苏州凌世智能科技有限公司



**MES系统**

2022年2月

目录

[一、 XXXX简介 6](#_Toc95315997)

[1. XXX集团 6](#_Toc95315998)

[二、 项目描述 6](#_Toc95315999)

[1. 项目概述 6](#_Toc95316000)

[2. 客户现状 6](#_Toc95316001)

[3. 项目目标 6](#_Toc95316002)

[4. 项目范围 6](#_Toc95316003)

[4.1 业务范围 6](#_Toc95316004)

[4.2 模块范围 7](#_Toc95316005)

[三、 设计依据与原则 7](#_Toc95316006)

[1. 项目建设参考标准及规范 7](#_Toc95316007)

[2. 设计原则 8](#_Toc95316008)

[2.1 可靠性 8](#_Toc95316009)

[2.2 安全性 8](#_Toc95316010)

[2.3 可扩展性 9](#_Toc95316011)

[2.4 先进性 9](#_Toc95316012)

[2.5 实用性 9](#_Toc95316013)

[2.6 可维护性 9](#_Toc95316014)

[2.7 标准化和开放性 9](#_Toc95316015)

[2.8 模块化 10](#_Toc95316016)

[2.9 网络设备的互连性 10](#_Toc95316017)

[2.10 人性化服务 10](#_Toc95316018)

[四、 系统总体设计 10](#_Toc95316019)

[1. 总体设计要求 10](#_Toc95316020)

[2. 软件架构 12](#_Toc95316021)

[3. 服务器网络拓扑结构 13](#_Toc95316022)

[3.1 软件性能： 14](#_Toc95316023)

[3.2 服务器性能： 14](#_Toc95316024)

[3.3 服务器配置： 14](#_Toc95316025)

[五、 系统功能设计 15](#_Toc95316026)

[1. 系统功能模块及清单 15](#_Toc95316027)

[六、 可靠性、安全性、可扩展性等 16](#_Toc95316028)

[1. 可靠性解决方案 16](#_Toc95316029)

[2. 安全性解决方案 17](#_Toc95316030)

[3. 可扩展性解决方案 18](#_Toc95316031)

[4. 先进性解决方案 20](#_Toc95316032)

[5. 实用性解决方案 20](#_Toc95316033)

[6. 可维护性解决方案 21](#_Toc95316034)

[7. 标准化和开放性解决方案 22](#_Toc95316035)

[8. 模块化解决方案 23](#_Toc95316036)

[9. 网络设备互连性解决方案 23](#_Toc95316037)

[10. 人性化服务解决方案 24](#_Toc95316038)

[七、 系统交互接口 25](#_Toc95316039)

[1. 系统接口 25](#_Toc95316040)

[1.1 管理、实现方式、方案等 25](#_Toc95316041)

[1.2 服务接口的预留工作 27](#_Toc95316042)

[1.3 其他接口的预留工作 28](#_Toc95316043)

[2. 单点登录的预留工作 28](#_Toc95316044)

[八、 关键技术解决方案 28](#_Toc95316045)

[1. 基础技术介绍 28](#_Toc95316046)

[1.1 ABP（.Net Core）框架 28](#_Toc95316047)

[1.2 Web服务技术 29](#_Toc95316048)

[1.3 Vue 30](#_Toc95316049)

[1.4 Redis 31](#_Toc95316050)

[1.5 MQTT 31](#_Toc95316051)

[1.6 KAFKA 32](#_Toc95316052)

[1.7 Docker 32](#_Toc95316053)

[1.8 微服务 34](#_Toc95316054)

[2. 兼容性解决方案 34](#_Toc95316055)

[3. 高性能高并发解决方案 34](#_Toc95316056)

[a) 安全机制 35](#_Toc95316057)

[1. 系统平台安全 35](#_Toc95316058)

[1.1 身份鉴别 35](#_Toc95316059)

[1.2 访问控制 36](#_Toc95316060)

[1.3 安全审计设计 37](#_Toc95316061)

[1.4 通讯完整性、保密性 38](#_Toc95316062)

[1.5 软件容错 39](#_Toc95316063)

[九、 项目实施 39](#_Toc95316064)

[1. 标准规范的方法论 40](#_Toc95316065)

[1.1 项目管理流程 40](#_Toc95316066)

[2. 敏捷的项目管理工具 45](#_Toc95316067)

[2.1 JIRA 45](#_Toc95316068)

[2.2 DevOps 46](#_Toc95316069)

[3. 项目实施计划 47](#_Toc95316070)

[4. 项目组织架构及工作说明 47](#_Toc95316071)

[4.1 项目组织架构 47](#_Toc95316072)

[4.2 工作说明 48](#_Toc95316073)

[5. 实施过程及责任 49](#_Toc95316074)

[5.1 系统需求分析 49](#_Toc95316075)

[5.2 系统功能设计 50](#_Toc95316076)

[5.3 系统功能开发 50](#_Toc95316077)

[5.4 系统测试 50](#_Toc95316078)

[5.5 产品交付验收 51](#_Toc95316079)

[5.6 上线期维护服务 51](#_Toc95316080)

[6. 系统交付 53](#_Toc95316081)

[6.1 交付项目 53](#_Toc95316082)

[6.2 交付方式 53](#_Toc95316083)

[6.3 授权方式 53](#_Toc95316084)

[7. 系统实施 53](#_Toc95316085)

[8. 项目验收 54](#_Toc95316086)

[8.1 项目验收依据 54](#_Toc95316087)

[8.2 项目验收时间安排 55](#_Toc95316088)

[8.3 项目验收内容 55](#_Toc95316089)

[十、 产品售后服务 55](#_Toc95316090)

[1. 服务目标 55](#_Toc95316091)

[2. 凌世运维服务 56](#_Toc95316092)

[2.1 服务形式及内容 56](#_Toc95316093)

[2.2 服务流程 57](#_Toc95316094)

**版本信息**

|  |  |  |  |
| --- | --- | --- | --- |
| **版本号** | **日期** | **描述** | **提交人** |
| 0.1 | 2022年2月9日 |  | Link石 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**术语解释**

| **序号** | **术 语** | **解 释** |
| --- | --- | --- |
| 1 | API | 应用程序接口，用来实现应用程序功能调用 |
| 2 | OPC | 通用的开放工业通讯协议，实现设备实时数据交换 |
| 3 | HTTP | 应用层通信协议，用来实现IT系统内部客户端与服务器端的通信；也可以用来实现应用系统之间的通信 |
| 4 | KAFKA | 大数据技术，分布式消息队列实现高吞吐量的现场数据在服务端的接入 |
| 5 | Redis | 内存数据库，用来实现高速数据内存存取 |
| 6 | Flink | 大数据技术，流式数据处理引擎 |

# XXXX简介

## XXX集团

# 项目描述

## 项目概述

本项目工作说明书是XX有限公司（以下简称“甲方”）和苏州凌世智能科技有限公司（以下简称“乙方”）关于液压科技生产报工项目签署的《合同》的附件，旨在界定项目服务明细条款。本项目说明书随主合同签署同时生效。

本工作说明书定义了甲方和乙方在本项目中约定的具体系统模块，功能，实施范围、任务、实施计划、基本假定和交付项等内容，如与其他协议或约定存在差异，以本SOW为准。本SOW将作为项目执行过程控制和管理的指导文件。

## 客户现状

## 项目目标

## 项目范围

### 业务范围

次项目业务范围：仅限于XX有限公司

此项目实施范围：

### 模块范围

# 设计依据与原则

## 项目建设参考标准及规范

我司严格遵循ISO9001和CMMI3认证质量管理系列标准对项目产品进行研发及质量管理，并严格遵循计算机软件等相关规范。本项目建设方案编制的主要依据和标准规范是：

* 《国家信息化领导小组关于我国电子政务建设指导意见》；
* 《计算机软件需求说明编制指南》
* 《软件设计和开发规范》；
* 《[计算机软件测试文档编制规范](http://www.csres.com/detail/190699.html)》
* 《计算机信息系统安全保护等级划分准则》（GB17859-1999）
* 《信息化工程监理规范》（GB/T 19668）
* 《信息安全技术 入侵检测系统技术要求和测试评价方法》（GB/T 20275-2006）
* 《信息安全技术 网络脆弱性扫描产品技术要求》（GB/T 20278-2006）
* 《信息安全技术 网络脆弱性扫描产品测试评价方法》（GB/T 20280-2006）
* 《云计算综合标准化体系建设指南》
* 《ISO9001质量管理体系文件》（GB/T19001-2016）
* 《CMMI3 管理体系》
* 《OPC技术规范》（2014）
* 《计算机软件文档编制规范》（GB/T 8567-2006）
* 《计算机软件需求说明编制指南》（GB/T 9385-1988）
* 《计算机软件需求规格说明规范》（GB/T 9385-2008）
* 《计算机软件测试文件编制规范》（GB/T 9386-1988）
* 《计算机软件质量保证计划规范》（GB/T 12504-1990）
* 《计算机软件单元测试》（GB/T 15532-1995）
* 《信息处理 数据流程图、程序流程图、系统流程图、程序网络图和系统资源图的文件编制符号及约定》（GB/T 1526-1989）
* 《软件工程术语》（GB/T 11457-1995）
* 《信息技术 软件生存周期过程》（GB/T 8566-2007）

## 设计原则

本项目涉及到系统必须以实用为原则，采用成熟的并且通过实践考研的先进技术和解决方案。充分考虑性价比，以可靠性、安全性、先进性、实用性、标准化和开放性、模块化、网络设备互连性和人性化设计为原则，进行系统设计。

### 可靠性

系统将采用主流的技术和配套设备，并且在实际工程中有成功案例，系统各环节具备故障分析与恢复和容错能力，保证系统可靠性和容错性，使系统能不间断正常运行和有足够的延时来处理系统的故障，确保在发生意外故障和突发事件时，系统保持正常运行，把各种可能的风险降至最低。

### 安全性

在考虑信息资源共享的同时，也考虑信息的保护和隔离，系统在各个层次对访问进行控制，使用健全的备份和恢复策略增强系统的安全性；通过集成相关安全产品和安全服务，构造多层防御的安全保障体系，确保系统安全、高效、可靠运行。

### 可扩展性

任何一个项目系统都会随着时间的推移，需求难免发生变更或新增，所以做好需求的可扩展性也是必要的；需要保证系统后期的扩展，系统应具备良好的输入输出接口，可为各种增值业务提供接口，同时，系统可以进行功能的定制开发，可以实现与其他系统的互联互通。

### 先进性

充分应用先进和成熟的技术，使用主流技术，把科学的管理理念和先进的技术手段紧密结合起来，采用国际或国内目前的先进技术，使系统具有较高的性能，负荷当今技术发展方向，确保系统具有较强的生命力，有长期的使用价值，负荷未来的发展趋势。

### 实用性

配置和设计从系统目标和实际需求出发，吸收行业经验，最大限度的满足整体应用和公用设施管理的各项需求，具体功能要实用有效，充分发挥系统的价值。选择成熟的，经济的优质产品并在系统合理配置和兼容性方面进行充分论证。

### 可维护性

软件工程实践表明，没有缺陷的系统是不存在的。为此，信息系统在设计和开发阶段，应充分考虑系统的可维护性。系统应操作简单，实用性高，具有易操作、易维护的特点，在出现故障时，能得到及时、快速地进行诊断，方便系统维护。

### 标准化和开放性

以标准化和开放性为原则，通过相应的网关软件或接口服务器将各种不同数据接口转换成标准、统一的方式(采用基于TCP／IP的协议，以标准以太网为数据传送媒介)与集成数据服务软件交互数据，数据服务软件对于不同的分系统采用统一的处理方式，并采用遵循国际通用规范的大型数据库，从而与其它应用软件共享数据。系统的开放性设计完全遵循国际主流标准以及工业标准。系统的标准化和开放性为增加新的系统和系统升级提供便利。

### 模块化

严格遵循模块化的结构方式进行开发，系统软件功能模块根据用户的实际需要和控制逻辑来编制，具有对其它系统集成的能力。接口开发兼容性强，界面标准化，规范化。对于各种标准接口(OPC，BACnet，RS485／422／232，ModBus，HART，ProfiNET, ProfiBUS，MQTT, KAFKA等) 和非标接口都能够实现和其它系统的信息交换及通讯协议的转换。

### 网络设备的互连性

系统基于网络实现源数据的采集和分析，保证系统高效、可靠、安全运行。系统的网络结构采用分层次的结构模式，从而满足工程分步实施的要求和很强的信息处理及数据通信的能力。在物理上和逻辑上可以实现相互之间的互连。

### 人性化服务

不同级别的管理人员按照不同的操作权限进行管理。系统的管理界面为用户提供个性化的友好中文界面，同时采用图形方式来显示信息点的状态，以及采用表格填充的方式来编制应用软件，能够通过图形检索方式查看站点内各个层面和不同区域的情况，各个系统的工作情况和设备工作状态。

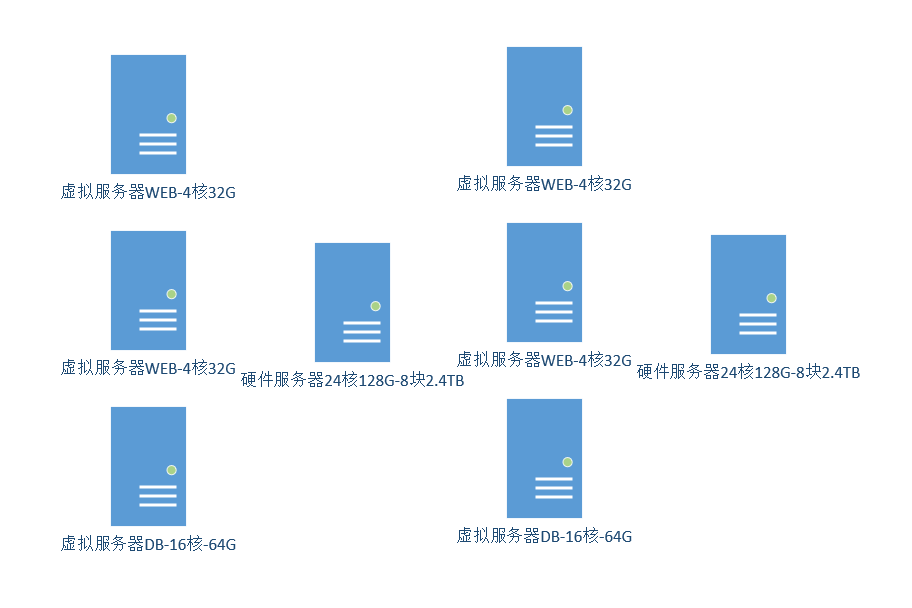
# 系统总体设计

## 总体设计要求

* **部署方式：**
* 物理服务器部署
  + 应用服务器，保证系统运行稳定性、安全性、流畅性、可维护性、可恢复性、并发性、冗余防灾等要求
  + 数据库服务器，保证数据存储的安全性和稳定性和容量等
  + 支持多租户
* **系统运行平台与应用环境：**
* 运行平台：支持Windows、Linux运行环境；
* 应用环境：支持ie9以上浏览器访问
* **架构：**前后台分离，主流框架平台和技术应用
* 后端框架：ABP框架(.Net Core)
* 前端框架: Vue
* **数据库：**
  + 支持MySQL,Oracle,SQLServer关系型数据库；
  + 至少能够存取5年数据；
  + 数据库的实时性、可恢复性、并发操作、一致性、安全性和开放性
* **接口要求：**
  + 系统接口：开放性的接口，用于预留与未来新增系统的信息交互功能
  + 在与办公网进行接口时，设置防火墙，将办公网与设备网分开。
  + 基于SOA架构模式，以服务为导向的接口设计，采用多种接口实现模式以及标准接口技术，提供全方位的接口服务。达到其他应用系统信息共享、协同运作的目的
* **权限管理**
  + 系统需设置不同的用户权限并详细描述各种权限职责的划分；
  + 系统应具有完善的安全和权限管理功能，能够为不同级别的操作人员提供不同的操作权限；
  + 网络用户根据所属权限以网页形式登录系统，直接在本机生成、下载所需数据报表；
  + 所有操作能够自动记录在工作日志数据库中。系统软件设置支持多种权限分区和密码设置，为系统管理员、工程师、管理人员等提供分级密码，并对所有操作自动进行事件记录。
* **用户界面**
  + 友好的人机接口界面
  + 画面信息响应时间不高于2秒，数据刷新时间不高于3秒；
  + 用户界面实现点击每个测量点进入详细页面，能源测量点的详细页面分类显示能耗数据、报警信息、配置信息，并提供实时数据显示，历史数据查询，历史数据趋势图功能；
  + 用户可以调用系统默认的监控画面或进行个性化的图形定义，包括数字显示、刻度盘显示、柱状图显示或趋势图，可以是单图或分层图形；
  + 支持实时监控，通过在其他系统的链接的网页，支持通过Web方式集成多个软件界面；
  + 系统图形化展示应支持放大显示任意时间范围内的能耗数据功能；
  + 系统应支持将能耗图形化界面以灵活方式添加、减少、排序；
  + 系统软件应采用多窗口图形技术，增强图形界面功能，运行操作指导；
  + 画面采用树状结构显示各功能项；画面能体现各能源点的系统图并能在管道上显示主要计量仪表的大体位置；
  + 将最关注的数据或信息调到主界面显示；
  + 画面须显示整个系统的总线系统图，能够体现总线上仪表数据和总线状态；
  + 所有画面要求风格统一，整体画面的架构由投标方规划，并最终由甲方确定；
* **移动端应用：**
  + 支持移动端操作、如安卓
  + 移动端与PC端用户验证和权限一致

## 软件架构

## 服务器网络拓扑结构



服务器架构图

### 软件性能：

* MES系统可以支持5000个以上的在线用户；支持物料种类二十万以上；
* 系统登录界面响应时间≤2秒；流程界面响应时间≤3秒；查询平均响应时间≤2秒；统计平均响应时间≤5秒；其他数据界面响应时间≤3秒；
* 单个订单的自动排产时间在40秒之内；
* 单个工位的报工数据提交在3秒之内；
* 生产进度、质量统计、工时汇总等单个报表统计时间2分钟之内；
* 主机与备份系统的切换时间≤10秒，联机数据备份不造成已完成事务的数据丢失。

### 服务器性能：

* 平均CPU占用率 ≤ 60%；CPU峰值占用率≤ 85%；
* 系统最大占用磁盘控制 ≤ 50%；系统日志最大占用磁盘空间 ≤ 85%；
* 平均内存占用率 ≤ 60%；内存峰值占用率 ≤ 85%；

### 服务器配置：

|  |  |  |
| --- | --- | --- |
|  | Web服务器 | DB服务器 |
| 操作系统 | CentOS | CentOS |
| CPU | 4核 | 16核 |
| 内存 | 32G | 64G |
| 磁盘 | 500G | 1.5TB |

# 系统功能设计

## 系统功能模块及清单

|  |  |  |
| --- | --- | --- |
| **序号** | **模块名称** | **说明** |
| 1 | **模块管理** | 模块增加，更新，删除等功能。 |
| 2 | **角色管理** | 角色增加，更新，删除，配置模块等功能。 |
| 3 | **部门** | 部门增加，更新，删除等功能。 |
| 4 | **用户管理** | 用户增加，更新，删除，密码修改，角色配置等功能。 |
| 5 | **工厂管理** | 工厂增加，更新，删除等功能。 |
| 6 | **车间管理** | 车间增加，更新，删除等功能。 |
| 7 | **设备管理** | 设备增加，更新，删除等功能。 |
| 8 | **产线管理** | 产线增加，更新，删除等功能。 |
| 9 | **工序管理** | 工序增加，更新，删除等功能。 |
| 10 | **工位管理** | 工位增加，更新，删除等功能。 |
| 11 | **打印管理** | 打印机增加，更新，删除，打印模板增加，更新，删除等功能。 |
| 12 | **溯源管理** |  |
| 13 | **条码枪管理** | 条码枪增加，更新，删除功能，需要关联到工位。 |
| 14 | **铸件信息** | 铸件信息通过扫码枪增加，通过每个工序产生不同的铸件工序流程信息。 |
| 15 | **X-RAY** | 条码枪读取信息自动增加，产品序号、属性、检测结果  时间等 |
| 16 | **X-RAY报表展示** | 展示X-RAY工序生产信息进度及合格率，数量。 |
| 17 | **热处理** | 记录热处理信息。 |
| 18 | **机加工信息** | 增加机加工信息。 |
| 19 | **机加工信息报表展示** | 展示机加工实时信息。 |
| 20 | **机加工暂存信息** | 记录机加工暂存信息。 |
| 21 | **机加工暂存展示** | 报表展示，暂存区的信息。 |
| 22 | **He Leak Insp** | 记录 He Leak Insp信息。 |
| 23 | **涂装信息** | 记录涂装信息。 |
| 24 | **待涂装信息** | 记录待涂装信息。 |
| 25 | **粗加工信息** | 记录粗加工信息。 |
| 26 | **精加工信息** | 记录精加工信息。 |
| 27 | **Balance** | 记录Balance信息。 |
| 28 | **包装信息** | 记录包装信息。 |
| 29 | **不包装信息** | 记录不包装信息。 |
| 30 | **PDA** | PDA增加，更新，删除。 |
| 31 | **原料信息** | 铝锭原料信息增加，更新，删除等功能。 |
| 32 | **熔炼信息** | 熔炼信息通过扫码增加，删除等功能。 |
| 33 | **自定义编码功能** | 自动产生唯一，不重复编码功能。 |
| 34 | **通知系统** | 发送通知信息。 |
| 35 | **大屏功能** | 大屏展示信息。 |

# 可靠性、安全性、可扩展性等

## 可靠性解决方案

我方提供的本项目系统具有高可用性，系统选用的技术和配套设备采用主流技术和产品，同时有成功实施案例。本项目系统采用相对独立化分级设计，对系统关键设备考虑冗余，对重要数据进行备份。我方选用高可靠性的终端设备，同提供优质的售后服务，以保证系统长期、可靠的运行。

为达到可靠性、可维护性和可用性，本项目系统种采用如下成熟理念：

|  |  |
| --- | --- |
| 序号 | 采用的成熟理念 |
| 1 | 关键设备冗余方案，对网关进行双机热备，冗余处理，当主设备异常时，备用设备自动启动，保证系统正常运行 |
| 2 | 备份方案，基于云端应用，系统提供从云平台层到虚拟机层的备份能力，包括多种组网架构和丰富的备份功能，解决了因备份需求多样化而导致的复杂性问题。同时，结合虚拟机镜像快照功能，实现应用层级的快速备份和恢复，并针对不同应用要求提供全量和增量等多种备份模式。 |
| 3 | 使用成熟可靠的软件设计方法，如模块化、组件化、面向对象的设计方法等，使用成熟稳定的软件开发技术和平台，Linux操作系统、MS SQL、MySQL、Oracle数据库、.net Core平台框架、OPC、MQTT等； |
| 4 | 在软件系统、框架设计、组件的设计、开发、测试、部署和维护过程中、严格遵循ISO-9001质保程序； |

## 安全性解决方案

我方应XX要求，在项目研发的需求分析阶段、系统框架设计阶段、项目详细设计阶段中充分考虑项目的安全性，我们将结合以往的成功案例中的安全策略以及本项目的安全需求，提供物理安全、数据安全、网络安全、防火墙策略、应用程序安全等技术方案。安全性由网络登录验证、数据库登录验证、应用系统使用验证三级组成。网络登录验证由操作系统完成，用于对具有网络资源访问权限用户的验证；数据库登录验证由数据库服务器完成，用于对具有数据库访问权限用户的验证；系统使用验证由应用系统完成，用于对具有应用系统使用权限用户的验证；应用系统将采用三种验证方式相结合的方式验证用户。

并未满足XX中对系统安全要求，我方郑重承诺满足：

1. 具备权限控制、系统日志记录功能，对进入系统的用户的操作进行记录，包括合法用户的操作和非法用户的尝试性登录；可以根据日志进行事后分析，从而找到事故的发生原因、责任者或非法用户
2. 拥有完整、合理、统一的服务器、数据库、系统文件、数据等各方面的安全控制，建立完整的安全管理机制
3. 提供完整的软件安全策略，包括渗透测试，开源代码管理，提供开源软件评估策略
4. 有完整的数据保护隐私策略：

* 数据传输安全性，为保证数据传输的安全性，使得所传输数据不被盗窃、更改，应用系统所采集的重要原始数据可采用网络加密传输、数据库加密传输或应用系统数据加密相结合的技术。
* 数据存储安全性，重要数据因某种原因需用存储介质进行长期备份存储时，可采用加密算法对数据进行加密，使得非法用户不能理解其含义，当合法用户访问时再将其还原。
* 系统维修时的数据安全性，当系统需要检修或维修时，有可能对系统进行调试，在调试时我们将采用切换到临时运行环境的方法，使系统在调试时与正式存储设备（数据库）隔离，维修结束正式使用时，再将系统与正式存储设备（数据库）相连接。这样就可以保证系统在维修时已有数据的安全。
* 原始数据的安全性，为了保证原始数据的原始性，原始数据一旦保存，便不能被更改；对错误数据只能采取增加一条记录来修正的方式处理，对修正数据应加标志以保证正确性，同时对于修正操作应做数据修正日志，记录修正人相关信息及修正原因等。

保证本项目在实际应用中的安全性，详细请见“第九章 系统安全解决方案”

## 可扩展性解决方案

在系统设计的过程中，最大限度地增强系统的价值，最大限度地吻合业务应用的需求，充分考虑系统今后的硬件扩展、功能扩展、应用扩展、集成扩展等多层面的延伸，整个系统设计过程始终遵循面向数据价值，围绕系统应用，依靠业务部门，注重实效的方针。保证系统的开放、可扩展性，以便提高系统的稳定性和可靠性，满足用户需求不断发展变化的要求，便于应用程序的升级及扩展，减少应用系统再开发（二次开发、定制）的工作量从而降低成本。

下面将从以下几方面论述系统可扩展性：

* 总体架构

总体架构基于 SOA 的思想进行设计。总体架构的搭建能够为各项业务应用提供基础支撑。SOA 是一种架构模型，它可以根据需求通过网络对松散耦合的粗粒度应用组件进行分布式部署、组合和使用。服务层是 SOA 的基础，可以直接被应用调用，从而有效控制系统中与软件代理交互的人为依赖性。SOA 的关键是“服务”的概念，服务是 SOA 架构的基本单元，在 SOA 体系中，每个功能都被封装成一个服务，因此，需要对系统所有的接口与组件服务进行注册管理，以便于与外部系统的整合和流程交互，也便于今后的升级扩展。为对不同系统提供的各类服务进行统一管理，需要建立一套完整的服务注册和集成调用的标准规范，并提供了相应的服务管理平台。

本系统的所有服务接口将按照标准规范进行封装、注册、发布、调用，由服务管理平台统一完成权限认证、消息路由等功能，并基于该标准去调用其它系统或公共组件的服务接口，从整体架构上增强了系统的扩展性。

* 组件化结构：

采用全组件化结构设计，每个组件都被独立地实现，并通过标准接口联系在一起。每个功能组件在功能上独立，同时可根据用户需求灵活配置、组合，实现平滑升级扩容，功能实体可使业务和开发人员根据具体使用要求增加或减少系统应用模块。

* 标准化接口：

采用标准统一的接口设计，所有功能实体间的数据交换以及对其他模块的数据引用都通过标准接口完成，使多个组件对接时在开放性、稳定性、扩展性与集成性上有着很好的适配空间。

* 开放的功能包：

系统处理组件化结构设计与标准化接口设计以支撑开放体系结构外，为了方便用户个性应用的开发，还封装系统及其组件所需的二次开发应用工具包，使其他技术团队对平台进行二次开发时能够更好地复用。

* 分层架构设计：

采用横向分层和纵向分割架构设计。将层与层之间相互分离，每层的应用和服务，采用独立的模块开发和部署，模块间交互标准化，新增功能模块分解到各层，以插件形式加入原系统，既不影响整体架构，也不影响本层功能提供，具备高模块化设计，保证了系统功能的可扩展性。纵向分割是将业务和可复用服务分离出来，通过分布式服务框架调用。新增产品可以通过调用可复用的服务实现自身的业务逻辑，而对现有产品没有任何影响。可复用服务升级变更的时候，也可以通过提供多版本服务队应用实现透明升级，不需要强制应用同步变更。

* 部署和升级的扩展性：

系统采用的软件开发技术都属于开源系统项目，软件系统可运行于通用的主流硬件平台上，不依赖于特定的、专用的硬件设备或者系统软件。系统配置（硬件系统、操作系统、数据库系统）的升级一般情况下，不会引起系统的修改和再次开发，如确实需要对原系统进行必要的升级改动时，由我司负责相应的升级工作，以适应新的硬件平台。

## 先进性解决方案

在项目中应用当今主流技术及性能，采用国际或国内目前的先进技术，并考虑到今后将会出现的新技术，在设计阶段设计具有前瞻性，技术先进性、架构先进性、设计先进性、应用先进性的方案，确保系统在建成后的较长时间内不落后。

**技术先进性**

系统使用.Net Core、Vue、Bootstrap4、TypeScript、MySQL、EF Core等主流框架进行开发

**架构先进性**

系统遵循域驱动设计（DDD），领域驱动包含四个基本层，**表示层**为用户提供界面。使用应用程序层来实现用户交互。**应用层**在表示层和域层之间进行中介。编排业务对象以执行特定的应用程序任务。**域层**包括业务对象及其规则。这是应用程序的核心。**基础结构层**提供通用技术功能，这些功能主要使用第三方库来支持更高的层。

## 实用性解决方案

从实际应用的角度来看，实用性更加重要，为了提高系统的实用性，我司主要以下几个方面进行设计规划：

* + 采取总体设计、分步实施的技术方案，在总体设计的前提下，系统实施中可首先进行业务处理层及管理中的低层管理，稳步向中高层管理及全面自动化过渡，这样做可以使系统始终与用户的实际需求紧密连在一起，不但增加了系统的实用性，而且可使系统建设保持很好的连贯性；
  + 全部人机操作设计均应充分考虑不同用户的实际需要；
  + 用户接口及界面设计将充分考虑人体结构特征及视觉特征进行优化设计；界面尽可能美观大方，操作简便实用。

## 可维护性解决方案

应系统可维护性要求，我司将充分考虑系统的可维护性，使用先进的[软件开发技术](https://baike.baidu.com/item/%E8%BD%AF%E4%BB%B6%E5%BC%80%E5%8F%91%E6%8A%80%E6%9C%AF)和工具。利用先进的软件开发技术和工具是软件开发过程中提高软件质量、降低成本的有效方法之一，也是提高可维护性的有效技术。在软件设计的每个阶段努力提高系统的可维护性，在每个阶段结束前的审查和[复审](https://baike.baidu.com/item/%E5%A4%8D%E5%AE%A1)中．着重对可维护性进行复审。在需求分析阶段的复审中，对将来要扩充和修改的部分加以说明。在讨论软件可移植性问题时．充分考虑可能影响软件维护的系统界面。在软件设计的复审中．从便于修改、[模块化](https://baike.baidu.com/item/%E6%A8%A1%E5%9D%97%E5%8C%96)和功能独立的目标出发．评价软件的结构和过程．从软件质量的角度全面评审总体结构设汁、过程设引和界面设计。同时对将来可能修改的部分预先做准备。

我司将按如下要求设计系统，保证系统的可维护性：

* + “严禁”要求
* 严禁将IP地址、端口等系统参数配置写在代码中
* 严禁将主机域名、IP地址、端口号、文件路径等信息写在代码中，会给开发、测试、部署及运维带来不必要的麻烦，请使用参数文件存放系统配置参数信息。
* 严禁使用直接IP地址对主机进行访问，对主机的访问请通过主机域名进行。
  + “必须”要求
* 必须通过界面修改数据
* 必须记录可审计日志
* 必须对日志输出分级
* 必须具备应用软件版本管理机制
* 必须具备完整的运行时序
* 必须具备完善应用监控和健康性检查手段
* 必须具备完备的备份和归档策略
* 必须具备完备的数据清理机制
* 必须具备应急恢复机制

## 标准化和开放性解决方案

应标准化和开放性要求，我司采用开放性的平台，充分考虑智慧能源管理系统与其它系统的数据接口。根据我司对系统信息化建设应用系统需求和系统目标的分析，实现方法是：快速适应系统的业务需求，应用先进的软件设计思想，同时充分考虑系统长期发展的前瞻性要求，多层 B/S 架构体系之上实现系统的灵活性、安全性，并使系统具有良好的可管理性，重点考虑以下几点：

* + 总体规划、分步实施

系统必须本着“整体规划，统一组织，分步实施”的原则进行开发建设，系统建设应在建设之初的统一规划下，充分考虑以上多方的情况，有机的、分步骤的逐步完善。此外，系统的建设涉及众多新的和复杂的软硬件技术，工程实施环节复杂，我司将按照总体设计的标准规划来进行分步实施。

* + 标准化的开发和设计

系统开发与建设实现工作标准统一、业务流程统一、服务程序统一。在业务、软件产品、通信技术等各方面采用行业、国家和国际标准化组织制定的有关技术规范与标准。保证信息流传递快速顺畅，网络运行安全可靠。

采用标准化规范设计，如提示规范、验证规范、界面规范、控件规范、组件规范，开发代码规范、文档规范等。

* + 标准化的接口

使用通用、标准的协议，通过相应的网关软件或接口协议将各种不同数据接口转换成标准、统一的方式(如采用基于TCP／IP的协议，以标准以太网为数据传送媒介)，快速与其他设备或系统集成。

* + 开放性

采用遵循国际通用规范数据库，提供标准规范的接口，方便与其他系统的实现数据共享和被其他系统调用与集成，同时我方承诺提供应用软件的设计文档、测试文档、操作手册、维护手册、开发说明文档和方案，并提供二次开发服务。

任何新体系的引进都必须保证不能影响原有业务系统的性能，保证关键业务系统的正常运转，这是引进新的信息技术的前提。本系统将充分考虑现状，最大程度地保护用户现有软硬件和网络投资。将准备弃用的原有系统中的数据完整地迁移到新系统中，对保留使用的原有系统进行全面整合，加以充分利用。

## 模块化解决方案

应模块化要求，我司[采用模块化设计](https://baike.baidu.com/item/%E4%BA%A7%E5%93%81%E6%A8%A1%E5%9D%97%E5%8C%96%E8%AE%BE%E8%AE%A1" \t "_blank)，按照自顶向下研究分类，包括系统级模块、产品级模块、组件级模块；再按照功能及加工和组合要求研究分类，包括基本模块、通用模块、专用模块；然后按照接口组合要求研究分类，包括内部接口模块、外部接口模块。通过模块的选择和组合，快速满足功能需求，提高效率，减少成本，同时提升产品质量，方便后期维护。

系统后端模块化设计开发：模块化设计开发，严格遵循模块化的结构方式进行开发，系统软件功能模块根据用户的实际需要和控制逻辑来编制，同时保证对其它系统集成的能力

系统前端：将前端分为Template 模块、JS 模块、CSS 模块三类，独立提供功能，使前端设计标准化，规范化，方便操作，提高效率。

通讯模块：通讯模块集成，支持OPC UA、BACnet，RS485／422／232，ModBus，HART，ProfiNET, ProfiBUS，MQTT, KAFKA等主流标准协议和消息中间件，自定义开发非标接口，保证系统接口兼容性，模块化配置选择通讯协议和消息中间件，保证与其他系统的信息交互和通讯协议的转换。

## 网络设备互连性解决方案

多标准规范的接口支持：集成OPC UA、BACnet，RS485／422／232，ModBus，HART，ProfiNET, ProfiBUS，MQTT等多种通讯协议和消息中间件，保证系统与设备的互联性，系统与系统间的互联性，同时高规范性、高稳定性标准协议的应用，保证了系统高效、可靠、安全运行。

在系统设计时，采用分层次的结构模式，从而满足工程分步实施的要求和很强的信息处理及数据通信的能力。在物理上和逻辑上可以实现相互之间的互连。

## 人性化服务解决方案

切实满足工作快捷有效的符合需要的用户操作流程，采用Vue单页面技术。

* 减少HTTP请求数

用户在打开一个页面的时候，后台程序响应用户所需的时间并不多，用户等待的时间主要花费在下载页面元素上了，即HTML、CSS、JavaScript、图片等，统计显示。所以，想要提高页面打开速度，就要减少HTTP请求数：

* + 减少不必要的HTTP请求，例如用CSS圆角代替圆角图片，减少图片的使用。
  + 对一个页面涉及到的所有零星图片都包含到一张大图中去，这样一来，当访问该页面时，载入的图片就不会像以前那样一幅一幅地慢慢显示出来了。
* 感官体验

感官体验是用户体验中最直接的感受，给用户是呈现视听上的体验，系统的舒适性很关键，用户第一眼看到系统的感受。改善方法：对于系统的调整需要针对系统的目标人群进行分析，然后再在系统的设计细节上进行适当的改善，如系统设计风格、色彩的搭配、页面的布局、页面的大小、图片的展示、系统字体的大小、LOGO的空间等。

* 交互体验的改善

交互体验是呈现给用户操作上的体验，强调易用/可用性。

* 目标信息要醒目而亲近

在关注缩短完成路径这个问题的时候，优化操作步骤是第一位的。

* 目标信息保持更新

在更新系统上内容的时候，不要过于追求量，而要追求质，更新再多的信息，如果不能保证质量的话，这样就等于没有更新。且系统的内容只有经常的更新，才能给人们一种新鲜的感觉，也能满足用户的需求。

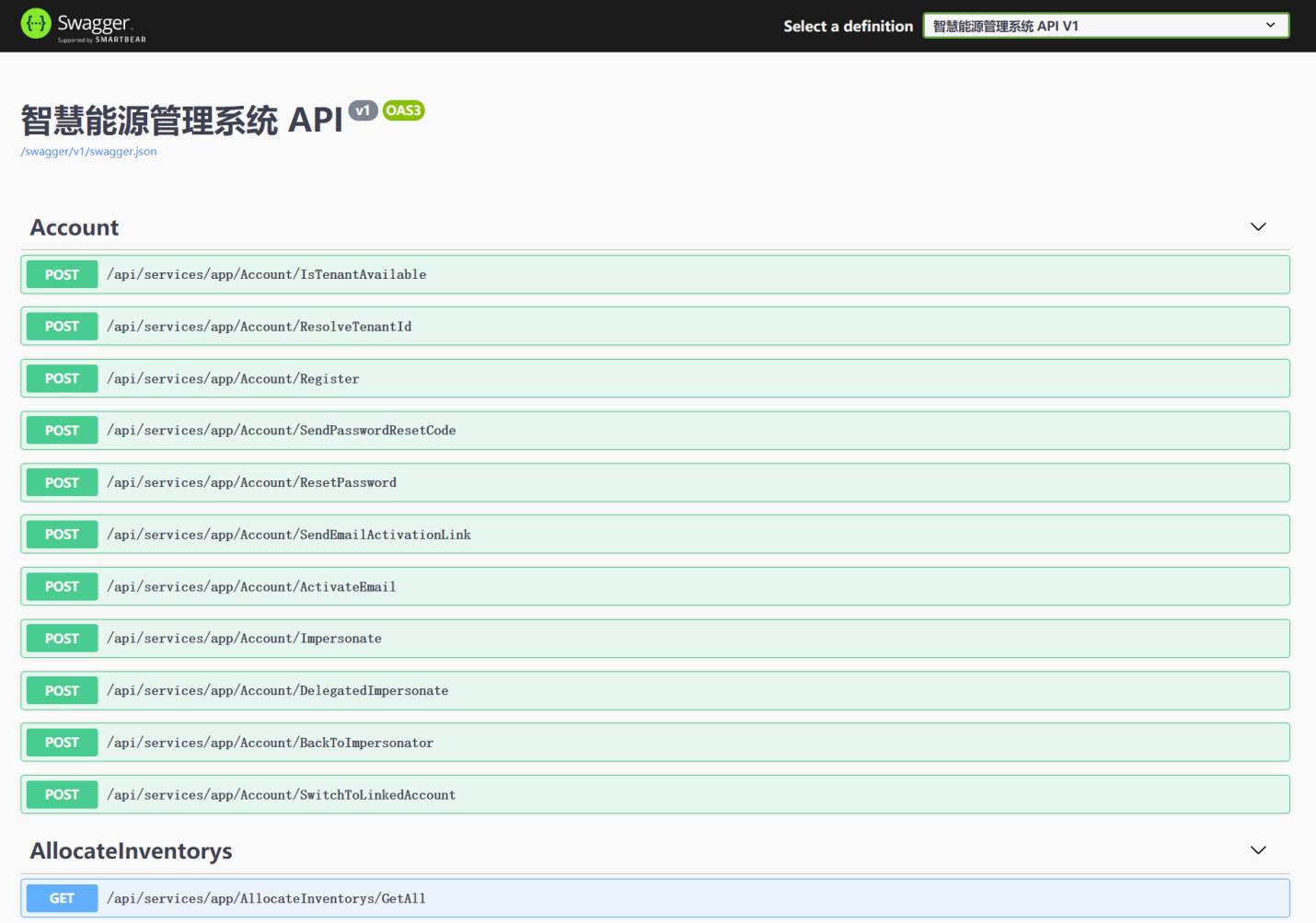
# 系统交互接口

## 系统接口

### 管理、实现方式、方案等

（1） 接口文档管理工具（Swagger）

Swagger 是一个规范且完整的框架，用于生成、描述、调用和可视化 RESTful 风格的 Web 服务。

Swagger 的目标是对 REST API 定义一个标准且和语言无关的接口，可以让人和计算机拥有无须访问源码、文档或网络流量监测就可以发现和理解服务的能力。当通过 Swagger 进行正确定义，用户可以理解远程服务并使用最少实现逻辑与远程服务进行交互。与为底层编程所实现的接口类似，Swagger 消除了调用服务时可能会有的猜测。

（2） 实现方式

安装NuGet软件包，将Swashbuckle.AspNetCore NuGet软件包安装到项目框架的Web项目中，将以下Swagger的配置代码添加到Startup.cs的ConfigureServices方法中。



然后将此代码添加到Startup.cs的Configure方法中，通过访问“/swagger”完成Swagger UI的集成工作。



### 服务接口的预留工作

系统默认接口为基于Https协议的RESTful风格API，在RESTful架构中，每个网址代表一种资源（Resource）。为了兼容其他设备及第三方系统的对接，该项目除了支持默认的RESTful风格接口以外，系统还支持如WebService、WebSocket、MQTT等协议接口。

* 1. WebService

Web Service是一个平台独立的，低耦合的，自包含的、基于可编程的web的应用程序，可使用开放的XML（标准通用标记语言下的一个子集）标准来描述、发布、发现、协调和配置这些应用程序，用于开发分布式的交互操作的应用程序。

* 1. WebSocket

WebSocket是HTML5开始提供的一种浏览器与服务器间进行全双工通讯的网络技术。 WebSocket通信协议于2011年被IETF定为标准 RFC 6455，WebSocketAPI 被 W3C 定为标准。

在 WebSocket API 中，浏览器和服务器只需要要做一个握手的动作，然后，浏览器和服务器之间就形成了一条快速通道。两者之间就直接可以数据互相传送。

* 1. MQTT

MQTT（Message Queuing Telemetry Transport，消息队列遥测传输协议），是一种基于发布/订阅（publish/subscribe）模式的"轻量级"通讯协议，该协议构建于TCP/IP协议上，由IBM在1999年发布。MQTT最大优点在于，可以以极少的代码和有限的带宽，为连接远程设备提供实时可靠的消息服务。作为一种低开销、低带宽占用的即时通讯协议，使其在物联网、小型设备、移动应用等方面有较广泛的应用。

MQTT是一个基于客户端-服务器的消息发布/订阅传输协议。MQTT协议是轻量、简单、开放和易于实现的，这些特点使它适用范围非常广泛。在很多情况下，包括受限的环境中，如：机器与机器（M2M）通信和物联网（IoT）。其在，通过卫星链路通信传感器、偶尔拨号的医疗设备、智能家居、及一些小型化设备中已广泛使用。

### 其他接口的预留工作

由于系统采用的是DDD领域模型架构，框架本身默认支持模块化设计，当遇到其他接口需求时，系统可在不改变现有接口实现的情况下，动态增加新的模块，专门用于其他接口的实现。

## 单点登录的预留工作

**单点登录登出(SSO)**

在各种类型的应用上实现单点登录登出。

**API访问控制**

为各种各样的客户端颁发Access Token令牌,如服务与服务之间的通讯、网站应用、SPAS和本地应用或者移动应用。

# 关键技术解决方案

## 基础技术介绍

### ABP（.Net Core）框架

ABP是ASP.NET Boilerplate Project，是一个开箱即用的框架，可以作为asp.net分布式应用，ABP框架定位于快速开发，是一个用于最快实践和流行开发现代Web应用程序的新起点，旨在成为一个通用的Web应用程序框架和项目模板。 ABP基于DDD的经典分层架构思想，实现了众多DDD的概念。

具有如下优点：

服务端

* ASP.NET MVC和Web API
* 实现领域驱动设计DDD，包括实体、存储库、领域服务、应用程序服务、DTO、UnitOfWork...
* 实现分层架构：领域层、应用程序、表示层、基础结构层
* 提供基础设施，为大型项目开发可重用和可组合的模块，使其依赖注入（使用Castle Windsor作为DI容器）。
* 提供严格的模型和基类，轻松使用对象关系映射ORM（直接支持EF）。
* 支持并实现数据库迁移Migration
* 包括简单灵活的本地化系统，多语言。
* 包含服务端全局领域事件和EventBus
* 管理异常处理和验证
* 为应用程序服务创建动态WebApi层
* 提供基本和辅助类实现常见任务
* 使用约定优于配置的原则

客户端

* 提供项目模板为单页应用程序SPA和多页应用程序MPA，模板基于Bootstrap。
* 常用jQuery和AngularJS库
* 创建动态JS代理以轻松调用应用程序服务，使用动态Web API层。
* 包含独特的任务API

### Web服务技术

Web API可以构建面向各种客户端的服务，可以对接各种客户端（浏览器，移动设备），构建http服务的框架。

Web API的主要功能：

* + 支持基于Http verb (GET, POST, PUT, DELETE)的CRUD (create, retrieve, update, delete)操作通过不同的http动作表达不同的含义，这样就不需要暴露多个API来支持这些基本操作。
  + 请求的回复通过Http Status Code表达不同含义，并且客户端可以通过Accept header来与服务器协商格式，例如你希望服务器返回JSON格式还是XML格式。
  + 请求的回复格式支持 JSON，XML，并且可以扩展添加其他格式。
  + 原生支持OData。
  + 支持Self-host或者IIS host。

### Vue

Vue 是一款十分流行且好用的 Web 前端框架,具有如下特性：

* 跨平台:

渐进式 Web 应用，借助现代化 Web 平台的力量。高性能、离线化、零安装。

原生，借助来自 Ionic、NativeScript 和 React Native 中的技术与思想，构建原生移动应用。

桌面，借助你已经在 Web 开发中学过的能力，结合访问原生操作系统 API 的能力，创造能在桌面环境下安装的应用，横跨 Mac、Windows 和 Linux 平台。

* 速度与性能:

代码生成，Vue 会把你的模板转换成代码，针对现代 JavaScript 虚拟机进行高度优化，轻松获得框架提供的高生产率，同时又能保留所有手写代码的优点。

统一，在服务端渲染应用的首屏，像只有 HTML 和 CSS 的页面那样几乎瞬间展现，支持 node.js、.NET、PHP，以及其它服务器，为通过 SEO 来优化站点铺平了道路。

代码拆分，Angular 应用通过新的组件路由（Component Router）模块实现快速加载，提供了自动拆分代码的功能，为用户单独加载它们请求的视图中需要的那部分代码。

* 生产率

模板，通过简单而强大的模板语法，快速创建 UI 视图。

Vue 命令行工具，命令行工具：快速进入构建环节、添加组件和测试，然后立即部署。

各种 IDE，在常用 IDE 和编辑器中获得智能代码补全、实时错误反馈及其它反馈等特性。

### Redis

Remote Dictionary Server(Redis) 是一个由Salvatore Sanfilippo写的key-value存储系统。Redis 是一个高性能的key-value数据库，开源，遵守BSD协议、支持网络、可基于内存亦可持久化的日志型，并提供多种语言的API。

Redis的特点：

* 内存数据库，速度快，也支持数据的持久化，可以将内存中的数据保存在磁盘中，重启的时候可以再次加载进行使用。
* Redis不仅仅支持简单的key-value类型的数据，同时还提供list，set，zset，hash等数据结构的存储。
* Redis支持数据的备份，即master-slave模式的数据备份。
* 支持事务

### MQTT

MQTT(消息队列遥测传输)是ISO 标准(ISO/IEC PRF 20922)下基于发布/订阅范式的消息协议。MQTT协议是为大量计算能力有限，且工作在低带宽、不可靠的网络的远程传感器和控制设备通讯而设计的协议，它具有以下主要的几项特性：

* 使用发布/订阅消息模式，提供一对多的消息发布，解除应用程序耦合；
* 对负载内容屏蔽的消息传输；
* 使用 TCP/IP 提供网络连接；
* 有三种消息发布服务质量：
* “至多一次”，消息发布完全依赖底层 TCP/IP 网络。会发生消息丢失或重复。这一级别可用于如下情况，环境传感器数据，丢失一次读记录无所谓，因为不久后还会有第二次发送。
* “至少一次”，确保消息到达，但消息重复可能会发生。
* “只有一次”，确保消息到达一次。这一级别可用于如下情况，在计费系统中，消息重复或丢失会导致不正确的结果。
* 小型传输，开销很小（固定长度的头部是 2 字节），协议交换最小化，以降低网络流量；
* 使用 Last Will 和 Testament 特性通知有关各方客户端异常中断的机制。

### KAFKA

Kafka [1]  是一种高吞吐量 [2]  的分布式发布订阅消息系统，有如下特性：

* 通过O(1)的磁盘数据结构提供消息的持久化，这种结构对于即使数以TB的消息存储也能够保持长时间的稳定性能。
* 高吞吐量 [2]  ：即使是非常普通的硬件Kafka也可以支持每秒数百万 [2]  的消息。
* 支持通过Kafka服务器和消费机集群来分区消息。
* 支持[Hadoop](https://baike.baidu.com/item/Hadoop" \t "_blank)并行数据加载。

### Docker

Docker 是一个开源的应用容器引擎，让开发者可以打包他们的应用以及依赖包到一个可移植的容器中,然后发布到任何流行的[Linux](https://baike.baidu.com/item/Linux" \t "_blank)机器或Windows 机器上,也可以实现虚拟化,容器是完全使用沙箱机制,相互之间不会有任何接口。具有如下优势：

* 更高效的利用系统资源

docker对系统资源的利用率更高，无论是应用执行速度，内存损耗或者文件存储速度，都要比传统虚拟机技术更高效。因此，相比虚拟机技术，一个相同配置的主机往往可以运行更多数量的应用。

* 更快速的启动时间

传统的虚拟机技术启动应用服务往往需要数分钟，而docker容器应用，由于直接运行于宿主内核，无需启动完整的操作系统，因此可以做到秒级，甚至毫秒级的启动时间，大大的节约了开发测试，部署的时间。

* 一致的运行环境

开发过程中常见的一个问题是环境一致问题，由于开发环境，测试环境，生产环境不一致，导致有些bug并未在开发过程中发现。而docker的镜像提供了除内核外完整的运行时环境，确保环境一致性，从而不会在出现“这段代码在我机器上没问题”这类问题。

* 持续支付和部署

对开发和运维人员来说，最希望就是一次创建和部署，可以在任意的地方运行。（定制应用镜像来实现集成、持续支付、部署。开发人员可以通过dockerfile来进行镜像构建，并结合持续集成系统进行集成测试，而运维人员则可以直接在生产环境中快速部署该镜像，甚至结合持续部署系统进行自动部署）。而且使用dockerfile使镜像构建透明化，不仅仅开发团队可以理解应用运行环境，也方便运维团队理解应用运行所需条件，帮助更好的生产环境中部署该镜像。

* 更轻松的迁移

由于docker确保了执行环境的一致性，使得应用的迁移更加的容易。docker可以在很多平台上运行，无论是物理机、虚拟机、公有云、私有云、甚至是笔记本、其运行结果是一致的。因此用户可以很轻易的将在一个平台上运行的应用，迁移到另一个平台上，而不用担心运行环境的变化导致应用无法正常运行的情况。

* 更轻松的维护和拓展

docker使用的分层存储以及镜像的技术，使得应用重复部分的复用更为容易，也使得应用的维护更新更加简单，基于基础镜像进一步扩展镜像也变得十分简单。此外，docker团队同各个开源项目团队一起维护了一大批高质量的官网镜像，既可以直接在生产环境使用，又可以作为基础进一步定制，大大的降低了应用服务的镜像制作成本。

### 微服务

微服务架构是一种架构模式或者说是一种架构风格。它提倡将单一应用程序划分成一组小的服务，每个服务运行独立的自己的进程中，服务之间互相协调、互相配合，为用户提供最终价值。服务之间采用轻量级的通信机制互相沟通（通常是基于 HTTP 的 RESTful API) 。每个服务都围绕着具体业务进行构建，并且能够被独立地部署到生产环境、类生产环境等。

微服务有如下优点：

* 微服务是松藕合的，无论是在开发阶段或部署阶段都是独立的。
* 能够快速响应, 局部修改容易, 一个服务出现问题不会影响整个应用。
* 易于和第三方应用系统集成, 支持使用不同的语言开发, 允许你利用融合最新技术。
* 每个微服务都很小，足够内聚，足够小，代码容易理解。团队能够更关注自己的工作成果, 聚焦指定的业务功能或业务需求。
* 开发简单、开发效率提高，一个服务可能就是专一的只干一件事, 能够被小团队单独开发，这个小团队可以是 2 到 5 人的开发人员组成

## 兼容性解决方案

我方在本项目的兼容性将做好大量工作，从开发平台选型、运行平台支持、数据库、应用环节等方便，保证系统的兼容性，包括以下内容：

开发平台：.Net Core

运行平台：支持Window、Linux

数据库：支持Oracle、MS SQL Server、MySQL

应用环境：兼容Internet Explorer 6.0或更高版本、Firefox等

## 高性能高并发解决方案

应客户文件中对高性能的需求，我方在本项目中使用高性能的架构，保证系统的高性能和高并发，以用户为中心，提供快速的处理能力和访问体验，主要参数有较短的响应时间、较大的并发处理能力、较高的吞吐量与稳定的性能参数。

主要分为前端优化、浏览器优化、应用层优化、代码层优化与存储层优化：

* 前端优化：系统业务逻辑前端的部分。
* 浏览器优化：减少 HTTP 请求数，使用浏览器缓存，启用压缩，CSS JS 位置，JS 异步，减少 Cookie 传输；CDN 加速，反向代理。
* 应用层优化：处理系统业务的服务器。使用缓存（如Redis），异步，集群等方式，消息队列（如KAFKA、Rabbit MQ、MQTT等）。
* 代码优化：合理的架构，多线程，资源复用（对象池，线程池等），良好的数据结构，Cache 等。
* 存储优化：缓存、固态硬盘、光纤传输、优化读写、磁盘冗余、分布式存储（HDFS）、NoSQL 等。

### 安全机制

1. 物理安全

物理安全主要实现以下策略： 设备安全策略、通讯线路安全策略、环境安全策略、物理隔离策略等

1. 系统安全

系统安全主要实现以下策略：身份鉴别策略、访问控制策略、安全审计策略、通讯完整性与保密性策略、抗抵赖策略、剩余资源保护和控制策略、软件容错策略、入侵检测策略以及恶意代码防范策略等

## 系统平台安全

### 身份鉴别

**基本要求**

* 应提供专用的登录控制模块对登录用户进行身份标识和鉴别；
* 应提供登录失败处理功能，可采取结束会话、限制非法登录次数和自动退出等措施；
* 应启用身份鉴别、用户身份标识唯一性检查、用户身份鉴别信息复杂度检查以及登录失败处理功能，并根据安全策略配置相关参数；
* 应提供用户身份标识唯一和鉴别信息复杂度检查功能，保证应用系统中不存在重复用户身份标识，身份鉴别信息不易被冒用；
* 应对同一用户采用两种或两种以上组合的鉴别技术实现用户身份鉴别。

**实现方式**

* 系统内的每个角色有可靠的身份识别，访问权限控制，防止对信息的非法使用、调阅、修改和破坏。
* 信息在存储和传送过程中采取加密保护。
* 具有防非法修改和防静态分析等自身防护功能。
* 采取应用系统运行日志、应用系统操作日志、系统运行监控和故障报警等手段，加强对系统运行进行监控
* 提供专用的登录控制模块对登录用户进行身份标识和鉴别。包括登录认证、长时间未操作提示、错误登录提示等。利用SESSION技术实现当其空闲操作的时间超过规定值(通常为l0min以内)后，在该用户需要执行其它操作之前，对该用户重新进行身份鉴别，要求重新登录。系统提醒SESSION（空闲操作的时间超过规定值）时间，并自动限制用户设置时间在l0min以内。

### 访问控制

**基本要求：**

* 应提供访问控制功能控制用户组/用户对系统功能和用户数据的访问；
* 应由授权主体配置访问控制策略，并严格限制默认用户的访问权限；
* 应提供访问控制功能，依据安全策略控制用户对文件、数据库表等客体的访问；
* 访问控制的覆盖范围应包括与资源访问相关的主体、客体及它们之间的操作；
* 应授予不同帐户为完成各自承担任务所需的最小权限，并在它们之间形成相互制约的关系；
* 应具有对重要信息资源设置敏感标记的功能；
* 应依据安全策略严格控制用户对有敏感标记重要信息资源的操作。

**实现方式：**

* 提供访问控制功能控制用户组/用户对系统功能和用户数据的访问
* 由授权主体配置访问控制策略，并严格限制默认用户的访问权限，系统提供系统数据是否涉密属性设置功能，并提供强制访问涉密数据控制策略，包括无、只读、读写。
* 提供访问控制功能，依据安全策略控制用户对文件、数据库表等客体的访问。
* 访问控制的覆盖范围包括与资源访问相关的主体、客体及它们之间的操作；
* 授予不同帐户为完成各自承担任务所需的最小权限，并在它们之间形成相互制约的关系；利用应用支撑平台的统一用户管理服务，可以实现角色的设置及角色权限授予、回收。系统默认拥有系统管理人员（系统管理员、安全保密管理员、安全审计员），系统用户（重要、一般等）三类角色。

### 安全审计设计

**基本要求：**

* 应提供覆盖到每个用户的安全审计功能，对应用系统重要安全事件进行审计；
* 应保证无法单独中断审计进程，无法删除、修改或覆盖审计记录；
* 审计记录的内容至少应包括事件的日期、时间、发起者信息、类型、描述和结果等；
* 应提供对审计记录数据进行统计、查询、分析及生成审计报表的功能。

**实现方式：**

* 通过部署网络与数据库审计系统，并结合集中安全管理平台审计功能，综合实现物理安全审计要求。
* 安全审计范围至少覆盖应用中间件、数据库及系统本身的身份鉴别、访问控制、数据完整性、数据库维护等功能。审计记录信息包括被审计对象、时间、审计人、具体操作，能对对安全事件的事后追查提供足够的信息。
* 安全审计与身份鉴别、访问控制、数据完整性等安全功能的设计紧密结合，并为下述可审计事件产生审计记录：
* 审计功能的启动和关闭：记录审计功能的启动和关闭时间、操作人员、是否成功、故障编号等；
* 系统内用户增加、删除：记录用户增加、删除时间、操作人、操作是否成功、故障编号等；
* 用户权限的更改：记录用户权限更改的时间、更改人、更改值、是否成功、故障编号等；
* 系统管理员、系统安全员、审计员和一般操作员所实施的操作：记录操作员类型、操作员编号、操作、操作时间、是否成功、故障编号等；
* 其它与系统安全有关的事件或专门定义的可审计事件：系统提供审计对象和审记事件的定义，以及审计记录信息的定义，并提供标准的开发接口，便于扩展开发。
* 审记系统采用松偶合设计技术，数据与采集功能独立，当审计系统出现异常时，存储的审计记录不会受到影响。系统审计记录保存周期默认大于等于六个月（用户可以自行设置，但系统会限制其设置周期），在小于六个月时，系统不提供用户对审计记录的清空和删除功能。

### 通讯完整性、保密性

**基本要求：**

* 应采用约定通信会话方式的方法保证通信过程中数据的完整性；
* 应采用密码技术保证通信过程中数据的完整性；
* 在通信双方建立连接之前，应用系统应利用密码技术进行会话初始化验证；
* 应对通信过程中的整个报文或会话过程进行加密。
* 实现方式
* 采用约定通信会话方式的方法保证通信过程中数据的完整性
* 采用MD5单向加密密码技术保证通信过程中数据的完整性
* 通信双方建立连接之前，应用系统利用密码技术进行会话初始化验证
* 对通信过程中的整个报文或会话过程采用MD5单向加密密码技术进行加密

### 软件容错

**基本要求：**

* 在故障发生时，应用系统应能够继续提供一部分功能，确保能够实施必要的措施；
* 应提供数据有效性检验功能，保证通过人机接口输入或通过通信接口输入的数据格式或长度符合系统设定要求；
* 应提供自动保护功能，当故障发生时自动保护当前所有状态，保证系统能够进行恢复。
* **实现方式：**
* 在故障发生时，应用系统能够继续提供一部分功能，如非登录用户的公共功能。能够实施必要的措施
* 提供数据有效性检验功能，保证通过人机接口输入或通过通信接口输入的数据格式或长度符合系统设定要求；
* 采用集群技术和备份技术，提供自动保护功能，当故障发生时自动保护当前所有状态，保证系统能够进行恢复。
* 软件程序应当能够自动保护当前所有状态，保证系统能够进行恢复，例如记录故障发生时的系统状态功能（系统状态的快照），便于事后分析故障原因。
* 在程序开发时通过安全编程实现。

# 项目实施

凌世实施团队具备专业的实施经验，确保项目能够按时保质交付。在项目实施过程中提交清晰的实施计划和步骤，严格遵守和跟进，保持定期的客户沟通；现场实施期间，项目经理需要每天参加进度早会, 撰写日报，周报，月报，与客户团队紧密合作与沟通；任何偏离计划的异常事项，项目经理需当天出具报告，并向客户管理团队汇报改进措施；详细的需求设计方案文档；任何需要定制化开发的功能，需与用户业务部门详细了解需求，并撰写开发设计方案，并通过用户审核后方可开始；采用Jira敏捷项目管理根据，代码管理采用TFS管理版本，保证代码一致性和完整性；严格的测试计划以及灾难恢复机制，以确保每一个功能的完善和可靠。

根据客户要求，该项目开发及实施周期比较短，系统涉及的功能模块与业务领域相对较广。我们假设基础建设、网络部署等客观条件都符合既定计划进度，特制定了如下项目实施计划。如果在项目实施阶段，由于非凌世原因的其他条件（包括以上假设描述，但不限于这些假设）而导致系统的开发与实施有所变更，双方根据实际情况协商解决，但是凌世不对因此而产生的项目进度变更承担责任和任何损失。

## 标准规范的方法论

成功的项目实施是一个复杂的工程问题，存在一定风险；为了最大程度降低风险，凌世在长期的项目实践中总结出一套能避免或把风险降为最低的方法和工具，规范项目管理，保证分析人员、设计人员、开发人员、实施人员等能够有条不紊的完成各自工作。

凌世针对软件产品有一套标准规范的方法论，将整个项目过程中项目管理、软件开发、质量管理、变更管理、安全管理等从流程到模板工具紧密关联起来，保证项目成功实施和高质量的交付

### 项目管理流程

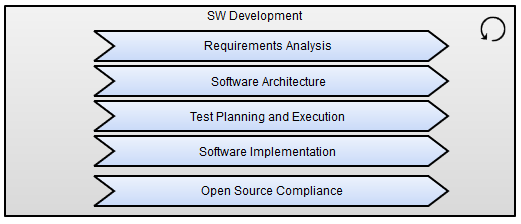
#### 100项目管理：

项目管理的目标是根据项目的要求和约束，确定、建立和控制项目所需的活动和资源，其主要内容包括项目启动与计划、项目汇报、风险管理、经验教训。

|  |
| --- |
| **过程结果** |
| 1. 定义和描述了项目的工作范围（项目章程/项目管理计划） 2. 对完成工作所需的活动和资源进行定义和估计 3. 确定并监控项目内部以及与其他项目和组织单位的接口（会议概述） 4. 对项目进度进行监控和报告（项目报告、质量检查） 5. 识别和分析项目风险。（风险表） 6. 执行流程裁剪 |
| **项目启动** |
| 1. 任命项目经理/产品负责人/项目团队 2. 在Confluence/JIRA/Sharepoint中建立项目 |

#### 200软件开发：

软件开发主要包括需求分析、架构设计、测试计划与执行、软件开发实现和开源法规的遵守。

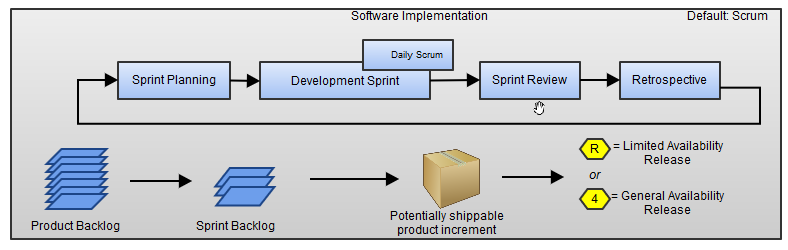
****

* 需求管理
* 使用敏捷管理工具管理和追踪需求（例如Track & Release或Team Foundation Server）
* 按要求标记问题
* 描述情景和期望
* 定义可测试验收标准
* 需求优先排序，将需求计划分解到每个软件版本中
* 需求确认，与相关方确认需求
* 定义质量要求并确定其优先级
* 架构与设计

架构与设计主要描述了产品的信息和目标，说明了系统的架构约束、系统范围和详细内容、解决方案、运行环境、部署环境、质量方案和技术风险等。

* 软件开发实现

软件实现的目标是生成正确反映软件体系结构的经过验证的软件单元。



* 测试计划与执行

在制定测试计划时，明确定义测试的范围，并说明所做的所有假设。在测试开始前确定所需的任何其他软件，并说明用于确定此测试级别何时完成的完成标准，测试管理计划支持以下具体目标：

* 确定相关项目应测试的项目（高级别）
* 确定并描述将用于覆盖测试需求的测试策略
* 确定所需的资源，并提供测试工作的高水平估计
* 列出测试活动期间将提供的可交付成果
* 列出主要的测试活动

#### 300质量管理与保证

质量保证流程的目标是提供独立和客观的保证，确保产品和流程符合预定义的流程和指南，并解决和进一步防止不符合项，主要包括软件质量管理和项目具体质量保证计划。

过程结果：

1. 开发、实施和维护阶段执行质量保证的策略
2. 质量保证独立客观地进行，不存在利益冲突
3. 对产品、过程与活动的相关要求进行验证、记录并传达给相关方。
4. 将不符合项升级到适当管理级别。

#### 400变更和问题解决管理

主要内容：

* 事件管理

事件管理是负责管理所有事件生命周期的过程。事件管理确保尽快恢复正常的服务操作，并将业务影响降至最低，事件是IT服务的意外中断（备注：链断了）或IT服务质量的降低，尚未影响服务的配置项发生故障也是一个事件。主要内容包括：

* 事件定义与区分
* 如何报告和处理事件
* 事件管理流程
* 问题管理

负责管理所有问题的生命周期的过程。问题管理积极预防事件的发生，并将无法预防的事件的影响降到最低，问题是一个或多个事件的原因。在创建问题记录时，通常不知道原因，问题管理流程负责进一步调查。主要包括：

* 问题定义与区分
* 如何报告和处理事件

#### 500配置管理

* 配置管理计划

配置管理计划的目标是建立和维护项目或产品的所有工作的完整性，并将其提供给相关方。

包括以下内容：

* + - 定义项目中定义的基线策略
    - 定义项目中使用的分支策略
    - 版本控制方案
    - 构建、部署和发布
    - 命名约定

#### 600信息安全

信息安全主要包括安全相关声明、安全风险评估、安全概念和安全风险清单。

* 安全相关声明

定义安全性要求，产品的功能必须提供安全性，以便开发商、运营商等能够满足其安全性要求。

* 安全风险评估

信息安全风险评估（信息安全威胁和风险分析）检查系统的开发环境和运行系统，识别需要保护的所有资产的安全相关威胁。风险分析部分评估所有已识别是威胁和相关处理后剩余的风险，并确定必须满足的要求，以便在可接受的风险水平上开发和运行系统。

* 安全概念

关于安全性的一系列要求是安全概念的核心部分。

* + - 确定了应该在代码中或在部署环境中为自己和外部基础结构组件（系统强化）实现的安全措施的需求
    - 详细说明了如何保护开发环境和部署管道，以及对管理和操作过程的描述
    - 详细说明在其生命周期结束时安全退役的问题。
* 安全风险清单

安全风险清单包含与安全相关的所有技术风险。什么是安全风险，这是指项目的一个组件可能易受攻击或没有可用的缓解措施的漏洞。还包括尚未实现或仅部分实现且非常重要的安全需求。另请参见安全风险列表。

#### 700数据保护

数据保护模块中详细定义了在项目中何时定义、评估、检查和审查相关数据保护内容、文档以及其流程，说明了与工程项目中的数据保护和信息安全合作伙伴（DSPiE）的关系，其主要内容包括流程说明、数据保护影响评估、数据保护检查表、客户信息传单-处理器、用户信息传单-控制器和数据保护审查。

## 敏捷的项目管理工具

自20世纪90年代兴起的敏捷软件开发方法，以其高效和灵活的特点为业界接受。敏捷强调坐在一起，共同工作，这就必须要求有相关项目管理工具的支持。JIRA和DevOps很好的支持了项目管和软件开发过程中敏捷的需求。凌世应用JIRA和DevOps等工具掌控整个项目过程，提高工作了效率，同时保证了产品的质量。

### JIRA

JIRA是项目与事务跟踪工具，是一款敏捷的项目管理工具，被广泛应用于缺陷跟踪、客户服务、需求收集、流程审批、任务跟踪、项目跟踪和敏捷管理等工作领域，在实际工作中使工作效率提高很多，可以随时了解问题和项目的进展情况。

主要功能如下：

* 问题追踪和管理：用它管理项目，跟踪任务、bug、需求，通过jira的邮件通知功能进行协作通知
* 问题跟进情况的分析报告：可以随时了解问题和项目的进展情况
* 项目类别管理功能：可以将相关的项目分组管理
* 组件/模块负责人功能：可以将项目的不同组件/模块指派相应的负责人，来处理所负责的组件的Issues
* 项目email地址功能：每个项目可以有不同的email（该项目的通知邮件从该地址发出）
* 无限制的工作流：可以创建多个工作流为不同的项目使用

### DevOps

DevOps（Development和Operations的组合词）是一组过程、方法与系统的统称，用于促进开发（应用程序/软件工程）、技术运营和质量保障（QA）部门之间的沟通、协作与整合。它是一种重视“软件开发人员（Dev）”和“IT运维技术人员（Ops）”之间沟通合作的文化、运动或惯例。透过自动化“软件交付”和“架构变更”的流程，来使得构建、测试、发布软件能够更加地快捷、频繁和可靠。DevOps具有如下优点：

* 可预测性: DevOps大大降低了新版本的意外失败率。
* 可重现性: 对所有内容进行版本化，以便可以随时恢复早期版本。
* 可维护性: 在新版本崩溃或有问题的情况下，轻松回滚。
* 上市时间: 通过简化交付过程，DevOps将上线时间减少了50%，在线应用程序尤其如此。
* 更高的质量: 由于和运维团队紧密合作，产品问题尽早暴露，DevOps帮助团队提高开发质量。
* 降低风险: DevOps在软件交付生命周期中包含了风险控制方面的内容，有助于降低项目研发风险。
* 弹性: 软件系统的运行状态更加稳定、安全，并且变更是可审查的。
* 降低成本: DevOps在软件开发过程中提高了效率，降低了研发成本。
* 代码拆分: DevOps基于敏捷编程方法，因此，它允许将较大的代码库分解为较小的、可管理的模块。

## 项目实施计划

## 项目组织架构及工作说明

### 项目组织架构

本项目组织与主要人员图标如下，待项目启动会议(Kick-off meeting)前确认组织成员，所有项目关键人员皆由经验丰富的顾问专家出任。

### 工作说明

项目组织中各成员的工作说明如下：

|  |  |  |
| --- | --- | --- |
| **组别** | **人员组成** | **职责描述** |
| 项目管理组 | 项目经理 | 负责对项目建设过程中的一切技术管理、内外部资源协调、部门和人员关系协调、重大问题的决策。  负责项目实施各阶段的各项计划的制定；监督项目进度。  项目变更管理。 |
| 主任工程师 | 负责项目建设过程中的技术方案、技术管理与支持。  负责项目实施各阶段的技术问题。  复审设计文档。  负责项目建设过程中的技术指导。 |
| 研发团队 | 需求分析师 | 负责项目的需求分析，架构设计，编制项目系统开发中涉及到的外部接口细则，落实系统外部接口的实施，协调各系统间的接口，应用软件的研制、开发、调试、手册编写、程序维护等。 |
| 系统架构师 |
| 开发工程师 |
| 测试工程师 | 负责系统各项测试环节的测试方案编写及具体测试工作实施 |
| 实施团队 | 实施工程师 | 负责硬件设备、系统软件（如数据库、系统等）的现场安装和测试、用户上线操作指导，并提交现场测试报告和用户使用情况报告。 |
| 质量团队 | 质量工程师 | 负责项目质量管理计划制定和项目质量控制 |
| 服务支持团队 | 系统运维工程师 | 负责有关文档的整理编辑，系统培训组织，系统平台售后运维等。 |

## 实施过程及责任

### 系统需求分析

**目标****：**

通过需求收集，准确的挖掘出XX的真实需求并形成基准，以便指导后续的系统设计和开发，从而确保项目的成功。

**工作内容：**

* 协调组织召开最终用户需求会议，获取各部门的需求；协调让各部门提供相关资料。
* 凌世负责根据XX各部门主要人员所提出的需求形成需求文件。

**交付交付物及验收标准：**

* 提交需求收集文件, 签收

### 系统功能设计

**目标：**

* 在已经形成的需求基准基础上，通过凌世的业务及技术专家设计出方便，适合于XX的系统功能。

**工作内容：**

* 根据用户需求、进行功能设计，UI等各方面的设计。
* 将设计结果形成资料文档

**交付物及验收标准：**

* 提交系统设计，功能设计文档。
* 签收文件。

### 系统功能开发

**目标**

* 基于前期的功能设计，凌世的进行相关软件的功能进行开发，以满足要求。

**工作内容**

* 根据设计文档进行功能开发，完善软件功能。
* 开发过程中对新增功能，以及关联功能进行测试。

**交付物及验收标准**

* 完成开发工作，提供可供测试的软件系统。

### 系统测试

**目标：**

* 通过全方位的功能测试，集成测试，压力的测试，确认系统上线适合使用，无重大BUG，系统能够稳定的运行。

**工作内容**

* 编写阶段测试计划，包括测试时间，测试各方参与人员，测试目标。
* 编写各种测试用例：功能测试用例，集成测试用例，压力测试用例。
* XX对各种测试用例进行确认可行后方按测试计划执行。
* 测试后需要根据测试用例记录测试结果，并且需评估测试结果是否符合预期，符合XX的要求。
* 对于测试不符合要求的，需要多方重新检讨直到测试通过。

### 产品交付验收

**目标**

* 凌世确保报工管理系统所有功能按要求完成并成功上线运行。

工作内容

* 报工管理系统安装部署到服务器且可运行。
* 报工管理系统系统的用户培训：系统管理培训，用户操作培训。

**交付物及验收标准**

* 系统操作守则。
* 系统安装及部署文档。
* 系统维护文档

### 上线期维护服务

在项目上线期间凌世将提供维护服务。

* 维护开始：每一次交付验收。
* 维护停止：最终一次交付验收。

服务：

1. 一般服务：当发生三级和四级状况时, 可于正常上班时间(AM 9:00 ~ PM 5:00)，  
   提供(5\*8小时)On-Call服务
2. 紧急服务：当发生一级和二级状况时, 提供紧急服务
   1. 紧急咨询：提供7\* 24小时服务
   2. 到点服务：提供到公司或工位服务
   3. 如有紧急事件影响生产活动，将于半小时内回复讯息, 4小时内解决
3. 需提供必要的远程网络联机功能, 包含远程桌面联机, VNC…等  
   以及需提供登入系统的操作权限

|  |  |  |
| --- | --- | --- |
| 故障级别 | 故障现象 | 响应时间 |
| 一级故障 | 系统停机，对客户的业务运作有重大影响 | 工程师需立即响应、采用远程控制支持和现场支持方式，4小时内解决 |
| 二级故障 | 系统操作性能严重降级，对客户的业务运作有严重影响 | 采取与一级故障同样的响应方式 |
| 三级故障 | 系统操作性能受损，客户的大部分业务运作仍可正常工作 | 工程师在接到报警后4小时内给予响应、尝试远程诊断和故障排除，必要时到现场进行服务 |
| 四级故障 | 产品功能、安装或配置方面需要信息或支持，对客户的业务运作几乎无影响 | 工程师在接到客户咨询后，必须4个工作日内给予客户答复，并负责跟踪问题的处理过程，解答客户的问题，直至问题解决 |

## 系统交付

### 交付项目

提供如下全部规格。

|  |  |
| --- | --- |
| **文件** | **说明** |
| 系统操作手册 | * 用户使用手册 * 系统安装手册 * 系统部署手册 * 管理员操作手册 * 系统维护手册 |

### 交付方式

软件: 电子版 1份

文件: 纸本印出 1份

语言: 中文

### 授权方式

## 系统实施

系统实施主要通过对系统基础概念，核心模块，系统基本操作等方面的培训，让技术人员，管理人员和应用人员等不同角色的用户理解并熟练使用报工管理系统。

|  |  |  |  |
| --- | --- | --- | --- |
| 序列 | 内容 | 描述 | 时间（人天） |
| 1 | 部署 | 服务器环境部署 | 30 |
| 2 | 部署 | 数据库环境部署 |
| 3 | 部署 | 系统服务器部署 |
| 4 | 建模 | 工厂建模 |
| 5 | 数据 | 基础数据维护 |
| 6 | 测试 | WEB测试 |
| 7 | 测试 | PDA测试 |
| 8 | 测试 | 客户端测试 |
| 9 | 优化 | WEB功能优化 |
| 10 | 优化 | PDA功能优化 |
| 11 | 优化 | 客户端功能优化 |
| 12 | PDA | 安装测试 |
| 13 | PDA | 环境配置 |
| 14 | 客户端 | 安装测试 |
| 15 | 客户端 | 环境配置 |
| 16 | 培训 | WEB端使用培训 | 60 |
| 17 | 培训 | PDA端使用培训 |
| 18 | 培训 | 客户端使用培训 |
| 19 | 运行 | 异常处理 |
| 20 | 运行 | 数据验证 |
| 21 | 运行 | 功能优化 |
| 22 | 陪产 | 数据验证 | 60 |
| 23 | 陪产 | 异常处理 |
| 24 | 陪产 | 异常处理培训 |
| 25 | 陪产 | 数据维护培训 |
| 26 | 陪产 | 系统维护培训 |
| 27 | 陪产 | 功能优化 |

## 项目验收

### 项目验收依据

在此次工程项目中，作为验收依据的主要包括：

* 双方签订的《项目合同书》；
* 双方签订的《项目技术规格说明书》；
* 项目实施中的《项目需求变更报告》；
* 其他项目相关文档及资料。

在这些文档当中，将以双方（用户方、实施方）认可《项目技术规格说明书》和《项目需求变更报告》这两份文档作为主要的验收依据。

### 项目验收时间安排

本项目的验收时间安排为：

在完成系统的安装、配置任务后，并在所有测试运行通过后，申请用户方进行系统的初验。系统正常试运行1个月后，组织对系统进行正式验收

### 项目验收内容

凌世完成章上所述各项工作内容, 并于以下各阶段提交验收报告, 得到XX的签字确认后, 项目即告验收.

1. 需求确认文档签验。

时间: 系统需求收集完成

1. 功能验收报告签验。

时间: 系统开发完成

1. 项目最终验收报告签验。

时间: 系统稳定运行, 并完成培训及其他文件交付

# 产品售后服务

## 服务目标

为适应各工位复杂的环境对管理系统高稳定性支持的要求，凌世提供专业的运维服务团队对管理系统进行5\*8小时支持。

* 通过对管理系统运维项目的功能需求进行综合分析和整体评估，提供系统运维支持整体解决方案。包括持续系统监控，技术支持，预防性维护及灾难恢复流程。
* 配合管理系统上线项目，实施项目的知识转移和培训工作，保证项目和运维转移无缝衔接。
* 运维团队能够和用户IT团队紧密合作，了解业务发生的变化，并对新需求进行收集和分析，并形成不断改进的知识文档。
* 确保系统运维工作符合XX对系统指标要求，含SLA（服务等级协定）、系统可用性。

## 凌世运维服务

系统终验合格之日起1年内，提供软件平台质保。在质保期内，我方将以远程的电话/邮件等方式提供持续高效的日常运维服务。

如果系统出现故障，我方工程师将在接到用户通知后尽快提供服务；我们会对现场服务工程师进行专业培训，能够处理日常异常问题。当现场服务人员无法解决的问题自动升级到凌世运维服务工程师的响应和支持；工程师及现场人员无法解决的问题自动升级为派遣专家到现场的支持。

### 服务形式及内容

* 基础服务，指：专业IT运维服务平台，服务时间内随时待命以保证迅速及时地响应；周期性检查以做预防性维护
* 日常邮件电话服务或现场服务，以服务小时的形式计算，指：针对相关异常/问题以及咨询的解答；甄别应用的功能性故障；帮助分析跟进异常/问题
* 紧急电话/现场服务：对紧急情况的现场或远程的支持

服务的产品只针对智慧能源管理系统，并且这些产品安装在项目合同指定的运行环境下。

### 服务流程

凌世运维整体解决方案符合ITIL V4流程规范：

* 清晰的服务支持模型。描述IT 运维服务台与第二层，第三层的运维支持人员的运维流程模型；
* 系统事件记录，跟踪流程；
* 根本原因分析流程；
* 变更管理流程；
* 系统预防性巡检计划；
* 系统可靠性评价指标及报告

凌世提供的售后服务内容适用以下规定的服务标准和服务等级协议

等级一、紧急情况：系统发生严重故障导致系统崩溃，宕机；自动生产停滞，只能采用手工操作

等级二、重要情况：系统主要功能异常，但是不影响工艺生产；系统资源CPU、内存等消耗逐渐增大，系统运行速度缓慢；系统主要配置数据丢失，需要软件重新配置安装

等级三、标准情况：非主要功能缺陷，不影响系统正常使用；

等级四、次要情况：非功能缺陷，只是界面调整因美观或操作方式优化引起的变更