# 解决方案

## 方案概述

### 总体目标

直销银行线开户系统（即电子账户系统）建设的总体目标如下：使客户无须到实体网点，通过互联网注册即可成为我行客户，享受我行金融服务。该系统旨在突破我行地域限制，在全国范围开展业务。电子账户系统的建立对于直销银行来说具有不可改变的战略意义。

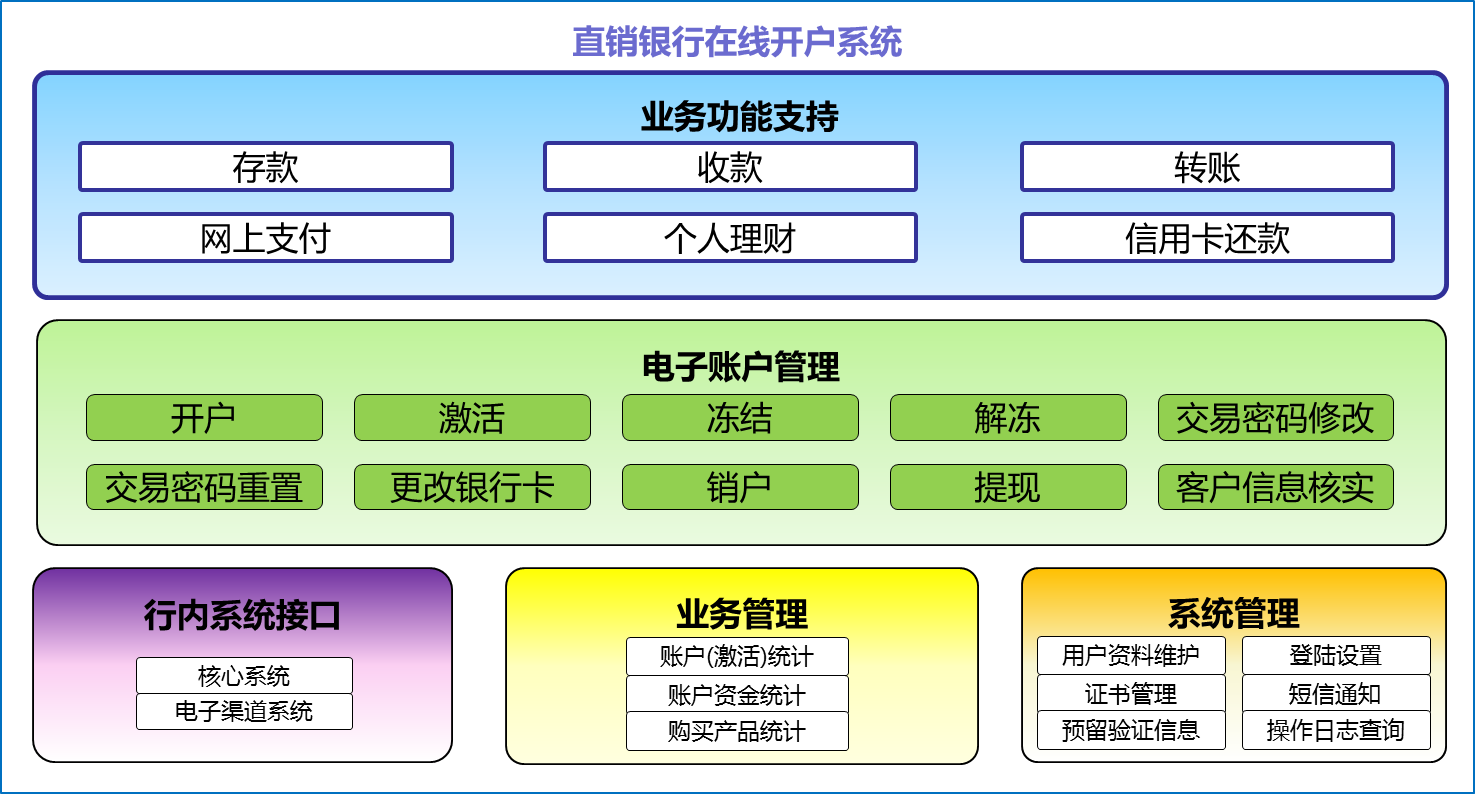
### 总体业务处理流程

直销银行在线开户系统总体业务处理流程如下图所示：



### 总体功能架构

直销银行在线开户系统总体处理流程如下图所示：



直销银行在线开户系统的主要包括业务功能支持、电子账户管理、行内系统接口、业务管理（统计分析与报表）、系统管理等五个功能模块，详细描述如下。

#### 业务功能支持

电子账户等同于借记卡，开立时即分配卡号并生成活期帐户。电子账户性质属于人民币个人结算账户，客户在网上开立，不需要实物卡，开户时不涉及密码信封、重空等实物凭证不需要记账，业务权限主要包括以下内容：

* 存款
  + 活期
  + 各类定期
  + 各类通知存款
* 收款
  + 本行借记卡转入（含本行电子账户转入）
  + 本行贷记卡转入
  + 他行借记卡转入
  + 他行贷记卡转入
  + 其他类型账户转入
* 转账
  + 向绑定的本行同名账户转账
  + 向绑定的他行同名账户转账
* 网上支付
  + 网上支付签约（同原有电子渠道）
  + 网上支付（同原有电子渠道）
* 个人理财
  + 各类理财产品购买赎回
  + 各类基金、货币基金产品购买赎回
* 信用卡还款
  + 本行信用卡还款
  + 他行信用卡还款

#### 电子账户管理

* 开户
  + 功能描述

用户注册成功后，进行电子账户的开户。开户成功后电子账户为未激活状态。

* + 业务流程

入口：

1. 通过“电子账户开户”按钮/链接进入；

2. 通过购买流程进入。

开户业务处理流程如下图所示：



* + 业务规则

交易前：新客户。

交易中:

1. 核心系统开户时，电子账户的开户行默认为北京银行总行营业部。

2. 前端先按照公安部发布的规则对身份证进行初步校验。

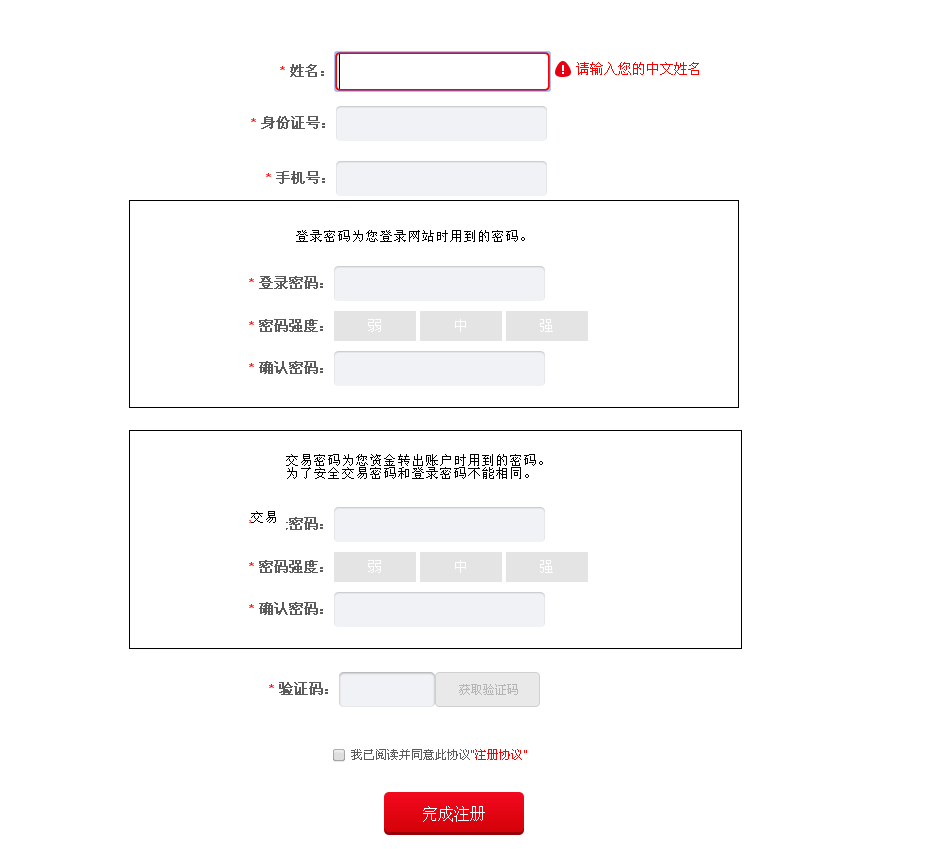
3. 一个身份证只能开立一个电子账户。

4. 开卡没有实体卡、密码信封、重空等实物凭证。

5. 电子账户按照普通活期账户利率进行计息。

* + 页面要素

**基本信息输入页面：**现有页面增加交易密码要素



**开户成功结果页面**：要素与现有页面一致，成功后进入激活页面。

* 激活
  + 功能描述

激活分为上传身份证和绑定银行卡两步，这两步没有先后顺序，因此这两步顺序可能进行调整，正式上线前也可能摘除上传身份证照片的步骤。

绑定银行卡分为绑定本行卡，绑定他行卡走银联渠道，绑定他行卡走转账渠道。客户选择他行卡时需要填写开卡行以及卡号，如果客户选择本行卡只需要填写卡号。

* + 业务流程

激活业务处理流程（前端）如下图所示：



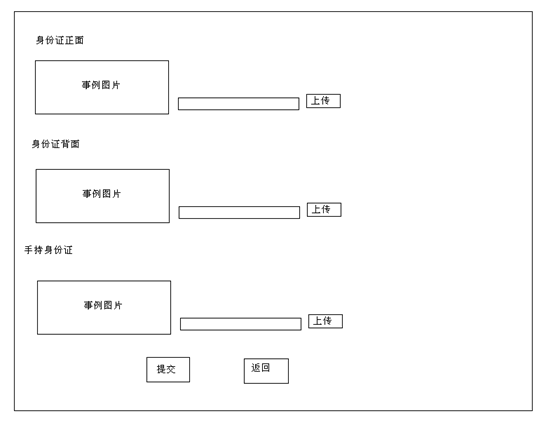
* + 业务规则

交易前：已成功注册且为登录状态。

交易中：身份证图片的大小不能超过2MB。

交易后：无论成功失败都记录日志。

* + 页面要素



* 冻结
  + 功能描述

用户可以对电子账户进行冻结。冻结交易的操作需要经过两重验证：

1.账户交易密码验证；2.手机短信验证码。

通过以上两步后，即可将账户冻结。冻结成功后，账户将无法进行产品的买卖和资金的转出，只能转入。

* + 业务流程

1.点击“冻结”按钮，进入电子账户冻结页面。

2.在电子账户冻结页面中，显示电子账户信息，用户输入账户交易密码和手机验证码，点击“冻结”按钮后，平台提示信息“冻结后，账户将无法进行产品的买卖和资金的转出，您还要冻结电子账户吗？”用户点击“确认”按钮后，系统将冻结指令发送至核心系统。

3.显示电子账户冻结结果。

* + 业务规则

无论成功失败都记录日志。

* + 页面要素

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
|  | 姓名 | 文本 |  |  |  |  |
|  | 电子账户 | 文本 |  |  |  |  |
|  | 开户行 | 文本 |  |  |  |  |
|  | 证件号码 | 文本 |  |  |  |  |
|  | 手机号 | 文本 |  |  |  |  |
|  | 手机验证码 | 输入框 |  | N/6 | 是 |  |
|  | 发送手机验证码 | 链接 |  |  |  | 显示在“手机验证码”输入框右侧；  点击后向用户手机号发送验证码，发送成功后隐藏此链接，显示“发送手机验证码已发送，您可以在\*\*秒后重新发送”文字信息。 |
|  | 发送手机验证码已发送，您可以在\*\*秒后重新发送 | 文本 |  |  |  | 显示在“手机验证码”输入框右侧；  发送验证码成功后显示，秒数从60（验证码的有效时间）到0倒数；倒数至0后，隐藏此文字信息，显示“发送手机验证码”链接。 |
|  | 交易密码 | 输入框 |  | N/6 | 是 |  |
|  | 冻结 | 按钮 |  |  |  |  |

* 解冻
  + 功能描述

用户先通过电话银行申请账户解冻，客服人员进行身份核实后，用户需要在平台再次认证才可解冻。

* + 业务流程

1.用户已通过后台的身份核实。

2.点击“解冻”按钮，进入电子账户解冻页面。

3.在电子账户解冻页面中，显示电子账户信息，用户输入账户交易密码和手机验证码，点击“解冻”按钮后，平台提示信息“解冻后，账户将恢复产品买卖和资金转出功能，您确定是否解冻电子账户？”用户点击“确认”按钮后，系统将解冻指令发送至核心系统。

4.显示电子账户解冻结果。

* + 业务规则

交易前:用户已通过电话银行的身份核实。

交易后:无论成功失败都记录日志。

* + 页面要素

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
|  | 姓名 | 文本 |  |  |  |  |
|  | 电子账户 | 文本 |  |  |  |  |
|  | 开户行 | 文本 |  |  |  |  |
|  | 证件号码 | 文本 |  |  |  |  |
|  | 手机号 | 文本 |  |  |  |  |
|  | 手机验证码 | 输入框 |  | N/6 | 是 |  |
|  | 发送手机验证码 | 链接 |  |  |  | 显示在“手机验证码”输入框右侧；  点击后向用户手机号发送验证码，发送成功后隐藏此链接，显示“发送手机验证码已发送，您可以在\*\*秒后重新发送”文字信息。 |
|  | 发送手机验证码已发送，您可以在\*\*秒后重新发送 | 文本 |  |  |  | 显示在“手机验证码”输入框右侧；  发送验证码成功后显示，秒数从60（验证码的有效时间）到0倒数；倒数至0后，隐藏此文字信息，显示“发送手机验证码”链接。 |
|  | 交易密码 | 输入框 |  | N/6 | 是 |  |
|  | 解冻 | 按钮 |  |  |  |  |

* 交易密码修改
  + 功能描述

用户通过本功能修改电子账户的交易密码。

* + 业务流程

1．用户点击“交易密码修改”链接，进入交易密码修改页面。

2．用户输入原密码及新密码，点击“确认”。

3．发送短信验证码，客户输入短信验证码；

4．校验短信验证码是否正确，两次输入的新密码是否一致，并向后台系统校验原密码是否正确，如果修改失败，则显示修改失败页面并提示用户错误信息；如果修改成功，则显示修改成功页面，后台系统更新密码信息。

* + 业务规则

交易中

1．向后台系统校验原密码是否正确。

2．校验两次输入的新密码是否一致。

3.校验验证码是否正确。

交易后：无论成功失败都记录日志。

* + 页面要素

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
| 1 | 原交易密码 | 输入框 |  | N/6 | 是 |  |
| 2 | 新交易密码 | 输入框 |  | N/6 | 是 |  |
| 3 | 再次确认新交易密码 | 输入框 |  | N/6 | 是 |  |
| 4 | 手机号 | 文本 |  |  |  |  |
| 5 | 手机验证码 | 输入框 |  | N/6 | 是 |  |
| 6 | 发送手机验证码 | 链接 |  |  |  | 显示在“手机验证码”输入框右侧；  点击后向用户手机号发送验证码，发送成功后隐藏此链接，显示“发送手机验证码已发送，您可以在\*\*秒后重新发送”文字信息。 |
| 7 | 发送手机验证码已发送，您可以在\*\*秒后重新发送 | 文本 |  |  |  | 显示在“手机验证码”输入框右侧；  发送验证码成功后显示，秒数从60（验证码的有效时间）到0倒数；倒数至0后，隐藏此文字信息，显示“发送手机验证码”链接。 |
| 8 | 确认 | 按钮 |  |  |  |  |

* 交易密码重置
  + 功能描述

用户通过填写身份证号，上传身份证图片，手机验证码申请密码重置，客服人员通过上传信息以及手机号近期修改时间进行身份核实后，给客户发送重置交易密码短信。

* + 业务流程

1.上传身份证照片

2.后端审核，手机号近期无变更

3.短信提示可以重置了

4.点击重置按钮

5.发送短信验证码

6.用户两次输入新密码，并输入手机验证码，点击“确认”。

7.平台校验两次输入的新密码是否一致，校验手机验证码，如果校验成功，则向后台系统发起更新交易密码请求，更新成功后显示重置成功页面。

* + 业务规则

交易前：用户已通过电话银行的身份核实。

交易中：校验验证码是否正确。

交易后：无论成功失败都记录日志。

* + 页面要素

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
| 1 | 新交易密码 | 输入框 |  | N/6 | 是 |  |
| 2 | 再次确认新交易密码 | 输入框 |  | N/6 | 是 |  |
| 3 | 手机号 | 文本 |  |  |  |  |
| 4 | 手机验证码 | 输入框 |  | N/6 | 是 |  |
| 5 | 发送手机验证码 | 链接 |  |  |  | 显示在“手机验证码”输入框右侧；  点击后向用户手机号发送验证码，发送成功后隐藏此链接，显示“发送手机验证码已发送，您可以在\*\*秒后重新发送”文字信息。 |
| 6 | 发送手机验证码已发送，您可以在\*\*秒后重新发送 | 文本 |  |  |  | 显示在“手机验证码”输入框右侧；  发送验证码成功后显示，秒数从60（验证码的有效时间）到0倒数；倒数至0后，隐藏此文字信息，显示“发送手机验证码”链接。 |
| 7 | 确认 | 按钮 |  |  |  |  |

* 更改银行卡
  + 功能描述

用户更改电子账户所绑定的银行卡。首先提交更换银行卡申请；然后从更换后的银行卡向电子账户汇款；最后通过“待验证”链接确认是否更换成功。

* + 业务流程

1.用户点击“更换银行卡”链接，进入更换页面。

2.用户在更换页面输入新的银行卡卡号，本行卡需要输入银行卡密码，点击“更换”按钮。

3.若本行卡，平台校验新卡的真实性及卡状态是否正常，校验成功，显示结果页面。

4.若他行卡，显示结果页面。

5.在结果页面提示用户需要于3日内从更换后的银行卡向电子账户汇款。

6.此后页面提供“待验证”链接，用户点击后，平台向后台系统查询有无更换后的银行卡向电子账户汇款的明细信息，如有，则更换成功。

* + 业务规则

交易中：每个用户只能绑定一张银行卡。

交易后：无论成功失败都记录日志。

* + 页面要素

更改银行卡输入页面要素：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
| 1 | 姓名 | 文本 |  |  |  |  |
| 2 | 身份证号 | 文本 |  |  |  | 第7位至第15位显示为\*  例如：110102\*\*\*\*\*\*\*\*\*316 |
| 3 | 卡类型 | 单选 |  |  |  | 本行借记卡，他行卡 |
| 4 | 卡号 | 输入框 |  | N/20 |  |  |
| 5 | 卡密码 | 输入框 |  | N/6 |  | 本行卡时显示 |
| 6 | 下一步 | 按钮 |  |  |  |  |

更换银行卡结果页面要素：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
| 1 | 您已成功提交申请，请于3日内从更换后的银行卡向电子账户汇款。 | 文本 |  |  |  |  |

* 销户
  + 功能描述

用户进行电子账户的注销。

* + 业务流程

1.点击“注销”按钮，进入电子账户注销页面。

2.在电子账户注销页面中，显示电子账户信息，用户输入交易密码和手机验证码，点击“注销”按钮后，平台判断电子账户中是否还有生效的产品，如果有，则显示提示信息“您还有未到期的产品，不能注销电子账户！”。

3.如果没有，则判断电子账户是否零余额，如果不是零余额，则显示提示信息“您的电子账户中还有余额，请先将余额转出再进行注销电子账户操作!”。

4.如果是零余额账户，则提示信息“您确定要注销电子账户吗？”用户点击“确认”按钮后，系统将注销指令发送至核心系统。

5.显示电子账户注销结果。

* + 业务规则

交易中：

1.电子账户为零余额且没有还在生效的产品才可销户。

2.系统将在申请销户的当天自动结算账户内资金的活期利息。待客户确认利息金额后，转出至客户的绑定账户同时完成销户。活期利息计算的截止日期为申请销户的当天。

3.申请销户时，用户需确保绑定账户为有效账户，否则利息将无法正常转出至绑定账户。

4.转出利息当日到账。

交易后：无论成功失败都记录日志。

* + 页面要素

注销账户页面要素：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
| 1 | 姓名 | 文本 |  |  |  |  |
| 2 | 电子账户 | 文本 |  |  |  |  |
| 3 | 开户行 | 文本 |  |  |  |  |
| 4 | 证件号码 | 文本 |  |  |  |  |
| 5 | 手机号 | 文本 |  |  |  |  |
| 6 | 手机验证码 | 输入框 |  | N/6 | 是 |  |
| 7 | 发送手机验证码 | 链接 |  |  |  | 显示在“手机验证码”输入框右侧；  点击后向用户手机号发送验证码，发送成功后隐藏此链接，显示“发送手机验证码已发送，您可以在\*\*秒后重新发送”文字信息。 |
| 8 | 发送手机验证码已发送，您可以在\*\*秒后重新发送 | 文本 |  |  |  | 显示在“手机验证码”输入框右侧；  发送验证码成功后显示，秒数从60（验证码的有效时间）到0倒数；倒数至0后，隐藏此文字信息，显示“发送手机验证码”链接。 |
| 9 | 交易密码 | 输入框 |  | N/6 | 是 |  |
| 10 | 注销 | 按钮 |  |  |  |  |

电子账户利息信息页面要素：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
|  | 应结利息 | 文本 |  |  |  |  |
|  | 转入银行卡号 | 文本 |  |  |  |  |
|  | 开户行 | 文本 |  |  |  |  |
|  | 确认 | 按钮 |  |  |  |  |
| 温馨提示：   1. 系统将在申请销户的当天自动结算账户内资金的活期利息。活期利息计算的截止日期为申请销户的当天。 2. 请确保绑定账户为有效账户，否则利息将无法正常转出至绑定账户。 3. 如果绑定账户为本行借记卡账户，则转出利息当日到账，如果为非本行账户，则到账时间在3-5个工作日左右。 | | | | | | |

* 提现
  + 功能描述

客户将卡内余额转账转账给客户绑定银行卡。

* + 业务流程

即“业务功能支持/转账”功能，业务流程同原有网上银行业务流程。绑定卡为本行卡时，走行内转账；绑定卡为他行卡时，通过超级网银进行转账。

* + 业务规则

交易前：账户有活期余额。

交易中：单笔5万，每日限额50万。

交易后：无论成功失败都记录日志。

* + 页面要素

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **序号** | **名称** | **输入/输出** | **缺省值** | **类型/长度** | **是否必输** | **备注** |
|  | 姓名 | 文本 |  |  |  |  |
|  | 电子账户 | 文本 |  |  |  |  |
|  | 绑定卡卡号 | 文本 |  |  |  |  |
|  | 绑定卡开卡行 | 文本 |  |  |  |  |
|  | 转出金额 | 输入框 |  |  |  |  |
|  | 手机验证码 | 输入框 |  |  |  |  |
|  | 发送手机验证码 | 链接 |  |  |  | 显示在“手机验证码”输入框右侧；  点击后向用户手机号发送验证码，发送成功后隐藏此链接，显示“发送手机验证码已发送，您可以在\*\*秒后重新发送”文字信息。 |
|  | 发送手机验证码已发送，您可以在\*\*秒后重新发送 | 文本 |  |  |  | 显示在“手机验证码”输入框右侧；  发送验证码成功后显示，秒数从60（验证码的有效时间）到0倒数；倒数至0后，隐藏此文字信息，显示“发送手机验证码”链接。 |
|  | 交易密码 | 输入框 |  | N/6 | 是 |  |
|  | 提交 | 按钮 |  |  |  |  |

* 客户信息核实

本功能属于“内部管理”功能。

* + 功能描述

后台需要将客户注册的个人信息与上传的身份证照片，来账信息进行核对。要求个人信息与上传的身份证信息必须匹配、来账信息必须与个人信息和上传的身份证信息匹配。

1.个人信息：姓名、身份证.

2.上传的身份证照片：正面照片，反面照片，身份证正面在脸旁边的照片.

3.来账信息：客户他行卡的姓名、身份证

4. 使用人员：

客户信息审核功能使用人员分为操作员和管理员。操作员进行具体审核操作。管理员可以查看操作员完成情况（包括每日完成数量，审批通过笔数，审批不通过笔数等。以及可以查看具体每一部审核的情况。

5.使用说明：

系统将按时间顺序分配给操作员第一个未被处理的审核信息，该信息必须是客户完成绑定银行卡以及上传身份证照片后的信息。该信息分配给操作员后状态自动置为受理中，确保不会有2个操作员同时查出同一条记录的情况。信息包括客户上传三证信息、照片，联网核查返回的三证信息、照片、以及联网核查状态，如果是联网核查返回信息校验失败，则需要双人复核，或者管理员授权。

* + 业务流程



#### 行内系统接口

在线开户系统完成其业务逻辑需要与行内系统进行交互，分为“与核心系统接口”和“与电子渠道系统接口”两类。其中前者是必须的，后者是根据需要选择，具体如下：

* 与核心系统接口

电子账户等同于借记卡，开立时即分配卡号并生成活期帐户，电子账户性质属于人民币个人结算账户。因此与账户及资金往来相关的所有业务功能均需要在线开户系统与核心系统进行对接，使用原有或者新增的核心交易来实现：

* + 业务功能支持所包含的全部交易接口
  + 电子账户管理所包含的部分交易接口
* 与电子渠道系统接口

在线开户业务受理渠道包括微信客户端、手机客户端和PC客户端。本方案包含PC客户端的开发建设，提供基于PC客户端使用的业务功能界面和各类管理界面。为了提高实施效率，合理利用已有IT资源，在线开户系统将开放标准服务调用接口（即后台交易接口），以便与行内电子渠道已有系统进行平滑对接，包括以下系统：

* + 原有网上银行
  + 原有手机银行
  + 原有微信银行

#### 业务管理

业务管理功能位于在线开户系统“内部管理功能域”，属于向业务管理人员提供的内部管理功能，主要包括相关的统计分析及报表功能，具体如下：

* 账户统计
  + 已开立账户统计表
  + 已激活账户统计表
* 资金统计
  + 电子账户日均来账和资金来源统计表

1.报表信息说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **填写项** | **填写内容** | **解释说明** |
| 1 | 报表名称 | 电子账户日均来账和资金来源统计表 |  |
| 2 | 报表分类 | 个人业务 |  |
| 3 | 报表主管部门 | 直销银行部 |  |
| 4 | 报表需求提出人 |  |  |
| 5 | 功能描述 | 查询客户的电子账户每天来自行外的和行内的资金，以及账户日均余额和电子账户总额 |  |
| 6 | 是否需要导出 | 可导出xls、word、txt、pdf |  |
| 7 | 是否需要加密 |  |  |
| 8 | 是否需要审批 |  |  |
| 9 | 是否具有该表管理的权限 |  |  |
| 10 | 查询条件1 | 开始日期 |  |
| 11 | 查询条件2 | 结束日期 |  |
| 12 | 报表表头1 | 开始日期 |  |
| 13 | 报表表头3 | 行内资金 |  |
| 14 | 报表表头3 | 行外资金 |  |
| 15 | 报表表头4 | 汇总 |  |
| 16 | 查询时间范围 |  |  |
| 17 | 数据保存时间 |  |  |

2.报表表样：

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **行内金额** | **行外金额** | **汇总** |
|  |  |  |  |

3.报表权限：

|  |  |  |  |
| --- | --- | --- | --- |
| **权限** | **总行** | 部室名称 | 直销银行部 |
| **分行** | 部室名称 | 说明 |
| **管辖行** | 全部管辖行或管辖行名称 | 说明 |
| **支行** | 全部支行或支行名称 | 说明 |
| **特殊权限组** | 无或权限组名称 | 说明 |

* 产品统计
  + 电子账户购买产品统计表

1.报表信息说明：

|  |  |  |  |
| --- | --- | --- | --- |
| **编号** | **填写项** | **填写内容** | **解释说明** |
| 1 | 报表名称 | 电子账户购买产品统计表 |  |
| 2 | 报表分类 | 个人业务 |  |
| 3 | 报表主管部门 | 直销银行部 |  |
| 4 | 报表需求提出人 |  |  |
| 5 | 功能描述 | 查询客户通过电子账户购买产品的金额 |  |
| 6 | 是否需要导出 | 可导出xls、word、txt、pdf |  |
| 7 | 是否需要加密 |  |  |
| 8 | 是否需要审批 |  |  |
| 9 | 是否具有该表管理的权限 |  |  |
| 10 | 查询条件1 | 开始日期 |  |
| 11 | 查询条件2 | 结束日期 |  |
| 12 | 报表表头3 | 产品名称 |  |
| 13 | 报表表头1 | 日期 |  |
| 14 | 报表表头2 | 产品名称 |  |
| 15 | 报表表头3 | 购买金额 |  |
| 16 | 数据保存时间 |  |  |
| 17 | 查询时间范围 |  |  |

2.报表表样：

|  |  |  |  |
| --- | --- | --- | --- |
| **开始日期** | **结束日期** | **产品名称** | **购买金额** |
|  |  |  |  |

3.报表权限：

|  |  |  |  |
| --- | --- | --- | --- |
| **权限** | **总行** | 部室名称 | 直销银行部 |
| **分行** | 部室名称 | 说明 |
| **管辖行** | 全部管辖行或管辖行名称 | 说明 |
| **支行** | 全部支行或支行名称 | 说明 |
| **特殊权限组** | 无或权限组名称 | 说明 |

#### 系统管理

系统管理功能属于在线开户系统向用户提供的自有管理功能，具体如下：

* 用户资料维护
* 证书管理
* 预留验证信息
* 登陆个性化设置
* 短信通知设置
* 操作日志查询

## 方案建议

直销银行在线开户系统在方案设计时需要充分考虑以下非功能因素，并在系统建设中付诸实践。

### 开放性

系统架构清晰，基于IT业界标准。对系统中的各种网络协议、硬件接口、数据接口等进行统一规划，满足北京银行现有系统的信息交互要求，为未来的系统扩展奠定基础。

### 可扩展性

根据银行业务发展，能够灵活扩展设备容量和提升设备性能；具备技术升级、设备更新的灵活性。

根据北京银行直销银行的业务发展和现有IT架构，提供整体技术架构方案，包括产品本身的软件架构、硬件架构（例如系统部署方案与灾备方案）、交互涉及的上下游系统的整体IT架构方案。并提供产品的性能报告，包括系统运行负载的压力测试的可靠性报告等。

### 参数配置灵活

提供强大的参数配置功能，包括系统参数配置、业务开发参数配置，支持系统或业务功能的快速调整及扩展，以便有效减少后续实施费用和开发周期。能够提供简单、便捷、灵活的配置功能，实现不同业务不同管理流程，新增业务功能开发可以通过配置复用原有流程，减少开发工作量。

对于业务管理功能（统计报表功能），系统能够提供简单、便捷、灵活的查询功能和二次开发功能，并可以导出到EXCEL中。

### 松散耦合

在系统设计之初，制定松散耦合的设计原则，保证在更新系统或模块时其它与之有交互的系统或模块能正常运行。

软件松耦合一般体现在系统与外围系统或系统内部间的交互关系，在实现过程中一般采用如下措施：

1. 基于消息的通信

系统与外部系统进行集成时，采用标准WEB服务进行系统间的集成，减少业务系统间的耦合度。

1. 面向服务的编程

系统内部子系统及模块间集成时，将接口封装成SCA服务，将服务发布成一个公共的构建包，供其它子系统或模块调用，将系统内部间交互的网状结构变成与公共构建包交互的星型结构，降低了模块间的耦合度。

### 易复用

系统中的一些公共服务有可能会被其它系统也会使用，为避免重复开发，浪费资源，应该尽可能使组件化，达到较强的易复用性。

易复用性主要做到以下几点：

1. 代码高内聚低耦合
2. 不对特定框架有强依赖
3. 代码灵活，可经过配置更改功能实现

系统在设计阶段应充分考虑到易复用性的要求，积累了很多可复用构件，包括金融DataCell、复选树等。例如金额标签，如果使用普通的输入框，一则容易输错，二则不易辨认；通过封装标签的形式，可以方便的实现代码的复用。

### 易维护

系统规模大，功能复杂，代码量很大，业务上存在变化的可能，未来的运维人员有可能不是开发者，所以，系统必须要做到易维护。

系统在设计阶段，每个模块都有详细的设计文档；如果初期业务日志和版本管理做不好，会造成在UAT测试阶段暴露出错误排查困难等问题，同时需要通过规范化业务日志和版本管理。

易维护性主要做到以下几个方面：

1. 项目各个模块有详细的开发设计文档
2. 统一的编程规范，如文件命名、代码注释规范；
3. 详细的业务日志，备份历史日志；
4. 统一的数据模型规范，如视图、存储过程、业务表、关系表等的规范命名；
5. 高标准的版本管理，包括应用程序、数据、脚本、文档等的版本控制。

### 安全性

提供完整的信息管理机制和控制手段；提供系统备份、数据恢复、事故监控和网络安全保密等技术措施。满足业务对系统服务时间的要求，需要满足7\*24小时的服务。

系统数据安全（如：数据的完整性、一致性和有效性，数据传输过程的安全、数据加密等）；

用户安全管理（如：用户授权管理、权限管理、认证管理等）。

系统在采用职能、角色、功能的权限管理模型，使用EOS DAS组件做数据访问，禁止拼接SQL防止SQL注入，LDAP认证，对密码信息、授权信息加密处理，网络隔离，数据定时备份等措施来做到系统安全。

常见的信息系统安全性措施如下：

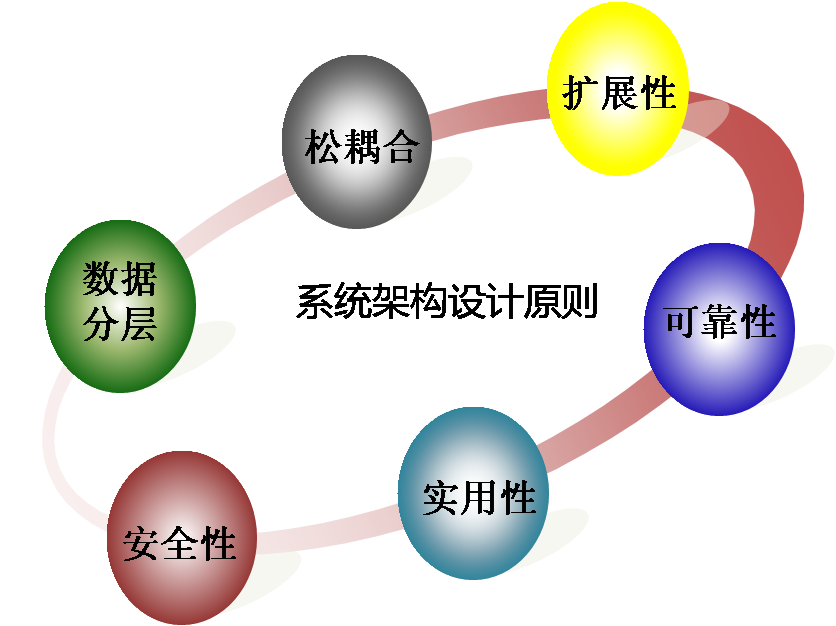
1. 严格的权限管理模型；
2. 防止恶意代码攻击；
3. 敏感信息加密处理；
4. 生产、准生产、开发测试环境网络隔离；
5. HTTPS协议访问代替HTTP协议；
6. 灾备方案。

## 技术设计

### 架构设计原则

在普元开发平台的架构层次规范、模块规范、异常与日志处理规范等固化的开发设计规范的基础上，根据直销银行在线开户系统业务的需要，实现统一架构技术体系、统一业务控制处理体系、统一系统维护、备份和开发管理体系。

直销银行在线开户系统设计遵循以下原则：



* 数据分层设计原则：系统分为源数据层与应用数据层两个数据层，用于分类存放、备份数据。
* 松藕合分层设计原则：系统基于SSH分层架构设计，在适应系统需求的原则下，系统结构上采用组件模块化设计，达到低耦合的分层结构的目的，利于团队成员的分工协作，提高开发效率，降低项目风险，独立实现各个业务模块的功能设计，以完成整个系统的开发。
* 扩展性原则：业务组件模块化设计原则，大量采用工厂设计模式、策略设计模式、适配器设计模式等进行接口、类设计，组件之间的重新装配以及添加新的业务处理组件可满足不断扩展的业务需求。
* 实用性原则：满足用户的操作习惯，满足用户实用性、易用性、适应性和易扩展性原则。
* 可靠性原则：系统满足可靠性，出现异常时，系统提示简洁易懂的警告信息，方便用户进行交互处理，系统对重要业务进行容错处理，提高系统的健壮性要求。
* 安全性原则：系统采用五层安全体系，即基础设施安全、网络安全、系统安全、用户安全、数据安全和业务安全。系统必须具备高可靠性，对使用信息进行严格的权限控制，技术上采用严格的安全与保密措施，保证系统的可靠性、保密性和数据一致性等。

### 系统总体架构

本方案中，直销银行在线开户系统将采用普元金融统一平台(EOS4Banking)进行业务功能开发以及的业务流程灵活定制、扩展与流程审批，同时在页面框架去做灵活的展现。本方案使用到了金融统一平台中的统一开发平台EOS和业务流程平台BPS。

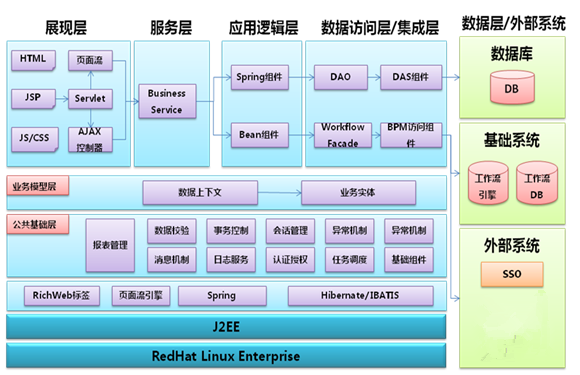
系统的总体架构如下图所示：



### 系统技术架构

直销银行在线开户系统从技术实现逻辑上分为：展现层、服务层、应用逻辑层及数据访问层四层。应用系统基于J2EE平台，集成业界流行的开源技术框架，如：Spring、Hibernate、IBatis等；应用系统基于中信银行统一开发平台提供的基础应用框架，主要包括认证授权、会话管理、事务控制、异常处理、日志服务及数据访问服务等技术组件。应用系统还集成了工作流引擎、SSO单点登录等系统。

系统的技术架构如下图所示：



### 系统数据架构

数据架构设计是一个应用系统设计中非常重要的组成部分，因为数据架构设计的优劣会直接影响应用系统的使用效果，以及未来的可扩展空间。所以，一个成功的数据模型设计的关键是制定一个可行的设计方案，设计方案由所设计的数据库目标和业务需求决定，具备准确表示业务含义、容易使用和维护、对最终用户操作的响应时间合理、便于数据库结构地改进、便于数据检索和修改、尽可能少地存储冗余数据等特点。

#### 数据架构设计原则

一般来说，数据架构设计时应考虑如下几点：

1. 按照业务模块设计数据架构。
2. 准确表示业务含义,主要体现在表名、视图名、字段名、字段类型上符合业务含义
3. 容易扩展，可以通过横表转纵表的方式（列转行）实现。
4. 容易使用和维护，表中字段应该简洁明了，尽量少使用复合主键。
5. 数据库结构便于改进。
6. 尽可能减少冗余数据存储，降低维护难度。
7. 在设计数据模型时，必须考虑公共数据在数据类型定义方面的一致性。例如，在系统中会记录与周边系统交互所需要的关键数据（包括CRM系统中的CRM客户编号，核心系统中的核心客户编号、帐号等），所以在进行数据模型设计时应考虑数据类型及数据长度是否一致的问题。
8. 为了解决系统间的数据交互提供了数据标准。在设计数据模型时，在满足信贷流程处理需求的前提下，尽量遵循数据标准。如果发现存在标准与需求矛盾的情况，或数据标准存在缺陷的情况下，会提出调整标准的建议。

在设计数据模型时，应充分考虑在实体中包括关键关联属性，减少实体间的依赖关系，从而保证了数据基础基本完整，达到在数据层面保证对这种情况的适应能力的目标。

#### 数据架构设计实现

进行数据模型设计时，从应用设计的角度考虑，主要具有下列几方面的特点：

* 业务涉及多种、大量、繁多的信息，因此需要划分出关键实体和辅助实体，并准确定义实体之间的关联关系；
* 业务处理涉及多个环节、多个用户、处理周期长，因此需要将业务自身数据、对业务处理的数据及状态需要结合起来看；
* 部分业务环节处理逻辑相对复杂，或需要多个工作日完成，因此需要满足数据暂存的需求；
* 作为业务处理依据的客户信息随时间会发生变化，流程结束后的数据归档，因此需要满足保存历史数据的需求；
* 流程中路径判定、资源选择是需要应用与工作流平台互动；
* 对特定流程任务，重新上报时，会沿用前人员，因此需要保存流程中各个任务日志；
* 流程中的任务步骤会有增减变化，因此需要保证配置的灵活性。
* 为了满足上述的业务处理特点，在进行数据模型设计是，根据业务特征的不同，把涉及到的实体分成两大类：
* 关键实体：即业务流程中主要被处理的实体。

辅助实体：即描述辅助信息的实体，记录了对关键实体进行处理的辅助信息。

### 系统模块设计

一般来说，模块设计应满足以下原则：

1. 模块独立性

模块独立性主要表现在应用逻辑的独立，具备完整独立的应用功能，并且不同模块间的交互做到尽可能少并且保证交互方式统一，同时减少公共的变量和数据结构。

模块独立性限制了出现的错误范围，提高系统的易排错性，保证软件开发的效率和质量。

1. 耦合度

耦合度主要是模块外部考察模块独立性程度,用来衡量多个模块间的相互联系；耦合类型有数据耦合、控制耦合、公共耦合、内容耦合四种方式。

1. 数据耦合：指两个模块彼此交换数据。如一个模块的输出数据是另一个模块的输入数据，或一个模块带参数调用另一个模块，下层模块又返回参数。
2. 控制耦合：若在调用过程中，两个模块间传递的不是数据参数而是控制参数，则模块间的关系即为控制耦合。
3. 公共耦合：若多个模块对同一个数据区进行存取操作，它们之间的关系称为公共耦合。公共数据区可以是全程变量、共享的数据区、内存的公共复盖区、外存上的文件、物理设备等。当两个模块共享的数据很多，通过参数传递不方便时，可以使用公共耦合。
4. 内容耦合：内容耦合是耦合程序最高的一种形式，内容耦合往往表现为以下几种形式：

* 一个模块访问另一模块的内部代码或数据
* 两个模块有一部分代码重迭
* 一个模块有多种功能

在模块划分时，应当尽量使用数据耦合，少用控制耦合（尽量转成数据耦合），限制公共耦合的范围，完全不用内容耦合。

### 系统接口设计

#### 接口设计原则

一般来说，大部分系统都需要做接口设计，除非功能极其简单，又无系统间交互的微型系统可以不做接口设计。

为保证应用系统的健壮性和完整性，应用接口设计应满足以下原则：

1、高性能：在系统的设计容量和并发范围内，对外部系统的访问请求提供快速的响应；

2、可伸缩性：在系统资源被充分利用的情况下，可以实现系统的平滑移植和动态扩展；

3、高可用性：实现7×24小时的高可用性，支持容错和自动恢复的功能，支持多机集群的部署方式，并能实现透明地故障迁移和数据恢复；

4、可靠性：防止非系统级的数据丢失或出错；对系统级的数据丢失和出错提供容错和恢复的机制；

5、可扩展性：支持系统新业务的扩展；

6、健壮性：系统应提供可靠的运行环境，保证系统在不可预知的出错情况下，能够提供正确的处理机制；

7、安全性：提供完善的信息安全机制，以实现对信息的全面保护。

#### 外部接口设计

外部接口指得是系统与外围系统间的接口，外部接口采用Web服务的方式主要是考虑到它的时效性强，可移植性好，系统耦合度低等好处；采用消息中间件式，一方面是因为消息方式可靠，消息数据不容易丢失；另一方面其伸缩性和扩展性强，不依赖特定平台和操作系统，可移植性和安全性好。除此之外，也有部分历史遗留原因，外围系统提供的接口就是Web服务或者消息接口。

外部接口的实现方式从时效性来说一般分为同步接口和异步接口。同步接口一般采用Web服务的方式调用；异步接口一般采用消息中间件通信，使用JMS的方式实现。下面详细比较一下同步接口和异步接口的特点：

1. 时效性

顾名思义，同步接口同步调用，调用端（也称客户端）发起Web服务调用，服务端响应Web服务请求并返回结果，服务端返回结果之前，客户端阻塞。异步接口则不同，客户端向目标队列或者主题上发送请求消息后可以继续执行下面的逻辑，不需要阻塞等待结果，结果可能很快得到反馈，也可能较长时间没有响应。

1. 稳定性

同步接口是由Web服务使用HTTP协议，无法保证每次调用都能成功，有可能失败，失败后本次调用的信息不会保存。异步接口通过JMS方式实现，相对Web服务表现得更为稳定，只要保证消息发送到目标队列，消息将不会丢失，不需要关注接收端是否在线，是否接受，由接收端自己控制是否接收到消息。

总之，对于需要得到返回结果，同时其稳定性相对不那么高的服务，如流程接口、规则引擎接口和核心查询类接口，建议使用Web服务；对于时效性要求不高，稳定性要求高的接口，如交易类接口、信息同步类接口，则适合使用异步JMS接口。

#### 内部接口设计

内部接口分主要指的是模块间接口。

模块间耦合方式绝大多数是数据耦合，针对此种耦合方式，接口实现中注意以下几点：

1、数据封装性

封装的好处是可以在不改变接口的情况下改变参数对象内部的数据结构，在不影响其他代码的同时扩展功能。封装可以是非持久化实体、持久化数据实体、查询实体、Pojo对象、Map等。

数据封装性上，在Ajax调用逻辑流、Java调用SCA服务时的数据封装性做的比较好，但是页面流跳转时参数传递较多且分散。

2、星型结构的接口设计

为实现模块间的松散耦合，避免模块循环依赖，模块间接口调用不采用直接调用的方式，而通过一个中间构件包间接调用的方式实现。

在接口设计上，贯彻星型结构设计原则，每个构建包除了依赖几个公共构建包，如common、model和integration外，没有直接相互依赖。

### 基础技术设计

#### 事务

在逻辑流中，可以使用三个图元来控制事务的开始、提交和回滚（在逻辑流编辑器右边的选用板可以见到这三个图元）。

* 事务开始图元相当于开启一个新的事务，事务开始图元后面必须有一个事物提交或者事务回滚图元。
* 事务提交图元相当于将当前的事务提交，事务提交图元前面必须有一个匹配的事务开始图元。
* 事务回滚图元相当于将当前的事务回滚，事务回滚图元前面必须有一个匹配的事务开始图元。

一个事务从“事务开始”图元开始，遇到“事务提交”图元时，该事务向数据库提交；遇到“事务回滚”图元，则将该事务回滚。逻辑流中的事务图元不能嵌套，即在“事务开始”图元之后和“事务提交”或者“事务回滚”之前，不能再有其他的“事务开始”图元。

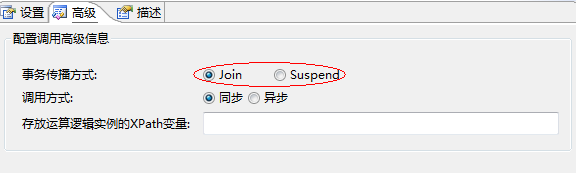
当一个图元在执行的过程中发生异常的时候，可以用异常线将流程转到指定的图元上。异常线会根据在异常线上面定义的具体异常类型，进行匹配，只有异常类型匹配才会按照异常线进行流转。

对于循环当中的异常处理，系统会根据不同的连出点捕获异常情况：

* 当异常连线从循环内的某一个图元连出时候，会捕获当前图元发生异常的情况；
* 当异常连线从循环的结束图元连出时候，会捕获当前循环内发生的任何异常。

##### 描述

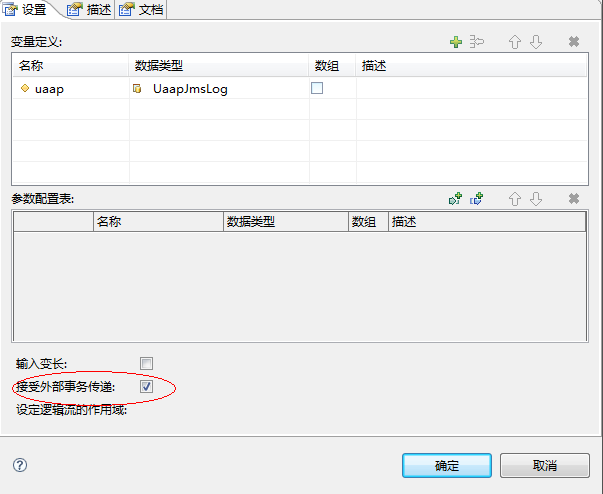
验证EOS平台提供的事务控制图元，以及对于事务传播的控制。在逻辑流中有三种调用图元，分别是"调用运算逻辑"、"调用服务"和"调用逻辑流"，在执行这些图元的时候，可以设置事务传播方式为，Join和Suspend两种：



* Join是指被调用的运算逻辑（服务或逻辑流）参加本逻辑流中的事务，而不开始新的事务（即使显式的使用了"事务开始"图元或者事务管理器接口中的事务开始API）。
* Suspend是在执行被调用的运算逻辑（服务或逻辑流）之前，将本逻辑流中的事务挂起。

在某些情况下，一个逻辑流必须在一个单独的事务内完成，不能join到调用它的逻辑流的事务中去，这个时候，可按下面的步骤操作：

* 打开该逻辑流编辑器；在空白处双击鼠标，弹出对话框。
* 在该对话框中的"设置"标签页中，取消"接受外部事务传递"的选项。
* 此时，在其他的逻辑流中调用这个逻辑流的时候，可以看到"事务传播方式"只能设置为suspend。





创建一个场景：逻辑流A调用逻辑流B，逻辑流A执行数据库写入操作，逻辑流B也执行数据库写入操作。

在此场景下验证一些事务情况：

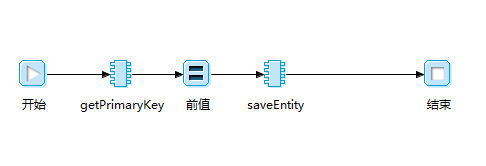
1、异常抛出情况时事务是否会回滚。

2、嵌套事务时的事务隔离级别验证（REQUIRED 还是 REQUIRES\_NEW）。

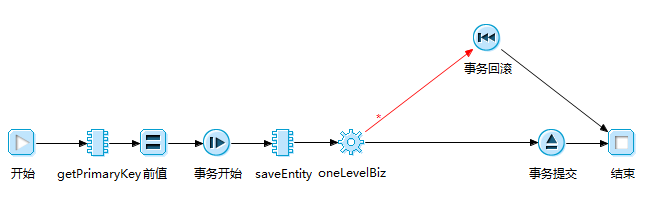
##### 实践细节

创建两个逻辑流oneLevel和twoLevel,并相应增加数据库写入逻辑，oneLevel的“接受外部事物传递”设置为True，towLevel中调用的oneLevel图元的事务传播方式为Join。

oneLevel：定义一个数据库表实体，并通过getPrimaryKey构件生成此对象的主键，使用“赋值”图元对这个对象其他属性设置值，调用saveEntity构件将此对象写入数据库。

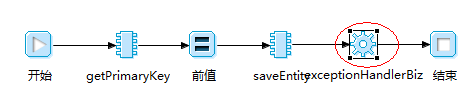


twoLevel：定义一个数据库表实体，并通过getPrimaryKey构件生成此对象的主键，使用“赋值”图元对这个对象其他属性设置值，设置“事务开始”图元，调用saveEntity构件将此对象写入数据库，然后调用oneLevel逻辑流，设置“事务回滚”图元，并使用“异常线”连接oneLevel，“连接线”至结束，最后，增加“事务提交”图元。



场景一：twoLevel->oneLevel,oneLevel抛异常，oneLevel、twoLevel都回滚

在oneLevel逻辑流中saveEntity构件之后，增加抛出异常逻辑模拟异常抛出。

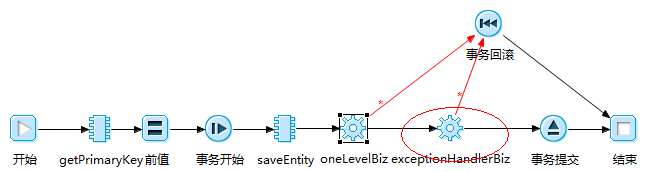


预期：事务在twoLevel启动，twoLevel执行完数据库写入，oneLevel也执行完数据库写入，之后抛出异常，twoLevel捕获到异常并回滚oneLevel和twoLevel的操作，数据库中无新增记录。

实际：twoLevel捕获到了oneLevel的异常，并成功回滚了oneLevel及twoLevel的数据库写入操作，数据库中无新增记录。

场景二：twoLevel->oneLevel,twoLevel完成写入及调用oneLevel，并在twoLevel中抛出异常,oneLevel,twoLevel都回滚。

在twoLevel调用oneLevel逻辑流图元之后增加异常抛出逻辑模拟异常抛出。

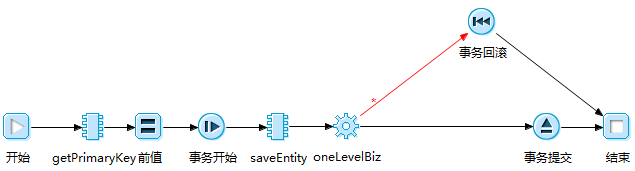


预期：事务在twoLevel启动，twoLevel执行完数据库写入，oneLevel也执行完数据库写入，twoLevel抛出异常，twoLevel捕获到异常并回滚oneLevel和twoLevel的操作，数据库中无新增记录。

实际：twoLevel捕获到了自己的异常，并成功回滚了oneLevel及twoLevel的数据库写入操作，数据库中无新增记录。

场景三：twoLevel->oneLevel,twoLevel发起事务并完成写入，oneLevel完成写入，事务提交。

在twoLevel中调用oneLevel，并完成相应数据库写入操作。

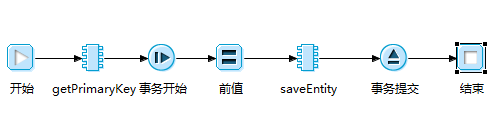


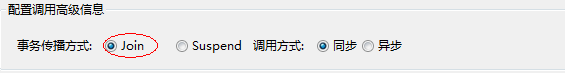
预期：事务在twoLevel启动，twoLevel执行完数据库写入，oneLevel也执行完数据库写入，twoLevel提交事务成功，数据库中新增两条记录。

实际：twoLevel提交事务成功，数据库中并新增两条记录。

场景四：twoLevel->oneLevel,twoLevel发起事务并完成写入，oneLevel发起事务并完成写入，事务提交，twoLevel抛出异常，oneLevel、twoLevel事务都回滚。

在oneLevel中增加事务控制，并在twoLevel调用oneLevel图元中将“事务传播方式”设置为“join”。



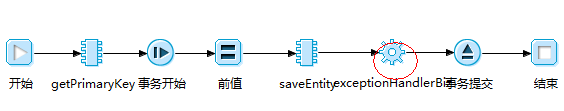


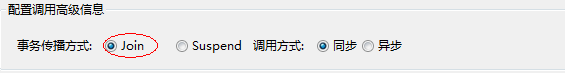
预期：事务在twoLevel中启动，twoLevel执行完数据库写入，并调用oneLevel，oneLevel由于配置了“事务传播方式”为“join”，则oneLevel设置的事务启动融入到了当前事务中，形成一个事务，oneLevel执行完数据库写入，twoLevel抛出异常，oneLevel、twoLevel事务都回滚，数据库中无新增记录。

实际：oneLevel、twoLevel事务全部回滚，数据库中无新增记录。

场景五：twoLevel->oneLevel，oneLevel执行完数据库写入并抛出异常，twoLevel捕获异常并回滚。

在oneLevel中增加事务控制，增加异常抛出逻辑，并在twoLevel调用oneLevel图元中将“事务传播方式”设置为“join”。



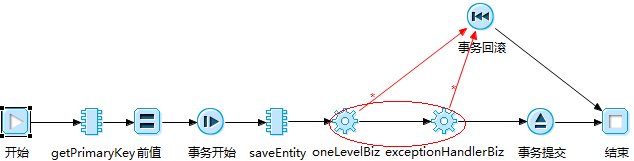
场景流在

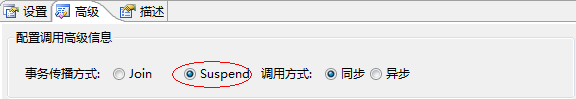
预期：事务在twoLevel中启动，twoLevel执行完数据库写入，并调用oneLevel，oneLevel由于配置了“事务传播方式”为“join”，则oneLevel设置的事务启动融入到了当前事务中，形成一个事务，oneLevel执行完数据库写入，然后抛出异常，oneLevel、twoLevel事务都回滚，数据库中无新增记录。

实际：oneLevel、twoLevel事务全部回滚，数据库中无新增记录。

场景六：twoLevel->oneLevel,调用oneLevel的“事务传播方式”修改为“suspend”，twoLevel抛出异常。

修改twoLevel中调用oneLevel图元的“事务传播方式”值为“suspend”，这样当oneLevel启动事务时判断，如果当前没有事务，则创建新事务，如果有事务，则挂起当前事务后创建新事务。





预期：事务在twoLevel中启动，twoLevel执行完数据库写入，并调用oneLevel，oneLevel由于配置了“事务传播方式”为“suspend”，则oneLevel设置的事务启动并挂起当前事务，形成一个新事务，oneLevel执行完数据库写入，twoLevel然后抛出异常，oneLevel写入成功，但twoLevel事务回滚，数据库中有一条新增记录。

实际：oneLevel数据库写入成功，twoLevel事务回滚，数据库中有一条新增记录。

#### 访问方式

在系统的模块中，为保证系统的松散耦合、易于扩展、方便维护与升级，采用“基于抽象接口层访问方式进行系统的设计与实现”。

1. 逻辑流
   * 1. 提取A模块中能够提供独立的、单一事务性的用例功能，并实现此用例功能。
     2. 在A模块构件包的“构件”中建立此功能服务的逻辑流。
     3. 在Integration构件包中采用EOS提供的Java调用逻辑流的方案形成统一的逻辑流调用工厂。
     4. 在Integration构件包中为A模块的服务建立Java类，并在此类中调用逻辑流工厂，通过传入所需调用的A模块的逻辑构件全名、逻辑流名称、参数实现对于A模块目标逻辑流的调用，此Java类用于封装对于A模块的逻辑访问。（Integration无需依赖A模块或服务提供方）
     5. 在Integration构件包的“服务”中的Compositex，将A模块的Java类的功能定义成为SCA服务（内部服务）。
     6. B模块如需使用A模块的服务，需要与Integration构件包建立依赖互访，并调用Integration下A模块暴露的SCA服务即可实现对于A模块用例功能的访问。
2. 页面流
   * 1. B模块页面流需要使用A模块页面流时，不能通过子页面流的方式来实现（因为需要配置依赖），所以，可以使用一个jsp页面来代替子页面流图元。
     2. 在jsp页面内通过使用<iframe src=”A模块页面流全名”>的方式将A模块页面流加入到B模块页面流中。
     3. B模块页面如需使用或提交A模块页面流时，也可以在B模块具体页面上增加<iframe src=”A模块页面流全名”>标签来实现对于A模块页面流的访问。
3. 优缺点

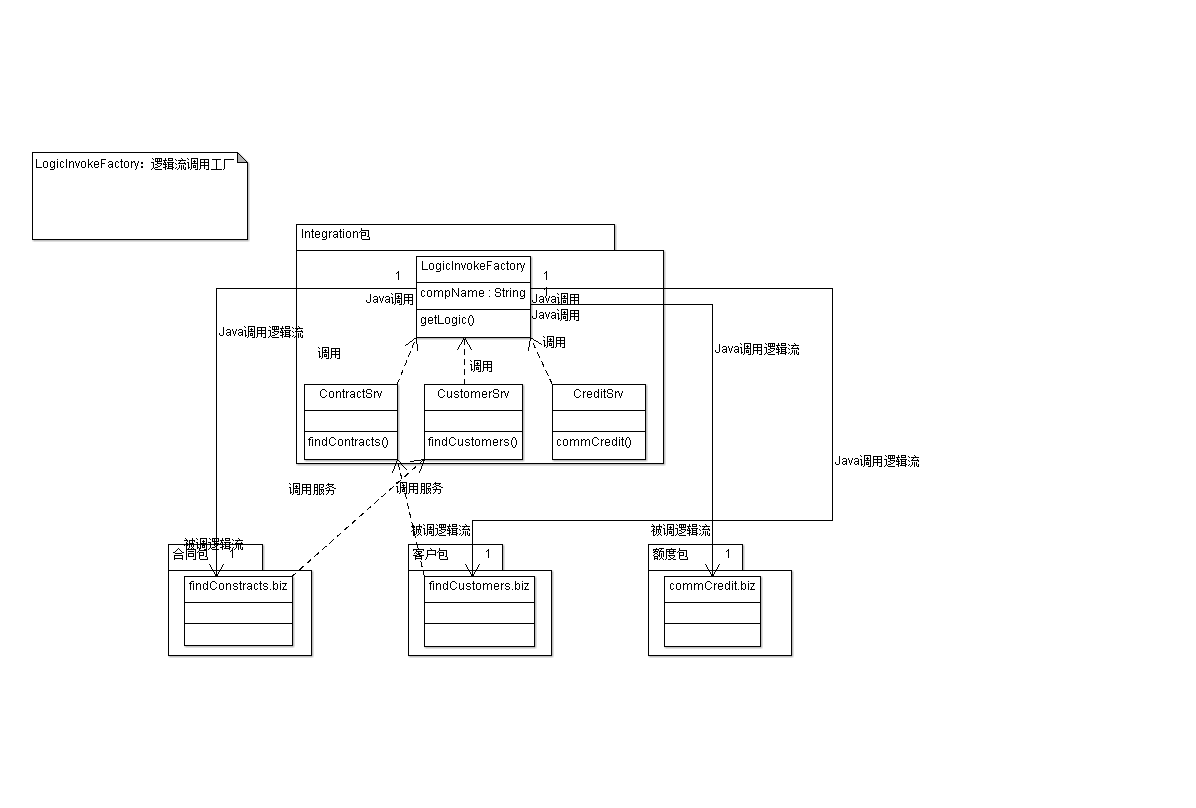
优点：

* Integration构件包无需依赖任何模块的构件包；
* Integration构件包不存放各个模块的功能实现，只是保留引用；
* A模块与B模块间不存在依赖关系；
* A模块与B模块互访只需要依赖Integration构件包；
* 依赖关系为单向。

缺点：

* 需要在Integration构架包中创建Java类实现逻辑流调用工厂；
* 在使用逻辑流调用工厂时需要传递目标逻辑流所在逻辑构件全名及逻辑流名称；
* A模块、B模块需要将对外暴露的逻辑形成逻辑流或者WebService（你能是SCAService或Java构件），才能保证Integration看的到并使用。

1. 示意图



**图3：基于抽象接口层访问实现图**

#### 系统审计

通过EOS平台提供的拦截器功能，可以使用一个实体拦截器拦截当前用户操作的业务实体，并将用户操作并变化的实体属性记录到相应的审计表中。

通过配置相关handler-\*.xml来实现对应handler的配置，拦截器配置文件路径：%EOS\_HOME%\EOSserver\working\%APP\_NAME%\config\handler-\*.xml

实体拦截器有如下的拦截时机：

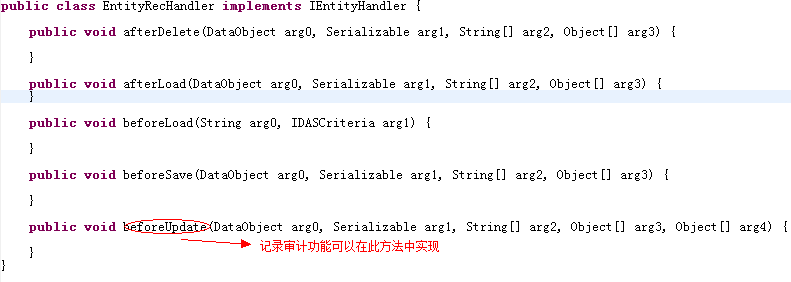
* 在装载数据（Load）之前；
* 在装载数据（Load）之后；
* 在保存数据（Save）之前；
* 在修改数据（Update）之前；
* 在删除数据（Delete）之后。

上面的时机，是指通过EOS提供的数据服务（DAS）功能进行数据库的持久化操作中发生的。但在通过基础构件库进行数据的批量操作（如insertEntityBatch、updateEntityBatch和deleteEntityBatch）时，配置的实体拦截器将不起作用。

1、实体拦截器配置文件路径：EOSserver\working\应用名\config\handler-entity.xml,需要修改此文件，可以根据需要增加多个Handler，以下为用例。如图：



2、IEntityHandler用户需要实现该接口，实现类可以在对持久化实体操作的时候，执行特定的用户业务操作，达到对在执行持久化得时候进行拦截的目的。如图：



待实现方法：afterLoad、beforeLoad、beforeSave、beforeUpdate、afterDelete。

由于审计功能非常占用系统IO，在大压力情况下容易造成系统瓶颈，所以，为避免出现性能瓶颈，对于业务要求的业务字段或属性增加拦截功能，提供对于这些业务信息的审计功能。

#### 乐观锁

乐观锁一般是通过为数据库表增加一个“version”字段来实现。读取出数据时，将此版本号一同读出，之后更新时，对此版本号加一。完成对数据的加工处理需要提交写操作时，将提交数据的版本数据与数据库表对应记录的当前版本信息进行比对，如果提交的数据版本号大于数据库表当前版本号，说明在此期间没有其他进程对数据进行过修改，予以更新。否则认为是过期数据。当前操作失败，不予提交。对操作员来说，需要重新提交一次。

由于EOS平台的持久层基于开源项目Hibernate为基础，Hibernate支持乐观锁机制，并经过压力测试，性能没有问题，但EOS平台开发工具层面没有提供可视化的修改来为每个实体增加Version字段，故需要通过修改数据集反向生成的HBM文件来实现乐观锁。Hibernate通过HBM文件来实现类和实体表之间的对照关系等功能。通过手工操作在该文件中加入一个version字段和数据库中的乐观锁的字段对应，来完成乐观锁的功能。

#### 异常定义及处理

异常定义：

* BizException—业务异常，在事务执行中抛出此异常时，无需触发事务回滚机制。
* RollbackableBizException—业务回滚异常，在事务执行中抛出此异常时，需要触发事务回滚机制，需要继承BizException。
* BizIgnoreException—业务忽略异常，并非真正意义的异常，只是在逻辑调用中，通过抛出此事务来实现通知页面执行特定处理。
* WFException—工作流异常，用于封装工作流异常信息，并将异常信息返给应用系统内，应用系统内需要根据工作流异常信息来再次封装异常。
* RulesException—规则引擎异常，用于封装规则引擎异常信息，并将异常信息返给应用系统内，应用系统内需要根据规则引擎异常信息来再次封装异常。
* BaseRuntimeException—异常基础类，所有自定义异常类需要继承此类，此类将会继承java.lang.Exception类并实现java.io.Serializable接口。

异常处理：

1. 在构件包中建立异常资源文件，针对错误码，包括显示在页面的提示和真正的错误提示，例如：

auth\_0001=无效的用户名或者密码![用户{0}密码错误！]

其中{0}表示变量，如果异常提示中有多个变量，依次为{0}，{1}…[]内的信息为真正的错误信息。如果没有，则和前面信息一致。显示给用户的信息为[]前的信息。

异常资源文件为构件包资源配置目录下resources/exception/exception.properties

1. 错误码的构成规则：**构件包名+”\_”+四位数字**：例如auth\_0001
2. 构件包comm提供如下运算逻辑：

调用路径：com.cdb.EOS.uaap.comm.bizlets.BusiException.throwBusiException

接口：

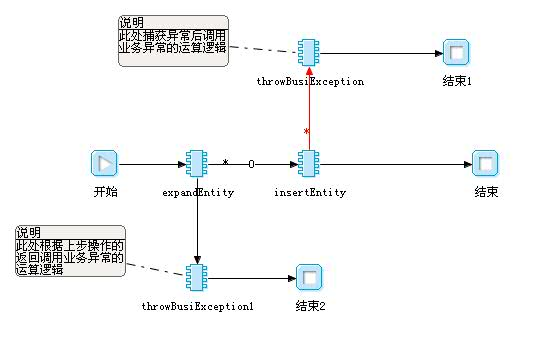
**publicstaticvoid**throwBusiException(String errCode, String logFlag,String... params) **throws** com.eos.system.exception.EOSException

第一个参数是资源文件中定义的错误码

第二个参数为是否写业务日志的标志，缺省为“true”，其他值为不写

第三个参数开始，为异常资源中的变量，当异常资源定义中，对应错误码的信息汇总有{0},{1},{2}，则需要设置第三、四、五个参数

1. 使用规范：
   1. 本运算逻辑可以代替逻辑流中的异常图元使用，但必须在之后增加结束图元，例如：



**图 18：异常控制**

调用了业务异常运算逻辑的情况下，不再需要调用写业务日志的运算逻辑。

#### 公共设计

提供实现业务功能的支撑及无业务含义的通用技术处理，为系统提供标准、统一、快捷的支持。

* 校验
* 主要用于保证数据本身的正确性和合理性；
* 包括客户端校验、服务器端校验，和接口数据校验；
* 缓存
* 主要用于存放系统中常用静态数据，避免过多访问数据库；
* 不发生变化的数据，如系统代码表，通常以常量类的形式存在于JVM或Context中；
* 常量类一般用Map来实现，可以用Lazy Load或者Pre Load来加载；
* 当加载开销超过JVM虚拟机设置堆栈内存大小的10%时，则认为其加载开销大；
* 如果加载开销较小，使用Lazy Load, 即在第一次使用时才载入；
* 如果加载开销较大，使用Pre Load,即在系统启动时预先载入；
* 日志
* 用于记录系统所发生的历史行为，从功能角度可以分为：业务日志、程序日志。
* 业务日志主要纪录主要的业务操作，面向业务系统管理员；
* 程序日志用于记录程序的参数，中间结果以及必要的调试信息。与业务日志相比，程序日志包含了更多实现层次的详细信息，主要面向开发人员和系统维护人员。
* 应用系统的日志信息等级从高到低包括：
* 致命（Fatal）：
  + 严重的错误，系统无法正常运行，如硬盘空间满等；
* 错误（Error）：
  + 系统可以继续运行，但最好要尽快修复的错误；
* 警告（Warn）：
  + 系统可以正常运行，但需要引起注意的警告信息；
* 信息（Info）：
  + 系统运行的主要关键时点的操作信息，一般用于记录业务日志。但同时，也应该有足够的信息以保证可以记录再现缺陷的路径；
* 调试（Debug）：
  + 系统运行中的调试信息，以便于开发人员进行错误分析和修正。一般用于程序日志，关心程序操作（细粒度），不太关心业务操作（粗粒度）；
* 跟踪（Trace）：
  + 系统运行中的跟踪信息，比调试信息更加详细和全面，更关心底层的信息，如参数，调用，数据项等；

通常在生产环境下只打开Info级别及以上的日志。在开发及测试环境才打开Debug日志。Trace日志一般不用；

* 文档
* 支持基于数据库创建PDF、Excel、Word格式的文件, 但不支持不同格式间的转换；
* 支持基于模板的PDF、Excel、Word文档或报表生成；
* 对于简单的PDF格式化文件，系统使用iText作为服务器端的PDF文件生成工具库。对于有复杂显示需求的文件，使用报表工具进行处理。
* 系统采用开源项目Apache POI作为操作Excel文件的工具库。
* 统采用开源项目Apache POI作为操作Word文件的工具库。
* 异常
* 在异常文件中(exception.properties)配置异常消息编码，如： code=message；
* 异常消息内容是需要经过UTF-8编码的；

逻辑流开发中为每个异常抛出配置异常编码，逻辑流层次抛出EOSRuntimeException异常，当捕获异常并抛出时，会将对应异常编码信息抛到上一层次；

### 系统物理部署

系统物理部署拓扑结构如下图所示：

 直销银行在线开户系统将部署在生产局域网内，内部用户可通过办公网进行访问。两台应用服务器采用集群方式部署，任意一台故障时不影响业务处理；用户请求通过F5进行分发；数据库服务器采用双机备份方式，数据存放在磁盘阵列。

## 软硬件配置

### 硬件配置

为提高系统的可扩展性、负载均衡、高可用性，系统部署水平集群环境，集群环境部署两个WAS应用服务器。

系统拟采用如下硬件配置：

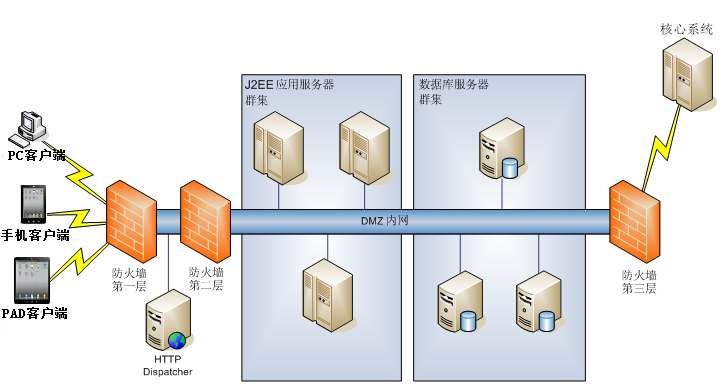
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **服务器/存储** | **数量** | **类型** | **配置** | **安装软件** | **操作系统** |
| 1 | 数据库服务器(HA) | 2 | 高性能PC Server | 4CPU32G | Oracle10g | Red Hat Linux V5.0 |
| 2 | 应用服务器(Cluster) | 2 | 高性能PC Server | 4CPU 32G | WAS V7 for Linux | Red Hat Linux V5.0 |

### 软件配置

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 系统设备 | 操作系统 | 应用软件 | 开发语言 |
| 1 | 数据库服务器 | Red Hat Linux | Oracle10g |  |
| 2 | 应用服务器 | Red Hat Linux | WAS V7 for Linux |  |

### 网络拓扑

直销银行在线开户系统不需要特定的网络布局，建议可采取典型的有防火墙的DMZ架构。网络结构如下图所示：



在该架构下，Web客户端的请求通过第一层的防火墙到达HTTP Dispatcher，然后通过第二层的防火墙到达J2EE应用服务器节点；如需要对遗留系统进行数据的读取，发出的请求将通过第三层防火墙而到达核心系统。

## 性能指标

性能需求描述系统性能方面的静态数值和动态数值方面定量的需求，包括：用户量、交易吞吐量、系统响应时间等。

* 系统支持在未来3年内在线开户数量达到100000。
* 系统支持在未来3年内同时在用户数量是10000。
* 系统支持并发用户数量是1000。
* 所有的目标响应时间需求都是基于用户浏览器提交请求开始到接收到全部重要输出信息的端到端的响应时间。
* 终端用户交互式访问系统的登陆界面的响应时间目标是小于2秒。
* 终端用户98%的简单查询访问的响应时间目标是小于2秒。
* 终端用户98%的复杂查询访问的响应时间目标是小于3秒，视查询的复杂程度。
* 终端用户98%的业务流程处理访问的响应时间目标是小于5秒。

系统性能相关指标及指标值如下表所示：

|  |  |
| --- | --- |
| **指标** | **指标值** |
| **㈠ 使用对象** | |
| 业务量（日登陆系统用户数\*每人每日操作系统次数） | 平均: 1000\*5=5000  峰值: 200\*10=2000 |
| 同时在线用户数 | 10000\*30%=3000 |
| 并发用户数 | 2000\*30%\*20%=120 |
| **㈡ 系统响应时间及可用性（**最大并发数情况下**）** | |
| 管理功能录入交易响应时间 | <1 秒 |
| 简单查询交易响应时间 | <2秒 |
| 复杂查询交易响应时间 | <3秒 |
| 流程交易的响应时间 | <5 秒 |
| 报表交易平均响应时间 | <5秒 |
| 每天维护（备份/数据清理等）时间不超过60分钟 | |
| 业务功能测试通过率达到100% | |
| 在投产1年内,可用性指标不小于99%（剔除由于运行维护不当导致的停止服务时间） | |
| **㈢ 批处理时间限制** | |
| 每日批处理时间 | <2小时内完成 |
| 月末、年末批处理时间 | <3小时内完成 |
| 批处理时效性 | T+1 |
| 约束条件 | 具体批处理完成时间依赖于上游系统供数时间 |

## 信息安全

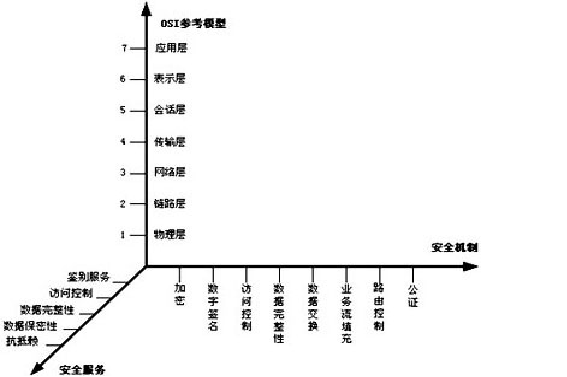
### 安全服务定义

随着北京银行管理信息化建设的推进与深入实施，应用系统的日常业务已经无法脱离网络和信息技术的支持。Internet是一个开放的公共网络，在此公共网络上开展、拓展传统业务，并维系系统日常正常运作的基础上，首要考虑的就是安全因素。如何有效保护自己的信息安全，应根据信息需要保护的程度以及系统安全设施情况，从多个侧面、多角度进行分析，从而制定相关保护策略。应用系统的安全体系应该是全方位的，而不仅仅是加密、防火墙或其他安全措施的堆砌。同时，“安全不仅仅是技术，而是策略、技术、管理的综合”，基于这一实施理念，在通过一定安全技术手段完成对信息内容的保护后，还需制定安全方面的相关制度，落实相关的具体执行措施，才能真正达到保护系统安全的最终目的。

当然，系统的安全设计与用户的需求以及应用环境密切相关。在系统的建设过程中，我们将根据用户的需求，充分发挥我方安全专家在长期从事安全系统设计过程中所积累的技术优势和丰富实践经验，与用户密切合作，共同建立完善的安全体系，确保系统安全可靠地运行。

在ISO 7498-2中描述了开放系统互连安全结构，其中提出设计安全信息系统基础架构应包含的内容：

1. 五类安全服务(鉴别服务、访问控制、数据完整性、数据保密性、不可否认性)；
2. 为安全服务提供支持的八类安全机制和普遍安全机制；



其中，五类安全服务分别为：

* 鉴别/认证（Authentication）

每一个计算机或计算机网络用户必须提供有效的名称标识已证明其合法身份。

* 访问控制（Access Control）

对于每一个经过认证的合法用户，根据预先定义的策略拥有相应的网络访问控制能力。

* 数据完整性（Data Integrity）

数据信息的内容在未经授权下，不能被有意或无意的改变。

* 数据机密性（Data Confidentiality）

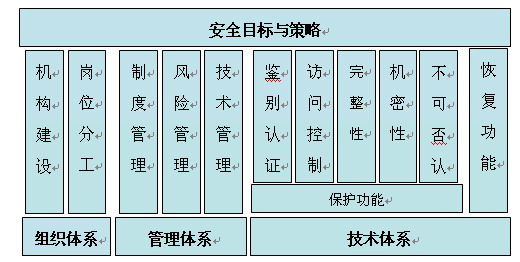
数据信息在未被授权的情况下不会被泄露。

* 抗诋毁/不可否认性（Non-repudiation）

数据信息的发送者（或接受者）不能否认其数据发送（或接受）的行为。

### 信息安全框架

针对北京银行技术规范要求以及行业内相关成功解决方案，在进行北京银行信息系统安全建设时，我们提出完整的信息安全框架，指导信息系统安全以及相关支撑体系的建设。



其中安全目标与策略为我们安全设计的实现目标，主要内容由三部分组成：组织体系、管理体系、技术体系。

组织体系，主要进行内部安全机构设置，人员、岗位的分工设置。

管理体系，主要包括设计安全策略与相应制度，并根据岗位分工进行风险管理以及技术管理。

技术体系，主要包括从技术角度系统应具备的保护、恢复两种技术能力。

在这三个主要体系中，由于任何安全管理或技术手段都离不开人员的组织与实施，因此，组织体系是整个结构的核心环节，而安全管理是组织体系进行技术体系实施和管理的具体操作，管理体系是整个结构中承上启下作用的重要环节，缺乏良好的安全管理结构体系，即使有了构建优秀的技术结构，仍然会面对无法有效管理使用的窘境，导致无法达到预期安全目标。技术结构是具体的各种安全功能和需求的技术实现，如防火墙，数据加密等技术产品都是通过不同的方式满足不同的安全保护功能。下图展现了三种体系间的关系。

技术体系

管理体系

三种体系间关系图

### 安全设计原则与考虑方法

#### 安全设计原则

* 遵照业界通行准则，完全遵循国际上有关的数据安全标准；
* 采用当前先进的数据安全技术和产品，并确保系统达到所设计的安全强度；
* 在安全性与系统效率之间取得平衡。由于安全程度与效率成反比，在设计安全系统时，我们尽可能地兼顾了系统效率的需求。
* 安全子系统与系统的其他部分之间界面清晰，接口明确，保持安全子系统的相对独立性。
* 系统所采用的安全产品是成熟、可靠的，系统能安全、稳定地运行；
* 严密、规范的密钥管理体系，解决密钥管理的重要课题，确保整个系统的安全；
* 在保证安全强度的前提下，降低系统前期投资和后期维护费用，并尽量使系统简单、易用；
* 安全子系统采用模块化设计，保持高度的可升级性、可扩展性。例如，支持密码算法的可替换性，支持向证书体系的过渡等；

现有安全方案要兼顾整个系统今后的业务扩展和系统升级、扩容，避免或减少以后因发展而可能带来的重复投资。

#### 安全策略

##### 物理安全策略

物理安全策略主要保证系统的物理安全因素，防范因为物理介质、设施位置，环境因素等意外的威胁。

##### 网络安全策略

安全策略就是规定哪些是受保护的信息，哪些是不被保护的信息。最基本的策略是禁止一切未被允许的活动。安全策略应考虑以下几个方面：

一、职责划分

安全策略要根据网络资源的职责确定哪些人允许使用某一设备，对每一台网络设备要确定哪些人能够修改它的配置。更进一步要明确的是授权给某人使用某网络设备和某资源的目的是什么，他可以在什么范围内使用。并确定对每一设备或资源，谁拥有它的管理权，即他可以为其他人授权，使之能够正常使用该设备或资源，并制定授权程序。

二、类型限制

安全策略还必须说明网络使用的类型限制，定义可接受的网络应用或不可接受的网络应用，要考虑对不同级别的人员给予不同级别的限制。网络安全策略声明每个用户都要对他们在网络上的言行负责。所有违反安全策略、破环系统安全的行为都是禁止的。

必须杜绝用户共享账号的情况。安全管理还要确定是否要为特殊情况制定安全策略。例如是否允许某些上级组织来检查系统的安全弱点。

三、授权管理

安全策略在确定对每个资源管理授权者的同时，还要确定他们可以对用户授予什么级别的权限。如果没有资源管理授权者的信息，就无法掌握究竟哪些人在使用网络。对于网络中的关键通信资源，对其可授权范围应尽可能小，范围越小就越容易管理，相对也就越安全。在对资源授权者管理的同时，要制定对用户授权的过程设计，以防止对授权职责的滥用。

安全策略中应明确指明每个资源的系统级管理员，但在网络的使用中，难免会遇到用户需要特殊权限的时候。一种处理办法是尽量只分配给用户够完成任务所需的最小权限。另外在网络安全策略中要包含对特殊权限进行监测统计的部分，如果对授与用户的特殊权限不可统计，就难以保证整个网络不被破坏。

四、用户管理

对于为用户初始化账号使用的口令，以及用户自己对口令的选择要非常慎重。因为对一个再好的网络安全系统，如果用户使用了很差的口令，那么系统的安全性也会很差。

五、安全策略

关于用户的权利与责任中，需要指明用户必须明确了解所用的计算机网络的使用规则。其中包括是否允许用户将账号转借给他人，用户应当将他们自己的口令保密到什么程度。用户应在多长时间内更改他们的口令，对其选择有什么限制。是希望用户自身提供备份还是由网络服务提供者提供。事实上网络安全策略中所能达到的只是用户希望达到个人稳私与网络管理人员为诊断、处理问题而收集用户信息的矛盾的一个折中。在安全策略中必须确定在什么情况下管理员可以读用户的文件，在什么情况下网络管理员有权检查网络上传送的信息。

六、恢复策略

安全策略中另一重要的部分是当安全策略被破坏时所采取的策略。对于发生在本网络内部的安全问题，要从主干网向部门网逐级过滤、隔离。部门网要与主干网形成配合，防止破坏漫延。对于来自整个网络以外的安全干扰，除了必要的隔离与保护外，还要与对方所在网络进行联系，以进一步确定消除掉安全隐患。每一个网络安全问题都要有文档记录，包括对它的处理过程，并将其送至全网各有关部门，以便预防和留作今后进一步完善网络安全策略的资料。

七、跟踪审计

必须有强有力的跟踪审计措施，对于谁、在什么时候、什么地方、以何种方式、对什么对象、做了什么操作、发生什么结果等都应该有详尽的记录。同时，审计记录应该加密存储，并且设置访问权限，确认只有授权的管理人员才可能获取到日志记录。

八、管理策略

在考虑网络安全时，网络安全管理策略是必须要考虑的一个重要问题。很多公司在考虑安全问题时，通常会比较注意技术上的考虑，对管理上的问题考虑得不够。往往技术上作了种种防范，但由于忽视管理使技术上的防范形同虚设。这就像买了一把好锁，但从不去使用它。我们认为一个好的安全管理策略比技术上的安全策略更加重要。

##### 应用安全策略

应用安全策略包括：系统安全策略、信息存储安全策略、防病毒措施几个方面。

系统安全策略主要包括：操作系统安全、数据库安全、数据存储安全几个方面。操作系统安全主要包括业务系统运行操作系统的稳定性以及安装维护时的安全控制策略。数据库安全策略主要包括用户角色的管理、数据保护、数据库备份。数据存储安全策略主要就数据库系统所管理的数据和资源提供安全保护。

信息存储安全包括两方面内容。一是信息访问的可控性。二是信息内容的隐蔽性。信息访问的可控性是只有被授权的、安全级别与数据的机密性要求一致用户才被允许访问相应的数据。而所有未经授权的用户，如黑客、恶意的内部用户，则不能对信息有任何的操作，包括读取、删除、复制等等。信息内容的隐密性是指即未经授权的人，即使采用各种手段获得了数据的访问权，实际的信息内容对他来说也应该是不可理解的。这主要通过数据库加密或各种文件加密等应用层加密来实现。

#### 安全解决方案

根据安全服务定义、安全设计的考虑方法与原则、安全策略，我们从物理安全、网络安全、系统安全、应用安全、数据存储、运营安全几个方面分别给出解决方案。

##### 物理安全

系统的物理安全是整个系统安全的基础，是系统安全的第一道防线。为了使系统的危险减至最低限度，首先需要选择适当的设施位置，同时要充分考虑水灾、地震、电磁干扰与辐射、犯罪活动以及工业事故等的威胁。

##### 网络安全

系统运行在北京银行内部网络系统中。内部提供Internet访问功能，因此在网络中存在着被攻击的可能性。为了避免这种情况的发生，建议在系统中使用防火墙和入侵检测系统以保证网络安全。

* + 1. **防火墙**

防火墙（Firewall）是一类防范措施的总称, 是一种非常有效的网络安全模型。在Internet上,通过它来隔离风险区域（即Internet或有一定风险的网络）与安全区域（Intranet）的连接, 但不妨碍人们对风险区域的访问。防火墙可以监控进出网络的通信量, 从而完成看似不可能的任务：仅让安全、核准了的信息进入, 同时又抵制对企业构成威胁的数据。

* + 1. **入侵检测系统**

防火墙提供了被动的安全保护，为了提高安全性，还可以增加主动安全措施，比如说入侵检测系统。入侵检测系统主要有两种，一种是基于主机的入侵检测系统，另外一种是基于网络的入侵检测系统。

利用入侵检测系统可以检测出那些能被入侵者利用的网络安全漏洞或者主机系统漏洞，并提供相应的修补建议。使用这种工具，系统管理员可以及时发现系统安全漏洞并予以修补。

#### 操作系统安全

操作系统安全指系统使用操作系统的稳定性以及安装维护时的安全控制。在系统选型时，首先要选择安全级别较高、稳定性强的操作系统，充分利用操作系统提供的安全机制，如登录身份认证机制、访问控制机制（如访问控制列表ACL）、日志和审计踪迹机制等；在安装服务器时去掉Telnet、FTP、NFS等冗余功能；经常从软件厂家和权威的安全站点去查询有关其漏洞的报告，并安装他们提供的补丁程序，以修补系统中已知的安全隐患和漏洞；限制超级用户的权限，根据“最小需要”的原则为用户分配权限。提高主机系统的安全可靠性还可以通过使用系统安全扫描工具来完成，这种工具可以发现系统软件本身固有的安全漏洞或由于配置不当引发的安全漏洞，并提供相应的修补建议。使用这种工具，系统管理员可以方便地加固系统。

#### 数据库安全

在应用系统日常使用中，数据库担负着日益艰巨的集中处理大量信息的任务，随着网络化开放环境的发展，数据库安全问题日渐突出。数据库安全方案主要包括三方面内容：用户角色管理、数据保护、数据库备份。

用户角色管理通过建立不同的用户组和用户口令验证，有效地防止非法的用户进入数据库系统，造成不必要的麻烦和损坏；另外在数据库中，可以通过授权来对用户的操作进行限制，即允许一些用户可以对服务器进行访问，也就是说对整个数据库具有读写的权利，而大多数用户只能在同组内进行读写或对整个数据库只具有读的权利。另外，特别强调对特殊账户的保密管理。

数据库的数据保护主要是数据库的备份，当计算机软硬件发生故障时，利用备份进行数据库恢复，以恢复破坏的数据库文件或控制文件或其他文件。另一种数据保护就是日志，数据库实例可提供日志，以记录数据库中所进行的各种操作，包括修改、调整参数等，在数据库内部建立一个所有作业的完整记录。再一个就是控制文件的备份，它一般用于存储数据库物理结构的状态，控制文件中某些状态信息在实例恢复和介质恢复期间用于引导数据库。

数据库备份主要有以下几种方式，逻辑备份、物理备份。其中逻辑备份主要指：将某个数据库的记录读出并将其写入到一个文件中，这是经常使用的一种备份方式。物理备份主要指系统维护人员对数据库的所有内容进行拷贝，方式可以是多种，有脱机备份和联机备份。

#### 应用安全

应用安全指在应用系统中实现的安全技术措施，以保证业务数据在操作流程中的安全性。应用安全从认证、访问控制、数据完整性、数据机密性和不可否认性五个方面进行保证。

同时，在应用安全的设计过程中，还应该注意安全措施独立性和灵活性的原则。安全措施独立性是指安全控制子系统和应用系统是非相关的，对安全子系统的任何改变不会影响应用系统。安全措施应该是一种黑箱操作，对应用系统来讲，调用某种安全措施的时候，只需将相应的安全请求送往安全子系统，安全子系统向应用系统返回结果，应用系统接收到结果后就可认为其已经经过安全处理，而不考虑安全措施是如何完成的。比如说，应用系统需要对某个数据进行加密，则只需将待加密数据送往加密子系统，加密子系统对其加密后返回给应用系统，应用系统则可认为数据已是加过密的，而不关心加密的实现过程。这样做的好处是可以采用灵活的安全控制措施。在实际的应用系统中，对安全级别的不同要求，对安全措施成本的考虑，都可能引起安全措施的多样化需求。采用安全措施独立性原则可以充分满足客户对安全措施灵活性的需要。

应用系统将对进入系统的数据进行合法性验证，杜绝非法数据污染数据空间。

应用系统将对试图进行的非法操作在主控台予以及时的反映，并根据操作级别设定向管理员进行告警，并记入应用程序日志。

* + 1. **用户认证、访问控制**

在线开户系统的用户使用Windows客户端，我们建议采用Windows域进行用户管理，利用证书进行系统用户认证的技术，并且采用J2EE权限管理策略实现访问控制。

 首先对一些背景知识进行简单的介绍：

（一）证书

公钥证书，通常简称为证书，是一种数字签名声明，将公钥值与拥有对应私钥的个人、设备或服务身份绑定到一起。可以颁发证书来执行各种功能，如 Web 用户身份验证、Web 服务器身份验证、安全电子邮件 (S/MIME)、IP 安全、事务处理层安全 (TLS)，以及代码签名等。证书还可以从一个证书颁发机构颁发给另一个证书颁发机构，以便创建证书层次结构。

（二）证书的安全机制

证书对许多安全服务都很有用。证书可以用于下列用途：

身份验证，验证某人或某物的身份。

隐私，确保只有特定用户才可使用信息。

加密，掩盖信息，使未授权的用户不能将其解密。

数字签名，认可信息并保证信息的完整性。

身份验证，身份验证对于安全通讯十分重要。用户必须在与另一方通讯时有能力证实双方的身份。网络上的身份验证很复杂，因为通讯各方在通讯时实际上没有接触。这会为不法分子截获消息或冒充另一个人或实体提供了可乘之机。

数字证书是提供身份验证的常用保密措施。证书使用加密技术解决物理上缺乏联系的通讯问题。使用这些技术限制不法分子截获、篡改或伪造消息的可能性。这些加密技术使证书难以修改。因此，实体要冒充他人很困难。

具体的方案如下：

利用windows2000建立域，为每个用户在域当中建立一个账号，这些账号同时也是应用系统的用户。利用windows2000中的证书服务建立企业级的证书服务器，这样向客户发放的证书是与客户一一对应的。当用户访问应用系统时，系统会得到用户的证书并取得用户的信息，再到数据库当中取得此用户在系统当中的权限信息并返回给客户。具体流程如下图所示。



这样，我们就有效的实现了用户的认证与访问控制。

* + 1. **机密性与完整性保证**

机密性保证是指信息不被非法窃听，这主要通过加密手段完成。如果以加密实现的通信层次来区分，加密可以在通信的三个不同层次来实现，即链路加密（位于网络层以下的加密），端到端加密（传输前对文件加密，位于网络层以上的加密）。

链路加密侧重与在通信链路上而不考虑信源和信宿，是对保密信息通过各链路采用不同的加密密钥提供安全保护。链路加密是面向节点的，对于网络高层主体是透明的，它对高层的协议信息（地址、检错、帧头帧尾）都加密，因此数据在传输中是密文的，但在中央节点必须解密得到路由信息。

端到端加密则指信息由发送端自动加密，并进入TCP/IP数据包回封，然后作为不可阅读和不可识别的数据穿过互联网，当这些信息一旦到达目的地，将自动重组、解密，成为可读数据。端到端加密是面向网络高层主体的，它不对下层协议进行信息加密，协议信息以明文形式传输，用户数据在中央节点不需解密。

加密可以采用软件加密和硬件加密两种方式。

根据网络层不同的传输协议采用不同的安全设计方案，如下图所示：

网络传输机密性保证

SSL加密

HTTPS、T3S、LDAP

TCP层

IP层（IPsec加密）

加密层次主要针对IP层、SSL层和HTTP（T3）层：

1、IPSec传输模式是同一个Windows2000域的两个基于Windows2000的系统之间实现程序通信的安全策略，为IP网络层（局域网或广域网）通信提供对应用程序透明的加密服务，并为Window2000操作系统提供其他网络访问保护。使用IPSec传输模式提供客户机到服务器、服务器到服务器和客户机到客户机之间的端对端安全；

2、SSL加密用于在Web访问时，对浏览器和Web服务器间信息交互进行加密，以及验证Web客户端和Web服务器的身份（相互身份验证）；HTTPS基于SSL，在Web客户端和服务器之间建立加密的HTTP连接。T3S是Bea Weblogic Server的专用加密协议，速度优于HTTPS。

3、对客户端IE浏览器和Web服务器之间的数据传输，为保证连接的安全性，对关键页面访问采用SSL技术进行加密。IE浏览器和Web服务器之间通过HTTPS、T3S建立安全的通信进程。

建议方案如下：

在windows2000域的中建立企业级证书服务器，利用这个证书服务器向web服务器分发web服务器证书和根证书；向客户端分发客户端证书和根证书，在Web服务器上进行相应的设置。这样，当客户端用https协议访问web服务器时就可以建立起SSL通道对传输的数据进行加密。服务器结构如下图所示。



这样，我们就有效的保证了传输数据的机密性与完整性。

#### 数据存储安全

数据将存放在数据库中，因此数据库的安全性对于整个系统的安全性是至关重要的。对数据库系统所管理的数据和资源提供安全保护，一般包括以下几点：

1. 物理完整性，即数据能够免于物理方面破坏的问题，如掉电、火灾等；
2. 逻辑完整性，能够保持数据库的结构，如对一个字段的修改不至于影响其它字段；
3. 元素完整性，包括在每个元素中的数据是准确的；
4. 对数据库中的存放重要数据的表采用DAC校验，防止非法修改；
5. 数据的加密，对重要字段如密码，采用DES算法用密钥加密存储；
6. 可获得性，指用户一般可访问数据库和所有授权访问的数据；
7. 可审计性，能够追踪到谁访问过数据库；
8. 制订合理的数据库备份和灾难恢复策略，防止数据库崩溃。

要实现对数据库的安全保护，一种选择是安全数据库系统，即从系统的设计、实现、使用和管理等各个阶段都要遵循一套完整的系统安全策略；二是以现有数据库系统所提供的功能为基础构件安全模块，旨在增强现有数据库系统的安全性。

数据不但在传输过程中可以保证完整性，还要保证数据在数据库存储时的完整性。为了保证存储在数据库中的数据的完整性，可以通过对数据库的关键字段产生DAC来实现，防止非法篡改数据库数据。也就是说数据虽然是明文方式存储在数据库表中，但数据可同时将关键信息以一定的算法构成另外一个字段存放在数据库中，操作发生时，系统将根据算法进行判断，也就是说任何手工修改数据库关键信息后，将无法进行正常操作。

#### 防病毒措施

为保证应用系统安全可靠的运行，建议采用以下几方面的防病毒措施：

* 桌面系统防病毒
* 服务器防病毒（WEB服务器、应用服务器）
* 网络防病毒（SMTP、HTTP、FTP）

建议采用企业级的防病毒软件，如NAI公司的McAfee以及Symantec公司的Norton企业级防病毒软件。