于进行数据传输，数据协议用于请求和响应结果的内容数据。  dubbo Server：  网络通讯：负责监听和接收来自客户端的请求，将收到请求包交给工作线程去处理。  热部署：负责服务启动时jar和class的加载，当jar有变更时实现服务不中断的热部署。  服务代理：在服务启动时生成服务的静态代理，每次请求都是先调用服务代理，由代理去直接调用真实服务，避免了反射调用对性能的影响。  权限控制：SCF提供了两种机制，一是通过授权码的机制，每一个授权码都对应着一个或者几个方法，只有拥有该授权码的调用者才能调用这些方法；一是通过IP黑白名单来控制只有哪些IP能访问或哪些IP不能访问。  服务管理：通过Telnet到指定的端口，可以对服务运行情况实时监控，例如：time方法每次的执行时间，参数，调用者IP，count每秒并发数，control对服务进行控制等。有了这些功能可以对服务进行实时的监控，分析服务的运行情况，当服务出现问题时可以快速定位问题。  因此 dubbo框架具有高性能、跨平台、便捷开发、高扩展性、安全性等特性。  高性能：客户端和服务端特定的通讯模型，序列化组件对序列化和反序列化性能以及结果字节数组大小的严格控制，通讯协议的针对性设计等使得 dubbo 比.net平台的 WCF、Java平台的EJB、RMI，跨平台的Web Service等性能都要好。在追求性能的同时支持客户端的HA (High Availability)，容错机制为服务提供了良好的可靠性保证。  跨平台：由于特定系统的需要，会使用特定的语言或平台来开发，这么一来会存在多个平台的情况，比如有 java、.net、c++等，想要实现这些系统的简单高效的跨平台调用是一个非常麻烦的事情，dubbo 通过为不同平台提供不同的客户端来实现跨平台，目前已有 java、.net、c++等平台的客户端。服务开发人员在开发服务时无需关心这些细节，所开发出来的服务就能很好的支持跨平台调用。  便捷开发：这是dubbo设计的主要目标之一，开发人员不需要了解网络通讯，不需要关心如何跨平台，不需要知道如何塔建一个服务集群以及怎么做HA，这些框架都做了封装，需要做的只是定义一个接口，实现接口方法，标注注解。可以便捷的使一般程序员能够开发出优秀程序员一样高效、安全、稳定的服务。对于服务调用者也是非常简单的，所有服务开发人员不需要了解的东西调用者同样也不需要了解，对调用者来说调用远程服务的方法就像是调用本地方法一样。  高扩展性：当服务有压力时可以随时加服务器，增加服务节点来分担压力，客户端的负载均衡模块可以根据服务结点的压力情况来调整各结点的权重并平衡压力。  安全性：dubbo可以对调用者进行权限授权，不同的调用者只能调用对他授权的方法，这对有些对外暴露的服务又想对访问进行授权将会非常有用。   * 系统架构：   支付平台2.0架构图 课题特色少有人研究 目前市场上已存在大量聚合支付平台，但是行业野蛮发展、鱼目混珠，其中不乏一些平台并不能满足监管政策。也即是，很少有针对聚合支付较好的研究论文。 经济效益 “聚合支付平台系统”担负着提升公司效益的作用，以某金融公司的聚合支付平台系统为例，给公司创造着经济效益：  提升业务效率：传统模式下，对于新的业务需要单独开发，不能灵活复用，造成项目上线周期长，错过市场时机；同时，一些小企业不具备开放能力，也因此上线支付能力遇到困难。  系统稳定性：由于多数企业开发能力、技术资源优先，通常系统的底层通道单一，也因此系统的稳定性过度依赖于底层单一通道；  释放人力资源：释放技术资源，避免重复工作、多次开发； 社会效益 “聚合支付平台系统”作为业务支付阶段的重要一环，承担着较大的社会责任。以某金融公司的聚合支付平台系统为例，主要具有以下社会意义：  提高了用户提现、还款时的体验，更加多途径、更加便利的实现提现与还款操作。有效减小了用户的操作成本，对降低用户逾期率有显著效果，让金融回归“普惠”的本质；  给合作商户提供更快速、便捷的入款途径，合理地避免了因账期问题导致的商户资金周转问题，为友商创造了良性的运营环境；  方便金融监管，通过支付平台对金融各业务线支付订单账务统一管理，统一对小贷各业务的交易数据管理，方便统一上报，有效降低了监管成本； 研究基础与工作条件软件工程知识 通过软件工程学科的学习，已掌握了软件工程的UML设计基本理论知识，并且已独立完成了多项应用的设计工作。目前已具备数据库设计和php、java开发的基本技术开发能力，灵活运用eclipse开发工具。熟悉如何从管理和技术两方面研究如何更好地开发和维护计算机软件。熟悉在软件生命周期全过程中应完成的任务的性质，在概念上可以把软件生命周期划分成问题定义、可行性研究、需求分析、概要设计、详细设计、编码和单元测试、综合测试以及维护等多个阶段工作。 工作基础 已先后在两家互联网公司实习获得了相关的行业知识。参与过某知名互联网公司钱包产品的设计，及某知名互联网公司金融业务的支付平台的设计工作。对于聚合支付平台系统的大致流程有了初步的尝试。通过在企业实际实习实践过程中，发现的企业内存在的一些问题和熟悉监管政策，建立了适用于当下最高效且符合监管的要求的设计方案，并积极的在现有行业现有关联产品的基础上加以创新。 开发环境 已自行租用windows2012云服务器，方便自己学习使用。 研究思路 通过查阅相关的文献材料，了解到国内外的支付行业存在的问题、行业现状和发展趋势。学习到了支付系统的相关理论和方法，制定了本课题的技术路线和研究方法。后续将进一步调研大量行业经验和阅读论文材料，进一步增加研究与设计的能力。 论文计划  1. 第一阶段：文献阅读，需求整理   2018年11月 - 2019年2月熟悉项目需求，了解相关技术，整理需求。   1. 第二阶段：系统设计   2019年2 月- 2019年3月 完成系统的概要设计和详细设计。  2019年4月 - 2019年5月 完成前端系统及后台管理平台的原型。实现基本支付功能，做为进一步研究的依据。   1. 第二阶段：系统的实现   2019年6月 - 2019年9月 实现聚合支付平台中的主要功能模块。  2019年10月 - 2019年10月 系统主要模块单元测试，以及模块功能的完善和代码优化。   1. 第二阶段：论文撰写   2019年10月 - 2019年11月 完成论文撰写，提交学术论文。   1. 第三阶段：文档总结   2019年11月 - 2020年3月 总结并完成毕业论文。 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **三、研究方案设计及可行性分析（包括：研究方法，技术路线，理论分析、计算、实验方法和步骤及其可行性等）（不少于800字）**  **北 京 邮 电 大 学**  **研 究 生 论 文 开 题 报 告**  学 号 2017180215  姓 名 李超  申请学位 工学硕士  专业名称 软件工程  研究方向 企业信息化  导师姓名 崔毅东  所在院系 软件学院  报告日期： 2018 年 12 月 3 日  目录  [1. 立题依据 4](#_Toc11351)  [1.1. 研究目的 4](#_Toc24767)  [1.2. 研究意义 4](#_Toc563)  [1.2.1. 社会效益 5](#_Toc12606)  [1.2.2. 经济效益 5](#_Toc15251)  [1.3. 现状 5](#_Toc8483)  [2. 研究内容和目标 5](#_Toc16890)  [2.1. 研究目标和效果 5](#_Toc3411)  [2.2. 鉴权签约 6](#_Toc31325)  [2.2.1. 目的 6](#_Toc31729)  [2.2.2. 需求描述 6](#_Toc15184)  [2.2.3. 流程图 7](#_Toc5714)  [2.3. 代付 8](#_Toc18845)  [2.3.1. 目的 8](#_Toc29629)  [2.3.2. 需求描述 8](#_Toc13960)  [2.3.3. 流程图 9](#_Toc17174)  [2.4. 扣款产品 10](#_Toc5458)  [2.4.1. 目的 10](#_Toc4826)  [2.4.2. 需求描述 10](#_Toc27066)  [2.4.3. 流程图 10](#_Toc12833)  [2.1. 收银台需求 12](#_Toc32646)  [2.1.1. 总体流程 12](#_Toc14971)  [2.1.2. 支付授权流程页面详述 13](#_Toc7006)  [2.1.3. 支付流程页面详述 18](#_Toc4577)  [2.2. 路由中心 20](#_Toc14210)  [2.2.1. 目的 20](#_Toc27928)  [2.2.2. 路由说明 20](#_Toc21274)  [2.2.3. 路由流程 21](#_Toc24114)  [2.3. 鉴权签约路由 21](#_Toc20798)  [2.3.1. 目的 21](#_Toc6163)  [2.3.1. 决策因子 21](#_Toc3289)  [2.3.2. 决策规则 22](#_Toc5925)  [2.3.3. 示例说明 22](#_Toc12571)  [2.4. 付款路由 23](#_Toc8348)  [2.4.1. 目的 23](#_Toc8733)  [2.4.2. 决策因子 23](#_Toc21759)  [2.4.3. 决策规则 24](#_Toc24532)  [2.4.4. 路由流程图 24](#_Toc14290)  [2.5. 扣款路由 25](#_Toc23958)  [2.5.1. 目的 25](#_Toc23941)  [2.5.2. 决策因子 25](#_Toc23218)  [2.5.3. 决策规则 25](#_Toc14890)  [2.5.4. 路由流程图 26](#_Toc27891)  [2.6. 收银台支付路由 26](#_Toc7799)  [2.6.1. 目的 26](#_Toc13312)  [2.6.2. 决策因子 27](#_Toc7701)  [2.6.3. 决策规则 27](#_Toc9464)  [2.7. 解决的主要问题 27](#_Toc12635)  [3. 研究内容和目标 28](#_Toc6079)  [3.1. 研究方法 28](#_Toc13789)  [3.2. 技术路线 28](#_Toc19409)  [3.3. 可行性分析 28](#_Toc19579)  [4. 课题特色 31](#_Toc2203)  [4.1. 少有人研究 31](#_Toc18640)  [4.2. 经济效益 31](#_Toc23974)  [4.3. 社会效益 32](#_Toc17713)  [5. 研究基础与工作条件 32](#_Toc26909)  [5.1. 软件工程知识 32](#_Toc18337)  [5.2. 工作基础 32](#_Toc32623)  [5.3. 开发环境 33](#_Toc1890)  [5.4. 研究思路 33](#_Toc9505)  [6. 论文计划 33](#_Toc8203) 立题依据研究目的 目前市场上存在大量电商平台和类电商平台，这些平台具有多样的支付结算需求，通过调研市场实际情况，总结起来主要有以下痛点：  行业合规性：目前市场上多数的企业没有持有第三方支付牌照。一方面，对于企业来讲，为了更好的为用户提供优质服务，针对用户的支付结算需求是必要的；另一方面，每年市场上都会存在平台跑路现象，为了避免企业设立资金池、捐款跑路，关于“二清”监管是非常有必要的。由此，支付的合规性问题，一直是行业的要解决的一个痛点。  释放人力资源：传统模式下，当同一个集团旗下具有多个业务线时，每个业务线会分别与外部支付公司进行开发对接，这样造成重复对接、严重浪费技术资源；同时，对于财务而言，由于需要管理的系统较多，也因此产生了很多的财务管理工作。  解决信息孤岛：传统模式下，当同一个集团旗下具有多个业务线时，每个业务线会分别与外部支付公司进行开发对接，每一条业务线的用户数据分别存储于各自的数据库，用户信息散落在各个业务系统，无法做到资源共享，最大化利用。对于用户而言，同一个集团或同一个APP下的不同业务线需要分别绑定银行卡，用户体验极其不好；对于企业而言，由于各个业务线的用户数据是割裂的，因此不能对用户的数据资源进行有效利用。  提升业务效率：传统模式下，对于新的业务需要单独开发，不能灵活复用，造成项目上线周期长，错过市场时机；同时，一些小企业不具备开放能力，也因此上线支付能力遇到困难。  系统稳定性：由于多数企业开发能力、技术资源优先，通常系统的底层通道单一，也因此系统的稳定性过度依赖于底层单一通道；  总结下来，“聚合支付平台系统”解决了以下问题：1.平台仅过信息流，不做资金流，从而解决了“二清”的合规性问题；2.释放技术资源，避免重复工作、多次开发；3.实现集团企业多业务信息共享；4.提升支付结算效率； 研究意义 “聚合支付平台系统”是解决企业合规的开展支付业务，在不同渠道重复对账、独立统计的弊端，实现收款、结算、数据统计的全面聚合，打造统一接口、SDK管理，实现一次接入享受聚合平台所有通道，极大的简化了多次对接支付通道的繁琐流程、释放了多次开发对接的技术资源。 社会效益 “聚合支付平台系统”作为业务支付阶段的重要一环，承担着较大的社会责任。以某金融公司的聚合支付平台系统为例，主要具有以下社会意义：   1. 提高了用户提现、还款时的体验，更加多途径、更加便利的实现提现与还款操作。有效减小了用户的操作成本，对降低用户逾期率有显著效果，让金融回归“普惠”的本质； 2. 给合作商户提供更快速、便捷的入款途径，合理地避免了因账期问题导致的商户资金周转问题，为友商创造了良性的运营环境； 3. 方便金融监管，通过支付平台对金融各业务线支付订单账务统一管理，统一对小贷各业务的交易数据管理，方便统一上报，有效降低了监管成本；  经济效益 “聚合支付平台系统”担负着提升公司效益的作用，以某金融公司的聚合支付平台系统为例，给公司创造着经济效益：   1. 提高用户还款成功率： 2. 用户多种途径还款，显著降低了用户的逾期率； 3. 多层多支付通道自动路由切换，智能选择成本最低的支付通道，有效降低了金融各业务的运营成本； 4. 优秀的用户体验，对用户留存做出了巨大贡献；  现状 目前市场上聚合支付平台近百家，行业野蛮发展，缺乏行业规范。其中一些平台并不能很好的满足监管的要求。 研究内容和目标研究目标和效果 支付通道的统一管理，包括信息与对接的管理，避免重复对接，释放资源，更好的支持各业务线的增量需求，支持新增业务线的快速对接，统一放支付通道的管理及使用。智能路由建立，根据费率、支付成功率，实现支付通道的用户无感知自动切换，同时起到容灾备份的作用。  项目主要功能：   1. 鉴权绑卡：用于核验用户提供的信息一致性，以及对用户审核进行核验； 2. 鉴权签约：用户通过请求并回填短信验证码与平台签约，授权平台进行扣款； 3. 代扣收款：根据与用户的约定，系统可以根据业务需要，对客户进行扣款； 4. 协议支付收款：用户完成短信签约后，系统可以根据业务需要，对客户进行扣款； 5. 收银台产品：具备前端页面，集优惠券营销、多种支付方式并存的支付方式，主要用于用户主动完成支付行为； 6. 代付产品：商户将自有资金结算给用户，如：现金贷业务，用户的提现操作； 7. 支付路由配置：根据业务需求，自动化路由最有的支付通道； 8. 后台管理：用于管理支付产品管理、业务线管理、用户管理、对账管理等；  鉴权签约目的 通过绑卡的形式进行用户信息一致性校验、鉴权与签约。免去后续支付时输入四要素。 需求描述  1. 鉴权绑卡规则：   在需要使用代收付的业务场景下，业务方发起银行卡绑定关系建立请求到支付平台，支付平台接收到交易请求后，在进行报文合法性和业务合法性校验后，建立请求到支付公司，接收支付公司返回结果后，存储订单信息并向业务方返回绑定交易的状态信息。  鉴权成功后，根据用户身份证号生成唯一支付用户ID，留存用户信息项目：姓名、身份证号、银行卡号、预留手机号、业务方编号、业务用户ID等。用户绑定的银行卡继承在业务方，即：各个业务方下的用户银行卡列表，是支付用户银行卡列表的子集。   1. 鉴权方式：   有短信验证，支付公司提供的需要用户短信授权的鉴权方式，有短验证需要配合使用“绑卡确认”、“发送短信验证码”接口；  无短信验证，支付公司提供的无需用户短信授权的鉴权方式，无短验证不需要使用“绑卡确认”、“发送短信验证码”接口。业务方可根据需要选择“有短”或“无短”鉴权方式，目前项目以及合作支付通道，均采用“无短”模式。   1. 绑定卡种选择：   如果绑定的是借记卡，请求报文主要包括：商户号（业务方）、商户订单号、业务用户ID、业务类型、卡用途、渠道编码、银行编码、卡号、卡种、证件号、证件类型、姓名、预留手机号；  如果绑定的是信用卡，求情报文主要包括：商户号（业务方）、商户订单号、业务用户ID、业务类型、卡用途、渠道编码、银行编码、卡号、卡种、证件号、证件类型、姓名、预留手机号、信用卡有限期、CVV2号；目前项目不支持信用卡。   1. 多渠道下绑卡规则：   用户的某个银行卡在进行首次鉴权时，为了节省成本，仅鉴权（认证绑卡）主通道。在执行支付路由，当第N个通道不能使用时，自动路由到第（N+1）个通道，如果第（N+1）个通道下该银行卡尚未进行鉴权，调起平台包装的“鉴权+代扣”组合产品，依次类推。 流程图  代付目的 完成资金出款，如用户的提现操作、给商户返佣的操作。 需求描述 由业务方发起交易请求，支付平台在通过报文合法性和业务合法性校验后，向支付公司发起的代付交易请求，银行渠道负责验证由支付公司发送的卡信息相关数据，通过风控规则后，进行资金的出款。  当同一个业务方多个支付通道时，支付平台负责筛选当前订单下的可用通道，并计算各可用支付通道成本费用和通过率（也可由业务方指定由哪一个支付通道出款），做出路由决策。（参见支付路由章节）  重要说明：同一支付订单号，保证出款成功一次。 流程图  扣款产品目的 完成资金入款。应用场景，如：现金贷用户主动操作还款、系统发起还款。 需求描述 由业务方发起交易请求，支付平台在通过报文合法性和业务合法性校验后，向支付公司发起的代付交易请求，银行渠道负责验证由支付公司发送的卡信息相关数据，通过风控规则后，进行资金的扣款。  当同一个业务方多个支付通道时，支付平台负责筛选当前订单下的可用通道，并计算各可用支付通道成本费用和通过率（也可由业务方指定由哪一个支付通道出款），做出路由决策。（参见支付路由章节）  重要说明：同一支付订单号，保证扣款成功一次。 流程图  收银台需求总体流程  * 流程图      * 重要节点说明  1. 当用户选择进件时“原绑定的代扣卡”时进行支付时，首先判定该卡是否已在快捷支付通道授权。如果没有授权，则进入填写四要素页面，且原代扣卡信息反显，待用户确认授权（为保证安全，仅手机号可以编辑修改）；如果已经授权，则进入下一个节点。 2. 是否已设置过密码，该支付用户是否已设置过支付密码。 3. 是否需要短验，底层支付公司对该订单是否需要进行短信验证码校验。一期连连快捷支付短验规则：首次支付一定需要短验；再次支付时，当订单金额大于5000元需要短验。 4. 用户选择“新绑定的快捷卡”则不需要再次授权，直接进入后续节点；  * 入口流程页面详述   该流程为用户选择“已绑定的代扣卡”进行首次支付的流程，即：使用“未授权”卡进行支付的流程；   * 入口页面     收银台的入口页面，即业务方页面。用户在业务方页面点击“还款”相应操作时，业务方生成订单并后端请求支付平台接口，支付平台生成“预支付订单”返回给业务方后端，业务方发起支付请求唤起支付平台收银台。出于安全性考虑，订单及用户银行卡信息应当避免通过前端传输。  该环节接口主要传递参数：业务订单号，业务订单时间，业务订单金额，业务商品名称，业务方编号，业务用户ID，支付用户ID，请求时间，回调地址，订单有效期，备注。 支付授权流程页面详述   该流程为用户选择“已绑定的代扣卡”进行首次支付的流程，即：使用“未授权”卡进行支付的流程；   * 收银台页面     收银台页面主要有两个状态：仅有一张代扣卡和已主动绑定过快捷卡。   * 规则说明：   用户点击“去支付”时，当该笔订单金额大于当前选中银行卡所支持的金额时，页面给出提示“已超出该银行卡限额，请使用其他银行卡支付”；  当用户存在多张银行卡时，按照该每张银行卡的单笔限额由大到小排列，限额相同时按照绑卡时间由近到远依次排列；   * 字段说明：  1. 账单名称：该笔订单名称，即业务商品名称，由业务方传入； 2. 待支付金融：该笔订单的金额，由业务方传入； 3. 单笔限额：该银行卡所支持快捷支付的单笔限额，取支付通道和支付平台的最小值； 4. 单日限额：该银行卡所支持快捷支付的单日限额，取支付通道和支付平台的最小值；  * 确认银行卡页     用户选择原绑定的代扣卡，进行首次支付时，支付平台底层更换了支付通道（由原代扣通道切换至快捷支付通道），需要用户对该银行卡重新在新的快捷支付通道绑卡授权。一期连连支付该环节不需要短信验证码校验，未来的快捷支付通道可能会有短信校验环节。   * 规则说明：  1. 为提升用户体验，该页面的要素信息有则反显； 2. 为了保证绑定银行卡为本人卡，该页面反显的要素信息仅预留手机号可以编辑修改，其余要素信息仅反显不可修改。为防止信息被篡改，其余要素信息由后端传输； 3. 用户需要勾选底部协议才可进行确认绑定； 4. 用户在提交信息前，前端页面对手机号信息进行有效性核验，不符合手机号规则时，前端给出相应的提示信息“您填写的手机号有误，请重新填写”； 5. 该页面的四要素信息，将有支付平台传输至底层支付公司进行核验；  * 字段说明：  1. 卡号：用户银行卡号； 2. 开户行：该银行卡所属银行； 3. 姓名：用户姓名； 4. 身份证号：用户身份证号； 5. 预留手机号：该银行卡在银行预留的手机号；  * 设置支付密码页     切入到支付场景中的设置支付密码页面。   * 规则说明：   6位数字密码，调起自有键盘；   1. 密码规则：不能为6位完全连续数字，不能为6位相同数字； 2. 用户在点击“确定”时，如果输入的密码不合法，前端给出相应的信息提示“您输入的密码格式有误，请重新输入”  * 确认支付密码   切入到支付场景中的确认支付密码页面。   * 规则说明：  1. 6位数字密码，调起自有键盘； 2. 用户在点击“确定”时，如果输入的密码不合法，前端给出相应的信息提示“您输入的密码格式有误，请重新输入”； 3. 用户在点击“确定”时，如果输入的密码与首次不一致，前端给出相应的信息提示“您两次输入的密码不一致，请重新输入”  * 输入短信验证码页     切入到支付场景中的输入短信验证码页面。   * 规则说明：  1. 该短信验证码由支付公司下发，发送条件可能未知且不可控； 2. 6位数字短信验证码，调起自有键盘，首条短验自动触发； 3. 重新获取时间间隔60s，该时间间隔由前端页面限制； 4. 短信验证码过期时间10min，该时效由支付公司限制； 5. 为防止恶意刷短信验证码，单位时间内获取次数限制由支付公司控制； 6. 用户在点击“确定”时，如果输入的短信验证码不合法或不正确，前端给出相应的信息提示“验证码不正确，请重新输入”； 7. 用户在点击“取消”时，隐藏蒙层，停留在当前页，即：“确认银行卡”页；  * 支付结果页     切入到支付场景中的支付等待和支付结果页。   * 规则说明：  1. 通常等待10s内能获得确切的支付结果； 2. 等待页面为在当前页面上蒙层等待，具体以设计稿为准；  支付流程页面详述   该流程为用户选择“已绑定的代扣卡”进行再次支付的流程，或用户选择“新绑定的快捷卡”进行支付的流程，即：使用“已授权”卡进行支付的流程；   * 收银台页面   同2.3.1；   * 输入支付密码     收银台的入口页面，即业务方页面。用户在业务方页面点击“还款”相应操作时，唤起支付平台收银台。   * 规则说明：  1. 6位数字密码，调起自有键盘； 2. 用户在点击“确定支付”时，如果输入的密码不合法，前端给出相应的信息提示“您输入的密码格式有误，请重新输入”； 3. 用户在点击“确定支付”时，如果输入的密码不正确，前端给出相应的信息提示“您输入的密码有误，请重新输入”  * 输入短信验证码页   校验短信验证码，同2.3.5；用户在支付时，短信验证码流程可能会有，可能没有；   * 支付结果页   支付等待及结果页同2.3.6； 路由中心目的 1.银行覆盖度，支持更多的银行；2.时效性，不同的支付通道支付时效性有差异，如代付是否实时到账；3.节省成本，每笔支付均有通道手续费，针对不同的订单选择最优的支付通道；4.容灾备份，保证系统稳定。简言之，基于已经接入的通道，选择合适的通道，尽最大能力、最优通道完成支付操作。  最大能力是指，只要已接入的支付通道能够完成该订单，那就一定要完成支付；最优通道是指，在所有已接入的可用通道中，选取最稳定、成本最小的通道进行支付。  基于市场上裸代扣产品受监管限制，在保证用户体验的情况下，扩展更多系统扣款通道。 路由说明 支付产品包装关系，决定了支付路由的设计，支付平台包装的支付产品：   1. 付款产品，底层主要使用第三方支付公司的代付产品、银企直连通道； 2. 收款产品，底层主要使用第三方支付公司的代扣产品和协议支付产品； 3. 收银台产品，底层主要使用第三方支付公司的协议支付和快捷支付产品；    路由流程 业务方发起鉴权绑卡请求，通过“安全机制”和“支付风控”校验后，按照支付平台的路由策略执行：  1.支付平台先执行“决策器”。“决策器”根据“决策规则”筛选出针对该笔订单的“最优通道”；  2.调起“执行器”。“执行器”根据“路由排序表”选择当前条件下的最优通道，通过该最优通道发起鉴权签约。  3.定时调起“查询器”。“查询器”定时对执行结果进行扫描，当收到确切的“绑卡成功”或“绑卡失败”结果时，将结果置为终态并记录通道成功率，通知业务方，完毕。 鉴权签约路由 此处的鉴权签约主要是指“协议支付鉴权签约api”，不包含收银台流程中的绑卡流程。如下图所示，所有支付通道代扣支持的银行的全集为鉴权路由所支持的银行卡列表。   目的 鉴权签约是支付的前提条件，为后续支付环节路由埋下伏笔。 决策因子  1. 支持卡行，支持当前订单的银行的支付通道； 2. 支持卡种，根据业务属性，当订单银行卡为信用卡时，支持信用卡的支付通道； 3. 交易限额，单笔限额的大小，支付公司下的银行层； 4. 综合指数，根据商务合作中的流量分配，支付公司的服务情况、技术成熟度、业内口碑、近期重大事件等因素人工预留优先级，该参数是支付公司下的“银行层”； 5. 时效性，根据支付（收款）后信息流回盘时间、资金流实际到账时间，对支付公司的到账时效性，分1、2、3级，支付公司下的银行层； 6. 成本费用，该订单在各个支付通道的手续费用，不区分支付通道下的各个银行； 7. 通道的QOS，失败率、掉单率，不区分支付通道下的各个银行； 8. 系统响应，网络响应时间等，不区分支付通道下的各个银行；  决策规则 优先级规则：可用通道（支持卡行、卡种账户类型、限额）>>综合指数>时效性>成本费用>通道稳定性>响应时间。  鉴权签约支持的银行，是针对该订单在当前状态下所有“可用通道”支持银行的合集。  当某个银行有多个可用通道时，依次按照综合情况、时效性、成本费用、通道稳定性、响应时间的优先级，筛选出最优的通道。 示例说明  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | | 通道 | 可用性 | | | 优先级 | | | | | | 是否支持该行 | 是否支持该卡种 | 该行单笔限额 | 该行综合指数 | 该行时效性 | 通道成本费用 | QOS | 系统响应 | | 甲 | 是 | 是 | 5000 | 0 | 1 | 2元/笔 |  |  | | 乙 | 是 | 是 | 10000 | 0 | 1 | 0.2% |  |  | | 丙 | 是 | 是 | 10000 | 0 | 1 | 3元/笔 |  |  | | 丁 | 是 | 是 | 10000 | 0 | 2 | 2.5元/笔 |  |  | | 戊 | 否 | - | - | 0 | - |  |  |  |   举例说明：如上表所示，订单a为招商银行借记卡，订单金额6000元，有甲、乙、丙、丁4家支付通道支持招商银行借记卡，且4家支付通道的招商银行单笔限额分别为5000元、10000元、10000元、10000元，手续费率分别为2元/笔、3元/笔、比例计费0.2%、2.5元/笔。  这时，在不拆单的情况下，针对该订单的可用通道有乙、丙、丁3家，而后依次按照综合指数、时效性、成本费用等优先级筛选出最优的通道。“综合指数”为默认值，则在该环节各个通道的优先级一致，优选“乙、丙、丁通道”；接着乙、丙、丁“时效性”等级分别为1、1、2，则在“时效性”过滤环节优选“乙、丙通道”；由于丙通道相对于乙通道成本更低，则在成本环节优选“丙通道”； 付款路由目的 容灾备份，系统稳定，银行覆盖度，节省成本是支付路由选择支付通道的最主要的规则。 在可用的通道范围内，优先选取最稳定、最省钱的通道。  提升支付产品的QOS，这体现在系统的可靠性、稳定性、性能和可用性上，通过屏蔽掉无法连接、不稳定、性能低的通道来提升这些指标；降低运营成本，一个设计良好的支付路由，可以大大降低运营投入。 决策因子  1. 支持卡行，支持的银行列表； 2. 支持卡种，根据业务属性，当订单银行卡为信用卡时，支持信用卡的支付通道； 3. 交易限额，单笔限额，超过这个额度，需要变换通道（本期不拆单）； 4. 备付金余额，业务方在支付渠道的备付金余额； 5. 账户类型，对公、对私； 6. 综合指数，根据商务合作中的流量分配，支付公司的服务情况、技术成熟度、业内口碑、近期重大事件等因素人工预留优先级，该参数是支付公司下的“银行卡层”； 7. 到账时效，实时、非实时或在某时间段实时（结合发起时间，有通道区分工作时间和非工作时间），分1、2、3级，支付公司下的银行层； 8. 成本费用，该订单在各个支付通道的手续费用，不区分支付通道下的各个银行；； 9. 通道的QOS，掉单率，不区分支付通道下的各个银行；； 10. 网络响应时间延迟信息，不区分支付通道下的各个银行；；  决策规则 优先级规则：通道可用（支持卡行、卡种、限额、备付金可用余额、账户类型）>>综合指数>时效>成本费用（成本、活动优惠）>通道稳定性>响应时间。 路由流程图  扣款路由目的 路由的主要目的：容灾备份，系统稳定，银行覆盖度，成本最低，体验最优。即： 在符合业务要求的通道范围内，优先选取最稳定、最省钱、体验最优的支付通道。  提升支付产品的QOS，这体现在系统的可靠性、稳定性、性能和可用性上，通过屏蔽掉无法连接、不稳定、性能低的通道来提升这些指标；降低运营成本，一个设计良好的支付路由，可以大大降低运营投入。 决策因子  1. 支持卡行，支持的银行列表； 2. 支持卡种，根据业务属性，当订单银行卡为信用卡时，支持信用卡的支付通道； 3. 交易限额，单笔限额，超过这个额度，需要变换通道（本期不拆单、不考虑单日单月） 4. 账户类型，对公、对私； 5. 综合指数，根据商务合作中的流量分配，支付公司的服务情况、技术成熟度、业内口碑、近期重大事件等因素人工预留优先级，该参数是支付公司下的“银行卡层”； 6. 到账时效，到账时效，扣款通常都是实时的； 7. 成本费用，该订单在各个支付通道的手续费用，一般为两种计费方式：单笔计费、按比例计费，不区分支付通道下的各个银行； 8. 通道的QOS，掉单率，不区分支付通道下的各个银行； 9. 网络响应时间延迟信息，不区分支付通道下的各个银行；  决策规则 优先级规则：通道可用（支持卡行、可用限额、卡类型、账户类型）>>综合指数>时效>成本费用（成本、活动优惠）>通道稳定性>响应时间。 路由流程图  收银台支付路由目的 1.为系统扣款扩展更多通道；2.银行覆盖度，支持更多的银行；3.节省成本，每笔支付均有通道手续费，针对不同的订单选择最优的支付通道；4.容灾备份，保证系统稳定。简言之，基于已经接入的通道，选择合适的通道，尽最大能力、最优通道完成支付操作。 决策因子  1. 支持卡行，支持的银行列表； 2. 支持卡种，根据业务属性，当订单银行卡为信用卡时，支持信用卡的支付通道； 3. 交易限额，单笔限额，超过这个额度，需要变换通道（本期不拆单、不考虑单日单月）； 4. 账户类型，对公、对私； 5. 综合指数，根据商务合作中的流量分配，支付公司的服务情况、技术成熟度、业内口碑、近期重大事件等因素人工预留优先级，该参数是支付公司下的“银行卡层”； 6. 到账时效，到账时效，扣款通常都是实时的； 7. 成本费用，该订单在各个支付通道的手续费用，一般为两种计费方式：单笔计费、按比例计费，不区分支付通道下的各个银行； 8. 通道的QOS，掉单率，不区分支付通道下的各个银行； 9. 网络响应时间延迟信息，不区分支付通道下的各个银行；  决策规则 当默认系统扣款卡开通的协议支付通道数小于3个时，用户在主动支付选择该默认卡进行主动支付时需要扩展新通道，让用户输入短信验证码完成签约，同时系统执行扣款；当系统默认扣款卡开通的协议支付通道数达到3个后，用户在主动支付选择该默认卡时时按照路由规则进行支付。  已有签约通道小于3个时，扩展新协议支付通道规则：在剩余未签约通道中，依照扣款扣款路由的优先级规则，进行签约+支付；  已有签约通道达到3个之后，扣款优先级规则：通道可用（支持卡行、可用限额、卡类型、账户类型）>>综合指数>时效>成本费用（成本、活动优惠）>通道稳定性>响应时间；  用户自主添加新的银行卡，签约通道优先级规则：通道可用（支持卡行、可用限额、卡类型、账户类型）>>时效>成本费用（成本、活动优惠）>通道稳定性>响应时间。 解决的主要问题 统一管理支付通道，更稳定可靠；  释放各业务人员开发时间。  解决目前业务方单一支付通道的痛点。  解决目前业务方手工对账的痛点。 研究内容和目标研究方法 支付系统是提供完善的支付服务平台，各个业务线的接入同意通过支付系统进行服务，由接口服务负责支付的相关操作， 如用户的鉴权、用户的提现，用户的扣款，用户支付密码，用户绑卡列表等：所有通过支付相关的功能对应的银行，支付系统会不断完善，补充做一个完善的支付平台；服务前期是通过内部服务进行完成调用，业务方请求数据到支付平台， 我们会对用户信息进行数据校验， 防止一些用户不正确的行为进行屏蔽过滤，并进行合理的扣款付款等服务 技术路线  1. 用户可以通过绑定银行卡进行鉴权； 2. 可以对用户的银行卡进行付款包括对公，对私，小银行等； 3. 代扣用户的款项的功能，进行自动还款； 4. 设置安全的支付密码； 5. 设计的性能指标； 6. 支持高并发，平均响应时间10秒以内； 7. 整个系统性能稳定，可以满足全公司支付的使用需要； 8. 所有支付的请求、展现的服务器监控，确保在点击异常和展示异常方面可以做到立即通知和及时修复； 9. 所有增删改操作需记录日志（操作时间+操作人+操作类型）；  可行性分析  * 技术问题：   dubbo服务  dubbo是一个分布式服务框架，一种中间层服务框架，其具有高可靠性、高性能、高并发等特性，支持跨平台，同时支持事件驱动、多协议、异步等功能，是公司整体架构的重要组成部分。  dubbo主要由4部分组成：  dubbo Server：服务容器，用于宿主开发人员所开发的dubbo服务，接收和处理来自客户端的请求。  dubbo Client：客户端，调用者就像在调用本地接口一样方便，同时还提供了负载均衡、容错等机制。客户端现支持 Java等主流平台。  dubbo 序列化：序列化支持多种序列号， dubbo序列化、 hessian2序列化、 json序列化、 java序列化等跨平台的二进制序列化解决方案。  dubbo Protocol：协议，分为传输协议和数据协议。传输协议用于进行数据传输，数据协议用于请求和响应结果的内部数据。  其架构图如图4所示：  dubbo Client：  负载均衡：该模块提供了对服务集群的均衡调度，坏节点自动移除检测等功能。通过请求队列长度值来平衡服务节点的权重，以及动态计算的请求超时时间（队列越长超时时间越短），避免了服务被压死或调用者被堵死的情况。  动态代理：该模块负责创建客户端的服务代理，拦截调用请求，将请求转换成 dubbo 协议通过网络通讯模块发送出去，同时将响应结果返回值返回给调用者。  网络通讯：该模块负责发送请求协议包和接受来自服务端的响应，该模块通过Session ID和等待窗口的机制实现了服务的异步调用。  dubbo 序列化（跨平台二进制序列化）：该组件的目标是高效、跨平台，序列化出来结果的字节少，使用简单。不同语言平台遵循该结构进行序列化和反序列化，进而实现跨平台。  dubbo Protocol（通讯协议）：分为传输协议和数据协议。传输协议用于进行数据传输，数据协议用于请求和响应结果的内容数据。  dubbo Server：  网络通讯：负责监听和接收来自客户端的请求，将收到请求包交给工作线程去处理。  热部署：负责服务启动时jar和class的加载，当jar有变更时实现服务不中断的热部署。  服务代理：在服务启动时生成服务的静态代理，每次请求都是先调用服务代理，由代理去直接调用真实服务，避免了反射调用对性能的影响。  权限控制：SCF提供了两种机制，一是通过授权码的机制，每一个授权码都对应着一个或者几个方法，只有拥有该授权码的调用者才能调用这些方法；一是通过IP黑白名单来控制只有哪些IP能访问或哪些IP不能访问。  服务管理：通过Telnet到指定的端口，可以对服务运行情况实时监控，例如：time方法每次的执行时间，参数，调用者IP，count每秒并发数，control对服务进行控制等。有了这些功能可以对服务进行实时的监控，分析服务的运行情况，当服务出现问题时可以快速定位问题。  因此 dubbo框架具有高性能、跨平台、便捷开发、高扩展性、安全性等特性。  高性能：客户端和服务端特定的通讯模型，序列化组件对序列化和反序列化性能以及结果字节数组大小的严格控制，通讯协议的针对性设计等使得 dubbo 比.net平台的 WCF、Java平台的EJB、RMI，跨平台的Web Service等性能都要好。在追求性能的同时支持客户端的HA (High Availability)，容错机制为服务提供了良好的可靠性保证。  跨平台：由于特定系统的需要，会使用特定的语言或平台来开发，这么一来会存在多个平台的情况，比如有 java、.net、c++等，想要实现这些系统的简单高效的跨平台调用是一个非常麻烦的事情，dubbo 通过为不同平台提供不同的客户端来实现跨平台，目前已有 java、.net、c++等平台的客户端。服务开发人员在开发服务时无需关心这些细节，所开发出来的服务就能很好的支持跨平台调用。  便捷开发：这是dubbo设计的主要目标之一，开发人员不需要了解网络通讯，不需要关心如何跨平台，不需要知道如何塔建一个服务集群以及怎么做HA，这些框架都做了封装，需要做的只是定义一个接口，实现接口方法，标注注解。可以便捷的使一般程序员能够开发出优秀程序员一样高效、安全、稳定的服务。对于服务调用者也是非常简单的，所有服务开发人员不需要了解的东西调用者同样也不需要了解，对调用者来说调用远程服务的方法就像是调用本地方法一样。  高扩展性：当服务有压力时可以随时加服务器，增加服务节点来分担压力，客户端的负载均衡模块可以根据服务结点的压力情况来调整各结点的权重并平衡压力。  安全性：dubbo可以对调用者进行权限授权，不同的调用者只能调用对他授权的方法，这对有些对外暴露的服务又想对访问进行授权将会非常有用。   * 系统架构：   支付平台2.0架构图 课题特色少有人研究 目前市场上已存在大量聚合支付平台，但是行业野蛮发展、鱼目混珠，其中不乏一些平台并不能满足监管政策。也即是，很少有针对聚合支付较好的研究论文。 经济效益 “聚合支付平台系统”担负着提升公司效益的作用，以某金融公司的聚合支付平台系统为例，给公司创造着经济效益：  提升业务效率：传统模式下，对于新的业务需要单独开发，不能灵活复用，造成项目上线周期长，错过市场时机；同时，一些小企业不具备开放能力，也因此上线支付能力遇到困难。  系统稳定性：由于多数企业开发能力、技术资源优先，通常系统的底层通道单一，也因此系统的稳定性过度依赖于底层单一通道；  释放人力资源：释放技术资源，避免重复工作、多次开发； 社会效益 “聚合支付平台系统”作为业务支付阶段的重要一环，承担着较大的社会责任。以某金融公司的聚合支付平台系统为例，主要具有以下社会意义：  提高了用户提现、还款时的体验，更加多途径、更加便利的实现提现与还款操作。有效减小了用户的操作成本，对降低用户逾期率有显著效果，让金融回归“普惠”的本质；  给合作商户提供更快速、便捷的入款途径，合理地避免了因账期问题导致的商户资金周转问题，为友商创造了良性的运营环境；  方便金融监管，通过支付平台对金融各业务线支付订单账务统一管理，统一对小贷各业务的交易数据管理，方便统一上报，有效降低了监管成本； 研究基础与工作条件软件工程知识 通过软件工程学科的学习，已掌握了软件工程的UML设计基本理论知识，并且已独立完成了多项应用的设计工作。目前已具备数据库设计和php、java开发的基本技术开发能力，灵活运用eclipse开发工具。熟悉如何从管理和技术两方面研究如何更好地开发和维护计算机软件。熟悉在软件生命周期全过程中应完成的任务的性质，在概念上可以把软件生命周期划分成问题定义、可行性研究、需求分析、概要设计、详细设计、编码和单元测试、综合测试以及维护等多个阶段工作。 工作基础 已先后在两家互联网公司实习获得了相关的行业知识。参与过某知名互联网公司钱包产品的设计，及某知名互联网公司金融业务的支付平台的设计工作。对于聚合支付平台系统的大致流程有了初步的尝试。通过在企业实际实习实践过程中，发现的企业内存在的一些问题和熟悉监管政策，建立了适用于当下最高效且符合监管的要求的设计方案，并积极的在现有行业现有关联产品的基础上加以创新。 开发环境 已自行租用windows2012云服务器，方便自己学习使用。 研究思路 通过查阅相关的文献材料，了解到国内外的支付行业存在的问题、行业现状和发展趋势。学习到了支付系统的相关理论和方法，制定了本课题的技术路线和研究方法。后续将进一步调研大量行业经验和阅读论文材料，进一步增加研究与设计的能力。 论文计划  1. 第一阶段：文献阅读，需求整理   2018年11月 - 2019年2月熟悉项目需求，了解相关技术，整理需求。   1. 第二阶段：系统设计   2019年2 月- 2019年3月 完成系统的概要设计和详细设计。  2019年4月 - 2019年5月 完成前端系统及后台管理平台的原型。实现基本支付功能，做为进一步研究的依据。   1. 第二阶段：系统的实现   2019年6月 - 2019年9月 实现聚合支付平台中的主要功能模块。  2019年10月 - 2019年10月 系统主要模块单元测试，以及模块功能的完善和代码优化。   1. 第二阶段：论文撰写   2019年10月 - 2019年11月 完成论文撰写，提交学术论文。   1. 第三阶段：文档总结   2019年11月 - 2020年3月 总结并完成毕业论文。 |

|  |
| --- |
| **四、本研究课题可能的创新之处（不少于500字）**  4.1.少有人研究  目前市场上已存在大量聚合支付平台，但是行业野蛮发展、鱼目混珠，其中不乏一些平台并不能满足监管政策。也即是，很少有针对聚合支付较好的研究论文。  4.2.经济效益  “聚合支付平台系统”担负着提升公司效益的作用，以某金融公司的聚合支付平台系统为例，给公司创造着经济效益：  提升业务效率：传统模式下，对于新的业务需要单独开发，不能灵活复用，造成项目上线周期长，错过市场时机；同时，一些小企业不具备开放能力，也因此上线支付能力遇到困难。  系统稳定性：由于多数企业开发能力、技术资源优先，通常系统的底层通道单一，也因此系统的稳定性过度依赖于底层单一通道；  释放人力资源：释放技术资源，避免重复工作、多次开发；  4.3.社会效益  “聚合支付平台系统”作为业务支付阶段的重要一环，承担着较大的社会责任。以某金融公司的聚合支付平台系统为例，主要具有以下社会意义：  提高了用户提现、还款时的体验，更加多途径、更加便利的实现提现与还款操作。有效减小了用户的操作成本，对降低用户逾期率有显著效果，让金融回归“普惠”的本质；  给合作商户提供更快速、便捷的入款途径，合理地避免了因账期问题导致的商户资金周转问题，为友商创造了良性的运营环境；  方便金融监管，通过支付平台对金融各业务线支付订单账务统一管理，统一对小贷各业务的交易数据管理，方便统一上报，有效降低了监管成本； |
| **五、研究基础与工作条件（1.与本项目相关的研究工作积累基础 2.包括已具备的实验条件，尚缺少的实验条件和拟解决途径）（不少于500字）**  5.1.软件工程知识  通过软件工程学科的学习，已掌握了软件工程的UML设计基本理论知识，并且已独立完成了多项应用的设计工作。目前已具备数据库设计和php、java开发的基本技术开发能力，灵活运用eclipse开发工具。熟悉如何从管理和技术两方面研究如何更好地开发和维护计算机软件。熟悉在软件生命周期全过程中应完成的任务的性质，在概念上可以把软件生命周期划分成问题定义、可行性研究、需求分析、概要设计、详细设计、编码和单元测试、综合测试以及维护等多个阶段工作。  5.2.工作基础  已先后在两家互联网公司实习获得了相关的行业知识。参与过某知名互联网公司钱包产品的设计，及某知名互联网公司金融业务的支付平台的设计工作。对于聚合支付平台系统的大致流程有了初步的尝试。通过在企业实际实习实践过程中，发现的企业内存在的一些问题和熟悉监管政策，建立了适用于当下最高效且符合监管的要求的设计方案，并积极的在现有行业现有关联产品的基础上加以创新。  5.3.开发环境  已自行租用windows2012云服务器，方便自己学习使用。  5.4.研究思路  通过查阅相关的文献材料，了解到国内外的支付行业存在的问题、行业现状和发展趋势。学习到了支付系统的相关理论和方法，制定了本课题的技术路线和研究方法。后续将进一步调研大量行业经验和阅读论文材料，进一步增加研究与设计的能力。 |

**学位论文工作计划**

|  |  |  |
| --- | --- | --- |
| 时间 | 研究内容 | 预期效果 |
| 2018年11月 - 2019年2月 | 熟悉项目需求，了解相关技术，整理需求。 | 完整需求的整理工作。 |
| 2019年2 月- 2019年3月 | 系统的概要设计和详细设计。 | 完成系统的概要设计和详细设计。 |
| 2019年4月 - 2019年5月 | 前端系统及后台管理平台的原型设计，实现基本测量功能，做为进一步研究的依据。 | 完成前端系统及后台管理平台的原型。实现基本支付功能，做为进一步研究的依据。 |
| 2019年6月 - 2019年9月 | 聚合支付平台中的主要功能模块的实现工作。 | 实现聚合支付平台中的主要功能模块。 |
| 2019年10月 - 2019年10月 | 系统主要模块单元测试，以及模块功能的完善和代码优化。 | 完成系统主要模块单元测试，以及模块功能的完善和代码优化。 |
| 2019年10月 - 2019年11月 | 论文撰写。 | 完成论文撰写，提交学术论文。 |
| 2019年11月 - 2020年3月 | 总结并完成毕业论文修改。 | 总结并完成毕业论文的终稿。 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 评  定  小  组  成  员 | 姓 名 | 职 称 | 单位名称 | 职务 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| 导师意见： | | | | |
| 选题来源于科研项目，研究内容合理，技术方案可行。  同意开题。 | | | | |
| 导师（签名）：  日期： 年 月 日 | | | | |
| 开题报告小组意见： | | | | |
| 组长（签名）：  日期： 年 月 日 | | | | |
| 学院意见（签章）： | | | | |
| 负责人：  日期： 年 月 日 | | | | |