|  |
| --- |
| f |



中国南方电网有限责任公司

终端安全管控平台项目

建设方案

楚雄供电局

2017 年 6月

批 准：

审 核：

校 核：

编 写：

目 录

[1 概述 1](#_Toc4634)

[1.1 背景 1](#_Toc3226)

[1.2 项目依据 3](#_Toc25174)

[1.3 目标 3](#_Toc29788)

[1.4 术语 4](#_Toc7110)

[2 现状 5](#_Toc1925)

[2.1 国内发展状况及产品介绍 5](#_Toc28417)

[2.2 现状分析 5](#_Toc1613)

[3 项目必要性 6](#_Toc11610)

[3.1 桌面终端规范管理的需要 6](#_Toc11986)

[3.2 桌面终端安全管控的需要 7](#_Toc20032)

[3.3 降低桌面终端运维管理及管控成本的需要 7](#_Toc11620)

[4 设计思路和原则 7](#_Toc7386)

[5 终端安全管控平台技术架构设计 8](#_Toc6211)

[5.1 整体架构 8](#_Toc14165)

[5.1.1 IT基础架构 9](#_Toc6701)

[5.1.2 硬件、支撑软件设计 10](#_Toc5554)

[5.1.3 应用软件架构 11](#_Toc19047)

[5.1.4 客户端架构设计 12](#_Toc15401)

[5.1.5 后台服务架构设计 14](#_Toc26164)

[5.2 系统部署策略 15](#_Toc7499)

[5.2.1 客户端部署策略 15](#_Toc32467)

[5.2.2 接入层部署策略 16](#_Toc4745)

[5.2.3 应用服务器层部署策略 16](#_Toc2157)

[5.2.4 存储层部署策略 17](#_Toc17883)

[5.3 环境设计 17](#_Toc22958)

[5.3.1 开发环境 17](#_Toc32142)

[5.3.2 测试环境 17](#_Toc26835)

[5.3.3 生产环境 18](#_Toc897)

[5.4 建设以及运维体系 18](#_Toc14963)

[5.5 平台运维体系 18](#_Toc27485)

[5.5.1 平台运维流程 20](#_Toc22352)

[5.5.2 运维规范基础建设 21](#_Toc22417)

[5.5.3 运维队伍建设 21](#_Toc2546)

[5.6 关键技术处理机制 22](#_Toc6929)

[5.6.1 静态信息管理 22](#_Toc27101)

[5.6.2 动态信息管理 22](#_Toc17965)

[5.6.3 运维管理 22](#_Toc10648)

[6 技术选型 22](#_Toc24310)

[6.1 选型原则 23](#_Toc1514)

[6.1.1 客户端选型原则 23](#_Toc18965)

[6.1.2 后端选型原则 23](#_Toc26609)

[6.2 客户端选型 23](#_Toc20216)

[6.2.1 客户端开发构建工具选型 23](#_Toc6400)

[6.2.2 客户端开发网络协议库选型 24](#_Toc15039)

[6.2.3 客户端开发语言选型 24](#_Toc7787)

[6.2.4 客户端开发GUI框架选型 24](#_Toc32603)

[6.3 服务器端选型 26](#_Toc17573)

[6.3.1 服务器端接入层选型 26](#_Toc22699)

[6.3.2 服务器端应用服务器选型 26](#_Toc23645)

[6.3.3 服务器端存储层选型 27](#_Toc4012)

[6.3.4 服务器端开发语言选型 27](#_Toc384)

[6.3.5 服务端消息队列中间件选型 28](#_Toc29681)

[7 未来省级集中建设策略展望 30](#_Toc17223)

[8 整体计划 30](#_Toc9719)

[8.1 前期准备阶段 31](#_Toc32189)

[8.2 系统建设阶段 32](#_Toc24860)

# 概述

## 背景

*本小节将从国家政策、南网信息化战略、南网内网安全管控现状等方面描述楚雄供电局终端安全管控平台建设所处的大背景，论证安全管控建设的重要性和必要性。*

近年来，信息安全形势日益严峻。一是来自境内外敌对势力的入侵、攻击、破坏越来越严重。二是针对基础信息网络和重要信息系统的违法犯罪持续上升。不法分子利用一些安全漏洞，使用病毒、木马、网络钓鱼等技术进行网络盗窃、诈骗、黑客攻击等违法犯罪活动，利用木马和“僵尸网络”实施分布式拒绝服务攻击的威胁加剧，防范难度越来越大。

随着云南电网公司不断提高风险防范意识，建立健全了网络与信息安全体系，网络端和服务器端的安全防护得以有效提升。然而，针对桌面终端的安全管理上还有很大的不足，目前楚雄供电局在桌面终端管理上，从资产管理方面楚雄供电局通过南方电网的资产管理系统和云南电网的ITSM系统进行管理，这两种资产管理方式目前还是传统的台账式管理模式，桌面终端的硬件配置信息复杂，更换频繁，台账式管理模式，很难实现资产的准确统计，难以实现桌面终端的责任人、使用人动态管理。通过华为的NAC完成终端的网络准入和设备管理，采用赛门铁克安全软件和金山安全软件实现病毒查杀、漏洞修复等基础安全防护，桌面终端设备的运维管理、应用运维管理则通过委托第三方进行现场运维，而设备运维也仅限于桌面终端的硬件可用性运维，应用运维也仅限于保证用户登录业务系统的可用性，对桌面终端安全管理、和合规性管理等方面尚存在诸多空白。通过2016年楚雄供电局针对部分桌面终端的软硬件配置及运行数据进行了采集、统计分析，发现终端运行存在大量安全问题，其中安装游戏类软件370多台，开放共享目录的终端有808台；有247款程序在侦听端口和提供网络服务。在桌面终端的配置上、安装软件的规范性、运行软件的规范性，系统合规性方面存在很多潜在隐患，信息安全风险较大，目前缺乏有效的技术管控手段。

另外，当前的桌面终端运维服务主要分为三个方面，一个是硬件检修，桌面终端属于精密电子产品，且大多开机时间较长，工作环境等因素不可避免的导致桌面终端出现这样或那样的硬件问题，出现问题后轻则影响桌面终端的使用效果，重则发生系统不能启动或数据丢失；第二是软件环境运维，目前桌面终端使用的应用软件越来越多，CSGII相应业务系统的正常运行都需要对本地环境进行相应的设置，大多业务系统应用故障都是由于系统环境设置不正确或各类插件异常导致，这些业务系统环境问题对一线员工来说难度较大，需要相应技术人员现场处理；第三是操作系统环境运维，网省公司针对桌面终端安全管理的相关要求，都需要通过现场服务，对操作系统及相关应用软件进行相应配置或打补丁来实现。

总体来讲，硬件维修方面，供电局购置的桌面终端大多为国内外大厂产品，产品质量有保障，售后服务支持较好，硬件故障所占比例已逐年减少。但是软件环境和操作系统相关的故障以及信息安全配置方面的故障，却呈快速增长状态，由于当前只能通过第三方服务工程师进行人工现场运维，效率低、成本高。随着南方电网CSGII的推广实施，一线员工的工作越来越依赖信息系统，桌面终端的工作状况直接影响到一线员工的工作质量与工作效率，为保证桌面终端良好的工作状态，楚雄供电局每年投入大量的人力物力，针对桌面终端提供维修服务，由于桌面终端分布在楚雄供电局辖区内的1市9县，地域跨度大，当终端出现故障后，运维响应较慢，严重影响业务工作开展。

另外，在十六届四中全会通过的《关于加强党的执政能力的决议》中，将信息安全与政治安全、经济安全、文化安全并列为四大主题之一，网络与信息安全已成为国家安全的重要组成部分。中共中央办公厅、国务院办公厅印发《2006-2020年国家信息化发展战略》指出“到2020年，国家信息安全保障水平大幅提高”和“全面加强国家信息安全保障体系，大力增强国家信息安全保障能力”，是我国信息化发展的战略目标和战略重点。

2007年，公安部、国家保密局、国家密码管理局、国务院信息化工作办公室共同发布了《信息安全等级保护管理办法》（公通字 [2007]43号）和《关于开展全国重要信息系统安全等级保护定级工作的通知》（公信安[2007]861号），确立了安全等级保护作为国家信息安全的基本制度，并明确指出“铁路、银行、海关、税务、民航、电力、证券、保险等重要行业率先开展信息系统的安全保护等级工作”。

国资发[2007]8号文《关于加强中央企业信息化工作的指导意见》指出“推进中央企业信息化，是我国信息化发展战略的重要内容，是促进企业发展方式根本性转变的必然选择” 。并要求“把信息安全放在企业信息化的重要位置，要正确处理好加快信息化发展与保障安全的关系。对处于国家重要行业中的中央企业，如军工企业、电力行业等，尤其要强调信息安全”。国家电力监督管理委员会依据《电力安全生产监管办法》（电监会2号令），专门成立电力行业网络与信息安全领导小组，统一领导电力行业网络与信息安全监督管理工作，组织落实国家基础信息网络和重要信息系统安全保障工作的方针、政策和各项重大部署，并要求“努力推进电力行业网络与信息安全工作。加快推进电力行业网络与信息安全法规体系及标准规范建设，建立电力行业信息安全等级保护制度，确保重要时段电力行业网络与信息安全”。

针对上述现状，楚雄供电局迫切需要开发一套针对桌面终端的安全管控及自动化运维的技术支持平台，通过技术手段解决楚雄供电局桌面终端安全管控和软件环境运维方面存在的问题，在降低管理难度的同时，提升桌面终端的规范化、标准化管理水平。

## 项目依据

1. 《2006-2020年国家信息化发展战略》（中办发[2006]11号）
2. 《关于加强中央企业信息化工作的指导意见》(国资发[2007]8号)
3. 《终端计算机系统安全等级技术要求》（GA/T671）
4. 《中国南方电网有限责任公司信息安全保障体系》
5. 《中国南方电网有限责任公司发展战略纲要》
6. 《南方电网公司“十三五”信息化规划》
7. 《南方电网公司面向服务的信息技术构架（SOA）框架规范》
8. 《南方电网公司管理信息系统安全等级保护标准》
9. 《南方电网公司信息化项目管理办法》
10. 《南方电网公司信息系统建设管理办法》
11. 《南方电网公司信息安全保障体系》
12. 《南方电网公司信息化工作管理规定》
13. 《南方电网公司信息化项目管理办法》

## 目标

*本小节结合未来电力行业、电力业务、信息技术等方面的发展趋势明确南网楚雄供电局终端安全管控平台的建设目标。*

当今信息技术发展日新月异，全球互联网正在向下一代升级，云计算、虚拟化，物联网、微服务等方兴未艾，为电网企业信息化发展提供了新的动力。电力行业作为传统的能源型工业，更应密切关注信息技术发展新趋势、新挑战，结合行业实际情况积极把握新技术带来的新机遇，深入研究，超前谋划，形成新的核心竞争力，掌握未来发展的主动权。

南方电网公司楚雄供电局终端安全管控平台的主要任务是配合信息部门建立一体化的公司软硬件管理控制平台中心，实现南网企业内网软硬件安全管控标准的固化。更加有利于企业内网软硬件资产安全的有效，便捷的管理，并为企业节省管控的成本。

## 术语

*本小节解释本方案中应用到的术语。如SOA、Qt等*

|  |  |
| --- | --- |
| **术语** | **解释** |
| **SOA** | 即面向服务的体系结构(Service-Oriented Architecture)，是一个组件模型，它将应用程序的不同功能单元(称为服务)通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种这样的系统中的服务可以一种统一和通用的方式进行交互。 |
| **Boost** | [Boost库](http://baike.baidu.com/item/Boost%E5%BA%93" \t "http://baike.baidu.com/_blank)是一个可移植、提供[源代码](http://baike.baidu.com/item/%E6%BA%90%E4%BB%A3%E7%A0%81" \t "http://baike.baidu.com/_blank)的C++库，作为标准库的后备，是C++标准库的补充。 Boost库由C++标准委员会库工作组成员发起，在C++社区中影响甚大。Boost由于其对跨平台的强调，对标准C++的强调，与编写平台无关。Boost包含很多模块，能极大的提高开发效率，比如日志，单元测试，asio网络模块等。 |
| **Qt** | Qt是一个1991年由奇趣科技开发的跨平台[C++](http://baike.baidu.com/item/C++" \t "http://baike.baidu.com/_blank)[图形用户界面](http://baike.baidu.com/item/%E5%9B%BE%E5%BD%A2%E7%94%A8%E6%88%B7%E7%95%8C%E9%9D%A2" \t "http://baike.baidu.com/_blank)应用程序开发框架。现今已发展得非常庞大，该库久经工业考验，无论性能和功能都是C++第三方库和框架中最好的，最著名的框架之一。用其开发的大型软件有：WPS，Autodesk Maya，KDE，Skype客户端。 |
| **ActiveMQ** | ActiveMQ 是Apache出品，最流行的，能力强劲的开源消息队列中间件。企业级应用多用它。支持JMS标准和J2EE规范。 |
| **JMS** | JMS即[Java消息服务](http://baike.baidu.com/item/Java%E6%B6%88%E6%81%AF%E6%9C%8D%E5%8A%A1" \t "http://baike.baidu.com/_blank)（Java Message Service）应用程序接口，是一个[Java平台](http://baike.baidu.com/item/Java%E5%B9%B3%E5%8F%B0" \t "http://baike.baidu.com/_blank)中关于面向[消息中间件](http://baike.baidu.com/item/%E6%B6%88%E6%81%AF%E4%B8%AD%E9%97%B4%E4%BB%B6" \t "http://baike.baidu.com/_blank)（MOM）的[API](http://baike.baidu.com/item/API/10154" \t "http://baike.baidu.com/_blank)，用于在两个应用程序之间，或[分布式系统](http://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E7%B3%BB%E7%BB%9F" \t "http://baike.baidu.com/_blank)中发送消息，进行异步通信。Java消息服务是一个与具体平台无关的API，绝大多数MOM提供商都对JMS提供支持。JMS是一种与厂商无关的 API，用来访问消息收发系统消息，它类似于[JDBC](http://baike.baidu.com/item/JDBC" \t "http://baike.baidu.com/_blank)(Java Database Connectivity)。 |
| **Libcurl** | libcurl是一个纯C语言的库，主要功能就是用不同的协议连接和沟通不同的服务器, libcurl当前支持http, https, ftp, gopher, telnet, dict, file, 和ldap 协议。libcurl同样支持HTTPS证书授权，HTTP POST, HTTP PUT, FTP 上传。 |
| **Json** | [JSON](http://baike.baidu.com/item/JSON" \t "http://baike.baidu.com/_blank)([JavaScript](http://baike.baidu.com/item/JavaScript" \t "http://baike.baidu.com/_blank) Object Notation, JS 对象标记) 是一种轻量级的数据交换格式。它基于 [ECMAScript](http://baike.baidu.com/item/ECMAScript" \t "http://baike.baidu.com/_blank) (w3c制定的js规范)的一个子集，采用完全独立于编程语言的文本格式来存储和表示数据。简洁和清晰的层次结构使得 JSON 成为理想的数据交换语言。 易于人阅读和编写，同时也易于机器解析和生成，并有效地提升网络传输效率。是互联网企业在基于http的通信中XML的代替品，更加容易阅读和解析，其解析框架一般都是高性能。 |

# 现状

## 国内发展状况及产品介绍

随着互联网的发展，传统的信息安全形式也越来越严峻，各种网络安全事件层出不穷，上至国家下至企业都对信息安全提出了更高的要求，通过多年的建设，云南电网公司对业务系统、核心网络设备、数据中心机房等信息安全做了大量的工作，取得了良好的效果，但针对分布范围更广，用户数量更大，使用场景更复杂的桌面终端安全管控却存在许多不足。

华为的AnyOffice，云南电网的网络准入就是采用的AnyOffice通过和防火墙配合实现NAC，AnyOffice主要核心思想是从身份和设备可识别（Identity）、数据不泄密（Privacy）和设备可管理（Compliance）三个方面帮助企业全面应对挑战，为企业提供业界最广泛的安全性和最简单易用的管理方案。云南电网的AnyOffice从应用上更多关注的是网络准入和设备管理。

360天擎终端安全管系统是360面向政府、企业、金融、军队、医疗、教育、制造业等大型企事业单位推出的集防病毒与终端安全管控于一体的解决方案。360天擎终端安全管理系统，以大数据技术为支撑、以可靠服务为保障，它能够为用户精确检测已知病毒木马、未知恶意代码，有效防御APT攻击，并提供终端资产管理、漏洞补丁管理、安全运维管控、网络安全准入、移动存储管理、终端安全审计、XP盾甲防护诸多功能。360天擎更多关注的是桌面终端的安全。

北信源VRV EDP终端安全管理产品针对网络管理工作中遇到的实际管理需求，进行网络安全资源规划,如防止移动设备非法接入内部网络、IP资源分配管理、静态IP同MAC地址的绑定、提供设备资源登记及硬件设备（硬盘、CPU、内存等）变化报警、进行内部网络连接阻断等功能。同时，为有效解决网络中计算机非法接入和非法外联问题， VRV EDP系统提供实时监控的强大功能，即实时报警以及实时切断非法计算机同内网的连接。北信源终端安全管理产品主要侧重于网络安全管理。

## 现状分析

近年来，信息安全形势日益严峻。一是来自境内外敌对势力的入侵、攻击、破坏越来越严重。二是针对基础信息网络和重要信息系统的违法犯罪持续上升。不法分子利用一些安全漏洞，使用病毒、木马、网络钓鱼等技术进行网络盗窃、诈骗、黑客攻击等违法犯罪活动，利用木马和“僵尸网络”实施分布式拒绝服务攻击的威胁加剧，防范难度越来越大。

近年来，云南电网公司不断提高风险防范意识，建立健全了网络与信息安全管理制度，然而，针对桌面终端的安全管理上还有很大的不足，目前楚雄供电局在桌面终端管理上，从资产管理方面楚雄供电局通过南方电网的资产管理系统和云南电网的ITSM系统进行管理，这两种资产管理方式目前还是传统的台账式管理模式，桌面终端的硬件配置信息复杂，更换频繁，台账式管理模式，很难实现资产的准确统计，难以实现桌面终端的责任人、使用人动态管理。通过华为的AnyOffice完成终端的网络准入和设备管理，采用赛门铁克安全软件和金山安全软件实现病毒查杀、漏洞修复等基础安全防护，桌面终端设备的运维管理、应用运维管理则通过委托第三方进行现场运维，而设备运维也仅限于桌面终端的硬件运维，应用运维也仅限于保证用户在用业务系统的可用性，对应用的安全管理、桌面终端的合规性管理等方面缺乏全面有效的技术手段，存在较大的安全隐患。桌面终端的运维工作都需要运维人员现场运维，运维效率低，成本高。

现有基础：

* 网络情况：目前楚雄供电局及下属县公司都在云南电网公司内网之内，实现了省级内网覆盖，随着2013年综合数据网IP网络改造、2013年局域网升级与优化等项目的实施，供电局的网络向万兆的升级，骨干层网络已实现万兆平台的升级，各县公司网络的接入带宽也升级到千兆。
* 设备情况：目前楚雄供电局信息中心机房已经简称虚拟化平台，可根据项目具体需要配置相应的计算资源和存储空间。
* 现有信息系统建设及应用情况：目前楚雄供电局准入系统采用华为的AnyOffice，安全软件局本部采用赛门铁克，县公司采用金山安全软件，部分用户还存在自主安装其它安全类软件的情况。
* 存在的问题分析：目前楚雄供电局在桌面终端的管控上，存在资产配置情况不清晰，资产责任人和使用人不明确，应用管理缺失，信息安全隐患大，桌面安全类软件功能重叠或相互独立，难以实现桌面终端的统一安全管理，桌面终端运维难度大、效率低，缺乏自动化运维或故障诊断的能力等问题。

# 项目必要性

## 桌面终端规范管理的需要

楚雄供电局目前对桌面终端的管理缺少相应的规范，在桌面终端的资产管理上通过南网的资产管理系统和云网的ITSM系统进行管理，但这两套系统都是台帐类管理模式，需要通过人工录入，难以跟踪桌面终端的配置变化，资产台帐的准确性差；在桌面终端的责任人和使用人关联上也缺乏相应的技术手段，传统的台帐式管理难以及时跟踪桌面终端的责任人及使用人的变化，久而久之供电局内部的桌面终端资产台帐和实物帐物不符现象严重；在对桌面终端的应用管理上缺乏相应的规范，桌面终端的应用安装及运行无管控，存在较大的安全隐患；通过桌面终端安全管控平台的建设，通过技术手段精准获取桌面终端配置信息和应用软件信息，通过对采集的相关数据进行分析，制定相应的桌面终端管理规范，最终实现桌面终端的规范化管理。

## 桌面终端安全管控的需要

南方电网公司对信息安全合规性管控越来越严格，楚雄供电局针对桌面终端的合规性管控目前缺少相应的技术手段，从合规性检查及不合规项目的整改上大多需要手工进行，要及时进行桌面终端的信息安全合规性检查和修复，并跟踪整改结果，如果没有信息系统的支撑是难以实现的，通过桌面终端安全管控平台的建设可实时对桌面终端的合规性进行检测，根据安全管控的设置可实现不合规项的报警或自动修复。

## 降低桌面终端运维管理及管控成本的需要

楚雄供电局的桌面终端分布在全州70余个乡镇，在用桌面终端的数量在3000台左右，目前的桌面终端运维采用现场运维的模式，桌面终端的运维及管控效率低，成本高，通过桌面终端安全管控平台的建设，可实时监控和获取桌面终端的运行情况，通过应用运行策略和安全管控策略可实现桌面终端的远程监控和相关项目的自动运维，可大幅降低桌面终端的运维及管控成本，提升运维及管控效率。

# 设计思路和原则

*本章节结合电力业务和信息技术的发展现状和未来趋势，描述终端安全管控平台设计的思路和需要遵循的技术原则。*

楚雄供电局桌面终端安全管控平台充分考虑南方电网和云南电网对桌面终端资产管理和信息安全合规性管控的相关要求，并结合楚雄供电局桌面终端运维管理的相关要求，建设一套符合楚雄供电局桌面终端运维管理和信息安全管控的信息系统，实现楚雄供电局桌面终端全面安全管控。

为了保证平台的可靠运行，需要具备很强的高可用性、高性能、可扩展性和易维护性，还要充分考虑未来与其他系统进行数据的同步、交换，所以需要具备集成性和先进性，同时需要保证系统的安全性。

桌面终端安全管控平台建设以价值为导向，以战略为驱动，以能力为基础。在设计中，既考虑桌面终端管理的最新理论与最佳实践，又充分考虑南方电网信息安全建设与管理能力和现状。桌面终端安全管控平台设计需遵循如下的原则：

1. 遵循信息化规划

桌面终端安全管控平台要全面贯彻落实科学发展观，以公司战略为导向，以创新为载体，引入先进的经营理念，在桌面终端的运维及安全管控上按照“统一管理、统一规划、统一标准、统一建设（四统一）”原则和定制化开发的研发建设思路，建设符合南网IT资产的管理要求、满足南方电网的信息安全合规性、适应楚雄供电局桌面终端运维管理需求的桌面终端安全管控平台。

1. 战略承接性

开展桌面终端安全管控平台建设，是为了为全面落实南方电网和云南电网对桌面终端的管控要求，提升桌面终端的信息安全防护能力，降低桌面终端的安全风险。桌面终端安全管控平台建设必须紧密承接公司战略思路，推动战略落地。

1. 重点突出性

在借鉴国内外内网安全管控系统和终端运维系统建设最佳实践的基础上，充分理解和吸收南方电网、云南电网及楚雄供电局在桌面终端管理上的最新理念与精髓，抓住楚雄供电局当前桌面终端资产准确性和信息安全管控的主要矛盾，予以重点解决。

1. 功能实用性

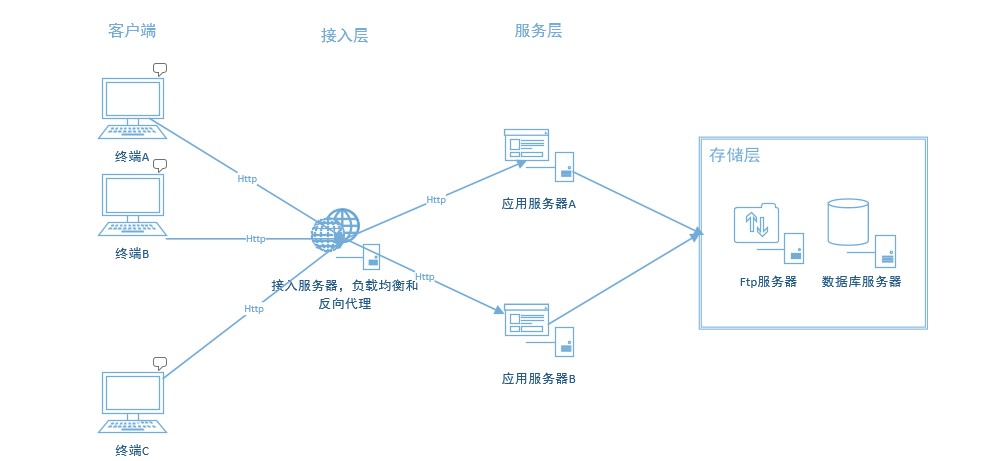
桌面终端安全管控平台的功能设计，将紧密贴合楚雄供电局针对桌面终端的安全管控及运维需求，防止出现华而不实，追求简单实用，体现公司管理精益化“精、简、益、善”的精髓。

# 终端安全管控平台技术架构设计

## 整体架构

*本小节描述营销信息一体化系统的整体架构，清晰管理前端展现、后台实现、客户端实现、数据库、访问接口、底层支持协议等逻辑层次划分结构。*

### IT基础架构



终端安全管控平台是楚雄电网内网资产的管控系统。由于楚雄下属的组织机构公司布面积广、各机构软硬件环境形式多样，该平台可靠性质量要求高，要尽可能减少客户端请求的排队。因此对系统的并发处理性能，负载均衡能力，网络架构都提出了较高的要求。

**客户端：** 管控平台数据采集，以及安全等管控功能的最终执行端，具体的管控功能可以通过其设计的插件系统无缝扩展，所有终端均会访问接入层。

**接入层：**因为部署的终端数量多，分布广，需要设计该层为请求入口，做后端做负载均衡处理，分散服务器压力。同时该层也负责反向代理服务，简而言之就是把特定的终端请求的响应又返回给特定的终端。

**服务层：**后台服务层结合消息队列中间件，采用后端开发效率最高的Java语言开发管控业务服务组件，在服务层完成面各种管控业务服务。对于管理员用户验证、权限、消息审批流等其他应用系统共有的功能服务，则可采用通用的开源组件，大大减少开发难度。

**存储层：**目前大型国有企业内部应用系统采用的数据系统主要为DB2和Oracle两种产品，中小型的企业内部可能采用SQLServer、MySQL等。因此数据存储层主要是对Oracle的数据库Java访问接口（JDBC）做一定的封装，然后采用开源的数据库连接池，在减小数据库服务器压力的同时也便于上层服务采用统一的方式进行最高效的数据库访问。

### 硬件、支撑软件设计

*本小节将根据终端安全管控平台的特点和性能要求，明确硬件、支撑软件及网络等配套设施需要满足的性能指标。*

明确系统所需的软、硬件平台资源、物理架构图，硬件资源主要包括计算资源、存储资源、网络资源，软件资源主要包括数据库、中间件。计算、存储资源应根据《中国南方电网有限责任公司软硬件平台架构和资源分配指导意见》进行估算。

所有设备可采用数据中心的虚拟计算资源，采用两台虚拟机作为接入服务器其中一台开启软负载均衡，两台虚拟机作为应用服务器，所有服务部署在两台应用服务器上，数据库采用数据中心的Oracle或MySQL数据库，使用数据中心存储。软硬件资源需求如下：

* 接入服务器（Nginx负载均衡服务器，反向代理服务器）：2台，CPU\*1\*2Core，内存≥8GB，硬盘≥146GB，操作系统CentOS7。
* 应用服务器（Tomcat服务器）：2台，CPU\*1\*2Core，内存≥8GB，硬盘≥500GB，操作系统CentOS7。
* 数据库服务器（Oracle服务器）：1台，CPU\*1\*4Core，内存≥32GB，硬盘≥2TB，操作系统CentOS7。
* 文件服务器（MongoDB服务器）：1台，CPU\*1\*2Core，内存≥8GB，硬盘≥2TB，操作系统CentOS7。

因为本平台是动态更新的系统，系统客户端程序及相关组件可根据需要随时进行动态更新（手动或自动），系统需要一个放置系统客户端组件的MongoDB服务器，客户端动态更新的内容由此服务器进行下载，由于FTP服务器不进行业务处理，只作为文件下载的功能使用，对系统性能要求不高，但功能发布频繁时，瓶颈就会转移到网络IO上了，所以MongoDB服务器可能存在网络流量较大的情况，为此MongoDB服务器建议采用中低端刀片服务器，但FTP服务器建议采用单独的存储设备，以保证具有较高的IO并发能力及数据安全。

最后对于客户端层面的软件设备，本平台只暂时支持windows平台（windows 7 32/64位及其以上），应用层设备可以对最新客户端的有较好的展现效果，当采用跨平台展现方式时，应用层设备系统环境则无特别要求。

客户端的开发，为保证安全管控平台客户端在整体性能上满足要求，高可靠性，稳定性,便捷性要求，客户端开发采用C/C++进行实现其核心处理功能，在客户端前端可采用Windows平台下的展现方式和非Windows平台展现方式两种；为保证系统开发规范性和开发的高效、有序，应在系统建设时采用统一的开发平台，以从根本上保证终端安全管控平台的规范性，也可极大的提高系统建设速度及成功率；为此需要根据终端安全管控平台建设方案，进行相应开发平台的研发工作。

### 应用软件架构

*本小节将根据终端安全管控平台建设要求，明确管控平台系统的软件架构。*

应用软件架构客户端和后台服务层两大部分，具体如下图：

**客户端应用层**

Http

Http AMQP MQTT

**远程协助**

**消息推送**

**查**

**询**

**统**

**计**

**终端策略维护**

**文件推送**

**应用运行管理**

**设备运行管理**

**资产历史信息**

**资产基本信息**

**运**

**维**

**管**

**理**

**运**

**行**

**管**

**理**

**资**

**产**

**管**

**理**

**ActiveMQ 消息推送**

**应用安全管理**

**系统安全管理**

**网络安全管理**

**信息安全管理**

**终**

**端**

**安**

**全**

**权限管理**

**人员管理**

**组织机构管理**

**系**

**统**

**管**

**理**

**业**

**务**

**逻**

**辑**

**Nginx Http负载均衡，反向代理**

**MongoDB文件服务**

### 客户端架构设计

终端安全管控平台的业务可扩展性决定了客户端的可扩展性，要想达到系统对应用层的要求，客户端架构设计应考虑以下因素：

**01**

**02**

**04**

**03**

具有良好的护交性和护交速度

符合用户操作习惯，具有优秀的用户体验

具有高可靠性，遵循Fail-safe原理

支持插件模式，可按需扩展客户端功能

基于以上因素，我们设计客户端架构如下图：

**插件运行容器**

**操作系统原生API，或第三方开源库（Qt，Libcurl等）**

**各种功能插件（设备采集插件等）**

**更新模块**

**客**

**户**

**端**

**守**

**护**

**程**

**序**

**客**

**户**

**端**

**运**

**行**

**平**

**台**

1. **守护进程**

守护进程(daemon)是一类在后台运行的特殊进程，用于执行特定的系统任务。该守护进程是客户端高可靠的保证，遵循Fail-Safe原理，一直不间断的监视主程序运行状况，主程序容器一旦意外终止，它会重新启动主程序。以windows 服务的形式提供。

1. **插件运行容器**

其实相当于一个插件系统，它实现没有任何业务上的功能，只对插件的装载和卸载做实现，该模块采用微内核的方式，实现尽可能小。以exe可执行文件的形式提供。

1. **插件模块**

根据需求，实现各种各样的插件，插件需要遵守一定的接口编写规范，才能正确被插件运行容器装载和卸载，一般以DLL动态链接库的形式提供。

1. **更新模块**

实现客户端插件，以及其他组件的更新任务，该模块是独立了，几乎不受其他模块的影响，一旦运行，就连接至服务器把最新版本的插件，组件下载至客户端，并重启客户端，完成插件的安装。

1. **本地操作接口**

本地操作接口是操作系统提供的任何API，还有利用第三方开源组件/库的API的集合，使程序开发方便，提高开发效率。

#### 客户端架构约束

上述架构设计的目的是将系统开发中工作量最大的客户端开发通过插件的形式定制出各种客户端功能或界面，要想使客户端展现架构的能力最大化，真正做到无限制扩展且可靠，那么对整体架构的划分要有一定的约束，即各模块之间的职责要分工明确，插件运行主程序，不能有任何的业务逻辑存在。

### 后台服务架构设计

后台服务架构是整个系统运行的核心和关键，面对众多的不同技术水平的应用开发维护人员，使用的框架必须满足如下要求：从开发维护角度看其必须易开发、易部署、易维护。从运营角度看必须是安全可靠的、可不间断运行的、支持大型企业级应用、易监控、易扩展。同时为使得系统平稳过渡其必须能够很好地利用现有软件资源，便于现有软件资源的快速整合复用与重构，支持常见中间件服务的调用。

必须是安全可靠而高效的, 可不间断运行的.不应因为具体应用服务代码的缺陷而导致系统崩溃。

支撑大型应用软件，支持高压力的多并发请求，支持线程和进程模式并可配置。系统的承受能力应该取决于机器的配置乃至操作系统的限制，而架构本身不应有任何限制。应该能通过配置的方式灵活地适应各种不同的情况。

所以JEECG是一款基于代码生成器的J2EE快速开发平台，开源界“小普元”超越传统商业企业级开发平台。引领新的开发模式(Online Coding模式(自定义表单) - > 代码生成器模式 - > 手工MERGE智能开发)， 可以帮助解决Java项目60%的重复工作，让开发更多关注业务逻辑。既能快速提高开发效率，帮助公司节省人力成本，同时又不失灵活性。具备：表单配置能力（无需编码）、移动配置能力、工作流配置能力、报表配置能力（支持移动端）、插件开发能力（可插拔）

1. **可插拔的系统**

所谓可热拔插的系统是指可以在系统运行的状况下给它动态的增加新的功能或者卸载不需要的功能。插件开发，业务功能组件以插件方式集成平台，也可以单独部署发发布，有力支撑了SAAS云应用系统需求

1. **可动态配置的系统**

软件系统的运行环境日益复杂,复杂性越来越难以用传统的方式控制。

JEECG的支持一下可动态配置的特性：

* Online Coding在线开发(通过在线配置实现一个表模型的增删改查功能，无需一行代码，支持用户自定义表单布局)。
* 多数据源（在线配置数据源，数据源工作类封装）。
* 在线配置报表（无需编码，通过在线配置方式，实现曲线图，柱状图，数据等报表）
* 动态报表功能（用户输入一个sql，系统自动解析生成报表）
* 代码生成器，支持多种数据模型,根据表生成对应的Entity,Service,Dao,Action,JSP等,增删改查功能生成直接使用
* 在线流程定义，采用开源Activiti流程引擎，实现在线画流程,自定义表单,表单挂接,业务流转，流程监控，流程跟踪，流程委托等
* 自定义表单,支持用户自定义表单布局，支持单表、列表、Select\Radio\Checkbox\PopUP\Date等特殊控件
* 数据权限（精细化数据权限控制，控制到行级，列表级，表单字段级，实现不同人看不同数据，不同人对同一个页面操作不同字段）

1. **稳定、高效的系统**

使用Java语言开发的服务器后端服务一般非常稳定高效，这得益于日益增长的JVM虚拟机技术（JIT），和Java工业开发体系框架，使得Java越来越能担当重任。

## 系统部署策略

*本小节将根据该管控平台的特点和性能要求，明确系统的部署方式和部署策略。*

### 客户端部署策略

客户端软件通过AnyOffice推送到所有在用桌面终端静默安装，分发在楚雄市下各县级单位，通过在桌面终端安装桌面安全终端并注册自启动服务，客户端软件在推送前需要经过第三方安全测试，如只有通过360开放软件平台的测试，在推送安装过程中才能够不被拦截，保证推送的成功率。

**楚雄供电局总局（AnyOffice管理端）**

**南华供电局（anyoffice客户端）**

**元谋供电局（anyoffice客户端）**

**禄丰供电局（anyoffice客户端）**

### 接入层部署策略

桌面终端安全管控平台接入层采用供电局集中部署，服务器部署在楚雄供电局信息中心机房，保证辖区内所有桌面终端都可以访问，接入层的负载均衡服务器软件采用业界最高性能的Nginx，在接入层的Linux服务器上，由开发人员安装Ngnix软件，并通过配置文件为其配置负载均衡和反向代理的功能，转发http请求并分散到各个后端tomcat应用服务器。

### 应用服务器层部署策略

桌面终端安全管控平台应用服务层采用供电局集中部署，服务器部署在楚雄供电局信息中心机房，保证辖区内所有桌面终端都可以访问，被Nginx正常转发，应用服务器采用业界最流行的tomcat J2EE容器，完全支持Servlet 3.0规范。由开发人员安装tomcat并部署在2台或多台Linux服务器上，并把相应含有应用业务逻辑的war包部署在tomcat软件webapp的目录之下。

消息服务器属于应用服务器层级，它用来支撑服务器主动发送消息和文件的推送通知，采用大型企业内网常用的Active MQ消息队列中间件，支持消息事务。该软件由开发人员按照官方配置部署到一台Linux服务器上，并不建议与tomcat安装在同一个服务器，因为需要把压力集中在MQ Server上。

### 存储层部署策略

桌面终端安全管控平台应用存储层采用供电局集中部署，服务器部署在楚雄供电局信息中心机房，保证辖区内所有桌面终端都可以访问，被tomcat应用服务器正常访问，存储数据库采用国有大型企业最常用的Oracle。由开发人员安装并部署在1台Linux服务器上。

存储层的部署还包括部署Ftp文件服务器，用来支撑App Store以及其他文件的存储服务器，Ftp服务器软件，并无特别要求。

## 环境设计

*本小节描述终端安全管控系统需要必备的各种软件运行环境及这些运行环境的重要性与必要性。由于终端安全管控平台设参与下属的组织机构多，地理分布面广，对系统的各个功能需要从开发、测试到发布上线的所有环节进行管控，并将这些环节分隔开，在不同的运行环境中检验代码实现的运行效果，以确保平台运行的稳定性和高效性。*

终端安全管控平台是一个具有大并发，并且具有高可靠性的系统，该平台需要考虑24\*7的不间断运行，该系统的客户端组件功能不可避免的要时常进行更新，为了保障系统能够实现24\*7的目标，又要保系统的功能更新，需要在新功能正式发布至生产环境前进行充份的测试和验证，保证在生产环境发布一次成功。为此营销信息一体化系统必须具备一套完整的从开发至测试，从模拟至生产的多个流程，并对所有环境的工作内容及各环境间的工作流程进行清晰和明确。

### 开发环境

终端安全管控平台需要具备一个开发环境，所有终端安全管控平台的新功能开发及删除都在此环境上进行，此环境应具备完善的版本管理及分支管理，以保证不同版本功能的bug修正及不同版本间的同步或恢复操作。

### 测试环境

终端安全管控平台需要搭建一个测试环境，所有新研发功能都需要发布至此服务器进行功能及性能测试，以保证新研发功能符合设计要求，测试环境包含功能测试环境和压力测试，功能测试环境对硬件设备性能要求不高，所以用本地的开发机器就可以胜任测试环境，但是该环境不具备同生产环境一致的硬件架构模式。为了节约供电局的硬件资源压力测试就暂时不重新分配一个与生产环境的一致的硬件及软件平台了，直接在生产环境下进行压力测试，更能真实的反映系统的最大吞吐率及负载极限。

### 生产环境

生产环境是直接承载终端安全管控系统所有数据和操作的实时运行平台，对系统稳定性要求高和性能不需要追求很高，生产环境上所有要进行发布的功能，都必须经过测试环境的验证后，方可发布至生产环境。生产环境应按照系统实际承载量进行合理计算并按照满配进行硬件配置部署，以保证生产系统的性能并应对突发事件。

## 建设以及运维体系

*本小节将梳理终端安全管控平台的运维服务体系。要确保终端安全管控平台的顺利上线实施，同时上线后能够及时响应客户需求，需要建立快速响应的运维机制，并提供小而精的运维服务。本文构想终端安全管控平台的建设运维系统主要包含平台运维体系一个方面的主要内容：*

*平台运维体系：包括平台运维流程的建立、服务体系建设、运维队伍建设、运维规范制度建设、自动监控等方面的内容；*

## 平台运维体系

平台运维体系需要采用多级服务模式，可随着企业承受的经济成本的能力，逐步扩张。最开始由于考虑成本支出，只需要采用二级服务模式支撑就能满足系统的运维需要。

* **一线（现场运维团队）初步支持**

开发公司成立专门的市级运维团队（1-2人），通过在线服务等即时通讯工具方式受理和记录服务请求，调配服务人员，任务分派，根据服务的基本流程，跟踪服务请求处理的完成情况并及时回访。

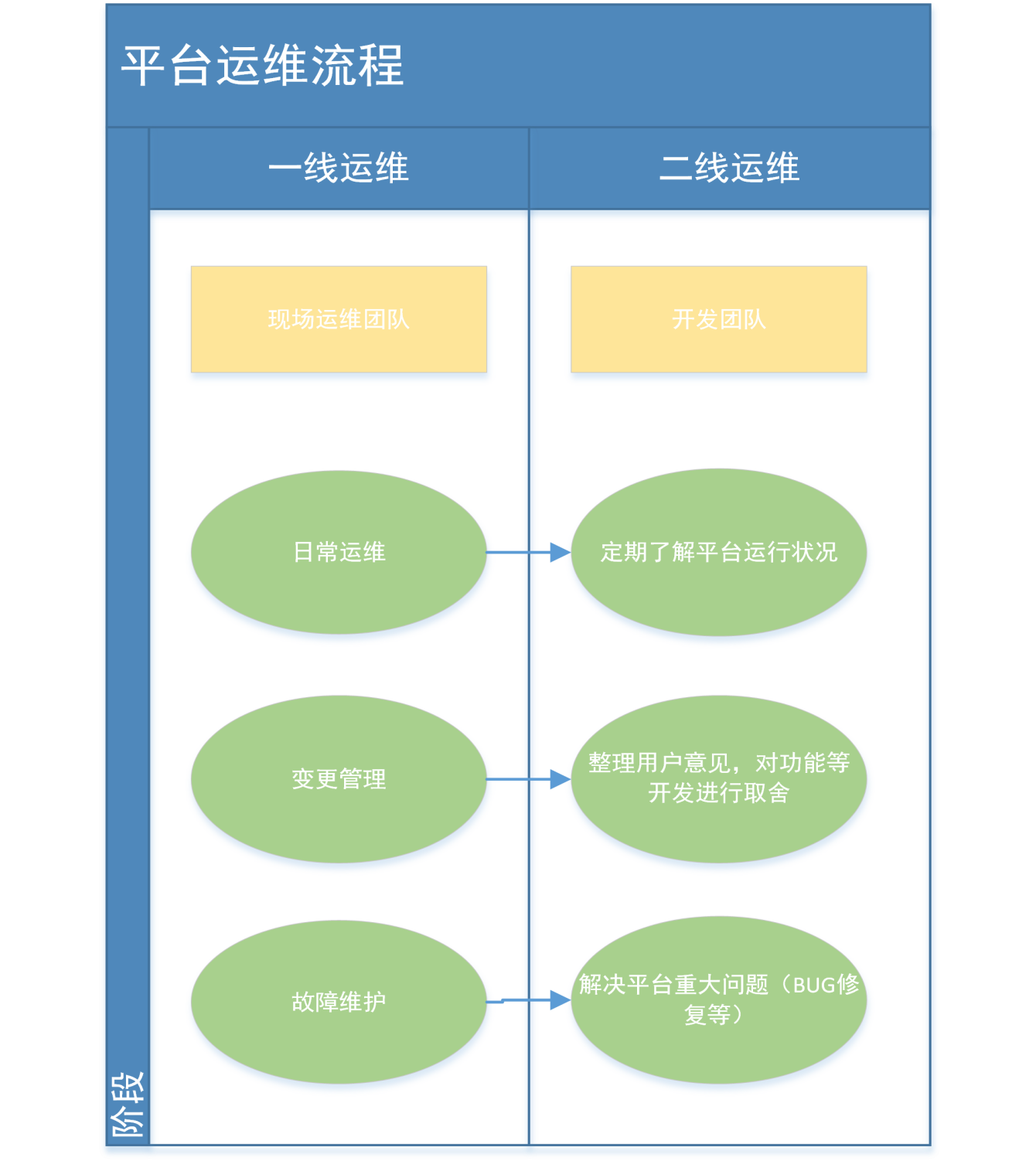
运维团队负责现场软硬件巡检、及时处理和反映突发事件。主要考核处理问题的时间效率，以及及时反馈；

在一线服务请求不能解决的重大问题情况下，二线（开发团队）迅速相应并开始问题解决过程(可根据实际需要，提供电话支持、现场支持、即时通讯工具等多种方式进行问题解决。同时，二线承担系统长期优化和系统消缺等工作。

* **二线（研发团队）开发支持**

对一线、不能现场解决的重大问题进行处理，主要考核处理问题的质量和数量；对客户新增业务需求组织开发，完成功能的实现，对现有系统提供代码级的维护支持。

### 平台运维流程



* **日常运维**

编写平台运维大纲、系统巡检手册等文档。固化日常运维人员的输入输出，使不同的运维人员有统一的运维流程。

* **变更管理**

在长期的运维过程中，运维人员根据系统的特点提出的优化建议、对服务器进行故障的消缺、实施组提出业务方面的需求等等。通过变更管理流程，可以对这些问题进行控制及监督。各种问题将以Word或Excel文件的形式整理汇总，由运维负责人根据问题分类及分发，运维人员记录整个问题的处理解决过程。通过变更管理流程，不但可以对问题进行监督，还能对运维处理情况进行考核。

* **故障管理**

在运维过程中遇到的重大故障，影响了平台的正常运行，需要按照故障管理流程进行处理。发现故障以后，需要依在尽可能短的时间内对系统进行恢复。现场运维人员负责记录故障处理的整个过程，形成故障报告，便于进行查阅及学习。

另外，在新系统上线时，要形成系统安装报告、配置文档、操作手册、测试报告等详细的文档，以便新系统加入后可以更快速熟悉并监控起来；使用第三方的开源系统监控工具对各生产系统进行辅助监控，加强监控的力度。

通过一整套的运维流程，可以使运维更加规范化、更加完善。

### 运维规范基础建设

为了保证系统安全、可靠运行，应重视相关规范技能的建设，要求做到每一步操作都有章可循，将误操作概率降到最低：

* 《Linux shell 编程与服务器管理》----内容包括Linux基本的Linux Bash编写，基本操作等。
* 《Linux系统管理技术手册(第二版)》------ 学习linux的基础实用知识

### 运维队伍建设

* 针对平台运行维护各项指标要求，安排具有初级专业技能的技术人员，为了节约成本，对IT相关证书不做要求。
* 通过专业人员建的通密合作，有力保障系统的长期运维和优化工作的开展。

## 关键技术处理机制

*本小节将结合终端安全管控平台的建设要求，讲述建设过程中在技术方面可能遇到的问题及相应的处理机制。*

覆盖面广、使用者多、影响范围大，系统性能是确保系统的高可用性、高效性、稳定性等关键指标的核心，不同的功能，会对系统性能造成不同程度的影响，因此必须针对系统不同的功能，在技术上采用不同的处理机制，以保证系统的高效、稳定，而不会因为对某功能的处理不当造成对系统整体性能的影响。

终端安全管控平台处理的功能主要包括：

### 静态信息管理

主要采集本地计算机信息，数据量小，属于一般场景下的数据收发，http短链接，用普通json串传送到后端服务器即可。而且静态信息短时间内不会变化，每天无需采集多次提交。

### 动态信息管理

主要采集本地计算机变化频率高的信息，如网络流量，CPU，内存使用率等。与服务器互交频繁，也采用http 和json方式与服务器通信，每隔5分钟提交一次，不会对后端服务器造成明显压力。

### 运维管理

运维管理包括很多，其中主要关键点是解决服务端推送的问题，推送这个行为往往意味着服务器主动发起连接并推（Push）向客户端，也有一种方式实现，“伪”推送，其实现方式是客户端主动发起连接向服务端拉取(Pull)消息，但是后者基于客户端会间隔拉取检查服务端有没有消息，这样会给服务端造成无端压力，所以采用前者，用基于消息队列的中间件来实现服务端推送问题。

# 技术选型与执行方案

*本小节将对终端安全管控中使用的关键技术进行优劣对比及分析，推荐综合性价比高的技术模型，并且给出利用这些工具和技术框架组合利用起来的细化执行方案*

## 选型原则

终端安全管控平台是并非国家大型项目，也并不涉及南方电网核心业务数据，数据的安全性要求也不高，开发人员不够，因此在系统建设中对依赖组件，框架和库不做特殊要求，都可以采用国内外主流的框架和库进行开发，提高开发效率。

另外，前后端选型还应考虑以下原则：

### 客户端选型原则

* 支持主流windows操作系统，XP除外。
* 支持32位和64位系统。

### 后端选型原则

* 尽量选择企业级快速开发框架
* 尽量选择开源免费的基础框架和库，还有工具
* 尽量选择开发效率高，且跨平台的语言

## 客户端选型

### 客户端开发构建工具选型

因为客户端需要运行在除XP外，各种主流的windows操作系统上，所以客户端开发编译工具采用微软官方的Visual Studio系列，其中C++的构建链接工具实属MSVC最为强大，与windows平台最为契合，其他还包括方便的内置调试器，性能热点探测器，代码分析工具。Visual Studio 系列选择Visual Studio 2013，因为VS2013支持主流的C++ 11、C++ 14标准的特性外，编译出的程序能正确在windows 7 非SP1下运行。Visual Studio 2015，而编译出来的程序，不能在windows 7 非SP1下运行，因为VS2015的运行库（runtime）不能在非SP1下安装，所以，不能选择版本高的VS2015开发。

### 客户端开发网络协议库选型

因为客户端之后可能需要通过多种协议访问服务端，除http之外还可能有https，ftp等，所以选择Curl网络测试命令行工具开源出的底层库-----libcurl，该库支持主流网络应用层协议，比如http，https，smtp，imap，ftp，ldap，pop3等。而且该库的实现小而精，并不庞大臃肿，且线程安全，可跨大部分平台系统，包括windows，Linux，FreeBSD，Mac OS X等。

### 数据传输压缩库选型

数据压缩库采用zlib库，zlib是一种无损数据压缩的库，并且使用广泛，从操作系统内部实现，到网络服务器软件的底层都采用此压缩库。Zlib封装了deflate压缩算法（gzip也封装了deflate），而且deflate压缩算法其中用了哈夫曼编码，该编码对于文本类压缩有很高的压缩率。

### 客户端开发语言选型

因为客户端大量的工作都需要与操作系统打交道，获取系统级别的数据使用操作系统提供的原生API更方便，所以采用与C语言兼容的C++ 。 并且C++ 标准有STL，配合STL的各种基础数据结构容器（map，hash，list等）和算法，可以提高开发效率，已经增加开发更加大型系统级应用的可能性。

除此之外，C++ 标准近几年也是演进非常之快，与时俱进，越来越有现代编程语言的风格，抛开传统的C++ 98标准外，从C++ 11开始支持函数式编程风格，再次之后C++ 14标准又对C++ 11进行了完善，在编译期更加有利于编译器探测出错误与隐患。而且最新的C++ 17标准已经制定完成，正等待大规模的编译器产商的实现。相比之C语言就只有很老的C89,C99,C11标准，且最新的C11标准用户并不广泛。

最后，C++ 一直伴随这有一个开源的准标准库----- boost。 该库由C++标准委员会维护，长期更新，该库的众多模块一直经受各种工业级的考验，质量得以保证，客户端的日志模块就是采用boost.log模块，支持按天旋转（day rotation），并且是高性能。客户端的单元测试框架采用的也是boost.unittest来完成实现。

### 客户端开发GUI框架选型

客户端的界面到后期具有一定的要求，具有不低的性能，同时又有较高的开发效率和利于人机互交的UI界面和美化潜能。由于项目原因，就仅仅限定在了使用C++语言的GUI框架。所以选择Qt来开发客户端的GUI界面。值得一提的是，由于客户端要维持可扩展性，必须是支持插件的，而Qt就有内置的插件框架。

之所以选择Qt，主要考虑以下几点方面：

* Qt自1991年诞生，虽项目经过几次辗转，但是历史悠久，久经工业界锤炼（比C++ 所提出的STL历史还要悠久），学习成本不高。
* Qt框架是商业软件，开源免费的版本虽然不提供官方技术支持，但是其文档质量在C++开源框架选择中，质量最高。
* 使用Qt框架的受众多，全球每年都会举办Qt开发者大会，参加人数众多。网络上的Qt资料很多，容易查找解决方案。
* Qt 性能高，其基础底层库每个版本都会进行大量优化，其中字符串类或者其他容器类都有CPU扩展指令集的优化，例如SSE指令集等。
* Qt与时俱进，紧跟C++ 新标准的发展，支持新的语言标准，并且其GUI的开发方式也紧跟潮流，Qt5的GUI可以采用QML技术，该技术类似Web前端的方式更先进的描述界面的方式与javascript结合编写业务性不强的UI界面，需要性能的地方再利用C++语言编写。缺点是UI界面与C++通信复杂。
* Qt使用旧式的方式Widget开发GUI其效率也不低，界面设计用Qt官方提供的界面设计器拖拽界面，完全使用C++实现逻辑，并且GUI的各种控件，组件非常成熟。例如：Qt的文本框控件支持富文本编辑。Qt的Widget控件还支持QSS美化界面，QSS继承于CSS2标准，支持CSS盒模型。
* Qt并非仅仅是一个GUI框架，它试图统一桌面端的开发，提供了应有尽有的模块，QtWebKit模块支持html页面渲染，QtNetWork模块支持WebSocket协议，http/https，ftp等协议，QtBluetooth模块支持蓝牙协议，QtSQL模块提供主流数据库的后端接口和前端查询功能，QtCharts和QtDataVisualization模块提供Qt强大的图表和数据可视化的展示的能力。Qt还有更多强大的模块和类。
* Qt框架编写出了很多著名大型的软件，有众多成功案例：

1. WPS Office的后期新版很多功能，组件是由Qt编写。
2. 国产Deepin Linux内置的大部分桌面模块，软件均由Qt编写。
3. Google Earth也是Qt编写
4. 国内著名软件YY语音客户端由Qt编写
5. 著名浏览器Opera（version 10之前的图形后端）
6. 著名IM工具Skype
7. 著名数学软件[Mathematica](https://link.zhihu.com/?target=http://zh.wikipedia.org/wiki/Mathematica" \t "https://www.zhihu.com/question/_blank)的GUI部分Linux和windows版本由Qt编写
8. 虚拟化软件Virtual Box
9. AutoDesk Maya，著名3D建模和动画软件
10. 文档生成工具Doxygen由Qt编写

## 客户端开发执行方案细化

### Libcurl网络协议库的包装

Lincurl的场景主要是用于客户端与Web服务器的http通信传输数据的一个接口库，但是由于这样的协议库可能会产生变化或替换，所以在产品代码实现阶段，我们会将libcurl的接口上再包装一层形成一个自己的稳定版本接口，这样就降低了对底层网络协议库的依赖。这样的方法其实早已在设计模式中存在，叫适配器模式(Adaptor)。由于libcurl是纯C语言的接口，为了方便，就在其接口上包装一个C++的接口类，根据需求提供相应的C++类接口。这样lincurl的初始化与销毁就与C++对象的生命周期关联上了。当然，对应的libcurl库的初始化和销毁的代码就在C++类中的构造函数和析构函数中实现。

### 插件系统的设计

因为客户端需要具有高扩展性，通过插件的方式扩展客户端GUI窗口或者功能。所以需要设计一个插件系统来容纳运行插件，插件以DLL动态链接库的实体形式存在。这样就就必须结合C++ 的多态与DLL编程结合的方式来编写插件系统。令人高兴的是，Qt已经有相应的插件系统模块了。

Qt的插件模块有两种，一种是增加Qt框架本身功能的插件系统，另一种是增加Qt编写的应用功能的插件系统。所以根据需求，我们采用后者。采用后者的方法需要通过以下开发步骤：

1. 用C++ 纯虚基类定义一组接口，该接口就是插件的接口
2. 使用Q\_DECLARE\_INTERFACE宏对Qt的元对象系统描述该接口的相关信息
3. 利用C++的继承机制继承C++的QObject和接口类，并对插件功能做出实现
4. 使用Q\_PLUGIN\_METADATA宏导出插件
5. 编译出插件DLL
6. 利用Qt自带的QPluginLoader类在主应用程序中加载或卸载插件

### 客户端采集数据与Web服务端通信方式设计

由于考虑产品阶段的客户端的数量会很庞大，很可能是省级的规模，数万台终端会访问Web服务器，所以面临并发量大，流量大的问题，因为单次客户端数据采集的通信数据量不小，之前的方式是采用json数据通过http协议POST到Web服务器端，但是面对大流量的问题，需要采取相应的减少数据流量的措施，首先考虑到的解决方案是压缩需要POST的数据，服务端再解压缩数据。考虑到二进制级别的压缩成本略高，所以采用文本形式的压缩。

二进制级别的数据压缩有以下几个缺点：

* 客户端和服务端难以调试，日志不方便打印分析
* 效率并不比基于文本的数据高上很多

http整个协议其实都是文本方式，现今整个互联网都在应用http协议，互联网的特点就是大并发，大流量的。所以不见得http会损失传输效率。而且传输的数据会采用zlib压缩，该库使用广泛，压缩和解压缩速度快，对文本的压缩率很高。

对于待压缩的文本数据，还是以json的方式表示，原因有以下：

* Json文本能以对象的方式描述数据结构，容易理解阅读
* Json是文本交换格式，所以在客户端与服务端联调的时候，容易分析问题
* Zlib对于长文本的格式压缩率可观

首先实现执行方案，步骤是这样的：

1. 客户端生成数据的json格式文本串
2. 把json串以UTF8的编码方式进行编码
3. 对UTF8编码的json串进行zlib压缩
4. POST已压缩的数据POST到服务端

最后，服务端以相反的解压缩，解码步骤解析数据，并存入数据库。

另外值得注意的是， JAVA实现 在JDK中,在java.util.zip包中已经内置了ZLIB的实现,所以后端开发人员只需简单学习接口使用并调用即可。客户端的zlib需要从官网下载并编译，才能集成到客户端工程中使用。从接口使用上来看，C接口的zlib也同样易于使用简单，学习成本不大。

### Zlib压缩库的包装

理由及原因参考[这里](#_Libcurl网络协议库的包装)。 同libcurl的包装一致。

### 客户端GUI的设计

Qt 框架自带了一些工具软件方便GUI界面的设计，其中Widget方式的界面设计工具就是Qt Designer，它底层以编辑类XML的树形结构描述UI层级控件（.ui文件）。Qt Dsigner中内置了GUI界面大多数的控件，可以以拖拽的方式对界面进行布局设计，完全实现UI界面与代码逻辑分离。

Qt Designer本质上就是一个文本编辑器，只不过该文本的结构描述了UI界面的构成。Qt Designer中的每种类型的控件在Qt代码中有对应的C++类实现，编辑完成的界面通过Qt的UIC编译器可以编译出对应的C++界面框架代码，代码中暴露出了UI中的控件对象，然后开发人员可以直接使用对象来调用对象的方法进行开发GUI的功能，而GUI中的控件都会有各自的事件，开发人员使用Qt的信号槽机制与特定控件对象的事件关联起来，编写对应的槽函数（事件响应函数），来实现代码逻辑。

## 服务器端选型

### 服务器端接入层选型

接入层的服务器软件需要经受住大并发的考验，因为平台系统的所有终端的http请求都会由这入口进入，所以主要选择是Nginx来做负载均衡和反向代理，选择其主要有以下理由：

* 大型互联网企业的业务特点都是高并发，大流量的，这些企业主流都采用Nginx做负载均衡和反向代理，经受百般锤炼。
* Nginx在Linux下采用能应对高并发的Epoll这样的底层网络I/O模型，为高并发提供底层支撑。
* Nginx采用异步非阻塞模型，更加有利于性能提高。
* Nginx通过配置文件配置所需功能，配置方便，网络资料齐全，而且还具有一定的可扩展性，可以用lua编写插件扩展功能。

### 服务器端应用服务器选型

应用Web服务器（容器）采用tomcat。选择其主要考虑到以下几个方面：

* J2EE规范实现得最好的应用Web容器，对新的Servlet规范跟进块
* 配置简单，易于使用
* 大型互联网企业在这样类似的架构下主流都选择tomcat做Web容器
* 大型热门的开源项目，如果遇到BUG，得到的社区反馈率快，质量也高

### 服务器端存储层选型

存储层的服务器数据库，主要能承受大量的高性能的查询能力，又对于大型国有企业内部平台，一般主流都采用Oracle，选择Oracle主要考虑以下几个方面：

* 对于南方电网这样的国企有专业的Oracle运维团队
* Oracle很多年以来的发展，使它经受住了各方面的复杂考验
* Oracle基本成长于传统大型行业的数据存储，比如银行，金融这样对安全性，健壮性要求高的行业。

### 服务器端开发语言选型

由于面向对象的设计模式能够帮助减轻大型软件概念和抽象核心的重新发现和重新生成，一直被人们所推崇。但在实践和服务端编程招聘市场中，掌握Java语言的程序员相比其他语言多很多，Java语言有成熟的技术栈，成熟的IDE和各式各样提高开发效率的工业级框架，数不清的类，使后台系统开发成本变得更低，编写出的代码更加容易理解阅读，尤其是Spring框架与Java语言互相促进，许多软件公司转而采用JAVA技术来构建大型服务软件以应付复杂环境，而这几年的社会现象观察出，互联网技术应用蓬勃发展，越来越大的改变了人们的生活方式，对Java技术的依赖需求就越大。而且随着HotSpot JVM的JIT等虚拟机优化技术的提升，Java即使跟与效率见长的C语言比运行效率也不会有多少差距了，况且终端安全管控平台的主要是基于网络的平台，显然主要性能瓶颈是在网络IO上，而非运行效率上，从性能上来说，与采用何种语言无关，反而其他语言并不是什么灵丹妙药。

为了解决大型软件的设计危机，软件工程界诞生了敏捷与测试驱动等软件快速开发方法，采用接口、组件、设计模式、框架等现代软件设计概念与方法寻求解决方案。设计模式提供了一种应对解决重复的一种抽象方法论，此方法提供了标准软件开发的一种解决方式。框架为应用程序提供了可重用的软件构件，框架实例化设计模式帮助开发人员避免一些通用软件构件的重新生成，产生准完备的应用程序骨架，这个骨架可以通过从框架中可重用的构件进行配置定制。由于框架与关键的大型网络服务软件的编程任务是紧密结合的，例如服务初始化、错误处理、流控制、事件分离、并发控制等等，可重用的范围比使用传统函数库大大增强，再结合代码自动生成器会大大提高软件的开发速度降低开发与维护成本。

选择平台的原则是技术先进性、技术实施的可行性、开发一致性、平台经济性，尽可能作到简单划一，避免过多的技术平台和手段的混合使用。选择适应性开放性强的平台，有利于提高系统的适应性和灵活性，降低用户的维护成本和开发者的开发成本。大型软件的各种复杂性往往是由于采用臃肿技术、设计过度导致的，许多本来是替代臃肿设计方式的新设计方式往往在发展的过程中导入了另一种复杂性。虽然这个世界没有一劳永逸的万能解决方案，而且很多人认为Java会导致系统的过度设计。其实解决问题的思想本质是一样的，万变不离其宗，Java也可以使用面向接口编程的设计。

因为其他编程语言的后端技术体系中缺少成熟的各种中间件，框架及组件，导致软件设计者经常从底层开始构建、验证和维护软件系统，这样的开发既费时又费力，在严酷的市场竞争环境下往往处于被动地位。随着市场与技术的发展，计算机硬件越来越便宜、处理能力越来越强大，网络的速度与承载能力也越来越快越高，导致人们对软件的要求越来越高，需求越来越复杂，而提供服务的软件也越来越大、越来越慢，开发和维护成本越来越高。

终端安全管控平台提建设时后台应采用一个JEECG的企业级开发框架，可无限扩展，可动态加载。

并且一个开发人员就能承担Web前端后台的开发，实现管理端的功能，尤其是当前广泛使用的前端后端所必需的组件。这个开发框架面向企业内部的OA平台，ERP平台，CRM平台的开发者，提取各个不同应用软件的共同部分。通过参数化配置，构造高效的、稳定的、安全的、并且和具体应用无关的、统一软件开发环境与基础框架，结合代码生成器的使用，可降低软件的开发维护成本以及客户使用成本，大大简化网络应用软件的开发。

因此后台架构我们将全部采用Java语言来开发。

### 服务端文件存储选型

文件存储服务主要是为了终端安全管控平台的的企业应用下载商店提供支撑，类似于App Store。由管理人员经过审核通过的软件，工具进行上传。然后管理端进行文件推送可以把存储地址推送给客户端，由客户端进行下载。

考虑到Ftp所搭建的服务器在分布式集群的方面不可靠，不能随着文件应用商店的容量和规模增长，而且Ftp并不能支撑高并发量的考验。所以使用MongoDB和它内置的GridFS 来做文件应用的存储。

MongoDB使用高效的二进制存储，适合存储大型对象（高清视频，图片文件等）。

考虑MongoDB技术体系做文件存储还基于以下原因考虑：

* 文件存储格式为Bson，使用易于掌握和理解的Json风格语法。相对Json来说，Bson拥有更好的性能，主要表现为更快的遍历速度、操作更简易、增加了额外的数据类型。
* 支持多种复制模式，提供冗余及自动故障转移。支持Master-Slave、Replica Pairs/Replica Sets、有限Master-Master模式。
* 性能高、速度快。在多数场合，其查询速度对于MySQL要快的多，对于CPU占用非常小。部署很简单，几乎是零配置。
* 自动处理碎片，支持自动分片功能实现水平扩展的数据库集群，可以动态添加或移除节点。
* 内置GridFS，支持海量存储。
* 可通过网络访问，采用高效的MongoDB网络协议，在性能方面要优于http或Rest协议。
* 对于一般的文件存储，单个对象的4到16M的存储容量能够满足需求，但无法满足对于一些大文件的存储，如高清图片、设计图纸、视频等，因此在海量数据存储方面，MongoDB提供了内置的GridFS，可以将一个大文件分割成为多个较小的文档，可以指定文件分块标准，对用户是透明的。
* MongoDB的客户端支持多种语言，Java，C/C++都包括。

### 服务端消息队列中间件选型

消息队列中间件主要是为了服务端的消息文件推送提供支撑，避免接入层和应用服务器消耗资源，减小压力。当今MQ的中间件有很多实现，包括RabbitMQ， Active MQ， Kafka，ZeroMQ等，选择很多，每个都有自己不同的应用场景。

其实使用生产者-消费者模式在服务间进行异步通信是一种比较好的思想。

因为服务间依赖由强耦合变成了松耦合。消息中间件都会提供持久化机制，在消费者负载高或者掉线的情况下会把消息保存起来，不会丢失。就是说生产者和消费者不需要同时在线，这是传统的请求-应答模式比较难做到的，需要一个中间件来专门做这件事。其次消息中间件可以根据消息本身做简单的路由策略，消费者可以根据这个来做负载均衡，业务分离等。

RabbitMQ是一个AMQP实现，传统的messaging queue系统实现，基于Erlang。老牌MQ产品。AMQP协议更多用在企业系统内，对数据一致性、稳定性和可靠性要求很高的场景，对性能和吞吐量还在其次。

Kafka是linkedin开源的MQ系统，主要特点是基于Pull的模式来处理消息消费，追求高吞吐量，一开始的目的就是用于日志收集和传输，0.8开始支持复制，不支持事务，适合产生大量数据的互联网服务的数据收集业务。

ZeroMQ只是一个网络编程的Pattern库，将常见的网络请求形式（分组管理，链接管理，发布订阅等）模式化、组件化，简而言之socket之上、MQ之下。对于MQ来说，网络传输只是它的一部分，更多需要处理的是消息存储、路由、Broker服务发现和查找、事务、消费模式（ack、重投等）、集群服务等。

ActiveMQ是Apache下的顶级开源消息中间件项目，社区活跃，老牌MQ系统，与RabbitMQ比较接近，支持JMS标准，支持订阅者-发布者语义的topic功能，并且支持上GB的文件推送。而且支持AMQP，MQTT，STOMP，OpenWire这四种消息协议。

Nats，较新的消息系统，Golang开发。在吞吐量方面远远高出ActiveMQ，Kafka，RabbitMQ等。官方提供多种语言的客户端。Nats的设计目标是高性能，易于使用，可伸缩，轻量。官方建议的使用场景是：寻址发现，命令和控制，负载均衡，多路可伸缩能力，定位透明，容错。但是由于官方的设计哲学认为，高质量的QoS（Quality of Service）应该在客户端构建，故只建立的请求-应答，所提不提供下列服务：

1. 持久化
2. 事务处理
3. 增强的交付模式
4. 企业级消息队列

EMQTT也是较新的Erlang实现MQTT协议的消息系统，但是MQTT协议设计之初的考虑就是消息负载小，低延迟，低功耗的车联网，物联网的消息系统，主要擅长远程传感器级别的设备的消息指令控制。不擅长GB级别的文件级消息推送。

根据以上的MQ中间件的比较，最后选择ActiveMQ。

## 服务端开发执行方案细化

### Nginx负载均衡及反向代理的配置

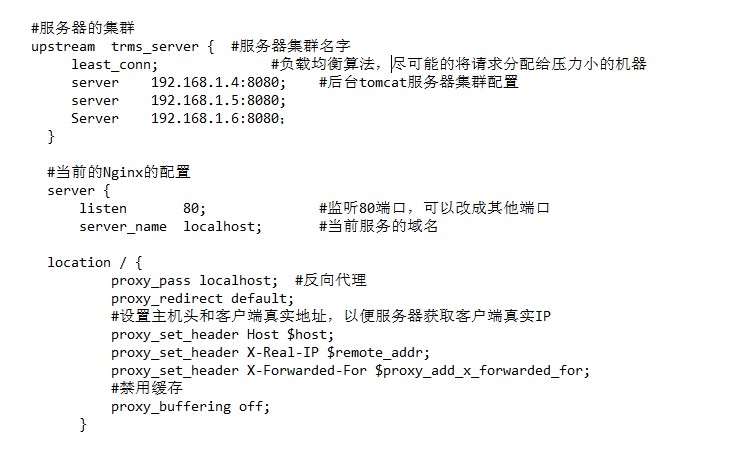
负载均衡的目的是为了解决单个节点压力过大，造成Web服务响应过慢，严重的情况下导致服务瘫痪，无法正常提供服务。

# Nginx负载均衡是通过upstream模块来实现的，内置实现了三种负载策略，配置还是比较简单的。

# 轮询（默认） Nginx根据请求次数，将每个请求均匀分配到每台服务器

1. 最少连接  
    将请求分配给连接数最少的服务器。Nginx会统计哪些服务器的连接数最少。
2. IP Hash   
    绑定处理请求的服务器。第一次请求时，根据该客户端的IP算出一个HASH值，将请求分配到集群中的某一台服务器上。后面该客户端的所有请求，都将通过HASH算法，找到之前处理这台客户端请求的服务器，然后将请求交给它来处理。

修改Nginx的配置文件达到目的，打开Nginx根目录下的conf文件夹，找到nginx.conf文件，以文本编辑器打开。按以下方式编辑。



基本常用Nginx命令：

./nginx 启动nginx

./nginx -t -c [配置文件路径] 检查Nginx配置是否正确

./nginx -s reload 不用关闭Nginx，重新载入配置，平滑重启

### Tomcat Web容器服务配置

考虑之后的产品环境可能是省级联通的服务，需要承受比以往更大的并发和流量，所以负载均衡服务器之后连接的是tomcat服务器集群，这样就可以根据并发量来增加或减少服务器。达到资源的合理利用。每台tomcat服务器下都需要部署java生成的war应用。

以下是tomcat的基本配置和使用：

1. 为每台需要充当tomcat服务器的机器上下载和安装tomcat和JDK（tomcat运行需要JDK中的JRE支持）。
2. 可根据网络上参考配置tomcat的JDK运行环境
3. Tomcat的根目录是这样的结构



1. 编译好的war文件就是需要部署在webapp目录底下
2. 在bin中可以找到启动和关闭tomcat的脚本

以下是tomcat基本的命令：

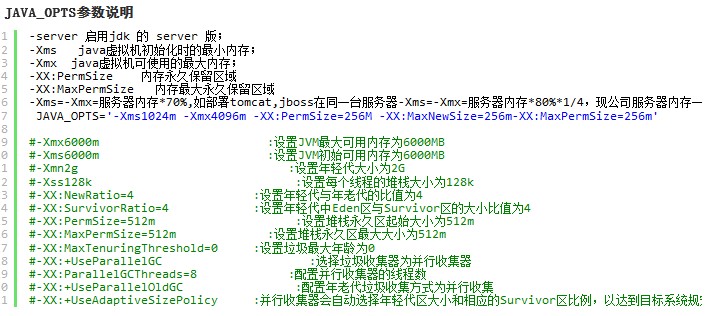
./startup.sh 启动tomcat

./shutdown.sh 关闭tomcat

Tomcat可以通过一定的配置来调优性能，以下是tomcat性能优化的几点：

* Tomcat依赖的JVM性能调优

Tomcat内存优化主要是对 tomcat 启动参数优化，我们可以在 tomcat 的启动脚本 catalina.sh 中设置 JAVA\_OPTS参数。可根据以下参数在实际生产环境中调优



* 调整tomcat线程数

Tomcat通过线程池来为用户访问提供响应，对于上线的系统初步估计用户并发数量后，再调整线程池容量。例如，用户并发数量在100左右时，可以设置minProcessors="100"，maxProcessors="100"。将最大和最小设置为一样后，线程池不会再释放空闲的线程，当用户访问突然增加时，不需要再消耗系统资源去创建新的线程。

* 调整tomcat最大连接数

这个其实最复杂，即使用户并发量大，但是系统反应速度快，也没必要把这个值设置太高，高了系统需要消耗大量的资源去切换线程，但是如果设置太低也会造成应用无法满足用户并发需要。因此设置这个最好能够结合整个系统的跟踪与调优，使系统达到最好的平稳状态，一般设置为maxProcessors的1.5倍即可。

* 调整tomcat http网络超时

主要是HTTP协议也有个连接过程，客户端连接到服务器上后，如果长时间没有得到处理就会被释放。如果服务器处理速度较慢，但是希望每个用户都能得到有效处理，或者网络环境不好，需要保证用户不会因为超时中断，也可以把时间加长。但是一般设置成connectionTimeout="30000"即可。太长对系统来说价值不大，反而会浪费系统资源在无谓的长连接上。

* 调整tomcat 压缩功能

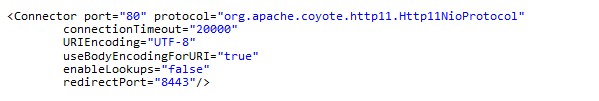
tomcat作为一个应用服务器，也是支持 gzip 压缩功能的。我们可以在 server.xml 配置文件中的 Connector 节点中配置如下参数，来实现对指定资源类型进行压缩。



注意，如果使用Nginx代理，那么tomcat自身不需要进行压缩，会给tomcat服务器增加压力，Nginx就可以配置压缩功能

* 调整tomcat的网络IO模式

Tomcat的网络IO模式有三种，BIO,NIO,APR。选择NIO。因为第一种性能底下，第三种性能最高，但是需要安装配置，第二种配置方便，而且是Java高性能的异步IO处理方式。要以这样的方式运行tomcat，需要在sserver.xml里的Connector节点,修改protocol为



重启tomcat服务器即可生效。

以下是tomcat的配置示例：



### Jeecg应用开发框架

Jeecg企业级快速开发框架是为了让后端的人员也能开发前端，生成前端需要展示的数据，方面开发，在开发人员不足的时候，这是一个妥协。该框架能从前端到后端的数据需求全部通过生成器生成好了，开发人员只需要通过Java代码编写业务逻辑。

Jeecg快速开发框架官方网站提供详细的[文档和入门视频](https://pan.baidu.com/share/home?uk=2668473880" \l "category/type=0)

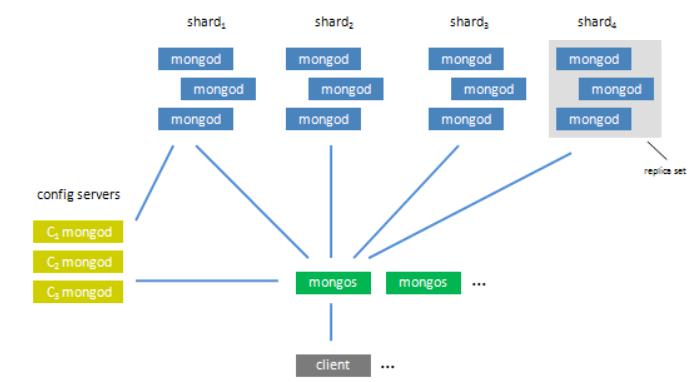
其中经过细化分析，可以着重阅读以下文档，即可完成日常开发需求。

* JEECG开发环境入门（Maven-Eclipse）
* JEECG online表单对外接口
* JEECG online表单开发指南
* JEECG开发指南
* JEECG权限开发手册

### MongoDB文件服务器配置

根据产品的需求，基本上选定MongoDB和其内置的GridFS了。

GridFS结合自动分片及自动复制技术，可以实现高性能的分布式数据库集群架构，从而进行海量数据存储。安装MongoDB的服务器



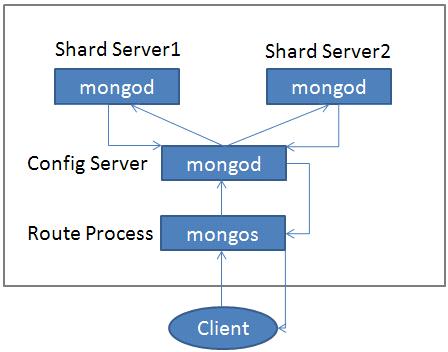
MongoDB Sharding Cluster需要三种角色：

（1）Shard Server：即存储实际数据的分片，每个Shard可以是一个mongod实例，也可以是一组mongod实例构成的Replica Set。

（2）Config Server：用来存储所有shard节点的配置信息、每个chunk的shard key范围、chunk在各shard的分布情况、该集群中所有DB和collection的sharding配置信息。

（3）Route Process：这是一个前端路由，客户端由此接入，然后询问Config Servers需要到哪个shard上查询或保存记录，再连接相应的shard进行操作，最后将结果返回给客户端，而这一切对客户端是透明的，客户端不用关心所操作的记录存储在哪个shard上。

为了测试方便，下面在同一台物理机器上构建一个简单的Sharding Cluster，如下图所示。



配置测试环境如下：

模拟2个Shard服务器和1个Config服务器，均运行在本机127.0.0.1上，只是端口不同：

（1）Shard Server1：127.0.0.1：27020。

（2）Shard Server2：127.0.0.1：27021。

（3）Config Server：127.0.0.1：27022。

（4）Route Process：127.0.0.1：27017。

启动相关服务进程：

c:\mongodb 2.0.0\bin>mongod --shardsvr --dbpath "c:\mongodb 2.0.0\db" --port 27020

d:\mongodb 2.0.0\bin>mongod --shardsvr --dbpath "d:\mongodb 2.0.0\db" --port 27021

e:\mongodb 2.0.0\bin>mongod --configsvr --dbpath "e:\mongodb 2.0.0\db" --port 27022

e:\mongodb 2.0.0\bin>mongos --configdb 127.0.0.1:27022

配置Sharding：

(1)e:\mongodb 2.0.0\bin>mongo

(2)use admin

(3)db.runCommand( { addshard : "127.0.0.1:27020", allowLocal : 1,

maxSize:2 , minKey:1, maxKey:10 } )

(4)db.runCommand( { addshard : "127.0.0.1:27021", allowLocal : 1, minKey:100  } )

(5)config =connect("127.0.0.1:27022")

(6)config = config.getSisterDB("config")

(7)ecDocs=db.getSisterDB("ecDocs")

(8)db.runCommand({enablesharding:"ecDocs"})

(9)db.runCommand( { shardcollection : "ecDocs.filedocs.chunks", key : { files\_id : 1 } } )

(10)db.runCommand( { shardcollection : "ecDocs.filedocs.files", key : { \_id : 1 } } )

以上的ecDocs是指数据库名，filedocs是指用户自定义的GridFS的文件集合名，系统默认文件集合名为fs。

之后通过Java和C++的Driver访问MongoDB的Router，来实现读写文件。

### ActiveMQ消息推送配置

ActiveMQ为了满足消息推送的需求，支持pub/sub模式，支持Java，C++的客户端驱动。

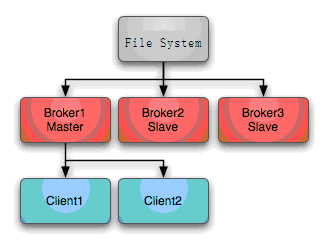
单点的ActiveMQ作为企业应用无法满足高可用和集群的需求，所以ActiveMQ提供了master-slave、broker cluster等多种部署方式，但通过分析多种部署方式之后我认为需要将两种部署方式相结合才能满足省级并发的分布式和高可用的需求，所以后面就重点将解如何将两种部署方式相结合

Master-Slave部署方式

###### 1）shared filesystem Master-Slave部署方式

  主要是通过共享存储目录来实现master和slave的热备，所有的ActiveMQ应用都在不断地获取共享目录的控制权，哪个应用抢到了控制权，它就成为master。

   多个共享存储目录的应用，谁先启动，谁就可以最早取得共享目录的控制权成为master，其他的应用就只能作为slave。



###### 2）shared database Master-Slave方式

   与shared filesystem方式类似，只是共享的存储介质由文件系统改成了数据库而已。

###### 3）Replicated LevelDB Store方式

   这种主备方式是ActiveMQ5.9以后才新增的特性，使用ZooKeeper协调选择一个node作为master。被选择的master broker node开启并接受客户端连接。其他node转入slave模式，连接master并同步他们的存储状态。slave不接受客户端连接。所有的存储操作都将被复制到连接至Master的slaves。

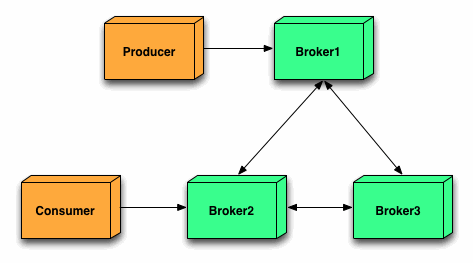
如果master死了，得到了最新更新的slave被允许成为master。fialed node能够重新加入到网络中并连接master进入slave mode。所有需要同步的disk的消息操作都将等待存储状态被复制到其他法定节点的操作完成才能完成。所以，如果你配置了replicas=3，那么法定大小是(3/2)+1=2. Master将会存储并更新然后等待 (2-1)=1个slave存储和更新完成，才汇报success。至于为什么是2-1，熟悉Zookeeper的应该知道，有一个node要作为观擦者存在。

单一个新的master被选中，你需要至少保障一个法定node在线以能够找到拥有最新状态的node。这个node将会成为新的master。因此，推荐运行至少3个replica nodes，以防止一个node失败了，服务中断。

2、Broker-Cluster部署方式

前面的Master-Slave的方式虽然能解决多服务热备的高可用问题，但无法解决负载均衡和分布式的问题。Broker-Cluster的部署方式就可以解决负载均衡的问题。

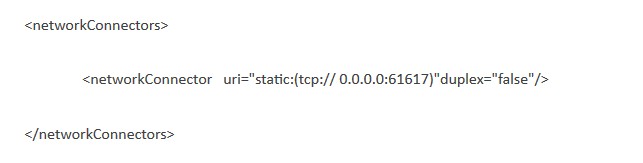
 Broker-Cluster部署方式中，各个broker通过网络互相连接，并共享queue。当broker-A上面指定的queue-A中接收到一个message处于pending状态，而此时没有consumer连接broker-A时。如果cluster中的broker-B上面由一个consumer在消费queue-A的消息，那么broker-B会先通过内部网络获取到broker-A上面的message，并通知自己的consumer来消费。类似下图描述



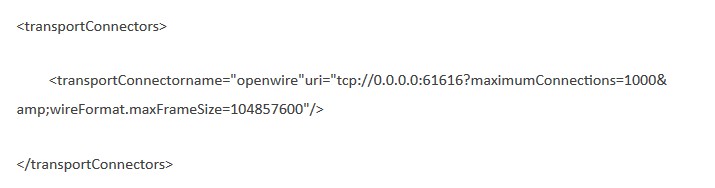
* 静态broker cluster部署

需要在activemq.xml配置文件中静态指定broker需要建立连接的其他broker，比如把两个broker作为一个cluster，分别是broker-A，broker-B。

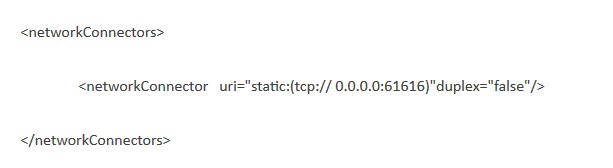
首先在broker-A节点的配置文件中添加networkConnector节点：



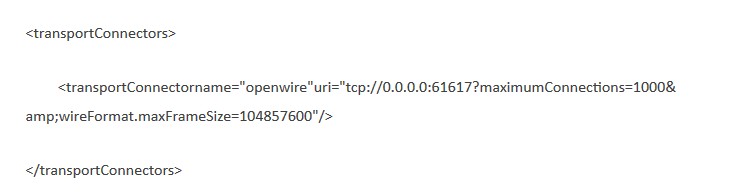
修改broker-A节点的服务提供端口为61616：



在broker-B节点添加networkConnector节点：



修改Broker-B节点中的服务提供端口为61617：



最后分别启动broker-A和broker-B。

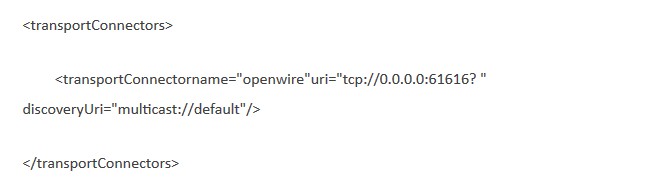
* 动态broker cluster部署

在activemq.xml文件中不直接指定Broker需要建立桥连接的其他Broker，由activemq在启动后动态查找。

首先在broker-A节点中添加networkConnector节点：



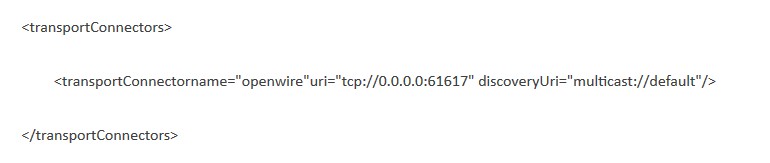
修改broker-A节点中的服务提供端口为61616：



在broker-B节点中添加networkConnector节点：



修改broker-B节点中的服务提供端口为61617：



启动Broker-A和Borker-B。

静态broker cluster在实际中大量使用，动态的反而不多，考虑到案例多，选择静态broker cluster部署。

# 未来省级集中建设策略展望

*本小节根据未来南网在省级集中的战略规划，对终端安全管控平台如何顺利过渡到省级集中进行整体描述，对过渡过程中需要采用的方式方法及可能面临的问题进行说明。*

楚雄供电局终端安全管控建设完成后，并趋于稳定，然后推广到云南省个地级市运行，通过1至2年的稳定期，当系统软硬件架构和可靠性经过充份的验证并趋于稳定时，可考虑启动终端安全管控平台的省级集中建设工作，通过采用分布式建设来对系统软硬件架构进行充份验证，并通过各地级市局公司实际建设过程中统一并稳定需求，为建设省级集中的终端安全管控平台做好技术及业务上的准备。未来省级集中建设大致可分为省级硬件环境建设和各地区数据和业务集中迁移两个步骤。

# 整体计划

*本小节将对整体工作开展制定推进计划。终端安全管控平台建设是涉及范围广的系统工程，需要从全局角度制定详细周密的计划来确保整体工作顺利有序的推进。*

营销信息一体化建设是一项重大的系统工程，为了降低风险、提高效率，最大限度的发挥一体化建设的经济价值，整个营销信息一体化建设工作应按“起点适当、起步平稳”、“适时过渡、积极推进”的原则展开，建议分两阶段推进工作。

* + **前期准备阶段：** 主要是结合楚雄供电局组织机构分布，软硬件能力及员工数量的特点，编制《终端安全管控平台建设方案》，并根据方案进行技术架构原型设计、开发及验证。
  + **系统实施阶段：** 根据《终端安全管控平台建设方案》开展功能梳理、典型设计、系统开发、系统测试、试点实施、验收推广等工作。

## 前期准备阶段

本阶段的主要工作是编制《终端安全管控平台建设方案》，并进行技术架构原型设计、开发及验证。主要工作如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **序号** | **项目名称** | **具体内容** | **起止时间** | **输出工件** |
| **1** | 《终端安全管控平台建设方案》框架编写 | 结合终端安全管控平台的特点编制《终端安全管控平台建设方案》框架 | 2017.6.5-2010.6.9 | 《终端安全管控平台建设方案》框架 |
| **2** | 《终端安全管控平台建设方案》审定 | 由犀鸟科技上级领导审定 | 待定 | 框架修订意见 |
| **3** | 《终端安全管控平台建设方案》内容编写 | 根据修订意见调整框架，并编写内容 | 待定 | 《终端安全管控平台建设方案》初稿 |
| **4** | 《终端安全管控平台建设方案》审定 | 由犀鸟科技公司研发部门审定 | 待定 | 审定意见 |
| **5** | 《终端安全管控平台建设方案》修改 | 根据审定意见修改建设方案内容，并征求犀鸟研发部意见 | 待定 | 《终端安全管控平台建设方案》征求意见稿 |
| **6** | 《终端安全管控平台建设方案》审定评稿 | 犀鸟科技公司联合审评 | 待定 | 《终端安全管控平台建设方案》定稿 |

## 系统建设阶段

本阶段的主要工作包括功能梳理系统开发、系统测试、试点实施、试点验收及系统推广等工作。

1. **功能梳理：**本阶段的工作内容是完成现有平台的功能梳理，使开发和测试统一化、标准化、规范化。
2. **系统开发：**本阶段的工作内容是根据功能完成代码实现，确保功能实现完全遵循典型设计要求。
3. **系统测试：**本阶段的工作内容是进行功能测试和压力测试，确保系统功能实现的正确性、适用性和高效性。功能测试目的是严格地测试所有系统功能，以验证所有硬件和软件的功能运行情况。压力测试的目的是测试系统在一定饱和状态下，例如CPU、内存在饱和使用情况下，系统能够处理的会话能力以及系统是否会出现错误。
4. **试点实施：**本阶段的工作内容是选择客户现场调试安装，并根据客户的沟通交流中改进平台
5. **试点验收：**本阶段的工作内容是对客户单位使用的结果阶段性验收，包括系统功能完备性、系统数据完整性、系统总体性能、系统运行情况、系统运行管理保障等，以确保系统在投入运行后的可行性、稳定性。
6. **系统推广：**在全省范围内推广实施终端安全管控平台。