

中国电信集团公司部门文件

信息[2007]21号

关于做好 IT 系统对 e8-2 产品支撑的通知

北京公司、西藏公司、北方公司；股份公司并转各省级电信有限公司：

为支撑“我的 e 家”客户品牌推广，强化产品内涵填充，做好全面启动 e8-2 的各项准备工作（见市场[2007]70 号）的 IT 支撑准备。集团公司在 9 月份下发的《关于做好 BSS 对客户品牌经营工作支撑的通知》（信息[2007]19 号）以及 11 月 13 日的“我的 e 家定制终端 IT 支撑电视电话会议”基础上，完善 IT 支撑方案并对相关工作进一步明确如下：

一、e8-2 产品涉及主要 IT 系统支撑要求

各省 CRM 系统要新增或完善营销资源管理能力，实现终端属性和规格数据的管理以及营业厅库存管理；需实现营业厅发放终端并将序列号送服务开通系统；新增 WLAN 漫游和 e 家电话客户资料及产品信息送集团 CRM 功能，以便集中开通和管控。

服务开通系统要新增与 ITMS 的接口，完成零配置功能；新增与软交换设备开通接口，实现 e 家电话开通。计费系统要实现 WLAN 漫游计费、e 家电话计费和多终端上网计费的能力。网上客服要完成与互联星空和 e 家客户端单点登录的功能，提升客户感知。详细方案见附件《我的 e 家定制终端 IT 支撑方案 V1.0》。

二、IT 系统改造、联调测试的时间要求

上海、江苏、福建、广东四试点省份请按照业务推进要求于 2007 年 12 月底前完成试点改造 IT 系统在所有本地网的配置、测试工作；其余省应于 2007 年 12 月底前完成 CRM、计费、开通、网客系统相关功能的开发、配置和测试，1 月份完成集成测试。各省要着重保证服务开通与 ITMS 接口的稳定性，保证计费的及时和准确性。

已经与集团 CRM 完成 WLAN 全国漫游和 e 家电话集中管控接口测试的上海、江苏、浙江、福建、广东、云南、四川七省，请积极配合集团公司完成端到端业务加载测试；请其他省份及早准备，最迟于 2007 年 12 月 15 日与集团进行接口测试，并准备于 2008 年 1 月 10 日前与集团公司进行端到端业务加载测试。

三、联系人

（一）集团公司联系人

王桂荣（总联系人，侧重零配置），电话 010-58501473，
EMAIL: wanggr@chinatelecom.com.cn;

张少明（侧重单点登陆），电话 010-58502249，
zhangsm@chinatelecom.com.cn

周伟（侧重 CRM 互联），电话 010-58501465，
EMAIL: zhouwei@chinatelecom.com.cn;

苏更殊（侧重业务开通），电话 10-58501841，
EMAIL: sugu@chinatelecom.com.cn;

冯京（侧重计费），电话 010-58501611，EMAIL:
fengj@chinatelecom.com.cn;

（二）集团 BSS/OSS 支撑中心联系人

刘斌，电话 020-38639056，Email: liubin@gsta.com

刘智琼，电话 020-38639767，Email: liuzq@gsta.com

附件：我的 e 家定制终端 IT 支撑方案 V1.0



附件：

我的 e 家定制终端 IT 支撑方案

V1.0

中国电信集团公司

2007 年 11 月

目 录

1	综述.....	4
1.1	背景.....	4
1.2	适用范围.....	4
2	我的 E 家定制终端与产品填充简介	4
2.1	E 家定制终端设备.....	4
2.1.1	设备形态.....	4
2.1.2	设备在网络中的位置.....	6
2.2	E 家统一客户端.....	7
2.2.1	主要功能.....	7
2.2.2	网络结构.....	9
2.2.3	登录流程.....	10
2.3	WLAN 漫游及 E 家电话简述.....	12
2.3.1	WLAN 漫游.....	12
2.3.2	e 家电话.....	15
2.4	E8-2 套餐框架	18
3	我的 E 家定制终端 IT 支撑方案	20
3.1	总体支撑要求	20
3.1.1	CRM 支撑要求	20
3.1.2	服务开通支撑要求.....	23
3.1.3	计费结算支撑要求.....	23
3.1.4	网上客服中心支撑要求.....	23
3.1.5	网上客服-单点登陆要求.....	23
3.1.6	典型开通流程示例.....	24
3.2	零接触业务配置支撑要点	25
3.2.1	设备由营业前台统一管理及发放.....	26
3.2.2	设备由外线部门统一管理及发放.....	26
3.2.3	宽带密码下发.....	26
3.3	单点登陆接口	27

3.3.1	基本流程.....	27
3.3.2	接口定义.....	29
3.3.3	附表.....	36
4	WLAN 漫游及 E 家电话开通支持.....	39
4.1	流程.....	40
4.1.1	省 CRM/服务开通同步 WLAN 业务数据流程.....	40
4.1.2	省 CRM/服务开通同步 e 家电话数据流程.....	41
4.2	接口概述.....	41
4.2.1	集团 CRM 与省 CRM/服务开通系统接口.....	41
4.2.2	集团 CRM 与后台业务系统接口.....	42
4.3	接口.....	42
4.3.1	GetCustOrderInfo (获取各省用户业务数据接口).....	42
4.3.2	UpdateWLANInfo (集团 CRM 与 WLAN 全国漫游认证系统的数据同步接口).....	45
4.3.3	UpdateVoipNumber (集团 CRM 与综合通信平台接口).....	48
5	附件.....	50
5.1	附件一 终端资源管理.....	50
5.1.1	终端资源管理流程描述.....	50
5.1.2	e 家终端设备参数配置.....	52
5.1.3	e 家终端设备实物入库.....	52
5.1.4	e 家终端设备实物入库撤销.....	53
5.1.5	e 家终端设备下发.....	54
5.1.6	e 家终端设备调拨.....	55
5.1.7	e 家终端设备回收.....	56
5.1.8	e 家终端设备退库.....	57
5.1.9	e 家终端设备盘点.....	58
5.1.10	定制 e 家终端设备信息管理.....	59
5.2	附件二：加解密算法.....	60
5.2.1	概要说明.....	60
5.2.2	Java 下 3DES 调用例子.....	60
5.2.3	.NET 下 3DES 调用例子.....	65

1 综述

1.1 背景

中国电信面向家庭客户，推出“我的 e 家”客户品牌，旨在通过“我的 e 家”客户品牌，为家庭客户提供全方位综合信息服务，让家庭成员可以更加自由、便利、可靠地在家庭网络环境中使用电话、宽带、无线上网、视频娱乐、信息定制与推送等多种应用。

为支撑我的 e 家客户品牌推广，通过定制终端的部署有效承载组合产品，增强客户感知。定制终端包括硬终端：e 家终端设备的 e8-B，e8-C；软终端：e 家统一客户端。

为有效支撑“我的 e 家”定制终端的引入，需要 IT 系统的有力支撑。

1.2 适用范围

本文适用于指导各省我的 e 家定制终端的部署，特别是涉及定制终端的资源管理、受理、开通、单点登陆时，IT 系统的配置、改造与完善工作。同时本文还描述了 WLAN 漫游及 e 家电话开通方案，适用于指导省 CRM 与集团 CRM 的接口开发，以支持 WLAN 全国漫游及 e 家电话的开通。

2 我的 e 家定制终端与产品填充简介

2.1 e 家定制终端设备

2.1.1 设备形态

包括以下两种形态的 e 家终端设备：

物理接口	e 家终端设备形态
------	-----------

		e8-B型	e8-C型
上行	ADSL2+ 或 LAN	√	√
下行	4 * 以太网接口	√	√
	1 * WLAN接口	√	√
	1 * USB Host 接口（至少1个）	√	√
	1 * POTS接口		√
		本 产 品 形 态 特 征： 设备具备高速上 网、允许多用户接入 特征、方便扩展，未 内置IAD功能	本 产 品 形 态 特 征： 设备具备高速 上网、允许多用户 接入特征、方便扩 展，内置IAD功能

表 2-1 e家终端设备形态

2.1.2 设备在网络中的位置

2.2 e 家统一客户端

2.2.1 主要功能

我的 e 家统一客户端软件是中国电信面向家庭和个人用户推出的一款集综合通信与互联网应用与内容服务于一体的综合信息服务终端软件。

e 家统一客户端包括用户登录、e 家客服、e 家信息、e 家娱乐、e 家通信五个频道。

用户通过宽带账号登录客户端，登录前可先进行宽带拨号，客户登录后，点击页面栏目，即直接进入相应的服务页面。

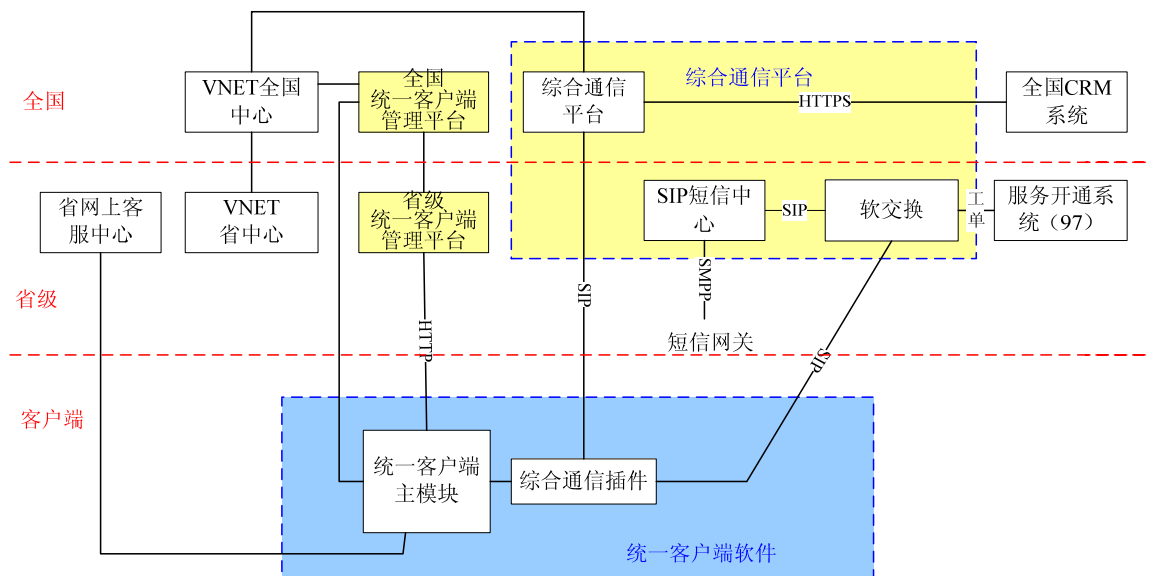
e 家客服链接到网上客服中心“我的 e 家”客户服务专区，由网上客服中心提供服务，客户无须再次进行身份验证。

e 家信息和 e 家娱乐链接到互联星空“我的 e 家”信息娱乐专区，由互联星空平台提供服务，客户无须再次进行身份确认。

e 家通信为客户提供 e 家通信等通信服务，栏目集成相应的通信插件。



2.2.2 网络结构



个人家庭统一客户端系统主要由个人家庭统一客户端软件、统一客户端管理平台、综合通信平台组成。

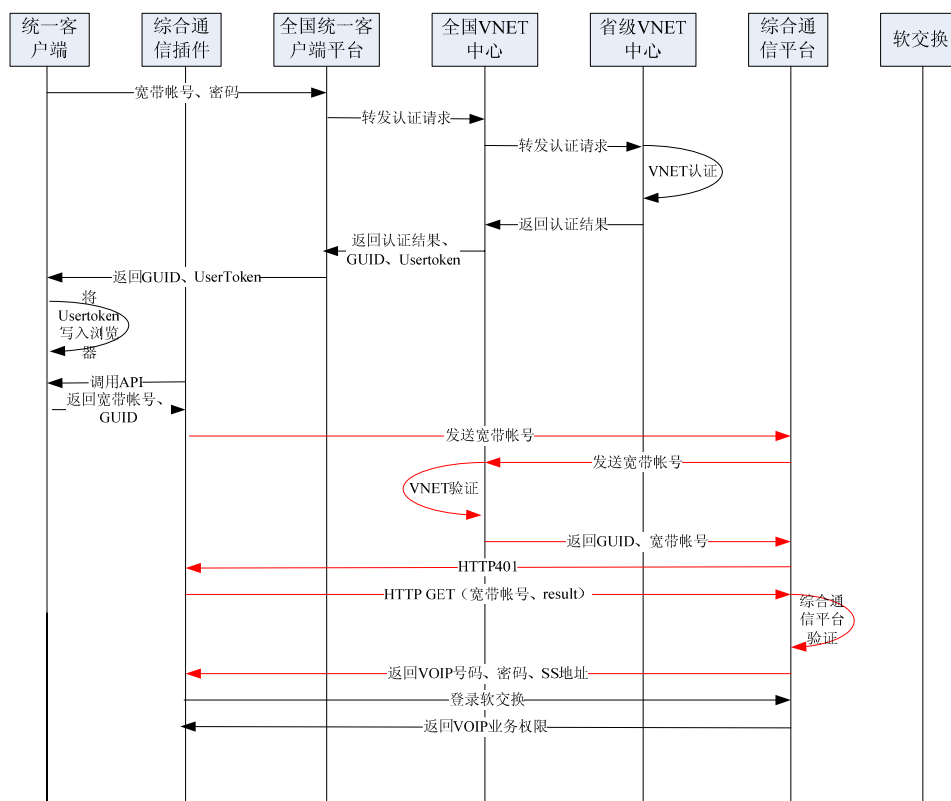
统一客户端软件由主程序模块和综合通信插件构成，综合通信插件提供e家电话、短信、通讯录等与通信相关的功能。

统一客户端管理平台：客户端管理平台是为了对客户端实现集中管理而专门构建的系统，由全国客户端管理平台和省级客户端管理平台两部分组成，全国管理平台负责转发客户端的用户认证请求，同时还是全国的客户端管理服务中心，负责统一客户端软件版本管理、插件管理以及用户数据汇总统计，全国客户端管理后台负责和各级客户端管理后台通信，不直接面向最终用户；省级客户端管理后台负责直接面对最终用户，提供软件版本更新、插件下载、通知下发、数据搜集的功能，同时还负责将本省的 SP 插件递交全国管理后台审核并定期上报省级用户数据。

互联星空业务支撑平台：承担统一客户端的用户认证功能，并支持用户漫游。
客户端用户在进行用户认证时由全国客户端管理平台向互联星空全国中心转发认证请求，再由全国中心转到用户的归属省认证。

综合通信平台：提供 e 家电话、短信等客户端的通信能力，完成用户在通信系统的注册、认证、注销功能，并通过与软交换平台、短信中心平台交互，实现客户端的语音通信和短信收发功能。

2.2.3 登录流程



e 家客户端登陆流程：

- 1) 统一客户端向全国统一客户端平台发起登录认证请求，请求中携带用户宽带帐号/密码；
- 2) 统一客户端平台向全国互联星空平台发起认证请求；

- 3) 互联星空全国中心通过省间漫游接口转发到互联星空省平台进行校验，校验完成，互联星空省平台向全国平台发送 UserID 和该宽带帐号的 e 家属性。
- 4) 全国互联星空平台向全国统一客户端平台返回认证结果，包含 GUID、Usertoken 等相关信息；
- 5) 统一客户端平台向统一客户端返回认证结果，UserToken，GUID 等相关参数，完成登陆。

e 家电话认证流程：

- 1) 综合通信插件通过调用统一客户端的 API 接口，获取客户端登录信息；
 - 2) 统一客户端向综合通信插件返回宽带帐号和 GUID；
 - 3) 综合通信插件向综合通信平台发起认证请求，请求中携带用户宽带帐号；
 - 4) 综合通信平台向全国互联星空平台发起认证请求，请求中携带用户 ADSL 帐号；
 - 5) 全国互联星空平台查询该帐号登录状态，如果已经登录，向综合通信平台返回该宽带帐号对应的 GUID；
 - 6) 综合通信平台向综合通信插件发送 401 消息，其中包含加密的鉴权挑战字，该加密的鉴权挑战字是由综合通信平台生成明文鉴权挑战字后与 GUID 通过 3DES 算法加密后生成；
 - 7) 客户端对加密的的鉴权挑战字进行解密得到明文的鉴权挑战字，将鉴权挑战字，宽带帐号和 GUID 采用 MD5 算法得到挑战结果，客户端向综合通信平台 AS 发送挑战结果，综合通信平台将鉴权挑战字，宽带帐号和 GUID 采用 MD5 算法得到挑战结果进行鉴权，比较由综合通信插件返回的结果；
 - 8) 如果鉴权通过，综合通信平台根据登录使用的宽带帐号查询，如果该宽带帐号对应的用户申请了 e 家电话业务，则由综合通信平台返回该用户的 VOIP 号码、密
-

码及软交换的 BAC 地址, 综合通信平台的 BAC 的地址信息等;

9) 综合通信插件向软交换发送 VOIP 号码和密码, 请求登录;

10) 软交换返回登录结果和该 VOIP 号码的业务权限。

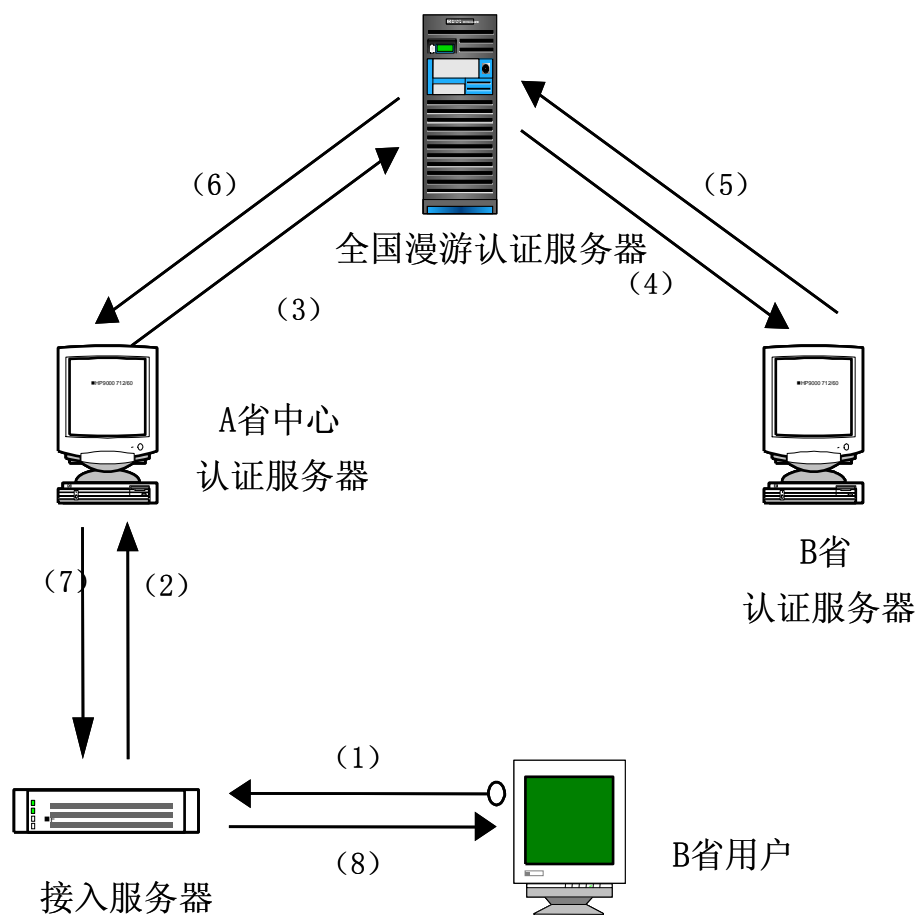
2.3 WLAN 漫游及 e 家电话简述

2.3.1 WLAN 漫游

全国 WLAN 漫游的目标客户为: 我的 e 家高端客户和商务领航客户。

漫游帐号格式要求参见《关于正式开通 WLAN 全国漫游的通知》(网发[2003]122 号)。

2.3.1.1 WLAN 漫游认证流程:



(1) B 省用户以漫游方式上网，用户使用原账号名和口令；

(2) 省 A 城域网接入服务器（BRAS）向 A 省中心 Radius 服务器发出认证请求；

(3) A 省中心 Radius 服务器将漫游认证请求转发到全国中心 Radius 服务器；

(4) WLAN 全国漫游认证系统查询 radius 库判断该用户是否是我的 e 家高端用户或是商务领航用户。如果不是则认证不通过，转到 6。如果是，则将漫游认证请求转发到开户 B 省中心 Radius 服务器；全国中心保存认证信息；

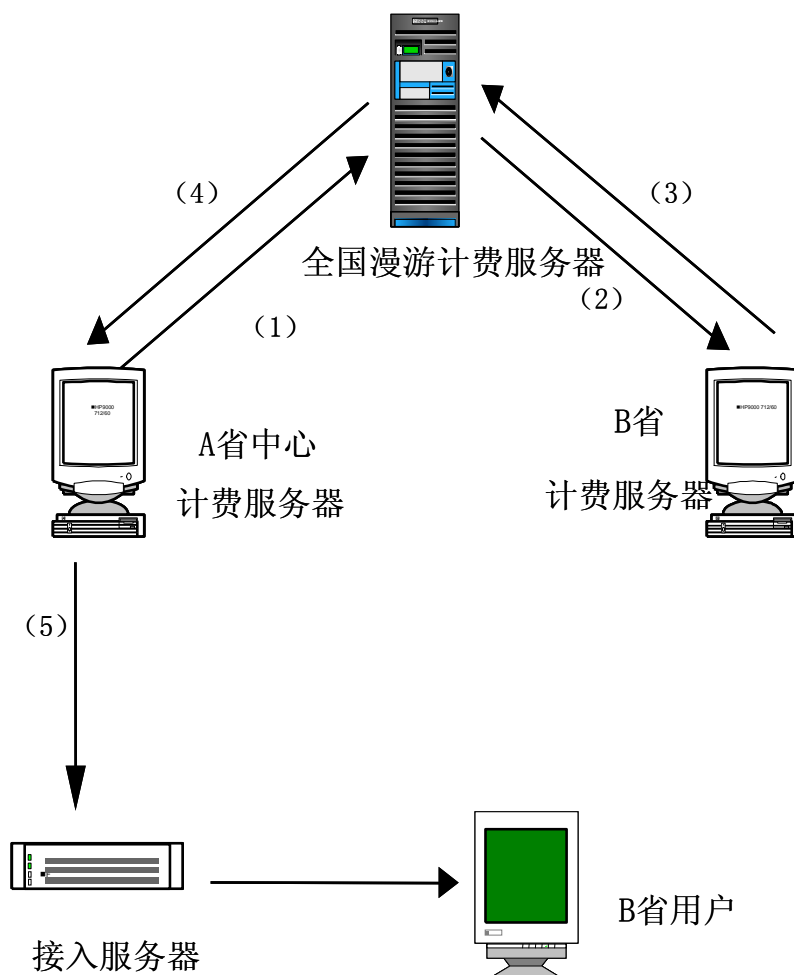
(5) B 省中心 Radius 服务器对本省用户进行认证，并向全国中心 Radius 服务器发送授权信息；

(6) 全国中心 Radius 服务器将授权回应给 A 省中心 Radius 服务器；全国中心保存认证授权信息；

(7) A 省中心 Radius 服务器向 BRAS 授权；

(8) BRAS 与用户接入设备建立连接。

2.3.1.2 WLAN 漫游计费流程:



- (1) A 省计费服务器收到 NAS 的计费请求，发现是异地开户的用户，则向全国中心转发计费请求；
 - (2) 全国中心收到计费请求后，存储后转发给开户省 B 的计费服务器；
 - (3) 开户省 B 计费服务器在成功记录后，发送响应信息给全国中心；
 - (4) 全国中心在存储响应信息后转发给漫游省 A 计费服务器；
 - (5) 漫游省 A 的计费服务器收到响应信息后，记录计费信息。成功记录后向 NAS 发回响应。
-

2.3.1.3 WLAN 漫游计费和结算：

◆ 计费

- 中国电信的 WLAN 用户的省际漫游计费标准：0.2 元/分钟
- 上网时长按月累计，不足 1 分钟按照 1 分钟计算；

◆ 结算

- WLAN 全国漫游统一结算比例，开户省/漫游省=5/5（结算给漫游地 0.1 元/分钟）。
- 结算时长按月累计，不足 1 分钟按 1 分钟算。

2.3.2 e 家电话

2.3.2.1 “e 家电话”主要功能

◆ VoIP：

- 基本电话功能
- 增值业务

◆ 同号同振

◆ 被叫彩铃

- 补充业务

◆ 主叫号码显示

◆ 呼叫前转（遇忙、无条件、无应答）

- 视频通信功能
-

◆ 短信

- 本网/异网短信互发
- 语音同号时，短信做同号处理
- 短信同号时，同号终端同时接收
- 可捆绑手机或小灵通终端，与客户端同时接收短信

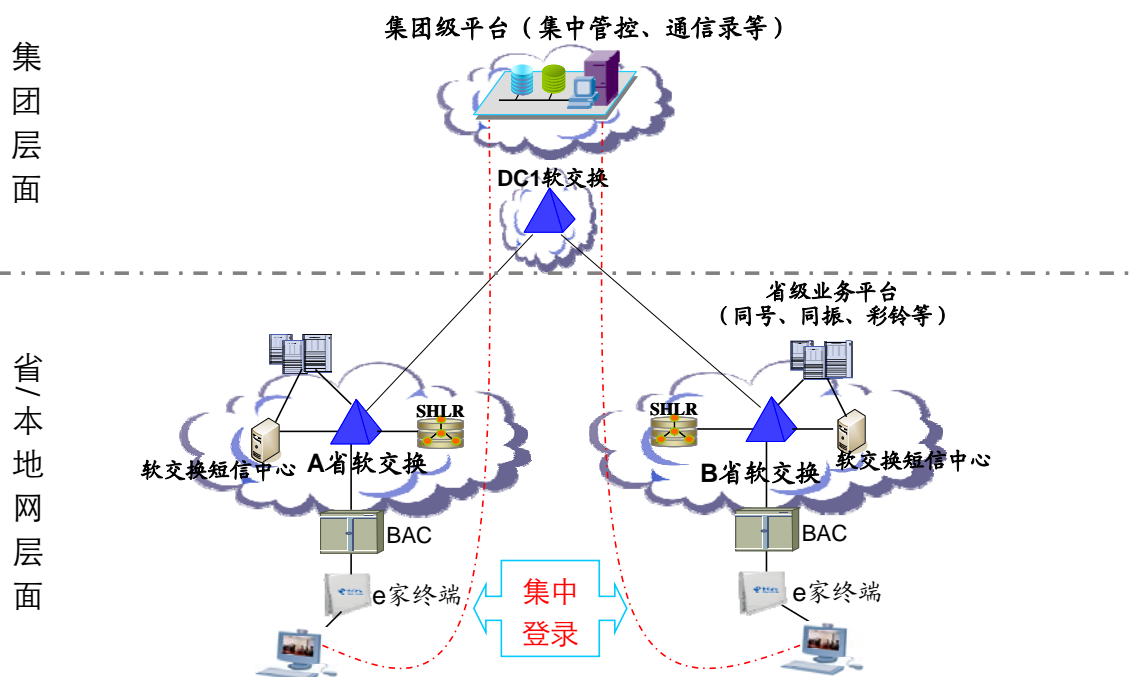
◆ 通信录

- 通过通信录进行联系人管理
- 点击通信录发起电话呼叫 (VoIP) 和短信业务

◆ 集中管控

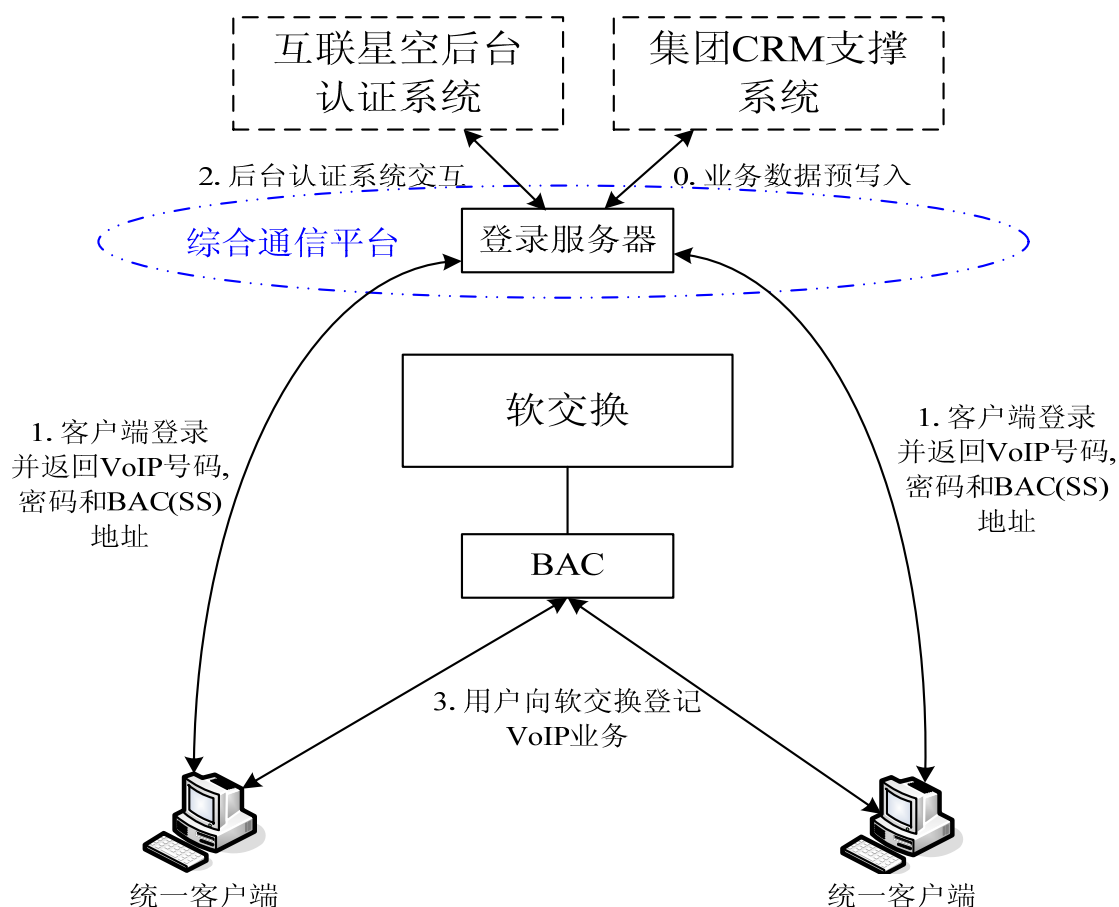
- 集团集中控制用户开放范围
- 支持软客户端的漫游功能

2.3.2.2 “e 家电话”业务组网架构



2.3.2.3 集中管控方案

e 家电话采用集中登录、归属服务的方案，e 家电话用户统一在全国综合通信平台登录，认证鉴权通过后，由各省软交换网络提供服务。



2.4 e8-2 套餐框架

e8-2 套餐相比原来 e8 套餐的主要创新点在于：

- 1、引入了 e8-B 定制终端，而非原来的普通 ADSL Modem
- 2、丰富的产品填充：e 家电话、WLAN 漫游、iTV 等

e8-2 套餐框架如下：（包含参数为示意）

◆ 必选包：

□ e 家通信

高速宽带：4M，不限时，提供 多终 端、多帐号上网功能

WLAN 无线宽带：20-100 小时

通话时长（含固定电话及软终端）：400 分钟

固话来电显示和彩铃，每月送 3 个铃音

□ e 家娱乐

iTV：多套直播节目， 提供时移、回看功能

星空精品包：在线杀毒、防火墙、影视、教育等

□ e 家信息

资讯频道、电子杂志等

社区信息等

◆ 优惠可选包

□ 小灵通特惠包：加 10-15 元/月，可捆绑 1 部小灵通（最多 3 部），免月租，共享通话时长，互打免费

□ 通信助理包：3-5 元/月，家庭通信助理工具（百事通、呼叫等待、号码簿等）

□ iTV 视频点播

3 我的 e 家定制终端 IT 支撑方案

3.1 总体支撑要求

3.1.1 CRM 支撑要求

■ 终端资源管理

需要实现对 e 家终端设备的资源管理，包括 e 家终端设备参数配置、入库、出库、分配、调配、回收、盘点等功能。

➤ e 家终端设备参数配置

针对 e 家终端设备的各项属性参数进行配置维护的功能，终端设备属性包括：e 家终端设备型号、e 家终端设备 ID、供应商、来源、状态等。

➤ e 家终端设备实物入库

e 家终端设备实物入库是指对 e 家终端设备实物入库操作。

➤ e 家终端设备实物入库撤销

e 家终端设备实物入库撤销是指对 e 家终端设备实物入库操作的撤消处理。

➤ e 家终端设备下发

e 家终端设备下发是指将 e 家终端设备由上级下发给下级的过程。

➤ e 家终端设备调拨

e 家终端设备调拨是指对已经下发到渠道管理部门的且未被使用的 e 家终端设备进行调配的操作。从一个部门调配到另外部门的操作，是由上一级机构生成调拨单，地市之间调拨由省分生成调拨单，区县之间调拨由地市生成调拨单，营业厅之间调拨由区县生成调拨单，由被调拨单位进行实物调出，调入单位校验后进行实物入库。

➤ e 家终端设备回收

e 家终端设备回收是指对下发的 e 家终端设备进行回收处理。

➤ e 家终端设备退库

e 家终端设备退库是指将准备退库的 e 家终端设备状态改为已退库，以便退回厂家后处理。

➤ e 家终端设备盘点

e 家终端设备盘点是指对 e 家终端设备的信息进行盘点。通过定期盘点对 e 家终端设备库存进行监控，有效的把 e 家终端设备资源管理起来，降低库存，同时对 e 家终端设备销售情况进行有效的监控。

➤ 定制 e 家终端设备信息管理

定制 e 家终端设备关键信息指定制 e 家终端设备符合中国电信制定的标准，在外观、专用键、菜单呈现、接口及通用要求等方面的信息。

定制 e 家终端设备关键信息管理指对各种定制 e 家终端设备机型定制信息的管理。

■ 产品管理

完成 e 家终端设备涉及的产品、服务、销售品的配置。功能要求如下：

- 支持 e 家终端设备产品的定义和配置，支持 e 家终端产品与营销资源（e 家定制终端）的关联关系；
- 应能设置 e 家终端设备产品与宽带等其它产品的依赖、互斥关系；
- 支持 e 家终端设备产品，加入我的 e8 销售品的配置，可以针对其收取一次性费用；

■ 综合客服

综合客服支持 e 家终端设备相关的受理、收费、查询以及投诉故障等各项业务的办理。功能要求如下：

➤ 订购

- 不能在没订购宽带的情况下单独订购 e 家终端设备；
- 记录 e 家终端设备产品实例信息；
- 根据产品管理配置的价格以及用户订购的 e 家终端设备套餐，自动计算一次性费用。
- 支持 e 家电话业务的受理，记录对应的宽带账号，支持 e 家电话号码与固话同号
- 支持 WLAN 漫游的受理
- 省 CRM/服务开通与集团 CRM 的接口作为 WLAN 漫游与 e 家电话开通中的一个环节

➤ 退订

- 检查是否有依赖于 e 家终端设备的产品订购，如有则同步触发相关产品的退订
- 退订 e 家终端设备产品时，需同时拆除宽带；

➤ 订单分解

- 根据产品、服务等规则，对受理的 e 家终端设备套餐客户订单进行分解，形成服务定单，发送给服务开通系统；

➤ 竣工

- 在 CRM 系统完成订单的竣工归档；
 - 在订单竣工时，根据套餐的生失效时间规则，动态生成套餐的生失效时间，连同订单的其他数据同步给计费系统。
-

3.1.2 服务开通支撑要求

接收 CRM 系统的服务定单，根据流程模版进行流程调度，与业务平台、ITMS 交互，实现 e 家终端设备销售品包含产品的开通，功能要求如下：

- 应能支持 e 家终端设备关联流程配置和协同调度；
- 应能支持 e 家终端设备定单的优先级处理；
- 在原有流程上，增加数据传送到 ITMS 的环节；
- 支持 e 家电话在 SHLR 的自动开通流程。

3.1.3 计费结算支撑要求

- 能够从软交换采集话单，支持 e 家电话计费；
- 支持 WLAN 漫游计费，支持按时长计费、按流量计费；
- 支持多终端上网计费。

3.1.4 网上客服中心支撑要求

参照“中国电信网上客服中心业务规范（V1.0）补充规范”的要求，

重点实现清单、帐单、余额查询，积分查询、积分兑换，能力通过 CRM、计费等核心系统提供接口服务来实现。

3.1.5 网上客服-单点登陆要求

需要完成网上客服、我的 e 家统一客户端和互联星空平台的单点认证。具体要求如下：

- 场景一：用户登录我的 e 家客户端后，访问网上客服的时候无需再次登录。
- 场景二：用户登录网上客服后，访问互联星空的时候无需再次登录。

约束：

- 网上客服中心、互联星空与我的 e 家统一客户端相互之间的登录用户指宽带用户，随着业务的不断扩展，该用户将可能涉及全产品。
- 通过点击特定导航按钮实现。如果用户直接在浏览器地址栏输入网上客服、互联星空网站的网址，则无法实现单点登录。

3.1.6 典型开通流程示例

典型的 e 家终端设备开通流程示例如下：

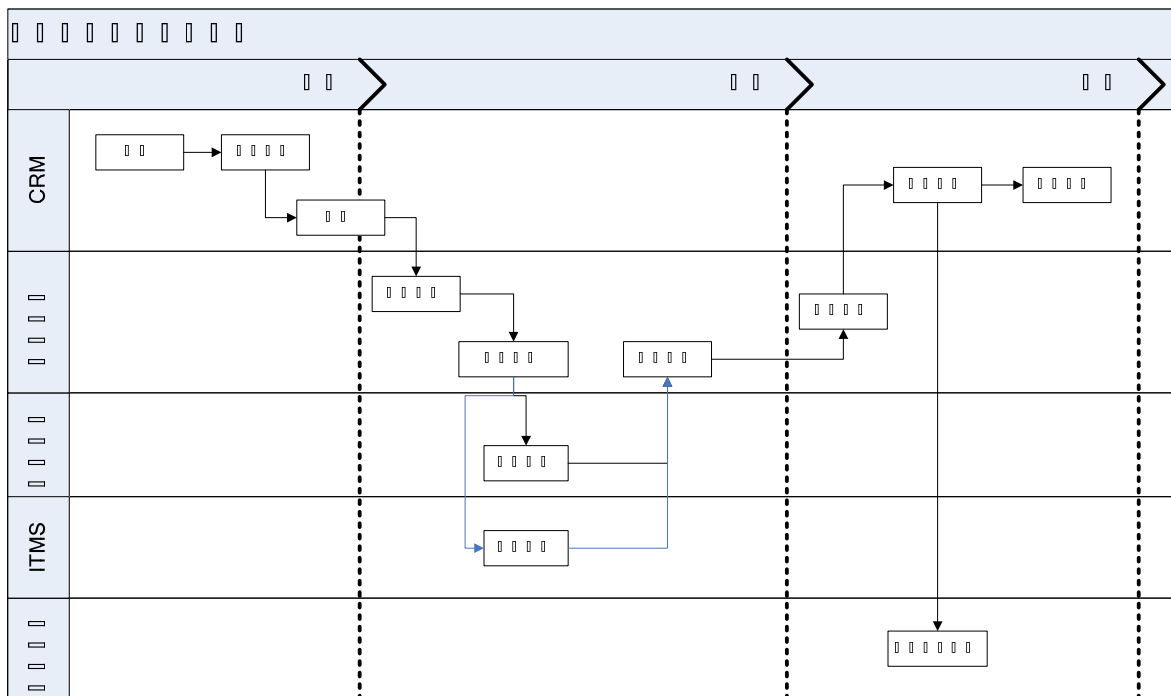


图 2-3 典型开通流程

流程说明：

- 在 CRM 系统受理 e 家终端设备套餐，生成客户订单；
- CRM 系统依据业务规则，分解客户订单，形成关联的服务定单；
- CRM 系统完成稽核后，将服务定单送往服务开通系统；
- 服务开通接收到服务定单后，根据对应的产品流程模板，进行流程实例化；
- 基于实例化的流程和关联关系，进行协同调度；
- 在施工环节，根据是否能自动处理，生成人工工单或自动工单，发往业务平台；
- 把业务参数信息，发往 ITMS，由 ITMS 实现对 e 家终端设备的参数下发配置；
- 业务平台、ITMS 处理完成后，在服务开通系统进行工单回笼；
- 全部环节处理完毕后，进行服务定单的竣工处理；
- 服务开通系统将竣工后的服务定单发往 CRM 系统；
- CRM 系统待所有关联的服务定单均已完成竣工后，进行订单的竣工处理，并进行客户回访；
- CRM 系统同时将竣工后的订单（客户资料、产品实例、订购关系、支付关系、套餐生效时间等）同步给计费系统。

注：如果套餐中包含 WLAN 漫游及 e 家电话的开通，则开通需包括用户数据同步到集团 CRM 环节。

3.2 零接触业务配置支撑要点

根据设备统一管理及发放部门的不同，我的 e 家终端设备有以下两种可能出现

的设备管理模式：

模式一：设备由营业前台统一管理及发放；

模式二：设备由外线部门统一管理及发放。由于设备管理模式的不同，受理开通流程会存在一些差异。

3.2.1 设备由营业前台统一管理及发放

- 开通需要增加服务开通与 ITMS 交互的环节；
- 营业厅受理，设备发放给客户时，需录入 e 家终端信息；
- 在已安装的电话上加载 ADSL 功能时减少了原有的到外线的工单，不需要安装人员上门；
- 新装电话和新装 ADSL 同时进行，保留到外线的工单，安装人员上门时只需要完成外线安装、线路调试以及将 e 家终端和线路正确连接并上电，配置业务由 ITMS 远程自动完成。

3.2.2 设备由外线部门统一管理及发放

- 开通需要增加服务开通与 ITMS 交互的环节；
- 外线安装人员上门时只需要完成外线安装、线路调试以及将 e 家终端和线路正确连接并上电，配置业务由 ITMS 远程自动完成；外线在领取设备时，需要将安装的 e 家终端设备标识录入 ITMS。

3.2.3 宽带密码下发

如 radius 中的宽带密码是密文保存的，由于实现零接触，ITMS 需要知道宽带帐号及密码明文，同时考虑密码安全性，建议采用以下方式实现宽带密码下发：

- 开通工单中标示 radius 需要传明文密码给服务开通/ITMS，ITMS 完成帐号、密码下发后，在 ITMS 中删除密码明文；
-

- 如果需要再次下发账号、密码，需由用户进行更改宽带密码的操作，再通过 radius 以明文方式传给服务开通/ITMS。

3.3 单点登陆接口

3.3.1 基本流程

3.3.1.1 场景一

场景描述：用户登录客户端后，访问网上客服的时候无需再次登录即可访问网上客服网站。

在客户端上，有一个专门的“e 家客服”链接到各省的网上客服网站的按钮，当点击按钮之后，客户端按照本方案执行单点登录过程，登录成功之后在客户端显示一个“简单菜单”，“简单菜单”属于网上客服网站的页面，当验证失败的时候，需要显示一个默认的错误页面，该错误页面由网上客服网站提供并且链接。

当点击“简单菜单”上某一个链接的时候，则弹出浏览器界面全屏显示网上客服的界面。

“简单菜单”的功能界面大小需要根据客户端的大小进行设计。一般情况下的大小为：宽 $X = 234$ ，高 $Y = 342$ ，本大小是指内页的大小而非整个客户端的大小。如果高度超过 Y 则实用上下滚动条。

“简单菜单”的界面示例如下图：



3.3.1.2 场景二

场景描述：用户登录网上客服后，访问互联星空的时候无需再次登录。

客户在网上客服网站上用宽带帐号成功登录之后，当点击“e 家信息”和“e 家娱乐”按钮的时候，网上客服网站按照本方案执行单点登录过程，登录成功之后，弹出互联星空相关服务界面，为客户提供服务。

3.3.2 接口定义

3.3.2.1 接口说明

说明：

发起方：是指遵循本单点登录方案的请求者，并按照协议提供请求参数，可以是客户端、网上客服网站或者其它待扩充平台。

接收方：是指遵循本单点登录方案的接收者，并且按照流程执行登录过程后最终为客户提供服务的一方，可以是网上客服网站、互联星空网站或者其它待扩展平台。

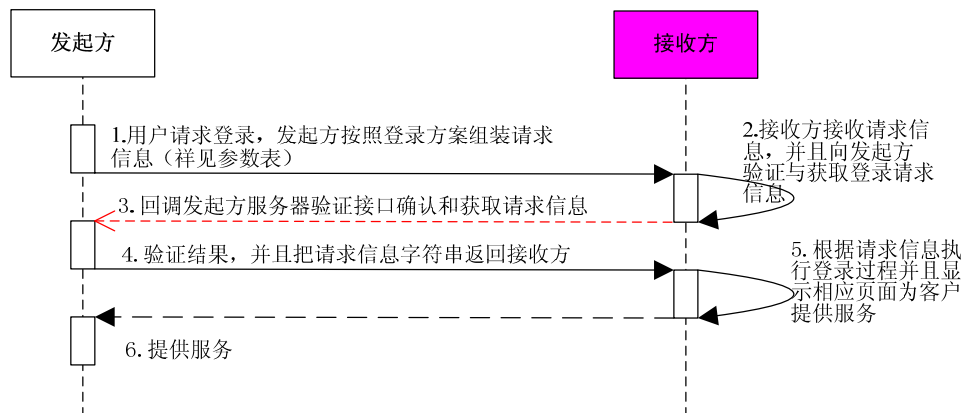
客户端、网上客服网站、互联星空网站及其它待扩充平台相互之间执行单点登录过程的对照表：

接收方 发起方	客户端	网上客服网站	互联星空网站	其它待扩充
客户端		可以	可以	待定
网上客服网站	不可以		可以	待定
互联星空网站	不可以	不可以		待定
其它待扩充	待定	待定	待定	

接收方对在短时间内来自同一个发起方的包含有同样验证条件（如宽带帐号相同）的登录认证限制次数，如客户端在短时间内用同一个宽带帐号 A 登录多次，则由接收方根据自己的配置限制在一个时间范围内（如 5 秒内），A 可以同时请求的次数为 3 次。

为便于查询登录数据和登录分析，每一个接收方需要对发起方发起的单点登录请求的历史数据进行保留。

登录流程:



流程说明:

- 用户通过发起方（客户端、网上客服网站或者其它待扩充平台）请求登录到用户已经登录所在省份的接收方（网上客服网站、互联星空网站或者其它待扩充平台），由发起方按照本登录方案发送和组装登录请求信息（验证请求登录字符串），接口请见主调接口 **CrossDomainSSO**;
- 用户所在省份的接收方（网上客服网站、互联星空网站或者其它待扩充平台）根据发起方的信息到发起方相关验证接口（见验证接口 **CrossDomainSSOCheck**）验证请求信息的合法性并且取得登录关键信息，接口验证请求信息详见参数表；如果是客户端（包括所有类型的 GUI 客户端，下同）发起则由接收方到客户端所在省份的服务器上验证。
- 发起方根据接收方传递的验证信息（验证请求登录字符串）对信息进行验证并且响应请求登录信息给接收方，同时“验证请求登录字符串”在发起方的服务器上失效，如果接收方再次发起请求含有相同“验证请求登录字符串”进行验证和获取登录请求的时候则为非法请求，当接收方在规定的时间内（自发起方发起接口时起算，具体时间由各省自定）没有调用验证的时候，则“验证请求登录字符串”也自动失效。

- 将验证结果和请求登录信息返回给接收方；
- 接收方根据请求登录信息执行登录过程并且最终跳转到相应的界面，成功则为客户提供服务，失败则提示相应的网站失败页面；如果是客户端发起的登录请求则显示按照客户端大小的一个错误页面；
- 所有验证通过之后，由接收方为客户提供相应的服务。

3.3.2.2 参数说明

3.3.2.2.1 主调接口（登录请求）

接口说明：

接口名称	跨域单点认证登录请求
接口描述	用户在发起方（客户端、网上客服网站或者其它待扩充平台）请求登录到接收方（网上客服网站、互联星空网站或者其它待扩充平台），有发起方发送“验证请求登录字符串”给接收方，接收方据此字符串获取请求登录信息
接口协议	采用 HTTP 重定向方式
接口方法	http://xxx/CrossDomainSSO?ReqStr =请求参数 xxx 网址说明： 1) 当接收方为互联星空的时候，则为互联星空网站网址，网站域名格式为“省份缩写.vnet.cn” 2) 当接收方为网上客服的时候，则为网上客服网站网址，网站域名格式为“省份缩写.ct10000.com” 3) 其它待扩充网站相应的网址另定
接口要求	1) 本接口分别在网上客服网站、互联星空网站或者其它待扩充平

	<p>台上实现，客户端不实现；</p> <p>2) 接口方法名（CrossDomainSSO）统一名称，后不加任何文件后缀名，如.aspx、.jsp 或者.php 等；</p> <p>3) 请求参数为“请求参数”表中列明的字段采用如下方式发送给对方网站：</p> <p>UrlEncoding(Base64(3DES(SysId+\$+SeqNbr+\$+TimeStamp+\$+Authenticator)))</p> <p>4) 3DES 密钥为双方协定，附件二对 3DES 加解密算法在 Java 平台和.NET 平台的实现进行举例。</p>
--	---

请求参数：

名称	说明	数据类型	长度
SysId	用于标示发起方的唯一标识，详见《附表一》	String	2
SeqNbr	“验证请求登录字符串”，此字段用于接收方向发起方发起验证登录合法性和获取请求登录信息，由发起方自动生成，某省不重复，接收方调用验证接口 CrossDomainSSOCheck 后此字符串即失效	String	20
TimeStamp	当前时间戳（24 小时制） 格式：yyyy-MM-dd HH:mm:ss	String	19
Authenticator	认证码。取值： Base64(SHA1(SysId+\$+SeqNbr+\$+TimeStamp))		
<p>注：</p> <p>■ 各字段中不包含\$字符，下同。</p>			

- SeqNbr 的用法：如当客户端要求获取某省网上客服网站认证的时候，则 SeqNbr 由客户端所在省份的服务器上维护，当客户端发起单点登录请求到网上客服网站的时候，由网上客服网站到客户端所在省份的服务器上回调验证接口（见“验证接口”）来验证客户端到网上客服网站的请求是否合法并且从客户端所在省份的服务器上获取请求信息，验证接口调用之后 SeqNbr 则在客户端所在服务器上失效（通过同样的 SeqNbr 再次进行验证不能验证成功），在网上客服网站规定的时间内没有调用验证接口时此字符串在规定的时间内失效。

3.3.2.2.2 验证接口

接口说明：

接 口 名称	接收方向发起方验证登录请求并且获取请求登录信息
接 口 描述	接收方为验证请求方的登录请求信息，向发起方验证请求信息的合法性同时获取用于在接收方进行登录的“请求登录信息”，验证通过并且获得请求登录信息之后，由接收方自行执行登录过程并且显示相应的界面为客户提供服务，让客户无需再次验证即可得到服务。
接 口 协议	采用 Web Service 方式
接 口 方法	http://xxx/CrossDomainSSOCheck xxx 网址说明： 1) 当发起方为网上客服的时候，则为网上客服网站网址，网站域名格式为“省份缩写.ct10000.com” 2) 当发起方为客户端的时候，则为客户端所在省份的服务器的域名或

	<p>者 IP 地址，域名和 IP 地址根据实际情况另定</p> <p>3) 其它待扩充网站相应的网址另定</p>
接口要求	<ol style="list-style-type: none"> 对 Web Service 的调用采用 SOAP 调用方法，承载在 HTTP 协议上； 本接口分别在客户端所在服务器、网上客服网站、互联星空网站上实现； 接口方法名（CrossDomainSSOCheck）统一名称，后不加任何文件后缀名，如.aspx、.jsp 或者.php 等； Web Service 的请求参数和应答参数都以字符串方式发送给对方以减少协议开销，字符串编码方式如下： <p>请求参数：Base64(3DES(SeqNbr+\$+TimeStamp+\$+Authenticator))</p> <p>应答参数：Base64(3DES(LoginType+\$+LoginName+\$+ProvinceCode+\$+ForwardURL+\$+TimeStamp+\$+SysId+\$+SeqNbr+\$+ResultCode+\$+StandBy01+\$+StandBy02+\$+StandBy03+\$+Authenticator))</p> 3DES 密钥为双方协定，附件二对 3DES 加解密算法在 Java 平台和.NET 平台的实现进行举例。

请求参数：

名称	说明	数据类型	长度
SeqNbr	接收方验证请求合法性并且要求获取请求登录信息的“验证请求登录字符串”，本字段是发起方通过 CrossDomainSSO 接口提交给接收方的 SeqNbr 字段	String	10

	的内容。		
TimeStamp	当前时间戳 yyyy-MM-dd HH:mm:ss	String	19
Authenticator	认证码。取值： Base64(SHA1(SeqNbr+\$+TimeStamp))		
当验证通过之后，SeqNbr 即在发起方的服务器上失效。			

应答参数：

名称	说明	数据类型	长度（字节）
LoginType	登录类型编码，详见《附表三》	String	2
LoginName	用户登录帐号，可以是宽带帐号、客户编码等	String	60
ProvinceCode	用户登录帐号对应的省份编码，见《附表二》	String	5
ForwardURL	需要访问的最终目标地址，不能为空。 ForwardURL 中不能包\$符号。全路径		
TimeStamp	当前时间戳（24 小时制） 格式：yyyy-MM-dd HH:mm:ss	String	19
SysId	用于标示发起方的唯一标识，详见《附表一》	String	2
SeqNbr	接收方发起验证的“验证请求登录串”原样响应	String	10
StandBy01	预留字段一，各省根据业务需要使用，没有用到改字段的时候为空字符串，下同	String	
StandBy02	预留字段二	String	

StandBy03	预留字段三	String	
ResultCode	调用返回结果，详细编码含义见《附表四》	String	4
Authenticator	认证码。取值： BASE64(SHA1(LoginType+\$+LoginName+\$+ProvinceCode+\$+ForwardURL+\$+TimeStamp+\$+SysId+\$+SeqNbr+\$+ResultCode+\$+StandBy01+\$+StandBy02+\$+StandBy03))	String	
注： 1.LoginName 的内容根据 LoginType 的内容不同为不同的含义，根据《附表三》中说明的登录类型编码对应不同的登录名称，如当 LoginType=02 时，则 LoginName 填写宽带帐号； 2.ForwardURL 此字段对应的 URL 指客户在发起方点击的跳转详细地址，可以对应到接收方的主页面，也可以对应到接收方内的某个功能点页面地址。			

3.3.3 附表

3.3.3.1 附表一：发起方唯一标识 SysId

编码	发起方名称
01~10	集团预留
11	我的 e 家统一客户端
12	网上客服中心网站
13	互联星空网站

14~50	待定
51~99	各省自扩展用

3.3.3.2 附表二：省份标识编码 ProvinceCode

省份码	省份名称	省份码	省份名称
01	北京	19	湖南
02	上海	20	广东
03	天津	21	广西
04	重庆	22	海南
05	河北	23	四川
06	山西	24	贵州
07	内蒙古	25	云南
08	辽宁	26	西藏
09	吉林	27	陕西
10	黑龙江	28	甘肃
11	江苏	29	青海
12	浙江	30	宁夏
13	安徽	31	新疆
14	福建	32	台湾
15	江西	33	香港
16	山东	34	澳门

17	河南	35	全国
18	湖北	36	海外

3.3.3.3 附表三：登录类型编码 LoginType

编码	登录类型
01	客户 ID 方式登录
02	宽带帐号方式登录
03~50	集团保留待定
51~99	各省自扩展用

3.3.3.4 附表四：调用返回结果 ResultCode

编码	发起方名称
0000	成功
1001	网络错误，无法连接对方服务器
1002	调用发起方的 CrossDomainSSOCheck 接口（验证接口）时，Web Service 调用失败，传输错误
1003	操作过程中出现错误
1004	加密回传参数过程中出现错误
1101	获取请求参数，加密字符串为空
1102	解密请求参数过程中发现参数数组长度不够
1103	解密请求参数过程中发现 HashCode 解密错误

1201	调用 CrossDomainSSOCheck 接口，在发起方对 Authenticator 进行校验失败，原因：接收方调用验证接口时送给发起方的 Authenticator 校验失败
1202	调用 CrossDomainSSOCheck 接口，在接收方对 Authenticator 进行校验失败，原因：接收方获得发起方返回验证结果中的 Authenticator 校验失败
1301	参数 SysId 错误
1302	参数 ProvinceCode 错误
1303	无效的 TimeStamp
1304	解密请求参数后发现 GUID 为空
1305	解密请求参数后发现 GUID 无效
9999	未知错误

4 WLAN 漫游及 e 家电话开通支持

各省 WLAN 用户与 WLAN 全国漫游认证系统的数据同步，实现 WLAN 全国漫游；

- 省 CRM/服务开通将需要同步的 WLAN 用户数据发送到集团 CRM，集团 CRM 再同步给 WLAN 全国漫游认证系统。

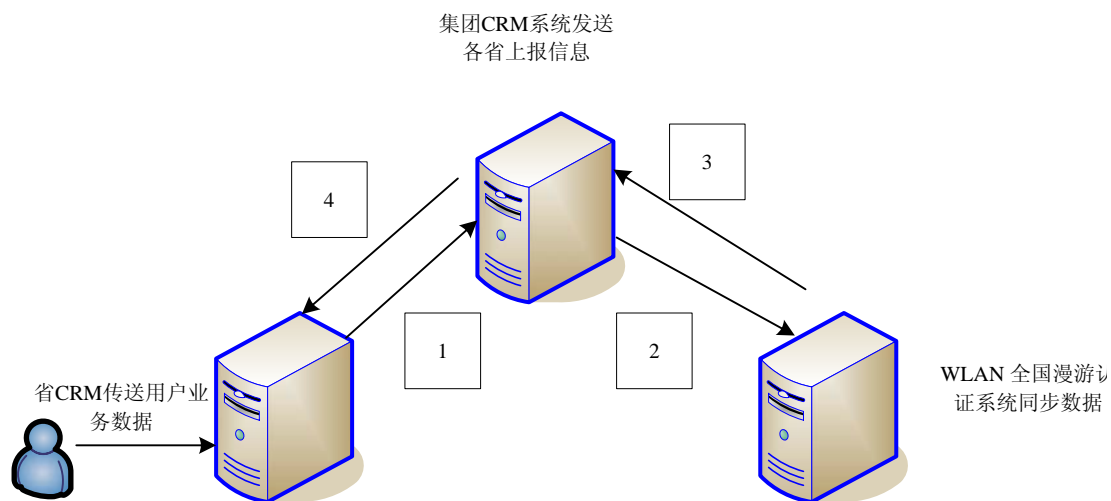
各省 e 家电话用户数据写入集团综合通信平台，实现 e 家电话业务开通。

- 省 CRM/服务开通将需要更新的 e 家电话用户数据发送到集团 CRM，集团 CRM 再同步给全国综合通信平台。

省 CRM/服务开通与集团 CRM 的接口作为 WLAN 漫游与 e 家电话开通中的一个环节。

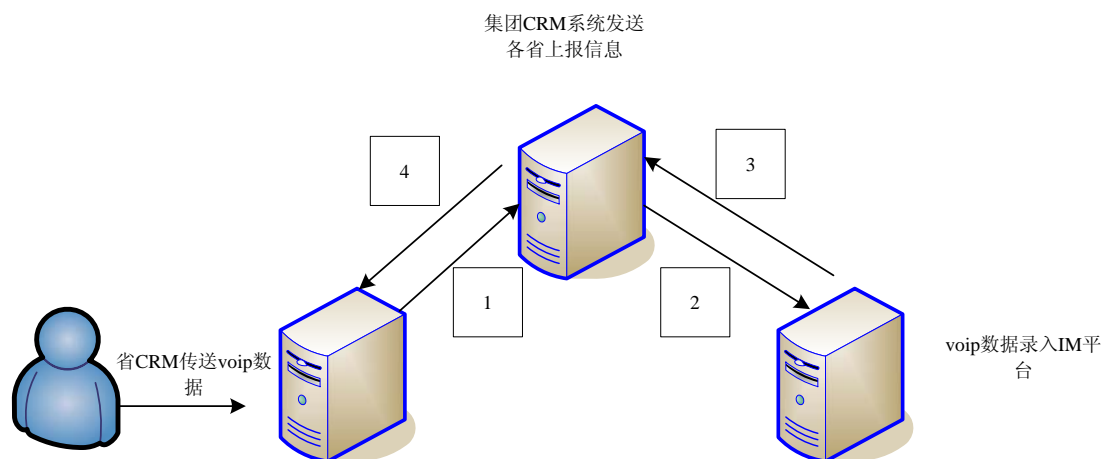
4.1 流程

4.1.1 省 CRM/服务开通同步 WLAN 业务数据流程



- 1、省 CRM/服务开通将需要开通 WLAN 漫游的业务数据发送到集团 CRM
- 2、集团 CRM 根据各省上报的信息同步给 WLAN 全国漫游认证系统
- 3、WLAN 全国漫游认证系统对数据进行同步，将操作结果返回给集团 CRM
- 4、集团 CRM 将结果返回省 CRM/服务开通

4.1.2 省 CRM/服务开通同步 e 家电话数据流程



- 1、省 CRM/服务开通将需要开通 e 家电话的业务数据发送到集团 CRM
- 2、集团 CRM 根据各省上报的 VoIP 数据同步给 IM 系统
- 3、IM 系统对数据进行同步，将操作结果返回给集团 CRM
- 4、集团 CRM 将结果返回省 CRM/服务开通

4.2 接口概述

集团 CRM 提供统一接口获取不同业务用户的数据，再根据不同业务与相应的后台系统做接口，更新用户业务数据。

4.2.1 集团 CRM 与省 CRM/服务开通系统接口

接口名称	接口说明	实现方式
GetCustOrderInfo	省 CRM 发送业务用户信息给集团 CRM 省 CRM-> 集团 CRM	实时 webservice

4.2.2 集团 CRM 与后台业务系统接口

接口名称	接口说明	实现方式
UpdateWLANInfo	集团 CRM 系统通过该接口将各省上报的信息同步给 WLAN 全国漫游认证系统 集团 CRM-〉WLAN 全国漫游认证系统	实时 webservice
UpdateVoipNumber	集团 CRM 系统通过该接口将各省上报的 e 家电话用户数据写入综合通信平台 集团 CRM-〉综合通信平台	实时 webservice

4.3 接口

4.3.1 GetCustOrderInfo（获取各省用户业务数据接口）

集团 CRM 提供的获取各省用户业务数据接口，由各省 CRM/服务开通将业务的数据发往集团 CRM 系统。

A. 接口基本信息

接口名称	接口提供方	接口使用方	接口协议
GetCustOrderInfo	集团CRM系统	省CRM/服务开通系统	Web Service

B. 输入参数

参数名称	是否可为空	类型	长度	出现次数	描述信息
InterfaceType	N	String	25	1	接口类型（参见接口类型表）
ProvinceNumber	N	Int	10	1	区号：10 表示北京， 21表示上海等
Count	N	Int	4	1	记录数目，最大值100
OrderInfo	N	Order Info_item		1..100	业务数据

OrderInfo_item

参数名称	是否可为空	类型	长度	出现次数	描述信息
AccNbr	N	String	20	1	号码, 为输入的产品业务号等
AccNbrType	N	Int	2	1	为输入的产品号码类型, 1电 话, 小灵通, 2宽带帐号, 3 Voip号码
Operator	N	Int	1	1	操作：增加1、删除0、更新2
ErrInfo	Y	String	255	1	错误信息
Property	N	Property_item		1..n	业务数据的属性

Property_item

参数名称	是否可为空	类型	长度	出现次数	描述信息
PropertyName	N	String	20	1	业务数据的属性名称
PropertyValue	Y	String	255	1	业务数据的属性值

接口类型表

取值	描述信息
----	------

UpdateWLANInfo	从省 CRM/服务开通向集团 CRM 传递 WLAN 数据
UpdateVoipNumber	从省 CRM/服务开通向集团 CRM 传递 e 家电话用户数据

当 InterfaceType 取值 UpdateWLANInfo 时, PropertyName 目前允许的值有

字段名称	是否可为空	类型	长度	描述信息	说明
CustomerID	N	String	20	客户 ID	系统内部序列号
IdentityNm	N	String	22	客户标识码	统一客户视图规范要求的
UserName	N	String	100	用户帐号	该帐号为用户输入的后缀形式: username@ 后缀
UserPassword	N	String	20	用户密码	明文保存
UserType	N	String	20	用户属性	0 我的 e 家高端用户, 1 商务领航企业用户
Business_Type	N	String	20	业务属性	0 WLAN 全国漫游
Validity	N	String	10	有效期	如果没有有效期需求, 可设置为永久有效, 默认: 3000-01-01

当 InterfaceType 取值 UpdateVoipNumber 时, PropertyName 目前允许的值有

字段名称	是否可为空	类型	长度	描述信息	说明
CustomerID	N	String	20	客户 ID	系统内部序列号
IdentityNm	Y	String	40	客户标识码	统一客户视图规范要求的
VOIPNumber	N	String	20	Voip 号码	

VOIPPassword	Y	String	250	Voip 号码对应的 密码	
ADSLAccount	N	String	40	ADSL 帐号	
ADSLType	N	String	4	绑定的宽带账号 类型，取值如下： 00: 非 e 家账号 01: e6 02: e8	
SSURL	N	String	40	软交换的认证域 名	

C. 返回数据

参数名称	是否可为空	类型	长度	出现次数	描述信息
ResultCode	N	String	1	1	是否成功(0:成功;1:失败)
ResultReason	Y	String	255	1	调用失败原因

4.3.2 UpdateWLANInfo（集团 CRM 与 WLAN 全国漫游认证系统的数据同步接口）

集团 CRM 系统通过该接口将各省上报的信息同步给 WLAN 全国漫游认证系统，进行数据的批量发送、处理。

A. 接口基本信息

接口名称	接口提供方	接口使用方	接口协议
UpdateWLANInfo	WLAN全国漫游 认证系统	集团CRM系统	Web Service

B. 输入参数

参数名称	是否可为空	类型	长度	出现次数	描述信息
InterfaceType	N	String	25	1	接口类型（参见接口类型表）
ProvinceNumber	N	Int	10	1	区号：10表示北京，21表示上海等
Count	N	String	4	1	记录数目，最大值100
OrderInfo	N	OrderInfo_item		1..100	业务数据

OrderInfo_item

参数名称	是否可为空	类型	长度	出现次数	描述信息
AccNbr	N	String	20	1	号码，为输入的产品业务号或客户编号等
AccNbrType	N	Int	2	1	为输入的产品号码类型，1电话，小灵通，2宽带帐号，3Voip号码
Operator	N	Int	1	1	操作：增加1、删除0、更新2
ErrInfo	Y	String	255	1	错误信息
Property	N	Property_item		1..n	业务数据的属性

Property_item

参数名称	是否可为空	类型	长度	出现次数	描述信息
PropertyName	N	String	20	1	业务数据的属性名称
PropertyValue	Y	String	255	1	业务数据的属性值

PropertyName 允许的值目前有

字段名称	是否可为空	类型	长度	描述信息	说明
CustomerId	N	String	20	客户 ID	
IdentityNm	N	String	22	客户标识码	
UserName	N	String	100	用户帐号	
UserPassword	N	String	20	用户密码	
UserType	N	String	20	用户属性	0 我的 e 家高端用户，1 商务领航企业用户，2 以上两者以外
Business_Type	N	String	20	业务属性	0 WLAN 全国漫游
Validity	N	String	10	有效期	如果没有有效期需求，可设置为永久有效，默认：3000-01-01

C. 返回数据

参数名称	是否可为空	类型	长度	出现次数	描述信息
OrderInfo	Y	OrderInfo_item		1..n	返回行项目（仅含流水号和错误信息），返回null表示无错误。

注：返回的 OrderInfo 行项目中，只需包含标识符字段 AccNbr 、AccNbrType 和错误信息 ErrInfo，其余字段均留空

4.3.3 UpdateVoipNumber（集团 CRM 与综合通信平台接口）

集团 CRM 系统通过该接口将各省上报的 e 家电话用户数据写入综合通信平台。

A. 输入参数

接口名称	接口提供方	接口使用方	接口协议
UpdateVoipNumber	综合通信平台	集团CRM系统	Web Service

B. 输入参数

参数名称	是否可为空	类型	长度	出现次数	描述信息
InterfaceType	N	String	25	1	接口类型（参见接口类型表）
ProvinceNumber	N	Int	10	1	区号：10 表示北京，21 表示上海等
Count	N	String	4	1	记录数目，最大值100
OrderInfo	N	OrderInfo_item		1..100	业务数据

OrderInfo_item

参数名称	是否可为空	类型	长度	出现次数	描述信息
AccNbr	N	String	20	1	号码, 为输入的产品业务号或客户编号等
AccNbrType	N	Int	2	1	为输入的产品号码类型，1 电话，小灵通，2 宽带帐号，3 Voip 号码
Operator	N	Int	1	1	操作：增加1、删除0、更新2

ErrInfo	Y	String	255	1	错误信息
Property	N	Property_item		1..n	业务数据的属性

Property_item

参数名称	是否可为空	类型	长度	出现次数	描述信息
PropertyName	N	String	20	1	业务数据的属性名称
PropertyValue	Y	String	255	1	业务数据的属性值

PropertyName 允许的值目前有

字段名称	是否可为空	类型	长度	描述信息
CustomerID	N	String	20	客户 ID
IdentityNm	N	String	22	客户标识码
VOIPNumber	N	String	20	Voip 号码
VOIPPassword	N	String	20	Voip 号码对应的密码
ADSLAccount	N	String	40	ADSL 帐号
ADSLType	N	String	4	绑定的宽带账号类型，取值如下： 00: 非 e 家账号 01: e6 02: e8
AreaCode	N	String	5	地区号
SSURL	N	String	40	软交换的认证域名

C. 返回数据

参数名称	是否可为空	类型	长度	出现次数	描述信息
ResultCode	N	String	1	1	是否成功(0:成功;1:失败)
OrderInfo	Y	OrderInfo_item		1..n	返回行项目（仅含标识号和错误信息）

注：返回的 OrderInfo 行项目中，只需包含标识符字段 AccNbr、AccNbrType 和错误信息 ErrInfo，其余字段均留空。

5 附件

5.1 附件一 终端资源管理

实现对 e 家终端设备的资源管理，包括 e 家终端设备参数配置、入库、出库、分配、调配、回收、盘点等功能。

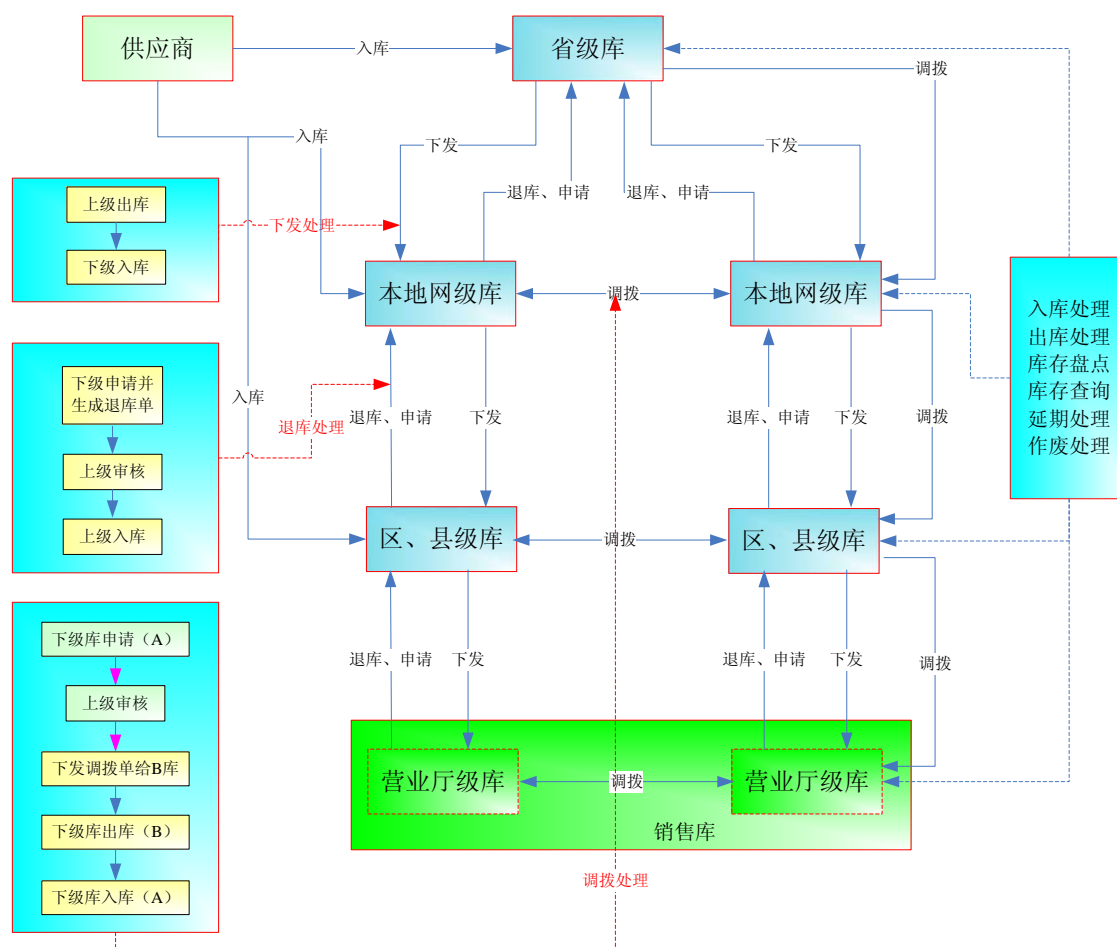
5.1.1 终端资源管理流程描述

在终端资源管理流程中，主要包括下发、退库、调拨流程的管理，对于终端资源的申请在审核批准后一般会触发一个下发或者调拨处理流程。对于调拨流程可以由一个下级库发起申请来触发，由上级审核通过后下发调拨单；也可以由上级直接下发调拨单来触发调拨处理。

对于入库处理、出库处理、库存盘点、库存查询、延期处理、作废处理主要是针对仓库中终端资源进行具体操作。

对于省级、本地网、区县级相关的资源库存管理在 MSS 中进行管理，对于销售库的管理则在 CRM 系统中管理。对于需要进行资源申购要求的资源申请，把资源申购相关信息送到 MSS 系统中，在 MSS 系统中完成采购流程。MSS 系统在完成采购后把相关的资源采购信息送到营销类资源管理系统，在营销类资源管理系统中完成入库动作。相关的资源库存管理在 MSS 中进行管理，省、市级的库存描述请参照 MSS 中的库存管理，在 MSS 系统没有建设或者在 MSS 系统中相关功能未具备的情况下，相关库存的管理可以先放在营销资源管理功能中进行管理。

终端资源的相关处理如下图所示：



注：图中红色虚线指向指对相关处理功能的流程处理，蓝色虚线指向指对具体仓库的操作功能。

5.1.2 e 家终端设备参数配置

针对 e 家终端设备的各项属性参数进行配置维护的功能，其中 e 家终端设备属性包括：e 家终端设备型号、e 家终端设备 ID、供应商、来源、状态等

[功能要求]

■ 支持针对终端各种参数进行增加、修改、删除等操作，参数包括：

- ☐ 终端型号
- ☐ 价格
- ☐ 销售类型
- ☐ 供应商
- ☐ 来源
- ☐ 等

■ 针对以上操作要根据操作员类型进行权限划分

5.1.3 e 家终端设备实物入库

e 家终端设备实物入库是指对 e 家终端设备实物入库操作, 对实物串号与计划入库串号进行比对，防止串机套机行为的发生。

■ 支持根据条件信息进行查询，条件信息包括：

- ☐ 终端类别
 - ☐ 入库批次
 - ☐ 终端型号
-

- ☐ 仓库类型
 - ☐ 入库日期
 - ☐ 等
- 支持根据批次显示该批次所有入库终端的汇总信息，包括：
- ☐ 序号
 - ☐ 终端型号
 - ☐ 终端来源
 - ☐ 数量
 - ☐ 等
- 支持对入库信息进行打印。
- 支持记录操作历史。

5.1.4 e 家终端设备实物入库撤销

e 家终端设备实物入库撤销是指对 e 家终端设备实物入库操作的撤消处理。

[功能要求]

- 支持根据各种参数条件信息进行了查询，参数信息包括：
- ☐ 终端类别
 - ☐ 入库批次
 - ☐ 终端型号
 - ☐ 仓库类型
 - ☐ 入库日期
-

☐ 等

- 支持对没有进行过下发、调拨、回收、销售、回退等业务操作的终端撤消入库。
- 支持终端入库信息按照批次进行撤消操作。
- 支持入库撤消信息的打印。
- 支持记录操作历史。

5.1.5 e 家终端设备下发

e 家终端设备下发是指将 e 家终端设备由上级下发给下级的过程。

[功能要求]

- 支持根据各种参数信息进行查询，按照批次自动显示可下发终端的电子串号等基本信息。参数信息包括：
 - ☐ 申请信息
 - ☐ 终端类别
 - ☐ 当前地区
 - ☐ 本次下发地区
 - ☐ 终端类型
 - ☐ 终端名称
 - ☐ 等
 - 支持在下发过程中形成统一的下发批次号。
 - 支持以文件和单个串号两种操作方式，进行终端的下发。
 - 能够对查询出的终端进行选择下发。
-

- 支持对终端下发的确认。
- 支持对下发汇总明细进行打印。
- 支持对下发终端的信息进行校验：
 1. 校验非法字符（空格等）。
 2. 校验串号是否存在。
- 支持记录操作历史。

5.1.6 e 家终端设备调拨

e 家终端设备调拨是指对已经下发到渠道管理部门的且未被使用的 e 家终端设备进行调配的操作。从一个部门调配到另外部门的操作，是由上一级机构生成调拨单，地市之间调拨由省分生成调拨单，区县之间调拨由地市生成调拨单，营业厅之间调拨由区县生成调拨单，由被调拨单位进行实物调出，调入单位校验后进行实物入库。

- 支持根据参数条件信息进行调拨操作，参数信息包括：
 - ☐ 终端类型
 - ☐ 终端型号
 - ☐ 品牌
 - ☐ 所属地
 - ☐ 等
 - 支持显示被调拨的终端列表信息，包括：
 - ☐ 终端类别
 - ☐ 终端串号
-

- ☐ 源库
- ☐ 目标库
- ☐ 终端型号
- ☐ 日期
- ☐ 数量
- ☐ 等

■ 支持批量调拨功能。

■ 支持对调拨的终端信息进行校验：

1. 校验非法字符（空格等）。
2. 校验串号是否存在。
3. 校验串号是否处于调出状态。

■ 支持记录操作历史。

5.1.7 e 家终端设备回收

e 家终端设备回收是指对下发的 e 家终端设备进行回收处理。

■ 支持根据各种参数信息进行查询。参数信息包括：

- ☐ 终端类别
 - ☐ 终端类型
 - ☐ 终端原库地址
 - ☐ 目标地址
 - ☐ 等
-

- 支持根据查询出的可回收终端信息列表进行选择回收。
- 支持根据操作权限进行终端回收。
- 支持打印回收汇总清单。
- 支持记录操作历史。

5.1.8 e 家终端设备退库

e 家终端设备退库是指将准备退库的 e 家终端设备状态改为已退库，以便退回厂家后处理。

- 支持根据参数条件查询需要回退的终端，参数信息包括：
 - ☐ 终端类别
 - ☐ 终端类型
 - ☐ 地区
 - ☐ 回退日期
 - ☐ 等
 - 支持回退终端的相关信息的显示，包括：
 - ☐ 当前仓库
 - ☐ 回退日期
 - ☐ 终端类型
 - ☐ 电子串号
 - ☐ 回退方式
 - ☐ 回退方
-

☐ 等

- 支持裸机回退，代理商回退，营业厅回退三种回退方式。
- 支持记录操作历史。

5.1.9 e 家终端设备盘点

e 家终端设备盘点是指对 e 家终端设备的信息进行盘点。通过定期盘点对 e 家终端设备库存进行监控，有效的把 e 家终端设备资源管理起来，降低库存，同时对 e 家终端设备销售情况进行有效的监控。

- 支持通过选择地区、终端类型、终端来源查询出按照终端归属仓库，终端来源，终端类型进行分类统计的非备用终端和备用终端发放、在库、出库信息。包括：

- ☐ 文本入库终端电子串号数
 - ☐ 上级下放终端数量
 - ☐ 回收下级终端数量
 - ☐ 回退转在库终端数量
 - ☐ 回退厂家终端数量
 - ☐ 代理商非买断终端数量
 - ☐ 营业回退终端数量
 - ☐ 营业系统销售终端数量
 - ☐ 备用终端转在库量
 - ☐ 备用终端在库量
 - ☐ 代理商买断终端数量
 - ☐ 维修终端数量
-

- ☐ 等

- 支持通过查询条件查询仓库盘点信息。参数条件包括：

- ☐ 地市库

- ☐ 终端类别

- ☐ 终端来源

- ☐ 终端类型

- ☐ 等

- 支持记录操作历史。

5.1.10 定制 e 家终端设备信息管理

定制 e 家终端设备关键信息指定制 e 家终端设备符合中国电信制定的标准，在外观、专用键、菜单呈现、接口及通用要求等方面的信息。

定制 e 家终端设备关键信息管理指对各种定制 e 家终端设备机型定制信息的管理。

- 提供定制信息从定制终端产品库导入的功能。

- 提供定制信息的发布、查询。定制信息包括：

- ☐ 机型

- ☐ 定制版本

- ☐ 外观

- ☐ 开关机

- ☐ 界面

- ❑ 专用键
- ❑ 菜单呈现
- ❑ 通用要求
- ❑ 等

5.2 附件二：加解密算法

5.2.1 概要说明

考虑到各个不同平台的开发架构和模式不相同，本章节主要从目前主流的两个开发架构平台进行说明，具体的加解密算法的例子。

Java 与 .NET 不同平台联调的时候需要注意：

1. 在 Java 中解密从 .NET 返回的 3DES 时需注意：.NET 返回的结果是字节数组，需要将其转换为字符串，这时需要注意转换的时候把字符串转为 UTF-8

2. 在 .NET 平台下，SHA1 的结果都是大写，因此在 Java 平台下需要将结果转换为大写。

5.2.2 Java 下 3DES 调用例子

```
package com.doone.basefun;

import java.io.IOException;

import javax.crypto.*;

import javax.crypto.spec.SecretKeySpec;
```

```
import sun.misc.BASE64Decoder;

import sun.misc.BASE64Encoder;


/**
 * @说明 字符串 DESede(3DES) 加密、解密
 *
 * @version v0.0.1
 *
 * @author
 *
 * @创建时间 2005-12-26 12:00
 *
 *
 * @修改历史
 *
 * <li>版本号 修改日期 修改人 影响面 修改说明
 *
 * <li>v0.0.2 2005-10
 *
 * <li>v0.0.3 2005-12-26
 */
public class EThreeDES{

    //定义加密算法，在 Java 下使用 DESede
    private static final String Algorithm = "DESede";


    /**
     * 对输入的字串进行 3DES 加密
```

```
* @param keybyte[]      加密密钥，长度为 24 字节
* @param src[]          要加密的字串
*/

public byte[] encrypt3DES(byte[] keybyte, byte[] src) {

    try {

        SecretKey deskey = new SecretKeySpec(keybyte,
Algorithm);

        Cipher c1 = Cipher.getInstance(Algorithm);

        c1.init(Cipher.ENCRYPT_MODE, deskey);

        return c1.doFinal(src);

    } catch (java.security.NoSuchAlgorithmException e1) {

        e1.printStackTrace();

    } catch (javax.crypto.NoSuchPaddingException e2) {

        e2.printStackTrace();

    } catch (java.lang.Exception e3) {

        e3.printStackTrace();

    }

    return null;

}

/**
```

```
* 对输入的字串进行 3DES 解密

* @param keybyte[]    加密密钥，长度为 24 字节

* @param src[]        要解密的字串

*/

public byte[] decrypt3DES(byte[] keybyte, byte[] src) {

    try {

        SecretKey deskey = new SecretKeySpec(keybyte,
Algorithm);

        Cipher c1 = Cipher.getInstance(Algorithm);

        c1.init(Cipher.DECRYPT_MODE, deskey);

        return c1.doFinal(src);

    } catch (java.security.NoSuchAlgorithmException e1) {

        e1.printStackTrace();

    } catch (javax.crypto.NoSuchPaddingException e2) {

        e2.printStackTrace();

    } catch (java.lang.Exception e3) {

        e3.printStackTrace();

    }

    return null;

}
```

```
/**
 * 转为 base64 编码
 * @param btParm[]      要转换的字串
 */
public String encodeBase64(byte[] btParm) throws IOException {
    BASE64Encoder encoder=new BASE64Encoder();
    String sParm = encoder.encodeBuffer(btParm);
    return sParm;
}
```

```
/**
 * 反转 base64 编码
 * @param sParm    要反转的字符串
 */
public byte[] decodeBase64(String sParm) throws IOException {
    BASE64Decoder decoder=new BASE64Decoder();
    byte[] btParm = decoder.decodeBuffer(sParm);
    return btParm;
}
```

```
public static void main(String[] args) throws IOException
```

```
{  
  
    //解密例子，加密类推  
  
    EThreeDES des = new EThreeDES();  
  
    String                                after_3desbase64                                =  
    "E7kk0pvhKq7lB2lK98zXG+1YYPz+dxSCK2076U4g3BcUTrBI0TvEETk2aSHf6tWCw0CRI8Q6  
    pABlBI3lclm9vdrmMuSyNNh4Yelfq0xJLZ+MqYeWWE80inE/8bUy5cU0o7hNxLolffQ=";  
  
    byte[] btBy = des.decodeBase64(after_3desbase64);  
  
    System.out.println("解密 BASE64 后的字符串:" + new String(btBy));  
  
    byte[] srcBytes = des.decrypt3DES("XXXXXXXXXXXXXXXX".getBytes(), btBy);  
  
    System.out.println("解密 3DES 后的字符串:" + new String(srcBytes, "utf-  
8"));  
  
    }  
  
}
```

5.2.3.NET 下 3DES 调用例子

```
    //Authenticator 生成  
  
    string                                Authenticator                                =  
    StringHelper.Base64Encode(StringHelper.SHA1Encode(LoginName + "$" +  
    ProvinceCode + "$" + ForwardURL + "$" + TimeStamp + "$" + SysId + "$" +  
    SeqNbr));  
  
    //加密应答参数  
  
    Encryptor encrypt = new Encryptor(EncryptionAlgorithm.TripleDes);
```

```
byte[] IV =
Encoding.UTF8.GetBytes(ConfigurationManager.AppSettings["3DesIV"]);

encrypt.IV = IV;

byte[] key =
Encoding.UTF8.GetBytes(ConfigurationManager.AppSettings["3DesKey"]);

string ResultStrSource = LoginName + "$" + ProvinceCode + "$" +
CityCode + "$" + ForwardURL + "$" + TimeStamp + "$" + SysId + "$" +
SeqNbr + "$" + Authenticator;

byte[] SourceData = Encoding.UTF8.GetBytes(ResultStrSource);

string ResultStr =
Convert.ToBase64String(encrypt.Encrypt(SourceData, key));
```

=====

Encryptor.cs

```
using System;
using System.Security.Cryptography;
using System.IO;

namespace Common.Security
{
    public class Encryptor
    {
        private EncryptTransformer transformer;
        private byte[] initVec;
        private byte[] encKey;
        public Encryptor(EncryptionAlgorithm algId)
        {
```

```
        transformer = new EncryptTransformer(algid);
    }
    public byte[] IV
    {
        get
        {
            return initVec;
        }
        set
        {
            initVec = value;
        }
    }
    public byte[] Key
    {
        get
        {
            return encKey;
        }
    }
    public byte[] Encrypt(byte[] bytesData, byte[] bytesKey)
    {
        //Set up the stream that will hold the encrypted data.
        MemoryStream memStreamEncryptedData = new
MemoryStream();
        transformer.IV = initVec;
        ICryptoTransform transform =
transformer.GetCryptoServiceProvider(bytesKey);
        CryptoStream encStream = new
CryptoStream(memStreamEncryptedData,
            transform,
            CryptoStreamMode.Write);
        try
        {
            //Encrypt the data, write it to the memory stream.
```

```
        encStream.Write(bytesData, 0, bytesData.Length);
    }
    catch(Exception ex)
    {
        throw new Exception("Error while writing encrypted
data to the stream: \n"
        + ex.Message);
    }
    //Set the IV and key for the client to retrieve
    encKey = transformer.Key;
    initVec = transformer.IV;
    encStream.FlushFinalBlock();
    encStream.Close();
    //Send the data back.
    return memStreamEncryptedData.ToArray();
} //end Encrypt
}
}
```

=====

Decryptor.cs ()

```
using System;
using System.Security.Cryptography;
using System.IO;
```

```
namespace Common.Security
{
    public class Decryptor
    {
        private DecryptTransformer transformer;
        private byte[] initVec;
        public byte[] IV
        {
```

```
        set
        {
            initVec = value;
        }
    }

    public Decryptor(EncryptionAlgorithm algId)
    {
        transformer = new DecryptTransformer(algId);
    }

    public byte[] Decrypt(byte[] bytesData, byte[] bytesKey)
    {
        //Set up the memory stream for the decrypted data.
        MemoryStream memStreamDecryptedData = new
MemoryStream();
        //Pass in the initialization vector.
        transformer.IV = initVec;
        ICryptoTransform transform =
transformer.GetCryptoServiceProvider(bytesKey);
        CryptoStream decStream = new
CryptoStream(memStreamDecryptedData,
            transform,
            CryptoStreamMode.Write);
        try
        {
            decStream.Write(bytesData, 0, bytesData.Length);
        }
        catch(Exception ex)
        {
            throw new Exception("Error while writing encrypted
data to the stream: \n"
                + ex.Message);
        }
        decStream.FlushFinalBlock();
        decStream.Close();
        // Send the data back.
        return memStreamDecryptedData.ToArray();
    }
}
```

```
        } //end Decrypt
    }
}
```

```
=====
===
```

DecryptTransformer.cs

```
using System;
using System.Security.Cryptography;

namespace Common.Security
{
    public class DecryptTransformer
    {
        private EncryptionAlgorithm algorithmID;
        private byte[] initVec;
        internal byte[] IV
        {
            set
            {
                initVec = value;
            }
        }
        internal DecryptTransformer(EncryptionAlgorithm deCryptId)
        {
            algorithmID = deCryptId;
        }
        internal ICryptoTransform GetCryptoServiceProvider(byte[]
bytesKey)
        {
            // Pick the provider.
            switch (algorithmID)
            {
                case EncryptionAlgorithm.Des:
```

```
        {
            DES des = new DESCryptoServiceProvider();
            des.Mode = CipherMode.CBC;
            des.Key = bytesKey;
            des.IV = initVec;
            return des.CreateDecryptor();
        }
        case EncryptionAlgorithm.TripleDes:
        {
            TripleDES des3 = new
TripleDESCryptoServiceProvider();
            des3.Mode = CipherMode.ECB;
            des3.Padding = PaddingMode.PKCS7;
            //des3.Mode = CipherMode.CBC;
            return des3.CreateDecryptor(bytesKey, initVec);
        }
        case EncryptionAlgorithm.Rc2:
        {
            RC2 rc2 = new RC2CryptoServiceProvider();
            rc2.Mode = CipherMode.CBC;
            return rc2.CreateDecryptor(bytesKey, initVec);
        }
        case EncryptionAlgorithm.Rijndael:
        {
            Rijndael rijndael = new RijndaelManaged();
            rijndael.Mode = CipherMode.CBC;
            return rijndael.CreateDecryptor(bytesKey,
initVec);
        }
        default:
        {
            throw new CryptographicException("Algorithm
ID '" + algorithmID +
                "' not supported.");
        }
    }
```

```
    }  
    } //end GetCryptoServiceProvider  
}  
}
```

=====
===
EncryptTransformer.cs

```
using System;  
using System.Security.Cryptography;  
  
namespace Common.Security  
{  
    public enum EncryptionAlgorithm {Des = 1, Rc2, Rijndael, TripleDes};  
    public class EncryptTransformer  
    {  
        private EncryptionAlgorithm algorithmID;  
        private byte[] initVec;  
        private byte[] encKey;  
        internal EncryptTransformer(EncryptionAlgorithm algId)  
        {  
            //Save the algorithm being used.  
            algorithmID = algId;  
        }  
        internal byte[] IV  
        {  
            get  
            {  
                return initVec;  
            }  
            set  
            {  
                initVec = value;  
            }  
        }  
    }  
}
```

```
        }
    }
    internal byte[] Key
    {
        get
        {
            return encKey;
        }
    }
    internal ICryptoTransform GetCryptoServiceProvider(byte[]
bytesKey)
    {
        // Pick the provider.
        switch (algorithmID)
        {
            case EncryptionAlgorithm.Des:
            {
                DES des = new DESCryptoServiceProvider();
                des.Mode = CipherMode.CBC;
                // See if a key was provided
                if (null == bytesKey)
                {
                    encKey = des.Key;
                }
                else
                {
                    des.Key = bytesKey;
                    encKey = des.Key;
                }
                // See if the client provided an initialization
vector
                if (null == initVec)
                { // Have the algorithm create one
                    initVec = des.IV;
                }
            }
        }
    }
}
```

```
        else
        { //No, give it to the algorithm
            des.IV = initVec;
        }
        return des.CreateEncryptor();
    }
    case EncryptionAlgorithm.TripleDes:
    {
        TripleDES des3 = new
TripleDESCryptoServiceProvider();
        //des3.Mode = CipherMode.CBC;
        des3.Mode = CipherMode.ECB;
        des3.Padding = PaddingMode.PKCS7;
        // See if a key was provided
        if (null == bytesKey)
        {
            encKey = des3.Key;
        }
        else
        {
            des3.Key = bytesKey;
            encKey = des3.Key;
        }
        // See if the client provided an IV
        if (null == initVec)
        { //Yes, have the alg create one
            initVec = des3.IV;
        }
        else
        { //No, give it to the alg.
            des3.IV = initVec;
        }
        return des3.CreateEncryptor();
    }
    case EncryptionAlgorithm.Rc2:
```

```
{  
    RC2 rc2 = new RC2CryptoServiceProvider();  
    rc2.Mode = CipherMode.CBC;  
    // Test to see if a key was provided  
    if (null == bytesKey)  
    {  
        encKey = rc2.Key;  
    }  
    else  
    {  
        rc2.Key = bytesKey;  
        encKey = rc2.Key;  
    }  
    // See if the client provided an IV  
    if (null == initVec)  
    { //Yes, have the alg create one  
        initVec = rc2.IV;  
    }  
    else  
    { //No, give it to the alg.  
        rc2.IV = initVec;  
    }  
    return rc2.CreateEncryptor();  
}  
case EncryptionAlgorithm.Rijndael:  
{  
    Rijndael rijndael = new RijndaelManaged();  
    rijndael.Mode = CipherMode.CBC;  
    // Test to see if a key was provided  
    if (null == bytesKey)  
    {  
        encKey = rijndael.Key;  
    }  
    else
```

```
        {
            rijndael.Key = bytesKey;
            encKey = rijndael.Key;
        }
        // See if the client provided an IV
        if(null == initVec)
        { //Yes, have the alg create one
            initVec = rijndael.IV;
        }
        else
        { //No, give it to the alg.
            rijndael.IV = initVec;
        }
        return rijndael.CreateEncryptor();
    }
    default:
    {
        throw new CryptographicException("Algorithm
ID " + algorithmID +
        " not supported.");
    }
}
}
}
}
```

=====

SymmetricCryptographer.cs

using System;

namespace Common.Security

{

```
public class SymmetricCryptographer
{
    private EncryptionAlgorithm algorithm;
    private byte[] initVec;
    private byte[] encKey;
    public byte[] IV
    {
        get
        {
            return initVec;
        }
        set
        {
            initVec = value;
        }
    }
    public byte[] Key
    {
        get
        {
            return encKey;
        }
        set
        {
            encKey = value;
        }
    }
    public SymmetricCryptographer(EncryptionAlgorithm algorithm)
    {
        this.algorithm = algorithm;
    }
    public byte[] Encrypt(byte[] plainData)
    {
        return Encrypt(plainData, Key);
    }
}
```

```
    }  
    public byte[] Encrypt(byte[] plainData, byte[] key)  
    {  
        Encryptor encryptor = new Encryptor(algorithm);  
        encryptor.IV = IV;  
        return encryptor.Encrypt(plainData, key);  
    }  
    public byte[] Decrypt(byte[] cipherData)  
    {  
        return Decrypt(cipherData, Key);  
    }  
    public byte[] Decrypt(byte[] cipherData, byte[] key)  
    {  
        Decryptor decryptor = new Decryptor(algorithm);  
        decryptor.IV = IV;  
        return decryptor.Decrypt(cipherData, key);  
    }  
}  
}
```

主题词:电信 信息化 转型业务支撑 通知

抄 送: 中国电信股份有限公司广州研究院

印发单位: 中国电信集团公司综合部 2007 年 11 月 27 日印发
