# 一、实现功能点

## 1.1登录用户体系

**主要功能描述：**

针对手机银行登录环节进行优化，实现对未整合的用户在登录时提示进行整合；支持使用网银用户名进行绑定；优化登录验密失败的提示语；对已整合且不符合新密码规则的用户强制改密。

**技术方案：**

#### 业务流程及界面



1、用户登录手机银行时（含普通登录和绑定的登录），判断该用户是否已进行整合，若未完成整合，则提示客户进行整合：“尊敬的客户，您可对手机银行、网上银行服务进行升级，升级后即可实现使用手机银行密码登录手机银行和网上银行”，提示页面显示升级和暂不升级按钮，暂不升级倒计时五秒后可点击。整合提示开关后期可进行修改或关闭。

2、若用户点击升级，则进行升级，升级时判断用户登录密码是否符合电子渠道密码要求，若不符合，则进入密码修改环节，若用户选择放弃修改密码，则回到提示升级页面。

3、在升级成功后提示客户升级完成，并可使用手机银行注册手机号、网上银行用户名登录手机银行和网银，手机银行登录密码成为手机银行和网银的唯一登录密码。

4、已升级的用户下次登录时，使用手机银行登录密码进行登录；已升级的用户下次绑定时，可输入手机银行注册手机号或网银用户名，与手机银行登录密码进行登录。

5、若用户点击暂不升级，则跳过升级流程，下次登录（含普通登录和绑定登录）时，继续判断是否整合并提示。

6、已升级的用户在登录（含普通登录和绑定登录）时，若验证密码失败，则提示用户“密码错误。今日再连续输错X次，您的手机银行将锁定无法登录。您已升级手机银行、网上银行服务，请使用电子渠道统一登录密码进行登录。”

#### 手机银行绑定优化

**功能描述**

针对手机银行绑定功能进行优化，支持客户输入手机号或网银用户名进行绑定。

**业务流程及界面**

**业务规则**

* 1、当用户输入11位1开头数字进行绑定时，作为手机号进行判断。
* 2、当用户输入的不是11位1开头的数字时，作为网银用户名进行判断。

#### 手机银行修改密码优化

**功能描述**

对手机银行修改密码功能进行优化，将对简单密码判断规则改为与网银一致。

**业务流程及界面**

同现有流程。

**业务规则**

采取渠道密码通用规则。

#### 手机银行在线开通优化

**功能描述**

当用户通过手机在线开通手机银行时，判断用户是否已开通网上银行，若已开通网上银行，则复用网银的登录密码。

**业务流程及界面**

1、用户已开通网上银行，通过手机银行客户端在线开通手机银行时，验证网银登录密码后，直接将网银登录密码复用在手机银行。

客户姓名：张三

银行账号：62270000000000000000

手机号码：18912341234

证件类型：身份证

证件号码：440011198801011122

开通版本：高级版

手机银行登录密码与网上银行登录密码一致。您可开通后登录手机银行或网上银行进行修改。

确定

**业务规则**

* 若用户为网银存量用户，未开通手机银行，则验证网银登录密码后，直接复用网银登录密码。

#### 手机银行设置用户名

**功能描述**

对已整合的手机银行用户，支持在手机银行设置电子渠道用户名，并支持设置是否仅使用用户名登录。

**业务流程及界面**

1、在手机银行安全中心中增加“用户名设置”菜单。

2、设置用户名时需验证短信验证码。

3、设置用户名之后可设置仅使用用户名登录手机/网银。

**业务规则**

* 仅已整合或新开通的手机银行用户可以设置用户名，未整合的老用户在进入设置用户名菜单时提示不可使用；
* 设置的用户名为电子渠道用户名，同一客户在手机银行修改用户名后，网银用户名同步变更，反之同样。
* 用户可在手机银行设置是否仅使用用户名登录手机银行和是否仅使用用户名登录网银。
* 若用户设置仅使用用户名登录手机银行，则用户在绑定设备时需输入用户名，若输入的是手机号，则提示用户不能用手机号登录。
* 若用户设置仅使用用户名登录网银，则用户仅能使用用户名登录网银。

#### 渠道信息同步

**功能描述**

手机银行的收款人名册和关注基金与个人网银可相互同步。

**业务流程及界面**

**业务规则**

* 支持客户同步和取消同步手机银行和个人网银的收款人名册和关注基金。新客户默认不同步。
* 对于已设置同步的客户，新增、修改、删除记录在手机银行和个人网银同步操作。对于未设置同步和取消同步的客户，新增、修改、删除记录在手机银行和个人网银分别操作。
* 客户同步和取消同步记录时，提示是否同步/取消同步记录，同步客户记录时，系统以手机银行、个人网银记录的并集进行同步，剔除重复项。
  + 如客户已开通手机银行和个人网银，客户确认同步记录，系统提示客户同步成功。
  + 如客户目前只开通一个渠道，不展示同步记录的链接，在客户两渠道均开通后再展示同步记录的链接。
  + 客户确认设置取消同步记录，手机银行显示同步前的手机银行记录和同步后的变更记录，个人网银显示同步前的个人网银记录和同步后的变更记录。
* 手机银行收款人名册同步个人网上银行：
  + 手机银行活期转活期和定期转活期的收款人名册增加个人网银的建行账户收款账户记录的个人活期账户记录；
  + 手机银行活期转定期收款人名册增加个人网银的建行账户收款账户记录的个人定期账户记录；
  + 手机银行向企业转账收款人名册增加个人网银的建行账户收款账户记录的对公账户记录；
  + 手机银行普通跨行转账和加急跨行转账的收款人名册增加个人网银的他行收款账户记录；
  + 手机银行信用卡还款收款人名册增加个人网银的信用卡还款的收款人名册；
  + 手机银行手机到手机转账收款人名册增加个人网银的建行手机银行客户收款账户记录；
  + 对于个人网银未保留的收款人名册内容（手机银行公益捐款）不予以同步。
  + 支持手机银行客户按收款人名册不同类别分别同步个人网银收款人名册和分别取消同步个人网银收款人名册。
* 手机银行的“关注基金”同步即增加个人网银的“关注基金”信息。

## 1.2转账汇款

**主要功能描述：**

(1) **“智能转账”**：整合目前的“活期转活期”、“向手机转账”、“普通跨行转账”、“加急跨行转账”、“向企业转账”、“信用卡转账”六个功能整合为“智能转账”功能。

* CCVEP将在“转账汇款”菜单下新增“境内转账”子菜单，以及在“我的转账”功能中添加对智能转账交易的入口。后续本功能上线并稳定后，将逐步替代上述六个功能的相关交易；
* CCVMB将在新版手机银行使用“智能转账”交易全面替代上述六个功能的相关交易。

(2) **整合转账收款人名册**:通过与手机银行的收款人名册合并，打通用户对转账渠道间的沟隔，达到“一处使用，多处通用”的体验效果。

(3) **我的转账**: 对客户已保存的收款人，,免去客户选择功能菜单以及填写收款人信息的步骤，甚至可以仅通过鼠标点击直接录入全部转账信息。

**技术方案：**

智能转账总览图

#### 获取收款行信息交易流程

**通过CCVEP/CCVMB自身识别收款账户所属行别信息**



注:本流程图以CCVEP为例绘制

* **设计原则和目标**

1、除了后端对卡BIN规则供数外，网银自身维护一张卡BIN规则学习表，用于对未能识别的卡号进行学习与人工维护。

2、对于账号识别，建行卡号的识别率需达到**100%**；建行存折或其他类账户可以通过学习或调用后端服务进行识别。对于未能识别的账号，系统则显示收款行为“中国建设银行”(客户可修改)。

3、如系统识别账号所属的收款行有误或未能识别,允许客户修改收款行信息。

* **收款账号识别处理过程**

1、用户输入收款账号(也可采用收款人名册列表选择)，光标移出输入域后,页面判断输入的收款账号是否为手机号。若账号是以1为首的11位数字，显示收款行为“中国建设建行(手机转账)”。

2、当客户输入的收款账号不是以1为首的11位数字，则当作账号处理，如收款账户为卡(16位和19位的账号)，网银根据账号查询标准卡BIN信息表，如符合建行的卡BIN规则，则页面返显收款行栏位为“中国建设银行”。如符合他行的卡BIN规则，则页面在收款行栏位返显对应他行的行名。

3、如收款账户不为卡，可能为建行存折或对公账号，系统调用成开的账户信息查询接口，如接口成功返回账户信息，页面则返显收款行栏位为“中国建设银行”。

4、如通过上述标准卡BIN信息表和账户信息查询接口均无法识别账号，系统查询账号识别学习表，如学习表中符合此账号放入规则学习记录，则页面在收款行栏位返显对应行名。

5、对于跨行转账场景，客户输入交易金额后，如交易金额未超过5万元且开通了加急跨行通道的收款行（系统默认工、农、中、交、招行已开通，可参数追加），无需客户选择网点，如超过5万元或未开通加急跨行通道的收款行，需走普通跨行通道，需展示收款省份、城市、收款网点栏位供客户选择。

【容错处理】

1、如不能识别收款行的账号，界面提示客户选择对应收款行。

2、如客户认为智能识别出的收款行有误，客户可以更换收款行。

【智能学习】

账户识别学习表用于对于系统未能识别的卡号以及存折进行学习和记录，由CCVEP/CCVMB进行维护。当账号类型为卡，当前系统未能识别由客户选择收款行信息的，系统会“记住”用户的选择，当在一段时间内客户的选择达到一定数量，系统认为由用户行为产生卡BIN规则具有通用性和可信性并将此规则记录到账户识别学习表。另外，此表的记录也可以由系统维护员进行人工维护。

#### 转账收款人名册整合



处理过程：

1、客户进行智能转账转账时，如点击收款人名册图标，网银从通过P1-SVC获取该客户的收款人名册并在页面返显收款人列表（包含收款人姓名、账号、收款行、别名等信息）。客户选择某一收款人后，自动填入收款人信息，包括收款人姓名、账号、行别。

2、客户进行转账时，也可从最近转账栏位中选择某个收款人，该栏位展示客户最近网银和手机渠道操作的8笔交易的收款人名称（数据P1-SVC从中获取）。客户选择某一收款人后，自动填入收款人信息，包括收款人姓名、账号、行别。

#### 标准卡BIN表数据更新流程

P1SVC

OJMS

CCVEP/CCVMB

1、SVC系统每天1点从武开参数管理系统(OJMS)下载标准卡BIN参数文件。

2、CCVEP/CCVMB系统每天2点从SVC“标准卡BIN参数文件”中下载卡BIN增量参数文件。

3、下载完成后，将数据导入到本地表标准卡BIN机构表(BC\_CARDBIN\_SERVICE)中。

#### 账户识别学习表自动学习与人工维护流程



* **学习流程**

1、客户完成本次转账交易后，智能转账功能会将此笔交易的卡BIN、实际收款行、系统识别收款行信息、转账交易结果推送至学习组件。

2、学习组件根据推送信息进行分析与统计。

（1）系统对如下情况进行归类：

* + - 标准卡BIN表已存在的卡BIN，学习组件**不进行**学习。
    - 当系统未能识别卡BIN而由客户手工选择了收款行，且转账交易成功的情况，视为新增规则类。
    - 当系统未能识别的卡BIN而由客户手工选择收款行，且转账交易失败的情况，视为操作异常类。

**注：。**

（2）对于上述情况的数据，系统会对相同情况的数据进行累计，如累计超过300笔（参数化配置）记录，且连续发生5天(参数化配置)的规则，系统认为该规则可信且有效，并记录到账户识别学习表中。

3、对于账户识别学习表中上述不同分类的规则，客户在智能转账交易时，当收款卡的卡BIN属于下列情况,会进行如下处理。

* + - 新增规则类：页面中返显的收款行别应为账户识别学习表中记录的收款行名。
    - 操作异常类：如客户选择的收款行与账户识别学习表中登记的收款行一致时，应提醒客户认真核对数据，避免转账失败。

4、系统维护员可以学习规则组件使用的参数以及账户识别学习表中的记录进行维护。另外应对定期导出账户识别学习表的记录进行分析，如后端系统提供的标准卡BIN表缺漏数据，则提醒后端系统尽快修正。

## 1.3代理保险

**主要功能描述：**

为进一步扩展和完善客户渠道的业务功能，更好地支持业务发展，拟在手机银行客户端、个人网上银行代理保险功能中增加车险产品代理业务，实现在线精确报价、实时投保，并支持在我的保单、购买记录功能中查询车险相关信息。新增万能险追加保费及部分领取等功能，并对试算功能进行优化。

#### 车险

1. 在保险超市增加车险产品，可通过保险搜索功能进行查询。支持车险作为推荐产品在保险超市首页展示。
2. 车险产品介绍展示内容与现有保险超市其他保险产品一致，由后端配置前端展示。未登录允许查看产品介绍详情。
3. 车险投保人：投保人是指与保险人订立保险合同，并按照合同约定负有支付保险费义务的人。被保险人是指其财产或者人身受保险合同保障，享有保险金请求权的人。投保人可以为被保险人。车主是指车辆的实际所有人。
4. 投保人默认为渠道签约客户本人，投保人姓名、证件类型及号码默认返显，不可更改。
5. 投保人、被保险人、车主可以为同一人，或非同一人。系统默认三者为同一人。
6. 客户可分别办理商业险或交强险与车船税，但单独购买商业险需已购买交强险。交强险与车船税需同时购买，具体规则由银保系统确定。
7. 支持为本人及他人车辆购买车险。
8. 上一年度已投保客户输入车辆信息后返回被保险人、车主等信息，支持修改，提交后系统默认返显客户上一年度已购买车险情况，客户可修改。
9. 新车未上牌中车辆信息由后端返回给客户选择，返回信息以后端返回为准。
10. 在核保前界面提供自动“暂存保单”功能，并支持客户在投保过程中主动暂存相关信息。
11. 车险投保要素由后端返回后展示。
12. 车险仅支持以二代居民身份证签约个人手机银行的客户办理。
13. 缴费账户支持为手机银行签约和非签约账户，账户类型包括龙卡通、活期存折、准贷记卡、理财卡。
14. 其余投保规则同现有保险投保规则一致。
15. 车险缴费规则同现有缴纳保费规则一致。
16. 非实时投保的车险保单支持在手机银行进行契撤交易。
17. 车险购买成功后支持分享至社交网络平台。客户点击分享按钮后，选择要分享的社交网络平台路径后完成分享。
18. 在“我的保单”、“保单购买记录”中增加车险信息。

#### 万能险追加保费

1、支持对已生效且为有效状态的万能险保单进行追加保费。

2、保单是否支持追加保费状态由银保系统返回。通过“我的保单”-“账务交易”，选择“追加保费”进行交易，交易结果发保险公司审核后返回。

3、电子渠道仅支持在电子渠道投保的保单，不支持纸质保单进行追加保费。

4、追加保费的金额必须大于零，允许输入小数，金额根据保险产品由保险公司配置。

5、对于追加保费成功的万能险保单，由系统判断校验不支持犹豫期退保。

6、万能险追加保费部分支持当日追加撤销。

7、追加保费需收取手续费，手续费由保险公司收取。

8、界面展现要素以后端系统返显为准，支持参数配置。

#### 万能险申请部分领取

1. 支持对已生效且为有效状态的万能险保单进行申请万能险现金价值部分领取。
2. 保单是否支持部分领取状态由银保系统返回。通过“我的保单”-“账务交易”，选择“申请部分领取”进行交易，交易结果发保险公司审核后返回。
3. 电子渠道仅支持在电子渠道投保的保单进行申请部分领取交易。
4. 部分领取金额入账账户仅限投保人手机银行签约活期储蓄账户，账户类型包括龙卡通、活期存折、准贷记卡、理财卡等。
5. 可领取金额由保险公司返回，申请部分领取金额不可超出允许的可领取金额，且须大于零，允许输入小数。
6. 申请部分领取需收取手续费，手续费由保险公司收取。
7. 界面展现要素以后端系统返显为准，支持参数配置。

#### 试算优化

1、由银保系统返回保险产品是否支持试算。支持试算的产品，在产品介绍详情页面展示“试算”功能按钮。

2、产品试算要素由银保系统返回，支持模版参数化配置。

3、客户输入试算信息，点击试算按钮，成功获取试算结果。试算结果界面，新增“投保”按钮，点击进入投保界面，系统默认自动带入客户填写的试算信息，仅“保费”及“保额”栏位带入保险公司返回的试算结果。

4、其他规则、流程与现有规则、流程保持一致。

5、界面展现要素以后端系统返显为准，支持参数配置。

#### 报表统计

将车险投保、追加保费、申请部分支取等交易额算入代理保险交易额中。

将车险投保、追加保费、申请部分支取等交易量算入代理保险交易量中。

本渠道投保的保单发生当日撤单、退保交易在统计中减去对应交易额和交易量。

统计纬度：统计时间、统计机构

## 1.4基金资讯服务

## 1.5智能安全

**主要功能描述：**

根据账务类、登录类将现有交易分类，列举出较有代表性的功能场景.从通用盾的安全性，指纹、手势密码的便捷性出发，结合用户使用频率、交易对象、交易地点，进行新的设计。

**技术方案：**

#### 绑定智能安全（优化半绑定）

P1(手机银行)

P1-SVC

移动银行客户端

交易过程：

1.客户端发起登录交易，服务端判定是否需要加验账户密码，如需要加验，则返回客户端进行验账密交易。验证成功后，重发绑定交易成功。

#### 转账智能安全

移动银行客户端

P1(手机银行)

P1-SVC

P8支付结算

P4

交易过程：

1.客户端发起转账交易，手机银行服务端判断是否符合便捷转账，还有判断是是否异地大额，返回可用的安全验证方式。可视以上情况返回客户端使用对应的可用安全校验方式或者加验安全校验。

#### 缴费智能安全

移动银行客户端

P1(网站)

P1-SVC

P4

P8支付结算

交易过程：

1.客户端发起转账交易，网站服务端判断是否符合小额指纹缴费条件，返回校验标识到客户端，客户端校验指纹成功后，后续交易路径保持不变。

#### 支付智能安全

移动银行客户端

P1(网银)

P1-SVC

P4

P8支付结算

交易过程：

1.客户端发起转账交易，网银服务端判断是否符合小额指纹支付条件，返回校验标识到客户端，客户端校验指纹成功后，后续交易路径保持不变。

## 1.6动态换肤

**主要功能描述：**

动态换肤主要根据客户的需求，由客户选择自己喜爱的颜色，在App运行过程中更换各个页面组件的颜色以及字体的颜色，改变App的整体风格，增加用户粘性。

**技术方案：**

整体流程



1、由客户端(App)定义换肤接口并定义换肤方法，所有需要换肤的控件(例按钮、输入框、图片控件)实现该接口，在接口换肤方法中实现各自需要的换肤效果(例按钮需要改变点击效果图片、输入框需要改变字体颜色、图片控件需要改变图片颜色)

2、定义换肤控制器，用于打开页面换肤逻辑处理。

3、定义换肤数据中心，用于保存用户选择的换肤颜色以及页面是否已经换肤标识

4、换肤整体流程

程序启动——初始化换肤数据中心以及控制器——打开页面——换肤控制器获取颜色传递给页面——页面遍历当前需要换肤的控件传递颜色——换肤控件调用各自换肤的实现方法。

## 1.7页面布局配置

**主要功能描述：**

#### 实现多种样式的页面展示效果

在界面设计上以文本、图片、输入框、选择项等基本界面元素为基础，结合不同平台操作特性，引入九宫格、弹出框等特色界面，同时对客户的特殊手势、按键操作等做出响应。

基本界面元素可以满足客户输入、选择等基本操作要求，扩展元素作用主要体现在美化基本元素界面和构建复杂界面元素上。界面布局由页面元素定义决定，允许使用内嵌页面，如在每个交易页面中内嵌快捷菜单页面，输入框支持历史输入和常用输入选择，密码输入支持安全键盘。

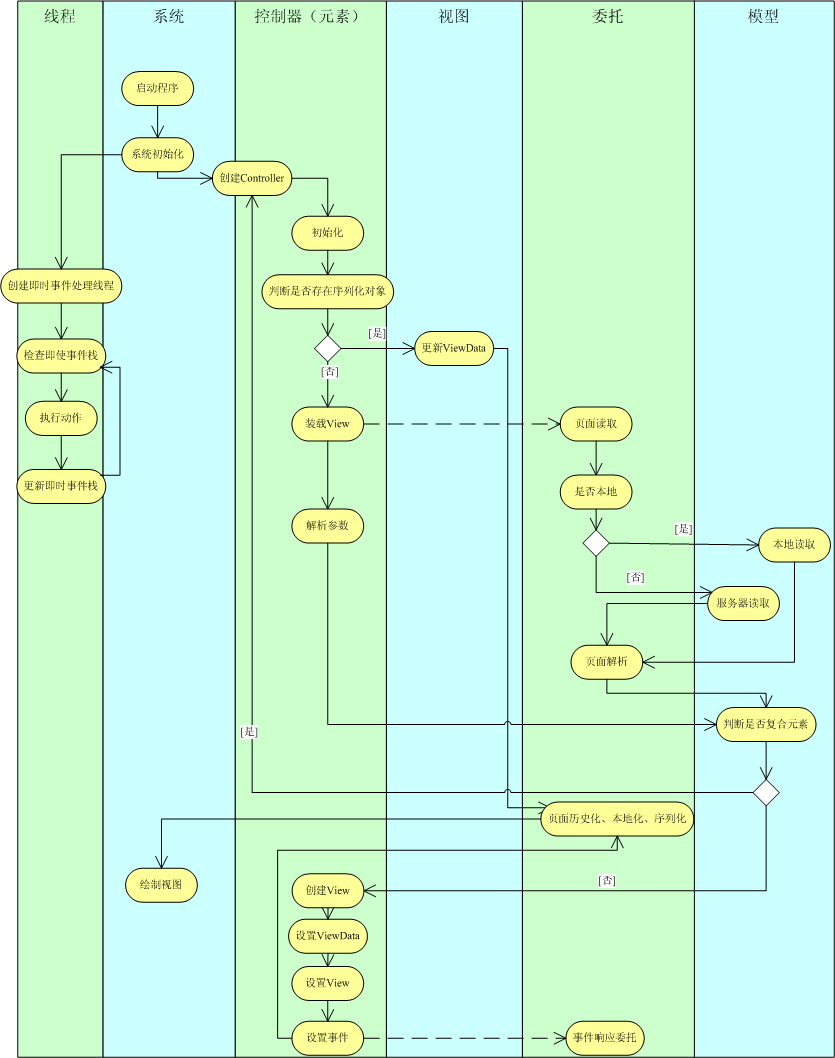
对于客户常规操作进行丰富展现，对于平台下特性操作如触摸、滑动、重力感应等进行定制响应，界面展现上满足客户对体验丰富性的要求。如支持客户滑动菜单项对菜单位置进行调整，多点触摸进行界面放大缩小，自动横屏切换，交易操作及界面切换支持动画效果，晃动更换主题。

#### 参数化配置的设计

系统中界面布局、控件样式皆有页面文档定义，事件委托、响应依赖于页面元素的扩展和系统UI库的实现支持，完美的实现界面配置化，通过调整页面文档及元素，可动态调整已有界面布局及样式。

**技术方案：**

应用处理流程图如下：



#### 主流程

1. 程序启动，依赖于操作系统功能完成；
2. 系统初始化对系统环境、系统参数进行初始化，完成基础数据准备工作；
3. 启动线程，对事件进行管理，此处管理事件为即时类事件，即与客户操作异步进行的事件，采用栈对事件进行管理，进入支流程1；
4. 创建视图控制器，目的是创建出系统展现的视图；
5. 对控制器进行初始化，定义视图控制器自身属性；如果存在序列化视图控制器对象，则直接启用，并且更新控制器中视图（view）对象的视图数据（VeiwData）对象，进入步骤13）；否则，进入步骤6）；
6. 试图控制器显式发送消息给视图生成委托（ViewDelegate）对象，消息分两种，页面参数和元素参数，如果启动视图加载，使用页面参数，进入步骤7）；如果页面视图加载过程中递归，使用元素参数，进入步骤9）；
7. 委托类读取页面，如果页面本地化，则向服务定位器发送本地读取消息，进入步骤8）；否则发送服务器通讯消息，进入步骤8）；
8. 委托类进行页面解析，解析出元素时向视图生成模型发送消息；
9. 视图生成模型判断元素类型，如果是复合元素，则创建视图控制器，进入步骤4）；否则，进入步骤10）；
10. 生成视图（view）对象，设置视图（view）对象视图数据（ViewData）对象；
11. 委托完成，控制器设置视图（view）对象，处理视图（view）对象层次；
12. 控制器关联视图所属事件与委托；
13. 控制器显式发送消息给视图委托（ViewDalegate）对象，执行视图存储动作；
14. 视图委托（ViewDalegate）对象判断如果需要序列化且未序列化，将控制器对象序列化；如果需要本地化， 且未本地化，将页面流做文件本地化；
15. 完成视图加载，系统绘制视图。

#### 支流程1

1. 线程检查即时事件栈，判断事件要素，执行事件；事件要素作为事件属性，决定事件执行条件、执行结果处理方式等等；
2. 执行完成，更新即时事件栈，发送指定消息，进入步骤1）。

#### 支流程2

1. 客户操作触发事件；
2. 事件委托类执行相应动作，完成对事件的相应，更新视图数据（ViewData）对象，通过视图数据（ViewData）对象改变观察者状态；

视图数据（ViewData）对象向所有观察者发送消息，完成视图更新；

## 1.8白名单机制

**主要功能描述：**

白名单的目的是在创新功能未正式推出前，先给部分建行客户提供试用。因此需要建立创新功能白名单，要在白名单范围内的客户才可以使用，否则默认不显示创新功能。

**技术方案：**

白名单方案整体流程图



1. 判断创新白名单控制开关是否开启，未开启则返回验证通过，如果开启则取技术版本号；
2. 如果技术版本号不存在，返回默认白名单类型和默认版本类型；
3. 如果技术版本号存在，则根据USERTOKENT查询版本信息和白名单信息；
4. 判断版本号是否在配置版本范围内，如果不在则返回白名单类型和版本类型，另外如果是新版本且没配置在后台的，返回未配置版本提示标识；
5. 如果版本号在配置范围内则判断版本类型，如果是业务版直接返回白名单类型和版本类型；
6. 如果是技术版则判断日期是否在配置范围内，如果不在则返回异常；
7. 如果在配置范围内则判断是否在白名单内，如果在则返回白名单类型和版本类型。

如果不在白名单内，则返回异常

## 1.9框架技术支持

**设计思路**

一体化是系统最基本的建设原则。只有真正实现了一体化才能真正合理地利用企业资源，实现信息高度共享，为企业管理创造效益。一体化可以从以下几个方面来理解：

**1、** 数据的一体化

1. 必须站在全局角度进行数据结构的分析和设计，充分考虑数据的完整性和共享性以 及相关性；
2. 2） 数据集中存放和管理。

**2、** 体系框架一体化 按一体化思想构造软件体系结构，合理建立对象模型。

**3、** 应用的一体化 业务相互关联，协同工作，本身就是一体化的。所以构造相关应用时必须考虑一体化的问题。

**4、** 与其他系统的集成 必须与其他管理系统进行集成，处理好与它们的接口，保证整个的一体化。

**平台化**

建立统一的系统基础平台，以达到如下目的：

1、 便于系统升级扩充；

2、 便于整合其他系统；

3、 有利于系统稳定性；

4、 减少系统研发工作量，提高研发工作效率，缩短项目实施周期。

**标准化**

系统开发管理过程需按照IT行业规范进行的，采用CMM的思想进行精细化管理，不管是软件结构、功能、界面、文档、过程有规可寻。

**开放性**

系统必须坚持开放性原则。系统肯定面临整合目前现有其他系统和升级改造以及未来接入其他系统的问题。而解决这些问题的根本做法就是建设开放的软件体系结构。

**方便实用**

实用性指的是系统是否能切合用户的实际需要。是否符合操作人员和管理人员工作习惯。是否符合行业的管理规范。

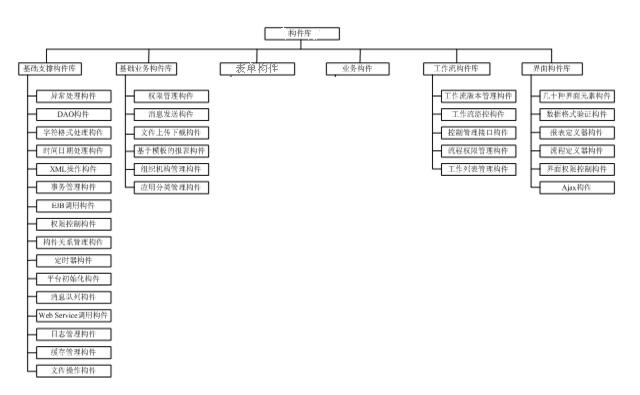
**采用先进的管理思想**

建立合理的开发流程及团队成员密切的合作; 建立共同的工作框架、规范; 学习一些成功开发过程、分析方法、设计思想、体系结构、设计模式等. 学习合理统一开发过程（RUP）的一些实践;

**先进的软件开发技术**

采用先进的基于java的J2EE技术，构造稳定的，开放的，安全的技术平台。采用B/S软件体系结构。 采用Spring ,EJB, Hibernate,WebWork2等成熟软件框架。 采用UML，按照RUP方法管理开发过程。



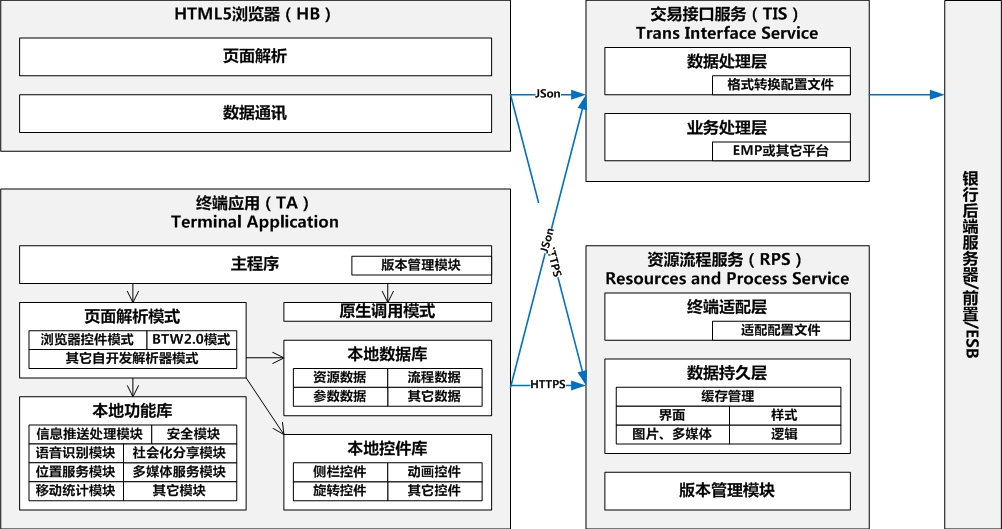


系统基础架构是整个平台的基础，处于整个系统的最底层，其它的业务系统都构建在它的上面，它为业务系统提供服务，基础架构包括视图控制层，业务逻辑层，数据持久层等。 系统基础架构的设计和实现必须满足稳定性，安全性，高效性和可扩展性等技术要求。在系统基础架构的设计和实现中我们引进了面向切面（AOP）的编程技术、MVC WEB编程技术和O/R Mapping技术等多种先进的IT技术；同时我们对系统采用了多层结构，整个基础

# 二、系统技术架构

## 2.1技术架构

### 整体架构图

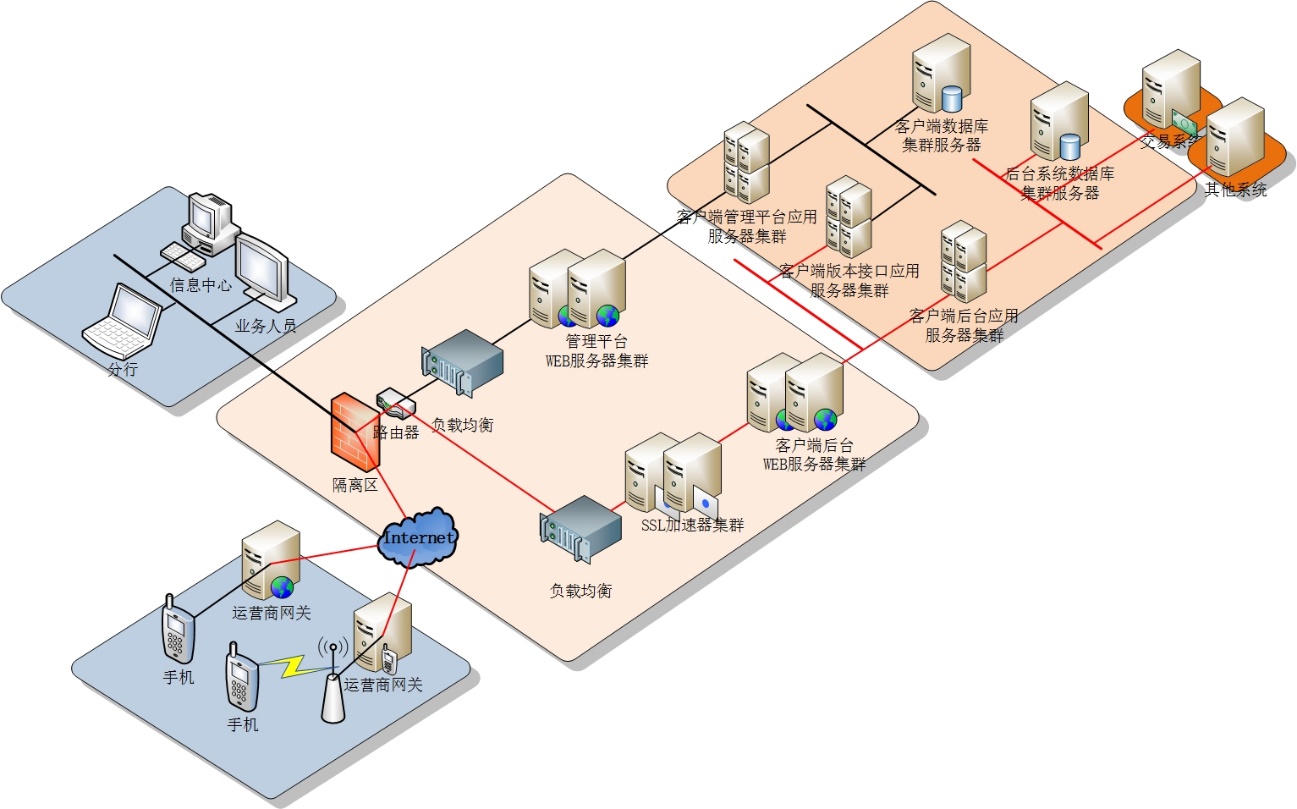


整体架构图

系统服务端划分为交易接口服务（TIS）和资源流程服务（RPS），终端应用和浏览器根据访问需求分别访问，实现界面展现、业务风格化和核心业务的分离。

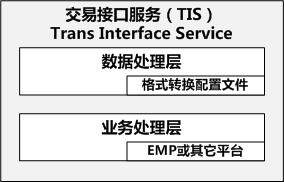
* 交易接口服务（TIS）：向外提供移动金融交易接口，当前封装为JSon接口。
* 资源流程服务（RPS）：组织交易流程，展现交易界面，提供展现资源。
* 终端应用（TA）：运行于终端的移动应用。
* HTML5浏览器（HB）：通过浏览器直接访问服务端业务。

### 网络结构图



### 交易接口服务（TIS）架构

* 概述



交易接口服务（TIS）架构图

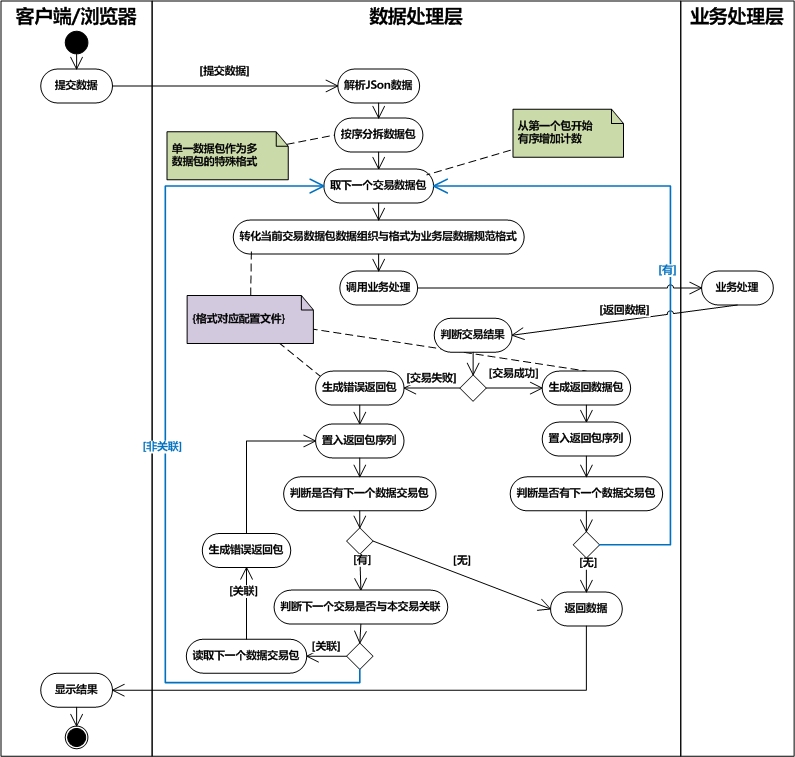
交易接口服务器（TIS）分为数据处理层和业务处理层两个大层，职责划分如下：

1. 数据处理层

* 对外部输入数据进行分拆、解析、转换，作为业务层的数据输入
* 对业务层的输出数据进行整合转换，返回外部数据请求端
* 数据处理层仅检查通讯应用层JSon协议子层数据传输的正确性，不对JSon协议子层之上的业务数据做任何处理

1. 业务处理层

* 安全认证：负责移动金融相关的安全认证
* 业务整合：根据移动金融业务特点，整合自身、后端服务的相关原子交易，向外部提供独立的、完整的业务服务
* 交易流程及业务要点



交易接口服务（TLS）流程图

1. 终端提交数据到交易接口服务器，数据格式在后续接口规范章节描述；
2. 交易接口服务器数据处理层解析JSon数据；
3. 数据处理层分拆数据包。数据包支持多包格式，单一数据包为多包的特殊格式；
4. 数据处理层取下一个交易数据包。如果是第一次取，则取的是第一个交易数据包；
5. 数据处理层根据格式对应配置，转化本交易数据包为业务处理层不要的数据包。当前版本可直接把数据转化为业务处理层（EMP）需要接收的数据；
6. 数据处理层调用业务处理层进行业务处理。本质上是流程的转移；
7. 业务处理层进行业务处理，并返回业务处理结果数据给数据处理层；
8. 数据处理层判断交易结果，如果交易失败，转步骤9，如果交易成功，转步骤15；
9. 数据处理层根据格式对应配置，生成JSon格式的错误返回包；（本错误返回包数据JSon协议包体部分）
10. 数据处理层把错误返回包置入返回包序列；（此步骤为应对多包组合情况）
11. 数据处理层判断是否有下一个数据交易包，如果有，转步骤12，如果没有，转步骤18；
12. 数据处理层判断下一个交易是否与本交易关联，如果有关联，转步骤13，如果无关联，转步骤4；
13. 数据处理层读取下一个交易数据包；
14. 数据处理层针对下一个交易数据包，直接生成错误返回包，转步骤10。（这么处理的原因是：如果是有序的关联交易，前面交易出错，后续交易则不再进行，直接生成错误返回包）；
15. （接步骤8）数据处理层生成数据返回包；
16. 数据处理层把返回包置入返回包序列；（此步骤为应对多包组合情况）
17. 数据处理层判断是否有下一个数据交易包，如果有，转步骤4，如果没有，转步骤18；
18. 数据处理层返回交易完成的数据包（单包或多包）给终端；
19. 终端显示结果，结束。

### 资源流程服务（RPS）架构

* 概述



资源流程服务（RPS）架构图

资源流程服务器（RPS）分为终端适配层、数据持久层和版本管理模块三部分，职责划分如下：

1. 终端适配层

* 获取终端特征信息并持续更新维护；
* 根据终端特征，针对性选择资源返回；
* 根据需求，对不符合终端格式要求的资源进行转换。包括格式转换、资源的增加/删除/修、数据压缩、数据打包、数据签名等；

1. 数据持久层

* 资源的获取和缓存

1. 版本管理模块

* 资源版本的维护与管理
* 交易流程及业务要点

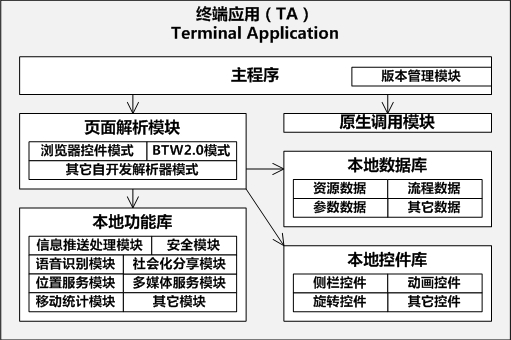


资源流程服务（RPS）流程图

1. 终端提交请求道资源流程服务器；
2. 资源流程服务器终端适配层获取终端特征；
3. 终端适配层根据终端请求和特征到数据持久层获取资源；
4. 数据持久层判断本资源是否需要且已经缓存；如果没有缓存，读取资源并缓存；
5. 数据持久层向终端适配层返回资源数据；
6. 终端适配层判断是否需要进行数据转换和压缩，如果需要，则进行数据转换和压缩；
7. 终端适配层向终端返回数据；
8. 终端显示结果，结束。

### 终端应用（TA）架构

* 概述



终端应用（TA）架构图

终端应用（TA）分为原生开发模式、页面解析模式以及两者的混合模式（Hybrid模式），当前主流的金融应用，大多采用混合模式，只是原生部分和解析部分的比例各不相同。而其中的页面解析模式也根据使用方式的不同而有很大差别，具体划分如下：

1. 页面开发遵循标准协议（如HTML5），终端调用浏览器控件进行全部解析
2. 对页面协议（如HTML5）进行一定的调整，部分功能、界面调用原生的方式实现，其它调用浏览器控件进行解析。
3. 自定协议规范，完全自身实现BS模式。协议规范需包含界面展现、界面布局、多媒体资源、流程逻辑等几个部分

以上三种方案可根据需要组合选择，本方案不对具体方案进行选择，仅描述其基本组成和共性部分。如果方案2无法通过验证，则无法满足界面风格灵活性的要求，需要对复杂的界面采用原生或自定协议规范实现。

终端应用（TA）分为主程序、页面解析模块、本地功能库、原生调用模块、本地数据库、本地控件库和版本管理模块共七个主要模块组成，职责划分如下：

1. 主程序

* 主程序负责终端应用的初始化及生命周期管理。任何模式都需要本模块

1. 页面解析模块

* 负责各种不同类型的页面解析。页面模式和混合模式需要有本模块

1. 本地功能库

* 一些本地功能库，如摄像头、二维码、安全加密等功能

1. 原生调用模块

* 原生功能

1. 本地数据库

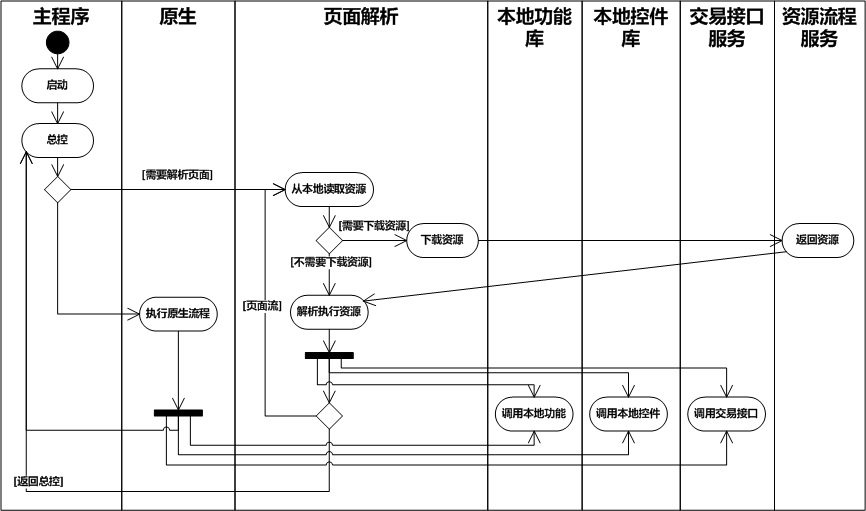
* 数据持久化管理

1. 本地控件库

* 本地的界面控件

1. 版本管理模块

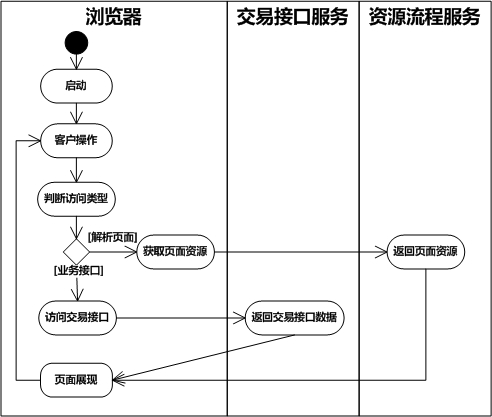
* 版本维护
* 交易流程及业务要点



终端应用（TA）流程图

1. 启动主程序
2. 进入总体控制部分
3. 总控根据条件，分别调用页面解析或原生模块
4. 页面解析和原生模块在执行时，可分别调用本地功能库、本地控件库，以及与交易接口服务器进行通讯
5. 页面解析模式在必要时从资源流程服务器获取资源

### 浏览器模式架构

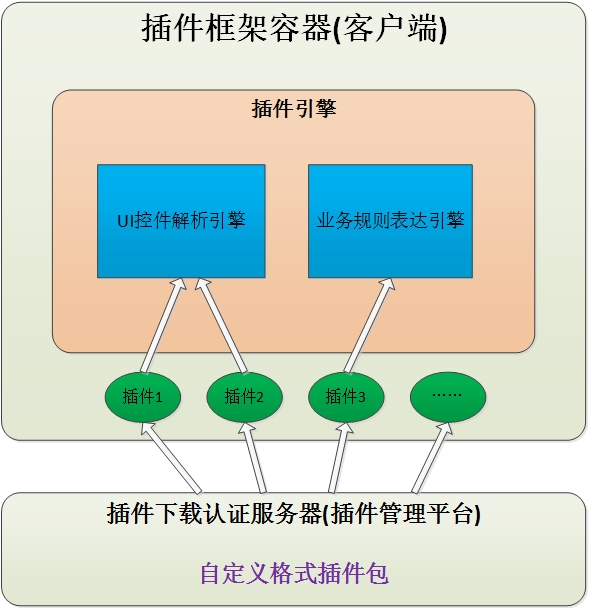


本质上，浏览器模式是终端应用（TA）的一种特殊情况，可看做是纯B/S模式的终端应用，其通讯协议需严格遵循通用页面规范。

因无法保证缓存页面的安全性，浏览器模式不得采用HTML5的缓存机制。

**3.2.1.7产品可拓展性插件化**

* **整体机制**



整体机制上包括客户端、插件管理平台、插件包三部分。具体说明如下：

* **客户端**

包括插件框架容器、UI控件解析引擎以及业务规则表达引擎。

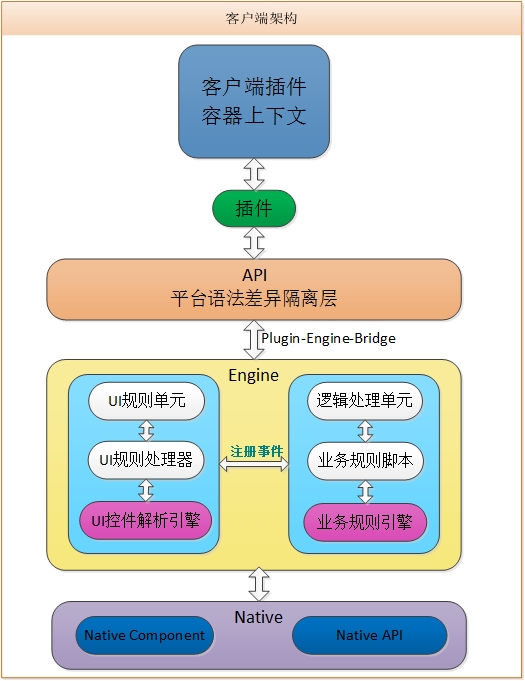
* **插件管理平台**

用于管理所有的插件包，以及对客户端提供插件下载查询功能。

* **插件包**

单个业务功能程序包的集合。

* **客户端**



整体上包括插件框架容器部分、API语法层、UI控件解析引擎部分、业务规则表达引擎四个部分。

* **插件框架容器**

插件运行时环境，能够对插件包进行组织和调用引擎运行。

* **API语法层**

定义统一的API语法，屏蔽平台差异，各平台引擎分别根据API，在引擎中实现。

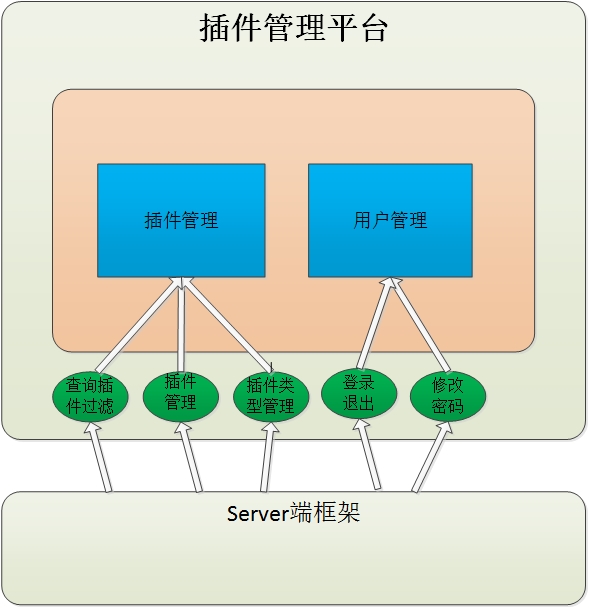
* **UI控件解析引擎**

用于解析插件包中的UI描述语言，转换为本地控件，并注册事件。

* **业务规则表达引擎**

处理UI事件的，能够在事件中进行一些业务逻辑处理（包括：数据获取、数据校验、数据格式化、控件属性设置、UI切换、系统级API调用）

* **插件管理平台**



整体上包括插件管理部分、用户管理部分。

* **插件管理**

包括插件类型管理部分，和插件管理和客户端查询插件过滤部分。

* **用户管理**

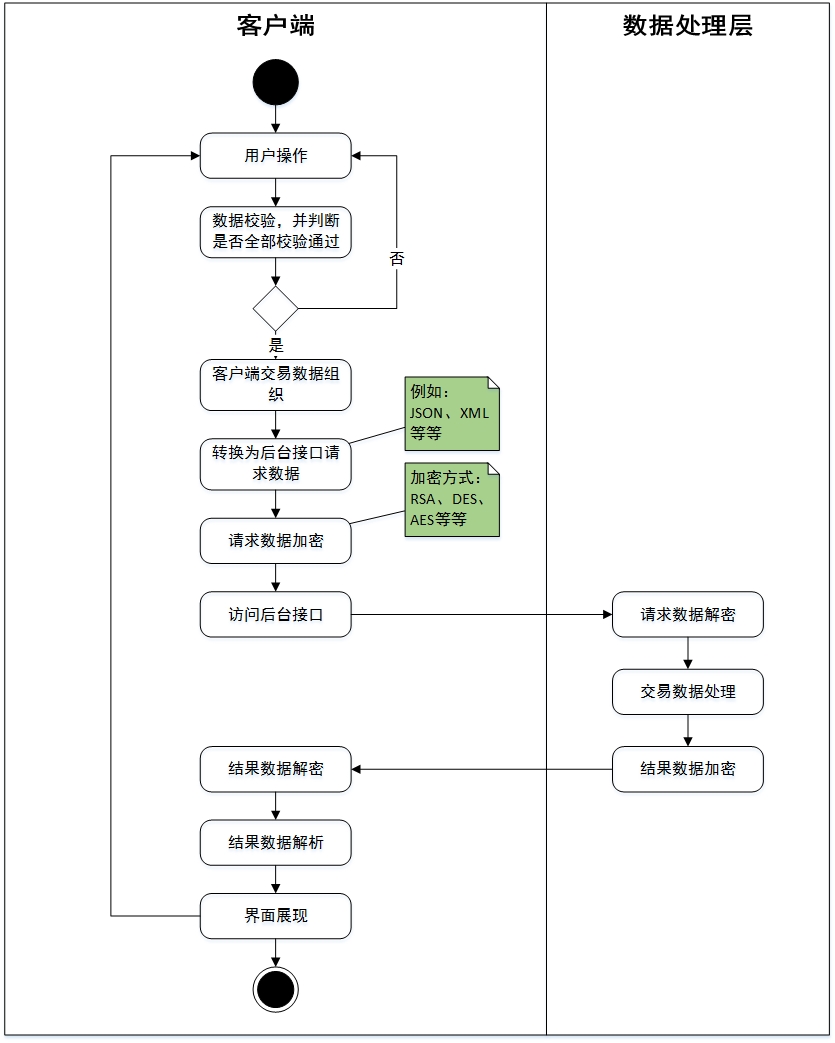
用于用户的密码修改,和用户登录退出的功能。

* **客户端查询插件过滤**

在客户端登录后发送查询插件请求，返回插件查询列表。

## 2.2处理流程

### 客户端数据交互流程



**具体处理流程描述如下：**

1. 用户进行相关操作。
2. 客户端根据用户的操作，进行数据校验通过后，组织用户交易数据。
3. 将组织好的交易数据转化为后台接口要求的通信数据报文。
4. 将通信数据报文进行加密处理，加密方式例如：RSA、DES、AES等等，该场景比较推荐RSA加密方式。
5. 请求后台接口处理。
6. 后台首先进行数据解密，然后进行业务处理，之后将处理完成的交易结果数据报文进行加密处理。
7. 客户端收到结果数据后，首先进行解密处理。
8. 客户端对结果数据进行解析并显示客户端交易处理界面。
9. 客户继续操作，返回步骤1。

### 后台交易接口处理流程

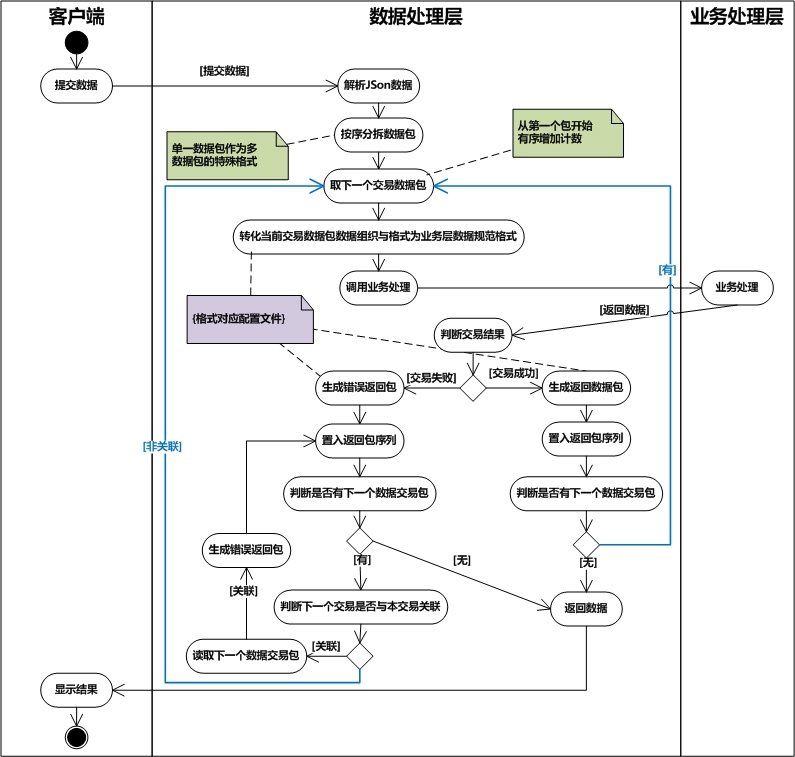
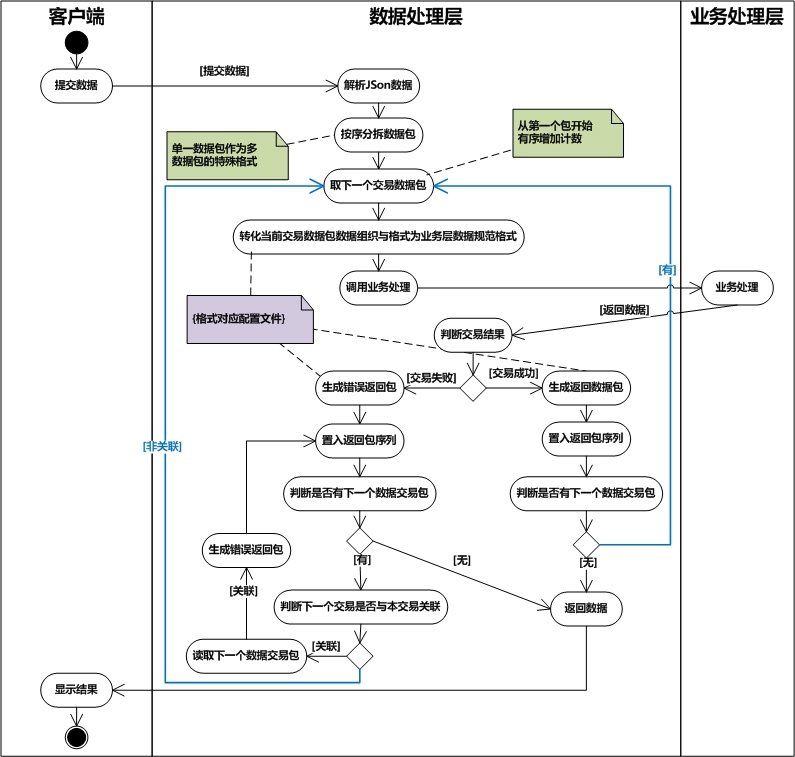
交易接口系统分为数据处理层和业务处理层两部分（如下图所示），分别如下描述：

* **数据处理层**

1. 对外部输入数据进行分拆、解析、转换，作为业务层的数据输入。
2. 对业务层的输出数据进行整合转换，返回外部数据请求端。
3. 数据处理层仅检查通讯应用层JSON协议子层数据传输的正确性，不对JSON协议子层之上的业务数据做任何处理。

* **业务处理层**

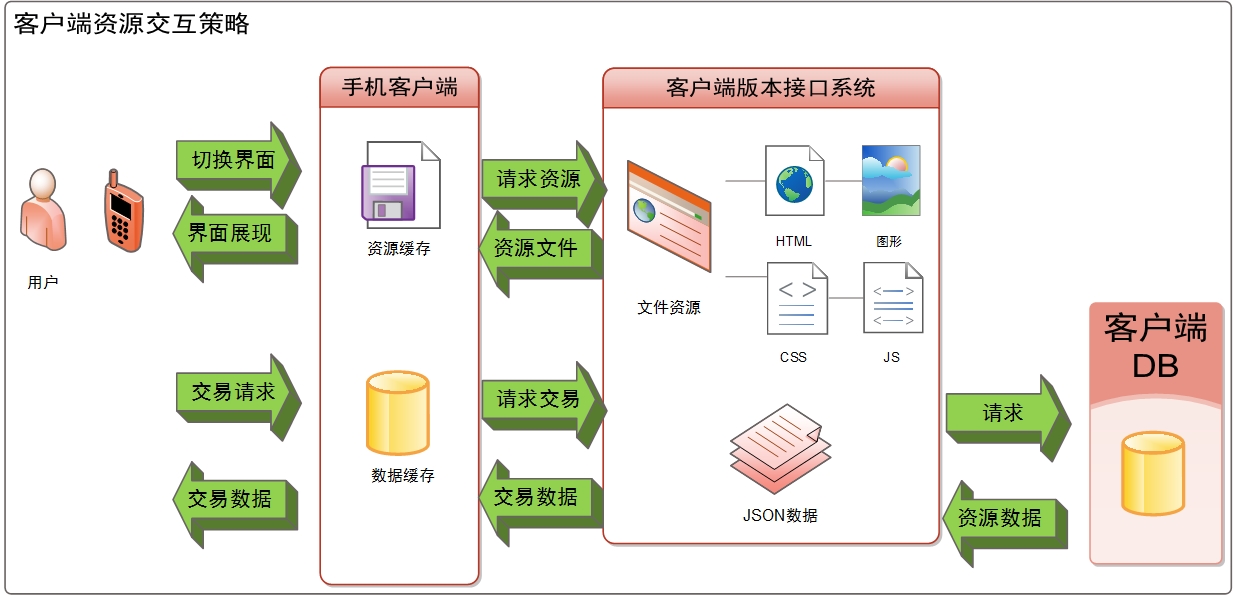
1. 安全认证：负责移动金融相关的安全认证。
2. 业务整合：根据移动金融业务特点，整合自身、后端服务的相关原子交易，向外部提供独立的、完整的业务服务。



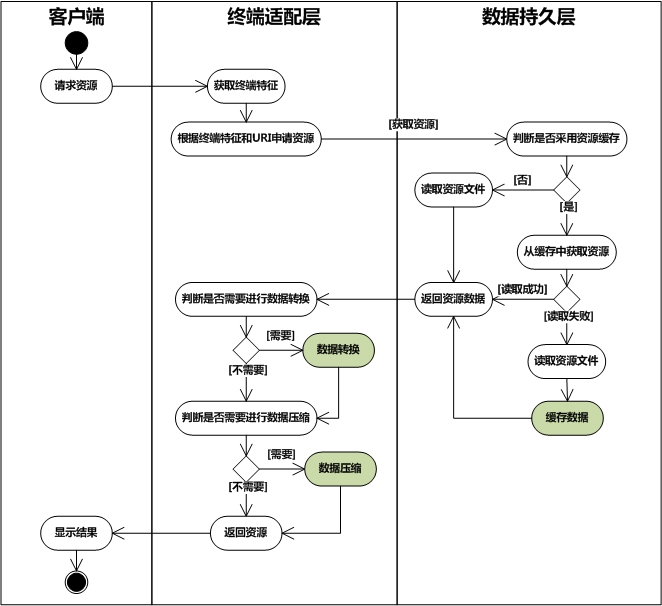
**具体处理流程描述如下：**

1. 终端提交数据到交易接口服务器，数据格式在后续接口规范章节描述。
2. 交易接口服务器数据处理层解析JSON数据。
3. 数据处理层分拆数据包。数据包支持多包格式，单一数据包为多包的特殊格式。
4. 数据处理层取下一个交易数据包。如果是第一次取，则取的是第一个交易数据包。
5. 数据处理层根据格式对应配置，转化本交易数据包为业务处理层不要的数据包。当前版本可直接把数据转化为业务处理层需要接收的数据。
6. 数据处理层调用业务处理层进行业务处理。本质上是流程的转移。
7. 业务处理层进行业务处理，并返回业务处理结果数据给数据处理层。
8. 数据处理层判断交易结果，如果交易失败，进入步骤9；如果交易成功，进入步骤15。
9. 数据处理层根据格式对应配置，生成JSON格式的错误返回包。
10. 数据处理层把错误返回包置入返回包序列。
11. 数据处理层判断是否有下一个数据交易包，如果有，进入步骤12；如果没有，进入步骤18。
12. 数据处理层判断下一个交易是否与本交易关联，如果有关联，进入步骤13；如果无关联，返回步骤4。
13. 数据处理层读取下一个交易数据包。
14. 数据处理层针对下一个交易数据包，直接生成错误返回包，返回步骤10。
15. 数据处理层生成数据返回包。
16. 数据处理层把返回包置入返回包序列。
17. 数据处理层判断是否有下一个数据交易包，如果有，返回步骤4；如果没有，进入步骤18。
18. 数据处理层返回交易完成的数据包（单包或多包）给终端。
19. 终端显示结果，结束。

### 客户端资源交互策略



**处理流程如下图所示：**



如上图所示，资源流程服务器分为终端适配层、数据持久层两部分，具体描述如下：

* **终端适配层**

1. 获取终端特征信息并持续更新维护。
2. 根据终端特征，针对性选择资源返回。
3. 根据需求，对不符合终端格式要求的资源进行转换。包括格式转换、资源的增加/删除/修、数据压缩、数据打包、数据签名等。

* **数据持久层**

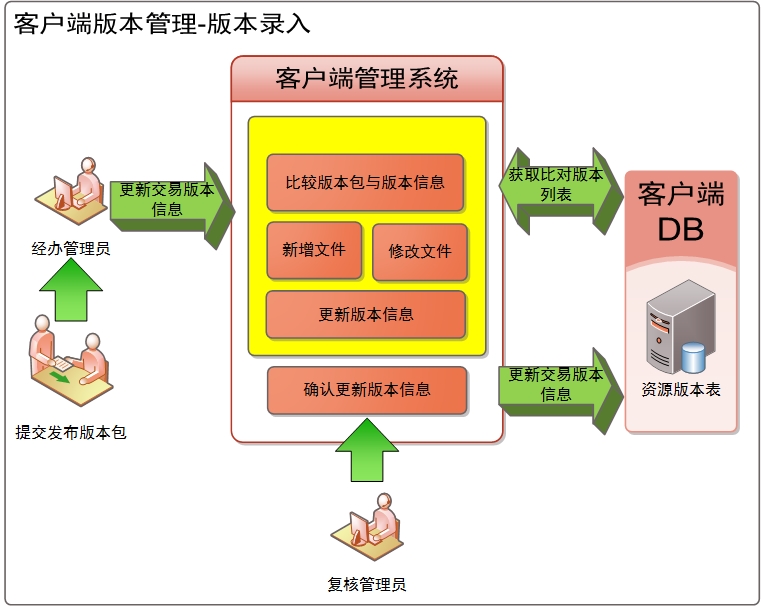
资源的获取和缓存。

**具体处理流程描述如下：**

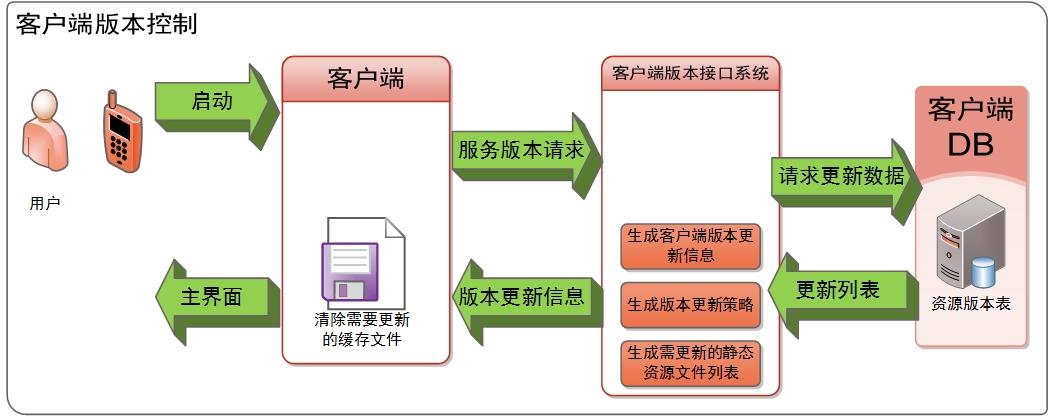
1. 终端提交请求道资源流程服务器。
2. 资源流程服务器终端适配层获取终端特征。
3. 终端适配层根据终端请求和特征到数据持久层获取资源。
4. 数据持久层判断本资源是否需要且已经缓存；如果没有缓存，读取资源并缓存。
5. 数据持久层向终端适配层返回资源数据。
6. 终端适配层判断是否需要进行数据转换和压缩，如果需要，则进行数据转换和压缩。
7. 终端适配层向终端返回数据。
8. 终端显示结果。

### 客户端版本策略

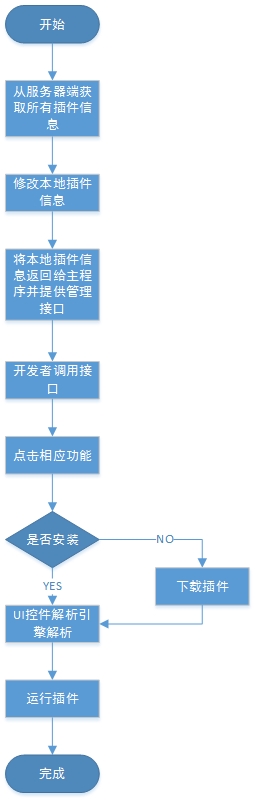
* **版本录入**



* **客户端版本控制**



### 可拓展性插件运行流程



**可拓展插件运行流程描述：**

1. **容器的初始化**

从服务器获取插件描述文件，文件包含插件的图标、显示文字、ID等是否可用。再读取本地插件描述，检查插件是否需要安装、有更新、已作废。更新本地插件信息，将信息封装，提供给主程序。

1. **容器管理接口**

容器提供插件安装、升级、删除接口给主程序，当主程序点击某个已安装插件时，要将ID传给UI控件解析引擎，以便解析。

1. **插件的运行管理**

当UI控件解析引擎和逻辑引擎解析完成后，容器需要提供接口，以便插件运行。

## 2.3移动客户端适配机制

### IOS客户端

iOS根据iPhone 3GS、iPhone 4(S)、iPhone 5(S)、iPhone 6、iPhone 6 Plus五种分辨率来适配。

其中iPhone 3GS和iPhone 4(S)屏幕分辨率分别为480 x 320和960 x 640，成等比率，所以只需要在提供图片时候提供两套分辨率素材分别命名为xxx.png和xxx@2x.png，在读取图片资源时系统会根据设备分辨率自动识别读取的图片。

iPhone 5(S)分辨率为1136 x 640,在使用与屏幕高度无关的素材时自动读取与iPhone 4(S)相同的素材xxx@2x.png，在使用依赖屏幕高度的素材（如全屏背景图片）时，读取为iPhone 5准备的图片，命名为xxx-568h@2x.png。

iPhone 6和iPhone 6 Plus分辨率分别为1334 x 750和2208 x [1242，使用xxx@3x.png这类的图片素材。](mailto:1242，使用xxx@3x.png这类的图片素材。)

### Android客户端

由于市面上的Android的手机型号多样，客户端产品兼容性和适配是一个很重要的工作，以下说明Android兼容性和适配方案，以达到对不同厂商手机的自动兼容。

* **使用丰富多样的布局配置文件**

Android根据分辨率分高分辨率(hdpi)、超高分辨率(xhdpi)、超超高分辨率(xxhdpi)、、超超超高分辨率(xxxhdpi)，对应在工程中设置五种布局文件： values -hdpi、values -xhdpi、values –xxhdpi、values -xxxhdpi及values，其中values为默认资源引用文件，在所有其他values文件中找不到的资源引用才会在values中查找，因此values用于设置分辨率无关布局，其他布局文件资源引用根据不同分辨率进行设置，系统会自动根据设备分辨率匹配最接近的布局资源。

* **文字、布局编码原则**

布局中尽量使用相对布局而非线性布局，表格布局中使用比重的方式，善于使用fill\_parent 和wrap\_content 不建议使用线性布局,大规模相同的属性应定义成style 以便于重复使用，对高度特别敏感的控件在代码中动态更改高度以便于适应不同分辨率。

* **代码实现技巧**

不同分辨率，界面的字体大小，字体等需要不同的样式且需要动态生成的情况下，把不同分辨率的配置信息保存到不同的配置文件夹下面，安卓不同分辨率的手机在运行程序时会自动到res文件夹下面找到最匹配的layout和values文件夹加载页面信息，同时代码中采用引用资源文件的形式，有效减少针对不同分辨率字体大小适配的大量适配工作。

根据常见的手机型号推出多种可选的分辨率的配置，客户在使用时可以自定义手机型号；客户端系统启动时默认优先使用客户自定义的分辨率，若客户没有定义，再根据当前机型信息查找最适合的分辨率来展示应用界面。

### HTML手机页面适配

由于移动手机多样化，尤其是Android系统的手机各式各样，手机的分辨率、屏幕大小、密度等等各个参数各异，因此针对移动开发的HTML页面需要高度的页面适配机制以适应不同的手机，从而得到最好的展现效果。

因此合理的利用viewport就可以做到页面的手机屏幕适配。

**viewport参数：**

|  |  |
| --- | --- |
| **width** | 控制 viewport 的大小，各个浏览器的默认视口宽度不同。范围：200-10,000，也可设成device-width（单位为缩放为 100% 时的 CSS 的像素）。 |
| **height** | 控制 viewport 的高度，范围：223-10,000 也可设成device-height。 |
| **target-densitydpi** | 一个屏幕像素密度是由屏幕分辨率决定的，通常定义为每英寸点的数量（dpi）。Android支持三种屏幕像素密度：低像素密度，中像素密度，高像素密度。一个低像素密度的屏幕每英寸上的像素点更少，而一个高像素密度的屏幕每英寸上的像素点更多。Android Browser和WebView默认屏幕为中像素密度。 |
| **initial-scale** | 初始缩放。即页面初始缩放程度。这是一个浮点值，是页面大小的一个乘数。例如，如果你设置初始缩放为“1.0”，那么，web页面在展现的时候就会以target density分辨率的1:1来展现。如果你设置为“2.0”，那么这个页面就会放大为2倍。 |
| **minimum-scale** | 页面显示时最小的缩放值，默认值：0.25，范围：>0-10.0 |
| **maximum-scale** | 页面显示时最大的缩放值，默认值：5.0，范围：>0-10.0 |
| **user-scalable** | 用户调整缩放。即决定用户是否可以放大或者缩小页面。如果设置为yes则是允许用户对其进行改变，反之为no。默认值是yes。如果你将其设置为no，那么minimum-scale 和 maximum-scale都将被忽略，因为根本不可能缩放。 |

为了防止Android Browser和WebView 根据不同屏幕的像素密度对你的页面进行缩放，你可以将viewport的target-densitydpi 设置为 device-dpi。当你这么做了，页面将不会缩放。相反，页面会根据当前屏幕的像素密度进行展示。在这种情形下，你还需要将viewport的width定义为与设备的width匹配，这样你的页面就可以和屏幕相适应。

**范例：**

1. **设置屏幕宽度为设备宽度，禁止用户手动调整缩放：**

|  |
| --- |
| <metaname="viewport" content="width=device-width,user-scalable=no" /> |

1. **设置屏幕密度为高频，中频，低频自动缩放，禁止用户手动调整缩放：**

|  |
| --- |
| <meta name="viewport" content="width=device-width,target-densitydpi=high-dpi,initial-scale=1.0, minimum-scale=1.0, maximum-scale=1.0, user-scalable=no" /> |

## 2.4安全策略

* **加密技术**

加密技术是电子商务采取的基本安全措施，双方应用在传递一些重要的数据时必须对数据进行加密。

* **对称加密**

发送方和接收方用同一个密钥去加密和解密数据。它的最大优势是加/解密速度快。可以选择常用的DES、AES等对称加密算法。

* **非对称加密**

非对称加密又称公钥加密，使用一对密钥来分别完成加密和解密操作，其中一个公开发布（即公钥），另一个由用户自己秘密保存（即私钥）。信息交换的过程是：甲方生成一对密钥并将其中的一把作为公钥向其他交易方公开，得到该公钥的乙方使用该密钥对信息进行加密后再发送给甲方，甲方再用自己保存的私钥对加密信息进行解密。

在移动金融开放平台中，数据在传输过程中为了防篡改，发送方必须对数据用私钥进行签名，接收方然后用公钥验签。为了保证数据的私密性发送方用公私加密，接收方然后用私钥解密。常用的非对称加密算法有RSA。

* **认证技术**

认证技术是用电子手段证明发送者和接收者身份及其文件完整性的技术，即确认双方的身份信息在传送或存储过程中未被篡改过。

采用RSA非对称密钥认证方式，在建立安全通道时，手机和PC浏览器需要获得服务器的公钥和证书，也就是成对服务器端的认证；然后需要用RSA公共密钥生成认证信息，并把认证信息发送给后台应用，后台应用使用RSA私有密钥对认证信息进行验证。若验证成功，认为服务器是可靠的、真实的。之后，手机、PC浏览器各自根据认证信息和服务器信息，按一定的算法，产生动态密钥，建立起安全传输通道。

手机客户端模式还加入了客户签名，进一步加强了多因素认证，确保了参与交易的各个环节的合法和真实性。

* **数字签名**

数字签名也称电子签名，如同出示手写签名一样，能起到电子文件认证、核准和生效的作用。数字签名机制提供了一种鉴别方法，以解决伪造、抵赖、冒充、篡改等问题。

实现过程：发送方对信息先进行摘要计算，然后采用非对称算法的私钥对摘要怎行签名，接收方再用公钥验签。

* **数字证书**

数字证书是是由权威机构－－CA证书授权中心发行的，能提供在Internet上进行身份验证的一种权威性电子文档，人们可以在互联网交往中用它来证明自己的身份和识别对方的身份。

第三方应用接入移动金融开放平台，双方交换数字证书，数字证书可以实现双方的身份认证，数据传递的机密性，完整性。发送方对数据不能抵赖。

* **网络通信安全**

通讯采用HTTPS协议，SSL握手时验证服务器证书合法性，加密算法要求1024位以上。SSL协议保证传输通道的安全，包括通讯数据加密、服务器身份认证。

前端与后台服务器、后台服务器与其他系统之间建立SSL加密通道。通过成熟和标准的安全协议、安全算法与安全密钥的多重保障，来实现数据的完整性、保密性和可信路径的支持。

TLS1.0（SSL3.0）协议采用以非对称密钥算法为基础，采用动态的密钥来保护传输数据。传输密钥是在RSA算法的保护下，由服务器和客户端协商生成，利用传输密钥加密所要传递的数据。同时由于采用的是动态密钥，动态密钥的生存周期仅为一次会话过程。在TLS1.0协议中有完整的关于会话管理、数据完整性、防篡改的控制机制。

* **数据保密**

系统采用非对称算法（RSA）进行安全通道的建立，用对称算法（动态密钥）对安全通道的传输数据进行保护。采用RSA算法，确保密钥的可靠性。动态密钥是由前端产生的随机数和后台系统产生的随机数按一定算法得到的，即动态密钥的值是随机的、没有规律的。同时，动态密钥只在会话期间有效，且设定一定的存活时间，在会话结束或超过存活时间，动态密钥作废，从而提高系统的安全性。

在进行交易数据传输前，必须在手机与后台系统之间、PC浏览器与后台系统之间建立端对端的安全通道。对于所有的交易数据，都会在动态密钥的保护下通过安全通道进行传输。对交易数据的加密采用3DES算法或RC4加密，在确保数据的安全前提下，减少了加密所需要的时间（RC4是流加密方式，RC4加密的速度比DES快10倍）。

* **产品的完整性**

对于所有交易数据，前端和系统后台都会对交易数据进行摘要（MD5）处理，产生交易数据的校验信息，以防止数据在传输中途被修改或丢失。若接收到数据的摘要验证不通过，即认为数据被破坏，要求交易重新进行。

所有的客户端程序都是在签名后供客户下载，保证客户端程序不会被篡改，从而保证客户端数据不会被伪造。

* **系统抗抵赖性**

每次交易，用户都先要输入用户登录密码，先进行用户密码的验证，若密码错误，交易终止。故对于所有的成功交易，交易数据会包含用户输入的正确密码，是用户对交易的认可。

系统通过日志记录的方法，详细记录了电子交易数据，这也是符合《电子签名法》的抗抵赖性的做法。

* **移动客户端程序安全**

iOS客户端程序通过电子市场本身验证机制可有效的保证程序包得安全。

Android客户端利用progurd对代码进行混编，还可以结合对客户端安装包进行加壳处理，可以非常有效阻止程序被反编译。

使用签名验证机制，保证客户端应用包安全，在签名变动或代码恶意篡改的情况下，客户端无法运行。

* **重复提交控制机制**

重复提交有两种情况，一种是客户不小心连续点击了两次或多次的情况，另一种是通过后台程序连续进行交易的情况。对于前一种情况，我们使用TOKEN的方式进行控制，对于多次点击请求，我们只处理一次。对于后一种情况，我们使用时间进行控制，两次交易必须间隔一定的时间才能进行。

另外，移动客户端产品可以针对提交按钮进行点击次数控制，当用户点击一次之后，数据验证通过正常提交后台时，则禁止再次点击；如果验证失败，则允许用户可以再次点击提交按钮。

* **客户端检查客户用户环境**

客户端可检查客户用户环境，可检测ROOT或越狱状态，并进行可配置的提醒。

* **交易权限控制**

系统可对用户的权限进行灵活控制和管理。不同安全级别的用户分配不同的权限，权限低的用户只能看到或操作一些基本的功能交易，权限高的用户能够看到或操作动账类的交易。

* **移动硬件设备信息绑定**

手机客户端借助系统后台，可以将硬件设备号和客户号进行绑定，即实行设备号和客户号的多对一关系，同时系统可设置开关，实行设备号和客户号的一对一严格控制。客户在访问手机、平板客户端时，系统将验证客户号与设备号是否对应。在某些需要短信验证的时候，也会将验证短信发送到绑定的手机号上。

* **敏感数据处理**

针对产品中出现的敏感数据，例如：密码、银行账号、身份证等敏感数据打印成\*号，避免用户个人信息的泄露。

在客户退出产品时，清除客户端缓存数据。

* **会话控制**

系统对交易会话实行以下控制：

1. 交易开始时，清除再创建交易区中的数据。
2. 交易结束时，清除交易区的数据。
3. 登录的时候先清除存储的所有数据。
4. 页面超时控制在15分钟之内（可自行配制超时时间）。
5. 会话超时退出给出提示。

# 三、性能保障

**负载均衡 - 高并发性，高效率**

1. 将大用户量的并发访问分担到多台节点机上并行处理，缩短用户等待响应的时间，系统的处理能力得到提高，并且能够接受更多用户的并发访问。
2. 将单个用户重负载的运算分担到多个节点机上做并行处理，每个节点机有多个CPU，也做并行处理，处理结束后，将结果汇总，返回给用户。系统处理能力得到大幅度提高，响应时间大幅缩短，提高系统性能。
3. 软件架构方面，考虑并发和性能。运用Value Object，空对象模式和MVC 模式，结合适当的缓存机制，优化数据库调用来提高系统性能。

**响应时间定义**

录入交易响应时间：<2秒

查询交易响应时间：<3秒

报表交易平均响应时间：<30秒

交易超时时间：45秒

**高可用性 – 可维护性**

高可用性意味着最大化的服务器开机时间，也就是最小化的服务器计划外的停机时间，从而可以很好地保护关键性数据、提高生产效率。可运用true cluster系统（数据库 服务器采用RAC技术），当一台节点机出现故障时，其他节点机仍旧可以正常工作，客户机的访问不会中断，而出现故障的服务器可离线修理，修复后，可连入集群中继续工作，可以保证当n台节点机中有n-1个同时出现故障时，系统仍能正常运行。当然，这种情况下，如果并发访问用户数多或运算量大时，剩余服务器的工作负载将是非常重的。

**可扩展性**

先使用一个工作单元进行数据处理，随着业务的发展，当发现处理能力不足时，可再级联第二个工作单元，即可获得处理性能的提高。同时也可以通过硬件扩容比如增加内存，使用增加NAS存储等方式来进行扩展。

软件架构方面，可采用工厂设计模式等来提高接口的扩展性以适应不同的，新增的业务，可采用对象创建型模式来扩展不断变化的业务流程

# 四、接口设计方案

## 3.1请求

### 请求头

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| MBC\_User\_Agent | 是 | String | 客户端信息，格式为：应用名/平台号/固件版本/当前客户端版本/页面版本/设备号。  其中手机银行应用名：MBCCCB； 网点地图应用名：MBCMAP；悦生活应用名：EBSLIFE |
| UserToken | 是 | String | Token信息 |
| MBC-User-Info | 是 | String | 用户信息，格式为：UserToken/\*/IMEI(国际移动设备身份码)/\*/手机操作系统/\*/内核版本号 OS Version/\*/设备型号 Device Model/\*/手机分辨率 screenSize/\*/WIFI MAC/\*/获取邻近的小区号/\*/获取基站cid/\*/获取经纬度 |

### 请求体

POST请求参数：

JSON=”json参数字符串”

SKEY=” CVel6E”

MBSKEY=” DgQrwVcCPy7YuUMj68H0iBpEiQOSiq5zmuGZj0JweDJokGpjmPL4j”

USERID="TESTA0000000000001"

BRANCHID="010231000"

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| JSON | 是 | String | json报文头+json报文体 |
| SEKY | 是 | String | sessionID |
| MBSKEY | 是 | String | MBSKEY |
| USERID | 是 | String | 用户id |
| BRANCHID | 是 | String | 分行号 |

#### Json报文头格式

|  |
| --- |
| "header":{  “agent”:”mbp1.0”,  “version”:”4.00”,  “device”:” HTC,m10,6.0.1”,  “platform”:” HTC,m10,6.0.1”,  “local”:”zh\_cn”,  “ext”:””  }, |

说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| header | 是 | String | 报文头标识 |
| agent | 是 | String | 终端应用标识，基于浏览器的也是一种终端应用 |
| version | 是 | String | 终端应用版本，建议格式：“x.y”，其中x为大版本，y为小版本 |
| device | 否 | String | 设备信息，用来区别终端特性，建议格式：“厂商名,产品名,型号” |
| platform | 否 | String | 操作系统信息，建议格式：“厂商名,产品名,版本号” |
| local | 是 | String | 国家及语言，标准格式 |
| ext | 否 | String | 扩展字段 |

#### Json报文体格式

请求包的值域由多个对象数据组成。request下每一个{}代表一个交易请求。

|  |
| --- |
| "request":[  {  “txCode”:” NP0001”,  “id”:”id0”,  “version”:”1.0.0”,  “preassociated”:””,  “params”:{  “key1”:”value1”,  “key2”:”value2”,  “key3”:”value3”  }  },  {  “txCode”:” NP0002”,  “id”:”id1”,  “version”:”1.0.0”,  “preassociated”:”id0”,  “params”:{  “key1”:”value1”,  “key2”:”value2”,  “key3”:”value3”  }  }  ] |

说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| request | 是 | String | 请求包标识 |
| txCode | 是 | String | 交易码，如：NP0001 |
| id | 是 | String | 子交易唯一标示。可以通过关联标志关联起来。 |
| version | 否 | String | 当前调用接口的版本。如果不填写，则缺省为最新版本 |
| preassociated | 否 | String | 关联标志。用于多交易组包，表示该交易依赖于哪个交易。缺省为空。 |
| params | 是 | String | 请求参数 |

## 3.2 响应

### 响应头

响应码:

|  |  |
| --- | --- |
| status | 说明 |
| 200 | 成功 |
| 701 | 当客户端用户未上传凭证，且正在进行需要凭证交易时，返回该错误码给客户端 |
| 702 | 当客户端用户上传凭证且凭证无效后，手机银行应用返回该错误码给客户端 |

返回字段:

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| HEAD\_SKEY | 是 | String | sessionID |
| HEAD\_MBSKEY | 是 | String | MBSKEY |
| HEAD\_USERID | 是 | String | 用户id |
| HEAD\_BRANCHID | 是 | String | 分行号 |

### 响应体

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| JSON | 是 | String | json报文头+json报文体 |

#### 响应Json报文头格式

|  |
| --- |
| "header":{  “server”:”weblogic10.3”,  “status”:"01",  “local”:”zh\_cn”,  “ext”:””  }, |

说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| header | 是 | String | 报文头标识 |
| server | 否 | String | 服务器版本 |
| status | 是 | String | 返回结果状态。  00：全部失败。  01：全部成功。  02：部分成功，部分失败。 |
| local | 是 | String | 国家及语言，标准格式 |
| ext | 否 | String | 扩展字段 |

#### 响应Json报文体格式

返回包的值域由多个对象数据组成。每一个对象数据为一个交易返回包，与请求包的顺序一一对应。

|  |
| --- |
| "response":[  {  “txCode”:” NP0001”,  “id”:”0”,  “version”:”1.0.0”,  “status”:”01”,  “result”:{  “key1”:”value1”,  “key2”:”value2”,  “key3”:”value3”  }  },  {  “txCode”:” NP0002”,  “id”:”1”,  “version”:”1.0.0”,  “status”:”01”,  “result”:{  “key1”:”value1”,  “key2”:”value2”,  “key3”:”value3”  }  }  ] |

说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| response | 是 | String | 返回包标识 |
| txCode | 是 | String | 交易码，如：NP0001 |
| id | 是 | String | 子交易唯一标示。可以通过关联标志关联起来 |
| version | 否 | String | 当前返回接口的版本。如果不填写，则缺省为请求提交的版本。 |
| status | 是 | String | 返回结果状态。01为成功，其它为失败 |
| result | 否 | String | 返回参数。 |
| marketing | 否 | String | 返回包标识，用来标识这是一个“营销类”的交易控制。  父节点路径：  / |
| marketCode | 否 | String | 用来标识营销类型。  shake：天天摇  父节点路径：  /marketing/ |
| marketData | 否 | String | 营销参数，具体内容结构根据不同的营销类型与客户端自行约定。  父节点路径：  /marketing/ |

#### 错误处理格式

##### 安全控制

用于安全控制，例如音频盾返回给客户端的Json格式为：

|  |
| --- |
| "safe":{  　 　"txCode":"SJ0005, SJ0004, SJ0003",  　　 "safeCode":"AUDIO\_SIGN",  　 　"safeData":{ }  　　 } |

说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| safe | 是 | String | 返回包标识，用来标识这是一个“安全类”的交易控制 |
| txCode | 是 | String | 具体交易的交易码，告诉客户端把处理结果追加到哪个交易，可以包含多个交易码，用“，”分开 |
| safeCode | 是 | String | 用来标识安全类型，比如AUDIO\_SIGN代表音频盾 |
| safeData | 是 | String | 安全控制传递的参数，具体内容结构根据不同的安全类型与客户端自行约定 |

##### 非交易错误

非交易错误是指MBS后端接收客户端请求，并且按照请求的要求处理相应的应用逻辑，当在处理过程中出现关于整个交易的异常时返回全局的错误，比如参数错误，客户端冲号错误等。

其中冲号错误的错误码为：0760I3000002，需客户端跳转到首页。

|  |
| --- |
| “error”: {  “errCode”:” 0760I3000002”,  “errMessage”:” 非法登录或有其他用户使用您的客户号登录本网站”  } |

说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| error | 是 | String | 返回包标识 |
| errCode | 是 | String | 错误码。 |
| errMessage | 是 | String | 错误提示信息。 |

##### 交易错误

交易错误是指处理交易过程中的错误。当返回交易包的status不等于01时，表明交易错误，返回错误信息。如果某子交易B所依赖的子交易A失败，则该子交易B不必要执行，那么该子交易B的错误信息与所依赖的子交易A错误信息一致。

|  |
| --- |
| "response":[  {  “txCode”:” NP0001”,  “id”:”0”,  “version”:”1.0.0”,  “status”:”00”,  “result”:{  “errCode”:” 0760E3000008”,  “errMessage”:”密码错误，请重试”  }  }  {  “txCode”:” NP0002”,  “id”:”0”,  “version”:”1.0.0”,  “status”:”00”,  “result”:{  “errCode”:” 0760E3000008”,  “errMessage”:”密码错误，请重试”  }  }  ] |

说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段名称 | 是否必填 | 数据类型 | 备注 |
| response | 是 | String | 返回包标识 |
| txCode | 是 | String | 交易码 |
| id | 是 | String | 子交易唯一标示。可以通过关联标志关联起来。 |
| version | 否 | String | 当前返回接口的版本。如果不填写，则缺省为请求提交的版本。 |
| status | 是 | String | 返回结果状态。01为成功，其它为失败 |
| result | 否 | String | 返回参数。 |
| errCode | 是 | String | 错误码。status不为01时必有此项 |
| errMessage | 是 | String | 错误提示信息。status不为01时必有此项 |

## 3.3配置响应参数信息

在/mbs/src/pages/NEWUI/FMT/CN/STY2/ 中新建xml配置文件格式如下：

<?xml version=*"1.0"* encoding=*"UTF-8"*?>

<page>

<f key=*"name"* type=*"T"* value=*"Json"* />

<f key=*"acc"* type=*"V"* tag=*"335"* />

<c key=*"accList"* tag=*"335"*>

<f key=*"accNo"* type=*"V"* tag=*"254"* />

</c>

</page>

field代表返回的参数, 属性如下

|  |  |  |
| --- | --- | --- |
| 属性 | 是否必填 | 说明 |
| key | 是 | 客户端取得参数的属性名，不可重复 |
| tag | 否 | 设置在IBSEnv/InArea/OutArea的属性名 |
| type | 是 | E: 设置在IBSEnv的数据  V: 设置在InArea/OutArea的数据，先从OutArea获取数据  T: 字符串, 属性值保存在value中 |
| value | 否 | type为T的属性值 |

field代表返回的参数列表, 属性如下

|  |  |  |
| --- | --- | --- |
| 属性 | 是否必填 | 说明 |
| key | 否 | 参数列表名，不可重复 |
| tag | 是 | E: 设置在IBSEnv的数据  V: 设置在InArea/OutArea的数据，先从OutArea获取数据  T: 字符串, 需要有value |
| c | 是 | 列表的某个子元素, 用于计算循环次数 |

# 测试方案

### 测试团队构成

#### 职责

软件测试是软件开发过程中的重要组成部分，主要负责：

1、测试需求分析；

2、编写、优化测试案例，执行测试，提交各阶段测试报告；

3、跟踪缺陷生命周期；

4、从用户体验和测试的角度提出自己的看法；

5、编写各测试阶段的测试文档。

#### 角色划分

在人力资源有限的情况下，一个团队成员可能会同时承担多个角色。

|  |  |
| --- | --- |
| 角色名称 | 负责内容 |
| 项目经理 | 1. 测试任务的接受、大方面的决策、测试人员的调配等； 2. 主动每天跟各个专项跟进人，沟通任务信息； 3. 每周一次收集任务跟踪表，了解详细信息； 4. 对测试风险、测试问题进行决策判断； 5. 对客户、公司进行沟通和交代。 |
| 专项跟进人 | 1. 编写测试计划、评审测试案例； 2. 掌握专项测试任务的情况，包括：任务进度、风险、问题、人员状态、业务方反馈的信息等，推动各测试问题解决； 3. 为测试工作提供技术上支持； 4. 向项目经理汇报专项任务的信息。 |
| 测试工程师 | 1. 编写、优化测试案例、评审测试案例； 2. 执行测试、跟踪缺陷生命周期、提交各阶段测试报告；   3、 及时与业务沟通各测试问题；  4、从用户体验和测试的角度提出自己的看法。 |

### 测试流程及规范

#### 测试流程示意图：

**测试总结**

**需求**

**分析**

**编写测试计划**

**设计测试案例**

**&**

**测试案例评审**

**跟踪缺陷生命周期**

**SIT**

**测试&回归测试**

**测试执行**

**测试设计**

**测试策划**

**UAT**

**测试**

**&**

**回归测试**

**编写测试报告**

#### 测试策略

系统测试类型及各种测试类型所采用的方法、工具等介绍如下：

##### 功能测试

|  |  |  |
| --- | --- | --- |
| 测试范围 | | 验证数据精确度、数据类型、业务功能等相关方面的正确性 |
| 测试目标 | | 核实所有功能均已正常实现，即是否与需求一致 |
| 技 术 | | 采用黑盒测试、边界测试、等价类划分等测试方法 |
| 工具与方法 | | 手工测试 |
| 开始标准 | 开发阶段对应的功能完成并且测试用例设计完成 | |
| 完成标准 | | 测试用例通过并且最高级缺陷全部解决 |
| 需考虑的特殊事项 | |  |

###### 用户界面（UI）测试

|  |  |  |
| --- | --- | --- |
| 测试范围 | | 1．导航、链接、Cookie、页面结构包括菜单、背景、颜色、字体、按钮名称、TITLE、提示信息的一致性等。  2．友好性、可操作性（易用性） |
| 测试目标 | | 核实各个窗口风格（包括颜色、字体、提示信息、图标、TITLE等等）都与需求保持一致，或符合可接受标准，能够保证用户界面的友好性、易操作性，而且符合用户操作习惯。 |
| 技 术 | | WEB测试通用方法 |
| 工具与方法 | | 手工测试、目测 |
| 开始标准 | 界面开发完成 | |
| 完成标准 | | UI符合可接受标准，能够保证用户界面的友好性、易操作性，而且符合用户操作习惯 |
| 测试重点与优先级 |  | |
| 需考虑的特殊事项 | |  |

##### 非功能测试

|  |  |  |
| --- | --- | --- |
| 测试范围 | | 多用户长时间在线操作时性能方面的测试 |
| 测试目标 | | 核实系统在大流量的数据与多用户操作时软件性能的稳定性，不造成系统崩溃或相关的异常现象 |
| 技 术 | | 手工测试、自动化测试 |
| 开始标准 | 自动化测试脚本设计并评审通过且项目组移交系统测试 | |
| 完成标准 | | 系统满足用户需求中所要求的性能要求 |
| 测试重点与优先级 |  | |
| 需考虑的特殊事项 | |  |

###### 安全性测试

|  |  |  |
| --- | --- | --- |
| 测试范围 | | 1. 用户、管理员的密码安全 2. 权限 3. 非法攻击 |
| 测试目标 | | 1. 用户、管理员的密码管理  2．应用程序级别的安全性：核实用户只能操作其所拥有权限能操作的功能。  3．系统级别的安全性：核实只有具备系统访问权限的用户才能访问系统。 |
| 技 术 | | 代码包或者非法攻击工具 |
| 工具与方法 | | 手工测试 |
| 开始标准 | 功能测试完成 | |
| 完成标准 | | 执行各种非法操作无安全漏洞且系统使用正常 |
| 测试重点与优先级 |  | |
| 需考虑的特殊事项 | |  |

###### 兼容性测试

|  |  |  |
| --- | --- | --- |
| 测试范围 | | 1．使用不同版本的不同浏览器、分辨率、操作系统分别进行测试。  2．不同操作系统、浏览器、分辨率和各种运行软件等各种条件的组合测试。 |
| 测试目标 | | 核实系统在不同的软件和硬件配置中运行稳定 |
| 技 术 | | 黑盒测试 |
| 工具与方法 | | 手工测试 |
| 开始标准 | 项目组移交系统测试 | |
| 完成标准 | | 在各种不同版本不同类项浏览器、操作系统或者其组合下均能正常实现其功能（此测试根据开发提供依据决定测试范围） |
| 测试重点与优先级 |  | |
| 需考虑的特殊事项 | |  |

##### 回归测试

|  |  |  |
| --- | --- | --- |
| 测试范围 | | 所有功能、用户界面、兼容性、安全性等测试类型 |
| 测试目标 | | 核实执行所有测试类型后功能、性能等均达到用户需求所要求的标准 |
| 技 术 | | 黑盒测试 |
| 工具与方法 | | 手工测试和自动化测试 |
| 开始标准 | 每当被测试的软件或其环境改变时在每个合适的测试阶段上进行回归测试 | |
| 完成标准 | | 95%的测试用例执行通过并通过系统测试 |
| 测试重点与优先级 | 测试优先级以测试需求的优先级为参照 | |
| 需考虑的特殊事项 | | 软硬件设备问题 |

##### 测试实施阶段

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 测试类型 | 测试阶段 | | | |
| 单元测试 | 集成测试 | 系统测试 | 验收测试 |
| 功能测试 | X | ✓ | ✓ | X |
| 性能测试 | X | ✓ | ✓ | X |
| 安全性测试 | X | ✓ | ✓ | X |
| 兼容性测试 | X | ✓ | ✓ | X |
| 用户界面（UI）测试 |  | X | ✓ | X |
| 回归测试 | 每当被测试的软件或其环境改变时在每个合适的测试阶段上进行回归测试 | | | |
| 备注：“✓”表示由测试组执行，“X”表示由项目组执行； | | | | |

##### 测试通过标准

系统无业务逻辑错误和二级的BUG。经确定的所有缺陷都已得到了商定的解决结果。所设计的测试用例已全部重新执行，已知的所有缺陷都已按照商定的方式进行了处理，而且没有发现新的缺陷。

注：缺陷的严重等级说明：

A：严重影响系统运行的错误；

B：功能方面一般缺陷，影响系统运行；

C：不影响运行但必须修改；

D：合理化建议。

##### 测试需求及测试用例追溯表

**参照测试需求列表及测试用例列表**

##### 测试用例模板

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 用例标识 | | 01 | | | 项目名称 | \*\*\*系统 | | | | | |
| 开发人员 | |  | | | 模块名称 | \*\*\* | | | | | |
| 用例作者 | |  | | | 参考信息 |  | | | | | |
| 测试类型 | | 功能测试 | | | 设计日期 |  | | 测试人员 | |  | |
| 测试方法 | | 手工、黑盒 | | | 测试日期 |  | |  | |  | |
| 用例描述 | |  | | | | | | | | | |
| 前置条件 | |  | | | | | | | | | |
| 编号 | 测试项 | | 操作步骤 | 预期结果 | | | 数据 | | 实际结果 | | 结果比较说明 |
| 1 |  | |  |  | | |  | |  | |  |
| 2 |  | |  |  | | |  | |  | |  |
| 3 |  | |  |  | | |  | |  | |  |

##### 测试进度

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **测试用例ID** | **开始日期** | **完成日期** | **测试人** | **备注** |
| 01 |  |  |  |  |
| 02 |  |  |  |  |
| 03 |  |  |  |  |
| 04 |  |  |  |  |
| … |  |  |  |  |
| … |  |  |  |  |
| … |  |  |  |  |
| … |  |  |  |  |

# 六、上线方案

系统的整体应用架构



1. 运行环境（软、硬件）说明；

1、硬件设备：

路由器

防火墙

F5（SSL证书加速）

AP服务器

WEB服务器集群

数据库服务器（公共库、流水库）

后台服务器

ECTIP反向交易服务器

2、系统软件：

RedHat Linux6.4

WebLogic Serve

Apache web

Haproxy

Oracle 11gR2

1. 系统的工作过程中不得不包含的人工处理过程（如果有的话），如：在日常作业中未自动化处理、需人工干预或特殊处理的内容等；

手机银行项目组数据库部署由于涉及到迁移数据和新交易数据，需要人工配合生成数据迁移工作。

1. 尚未解决而需要采取临时处理措施的内容。

系统监控需要进一步加入到概要设计的范围，系统监控是系统有效稳定运行的前提条件，也是保证生产的必备手段，概要设计中需要加强对系统监控模块的设计。

### 初始化

#### 数据库

##### 执行备份脚本

执行上线数据库脚本前，请执行“MBS\_BACKUP.sh”脚本备份本次上线数据库变化的表结构和数据。

具体操作步骤：

1. 以oracle身份的用户登录手机银行数据库终端；
2. 修改文件执行权限：chmod 775 MBS\_BACKUP.sh；
3. 执行备份脚本“MBS\_BACKUP.sh”，按提示输入B2C20用户的“密码”；
4. 执行完成后，检查日志文件“bjdata20121215.log”是否备份成功；

##### 执行上线脚本

上线脚本分2批执行，不停业执行和停业执行，具体操作步骤如下：

1. **不停业执行脚本：**
2. 以B2C20用户登录SQLPLUS ；
3. 按顺序执行前3个SQL脚本（执行顺序为MBS\_1.sql、MBS\_2.sql、MBS\_SPOOL\_DATA.sql）；
4. 执行完“MBS\_SPOOL\_DATA.sql”后，确认“/smsfs/b2c20\_check/exp\_temp\_dir/20121215/data”目录下是否有3个表数据文件。并将这3个数据文件传到入口网站wapdb数据库服务器“/home/db/oracle/orarch\_wapdb/dump\_dir/20121215/data”目录下。
5. 执行完每个SQL脚本后，检查相应日志文件的内容，看是否有异常；
6. 如果执行成功则手工提交事务COMMIT；否则执行ROLLBACK；
7. 执行MBS\_VERIFY.sql验证脚本，并检查日志记录文件MBS\_VERIFY.log的内容，看是否与脚本中的说明一致；
8. **部署完所有应用服务后，执行MBS\_3.sql，升级客户端页面版本；**
9. **停业执行脚本：**
10. 以B2C20用户登录SQLPLUS ；
11. 按顺序执行后3个SQL脚本（执行顺序为MBS\_4.sql、MBS\_5.sql、MBS\_6.sql）**注意：MBS\_5请手工逐条执行**；
12. 执行完每个SQL脚本后，检查相应日志文件的内容，看是否有异常；
13. 如果执行成功则手工提交事务COMMIT；否则执行ROLLBACK；

执行MBS\_VERIFY\_2.sql验证脚本，并检查日志记录文件MBS\_VERIFY\_2.log的内容，看是否与脚本中的说明一致；

#### 服务端

##### 系统程序备份

备份路径如下程序：

（1）应用WAR包：

/home/mw/weblogic10/bea/user\_projects/domains/mbs/mbs.war

（2）定时WAR包：

/home/mw/weblogic10/bea/user\_projects/domains/mbsTimejob /mbsTimejob.war

##### 应用服务mbs的更新

###### 上传应用程序包

将mbs-86.war上线包上传到200.1.5.86机器的/home/mw/weblogic10/bea/user\_projects/domains/mbs目录下，然后删除mbs.war，再把mbs-86.war重命名为mbs.war。(**200.1.5.86服务器**)

将mbs-93.war上线包上传到200.1.5.93机器的/home/mw/weblogic10/bea/user\_projects/domains/mbs目录下，然后删除mbs.war,再把mbs-93.war重命名为mbs.war。(**200.1.5.93服务器**)

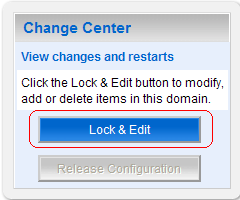
###### 发布程序包

此操作针对WEBLOGIC10.3版本。

登陆到weblogic控制台

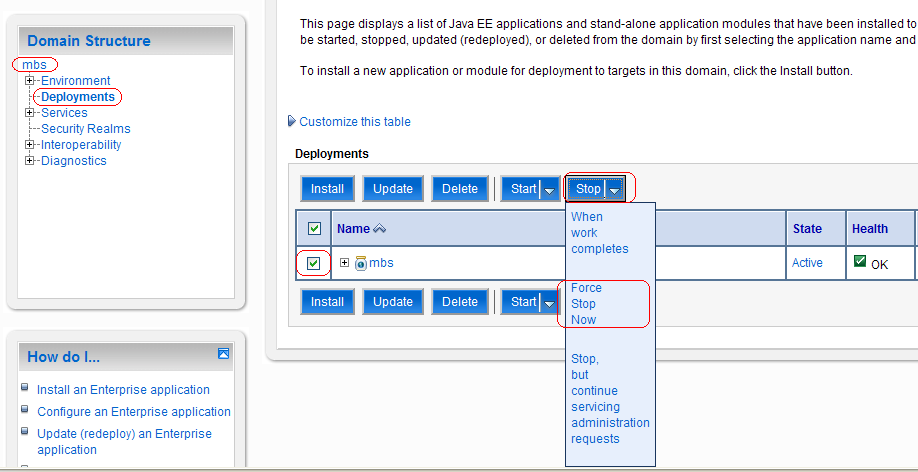
地址：[http://200.1.5.86:7080/86mbsnew](http://200.1.5.86:7080/console)

然后点击下面的“Lock & Edit”按钮：

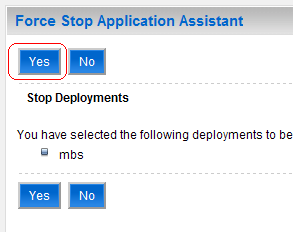


删除mbs应用

依次选择mbs >>[Deployments](http://128.128.97.45:7080/console/console.portal?_nfpb=true&_pageLabel=AppDeploymentsControlPage)>>选项：



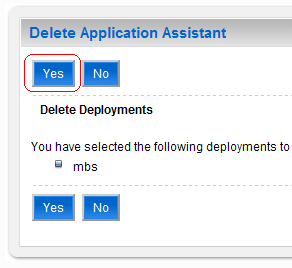
在右边出现的窗口中选择mbs应用左边的复选按钮，然后点击“stop”按钮，再选择下面的“Force Stop Now”按钮停止服务；



在出现的上面界面中点击“Yes”按钮



在出现的上面界面中选中mbs应用服务左边的复选框，然后点击上面的“Delete”按钮；

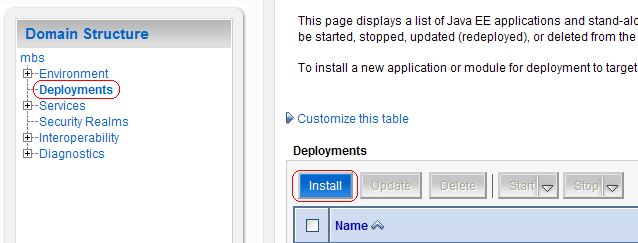


点击“Yes”按钮删除mbs应用。

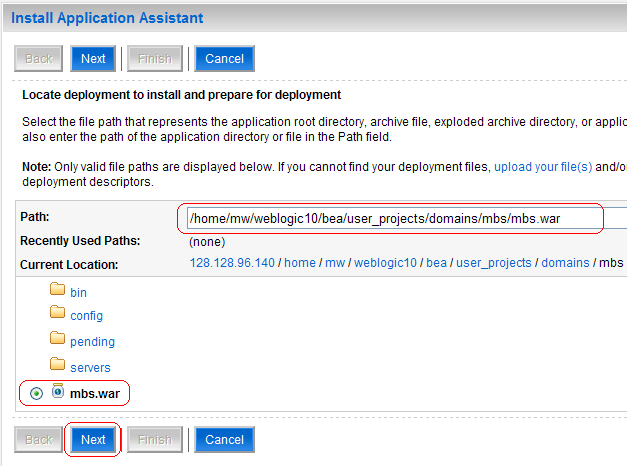
部署mbs应用程序

安装应用程序

在weblogic控制台，依次选择mbs >>[Deployments](http://128.128.97.45:7080/console/console.portal?_nfpb=true&_pageLabel=AppDeploymentsControlPage)>>Install选项：

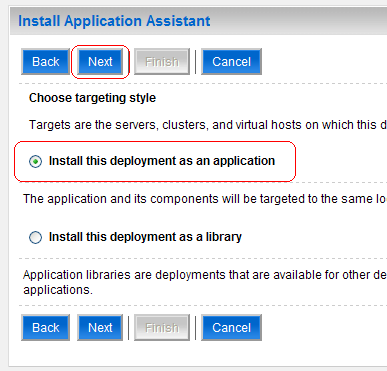


选择应用部署路径

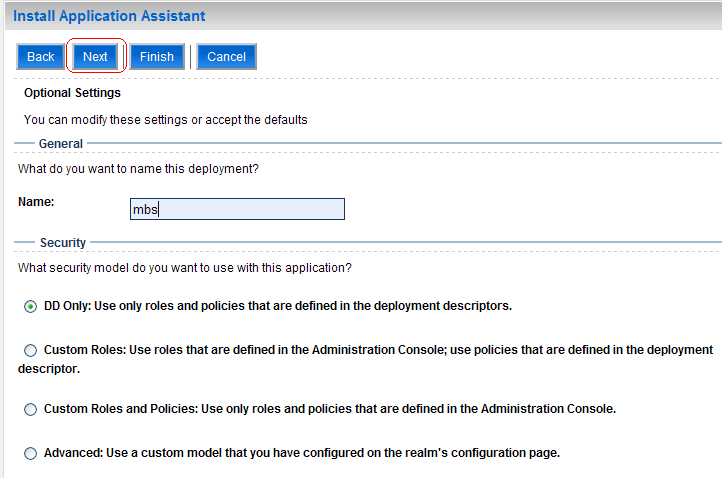


在上图中选中mbs.war应用前的单选按钮，然后点击Next

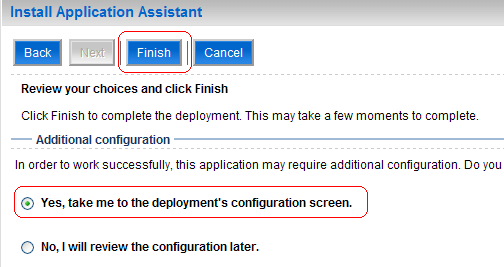
选择应用安装类型



使用默认选择，点击Next

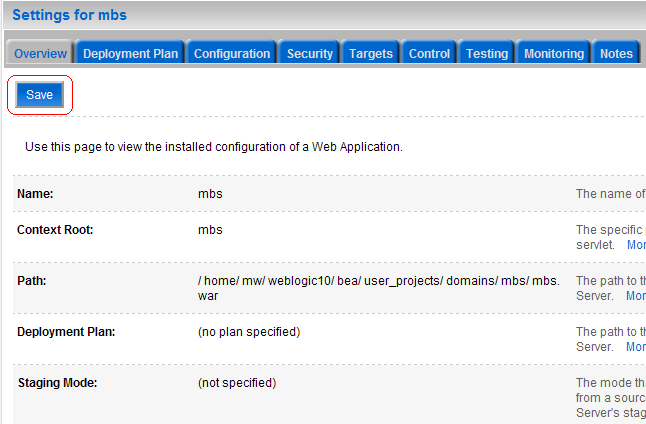


继续使用默认选择，点击Next

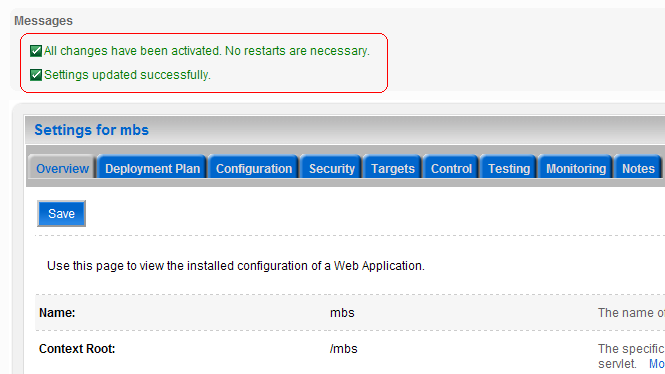


继续使用默认选择，点击Finish

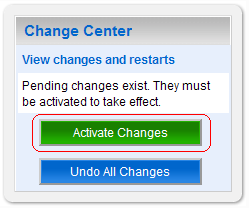
保存部署



点击上图的Save按钮保存安装，如果安装成功，则出现下图的提示



激活应用



点击上图的“Activate Changes”按钮激活应用。

重启weblogic服务

进入“/home/mw/weblogic10/bea/user\_projects/domains/mbs ”目录，先执行./stop.sh脚本停止服务，然后执行./start.sh启动服务。

备机更新

目前手机银行生产服务器采用两台服务器互备的模式运行，在更新完主服务器后，也需要同时对相应备份服务器的程序进行更新。

主服务器和备份服务器的程序目录结构一样。不同之处在于服务器的IP地址不同。更新备份服务器的操作步骤参考相应的主服务器程序更新步骤。

|  |  |  |  |
| --- | --- | --- | --- |
| **子系统** | **主服务器** | **备份服务器** | **登陆用户** |
| 平台服务器 | 200.1.5.86 | 200.1.5.93  200.1.5.92 | mbs |

##### 应用服务mbsTimejob的更新

###### 上传应用程序包

将mbsTimejob-86.war上线包上传到200.1.5.86机器的/home/mw/weblogic10/bea/user\_projects/domains/mbsTimejob目录下，然后删除mbsTimejob.war，再把mbsTimejob-86.war重命名为mbsTimejob.war。(**200.1.5.86服务器**)

将mbsTimejob-93.war上线包上传到200.1.5.93机器的/home/mw/weblogic10/bea/user\_projects/domains/mbsTimejob目录下，然后删除mbsTimejob.war，再把mbsTimejob-93.war重命名为mbsTimejob.war。(**200.1.5.93机器**)

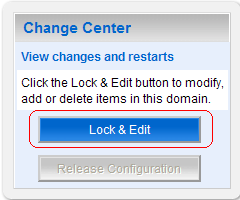
###### 发布程序包

此操作针对WEBLOGIC10.3版本。

登陆到weblogic控制台

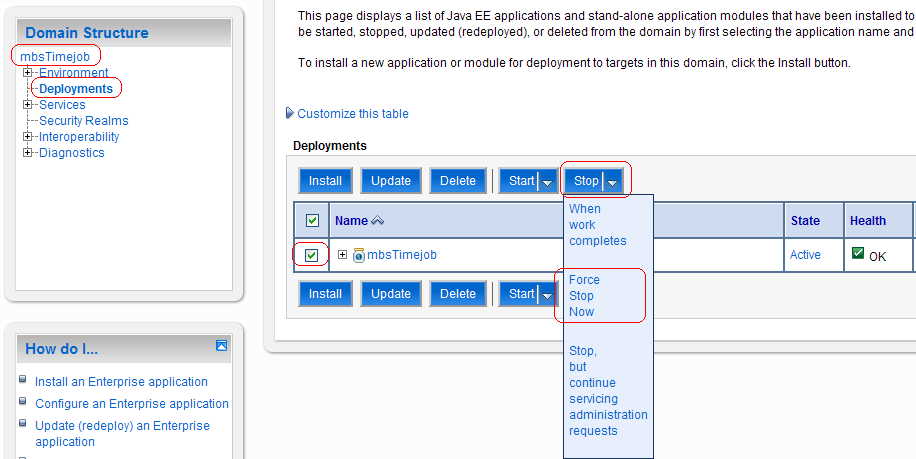
地址：[http://200.1.5.86:7010/86mbsTimejobnew](http://200.1.5.86:7010/86mbsTimejobnew%20)

然后点击下面的“Lock & Edit”按钮：

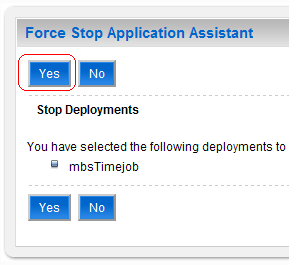


删除mbsTimejob应用

依次选择mbsTimejob>>[Deployments](http://128.128.97.45:7080/console/console.portal?_nfpb=true&_pageLabel=AppDeploymentsControlPage)>>选项：



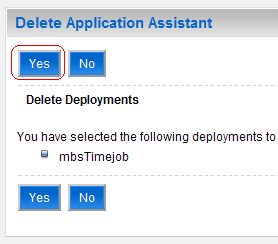
在右边出现的窗口中选择mbsTimejob应用左边的复选按钮，然后点击“stop”按钮，再选择下面的“Force Stop Now”按钮停止服务；



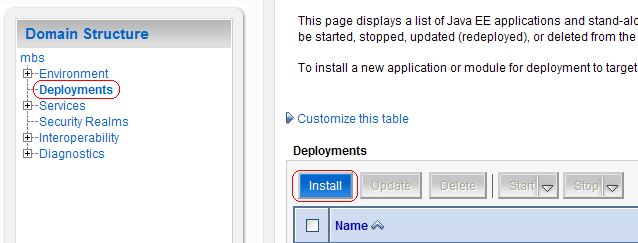
在出现的上面界面中点击“Yes”按钮



在出现的上面界面中选中mbsTimejob应用服务左边的复选框，然后点击上面的“Delete”按钮；



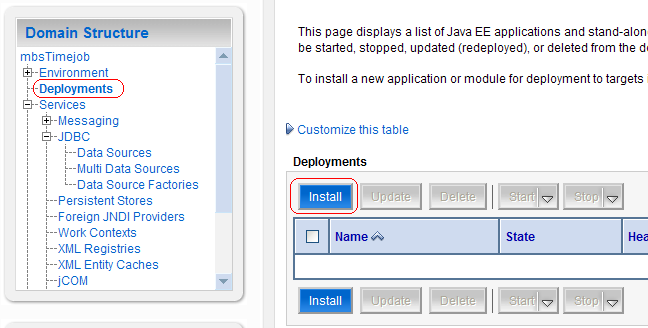
点击“Yes”按钮删除mbsTimejob应用。



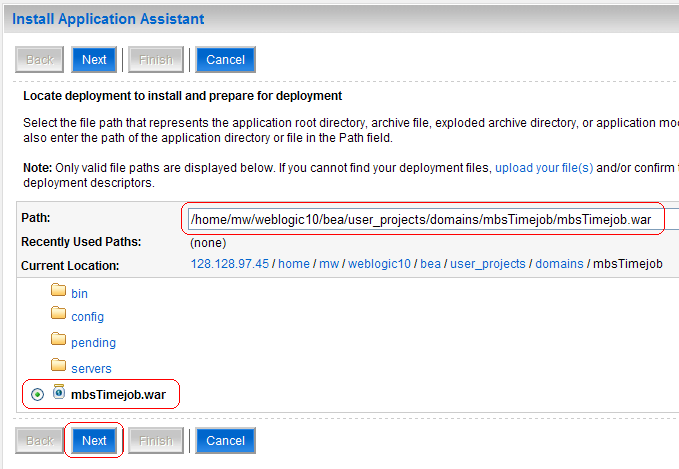
部署mbsTimejob应用程序

安装应用程序

在weblogic控制台，依次选择mbsTimejob>>[Deployments](http://128.128.97.45:7080/console/console.portal?_nfpb=true&_pageLabel=AppDeploymentsControlPage)>>Install选项：

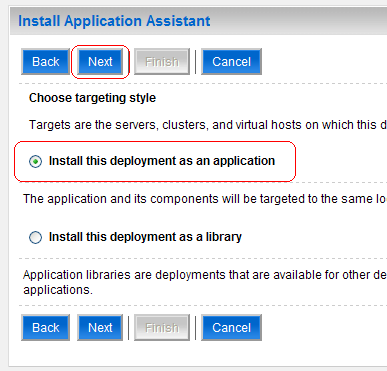


选择应用部署路径

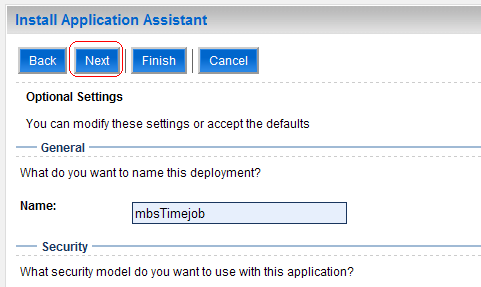


在上图中选中mbsTimejob.war应用前的单选按钮，然后点击Next

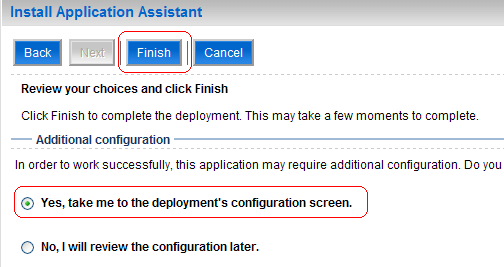
选择应用安装类型



使用默认选择，点击Next

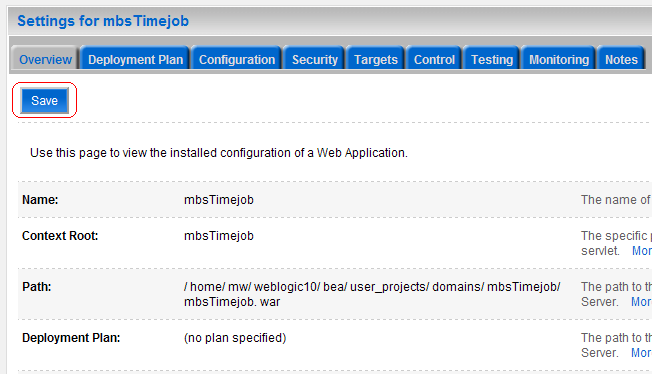


继续使用默认选择，点击Next



继续使用默认选择，点击Finish

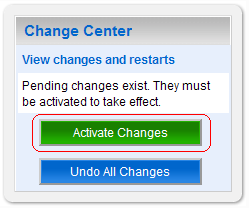
保存部署



点击上图的Save按钮保存安装，如果安装成功，则出现下图的提示



激活应用



点击上图的“Activate Changes”按钮激活应用。

##### 重启weblogic服务

进入“/home/mw/weblogic10/bea/user\_projects/domains/mbsTimejob”目录，先执行./stop.sh脚本停止服务，然后执行./start.sh启动服务。

**如果服务启动正常，则停止服务，原因是定时任务只在200.1.5.93机器上运行，200.1.5.86机器上部署的定时任务只是作为备份，不用启动服务。**

##### 备机更新

目前手机银行生产服务器采用两台服务器互备的模式运行，在更新完主服务器后，也需要同时对相应备份服务器的程序进行更新。

主服务器和备份服务器的程序目录结构一样。不同之处在于服务器的IP地址不同。更新备份服务器的操作步骤参考相应的主服务器程序更新步骤。（**确保200.1.5.93的定时服务是启动的**）

|  |  |  |  |
| --- | --- | --- | --- |
| **子系统** | **主服务器** | **备份服务器** | **登陆用户** |
| 平台服务器 | 200.1.5.86 | 200.1.5.93/92 | mbs |

### 应急方案

#### 执行回退脚本

当上线失败时，执行“MBS\_RETURN.sql”中的脚本回退本次上线数据库变化的表结构和数据修改(**停业执行**)。

具体操作步骤：

1. 以B2C20用户登录SQLPLUS；
2. 执行MBS\_RETURN.sql回退脚本；
3. 检查日志记录文件MBS\_RETURN.log的内容，看是否有异常；
4. 如果执行成功则提交事务：COMMIT；

#### 应用程序的回退

平台程序使用备份的WAR包覆盖现有的WAR包，然后重新在weblogic控制台部署，并重启服务；

WEB后台应用程序的回退，将备份程序解压缩后，拷贝对应的程序到原来的目录覆盖，并重启服务。

#### 后台日终处理 （涉及应用MANAGER）

上线期间，重新部署或停止MANAGER服务时，视操作的时间段可能影响以下的日终服务，请确认并执行相关手动补做操作。

目前后台有一些日终处理每天会定时运行，在上线前，可以通过修改/usr/wl6/Manager/config/etc/web.ini文件中的onoff参数值为1（1:维护状态； 0:正常运行状态），将系统处于维护状态，当天的日终处理就不再运行，维护状态只针对日终，对其他功能没有影响；待上线完成后（第二天凌晨7:30前），再将onoff参数值改为0即可。

对于CAS流水处理，待onff参数值改为0之后需要进行补做。

日终处理包括以下定时任务：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **日终名称** | **运行时间** | **频率** | **重复** |
| 1 | 同步基金信息日终 | 9:20 | 运行一次 | 每天 |
| 2 | CAS流水处理 | 2:30 | 运行一次 | 每天 |
| 3 | 同步联行号信息日终 | 8:00 | 运行一次 | 每天 |
| 4 | 上传信用卡挂失文件日终 | 18:05 | 运行一次 | 每天 |
| 5 | 同步营业网点、客户、账户数据日终 | 18:30 | 运行一次 | 每天 |
| 6 | 日志归档日终 | 19:00 | 运行一次 | 每天 |
| 7 | 同步商户信息 | 7:30 | 运行一次 | 每天 |
| 8 | 同步他行信息 | 7:50 | 运行一次 | 每天 |
| 9 | 生成验证码日终 | 0:05 | 运行一次 | 每天 |
| 10 | 同步ATM特约取款数据 | 7:25 | 运行一次 | 每天 |
| 11 | ODS供数服务 | 2:00 | 运行一次 | 每天 |
| 12 | 从日志表获取客户号和手机号服务 | 2:00 | 运行一次 | 每天 |
| 13 | 卡BIN数据同步服务 | （每30分钟自动运行） | 每30分钟自动运行 | / |
| 14 | 清理客户账户子账户表数据 | 3:30 | 运行一次 | 每天 |
| 15 | 同步企业缴费项目数据 | 7:20 | 运行一次 | 每天 |
| 16 | 归档BC\_LOG\_EB表数据到历史表 | 22:00 | 运行一次 | 每天 |
| 17 | 同步商户网点排队AUM数据服务 | 02：00 | 运行一次 | 每天 |
| 18 | 周末尽情摇生成中奖时间点和客户摇奖资格 | 00：30 | 运行一次 | 每天 |
| 19 | 同步网点业务类型 | 02：30 | 运行一次 | 每天 |
| 20 | 从CAS导验证码到手机银行 | 02：00 | 运行一次 | 每天 |
| 21 | 客户之声反馈数据导给CAS | 02：00 | 运行一次 | 每天 |

#### 业务功能上线BUG处理

##### 手机银行登录密码的有效期检查

手机银行登录密码的有效期设置为180天，超过180天的登录密码需强制修改（旧版不处理，要兼容客户可正常登录）

当该功能出现问题时需要执行SQL脚本回退

SQL：UPDATE CM\_SYS\_PARA\_EB SET SYS\_PARA\_VALUE='0101' WHERE SYS\_PARA\_CODE='UPDATE\_PASS\_FLAG' AND SYS\_ID='\*' AND LANGUAGE='CN';

对于在试运行或者正式投产后若有发现重大BUG的业务功能，可登录手机银行监控平台关闭相关交易，并将应用运行日志传给项目组，由项目组分析并指定解决方案。

##### 单一绑定的控制

当该功能出现问题时需要执行SQL脚本回退

SQL：

UPDATE CM\_SYS\_PARA\_EB SET SYS\_PARA\_VALUE='0' WHERE SYS\_PARA\_CODE='SINGLE\_BIND\_SWITCH' AND SYS\_ID='\*' AND LANGUAGE='CN';

##### 签约之后超时绑定的控制

当该功能出现问题时需要执行SQL脚本回退

SQL：

UPDATE CM\_SYS\_PARA\_EB SET SYS\_PARA\_VALUE='0' WHERE SYS\_PARA\_CODE='OVERTIME\_BIND\_CTRL' AND SYS\_ID='\*' AND LANGUAGE='CN';

##### 分流ECTIP交易迁移

mbs\_timejob定时任务的定时更新人民币牌价 910100接口：先请求 P1SVC，如果 P1SVC有返回数据就更新数据库中的人民币牌价信息，且不再请求 ECTIP；如果 P1SVC没返回就请求 ECTIP更新数据库。定时更新人民币牌价每天只做一次。

当该功能出现问题时需要执行 SQL脚本回退

SQL:

UPDATE CM\_SYS\_PARA\_EB SET SYS\_PARA\_VALUE='0' WHERE SYS\_PARA\_CODE='P1\_SERVER' AND SYS\_ID='\*' AND LANGUAGE='CN'

;

#### 上线程序遗漏或配置错误处理

如果上线程序中遗漏了部分程序或某个配置文件配置错误，视现场问题情况由项目组确定是以全量或增量方式进行重新打包，并提交后进行重新部署。